

Mikko Känninen

# Yleiskatsaus IoT-ympäristöihin ja anturijärjestelmä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

6.11.2016

|   |  |
|---|--|
| Tekijä<br>Otsikko<br><br>Sivumäärä<br>Aika  | Mikko Känninen<br>Yleiskatsaus IoT-ympäristöihin ja anturijärjestelmä<br>56 sivua<br>6.11.2016 |
| Tutkinto  | Insinööri (AMK)  |
| Koulutusohjelma   | Tietotekniikka   |
| Suuntautumisvaihtoehto  | Sulautetut järjestelmät  |
| Ohjaaja   | Lehtori Keijo Länsikunnas  |
| <p>Insinööriyön tarkoituksena oli perehtyä esineiden internetiin (engl. Internet of Things, IoT) yleisellä tasolla ja tutustua pintapuolisesti sen standardeihin ja protokoliin. Big data ja IoT ovat saaneet suuren huomion teknologiateollisuuden trendinä, joten niiden tuntemus edes jossain määrin on pakollista. Toisena tavoitteena oli toteuttaa toimiva prototyyppi IoT-sovelluksesta, joka sisältäisi mittalaitteiston ja tiedontallennusratkaisun sekä verkkosivuston tiedon esittämistä varten.</p> <p>Ensimmäisessä työvaiheessa tutustuttiin jo olemassa tai kehitteillä oleviin IoT-protokoliin ja standardeihin. Tarkoituksena oli selvittää niiden mahdollinen soveltuvuus ja käytettävyys prototyypin valmistuksessa.</p> <p>Seuraavassa vaiheessa valittiin komponentit prototyyppiin. Vertailun jälkeen alustaksi valikoitui PSoC4, jossa prosessorina toimii ARM Cortex0. Datayhteyttä varten valittiin SIM900 GPRS/GSM Shield ja antureiksi kolme DS18B20-lämpöanturia. Prototyypin ohjelmisto toteutettiin C-kieltä hyödyntävällä PSoC Creator -ohjelmistolla. Työssä toteutettiin myös verkkosivu anturidatan graafista esittämistä varten. Tehdyn työn lopputuloksena syntyi itsenäinen anturijärjestelmän prototyyppi, ja perehtymisvaiheessa kerätty tieto IoT-protokollista edesauttaa muuntamaan prototyypin langattomaksi terraarion ilmaston hallinta- ja säätöjärjestelmäksi.</p> |  |
| Avainsanat  | IoT-protokollat, IoT-standardit, PSoC4, IoT-sovellukset  |

|  |  |
|--|--|
| Author<br>Title<br>Number of Pages<br>Date   | Mikko Känninen<br>Basics of IoT environments and measurement system<br>56 pages<br>6 November 2016 |
| Degree   | Bachelor of Engineering  |
| Degree Programme   | Information Technology   |
| Specialisation option  | Embedded Systems   |
| Instructor   | Keijo Länsikunnas, Senior Lecturer   |
| <p>One of the main goals of the final year project was to study the term the Internet of Things (IoT) in general and to explore IoT-related standards and protocols. The main reason is that, Big data and IoT, are the biggest rising trends in the field of technology. Understanding the main characteristics of both is mandatory. The second objective was to build a working prototype of an IoT application. This device would include a microcontroller, a few sensors as measuring equipment, a data storage solution and a web-page for presenting collected information.</p> <p>The first phase of the project was learning about existing or emerging IoT protocols and standards. The aim was to find out if they are potentially suitable and if they can be integrated into the prototype somehow.</p> <p>At the second phase, the goal was to choose suitable components for the prototype. After comparing features of the different development boards available, the PSoC4 development board with an ARM Cortex0 processor was chosen and for the data connection SIM900 GPRS/GSM Shield was chosen. Sensor data was created using three DS DS18B20 temperature sensors. The software for the prototype was written in the C language using PSoC Creator IDE. The final year project included a web-page for the graphical presentation of the data values. All the initial goals were met and the result was a working independent sensor unit prototype, and furthermore, all the information gathered about the IoT protocols will help out to transform the prototype into a wireless terrarium climate control system.</p> |  |
| Keywords   | IoT protocols, IoT standards, PSoC4, IoT applications  |

## Sisällys

### Lyhenteet

|     |   |    |
|-----|---|----|
| 1   | Johdanto  | 1  |
| 2   | IoT:n perusteet, standardit ja protokollat          | 2  |
| 2.1 | IOT-siirtoyhteyskerrosprotokollat                   | 3  |
| 2.2 | Verkkokerroksen reititys- ja kapselointiprotokollat | 15 |
| 2.3 | Istuntokerroksen protokollat                        | 20 |
| 2.4 | IoT ja tietoturva                                   | 24 |
| 2.5 | Haasteet ja tulevaisuus                             | 26 |
| 3   | Prototyypin laitteiston määrittely                  | 28 |
| 3.1 | Laitteiston käytettävyys ja valintaperusteet        | 28 |
| 3.2 | PSoc4-mikro-ohjain                                  | 31 |
| 3.3 | Väylät  | 32 |
| 3.4 | DS18B20-lämpöanturi                                 | 37 |
| 3.5 | SIM900-moduuli                                      | 39 |
| 3.6 | Laitteistokaavio ja toimintamalli                   | 41 |
| 4   | Prototyypin laitteisto- ja ohjelmistototeutus       | 42 |
| 5   | Yhteenveto ja jatkokehitys                          | 48 |
|     | Lähteet   | 50 |



## Lyhenteet

|                  |   |
|------------------|---|
| ARM              | Advanced RISC Machines, 32-bittinen mikroprosessoriarkkitehtuuri.   |
| CA               | Carrier Agregation, LTE-A-tekniikka, jolla suurennetaan siirtokapasiteettia lisäämällä väylän leveyttä.                           |
| CMOS             | Complementary Metal Oxide Semiconductor, mikropiiritekniikka, joka perustuu kanavatransistoreihin.                                |
| CRC              | Cyclic redundancy check, tiivistealgoritmi tiedostonsiirtovirheiden havaitsemiseksi.  |
| FPGA             | Field-programmable gate array, logiikkaohjelmoitava mikropiiri, joka tukee uudelleenohjelmointia.                                 |
| GPRS             | General Packet Radio Service, pakettikytkentäinen tiedonsiirtopalvelu joka toimii GSM-verkoissa.                                  |
| HAN              | Home Area Network, aluekohtaisesti rajattu verkko, joka on suunnattu erityisesti älylaitteille.                                   |
| I <sup>2</sup> C | Kaksisuuntainen sarjamuotoinen tiedonsiirtoväylä.   |
| IEEE             | Institute of Electrical and Electronics Engineers, kansainvälinen elektroniikka-alan, tieto- ja viestintätekniikan alan järjestö. |
| IETF             | The Internet Engineering Task Force, Internet- ja verkkopuolen protokollien standardoinnista vastaava organisaatio.               |
| IoT              | Internet of things, esineiden internet.   |
| ISM              | Industrial, Scientific and Medical, radiotaajuuskaista-alue, jonka hyödyntämiseen ei vaadita erillisiä lupia.                     |
| LLN              | Low power and Lossy Networks, alhaisen tehon häviölliset verkot.  |

|      |   |
|------|---|
| MAC  | Media Access Control, tietoverkoissa liikennöintiä hoitava järjestelmän osa.      |
| PDO  | PHP Data Objects, PHP 5.1 -version tuoma tietokantaohjelmointimalli.              |
| PHY  | OSI-mallin fyysinen kerros.   |
| QoS  | Quality of service, tietoliikenteessä käytössä oleva laatutason määritelmä.       |
| RFC  | Request for Comments, IETF-standardeja.   |
| ROM  | Read Only Memory, lukumuisti.   |
| SIM  | Subscriber Identity Module, älykortti, jolla yksilöidään matkapuhelinliittymä.    |
| SRAM | Static Random Access Memory, kiikkupireillä toteutettu staattinen muistityyppi.   |
| TDMA | Time Division Multiple Access, kanavanvaraustekniikka.                            |
| TTL  | Transistor-transistor logic, digitaalielektronikan logiikkapiiriperhe.            |
| TVWS | TV White Space, televisiolähetyksien vapaat taajuusalueet.                        |
| UDP  | User Datagram Protocol, tiedostonsiirtoprotokolla.                                |
| UNB  | Ultra Narrow Band, modulointitapa suurien datamäärien lähettämiseen.              |
| WAN  | Wide Area Network, maantieteellisesti laajoja alueita kattava tiedonsiirtoverkko. |

## 1 Johdanto

Internet of Things, eli esineiden internet laajenee jatkuvasti. Verkossa kommunikointiin kykenee jo suuri määrä päivittäisessä käytössä olevista laitteista. Tämä mahdollistaa datan keruun vaivattomasti laitevalmistajien ja käyttäjien hyväksi. Tämän seurauksena pystytään haluttujen tunnuslukujen pohjalta toteuttamaan entistä parempia tuotteita tai palveluita.

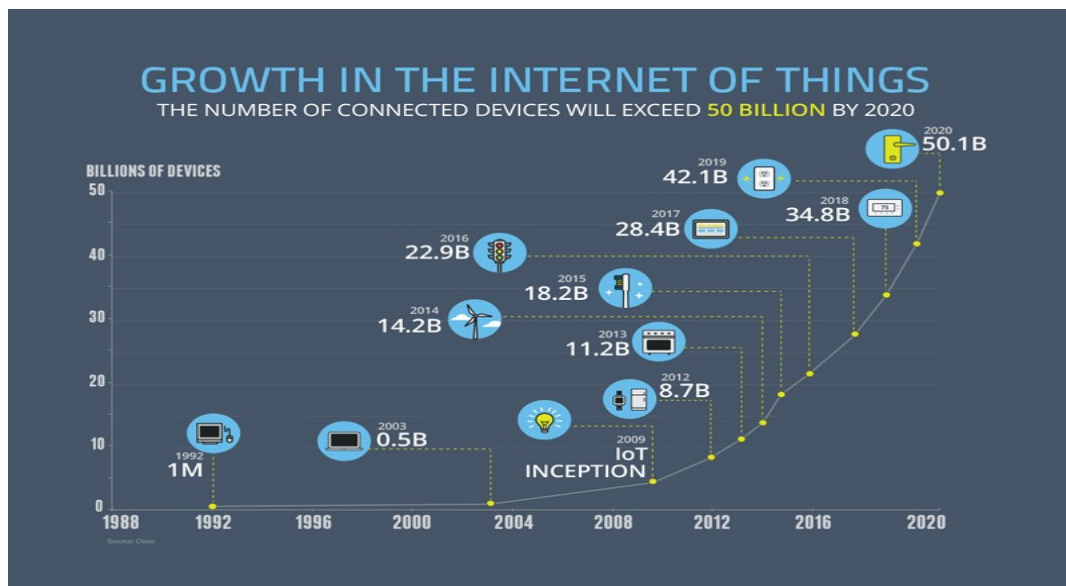
Insinööriyön tarkoituksena on perehtyä erilaisiin IoT-protokolliin ja tutustua IoT:hen yleisellä tasolla sekä luoda itsenäinen anturijärjestelmän prototyyppi, joka mahdollistaa reaaliaikaisen anturidatan seuraamisen web-käyttöliittymän kautta ja aikaisemmin kerättyjen näytteiden tarkastelun. Työ on saanut alkunsa aikaisemmin toteutetusta Antin kasvimaan -projektista, joka oli tarkoitettu seuraamaan lämpötiloja kasvinmaan lähiympäristössä ja varautumaan muutoksiin. Tämän vuoksi määrittely on viisainta aloittaa siitä, mitä aiemmin valmistetusta laitteesta puuttuu ja mitä siitä täytyy muuttaa. Laite koostui yhdestä lämpöanturista, mikroprosessorista ja radiolähettimestä sekä palvelimeen kytketystä vastaanotinpuolesta. Virtalähteenä laitteessa toimi muutaman AA-pariston yhdistelmä. Tavoitteena oli saada aikaan toimiva konsepti, jonka pohjalta lähettäisiin kartoittamaan mahdollisuutta hyödyntää laitetta routamittauksessa. Tämä tavoite kuitenkin hylättiin ennen työn aloittamista kolmannen osapuolen toimesta, joten tavoitteeksi vaihtui konseptin luominen omaan harrastekäyttöön.

Määrittelyvaiheessa ensimmäisenä pohditaan prototyypin anturien vähimmäismäärä, joka olisi riittävä alustava otanta toteamaan useamman anturin toiminta. Lähinnä virrankulutuksen ja mittauksen keston osalta tätä määrää pystyttäisiin tarpeen tullen laajentamaan ryppäissä virtalähteen rajoissa. Seuraavana pohdintavuorossa on datasiirto laitteesta pilveen ja sen vaatimuksista. Vaatimuksena on pitkä toimintasäde ja riippumattomuus kaapelikytkentäisistä verkoista virran ja datan osalta. Virtalähteessä vaatimus on kyetä pitämän järjestelmää mahdollisimman kauan toimintakuntoisena.

## 2 IoT:n perusteet, standardit ja protokollat

Esineiden internet ei ole kovin uusi asia, vaikka sitä joka puolella esitellään ja mainitaan usean tietotekniikka-alalla olevan yrityksen mainoksissa. Se on ollut olemassa melkein yhtä kauan, kuin laitteita on ollut mahdollista kytkeä verkkoon, siis jopa kymmeniä vuosia. Kuitenkin vasta muutamia vuosia sitten se on noussut teknologiateollisuuden trendiksi. IoT ei ole siis itsessään mikään standardi, eikä ole erillisiä määritteitä, mitä sen alle voi luokitella.

Vaikka näitä älykkäitä laitteita on ollut kauan, trendiksi ne ovat kehittyneet vasta viime vuosina otollisten mahdollisuuksien myötä. Esimerkkinä voi mainita teknologiatuotteiden alhaisemman hinnan anturien ja piirien tuotantokustannusten laskun myötä. Myös koko ja virrankulutus komponentitasolla on saatu niin pieneksi, että äly voidaan lisätä laitteeseen ilman suurta vaivaa ja suunnittelutyötä. Myös IPv6-teknologia vaikuttaa tähän, sillä se mahdollistaa, että jopa jokaiselle asunnon pistorasialle voidaan jakaa tarvittaessa IP-osoite ja ei tarvitse huolehtia osoitteiden riittävydestä. Matkapuhelinverkkojen kehitys ja niiden kattavuuden laajentuminen on myös ollut suuressa roolissa. Se on parantanut verkkoon liittymisen mahdollisuuksia kasvavassa määrin etenkin rajaseuduilla. Vuoteen 2020 mennessä voidaan odottaa ennusteiden mukaan, että pelkästään näitä älylaitteita on kytkettynä verkkoon jopa 50 miljardia [1] (kuva 1).

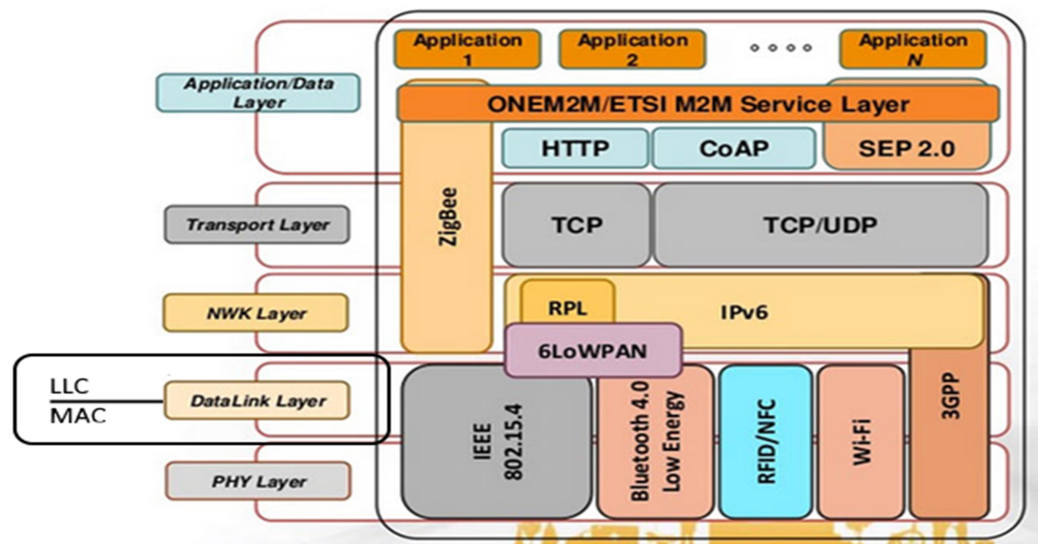


Kuva 1. Usean alan ammattilaisten arviointiin perustuva ennuste tulevaisuuden internetiin kytkettyjen laitteiden määrästä vuonna 2020 [1].

Seuraavissa luvuissa perehdytään eri protokollisiin liittyen eri verkkokerroksiin ja käyttö-tarkoituksiin. Tässä kuitenkin esitellään vai muutamia, niin standardisoituja kuin valmisteluvaiheessa olevia. Tämä siitä syystä, että näitä erilaisia protokollia syntyy todella paljon uusien tekniikoiden kehittyessä ja versioiden edetessä moni saa lisäyksiä ja kehittyä aivan uudeksi.

## 2.1 IOT-siirtoyhteyskerrosprotokollat

Siirtoyhteyskerroksen protokollat on usein toisen kerroksen protokolliksi kutsuttu ryhmä ja päätehtävänä on tarjota väylä kahden laitteen kommunikointiin (kuva 2). Seuraavassa esitellään muutamia fyysisen ja MAC-tason protokollia. Monessa esiteltävistä standardeista nämä on yhdistetty.



Kuva 2. Siirtoyhteyskerroksen alikerroksille sijoittuvat LLC (engl. Logical Link Control) ja MAC (engl. Media Access Control) sekä kuvaus tekniikoista kerrosten tasoilla [2].

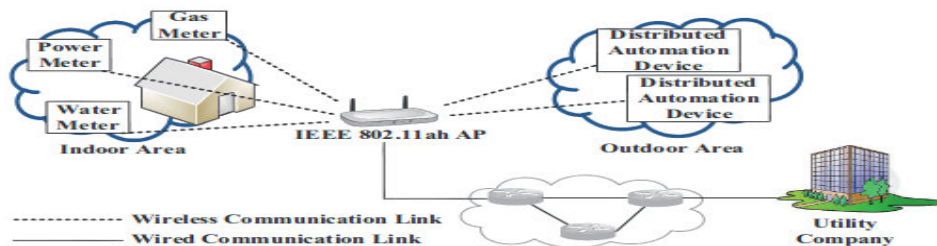
### IEEE 802.15.4e

IEEE 802.15.4e on yleisin käytetty MAC-IoT-standardi. Se määrittää kehysformaatin, lähde- ja vastaanottajan osoitteet sisältävät otsikot ja sen, kuinka solmut keskustelevat

keskenään. Perinteisten verkkojen kehysformaatit eivät sovellu vähävirtaisiin niin sa-  
nottuihin multihopping networking -IoT-ympäristöihin päällekkäisyysmahdollisuuksien  
takia. Ajan synkronointia ja kanavavaihtoja hyödyntäen se mahdollistaa IoT-  
ympäristössä vaaditun korkean luotettavuuden ja alhaiset kustannukset. [3.]

### IEEE 802.11 ah

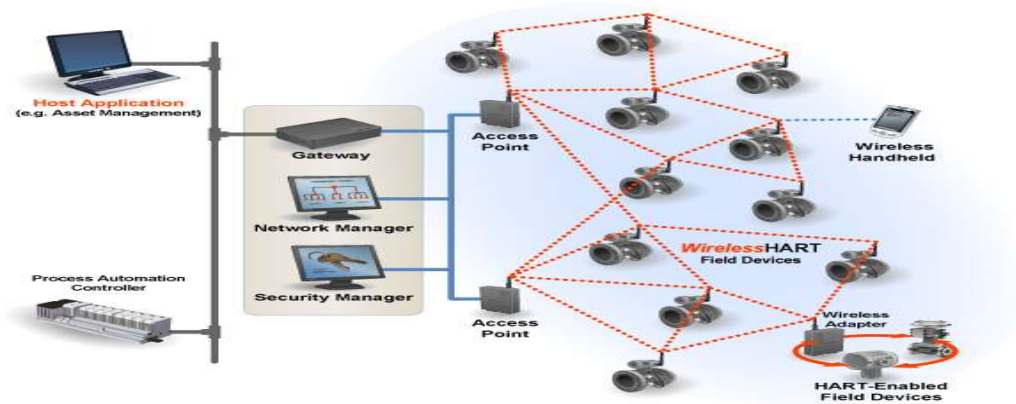
IEEE 802.11 ah on vähävirtainen versio alkuperäisestä IEEE 802.11:sta, joka tunne-  
taan myös nimellä Wi-Fi, ja se on suunniteltu vähäisemmällä päällekkäisyyksien mää-  
rällä. Wi-Fi:ssä kehyksen koko on 30 tavua, ja se on liian suuri IoT-sovelluksien kanssa  
käytettäväksi, jolloin ah:ssa on päädytty noin 12 tavuun. Verkko muistuttaa ulkoisesti  
paljon tavallista Wi-Fi:ä (kuva 3). [3.]



Kuva 3. Esimerkki IEEE 802.11 ah -älyverkosta [4].

