



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Jesse Uusitalo

TIEDONKERUUYKSIKÖN MUODOSTAMINEN

Tekniikka ja liikenne
2011

TIIVISTELMÄ

Tekijä	Jesse Uusitalo
Opinnäytetyön nimi	Tiedonkeruuyksikön muodostaminen
Vuosi	2011
Kieli	suomi
Sivumäärä	37 + 1 liitettä
Ohjaaja	Jukka Matila

Tämä opinnäytetyö käsittelee tiedonkeruuyksikön muodostamista Vaasan ammattikorkeakoululla kehitettyyn kosteusmittausjärjestelmään. Kosteusmittausjärjestelmä on osa Matalaenergiarakentamisen edistäminen kerrostalotuotannossa - hanketta. Tämän työn tuloksena muodostettu tiedonkeruuyksikkö tulee toimimaan Vaasan Suvilahteen rakennetussa kerrostalossa. Tiedonkeruuyksikkö kerää tietoa kosteuden ja lämpötilan mittaussyksiköiltä. Keräämänsä tiedon se tallentaa Vaasan ammattikorkeakoulun MySQL-tietokantaan.

Työssä perehdytään erilaisiin tekniikoihin, joiden avulla tietoa voidaan siirtää. Työn tilaajana on Vaasan ammattikorkeakoulu. Työssä päädyttiin käyttämään avoimia ohjelmia, sillä näin saatiin minimoitua kustannukset. Tiedonkeruuyksikkö toteutettiin käyttämällä Pythonilla ohjelmoituja skriptejä ja ssh-yhteyttä käyttäen.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietotekniikan koulutusohjelma

ABSTRACT

Author	Jesse Uusitalo
Title	Forming a data-collection unit
Year	2011
Language	Finnish
Pages	37 + 1 Appendices
Name of Supervisor	Jukka Matila

This thesis deals with forming a data-collection unit for humidity measuring system, created at Vaasa University of Applied Sciences. Measuring system is a part of a project to encourage apartment-house production to build low-energy apartment houses. The formed data-collection unit will be placed to an apartment-house at Vaasa Suvilahti. Data-collection unit will collect data from temperature- and humidity-measuring units. The collected data will be saved to a MySQL-database located at Vaasa University of Applied Sciences.

This thesis goes through few techniques to collect and transmit data. The client for this project was Vaasa University of Applied Sciences. Thesis was implemented using open source programs, and this minimized the costs of this project. Data-collection unit uses Python-scripts and SSH connection.

Keywords: data-collection unit, DCU, low-energy

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVIOT	6
TAULUKOT	6
1 JOHDANTO	7
2 TAVOITTEET JA ONGELMAT	9
2.1 Tavoitteet	9
2.2 Menetelmät	9
3 LAITEYMPÄRISTÖ JA KÄYTETYT TEKNIIKAT	11
3.1 Mittausjärjestelmä	11
3.2 Modbus	14
3.3 OPC	16
3.4 MySQL	16
3.5 Ubuntu	16
3.6 Windows XP	17
3.7 Python	17
3.8 SSH	17
3.9 RS-485	17
4 TIEDONKERUUYKSIKÖN MUODOSTAMINEN	19
4.1 PC	19
4.2 OPCtechs OPC Modbus Server	21
4.3 OPC Data Logger Pro	22
4.4 Yhteys Vaasan ammattikorkeakoululle	24
4.5 Etäyhteys	24
5 TOTEUTUS	26
5.1 Windows XP -versio	26
5.2 Linux-versio	28
5.2.1 Linux-ohjelmat	28
5.2.2 Ohjelmat ja niiden asennus	28

5.3 Kuvaus järjestelmästä.....	31
5.3.1 Luokkakaavio.....	31
5.3.2 Viestiyhteykskaavio.....	31
5.4 Kuvaus Python-skripteistä.....	32
5.4.1 Current_all.....	32
5.4.2 Moisture_measurement	33
6 TESTAUSSUUNNITELMA	34
7 YHTEENVETO	35
LÄHTEET	36

KUVIOT

Kuvio 1. Kuvitteellinen esimerkki eteläseinästä.	12
Kuvio 2. Kuvaus järjestelmästä. /5/.....	13
Kuvio 3. Kuva mittausyksiköstä. /5/.....	13
Kuvio 4. Modbus-viestin rakenne. /4/.....	14
Kuvio 5. ASCII viestin mallikehys. /10/	15
Kuvio 6. RTU viestin mallikehys. /10/.....	15
Kuvio 7. Intel D945GSEJT. /17/	20
Kuvio 8. SPIRE PowerCube. /15/	20
Kuvio 9. OPCTechsModbus Server.	22
Kuvio 10. OPC Data Logger Pro.....	23
Kuvio 11. Kuvaus RealVNC ohjelmasta. /14/.....	25
Kuvio 12. Luokkakaavio järjestelmästä.	31
Kuvio 13. Viestiyhteyksikaavio järjestelmästä.....	32

TAULUKOT

Taulukko 1. Emolevyjen vertailu.	19
Taulukko 2. OPC Server ohjelmat.	21
Taulukko 3. OPC Logger ohjelmat.	23

1 JOHDANTO

Energiankulutus Suomessa kasvaa, vaikka tavoitteena on kulutuksen vähentäminen. Säästötoimia on useilla toimialoilla jo toteutettu, mutta paljon on vielä tehtävää. Energiankulutuksen vähentäminen onkin nyky-yhteiskunnan suurimpia haasteita. Kulutuksen kasvaessa pyritään jatkuvasti keksimään uusia tapoja kulutuksen vähentämiseksi. Energiatehokkuuden parantaminen tuo taloudellista hyötyä ja vähentää ympäristöä kuormittavia päästöjä.

Rakennusteollisuuden, rakennustoiminnan ja rakennuskannan lämmön- ja sähkönkulutuksen määrä vastaa kansallisen rakennuspoliittisen ohjelman mukaan yli 40 prosenttia tuotetusta primäärienergiasta. /20/ Energiatehokkuuden parantamisen kannalta on rakennus- ja kiinteistöala keskeinen ala Suomessa.

Vaasassa on käynnissä hanke Matalaenergiarakentamisen edistämiseen kerrostalotuotannossa. Vaasan Suvilahteen asuntomessualueen yhteyteen rakennetaan neljä samanlaista kerrostaloa. Ensimmäisen rakentaminen aloitettiin vuonna 2009, toisen 2010, kolmannen 2011 ja neljännen rakentaminen aloitetaan 2012. Kerrostalot rakennuttaa Lakea Oy. Tavoitteena talojen suunnittelussa ja toteutuksessa on energiätehokkuus. /1/

Hankkeessa ovat mukana Pohjanmaan YH rakennuttajat, Peab Oy, Ekolämpö Oy, Vaasan energiainstituutti sekä Tampereen teknillinen yliopisto. Hankkeen koordinaattorina toimii Vaasan ammattikorkeakoulu. /1/

Vaasan ammattikorkeakoulun tietotekniikan ja rakennustekniikan osastot suunnittelivat ja toteuttivat mittausjärjestelmän kosteuden ja lämpötilan jatkuvaan mittaamiseen. Mittausjärjestelmä on osa Matalaenergiarakentamisen edistäminen kerrostalotuotannossa -hanketta. Hankkeen tavoitteena on selvittää paras mahdollinen eristämisen taso ja tutkia kasvaako riski kosteuden tiivistymisen rakenteiden sisällä eristeiden paksuuden lisääntyessä. Hankkeessa selvitetään myös erilaisten seinärakenteiden vaikutus lämpötiloihin ja kosteusarvoihin.

Tämän opinnäytetyön tarkoituksena on toteuttaa tiedonkeruuyksikkö, joka kerää tiedon mittausjärjestelmästä ja siirtää tiedon verkon yli ammattikorkeakoulun

MySQL-tietokantapalvelimelle. Tämä opinnäytetyö tulee olemaan osana tähän asti suurinta rakennuslaboratorioiden ulkopuolella tehtyä mittausjärjestelmää.

2 TAVOITTEET JA ONGELMAT

2.1 Tavoitteet

Vaasan ammattikorkeakoululla suunniteltiin ja toteutettiin lämpötilan ja kosteudenmittausjärjestelmä osana Matalaenergiarakentamisen edistäminen kerrostalotuotannossa -hanketta. Mittausjärjestelmän suunnittelussa ja toteutuksessa olivat mukana tietotekniikan ja rakennustekniikan osastot. Hankkeen olennainen osa oli jatkuva lämpötilan ja kosteuden mittaaminen tietotekniikan osaston tätä varten kehittämällä ja valmistamalla mittausjärjestelmällä. Järjestelmä on suurin rakennuslaboratorioiden ulkopuolella toteutettu mittausjärjestelmä Suomessa.

Hankkeen tavoitteena on selvittää paras mahdollinen eristämisen taso, ja tutkia kasvaako riski kosteuden tiivistymisen rakenteiden sisällä eristeiden paksuuden lisääntyessä. Hankkeessa selvitetään myös erilaisten seinärakenteiden vaikutus lämpötiloihin ja kosteusarvoihin.

Tämän opinnäytetyön tavoitteena on tutkia erilaisia tiedonkeräysohjelmia ja toteuttaa tiedonkeruuyksikkö, joka kerää tiedon mittausjärjestelmästä, ja siirtää tiedon internetin yli Vaasan ammattikorkeakoulun MySQL-tietokantapalvelimelle.

