



Markus Oikarinen

KAHDEN ARDUINON VÄLINEN KOMMUNIKAATIO ZIGBEE- YHTEYTTÄ KÄYTTÄEN

KAHDEN ARDUINON VÄLINEN KOMMUNIKAATIO ZIGBEE- YHTEYTTÄ KÄYTTÄEN

Markus Oikarinen
Opinnäytetyö
29.5.2012
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

OULUN SEUDUN AMMATTIKORKEAKOULU TIIVISTELMÄ

Koulutusohjelma Tietotekniikan koulutusohjelma	Opinnäytetyö Insinööriö	Sivuja 29	Liitteitä 3
Suuntautumisvaihtoehto Langaton tietoliikenne	Aika 29.5.2012	+	
Työn tilaaja Teemu Korpela	Työn tekijä Markus Oikarinen		
Työn nimi Kahden Arduinon välinen kommunikaatio ZigBee-yhteyttä käyttäen			
Asiasanat Arduino, ZigBee, XBee			

Insinööriöön tarkoituksena oli tutkia Arduinoa ja sen XBee-lisäosaa sekä saada toimiva kommunikaatio kahden Arduinon välille. Työssä kerrotaan yleisesti Arduinosta ja sen toiminnasta. Arduino on avoimen lähdekoodin elektroniikka-alusta.

Työn tarkoituksena on auttaa niitä henkilöitä, jotka tulevaisuudessa tutkivat Arduinoa ja haluavat siihen langattoman ZigBee-yhteyden. Työn tilaajana toimi Oulun seudun ammattikorkeakoulun tekniikan yksikkö.

Työssä saatiin toteutettua kahden Arduinon välille kommunikaatio XBee-lisäosaa käyttäen. Vaikka XBeen valmistajan MaxStreamin luvattuihin kantavuuksiin ei päästyäkään, niin ZigBee-yhteys toimi hyvin.

Työ toteutettiin pääosin Oulun seudun ammattikorkeakoulun laboratoriotiloissa keväällä 2012. Työtä voidaan tämän jälkeen käyttää hyväksi Arduinon tutustumisessa ja se helpottaa Arduinoiden välisen langattoman yhteyden luomisessa.

OULU UNIVERSITY OF APPLIED SCIENCES ABSTRACT

Degree Programme Information Technology and Telecommunications	Thesis Bachelor's Thesis	Number of pages 29	Appendices 3
<hr/>		+	
Line Telecommunications	Date 29.5.2012	<hr/>	
Commissioned by Teemu Korpela	Author Markus Oikarinen	<hr/>	
Thesis title The communication of two Arduinos using ZigBee communication			
<hr/>			
Keywords Arduino, ZigBee, XBee			
<hr/>			

The purpose of this thesis work was to study Arduino and the XBee shield as well as to get a working communication between two Arduinos. In general the work describes the Arduino and how it works. The objective of the work is to help those people, who will explore Arduino in future and want to get a wireless ZigBee connection to it. The work was ordered from Oulu University of Applied Sciences.

In this work, two Arduinos got a wireless ZigBee connection using XBee shield. Although Xbee's manufacturer MaxStream promised that module can communicate further than we reached, the test was successful.

The work was implemented mainly in Oulu University of Applied Science's laboratory in the spring of 2012. After this work can be used to explore Arduino and it helps creating wireless communication between two Arduinos.

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
LYHENTEET ja TERMIT	6
1 JOHDANTO.....	7
2 ARDUINO	8
2.1 Historia	8
2.2 Arduino Mega 2560	8
2.3 Ohjelmistokehitysympäristö	10
2.4 FreeRTOS	11
3 ZIGBEE	12
3.1 Historia	12
3.2 ZigBee Alliance.....	12
3.3 IEEE 802.15.4.....	13
3.4 ZigBeen toimintataajuudet	14
3.5 CSMA-CA & DSSS	15
3.6 ZigBeen verkkotopologiat	16
3.7 XBee	17
4 TYÖN TOTEUTUS	19
4.1 Yhteyden muodostaminen Arduino XBee-moduuliin X-CTU-ohjelmalla .	19
4.1.1 Yhteyden testaaminen	19
4.1.2 Asetusten muuttaminen.....	21
4.2 Ohjelman lähetys Arduinoon ohjelmistokehitysympäristössä	22
4.3 Kahden Arduinon välinen kommunikaatio .Error! Bookmark not defined.	
4.4 Yhteyden kantavuuden testaus	26
5 POHDINTA.....	27
LÄHTEET	28
LIITTEET	
Liite 1. Arduinoon lähetettävä Ledin ON/OFF -sovellus	
Liite 2. Toiseen Arduinoon lähetettävä sovellus	

LYHENTEET JA TERMIT

802.15.4	IEEE:n standardi.
Arduino	Avoimen lähdekoodin elektroniikka-alusta.
XBee	Lisäosa Arduinolle, joka käyttää ZigBee-teknologiaa.
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance. MAC:n käyttämä tekniikka, joka määrittelee, milloin ZigBee-laite lähettää tietoa.
DSSS	Direct Sequence Spread Spectrum. Lähetettävän tiedon muokkaukseen oleva tekniikka.
FFD	Full Function Device. Täyden toiminnan laite.
FreeRTOS	Reaaliaikakäyttöjärjestelmä sulautetuille laitteille.
IEEE	Institute for Electrical and Electronics Engineers. Kansainvälinen tekniikan alan järjestö.
ICSP	In Circuit Serial Programming. Menetelmä suoraan ohjelmointiin.
MAC	Medium Access Control. Menetelmä tiedonsiirron kontrolloimiseen.
PHY	Physical Layer. OSI-mallin alin kerros.
RFD	Reduced Function Device. Rajoitetun toiminnan laite.
USB	Universal Serial Bus. Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen.
X-CTU	MaxStreamin kehittämä ohjelma.
ZigBee	IEEE 802.15.4 -standardin mukainen lyhyen kantaman tietoliikenneverkko.

1 JOHDANTO

Tässä työssä on tarkoituksena rakentaa toimiva kommunikaatio kahden Arduino Mega 2560:n välille ZigBee-yhteyttä käyttäen. Arduino Mega 2560 on mikrokontrollerialusta, joka yksinkertaisimmillaan voidaan liittää tietokoneeseen USB-kaapelilla. Arduinoon liitetään lisäksi XBee-moduulit, joita käyttämällä ZigBee-yhteys muodostuu. Työssä tutkitaan yhteyden kantavuutta eli sitä, miten pitkällä välimatkalla yhteys toimii.

