

---

# Arduino-lämpömittari ZigBee-yhteydellä



Ammattikorkeakoulun opinnäytetyö

Tietotekniikan koulutusohjelma

Riihimäki, 22.9.2013

Jani-Pekka Laukkanen



Riihimäki  
Tietotekniikan koulutusohjelma

---

<b>Tekijä</b>	Jani-Pekka Laukkanen	<b>Vuosi</b> 2013
<b>Työn nimi</b>	Arduino-lämpömittari ZigBee-yhteydellä	

---

## TIIVISTELMÄ

Opinnäytetyön tarkoitus oli toteuttaa lämpötila-anturilla varustettu Arduino, joka lähettää anturin tiedot langattomalla ZigBee-yhteydellä toiselle Arduinolle. Toinen Arduino vastaanottaa datan ja tulostaa sen näytölle. Lämpötila-anturi ja näyttö voivat siis olla fyysisesti eri paikoissa ZigBee-yhteyden kantaman sisällä. Valmis tuotos jäi omaan käyttöön ja sitä käytetään varastotilan lämpötilan seurantaan.

Teoriaosuudessa käsiteltiin Arduino-kehitysalustan ja ZigBee-tekniikan perusteita. Niihin perehdyttiin vastaavasti aiheita käsittelevien Internet-sivustojen ja kirjallisuuden avulla. Työn toteutuksessa apuna käytettiin Arduino Projects Book -kirjaa, jonka esimerkit selventävät Arduinon kytkentöjä ja ohjelmointia.

Arduino-lämpömittarilla suoritettiin testit, joissa mitattiin lämpötila-anturin tarkkuus verrattuna toiseen digitaaliseen lämpömittariin ja langattoman yhteyden signaalin kantamaa. Tulosten perusteella lämpötila-anturi on tarpeeksi tarkka, mutta langaton yhteys Arduinojen välillä toimi vain yhdessä varastotilassa, joka sijaitsee 12 metrin etäisyydellä mittauspisteestä. Testauksessa havaittiin, että signaalin kantama jäi alle XBee-moduulin spesifikaation signaalin kantamasta sekä sisällä että ulkona.

Työtä on mahdollista kehittää eteenpäin parantamalla mittaustarkkuutta ja luomalla anturiverkko, jolloin lämpötilaa voidaan mitata monesta eri paikasta samaan aikaan. Kuitenkin Arduino-lämpömittari on sellaisenaan toimiva ja haluttu lopputulos saavutettiin.

**Avainsanat** Arduino, lämpömittari, ZigBee, XBee

**Sivut** 24 s. + liitteet 3 s.

Riihimäki  
Degree Programme in Information Technology

---

<b>Author</b>	Jani-Pekka Laukkanen	<b>Year</b> 2013
<b>Subject of Bachelor's thesis</b>	Arduino thermometer with a ZigBee connection	

---

ABSTRACT

The purpose of this Bachelor's thesis was to create an Arduino equipped with a temperature sensor. This Arduino sends the sensor's information to a second Arduino using a wireless ZigBee connection. The second Arduino receives the data and prints it on a display screen. The temperature sensor and the display screen can physically be in different locations within the ZigBee connection's signal range. The final product was left in personal use and it is used to measure temperature in a storage unit.

The theoretical part of this thesis covered the basics of Arduino development platform and ZigBee technology. They were familiarized by studying related Internet sites and literature accordingly. The implementation of the product was finished with the help of the Arduino Projects Book, which clarifies the wiring and programming of Arduino.

A series of tests were conducted with the Arduino thermometer to measure the accuracy of the temperature sensor compared to another digital thermometer and the range of the wireless connection. Based on the results the temperature sensor is accurate enough, but the wireless connection between the two Arduinos only worked in one storage unit located 12 meters from the measuring point. During the tests, it was noticed that the signal range stayed below the specification of the XBee module both inside and outside.

It is possible to further develop the product by improving the measurement accuracy and creating a sensor network so that the temperature can be measured from many different locations at the same time. However the Arduino thermometer is functional as it is and the desired result was achieved.

**Keywords** Arduino, thermometer, ZigBee, XBee

**Pages** 24 p. + appendices 3 p.

# SISÄLLYS

1	JOHDANTO.....	1
2	ARDUINO.....	1
2.1	Arduino-kehitysalusta .....	1
2.2	Arduino Uno R3 .....	2
2.3	Arduino IDE.....	4
2.4	Arduinon käyttökohteet.....	6
3	ZIGBEE.....	6
3.1	IEEE 802.15.4-standardi .....	6
3.2	ZigBee Alliance.....	7
3.3	ZigBee-verkon tekniikka.....	7
3.4	ZigBee-verkon tietoturva .....	7
3.5	ZigBee-verkon osat .....	8
3.6	ZigBee-verkot ja topologiat .....	9
3.7	ZigBeen käyttökohteet .....	9
3.8	XBee.....	10
4	TYÖN TOTEUTUS .....	11
4.1	Tarvikkeet.....	11
4.2	Lämpötila-anturin kytkeminen.....	14
4.3	LCD-näytön kytkeminen.....	14
4.4	Langattoman yhteyden luominen .....	16
4.5	Ohjelmointi.....	17
4.6	Testaus.....	20
5	YHTEENVETO .....	21
	LÄHTEET .....	23
Liite 1	Lähtävän pään Arduinon ohjelmakoodi	
Liite 2/1	Vastaanottavan pään Arduinon ohjelmakoodi	
Liite 2/2	Vastaanottavan pään Arduinon ohjelmakoodi	

---

## LYHENTEET JA TERMIT

Arduino	Arduino on avoimeen laitteistoon ja lähdekoodiin perustuva ohjelmoitava elektroniikka-alusta.
Baudi	Baudi on tiedonsiirtonopeuden suure, joka kuvaa elektronisen signaalin muutosnopeutta sekunnissa.
CSMA/CA	Carrier Sense Multiple Access With Collision Avoidance on tietoliikenteen siirtotien varausmenetelmä, jolla useat lähetävät laitteet jakavat samaa siirtotietä.
DSSS	Direct Sequence Spread Spectrum eli suorasekvennoinnissa lähetettävä sanoma jaetaan pieniin osiin ja lähetetään koko taajuusalueella yhtenä signaalina.
IEEE	Institute of Electrical and Electronics Engineers on kansainvälinen tekniikan alan järjestö.
Mikrokontrolleri	Mikrokontrolleri, jossa on mikroprosessori ja joitain muisti- ja liityntälohkoja, on pieni tietokone, joka ei välttämättä tarvitse muuta kuin virtalähteen toimiakseen.
O-QPSK	Offset Quadrature Phase Shift Keying eli ortogonaalisessa nelivaiheisessa vaiheavainnuksessa moduloiva signaali muuttaa kantoaallon vaihetta suoraan, jolloin hetkellinen vaihe ilmaisee kahta bittiä.
Pseudokohina	Pseudokohina on keinotekoisesti luotua kohinaa, jolla on tilastollisesti satunnaisia ominaisuuksia.
PWM	Pulse-Width Modulation eli pulssinleveysmodulaatio on moduloitintapa, jossa kuormaan menevää jännitettä säädetään muuttamalla pulssisuhdetta.
XBee	Radioantenni, joka perustuu IEEE 802.15.4-standardiin.
ZigBee	ZigBee on IEEE 802.15.4-standardin mukainen pienitehoinen, lyhyen kantaman tietoliikenneverkko, jonka tarkoituksena on pienten ja yksinkertaisten laitteiden liittäminen verkkoon langattomasti.

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on toteuttaa lämpötila-anturilla varustettu Arduino, joka lähettää anturin tiedot langattomalla ZigBee-yhteydellä toiselle Arduinolle. Toinen Arduino vastaanottaa datan ja tulostaa sen näytölle. Toisin sanottuna lämpötila-anturi ja näyttö ovat fyysisesti eri paikoissa langattoman yhteyden kantaman sisällä. Valmiin tuotoksen toiminta testataan lämpötila-anturin tarkkuuden ja signaalin kantaman osalta. Testauksessa käytetään vertailukohteena toista digitaalista lämpömittaria. Testaus suoritetaan sekä sisä- että ulkotiloissa.

Työssä myös tutustutaan Arduino-kehitysalustaan ja ZigBee-verkkoihin. Toteutuksessa käytetään Arduino Uno R3 -alustaa, joten työssä käsitellään sen ominaisuuksia. ZigBee-protokollaa käsitellään yleisesti sen toiminnan kannalta.

Työ toteutetaan itsenäisesti henkilökohtaisena projektityönä ja valmis tuotos jää omaan käyttöön varastotilan lämpötilan seurantaan. Tavoitteena on oppia elektroniikasta, langattomista verkoista ja ohjelmoinnista, koska kaikkia näitä osa-alueita tarvitaan työn toteuttamisessa.

## 2 ARDUINO

### 2.1 Arduino-kehitysalusta

Arduino on avoimeen laitteistoon ja lähdekoodiin perustuva ohjelmitava elektroniikka-alusta. Arduino-piirilevyn ja ohjelmiston helppokäyttöisyys ja joustavuus mahdollistavat sen, että suunnittelijat sekä harrastelijat voivat luoda vuorovaikutteisia laitteita. (Arduino.)

Arduino-työryhmä perustettiin vuonna 2005 Italiassa. Työryhmään kuului Hernando Barragan, Massimo Banzi, David Cuartielles, Dave Mellis, Gianluca Marino ja Nicholas Zambetti. Heidän tavoitteenaan oli yksinkertaistaa Wiring elektroniikka-alustaa, jotta saataisiin harrastelijoiden käyttöön sopiva alusta. Tuloksena syntyi Arduino, jonka ominaisuuksiin kuuluu ohjelmointiympäristö, joka perustuu Processing-ohjelmointikieleen, USB-yhteys käyttöä ja ohjelmointia varten, sekä alhainen hinta. (Arduino: A Brief History.)

Arduino on työkalu, jolla tehdään pieniä tietokoneita, joita laajennetaan kytkimillä, sensoreilla, servoilla tai muilla vastaavilla fyysisillä laitteilla. Näiltä laitteilta Arduino ottaa vastaan tietoa tai ohjaa niitä mikrokontrollerin avulla. (Arduino - Introduction.)