Jokaisessa mittausyksikössä on vähintään kolme kosteuden ja lämpötilan mittaamiseen tarkoitettua anturia. Näiden antureiden avulla mittausyksikkö mittaa kosteuden ja lämpötilan rakennuksen lämpöeristeen sisä- ja ulkopinnoilta. Lisäksi mittausjärjestelmään on lisätty kahdeksan ilmanvaihtokonetta, joilta kerätään tietoa huoneilman lämpötilasta. /6/

Tiedonkeruuyksikön tuli koostua pienestä PC:stä, jossa ei saanut olla tuuletinta sekä ohjelmista, joiden avulla tieto saatiin mittausyksiköltä palvelimelle. Rajoituksia työhön loivat mittausyksiköiden Modbus-protokolla ja koulun palvelimen MySQL-tietokanta.

2.2 Menetelmät

PC:n valinta aloitettiin etsimällä fyysisesti mahdollisimman pientä, mahdollisimman edullista ja ilman tuuletinta toimivaa PC:tä. Tiedonkeruuyksikön tuli koostua

pienestä PC:stä, joka ei mielellään sisältäisi tuuletinta, sillä sen pitää toimia kerrostalon kellarin sähkökaapissa, jossa pöly saattaa tukkia tuulettimen.

Tiedonkeruuyksikön tuli koostua jo olemassa olevista ohjelmista. Mittausjärjestelmän ModBus-protokolla ja koulun MySQL-tietokanta toivat haasteita, sillä tarkoituksena oli löytää mahdollisimman edullinen ohjelmisto ja nämä kaksi ominaisuutta sisältävät ohjelmistot olivat yleisesti melko kalliita. Ohjelmien valinta aloitettiin vertailemalla eri ohjelmia, niiden ominaisuuksia ja hintaa.

3 LAITEYMPÄRISTÖ JA KÄYTETYT TEKNIIKAT

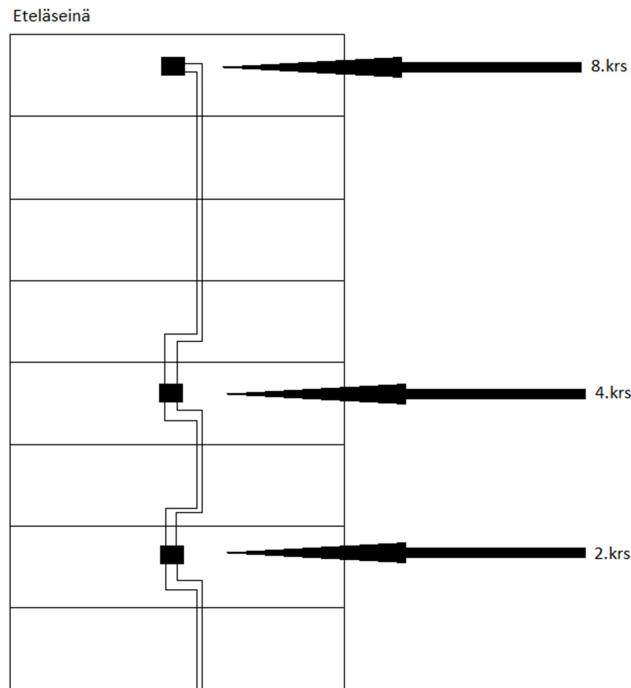
Tässä luvussa käydään läpi mittausjärjestelmän rakennetta, sekä opinnäytetyössä tutkittuja ja käytettyjä tekniikoita. Modbus-protokolla ja MySQL toimivat työn rajaajina.

Tiedonkeruuyksiköstä tehtiin kaksi erilaista kokoonpanoa, joista toisessa käytettiin Windows XP:tä ja toisessa Ubuntu Linuxia. Windows-versiossa käytettiin tiedon siirtoon OPC standardin mukaista ohjelmaa, sillä OPC on todella laajasti käytetty standardi ja se lisäsi laajennettavuuden mahdollisuutta. Linux-versiossa käytettiin tiedonsiirtoon Python-skriptejä. Etäyhteytenä molemmissa versioissa käytettiin SSH:ta. Tiedonsiirtolinja mittausjärjestelmässä oli RS-485 –tekniikan mukainen.

3.1 Mittausjärjestelmä

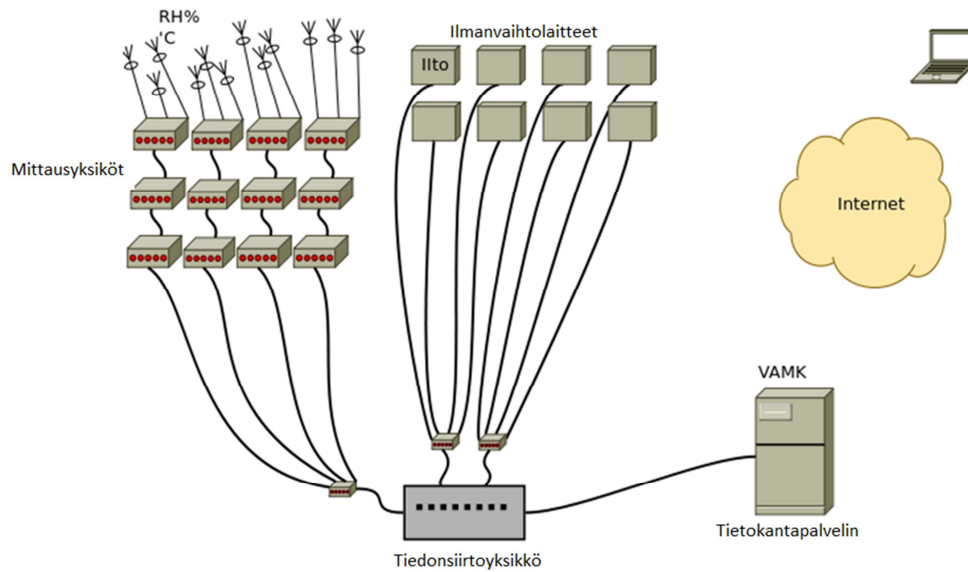
Mittausjärjestelmä on Vaasan ammattikorkeakoulun tietotekniikan osaston lehtori Jukka Matilan sekä opiskelijoiden Hannu Savelan ja Aleksander Sadalan suunnittelema ja toteuttama. Mittausjärjestelmä koostuu 12 mittausyksiköstä. Mittausyksiköt on asennettu kiinteästi rakennukseen.

Mittausyksiköt on sijoitettu toiseen, neljänteen ja kahdeksanteen kerrokseen, jokaiselle pääilmansuunnalle. Mittausyksiköt sijaitsevat asuntojen parvekkeilla, upotettuna seinään, jotta ne eivät olisi häiritsevästi esillä. Yhden ilmansuunnan laitteet ovat kytkettynä saamaan linjaan. Kuviossa 1 on kuvitteellinen esitys kerrostalon eteläpuoleisesta seinästä.



Kuvio 1. Kuvitteellinen esimerkki eteläseinästä.

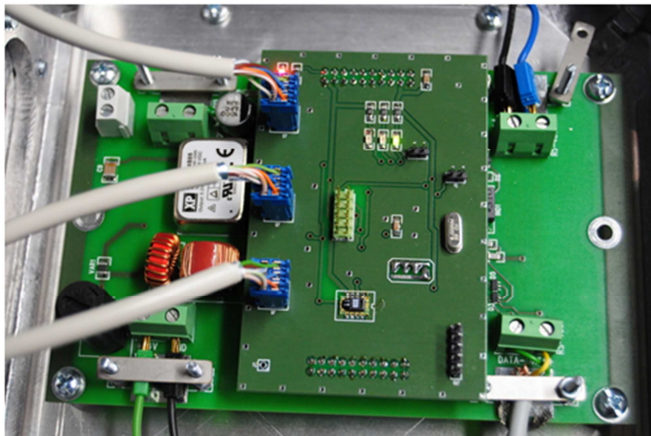
Jokaisessa mittausyksikössä on vähintään kolme kosteuden ja lämpötilan mittaamiseen tarkoitettua anturia. Näiden antureiden avulla mittausyksikkö mittaa kosteuden ja lämpötilan rakennuksen lämpöeristeen sisä- ja ulkopinnoilta. Neljä mittausyksikköä mittaa lisäksi myös ulkoilman lämpötilaa ja kosteutta. Lisäksi mittausjärjestelmään lisättiin loppuvaiheessa kahdeksan ilmanvaihtokonetta, joilta kerätään tietoa huoneilman lämpötilasta. Ilmanvaihtolaitteet on kytketty omaan linjaan ja niiden tiedonsiirtoyksikkö on valmistettu ILTO Oy:llä. /5/



Kuvio 2. Kuvaus järjestelmästä. /5/

Mittausyksikön teknisistä ominaisuuksista kerrotaisiin sen verran, että mikrokontrollerina on ATmega64, ja että tieto liikkuu 1500V isoitua RS485-väylää pitkin. Mittausyksikön tehonsyöttö on 24V ja jatkuva tehonkulutus on 4,8W. /5/

Mittausyksiköt on ohjelmoitu keskustelemaan Modbus ASCII -protokollaa, ja ilmanvaihtolaitteet taas käyttävät Modbus RTU:ta. Modbus ja RS-485 mahdollistavat monen laitteen yhteisen linjan. Kuviossa 3 on kuva mittausyksiköstä.



