



TILAVERKKO

Luokkahuoneiden käyttöasteen havaitse-
minen ja tiedon lähettäminen langattoman
hyppyverkon avulla

Miikka Mikkonen

Opinnäytetyö
Joulukuu 2012
Tietotekniikan koulutusohjelma
Sulautetut järjestelmät ja elektroniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Sulautetut järjestelmät ja elektroniikka

MIIKKA MIKKONEN:

Tilaverkko

Luokkahuoneiden käyttöasteen havaitseminen ja tiedon lähettäminen langattoman hyppiverkon avulla

Opinnäytetyö 57 sivua, josta liitteitä 17 sivua
Joulukuu 2012

Opinnäytetyötä alettiin suunnitella Tampereen Ammattikorkeakoulussa havaitun tilanpuutteen takia. Varattuja tiloja ei ole käytetty tehokkaasti, ja tähän tuli keksiä ratkaisu. Tätä varten tarkoituksena oli rakentaa järjestelmä, joka havaitsee tilankäytön, eli onko tila käytössä vai vapaana, ja lähettää tämä tieto langattomasti. Mahdollisuuksien mukaan järjestelmän tuli myös arvioida, kuinka monta henkilöä tilassa milläkin hetkellä on. Tilankäytön havaitsemisesta hankaloittaa oppilaitoksen säädökset, joiden mukaan tiloihin ei saa asentaa kameroita. Näin ollen on myös tutkittava, millä antureilla saadaan tilankäyttö havaittua luotettavasti. Järjestelmästä oli myös rakennettava huokea ja yksikköhinnasta saatava pieni johtuen mahdollisuudesta, että laitteita tulisi sijoittaa satoja kappaleita kattamaan koko oppilaitos. Tämä estää valmiiden verkkojen, kuten WLAN:in käytön, joten sovellukseen suunniteltiin täysin yksilöllinen hyppiverkko. Tämän, tilaverkoksi kutsutun, sovelluksen testaamiseen käytettiin nk. demo-ympäristöä. Demo-ympäristön aikana listoihin merkattiin tilaa käyttävien ihmisten lukumäärä, ja tätä tietoa verrattiin järjestelmän mittaamiin arvoihin.

Työssä suunniteltu ja toteutettu järjestelmä onnistuu sen pääasiallisesta tehtävästä erinomaisesti. Kolmen eri työssä tutkitun anturin yhteistiedolla saadaan luotettava tieto tilan aidosta käyttöasteesta. Valituilla antureilla ei kuitenkaan pystytä arvioimaan tarkasti tilassa olevien ihmisten lukumäärää. Työn onnistunein osa on suunniteltu hyppiverkko, joka osaa korjata itsensä häirintätilanteissa palautuen normaaliin toimintaan. Hyppiverkossa on myös paljon laajennusvaraa sekä monipuolisuutta toimiakseen useassa erityyppisessä sovelluksessa.

Työn aikana suunniteltu järjestelmä toimii luotettavasti, mutta kaupalliseen tarkoitukseen järjestelmä ei ilman jatkokehitystä sovellu. Tätä jatkokehitystä on suoritettu opinnäytetyön ulkopuolella.

Kaikki projektin aikana kirjoitettu ohjelmakoodi, piirikaaviot ja suunnitellut piirilevyt ovat joitain opinnäytetyössä näkyviä poikkeuksia lukuun ottamatta määritelty luottamukselliseksi. Toteutuksesta esitellään sen toimintaperiaate ja siitä tehdyt mittaukset.

Asiasanat: tilankäyttö, käyttöaste, langaton, hyppiverkko, verkko, anturi, tilaverkko

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Information and Communications Technology
Embedded Systems and Electronics

MIIKKA MIKKONEN:

Space Network

Determining the utilization rate of a space and sending this information via a custom multi-hop wireless network

Bachelor's thesis 57 pages, appendices 17 pages
December 2012

This thesis was commissioned after a discovery was made in Tampere University of Applied Sciences; reserved classrooms were not efficiently used and a solution of determining whether these spaces were in use or not, had to be found. Therefore a system that can detect whether the space is in use or not and a way to send this information wirelessly were to be implemented. If possible, a secondary objective was to determine how many people would occupy the space at any given time. A troublesome decree prevents the use of cameras in these spaces, so one part of the thesis is to determine which sensors are most suitable for the required task. The costs of implementing this system should be kept at a minimum, due to the possibility that hundreds of devices could be needed to cover the entire institute. This prevents the use of available wireless networks, such as Wi-Fi, so a completely unique and custom wireless network was developed for this system. This system, called Space Network, was tested using a test environment set up in the 3rd floor of wing A in TAMK. People were required to record their usage of these spaces in lists, and this information was then compared with the measurements sent by the system.

The designed and implemented system succeeds in its primary task of determining the utilization of spaces outstandingly. Using three different sensors in tandem, a reliable state of the space could be determined. However these sensors cannot determine reliably how many people are situated in these spaces at given times. The most successful part of this thesis however is the wireless network. It is truly self-healing, meaning it can repair itself in case other wireless signals disturb its operation. It is also very scalable and versatile to use in other systems and applications.

Even though the system already works well, it is not yet ready for commercial use. Future improvements in the system should be made to further improve the wireless network with the implementation of channel-hopping to broaden the used spectrum from one fixed frequency. This and other improvements, especially in the field of energy-efficiency should be made before the possible commercializing of the system.

All written code, circuit diagrams and designed circuit boards, other than those presented in this thesis, are defined as confidential. Only the working principle and measurements made of the final system are presented in this thesis.

Key words: space, wireless, hop, network, utilization rate, sensors

SISÄLLYS

1	JOHDANTO.....	7
2	TILAVERKON SUUNNITTELU	8
2.1	Anturipaketti	8
2.2	Keräin.....	8
2.3	Tilaverkon toimintaperiaate	9
2.4	Mikrokontrolleri.....	9
2.4.1	A/D-muunnin	11
2.4.2	Keskeytykset	11
2.4.3	SPI-väylä.....	13
2.5	Lähetinvastaanotinpiiri	15
2.5.1	Enhanced Shockburst™	15
2.5.2	Multiciever™ ja putket	17
2.5.3	Kytkeminen mikrokontrolleriin	17
2.5.4	Testaaminen	18
3	ANTURIT	19
3.1	Liikeanturi.....	19
3.2	Äänianturi	20
3.2.1	Kytkeä	21
3.2.2	Mittaukset.....	22
3.3	Valoisuusanturi	23
3.3.1	Kytkeä	23
3.3.2	Mittaukset.....	24
4	VERKON SUUNNITTELU.....	25
4.1	Verkon topologia	25
4.1.1	Tähtitopologia	25
4.1.2	Puutopologia	26
4.1.3	Mesh-topologia	27
4.1.4	Hybriditopologia	27
4.2	Verkon looginen osoitteellisuus.....	28
4.3	Verkkoon liittyminen	29
5	VERKON TESTAUS.....	31
5.1	Toimintaetäisyys	31
5.2	Toimintanopeus	32
6	PROTOTYYPPI JA DEMO-YMPÄRISTÖ	34
6.1	Testausjärjestelyt	35
6.2	Mittaustulokset.....	36
6.2.1	Liike	36
6.2.2	Valoisuus.....	36
6.2.3	Ääni	36
7	YHTEENVETO	38
	LÄHTEET	39
	LIITTEET	41
	Liite 1. nRF24L01-esimerkkikytkentä	41
	Liite 2. Viestipaketin lähettämiskoodi	42
	Liite 3. SPI-väylän funktioita	43
	Liite 4. nRF24L01:n spektri	44
	Liite 5. Äänimittauspöytäkirja.....	45
	Liite 6. Valoisuusmittauspöytäkirja.....	46
	Liite 7. Valoisuuskuvaaja	47

Liite 8. Hybriditopologian esimerkki	48
Liite 9. Verkon lähetysfunktio.....	49
Liite 10. Verkon toimintaetäisyys siirtonopeudella 250 kbps	50
Liite 11. Verkon toimintaetäisyys siirtonopeudella 2 Mbps	51
Liite 12. Verkon viestin välittämiseen kuluva aika nopeudella 250 kbps.....	52
Liite 13. Verkon viestin välittämiseen kuluva aika nopeudella 2 Mbps	53
Liite 14. Prototyypin kytkentäkaavio	54
Liite 15. Liikeantureiden data	55
Liite 16. Valoisuusantureiden data	56
Liite 17. Ääniantureiden data	57

ERITYISSANASTO

CRC	Cyclic Redundancy Check, 50-luvulla kehitetty nykyään erittäin yleinen virheentarkistusmenetelmä, jossa vastaanotettu viesti jaetaan tunnetulla tarkistusavaimella. Jos jakotulos on sama kuin viestin mukana tuleva CRC arvo, on viesti tulkittu virheettömäksi.
EEPROM	Electronically Erasable Programmable Read-Only Memory, haihtumaton puolijohdemuisti, jota ei tarvitse tyhjentää lohkoittain kuten Flash-muistia, vaan voidaan kirjoittaa tavu kerrallaan. EEPROM on paljon hitaampaa kuin Flash-muisti.
ISM	Industrial, Scientific and Medical, maailmanlaajuinen radio- taajuuskaista, jonka käyttö ei vaadi erillistä lupaa. Alun perin kehitetty teollisuuden, tieteen ja lääketieteen käyttöön.
Node	Elektroninen laite, joka on kytkeytynyt verkkoon ja kykenee lähettämään, vastaanottamaan tai välittämään viestin verkossa.
RGB LED	Red Green Blue Light-Emitting Diode, diodikotelointi, jossa on kolme eriväristä valodiodia, joita sekoittamalla voidaan luoda mikä tahansa väri.
USB	Universal Serial Bus, yleisesti käytetty ja standardoitu sarjavyölääarkkitehtuuri.

1 JOHDANTO

Opinnäytetyössä rakennetaan, ohjelmoidaan sekä testataan anturiverkko, jonka tarkoituksena on tunnistaa luokkahuoneiden käyttöaste, eli ovatko tilat käytössä vai vapaina. Työn tilaajalla, Tampereen Ammattikorkeakoululla, on ollut tehottomuutta varattujen luokkien käytössä. Vuonna 2012 alkaneen H-siiven remontin takia luokkia on vähemmän käytössä kuin koskaan, joten tällä anturiverkolla olisi tarkoitus tutkia luokkien aitoa käyttöastetta.

Työn rajoituksena ovat oppilaitoksen säädökset, joiden mukaan luokkiin ei saa asentaa kameroita, joten luokissa on käytettävä joitain muita antureita. Työn tarkoituksena on siis myös tutkia, mitkä anturit toimivat parhaiten käyttöasteen mittaamisessa sekä mahdollisesti arvioimaan, kuinka monta ihmistä luokassa milläkin hetkellä on. Tätä varten testaamiseen käytetään demo-luokkia, joissa anturit keräävät tietoa luokan tapahtumista.

Anturiverkon tulee toimia siten, että antureita voidaan sijoittaa koko oppilaitoksen kaikkiin luokkiin niin, että laitteet eivät häiritse toisiaan tai muuta tietoliikennettä. Antureiden asentamisesta tulee tehdä myös vaivatonta, jotta tämän tyyppinen järjestelmä olisi käytännöllistä asentaa esimerkiksi TAMK:n kokoiseen ympäristöön.

2 TILAVERKON SUUNNITTELU

Tilaverkon suunnittelu aloitettiin verkon pakollisten ominaisuuksien määrittämisellä. Tilaverkon tulisi suoriutua luokkahuoneen tilankäytön havaitsemisesta häiritsemättä muita langattomia yhteyksiä. Mahdollisissa virhetilanteissa sen tulisi pystyä korjaamaan itsensä automaattisesti. Tämän lisäksi tilaverkon tulisi olla helppo asentaa ja edullinen valmistaa.

2.1 Anturipaketti

Luokkahuoneissa antureita sisältävää laitetta kutsutaan tästä lähtien anturipaketiksi. Jokaiseen luokkahuoneeseen sijoitetaan yksi tai enemmän anturipaketteja, jotka mittavat ympäristöään siihen asennetuilla antureilla. Jokaisella laitteella on sille yksilöllinen tunniste, sarjanumero, joka voi esiintyä vain yhdessä verkkoon liittyneessä laitteessa. Jokainen laite toimii myös anturiverkon solmupisteenä, jonka kautta muut verkon viestit välitetään kohti oikeata määränpäättä.

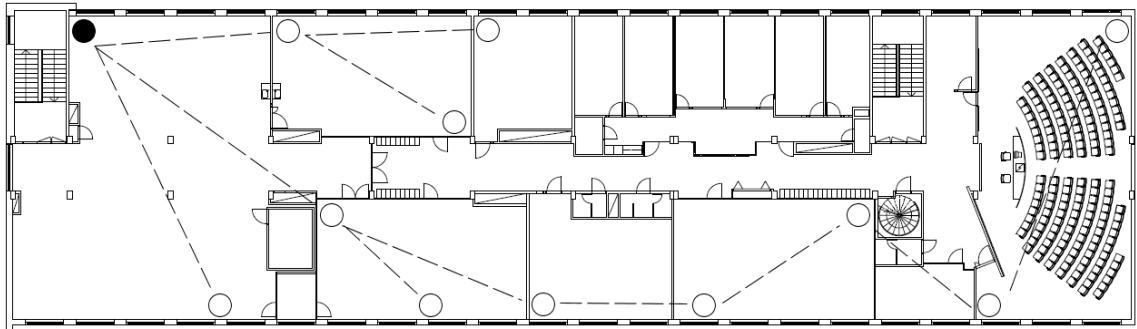
Anturipakettien tulee suoriutua tilankäytön havaitsemisesta anturien avulla. Anturipaketit eivät tulkitse mitattuja anturitietoja, vaan vain lähettävät ne eteenpäin. Anturipaketin tulisi myös pystyä vastaanottamaan, lähettämään ja välittämään verkon viestejä.

2.2 Keräin

Keräimen tehtävä on kerätä kaikkien verkossa olevien laitteiden lähettämä tieto, ja lähettää tämä tieto eteenpäin palvelimelle. Palvelimen tehtävänä on analysoida tieto, ja esittää tämä loppukäyttäjälle. Keräin voi myös ottaa yhteyden mihin tahansa verkkoon liittyneistä laitteista. Tätä opinnäytetyötä varten ei palvelinosaa käsitellä, vaan työ keskittyy antureiden ja hyppyverkon suunnitteluun. Kaikki kerätty tieto siirretään suoraan tietokoneelle, jossa tätä tietoa voidaan käsitellä ja analysoida. Keräimiä sijoitetaan anturipakettien määrän mukaan. Alustava idea on, että näitä keräimiä voitaisiin sijoittaa esimerkiksi joka siiven eri kerroksiin. Kuinka tiheään tai harvaan näitä keräimiä voidaan sijoittaa, määritetään käytännön testeillä.

2.3 Tilaverkon toimintaperiaate

Tilaverkon toimintaa kuvaa kuva 1, jossa TAMK:n A-siiven kolmannen kerroksen jokaiseen luokkahuoneeseen on sijoitettu yksi tai useampi anturipaketti. Kuvassa anturipakettia esittää läpinäkyvä ympyrä, ja keräintä musta ympyrä. Langatonta tiedonsiirtoa näiden välillä kuvaa katkoviivat verkon jäsenien välillä.



Kuva 1 Anturiverkon toimintaperiaate

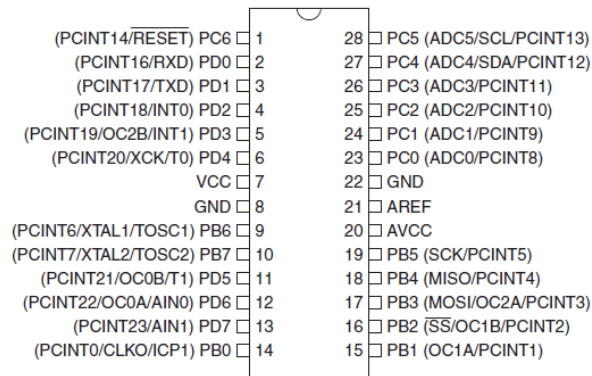
Verkosta tehdään täysin dynaaminen tarkoittaen sitä, että käyttäjän ei tarvitse asennuksessa koskea laitteen asetuksiin millään tavoin. Käyttäjän tulee vain kertoa palvelimelle, missä mikäkin laite fyysisesti sijaitsee.

Tätä ”linkkaamista” ei tätä työtä varten suoriteta, mutta sen idea on seuraava: Käyttäjä lukee laitteen sarjanumeron tälle tarkoitettulla laitteella, esimerkiksi tabletilla. Tälle sarjanumerolle asetetaan jokin yksilöllinen tunnistus, esimerkiksi A3-20_1, joka tarkoittaisi tilan A3-20 ensimmäistä anturikeräintä. Tämä asetettu nimi sekä laitteen sarjanumero lähetetään palvelimelle, joka tallentaa tiedon. Keräin lähettää anturipaketilta vastaanotetun anturidatan ja tämän sarjanumeron palvelimelle. Näin ollen palvelin voi ilmoittaa loppukäyttäjälle anturidatan tulleen sijainnista A3-20_1, eikä anturipaketin 32-bittisestä sarjanumerosta. Jos samassa tilassa on useita anturipaketteja, voidaan tämän nimeämisen avulla yhdistää samasta tilasta tulneiden anturipakettien tiedot.

2.4 Mikrokontrolleri

Antureista saatava data on joko digitaalista tai analogista. Data on tulkittava sekä siirrettävä lähetinvastaanotinpiirille, joka lähettää sen eteenpäin. Mikrokontrolleriksi valittiin Atmel ATmega328p (kuva 2), koska sen ominaisuudet ovat sovellukselle oivalliset ja

kontrollerisarjan ominaisuudet ovat tuttuja aikaisemmasta kokemuksesta. Kontrolleri sisältää kaikki tarpeelliset ominaisuudet, ja siinä on myös hieman laajennusvaraa.