Arduino-alustat voivat toimia itsenäisesti sulautettuina järjestelminä tai kommunikoida toisen tietokoneen ohjelmiston kanssa. Arduinon toiminta perustuu ATmega8- ja ATmega168-mikrokontrollereihin, jotka on julkaistu Creative Commons -lisenssin alla. Tämä lisenssi mahdollistaa sen, että piirisuunnittelijat saavat muokata ja laajentaa Arduinon mikrokontrollere-

ja. ATmega-mikrokontrollerit kuuluvat Atmelin AVR tuoteperheeseen, joka sisältää laajan valikoiman 8-bittisiä mikrokontrollereja. Kuvassa 1 on esitetty ATmega8-mikrokontrolleri. (Arduino - Introduction; MegaAVR Microcontroller.)



Kuva 1. ATmega8-mikrokontrolleri (ATmega8)

Piirilevyt, joita Arduino-alustat käyttävät, voidaan rakentaa käsin valmiiden kytkentäkaavioiden mukaan tai ostaa esiasennettuina. Arduino-projektin avoimuus on johtanut siihen, että markkinoille on tullut samankaltaisia alustoja, jotka perustuvat Arduinoon. Kuitenkin Arduinon kehittäjät ovat painottaneet, että Arduino-nimeä saa käyttää vain virallisista tuotteista. Monet valmistajat ovat kiertäneet säännön käyttämällä Duino-nimeä omissa tuotteissaan. (Arduino - Introduction; Arduino - Policy.)

### 2.2 Arduino Uno R3

Arduino Uno R3 on kolmas kehitysversio Arduino Uno -piirilevystä. Se on uusin USB-väylää käyttävä Arduino, ja samalla se on myös referenssimalli Arduino-alustan muille tuotteille. Fyysiseltä kooltaan piirilevy on 6,9 cm pitkä ja 5,3 cm leveä. Kuvassa 2 on esitetty Arduino Uno R3:n piirilevy. (Arduino Uno.)



Kuva 2. Arduino Uno R3

Arduino Uno R3 eroaa aiemmista Uno-piirilevyistä siten, että FTDI USB-sarjaliitännän ajuripiirin sijasta se käyttää ATmega16U2-mikrokontrolleria, joka on ohjelmoitu USB-sarjaliitännämuunnimeksi. Lisäksi R3:een on lisätty uusia pinnejä ja muutettu nollauspiirin toimintaa. (Arduino Uno.)

Arduino Uno R3:n tärkeimmät ominaisuudet ovat seuraavat:

- ATmega328-mikrokontrolleri
- 5V käyttöjännite
- suositeltu syöttöjännite: 7-12V
- 14 digitaalista I/O-pinniä, joista 6 on PWM-ohjattuja lähtöjä
- 6 analogista tuloa
- yhden I/O-pinnin tasavirta: 40mA
- 3,3V-pinnin tasavirta: 50mA
- 32 kB Flash-muistia
- 2 kB SRAM-muistia
- 1 kB ohjelmoitavaa EEPROM-muistia
- 16 MHz kellotaajuus (Arduino Uno.)

Arduino Uno R3:n piirilevy saa virtansa USB-väylästä, ulkoisesta virtalähteestä tai paristoista. Käytössä olevat virta-pinnit ovat VIN, 5V, 3,3V, GND ja IOREF. VIN-pinnin kautta voidaan syöttää jännitettä piirilevylle, jos käytössä on erillinen virtalähde. GND on maadoituspinni. 5V- ja 3,3V-pinnit antavat nimensä mukaisesti vastaavat jännitteet ulostulona. IOREF-



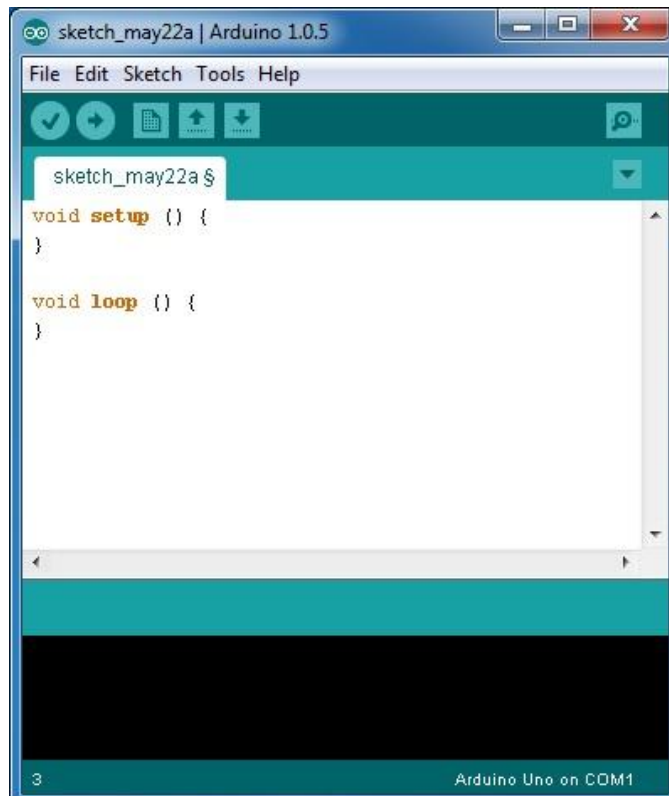
pinniä käytetään joidenkin shield-lisälaitteiden kanssa, jolloin shield voi lukea IOREF-pinnistä piirilevyn käyttämät jännitteet ja lähettää ne eteenpäin. (Arduino Uno.)

Jokaista 5V:n digitaalista I/O-pinniä voidaan käyttää joko tulona tai lähtönä. Niitä voidaan myös ohjata mikrokontrollerin avulla käyttämällä pinMode, digitalWrite ja digitalWrite -ohjelmointikomentoja. Lisäksi osalla pinneistä on erikoistoimintoja. Pinnejä 0 ja 1 käytetään sarjamuotoisen datan vastaanottamiseen ja lähettämiseen. Nämä pinnit on kytketty ATmega16U2 USB-sarjaliitäntämuuntimen vastaaviin pinneihin. Pinnejä 2 ja 3 voidaan käyttää keskeytyssignaalin luomiseen. Pinnit 3, 5, 6, 9, 10 ja 11 toimivat 8-bittisinä PWM-lähtöinä, joita ohjataan analogWrite-ohjelmointikomennolla. Pinneissä 10, 11, 12 ja 13 voidaan käyttää SPI-yhteyttä. Pinnissä 13 on myös sisäänrakennettu LED-valo. (Arduino Uno.)

Arduino Uno R3:n analogiset pinnit toimivat 10-bittisellä tarkkuudella 0-5V välillä. Analogiset pinnit A4 ja A5 tukevat TWI-yhteyttä. Lisäksi piirilevyllä on vielä AREF ja Reset -pinnit. AREF-pinnillä voidaan muuttaa kaikkien analogisten pinnien jännitteen ylärajaa. Reset-pinni nimensä mukaisesti asettaa piirilevyn aloitustilaan, mutta itse pinniä käytetään vain silloin, kun Reset-nappia ei ole mahdollista painaa. (Arduino Uno.)

### 2.3 Arduino IDE

Arduino IDE on ohjelmointiympäristö, jolla kirjoitetaan koodi, jota Arduinon mikrokontrolleri suorittaa. Ohjelma on ladattavissa ilmaiseksi ja se toimii Windows, Macintosh OSX ja Linux käyttöjärjestelmissä. Ohjelman uusin versio kirjoitushetkellä on 1.0.5. Kuvassa 3 on esitetty Arduino IDE:n pääikkuna. (Arduino - Introduction.)



Kuva 3. Arduino IDE:n pääikkuna

Arduino IDE perustuu Processing-ohjelmointikieleen, ja sen mukana on Wiring-ohjelmakirjasto, jonka avulla yleiset siirännän operaatiot helpotuvat. Arduinon ymmärtämä koodi kirjoitetaan C++-kielellä. Koko ohjelmakoodia kutsutaan nimellä Sketch. (Wikipedia - Arduino.)

Ohjelmointiympäristön koodieditorin ominaisuuksiin kuuluu automaattinen sisennys, koodin väritys ja sulkeiden sovitus. Lisäksi koodin lähettäminen Arduinon piirilevylle onnistuu yhdellä napin painalluksella. Ohjelmointiympäristö tekee automaattisesti tarvittavat muutokset koodiin lisäämällä alkuun ylätunnisteen ja loppuun main-funktion. Sitten ohjelmakoodi käännetään ja lopuksi lähetetään USB:n välityksellä piirilevylle. (Wikipedia - Arduino.)

Yksinkertaisuudestaan johtuen Arduino IDE:llä on puutteita etenkin virheiden havaitsemisessa. Ohjelmointiympäristö ei kerro käyttäjälle puutteellisista komennoista tai virhetiloista. Koodin simulointia ei myöskään ole saatavilla, jolloin käyttäjä joutuu itse testaamaan koodia yrityksen ja erehdyksen kautta.

Arduinon ohjelmakoodi koostuu kahdesta pääfunktioista. Setup ajetaan, kun ohjelma käynnistyy. Siinä alustetaan tarvittavat muuttujat, pinnit ja ohjelmakirjastot. Loop ajetaan setupin jälkeen ja sitä toistetaan, kunnes Arduinon virta katkaistaan. Loopin tehtävänä on ohjata piirilevyä ja siihen kytkettyjä laitteita. (Arduino - Language Reference.)