Työssä kerrotaan yleisesti myös Arduinosta sekä ZigBeestä. Työn tarkoituksena on auttaa tulevia Arduinoon tutustujia saamaan aikaan suhteellisen helposti toimiva langaton yhteys. Työn on tilannut OAMK:n opettaja Teemu Korpela ja työtä valvoo Kari Jyrkkä. Työ toteutetaan OAMK:n OCTES-hankkeeseen, joka on osa Northern Periphery -kokonaisuutta. OCTES on kansainvälinen hanke, jossa on mukana organisaatioita ja oppilaitoksia myös Islannista, Irlannista ja Skotlannista.

2 ARDUINO

Arduino on avoimen lähdekoodin elektroniikka-alusta. Se perustuu joustavaan, helposti käytettävään piirikorttiin ja ohjelmistoon. Arduino on tarkoitettu suunnittelijoille, harrastajille, taiteilijoille ja kenelle tahansa, joka on kiinnostunut luomaan interaktiivisia laitteita tai ympäristöjä. (1.)

Arduino-projektit voivat olla itsenäisesti toimivia tai kommunikoida tietokoneella pyörivien ohjelmistojen kanssa (esim. Flash, Processing, MaxMSP). Arduino-alustat voidaan rakentaa itse käsin tai ostaa ne esiasennettuina, ohjelmistot ovat internetistä ladattavissa ilmaiseksi. (1.)

2.1 Historia

Arduino-projekti aloitettiin vuonna 2005 Italiassa ja Arduinoa on myyty 2/2010 mennessä jo yli 120 000 kappaletta. Arduinon kilpailijana pidetään Basic Stampia, joka on ollut olemassa jo 1990-luvun alusta. Arduino ratkaisee monia Basic Stampin ongelmia ja onkin ottanut valta-aseman. Basic Stampin ongelmia verrattuna Arduinoon ovat muun muassa sen kalliimpi hinta, hitaus, pienimuistisuus ja suljettu ympäristö. Arduino taas on open source -periaatteella toimiva. (2.)

Arduino on lyhyen ikänsä ajan ollut erittäin suosittu. Siihen on monia syitä, mutta ainakin avoin ohjelmistoympäristö, avoin laitteisto, kehitysalustan halpa hinta, Atmelin tekemät mikroprosessorit ja helppokäyttöisyys ovat Arduinon merkittäviä etuja. (1.)

2.2 Arduino Mega 2560

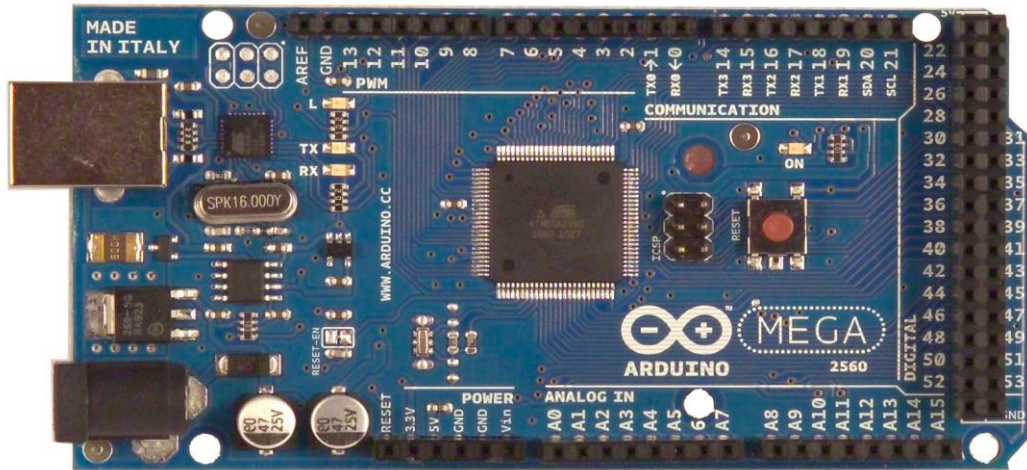
Arduino Mega 2560 (kuva 1) on mikrokontrollerialusta, joka perustuu ATmega1280:een. Mega sisältää 54 digitaalista input/output-pinniä (joista 14:ää voidaan käyttää PWM ulostulona), 16 analogista sisääntuloa, 4 UARTia

(laitteistosarjaporttia), 16 MHz:n kideoskillaattorin, USB-yhteyden, virtaliittimen, ICSP:n (In Circuit Serial Programming) ja reset-napin. Se sisältää kaiken, mitä tarvitsee mikrokontrollerin tukemiseen; yksinkertaisimmillaan se voidaan vain yhdistää tietokoneeseen USB-kaapelilla. Mega on lisäksi yhteensopiva suurimmaksi osaksi laajennuskortteja, jotka on suunniteltu Arduinon muille mikrokontrollereille kuten Duemilanovelle tai Diecimilalle. (3.)

Sen lisäksi että, Arduino Mega 2560 sisältää kaikki edellisen Arduino Mega -alustan ominaisuudet, se käyttää ATmega8U2:ta FTDI-sirun sijaan. Tämä mahdollistaa nopeammat siirtonopeudet ja siinä on kaksi kertaa enemmän flash-muistia kuin edellisellä Mega-alustalla. Ajureitakaan ei tarvita enää Linuxille tai Macille; tosin Windowsille tarvitaan vielä inf-tiedosto. Arduino Mega 2560 on 100-prosenttisesti yhteensopiva kaikille laajennusosille ja ohjelmistoille, jotka on suunniteltu Arduino Megalle. (3.)

Arduino Mega 2560:n tärkeimmät ominaisuudet ovat seuraavat:

- ATmega2560 mikrokontrolleri
- syöttöjännite: 7 – 12 V
- 54 digitaalista I/O-pinniä (14 PWM-lähtöä)
- 16 analogista sisääntuloa
- 256 kB Flash-muistia
- 16 MHz:n kellotaajuus.