Kuva 2 Atmel ATmega 328p -kontrollerin 328 nastajärjestys [1, s. 2]

Kontrollerin nastoista käyttöön otetaan seuraavat:

- PD0 ja PD1, joita käytetään varsinkin testausvaiheessa tulostamaan tietokoneelle tietoa laitteen tekemisistä.
- PD2 ja PD3, jotka ovat ulkoisia keskeytyspinnejä.
- PB1 – PB5, joihin kytketään ulkoinen lähetinvastaanotinpiiri.
- PC0 ja PC1, joihin kytketään antureiden analogiset mittapisteet.
- PD3 – PD5, joihin kytketään RGB-ledi näyttämään käyttäjälle laitteen tilaa.

Laitteen suunnittelun helpottamiseksi käytetään Arduino Nano -mikrokontrollerialustaa (kuva 3), jossa on valittu mikrokontrolleri, sekä kaikki tarvittavat oheiskomponentit, kuten jänniteregulaattori ja USB-sarjamuunnin. Työssä ei kuitenkaan käytetä Arduinon kehitysympäristöä, vaan kaikki ohjelmointi tehdään GCC-kääntäjää käyttävällä Atmel AVR Studiolla.



Kuva 3 Arduino Nano -mikrokontrollerialusta [2]

Arduino-kehitysalustaa käytettäessä on otettava huomioon, että käyttöjännite on viisi volttia. Mikrokontrolleriin on kytketty ulkoinen 16 MHz:n kideoskillaattori, joka määrittää kontrollerin kellotaajuuden. Mikrokontrolleriin on myös esiohjelmoitu Arduinon

oma bootloader, joka mahdollistaa kontrollerin ohjelmoinnin USB-portin kautta. Tämän bootloaderin käynnistys kestää muutaman sekunnin, joten kontrolleri ei välittömästi käynnisty virran kytkemisen jälkeen. Kehitysalustassa on myös FTDI:n valmistama USB-sarjamuunnin, joka sisältää 3,3 voltin ulostulon. Tätä voidaan käyttää ulkoisten laitteiden käyttöjännitteenä.

2.4.1 A/D-muunnin

Atmel ATmega328p -mikrokontrollerissa on 8-kanavainen kymmenbittinen A/D-muunnin. Koska elektronisten komponenttien toiminta muuttuu hieman lämpötilan funktiona, on mikrokontrollerissa oma lämpötila-anturi, jolla lämpötilan muutoksen aiheuttamia häiriöitä kompensoidaan. [1, s. 252.] Kun A/D-muunnos on valmis, aiheuttaa se keskeytyksen. Tässä sovelluksessa kuitenkin käytetään niin sanottua blokkausmetodia, jolloin tulosta jäädään odottamaan. Tämä johtuu siitä, että mittauksia suoritetaan anturikohtaisesti useita sarjassa tarkemman ja luotettavamman mittaustuloksen saavuttamiseksi.

Referenssijännitteeksi valittiin sisäisen 1,1 voltin sijasta ulkoinen käyttöjännite. Näin ollen A/D-muunnoksessa mitattavan jännitteen askel, eli resoluutio on

$$\frac{\text{max value} - \text{min value}}{2^{10}} = \frac{5 \text{ V}}{1024} = 4.9 \text{ mV}$$

ja suurin mahdollinen mittausrvirhe on kaksi kertaa tämä arvo eli 9.8 mV [1, s. 252].

2.4.2 Keskeytykset

Atmel ATmega328p -mikrokontrollerissa on useita keskeytyslähteitä. Näistä kattava lista on piirin datalehden sivuilla 66–67. Näillä sivuilla sijaitsevasta taulukosta on poimittu taulukon 1 sisältämät keskeytykset lähempään tarkasteluun.

Taulukko 1 ATmega328p keskeytykset [1, s. 66–67]

VectorNo.	Program Address	Source	Interrupt Definition
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
19	0x0024	USART, RX	USART, Rx Complete
21	0x0028	USART, TX	USART, Tx Complete

Keskeytykset INT0 ja INT1 ovat ulkoisia keskeytyksiä. Nämä keskeytykset voidaan ohjelmoida liipaistumaan laskevalla reunalla, nousevalla reunalla tai muutoksen havaittuaan. Koska pienimmän vektorinumeron keskeytykset ovat prioriteetiltaan korkeammalla, suoritetaan nämä ulkoisen signaalin aiheuttamat keskeytykset ensin [1, s. 10].

Taulukon seuraavat keskeytykset ovat laskureiden aiheuttamia. Laskuri 1 on 16-bittinen ja laskurit 0 ja 2 ovat 8-bittisiä. Näitä voidaan käyttää useaan eri tarkoitukseen, mutta tässä sovelluksessa näiden tarkoitus on aiheuttaa keskeytys, aina kun laskuri vuotaa yli, eli sen arvo pyörii ympäri. Kuinka kauan tähän menee, riippuu laskurin esijakajasta, prosessorin kelloaajuudesta, laskuriin asetetusta alkuarvosta sekä laskurin bittilukumäärästä.

Tässä sovelluksessa käytetään keskeytystä Timer/Counter1 Overflow, joka määritetään laukaistumaan kerran sekunnissa. Tätä tietoa käytetään hyväksi ajoittamaan anturipaketti lähettämään tietyin väliajoin anturitietonsa. Jotta voitaisiin saada keskeytys kerran sekunnissa, asetetaan esijakajaksi 256. Näin ollen 16 MHz:n kelloaajuudella tapahtuu

$$\frac{16000000 \frac{1}{s}}{256} = 62500 \text{ laskurin askelta sekunnissa.}$$

Koska laskuri on 16-bittinen, mahtuu siihen $2^{16} = 65536$ kellon ”naksasta”. Kun laskuriin asetetaan aina sen pyörähdettyä ympäri arvo $65536 - 62500 = 3036$, laukaistuu keskeytys tasan sekunnin välein.

Valitun mikrokontrollerialustan kideoskillaattori on tarkkuudeltaan noin 20 ppm (0.002 %). Näin ollen päivässä virhe voi olla enintään

$$24 \text{ h} \cdot 60 \text{ min} \cdot 60 \text{ s} \cdot 0.000002 = 1.76 \text{ s}$$

Tämä virhe ei järjestelmään vaikuta, koska ajastimen tarkoitus on ainoastaan pitää huoli siitä, että anturipaketti lähettää tietyin väliajoin mittaustietonsa keräimelle. Ohessa on koodinpätkä Timer/Counter 1:sen keskeytysohjelmasta. Keskeytysohjelma kasvattaa sekunnin välein timer1s-nimisen muuttujan arvoa, ja vilkuttaa laitteen RGB-lediä valitulla värillä, jos ledin vilkkutila on asetettu aktiiviseksi.

```
ISR(TIMER1_OVF_vect)
{
    // Reload timer with pre-calculated value to match 1 Hz
    TCNT1 = 3036;
    // Increment 1s time
    timer1s++;
    // Blink LED with active color if blinking is set on
    if (blinkStatus == BLINKING) TOGGLEBIT(LED_PORT, activeColor);
}
```

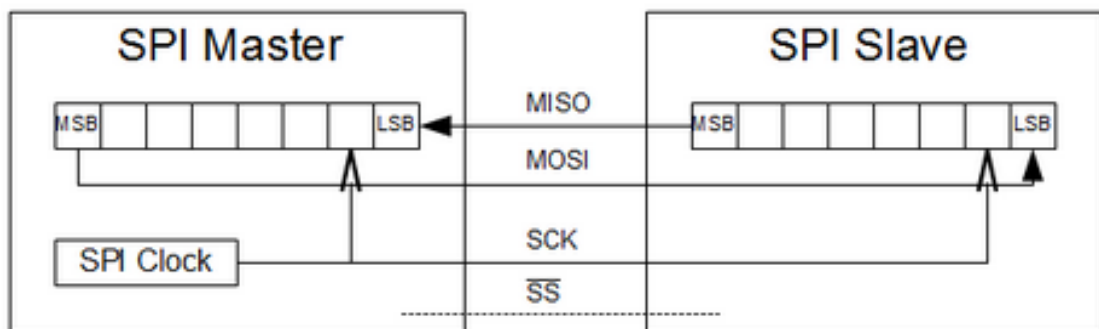
USART-sarjaliikenneyksikön (Universal Synchronous/ Asynchronous Receiver/ Transmitter) keskeytykset liipaistuvat, aina kun USART on vastaanottanut tai lähettänyt yhden tavun. Näitä ominaisuuksia käytetään hyväksi, kun lähetetään tietoa testauksen aikana tietokoneelle. Koska USART voi hoitaa vain tavun kerrallaan, implementoitiin koodiin rengaspuskuri hoitamaan lähetys- ja vastaanottopuolta. Näin ollen laite voi käsitellä puskurin rajoittaman määrän tietoa kerrallaan. Tämä ominaisuus kytketään pois päältä lopullisessa versiossa, koska tiedon lähettäminen sarjaportin kautta on hidasta. Tämän takia kaikki koodinpätkät, joissa käytetään sarjaporttia tulosteisiin, on määritelty kääntäjälle komennoilla, jotka voidaan estää yhtä riviä muokkaamalla.

2.4.3 SPI-väylä

Serial Peripheral Interface -väylän (SPI) avulla otetaan mikrokontrollerilla yhteys laitteen lähetinvastaanotinpiiriin. SPI-väylä on synkroninen sarjaliikenneväylä, jossa sekä lähettäjä että vastaanottaja käyttävät samaa tahdistussignaalia bittien tunnistamiseen. Yleensä, kuten tässäkin sovelluksessa, SPI-väylää käytetään lyhyen matkan tiedonsiirtoon piirien välillä, jotka sijaitsevat samalla piirilevyllä tai ainakin samassa laitteessa. SPI-väylässä on aina yksi master-laite ja yksi tai useampi slave-laite. Master- ja slave-laite molemmat lähettävät ja vastaanottavat dataa samanaikaisesti, jota kutsutaan Full-

duplex tiedonsiirroksi. Master-laitteen tehtävänä on tuottaa tahdistussignaali tiedonsiirtoa varten.

Jos piirinvalintalinja SS (Slave Select) on aktiivisena, jokaisella tahdistussignaalin (SCK) laskevalla tai nousevalla reunalla Master-laite siirtää siirtorekisteriin kirjoitetun tavun eniten merkitsevän bitin (MSB) linjasta MOSI (Master-Out-Slave-In) slave-laitteelle. Samanaikaisesti slave-laite lähettää linjasta MISO (Master-In-Slave-Out) oman eniten merkitsevän bitin (kuva 4). Siirtorekistereiden sisältöä siirretään kuvan 4 tapauksessa joka lähetyksen jälkeen yksi bitti vasemmalle. Tämä toistetaan 8 kertaa yhden tavun lähettämiseksi. Näin ollen molemmat laitteet vastaanottavat toisen laitteen tiedon ja lähettävät oman tietonsa samanaikaisena. [3, s. 151–152.]



Kuva 4 SPI-väylän toimintaperiaate [4]

Käytettävä lähetinvastaanotinpiiri lähettää SPI-väylän kautta aina ensimmäisenä paluuviestinä rekisterin 0x07 (STATUS) tilan, joka sisältää tiedon vastaanotetuista ja lähetetyistä viesteistä [5, s. 50].

SPI-väylään on mahdollista liittää useita eri laitteita. Slave-laite voi olla aktiivisena vain, jos sen piirinvalinta SS (Slave Select) on aktiivisena. Jokaiselle slave-laitteelle pitää kytkeä oma piirinvalintalinjansa, jolla master valitsee valitun tai valitut slave-laitteet. Tässä sovelluksessa käytetään vain yhtä slave-laitetta, joka on lähetinvastaanotinpiiri.

SPI-väylä voidaan alustaa eri tavoilla. Tätä sovellusta varten väylä asetetaan 4 MHz:n taajuudelle, joka sopii valitulle lähetinvastaanotinpiirille (ks. 2.3).

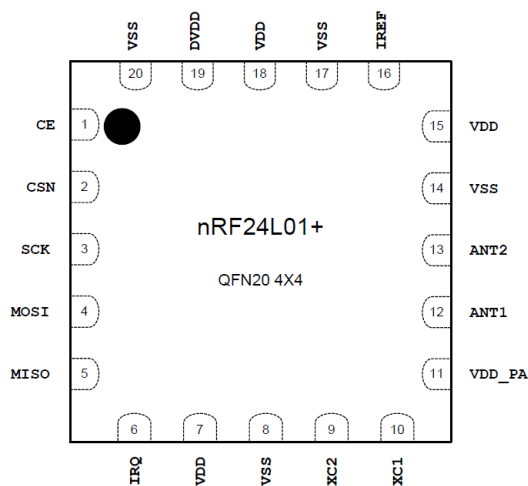
```

void SPI_MasterInit(void)
{
    // Set MOSI, SCK, SS and CSN to output
    DDR_SPI = BIT(MOSI) | BIT(SCK) | BIT(SS) | BIT(CSN);
    // Enable SPI, Master mode @ F_CPU / 4 = 16 MHz / 4 = 4 MHz
    SPCR = BIT(SPE) | BIT(MSTR);
}

```

2.5 Lähetin vastaanotinpiiri

Lähetin vastaanotinpiiriksi valittiin Nordic Semiconductorsin nRF24L01+ (kuva 4). Piirin valintaan vaikutti sen edullinen hinta sekä toiminta 2,4 GHz:n vapaalla ISM-taajuusalueella.



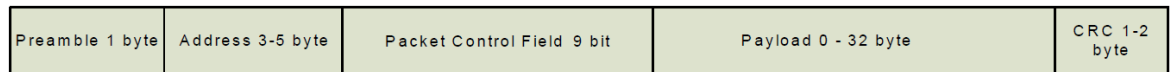
Kuva 4 Lähetin vastaanotinpiiri [5, s. 10]

Järjestelmän suunnittelun ja rakentamisen helpottamiseksi käytetään iteastudiosta ostettua moduulia, jossa piirin oheiskomponentit ovat valmiiksi sijoitettuna (liite 1). Moduulin kytkentä ja piirilevy ovat lähetin vastaanottimen datalehden sovellusohjeen mukaisia [5, s. 64].

2.5.1 Enhanced Shockburst™

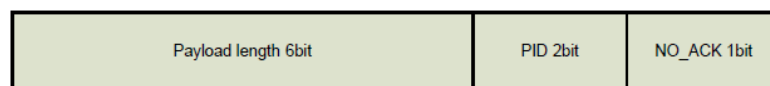
Piiri käyttää 2-tasoista GFSK-modulaatiota, jossa loogista ykköstä vastaa tietty taajuus ja loogista nollaa toinen taajuus. Signaali suodatetaan Gaussin suotimella, joka tekee signaalin muodosta pyöreämmän ja näin ollen huomattavasti kaventaa lähetetyn signaalin spektrin leveyttä.

Lähetinvastaanotinpiirin lähettämän sanoman kehyksen rakenne on kuvan 5 mukainen. Alussa on tahdistustavu (preamble), joka on joko 01010101 tai 10101010. Tätä seuraa joko kolmen tai viiden tavun mittainen osoite, joka tätä sovellusta varten on asetettu viiteen tavuun. Kolmas kohta on yhdeksän bitin paketinhallintakenttä.



Kuva 5 Lähetetyn viestin rakenne [5, s. 28]

Paketinhallintakentän (kuva 6) alussa on kuuden bitin mittainen kenttä, joka ilmaisee viestin tavulukumäärän. Tätä seuraa kahden bitin PID-kenttä (Packet Identity), jolla voidaan tulkita, onko vastaanotettu viesti uusi vai uudelleenlähetetty. Viimeisenä on yhden bitin NO_ACK-lippu, joka ilmaisee kuitataanko vastaanotettu viesti saapuneeksi vai ei.



Kuva 6 Paketinhallintakentän rakenne [5, s. 28]

Paketinhallintakentän jälkeen on 0-32 tavun viestisisältö, jonne haluttu lähetettävä tieto kirjoitetaan. Viimeisenä viestissä on joko yhden tai kahden tavun pituinen CRC-virheentarkistuskenttä, jolla varmistetaan, että vastaanotettu viesti on virheetön. Jos CRC-tarkistus ei täsmää, poistaa lähetinvastaanotinpiiri viestin automaattisesti. [5, s. 28–30.]

Lähetinvastaanotinpiiri käyttää viestien lähettämiseen verkossa omaa Enhanced Shockburst™ -protokollaa. Tämän protokollan ominaisuuksiin kuuluu automaattinen paketinhallinta, johon kuuluvat mm. automaattinen vahvistusviesti (Auto Acknowledgement, ACK) sekä automaattinen uudelleenlähetys (Auto Retransmission, ART). Piirissä on myös 6 putkea (pipe), jotka mahdollistavat 1:6 tähtiverkon rakentamisen.

ACK-viesti on piirin sisäinen toiminto, joka automaattisesti lähettää vahvistusviestin, jos sen vastaanottama viesti on virheetön. Kun ACK-viesti on valittu aktiiviseksi, lähetävä piiri siirtyy automaattisesti viestin lähettämisen jälkeen kuuntelutilaan. Jos tämä vahvistusviesti ei saavu tietyllä aikavälillä tulkitaan, että viesti ei saavuttanut määrän-

päätään. Tässä sovelluksessa käytetään myös ei-vahvistettuja viestejä, jolloin viestiin kirjoitetaan NO_ACK -lippu, joka kertoo vastaanotetulle laitteelle, että tätä viestiä ei vahvisteta. Automaattista uudelleenlähetystä ohjataan rekisterin 0x04 (SETUP_RETR) sisällöllä. Bitit 7:4 ilmaisevat uudelleenlähetyksen välillä pidettävän väliajan pituutta ja bitit 3:0 ilmaisevat uudelleenlähetyksen lukumäärää.