## 2.4 Arduinon käyttökohteet

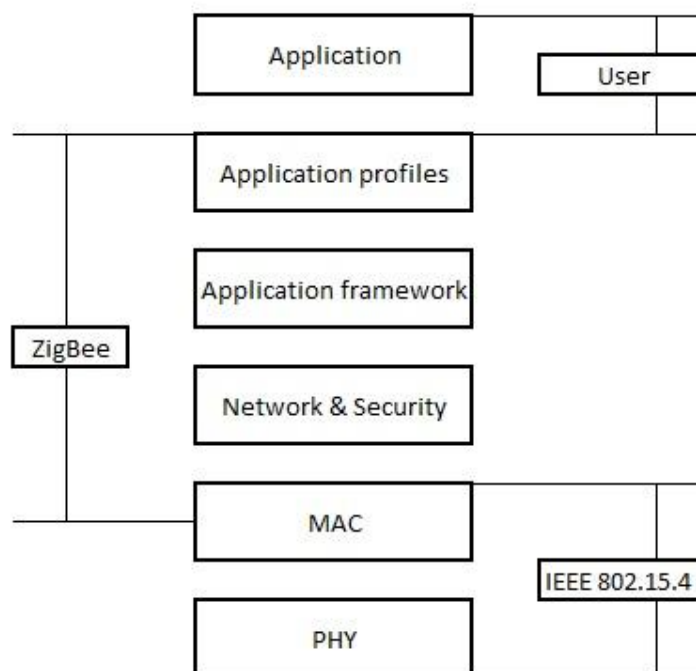
Arduinon joustavuuden ja laajennettavuuden ansiosta sillä voidaan toteuttaa monia eri tehtäviä. Ehkä suurimpana käyttökohteena on robotiikka, koska Arduino voidaan ohjelmoida ohjaamaan siihen kiinnitettyjä laitteita. Arduinon kotisivuilla on Playground-osio, jossa käyttäjät voivat jakaa omia tuotoksiaan. Sieltä löytyy muun muassa ohjeita autonomisten Arduino-robottien rakentamiseen. (Arduino - Playground.)

## 3 ZIGBEE

### 3.1 IEEE 802.15.4-standardi

ZigBee-protokolla perustuu IEEE 802.15.4-standardiin, jonka tarkoituksena on luoda pohja pienitehoiselle ja lyhyen kantaman langattomalle WPAN-verkolle, jota käytetään yksinkertaisissa laitteissa. Standardin tavoitteena on mahdollistaa joustavuus, alhaiset kustannukset, luotettava tiedonsiirto ja pieni virrankulutus. Toisaalta tämä johtaa siihen, että tiedonsiirtonopeudet ja signaalin kantama jäävät vaatimattomiksi. Standardi toimii kansainvälisellä lisensoimattomalla taajuuskaistalla. (IEEE 802.15.4.)

ZigBee-protokolla on rakennettu 802.15.4-standardin fyysisen ja MAC-kerroksen päälle. ZigBee sisältää osan MAC-kerroksesta, verkko- ja tietoturvakerroksen, ohjelmistokehityksen kerroksen sekä ohjelmistoprofiilin kerroksen. Lopullinen sovelluskerros kuuluu käyttäjälle. Kuvassa 4 on esitetty ZigBee-laitteen käyttämä protokollapino. (ZigBee: IEEE 802.15.4 2007, 10.)



Kuva 4. ZigBee-laitteen käyttämä protokollapino

### 3.2 ZigBee Alliance

ZigBee Alliance on vuonna 2002 perustettu järjestö, jonka tehtävänä on ylläpitää ja kehittää ZigBee-teknologiaa. Järjestöön kuuluu monia kansainvälisiä yrityksiä ja järjestön jäsenillä on oikeus käyttää ZigBeeta kaupallisissa tarkoituksissa. ZigBee Alliancen uusin ZigBee-spesifikaatio julkaistiin vuonna 2006. (ZigBee Alliance.)

### 3.3 ZigBee-verkon tekniikka

ZigBee käyttää yleisimmin vapaassa käytössä olevaa 2,4 GHz ISM-taajuusaluetta, jossa on 16 kanavaa ja 250 kb/s tiedonsiirtonopeus. Lisäksi käytössä on Pohjois-Amerikan 915 MHz taajuusalue 10 kanavalla ja yksikanavainen Euroopan ja Japanin 868 MHz taajuusalue. Vastaavasti näiden taajuusalueiden tiedonsiirtonopeudet ovat 40 kb/s ja 20 kb/s. Korkeammalla taajuudella saavutetaan huomattavasti paremmat tiedonsiirtonopeudet, mutta joissakin tapauksissa matalammat taajuudet ovat käytännöllisempiä niiden paremman läpäisykyvyn ja virransäästön takia. ZigBeen signaalin kantama vaihtelee 10 metristä 100 metriin. Signaalin kantamaan vaikuttaa käytetty antenni, taajuus, lähetysteho ja ympäristö. (ZigBee: IEEE 802.15.4 2007, 15-17.)

Tiedonsiirrossa ZigBee hyödyntää CSMA/CA-menetelmää siirtotien vaurastamiseen. Ennen tiedon lähettämistä laite kuuntelee käyttämäänsä kanavaa ja tarkastaa, onko kanava vapaa. Jos kanava on varattu, laite odottaa satunnaisen ajan ja tarkastaa kanavan myöhemmin uudestaan. (ZigBee: IEEE 802.15.4 2007, 15.)

Tiedon lähetyksessä ZigBee käyttää O-QPSK-modulaatiota, jonka avulla pseudokohina moduloidaan informaatio-signaalin kanssa. Saatu signaali edelleen käsitellään käyttämällä DSSS-modulaatiota, jossa lähetettävä sanoma jaetaan osiin ja levitetään käytössä olevalle taajuusalueelle kertomalla signaali kantoaallon kanssa. (ZigBee: IEEE 802.15.4 2007, 11.)

Kun sanoma on vastaanotettu, vastaanottopää lähettää kiittauksen. Jos lähetettävä pää ei saa kiittausta tietyn ajan kuluessa, sanoma lähetetään uudestaan. ZigBee sisältää myös tuen pieniä latensseja vaativille laitteille, jolloin tiedonsiirto pysyy nopeana ja luotettavana. (ZigBee: IEEE 802.15.4 2007, 6.)

ZigBee-verkon laitteilla on 802.15.4-standardin mukaan kaksi eri osoitetta. MAC-osoite on 64-bittinen yksilöllinen osoite, jonka avulla laite tunnistetaan. Verkko-osoite on 16-bittinen paikallinen osoite, joka annetaan laitteelle verkon sisällä. (ZigBee: IEEE 802.15.4 2007, 6.)

### 3.4 ZigBee-verkon tietoturva

ZigBeen tietoturva, joka perustuu 128-bittiseen AES-salausmenetelmään, laajentaa 802.15.4-standardin tietoturvaa. 802.15.4-standardin MAC-kerros mahdollistaa salausavaimien ja pääsilystojen käyttämisen. ZigBeen

tietoturvan ominaisuuksiin kuuluu salausavaimien luominen ja siirtäminen, laitehallinta ja kehysten suojaaminen. (Getting Started with ZigBee and IEEE 802.15.4, 18-20.)

Keskeinen osa ZigBeen tietoturvassa on Trust Center, jonka tehtävänä on tunnistaa verkkoon liittyvät laitteet, ylläpitää ja jakaa salausavaimia ja ottaa käyttöön päästä päähän tietoturvan kahden laitteen välille. Trust Centerinä yleensä toimii verkon koordinaattori, mutta se voi myös olla jokin erillinen laite. (Getting Started with ZigBee and IEEE 802.15.4, 18-20.)

ZigBee käyttää kolmea erityyppistä salausavainta tietoturvan hallinnassa. Yleisavainta käytetään kahden laitteen välillä, kun ne luovat linkkiavaimen. Linkkiavainta käytetään kahden laitteen välillä viestien salaamiseen. Verkoavain on kaikkien saman verkon laitteiden tiedossa ja sitä käytetään verkkoon yhdistämisessä. (Getting Started with ZigBee and IEEE 802.15.4, 18-20.)

ZigBeessä on käytössä kaksi tietoturvasoa. Normaalilla tasolla käytetään verkko- ja linkkiavaimia, jotka on esijaettu laitteille. Laitteet eivät tarvitse todennusta liittyäkseen verkkoon. Trust Center ei ole pakollinen, mutta sitä voidaan käyttää verkkoavaimen jakamiseen. Yleisavaimen käyttö ja linkkiavainten luonti on valinnaista. (Getting Started with ZigBee and IEEE 802.15.4, 18-20.)

Korkealla tasolla kaikkien kolmen avaimen ja Trust Centerin käyttäminen on pakollista. Trust Center siirtää verkko- ja linkkiavaimet laitteisiin ja jokaisen laitteen pitää todentaa itsensä naapurilaitteiden kanssa. (Getting Started with ZigBee and IEEE 802.15.4, 18-20.)

### 3.5 ZigBee-verkon osat

ZigBee-verkko koostuu kolmesta erityyppisestä laitteesta. Coordinator eli koordinaattori on verkon juuri ja se voi toimia siltana eri verkkojen välillä. Koordinaattori säätää parametrit verkon luomista varten ja tallentaa tietoja verkosta kuten esim. radio-kanavista ja salausavaimista. Jokainen ZigBee-verkko tarvitsee yhden koordinaattorin. Koordinaattorilla on aina verkon ensimmäinen osoite 0. (ZigBee: IEEE 802.15.4 2007, 7.)

Router eli reititin toimii tiedon välittäjänä laitteiden välillä. Sen päätarkoitus on kasvattaa verkon kantamaa. Reititin voi myös toimia päätelaitteena. (ZigBee: IEEE 802.15.4 2007, 7.)

End device eli päätelaite on yksinkertaisin näistä kolmesta laitteesta. Se ei vaadi paljon resursseja eikä sillä ole monia toimintoja. Päätelaite voi pelkästään kommunikoida koordinaattorin tai reitittimen kanssa. (ZigBee: IEEE 802.15.4 2007, 7.)

### 3.6 ZigBee-verkot ja topologiat

ZigBee-verkkoja voi olla kahta eri tyyppiä. Radiomajakalla varustetussa verkossa reititin lähettää säännöllisiä sanomia muille verkossa oleville laitteille varmistaakseen niiden läsnäolon. Kaikki verkon laitteet rekisteröityvät koordinaattorille ja tarkastavat onko sanomia saatavilla. Kun sanomat on siirretty, laitteet menevät lepotilaan ja heräävät myöhemmin koordinaattorin määräämän aikataulun mukaisesti. Myös itse koordinaattori on lepotilassa saman aikataulun mukaisesti. Tällaisessa verkossa laitteet voivat olla sanomien välisen ajan lepotilassa, jolloin virrankulutus pysyy alhaisena. Toisaalta verkko vaatii tarkkaa ajoitusta, mikä saattaa lisätä kustannuksia. (ZigBee: IEEE 802.15.4 2007, 8.)

