

Janne Meriläinen

Langattoman valonohjausjärjestelmän suunnittelu ja protoympäristön luominen

Puhelin, tietokone, jne..

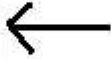
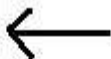


Raspberry Pi/Verkkosivut

wifi/kaapeli



Reititin



Ohjaus ZigBee

Valtavallo LED-valo

Solmu

Insinööri (AMK),

tietotekniikka

Kevät 2016



TIIVISTELMÄ

Tekijä: Meriläinen Janne

Työn nimi: Langattoman valonohjausjärjestelmän suunnittelu ja protoympäristön luominen

Tutkintonimike: Insinööri (AMK), tietotekniikka

Asiasanat: ZigBee, Raspberry Pi, langaton valonohjaus

Teknologian nopea kehittyminen on tuonut valtavasti mahdollisuuksia uusien sovellusten ja ominaisuuksien kehittämiseksi elektroniikkalaitteille. Etenkin langattomuus ja helppokäyttöisyys ovat koko ajan suuremmassa roolissa. Valaistuksen hallinnoiminen ei poikkea tästä trendistä.

Tämän työn tavoitteena oli suunnitella langaton valonohjausjärjestelmä ja luoda prototyyppi laitteistosta. Työn tilaajana toimi Valtavalo Oy ja tekopaikkana yrityksen tehdas Kajaanissa. Tehdas valmistaa energiatehokkaita ja laadukkaita LED-valoputkia.

Työn teoriaosuus sisältää syvällisen selvityksen ZigBee-standardista ja hieman suppeamman osuuden Raspberry Pi:stä. Itse työn suoritus sisältää suunnitelman järjestelmästä ja työn vaiheiden läpikäynnin. Työn osia ovat Raspberry Pi, käyttöympäristö, verkkosivut, testilaitteisto sekä lopullinen laitteisto.

Työssä suunniteltiin langaton valonohjausjärjestelmä, joka käyttää japanilaista ZigBee-radioverkkoteknologiaa. Raspberry Pi ohjaa radioverkkoa, jossa reititin hallitsee LED-valoja. Raspi pitää yllä verkkosivua, joka toimii käyttöympäristönä valonohjausjärjestelmälle. Työn lopputuloksena saatiin kattava suunnitelma järjestelmästä. Lisäksi prototyyppikokonaisuus valmistettiin osittain.

ABSTRACT

Author: Meriläinen Janne

Title of the Publication: Wireless lightning control system and creation of prototype

Degree Title: Bachelor of Engineering, Information technology

Keywords: ZigBee, Raspberry Pi, wireless light controlling

Development of technology in today's world has brought enormous amounts of growth in equipments additional features. Especially wirelessness and ease of use are getting bigger role all the time. Managing lighting doesn't differ from this trend.

The purpose of this thesis was to design wireless lightning control system and create a prototype environment for it. The commissioner of the thesis was Valtavalo Oy and place to do was their factory in Kajaani. Factory produces energy efficient and good quality LED light tubes.

The theory part of the thesis includes deep report of ZigBee standard and a tinier report of Raspberry Pi. The thesis includes a system plan and the study explains it. Parts of the work are Raspberry Pi, operating environment, website, test hardware and final hardware.

In this thesis wireless lightning control system was designed. The system uses Japanese ZigBee radio network technology. Raspberry Pi supervises radio network where gateway manages LED lights. Raspberry Pi maintains website which is operating environment for lightning control system. Final result of thesis brought comprehensive plan of the system. Also some parts of the prototype completed.

SISÄLLYSLUETTELO

1 JOHDANTO.....	1
2 ZIGBEE	2
2.1 Nimen alkuperä	2
2.2 Mikä on ZigBee	2
2.3 ZigBee Allianssi.....	4
2.4 Protokollapinin merkitys	7
2.5 IEEE 802.15.4	7
2.5.1 WPAN-verkon komponentit	7
2.5.2 CSMA-CA ja DSSS	8
2.5.3 Tietoturva	9
2.5.4 Fyysinen kerros.....	9
2.5.5 Siirtokerros	11
2.6 Verkkotopologiat.....	14
2.6.1 Tähtitopologia.....	14
2.6.2 Peer-to-peer-topologia	15
2.6.3 Puutopologia	16
2.6.4 Mesh-topologia.....	16
2.6.5 Reititys	17
2.6.6 Omatoimisuus ja vikasietoisuus	18
2.7 Ylemmät kerrokset	18
2.7.1 Verkkokerros	19
2.7.2 Sovelluskerros.....	19
2.7.3 Verkkoliikenne.....	20
3 RASPBERRY PI.....	21
3.1 Tekniikka	21
4 TYÖN SUORITTAMINEN.....	22
4.1 Lähtöselvitys.....	22
4.2 Järjestelmä.....	23
4.3 Verkkosivut.....	28
4.4 Raspberry Pin käyttöönotto	29
4.5 NetLed testilaitteisto	29

4.6 NetLed laitteisto.....	31
5 YHTEENVETO	34
LÄHDELUETTELO	35
LIITTEET	39

LYHENTEET

AES	Advanced Encryption Standard
APS	Application Support
ARQ	Automatic Repeat Request
BPSK	Binary Phase Shift Keying
CCA	Clear Channel Assessment
CSMA-CA	Carrier Sense Multiple Access With Collision Avoidance
DSSS	Direct Sequence Spread Spectrum
EDM	Energy Detection Measurement
FFD	Full Function Device
IEEE	Institute of Electrical and Electronics Engineers
LLC	Logical Link Layer
LQI	Link Quality Indication
LR-WPAN	Low-rate Wireless Personal Area Network
MAC	Media Access Control
OSI	Open Systems Interconnection
O-QPSK	Offset Quadrature Phase-Shift Keying
PD-SAP	Physical Data Service Access Point
PHY	Physical Layer
PHY PIB	Physical Personal Area Network Information Base
PLME	Physical Layer Management Entity
PPDU	Physical Protocol Data Unit
RFD	Reduced Function Device
RPI	Raspberry Pi
SAP	Service Access Point
SSH	Secure Shell
WANET	Wireless Ad hoc Network
WPAN	Wireless Personal Area Network
ZC	ZigBee Coordinator
ZDO	ZigBee Device Object
ZED	ZigBee End Device
ZR	ZigBee Router

1 JOHDANTO

Työn tilaajana toimi Valtavalo Oy. Valtavalo on vuonna 2008 perustettu yritys, joka valmistaa laadukkaita sekä energiatehokkaita LED-valaisimia. Yrityksen henkilöstöpuoli sijaitsee Oulussa ja tehdas Kajaanissa. Sain opinnäytetyön ollessani työharjoittelussa Kajaanin tehtaalla.

Valtavalon päätuotteisiin kuuluvat LED-valoputket, joiden tarkoitus on korvata vanhanaikaiset loisteputket. Putkien ominaisuuksiin kuuluu korkea valoteho, pitkä ikä ja energiatehokkuus. Yritys lupaa putkilleen 125 tuhannen tunnin käyttöiän ja jopa 7 vuoden takuun. Valoputkia saa standardimitoissa välillä 45 - 150 senttimetriä. Värvaihtoehtoja on kaksi: luonnonvalkoinen ja hieman sinertävämpi versio. Lisäksi valojen päällä olevan kuvun voi valita kahdesta: kirkaasta ja haaleammasta. Valtavalon sisältöön kuuluu myös muita valaistustuotteita.

Opinnäytetyön idea oli lähteä suunnittelemaan kyseisille valoille langatonta, muokattavaa ja jopa automaattista ohjausta. Tarkoituksena oli pystyä hallitsemaan isoa valokokonaisuutta, esimerkiksi ison parkkihallin, valaistusta muualtakin kuin pelkistä katkaisimista ja säätämään sen toimintoja tarpeen mukaan lisäämällä antureita sekä muuta automatiikkaa. Valojen ohjauksen tulisi olla yksinkertaista ja helppoa.

Sovimme työn tarkoitukseksi lähteä rakentamaan systeemiä Raspberry Pin ympäristöön. RPI ohjaisi ZigBee-reititintä ja valot toimisivat ZigBeen solmukohtina. Työn suurimmaksi haasteeksi ei muodostunut kuitenkaan itse minitietokone vaan Aasian suunnalta tullut laitteisto.

Tulevassa tekstissä esitellään teoriaa ZigBeestä sekä Raspberry Pistä. Tämän jälkeen siirrytään työn tapahtumiin.

2 ZIGBEE

2.1 Nimen alkuperä

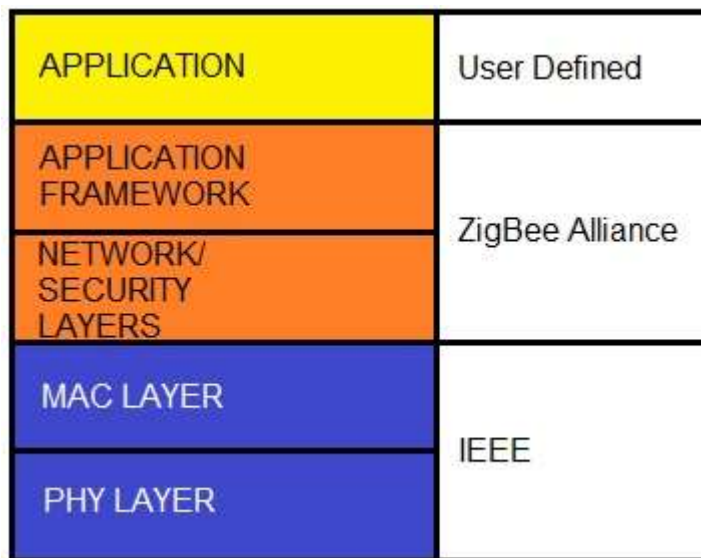
Tapa, jolla mehiläiset viestittävät toisilleen uudesta ruokapaikasta, on verrattavissa ZigBee-periaatteeseen. Äänetön ja tehokas ”siksakki”-tanssi, jota mehiläiset käyttävät lentäessään kukasta kukkaan, kertoo mehiläisyhteiskunnalle tietoja ruokalähteen sijainnista, suunnasta ja etäisyydestä. Zig viittaa termiin zigzag, eli siksakki ja bee on suomeksi mehiläinen. Tämä mehiläisten tapa viestiä on suoraan verrattavissa ZigBeen mesh-topologiaan. [4.]

2.2 Mikä on ZigBee

ZigBee on lyhyen kantaman vähän virtaa kuluttava radiotaajuusteknologia, joka on erityisesti kehitetty langattomia automaatio- ja ohjausjärjestelmiä varten. Tarkoituksena oli luoda alhaisella tehonkulutuksella ja pienillä valmistuskustannuksilla varustettu langaton verkkoprotokolla, jonka pääprioriteetit ovat luotettavuus, turvallisuus ja reaktioaika. Vaatimus alhaisesta tehonkulutuksesta on johtanut ZigBeessä ilmenevään matalaan siirtonopeuteen ja suhteellisen lyhyeen kantamaan. Pitkää kantamaa ei tarvita, koska yhdessä ZigBee-verkossa voi olla tuhansia laitteita, jotka kaikki voivat jakaa tietoa toisilleen. Näin ollen laitteesta laitteeseen -tekniikka mahdollistaa pitkän kantaman. [1.][2.][7.]

