Tapani Heino

Älykäs äänestysjärjestelmä

Zigbeellä toteutettu mesh-pohjainen älyverkko

Metropolia Ammattikorkeakoulu
Insinööri (ylempi AMK) -tutkinto
Automaation koulutusohjelma
Opinnäytetyö
4.3.2011

Metropolia

Tiivistelmä

Tekijä(t) Otsikko Sivumäärä Aika	Tapani Heino Älykäs äänestysjärjestelmä. Zigbeellä toteutettu mesh-pohjainen älyverkko 61 sivua + 2 liitettä 4.3.2011
Tutkinto	insinööri (ylempi AMK)
Koulutusohjelma	automaatioteknologia
Suuntautumisvaihtoehto	automaation tietotekniikka
Ohjaaja	lehtori Jukka-Pekka Pirinen

Opinnäytetyössä selvitettiin langattomien mesh-verkkojen ominaisuuksia tekemällä älykäs äänestysjärjestelmä. Työssä perehdyttiin aluksi langattomiin tekniikoihin, niiden avulla mesh-tekniikkaan, ja tämän jälkeen tutustuttiin saatavilla oleviin laiteratkaisuihin. Näiden avulla pystyttiin määrittelemään ja toteuttamaan oppilaspohjainen älykäs äänestysjärjestelmä.

Työssä tutkittiin aluksi langattomien sekä mesh-verkkojen ominaisuuksia tarkasti, jotta niiden ominaisuuksia pystyttäisiin hyödyntämään käytettävässä työssä. Näiden tietojen avulla päädyttiin Zigbee-mesh-tekniikkaan, josta saatiin isoin hyöty rakennettavassa järjestelmässä.

Työn rakennusvaiheeseen tilattiin Digiltä kehitystyökalut, joiden avulla oli tarkoitus rakentaa vastausjärjestelmä, joka eroaa olemassa olevista ratkaisuista mesh-ominaisuuksiensa ja älykkyytensä ansiosta. Järjestelmään rakennettiin älykkyys jokaiseen solmupisteeseen, aina lähettimistä tietojen lukuun asti.

Työn haaste tuli järjestelmän laajuudesta ja prosessoitavan tiedon kuljetuksesta. Oppilaiden vastaukset piti pystyä kuljettamaan lähettimiltä verkon läpi koordinaattorille, josta ne siirtyivät IP-pohjaisena verkkoon XML-muodossa serverille, josta vastaukset pystyttiin lukemaan miltä tahansa verkossa olevalta koneelta Excelin avulla.

Käytännön projekti onnistui kohtuullisesti. Tiedon prosessoinnissa havaittiin jonkin verran virhekirjoituksia, jotka alensivat järjestelmän luotettavuutta. Lähetyksen varmentaminen olisi vaatinut tarkempaa operointia tiedon prosessoinnissa. Toisekseen huomattiin, että tällaisen järjestelmän toteutus mesh-pohjaisena ei tuo järjestelmään mitään selkeää lisäarvoa.

Loppujen lopuksi työhön oltiin kuitenkin tyytyväisiä. Sen avulla selvitettiin että mesh-verkot ovat jo hyvin pitkälle hiottuja, ja niillä pystyy luomaan sovelluksia, joilla saadaan merkittävää lisäarvoa monille järjestelmille. Niiden itsekorjautuvuus ja älykkyys taannevat niiden voimakkaan yleistymisen tulevaisuudessa.

Avainsanat	mesh, äänestysjärjestelmä, älykäs

Metropolia

Abstract

Author(s) Title Number of Pages Date	Tapani Heino Intelligent voting system. Intelligent voting system implemented with a ZigBee mesh network 61 pages + 2 appendices 4 March 2011
Degree	Master of engineering
Degree Programme	Automation technology
Specialisation option	Automation information technology
Instructor	Jukka-Pekka Pirinen, Lecturer

This thesis project investigated the charasteristics of wireless mesh networks by making an intelligent voting system. The project started by examining wireless technologies and mesh technology and by then exploring the hardware solutions available for them. With the help of this information, it was possible to define and implement a school-based intelligent voting system.

First, wireless technologies and mesh utilities were carefully studied so that their properties could be used in the project. On the basic of the results, Zigbee technology was selected because it offered the biggest benefits for the project.

At the start of the construction phase development kits were ordered from Digi to build an intelligent voting system which differs from the existing systems by its mesh qualities and intelligence. In the system intelligence was built in each node, from transmitters to readers.

The challenge of the project derived from the scope of the system and data transportation. Students' replies had to be able to be carried through the net from the coordinator, from where the reply moved to the IP-based network, to the server in XML format, and from where the reply could be read on any computer in the network by using Excel.

The project succeeded reasonably well. In data processing there were found some error some writings, which reduced the reliability of the system. Certification of transmission would have required further operations in data processing. In addition, it was discovered that the mesh system does not provide any clear added value to the system.

All in all, the results of the project were satisfying. It was found that the mesh networks are already highly polished and they can be used to create applications which yield significant added value for many systems. Self-healing and intelligence features will enable future mushrooming of mesh networks.

Keywords	mesh, voting system, intelligence

Sisällys

1 Johdanto 1

2 Älykkäät anturiverkot 3

2.1 Määritelmä 3
2.2 RF:n Perusteet 4
2.3 Langattomat mesh-verkot 9
2.4 Langattomien standardien vertailu 12
2.5 Projektin laitteiston valinta 14
2.6 Zigbee 15
2.7 Sovellukset 19
2.8 Interaktiivinen opiskelu 22

3 Testijärjestelmän suunnittelu ja toteutus 23

3.1 Määritys 23
3.2 Laitteisto 23
3.3 Ohjelmisto 29
3.4 Suunnittelu 31
3.5 Työn kulku 33
3.5.1 Verkon määritys 34
3.5.2 Prosessorien ohjelmointi 39
3.5.3 Kaukosäätimen rakentaminen 41
3.5.4 Tiedon prosessointi 45

4 Työn testaus 57

5 Päätelmät 61

Lähteet 62

Liitteet

Liite 1. IEEE 802-standardit

Liite 2. Xbee PRO S2B

1 Johdanto

Opinnäytetyön tarkoitus on perehtyä älykkäisiin anturiverkkoihin. Tätä varten työssä määritellään, mitä ne ovat ja missä niitä käytetään, ja rakennetaan niiden avulla älykäs äänestysjärjestelmä. Tämän jälkeen analysoidaan työn onnistumista, sekä projektista saadun datan avulla mietitään, minkälainen nykyaikainen tekniikka on ja mihin tämä tulee kehittymään.

Antureiden perustarkoituksena on tuottaa mittausdataa ja tämä tieto täytyy jotenkin välittää eteenpäin prosessoitavaksi. Tämä opinnäytetyö perehtyy juuri tähän prosessointi- ja tiedonvälitysvaiheeseen. Yksi uusimmista kehityshaaroista on itsenäiset älyverkot, jotka pystyvät viestimään ja tekemään päätöksiä omatoimisesti. Nämä verkot tarjoavat antureille alustan, jolla niiden tuottamaa dataa voidaan suoraan prosessoida ja välittää paikan päällä. Tämä kehitys on ollut mahdollista pääsääntöisesti kommunikointi- ja verkottumistapojen monipuolistumisen sekä tekniikoiden vähävirtaistumisen johdosta.

Älykkäät anturiverkot ovat syntyneet hyvin pitkälle teollisuuden tarpeista kontrolloida tieto- ja materiaalivirtoja. Kilpailun kiristyessä niiden avulla haetaan yhä kustannustehokkaampaa tapaa toimia. Tekniikan kehittyessä ja laitteiden yleistyessä ne ovat alkaneet yleistyä yhä useammilla muillakin aloilla, kuten kuluttajatuotteissa, terveydenhuollossa jne.

Erilaisia tekniikoita tiedonsiirtoon, protokolliin ja mittausdatan prosessointiin on tarjolla hyvin mittava määrä. Tätä silmällä pitäen rakennetaan oma sovellus, jolla testataan, kuinka vaikea valinta on erilaisten tekniikoiden välillä. Samalla luonnollisesti halutaan nähdä, kuinka pitkälle tämän päivän tekniikka taipuu. Tämän jälkeen tutkitaan sovelluksen toimivuutta ja tätä kautta myös laitteistojen mahdollisia pullonkauloja.

Sovellukseksi on määritelty älykäs äänestysjärjestelmä. Tällä on tarkoitus pystyä pitämään oppilasluentoja interaktiivisesti. Äänestysjärjestelmässä on tarkoitus saada tehtyä demokäyttöön yksilöllinen langaton pienikokoinen äänestyslaite, jolla voidaan vastailla esitettäviin kysymyksiin. Kerätyt tiedot pitäisi tämän jälkeen saada

prosessoitua helposti esitettävään muotoon. Älykkyys järjestelmään taas kehitetään laitteiston omatoimisen verkottumisen, tiedon prosessoinnin ja äänestyslaitteen ympärille.

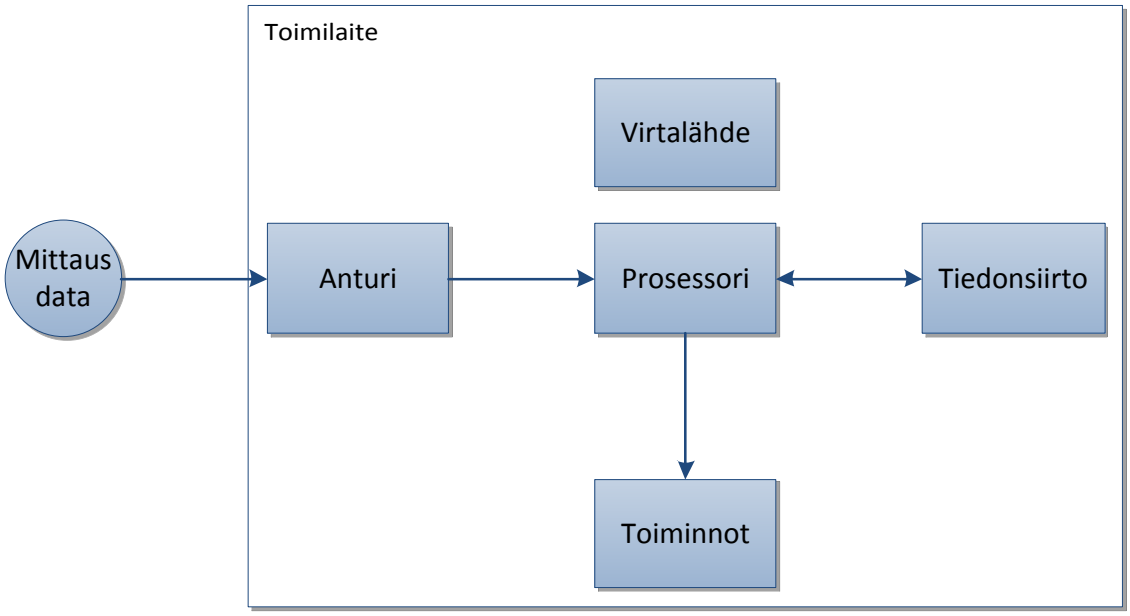
Lopuksi vielä pohditaan nykyisiä älykkäitä anturiverkkoja ja käytännön projektin tuloksien avulla mietitään, onko nykyinen tekniikka vielä lapsen kengissä vai jo nyt täysiverisesti hyödynnettävää. Samalla visioidaan, mihin suuntaan ja millä vauhdilla tekniikka tulee jatkossa kehittymään.

2 Älykkäät anturiverkot

2.1 Määritelmä

Älykäs anturiverkko ei ole mikään tietty yksittäinen laite tai keino, vaan osiensa summa. Osat taas koostuvat laitteiston verkottumisesta sekä kyvystä prosessoida dataa. Itsenäisyys erottaa tämän laitteiston perinteisistä ratkaisuista, eli älykäs anturiverkko pystyy verkottumaan, prosessoimaan dataa, tekemään tarvittaessa päätöksiä ja kaikkea edellä mainittua itsenäisesti.

Verkot muodostuvat eri topologisista rakenteista, ja ne pitävät sisällään erilaisia toimilaitteita. Kuvassa 1 on hahmoiteltu yksinkertainen karkea malli toimilaitteesta. Verkko koostuu vaihtelevasta määrästä toimilaitteita, joilla kaikilla on oma funktionsa. Kuva auttaa hahmottamaan, että laitteet eivät sinänsä tarjoa mitään uutta ja mullistavaa. Perustoimilaitteesta taas saadaan eri tason laite muuttamalla sen tiedonsiirtotapa mesh-pohjaiseksi ja ohjelmoimalla paikalliseen mikroprosessoriin lisää toiminnallisuutta tarvittaessa. Tämän jälkeen toimilaite voi toimia itsenäisesti osana erilaisia topologisia verkkoja.



Kuva 1. Yksittäinen toimilaite

Tämän tekniikan mahdollistaa tiedonsiirron kehittyminen ja laitteiston vähävirtaistuminen. Tiedonsiirrossa tärkeitä asioita ovat langattomat reitittävät mesh-verkot, joiden avulla laitteet voivat muodostaa erimuotoisia topologisia verkkoja yksinkertaisesti, sekä uudet tiedonsiirtoprotokollat, jotka tukevat tätä tekniikkaa ja ovat tietoturvallisia sekä vähävirtaisia. Prosessorien ja oheistekniikan kehittyminen taas tulee toimimaan yhä pienemmillä virtamäärillä ja mahdollistaa laitteiden itsenäistymisen kiinteistä virtalähteistä paristotoimintaan, jolloin laitteiden sijoittaminen ei ole enää paikkariippuvainen. Seuraava luonnollinen kehityslähde lienee vihreä virta, jolloin laitteet keräävät ympäristöstään itse tarvitsemansa virran.

2.2 RF:n perusteet

Langaton tiedonsiirto on nykyään hyvin yleistä. Se on käytännössä levinnyt kaikille toimialoille, ja laajaan määrään tekniikan sovelluksia. Langatonta tiedonsiirtoa pystyy toteuttamaan erilaisilla tekniikoilla, mutta kaikkein yleisin on radioaaltoihin (radio frequency, rf) perustuva tekniikka.

Perustoiminta

Langaton radiotekniikka perustuu yksinkertaisimmillaan seuraaviin kompenetteihin:

-lähetin ja vastaanotin
-lähettimen antenni
-vastaanottimen antenni
-ympäristö

Kuvassa 2 on havainnollistettu yksinkertaisesti lähetyksen ja vastaanoton toimintaa. Tieto on lähettevä vahvistimen kautta, josta se kulkee ympäristön lävitse vastaanottajalle vahvistimen kautta. [1]

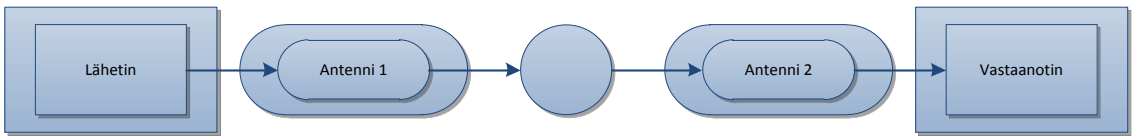
Kuva 2. RF:n yksinkertaistettu toiminta

Kuuluvuuden maksimointi

Lähetyksen pituuden maksimoinnissa on kiinnitettävä huomiota seuraaviin asioihin:

-lähettimen tehon (TX) lisääminen
-vastaanottimen herkkyyden (RX) lisääminen
-antennien vahvistuksen lisääminen
-ympäristön esteettömyys lähettimeltä vastaanottimelle

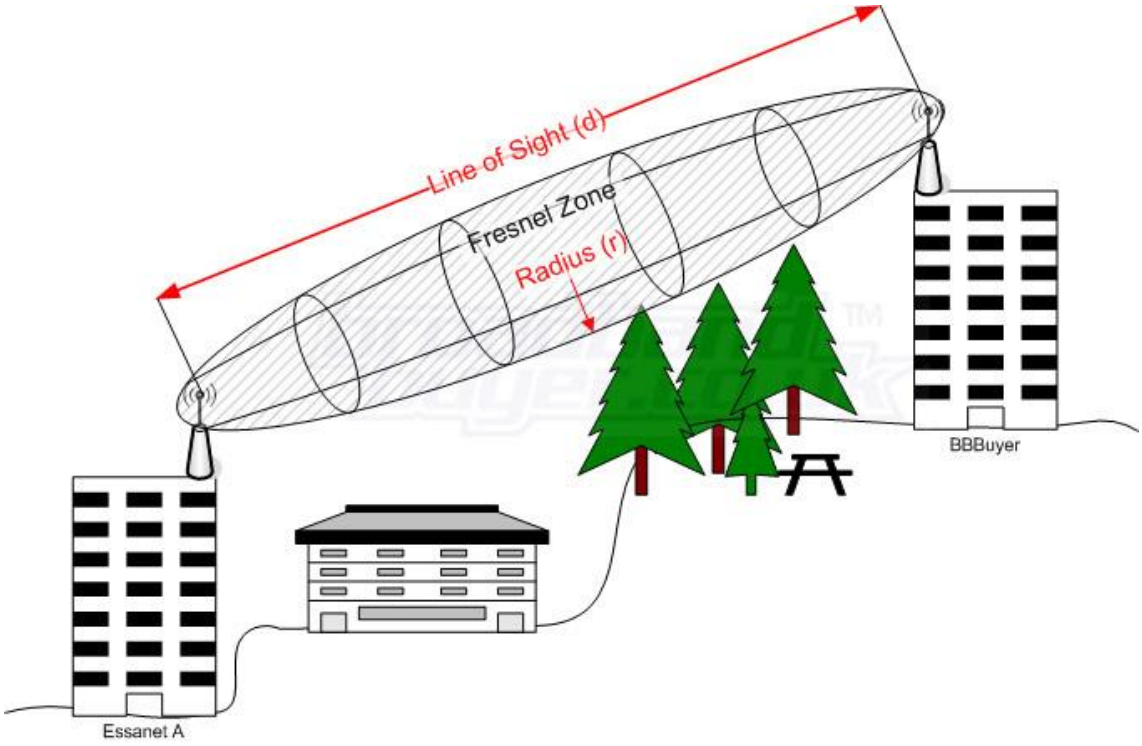
Kuvassa 3 on havainnollistettu parempaa kuuluvuutta. Lisää kantamaa tiedonsiirtoon saadaan kasvattamalla lähettimen tehoa, lisäämällä antennin vahvistusta ja hävittämällä ympäristön esteet tieltä. Lopuksi on vielä parannettava vastaanottavan antennin vahvistusta ja lisättävä vastaanottimen herkkyyttä. Tällä tavalla päästään parempaan lopputulokseen kuin aikaisemmassa esimerkissä. [1]



Kuva 3. Kantaman maksimointi.

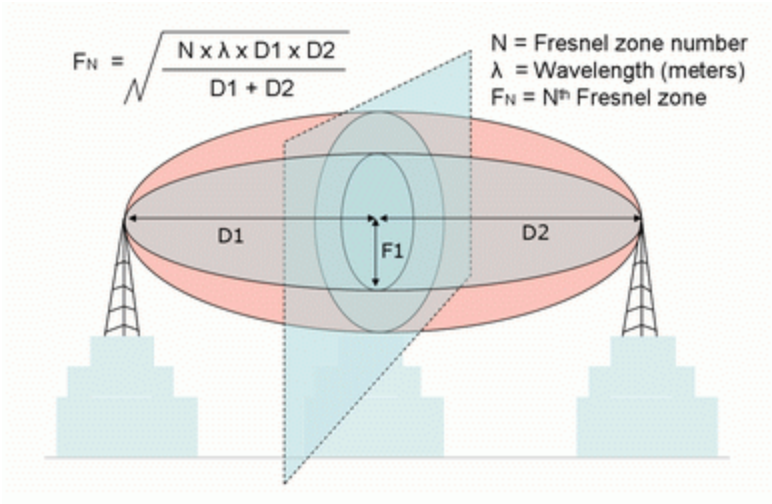
Fresnel Zone

RF:ssä näkyvyys (Line of Sight) ei ole suora viiva vaan amerikkalaisen jalkapallon muotoinen ellipsi alue kahden antennin välillä. Fyysikko Augustin-Jean Fresnelin mukaan nimetyn Fresnel Zonen avulla pystytään laskemaan tarvittava tyhjä alue antennien suoran näköyhteyden ylä- ja alapuolella. Käytännössä jos antennien välissä on esteitä, lähetysteho heikkenee esteiden määrästä riippuen lopulta pisteeseen, jolloin yhteys ei enää toimi ollenkaan. Käytännön sääntönä on, että esteiden maksimi saa olla 40 prosenttia, mutta suositus hyvän yhteyteen on 20 prosenttia tai alle. [1]



Kuva 4. Fresnel Zone [2]

Fresnelin vyöhykkeen säteiden laskentaan voidaan käyttää kaavaa approksoimaan niiden etäisyyttä. Kuvassa 5 on esitetty kaava ja yksinkertainen kuva sen käytöstä. Käytännössä Fn tarkoittaa Fresnelin vyöhykkeen sädettä, mitä lasketaan, d1 ja d2 tarkoittavat etäisyyksiä antenneihin metreissä, lambda taas laskettavan aallon pituutta, ja N kuvaa, monetta Fresnelin vyöhykettä laskussa käytetään.



Kuva 5. Fresnelin kaava [3]

Myös valmiiksi laskettuja taulukkoarvoja Fresnelin säteistä löytyy. Taulukossa 1 on lueteltu muutamia säteen halkaisijoita eri etäisyyksillä 868 MHz:n ja 2,4 Ghz:n taajuuksilla. Vertailemalla taulukon 868 MHz:n arvoja eri etäisyyksillä voidaan huomata, että ellipsin arvon kasvu suhteessa etäisyyden kasvuun ei kasva samassa suhteessa.
Jos loitonnamme antenneja yhdestä metristä sataan metriin, ei ellipsin halkaisija satakertaistu, vaan karkeasti kymmenkertaistuu. [4]

Taulukko 1. Fresnelin taulukkoarvot ellipsin säteestä eri etäisyyksillä.

Etäisyys	**868 Mhz Modeemit**	**2.4 Ghz Modeemit**
1 m	0,29 m	0,18 m
3 m	0,51 m	0,31 m
10 m	0,93 m	0,56 m
30 m	1,61 m	0,97 m
100 m	2,94 m	1,77 m
300 m	5,09 m	3,06 m
1600 m	11,76 m	7,07 m
8 km	26,29 m	15,81 m
16 km	37,18 m	22,36 m

Taajuuden valinta

Taajuuden valitsemiseen Rf:ssä on otettava huomioon useampi asia. Ensimmäisenä vaikuttaa maantieteellinen sijainti. Maailmanlaajuisesti operoitaessa on käytettävä 2,4 GHz:n taajuutta. Paikallisesti jollain tietyllä alueella taas, voidaan käyttää myös alueellista taajuutta, esim. Euroopassa käytetään lisenssivapaita taajuuksia 433 MHz:n, 868 Mhz:n ja 5.7 GHz:n alueilla.

Toinen tärkeä tekijä on Rf:n suorituskyky. Tätä varten pitää miettiä haluttua kantamaa, radioaaltojen läpäisykykyä sekä tarvittavia antenniratkaisuja. Näistä saatujen kriteerien avulla voidaan valita toimivin ratkaisu.

