Antti Oikarinen

Mittausmoduulit SensorTagin antureille ja Yocto-projektin kääntäminen Raspberry Pi:lle

Insinööri (AMK),

tietotekniikka

Kevät 2016

KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

TIIVISTELMÄ

Tekijä(t): Oikarinen Antti

Työn nimi: Mittausmoduulit SensorTagin antureille ja Yocto-projektin kääntäminen Raspberry Pi:lle.

Tutkintonimike: Insinööri (AMK), tietotekniikka

Asiasanat: Yocto, SensorTag, BLE,

Tämän insinöörityön tarkoituksena oli suunnitella mittausmoduulit SensorTagin antureille ja tutustua Yocto-projektiin ja sen mahdollisuuksiin. Työn toteutukseen kuului mittausmoduulien tekeminen SensorTagin antureille ja Yocto-alustan kääntäminen Raspberry Pi:lle. Lisäksi mittausmoduulien tuottama tieto oli tarkoitus tallentaa tietokantaan, joka sijaitsee Raspberry Pi:llä. Mittausmoduulien oli tarkoitus toimia käännetyllä Yocto-alustalla Raspberry Pi:llä. Insinöörityön tilaajana toimi Kajaanin ammattikorkeakoulu.

Työn teoriaosassa käydään läpi Bluetooth-teoriaa, laitetta nimeltä SensorTag ja Yocto-projektia. Bluetooth-teoria jakautuu perinteiseen Bluetoothiin ja Bluetooth Low Energyyn. SensorTagin teoriaosassa esitellään laite nimeltä SensorTag ja sen ominaisuuksia. Yocto-projektin teoriaosuudessa tutustutaan Yocto-projektiin ja siihen, mitä sillä voi tehdä.

Työn lopputuloksena saatiin aikaan toimivat mittausmoduulit SensorTagin antureille ja käännetty Yocto-alusta Raspberry Pi:lle. Käännetyllä Yocto-alustalla pystytään käyttämään luotuja mittausmoduuleja ja tallentamaan mitattu tieto tietokantaan.

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Author(s): Oikarinen Antti

Title of the Publication: Measurement modules for SensorTags sensors and compiling the Yocto environment for Raspberry Pi.

Degree Title: Bachelor of Engineering, Information technology

Keywords: Yocto, BLE, SensorTag

The aim of this Bachelor`s thesis was to design measuring modules for SensorTag sensors and get familiar with the Yocto project and its possibilities. The implementation included designing the measurement modules for SensorTags sensors and compiling the Yocto environment for Raspberry Pi. In addition, the data produced by the measuring modules were to be stored in the database located in Raspberry Pi. The measuring modules were to operate in the Yocto environment. The thesis was commissioned by Kajaani University of Applied Sciences.

The thesis introduces the theories of Bluetooth, the device called SensorTag and the Yocto project. The theory of Bluetooth is divided into Classic Bluetooth and Bluetooth Low Energy. The theory about SensorTags gives an insight into the device called SensorTag and its features. The Yocto project theory introduces the project and its possibilities.

The outcome of this thesis included the functional measurement modules for SensorTag and the compiled Yocto environment for Raspberry Pi. The compiled Yocto environment is capable of using the completed measurement modules and storing the measured data into the database.

ALKUSANAT

Haluan kiittää Eero Huuskoa ja Markku Karppista insinöörityöni aiheesta ja työn ohjeistuksesta. Lisäksi haluan kiittää Asko Kinnusta työn valvomisesta ja Eero Soinista työn kielellisestä ohjauksesta.

SISÄLLYS

1 JOHDANTO 1

2 BLUETOOTH 2
2.1 Historia 2
2.2 Versiot 2
2.3 Toimintaperiaate 3
2.3.1 OSI-malli 3
2.3.2 Protokollapino 5
2.3.3 Fyysinen kerros 7
2.3.4 Taajuushyppely 8
2.3.5 Topologiat 10
2.4 Bluetooth Low Energy 11
2.4.1 Yleistä 11
2.4.2 Virrankulutus 13
2.4.3 BLE-ohjelmistomalli 15
2.4.4 Käyttökohteet 17

3 CC2541 SENSORTAG 21
3.1 SensorTag 21
3.2 Laitteisto 23
3.2.1 Anturit 23
3.2.2 Bluetooth-lähetys 25
3.3 SensorTagin pikakäyttöönotto 26
3.3.1 Käytön valmistelu 26
3.3.2 Laitealustat 26
3.3.3 Ohjelmiston asentaminen 26
3.4 Käyttö valmiin ohjelmiston avulla 27

4 YOCTO-PROJEKTI 28
4.1 Määritelmä 28
4.2 Toiminta 28
4.3 Yocto-projektin ja OpenEmbedded-projektin liitto 29
4.4 Poky 30
4.4.1 BitBake 31
4.4.2 OpenEmbedded-Core 32

4.4.3 Metadata 32
4.5 Etuja 32

5 TYÖN SUORITTAMINEN 34
5.1 Tavoite 34
5.2 Suunnittelu 34
5.3 Kehitysympäristöjen asentaminen 35
5.3.1 Raspberry Pi:n käyttöönotto 35
5.3.2 Yocto-projektin asentaminen 35
5.4 Mittausmoduulien teko 38
5.4.1 Lämpötila-anturi 40
5.4.2 Kiihtyvyysanturi 41
5.4.3 Ilmankosteusanturi 42
5.4.4 Gyroskooppianturi 42
5.4.5 Magnetometrianturi 43
5.5 Tietokannan suunnittelu ja asentaminen 44

6 TYÖN TESTAAMINEN 46
6.1.1 Kytkennän kokoaminen 46
6.1.2 Yhteyden muodostaminen 46
6.1.3 Anturien lukeminen Raspberry Pi:llä 47
6.1.4 Yocto-projekti Raspberry Pi:lle 50
6.1.5 SensorTagin tarvitsemat kirjastot Yocto-projektiin 53
6.1.6 Mittausmoduulien testaaminen Yocto-projektissa 55
6.1.7 Analyysi työstä 59

7 YHTEENVETO 60

LÄHTEET 61

SYMBOLILUETTELO

AES	Advanced Encryption Standard
AFH	Adaptive Frequency Hopping
API	Application programming interface
ARM	Advanced RISC Machines
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
BQP	Bounded error quantum polynomial time
BQRB	Bluetooth Qualification Review Board
DPSK	Differential phase-shift keying
DQPSK	Differential Quadrature Phase Shift Keying
EDR	Enhanced Data Rate
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GFSK	Gaussian frequency-shift keying
GMSK	Gaussian Minimum Shift Keying
GNOME	GNU Network Object Model Environment
GPS	Global Positioning System
HCI	Host Controller Interface
HDMI	High Definition Multimedia Interface
HS	High Speed
IPSP	Internet Protocol Support Profile
IPV4	Internet Protocol version 4
IPV6	Internet Protocol version 6
ISI	Intersymbol Interference
ISM	Industrial, Scientific and Medical

ISO	International Organization for Standardization
L2CAP	Logical Link Control and Adaption Protocol
LL	Link layer
LLC	Logical link control
LMP	Link Manager Protocol
MAC	Media Access Control
MIPS	Microprocessor without Interlocked Pipeline Stages
OSI	Open Systems Interconnection
PHY	Physical layer
PSK	Phase Shift Keying
RAM	Random Access Memory
RF	Radio frequency
SD	Secure Digital
SIG	Bluetooth Special Interest Group
SPI	Serial Peripheral Interface
SSH	Secure Shell
SSP	Secure Simple Pairing
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
UUID	Universally unique identifier
WLAN	Wireless local area network

1 JOHDANTO

Nykypäivänä erilaisien antureihin perustuvien langattomien mittausjärjestelmien määrä on kasvanut ja niiden käyttökohteet ovat lisääntyneet. Tekniikan kehittyessä on tullut mahdolliseksi luoda langaton mittausjärjestelmä minne vain. Mittausjärjestelmät voivat toimia pelkällä nappiparistolla pitkiä aikoja ja mitata vain mittauskäskyn tullessa.

Bluetooth Low Energy eli BLE on yksi mahdollisista tiedonsiirtovaihtoehdoista suunniteltaessa langatonta mittausjärjestelmää. BLE:n tarjoamia etuja ovat vähentynyt virrankulutus, standardisoitu sovelluskehitysarkkitehtuuri ja matalat kehitys- ja käyttökustannukset.

Mittausjärjestelmät tarvitsevat toimiakseen käyttöjärjestelmän. Aina mittausjärjestelmän resurssit eivät kuitenkaan mahdollista isojen käyttöjärjestelmien käyttöä. Tällöin mittausjärjestelmän kehittäjä voi haluta kehittää käyttöjärjestelmänsä itse. Yksi vaihtoehto kehittämiseen on Yocto-projekti. Yocto-projekti tarjoaa mahdollisuuden kääntää juuri omiin tarpeisiin sopiva käyttöjärjestelmä.

Insinöörityön aiheeni sain Kajaanin ammattikorkeakoululta Eero Huuskolta ja Markku Karppiselta. Tämän insinöörityön tarkoituksena on toteuttaa mittausmoduulit SensorTag-laitteen antureille ja tutustua Yocto-projektiin ja sen mahdollisuuksiin. Mittausmoduulien on tarkoitus mitata SensorTagin antureita ja tallentaa antureilta saatu tieto tietokantaan. Tämän lisäksi insinöörityössä on tavoitteena päästä testaamaan mittausmoduuleja käännetyllä Yocto-alustalla.

2 BLUETOOTH

2.1 Historia

Vuonna 1994 Ericsson tutki erilaisia mahdollisuuksia langattomaan tiedonsiirtoon puhelimien ja tietokoneiden välillä. Tästä sai alkunsa Bluetooth. Ericsson otti vuonna 1997 neljä muuta isoa tekniikanalan yhtiötä mukaan kehitystyöhönsä. Yritykset olivat IBM, Intel, Nokia ja Toshiba. Niiden tarkoituksena oli luoda avoin spesifikaatio lyhyen kantaman radioteknologialle. Tekniikan oli tarkoitus korvata kaapelointi laitteiden välillä. Tekniikkaa kuvattiinkin yksinkertaiseksi, halvaksi, luotettavaksi ja tehon kulutukseltaan pieneksi. Yritysten ryhmittymää alettiin kutsua nimellä SIG eli Special Interest Group. Nykyään SIG:ssä on yli 20 000 jäsentä ja jäsenmäärä kasvaa koko ajan. [1, s. 325.],[2.]

2.2 Versiot

Bluetoothin ensimmäiset versiot 0.8 ja 0.9 olivat vielä salaisia. Varsinainen suositus julkaistiin heinäkuussa 1999, ja sen versionumero oli 1.0. Versio sisälsi aika paljon virheitä, joiden takia julkaistiinkin joulukuussa 1999 versio 1.1. Suositus 1.1 ei sisältänyt juurikaan uusia asioita, vaan korjasi ison määrän version 1.0 virheitä. Versio 1.1 oli kaikkien vapaasti ladattavissa. Tämän jälkeen julkaistut suositukset ovat laitevalmistajia ja Bluetooth SIG:n jäseniä varten. Marraskuussa 2003 julkaistiin versio 1.2 päivittämään versio 1.1. Sen tärkein parannus oli taajuushyppely (AFH). [1, s. 325.]

Vuonna 2004 julkaistiin versio 2.0, jonka suurin parannus oli EDR (Enhanced Data Rate). EDR nosti mahdollisen tiedonsiirtonopeuden kolminkertaiseksi eli kolmeen Mbit/s, vaikkakin käytännössä nopeus jää 2,1 Mbit:iin/s. Versio 2.1 julkaistiin vuonna 2007. Se toi mukanaan SSP- (Secure Simple Pairing) tekniikan parantamaan Bluetoothin tietoturvallisuutta. [3.]

Vuonna 2009 julkaistiin versio 3.0, jossa tiedonsiirtonopeus kasvoi teoriassa 24 Mbit:iin/s High Speed tekniikan avulla. Tiedon siirtoa ei kuitenkaan suorita

Bluetooth-linkki, vaan Bluetooth-linkki suorittaa yhteyden neuvottelun ja perustamisen ja data siirretään 802.11g standardin WLAN-verkon avulla. [3.]

Vuonna 2010 julkaistiin versio 4.0, joka sisälsi protokollat Classic Bluetooth, Bluetooth High Speed (HS) ja Bluetooth Low Energy (BLE). Version suurin parannus oli vähentynyt virrankulutus BLE:n ansiosta. [3.]

Versio 4.1 ei tuonut mukanaan laiteuudistuksia, vaan päivitykset olivat kaikki ohjelmistoon liittyviä, joten kaikki 4.0-versiota tukevat Bluetooth-laitteet voidaan päivittää tukemaan 4.1-versiota. Päivitykset paransivat käytettävyyttä ja toivat ohjelmistonkehittäjille uusia mahdollisuuksia. Päivitys sisälsi myös laitteille mahdollisuuden käyttää IPv6:n mukaista IP-osoitetta entisen IPv4:n sijasta. Näin osoitteiden määrä kasvoi huomattavasti. [3.]

Versio 4.2 julkaistiin joulukuussa vuonna 2014. Versio toi mukanaan parannuksia tiedonsiirtonopeuksiin, turvallisuuteen ja IPSP:n (Internet Protocol Support Profile), joka mahdollistaa Bluetooth Smart-anturien yhdistämisen suoraan Internetiin IPv6/6LoWPAn avulla. Päivitys sisältää myös laitteistopäivityksen, mutta joitakin sen ominaisuuksista saa käyttöön myös vanhemmassa versiossa ohjelmistopäivitysten myötä. [3.]

2.3 Toimintaperiaate

2.3.1 OSI-malli

Vuonna 1977 kansainvälinen standardisoimisjärjestö ISO (International Organization for Standardization) asetti komitean luomaan tietoliikenteelle toimintamalli. Kehitystyön tuloksena syntyi OSI-viitemalli (Open Systems Interconnection). Viitemallin tavoitteena oli luoda arkkitehtuuri, jossa tietoliikennejärjestelmä on jaettu kerroksiin. Pääideana oli, että jokainen kerros n tuottaa palveluja kerrokselle n+1 ja käyttää hyväkseen kerroksen n-1 tuottamia palveluita. Kuvassa 1 on esitelty OSI-mallin kerrokset. [1, s. 7.]

Kuva 1. OSI-malli [4].

OSI-viitemallin muodostaa seitsemän eri kerrosta, jotka ovat fyysinen kerros, siirto-, verkko-, kuljetus-, istunto-, esitystapa- ja sovelluskerros. Kerrosten määrittelyn perusteena olivat seuraavat vaatimukset.

- Rajoitetaan kerrosten lukumäärää siten, että niiden kuvaaminen ja integrointi on mahdollisimman yksinkertaista.
- Muodostetaan kerrosten väliset rajapinnat sellaisiin kohtiin, joissa kerrosten väliset yhteydet ovat minimissään.
- Sijoitetaan eri kerroksiin sellaiset aiheet, joiden tekniikka poikkeaa toisistaan. Ei esimerkiksi sekoiteta samaan kerrokseen toimintoja, jotka liittyvät sekä siirrettävän tiedon esitysmuotoon että sen sähköiseen koodaukseen.
- Yhdistetään samanlaiset toiminnot samaan kerrokseen.
- Mahdollistetaan muutokset kerroksiin ilman, että ne vaikuttavat alapuolella ja yläpuolella oleviin kerroksiin.
- Muodostetaan kerroksen rajapinnat siten, että kerros liittyy vain ja ainoastaan ylä- ja alapuolella oleviin kerroksiin.

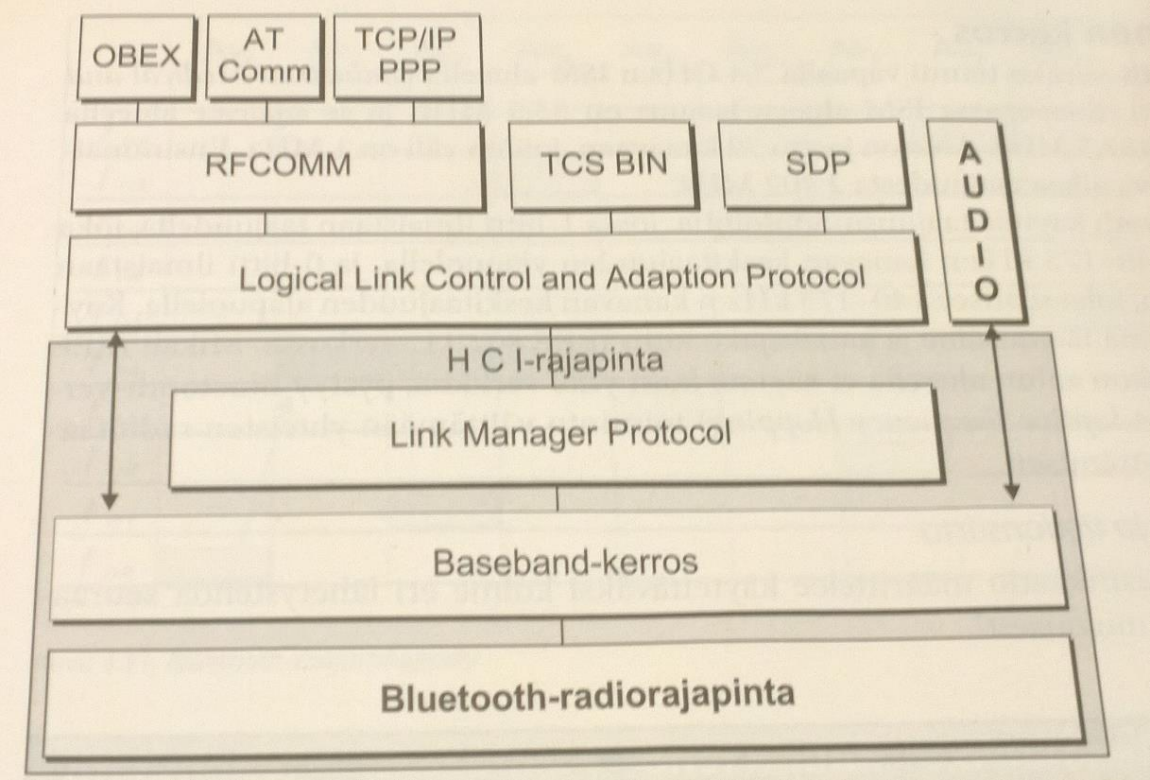
Fyysinen kerros on tietoliikenneyhteyden näkyvä osa kuten kaapeli. Se määrittelee siirtoyhteyden fyysiset, mekaaniset ja toiminnalliset ominaisuudet. Kerroksen tehtävänä on muuttaa bittejä sähköisiksi pulsseiksi, valoksi tai radiotien signaaleiksi.

Siirtoyhteyskerroksen tehtävänä on luoda yhteys kahden pisteen välille, korjata mahdolliset siirtovirheet ja purkaa yhteys. Kerros jakautuu kerroksen sisällä MAC- ja LLC-kerrokseen. MAC-kerroksessa varataan siirtoyhteys tiedonsiirtoa varten, ja LLC-kerroksen tehtävänä on havaita virheet, korjata virheet ja huolehtia, että fyysiselle kerrokselle ei anneta siirrettäväksi enempää tietoa kuin siirtotie tai vastaanottaja voi käsitellä.

Verkkoyhteyskerros tarjoaa yhteyden, joka on riippumaton verkon rakenteesta ja kytkentätekniikasta. Kuljetuskerroksen tarkoituksena on tarjota kahden päätepisteen välille tiedonsiirtoyhteys. Istuntokerros huolehtii sovellusten välisistä ohjaustoiminnoista, esimerkiksi yhteyden muodostamisesta, siihen liittyvien siirtoyhteyspalveluiden varaamisesta, yhteyden ominaisuuksien sopimisesta osapuolien välillä, yhteyden varmistamisesta tarkastuspisteillä, yhteyden lopettamisesta ja resurssien vapauttamisesta. Esitystapakerroksessa sovitaan tiedon esitystapa päätelaitteiden välillä. Sovelluskerroksessa tarjotaan sovelluksille rajapinta OSI-järjestelmään. [1, s. 8–11.]

2.3.2 Protokollapino

Protokollapino Bluetooth-spesifikaation mukaan kattaa fyysisen kerroksen ja siirtoyhteyskerroksen. Isäntäkone, johon on sulautettu Bluetooth, ohjaa tietoliikennettä HCI-rajapinnan kautta. Kuvassa 2 on esitelty Bluetooth-protokollapino. [1, s. 326.]



Kuva 2. Bluetooth-protokollapino [1].

OSI 1 -kerroksella ovat radiotie ja osa Baseband-kerroksesta. OSI 2 -kerroksen sisältämät toiminnot sisältyvät sekä Baseband-kerrokseen että LMP- ja L2CAP-kerroksiin. Huomattavaa on, että Bluetooth ei noudata puhtaasti OSI-kerrosmallia. Raja OSI 1- ja OSI 2 -kerroksen välillä kulkee keskellä Baseband-kerrosta. Bluetooth-arkkitehtuuri ei myöskään noudata kerrosarkkitehtuurin perusperiaatteita. Perusperiaatteen mukaan yksi kerros keskustelee vain naapurikerroksiensa kanssa, mutta Bluetooth sallii osalle toiminnoistaan luvan keskustella muidenkin kerrosten kanssa. Esimerkiksi puheen siirto on tällainen toiminto. [1, s. 327.]