Tässä sovelluksessa käytetään dynaamista paketin pituutta, joka on kuitenkin rajattu 24 tavuun, ACK-viestiä ja maksimimäärää (15) uudelleenlähetyksetkoja, joiden välinen odotusaika on 500 mikrosekuntia.

2.5.2 Multiciever™ ja putket

Multiciever™ on piirin ominaisuus, jolla mahdollistetaan kuusi rinnakkaista vastaanot-toputkea, joista jokaisella on oma yksilöllinen fyysinen osoite. Näin ollen laite voi yhtä aikaa kuunnella kuutta eri fyysistä osoitetta. Jokaista kuutta putkea voi ohjata erikseen eri asetuksilla, mutta niille ovat yhteisiä valittu CRC-tarkistustapa, osoitteen pituus, toimintataajuus, lähetysnopeus ja viestin vahvistus (ACK).

2.5.3 Kytkeminen mikrokontrolleriin

Lähetinvastaanottimen suurin käyttöjännite on 3,6 voltia. Käyttöjännite piirille on otettava 5 voltin jänniteregulaattorin sijasta FTDI-piirin 3,3 voltin ulostulosta (ks. 2.2). Piirin datalinjat toimivat siitä huolimatta 5 voltin logiikkatasoilla, joten nämä voidaan suoraan kytkeä valittuun mikrokontrolleriin. [5, s. 12.]

Piiriä ohjataan SPI-väylän kautta maksimissaan 10 MHz:n taajuudella [5, s.50]. Piirille kytketään ohjausnastat SCK, MOSI ja MISO. Näiden lisäksi lähetinvastaanotinpiirille on annettava ohjaussignaali CE (Chip Enable), jolla piirin valittu tila asetetaan aktiiviseksi, ja CSN (Chip Select Not), jolla erotetaan SPI-väylän kautta annetut käskyt toisistaan.

Esimerkki funktiosta, jolla lähetinvastaanotin asetetaan lähetystilaan

```
void TX_Mode(void)
{
    // Set CE low
    CLEARBIT(PORT_SPI, CE);
    // Clear bit PRIM_RX
    SPI_WriteRegister(CONFIG, SPI_ReadRegister(CONFIG) & ~PRIM_RX);
    // Set CE high
    SETBIT(PORT_SPI, CE);
}
```

Alhaalla aktiivinen IRQ-linja kytketään mikrokontrollerin INT0-keskeytystulonastaan. Linja aktivoituu, kun piiri on vastaanottanut viestin tai piirin lähettämä viesti on onnistunut tai epäonnistunut. Parempaan virrankulutuksen saavuttamiseksi voidaan mikrokontrolleri herättää unitilasta tämän keskeytyksen avulla.

Liitteessä 2 on lähetinvastaanotinpiirille kirjoitettu ohjelmakoodi viestin lähettämistä varten, ja liitteessä 3 on sen kutsuman SPI-väylän ohjelmakoodi kommentteineen. Anturi-paketti ja keräin kutsuvat tätä funktiota, kun lähetysosoite on kirjoitettu lähetinvastaanotinpiirille, ja haluttu viesti on kirjoitettu mikrokontrollerin txBuff-puskuriin.

2.5.4 Testaaminen

Piirin toimintaa testattiin lähettämällä purskeita eri kanavilla ja siirtonopeuksilla 24 tavun kokoisilla viestisisällöillä. Liitteessä 4 on nähtävissä lähettimen spektri kanavalla 5 siirtonopeudella 250 kbps. Taulukossa 2 esitetään mitattu ja datalehden ilmoittama kaistanleveys ja keskitaajuus.

Taulukko 2 Mittaustulokset verrattuna ilmoitettuihin arvoihin

Parametri	Ilmoitettu arvo	Mitattu arvo
Kaistanleveys	Alle 1 MHz	700 kHz
Keskitaajuus	$2400 \text{ MHz} + (\text{kanavanumero} * 1 \text{ MHz}) = 2405 \text{ MHz}$	2405,09 MHz

Mittaustuloksilla todettiin, että lähetinvastaanotinpiiri oheiskomponentteineen toimii ilmoitetulla tavalla, joten tätä kytkentää voidaan käyttää järjestelmän toteutuksessa.

3 ANTURIT

Luokkahuoneen tilankäytön havaitsemisen mahdollistaa eri parametreja ympäristöstään havaitsevat anturit. Laitetta suunniteltaessa mietittiin monia eri anturivaihtoehtoja, mutta testien ja suunnittelun jälkeen laitteeseen valittiin liike-, ääni- ja valoisuusanturit. Liikeanturi havaitsee huoneessa tapahtuneen liikkeen, äänianturi havaitsee luokassa olevat ääniaallot ja valoisuusanturi havaitsee ovatko luokkahuoneessa valot päällä vai pois päältä. On syytä olettaa, että näiden antureiden yhteistoiminnalla luokan käytön havaitseminen olisi luotettavaa.

3.1 Liikeanturi

Luokassa tapahtuva liike havaitaan pyrosähköisellä infrapuna-anturilla (Pyroelectric Infrared, PIR). Pyrosähköisellä aineella on ominaisuus luoda hetkellinen jännite, kun ainetta lämmitetään. Kun lämpötila pysyy vakiona, jännite hiljalleen katoaa vuotovirran takia. Pyroelektristä anturia voidaan siis kuvata kondensaattorina, jota voidaan ladata lämpösäteilyn avulla. [6, s. 32–113.] PIR-anturi ei sinänsä mittaa infrapunasäteilyn määrää, vaan sen muutoksen nopeutta. Kun lämmin kappale liikkuu anturin ohi, tapahtuu lämpösäteilyn määrässä hetkellinen, nopea muutos. [7.] Tällä tavoin havaitaan liike, joka tämän anturin tarkoituksena onkin.

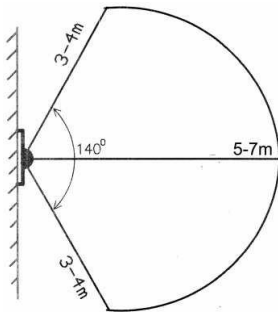
Anturin aiheuttama jännite on vahvistettava, jotta mikroprosessori voi tämän signaalin havaita. Aina kun signaali on aktiivisena, tulkitaan, että anturin läheisyydessä on tapahtunut liikettä. Tämä signaali kytketään mikrokontrollerin keskeytysnastaan INT1. Näin ollen mikrokontrolleri voi havaita jokaisen liikkeen ilman, että tämän tarvitsisi kuunnella anturia kaiken aikaa.

PIR-anturiksi valikoitui RE200B. Anturi on kiinnitettynä moduuliin (kuva 7), joka sisältää anturin, sen signaalin vahvistavan BISS0001-piirin, sekä Fresnelin linssin anturin päälle moduulilevyyn asennettuna, joka kohdistaa tulevan säteilyn itse anturiin. Anturin signaalin vahvistava BISS0001-piiri myös ohjaa moduulin ulostuloa. Ulostulo on 3,3 V, kun liikettä on havaittu ja 0 V, kun liikettä ei ole havaittu.. Liikkeen havaittuaan signaali pysyy ylhäällä minimissään viisi sekuntia havainnon jälkeen. [8.]



Kuva 7 PIR – moduuli [8]

Anturin havaitsemiskulma on n. 140 astetta vaaka- ja pystyakselilla ja sen havaitsemisetaisyys vaihtelee 3–7 metrin välillä riippuen lämpölähteen sijainnista Fresnelin linssin suhteen (kuva 8).



Kuva 8 Fresnelin linssin havaitsemiskulma ja -etäisyys [8]

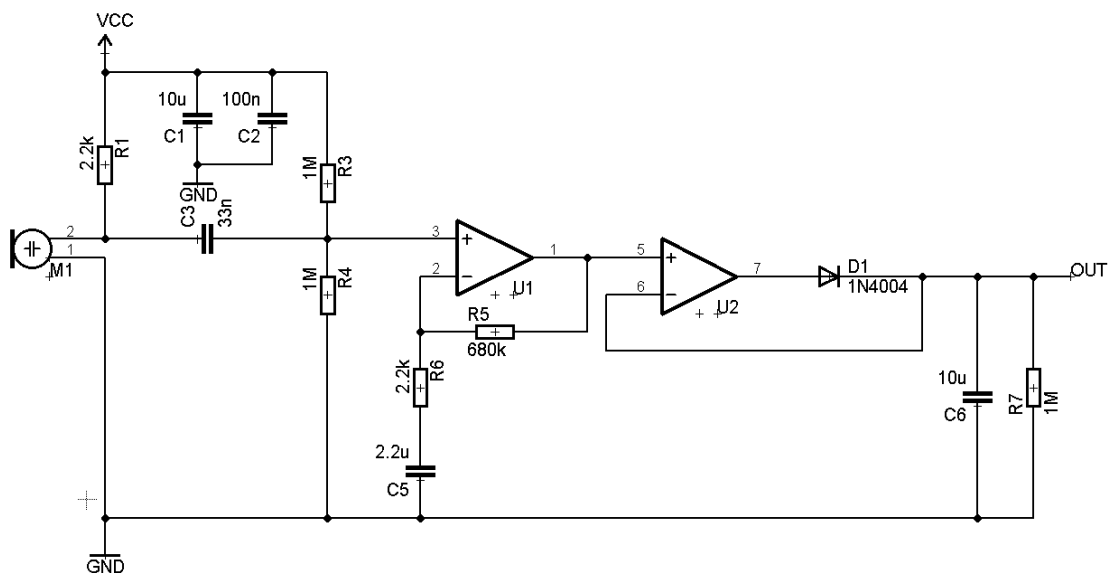
3.2 Äänianturi

Äänen havaitsemiseen käytetään elektreettimikrofonia. Elektreettimikrofonin toiminta perustuu mikrofoniin olevan kalvon liikkeeseen äänenpaineen johdosta. Kalvo toimii kuten kondensaattorin levy. Kalvon liikkeessä edestakaisin muuttuu mikrofoniin kapasitanssi, joka taas johtaa potentiaalimuutokseen eli jännitteeseen. Näin ollen kalvon liike äänenpaineen johdosta muodostaa sitä vastaavan jännitteen. Kalvon ja paikallaan pysyvän levyn välinen jännite on pidettävä korkealla verrattuna muodostuvan signaalin jännitteeseen. Kondensaattorimikrofoneissa yleensä tämä polarisaatiojännite on syötettävä erikseen, mutta elektreettimikrofoneissa tämän hoitaa polarisoitu materiaali, eli nk. elektreetti. Tämä elektreetti on pysyvästi varattu tiettyyn polarisaatiojännitteeseen, joten ulkoista jännitettä ei tarvitse siihen syöttää. [9, s. 1412–1413.]

3.2.1 Kytcentä

Valitun LD-MC-0905P -elektreettimikrofonin synnyttämä jännite on suurimmillaan vain muutamia millivolteja, joten signaali tulee vahvistaa. Anturiin rakennetaan myös verhoikäyräilmaisim, jotta äänisignaalin amplitudi on helpommin A/D-muuntimella luettavissa.

Kuvassa 9 esitellään suunnitellun äänianturin kytkentäkaavio. Kondensaattorit C_1 ja C_2 suodattavat käyttöjännitteen operaatiovahvistimille tasaiseksi. Kondensaattori C_3 poistaa mikrofonin M_1 muodostaman jännitteen tasakomponentin. Valitun elektreettimikrofonin lähtöimpedanssi on $2,2\text{ k}\Omega$, joten suurimman mahdollisen lähtöjännitteen saavuttamiseksi vastus R_1 on tämä sama $2,2\text{ k}\Omega$ [10, s. 1]. Vastukset R_3 ja R_4 lisäävät puolet käyttöjännitteestä sisääntulosignaaliin. Sisääntulon tasajännitekomponenttia ei kuitenkaan vahvisteta, koska tasajännitelinja on estetty kondensaattorilla C_5 . Analogiasignaali vahvistetaan operaatiovahvistimella U_1 , jonka vahvistuskerroin on $(1 + R_5/R_6) = 310$. Diodi D_1 ja operaatiovahvistin U_2 toimivat yhdessä nk. superdiodina, jonka kynnysjännite on lähes nolla. Tämä superdiodi sekä kondensaattori C_6 ja vastus R_7 toimivat yhdessä verhoikäyräilmaisimena, joka seuraa äänisignaalin amplitudia.



Kuva 9 Äänianturin kytkentäkaavio

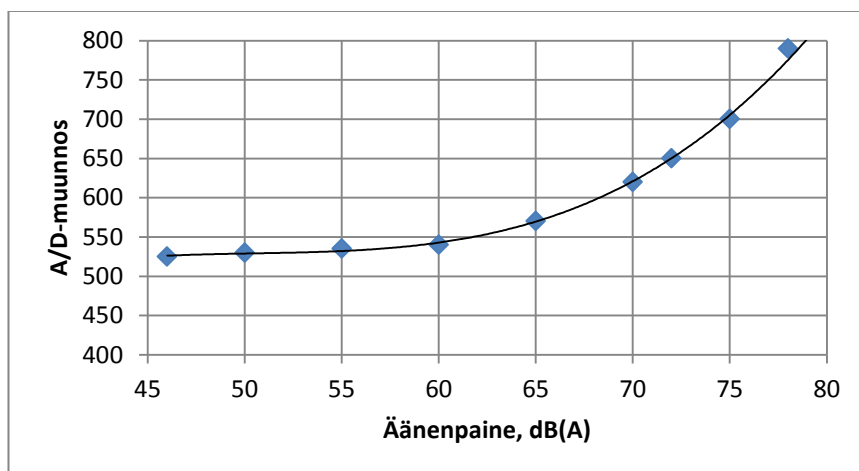
Näin ollen lähtöjännite on puolet käyttöjännitteestä, johon on summautunut mikrofonin vahvistettu äänisignaali sekä operaatiovahvistimen sisääntulon offset-jännite. Tämä tar-

koittaa sitä, että ulostulosta A/D-muuntimella mitattu arvo on lähellä arvoa 512, jos mitattava tila on hiljainen.

3.2.2 Mittaukset

Jotta anturin arvoista voidaan tehdä mitään päätelmiä, nämä tulokset on sidottava oikeisiin fysikaalisiin yksikköihin. Tähän käytettiin äänenpainetta mittaavaa TES1352H-desibelimittaria ja kaiuttimella tuotettua ääntä. Mittauksessa käytettiin äänilähteenä valkoista kohinaa, ja tätä ääntä kuunneltiin n. 20 cm päästä anturin testikokoonpanolla sekä tähän tarkoitukseen suunnitellulla äänenpainemittarilla ihmisen korvan kuuloa vastaavalla A-painotuksella [11].

Anturin lähtö liitettiin mikrokontrollerin A/D-muuntimen nasaan ja A/D-muunnostulos tehtiin 250 ms:n välein. Mittaustulokset ovat nähtävissä mittauspöytäkirjassa liitteessä 5 ja kuvassa 10.



Kuva 10 Äänianturin A/D-muunnostulos äänenpaineen suhteen

Tärkeimpänä mittaustuloksena on laitteen pienin havaitsema äänenpaine, eli laitteen herkkyys. Herkkyudeksi muodostui 46 dB(A), joka vastaa todella hiljaista keskustelua. Anturi pystyy havaitsemaan normaalin keskustelun metrien päästä, joka on riittävää tälle sovellukselle. Kohinataso liikkui A/D-muuntimen arvojen 517 ja 524 välillä.

3.3 Valoisuusanturi

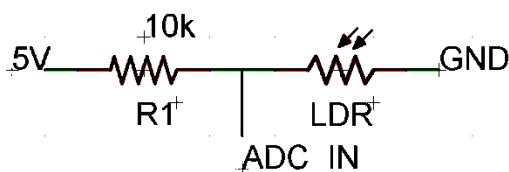
Valoisuusanturin tarkoituksena on mitata luokassa olevan valon määrää. Kun valoisuus muuttuu äkillisesti, tapahtuu luokkahuoneessa jonkinlainen muutos, eli valojen kytkeminen päälle tai pois päältä.

Valoisuusanturina käytetään LDR-valovastusta (Light Dependent Resistor), eli vastusta, jonka resistanssi pienenee valotehon kasvaessa. Valovastus käyttää hyväkseen suuri-resistanssisia puolijohteita. Valon osuessa puolijohteeseen sen energia siirtyy puolijohteen sitomille elektroneille ja tämän energian turvin ne vapautuvat ja siirtyvät johtavalle kaistalle. Näin ollen vastuksen johtavuus kasvaa, ja sen resistanssi pienenee. [12, s. 5–84.]

3.3.1 Kytkeä

Valovastuksen resistanssisiksi ilmoitetaan pimeässä tilassa vähintään 2 M Ω ja 10 Luksin valossa 16–50 k Ω . Mittauksilla todettiin vastusarvon olevan alle 2 k Ω hyvin valaistussa toimistoympäristössä ja yli 20 k Ω toimistoympäristössä päivällä, kun valot ovat pois päältä.

