

Turo Salminen

ENERGIAN KULUTUKSEN MINIMOINTI ENERGIAOMAVARAISESSA SULAUTETUSSA JÄRJESTELMÄSSÄ

Diplomityö
Informaatioteknologian ja viestinnän tiedekunta
Tarkastaja: Tero Kivimäki
Tarkastaja: Jukka Vanhala
Marraskuu 2021

TIIVISTELMÄ

Turo Salminen: Energian kulutuksen minimointi energiaomavaraisessa sulautetussa järjestelmässä
Diplomityö
Tampereen yliopisto
Tietotekniikan DI-tutkinto-ohjelma
Marraskuu 2021

Energian keräämisellä tämän työn yhteydessä tarkoitetaan pienten vapaasti saatavilla olevien energioiden muuttamista sähköksi. Energiankeräin ja energianhallintapiiri muodostavat energiankeräyslaitteiston, joka pystyy muuttamaan ja varastoimaan esimerkiksi mekaanista energiaa sähköksi, ja syöttämään tätä ulos sopivalla ulostulojännitteellä. Energiankeräyslaitteistot tuottavat tehoa 1 – 100 W sitä käyttöenergianaan käyttävälle laitteistolle. Energiaa käyttävä laitteisto suorittaa jotain tehtävää ja sillä voi olla prosessointikykyä tai se voi olla analogielektroniikkaa. Tehtävä on yksinkertainen johtuen käytettävissä olevan tehon määrästä, kuten esimerkiksi sähkölukko, joka lukee kulunvalvontatietoja avaimesta ja saa käyttöenergiansa joko ovenkahvan tai avaimen kääntöliikkeestä. Yhä useammin laitteistoilta vaaditaan langattomuutta ja perinteisesti tällaiset laitteet ovat paristokäyttöisiä, langattomaan tiedonsiirtoon kykeneviä laiteita. Jos vapaata energiaa on saatavilla toimintaympäristössä, energiaomavarainen laite pystyy toteuttamaan saman toiminnallisuuden ja jättämään pois pariston säästämällä laitteiston käyttökuluissa. Säästö tulee esimerkiksi paristojen hinnassa ja vaihtokuluissa. Jotta langaton tiedonsiirto on mahdollista toteuttaa energiaomavaraiseen laitteistoon, on radion ja tietoa tuottavan laitteen oltava erittäin vähän virtaa kuluttavia. Tämän saavuttamiseksi piirivalmistajat kehittävät jatkuvasti pienitehoisempia radioita ja mikrokontrollereita. Ohjelmistokehittäjän vastuulle jää tehdä edellä mainittuja hyvällä hyötysuhteella käytäviä ohjelmistoja.

Diplomityö jakautuu kahteen osaan. Kirjallisuustutkimusosassa on tutkittu energiankeräysmenetelmiä ja langatonta tiedonsiirtoa. Tavoitteena on ollut ymmärtää, miksi IoT-tiedonsiirtolaitteiston energiankulutuksen minimointia tarvitaan. Toisessa osassa kerrotaan projektista, jossa työ tehty sekä järjestelmästä, minkä avulla työtä on tehty. Työosuudessa on kehitetty tiedonsiirtosovellusta Internet of Things (IoT) -laitteistolle, joka saa kaiken käyttösähkösä energiankeräyslaitteistolta. Kun sen toiminnallisuus on saatu valmiiksi, sovellusta on optimoitu virrankulutuksen suhteen mahdollisimman pieneksi. Tavoitteena työssä on ollut tutustua tapoihin, joiden avulla energiankulutusta saadaan minimoitua järjestelmäpiirissä, joka sisältää ja käyttää Bluetooth Low Energy (BLE) radiota langattomaan tiedonsiirtoon.

Työn tuloksena saavutettiin huomattava sähkönkulutuksen väheneminen. Minimivirrankulutus optimoimattomassa tiedonsiirtosovelluksessa oli noin 540 μA ja keskiarvo noin 1,1 mA. Optimoinnin tuloksena saavutettiin minimivirrankulutus noin 3,1 μA , kun järjestelmä on syväunutilassa, ja keskiarvokulutus noin 295 μA . Nämä säästöt saavutettiin optimoimalla radioliikenteen toimintaa ja sammuttamalla mahdollisimman paljon turhaan virtaa kuluttavia järjestelmäpiirin osia. Ohjelmistossa pidettiin myös huoli, että syväunutilan ajaksi kaikki sovelluksen käyttämät osat sammutettiin. Ohjelmistoa kehitettäessä tutustuttiin mahdollisimman tarkasti sen toimintaan ja se käytännössä optimoitiin tätä kyseistä sovellusta varten.

Avainsanat: energian kerääminen, energiankulutuksen minimoiminen, sulautettu järjestelmä, bluetooth low energy, apollo3

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Turo Salminen: Minimisation of energy consumption in energy self-sufficient embedded system
Master's thesis
Tampere University
Degree Programme in Information Technology, MSc (Tech)
November 2021

Energy harvesting, in the context of this work, means collection of low amounts of freely available energies and transforming them to electricity. The energy harvester and the energy management system form an energy harvesting device, which is capable to convert and to store, for example, mechanical energy as electricity and also is capable to feed this out with a configurable output voltage. Energy harvesting device typically produces 1 – 100 W power for an energy consuming device. The energy consuming device carries out some specific tasks. It can have, for example, some processing power or it can be an analogic electronic circuit. The task normally is simple due to the minor amount of available energy like, for example, an electric lock, which reads permission information from the key and gets all of its energy from the turning of either the door-handle or the key. More and more often devices have a requirement of being able to do wireless communication and traditionally devices of this kind are battery-operated with a radio-transceiver. If there is free energy available in the environment of the device, the energy harvesting device can omit the battery, but implement similar functionality creating savings in the operating costs. The savings will come, for example, from the price of the battery and in the battery replacement expenses. For wireless communication to be possible in the energy self-sufficient device, the radio and the device that produces the information have to be efficient in power consumption. To achieve this circuit manufacturers continuously develop new lower power radios and microcontrollers. The software developer's responsibility is to develop software that uses these technologies efficiently.

The diploma work is divided into two parts. In the literacy review methods and technologies for energy harvesting and the wireless data transfer have been studied. The objective has been to understand why the minimization of the consumption of energy in the IoT-data transfer equipment is required. The practical part of work introduces the project and the system that has been used for the work. In the work part, a data transfer software for an IoT-equipment, which gets all of its electrical power from the energy harvesting device, has been developed. After the functionality of the software was achieved, the current consumption of the application was optimized as low as possible. In the work the objective has been to become acquainted with ways to minimize current consumption in a system-on-chip.

A considerable decrease in the consumption of the electrical current was reached as a result of the work. The minimum current consumption was about 540 μA and the average about 1,1 mA before optimization. As a result of the optimization the current consumption of about 3,1 μA and average of about 295 μA was achieved. These savings were achieved by optimizing the operation of the radio communication and by turning off as many as possible of the current consuming parts of the system-on-chip. When developing the software, its operation was studied as carefully as possible and it was then optimized for the specific application used in this work.

Keywords: energy harvesting, energy consumption optimization, embedded system, bluetooth low energy, apollo3

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä diplomityö on tehty informaatioteknologian ja viestinnän tiedekunnassa, sähkötekniikan yksikössä, elektroniikan ja sulautettujen järjestelmien tutkimusryhmässä, ENOMA-projektissa. Työn tekemiseen kului noin vuosi aikaa ja asiat kuten energian kerääminen, Bluetooth Low Energy -teknologia, FreeRTOS, ARM Cortex M4F-prosessori ja Ambiq Micro Apollo3 olivat uutta minulle.

Työ on tehty lähes kokonaan etänä johtuen maailmanlaajuisesta koronaviruspandemiasta ja sitä tehdessäni asuin Hervannassa, Siitamassa ja Tammelassa. Matkan varrella olen oppinut paljon sulautetun ohjelmiston kehittämisestä, vianhausta ja C-kielestä, jolla ohjelmisto on kokonaisuudessaan toteutettu. Oppia on kertynyt myös erilaisten kehitysympäristöjen ja mittalaitteiden käyttämisestä. Tämä raportti on myös minun ensimmäinen tämän kokoluokan kirjallinen työ.

Erityisen mielenkiintoista työtä tehdessä oli päästä miettimään kuinka vähentää mikrokontrollerijärjestelmän virrankulutusta. Mikrokontrollerijärjestelmät ovat olleen pääkiinnostuksen kohteeni DI-opinnoissani. RTOS-osatehtävän tekeminen ja sen ajallisen toiminnan ohjaaminen käyttäen RTOS:n API-rajapintoja oli myös mielenkiintoista. Energiankeräys motivoi aiheena koska oikein toteutettuna energiaomavarainen laitteisto säästää luonnonvaroja ja ympäristöasiat ovat lähellä sydäntäni.

Haluan kiittää Tero Kivimäkeä ja Jukka Vanhalaa tämän diplomityön ohjaamisesta ja tarkastamisesta. Lisäksi haluan kiittää äitiä tuesta koko opiskelujen ajalla, Markoa avusta Siitamassa ja eri projekteissa. Jania aina niin mielenkiintoisista mm. tietotekniikkaan liittyvistä keskusteluista, jotka ovat motivoineet minua eteenpäin. Kiipeilykavereita parhaasta seurasta kallioilla ja kiipeilykeskuksella.

Tampereella, 16.11.2021

Turo Salminen

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. ENERGIAN KERÄÄMINEN	4
2.1 Energiankeräyslaitteisto	4
2.2 Pietsosähköiset keräimet	5
2.3 Sähköstaattinen energiankeräin	5
2.4 Magnetrostriktiiviset materiaalit	7
2.5 Induktiiviset keräimet	7
3. LANGATTOMAT IOT-TIEDONSIIRTOPROTOKOLLAT	12
3.1 Langattomien tiedonsiirtoteknologioiden vertailua	12
3.2 Vertailu projektin kannalta	13
3.3 Bluetooth Low Energy	15
3.3.1 Arkkitehtuuri	15
3.3.2 Yleinen saavutettavuusprofiili GAP	17
3.3.3 Piirreprotokolla ATT	19
3.3.4 Yleinen piirreprofiili GATT	20
3.3.5 Sovellukset	21
4. PROJEKTIN KUVAUS	23
4.1 ENOMA	23
4.2 Prototyypijärjestelmä	24
5. MIKROKONTROLLERIJÄRJESTELMÄ	26
5.1 FreeRTOS	26
5.2 Ambiq Micro Apollo3 blue	28
5.2.1 BLE-lohko	31
5.2.2 AD-muunnin	32
5.2.3 Apollo3 Blue MCU EVB -kehitysalusta	34
5.2.4 Ambiq Data Transfer Profile	35
5.3 Kehitysympäristö	36
5.3.1 Mittaus- ja virheenjäljitysympäristö	37
5.3.2 Mittauslaitteisto	37
5.3.3 ADXL1003-kiihtyvyyssanturi protolevy	38
6. LAITTEISTON TOIMINTA	40
6.1 Toiminnan kuvaus	40
6.2 Virrankulutuksen minimoiminen	42
6.3 Mittaustulokset ja analyysi	44
6.4 Järjestelmän jatkokehitysideoita	49
7. YHTEENVETO	51
LÄHTEET	53
LIITE A: KIIHTYVYSSANTURIN MITTAUS -OHJELMA ESIMERKKI	57

KUVALUETTELO

<i>Kuva 1</i> Energiankeräyslaitteisto [4].....	4
<i>Kuva 2</i> Pietosähköinen energiankeräin [1].....	5
<i>Kuva 3</i> Sähköstaattinen energiankeräin -vakio varaus [1].....	6
<i>Kuva 4</i> Sähköstaattinen energiankeräin -vakio jännite [1].....	6
<i>Kuva 5</i> Magnetostriktiivinen energiankeräin [26].....	7
<i>Kuva 6</i> Sähkömagneettinen induktio [25].....	8
<i>Kuva 7</i> Pyörivä sähkögeneraattori [25].....	8
<i>Kuva 8</i> Sähkömagneettiseen induktioon perustuva energiankeräin [1].....	9
<i>Kuva 9</i> Induktiivisen keräimen ulostulojännite (RMS).....	10
<i>Kuva 10</i> Induktiivisen keräimen tehontuotto.....	10
<i>Kuva 11</i> Radioteknologioiden teho vs kantama vs tiedonsiirtonopeus [24].....	14
<i>Kuva 12</i> BLE-arkkitehtuurin lohkokaavio [41].....	16
<i>Kuva 13</i> L2CAP-paketti [41].....	17
<i>Kuva 14</i> Yhteystapahtuma ja aikaväli [41].....	18
<i>Kuva 15</i> GAP-turvallisuus tiloja ja tasoja [41].....	19
<i>Kuva 16</i> BLE-piirre [41].....	20
<i>Kuva 17</i> BLE-rengin ilmoituspaketti [41].....	20
<i>Kuva 18</i> Rengin toteuttama palvelu.....	21
<i>Kuva 19</i> BLE-sovelluksen hoitama tehtävä [41].....	22
<i>Kuva 20</i> Tampereen yliopistossa kehitetty energiankeräin.....	24
<i>Kuva 21</i> Tampereen yliopistossa kehitetty energiaomavarainenlaitteisto.....	25
<i>Kuva 22</i> FreeRTOS-osatehtävän tilat [24].....	27
<i>Kuva 23</i> Apollo3 blue -lohkokaavio [31].....	29
<i>Kuva 24</i> N-MOSFET [34].....	29
<i>Kuva 25</i> Apollo3 blue BLE-osalohko [31].....	32
<i>Kuva 26</i> AD-muuntimen lohkokaavio [31].....	33
<i>Kuva 27</i> Apollo3 kehitysalusta [2].....	34
<i>Kuva 28</i> AMDTP-paketti [43].....	35
<i>Kuva 29</i> AMDTP:n toiminnan lohkokaavio [41].....	36
<i>Kuva 30</i> Projektissa valmistettu ADXL1003-kiihtyvyyssanturi protolevy.....	39
<i>Kuva 31</i> Lähettimen optimoimaton virrankulutus ajanfunktiona [mA / s].....	44
<i>Kuva 32</i> Lähettimen optimoimattoman lähetyksen kuvaaja.....	45
<i>Kuva 33</i> Lähettimen optimoitu lähetys.....	45
<i>Kuva 34</i> Maksimi BLE-yhteyden aikaväli, 0 latenssi.....	46
<i>Kuva 35</i> BLE-yhteyden aikavälin pidentäminen liikaa.....	46
<i>Kuva 36</i> Lyhyt aikaväli ja latenssi 20.....	47
<i>Kuva 37</i> Syväunesta poistumisaika.....	47
<i>Kuva 38</i> Syväuneen meno.....	48
<i>Kuva 39</i> Tiedonlähettäminen ja unitilat.....	49

LYHENTEET JA MERKINNÄT

ADC	Analog to Digital Converter, analogia-digitaalimuunnin
AMDTP	Amibq Micro Data Transfer Protocol, Bluetooth-profiili
ATT	Attribute protocol, piirreprotokolla
BLE	Bluetooth Low Energy, langaton tiedonsiirtoprotokolla
Bluetooth SIG	Bluetooth Special Interest Group, standardoimisorganisaatio
BOD	Brown-Out-Detection, käyttöjännitteen alarajan tarkkailu
CC-lisenssi	Creative Commons -lisenssi
DMA	Direct Memory Access, oikosiirto-ohjain
emf	electromotive force, sähkömotorinen voima
FS	Full-Scale, mitta-alueen maksimiarvo
GAP	Generic Access Profile, yleinen saavutettavuusprofiili
GATT	Generic Attribute Profile, yleinen piirreprofiili
GPIO	General Purpose I/O, ohjelmoitava sisään-ulos
HCI	Host Controller Interface, BLE-arkkitehtuurin osa
IoT	Internet of Things, esineiden internet
ISM	Industrial, Scientific and Medical, teollisuuden, tieteen ja terveydenhuollon
ITM	Instrumentation Trace Macrocell, vianhaku- ja kommunikointiväylä
LR-WPAN	Low-Rate Wireless Personal Area Network, matalanopeuksinen langaton henkilökohtainen verkko
L2CAP	Logical Link Control and Adaptation Protocol, Looginen yhteydenohjaus ja sovitusprotokolla
MIT-lisenssi	The MIT (Massachusetts Institute of Technology) -lisenssi
MTU	Maximum Transmit Unit, maksimisiirtoyksikkö
NMR	Normal Mode Rejection, häiriönvaimennuskyky
OTA	Over-the-Air, langaton ohjelmistopäivitys
RFID	Radio Frequency Identification, radioteknologia
RMS	Root Mean Square, neliöllinen keskiarvo, tehollisarvo
RTOS	Real-Time Operating System, reaaliaikakäyttöjärjestelmä
RX	Receive, vastaanotto
SDK	Software Development Kit, ohjelmistokehityskirjasto
SRAM	Static Random Access Memory, staattinen hajasaantimuisti
SoC	System on Chip, järjestelmäpiiri
TPIU	Trace Port Interface Unit, vianhaku- ja kommunikointiväylä
TX	Transmit, lähetys
ULP	Ultra-Low Power, erittäin pienitehoinen
URL	Uniform Resource Locator, verkkosivun osoite
USB	Universal Serial Bus, sarjaväyläarkkitehtuuri
UUID	Universally Unique Identifier, universaali tunnistenumero
WLAN	Wireless Local Area Network, langaton lähiverkko
WPAN	Wireless Personal Area Network, langaton henkilökohtainen verkko

1. JOHDANTO

Ihminen on oppinut keräämään energiaa ympäristöstään jo kauan sitten. Vanhoja menetelmiä tähän on tuulimyllyt, joita käytettiin muun muassa viljan jauhamiseen, ja vesivoimalla toimivat myllyt, joita käytettiin myös esimerkiksi sahoina eli sahamylyinä. Sittemmin vesi- ja tuulivoimaa on käytetty sähkön tuottamiseen pyörittämällä sähkögeneraattoria liike-energian avulla. Näitä energian keräystapoja yhdistää se, että ne ovat pienimmilläänkin melko kookkaita ja kalliita investointeja.

Nykyaikana energiaa kerätään yhä enemmän myös aurinkopaneeleilla, joilla auringon säteilemä energia muutetaan sähköksi erilaisilla menetelmillä. Myös tämä vaatii kiinteän asennuksen ja suhteellisen paljon pinta-alaa. Joissakin sähköautoissa on kattopinta-alaa peitetty aurinkopaneeleilla energian keräämiseksi [5].

Edullista näissä edellä mainituissa tavoissa on, että niistä on saatavilla suuria määriä energiaa pitkäksi aikaa luokassa kilo- megawatteja [6]. Huonoa se, että ne ovat sidottuna sijaintiinsa, kuten koskien yhteyteen tai alueille missä on tarpeeksi tuulta vuosittain.

Pienen kokoluokan energiankeräystekniikoita ovat muun muassa pietsosähköisyys, lämpösähköinen generaattori, sähkömagneettinen induktio, sähkömagneettinen säteily, valosähköinen ilmiö. Näistä energiaa kerätään milli- ja mikrowatti luokassa [6] ja nämä ovat kiinnostavia nykyaikaisia energiankeräinjärjestelmiä kehitettäessä.

Nykyaikana on myös tarve kehittää mahdollisimman hyvällä hyötysuhteella toimivia pieniä Internet-of-Things- (IoT) eli tiedonsiirto-, pääte- ja oheislaitteistoja, kuten mikrokontrollereita, radioita, sensoreita ja muuta elektroniikkaa, jotka käyttävät energiankeräimien tuottamaa sähköä käyttövoimanaan mahdollisimman hyvällä hyötysuhteella. Näitä tarvitaan tutkittaviin ja käytössä yleistyviin energiaomavaraisiin laitteistoihin, jotka pyrkivät korvaamaan akut tai paristot, joista kertyy ongelmajätettä.

Energiaomavaraiset laitteet voivat toimia yksinkertaisena valokytkimen tapaisena pääte-laitteena, mutta usein laitetta käytetään keräämään ja välittämään tietoa. Esimerkiksi autonrenkaan sisälle sijoitettua energiaomavaraista laitetta voidaan käyttää välittämään tietoa renkaan ilmanpaineesta ja lämpötilasta. Tietoa kerätään myös älylukoista, vaatteista, sydämentahdistimista ja esimerkiksi kulunvalvonta-, kosteus- ja lämpötilatietoja rakennuksista.

RFID-tekniikkaan perustuvalla energiankeräimellä varustettuja henkilönlämpötilamonitoreja käytetään myös globaalin COVID-19 pandemian torjumiseen [21]. Laite kerää energiansa RFID-lukijasta käyttäen radioaaltoja ja siltä voidaan lukea tietoa 3,5 metrin etäisyyteen asti. Terveydenhuollon henkilökunta voi töihin tullessaan ladata laitteen käyttämällä sitä RFID-lukijassa, mitata sen jälkeen lämpötilan otsaltaan, ja tulostaa mittauksen lukijan kautta esimerkiksi televisionäytölle [22].

Pienet energiaomavaraiset laitteistot ovat kärsineet elektroniikan suuresta virrankulutuksesta. Mikrokontrollereiden ja radiolähettimien virrankulutus sekä energiavarastojen ja energian varastointiin käytettävän elektroniikan hyötysuhde ovat parantunut huomattavasti 2000-luvulla ja tämä on laajentanut energiaomavaraisten laitteiden käyttömahdollisuuksia.

Tietoteknisen elektroniikan ja tietoteknisten laitteiden energian kulutuksen minimoimista ja omavaraisuutta tulisi kehittää ja siihen tulisi kiinnittää huomiota suunnittelussa myös ilmaston lämpenemisen vuoksi. Hiilidioksidipäästöjen (CO₂) vähentäminen tulee olemaan koko ajan suurempi teema maailman laajuisesti ja ohjaamaan markkinoita ja lainsäädäntöä. Ihmisen aiheuttama ilmaston muutos pakottaa ihmiskunnan tähtäämään energiatehokkuuden parantamiseen ja CO₂ päästöjen vähentämiseen. Seuraukset kuten keskilämpötilan nousu, poikkeavat tulvat ja kuivuus sekä merenpinnan nousu ovat jo nyt nähtävissä [19].

Energiaomavaraisilla laitteistoilla, joista raskasmetalleja sisältävät akut ja paristot ovat korvattu energiakeräimellä, aiheutuisi myös pienempi kuormitus ympäristölle. Vaikka Suomessa jätteenlajittelu paranee jatkuvasti, pääkaupunkiseudulla vuonna 2018 sekajätteestä 1 % on jättejakeessa sähkölaitteet ja akut [13]. Sekajätettä Suomessa vuonna 2018 kertyi 1,5 miljoonaa tonnia [12], joten jätettä sähkölaitteet ja akut menee sekajätteeseen pelkästään Suomessa noin 15 000 tonnia vuodessa. Tästä osa sisältää akkuja ja paristoja, ja muodostaa maailmanlaajuisesti huomattavan määrän, joka päättyy kaatopaikoille tai jätteenpolttolaitokselle. Tarpeeseen hyvin suunniteltu energiaomavarainen laite, voi toiminnallaan säästää hiilidioksidipäästöjä oikein käytettynä. Tällaisia voisivat olla valvontalaitteet, jotka ilmaisevat huollontarpeesta ennen jonkin laitteen pysyvää vioittumista tai varoittavat mahdollisen kosteusvaurion synnystä jossakin rakenteessa, tai kulunvalvonta, jolla voitaisiin säätää valo- ja lämmitystä kiinteistöissä.

Tämä diplomityö on jaoteltu lukuihin seuraavasti: luku 2 kertoo energian keräämisestä, luku 3 langattomasta tiedonsiirrosta, luku 4 projektista johon työ on tehty, luku 5 esittelee

fyysisen laitteiston ja työkalut, luku 6 ohjelman toiminnan ja virrankulutuksen minimoimiseksi tehdyt asiat, sekä analyysin mittaustuloksista ja viimeisessä luvussa 7 on lopuksi vielä työn yhteenveto ja pohdinta.

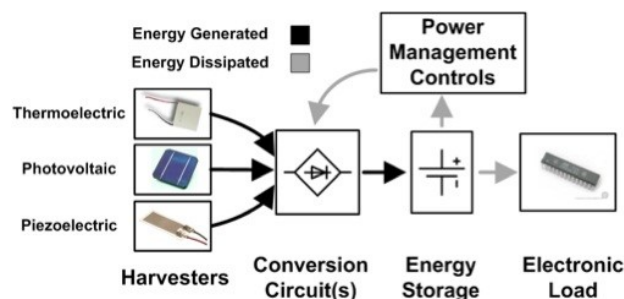
2. ENERGIAN KERÄÄMINEN

Tämän diplomityön puitteissa energian kerääminen tarkoittaa ympäristöstä vapaasti saatavan energian muuntamista sähköksi käyttämällä erilaisia energiankeräinlaitteita. Ulostulevat sähkötehot keräinlaitteistolta ovat alle 100 milliwattia. Tämä sulkee pois esimerkiksi alle neliömetrin kokoiset aurinkopaneelijärjestelmät mökkisähkötä varten, tai muun isomman luokan hajautetun paikallisen energian tuotannon, kuten isommat aurinkopaneeli- tai tuulisähkölaitteet ja aggregaatit.