OSI-mallin ylemmät kerrokset ovat Bluetoothissa valmistajakohtaisia. Tuotteen on käytävä läpi sertifiointiprosessi BQP, jotta se saa luvan käyttää Bluetooth-tuotemerkkiä. Sertifiointiprosessin suorittaa Bluetooth SIG:n alaisuudessa toimiva BQRB (Bluetooth Qualification Review Board). [1, s. 327.]

2.3.3 Fyysinen kerros

Bluetooth toimii vapaalla 2,4 GHz:n ISM-alueella (Industrial Medical and Scientific). Euroopassa taajuusalueena toimii 2400–2483,5 MHz, joka on jaettu 79 kanavaan, joiden väli on 1 MHz. Tiedonsiirtoon Bluetooth käyttää erilaisia modulointimenetelmiä ja taajuushyppelyä. Alun perin taajuusmodulointiin käytettiin GFSK-menetelmää (Gaussian frequency shift keying). GFSK-modulaatiossa binaarinen 1 edustaa positiivista taajuuspoikkeamaa ja 0 negatiivista taajuuspoikkeamaa. Moduloitu signaali suodatetaan käyttäen suodatinta, jolla on Gaussin-vastekäyrä, jotta liian kaukana päästökaistasta olevat signaalit eivät pääse läpi. Suodatin, joka noudattaa Gaussin-käyrää, on siis kaistanpäästösuodatin. Gaussin-suodattimen käyttötarkoituksena on se, että suodatin ”hidastaa” moduloidun signaalin taajuusmuutoksia. Hypättäessä nopeasti 0-tilaa vastaavasta taajuudesta 1-tilaa vastaavaan taajuuteen tapahtuu nopea taajuuden muutos. Nopeat taajuuden muutokset kasvattavat taajuuskaistaa ja voivat aiheuttaa häiriöitä muille kanaville. Gaussin suodattimella pehmennetään näitä taajuuden muutoksia ja pienennetään kunkin kanavan taajuuskaistaa. [5.]

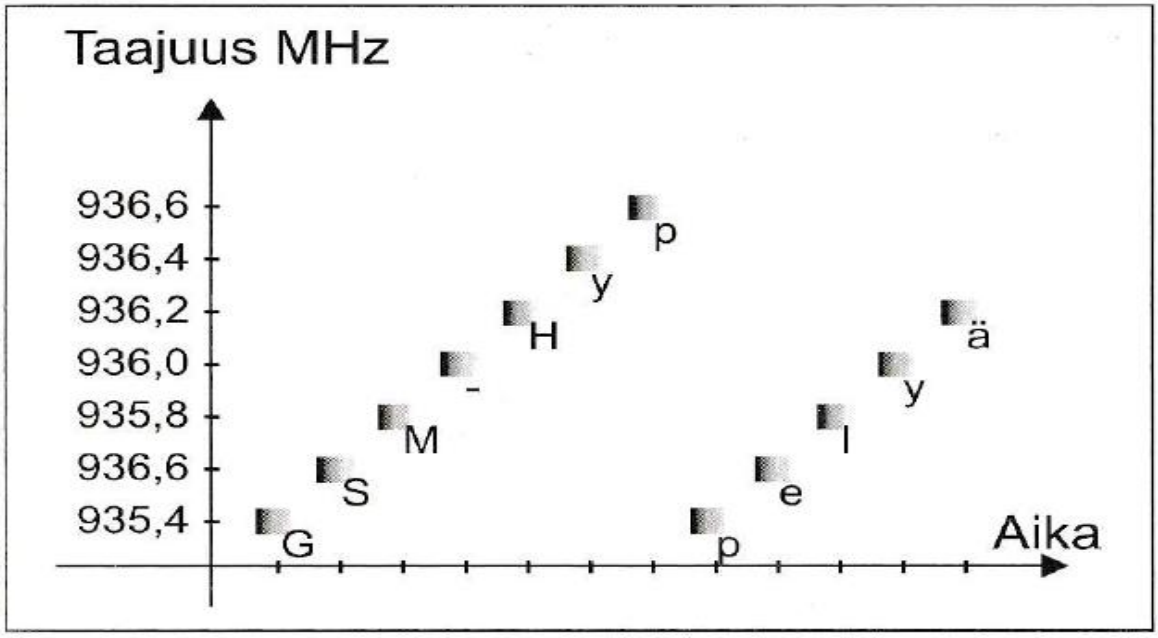
Kasvavien tiedonsiirtomäärien myötä Bluetooth 2 -versiossa otettiin käyttöön EDR-ominaisuus. EDR-ominaisuuden avulla tiedonsiirtonopeus jopa kolminkertaistui 3 Mbit/s:ssa. Modulointiin EDR käyttää PSK-modulaation eli vaiheavainnusmodulaatiomenetelmän kahta muotoa, π/4 DQPSK- ja 8DPSK-modulaatiota. 2 Mbit/s nopeus saadaan aikaan π/4 DQPSK-menetelmän avulla, ja 3 Mbit/s saavutetaan 8DPSK-menetelmällä yhteyden ollessa riittävän hyvä. [5.]

Bluetooth 3 -versiossa tiedonsiirtonopeus kasvoi teoriassa jopa 24 megabittiin sekunnissa High Speed -tekniikan avulla. Data kuitenkin siirretään 802.11g-standardin WLAN-verkon avulla. Bluetooth-linkki suorittaa vain yhteyden neuvottelun ja perustamisen. [5.]

2.3.4 Taajuushyppely

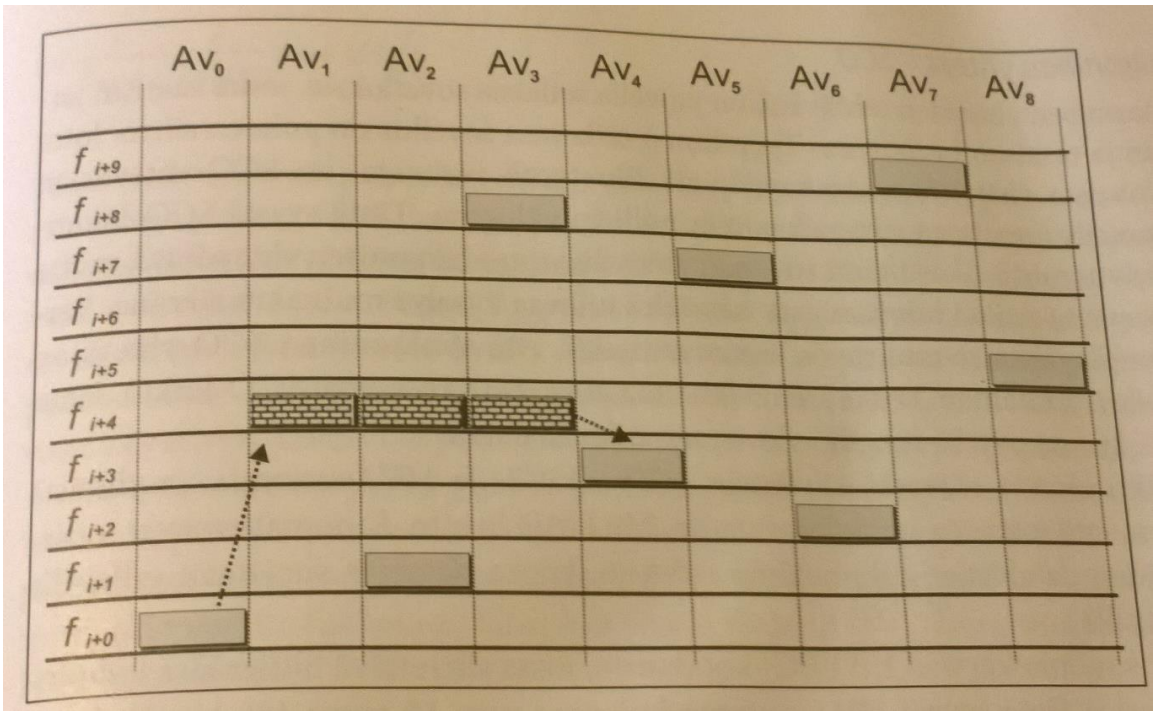
Tiedonsiirto Bluetooth-tekniikassa perustuu taajuushyppelyyn. Taajuushyppely on hajaspektritekniikoihin kuuluva, pääasiassa langattomissa järjestelmissä käytettävä menetelmä. Taajuushyppelystä on monenlaista hyötyä. Radiotiellä liikkuvia taajuuksia ja niiden käyttäytymistä ei voida ennustaa. Radioaaltojen käyttäytymiseen vaikuttavat sekä taajuus että paikka. Monitie-etenemisestä johtuvat häiriöt voidaan keskiarvoistaa hyppimällä taajuusalueelta toiselle. Taajuushypyn tulisi tapahtua ainakin 7 kanavan päähän edellisestä kanavasta, jotta saatava hyöty olisi tarpeeksi suuri. Myös ulkoiset häiriöt voidaan keskiarvoistaa, koska radiotiellä esiintyvät häiriöt ovat aikaan ja taajuuteen sidottuja. Siis häiriö esiintyy hetkellisesti joillakin tietyllä taajuudella. Taajuushyppely vaikeuttaa salakuuntelua, koska hyppelyjärjestys on tiedettävä voidakseen seurata tietoliikennettä. Hyppely on niin vauhdikasta, että vaikka hyppelyjärjestyksen tietää, sen seuraaminen on mahdotonta. Taajuushyppelyssä siirtoaika jaetaan kiinteän mittaisiin aikaväleihin. Käytettävissä oleva siirtokaista jaetaan kiinteisiin alikanaviin, ja sovitaan näennäissatunnaisesta hyppelyjärjestyksestä, jota lähettäjä ja vastaanottaja noudattavat. [1, s. 116–117.]

Tiedonsiirron aikana data pilkotaan pienempiin osiin, ja jokainen osa siirretään omalla taajuudellaan sovitun hyppelyjärjestyksen mukaisesti. Taajuushyppelyn periaate on esitelty kuvassa 3. [1, s. 117.]



Kuva 3. Taajuushyppely [1].

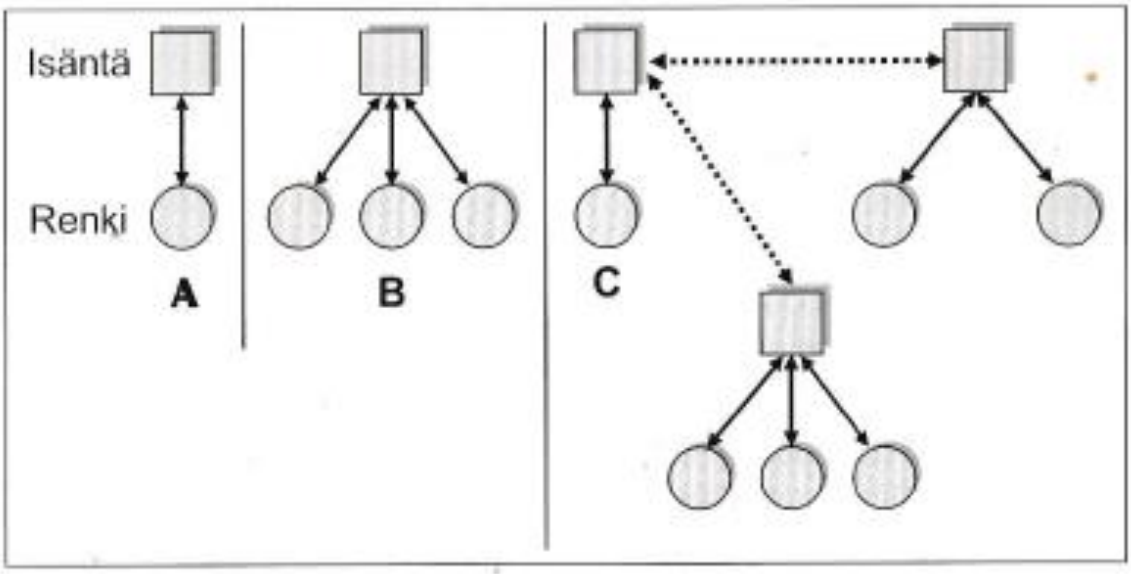
Erikoista Bluetooth-verkon taajuushyppelyssä on se, että sanomat, joiden pituus on suurempi kuin yksi aikaväli, siirtyvät kokonaisuudessaan samalla taajuudella. Hyppely siirtyy tämän jälkeen taajuudelle, jolle se olisi mennyt, mikäli sanoma olisi ollut yhden aikavälin mittainen. Periaate hyppelystä on esitelty kuvassa 4. Verkossa olevat muut laitteet hyppelevät normaalin kaavan mukaisesti, jos sanoman siirto jatkuu samalla taajuudella pidempään kuin yhden aikavälin. Ainoastaan isäntä ja sanoman lähettäjä/vastaanottaja jatkavat samalla taajuudella. [1, s. 116–117, 328–329.]



Kuva 4. Bluetooth-taajuushyppely [1].

2.3.5 Topologiat

Bluetooth toimii isäntä-renki-periaatteella: yksi laite toimii isännän roolissa ja muut laitteet ovat renkejä. Isäntälaite on laitteena aivan samanlainen kuin muutkin verkon laitteet, se on vain ottanut roolikseen olla isäntä. Pienimmillään Bluetooth-verkon voi muodostaa isäntä ja renki. Tällainen verkko on käytännössä kaksipisteyhteys (Point To Point), jossa samanlaiset laitteet välittävät tietoja toisilleen. Kuvassa 5 kohdassa A on esitelty tämänlainen topologia. Bluetoothissa voi olla aktiivisena kerrallaan vain 8 laitetta, mutta siinä voi olla iso määrä passiivisia laitteita, joita isäntä tarvittaessa herättää. Kuvassa 5 kohdassa B esitellään topologia, jossa verkossa on yksi isäntä ja kolme renkiä. Useamman isännän muodostama verkko on esitelty kuvassa 5 kohdassa C. [1, s. 325–326.]



Kuva 5. Bluetooth-topologiat [1].

2.4 Bluetooth Low Energy

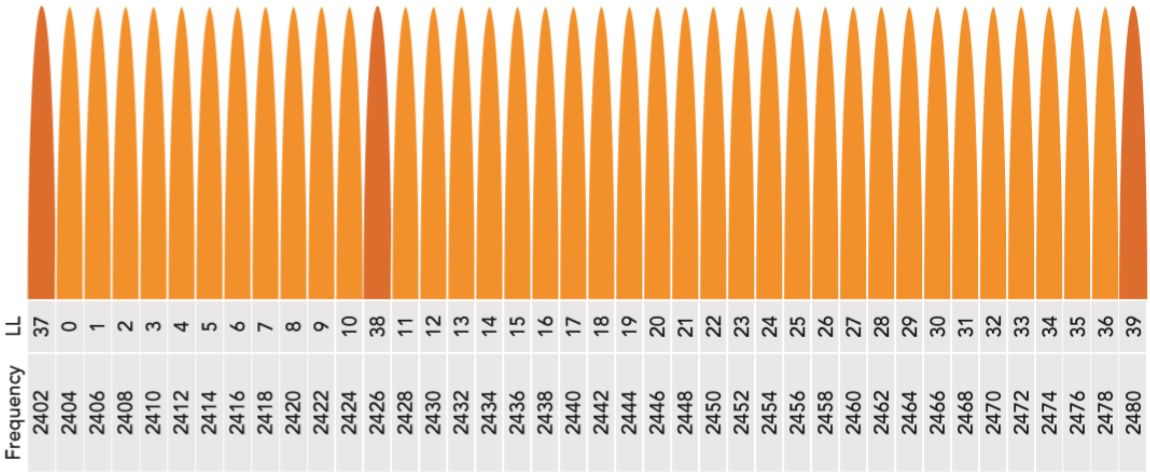
2.4.1 Yleistä

Bluetooth Low Energy (BLE) sai alkunsa Bluetooth 4.0 -versiosta. Sitä kutsuttiin alun perin nimellä Wibree, joka oli teollisuuden standardi mahdollistamaan langaton yhteys pienten laitteiden välille. Wibree-tekniikkaa kehitti alun perin Nokian tutkimuskeskus saadakseen pienitehoiset laitteet käyttämään Bluetooth-tekniikkaa kommunikointiin. [6.]

BLE:n tärkeimmät uudistukset ovat vähentynyt virrankulutus huippu-, keskiarvo- ja lepotilassa, standardoitu sovelluskehitysarkkitehtuuri, kyky toimia nappiparistolla jopa vuosia, matalat kehitys- ja käyttökustannukset, usean valmistajan laitteiden yhteensopivuus ja 128-bittinen AES-datansalaus. [7.]

BLE käyttää taajuusalueenaan ISM-taajuusaluetta (2,4–2.483,5 GHz), kuin myös Classic Bluetooth. Taajuuskaista on jaettu 40:een 2 MHz:n kanavaan. BLE käyttää GFSK-modulaatiota ja taajuushyppelyä 40 kanavallaan saavuttaen 1

Mbit/s tiedonsiirtonopeuden. Kuvassa 6 on esitelty BLE:n kanavat. Kolmea kanavista käytetään yhteydenmuodostamiseen ja loppuja 37 kanavaa tiedon lähettämiseen. Yhteyden muodostamiseen tarkoitetut kanavat näkyvät kuvassa 6 tummennettuina. Kanavien paikat on valittu tarkasti WLAN-kanavien väliin. [8.],[9.]



Kuva 6. BLE:n kanavat. Tummennetut kanavat ovat mainostuskanavia [8].

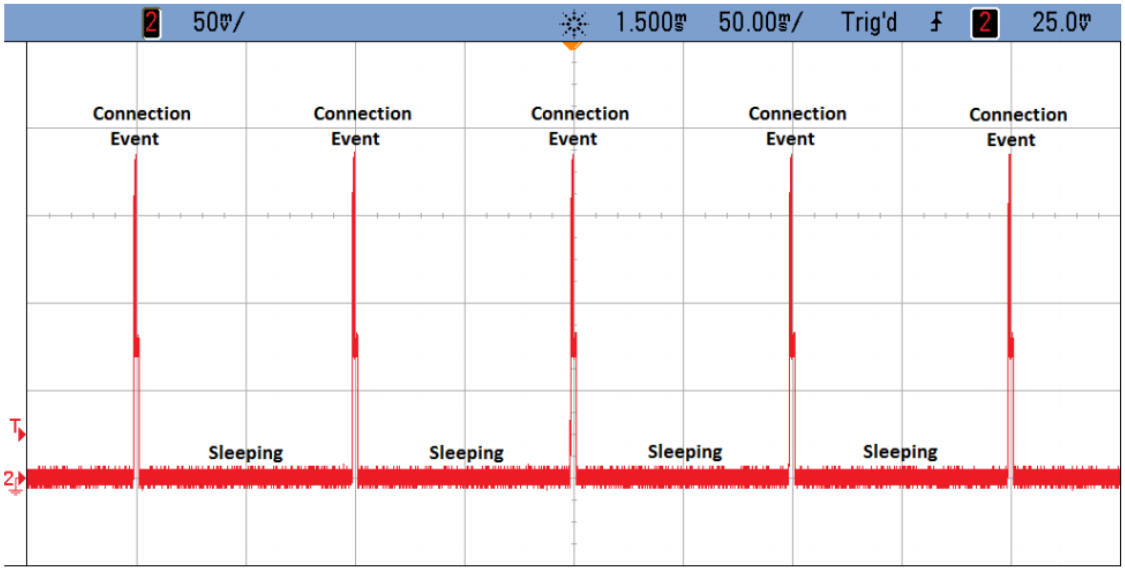
BLE:n tarvitsee skannata vain kolmea kanavaa etsiäkseen muita laitteita. Skannaaminen saadaan suoritettua 0,6–1,2 millisekunnissa. Classic Bluetoothin vastaavasti täytyy skannata 32 kanavaa, joka vie joka kerta 22,5 millisekuntia. Pelkästään tämän keinon avulla BLE säästää 10–20-kertaisen määrän virtaa. Kuitenkaan 3 kanavan skannaaminen ei ole pelkästään positiivista. Mahdollisuus siihen, että toinen radiolähetin lähettää samalla taajuudella, kasvaa, joten myös signaalin tuhoutumisen riski kasvaa. Tämä on syy, miksi Bluetooth SIG on valinnut yhteyden muodostamiseen käytettävien kanavien paikat tarkasti WLAN-kanavien väliin kuvassa 6 näkyviin kohtiin: välttääkseen skannaussignaalien sekoittumisen WLAN-signaaleihin. [8.],[9.]

Yhteyden muodostaminen tapahtuu BLE-laitteilla seuraavasti: Mainostaja lähettää tietoja ilman BLE-yhteyttä. Skanneri skannaa mainostajien lähetyksiä. Alustaja on laite, joka vastaa mainostajalle pyytäen yhteyttä. Mainostajan hyväksyessä yhteyspyynnön syntyy yhteys. Yhteydessä ollaan joko orja tai isäntä. Laitteesta, joka alustaa yhteyden tulee, isäntä, ja laitteesta, joka hyväksyy

yhteyspyynnön, orja. Laitteella voi olla useita eri yhteyksiä, joten ne voivat toimia molemmissa rooleissa sekä isäntänä että orjana. [10.]

