

Mika Jyrälä

## **ANTURILAITTEEN OHJAUS**

## **ANTURILAITTEEN OHJAUS**

Mika Jyrälä  
Opinnäytetyö  
Syksy 2017  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma, älykkäät järjestelmät

---

Tekijä: Mika Jyrälä  
Opinnäytetyön nimi: Anturilaitteen ohjaus  
Työn ohjaaja: Kari Jyrkkä  
Työn valmistumislukukausi ja -vuosi: Syksy 2017  
Sivumäärä: 34

---

Opinnäytetyön tarkoituksena oli parantaa Thingsee-laitteperheen anturilaitteiden muokattavuutta. Tällä hetkellä anturilaitteiden toiminta määritellään, kun anturilaitteet ohjelmoidaan ja toimitetaan asiakkaalle.

Työssä toteutettiin Thingsee-anturilaitteperheelle ohjelmisto, jonka avulla pystytään muuttamaan anturilaitteiden toiminta ilman uudelleenohjelmointia. Anturilaitteiden oman ohjelmiston rinnalla toimii skriptillä toteutettu osio. Tällä tavoin anturilaitteiden toimintaa voidaan muuttaa vaihtamalla anturilaitteessa olevaa skriptiä. Skripti päivitetään ilmarajapinnan kautta, jolloin anturilaitteen uudelleenohjelmointia ei tarvitse tehdä.

Opinnäytetyön lopputuloksena saatiin toteutettua anturilaitteille ohjelmisto, jota ohjataan erillisellä PAWN-skriptillä. Skripti lähetetään anturilaitteille pilvipalvelun avulla. Sieltä skripti menee HUB-laitteelle (reitittimelle). HUB välittää skriptin määritellylle anturilaitteelle sen tunnuksen perusteella.

---

Asiasanat: Thingsee, IoT, sensorit, PAWN, Wirepas, skripti, etäohjaus

## ABSTRACT

Oulu University of Applied Sciences  
Degree programme, Intelligent Systems

---

Author: Mika Jyrälä  
Title of thesis: Sensor control  
Supervisor: Kari Jyrkkä  
Term and year when the thesis was submitted: Fall 2017  
Pages: 34

---

The aim of the thesis was to improve the modifiability of the sensors in the Thingsee family. Currently, the operation of the sensor devices is defined when the sensor devices are programmed and delivered to the customer.

In this thesis the Thingsee sensor software was updated so that it can change the operation of the sensor devices without re-programming. In addition of the sensor software there is an additional script section. Operation of the sensor devices can be changed by changing the script in the sensor device. The script is updated through the air interface, so no re-programming of the sensor is required.

As a result of the thesis, software was implemented for sensor devices, which are controlled by a separate PAWN script. The script is sent to the sensors via a cloud service. Then the script goes to the HUB (router). The HUB passes the script to the specified sensor device based on its identifier.

---

Keywords: Thingsee, IoT, sensors, PAWN, script, Wirepas, remote control

## **ALKULAUSE**

Kiitän Haltian Oy:tä mahdollisuudesta tehdä opinnäytetyö mielenkiintoisesta aiheesta IoT-tuotekehitysprojektissa. Lisäksi kiitän Haltian Oy:n Jussi Kivilinnaa ja Timo Voutilaista hyvistä neuvoista ja vinkeistä työn aikana.

7.11.2017

Mika Jyrälä

# SISÄLLYS

1 JOHDANTO	9
2 ESINEIDEN INTERNET	10
3 IOT-TIEDONSIIRTO	13
3.1 WLAN-tekniikka	13
3.2 PAN-järjestelmät	13
3.3 Matkapuhelinverkot	14
3.4 IoT-verkkojen topologiat	15
4 PÄÄTELAITTEIDEN HAASTEET	16
4.1 Virrankulutus	16
4.2 Laitteen koko	17
4.3 Huolto	18
5 KÄYTETYT LAITTEET JA TEKNOLOGIAT	19
5.1 Thingsee-anturilaitteet	19
5.2 Thingsee HUB-reititin	20
5.3 Nordic Semiconductor nRF52-piiri	20
5.4 Wirepas-verkkotekniikka	20
5.5 PAWN-skriptikieli	21
6 PAWN-SCRIPTIN TOIMINTA ANTURIJÄRJESTELMÄSSÄ	23
6.1 PAWN-skripti ja anturilaite	24
6.2 Skriptin toiminta anturilaitteessa	24
6.3 Skriptin siirto ja tallennus anturilaitteeseen	26
6.4 Thingsee HUBin toiminta	28
6.5 Pilvipalvelun toiminta	29
6.6 Skriptin testaus	29
6.7 Ohjelmiston jatkokehitys	30
7 YHTEENVETO	31

## SANASTO

AMX	PAWN Abstract Machine, PAWN-skriptin funktiokirjasto
AWS	Amazon Web Services, Amazonin pilvipalvelu
endpoint	Wirepas-tekniikan tietoliikenteen ohjausportti
IoT	Internet of Things, esineiden internet
JSON	JavaScript Object Notation, yksinkertainen tiedostomuoto tiedonvälitykseen
LAN	Local Area Network, lähiverkko
MQTT	Message Queuing Telemetry Transport, rajapinta satunnaiselle tiedonvälitykselle
PAN	Personal Area Network, henkilökohtainenverkko, määritellyille laitteille avoin verkko
PAWN	Ohjelmointikieli, jossa on keskitytty suoritusnopeuteen, vakauteen ja ohjelman yksinkertaisuuteen
POD ja TAG	Antureita sisältävä moduuli
Skripti	Komentosarjakieli
SoC	System-on-Chip, mikrokontroller, johon on integroitu oheislaitteita, kuten WLAN tai Bluetooth
Thingsee HUB	Kompakti reititin, joka lähettää mittausdatan internettiin
WAN	Wide Area Network, maantieteellisesti hajautettu tietoliikenneverkko
Wirepas	Tamperelaisen Wirepas-yhtiön kehittämä verkkoprotokolla

WLAN      Wireless Local Area Network, langaton lähiverkko



# 1 JOHDANTO

Tämä opinnäytetyö on tehty Haltian Oy:n Thingsee-laitteperheen anturilaitteille. Thingsee-laitteperhe koostuu Thingsee HUB -laitteesta ja Thingsee-anturilaitteista. Thingsee-anturilaitteissa on sensorit lämpötilan, valoisuuden, kosteuden, ilmanpaineen ja liikkeen mittaamiseen. Antureissa on myös magneettikytkin, jolla voidaan havaita mekaanista liikettä. Thingsee HUB -laite toimittaa anturilaitteista saadut tiedot asiakkaan järjestelmään internet-yhteyden kautta. (1.)

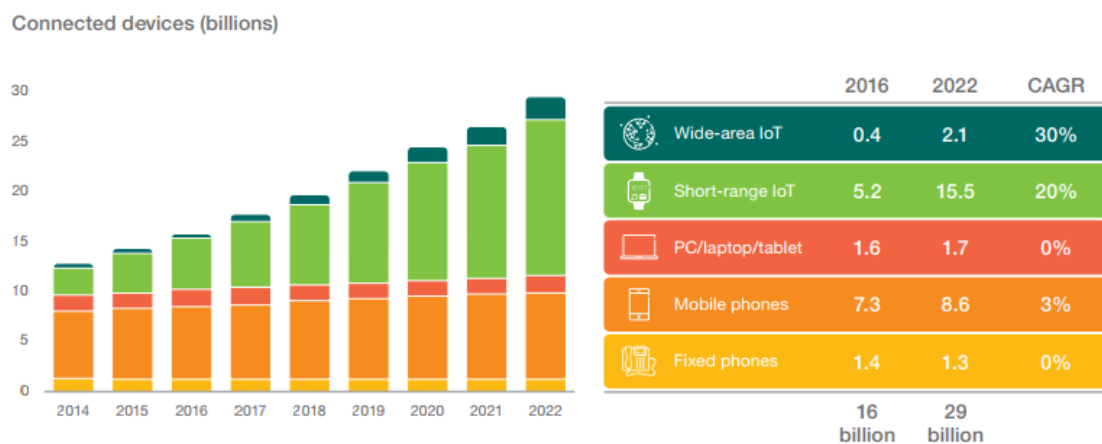
Tällä hetkellä anturilaitteiden toiminta määritellään, kun anturilaitteet ohjelmoidaan ja toimitetaan asiakkaalle. Ohjelmoinnissa määritellään, mitä eri sensoreita anturilaitteet käyttävät ja millaisella taajuudella sensoreita luetaan sekä missä vaiheessa luettuja sensoritietoja lähetetään Thingsee HUBille. Jos anturilaitteiden toimintaa halutaan muuttaa, on anturilaitteet ohjelmoitava uudelleen erillisellä ohjelmointilaitteella.