Energiaomavaraisella laitteistolla tarkoitetaan tässä työssä laitteistoa, joka saa kaiken tarvitsemansa käyttösähkön energiankeräinlaitteistolta ja kuluttaa sitä mahdollisimman vähän ja tehokkaasti. Se pystyy muuttamaan saamansa energian haluttuun muotoon, tavallisesti tasasähköksi ja sisältää energiavaraston, kuten akun. Laitteisto myös suorittaa jotain tehtävää ja voi olla yhteydessä ulkomaailmaan langattomasti, jos on tarve siirtää jotain tietoa. Tehtävä on tyypillisesti yksinkertainen johtuen laitteiston luonteesta.

2.1 Energiankeräyslaitteisto

Energiankeräyslaitteisto koostuu seuraavista komponenteista. Energiankeräin, eli jokin teknologia, joka muuttaa vapaasti saatavissa olevaa energiaa, esimerkiksi valoa, ääntä tai lämpötilaeron sähköksi. Energian talteenotto- tai hallintapiiri hoitaa energiankeräimen tuottaman sähkön muuttamisen muotoon, missä se on varastoivissa tai kuormana toimivan laitteiston käytettävissä. Energiavarastona käytetään yleensä superkondensaattoria tai akkua. Kuorma, eli tuotettua sähköä käyttävä laite voi olla esimerkiksi mikrokontrolleista, radiosta ja anturista koostuva tiedonsiirtoon käytettävä IoT-laite [4]. Kuva 1 esittää energiankeräyslaitteiston lohkokaavion.



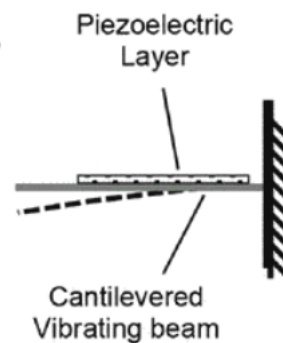
Kuva 1 Energiankeräyslaitteisto [4]

Energian talteenotto- ja hallintaelektronikka koostuu jännitteen tasasuuntaajasta ja hakurivirtalähteestä, jolla ulostuleva tasajännite voidaan sovittaa kuormalle sopivaksi.

2.2 Pietsosähköiset keräimet

Nämä energiankeräimet muuttavat mekaanista energiaa sähköksi pietsosähköisen ilmiön avulla. Materiaalit, joilla on pietsosähköisiä ominaisuuksia, muodostavat sähköisen varauksen, kun niihin kohdistetaan mekaanista painetta. Tästä mekaanisesta energiasta ei kuitenkaan muutu kuin osa. Ne myös toimivat toisin päin, eli muuttavat fyysistä muotoaan, kun niiden sähkövaraukseen kohdistetaan muutos. Tästä hyvänä esimerkkinä pietsosähköinen kaiutin ja mikrofoni [30].

Tämänlaisella energiankeräimellä voitaisiin kerätä sähköä värinästä esimerkiksi siten, että keräin on pitkänmallinen ja kapea levy, joka on kiinnitetty toisesta päästä. Toisessa päässä on massa, joka saa levyn ja pietsosähköisen materiaalin värähtelemään värinän voimasta. Kuva 2 esittää tällaisen keräimen.



Kuva 2 Pietsosähköinen energiankeräin [1]

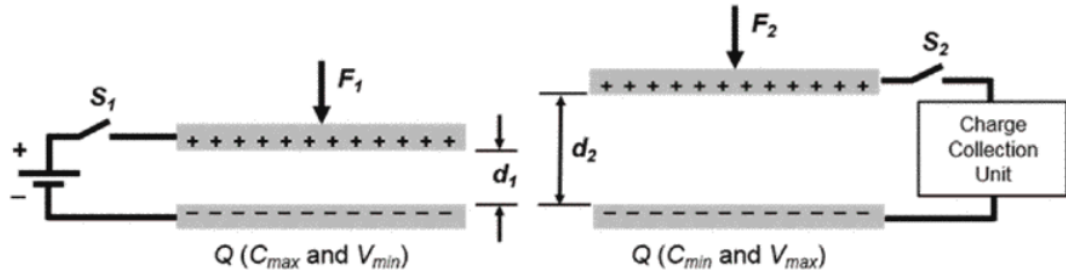
Pietsosähköinen materiaali venytyksen alaisena tuottaa vaihtosähköä levyn taipuessa ylös ja alas. Ulostuleva jännite tällaisessa voi olla satoja voltteja, mutta vain kapealla kaistalla, missä värinä on energiankeräimen resonanssitaajuudella [1].

Pietsosähköisen energiankeräimen tuottama teho on kapasitiivista ja sen ulostuloimpedanssi on korkea. Nämä ominaisuudet aiheuttavat tarpeen tehonsovituspierin impedanssien sovittamiseksi, koska virtaa kuluttavat kuormat ovat pieni-impedanssisia [26].

2.3 Sähköstaattinen energiankeräin

Kondensaattori koostuu kahdesta erimerkkisen varauksen omaavasta levystä, jotka ovat erillään toisistaan, joko ilman tai jonkin eristeen avulla. Sähköstaattisen energiankeräimen voi valmistaa kondensaattorista, jonka kapasitanssia voidaan mekaanisesti muuttaa muuttamalla näiden kahden levyn etäisyyttä toisistaan. Energian keräämiseen tällaisesta keräinlaitteesta on kaksi tapaa. Ensimmäisessä kondensaattorin varaus pidetään

vakiona ja toisessa pidetään jännite vakiona varauksen voidessa muuttua. Ensimmäisessä tapauksessa energiaa kerätään siten, että levyt ovat alkutilanteessa lähellä toisiinsa ja kondensaattoriin ladataan alkuvaraus. Kuva 3 esittää tällaista tilannetta.



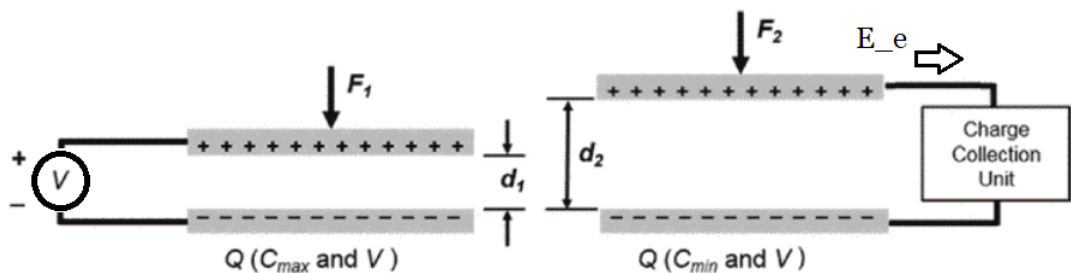
Kuva 3 Sähköstaattinen energiankeräin -vakio varaus [1]

Varaus on vakio ja kapasitanssi on nyt maksimiarvossaan. Sitten kondensaattorin levyjä vedetään kauemmas toisistaan ja kun on saavutettu maksimietäisyys, kondensaattori puretaan. Maksimietäisyydellä kapasitanssi on minimiarvossaan ja jännite on kasvanut alkutilanteen arvosta, koska varaus on pysynyt samana, kuten nähdään alla olevasta kaavasta

$$Q = C_{max}V_{min} = C_{min}V_{max} \quad (1)$$

jossa Q on varaus, C on kapasitanssi ja V on jännite.

Toisessa tilanteessa, missä jännite pidetään vakiona ja varauksen annetaan muuttua, kondensaattorin kapasitanssi on maksimiarvossaan ja siihen kytketään tasajännitelähde. Kuva 4 esittää tätä tapausta. Tasajännitelähteen ulostulojännite pidetään samana, kun levyt vedetään kauemmaksi toisistaan mekaanisen energian avulla. Kondensaattorin kapasitanssi ja varaus pienenee minimiarvoonsa ja tuottaa sähköenergiaa.



Kuva 4 Sähköstaattinen energiankeräin -vakio jännite [1]

Sähköenergian määrä, joka saadaan aikaiseksi ulkoisen työn avulla, saadaan kaavalla

$$E_e = \frac{1}{2}(C_{max} - C_{min})V^2 \quad (2)$$

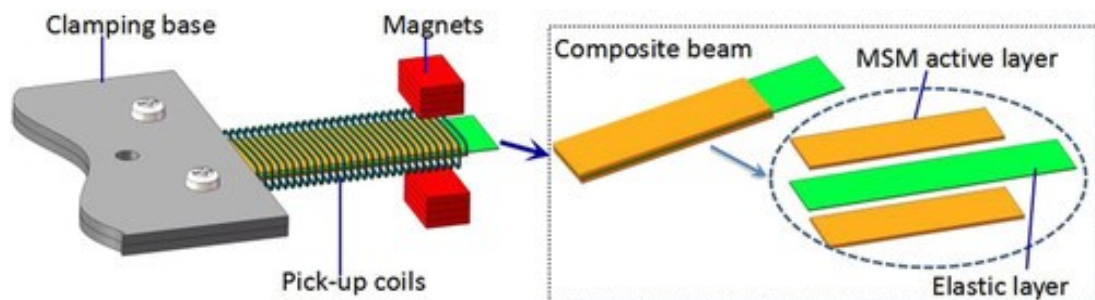
jossa E_e on saatava sähköenergia, C on kapasitanssi ja V on jännite.

Vaatus kondensaattorin lataamiselle alkuvaraukseen molemmissa tilanteissa ja tasajännitelähteen käyttö toisessa rajoittaa tämän menetelmän käyttöä joissakin energiankeräämistapauksissa [1].

2.4 Magnetrostriktiiviset materiaalit

Nämä ovat ferromagneettisia materiaaleja, jotka muuttavat muotoansa niihin kohdistuva magneettikentän muuttuessa. Muutos koossa tai tilavuudessa on hyvin pientä. Tämä toimii myös toisin päin, eli kun näitä materiaaleja mekaanisesti rasitetaan, eli niiden kokoa muutetaan, saadaan aikaan rasituksen mukaan muuttuva magneettikenttä. Tätä kutsutaan myös Villarin ilmiöksi. Tätä muuttuvaa magneettikenttää voidaan käyttää synnyttämään kelaan sähkövirta ja siten nämä materiaalit soveltuvat energian keräämiseen. Materiaalien on oltava esimagnetoituja tai magnetoituina kestopagneeteilla. Kuva 5 esittää tällaista keräintä. Kuvassa esimagnetointi on tehty ulkoisilla kestopagneeteilla, joiden avulla ulostulevan jännitteen määrä on saatu suuremmaksi. Magnetrostriktiivinen keräin, joka on kuvassa oranssi-vihreä -liuska, värähtelee tärinän voimasta suhteessa magneetteihin, jotka ovat kiinnitetty kiinteästi laitteiston runkoon [1][26].

Energian keräämiseen tätä teknologiaa voidaan käyttää paikoissa missä on saatavilla matalataajuisia tärinää. Teknologia tuottaa vaihtosähköä, joka voi amplitudiltaan olla millivolteista muutamiin voltteihin ja tehontuotoltaan maksimissaan luokassa milliwatteja [26].



Kuva 5 Magnetrostriktiivinen energiankeräin [26]

Magnetrostriktiivisyyttä hyödyntävä energiankeräin on ulostuloltaan induktiivinen ja matala-impedanssinen. Tästä on hyötyä energian talteenottopiirin kannalta, sillä kuormat, joihin energiaa käytetään, ovat myös matala-impedanssisia [26].

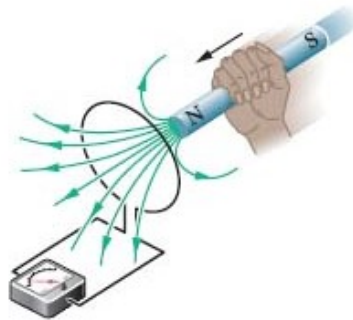
2.5 Induktiiviset keräimet

Induktiivinen energiankeräin on käytännössä sähkögeneraattori, joka käyttää sähkömagneettista induktiota tuottaakseen sähkövirtaa. Faradeyn induktiolain mukaan, johtimeen

voidaan indusoida sähkömotorinen voima (emf) muuttamalla magneettikentän voimakkuutta johtimessa. Kuva 6 esittää tätä muutosta, jossa magneettia liikutetaan suhteessa johtimeen. Virran ja emf:n voimakkuus riippuu magneettikentän voimakkuudesta ja muutosnopeudesta. Faradayn laki on kuvattu kaavassa

$$emf = -\frac{d\Phi_B}{dt} \quad (3)$$

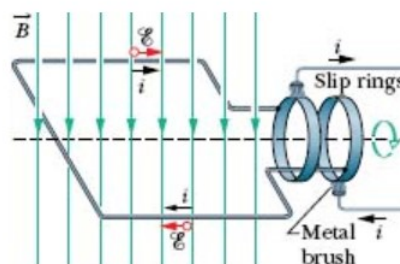
jossa emf on sähkömotorinenvoima ja Φ_B on magneettivuon voimakkuus. Sähkömotorinen voima joka johtimeen indusoituu, on käytännössä jännitettä ja voidaan merkitä myös V_{emf} [1].



Kuva 6 Sähkömagneettinen induktio [25]

Induktiivinen keräin koostuu kuparilangasta käämistä kelasta, joka toimii myös sähköjohtimena, kestopagneetista ja rungosta. Kestomagneettia tai kuparikelaa liikutetaan suhteessa toisiinsa mekaanisen liike-energian avulla, niin että toinen on paikoillaan kiinni rungossa, tämä aiheuttaa kelaan kohdistuvassa magneettikentässä muutoksen ja kelaan syntyy jännite sähkömotorisenvoiman takia. Näin saadaan muutettua mekaanista liike-energiaa sähköksi [1].

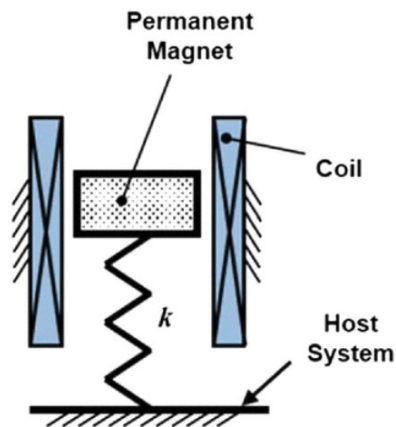
Induktiivinen keräin voi olla pyörivä tai lineaarinen. Pyörivässä sähkögeneraattorissa muutetaan liike-energiaa sähkömotoriseksi voimaksi pyörittämällä kela paikoillaan olevassa magneettikentässä. Kuva 7 havainnollistaa tätä toimintaa.



Kuva 7 Pyörivä sähkögeneraattori [25]

Yllä olevasta kuvasta voidaan havaita, että virran i suunta vaihtuu johdinkiepin ollessa pystysuorassa, siis kuvan ylä- ja alaosassa olevat i -vektorit vaihtavat suuntaa. Tämä johtuu siitä, että muutos magneettikentässä joka johtimeen kohdistuu vaihtaa etumerkkiä. Ylempi i -vektori liikkuu kuvassa kohti magneettikenttää ja johdinkiepin yläosaan indusoituu negatiivinen jännite, alempi i -vektori ja johdinkiepin alaosa liikkuu pois päin siitä ja siihen indusoituu positiivinen jännite [25].

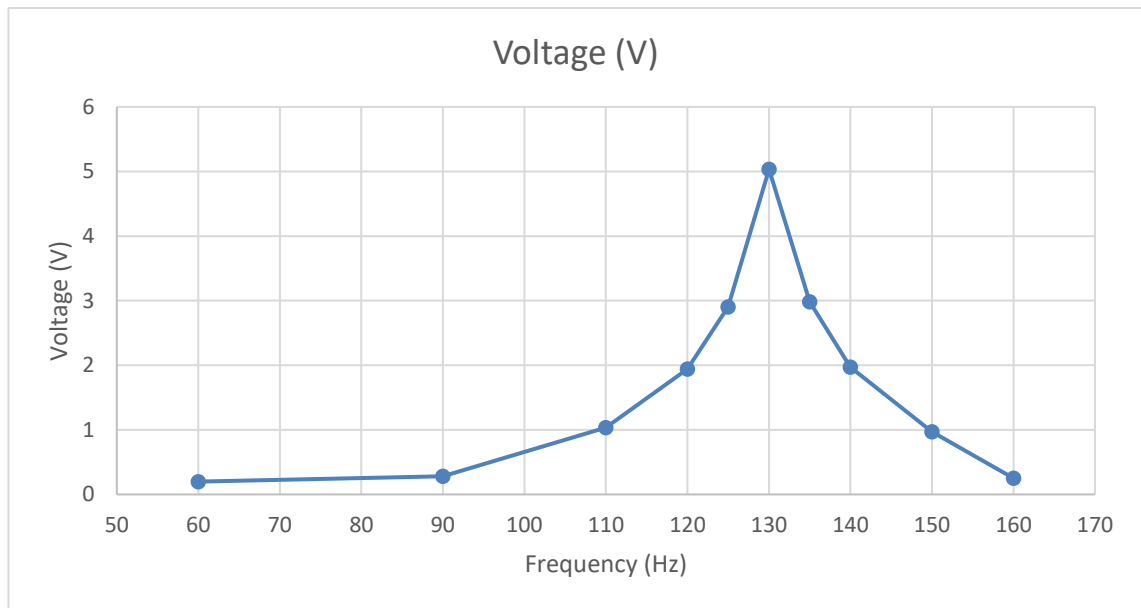
Linearisessa induktiivisessa keräimessä kela ja magneetti liikkuvat suhteessa toisiinsa edestakaisin yhdellä akselilla. Tällainen voi olla esimerkiksi putki, jonka keskikohdan ympärille on käämitty kuparilangasta kela ja sen sisällä liikkuu kestopagneetti edestakaisin. Kuva 8 esittää tällaista energiankeräintä. Kestomagneetti siis kulkee edestakaisin yhdellä akselilla, kelan läpi [3].



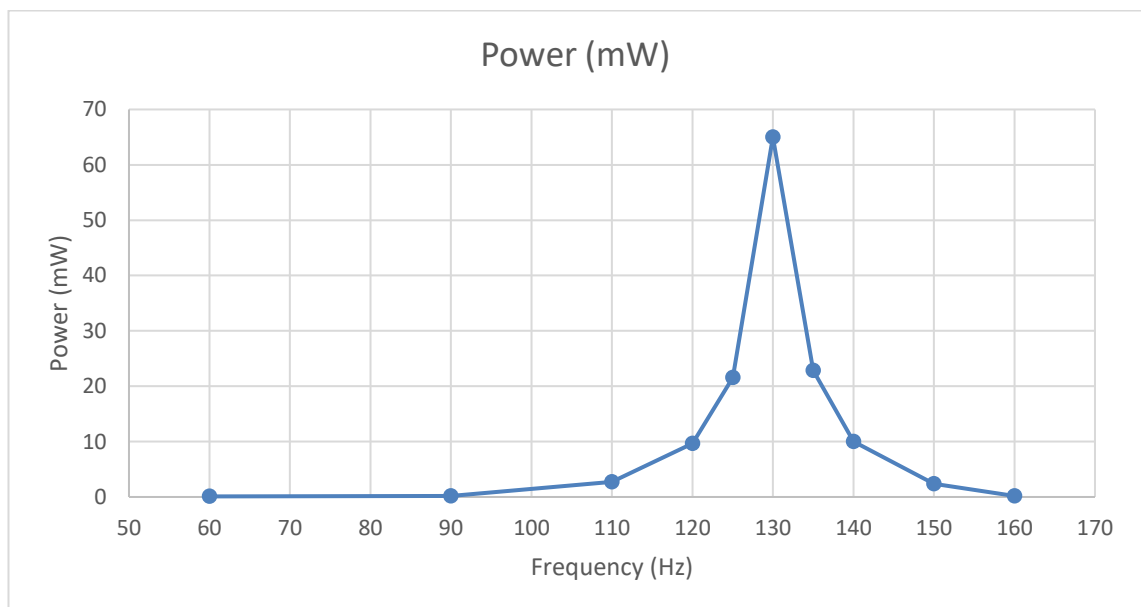
Kuva 8 Sähkömagneettiseen induktioon perustuva energiankeräin [1]

Lineaarisen keräimen tehontuottoon vaikuttaa käämityn kuparilangan kierroslukumäärä, kestopagneetin voimakkuus sekä nopeus, jolla käämi ja magneetti liikkuvat toisiinsa nähden, johon puolestaan vaikuttaa kestopagneetin massa, jousen jäykkyys ja värinän taajuus. Värinän taajuuden on oltava kohtalaisen suurta, jotta saadaan aikaiseksi riittävä liikenopeus. Tämän vaatimuksen seurauksena lineaarinen induktiivinen keräin ei ole tehontuotoltaan optimaalinen vaihtoehto tilanteissa, joissa tarjolla oleva värinä on matalataajuisia. Yllä mainitut tehontuottoon vaikuttavat ominaisuudet vaikuttavat myös keräimen kokoon. Laitteisto optimoidaan suunnitteluvaiheessa jollekin taajuusalueelle näiden parametrien pohjalta. Esimerkiksi jousi ja magneetti valitaan siten, että keräimen resonanssitaajuus vastaa tarjolla olevan lähde-energian taajuutta. Resonanssitaajuudella massan liikerata on maksimissaan. Tämä tapahtuu, kun jousen ja magneetin yhdistelmän ominaistaajuus ja värinän taajuus ovat samat. Liikerata vaimenee värinän taajuuden kasvaessa tai laskiessa tältä resonanssitaajuudelta. Vaimenemisen voimakkuuteen vaikuttaa järjestelmän kyky vastustaa värinää, tai jousen jäykkyyden ja massan suhde [1][25].

Seuraavat kuvat ovat projektissa käytetyn lineaarisen induktiivisen keräimen jännitteen- ja tehontuoton mittauksista. Kuva 9 havainnollistaa ulostulevan jännitteen suhdetta taajuuteen. Kuva 10 havainnollistaa ulostulevan tehon suhdetta taajuuteen.



Kuva 9 Induktiivisen keräimen ulostulojännite (RMS)



Kuva 10 Induktiivisen keräimen tehontuotto

Teho on laskettu 389,9 ohmin kuormavastukseen. Mittaus on tehty tärinäällä energi-ankeräintä sinimuotoisella tärinäällä eri taajuuksilla. Ulostulevasta jännitteestä on laskettu teho taajuuksille, jotka on kuvissa merkitty pisteillä. Kuvaajat demonstroivat hyvin resonanssin vaikutusta tehontuottoon, sen ollessa suurimmillaan 130 Hz kohdalla.

Energiakeräimenvärähtelijän resonanssitaajuudella tarkoitetaan taajuutta, jolla saadaan suurin teho tuotettua. Tämä taajuus saadaan kaavalla:

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (4)$$

jossa f_0 on harmonisen värähtelijän taajuus, k jousivakio ja m magneetin massa [25]. Keräimessä magneetti on massaltaan 3 g ja jousivakio 2170. Harvesterin resonanssitaajuuteen vaikuttavat myös harvesterin syöttämän kuorman, eli energiankeräyslaitteiston, sähköiset ominaisuudet. Kaavalla 4 laskettu resonanssitaajuus on ~135 Hz ja mitattu 130 Hz, ero näissä johtuu esimerkiksi siitä, että magneetti on liimattu jouseen ja liima lisää magneetin massaa ja siitä, että energiankeräintä kuormitetaan sähköisesti. Värähtelijän resonanssitaajuus on kuitenkin määräävä tekijä resonanssitaajuuden suhteen, muiden vaikutusten ollessa vähäisempiä.

3. LANGATTOMAT IOT-TIEDONSIIRTOPROTOKOLLAT

3.1 Langattomien tiedonsiirtoteknologioiden vertailua

Projektissa ei ole tarve siirtää tietoa kuin maksimissaan muutamia kymmeniä metrejä, tiedonsiirtonopeudella 14kbps – 30kbps. Vaihtoehtoina käytettäväksi langattomaksi tiedonsiirtoteknologiaksi on lyhyen kantaman matalaenergia IoT-radiot, kuten esimerkiksi Bluetooth Low Energy, Radio Frequency Identification (RFID) ja Zigbee.

Langattomia tiedonsiirtoteknologioita on olemassa todella monenlaisia. Kaikki ne ovat erikoistuneet jollekin osa-alueelle, kuten tiedonsiirtonopeus, kantama, hinta tai tehonkulutus, ja keskittyvät joihinkin ominaisuuksiin näiden alueiden sisällä. Esimerkkeinä voisi olla Wireless Local Area Network (WLAN)- ja matkapuhelin -verkkoteknologiat. Näissä tiedonsiirtomäärät ovat samanlaisia, mutta kantama eroaa paljon. Näiden osa-alueiden parametreihin voi vaikuttaa jossain määrin sovellustasolla. Energiatavokkaan sovelluksen toteuttamisessa yhteensovitetään resursseja sekä vaatimuksia, esimerkiksi siirrettävän tiedon määrän ja tiedonsiirtonopeuden suhteen, tai lähetystehon asetusten, eli kantaman vaatimusten mukaan. Sovelluksessa ei kannata käyttää maksimi suoritusnopeutta, jos se ei sitä vaadi. Mutta ei ole järkeä väärinkäyttää radiota, joka on kehitetty pitkälle kantamalle ja suurelle tiedonsiirtonopeudelle ajamalla sitä aliteholla, vaan tällöin tulisi vaihtaa teknologia soveltuvampaan.