Kolmantena asiana on muistettava, että joka mantereella on omat lainsäädännöt ja näitä valvovat instanssit. Yhdysvalloissa valvontaa suorittaa FCC, Euroopassa ETSI ja osittain APAC. Suunnittelussa on otettava huomioon, että taajuudet ovat tarkasti

reguloituja ja määriteltyjä. Uuteen paikkaan järjestelmää lokalisoitaessa on paikalliset määräykset ja velvoitteet muistettava tarkistaa, koska nämä eivät välttämättä ole suoraan sellaisenaan valmiita impletoitamaan, varsinkin mannerten tasolla liikuttaessa. [4]

Radiomodeemin toiminta

Radiomodeemin toiminta on periaatteellisella tasolla hyvin yksinkertaisesta. Modeemin tehtävä on moduloida ja demoduloida. Modulointi taas tarkoittaa yksinkertaistettuna tiedon lähettämistä jonkin siirtotien avulla, kuten esimerkiksi radioaaltojen välityksellä. Modulointi itsessään sovittaa siirrettävän tiedon siirtotielle. Nykyaikaiset radiotiet käyttävät FHSS- ja DSSS-modulointimenetelmiä tiedonsiirrossa. [4]

FHSS

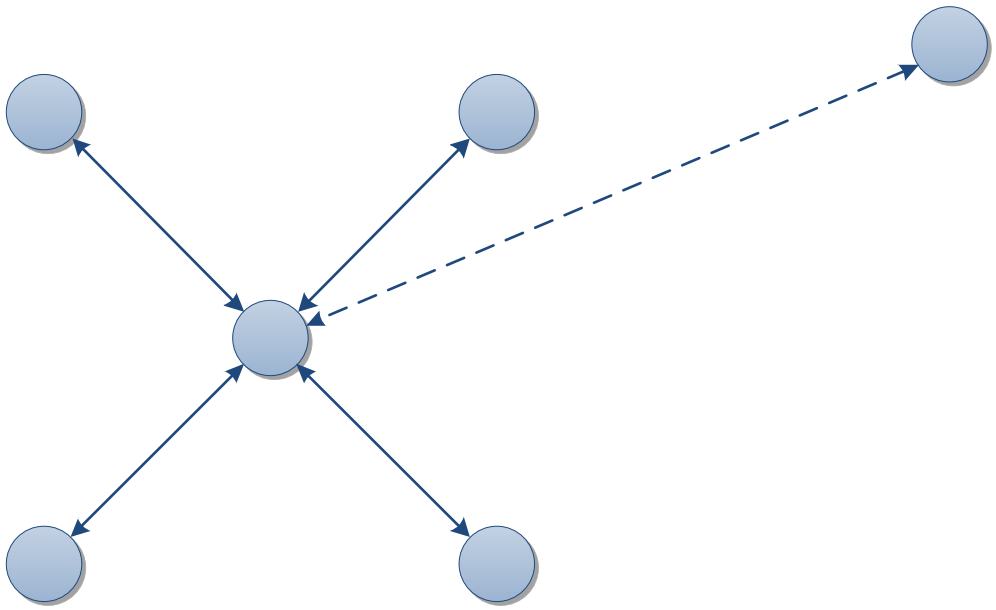
FHSS-lyhenne tulee sanoista Frequency Hopping Spread Spectrum ja suomennettuna puhutaan taajuushyppelystä. Taajuushyppelyssä tietoa lähetetään esimerkiksi 2400 MHz:n ja 2480 MHz:n taajuuksilla välillä satunnaisesti tai etukäteen sovitulla tavalla, siten että sekä lähettäjä ja vastaanottaja tietävät lähetystavan, eli vastaanotto seuraa lähetintä. Taajuusalue jaetaan suureen määrään kanavia tietyllä alueella, tieto pilkotaan osiin, ja jokainen osa lähetetään tietyin aikavälein omilla kanavillaan. Koko taajuusaluetta käytetään hyväksi, vaikka tieto siirtyykin tietyllä ajanhetkellä vain tietyllä taajuusalueella.

DSSS

DSSS lyhenne tulee sanoista Direct Sequency Spread Sprectrum, ja suomennettuna puhutaan suorasekvenssoinnista. Tässäkin tekniikassa käytetään kaistaa useamman megahertsin välillä samaan tyyliin kuin FHSS:ssa, ja tieto pilkotaan pieniin osiin ja lähetetään koko taajuusalueella. Erona on se, että lähettävä tieto lähetetään yhtenä signaalina. Tästä tulee se ongelma, että muut signaalit saattavat aiheuttaa ongelmaa. Tästä johtuen lähetettävä tieto sekoitetaan kohinamaiseen kantoaaltoon. Sekoittamalla nämä kaksi signaalia pystytään haluttu tieto lähettämään ja purkamaan vastaanottimella erottamalla kantoaalto varsinaisesta tiedosta.

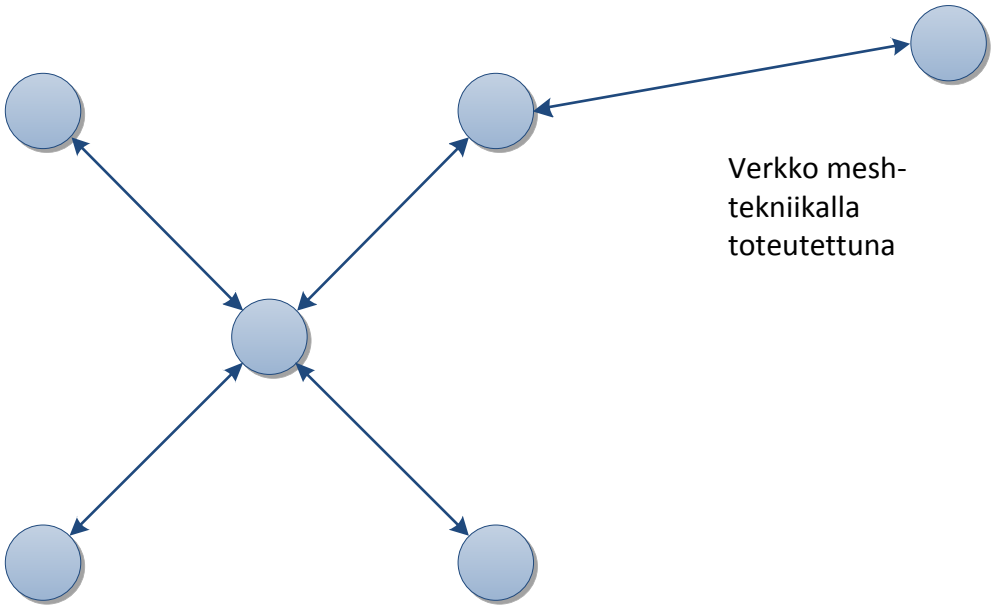
2.3 Langattomat mesh-verkot

Langaton mesh-verkko, eli wireless mesh tarkoittaa langatonta reitittävää verkkoa. Perinteisen verkon huonoina puolina on, että jos tukiasemaa vaihdetaan, voi yhteys katketa, ja jos jokin uusi laite haluaa liittyä verkkoon kantaman ulkopuolelta, on se mahdotonta. Kuvassa 6 havainnollistetaan yksinkertaisen point-to-multipoint-verkon ongelmaa.



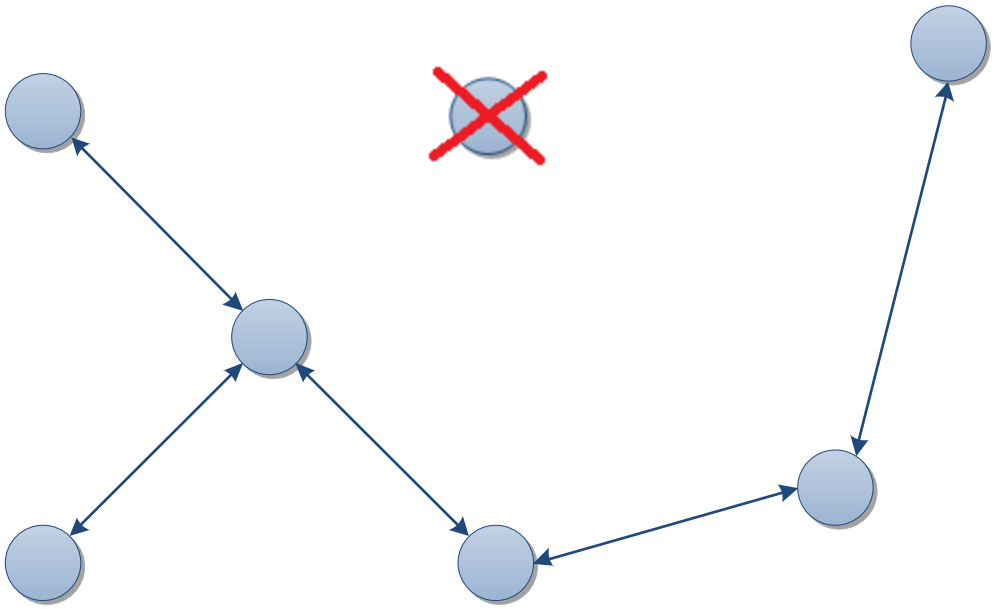
Kuva 6. Perinteisen verkon liittymisongelma.

Tekemällä sama järjestelmä mesh-pohjaisesti kuvan 7 mukaisesti pystytään tämä ongelma ratkaisemaan. Mesh-verkon ensimmäinen ja päällimmäisin etu on verkon laajentaminen. Terminä puhutaan mesh node hoppingista, eli uudet laitteet voivat kiinnittyä verkon mihin tahansa solmukohtiin sen sijaan, että niiden tarvitsisi suoraan muodostaa yhteys vain yhteen tiettyyn tukiasemaan.



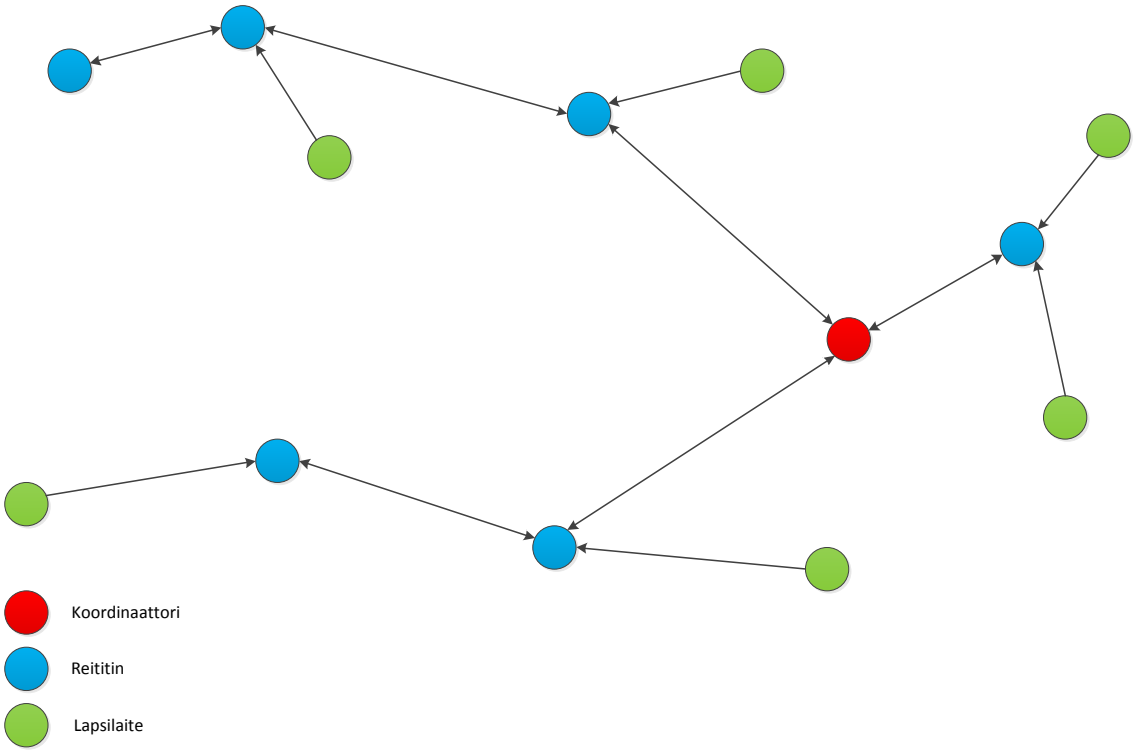
Kuva 7. Verkko toteutettuna mesh-pohjaisesti.

Mesh-verkkojen seuraavat kaksi hyötyä saadaan johdattemalla ajatusta eteenpäin siitä, että laitteet voivat vapaasti liittyä verkossa eri solmupisteisiin. Ensimmäinen hyöty tulee otettaessa uusi toimilaite käyttöön. Sen voidaan antaa liittyä itse suoraan ensimmäiseen vapaaseen solmupisteeseen. Toinen merkittävä hyöty koskee vikatilanteita. Perinteinen verkko on hyvin häiriöherkkä, ja jos tukiasemaan tulee vikaa, voi koko verkko haavoittua. Mesh verkoissa taas laitteet voivat järjestyä uudestaan eri solmupisteiden kautta, eli verkko pystyy korjaamaan itse itseään kuvan 8 mukaisesti.

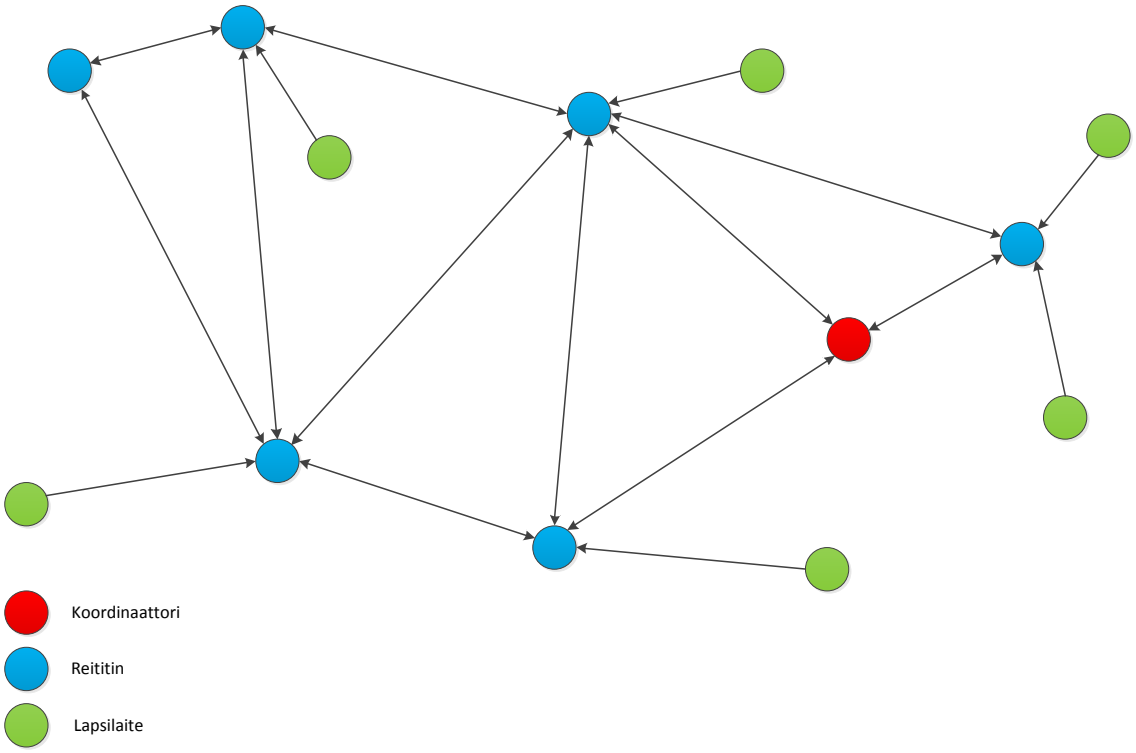


Kuva 8. Mesh-verkon uudelleen järjestäytyminen vikatilanteessa.

Hieman totuttelua ajatusmaailmassa vaatii tyypillisen mesh-verkon topologian ominaispiirre. Toimilaitteet jakaantuvat yleensä koordinaattoriin, reitittimiin ja päätelaitteisiin. Suhde, millä nämä laitteet toimivat, on aikuinen – lapsi-periaatteella. Koordinaattorit ja reitittimet toimivat aikuisina ja päätelaitteet lapsina. Aikuiset voivat puhua keskenään ja lasten kanssa, mutta lapset voivat puhua ainoastaan aikuisten kanssa. Verkon solmukohtien on siis oltava aikuisia. Tätä ajatusta on selvennetty kuvien 9 ja 10 avulla. Jos tehdään puuverkko, mikä sisältää kaikkia toimilaitetyyppejä, voidaan kuvan 9 avulla havainnollistaa, minkälainen se voisi olla. Kuvassa on näytetty, miten koordinaattorit, reitittemet ja päätelaitteet reitittyvät kaikki loogisimmalla tavalla koordinaattorin ympärille. Kuvassa 10 taas on havainnollistettu, miten reitittemet pystyvät tarpeen vaatiessa keskustelemaan myös keskenään. Jos jokin reitittimistä hajoaisi, etsisivät lapsilaitteet ja muut reitittemet uuden lyhyimmän reitin koordinaattorin luo.



Kuva 9. Puu-mesh-verkko koordinaattorilla, reitittimellä ja päätelaitteilla



Kuva 10. Puuverkko, jossa verkon jokainen solmukohta on havainnollistettu.

Yhteenvetona voidaan siis todeta, että mesh-verkoilla pystytään luomaan monipuolisia verkkoja, joihin on helppo lisätä uusia toimilaitteita. Mesh-verkot tukevat kaikkia verkon topologeja, niistä voidaan tehdä tähti-, peer-to-peer- ja klusterisia puuverkkoja. Vikatilanteissa nämä verkot pystyvät ryhmittymään uudestaan ja täten siis korjaamaan itse itseään. [5]

2.4 Langattomien standardien vertailu

Haluttaessa toteuttaa langaton lähiverkko on tiedostettava että täydellistä järjestelmää ei ole olemassakaan, vaan laitteiston käyttötarkoitus on määriteltävä tarkkaan ja valittava käytettävä järjestelmä sen pohjalta, mikä kohtaa parhaiten sille annetut kriteerit. Yksinkertaisena esimerkkinä voidaan todeta, että jos halutaan nopeaa tiedonsiirtoa pitkällä kantamalla, on se mahdollista, mutta hintana on kova virrankulutus, joten tällaista järjestelmää olisi siis mahdotonta toteuttaa paristopohjaisesti.

Taulukossa 2 on vertailtu tekniikoina erilaisten WiFin, Bluetoothin ja Zigbeen ominaisuuksia hahmoittamaan, kuinka joka tekniikalla on vahvuutensa ja heikkoutensa. WiFin ehdottomia vahvuuksia ovat sen tiedonsiirtonopeus ja kantama, mutta heikkouksina virran kulutus ja monimutkaisuus verkon kannalta. Zigbee taas on erittäin vähän virtaa vievä, yksinkertainen, hyvällä kantamalla varustettu ja todella laajalla verkon laajennuksilla tuettu, mutta sillä on heikko tiedonsiirtonopeus. Bluetooth on taas ikään kuin kompromissi näiden väliltä, sen ainoa selkeä heikkous on lyhyt kantama.

Taulukko 2. Vertailu erilaisista langattomista tekniikoista. [6]

Ominaisuudet	**IEEE 802.11b WiFi**	**IEEE 802.15.1 Bluetooth**	**IEEE 802.15.4 Zigbee**
Virran kulutus	Tunteja	Päiviä	Vuosia
Vaikeus	Erittäin vaikea	Vaikea	Yksinkertainen
Nodes\Master	32	7	64000
Viive	Max. 3 sekuntia	Max. 10 sekuntia	Max. 30 millisekuntia
Kantama	100-1000 m	10 m	70-300 m (ETSI), 1600 m (FCC)
Laajennattavuus	Roaming mahdollisuus	Ei	Kyllä
RF Datanopeus	11 Mbps	1 Mbps	250 Kbps
Tietoturva	Authentication Service Set ID (SSID)	64-bit, 12-8bit	128-bit AES, Application Layer user defined

2.5 Projektin laitteiston valinta

Tässä opinnäytetyössä suunnitellaan ja tehdään langaton älykaukosäädin, joten ominaisuudet, jotka korostuvat perusverkon kannalta ovat virrankulutus, laajennettavuus, helppokäyttöisyys ja tuki tarpeeksi monelle laitteelle verkossa. Älykkyyttä varten verkon täytyy tukea myös monimutkaisempia topologeja ja mesh-pohjaisuutta.

Näitä kriteerejä käyttäen kartoitetaan mahdollisia vaihtoehtoja IEEE 802 –standardista, joka käsittää LAN (Local Area Networks)- ja MAN (Metropolitan Area Networks) -verkot. Tutkailemalla IEEE 802 -standardeja liitteestä 1 voidaan havaita että projektiin sopivimmat tekniikat löytyvät alueelta 802.15.1 – 802.15.6. IEEE 802.15 käsittää alueena WPAN (Wireless Personal Area Network) -standardin.

Kun IEEE 802.15 -standardin listaa tutkitaan tarkemmin, voidaan havaita, että esiin nousee 802.15.4, joka vähävirtaisuudellaan ja mesh-networking ominaisuuksillaan kohtaa projektille annetut kriteerit. Standardi tukee mesh-verkkoja, ja jakaantuu kahteen osaan: low-rate WPAN- ja high-rate WPAN –mesh-verkkoihin. Tuki low-rate mesh-verkoille, joka perustuu IEEE 802.15.4 -standardiin, on tärkeä, koska tämän avulla virran kulutusta voidaan rajata merkittävästi ja laitteen toimintaikä paristoilla kasvaa parhaimmillaan vuosiin. Standardista löytyy tuki myös kaikille verkon perustoiminnoille, ja mesh-verkkojen perustoiminnoille, kuten solmukohtiin liittymisille.

2.6 Zigbee

Kun sopiva IEEE 802 -standardi on valittu, täytyy seuraavaksi kartoittaa, mikä järjestelmä tämän standardin päälle kannattaa rakentaa hyödyntämään mesh-verkkoja. Zigbee on tässä vertailussa vahvoilla. Syitä tähän on hyvin monia: kantama, vähävirtaisuus, helppokäyttöisyys, mesh-tuki, pieni fyysinen koko ja tietoturva. Muut järjestelmät karsiutuvat näiden ominaisuuksien puutteiden takia. Perinteinen RF-järjestelmä ei taivu mesh-pohjaisuuteen, Bluetoothin ja RFID:n kantama on liian huono, WiFi vie liikaa virtaa jne.