Verkossa, jossa ei ole radiomajakkaa, laitteet heräävät lepotilasta säännöllisesti satunnaisin väliajoin ilmoittaakseen läsnäolonsa verkossa. Kun laite havaitsee tapahtuman, se herää lepotilasta ja lähettää sanoman. Koordinaattorilla ja reitittimellä on vastaanottimet koko ajan päällä, jotta ne voivat kuunnella jokaista laitetta. Tällaisessa verkossa virrankulutus on epäsymmetrinen, koska vastaanottimet ovat päällä, mutta päätelaitteet voivat olla lepotilassa suurimman osan ajasta. (ZigBee: IEEE 802.15.4 2007, 9.)

ZigBee tukee kolmea eri verkkotopologiaa. Tähtitopologia koostuu koordinaattorista ja päätelaitteista. Verkon keskellä on koordinaattori, jonka kautta kaikki liikenne kulkee. Tähtitopologian etuna on sen yksinkertaisuus, mutta koordinaattori saattaa aiheuttaa pullonkaulan, jos päätelaitteita on paljon. Lisäksi verkossa ei ole vaihtoehtoisia reittejä liikenteelle. (Elahi 2009, 35-39.)

Puutopologiassa verkon juurena toimii koordinaattori, johon on liitetty reitittimiä ja päätelaitteita. Reitittimet ovat solmupisteitä, joihin voidaan liittää lisää reitittimiä ja päätelaitteita, jolloin syntyy puurakenne. Puutopologiassa verkon kantamaa voidaan jatkaa lisäämällä solmupisteitä. Toisaalta, jos solmupiste vikaantuu, liikenne ei toimi solmupisteeseen liitettyihin laitteisiin. (Elahi 2009, 35-39.)

Mesh-topologia koostuu myös koordinaattorista, reitittimistä ja päätelaitteista. Mesh-verkko muistuttaa puutopologiaa, mutta sen erikoisuutena on redundanssi eli solmupisteiden välillä on monia vaihtoehtoisia reittejä. Mesh-verkko pystyy automaattisesti mukautumaan topologian muutoksiin, mikä parantaa verkon vikasietoisuutta. (Elahi 2009, 35-39.)

### 3.7 ZigBeen käyttökohteet

ZigBee on erityisesti tarkoitettu vähävirtaisiin sulautettuihin järjestelmiin, jotka eivät tarvitse nopeaa tiedonsiirtoa. Etenkin paristokäyttöiset laitteet hyötyvät ZigBeestä. Tyypilliset ZigBeen käyttökohteet ovat sensoriverkot ja automaatio. ZigBee-laitteita voidaan käyttää esim. teollisuudessa erilaisissa mittauskohteissa. (Wikipedia - ZigBee.)

### 3.8 XBee

XBee on Digi International -yhtiön kehittämä tuotemerkki radio-antenneille, jotka perustuvat IEEE 802.15.4-standardiin. Tuoteperhe koostuu XBee- ja XBee PRO-moduuleista, joihin kuuluu monia eri malleja ja antennityyppejä. XBee PRO -moduulien etuna on suuremmat lähetystehot, jolloin signaalin kantama on parempi. Lisäksi jotkin XBee PRO -moduulit ovat ohjelmoitavissa. Eri malleihin kuuluu tuki 802.15.4-, ZigBee-, 9XStream- ja DigiMesh-yhteyksille. Kuvassa 5 on esitetty XBee 802.15.4 1mW lanka-antenni. (Digi - XBee.)

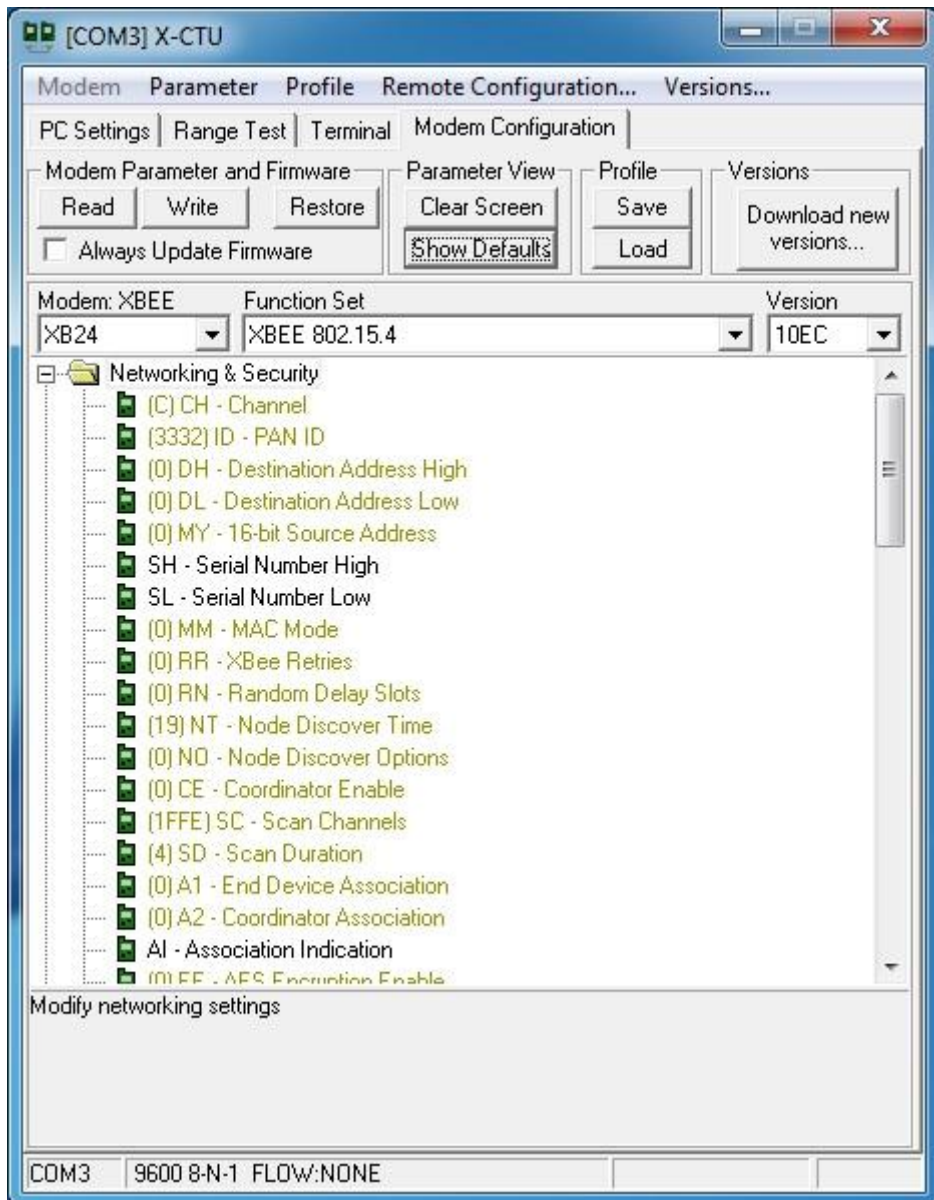


Kuva 5. XBee 802.15.4 1mW lanka-antenni

XBee 802.15.4 -moduulin tärkeimmät ominaisuudet ovat seuraavat:

- 1mW lähetysteho (0 dBm)
- toimintataajuus 2,4 GHz
- signaalin kantama sisällä 30 metriä ja ulkona 90 metriä
- vastaanottimen herkkyys -92 dBm
- virrankulutus 3,3V jännitteellä lähetyksessä 45mA ja vastaanotossa 50mA (Digi - XBee 802.15.4.)

XBee-moduulien asetuksia ja ohjelmistoa voidaan muokata Digi Internationalin kehittämällä X-CTU -ohjelmalla yhdistämällä XBee-moduuli USB:n välityksellä tietokoneeseen. Kuvassa 6 on esitetty X-CTU -ohjelman pääikkuna.



Kuva 6. X-CTU -ohjelman pääikkuna

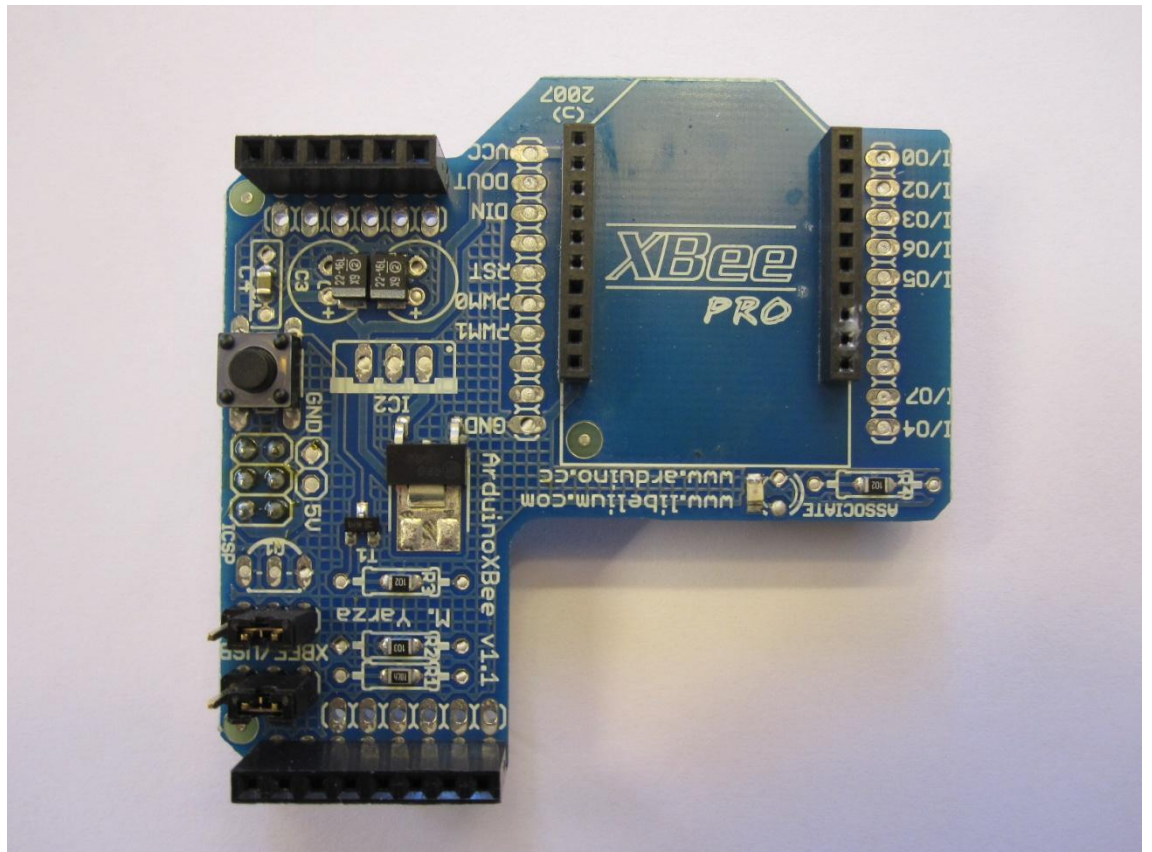
## 4 TYÖN TOTEUTUS

### 4.1 Tarvikkeet

Työn toteutusta varten hankittiin Arduino Starter Kit -pakkaus, joka sisältää mm. Arduino Uno R3 -alustan, Arduino-projektikirjan, koekytkentälävyn, johtimia, vastuksia, lämpötila-anturin, LCD-näytön ja muita komponentteja. Kuvassa 7 on esitetty Arduino Starter Kit -pakkaus.