2.4.2 Virrankulutus

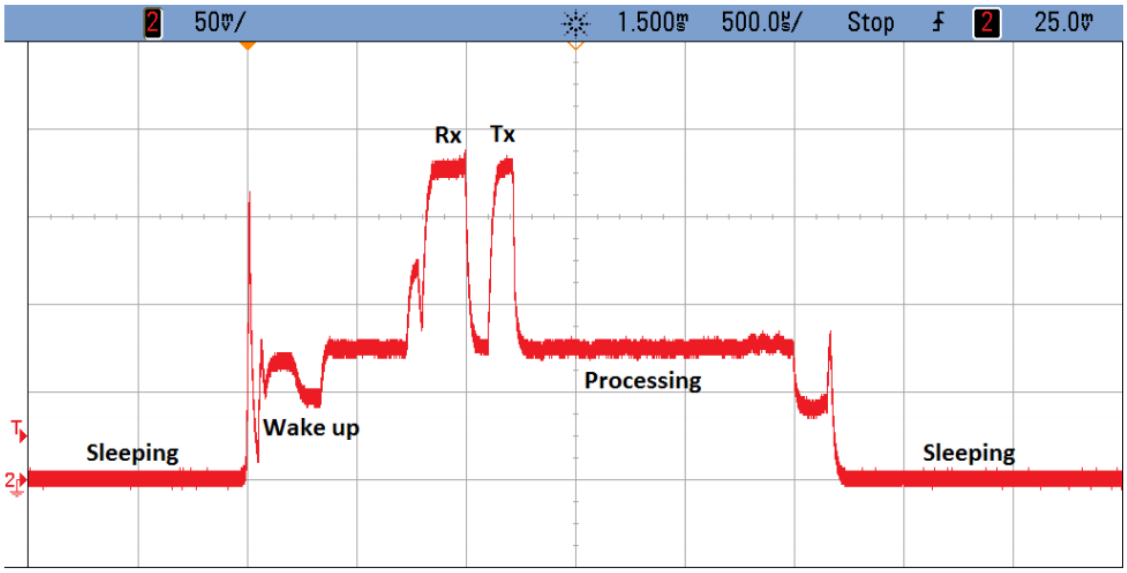
BLE säästää virtaa kytkemällä radiolähettimen päälle vain hyvin lyhyeksi aikaa kerrallaan. Siis BLE-laite lähettää tietoa vain hyvin pienen osan ajastaan ollessaan kytkettynä. Kuvasta 7 nähdään virrankulutuksen olevan huipussaan lähetystapahtuman aikana. BLE-laitteiden virrankulutuksen arviointi pelkän huippuvirrankulutuksen avulla on ongelmallista, koska laite on lähetystilassa vain pienen ajan kerrallaan. Tarkemman arvion virrankulutuksesta saa arvioimalla, kuinka usein yksittäinen lähetystapahtuma tulee toistumaan. [11.]



Kuva 7. Virrankulutus BLE-yhteydessä [11].

Kuvassa 8 on esitelty yksittäinen BLE-lähetystapahtuma virrankulutuksen näkökulmasta. Kuvasta huomataan, että BLE-laitteella on useita eri tiloja, jossa sen täytyy käydä suorittaakseen lähetysrutiinin. Lähetysrutiinin, joka sisältää laitteen etsimisen, tietojen lähettämisen, yhteyden varmistamisen ja yhteyden katkaisemisen, BLE pystyy suorittamaan 3 millisekunnissa. Kuvassa 8 tämä rutiini on lepotila (Sleeping), herääminen (Wake Up), tiedon vastaanottaminen (Rx), tiedon lähettäminen (Tx), käsittely (Processing) ja paluu lepotilaan

(Sleeping). Vastaavasti Classic Bluetooth -laitteelta saman rutiinin suorittaminen vie yleensä satoja millisekunteja, mikä vie prosessina moninkertaisen määrän energiaa. [9.],[11.]



Kuva 8. Virrankulutus yksittäisessä yhteydessä [11].

Kuvasta 8 huomataan, että yksittäisen lähetystapahtuman sisällä virrankulutus ja ajankäyttö vaihtelevat eri tilojen välillä. Tiedon vastaanottaminen ja lähettäminen kuluttavat eniten virtaa. Herääminen ja tiedonkäsittely kuluttavat nekin paljon virtaa verrattuna lepotilaan. Lepotilassa oleminen ei taas kuluta virtaa juurikaan. Huomioon on kuitenkin otettava se seikka, että lepotilassa laite ei skannaa mainostajien lähetyksiä eikä tee itseään näkyväksi muille laitteille. Kokonaisvirrankulutus BLE-järjestelmässä riippuu siis hyvin pitkälti siitä, kuinka usein laite on herätettävä lähettämään tai vastaanottamaan tietoa. [11.]

BLE ja Classic Bluetooth käyttävät molemmat GFSK-modulaatiota, mutta siinä on pieni eroavaisuus. Classic Bluetooth käyttää modulaatioindeksiä 0,35 ja BLE 0,45–0,50. BLE:n käyttämä modulaatioindeksi on lähellä GMSK:n (Gaussian minimum shift keying) tasoa. Tämän avulla saavutetaan pienempi virrankulutus ja myös BLE:n kantama paranee verrattuna Classic Bluetoothiin. [8.],[11.]

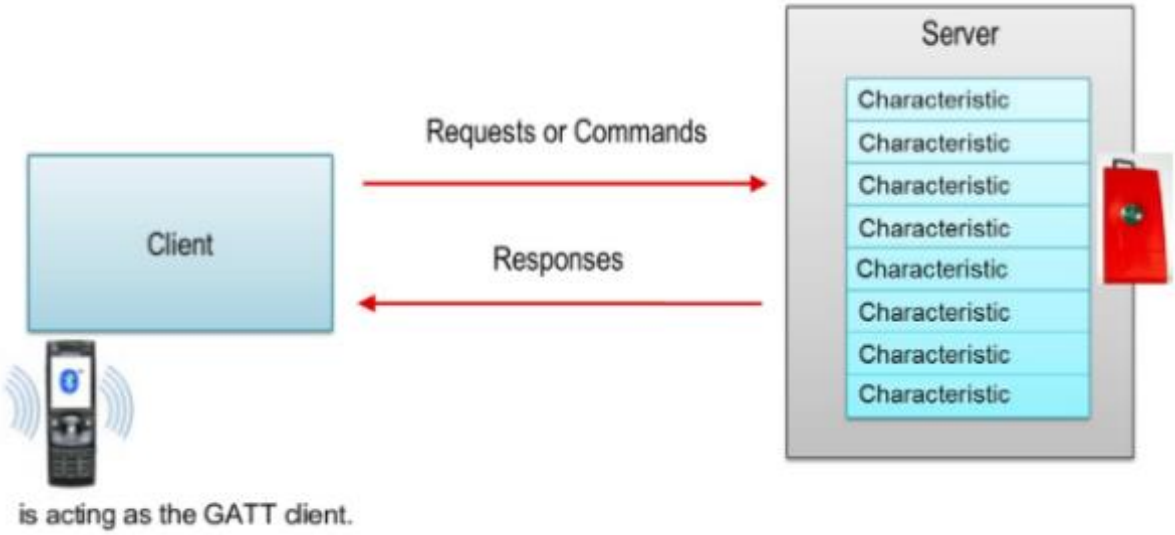
Vähentyneen virrankulutuksen ansiosta laitteet, jotka käyttävät BLE-tekniikkaa voivat käyttää virtalähteenään myös nappiparistoja. Muun muassa seuraavat sovellusalueet ovat alkaneet hyödyntää BLE-tekniikkaa: älypuhelimet, kodin

elektroniikka, anturiverkot, terveys ja liikunta, terveydenhoito, mittausjärjestelmät, lelut, kauko-ohjaus ja puettavat laitteet kuten älykellot. Monien sovelluksien on vähentyneen virrankulutuksen ansiosta mahdollista toimia jopa vuosia ilman akun tai pariston vaihtamista. Tämän takia BLE-tekniikka on toimiva ratkaisu esimerkiksi langattomissa anturijärjestelmissä. [7.]

2.4.3 BLE-ohjelmistomalli

BLE-applikaatioiden ohjelmointirajapinta perustuu tyypillisesti GATT-konseptiin, joka määrittelee kaksi roolia: asiakas ja palvelin. GATT-roolit eivät ole välttämättä sidottuja tiettyyn GAP-rooliin, vaan ne voidaan määritellä ylemmän kerroksen profiilien avulla. Laite voi myös toimia molemmissa rooleissa, toisessa yhteydessä asiakkaana ja toisessa palvelimena. [12.]

Asiakkaan tehtävä on aloittaa GATT-komennot ja pyynnöt sekä hyväksyä vastaukset. Palvelin vastaanottaa GATT-komennot ja pyynnöt sekä palauttaa vastauksen asiakkaalle. Toiminta on esitelty kuvassa 9. Asiakas eli kännykkä lähettää komentoja tai pyyntöjä SensorTagille eli palvelimelle, ja palvelin vastaa komentoihin ja pyyntöihin. [12.]



Kuva 9. GATT-asiakas ja -palvelin [12].

Ominaisuuksia (Characteristic) tulee ajatella ryhmänä informaatiota, joita kutsutaan attribuuteiksi. Attribuutit ovat perustietoa, jota siirretään laitteiden välillä. Ominaisuudet järjestävät ja käyttävät attribuutteja data-arvoina, ominaisuuksina ja rakenteen tietoina. [12.]

Tyypillinen ominaisuus koostuu seuraavista attribuuteista:

- Ominaisuuden arvo: Tämä arvo on ominaisuuden data-arvo.
- Ominaisuuden määrittely: Määrittelijä säilyttää ominaisuuden ominaisuudet, sijainnin ja ominaisuuden arvon tyypin.
- Asiakkaan ominaisuuden konfiguraatio: Tämä konfiguraatio mahdollistaa GATT-palvelimen määrittää, mitkä ominaisuudet lähetetään GATT-palvelimelle tai mitkä ominaisuudet lähetettynä GATT-palvelimelle vaativat kuittauksen.
- Ominaisuuden käyttäjän kuvaus: Kuvaus on ASCII-merkkijono, joka kuvaa ominaisuutta.

Nämä attribuutit tallennetaan attribuuttitaulukkoon GATT-palvelimelle. Seuraavat ominaisuudet liittyvät jokaiseen attribuuttiin.

- Kahva. Tämä ominaisuus on attribuutin indeksi taulukossa. Jokaisella attribuutilla on uniikki kahva.
- Tyyppi. Tämä attribuutti kuvaa, mitä attribuutin data edustaa. Tätä attribuuttia kutsutaan termillä UUID (Universal unique indentifier). UUID määritellään osittain Bluetooth SIG:ssä, ja loput siitä on käyttäjän määriteltävissä.
- Oikeudet. Tämä attribuutti valvoo ja kertoo, miten GATT-asiakaslaite voi käyttää attribuutin arvoa.

Kokoelmasta ominaisuuksia, jotka yhdessä suorittavat tietyn tehtävän, muodostuu palvelu. Bluetooth-profiili muodostuu yhdestä tai useammasta palvelusta. [12.]

2.4.4 Käyttökohteet

Bluetoothin käyttökohteet ovat laajentuneet huomattavasti Bluetooth Low Energyn eli BLE:n myötä.

Mobiilipuhelimet

Melkein kaikissa kännyköissä ja älypuhelimissa on Bluetooth-tekniikkaa, joka mahdollistaa näiden laitteiden kommunikoinnin esimerkiksi handsfree-systeemin kanssa. Uudemmat älypuhelimet ovat lähes kaikki BLE-tuella varustettuja. Nykyään on jo vaikeampaa löytää älypuhelinta ilman BLE-tukea. BLE:n ansiosta älypuhelin voidaan yhdistää lukemattomaan määrään BLE-laitteita. [13.]

Tietokoneet ja oheislaitteet

Tietokoneissa BLE-tekniikkaa voidaan hyödyntää monella eri tavalla. Näppäimistö, hiiri, ja tulostin voivat kaikki olla langattomia ja kommunikoida keskusyksikön kanssa Bluetooth 4.0 -yhteyden avulla. Kannettavat tietokoneet ovat yleensä varustettuja BLE-tuella, ja sen takia onkin yleistä käyttää BLE-yhteensopivia laitteita kannettavan tietokoneen kanssa. Viime vuosina hurjasti suosiotaan kasvattaneet tabletit kommunikoivat paljon BLE:n välityksellä. Tietokoneet ja tabletit ovat erinomainen ratkaisu tiedon keräämiseen esimerkiksi langattomista antureista. [14.]

Autoteollisuus

Autoteollisuus on tasaisesti kasvava markkina-alue BLE-laitteille. Autoissa on tällä hetkellä 10–15 tapaa hyödyntää BLE-tekniikkaa, esimerkiksi handsfree, musiikin toistaminen, älypuhelinapplikaatiot, auton diagnostiikka, navigaatio, etäisyysanturit ja viihdejärjestelmät.

BLE-handsfree-laite on nykyään vakiolaitteena uusissa ajoneuvoissa. Huolet turvallisuudesta sekä tiukentuneet lait koskien puhelimen käyttöä ajaessa lisäävät handsfree-laitteiden myyntiä. Kaikki maailman suurimmat ajoneuvojen valmistajat tarjoavat BLE-handsfree-laitteita ajoneuvoihinsa. BLE:n etu handsfree-toteutuksissa on sen halpuus, yksinkertaisuus ja helppokäyttöisyys.

BLE:n avulla myös oman laitteen kytkeminen auton äänentoistolaitteistoon on helppoa. Kuljettaja voi kytkeä esimerkiksi älypuhelimensä BLE:n avulla auton äänentoistojärjestelmään ja kuunnella mitä haluaa. Älypuhelinsovellukset ovat muutenkin tulleet osaksi autojen viihdejärjestelmiä. Niiden avulla voidaan kuunnella musiikkia ja katsoa videoita sekä applikaatioita voidaan käyttää auton kosketusnäytön kautta. Applikaatiot voivat auttaa esimerkiksi navigoinnissa, säätiedoissa, liikennemäärissä ja muissa sovelluksissa, jotka parantavat ajokokemusta. Autonvalmistajat ovat myös alkaneet kehittää sovelluksia autojaan varten. Sovellusten avulla voidaan esimerkiksi tarkkailla auton mekaanisia ja elektronisia vikoja. Auton valmistajien kannalta tämän tyyppisellä kehityksellä voidaan vähentää kuparijohtojen tarvetta autoissa langattomien antureiden ansiosta. Näin autot ovat kevyempiä, polttoaineenkulutus laskee ja autot ovat halvempia valmistaa. Langattomilla antureilla voi olla tulevaisuudessa mahdollisuus tarkkailla kuljettajan terveyttä ja vireystilaa ajonaikana. [15.]

Varoittimet ja vähittäismyynti

BLE-tekniikkaan perustuvat varoittimet, etenkin Ibeacon, ovat yleistymässä. Varoittimia käytetään esimerkiksi isoissa kauppakeskuksissa. Kauppakeskuksen sisällä suunnistamiseen GPS on liian epätarkka tekniikka. Varoittimien avulla kuluttajan on mahdollista suunnistaa kauppakeskuksessa yhdistämällä puhelimensa BLE:n avulla varoittimiin. Kauppiaat hyötyvät varoittimista voiden kohdentaa markkinointia jo kauppakeskuksessa oleviin asiakkaisiin. Asiakas voi saada ollessaan esimerkiksi elektroniikkaosastolla puhelimeensa erikoistarjouksia elektroniikkatuotteista. [16.]

Kuluttajaelektroniikka

Kuluttajaelektroniikassa BLE-sovellusmahdollisuudet ovat rajattomat. Niitä voidaan hyödyntää liikenteessä ja kotona. Langattomuuden ansiosta laitteiden määrä kasvaa koko ajan. Käyttökohteita ovat muun muassa puettavat laitteet, langaton äänentoisto ja televisiot. Langaton äänentoisto on nykyään jo arkipäivää, puhelimen avulla voidaan toistaa musiikkia kodin viihdejärjestelmästä. Puettavat laitteet ovat yksi uusista BLE:n käyttöalueista. Langattomuus yhdistettynä puettavaan elektroniikkaan ja näiden helppo

yhdistettävyys älylaitteisiin on lisännyt laitteiden kehittäjien mahdollisuuksia huomattavasti. [17.]

Terveys ja hyvinvointi

Turvallinen, luotettava ja tehokas BLE on jo tällä hetkellä käytössä miljoonissa terveys- ja hyvinvointilaitteissa ja määrän odotetaan kasvavan tulevaisuudessa. BLE on hyvä langaton ratkaisu terveys- ja hyvinvointiapplikaatioihin niin sairaalassa kuin kotona. 128 bittinen salaus takaa korkean turvallisuuden verkossa, pitäen potilaan sekä mittausten tiedot salattuna lähetyksen ajan. Taajuushyppely estää lähes kokonaan verkon häiriöt ja tarjoaa vahvan suojan muista elektronisista laitteista tulevia häiriöitä vastaan.

Laitteet, joissa BLE:tä hyödynnetään, voivat olla esimerkiksi verenpainemittareita, sykemittareita, astmainhalaattoreita ja stetoskooppeja. BLE:n etu näissä laitteissa on helppo yhdistettävyys ja mahdollisuus lähettää antureilta saatu data BLE-laitteelle. Näiden tekniikoiden avulla myös kotihoidossa olevalta potilaalta on helpompi saada tiedot lääkärin käyttöön. [18.]

Urheilu ja kuntoilu

BLE:n ansiosta antureiden pienentyneen koon ja vähentyneen virrankulutuksen ansiosta antureita voidaan hyödyntää entistä paremmin urheilussa. Käyttökohteita antureille urheilussa ovat niin urheilijat kuin myös heidän varusteensa. Antureiden toimiminen kuukausia tai jopa vuosia pelkällä nappiparistolla on johtanut siihen, että markkinoilla on paljon laitteita, joilla kuluttaja voi seurata omaa harjoitteluaan. Laitteet voivat kerätä tietoa fyysisestä suorituksesta ja lähettää datan suojatusti älypuhelimelle tai tabletille. Antureiden pienen koon ansiosta niitä voidaan myös liittää urheilussa käytettäviin välineisiin, esimerkiksi tennismailaan tai juoksijan piikkareihin. Tietoa voidaan analysoida suorituksen jälkeen tai jopa sen aikana. Valmentajat voivat analysoida tietoa kesken harjoituksen ja muokata harjoittelua kerätyn datan perusteella. [19.]

Älykodit

BLE on ideaalinen kodin automatisointiin. Ongelmana kodin automatisoinnissa on ollut johtojen paljous, BLE poistaa tämän ongelman. Kodin laitteet voidaan yhdistää turvallisesti BLE:n avulla älypuhelimeen, tablettiin tai tietokoneeseen. Tämän ansiosta asukas voi hallita kodin valoisuutta, turvallisuutta ja energian kulutusta kätevästi älylaitteensa avulla. Näin voidaan kontrolloida esimerkiksi kodin lämpötilaa kätevästi parilla napin painalluksella. [20.]

3 CC2541 SENSORTAG

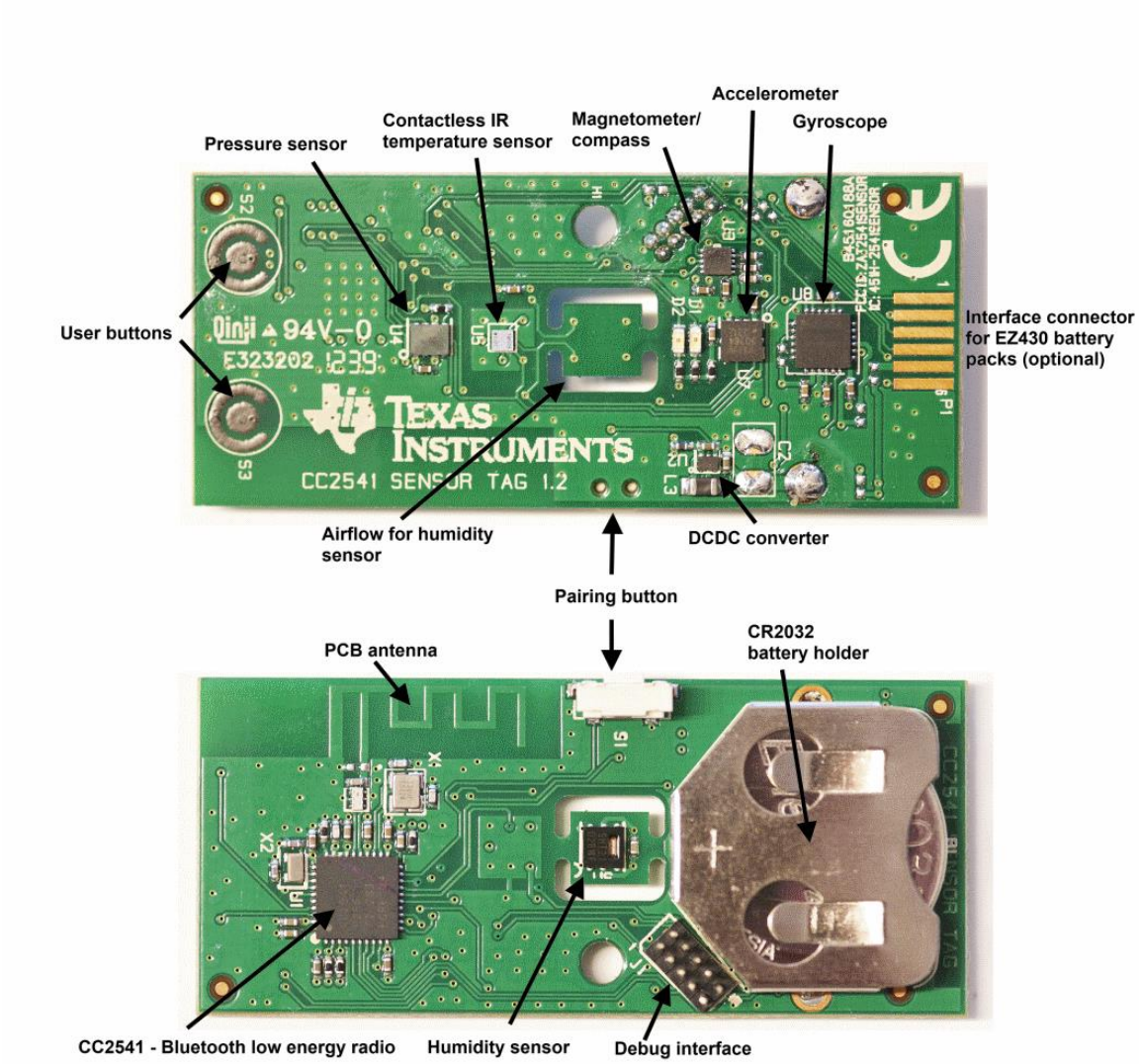
3.1 SensorTag

CC2541 SensorTag on Texas Instrumentsin valmistama vähävirtainen Bluetooth-kehitystyökalu, joka keskittyy langattomiin anturisovelluksiin ja on suunniteltu ensisijaisesti älypuhelinsovelluksien kehittäjille. SensorTagin on tarkoitus helpottaa sovellusten kehittäjien työtä. Kehitystyökalun avulla kehittäjien ei tarvitse käyttää aikaa laitteiston rakentamiseen, vaan kehitystyön voi aloittaa heti. SensorTagin käyttäminen ei siis edellytä yksinkertaisimmillaan lainkaan tuntemusta sulautetuista järjestelmistä tai laitteista. Kuvassa 10 on SensorTag pakkauksensa päällä. [21.]