### WirelessHART

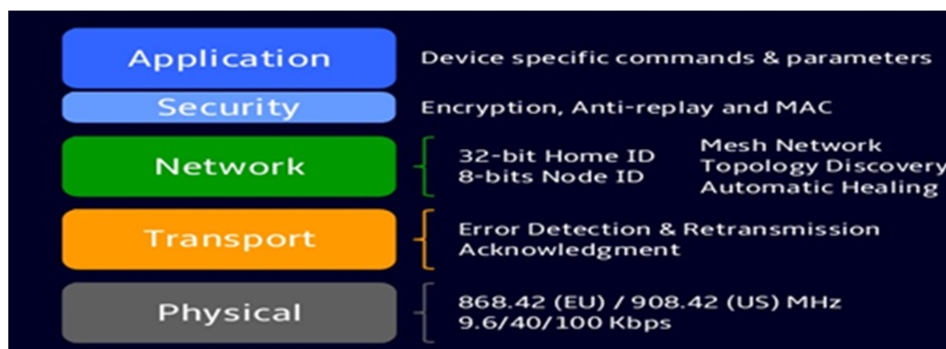
HART (Highway Addressable Remote Transducer) on kaksisuuntainen teollisuudessa  
käytetty protokolla, jota hyödynnetään laitteiden ja isäntäjärjestelmien kommunikaati-  
ossa. WirelessHART on HART-siirtoyhteyskerrosprotokollan langaton lisäosa, joka jul-  
kaistiin syksyllä 2007. Se operoi IEEE 802.15.4- fyysisen kerroksen päällä ja adoptoi  
TDMA:n sen MAC-kerrokselta. Tämä MAC-protokolla hyödyntää kehittyneitä salausta  
ja laskee viestien yhtenäisyyttä ollakseen luotettava. Standardi tarjoaa yhden hypyn,  
kahden laitteen välisen ja päästä päähän -salausmekanismit. Verkossa on kolme pää-  
elementtiä, langattomat laitteet, yhdyskäytävät ja verkonhallintalaitteet (kuva 4). [3; 5;  
6.]



Kuva 4. Yksinkertainen WirelessHART-verkon kuvaus [6].

## Z-Wave

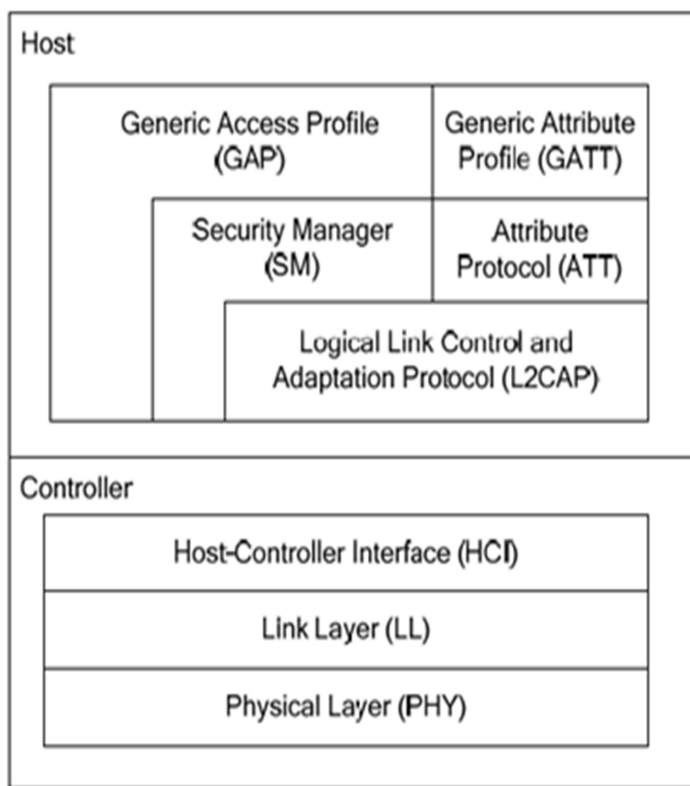
Z-Wave on alhaisen siirtonopeuden vuorosuuntainen protokolla. Se on suunniteltu matalien vaatimuksien verkkoihin ja sitä hyödynnetään paljon esimerkiksi yksinkertaisessa kodinautomaatiossa. Protokollan pääasiallinen tehtävä on kommunikointi luotettavasti lyhyillä ohjauksenkomennolla verkon muiden laitteiden kanssa. Tämä tapahtuu isäntä-orkitehtuuriin pohjautuen. Protokolla ei sovi suoratoistoon tai aikakriittisen datan järjestelmiin alhaisen siirtonopeuden (kuva 5) ja pakettien koon vuoksi. [3; 7.]



Kuva 5. Z-Wave-protokollapino ja muutamia yksityiskohtia [8].

## Bluetooth Low Energy

Bluetooth Low Energy tunnetaan myös nimellä Bluetooth Smart tai Version 4.0+. Se on sovellusystävällinen ja kehitetty IoT-ympäristöihin. Bluetooth LE on teollisen standardin langaton protokolla, joka on suhteellisen yhteensopiva muitten Bluetoothia tukevien laitteiden kanssa. Yhteensopivuuden vuoksi se tuo sovelluskehitykseen ja käyttöönottoon kustannustehokkuutta. Bluetooth LE tarjoaa todella tiukan 128-bittisen AES-salauksen. Protokolla on myös tehonkulutuksen suhteen erittäin alhaisella tasolla niin huipulla, keskiarvoiltaan ja ollessaan toimettomana. Se pääsee jopa kymmenekseen tavallisen Bluetoothin virrankulutuksesta. Bluetooth LE:ssä on määritelty useita valmiita profiileja käyttötarkoituksen mukaan tai on mahdollista hyödyntää GATT (engl. Generic Attribute) -profiilia uuden luomiseksi (kuva 6). [3; 9.]

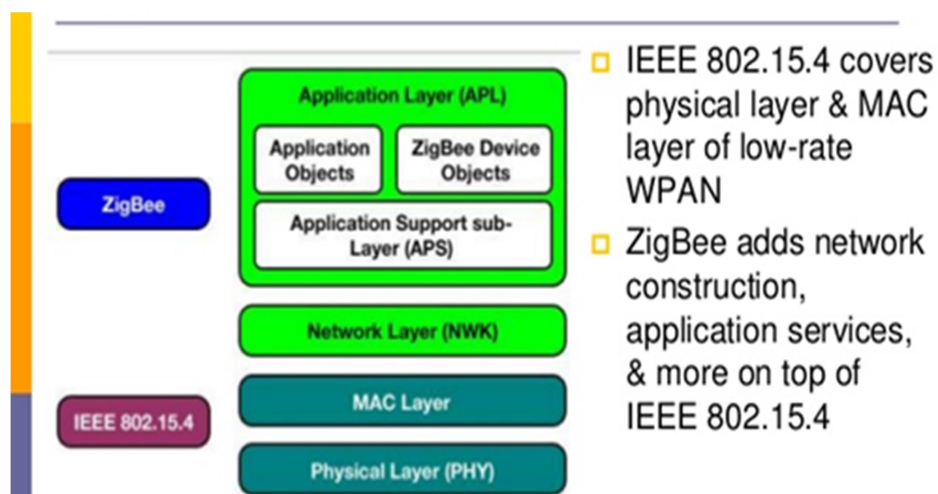


Kuva 6. Bluetooth LE -protokollapino [7].



## ZigBee Smart Energy

ZigBee Smart Energy on yleinen erityisesti HAN-järjestelmissä käytetty protokolla ja sitä käytetään myös kauko-ohjaimissa ja terveydenhoitoalan IoT-sovelluksissa. Se pohjautuu IEEE 802.15.4:ään (kuva 7) ja sitä käytetään luomaan pienehköjä vähävirtasia verkkoja. Tämä kuitenkin rajoittaa kantaman alle 100 metriin ja tiedonsiirtonopeuden suhteellisen alhaiseksi, maksimissaan noin 250 kb/s. Itse ZigBee-standardi määrittelee kolme profiilipinoa, ZigBee, ZigBee Pro ja ZigBee IP. Nämä verkot ovat mesh-tyyppisiä, eli jokainen laite pystyy kommunikoimaan toisten kanssa. ZigBee IP tuo Smart Energy Protocol 2 -version ja mahdollistaa IPv6-osoitteiston hyödyntämisen. [3; 11; 12.]



Kuva 7. ZigBee-protokollapino [13].

## DASH7

DASH7 toimii siirtoyhteyskerroksella (kuva 8) ja se on käytännössä avoimen lähdekoodin RFID-standardi. Se tarjoaa vähävirtaisen vaihtoehdon suhteellisen pitkällä, noin 1–2 km:n kantamalla. Siirtonopeus kuitenkin jää maksimissaan 200 kbps ja siksi toimiikin parhaiten sensoriverkoissa ja pieniä datamääriä siirrettäessä. Dash7 määrittää neljä laiteluokkaa. Blinker lähettää ainoastaan lähettää dataa. EndPoint lähettää ja vastaanottaa tietoa sekä tukee herätystoimintoja. Subcontroller on täydellä toiminnallisuudella varusteltu laite, joka ei ole koko aikaa aktiivisena vaan se herää kun verkon laitteita skannataan, kuten myös EndPoint-laitteet. Gateway yhdistää DASH7-verkon muihin

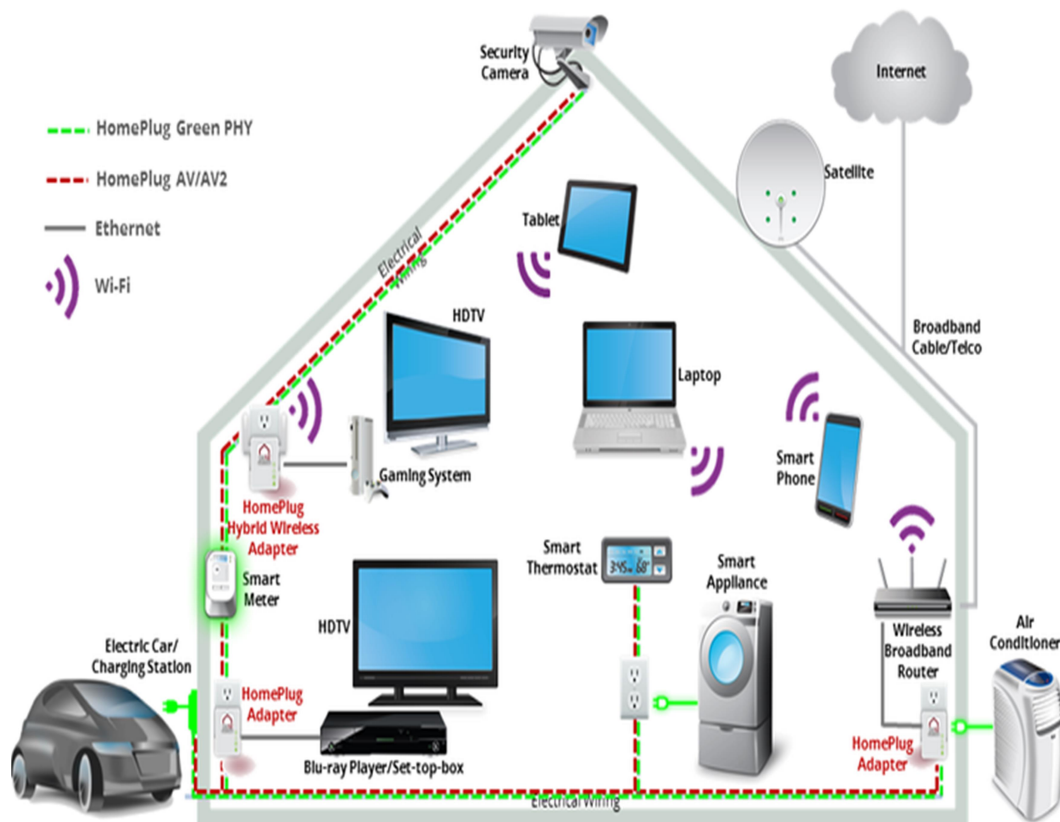
verkkoihin. Laiteella on vain aktiivinen tila, ja se kuuntelee, jos ei ole lähetettävää dataa. Kaikki DASH7-verkon laitteet tukevat vähintään yhtä mainittua luokkaa ja kahta kommunikaatiomallia, veto/haku ja työntö/talletus. [3; 14.]

| DASH 7 OSI Layers |             |                |                    |
|-------------------|-------------|----------------|--------------------|
| 7                 | Application | UDP/LLDP based |                    |
| Queries           | 6           | Presentation   | M2 File System     |
|                   | 5           | Session        | Dialog Stack       |
|                   | 4           | Transport      | M2QP               |
|                   | 3           | Network        | M2NP, M2DP, M2AdvP |
| 2                 | Datalink    | DASH7 MAC      |                    |
| 1                 | Physical    | 433 MHz GFSK   |                    |

Kuva 8. DASH7-protokollapino [15].

## HomePlug

HomePlug on MAC- ja PHY-kerrokset kattava protokolla, jonka on kehittänyt HomePlug Powerline Alliance, joka keskittyy pääasiassa kodinautomaatioon ja erinäisiin sähköisten kulkuneuvojen sovelluksiin (kuva 9). Se kattaa kolme versiota, HomePlug-AV, HomePlug-AV2 ja HomePlugGP. IoT-sovelluksiin, kodinautomaatioon ja älyverkkoihin näistä keskittyy GP. Suunnittelun pohjana on ollut vähentää HomePlug-AV:n kustannuksia ja virrankulutusta, mutta säilyttää yhteensopivuuden, luotettavuuden ja kohtuullisen kantaman. [3; 16.]



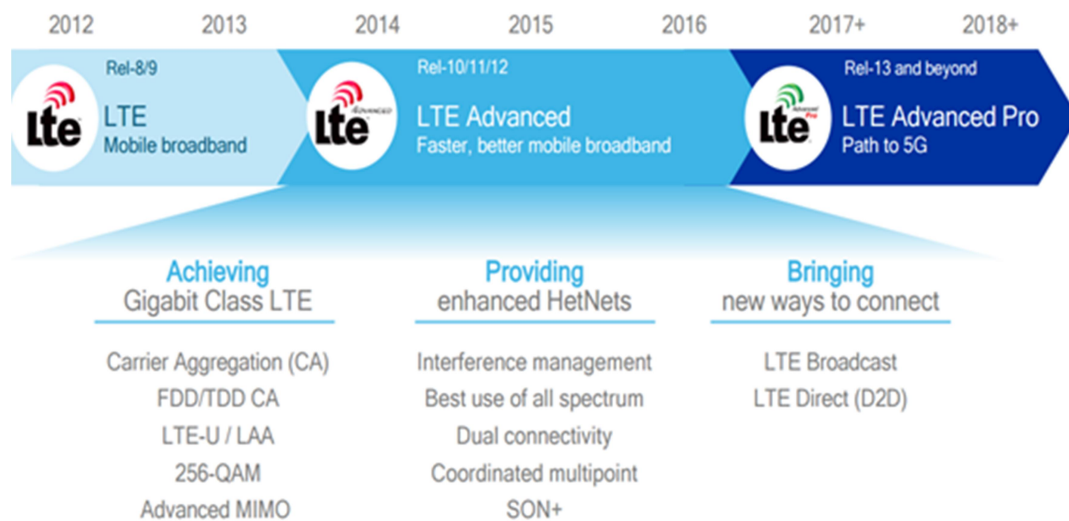
Kuva 9. Esimerkki HomePlugin hyödyntämisestä [17].

### G.9959

G.9959 on International Telecommunication Unionin määrittely lyhyen kantaman kapean kaistan digitaalisen radiokommunikaation lähettimiin. Se kattaa PHY- ja MAC-kerrokset. Se on suunniteltu reaaliaika-sovelluksia silmällä pitäen, esimerkkeinä kodin-elektronikka, AV-laitteet ja turvajärjestelmät. Se pohjautuu suurelta osin Z-Wave-protokollaan, ja se määrittelee myös taaksepäin yhteensopivuuden tämän kanssa. [3; 18.]

## LTE-A

Long-Term Evolution Advanced (LTE-A) on erityisesti suunniteltu laitteiden ja IoT-sovellusten väliseen kommunikaatioon matkapuhelinverkoissa silmällä pitäen suurempaa siirtokapasiteettia. Protokollan etuja ovat monipuolisuus ja alhaisemmat kustannukset muihin matkapuhelinverkoja hyödyntäviin protokolliin nähden. Se täyttää ITU:n IMT Advanced -standardin vaatimukset. Tiedonsiirron maksiminopeudet ovat kasvaneet latausten osalta 3 Gbps:iin ja lähetyksen osalta 1,5 Gbps:iin. Sillä on korkeampi spektrinen tehokkuus, jopa 30 bps/Hz. Myös samanaikaisten aktiivisten yhteyksien määrää on kasvatettu. Suurimmat protokollan uudistukset verrattuna aikaisempiin ovat Carrier Aggregation (CA), joka perustuu kaistan leveyden kasvattamiseen useamman taajuusalueen yhtäaikaisella käytöllä, ja Relay nodes (RN), joka tarjoaa vähävirtaisen ja edullisen ratkaisun verkon kattavuuden laajentamiseen halutuille alueille. LTE-A:sta on kehitteillä seuraava versio nimeltään LTE-A Pro. Se julkaistaan oletettavasti jo ennen vuotta 2018 (kuva 10). [19; 20; 21; 22.]



Kuva 10. LTE-tekniologioiden arvioitu kehityssuunta tulevaisuudessa [22].

## LoRa, LoraWAN

LoRa on fyysinen kerros tai langaton modulointitapa pitkän matkan kommunikaatiolin-kin luontiin. Se perustuu Chirp-hajaspektrimodulaatioon, joka on ollut käytössä kauan armeija- ja avaruusteknologioissa pitkän kantaman ja korkean häiriöherkkyyden vuoksi. LoRaWAN on MAC-protokolla pitkän kantaman ja alhaisen tehon verkkojen muodostamiseen. OSI-mallissa se karkeasti sijoittuu kerroksille 2 ja 3, LoRa:n määrittäessä ensimmäisen. LoRaWAN on WAN-sovelluksiin tarkoitettu protokolla, ja se hyödyntää tähtitopologiaa, jossa yhdyskäytävät reitittävät viestit solmujen ja sovellusten välillä. Solmut ovat yhden hypyn langattomassa kontaktissa yhteen tai useampaan yhdyskäytävään. LoRaWAN määrittää kolme solmutyyppiä: luokka A, jossa ne sallivat kaksisuuntaisen kommunikaation, mutta vasta kun solmu on lähettänyt uplink-viestin, se avaa kaksi vastaanotto kanavaa downlinkkiä varten; luokka B, jossa solmut sallivat kaksisuuntaisen kommunikaation ajastetuilla hetkillä; luokka C, jossa solmut pitävät vastaanottokanavia auki ja sulkevat ne ainoastaan lähettäessään. [23; 24.]

## Lora, Symphony Link

Syphony Link on myös LoRa-pohjainen fyysisen tason protokolla, joka optimoi kantaman ja vähäisen tehonkulutuksen. Suunnittelussa on huomioitu teollisuuden tarpeita: siinä on huomioitu joitakin matkapuhelinverkko- ja 802.11-standarteista. Asynkronisten ominaisuuksien (kapselointi ja uplink/downlink-koordinointi) ansiosta se tarjoaa nelinkertaisen kapasiteetin verrattuna LoRaWAN:iin. LoRaWAN myös salaa kaiken liikenteen, lähtevät ja saapuvat viestit kahden pisteen välisessä liikenteessä. Sen hyödyntäminen on vaikeaa sellaisissa järjestelmissä kuin valojenhallinta, joissa käytetään useammalle laitteelle välitettäviä yleiskomentoja. Syphony Linkistä taas löytyy useamman laitteen lähetysistuntoavaimet, ja ne mahdollistavat ryhmäosoitteiden jaon, jota esimerkiksi voi käyttää laitepäivitysten asentamiseen. [25; 26.]

### Weightless-N

Weightless-N on avoin alhaisen tehon tähtiverkkoarkkitehtuuri, joka toimii alle gigahertsin taajuusalueella. Se hyödyntää UNB-teknologiaa (ultra narrow band), ja toimintasäde on jopa useita kilometrejä. Teknologia tukee siirrettävyyttä, ja verkko kykenee reitittämään viestit päätteille automaattisesti. Verkot voivat olla päällekkäin, jolloin pääteasema kyselyillä varmistetaan, mitkä laitteet on rekisteröity kyseiseen verkkoon, ja datan määränpää ja reitti määrittyvät oikein. [27; 28.]

### Weightless-P

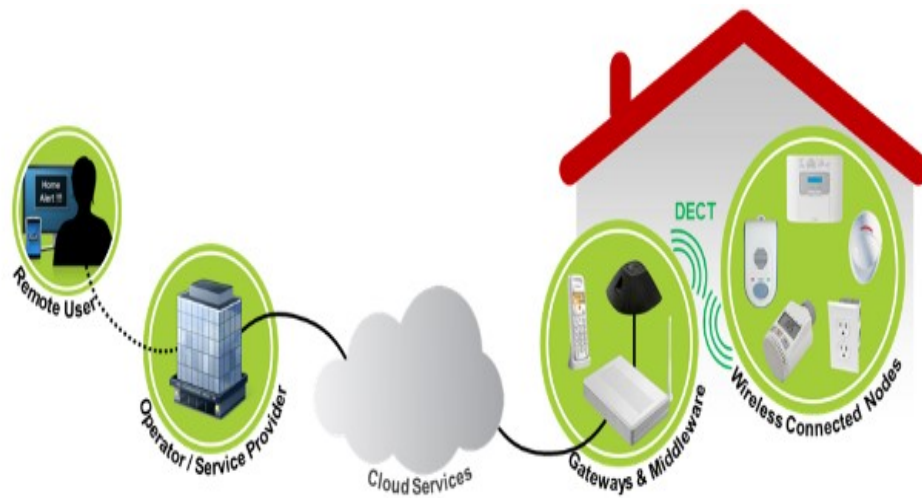
Weightless-P on Weightless-organisaation lippulaiva, korkean suorituskyvyn LPWAN-teknikka ja suunniteltu erityisesti lot:n tarpeiden mukaan. Protokolla käyttää kapean kaistan modulaatiota ja kaksisuuntaista kommunikaatiota. Weightless-P lupaa erittäin korkean palvelun laadun (QoS), ja myös samaa suorituskykyä ja luotettavuutta kuin matkapuhelinverkoissa, mutta vähäisemmillä kustannuksilla. Ilmoitettu tehonkulutus on odotustilassa 100  $\mu$ W, joka on huomattavasti pienempi kuin matkapuhelinverkkojen 3 mW. [27; 29.]

