

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Tietokonetekniikka

Opinnäytetyö

Pekka Harjamäki

Langaton valvontaverkko

Työn ohjaaja Kai Poutanen
Ylöjärvi 11/2010

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma, Tietokonetekniikka

Tekijä: Pekka Harjamäki
Työn nimi: Langaton valvontaverkko
Sivumäärä: 25 sivua ja 8 liitesivua
Valmistumisaika: 11/2010
Työn ohjaaja: Kai Poutanen

TIIVISTELMÄ

Tässä työssä toteutettiin kiinteistön valvontaan soveltuva monipuolinen ja helposti laajennettava langaton sensoriverkosto. Sensorit kykenevät valvomaan huoneiden lämpötilaa ja kosteutta sekä ilmaisemaan mahdollisia vesivuotoja. Ne myös ilmoittavat palamaan jääneistä valoista esimerkiksi kellaritiloissa.

Sensoreiden tiedot tallennetaan keskitetysti MySQL-tietokantaan, josta ne välitetään edelleen Zabbix-valvontasovellukselle. Sovelluksesta voidaan tarvittaessa katsoa internetin välityksellä lähes reaaliaikaista tietoa eri sensoreiden tilasta. Lisäksi sovellus osaa automaattisesti lähettää varoituksen sekä sähköpostilla että SMS-viestillä, mikäli jokin ennalta määritetty raja-arvo ylittyy.

Mittalaitteen ytimenä toimii PIC16F88-mikrokontrolleri, joka kommunikoi vastaanottimen kanssa ZigBee-verkon välityksellä käyttäen XBEE-radiomodeemeita. ZigBee-verkko valittiin sen hyvän kantomatkan, melko pienen virrankulutuksen ja yksinkertaisen liitettävyyden vuoksi.

Avainsanat PIC, XBEE, ZigBee, SHT15

TAMK University of Applied Sciences
Department of Computer Engineering, Computer Engineering

Writer: Pekka Harjamäki

Thesis: Wireless sensor network

Pages: 25 pages and 8 appendix pages

Graduation time: 11/2010

ABSTRACT

This thesis describes the build process of diverse and easily expandable wireless sensor network for property. The sensors can monitor rooms temperature and humidity, detect possible water floods and also send out notification if lights are accidentally left on. The latter can happen for example in basement, which is quite seldom visited.

Data from the sensors is stored to centralized MySQL-database, and from there it is transferred to Zabbix monitoring software. Zabbix allows user to see almost real-time information over internet and also it can send alert via email and SMS if certain values are exceeded.

The core of the sensor device is PIC16F88-microcontroller and it communicates with receiver over ZigBee-network, using XBEE-radio modems. ZigBee-network was selected mainly because it has good range even indoors, radio modem has low current usage and it's easy to interface with microcontroller.

Keywords: PIC, XBEE, ZigBee, SHT15

Esipuhe

Opinnäytetyössä kuvattu laite on kehitetty pääasiassa omaan käyttöön, sillä vastaavia järjestelmiä on jo kaupallisesti saatavilla, mutta ne ovat minulle toiminnallisuudeltaan liian suppeita tai erittäin kalliita. Näin ollen oman järjestelmän kehitys oli taloudellisesti kannattavin ratkaisu.

Pitääkseni opinnäytetyön rajauksen kohtuullisena, nykyinen järjestelmä on kehitetty pelkästään tilojen valvontaan. Koska käytössä on kuitenkin kaksisuuntainen radiotie, voidaan laitteen toiminnallisuutta myöhemmin laajentaa myös talojen toimintojen ohjaukseen, kuten tuuletuksen ja lämmityksen säätelyyn.

Kiitän Kai Poutasta työn valvonnasta ja Sanoma Data Oy:n henkilöstöä arvokkaista kommentteista työn osalta.

Ylöjärvellä 12.11.2010

Pekka Harjamäki

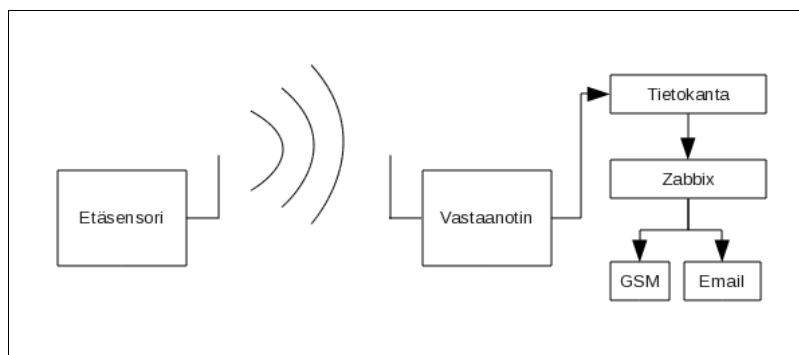
SISÄLLYSLUETTELO

1 JOHDANTO.....	1
2 TOIMINTAPERIAATE.....	2
2.1 Zabbix-valvontaohjelmisto.....	2
2.2 ZigBee-verkko.....	4
2.3 SHT15-sensori.....	5
2.4 PIC-mikrokontrolleri.....	8
2.4.1 Mikrokontrollerin sovelluskehitys.....	10
2.4.2 Mikrokontrollerin ohjelmointi.....	11
2.5 XBEE-radiomodeemi.....	12
3 LAITTEISTON SUUNNITTELU JA TOTEUTUS.....	14
3.1 Vastaanotin.....	14
3.2 Etäsensori.....	15
3.2.1 Kosteustunnistin.....	17
3.2.2 Valosensori.....	18
4 OHJELMISTOJEN TOTEUTUS.....	20
4.1 Etäsensarin ohjelmisto.....	20
4.2 Vastaanottimen ohjelmisto.....	21
4.2.1 Tietokannan rakenne.....	21
4.2.2 Valvontajärjestelmän ohjelmisto.....	23
5 YHTEENVETO.....	23
LÄHTEET.....	24
LIITTEET.....	25

1 JOHDANTO

Opinnäytetyössä on kuvattu valvontajärjestelmän ensimmäinen prototyyppi. Ero lopulliseen versioon tulee olemaan pääasiassa se, että laite kykenee valvonnan lisäksi myös ohjaamaan etälaitteita, kuten puhaltimia ja lämmitystä.

Valvontajärjestelmän toiminta (kuva 1) perustuu siihen, että akkukäyttöinen sensori lähettää ZigBee-verkossa vastaanottimelle mittaustiedot ennalta määrätyin väliajoin. Vastaanottimen ohjelmisto tallentaa saamansa tiedot MySQL-tietokantaan, josta Zabbix-valvontasovellus noutaa ne jatkokäsittelyä varten. Mikäli mitta-arvot poikkeavat asetetuista raja-arvoista, lähetetään hälytys eteenpäin sekä sähköpostilla, että tekstiviestillä.



Kuva 1. Järjestelmän toimintaperiaate

Prototyyppiversion ensimmäisenä käyttökohteena tulee olemaan vanhan omakotitalon maakellari. Laitteen on tarkoitus valvoa, ettei kyseinen tila jäädy, eikä toisaalta pääse liian kosteaksi. Etäsensoriin ominaisuuksiin kuuluu myös vesitunnistin ja valotunnistin. Vesitunnistimen tehtävänä on ilmoittaa, mikäli tilaan tulvii pintavettä sadekauden aikana, ja valotunnistin taas ilmoittaa, jos valot unohtuvat palamaan.

Fyysinen laite on koottu osittain valmiista osista ja osittain itse alusta alkaen valmistetuista osista. Esimerkiksi valmis vastaanotin oli niin halpa, ettei sitä olisi ollut mahdollista valmistaa itse samaan hintaan. Etäsensori taas on sen verran uniikki, ettei vastaavaa löydy kaupallisesti valmistettuna.

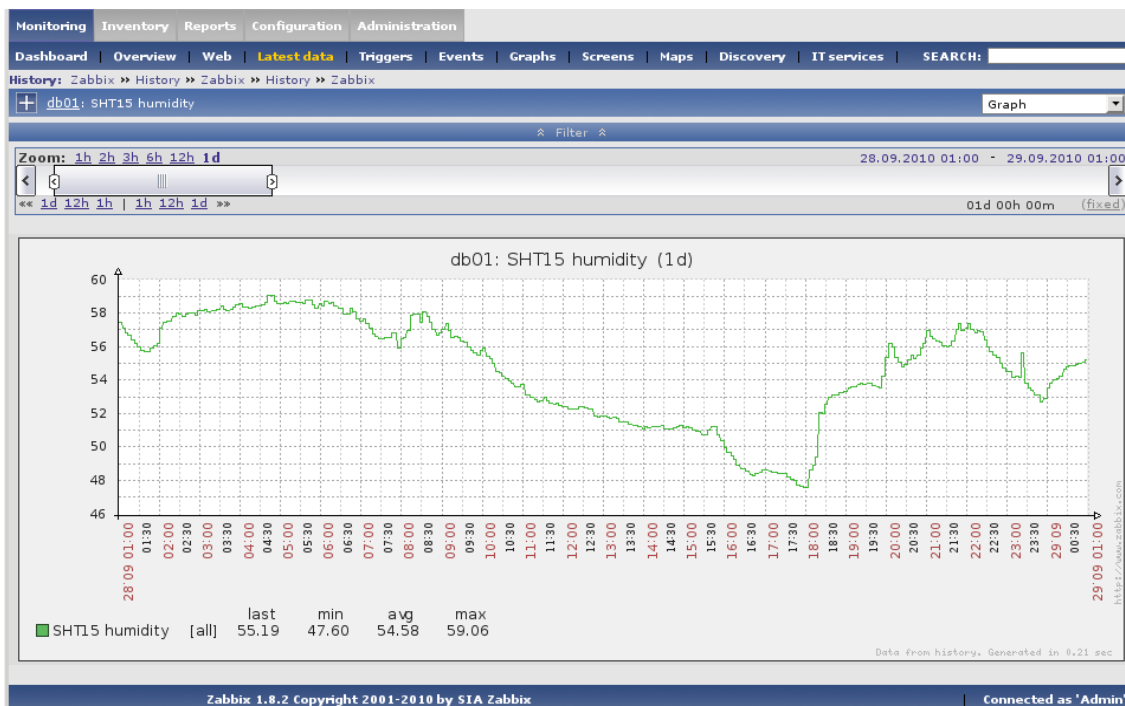
2 TOIMINTAPERIAATE

Tässä kappaleessa tarkastellaan projektin olennaisimpien osien toimintaperiaatetta. Itse laite sekä ohjelmisto on kuvattu luvuissa 3 ja 4.