Pääasiallinen tiedonsiirtoteknologia projektissa on Bluetooth Low Energy (BLE)-radio, joka on esitelty Bluetooth 4.0 versiossa, vuonna 2010 [51]. Sitä ei tule sekoittaa perinteisempään Bluetooth Classic -radioon, vaikka molemmat ovat Bluetooth Special Interest Group:in (Bluetooth SIG) ylläpitämiä standardeja. Bluetooth nimestä huolimatta nämä eivät ole yhteensopivia teknologioita. Ne eroavat siten että BLE on alusta alkaen suunniteltu erittäin pienivirtaiseksi radioksi ja Bluetooth Classic korvaamaan kaapelit, joilla laitteita on yhdistetty toisiinsa, kuten tietokoneen ja näppäimistön välillä. Molemmat toimivat vapaalla 2.4 GHz:n Industrial, Scientific and Medical (ISM) -radiotaajuusalueella [10]. Joissakin Bluetooth Classic radioissa on myös BLE-yhteensopivuus eli 'dual-mode', kuten useimmissa älypuhelimissa. BLE on wireless personal area network (WPAN) teknologia ja se sijoittuu jonnekin lyhyen kantaman, keskinopean ja matalan energian kulutuksen paikoille.

RFID-teknologia on Neuvostoliitossa jo 40-luvulla vakoilutarkoituksiin kehitetty radioteknologia, mutta voidaan tuntea yleisemmin kulunvalvonnasta, missä RFID-tagia näyttämällä lukijaan voidaan ohjata sähkölukkoja. Tagit voivat olla aktiivisia tai passiivisia, ja ne sisältävät antennin ja yksinkertaisen mikroprosessorin piirilevyllä. Aktiivinen tagi voi olla osana järjestelmää, joka saa virtaa akusta tai paristosta. Tämä eroaa passiivisesta tagista siten, että passiivinen tagi saa energiansa radioaalloista lukijalta antennin avulla, eikä sisällä energiavarastoa. Lukija on yhteydessä verkkoon ja kommunikoi tagin kanssa siten, että yleensä tagi lähettää vain oman uniikin tunnisteensa [15]. RFID-kantama on senttimetreistä yli 20 metriin passiivisena ja aktiivisena 100 metriin asti. Se toimii taajuuksaistoilla 125 – 134 kHz, 13,56 MHz sekä 433 MHz ja 860 – 960 MHz, eli matala, korkea ja erittäin korkea taajuusalue [18].

Zigbee on IEEE 802.15.4-2011 standardin mukainen, low-rate wireless personal area network (LR-WPAN) -radioteknologia, joka on suunniteltu käytettäväksi sulautetuissa järjestelmissä [16]. Se toimii 2.4 GHz:n, 868- ja 915 MHz:n ISM-taajuusalueilla, se on halpa ja yksinkertainen käyttää, ja kuluttaa vähän virtaa. Se on tarkoitettu paristokäyttöisiin toteutuksiin, missä tietoa siirretään vähän, harvoin ja lähelle, esimerkkeinä langaton valokytkin ja anturiverkot. Zigbee-teknologia määritelmänsä perusteella pyrkii olemaan yksinkertaisempi ja halvempi toteuttaa kuin Bluetooth. Tiedonsiirtonopeudet ovat 20 kbps tai 250 kbps taajuusalueilla 868 MHz ja 2.4 GHz [17].

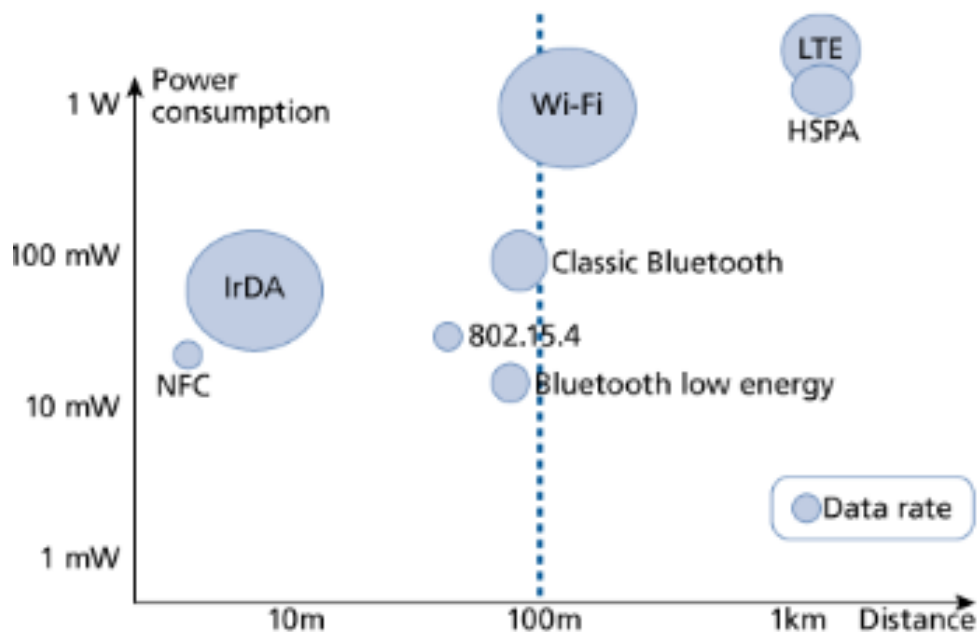
Zigbeeen etu BLE:hen verrattuna on MESH-verkko-ominaisuus. Tämä verkko pystyy konfiguroimaan itse itsensä ja uudelleenkonfiguroitumaan, jos jokin laite katoaa verkosta. Laitteet voivat välittää tietoa, luoden pitkiä verkkoja, jossa tieto hyppii laitteelta toiselle, kunnes saavuttaa määränpäänsä. Huonoa tässä ominaisuudessa on, että unitilassa olevat laitteet eivät osallistu tähän vaan nukkuvat ja verkon latenssi on korkeampi kuin point-to-point-yhteydessä. Zigbee ei myöskään ole yhteensopiva älypuhelimien tai kannettavien kanssa, koska näissä ei ole sitä tukevaa radiota.

3.2 Vertailu projektin kannalta

BLE on valmis, robusti ja globaali standardi, jolle löytyy tuki lähes kaikista älypuhelimista ja kannettavista. BLE-teknologian käyttämiseksi laitteessa ei tarvitse maksaa korkeita maksuja, eli laitteet, joissa on BLE-radio, voivat olla halpoja. Siinä on erinomainen tehonkulutuksen hyötysuhde ja yksinkertainen protokolla. Verrattuna Zigbee:hen, BLE ei voi muodostaa pitkiä MESH-verkkoja, joten maksimietäisyys on aina säde päätelaitteesta, jonka kanssa oheislaitteet keskustelevat. Toisaalta, päätelaite voi olla tietokone joka on verkossa. Zigbee ja BLE tarvitsevat molemmat reitittimen ollakseen yhteydessä internettiin [20].

RFID:tä voidaan verrata valinnassa vain sen aktiivisena versiona, siten että se toimisi lähettimenä. Mutta tietoa siitä miten, kuinka paljon ja nopeasti, sillä pystyisi dataa lähettämään ei tunnu olevan saatavilla. Vaikuttaa siltä, että se teknologiana on soveltuvin asioiden liikkuvuuden ja sijainnin seurantaan. Esimerkiksi merikonteissa tai ajoneuvoissa tietulleissa, missä on tarve vain identifioitua, eli lähettää ainutlaatuinen tunniste, joko säännöllisesti tai vastauksena. Näyttää siltä, että se ei sovellu tarkoitukseen tällä hetkellä ja, että muut teknologiat ovat ainakin helpompia ja halvempia.

Kuva 11 esittää eri radioteknologioita ja sen perusteella voi havaita, että BLE kuluttaa vähemmän energiaa kuin Zigbee, joka sijoittuisi kuvan kohtaan 802.15.4 ja sillä on mahdollista saavuttaa suurempi tiedonsiirtonopeus. Kuvassa ympyrän koko kuvaa tiedonsiirtonopeutta.



Kuva 11 Radioteknologioiden teho vs kantama vs tiedonsiirtonopeus [24]

Bluetooth Low Energy täyttää parhaiten projektin tarpeet, koska sen kantama on riittävä palvelemaan kehitettävää sovellusta, tiedonsiirtonopeus sopiva ja energiatehokkuus erinomainen. Sen avulla on mahdollista yhdistää monenlaisiin päätelaitteisiin, kuten älypuhelimiin ja kannettaviin tietokoneisiin. Tämä mahdollistaa päätelaitesovellusten kehittämisen erilaisille alustoille, joilla tietoa voidaan kerätä tai havainnoida, oheislaitteen ollessa yksinkertaisimmillaan alle 15 \$ [53] kehitysalusta. Seuraavissa alaluvuissa on tarkasteltu BLE:tä syvemmin.

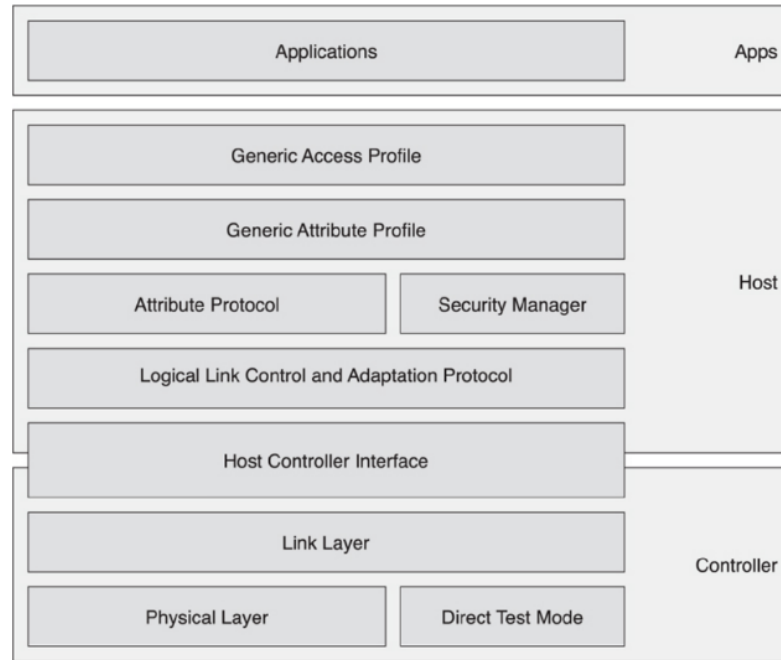
3.3 Bluetooth Low Energy

Bluetooth SIG on hallinnut kaikkea BLE:n suunnittelua raudasta ohjelmatasolle. Ensimmäiset tavoitteet ovat olleet erittäin pieni virrankulutus ja hinta. Nämä vaatimukset ovat muodostaneet kolme lähtökohtaa suunnittelulle, jotka ovat ISM-kaista, Bluetooth SIG:n kontrolloima kehitystyö ja standardoiminen ja nappiparistot. ISM-kaistan käyttämisestä ei tarvitse maksaa, tarvitsee vain noudattaa sen sääntöjä, kuten esimerkiksi lähetystehon maksimiarvoja. Bluetooth SIG -organisaation avulla hankittu patentti ja standardoiminen on halvempaa, mikä tekee valmiista BLE-tuotteista halvempia, koska organisaatiolle ei tarvitse maksaa korkeita maksuja. Kun laite suunnitellaan toimivaksi nappiparistoilla, se pakottaa laitteelle erittäin pienen virrankulutuksen, joka taas rajaa tiedonsiirtonopeuden ja käyttökohteet sellaisiksi, missä tietoa ei tarvitse siirtää jatkuvasti suuria määriä. Nappiparistot ovat myös halpoja ja helposti saatavilla. Niistä saatava maksimivirta on pieni, noin 15 mA, kokonaisenergian ollessa noin 230 mAh 3 V. Tämä maksimivirta rajoittaa radion maksimitehoja, ettei paristo ylikuormitu. Erittäin pieni virrankulutus myös pakottaa BLE:n olemaan lyhyen kantaman teknologia, pieni lähettimen- ja vastaanottimen tehonkulutus ei salli suuria tiedonsiirto etäisyyksiä [41][51].

BLE:ssä oheislaitteohjelmisto, esimerkiksi BLE-anturisovelluksessa, on tehtäviltään ja toiminnoltaan pienempi kuin päätelaitteen ohjelmisto, eli tietoa vastaanottava sovellus esimerkiksi kannettavalla tietokoneella. Näin saadaan työtä siirrettyä pois sieltä, missä tarvitaan erittäin matalaa virrankulutusta sinne, missä on enemmän resursseja käytettävänä. Oheislaitteohjelmisto on myös kooltaan pienempi ja tarvitsee vähemmän ohjelma-muistia [41].

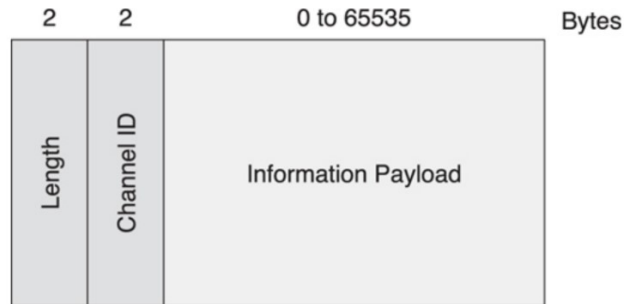
3.3.1 Arkkitehtuuri

BLE-arkkitehtuuri on jaettu kolmeen päälohkoon, sovellukset (Apps), isäntä (Host) ja ohjain (Controller). Ohjain hoitaa raudan matalimman tason toimintoja, kuten radioaaltojen avulla tapahtuvan tietoliikenteen muuntamisen digitaaliseen muotoon ja siitä edelleen paketeiksi ja toisinpäin. Ohjain kommunikoi isännän kanssa ”Host/Controller Interface” (HCI) -rajapinnan kautta. Isäntä on ohjelmistokokoelma, joka hoitaa kommunikoinnin sovelluksen ja ohjaimen välillä. Se sisältää protokollia ja profiileja, joiden avulla BLE hoitaa muun muassa yhteyksien muodostamisen, pakettien rakentamisen, lähettämisen ja vastaanottamisen sekä tiedon kapseloinnin. Nämä kaikki kolme lohkoa voivat olla yhdellä integroidulla piirillä. Vaihtoehtoisesti ne voivat olla jaettu siten, että ohjain on muista erillisellä piirillä tai vain sovellus omallaan. Ne voivat myös kaikki olla omilla piireillään. Kuva 12 esittää BLE-arkkitehtuurin lohkokaaavion.



Kuva 12 BLE-arkkitehtuurin lohkokaavio [41]

Ohjelmistokehittäjän ei sen tarkemmin tarvitse tuntea kuin tasoja: sovellus (Applications), yleinen saavutettavuus profiili (Generic Access Profile (GAP)), yleinen piirreprofiili (Generic Attribute profile (GATT)), piirreprotokolla (Attribute Protocol (ATT)) ja turvallisuushallinta (Security Manager). Mainitaan muista tässä muutamalla sanalla mitä ne ovat ja keskitytään edellä mainittuihin syvemmin seuraavissa aliluvuissa. Fyysinen taso (Physical layer), hoitaa radiota ja radioliikennettä ja sen AD-muunnosta, modulointia ja demodulointia. Linkki taso (Link Layer) kytkee HCI:n ja fyysisen tason ja hoitaa radion ajallista käyttäytymistä. Välitön testitila (Direct test mode) tarjoaa työkalut testata sulautettua BLE-järjestelmää ja kalibroida radiota ohitse yllä olevien tasojen jo valmiissa toteutuksessa. HCI-käyttöliittymästä on jo mainittu yllä, se on standardi protokolla, jonka avulla isäntä ja ohjainlohko kommunikoivat. Looginen yhteydenohjaus ja sovitusprotokolla (Logical Link Control and Adaptation Protocol (L2CAP)) vastaa protokollien kanavoinnista ja pakettien paloittelusta HCI:lle lähetettäessä tietoa ja kokoamisesta vastaanottaessa sitä. Protokollakanavia BLE:ssä on kolme: piirreprotokolla, BLE-signaali ja turvallisuushallinta. Kuva 13 esittää L2CAP-pakettia.



Kuva 13 L2CAP-paketti [41]

Paketti sisältää pituutensa lisäksi protokollakanavan tunnusteen ja dataa, jonka pituuden määrittää maksimisiirtoyksikkö (Maximum Transmission Unit (MTU)), tämä pituus on myös piirreprotokollan paketin maksimipituus. MTU on vakiona 23 tavua ja kaikki BLE-laitteet tukevat vähintään sen verran [41].

3.3.2 Yleinen saavutettavuusprofiili GAP

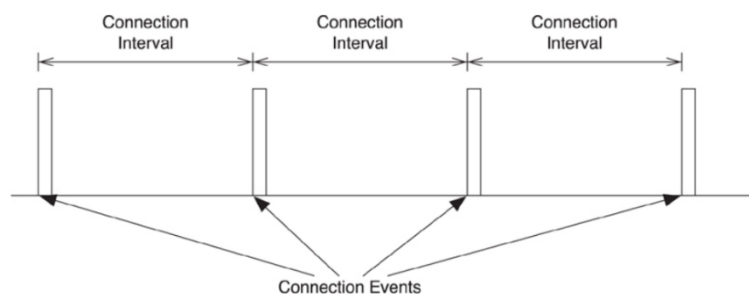
Tässä profiilissa määritellään, kuinka BLE-laitteiden välinen kanssakäyminen tapahtuu. Kanssakäymiseen kuuluu osa-alueita, kuten laitteiden roolit, itsensä mainostaminen lähettämällä radiopaketteja, yhteyden muodostaminen ja pariutuminen, ja yhteyden turvallisuusvaatimukset ja -asetukset.

Rooleja laitteilla voi olla: lähettäjä tai kuuntelija ilman yhteyttä ja pääte- tai oheislaitte. Lähettäjä lähettää jotain tietoa, esimerkiksi sijaintitietoa ja siihen ei voi muodostaa yhteyttä, sen ei tarvitse edes sisältää radiovastaanotinta. Kuuntelija voi olla laite, jossa on vain radiovastaanotin, mutta se voi olla myös täydellinen BLE-laitteisto kuten älypuhelin. Se kuuntelee lähettimen lähettämää tietoa ja suorittaa jotain sovellusta, esimerkiksi piirtää kuvaajaa älypuhelimien näytölle. Oheislaitte mainostaa itseänsä paketeilla missä se ilmaisee, että siihen on mahdollista muodostaa yhteys ja päätelaite kuuntelee näitä paketteja ja aloittaa yhteyden muodostamisen. Päätelaite on aina yhteyden muodostamisen aloittava osapuoli. BLE-laitteilla kuten älypuhelin, voi olla samanaikaisesti useita rooleja, eri laitteiden kanssa.

Oheislaitteet mainostavat itseänsä kaikille, jotka ovat kuulolla. Ne kertovat mainospaketissa onko niihin mahdollista muodostaa yhteys. Dataa mainospaketissa voi olla vain 31 tavua, joka voi olla esimerkiksi laitteen nimi tai muu tunniste. Päätelaite voi lähettää "Scan Request" -paketin lisätiedon pyytämiseen oheislaitteelta. Jos oheislaitteelta löytyy pyydetty tieto, lähettää se sen "Scan Response" -paketissa. Mainostaminen tapahtuu tasa-aikavälein kolmella kanavalla, jotka ovat määritelty vain tätä varten BLE-arkkitehtuurissa. Kun päätelaitesovellus on valmis muodostamaan yhteyden oheislaitteeseen, se alkaa kuuntelemaan näitä kolmea kanavaa, löytääkseen sopivia oheislaitteita.

Yhteyden muodostaminen alkaa, kun päätelaite lähettää yhteydenpyyntö-paketin oheislaitteelle, samalla kanavalla missä se on kuullut oheislaitteen mainoksen. Näin kaksi laitetta kuulevat toisensa. Oheislaitte kuuntelee kanavaa hetken mainoksen lähetyksen jälkeen, jotta se kuulee mahdollisen yhteydenmuodostuspyynnön. Kun oheislaitte kuulee tämän pyynnön, yhteys on luotu, mutta ei vahvistettu. Yhteys on vahvistettu vasta, kun päätelaite on vastaanottanut ensimmäisen datapaketin, päätelaite siis vahvistaa yhteyden. Tässä vaiheessa päätelaitetta kutsutaan nimellä isäntä (Master) ja oheislaitetta nimellä renki (Slave). Isäntä hoitaa aina yhteyden eri parametrien määrittelyn, mutta renki voi ehdottaa jotain parametreja. On isännän päätettävissä, kuunnellaanko näitä ehdotuksia.

Kun laitteet ovat yhteydessä toisiinsa, isännän on lähetettävä paketti jokaisen yhteystapahtuman (Connection Event) aikana. Yhteystapahtuma kestää yhteystapahtuman aikavälin (Connection Interval) verran. Tänä aikana isäntä ja renki voivat vaihtaa tietoa ja tämä tapahtuu aina samalla radiotaajuudella. Aikaväli on 1,25 ms kerrannainen välillä 7,5 ms – 4 s. Jos varsinaista viestintätarvetta ei ole, isäntä lähettää tyhjän paketin rengille ja renki vastaa siihen. Jos renki ei vastaa, isäntä lähettää seuraavan paketin seuraavalla yhteystapahtuman aikahetkellä. Jos renki ei vastaa ollenkaan yhteyden ylläpidon aikakatkaisu ajan kuluessa (Supervision Timeout), isäntä katkaisee yhteyden. Yhteystapahtuman aikaväli sovitaan yhteyden muodostuksen yhteydessä ja renki voi esittää tähän toiveen, isäntä kuitenkin päättää aina lopullisen aikavälin. Kuva 14 havainnollistaa tätä toiminnallisuutta.



Kuva 14 Yhteystapahtuma ja aikaväli [41]

Oheislaitteen renginvastelatenssi (Slave latency) -asetuksella määritellään kuinka monta peräkkäistä yhteystapahtumaa renki voi jättää huomioimatta. Asetus on positiivinen kokonaisluku, jossa nolla tarkoittaa jokaiseen tapahtumaan vastaamista. Asetus tulee määritellä siten että yhteyden ylläpidon aikakatkaisun aikaväli ei täyty. Esimerkiksi jos latenssi on 2, yllä olevassa kuvassa renki vastaa yhteystapahtumaan vain ensimmäisessä ja viimeisessä tapahtumassa. Jos "Connection interval" -aikaväli kuvassa olisi 7,5 ms, aikakatkaisun täytyy olla pidempi kuin 22,5 ms, että renki ehtii vastaamaan [41][51].

Yhteyden turvallisuusvaatimukset ja -asetukset määrittävät kaksi turvallisuustilaa ja muutamia näiden tasoja. Jokainen yhteys alkaa tilasta yksi ja tasolta yksi, joka käytännössä ei takaa mitään turvallisuutta. Muuttaakseen tätä tasoa laitteiden on muodostettava yhteys. Tila yksi määrittelee erilaisia salauksen tasoja yhteydessä ja tila kaksi erilaisia tasoja tämän päälle, jossa määritellään tiedon allekirjoittamista. Kuva 15 esittää näitä tiloja ja niiden tasoja.

	Unencrypted Link and Unauthenticated Pairing	Unencrypted Link and Authenticated Pairing	Encrypted by Using Unauthenticated Pairing	Encrypted by Using Authenticated Pairing
Security Mode 1 Level 1	Send data	Send data	Send data	Send data
Security Mode 1 Level 2	Encrypt then send data	Encrypt then send data	Send data	Send data
Security Mode 1 Level 3	Not possible—must repair by using an authenticated pairing	Encrypt then send data	Not possible—must repair by using an authenticated pairing	Send data
Security Mode 2 Level 1	Sign data using unauthenticated key	Sign data by using authenticated key	Send data	Send data
Security Mode 2 Level 2	Not possible—must repair by using an authenticated pairing	Sign data by using authenticated key	Not possible—must repair by using an authenticated pairing	Send data

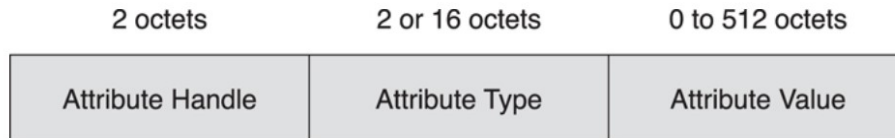
Kuva 15 GAP-turvallisuus tiloja ja tasoja [41]

Renki voi esimerkiksi vaatia tietoon käsiksi pääsemiseksi isäntää tunnistautumaan. Tunnistautuminen tapahtuu käyttäen digitaalista allekirjoitusta ”connection Signature Resolving key” -avaimen avulla. Tämä sijoitetaan BLE-pakettiin datan jälkeen.