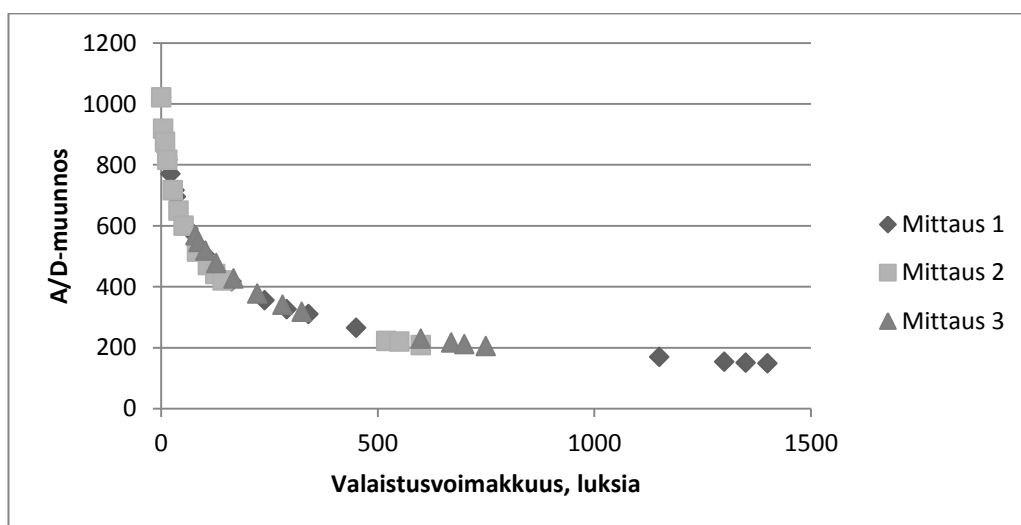
Näin ollen jännitteenjakoon valittiin vastusarvo näiden arvojen välistä, jotta eri valoisuustilanteet saataisiin mahdollisimman tarkasti mitattua. Valovastus liitetään mikrokontrolleriin kuvan 11 mukaisesti.



Kuva 11 Valovastuksen liittäminen mikrokontrolleriin

3.3.2 Mittaukset

Äänianturin tavoin valoisuusanturille suoritettiin mittaukset arvojen sitomiseen fysikaaliseen yksikköön. Valoisuuden yksikkö on luks ja tämän mittaamiseen käytettiin Hagnerin EC1-luksimittaria ja työssä käytettävää valoisuusanturia. Mittaukset suoritettiin TAMK:n pimeätilassa, jossa kattovalojen voimakkuutta voitiin portaattomasti säätää. Tämän lisäksi mitattiin arvot ikkunan läheisyydessä ulkoa tulevan valon vaikutuksien selvittämiseksi. Mittaustulokset ovat nähtävissä mittauspöytäkirjassa liitteessä 6 ja kuvassa 12.



Kuva 12 Valoisuusanturin antama A/D-muunnostulos Luksien suhteen

Mittaustuloksista voidaan päätellä, että rakennetulla valoisuusanturilla voidaan mitata eri valoisuustilanteissa huoneen valoisuus, ja että tämä mitattu arvo voidaan tarkasti sitoa sitä vastaavaan luksimäärään. Anturin päätarkoituksena ei ole tarkasti mitata, kuinka monta luksia tilassa milläkin hetkellä on, vaan tietää ovatko tilassa valot päällä vai pois päältä. Tulokset yhdistettiin yhteen kuvaajaan liitteeseen 7, ja tästä kuvaajasta mittaustulosten avulla määriteltiin raja-arvot eri valaistustilanteisiin.

Taulukko 3 Valoisuuksien raja-arvot

Luksia	ADC	Selitys	Valot päällä/pois
0-50	> 600	Pimeää	Pois
50-100	500-600	Himmeää	Ei varmaa tietoa
100-200	410-500	Valoisaa	Päällä
> 200	< 410	Erittäin valoisaa	Päällä

4 VERKON SUUNNITTELU

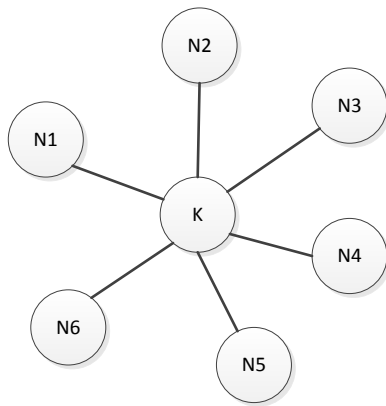
Kun antureiden data on mitattu, on tämä tieto siirrettävä eteenpäin. Tähän käytetään langatonta tiedonsiirtoa antureiden asennuksen helpottamiseksi. Laitteessa käytetään edellä mainittua Nordic Semiconductorsin lähetinvastaanotinpiiriä. Työtä varten suunniteltiin ja toteutettiin yksilöllinen kaksisuuntainen hyppyverkko. Valmista kaupallista verkkoa ei valittu sovellukseen kalliiden lisenssimaksujen takia.

4.1 Verkon topologia

Verkon suunnittelussa on ensimmäisenä valittava verkon rakenne, eli topologia. Tätä varten tutkittiin olemassa olevia topologioita, ja näiden avulla suunniteltiin tätä työtä varten soveltuva verkkotopologia.

4.1.1 Tähtitopologia

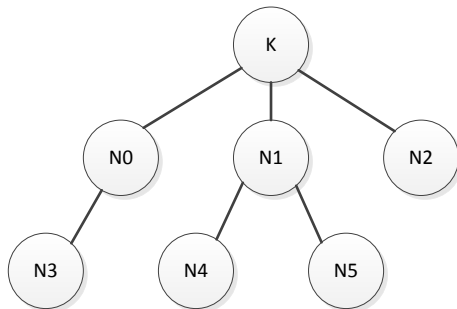
Tähtitopologiassa (kuva 13) on yksi koordinaattori (K), johon kaikki sen ympärillä olevat nodet (N1–N6) ovat liittyneet. Kaikki verkon tieto kulkee koordinaattorin kautta eivätkä nodet voi lähettää tietoa suoraan toisilleen. [13.] Koska kyseisen sovelluksen nodet eivät koskaan lähetä toisilleen tarkoitettua tietoa, ei tämän ominaisuuden puuttuminen haittaa. Tämän topologian suuri heikkous on kuitenkin sen häiriöherkkyys, koska jos koordinaattori ei ole toiminnassa, koko verkko on alhaalla. Samoin jos noden lähetysteho ei ole riittävä yltääkseen suoraan koordinaattorille, ei kyseinen node voi liittyä verkkoon. Toisaalta yhden noden puuttuminen ei haittaa muun verkon toimintaa, joten muu verkko voi toimia normaalisti.



Kuva 13 Tähtitopologia

4.1.2 Puutopologia

Puutopologia (kuva 14) on hierarkkinen topologia, joka tähtiverkon tavoin sisältää yhden koordinaattorin, mihin kaikki verkon laitteet ovat liittyneet. Tietoa ei tarvitse kuitenkaan lähettää suoraan, vaan se voidaan lähettää sekundäärisen pisteen kautta. Puutopologiaa voidaan kutsua laajennetuksi tähtitopologiaksi, jossa tähtitopologian koordinaattoreita on kytketty toisiinsa. [13.] Kuvan 14 tapauksessa tähtiverkon tapaisia koordinaattoreita olisivat nodet N0 ja N1.

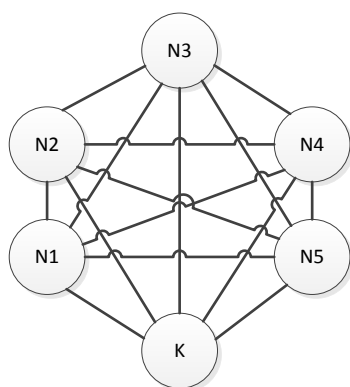


Kuva 14 Puutopologia

Puutopologian haitat ovat samantyyppiset kuin tähtitopologian, mutta puutopologia mahdollistaa paljon suuremman ja joustavamman verkon rakentamisen.

4.1.3 Mesh-topologia

Laajin ja monimutkaisin topologia kaikista on nk. mesh-topologia (kuva 15). Mesh-topologiassa kaikki verkon nodet ovat liittyneet toisiinsa nopeimman ja varminnan verkon rakentamiseksi. Eri hyppyjen määrä on $n(n-1) / 2$ eli jo kuuden laitteen verkossa hyppyjen määrä on 15 [14]. Topologian haittapuolena on hanakala toteutus sen monimutkaisuuden takia. Langattomassa verkossa on myös lähes mahdotonta saada kaikki verkon nodet samalle kuuluvuusalueelle.



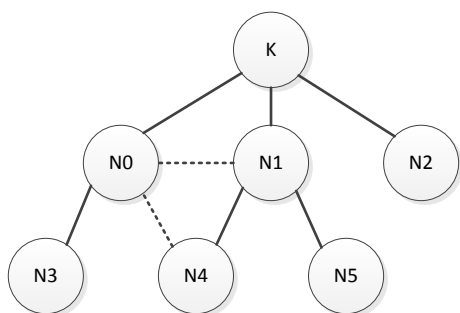
Kuva 15 Mesh-topologia

4.1.4 Hybriditopologia

Verkon vaatimuksien ja toimintavarmuuden takaamiseksi sekä laitteiden tuomien rajoitusten vuoksi oli hyödynnettävä edellä mainittujen ratkaisujen parhaat puolet. Lopulliseksi topologiaksi muodostui puutopologiamainen rakenne, jossa mahdollisuuksien mukaan jokaisella verkkoon liittyneellä laitteella on monta reittiä verkon varmuuden parantamiseksi. Vain yksi reitti on aktiivisena kerrallaan, joten tätä lopullista ratkaisua voitaisiin kutsua moniulotteiseksi puutopologiaksi, jossa jokainen eri mahdollisuus viestin perille saapumiseen olisi oma ulotteisuus. Topologiaa voitaisiin myös kutsua osittaiseksi mesh-topologiaksi (partial mesh topology), mutta koska verkon kaikki viestit edelleen kulkevat yhteisen koordinaattorin kautta, vaikkakin sekundääristen pisteiden kautta, on moniulotteinen puutopologia paremmin ratkaisua kuvaava rakenne.

Kuvassa 16 yhtenäinen viiva kuvaa verkon primääristä yhteyttä ja katkoviivoilla kuvataan hyppypisteen vaihtoehtoiset reitit verkossa. Lähetinvastaanotinpiirin ominaisuuksiin kuuluva ACK-viesti (ks. 2.3.2) mahdollistaa viestin perille saapumisen tunnistami-

sen. Näin ollen laite tietää, saako se välitettyä viestin eteenpäin verkossa. Jos viestin välittäminen epäonnistuu, käy se läpi vaihtoehtoiset reittivalintansa. Jos jokin näistä vaihtoehtoisista reiteistä on toiminnassa, siirtyy node käyttämään tätä yhteyttä. Jos mikään verkon reiteistä ei ole toiminnassa, irrottautuu laite verkosta ja etsii sen uudestaan.



Kuva 16 Hybriditopologia

Kuvan 16 tapaisen loogisen verkkorakenteen fyysinen rakenne voisi olla liitteen 8 tapainen. Liitteessä jokaisen noden sekä koordinaattorin ympärillä on erilaisilla katkovii-voilla kuvattu näiden kuuluvuusalueet. Kuvassa oletetaan, että kuuluvuus on yhtä suuri joka suuntaan ilman esteitä, eli näköyhteyttä vastaava tilanne.

4.2 Verkon looginen osoitteellisuus

Verkko suunniteltiin niin, että jokaista mahdollista reittiä vastaa yksi luku. Jokaisen verkon yhdyspisteen jakamat reitit rajoitetaan neljään. Näin ollen jokaisella yhdyspisteellä on yksi yhteys ylöspäin hierarkiassa ”vanhemmalle”, ja neljä alaspäin hierarkiassa ”lapsille”. Vain yksi vanhempi on kerrallaan aktiivinen, ja tämä vanhempi ei vaihdu, ellei tämä yhteys katkea jostain syystä. Näiden viiden yhteyden lisäksi jokainen yhdyspiste myös kuuntelee yhtä, jokaiselle verkon laitteelle yhteistä osoitetta. Tätä käytetään hyväksi verkkoon liittymisvaiheessa.

Verkko käyttää siis jokaista hyppypistettä kohden kuutta yhteyttä. Multiciever™ -ominaisuuden avulla voidaan jokaiselle yhteydelle määrittää oma fyysinen kuunte-luosoite (ks. 2.3.3). Näin ollen voidaan jokaiselle yhteydelle asettaa yksi putki. Tämä helpottaa verkon suunnittelua, ja mahdollistaa mm. verkon virhetilanteiden huomaami-sen, kun jokin muu kuin putkeen yhteyden ottanut laite lähettää tämän putken osoitteel-la.

Neljän luvun varastoiminen kuluttaa kaksi bittiä, joten loogisen verkon koon ollessa 32 bittiä voi verkossa olla maksimissaan $32 / 2 = 16$ hyppyä. Jotta verkko voisi toimia molempiin suuntiin, ei tätä maksimimäärää saa ylittää, koska tällä luvulla löydetään aina tietty yhdyspiste verkossa. Jos verkkoa ei tarvitse käyttää kuin yhteen suuntaan, ei hypyen maksimimäärälle ole mitään rajoitusta. Toisaalta, jos kaksisuuntaisen verkon tarvitsisi olla suurempi, voidaan verkon loogista kokoa suurentaa esimerkiksi 64 bittiin, joka mahdollistaisi 32 hypyn verkon.

4.3 Verkkoon liittyminen

Laitetta käynnistäessä tai yhteyden katkettua alkaa laite etsiä verkkoa. Laitteita, joilla on suora tai välillinen yhteys päätepisteeseen, kutsutaan verkkoon liittyneeksi ja laite, jonka viestit eivät saavuta päätepiistettä, ovat verkosta ulkona.

Verkkoon liittyminen tapahtuu seuraavasti:

1. Laite lähettää yleisen kyselyn (PING) yleisellä osoitteella, joka on kaikille verkkoon liittyneille laitteille sama.
2. Kaikki laitteet, jotka kuulevat tämän viestin, vastaavat kyselyn lähettäneelle laitteelle oman sijaintinsa verkossa.
3. Jos laitteita on useita samalla tasolla, valitaan vähiten lapsia omaava laite verkon kuorman tasaamiseksi.
4. Kaikki kyselyn vastaukset tulkitaan, ja näistä valitaan parhaat hypyt. Näille laitteille lähetetään nyt yksitellen yhteydenottopyyntö, jolla varataan yksi laitteen lähetinvastaaottimen putkista (ks. 2.3.3).
5. Varauksiin onnistuneesti vastanneiden laitteiden osoitteet tallennetaan laitteen muistiin.
6. Laite on nyt liittynyt verkkoon ja on valmis lähettämään, vastaanottamaan ja välittämään verkon viestejä sekä vastamaan muiden laitteiden lähettämiin yleisiin kyselyihin verkkoon liittymisestä.

Tämä prosessi mahdollistaa verkon täydellisen rakentumisen automaattisesti heti, kun verkon keräin on valmis vastaamaan yleisiin kyselyihin. Tämä myös tekee verkosta täysin itsestään korjaantuvan (self-healing network), koska virheen tapahtuessa etsitään verkko uudestaan.

Liitteessä 9 on esitelty anturipaketin funktion NetworkSendParent() toiminta, jonka tehtävänä on lähettää viesti verkon topologiassa kohti keräintä. Tätä funktiota kutsutaan, kun lähetettävä viesti on kirjoitettu mikrokontrollerin txBuff-puskuriin.

5 VERKON TESTAUS

Suunnitellun ja toteutetun verkon toimintaa testattiin sen toimintaetäisyyden ja toimintanopeuden suhteen.

5.1 Toimintaetäisyys

Verkon toimintaetäisyyden määrittämiseksi suoritettiin mittaus kahdella testilaitteella. Toinen laite asetettiin pysyvästi kuuntelutilaan ja toinen lähettämään jatkuvasti 24 tavun viestejä kuuntelutilassa olevalle laitteelle. Molempiin asetettiin samat asetukset, kuin mitä lopulliseen tilaverkkoon asetettaisiin. Jokaisen viestin jälkeen viestin lähettänyt testilaitte tulostaa rekisterin 0x08 (OBSERVE_TX) bitit 3:0 jotka kertovat, kuinka monta kertaa lähetinvastaanotin on joutunut uusimaan lähetyksen, että viesti on saapunut määränpäähensä. Tämän tiedon avulla saadaan selville, kuinka luotettavaa on tiedonsiirto kahden eri verkon jäsenen välillä eri etäisyyksillä. Mittaukset suoritettiin lähetysnopeuksilla 250 kbps sekä 2 Mbps.

Mittaustulokset ovat näkyvissä liitteissä 10 ja 11. Mustalla ympyrällä kuvataan kuuntelulaitteen sijaintia, johon viestejä lähetettiin. Tummallalla alueella on kuvattu parasta mahdollista kuuluvuutta eli aluetta, jossa jokainen viesti saapuu perille ensimmäisellä yrittämällä. Vaaleammalla alueella jokainen viesti saapuu perille alle maksimimäärän uudelleenlähetyksiä (15). Käytännössä uudelleenlähetykset tällä alueella vaihtelivat 0–10 uudelleenlähetyksen välillä. Valkoisella alueella ei ollut joko mitään kuuluvuutta tai viestien saapuminen perille ei ollut taattua. Vaikka jossain kohdissa viesti saattaisi päästä määränpäähensä suurimman osan aikaa, ei tämä alue sovi asennuspaikaksi sen huonon kuuluvuuden takia. A3-kerroksessa sijaitseva häiriösuojattu tila (H) estää sähkömagneettisen säteilyn pääsyn sen sisälle ja näin ollen myös häiritsee sen takana olevaa aluetta.

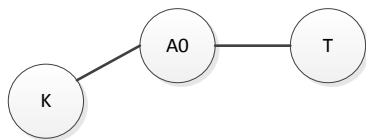
Erot ovat selkeät nopeimman ja hitaimman lähetysnopeuden välillä. Hitaimman lähetysnopeuden paras alue on jopa suurempi kuin nopeamman lähetysnopeuden koko kuuluvuusalue. Tämä oli ennalta arvattavissa lähetinvastaanotinpiirin datalehden perusteel-

la. Laitteen herkkyys putoaa -94 dBm:stä -82 dBm:iin siirtonopeuden noustessa 250 kbps:ssa 2 Mbps:iin [5, s. 16].