### Weightless-W

Weightless-W on avoin standardi, joka toimii televisiolähetysten vapaitten taajuuksien spektrissä (TVWS). Standardi on saanut alkunsa siitä, mitä Neul Ltd yritti kehittää ennen kuin siirtyi Huaweiin omistukseen. TVWS on teoriassa erittäin kiinnostava, mutta sen hyödynnettävyyttä heikentää se, ettei sitä ole saatavilla kaikkialla. Toinen ongelma on erityisesti siihen kytkettävien laitteiden antennit. Eritoten pienten antennien rakentaminen muuttavalle taajuusalueelle 470–790 MHz on melko lailla mahdotonta. Kuitenkin se tarjoaa jopa 10 Mbit/s -nopeuden paikoissa, joissa TVWS on käytettävissä ja antennit eivät koidu ongelmaksi. [27; 30.]

## DECT ULE

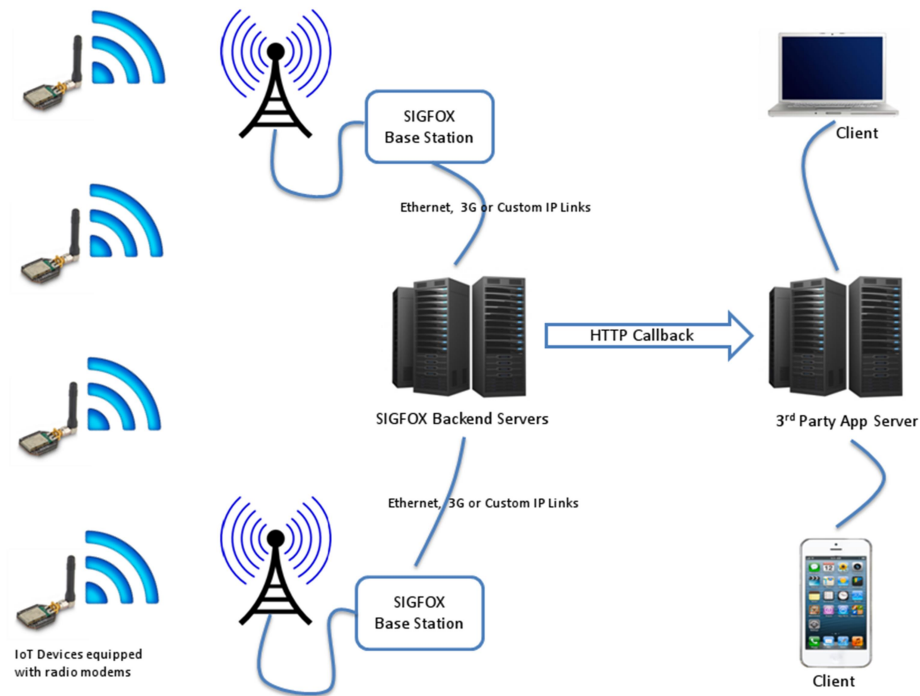
Digital Enhanced Cordless Telecommunications julkaistiin vuonna 1987, ja standardi on käytössä ympäri maailmaa langattomassa puhelinviestinnässä. DECT on kehittynyt siitä lähtien, ja viimeisin versio on vähävirtainen DECT ULE. Se hyödyntää tähtitopologiaa, ja lupaa sisätiloissa jopa 70:n ja ulkokäytössä yli 600 metrin kantaman. DECT ULE on ohjelmallinen lisäosa ja mahdollista implementoida DECT-yhteensopiviin laitteisiin. Protokollan pääasialliset sovellukset keskittyvät älykkäisiin HAN-laitteisiin (kuva 11). [31.]



Kuva 11. DECT ULE:n sovelluskohde [31].

## SIGFOX

SIGFOX tarjoaa matkapuhelinverkon tyyppisen tukiasemapohjaisen alustan (kuva 12) ja ratkaisuja IoT- ja M2M-sovelluksille. Se toimii pääasiallisesti osa-alueilla, joissa riittävät lyhyet viestit, mutta tarvitaan laajaa kattavuutta. SIGFOX käyttää UNB-teknologiaa, ja radiolinkissä hyödynnetään lisensoimattomia ISM-kaistoja. Taajuudet vaihtelevat hiukan valtiosta ja sen säädöksistä riippuen. Yleensä Euroopassa käytössä on 868 MHz:n kaista ja Yhdysvalloissa 915 MHz:n. Kantamaksi ilmoitetaan 30–50 km avoimessa maastossa ja häiriöllisessä ympäristössä se jää noin 3–10 km:iin. [32; 33.]



Kuva 12. SIGFOX-arkkitehtuuri, jossa laitteet kytkeytyvät verkkoon tukiasemien kautta [34].

Kuten näistä lyhyistä esittelyistä voi jo päätellä, siirtoyhteysprotokollia on melkoinen määrä useisiin käyttötarkoituksiin, ja niitä muodostuu koko ajan lisää tekniikoiden ja tarpeiden kehittyessä. Monessa standardissa myös ilmoitetaan kantamat, jotka eivät vastaa aina todellisuutta. Muistettava on, että tulokset eivät välttämättä valehtele, sillä yleensä nämä valmistajien antamat arvot pätevät parhaassa mahdollisessa ympäristössä. Kun lisätään maasto, fyysiset esteet ja sivullisten laitteiden aiheuttamat häiriöt, saadaan todelliset tulokset.



## 2.2 Verkkokerroksen reititys- ja kapselointiprotokollat

### Reititysprotokollat

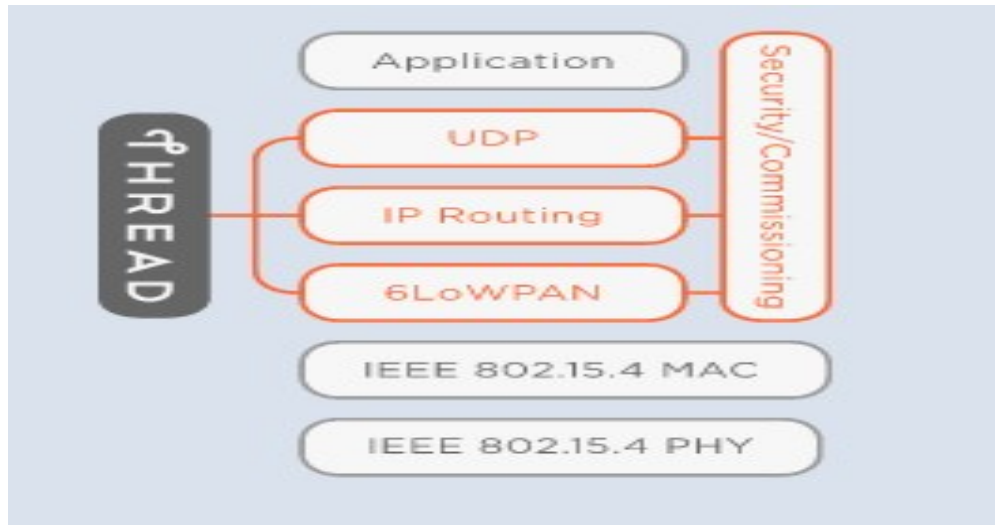
Reititysprotokollien tehtävänä on vastata liikenteen reitityksestä laitteiden välillä ja niitä ohjaavista algoritmeista. Osa protokollista toimii osittain myös siirtoyhteyskerroksella, mutta OSI-mallissa ne sijoittuvat pääasiassa kolmannen eli verkkokerroksen alle.

### RPL

RPL on ROLL-työryhmän luoma protokolla IPv6-reititykseen vähäisen tehon häviöllisissä verkoissa (LLN). Nämä verkot koostuvat usein laitteista, joilla on rajallinen laskenta- ja datansiirtokyky. Nämä verkot ovat usean laitteen kattavia, jopa tuhansia yhdessä kokonaisuudessa, ja tämä tuo haasteita erityisesti reitityksen kannalta. Yleisin RPL-järjestelmä on koottu laitteista, jotka on kytketty toisiinsa useamman hypyn ja juurilaitteiden kautta. Nämä juurilaitteet vastaavat datan koonnista ja hallinnointitehtävistä. Niille luodaan Destination Oriented Directed Acyclic Graph (DODAG), joka kartoittaa reitin optimointivaatimukset. RPL voi kierrättää useita viestintä- ja liikennöintitietoja laitteiden välillä riippuen datavirran vaatimuksista. Se tukee monesta yhteen pisteeseen- yhdestä moneen- ja yhdestä yhteen -liikennettä. [35.]

### Thread

Thread on melko uusi protokolla, joka julkaistiin vuonna 2014. Protokolla tähtää pääasiassa älykotiympäristöön. Tämä 6lowPAN:iin ja IEEE802.15.4:iin perustuva protokolla (kuva 13) on luotu täydentämään WiFi:ä, joka on puutteista huolimatta yleisyytensä vuoksi melko otollinen kodinautomaatioon. Protokolla on myös melko helppo implementoida IEEE802.15.4-yhteensopivaan laitteeseen sovelluspäivityksellä. Se tukee silmukka-arkkitehtuuria ja kykenee hallitsemaan jopa 250 korkean turvallisuustason laitetta. [36; 37.]



Kuva 13. Thraedin sisältö ja sijoittuminen verkkokerroksilla (reititys) [37].

## CORPL

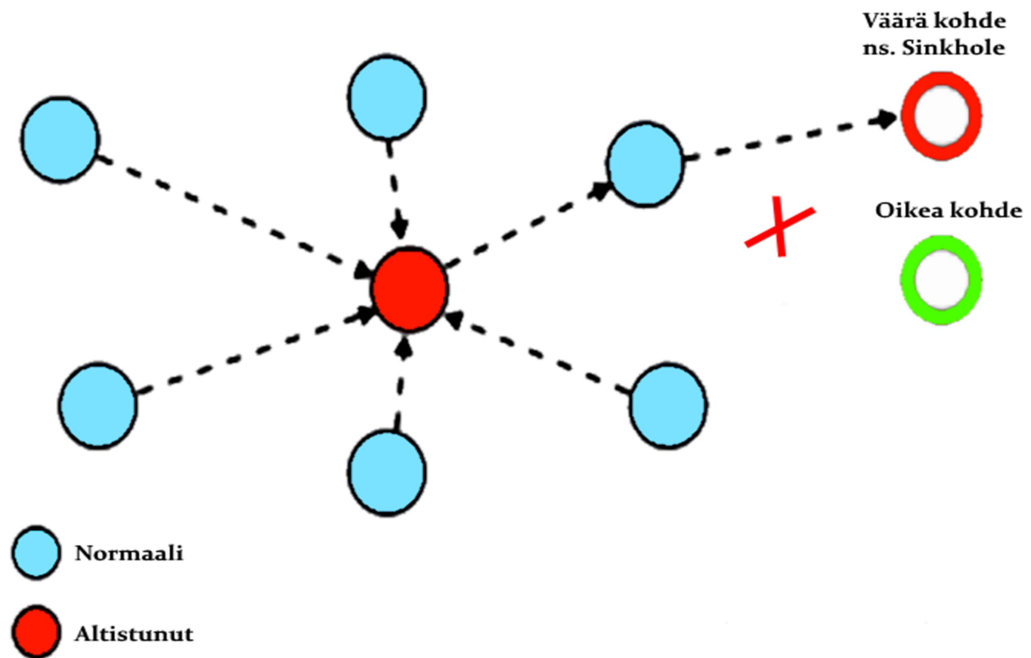
CORPL-protokolla on kognitiivisiin verkkoihin suunniteltu RPL:n lisäosa, joka myös hyödyntää DODAG-topologiaa. Se hyödyntää opportunistista lähestymistapaa. Järjestelmässä laitteilla on useita edelleenlähetyskokoelmia ja koordinoitikaavio, joka huolehtii, että vain parhaanlaatuinen vastaanottaja lähettää viestiä edelleen. Jokainen laite pitää itsenäisesti yllä edelleenlähetyskokoelmaa ja päivittää sitä tarvittaessa naapurilaitteille, jotka päivittävät sitten oman kokoelmansa tämän mukaan. [38.]

## CARP, R-CARP

Tarjotakseen korkean saatavuuden verkossa CARP sallii usean isännän jakaa sama IP-osoite ja virtuaali-isäntätunnus. Tällöin käyttäjä ei huomaa muutaman isännän toimimattomuutta, vaan muut ottavat haltuun niiden tehtävät. Jaetun IP-osoitteen lisäksi jokaisella isännällä on myös laitekohtainen IP-osoite huolto- ja säätötoimintoja varten.

Itsessään CARP ei ole niinkään käytetty IoT-ympäristöissä, koska se on altis reititysvirhehyökkäyksille (kuva 14). Sen päälle on rakennettu R-CARP, joka tarjoaa tähän rat-

kaisuksi mainepohjaisen reitityksen ja lyhyen allekirjoituksen, jolloin liikenteelle haitalliset laitteet voidaan sulkea pois verkosta. R-CARP on käytössä esimerkiksi vedenalaisissa anturiverkoissa, joihin sen kerrotaan soveltuvan kilpailijoitaan paremmin turvallisuusominaisuuksien vuoksi. [39; 40.]



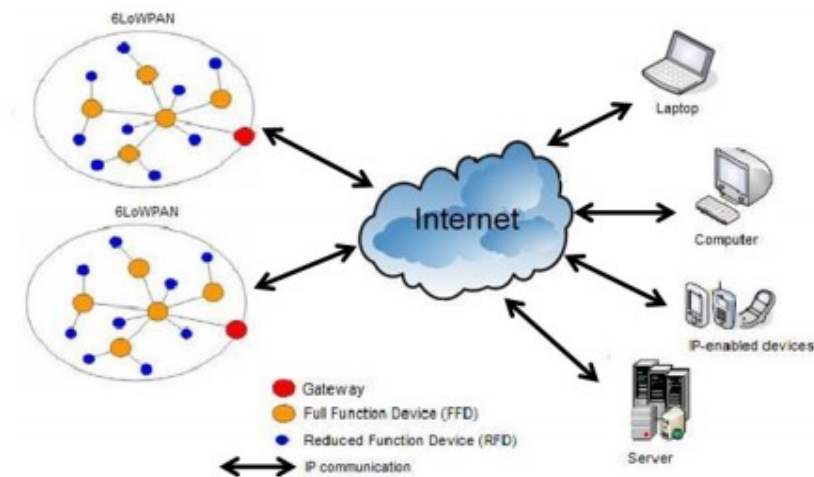
Kuva 14. Yksinkertainen esitys reititysvirhehyökkäyksestä (engl. sinkhole attack), jossa altistunut laite ohjaa datavirrat virheelliseen kohteeseen [40].

## Kapselointiprotokollat

Kapselointiprotokollien tärkein tehtävä on mahdollistaa tehokas kommunikointi tarjoamalla ratkaisu IoT:n ja IPv6:n osoitteisto-ongelmaan. Useissa teknologioissa datalinkin kehyksen koko ei ole riittävä, ja siksi IETF on kehittänyt seuraavia standardeja ratkaisuksi.

## 6LoWPAN

6LoWPAN on yksinkertainen verkko, joka toimii yhdyskäytävänä ja tarjoaa IPv6:n toiminnallisuudet IEEE 802.15.4-verkkojen päälle. Verkko muodostuu vajaan ja täyden toiminnallisuuden lyhyen kantaman alhaisen suorituskyvyn laitteista. Täyden toiminnallisuuden laitteet yhdistävät vajaan verkkoon ja yhdyskäytävään. Yhdyskäytävä kytkee kokonaisuuden verkkoon, sillä on IPv6-verkkotunnus ja vastaa kommunikoinnista IP-tekniikkaan pohjautuvien laitteiden kanssa verkossa (kuva 15). [3; 41.]



Kuva 15. 6LoWPAN-arkkitehtuuri [41].

## 6TiSCH

6TiSCH on IETF:ssä kehitteillä oleva standardi, jonka tarkoituksena on mahdollistaa IPv6-kommunikointi hyödyntäen IEEE802.15.4e:n TSCH-tilaa. 6TiSCH-arkkitehtuuri määrittää kuinka tiettyyn deterministiseen IPv6-virtaan kuuluvat paketit merkitään ja reititetään tai välitetään. Se kattaa IPv6-verkkokerroksen salauksen ja linkkien hallinnan. Naapurilaitteiden havaitseminen ja reititys sisältyvät standardiin. [42; 43; 44.]

## 6Lo

6Lo-työryhmä muodostui kun ymmärrettiin, ettei 6tisch ja 6LowPan yksistään riitä. Esimerkkeinä tästä muodostuivat IPv6 over Bluetooth LE, IPv6 over IEEE485 MS/TP-verkoissa, IPv6 over DECT/ULE, IPv6 over NFC, IPv6 over IEEE802.11ah ja IPv6 over (WIA-PA) 6Lo. Edellä mainitut protokollat ovat kehitteillä määrittämään tavan välittää IPv6-datasähkeitä niille määriteltyjen datayhteyksien kautta. Näistä RFC-hyväksytyt ovat IPv6 over G.9959 ja IPv6 over Bluetooth LE. [45.]

### IPv6 over G.9959

RFC 7428:ssa headerin pakkaukseen hyödynnetään samaa tekniikka kuin 6lowPAN. RFC 7428 määrittää kehysformaatin IPv6-pakettien lähettämisen, ja se myös määrittää jaetun verkkoavaimen turvallisuustason. Vaativampaa salausta vaativissa sovelluksissa se täytyy toteuttaa erillään. [3; 45.]

### IPv6 over Bluetooth Low Energy

RFC 7668 määrittelee IPv6 over Bluetooth LE:n, ja se hyödyntää suurimmaksi osaksi 6lowPAN-tekniikoita pakkauksessa, mutta fragmentaatio-ominaisuudet ei ole käytössä. Vähävirtaisten laitteiden reititys tehdään keskitetyn solmun kautta. [3; 45.]

### 2.3 Istuntokerroksen protokollat

Istuntokerros on OSI-mallissa viides kerros. Se ottaa kantaa ohjelmistosovelluksien toteutuksiin ja niiden yhdistämiseen, mutta ei vastaa itse verkon toteutuksista. Näiden protokollien pääasiallinen tehtävä on aloittaa, hallita ja lopettaa istuntoja. [46.]

#### MQTT

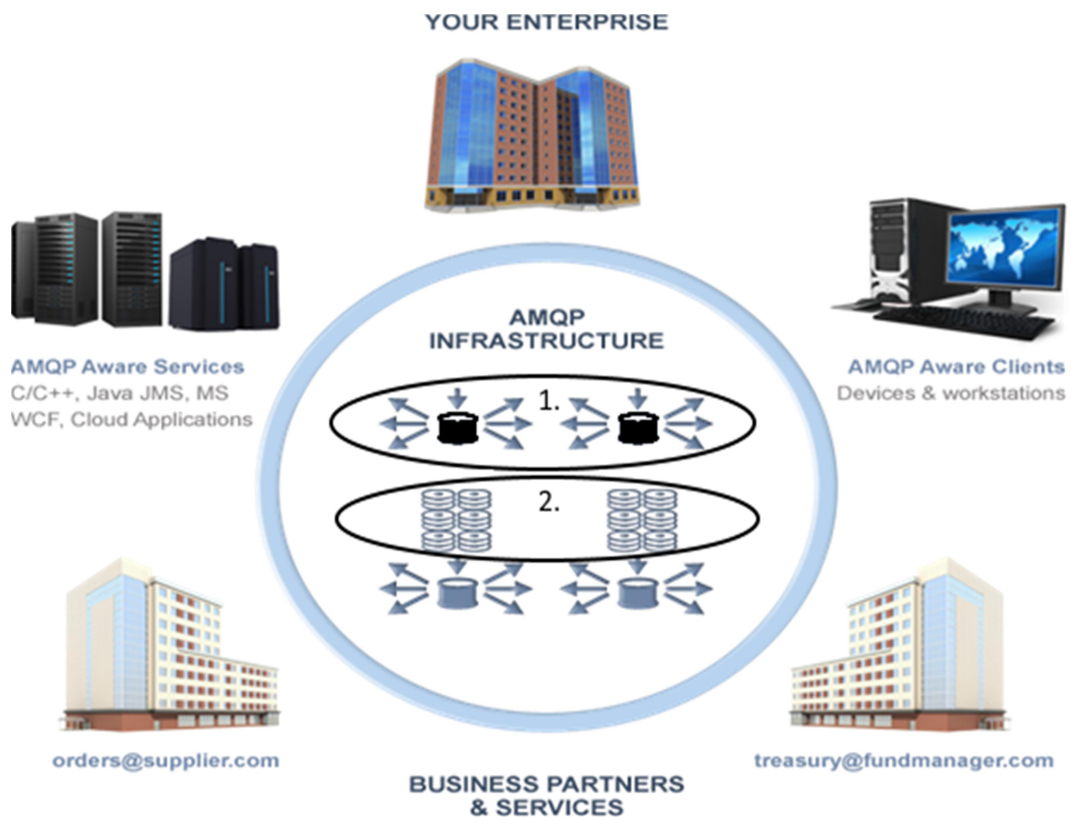
Message Queue Telemetry Transport on käyttäjä-palvelinrajapinnan viestinvälitysprotokolla. Se kehitettiin IBM:n toimesta vuonna 1999 ja OASIS standardisoi sen noin kolme vuotta julkaisusta. Kehitettäessä käyttötavoitteena oli saavuttaa protokolla öljyteollisuuden tarpeisiin, jolla on minimaalinen tehonkulutus ja kaistanleveys. Protokolla noudattaa julkaisu-tilaaja-arkkitehtuuria (kuva 16). Arkkitehtuurissa on kolme toimijaa, julkaisija, välittäjä ja tilaaja. IoT:ssä vähävirtaiset sensorit ovat julkaisijan osassa: ne lähettävät datan ja siirtyvät lepotilaan. Välittäjä toimittaa datan edelleen sitä hyödyntäville tilaajille. [47; 48; 49.]

| Control packet | Direction of flow                          | Description                                |
|----------------|--|--|
| CONNECT        | Client to Server                           | Client request to connect to Server        |
| CONNACK        | Server to Client                           | Connect acknowledgment                     |
| PUBLISH        | Client to Server<br>or<br>Server to Client | Publish message                            |
| PUBACK         | Client to Server<br>or<br>Server to Client | Publish acknowledgment                     |
| PUBREC         | Client to Server<br>or<br>Server to Client | Publish received (assured delivery part 1) |
| PUBREL         | Client to Server<br>or<br>Server to Client | Publish release (assured delivery part 2)  |
| PUBCOMP        | Client to Server<br>or<br>Server to Client | Publish complete (assured delivery part 3) |
| SUBSCRIBE      | Client to Server                           | Client subscribe request                   |
| SUBACK         | Server to Client                           | Subscribe acknowledgment                   |
| UNSUBSCRIBE    | Client to Server                           | Unsubscribe request                        |
| UNSUBACK       | Server to Client                           | Unsubscribe acknowledgment                 |
| PINGREQ        | Client to Server                           | PING request                               |
| PINGRESP       | Server to Client                           | PING response                              |
| DISCONNECT     | Client to Server                           | Client is disconnecting                    |

Kuva 16. MQTT:n julkaisu-tilaaja-arkkitehtuurin kommunikointi [49].