Salaukseen käytetään 128-bittistä kehittyntä salausstandardia. Tunnistautumisen onnistuessa yhteys salataan ”Short Term Key” -avaimen avulla ja toinen avain jaetaan tämän suojatun yhteyden avulla. Avaimen nimi on ”Long Term Key” ja sitä käytetään jatkossa salauksen purkuun [52].

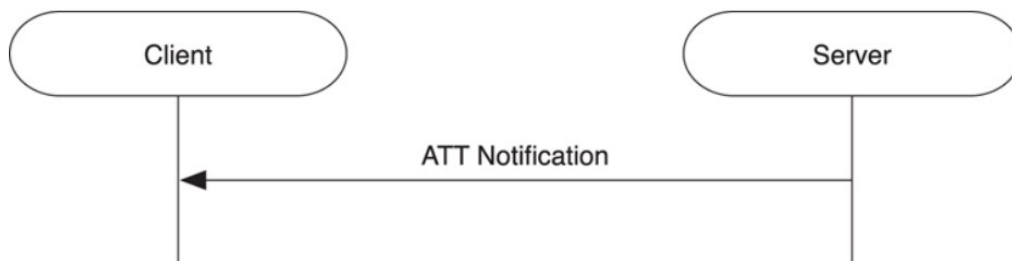
3.3.3 Piirreprotokolla ATT

Piirre tarkoittaa dataa, jolla on jokin nimi (handle), tyyppi (type) ja arvo (value). Protokolla tarkoittaa tässä yhteydessä standardoitua tapaa, jolla isäntä pääsee käsiksi rengin tarjoamiin piirteisiin. Kuva 16 esittää piirrettä.



Kuva 16 BLE-piirre [41]

Protokolla tarjoaa tähän kuusi viestiä: pyyntö (request), vastaus (response), käsky (command), esitys (indication), varmistus (confirmation) ja ilmoitus (notification). Isäntä voi siis lähettää rengille käskyn, mihin se ei odota vastausta, tai pyynnön, johon se odottaa rengiltä vastauksen tai virhekoodin. Renki voi lähettää isännälle esimerkiksi ilmoituksen, johon se ei odota vastausta. Kuva 17 havainnollistaa tätä tilannetta. Tai sitten esityksen, johon se odottaa isännän vastaavan. Käskyviesteistä esimerkkeinä voidaan mainita kirjoituskäsky (Write Command), jolla voi kirjoittaa piirteen arvoon ja joka on mahdollista lähettää milloin vain, eikä siihen tarvitse vastata. Ilmoitus (Notify) -viestillä oheislaitte voi lähettää piirteiden tietoa, omasta aloitteestaan, eikä sekään vaadi vastausta. Nämä eroavat esimerkiksi luku pyynnöstä (Read Request) siten, että siihen tarvitsee vastata ja vastauksen on mahdollista yhteen tapahtumaan. Seuraava "Read Request" voidaan lähettää vasta kun ensimmäiseen on vastattu ja jos tietoa halutaan paljon, täytyy näitä tapahtumia toteuttaa jotenkin organisoidusti.



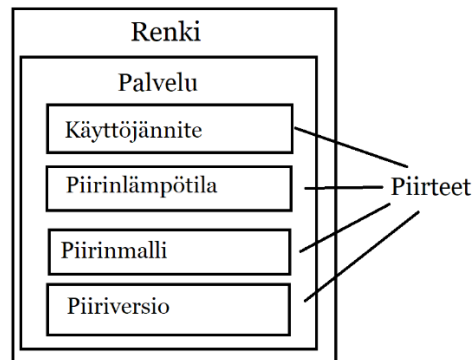
Kuva 17 BLE-rengin ilmoituspaketti [41]

Piirreprotokolla sisältää myös MTU:n pituuden neuvotteluun tarvittavan prosessin. Vain isäntä voi aloittaa tämän neuvottelun ja pituudeksi määräytyy aina osapuolien lyhyempi MTU. Isäntä sisällyttää MTU:n pituuden pyyntöön oman vastaanottimensa pituuden ja renki tekee vastatessaan samoin. MTU:n pituus on aina sama, eikä vaihtelee, vaikka laite toteuttaisikin molempia rooleja [41].

3.3.4 Yleinen piirreprofiili GATT

Profiili kertoo mitä laitteelta voi odottaa ja mitä se pystyy tekemään. Profiilit sisältävät palveluja (services), jotka ovat kokoelma piirteitä (characteristics). Palvelu muodostaa

piirteistä tarjoamansa kokonaisuuden, esimerkiksi kuvitteellisen mikrokontrollerin tilapalvelun, joka voisi sisältää piirteet käyttöjännitteen- ja piirinlämpötilanarvosta, valmistajan nimen, tuotteen mallin ja piirin valmistusversion. Kuva 18 esittää tätä kokonaisuutta.



Kuva 18 Rengin toteuttama palvelu

Piirteet sisältävät jonkin tiedon tai datan ja eri palvelut voivat käyttää samoja piirteitä muodostaakseen uusia palveluja. Piirteenkuvaukset (characteristic descriptions) on vapaaehtoinen tarkennus piirteiden sisällöstä, piirre on siis dataa ja piirteenkuvaukset on sen yksikkö, esimerkiksi °C. Piirteenkuvauksista ei ole pakko olla tai niitä voi olla useita. Oheislaitte voi olla suunniteltu jollekin toiselle profiilille ja toteuttaa useampia palveluita kuin profiili päätelaitteella, jonka kanssa yhteys on muodostettu. Päätelaitesovelluksen toteuttama profiili kuitenkin toimii, jos oheislaitteelta löytyy sen tarvitsemat palvelut, vaikka se ei toteuttaisikaan samaa profiilia. Palvelu määrittelee itsensä joko 16- tai 128-bittisellä Universally Unique Identifier (UUID) -numerolla, jonka avulla sen voi tunnistaa (service declaration), ja piirteillä on myös oma UUID-numero. UUID-numero muodostetaan BLE-standardissa määritellyn "Bluetooth Base UUID" -numeron päälle.

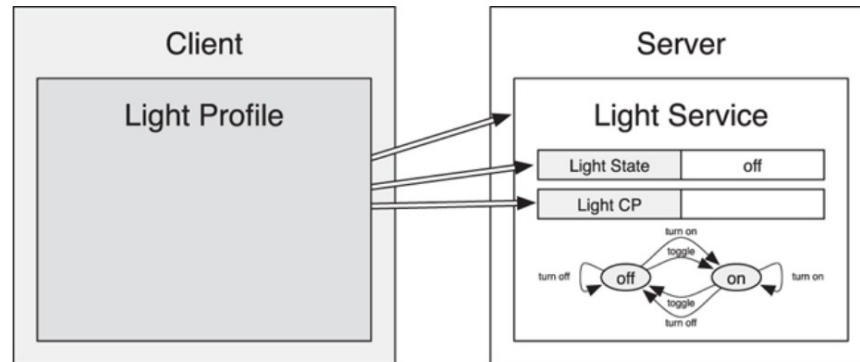
Piirteille on määritetty mitä niille voi tehdä. Esimerkiksi jokin piirre voi olla vain luettavissa (Readable), toista piirrettä voi lukea ja kirjoittaa (Readable and Writable) ja jotain kolmatta vain kirjoittaa. Piirteiden luku voi vaatia todistuksen aitoudesta, jossa päätelaite todistaa olevansa aito. Piirteillä voi olla myös lupa ominaisuus, jolla oheislaitte voi määrittää, onko päätelaitteella lupa lukea sen tietoa.

Myös tässä profiilissa laitteilla täytyy olla joko isännän tai rengin rooli. Renki omistaa tai tuottaa tietoa, mistä isäntä on kiinnostunut ja mihin se yrittää päästä käsiksi. Jos laitteella on monta roolia, rooli vaihtelee sen mukaan luovuttaako se tietoa vai ottaako se sitä vastaan [41][51].

3.3.5 Sovellukset

Sovellukset rakentavat toimintansa GATT:n ominaisuuksien, kuten piirteiden, palveluiden ja profiilien päälle. Ne hoitavat sitä mitä vastaanotetulla tiedolla tehdään ja miten

uutta tietoa tehdään. Älypuhelin voi esimerkiksi tulostaa tietoja näytölle. Lähettävän laitteen sovellus voi tehdä analogisia mittauksia tai käyttää jotain digitaalista kommunikointiväylää ulkoisen oheislaitelohkon kanssa, kuten esimerkiksi kosteus- ja lämpötila-anturin. Kuva 19 esittää sovelluksen hoitamaa tehtävää kytkeä lediä päälle ja pois.



Kuva 19 BLE-sovelluksen hoitama tehtävä [41]

Päätelaitteen sovelluksen tulee pystyä käsittelemään useita itseään mainostavia oheislaitteita ja sitä mihin niistä se yrittää luoda yhteyden. Isäntäsovelluksella voi olla useita renkiä, joihin se on yhteydessä samanaikaisesti ja kommunikoi näiden kanssa vuorotellen. Päätelaitesovellus voi olla myös sellainen, että se kuuntelee tietoa lähettävää oheislaitetta, johon ei voi muodostaa yhteyttä, ns. ”broadcaster”. Sovelluksen on tunnistettava kaikesta BLE-radioliikenteestä, että tämä on tietoa mikä sitä kiinnostaa. Toiminnallisuus laitteiden nimien tulostamisesta ihmisen luettavissa olevassa muodossa, tai vain sovelluksen kanssa yhteensopivien laitteiden, on sovelluksen vastuulla.

Samalla tavalla, jos oheislaitte halutaan yhdistävän vain johonkin ennalta turvalliseksi määriteltyyn päätelaitteeseen, se on sovelluksen tehtävä. Oheislaitteessa voi olla painonappeja, joilla saadaan yhteys lopetetuksi, mainostaminen uudelleen aloitetuksi tai jokin muu toiminto aikaiseksi ja näiden toimintojen hoitaminen on myös sovelluksen tehtävä.

Sovellusten tehtäviksi molemmissa rooleissa jää myös yhteyksien muodostamisen ja pariutumisen eri tapojen toteuttaminen. Esimerkiksi, jos oheislaitte mainostaa itseään vain silloin, kun sillä on jotain uutta tietoa, laitteet muodostavat nopean yhteyden, vaihtavat tietoa ja katkaisevat yhteyden. Oheislaitte palaa takaisin unitilaan ja päätelaitte odottaa uutta mainosta oheislaitteelta [41].

4. PROJEKTIN KUVAUS

4.1 ENOMA

ENOMA-projektin lähtökohta oli tutkia ja kehittää ratkaisuja energian keräämiseen ympäristöstä ja sen muuttamiseen sähköksi ja varastoitumiseen pienelektroniikan käyttöön. Projektia ovat rahoittaneet useat yritykset ja Business Finland (määräraha 6698/31/2018). Siinä oli mukana koordinoiva yritys, 15 muuta yritystä ja 3 tutkimusorganisaatiota, kuten muun muassa VTT, Oulun- ja Tampereen yliopistot, Abloy, Cargotec ja Nokian Renkaat Oy. Tavoitteina oli kehittää uusia innovaatioita, tuotteita ja ominaisuuksia [54].

Motivaatio ratkaisujen kehittämiseen tulee esimerkiksi asennus ja ylläpitokulujen pienentymisestä, kun laitteet eivät vaadi akkuja tai sähkökaapelointia toimiakseen. Tämä lisää laitteiden sijoitusmahdollisuuksia paikkoihin, minne on hankala johdottaa antureita, ja vähentää ylläpitoon käytettävää työtä, kun paristoja ja akkuja ei tarvitse käydä vaihtamassa. Ratkaisuilla voidaan pienentää ympäristövaikutuksia, käyttöönoton kustannuksia ja parantaa käyttöönoton joustavuutta, kun johdottamisesta voidaan luopua.

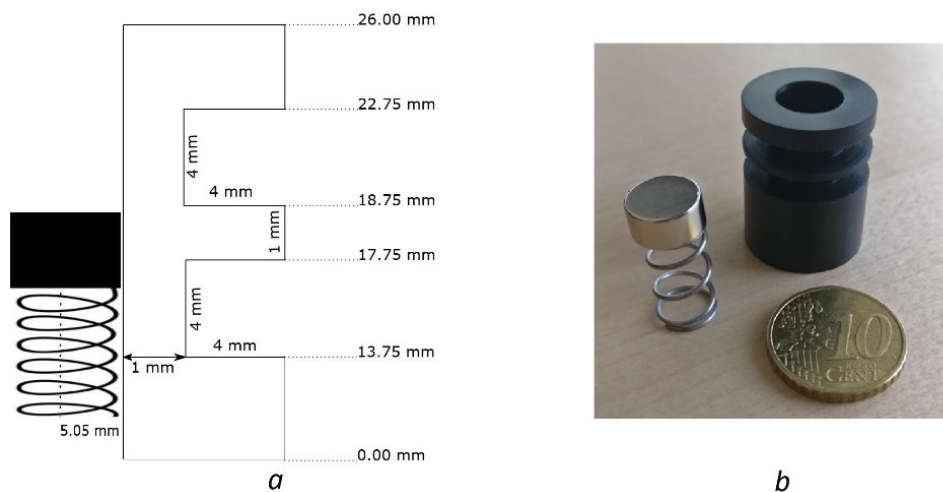
Energiankeräimiä voidaan käyttää paikoissa, joissa on soveltuvaa lähde-energiaa saatavilla. Koneet voivat tuottaa hukkalämpöä, tärinää tai muuten voimakkaita liikkeitä kuten iskuja. Ne sopivat sovelluksiin, missä ei ole tarvetta siirtää suuria määriä tietoa. Tämä sallii laitteiden olla pois päältä pitkiä aikoja tiedonsiirtojen välillä. Esimerkiksi sähkömoottorin laakeroinnin kuntoa analysoidaan kerran minuutissa, 500 ms ajan. Toinen esimerkki voisi olla, että suuren maansiirtokoneen rengaspaineita tarkkaillaan niiden eliniän maksimoiseksi. Tällaisia koneita voi olla useita ja paristojen vaihtaminen säännöllisesti on työlästä.

Tampereen yliopiston tehtäväksi projektissa on tullut kehittää induktiivista energiankeräintä. Tarkoitus on myös tehdä kerättyä energiaa hyväksikäyttävä mahdollisimman pienellä energian kulutuksella toimiva IoT-laitteisto tiedonsiirtoa varten. Tämä siksi, että tällaista energiaomavaraista laitteistoa voitaisiin testata jossakin reaali maailman toteutuksessa. Tavoitteena testilaitteistolle oli, että joltain anturilta mitattu tieto saadaan siirrettyä langattomasti noin kymmenen metrin päähän, esimerkiksi jonkin huoneen tai tilan sisällä.

4.2 Prototyypijärjestelmä

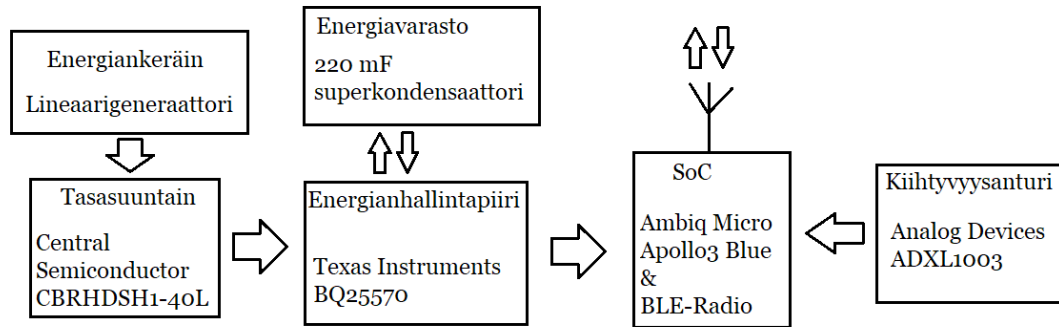
Tampereen yliopistossa on kehitetty prototyypijärjestelmää Nokian Renkaat Oy:n kanssa. Järjestelmä sisältää energiankeräimen, -talteenoton, kiihtyvyyssanturin ja tiedonsiirtolaitteiston eli lähettimen. järjestelmän testaamiseksi on kehitetty myös vastaanotin. Järjestelmä tullaan asentamaan pakettiauton renkaan sisälle ja optimoimaan 80 km/h nopeudelle. Järjestelmä lähettää renkaan sisältä kiihtyvyyssanturilta mitattua tietoa langattomasti vastaanottimelle. Vastaanotettua mittaustietoa tullaan käyttämään energiankeräimen toiminnan soveltuvuutta ja laatua analysoitaessa.

Tampereen yliopistossa on valittu sähkömagneettista induktiota käyttävä energiankeräin koska se on pitkälle tutkittu, yksinkertainen rakentaa eikä vaadi kalliita tai monimutkaisia rakenteita eikä materiaaleja. Sillä on helppo kerätä energiaa tärinästä, jota on reilusti saatavilla käyttöympäristössä. Eri induktiivisista keräintyypeistä on keskitytty kehittämään lineaarista sähkögeneraattoria. Kuva 20 esittelee kehitettyä energiankeräintä.



Kuva 20 Tampereen yliopistossa kehitetty energiankeräin

Tampereen yliopistossa tehdään myös mallinnusta ja simulaatiota lineaarisen sähkögeneraattorin energian tuotosta ja kuvan 8 mukainen energiankeräin on rakennettu näiden mallinnusten tulosten perusteella. Energiankeräimen tehontuottoa on simuloitu Matlab-ohjelmiston päälle tehdyllä työkalulla ennen valmistusprosessia. Tavoitteena oli minimoida energiankeräinlaitteen koko ja paino, siten että keskimääräinen minimituotto on 1 mW. Laitetta simuloitaessa käyttöä vastaavilla parametreilla todettiin, että keskimääräinen tuotto noin 50 mW. Kuva 21 esittää Tampereen yliopistossa kehitetyn energiamavaraisten prototyypijärjestelmän arkkitehtuuria.



Kuva 21 Tampereen yliopistossa kehitetty energiaomavarainenlaitteisto

Energiankeräin on yhdistetty energianhallinta ja -varastointipiiriin, joka on tehty käyttäen Texas Instruments BQ25570 piiriä. Koska energiankeräin tuottaa vaihtomuotoista ulostulojännitettä on, näiden välissä on Central Semiconductor -valmistajan Schottky-diodeista valmistettu tasasuuntainpiiri, mallia CBRHDSH1-40L. Energianhallintapiiri voidaan konfiguroida erilliskomponenteilla toimimaan eri energiankeräinteknologioiden kanssa, mutta tässä yhteydessä se on asetettu käyttämään induktiivista keräintä. Piiri kerää energiaa 220 millifaradin superkondensaattoriin 5,5 voltin jännitteeseen ja laskee sen 3,3 volttiin kuormana olevan mikrokontrollerijärjestelmän käyttöön sopivaksi [45][46].

Järjestelmän keräämää energiaa käyttää käyttöenergianaan siihen liitetty mikrokontrollerijärjestelmä. Se koostuu Apollo3 Blue -järjestelmäpiiristä ja Analog devices ADXL1003 -kiihtyvyyssanturista. Apollo3 Blue -järjestelmäpiiri sisältää myös piirille integroidun Bluetooth Low Energy -järjestelmän. Tästä mikrokontrollerijärjestelmästä puhutaan kattavammin seuraavissa luvuissa [31][44].

5. MIKROKONTROLLERIJÄRJESTELMÄ

Projektissa on kehitetty mikrokontrollerijärjestelmää, jonka tulisi toimia mahdollisimman pienellä energiamäärällä ja pystyä kommunikoimaan langattomasti. Järjestelmä, sen vaatimukset ja valintaperusteet, ja kehitysympäristö on esitelty seuraavissa aliluvuissa.

5.1 FreeRTOS

Ohjelmisto käyttää FreeRTOS-versiota 10.1.1. Tämä on reaaliaikakäyttöjärjestelmäydin (RTOS-kernel) mikrokontrollereille ja se hoitaa järjestelmässä ajanhallintaa, kuten uni- ja lepotiloja ja sovellusohjelmien ajoa. Sitä on kehitetty yli 15 vuotta yhteistyössä piirivalmistajien kanssa ja se on markkinajohtaja RTOS-ydin mikrokontrollereille ja pienprosessoreille. Sen tekee hyväksi helppokäyttöisyys, pieni koko, saatavuus ja luotettavuus, ja se on ilmainen MIT-lisenssin alaisuudessa [9].

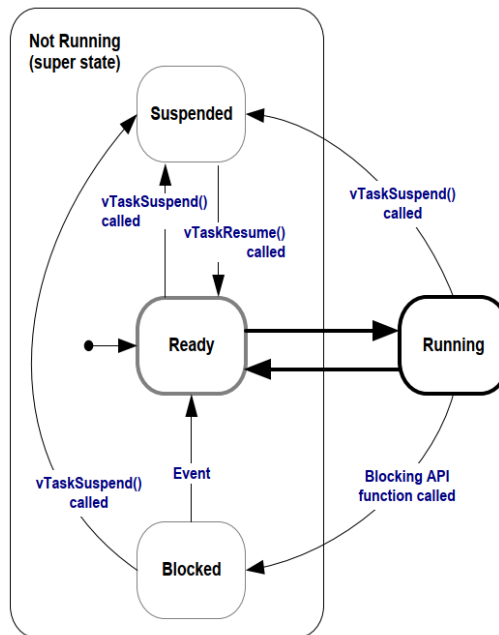
RTOS-ytimen käyttö on aina sovelluskohtaista. Yksinkertaisille ohjelmistoille voi olla parempi ratkaisu olla käyttämättä RTOS-ydintä. Isommissa ohjelmistoissa sen hyöty on selvä. Se mikä on iso- ja mikä pieniohjelmisto, on kehittäjän itse päätettävä.

RTOS-ydin on vastuussa ohjelmiston ajamisen ajallisesta toiminnasta. Se siis abstrahoi ajastukseen liittyvää toimintaa, kuten milloin asioita suoritetaan. Tämä lisää ohjelmiston yhteensopivuutta eri alustoille. Mikrokontrollerin ajastimet ja niiden käytön hoitaa ydin. Ydin myös pitää huolen kaikista tapahtumista, sovellusohjelmat voivat vain jäädä odottamaan jotain tapahtumaa, ilmoittaa ytimelle olevansa odottavassa tilassa ja ydin voi hoitaa muita tehtäviä. Se että tapahtumia ei tarvitse jäädä odottamaan silmukassa estäen kaiken muun toiminnan, säästää resursseja. Mutta tehtävät, mitä ydin hoitaa, myös kuluttaa aikaa eikä tule ilmaiseksi [27].

Sulautetun järjestelmän ohjelmisto sisältää usein molempia, kovia- ja pehmeitä vasteaikavaatimuksia. Kova vasteaikavaatimus on, että järjestelmä ei enää toimi, jos vaatimusta rikotaan. Tällainen voisi olla Bluetooth radion toimintaan liittyvät tehtävät. Jos jotain tehtävää ei suoriteta ajoissa, voidaan yhteys menettää. Pehmeä vasteaikavaatimus taas on, että sen rikkominen ei vaikuta järjestelmään, mutta aiheuttaa toleranssia järjestelmään. Jos kiihtyvyyssanturilta mitataan arvo kerran sekunnissa ja jos aikaväli heiluu yhden tai kymmenen millisekuntia, se ei välttämättä haittaa mittausta. Toleranssin olemassaolo täytyy vain tiedostaa [27].

Ydin hoitaa, että sovellusohjelmat saavat prosessorilta aikaa. Siksi sitä voisikin kutsua reaaliaikavuorontajaksi (real-time scheduler). Sovellusohjelma on yksi säie (thread) tai

erityisesti FreeRTOS:n kohdalla puhuttaessa osatehtävä (task). Mikrokontrollereissa on usein vain yksi prosessori, joten vain yhtä osatehtävää voi suorittaa kerrallaan. Tehtävällä on muutama tila, missä se voi olla. Se voi olla ajossa, eli suorituksessa, tai se voi olla odottamassa ajoa. Tässä odottavassa tilassa on kolme osatilaa. Odotetaan vuoroa prosessorilta valmiina suorittamaan tehtävää, odotetaan jotain tapahtumaa, tai tehtävä on pysäytetty. Kuva 22 havainnollistaa näitä eri tiloja.



Kuva 22 FreeRTOS-osatehtävän tilat [24]

Osatehtävät suorittavat jotakin selkeää ja rajattua tehtävää, esimerkiksi AD-muunnosta, ulkoiseen muistiin tallennusta tai kommunikointia jonkin oheislaitteen kanssa. Sovellukset koostuvat osatehtävistä, joita RTOS vuorontaa. Ne ovat modulaarisia, joten niiden kehitystyö voidaan helposti jakaa eri henkilöille ja niitä on helppo lisätä valmiisiin ohjelmiin. Ne on myös helppo testata ytimen päällä ajettuna, ilman muita sovellusohjelmia [27].

Osatehtävillä on jokin prioriteetti, toisilla korkeampi kuin toisilla. Nämä pystyvät keskeyttämään alemman prioriteetin osatehtävän suorituksen, tai menemään suoritukseen ensimmäisenä odottavasta tilasta, ohitse muiden. Prioriteettiasetuksilla säädetään järjestelmän vasteaika-vaatimuksia ja tähän FreeRTOS tarjoaa hyvät ja monipuoliset työkalut. Näistä asetuksista tulee helposti monimutkaisia ja näihin syvemmin perehtyminen on tämän työn ulkopuolella [27].