5.2 Toimintanopeus

Verkon maksimikoko määräytyy viestin välitysajan perusteella. Tätä varten suoritettiin tarkat mittaukset, kuinka kauan anturikeräimellä kestää välittää viesti verkossa. Saleae Logic -logiikka-analysaattori kytkettiin laitteen prototyyppiin, ja tällä tutkittiin eri signaaleja. Mittaukset suoritettiin siirtonopeuksilla 250 kbps ja 2 Mbps. Tulokset ovat nähtävissä liitteissä 12 ja 13. Liitteissä näkyvät mikrokontrollerin ja lähetinvastaanotinpiirin välinen tiedonsiirto. Liitteessä 12 on myös selitetty, mitä lähetinvastaanotinpiiri ja mikrokontrolleri tekevät milläkin ajanhetkellä.

Verkon maksimikokoa testattiin kuvan 17 mukaisella järjestelyllä. Yksi anturipaketti (A0) otti yhteyden keräimen (K) muodostamaan verkkoon. Anturipaketin yhteen putkeen lähetettiin testilaitteesta (T) eri väliajoilla verkon viestejä, jotka anturikeräimen tuli välittää verkon keräimelle. Testilaitteella simuloitiin siis suurempaa verkkoa, jonka kaikki tieto lähetetään solmupisteelle A0. Lähetysväliaikaa pienennettiin, kunnes verkko ei enää pystynyt siirtämään kaikkea tietoa keräimelle.



Kuva 17 Testausjärjestelyt

Lyhyimmäksi väliajaksi muodostui noin viisi kertaa yhteen viestin välitykseen kuluva aika. Lähetysnopeudella 250 kbps tämä väliaika oli n. 11 ms ja nopeudella 2 Mbps hie-
man yli 5 ms. Näistä arvosta saadaan laskettua verkon teoreettinen suurin mahdollinen koko. Taulukkoon 4 on kerätty verkkoon liittyneiden anturipakettien maksimilukumäärä eri tilanteissa.

Taulukko 4 Verkkoon liittyneiden anturipakettien maksimimäärä eri tilanteissa

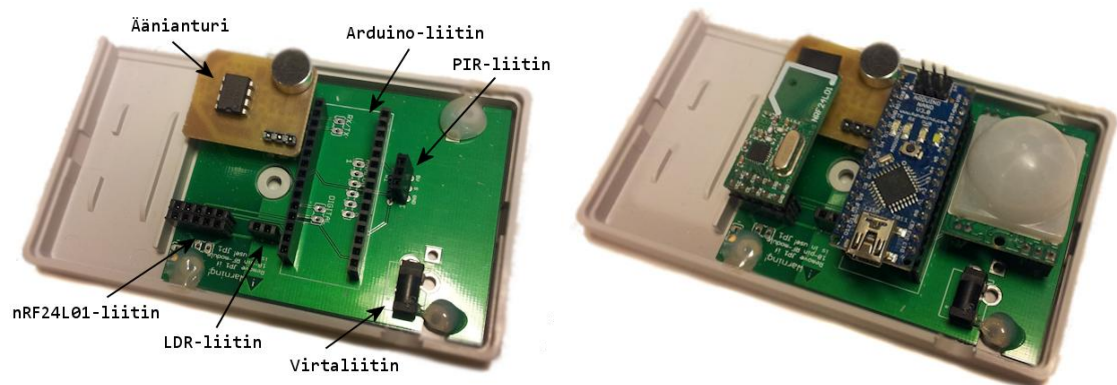
Lähetysväliaika	250 kbps	2 Mbps
1 s	89	194
30 s	2667	5825
1 min	5333	11650
5 min	26667	58252
15 min	80000	174757

Kuten taulukosta 4 nähdään, on yhden keräimen varaan rakentuva verkko lähetyksenopeudesta riippumatta riittävän nopea suoriutumaan sen tehtävästä. Jos verkkoon mahtuu muutama tuhat tai muutama kymmentä tuhatta laitetta, ei tällä ole käytännön merkitystä, koska koko oppilaitoksen kattava verkko ei olisi näin laaja. Näin suurissa verkoissa rajoittavaksi tekijäksi muodostuisikin keräimen kyky siirtää kaikki vastaanotettu tieto palvelimelle, eikä verkon hyppypisteiden nopeus siirtää tietoa eteenpäin.

Suoritettujen testien perusteella pääteltiin, että lähetyksenopeus on parasta asettaa alhaisimmalle asetukselle. Mikrokontrollerin kelloaajuutta voidaan pienentää virrankulutuksen vähentämiseksi vähintään puoleen, jolloin mikrokontrolleria voitaisiin käyttää piirin omalla sisäisellä oskillaattorilla.

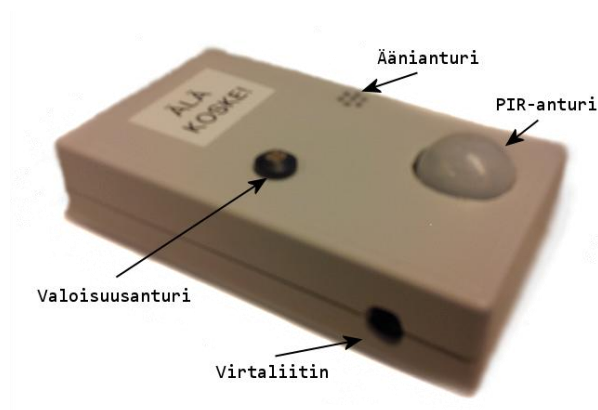
6 PROTOTYYPPI JA DEMO-YMPÄRISTÖ

Käytännön testien jälkeen rakennettiin anturipaketin ja keräimen prototyyppi. Keräimellä toimii fyysisesti täysin sama laite kuin anturipaketissa, mutta se sisältää eri ohjelman. Prototyypin piirikaavio on esitelty liitteessä 14. Piirikaavion mukainen piirilevy suunniteltiin ja tilattiin työtä varten. Tämä piirilevy ja siihen kiinnitettävät moduulit sijoitettiin TME:ltä tilattuun ABS-26 -yleiskoteloon (kuva 17).



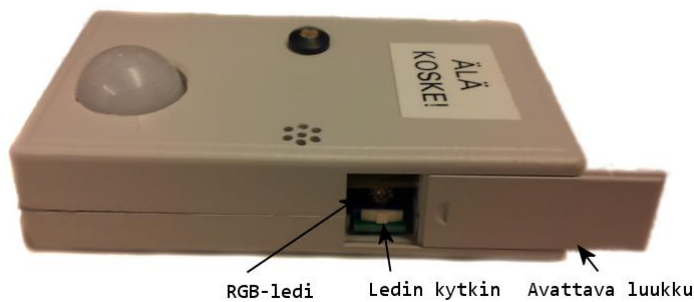
Kuva 17 Prototyypin piirilevy ilman moduuleita ja moduulit kiinnitettynä

Koteloa muokattiin niin, että prototyypin liike- ja valoisuusanturit saatiin sijoitettua kotelon ulkopuolelle. Virtaliittimelle ja äänianturille porattiin niille sopiviin kohtiin reiät (kuva 18).



Kuva 18 Koteloitu anturipaketin prototyyppi

Kotelon avattavan luukun taakse sijoitettiin anturipaketin tilan ilmaiseva RGB-ledi sekä sen päälle ja pois kytkemiseen tarkoitettu kytkin (kuva 19).

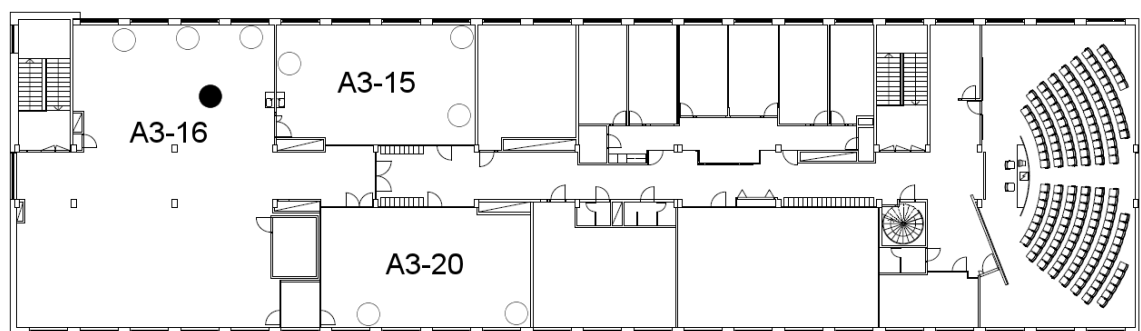


Kuva 19 Koteloitu anturipaketin prototyyppi

Prototyyppinä valmistettiin yhdeksän kappaletta, ja näitä käytettiin langattoman hyppiverkon testaamiseen. Prototyyppien avulla tutkittiin, kuinka hyvin rakennettu järjestelmä suoriutuu tilankäytön havaitsemisesta.

6.1 Testausjärjestelyt

Kahdeksan anturipakettia ja yksi keräin sijoitettiin TAMK:n A-siiven kolmanteen kerrokseen. Anturipaketteja sijoitettiin laboratoriotiloihin A3-15, A3-16 ja A3-20 (kuva 20). Nämä anturipaketit asetettiin lähettämään anturitietonsa minuutin väliajoin keräimelle, joka siirsi kaiken vastaanotetun tiedon sarjaliikenteenä tietokoneelle. Tietokoneeseen asennettiin sarjaliikennettä tutkiva ohjelma, joka tallensi kaiken keräimen lähettämän tiedon taulukkomuotoon (.csv) jatkotarkastelua varten.



Kuva 20 Demo-ympäristön järjestelyt

Luokkien käyttäjät merkkasivat 15 minuutin tarkkuudella, koska tulivat tilaan sisään ja koska poistuivat tilasta. Näin saatiin tieto siitä, kuinka monta ihmistä tilassa oli milläkin hetkellä. Näitä listoja täytettiin vaihtelevalla menestyksellä, mutta eritoten tilan A3-20 listat oli täytetty luotettavalla tavalla. Koska muiden tilojen merkkauksissa on suuriakin puutteita tietyillä ajanjaksoilla, käytetään analysoinnissa vain tilan A3-20 mittausdataa.

6.2 Mittaustulokset

Testin aikana keräin vastaanotti yli 60000 mittaustietoa anturipaketeilta. Tätä tietoa analysoitiin ja verrattiin luokkiin sijoitetuista henkilölistoista saatuun tietoon siitä, kuinka monta henkilöä tilassa oikeasti oli milläkin hetkellä.

6.2.1 Liike

Minuutin tarkasteluväli on liikkeen havaitsemiseen liian pieni aikaväli. Näin ollen liikkeiden lukumäärä summattiin yhteen viiden minuutin aikaväliltä. Liitteen 15 kuvassa 1 on tilan liikeantureiden yhteensä havaitsema liike. Summatusta liikkeen määrästä on vaikea tehdä johtopäätöksiä, joten tämä tieto sievennettiin niin, että jos liikettä on havaittu, arvo on yksi ja jos liikettä ei ole havaittu, arvo on nolla. Molempien anturipaketien tiedot yhdistettiin yhteen kuvaajaan liitteen 15 kuvaan 2. Tutkitulla aikavälillä on kaksi hetkeä, jolloin listoihin merkattu tieto kertoisi, että tilassa on henkilöitä, mutta liikettä ei ole havaittu. Tarkasteluaikaa nostettiin 15 minuuttiin 5 minuutin sijasta ja tällä tavalla saatiin täysin tilankäyttöä vastaava kuvaaja (liite 15 kuva 3).

6.2.2 Valoisuus

Valoisuusantureiden tietoa ei käsitelty millään tavalla, vaan vartin välein havaittu tieto sijoitettiin taulukkoon. Näiden antureiden avulla havaitaan valojen päälle laitto, kun ensimmäiset henkilöt astuvat tilaan aamulla, ja kun viimeiset poistuvat tilasta illalla havaitaan valojen sammutus (liite 16 kuva 1 ja kuva 2). Tämä tieto voidaan yhdistää muiden antureiden mittaustuloksiin varmemman päätelmän saavuttamiseksi.

6.2.3 Ääni

Ääniantureiden mittaama tieto oli kaikista vaikein tulkita. Koska anturi ei voi erottaa puhetta muusta taustamelusta, on tätä tietoa käsiteltävä jollain tapaa. Esimerkiksi tilassa A3-20 on ilmastoinnin tuottama melu selvästi normaalia luokkahuonetta suurempi. 15 minuutin aikavälillä vastaanotettujen A/D-muunnostulosten suurin arvo vähennettiin

pienimmällä arvolla (liite 17 kuva 1). Tällä tavalla saadaan tilan taustamelu poistettua mittaustuloksista.

Alun perin ääniantureilla oli tarkoitus arvioida, kuinka monta ihmistä tilassa milläkin hetkellä on. Testin aikana kerätyn datan perusteella huomattiin, että tämä ei kuitenkaan onnistu. Ääniantureiden tuottama data ei sovellu tilan henkilömäärän arviointiin, mutta tätä tietoa voidaan käyttää hyväksi tilankäytön havaitsemisessa. Näin ollen määritettiin raja-arvo, milloin tilassa arvioidaan olevan ihmisiä. Arvoksi muodostui A/D-muunnoksien erotustulos 50. Arvo on yksi, jos tilassa havaittu melu vaihtelee riittävästi, ja arvo on nolla, jos melu pysyy tasaisena (liite 17 kuva 2). Äänianturilla havaittu tilankäyttö ei ole täydellistä, mutta siksi järjestelmän suunniteltiin käyttävän useita eri antureita.

Äänianturi ei lineaarisesti muodosta A/D-muunnostuloksia taustamelun suhteen (ks. 3.2.2). Tämän takia eri tiloille tulisi taustamelun perusteella määrittää eri raja-arvot. Tämä vaatisi kuitenkin lisätutkimusta, jota tämän opinnäytetyön puitteissa ei ole suoritettu.

Kaikkien kolmen anturin tiedot yhdistämällä saadaan rakennettua tarkka kuva siitä, mikä tilan aito käyttöaste on. Testien aikana huomattiin, että pelkästään hyvin sijoitetulla liikeanturilla voidaan havaita tilan käyttö luotettavasti. Työhön valittu liikeanturi ei ole markkinoiden parhaita. Joillain antureilla päästään jopa 12 metrin havaitsemisetaisyys-teen työssä käytetyn 5-7 metrin etäisyyden sijasta [15]. Opinnäytetyön aikana kerätyn tiedon avulla voidaan rakentaa järjestelmä, joka pystyy todella luotettavasti ja nopeasti ilmoittamaan, onko tila käytössä vai ei.

7 YHTEENVETO

Viimeisimmässä, anturipaketin ja keräimen yhdistävässä, ohjelmistoversiossa oli yhteensä noin 6000 ohjelmariviä kommentteineen. Suunniteltu järjestelmä suoriutuu sen pääasiallisesta tehtävästä, tilankäytön havaitsemisesta ja tämän tiedon lähettämisestä langattoman hyppyyverkon avulla hyvin. Demo-ympäristössä huomattiin kuitenkin, että muu langaton tiedonsiirto, oletettavasti WLAN, häiritsi suunnitellun verkon toimintaa. Verkko onnistui kuitenkin aina korjaamaan itsensä toimivaan tilaan, kuten sen tulikin toimia. Verkosta tulisi kuitenkin tehdä paremman häiriöitä sietävä, jotta tätä verkon korjausta ei tarvitsisi usein suorittaa.

Järjestelmä ei tällä anturikokoonpanolla pysty tarkasti mittaamaan luokassa olevien ihmisten lukumäärää, vaan tätä varten olisi järjestelmään liitettävä joitain muita antureita. Perinteisillä antureilla on kuitenkin vaikeaa havaita tilankäyttäjien lukumäärää, koska tilat ovat niin erilaisia. Ilmastointi, ovatko ovet tai ikkunat kiinni vai auki, huoneen tilavuus, minkälainen tapahtuma tilassa on, kuinka äänekkäitä tilan käyttäjät ovat ja kaikki muu vaikuttavat siihen, mitä arvoja anturit tuottavat. Näiden arvojen sitominen tilankäyttäjien tarkkaan lukumäärään on melko mahdotonta.

Toteutettu hyppyyverkko on työssä parhaiten onnistunut osa, ja sitä voidaan käyttää useassa eri sovelluksessa. Toteutettua verkkoa onkin jo suunniteltu käytettäväksi muissa sovelluksissa, kuten hälytinja järjestelmässä ja kulunvalvonnassa. Kehitetty järjestelmä ei ole valmis esimerkiksi kaupalliseen käyttöön, vaan jatkokehitys tietyillä osa-alueilla on tarpeellista. Tilaverkon kehitystä on jatkettu, ja siihen on suunnitteilla oma piirilevy, jossa ei käytetä valmismoduuleita. Taajuushyppely, virtaa säästävä aikajakokanavointi ja 128-bittinen AES-salaus on myös jo osin toteutettu. Myös moduulityyppinen antureiden liittäminen I2C-väylän kautta on suunnitteilla.