Kuva 8. Arduino XBee shield

Työssä käytetty XBee-moduuli on 1mW lanka-antenni, joka hyödyntää 802.15.4-standardin yhteyttä. Sen avulla on mahdollista ottaa langaton yhteys toiseen Arduinoon. Se valittiin, koska ZigBee-yhteyttä tukevat XBee-moduulit ovat kalliimpia ja työssä käytetyn verkon yksinkertaisuuden vuoksi ZigBee sisältää liikaa ylimääräisiä ominaisuuksia. Muutoin työn toteutus on samankaltainen, vaikka käytössä olisi ZigBee-moduuli.

Kaikki työssä käytettävät tarvikkeet on hankittu omaan käyttöön. Tarvikkeiden kustannukset on esitetty taulukossa 1.

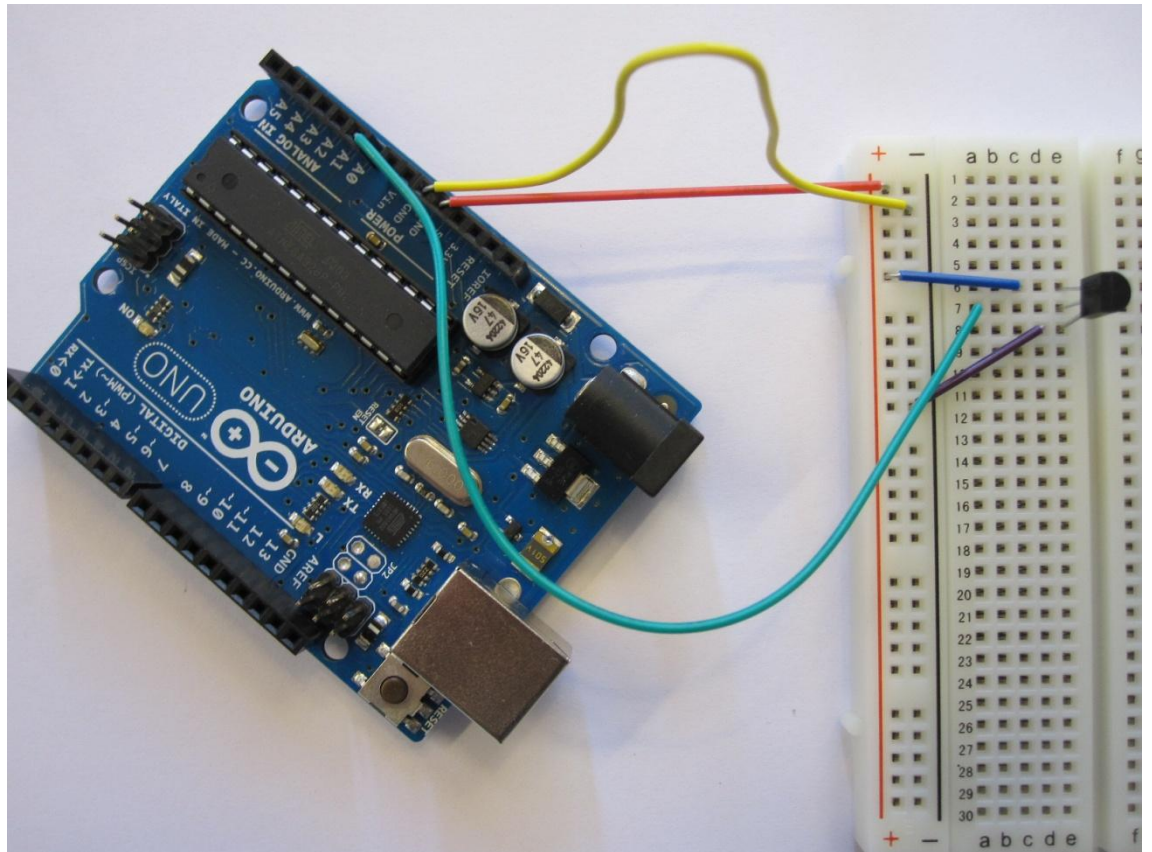
Taulukko 1. Materiaalikustannukset sisältäen alv 24%

Tuote	Kappalemäärä	Yksikköhinta	Kokonaishinta
Arduino Starter Kit	1	99,95€	99,95€
Arduino Uno R3	1	25,00€	25,00€
XBee 1mW lanka-antenni	2	30,20€	60,40€
XBee shield	2	15,00€	30,00€
Yhteensä			215,35€

Näistä osista on tarkoitus rakentaa kaksiosainen Arduino-lämpömittari, joka langatonta 802.15.4-standardin yhteyttä käyttäen siirtää lämpötila-anturin tiedot toisen Arduinon LCD-näytölle. Komponentit yhdistetään käyttämällä koekytkentälevyä, mikä mahdollistaa virtapiirien testaamisen ilman komponenttien yhteen juottamista.

### 4.2 Lämpötila-anturin kytkeminen

Lämpötila-anturi kytketään ensimmäiseen Arduino Uno R3:en eli lähettävään päähän. Käytetty anturi on mallia TMP36, joka pystyy mittaamaan lämpötiloja  $-55^{\circ}\text{C}$  ...  $150^{\circ}\text{C}$  väliltä. TMP36:ssa on kolme pinniä, joista yksi liitetään Arduinon 5V pinniin, toinen liitetään analogiseen pinniin A0 ja kolmas liitetään maadoituspinniin. Kuvassa 9 on esitetty lämpötila-anturin liittäminen koekytkentälevyyn. (Low Voltage Temperature Sensors TMP35/TMP36/TMP37 2010, 4.)



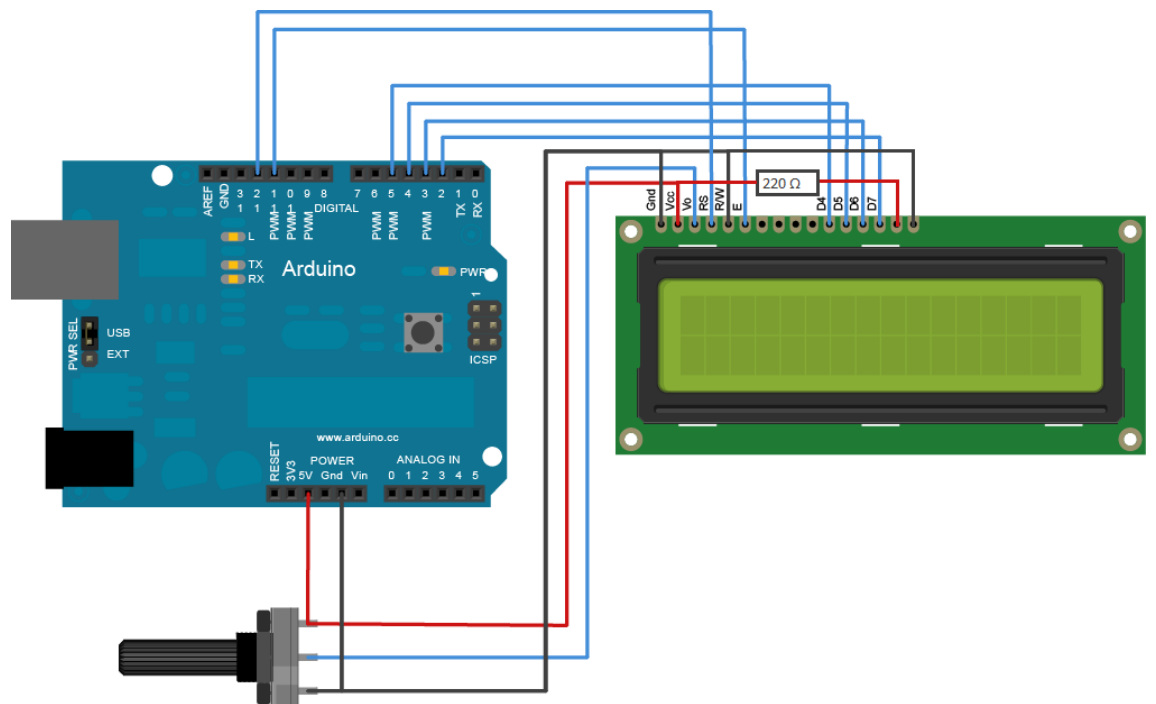
Kuva 9. Lämpötila-anturi koekytkentälevyyn liitettynä

Koekytkentälevyn kytkennät on tehty seuraavasti. Punainen johdin on liitetty Arduinon 5V-pinniin ja koekytkentälevyn plus-johtimeen. Keltainen johdin on liitetty Arduinon maadoituspinniin ja koekytkentälevyn miinus-johtimeen. Sininen johdin on yhdistetty lämpötila-anturin litteästä päästä katsottuna oikeanpuoleisen pinnin ja koekytkentälevyn plus-johtimen välille. Purppura johdin on liitetty lämpötila-anturin vasemmanpuoleisen pinniin ja koekytkentälevyn miinus-johtimeen. Vihreä johdin on kytketty lämpötila-anturin keskimmäisestä pinnistä Arduinon analogiseen A0-pinniin, josta anturin lämpötilatiedot luetaan. (Fitzgerald 2012, 44-45.)