Reaaliaikakäyttöjärjestelmä ydin mittaa aikaa käyttämällä aika-askelkeskeytyksiä. Tämä on ytimen aikaresoluutio. Jos askel on 10 millisekuntia, niin osatehtävät voivat ajastaa

itseään vain tätä huonommilla tarkkuuksilla, esimerkiksi tehtävä ei voi nukkua 5 millisekuntia. Kun tämä keskeytys tapahtuu ydin tarkastaa, onko jokin tehtävä valmiina suoritukseen. Aikaa kuluu siis hukkaan aina, kun aika-askelkeskeytystä suoritetaan. FreeRTOS-ydin tukee myös tilaa nimeltä 'tickless idle mode', eli aika-askelkeskeytyksetön toimeton tila, tässä tilassa, kun kaikki osatehtävät ovat toimeentona, ydin voi mennä syvä uneen aina "Idle-task" -osatehtävään mentäessä. Tämä on RTOS:n sisäinen osatehtävä ja se on ajossa aina, kun mitään muuta ei ole suoritettavana. Se ei kuitenkaan tee mitään ja mahdollistaa syväuneen menemisen. Normaalisti tästä unesta herättäisiin askeltaamaan kelloa aina joka askeleella. Tässä tilassa mikrokontrolleri voi olla kaikista syvimässä unessa maksimian, ilman väliheräämisiä ja korjata ajan herätessään. Tämä korjaus tehdään siten, että FreeRTOS-ydintä konfiguroitaessa asetetaan maksimiuniat eri osille järjestelmää. Tämä maksimiuniaika on siis aina määritelty ja kontrolleri herää viimeistään tässä ajassa ja tarvittaessa palaa takaisin uneen. Aika on kuitenkin hyvin paljon pitempi kuin järjestelmässä, joka ei käytä aika-askelkeskeytyksetöntä toimetonta tilaa [27][28].

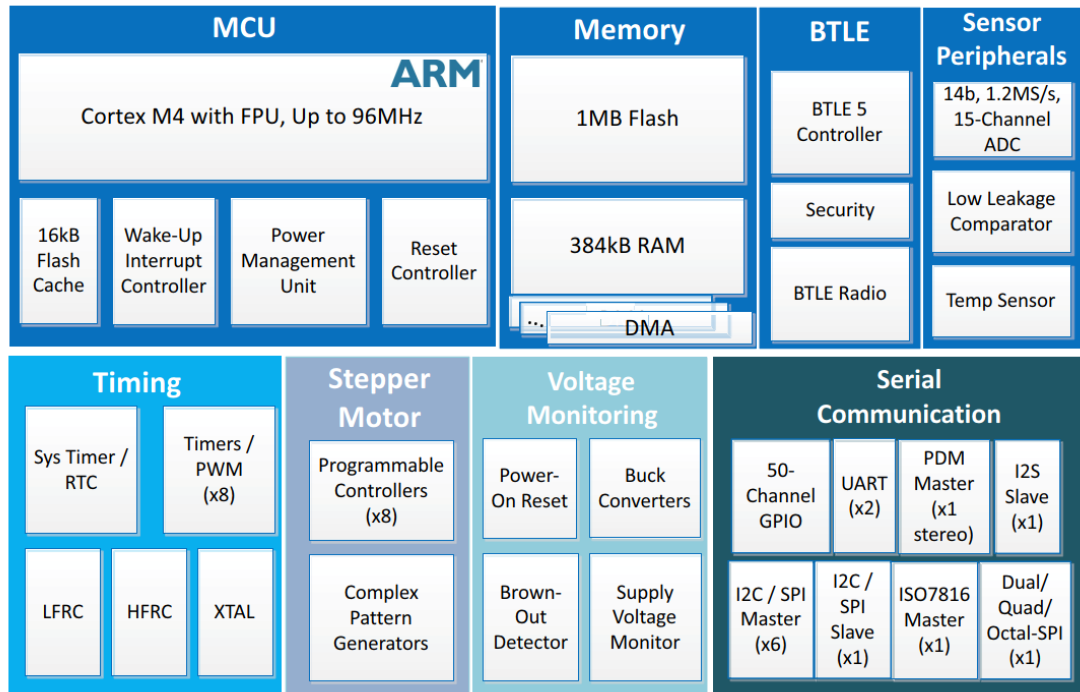
Muutamia muita mikrokontrolleri-RTOS-ytimiä on Microsoft Azure RTOS ja Mbed OS ARM Cortex-M kontrollereille. Etuna näissä on se, että Azure RTOS on yhteensopiva Azure IoT-pilvipalvelun kanssa ja siinä on tuki TCP/IP protokollille ja tiedostojärjestelmille. Mbed OS taas on yhteensopiva Mbed Studio desktop IDE:n kanssa ja sille on laaja käyttäjäyhteisö, mistä voi saada tukea [29].

5.2 Ambiq Micro Apollo3 blue

Tämä langaton mikrokontrollerijärjestelmäpiiri sisältää integroidun BLE-radion ja -kontrollerin. Piirille luvataan erittäin pientä virrankulutusta, $6 \mu\text{A} / \text{Mhz}$, ja syväunitilaa, jossa saavutetaan $1 \mu\text{A}$:n minimivirrankulutus, ja josta voidaan herätä reaaliaikakellokeskeytyksellä. Prosessorina piirillä on tehokas ARM Cortex-M4F, 48 MHz:n kellolla, joka voidaan kiihdyttää väliaikaisesti 96 MHz:iin. Kontrolleri toimii 1,755 – 3,63 voltin jännitteellä ja sen suunnittelun lähtökohtana ovat olleet paristokäyttöiset laitteet. Siinä on kaikkien perusoheislaitelohkojen, kuten sarjaliikenne ja Analog-to-Digital-Converter (ADC), lisäksi muun muassa integroitu Direct-Memory-Access (DMA) -ohjain. Kuva 23 esitetään järjestelmäpiirin lohkokaavio [31].

Piirin pieni virrankulutus on saavutettu Ambiq Micro:n patentoimalla Sub-Threshold Power Optimized Technology (SPOT™) arkkitehtuurilla, joka mahdollistaa erittäin hyvän tehohyötysuhteen. Piiri valmistaja kertoo vuonna 2019 saavuttaneensa Apollo3 blue -pii-

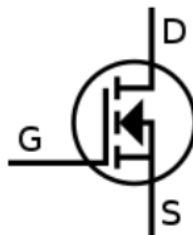
rillä maailman parhaimman energiankulutuksen hyötysuhteen. Piirin on valmistanut Taiwanilainen TSMC yritys, käyttäen heidän erittäin pienitehoisten laitteiden 40 nanometrin piirivalmistusprosessiaan (40ULP) [32].



Kuva 23 Apollo3 blue -lohkokaavio [31]

SPOT™ teknologialla valmistettu Apollo-sarja on teollisuuden ensimmäinen langaton järjestelmäpiiri, joka käyttää toiminnassaan hyödyksi transistorin alle kynnyksjännitteen toiminta-alueita.

Normaalisti perinteinen CMOS-kanavatransistori tarvitsee 1,8 voltin kynnyksjännitteen, eli jännite-eron hilan (gate) ja lähteen (source) välille, jotta virta kulkee nielusta (drain) lähteeseen. Transistori on tällöin loogisessa 1-tilassa. Alle kynnyksjännitetilassa hilan ja lähteen välille tarvitsee tuottaa vain 0,3 voltin kynnyksjännite, että se on 1-tilassa. Kuva 24 esittää CMOS-kanavatransistoria, jossa D on nielu, G on hila ja S on lähde.



Kuva 24 N-MOSFET [34]

Vaikka transistoria ei ajettaisi hilajännitteellä loogiseen 1-tilaan asti, virtaa kulkee silti nielusta lähteeseen ja alle kynnysjännitetilassa tästä virrasta voidaan päätellä looginen 1-tila. Kynnysjännitteeseen hukkuva teho on jännitteen neliö ja tämän takia jännitteen alentua saavutetaan tehohyötysuhteen paraneminen. 0,3 voltin kynnysjännitteen toimintatilassa voidaan säästää 36 kertainen teho verrattuna 1,8 volttiin. Kuitenkaan koko järjestelmäpiiri ei käytä tätä tilaa, vaan se on optimoitu niin, että suurin hyöty saadaan aikaiseksi. Jotkin osat piirissä toimivat normaalisti eivätkä edes tarvitse erityistä optimointia [35].

TSMC:n 40ULP on piirivalmistusteknologia, jolla se on saavuttanut 70 % vähennyksen vuotovirroissa ja pienentänyt tehonkulutusta 30% aiempiin 40 nanometrin teknologioihin verrattuna. Teknologialla pystytään valmistamaan piirejä erittäin pienivuotovirtaisilla transistoreilla ja staattinen-hajasaantimuisti (SRAM) -soluilla. Teknologia tukee 0,7 voltin käyttöjännitteen standardisoluja ja soveltuu esimerkiksi IoT-laitteille [36][37].

Vertailukohtana Apollo3:lle on tarkasteltu Maxim Integrated -piirivalmistajan piiriä MAX32668, joka kuuluu DARWIN-sarjaan. Tässä on integroitu BLE-radio omalla mikrokontrollerillaan, ARM Cortex-M4F -prosessori 96 MHz maksimitaajuudella ja samanlaiset oheislaitelohkot kuin Apollo3:ssa. Kontrolleri on mahdollista saada myös kahdella Cortex-M4F -prosessorilla. Valmistaja mainostaa piirin kuluttavan erittäin vähän virtaa ja sopivan paristoilla toimiviin sovelluksiin [38]. Alla olevaan taulukkoon on kerätty joitain vertailtavissa olevia, tähän työhön liittyviä, lukuarvoja.

Taulukko 1. *Apollo3 vs MAX32668 [31][38].*

	BLE TX (0 dBm)	BLE RX	Syvä-uni (μ A)	Ajonaikainen- tehonkulutus (μ A / MHz)	Käyttö- jännite
Apollo3	3,00 mA	3,00 mA	1.2	6,0	3,3 V
MAX32668	4,30 mA	3,20 mA	10.0	27,3	3,3 V

Mikrokontrollereiden tai -järjestelmäpiirien soveltuvuuden tai paremmuuden arviointi erittäin pientä virrankulutusta vaativille sovelluksille ei ole näin yksinkertaista. Se että katsottaisiin vain datalehdeltä raportoituja virrankulutuksen arvoja, ei kerro todellista hyötysuhdetta. Valmistajat referoivat virrankulutuksen arvot erilaisiin tilanteisiin, käyttävät erilaisia nimiä samoille tiloille ja myös samanlaisia nimiä erilaisille. Kontrollerin oheislaitteet voivat kuluttaa huomattavan määrän virtaa ja kyky hallita näiden erilaisia virtaa kuluttavia tiloja voi olla ratkaisevaa, vaikka ilmoitettu ajonaikainen virrankulutus olisikin pienempi, yksi esimerkki tällaisesta voisi olla DMA-ohjain, joka sallii prosessorin olla unessa AD-muunnoksen tapahtuessa. Syvä-unitilasta herääminen voi olla hidasta, muistin ja väli-

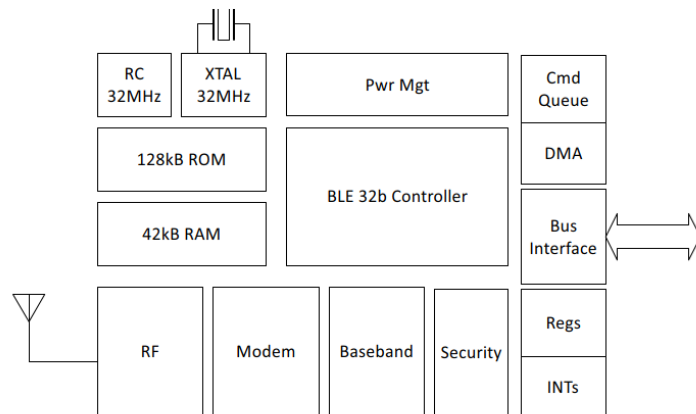
muistin ylläpitäminen viedä liikaa virtaa. Käyttöjännite ja kellotaajuus, millä ilmoitettu tehonkulutus on saavutettu, voi olla epärealistinen käytännön toteutuksissa. Tärkeää on myös, miten syväunitilasta voidaan herätä ilmoitetussa pienimmän virrankulutuksen tilassa, onko esimerkiksi reaaliaikakelloherätys mahdollinen. Se kuinka paljon RAM-muistia on päällä unitilassa, vaikuttaa virrankulutukseen. Huomioitavaa on myös, millaiset turvaominaisuudet ovat mahdollisia, kuten Brown-Out-Detection (BOD), eli laskeeko käyttöjännite liian matalaksi [39][40].

Suunnittelijoiden tulisi ensiksi määritellä laitteistovaatimukset ja vasta niiden perusteella etsiä kontrolleria. Laitteistovaatimukseen kuuluu esimerkiksi vasteaikavaatimukset ja käyttöjännitteen lähde, kuten paristo. Millaisia oheislaitteistoja tai muistia tarvitaan ja tarvitseeko näiden olla koko ajan päällä. Kun vaatimukset ovat selvät, vasta sitten voidaan vertailla eri valmistajien tuotteita kyseiseen toteutukseen.

Koska Ambiq:n Apollo3 järjestelmäpiiri edustaa markkinoiden parhainta virrankulutuksenhyötysuhdetta, niin ajonaikaista kuin unitilojen, ja se on toteutuksensa avulla saavuttanut merkittäviä edistysaskelia erittäin pienitehoisten langattomien järjestelmäpiirien alalla [33], sopii se projektin vaatimukseen hyvin ja on tutkimuskohteena mielenkiintoinen. Projektissa ei tarvita nopeaa vasteaikaa unesta herätessä. Tavoite on mahdollisimman pieni kokonaiskulutus ja että kontrolleri nukkuisi paljon. SRAM- ja FLASH-muistin voi sammuttaa syväunitilan ajaksi, koska suuria määriä tietoa ei tarvitse säilöä, mittaustulokset lähetetään heti valmistuttuaan.

5.2.1 BLE-lohko

Mikrokontrolleri järjestelmäpiirille on integroitu oma autonominen BLE5-yhteensopiva järjestelmä, joka koostuu radiovastaanottimesta, modeemista, kantataajuusyksiköstä ja mikrokontrollerista. Kuva 25 esittää Apollo3 blue BLE-järjestelmän lohkoakaavion. BLE-järjestelmä toimii 32 MHz kellotaajuudella, joko omasta sisäisestä tai ulkoisesta kellolähteestä.



Kuva 25 Apollo3 blue BLE-osalohko [31]

Kommunikointi järjestelmän kanssa tapahtuu omaa DMA-toimintoja tukevaa BLE-väylää käyttäen. Järjestelmä hoitaa BLE-arkkitehtuurin 'Controller' - ja "Host Controller Interface" (HCI) -osaa.

Järjestelmä tukee neljää samanaikaista yhteyttä projektissa käytetyllä B0-siruverziolla. Lähetys- ja vastaanottoteho on valmistajan ilmoittamien arvojen mukaan 0 dBm lähetysteholla 3 mA. Vastaanotin toimii -94 dBm herkkyydellä ja lähetysteho on mahdollista säätää arvojen -10 dBm, 0 dBm ja +3 dBm välillä. Lähetystehoa voidaan kasvattaa vielä ulkoisella tehovahvistimella tarvittaessa, jolle Apollo3 tarjoaa ulostulon. Järjestelmä pysyy säätämään omaa tilaansa ja menemään virransäästötilaan tarvittaessa, kun aktiivista tehtävää ei ole. BLE-lohko mahdollistaa myös Over-the-Air (OTA) -päivityksen, eli langattoman ohjelmistopäivityksen, missä pysyvästi tallennettu mikrokontrollerin ohjelmisto päivitetään [31].

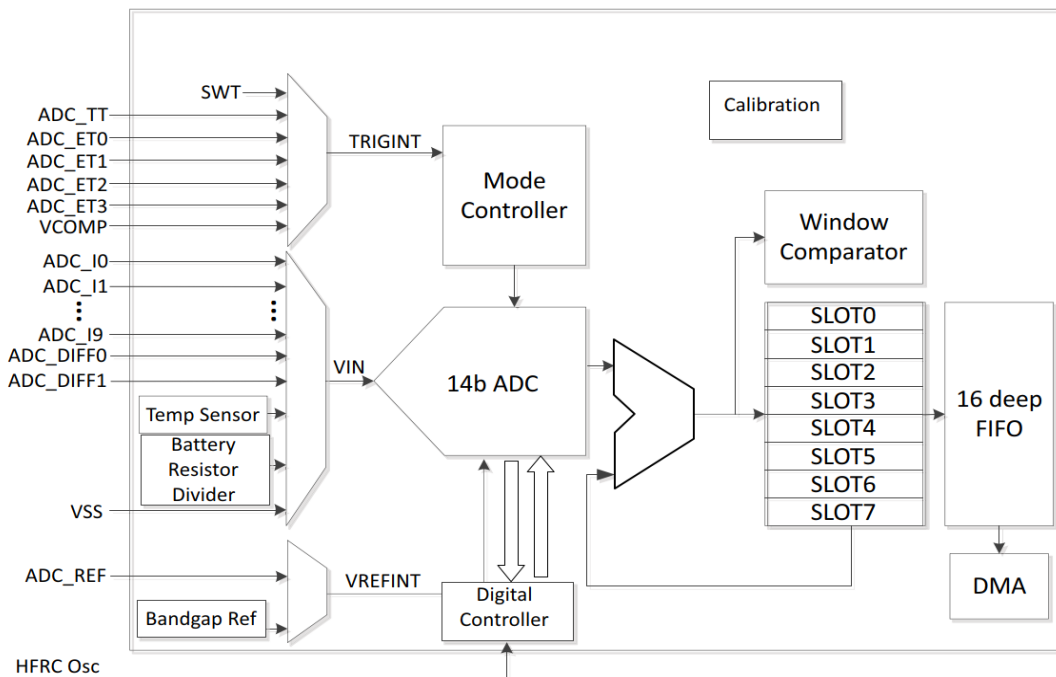
BLE-lohkoa ei ole pakko käyttää, sen voi kytkeä täysin pois päältä ja ohjelmallisesti se onkin pois päältä mikrokontrollerin käynnistyessä. Valmistaja kuitenkin suosittelee sen konfiguroimista ensiksi päälle ja sen jälkeen syväuneen, että voidaan olla varmoja virransäästöstä. Tällä voi virransäästämisen lisäksi säästää muutamia komponentteja ja tekee järjestelmäpiiristä soveltuvamman myös muihin sovelluksiin, jotka eivät tarvitse langatonta tiedonsiirtoa [31].

5.2.2 AD-muunnin

Johtuen projektin vaatimuksista siinä käytettävälle kiihtyvyyssanturille, on jouduttu valitsemaan anturi, missä on analoginen ulostulo. Tästä syystä joudutaan käyttämään mikrokontrollerin AD-muunninta muuttamaan ulostulo binäärimuotoon. Sen konfiguroimisessa tärkeitä suureita ovat muun muassa näytteenottotaajuus, joka määrittää maksimitaajuuden, jota pystytään mittaamaan ja näytteen resoluutio, joka kertoo kuinka monta

arvoa yksi näyte sisältää. Apollo3 Blue sisältää konfiguroitavissa olevan maksimissaan 14-bitin AD-muuntimen, jossa on esimerkiksi keskiarvoistuspiiri, FIFO ja DMA, mahdollisuus mitata piirin lämpötila ja sen maksiminopeus on jatkuva 2,67 MS/s. Kuva 26 esittää muuntimen lohkokaavion.

AD-muunnin on konfiguroitu käyttämään DMA:ta virrankulutuksen minimoimiseksi. Kun DMA on konfiguroitu, prosessori liipaisee muunnoksen käyntiin ja menee syväuneen odottamaan DMA-valmis keskeytystä. SRAM, minne DMA kirjoittaa mittaustulokset, täytyy pitää päällä syväunitilassa, että sinne voidaan kirjoittaa. AD-muunnin voidaan konfiguroida keskiarvoistamaan näytteitä 2^n kertaa, missä n on 0 – 7. Näitä keskiarvoja kerätään 16 paikkaiseen FIFO-muistiin ja kun se on täynnä, DMA siirtää näytteet SRAM:iin. Kun DMA on kerännyt konfiguroidun määrän näytteitä, se laukaisee keskeytyksen ja prosessori voi herätä syväunesta. Tämänlainen menetelmä missä automaattinen oheislaitelohko toimii ilman tarvetta prosessoriajalle, säästää resursseja ja virtaa, kun prosessori voi nukkua.



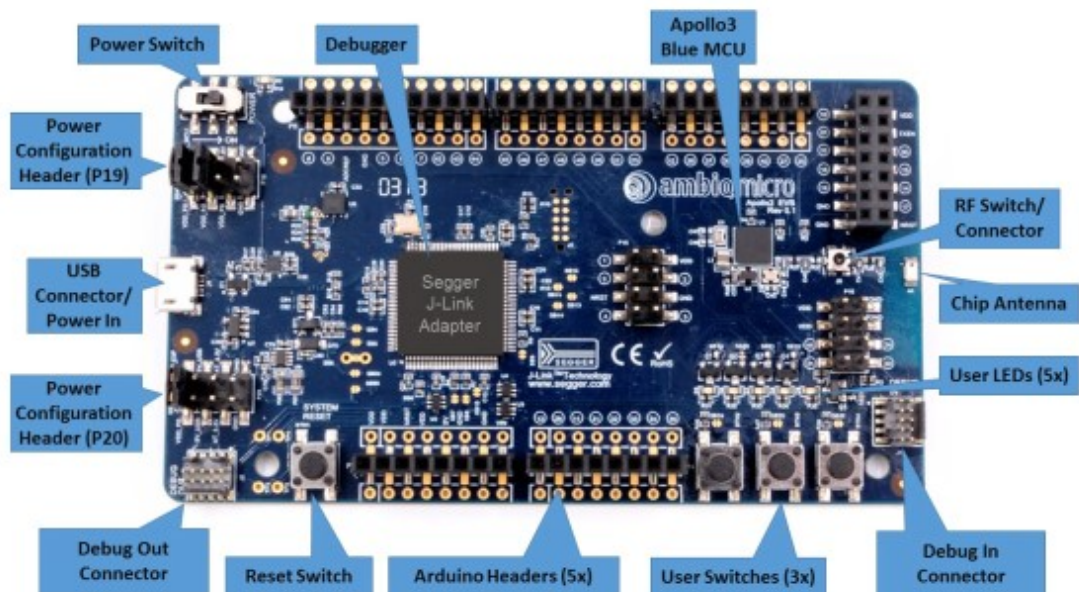
Kuva 26 AD-muuntimen lohkokaavio [31]

Raja-arvovertain (window comparator), voidaan konfiguroida aiheuttamaan keskeytys, jos mitattu jännite muuttuu määritellyn jännitearvon ulko- tai sisäpuolelle, siis ylittää raja-arvon kummasta suunnasta tahansa. Nämä ylitykset ovat kaksi erillistä keskeytystä.

5.2.3 Apollo3 Blue MCU EVB -kehitysalusta

Kehitystyötä tehtiin alussa Sparkfun:in kehitysalustalla, mutta kun huomattiin, että tämä ohjelmoimiseen käytetty kehitysympäristö ei salli muutoksia SRAM-asetuksiin ja Keil, joka esitellään luvussa 4.4, sallii, siirryttiin pois tästä. SRAM-asetuksilla voitiin määritellä unitilassa SRAM-alueita pois päältä ja säästää virtaa tällä tavalla. Sparkfun:in kehitysalustalle ohjelmistoa kehitettäessä SRAM-asetuksiin kajoaminen aiheutti prosessorin täydellisen jumitutumisen. SparkFun:in kehitysalustat on myös suunniteltu siten, että virranmittaukseen suunnitellun pinnan kautta kulki sähköä esimerkiksi "power-on" -ledille, eikä se sallinut järjestelmäpiirin todellisen virran mittaamista. Sparkfun kehitysalustoista ja ympäristöstä löytyy tietoa heidän nettisivuiltaan [7].

Kehitysalusta, jolla ohjelmistoa kehitettiin ja testattiin, on Ambiq Micron Apollo3 Blue MCU EVB [2]. Siinä on Arduino yhteensopivat liityntäpinnit, mikro-USB liityntä sisään rakennetulle Segger J-link virheenjäljittin- ja ohjelmointipiirille, Bluetooth Low Energy antenni, painonappeja ja muuta tarpeellista kehitystyötä helpottavaa liityntää mikrokontrollerille. Kuva 27 esittää kehitysalustaa. Yksi tärkeä ominaisuus on, että kehitysalusta on suunniteltu siten että mikrokontrollerin virrankulutusta on helppo mitata.



Kuva 27 Apollo3 kehitysalusta [2]

Levy on suunniteltu siten, että kaikki Apollo3-piirille menevä virta voidaan mitata helposti tähän omistetun pinnan avulla. Yllä olevassa kuvassa liitinrima P19. Levyn valmistuksen laatu on myös erinomaista.

