



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Jesse Koski

LANGATON LÄMPÖTILA- JA  
KOSTEUSMITTAUSJÄRJESTELMÄ  
ZIGBEE-PROTOKOLLAA KÄYTTÄEN

Tekniikka ja liikenne  
2016

## TIIVISTELMÄ

Tekijä	Jesse Koski
Opinnäytetyön nimi	Langaton lämpötila- ja kosteusmittausjärjestelmä ZigBee-protokollaa käyttäen
Vuosi	2016
Kieli	suomi
Sivumäärä	45
Ohjaaja	Santiago Chávez Vega

---

Opinnäytetyö toteutettiin Vaasan ammattikorkeakoululle koulutusohjelman päättötöyänä. Työn tavoitteena oli toteuttaa langaton lämpötilan, kosteuden ja kastepisteen mittaus käyttäen ZigBee-tekniikkaa.

Toteutuksessa käytettiin mikrokontrollerina Arduinoa, johon on kytketty kiinni SHT71-anturi sekä ZigBee-moduuli. Arduinon ohjelmakoodissa luetaan anturia, josta tiedot lähetetään ZigBee-moduulin kautta toiselle ZigBee-moduulille, joka on Raspberry Pi -laitteeseen kytketty kiinni. Raspberry Pi:ssä käytetään LAMP-kokoelmaa (Linux, Apache, MySQL, php). Arvojen vastaanottamiseen käytetään Raspberry Pi:ssä Python-ohjelmakoodia, joka lukee sarjaporttia, vastaanottaa arvot ja kirjoittaa ne suoraan MySQL-tietokantaan. Apache-palvelimella on php ohjelmakoodi, joka lukee SQL-tietokannasta arvot ja tulostaa ne viivakaavioon.

Opinnäytetyön tuloksena saatiin toimiva prototyyppi, joka tietyn väliajoin mittaa lämpötilan, kosteuden ja kastepisteen. Tämä voidaan lukea suoraan verkkoselaimesta viivakaaviomuodossa, joka on saatavilla kaikilla samassa lähiverkossa olevilla laitteilla.

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Information Technology

## ABSTRACT

Author	Jesse Koski
Title	Wireless temperature and humidity measurement system implemented using ZigBee protocol
Year	2016
Language	Finnish
Pages	45
Name of Supervisor	Santiago Chávez Vega

---

The aim of this thesis was to implement an RF temperature, humidity and dew point monitoring system using ZigBee technology.

The system was implemented using Arduino as microcontroller which is connected to SHT71 sensor and ZigBee module. In Arduino C/C++ code reads sensor data and send data through ZigBee to another ZigBee module wirelessly which is connected to Raspberry pi. Raspberry Pi is using LAMP-stack (Linux, Apache, MySQL, PHP). To receive data wirelessly Raspberry Pi is using Python code to read serial port continuously which is connected to ZigBee module. At the same time Python code is writing data values to SQL-database. In Apache web server there is running PHP code which is getting data values from MySQL and writes them to line chart.

The result of this thesis was a fully functional prototype system, which at certain intervals measures temperature, humidity and dew point values. All values are readable in web-browser in any device on same local network in line chart form.

---

Keywords                      Raspberry Pi, Arduino, ZigBee, C/C++, Python

## SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO .....	9
2	KÄYTETYT KOMPONENTIT JA TEKNOLOGIAT .....	11
	2.1 Arduino .....	11
	2.2 Arduino MEGA 2560 .....	11
	2.3 Raspberry Pi.....	13
	2.4 Raspberry Pi Model B.....	14
	2.5 ZIGBEE .....	15
	2.5.1 IEEE 802.15.4-standardi .....	15
	2.5.2 XBee.....	16
	2.5.3 XBee Adapter kit.....	17
	2.5.4 XBee-moduulin konfigurointi.....	19
3	KÄYTETYT OHJELMOINTIKIELET .....	20
	3.1 C-kieli .....	20
	3.2 C++-kieli .....	21
	3.3 Python-kieli.....	22
	3.4 PHP-kieli.....	23
	3.5 SQL-kieli.....	24
4	JÄRJESTELMÄN TOIMINTA JA TOTEUTUS .....	25
	4.1 Lämpötilan ja kosteuden mittaus .....	26
	4.1.1 SHT71-anturin kytkentä.....	27
	4.1.2 Arvojen lukeminen SHT71-anturilta .....	28
	4.2 ZigBee-verkon toiminta ja XBee-moduuli .....	31
	4.2.1 XBee-moduulin kytkentä XBee-adapter kitin avulla.....	31
	4.2.2 IEEE 802.15.4 liikenteen tutkiminen packet snifferillä.....	32
	4.3 Arvojen vastaanotto ja käsittely Raspberry Pillä.....	35

4.3.1	XBee-moduulin kytkentä XBee-adapter kitin avulla.....	35
4.3.2	Tulevan datan käsittely Python-koodilla.....	36
4.3.3	LAMP eli Linux, Apache, MySQL ja PHP .....	38
5	TESTAUS.....	42
5.1	Testilaitteisto.....	42
5.2	Testaus ja ilmentyvät ongelmat .....	42
6	YHTEENVETO .....	43
	LÄHDELUETTELO.....	44

**LYHENTEET JA TERMIT**

<b>Apache</b>	Avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma
<b>Arduino</b>	Avoimeen laitteistoon perustuva mikro-ohjain-/elektroniikka-alusta ja ohjelmointiympäristö
<b>AVR</b>	Atmelin mikro-ohjainperhe, sisältää laajan valikoiman 8-bittisiä mikro-ohjaimia.
<b>PHP</b>	Palvelinpuolen ohjelmointikieli
<b>HTML</b>	HyperText Markup Language, standardoitu kuvauskieli
<b>IoT</b>	The Internet of Things, esineiden internet
<b>SQL</b>	Structured Query Language, kyselykieli
<b>LAMP</b>	Linux, Apache, MySQL/MariaDB, PHP, Perl/Python, koelma avoimen lähdekoodin ohjelmia
<b>ZigBee</b>	IEEE 802.15.4-standardin mukainen lyhyen kantaman tietoliikenneverkko
<b>WPAN</b>	Wireless Personal Area Network, langaton likiverkko
<b>Python</b>	Monipuolinen ohjelmointikieli
<b>C-kieli</b>	Yleiskäyttöinen ja rakenteinen käännettävä tietokoneiden ohjelmointikieli
<b>C++-kieli</b>	Ohjelmointikieli kuin C-kieli, mutta lisätty muun muassa olio-ohjelmointiin ja generisyyteen liittyviä ominaisuuksia
<b>SHT71</b>	Lämpötila- ja ilmankosteusanturi.

<b>ASCII</b>	American Standard Code for Information Interchange. 7-bittinen, eli 128 merkkipaikan laajuinen merkistö.
<b>bps</b>	Bits per second, bittä sekunnissa

**KUVIOLUETTELO**

<b>Kuvio 1.</b>	Arduino Mega 2560	s. 12
<b>Kuvio 2.</b>	Arduino Mega 2560-kytkentäkaavio	s. 12
<b>Kuvio 3.</b>	Raspberry Pi	s. 13
<b>Kuvio 4.</b>	Raspberry Pin GPIO-pinnien asettelu	s. 14
<b>Kuvio 5.</b>	ZigBee-laitteen protokollapino	s. 15
<b>Kuvio 6.</b>	Taajuusalueet, kanavat ja nopeudet	s. 16
<b>Kuvio 7.</b>	Xbee S1-moduuli 1 mW antennilla	s. 17
<b>Kuvio 8.</b>	XBee Adapter kitin mukana tulevat komponentit	s. 18
<b>Kuvio 9.</b>	Xbee Adapteri valmiiksi koottu ja juotettu	s. 18
<b>Kuvio 10.</b>	X-CTU –ohjelma	s. 19
<b>Kuvio 11.</b>	Järjestelmän toimintaperiaate	s. 26
<b>Kuvio 12.</b>	SHT71-anturin kytkentä	s. 27
<b>Kuvio 13.</b>	Selitys nimille	s. 29
<b>kuvio 14.</b>	Vuokaavio ohjelmakoodin toiminnasta Arduinossa	s. 30
<b>Kuvio 15.</b>	XBee adapter kitin pinnit ja XBee-moduuli	s. 31
<b>Kuvio 16.</b>	CC2420DK Packet Sniffer	s. 32
<b>Kuvio 17.</b>	Kahdesta ZigBee paketista kuvakaappaus	s. 33
<b>Kuvio 18.</b>	Tarvittava osa ASCII-taulukosta	s. 34
<b>Kuvio 19.</b>	FTDI-kaapeli ja XBee-adapterin kytkentä	s. 35
<b>Kuvio 20.</b>	Vuokaavio Python-ohjelmakoodin toiminnasta	s. 36
<b>Kuvio 21.</b>	Taulukon tarkistus	s. 40
<b>Kuvio 22.</b>	PHP-koodin tulostama viivakaavio	s. 41



## 1 JOHDANTO

Opinnäytetyö tehtiin Vaasan ammattikorkeakoululle. Ohjaajana toimi lehtori Santiago Chávez Vega.

Nykyaikana kasvavissa määrin tulee koko ajan erilaisia IoT-laitteita markkinoille eri käyttötarkoituksilla. Työssä oli tarkoitus tutkia ja kehittää toimiva IoT-kokonaisuusprototyyppi alusta loppuun.

Tässä työssä tarkoituksena oli kehittää langaton lämpö- ja kosteusmittaus käyttäen ZigBee-verkkoa. Ajatuksena oli alunperinkin käyttää anturin datan keräämiseen jotain halpaa ja hyvää kehitysalustaa. Arduino osoittautui hyväksi vaihtoehdoksi, koska siitä löytyy valmiiksi hyviä kirjastoja, joita pystyi tässä työssä käyttämään hyödykseen. Arduino on myös samalla täysiverinen AVR-kehitysalusta.

Langattoman datan vastaanottamiseen, tietokantaan ja selainpohjaisen kuvaajan näyttämiseen oli tarkoitus käyttää tietokonetta. Tähän tehtävään sopi hyvin linux-pohjainen Raspberry Pi.

### **Vaasan ammattikorkeakoulu**

Vaasan ammattikorkeakoulu (VAMK) aloitti toimintansa väliaikaisena ammattikorkeakouluna 1.8.1996. Teknillistä oppilaitosta lukuun ottamatta oppilaitokset jakautuivat 1.8.1999 tapahtuneen vakinaistamisen yhteydessä ammattikorkeakouluun ja toisen asteen oppilaitoksiin. VAMKiin muodostettiin tekniikan ja liikenteen, liiketalouden ja matkailun sekä sosiaali- ja terveysalan toimialayksiköt. Vakinaistamisen yhteydessä VAMKin vahvuuksiksi todettiin monialaisuus, monikielisyys, kansainvälisyys ja korkea teknologinen osaaminen.