LÄHTEET

1. Atmel. ATmega prosessorit. Datalehti. 05/2011. Luettu 22.2.2012.
<http://www.atmel.com/Images/doc8271.pdf>
2. Generation Robots. Arduino Nano board 3.0. Luettu 9.9.2012.
http://www.generationrobots.com/boutique_us/images_produits/arduino-nano-z.jpg
3. Barnett, R., O’Cull, L., Cox S. & Cox S. Embedded C Programming and the Atmel AVR. 2. Kuvitettu painos, Cengage Learning, 2007. ISBN 1418039594, 9781418039592.
4. Serial peripheral interface in AVR microcontrollers. Luettu 10.12.2012.
<http://www.embedds.com/serial-peripheral-interface-in-avr-microcontrollers/>
5. Nordic Semiconductors. nRF24L01P. Datalehti. 09/2008. Luettu 31.8.2012.
https://www.nordicsemi.com/eng/content/download/2726/34069/file/nRF24L01P_Product_Specification_1_0.pdf
6. Webster, J. The measurement, instrumentation, and sensors handbook. Springer 1999. ISBN 978-3-540-64830-7.
7. How PIR Motion Sensors Work. Luettu 24.2.2012.
<http://www.gadgetshack.com/motionsensor.html>
8. Futurlec. Technical Information – PIR Sensor Module. Luettu 13.10.2012.
http://www.futurlec.com/PIR_Module.shtml
9. Crocker, J. Handbook of Acoustics. John Wiley & Sons 1998. ISBN 047125293X, 9780471252931.
10. SPECIFICATION of LD-MC-0905P. Luettu 19.11.2012.
<http://www.tme.eu/fi/Document/9b07afd90a47f1182065e94e017d67da/ld-mc-0905p.pdf>
11. Sound Power and Sound Pressure. Luettu 18.11.2012.
<http://www.usmotors.com/TechDocs/ProFacts/Sound-Power-Pressure.aspx>
12. Godse, A.P., Bakshi, U.A. 2008. Electronic Devices. 3. Uudistettu painos. Technical Publications, 2008. ISBN 8184314205, 9788184314205.
13. IEEE 802.15.4. Antti Havanko. Luettu 21.11.2012.
<https://trac.assembla.com/Langaton/wiki/IEEE802.15.4>
14. Telecommunications & Network Security. Luettu 13.10.2012.
<http://lipas.uwasa.fi/~h79423/ttp/ttp03.html>

15. Passive Infrared Sensors: PaPIRs (EKMB) Specifications. Luettu 13.12.2012.
<http://pewa.panasonic.com/components/built-in-sensors/passive-infrared-sensors/papirs-ekmb/specifications/>

Liite 2. Viestipaketin lähettämiskoodi

```

uint8_t SendPayload(uint8_t ack)
{
    /* Kirjoitetaan puskurin txBuff sisältö, jonka koko on PAYLOAD_WIDTH,
       lähetinvastaanottimen puskuriin ilman ACK-lippua */
    if (ack == NO_ACK) SPI_Write(W_TX_PAYLOAD_NO_ACK, txBuff, PAYLOAD_WIDTH);

    /* Kirjoitetaan puskurin txBuff sisältö, jonka koko on PAYLOAD_WIDTH,
       lähetinvastaanottimen lähetyspuskuriin ACK-lipulla */
    else SPI_Write(W_TX_PAYLOAD, txBuff, PAYLOAD_WIDTH);

    // Siirrytään lähetystilaan
    TX_Mode();

    // Käytetään linjaa CE 10us ylhäällä, jolloin puskuriin kirjoitettu tieto
       lähetetään */
    CEH;
    _delay_us(10);
    CEL;

    timeout = 0;

    // Odotetaan keskeytyksen antamaa TX_DS tai MAX_RT keskeytystä
    while (interruptTX == CLEAR)
    {
        _delay_us(1);
        timeout++;

        /* Varmistetaan, että koodi ei jää ikuisen while-silmukkaan käyttämällä
           koodipohjaista aikakatkaisua */
        if (timeout > TIMEOUT_SW)
        {
            // Jos DEBUG-tila on päällä, lähetetään tieto aikakatkaisusta
#ifdef DEBUG
            printf(" Timeout in SendPayload()\n");
#endif // DEBUG

            return TIMEOUT;
        }
    }

    // Jos lähetys onnistui palautetaan SUCCESS ja kuitataan keskeytyslippu
    if (interruptTX == TX_DS)
    {
        interruptTX = CLEAR;
        return SUCCESS;
    }

    // Jos lähetys epäonnistui palautetaan ERROR ja kuitataan keskeytyslippu
    else if (interruptTX == MAX_RT)
    {
        interruptTX = CLEAR;
        return ERROR;
    }

    // Tuntematon virhetila, johon ei koskaan pitäisi tulla
#ifdef DEBUG
    printf(" Unknown error @ SendPayload()\n");
#endif
    interruptTX = CLEAR;
    return ERROR;
}

```

Liite 3. SPI-väylän funktioita

```
// Esimerkeissä käytetyt makrot

#define SETBIT(x, y)    (x |= y)
#define CLEARBIT(x, y) (x &= ~y)

#define CSNH            SETBIT(PORT_SPI, CSN)
#define CSNL            CLEARBIT(PORT_SPI, CSN)

#define CEH             SETBIT(PORT_SPI, CE)
#define CEL             CLEARBIT(PORT_SPI, CE)

uint8_t SPI_RW(uint8_t data)
{
    // Lähetetään data
    SPDR = data;

    // Odotetaan, että data on lähetetty
    while (!(SPSR & (1 << SPIF)));

    // Palautetaan vastaanotettu data
    return SPDR;
}

uint8_t SPI_Write(uint8_t cmd, uint8_t *pBuf, uint8_t len)
{
    uint8_t status;

    // Asetetaan linja CSN alas, jotta piiri osaa vastaanottaa lähetetyn tiedon
    CSNL;

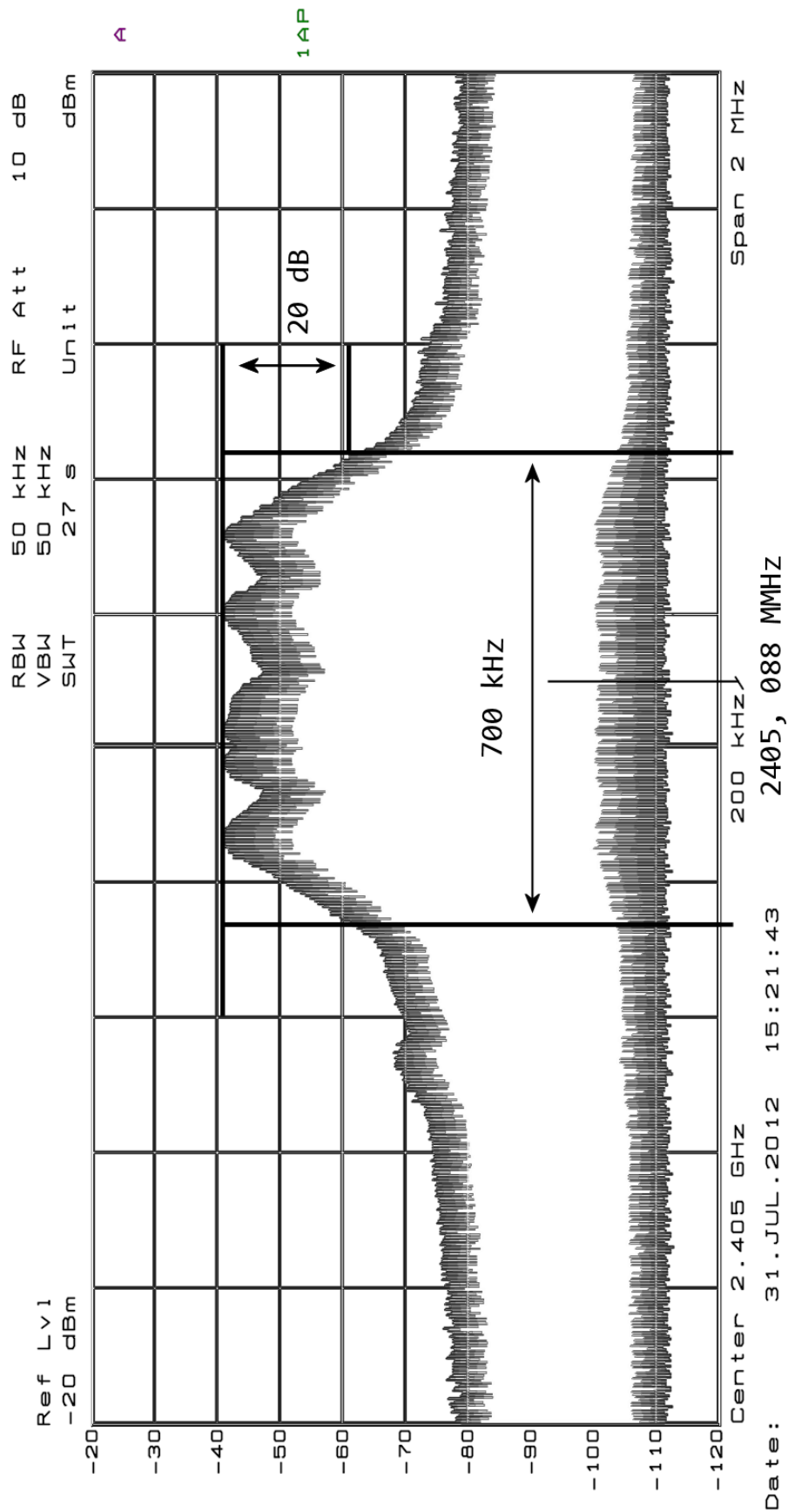
    // Lähetetään komento cmd
    status = SPI_RW(cmd);

    // Lähetetään taulukon pBuf sisältö
    while(len--) SPI_RW(*pBuf++);

    // Asetetaan linja CSN ylös, jotta piiri tietää lähetetyn tiedon päättyneen
    CSNH;

    // Palautetaan rekisterin 0x07 (STATUS) sisältö
    return status;
}
```

Liite 4. nRF24L01:n spektri



Liite 5. Äänimittauspöytäkirja

Mittauspöytäkirja - Äänenpainemittaus
19.11.2012

Miikka Mikkonen

Mittaustyökalut: TES1352H - äänenpainemittari
Atmel ATmega328p A/D-muunnin

Mittaus Valkoinen kohina A-painotus

dB(A)	ADC	Kommentti
<46	517-524	Kohinataso
46	525	Matalin havaittu äänenpaine
50	530	
55	535	
60	540	
65	570	
70	620	
72	650	
75	700	
78	790	
80	825	
81	860	

Liite 6. Valoisuusmittauspöytäkirja

Mittauspöytäkirja - Valoisuusmittaus
22.10.2012

Miikka Mikkonen

Mittaustyökalut: Hagner EC1 - luksimittari
Atmel ATmega328p A/D-muunnin

Mittaus 1 Anturi vaakasuorassa, pimennysverhot alhaalla

Lux	ADC	Kommentti
0	1021	Valot pois päältä
13	842	Valot himmeimmällä asetuksella
15	817	
22	770	
31	717	
34	696	
63	590	
102	504	
163	418	
239	356	
290	326	
340	310	
450	265	Yksi valo suoraan yläpuolella täysin päällä
1150	170	Kaikki huoneen valot päällä
1300	154	
1350	151	
1400	149	

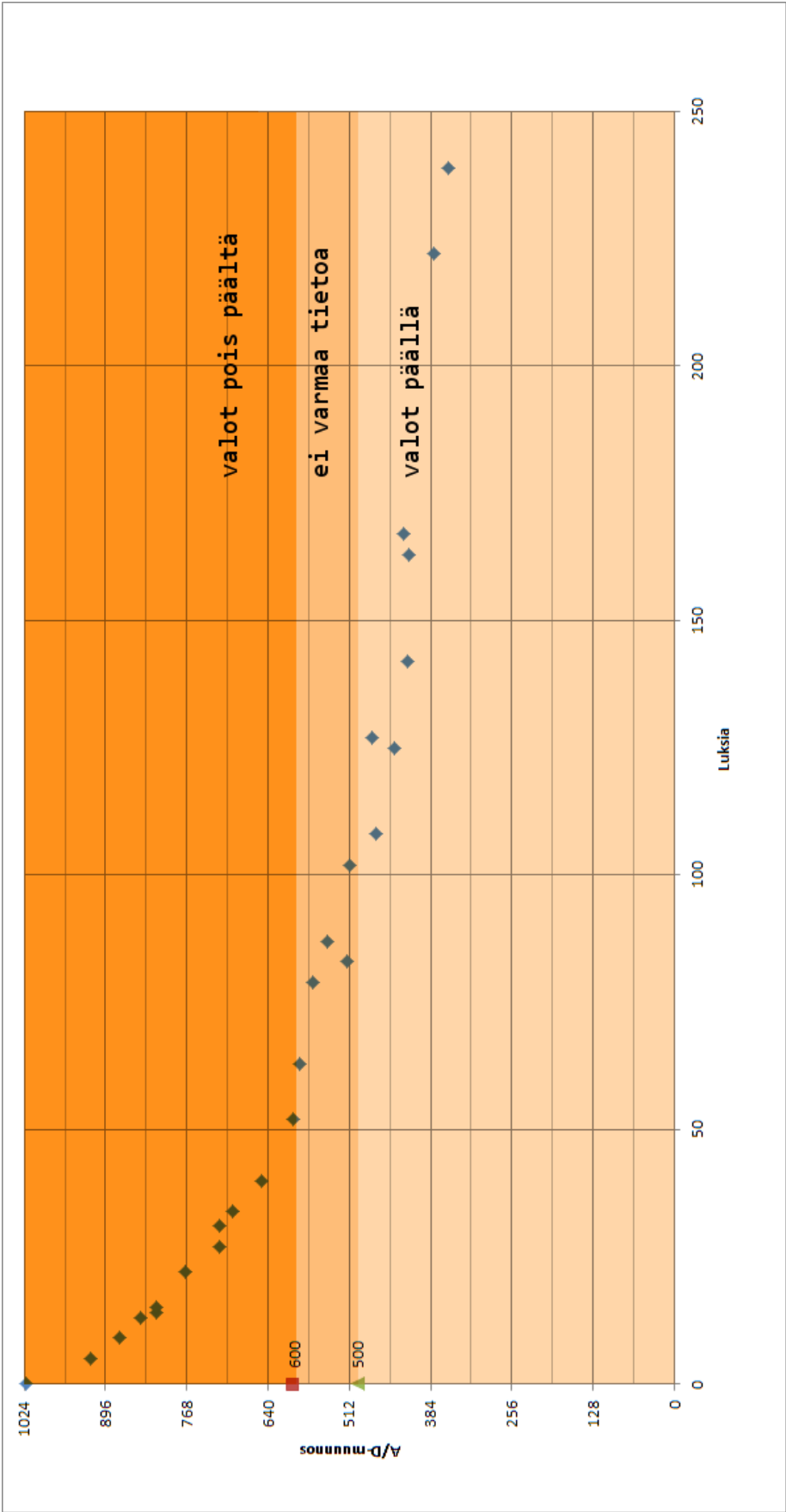
Mittaus 2 Anturi pystysuorassa, pimennysverhot alhaalla

Lux	ADC	Selitys
0	1021	Valot pois päältä
5	919	Valot himmeimmällä asetuksella
9	875	
14	817	
27	717	
40	650	
52	600	
83	515	
108	471	
125	442	
142	421	Yksi valo täysin päällä (valo ei osu suoraan anturiin)
520	222	Kaikki huoneen valot päällä
550	220	
600	208	Kaikki valot päällä noin minuutin ajan

Mittaus 3 Anturi pystysuorassa, ulkoa tuleva valo osuu anturiin

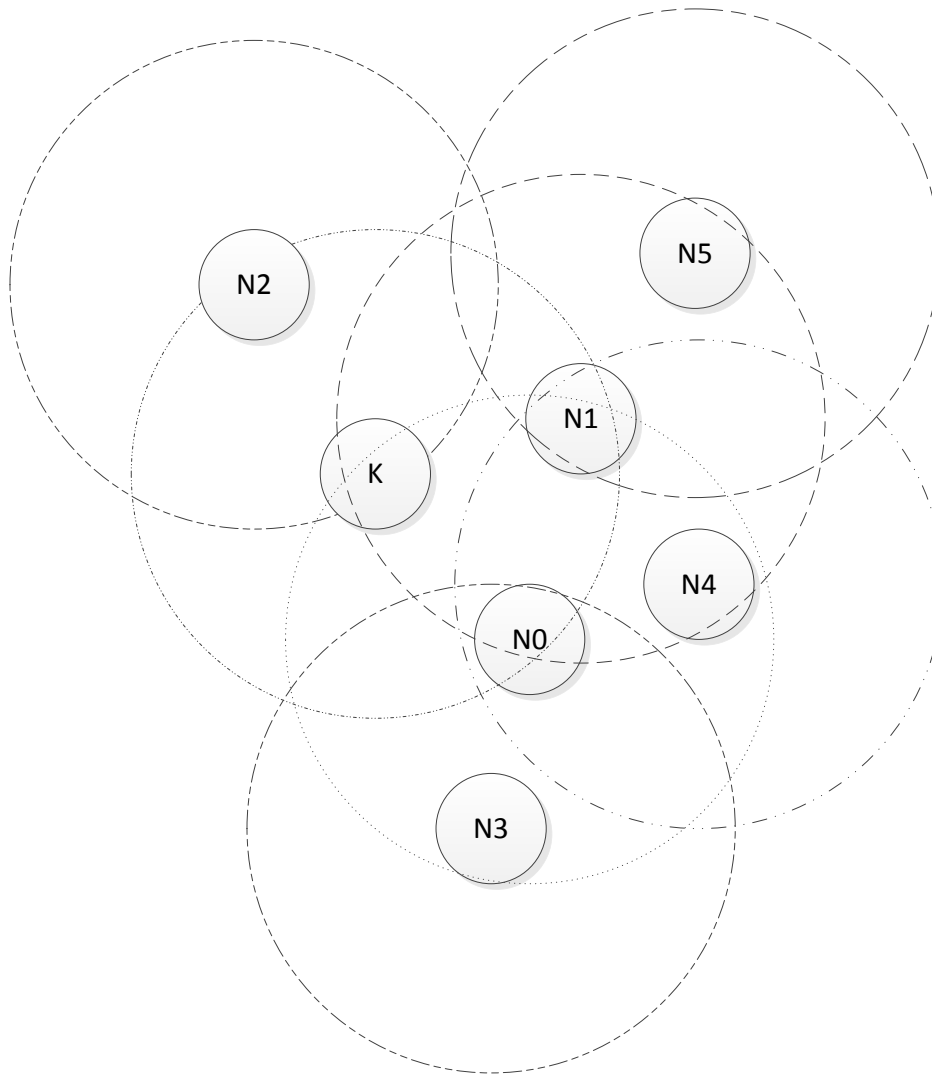
Lux	ADC	
79	569	Valot pois päältä
87	547	Valot himmeimmällä asetuksella
102	518	
127	477	
167	427	
222	378	
280	341	
325	318	Yksi valo täysin päällä (ei osu suoraan anturiin)
600	229	Kaikki huoneen valot päällä
670	217	
700	212	
750	205	Kaikki valot päällä noin minuutin ajan