Opinnäytetyön tavoitteena on suunnitella ja toteuttaa Thingsee-anturilaitteperheelle ohjelmisto, jonka avulla pystytään muuttamaan anturilaitteiden toimintaa ilman uudelleenohjelmointia. Anturilaitteiden oman ohjelmiston rinnalla toimisi skriptillä toteutettu lohko ja tällä tavoin anturilaitteiden toimintaa voitaisiin muuttaa vaihtamalla anturilaitteessa olevaa skriptiä. Skripti päivitetäisiin ilmarajapinnan kautta, jolloin erillisellä ohjelmointilaitteella ohjelmointia ei tarvitsisi tehdä.

## 2 ESINEIDEN INTERNET

Esineiden internet tunnetaan yleisesti nimellä Internet of Things tai yksinkertaisemmin lyhenteellä IoT. IoT on eniten kasvava ala globaalissa teknologiateollisuudessa tällä hetkellä. Kaupallisten hankkeiden määrä jatkaa kasvuaan ilman merkkejä hidastumisesta. IoT tulee varmaan muuttamaan yritysten ajattelutapaa oman datan käytöstä liiketoiminnan kehittämisessä.

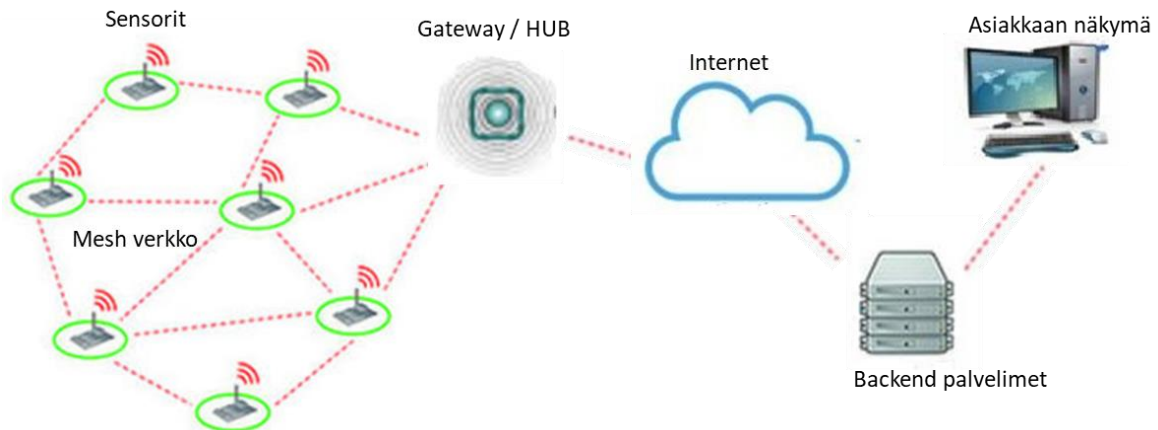
Esineiden internetissä laitteet keräävät tietoa, ovat yhteydessä toisiinsa ja jakavat keräämäänsä tiedon. Käytännössä nämä laitteet voivat olla yksittäisiä antureita, antureista koostuvia verkkoja, kodinkoneita, teollisuuden prosesseissa olevia antureita, periaatteessa mitä tahansa laitteita, jotka voivat kerätä ja lähettää tietoa internetiin käytettäväksi ja analysoitavaksi. Näiden IoT-laitteiden lukumäärän odotetaan kasvavan nykyisestä 5,2 miljardista 15,5 miljardiin vuoteen 2022 mennessä (kuva 1). Kuvassa laitteet on jaettu lyhyen kantaman ja pitkän kantaman segmentteihin. Lyhyen kantaman segmentti koostuu suurelta osin lisensioimattomista radioteknologioista, joiden tyypillinen kantomatkana on noin 100 metriä, kuten WIFI, Bluetooth ja ZigBee. Tähän luokkaan kuuluvat myös kiinteät verkkoyhteydet ja powerline-laitteet eli sähköverkossa tapahtuva tiedonsiirto. (2.)



KUVA 1. Verkkoon liitetyt laitteet vuonna 2022 (2. s. 16)

Kun dataa lähetetään IoT-laitteesta, on data lähetettävä, tallennettava ja käsiteltävä jossakin, sillä monet dataa keräävät laitteet eivät sisällä riittävästi muistia.

Useimmiten data tallennetaan pilvipalveluun, jota tietokoneet ja muut laitteet voivat käyttää internetin välityksellä. Pilvipalvelu sisältää palvelimet, tallennusjärjestelmät ja sovellukset, joita asiakkaat voidaan käyttää ja hyödyntää internetin kautta (kuva 2).



*KUVA 2. IoT verkon rakenne, anturit Mesh-verkossa*

Tyypillisiä nykypäivän IoT-laitteiden käyttäjiä ovat perinteinen teollisuus, hyvinvointiteknologian yritykset, vaateteollisuus ja autoteollisuus.

Perinteisessä teollisuudessa mitataan esimerkiksi koneiden värinää. Tästä tiedosta voidaan päätellä koneiden rikkoutuminen tai huollon tarve ja järjestelmä voi ilmoittaa vikaantumassa olevasta koneesta koneita ylläpitävälle taholle. Työkoneiden tai jalostettavien hyödykkeiden sijaintia teollisuusalueilla voidaan myös seurata.

Hyvinvointiteknologiassa käytössä on esimerkiksi aktiivisuusrannekkeet, unta seuraavat patjat, vireyttä seuraavat rannekkeet ja liikeanturit vanhuksille. On olemassa myös verenpainemittareita, jotka voivat välittää mittaustiedon suoraan terveyskeskukseen matkapuhelinverkon välityksellä.

Vaateteollisuus kehittää vaatteisiin sijoitettavia antureita, joilla seurataan liikunnan aktiivisuutta.

Autoteollisuus käyttää antureita vikojen havaitsemiseen, paikantamiseen ja kuljettajaa avustaviin järjestelmiin. Kehittyneistä autoista lähetetään antureiden mitaus- ja sijaintitietoja palvelukeskuksiin silloin, kun auto havaitsee vikaantumisen tai auto joutuu onnettomuuteen.

### **3 IOT-TIEDONSIIRTO**

Langattomat tietoliikennejärjestelmät jaotellaan tyypillisesti niiden kantaman mukaan WAN (Wide Area Network, maantieteellisesti hajautettu tietoliikenneverkko)-, LAN (Local Area Network, lähiverkko)- ja PAN (Personal Area Network, henkilökohtainenverkko, määritellyille laitteille avoin verkko) -järjestelmiin.

#### **3.1 WLAN-tekniikka**

WLAN-tekniikkaa (Wireless Local Area Network, langaton lähiverkko) käytetään yleisesti tietokoneiden liittämiseksi tietoverkkoihin. Tekniikka kuluttaa paljon virtaa, koska se on kehitetty suurten tietomäärien ja nopeiden yhteyksien tarpeisiin ja tarvitsee toimiakseen jatkuvan virransyötön. IoT-järjestelmät eivät tarvitse suuria tiedonsiirtonopeuksia eivätkä tiedonsiirtotarpeet ole isoja.

Akkukäyttöisenä WLAN-verkossa toimivat IoT-laitteet toimivat maksimissaan noin yhden viikon. WLAN-verkkoa käyttäviä IoT-laitteita voivat olla erilaiset yksittäiset anturit, jotka on kytketty verkkovirtaan ja olemassa olevaan kodin tai teollisuuden WLAN-verkkoon. Tällaisia laitteita ovat esimerkiksi kiinteästi asennettavat varashälyttimen keskusyksiköt ja kosteusanturit. WLAN-tekniikka ei näin ollen sovellu IoT-päätelaitteiden (anturilaitteiden) tietoliikenne tavaksi.

#### **3.2 PAN-järjestelmät**

Lyhyen kantaman RF-radioon pohjautuvia teknologioita on tarjolla monia eri vaihtoehtoja. Alla käydään läpi muutama yleistynyt teknologia.

##### **Bluetooth**

Bluetooth on käytössä erityisesti kuluttajille suunnatuissa laitteissa. Näitä ovat esimerkiksi matkapuhelimet ja suurin osa langattomista laitteista esimerkiksi korvakuulokkeet, aktiivisuutta mittaavat rannekkeet, yleisesti kaikki lisälaitteet joita voidaan käyttää langattomasti, vaikkapa matkapuhelimeissa. Bluetooth voidaan jakaa kahteen ryhmään, perinteiset Bluetooth-versiot 2.1, 3 ja Bluetooth Low Energy -versiot (BLE) 4.0, 4.1, 4.2 ja 5.0. BLE on kehitetty vähän virtaa kuluttavaksi järjestelmäksi, jossa tietoa voidaan lähettää pieniä määriä kerrallaan. BLE-

tekniikalla toteutetut anturilaitteet voivat toimia pelkällä paristolla useita vuosia. BLE tukee myös lisästandardin avulla Mesh-verkkoprotokollaa. Mesh mahdollistaa anturiverkkojen muodostamisen BLE-tekniikalla. (3.)

Haltian käyttää osassa IoT-sovelluksista BLE-tekniikkaan perustuvaa Wirepasin kehittämää Wirepas-tiedonsiirtotekniikkaa. Wirepas toimii BLE-tekniikan päällä ja tarjoaa helpon käyttöönoton ja verkon skaalautuvuuden. (4.)