ZigBeen kehitys aloitettiin 2000-luvulla ja sen ensimmäinen versio julkaistiin vuonna 2004. Sen kaksi alinta kerrosta rakentuvat IEEE 802.15.4 -standardin vähävirtaisen WPAN:n PHY- ja MAC-kerroksiin. Ylemmät kerrokset määrittelee ZigBee Allianssi (kuva 1). Kerrokset kokonaisuudessaan perustuvat OSI-referenssimalliin (kuva 2). OSI on ohjemalli protokollapinolle. Vuonna 2003 julkaistiin ensimmäinen versio IEEE 802.15.4 -standardista. Se suunniteltiin vastaavin lähtökohdin kuin ZigBee. [1.][2.][7.]

ZigBee-tyylisten itsenäisesti toimivien digitaalisten radioverkkojen kehitys aloitettiin 1990-luvulla. IEEE 802.15.4-2003 -ZigBee ratifioitiin 2004-luvun lopussa. ZigBee Allianssi julkaisi saatavuuden spesifikaatio 1.0:lle kesäkuussa 2005. Tämä spesifikaatio tunnetaan nimellä ZigBee 2004 –spesifikaatio, ja ensimmäisen ZigBee-version nimi näin ollen on ZigBee 2004. Syyskuussa 2006 julkaistiin ZigBee 2006 -spesifikaatio vanhentaen vuoden 2004 version. On myös julkaistu Zigbee 2007 -versio. Tämä versio on yhteensopiva edeltäjänsä kanssa, sisältäen vain pieniä muutoksia aikaisempaan. ZigBee 2007:n kanssa samaan aikaan kehitettiin ZigBee Pro, joka ei varsinaisesti ole oma versio, vaan sisältää lisäominaisuuksia. [1.][2.][7.]



Kuva 1. ZigBeen protokollapino



Kuva 2. OSI-malli [8]

2.3 ZigBee Allianssi

ZigBee Allianssi on perustettu vuonna 2002. Se on ryhmä yrityksiä, jotka ylläpitävät ja kehittävät ZigBee-standardia. ZigBee on tämän ryhmän rekisteröity tavaramerkki. Allianssi julkaisee ohjelmistoprofiileja, jotka mahdollistavat yhteensopivuuden erinäisten myyjien tuotteille. Yhteys IEEE 802.15.4:n ja ZigBee Allianssin välillä on samanlainen kuin IEEE 802.11:n ja Wi-Fi Allianssin välillä.[7.]

ZigBee Allianssin tavoite on vaikuttaa maailmanlaajuiseen näkökulmaan. He toimivat yhdessä kehittäen standardeja, jotka luovat vapautta ja joustavuutta älykkäämmälle ja kestävämmälle maailmalle.

Allianssin tulos: [9.]

- Ympäristöystävällinen, vähävirtainen ja avoin maailmanlaajuinen langaton verkotusstandardi, joka keskittyy valvonta-, hallinnointi- ja sensoriohjelmiin.
- Vähävirtaisen standardin johdosta tuotteet toimivat pitkään kierrätysenergialla tai pattereilla/akuilla.
- Mahdollista yhdistää erityyppisiä laitteita samaan verkkoon.
- Laitteiden kommunikointi eri ympäristöissä ja ympäri maailman.
- Laadukkaat tuotteet ZigBee-sertifikaatin kautta.

Jos haluaa käyttää ZigBee-teknologiaa tuotteissaan, on jäsenyys ZigBee Allianssissa pakollista. Jäsenyys varmistetaan vuosimaksuilla. Jäsenyyden tasoja on kolme:

Adopter(~3700e): Pääsy viimeisiin hyväksytyihin spesifikaatioihin, käyttöoikeudet ZigBee-jäsenlogoon, osallistumisoikeus yhteentoimivuustapahtumiin ja pääsy työryhmädokumentteihin ja kehitystoimintaan.

Participant(~9200e): Tarjoaa täyden osallisuuden Allianssin komiteassa, työryhmissä ja jäsentapaamisissa, äänestysoikeustyöryhmissä sekä omistaa aikaisen pääsyn kaikkiin Allianssin standardeihin ja kehitysvaiheessa oleviin spesifikaatioihin.

Promoter(~51200e + liittymismaksu ~23300e): Automaattinen äänestysoikeus kaikissa työryhmissä, viimeiset hyväksymisoikeudet kaikkiin standardeihin ja paikka Allianssiohjaajien hallituksessa.

Taulukossa 1 on tarkennettu jäsenyystasojen oikeuksia.

Taulukko 1. Jäsenyystasojen oikeudet [10]

ZigBee Alliance Member Benefits	Promoter *\$55,000 USD/year	Participant \$9,900 USD/year	Adopter \$4,000 USD/year
Receive a seat on the Board of Directors	✓		
Ratify proposed specifications	✓		
Automatic Voting Rights in all work groups	✓		
Right of first refusal on Open House Platinum sponsorships	✓		
Contribute input to requirements documents	✓	✓	
Contribute intellectual property (IP) to specification(s) and provide input into the specification evolution	✓	✓	
Gain early access to ZigBee specifications, be the first to market	✓	✓	
Receive access to final, ratified ZigBee specification(s)	✓	✓	✓
Eligible as candidate for Technical Committee	✓	✓	
Propose committees and work items	✓	✓	
Attend committee meetings	✓	✓	
Participate and vote in work groups	✓	✓	
Vote on proposed specifications	✓	✓	
Eligible to chair work groups	✓	✓	
Attend ZigBee Alliance face-to-face member meetings	✓	✓	
Attend Alliance interoperability test events	✓	✓	✓
Attend Alliance workshops and developers conferences	✓	✓	✓
Participate in marketing events	✓	✓	✓
Participate in press-release development	✓		
Attend and exhibit at Alliance trade show booths	✓	✓	✓
Use ZigBee Alliance logo (within usage guidelines)	✓	✓	✓
Receive ZigBee Alliance marketing collateral	✓	✓	✓
Access ZigBee Alliance Members' Area web pages	✓	✓	
Access ZigBee Alliance Adopters Area web pages			✓
Access Alliance e-mail reflectors	✓	✓	
Access to work group e-mail reflectors, teleconferences and documents	✓	✓	
Receive ZigBee Alliance Member Newsletter	✓	✓	✓

2.4 Protokollapinon merkitys

OSI-malli on kehitetty helpottamaan ymmärrystä ja suunnittelua verkkoympäristöjen luomista varten. Se ei itsessään ole protokolla vaan ohjemalli, kuinka kommunikointi eri systeemien välillä onnistuu vaihtamatta raudan ja ohjelmistojen logiikkaa. Pääperiaate on rakentaa kerroksittainen pino, jonka jokaisella kerroksella on oma tehtävänsä. Pinon alempi kerros tarjoaa palveluja yhtä kerrosta ylemmäs ja ylempi kerros käyttää yhtä kerrosta alemman palveluja. [8.][11.]

2.5 IEEE 802.15.4

IEEE 802.15.4 on standardi, joka määrittelee PHY- ja MAC-kerroksen pieninopeuksiselle langattomalle likiverkolle (LR-WPAN). Sitä ylläpitää IEEE 802.15 -työryhmä, joka julkaisi sen 2003. Standardi on perusta ZigBee-, ISA100.11a-, WirelessHART- ja MiWi-spesifikaatioille, jotka kaikki laajentavat standardia kehittämällä ylempiä kerroksia protokollapinoon. [1.][3.][12.]

LR-WPAN keskittyy alhaiseen hintaan, energiankulutukseen ja nopeuteen. Tätä yhteyttä voi verrata Wi-Fi-yhteyteen, joka tarjoaa enemmän nopeutta samalla vieden enemmän energiaa. Muita ominaisuuksia ovat mm. luotettava tiedonsiirto ja helppo käyttöönotto. [1.][3.][12.]

Standardista on tullut myös uudempia versioita, mutta koska ZigBee perustuu vanhempaan versioon, jätetään muut käsittelemättä. [1.][3.][12.]

2.5.1 WPAN-verkon komponentit

WPAN-verkko koostuu erilaisista komponenteista, joiden välillä ja kautta data kulkee. Laitteita on fyysisesti määriteltä kahta eri tyyppiä: FFD ja RFD. FFD-laite voi hyödyntää standardin kaikkia tehtäviä sekä toimintoja ja voi toimia kaikissa verkkotopologioissa. Se voi toimia normaalina solmukohtana, verkon

koordinaattorina ja keskuksena. RFD:n toimintoja on karsittu ja ne onkin tarkoitettu todella yksinkertaisiksi laitteiksi ollen näin kustannustehokas vaihtoehto. Ne eivät voi koskaan toimia koordinaattoreina ja voivat kommunikoida vain FFD:n kanssa. Toiminta on myös rajoitettu vain tähtitopologiaan. RFD:t yleensä nukkuvat suurimman osan ajasta taaten erittäin pienen virrankulutuksen. WPAN-verkko vaatii aina yhden FFD:n, joka toimii verkkokoordinaattorina. Laitteita yhdessä verkossa voi olla jopa 65 536 kappaletta. [3.][12.][13.]

ZigBee määrittelee eri termit verkkokomponenteille. ZC on käytännössä FFD-laite. Sillä on eniten toimintoja, ja se toimii verkon koordinaattorina. Se toimii verkon luotettavuuskeskuksena ja turvallisuusavainten säilönä. ZR on solmukohta, joka voi toimia toimintosolmuna ja datan välittäjänä. ZED-laite voi toimia vain päätelaitteena ja on verrattavissa RFD-laitteeseen. [7.]

2.5.2 CSMA-CA ja DSSS

CSMA-CA on tietoliikenteen siirtotien varausmenetelmä. Menetelmässä useat lähettävät laitteet voivat käyttää samaa kanavaa. Koska päällekkäin lähetetyt viestit aiheuttavat törmäyksellään häiriintymisen lähetyksessä tai vastaanotossa, pyritään menetelmällä estämään kahden samanaikaisen lähetyksen tapahtuminen. Vastaanotin kuuntelee mahdollista signaalia ja varmistaa kanavan vapauden ennen lähetystä. Kun siirtotie on vapaa, pakettitiedosto lähetetään kokonaisuudessaan. [3.][14.]

Törmäysten välttäminen kahden laitteen aistiessa kanava vapaaksi on toteutettu yksinkertaisella menetelmällä. Kun laite havaitsee kanavan vapaaksi, odottaa se satunnaisen määrän aikaa ennen paketin lähettämistä. Eli kun kaksi laitetta huomaa kanavan vapaaksi, odottaa kumpikin hyvin todennäköisesti erimittaisen satunnaisen ajan. Törmäminen teoriassa on mahdollista, mutta epätodennäköistä. [3.][14.]

DSSS:ää eli suorasekvenssihajaspektritekniikkaa käytetään signaalin moduloimiseen. Signaali moduloidaan näennäissatunnaisella kohinalla. Signaalin

vastaanottaja palauttaa signaalin demoduloimalla käyttäen samaa PN-koodia millä itse modulointi tapahtui. [3.][14.]