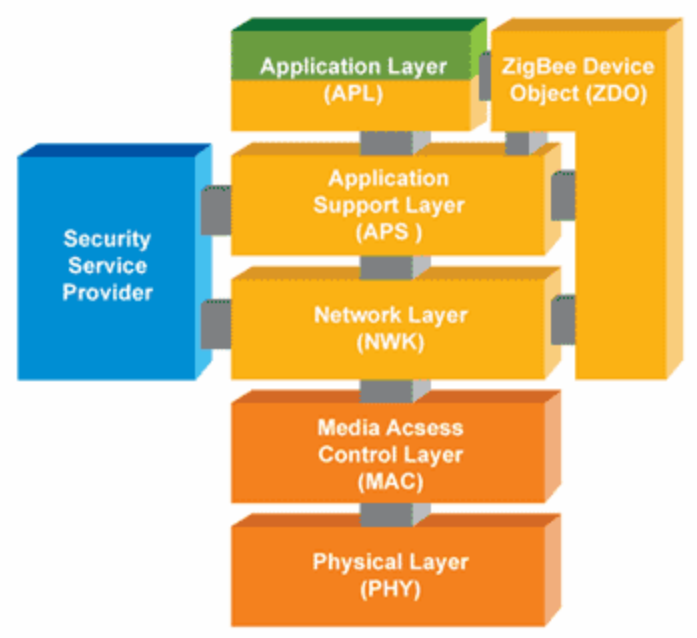
Zigbeen yhteydessä on hyvä mainita Zigbee Alliance, joka on 2002 perustettu avoin voittoa tavoittelematon yhdistys, jonka agendana on Zigbeen kehitys. Järjestön tavoitteena on ollut alusta asti kehittää yhtenäistä standardia joustavaan ja älykkäämpään verkottumiseen, josta on syntynyt Zigbee. Kehitys jatkuu koko ajan laajan jäsenistön tuella, järjestöön kuuluu monia suuria yrityksiä, kuten Intel ja Samsung. [7]

Zigbee on verkkoprotokolla joka perustuu 802.15.4 -standardiin. Zigbeen toimintaperiaatteen pohjana on 802.15.4 -standardi, ja tämän järjestelmän päälle on rakennettu oma mesh-verkkoja tukeva protokolla. Zigbeen hyviä puolia ovat mm. sen yhtenäinen standardi, valmistajien laaja tuki, langattomuuden luotettavuus ja vähävirtaisuus.

802.15.4 tukee itsessään point-to-point- ja tähtiverkkoja sekä koordinaattoria ja päätelaitteitteita. Kun tähän lisätään Zigbee, laajenevat verkon ominaisuudet. Tämän jälkeen laitteistoa pystyy käyttämään mesh-verkkona. Järjestelmä tukee myös reitittimiä, erilaisia verkkotopologeja ja verkkohyppyjä toimilaitteiden välillä koordinaattorille.

Kuvassa 11 on havainnollistettu Zigbeen protokollapinoa. Se on rakennettu perustuen IEEE 802.15.4 -standardiin, ja tästä johtuen kaksi alimmaista kerrosta ovat identtiset

kyseiseen standardiin. Ylemmät kerrokset on taas luotu Zigbeetä varten. Niiden avulla tuetaan verkon laajempaa toimivuutta, ja tarjotaan yksinkertainen liittymä verkon ja loppukäyttäjän sovellusten väliin.



Kuva 11. Zigbeen toimintaperiaate [8]

Zigbeen protokollan toiminta

Zigbee node-kohdat eli solmupisteet, mitä voi olla Pan (Personal area network) -verkoissa:

- Koordinaattori
  - Pan-verkossa voi olla vain yksi koordinaattori
  - Luo ja hallitsee Pan-verkkoa
  - Virrat aina päällä
- Reititin
  - Valinnainen
  - Voi olla useampia
  - Virrat aina päällä
- Loppulaite
  - Voi olla useita
  - Vähävirtainen

Datan lähetysmuodot:

- Broadcast
    - Tieto lähetetään koko verkkoon
- Unicast
    - Yksilöllinen lähetys

Verkon muodostus

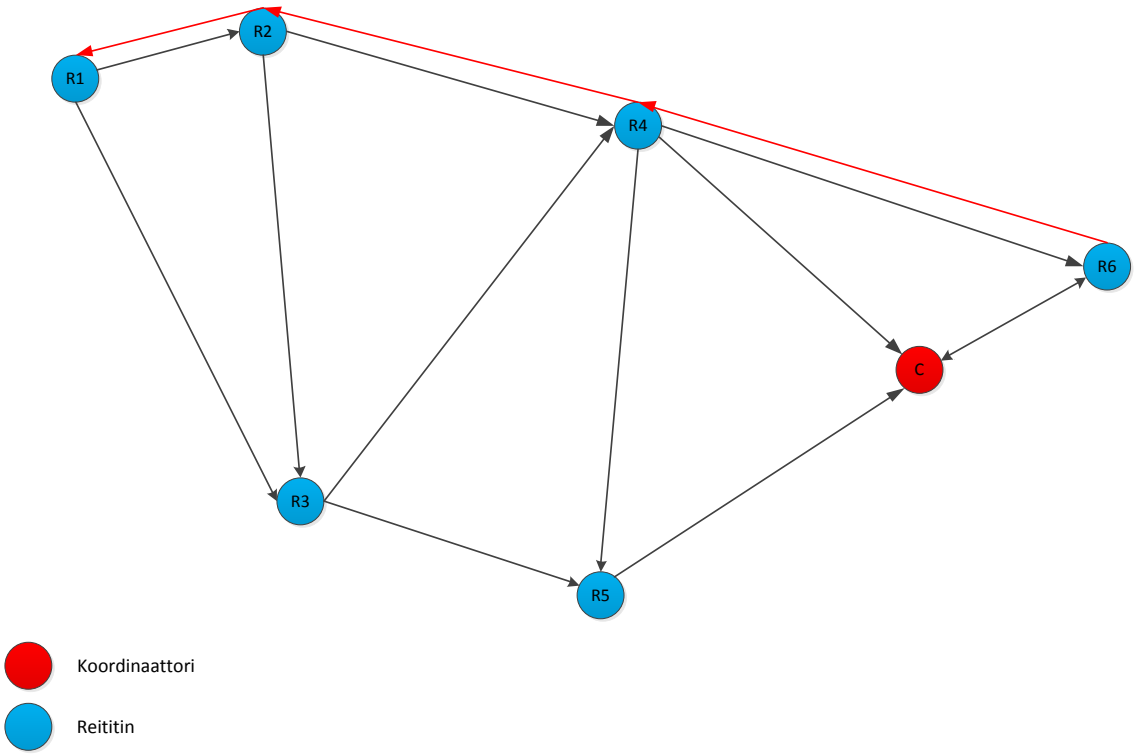
Verkkoa muodostettaessa koordinaattori valitsee käyttämättömän kanavan. Tämän jälkeen se skannaa kaikki kanavat läpi ja lähettää kutsun liittymistä varten. Lopuksi se kuuntelee vastaukset ja kirjaa tulokset. Koordinaattorin käynnistyttyä se antaa muiden laitteiden liittyä siihen sille määritetyn ajan verran.

Reitittimen käynnistyessä sen pitää löytää jokin toinen reititin, joka on jo liittynyt verkkoon, tai verkon koordinaattori. Tätä varten se lähettää broadcast-kutsun koko verkkoon, ja sille vastataan yksilöllisesti unicastilla joko reitittemeltä tai koordinaattorilta, joka sallii verkkoon liittymisen. Tämän jälkeen reititin yrittää liittyä verkkoon tätä kautta.

Loppulaitteet yrittävät liittyä verkkoon samalla tavalla kuin reititin, etsimällä vapaan reitittemen tai koordinaattorin, mikä huolii sen liittymään verkkoon. Ainoa ero on, että loppulaitteet toimivat verkossa lapsilaitteina eli etsivät reitittimistä tai koordinaattorista itselleen vanhempaa huolehtimaan niistä.

Tiedonsiirto

Tiedonsiirrossa käytetään lyhimmän reitin etsimiseen broadcastia. Kuvassa 12 havainnollistetaan, miten verkko toimisi, jos reititin 1 haluaisi viestiä reitittimen 6 kanssa. Aluksi reititin 1 lähettäisi broadcastin koko verkon lävitse. Viestin saavuttua reititin 6 päättelee lyhimmän reitin takaisin. Jatkossa tätä reittiä käytetään oletuksena niiden tiedonsiirtoon, jos verkon rakenne ei muutu.



Kuva 12. Lyhimmän tiedonsiirtokanavan etsintä.

Huomionarvoinen seikka on, että jos dataa lähetettäisiin edellä mainittua reittiä, tulee varmistuksia matkalla paljon, koska mesh-pohjainen node hopping vaatii varmistukset. Aina kun tieto saavuttaa seuraavan solmupisteen, lähettää se varmistuksen sen saapumisesta. Jos tieto kulkee reititin 1:stä reititin 2:een, lähettää se kuittauksen reititin 1:lle ja tiedon eteenpäin reititin 4:lle. Toisin sanoen jos halutaan lähettää tieto reititin 1:stä reititin 6:een ja sieltä kuittaus takaisin reititin 1:een, tulee tiedonsiirtoa joka pykälällä kaksi kappaletta, sekä mennessä että tullessa. Koko reittiin menisi siis 12 kappaletta, 6 mennessä ja 6 tullessa.

Viimeisenä tiedonsiirtoon liittyvänä esimerkkinä voidaan ottaa koordinaattorin läpi tuleva tieto. Jos se pitäisi levittää koko verkkoon, levittäisi koordinaattori sen älykkäänä broadcastina. Se kertoisi tiedon lähimmille reitittimilleen. Nämä taas levittäisivät sitä eteenpäin niiden vieressä oleville solmupisteille jne. [5]

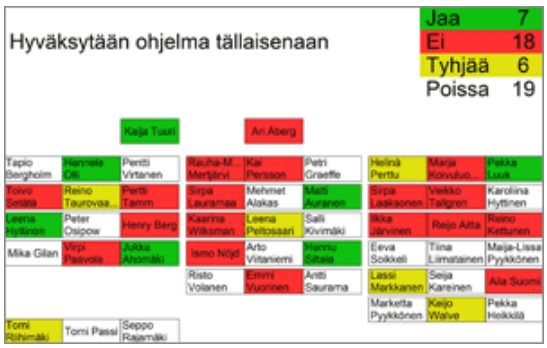
2.7 Sovellukset

Tutkittaessa ja rakennettaessa uutta järjestelmää on hyvä perehtyä jo olemassa oleviin järjestelmiin. Koska työssä luodaan langaton äänestysjärjestelmä Zigbee-pohjaisella mesh-verkoilla, tutustutaan tässä luvussa sekä olemassa oleviin äänestysjärjestelmiin että sovelluksiin, jossa on käytetty Zigbee-pohjaisia mesh-verkkoja, jotta tiedetään, mihin kaikkeen sitä voidaan soveltaa.

Langattomat äänestysnapit

Aluksi selvitetään olemassaolevia langattomia äänestysjärjestelmiä. Ensimmäinen mielenkiintoinen löytyy Replysystemsistä [1]. Reply Minikaukosäädin tukee viittä vastausvaihtoehtoa. Vastauslaite on pienikokoinen ja langaton. Vastaanottimen sanotaan pystyvän tukevan 250:tä äänestyslaitetta. Laitteisto toimii 2.4 GHz:n taajuudella ja toimii FHSS-taajuushyppelyllä. Vastaanotin näyttäisi toimivan tietokonepohjaisesti, koska se vaatii ohjelma-asennuksia. Laitteistoa mainostetaan interaktiiviseen vastaamiseen sopivana. Lähettimen ominaisuuksissa todetaan äänestyksen mukana lähetettävän yksilölliseen tiedon, mutta enempiä mainintoja ei ole, tukeeko loppujärjestelmä tätä yksilöllistä tietojen erittelyä.

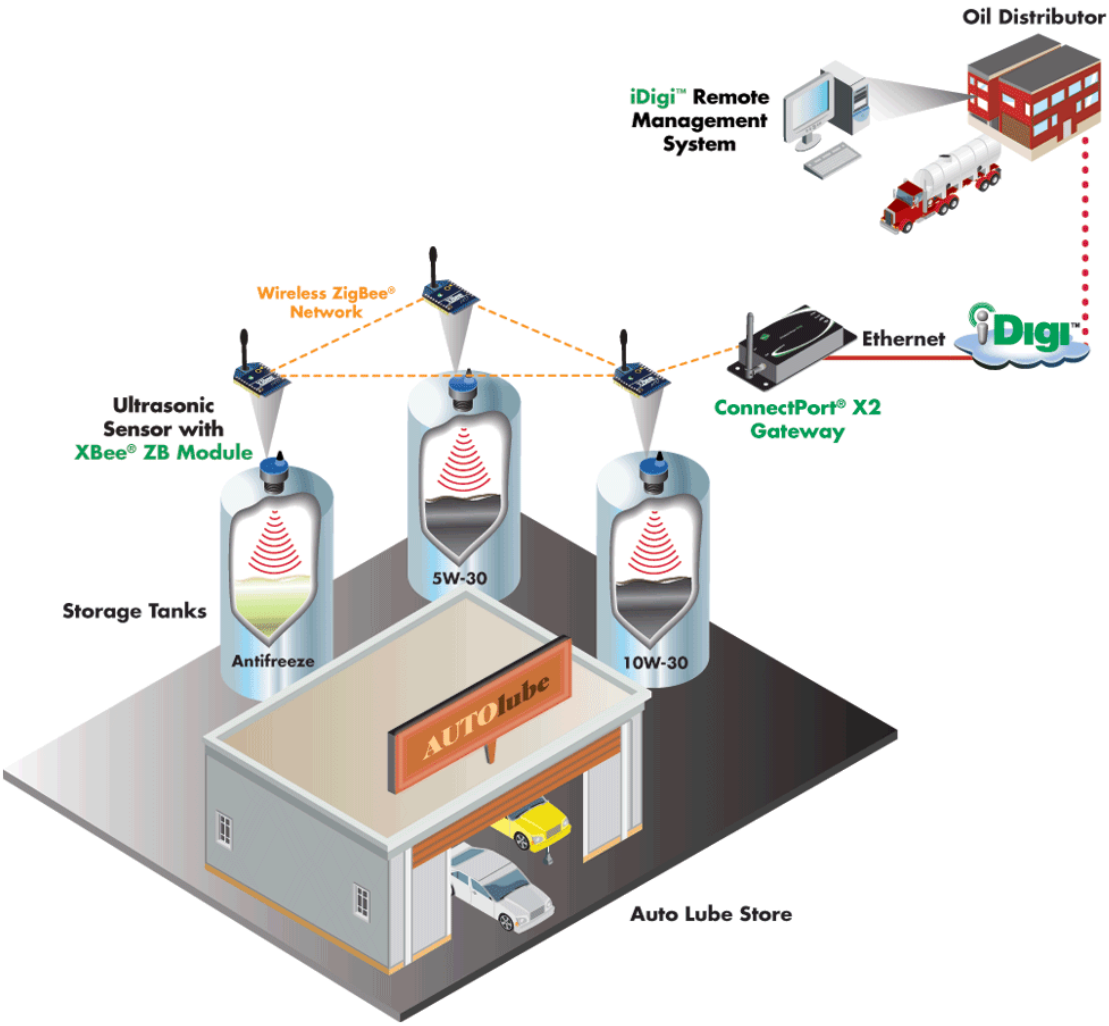
Toinen järjestelmä löytyy suomalaisesta Valtuusto-järjestelmästä kunnanvaltuusto.net-sivustolta [2]. Järjestelmä on myös radiotekniikalla toteutettu, 859.3012 MHz:n taajuudella. Järjestelmän demokuvat tekevät ohjelman mielenkiintoiseksi. Kuvan 13 mukaisesti käyttöliittymä näyttää tukevan suoraan paikka-asettelua ja täysin yksilöllistä äänestystä. Järjestelmän mainostetaan tukevan mm. läsnäolotietojen talteen ottoa.



Kuva 13. Valtuusto järjestelmän kuvanäkymä

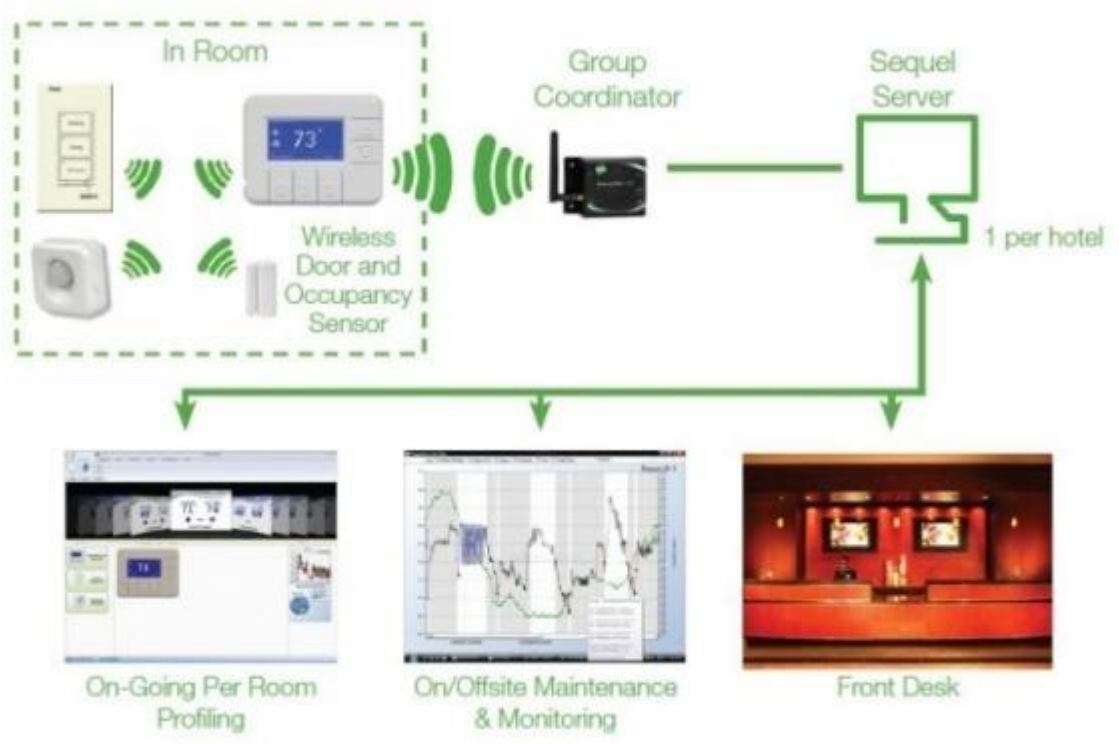
Zigbeen sovellukset

Digin sovellusesimerkkinä löytyy Zigbeellä toimiva kuvan 14 mukainen nestesäiliöiden sisällön valvonta. Järjestelmässä valvotaan tankkien sisältöä, ja niiden pinnan laskiessa järjestelmä ilmoittaa automaattisesti toimittajille täyttöpyynnön. [11]



Kuva 14. Zigbee-pohjainen nestesäiliöiden pinnantason valvonta.

Teollisuuden sovellusten lisäksi toinen nopeasti nouseva kehityshaara on älykodit. Tästä esimerkiksi on otettu Schneider Electricsin hotellipuolelle tuoma huoneiden energianhallintasovellus. Järjestelmän avulla kontrolloidaan isoimpia energiahukkaajia hotellihuoneissa, jotka tulevat valaistuksesta ja HVAC (Heating, Ventilation and Air Condition) –järjestelmistä. Käytännössä järjestelmässä valvotaan huoneissa oloa ovi- ja liiketunnistimien avulla, ja vieraiden muualla ollessa termostaatti- ja valaistuksenohjauksen avulla pidetään huoli, ettei energiaa hukata turhaan. Näitä tietoja voidaan tarkastella ja säätää suoraan serveriltä kuvan 15 mukaisesti. [12]

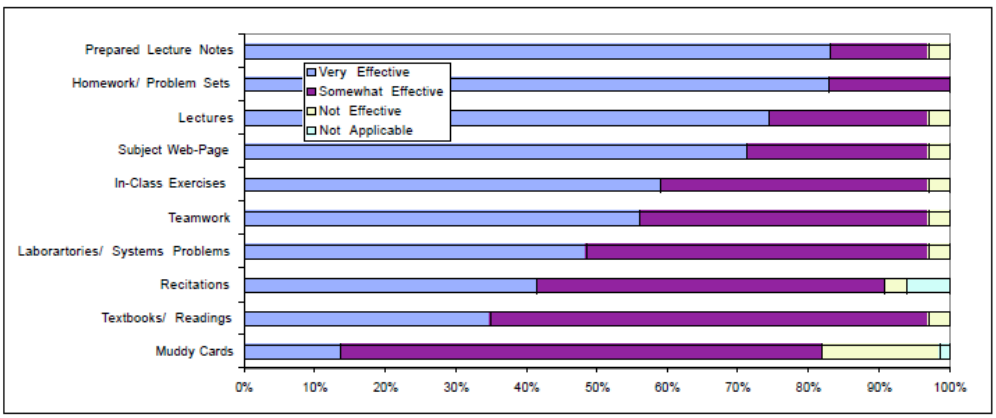


Kuva 15. Hotellihuoneiden energianhallintasovellus.

2.8 Interaktiivinen opiskelu

Interaktiivinen eli vuorovaikutteinen opiskelu on osoittautunut yhdeksi hyödylliseksi opetusmetodiksi. Tutkimusten mukaan opiskelu tehostuu vuorovaikutuksen myötä, ja yhteiskunnan muutosten myötä myös koulujärjestelmä on muutosten edessä. Perinteisen luennoimisen osoittautuessa haastavammaksi opiskelumuodoksi yhä useammalle oppilaalle on myös opetuksen tehostuttava. Ammatillisessa koulutuksessa painotetaan yhä enemmän projektin hallintataitoja, sosiaalisuutta ja dynaamisuutta. Tähän vastaamaan opetus on myös muuttunut dynaamisemmaksi. Opettajien muuttuessa enemmän ohjaajiksi ja kurssien ryhmätyöpainoitteisiksi projekteiksi voidaan silti vieläkin painottaa teorian merkitystä. Mutta vaikka koulumaailma on vastannut muutoksiin, voidaan aina miettiä, miten voitaisiin vielä kehittyä. Yksi osa-alue on myös teorian muuttaminen interaktiivisempaan suuntaan. Isojen ryhmien kanssa tämä voi olla haasteellista, mutta yksilöllisillä äänestimillä ja välittömällä palautteella voidaan tarjota yhtä ratkaisua tähän kehitykseen.

MIT (Massachusetts Institute of Technology) teki aktiivisen oppimisen tutkimuksen erilaisten opetusmetodien avulla. Kuvassa 16 on esitetty sen tuloksia. Tutkimuksessa todettiin pähkinänkuoressa, että vaikka perinteiset oppitunnit ovat oppilaiden mielestä tehokkaita, arvostavat he silti enemmän uusia tarkemmin valmisteltuja tunteja ja ongelmalähtöistä oppimista. Opettajapuolelta taas todettiin, että on vaikeampi muuttaa opetuksen tapaa kuin sen sisältöä. Isoimpina haasteina opetuksen uudistuksessa mainittiin sen huolellinen suunnittelu, toteutus, raaka työ ja vielä satunnainen kädenvääntö varmastimaan uusien opetusmetodien onnistuminen.[13]



Kuva 16. MIT:n erilaisten opetusmenetelmien tutkittuja tehokkuuksia.