Kuvio 3. Kuva mittausyksiköstä. /5/

3.2 Modbus

Modbus on Modiconin, nykyisen Schneider Electricin vuonna 1979 esittelemä tiedonsiirtoprotokolla. Modbus-protokolla on viestintärakenne, joka mahdollistaa isäntä-orja/asiakas-palvelin yhteyden älykkäiden laitteiden välille. Se on de facto -standardi, sekä avoin ja laajimmin teollisuudessa käytetty verkkoprotokolla. Sadat eri valmistajat käyttävät sitä tuhansissa eri laitteissa siirtämään tietoa laitteiden välillä. /9/

Modbus:ia käytetään monenlaisissa isäntä-orja -järjestelmissä, monitoroinnissa ja laitteiden ohjaamisessa. Sen avulla voidaan monitoroida esimerkiksi erilaisia mittalaitteita ja antureita käyttäen tietokoneita ja niiden käyttöliittymiä. ModBus:ia ei käytetä ainoastaan teollisuudessa, vaan myös rakennus-, infrastruktuuri-, kuljetus- ja energiasovellukset käyttävät sitä hyväkseen. /9/

Modbus-protokollasta on olemassa sarjaliikenneversiot Modbus/ASCII ja Modbus/RTU sekä ethernet-versio Modbus TCP/IP. Riippumatta versiosta Modbus-viestin perusrakenne pysyy aina samana. /4/

Field	Description
Device address	Address of the receiver
Function code	Code defining message type
Data	Data block with additional information
Error check	Numeric check value to test for communication errors

Kuvio 4. Modbus-viestin rakenne. /4/

Kun laitteet on ohjelmoitu kommunikoidaan Modbus-verkossa käyttäen ASCII-siirtoilaa, lähetetään jokainen 8-bitin tavu kahtena ASCII-merkinä. Viestit siirretään heksadesimaalimuodossa (0-9, A-F), jossa yksi heksadesimaali vastaa yhtä ASCII merkkiä. Modbus ASCII -siirtoilman suurin etu on se, että se sallii jopa sekunnin viiveen merkkien välillä. /10/

Modbus ASCII -viesti alkaa kaksoispisteellä, ja se päättyy rivinalkuunpalautusmerkkiin sekä rivinvaihtomerkkiin. Modbus-verkkoon kytketyt laitteet monitoroivat verkkoa jatkuvasti odottaen kaksoispistemerkkiä, jonka jälkeen ne siirtyvät viestikehyksen seuraavaan osaan selvittämään, vastaako osoitetieto sen omaa

osoitetta. ASCII-siirtoilassa käytetään LRC (Longitudinal Redundancy Check) -virheentarkistusta. Kuvassa on ASCII -mallikehys.

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	n CHARS	2 CHARS	2 CHARS CRLF

Kuvio 5. ASCII viestin mallikehys. /10/

Kun laitteet on ohjelmoitu keskustelemaan Modbus-verkossa käyttäen RTU (Remote Terminal Unit) -siirtotietä, jokainen 8-bittinen tavu viestissä sisältää kaksi heksadesimaalimerkkiä. Tämän siirtotien etu on se, että sen suurempi merkkien tiheys sallii paremman datanläpimenoarvon kuin ASCII -siirtoila samalla baudinopeudella. Heikompaa on se, että koko viestin täytyy mennä läpi jatkuvana virtana. /10/

Modbus RTU viesti alkaa ja loppuu ns. hiljaisella aikavälillä. Hiljainen aikaväli on vähintään 3,5 merkin pituinen. Ensimmäinen lähetettävä viestin osa on osoite. Viestit siirretään heksadesimaalimuodossa (0-9, A-F). Laitteet monitoroivat verkkoa jatkuvasti myös hiljaisten aikavälien aikana. Ensimmäisen viestinosan (osoite) saavuttua laitteet purkavat viestin tarkastaakseen, mille osoitteelle se on osoitettu. Koko viesti täytyy lähettää yhtenä virtana, sillä jos viestin keskellä on 1.5 merkin hiljainen aikaväli, se tuhoataan ja laite alkaa odottaa uutta osoitetta. Samankaltainen tapahtuma seuraa siitä, jos viesti alkaa ennen kuin 3.5 merkin pituinen hiljainen aikaväli ei täyty, ja viesti alkaa ennaikaisesti. Kuviossa (6) on tyypillinen RTU viestikehys. RTU siirtoilassa käytetään CRC(Cyclic Redundancy Check). /10/

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	n x 8 BITS	16 BITS	T1-T2-T3-T4

Kuvio 6. RTU viestin mallikehys. /10/

3.3 OPC

OPC on avoin tiedonsiirtostandardi. Se on teollisuuden automaatio-sovelluksissa laajasti käytetty tiedonsiirron standardi. OPC on sarja standardisoituja määrittämiä. Ensimmäinen standardi, nimeltään OPC Specification (nykyisin Data Access Specification) syntyi Microsoftin ja monen johtavan automaatio-toimittajien yhteistyöstä. Alunperin se pohjautuu Microsoftin OLE COM (component object model) ja DCOM (distributed component object model) -teknologioihin. OPC Specificationin avulla määriteltiin standardisoidut objektit, rajapinnat ja menetelmät tuotannon automaatio-ohjelmiin, tarkoituksena helpottaa prosessien yhtenäistä hallintaa. /11/

3.4 MySQL

MySQL on relaatiotietokantojen hallintajärjestelmä. Se varastoi tietokannat ja niiden sisältämät taulut, sarakkeet sekä rivit, ja käsittelee niitä yhtenä kokonaisuutena. Ohjelman avulla käyttäjille annetaan eritasoisia oikeuksia. MySQL kirjoittaa myös lokia käyttäjien toimista ja hoitaa vastaukset tietokantakyselyihin. Relaatiotietokanta on lyhyesti sanottuna joukko yhteen liitettyjä tauluja. Taulut koostuvat sarakkeista ja riveistä. Taulut ovat tiettyyn sarakkeeseen merkittyjen arvojen pohjalta suhteessa toisiinsa. MySQL on maailman suosituin avoimen lähdekoodin tietokantaohjelmisto. Ohjelmiston valtteja ovat nopeus, luotettavuus ja helppokäyttöisyys. MySQL:ää käyttävät niin yksityiset kuin yrityksetkin. /7/

3.5 Ubuntu

Ubuntu on avoimesta lähdekoodista koostuva Linux-käyttöjärjestelmä. Se on alunperin rakennettu Debian-projektin tekemälle työlle. Ubuntu sisältää peruskäyttöön tarvittavat ohjelmat, ja ohjelmia on helppo asentaa itse lisää. Ubuntu on maksuton ja sen asentaminen on helppoa. Ubuntu julkaistaan puolen vuoden välein uusi versio, jota tuetaan vähintään 18kk. Ubuntu LTS (Long-Term Support) -versioita tuetaan kolmesta viiteen vuoteen. /16/

3.6 Windows XP

Windows XP on Microsoftin vuonna 2003 julkaisema käyttöjärjestelmä. Windows XP on vuodesta 2004 lähtien ollut suosituin käyttöjärjestelmä. Se on saanut jo kaksi seuraajaa, mutta se on silti näitä kahta uudempaa yleisempi. /18/ Microsoft on ilmoittanut, että Windows XP:n tuki loppuu 8.4.2014. /8/

3.7 Python

Vuonna 1991 julkaistu Python on avoin, yksinkertainen ja helposti opittava yleiskäyttöinen olioskriptikieli. Python ohjelmointi perustuu kirjastojen käyttöön, ja kieli itsessään on hyvin pieni. Pythonin erikoisuus on pakolliset sisennykset, jotka ilmoittavat esimerkiksi, mitkä lauseet kuuluvat mihinkin kontrollirakenteeseen. Sisennykset helpottavat ohjelmien luettavuutta. /2/

3.8 SSH

Secure Shell (SSH) on protokolla salatulle etäkirjautumiselle ja muille salatuille verkkopalveluille, jotka toimivat suojaamattomassa verkossa. Se on tarkoitettu yleiskäyttöiseksi käyttäjän autentikointi protokollaksi. /19/ Se mahdollistaa salatun terminaalilyhteyden ja turvallisen etäkäytön. SSH-asiakasohjelmalla otetaan yhteys SSH-palvelinohjelmaan. Näiden välille muodostuu turvallinen salattu tunneli. /13/

3.9 RS-485

RS-485 on balansoitu tiedonsiirtoliitäntä, jossa signaali kytketään kahdella johtimella. Signaalin tila määräytyy näiden kahden välisestä jännite-erosta. Siirtolinja pitää päättää päätevastuksilla, koska balansoiduissa siirtoliitännöissä käytetyissä suurissa tiedonsiirtonopeuksissa signaalin heijastuminen aiheuttaisi muuten ongelmia. Päätevastuksen resistanssin täytyy vastata siirtojohdon impedanssia. RS-485 -sarjaliitännässä voidaan kytkeä yhteensä 32 lähetin-vastaanotinpiiriä saman kahden johtimen muodostamaan verkkoon. 32:stä vain yksi kerrallaan voi olla aktiivisena, muiden ollessa suurimpedanssisessa tilassa. Lähetin voidaan kytkeä tai poistaa linjalta DE-signaalilla. Yleensä kaikki laitteet kuuntelevat linjaa koko ajan. RS-485 -sarjaliitännässä voi tiedonsiirtonopeus olla yli 10Mbs, maksimino-

peus riippuu mm. lähetin-vastaanotinpiireistä. Tiedonsiirtojohdin RS485:ssä voi olla 1200m. /3/

4 TIEDONKERUUYKSIKÖN MUODOSTAMINEN

Tässä luvussa kerrotaan tiedonkeruuyksikön muodostamisesta sekä laitteiston ja ohjelmien valitsemisesta. Käydään läpi myös sitä, kuinka päädyttiin valittuihin tuotteisiin.