2.5.3 Tietoturva

Tietoturvallisuus on ensiarvoisen tärkeää kaikissa verkkoon liittyvissä tapauksissa, eikä ZigBee poikkea näistä millään tavoin. Ulkopuolisten pääsy järjestelmään ei ole suotavaa, ja ajatus automaatiota hoitavien tapahtumien manipuloinnista on suuri riskitekijä. Kuten kaikissa radioverkoissa, viestien liikennettä voidaan hidastaa tai estää voimakkaalla kohinalla tai haittasignaaleilla. [1.][2.]

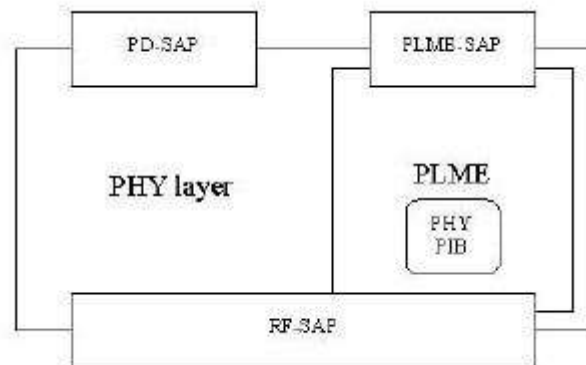
Todentamisella ZigBee-verkko valitsee, mitkä laitteet ovat hyväksytyjä. Jos laite kuuluu hyväksytyjen listaan, se on ns. luotettu laite. Verkko päästää läpi vain luotettujen laitteiden lähettämät paketit. Tämä tapa hylkää ulkopuolisten pakettien etenemisen. Ulkopuolisia laitteita voidaan hallita sulkemalla verkko, jolloin uusia laitteita ei oteta osaksi verkkoa. [1.][2.]

Salauksella estetään ulkopuolisten pääsy pakettien dataan, joita ZigBee-viestit sisältävät. ZigBee käyttää salausalgoritmia, jolla salataan lähetettävät ja puretaan vastaanotettavat viestit. Jotta salatut viestit olisivat saman kielisiä, täytyy algoritmin avain olla tunnettu solmujen välillä. Yleisin salaustapa on 128-bittinen AES. Verkkoon voidaan lisätä monia muitakin tietoturvallisuustekijöitä, mutta tällöin lähetettävien pakettien koko kasvaa ja seurauksena saattaa olla liian suuri virrankulutus tai viestien hidastuminen. [1.][2.]

2.5.4 Fyysinen kerros

Protokollapinon ensimmäinen ja alin kerros on fyysinen kerros. Tämä kerros tarjoaa rajapinnan MAC-kerroksen ja laitteistolta tulevan datan välillä. IEEE 802.15.4 -standardin fyysinen kerros tarjoaa ylemmille kerroksille kahden eri SAP:n kautta pääsyn kahdenlaisiin palveluihin. SAP on piste, jonka kautta protokollapinon kerrokset pyytävät tai tarjoavat palveluita viereisille kerroksille.

PD-SAP:n avulla ylempi kerros pääsee fyysisen kerroksen datapalveluun käsiksi. Tämä mahdollistaa pakettiyksikön (PPDU) lähettämisen ja vastaanottamisen fyysistä radiokanavaa pitkin. Hallintapalveluita kerroksesta kutsutaan fyysisen kerroksen hallintaoliolla (PLME), joka tarjoaa rajapinnat toimintoon ja ylläpitää tietokantaa (PHY PIB). Tietokannassa on fyysisen kerroksen hallitsemiseen tarvittavat attribuutit. Kuvassa 3 on esitetty malli fyysisen kerroksen rakenteesta. [3.][12.]



Kuva 3. Fyysisen kerroksen rakenne [3]

Muita fyysisen kerroksen tehtäviä: [3.]

- Radiolähettimen aktivointi ja deaktivointi.
- Saapuvan signaalin voimakkuuden arviointi (EDM). Verkkokerroksen kanavanvalinta-algoritmi käyttää tätä hyödykseen.
- Vastaanotettavien pakettien voimakkuuden ja laadun kuvaus (LQI). Verkkokerros tulee käyttämään tätä myöhemmin, mutta asiaa ei ole vielä määritelty.
- Tunnistus, onko tietty kanava jo toisen laitteen käytössä (CCA). Kolme erilaista toimintatapaa. Tunnistaa parhaassa tapauksessa kanavan varatuksi riippumatta sitä käyttävästä protokollasta. Toimintoa käytetään CSMA-CA:n kanssa.

Fyysinen kerros tarjoaa kolme vaihtoehtoista taajuutta jaoteltuina 27 kanavalle (taulukko 2). Taajuusalue 2,4-2,5 GHz on maailmanlaajuisesti vapaassa käytössä. 868 MHz taajuusalue on tarkoitettu Eurooppaan ja 915 MHz taajuusalue Pohjois- ja Etelä-Amerikkaan, Australiaan jne. [3.]

2,4-2,5 GHz taajuusalueen etuna globaalisuuden lisäksi on sen moninkertainen siirtonopeus. Matalammat taajuudet kuitenkin läpäisevät paremmin materiaaleja, mikä tapauskohtaisesti voi olla tarpeellista. [3.]

Suorasekvenssi hajaspektritekniikka(DSSS) on käytettävä ilmarajapinta kaikille taajuuksille. Modulaatio 868/915 MHz:n taajuuksille on BPSK ja 2,4 GHz:n taajuudella O-QPSK. [4.]

Taulukko 2. Kanavien ominaisuudet [3]

Taajuus	Kanavan numero	Siirtonopeus	Modulaatio
868 MHz	0	20kb/s	BPSK
915 MHz	1-10	40kb/s	BPSK
2450MHz	11-26	250kb/s	O-QPSK

2.5.5 Siirtokerros

OSI-mallin toinen kerros on siirtokerros, joka tarjoaa menettelytavat välittää tietoa tietoverkon kokonaisuuksien välillä (solmukohtat, päätelaitteet jne). Se voi tarjota myös virheenkorjausta virheille, jotka ovat tapahtuneet fyysisessä kerroksessa. Yleisesti siirtokerros koostuu kahdesta alakerroksesta: LLC:stä (Logical link control) ja MAC:stä (kuva 4). [15.]

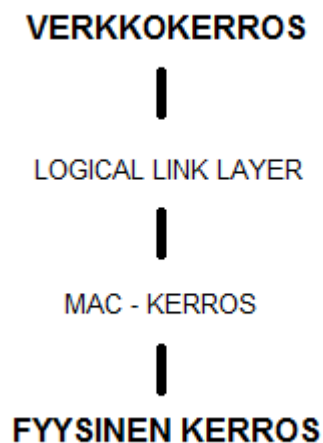


Kuva 4. Siirtokerroksen alakerrokset [15]

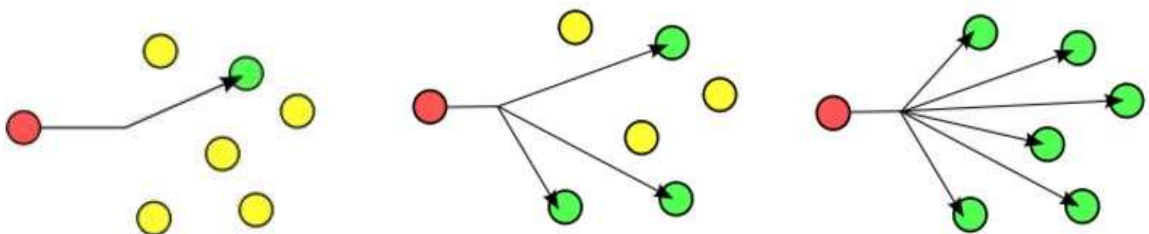
MAC-kerros tarjoaa osoitteistamista ja kanavanhallintatyökaluja mahdollistaen jaetun tiedon ohjausta verkon elimille. Kerros toimii rajapintana fyysisen kerroksen ja LLC-alakerroksen välillä (kuva 5). Yksinkertaistaen tämä osa emuloi kaksisuuntaisen loogisen kommunikoinnin kanavaa monisolmuoisessa verkossa. Kanava tarjoaa palvelua, jossa viesti lähetetään yhdelle (unicast), useammalle (multicast) tai kaikille verkon osille (broadcast) (kuva 6). [15.][16.]

MAC-kerroksessa tapahtuvat toiminnot: [16.]

- Kehyksen rajausta ja tunnistusta
- Kohdeaseman osoitteistaminen
- Lähdeaseman osoitetietojen luovuttaminen
- LLC:n avoimen datan siirto
- Virheitä vastaan suojautuminen
- Fyysisen lähetyksen kontrollointi



Kuva 5. Fyysisen kerroksen yhteys verkkokerrokseen



Kuva 6. Vasemmalta oikealle: unicast, multicast, broadcast. [17][18][19]

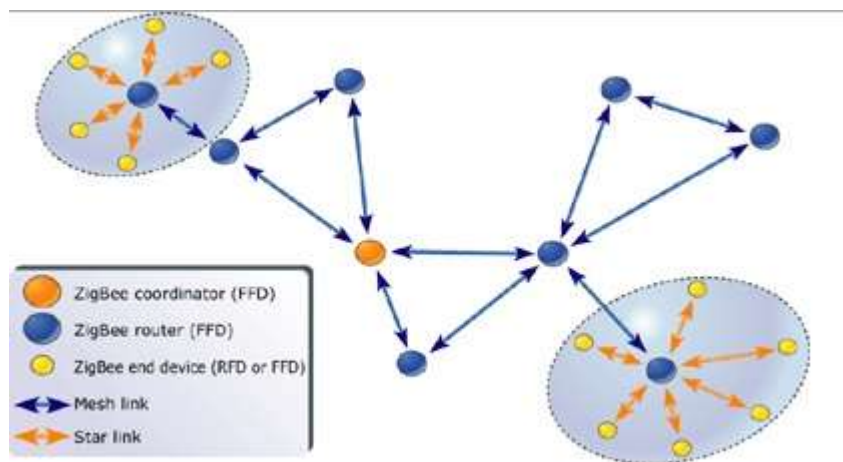
LLC tarjoaa multipleksaus-toimintoja, jotka mahdollistavat useiden verkkoprotokollien olemassaolon samassa monisolmuisessa verkossa. Verkkoprotokollia ovat mm. IP, IPX, Decnet ja Appletalk. Se voi myös tarjota tiedon sujuvaa hallintaa (flow control) ja ARQ-mekaniikkaa (automatic repeat request) virheiden hallintaan. LLC-alakerros toimii rajapintana MAC-alakerroksen ja verkkokerroksen välillä (kuva 5). [20.]

LLC:n päätehtävät: [20.]

- Lähetettävän protokollan multipleksaus ja vastaanotettavan koodin purkaminen.
- Solmulta solmulle sujuvan viestinnän tarjoaminen sekä virheiden hallinta.