3 Testijärjestelmän suunnittelu ja toteutus

3.1 Määritys

Kun älykkäät anturiverkot on määritelty ja käyty läpi sen yleisimmät sovellukset, on aika lähteä rakentamaan omaa sovellusta. Työn tarkoitus on rakentaa oppilas-ympäristöön järjestelmä, jolla oppilaat voivat tuntien aikana vastata erilaisiin kysymyksiin ja äänestyksiin omilla henkilökohtaisilla valitsimillaan. Äänestyslaitteiston on tarkoitus pystyä sisältämään kyseisen lähettimen tiedot, jotta tulokset pystytään taulukoimaan ja laitteet identifioimaan. Järjestelmän avulla pitää pystyä näkemään halutessaan eriteltyä dataa, kuten yhden oppilaan oikeiden vastausten prosentit, tai vaihtoehtoisesti koko luokan vastaukset tiettyyn kysymykseen. Järjestelmän tulee luonnollisesti toimia langattomasti. Työssä rajataan demoa siten, että vastauslaitteilla voi vastata vain monivalintakysymyksiin. Testiä varten suunnitellaan järjestelmä, josta löytyy äänestyslaitteisto, vastaanottimet sekä tiedon prosessointitapa vastausten käsittelyyn ja tuomiseen sekä esitettävään että taulukoitavaan muotoon.

3.2 Laitteisto

Projektia varten on tutkittu erilaisia kehitysalustoja, millä sen voisi toteuttaa. Alkumäärityksenä käytettävälle laitteistolle on löytää mahdollisimman kattava kehitysalusta, mikä tukee mielellään kaikkia projektin määrityksessä olleita asioita. Kartoituksessa löytyy usealta eri valmistajilta Zigbee-kehitysalustoja. Näitä vertailemalla projektin määrityksiin sopii parhaiten Digin Xbee-PRO ZB Programmable Dev Kit- ja iDigi™ Professional Development Kit ZB-kehitysalustat, jotka yhdessä tarjoavat kaikkein kattavimmat ominaisuudet ja laitteistot. Testialustojen avulla voidaan projektiin tehdä kolme äänestyslähetintä, hallinnoida sitä suoraan yksittäiseltä koneelta tai verkon yli ja vahvistaa verkon rakennetta mukana tulevalla seinäreitittimellä. Tämän lisäksi kolme mukana tulevaa moduulia sisältää valmiiksi sisäiset prosessorit. Näiden lisäksi Digi tarjoaa alustoilleen ilmaisen sovelluskehitysserverin.

Kehitysalustojen osat

XBee-PRO ZB Programmable Dev Kit [14]

- XBee-PRO ZB Programmable modules (international variant)
- XBee USB interface board
- XBee RS-232 interface board
- Debugger
- XStick ZB
- Accessories and power supply (including international plug kit)

iDigi™ Professional Development Kit ZB [15]

- Connectport X4 gateway w/ Ethernet
- XBee & XBee-PRO ZB modules
- RS-232 and USB interface boards
- XBee-PRO ZB wall router

Connectport X4

Connectport on työn tärkein yksittäinen laite. Sen tarkoitus on toimia rakennettavan verkon koordinaattorina, ja sen avulla vastaanotetut tiedot prosessoidaan ja siirretään verkkoon.

Tarkemmin eriteltynä Connectportin tehtävänä on tarjota IP-pohjainen verkkoyhteys Zigbee-verkon päätelaitteisiin, kommunikoida niiden kanssa ja kerätä dataa niistä. Kerättävä data pystytään prosessoimaan suoraan jo Connectportissa Python-ohjelmointikielen avulla. Tämä taas tarjoaa mahdollisuuden reagoida suoraan mahdollisiin muutoksiin ja lähettää mittausdataa XML-muotoon tallennettuna serverille. Sieltä data pystytään jatkoprosessoimaan monelle ohjelmalle, esimerkiksi tässä tapauksessa Excelille. Kuvassa 17 on esitelty Connectport.

Kuva 17. Connectport X4 [16]

Xbee PRO S2B -moduulit

Moduuleina työssä käytetään Maxstreamin tekemiä Xbee PRO S2B -moduuleja. Moduuleja on kahta rinnakkaismallia, ja ne ovat antennejaan lukuunottamatta identtisiä kuvan 18 mukaisesti. Ohjelmoitavia malleja testikiteissä on yhteensä kolme kappaletta. Yhtä moduuleista käytetään rakennettavaan demolähettimeen ja kahta muuta kehitysalustojen kanssa verkon demoamiseen.

Kuva 18. Xbee PRO S2B -moduulit [17]

Moduulit ovat yksinkertaistaen piirilevyjä, jotka sisältävät radiolähetyksen ja vastaanoton, ohjelmoitavan mikroprosessorin sekä antennin. Vaikka moduulit eivät ole fyysisiltä mitoiltaan kovinkaan paljoa postimerkkejä isompia, ovat ne silti hyvin kehittyneitä. Yleensä tiedon siirto hoidetaan yhdellä laitteella ja tiedon prosessointi toisella, ja tämän päälle tehdään vielä verkon asemoinnit. Mutta koska moduulien avulla toimitaan Zigbeellä, osaa se hoitaa sekä tiedonsiirron että automaattisen verkoittumisen. Laitteen sisällä oleva mikroprosessori taas pystyy hoitamaan datan käsittelyä suoraan lokaalisti piirillä niin haluttaessa. Tämä tarkoittaa sitä, että laite on todella monipuolinen ja sen sovellusmahdollisuudet ovat hyvin laajoja. Liitteessä 2 sivulla 1 on moduulien tarkat tekniset määritelmät. Huomionarvoinen kohta laitteen määritelmässä on laitteen kantosäde, joka on sisätiloissa 90 metriä ja ulkona jopa 3200 metriä. Nämä ovat teoreettisia maksimiarvoja, joihin vaikuttavat Fresnel zone ja muut tekijät, mutta kantama on erinomainen nykytekniikalle. Laite toimii 2,4 GHz:n taajuudella, joten sitä voidaan käyttää maailmanlaajuistesti. Järjestelmä sisältää myös vakiona salaustekniikat. Viimeisenä mainittakoon virran kulutus, joka on todella pieni. Power-Down-tilassa laite vie sähköä vain 4 mikroampeeria. Käytännössä laitetta pidettäessä nukkumistilassa ja lähetettäessä mittausdataa vaikka muutaman kerran päivässä laite kestänee vuosia paristokäytölläkin.

Moduulit ovat 20-jalkaisia ja liitteessä 2 sivulla 2 on käyty läpi niiden pinnijärjestys. Taulukkoon perehtymällä voidaan nähdä, että myös näistä moduuleista löytyvät kaikki nykyaikaiset komennot, mitä voi odottaa löytävänsä hankkiessaan perusmikroprosessoripiirin. Moduulit tukevat analogisia sisääntuloja ja digitaalisia I/O-väyliä, ja monet pinneistä voidaan määrittää käyttämään mitä vain näistä. Moduuleista löytyy suoraan myös tuki suoralle PWM-lähdölle (Pulse-width modulation), UART-liitäntä (Universal Asynchronous Receiver Transmitter) ja suorat tuet verkon hallinnoinin kannalta hyödyllisille Comission buttonille sekä yhteysledille.

Testialustat

Molempien kehitysalustojen mukana tulee yksi USB- ja yksi RS-232 testialusta, eli niitä on yhteensä neljä kappaletta. Alustat ovat kuvan 19 mukaisia. Tällaiset testialustat ovat suunnitteluvaiheessa välttämättömiä, koska niiden avulla moduulit voidaan ohjelmoida ja seurata niiden luku- ja kirjoitustiloja yhteysledien avulla, ja ne mahdollistavat moduulien sisäisen piirin ohjelmoinnin. Alustoista löytyy myös painonappeja, ledejä yms., mitkä helpottavat moduuleille luotujen ohjelmien testausta. Tässä projektissa alustoja käytetään alkumääritysten tekemiseen, ohjelmointiin sekä osittain verkon testaamiseen.

Kuva 19. Development kit -sarjan mukana tuleva RS-232-testialusta [18]

Xstick

Xstick on USB-pohjainen kuvan 20 mukainen moduuli, joka tarjoaa lyhyen matkan kantamaa Zigbee-verkkoihin. Laitteen avulla voi esim. luoda pc-ohjelman ja käyttää sitä kommunikoimaan verkon kanssa. Laitteella voi myös viestiä verkon laitteiden kanssa yksinkertaisesti terminaalin välityksellä AT-komennoilla. Luonnollisesti valmiissa projektissa käytetään kehittyneempää API-ohjelmointirajapintaa. Xstickin tarkoitus onkin lähinnä auttaa määritysvaiheessa testaamaan laitteiston toimintaa sekä loppuvaiheessa olemaan vianmäärityksen ja testauksen tukena.

Kuva 20. Xstick ZB [19]

Wall Router

Digin seinäreititin otetaan projektiin mukaan tukemaan verkon rakennetta. Seinäreitittimen paras puoli on sen helppokäyttöisyys. Laite on valmiiksi koteloitu kuvan 21 mukaisesti, se toimii tavallisen verkkovirran avulla, ja se on esiohjelmoitu valmiiksi reitittimeksi. Asentamalla tällaisia voidaan verkon kokoa ja käyttösädettä helposti kasvattaa. Laitteen avulla verkko pystyy huolehtimaan useammasta lapsilaitteesta, tarjoaa useampia liittymisväyliä ja siten luotettavampaa verkottumista.

Kuva 21. Xbee-PRO ZB wall router [20]

USB multilink debugger

Debugger on viimeinen työssä tarvittava laitteistopuolen väline. Debuggeria tarvitaan moduulien ohjelmointiin. Laite kytketään tietokoneen USB-liitännän kautta testialustaan. Laitteen avulla voidaan ohjelmoida moduulien mikroprosessoria Codewarrior-ohjelmiston avulla. Kuvassa 22 on esitelty USB multilink debugger.

Kuva 22. Usb Multilink debugger [21]

3.3 Ohjelmisto

Työssä käytettävää eri ohjelmistoa on suhteellisen paljon, koska erillisiä vaiheita on huomattava määrä. Ohjelmistoa kertyy moduulien määritykseen, prosessoreiden ohjelmointiin, Connectportin hallinnointiin, Python-ohjelmointikieleen ja XML-kieleen. Tämän lisäksi työssä käytetään muita perusohjelmia laitteiston käyttöön ja dokumentointiin, niitä sen kummemmin tässä esittelemättä.

X-CTU

X-CTU on MaxStreamin ohjelma Xbee-radiomodeemien ohjelmointiin ja testaukseen. Ohjelman avulla moduulit voidaan mm. ohjelmoida, tarkistaa niiden tyyppi, resetoida ne ja testata niiden toimintaa. Ohjelma sisältää myös integroidun terminaalin, josta on helppo seurata moduulien tiedonvälitystä ja jolla niitä voi ohjata AT-komentojen avulla. Ohjelmassa moduulit on myös helppo ohjelmoida API-järjestelmälle sen graafisen käyttöjärjestelmän avulla. Ohjelma myös auttaa laitteiston toiminnan seuraamista.

Codewarrior

Codewarrior-ohjelmaa käytetäään Xbee PRO S2B -moduulien mikroprosessorien ohjelmointiin. Ohjelmoimalla moduulin sisäistä mikroprosessoria voidaan piirillä suorittaa hyvin paljon asioita, mitkä muuten eivät olisi mahdollista. Tässä projektissa esimerkiksi ohjelmoimalla mikroprosessoria tallennetaan moduuliin suoria vastausvaihtoehtoja, mitkä eivät olisi mahdollisia muulla tavoin.

Digi Device Discover

Digi Device Discoveriä käytetään Connectportin etsimiseen verkosta. Se selvittää laitteiden verkon IP-osoitteet ja submaskit. Tämän jälkeen se listaa löytämänsä laitteet.

Python

Python on yksinkertaisemmasta päästä oleva monipuolinen tulkattava dynaaminen ohjelmointikieli. Pythonista on olemassa useita eri versioita. Tässä projektissa käytetään Pythonin 2.4.3 -kääntäjäversiota, joka on yhteensopiva käytettävän laitteiston kanssa. Sen avulla kerätään ja listataan vastaukset suoraan serverille XML-muodossa.

XML

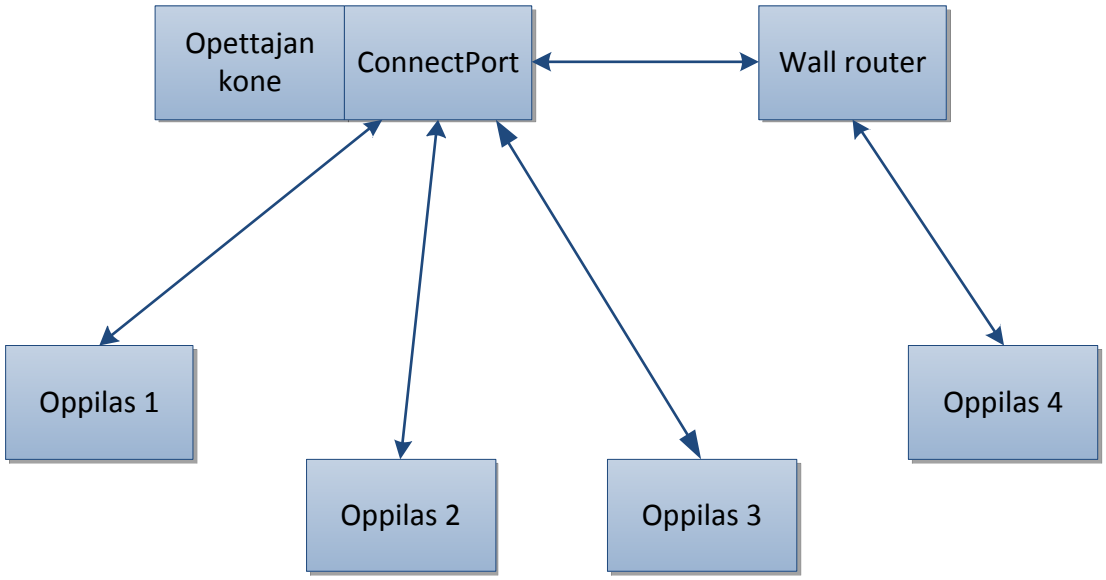
XML-kielen (eXtensible Markup Language) avulla voidaan tallennettavan tiedon sisälle kuvailla myös tietojen merkitystä. Toisin sanoen XML ei ole niinkään puhtaasti oma kielensä, vaan enemmänkin merkintäkieli, joka tuo lisäarvoa tiedon tallentamiseen ja välittämiseen eri ohjelmien välillä. Tässä työssä fyysinen napin painallus muutetaan digitaaliseksi tiedoksi, ja lähetetään radioteitse vastaanottimelle, joka taas tallentaa tämän tiedon XML-taulukkoon sille siellä kuvattuun kohtaan.

3.4 Suunnittelu

Laitteiden valinnan jälkeen työn määritelmää voidaan tarkentaa. Työ fokusoituu neljään kiintopisteeseen: älykkään verkon suunnitteluun, logiikan ohjelmointiin, kaukosäätimien rakentamiseen ja tiedon prosessointiin. Yllä esitettyjen laitteiden avulla kaikki nämä pystytään toteuttamaan. Laitteiston avulla on tarkoitus saada rakennettua testijärjestelmä, jonka avulla neljä opiskeljaa pystyy vastaamaan monivalintakysymyksiin omilla äänestysnapeillaan ja josta vastaustiedot automaattisesti tilastoidaan Exceliin luettavaan muotoon.

Älykäs verkko

Älykkään mesh-verkon suunnittelussa pitää aluksi miettiä kaikkien laitteiden funktio. Koska äänestyslaitteet ovat lähetykseen tarkoitettuja laitteita, ne nimetään päätelaitteiksi eli vanhempi – lapsi-laitteista lapsiksi X-CTU-ohjelman avulla. Lasten ei tarvitse viestiä keskenään, vaan ne viestivät vanhemmille. Kuvassa 23 on havainnollistettu järjestelmän toimintaa. Äänestyslaitteet ovat siis aina lapsilaitteita ja verkon muut laitteet vanhempia eli reitittimiä ja koordinaattori. Jokainen vanhempi pystyy huolehtimaan maksimissaan 12 lapsesta, joten tästä syystä vanhempia on syytä olla useampia jo valmiiksi, jos äänestysnappeja halutaan lisätä. Toinen syy useammalle vanhemmalle on se, että jos äänestyslähetin ei jostain syystä löytäisi opettajan konetta ollessaan esim. liian kaukana tai jonkin esteen takana, voi se kiinnittyä johonkin toiseen vanhempaan, joka sitten välittää sen tiedot opettajan koneelle. Opettajan on tarkoitus hallinnoida järjestelmää joko Xstickillä suoraan koneeltaan tai sitten Connectportilla verkon yli. Xstick on kuitenkin tarkoitettu enemmänkin laitteiden etsimiseen ja päivittämiseen sekä vian hallintaan, kun taas Connectportin avulla datan keräys ja varsinainen prosessointi onnistuu paremmin.



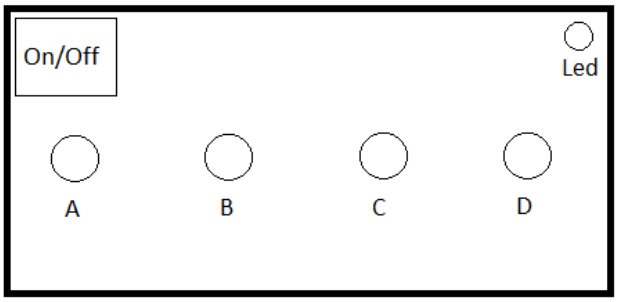
Kuva 23. Laitteiston kaaviomallinnus

Logiikan ohjelmointi

Logiikan ohjelmoinnilla tarkoitetaan tässä tapauksessa lähettimien sisällä olevien mikroprosessoreiden ohjelmointia, jolla saadaan toteuttua yksilölliset äänestysvalinnat. Tätä varten piirit ohjelmoidaan CodeWarrior-ohjelmointialustalla USB multilink debuggeria käyttäen.

Äänestyslähetin

Äänestyslähetin rakennetaan aluksi kytkentäalustalle testattavaksi, minkä jälkeen sille tehdään oma piirilevy. Fyysisesti äänestyslähettimestä pitäisi löytyä kuvan 24 mukaisesti neljä valintanäppäintä, yhteysledi ja on/off-kytkin. Luonnollisesti laitteen sisälle pitää vielä mahduttaa myös paristo sekä paikka antennille. Fyysisiltä mitoiltaan laitteen olisi tarkoitus olla maksimissaan kokoluokkaa 100 x 50 x 30 mm LxKxS. Luonnollisesti piirilevyn koko vaikuttaa tähän. Ulkokuori tehdään pleksistä demoversioon.



Kuva 24. Äänestyslähettimen hahmotelma

Tiedon prosessointi

Viimeinen vaihe järjestelmän kuntoon saattamiseksi on äänestyslähettimiltä saatavan tiedon prosessointi. Kun oppilaille esitetään kysymys ja he vastaavat äänestyslähettimien avulla, on tieto saatava lajiteltua. Ensinnäkin kuka vastasi ja mitä vastasi. Tämän jälkeen tiedot ovat saatava taulukoitua talteen luettavaan ja lajiteltavaan muotoon. Tässä avuksi tulee Connectport, joka tukee Python-ohjelmointikieltä, jonka avulla sisään tullut tieto pystytään prosessoimaan XML-tiedostoksi serverille, jota Excel ymmärtää ja jossa se voidaan taulukoida ja esittää. Laitteistoa on tarkoitus ohjata verkon yli IP-pohjaisesti.

3.5 Työn kulku

Työstä tekee hyvin haastavan sen tekninen vaikeus ja moninaisuus. Kun työssä yhdistyvät langaton radiotekniikka, tiedonsiirtoprotokollat, ohjelmointi useammalle kielellä eri alustoille, elektroniikan rakentaminen, tiedon prosessointi, sekä näiden kaikkien yhteensovitus, on todennäköisesti ongelmia edessä. Tästä johtuen loogisin tapa lähteä liikkeelle on murtaa kokonaisuus loogisiin osiin suunnittelun mallin mukaisesti: verkon määritys, logiikan ohjelmointi, kaukosäätimien rakentaminen ja tiedon vastaanoton prosessointi sekä lopuksi kaikkien näiden yhteen sovittaminen.

Verkon määritys keskittyy verkon toimilaitteiden nimeämiseen, niiden asemointiin verkossa, moduulien ohjelmointiin tietyiksi toimilaitteiksi, tiedonsiirron tapojen valintaan ja tietoturvan miettimiseen. Tämän jälkeen laitteisto testataan siten, että laitteiden pitää löytää toisensa ja muodostaa toimiva halutunlainen verkko. Tämä muodostaa työn pohjan ja rungon, jonka päälle rakennetaan kaikki muu.

Logiikan ohjelmoinnilla määritellään piireille halutut toiminnot. Toimivan verkon jälkeen tämä osio on enemmänkin yhteydessä äänestyslähettimeen, joka antaa reunaehtoja halutulle logiikalle. Käytännössä logiikalla pitää määritellä halutut toiminnot äänestyslähetintä varten.

Äänestyslähetin itsessään on enimmäkseen puhdasta elektroniikkaa ja mekaniikkaa. Koska halutut toiminnot on jo määritelty, on tätä varten suunniteltava piirilevy ja rakennettava se. Periaatteessa tämän pitäisi olla helpoin osa työtä, mutta teorian muuttaminen konkreettisesti toimivaksi laitteeksi saattaa osoittautua helposti hyvinkin työlääksi.

Tiedon prosessoinnissa täytyy yhdistää datavirrat yhteen, kerätä niistä haluttu tieto ja prosessoida se järkevään muotoon. Tämän jälkeen se pitäisi saada tallennettua serverille, josta se voidaan helposti lukea mille tahansa yksittäiselle koneelle ja ohjelmalle.

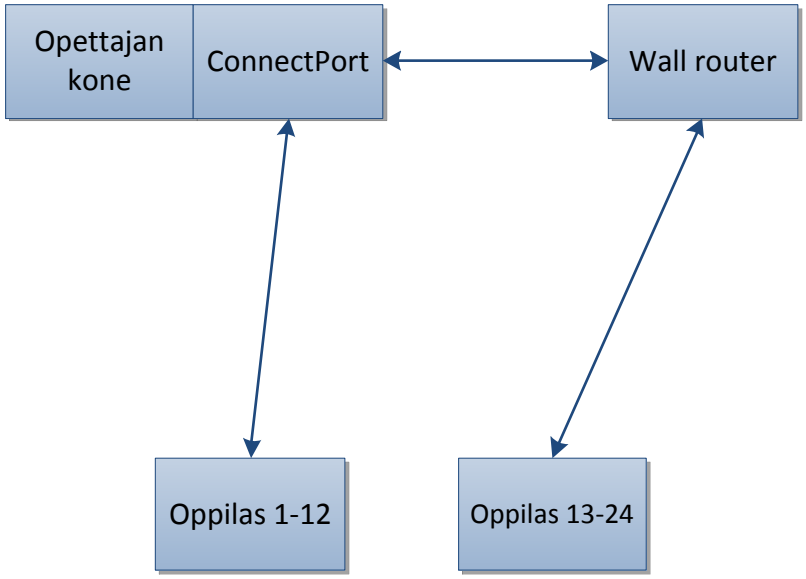
Lopuksi on tarkoitus saada kaikki nämä välivaiheet toimimaan saumattomasti yhteen. Tavoitteena on tehdä loppukäyttäjän kannalta mahdollisimman yksinkertainen ja ehyt toimiva kokonaisuus.