Tutkiessa erilaisia tekniikoita, päädyttiin OPC-standardin mukaisiin ohjelmiin, sillä OPC on teollisuudessa todella laajasti käytetty, ja sen avulla järjestelmä oli helposti laajennettavissa. OPC-ohjelmien pääpaino oli kuitenkin yleensä monitoroinnissa, ja tietokantatallennus vain valinnainen lisäosa, minkä vuoksi ohjelmistokokonaisuudet olivat erittäin arvokkaita. Päädyttiin etsimään kahta eri ohjelmaa, yhtä tiedon hakemiseen laitteilta, ja toista tiedon siirtämiseen tietokantaan.

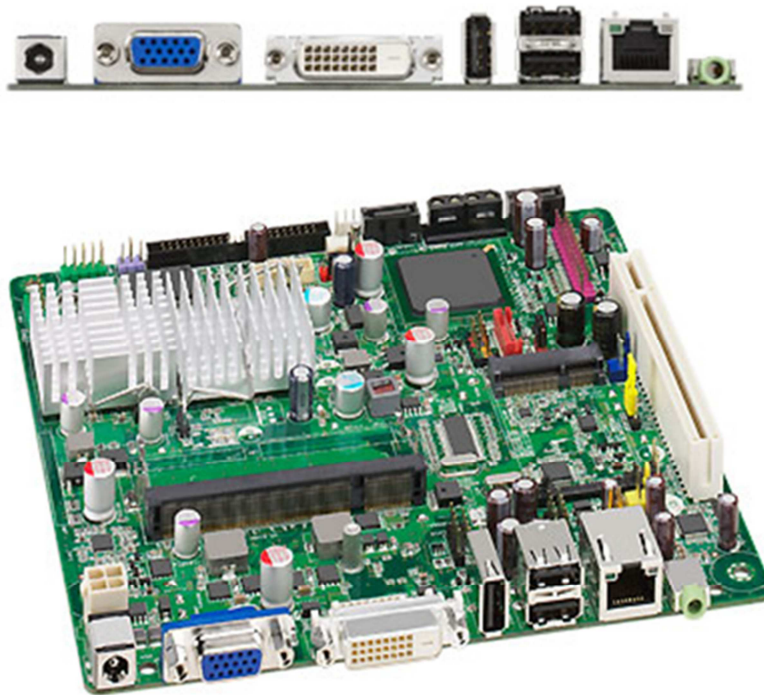
4.1 PC

Sopivan emolevyn etsiminen aloitettiin verkkokauppojen sivuilta, emolevyjen hintaa ja suorituskykyä vertailemalla. Vaatimuksia olivat passiivinen jäähdytys ja edullinen hinta. Hetken etsinnän jälkeen löytyi Intel D945GSEJT mini ITX -emolevy.

Taulukko 1. Emolevyjen vertailu.

Emolevy	Virtalähde	Proessori	Koko	Hinta
Intel D945GSEJT	12V single supply	Intel Atom N270, 1.6GHz	170mm X 170mm	105-125€
JNF92-270-LF	erillinen miniITX power supply	Intel Atom N270, 1.6GHz	170mm X 170mm	190-260€
fit-PC2i	12V single supply	Intel Atom Z510 1.1GHz	101x115x27 mm	270 €

Intel D945GSEJT mini ITX:n koko oli fyysisesti 170mm x 170mm, ja sen suorituskyky vastasi peruskäyttöön tarkoitettua tietokonetta. Valinnan vahvasti emolevyn hinta ja sen toimiminen 12V virtalähteellä. Se ei siis vaatinut erillistä virtalähdettä.



Kuvio 7. Intel D945GSEJT. /17/

Emolevy ostettiin Insinööritoimisto Timo Västi Ky:stä, ja sen mukana saatiin kotelo, kovalevy, hiiri ja näppäimistö. Kotelo oli mallia SPIRE PowerCube M - SP2012B-120W-EU.



Kuvio 8. SPIRE PowerCube. /15/

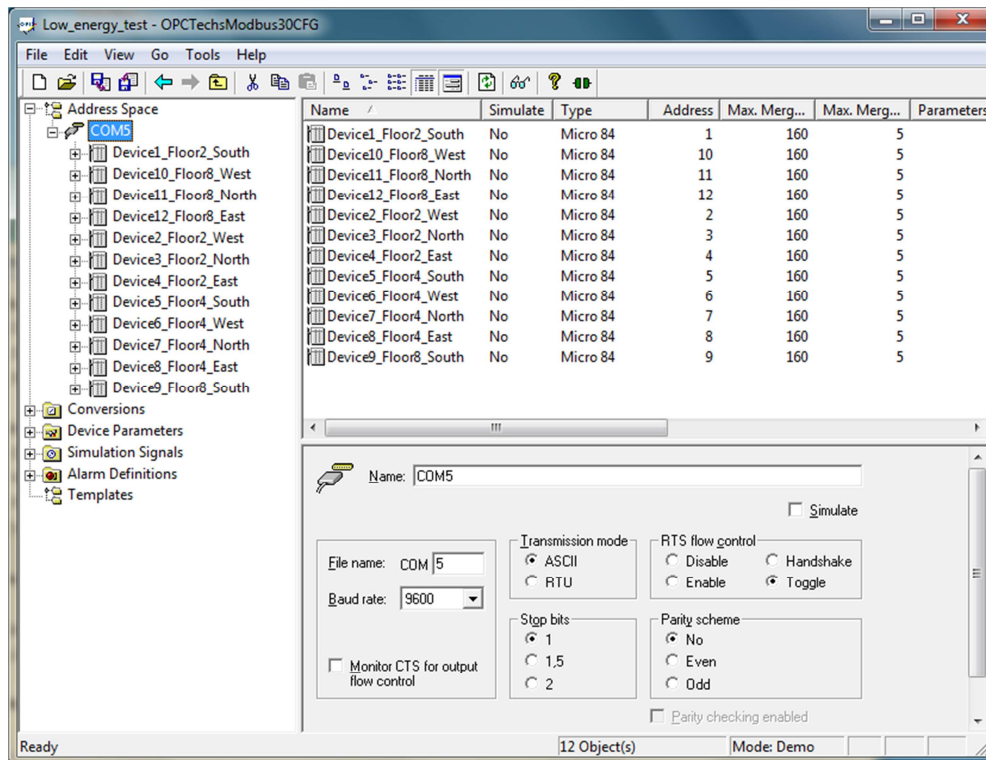
4.2 OPCtechs OPC Modbus Server

Vertailtaessa erilaisia OPC ModBus -servereitä osoittautuivat ne yleensä todella kalliiksi. Hyvin yleistä oli myös, että ohjelmien ominaisuudet olivat erittäin laajat ,eikä kovin rajatuilla ominaisuuksilla varustettua ohjelmaa löytynytäkään. Hintojen vertailun lisäksi asennettiin valituista ohjelmista demoversio tietokoneelle. Demoversion avulla oli mahdollista tutkia ohjelman käytettävyyttä ja ominaisuuksien toimintaa.

Taulukko 2. OPC Server ohjelmat.

Tarjoaja	Hinta
Kepware	795.00 USD
MBX	795.00 USD
Imperioustech	599.00 USD
Matrikon	500.00 USD
OPC Systems	495.00 USD
OPC Techs	368.00 USD

Ohjelmista valikoitui OPCTechs -yhtiön OPC Modbus Server -niminen ohjelma. Se oli hinnaltaan edullisin ja sen käytettävyys oli vaivatonta. Lisäksi ohjelman demoversion ainoa rajoite oli kahden tunnin käyttö kerrallaan, mikä tarkoitti, että kun kaksi tuntia oli kulunut, ohjelma vain lopetti tiedon hakemisen. Ohjelma kuitenkin toimi taas seuraavalla tietokoneen käynnistyskerralla kaksi tuntia. Tämä mahdollisti kokonaisen tiedonkeruuyksikön kokoonpanon testauksen ennen ohjelmien ostamista.



Kuvio 9. OPCTechsModbus Server.

OPC Modbus server on OPCtechsin vuonna 2005 kehittämä OPC 3.0 standardin mukainen OPC -serveriohjelma. Sen avulla voidaan hakea tietoa kaikilta laitteilta, jotka tukevat Modbus ASCII tai RTU -protokollaa, ja jakaa sitä OPC-tagina taas eteenpäin.