### 4.3 LCD-näytön kytkeminen

LCD-näyttö kytketään toiseen Arduino Uno R3:en eli vastaanottavaan päähän. Käytetty näyttö on mallia WH1602B-TMI-ET aakkosnumeerinen

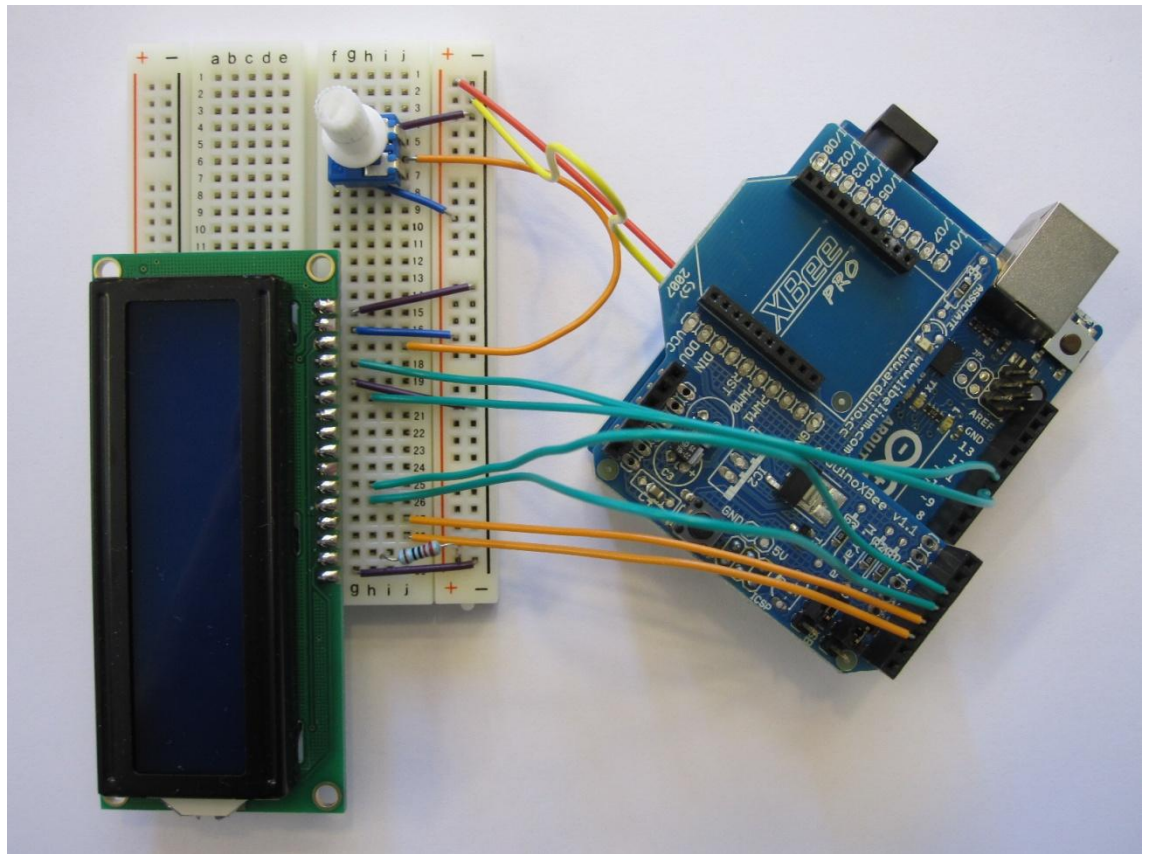
16x2 merkin LCD-näyttö. Kuvassa 10 on esitetty kaavio LCD-näytön kytkennöistä Arduinoon.



Kuva 10. Kaavio LCD-näytön kytkennöistä Arduinoon (LCD)

LCD-näytön pinnit on kytketty seuraavasti. Gnd-maadoituspinni on yhdistetty Arduinon maadoituspinniin. R/W-pinni, joka vaihtaa LCD-näytön luku- tai kirjoitustilaan, on myös yhdistetty Arduinon maadoituspinniin. Vcc on näytön virtapinni ja se on liitetty Arduinon 5V-pinniin. Kuvan 10 alareunassa on potentiometri, joka on kytketty Arduinon 5V- ja maadoituspinneihin. Keskimäinen potentiometrin pinni on liitetty LCD-näytön V0-pinniin. Potentiometriä säätämällä voidaan muuttaa näytön kontrastia. RS-pinni, joka määrittää mihin merkit tulostuvat näytöllä, on yhdistetty Arduinon digitaaliseen pinniin 12. E-pinni on kytketty Arduinon digitaaliseen pinniin 11 ja se lähettää LCD-näytölle tiedon tulevista komennoista. Datapinnejä D4-D7 käytetään merkkien lähettämiseen näytölle ja ne on vastaavasti liitetty Arduinon digitaalisiin pinneihin 5-2. Lisäksi näytössä on vielä taustavalon pinnit LED-plus ja LED-miinus. LED-plus on yhdistetty 220  $\Omega$  vastuksen kautta Arduinon 5V-pinniin ja LED-miinus on liitetty maadoituspinniin. Kuvassa 11 on esitetty LCD-näytön kytkennät koekytkentälevyllä. (Fitzgerald 2012, 116-117.)



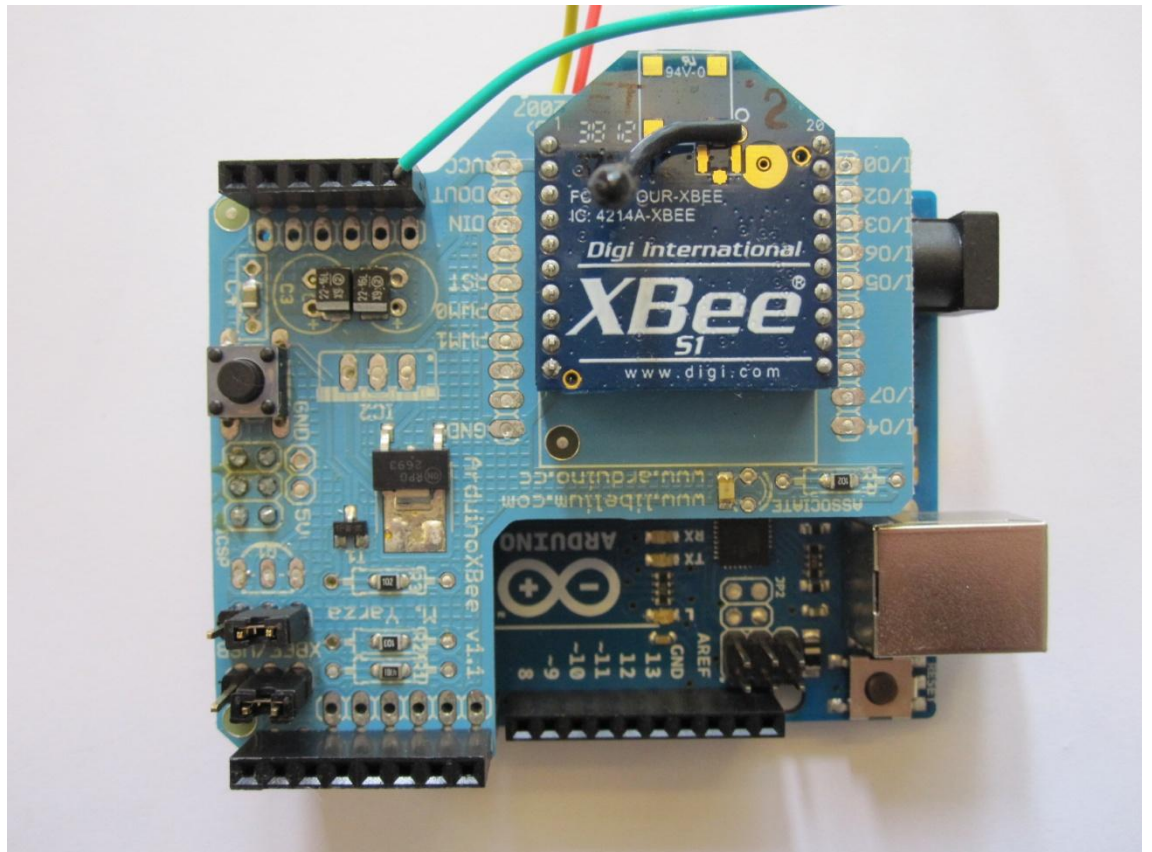


Kuva 11. LCD-näyttö koekytkentälevyyn liitettynä

Kuvassa 11 on huomioitava, että Arduinoon on valmiiksi asennettu XBee shield-lisälaite myöhempää käyttöä varten.

#### 4.4 Langattoman yhteyden luominen

Arduinot saadaan kommunikoidaan keskenään käyttämällä XBee-moduulia, joka toimii radioantennina 802.15.4-standardin yhteydelle. Kumpaankin Arduinoon asennetaan XBee shield-lisälaite ja XBee-moduuli liitetään sille varattuun paikkaan. Kuvassa 12 on esitetty lähettävän pään Arduino, johon on asennettu XBee-moduuli.



Kuva 12. XBee-moduuli asennettuna lähettävän pään Arduinoon

Työssä käytetty verkkotopologia on Point-to-Point -verkko, koska tiedonsiirtoa tarvitaan vain kahden pisteen välillä. Tällöin kummankin XBee-moduulin pitäisi toimia keskenään suoraan oletusasetuksilla. Oletuksena XBee-moduulin tiedonsiirtonopeus on 9600 baudia. Tiedonsiirtonopeus pitää olla sama sekä lähettävässä että vastaanottavassa päässä. Lisäksi XBee-moduulien pitää olla samassa verkossa samalla kanavalla, jotta langaton yhteys toimii. Oletuksena verkon PAN ID on 3332. Käytetty kanava on heksadesimaaliarvoltaan 0x0C eli desimaalina 12, mikä vastaa 2,410 GHz taajuutta. Myös verkon osoitteiden pitää olla oikein määritettynä. XBee-moduulin oma osoite MY sekä kohdeosoitteet DH ja DL ovat oletuksena 0. (Arduino - ArduinoXBeeShield.)