## AMQP

The Advanced Message Queuing Protocol on avoin turvallinen binäärinen siirtoprotokolla, joka toimii TCP:n päällä ja on suunniteltu erityisesti finanssialan tarpeisiin. Suurimpana erona MQTT-protokollaan on välittäjän rakenne. AMQP:ssä välittäjä jakautuu kahteen osaan, pörssi ja aihejonot (kuva 17). Pörssi vastaa julkaisijoiden viestien siirosta oikeisiin jonoihin ennalta määriteltujen ehtojen mukaisesti. Protokolla ei ole sidottu mihinkään tiettyyn topologiaan, lähteeseen tai vastaanottomalliin. Protokollaa hyödyntää usea suuri finanssialan toimija, esimerkiksi Barclays ja Bank of America. [50; 51; 52.]



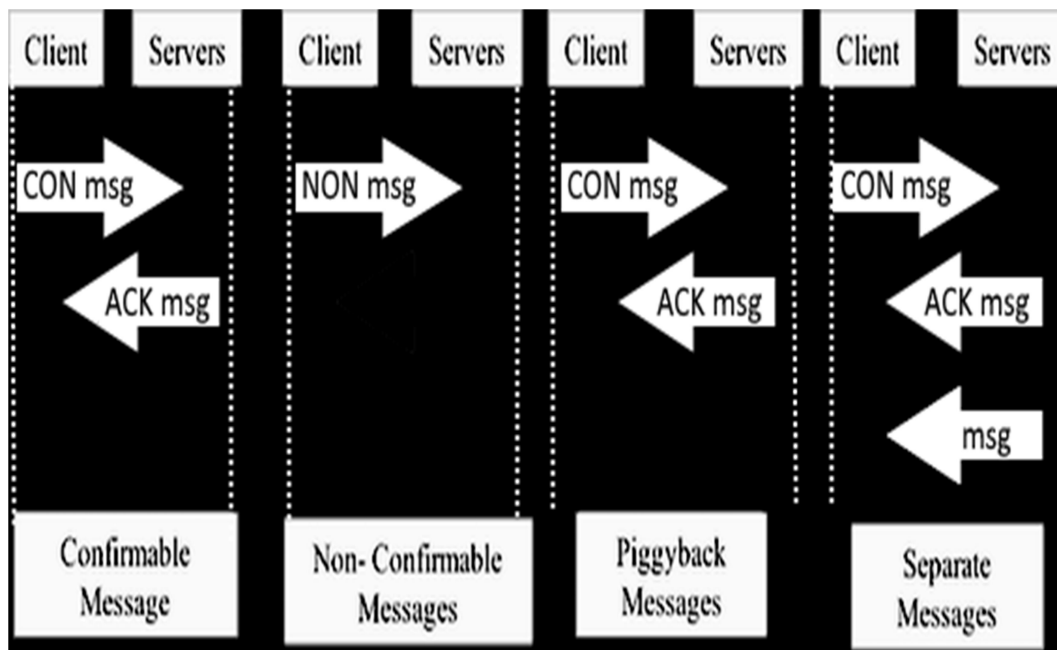
Kuva 17. AMQP:n infrastruktuuri, jossa pörssi on esitetty kohdassa 1 ja jonot kohdassa 2 [52].

## CoAP

Constrained Application Protocol on erikoistunut web-siirtoprotokolla. Se perustuu samaan REST-malliin kuin HTTP ja on suunniteltu erityisesti laitteiden väliseen kommu-

nikointiin. Kehittäjän kannalta protokolla on yksinkertainen, ja datan vastaanotto sensoreilta ei eroa huomattavasti WEB-API:n kautta tapahtuvasta.

CoAP on rakennettu UDP:n päälle yleisesti HTTP:ssä käytetyn TCP:n sijasta. CoAP sisältää neljä viestimallia (kuva 18), confirmable ja non-confirmable, jotka edustavat vahvistettua ja vahvistamatonta lähetystä, piggyback, jota käytetään asiakas-palvelin kommunikointiin. Tässä palvelin vastaa suoraan saatuaan viestin. Separate-mallissa palvelin lähettää ensin vahvistuksen ja vasta sen jälkeen viestin. Se on useimmiten käytössä kun viestin lähettäminen tai muodostaminen kestää kauemmin palvelimen päässä. [3; 53.]



Kuva 18. CoAP-viestimallit [53].

## XMPP

XMPP on avointa lähdekoodia tukevan Jabber-yhteisön kehittämä avoin TCP-pohjainen protokolla pikaviestintään. Se tukee arkkitehtuureista molempia, julkaisija-tilaaja- ja vaatimus-vastausmallia. Protokolla on suunniteltu lähinnä reaaliaikasovelluk-



sille, mutta tukee tehokkaasti lyhyiden laiteiden keskinäistä viestien välitystä (m2m). Kuitenkaan se ei ole vielä saavuttanut suurta suosiota IoT:n puolella muutamien puutteiden vuoksi: esimerkkinä mainittakoon käytettävien XML-viestien prosessointivaatimusten luoma tehonkulutus ja QoS:n puute. [3; 54; 55.]

## DDS

The Data Distribution Service on välittäjä-tason protokolla, joka integroi järjestelmän komponentit yhteen ja vastaa IoT:n vaativiin tarpeisiin. Protokollan datakeskeisyys on ideaali IoT-puolella, ja se tarjoaa 23 palvelunlaatutasoa (QoS). DDS:ssä on määritelty kaksi alitasoa datakeskeinen julkaisu-tilaaja- ja paikallinen datarekonstruointikerros. Ensimmäinen vastaa viestien toimittamisesta ja tilaaja hallinnasta ja toiseksi mainittu hoitaa tarvittaessa yksinkertaisen integroinnin sovelluserrokseen. [1; 56; 57.]

## Yhteenveto

Protokollia on useita erilaisia, ja tässä niistä oli mainittu vain muutamia. Näistä voidaan valita sopiva vaatimusten mukaan (kuva 19), esimerkkinä XMPP:n XML-viestien tuki ja CoAP REST-toiminnot. Mainituista yleisimmin IoT:ssä käytetty on alhaisen tehonkulutuksen omaava MQTT-protokolla.

| Protocols | UDP/TCP | Architecture | Security and QoS | Header Size (bytes) | Max Length(bytes) |
|-----------|---------|--------------|------------------|---------------------|-------------------|
| MQTT      | TCP     | Pub/Sub      | Both             | 2                   | 5                 |
| AMQP      | TCP     | Pub/Sub      | Both             | 8                   | -                 |
| CoAP      | UDP     | Req/Res      | Both             | 4                   | 20 (typical)      |
| XMPP      | TCP     | Both         | Security         | -                   | -                 |
| DDS       | TCP/UDP | Pub/Sub      | QoS              | -                   | -                 |

Kuva 19. Luvussa 2.3 käsiteltyjen standardien ominaisuuksien vertailu [3].

## 2.4 IoT ja tietoturva

Tuoreena trendinä IoT on tuonut uusia uhkia ja jossain määrin niitä vähätellään älylaitemarkkinoilla. Teollisuuden ja terveysteknologian puolella nämä vaarat otetaan huomioon, sillä näillä alueilla laitteen tai järjestelmän hallinnan joutuminen väärälle taholle voi johtaa jopa ihmishenkien menetykseen. Esimerkiksi tehtaan palohälytysjärjestelmän tai kriittisiä elintoimintoja tukevan järjestelmän deaktivointi voi johtaa mittaviin menetyksiin. Uuhiin kuitenkin onneksi suhtaudutaan vakavasti, ja haavoittuvuuksia tutkitaan ja testataan ennen tuotantoon saattamista.

Tietoturvaongelma kohdistuu erityisesti yksinkertaisiin kodin pienälylaitteisiin, sillä usein ajatellaan, että yksinkertaisesta kodinkoneesta tai lelusta voisi olla haittaa. Tämä kuitenkin on osoitettu jo epätodeksi useaan otteeseen, viimeisimpänä varmaan mainittavista bottiverkon rakentaminen ja valjastaminen osaksi suurempaa palvelunestohyökkäystä. Palvelunestohyökkäykset eivät usein tuota pitkäaikaista haittaa, mutta hyökkäyksen kohteelle se on aina tappio imagollisista ja taloudellisista näkökulmista. Tässä vaiheessa voidaan miettiä, kenelle vastuu näistä tappioista siirtyy, sillä usein hyökkääjää ei saada oikeuteen ja kohde ei ole voinut estää näitä hyökkäyksiä omilla toimillaan. En usko, että menee kauaakaan, kun hyökkäyksessä käytetyn laitteen valmistajan katsotaan laiminlyöneen vastuutaan ja korvaukset siirretään taholle, jolta voidaan saada taloudellista hyvitystä.

Taloudellisten riskien vuoksi IoT-laitteen turvallisuus on huomioitava joka vaiheessa, kuten sulautettuihin järjestelmiin erikoistunut yritys Windriverkin dokumentissaan mainitsee ja samalla määrittelee viisi huomionarvoista kohtaa:

### Turvallinen käynnistysprosessi

Laitteen ensimmäisellä tai päivityksen jälkeisellä käynnistyskerralla pitää laitteen kyetä varmistamaan ohjelmiston aitous ja koskemattomuus. Tämä varmistetaan esimerkiksi ohjelmiston digitaalisella allekirjoituksella, jolloin laite ei suorita haitallista kolmannen osapuolen sovellusta. [58.]

## Hallinta- ja käsittelyoikeudet

Määritellään tarkasti kenellä, millä ja mistä on oikeus käyttää laitetta. Sekä mihin ominaisuuksiin nämä oikeudet ovat. Esimerkiksi laitteen yksisuuntainen verkkoliikenne, jolloin laite ei suorita haitallisia komentoja tai asenna päivityksiä verkon kautta. [58.]

## Laitteiden todennus

Varmistetaan, että laitteen kytkeytyessä verkkoon tiedetään, että kyseessä on laite X, ja sen tarkoitus siellä. Laitteelle ei siis anneta turhia valtuuksia tai oikeuksia, ja jos mahdollista laite toimii verkossa normaalin käyttäjän tavoin. Tällöin ongelmien koitua se on helppo tunnistaa ja poistaa verkosta. [58.]

## Palomuurit ja tunkeilijan havaitsemisjärjestelmät

Laitteessa on hyvä olla palomuri tai jonkinlainen järjestelmä todentamaan datapaketteja. Tämä pitää kuitenkin hoitaa mahdollisimman vähillä kustannuksilla. Sen ei tarvitse välttämättä puuttua perusviestintään, vaan ainoastaan kriittisten viestien käsittelyyn. Nämä ongelmat saa usein ratkaistua oikeilla protokollavalinnoilla ja turvautumalla valmiisiin toimiviksi todettuihin tietoturvaratkaisuihin. [58.]

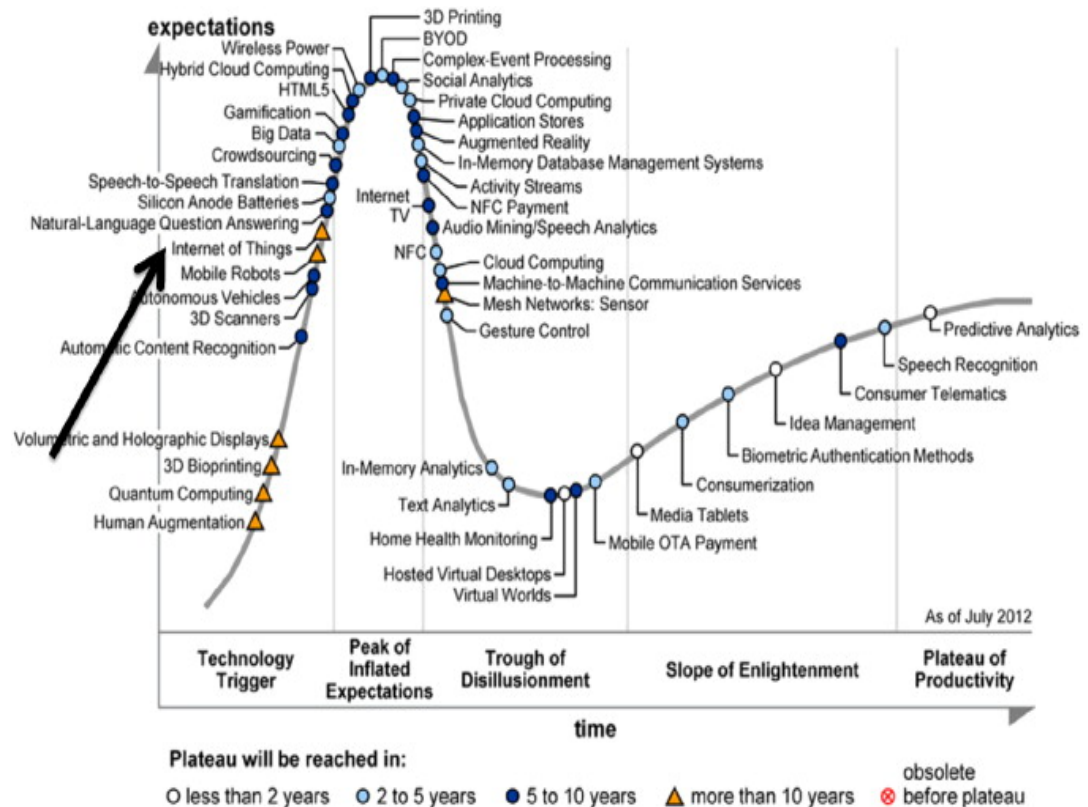
## Päivitykset ja korjaukset

Järjestelmäpäivitykset ohjelmiston uudet versiot on kyettävä hoitamaan sulavasti, ja toiminnallisuuksiin vaikuttavien osien osalta on kyettävä todentamaan oikeellisuus. Parhaaseen mahdolliseen tulokseen päästään, kun tämä pystytään tekemään ilman laitteen poistamista verkosta ja vaarantamatta laitteen koskemattomuutta. Käytännössä päivitys tehdään salattua tai muutan turvattua väylää pitkin.

Pelkästään näiden kohtien läpi käynnillä saadaan parannettua turvallisuutta ja vältetään suurimmilta uhilta eikä anneta hyökkääjille helppoa väylää. Paras suhtautumien älylaitteeseen on suhtautua kuin ummikkoon tietojärjestelmän käyttäjään: ei anneta liikaa oikeuksia ja vastuuta, ja rajoitetaan tarpeettomat toiminnot. Kuudenneksi vielä lisäksi käyttäjien valistuksen tilanteissa ja järjestelmissä, joissa se on mahdollista. Esimerkkinä valistus siitä, mitä voi seurata henkilökohtaisen älylaitteen tahattomasta kytkemisestä yritysverkkoon. [58.]

## 2.5 Haasteet ja tulevaisuus

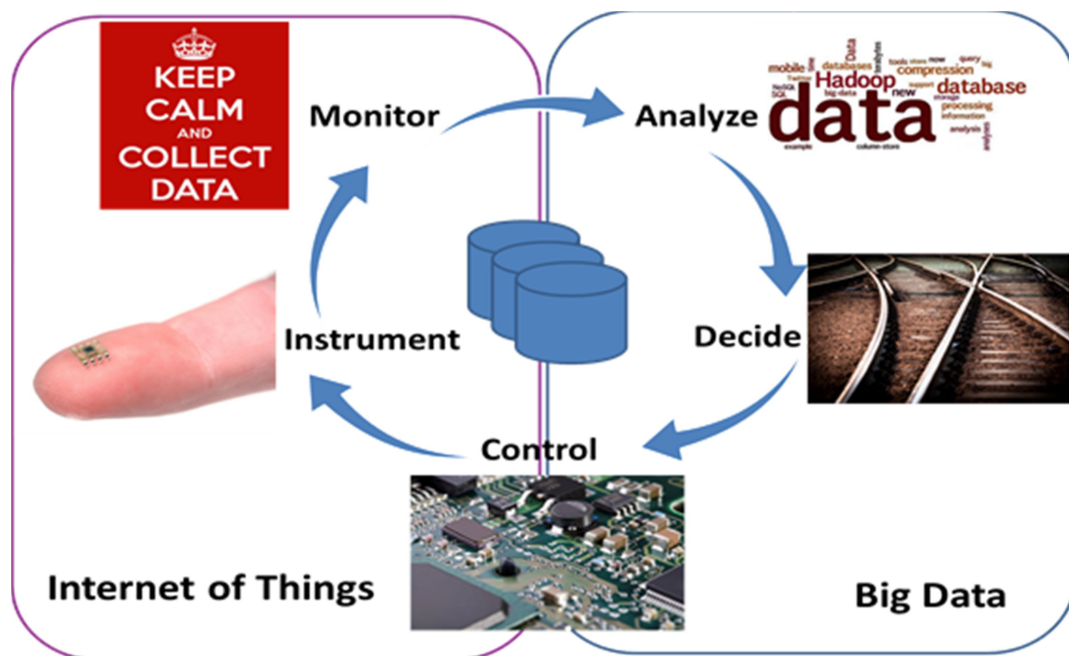
IoT, pilvi ja big data ovat kiistämättä trendejä tällä hetkellä. Kuitenkin seuraavan 5–10 vuoden sisällä nähdään pieni romahdus näissä trendeissä, mikäli Gartnerin IT-hypen kiertoarvioon on uskominen (kuva 20). Pian tullaan pisteeseen, jossa teollisuudella on paljon ideoita ja toteutuksia, joita ei välttämättä enää tai vielä tarvita. Tietokannat ovat täynnä sensoridataa, jota ei voida tai saada käyttää hyväksi missään.



Kuva 20. Gartnerin arvio IoT-trendeistä vuodelta 2012 [59].

IoT on tuonut uudet mahdollisuudet kerätä tietoa. Voidaan jo puhua tietofarmeista (kuva 21), ja ne toimivat melkein kaikkialla verkon laajan kattavuuden vuoksi. Pelkästään matkapuhelimet voivat kerätä suuria määriä dataa, ja kun tähän liitetään ostotottumukset etukorteilta, voidaan koota melko tarkka profiili henkilöstä. Tämä profiili on monelle yritykselle kultakaivos. Se mahdollistaa kohdenetun mainonnan, ja jopa tuotekehityksen oikean suunnan määrittämisen kohtalaisin alhaisin kuluin.

Kuka omistaa sensorien tai laitteiden keräämän datan, ja onko sen tuottajalla oikeus korvauksiin? Minkälainen data on yksityistä? Kun ihmiset ymmärtävät antamiensa tietojen arvon yrityksille, se voisi melko varmasti johtaa kehityksen jarrutukseen, pisteeseen, jossa säädetään pelisäännöt. Tämä tulee hidastamaan kehitystä ja innovointia, koska yritykset eivät uskalla eikä niiden kannata lähteä hakemaan ennakkotapauksia oikeudesta.



Kuva 21. IoT ja Big data toistensa tukena. Yksinkertaistettu datafarmi, joka kerää tietoa IoT-laitteen avulla ja tekee päätökset toimista analysoinnin tulosten perusteella [60].

Suurimmat ongelmat ovat itsenäisten laitteiden keräämän datan oikeuksissa, sillä laitteiden on mahdollista kerätä dataa, jonka antamiseen ei pystytä pyytämään lupaa, toisin kuin esimerkiksi etu- ja jäsenkorttien kohdalla. Tässä datankeräyksessä on kuitenkin vielä pysyttävä varovaisena niin kauan, kunnes saadaan jonkinlaiset lailliset kehykset, missä toimia. [59; 60; 61; 62.]

### 3 Prototyypin laitteiston määrittely

Insinööriyössä tehdyn perehtymisen jälkeen, lähdin toteuttamaan itsenäistä anturijärjestelmän prototyyppiä. Työ aloitettiin määrittelemällä laitteelle vaatimukset, jotka työnsäni olivat kustannustehokkuus, saatavuus ja autonominen toiminta. Nämä asetettiin laitteiston valinnassa päätekijöiksi.

#### 3.1 Laitteiston käytettävyys ja valintaperusteet

Ensimmäisen vaiheen jälkeen kun määrittelyn runko oli saatu kokoon, oli vuorossa osien ja tekniikoiden valinta. Laitteiston osalta lähdettiin liikkeelle saatavuuden, hinnan ja virrankulutuksen ehdoilla. Tähän päädyttiin, koska kyseessä olevan prototyypin tulisi olla mahdollisimman itsenäinen laite, jonka pystyy toteuttamaan kustannustehokkaasti.

Mikro-ohjainta ja kehitysalustaa lähdin pohtimaan kokonaisuuden vaatimusten pohjalta ja tiukan aikataulun takia valitsin vaihtoehtoiksi kolme itselleni kohtalaisen tuttua kehitysalustaa: ATmega328p (Arduino Uno), PSoC1 (CY7C64215-28PVXC) ja PSoC4 (CY8CKIT-049-4200). Kaikissa alustoissa oli hyvät liitännät ja laaja valikoima erilaisia väylämahdollisuuksia, joten lähdin pohtimaan muita aspekteja valintaa tehdessä.