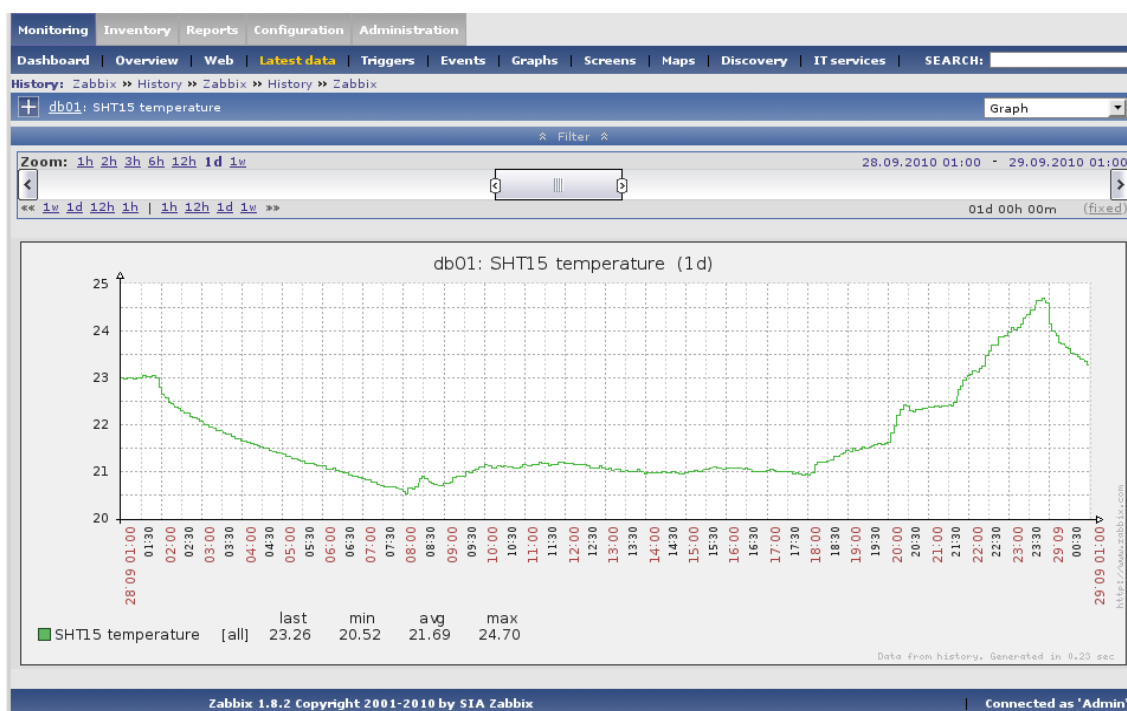
2.1 Zabbix-valvontaohjelmisto

Zabbix on yrityskäyttöön suunnattu avoimen lähdekoodin valvontasovellus, jonka vahvuuksia ovat mm. monipuoliset valvonta- ja raportointimahdollisuudet sekä skaalautuvuus niin pieniin kuin suuriinkin järjestelmiin. Sovellus on kehitetty pääasiassa palvelimien ja ohjelmistojen valvontaan, mutta sen joustavan rakenteen ansiosta sillä voidaan valvoa myös erilaisia ulkoisia sensoreita.

Sovellus tarjoaa tiedonkeräyksen lisäksi myös mahdollisuuden automaattisiin hälytyksiin sähköpostilla, pikaviestimillä ja SMS-viesteillä ennalta määrättyjen raja-arvojen ylityttyä. Lisäksi ohjelmasta voidaan katsoa www-sivun kautta lähes reaaliaikaista tietoa sensoreiden tilasta ja myös tuottaa graafisia esityksiä kerätystä tiedosta (kuvat 2 ja 3).



Kuva 2. Valvontasovelluksen tuottama graafinen esitys huonetilan ilmastusta



Kuva 3. Valvontasovelluksen tuottama graafinen esitys huonetilan lämpötilasta

Ohjelmisto koostuu kolmesta eri osasta, jotka ovat palvelin, asiakas ja käyttöliittymä. Palvelinsovelluksen tehtävänä on vastaanottaa tietoa muilla palvelimilla ajossa olevilta asiakasohjelmistoilta ja tallentaa se tietokantaan. Palvelinsovellus voi myös jossain määrin osallistua palveluiden valvontaan, kuten esimerkiksi tarkkailla verkkosivujen vasteaikaa, mutta pääasiassa sen tehtäviin kuuluu tiedon tallennus ja hälytyksien välitys.

Asiakassovellus puolestaan suorittaa varsinaisen seurantatiedon keräämiseen. Seurantatiedot ovat yleensä palvelimen suorituskykyyn tai ”terveyteen” viittaavia lukuja, kuten muistin, levyväylän ja suorittimen käyttöaste. Lisäksi asiakassovellus hyväksyy myös ulkopuolisesta lähteestä tulevat kokonais- ja desimaaliluvut, joten sille voidaan syöttää käytännössä mitä tahansa numeerista tietoa.

Ohjelmiston kolmas osuus on PHP:lla toteutettu internetin kautta toimiva käyttöliittymä, jolla voidaan seurata järjestelmien toimintaa, hallita hälytyksiä sekä muokata järjestelmän rakennetta. Käyttöliittymä on tarkoitettu pääasiassa selaimen kautta käytettäväksi, mutta viime aikoina markkinoille on alkanut ilmestyä myös natiiveja mobiiliversioita älypuhelimille.

Käyttöliittymän kautta hallittavia rakenteeseen liittyviä asioita ovat mm. palvelimien liittäminen ja poistaminen valvonnasta sekä valvottavien lukujen ja niihin liittyvien toimintamallien muokkaus.

2.2 ZigBee-verkko

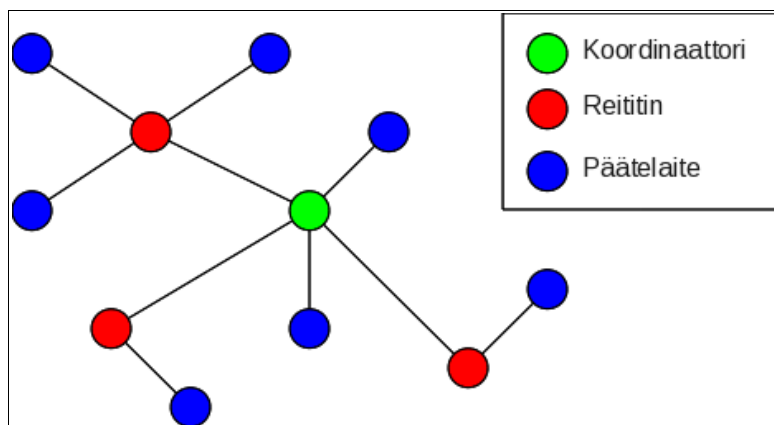
ZigBee on avoimiin standardeihin perustuva langaton verkkoteknologia, joka on kehitetty pääasiallisesti laitteiden väliseen kommunikointiin. Verkko toimii IEEE 802.15.4-spesifikaatioiden mukaisesti taajuuksilla 2,4 GHz, 900 MHz ja 868 MHz. /1/

Standardia kehittää ZigBee-allianssi, johon kuuluu yli 200 yritystä, kuten Pioneer, Samsung, Honeywell ja LG. Allianssin tarkoituksena on kehittää langaton verkkostandardi, joka on yhteensopiva eri valmistajien välillä ja johon on mahdollista kehittää yksinkertainen, halpa ja vähävirtainen radiomodeemi. /2, 3/

Suurin ero ZigBee-verkolla verrattuna Bluetooth- ja Wi-Fi-verkkoihin on alhainen siirtonopeus ja päätelaitteiden mahdollisuus olla vähävirtaisessa unitilassa suhteellisen pitkiä aikoja, ilman että ne putoavat pois verkosta. Alhaisen siirtonopeuden takia, verkossa ei ole mielekasta siirtää videoita tai ääntä, mutta toisaalta verkko onkin alun perin suunniteltu lähinnä ohjaukseen ja sensoritiedon välittämiseen.

Yksittäinen verkko (kuva 4) koostuu enintään 65 536 laitteesta, joista yhden on oltava verkon koordinaattori ja muut laitteet ovat joko päätelaitteita tai reitittimiä. Edellä mainittu rajoitus johtuu 16-bittisestä osoitevaruudesta, joka ei salli kuin enintään 2^{16} osoitetta.

Tarvittaessa rajoitusta voidaan kuitenkin kiertää rakentamalla samalle fyysiselle alueelle useita eri paikallisverkkoja. Näiden tunnuksena toimii 16-bittinen luku nollan ja 16383 välillä. Paikallisverkkoja voidaan myös käyttää eri toimintojen eriyttämiseen. Tällä toimenpiteellä ei tosin saada käyttöön lisää radiokaistaa, mutta voidaan vähentää järjestelmän vikaherkkyttä.



Kuva 4. Esimerkki ZigBee-verkosta

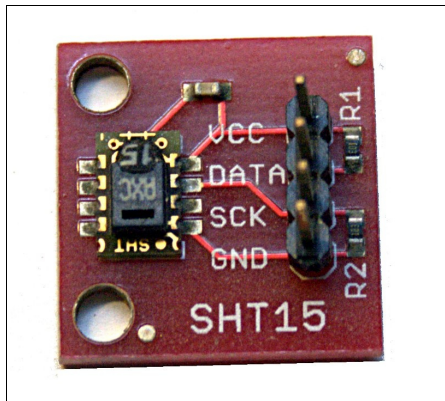
Verkossa kommunikointi tapahtuu enintään 127 tavun mittaisten viestien avulla, joita voidaan lähettää joko broadcast- tai unicast-lähetyksinä. Unicast-viesti on tarkoitettu ainoastaan yhdellä laitteelle ja viestin kohdennus tapahtuu käyttämällä laitteille liittymisen yhteydessä annettua 16-bittistä tunnistetta.