Kuva 10. SensorTag.

Peruspakkaus, jossa SensorTag myydään, sisältää CC 2541 SensorTag -laitteen kotelolla, CR2032-pariston ja käyttöohjeet. SensorTag kommunikoi Bluetooth 4.0 -yhteyden avulla muiden laitteiden kanssa. Tämän ansiosta

älypuhelimen ja SensorTagin yhdistäminen kommunikoimaan keskenään on helppoa. SensorTag toimii 2,4 GHz:n taajuusalueella. SensorTag-piirilevy on esitelty kuvassa 11. [22.]



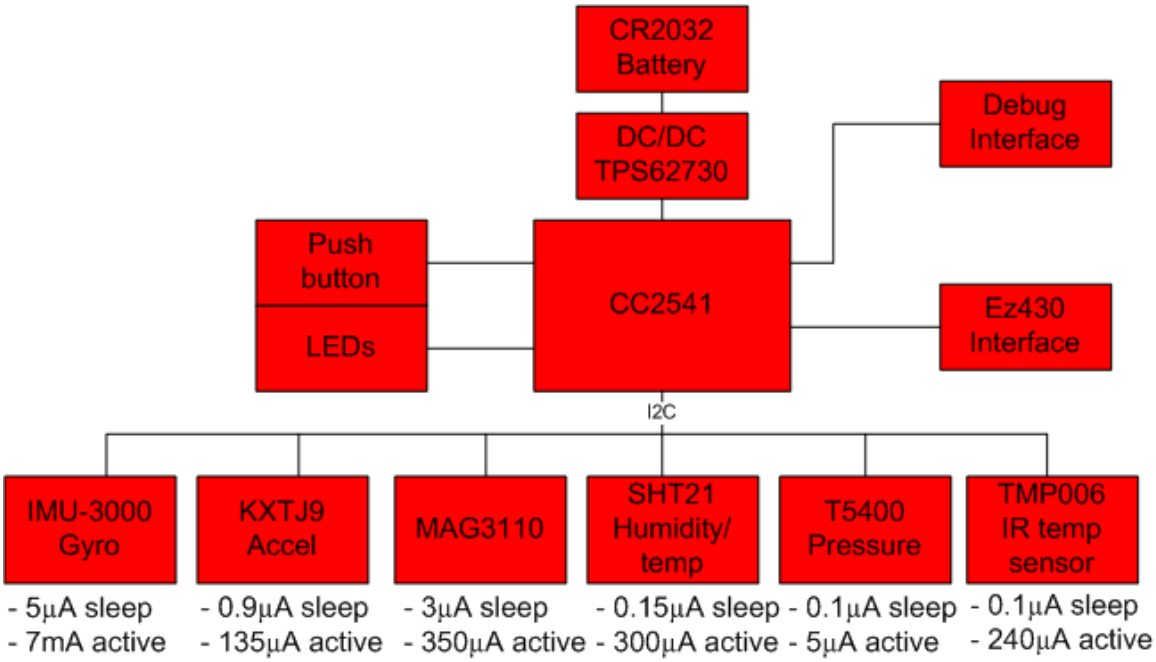
Kuva 11. SensorTag-piirilevy [23].

SensorTag-piirilevy sisältää kiihtyvyys-, lämpötila-, kosteus-, gyroskooppi-, paine- ja magnetometrianturit. Muita laitteita, joita SensorTagin kanssa voi hankkia, ovat cc-Debugger, cc 2540 USB-Dongle ja akkupaketti. [22.]

3.2 Laitteisto

3.2.1 Anturit

SensorTagin antureiksi on valittu pienikokoisia ja halpoja antureita, jotka käyttävät i2c-väylää, ja ne on yhdistetty kaikki samaan väylään. Anturit erotetaan toisistaan väylästä aktiivisen signaalin perusteella. Virran säästämiseksi kaikki anturit ovat oletuksena poissa päältä, ja mittausten välissä anturit ovat lepotilassa. Anturit ja niiden virrankulutus on esitelty kuvassa 12. Dataa voidaan lukea joko yhdeltä tai useammalta anturilta yhtä aikaa. [23.]



Kuva 12. Anturit ja niiden virrankulutus [23].

Kiihtyvyysanturi

SensorTag sisältää Kionix KXTJ9 -kiihtyvyysanturin. Tämä kiihtyvyysanturi mittaa kiihtyvyyttä X-, Y- ja Z-akselin suuntaisesti. Anturilla voidaan myös mitata painovoiman suuntaa. Mittaustarkkuus on 14 bittiä, ja anturi käyttää I2C-väylää. Virrankulutus on lepotilassa 10 µA, alhaisilla mittaustarkkuuksilla 10 µA ja korkeilla mittaustarkkuuksilla 325 µA. Käyttöjännitealue on 1,8–3,6 V ja

toimintalämpötila-alue -40–80 °C. Anturilla voidaan mitata kiihtyvyyttä 2g-, 4g- tai 8g-tarkkuudella. Mittaustarkkuuden päättää käyttäjä. Anturia käytetään pääasiassa sulautetuissa järjestelmissä. [24.]

Lämpötila-anturi

TMP006-lämpötila-anturi mittaa lämpötilaa tyypillisesti 1 °C tarkkuudella. Anturilla voidaan mitata mittauskohteen lämpötilaa, vaikka anturilla ei ole kontaktia mittauskohteeseen. Mittausetäisyys riippuu mittauskohteen koosta. Datalehti suosittelee mittausetäisyydeksi mittauskohteen säde jaettuna kahdella. Esimerkiksi jos mittauskohteen säde on 2 metriä, niin mittausetäisyyden tulisi olla 1 metri. Anturi toimii I2C-väylässä, ja sen virrankulutus on tyypillisesti 240 µA ja lepotilassa 1µA. Käyttöjännitealue on 2,2–3,6 V ja toimintalämpötila-alue -40–125 °C. Vähäinen virrankulutus mahdollistaa anturin käyttämisen sovelluksissa, jotka saavat virtansa nappiparistosta. [25.]

Ilmankosteusanturi

Sensirion SHT21-ilmankosteusanturi mittaa suhteellista ilmankosteutta ja lämpötilaa. Ilmankosteutta mitataan 12 bitin tarkkuudella ja lämpötilaa 14 bitin resoluutiolla. Käyttöjännitealue on 2,1–3,6 V ja toimintalämpötila-alue -40–125 °C. Anturi mittaa suhteellisen kosteuden kahdeksassa sekunnissa. [26.]

Gyroskooppianturi

IMU-3000 gyroskooppi mittaa pyörimistä 3 akselin suuntaisesti (x, y ja z) 16 bitin tarkkuudella. Gyroskoopilta saatava data voidaan yhdistää kiihtyvyysanturin dataan, ja näin saadaan selville SensorTagin sijainti 3D-tasossa. Käyttöjännitealue on 2,1–3,6 V ja toimintalämpötila-alue -40–85 °C. Anturin virrankulutus on 5,9 mA, kun mitataan kolmea akselia yhtä aikaa. Lepotilassa anturin virran kulutus on 5 µA. [27.]

Ilmanpaineanturi

Epcos T5400 mittaa ilmanpainetta 16 bitin mittaustarkkuudella. Anturilla on i2C- ja SPI-liitäntä. Tämän ansiosta T5400 voidaan liittää melkein mihin mikrokontrolleriin tahansa. Anturin koko on 2,78 millimetriä × 2,23 millimetriä. Virrankulutus on 0,1 µA lepotilassa ja aktiivisena 5 µA. Pienen koon ja alhaisen virrankulutuksen ansiosta anturia käytetään muun muassa puhelimissa, korkeusmittareissa ja navigaatiolaitteissa. Käyttöjännitealue on 1,7–3,6 V ja toimintalämpötila-alue -30–85 °C. Painetta voidaan mitata väliltä 300–1100 hPa. Anturi sisältää 16-bittisen AD-muuntimen. [28.]

Magnetometrianturi

Freescale MAG3110 -magnetometrianturi mittaa magneettikentän suuntaa x-, y- ja z-akseleiden suuntaisesti. Anturilta saatavaa dataa voidaan käyttää kiihtyvyysanturin datan kanssa ja näin saadaan aikaan kompassi. Magnetometrillä voidaan mitata maan magneettikenttiä, mutta myös elektroniikassa syntyviä magneettikenttiä. Anturin näytteistystaajuus on 80 kHz, ja I2C-väylä toimii 400 kHz:n taajuudella. Magnetometrin käyttöjännitealue on 0,95–3,6 V ja toimintalämpötila-alue -40–85 °C. [29.]

3.2.2 Bluetooth-lähetys

SensorTag käyttää yleensä virtalähteenään nappiparistoa. Lähetys käynnistetään tai lopetetaan painamalla sivunappia. Lähetyksen ollessa käynnissä SensorTagissa vilkkuu vihreä led. Yhteyden ollessa päällä napin painamisella yhteys katkeaa, anturit palaavat alkutilaan ja kaikki data-arvot asetetaan nollaksi. [23.]

SensorTagissa on vähävirtainen Bluetooth-lähetinpiiri, jossa on 256 kilobittiä flash-muistia ja 8 kilobittiä RAM-muistia. Laite on esiohjelmoitu vähävirtaisella Bluetooth-laiteohjelmistolla antureille ja laite pystyy kommunikoimaan Bluetooth 4.0 -version avulla muiden laitteiden kanssa. [23.]

3.3 SensorTagin pikakäyttöönotto

3.3.1 Käytön valmistelu

Avataan paketti, jonka sisällä on CC 2541, kotelo, CR2032-nappiparisto ja käyttöohjeet. Ensimmäisenä asennetaan nappiparisto paikalleen. Asennettaessa nappiparisto ensimmäisen kerran kiinni tulisi paristo irrottaa ja kiinnittää muutaman kerran, että saadaan varmasti kontakti. Käynnistetään SensorTag painamalla sivunappia ja vihreän ledin tulisi alkaa vilkkua. Sitten laitetaan punainen suojakotelo SensorTagin ympärille. Antureille ei tarvitse tehdä mitään, vaan ne käynnistyvät automaattisesti. [22.]

3.3.2 Laitealustat

SensorTagin käyttämiseen tarvitaan tällä hetkellä Bluetooth 4.0 -yhteys. Yhteyden voi muodostaa esimerkiksi älypuhelimesta tai tabletista. Tämän Bluetooth 4.0 -laitteen tulee tukea Bluetooth-vähävirtaista -API:a. Yhteys toimii 2,4 GHz taajuusalueella. SensorTag tukee tällä hetkellä esimerkiksi seuraavia laitealustoja: iPhone 5, iPad (3) tai uudempi, iPod Touch (5. sukupolvi), MacBook Air (2011-malli ja uudemmat), MacBook Pro (2012-malli ja uudemmat), Windows 7/XP PC CC2540 USB -laitteen avulla, Bluetooth 4.0 -lähetintä käyttäen BLE Device Monitor, osaa TI BLE stack v1.30 ja uudemmat, Android, Windows Phone 8.1, Raspberry Pi ja Blackberry 10. [22.]

3.3.3 Ohjelmiston asentaminen

Tällä hetkellä SensorTagille löytyy valmiina IOS- ja Android-ohjelmisto. IOS-laitteelle ohjelmiston voi ladata Apple Storesta. Ohjelman nimi on TI Bluetooth SensorTag. Ohjelma löytyy myös käyttäen hakusanaa SensorTag. Androidille ohjelman voi ladata esimerkiksi Google Play -kaupasta. Ohjelman nimi on Bluetooth SensorTag. Ohjelman avulla voi lukea SensorTagin antureita. [22.]

3.4 Käyttö valmiin ohjelmiston avulla

Laitteessa, jolla SensorTagia aikoo käyttää, tulee laittaa Bluetooth 4.0 päälle. SensorTagissa painetaan sivunapilla lähetys päälle. Tämän toiminnan huomaa vihreän ledin vilkkumisesta. Avataan SensorTag-ohjelma, joka etsii automaattisesti kaikki vähävirtaiset Bluetooth-laitteet alueella. Laitteen löytyessä valikkoon ilmestyy SensorTag-logo. Jos laitetta ei löydy, SensorTag menee lepotilaan 30 sekunnin kuluttua. Uudestaan yhdistämiseen tulee painaa SensorTagin sivunappia uudestaan. [22.]

4 YOCTO-PROJEKTI

4.1 Määritelmä

Linux Foundation -työryhmä määrittelee Yocto-projektin seuraavasti: Yocto-projekti tarjoaa avoimen lähteen, korkealaatuisen infrastruktuurin ja työkalut helpottaakseen kehittäjiä luomaan heidän omat kustomoidut Linux-jakelunsa riippumatta laitearkkitehtuurista usealle eri markkina-alueelle. Yocto-projektin tarkoituksena on tarjota helppo aloituspiste kehittäjille. [30, s. 7.]

Yocto-projekti on avoimen lähdekoodin yhteistyöprojekti, joka tarjoaa malleja, työkaluja ja menetelmiä helpottamaan kustomoidun Linuxiin perustuvan systeemin luomista sulautetuissa laitteissa välittämättä laitteistoarkkitehtuurista. Linux Foundationin isännöidessä projektia projekti säilyttää riippumattomuuden sen jäsenorganisaatioihin, jotka osallistuvat projektiin eri tavoilla ja tuottamalla resursseja projektille. [30, s. 7.]

Yocto-projekti kehitettiin vuonna 2010 yhteistyössä monien laitevalmistajien, avoimen lähdekoodin käyttöjärjestelmien, tavarantoimittajien ja elektroniikkayhtiöiden toimesta. Kehittäjien tavoitteena oli vähentää samankaltaista työtä ja tuottaa resursseja sekä informaatiota uusille ja kokeneille käyttäjille. Yksi näistä resursseista on OpenEmbedded-Core, ydinsysteemin komponentti, jonka tarjoaa OpenEmbedded-projekti. Yocto-projekti on sen vuoksi yhteisön avoimen lähteenprojekti, joka liittää yhteen useita yrityksiä, yhteisöjä, projekteja ja työkaluja keräten yhteen ihmisiä, joilla on tarkoituksena rakentaa Linuxiin perustuva sulautettujärjestelmä. [30, s. 7.]

4.2 Toiminta

Yocto-projektia voidaan lähestyä ajattelemalla sitä tietojenkäsittelylaitteena. Sisään syötettävä data kuvailee mitä halutaan, se on sisään syötettävä spesifikaatio. Ulos tulevaksi dataksi halutaan Linux-pohjainen sulautettu järjestelmä. Jos ulos tuleva tuote toimii Linux-pohjaisella käyttöjärjestelmällä,

generoitu lopputulos koostuu osista käyttöjärjestelmää, kuten Linux-ytimestä, alkulatausohjelmasta (bootloader) ja juuritiedostojärjestelmästä (root file system bundle), jotka on laitettu oikeaan järjestykseen.

Yocto-projektin työkalut ovat läsnä jokaisessa työvaiheessa valmistaakseen juuritiedostojärjestelmäpaketin ja muut suoritteet. Aikaisemmin käännettyjen apuohjelmien ja muiden ohjelmistokomponenttien uudelleenkäyttö on maksimissaan, kun käännetään toista applikaatiota, kirjastoa tai ohjelmistokomponenttia oikeassa järjestyksessä valittujen asetuksien kanssa sisältäen tarvittavien lähdekoodien hakemisen luotettavista lähteistä, kuten Linux Kernel Archives (www.kernel.org), GitHub ja www.SourceForge.net.

Yocto-projekti valmistaa käännösympäristön, apuohjelmat ja työkaluketjut, joten sen riippuvuus isäntäohjelmistoihin on vähentynyt, mutta mikä tärkeämpää, sen determinismisyys on kasvanut. Eli Yocto-projektin käännöstyökalu ei välitä, mistä siihen syötettävä data on peräisin, vaan ainoastaan suorittaa tehtävänsä. Apuohjelmat, versiot ja asetukset ovat samat, mutta riippuvuus isäntäohjelmiin on minimoitu.

Yocto-projekti muodostuu useammasta projektista, joita ovat esimerkiksi Poky, Bitbake ja OpenEmbedded-Core. Kaikilla näillä on oma osansa Yocto-projektissa, ja ne täydentävät toisiaan. [30, s. 8.]

4.3 Yocto-projektin ja OpenEmbedded-projektin liitto

OpenEmbedded-projekti syntyi tammikuussa vuonna 2003, kun osa OpenZaurusprojektin ydinkehittäjistä alkoi työskennellä uuden käännösjärjestelmän parissa. OpenEmbedded-käännösjärjestelmä on alusta alkaen saanut inspiraationsa tehtäväaikatauluttajasta ja perustunut Gentoo Portagen -pakettijärjestelmään nimeltä Bitbake. Projektin ohjelmistokokoelma ja tuettujen laitteiden määrä on kasvanut nopeasti.

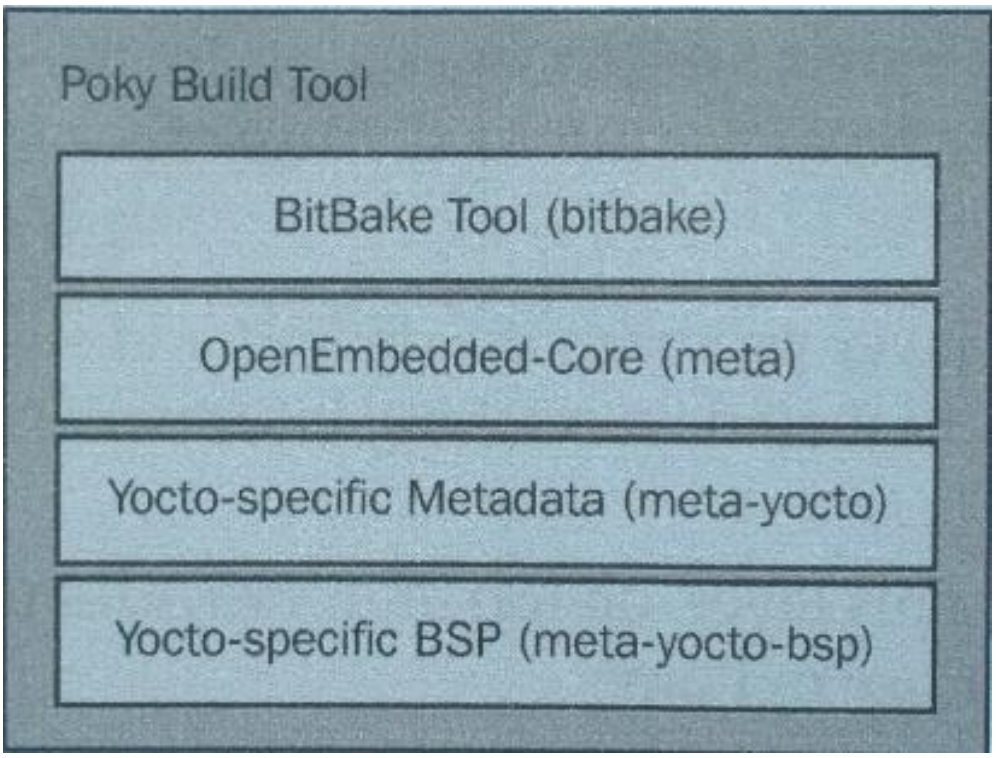
Koordinoimattoman kehityksen seurauksena OpenEmbedded-projektia on vaikeaa käyttää tuotteissa, jotka vaativat vakaan ja viimeistellyn koodipohjan.