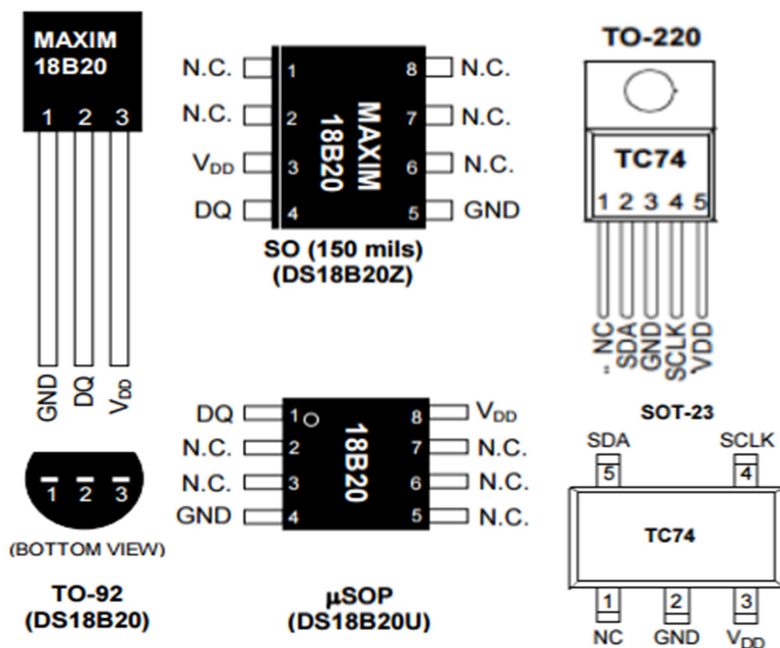
PSoC1:ssa ja Arduinossa on vain 8-bittinen prosessori, joten PSoC4:n 32-bittinen prosessori ja korkeampi kellotaajuus ottivat suorituskyvyn osalta voiton (taulukko 1). Myös virrankulutuksen osalta näyttivät tulokset kääntyvän PSoC4-kehitysalustan valinnan suuntaan (taulukko 1). PSoC1-alustan jätin pois huonomman kehitysympäristön vuoksi ja koska virrankulutus suhteessa suorituskykyyn on PSoC4-alustan puolella. Arduino Uno oli edukseen valmiiden kirjastojen, esimerkkien suuren määrän ja valmiiden helposti kytkettävien moduulien osalta. Kuitenkin päädyin vähävirtaisuuden, edullisimman hankintahinnan ja tutumman kehitysympäristön takia PSoC4 CY8CKIT-049-4200 -kehitysalustaan. [63; 64; 65; 66.]

Taulukko 1. Kehitysalustojen viitteelliset virrankulutusarvot ja suorituskyky [63; 64; 65].

| Mikro-ohjain                | Virransyötön tila | Virta (mA) | MIPS/MHz | MHz (up to) |
|-----------------------------|-------------------|------------|----------|-------------|
| ATmega328p<br>(Arduino Uno) | Aktiivinen        | 46,5*      | 1        | 20          |
|                             | Sleep             | 32*        |          |             |
| PSoC1                       | Aktiivinen        | 20         | 0,17     | 24          |
|                             | Sleep             | 3,4        |          |             |
| PSoC4                       | Aktiivinen        | 14         | 0,9      | 50          |
|                             | Sleep             | 3          |          |             |

\*Virta-arvot ovat tietyn alustan viitteelliset ja saatavana on myös vähävirtaisempia malleja.

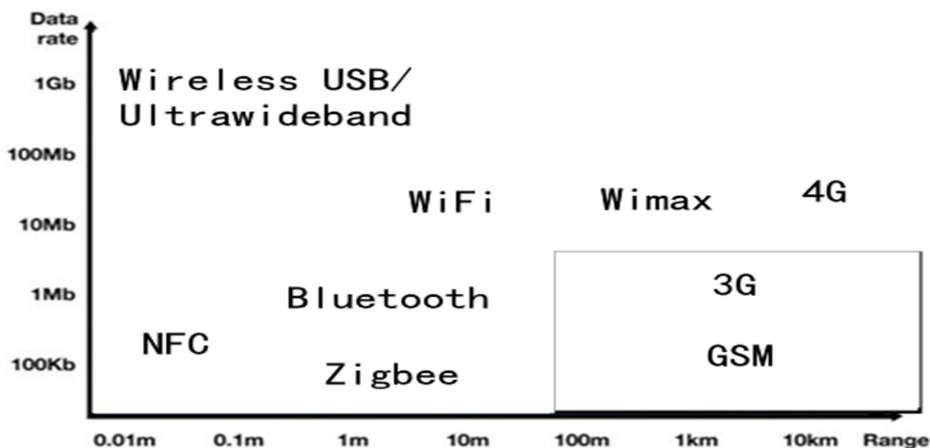
Anturin valinnassa lähdettiin ajatuksesta, ettei haluta erillisiä AD-muuntimia laitteeseen tai väylään. Tässä karsiutui osa valmiina olleista lämpöantureista, ja saatavilla olevista jäi kaksi vaihtoehtoa, TC74A2 ja DS18B20 (kuva 22), joista molemmat operoivat halutulla lämpötila-alueella. Lämpötilanmittaus ja muunnosnopeudet erosivat antureissa suuresti. Datalehden perusteella TC74A2 oli valmis jo 250 ms:n jälkeen ja DS18B20 vasta 750 ms:n. Nopeus ei tässä kuitenkaan ollut kriittisenä perusteena, vaan toisena vaatimuksena oli laitteen muokattavuus.



Kuva 22. Lämpöantureiden eri koteloinnit ja jalkojen järjestykset [67; 68].

Muokattavuutta lähdettiin arvioimaan dataväylän ja virransyötön kannalta. Vastakkain asettuivat I2C- ja 1-Wire-väylät. Valitussa mikrokontrollerissa oli suoraan kirjastot I2C:lle ja väylänopeudet olivat suuremmat. Kuitenkin 1-Wiren eduksi osoittautui sen helppo toteutus ja pitemmät välimatkat. Tulevaisuudessa 1-Wire-väylään saa lisättyä todella helposti antureita tai vaikka oman orjalaitteen, ilman osoitteistoyhteentörmäyksiä (TC74 7bit vastaan DS18B20 64bit). Valinta kallistui myös I/O-pinnien tarpeen kannalta 1-Wire väylään ja päädyin DS18B20 lämpösensoriin. [65; 66.]

Seuraavaksi valintavuorossa oli se, millä tavalla yhdistän laitteet verkkoon. Käsitellyistä löytyi paljon vaihtoehtoja, mutta suurin osa oli niin keskeneräisiä tai niiden toteuttaminen ja integrointi järjestelmään vaati paljon työtä jo olemassa oleviin todella hyvin dokumentoituihin tekniikoihin verrattuna. Näistä mainittavia olivat radiolähetys, Wi-Fi, 3G ja GSM/GPRS sekä muutamat hylkäämäni tekniikat, Bluetooth LE, NFC, ZigBee ja Wireless USB, koska niiden kantama rajoittuu selvästi alle tai maksimissaan 100 metriin (kuva 23). Moduulien hintahaarukka liikkuu edullisten vaihtoehtojen osalta 3 ja 60 euron välillä. Seuraavaksi piti pohtia, onko tarkoituksenmukaista käyttää laitetta asutuksen lähellä, jolloin voidaan siirtää data tukiaseman kautta, vai yhdistetäänkö laite suoraan verkkoon.



Kuva 23. Suuntaa antavaa vertailua tekniikoiden ja kantamien välillä. Neliön sisältö kuvaa riittäviä arvoja tähän käyttötarkoitukseen.

Kun prototyypin tarkoituksena oli tehdä autonominen sensoryksikkö, Wi-Fi tai radiolähetys eivät olleet vaihtoehtoina toimivia lyhyen kantamansa ja erillisen tukiaseman tarpeen takia. Tarkastelun alle otin siis moduuleista LTE 4G:n, 3G:n ja GSM/GPRS:n.

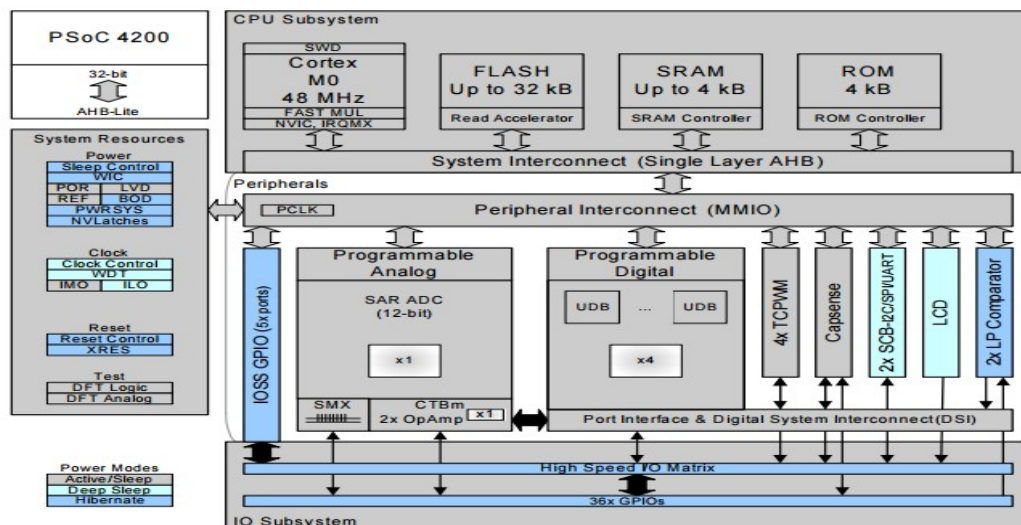


Kaikki kolme moduulia kykenevät siirtämään käsiteltävät datat riittävällä nopeudella. Saatavilla olevista vaihtoehdoista otin käsittelyyn kolme, SIM900 noin 5–10 €, SIM5320 noin 22–40 € ja SIM7100C noin 48–55€. Hinta päättyi olemaan ratkaiseva valintaperuste, ja valinta oli SIM900-moduuli. Suurempia datamääriä siirrettäessä valinta olisi kohdistunut SIM7100C 4G -moduuliin. [69; 70; 71.]

### 3.2 PSoc4-mikro-ohjain

PSoC (*Programmable System-on-Chip*) -mikro-ohjaimet ovat Cypress Semiconductor Corporationin kehittämä tuoteperhe. Cypress aloitti 8-bittisen PSoC1-piirin toimitukset vuonna 2002. Tuoteperheeseen kuuluvat myös 8-bittinen PSoC3 ja ARM-pohjaiset 32-bittiset PSoC4 ja PSoC5/5LP.

PSoC4 on muunneltavissa oleva ARM® Cortex™-M0 -suorittimella varustettu sulautettu kehitysympäristö, joka sisältää ohjelmoitavia digitaalisia ja analogisia lohkoja, joissa on joustava automaattinen reititys (kuva 24). Säädettävät lohkot ovat yksi suurimmista eroista muihin markkinoilla oleviin kehitysalustoihin verrattuna. PSoC-kehitysalustaa voisi kuvailla säädettäväksi sovelluskohtaiseksi yhdistelmäpiiriksi. Se sisältää FPGA:lle tyypillisen määrittelyn virran kytkennän yhteydessä, mutta tämä tapahtuu lataamalla ohjeet sisäisestä flash-muistista. PSoC:ssa on lisäksi kaksi muuta muistialuetta SRAM-dataa varten ja I/O-rekisterit lohkojen asetuksille.



Kuva 24. PSoC4200:n lohkokkaavio [66].

PSoC4:ssa on 48 MHz:n kellotaajuudella toimiva Cortex™-M0, joka on tuotepereen pienin tarjolla oleva ARM-prosessori. Pieni piipinta-ala, vähäinen virrankulutus, 32-bittisyys ja vain 56 ohjetta tekee suorittimesta erittäin helposti lähestyttävän. Arkkitehtuuri on C-kieli ystävällinen, jolloin kehitystyöstä tulee helppoa ja nopeaa. [65; 66.]

### 3.3 Väylät

Tiedonsiirtoon laitteiden välillä järjestelmässä tarvitaan sopivia reittejä ja näitä johtimia kutsutaan väyliksi, esimerkkinä osoite-, data- ja ohjausväylät. Niiden toiminnallisuus perustuu tiedonsiirtoon määritellyn protokollaan. Protokolla määrittää loogisen toiminnan, kuten ajastuksen dataväylässä, suunnat, osoitteistot ja toimijat.

Ensisijaisesti käytettävä väylä määräytyy valitun laitteiston mukaan, mutta jos mahdollista on huomioitava käyttötarve, eli tarvittavat siirtonopeudet ja datamäärät. Tällöin saadaan optimoitua kokonaisuus, väylästä ei muodostu järjestelmässä pullonkaulaa ja se ei kuormita turhaan järjestelmää, esimerkiksi vähävirtaiset väyläarkkitehtuurit.

#### 1-Wire

1-Wire on Dallas Semiconductor Corporationin suunnittelema kommunikointiprotokolla, joka on tarkoitettu alhaisille tiedonsiirtonopeuksille ja alun perin tarkoitettu lähekkäin olevien laitteiden kanssa kommunikointiin. Nimensä mukaan väylä mahdollistaa sen, että signaalit ja virransyöttö kulkevat yhdessä kaapelissa. 1-Wire on vuorosuuntainen (engl. *half duplex*) protokolla, ja tällöin kommunikaatio tapahtuu kahteen suuntaan, mutta lähettäviä osapuolia voi olla vain yksi kerrallaan.

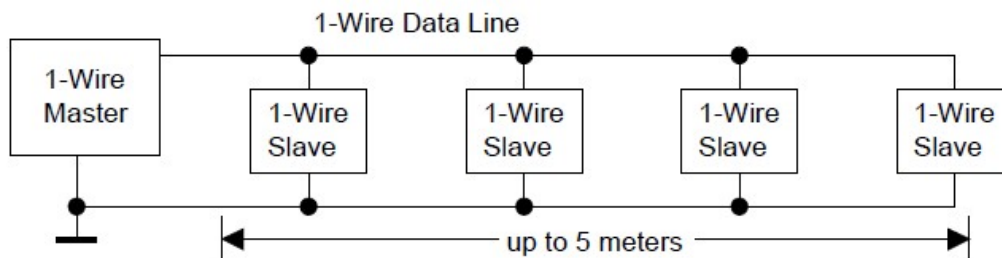
1-Wiressä on vain yksi isäntälaitte, esimerkiksi mikro-ohjain, erikoisvalmisteinen piiri tai tietokone. Se vastaa ohjauksien syötöstä orjalaitteille, joita taas voi olla useita. Protokollalla on samankaltaisia piirteitä I<sup>2</sup>C-väylän kanssa, mutta nopeudet jäävät alhaisemmiksi. Kuitenkin väylä voi olla pitempi kuin I<sup>2</sup>C:ssä. 1-Wire on yleensä sovelluksissa, joissa suuri päivitysnopeus ei ole kriittinen vaatimus, esimerkkinä voi mainita lämpömittarit ja säähavaintolaitteistot.

#### *Rajoitukset ja topologiat*

1-Wire-väylän toimivuuden määrittävät sen kriittiset vaatimukset. Ne ovat etäisyys, joka kertoo välimatkan kauimpana olevaan orjalaitteeseen, ja paino ("johtopaino"), joka ilmaisee verkkoon kytkettyjen kaapeleiden yhteis-pituutta. Myös jokainen orjalaitte (esim. lämpöanturi) lisää kokonaispainoa ja vakioarvona verkkoon lisätään painoa 0,5–1 metriä laitetta kohden.

Topologioiden suhteen 1-Wire on melko vapaa standardi, mutta yleensä rakenne voidaan määrittää kolmeen pääryhmään: lineaarinen, jossa päälinjan muodostaa yksi pari pisimmällä olevalle orjalaitteelle asti ja muut orjat kytkeytyvät siihen alle 3 m pitkillä kaapeloinneilla, katkaistu, joka on muuten samankaltainen kuin lineaarinen, mutta siinä päälinjaan liitetyt orjat ovat 3 m:n tai pitempien sivuhaarojen päissä, ja tähti, väylä jakaantuu lähellä isäntää yhdestä pisteestä useampaan erimittaiseen haaraan. Orjalaitteet sijoittuvat haarojen pätekohtiin.

Verkon painoa rajoittaa kyky, miten nopeasti kaapelin varaustilaa pystytään muuttamaan. Esimerkiksi yksinkertaisella ylösvedolla maksimipaino voi olla 200 metriä, mutta tämä kuitenkin voidaan kiertää kehittyneimmillä järjestelmillä, joissa on käytössä korkeammat virta-arvot ja aktiiviset ylösvedot. 1-Wire-verkon painoa saadaan myös alaispäin käyttämällä kytkinvalitsinta, esimerkiksi kun käytetään 300 metrin painoarvoista kolmihaaratähtiverkkoa. Se saadaan kevennettyä 100 metriin aktivoimalla vain yksi haara kerrallaan. 1-Wire-verkkojen suunnitteluun on annettu ohjesäännöksi noudattaa mahdollisimman kevyttä ratkaisua ja yksinkertaista topologiaa. Työssäni myös päädyin käyttämään lineaarista topologiaa (kuva 25).



Kuva 25. Yksinkertainen lineaarista topologiaa noudattava 1-Wire-kytkentä.

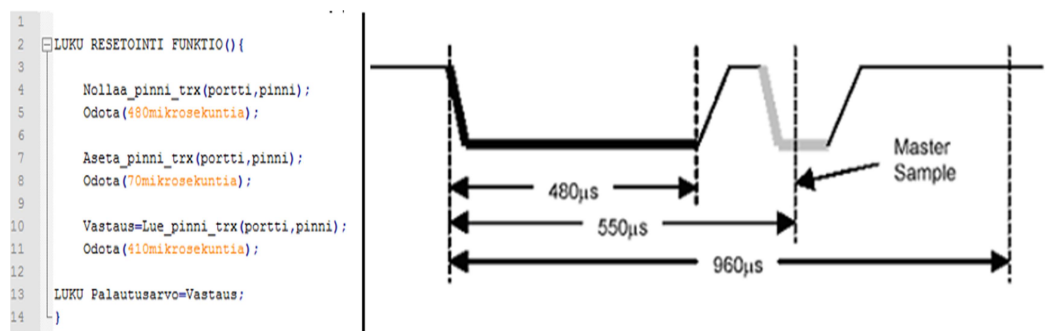
### *Väylän toimintalogiikka*

1-Wire hyödyntää CMOS/TTL-logiikkatasoja, joissa looginen 0 on tasolla alle 0,8 voltia ja looginen 1 tasolla yli 2,2 voltia. Vastuksella linja pidetään lepotilassa resistiivisesti heikosti ylösvedettynä. Kirjoitettaessa ja nollattaessa väylää pakotetaan alas. Tämä keino toimii pienissä verkoissa, mutta suuremmissa kokonaisuuksissa joudutaan turvautumaan ohjattuihin ylösvetopiireihin. 1 bitti lähetetään väylään vetämällä linja alas lyhyeksi ajanjaksoksi, jolloin näytteenotto osuu hetkeen, jolloin linja on arvoltaan 1. 0-bittiä varten linjaa pidetään alhaalla täyden kirjoitusjakson verran, joka on standardilla 60  $\mu\text{s}$  ja nopeutetulla väylällä 8  $\mu\text{s}$ . Standardi väylänopeus on 15,4 kbps ja nopeutetun väylän vastaavasti jopa 125 kbps. Erillistä kellopulssia ei järjestelmässä vaadita, vaan orjat ajastavat toimintansa sisäisellä oskillaattorilla, joka on synkronoitu isännän signaalipulssin laskevaan reunaan.

### Osoitteisto, kommunikaatio ja komennot

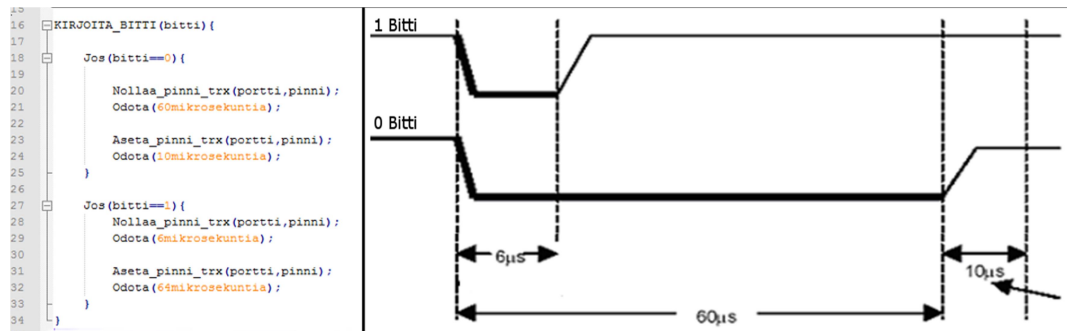
Jokaisella 1-Wire-orjalla on määritelty uniikki 64-bittinen sarjanumero, jota käytetään osoitteena kommunikoitaessa. Osoite koostuu kolmesta osasta, ja se alkaa vähiten merkitsevällä tavulla. Ensimmäinen on 8-bittinen laitetyypin kertova perhekoodi (engl. family code). Seuraavat 48 bittiä ovat osoite ja viimeisenä 8-bittinen CRC tarkistusavainta varten, joka on muodostettu 7 ensimmäisen tavun perusteella.

Kommunikaatio väylässä alkaa linjan nollauksella (kuva 26), isäntä pakottaa linjan alas 480  $\mu\text{s}$  sekunniksi, tämän jälkeen vapauttaa linjan ja odottaa 70  $\mu\text{s}$ . Ajan loputtua isäntä ottaa näytteen, ja jos se on 1, linjalla ei ole orjalaitteita. Jos laitteet ovat läsnä, orjat vastaavat vetämällä linjan alas ja näyte saa arvon nolla. Tämä linjan alustuksen laskeva reuna toimii myös synkronointipulssina orjalaitteille.



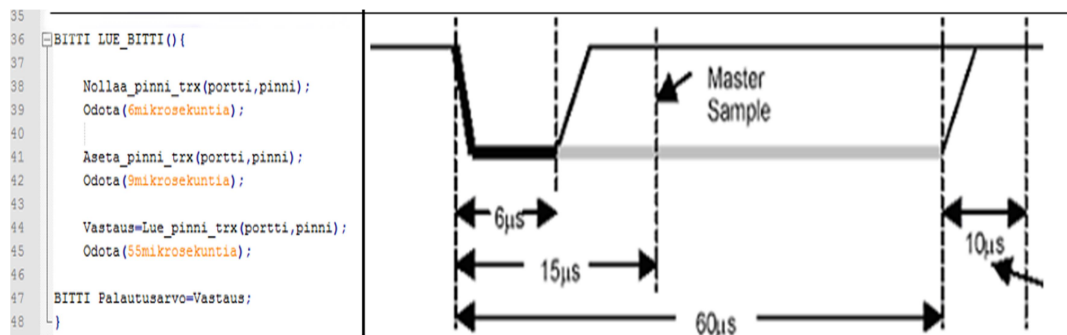
Kuva 26. 1-Wire-nollausoperaation aaltomuoto ja sen pseudokooditoteutus.