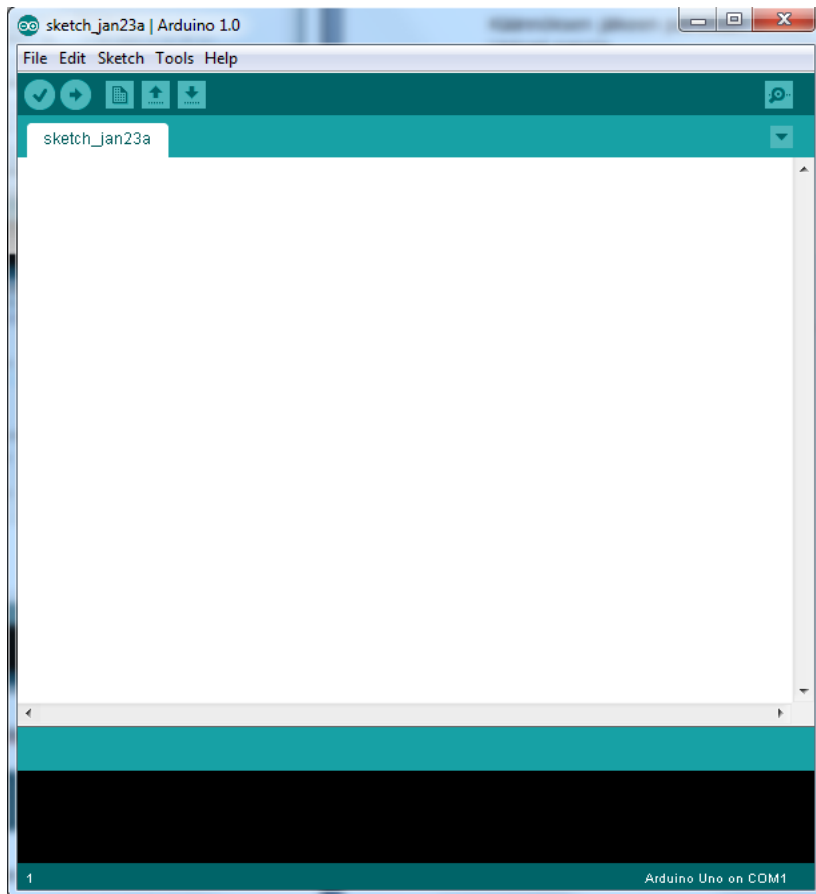


KUVA 1. Arduino Mega 2560 (3)

2.3 Ohjelmistokehitysympäristö

Arduinon kehitysympäristö (kuva 2) sisältää tekstieditorin koodin kirjoittamiseen, viestialueen, tekstikonsolin, työkalurivin (jossa on yleisimmille toiminnoille napit) sekä useita valikoita. Kehitysympäristö yhdistetään Arduino-laitteistoon, joka lataa siihen ohjelmia ja kommunikoi Arduinon kanssa. (4.)

Ohjelmia, jotka kirjoitetaan Arduinon tekstieditorissa, kutsutaan luonnoksiksi (sketches). Nämä luonnokset tallennetaan tiedostomuotoon .ino. Viestialue antaa palautetta luonnoksista ja näyttää samalla myös mahdolliset virheet. Konsoli näyttää tekstin outputin, joka sisältää täydelliset virheilmoitukset ja muun informaation. Työkalurivin painikkeiden avulla voi tarkistaa ja ladata ohjelmia, luoda, avata ja tallentaa luonnoksia sekä avata serial monitorin. (4.)



KUVA 2. Arduinon ohjelmistokehitysympäristö

2.4 FreeRTOS

FreeRTOS on reaaliaikakäyttöjärjestelmä sulautetuille laitteille ja perustuu avoimeen lähdekoodiin (open source). Se on yhteensopiva monien mikrokontrollereiden kanssa. Arduinon kanssa toimii DuinOS, joka perustuu FreeRTOS:n mikrokerneliin. Ensimmäinen versio julkaistiin 2009 ja uusin versio Alpha 0.2.0 2010 on yhteensopiva Arduino IDE:n kanssa. Ohjelmisto on kirjoitettu C-kielellä ja siten se on helposti portattava. (5.)

DuinOS-reaaliaikakäyttöjärjestelmän hyviä ominaisuuksia ovat mm. moniajo, tehtävien priorisointi, semaforit ja mutexit. Lisäksi DuinOS:illa on aktiivinen ja laaja käyttäjäyhteisö. (5.)

3 ZIGBEE

ZigBee on lyhyen kantaman langaton tekniikka, jonka tarkoituksena on pieni virrankulutus ja halvat valmistuskustannukset. Se perustuu MESH-verkkokerroksen standardiin, jossa määritetään radiorajapinta ja MAC ZigBeelle. ZigBee on erityisesti kehitetty langattomia ohjaus- ja automaatiojärjestelmiä varten. (6.)

3.1 Historia

ZigBee on suhteellisen uusi tekniikka, sillä sen kehitys aloitettiin 2000-luvulla ja ensimmäinen versio julkaistiin vuonna 2004. IEEE 802.15.4 -standardi, johon ZigBee perustuu, on suunniteltu samanlaisin lähtökohdin ja sen ensimmäinen versio tuli vuonna 2003. (6.)

ZigBeellä on monia eri versioita ja ne erotellaan niiden ratifiointivuoden mukaan. ZigBee 2004 on ensimmäinen versio ja sen korvasi ZigBee 2006, joka siis on hyväksytty vuonna 2006. ZigBee 2006:n korvasi seuraavana vuonna ilmestynyt ZigBee 2007. (6.)

3.2 ZigBee Alliance

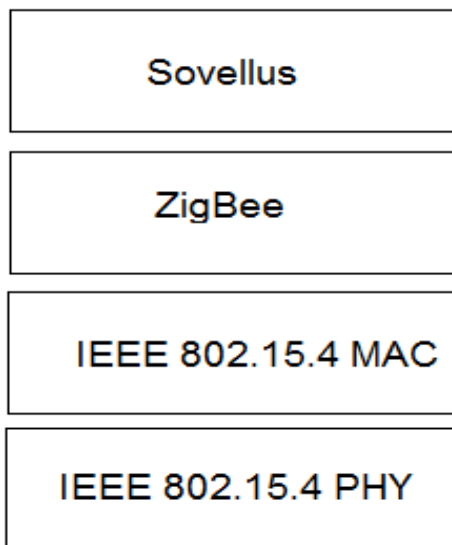
Vuonna 2002 perustettiin jäsenyhtiöistä koostuva kattojärjestö, jonka tarkoituksena oli luoda, ylläpitää ja kehittää ZigBee-teknologiaa. Kattojärjestön nimeksi tuli ZigBee Alliance. Allianssiin kuuluu monia kansainvälisiä suuryrityksiä kuten Philips, Hewlett Packard ja Intel. (7.)

Allianssiin liittymisessä on se etu, että se antaa oikeuden käyttää standardia kaupallisissa tarkoituksissa. Allianssiin kuului tammikuussa 2012 jo yli 350 jäsentä. Jäsenyystapoja on kolme, jotka ratkaisevat, kuinka paljon jäsen voi vaikuttaa asioihin. (7.)

3.3 IEEE 802.15.4

IEEE 802.15.4 -standardi julkaistiin vuonna 2003 ja se mahdollistaa lyhyen kantaman langattoman kommunikoinnin. Standardi on osa IEEE 802.15 -standardiperhettä, jonka standardit ovat määrittämässä Wireless Personal Area Network –alueella toimivan langattoman tiedonsiirron. IEEE 802.15.4 -standardin tavoitteena ominaisuuksien suhteen on ollut helppo käyttöönotto, luotettava tiedonsiirto, lyhyenkantaman toiminta, halpa hinta ja kelpo akunkesto. Standardi on kuitenkin siirtonopeuksiltaan vaatimaton. (8, s. 6.)