Seurauksena tästä on syntynyt Poky. Poky aloitettiin osana OpenEmbedded-projektia, ja se oli viimeistellympi ja vakaampi koodipohja eri laitearkkitehtuureille. Tämä laski Pokyn kynnystä alkaa kehittää korkeampilaatuisia teknologioita, kuten IDE-liitännäisiä ja QEMU-integraatiota.

Marraskuussa 2010 julkaistiin Yocto-projekti jatkamaan tätä työtä Linux Foundationin sponsoroimana projektina. Yocto-projekti ja OpenEmbedded-projekti yhdistivät aikaansaannoksensa ydinkäännösjärjestelmän kehityksessä. Tätä alettiin kutsumaan nimellä OpenEmbedded-Core, joka käyttää parhaita osia Pokysta ja OpenEmbedded-projektista korostaen lisääntynyttä lisäosien ja metadatan käyttöä. [30, s. 10.]

4.4 Poky

Poky on Yocto-projektin viitejärjestelmä, ja se koostuu kokoelmasta työkaluja ja metadataa. Se on alustariippumaton ja suorittaa ristiin käännöksen käyttäen BitBake-työkalua, OpenEmbedded-ydintä ja valittua oletusdataa. Se tarjoaa mekanismin kääntää ja koota tuhansia avoimen lähteen jakeluprojekteja muodostamaan täysin kustomoidun, valmiin ja yhtenäisen Linux-ohjelmistopinon. Pokyn päätehtävä on tarjota kaikki mahdolliset toiminnot, joita kehittäjä tarvitsee. Pokyn rakenne on esitelty kuvassa 13. Pokyn perusosat ovat BitBake-työkalu, OpenEmbedded ydin, meta-yocto ja meta-yocto-bsp. [30, s. 8–9.]



Kuva 13. Poky-käännöstyökalun rakenne [30, s. 9.].

4.4.1 BitBake

BitBake on tehtäväaikatauluttaja, joka jäsentää koodia, jossa on Pythonia ja komentorivi komentosarjoja sekaisin. Parsittu koodi generoi ja ajaa tehtäviä, jotka ovat yksinkertaisesti kasa askeleita järjestyksessä, joka perustuu koodin riippuvuuksiin. [30, s. 9.]

BitBake arvioi kaikki saatavilla olevat asetustiedostot ja metadatan. Se käsittelee dynaamisten muuttujien laajentumisen, riippuvuudet ja koodin generoinnin. BitBake pitää huolta, että kaikki tehtävät tulevat käsiteltyä oikeassa järjestyksessä varmistaakseen tehtävien valmistumisen, maksimoiden prosessin resurssien käytön vähentääkseen käännösaikaa ja ollakseen ennustettava. BitBake:n kehittäminen on keskittynyt *bitbake-devel@lists.openembedded.org* sähköpostilistalle ja sen koodi löytyy Pokyn alihakemistosta. [30, s. 9.]

4.4.2 OpenEmbedded-Core

OpenEmbedded-Coren metadatakokoelma tuottaa moottorin Pokyn käännöstyökalulle. Se on suunniteltu tuottamaan ydintoiminnot, ja sen tulee olla mahdollisimman puhdas. Se tuottaa tuen viidelle eri prosessoriarkkitehtuurille, joita ovat ARM, x86, x86-64, PowerPC, MIPS ja MIPS64. [30, s. 9.]

4.4.3 Metadata

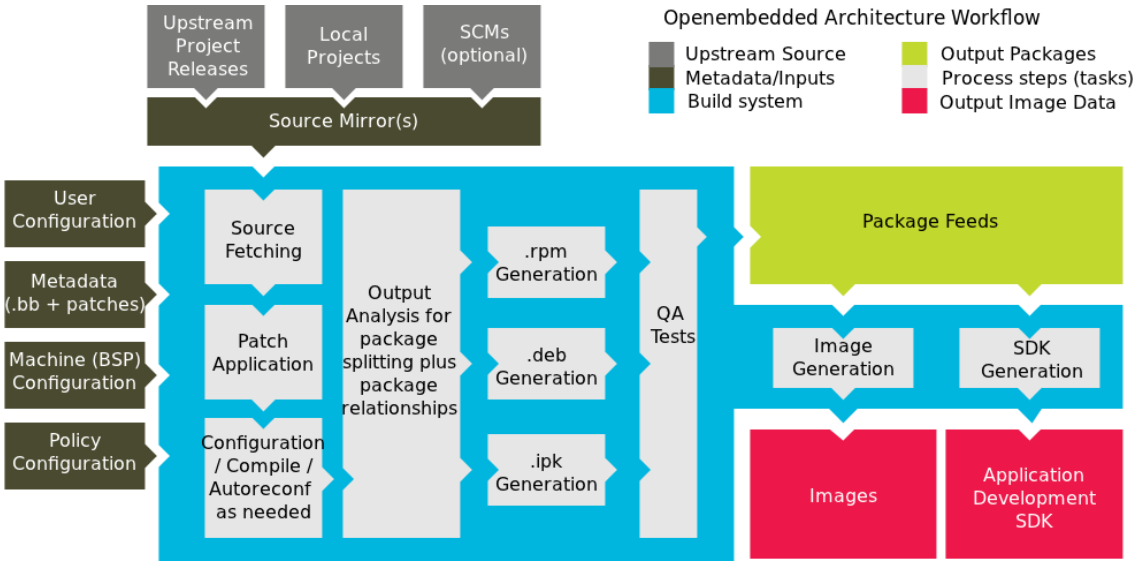
Metadata, toiselta nimeltään metatieto, on sekoitus Pythonia ja komentorivikomentosarjoja tekstitiedostoissa, jotka luovat valtavan joustavan järjestelmän. Poky käyttää tätä laajentaakseen OpenEmbedded-Corea sisällyttäen siihen kaksi eri kerrosta, jotka ovat meta-yocto ja meta-yocto-bsb. Meta-yocto-kerros tuottaa oletus- ja tuetut jakelut, visuaalisen laadun ja metadatan informaation seurannan. Meta-yocto-bsb-kerros tuottaa laitteiston ylläpidon Pokyn käyttöön. [30, s.10.]

4.5 Etuja

Usein sulautettujen järjestelmien ohjelmoinnissa ongelmana on, että ohjelmointi on hankalaa johtuen eri laitearkkitehtuureista. Linux Foundationin tavoitteena oli kehittää avoimen lähdekoodin alusta, jossa sulautetun laitteen prosessoriarkkitehtuuri ei vaikuta kehitystyöhön. Yocto-projekti tarjoaa tuen seuraaville prosessoriarkkitehtuureille: ARM, MIPS, PowerPC ja X86, sisältäen myös X86-64. Näin ollen Yocto-projekti tarjoaa ennen kaikkea joustavan alustan, jonka pohjalle kehittäjien on helppo aloittaa kehitystyö. Joustavan alustan ja eri prosessoriarkkitehtuurien tuen ansiosta se myös vähentää kaksinkertaista työtä, ja sen ylläpitäminen on vaivattomampaa.

Yocto-projektin komponentteja käytetään suunnitteluun, kehittämiseen, testaamiseen, luomiseen, simulointiin ja kokonaisen ohjelmiston testaamiseen

käyttäen Linux, Windows, GNOME-mobiilipohjaisia applikaatiokehyksiä ja Qt-kehyksiä. Kuvassa 14 on esitelty Yocto-projektin kehitysympäristöä.



Kuva 14. Yocto-projektin kehitysympäristö [29].

Yocto-projekti tarjoaa viimeisimmän Linux-ytimen, joukon sopivia järjestelmäkomentoja ja kirjastoja sulautettua ympäristöä varten. Yocto-projektin avulla saadaan käytettäväksi myös järjestelmän osat, kuten x11, GTX +, Qt, Clutter ja sdl. Näiden avulla voidaan luoda rikas käyttäjäkokemus myös laitteissa, jotka sisältävät näytön. Laitteissa, jotka eivät käytä näyttöä tai sisältävät muunlaisen käyttöliittymän, kyseisiä komponentteja ei tarvitse asentaa. Yocto-projektin avulla luodaan keskittynyt ja vakaa ydin yhteensopivaksi sulautetulle projektille, jolla voidaan helposti ja luotettavasti luoda ja kehittää ohjelmistoja. Yocto-projektissa on laaja tuki erilaisille laitteille ja laitteiden emulointi hoidetaan Qemu-emulaattorin avulla.

Yocto-projektilla voidaan generoida levykuvia eli imageja monenlaisille laitteille. Koska Yocto-projektilla kehitetty image voidaan käynnistää myös Qemu-emulaattorin sisällä, kehitysympäristö toimii hyvin testialustana kehitettäessä sulautettuja ohjelmistoja. Tavallisesti koko järjestelmää Qemu-emulaattorilla emuloitavat prosessoriarkkitehtuurit ovat x86, x86-64, Arm, MIPS ja PPC-prosessoriarkkitehtuurit, kuten myös tietty laite, kuten Raspberry Pi. [31.]

5 TYÖN SUORITTAMINEN

5.1 Tavoite

Työn tavoitteena on tutustua Yocto-projektiin ja toteuttaa mittausmoduulit, joilla voidaan mitata SensorTag-laitteen antureita. Työ on tarkoitus toteuttaa käyttäen Raspberry Pi -tietokonetta sekä laitetta nimeltä cc2541 SensorTag. Mittausmoduuleilta mitattu data on tarkoitus tallentaa MySQL-tietokantaan. Mittausmoduulit toteutetaan jokaiselle anturille erikseen. Yocto-projektiin tutustumisen jälkeen tavoitteena on kääntää Yocto-alusta Raspberry Pi:lle, jolla voidaan testata tehtyjä mittausmoduuleja.

5.2 Suunnittelu

Työn suorittamiseen tarvitaan seuraavat laitteet: Raspberry Pi -tietokone ja SensorTag. Lisäksi Raspberry Pi:hin tarvitsee yhdistää Bluetooth-lähetin, jossa on tuki Bluetooth 4.0 -versiolle. Raspberry Pi:n käyttämiseen tarvitaan myös Micro SD -muistikortti, näppäimistö, näyttö, HDMI-kaapeli, RJ45-kaapeli ja Micro USB -kaapeli.

Mittausmoduulien on tarkoitus toimia niin, että Raspberry Pi toimii järjestelmän tietokoneena. Raspberry Pi:n tehtävät ovat tiedon kysyminen ja lukeminen SensorTagilta, tiedonkäsittely ja tiedon tallentaminen MySQL-tietokantaan. Tietokanta on Raspberry Pi:llä. SensorTag toimii mittaajana sekä tiedon lähettäjänä. Mittausmoduulit on tarkoitus saada toimimaan aluksi Raspberry Pi:llä ja Wheezy-käyttöjärjestelmällä ja mahdollisesti myöhemmin myös Yocto-alustan päällä.

Työssä on tarkoitus tutustua Yocto-projektiin käytännössä, joten tutustuminen on luonnollista aloittaa asentamalla Yocto-projektialusta Ubuntuun. Tämän jälkeen voidaan kääntää Yocto-alusta myös Raspberry Pi:lle. Raspberry Pi:lle laitetaan aluksi Rasbian Debian Wheezy -käyttöjärjestelmä. Viimeisessä vaiheessa käännetään BLE-tuen omaava Yocto-alusta Raspberry Pi:lle.

5.3 Kehitysympäristöjen asentaminen

5.3.1 Raspberry Pi:n käyttöönotto

Työssä käytettyyn Raspberry Pi:hin käyttöjärjestelmäksi valittiin Rasbian Debian Wheezy. Asentaminen tapahtuu lataamalla Rasbian Debian Wheezy osoitteesta *http://www.raspberrypi.org/downloads/*. Raspberry Pi:n käyttöjärjestelmä asennetaan Micro SD -muistikortille esimerkiksi WIn32DiskManager-ohjelmalla. Käyttöjärjestelmä on sen jälkeen valmis käytettäväksi. Työssä Rasberry Pi:lle tarvitsee asentaa tuki Pythonille ja Bluetooth Low Energylle.

Bluetooth 4.0 -lähettimen asentamiseksi Raspberry Pi:hin tarvitsee asentaa muutamia kirjastoja. Asentaminen tapahtuu seuraavasti: *sudo apt-get install libusb-dev libdbus-1-dev libglib2.0-dev libudev-dev libical-dev libreadline-dev*. Seuraavaksi tarvitsee ladata bluez seuraavalla tavalla: luodaan kansio *sudo mkdir bluez*, mennään kansioon *cd bluez* ja ladataan paketti komennolla *sudo wget www.kernel.org/pub/linux/bluetooth/bluez-5.11.tar.xz*. Seuraavaksi ladattu paketti puretaan ja bluez asennetaan Raspberry Pi:lle seuraavan komentoketjun mukaisesti: *sudo unxz bluez-5.11.tar.xz*, *sudo tar xvf bluez-5.11.tar*, *cd bluez-5.11*, *sudo ./configure --disable-systemd*, *sudo make* ja *sudo make install*. Tämän jälkeen käynnistämällä Raspberry Pi uudelleen ja kytkemällä Bluetooth 4.0 -lähetin USB-porttiin Raspberry Pi on valmis käytettäväksi. [32.]

5.3.2 Yocto-projektin asentaminen

Asennettaessa Yocto-projektia tarvitaan Linux-käyttöjärjestelmä ja siinä ainakin 50 gigatavua vapaata tilaa. Käyttöjärjestelmäksi käy esimerkiksi jokin seuraavista Linux-julkaisuista: Ubuntu, Fedora, openSUSE, CentOS tai debian. Yocto-projektiryhmä tarkastaa jatkuvasti uusia Linux-julkaisuja yhteensopivaksi

Yocto-projektin kanssa. Yleensä, jos käytetään toiseksi uusinta julkaisua, sen tulisi toimia ongelmitta. Ladattavat ja asennettavat paketit riippuvat käytössä olevasta käyttöjärjestelmästä sekä siitä, mikä on kohderauta.

Asennettaessa Yocto-projekti Ubuntu- tai Debian-käyttöjärjestelmään tarvittavat olennaiset sekä graafiset paketit saadaan ladattua syöttämällä *$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \ build-essential chrpath socat libsdl1.2-dev xterm* komentoriville.

Seuraavaksi syötetään seuraavat komennot komentoriville:

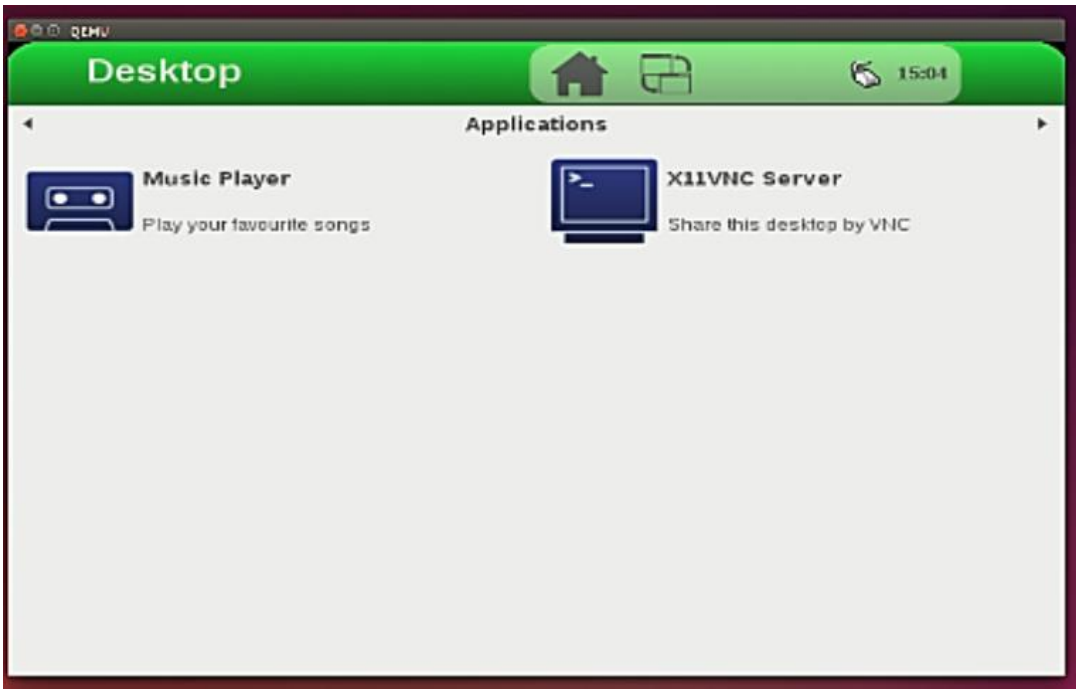
- $ git clone http://git.yoctoproject.org/git/poky
- $ cd poky
- $ git checkout -b dizzy origin/dizzy
- $ source oe-init-build-env

Ensimmäinen komento luo poky-nimisen varaston, joka on klooni Yocto-projektin poky varastosta. Toisella komennolla mennään poky-kansioon. Sitten syötetään kolmas komento, joka tarkistaa paikallisen haaran ja nimeää sen nimellä dizzy. Neljännellä komennolla suoritetaan ympäristön asennuskomentosarja. Komentosarja määrittelee asennettavalle ympäristölle asetukset ja luo kansion, tässä tapauksessa nimeltä build. Kansio sijaitsee lähdekansiossa, joka on tässä tapauksessa poky. Build-kansio sisältää kaikki komentosarjan ajamisen aikana luodut tiedostot. Komentosarjan ajamisen jälkeen ollaan komentorivillä kansiossa build.

Seuraavaksi tarkastellaan tiedoston *local.conf* sisältöä. Tiedosto löytyy kansiosta build. Oletusasetukset voivat toimia sellaisenaankin, mutta riippuen käytössä olevasta tietokoneesta ja kohdearkkitehtuurista voi olla tarpeen tehdä seuraavia muutoksia. Oletusarvona asennukselle on kohdassa *MACHINE qemux86*, joka tuottaa levykuvan, jota voidaan käyttää QEMU-emulaattorissa ja sen kohde on 32-bittinen Intel-arkkitehtuuri. Toinen säädettävä asia samassa tiedostossa ovat *BB_NUMBER_THREADS-* ja *PARALLEL_MAKE-muuttujat*. Oletuksena näiden arvo on tietokoneen käyttämien ytimien määrä. Kuitenkin jos käytössä on

useampi kuin yksi ydin, tulisi muuttujien arvon olla ytimien määrä kerrottuna kahdella +1.

Viimeinen komento ennen kuin QEMU-emulaattori voidaan käynnistää, on *bitbake -k core-image-sato*. Komennon suorittamisen kesto vaihtelee ensimmäisellä kerralla kolmesta tunnista vuorokauteen riippuen käytössä olevan tietokoneen ja nettiyhteyden nopeudesta. Emulaattori käynnistetään komennolla *runqemu qemux86*. Kuvassa 15 on esitelty QEMU-emulaattorin aloitusnäyttö. [31.]



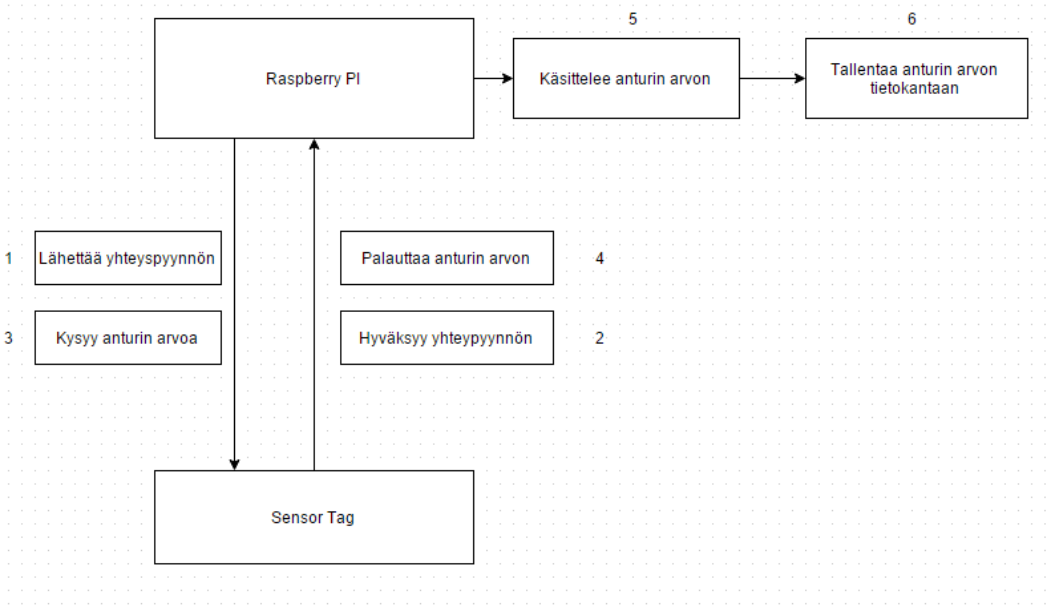
Kuva 15. QEMU-emulaattori, aloitusnäyttö.

Yleisesti QEMU-emulaattoria voidaan käyttää esimerkiksi tehdyn imagen emuloimiseen. Tässä työssä sitä voisi hyödyntää Rasbian Debian Wheezy -imagen käyttöön ilman Raspberry Pi:tä tai käännetyn Yocto-alustan testaamiseen. Kuitenkin koska käytössä on laite Raspberry Pi, niin testit voidaan hoitaa suoraan laitteella.