Alustuksen jälkeen linjalta valitaan laite lähettämällä parituskomento ja sen jälkeen osoite linjaan, jolloin muut orjat menevät odotustilaan. Seuraavaksi lähetetään laitekoh-tainen komento ja tarvittaessa luetaan data väylästä. Kirjoitettaessa väylään linja pako-tetaan alas 6  $\mu$ s:ksi, minkä jälkeen se palautetaan joko ylös vastaamaan arvoa 1 tai pidetään alhaalla arvossa 0 ja palautetaan ylös 54  $\mu$ s:n jälkeen. Kokonaisuudessaan kirjoitusjakson kesto on 70  $\mu$ s (kuva 27).



Kuva 27. Kirjoitus 1-Wire-linjaan, aaltomuoto ja pseudokoodiesitys.

Datan luku alkaa pakottamalla linja nolleen 6  $\mu$ s:ksi, minkä jälkeen se vapautetaan. Isäntä ottaa näytteen 9  $\mu$ s:n kuluttua linjan vapautuksesta. Kokonaisen lukujakson pi-tuus on 60, ja siihen lisätään 10  $\mu$ s:n odotus (kuva 28). [72; 73; 74.]



Kuva 28. 1-Wire-linjan lukeminen, aaltomuoto ja pseudokoodiesitys.

## Sarjaliikenne ja UART

Sarjaliikennettä on se, että lähetetään dataa bitti kerrallaan vastaanottimelle. Sarjaliikenteen ympärille on ajan myötä muodostunut useita protokollia ja niiden toimintaperiaate on sama. Kuitenkin fyysiset ominaisuudet, viestin pituus, jännitetasot ja ajoitukset vaihtelevat protokollien mukaan.

Sarjaliikenne jaetaan ajastuksen mukaan kahteen ryhmään, asynkroniseen, jossa ei käytetä erillistä linjaa kellolle, ja synkroniseen, jossa on erillinen linja kellolle. Asynkronisessa sarjaliikenneprotokolla on erittäin mukautuva eikä tarkasti määritelty, ja tärkeintä on aina, että kommunikoivien laitteiden asetukset vastaavat toisiaan. Kuitenkin protokollassa on sääntöjä ja mekanismeja, joilla virheetön tiedonsiirto varmistetaan. Niitä ovat nopeus, datan bittienmäärä, pariteettibitti ja synkronointibitit.

Sarjaliikenteessä kommunikaatio voi olla yksisuuntaista (engl. simplex), jolloin on yksi lähettävä ja yksi vastaanottava osapuoli, tai vuorosuuntaista (engl. half-duplex), jossa molemmat osapuolet voivat olla vuorollaan lähettäviä tai vastaanottavia. Suurin osa nykyaikaisista sarjaliikenneväylistä, kuten PC-sarjaliikenneportit, on kaksisuuntaisia (engl. full-duplex), eli ne ovat kykeneviä lähettämään ja vastaanottamaan dataa yhtäaikaaisesti.

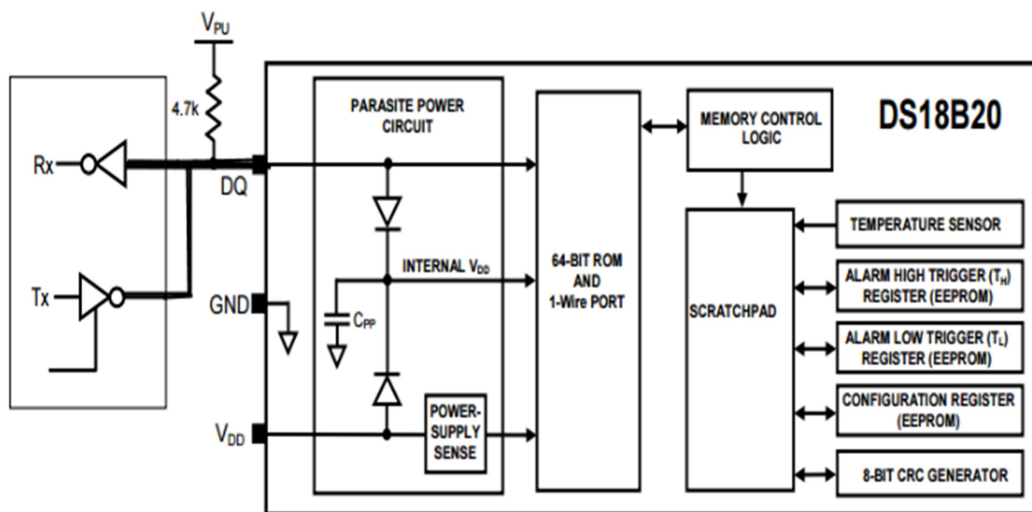
Bittinopeus määrittää, kuinka nopeasti data liikkuu sarjaväylässä. Yleisimpiä standardien käyttämiä arvoja ovat 1200, 2400, 4800, 9600, 19200, 38400, 57600, ja 115200 bps. Viestikehys sisältää 2–3 synkronointibittiä. Ne ovat start, joka kertoo, milloin datapaketti alkaa vetämällä linjan alas nolnaan, ja 1–2 stop-bittiä, jotka taas nostavat linjan lopuksi odotustilaan.

Kehys sisältää 5–9 bittiä dataa, ja se alkaa vähiten merkitsevällä bitillä. Datan jälkeen kehyksessä tulee pariteettibitti. Sen tarkoitus on toimia matalan tason virheetarkistuksessa. Määriteltävät tilat ovat pariton ja parillinen. Esimerkkinä tavu 01101000, jossa on kolme 1-bittiä. Tällöin pariteettitilassa parillinen olisi 1 ja tilassa pariton 0. Pariteetin asettaminen ei ole pakollinen, ja usein se jätetään pois kommunikaation nopeuttamiseksi, eikä siitä ole kummoista suojaa yhteyden ollessa häiriöinen. Tämän vuoksi siihen ei tuhlata bittejä, vaan käytetään enemmän CRC:ä (cyclic redundancy check), jolloin esimerkiksi pieniä kohinasta aiheutuneita virheitä ei tulkita ehjäksi dataksi. Mainittujen arvojen pitää olla toisiaan vastaavat molemmissa laitteissa, jotta virheetön kommunikointi on mahdollista.

UART (engl. universal asynchronous receiver/transmitter) on yleensä piiri, laite tai mikropiirin lohko, jolla saadaan hallittua sarjaliikennelinjat ja pakettien muodostaminen. Universaali viestii siitä, että dataformaatti ja baudinopeus ovat säädettäviä. Se vastaa asynkronisesta sarjaliikenteestä, eli siinä ei ole erillistä kellosignaalia. Käytännössä sen voi toteuttaa vähemmällä määrällä I/O-pinnejä ja kaapeleita. Tästä on hyötyä erityisesti sulautettujen järjestelmien kanssa, joissa käytettävissä on rajallinen määrä pinnejä. Tämä kuitenkin johtaa siihen, että kommunikoinnista tulee hiukan hankalampaa, sillä virheellinen nopeus aiheuttaa usein datahäviöitä tai vääristyneitä viestejä. [75; 76; 77.]

### 3.4 DS18B20-lämpöanturi

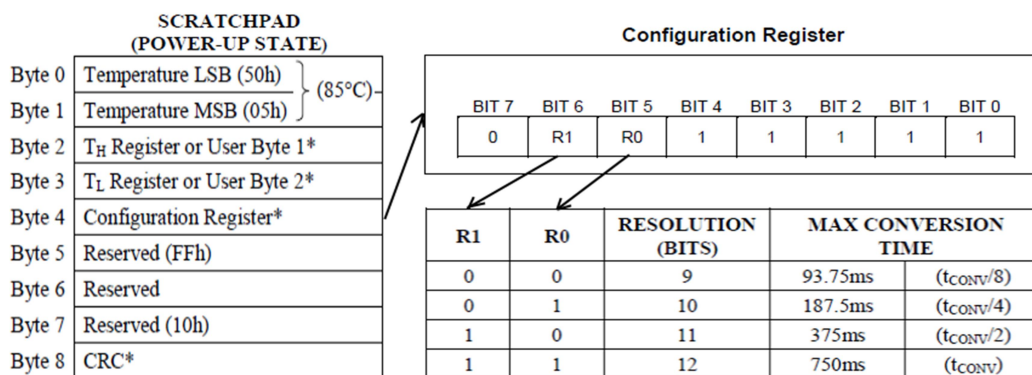
DS18B20 on digitaalinen lämpöanturi, joka tuottaa lämpötilan suoraan celsiusasteisena 9–12-bittisellä resoluutiolla. Se tarjoaa myös hälytystoiminnon käyttäjän määrittelemille ylä- ja alarajoille. Tämä 1-Wire-anturi tarvitsee ainoastaan yhden linjan kommunikaatiota varten (kuva 29), ja sen on mahdollista ottaa virta datalinjasta. Toimintajännite on 3,0–5,5 voltia. Anturi mittaa lämpötila-alueella  $-55$ – $125$  °C. Toleranssi on 0,5 astetta molempiin suuntiin alueella  $-10$ – $85$  °C. Lämpötilan muunto 12-bittiseksi sanaksi kestää piirissä enintään 750 millisekuntia.



Kuva 29. DS18B20-lohkokaavio ja yksinkertainen kytkentä isäntälaitteeseen.

Kun isäntälaitte on lähettänyt alustuspulssin ja laitteen tunnistus on suoritettu, anturille välitetään muuntokomento, jolloin anturi mittaa lämpötilan ja tekee A/D-muunnoksen. Lopuksi se välittää saadun arvon 2-tavuiseen rekisteriin ja palaa odotustilaan. Jos lämpötilamittaus on kesken, anturi lähettää arvoa 0, ja kun muunnos on valmis, arvo on vastaavasti 1.

DS18B20-anturin muisti koostuu SRAM-välimuistista, jossa on pysyvä EEPROM-varasto (kuva 30). Se antaa suoraan celsiusarvon 9–12-bittisellä tarkkuudella, sen mukaan, mitä on määritelty piirinmuistin tavussa 4 sijaitsevaan asetusrekisteriin (kuva 30). Lämpötilalle on varattu muistista tavut 0 ja 1. EEPROM:ssa on varattu 2 tavua hälytystoiminnon ylä- ja alarajoille. Tavut 5–7 on varattu, ja niitä ei voi ylikirjoittaa. Muistin tavu 8 sisältää CRC-koodin, ja tämä muistin osio on vain luettavissa. Kun virta kytetään, EEPROM:n rekisterit tallentuvat, ja käynnistyksen yhteydessä niiden data ladataan oikeisiin kohtiin välimuistissa. Isäntä pystyy myös laitekohtaisella komennolla pakottamaan tiedon uudelleen latauksen EEPROM:sta välimuistiin.



Kuva 30. DS18B20-välimuistin rakenne ja asetusrekisteri.

Kommunikointi DS18B20-anturin kanssa alkaa isäntälaitteelta tulevalta alustuspulssilla. Tähän sensori vastaa lähettämällä lyhyen pulssin ilmoituksena paikallaolosta. Saatuaan orjalaitteilta vastauksen isäntälaitte voi alkaa lähettää ROM-komentoja. Nämä komennot on mahdollista rajata koskemaan tiettyä laitetta lähettämällä sen 64-bittinen osoite. Esimerkiksi 55h ja perään osoite rajaisi kaikki muut laitteet pois kommunikoinnista siihen asti, kunnes isäntä antaa alustuspulssin.



Isäntälaitteen antamaa ROM-komentoa seuraa funktiokomento. Näiden komentojen avulla isäntälaitte voi lukea ja kirjoittaa anturin välimuistiin, antaa lämpötilanhakukäskyn tai määrittää virransyöttötavan. DS18B20-anturin kommunikointi seuraa aina kaavaa alustus tai tunnistus, ROM-komento ja viimeisenä funktiokomento. Tästä poikkeamisesta seuraa yleensä virheellistä dataa tai datan yhteentörmäyksiä. [68; 74.]

### 3.5 SIM900-moduuli

SIM900 on monipuolinen GSM/GPRS-moduuli. Se tukee neljää taajuusalueita, GSM850, EGSM900, DCS1800 ja PCS1900. Moduulissa on nopea 156 MHz:n kello- taajuudella toimiva AMR926EJ-S-prosessori. Teollisen standardin mukaisen väylän ja vähäisen virrankulutuksen ansiosta se sopii erittäin hyvin sulautettuihin järjestelmiin. Virran kulutus lepotilassa jää 1,5 mA:iin ja normaali käyttöjännite on 3,3–5 voltia. Lähetysvaiheessa SIM900:n virrankulutus nousee hetkellisesti jopa noin 2000 mA:iin, ja jatkuva kulutus aktiivisella moduulilla on 500 mA. Esimerkiksi lähetettäessä 16 data-alkiota virrankulutus on noin 2000mAs (taulukko 2).

Taulukko 2. SIM900-moduulin tehonkulutusarvojen kasvaminen pakettien koon kasvaessa [78].

| Data items | Packet size (bytes) | Power consumption (mAs) |
|------------|---------------------|-------------------------|
| 16         | 287                 | 1987                    |
| 32         | 511                 | 2180                    |
| 64         | 1981                | 2590                    |
| 128        | 4157                | 3270                    |
| 256        | 8509                | 3570                    |

SIM900 voi pitää yllä vain yhtä palvelua, eli GPRS- ja GSM-yhteys eivät toimi rinnakkain. Latausnopeus 80 kbps ja lähetysnopeus jää 20 kbps:iin. Tämä on kuitenkin riittävä yksinkertaisen datan siirtoa varten. Kokoa moduulilla on 24 mm x 24 mm x 3 mm ja painoa alle 4 grammaa, eli se on helppo saada sopimaan pieniinkin koteloihin. SIM900-

moduulin ohjataan sarjaportin kautta AT-komennoilla (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands).

GSM-standardia seurannut GPRS (engl. General Packet Radio Service) on pakettikytkentäinen tiedonsiirtopalvelu. Siitä käytetään myös usein nimitystä 2.5G. Se laajentaa GSM-standardin ja mahdollistaa teoriassa tiedonsiirron jopa nopeudella 172,2 kbit/s, kuitenkin käytännössä nopeudet jäävät noin 40 kbit:iin/s. Audion siirrossa standardi hyödyntää GSM-verkon arkkitehtuuria ja mahdollistaa internetyhteyden käyttäen IP- tai x.25-protokollaa. GPRS toi tullessaan uusina ominaisuuksina mahdollisuuden toimia asiakas-palvelin-tilassa IP-verkoissa. Se toi myös lähetystekniikat, PTP (engl. Point to point) ja PTMP (engl. Point to multipoint), ja tekstiviestin lähettämisen pakettina SMS (engl. Short message services). GPRS-tekniikassa on määritelty neljä liikenteen nopeuteen vaikuttavaa kanavankoodausluokkaa (taulukko 3).

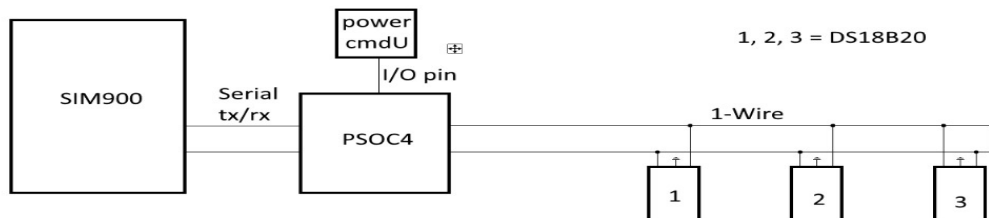
Taulukko 3. GPRS-koodauksen vaikutus verkon nopeuteen [79].

| Encoding scheme | Throughput  | Protection          |
|-----------------|-------------|---------------------|
| CS-1            | 9.05 Kbit/s | Normal (signalling) |
| CS-2            | 13.4 Kbit/s | Slightly lower      |
| CS-3            | 15.6 Kbit/s | Reduced             |
| CS-4            | 21.4 Kbit/s | No error correction |

SIM900-piiri hyödyntää 1970-luvulla alkunsa saaneita modeeminohjauskomentoja. Ne tunnetaan AT-komentoina, jonka juuri juontuu sanasta Attention, ja se kuvaa hyvin itse käytettävää komentorakennetta, joka on etuliite AT ja komento. AT antaa laitteelle tiedon komentolinjan avaamisesta, ja sitä seuraa suoritettava komentosarja. Esimerkiksi verkkoon rekisteröityminen tapahtuu komennolla AT+CREG. [78; 79; 80; 81.]

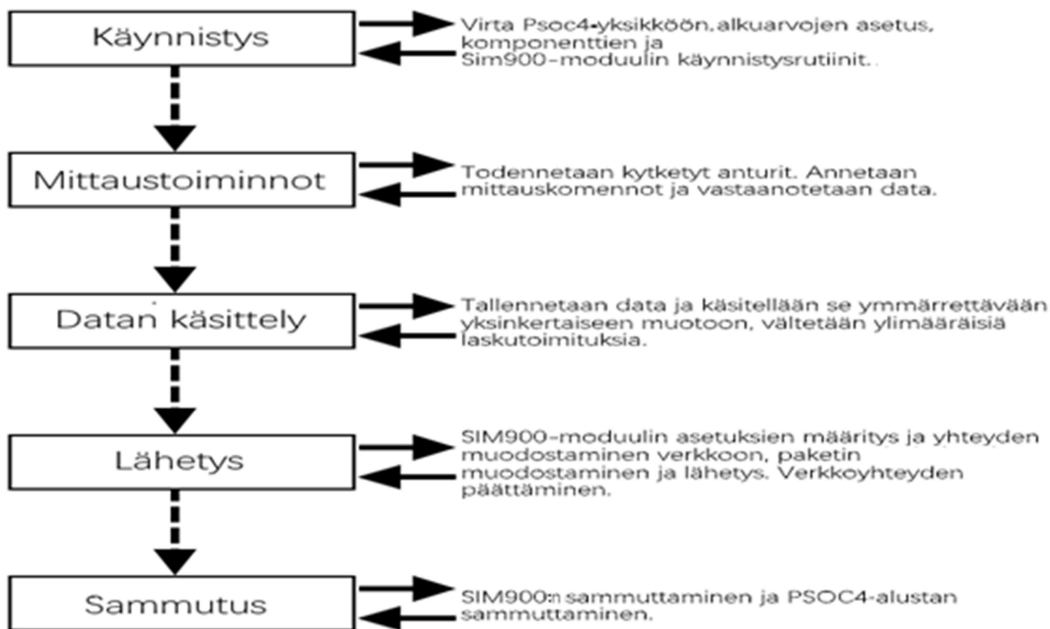
### 3.6 Laitteistokaavio ja toimintamalli

Kun insinööriyön laitteisto oli valittu ja tutkittu alustavasti ominaisuudet, seuraavana vaiheena oli tehdä kokoaminen ja suunnitelma fyysisestä rakenteesta (kuva 31). SIM900 vastaa lähetyspuolesta, PSOC4 komentaa sitä ja vastaa 1-Wire-isäntänä datan käsittelystä.



Kuva 31. Prototyypissä käytettävän laitteiston alustava kytkentä.

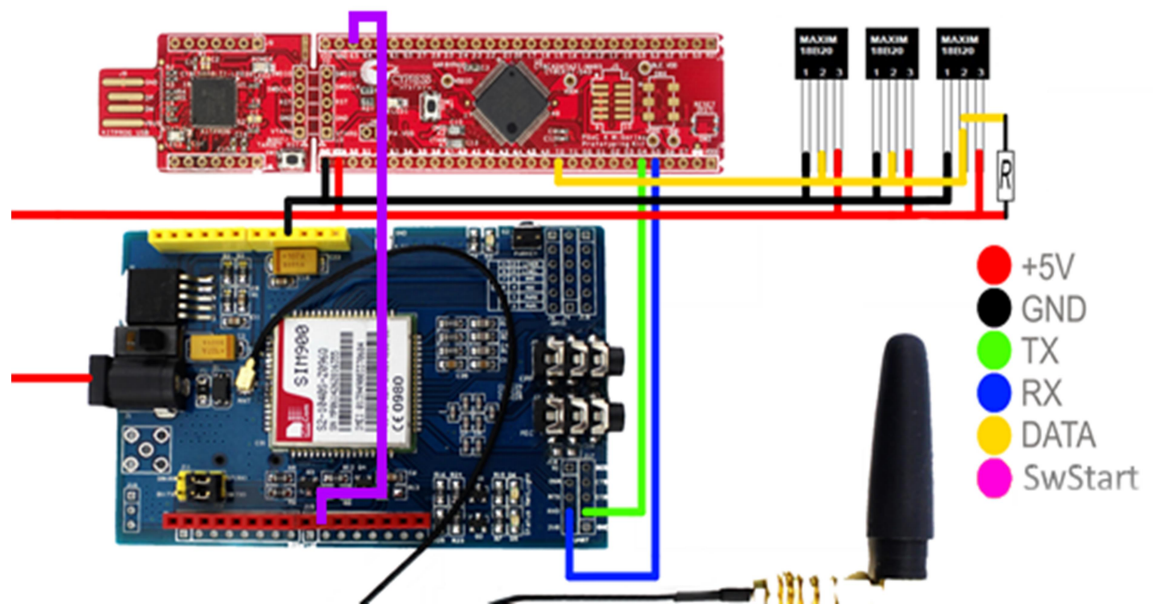
Seuraavaksi loin kuvauksen toimintakierrosta, jonka pohjalta itse ohjelmiston toteuttaminen lähtee. Kierro lähtee PSOC4-alustan käynnistyksestä ja päättyi laitteiden hallittuun sulkemiseen (kuva 32).



Kuva 32. Laite ja ohjelman toiminnot kuvattuna tapahtumien virtana.