ZigBee käyttää IEEE 802.15.4 –standardin vuoden 2003 versiota fyysisen ja MAC-kerroksen määrittelyyn. Sovellusosio sisältää käyttäjän, ZigBee, ZigBee Alliancen ja MAC sekä PHY IEEE:n määrittelemät kerrokset. Alempana kuvassa (kuva 3) esitetään ZigBee-laitteen käyttämä protokollapino. (8, s. 6.)



KUVA 3. ZigBee-laitteen käyttämä protokollapino

IEEE 802.15.4 jakaa laitteet kahteen ryhmään niiden verkko-ominaisuuksien perusteella. Ensimmäiseen ryhmään kuuluvat FFD (Full Function Device) -laitteet, jotka sisältävät kaikki toiminnot. FFD voi toimia PAN-verkkokoordinaattorina, koordinaattorina tai päätelaitteena. (8, s. 8.)

Toinen ryhmä on RFD-laitteet (Reduced Function Device), jotka ovat pääsääntöisesti paristokäyttöisiä ja esimerkiksi myös siirreltäviä laitteita. RFD-laitteissa on rajoitettu toiminnallisuus, sillä niistä on poistettu reititysominaisuudet, joten ne voivat olla suurimman osan ajastaan virransäästötilassa. (8, s. 8.)

Verkko vaatii aina yhden FFD:n toimimaan PAN-verkkokoodinaattorina. Verkkokoordinaattorin tehtävänä on aloittaa verkon muodostus laitteiden välille, antaa laitteille paikalliset verkko-osoitteet ja verkolle tietty verkkotunnus. Suurimmallaan verkko voi sisältää jopa 65 536 laitetta. (8, s. 8.)

3.4 ZigBeen toimintataajuudet

ZigBeellä on olemassa kolme eri toimintataajuutta. Suurin osa kanavista on määritelty vapaassa käytössä olevalla 2,4 – 2,5 GHz:n ISM-taajuusalueella, jossa tietoa voidaan siirtää 250 kbit/s:n nopeudella. Kyseinen taajuusalue on maailmanlaajuisessa vapaassa käytössä. Euroopan ja Japanin alueilla taajuus 868 MHz on myös käytössä, mutta sen tiedonsiirtonopeudet jäävät noin 20 kbit/s:iin. Pohjois-Amerikassa on lisäksi käytössä 915 MHz:n taajuus, jossa siirtonopeus on maksimissaan 40 kbit/s. (8, s. 6.)

2,4 GHz:n taajuusalueella etuna on siis moninkertainen tiedonsiirtonopeus verrattuna alempiin taajuusalueisiin (taulukko 1), joka on tehnyt siitä käytetyimmän taajuuden. Sen etuna on myös vapaa taajuuskaista. Joissain tapauksissa voi olla hyödyllistä käyttää matalampia taajuuksia, koska niiden läpäisykyky on parempi. (8, s. 6.)

TAULUKKO 1. ZigBeen keskitaajuudet ja niiden ominaisuudet. (8, s. 7.)

Keskitaajuus	Kanavien lukumäärä	Siirtonopeus	Käyttöalue
868 MHz	1	20 kbit/s	Mm. Eurooppa
915 MHz	10	40 kbit/s	Mm. Pohjois-Amerikka
2,4 GHz	16	250 kbit/s	Maailmanlaajuinen

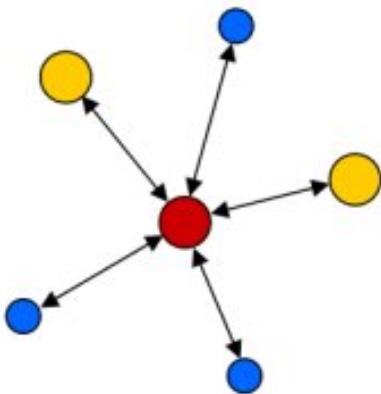
3.5 CSMA-CA & DSSS

IEEE 802.15.4 käyttää CSMA-CA (Carrier Sense Multiple Access With Collision Avoidance) -menetelmää kanavan varaamiseen. Tässä menetelmässä yhdellä kanavalla on useita lähetäviä laitteita (Multiple Access). Menetelmässä laitteet kuuntelevat kanavaansa ja lähettävät tietoa sen ollessa vapaa. Jos kanava on varattu, odotetaan satunnainen aika ja kuunnellaan kanavaa uudestaan. (8, s. 7.)

Tiedon välittämiseen käytetään DSSS (Direct Sequence Spread Spectrum) -tekniikkaa, jossa lähetettävään tietoon lisätään bittejä, jotta lähetettävä signaali saadaan levitettyä käytössä olevalle taajuusalueelle. Yleisimmin käytössä olevalla 2,4 GHz:n taajuudella käytetään modulointimenetelmänä ortogonaalista nelivaiheista vaiheavainnusta O-QPSK:ta (Orthogonal Quadrature Phase Shift Keying), jossa kantoaaltoa käsitellään siirtotiellä sopivaksi muuttamalla sen vaihetta. O-QPSK:ssa kantoaalto voi olla neljässä eri vaiheessa ja jokainen eri vaihe voidaan tulkita eri tilaksi. (8, s. 7.)

3.6 ZigBeen verkkotopologiat

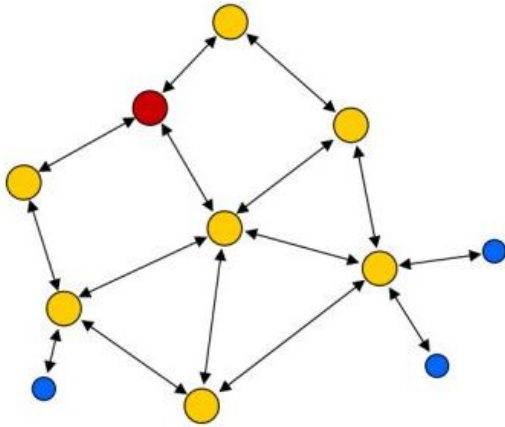
ZigBeelle on olemassa kaksi eri verkkotopologiaa: tähti- ja peer-to-peer-verkko. Tähtitopologiassa (kuva 4) verkon solmut keskustelevat PAN-koordinaattorin välityksellä. FFD-laite toimii PAN-koordinaattorina, joka muodostaa tällöin tähtitopologian mukaisen verkon. Koordinaattori samalla valitsee verkolleen oman PAN-tunnisteen, joka yksilöi verkon tämän kantavuusalueella. Tämän jälkeen toiset laitteet (FFD- tai RFD-laitteet), jotka toimivat samalla taajuuskanavalla, voivat halutessaan liittyä verkkoon. (10.)