5.2.4 Ambiq Data Transfer Profile

Projektissa on käytetty langattomaan BLE-tiedonsiirtoon Ambiq Data Transfer profiilia (AMDTP), joka on Ambiq:n kehittämä kustomoitu BLE-profiili. AMDTP:n määrittelemä palvelu ja palvelun piirteiden UUID:t on lueteltu taulukossa 2 [41].

Taulukko 2. *Ambiq Bluetooth UUID [41][43].*

Bluetooth SIG	Base UUID	00000000-0000-1000-8000-00805F9B34FB
Service Declaration	Ambiq Micro DTP	00002760-08C2-11E1-9073-0E8AC72E1011
Service Characteristic	Rx	00002760-08C2-11E1-9073-0E8AC72E0011
Service Characteristic	Tx	00002760-08C2-11E1-9073-0E8AC72E0012
Service Characteristic	ACK/Control	00002760-08C2-11E1-9073-0E8AC72E0013

Näille taulukossa mainituille palvelujen piirteille, Rx, Tx ja ACK/Control, on määritelty oikeuksia kuinka niiden sisältämään tietoon pääsee käsiksi. Attribuuttiprotokollassa tapoja on kuusi ja AMDTP käyttää niistä kolmea. Rx on määritelty olemaan "Write Command", Tx on "Notify" ja ACK/Control on "Write Command/Notify".

Profiiliin kuuluu kaksi ohjelmistoa, pääte- ja oheislaitteelle omansa, nimeltään AMDTPc ja AMDTPs, eli "client" ja "server". Tiedonsiirtoon näiden välillä on määritelty profiilissa paketti, joka ei ole BLE-paketti, vaan profiilin sisäinen tapa kapseloida tietoa. Kuva 28 havainnollistaa AMDTP-pakettia.

Length	Header	Data	Checksum (CRC32)
2 bytes	2 bytes	0 ~ 512 bytes	4 bytes

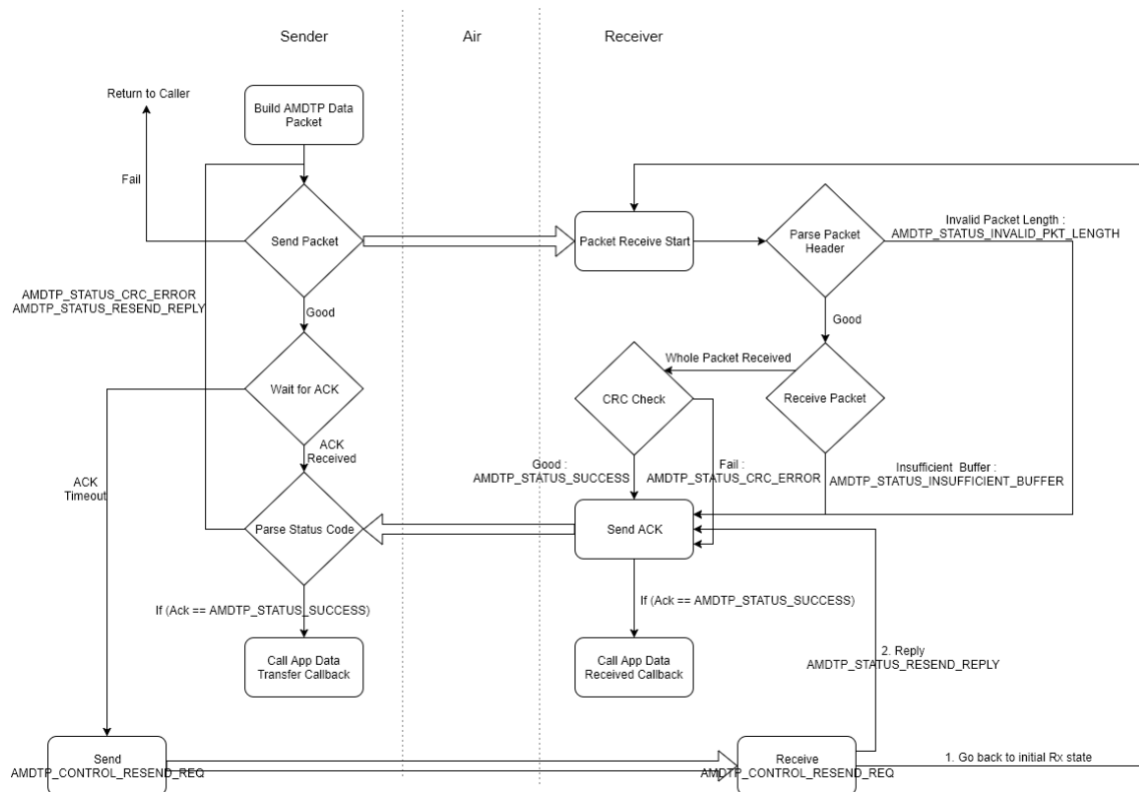
AMDTP Header Format

bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Pkt type				Serial Number				Enc	Ack	RFU						

Kuva 28 AMDTP-paketti [43]

Tämän AMDTP-paketti ratkaisun avulla pystytään siirtämään suuria määriä dataa, ja joko pääte- tai oheislaitte voi aloittaa tiedonsiirron. AMDTP pystyy pilkkomaan datan näihin lähetettäviin paketteihin ja koostamaan sen vastaanottavassa päässä takaisin kokoon.

AMDTP siis hoitaa datan paketoinnin BLE:lle niin, että käyttäjän ei tarvitse huolehtia tästä ja tilanteissa missä MTU voi olla ennalta määrittämätön ja jonka maksimiarvoa ei voida tietää ennalta. AMDTP vastaanottaessaan tietoa pystyy paketista päättelemään, onko dataa tulossa lisää vai ei. Kuva 29 havainnollistaa AMDTP-profiilin toimintaa [42][43].



Kuva 29 AMDTP:n toiminnan lohkokaavio [41]

Ambiq Software Development Kit (SDK) sisältää ohjeet, kuinka integroida AMDTP osaksi ohjelmistoa. Ohjeet ovat kuitenkin melko vaatimattomat. SDK on ladattavissa Ambiq:n sivuilta.

5.3 Kehitysympäristö

Kehitysympäristö tätä ohjelmistoa tehdessä oli ARM Keil Microcontroller Development Kit (MDK), joka on kattavin ohjelmistokehitysympäristö ARM-mikrokontrollereille ja sisältää kaiken tarvittavan sulautettujen ohjelmistojen luomiseen, kääntämiseen ja virheenjäljitykseen [8].

Hyvää ympäristössä on muun muassa se, että ohjelmisto tarkistaa ja visualisoi ohjelmistoa, kuten onko jokin määrittely (define) aktiivinen, tuottaako jokin funktiokutsu varoituksen tai virheen kääntäjältä, mikä helpottaa ohjelmistokehitystä. Funktioiden määrittelyyn ja lähdekooditiedostoon pääsee hyppäämään klikkaamalla valikon auki hiiren oikealla näppäimellä ja leijumalla funktion yläpuolella hiiren osoittimella ohjelma näyttää mitä funktio palauttaa ja mitä se odottaa parametreinaan. Käytettävissä on myös virheenjäljitys (debugger), jolla ohjelmaa voi ajaa käsky kerrallaan vikaa haettaessa.

Keil ympäristöön siirryttiin varsinkin sen vuoksi, että SparkFun:in ympäristössä, missä kehitystyötä tehtiin aluksi, ei voinut koskea SRAM-asetuksiin. Tämän epäillään johtuvan virheestä linkkerin muistinkäsittelyssä. Linkkeri-ohjelman vastuulla on koostaa ohjelmiston käännökset laitteiston fyysiseen muistiin. SRAM:iin sijoitettuja ajonaikana elintärkeitä muuttujia ei voida menettää, kun kontrolleri menee syväuneen ja sammuttaa osan SRAM-muistista. Jos linkkeri ei pysty järjestelemään muuttujia siten, että ne ovat muistialueella, joka ei sammu syväunitilassa, on seurauksena kontrollerin halvaantuminen. Keil myös kääntää ohjelmiston ja ohjelmoi mikrokontrollerin huomattavasti nopeammin.

5.3.1 Mittaus- ja virheenjäljitysympäristö

Ohjelmistoa kehitettäessä oli jatkuva tarve mitata mikrokontrollerin virrankulutusta ja havainnoida sen ajallista käyttäytymistä, kuten BLE-yhteyttä ja RTOS-toimintoja. Vianhaku ja toiminnallisuuden varmistusta oli myös tehtävä. Lähettimen ja vastaanottimen kanssa oli pystyttävä kommunikoimaan, lähettämään niille komentoja ja lukemaan niiden lähettämiä viestejä. Tähän käytettiin vastaanottimen päässä RealTerm-terminaali-ohjelmaa ja mikrokontrollerin UART-porttia. Lähettimen päässä käytettiin J-Link SWO Viewer -ohjelmaa, joka on "Instrumentation Trace Macrocell" (ITM) ja "Trace Port Interface Unit" (TPIU) -yhteensopiva, yksijohtiminen ARM-virheenjäljitysportti ja on osa Segger J-link ohjelmistoa. Seuraavassa aliluvussa on esitetty käytetty fyysinen mittalaitteisto [31].

5.3.2 Mittauslaitteisto

Hewlett-Packard 54645D 100 MHz -oskilloskoopilla havainnoitiin virrankulutuksen kuvaajaa ja sen käyttäytymistä ajan suhteen. Sitä käytettiin myös kehitystyön aikana ongelmien ratkomiseen ja ajallisen toiminnan kuten BLE-yhteyden asetusten muutoksien havainnoimiseen ja säätämiseen. Myös Pico Technology Picoscope 5444D MSO 200 MHz oskilloskoopia käytettiin samoihin tehtäviin ja myös kuvien ottamiseen tätä diplomityötä varten [47][49].

Hewlett-Packard 34401A -yleismittarilla mitattiin järjestelmäpiirin virrankulutusta. Virtamittaus tehtiin 10 ohmin 1 % vastuksen avulla, mittaamalla jännitehäviö tämän ylitse. Virta saadaan Ohmin lailla

$$I = \frac{U}{R} \quad (5)$$

jossa U on jännite vastuksen yli ja R on vastuksen resistanssi [25]. Virta laskettiin käytämällä jännitearvona yleismittarilla mitattua jännitteen keskiarvoa.

Mitattu jännite, minkä perusteella virta lasketaan, on voimakkuudeltaan hyvin pieni, joten käytetyn 34401A-mittarin mittatarkkuudella on merkitystä. Kaikki mittaukset on tehty mittarin tasajännite alueella. Nopea mittaus kuudella merkitsevällä luvulla toteuttaa integrointiajan, joka on 10 sähköverkon taajuuden jaksoa, eli 200 ms. Mittauksen resoluutio tällä tavalla on 0,000001 % 100 millivoltin mitta-alueesta (Full-Scale (FS)) ja nopeus 5 mittausta sekunnissa. Kyky vaimentaa sähköverkon hurinaa ja sen harmonisia komponentteja on 60 dB "Normal mode rejection" (NMR), jos sähköverkon taajuus pysyy vakiona. ± 3 % virhe taajuudessa muuttaa NMR:n arvon olemaan 30 dB. Virhe 100,0000 mV mitta-alueella on $\pm 0,0050$ % lukemasta + 0,0035 % mitta-alueesta [48].

Nopea mittaus viidellä merkitsevällä luvulla toteuttaa integrointiajan, joka on 0,2 sähköverkonjaksoa, eli 3 ms. Mittauksen resoluutio tällä tavalla on 0,00001 % 100 millivoltin mitta-alueesta (FS) ja nopeus 300 mittausta sekunnissa. Kyky vaimentaa sähköverkon hurinaa ja sen harmonisia komponentteja on 0 dB NMR. Mittatuloksen virhe on ilmoitettu käyttöohjeessa vain kuuden merkitsevän luvun mittaukselle, mutta tähän mittaukseen sisältyy lisävirhe 0,001 % mitta-alueesta + 20 μV . Se lisätäänkö tämä ilmoitettuun virheeseen, vai eikö muille kuin kuudelle merkitsevälle luvulle ole ilmoitettu virhettä, ei ole selvää [48].

Hidas mittaus neljällä merkitsevällä luvulla toteuttaa integrointiajan, joka on 1 sähköverkonjaksoa, eli 20 ms. Mittauksen resoluutio tällä tavalla on 0,000003 % 100 millivoltin mitta-alueesta (FS) ja nopeus 1000 mittausta sekunnissa. Kyky vaimentaa sähköverkon hurinaa ja sen harmonisia komponentteja on 60 dB NMR. Tähänkin mittaukseen sisältyy lisävirhe 0,001 % mitta-alueesta + 20 μV [48].

Kehitystyössä käytettiin lisäksi Hewlett-Packard 33120A -funktiogeneraattoria, joka emuloi kiihtyvyyssanturia ennen kuin kiihtyvyyssanturin prototyyppi oli käytössä, ja jolla testattiin mikrokontrollerin AD-muuntimen toimintaa ohjelmaa kirjoitettaessa. Funktiogeneraattorista syötettiin kanttiaaltoja 7,5 kHz:n taajuudella [48].

5.3.3 ADXL1003-kiihtyvyyssanturi prototyyppi

Analog Devices ADXL1003 -kiihtyvyyssanturille valmistettiin prototyyppilevy, jolle on juotettu myös pinnirima, minkä avulla levy kytketään kehitysalustaan. Kuva 30 esittää levyä. Tämä kiihtyvyyssanturi on samaa mallia mitä projekti käyttää Nokian renkaille valmistamassaan prototyyppijärjestelmässä.



Kuva 30 Projektissa valmistettu ADXL1003-kiihtyvyyssanturi prototylevy

Anturi mittaa kiihtyvyyden arvoja ± 200 g välillä, 15 kHz asti [44]. Nämä vaatimukset tulevat käyttökohteesta. Anturin on pystyttävä mittaamaan tarpeeksi hyvällä tarkkuudella tärinää pakettiauton renkaan sisällä, 80 - 100 km/h nopeudessa. Nokian Renkaat Oy:ltä saadun datan perusteella tärinän kiihtyvyydet olivat pääasiassa alle 200 g, ja tästä syystä on valittu ± 200 g anturi. Näytteenottotaajuuden tuli olla suuri siksi, että pystytään mittaamaan tarpeeksi datapisteitä ajassa, jossa rengas pyöri kerran 80 - 100 km/h nopeudessa. Tämä on 80 km/h nopeudella ja 215/60R17 pakettiautorengaalla, jonka ulkohalkaisija on 689,80 mm [50], ja kehä 2167 mm:

$$2167 \text{ mm} / \frac{80000 \text{ m}}{3600 \text{ s}} = 0,09752 \text{ s} \quad (5)$$

360 mittausa, siis 1 asteen välein yhdestä kierroksesta, vaatisi siis 4 kHz:n näytteenottotaajuuden.

Anturi on valmistettu CP-32-26 pakettiin, joka on kustomoitu "Lead Frame Chip Scale" -paketti. Sen mitat ovat 5 x 5 x 1,8 mm. Ja se painaa alle 0,2 grammaa [44].

6. LAITTEISTON TOIMINTA

6.1 Toiminnan kuvaus

Laitteisto koostuu kahdesta kehitysalustasta, joita käytetään suunnitteluun ja testaukseen ennen lopullista toteutusta. Ensimmäinen toimii päätelaitteena, joka vastaanottaa mittaustuloksia BLE:n kautta ja tulostaa ne sarjaportista, joka on kytketty kaapelilla tietokoneeseen Universal Serial Bus (USB) -muuntimen avulla. Tämä päätelaitteena toimiva laitteisto saa käyttösähkensä tietokoneesta USB-portin kautta.

Toinen kehitysalusta toimii oheislaitteena, tai lähettimenä, joka mittaa kiihtyvyyssanturin analogista ulostuloa AD-muuntimella ja lähettää mittaustulokset BLE:n kautta päätelaitteelle. Tämä laitteisto saa kaiken käyttösähkensä energiankeräyslaitteistolta.

Oheis- ja päätelaitteissa on molemmissa käytössä RTOS, joka käynnistetään pääohjelmassa, kun mikrokontrolleri on alustettu. Ohjelmiston osatehtävien suorittaminen siirtyy RTOS:n hoidettavaksi. Aluksi RTOS luo osatehtävät, suorittaa niiden alustusrutiinit ja siirtää sitten toiminnan vuorontajalle. Vuorontaja alkaa jakamaan prosessoriaikaa osatehtäville ja ohjelmiston suorittaminen alkaa.

RTOS toteuttaa funktiot, joita kutsutaan ennen ja jälkeen "Idle-task" -osatehtävää, jota suoritetaan, kun mitään muuta suoritettavaa ei ole. Funktioissa voidaan sammutella oheislaitelohkoja, esimerkiksi sarjaportti, ja oheislaitteen tapauksessa asettaa mikrokontrolleri syväunitilaan. Päätelaitteella ei ole tässä sovelluksessa virrankulutuksen suhteen vaatimuksia ja se voi olla koko ajan päällä säästämällä reaktioajassa sen mitä heräämiseen kuluu.

Päätelaitteessa on toteutettu RTOS:n sisäisten osatehtävien lisäksi vain yksi osatehtävä nimeltään "RadioTask". Tämä hoitaa BLE:n alustamisen, käynnistämisen ja sen tarvitseman keskeytyspalvelun. Täällä tehdään myös AMDTP:n käynnistäminen ja sarjakommunikointikeskeytyksen käsittely. Päätelaitteella toteutetaan BLE ja AMDTP -isännän roolia.

Oheislaitteessa on RTOS:n lisäksi kaksi osatehtävää. Alkuperäisen "RadioTask"-osatehtävän lisäksi on toteutettu uusi tehtävä nimeltään "AccelTask". RadioTask on muutoin samanlainen kuin päätelaitteessa, mutta konfiguroi oheislaitteen rengiksi. AccelTask konfiguroi AD-muuntimen ja DMA-ohjaimen tarvitsemat oheislaitteet ja toiminnot. Osatehtävä hoitaa niiden avulla kiihtyvyyssanturin ulostulon mittauksen ja mittaustulosten välittämisen AMDTP:n kautta BLE:lle ja edelleen päätelaitteelle.

AMDTP-profiiliin sovelluksessa oli vika, jonka löytämisessä ja korjaamisessa kului suhteellisen paljon aikaa. Tieto liikkui isännältä rengille mutta toiseen suuntaan ei, vaikka kaikki näytti olevan ohjelmallisesti kunnossa. Oikeita funktiota siis kutsuttiin, eikä virheilmoituksia tullut. Vika esti isäntää vastaanottamasta tietoa rengiltä, joka lähetti sitä ilmoitusviestiä (Notification) käyttäen. Laitteiden yhteyden muodostaminen ja muu kommunikaatio toimi oikein, koska siihen ei käytetä ilmoitusviestiä. Lopulta vian syyksi selvisi, että attRxHdl-tunniste, jonka avulla pitäisi käsitellä vastaanotettua tietoa, oli määritelty lähettämistä varten. Vika korjattiin uudelleen määrittelemällä attRxHdl-tunniste vastaanottamista varten ja sovellukseen luotiin uusi attTxHdl-tunniste, tiedon lähettämistä varten. Tunnisteita on esitelty taulukossa 3.

Taulukko 3. AMDTP-tunnistekorjaus.

Uusi tunniste (lähetys)	attTxHdl =	AMDTP_RX_HDL_IDX
Vanha tunniste (lähetys)	attRxHdl =	AMDTP_RX_HDL_IDX
Uusi arvo (vastaanotto)	attRxHdl =	AMDTP_TX_DATA_HDL_IDX

Lähetäessään ilmoitusviestin, renki laittoi paketin attRxHdl-tunnisteeseen arvon AMDTP_TX_DATA_HDL_IDX. Isäntäohjelmassa tätä ei saatu vastaanotettua, koska rxHdl-tunnisteessa oli lähetystä tarkoittava arvo AMDTP_RX_HDL_IDX.

Isäntäsovellukseen lisättiin ominaisuus automaattiseen yhteyden muodostamiseen rengin nimen perusteella. Tämä nimi on ihmiselle luettavissa olevassa muodossa. Käynnistyessään isäntä alkaa etsimään oheislaitteita ja yrittää muodostaa yhteyden laitteeseen, joka mainostaa itseään "Amdtp" -nimellä. Tämä mahdollistaa sen, että isäntä yrittää muodostaa yhteyttä vain laitteisiin, jotka se tunnistaa nimellä, eikä sen tarvitse kysellä rengiltä toteuttaako se isännän haluamaa profiilia tai palveluja. Toinen saavutettu etu on se, että jos renkiin haluttaisiin muodostaa yhteys esimerkiksi älypuhelimella, nimi erottuu laitelistauksesta selväkielisenä sen sijaan että näkyvissä olisi vain heksadesimaalimuotoinen "media access control address" (MAC) -osoite. Sovellukseen lisättiin myös toiminto sille, että jos yhteys katkeaa rengin toimesta, esimerkiksi sähkökatkoksen takia, isäntä alkaa uudelleen etsimään laitteita, jotka mainostavat itseänsä "Amdtp" -nimellä ja muodostaa yhteyden, jos sellainen löytyy uudestaan.

Oheislaite- eli renkisovellus alkaa automaattisesti mainostamaan itseään, jos yhteys isäntään jostain syystä katkeaa, mikä on BLE:n ominaisuus. Sovellukseen kuitenkin lisättiin ominaisuus, joka estää kiihtyvyyssanturimittausta käynnistymästä ennen kuin yhteys on lopullisesti muodostettu ja tietoa voidaan siirtää BLE:tä käyttäen. Yhteyden muodostamisessa ja vahvistamisessa kuluu jonkin aikaa ja näin säästetään energiaa, kun

AD-muunnosta ja muita siihen liittyviä toimintoja ei suoriteta turhaan. Odottaessaan yhteyden lopullista muodostamista, kiihtyvyyssmittausta suorittava osatehtävä on toimettomassa tilassa, eli ilmoittaa vuorontajalle, että se ei tarvitse prosessoriaikaa tällä hetkellä. Jos RTOS:ssa ei ole muuta tekemistä, se voi mennä Idle-task osatehtävään.

Kun laitteet ovat onnistuneesti muodostaneet yhteyden, käynnistetään kiihtyvyyssanturimittauksen suorittaminen. Ensiksi asetetaan päälle tarvittavat oheislaitelohkot, kuten AD-muunnin, DMA-ohjain ja kello AD-muunnoksen ajastamiseksi. AD-muunnos liipaisaan käyntiin ja osatehtävä siirtyy toimeettomaan tilaan vapauttaen prosessorin. AD-muunnoksen tulokset tallennetaan muistiin käyttäen DMA-ohjainta, eli AD-muunnin ja DMA-ohjain tallentavat tulokset muistiin ilman prosessoria. Kun määritelty määrä muunnoksia on saatu kerättyä, DMA-ohjain liipaisee keskeytyksen ja ilmoittaa RTOS:lle, että osatehtävä on valmiina jatkamaan suoritusta. Kun osatehtävä pääsee jatkamaan, se jakaa 14-bittiset AD-muunnokset BLE:n puskureihin sopivaan $2 * 8$ -bittiseen muotoon. Osatehtävä tarkistaa tämän jälkeen onko profiili tilassa, jossa se pystyy lähettämään tietoa. Jos profiili on vapaa, kutsuu osatehtävä profiilin tiedonlähettämiskäynnin. Seuraavaksi osatehtävä sammuttaa päälle laittamansa oheislaitteet ja ilmoittaa RTOS:lle olevansa toimeettomassa tilassa yhden sekunnin ajan. Tämän ajan kuluttua osatehtävä on jälleen valmis suorittamaan kiihtyvyyssanturimittauksia ja RTOS siirtää osatehtävän ajettavaksi heti kun mahdollista. Osatehtävä suorittaa kiihtyvyyssanturin mittaamiseksi samat toiminnot, mitä edellä on kuvattu. Liitteessä A on ohjelmaesimerkki tästä toiminnasta.

6.2 Virrankulutuksen minimoiminen

Kaikki mahdolliset oheislaitelohkot sammutetaan järjestelmän käynnistyksen yhteydessä, kuten esimerkiksi monet erilaiset sarjakommunikointilohkot. Myös kaikki liityntäpinnit (engl. pads) konfiguroidaan "General Purpose I/O" (GPIO) -pinneiksi ja kytketään pois päältä. Näin voidaan varmistua siitä, että mikään pinni ei kulutta virtaa turhaan. Pinnejä on 50 ja jos ne olisivat määrittelemättömässä tilassa vaikuttaisi se syväunitilan aikaiseen virrankulutukseen. FreeRTOS ja BLE konfiguroivat tarvitsemansa oheislaitteet myöhemmin oman alustusrutiininsa yhteydessä. Myös AD-muunnokseen tarvittava laitteisto alustetaan myöhemmin. Tällä tavalla tiedetään, että järjestelmässä ei ole mikään turha oheislaitelohko päällä.

Muistin virrankulutuksen minimoimiseksi koko väli- ja flash-muisti konfiguroidaan sammutettavaksi syväunitilassa. Flash-muistista vain puolet kytketään päälle ohjelman ollessa ajossa. Staattisesta lukumuistista kytketään päälle niin vähän kuin on tarpeen ohjelman toimivuuden kannalta. Siis 64 kB, eli 1/6-osa muistista, ja se konfiguroidaan säilyttämään sen sisältämä tieto syväunitilassa.