Liite 7. Valoisuuskuvaaja

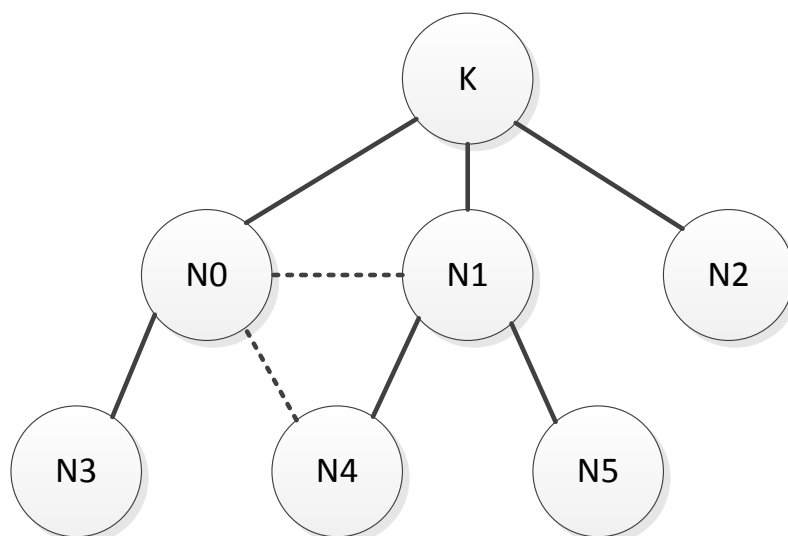


Liite 8. Hybriditopologian esimerkki

Fyysinen rakenne:



Ja sitä vastaava looginen rakenne:



Liite 9. Verkon lähetyshanktio

```

uint8_t NetworkSendParent(void)
{
    /* Käydään läpi kaikki vanhemmat
       node.pNodes sisältää vanhempien lukumäärän */

    for (uint8_t i = 0; i < node.pNodes; i++)
    {
        /* Taulukossa PN_STATUS[] on tieto siitä, onko PN_ADDR[] osoite
           aktiivinen vai ei */
        if (PN_STATUS[i] != NONACTIVE)
        {
            /* Osoite taulukossa PN_ADDR[] on 16-bittinen, mutta oikea fyysinen osoite
               on 5 tavua, joten tästä tiedosta sekä siitä, mihin putkeen (PN_PIPE[])
               on liitetty, lasketaan lähetettävän laitteen fyysinen osoite */
            DecryptAddress(TX_ADDRESS, PN_ADDR[i], PN_PIPE[i]);

            /* Kirjoitetaan laskettu osoite TX_ADDRESS lähetinvastaanotinpiiriin
               lähetysosoitteeksi */
            TX_Address(TX_ADDRESS);

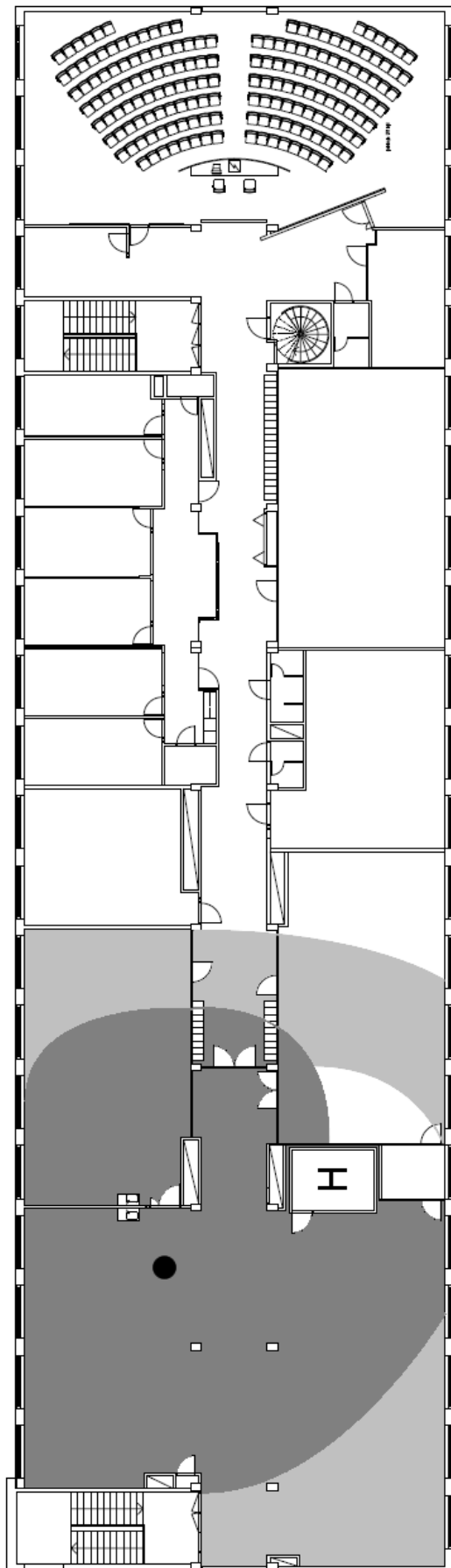
            /* Kirjoitetaan putken 0 kuunteluosoitteeksi sama lähetysosoite, jotta
               voidaan vastaanottaa ACK-viesti */
            RX_Address(TX_ADDRESS, 0);

            // Lähetetään viesti
            if (SendPayload() == SUCCESS)
            {
                // Lähetys onnistui, palataan kuuntelutilaan ja palautetaan SUCCESS
                RX_Mode();
                return SUCCESS;
            }
        }
    }
    /* Lähetys epäonnistui kaikille vanhemmille, lähetetään tieto tästä kaikille
       kaikille lapsille */
    NetworkSendDetachAll();

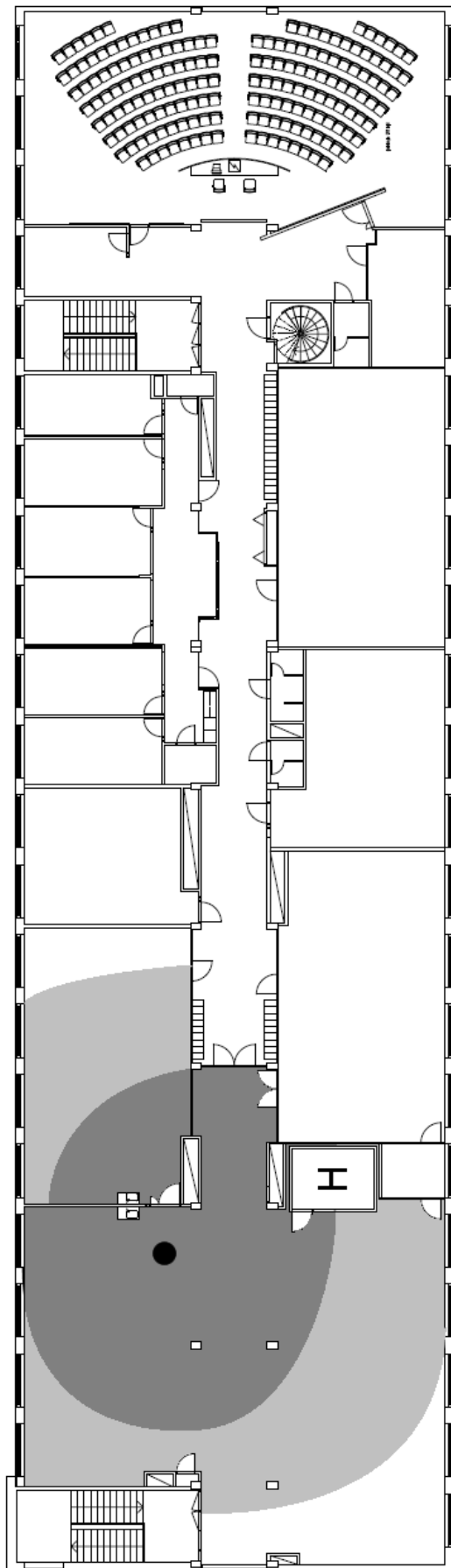
    // Palataan kuuntelutilaan ja palautetaan ERROR
    RX_Mode();
    return ERROR;
}

```

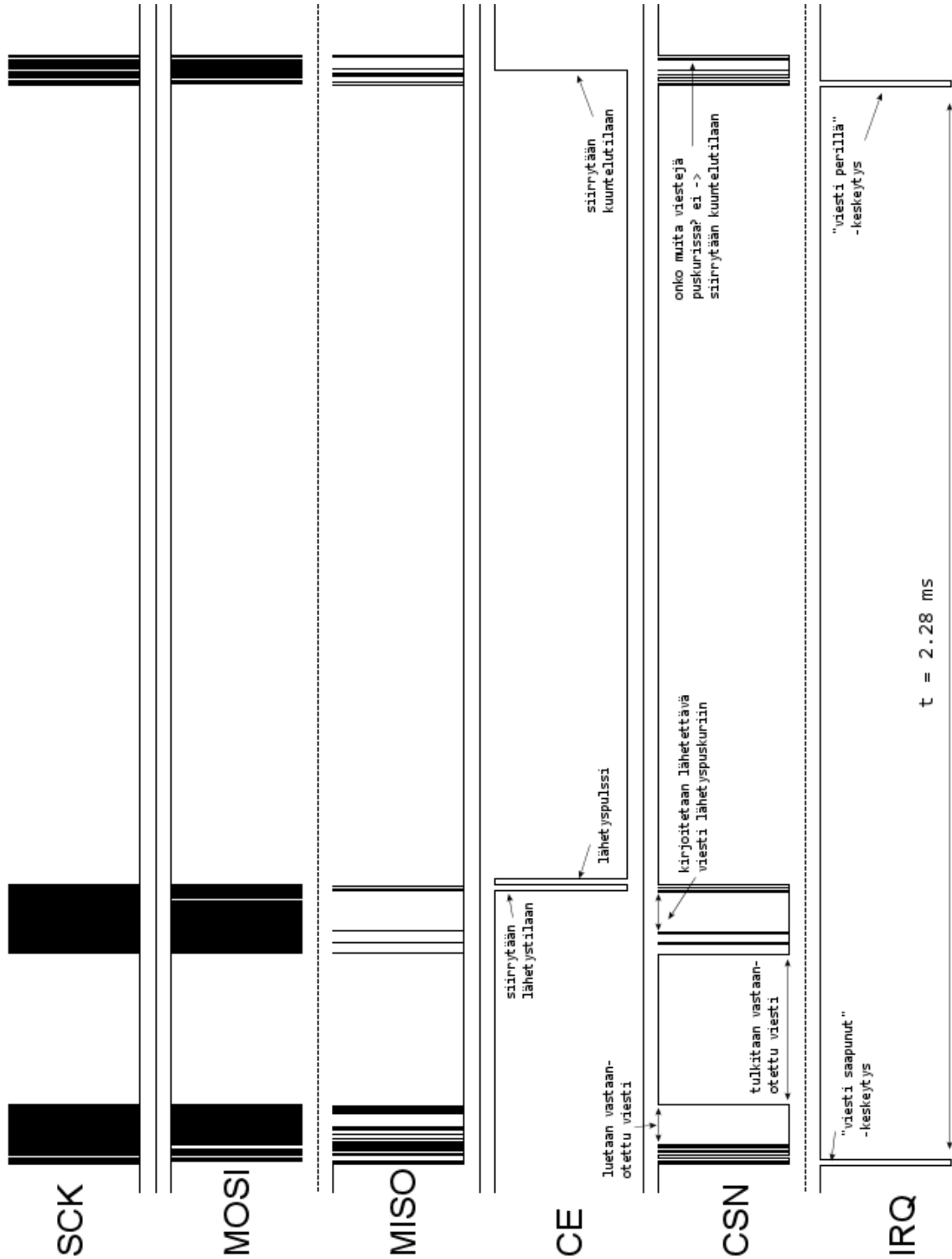
Liite 10. Verkon toimintaetäisyys siirtonopeudella 250 kbps



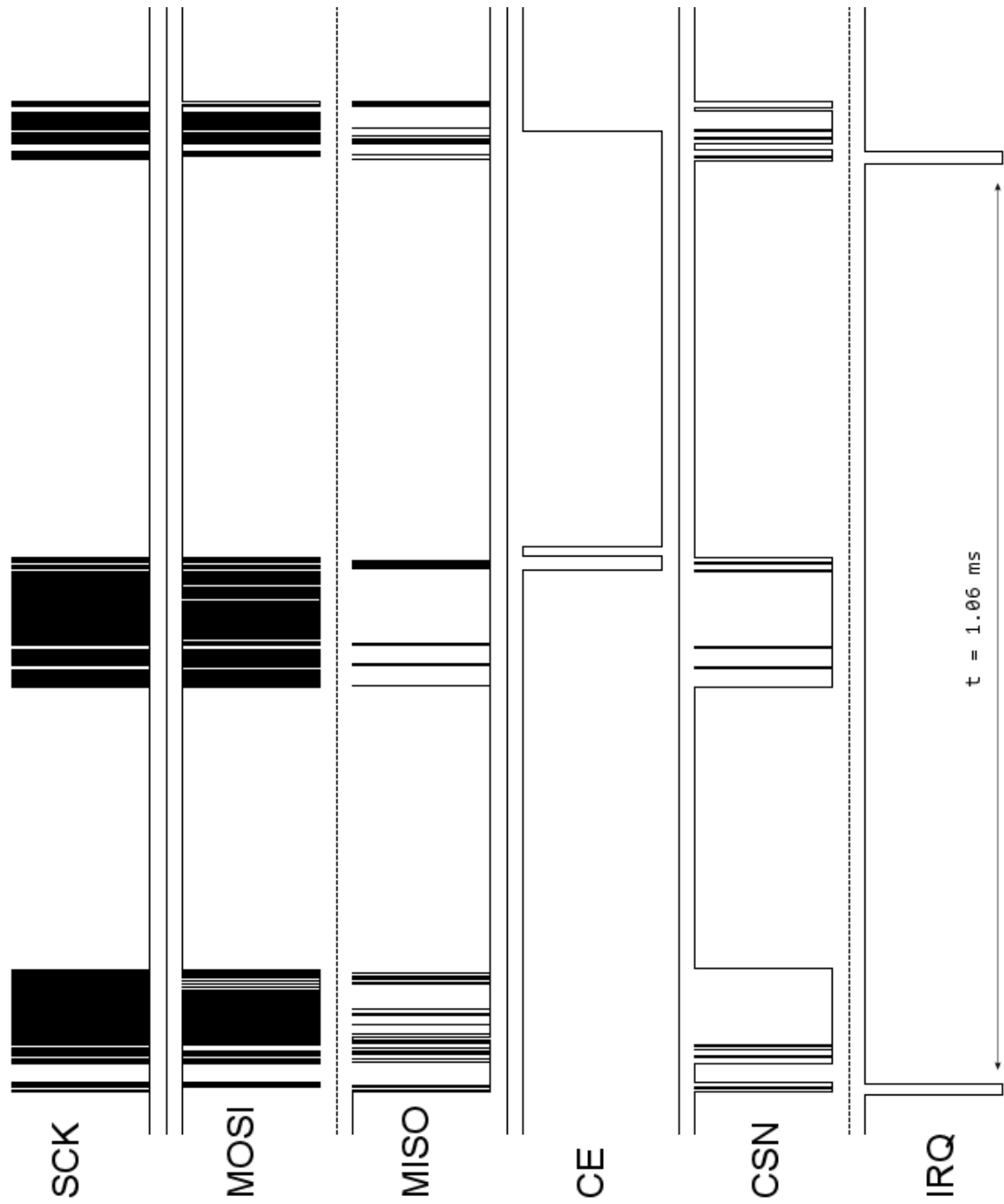
Liite 11. Verkon toimintaetäisyys siirtonopeudella 2 Mbps