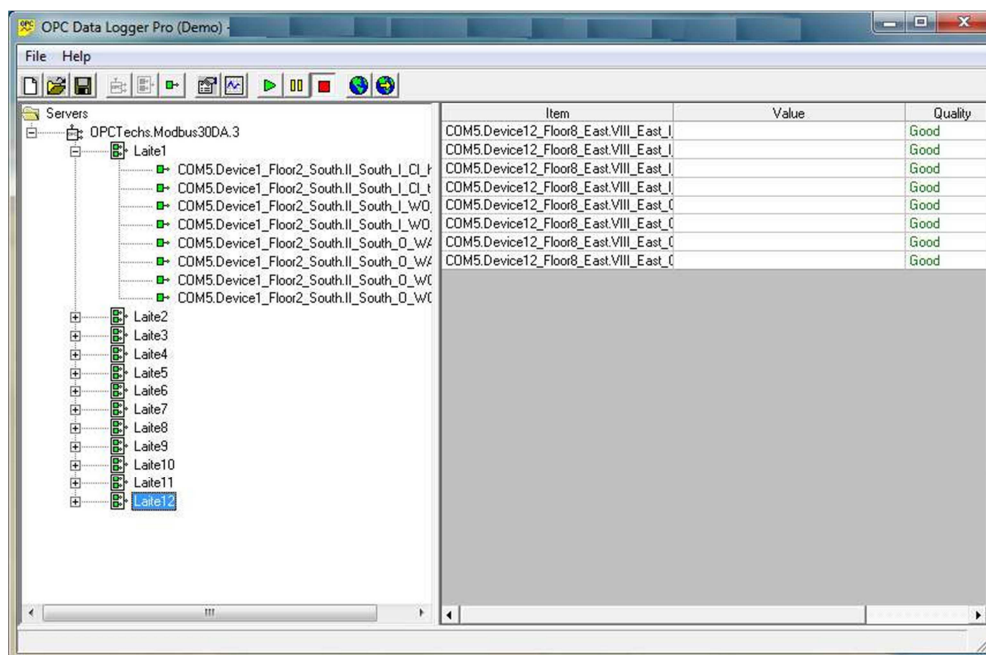
4.3 OPC Data Logger Pro

OPC-tietokantatalennusohjelman etsintä aloitettiin vertailemalla ohjelmien hintaa ja OPC-tagien määrää. Huomiota kiinnitettiin myös mahdollisiin ns. turhiin ominaisuuksiin, sillä ohjelman tuli olla myös mahdollisimman vähän tietokoneen suorituskykyä rasittava, jotta välttyttäisiin turhilta ongelmilta. Koska hinnan piti olla mahdollisimman alhainen, karsi tämä pois monta mahdollista vaihtoehtoa.

Taulukko 3. OPC Logger ohjelmat.

Tuote	Tarjoaja	Hinta	Tagi raja	Demo
MySQL DataBase Logging	www.softwaretoolbox.com	759.00 USD	rajaton	ei testattu
OPC Data Logger Pro	www.opclogger.com	299.95 USD	rajaton	toimi
OPC Database.NET	www.opcsystems.com	295.00 USD	250	ei toiminut

Rajattuani vaihtoehdot aloitin ohjelmien testaamisen ilmaisten testiversioiden avulla. Ongelmia tuotti se, että kaikissa muissa ohjelmissa paitsi OPC Data Logger Prossa oli testiversiosta poistettu mahdollisuus tallentaa tietoa tietokantaan. Juuri tärkein ominaisuus jäi siis monesta ohjelmasta näkemättä. OPC Data Logger Pro -ohjelman testiversion käyttö oli rajattu ajallisesti. Se tallensi tietoa puolen tunnin ajan, jonka jälkeen se vain lopetti tallennuksen. Päädyin OPC Data Logger Pro -ohjelmaan, sillä sen OPC-tagien määrää ei ollut rajattu, se oli helppokäyttöinen, ja kykenin testaamaan sen kaikki ominaisuudet jo testiversiolla.

**Kuvio 10.** OPC Data Logger Pro.

OPC Data Logger on tiedonkeräysohjelma, joka voidaan liittää mihin tahansa OPC serveriin ja kirjoittaa haettu tieto CSV-tiedostoihin, joita voidaan myöhem-

min esittää mm. Excel, Access ym. ohjelmilla. Tiedon kirjoittaminen voidaan sitoa aikaan, tiedon muutoksiin tai asettamalla erilaisia liipaisimia. Tiedon tallennus tapahtuu joko kirjoittamalla yli tai lisäämällä tietoa tiedostoon. Ohjelmalla voi tallentaa tietoa CSV-tiedostoksi, Microsoft SQL -serveriin, tai MySQL-serveriin. OPC Data Logger-ohjelman avulla saa luettavasta datasta myös muodostettua erilaisia taulukoita. Ohjelma osaa muodostaa yhteyden automaattisesti tahattoman sammumisen jälkeen. /12/

4.4 Yhteys Vaasan ammattikorkeakoululle

Tietokantayhteyden ollessa mahdollinen vain Vaasan ammattikorkeakoulun sisäverkosta, täytyi ensin muodostaa yhteys Vaasan ammattikorkeakoululle. Työssä päädyttiin käyttämään SSH-tunnelia, sen luotettavuuden ja Vaasan ammattikorkeakoulun valmiin SSH-palvelimen vuoksi. Yhteyden avaamista varten kirjoitettiin pieni skripti, jonka avulla Putty-ohjelma muodosti tietokoneen käynnistyessä SSH-tunnelin Vaasan ammattikorkeakoululle. Putty on ilmainen SSH- ja telnet-asiakasohjelma Windowsille. Putty on avoimen lähdekoodin ohjelma, ja sitä kehitetään ja tuetaan vapaaehtoisuutena.

4.5 Etäyhteys

Etäyhteys tiedonkeruuyksikköön oli myös tärkeä mahdollisten vikatilanteiden tutkimisen kannalta. Etäyhteysohjelmaksi valittiin RealVNC Server -niminen ohjelma, jonka avulla saatiin etätyöpöytäyhteys tiedonkeruuyksikköön. Etätyöpöytäyhteys tiedonkeruuyksikköön oli välttämätön, sillä Windows XP:n ja valittujen ohjelmien hallinta oli toteutettu graafisin työkaluin.

RealVNC on etähallintaohjelma, joka sallii tietokoneen, johon on asennettu RealVNC Server -ohjelma, työpöydän katselun ja hallinnan etäyhteydellä. Katse-
lua ja hallintaa varten tarvitaan RealVNC Viewer -niminen ohjelma. RealVNC on alustariippumaton eli Windows tietokoneesta voi ottaa yhteyden Mac- tai Linux-tietokoneeseen. /14/



Kuvio 11. Kuvaus RealVNC ohjelmasta. /14/

5 TOTEUTUS

Tässä luvussa kuvaillaan tiedonkeruuyksikön kokoonpanemista ja ensimmäisen version testausta. Johtuen monesta vastoinkäymisestä ja ongelmasta tiedonkeruuyksikön muodostaminen aloitettiin uudestaan, ja tämä toinen versio käydään myös läpi tässä luvussa.

5.1 Windows XP -versio

Kun tiedonkeruuyksikön kokoonpano oli valmisteltu, aloitettiin testaaminen. PC -laitteistona oli Intel D945GSEJT mini ITX -emolevy, 250GB kovalevy, ja siinä käyttöjärjestelmänä Windows XP Professional. Tiedon hakemiseen mittausyksiköiltä käytettiin OPCTechsin OPC Modbus Serveriä. Siihen oli määritetty kaikki mittausyksiköt ja niiden eri antureiden lämpötila- ja kosteusarvot erillisiksi OPC-tageiksi. OPC-tageja oli yhteensä 80. Tiedonsiirto tietokantaan tapahtui OPC Data Logger Pro:lla, johon oli myös määritetty kaikki OPC-tagit. Etätyöpöytäyhteyden tiedonkeruuyksikköön muodostettiin RealVNC:tä käyttäen. Etätyöpöytäyhteys oli tärkeä, sillä tiedonkeruuyksikön fyysinen sijainti oli ahdas ja koska tiedonkeruuyksikköä jouduttiin testivaiheessa useasti konfiguroimaan.

Testivaiheessa ilmanvaihtokoneita ei ollut vielä määritetty, sillä tieto ilmanvaihtokoneiden lisäämisestä saatiin vasta, kun ohjelmat oli jo valittu ja testikokoonpano asennettu. Kyse ei kuitenkaan ollut kuin uusien tagien lisäämisestä. Testaamista jatkettiin siis toistaiseksi ilman ilmanvaihtokoneita.

Testin ensimmäisessä vaiheessa ei tallennettavien tagien nimiä muutettu tietokantatallennusta varten OPC Data Logger Pro -ohjelmaan, vaan ohjelman annettiin tallentaa tietoa testikantaan tagin koko nimellä, joka sisälsi serverin-, mittausyksikön- ja anturinnimen sekä merkinnän, oliko kyse lämpötilasta vai kosteudesta. Näin ollen tallennettavan tagin nimestä muodostui todella pitkä ja se sisälsi pisteitä, jotka eivät ole toivottuja tietokantojen yhteydessä. Ohjelma kuitenkin toimi odotetusti. Ongelmia aiheutti kuitenkin tiedon hakeminen mittausyksiköiltä, sillä testiohjelman puolen tunnin aikarajan johdosta tahdottiin mahdollisimman monta lukemaa puolen tunnin aikana, mutta lyhennettäessä loggausaikaa yhteen minuut-

tiin, meni Modbus-linja tukkoon. Laitteet eivät vastanneet ja tämän johdosta paljastui OPC Data Logger Prosta eräs kyseenalainen ominaisuus. Ohjelma alkoi kirjoittaa vain arvoa 0 (nolla) tietokantaan, oikean vastauksen ollessa esimerkiksi 6520. Tehtiin kuitenkin päätös, että oikeiden Modbus-viestien ollessa neljän ASCII-merkin viestejä, erottuisivat nollat joukosta niin, että ne voitaisiin karsia pois jälkeensä.