## **ZigBee**

ZigBee-tekniikka on suunniteltu erityisesti vähän virtaa kuluttavaksi järjestelmäksi. Käyttökohteita ovat esimerkiksi kotien valaistuksen automatisointi ja teollisuuden mittalaitteet. ZigBee-verkossa on erikseen ZigBee-koordinaattori, joka pitää sisällään verkon hallintatiedot. ZigBee-reititin välittää tietoja ZigBee-verkon ulkopuolelle. ZigBee-päätelaitteet ovat antureita sisältäviä laiteita. Päätelaitteista mitattu tieto välittyy reitittimelle. ZigBeelaitteita voidaan myös yhdistää yhdeksi laitteeksi. Esimerkiksi koordinaattori ja reititin voivat olla yksi ja sama laite. (5.)

## **NFC**

NFC (Near Field Communications) on erittäin lyhyen kantaman tekniikka. Järjestelmä koostuu päätelaitteesta, jolla voidaan kirjoittaa tai lukea erillistä NFC-tagia. NFC-tagia ei itsessään sisällä virtalähdettä, vaan se saa käyttöjännitteen päätelaitteesta sähkömagneettisen induktion avulla. Induktiosilmukka toimii samaan tekniikalla kuin sähköhammasharja latautuu. Virransyöttö ja tiedonsiirto toimivat, kun NFC-tagia ja päätelaite ovat tarpeeksi lähelle toisiaan. Välimatka voi olla enintään muutamia senttimetrejä. Tekniikkaa käytetään esimerkiksi sähkötoimisissa lukoissa, kauppojen sähköisissä hintalapuissa ja etäluettavissa maksukorteissa.

### **3.3 Matkapuhelinverkot**

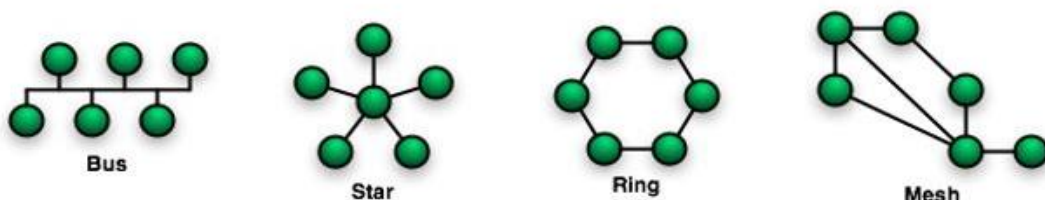
IoT-laitteet voivat käyttää matkapuhelinverkkoa suurten datamäärien siirtoon. Ongelmana on, että matkapuhelinverkon käyttö tiedonsiirtoon kuluttaa laitteiden virtalähdettä huomattavan paljon. Matkapuhelinverkko onkin käyttökelpoinen IoT-verkon reitittimien tiedonsiirtotekniikkana. Reitittimet keräävät mittausdatan

antureilta, kokoavat datan isommiksi kokonaisuuksiksi ja välittävät tiedon hallitusti pilvipalvelun taustajärjestelmään jatkokäsittelyä varten.

Matkapuhelinverkon 4G-standardia ollaan päivittämässä vastaamaan paremmin IoT-laitteiden tarpeita. Narrowband IoT eli lyhyemmin NB-IoT on tulossa 4G-verkkoihin. Tämän teknologian kerrotaan sopivan erityisesti IoT-järjestelmille. NB-IoT-tekniikkaa käyttävien IoT-päätelaitteiden etuja ovat esimerkiksi pitkä akunkesto, edullinen toteutus, helppo käyttöönotto, laaja peittoalue (4G verkon kattavuus) ja se, että tekniikka tukee suuria määriä IoT-laitteita. (6.)

### 3.4 IoT-verkkojen topologiat

Verkkotopologiat on jaettu perinteisesti kolmeen kategoriaan väylä-, tähti- ja rengasverkkoihin. Näiden lisäksi langattomien IoT-verkkojen yleistymisen johdosta on kehitetty sensoriverkkoihin paremmin soveltuva Mesh-topologia (kuva 3). Halitan käyttää sensorilaitteissaan Mesh-tekniikkaan pohjautuvaa tiedonsiirtoa. Mesh-verkossa kaikki laitteet (reititin, koordinaattori ja päätelaitteet) kykenevät olemaan yhteydessä suoraan toisiinsa. Tämä lisää Mesh-verkon vikasietoisuutta, koska yhden laitteen vikaantuminen ei estä muita laitteita kommunikoimasta keskenään ja näin verkon laitteiden tietoliikenteelle löytyy vaihtoehtoisia reittejä. Etuina on myös verkon peittoalueen kasvattaminen. Normaalisti yhden radiolinkin kantama on 10–100 m. Mesh-tekniikalla tuota kantamaa saadaan kasvatettua huomattavasti suuremmaksi jopa useisiin kilometreihin. Mesh-verkon haittapuolena voidaan pitää verkon viiveiden kasvamista koska kaukaisimman päätelaitteen mittaustieto pitää reitittää useiden solmujen kautta reitittimelle. (7.)



KUVA 3. Tietoliikenteessä käytetyt verkkotopologiat (8.)

## 4 PÄÄTELAITTEIDEN HAASTEET

Päätelaitteet ovat sensoriverkkojen dataa tuottava osa. Niiden suunnittelussa pitää ottaa huomioon monia eri seikkoja kuten virrankulutus, laitteen koko, käyttöolosuhteet sekä huollon ja päivitysten tarve. Mitä varmemmin laitteet toimivat ja mitä kauemmin niiden virta riittää, sitä enemmän sensoriverkosta on hyötyä järjestelmän käyttäjälle.

### 4.1 Virrankulutus

Virrankulutus on olennainen osa anturilaitteiden suunnittelua. Jos virrankulutusta ei saada pieneksi vaikuttaa se koko anturilaitteiden elinkaareen. Virrankulutukseen voidaan vaikuttaa lähetin- ja vastaanotintekniikan valinnassa ja määriteltäessä, millaisella aikavälillä päätelaitteet lukevat sensoreista mittaustietoja ja miten tiheään ne välittävät keräämänsä mittaustiedot eteenpäin. Jos anturilaitteen toiminta perustuu jonkin sensorin jatkuvaan lukemiseen, voi se aiheuttama virrankulutus nousta kokonaisuuden kannalta suureksi.

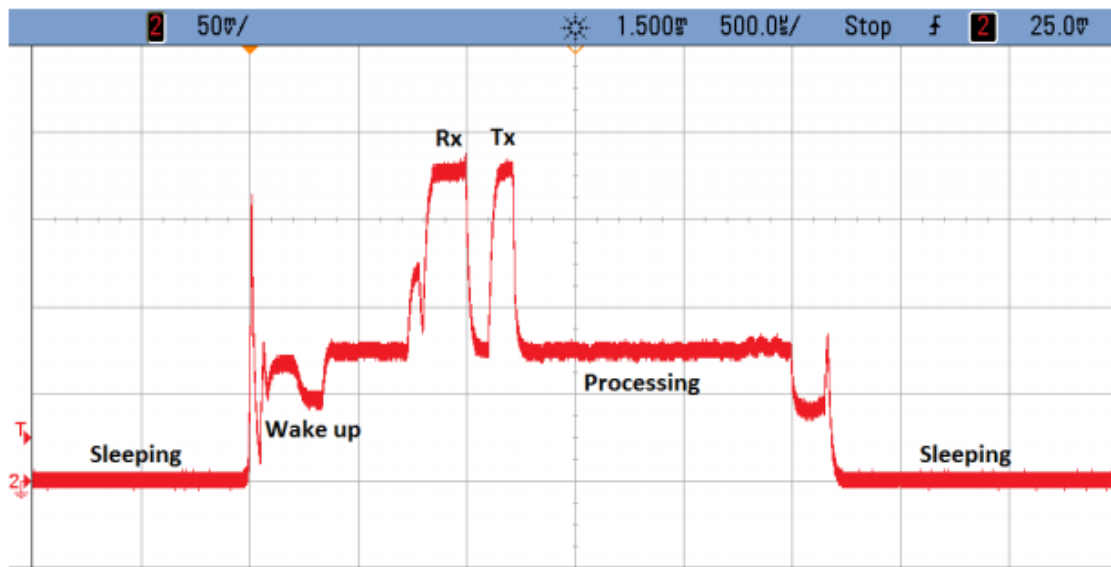
IoT tiedonsiirtokappaleessa vertailtiin eri tiedonsiirtotapoja IoT-laitteissa. WLAN-tekniikka ei sovellu paristo- tai akkukäyttöisiin laitteisiin suuren virrankulutuksen johdosta. Nykyiseen matkapuhelinverkkoon perustuva tiedonsiirto kuluttaa paljon virtaa. Tiedonsiirron ollessa käynnissä tyypillinen virta on n. 400 mA ja modeemin yhdistyessä matkapuhelinverkkoon maksimivirtapiikki voi olla jopa 2 A. (9.) Tähän virta ongelmaan on luvassa muutos, kun NB-IoT laajennus tulee yleiseen käyttöön ja laitevalmistajilta tulee edullisia tätä tekniikkaa käyttäviä modeemeja (6). Isoissa sensoriverkoissa, joissa voi olla satoja anturilaitteita matkapuhelinverkkoon pohjautuvat anturilaitteen eivät välttämättä ole järkeviä, vaan silloin voisi olla järkevää kerätä anturilaitteiden mittaustieto reitittimelle ja siirtää reitittimen kautta hallitusti pilvipalveluihin.