VAMKin toiminta muutettiin osakeyhtiömuotoiseksi vuoden 2010 alusta. Oy Vaasan ammattikorkeakoulu – Vasa yrkeshögskola Ab:n omistajia ovat Vaasan kaupunki, Vaasan yliopisto, Pohjanmaan liitto ja Pohjanmaan kauppakamari.

Koulu työllistää noin 210 henkilöä. Opiskelijoita Vaasan ammattikorkeakoulussa on noin 3300 seuraavilla koulutusaloilla: Sosiaali-, terveys- ja liikunta-ala, tekniikan ja liikenteen ala, yhteiskuntatieteiden, liiketalouden ja hallinnon ala. /1/

## 2 KÄYTETYT KOMPONENTIT JA TEKNOLOGIAT

### 2.1 Arduino

Arduino-mikro-ohjain on kehitysalusta, joka perustuu Atmellin AVR-mikrokontrolleriin, jota ohjelmoidaan Arduino IDE –ohjelmointiympäristöllä. Arduinon laitteisto on avointa lähdekoodia. Mikro-ohjaimien piirikaaviot ovat suoraan ladattavissa Arduinon sivuilta.

Ensimmäinen Arduino esiteltiin vuonna 2005, tarkoituksena olla halpa, helppo noviiseille ja ammattilaisille luoda laitteita, jotka toimivat heidän ympäristöissä käyttäen erilaisia sensoreita ja toimilaitteita. Arduinoista on paljon erilaisia versioita, jotka huomioivat niin isommat kuin pienemmätkin projektit sekä mahdollisia liitettäviä lisälevyjä: Esimerkkinä Arduino Ethernet Shield ja Arduino GSM Shield. Ohjelmointi tapahtuu natiivi USB-kaapelilla tai FTDI-kaapelilla, joka on kytketty tietokoneeseen.

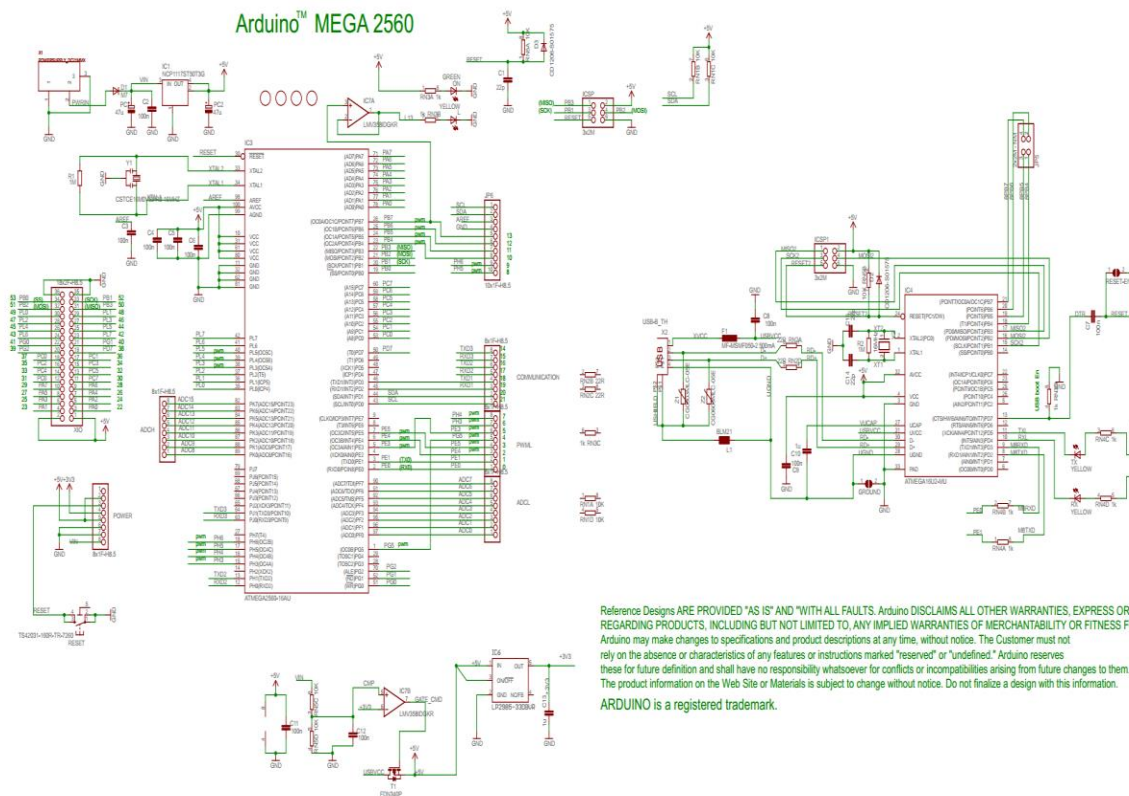
### 2.2 Arduino MEGA 2560

Tässä työssä käytettiin Arduino MEGA 2560:tä (**Kuvio 1.**), joka perustuu Atmelin Atmega 2560. Sen tärkeimmät ominaisuudet ovat:

- 54 kpl digitaalista sisään-ulosmenopinniä (joista 15 kpl niistä voidaan käyttää PWM-ulostulona)
- 16 analogista sisääntuloa
- 4 kpl UART-lähtöä/tuloa
- 16 MHz kellotaajuus
- 256 KB Flash-muistia
- USB yhteys, ulkoiselle virtalähteelle paikka, Reset-painike.
- Käyttöjännite on 5V. Kuitenkin on mahdollista syöttää ulkoiselta virtalähteeltä maksimissaan 20V jännitettä, jonka hoitaa Arduinon oma jänniteregulaattori. (**Kuvio 2.**) /2/



**Kuvio 1.** Arduino Mega 2560 /2/

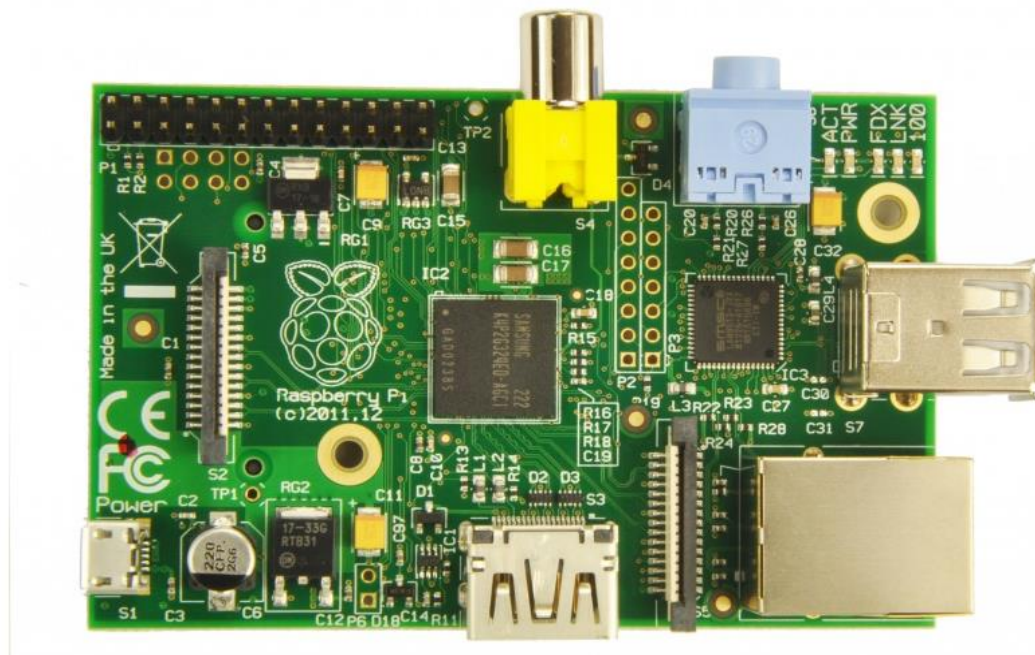


**Kuvio 2.** Arduino Mega 2560-kytkentäkaavio /2/

## 2.3 Raspberry Pi

Raspberry Pi (**Kuvio 3.**) on yhden piirilevyn pieni tietokone, jonka on kehittänyt brittiläinen Raspberry Pi Foundation. ARM-mikroprosessorilla toimiva Raspberry Pi on tarkoitettu halvaksi ja pieneksi tietokoneeksi, lähinnä ohjelmoinnin opetteluun. Käyttömahdollisuuksia on muitakin, esimerkiksi siitä saa olohuoneen media-PC:n asentamalla Kodin. /4/

Käyttöjärjestelmiä löytyy erilaisia, kuten: Raspbian, Ubuntu mate, Snappy Ubuntu Core, Windows 10 IOT Core, OSMC, OpenEelec. Yleisesti on suositeltava käyttää Raspbian käyttöjärjestelmää, se pohjautuu Linux Debianiin. Raspberry Pin mallista riippuen, käyttöjärjestelmä asennetaan SD-kortille tai MicroSD-kortille. /5/

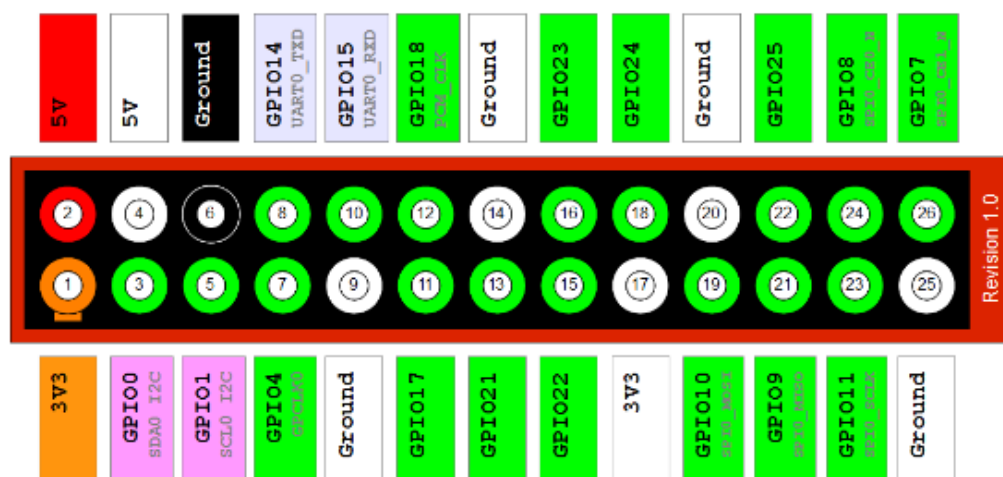


**Kuvio 3.** Raspberry Pi /4/

## 2.4 Raspberry Pi Model B

Tässä työssä käytettiin Raspberry Pi Model B:tä. Sen tärkeimmät tekniset ominaisuudet:

- Mitat: 85 mm x 56 mm x 17 mm
- Paino: 45 g
- Järjestelmäpiiri (SoC): BCM2835
- Prosessori: 700 Mhz, yksiytiminen ARM1176JZF-S
- RAM: 512 MB ja se on jaettu käyttöjärjestelmän ja näytönohjaimen kesken.
- 2 kpl USB-portteja
- HDMI
- RCA
- 10/100 Ethernet (RJ45)
- Massamuisti: SD-kortti
- Käyttöjärjestelmä: Debian (Raspbian)
- 26-pinninen GPIO-liitin (**Kuvio 4.**)

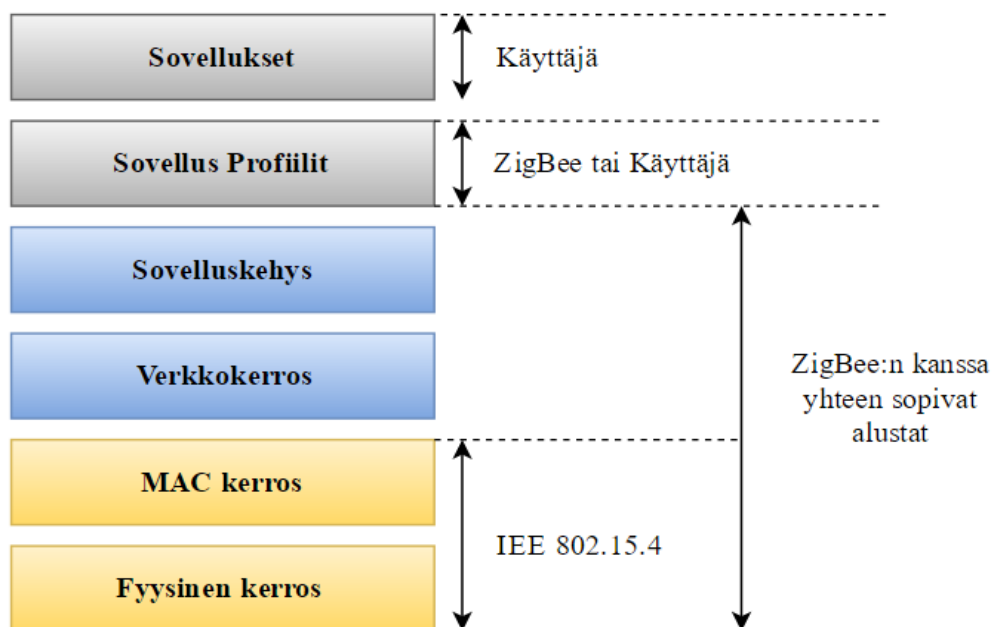


**Kuvio 4.** Raspberry Pin GPIO-pinnien asettelu /6/

## 2.5 ZIGBEE

### 2.5.1 IEEE 802.15.4-standardi

ZigBee-protokolla perustuu IEEE 802.15.4-standardiin. Se on tarkoitettu käytettävän yksinkertaisissa ja halvoissa laitteissa, joissa on lyhyt kantama, pieni tiedonsiirtonopeus ja tarve pienelle virrankulutukselle.



**Kuvio 5.** ZigBee-laitteen protokollapino /7/

Vaikka ZigBee ei kilpaile tiedonsiirtonopeudella, on sen tarkoitus olla luotettava tiedonsiirtoprotokolla ja pienen virrankulutuksen ansiosta paristotoimisten laitteiden on mahdollisuus toimia kuukausia tai vuosia yhtäjaksoisesti.

Taajuusalue	868 Mhz	915 Mhz	2.4GHz
Kanavat	1	10	16
Data nopeus	20/80 kbps	40 kbps	250 kbps

**Kuvio 6.** Taajuusalueet, kanavat ja nopeudet /7/

Zigbeeen siirtoetäisyydet XBee-tuotenimellä ovat 10 metristä 3,2 kilometriin. Yleisesti tähän vaikuttaa millä taajuusalueella toimitaan. Suuremmilla taajuuksilla päästään datansiirtonopeuksissa paljon korkeammalle, mutta samalla siirtoetäisyydet laskevat. Lähettimen teholla on myös etäisyyteen ja datan siirtonopeuteen suuri vaikutus.

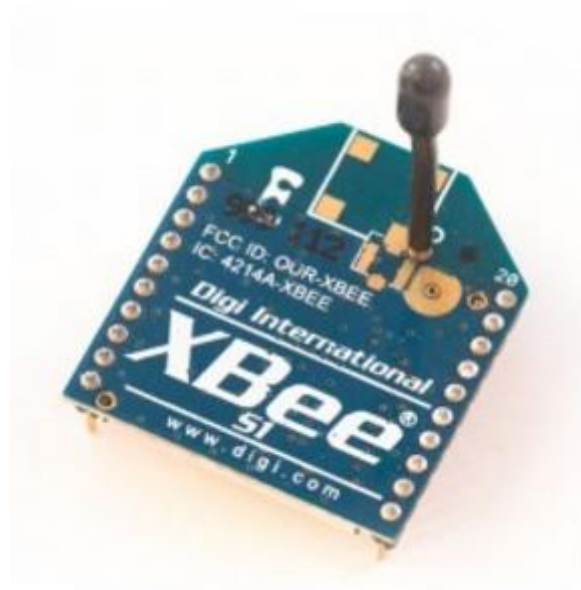
Erilaiset ympäristön esteet vaikuttavat signaalin vahvuuteen negatiivisesti, kuten metallielementit ja paksut betoniseinät.

### 2.5.2 XBee

XBee on Digi International –yhtiön kehittämä tuotemerkki radio-moduuleille, jotka perustuvat IEEE 802.15.4-standardiin. XBee-tuoteperheeseen kuuluu mallit XBee (**Kuvio 7.**), Xbee-PRO, XBee-DigiMesh-2.4, XBee-868LP, XBee-PRO-XSC, XBee-WIFI, XBee-PRO-868.

Yleisemmin kaikissa PRO-malleissa on suurempi lähetysteho, mikä tarkoittaa verrattuna ei PRO-malleihin, että siirtoetäisyys on parempi.





**Kuvio 7.** Xbee S1-moduuli 1 mW antennilla /8/

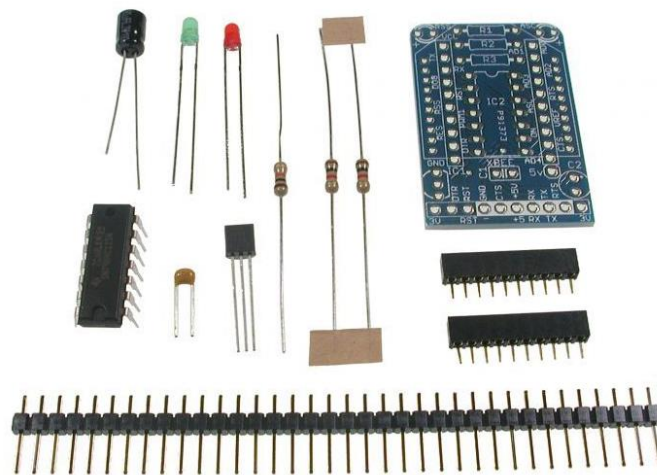
XBee-moduulin tärkeimmät ominaisuudet:

- toimintataajuus: ISM 2.4 GHz
- kantama sisätiloissa: 30 metriä, ulkotiloissa: 90 metriä (näkyvä yhteys)
- 1mW lähetysteho (0 dBm)
- vastaanottimen herkkyys -92 dBm (1% pakettivirhe suhde)
- virrankulutus: 3.3 V 45 mA siirtotilassa, 50 mA toimettona tai vastaanottotilassa. /8/

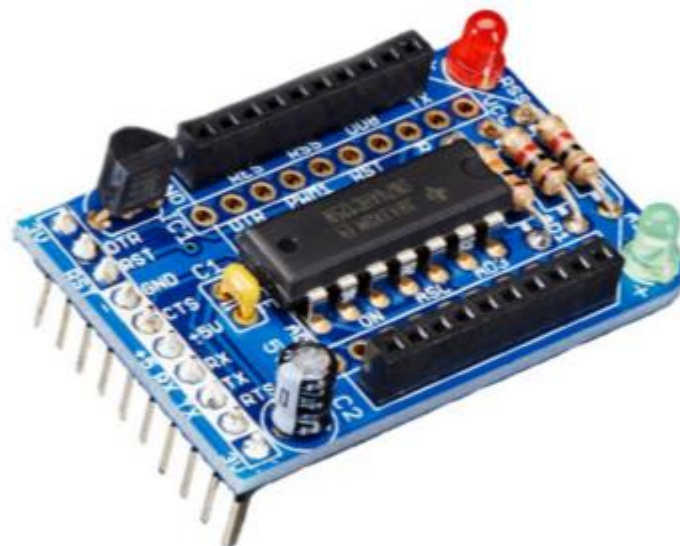
### 2.5.3 XBee Adapter kit

XBee-moduulille suunniteltu adapterisarja, (**Kuvio 8.**) jolla voi helpottaa itse XBee-moduulin kiinnitettävyyttä. Adapterisarja on erityisesti suunniteltu käytettäväksi FTDI-kaapelin kanssa, jolloin sen voi suoraan yhdistää USB:llä tietokoneeseen. Näin saadaan helppo liitettävyys sarjaliikenteelle. (**Kuvio 9.**)

Tärkeänä huomion on, että XBee Adapter kit sisältää 3.3V jänniteregulaattorin, jolla taataan, että jännitemäärä on oikea, sekä tasonsiirtopiiri, jolla taataan, että adapter on turvallista kytkeä eri Arduino-laitteisiin joiden jännite on 5V. /9/