Käytettäessä 16-bittistä tunnistetta pitää ottaa huomioon, että tunniste voi muuttua, ja siksi lähtävän laitteen pitää selvittää vastaanottajan osoite multicast-viestin avulla. Toisin kuin unicast-viesti, multicast-viesti kulkee koko verkon läpi ja on kohdennettu vastaanottajalle 64-bittisen uniikin tunnisteen avulla.

Laitteiden välisessä viestinnässä on myös mahdollista käyttää ns. ryhmä-ID:itä, joita voi samassa verkossa olla enintään 240 erilaista. Tästä määrästä on tosin vapaasti käytettävissä vain 220 kappaletta, sillä loput 20 on varattu verkon sisäiseen käyttöön. Ryhmä-ID:tä voidaan käyttää esimerkiksi niin, että yksi kytkin voi ohjata useita valaisimia samalla viestillä. Tällöin valaisimia voidaan liittää verkkoon ilman, että kytkimen ohjelmistoon tarvitsee tehdä muutoksia.

2.3 SHT15-sensori

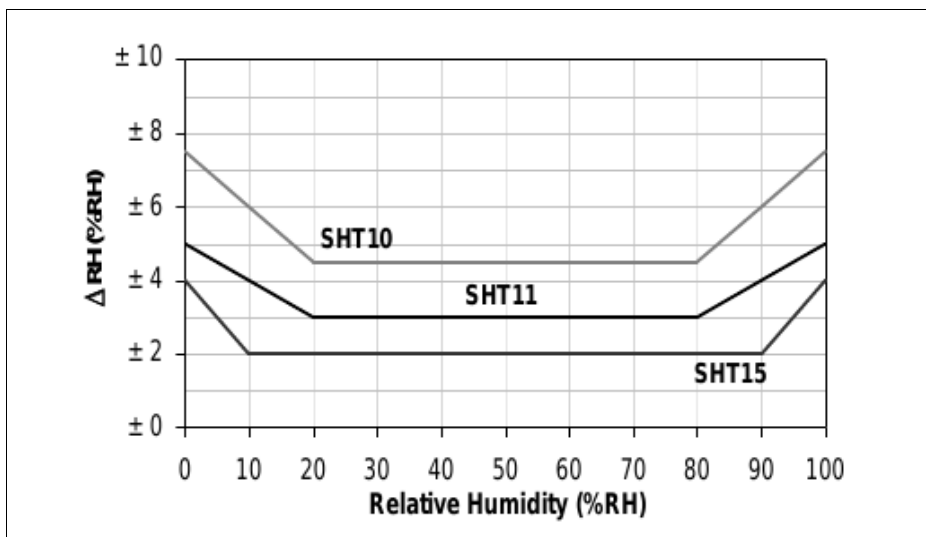
SHT15 on Sensitronin valmistama lämpötilan ja ilmankosteuden mittaukseen tarkoitettu komponentti, joka on esillä kuvan 5 piirilevyssä vasemmassa laidassa. Mittalaitteiden lisäksi piirissä on sisäinen A/D-muunnin, jonka tarkoituksena on vähentää siirtotien ja piirilevyn aiheuttamia epätarkkuuksia sekä mahdollistaa liittymisen digitaaliseen väylään.



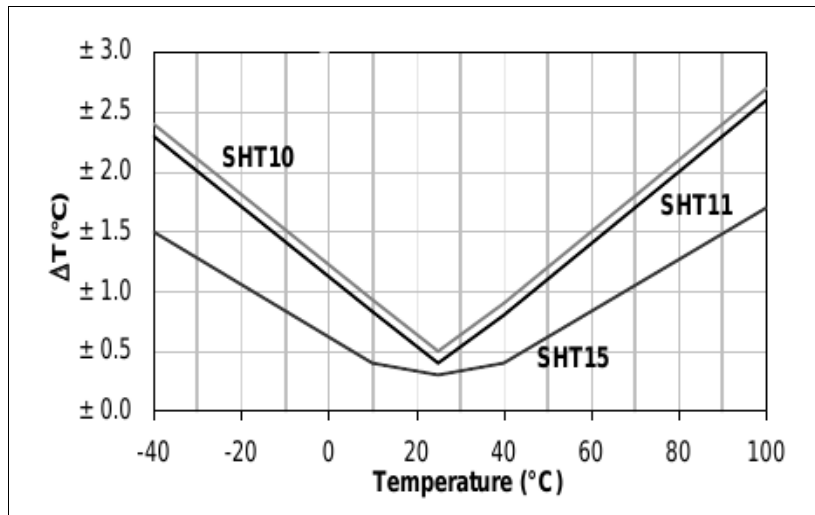
Kuva 5. SHT15-sensori

Kuvassa 5 esitetty piirilevy on Sparkfunin valmistama, ja siinä on valmiiksi asennettuna sekä SHT15-piiri että väylän päätevastukset. Opinnäytetyössä päädyttiin sensorin osalta valmiiseen piirilevyyn lähinnä siksi, että se nopeutti projektin valmistumista kohtuullisen pienillä kustannuksilla. Piirilevyn piirikaavio on liitteessä 2.

SHT15 esittää lämpötilan oletuksellisesti 14 bitin resoluutiolla ja ilmankosteuden 12 bitin resoluutiolla. Mikäli mittausten välistä aikaa halutaan pienentää, voidaan resoluutiota laskea siten, että lämpötila esitetään 12 bitillä ja kosteus 8 bitillä. Komponentin mittaustarkkuudet on esitetty kuvissa 6 ja 7.



Kuva 6. SHT15-sensorin ilmankosteuden mittaustarkkuus /6/



Kuva 7. SHT15-sensorin lämpötilan mittaustarkkuus /6/

Kommunikointi piirin kanssa tapahtuu käyttäen I2C-protokollaa muistuttavaa sarjaliikenneprotokollaa, jolla voidaan vaihtaa mittausparametreja sekä lukea mittaustulokset. Piiri on suunniteltu siten, että se voidaan tarvittaessa kytkeä olemassa olevaan I2C-väylään ilman että muut laitteet siitä häiriintyvät. Väyläohjain joutuu tosin silti vaihtamaan protokollaa riippuen siitä, haluaako se saada yhteyden I2C-laitteisiin tai SHT15-sensoriin.

Mittaustuloksien osalta piiri ei palauta suoraan käyttökelpoisia tuloksia, vaan ne pitää laskea valmistajan antamien kaavojen avulla. Piirin lämpötilalukema voidaan laskea kaavan 1 ja taulukon 1 avulla. Ilmankosteus puolestaan voidaan laskea käyttämällä kaavoja 2 ja 3 sekä taulukkoja 2 ja 3. Kahdesta jälkimmäisestä kaavasta ensimmäisellä korjataan sensorin epälineaarisuus ja toisella lasketaan lämpötilan vaikutus ilmankosteuteen.

$$T = d_1 + d_2 \cdot SO_T \quad (1), \text{ jossa } /8/$$

- SO_T = Piirin palauttama digitaalinen arvo.

Taulukko 1. Vakiot kaavaan 1 /8/

VDD	d ₁ (°C)	d ₁ (°F)	SO _T	d ₂ (°C)	d ₂ (°F)
5V	-40.1	-40.2	14bit	0.01	0.018
4V	-39.8	-39.6	12bit	0.04	0.072
3.5V	-39.7	-39.5			
3V	-39.6	-39.3			
2.5V	-39.4	-38.9			

$$RH_{linear} = C_1 + C_2 \cdot SO_{RH} + C_3 \cdot SO_{RH}^2 \quad (2), \text{ jossa } /8/$$

- SO_{RH} = Sensorin ilmoittama ilmankosteus.

Taulukko 2. Vakiot kaavaan 2 /8/

SO _{RH}	C ₁	C ₂	C ₃
12 bit	-2.0468	0.0367	-1.5955E-6
8 bit	-2.0468	0.5872	-4.0845E-4

$$RH = (T_{°C} - 25) \cdot (t_1 + t_2 \cdot SO_{RH}) + RH_{linear} \quad (3), \text{ jossa } /8/$$

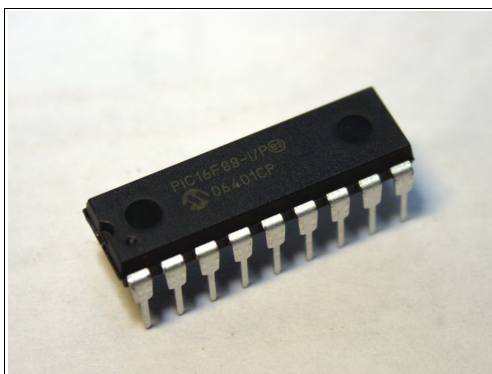
- $T_{°C}$ = Lämpötila celsiusasteina.
- SO_{RH} = Sensorin ilmoittama ilmankosteus.

Taulukko 3. Vakiot kaavaan 3 /8/

SO _{RH}	t ₁	t ₂
12 bit	0.01	0.00008
8 bit	0.01	0.00128

2.4 PIC-mikrokontrolleri

Projektissa käytetty mikrokontrolleri PIC16F88 (kuva 8) on Microchipin valmistama 8-bittinen suoritin, jonka tärkeimmät ominaisuudet on esitetty taulukossa 4. Siitä näkee, ettei mikrokontrolleri ole muistikapasiteetiltaan tai suorituskyvyltään kovinkaan tehokas. Lisäksi piiristä on jo kirjoitushetkellä uudempia versioita. Kyseiseen malliin päädyttiin kuitenkin siksi, että sitä oli valmiiksi varastossa.