Testin toisessa vaiheessa myös ilmanvaihtokoneet lisättiin mukaan ohjelmiin. Ilmanvaihtokoneiden lisäyksen jälkeen oli OPC Modbus Serverin tagimäärä kasvanut kahdeksastakymmenestä (80) sataankolmeen kymmeneenkuuteen (136). Niitä lisättäessä OPC Data Loggeriin, ei jostain syystä pystytty lisäämään kaikkia ilmanvaihtokoneiden tageja. Asiaa tukittaessa huomattiin osan ilmanvaihtokoneiden luettavista tiedoista olevan vain lukuoikeudet omaavissa rekistereissä, ja OPC Modbus Server ei suostunut jakamaan tätä tietoa eteenpäin. Testiä päätettiin kumminkin jatkaa ilman muutamaa tagia, sillä tärkeämpiä olivat kosteuden ja lämpötilanmittausyksiköt.

Valmisteluiden jälkeen, asetettiin ohjelmat käynnistymään automaattisesti, ja tiedonkeruuyksikkö käynnistettiin. PC:n suorituskyvyn ollessa rajallinen, kesti OPC Data Loggerin käynnistyminen noin kymmenen minuuttia. Ohjelma kirjoitti muutamana oikean arvon tietokantaan, jonka jälkeen se alkoi taas kirjoittaa nollaa. Sama toistui jokaisella kerralla, kun tiedonkeruuyksikkö käynnistettiin. Päätimme kumminkin jatkaa testaamista.

Testin kolmannessa vaiheessa luotiin uusi testitietokanta ja aloitettiin nimeämään OPC Data Loggerin tageja uudelleen, tietokantaan sopivilla nimillä. Tässä vaiheessa OPC Data Logger -ohjelmasta alkoi paljastua enemmän ongelmia. Uudelleen nimettyjen tagien tiedot eivät aina tallentuneet tietokantaan, ja nimiä muutettaessa OPC Data Logger kaatui aina välillä.

Ongelmista keskusteltiin Vaasan ammattikorkeakoulun lehtori Jukka Matilan kanssa, joka toimi työn valvojana ja työn tilaajana. Kohdatut ongelmat ja mahdolliset tulevat ongelmat käytiin läpi sekä keskusteltiin ohjelmien epävakauksesta. Keskustelun jälkeen tultiin siihen tulokseen, että on aloitettava tiedonkeruuyksi-

kön ohjelmapuolen kokoaminen uudelleen. Lehtori Jukka Matila oli ohjelmoinut testimielessä Python-ohjelmointikielellä muutaman yksinkertaisen skriptin, joilla sai haettua tietoa mittausyksiköiltä. Skriptien osoittautuessa toimiviksi, päätettiin, että tiedonkeruuyksikkö muodostetaan niiden ympärille.

5.2 Linux-versio

Alkuperäisen kokoonpanon epäonnistuessa, päätettiin vaihtaa lähestymistapaa ja siirtyä avoimiin ohjelmiin. Myös ohjelmien keskeisin määrittäjä eli käyttöjärjestelmä vaihdettiin avoimeen ja ilmaiseen. Käyttöjärjestelmäksi valittiin Ubuntu 10.04LTS Linux.

5.2.1 Linux-ohjelmat

Työssä päädyttiin käyttämään Ubuntu 10.04LTS Linuxia sen avoimuuden, laajan levinneisyyden, yleisen käytön ja pitkäkestoisen tuen vuoksi. Ubuntu avulla tietokoneen suorituskyky ei ollut niin kovalla rasituksella. Ubuntu 10.04LTS sisältää valmiiksi tuen Pythonia varten, Windows XP:lle joutuu asentamaan erillisen tulkin tätä varten.

Päätettäessä käyttää Ubuntu käyttöjärjestelmänä voitiin luopua etätyöpöytäohjelmista, sillä Ubuntu voidaan käyttää myös tekstipohjaisena. Etäyhteyteen riittää, Linuxia käytettäessä, terminaali, tai Windows:ia käytettäessä esim. Putty. Etäyhteys muodostetaan SSH-tunnelia käyttäen.

Python-skriptejä käytettäessä täytyi ottaa huomioon, että Python on yksinkertainen olioskriptikieli, joka tarvitsi erilaisia kirjastoja toimiakseen. Erilaisia kirjastoja tarvittiin muun muassa MySQL-tietokantayhteyttä ja sarjaporttia varten.

5.2.2 Ohjelmat ja niiden asennus

Kun Ubuntu oli asennettu, aloitettiin ohjelmien asennus. Ensimmäisenä päivitetiin Ubuntu ohjelmapakettilista. Tämä tehtiin avaamalla tehtävapalkista komentotulkki, eli terminaali. Tämän jälkeen komentotulkissa ajettiin järjestelmänvalvojan oikeuksilla komento *sudo apt-get update*, jolla pakettelista päivitettiin.

Kun pakettilista oli päivitetty, aloitettiin varsinaisten ohjelmien asentaminen. Ensimmäisenä ohjelmana asennettiin Python. Ubuntu 10.04LTS sisälsi Pythonin jo oletuksena, mutta varmistuksena uusimmista Python paketeista ajettiin komento *sudo apt-get install python*, jonka avulla Python asennetaan.

Kommunikointi mittausyksiköille tapahtui sarjaporttiin asennetun RS485-muuntimen kautta. Pythoniin tarvittiin moduuli nimeltä python-serial. Tämä mahdollisti sarjaportin käytön Pythonilla. Moduuli asennettiin komennolla *sudo apt-get install python-serial*.

Tietokantatallennus hoidettiin Python-skripteillä, ja Pythonin tietokantayhteyttä varten tarvittiin erillinen moduuli. Python-Mysqldb moduuli mahdollistaa Mysqldb-tietokantayhteyden Python skriptille. Asennus tapahtui komennolla *sudo apt-get install python-mysqldb*.

Tiedonkeruuyksikön tiedonhaku oli ajastettu, joten mahdollisimman tarkkaa aikaa ajatellen asennettiin NTP (Network Time Protocol), komennolla *sudo apt-get install ntp*. NTP:n avulla tietokoneeseen saadaan mahdollisimman tarkka kellonai-ka.

Etäyhteyttä varten asennettiin openssh-server -niminen paketti. Tämän avulla tiedonkeruuyksikköön saadaan tietoturvallinen yhteys salattua SSH-tunnelia käyttäen. Openssh-serverin asennus tapahtui komennolla *sudo apt-get install openssh-server*.

Openssh-serverin asennuksen jälkeen ajettiin terminaalissa komento *sudo cp/etc/ssh/sshd_config /etc/ssh/sshd_config.factory-defaults*, jolla tehtiin kopio ssh-serverin konfiguraatiodiestosta. Kopio mahdollisti alkuperäisten asetusten tarkastelun jälkikäteen.

Luotuun konfiguraatiodieston kopioon asetettiin lukuoikeudet, jotta sitä voitaisiin käyttää myöhemmin tarkastelua varten. Tällä estettiin muutoksien tekeminen vahingossa, pelkästään tarkastelua varten olevaan tiedostoon. Lukuoikeudet asetettiin komennolla *sudo chmod a-w /etc/ssh/sshd_config.factory-defaults*.

Ssh-serverin oletusasetusten tallentamisen jälkeen voitiin aloittaa asetusten muuttaminen avaamalla konfiguraatitiedosto. Tiedosto avattiin komennolla `sudo nano /sshd_config`. Muokkaamiseen käytettiin Nano-nimistä tekstinmuokkausohjelmaa terminaalien kautta.

Ssh-serverin konfiguraatitiedostoa muokattiin niin, että siitä poistettiin salasana-autentikointi. Autentikoinissa tullaan käyttämään ns. avaintiedostoa, jotta vain ne tietokoneet, joiden avaimet on määritelty tiedonkeruuyksikölle, voivat muodostaa siihen ssh-yhteyden. Tämä tehtiin muuttamalla kohtaa `#PasswordAuthentication yes` → `#PasswordAuthentication no`.

Autentikointiavain-tiedosto luodaan tietokoneessa, josta yhteys halutaan muodostaa. Se luodaan komennolla `ssh-keygen -t dsa`. Tiedoston luonnin jälkeen se kopioidaan siihen tietokoneeseen, johon yhteys halutaan muodostaa. Kopioinnin jälkeen ajetaan vielä komento `cat id_dsa.pub >> .ssh/authorized_keys`, jolla kopioidaan uusi avain luotettujen avainten joukkoon.

Mahdollisten verkkohyökkäysten tai luvattomien tunkeutumisien ollessa yleisiä, asennettiin niiden estämiseksi vielä paketti nimeltä `denyhosts`. Sen avulla väärillä tunnuksilla yrittävien osoitteet estetään kokonaan.

Kun kaikki tarvittavat ohjelmat oli asennettu, voitiin ssh-server käynnistää komennolla `sudo /etc/init.d/ssh restart`. Tämän jälkeen voitiin siirtyä säätämään päivitys asetuksia. Ensin käynnistettiin tehtäväpalkista `Update Manager`. Ikkunan auettua tuli näkyviin lista saatavilla olevista päivityksistä. Päivitykset asennettiin painamalla ikkunan oikeassa alakulmassa olevaa `Install Updates` -painiketta.