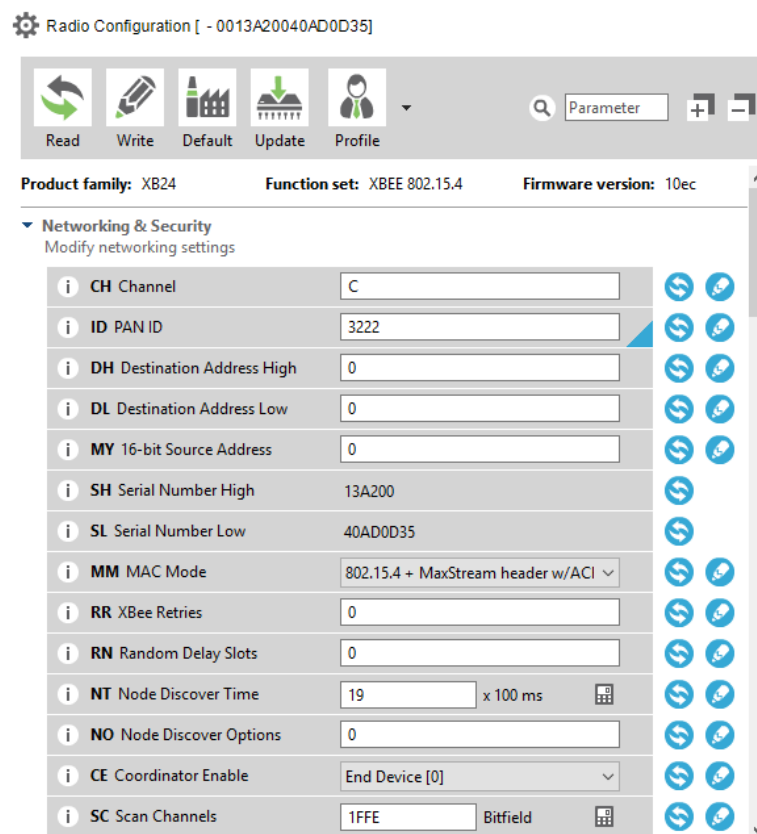
**Kuvio 8.** XBee Adapter kitin mukana tulevat komponentit /9/



**Kuvio 9.** Xbee Adapteri valmiiksi koottu ja juotettu. /9/

## 2.5.4 XBee-moduulin konfigurointi

Konfigurointi on helppoa hoitaa Digi International –yhtiön omalla heidän sivuilta vapaasti ladattavana olevalla X-CTU –ohjelmalla (**Kuvio 10.**). Konfigurointi tapahtuu tässä tapauksessa FTDI USB-kaapelilla tietokoneeseen, jossa on kiinni XBee-moduuli.



**Kuvio 10.** X-CTU –ohjelma

Konfigurointi on myös mahdollista tehdä jollain pääte-emulaattorilla, kuten PuTTYlla. Silloin on tiedettävä komennot, joilla keskustellaan XBee-moduulin kanssa. Esimerkiksi, jos yhteys on toimiva, kirjoitetaan PuTTYn –komentoriville +++ jolloin tulee vastaus OK.

## 3 KÄYTETYT OHJELMOINTIKIELET

### 3.1 C-kieli

C-ohjelmointikieli on kehitetty 1970-luvun alussa Unix-käyttöjärjestelmille. Sen on alunperin kehittänyt Dennis Ritchie AT&T Bell Labs -yhtiössä. C-kielestä on sittemmin tullut yksi laajimmin käytetyistä ohjelmointikielistä kaiken aikaa. C-kääntäjä löytyy saatavilla eri valmistajien suuresta osasta nykyisiä tietokonearkkitehtuureja ja käyttöjärjestelmiä.

C-kielestä on sittemmin tullut useita eri versioita. Tämän takia ANSI (the American National Standards Institute) on standardisoinut C-kielen, joka on nimeltään ANSI Standard C. ISO (International Organization for Standardization) hyväksyi standardin vuoden 1989 lopulla nimellä ISO/IEC 9899-1990. /10/

Huolimatta C-kielen matalan tason ominaisuuksista, kieli on suunniteltu järjestelmäriippumattomaan ohjelmointiin. Kun C-kieli on kirjoitettu standardiin mukaan, on koodin liikuteltavuus eri alustojen ja käyttöjärjestelmien välillä helppoa, muuttaen lähdekoodista vain vähän.

C-kieli on suoraan vaikuttanut useisiin kieliin jotka on kehitetty sen jälkeen. Näitä kieliä ovat: C#, D, Go, Java, JavaScript, Limbo, LPC, PERL, PHP ja Python. Eniten C-kieli näihin on tehnyt vaikutusta sen tunnistettavalla syntaksilla. /11/

Esimerkki C-kielen syntaksista. Ohjelma tulostaa tekstin Hello world!.

```
/* Tämä on kommentointi-osio, joka ei vaikuta koodiin millään tavalla */  
  
#include <stdio.h>  
  
main()  
{  
    printf("Hello World!");  
}
```

### 3.2 C++-kieli

C++-kieli pohjautuu C-kieleen ja sen on kehittänyt Bjarne Stroustrup 1980-luvulla. C++-kielen standardi on ISO/IEC 14882:1998 ja se vahvistettiin vuonna 1998. Nykyään uusin tarkistettu painos C++-kielestä on C++14 ja sen standardi on ISO/IEC 14882:2014, joka on julkistettu vuonna 2014.

C++-kieltä käyttävät sadattuhannet ohjelmoijat ympäri maailman ja se on käytössä hyvin erilaisilla sovellusalueilla. C-kieli on valittu C++-kielen pohjaksi seuraavista syistä: Se on monipuolinen, niukksanainen, matala-tasoinen, riittävä useimpiin järjestelmän tehtäviin, toimii joka puolella, toimii kaikkien kanssa, sopii hyvin UNIX-ohjelmointiympäristöön.

C++-kielen yleisin käyttötarkoitus ohjelmointikielenä on painottuminen järjestelmien ohjelmointiin. Tässä tarkoituksessa sen on tarkoitus olla parempi kuin C-kieli, tukea data-abstraktiota, tukea olio-ohjelmointia ja geneeristä ohjelmointia. /12/

Esimerkki C++-kielen syntaksista. Ohjelma tulostaa tekstin Hello world!.

```
#include <iostream>

int main()
{
    std::cout << " Hello World! ";
}
```

### 3.3 Python-kieli

Python-ohjelmointikieli on korkean tason ohjelmointikieli, jonka on kehittänyt Guido van Rossum 1980-luvun lopulla. Python-kielen on tarkoitus olla helppolukuinen, yksinkertainen ja mahdollistaa ohjelmoijille kehittää nopeasti sovelluksia. Kaikki korkean tason ohjelmointikielet, kuten C, C++, Java, PHP, Ruby ja Visual Basic ovat hyvin saman tyyliä. Näiden pääasiallinen ero on syntaksissa, saatavilla olevissa kirjastoissa ja kuinka kirjastoihin päästään käsiksi.

Yleisesti sanotaan, että Python-kieli on hyvä kieli aloittaa ohjelmointi, koska yksi Python-kielen tärkeimmistä ominaisuuksista on yksinkertaisuus. Yleisissä ohjelmissa usein Python-kieli vaatii useita rivejä vähemmän koodia suorittaa samanlainen asia, kuten C-kielessä.

Yleisesti Python-kieltä voidaan käyttää hoitamaan monenlaisia tehtäviä, kuten työpöytäsovelluksia, tietokantasovelluksia, verkko-ohjelmointia ja peli-ohjelmointia. Yksi Python-kielen vahvuuksista on järjestelmäriippumattomuus. Järjestelmäriippumattomuudella tarkoitetaan, kun koodi on kirjoitettu, esimerkiksi Windows käyttöjärjestelmälle, se toimii myös hyvin MAC OS -ja Linux – käyttöjärjestelmissä ilman, että itse Python-koodiin tehdään mitään muutoksia.

/13/

Esimerkki Python-kielen syntaksista. Ohjelma tulostaa tekstin Hello world!.

```
print ("Hello World")
```

### 3.4 PHP-kieli

Nykyisin tunnettu skriptikieli PHP on PHP/FI seuraaja. PHP/FI on luonut Rasmus Lerdorf vuonna 1994. Alunperin PHP tarkoitti Personal Home Page Tools, suomennettuna henkilökohtaisten kotisivujen työkalut. Nykyään PHP tunnetaan nimestä Hypertext Preprocessor

Kuitenkin nykyään tunnettu PHP-kieli on Andi Gutmans:n ja Zeev Suraskin käsialaa. PHP/FI 2.0:n lähdekoodi kirjoitettiin lähes kokonaan uusiksi ja se julkaistiin vuonna 1998 nimellä PHP 3. Tämänhetkinen vakaa versio PHP:sta on 5.6.19. julkaistu 2016 Maaliskuu.

PHP-kieli on yleiskäyttöinen skriptikieli ja se on tarkoitettu web-sovelluskehitykseen. Tavallisesti sitä käytetään upotettuna HTML-sivujen sisälle. PHP-koodin aloitustunniste on `<?php` ja lopetustunniste `?>`. Huomionarvoista on myös, että PHP-koodi on vain olemassa palvelimella. Satunnainen asiakas, joka käy web-sivulla ei koodia näe. /14/

Esimerkki PHP-kielen syntaksista upotettuna HTML-sivujen sisälle. Ohjelma tulostaa tekstin Hello world!.

```
<html>
  <head>
  </head>
  <body>

  <?php echo '<p>Hello World!</p>'; ?>

  </body>
</html>
```

### 3.5 SQL-kieli

SQL on standardisoitu kyselykieli, jolla päästään käsiksi relaatiotietokantoihin ja voidaan hallita niitä.

RDBMS eli (Relational Database Management System) relaatiotietokantojen hallintajärjestelmä, kuten MySQL, Oracle, PostgreSQL, SQLite, Access.

Esimerkki SQL-kyselyn syntaksista. Seuraava lause valitsee kaikki tiedot taulusta ”EsimerkkiTaulu” ja näyttää ne.

```
SELECT * FROM EsimerkkiTaulu;
```



## 4 JÄRJESTELMÄN TOIMINTA JA TOTEUTUS

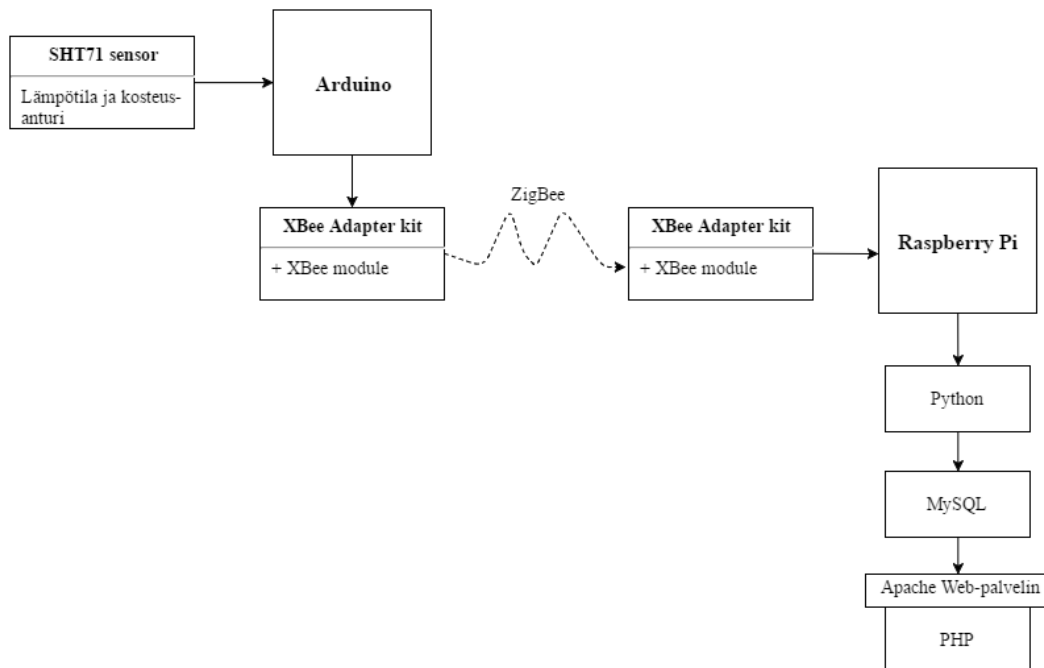
Lämpötilan ja kosteuden mittaus tapahtuu Arduinossa SHT71-anturilla, jossa Arduino lukee ohjelmakoodissa tietyin väliajoin (tässä tapauksessa 50 sekunnin välein) lämpötilan, kosteuden ja laskee näistä arvoista kastepisteen. Samalla, kun arvot luetaan, ne kirjoitetaan myös sarjaporttiin.