Kuva 8. PIC16F88-mikrokontrolleri

Taulukko 4. PIC16F88-mikrokontrollerin ominaisuuksia

Käyttöjännite	<ul style="list-style-type: none"> • 2.0 – 5.5V
Sisäisen oskillaattorin taajuudet	<ul style="list-style-type: none"> • 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz
Muisti	<ul style="list-style-type: none"> • Flash: 7168 tavua • SRAM: 368 tavua • EEPROM: 256 tavua
Ominaisuuksia	<ul style="list-style-type: none"> • 16 I/O-pinniä • 7-kanavainen 10-bittinen A/D-muunnin • AUSART
Laskurit	<ul style="list-style-type: none"> • Timer0 ; 8-bittinen ajastin • Timer1 ; 16-bittinen ajastin, joka toimii myös vähävirtaisessa unitilassa • Timer2 ; 8-bittinen ajastin

Saatavuuden lisäksi projektin kannalta olennaiset syyt kyseisen mikrokontrollerin valintaan olivat piirin sisäänrakennetuissa ominaisuuksissa. PIC16F88-mikrokontrollerissa on sisäinen A/D-muunnin, ja lisäksi se kykenee toimimaan ilman ulkopuolista oskillaattoripiiriä, joten näiden toimintojen osalta ei piirilevyllä tarvittu ylimääräisiä osia.

Toiminnalliselta kannalta tarkasteltuna mikrokontrolleri on kuvan 9 mukainen. Kuvassa on esitetty mm. piirin sisäisten väylien ja toimintojen keskeinen suhde.

Yksi vaihtoehto edellä mainitulle C-kääntäjälle on Mikroelektronikan kehittämä MikroC-kehitysympäristö. Sen kaupallinen versio maksaa vain noin neljäsosan siitä mitä kilpailijan tuote ja lisäksi kehitysympäristöstä löytyy ilmainen versio, joka toimii kuten kaupallinen, mutta rajoittaa ohjelman lopullista kokoa.

Näistä kahdesta vaihtoehdosta päädyttiin tässä projektissa jälkimmäiseen, koska suorittimen ohjelma ei ole rivimäärältään suuri. Lisäksi jos tuotetta myöhemmin kehitetään suuremmaksi kuin mitä ilmainen versio sallii, siirtyminen kaupalliseen versioon on helppoa, koska sovelluksen lähdekoodia ei tarvitse muuttaa.

2.4.2 Mikrokontrollerin ohjelmointi

Konekieliseksi käännetyn ohjelman siirto mikrokontrolleriin tapahtuu käyttämällä esimerkiksi kuvassa 10 esitettyä PICkit 2 -ohjelmointilaitetta, joka kykenee varsinaisen piirin ohjelmoinnin ohella myös toimimaan virtalähteenä pienille projekteille. Lisäksi sen kautta voidaan muodostaa sarjaliikenneyhteys kohdepiirille.

Ohjelmointilaitte on suunniteltu siten, että mikrokontrolleri on mahdollista ohjelmoida uudelleen, vaikka se olisikin jo kiinni kohdepiirissä. Tämä mahdollistaa ohjelmien nopean testaamisen, ja lisäksi vähentää mikrokontrollerin hajoamisen riskiä verrattuna sellaisiin ohjelmointilaitteisiin, joissa mikrokontrolleri pitää irrottaa ohjelmoinnin ajaksi.

Haittapuolena menetelmässä on tosin se, että kaksi I/O-pinniä pitää varata ohjelmointilaitteelle. Nämäkin pinnit ovat periaatteessa mahdollista saada käyttöön, mutta tällöin pitäisi ohjelmointiliittimen ja lopun piirin väliin lisätä suojapiiri, joka estää kohdepiiriä häiritsemästä ohjelmointia.



Kuva 10. PICKIT 2 -ohjelmointilaite

2.5 XBEE-radiomodeemi

Työssä käytetty ZigBee-verkkoon suunniteltu radiomodeemi on nimeltään XBEE ja sen olennaisimmat ominaisuudet on esitetty taulukossa 5. Käytössä on tuotteen 2.5-sarjan kahta eri mallia, joista lähetin käyttää integroidulla antennilla varustettua mallia (kuva 11) ja vastaanotin taas ulkopuolisella antenniliitännällä varustettua mallia (kuva 13).

XBEE-piiri sisältää lähetinvastaanottimen lisäksi myös nelikanavaisen 10-bittisen A/D-muuntimen, sekä useita I/O-pinnejä. Laitteen omaa A/D-muunninta ei tosin käytetty projektissa, sillä sen mittausalue ylettyy enintään 1,2 volttiin asti, kun taas PIC-mikrokontrolleri kykenee mittaamaan jännitteen koko toimintajännitteen alueelta.



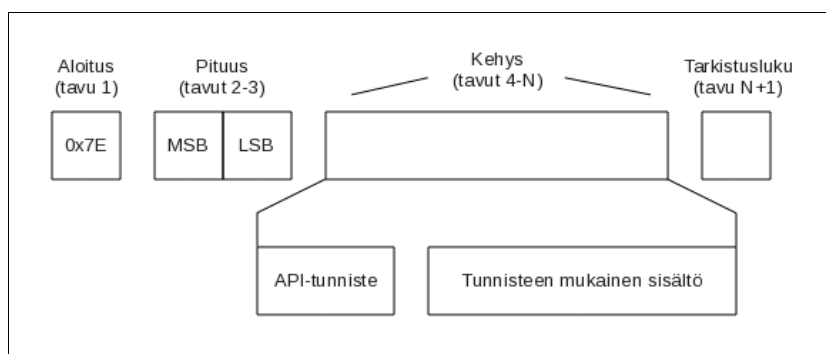
Kuva 11. XBEE-radiomodeemi integroidulla antennilla

Taulukko 5. XBEE-radiomodeemin ominaisuudet /4/

Kantomatka sisätiloissa	40 metriä
Kantomatka ulkona avoimessa tilassa	120 metriä
Lähetysteho	2 mW (+3 dBm), vahvistimen kanssa 1,25 mW (+1 dBm), ilman vahvistinta
Suurin tiedonsiirtonopeus	250 kbps
Vastaanottimen herkkyys	-96 dBm
Käyttöjännite	2,1 – 3,6 V
Suurin virrankulutus	40 mA (3,3 V käyttöjännitteellä)
Lepotilan virrankulutus	1 μ A (25°C:n lämpötilassa)
Taajuus	ISM 2,4 GHz
Käyttölämpötila	-40°C – 85°C
Valmistaja	Digi International

Yksinkertaisimmillaan XBEE-piiriä voi käyttää radiomodeemina siten, että kytkee sen suoraan kiinni mikrokontrollerin UART-piiriin. Näin toimittaessa XBEE käyttäytyy kuten langaton sarjaliikennekaapeli.

Jos radiomodeemin ja verkon ominaisuuksia halutaan käyttää tehokkaammin, on laitteeseen saatavissa monimutkaisempi ohjelmisto, joka mahdollistaa liikennöinnin piirin API-rajapinnan kautta. Siinä missä sarjaliikennetilassa riittää, että haluttu tieto vain lähetetään eteenpäin, pitää se API-rajapintaa hyödynnettäessä pakata määrätyn kehyksen sisälle (kuva 12).



Kuva 12. API-rajapinnan kehyksen rakenne /4/

API-rajapinnan käyttäminen vaatii ohjaavalta järjestelmältä monimutkaisempaa ohjelmistoa, mutta toisaalta se tarjoaa nopeamman ja luotettavamman tavan ohjata tietoliikennettä, etenkin monimutkaisen sensoriverkon tapauksessa. Käytännön erot tulevat vastaan esimerkiksi siinä, miten mikrokontrollerin asetuksia muutetaan sarjaliikennetilassa ja API-rajapinnan kautta.

Sarjaliikennetilassa modeemin hallinta vaatii ohjelmointitilan käyttöä, jonne päästäkseen pitää modeemille lähettää erityinen merkkisarja ja vasta tämän jälkeen voidaan lähettää varsinaiset käskyt. Lisäksi ohjelmointitilaan päästäkseen täytyy ohjaavan mikrokontrollerin pitää useiden satojen millisekuntien taukoa lähetyksissä ennen ja jälkeen merkkisarjan.

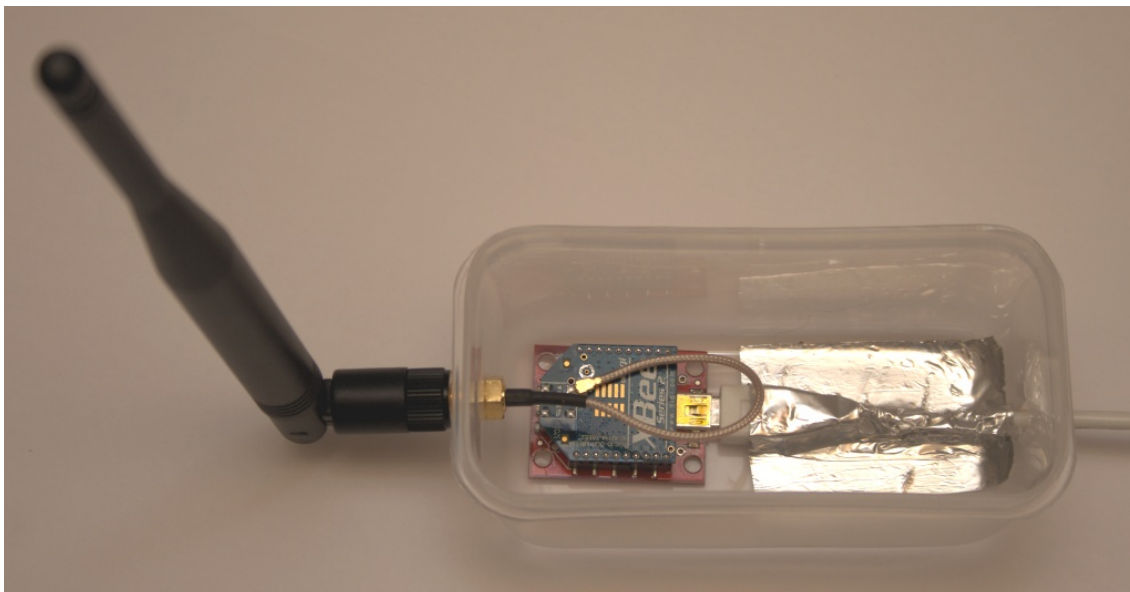
API-rajapintaa käytettäessä riittää, että ohjaava mikrokontrolleri lähettää komennon ohjauskehysten sisällä ja käsittelee vastauksen. Lisäetua rajapinta tuo myös siihen, että paluukehykset kertovat suoraan, mistä vastaanotettu tieto on peräisin. Tällöin ei ole riskiä sekoittaa verkosta tulleita viestejä ja kommentojen paluuarvoja keskenään.