2.6 Verkkotopologiat

ZigBee tukee monenlaisia verkkotopologioita, jotka määrittävät, kuinka verkon komponentit yhdistyvät toisiinsa. IEEE 802.15.4 -standardi itsessään määrittelee tähti- ja peer-to-peer-topologiat. Yleensä kuitenkin ZigBee-verkot ovat verkkotopologioiden yhdistelmiä, kuten puu- ja mesh-topologia ovat (kuva 7). [5.]

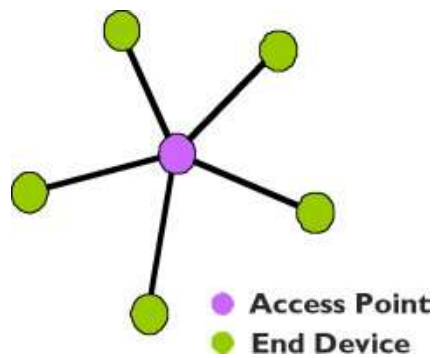


Kuva 7. Yhdistelmäverkko [21]

2.6.1 Tähtitopologia

Tähtitopologiassa verkko muodostaa tähtimäisen muodon (kuva 8). Verkon keskellä toimii verkkokoordinaattori eli FFD-laite, joka voi reitittää, aloittaa ja päättää tiedonsiirron etenemisen. Tämä FFD-laite on verkon sydän ja ainoa reitti

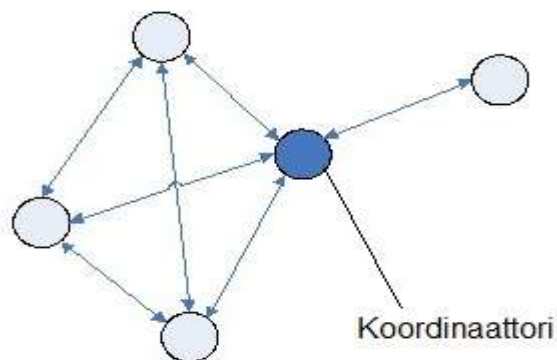
välittää tietoa muille solmukohtille. Muut solmukohtat eli RFD-laitteet ovat pääasiassa unessa ja ne ovatkin vain kommunikoinnin alku- tai päätepisteitä. [3.]



Kuva 8. Tähtimäinen topologia. [22]

2.6.2 Peer-to-peer-topologia

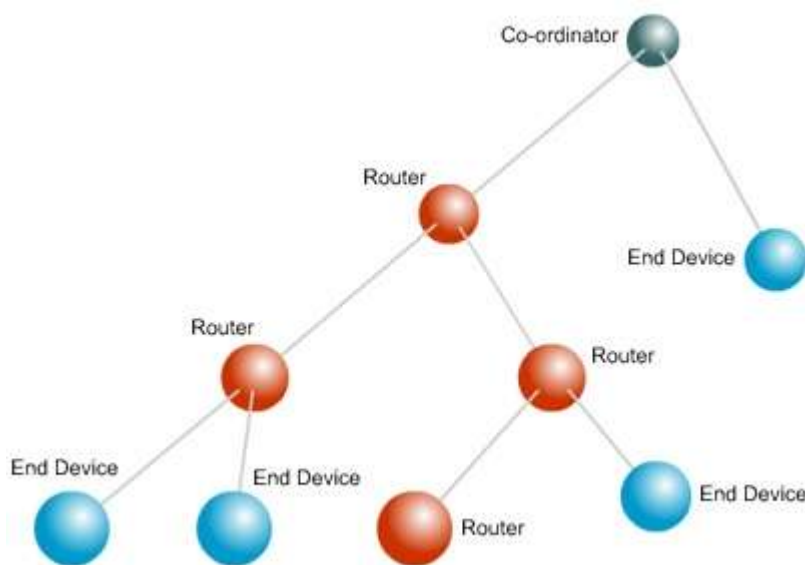
Peer-to-peer-verkon (kuva 9) rakentaa myös verkkokoordinaattori. Ero tähtitopologiaan verrattuna on päätelaitteet, jotka voivat lähettää tietoa muiden päätelaitteiden välillä kantomatkan sen salliessa. Näin ollen päätelaitteiden täytyy olla FFD-laitteita. [23.]



Kuva 9. Peer-to-peer-verkko. [24]

2.6.3 Puutopologia

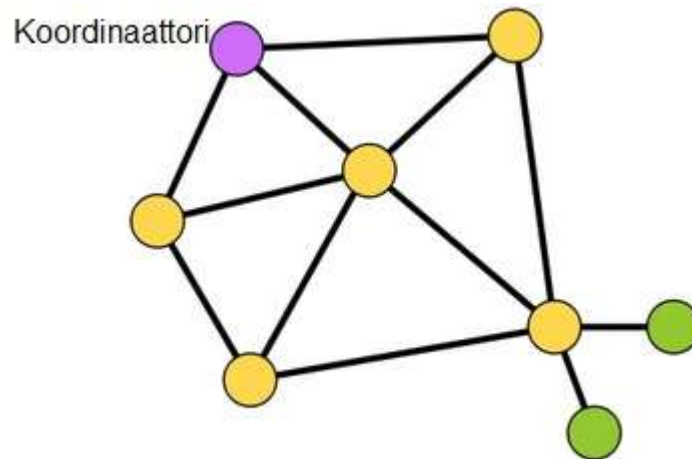
Puumainen verkko (kuva 10) käyttää ZigBee-termin ZR-laitetta solmukohtana muodostaen erilaisia tähtikytkentöjä puutopologian alahaaroihin. Tähtitopologiasta tutuksi tulleet pääte- eli ZED-laitteet ovat yhteydessä koordinaattoriin vain ZR-linkityksen kautta. Kaikkia ZR-laitteita ei ole yhdistetty toisiinsa ja tämä muodostaa puumaisen muodon. ZC rakentaa verkon ja on isäntä: FFD. Yleisesti puhuttaessa Puutopologia on laajennettu tähtitopologia. [25.]



Kuva 10. Puumainen topologia. [26]

2.6.4 Mesh-topologia

Mesh-verkotus (kuva 11) on kuin puutopologia, mutta erona kaikki ZR:t ovat yhteydessä toisiinsa. Yleisesti jokainen solmukohta on yhteydessä toisiinsa kantomatkan salliessa ja tieto voi kulkea esim. lyhintä tai mahdollista reittiä takaisin koordinaattorille. [27.]



Kuva 11. Mesh-topologia. [28]

2.6.5 Reititys

WANET (wireless ad hoc network) on ZigBeeen käyttämä menetelmä reitittämiseen. Menetelmä on monipuolinen ja muokattavissa, koska ad hoc ei vaadi valmista verkon rakennetta ja tiettyyn pisteeseen lukittuja laitteita. Verkon rakenteen muuttaminen on siis yksinkertaista: laitteita voi lisätä tai ottaa pois lennosta. [3.][6.]

Kun verkko ei sisällä valmiita reittejä reititystaulukossa tai kun vastaanottaja on radiokantaman ulkopuolella, aloittaa verkko reitityksen. Reititys aloitetaan reittipyynnöllä, joka on broadcast-viesti. Broadcast-viestin vastaanottajat tallentavat lähettäjän verkko-osoitteen, lisäävät kyseisen reitin hinta-arvoa (link cost) signaalivoimakkuuden perusteella ja välittävät viestiä puumaisesti eteenpäin kunnes päätepiste on saavutettu. Reiteistä valitaan paras reitin kokonaishinnan perusteella. Reitti tallennetaan pysyvästi reititystaulukkoon kun reitin päätepiste lähettää reittivastauksen takaisin unicast-viestinä. Signaalivoimakkuuden arvo kuvaa matkan pituutta. Päätepiste on mahdollista saavuttaa useita reittejä pitkin, sillä menetelmä käy puumaisesti läpi kaikki mahdolliset reitit päätepisteelle. Lyhintä ja nopeinta reittiä tullaan käyttämään muistista, kunnes jokin estää sen käytön (kuollut solmu, solmun vaihto jne.), jolloin reititys on tehtävä uudestaan. [3.][6.]

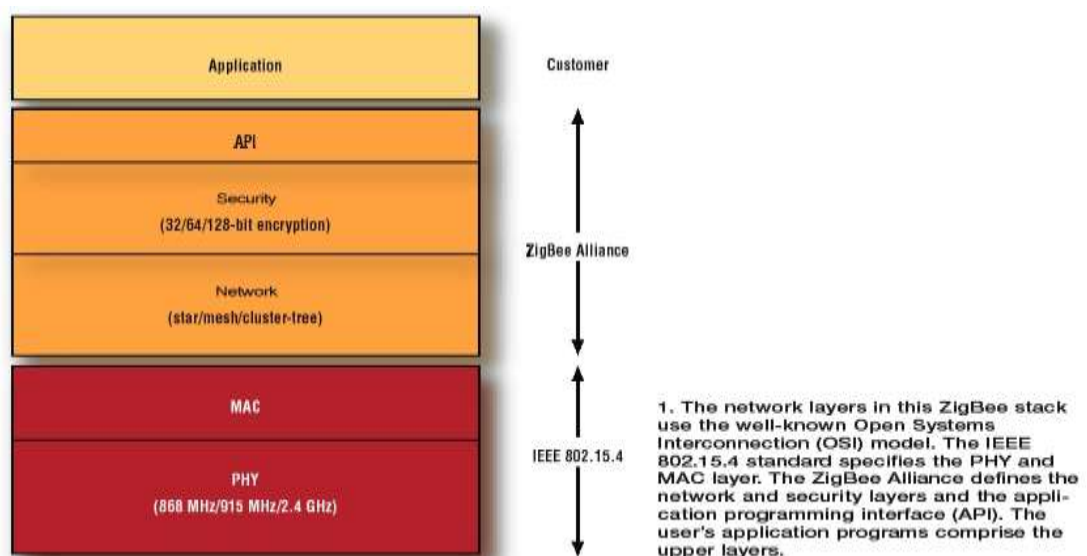
Reitittäminen on hidasta ja vaatii paljon viestiliikennettä, mutta valmiit reitit toimivat ilman ylläpitoviestejä. Kokonaisuudessaan reititystapa toimii pienellä muistimäärällä ja laskentateholla. [3.][6.]

2.6.6 Omatomisuus ja vikasietoisuus

ZigBee-verkolle on ominaista huomata vikatilanteet ja toipua niistä. Olennaista on, että verkko osaa toimia oma-aloitteisesti ja ilman ihmisen apua. Itseorganisointi (self-organizing) ja itsekorjautuvuus (self-healing) ovat luotettavuuden ja omatomisuuden kannalta tärkeitä ominaisuuksia. Itseorganisointi tarkoittaa ZigBee-verkon kykyä tunnistaa muita laitteita ympäristössä ja organisoida verkko rakenteelliseksi ja toimivaksi. Itsekorjautuvuuden tarkoitus on tunnistaa verkon solmujen tai kommunikaatiolinkkien vikatilanteet ja toipua niistä. [3.]

2.7 Ylemmät kerrokset

Zigbee Allianssin vastuulla on kuvassa 1 esitetyn protokollapinon kaksi ylemmää kerrosta. Kuvassa 12 on toinen esitystapa protokollapinosta.