Arduinon on kytketty ZigBee-moduuli sarjaporttiin, jolla lähetetään anturilta saadut arvot langattomasti eteenpäin.

Niin ikään Raspberry Pi:ssä on kytketty ZigBee-moduuli USB:llä kiinni, jolla voidaan vastaanottaa Arduinon lähettämät arvot. Raspberry Pi:ssä ajetaan Python-koodia, joka lukee koko ajan USB:tä. Kun saadaan arvot: lämpötila, kosteus, kastepiste ja päivämäärä. Nämä arvot kirjoitetaan SQL-tietokantaan talteen. Tämän jälkeen Python-koodi jää odottamaan taas uusia arvoja.

Raspberry Pi:ssä on myös Apache-palvelinohjelma käytössä, joka pyörittää PHP-ohjelmaa. Ohjelma kirjautuu MySQL-tietokantaan, joka lukee jatkuvasti uusia lämpötiloja, kosteutta, kastepistettä ja piirtää arvot viivakaavion. Arvoja on näin helppo tarkastella viivakaavio koordinaatistosta.

Kuviossa 11 Esitellään järjestelmän toimintaperiaatetta.



**Kuvio 11.** Järjestelmän toimintaperiaate.

#### 4.1 Lämpötilan ja kosteuden mittaaminen

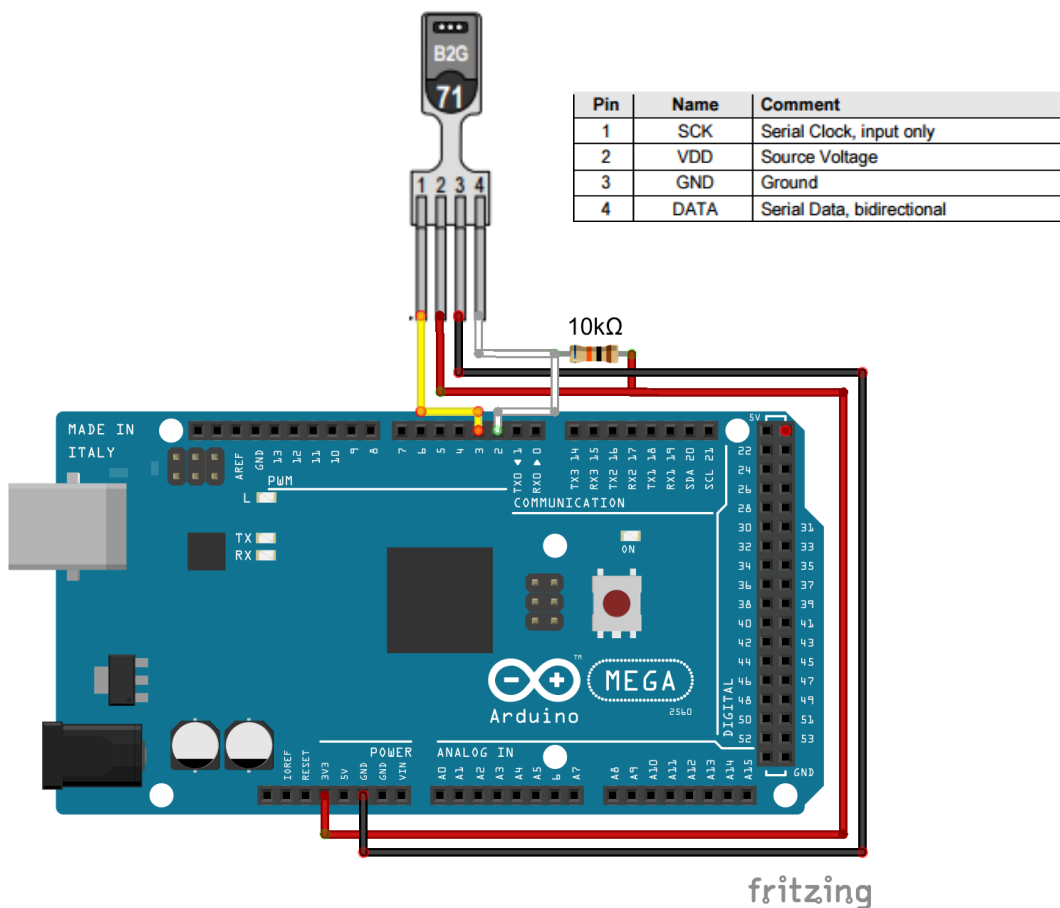
Anturina käytetään Sensirionin SHT71-anturia. SHT71 hyödyntää digitaalista ulostuloa. Tämän mahdollistaa 14-bittinen analogisesta digitaaliseen muunnin ja sarjaliitännäpiiri, johon molemmat sensorit (lämpötila ja kosteus) on kytketty.

Anturissa on neljä eri pinniä: SCK (Serial Clock, input only), VDD (Source Voltage), GND (Ground) ja DATA (Serial Data, bidirectional). Käyttöjännite anturissa voi pienimmillään olla 2.4V ja suurimmillaan 5.5V. Tyypillinen jännite on 3.3V, joka on tässä työssä käytössä. /16/

#### 4.1.1 SHT71-anturin kytkentä

Anturin kytkentä Arduinolle (**Kuvio 12.**) tapahtuu anturilta katsottuna: 1-pinni (SCK) kytketään Arduinon Digital 3:n, 2-pinni (VDD) kytketään Arduinon 3.3 Voltin tuloon, 3-pinni (GND) kytketään Arduinon maaliittimeen, 4-pinni (DATA) kytketään Arduinon Digital 2:n

Anturin 4-pinnin ja 2-pinnin välissä käytetään ylösvetovastusta (10k $\Omega$ ), joka varmistaa, että anturin tuloliitäntä on ylätilassa silloin, kun liitäntään ei ole kytketty muuta signaalia. /17/



**Kuvio 12.** SHT71-anturin kytkentä

#### 4.1.2 Arvojen lukeminen SHT71-anturilta

Arduinossa olevan ohjelmakoodin on tarkoitus olla hyvin suoraviivainen. Ohjelmakoodi perustuu Arduinon sensirion-kirjastoon ja sen esimerkkikoodiin.

Tämän ohjelmakoodin alussa määritellään anturin pinnit arduinon. Pinnit ovat dataPin = 2 ja sclkPin = 3. Näiden tyypit ovat const uint8\_t. Määritellään muuttujat temperature, humidity, ja dewpoint float -tyyppisiksi. Aikavälin seuranta tapahtuu muuttujilla trhMillis ja blinkMillis. Nämä alustetaan nolaksi. Näiden datatyyppi on unsigned long.

void setup() funktiossa kerrotaan, että halutaan käyttää sarjaliikennettä nopeudella 9600 bittiä sekunnissa Serial.begin(9600) ja odotetaan 11 millisekuntia ennen ensimmäistä komentoa delay(15).

Silmukkaan mentäessä funktiolla unsigned long curMillis = millis() saadaan nykyinen aika. Tätä seuraa if lause if (curMillis - trhMillis >= TRHSTEP), jolla tarkistetaan onko aika tehdä uusi mittaus, jos ehto on tosi mennään logData() funktioon. Jos ehto ei toteudu, odotetaan 11 millisekuntia ja palataan silmukkaan.

Funktiossa void logData() yksinkertaisesti käytetään Serial.print() funktiota, jolla tulostetaan arvot sarjaporttiin. XBee-moduuli on kytketty RX0 ja TX0 pinneihin, mikä mahdollistaa sarjakommunikaation moduulille. Arvojen tulostusta on myös mahdollista seurata suoraan tietokoneelta, jos Arduino on kytkettynä USB-johdolla tietokoneeseen. Arvojen tulostus käy seuraavasti:

```
Serial.print(temperature);  
  
Serial.print(" \t");  
  
Serial.print(humidity);  
  
Serial.print(" \t");  
  
Serial.print(dewpoint);
```