3.5.1 Verkon määritys

Työ alkaa verkon määrityksellä, joka laittaa reunaehdot kaikelle muulle. Päällimmäinen tarkoitus on saada laitteisto, jonka avulla 20 oppilasta voi opetustuntien aikana vastata kysymyksiin reaaliajassa langattomien yksilöllisten kaukosäätimiensä avulla. Tätä varten tarvitaan siis minimissään äänestyslaitteet ja tapa, jolla nämä tiedot voidaan kerätä. Demo tehdään kolmella lähettimellä, kuitenkin siten suunnitellen, että verkkoa pystyy laajentamaan tarpeen vaatiessa helposti.

Kun tätä lähtökohtaa lähetään murtamaan, on aluksi valittava laitteisto. Kuten yllä jo todettiin, laitteistoksi valittiin ZigBee sen ominaisuuksien vuoksi. Tällä tekniikalla jokaisesta lähettimestä saa tehtyä puhtaasti yksilöllisen omien tunnistetietojensa avulla, ja laitteet osaavat liittyä osaksi verkkoa häiritsemättä toisiaan. ZigBeen ongelma tai ominaisuus on, että yksi aikuinen eli isäntälaite pystyy huolehtimaan maksimissaan 12

lapsilaitteesta. Toisin sanoen yhdellä vastaanottimella projektin reunaehtoja ei pystyisi toteuttamaan. Tässä työssä erilaisia isäntälaitteita on yhteensä kolme kappaletta, eli teoreettinen maksimi on 36 oppilaan luokka. Xstick, joka pystyy toimimaan aikuisena, on kuitenkin valittu lähinnä opettajan työkaluksi varmistamaan verkon toimintaa ongelmatapauksessa, eli laitteisto suunnitellaan 24:lle lähettimelle, vaikka isompikaan määrä ei olisi ongelma. Käytännössä verkko tulee siis olemaan kuvan 25 mukainen, jossa opettajan koneella on käytössä Xstick, jota käytetään verkon testaamiseen ja ongelmien havainnointiin, wall router tukee verkon rakennetta ja tarjoaa lisää liitäntä mahdollisuuksia, Connectport taas kytketään reitittimen taakse IP-pohjaiseksi liitännäksi verkkoon. Nämä kaksi aktiivista aikuista huolehtivat niille ilmoittautuvista lapsista eli oppilaslähettimistä 1-24. Molempien maksimikapasiteetti on siis kaksitoista, josta tulee maksimiksi 24. Kuvassa 25 havainnollistetaan verkon toimintaperiaatetta.



Kuva 25. Verkon määritys

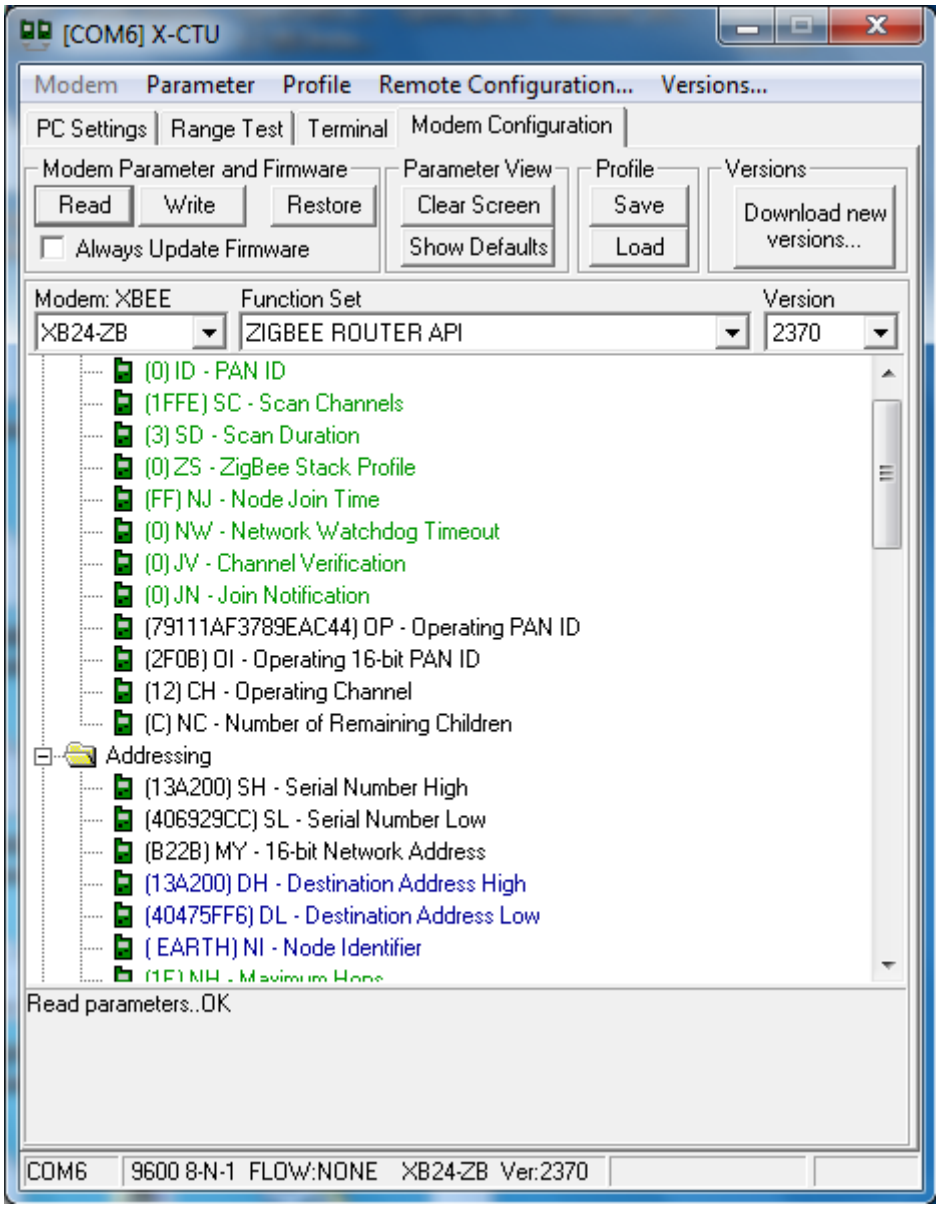
Laitteiston tultua Digiltä pystyttiin laitteiston rakentaminen aloittamaan. Laitteistopuolella kaikki oli hyvin selkeää ja kuten odotettiin, Connectportin virtalähteen johdon ja yhden antennin puuttumista lukuun ottamatta. Moduulit oli helppo liittää testialustoihin, ja mukana tulevat antennit asettuivat helposti paikoilleen. Testialustojen virtalähteet eivät olleet eurooppalaisittain yhteensopivia, mikä aiheutti hetken hämmennystä, mutta laitteiston mukana tuli myös vaihdettavat verkkoliittimet.

Wall router lienee laitteistosta helpoin, koska se kytketään suoraan itsenäisesti pistorasiaan. Xstickin pystyy taas laittamaan suoraan USB-paikkaan. Connectport kytketään reitittimen taakse.

Valitettavasti ohjelmistopuolella asiat eivät olleet näin hyvin. Laitteiston mukana ei tullut ohjelmistoa tai ohjeita. Tästä johtuen aloittaminen oli hyvin vaikeaa. Digin puolustuksena on todettava, että suunnilleen kaikki tarvittava löytyy sen sivustoilta. Vastapainona taas sanottakoon, että sivusto on sekavan oloinen, varsinkin jos ei ole tarkkaa tietoa mitä etsiä. Tästä johtuen kesti aikansa löytää tarvittavat lähtötiedot.

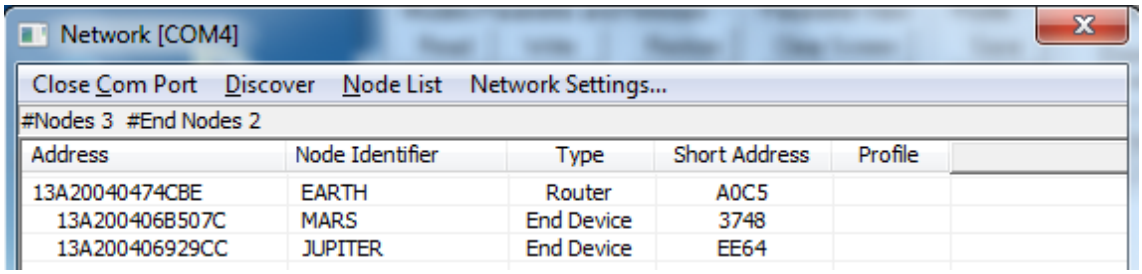
Ohjeiden ja ohjelmien löydyttyä pystyttiin aloittamaan konkreettinen määritys. Tarkoitus oli aluksi saada Xstick löytämään testialustat. Ohjelma, jota tätä varten käytetään, on X-CTU. Windowsin pitää saada laite tunnistamaan testialustat, jotka ovat USB- tai sarjaliitännällä. X-CTU tarjoaa COM-porttia, mistä kyseinen laite löytyy. Kun laitteen saa löytymään, pitää se määritellä. Määrittelyyn sisältyy hyvin paljon vaihtoehtoja. Laitteen toimintakuntoon saattamiseksi on vähintään määriteltävä sen positio verkossa, tiedon siirtämistapa, liittymisasetukset ja tunnistetiedot. Yhdistelmä, miten olla varma COM-portin toimivuudesta laitteen määrityksen kanssa, on alussa haastavaa.

Toimivaan kokoonpanoon päästiin, kun Xstick määriteltiin koordinaattoriksi ja kaikki tiedonsiirto sallittiin. Kuvassa 26 on esitetty Xstickin määritykset.



Kuva 26. Xstickin määritys X-CTU-ohjelmassa

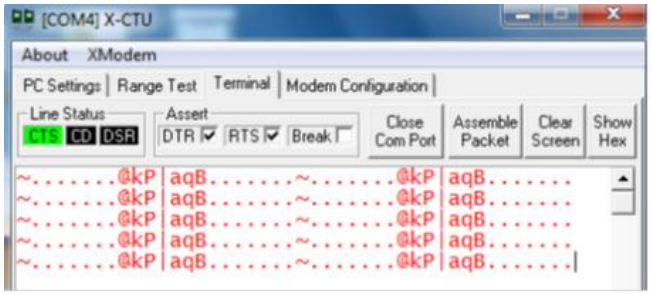
Tämän jälkeen kaksi testialustaa ja moduulia määriteltiin end deviceksi eli päätelaitteiksi. Kaikki laitteet määriteltiin käyttämään API-verkkoa. Xstick nimettiin verkkoon nimellä Earth. Testialustojen moduulit nimettiin nimellä Marsiksi ja Jupiteriksi. Loppujen lopuksi Xstick suostui löytämään ne kuvan 27 mukaisesti.



Kuva 27. Xstickillä löydetyt verkon laitteet.

Verkko pitää ymmärtää konfiguroida käyttämään API:a. Tämä oli yksi vaikeista asioista hahmottaa. Onnistunut tiedonsiirto näkyy helpoiten terminaalitilassa, jota myös X-CTU tukee suoraan ja joka on AT-pohjaista tekstitietoa. API taas tukee tiedonsiirrossa pakettien kehyksiä ja laitteiston ohjelmointia ilman tekstikomentoja. Kaikki laitteet ovat kuitenkin syytä määritellä API-verkkoon, josta pystyy suoraan graafisesti tekemään asetukset. Terminaalin kautta voi tässäkin muodossa ajaa datan siirtoa läpi sitä tarkkailen.

Testialustoihin ohjelmoitiin automaattinen tiedonlähetys sekunnin välein ja tunnistetietojen lähetys napin painalluksella. Tätä testattiin terminaalissa, josta pystyi suoraan reaaliaikaisesti seuraamaan Marsin lähettämiä tietoja kuvan 28 mukaisesti.



Kuva 28. Marsin lähettämää dataa Xstickin terminaali -ikkunassa

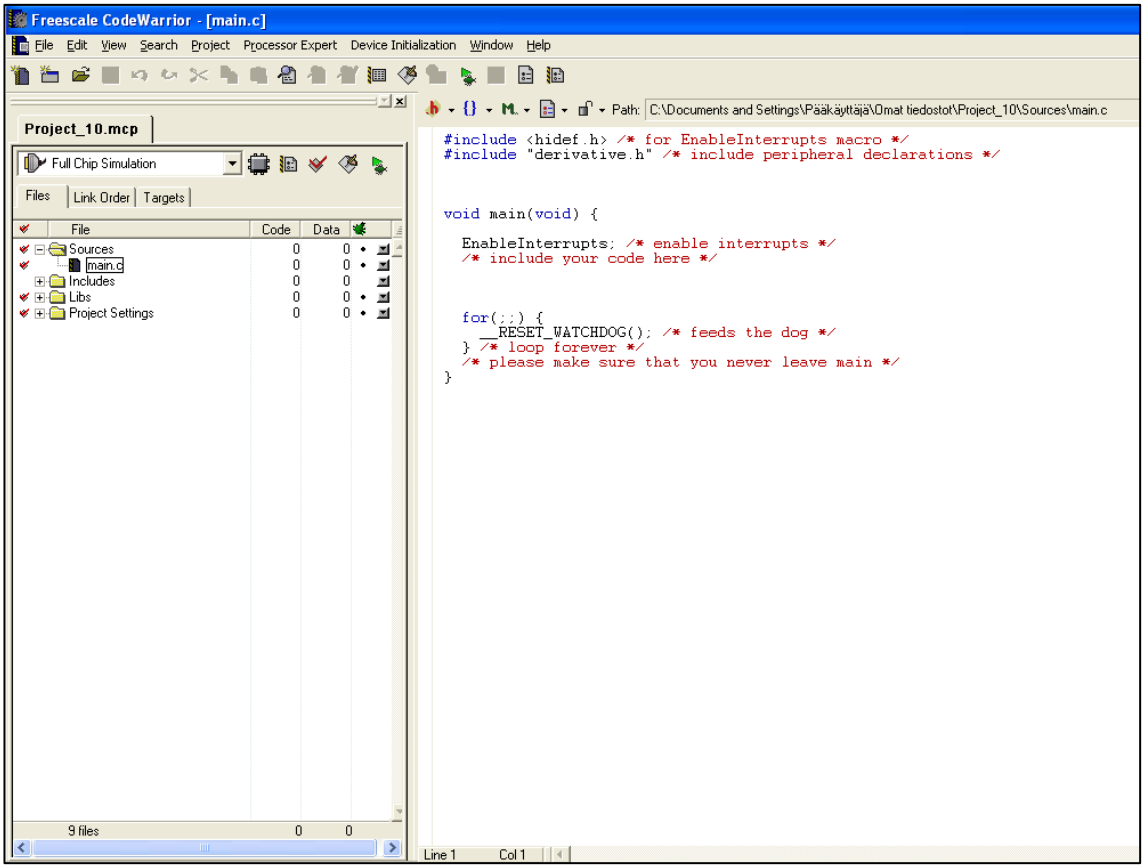
Kun prosessi oli saatu toimimaan kahdella lähettimellä, oli helppo määritellä myös viimeinen moduuli. Wall router ja Connectport ovat oletuksena jo aikuisia, joten niitä ei tarvitse vielä tässä vaiheessa projektia ottaa mukaan.

3.5.2 Prosessorien ohjelmointi

Xbee-moduuleita saa sekä sisäisellä mikroprosessoreilla että ilman. Tämä tarkoittaa, että perusmallilla ilman prosessoria pystyy tekemään lähinnä perustason tiedonsiirto-operaatioita, kun taas prosessorilla olevalla mallilla voi suorittaa paljon monimutkaisempia toimintoja. Prosessorin avulla moduuliin voi suoraan ohjelmoida haluttuja toimintoja ja prosessointia jo moduuliin päässä.

Prosessoreihin ohjelmoitavat komennot voivat olla käytännössä hyvin moninaisia, tietoa voidaan prosessoida, sisääntuloista voidaan ottaa esim. lämpötiladataa oikeiden antureiden kanssa ja ulostuloihin voidaan ohjelmoida vaikka valo- tai moottoriohjausta. Tässä projektissa ohelmoitavuudella keskitytään kaukosäätimen logiikkaan, joka on kohtuullisen yksinkertainen. Kaukosäätimen tarkoituksena on antaa käyttäjän vastata eri vaihtoehdoilla esitettyihin kysymyksiin. Laitteistolla pystyisi luonnollisesti lähettämään monimutkaisempaakin dataa. Määrityksenä kaukosäätimelle on pystyä vastata neljällä eri vastauksella, joten prosessoriin ohjelmoidaan neljä sisääntuloa näppäimille. Näistä näppäimistä sitten lähetetään tiedot vastaanottavalla aikuiselle muodossa Vastaus A, Vastaus B, Vastaus C ja Vastaus D.

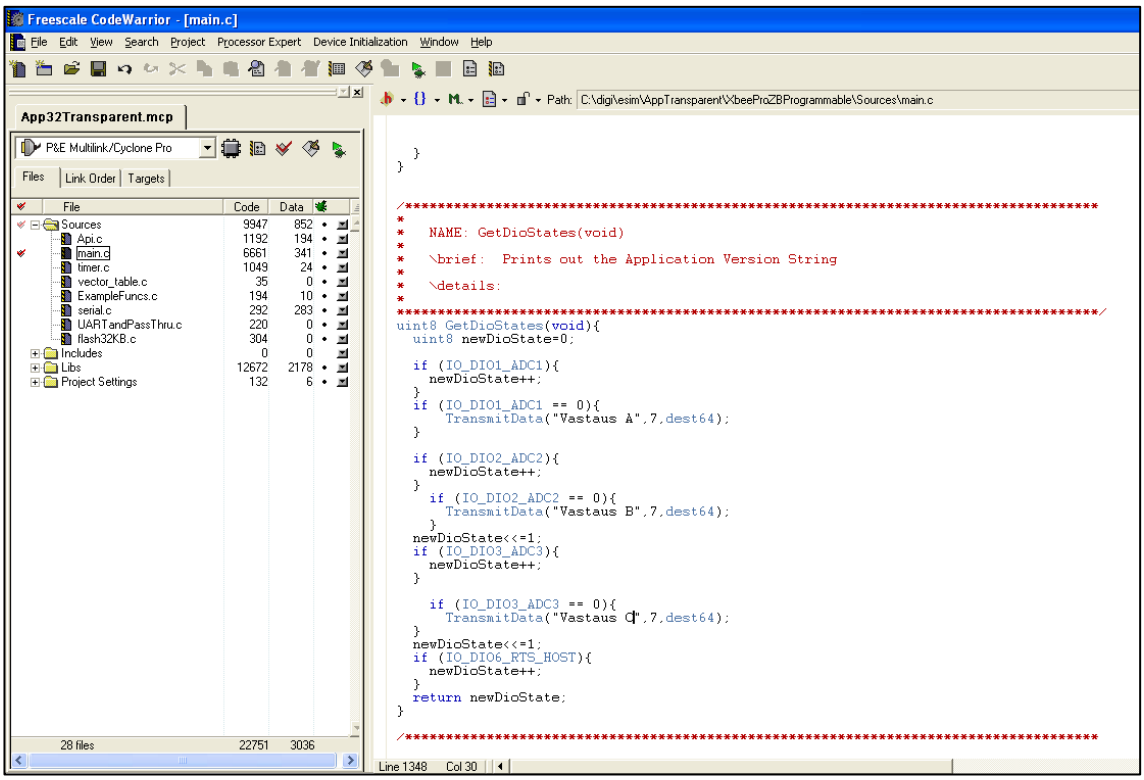
Prosessoreiden ohjelmointi tapahtuu USB multilink debugger -laitteen ja CodeWarrior-ohjelmiston avulla. Kuvassa 29 on esitelty Codewarrior-ohjelma. Tässäkin vaiheessa laitteistopuoli oli heti kunnossa, mutta ohjelmiston ja ohjeiden kanssa kesti aikansa, ennen kuin pääsi selville systeemistä. Ongelmia tuli CodeWarriorin yhteensopimattomuudesta Windows 7:n kanssa, käytännössä ohjelmisto oli pakko asentaa koneelle, josta löytyi XP.



Kuva 29. CodeWarrior-ohjelma

Itse ohjelmoinnissa Digi on lukinnut laitteestaan Rf:n ohjelmointilogiikan, joten sitä ei pysty sekoittamaan huonolla ohjelmoinnilla. Digi on myös rakentanut esimerkkiohjelman demoamaan varsinkin terminaalipuolen käytettävyyttä. Tähän valmiiseen ohjelmaan pystyy lähteä rakentamaan omaa sovellustaan. Koodirivien määrä pelkästään Rf:n ja tiedon lähetykseen, sen vaikeudesta puhumattakaan, on toisaalta sitä luokkaa, että ilman jotain pohjaa ohjelmointi olisi käytännössä mahdottomuus.

Neljän näppäimen nimeäminen haluttuihin sisääntuloihin luettavaksi ja niiden perusteella lähetettävän tiedon ohjelmointi oli suhteellisen hankalaa, mutta toteutettavissa. Kuvassa 30 on esimerkki näppäimistöstä lähetettävästä datasta. Myös tietojen ohjelmointi moduuleille otti aikansa USB multilink debugger asetusten kanssa, mutta onnistui lopuksi kuitenkin.



Kuva 30. Lähettimien näppäimistön ohjelmointi

Tässäkin vaiheessa helpoitti se, että laitteet ohjelmoitiin samalla tavalla, toisin sanoen ensimmäisen ohjelmoinnin jälkeen muut laitteet ohjelmoitiin nopeasti. Jokainen kaukosäädin nimettiin X-CTU:ssa erikseen, joten CodeWarriorissa kaikkiin moduuleihin tuli sama koodi.

3.5.3 Kaukosäätimen rakentaminen

Kaukosäätimen rakennuksessa tehdään yksi demoversio. Tämä pitää sisällään piirilevyn ja kotelon. Koelevy rakennetaan kuparinauhalevylle, josta siitä on helppo suunnitella myöhemmässä vaiheessa ammattiohjelmilla PCB-tiedosto useampien levyjen tekemiseen.

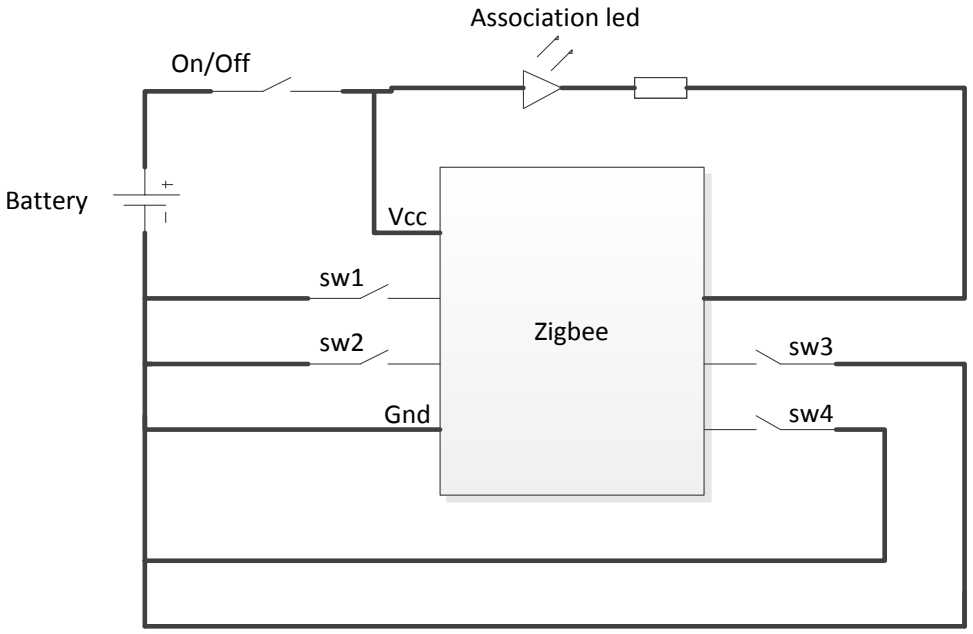
Rakennuksen idea on tehdä mahdollisimman pieni ja käytännöllinen kaukosäädin, jolla oppilas voi valita vastauksensa esitettyyn kysymykseen. Kaukosäätimen tulee sisältää neljä valintanappia, on/off-kytkin ja verkon toimintaa kuvaava ledi.