Kuva 12. Protokollapino [29]

2.7.1 Verkkokerros

Verkkokerroksen päätehtävä on verkon toiminnallisuus: se pitää huolen datapakettien kuljettamisesta niiden oikeisiin kohteisiinsa sekä se luo verkkoa liittämällä uusia solmukohtia verkkoon, jos toiminto on hyväksytty. Jos solmukohta poistetaan, verkkokerroksen tehtävä on pitää verkko yhtenäisenä ja näinollen poistaa näkymättömiä haaroja. Tarkalleen verkon koordinaattorin verkkokerros on verkon muokkaamisen taustalla. Kerros toimii myös MAC-kerroksen tulkkina ja ympäristönä seuraavalle kerrokselle, sovelluskerrokselle. [7.][30.]

2.7.2 Sovelluskerros

Sovelluskerros on korkein spesifikaation määrittelemä kerros. Se on rajapinta ZigBee-järjestelmän ja lopullisen käyttäjän välillä. Kerros sisältää spesifikaation määrittelemät osat: ZigBee Device Objectin (ZDO), Application Support (APS) -alikerroksen ja Application Objectin. Viimeisin on valmistajan määrittelemä osa ja periaatteessa se voidaan ilmoittaa osaksi kerrosta. [7.]

ZDO:n tehtävä on määritellä laitteiden roolit, hallinnoida turvallisuusavaimia ja yleistä verkon käytäntöä. Se määrittelee onko laite koordinaattori vai päätelaite ja hallitsee uusien solmukohtien liittymistä. ZDO myös vastaa laitteiden turvallisesta yhdistämisestä erilaisten turvallisuusmetodien avulla. [7.]

APS-alikerros on toinen sovelluskerroksen pääkomponenteista. Kerros toimii siltana verkkokerroksen ja muiden sovelluskerroksen elementtien välillä. Se pitää yllä päivittyvää tietokantaa laitteista verkossa. Tietokannan avulla löydetään sopiva laite vaadituille toiminnoille. [7.]

2.7.3 Verkkoliikenne

ZigBee-verkot muodostuvat pääpiirteittäin kolmesta erilaisesta liikennetyypistä. Mitä liikennetyypistä käytetään riippuu ympäröivästä järjestelmästä ja sen vaatimuksista. Esimerkiksi valvontalaitteet lääketieteissä vaativat reaaliaikaisia mittaustuloksia. Vastakohtana valo ja himmentävä automaatio, jonka sensorit mittaavat päivänvalon vaikutusta sisätiloihin, voivat toimia hitaammin. [31.][32.]

Liikennetyypit: [31.][32.]

- Jaksoittainen liikenne:
 - Dataa lähetetään tietyin aikaväleihin, yleensä sovelluksen ohjauksella.
 - Datamäärät ovat yleensä pieniä.
 - Solmukohta nukkuu väliajat ja saavuttaa erittäin vähäisen energiankulutuksen.
- Ajoittainen liikenne:
 - Datan lähetys tapahtuu yleensä ulkoisesta ärsykkeestä ja on epäsäännöllistä.
 - Laite voi poistua verkosta vähentääkseen energiankulutusta. Ärsyke liittyy sen takaisin.
- Jatkuva liikenne:
 - Dataa lähetetään kokoajan pienellä vasteajalla.
 - Reaaliaikainen tarkkailu virrankulutuksen kustannuksella.

3 RASPBERRY PI

Raspberry Pi on minitietokone, jonka brittiläinen Raspberry Pi Foundation julkaisi vuonna 2012. Kyseessä on erittäin edullinen ja kooltaan pieni tietokone. Foundation rakensi sen kehittääkseen aikuisten ja lasten tietotekniikan osaamista eri osa-alueille. [33.]

3.1 Tekniikka

Raspberry Pi:stä on julkaistu useita eri versiota, mutta tässä käydään vain läpi kehittyneempi versio Model B (julkaistu 2.2.2015). Kyseinen malli on halpa ja sen hinta on vain 35 dollaria. Se perustuu Broadcom BCM2835 -järjestelmäpiiriin, jonka jatkeena on ARM Cortex-A7: 900MHz:n moniydinprosessori. RPi:n muistimäärä on yksi gigatavu ja sen näytönohjain Broadcom VideoCore. Se sisältää neljä kappaletta USB 2.0 -portteja, HDMI-videoulostulon; 3,5mm audiojakin, 10/100 Ethernet RJ45 -portin ja SD-muistikorttipaikan. Lisäksi laitteessa on GPIO-pinnit. Laitteelle tuodaan virta 5V MicroUSB:n kautta ja laitteen virrankulutus on 420mA. Käyttöjärjestelmät pohjautuvat pääasiassa Linux-kerneliin. Yleisin käyttöjärjestelmä useista mahdollisista on Raspbian, jota Raspberry Pi Foundation itsenäisesti ylläpitää. [33.]



Kuva 13. Raspberry Pi 2 Model B [34]

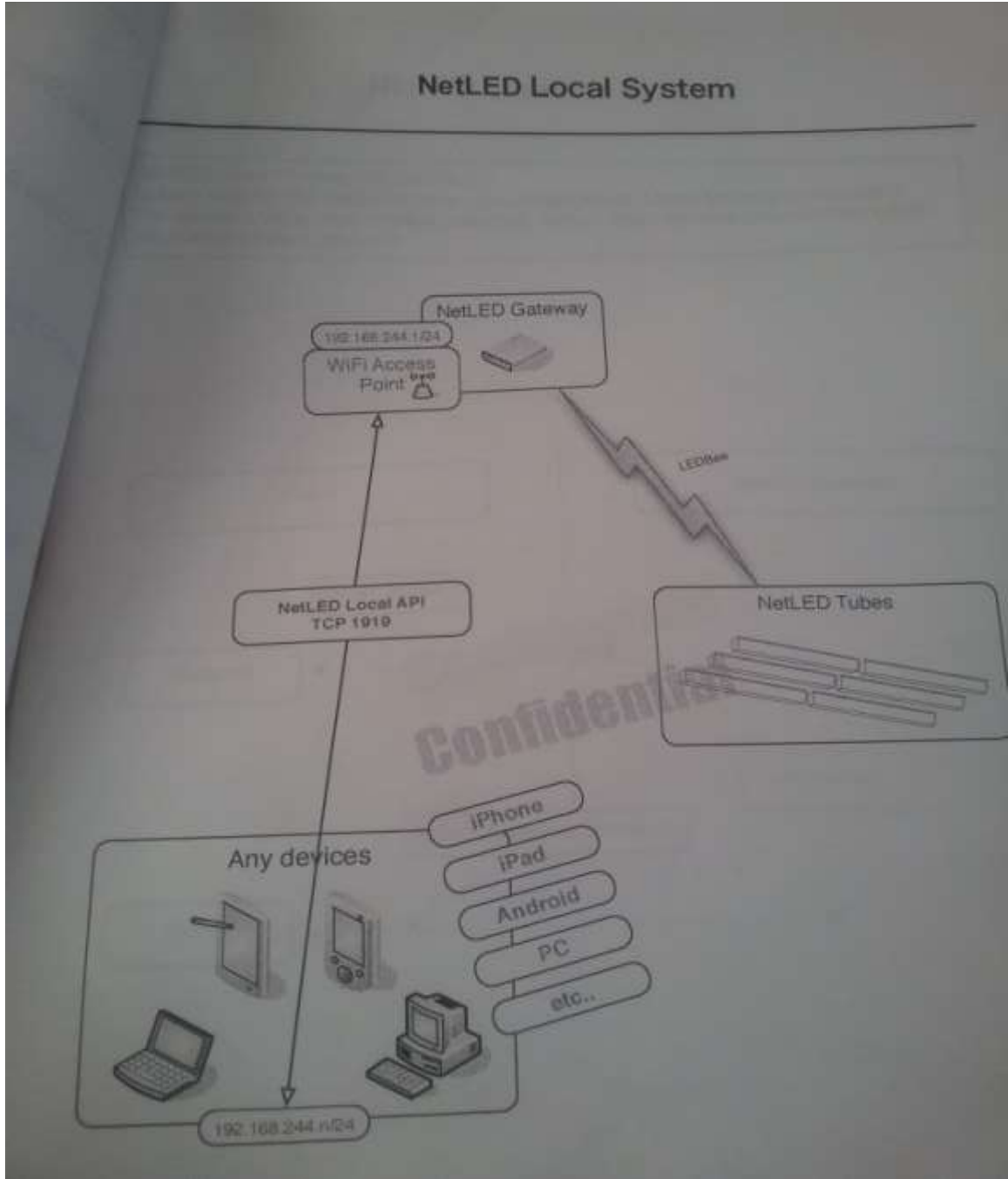
4 TYÖN SUORITTAMINEN

4.1 Lähtöselvitys

Työ polkaistiin käyntiin pienimuotoisella kokouksella, jossa sovimme projektin pääpiirteistä. Tarkoituksena oli lähteä suunnittelemaan ja luomaan laitteisto LED-tuotteen jatkeeksi. Laitteistolla pitäisi pystyä ohjaamaan ja hallitsemaan valaistusta etänä tai paikan päällä. Etenkin langattomuus ja energiatehokkuus olisivat tärkeitä ominaisuuksia. Palaverissa tuli ilmi Valtavalon hyvät suhteet Aasiaan etenkin Kiinaan, mutta myös Japaniin. Japanilainen yritys NetLed valmistaa edullista laitteistoa erityisesti LED-tuotteille. Laitteiston viestintä perustuu ZigBee-standardiin ja näin ollen tukisi energiatehokasta brändiä. NetLed-laitteiston ohjaus hoidettaisiin tietokoneella.

Jo vanha uutuuksena Raspberry Pi on ollut kovassa nosteessa uuden ajan minitietokoneista. Sen pieni koko ja suhteellisen nopea laskentateho yhdistettynä vähäiseen energiankulutukseen takaisi kaiken tarpeellisen ohjainlaitteena olemiseen. Raspberryn linux-ympäristö tukisi myös C++ -ohjelmointikieltä, joka on NetLed-laitteiston kieli. Tietokone voisi olla huomaamaton purkki katossa tai rakenteessa, johon otetaan yhteyttä tarpeen vaatiessa.

4.2 Järjestelmä



Kuva 14. NetLedin suunnitelma järjestelmästä

Järjestelmä on kokonaisuudessaan valoja hallinnoiva systeemi (kuva 14 ja 19). Systeemin radio-osan sydämenä toimii Gateway eli ZigBee-standardin reititin (kuva 15). Reititin on systeemin älyllinen elin ja viestien lähettäjä, vastaanottaja sekä käsittelijä. Kaikki halutut toiminnot radioverkossa koodataan reitittimelle. Muut komponentit radioverkossa ovat pääosin viestien vastaanotto- tai välityssolmuja. Nämä erittäin vähän virtaa kuluttavat ja usein uneksivat komponentit (kuva 16) ohjaavat valojen virtapiirejä ja mahdollisia sensoreita. Solmujen toiminnot ovat yleensä valo päälle/pois -tyyppisiä. Verkon komponenttien spesifikaatiot ja järjestelmän vaatimukset liitteenä.