Koska Haltian käyttää BLE:hen pohjautuvaa tekniikkaa, tarkastellaan BLE:n virrankulutusta tarkemmin. Tarkastelu osoittaa, että BLE:llä toteutettu sensoriverkkoon voidaan toteuttaa tiedonsiirto hyvin pienellä virralla. Esimerkiksi Texas Instrumentsin BLE-piirin CC2541 virrankulutus on Texas Instrumentsin tekemien mitausten perusteella todella pieni, noin 0,024 mA (10). Kyseinen virrankulutus on



laskettu seuraavilla arvoilla, lähetys sekvenssi toistetaan kerran sekunnissa, lähetyspulssin kesto on 2,7 ms ja pulssin aikana virtaa kuluu noin 8,53 mA. Koska BLE-piiri on lepotilassa suurimman osan ajasta, saadaan kokonaisvirrankulutus pidettyä pienenä.

Näin BLE-tekniikalla pystytään toteuttamaan todella pienivirtaisia sensoriverkkoja. Kuvan 4 oskilloskooppi mittauksesta nähdään BLE:n suhteelliset virrankulutukset. Kuvassa voi todeta, että yksi aikajakso muodostuu vastaanotto- ja lähetyspurskeesta (Rx/Tx) ja loput osat ovat kättely- ja valmistelujaksoja. Yhden purskeen kesto aika on noin 2,7 ms.



KUVA 4. BLE-vastaanotto- ja lähetyspurske (10.)

Texas Instrumentsin laskelmien mukaan tiedonsiirto toimisi CR2032-paristolla reilun vuoden. CR3032-pariston kapasiteetti on 230 mAh ja BLE:n virrankulutus on 0,024 mA. Tästä on laskettu, että pelkkä BLE-modeemin käyttö onnistuisi paristolla noin 400 päivää. (10.) Pitää muistaa, että tässä laskelmassa on mukana vain BLE-modeemin virrankulutus. Anturilaitteessa virtaa kuluttaa myös mikrokontrolleri ja siihen kytketyt sensorit.

## 4.2 Laitteen koko

Anturilaitteen koon määrittelee suurimmaksi osaksi siinä käytetty virtalähde tai paristo. Esimerkkinä voidaan käyttää Haltianin Thingsee POD- ja TAG-anturilaitteita. POD-anturilaitteessa on virtalähteenä 1500 mAh:n paristo, jonka toiminta

aika on jopa 3 vuotta. POD-anturilaitteen koko on 40 x 30 x 41 mm. TAG-anturi on pienempi ja siinä on virtalähteenä 225 mAh:n paristo, jonka toiminta-aika on noin puoli vuotta. Sen koko on huomattavasti pienempi kuin POD-anturilaitteen eli 30 x 30 x 6 mm. Kummassakin anturilaitteessa on samat toiminnallisuudet, joten anturilaitteiden koon määrittelee niissä käytetyt virtalähteet. (4; 11.)

### **4.3 Huolto**

Anturilaitteiden huoltovälin määrittelee suurimmaksi osaksi niiden virtalähteen kesto. Mitä suurempi kapasiteetti anturilaitteen virtalähteessä on, sitä kauemmin anturilaitte voi olla toiminnassa ilman pariston vaihtoa. Jos laitteessa on Over the air (OTA)-ohjelmointiominaisuus ja tarpeeksi muistia uudelleenohjelmointiin, voidaan koko ohjelmisto päivittää ilmaitse. Pitää muistaa, että koko ohjelmiston päivittäminen vaatii tiedonsiirtokaistalta kapasiteettia. Päivitettävän ohjelmiston koko voi olla jopa megatavuja ja laitteita kymmeniä tai satoja. Näin suuri tiedonsiirron tarve vie paljon aikaa ja anturiverkon tietoliikennekapasiteettia. Edellä mainitut seikat voivat vaikuttaa haitallisesti anturiverkon pääasialliseen tehtävään eli mittaustietojen välitykseen. Pitkäaikainen radiomodeemin päällä olo kuluttaa suhteellisesti paljon anturilaitteen pariston virtaa. Virrankulutusta voidaan pienentää tekemällä ohjelmisto päivitys skriptiosioon koko ohjelmiston sijaan. Jos OTA-päivitystä ei voida tehdä tai se ei ole järkevää toteuttaa, pitää ohjelmistopäivitys tehdä fyysisesti anturilaitteiden luona. Tällainen toiminta on kuitenkin tehotonta ja voi tulla kalliiksi, jos antureita on isolla alueella paljon.

## 5 KÄYTETYT LAITTEET JA TEKNOLOGIAT

Haltianin anturilaitteissa käytetään yleisesti käytössä olevia teknologioita. Ne on kehitetty Nordic Semiconductorin nRF52-alustalle. Anturilaitteiden ja HUB-laitteen välinen tiedonsiirto on toteutettu Wirepas-teknologialla. Anturilaitteet muodostavat Wirepas-teknologialla toteutetun Mesh-verkon. Verkossa anturilaitteet toimivat solmukohtina ja välittävät verkon mittaustietoja toisten anturilaitteiden kautta verkossa olevalle HUB-laiteelle. HUB-laite välittää verkon mittaustiedot internetissä olevalle pilvipalvelulle, josta loppukäyttäjä voi hakea ne käyttöönsä.

### 5.1 Thingsee-anturilaitteet

Thingsee POD- ja TAG-anturilaitteita (kuva 5) voidaan käyttää lämpötilan, kirkkauden, kosteuden, ilmanpaineen ja liikkeen mittaamiseen. Lisäksi niissä on magneettikytkin, joka reagoi magneettikenttään. Magneettikytkimen avulla on mahdollista seurata, onko ovi tai ikkuna avoin vai suljettu. Anturilaitteita voidaan myös käyttää varojen, tavaroiden ja ihmisten seurantaan ja valvontaan aivan uudella kustannustehokkaalla tavalla.



*KUVA 5. Thingsee-anturilaitteet (1.)*

Thingsee POD-anturilaitte on pienitehoinen ja kooltaan pienempi kuin golfpallo. Lisäksi sen käyttöikä on jopa useita vuosia. TAG-anturilaitte on kolikon kokoinen ja sopii hyvin sovelluksiin, jotka vaativat erityisen pienikokoista anturilaitetta. (1.)

Anturilaitteiden nRF52-kontrollerissa Wirepas-alustaan perustuva tiedonsiirto tekniikka toimii node-moodissa. Tällöin anturilaitteet toimivat päätelaitteina ja samalla ne pystyvät siirtämään muiden node-laitteiden mittaustietoa toisilleen. Näin ne muodostavat yhdessä Mesh-verkon.

## **5.2 Thingsee HUB-reititin**

Thingsee HUB on anturiverkossa oleva mittaustiedon välityslaite. HUB siirtää verkossa olevien anturilaitteiden mittaustiedon internetin pilvipalveluun. Pilvipalvelu kokoaa datan ja muokkaa sen palvelun käyttäjälle sopivaan esitysmuotoon. HUB-laite sisältää nRF52-alustan jonka avulla se on yhteydessä anturilaitteisiin Wirepas-verkon välityksellä. HUB-laitteessa on myös STM32-sarjan prosessori, jota käytetään NUTTX-reaaliaikajärjestelmällä. NUTTX on Linux-pohjainen reaaliaikakäyttöjärjestelmä. NUTTX:n päällä toimii palvelu, joka toteuttaa tietoliikenneyhteyden pilvipalveluun matkapuhelin- tai WLAN-verkon avulla.

## **5.3 Nordic Semiconductor nRF52-piiri**

nRF52-alusta on tehokas, erittäin pienitehoinen SoC-piiri. Se sopii erinomaisesti Bluetooth Low Energy-, ANT- ja 2,4 GHz:n RF-radiota käyttäville sovelluksille. nRF52-alusta on rakennettu 32-bittisen ARM-Cortex-prosessorin ympärille, jossa on 512 kB flash-muistia ja 64 kB RAM-muistia. nRF52-alusta on yhteensopiva Nordic Semiconductorin nRF51-, nRF24L- ja nRF24AP-sarjan tuotteiden kanssa. (12.)