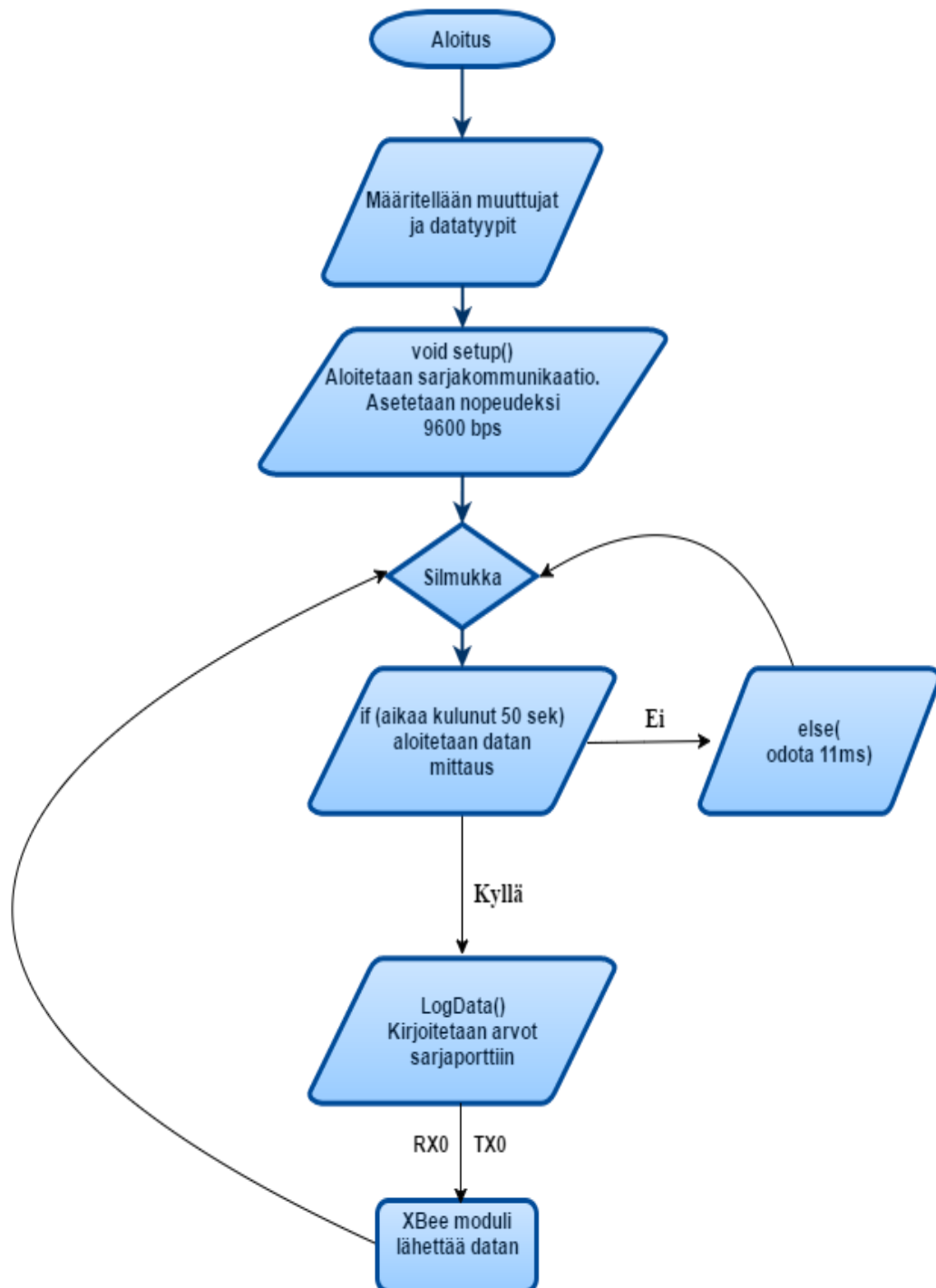
Syy miksi käytetään arvojen tulostusten välissä `Serial.print(" \t");` on se, että saadaan arvot pilkottua osiin SQL-tietokantaan kirjoittamista varten. Siitä lisää Python-osiossa.

Huomiotavaa on, että anturilta saamat arvot lasketaan suoraan Sensirion kirjastossa anturilta tulevalta raakadatalta. Tässä esimerkki Sensirion.cpp tiedostosta kastepisteen laskennasta: /18/

```
float Sensirion::calcDewpoint(float humi, float temp) {  
  
    float k;  
  
    k = log(humi/100) + (17.62 * temp) / (243.12 + temp);  
  
    return 243.12 * k / (17.62 - k);  
  
}
```

Humi	=	Suhteellinen kosteus
Log	=	Luonnollinen logaritmi
Temp	=	Lämpötila

**Kuvio 13.** Selitys nimille

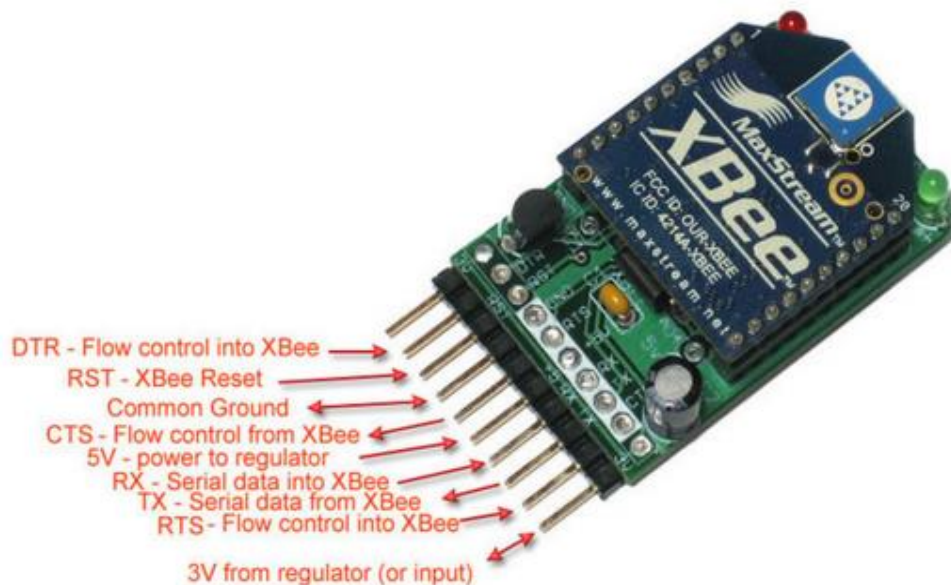


**Kuvio 14.** Vuokaavio ohjelmakoodin toiminnasta Arduinossa

## 4.2 ZigBee-verkon toiminta ja XBee-moduuli

### 4.2.1 XBee-moduulin kytkentä XBee-adapter kitin avulla

XBee-moduuli, joka on kytketty kiinni XBee adapter kittiin (**Kuvio 15.**) kytketään arduinon seuraavasti: Arduinon 5 voltin linjasta XBee adapter kitin 5v linjaan, joka reguloidaan XBee adapter kitin jänniteregulaattorilla 3.3 Volttiin. Arduinon maaliittimestä XBee adapter kitin maaliittimeen.



**Kuvio 15.** XBee adapter kitin pinnit ja XBee-moduuli /9/

Liikenteessä käytetään sarjaliikennettä, joten Arduinon RX-liittimestä XBee adapter kitin TX-liittimeen ja Arduinon TX-liittimestä XBee adapter kitin RX-liittimeen.

Muita XBee adapter kitin liittimiä tässä projektissa ei tarvita toimivan sarjaliikenteen toteuttamiseksi.

#### 4.2.2 IEEE 802.15.4 liikenteen tutkiminen packet snifferillä

IEEE 802.15.4 liikenteen tutkimisessa käytettiin Chipconin valmistamaa CC2420DK Packet Snifferiä (**Kuvio 16.**). Tässä osassa on tarkoitus, että voidaan todentaa data mitä ZigBeessä kulkee ja Arduinin lähettämä ohjelmakoodi ZigBeellä toimii oikein.

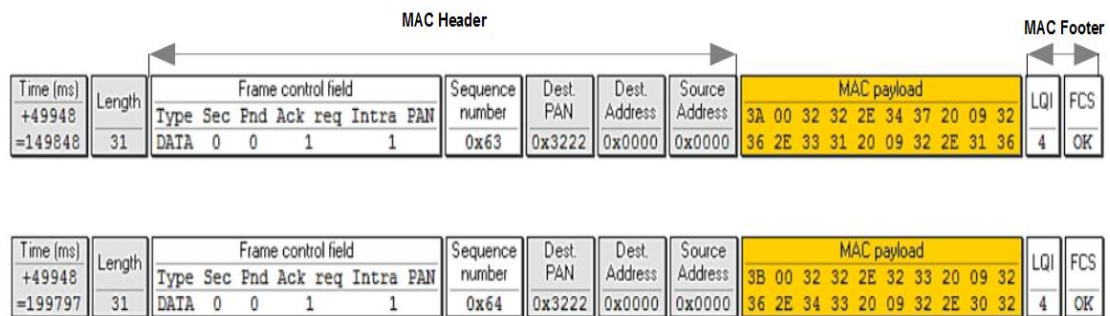


**Kuvio 16.** CC2420DK Packet Sniffer

CC2420DK Packet Sniffer on kytketty tietokoneeseen kiinni USB-kaapelilla. Tietokoneessa on asennettu Chipconin oma ohjelma, jolla voidaan tutkia IEEE 802.15.4-liikenteen paketteja.



Mittaus on otettu kanavalla 12 ja taajuudella 2.410 Ghz. Aiempana kuva kahdesta paketista, jotka on Arduinon XBee-moduulin lähettämiä.



**Kuvio 17.** Kahdesta ZigBee-paketista kuvakaappaus

Kuvan kohdassa, MAC payload –osassa, voidaan lähetetyt arvot todentaa oikeiksi. Ensimmäinen arvo on 3A ja se on nouseva-arvo, eli jokaisen paketin jälkeen se nousee yhden ylöspäin, kuten alemmasta paketista nähdään, se on 3B. Seuraava arvo 00 ja tässä tapauksessa se ei muutu.

Ensimmäisen MAC payloadin arvot, kahta ensimmäistä lukuunottamatta ovat: 32 32 2E 34 37 20 09 32 36 2E 33 31 20 09 32 2E 31 36. Nämä ovat heksadesimaaleja. Heksadesimaalijärjestelmä on kantalukujärjestelmä ja sen kantaluku on 16. Yksi 16-kantaisen järjestelmän merkki vastaa suoraan binäärijärjestelmän neljää peräkkäistä bittiä. Näin esimerkiksi 8-bittisen tavun arvo voidaan ilmaista kahden merkin pituisella heksadesimaaliluvulla. /19/

Hyvänä apuna on merkkien käänöksessä käyttää ASCII-taulukkoa. (**Kuvio 18.**)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>;</b>	91	5B	133	&#91;	<b>[</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>

**Kuvio 18.** Tarvittava osa ASCII-taulukosta /20/

Nämä edellä mainitut heksadesimaalit käännetään suoraan merkeiksi ASCII-taulukon avulla. Tuloksena saadaan 22.47 ”välilyönti” ”TAB” 26.31 ”välilyönti” ”TAB” 2.16

Luvut on tarkistettu Arduinossa eli lähetyspäässä sekä Raspberry Pi:ssä, joka on vastaanottavassa päässä. Luvut vastaavat toisiaan ja näin voidaan todeta, että luvut ovat oikein ja oikeassa muodossa.

### 4.3 Arvojen vastaanotto ja käsittely Raspberry Pillä

#### 4.3.1 XBee-moduulin kytkentä XBee-adapter kitin avulla

Kytkenän periaate on sama kuin Arduinossa, mutta Raspberry Pin USB-porttien ansiosta voidaan niitä käyttää tässä hyödyksi.