3 LAITTEISTON SUUNNITTELU JA TOTEUTUS

Toteutukseltaan sekä vastaanotin että etäsensori ovat kumpikin melko yksinkertaisia, sillä laitteiden pääasiallinen tarkoitus on ainoastaan tiedon keräys ja välitys eteenpäin. Laitteisiin olisi suorittimen muistikapasiteetin puolesta mahdollista lisätä monimutkaisempikin toimintalogiikka, mutta siitä ei ole kokonaisuuden kannalta etua, koska tiedon prosessointi tapahtuu kuitenkin lopulta tietokoneella.

3.1 Vastaanotin

Vastaanotin (kuva 13) on rakenteellisesti erittäin yksinkertainen, sillä se muodostuu vain XBEE-radiomodeemista, sen USB-sovittimesta ja ulkoisesta antennista. USB-sovitin on Sparkfunin valmistama ja sen kytkentäkaavio on liitteessä 3.

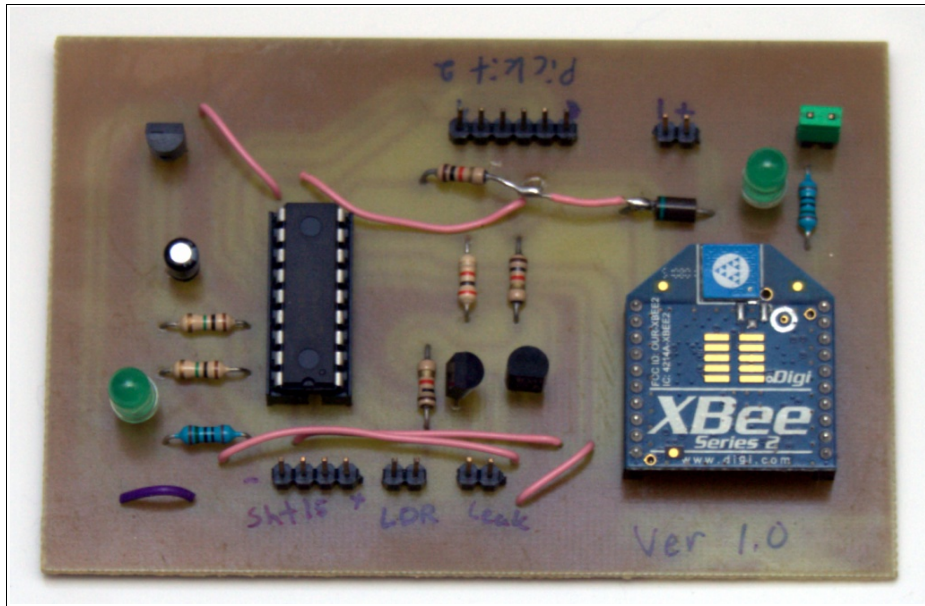


Kuva 13. Vastaanotin koteloituna

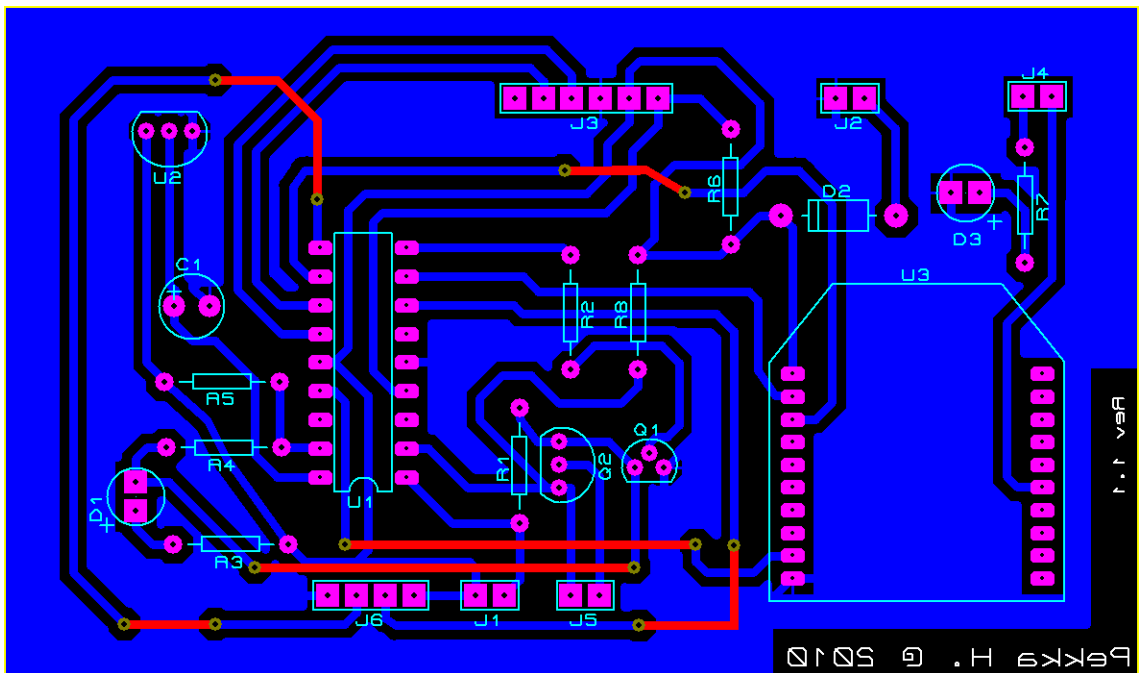
3.2 Etäsensori

Etäsensoren (kuvat 14 ja 15) suunnittelun lähtökohtana on ollut tehdä siitä mahdollisimman vähävirtainen ja komponenttimäärältään pieni. Käytännössä nämä tavoitteet saavutettiin siten, että vastaanotin käsittelee suurimman osan tiedoista ohjelmallisesti.

Tämän ratkaisumallin etuna on se, että sensorissa käytetyn mikrokontrollerin osalta voidaan suorituskyky vaihtaa pieneen virrankulutukseen ja halvempaan komponenttihintaan. Lisäksi järjestelmään voidaan myöhemmin tehdä korjauksia ja lisätä jopa uusia ominaisuuksia ilman, että piirilevyn rakenteeseen tai ohjelmistoon joudutaan välttämättä tekemään muutoksia.



Kuva 14. Etäsensorin piirilevyn 1.0 versio



Kuva 15. Lähettimen piirilevyn syövytysmaski

Järjestelmän kannalta tärkeimmät osat, joita ei piirilevyllä voitu ohjelmallisilla ratkaisuilla toteuttaa, ovat 2,5 V referenssiännitelähde (Kuva 15 ; U2), vahvistimena toimiva darlingtontransistori (Kuva 15 ; Q2) ja tavallinen NPN transistori (Kuva 15 ; Q1), joka toimii virtakytkimenä mittaussiireille.

Piirille sijoitettujen ledien tarkoitus on kertoa radiomodeemin tilasta (Kuva 15 ; D3) ja siitä milloin mittauksia tehdään (Kuva 15 ; D1). Näistä kahdesta ensin mainitun voi poistaa käytöstä irrottamalla piirilevyltä jumpperin J4, jolloin se ei kuluta turhaan virtaa.

Lopullisen piirilevyn ensimmäinen kehitysversio toteutettiin aluksi koekytkentälevyllä, joka mahdollisti kohtuullisen vaivattomat muutokset rakenteeseen. Kun piiri lopulta toimi odotetusti, siirrettiin kytkentä tietokoneelle Proteus-ohjelmistoon, ja tämän avulla saatiin suunniteltua piirilevyn rakenne.

Itse fyysinen piirilevy toteutettiin kotiooloissa valotusmenetelmällä käyttäen hyväksi helposti saatavilla olevia välineitä, kuten UV-valoa, natriumhydroksidia, ferrikloridia ja Marinol-100 -polttonestettä.

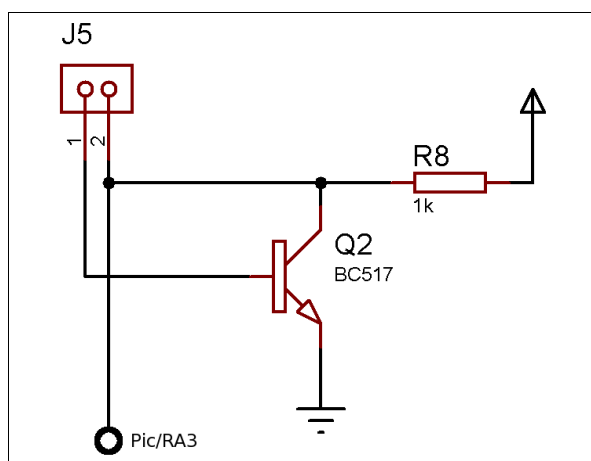
Kuvatun menetelmän etuna on edullisuuden lisäksi lyhyt odotusaika. Siinä missä piirilevyn valmistuttaminen olisi kestänyt useista viikoista jopa kuukauteen, kului toteutuksessa nyt vain muutamia tunteja. Rajoituksena on tosin se, että piirilevyn valmistaminen kaksipuoleisena on erittäin monimutkaista. Tämä ongelma on kuitenkin kierrettävissä siten, että piirilevy suunnitellaan yksipuoleiseksi ja ne vedot joita ei voida reitittää kuparille, toteutetaan hyppylangoilla.

3.2.1 Kosteustunnistin

Kosteustunnistin koostuu kolmesta eri osasta. Itse ”tunnistin” on rakennettu kahdesta 3 cm pituisesta kuparijohtimesta, jotka sijaitsevat 1 cm päässä toisistaan, sekä kaapelista, jolla johtimet kytketään piirilevylle. Lopun rakenteen muodostavat vahvistimena toimiva darlingtontransistori BC517 sekä 1 k Ω ylösvetovastus, jonka tarkoituksena on estää oikosulun muodostuminen ja lisäksi pitää A/D-muuntimen sisääntulolinja ylhäällä, silloin, kun tunnistin ei ole vedessä.