Suunnittelu

Suunnitteluvaiheessa työssä piti hahmotella omaan piirilevyyn tarvittavat komponentit moduulien datalehtien avulla, yrittää optimoida laitteen koko mahdollisimman pieneksi ja miettiä, miten kotelo kannattaisi rakentaa.

Työ alkoi liitteen 2 pinnijärjestyksen avulla. Kuvaa tutkimalla voidaan todeta, että minimikytkennät vaatimusten täyttämiseen on neljän I/O-väylän käyttäminen, käyttöjännite- ja maakytkennät sekä associate toiminnon käyttäminen pinni 15 ulostulosta (verkon toimintaa kuvaavaa lediä varten).

Kytkennästä itsestään ei tullut mikään monimutkainen vähäisten sisään- ja ulostulojen takia. Kuvassa 31 on laitteen piirikaavio.



Kuva 31. Kaukosäätimen piirikaavio

Piirikaavion suunnittelun jälkeen piti miettiä osasijoittelu piirilevylle. Tarkoituksena on tehdä mahdollisimman käyttäjäystävällinen laite, pieneen kokoon pakattuna. Kokona

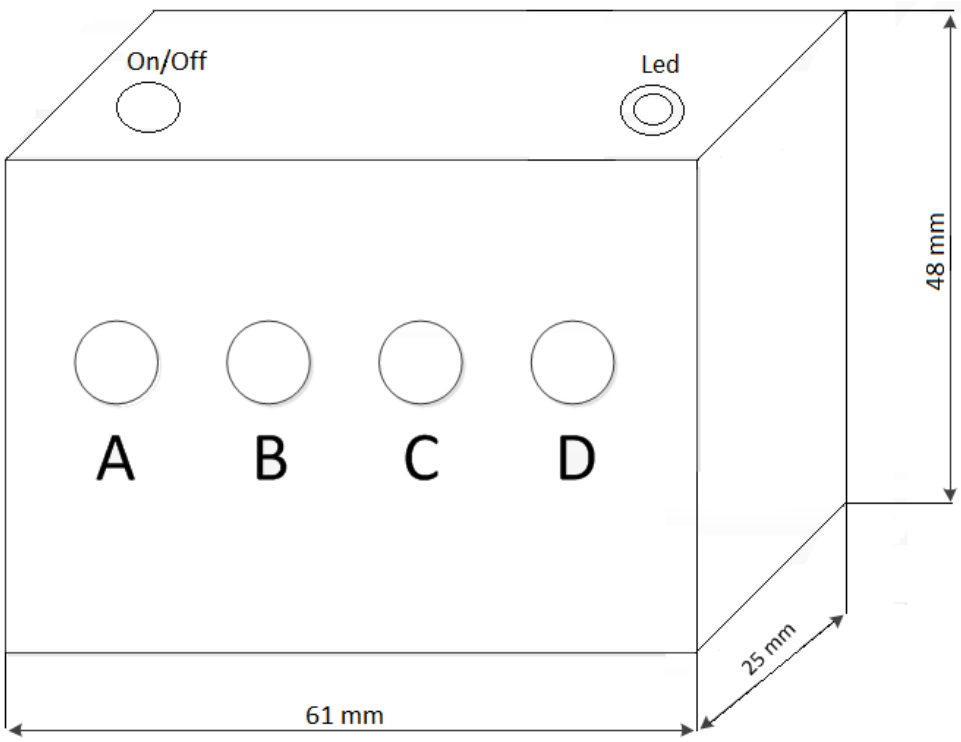
Xbee PRO S2B -moduuli itsessään on hämmästyttävän pieni ottaen huomioon sen monipuolisuuden, ainoastaan 24,38 x 32,94 millimetriä leveä ja pitkä.

Osasijoittelun isoimmat ongelmat tulivat pariston valinnasta ja moduulin rajoituksista. Pariston valinnassa painotettiin pientä kokoa. Tästä ja piirin vähävirtaisuudesta johtuen päädyttiin hieman normaalista poikkeavaan ratkaisuun, piirilevylle kiinteästi juotettavaan pieneen virtalähteeseen. Moduulissa ongelmia aiheutti ensimmäiseksi epästandardi pinniväli, joka ei suoraan sopinut piirilevylle, vaan vaati sovitekappaleita. Toiseksi haasteeksi tuli moduulin päätyosa, jossa on antenni. Tästä johtuen alueella ei saisi olla lähellä mitään häiriöitä aiheuttavia tekijöitä, joten tämä seikka oli otettava suunnittelussa huomioon.

Piirikaavion ja osasijoittelun jälkeen pystyttiin suunnittelemaan kotelo. Kotelon etulevyyn tulee neljä painonäppäintä näkyviin, mistä käyttäjä voi valita halutun vaihtoehdon. Kotelon yläreunaan tulee virtakytkin sekä led-yhteysvalo, mikä kertoo, että piiri on yhdistynyt verkkoon. Kotelosta tehdään muutama millimetri isompi kuin mitä piirilevy on.

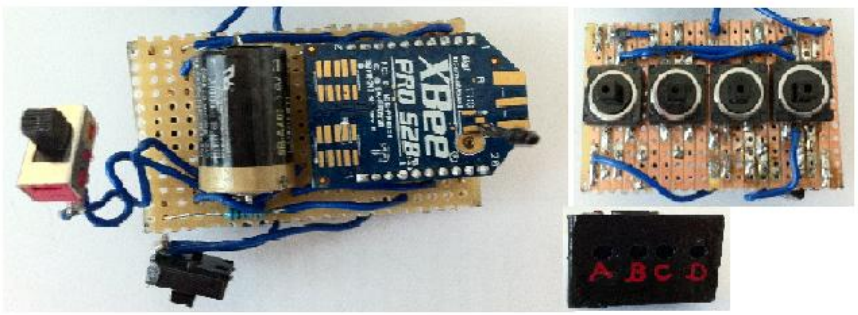
Rakennus

Rakennusvaiheessa osat sijoitettiin piirilevylle, kolvattiin kiinni, tehtiin tarvittavat katkokset ja vedettiin tarvittavat hyppylangat. Piirilevyn pienen fyysisen koon takia tämä vaihe osoittautui haastavaksi, ja piirilevyn molempia puolia jouduttiin käyttämään tehokkaaksi. Toinen haaste tässä vaiheessa oli moduulin pinnien sovittimien asentaminen. Nämä on käytännössä pakko olla, koska moduulin pinnijärjestys oli epästandardi ja koska moduuli haluttiin pystyä ottamaan tarvittaessa irti piiristä uudelleenohjelmointia varten. Kaukosäätimen kotelo päätettiin rakentaa pleksistä sen helpon muokattavuuden ansiosta. Kotelo tehtiin kuvan 32 mittojen avulla. Koteloon porattiin kytkimille ja ledille reiät, ja levyt yhdistettiin liimalla takakantta lukuunottamatta, joka tehtiin saranoilla, jotta moduuli voitaisiin halutessa irrottaa sieltä helposti.



Kuva 32. Kaukosäätimen kotelon mitat

Työvaiheena nämä osuudet olivat yllättävän työläät ja aikaa vievät. Lopputuloksena saatiin aikaiseksi toimiva kaukosäädin. Luonnollisesti parempaan tulokseen päästäisiin oikeilla piirilevyillä ja tehdastuotetulla kotelolla, mutta demovaiheeseen laite ajaa asiansa hyvin. Kuvassa 33 on esitelty valmis piirilevy ja kotelo.

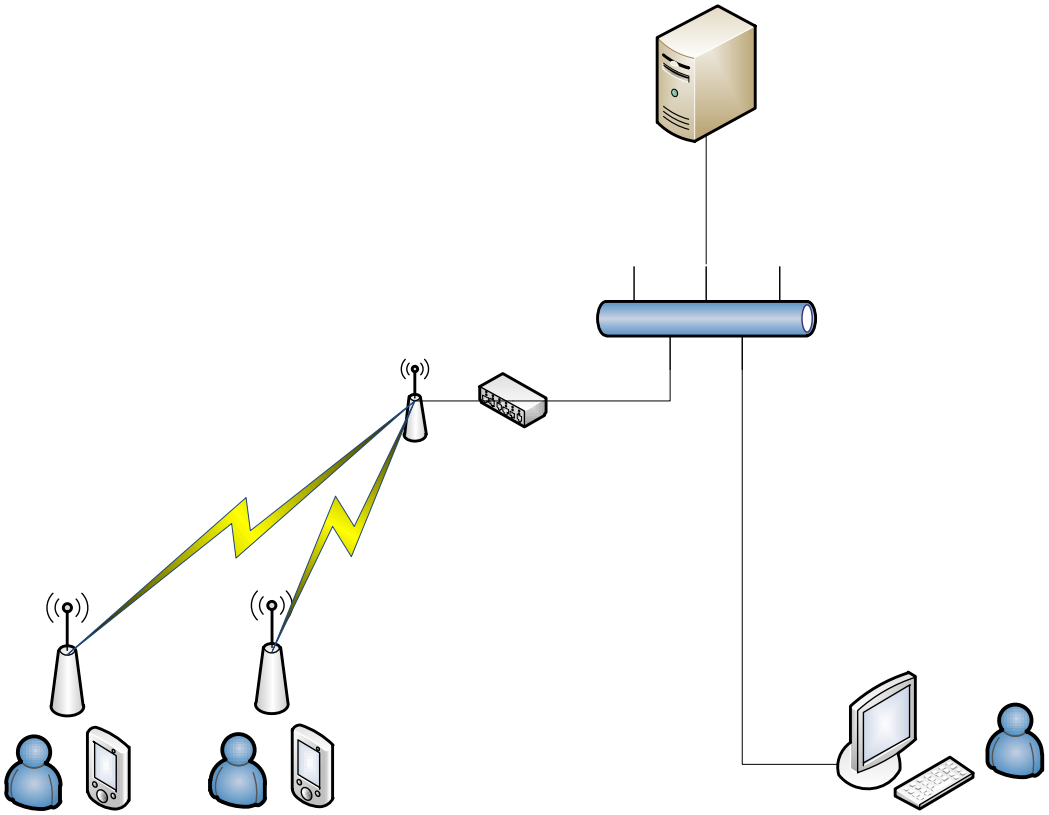


Kuva 33. Kuvia kaukosäätimen piirilevystä sekä kotelosta

3.5.4 Tiedon prosessointi

Laitteiston ja verkon määrityksen, moduulien ohjelmoinnin ja kaukosäätimen rakentamisen jälkeen viimeisenä tavoitteena oli saada kerättyä tiedot talteen. Tätä varten käytetään Connectportia.

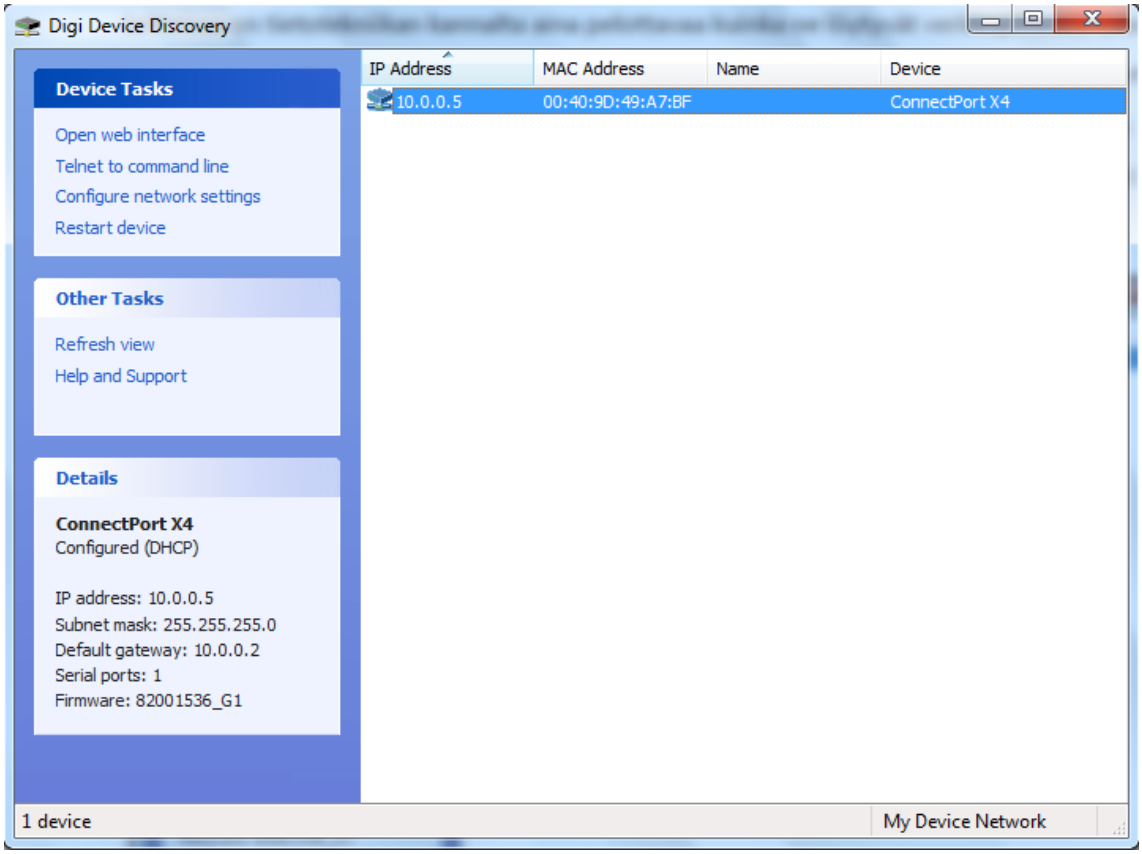
Tiedot on tarkoitus saada kerättyä helposti nähtäväksi sekä listatuksi järkevällä tavalla. Tätä tukee Digin Connectport monellakin tavalla. Laite valittiin, koska se tukee Python-ohjelmointikieltä ja sitä pystytään kontrolloimaan IP-pohjaisesti verkon yli. Näiden ominaisuuksien avulla on mahdollista suunnitella järjestelmä, mitä voi ohjata selainpohjaisesti verkon yli millä koneella vain, josta tiedot voidaan Pythonin avulla kerätä ja tallentaa XML-muotoon serverille, joka taas mahdollistaa tiedon loppukäsittelyn millä koneella vain, missä on internetyhteys ja Excel. Kuva 34 havainnollistaa tätä tiedonsiirtoa



Kuva 34. Tiedon virtaus järjestelmässä.

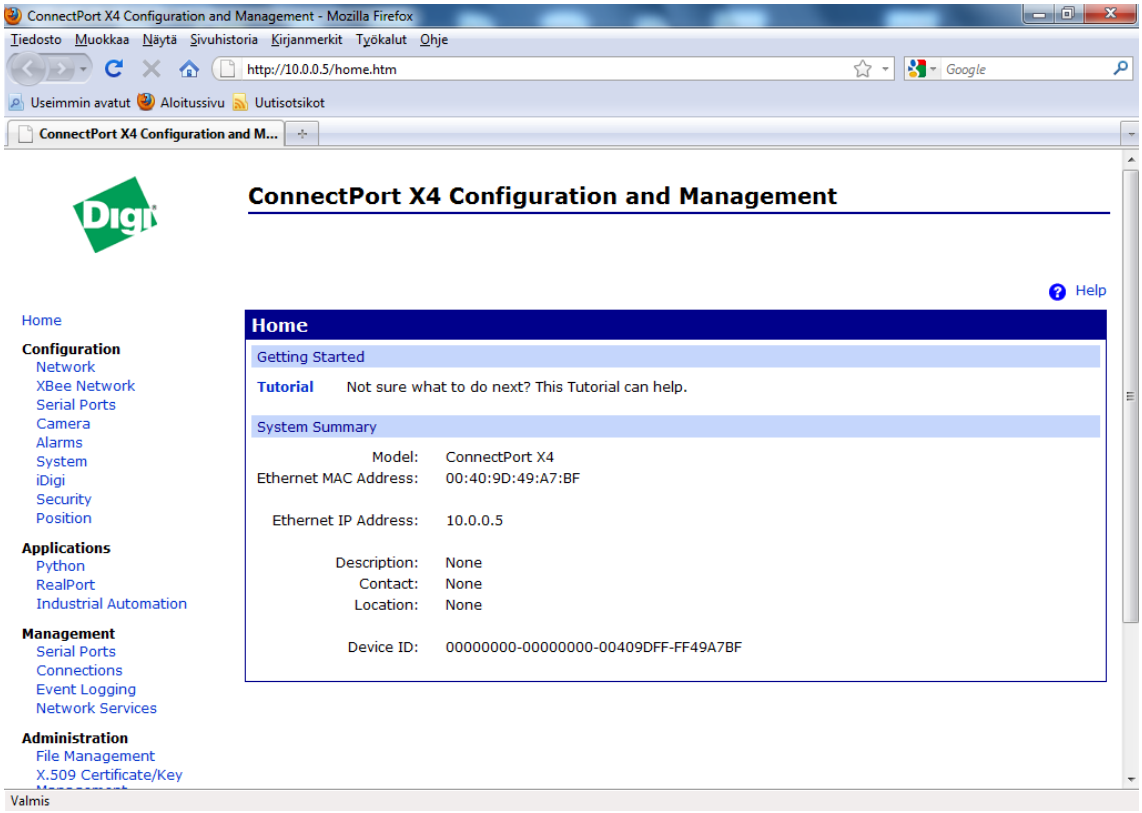
Connectportin asennus verkkoon

Connectportin täyden hyödyn saa ohjaamalla sitä verkon yli. Verkon yli ohjattavissa laitteissa on tietotekniikan kannalta aina haastavaa, kuinka ne löytyvät verkkopilvestä. Liian tiukat palomuurit yms. voivat hankaloittaa tätä vaihetta merkittävästi. Connectport asennettiin tästä johtuen kotiverkkoon, jossa on hieman vapaammat kulkuyhteydet. Tämänkin verkon tietoturva asetuksia jouduttiin säätämään hetken, mutta laite löytyi kohtuullisen hyvin Digin Device Discover utilityn avulla kuvan 35 mukaisesti.



Kuva 35. Digi Device Discoveryllä verkosta löydetty Connectport.

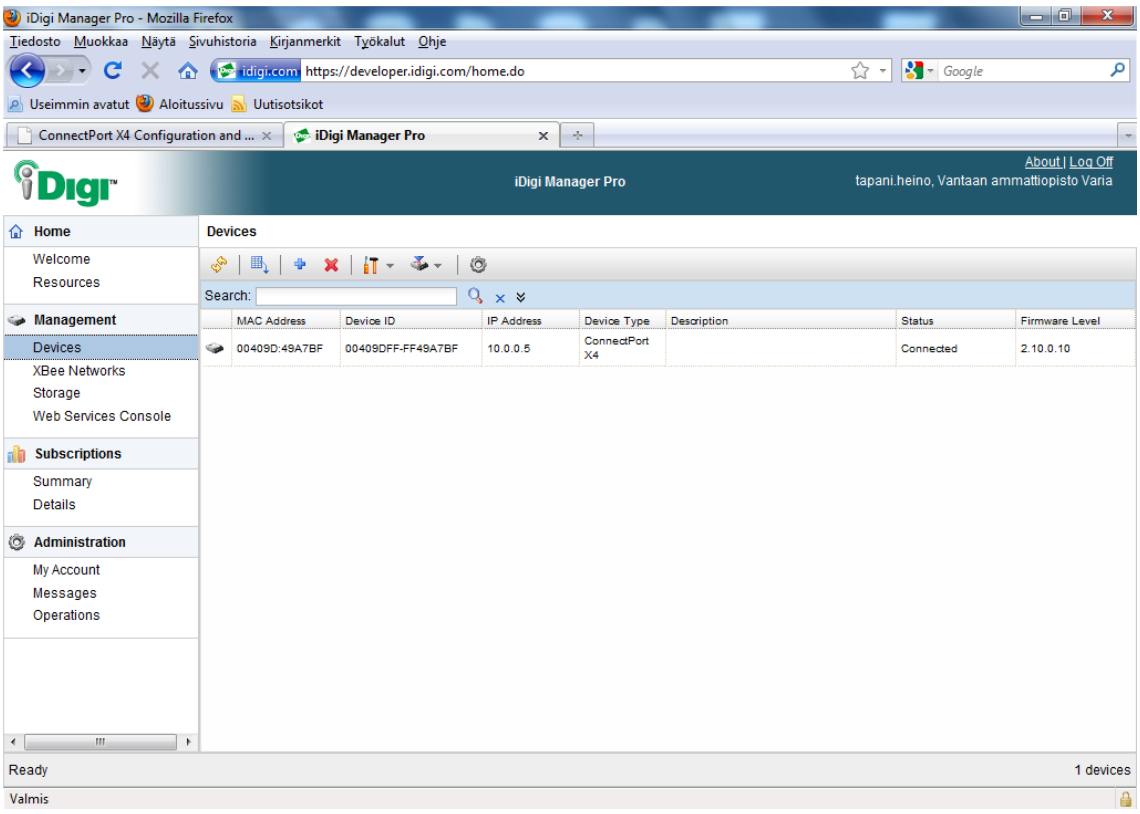
Kun Connectport on löydetty verkosta, voi sitä alkaa määritellä. Tämä onnistuu selainpohjaisesti laitteen löytymisen jälkeen ja tiedettäessä sen ip-osoite. Syöttämällä internetselaimeen laitteen ip-osoite päästään käsiksi sen hallintapaneeliin kuvan 36 mukaisesti.



Kuva 36. Connecportin määrittely internetselaimessa.

Hallintapaneelin toimivuus testattiin käynnistämällä Connectport internetselaimen kautta, laite käynnistyi uudestaan ohjelmistopohjaisesti, joten tämän jälkeen voitiin luottaa sen toimintaan. Seuraavaksi laite haluttiin liittää omaan serveriinsä, koska järjestelmää halutaan ohjata muualtakin kuin omasta sisäisestä kotiverkosta. Tässä kohtaa Digi on tullut vastaan Developers kit -sarjoissaan paljon. Se tarjoaa ilmaiseksi oman palvelimensa käyttöön sovelluskehitykseen. Toisin sanoen kehitysvaiheessa ei tarvitse rakentaa omaa web-pohjaista serveriään ja hioa sitä kyseiseen järjestelmään sopivaksi. Tämä säästää valtavasti aikaa, kun testijärjestelmä tätä varten on valmiina olemassa. Luomalla omat tunnuksensa ja rekisteröitymällä iDigi-palveluun voi oman laitteensa liittää Digin serveriin. Aluksi Digin serveri pitää laittaa Connectportin omiin

tietoihin sen omassa selainpohjaisessa järjestelmässä. Tämän jälkeen luomalla tili iDigin nettisivulle ja rekisteröitymällä saadaan otettua Digin serveri käyttöön. Laitteita etsimällä löydettiin Connectport. Sivujen avulla Connectport nimettiin omalla nimellään, etteivät laitteet mene sekaisin. Tämän jälkeen laite resetoitiin nettisivujen kautta, jotta varmistettaisiin niiden toimivuus. Tässä vaiheessa projektissa oli internetin yli ohjattava hallintapaneeli kuvan 37 mukaisesti.