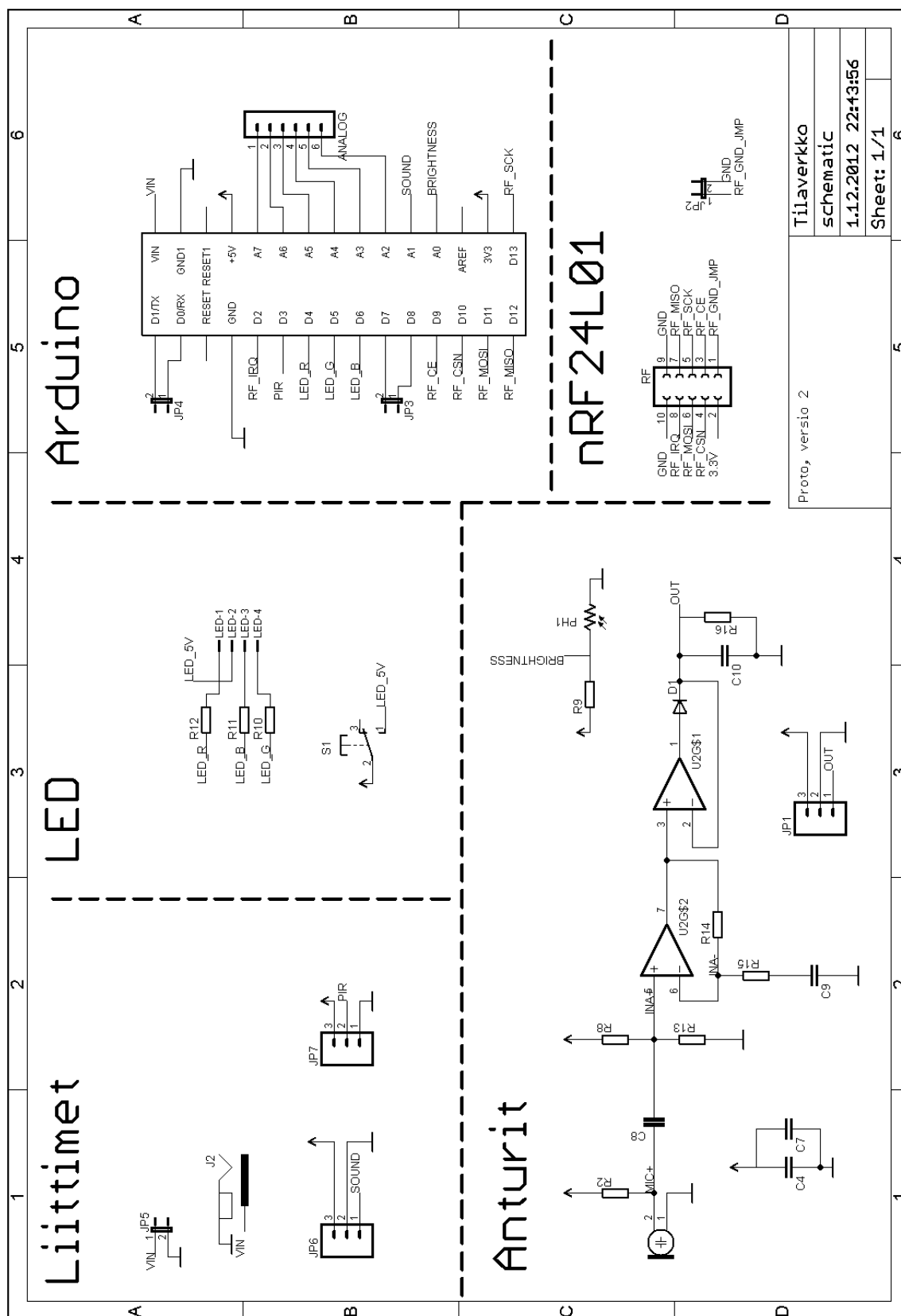
Liite 12. Verkon viestin välittämiseen kuluva aika nopeudella 250 kbps



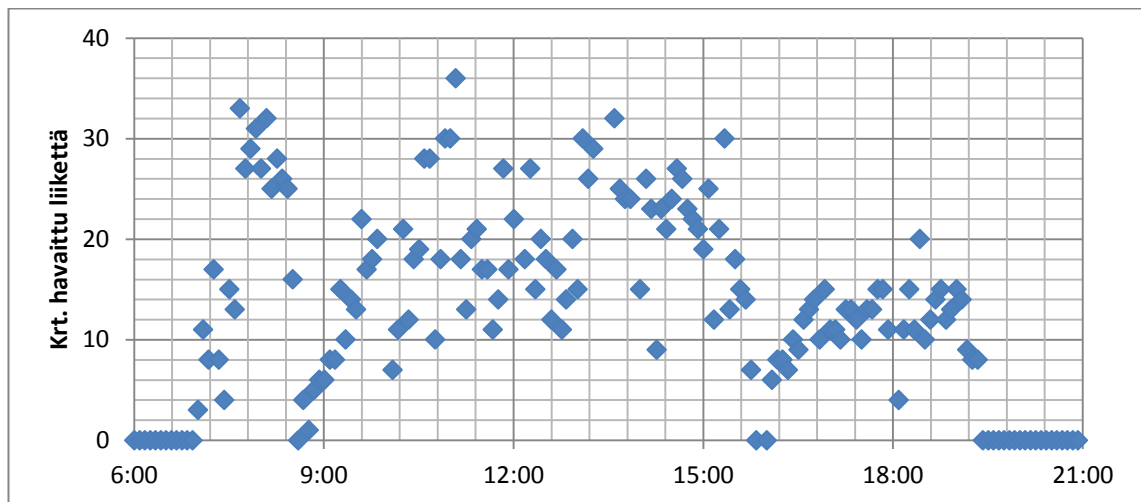
Liite 13. Verkon viestin välittämiseen kuluva aika nopeudella 2 Mbps



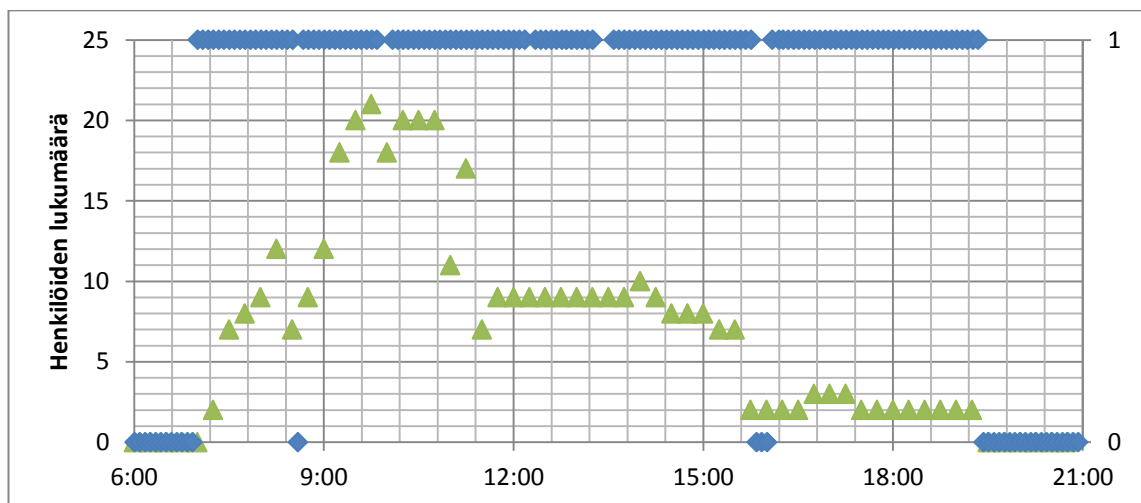
Liite 14. Prototyypin kytkentäkaavio



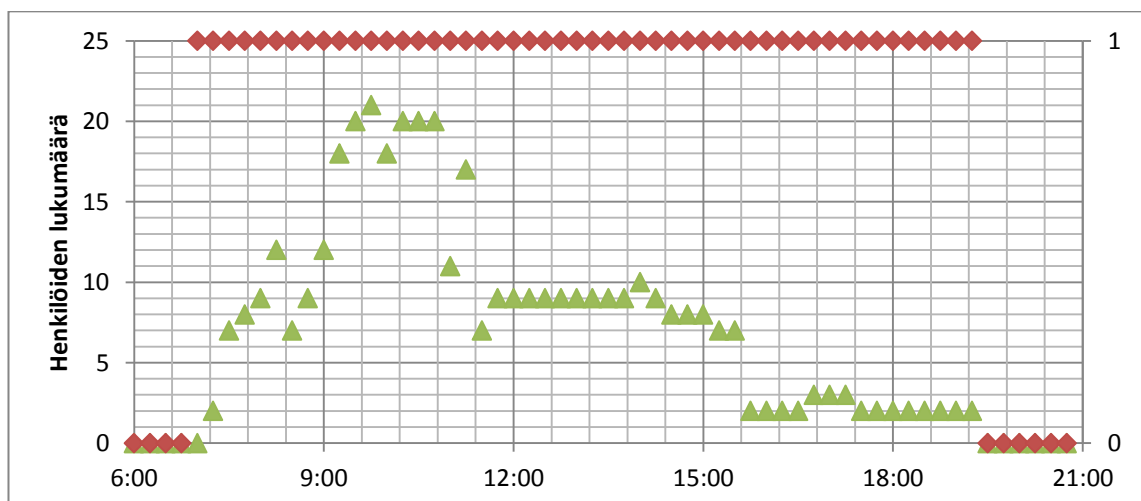
Liite 15. Liikeantureiden data



Kuva 1 Liikkeen määrä 5 minuutin sisällä

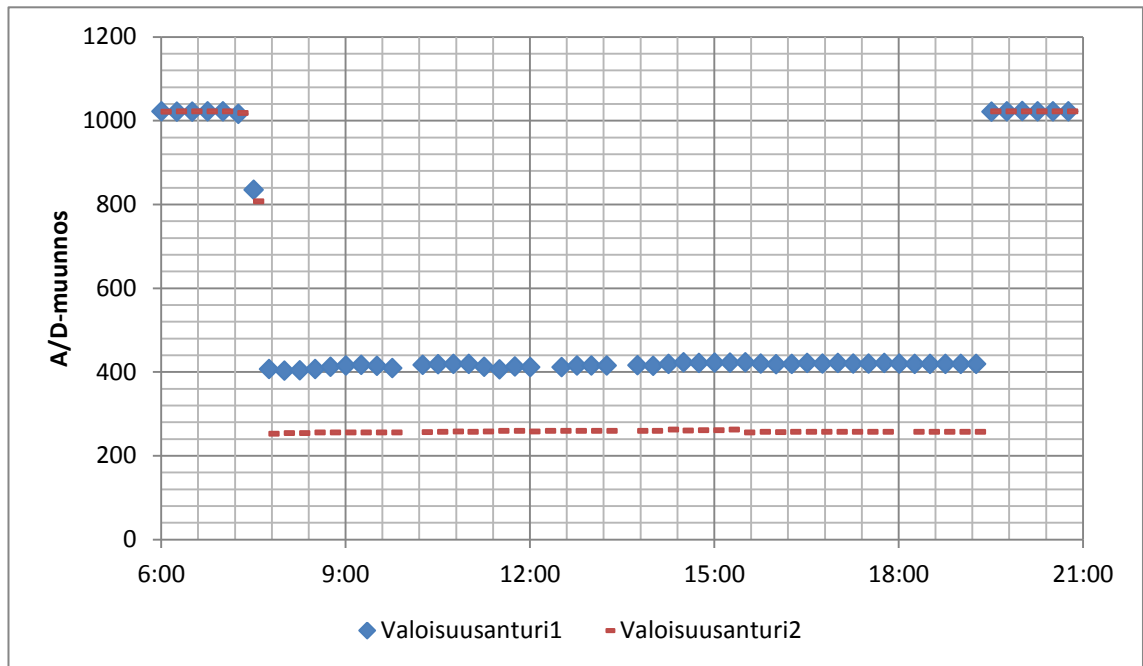


Kuva 2 Liikkeen havaitseminen 5 minuutin sisällä verrattuna henkilöiden lukumäärään

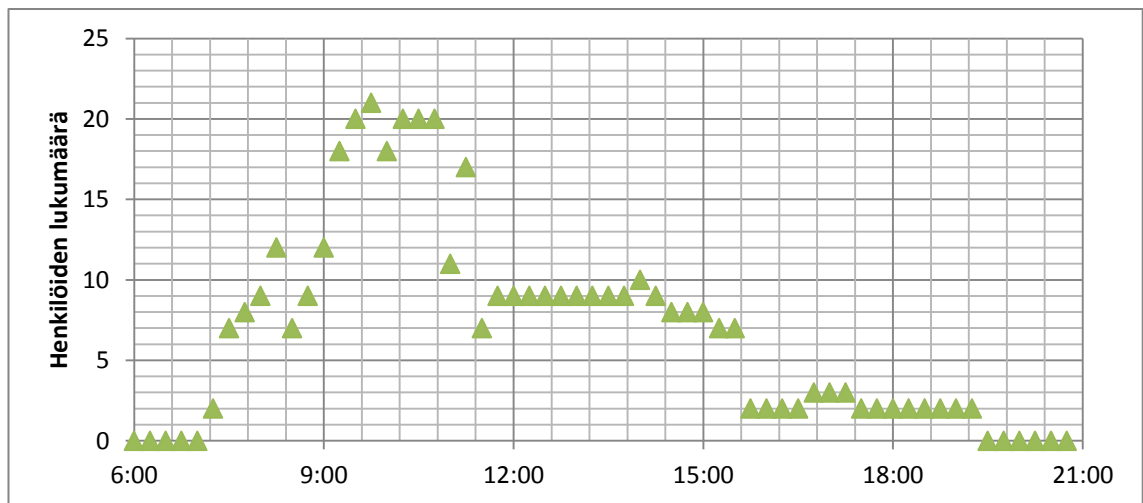


Kuva 3 Liikkeen havaitseminen 15 minuutin sisällä verrattuna henkilöiden lukumäärään

Liite 16. Valoisuusantureiden data

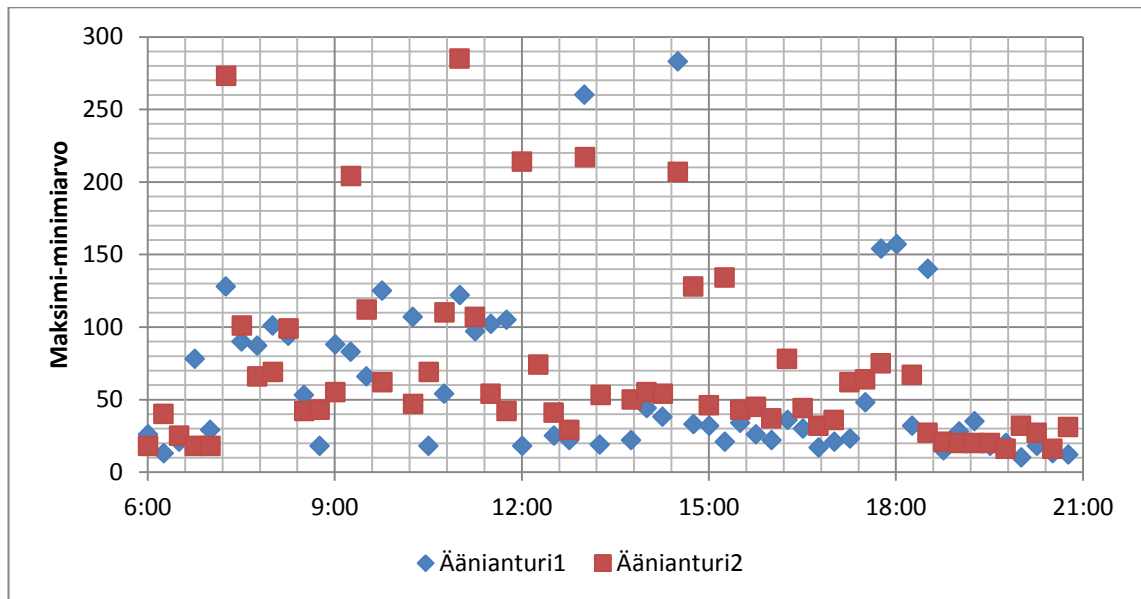


Kuva 1 Valoisuusantureiden mittaustiedot

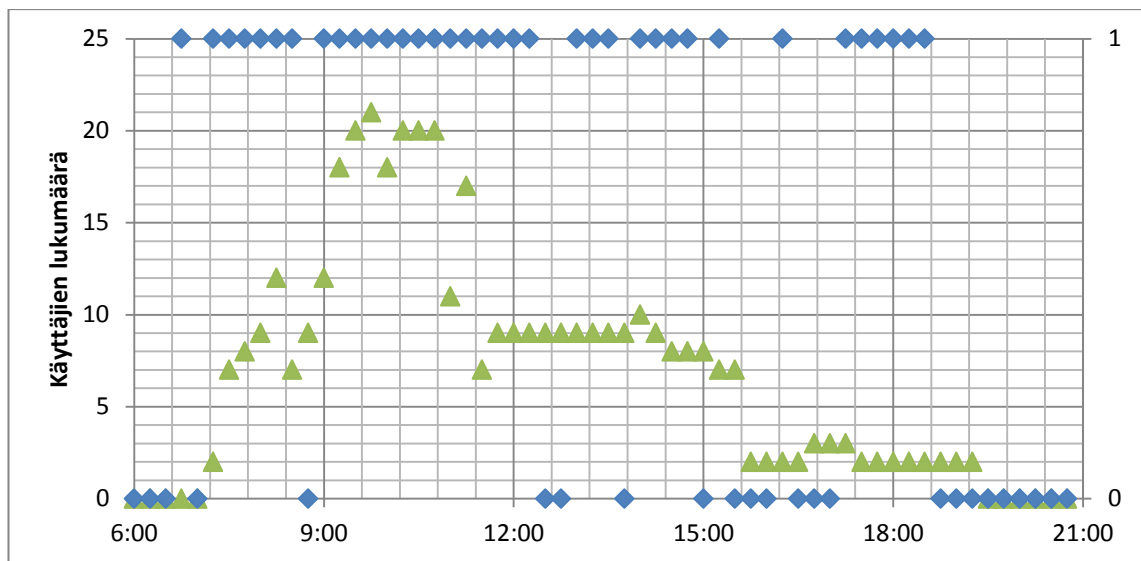


Kuva 2 Tilassa olleiden henkilöiden lukumäärä

Liite 17. Ääniantureiden data



Kuva 1 Ääniantureiden maksimi – minimi



Kuva 2 Ääniantureilla havaittu tilankäyttö