RTOS hoitaa osatehtävien ajastamisen ja siirtyy syväunitilaan, kun mahdollista. Se tarjoaa kaksi funktiota, joissa voidaan sammuttaa ja laittaa päälle asioita, ennen ja jälkeen syväunitilaa. Nämä funktiot ovat kuitenkin globaaleja, niitä kutsutaan aina kun syväunitilaan voidaan siirtyä, eivätkä ne ole osatehtäväkohtaisia. Joka kerta syväunitilaan siirryttäessä ja sieltä poistuttaessa kaikkien tarpeellisten laitteiden päälle ja pois laittaminen siellä veisi aikaa turhaan. Parempi tapa on hoitaa oheislaitteet ja pinnit pois päältä heti, kun niitä ei tarvita, kuten `AccelTask`-osatehtävässä on tehty.

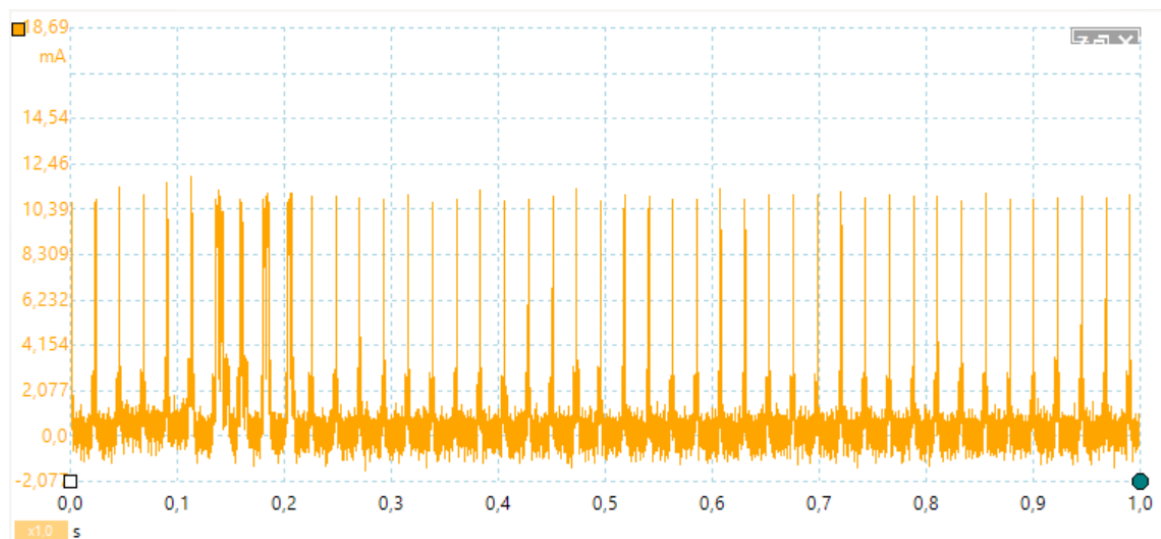
BLE-yhteystapahtuman aikavälin ja rengin latenssin asetukset vaikuttavat virrankulutukseen koska jokaisella yhteystapahtumalla tapahtuu radioliikennettä, vaikka mitään tietoa ei olisi tarve siirtää. Tämä on BLE-arkkitehtuurin mukaista toimintaa, mutta siihen voi jossain määrin vaikuttaa. Aikaväli voi olla esimerkiksi lyhyt ja rengillä paljon latenssia tai se voi olla niin pitkä kuin sovelluksen toiminta sallii. Näiden asetusten vaikutuksia on havainnollistettu seuraavassa aliluvussa. Sovellusta optimoitaessa yritettiin löytää paras kompromissi aikavälin ja latenssin suhteen niin, että ohjelma toimii. Paras suhde huomattiin olevan lyhyt aikaväli ja paljon latenssia. Siis nopea kommunikointi kun tietoa on siirrettävänä ja sitten pitkä väli seuraavaan yhteystapahtumaan, hylkäämällä tapahtumia latenssin avulla.

Sovellukseen on kirjoitettu ominaisuus lähetystehon pienentämiselle. Vakioteho on 0 dBm aina BLE:n alustamisen yhteydessä. Valmistajan rajapinta tarjoaa tälle vaihtoehdot "max value", +3 ja -10 dBm. Teho konfiguroidaan BLE-laitteiston alustamisen jälkeen virran säästämiseksi arvoon -10 dBm. Rajapintaa tarkemmin tutkittaessa huomattiin, ettei "max value" ole toimiva arvo ja palauttaa virheen. Vastaanottimen tehon muuttamiseksi rajapinta ei tarjoa funktioita. Kun lähdeohjelmätiedostoja tutkittiin rajapintaa syvemmälle, huomattiin että siellä olisi vaihtoehto myös -20 dBm asetukselle, mutta siihen ei pääse käsiksi rajapinnan kautta. Ehkä kyseinen kokoonpano ei tue tätä asetusta. Tätä olisi kuitenkin mielenkiintoista kokeilla muokkaamalla rajapintaa tähän soveltuvaksi.

AMDTP-profiiliin toteutettiin tarvittavat ominaisuudet MTU:n pituuden muokkausta varten ja renki ehdottaa BLE-yhteyttä muodostettaessa isännälle pisintä mahdollista pituutta. Isäntä kuitenkin tekee aina tämän päätöksen. Käytetyssä BLE-ohjelmistokokoelmassa, L2CAP on määritellyt maksimipituudeksi 251 tavua. Tämä muutos, missä renki ehdottaa pisintä sovelluksessa toimivaa MTU:ta on tehty sen takia, että L2CAP määrittelee BLE-paketin sisältämän datan määrän. Kun radio lähettää maksimimäärän dataa per paketti, se on kokonaisuudessaan päällä vähemmän aikaa.

6.3 Mittaustulokset ja analyysi

Kuva 31 esittää lähettimen virrankulutuksen kuvaajan yhden sekunnin, eli mittaus- ja lähetyksjakson ajalta, ilman optimointia, AMDTP-profiilin perusasetuksilla. Tämä on virrankulutuksen lähtöarvo, mistä sovellusta lähdetään optimoimaan. Virrankulutuksen kuvaajat on tehty oskilloskoopin matematiikkafunktiolla. Mittaukset on tehty mittamaalla A-kanavalla järjestelmäpiirin käyttöjännitelinjaan kytketyn vastuksen jännitelähteen puolelta ja B-kanavalla piirin puolelta. A-kanavan jännitteestä on vähennetty B-kanavan jännite ja erotus on jaettu vastuksen arvolla.



Kuva 31 Lähettimen optimoimaton virrankulutus ajanfunktiona [mA / s]

Yhteystapahtuman aikaväli on 22,5 ms. Latenssi on 0, eli renki vastaa isännälle jokaisessa yhteystapahtumassa. Mittaus tehdään myös yleismittarilla mittaamalla jännitehäviö 10 ohmin vastuksen yli ja virta lasketaan kaavalla 5. Mittauksen tulokset ovat: minimi 541 μ A, maksimi 5,14 mA, keskiarvo 1,06 mA, näytteiden määrä 38281.

Kuva 32 esittää tarkemmin osan lähettimen kiihtyvyyssanturidatan lähettämisestä BLE:llä. Tarkennus on yllä olevasta kuvasta aikaväliiltä 0,174 – 0,194 s. Lähetysteho kuvassa on BLE:n perusasetuksien mukainen 0 dBm.



Kuva 32 Lähettimen optimoimattoman lähetyksen kuvaaja

Kuvassa lyhyemmät terävät piikit ovat vastaanottimen kuluttaman virran kuvaajia ja niiden välissä olevat tasaisemmat kohdat lähettimen. Kuva auttaa havainnoimaan lähettimen tehon muutosta seuraavassa kuvassa alla.

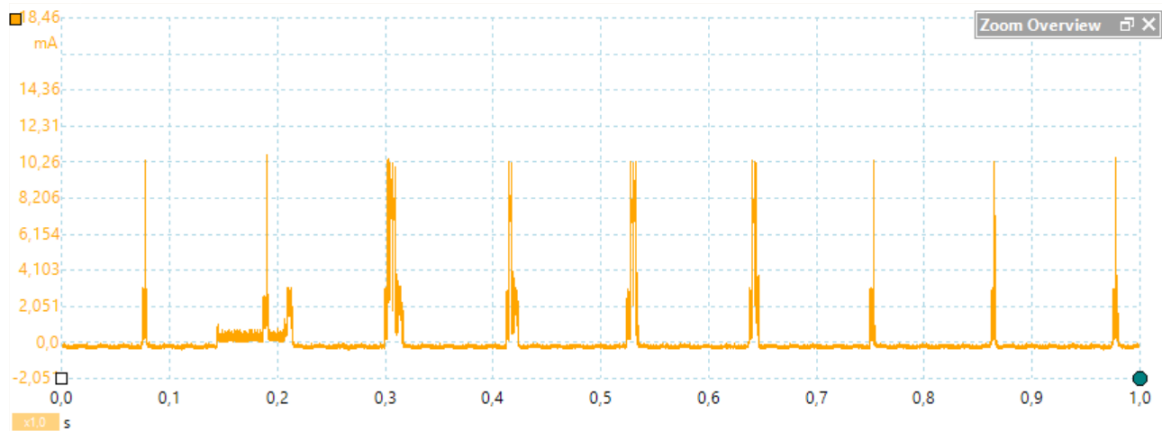
Kuva 33 esittää lähetyksen muuttamisen vaikutuksen virrankulutuksen kuvaajaan. Uusi lähetystehtä on -10 dBm. Muutos voidaan havaita virtapiikkien korostuessa, kun virtakuvaajan tasaiset kohdat laskevat. Piikit edustavat radion vastaanottimen kuluttamaa virtaa ja tasaiset kohdat kuvaavat uuden lähetyksen virrankulutusta.



Kuva 33 Lähettimen optimoitu lähetyksen kuvaaja

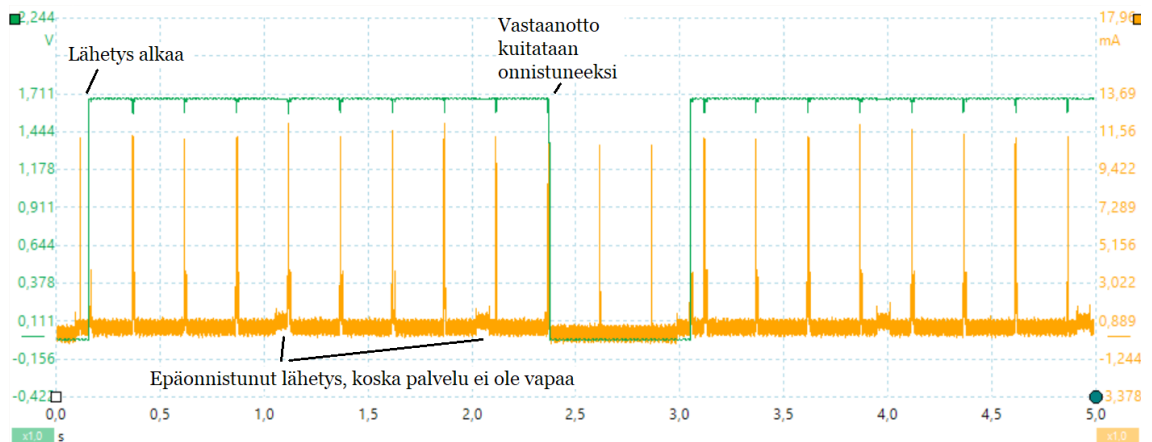
Virrankulutus putoaa noin 2 mA, sen ollessa aluksi noin 9 mA ja uudella arvolla noin 7 mA. 0 dBm tarkoittaa 1 mW tehoa, eli $1,8$ V käyttöjännitteellä millä järjestelmäpiiri toimii noin 555 μ A virtaa. -10 dBm tarkoittaa $0,1$ mW tehoa eli $55,5$ μ A virtaa tällä käyttöjännitteellä. Se miksi nämä luvut ei täsmää johtuu oskilloskoopin soveltumattomuudesta tällaisiin mittauksiin. Muutosta voidaan vain havainnoida ja se olisi erittäin vaikea todeta mittaamalla yleismittarilla johtuen sovelluksesta. Jos laite voitaisiin asettaa lähettämään tasaisesti dataa ilman muuta toiminnallisuutta, voitaisiin yleismittarilla todeta muutos tarkemmin. Tämä on kuitenkin diplomityön ulkopuolella.

Kuva 34 esittää yhteystapahtuman aikavälin muutoksen vaikutusta. Kuvassa on pisin yhteystapahtuman aikaväli millä sovellus toimii, renginvastelatenssi on nolla ja lähetysteho -10 dBm.



Kuva 34 Maksimi BLE-yhteyden aikaväli, 0 latenssi

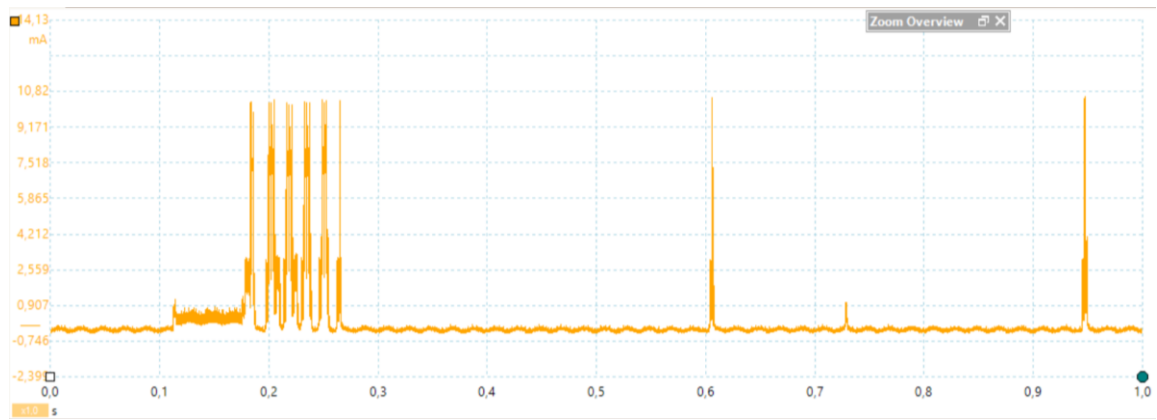
Yhteystapahtuman aikaväli on yllä olevassa kuvassa 112,5 ms. Yleismittarilla mitatuista arvoista lasketut virrankulutuksen tulokset ovat: minimi 3,47 μ A, maksimi 3,97 mA, keskiarvo 298 μ A, näytteiden määrä 3538. Jos aikaväliä olisi vielä kasvatettu sovellus olisi jättänyt lähetyksiä välistä. Syy tähän on se, että AMDTP-profiili, joka pilkkoo datan paketeiksi BLE:lle ja kokoaa sen jälleen vastaanottajan päässä, tarvitsee vastaanottajalta vahvistuksen, että kaikki data on vastaanotettu onnistuneesti. Kuva 35 esittää tätä ongelmatilannetta.



Kuva 35 BLE-yhteyden aikavälin pidentäminen liikaa

Kuvan kohdassa: Lähetys alkaa, asetetaan ulostulopinni 1-tilaan ohjelman kohdassa, jossa AMDTP-palvelu saa datapaketin lähetettäväksi. Kohdassa: Vastaanotto kuitataan onnistuneeksi, pinni asetetaan 0-tilaan ohjelman kohdassa, missä vastaanotetaan kuitaus, että koko paketti on siirtynyt onnistuneesti vastaanottajalle. Tämä toiminta ei ole BLE-arkkitehtuurin mukaista, vaan AMDTP-profiilin.

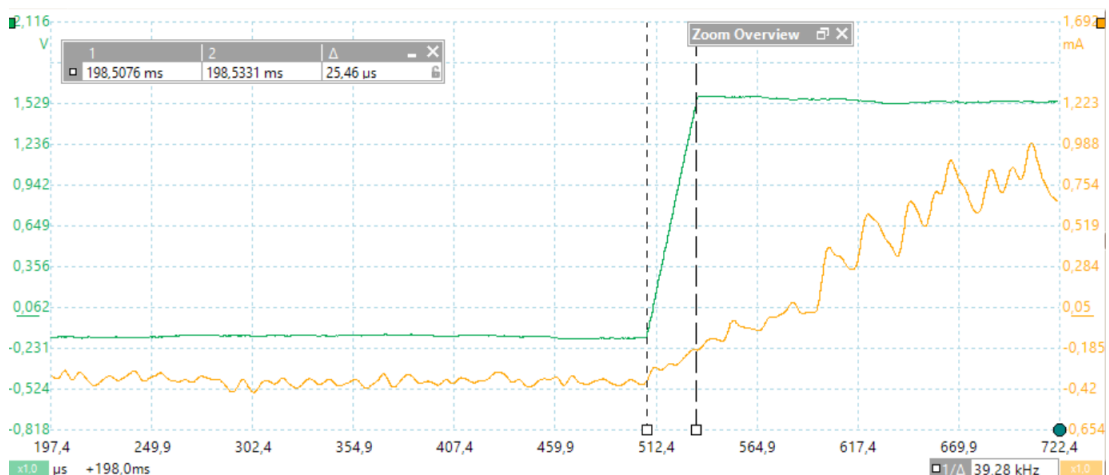
Kuva 36 esittää mahdollisimman lyhyen yhteystapahtuman aikavälin ja suuren latenssin arvon. Arvoja on haettu ensiksi pienentämällä aikaväliä niin pieneksi kuin sovelluksen toiminta sallii ja sen jälkeen alettu hakemaan maksilatenssia.



Kuva 36 Lyhyt aikaväli ja latenssi 20

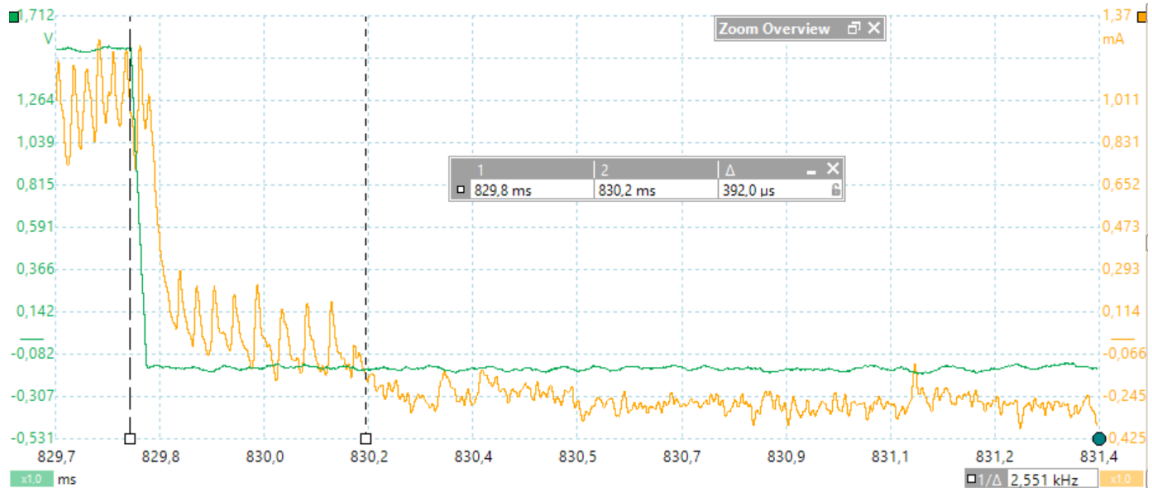
Kuvassa yhteystapahtuman aikaväli on 16,25 ms ja latenssi 20. Yleismittarilla mitatuista arvoista lasketut virrankulutuksen tulokset ovat: minimi 3,11 μA , maksimi 4,65 mA, keskiarvo 294 μA , näytteiden määrä 3505.

Muutama havainnollistus syväunesta ja sen toiminnasta. Testi on tehty siten, että ulos- tuloportti laitetaan päälle heti syväunitilasta poistuessa ja pois juuri ennen syväunitilaan menoa FreeRTOS:n tarjoamissa funktioissa. Tätä havainnoidaan oskilloskoopilla ja ver- rataan laitteen virran kulutukseen. Syväunitilasta heräämisaika voidaan nähdä virranku- lutuksen kuvaajasta, jota esittää Kuva 37. Kuvassa syväunitilasta herätään ajanhetkellä 198,5 ms, eli tiheäkatkoviivaisen kursorin kohdalla. Laatikko kuvassa vasemmalla näyt- tää, että toiseen kursoriin kuluu aikaa 25,46 μs . Pinnin tila on tässä kohdassa saavutta- nut 1-tilan.



Kuva 37 Syväunesta poistumisaika

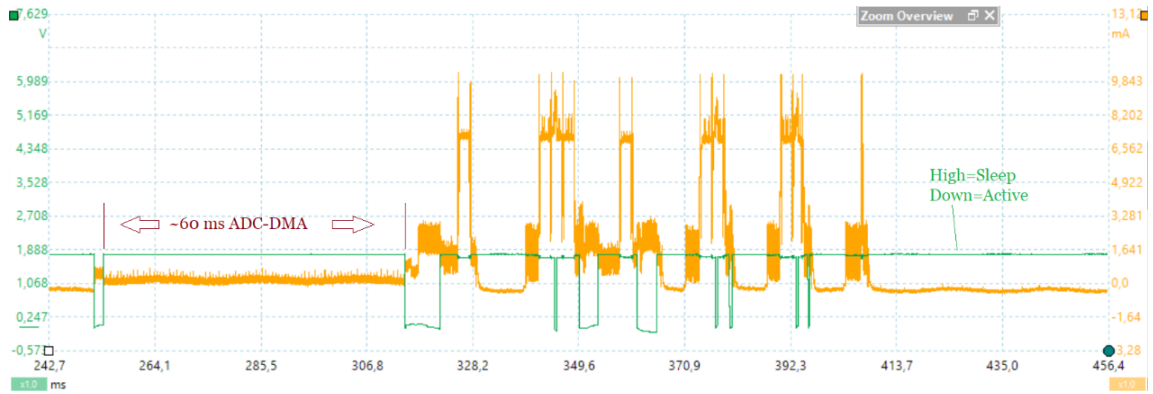
Syväunitilaan meno voidaan nähdä virrankulutuksen kuvaajasta, jota esittää Kuva 38. Kuvassa mennään syväunitilaan ajanhetkellä 829,8 ms, jossa on harvaviivainen kursori. Virrankulutuksen kuvaaja putoaa lopullisesti seuraavan kursorin kohdalla ja laatikko kuvan keskellä kertoo näiden välisen ajan olevan 392,0 μ s.



Kuva 38 Syväuneen meno

Käyttöohjeen mukaan nämä ajat ovat: syväunesta poistumisaika 25 μ s ja ajosta syväuneen 850 ns [31]. Kuvista on kuitenkin vaikea sanoa pitävätkö nämä ajat paikkaansa. Kuva 37 esittää kuinka virrankulutuksen kuvaaja alkaa hiljalleen nousemaan, ulostulo pinnan kytkeytyessä päälle 25 μ s kuluessa. Kuva 38 osoittaa että funktion jälkeen, josta mennään syväuneen, tapahtuu asioita vielä jonkin aikaa. Tätä 850 ns aikaa on siis tällä mittauksella vaikea todeta. Virtaa kuuluu selvästi ainakin noin 392 μ s, FreeRTOS syväuneenmeno-funktion jälkeen. Tämä havainnollistaa sitä, että syväunitilassa käytäessä aikaa kuluu hukkaan jonkin verran, varsinkin kun mittaus ei pysty näyttämään aikaa, mikä RTOS:lla menee palauttaa seuraavaksi suoritettava osatehtävä ajoon.

Kuva 39 esittää kuinka prosessori on unitilassa aktiivisen mittauksen ja lähetyksen aikana. Ulostulopinnan tilat ovat toisinpäin kuin edellä, pinni on ylhäällä prosessorin ollessa unessa. Kuva on tarkennus mittaus- ja lähetyssyklistä, joka voidaan nähdä myös Kuva 36 aikavälillä 0,1 – 2,8 s.



Kuva 39 Tiedonlähettäminen ja unitilat

Kuvasta näkee hyvin, että prosessori nukkuu DMA-ohjaimen kirjoittaessa AD-muunnoksen tuloksia muistiin ja silloin kun BLE-järjestelmän oma kontrolleri hoitaa omaa toiminnallisuuttaan tiedon lähettämiseksi. Muut hereilläoloajat liittyvät esimerkiksi AMDTP-profiiliin ja FreeRTOS:n toimintoihin.

6.4 Järjestelmän jatkokehitysideoita

Tässä työssä kehitetty järjestelmä täytti sille asetetut vaatimukset. Tätä raporttia kirjoitettaessa syntyi kuitenkin uusia ideoita, kuinka järjestelmää voisi vielä jatkossa kehittää eteenpäin. Näitä ideoita on esitetty seuraavissa kappaleissa.