## **5.4 Wirepas-verkkotekniikka**

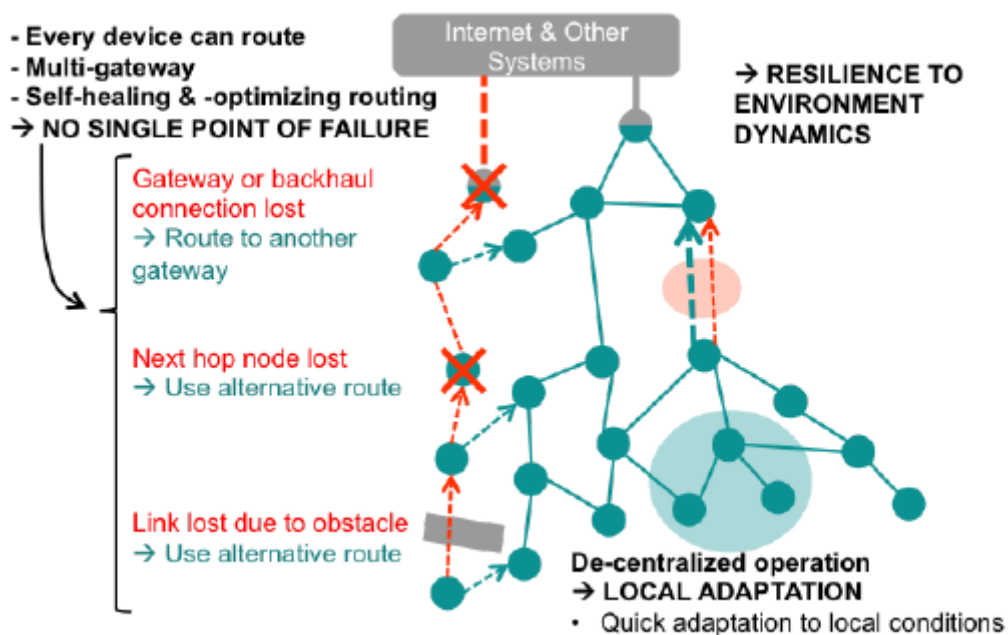
Wirepas Oy tekee optimoituja ratkaisuja kaikille IoT-sovellukselle. Koska eri IoT-laitteiden vaatimat yhteydet vaihtelevat, ei kaikille eri sovelluksille sovi samanlainen yhteys. Tähän ongelmaan Wirepas tarjoaa omia tietoliikennearatkaisuja. Wirepas-verkon muokattavuuden avulla voidaan optimoida mm. tehoa, kaistanleveyttä, toiminta-aluetta ja latenssia. (13.)

Wirepas-verkon älykkyys on verkossa laitteissa. Laitteet päättävät parhaiten itse verkon asetukset ja näin verkko ei tarvitse keskitettyä verkonhallintaa. Laitteiden

paikallinen päätöksenteko varmistaa, että ne toimivat aina samalla tavoin riippumatta verkon koosta tai verkon laitteiden sijainnista. (13.)

Laitteet valitsevat automaattisesti oman tehtävänsä verkon tilanteen mukaan. Tämä tarkoittaa sitä, että jokainen laite voi välittää verkossa liikkuvaa tietoa toisille verkon laitteille. Käyttäjän ei tarvitse määrittellä laitteiden rooleja. Roolien määrittely tapahtuu automaattisesti sen mukaan, mikä on verkon optimaalinen topologia tiettyä ajankohtana. (13.)

Mesh-verkossa laitteet voivat siirtää tietoja useiden eri reittien kautta. Verkon topologiaa optimoidaan jatkuvasti. Näin verkko mukautuu ympäristön ja verkon muutoksiin (kuva 6). Jokaisella verkon laitteella on useita reititysvaihtoehtoja, joita laitteet voivat käyttää mittaustietojen siirtoon. (13.)



KUVA 6. Wirepas-topologia (14, s. 12)

## 5.5 PAWN-skriptikieli

PAWN-skriptikieli on 32-bittinen ohjelmointikieli, jolla on C-kielen tyyppinen ulkoasu (15). Suoritusnopeus, vakaus, yksinkertaisuus ja pieni koodin tarve ovat olleet olennaisia suunnitteluperusteita, kun PAWN-ohjelmointikieltä on kehitetty.

PAWN-pääohjelma on skriptiä suorittava ohjelma. Se suorittaa skriptiä Abstract Machinen (AMX) avulla. Abstract Machine sisältää C-kielen funktiokirjastot ja toimii linkkinä varsinaisen C-ohjelman ja skriptin välillä. PAWN IDE -kääntäjällä muunnetaan kirjoitettu skripti Abstract Machinen ymmärtämään muotoon. PAWN IDE tuottaa P-koodia eli ajettavaa skriptiä. (15.)

Alla on esimerkkikoodi PAWN-skriptistä (kuva 7). Skriptillä voidaan tulostaa konsolille celsius- ja fahrenheit-asteet väliltä 0°– 3° 0,5°:n välein (kuva 8). Skripti voidaan suorittaa PAWN IDE -ympäristössä, jolloin koodilistauksesta nähdään, että skripti on C-koodin tyyppistä. Koodi poikkeaa C-koodista siten, että rivit eivät esimerkiksi lopu puolipisteeseen ja muuttujia alustettaessa niille ei anneta tyyppiä.

```
#include <rational>

main()
{
  new Rational: Celsius
  new Rational: Fahrenheit

  print "Celsius\t\t Fahrenheit\n"
  for (Celsius = 0; Celsius <= 3; Celsius=Celsius +0.50)
  {
    Fahrenheit = (Celsius * 1.8) + 32
    printf "%2.2r \t\t %2.2r\n", Celsius, Fahrenheit
  }
}
```

*KUVA 7. PAWN-skriptin esimerkkikoodi*

Celsius	Fahrenheit
0.00	32.00
0.50	32.90
1.00	33.80
1.50	34.70
2.00	35.60
2.50	36.50
3.00	37.40

*KUVA 8. PAWN-skriptin tuottama näkymä konsolille*

## 6 PAWN-SCRIPTIN TOIMINTA ANTURIJÄRJESTELMÄSSÄ

Anturilaitteiden ohjausmekanismia miettiessä lopullisiksi vaihtoehtoiksi jäivät taulukkomuotoinen ohjaus tai skriptiohjaus. Wirepas-ilmarajapinnan kannalta on sama, käytetäänkö taulukko- vai skriptiohjausta anturilaitteiden ohjauksessa. Kummassakin tapauksessa binääritiedosto täytyy siirtää ilmateitse anturilaitteen muistiin. Tiedoston siirto toimii samalla tavalla kummassakin ohjaustavassa.

Pohdittaessa, mitä toimintoja anturilaitteissa pitää pystyä ohjaamaan ja millaista logiikkaa ohjauksessa olisi hyvä olla, alkoi näyttämään siltä, että taulukkomuotoinen ohjaustapa ei ole toimiva. Lisäksi anturilaitteiden loppukäyttäjien eli asiakkaiden konfiguraatioiveet eivät ole tiedossa. Jos ohjaustavaksi valittaisiin taulukko, pitäisi taulukon rakennetta muuttaa aina, kun uusia ominaisuuksia halutaan anturilaitteen ohjaukseen. Vastaavasti myös anturilaitteen pääohjelmaa täytyisi muokata vastaanottamaan uusi ohjaustaulukko. Tilanne olisi sama kuin nykyisin eli anturilaitteisiin pitäisi tehdä jatkuvasti uusia ohjelmistoja ja taulukoita. Ohjelmistojen päivittäminen käytössä oleviin anturilaitteisiin pitäisi tehdä ne uudelleen ohjelmoimalla.

Anturilaitteen ohjaukseen olisi hyvä saada yksinkertaista logiikkaa, jota taulukko-ohjaus ei pysty tarjoamaan. Kiihtyvyyssanturilta saadaan yli 0,5 g:n hälytyssignaali ja tämän hälytyksen seurauksena lähetetään anturilaitteelta lämpötila- ja valomääräarvot asiakkaan järjestelmään. Jos asiakas haluaa lisätä myöhemmin kosteustiedon lähetyksen, on sen tekeminen mahdotonta taulukko-ohjauksella.

Skriptiohjaus on joustava tapa tehdä anturilaitteiden sensorien mittausarvoista päätelmiä, koska arvojen vertailut ja laskelmat tehdään kokonaisuudessaan skriptissä. Näiden mietteiden pohjalta päätettiin valita anturilaitteiden ohjaustavaksi skriptiohjaus.

## 6.1 PAWN-skripti ja anturilaite

Haltianin anturilaitteet käyttävät liikennöintiin Wirepas-teknologiaa. Tämä asettaa anturilaitteiden muistille ja ilmarajapinnan tietopaketeille rajoitteita. Wirepas-teknologia käyttää merkittävän osan anturilaitteen nRF52-kontrollerin resursseista. Tämä asettaa rajoituksia ohjauksen toteutukseen anturilaitteiden ohjelmistossa.