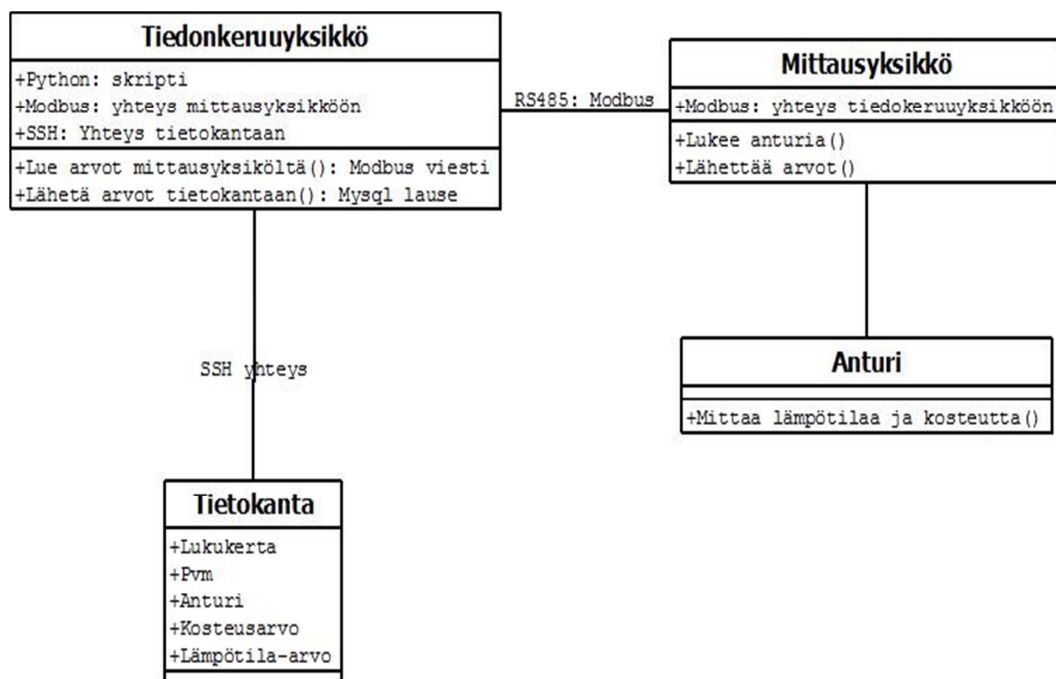
Kun päivitykset olivat asentuneet, varmistettiin, oliko vielä lisää asennettavia päivityksiä. Tämä tapahtui painamalla `Check`-painiketta. Kaikkien päivitysten asennuttua, voitiin poistaa automaattinen päivitysten tarkistus painamalla `Update Manager` -ikkunan `Settings` -painiketta.

Asetusikkunan auettua otettiin pois valintakohdasta *check for updates*, jotta käyttäjärjestelmä ei turhaan tarkistelisi päivityksiä. Päivitykset voi näin asentaa hallitusti.

5.3 Kuvaus järjestelmästä

5.3.1 Luokkakaavio

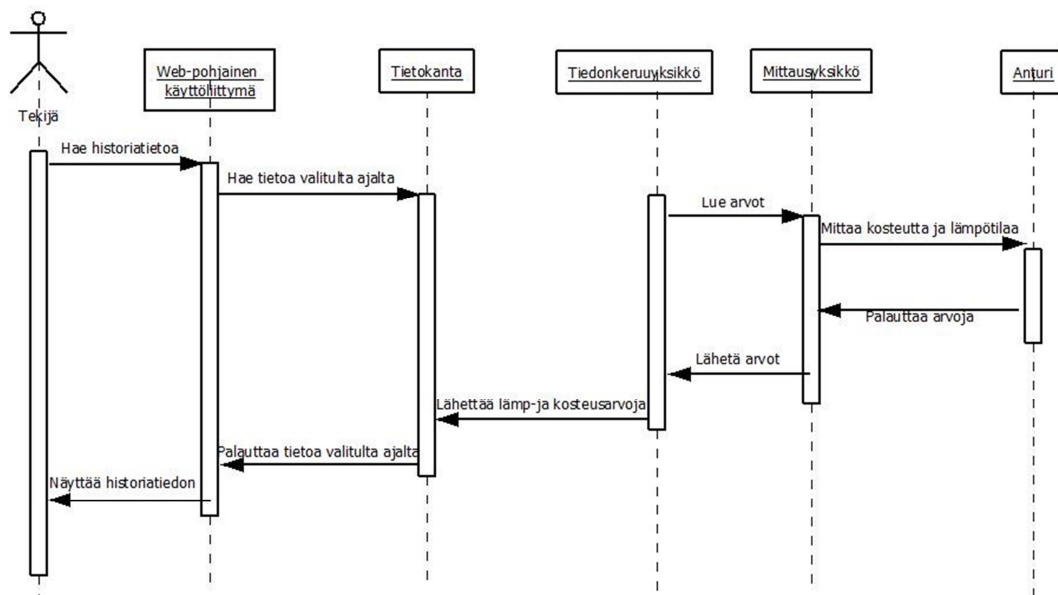
Kuvassa on esitetty mittausjärjestelmä luokkakaaviona. Luokkakaavion avulla esitetään eri osien assosiaatiot ja pyritään havainnollistamaan järjestelmän toimintaa. Kuvasta nähdään tiedonkeruuyksikön keskeinen rooli ja tämän työn tärkeys.



Kuvio 12. Luokkakaavio järjestelmästä.

5.3.2 Viestiyhteyksikaavio

Kuvassa havainnollistetaan järjestelmän toimintaa viestiyhteyksikaavion avulla. Kuvassa viestien aikajärjestys on ylhäältä alaspäin. Syy, minkä vuoksi tietokantalta ei mene viestiä tiedonkeruuyksikölle on, että tiedonkeruuyksikkö tallentaa tietoa ajastetusti ja sen tietokantayhteys on yksisuuntainen. Eli tiedonkeruuyksikkö vain tallentaa tietokantaan, mutta se ei lue sieltä mitään.



Kuvio 13. Viestiyhteyskaavio järjestelmästä.

5.4 Kuvaus Python-skripteistä

Työssä käytetään kahta Python-skriptiä. `Moisture_measurement.py` -niminen skripti tallentaa tietoa kantaan puolen tunnin välein. Tätä tietoa käytetään taulukoiden ja graafisten esitysten esittämiseen. Toinen skripti on nimeltään `current_all.py` ja sitä käytetään reaaliaikaisten arvojen hakemiseen. Molemmissa skripteissä on vielä virheitä ja esimerkiksi LRC –virheentarkistus ei toimi. Skripteissä on vielä kehittämisen varaa, mutta ne toimivat niin, että tietoa saadaan talteen.

5.4.1 Current_all

`Current_all`, eli tämänhetkiset kaikki, on skripti, jota käytetään reaaliaikaisten tietojen hakemiseen. Sillä haetaan tietoa viiden minuutin välein. Skriptin avulla saadaan mahdollisimman reaaliaikaiset tiedot lämpötila- ja kosteusarvoista. Sen hakemia arvoja ei tallenneta historiikkiin, vaan tietokannassa näkyy vain viimeisimmät arvot. `Current_all`:issa muodostetaan ja lähetetään modbus-viesti. Viestin lähettämisen jälkeen odotetaan vastausta. Jos vastaus tulee, se tallennetaan tietokantaan. Mikäli vastausta ei tule, kirjataan tapahtuma virhelokiin (`f.write(time_formatted + " - Address:" + "%02d" % (addr) + " ** No response **\n")`).

Current_all skriptissä käytetään virhelokia, johon kirjataan merkintä, kun mittauslaitteet eivät vastaa. Skripti on yksinkertainen ja helposti luettavissa. Skriptin ajastus on toteutettu Ubuntun crontab -tiedostoa hyväksikäyttäen. Crontab-tiedostoon voidaan ajastaa haluttujen ohjelmien käynnistys.

Current_all koostuu neljästä päätoiminnoista. Ensimmäinen päätoiminto on modbus-viestin muodostaminen (*message*="02x02x04x04x" *%(addr,3,0,8)*) ja lähettäminen (*ser.write(message)*). Toinen päätoiminto on viestin vastaanottaminen (*response = ser.read(100)*) ja virheentarkistus. Kolmas päätoiminto on oikeiden arvojen laskeminen. Ensin joudutaan purkamaan arvot vastauksesta, jonka jälkeen ne joudutaan laskemaan kaavalla, joka määräytyy anturin mallista riippuvasta laskukaavasta ($f_inner_temp=d1 + (d2 *inner_temp)$, $f_inner_hum=c1 + (c2 *inner_hum)+(c3*(inner_hum* inner_hum))$, $f_inner_hum=(f_inner_temp-25)*(t1+t2*inner_hum)+f_inner_hum$). Neljäs päätoiminto on mitattujen arvojen päivittäminen tietokantaan (*sql="update current_raw set dt='%(s', inner_temp=%d,inner_hum=%d, outer_temp=%d,outer_hum= %d,device_temp=%d,device_hum=%d where device_id=%d" \ %(time_formatted, inner_temp,inner_hum,outer_temp, outer_hum,device_temp,device_hum,addr)*).

5.4.2 Moisture_measurement

Moisture measurement.py on skripti, jolla varsinainen historiikki talon lämpötila ja kosteusarvoista kerätään. Moisture_measurement -skripti ajetaan puolen tunnin välein. Sen varsinainen toiminta on hyvin samankaltainen kuin Current_all:illa, mutta kerätty tieto tallennetaan tietokantaan pysyvästi, jotta arvoja voitaisiin tarkastella jälkepäin.