5.4 Mittausmoduulien teko

Työn tavoitteena oli tehdä jokaiselle anturille oma mittausohjelmansa ja tallentaa anturilta mitattu tieto MySQL-tietokantaan. Ohjelmointikieleksi valikoitui Python lähinnä sen tarjoamien pexpect- ja gattool-kirjastojen takia. Toinen vaihtoehto olisi ollut lähteä liikkeelle tutkimalla gattoolin-lähdekoodia ja toteuttaa mittausmoduulit C/C++-kielellä. Myös antureiden lähettämän datan käsittely ja tietokantaan tallennus tapahtuu Python-kielellä. Tietokanta sijaitsee Raspberry Pi:llä.

Yksinkertaistetusti jokainen mittausmoduuli toimii kuvassa 16 näkyvällä tavalla.



Kuva 16. Mittausmoduulien toiminta.

Raspberry Pi:n ja SensorTagin välinen tiedonsiirto tapahtuu Bluetooth 4.0 -yhteyden avulla. Yhteyden muodostamiseen käytetään Pythonin tarjoamaa pexpect-kirjastoa. Yhdistämiseen on tällä tavalla kaksi vaihtoehtoa: Joko koodata SensorTagin Bluetooth-osoite muuttujaan *bluetooth_adr = ”SensorTagin osoite”* tai syöttää SensorTagin Bluetooth-osoite komentoriviltä Python-tiedoston ajamisen aikana. Molemmat tavat on esitelty kuvassa 17.

```
# Toinen vaihtoehto on saada Sensor Tagin Bluetooth-osoite komentoriviltä
# bluetooth_adr = sys.argv[1] komennolla
# Esim. Lampotila.py B4:99:4C:64:BA:A2

# Kova koodattu versio
bluetooth_adr = "B4:99:4C:64:BA:A2"
tool = pexpect.spawn('gatttool -b ' + bluetooth_adr + ' --interactive')
tool.expect('\[LE\]>')

print "Valmiina yhteyteen. Paina sivunappia tarvittaessa"
tool.sendline('connect')
# testi onnistuneelle yhteydelle
tool.expect('\[CON\].*>')
```

Kuva 17. Python-koodi yhteyden muodostamiseen.

SensorTagin kanssa keskustellaan lähettämällä sille GATT-komentoja. Komennot löytyvät esimerkiksi lähteessä 31 olevasta SensorTag-attribuuttitaulukosta. Taulukossa 1 on kootusti muutamia antureille lähetettäviä komentoja ja SensorTagin palauttaman tiedon muoto.

Taulukko 1. SensorTagin anturien GATT-komentoja [31].

Anturi	Mittauksen aloitus	Luku	Datan muoto
Lämpötila	0x29 01	0x25	ObjectLSB:ObjectMSB:AmbientLSB:AmbientMSB.
Kiihtyvyys	0x31 01	0x2d	X : Y : Z
Gyroskooppi	0x5b 01	0x57	XLSB:XMSB:YLSB:YMSB:ZLSB:ZMSB
Magnetometri	0x44 01	0x40	XLSB:XMSB:YLSB:YMSB:ZLSB:ZMSBCoordinates
Ilmankosteus	0x3c 01	0x38	TempLSB:TempMSB:HumidityLSB:HumidityMSB
Ilmanpaine	0x55 01	0x51	TempLSB:TempMSB:PressureLSB:PressureMSB

SensorTagin anturit palauttavat luettaessa heksaluvun, joten antureilta saatavat arvot on ensin muutettava float-tyyppisiksi luvuiksi. Kuvassa 18 on esitelty funktio muuntamista varten. Tämän jälkeen algoritmien avulla voidaan laskea anturien palauttamat oikeat data-arvot. Algoritmit löytyvät lähteestä 23.

```
def floatfromhex(h):
    t = float.fromhex(h)
    if t > float.fromhex('7FFF'):
        t = -(float.fromhex('FFFF') - t)
        pass
    return t
```

Kuva 18. Heksaluvun muunnos liukuluvuksi.

Kappaleissa 5.4.1–5.4.6 nähdään mittausmoduulien toimintaa hieman tarkemmin. Kaikkien anturien lukeminen on teknisesti melko samanlaista, ainoastaan lähetettävät heksaluvut ja SensorTagin palauttama tiedon muoto muuttuvat. Lisäksi laskemiseen tarvittavat algoritmit ovat erilaisia. Lähteestä 23 löytyvät algoritmit on muutettu Python-koodiksi.

5.4.1 Lämpötila-anturi

Lämpötila-anturin lukeminen tehdään lähettämällä Raspberry Pi:ltä heksaluku *0x29 01* SensorTagille. SensorTag vastaa, että on valmis yhteyteen. Aloitetaan while-silmukka, jossa on aluksi sekunnin viive, joka toimii myös anturin mittaustuloksen päivitysvälinä. Lähetetään heksaluku *0x25* SensorTagille, millä pyydetään SensorTagin lämpötila-anturin arvo. GATT-taulukosta nähdään, että SensorTag palauttaa lämpötila-anturin arvon muodossa: *ObjectLSB:ObjectMSB:AmbientLSB:AmbientMSB*. SensorTag siis palauttaa 4 heksalukua Raspberry Pi:lle. Lämpötila-anturin arvon saamiseksi tarvitsee siis tallentaa *Lampo*-nimiseen muuttujaan kaksi ensimmäistä heksalukua järjestyksessä *ObjecktMSB* + *ObjectLSB* ja *ambT*-nimiseen muuttujaan kaksi seuraava heksalukua järjestyksessä *AmbientMSB* + *AmbientLSB*. Tallennetaan muuttujaan lämpöfunktion *floatfromhex(rval[2] + rval[1])* arvo ja *ambT* nimiseen muuttujaan *floatfromhex(rval[4] + rval[3])* arvo. Tämän jälkeen lasketaan lämpötila lähteestä 23 löytyvien algoritmien avulla. Koodi on esitelty kuvassa 19.

```
tool.sendline('char-write-cmd 0x29 01')
tool.expect('\[LE\]>')
while True:
    time.sleep(1)
    tool.sendline('char-read-hnd 0x25')
    tool.expect('descriptor: .*')
    rval = tool.after.split()
    Lampo = floatfromhex(rval[2] + rval[1])
    ambT = floatfromhex(rval[4] + rval[3])
    calcTmpTarget(Lampo, ambT)
```

Kuva 19. Lämpötila-anturin lukeminen.

5.4.2 Kiihtyvyysanturi

Kiihtyvyysanturin lukeminen tehdään lähettämällä Raspberry Pi:ltä heksaluku *0x31 01* SensorTagille. SensorTag vastaa, että on valmis yhteyteen. Aloitetaan while-silmukka, jossa on aluksi sekunnin viive. Viive toimii myös anturin mittaustuloksen päivitysvälinä. Lähetetään heksaluku *0x2d* SensorTagille, millä pyydetään kiihtyvyysanturin arvo. GATT-taulukosta nähdään, että SensorTagin vastaus tulee muodossa: *X:Y:Z*. Anturi siis palauttaa 3 heksalukua, jotka ovat x-, y- ja z-akselin arvot. Muunnetaan x-, y-, ja z-akselin arvot liukuluvuksi *floatfromhex*-funktiolla. Lasketaan akselikohtaiset kiihtyvyydet lähteestä 23 löytyvällä algoritmilla. Koodi on esitelty kuvassa 20.

```
tool.sendline('char-write-cmd 0x31 01')
tool.expect('\[LE\]>')

while True:
    time.sleep(1)
    tool.sendline('char-read-hnd 0x2d')
    tool.expect('descriptor: .*')
    rval = tool.after.split()
    x = floatfromhex(rval[1])
    y = floatfromhex(rval[2])
    z = floatfromhex(rval[3])
    xyz = calcAccel(x,y,z)
```

Kuva 20. Kiihtyvyysanturin lukeminen.

5.4.3 Ilmankosteusanturi

Ilmankosteusanturin lukeminen tapahtuu lähettämällä Raspberry Pi:ltä heksaluku *0x3c 01* SensorTagille. SensorTag vastaa, että on valmis yhteyteen. Aloitetaan while-silmukka, jossa on aluksi 5 sekunnin viive. Viive toimii myös mittaustuloksen päivitysvälinä. Ilmankosteusanturin lukemiseen viiveen tulee olla vähintään 3 sekuntia. Lähetetään heksaluku *0x38* SensorTagille, jolla pyydetään ilmankosteusanturin arvo. GATT-taulukosta nähdään, että SensorTagin vastaus tulee muodossa *TempLSB:TempMSB:HumidityLSB:HumidityMSB*. Anturi palauttaa 4 heksalukua. Tallennetaan *rawT*-nimiseen muuttujaan lämpötilan arvo seuraavasti *floatfromhex(rval[2] + rval[1]),* ja *rawH*-nimiseen muuttujaan ilmankosteuden arvo *floatfromhex(rval[4] + rval[3])*. Lasketaan lämpötila ja suhteellinen ilmankosteus lähteestä 23 löytyvien algoritmien avulla. Koodi on esitelty kuvassa 21.

```
tool.sendline('char-write-cmd 0x3c 01')
tool.expect('\[LE\]>')
while True:
    time.sleep(5)
    tool.sendline('char-read-hnd 0x38')
    tool.expect('descriptor: .*')
    rval = tool.after.split()
    rawT = floatfromhex(rval[2] + rval[1])
    rawH = floatfromhex(rval[4] + rval[3])
    calcHum(rawT, rawH)
```

Kuva 21. Ilmankosteusanturin lukeminen.

5.4.4 Gyroskooppianturi

Gyroskooppianturin lukeminen tapahtuu lähettämällä Raspberry Pi:ltä heksaluku *0x5b 07* SensorTagille. SensorTag vastaa, että on valmis yhteyteen. Aloitetaan while-silmukka, jossa on aluksi sekunnin viive. Viive toimii myös anturin mittaustuloksen päivitysvälinä. Lähetetään heksaluku *0x57* SensorTagille, millä pyydetään ilmankosteusanturin arvo. GATT-taulukosta nähdään, että

SensorTagin vastaus tulee muodossa: *XLSB:XMSB:YLSB:YMSB:ZLSB:ZMSB*. Anturi siis palauttaa 6 heksalukua. Muunnetaan x-, y-, ja z-akselien arvot liukuluvuksi *floatfromhex*-funktion avulla seuraavasti: *x=floatfromhex(rval[2] + rval[1]), y= floatfromhex(rval[4] + rval[3])* ja *z= floatfromhex(rval[6] + rval[5])*. Tämän jälkeen lasketaan akselikohtaiset kiihtyvyydet lähteestä 23 löytyvällä algoritmilla. Koodi on esitelty kuvassa 22.

```
tool.sendline('char-write-cmd 0x5b 07')
tool.expect('\[LE\]>')
while True:
    time.sleep(1)
    tool.sendline('char-read-hnd 0x57')
    tool.expect('descriptor: .*')
    rval = tool.after.split()
    x = floatfromhex(rval[2] + rval[1])
    y = floatfromhex(rval[4] + rval[3])
    z = floatfromhex(rval[4] + rval[3])

    xyz=calcGyro(x,y,z)
```

Kuva 22. Gyroskoopin lukeminen.

5.4.5 Magnetometrianturi

Magnetometrianturin lukeminen tapahtuu lähettämällä Raspberry Pi:ltä heksaluku *0x44 01* SensorTagille. SensorTag vastaa, että on valmis yhteyteen. Aloitetaan while-silmukka, jossa on aluksi sekunnin viive. Viive toimii myös anturin mittaustuloksien päivitysvälinä. Lähetetään heksaluku *0x40* SensorTagille, millä pyydetään magnetometrianturin arvo. GATT-taulukosta nähdään, että SensorTagin vastaus tulee muodossa: *XLSB:XMSB:YLSB:YMSB:ZLSB:ZMSBCoordinates*. Anturi siis palauttaa 6 heksalukua. Muunnetaan x-, y-, ja z-akselin arvot liukuluvuksi seuraavasti: *x=floatfromhex(rval[2] + rval[1])*, *y= floatfromhex(rval[4] + rval[3])* ja z= *floatfromhex(rval[6] + rval[5])*. Tämän jälkeen lasketaan magnetometrin

akselikohtaiset arvot lähteestä 23 löytyvällä algoritmilla. Koodi on esitelty kuvassa 23.

```
tool.sendline('char-write-cmd 0x44 01')
tool.expect('\[LE\]>')

while True:
    time.sleep(3)
    tool.sendline('char-read-hnd 0x40')
    tool.expect('descriptor: .*')
    rval = tool.after.split()
    x = floatfromhex(rval[2] + rval[1])
    y = floatfromhex(rval[4] + rval[3])
    z = floatfromhex(rval[6] + rval[5])
    xyz=calcMagn(x,y,z)
```

Kuva 23. Magnetometrin lukeminen.

5.5 Tietokannan suunnittelu ja asentaminen

Työssä tarvittava MySQL-tietokanta tulee olemaan Raspberry Pi:llä. Tietokantaan tarvitsee tallentaa 6 anturin mittaustulokset. Koska mittausmoduulit on suunniteltu erikseen, on luonnollista tehdä jokaiselle anturille oma taulunsa tietokantaan. Ennen tätä on kuitenkin asennettava MySQL-tietokanta ja tarvittavat Python-siteet Raspberry Pi:lle komennolla *sudo apt-get install mysql-server python-mysqldb*. Asennus kysyy root-käyttäjän salasanan asennuksen aikana. Asennuksen jälkeen kirjaudutaan sisään root-käyttäjänä ja luodaan tietokanta nimeltä tulokset. Otetaan tietokanta tulokset käyttöön ja luodaan testaaja käyttäjä. Annetaan käyttäjälle oikeudet käyttää tulokset-nimistä tietokantaa. Kuvassa 24 on esitelty tietokannan ja käyttäjän luominen. Työn kannalta ei ole oleellista miettiä tarkemmin, mitä oikeuksia kenenkin käyttäjän tulisi saada tai ovatko tietokantaratkaisut optimaalisia, oleellista on vain saada antureilta tuleva data tallennettua.

```
$ mysql -u root -p
Enter password:
mysql> CREATE DATABASE tulokset
mysql> USE tulokset

mysql> CREATE USER 'testaaja'@'localhost' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON tulokset.* TO 'testaaja'@'localhost'
mysql> FLUSH PRIVILEGES;
mysql> quit
```

Kuva 24. MySQL-tietokannan ja käyttäjän luominen.

Luodaan tietokantaan nimeltä tulokset taulut anturien lähettämää dataa varten. Jokaiselle anturille luodaan oma taulu. Tietokantaan yhteyden ottaminen ja tarvittavien taulujen luominen on esitelty kuvassa 25. Kuvassa kaikki taulut luodaan yhtä aikaa, mutta oikeasti luominen tapahtuu erikseen jokaisessa mittausmoduulissa.

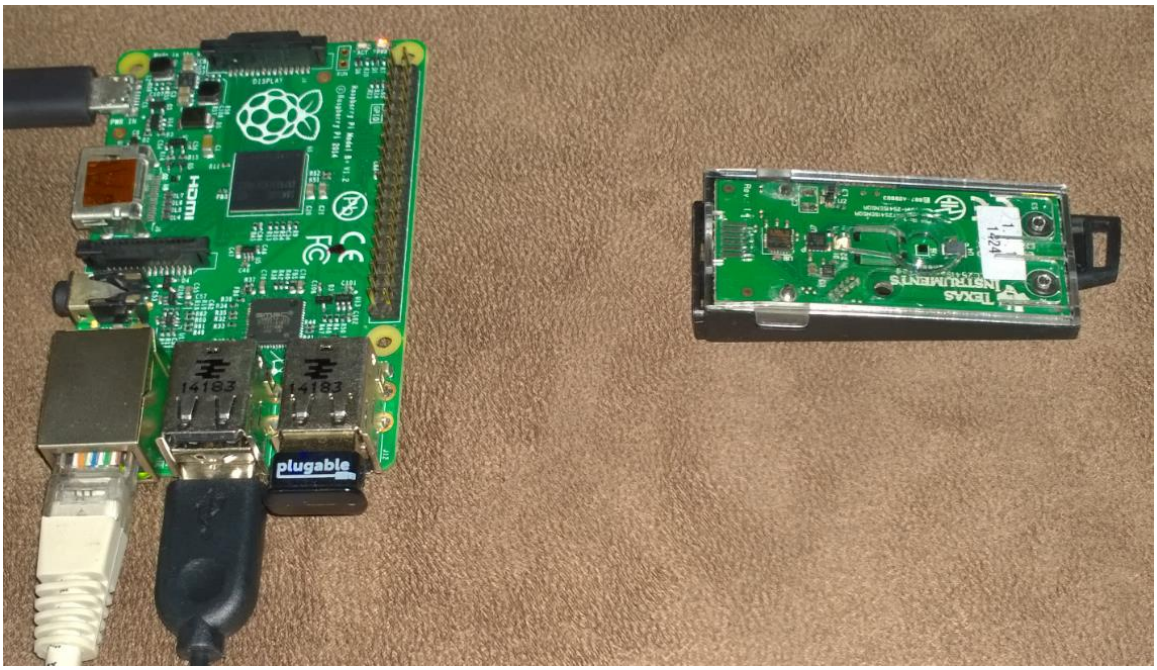
```
#Yhteys tietokantaan
con = MySQLdb.connect('localhost', 'testaaja', 'password', 'tulokset')
cur = con.cursor()
#Luodaan tarvittavat taulut
#Taulujen ensimmäinen termi on päivä määrä milloin mittaus on tehty.Loput ovat anturirien data-arvoja.
cur.execute("CREATE TABLE IF NOT EXISTS Magnetometri(Paivamaara DATE, x FLOAT, y FLOAT, z FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Lampotila(Paivamaara DATE,Lampo FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Ilmankosteus(Paivamaara DATE,Temperature FLOAT, RelativeHumidity FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Kiihtyvyys(Paivamaara DATE, x FLOAT, y FLOAT, z FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Gyroskooppi(Paivamaara DATE, x FLOAT, y FLOAT, z FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Barometri(Paivamaara DATE, paine FLOAT)")
```

Kuva 25. Taulujen luominen.

6 TYÖN TESTAAMINEN

6.1.1 Kytkennän kokoaminen

Kuvassa 26 on esitelty Raspberry Pi:n ja SensorTagin kytkentä. Raspberry Pi:hin on kytketty virtakaapeli, tiedonsiirtokaapeli RJ45, näppäimistö ja 4.0 Bluetooth -lähetin. Tiedonsiirto- ja virtakaapeli yhdistyvät Linuxilla varustettuun tietokoneeseen. Kytkentään voi kytkeä näytön Raspberry Pi:lle tai käyttää Raspberry Pi:tä terminaalin kautta SSH-yhteydellä. SensorTag on kasattu kohdan 3.3.1 mukaisesti.



Kuva 26. Raspberry Pi ja SensorTag.

6.1.2 Yhteyden muodostaminen

Käytettäessä Raspberry Pi:tä SSH-yhteyden kautta on ensimmäisenä otettava SSH-yhteys Ubuntusta komentorivin kautta Raspberry Pi:hin komennolla *ssh pi@10.42.0.2*. Kirjaudutaan sisään Raspberry Pi:hin. Etsitään SensorTagin

Bluetooth-osoite komennolla *sudo hcitool lescan*. Osoitetta tarvitaan myöhemmin, kun luetaan antureita. Etsimisrutiini on esitelty kuvassa 27.

```
pi@raspberrypi: ~
antti@antti-Aspire-V5-573:~$ ssh pi@10.42.0.2
pi@10.42.0.2's password:
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Oct 28 19:24:16 2015 from 10.42.0.1
pi@raspberrypi ~ $ sudo hcitool lescan
LE Scan ...
B4:99:4C:64:BA:A2 (unknown)
B4:99:4C:64:BA:A2 SensorTag
pi@raspberrypi ~ $
```

Kuva 27. SensorTagin etsiminen.

6.1.3 Anturien lukeminen Raspberry Pi:llä

Anturien lukeminen tapahtuu seuraavasti: Suoritetaan Python-tiedosto, jossa on koodi anturien lukemista varten. Lisäksi tarvitaan SensorTagin Bluetooth-osoite, joka on saatu kohdassa 5.5.2. SensorTagin Bluetooth-osoitteen voi koodata kiinteäksi suoraan mittaustiedostoon kuvassa 17 nähdyllä tavalla tai käyttäjä voi syöttää sen komentorivillä. Alapuolella olevissa esimerkeissä on käytetty tapaa, jossa SensorTagin Bluetooth-osoite on koodattu mittausmoduuliin. Kuvista 28–32 nähdään mittausmoduulien tuottama tulostus Raspberry Pi:llä. Mittaustuloksia kommentoidaan hieman tarkemmin kappaleessa 6.1.7.

Luetaan lämpötila-anturin arvo suorittamalla mittausohjelma komennolla *sudo ./Lampotila.py*.

```
pi@raspberrypi ~ $ sudo ./Lampotila.py
Valmiina yhteyteen. Paina sivunappia tarvittaessa
Lampotila on 24.37 C
2015-11-18 22:05:33
./Lampotila.py:57: Warning: Data truncated for column 'Paivamaara' at row 1
  """,(Aika,Lampo))
Tieto tallennettu tietokantaan.
Lampotila on 28.55 C
2015-11-18 22:05:34
Tieto tallennettu tietokantaan.
Lampotila on 23.23 C
2015-11-18 22:05:35
Tieto tallennettu tietokantaan.
^Z
[7]+  Stopped                 sudo ./Lampotila.py
pi@raspberrypi ~ $
```

Kuva 28. Lämpötila-anturin lukeminen.