Kuva 37. Digin serverialusta

Kun tämä haaste oli saatu toimimaan, seuraavana vaiheena oli lisätä päätelaitteita Connectportiin. Tämä vaihe osoitti järjestelmän potentiaalin. Koska kaikki laitteet oli määritelty ottamaan automaattisesti yhteys tiettyyn osoiteavaruuteen, osasivat päätelaitteet liittyä suoraan connectportiin heti, kun sen tietoturva-asetukset oli määritelty kuntoon. Hallintapaneelin laitepäivityksen jälkeen verkon laitteet löytyivät ja listautuivat sinne.

Nyt projektissa oli serveri, millä pystyi hallinnoimaan ja ohjaamaan laitteita verkon yli. Seuraava vaihe oli testata, että järjestelmä toimii varmasti. Tätä varten laitteiston asetuksia, kuten nimeämistä ja tunnistetietoja, muokattiin sekä tarkasteltiin niiden asetuksia. Tämän jälkeen laitteiden ja järjestelmän toimivuus varmistettiin samaan tyyliin kuin connectportin kanssa, siten että laitteita pyydettiin tunnistautumaan verkon yli Digin palvelimen kautta. Käytännössä tällä pyynnöllä haluttu laite vilkutti valoaan muutaman sekunnin. Tämä oli alku sille, että laitteisto pystyy itsenäisesti ryhmittymään verkkoon ja antamaan itsestään haluttaessa tunnistetietoja ja reagoimaan verkon yli tuleville käskyille. Samalla laitteisto alkoi olla myös valmis sen täyden potentiaalin käyttöön otolle Python-kielen avulla.

Python

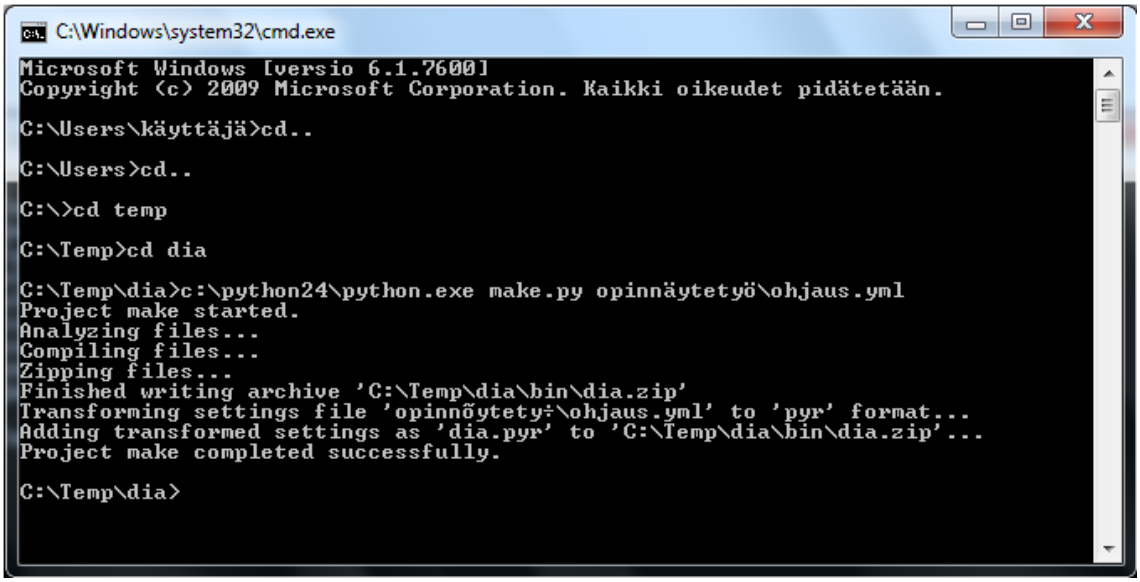
Python-kielen avulla laitteistoa pystyy ohjaamaan ajon aikana. Alustaan oli tarkoitus saada rakennettua Pythonin avulla järjestelmä, joka pystyy hyötykäyttämään testialustoja. Yksinkertaisena demona päätettiin luoda ympäristö, josta voitaisiin tarkkailla suoraan, mitä näppäintä käyttäjä painaa, ja antaa kenelle tarpeen vaatiessa informaatiota.

Python on itsessään ohjelmointikieli siinä missä muutkin, joten luonteva aloitus oli sen asennus. Digi tarjoaa suoraan sen laitteistojen käyttämää versiota Pythonista, joten se ladattiin koneelle. Teoriatasolla objektiivi on suhteellisen selvä, luoda ohjelma joka ottaa yhteyden haluttuihin laitteisiin, ja tulostaa niiden tiedot internetsivuilla ja antaa myös sitä kautta ohjata laitteita. Käytännössä objektiivi on haastava.

Selvittelyn jälkeen selvisi, että Digi toimittaa kehittäjille tukena iDigi Dia (Device Integration Application) softawarea. Käytännössä tämä on Python-pohjaisia esimerkkiohjelmia ja kirjastoja, jotka auttavat laitteisiin yhteyden muodostusta. Näiden avulla halutun toiminnon tekeminen vaikutti mahdolliselta. Esimerkkiohjelmista löytyi ohjelma, jolla ohjattiin kolmea eri laitetta vastaavalla tavalla, joten tätä muokkaamalla oli toivo saada oma ohjelma luotua. Käytännössä projektissa siis riitti Pythonin kääntäjän käyttö, itse koodia pystyttiin muokkaamaan suoraan tekstieditorissa.

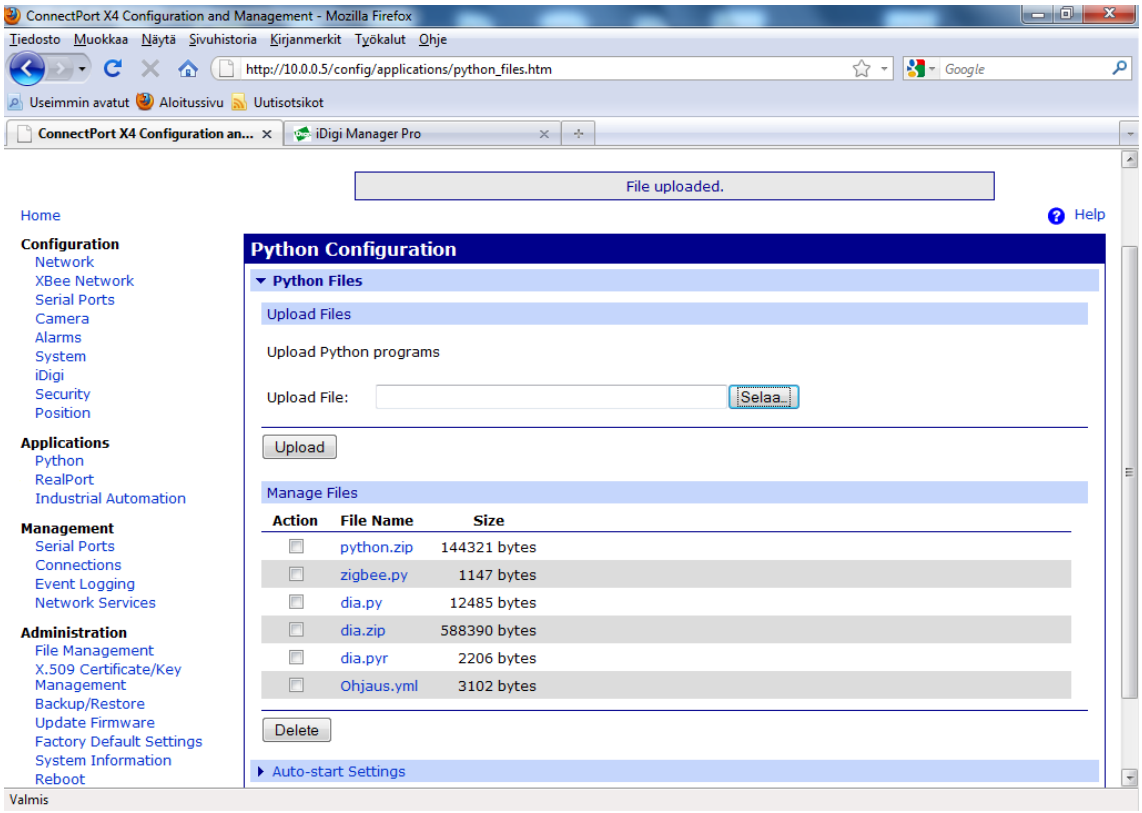
ZigBeen yksi fundamentaalisia periaatteita on, että vaikka kyseessä voi olla iso mesh-verkko, on jokainen laite yksilö, ja jokaisella laitteella on oma yksilöllinen tunnisteensa. Verkon jokaisella laitteella on siis omat yksilölliset osoitetietonsa ja tunnisteensa. Nämä tiedot saadaan Connectportin hallintapaneelin laitelistasta suoraan tietoon. Kun Python-koodissa halutaan esimerkiksi käyttöön kaksi testialustaa, on molemmat nimettävä erikseen, ja molemmilla pitää olla tiedossa niiden osoitetiedot. Testialusta yhden osoitetietojen kohdalla koodissa lukee extended_address: "00:13:a2:00:40:69:29:cc!". Tämä kertoo koodille suoraan laitteen yksilöllisen verkko-osoitteen. Samalla tavalla luotiin toisen testialustan tiedot koodiin.

Kun ohjelma on siinä kunnossa, että sitä haluaisi testata, on se käännettävä Pythonin avulla. Tämä tapahtuu command promptissa. Aluksi komentoriville on määriteltävä Pythonin asennuskansio ja sen käynnistys exe, tämän jälkeen make.py-käsky, jolla käännös tehdään, ja lopuksi käännettävän tiedoston hakemistopolku. Tämän jälkeen työ käännetään ja tallennetaan zip-muotoon kuvan 38 osoittamalla tavalla.



Kuva 38. Muokatun ohjelman kääntäminen

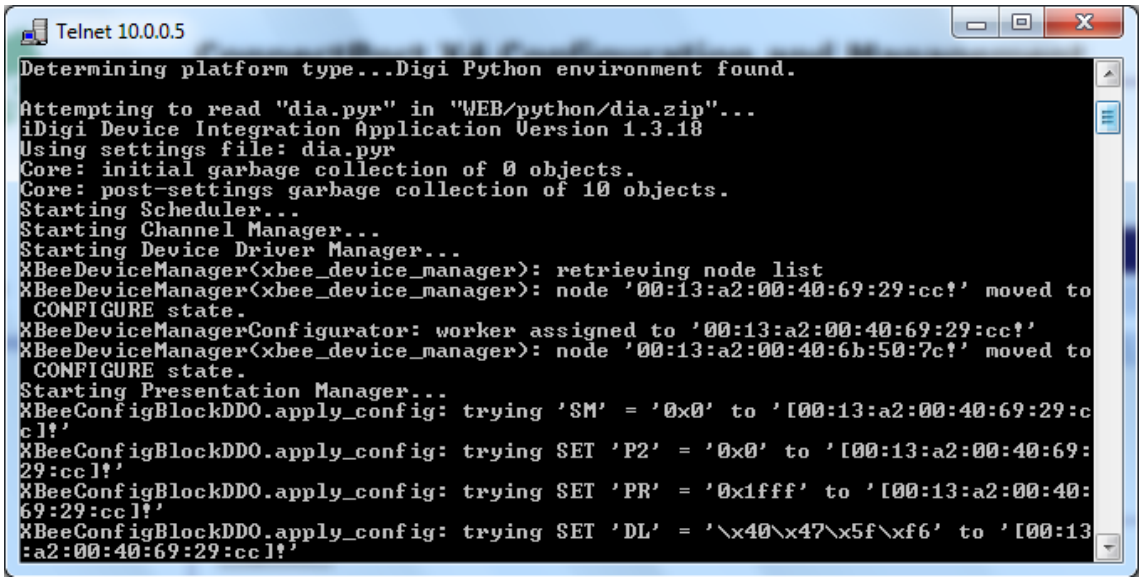
Kun työ on ohjelmoitu Pythonilla haluttuun muotoon ja käännetty, on se ladattava Connectportille. Tämä tapahtuu ottamalla yhteyttä Connectportiin esim. nettiselaimen kautta. Tiedosto ladataan sille kuvan 39 mukaisesti.



Kuva 39. Pythonin lataus Connectportille.

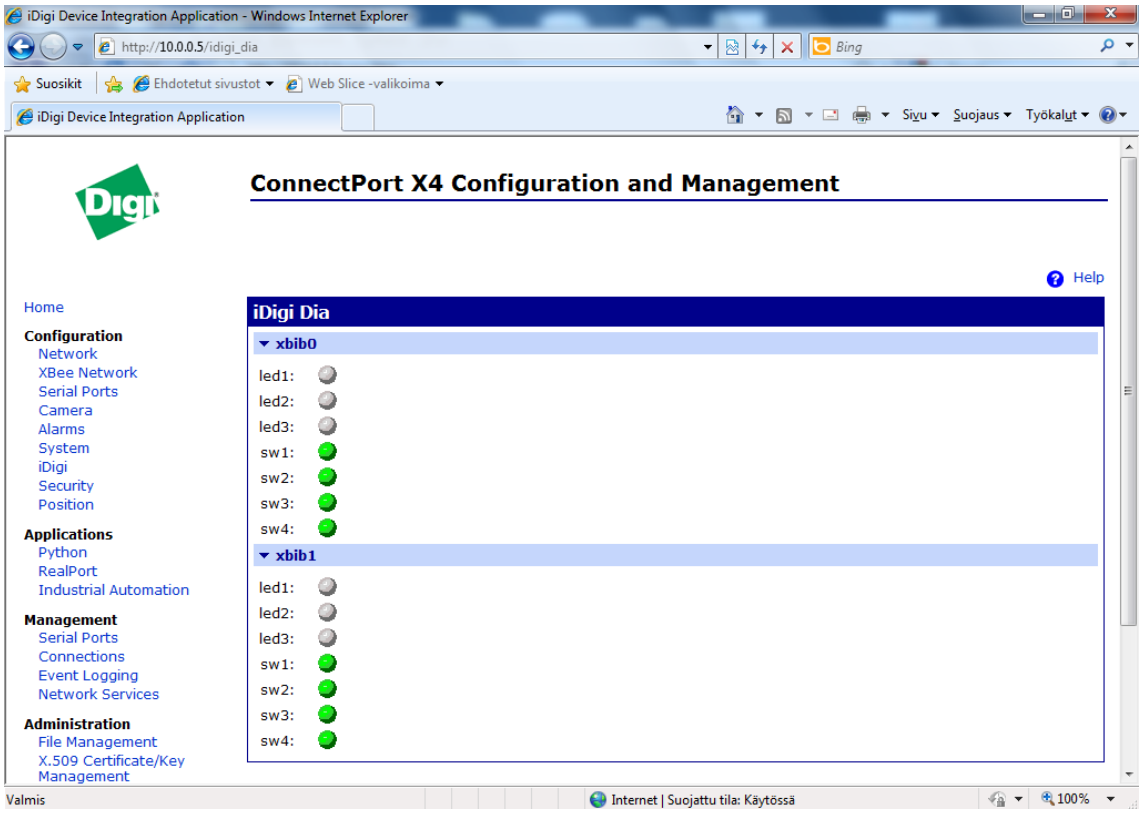
Ohjelman ajo

Tämän jälkeen testattiin, miten ohjelma toimii. Tätä varten on aluksi ohjelma käynnistettävä esim. Telnetin kautta ottamalla sillä yhteys Connectportiin ja käynnistämällä Python-komennolla haluttu tiedosto. Connectportilla ohjelmat voi myös haluttaessa laittaa käynnistymään automaattisesti, eli tätä vaihetta ei tarvitse aina toistaa, jos haluaa ohjelman käynnistyvän automaattisesti. Ohjelma lähtee lataamaan sen käynnistyttyä kuvan 40 mukaisesti.



Kuva 40. Python käynnistymässä Connectportissa.

Kun ohjelma on ladattu tausta-ajoon Connectportille, sitä voidaan testata käytännössä. Ohjelman toimintaa voi seurata ja hallita kuvan 41 mukaisesti suoraan internet selaimen yli tai sitten seurata sitä rivipohjaisesti Telnet-yhteyden kautta. Molemmissa järjestelmissä on omat hyvät ja huonot puolensa. Internetselaimen kautta toimiva seuraus ja hallinnointi on yksinkertaista ja visuaalisesti selkeää, mutta järjestelmä tulee hieman reaaliajasta perässä. Telnetin yli seurattaessa taas kaikki tapahtumat nähdään suoraan reaaliajassa, mutta loppukäyttäjälle tätä kautta ohjelman seuraaminen on turhan haastavaa.



Kuva 41. Äänestyslähettimien internetselaimessa toimiva hallinta- ja seurantapaneeli.

Tässä kohtaa molemmat järjestelmät todettiin toimiviksi. Selaimen kautta toimiva käyttö on vaivattoman oloista, ja on selkeää nähdä, miten napin painallukset näkyvät myös selaimessa. Selaimen kautta voi myös haluamastaan ledistä klikkaamalla saada ledin fyysisesti palamaan testialustalla. Tämä auttaa testialustojen tunnistamisessa ja demoaa interaktiivisuutta molempiin suuntiin. Ainoa huono puoli on, että napin painallus tulee viiveellä selaimeen. Telnetin yli toimintoja seuratessa kaikki toiminnot näkyvät reaaliajassa. Tästä näkymästä voidaan todeta, että napin painallukset otetaan talteen hyvin pienellä viiveellä, mikä on helpotus siinä mielessä, että ohjelma toimii. Toinen hyvä puoli on, että Telnetin kautta näkee suoraan kaikki suoritettavat komennot reaaliajassa. Huonoina puolina tekstiä tulee pahimmillaan sen verran nopeaa tahtia, että on vaikea välillä pysyä sen perässä. Toisekseen tekstisisällön ymmärrettävyys voi olla hyvin vaikeaa, jos ei ole ollut itse ohjelmaa suunnittelemassa.

XML

Nyt kun järjestelmä toimii tähän asti, viimeinen vaihe on varmistaa XML-tallennuksen toimivuus. Pythonin koodiin pitää määritellä keräyssykli, näytteiden määrä, mihin tallennetaan ja millä nimellä. Tässä kohtaa ongelmaa aiheutti logiikka, miten tiedot kannattaisi kerätä, kuinka nopea tallennussykli pidetään ja kuinka monta näytettä syklissä kerätään. Lopuksi todettiin, että maksimoimalla näytteenottoajan ja minimoimalla näytteenottomäärän päästään hyvään lopputulokseen. Tämän perusteella koodiin laitettiin interval-ajaksi näytteenottoon 1800 sekuntia eli puoli tuntia per kohta, ja näytteiden määräksi laitetaan arvo sen mukaan, montako lähetinyksikköä on käytössä. Vaikka näytteenoton aika on pitkä, jatkaa se silti eteenpäin aina, kun mitattavien näytteiden määrä on täynnä. Toisin sanoen kun kaikki osallistujat ovat vastanneet, jatkaa koodi eteenpäin ja aloittaa uuden näytteenoton. Uudet näytteenotot tallennetaan aina uuteen XML-tiedostoon. Jos kysymme kymmenen kysymystä, on meillä kymmenen XML-tallennustiedostoa. Näytteet voitaisiin periaatteessa kirjoittaa yhteenkin tiedostoon, mutta erillisiin tiedostoihin tallentaminen helpottaa yksittäisten vastausten tarkistelua. Perinteisten tiedostomuotojen kanssa ei olisi järkevää tallennella tiedonmuruja useampaan tiedostoon, mutta XML-koodilla esim. Excel voidaan määritellä hakemaan suoraan yhteen taulukkoon useampi XML-tiedosto aina automaattisen joten tästä ei siis aiheudu loppukäyttäjälle ylimääräistä vaivaa.

Excel

XML-tietojen koostamiseen päätettiin käyttää Excel 2010:tä. Kooste suunniteltiin siten, että etusivulla näkyvät ainoastaan tulokset eli mitä kukakin käyttäjä on vastannut ja ovatkin vastaukset oikeita vai vääriä. Serveriltä haettavat XML-tiedostot luetteloidaan toiselle välisivulle, mistä niitä pystyy tarkistelemaan lähemmin sekä päivittämään tarpeen vaatiessa.

Käytännössä XML-tiedostojen hakupolun määrittely Exceliin on hyvin helppoa. Isoin vaiva oli miettiä tietojen tallennuslogiikka XML-tiedostoihin, montako tallennettua tiedostoa pitää olla ja mitä niiden pitää sisältää jne. XML-tiedostoihin on määritelty äänestyslähettimien nimet, joten niiden avulla pystytään tiedostoja luettaessa

lajittelemaan data käyttäjien vastauksista lähettimen nimen perusteella. Ainoa huono puoli on muutamat haamutallennukset, joita XML-tiedostoihin välillä tallentaa Pythonin tallennusehdoista huolimatta. Toinen negatiivinen asia on, että vaikka Excelille opettaa Digin serverin salasanan, pitää se jostain syystä aina silti hyväksyä haettaessa tiedot serveriltä uudestaan. Muuten järjestelmä toimii suunnitellusti.

Järjestelmän toimintaa on havainnollistettu taulukoilla 3 ja 4 yksinkertaisen esimerkin avulla, jossa käyttäjät vastaavat annettuihin kysymyksiin kyllä tai ei. Taulukko 3 on Excelin etusivulla näkyvä näkymä. Esimerkin ohjelmassa Pelaaja 1 on vastannut kyllä kaikkiin kysymyksiin ja Pelaaja 2 ei. Taulukko näyttää kysymyksen, onko annettu vastaus oikein vai väärin sekä mikä oli kysymyksen oikea vastaus. Taulukko 4:ssä on toiselle välilehdelle haetut XML-tiedostojen koosteet. Nämä arvot siis haetaan suoraan serveriltä. Taulukon 3 Score boardiin on määritelty tietojen haku täältä. Solujen hakuehdoiksii on määritelty seuraava ehtolause:

```
=IF('Haettavat arvot'!D5="xbib1.sw1";"Oikein";"Väärin")
```

Tällä ehdolla haetaan Pelaaja 1:n vastaus ensimmäiseen kysymykseen. Jos hän on painanut kytkintä yksi eli sw1:tä, on hän vastannut oikein, muuten väärin. Samalla logiikalla katsotaan molempien pelaajien vastaukset kaikkiin kysymyksiin.

Taulukko 3. Exceliin tulostettavat vastaukset.