#### 4 Prototyypin laitteisto- ja ohjelmistototeutus

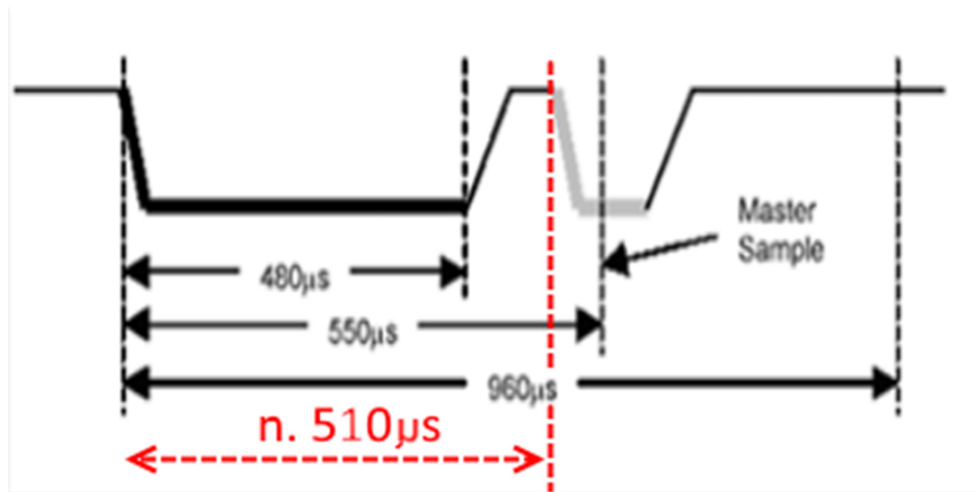
Prototyypin viimeistelyyn toteutukseen valitun laitteiston kokoaminen alkoi kytkemällä DS18B20-anturit ja lisäämällä 1-Wire-dataväylään 4,7 kilo-ohmin ylösvetovastus. Seuraavaksi määritettiin pinnit kahdelle PSoc4-UART-komponentille. Näistä kaksi hoiti kommunikoinnin SIM900-moduulin kanssa ja toinen pari oli varattu väylälle, jonka kautta muodostettiin yhteys tietokoneeseen testien tarkkailua varten. Viimeisenä juotettiin SIM900-moduulin pinnille silta ohjelmistokäynnistystä varten. Tämä kytki sen D9-pinniin ja tätä kautta PSoc4-alusta hoitaa moduulin käynnistämisen (kuva 33).



Kuva 33. PSoC4200-kehitysalusta ja laitteisto kytkettynä toisiinsa.

Ensimmäisenä vaiheena ohjelmiston osalta lähdin toteuttamaan puuttuvaa 1-Wire-komponenttia ja siihen vaadittavia kirjastoja PSoC4-alustalle. Dokumentaatioiden perusteella (ks. kuvat 26–29) loin kolme toimintoa, alustus, bitin kirjoitus ja lukeminen. Seuraavaksi ennen pseudokoodin kääntämistä alustalle kirjoitin yksinkertaisen testiympäristön, jossa pystyin kommunikoimaan sarjaportin kautta ja saamaan sensorien lähettämät viestit tietokoneeni näytölle analysoitavaksi mikroprosessorilta hyödyntäen PuTTY-pääte-emulaattoriohjelmistoa. Tähän PSoC4200-kehitysalustasta löytyi USB-UART-komponentti ja valmiit kirjastot.

Kun USB-UART:n toiminta oli varmistettu ja datan tulostus ruudulle onnistui, siirryttiin kääntämään aiemmin tehtyä pseudokoodia C-kielelle ja implementoimaan sitä alustaan. Toteutin 1-Wire-toiminnallisuuden kolmena funktiona, alustus, tavunlukeminen ja tavunkirjoitus. Alustusfunktion ensimmäiset testit epäonnistuivat, koska kiinni olleelta 1-Wire-orjalaitteelta ei saatu vastausta. Logiikka-analysaattorilla kuitenkin pian selvisi virhe ajastuksessa. Laite vastasi, mutta kuten esimerkkikuvassakin näkyy (kuva 34), osui näytteenotto juuri huipulle.



Kuva 34. Aikakaavio, jossa punainen kuvaa virheellistä näytteenottohetkeä.

Virhe korjautui osittain siirtämällä näytteenottohetkeä. Kuitenkin seuraavaksi ilmeni häiriöitä sarjaliikenteessä, ja kun oli siirrytty toiseen PSoc4200-alustaan, ajastukset jouduttiin palauttamaan alkuperäisiin. Ajastuksen virhe yksilöityi laiteviaksi, ja tämän jälkeen siirryin seuraavaan funktioon.

Itse datan luku ja kirjoitus toteutettiin Bit banging -tekniikalla. Tässä tekniikassa muutetaan tai kuunnellaan pinnin tilaa. Pinnille annetaan arvo 1 tai 0 määritellyksi ajanjaksoksi ja varmistetaan, että se osuu näytteenottohetkellä haluttuun arvoon.

Seuraavaksi kytkettiin anturit väylään ja kerättiin tunnistenumerosarjat talteen sivutoeutuksena tehdyllä 1-Wire-ID-lukijalla. Sitten toteutettiin funktio, joka suorittaa mittauskomennot ja datankeruun antureista. Molemmat, mittaus ja datan luku, tehdään antamalla ensin alustuskomento, minkä jälkeen yhdistä-komento ja halutun anturin ID. Näitä seuraa toimintoa vastaava komento, joko mittaa (0x44) ja odotusajanjakso tai lue

data tilapäismuistista (0xBE). Tämä rutiini toistettiin jokaisen kytketyn anturin osalta (taulukko 4), ja tiedot otettiin talteen lähetystä varten.

Taulukko 4. Kommunikaation kulku keskusteltaessa Rom-komennoilla DS18B20:n kanssa.

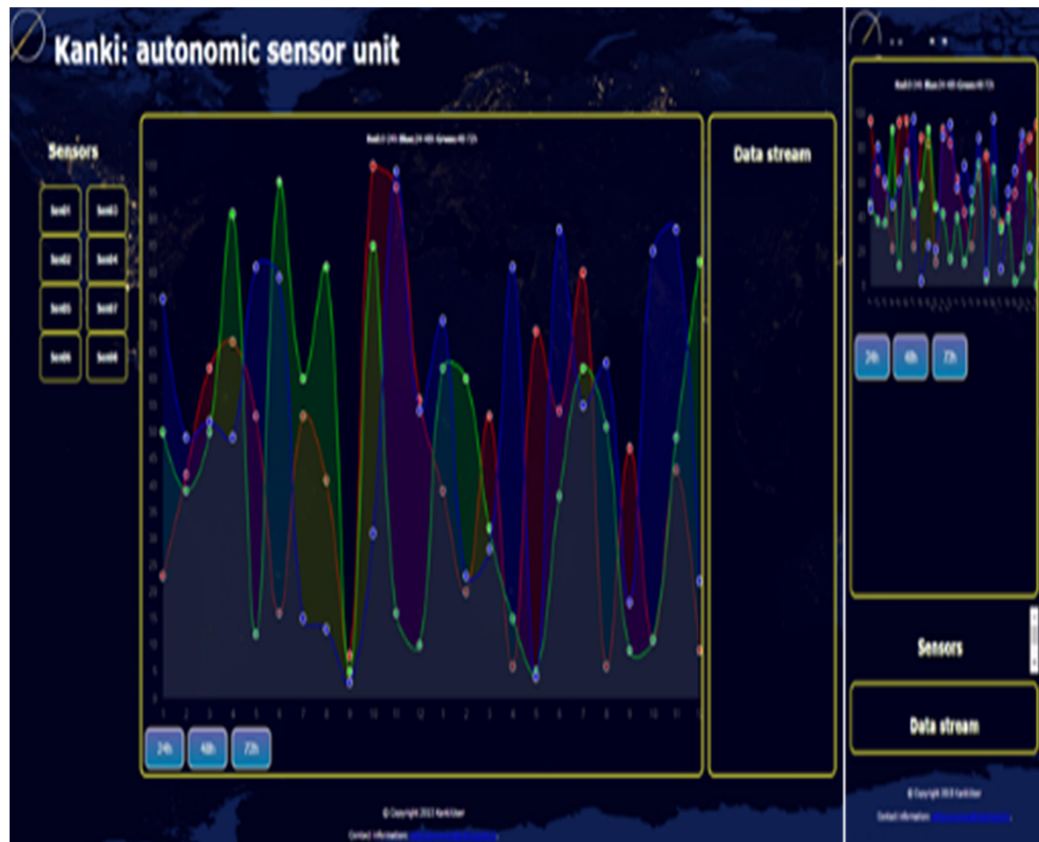
|               |                       |
|---------------|-----------------------|
| <b>Master</b> | <b>Reset</b>          |
| <b>Slave</b>  | <b>Presence</b>       |
| <b>Master</b> | <b>55 h</b>           |
| <b>Master</b> | <b>64-Bit ID</b>      |
| <b>Master</b> | <b>44 h</b>           |
| <b>Master</b> | <b>tx high (n) ms</b> |
| <b>Master</b> | <b>Reset</b>          |
| <b>Slave</b>  | <b>Presence</b>       |
| <b>Master</b> | <b>55 h</b>           |
| <b>Master</b> | <b>64-Bit ID</b>      |
| <b>Master</b> | <b>BE h</b>           |
| <b>Slave</b>  | <b>9 data bytes</b>   |

Viimeisenä isona kokonaisuutena toteutettiin yhteys SIM900-moduuliin ja sen kautta datan välitys palvelimelle. Moduulin kanssa kommunikointi fyysisen sarjaportin kautta, koska käytössä ollut alusta ei tukenut Arduinolle tarkoitettua ohjelmistosarjaporttia.

Ensimmäisenä tavoiteltiin OK-vastausta AT-komentoon, mikä onnistuikin moitteettomasti. Tästä jatkettiin kirjoittamalla AT-komennoista runko TCP-yhteyttä varten ja muotoilemalla ne suoraan UART-yhteensopiviksi merkkijonoiksi. Palvelimella oli tässä vaiheessa vain lokitiedosto, joka kirjasi sinne lähetetyn datan. Tässä vaiheessa tuli esille virtalähdeongelma, mikä näkyi SIM900-moduulin sammumisena aina datanlähetysvaiheessa. Virtalähteen vaihdon jälkeen saatiin ensimmäiset viestit tiedostoon. Tämän jälkeen moduulista lähetettävä datajono muotoiltiin POST-metodia noudattelevaksi. Lopuksi muodostin SIM900-moduulin käynnistysfunktion, jolla se saatiin käynnistettyä ilman virtanappia tai erillistä ohjattavan kytkimen lisäämistä. Sammutusta varten moduuli tarvitsi vain yhden AT-komennon rutiinin loppuun.

Tietokantayhteys ja verkkosivun toteutus

Datan esittämistä varten toteutin yksinkertaisen sivuston, jota lähdin rakentamaan informaatiiovetoisesti ja tavoitteena selkeä esitystyylillä. Erilaisia toimintoja ei sivu itsestään vaatinut, mutta halusin toteuttaa sen mukautuvana ja tällöin sen käytettävyys olisi siedettävä niin matkapuhelimella kuin myös tietokoneen ruudulta. Tämän toteuttamiseen käytin Flexbox-asettelupohjaa (kuva 35), joka asemoi ja venyttää elementtejä näytön mukaan.



Kuva 35. Vasemmalla kuvankaappaus täydestä näyttöruudusta ja oikealla vertailukohtaksi pienennetty ikkuna, jossa elementit Data stream ja Sensors ovat siirtyneet automaattisesti datanesitysrudun alle.

Koska verkkosivujen toteutus ja CCS oli yksi osa-alueista, jotka eivät olleet ennestään hallussa, valitsin valmiin Chart.js-kirjaston anturidatan ilmaisua varten. Kirjastossa oli useita valmiita malleja, joista valitsin linjakaavion. Tällöin tarvitsi ainoastaan määritellä käytettävä data ja ulkoisia ominaisuuksia, kuten väri ja koko (kuva 36).

```

label: "My First dataset",
fillColor : "rgba(200,0,0,0.2)",
strokeColor : "rgba(200,0,0,1)",
pointColor : "rgba(200,0,0,1)",
pointStrokeColor : "#fff",
pointHighlightFill : "#fff",
pointHighlightStroke : "rgba(200,0,0,1)",
data : dataSet[1]

```

Kuva 36. Graafisen diagrammin määrittelyt, koko annetaan luontivaiheessa.

Graafisen toteutuksen jälkeen luotiin MySQL-tietokanta ja lähdettiin toteuttamaan tietokantayhteyttä. Itse tietokanta koostui kahdesta taulusta, joista ensimmäiseen tuli laite-tunniste ja aika. Toiseen tallennettiin kirjaustunniste yhdistämistä varten sekä data ja anturin numero (kuva 37).

| idKIRJAUS | sensori_id | pvm                 | idKIRJAUS_DATA | kirjaus_id | arvo | nro |
|-----------|------------|---------------------|----------------|------------|------|-----|
| 5         | 1          | 2015-10-18 17:07:28 | 3              | 5          | 34   | 1   |
| 6         | 1          | 2016-11-05 00:00:00 | 4              | 5          | 22   | 2   |
| 7         | 1          | 2016-11-05 01:00:00 | 5              | 5          | 34   | 3   |
| 8         | 1          | 2016-11-05 02:00:00 | 6              | 6          | 21   | 1   |

Kuva 37. Datan tallennukseen käytettyjen taulujen rakenteet.

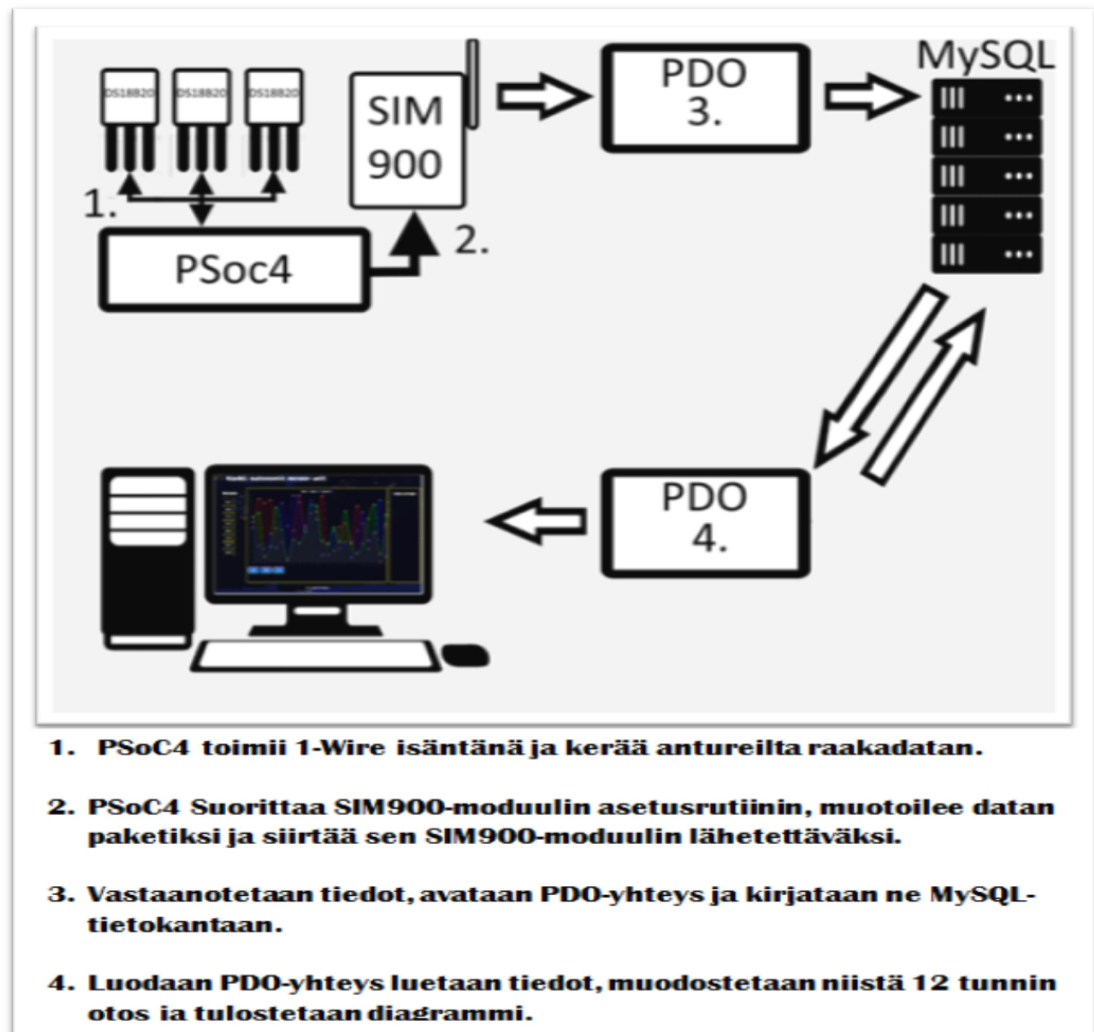
Datan kirjoituksessa ja datan lukemisessa tietokantoihin hyödynnettiin PDO-rajapintaa, joka on PHP 5.1 -versiosta lähtien mukana ollut laajennus. Kun PDO-yhteys oli muodostettu (kuva 38), valmisteltiin kirjauslause, suoritettiin se ja otettiin ID talteen tulevaa datankirjausta varten. Viimeisenä vaiheena oli valmisteella jokaiselle erotellulle datalle syötteet (kirjaus-ID, anturin lukema, anturin järjestysnumero).

```
$db = new PDO("mysql:host=mysql.metropolia.fi;dbname=generic;charset=utf8","user","wordpass");
```

Kuva 38. PDO-yhteyden muodostavan lauseen rakenne.



Lukeminen toteutettiin ottamalla viimeiseltä kahdeksatoista tunnilta arvot jokaiselle sensorille ja muodostamalla niistä omat taulunsa. Valmiit taulukot toimitettiin syötteiksi LineDataChart-funktiolle, jonka tehtävä on muodostaa niistä graafinen esitys annettujen määreiden mukaan (kuva 39).



Kuva 39. Valmiin laitteiston toimintasykleistä selitteinen.

Laitteiston yhdistämisen jälkeen se testattiin vaiheittain (kuva 39). Lopuksi suoritettiin koko toimintasyklin kattavat testit, joilla varmistettiin osakokonaisuuksien yhteentoimivuus. Kummassakaan testeistä ei havaittu virheitä toimivuuden osalta.

## 5 Yhteenveto ja jatkokehitys

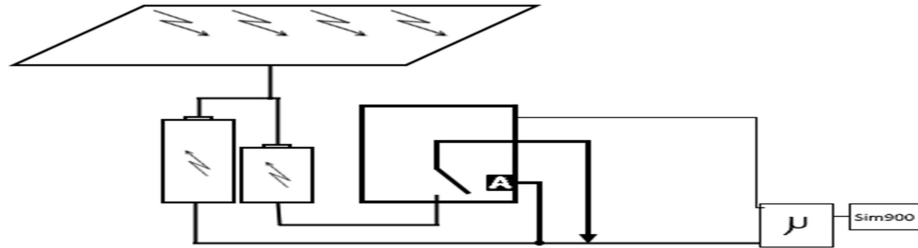
Insinööriyön ensimmäisessä vaiheessa perehdyin useaan protokollaan, ja niiden suuri määrä yllätti, vaikka suurinta osaa tutkimistani protokollista ei hyödynnetty käytännön prototyypin valmistamisessa. Tämä perehtymistyö selvensi IoT-kehityksen monimuotoisuutta ja sitä, kuinka haastava IoT on myös terminä ymmärtää.

Anturijärjestelmän prototyyppiin sain idean opinnäytetyötäni aluksi ohjanneen Juho Vesasen Antin kasvimaan -kokonaisuudesta, jolla seurattiin lämpötilaa ja välitettiin se radiolinkin kautta palvelimelle. Omassa työssäni lähdin rakentamaan pohjaa autonomiselle järjestelmälle, joka pystyy välittämään anturien keräämän datan suoraan pilveen. Järjestelmän alustana toimi PSoC4-mikrokontrolleri, johon liitettiin SIM900 GPRS -moduuli ja kolme 1-Wire-väylään kytkettyä lämpötila-anturia. Laitteiston ohjelmisto luotiin PSoC Creator -työkalulla käyttäen sen tukemaa C-kieltä ja joitakin valmiita kirjastoja. Verkkosivut tehtiin HTML5-pohjaisilla tekniikoilla, ja datan tallennus toteutettiin MySQL-tietokantapalvelinympäristössä.

Työn aikataulutuksen, määrittelyn ja kiireen vuoksi monia ominaisuuksia, jotka olivat mielessä aluksi, jouduttiin karsimaan. Kuitenkin ajatuksena oli prototyyppi, jonka pohjalta voisi saada aikaan käyttökelpoisen tuotteistettavan laitteen. Tässä onnistuin ja sain implementoitua halutut ominaisuudet toimivina valmistuneeseen prototyyppiin. Seuraavana kehitysvaiheena olisi laitteen vapaampi laajennettavuus, eli useampi väylä, joihin saisi suoraan liitettyä antureita ilman ohjelmiston muuttamista. Käytännössä toteutuksessa päälaite hakisi ensimmäisellä käynnistyksellä asetustiedoston, jossa hyväksytyjen antureiden tarvittavat ID-tiedot olisivat. Tämä tapahtuisi joko erilliseltä muistikortilta tai verkkoyhteyden kautta tietokannasta. Tässä ongelmaksi saattaisi muodostua suuri virrankulutus, eritoten jos käytössä olisi nykyinen laitteisto ja anturitunnisteiden haku vaatisi yhdistämisen verkkoon.