KUVA 4. Tähtitopologian mukainen verkko. Punainen ympyrä PAN-koordinaattori, siniset ja keltaiset ympyrät FFD- ja RFD-laitteita. (10.)

Peer-to-peer-verkossa (kuva 5) laitteet pystyvät kommunikoimaan kaikkien kantaman sisällä olevien FFD- ja RFD-laitteiden kanssa. Peer-to-peer-verkot voivat muodostaa täten monimutkaisiakin verkkorakenteita. Verkko muodostuu, kun jokin FFD-laite ryhtyy PAN-koordinaattoriksi. PAN-koordinaattori asettaa itsensä ryppään (cluster) johtajaksi, asettaa oman ryppään tunnuksen nollaksi, valitsee vapaan PAN-tunnisteen ja alkaa ilmoittaa verkkoaan naapurilaitteille. Tämän jälkeen mahdolliset FFD- ja RFD-laitteet voivat tulla verkkoon. (10.)

Jos PAN-koordinaattoriin tuleva laite on FFD-laite, se voi ryhtyä tekemään omaa ryppästään, jonka johtajana laite toimii. Tällöin muut laitteet voivat liittyä tähän ryppääseen. Tällä menetelmällä verkon koko voi kasvaa laajaksi, mutta haittapuolena on viiveen kasvaminen tiedonsiirrossa. (10.)



KUVA 5. Peer-to-peer-topologian mukainen verkko. Punainen ympyrä toimii PAN-koordinaattorina, keltaiset ympyrät FFD-laitteita ja siniset ympyrät RFD-laitteita. (10.)

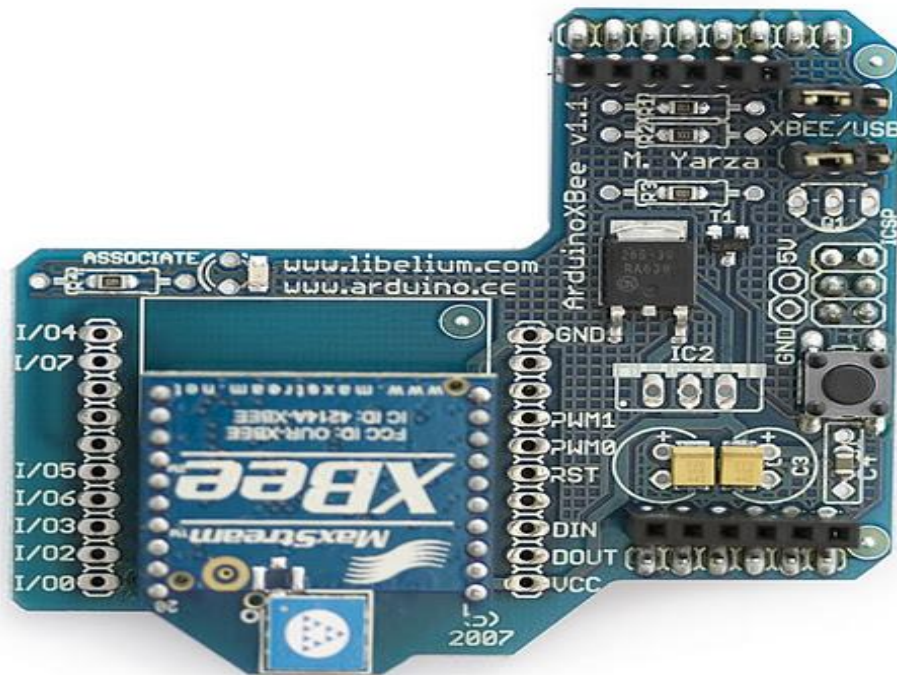
3.7 XBee

MaxStreamin kehittämä XBee-lisäosa (kuva 6) antaa Arduinolle mahdollisuuden kommunikoida langattomasti käyttäen ZigBee-protokollaa. XBeen RF-moduuli suunniteltiin täyttämään IEEE 802.15.4 -standardin vaatimukset ja tukemaan yksilöllisiä tarpeita edullisiin sekä vähävirtaisiin langattomiin anturiverkkoihin. Moduulit tarvitsevat vähän virtaa ja tarjoavat luotettavan tiedon toimituksen laitteiden välillä. Moduulit toimivat ISM 2,4 GHz:n taajuudella ja pin-for-pin-yhteensopivia toistensa kanssa. (11.)

XBee:n tärkeimmät ominaisuudet ovat seuraavat:

- pitkä kantama
 - sisätiloissa 30 metriä
 - ulkona parhaimmillaan 100 metriä
- lähetysteho 1 mW (0 dBm)
- vastaanottimen herkkyys -92 dBm
- vähävirtainen
 - TX-virta: 45 mA (@ 3,3 V)

- RX-virta: 50 mA (@ 3,3 V)



KUVA 6. Arduinon XBee-moduuli

4 TYÖN TOTEUTUS

Tässä työssä rakennettiin toimiva kommunikaatio kahden Arduino Mega 2560:n välille ZigBee-yhteyttä käyttäen. Kahteen Arduino Mega 2560:een liitetään XBee-moduulit, joiden kautta data liikkuu toisesta Arduinosta toiseen. Työssä käydään myös läpi miten XBee-moduulin yhteyttä testataan ja asetuksia muutetaan.

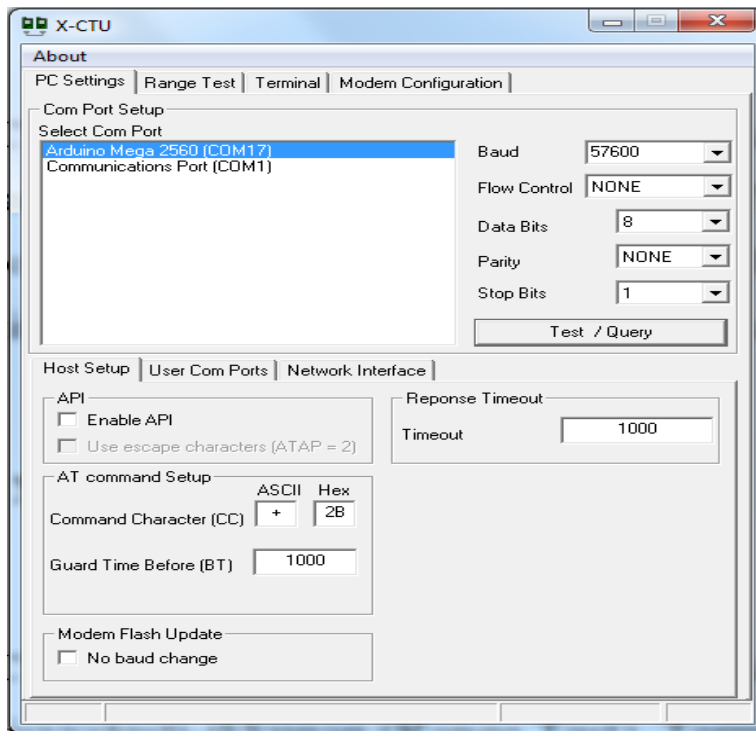
4.1 Yhteyden muodostaminen Arduino XBee-moduuliin X-CTU-ohjelmalla