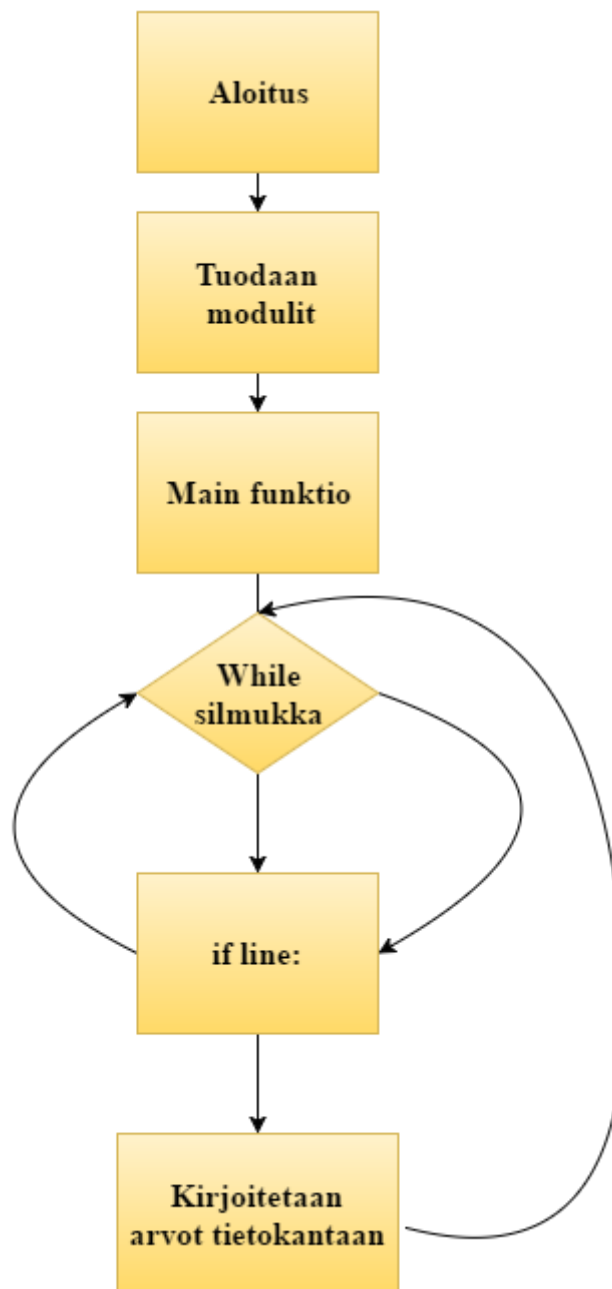


**Kuvio 19.** FTDI-kaapeli ja XBee-adapterin kytkentä /21/

Tässä kytkentä tehdään XBee adapter kittiä apuna käyttäen ja se liitetään kuvion 19, mukaisella tavalla. USB-liitin kytketään toiseen vapaana olevaan paikkaan Raspberry Piissä.

### 4.3.2 Tulevan datan käsittely Python-koodilla

Python ohjelmakoodin tehtävä tässä osassa on lukea tuleva data sarjaporttin /dev/ttyUSB0:n kautta, johon XBee-moduuli on kytketty kiinni. Tätä sarjaporttia lueataan jatkuvasti. Tämän jälkeen saatu data kirjoitetaan suoraan MySQL-tietokantaan.



**Kuvio 20.** Vuokaavio Python-ohjelmakoodin toiminnasta

Ohjelmakoodin alussa tuodaan moduulit mitä tässä ohjelmassa tarvitaan ja kerrotaan ne ohjelmakoodille mitkä ovat tuotuna muodossa `import moduuli`. Ennen ohjelmakoodiin tuontia ne on asennettava käyttöjärjestelmään. Tässä työssä linuxissa käytetään eri Python-moduulien asennuksessa pip-paketin hallintajärjestelmää.

Asennus tapahtuu suoraan komentorivillä. Esimerkkinä Pythonin MySQL-moduulin asennus tapahtuu komennolla: `pip install MySQL-python`.

Main funktiossa määritellään sarjaportti käyttöön `/dev/ttyUSB0` paikkaan.

```

serial = serial.Serial()           #määritellään serial funktio
serial.port = '/dev/ttyUSB0'       #kerrotaan mikä sarjaportti
serial.baudrate = 9600             #asetetaan nopeus
serial.timeout = 1                 #timeout aika 1 sekunti
serial.open()                      #avataan serial

```

Tämän jälkeen MySQL-tietokannan muuttujat, kirjautumistiedot ja tietokanta.  
*/22/*

```

db = MySQLdb.connect(host="localhost", user="root", passwd="5522",
db="temp_database")

cur = db.cursor()

```

Tehdääns `sys.stdin`:stä estoton tiedosto

```
fcntl.fcntl(sys.stdin, fcntl.F_SETFL, os.O_NONBLOCK)
```

While silmukkaan mentäessä määritellään datan pilkkominen osiin.

```
pieces = line.split("\t")
```

Määritellään muuttuja line ja dekodaus UTF-8 merkistöön.

```
line = serial.readline().decode('utf-8')
```

If lauseessa ehto toteutuu, kun sarjaportti vastaanottaa Arduinon lähettämän lämpötilan, kosteuden ja kastepisteen. Jos ei tule dataa, ohjelma lukee sarjaporttia niin kauan, että ehto toteutuu.

Ehdon toteuduttua if lauseessa, määritellään aikaleima, jossa on vuosi, kuukausi, tunti, minuutti ja sekunti. Kirjoitetaan vastaanotetut arvot, jota voidaan tarkastella konsolista.

```
datetimeWrite=(time.strftime("%Y-%m-%d")+
time.strftime("%H:%M:%S"))
```

```
print line
```

Tässä kohdassa kirjoitetaan tempLog tietokantaan: aikaleima, lämpötila, kosteus ja kastepiste. Kaikki Arduinolta tulevat arvot kirjoitetaan paloissa, jotta ne saadaan oikeisiin kohtiin tietokannassa. Ne on jaettu toisistaan \t merkeillä.

```
sql="""INSERT INTO tempLog (datetime,temperature,humidity,dewpoint) VAL-
UES (%s,%s,%s,%s)""", (datetimeWrite,pieces[0],pieces[1],pieces[2])
```

```
cur.execute(*sql) #Suoritetaan kaikki mitä yläpuolella määriteltä
```

```
db.commit() #Tehdään kirjoitus tietokantaan
```

Arduinolta tulevien arvojen tietokantaan kirjoituksen jälkeen, palataan odottamaan uutta dataa, joka tulee noin 50 sekuntia edellisen jälkeen.

### 4.3.3 LAMP eli Linux, Apache, MySQL ja PHP

Apache web-palvelimen tarkoitus tässä työssä on mahdollistaa arvojen lukeminen php tiedostosta paikallisesti. Apachen asentaminen käy seuraavalla komennolla, joka kirjoitetaan komentoriville: `sudo apt-get install apache`.

Apache web-palvelimen testaus tapahtuu Raspberry Pi:ssä itsessään selaimessa osoitteessa <http://localhost/> ja toiselta tietokoneelta, joka on samassa verkossa, on käytettävä Raspberry Pi:n omaa paikallista ip-osoitetta.

Näihin osoitteisiin mentäessä sivu ilmoittaa ”It works”, jonka jälkeen voi todeta, että palvelin toimii. Apachen oletushakemistopolku on `/var/www/html/`

PHP:n asentaminen käy samalla tavalla komentorivin kautta. PHP on asennettu tässä seuraavalla tavalla: `sudo apt-get install php5 libapache2-mod-php5`.

MySQL-tietokanta ja siihen tarvittavat muut paketit on asennettu seuraavalla komennolla: `sudo apt-get install mysql-server php5-mysql`. MySQL-asennuksessa myös kysytään root käyttäjän salasanaa, joka on hyvä pistää tässä vaiheessa muistiin. /23/

Sisään kirjautuminen MySQL-tietokantaan tapahtuu seuraavasti komentorivillä: `mysql -uroot -pSalasana`. Kun ollaan päästy kirjautumaan MySQL:n luodaan tietokanta, jonka nimi on `temp_database` seuraavalla komennolla: `create database temp_database;` Tietokannan luonnin jälkeen ilmoitetaan myös, että haluamme käyttää `temp_database` tietokantaa komennolla: `USE temp_database;`

Taulukon luonti tietokantaan tapahtuu seuraavasti: `CREATE TABLE tempLog(datetime DATETIME NOT NULL, temperature FLOAT(5,2) NOT NULL, humidity FLOAT(5,2) NOT NULL, dewpoint FLOAT(5,2) NOT NULL);`

Tästä voidaan vielä tehdä tarkistus komennolla: `DESCRIBE tempLog;`, jolla voidaan tarkistaa, että taulukko on oikein. (**Kuvio 21.**)

Field	Type	Null	Key	Default	Extra
datetime	datetime	NO		NULL	
temperature	float(5,2)	NO		NULL	
humidity	float(5,2)	NO		NULL	
dewpoint	float(5,2)	NO		NULL	