Virransyötössä seuraavana olisi tarkoitus laajentaa virtalähdettä toimimaan autonomisesti ja vastaamaan laitteen herättämisestä ja tarvittavista hälytyksistä. Virtalähdeyksikkö sisältäisi pääakun ja vara-akun (kuva 40), jotka latautuisivat paneelien kautta. Virtalähteen hallintayksikkö käyttäisi pienempää vara-akkua ja seuraisi isompikapasiteettisen tilaa. Isomman kapasiteetin laskiessa liian alhaiseksi se koko laitteisto siirtyisi käyttämään varavirtaa. Tämän jälkeen virtalähdeyksikkö lähettäisi ilmoituksen GPRS-yksikön kautta huoltonumeroon tai valvontasivustolle. Tällöin olisi mahdollista toimia,

ennen kuin yksikkö menettäisi kaiken toiminnallisuutensa. Tässä toteutuksessa tulisi kuitenkin haasteeksi etenkin Suomen oloissa kolme kuukautta kestävä pimeä jakso, jolloin jouduttaisiin turvautumaan pelkkään akkuvirtaan.



Kuva 40. Prototyypin virtalähdeyksikkö konsepti.

Kehitysideoiden vuoksi yhtenä haasteena prototyypissä oli sen rajaaminen kokonaisuutena. Eritoten työn alkuvaiheessa ideoita erinäisistä lisäyksistä syntyi paljon ja epäoleellisten karsiminen oli tärkeää aikataulun vuoksi. Esimerkkinä karsituista ideoista olivat etähuoltotila ja etäohjaus, jotka osoittautuivat loppujen lopuksi hyödyttömiksi, sillä laitteen vikaantuessa se vaatisi tarkistuksen.

Työn monimuotoisuuden vuoksi tutustuin todella laajasti, mutta pintapuolisesti, useisiin aihealueisiin, lakiteksteistä tarkoin määriteltyihin datalehtiin. Tämä antoi myös uutta näkökulmaa tulevaisuuden suuntauksiin teknologia alalla, tietoa IoT:sta liiketoimintana ja näkemystä sen kaupallisten mahdollisuuksien rajattomuudesta.

Erilaisten väylien kanssa työskentely ja eritoten 1-Wire-komponentin puute PSoC4-alustassa edesauttoi oman osaamiseni kehitystä. Vaikka 1-Wire on yksinkertainen, sen toiminnallisuuden ohjelmointi kokonaisuudessaan toi muutamia haasteita. Niistä mainittavia olivat virhetilanteen aiheuttama häiriö ajoituksessa ja alustassa käytetyn virtalähteen viallisuus. Ne toivat hyvin esille mittalaitteiston ja logiikka-analysaattorin tärkeyden vianselvityksessä.

Lopputuloksena syntyneeseen laitteeseen olin jokseenkin tyytyväinen, vaikkakin sen valmistumisen jälkeen ilmeni useita mahdollisia lisäyksiä ja kehitysehdotuksia. Erityisesti kiinnostavana aiheena oli verkkosivun toteutus ja tietokantayhteyden toteutus, sillä siitä ei aikaisempaa kokemusta ole. Tähän opettelin uusia tapoja tehdä asioita, kuten flexbox ja PDO-tietokantayhteys.

## Lähteet

1. Wellers, Daniel. 2015. Is this the future of the Internet of Things? Verkkodokumentti. World Economic Forum <<https://www.weforum.org/agenda/2015/11/is-this-future-of-the-internet-of-things/>> Luettu 3.6.2016.

2. Butler project overview. 2011. Verkkodokumentti. Butler. <<http://www.slideshare.net/butler-iot/butler-project-overview-13603599>> Luettu 3.6.2016.

3. Zalman, Tara. 2015. Internet of Things Protocols and Standards Verkkodokumentti. <[http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot\\_prot/](http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/)> Luettu 3.6.2016.

4. 802.11ah Wi-Fi (900 MHz) to Provide Low Power, Long Range Connectivity for the Internet of Things. 2014. Verkkodokumentti. CNXSoft. <<http://www.cnx-software.com/2014/02/21/802-11ah-wi-fi-900-mhz-to-provide-low-power-long-range-connectivity-for-the-internet-of-things/>> Luettu 3.6.2016.

5. Song, Jianping; Han, Song; Mok, Ai; Chen, Deji; Lucas, Mike; Nixon, Mark; Pratt, Wally. WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. Verkkodokumentti. <[http://www.cs.odu.edu/~nadeem/classes/cs795-CPS-S13/material/sense-005\\_slides.pdf](http://www.cs.odu.edu/~nadeem/classes/cs795-CPS-S13/material/sense-005_slides.pdf)> Luettu 4.6.2016.

6. WirelessHART - How it works. Verkkodokumentti. Fieldcomm Group. <[http://en.hartcomm.org/hcp/tech/wihart/wireless\\_how\\_it\\_works.html](http://en.hartcomm.org/hcp/tech/wihart/wireless_how_it_works.html)> Luettu 5.6.2016.

7. Z-Wave Protocol Overview. 2006. Verkkodokumentti. <[http://wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243\\_2\\_Z\\_Wave\\_Protocol\\_Overview.pdf](http://wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243_2_Z_Wave_Protocol_Overview.pdf)> Luettu 5.6.2016.

8. Fouladi, Behrang & Shand, Ghanoun. 2013. Hacking Z-Wave Home Automation Systems. Verkkodokumentti. <<http://www.slideshare.net/sensepost/hacking-zwave-home-automation-systems>> Luettu 5.6.2016.

9. Bluetooth Technology Basics, Bluetooth low energy. 2016. Verkkodokumentti. Bluetooth SIG. <<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>> Luettu 5.6.2016.

10. BLE Stack. 2013 Verkkodokumentti. Device Factory. <<http://devicefactory.com/ble-stack/>> Luettu 5.6.2016.

11. ZigBee Smart Energy. 2016. Verkkodokumentti. ZigBee Alliance. <<http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbeesmartenergy/>> Luettu 6.6.2016.

12. Smart Energy Profile 2 (SEP 2)– IP based Energy Management for the Home. 2013. Verkkodokumentti. ZigBee Alliance. <<http://www.zigbee.org/?wpdmdl=2148>> Luettu 6.6.2016.
13. Bandara, Dilum. 2015. Other types of networks: Bluetooth, Zigbee, & NFC. Verkkodokumentti. <<http://www.slideshare.net/DilumBandara/other-types-of-networks-2015-3>> Luettu 6.6.2016.
14. DASH7 Alliance Wireless Sensor and Actuator Network Protocol VERSION 1.0. 2015. Verkkodokumentti. DASH7 Alliance . <<http://www.dash7-alliance.org/?product=dash7-alliance-protocol-specification-v1-0>> Luettu 6.6.2016.
15. DASH7. Verkkodokumentti. All Bits Knowledge HUB. <<http://www.allbits.eu/internet-of-things/wireless-connectivity/dash7/>> Luettu 6.6.2016.
16. HomePlug Technology Overview. 2016. Verkkodokumentti. HomePlug Alliance. <<http://www.homeplug.org/tech-resources/techoverview/>> Luettu 8.6.2016.
17. Extend Wi-Fi with HomePlug. 2016. HomePlug Alliance. <<http://www.homeplug.org/explore-homeplug/hybrid/>> Luettu 8.6.2016.
18. ITU-T G.9959. 2015. Telecommunication standardization sector of ITU. Verkkodokumentti. International Telecommunication Union. <<https://www.itu.int/rec/T-REC-G.9959-201501-l/en/>> Luettu 8.6.2016.
19. Wannstrom, Jeannette. 2013. LTE Advanced. Verkkodokumentti. 3GPP. <<http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced>> Luettu 10.6.2016.
20. Agaur. 2012. Introduction of Relay Nodes in LTE-Advanced. Verkkodokumentti. Lte World <<http://lteworld.org/blog/introduction-relay-nodes-lte-advanced>> Luettu 10.6.2016.
21. Poole, Ian. LTE CA: Carrier Aggregation Tutorial. Verkkodokumentti. Adrio Communications. <<http://www.radio-electronics.com/info/cellulartelecomms/lte-long-term-evolution/4g-lte-advanced-carrier-channel-aggregation.php>> Luettu 10.6.2016.
22. Delivering on the LTE Advanced promise. 2016. Verkkodokumentti. Qualcomm Technologies. <<https://www.qualcomm.com/documents/delivering-on-the-lte-advanced-promise>> Luettu 10.6.2016.
23. A technical overview of LoRa and LoRaWAN. 2015. Verkkodokumentti. LoRa Alliance. <<https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRaWAN101.pdf>> Luettu 10.6.2016.
24. LoRaWAN Overview. 2016. Verkkodokumentti. The Things Network. <<https://www.thethingsnetwork.org/wiki/LoRaWAN/Overview>> Luettu 11.6.2016.

25. Symphony Link Protocol. 2015. Verkkodokumentti. Link Labs. <[http://info.link-labs.com/hubfs/Symphony\\_Protocol\\_Overview-2.pdf](http://info.link-labs.com/hubfs/Symphony_Protocol_Overview-2.pdf)> Luettu 20.6.2016.
26. Symphony Link vs. LoRaWAN. Verkkodokumentti. Link Labs. <[http://info.link-labs.com/hubfs/LPWAN\\_Technology\\_Explained.pdf](http://info.link-labs.com/hubfs/LPWAN_Technology_Explained.pdf)> Luettu 20.6.2016.
27. What Is Weightless. 2015. Link Labs. <<http://www.link-labs.com/what-is-weightless/>> Luettu 20.6.2016.
28. Weightless-N. Verkkodokumentti. Weightless SIG. <<http://www.weightless.org/about/weightlessn>> Luettu 20.6.2016.
29. Weightless-P. Verkkodokumentti. Weightless SIG. <<http://www.weightless.org/about/weightlessp>> Luettu 20.6.2016.
30. Weightless-W. Verkkodokumentti. Weightless SIG. <<http://www.weightless.org/about/weightlessw>> Luettu 20.6.2016.
31. Buckiewicz, Bob. Technical Overview of DECT ULE. Verkkodokumentti. LSR. <<https://www.lsr.com/white-papers/technical-overview-of-dect-ule>> Luettu 1.7.2016.
32. SIGFOX: LPWAN made for IoT. 2016. Verkkodokumentti. Thinxtra. <<http://www.thinxtra.com/2016/01/sigfox/>> Luettu 1.7.2016.
33. Poole, Ian. SIGFOX for M2M & IoT. Verkkodokumentti. Adrio Communications. <<http://www.radio-electronics.com/info/wireless/sigfox/basics-tutorial.php>> Luettu 1.7.2016.
34. Poller, Bert. 2013. Connecting Things to the Internet with SIGFOX. Verkkodokumentti. Ekito. <<https://www.ekito.fr/people/connecting-things-to-the-internet-with-sigfox/>> Luettu 2.7.2016.
35. IPv6 related standards and effort page – RPL. Verkkodokumentti. IoT6. <<http://iot6.eu/rpl>> Luettu 2.7.2016.
36. 11 Internet of Things (IoT) Protocols You Need to Know About. 2015. Verkkodokumentti. RS Components. <<https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about>> Luettu 2.7.2016.
37. Technology, Battery friendly design. Verkkodokumentti. Thread Group. <<http://threadgroup.org/technology/ourtechnology#technology>> Luettu 2.7.2016.
38. Aijaz, Adnan; Su, Hongjia; Aghvami, Abdol-Hamid. CORPL: A Routing Protocol for Cognitive Radio Enabled AMI Networks. Verkkodokumentti. <[https://pdfs.semanticscholar.org/bad0/0cdca01cf74c06d49208ef9cf5423e2e4df1.pdf?\\_ga=1.225218138.1946126132.1476540303](https://pdfs.semanticscholar.org/bad0/0cdca01cf74c06d49208ef9cf5423e2e4df1.pdf?_ga=1.225218138.1946126132.1476540303)> Luettu 3.7.2016.

39. Rhodes, Tom & Jude, Allan. Common Address Redundancy Protocol (CARP). Verkkodokumentti. FreeBSD. <<https://www.freebsd.org/doc/handbook/carp.html>> Luettu 3.7..2016.
40. Caposelle, Angelo; De Cicco, Gianluca; Petrioli, Chiara. 2015. R-CARP: A Reputation Based Channel Aware Routing Protocol for Underwater Acoustic Sensor Networks. Verkkodokumentti. <[http://senseslab.di.uniroma1.it/administrator/components/com\\_jresearch/files/publications/R\\_CARP\\_A\\_Reputation\\_Based\\_Chan.pdf](http://senseslab.di.uniroma1.it/administrator/components/com_jresearch/files/publications/R_CARP_A_Reputation_Based_Chan.pdf)> Luettu 3.7..2016.
41. Gee Keng Ee; Chee Kyun Ng; Nor Kamariah Noordin; Borhanuddin Mohd. Ali. 2010 A Review of 6LoWPAN Routing Protocols. Verkkodokumentti. APAN. <[http://journals.sfu.ca/apan/index.php/apan/article/viewFile/81/pdf\\_38](http://journals.sfu.ca/apan/index.php/apan/article/viewFile/81/pdf_38)> Luettu 3.7.2016.
42. Pascal, Thubert. 2014. 6TiSCH + RPL. Verkkodokumentti. <<http://www.slideshare.net/pascalhubert/rpl-telecom-bretagnengv2>> Luettu 3.7.2016.
43. Thubert P; Watteyne T; Wang Q. 2016. Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e. Verkkodokumentti. <<http://www.potaroo.net/ietf/ids/draft-ietf-6tisch-terminology-07.txt>> Luettu 3.7.2016.
44. IPv6 related standards and effort page – 6TiSCH. Verkkodokumentti. IoT6. <<http://iot6.eu/6tisch>> Luettu 3.7.2016.
45. Hong; Gomez; Choi; Ko. 2016. IPv6 over Constrained Node Networks(6lo) Applicability & Use cases. Verkkodokumentti. <[https://datatracker.ietf.org/doc/draft-hong-6lo-use-cases/?include\\_text=1](https://datatracker.ietf.org/doc/draft-hong-6lo-use-cases/?include_text=1)> Luettu 3.7.2016.
46. Kozierok, Charles M. 2005. The TCP/IP Guide. Verkkodokumentti. <[http://www.tcpiptime.com/free/t\\_SessionLayerLayer5.htm](http://www.tcpiptime.com/free/t_SessionLayerLayer5.htm)> Luettu 5.7.2016.
47. Zalman, Tara. 2015. Internet of Things Protocols and Standards. Verkkodokumentti. <[http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot\\_prot/#MQTT](http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/#MQTT)> Luettu 3.6.2016.
48. MQTT Essentials. Verkkodokumentti. HiveMQ. <<http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe>> Luettu 5.8.2016.
49. Rahul, Gupta. 2014. 5 Things to Know About MQTT – The Protocol for Internet of Things. Verkkodokumentti. IBM. <[https://www.ibm.com/developerworks/community/blogs/5things/entry/5\\_things\\_to\\_know\\_about\\_mqtt\\_the\\_protocol\\_for\\_internet\\_of\\_things?lang=en](https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_mqtt_the_protocol_for_internet_of_things?lang=en)> Luettu 5.8.2016.
50. Vasters, Clemens. 2015. The AMQP 1.0 Protocol - 1/6 - Introduction AMQP. Verkkodokumentti. <<https://www.youtube.com/watch?v=ODpeldUdClc>> Luettu 5.8.2016.

51. Zalman, Tara. 2015. Internet of Things Protocols and Standards Verkkodokumentti. <[http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot\\_prot/#AMQP](http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/#AMQP)> Luettu 3.6.2016.
52. AMQP is the Internet Protocol for Business Messaging. 2016. Verkkodokumentti. Oasis <<https://www.amqp.org/about/what>> Luettu 5.8.2016.
53. Carsten, Bormann. CoAP RFC 7252 Constrained Application Protocol. Verkkodokumentti. <<http://coap.technology/>> Luettu 5.8.2016.
54. An Overview of XMPP. Verkkodokumentti. XMPP.org. <<http://xmpp.org/about/technology-overview.html>> Luettu 6.8.2016.
55. Gakwaya, Daniel. 2016. A friendly introduction to XMPP. Verkkodokumentti. Blikoon. <<http://www.blikoon.com/xmpp/xmpp-a-soft-friendly-introduction>> Luettu 6.8.2016.
56. Holdmann, Michael. 2014. Efficient, Secure, Private IoT Architecture with XMPP Service Bus. Verkkodokumentti. <<https://www.linkedin.com/pulse/20140609124855-3975800-efficient-secure-private-iot-architecture-with-xmpp-service-bus>> Luettu 6.8.2016.
56. What is DDS. Verkkodokumentti. Object Management Group. <<http://portals.omg.org/dds/what-is-dds-3/>> Luettu 6.8.2016.
57. DDS Standard. Verkkodokumentti. Real-Time Innovations. <<https://www.rti.com/products/dds/omg-dds-standard.html>> Luettu 6.8.2016.
58. Security in the internet of things -Lessons from the Past for the Connected Future. 2015 Verkkodokumentti. Wind River Systems. <[http://www.windriver.com/whitepapers/security-in-the-internet-of-things/wr\\_security-in-the-internet-of-things.pdf](http://www.windriver.com/whitepapers/security-in-the-internet-of-things/wr_security-in-the-internet-of-things.pdf)> Luettu 10.8.2016.
59. Hype Cycle of emerging technologies. 2012. Verkkodokumentti. Science Direct <<http://www.sciencedirect.com/science/article/pii/S0167739X13000241#f000010>> Luettu 10.8.2016.
60. Blanger, David. 2015. What Does IoT + Big Data Mean to You? Verkkodokumentti. ECN, Advantage Business Media. <<https://www.ecnmag.com/blog/2015/04/what-does-iot-big-data-mean-you>> Luettu 10.8.2016.
61. Burtscher, Bertram; Fritz, Gernot. 2015. Big Data: who owns and who may use and exploit big data? Verkkodokumentti. Freshfields Bruckhaus Deringer. <[http://www.freshfields.com/en/global/Digital/Big\\_Data/?LangType=2057](http://www.freshfields.com/en/global/Digital/Big_Data/?LangType=2057)> Luettu 10.8.2016.
62. Alanko Markku. Laki – innovoinnin este vai kohde? Verkkodokumentti. Bigdata.fi. <<http://www.bigdata.fi/blogi/vierailijakirjoitus/laki-innovoinnin-este-vai-kohde>> Luettu 10.8.2016.



63. Arduino Power Consumption Normal & Sleep. 2013. Verkkodokumentti. Gadget Maker's <<http://gadgetmakersblog.com/arduino-power-consumption/>> Luettu 9.10.2015.
64. PSoC1 Power Savings Using Sleep Mode. Verkkodokumentti. Cypress. <<http://www.cypress.com/file/43371/download>> Luettu 9.10.2015.
65. PSoC4 Low-Power Modes and Power Reduction Techniques. Verkkodokumentti. Cypress. <<http://www.cypress.com/file/121271/download>> Luettu 9.10.2015.
66. PSoC4: PSoC4200 Family Datasheet. 2016. Verkkodokumentti. Cypress. <<http://www.cypress.com/file/138656/download>> Luettu 9.10.2015.
67. Tiny Serial Digital Thermal Sensor TC74. Microchip. Verkkodokumentti. Mouser electronics. <<http://www.mouser.com/ds/2/268/21462c-73653.pdf>> Luettu 9.10.2015.
68. DS18B20 Programmable Resolution 1-Wire Digital Thermometer. 2015. Verkkodokumentti. Maxim Integrated Products. <<http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>> Luettu 9.10.2015.
69. SIM900 GSM / GPRS. Verkkodokumentti. SIMCOM. <<http://simcom.ee/modules/gsm-gprs/sim900/>> Luettu 10.10.2015.
70. SIM5320. Verkkodokumentti. SIMCOM. <<http://simcom.ee/modules/wcdma-hspa/sim5320/>> Luettu 10.10.2015.
71. SIM7100. Verkkodokumentti. SIMCOM. <<http://simcom.ee/modules/lte/sim7100e/>> Luettu 10.10.2015.
72. Guidelines for Reliable Long Line 1-Wire Networks. 2008. Verkkodokumentti. Maxim Integrated Products. <<https://www.maximintegrated.com/en/app-notes/index.mvp/id/148>> Luettu 11.10.2015.
73. 1-Wire tutorial. Verkkodokumentti. Maxim Integrated Products. <<https://www.maximintegrated.com/en/products/1-wire/flash/overview/index.cfm>> Luettu 11.10.2015.
74. Choosing the Right 1-Wire® Master for Embedded Applications. 2008. Verkkodokumentti. Maxim Integrated Products. <<https://www.maximintegrated.com/en/app-notes/index.mvp/id/4206>> Luettu 4.3.2016.
75. Online course on Embedded Systems, module - 7 Serial Communication. Verkkodokumentti. EEHerald <<http://www.eeherald.com/section/design-guide/esmod7.html>> Luettu 11.10.2015.
76. RS-232 vs. TTL Serial Communication. 2010. Verkkodokumentti. Sparkfun. <<https://www.sparkfun.com/tutorials/215>> Luettu 11.10.2015.

77. Serial Communication. Verkkodokumentti. Sparkfun.  
<<https://learn.sparkfun.com/tutorials/serial-communication>> Luettu 11.10.2015.
78. Introduction to AT Commands. Verkkodokumentti. Developer's Home  
<<http://www.developershome.com/sms/atCommandsIntro.asp>> Luettu 4.5.2016.
79. SIM900 AT Commands Manual V1.11. 2015. Verkkodokumentti. SIMCom Wireless Solutions.  
<[http://media.micel.ee/simcom/SIM900/SIM900\\_AT%20Command%20Manual\\_V1.11.pdf](http://media.micel.ee/simcom/SIM900/SIM900_AT%20Command%20Manual_V1.11.pdf)> Luettu 4.5.2016.
80. Paller, Gabor; Szármes, Péter; Élő, Gábor. 2015. Power Consumption Considerations of GSM-connected Sensors in the AgroDat.hu Sensor Network. Verkkodokumentti. Research Gate.  
<[https://www.researchgate.net/publication/280154945\\_Power\\_Consumption\\_Considerations\\_of\\_GSM-connected\\_Sensors\\_in\\_the\\_AgroDathu\\_Sensor\\_Network](https://www.researchgate.net/publication/280154945_Power_Consumption_Considerations_of_GSM-connected_Sensors_in_the_AgroDathu_Sensor_Network)> Luettu Luettu 4.5.2016.
81. Introduction to the GPRS standard. 2016. Verkkodokumentti. CCM Benchmark group <<http://ccm.net/contents/693-the-gprs-standard>> Luettu 4.5.2016.