Kuva 15. Radioverkon reititin.



Kuva 16. Radioverkon solmukohta.

Käyttäjä voi hallita valaistusta yksinkertaisella verkkosivulla, jota ylläpitää Raspberry Pi. Verkkosivu (kuva 17) sisältää kaikki tarpeelliset toiminnot, joita valojen ohjaamiseen tarvitaan ja on salasanasuojattu. Käyttäjä voi ottaa verkkosivuun yhteyttä etänä millä laitteella vain. Valojen hallinta onnistuu myös paikallisesti RPI-tietokoneella. Jatkokehityksessä konkreettiset katkaisijat sekä valvovat sensorit lisätään systeemiin.

Uusi tekstiasiakirja.html x

file:///G:/html/Uusi%20tekstiasiakirja.html

4walled Faceit Using ICC Profiles in \ ASUS VG248QE Review wirelessflyingcord:n k 1001F

Janne

Mon Apr 18 2016 00:25:44 GMT+0300 (FLE Daylight Time)

Info of the day The best sport

VALOTEHO.

100% ▾

Shut down Set

AJASTUS.

Valaisimien päälläoloaika (älä jätä arvoja tyhjäksi, oletusarvo = 0):

8 h

45 min

Syötä!

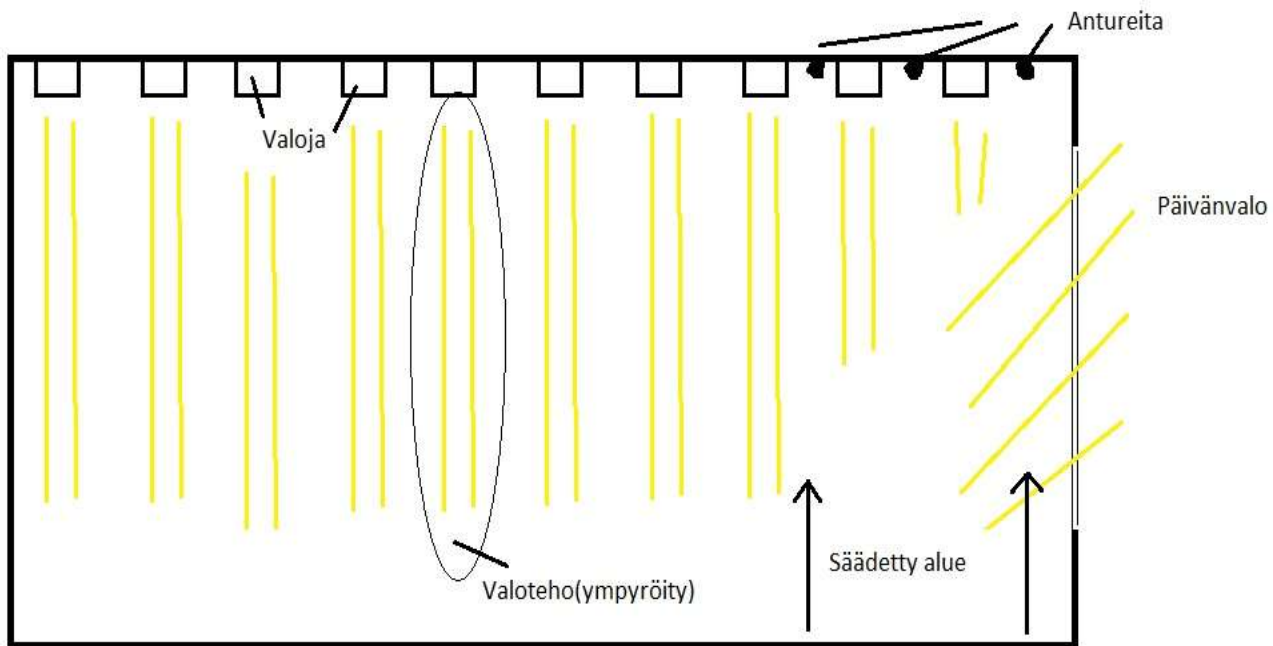
Lamput sammuu tänään klo: 9:10 31479

Start count 28 Stop count! Clear!

0.26.00

Kuva 17. Verkkosivujen raakaversio.

Luximääriä tarkkailevien antureiden tehtävä on vähentää virrankulutusta päiväsaikaan. Antureita sijoitellaan ikkunoiden tai muuta päivänvaloa sisään päästävien objektien läheisyyteen. Ne antavat reaaliaikaista tai ajastettua tietoa valon määrästä asennetuilla kohdilla. Tiedolla laitteisto automaattisesti hallinnoi alueen valojen tehokkuutta. Kun tilan haluttu luxitaso ja antureille annetut ohjearvot kohtaavat on laitteisto energiatehokkaimmillaan. (Kuva 18)



Kuva 18. Antureiden vaikutus.

Raspberry Pi:llä on pääasiassa kaksi tehtävää. Ensimmäinen tehtävistä on pyörittää verkkoympäristöä ja ottaa käyttäjän komentoja vastaan. Toinen tehtävä on ajaa radioverkon reitittimen ohjauskoodia ja antaa koodin välityksellä käskyjä reitittimelle. RPI on yhteydessä reitittimeen verkon kautta joko langattomasti tai langallisesti.



Kuva 19. Järjestelmän lohkokaavio.

4.3 Verkkosivut

Verkkosivujen tehtävä on mahdollistaa laitteiston ohjaus käyttäjälle paikasta riippumatta. Koska tungosta sivuille ei ole, pystyy yrityksen verkkoyhteys ja Raspberry Pi helposti pyörittämään palvelinta. Verkkosivut kasaavat toivotut toiminnot yhdeksi kokonaisuudeksi ja se toimii kuin tietokoneella pyörivä ohjelma. Verkkosivujen ohjelmointikieli on sekä html että javascript.

Vaaditut toiminnot:

- Valojen tehon säätö, sammutus
- Valojen ajastus: viikoittainen sekä väliaikainen ohjelma, poikkeustapaukset

- Jatkokehityksessä vikadiagnostiikka, antureiden säätö

Verkkosivujen teon piti olla yksi viimeisistä tehtävistä, mutta viivästykset ja ongelmat laitteiston kanssa siirsivät prioriteettitasoja. Sivusta valmistui erittäin raakaversio, johon on ns. hard-koodattu osioita. Liitteenä kuvassa 17 esiintyvän verkkosivun ohjelmointikoodi.

4.4 Raspberry Pin käyttöönotto

Raspberry Pin käyttöönotto lähti liikkeelle SD-kortin hankkimisella. SD-korttiin kirjoitettiin Windowsin kirjoitustyökalulla image-tiedosto, joka saatiin ladattua Raspberryn nettisivuilta. Kun RPI:hin laitettiin virrat SD-kortin ollessa paikallaan, lähti laite asentamaan käyttöjärjestelmää. Tässä vaiheessa laitteen nopeaa testausta lukuunottamatta se jätettiin rauhaan. Myöhemmin on tarkoitus siirtää reitittimen ohjauskoodi RPI:n linux-ympäristöön ja laittaa mahdollinen SSH-yhteys käyttöön.

4.5 NetLed testilaitteisto

NetLed lähetti karsitun laitteiston testattavaksi ennen lopullista kokonaisuutta. Paketti sisälsi hallinnointiohjelman Windowsille (kuva 20), karsitun reitittimen (kuva 21), solmukohdan (kuva 22) ja valoja imitoivan laitteiston (kuva 22). Laitteiston laatu päällepäin näytti hyvältä ja ohjelmakin lähti käyntiin ongelmitta. Epäonneksemme testatessamme laitteistoa sen toiminta oli hyvin epävakaata: joskus viestit menivät perille, toisinaan eivät. Suuria ongelmia tuotti myös solmukohtien yhdistäminen (connect) reitittimenä toimivaan USB-dongleen. Kuitenkin väliaikainen toiminta herätti toivoa. Päätimme tilata NetLedin lopullisen laitteiston.

Kuvassa 20 näkyvässä ohjelmassa keskellä on kaksi radioverkkoon yhdistettyä laitetta. Ohjelman yläosassa on erilaisia säätöjä laitteille: tehosäätö, mikä ledi palaa jne. Näytön vasemmassa alalaidassa toisesta valosta on laitettu palamaan kaikki valot ja taaemmasta vain yksi.



Kuva 20. Windows ohjelmisto, ledivalot.



Kuva 21. USB-dongle/reititin.

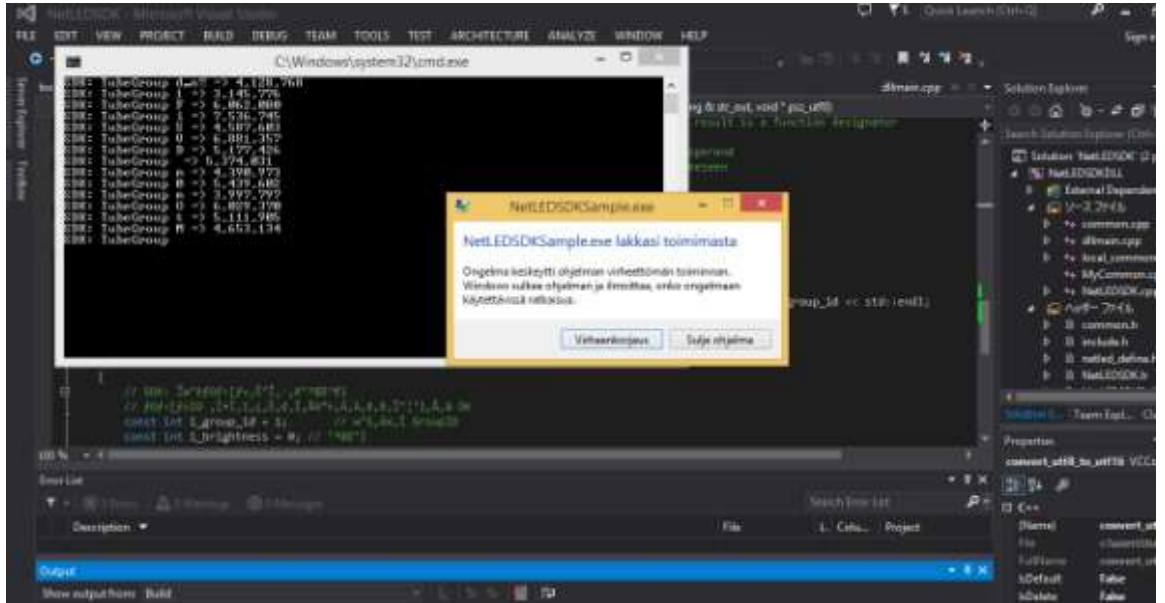


Kuva 22. Solmukohta ja ledivalo.