Tunnistimen toiminta perustuu siihen, että kun kuparijohtimet joutuvat veteen, ne muodostavat virtapiirin jonka vastus vesijohtovedellä testattuna on noin 800 k Ω . Vaikka muodostuneen piirin resistanssi onkin melko suuri, kulkee sen läpi silti riittävän suuri virta, jotta transistori saturoituu, ja tällöin puolestaan mittauspisteen jännite putoaa huomattavasti.

Transistori toimii kohtuullisen pienellä virralla siksi, että valitun komponentin vahvistuskerroin on jopa 30 000. Suuren vahvistuskertoimen ongelma tosin on siinä, että se tuo mukaan myös kohinaa. Tässä kyseisessä tapauksessa siitä ei ole haittaa, koska kytkentä toimii lähinnä kytkimen tavoin. /5/

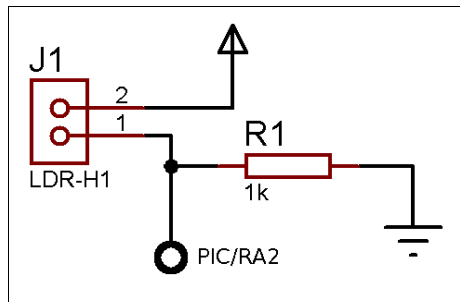


Kuva 16. Kosteustunnistimen kytkentä

3.2.2 Valosensori

Valoisuuden tunnistimen kytkentäkaavio on esitetty kuvassa 17. Suunnittelun lähtökohtana oli kehittää järjestelmä, joka pystyy ilmaisemaan lähinnä vain sen, onko tilassa valoisaa vai ei.

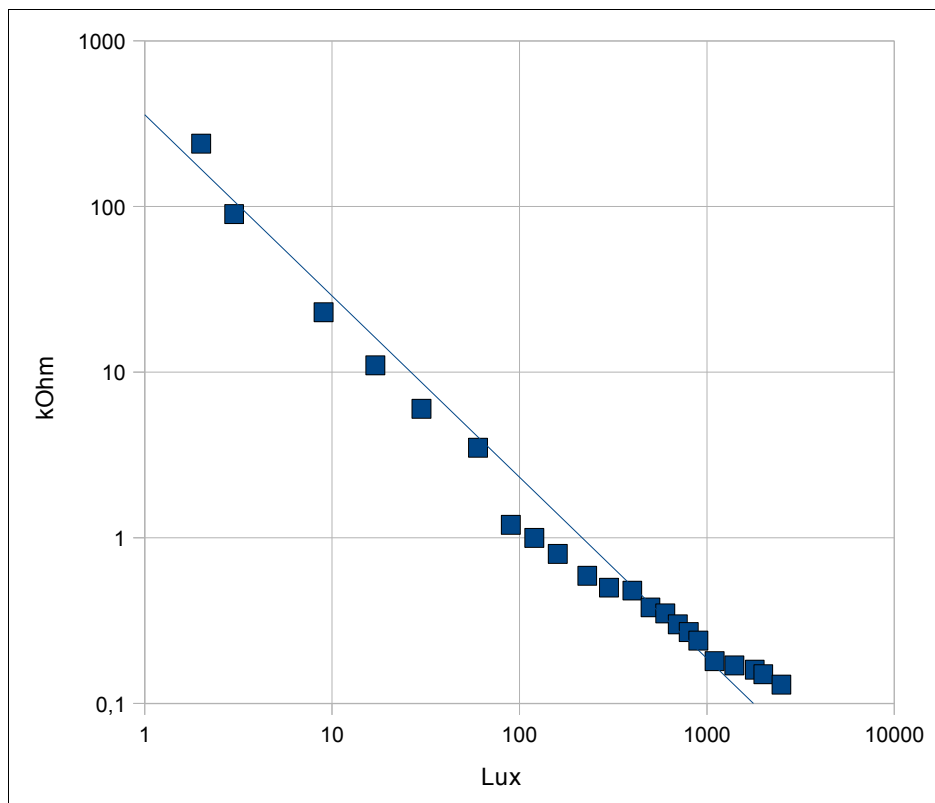
Yksinkertaisin rakenne edellä kuvatulle piirille on jännitteenjako 1 k Ω vastuksen ja LDR-valovastuksen kesken. Varsinainen päätös siitä, mikä luokitellaan valoisaksi ja mikä ei, voidaan myöhemmin tehdä ohjelmallisesti, koska jännitteenjaosta mitattu jännite syötetään mikrokontrollerin 10-bittiseen A/D-muuntimeen ja siitä eteenpäin valvontajärjestelmään.



Kuva 17. LDR-valovastuksen kytkentä

Koska projektissa käytetystä sensorista ainut tunnettu asia on valmistajan ilmoittama noin 10 k Ω keskimääräinen resistanssi 10 Lux:n valossa, on sen tarkempi toiminta-alue (kuva 18) selvitetty himmennettävän 45 W halogeenivalon, valoisuusmittarin ja vastusmittarin avulla.

Valoisuusmittarina on käytetty Mastech MS8229 -yleismittaria ja vastusarvot on mitattu Mastech MY-64 -yleismittarin avulla. Käytetyistä mittalaitteista johtuen tulokset eivät ole kovin tarkkoja, mutta ne antavat kuitenkin hyvän käsityksen siitä, millaisia arvoja komponentilta voi odottaa.



Kuva 18. LDR-valovastuksen vastekäyrä

Sensorin vastekäyrän lisäksi suunnittelussa vaadittiin tietoa siitä, millaisia valoisuusarvoja voidaan odottaa käyttötilanteessa esiintyvän, joten valoisuusmittarilla mitattiin kohdetila sekä ilman valoja että normaalissa valaistuksessa. Valonlähteenä tilassa toimii 11 W energiansäästölamppu ja etäisyys valonlähteeseen oli mittauksessa 1,5 metriä, joka on todennäköisin loppusijoituspaikka sensorille. Mittauksen tulokseksi todettiin, että kohdetilan valoisuus pimennettynä on oletetusti n. 1 Luxin luokkaa ja valaisimen ollessa päällä arvo on n. 15 Luxia.

4 OHJELMISTOJEN TOTEUTUS

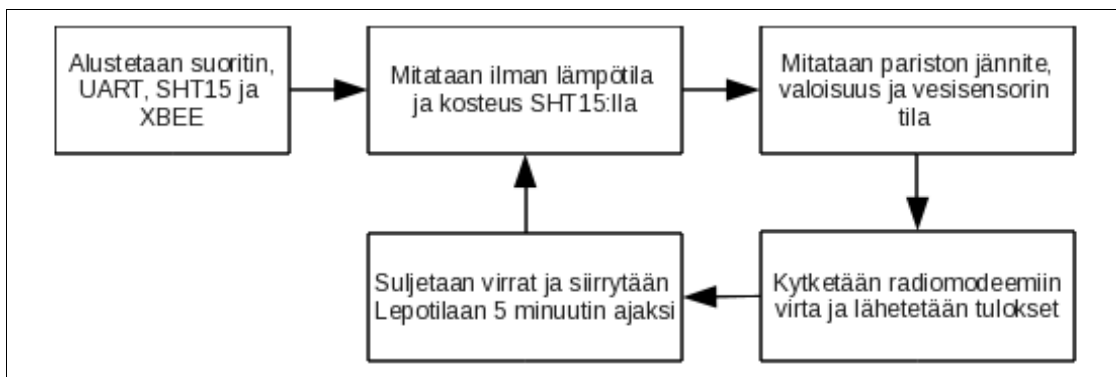
Sekä etäsensorin että vastaanottimen ohjelmistot on toteutettu C-ohjelmointikielellä, joka valittiin projektiin lähinnä siksi, että sen avulla on helppo tehdä nopeita muutoksia ja korjauksia ohjelmistoon.

Vastaanottimen suorituskyvyn kannalta ohjelmointikielen valinnalla ei ole suurtakaan merkitystä, sillä nykytietokoneen laskentakapasiteetti on joka tapauksessa riittävä. PIC-suorittimen tapauksessa C-kääntäjän tuottama ohjelma saattaa joiltain osilta olla hieman hitaampi kuin mitä vastaava assemblyllä kirjoitettu ohjelma olisi. Koska sensorilta ei kuitenkaan vaadita reaaliaikaista tietoa, ei tällä erolla ole väliä.

4.1 Etäsensorin ohjelmisto

Etäsensoria ohjaava ohjelmisto on kehitetty PIC12-, PIC16- ja PIC18-suoritinperheille suunnatulla mikroElektronikan julkaisemalla mikroC-kehitysympäristöllä. Koska kehitysympäristö tarjoaa kääntäjän lisäksi myös yksinkertaisen simulaattorin, on ohjelmistoa voitu osittain testata myös ilman fyysistä suoritinta.

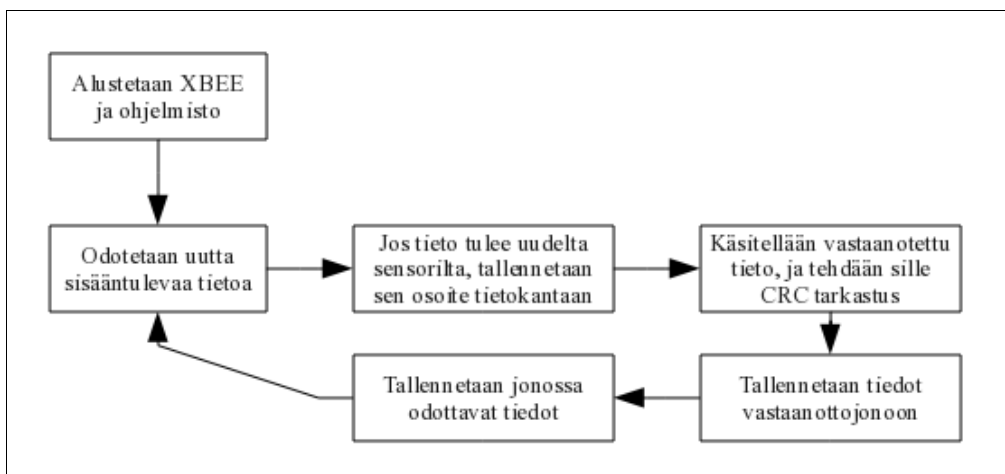
Ohjelmisto käyttää hyväkseen PIC:n sisäistä 7-kanavaista A/D-muunninta, UART-piiriä sekä oskillaattoria. Näiden mainittujen ominaisuuksien ansiosta ohjelmakoodista suurin osa keskittyy radiomodeemin ohjaukseen ja mittaustietojen pakkaukseen siirtoa varten.