Luetaan kiihtyvyysanturin arvo suorittamalla mittausohjelma komennolla *sudo ./Kiihtyvyys.py*.

```
pi@raspberrypi ~ $ ./Kiihtyvyys.py
Valmiina yhteyteen. Paina sivunappia tarvittaessa
2015-11-18 21:26:46
./Kiihtyvyys.py:43: Warning: Data truncated for column 'Paivamaara' at row 1
  """, (aika,accel(x),accel(y),accel(z)))
X-akseli 19.0 Y-akseli 21.0 Z-akseli 51.0
Tieto tallennettu tietokantaan.
2015-11-18 21:26:48
X-akseli 24.0 Y-akseli 25.0 Z-akseli 50.0
Tieto tallennettu tietokantaan.
2015-11-18 21:26:49
X-akseli 26.0 Y-akseli 26.0 Z-akseli 51.0
Tieto tallennettu tietokantaan.
^Z
[6]+  Stopped                 ./Kiihtyvyys.py
pi@raspberrypi ~ $
```

Kuva 29. Kiihtyvyysanturin lukeminen.

Luetaan gyroskooppi-anturin arvo suorittamalla mittausohjelma komennolla *sudo ./Gyroskooppi.py*.

```
pi@raspberrypi ~ $ sudo ./Gyroskooppi.py
Valmiina yhteyteen. Paina sivunappia tarvittaessa
2015-11-18 22:04:10
./Gyroskooppi.py:43: Warning: Data truncated for column 'Paivamaara' at row 1
  """, (aika,gyro(x),gyro(y),gyro(z)))
X-akseli 5502.0 Y-akseli -7673.0 Z-akseli -7673.0
Tieto tallennettu tietokantaan.
2015-11-18 22:04:11
X-akseli 2548.0 Y-akseli -2815.0 Z-akseli -2815.0
Tieto tallennettu tietokantaan.
2015-11-18 22:04:12
X-akseli 2682.0 Y-akseli 377.0 Z-akseli 377.0
Tieto tallennettu tietokantaan.
2015-11-18 22:04:13
X-akseli -197.0 Y-akseli 550.0 Z-akseli 550.0
Tieto tallennettu tietokantaan.
^Z
[4]+  Stopped                 sudo ./Gyroskooppi.py
pi@raspberrypi ~ $
```

Kuva 30. Gyroskooppianturin lukeminen.

Luetaan magnetometrinanturin arvo suorittamalla mittausohjelma komennolla *sudo ./Magnetometri.py*.

```
pi@raspberrypi ~ $ sudo ./Magnetometri.py
Valmiina yhteyteen. Paina sivunappia tarvittaessa
2015-11-18 22:04:49
./Magnetometri.py:43: Warning: Data truncated for column 'Paivamaara' at row 1
  """, (aika,magforce(rawX),magforce(rawY),magforce(rawZ)))
X-akselin magneettivuon tiheys 0.0 µT  Y-akselin magneettivuontiheys 0.0 µT Z-akselin magneettivuon tiheys 0.0 µT
Tieto tallennettu tietokantaan
2015-11-18 22:04:50
X-akselin magneettivuon tiheys -1636.0 µT  Y-akselin magneettivuontiheys 1539.0 µT Z-akselin magneettivuon tiheys 1216.0 µT
Tieto tallennettu tietokantaan
2015-11-18 22:04:51
X-akselin magneettivuon tiheys -1636.0 µT  Y-akselin magneettivuontiheys 1539.0 µT Z-akselin magneettivuon tiheys 1216.0 µT
Tieto tallennettu tietokantaan
^Z
[5]+  Stopped                 sudo ./Magnetometri.py
pi@raspberrypi ~ $
```

Kuva 31. Magnetometrianturin lukeminen.

Luetaan ilmankosteusanturin arvo suorittamalla mittausohjelma komennolla *sudo ./Ilmankosteus.py*.

```
pi@raspberrypi ~ $ ./Ilmankosteus.py
Valmiina yhteyteen. Paina sivunappia tarvittaessa
2015-11-06 21:25:22
./Ilmankosteus.py:45: Warning: Data truncated for column 'Paivamaara' at row 1
  """, (aika,t,rh))
lämpötila on 27.5 C Ilmankosteus on 61.8 %
Tieto tallennettu tietokantaan.
2015-11-06 21:25:25
lämpötila on 27.4 C Ilmankosteus on 62.4 %
Tieto tallennettu tietokantaan.
2015-11-06 21:25:28
lämpötila on 27.3 C Ilmankosteus on 62.8 %
Tieto tallennettu tietokantaan.
2015-11-06 21:25:31
lämpötila on 27.2 C Ilmankosteus on 63.3 %
Tieto tallennettu tietokantaan.
2015-11-06 21:25:35
lämpötila on 27.2 C Ilmankosteus on 63.7 %
Tieto tallennettu tietokantaan.
2015-11-06 21:25:38
lämpötila on 27.1 C Ilmankosteus on 64.0 %
Tieto tallennettu tietokantaan.
^Z
[3]+  Stopped                 ./Ilmankosteus.py
```

Kuva 32. Ilmankosteusanturin lukeminen.

6.1.4 Yocto-projekti Raspberry Pi:lle

Yocto-projektista voidaan kääntää levykuva Raspberry Pi:lle. Imagen kääntäminen tapahtuu seuraavasti: Tehdään haluttuun hakemistoon kansio nimeltä yocto komennolla *mkdir yocto*, mennään kansioon komennolla *cd yocto*, kloonataan poky komennolla *git clone -b dylan git://git.yoctoproject.org/poky.git*, mennään kansioon poky komennolla *cd poky* ja kloonataan Raspberry Pi:n metadata komennolla *git clone -b dylan git://git.yoctoproject.org/meta-raspberrypi*.

Tämän jälkeen alustetaan joitakin muuttujia ja luodaan build-niminen hakemisto komennolla *oe-init-build-env build*. Muokataan tiedoston *conf/local.conf* sisältöä esimerkiksi tekstieditori Nanolla. Optimoidaan käännösnopeutta käytettävissä olevan tietokoneen mukaiseksi. Asetetaan kohtaan THREADS ytimien ja säikeiden lukumäärä +1 ja laitteeksi *"raspberrypi"*. Mahdollisesti lisätään

tietokoneen muistin määrä kohtaan *GPU_Mem*. Jos kohtaa ei ole sen, voi tehdä itse. Kuvassa 32 näkyy esimerkkitiedoston *local.conf* sisältö.

```
local.conf ×
# option determines how many tasks bitbake should run in parallel:
#
#BB_NUMBER_THREADS ?= "4"
#
# The second option controls how many processes make should run in parallel when
# running compile tasks:
#
#PARALLEL_MAKE ?= "-j 4"
#
# For a quad-core machine, BB_NUMBER_THREADS = "4", PARALLEL_MAKE = "-j 4" would
# be appropriate for example.

#
# Machine Selection
#
# You need to select a specific machine to target the build with. There are a selection
# of emulated machines available which can boot and run in the QEMU emulator:
#
#MACHINE ?= "qemuarm"
#MACHINE ?= "qemumips"
#MACHINE ?= "qemuppc"
#MACHINE ?= "qemux86"
#MACHINE ?= "qemux86-64"
#
# There are also the following hardware board target machines included for
# demonstration purposes:
#
#MACHINE ?= "beagleboard"
#MACHINE ?= "genericx86"
#MACHINE ?= "genericx86-64"
#MACHINE ?= "mpc8315e-rdb"
#MACHINE ?= "routerstationpro"
#
# This sets the default machine to be qemux86 if no other machine is selected:
MACHINE ??= "raspberrypi"
```

Kuva 33. local.conf-tiedoston näkymä.

Seuraavaksi tulee asettaa *meta-raspberrypi:n* polku tiedostoon *conf/bblayers*. Kuvassa 34 on esimerkki siitä, miltä kansion sisällön tulisi näyttää.

```
bblayers.conf (~/yocto/poky/build/conf) - gedit
Open  Save  Undo
bblayers.conf
# LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
# changes incompatibly
LCONF_VERSION = "6"

BBPATH = "${TOPDIR}"
BBFILES ?= ""

BBLAYERS ?= " \
  /home/antti/yocto/poky/meta \
  /home/antti/yocto/poky/meta-yocto \
  /home/antti/yocto/poky/meta-yocto-bsp \
  /home/antti/yocto/poky/meta-raspberrypi \
  "
BBLAYERS_NON_REMOVABLE ?= " \
  /home/antti/yocto/poky/meta \
  /home/antti/yocto/poky/meta-yocto \
  "
Plain Text  Tab Width: 8  Ln 6, Col 14  INS
```

Kuva 34. bblayers.conf tiedoston sisältö.

Nyt ollaan tilanteessa, jossa image voidaan kääntää. Image voidaan kääntää joko *rpi-basic-image*- tai *rpi-hwup-image*-komennolla. Käännetään image komennolla *bitbake rpi-basic-image*, jolloin saadaan imagelle myös SSH-tuki. Käännökseen kuluva aika riippuu käytettävissä olevasta tietokoneesta ja nettiyhteydestä. Käännöksen asetukset ovat näkyvissä kuvassa 35.

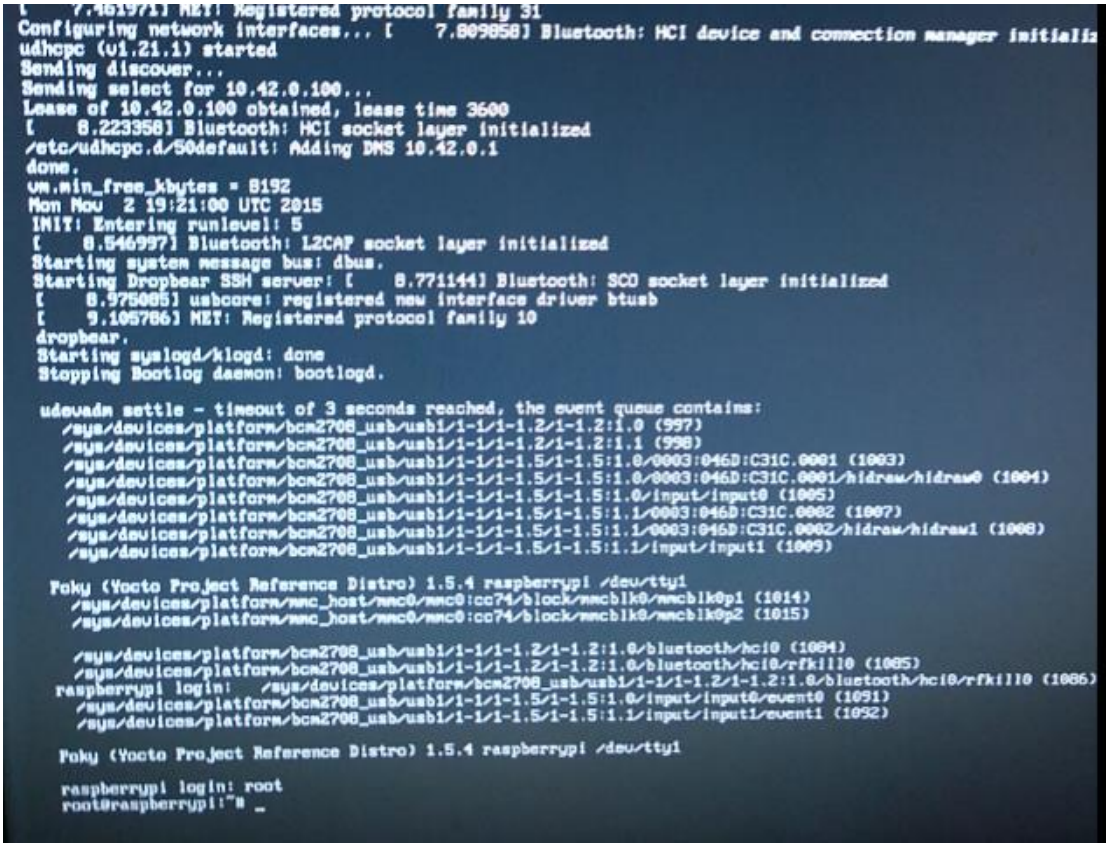
```
antti@antti-Aspire-V5-573: ~/yocto/poky/build
on of the build system; you may possibly experience unexpected failures. It is r
ecommended that you use a tested distribution.
Loading cache: 100% |##################################################| ETA:  00:00:00
Loaded 1201 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION        = "1.20.0"
BUILD_SYS         = "x86_64-linux"
NATIVELSBSTRING   = "Ubuntu-14.04"
TARGET_SYS        = "arm-poky-linux-gnueabi"
MACHINE           = "raspberrypi"
DISTRO            = "poky"
DISTRO_VERSION    = "1.5.4"
TUNE_FEATURES     = "armv6 vfp"
TARGET_FPU        = "vfp"
meta
meta-yocto
meta-yocto-bsp    = "dora:8e9e7e8def9b0b728937d34115f7165677c31bed"
meta-raspberrypi  = "dora:1c696a95f16337e4168a045479a386194d32e18f"

NOTE: Preparing runqueue
NOTE: Executing SetScene Tasks
NOTE: Executing RunQueue Tasks
WARNING: Failed to fetch URL ftp://ftp.debian.org/debian/pool/main/n/netbase/net
base_5.1.tar.gz, attempting MIRRORS if available
WARNING: Failed to fetch URL ftp://ftp.debian.org/debian/pool/main/d/dpkg/dpkg_1.17.1.tar.xz,
```

Kuva 35. Yocto-projektin käännösasetukset Raspberry Pi:lle.

Käännetty image löytyy kansiosta *tmp/deploy/images/*. Image voidaan laittaa muistikortille esim. Win32DiskManager-ohjelman avulla tai komentorivillä *sudo dd if=tmp/deploy/images/rpi-basic-image-raspberrypi.rpi-sdimg of=/dev/sdb bs=1M*, jossa sdb on muistikortin osoite, osoitteen voi tarkistaa komentorivillä komennolla lsblk. Muistikortti ei osaa hyödyntää koko kapasiteettiaan imagen laittamisen jälkeen, mutta sen koko kapasiteetti saadaan käyttöön kasvattamalla sen kokoa. Kasvattamisen voi tehdä joko komentorivillä tai yksinkertaisemmin Gparted-ohjelmalla. Kuvassa 36 on käännetty Yocto-projekti käynnissä Raspberry Pi:llä. Sisään kirjaudutaan käyttäjänimellä root ilman salasanaa.

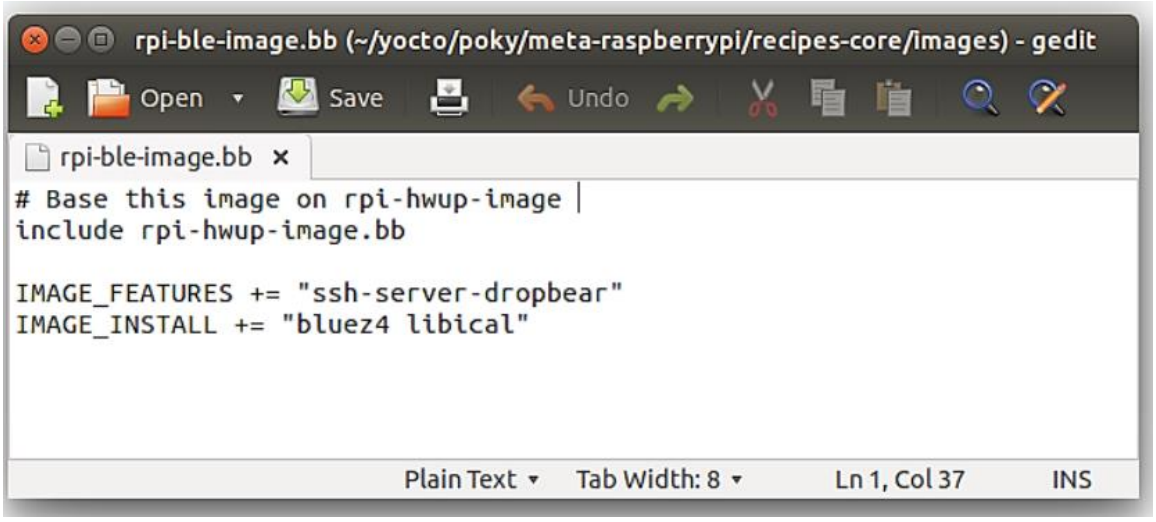


Kuva 36. Yocto-alusta Raspberry Pi:llä.

6.1.5 SensorTagin tarvitsemat kirjastot Yocto-projektiin

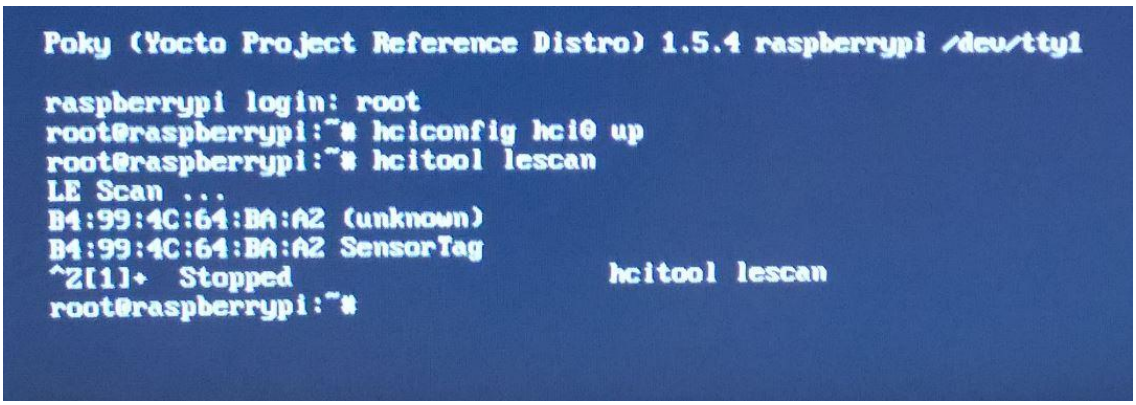
Käännetään uusi image Raspberry Pi:lle, että saadaan käyttöön avoimen lähdekoodin Bluetooth-kirjasto, josta löytyy tuki BLE-laitteille. Luodaan uusi

resepti kansioon *meta-raspberrypi/recipes-core/images/rpi-ble-image.bb*. Reseptin sisältö on kuvassa 37.



Kuva 37. BLE-resepti.

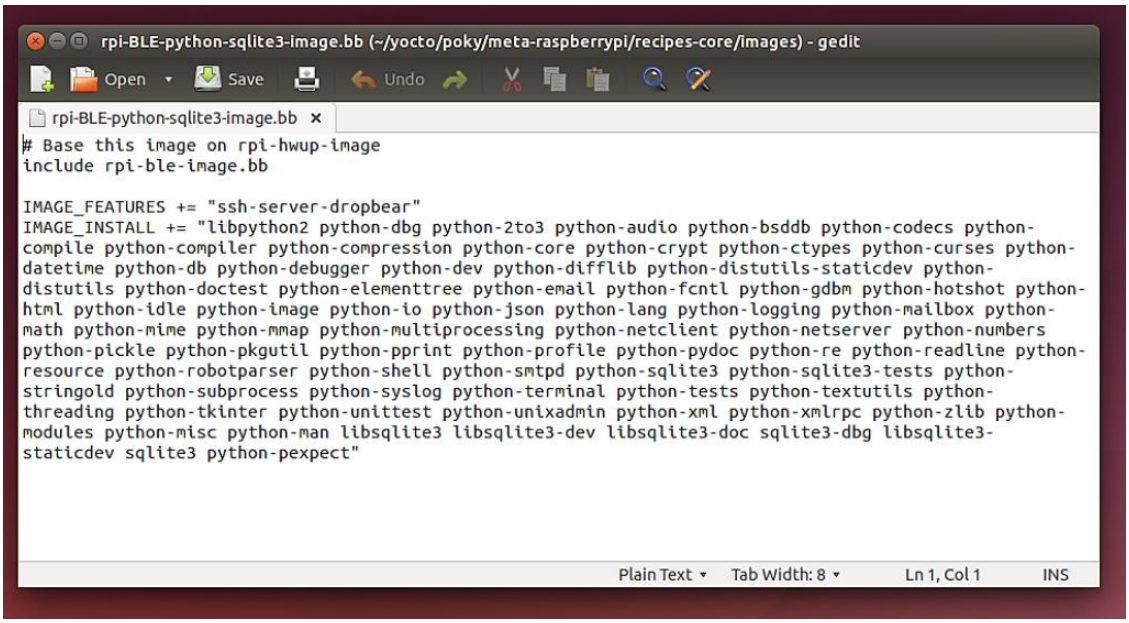
Käännetään uusi image poky-kansiosta seuraavasti: *oe-init-build-env build* ja sitten *bitbake rpi-ble-image*. Image tulee kansioon */tmp/deploy/images/*. Imagen asentaminen muistikortille suoritetaan samalla tavalla kuin kohdassa 5.3.3. Kuvassa 38 on käynnissä BLE-tuen omaava Yocto-projekti Raspberry Pi:llä. Sisään kirjaudutaan root-käyttäjänä ilman salasanaa. Käynnistetään Bluetooth-lähetin komennolla *hciconfig hci0 up*, etsitään BLE-laitteita *hcitool lescan* ja löydetään SensorTag.