Score board			
Kysymys	Pelaaja 1 vastasi	Pelaaja 2 vastasi	Oikea Vastaus
Onko 1 + 1 = 2 ?	Oikein	Väärin	Kyllä
Onko 2 + 1 = 3 ?	Oikein	Väärin	Kyllä
Onko 3 + 1 = 4 ?	Oikein	Väärin	Kyllä
Onko 4 + 1 = 5 ?	Oikein	Väärin	Kyllä
Onko 5 + 1 = 6 ?	Oikein	Väärin	Kyllä

Taulukko 4. Serveriltä haettavat XML-tiedostot mitkä ovat taulukoitu Exceliin.

Kysymys 1	**name**	**value**	**unit**	**timestamp**
Vastaaja 1	xbib1.sw1	True		1970-01-01 00:01:37
Vastaaja 2	xbib0.sw2	True		1970-01-01 00:01:38
Kysymys 2	**name**	**value**	**unit**	**timestamp**
Vastaaja 1	xbib1.sw1	True		1970-01-01 00:01:16
Vastaaja 2	xbib0.sw2	True		1970-01-01 00:01:17
Kysymys 3	**name**	**value**	**unit**	**timestamp**
Vastaaja 1	xbib1.sw1	True		1970-01-01 00:01:23
Vastaaja 2	xbib0.sw2	True		1970-01-01 00:01:24
Kysymys 4	**name**	**value**	**unit**	**timestamp**
Vastaaja 1	xbib1.sw1	True		1970-01-01 00:01:26
Vastaaja 2	xbib0.sw2	True		1970-01-01 00:01:26
Kysymys 5	**name**	**value**	**unit**	**timestamp**
Vastaaja 1	xbib1.sw1	True		1970-01-01 00:01:32
Vastaaja 2	xbib0.sw2	True		1970-01-01 00:01:32

4 Työn testaus

Työn testaukseen käytettiin kriteereinä verkon yhdistyvyyttä, hallinnointia ja tiedonprosennoinnin luotettavuutta. Näiden kolmen kriteerin yhteispainotuksella voidaan määritellä työn onnistumisaste. Luonnollisesti isoin painoarvo on äänestyksen sujuvuudella ja käyttäjäkokemuksella eli sillä, miltä järjestelmä näyttää ja miten se toimii loppukäyttäjän silmissä.

Verkko

Usein käytännön painotteisissa töissä tulee vastaan hankaluuksia, mihin osaan on osattu varautua ja mihin ei. Tässä projektissa ennakointiin, että verkon yhdistyminen voi olla mahdollinen hankala tekijä. Yllättävästi tämä osa-alue on ollut aivan alusta asti kaikkein varmin osa-alue. Verkko on jokaikinen kerta muodostonut yhä uudestaan luotettavasti ja toiminut ilman minkäänlaisia ongelmia. Verkon laitteistoa on testattu useassa huoneessa yhtä aikaa ja eri kombinaatioilla, mutta kertaakaan ei ole käynyt niin että yhteys ei olisi muodostunut tai että kaikkia verkon laitteita ei olisi löytynyt. Järjestelmä toimii todella helppokäyttöisesti, ensi määritysten jälkeen laitteet ovat pystyneet aina löytämään toisensa. Myös järjestelmän käyttövarmuus on hämmästyttävää. Ainoat katkokset on saatu aikaan ainoastaan tahallaan yrittämällä kahden laitteen välillä, muuten laitteisto toimii todella luotettavasti koko ajan.

Hallinnointi

Hallinnointi on ollut hieman kaksiteräinen miekka. Hallinnoinnin pystyttäminen oli paljon vaikeampaa kuin aluksi luultiin, kun taas sen ylläpito on hyvinkin helppoa. Tästä voitaneen vetää johtopäätös, että laitteiston kehittyneyden takia sen pystyttäminen voi olla raskasta ja sen ansiosta ylläpito helppoa. Todettakoon myös, että ensimmäisellä kerralla, kun tekee uusia asioita, kaikki tuntuu hankalalta. Seuraava verkko olisi todennäköisesti paljon nopeammin rakennettu alusta loppuun. Tästäkin huolimatta verkon alkumääritykset ovat hyvin tärkeitä, työläitä ja kriittisiä lopputuloksen kannalta

tällä järjestelmällä. Kun verkko kerran määritellään toimimaan kunnolla, säästää se mittavia määriä työtä, ja järjestelmän älykkyyden ansiosta tulevaisuuden työmäärät varmasti laskevat. Tiivistettynä tässä projektissa hallinnoinnin pystyttämiseen käytettiin mittavia määriä aikaa, mutta sen jälkeen kun kaikki toimii, ei järjestelmään pidä käytännössä koskeakaan.

Tiedon prosessointi

Tiedon prosessointi on ollut tämän projektin isoin haaste. Osittain tämä on ollut tietoinen valinta, koska projektin alussa päätettiin käyttää API-kehyksiä, siitäkin huolimatta, että tiedettiin niiden korkeamman kehittyneisyysasteen takia myös niiden käytön olevan hankalampaa. Tästä johtuen tässä kohdassa on edetty hyvin pienin askelin. Aluksi järjestelmä laitettiin toimimaan moduulin ja tietokoneen välillä, tämän jälkeen vasta moduulin ja Connectportin välillä ja lopuksi Connectportin ja serverin välillä. Joku voi miettiä, miksi tiedon prosessoinnin yhteydessä puhutaan vain tiedonsiirrosta, mutta valittava tiedonsiirtotapa ja -tie vaikuttavat yllättävän paljon prosessoitavaan tietoon, kuten missä muotoa tieto lähetetään ja missä kohtaa mitäkin osaa tiedosta prosessoidaan.

Oikean tiedonsiirtotavan järjestettyä myös tiedon prosessointimalli selkiintyi. Haluttiin, että äänestyslähetin lähettää halutun sanan tai kirjaimen valitusta I/O-linjasta suoraan Connectportiin. Tässä pisteessä tietoa haluttiin jo jalostaa, jotta sen voisi lähettää serverille universaalissa muodossa. Tätä varten Pythonin avulla karsitaan lähetetystä tiedosta lähettäjä ja valinta taulukoidaan XML-muotoon, ja tämä tieto lähetetään serverille. Tämä mahdollistaa sen, että serveriltä tiedot voi hakea suoraan Excelille, ja siellä prosessoida tiedot lopulliseen muotoon taulukoihin loppukäyttäjälle esitettäväksi.

Käytännössä tämä mahdollisesti pitkäkin tie voi aiheuttaa monia ongelmia. Määritysvaiheessa ongelmia oli API:n kanssa. Seuraava iso ongelma oli Pythonilla tietojen luku luetettavasti. Pythonin ensimmäinen iso ongelma oli vastaanotettavan datan luku riittävällä nopeudella. Jokainen napinpainallus pitää saada rekisteröityä sisään lyhyellä viiveellä, jottei puolet napinpainalluksista jää kirjaamatta sillä välin kun

ohjelma tekee jotain muuta. Toinen iso ongelma oli tietojen kirjaus, miten ne saadaan kirjoitettua XML-muotoon, ja siten, että ne olisivat helpoiten Excelillä luettavissa.

Järjestelmän testaus

Järjestelmää testattiin sen alkuperäiskäyttötarkoituksen mukaisesti. Testiin otettiin kolme yksilöllistä äänestyslaitetta, joilla vastattiin testikysymyksiin. Äänestyslaitteina toimi kaksi testialustaa ja itse tehty demoäänestyslaite. Testin ensimmäinen vaihe sujui erinomaisesti laitteiston verkottuessa automaattisesti.

Toisen vaiheen hallinnoinnissa ei tarvitse tehdä halutessaan mitään, koska Connectportin saa määriteltyä autokäynnistämään halutut ohjelmat, kuten äänestyslaitteiden lukuohjelman aina käynnistyksen yhteydessä.

Kolmantena vaiheena oli viiteen kysymykseen vastaaminen. Tämän jälkeen tulokset mentiin tarkistamaan Excelistä.

Ensimmäinen ongelma tuli salasanasuojatusta serveristä. Kentät eivät päivity automaattisesti, vaan ne pitää päivittää hiiren avulla, ja salasana pitää syöttää aina kun Excelin avaa uudelleen. Toisena ongelmana kolmella vastauksella mukaan tulee välillä useampi vastaus, vaikka minimi- ja maksimiarvot on määritelty tasan kolmeen. Siten järjestelmä ei toimi täysin luotettavasti ja vaivattomasti loppukäyttäjän silmissä.

Kokonaisuutena järjestelmä toimi alusta loppuun kohtuullisesti. Virheet olivat pieniä ja korjattavan oloisia. Järjestelmän alkuperäinen määritys siis tavoitettiin.

Analysointi

Järjestelmä toimii alustavasti halutulla tavalla. Alkuvaiheet ovat hyvin luotettavia ja vakuuttavia, mutta pullonkaula tulee Pythonin kohdalla. Tiedon keräys komentoscriptin avulla ei toimi täysin luotettavasti. Toisekseen järjestelmän saa täysin sekaisin siten, että yksi käyttäjä innostuu painamaan useamman kerran peräkkäin omia näppäimiään. Myös käyttövarmuudessa on siis huomauttamasti. Viimeisenä asiana tiedot pitäisi saada serveriltä talteen ilman jokakertaista salasanan varmistusta.

Komentoscriptien jouhevuutta ja näppäinten painallusten määrää pystynee kehittämään hieman koodia hiomalla, mutta todennäköisesti tällaisen järjestelmän saisi toimimaan paremmin ilman Zigbeetä radiotekniikan päällä. Serverin osuuden taas voisi saada toimimaan paremmin esim. poistamalla salasanat käytöstä. Lopputuloksena mesh-verkon toimintavarmuutta voidaan yleisellä tasolla pitää vähintäänkin vakuuttavana, mutta tällaisessa projektissa se ei pääse oikeuksiinsa. Jos käyttäisimme samaa järjestelmää esimerkiksi ottamaan mittausdataa vaikka kodin lämpötiloista ja kosteuksista esim. kerran tunnissa, eivät satunnaiset virheet haittaisi, koska tälläiset arvot voidaan poimia tehokkaasti XML-muodossa omista tunnisteista, ja vaikka jokin arvo olisi tallentunut kahdesti, ei se haittaisi mitään. Toisin sanoen järjestelmä toimii melkein täydellisesti, mutta sen käyttötarkoitus tarkkaa reaaliaikaista monivastausta tukevana sovelluksena ei vastaa parhaiten sen vahvuuksia ja käyttötarkoitusta. Myös Zigbeen vahvasta verkon itsenäisyydestä ei ole tällaisessa sovelluksessa sellaista etua, että sitä tarvitsisi käyttää. Loppuarviona Zigbee ja älyverkot vakuuttavat nykyisellä toimintavarmuudella todella hyvin. Ongelma on, että se ei sovellu optimaalisesti tällaiseen sovellukseen, vaan sen vahvuudet ovat mittausdatan keräämisessä, varsinkin olosuhteissa, missä sitä ei tarvitse tehdä jatkuvasti sekä missä verkottumista kaivataan enemmän, kuten paikoissa missä mittausdata tulee hajanaisemmin eri paikoista.

5 Päätelmät

Työssä kerättiin tietoa älykkäistä anturiverkoista ja näiden tietojen perusteella rakennettiin älykäs äänestysjärjestelmä. Aluksi kartoitettiin olemassa olevia järjestelmiä sekä määriteltiin prosessointi- ja tiedonsiirtovaiheen toteutus. Tämän selvityksen jälkeen rakennettiin konkreettinen järjestelmä, minkä avulla pystyttiin analysoimaan siihen valittuja tekniikoita.

Käytännön projektin voi todeta onnistuneen kohtuullisesti. Älyverkon rakennus onnistui, se pystyy verkottumaan, ja korjaamaan itse itseään, sitä pystyy hallinnoimaan verkon yli ja se raportoi itsenäisesti tekemisensä serverille, mistä sitä pystytään suoraan lukemaan Exceliin. Ainoa negatiivinen asia oli todeta testausvaiheessa, että mesh-tekniikka ei sovi optimaalisesti äänestysjärjestelmiin ja on periaatteessa liian hienostonut tällaiseen käyttöön.

Opinnäytetyön tavoitteet saavutettiin. Haastavaksi työn teki sen laajat kokonaisuudet. Teoriatason määritelmät veivät ennakoitua enemmän aikaa tekniikan vaikeusasteen ja aihealueen tiedon määrästä johtuen. Älykäs äänestysjärjestelmä toteuttiin laajan taustaselvityksen ja aihealueen osiin kategorisoinnin avulla. Isoin haaste oli saada eri osa-alueet toimimaan saumattomasti yhdessä.

Työn pohjalta voidaan todeta, että ZigBee alkaa olla valmis järjestelmä ja mesh-verkot ovat selvästi läpimurtonsa partaalla. Monipuolisuutensa ansiosta Zigbeellä pystyy kehittämään ja luomaan sovelluksia laaja-alaisesti. Laitteisto on todella pitkälle kehittynyttä ja vähävirtaista, mikä mahdollistaa jo nyt aitojen itsenäisten mesh-verkkojen täyden hyödyntämisen.

Lähteet

1 Application Note. 2004. Maxstream.

<ftp://66.77.174.26/support/images/XST-AN010a-MaximizingRange.pdf> Luettu 15.8.2010

2 Calculate Fresnel Zone. Broadband buyer.co.uk <http://www.broadbandbuyer.co.uk/BuyersGuide/FAQDetail.asp?SetID=91&TextID=687> Luettu 1.9.2010

3 RF Planning. Danets. <http://www.danets.com/turbotenna/rf%20planning1.php> Luettu 2.9.2010

4 Rf Basics. Digi. <http://www.google.fi/url?sa=t&source=web&cd=4&ved=0CDQQFjAD&url=ftp%3A%2F%2Fdelmation.xs4all.nl%2FOpvraagbare%2520bestanden%2FZigBee%2520presentaties%2F080528_1_Digi_RF_Basics.ppt&ei=Z0RWTZW6GYT5sgbnlommCw&usg=AFQjCNF2GJ1O8z9zCqg0DiLWcH-Ii6JgkQ> Luettu 4.9.2010

5 ZigBEE. ZigBee Alliance <http://www.google.fi/url?sa=t&source=web&cd=6&ved=0CEkQFjAF&url=http%3A%2F%2Fuk.farnell.com%2Fimages%2Fen%2Fede%2Fcomm%2Fpdfs%2F23.Zigbee.pps&ei=aqlXTbiuKoHGtAaQiaWlCw&usg=AFQjCNE69K-ZjfEb-B5uBLm0BhSGCFK0Mw> Luettu 6.9.2010

6 Making ZigBee buzz using the right development tools. Iar Systems. <http://iar.com/website1/1.0.1.0/1281/1/> Luettu 8.9.2010

7 ZigBee Alliance. <http://www.zigbee.org/Home.aspx> Luettu 6.9.2010

8 Zigbee protocol stack. Meshnetics. <http://www.meshnetics.com/learning/index.php?show=21> Luettu 17.10.2010

9 Reply mini. Replysystem. <http://www.replysystems.com/reply-products/replymini-worldwide.shtml> Luettu 27.10.2010

10 Langaton äänestysjärjestelmä. Valtuusto. <http://www.kunnanvaltuusto.net/> Luettu 27.10.2010

11 Monitoring Fluid Levels at an Automobile Quick Lube Station. Digi. <http://www.digi.com/learningcenter/stories/monitoring-fluid-levels-at-an-automobile-quick-lube-station> Luettu 9.1.2011

12 Hotels solution: In-Room Energy Management. Scheider Electric. <http://www.schneider-electric.com/sites/corporate/en/solutions/business_segments/hotels/guest-room-control/in-room-energy-management.page> Luettu 15.2.2010

13 ADOPTION OF ACTIVE LEARNING IN A LECTURE-BASED ENGINEERING CLASS. CDIO. <http://www.cdio.org/knowledge-library/documents/adoption-active-learning-lecture-based-engineering-class> Luettu 9.2.2011

14 XBee-PRO ZB Programmable Dev Kit (Europe and other international). Digi. <http://www.digi.com/products/model.jsp?lid=EN&pgid=227&pfid=179&mtid=3583&amtid=3583&pm=Y> Luettu 12.9.2010

15 iDigi™ Professional Development Kit ZB. Digi. <http://www.digi.com/products/wirelessdropinnetworking/idigi-kits/prodevkit.jsp#overview> Luettu 12.9.2010

16 Connectport X4 ZigBee-Ethernet Router. RS. <http://uk.rs-online.com/web/7043596.html> Luettu 2.2.2011

17 XBee-Pro S2B. Compel. <http://catalog.compel.ru/blog/2010/07/14/xbee-pro-s2b-dve-novye-versii-moshhnyx-zigbee-modulej/> Luettu 3.2.2011

18 XBIB-U-DEV DIGI INTERNATIONAL. SOS Electronics Europe. <http://www.soselectronic.com/?str=371&artnum=79850> Luettu 3.2.2011

19 XBee XStick USB Dongle. RS. <http://za.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=7043616> Luettu 6.2.2011

20 New XBee Wall Router Makes it Easy to Deploy and Extend ZigBee Networks. eHomeUppgrade. <http://www.ehomeupgrade.com/2007/12/11/new-xbee-wall-router-makes-it-easy-to-deploy-and-extend-zigbee-networks/> Luettu 8.2.2011

21 Low Power Contest per Microcontrollori. L'Elettronica Open Source. <http://it.emcelettronica.com/low-power-contest-microcontrollori> Luettu 6.2.2011

IEEE-standardit

Name	Description	Note
IEEE 802.1	Bridging (networking) and Network Management	
IEEE 802.2	LLC	inactive
IEEE 802.3	Ethernet	
IEEE 802.4	Token bus	disbanded
IEEE 802.5	Defines the MAC layer for a Token Ring	inactive
IEEE 802.6	MANs	disbanded
IEEE 802.7	Broadband LAN using Coaxial Cable	disbanded
IEEE 802.8	Fiber Optic TAG	disbanded
IEEE 802.9	Integrated Services LAN	disbanded
IEEE 802.10	Interoperable LAN Security	disbanded
IEEE 802.11 a/b/g/n	Wireless LAN (WLAN) & Mesh (Wi-Fi certification)	
IEEE 802.12	demand priority	disbanded
IEEE 802.13	Used for 100BASE-X Ethernet	
IEEE 802.14	Cable modems	disbanded
IEEE 802.15	Wireless PAN	
IEEE 802.15.1	Bluetooth certification	
IEEE 802.15.2	IEEE 802.15 and IEEE 802.11 coexistence	
IEEE 802.15.3	High-Rate wireless PAN	
IEEE 802.15.4	Low-Rate wireless PAN (e.g. ZigBee)	
IEEE 802.15.5	Mesh networking for WPAN	
IEEE 802.16	Broadband Wireless Access (WiMAX certification)	
IEEE 802.16.1	Local Multipoint Distribution Service	
IEEE 802.17	Resilient packet ring	
IEEE 802.18	Radio Regulatory TAG	
IEEE 802.19	Coexistence TAG	
IEEE 802.20	Mobile Broadband Wireless Access	
IEEE 802.21	Media Independent Handoff	
IEEE 802.22	Wireless Regional Area Network	
IEEE 802.23	Emergency Services Working Group	New (March, 2010)

Xbee PRO S2B Specification

Specifications of the XBee®/XBee PRO® ZB RF Module

Specification	XBee	XBee-PRO (S2)	XBee-PRO (S2B)
Performance			
Indoor/Urban Range	up to 133 ft. (40 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant
Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled	50mW (+17 dBm) 10mW (+10 dBm) for international variant	63mW (+18 dBm) 10mW (+10 dBm) for international variant
RF Data Rate	250,000 bps	250,000 bps	250,000 bps
Data Throughput	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)
Serial Interface Data Rate (software selectable)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)
Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled	-102 dBm	-102 dBm
Power Requirements			
Supply Voltage	2.1 - 3.6 V	3.0 - 3.4 V	2.7 - 3.6 V
Operating Current (Transmit, max output power)	40mA (@ 3.3 V, boost mode enabled) 35mA (@ 3.3 V, boost mode disabled)	295mA (@3.3 V) 170mA (@3.3 V) international variant	205mA, up to 220 mA with programmable variant (@3.3 V) 217mA, up to 232 mA with programmable variant (@3.3 V), international variant
Operating Current (Receive))	40mA (@ 3.3 V, boost mode enabled) 38mA (@ 3.3 V, boost mode disabled)	45 mA (@3.3 V)	47 mA, up to 62 mA with programmable variant (@3.3 V)
Idle Current (Receiver off)	15mA	15mA	15mA
Power-down Current	< 1 uA @ 25°C	3.5 µA typical @ 25°C	3.5 µA typical @ 25°C
General			
Operating Frequency Band	ISM 2.4 GHz	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960 x 1.297 (2.438cm x 3.294cm)	0.960 x 1.297 (2.438cm x 3.294cm)
Operating Temperature	-40 to 85º C (industrial)	-40 to 85º C (industrial)	-40 to 85º C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector	Integrated Whip, Chip, RPSMA, or U.FL Connector	Integrated Whip,PCB Embedded Trace, RPSMA, or U.FL Connector
Networking & Security			
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Number of Channels	16 Direct Sequence Channels	14 Direct Sequence Channels	15 Direct Sequence Channels
Channels	11 to 26	11 to 24	11 to 25
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
Agency Approvals			
United States (FCC Part 15.247)	FCC ID: OUR-XBEE2	FCC ID: MCQ-XBEEPRO2	FCC ID: MCQ-PROS2B
Industry Canada (IC)	IC: 4214A-XBEE2	IC: 1846A-XBEEPRO2	IC: 1846A-PROS2B
Europe (CE)	ETSI	ETSI	ETSI

Xbee PRO S2B Pin Signals

Pin Assignments for the XBee-PRO Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Default State	Description
1	VCC	-	-	Power supply
2	DOUT	Output	Output	UART Data Out
3	DIN / $\overline{\text{CONFIG}}$	Input	Input	UART Data In
4	DIO12	Both	Disabled	Digital I/O 12
5	$\overline{\text{RESET}}$	Both	Open-Collector with pull-up	Module Reset (reset pulse must be at least 200 ns)
6	RSSI PWM / DIO10	Both	Output	RX Signal Strength Indicator / Digital IO
7	DIO11	Both	Input	Digital I/O 11
8	[reserved]	-	Disabled	Do not connect
9	$\overline{\text{DTR}}$ / SLEEP_RQ/ DIO8	Both	Input	Pin Sleep Control Line or Digital IO 8
10	GND	-	-	Ground
11	DIO4	Both	Disabled	Digital I/O 4
12	$\overline{\text{CTS}}$ / DIO7	Both	Output	Clear-to-Send Flow Control or Digital I/O 7. CTS, if enabled, is an output.
13	ON / $\overline{\text{SLEEP}}$	Output	Output	Module Status Indicator or Digital I/O 9
14	VREF	Input	-	Not used for EM250. Used for programmable secondary processor. For compatibility with other XBEE modules, we recommend connecting this pin voltage reference if Analog sampling is desired. Otherwise, connect to GND.
15	Associate / DIO5	Both	Output	Associated Indicator, Digital I/O 5
16	$\overline{\text{RTS}}$ / DIO6	Both	Input	Request-to-Send Flow Control, Digital I/O 6. RTS, if enabled, is an input.
17	AD3 / DIO3	Both	Disabled	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Both	Disabled	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Both	Disabled	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Both	Disabled	Analog Input 0, Digital IO 0, or Commissioning Button