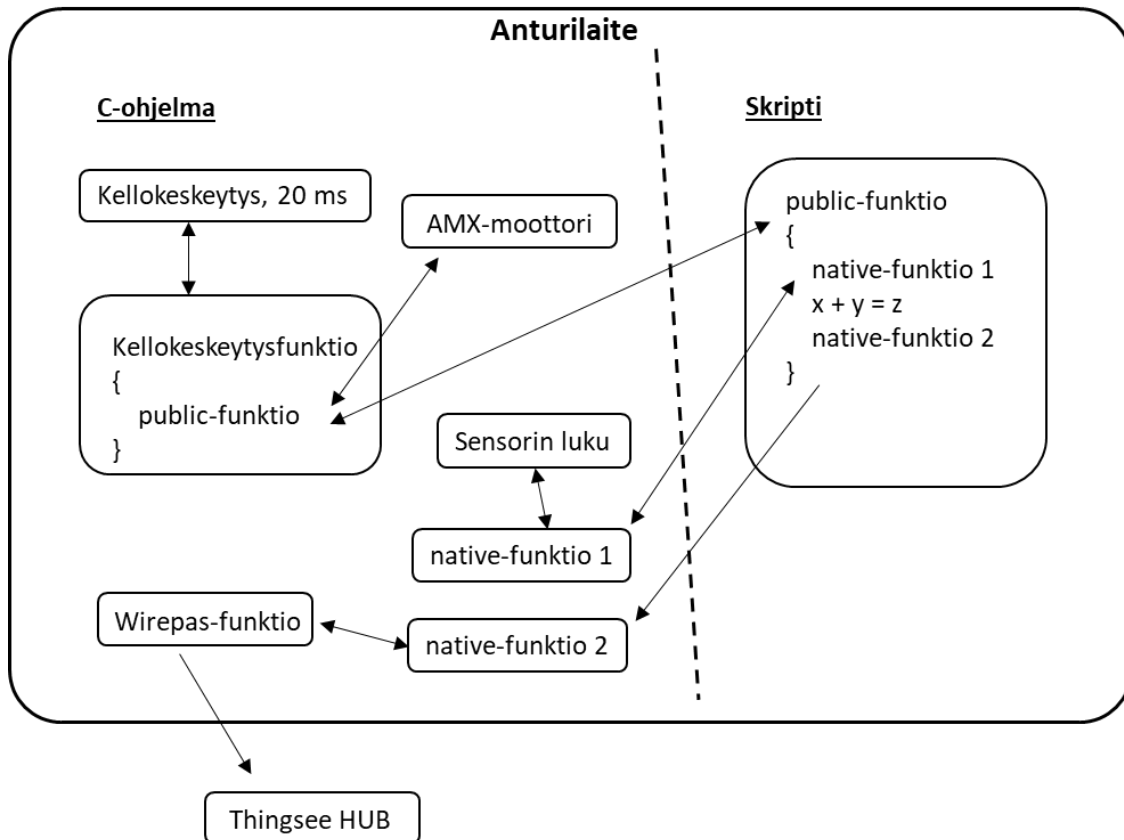
Anturilaitteiden ohjauskomennot jaetaan nRF52-perusohjelmiston ja PAWN-skriptin kesken. Perusohjelmassa on omat ohjelmafunktiot sensoreiden mittaustietojen lukemiselle, mittaustiedon lähetykselle ja vastaanotolle Wirepas-verkkoon. Skripti osaa kysyä perusohjelmalta sensorien mittaustietoja ja antaa näiden tietojen perusteella komentoja perusohjelmalle. Skripti voi lisäksi pyytää lisämittauksia tai käskä perusohjelmaa lähettämään mittaustiedot Wirepas-verkkoon. Näin saadaan anturilaitteen mittauksiin logiikkaa, jolla voidaan tehdä edellä kuvattua kaltaista ohjausta. Jos kiihtyvyyssanturilta saadaan yli 0,5 g:n mittaustieto, voidaan tämän mittauksen perusteella kysyä lämpötila, valomäärä ja kosteusarvot ja lähettää ne ja kiihtyvyyssiedot eteenpäin. Skriptin raja-arvoja ja logiikkaa muuttamalla saadaan anturilaitteen toiminta muuttumaan pelkällä skriptin vaihdolla anturilaitteeseen.

## 6.2 Skriptin toiminta anturilaitteessa

PAWN AMX-moottori on integroitu anturilaitteen C-ohjelmaan. Näin skripti voidaan suorittaa C-ohjelmassa AMX-moottorilla esimerkiksi HW-keskeytyksellä tai kellokeskeytyksellä. PAWN-skriptin funktiokirjastossa määritetyillä public-funktioilla toteutetaan skriptin toiminnallisuus. Public-funktioiden avulla tehdään kaikki mittaustietojen vertailut ja päätelmät skriptin sisällä. Skriptin native-funktioilla välitetään tietoja pääohjelman ja skriptin välillä. Public-funktioiden sisällä voidaan käyttää native-funktioita. Näin skripti voi kysyä mittaustietoja pääohjelman puolelta ja tehdä vertailuja sekä ohjauksia pääohjelmaan mittaustulosten perusteella. Kuvassa 9 on esitetty skriptin toiminta anturilaitteessa. Kellokeskeytys ohjaa kellokeskeytyksfunktiota 20 ms:n välein. Kellokeskeytyksfunktion public-funktio suorittaa skriptissä olevan public-funktion. Skriptin public-funktio kysyy native-funktio 1:n avulla C-puolen native-funktio 1:ltä sensorin mittaustietoa. Mittaustietoa perusteella skripti tekee päätelmiä ja tarvittaessa ohjaa native-funktio 2:lla C-



puolen native-funktio 2:sta. C-puolen native-funktio 2 lähettää mittaustiedon Wi-repas-verkon välityksellä HUB-laitteelle (kuva 9).



KUVA 9. PAWN-skriptin toiminta anturilaitteessa

Anturilaitteen sensoreista saadaan seuraavia arvoja: etäisyys, kiihtyvyyssanturin tiedot (x-, y- ja z -akseli) ja akun varaustaso (taulukko 1). Nämä mittausarvot pyydetään native-funktioilla skriptille käsiteltäväksi.

TAULUKKO 1. Anturilaitteen event-viestit

Event-viesti	Selite
EVENT_DISTANCE	etäisyys, mm
EVENT_TILT	kiihtyvyys (x, y, z), g
EVENT_BATT_CAPACITY	akun varaustila, %

Taulukossa 2 on native-funktiot, joita skripti käyttää mittaustietojen hakuun pääohjelmalta ja joilla skripti ohjaa pääohjelman funktioita.

## TAULUKKO 2. Skriptin native-funktiot

Native-funktio	Selite
PAWNinit_native	PAWN-skriptin alkumäärittelyt
Getdistance_native	etäisyyden haku skriptille
GetTilt_native	liikeanturin tilojen haku skriptille
GetTime_native	ajan haku skriptille
Sendreport_native	tietojen lähetys Wirepas-verkkoon
Getbatterylevel_native	akun varaustilan haku skriptille
SendScriptInfo_native	skriptinversion lähetys Wirepas-verkkoon

AMX-moottoria käytetään kellokeskeytysfunktion avulla. Kellokeskeytys alustetaan ensimmäisellä AMX-moottorin kierroksella, jolloin suoritetaan PAWNinit\_native-funktio. Skripti määrittelee kellokeskeytysfunktion keskeytysajan. Tämä aika on AMX-moottorin ja samalla skriptin suorituksen kellokeskeytysaika.

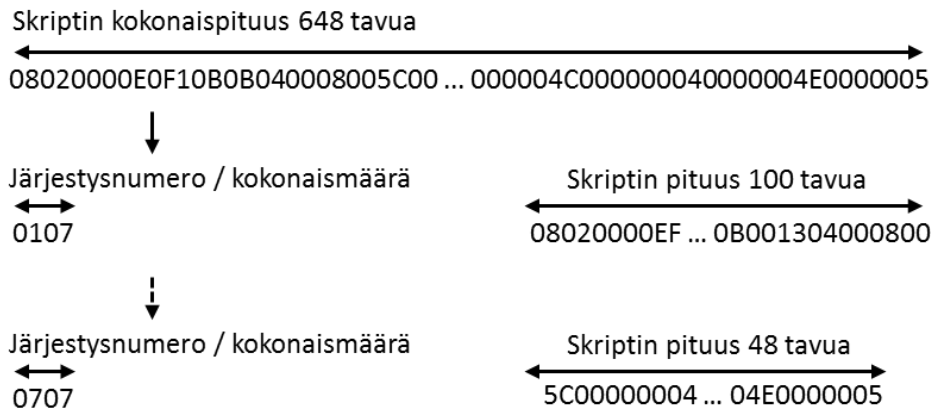
Kun anturilaitte käynnistyy, kellokeskeytys asetetaan PAWNinit\_native-funktiolla 10 millisekuntiin. Näin skriptiä ajetaan 100 Hz:n taajuudella. Tämän jälkeen skripti kysyy Getdistance\_native-funktion avulla pääohjelmalta etäisyysarvon. Skripti tarkistaa, että etäisyys on määritellyn mittausalueen sisällä, joka on 50–150 millimetriä. Tämän jälkeen skripti laskee saaduista etäisyysarvoista keskiarvon jokaiselle 10 mittaukselle. Näin saadaan laskettu etäisyyden keskiarvo 100 millisekunnin välein. Tämän jälkeen voidaan tarkistaa, onko keskiarvo hälytysraja-arvojen sisällä. Raja-arvot voivat olla esimerkiksi 100–120 millimetriä. Jos tulos on hälytysalueella, lähettää skripti Sendreport\_native-funktiolla mittausarvon pääohjelmalle. Pääohjelma ottaa vastaan mittausarvon n\_Sendreport\_native-funktiossa ja lähettää tiedon pilvipalvelulle Wirepas-verkon välityksellä.