**Kuvio 21.** Taulukon tarkistus

index.php-tiedoston tehtävä tässä on lukea SQL-tietokannasta aikaleima, lämpötila, kosteus ja kastepiste 5 sekunnin välein. Arvojen visualisoinnista vastaa Googlen API (Line Chart), johon myös tämä koodi perustuu täysin. PHP-koodi esiteltynä kokonaisuudessaan: /24/

```

<html>
<head>
<title>Temperature, Humidity & Dewpoint</title>
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
var data = google.visualization.arrayToDataTable([
['datetime', 'temperature °C', 'humidity %', 'dewpoint'],
<?php
header( "refresh:5;url=indexp.php" );
$con = mysqli_connect("localhost", "root", "5522", "temp_database");

$query = "SELECT * FROM tempLog";
$result = mysqli_query($con, $query);

mysqli_close($con);

while ($row = mysqli_fetch_array($result))
{
    $datetime = $row['datetime'];
    $temperature = $row['temperature'];
    $humidity = $row['humidity'];
    $dewpoint = $row['dewpoint'];
    echo "['$datetime', $temperature,$humidity,$dewpoint],";
}
?>
]);

var options = {
title: 'Temperature, Humidity & Dewpoint',
vAxis: { title: "Values" }
};

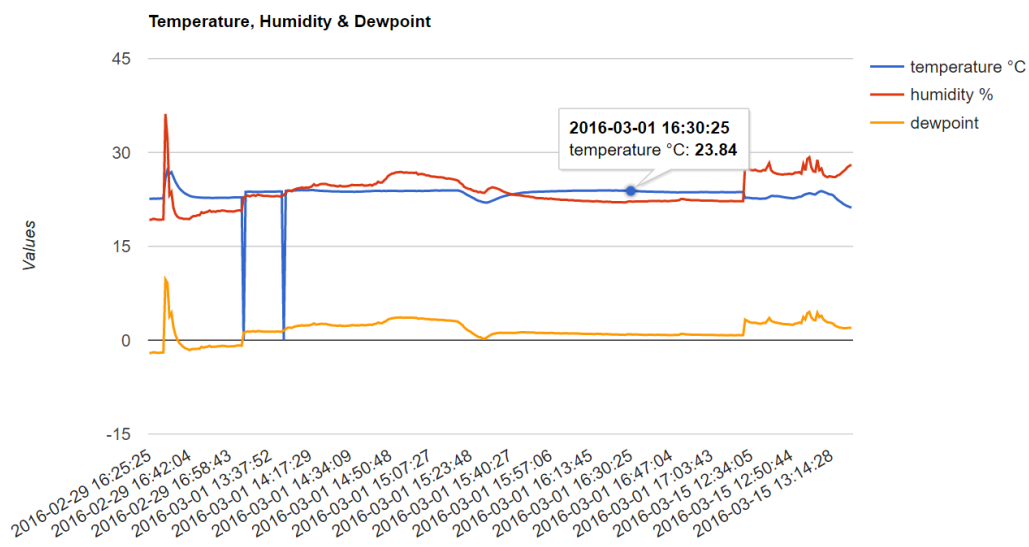
var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
chart.draw(data, options);
}
</script>
</head>
<body>
<div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```



PHP-koodi on sijoitettuna hakemistopolkuun `/var/www/html/`, joka on Apache web-palvelimen oletuskansio. Näin voidaan paikallisesta verkosta tarkastella PHP-koodin tulostamaa viivakaaviota.

PHP-koodin tulostama viivakaavio (**Kuvio 22.**), jossa piirtyvät lämpötila, kosteus ja kastepiste. Viivakaavion alapuolella näkyvät päivämäärä ja kellonaika milloin mittaus on tehty. Hiiren osoitinta vetämällä johonkin kohtaan viivakaaviossa kuten tässä kuvassa lämpötilan viivaan, näkyvät silloin mitattu lämpötila, päivämäärä ja kellonaika.



**Kuvio 22.** PHP-koodin tulostama viivakaavio

Huomiona (**Kuvio 22.**) kaksi lämpöarvoa, jotka näyttävät nolaa. Tämä johtuu testaus-vaiheesta tapahtuneesta Python-koodin uudelleen ajosta.

## 5 TESTAUS

### 5.1 Testilaitteisto

Testilaitteistoon pääasiallisesti kuuluivat Raspberry Pi Model B, 2 kappaletta XBee S1-moduuleita, 2 kappaletta XBee adapter kittejä, Arduino Mega 2560, SHT71-anturi. Arduino Megassa oli käytössä 9 V 1,0 A DC-virtalähde. Raspberry Pi model B:ssä 5 V 1,2 A DC-virtalähde.

ZigBee yhteyden testauksessa oli aluksi PC ja testaus tapahtui Arduinon PC:n välillä.

### 5.2 Testaus ja ilmentyvät ongelmat

ZigBee-yhteyden kantavuuden testaus Arduinon ja Raspberry Pin välillä tapahtui Technobothnian laboratoriossa sisätiloissa. Etäisyyttä moduleitten välillä oli noin 30 metriä ja välissä yksi ohut seinä. Tässä testissä kommunikaatiossa moduuleiden välillä ei ollut ongelmia ja testissä ei tullut minkäänlaisia ongelmia kahden tunnin testijakson aikana

Valmistajan Digi International, ilmoittamaksi kantamaksi luvataan 90 metrin etäisyys. Tämä vaatii ihanteellisen ja häiriöttömän ympäristön, sekä näköyhteyden moduleitten välillä. Digi International lupaa hieman kalliimman XBee-Pro-moduulin kantaman, jopa pystyvän yhden kilometrin etäisyyteen moduuleiden välisellä näköyhteydellä ja ihanteellisilla ympäristöolosuhteilla. /25/

Python-ohjelmakoodin käynnistämisesssä tai uudelleen käynnistämisesssä ensimmäisten arvojen lähetyksessä lämpötilan ensimmäinen arvo kirjoitettaessa tietokantaan välillä antaa luvun 0, joka näkyy kaksi kertaa. (**Kuvio 22.**) Tämä ei kuitenkaan ole suuri ongelma, koska ohjelmakoodin on tarkoitus pyöriä jatkuvasti. Kuitenkin testausvaiheessa tämä on hyvä ottaa huomioon.

## 6 YHTEENVETO

Opinnäytetyön tuloksena saatiin toimiva kokonaisuus RF-tekniikalla toteutettuna. ZigBee-tekniikka tähän työhön käytettynä ei ehkä ole paras mahdollinen sen lyhyen kantaman takia.

Tulevaisuutta ajatellen kokonaisuutta tulisi kehittää paremmaksi, lähtien mikrokontrollerista. Arduino Mega 2650 on tarpeettoman monipuolinen tähän projektiin. Parempana vaihtoehtona kävisi Arduino Mini tai Nano. Kuitenkin parhaaseen lopputulokseen pääsee sillä, että suunniteltaisiin mikrokontrolleri juurikin tähän tehtävään ja se tekee vain tarvittavan. Näin päästään parantamaan kustannustehokkuutta ja energiankulutuksen vähentyminen on huomattava. Ideana on myös erilaisten mittausten lisääminen mikrokontrollerin puolelle. Hyvänä mahdollisuutena on vielä mikrokontrollerilla releohjaus web-käyttöliittymän puolella tai mikrokontrollerin ohjelmakoodissa suoraan.

Alunperin työ oli tarkoitus toteuttaa käyttäen jotain LPWAN (Low-Power Wide-Area Network) tekniikkaa, kuten LoRa. Valitettavasti koululle ei ollut saapunut tuona aikana tarvittavia laitteita, kun työn aloitin. LoRassa päästään jopa yli kymmenkertaiseen kantavuuteen verrattuna ZigBee tekniikkaan. Tiedonsiirtonopeus on paljon pienempi LoRassa mikä tässä työssä ei tuota ongelmia, koska tiedonsiirtoon tarvittava nopeus on muutenkin pieni. Etuna LoRassa on myös sen vähäinen virrankulutus.

Pitäen silmällä teollisuuden ja automaation tarpeita, LPWAN on yleisesti siihen paljon parempi ratkaisu, kuin lyhyen kantaman WPAN tekniikat, kuten ZigBee.

Omasta mielestäni työ oli todella mielenkiintoinen ja haasteellinen. Työn aikana opin paljon uusia asioita ja kehityin paremmaksi hahmottamaan ja ymmärtämään toiminnallista kokonaisuutta.

## LÄHDELUETTELO

- /1/ VAMK Oy. Historia. Viitattu 10.2.2016  
[http://www.puv.fi/fi/about/vamk\\_oy/](http://www.puv.fi/fi/about/vamk_oy/)
- /2/ Arduino. Viitattu 22.2.2016  
<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- /3/ Mega2560-pin-map(1).png. Viitattu 21.3.2016  
[http://layoftheland.net/archive/art4645c-2012/weeks6-11/pressure\\_sensor/Mega2560-pin-map\(1\).png](http://layoftheland.net/archive/art4645c-2012/weeks6-11/pressure_sensor/Mega2560-pin-map(1).png)
- /4/ Raspberry Pi. Viitattu 21.3.2016  
[https://www.linux.fi/wiki/Raspberry\\_Pi](https://www.linux.fi/wiki/Raspberry_Pi)
- /5/ Operating Systems. Viitattu 21.3.2016  
<https://www.raspberrypi.org/downloads/>
- /6/ Raspberry Pi GPIO Layout – Pi 1 Model B Revision 1. Viitattu 21.3.2016  
<http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>
- /7/ Tampere University of Technology. IEEE 802.15.4 Viitattu 2.2.2016  
<http://www.cs.tut.fi/kurssit/TLT-6556/Slides/4-802.15ZigBee.pdf>
- /8/ Specifications, XBee. Viitattu 1.23.2016  
<https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>
- /9/ DESCRIPTION, xbee-radios, Reference Pinout. Viitattu 1.23.2016  
<https://www.adafruit.com/product/126>
- /10/ The Development of the C Language\*. Viitattu 23.3.2016  
<http://www.bell-labs.com/usr/dmr/www/chist.html>
- /11/ Related languages. Viitattu 24.3.2016  
[https://en.wikipedia.org/wiki/C\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
- /12/ 1.6 C and C++, 1.5 Use of C++, 2.1 What is C++. 24.3.2016  
The C++ Programming Language: Special Edition (3rd Edition), by Bjarne Stroustrup, ISBN-13: 978-0201700732
- /13/ Chapter 1: Python, what Python?. Viitattu 24.3.2016  
Learn Python in One Day and Learn It Well, ISBN-13: 978-1506094380

- /14/ PHP Tools, FI, Construction Kit, and PHP/FI. Viitattu 24.3.2016  
<http://php.net/manual/en/history.php.php>
- /15/ Introduction to SQL. Viitattu 29.3.2016  
[http://www.w3schools.com/sql/sql\\_intro.asp](http://www.w3schools.com/sql/sql_intro.asp)
- /16/ Product Summary, 2 Interface Specifications. Viitattu 30.3.2016  
[https://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokument e/Humidity\\_Sensors/Sensirion\\_Humidity\\_Sensors\\_SHT7x\\_Datasheet\\_V5.pdf](https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokument%20e/Humidity_Sensors/Sensirion_Humidity_Sensors_SHT7x_Datasheet_V5.pdf)
- /17/ Ylösvetovastus. Viitattu 30.3.2016  
<https://fi.wikipedia.org/wiki/Y1%C3%B6svetovastus>
- /18/ Libraries and Example Code. Viitattu 31.3.2016  
<http://playground.arduino.cc/Code/Sensirion>
- /19/ Heksadesimaalijärjestelmä. Viitattu 31.3.2016  
<https://fi.wikipedia.org/wiki/Heksadesimaalij%C3%A4rjestelm%C3%A4>
- /20/ ASCII Table and Description. Viitattu 31.3.2016  
<http://www.asciitable.com/>
- /21/ components\_ftdiconnected.jpg. Viitattu 4.4.2016  
<https://learn.adafruit.com/assets/6750>
- /22/ Python script, MySQL. Viitattu 7.4.2016  
<https://wingoodharry.wordpress.com/2015/01/05/raspberry-pi-temperature-sensor-web-server-part-2-setting-up-and-writing-to-a-mysql-database/>
- /23/ BUILD A LAMP WEB SERVER WITH WORDPRESS. Viitattu 7.4.2016  
<https://www.raspberrypi.org/learning/lamp-web-server-with-wordpress/worksheet/>
- /24/ Line Chart. Viitattu 7.4.2016  
<https://developers.google.com/chart/interactive/docs/gallery/linechart>
- /25/ XBEE® FAMILY FEATURES COMPARISON. Viitattu 8.4.2016  
[http://www.digi.com/pdf/chart\\_xbee\\_rf\\_features.pdf](http://www.digi.com/pdf/chart_xbee_rf_features.pdf)