4.1.1 Yhteyden testaaminen

XBee-moduulin toimivuutta Arduinossa voidaan testata käyttämällä X-CTU-ohjelmaa (kuva 7). X-CTU on MaxStreamin kehittämä vapaasti ladattava ohjelma, joka tukee esimerkiksi XBee-moduulia. Siinä on helppo ja yksinkertainen käyttöliittymä, joka on suunniteltu Maxstreamin radiomoduuleiden ohjausta varten.

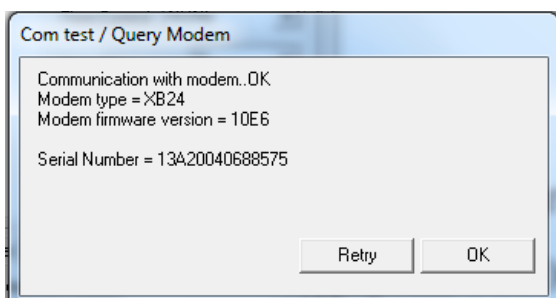
MaxStream on ohjelmisto, jolla voidaan muuttaa Arduino XBeen asetuksia ja päivittää esimerkiksi sen firmwarea. X-CTU:lla voidaan seurata myös XBeestä mahdollisesti lähtevää dataa, jolloin kahden Arduinon välisen yhteyden seuraaminen helpottuu. Testaus tapahtuu ilman Arduinon omia ohjelmistoa ja X-CTU pystyy suoraan ”keskustelemaan” XBee-moduulin kanssa ilman Arduino-pääalustaan puuttumista. X-CTU:n kautta tekemät asetusten muutokset menevät suoraan USB:n kautta XBee-piirille.

Tässä esimerkissä testattiin yhden Arduinon liitetyn XBeen yhteyden toimivuutta. Arduinon reset pitää tässä vaiheessa olla yhdistettynä GND:hen sekä XBeen on jumppereiden pitää tässä vaiheessa olla USB-asennossa. Arduino on yhdistettynä tietokoneeseen USB-kaapelilla.



KUVA 7. X-CTU-ohjelman perusnäkömää

Ohjelmalla voidaan testata XBee-moduulin toimivuus valitsemalla ensin oikea COM-portti sekä siirtonopeus (Baud). Jos yhteys toimii, näkyviin pitäisi tulla ruutu (kuva 8), jossa todetaan yhteyden toimivan ja kerrotaan moduulin tyyppi (XBee), firmware-versio ja sen sarjanumero.

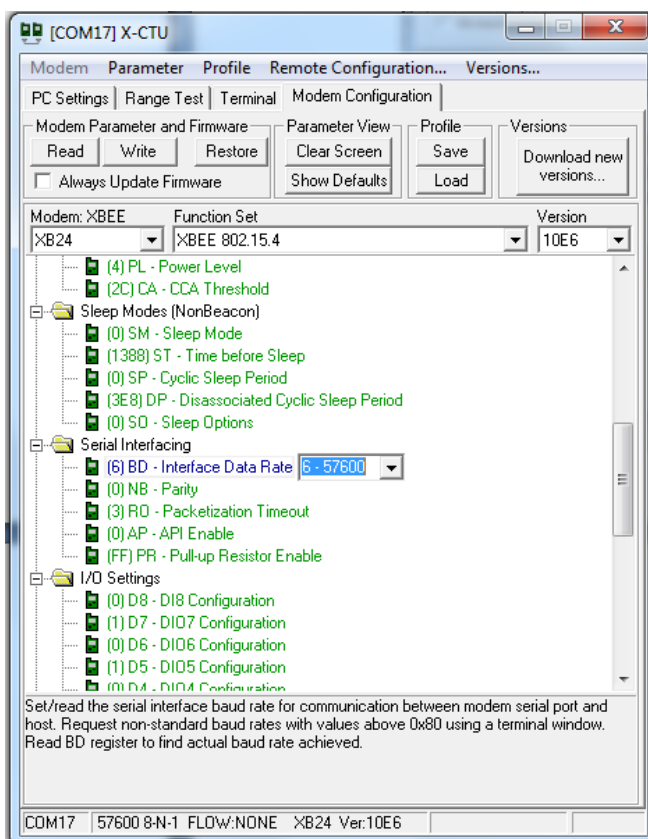


KUVA 8. Toimiva Com-testi XBeehen

4.1.2 Asetusten muuttaminen

Xbee-moduulin asetuksia pystytään muuttamaan onnistuneen yhteyden muodostamisen jälkeen samalla X-CTU-ohjelmalla. Modem Configuration -välilehdellä (kuva 9) luetaan ensin moduulin tiedot, jonka jälkeen asetuksia pystytään muuttamaan. Esimerkiksi siirtonopeuden muuttaminen on yksinkertaista.

Kun ensin painetaan Read-nappia ohjelman vasemmassa ylälaudassa, ohjelma lukee moduulista tiedot. Sen jälkeen Serial Interfacing –kohdasta valitaan BD - Interface Data Rate, josta voidaan valita siirtonopeus 1 200 ja 115 200 baudin väliltä.



KUVA 9. Baud raten valitseminen

4.2 Ohjelman lähetys Arduinoon ohjelmistokehitysympäristössä

Arduinon ohjelmistokehitysympäristössä on monia esimerkkisovelluksia, joita voidaan testata lähettämällä sovellus Arduinoon yksinkertaisesti. Tässä esimerkissä testataan Arduinon 4Display-Shieldiä. Ensin ohjelma käännetään Verify/Compile -napista (kuva 10). Kehitysympäristö näyttää mahdolliset virheet alaosassa sijaitsevassa tekstilaatikossa.



```
//example use of LCD4Bit_mod library

#include <LCD4Bit_mod.h>
//create object to control an LCD.
//number of lines in display=1
LCD4Bit_mod lcd = LCD4Bit_mod(2);

//Key message
char msgs[5][15] = {"Right Key OK ", //Nappien tekstit
                  "Up Key OK  ",
                  "Down Key OK ",
                  "Left Key OK  ",
                  "Select Key OK" };

int  adc_key_val[5] = {30, 150, 360, 535, 760 }; //nappien jännit
int  NUM_KEYS = 5;
int  adc_key_in;
int  key=-1;
int  oldkey=-1;

void setup() {
```

Done compiling.