### 6.3 Skriptin siirto ja tallennus anturilaitteeseen

Skriptin siirtäminen anturilaitteeseen toteutettiin käyttäen Wirepas-verkon tietopakettien siirto-ominaisuutta. Tietopakettien maksimikoko Wirepas-verkossa on 102 tavua. Lähetettävän skriptin koko on aina yli 102 tavua, joten skripti pitää ennen lähetystä jakaa 100 tavun osiin (kuva 10). Koska skripti jaetaan osiin, vastaanotossa tiedettävä, mikä osa skriptistä on mikäkin vastaanotettu osa. Ongelman ratkaisu oli lisätä jokaiseen lähetettävään skriptin osaan järjestysnumero

ja lähetettävien pakettien kokonaismäärä. Lähetettävä tietopaketti koostuu seuraavista osista: ensimmäinen tavu on paketin järjestysnumero, toinen tavu on pakettien yhteismäärä ja tavut 3–102 sisältävät skriptitietoa 100 tavua.

Esimerkki skriptin kokonaispituus on 648 tavua. Esimerkissä skripti on esitetty HEX-muodossa. Skripti jaetaan ennen anturilaitteelle lähettämistä 100 tavun pituisiin osiin. Koska skriptin pituus on 648 tavua, saadaan siitä kuusi 100 tavun osaa ja yksi 48 tavun osa. Jokaisen osan alkuun lisätään osan järjestysnumero ja osien kokonaismäärä.



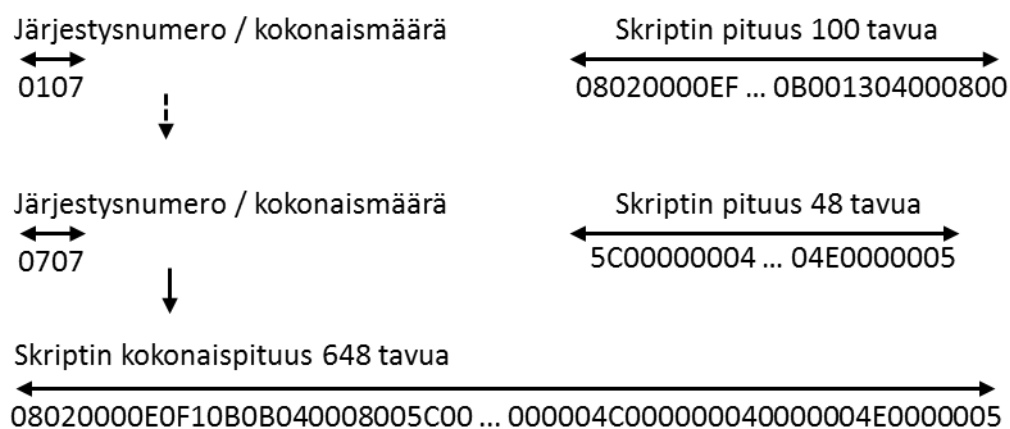
KUVA 10. Skriptin jakaminen osiin.

Skriptin vastaanotto ja tallennus anturilaitteessa käyttöä varten toteutetaan anturilaitteen pääohjelmassa. Skriptin vastaanottoon käytettiin Wirepas-verkon tarjoamia funktiota. Funktion avulla vastaanotetaan Wirepas-verkosta tulevia tietopaketteja. Kun anturilaitte vastaan ottaa tietopaketin, vastaanottofunktiossa tarkistetaan, että vastaanotettu tietopaketti on skriptitietoa. Tämä tapahtuu Wirepas-tietopakettien sisältämien määritysten avulla. Tietopaketti sisältää vastaanotetun tiedon koon tavuina, lähde- ja kohde-endpointit, kohde päätelaitteen osoitteen ja lähettäjän osoitteen. Endpointja voidaan ajatella eräänlaisina tietoportteina. Näin voidaan siirtää eri tarkoituksiin tarkoitettua tietoa ennalta määrättyihin portteihin. Skriptin osille määriteltiin oma endpoint-numero, jonka avulla siirretään vain skriptitietoa. Näin varmistetaan se, että skriptin vastaanottofunktiioon ei tule muuta kuin skriptitietoa.

Aina, kun skriptin osa on otettu vastaan, tallennetaan se RAM-muistiin. Kun kaikki skriptin osat on otettu vastaan (kuva 11), tehdään RAM-muistin synkronointi flash-muistiin. Näin varmistetaan, ettei flash-muistiin kopioida vaillinaista skriptia.

Skripti on tallennettuna flash-muistissa päätelaitteen uudelleenkäynnistämisen aikana. Jos skripti olisi vain RAM-muistissa, se katoaisi jokaisella anturilaitteen uudelleen käynnistämisellä. Kun anturilaitte käynnistetään, kopioidaan skripti aina flash-muistista RAM-muistiin.

Skriptin kokoaminen anturilaitteessa tapahtuu yhdistämällä saapuneet osa yhdeksi kokonaisuudeksi. 100 tavun mittaiset skriptin osat tallennetaan RAM muistiin. Tallennus järjestyksen määrää vastaanotettujen osien järjestysnumero. Näin skripti saadaan rakennettua oikein ja se voidaan ottaa käyttöön anturilaitteessa. Ennen käyttöönottoa tarkistetaan, että kaikki skriptin osat ovat saatu otettua vastaan.



KUVA 11. Skriptin kokoaminen

Skripti voidaan suorittaa flash- tai RAM-muistista. Jos skripti suoritetaan flash-muistista, pitää se lukea flash-muistista joka suorituskerta. Tällainen flash-muistin lukeminen kuluttaa enemmän virtaa ja on hitaampaa verrattuna siihen, jos skripti suoritettaisiin RAM-muistista. Parempi tapa skriptin suorittamiseen on suorittaa se RAM-muistista nopeuden ja virran kulutuksen vuoksi.

#### 6.4 Thingsee HUBin toiminta

Thingsee HUB toimii reitittimenä pilvipalvelun ja Wirepas-anturiverkon välillä. Anturilaitteille menevä skripti välitetään pilvipalvelun kautta HUB-laiteelle. Pilvipalvelusta tuleva skriptiviesti sisältää anturilaitteen osoitteen, jonka avulla HUB osaa välittää tulevan skriptin oikealle anturilaitteelle.

HUB jakaa vastaanotetun ohjausskriptin Wirepas-verkon vaatimiin 102 tavun pituisiin tietopaketteihin. Tietopaketit lähetetään viestissä määritellylle anturilaitteelle. Jokaiseen tietopakettiin lisätään paketin järjestysnumero ja pakettien kokonaismäärä. On myös mahdollista lähettää skripti kaikille Wirepas-verkon anturilaitteille yhtä aikaa. Tämä tapahtuu lähettämällä datapaketit broadcast-viestinä.

## **6.5 Pilvipalvelun toiminta**

Amazon Web Services (AWS) toimii pilvipalveluna, jossa on käytössä MQTT Broker. MQTT Brokerin tehtävänä on välittää tietoliikenne HUBin ja pilvipalvelun välillä. Skripti siirretään pilvipalvelusta anturilaitteelle MQTT Brokerin avulla. Tietojen siirtoon käytetään JSON-standardia. JSON on hyvin tuettu standardi eri ohjelmointikielissä, joten se soveltuu hyvin tiedonvälitykseen eri laitealustojen kesken.

## **6.6 Skriptin testaus**

Skriptin lähetystä ja vastaanottoa testattiin Wirepas-verkossa. Aluksi Wirepas-verkossa oli vain yksi anturilaitte, jonka avulla voitiin testata skriptin siirtoa ilman usean anturilaitteen tuomaa tiedonsiirtoviivettä ketjutetussa Mesh-verkossa. Skriptin lähetys HUBista toimi suunnitellulla tavalla. Tämä tarkistettiin Wirepas Terminal -ohjelmalla. Ohjelmalla voidaan tarkastella, mitä ja minkä sisältöisiä viestejä Wirepas-verkossa kulkee. Ohjelma voidaan määrittellä käyttäytymään, niin kuin se olisi itsenäinen Wirepas-päätelaitte. Näin pystyttiin tarkistamaan, että HUB lähettää kaikki skriptin osat verkkoon ja ne sisältävät skriptin tiedot, skriptin järjestysnumeron, osien kokonaismäärän ja oikean osan ositetusta skriptistä.

Seuraavaksi testattiin skriptin siirtoa oikeaan anturilaitteeseen. Testissä havaittiin, että kaikki skriptin osat eivät tulleet perille. Kun ongelmaan etsittiin syytä, havaittiin että, anturilaitteen ohjelmiston toimiessa skriptissä määritellyllä kellokeskeytyksellä ei anturilaitteen pääohjelman vastaanottofunktio pystynyt ottamaan kaikkia tulevia skriptin osia vastaan. Vastaanottofunktio hukkasi silloin tällöin jonkin skriptin osan. Tämän takia vastaanotettu skripti ei ollut täydellinen ja skriptin vastaanotto epäonnistui.