Kuva 38. Yocto-projekti BLE-tuella.

6.1.6 Mittausmoduulien testaaminen Yocto-projektissa

Tehtyjen mittausmoduulien testaamista varten Yocto-projektiin tarvitaan vielä tuki Pythonille ja tietokannalle sen takia, että voidaan ajaa Python-tiedostoja ja tallentaa mitattu tieto tietokantaan. Mittausmoduulien tietokantaratkaisu muutetaan tässä vaiheessa MySQL:stä sqlite3:een. Lähinnä sen takia, että tarvittavat lähdepaketit löytyvät helpommin sqlite3:lle. Python ja sqlite3 tukea varten tulee kääntää vielä yksi Yocto-alusta. Lisätään tarvittavat lähdepaketit kuvassa 39 olevaan reseptiin nimeltä *rpi-BLE-python-sqlite3-image.bb*. Osa tiedostoon tehdyistä Python lisäyksistä ei ole oleellisia työn kannalta.



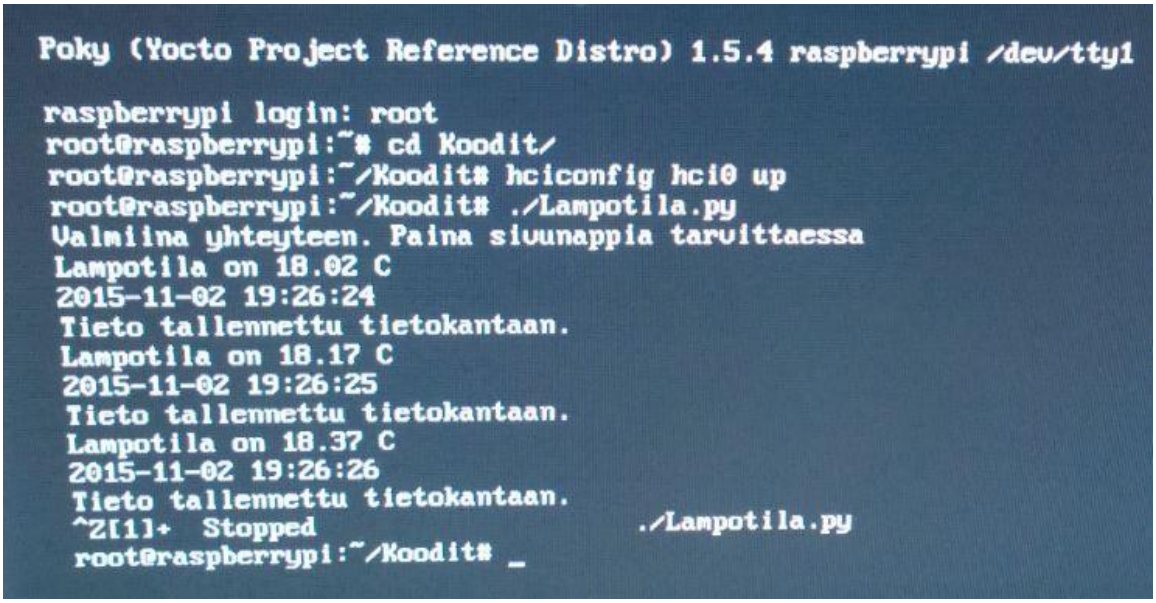
Kuva 39. *Rpi-BLE-python-sqlite3-image.bb*-tiedosto.

Tietokantakoodit tulee muuttaa sqlite3-muotoon kuvassa 40 näkyvällä tavalla. Sqlite3:a ei tarvitse enää erikseen asentaa, vaan Bitbake on asentanut sen. Sqlite3 ei edellytä käyttäjän luomista tietokannan käyttöön, joten se on nopea ja helppo ratkaisu testaamiseen. Kaikkien mittausmoduulien taulujen luominen on näkyvissä kuvassa 40.

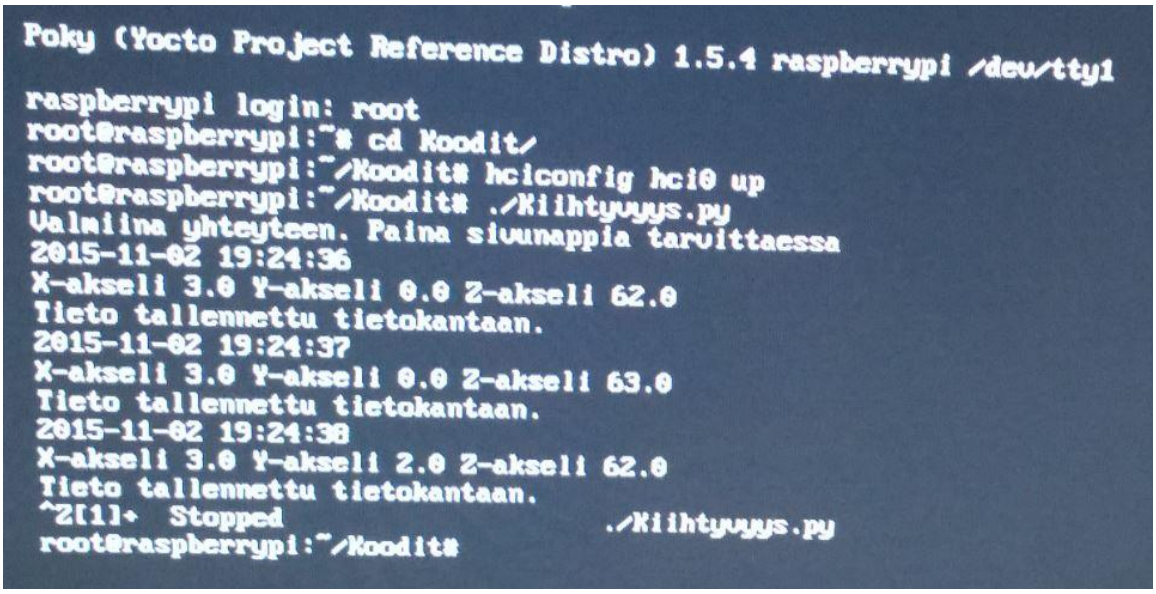
```
import sqlite3
#Yhteys tietokantaan, luo jos ei ole olemassa
con = sqlite3.connect('tulokset.db')
cur = con.cursor()
#Luodaan tarvittavat taulut
#Taulujen ensimmäinen termi on päivä määrä milloin mittaus on tehty.Loput ovat anturirien data-arvoja.
cur.execute("CREATE TABLE IF NOT EXISTS Magnetometri(Paivamaara DATE, x FLOAT, y FLOAT, z FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Lampotila(Paivamaara DATE,Lampo FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Ilmankosteus(Paivamaara DATE,Temperature FLOAT, RelativeHumidity FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Kiihtyvyys(Paivamaara DATE, x FLOAT, y FLOAT, z FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Gyroskooppi(Paivamaara DATE, x FLOAT, y FLOAT, z FLOAT)")
cur.execute("CREATE TABLE IF NOT EXISTS Barometri(Paivamaara DATE, paine FLOAT)")
```

Kuva 40. Sqlite3-tietokannan ja taulujen luominen.

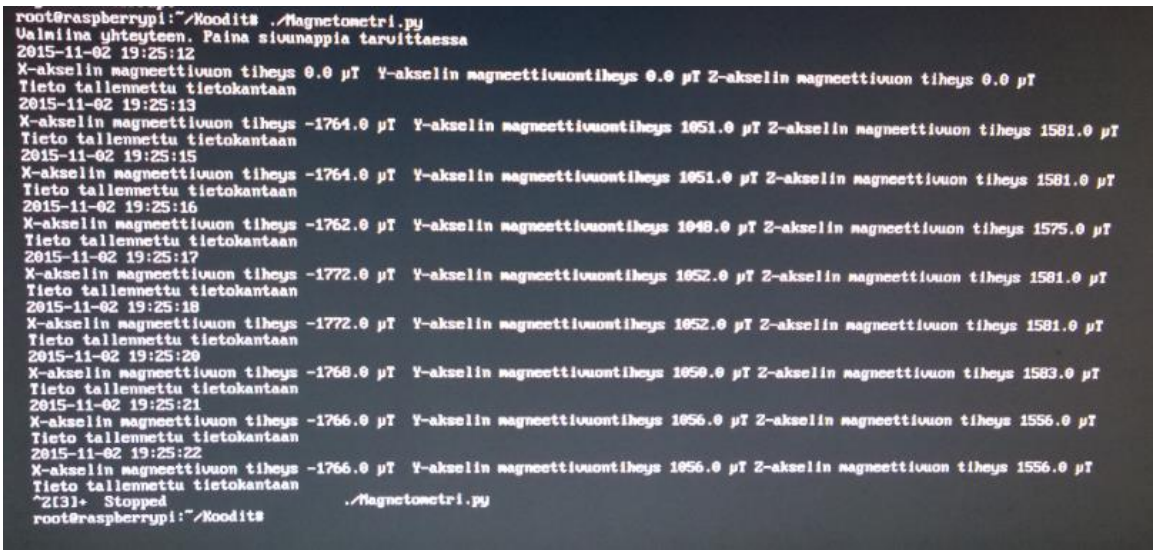
Nyt kaikki on valmista mittausmoduulien testaamiseen Yocto-alustalla. Siirretään mittausmoduulikoodit luodulle Yocto-alustalle ja testataan niiden toiminta. Tiedostojen käyttö toimii samalla lailla kuin kohdassa 6.1.3. Kuvissa 41–45 nähdään kuvat, joissa mittausmoduulit ovat mittaamassa Yocto-alustalla.



Kuva 41. Lämpötila-anturin lukeminen Yocto-alustalla.



Kuva 42. Kiihtyvyysanturin lukeminen Yocto-alustalla.



Kuva 43. Magnetometrianturin lukeminen Yocto-alustalla.

```
Poky (Yocto Project Reference Distro) 1.5.4 raspberrypi /dev/tty1

raspberrypi login: root
root@raspberrypi:~# cd Koodit/
root@raspberrypi:~/Koodit# hciconfig hci0 up
root@raspberrypi:~/Koodit# ./Gyroskooppi.py
Valmiina yhteyteen. Paina sivunappia tarvittaessa
2015-11-02 19:23:30
X-akseli 160.0 Y-akseli 572.0 Z-akseli 572.0
Tieto tallennettu tietokantaan.
2015-11-02 19:23:31
X-akseli 123.0 Y-akseli 538.0 Z-akseli 538.0
Tieto tallennettu tietokantaan.
2015-11-02 19:23:32
X-akseli 275.0 Y-akseli 1460.0 Z-akseli 1460.0
Tieto tallennettu tietokantaan.
2015-11-02 19:23:34
X-akseli 1100.0 Y-akseli 2783.0 Z-akseli 2783.0
Tieto tallennettu tietokantaan.
^Z[1]+  Stopped                 ./Gyroskooppi.py
root@raspberrypi:~/Koodit#
```

Kuva 44. Gyroskooppianturin lukeminen Yocto-alustalla.

```
Poky (Yocto Project Reference Distro) 1.5.4 raspberrypi /dev/tty1

raspberrypi login: root
root@raspberrypi:~# cd Koodit/
root@raspberrypi:~/Koodit# hciconfig hci0 up
root@raspberrypi:~/Koodit# ./Ilmankosteus.py
Valmiina yhteyteen. Paina sivunappia tarvittaessa
2015-11-02 19:24:49
lämpötila on 24.5 C Ilmankosteus on 54.3 %
Tieto tallennettu tietokantaan.
2015-11-02 19:25:00
lämpötila on 24.5 C Ilmankosteus on 53.9 %
Tieto tallennettu tietokantaan.
2015-11-02 19:25:10
lämpötila on 24.5 C Ilmankosteus on 53.8 %
Tieto tallennettu tietokantaan.
2015-11-02 19:25:21
lämpötila on 24.5 C Ilmankosteus on 54.1 %
Tieto tallennettu tietokantaan.
^Z[1]+  Stopped                 ./Ilmankosteus.py
root@raspberrypi:~/Koodit# _
```

Kuva 45. Kosteusanturin lukeminen Yocto-alustalla.

6.1.7 Analyysi työstä

Insinöörityössä on saatu aikaan mittausmoduulit SensorTagin antureille ja käännettyä toimiva Yocto-alusta Raspberry Pi:lle. Yocto-alustalle saatiin tuki Bluetooth Low Energylle, tarvittaville kirjastoille, Python-tiedostojen käytölle ja tietokannalle. Luodulla alustalla onnistuu tehtyjen mittausmoduulien testaaminen ja tiedon tallentaminen tietokantaan.

SensorTagin anturien lukeminen onnistuu tehdyllä Yocto-alustalla. Anturit palauttavat arvonsa Raspberry Pi:lle ja ne saadaan tallennettua tietokantaan. Arvojen luotettavuudesta ei sinällään saada työn aikarajoissa sen tarkempaa tietoa, mutta äkkiseltään testattuna ainakin lämpötila-anturi toimii oikein. Tästä päätellen luultavasti muutkin anturien palauttamat arvot ovat oikeita. Anturien palauttamien arvojen tarkempi analyysi edellyttäisi jonkinlaista verrokkimittausjärjestelmää varustettuna samoja arvoja mittaavilla antureilla.

Yocto-alustan olisi voinut myös kääntää kerralla lopulliseen muotoonsa, mutta kokemuksen puutteen johdosta työssä se tehdään vaiheittain. Tämä tapa myös helpottaa mahdollisten virheiden etsintää ja korjaamista. Työssä tehdyt Yocto-alustat toimivat suunnitellusti ja täyttävät niille asetetut vaatimukset.

7 YHTEENVETO

Tämän insinöörityön tarkoituksena oli toteuttaa mittausmoduulit SensorTag-laitteen antureille ja tutustua Yocto-projektiin. Työn tavoitteet saavutettiin hyvin. Lopputuloksena saatiin toteutettua mittausmoduulit SensorTagin antureille ja mittausmoduuleita päästiin testaamaan käännetyllä Yocto-alustalla. Yocto-projektin perusperiaatteet tulivat hyvin tutuksi työn aikana.

Henkilökohtaisina tavoitteina insinöörityössä minulla oli kehittyä sulautettujen laitteiden ohjelmoinnissa ja tutustua Yocto-projektin mahdollisuuksiin. Tavoitteena oli myös kehittyä tehdyn työn raportoinnissa. Henkilökohtaisiin tavoitteisiin päästiin ja ne osittain ylitettiin.

Työ oli monipuolinen ja sopivan haastava. Työn valmistumista hidastivat hieman sopivien laitteiden puute, hieman puutteellinen ohjeistus ja kokemattomuus toteuttaa käännetty Linux-pohjainen alusta. Nämä osaltaan johtivat työn aikataulun venymiseen. Insinöörityöstä jäi positiivinen mieli ja olen tyytyväinen lopputulokseen.

LÄHTEET

1 Granlund Kaj. Tietoliikenne. Jyväskylä: Docendo, 2007. s. 474. ISBN: 978-951-0- 32821-7

2 SIG Membership. Luettu 28.3.2014. [WWW-dokumentti]

< http://www.bluetooth.com/Pages/sig-membership.aspx >

3 Bluetooth. Luettu 29.3.2015. [WWW-dokumentti]

<http://en.wikipedia.org/wiki/Bluetooth>

4 Osimalli. Luettu 30.9.2015 [WWW-dokumentti]

<http://fi.wikipedia.org/wiki/OSI-malli>

5 Bluetooth radio interface, modulation, & channels. Luettu 30.9.2015. [WWW-dokumentti] <http://www.radio-electronics.com/info/wireless/bluetooth/radio-interface-modulation.php>

6 Bluetooth Low Energy Wibree. Luettu 13.10.2015. [WWW-dokumentti]< http://www.radio-electronics.com/info/wireless/bluetooth/bluetooth-low-energy-wibree.php>

7 Bluetooth Smart Technology. Luettu 13.10.1015. [WWW-dokumentti] <https://www.bluetooth.org/en-us/marketing/bluetooth-smart-technology>

8 Bluetooth Low Energy. Luettu 30.11.2015. [WWW-dokumentti] <http://www.litepoint.com/wp-content/uploads/2014/02/Bluetooth-Low-Energy_WhitePaper.pdf>

9 Bluetooth Goes Ultra-low-power. Luettu 13.10.2015. [WWW-dokumentti] <http://www.digikey.com/en/articles/techzone/2011/dec/bluetooth-goes-ultra-low-power>

10 Bluetooth Low energy Drives Design Innovations. Luettu 30.9.2015. [WWW-dokumentti] <http://www.radio-electronics.com/articles/wireless-technology/bluetooth-low-energy-drives-design-innovations-111>

11 Measuring Bluetooth® Low Energy Power Consumption. Luettu 30.11.2015. [WWW-dokumentti] <http://www.ti.com/lit/an/swra347a/swra347a.pdf?keyMatch=ble%20power%20consumption&tisearch=Search-EN-Everything>

12 CC2540 and CC2541 Bluetooth low energy Software Developer's. Luettu 13.10.2015.[WWW-dokumentti] <http://www.ti.com/lit/ug/swru271g/swru271g.pdf>

13 Your smartphone is the hub of your life. Luettu 28.3.2015. [WWW-dokumentti]< http://www.bluetooth.com/Pages/Phones.aspx>

14 Be more productive with Bluetooth technology. Luettu 28.3.2015. [WWW-dokumentti]< http://www.bluetooth.com/Pages/personal-computers.aspx>

15 Automotive market rapidly expanding beyond hands-free calling.Luettu 28.3.2015. [WWW-dokumentti] <http://www.bluetooth.com/Pages/Automotive-Market.aspx>

16 Bluetooth Smart Beacons in Retail. Luettu 28.3.2015. [WWW-dokumentti] <http://www.bluetooth.com/Pages/beacons-retail-location.aspx>

17 Transforming Consumer Electronics. Luettu 28.3.2015. [WWW-dokumentti] <http://www.bluetooth.com/Pages/Consumer-Electronics-Market.aspx>

18 Bluetooth technology creates huge opportunities in medical. Luettu 28.3.2015. [WWW-dokumentti]<http://www.bluetooth.com/Pages/Health-Wellness-Market.aspx>

19 Demand for Bluetooth Smart Fitness Devices Exploding. Luettu 28.3.2015. [WWW-dokumentti]<http://www.bluetooth.com/Pages/Sports-Fitness-Market.aspx>

20 Bluetooth Technology Makes Wireless Home Automation Possible. Luettu 28.3.2015. [WWW-dokumentti]<http://www.bluetooth.com/Pages/Smart-Home-Market.aspx>

21 CC2541 SensorTag Development Kit. Luettu 28.3.2015. [WWW-dokumentti] <http://www.ti.com/tool/cc2541dk-sensor>

22 CC2541 SensorTag Quick Start Guide. Luettu 28.3.2015. [WWW-dokumentti]<http://www.ti.com/lit/ml/swru324b/swru324b.pdf>

23 SensorTag User Guide. Luettu 29.3.2015. [WWW-doku-mentti]<http://processors.wiki.ti.com/index.php/SensorTag_User_Guide#>

24 Tri-Axis, User Selectable ± 2g, 4g, 8g, Digital (I2C) Output, 3x3x0.9mm LGA. Luettu 28.3.2015. [WWW-dokumentti] <http://www.kionix.com/accelerometers/kxtj9>

25 TMP006/B Infrared Thermopile Sensor in Chip-Scale Package (Rev. D). Luettu 28.3.2015. [WWW-dokumentti] <http://www.ti.com/product/TMP006/datasheet>

26 Datasheet SHT21. Luettu 28.3.2015. [WWW-dokumentti] <http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT21_Datasheet_V4.pdf>

27 MU-3000 Motion Processing Unit Product Specification Rev 1.3. Luettu 28.3.2015. [WWW-dokumentti] <http://www.invensense.com/mems/gyro/documents/PS-IMU-3000A.pdf>

28 SAW Components Digital barometric pressure sensor. Luettu 28.3.2015. [WWW-dokumentti] <http://en.tdk.eu/inf/57/ds/T5400.pdf>

29 Xtrinsic MAG3110 Three-Axis, Digital Magnetometer. Luettu 28.3.2015. [WWW-dokumentti]<http://cache.freescale.com/files/sensors/doc/data_sheet/MAG3110.pdf?fasp=1&WT_TYPE=Data%20Sheets&WT_VENDOR=FREES

CALE&WT_FILE_FORMAT=pdf&WT_ASSET=Documentation&fileExt=.pdf>

30 Otavio Salvador, Daiane Angolini. Birmingham-Mumbai:Packt Publishing Ltd 2014. s125. ISBN: 978-1-78328-233-3

31 Yocto Project Quick Start. Luettu 4.11.2015. [WWW-dokumentti] <http://www.yoctoproject.org/docs/1.8/yocto-project-qs/yocto-project-qs.html>

32 Setting up the Pi. Luettu 4.11.2015. [WWW-dokumentti] <https://learn.adafruit.com/pibeacon-ibeacon-with-a-raspberry-pi/setting-up-the-pi>

33 SensorTag attribute table. Luettu 5.11.2015. [WWW-dokumentti] <http://processors.wiki.ti.com/images/a/a8/BLE_SensorTag_GATT_Server.pdf