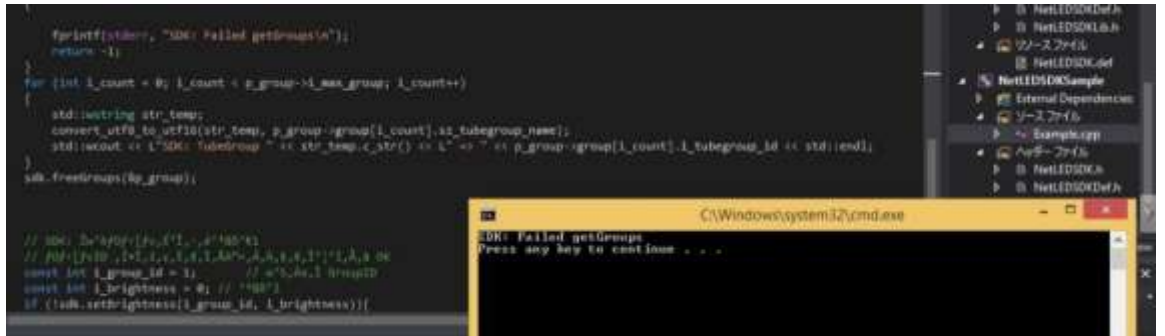
4.6 NetLed laitteisto

Lopullinen radioverkon laitteisto sisältää reitittimen (kuva 15), reitittimen C++ -ohjelmakoodin ja tietopaketin laitteistosta. Paketin piti olla täydellinen, toimiva ja muokattava kokonaisuus. Ajatus oli lisätä haluamme toiminnot C++ -koodiin ja liittää se rajapintana verkkosivujen toimintoihin. Koska koodin testaaminen, korjaileminen ja siirto Raspberry Pi:lle on oma projektinsa, lähdettiin ohjelmaa alustavasti ajamaan Windows-alustalla.

Ongelmia alkoi ilmetä jo ensimmäisellä testiajolla. Testiajon alkeellisimmassa tilanteessa reititin ja yksi solmukohta olivat kytkettyinä. Japanilaisen insinöörin kehittämä useita tuhansia, jopa kymmeniä tuhansia, rivejä sisältävä koodi ei toiminut halutulla tavalla. Microsoft Visual Studio ilmoitti ensimmäisellä testiajolla virheitä useita satoja kappaleita. Ohjelma myös kaatui jatkuvasti (kuva 23). Koodi oli tehty kehittyneellä tavalla (kuva 24), ja sen ymmärtäminen olikin erittäin vaikeaa. Ymmärtämisen lisäksi koodi piti saada toimimaan eli sitä piti debugata (kuva 25).



Kuva 23. Ongelmia ohjelman ajossa.



Kuva 24. Pätkä koodista.

```

//stat = WSAGetLastError();
//if (stat != 0)

if (connect(i_socket, (struct sockaddr *) &addr, sizeof(addr)) < 0)
{
    while ((i_result = select(i_socket + 1, &readfds, &writefds, NULL, &timeout)) != 0)
    {
        if (FD_ISSET(i_socket, &writefds))
        {
            break;
        }
        usleep(1000);
    }
    if (i_result <= 0) {
        shutdown(i_socket, 2);
        closesocket(i_socket);
        return -1;
    }
    i_value = 0;
}
return i_socket; //i_value);

```

Name	Value	Type
i_value	1	unsigned
i_result	0	int
i_count	-858993460	int
i_socket	172	int
writefds	{fd_count=0 fd_array=0x0077788 (172, 3415973836, 3415973836, 3415973836)}	fd_set
stat	-1	int
response	{dw_total_size=3415973836 dw_checksum=3415973836 b_cmd=204 ...}	_netled_

Kuva 25. Koodin debuggausta.

Ajan kuluessa koodin virheet vähenivät ja oppi C++ -kielestä lisääntyi, mutta lopullinen toiminta oli tavoittelemattomissa. Projektin aikana olimme useita kertoja yhteydessä Japaniin ja saimmekin sieltä apua. He lähettivät erilaisia kooditiedostoja, mutta niistä saatu edistys oli lähinnä näennäistä. Suomen ja Japanin aikaero tuotti myös omat ongelmansa. Pitkien keskustelujen jälkeen saimme sovittua ajan, milloin Netledin insinööri voisi kaapata työtietokoneen etänä ja katsoa mikä laitteistossa on vikana. Kaappauksia oli kokonaisuudessaan kolme kappaletta, mutta jokaisen yrityksen jälkeen olimme samassa tilanteessa. Edes koodin suunnitellut insinööri ei saanut laitteistoa toimimaan.

Projektin haastavimmaksi vaiheeksi muodostui siis laitteisto-ongelmat. Emme päässeet missään vaiheessa tekemään omia toimintoja koodiin ja jatkokehittämään tuotetta, vaikka suunnitelmat niistä oli valmiina. Jos olisimme saaneet koodin toimimaan ja valmiiksi Windowsilla, olisi seuraava vaihe ollut koodin siirtäminen Raspberry Pin linux-ympäristöön. Siirtäminen olisi varmasti vaatinut myös oman työnsä.

5 YHTEENVETO

Tämän työn tavoitteena oli suunnitella langaton valonohjausjärjestelmä ja rakentaa prototyyppiympäristö siitä.

Lähtökohtaisesti työn sisältö oli erittäin laaja, vaativa ja kunnianhimoinen. Sen yksittäiset osa-alueet vaativat paljon työtä ja testausta ennen niiden yhteen sovittamista. Yllätyksenä tuli kuinka paljon laajempi projekti olikaan, mitä se paperilla vaikutti. Yksinkertaisesti aika ja taito loppuivat kesken. Käteen jäi kuitenkin erittäin paljon kokemusta työelämästä ja projektinhallinnasta. Yksittäiset osa-alueet avasivat kokonaiskuvan hallintaa ja auttoivat keskittymään oikeisiin asioihin kerrallaan. Työn pystyi jaksottamaan eri päiville resurssien ja laitteiston saatavuuden mukaan. Stressinhallintakyky kasvoi ja ajattelu muuttui järjestelmällisemmäksi. Kokonaisuutena paketti oli erittäin laaja ja hyvä oppimisympäristö nuorelle insinöörille vaikei haluttuun lopputulokseen laitteiston osalta aivan päästykään.

LÄHDELUETTELO

[1] ZigBee Alliance, ZigBee Specification, ZigBee Document 053474r17, 2007

[2] Gislason, D, ZigBee Wireless Networking, Newnes, 2002

[3] IEEE, 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), 2003.

[4] Electrical Contractor, verkkosivu, saatavissa:

<http://www.ecmag.com/section/systemAs/buzz-zigbee>

[5] ZigBee Allianssi. ZigBee Alliance Press Kit (Compact Disc). May-August 2004.

[6] Perkins, C. Belding-Royer, E, & Das, S, Ad hoc On-Demand Distance Vector (AODV) Routing, 2003

[7] Wikipedia EN, ZigBee, verkkosivu, saatavissa:

<https://en.wikipedia.org/wiki/ZigBee>

[8] Wikipedia FI, OSI-malli, verkkosivu, saatavissa:

<http://fi.wikipedia.org/wiki/OSI-malli>

[9] ZigBee, ZigBee Alliance, verkkosivu, saatavissa:

<http://www.zigbee.org/zigbeealliance/>

[10] ZigBee, Membership Levels, verkkosivu, saatavissa:

<http://www.zigbee.org/zigbeealliance/join/>

[11] Wikipedia EN, OSI model, verkkosivu, saatavissa:

http://en.wikipedia.org/wiki/OSI_model

[12] Wikipedia EN, IEEE 802.15.4, verkkosivu, saatavissa:

http://en.wikipedia.org/wiki/IEEE_802.15.4

[13] Wikipedia FI, ZigBee, verkkosivu, saatavissa:

<http://fi.wikipedia.org/wiki/ZigBee>

[14] Wikipedia EN, Carrier sense multiple access with collision avoidance, verkkosivu, saatavissa:

http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_avoidance

[15] Wikipedia EN, Data link layer, verkkosivu, saatavissa:

https://en.wikipedia.org/wiki/Data_link_layer

[16] Wikipedia EN, Media access control, verkkosivu, saatavissa:

https://en.wikipedia.org/wiki/Media_access_control

[17] Wikipedia EN, Unicast, verkkosivu, saatavissa:

<https://en.wikipedia.org/wiki/Unicast>

[18] Wikipedia EN, Multicast, verkkosivu, saatavissa:

<https://en.wikipedia.org/wiki/Multicast>

[19] Wikipedia EN, Broadcastin (networking), verkkosivu, saatavissa:

[https://en.wikipedia.org/wiki/Broadcasting_\(networking\)](https://en.wikipedia.org/wiki/Broadcasting_(networking))

[20] Wikipedia EN, Logical link control, verkkosivu, saatavissa:

https://en.wikipedia.org/wiki/Logical_link_control

[21] Eet, gif-kuva, verkkosivu, saatavissa:

<http://m.eet.com/media/1041982/0405feat2fig3.gif>

[22] Sensorsmag, jpg-kuva, verkkosivu, saatavissa:

<http://www.sensorsmag.com/files/sensor/nodes/2007/1435/Figure2.jpg>

[23] Wikipedia FI, Vertaisverkko, verkkosivu, saatavissa:

<http://fi.wikipedia.org/wiki/Vertaisverkko>

[24] Ifn.et.tu-dresden, jpg-kuva, verkkosivu, saatavissa:

<http://www.ifn.et.tu-dresden.de/~marandin/ZigBee/ZigBeeTutorial.files/image001.jpg>

[25] Sassi 2009, s. 19-20, Jennic

[26] Jennic, Tree Topology, verkkosivu, saatavissa:

<http://www.jennic.com/elearning/zigbee/files/html/module2/module2-5.htm>

[27] Wikipedia EN, Mesh networking, verkkosivu, saatavissa:

http://en.wikipedia.org/wiki/Mesh_networking

[28] Sensorsmag, jpg-kuva, verkkosivu, saatavissa:

<http://www.sensorsmag.com/networking-communications/wireless-sensor/what-a-mesh-part-1-the-ins-and-outs-of-mesh-networking-1530>

[29] Electronicdesign, gif-kuva, verkkosivu, saatavissa:

http://electronicdesign.com/site-files/electronicdesign.com/files/archive/electronicdesign.com/files/29/11687/figure_01.gif

[30] Kinney, P. ZigBee Technology: Wireless Control that Simply Works. Kinney Consulting LLC, 2003.

www.zigbee.org/resources/documents/ZigBee_Technology_Sept2003.doc

[31] Craig, W. Zigbee: "Wireless Control That Simply Works". ZMD America Inc. , 2004.

http://www.zigbee.org/resources/documents/2004_ZigBee_CDC-P810_Craig_Paper.pdf

[32] Frenzel, L. Wireless Control That Simply Works, julkaisussa ZigBee Buzz Is Growing. ZigBee Alliance, 2004.

http://www.elecdesign.com/Files/29/7186/7186_01.pdf

[33] Wikipedia EN, Raspberry Pi, verkkosivu, saatavissa:

https://en.wikipedia.org/wiki/Raspberry_Pi

[34] Wikipedia EN, jpg-kuva, verkkosivu, saatavissa:

https://fi.wikipedia.org/wiki/Raspberry_Pi#/media/File:Raspberry_Pi_2_Model_B_v1.1_top_new.jpg