Skriptin vastaanotto-ongelma löydettiin ratkaisu. Ennen varsinaisen skripti tiedon lähetystä lähetettiin vastaanottavalle anturilaitteelle skriptin saapumisesta ilmoittava viesti. Kun anturilaitteeseen saapuu ilmoituksen tulevasta skriptistä, muutetaan anturilaitteen skriptiä ohjaava kellokeskeytys 1 Hz:iin ja keskeytetään skriptin suoritus. Näin vastaanottofunktiolla on tarpeeksi aikaa ottaa vastaan kaikki tulevat skriptin osat. Kellokeskeytyksellä voidaan kontrolloida aikaa, joka on käytetty skriptin vastaanotossa. Jos määritelty maksimi vastaanottoaika ylittyy, poistetaan vastaanotetut skriptin osat ja jatketaan vanhan skriptin käyttöä. Kun edellä mainitut muutokset oli tehty ohjelmistoihin, toimi skriptin siirto HUBilta anturilaitteeseen ilman ongelmia.

Seuraavaksi Wirepas-verkkoon lisättiin useampi anturilaitte. Näin saatiin aikaan ketjutettu Mesh-verkko, jossa skripti siirtyy HUBilta reitittävälle sensorilaitteelle ja siitä edelleen skriptiä vastaanottavalle sensorilaitteelle. Testikokoonpanossa Wirepas-verkko toimi odotusten mukaisesti eli lähetetty skripti siirtyi kokonaisuudessaan vastaanottavalle sensorilaitteelle. Monien testiajojen ja variaatioiden jälkeen voidaan todeta, että skriptin siirto HUBilta anturilaitteelle toimii.

## **6.7 Ohjelmiston jatkokehitys**

Skripti-ohjelmiston jatkokehittelyä on hyvä jatkaa anturilaitteen skriptin vastaanoton parantamisella. Skriptin saapumisesta ilmoittavan viestin avulla voisi skriptin lähetysrutiinia parantaa siten että anturilaitte voisi lähettää HUBille viestin, missä se kertoo olevansa valmiina vastaanottamaan skripti-tiedoston. Tämän kuittauksen perusteella HUB lähettää skriptin osat anturilaitteelle. Samaisella viestillä anturilaitte voisi ilmoittaa HUBille, että skripti on tullut kokonaisuudessaan perille, tai virheen sattuessa pyytää skriptin uudelleenlähetystä.

## 7 YHTEENVETO

Tässä opinnäytetyössä tavoitteena oli suunnitella ja toteuttaa Thingsee-anturilaitteperheelle ohjelmisto, jolla pystytään muuttamaan anturilaitteiden toimintaa ilman uudelleenohjelmointia. Pohdittavana oli taulukkomuotoinen ohjaus ja skriptiohjaus. Vaatimuksena oli, että ohjaustiedon siirto anturilaitteelle täytyy onnistua ilmateitse. Näin anturilaitteiden toimintojen muuttaminen ei vaatisi laitteen uudelleen ohjelmointia.

Taulukkomuotoisessa ohjauksessa haasteena on uusien ominaisuuksien lisääminen anturilaitteeseen. Ominaisuuksien lisääminen vaatii uuden taulukon tekemistä ja uusi taulukko pakottaa anturilaitteen ohjelmiston muuttamista vastaamaan uutta ohjaustaulukkoa.

Skriptiohjauksessa anturilaitteen oman ohjelmiston rinnalla voi toimia skriptillä toteutettu lohko. Näin anturilaitteiden toimintoja voidaan muuttaa vaihtamalla anturilaitteessa oleva skripti. Skriptiohjauksessa on helpompi tehdä anturilaitteen toimintoihin muutoksia sen käyttöönoton jälkeen. Kun skripti vaihdetaan anturilaitteeseen ja otetaan käyttöön, on anturilaitteen toiminnot muutettu.

Opinnäytetyössä päädyttiin ottamaan ohjaustavaksi skriptiohjaus ja skriptikieliksi PAWN-skriptikieli. PAWN-skriptikieli valittiin, koska se on suunniteltu silmillä pitäen laitteita, joissa RAM-muistin määrä on vähäinen. PAWN-kielessä skriptiä käyttävä osa saadaan sulautettua osaksi C-kieltä. Näin skriptin suorittaminen on nopeaa ja kuluttaa vähän RAM-muistia.

PAWN-skriptikielen lisääminen alkuperäiseen anturilaitteen pääohjelmaan sujui ilman suurempia ongelmia. Anturilaitteen RAM-muisti oli yksi kriittisistä alueista, mutta PAWN-skriptikieli osoittautui hyväksi vaihtoehdoksi juuri vähäisen muistitarpeen takia. PAWN-skriptin käyttöönotossa hyvänä apuna oli PAWNin Implementer's Guide (16). Ohjeessa oli hyvät esimerkit PAWNin toiminnasta ja käytöstä, sekä skriptin kääntämisestä ajettavaan muotoon.

Skriptin lähetys pilvipalvelusta sekä skriptin vastaanotto ja käsittely HUBissa anturilaitteen ymmärtämään muotoon oli mielenkiintoista. PAWNin AMX-moottorin

lisääminen anturilaitteen ohjelmaan ja skriptiosion saaminen toimimaan vaativat suurehkoja muutoksia anturilaitteen ohjelmistoon ja muistinkäsittelyyn.

Opinnäytetyön teko antoi hyvän läpileikkauksen antureihin pohjautuvista mittaus-tekniikoista sisältäen anturit, HUBit, tietoliikenteen ja pilvipalvelun. Opinnäytetyössä saavutettiin tavoitteet ja lopputuloksena on pilvipalvelun kautta muokattava anturilaite.



## LÄHTEET

1. Thingsee sensors. Haltian. Saatavissa: <https://www.haltian.com/Internet-Of-Things/Thingsee-POD-and-TAG>. Hakupäivä 22.8.2017.
2. Ericsson Mobility Report. 2017. Ericsson. Saatavissa: <https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-june-2017.pdf>. Hakupäivä 8.9.2017.
3. Bluetooth specifications. Bluetooth SIG. Saatavissa: <https://www.bluetooth.com/specifications>. Hakupäivä 20.9.2017.
4. Thingsee wirepas platform. Haltian. Saatavuus: <https://thingsee.com/thingsee-wirepas-platform>. Hakupäivä 18.9.2017.
5. Zigbee network topology. Zigbee Alliance. Saatavissa: <http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeepr/>. Hakupäivä 18.10.2017.
6. LTE evolution for IoT connectivity. 2017. Nokia. Saatavissa: <https://resources.ext.nokia.com/asset/200178>. Hakupäivä 20.9.2017.
7. SM5100B-D GSM/GPRS Module Hardware Specification. 2008. Shanghai Sendtrue Technologies. Saatavuus: [https://cdn.sparkfun.com/datasheets/Cellular/SM5100B-D\\_HW\\_spec\\_V1.0.0.pdf](https://cdn.sparkfun.com/datasheets/Cellular/SM5100B-D_HW_spec_V1.0.0.pdf). Hakupäivä 15.9.2017.
8. Measuring Bluetooth Low Energy Power Consumption (Rev. A). 2012. Texas Instruments. Saatavuus: <http://www.ti.com/lit/pdf/swra347>. Hakupäivä 15.9.2017.
9. THINGSEE POD & TAG DATA SHEET. Haltian. Saatavuus: <https://thingsee.com/thingsee-pod-tag-data-sheet>. Hakupäivä 1.9.2017.
10. Mesh Networking Specifications. Bluetooth SIG. <https://www.bluetooth.com/specifications/mesh-specifications>. Hakupäivä 20.9.2017.

11. Network topology. 2017. Wikipedia. Saatavuus: [https://en.wikipedia.org/wiki/Network\\_topology](https://en.wikipedia.org/wiki/Network_topology). Hakupäivä 2.11.2017.
12. nRF52. Nordic Semiconductor. Saatavissa: <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52832>. Hakupäivä 22.8.2017
13. Wirepas technology. 2017. Saatavissa: <https://wirepas.com/connectivity/technology/>. Hakupäivä 1.9.2017.
14. Wirepas Connectivity Overview. 2017. Wirepas. Saatavuus: <https://wirepas.com/download/> (vaatii kirjautumisen). Hakupäivä 18.10.2017.
15. PAWN. 2017. CompuPhase. Saatavissa: <https://www.compuphase.com/pawn/pawn.htm>. Hakupäivä 6.9.2017.
16. Pawn embedded scripting language Implementer's Guide. 2016. CompuPhase. Saatavissa: [https://github.com/compuphase/pawn/blob/master/doc/Pawn\\_Implementer\\_Guide.pdf](https://github.com/compuphase/pawn/blob/master/doc/Pawn_Implementer_Guide.pdf). Hakupäivä 19.9.2017.