Näitä asetuksia voidaan muokata X-CTU -ohjelmalla yhdistämällä XBee-moduuli USB:n välityksellä tietokoneeseen. Tähän voi käyttää erillistä XBee Explorer USB -adapteria tai käyttää Arduinoa hyödyksi. Tällöin kuitenkin Arduinon pitää olla ns. reset-moodissa eli Reset-pinni yhdistetään GND-pinniin ja XBee shield-lisälaitteen jumpperit siirretään USB-asettoon.

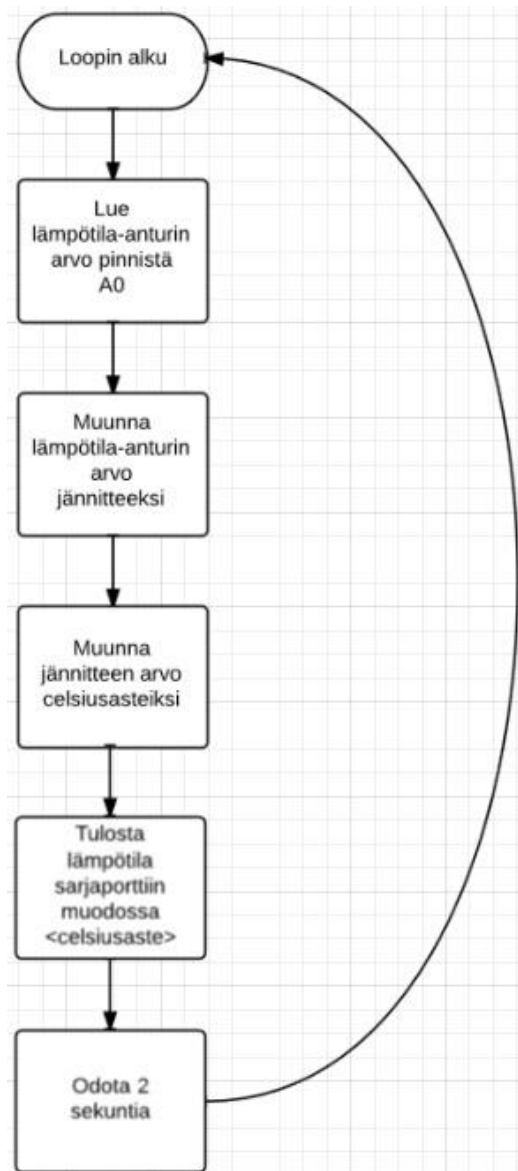
### 4.5 Ohjelmointi

Fyysisten kytkentöjen jälkeen molemmat Arduinot on ohjelmoitava, jotta ne toimisivat halutulla tavalla. Ohjelmointi tehtiin käyttämällä Arduino IDE -ohjelmaa, jonka avulla kirjoitettiin ja lähetettiin ohjelmakoodit mo-

lempiin Arduinoihin. Liitteissä 1 ja 2 on esitetty molempien Arduinojen ohjelmakoodit.

Ensimmäisen Arduino Uno R3:n eli lähettävän pään ohjelma toimii seuraavasti. Aluksi määritetään luettava pinni A0 ja lukuerottimet, joita käytetään jokaisen lähetettävän kokonaisluvun ympärillä, jotta vastaanottava Arduino osaa tulkita ne oikein. Setup-funktio avaa sarjamuotoisen yhteyden 9600 baudin tiedonsiirtonopeudella. Tätä yhteyttä käytetään XBee-moduulien kanssa Arduinojen välillä.

Loop-funktio suorittaa ohjelman pääosuden, jossa pinnistä A0 luetaan lämpötila-anturin antamat arvot. Nämä arvot muunnetaan jännitteeksi ja edelleen celsiusasteiksi. Lopuksi lämpötilatiedot tulostetaan sarjaporttiin lukuerottimet lisättyinä alkuun ja loppuun. Tämä sarjamuotoinen data siirtyy kahden sekunnin välein XBee-moduulin kautta langattomasti toiseen Arduinoon. Kuvassa 13 on esitetty lähettävän pään Arduinon loop-funktion tilakaavio.



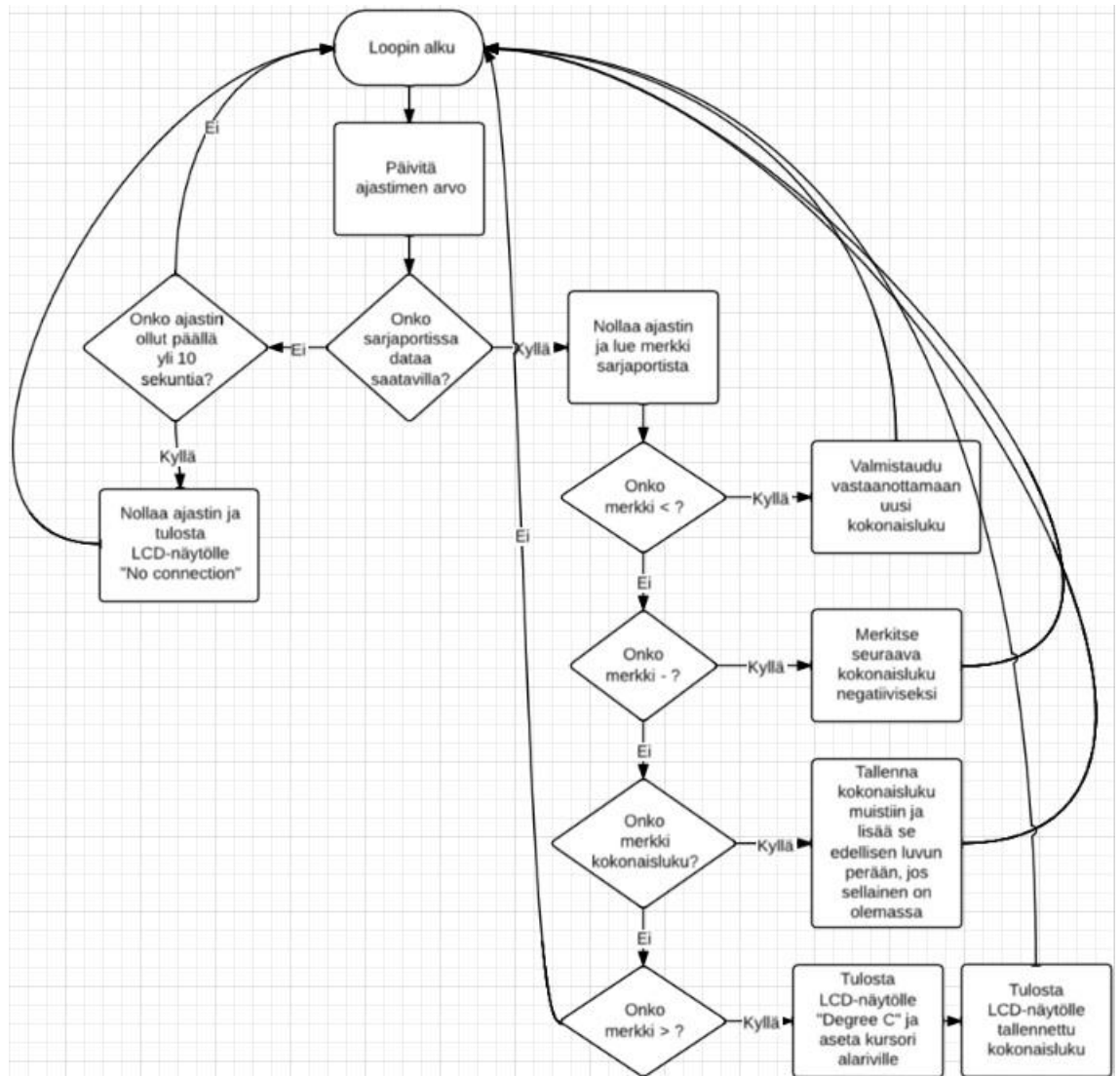
Kuva 13. Tilakaavio lähettävän pään Arduinon loop-funktiosta

Lämpötilatiedot on pyöristetty alaspäin asteen tarkkuuteen, jotta datan tulkitseminen vastaanottopäässä olisi yksinkertaisempaa. Lisäksi asteen tarkkuus on riittävä testauksen suorittamista varten.

Toisen Arduino Uno R3:n eli vastaanottavan pään ohjelma toimii seuraavasti. Aluksi otetaan käyttöön LiquidCrystal-kirjasto, joka sisältää LCD-näytön ohjaamiseen tarvittavat komennot, ja määritetään käytetyt LCD-näytön pinnit. Myös samat lukuerottimet, joita lähettävässä päässä käytetään, pitää määrittää. Lisäksi tarvitaan muutama muuttuja ajastimen käyttöä varten. Arduino ei sisällä sisäistä kelloa, joten ajastin tehdään keinotekoisesti millis-funktion avulla. Millis-funktio laskee millisekunteina ajan, jonka Arduino on ollut päällä. Setup-funktio avaa sarjamuotoisen yhteyden samalla 9600 baudin tiedonsiirtonopeudella kuin lähettävässä päässä. Samalla myös avataan yhteys LCD-näytölle ja ohjelmalle kerrotaan näytön koko, joka on 16 merkkiä kahdessa rivissä.

Loop-funktion alussa päivitetään ajastimen arvo. Sitten tarkastetaan, onko XBee-moduulin kautta tullut sarjaporttiin luettavaa dataa saataville. Jos dataa ei ole ollut saatavilla yli kymmeneen sekuntiin, LCD-näytölle tulostetaan "No connection". Jos dataa on saatavilla, ohjelma lukee datan merkki kerrallaan ja tallentaa kokonaisluvut muistiin. Kun viimeinen merkki eli lukuerotin on vastaanotettu, LCD-näytölle tulostetaan lämpötila celsiusasteina. Kuvassa 14 on esitetty vastaanottavan pään Arduinon loop-funktion tilakaavio.





Kuva 14. Tilakaavio vastaanottavan pään Arduinon loop-funktiosta

#### 4.6 Testaus

Valmiilla lämpömittarilla suoritetaan testit sisällä ja ulkona. Sisätilatestissä vastaanottava Arduino on paikallaan pöydällä ja lähettävää Arduinoa siirretään tarpeen mukaan. Etäisyys Arduinojen välillä mitataan ja lämpötilaa verrataan toisen digitaalisen lämpömittarin antamiin arvoihin. Sisätilatestissä Arduinojen välillä on paikasta riippuen muutamia seiniä, mutta huoneiden ovet pidetään auki. Taulukossa 2 on esitetty sisätilatestin tulokset.