LIITTEET

Liite 1: Vaatimusmäärittely – Langaton valonohjausjärjestelmä

Liite 2: Reitittimen spesifikaatiot

Liite 3: Solmukohtan spesifikaatiot

Liite 4: Sensorin spesifikaatiot

Liite 5: Verkkosivujen ohjelmakoodi

Valonhallinnan laajuus:

- jopa tuhansia putkia

Kantomatka:

- jopa yli kilometri, solmukohtien yli

Säätöominaisuudet:

- on/off
- himmennys
- ryhmien määrittäminen
- aikataulujen määrittäminen
- ajastus

Huollettavuus:

- helppohuoltoinen
- tarpeen vaatiessa

Luotettavuus:

- virheetön toiminta
- teknisesti kestävä

Turvallisuus:

- hyökkäysten estäminen
- kirjautumispalvelu

Käyttäjävälisyys:

- helppokäyttöinen
- iPhone, Android, Windows -tuki

Automaation määrä:

- valontarpeen mukaan säätävä
- liiketunnistus

Tarve antureille:

- kyllä

Käyttöympäristö:

- -40°C-60°C
- pölyinen
- auringon valo
- kosteus
- ahdas
- tärinä
- häiriösaiteily
- signaalin häiriöt
- paikat, joissa ei suotavaa käyttää langatonta verkkoa(mm.sairaalat)

Verkon nopeus

- kevyt tiedonsiirto

Virrankulutus

- vähäinen

OS:	Linux
Protocol:	LedBee™
Wireless Standard:	IEEE802.15.4
Frequency:	2.4GHz
Channels:	16 channels
Control:	O-PQSK, DSSS
Encryption:	AES-128bit

Max number of endpoints: 200 units



The gateway connects to the Internet and allows communication to the wirelessly enabled lights from the cloud platform. Multiple gateways can be used collectively to scale to any size.

In a Local Version, the gateway can connect directly with a smartphone or tablet to do simple scheduling and control. A Hybrid Version allows both cloud and local connectivity to the gateway.

NLE002-PA: Internal antenna type. Range: 100m

NLE002-EA: External antenna type. Antennae separate. Range: 400m~1km

Controller:

32-bit CPU
32KB RAM
160KB Flash

Temperature Range: -40° c ~ +85° c

Power Consumption:

Sending: 15.3mA
Receiving: 17.0mA
Deep sleep: 100nA

Voltage: DC2.0V ~3.6V

Protocol: LedBee™
Encryption: AES-128bit
Circuits: 1~4 PWM circuits
Input/Output: A/D

Size: 31mm x 16mm



NLE002-PA

The NetLED Engine uses the proprietary protocol LedBee™ which is based on IEEE 802.15.4 Like Zigbee, it has multi-hop and is self-healing. It is a protocol optimized for lighting and sensors.

Protocol:	LedBee™
Wireless Standard:	IEEE802.15.4
Frequency:	2.4GHz
Encryption:	AES-128bit
Motion Sensor:	Panasonic EKMC1601111
Light Sensor:	Panasonic AMS302T
Other:	Analog sensor port



The greatest feature of this sensor is that it networked into the system and can be flexibly used to trigger many different types of events. The analog sensor port allows for other sensors such as those for temperature to be connected.

```
<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>

<style>

.classname {

    -moz-box-shadow:inset 0px 1px 0px 0px #ffffff;

    -webkit-box-shadow:inset 0px 1px 0px 0px #ffffff;

    box-shadow:inset 0px 1px 0px 0px #ffffff;

    background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #ededed), color-stop(1, #dfdfdf) );

    background:-moz-linear-gradient( center top, #ededed 5%, #dfdfdf 100% );

    filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#ededed', endColorstr='#dfdfdf');

    background-color:#ededed;

    -webkit-border-top-left-radius:41px;

    -moz-border-radius-topleft:41px;

    border-top-left-radius:41px;

    -webkit-border-top-right-radius:41px;

    -moz-border-radius-topright:41px;

    border-top-right-radius:41px;

    -webkit-border-bottom-right-radius:41px;

    -moz-border-radius-bottomright:41px;

    border-bottom-right-radius:41px;

    -webkit-border-bottom-left-radius:41px;

    -moz-border-radius-bottomleft:41px;

    border-bottom-left-radius:41px;

    text-indent:0px;
```

```
border:2px solid #dcdcdc;

display:inline-block;

color:#777777;

font-family:Courier New;

font-size:23px;

font-weight:bold;

font-style:normal;

height:54px;

line-height:54px;

width:100px;

text-decoration:none;

text-align:center;

text-shadow:1px 1px 0px #ffffff;

}

.classname:hover {

background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #dfdfff), color-stop(1, #ededed) );

background:-moz-linear-gradient( center top, #dfdfff 5%, #ededed 100% );

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#dfdfff', endColorstr='#ededed');

background-color:#dfdfff;

}.classname:active {

position:relative;

top:1px;

}

#counter, #LiveTimeX

{

margin-top:0px;

padding-top:0px;
```

```
float:left;

}

#counter

{

margin-left:1%;

}

.clear

{

clear:both;

}

.painike {

border-radius: 20px;

background-color: #E9D610;

border: 3px dotted maroon;

}

#start {

background-color: #10D610;

};

</style>

</head>

<body>

<a href="file:///C:/Users/tuoma_000/Desktop/Uusi%20tekstiasiakirja2.html" class="classname">Janne</a>
```

```
<p id="demo"></p>
```

```
<button type="button" onclick="Aika()">Info of the day</button>
```

```
<button type="button" onclick="testi()">The best sport</button>
```

```
<form>
```

```
<p>*****</p>
```

```
<p>VALOTEHO.</p>
```

```
<select id="mySelect">
```

```
<option>100%</option>
```

```
<option>75%</option>
```

```
<option>50%</option>
```

```
<option>25%</option>
```

```
</select>
```

```
<br><br>
```

```
<input type="button" onclick="disable()" value="Shut down">
```

```
<input type="button" onclick="enable()" value="Set">
```

```
<p>*****</p>
```

```
<p>AJASTUS.</p>
```

```
<form action = "" method = "get">
```

```
Valaisimien päälläoloaika (älä jätä arvoja tyhjäksi, oletusarvo = 0): <br>
```

```
<input type="text" name="Time" value="0"> h <br>
```

```
<input type="text" class="painike" name="Time2" value="0"> min <br>
```

```
<input type="button" onClicK="matti(this.form);" value="Syötä!"/>
```

```
</form>

<p id="LiveTimeX"></p><p id="counter"></p>

<p class="clear"/>

<!--<p id="asd"></p-->

<p>*****</p>

<input type="button" id="start" class="painike" value="Start count!" onclick="doTimer()" />

<input type="text" id="txt" />

<input type="button" class="painike" style="display: none"; value="Stop count!" onclick="stopCount()" />

<input type="button" value="Clear!" onclick="clearCount()" />

<script>

var loopon = false;

var counter = 0;

$(".painike").fadeOut(10000);

function matti(form)

{

var testii = parseFloat(form.Time.value);

var testii2 = parseFloat(form.Time2.value);

var x = document.getElementById("LiveTimeX");

var aika = new Date();

var h = testii + aika.getHours();

var m = testii2 + aika.getMinutes();

var tunnit_tulostus = 0;
```

```
var i = 0;
```

```
var paiva = 0;
```

```
////-----
```

```
if (m >= 60)
```

```
{
```

```
h = h + 1;
```

```
m = m - 60;
```

```
}
```

```
if (h >= 48)
```

```
{
```

```
i = i + 1;
```

```
}
```

```
if (h >= 24)
```

```
{
```

```
h = h - 24;
```

```
i = i + 1;
```

```
}
```

```
if (m <= 9)
```

```
{
```

```
m = "0" + m;
```

```
}
```

```
////-----
```

```
if (i >= 1 && h <= 24)
```

```
{  
  
x.innerHTML="Lamput sammuu huomenna klo: " + h + ":" + m;  
  
}  
  
else if (h >= 24)  
  
{  
  
h = h + 24;  
  
h = h * 60;  
  
h = h + m;  
  
  
while (h >= 60)  
  
{  
  
h = h - 60;  
  
tunnit_tulostus = tunnit_tulostus + 1;  
  
}  
  
while (tunnit_tulostus >= 24)  
  
{  
  
paiva = paiva + 1;  
  
tunnit_tulostus = tunnit_tulostus - 24;  
  
}  
  
x.innerHTML="Lamput sammuu " + paiva + ":n" + " päivän päästä      "+" klo: " + tunnit_tulostus + ":" + h;  
  
}  
  
else  
  
{  
  
x.innerHTML="Lamput sammuu tänään klo: " + h + ":" + m;  
  
}
```



```
counter = (testii * 3600) + (testii2 * 60);

if (!loopon)
{
myLoop();
}

function myLoop ()
{
    loopon = true;
    setTimeout(function ()
    {
        document.getElementById("counter").innerHTML=counter;

        counter--;

        if (counter > 0)
        {
            myLoop();
        }

        else loopon = false;
    }, 1000);
}

document.getElementById("counter").innerHTML=counter;
}

///-----CLOCK-----///
```

```
///
```

```
///-----  
  
var myVar=setInterval(function(){myTimer()},1000);  
  
function myTimer()  
  
{  
  
var d=new Date();  
  
var t=d.toLocaleTimeString();  
  
document.getElementById("LiveTime").innerHTML=t;  
  
}  
  
///-----  
  
//  
  
///-----  
  
  
  
  
///-----TIMER-----  
  
//  
  
///-----  
  
var c=0;  
  
var t;  
  
var timer_is_on=0;  
  
  
function timedCount()  
  
{  
  
document.getElementById('txt').value=c;  
  
c=c+1;  
  
t=setTimeout  
  
(function(){timedCount()},1000);  
  
}
```

```
function doTimer()

{

if (!timer_is_on)

{

timer_is_on=1;

timedCount();

}

}

function stopCount()

{

clearTimeout(t);

timer_is_on=0;

}

function clearCount()

{

c=0;

}

///-----///

//

///-----///

function Aika()

{

document.getElementById("demo").innerHTML = Date();

}
```

```
function testi()

{

    window.open("http://www.nhl.com");

}

///-----SAMMUTA/SÄÄDÄ VALOT-----///

                                     ///

///-----///

function disable()

{

    document.getElementById("mySelect").disabled=true;

}

function enable()

{

    document.getElementById("mySelect").disabled=false;

}

///-----///

                                     //

///-----///

///-----///

                                     //

///-----///

/*function Timer()

{

    var y = 0;

    while (y < 20)

    {
```

```
y++;  
  
for (var i=0;i<1000;i++)  
  
    {  
  
    }  
  
    document.getElementById("asd").innerHTML = ("Valot palaa" + y +" sekunttia <br> ASD <br>");  
  
}*/
```

```
///-----SLEEP-----///  
  
///                                     ///  
  
///-----///  
  
    /*for (var i=0;i<1000;i++)  
  
    {  
  
    }*/  
  
///-----///  
  
///                                     //  
  
///-----///
```

```
</script>
```

```
<p id="LiveTime"></p>
```

```
</body>
```

```
</html>
```