Binary sketch size: 3042 bytes (of a 268048 byte maximum)

KUVA 10. Ohjelman käynnös Arduinoon ohjelmistokehitysympäristössä

Tämän jälkeen ohjelma lähetetään yksinkertaisesti Arduinoon painamalla Upload-nappia (kuva 11).

```
LCD4Bit | Arduino 0022
File Edit Sketch Tools Help
LCD4Bit LCD4Bit_mod.h
//example use of LCD4Bit_mod library

#include <LCD4Bit_mod.h>
//create object to control an LCD.
//number of lines in display=1
LCD4Bit_mod lcd = LCD4Bit_mod(2);

//Key message
char msgs[5][15] = {"Right Key OK ", //Nappien tekstit
                  "Up Key OK   ",
                  "Down Key OK  ",
                  "Left Key OK   ",
                  "Select Key OK" };

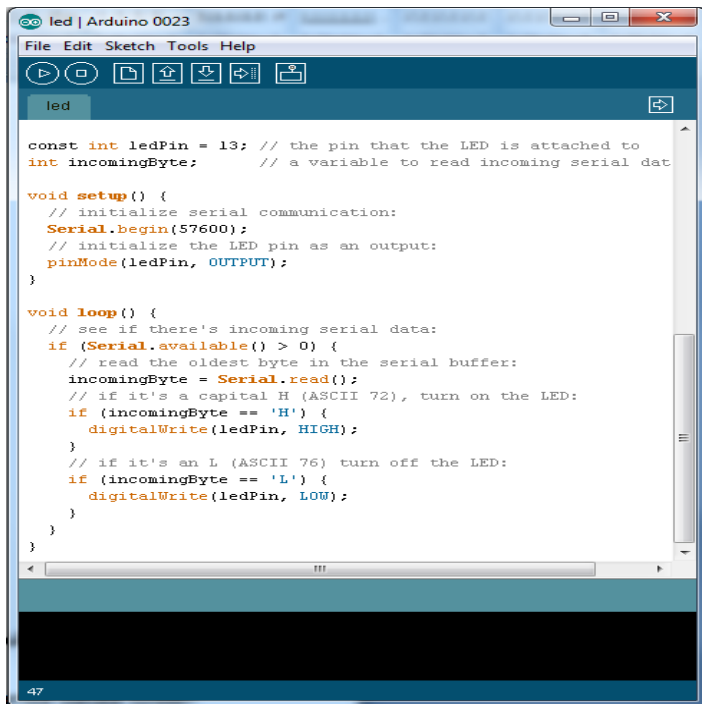
int  adc_key_val[5] = {30, 150, 360, 535, 760 }; //nappien jännit
int  NUM_KEYS = 5;
int  adc_key_in;
int  key=-1;
int  oldkey=-1;

void setup() {
  Done uploading.
  Binary sketch size: 3042 bytes (of a 258048 byte maximum)
  1
```

KUVA 11. Ohjelma lähetetään Arduino-alustalle

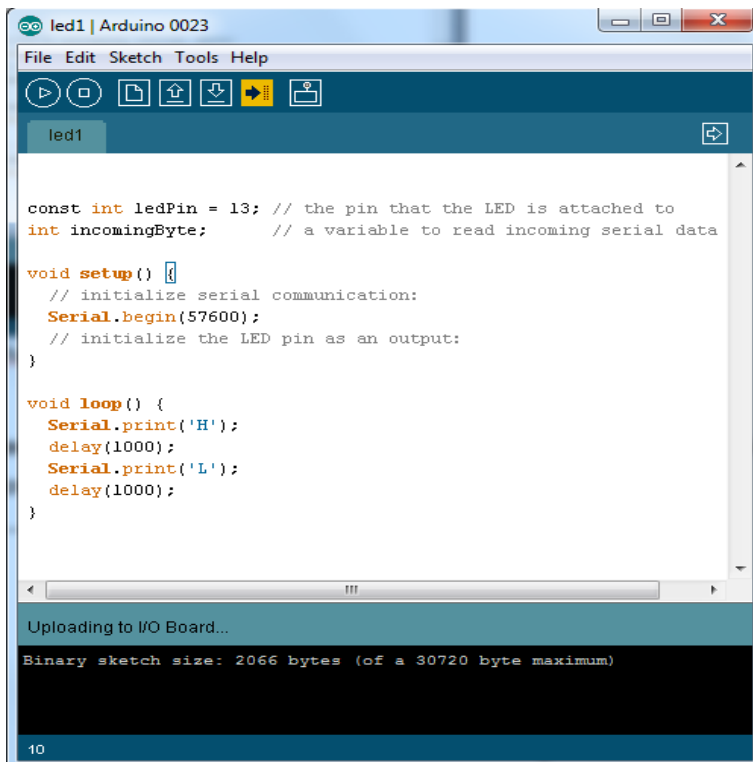
4.3 Kahden Arduinon välinen kommunikaatio

Arduinolle lähetetään esimerkkiohjelma (kuva 12), joka odottaa saavansa dataa toiselta Arduinolta. Jos Arduino vastaanottaa merkin 'H', Arduinoon syttyy ledi. Taasen jos Arduino vastaanottaa merkin 'L', Arduino sammuttaa ledin. Arduinossa ei ole nyt vielä tässä vaiheessa kytkettynä XBee-moduulia vaan Arduino on yhdistettynä pelkästään USB-kaapelilla tietokoneeseen.



KUVA 12. Arduinoon lähetetään esimerkksiovellus

Tämän jälkeen toiseen Arduinoon lähetetään sovellus (kuva 13), joka lähettää merkkiä 'H' ja 'L' vuorotellen sekunnin välein. Toinen Arduino on liitetty toiseen tietokoneeseen USB-kaapelilla.



KUVA 13. Toiseen Arduinoon lähetetään sovellus

4.4 Yhteyden kantavuuden testaus

Xbee-lisäosan valmistaja MaxStream lupaa ZigBee-yhteydelle sisätiloissa n. 30 metrin ja ulkona n. 100 metrin kantavuuden. Nämä ovat ns. ihanteellisissa ympäristöissä saavutettavia lukemia, joihin ei normaalisti pääse. Sääolosuhteet ja muut häiriötekijät vaikuttavat kantavuuteen.