Kuva 19. Yksinkertaistettu kaavio lähettimen ohjelmiston toiminnasta

4.2 Vastaanottimen ohjelmisto

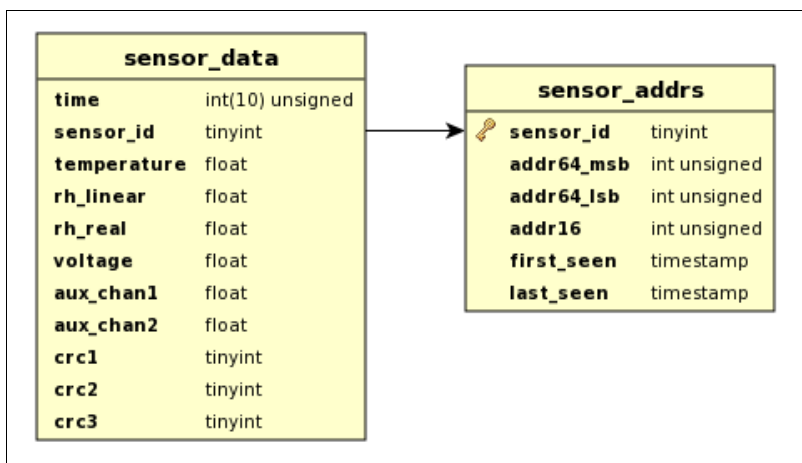
Vastaanottimen ohjelmisto on toteutettu Linux-ympäristöön C-kielellä ja kirjoitushetkellä se sisältää noin 1500 riviä ohjelmakoodia. Ohjelman yksinkertaistettu rakenne on esitetty kuvassa 20, ja kuten siitä on nähtävillä, ohjelman pääasiallinen tehtävä on tällä hetkellä lähinnä uusien sensoreiden tunnistus sekä tiedon tallennus tietokantaan.



Kuva 20. Yksinkertaistettu kaavio vastaanottimen sovelluksen toiminnasta

4.2.1 Tietokannan rakenne

Tietokanta (kuva 21) koostuu kahdesta InnoDB-muodossa olevasta taulusta, joiden rakenteet on selitetty taulukoissa 6 ja 7. ”sensor_addr”-taulussa pidetään kirjaa niistä sensoreista, joilta vastaanotin on saanut viestin ja ”sensor_data”-taulussa on taas tallennettuna varsinaiset viestit sensoreilta.



Kuva 21. Tietokannan rakenne

Taulukko 6. ”sensor_data”-taulun rakenne

Sarake	Sisältö
time	Kellonaika UNIXTIME-muodossa, jolloin lukemat on tallennettu
sensor_id	Numeerinen avain, joka sitoo mittauksien ”sensor_addrs”-taulussa esiintyvän sensorin osoitteeseen
temperature	Mitattu lämpötila
rh_linear	Sensorin ilmoittama lineaarinen ilmankosteus
rh_real	Lämpötilakompensoitu ilmankosteus
voltage	Pariston jännite mittaushetkellä
aux_chan1	A/D-kanava 1:n jännite
aux_chan2	A/D-kanava 2:n jännite
crc1	Lämpötilatiedon CRC8-tarkistuksen lopputulos
crc2	Ilmankosteuden CRC8-tarkistuksen lopputulos
crc3	A/D-kanavien tietojen CRC8-tarkistuksen lopputulos

Taulukko 7. ”sensor_addrs”-taulukon rakenne

Sarake	Sisältö
sensor_id	Numeerinen arvo, joka sensorille annetaan sen lähettäessä ensimmäisen viestinsä vastaanottimelle
addr64_msb	Radiomodeemin 64-bittisen osoitteen MSB-bitit (32 ylintä)
addr64_lsb	Radiomodeemin 64-bittisen osoitteen LSB-bitit (32 alinta)
addr16	Radiomodeemin 16-bittinen osoite
first_seen	Päiväys ja kellonaika, jolloin sensori on ensimmäisen kerran havaittu
last_seen	Päiväys ja kellonaika, jolloin sensori on lähettänyt viimeisimmän viestinsä

4.2.2 Valvontajärjestelmän ohjelmisto

Vastaanotetun tiedon siirto tietokannasta Zabbix-asiakasohjelmistolle tapahtuu yksinkertaisen Bash-skriptin avulla, joka noutaa ja tulostaa tietokannasta viimeisimmän 10 minuutin aikana kirjatun merkinnän. Jos tietoa ei löydy, tulostuu negatiivinen luku, joka on normaalin mitta-alueen ulkopuolella.

5 YHTEENVETO

Laitteen ensimmäinen versio on ollut käytössä lähes puolen vuoden ajan, ja se on toiminut odotetulla tavalla. Ainut ongelma ensimmäisen prototyypin kanssa on ollut akkujen kesto, sillä niissä riittää tehoa enintään kuukaudeksi. Kyseinen ongelma muodostuu pääasiassa siitä, että virtalähteenä käytetyt akut eivät ole uusia, ja lisäksi lähetin lähettää testitilassa neljä kertaa tiheämmin tilatietoja, kuin mitä se tekisi varsinaisessa seurantallassa.

Akun kestoa lukuun ottamatta laitteen toteutuksessa ei esiintynyt suurempia teknisiä ongelmia. Varsinaisena uutena asiana projektissa tuli eteen ZigBee-verkon ja siihen suunnatun XBEE-radiomodeemin käyttö. Radiomodeemin hyvän dokumentoinnin ansiosta tämäkään ei tosin noussut ylivoimaiseksi esteeksi.

Koska laite on osoittautunut toimivaksi valvontatyökaluksi, on sitä tarkoitus jatkokehittää tulevaisuudessa siten, että lähetinpiiriin voi vapaasti valita tarpeeseen sopivat sensorit ja sillä voi myös ohjata yksinkertaisia laitteita. Lisäksi järjestelmään on tarkoitus kehittää itsenäinen vastaanotin, joka kerää sensoreiden tiedot suoraan muistikortille.