Apollo3:n AD-muunnin tarjoaa kanavan, jolla voidaan mitata suoraan järjestelmäpiirin käyttöjännitettä. Se on mahdollista konfiguroida käyttämään raja-arvovertailijaa ja laukaisemaan keskeytys, kun käyttöjännite saavuttaa tarpeeksi korkean tason, esimerkiksi 3,3 V. Tällä jännitetasolla tiedetään, että energiavarastona toimiva super-kondensaattori on varautunut tarpeeksi. Ohjelmistoa muokattaisiin siten että, kun järjestelmäpiiri on saatu alustettua käynnistyksen jälkeen, menisi se syväunitilaan odottamaan konfiguroitua käyttöjännitetasoa ennen RTOS:n ja BLE:n alustamista. Kun tavoiteltu jännitetaso saavutetaan, herää järjestelmä syväunitilasta keskeytyksen avulla, ja RTOS:n ja BLE-osalohkon alustaminen aloitetaan. Tämä olisi hyvä toiminta energiankeräysjärjestelmässä, jossa hallintapiiri ei tarkkaile energiavaraston varauksen tasoa, koska BLE-yhteyden muodostaminen vie suhteellisen paljon energiaa. Tämä johtuu siitä, että renki joutuu mainostamaan itseään ja yhteyden muodostuksen aikana tapahtuu paljon radioliikennettä. Apollo3-järjestelmäpiiri käynnistyy jo 1,75 voltin käyttöjännitteentasolla ja nykyisessä sovelluksessa alustaa kaiken tarvitsemansa, aloittaa heti mainostamisen ja yrittää muodostaa yhteyden. Ensimmäisen radiolla lähetetyn paketin virrankulutuksen takia käyttöjännitteen taso saattaa pudota niin alas, että se aiheuttaa piirille uudelleenkäynnistyksen. Tämä ratkaisu takaisi, että järjestelmällä olisi tässä tilanteessa maksimi energia käytössään.

Järjestelmää saisi kehitettyä paremmaksi luomalla komentoja, joilla voisi isännän kautta ohjata AD-muunnoksen näytteenottotaajuutta, näytteiden määrää ja kuinka usein näytteitä otetaan. Tämän johdosta järjestelmä olisi monikäyttöisempi. Esimerkiksi jos huomataan, että kiinnostava taajuusalue tai kiihtyvyyden alue on mitattavaa aluetta huomattavasti alempana. Muuttamalla AD-muunnoksen asetuksia, saadaan tarkkuutta parannettua mittauksen kannalta kiinnostavalla alueella. Nämä muutokset tarvitsisivat muokkauksia myös BLE-yhteyden asetuksiin. Muokkaukset voisivat tapahtua ohjelmallisesti rengin puolella niin, että käyttäjän ei tarvitsisi laskea uusia arvoja. Asetuksia täytyisi kuitenkin olla jokin rajallinen määrä, että järjestelmän eri asetusvaihtoehdot olisi kohtuullisesti toteutettavissa ohjelmaan.

Paljon virtaa kuluttava vaihe järjestelmässä on mainostaminen ja yhteyden muodostaminen. Jos mainostetaan tiheällä aikavälillä, jolloin isännällä on suurin mahdollisuus kuulla mainos, kuluu radioliikenteeseen paljon energiaa. Jos mainostetaan harvoin säästetään energiaa, saattaa kestää kauan, että isäntä löytää rengin. Näiden optimoiminen toisi huomattavia säästöjä järjestelmän virrankulutukseen.

Nopea yhteyden muodostaminen on mahdollista käyttämällä kohdistettua mainostamista (engl. direct advertising). Tämä mahdollistaa nopeimman yhteyden muodostamisen siten, että rengin lähettämä kohdistettu mainospaketti (engl. direct advertising packet) sisältää sen oman osoitteen ja isännän osoitteen, johon yhteys halutaan muodostaa. Isäntä, joka vastaanottaa tällaisen paketin, lähettää yhteydenmuodostuspaketin välittömästi rengille, jolta kohdistettu mainospaketti vastaanotettiin [41].

Tärkein ratkaistava asia tässä tapauksessa on se, kuinka rengin sovellus toteutetaan siten, että isännän osoitteen määrittäminen on mahdollista. Osoitteen konfiguroiminen ei saisi olla liian hankalaa, jotta sovellus olisi mahdollisimman yleiskäyttöinen, eikä vaatisi tietoteknistä erityisosaamista. Esimerkiksi siten, että ensimmäisen kerran isäntä hakee renkiä kuuntelemalla mainoksia ja muodostaa yhteyden, rengin mainostaman nimen perusteella. Kun muodostaminen on tehty onnistuneesti, tallentaa renki isännän osoitteen pysyväismuistiin. Tämän jälkeen se voi lähettää jatkossa isännälle kohdistetun mainospaketin. Tilannetta varten, jossa halutaan vaihtaa isäntälaitte uuteen, tulisi olla yksinkertainen tapa pyyhkiä pysyväismuisti.

Energiankeräimen tärkein jatkokehitystarve on toimintataajuusalueen laajentaminen. Nykyisellään tehontuotto tapahtuu 120 ja 140 Hz välillä. Suuremmasta taajuuskaistasta olisi kuitenkin hyötyä useammassa sovelluksessa. Tehontuotto nykyisellään on erinomainen, eikä haittaisi, vaikka laajempi tehontuoton taajuusalue sitä hieman pienentäisi.

7. YHTEENVETO

Diplomityön kirjallisuustutkimusosuuden tarkoituksena oli ymmärtää mitä tarkoitetaan energian keräämisellä ja mitä menetelmiä siihen on. Työn jälkimmäisen osuuden tarkoituksena oli ymmärtää, miksi energian kulutuksen minimoimista tarvitaan kehitettäessä energiaomavaraisenlaitteiston energiaa käyttävää IoT-osuutta ja mitä menetelmiä siihen on.

Projektissa, johon diplomityö tehtiin, kehitettiin sähkömagneettiseen induktioon perustuva energiankeräin. Sille kehitettiin energianhallintapiiri, joka varaa tuotettua sähköenergiaa superkondensaattoriin. Energiankeräimen maksimi ulostuloteho saavutetaan 130 Hz resonanssitaajuudella. Ulostulotehoksi tällä taajuudella käyttäen 389,9 ohmin kuormavastusta mitattiin 65 mW. Tämän energiankeräyslaitteiston tehontuotto riittää hyvin kehitetyn IoT-laitteiston tehonkulutuksen palvelemiseksi.

Energiaomavaraisen järjestelmän energiaa käyttävä osuus on yleiskäyttöinen IoT-järjestelmä, joka siirtää suhteellisen suuren määrän tietoa langattomasti BLE-teknologiaa käyttäen. Se on toteutettu Ambiq Micro Apollo3 -järjestelmäpiirillä, jonka energiatehokkuus on markkinoiden huipputasoa ja sisältää integroidun BLE-järjestelmän.

BLE-teknologia on kehitetty erittäin pienitehoiseksi ja se on yleisesti käytössä. Sen käyttö on edullista ja se on yhteensopiva monien eri alustojen kanssa, kuten esimerkiksi älypuhelimien ja kannettavien tietokoneiden. Projektissa ei tarvitse siirtää tietoa kuin muutamia kymmeniä kilotavuja lähietäisyydelle. BLE on näiden kriteerien vuoksi hyvin soveltuva tämän kaltaisiin energiatehokkuutta vaativiin sovelluksiin.

Työssä kehitettiin onnistuneesti ohjelmistot vastaanottimelle ja lähettimelle. Ohjelmisto mittaa kiihtyvyyssanturitietoa ja siirtää sen langattomasti lähettimeltä vastaanottimelle. Kun ohjelmistot oli saatu toiminnallisuuksiltaan valmiiksi, lähettimen ohjelmisto optimoitiin virrankulutuksen suhteen. Jotta suurin mahdollinen tehonhyötysuhde saavutettiin, ohjelmisto optimoitiin laajalti juuri tähän kyseiseen sovellukseen.

Optimoinnin tuloksena saavutettiin huomattava sähkönkulutuksen väheneminen. Minimivirrankulutus optimoimattomassa tiedonsiirtosovelluksessa oli noin 540 μA ja keskiarvo noin 1,1 mA. Optimoinnin tuloksena järjestelmän syväunitilan aikaisessa minimivirrankulutuksessa päästiin arvoon noin 3,1 μA , ja keskiarvokulutuksessa arvoon noin 295 μA . Virrankulutuksen säästöt saavutettiin optimoimalla radioliikenteen toimintaa ja sammuttamalla käyttämättömät järjestelmäpiirin osat. Ohjelmistossa varmistettiin myös, että syväunitilan ajaksi kaikki sovelluksen käyttämät osat sammutettiin.

Työn tekemisessä kului paljon aikaa Ambiq SDK -ohjelmistokehityskirjaston, eli valmistajan tarjoaman ohjelmistokokoelman ja ARM Keil uVision -kehitysympäristön opettelemisessa, mutta toisaalta se kerrytti paljon uutta oppia ja kokemusta. Myös BLE-toiminnallisuuden ja asetusten opettelemisessa kului huomattava määrä aikaa.

Työ saavutti tavoitteet siinä, että kirjallisuustutkimusosuus perehtyy laajalti eri energiankeräysmenetelmiin ja energiakeräinteknologioihin. Työssä on tullut selville energiaomavaraisen järjestelmän hyödyt, kuten esimerkiksi paristoista luopuminen. Energiaa keräävän järjestelmän käyttö kuitenkin vaatii, että se suunnitellaan sopivaksi sen käyttöympäristöön. Ympäristössä on oltava saatavilla energiankeräimen lähde-energiaksi soveltuvaa energiaa, kuten tärinää. Työssä on tullut hyvin selville myös se, miksi energiankulutuksen minimoimista tarvitaan ja miten se on mahdollista toteuttaa energiaomavaraisessa sulautetussa IoT-järjestelmässä. Toteutettu lähetinsovellus hoitaa sille suunnitellun tehtävän tehonkulutuksella, joka on erinomainen saatavilla olevan tehon suhteen. Työssä tuli selville se, että tehonkulutusta voidaan pienentää optimoimalla sovellus käyttökohteeseensa.

Energiankeräimen tuottamalla sähköllä toimiva IoT-järjestelmä hyötyisi BLE-yhteyden muodostusprosessin parantamisesta. Mitä nopeammin yhteys lähettimen ja vastaanottimen välillä saadaan aikaiseksi, sitä nopeammin lähetin pääsee pienivirtaiseen tiedonlähetystilaan. Mainostaessaan se joutuu lähettämään mainospaketteja usein ja tämä radioliikenne kuluttaa suhteellisen paljon energiaa. Sopiva jatkokehitys sovellukselle olisi-kin mainostamisen ja yhteyden muodostamisen optimointi.

LÄHTEET

- [1] Dhadwal HS, Rastegar J. Energy harvesting: for low-power autonomous devices and systems. Bellingham, Washington: SPIE Press; 2017.
- [2] Ambiq Micro Inc., Apollo3 Blue EVB Board Revision 1.7 Quick Start Guide, July 2020, s. 5-14, <https://ambiq.com/apollo3-blue/>
- [3] Beeby, S, & White, Energy Harvesting for Autonomous Systems, Artech House, Norwood, N 2010, Available from: ProQuest Ebook Central, s 100 -103
- [4] Maurizio Di Paolo E. Microelectronic Circuit Design for Energy Harvesting Systems – 2.2 The Fundamentals of Energy Harvesting. 2017. s 13. <https://doi-org.libproxy.tuni.fi/10.1007/978-3-319-47587-5>
- [5] A. Kingatua, The How and Why of Energy Harvesting for Low-Power Applications, All About Circuits. Saatavissa (viitattu 5.8.2021): <https://www.allaboutcircuits.com/technical-articles/how-why-of-energy-harvesting-for-low-power-applications/>.
- [6] Lightyear, 2021. Lightyear. Saatavissa (viitattu: 5.8.2021): <https://lightyear.one/>
- [7] Kanoun, O (ed.) 2018, Energy Harvesting for Wireless Sensor Networks: Technology, Components and System Design, Walter de Gruyter GmbH, Berlin. Saatavissa (viitattu 4.8.2021): ProQuest Ebook Central.
- [8] SparkFun Electronics 2021. Using SparkFun Edge Board with Ambiq Apollo3 SDK. Saatavissa (viitattu 5.8.2021): <https://learn.sparkfun.com/tutorials/using-sparkfun-edge-board-with-ambiq-apollo3-sdk>
- [9] ARM limited. 2005-2019. MDK Microcontroller development kit. Saatavissa (viitattu 5.8.2021): <https://www2.keil.com/mdk5>
- [10] Amazon Web Services Inc. The FreeRTOS™ Kernel. Saatavissa (viitattu: 5.8.2021): <https://www.freertos.org/RTOS.html>
- [11] Bluetooth SIG, Inc 2021. Bluetooth® Wireless Technology. Saatavissa (viitattu: 5.8.2021): <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- [12] Suomen virallinen tilasto (SVT): Jätetilasto [verkkójulkaisu]. ISSN=1798-3339. 2018. Helsinki: Tilastokeskus. Saatavissa (viitattu: 9.8.2021): http://www.stat.fi/til/jate/2018/jate_2018_2020-01-15_tie_001_fi.html
- [13] Helsingin seudun ympäristöpalvelut -kuntayhtymä. 2018. Pääkaupunkiseudun sekajätteen koostumus vuonna 2018. Saatavissa (viitattu: 9.8.2021): <https://kivo.fi/yymmarramme/koostumustietopankki/>
- [14] Behr Technologies Inc 2020. 6 Leading Types of IoT Wireless Tech and Their Best Use Cases. Saatavissa (viitattu: 9.8.2021): <https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/>

- [15] Jia X, Feng Q, Fan T, Lei Q. RFID technology and its applications in Internet of Things (IoT). In: 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet). IEEE; 2012. p. 1282–5.
- [16] Gao M, Wang P, Wang Y, Yao L. Self-Powered ZigBee Wireless Sensor Nodes for Railway Condition Monitoring. IEEE transactions on intelligent transportation systems. 2018.
- [17] Aragón-Zavala A. Indoor Wireless Communications: From Theory to Implementation. 2.4 ZigBee. Somerset: John Wiley & Sons, Incorporated; 2017.
- [18] AtlasRFIDstore 2021. What is RFID? Saatavissa (viitattu: 10.8.2021): <https://www.atlasrfidstore.com/rfid-beginners-guide/>
- [19] IPCC, 2021: Summary for Policymakers. In: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Masson-Delmotte, V., P. Zhai, A. Pirani, S. L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T. K. Maycock, T. Waterfield, O. Yelekçi, R. Yu and B. Zhou (eds.)]. Cambridge University Press. In Press
- [20] Fodor E. 2020. Zigbee vs. Bluetooth: Choosing the Right Protocol for Your IoT Application. Saatavissa (viitattu 11.8.2021): <https://www.digi.com/blog/post/zigbee-vs-bluetooth-choosing-the-right-protocol>
- [21] European Centre for Disease Prevention and Control (ECDC) 2021. Questions and answers on COVID-19: Basic facts. Saatavissa (viitattu 11.8.2021): <https://www.ecdc.europa.eu/en/covid-19/questions-answers/questions-answers-basic-facts>
- [22] Powercast Corp. 2021. Temperature Scanning System. Saatavissa (viitattu 11.8.2021): <https://www.powercastco.com/products/temperature-scanner/>
- [23] <https://www.u-blox.com/zh/publication/white-paper/short-range-low-power-wireless-devices-and-internet-things-iot>
- [24] Andersson M. 2015. Short range low power wireless devices and Internet of Things (IoT). White paper. U-blox
- [25] Halliday D, Walker J, Resnick R. Fundamentals of physics. 7th ed. Wiley; 2005.
- [26] Liu H, Cao C, Sun X, Zhao L, Cong C. Magnetostrictive iron–gallium alloy harvester with efficient two-mode AC–DC converting technology for effective vibration energy harvesting. AIP advances. 2020.
- [27] Barry R. 2016. Mastering the FreeRTOS™ Real Time Kernel. A Hands-On Tutorial Guide. Real Time Engineers Ltd.
- [28] Amazon Web Services Inc. The FreeRTOS™ Kernel. Low Power Support. Saatavissa (viitattu: 17.8.2021): <https://www.freertos.org/low-power-tickless-rtos.html>
- [29] Tan J. RTOS Basics: Getting Started with Microcontrollers. Saatavissa (viitattu: 17.8.2021): <http://www.seeedstudio.com/blog/2021/04/26/rtos-basics-getting-started-with-microcontrollers/>

- [30] Ballou G. Electroacoustic Devices: Microphones and Loudspeakers. Oxford: Routledge; 2009.
- [31] Ambiq Micro, Inc. 2021. Apollo3 Blue MCU Datasheet, Revision 0.15.0.
- [32] Ambiq Micro, Inc. 2021. Ambiq Micro Achieves World-Leading Power Consumption Performance with TSMC 40ULP Technology. Saatavissa (viitattu: 20.8.2021): <https://ambiq.com/ambiq-micro-achieves-world-leading-power-consumption-performance-with-tsmc-40ulp-technology>
- [33] Ambiq Micro's Next-Generation Subthreshold Power-Optimized Technology (SPOT) Platform Features CEVA's Bluetooth Low Energy IP: Ambiq Micro's Apollo3 Blue Family Wireless System-on-Chip (SoC) already powered by the new Bluetooth licensing agreement. PR Newswire 2020 Mar 10.
- [34] Wikipedia. MOSFET. Saatavissa (viitattu: 20.8.2021): <https://fi.wikipedia.org/wiki/MOSFET>
- [35] Salas M. On the Threshold of a Low Power Revolution. Saatavissa (viitattu: 20.8.2021): <https://www.gsaglobal.org/forums/on-the-threshold-of-a-low-power-revolution/>
- [36] Taiwan Semiconductor Manufacturing Company Limited 2010-2021. Logic Technology. 40nm Technology. Saatavissa (viitattu: 20.8.2021): https://www.tsmc.com/english/dedicatedFoundry/technology/logic/l_40nm
- [37] McMahan S. 2019. Wireless SOC for Battery-Power Devices Achieves Under 6 μ A/MHz. Saatavissa (viitattu: 20.8.2021): <https://eepower.com/news/ambiq-micro-microcontroller-achieves-under-6cebca-mhz-with-tsmc-40ulp-technology/#>
- [38] Maxim Integrated. 2021. Low-Power Arm Cortex-M4 with FPU-Based Microcontroller with Bluetooth 5 for Wearables. Datasheet.
- [39] Albus Z, Valenzuela A, Buccini M. 2019. Ultra-low Power Comparison: MSP430F2x MCUs vs. Microchip XLP Tech Brief. Texas Instruments.
- [40] Hertz J. 2021. Why Comparing Low-Power MCUs Can Be So Difficult. Saatavissa (viitattu: 23.8.2021): <https://www.allaboutcircuits.com/news/why-comparing-low-power-mcus-difficult/>
- [41] Heydon R. Bluetooth low energy: the developer's handbook. 1st edition. Place of publication not identified: Prentice Hall; 2012.
- [42] Ambiq Micro, Inc. 2021. AmbiqSuite SDK v2.5.1. Saatavissa (viitattu: 24.8.2021): <https://ambiq.com/wp-content/uploads/2020/09/AmbiqSuite-R2.5.1.zip>
- [43] Ambiq Micro, Inc. 2017. AMDTP_example_user's_guide. Revision V0.1.
- [44] Analog Devices, Inc. 2018. ADXL1003 datasheet.
- [45] Texas Instruments, Inc. 2019. bq25570 nano power boost charger and buck converter for energy harvester powered applications. Datasheet. Revision G.
- [46] Central Semiconductor Corp. 2020. CBRHDSH1-40L. Datasheet. Revision 5.

- [47] Pico technology. PicoScope 5000 Series. Saatavilla (viitattu: 31.8.2021): <https://www.picotech.com/oscilloscope/5000/flexible-resolution-oscilloscope>
- [48] Keysight Technologies. 2021. 34401A Digital Multimeter, 6½ Digit. Saatavissa (viitattu:1.9.2021): <https://www.keysight.com/fi/en/product/34401A/digital-multimeter-6-digit.html>
- [49] Keysight Technologies. 2021. 54645D 100-MHz 200-MSa/s Mixed Signal Oscilloscope. Saatavissa (viitattu:): <https://www.keysight.com/fi/en/product/54645D/100mhz-200msas-mixed-signal-oscilloscope.html>
- [50] 1010tires.com®, Inc. 2021. Tire Size Calculator - Tire & Wheel Plus Sizing. Saatavissa (viitattu: 2.9.2021): <https://www.1010tires.com/Tools/Tire-Size-Calculator/215-60R17>
- [51] Afaned M. 2016. The Basics of Bluetooth Low Energy (BLE). Saatavissa (viitattu: 10.9.2021): <https://www.novelbits.io/basics-bluetooth-low-energy/>
- [52] Duque A. 2018. Deep Dive into Bluetooth LE Security. Saatavissa (viitattu: 13.9.2021): <https://medium.com/rtone-iot-security/deep-dive-into-bluetooth-le-security-d2301d640bfc>
- [53] SparkFun Electronics 2021. SparkFun RedBoard Artemis Nano. Saatavissa (viitattu: 20.9.2021): <https://www.sparkfun.com/products/15443>
- [54] ENOMA 2020. How to make equipment work without external energy? Projektin internetsivut. Saatavissa (viitattu: 5.11.2021): <https://www.energyharvesting.fi/>

LIITE A: KIIHTYVYYSANTURIN MITTAUS - OHJELMA ESIMERKKI

Alla on C-ohjelmointikielellä kirjoitettu kiihtyvyyssanturin mittaukseen käytetty RTOS-osatehtävä. Käynnistyessään se odottaa silmukassa, että yhteys isäntään saadaan muodostettua ja, että Amdtp-profiili käynnistyy. Sen jälkeen se käynnistää tarvitsemansa oheislaitteet ja siirtyy toiseen silmukkaan, missä se lipaisee AD-muuntimen mittaukset käyntiin ja ilmoittaa RTOS:lle olevansa toimitettomassa tilassa, odottamassa DMA-ohjaimen liipaisemaa keskeytystä AD-muunnoksen valmistumisen merkiksi. 14-bittiset tulokset jaetaan tämän jälkeen 8-bittisen taulukon alkioihin BLE:llä lähettämistä varten. Sitten data lähetetään käyttäen Amdtp-profiilia. Lopuksi osatehtävän käyttämät oheislaitteet ja pinnit sammutetaan ja ilmoitetaan RTOS:lle, että osatehtävä on toimiton yhden sekunnin ajan. Kun RTOS seuraavan kerran alkaa suorittamaan osatehtävää, käynnistetään oheislaitteet ja suoritetaan sama silmukka uudestaan.

```
void AccelTask(void *pvParameters)
{
    TickType_t xLastWakeTime;
    struct AccelData acceldata;
    BaseType_t xResult;
    const TickType_t x01s = pdMS_TO_TICKS(1000);
    const TickType_t x10s = pdMS_TO_TICKS(10000);
    uint32_t notification = 0;
    size_t acceldata_len = TX_PACKET_LEN;

    // Initialise the variable with the current time.
    xLastWakeTime = xTaskGetTickCount();
    power_down(); // Power down ADC, DMA, block task and sleep 1s
    wait_connection(&xLastWakeTime, x10s); // 10s loop for polling connection
                                           // with sleep state.
    power_up(); // Power up DMA, ADC

    while (1)
    {
        // Trigger adc, 900 samples at 15kHz, to start dma
        if (AM_HAL_STATUS_SUCCESS != am_hal_adc_sw_trigger(g_ADCHandle))
        {
            am_util_stdio_printf("Error - triggering the ADC failed.\r\n");
        }

        if (debug == true) am_util_stdio_printf("Enter blocked state\r\n");
        // Suspend task and wait to be notified from interrupt.
        xResult = xTaskNotifyWait(pdFALSE, // Don't clear bits on entry
                                  ULONG_MAX, // Clear all bits on exit
                                  &notification, // Return the notified value
                                  x01s); // Exit after 1000 milliseconds (ERROR)
```



```

if ((xResult == pdPASS) && ((notification & DMA_READY) != 0))
{
    if (debug == true) am_util_stdio_printf("DMA READY!\r\n");
    read_adc_samples_from_buff(&acceldata); // 14bit adc to 2x8bit for BLE
    // Send data via amdtp BLE profile to Client
    if (debug == true) am_util_stdio_printf("Send data to BLE\r\n");
    if (txState() == true) // if amdtp is idle i.e. ready to send
    {
        AmdtpsSendAccelData(&acceldata, acceldata_len);
    }
    else
    {
        am_util_stdio_printf("Amdtp state != AMDTP_STATE_TX_IDLE\r\n");
    }
}
else if ((xResult == pdPASS) && ((notification & DMA_ERROR) != 0))
{
    am_util_stdio_printf("DMA ERROR!\r\n");
}
else
{
    am_util_stdio_printf("Error - interrupt wait timeout or unknown "\
        notified value\r\n");
}

power_down(); // Power down ADC, DMA, block task and sleep 1s
if (debug == true) am_util_stdio_printf("Sleep\r\n");
vTaskDelayUntil(&xLastWakeTime, x01s);
power_up();// Power up DMA, ADC

} // while(1)
} // AccelTask()

```