Testasin yhteyden kantavuutta Oulun seudun ammattikorkeakoulun tiloissa sekä Oulun keskustan alueella. Lähettävä Arduino oli liitettynä tietokoneeseen ja toinen Arduino kannettavaan tietokoneeseen. Sisätiloissa pääsin parhaimmillaan lähes 20 metrin kantavuuteen, jolloin ledi vielä vilkkui vastaanottavassa Arduinossa. Heti kun yhteys ei enää toiminut, ledi lakkasi vilkkumasta. Seinien materiaalit vaikuttavat varmasti yhteyden toimivuuteen, jolloin kantavuus pienenee.

Ulkona Arduinon kantavuus testissä oli n. 40 metriä eli ei puoliakaan MaxStreamin lupaamasta kantavuudesta. Yhteys toimi kuitenkin suhteellisen kauas eli noin kaksi kertaa pitemmälle verrattuna sisätilaan.

Yhteyden kantavuutta voisi yrittää parantaa esimerkiksi vaihtamalla XBeen standardiantennin tehokkaampaan. Markkinoilla on monia eri vaihtoehtoja vaihtoantenneiksi, joita mainostetaan tehokkaammaksi kuin XBeen alkuperäinen antenni.

5 POHDINTA

Opinnäytetyön tarkoituksena on olla ohje tuleville Arduinoon tutustujille sekä rakentaa toimiva ZigBee-yhteys kahden Arduino Mega 2560:n välille. Työ onnistui tältä osin hyvin. Kahteen Arduinoon liitettiin XBee-lisäosat ja niiden välille saatiin toimiva ZigBee-yhteys, joka oli tavoitteenakin. Toinen Arduino lähetti komentoja toiseen Arduinoon, joka tietyn komennon saatuaan laittoi ledin päälle ja sammutti sen.

Tämän työn avulla henkilö, joka haluaa tutustua Arduinoon, pääsee helpommin asiaan käsiksi kuin tutkimalla esimerkiksi Arduinoa internetistä. Arduino on hyvinsoveltuva opetustarkoitukseen, joten työstä löytyy varmasti hyödyllistä tietoa esimerkiksi koulumme opiskelijoille, jotka voivat kehittää omia sovelluksiaan Arduinoon. Täten työtä voidaan käyttää hyväksi opetustarkoituksessa tai missä tahansa Arduinoon liittyvässä projektissa. Työssä käsitellään ZigBee-tekniikkaa myös yleisellä tasolla.

Loppujen lopuksi aihe oli hyvin opettava ja mielenkiintoinen insinööriyöksi, vaikka matkan varrella oli jonkun verran hankaluuksia, jotka hidastivat työn valmistumista. Kuitenkin työ antoi minulle suhteellisen paljon informaatiota Arduinosta ja siihen liittyvistä asioista.

LÄHTEET

1. What is Arduino. 2012. Arduino. Saatavissa:
<http://www.arduino.cc/en/Guide/Introduction>. Hakupäivä 26.1.2012.
2. Microcontroller Information, Basic Stamp. 2012. Parallax Inc. Saatavissa:
<http://www.parallax.com/ProductInfo/Microcontrollers/tabid/121/Default.aspx>.
Hakupäivä 3.5.2012.
3. Arduino Mega 2560. Arduino. Saatavissa:
<http://arduino.cc/en/Main/ArduinoBoardMega2560>. Hakupäivä 29.1.2012.
4. Arduino Development Environment. Arduino. Saatavissa:
<http://arduino.cc/en/Guide/Environment>. Hakupäivä 23.3.2012.
5. Features Overview.2012. FreeRTOS. Saatavissa:
http://www.freertos.org/FreeRTOS_Features.html. Hakupäivä 27.1.2012.
6. ZigBee 2007 Specification. 2007. Zigbee Alliance. Saatavissa:
<http://www.zigbee.org/Standards/Downloads.aspx>. Hakupäivä 15.2.2012.
7. About ZigBee Alliance. 2012. Zigbee Alliance. Saatavissa:
<http://zigbee.org/About/AboutAlliance/TheAlliance.aspx>. Hakupäivä 30.1.2012.
8. Hassinen, Jani 2005. ZigBee ja IEEE 802.15.4. Lappeenranta:
Lappeenrannan teknillinen yliopisto. Lähiverkot-erikoistyyökurssi. Saatavissa:
http://www2.it.lut.fi/kurssit/04-05/010626000/seminaarit/zigbee_ja_ieee802154_jari_hassinen_seminaari.pdf.
Hakupäivä 20.2.2012.
9. Lehonkoski, Lasse 2010. ZigBee-verkot. Lappeenranta: LUT, sähkötekniikan
koulutusohjelma. Kandidaatintyö. Saatavissa:

<http://www.doria.fi/bitstream/handle/10024/66229/Zigbee-verkot.pdf?sequence=1>. Hakupäivä 1.3.2012.

10. IEEE 802.15.4. 2007. Savonia ammattikorkeakoulu. Saatavissa: http://wirelessplatform.savonia.fi/index.php?option=com_content&task=view&id=30&Itemid=53. Hakupäivä 15.2.2012.

11. XBee / Xbee-PRO OEM RF Modules Datasheet. 2006. MaxStream. Saatavissa: <http://www.libelium.com/squidbee/upload/3/31/Data-sheet-max-stream.pdf>. Hakupäivä 20.2.2012.

/*

Physical Pixel

An example of using the Arduino board to receive data from the computer. In this case, the Arduino boards turns on an LED when it receives the character 'H', and turns off the LED when it receives the character 'L'.

The data can be sent from the Arduino serial monitor, or another program like Processing (see code below), Flash (via a serial-net proxy), PD, or Max/MSP.

The circuit:

* LED connected from digital pin 13 to ground

created 2006

by David A. Mellis

modified 30 Aug 2011

by Tom Igoe and Scott Fitzgerald

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/PhysicalPixel>

*/

```
const int ledPin = 13; // the pin that the LED is attached to
int incomingByte; // a variable to read incoming serial data into
```

```
void setup() {
  // initialize serial communication:
  Serial.begin(57600);
```

```
// initialize the LED pin as an output:
```

```
  pinMode(ledPin, OUTPUT);
}
void loop() {
  // see if there's incoming serial data:
  if (Serial.available() > 0) {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();
    // if it's a capital H (ASCII 72), turn on the LED:
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    // if it's an L (ASCII 76) turn off the LED:
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

```
const int ledPin = 13; // the pin that the LED is attached to
int incomingByte;    // a variable to read incoming serial data into

void setup() {
  // initialize serial communication:
  Serial.begin(57600);
  // initialize the LED pin as an output:
}

void loop() {
  Serial.print('H');
  delay(1000);
  Serial.print('L');
  delay(1000);
}
```