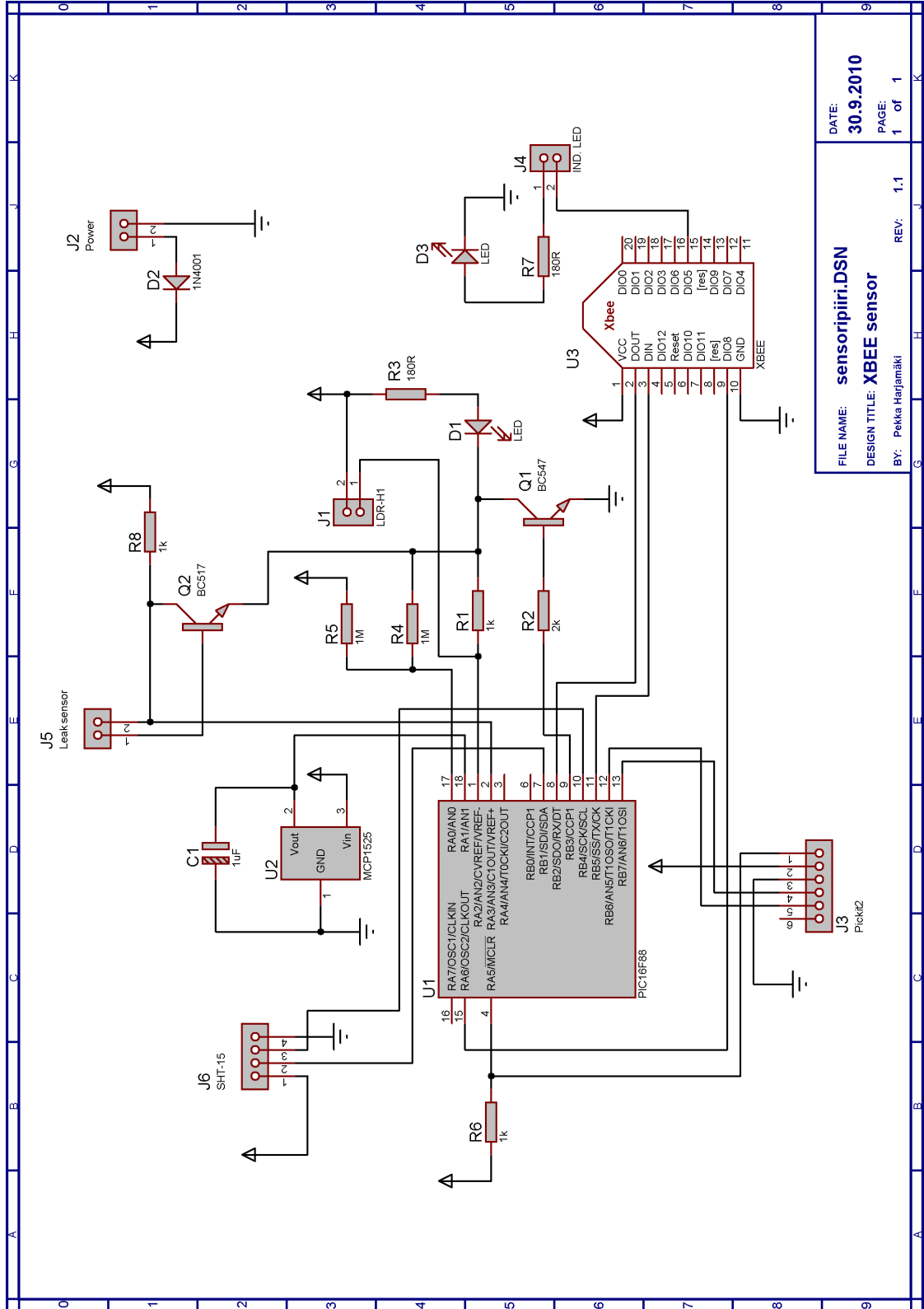
LÄHTEET

1. ZigBee Wireless Standard [www-sivu]. [Viitattu 3.10.2010].
<http://www.digi.com/technology/rf-articles/wireless-zigbee.jsp>
2. ZigBee Alliance Members [www-sivu]. [Viitattu 3.10.2010].
<http://www.zigbee.org/About/OurMembers.aspx>
3. ZigBee Alliance Mission [www-sivu]. [Viitattu 3.10.2010].
<http://www.zigbee.org/About/OurMission.aspx>
4. XBEE product manual [pdf]. [Viitattu 7.10.2010].
http://ftp1.digi.com/support/documentation/90000976_F.pdf
5. BC517-transistorin datalehti [pdf]. [Viitattu 7.10.2010].
<http://www.datasheetcatalog.org/datasheet/motorola/BC517.pdf>
6. SHT15-sensorin datalehti [pdf]. [Viitattu 8.10.2010].
<http://bit.ly/9jhFvP>
7. PIC16F88-mikrokontrollerin datalehti [pdf]. [Viitattu 9.10.2010].
<http://ww1.microchip.com/downloads/en/devicedoc/30487c.pdf>
8. SHT15-sensorin piirikaavio [pdf]. [Viitattu 13.10.2010].
<http://www.sparkfun.com/datasheets/Sensors/Pressure/SHT1x-Breakout-v13.pdf>
9. XBEE Explorer USB -piirikaavio [pdf]. [Viitattu 13.10.2010].
<http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Explorer-v15.pdf>

LIITTEET

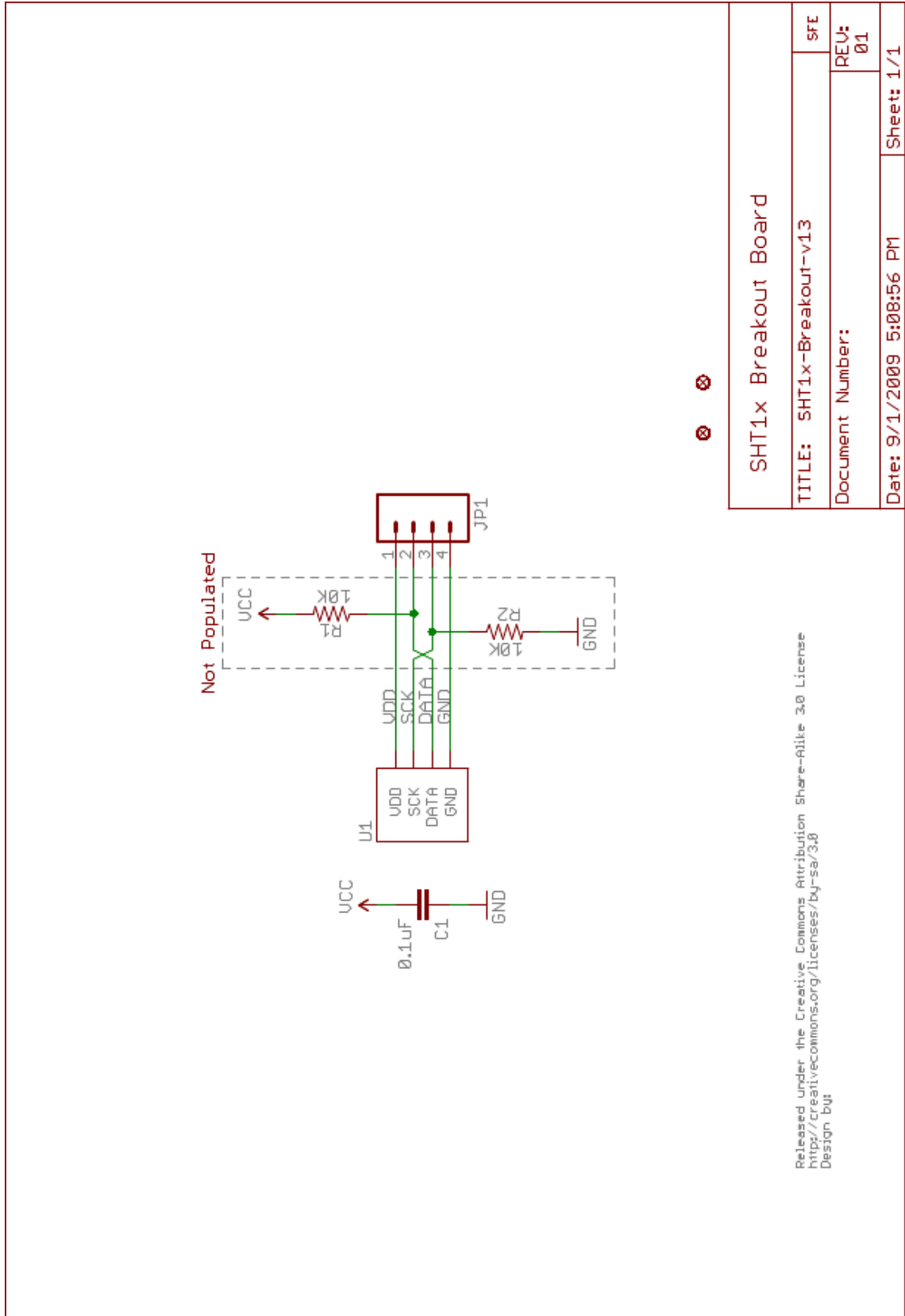
- Liite 1. Sensoripiirin piirikaavio
- Liite 2. SHT15-sensorin piirikaavio /8/
- Liite 3. XBEE Explorer USB -piirin piirikaavio /9/

Liite 1. Sensoripiirin piirikaavio

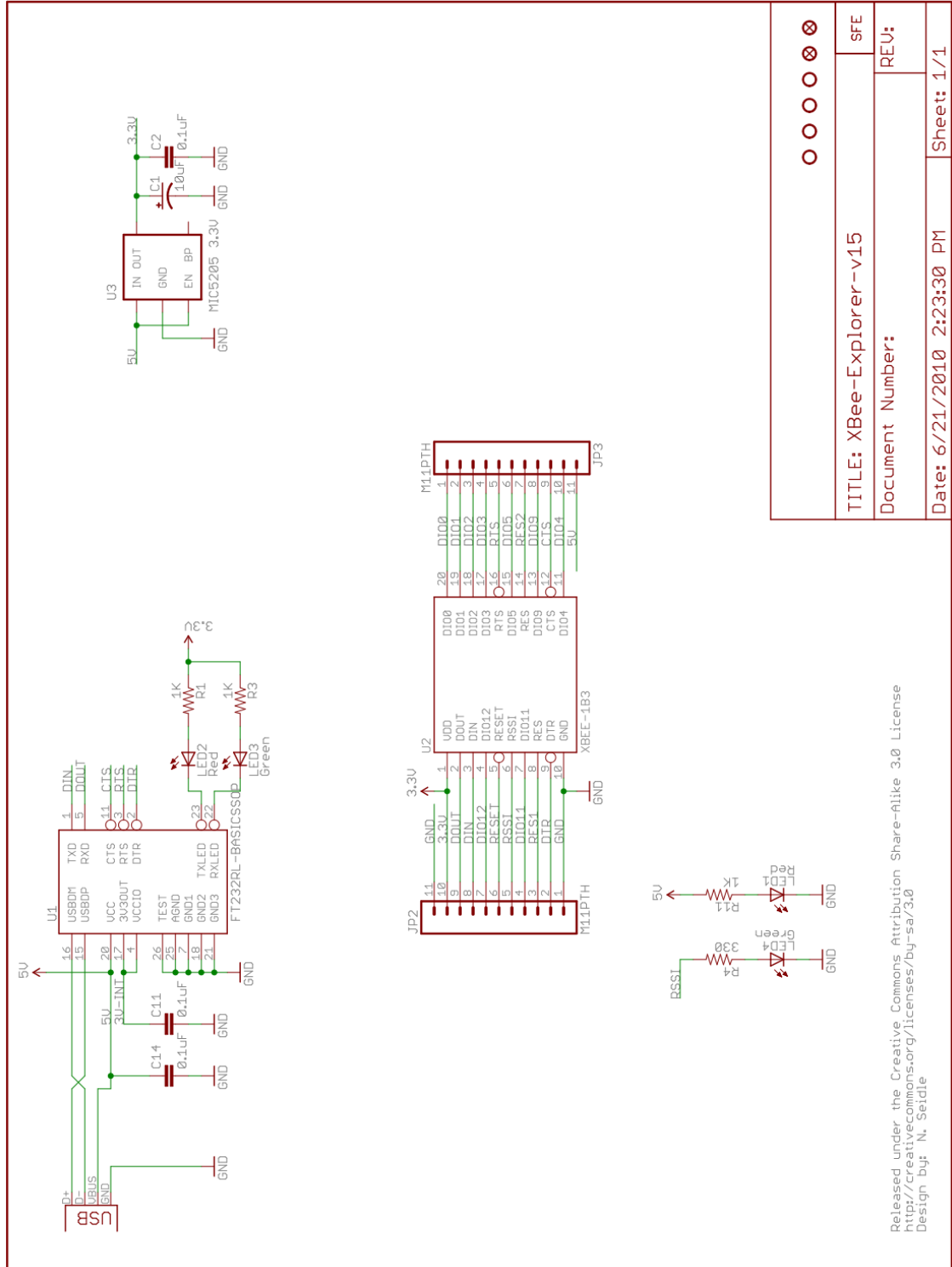


FILE NAME: **sensoripiiri.DSN**
 DESIGN TITLE: **XBEE sensor**
 BY: Pekka Harjamaäki
 REV: 1.1
 DATE: **30.9.2010**
 PAGE: 1 of 1

Liite 2. SHT15 sensorin piirikaavio



Liite 3. ”XBEE Explorer USB”-piirin piirikaavio



○ ○ ○ ○ ○ ⊗

TITLE: XBee-Explorer-v15

Document Number: SFE

REV: REU:

Date: 6/21/2010 2:23:30 PM Sheet: 1/1

Released under the Creative Commons Attribution Share-Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0>
 Design by: N. Seidle