Taulukko 2. Sisätilatestin tulokset

Etäisyys Arduinojen välillä	Arduinon näyttämä lämpötila	Toisen lämpömittarin arvo
1 m	23°C	23,2°C
5 m	23°C	23,6°C
10 m	24°C	23,8°C
15 m	23°C	23,9°C

Testiä ei voitu jatkaa pidemmälle tilanpuutteen vuoksi, mutta ainakin 15 metrin etäisyydellä Arduinojen langaton yhteys pysyi päällä. Tuloksissa on otettava huomioon, että Arduinon mittaama lämpötila on pyöristetty alaspäin asteen tarkkuuteen. Kuitenkin tuloksista voi päätellä, että lähettävään Arduinoon liitetty lämpötila-anturi antaa lähes samoja arvoja kuin testissä käytetty toinen lämpömittari.

Ulkotilatestissä vastaanottava Arduino on paikallaan avonaisen ikkunan edessä ja lähettävää Arduinoa siirretään tarpeen mukaan. Ulkotilatestissä Arduinojen välillä on lähes esteetön linja lukuun ottamatta varastotilojen ovia. Taulukossa 3 on esitetty ulkotilatestin tulokset.

Taulukko 3. Ulkotilatestin tulokset

Etäisyys Arduinojen välillä	Lähettävän Arduinon paikka	Arduinon näyttämä lämpötila	Toisen lämpömittarin arvo
12 m	Varasto 1, ovi kiinni	22°C	22,1°C
25 m	Varasto 2, ovi auki	21°C	21,3°C
25 m	Varasto 2, ovi kiinni	-	21,3°C
35 m	Avomaasto	26°C	26,1°C
45 m	Avomaasto	-	26,1°C

Ulkotilatestissä Arduinojen langaton yhteys katkesi, kun 25 metrin etäisyydellä olevassa varastossa pistettiin ovi kiinni, vaikka XBee-moduulin signaalin kantaman pitäisi riittää 30 metriin sisätiloissa. Mittausta jatkettiin avomaastossa, jolloin langaton yhteys katkesi 45 metrin etäisyydellä, mikä jäi vain noin puoleen XBee-moduulin väitetyistä signaalin kantamasta ulkona. Myös tässä testissä Arduinoon liitetty lämpötila-anturi antoi lähes samoja arvoja kuin vertailukohteena ollut lämpömittari. Tulosten mukaan Arduino-lämpömittaria voidaan käyttää ensimmäisessä varastossa.

## 5 YHTEENVETO

Opinnäytetyön tavoitteena oli toteuttaa lämpötila-anturilla varustettu Arduino, joka lähettää anturin tiedot langattomasti toisen Arduinon LCD-näytölle. Työssä päästiin tavoitteeseen ja tuloksena saatiin toimiva Arduino-lämpömittari.

Työn edetessä kävi ilmi, kuinka monipuolinen alusta Arduino on ja miten erilaisissa projekteissa sitä käytetään. Myös ZigBee-standardi yllättää positiivisesti vähävirtaisuudellaan ja verkko-ominaisuuksiltaan, kun sitä verrataan samankaltaiseen Bluetooth-protokollaan.

Suurimmaksi ongelmaksi työssä muodostui Arduinojen ohjelmointi. Langaton yhteys Arduinojen välillä toimii sarjamuotoisesti, mikä pitää ottaa huomioon tiedonsiirrossa. Tämä tarkoittaa sitä, että lämpötiladataan pitää liittää alku- ja loppumerkit, jolloin vastaanottopäässä data pystytään tulkit-

semaan oikein. Toimivan ohjelmakoodin luominen ja testaaminen vaati suurimman osan ajasta.

Arduino-lämpömittarin testaus suoritettiin sisällä ja ulkona. Testauksessa mitattiin lämpötila-anturin tarkkuutta ja XBee-moduulin signaalin kantamaa. Testauksen tulosten mukaan tuotos on tarpeeksi tarkka lämpötilan mittaamiseen, mutta signaalin kantama jäi alle XBee-moduulin spesifikaation signaalin kantamasta. Kuitenkin Arduino-lämpömittaria voidaan käyttää vain ensimmäisen varaston lämpötilan seurantaan, koska langaton yhteys katkesi toisessa varastossa.

Työtä on mahdollista kehittää eteenpäin esimerkiksi ohjelmoimalla Arduinot mittaamaan ja näyttämään lämpötila asteen kymmenesosan tarkkuudella. Lisäksi lämpötila-antureista voitaisiin rakentaa anturiverkko, joka mittaa lämpötilaa monesta eri paikasta, mutta anturiverkon luominen vaatii lisää laitehankintoja.

Opinnäytetyön tekeminen ja varsinkin Arduinon kanssa työskentely oli mielenkiintoista. Suosittelen elektroniikasta ja ohjelmoinnista kiinnostuneita tutustumaan Arduinoon ja sen tarjoamien projektien toteutuksiin.

## LÄHTEET

Arduino. Viitattu 21.5.2013. <http://arduino.cc/en/>

Arduino: A Brief History. Viitattu 21.5.2013.  
<http://www.kenleung.ca/portfolio/arduino-a-brief-history-3/>

Arduino - ArduinoXBeeShield. Viitattu 11.6.2013  
<http://arduino.cc/en/Main/ArduinoXbeeShield>

Arduino - Introduction. Viitattu 21.5.2013.  
<http://arduino.cc/en/Guide/Introduction>

Arduino - Language Reference. Viitattu 23.5.2013.  
<http://arduino.cc/en/Reference/HomePage>

Arduino - Playground. Viitattu 18.7.2013. <http://playground.arduino.cc/>

Arduino - Policy. Viitattu 22.5.2013. <http://arduino.cc/en/Main/Policy>

Arduino Uno. Viitattu 27.5.2013.  
<http://arduino.cc/en/Main/arduinoBoardUno>

ATmega8. Viitattu 27.5.2013.  
<http://www.hpselectronics.com/images/atmega8.jpg>

Digi - XBee. Viitattu 26.6.2013. <http://www.digi.com/xbee/>

Digi - XBee 802.15.4. Viitattu 26.6.2013.  
<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#specs>

Elahi, A., Gschwender, A. 2009. ZigBee Wireless Sensor and Control Network. Viitattu 2.7.2013.

Fitzgerald, S., Shiloh, M. 2012. Arduino Projects Book. Viitattu 5.6.2013.

Getting Started with ZigBee and IEEE 802.15.4. 2008. Daintree Networks. Viitattu 8.7.2013.  
[http://www.daintree.net/downloads/whitepapers/zigbee\\_primer.pdf](http://www.daintree.net/downloads/whitepapers/zigbee_primer.pdf)

IEEE 802.15.4. Viitattu 17.6.2013.  
<http://www.ieee802.org/15/pub/TG4.html>

LCD. Viitattu 6.6.2013.  
[http://arduino.cc/en/uploads/Tutorial/LCD\\_bb.png](http://arduino.cc/en/uploads/Tutorial/LCD_bb.png)

Low Voltage Temperature Sensors TMP35/TMP36/TMP37. 2010. Analog Devices. Viitattu 5.6.2013.  
[http://arduino.cc/documents/datasheets/TEMP-TMP35\\_36\\_37.pdf](http://arduino.cc/documents/datasheets/TEMP-TMP35_36_37.pdf)

MegaAVR Microcontroller. Viitattu 27.5.2013.

<http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx>

Wikipedia - Arduino. Viitattu 22.5.2013.

<http://en.wikipedia.org/wiki/Arduino>

Wikipedia - ZigBee. Viitattu 15.7.2013.

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

ZigBee Alliance. Viitattu 16.7.2013.

<http://www.zigbee.org/About/AboutAlliance/TheAlliance.aspx>

ZigBee: IEEE 802.15.4. 2007. Tampere University of Technology.

Viitattu 19.6.2013.

<http://www.cs.tut.fi/kurssit/TLT-6556/Slides/4-802.15ZigBee.pdf>

## LÄHETTÄVÄN PÄÄN ARDUINON OHJELMAKOODI

```
const int sensorPin = A0;
const char startOfNumberDelimiter = '<';
const char endOfNumberDelimiter = '>';

void setup(){
  Serial.begin(9600);
}

void loop(){
  int sensorVal = analogRead(sensorPin);
  float voltage = (sensorVal/1024.0) * 5.0;
  int temperature = (int)((voltage - 0.5) * 100);
  Serial.print(startOfNumberDelimiter);
  Serial.print(temperature);
  Serial.print(endOfNumberDelimiter);
  delay(2000);
}
```

## VASTAANOTTAVAN PÄÄN ARDUINON OHJELMAKOODI

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const char startOfNumberDelimiter = '<';
const char endOfNumberDelimiter = '>';
unsigned long previousMillis = 0;
long interval = 10000;

void setup(){
  Serial.begin(9600);
  lcd.begin(16, 2);
}

void processNumber(const long n)
{
  lcd.clear();
  lcd.print("Degree C");
  lcd.setCursor(0, 1);
  lcd.print(n);
}

void processInput()
{
  static long receivedNumber = 0;
  static boolean negative = false;

  byte c = Serial.read();

  switch(c)
  {

    case endOfNumberDelimiter:
      if (negative)
        processNumber(- receivedNumber);
      else
        processNumber(receivedNumber);

    case startOfNumberDelimiter:
      receivedNumber = 0;
      negative = false;
      break;

    case '0' ... '9':
      receivedNumber *= 10;
      receivedNumber += c - '0';
      break;
```

## VASTAANOTTAVAN PÄÄN ARDUINON OHJELMAKOODI

```
        case '-':
            negative = true;
            break;

    }
}

void loop(){
    unsigned long currentMillis = millis();
    if(Serial.available()>0){
        processInput();
        previousMillis = currentMillis;
    }
    else
    {
        if(currentMillis - previousMillis > interval) {
            previousMillis = currentMillis;
            lcd.clear();
            lcd.print("No connection");
        }
    }
}
```