Moisture-measurement:illa on samat päätoiminnot kuin current_all:issa. Ainoana erona on se, että current_all:issa arvot päivitetään tietokantaan, kun taas moisture_measurementissa arvot tallennetaan (*sql="insert into device_real values ('%(s',%d,%d, %f,%f, %f,%f, %f,%f, %f,%f)" \ %(time_formatted,counter[0], addr,f_inner_temp,f_inner_hum,f_outer_temp,)*, eli ns. lisätään tietokantaan.

6 TESTAUSSUUNNITELMA

Suunnitelmana on testata tiedonkeruuyksikön viansietokykyä eli sitä, miten se palautuu sähkökatkon jälkeen, ja kuinka se toimii, jos laitteet eivät vastaa sen kyselyihin. Lisäksi aiotaan testata, kuinka se reagoi, jos verkkoyhteys tietokantaan ei toimi. Testaus toteutetaan Technobothnian testiympäristössä.

Sähkökatko testataan yksinkertaisesti vetämällä sähköjohto seinästä kesken toiminnan. Tiedonkeruuyksikön käynnistymistä seurataan. Lisäksi tutkitaan, kauanko yksiköllä menee käynnistyä takaisin lähtötilanteeseen.

Tiedonkeruuyksikön reagoimista siihen, että laitteet eivät vastaa, tutkitaan kytkemällä virrat pois satunnaisista mittausyksiköistä. Lokitiedostoa käydään läpi, ja tarkastetaan, että osaako tiedonkeruuyksikkö siirtyä eteenpäin vai jääkö se odottamaan vastausta sammutetulta laitteelta.

Tietokantayhteyden katkeaminen simuloidaan kytkemällä tiedonkeruuyksikön ethernet-kaapeli irti toiminnan aikana. Reagointia voidaan tutkia tarkastamalla lokitiedosto.

Testin lopuksi suoritetaan tulosten tarkastelu, jonka avulla varmistetaan, että kaikilta laitteilta saadaan tietoa. Lisäksi tuloksista nähdään toimivatko Python-skriptin odotetusti. Testin tulokset on lisätty tämän työn liitteeksi (LIITE).

7 YHTEENVETO

Työssä päästiin tavoitteisiin eli saatiin muodostettua toimiva tiedonkeruuyksikkö, joka hakee tietoa mittausyksiköiltä käyttäen Modbus-protokollaa. Myös tietokantatallennus saatiin toimimaan. Työn toteutus tuli edulliseksi, sillä yhtään ohjelmaa ei loppujen lopuksi tarvinnut ostaa.

Työn alkuperäisissä vaatimuksissa oli laajennettavuus, ja siitä jouduttiin hieman karsimaan. Siirryttäessä käyttämään Linux-kokoonpanoa ja Python-skriptejä, suljettiin pois OPC ja sen tuomat laajennusmahdollisuudet, tosin luovuttaessa OPC:stä pienennettiin kustannuksia huomattavasti. Myös työn kesto venyi melkein kestävämmän pitkäksi, ja vaatimukset vaihtelivat useasti. Loppujen lopuksi työn tavoitteet täytettiin hyvin

Työ oli todella opettavainen, sillä siinä joutui tutustumaan todella moneen tekniikkaan. Tekniikkojen lisäksi tiedonhakua joutui harjoittelemaan, sillä tietoa oli välillä liian vähän ja välillä liikaa. Työn aikana oltiin myös yhteydessä eri ohjelmistotalojen toimittajiin, mikä myös omalta osaltaan valmensi työelämään.

Jatkossa työn laajennettavuutta ja yksinkertaisuutta tulisi kehittää. Tiedonkeruuyksikön vikatilanteisiin tulisi kiinnittää enemmän huomiota kehittämällä erilaisia lokitiedostoja ja hälytyksiä.

LÄHTEET

- /1/ Hahtokari Tapani 1.8.2009. Matalaenergiarakentamisen edistäminen kerrostalotuotannossa. [viitattu 11.3.2011]. Saatavilla [www-muodossa:](http://www.puv.fi/fi/com/projects/archive/view/matalaenergiarakentamisen_edistaminen_kerrostalotuotannossa/)
<URL:http://www.puv.fi/fi/com/projects/archive/view/matalaenergiarakentamisen_edistaminen_kerrostalotuotannossa/>
- /2/ Harsu, Maarit (2005). Ohjelmointikielet, 287-288.
- /3/ Koskinen, Jari 2006. Mikrotietokonetekniikka, 270-271.
- /4/ Lammert Bies 2010. Modbus message structure. [viitattu 12.3.2011]. Saatavilla [www-muodossa:](http://www.lammertbies.nl/comm/info/modbus.html)
<URL:http://www.lammertbies.nl/comm/info/modbus.html>
- /5/ Matila Jukka 2010. Kosteusmittausjärjestelmä. [viitattu 12.3.2011]. Saatavilla [www-muodossa:](http://matalaenergia.puv.fi/index.php?page=kosteusmittausjaerjestelmae)
<URL:http://matalaenergia.puv.fi/index.php?page=kosteusmittausjaerjestelmae>
- /6/ Matila Jukka 2010. Mittausjärjestelmän yleiskuvaus . [viitattu 9.3.2011]. Saatavilla [www-muodossa:](http://matalaenergia.puv.fi/index.php?page=mittausjaerjestelm-auml)
<URL:http://matalaenergia.puv.fi/index.php?page=mittausjaerjestelm-auml>
- /7/ Meloni, Julie (2003). MySQL,8-12.
- /8/ Microsoft 2011. Microsoft Product Lifecycle Search. [viitattu 104.2011]. Saatavilla [www-muodossa:](http://support.microsoft.com/lifecycle/?LN=en-gb&C2=1173)
<URL:http://support.microsoft.com/lifecycle/?LN=en-gb&C2=1173>
- /9/ Modbus FAQ: About The Modbus Organization. [viitattu 29.6.2010]. Saatavilla [www-muodossa:](http://www.modbus.org/faq.php)
<URL:http://www.modbus.org/faq.php>
- /10/ Modicon Inc. 1996. Modicon Modbus Protocol Reference Guide. [viitattu 20.3.2011]. Saatavilla [www-muodossa:](http://www.modbus.org/docs/PI_MBUS_300.pdf)
<URL:http://www.modbus.org/docs/PI_MBUS_300.pdf>
- /11/ OPC Foundation 2011. What is OPC [viitattu 20.3.2011]. Saatavilla [www-muodossa:](http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC)
<URL:http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC>
- /12/ OPC Data Logger. OPC Data Logger. [viitattu 17.3.2011]. Saatavilla [www-muodossa:](http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC)

- <URL:http://www.opclogger.com/products.php?product=opcdatalogger#pro>
- /13/ Pitkänen, Sari H. SSH - Secure Shell. Päivitetty, Suvi Rätty 5.11.2010 [viitattu 29.3.2011]. Saatavilla www-muodossa: <URL:https://wiki.uef.fi/display/opkmateriaalit/SSH++Secure+Shell>
- /14/ RealVNC Limited. VNC® - the original cross-platform remote control solution. [viitattu 1.4.2011]. Saatavilla www-muodossa: <URL:http://www.realvnc.com/vnc/index.html>
- /15/ Spirecoolers.com. [viitattu 20.3.2011]. Saatavilla www-muodossa: <URL:http://www.spirecoolers.com/main/product_detail.asp?ProdID=923>
- /16/ Ubuntu suomi. Esittely. [viitattu20.3.2011]. Saatavilla www-muodossa: <URL:http://wiki.ubuntu-fi.org/Esittely>
- /17/ Verkkokauppa.com. [viitattu 20.3.2011]. Saatavilla www-muodossa: <URL:http://www.verkkokauppa.com/productimages/orig/86546_02.jpg>
- /18/ W3schools. OS Platform Statistics. [viitattu 21.3.2011]. Saatavilla www-muodossa: <URL:http://www.w3schools.com/browsers/browsers_os.asp>
- /19/ Ylönen & Lonvick 2006. The Secure Shell (SSH) Authentication Protocol. [viitattu 25.3.2011]. Saatavilla www-muodossa: <URL:http://tools.ietf.org/html/rfc4252>
- /20/ Ympäristöministeriö 2010. Energiatohokkuus. Päivitetty 16.2.2010 [viitattu 9.3.2011]. Saatavilla www-muodossa: <URL:http://www.ymparisto.fi/default.asp?contentid=2194>

TESTIRAPORTTI

Testaus aloitettiin suunnitelman mukaisesti vetämällä virtajohto seinästä kesken laitteen toiminnan. Tämän seurauksena laite ymmärrettävästi sammui. Johdon takaisinkytkennän jälkeen laite käynnistyi uudelleen. Aikaa käynnistymiseen meni noin 50 sekuntia.

Tiedonkeruuyksikön reagoimista siihen, että satunnainen mittalaite ei vastaa, ei tarvinnut simuloida, sillä testiympäristössä ei ollut kahtatoista laitetta. Tämän johdosta osa laitteista ei vastannut. Tällä testillä kyettiin osoittamaan, että virhelokin kirjoitus toimii sillä virhelokiin ilmestyi ilmoitukset kaikista laitteista, jotka eivät vastanneet.

Tietokantayhteydenkään simulointia ei tarvittu, sillä Technobothnian verkko oli kaatunut testipäivänä. Tästä johtuen laite ei saanut ollenkaan verkkoyhteyttä. Jatkoa ajatellen onkin hyvä, että virheloki tällaisia tilanteita varten on kehitteillä.

Testin tulokset olivat odotettuja vastaavia. Niiden avulla saatiin osin testattua skriptien toimivuus. Turhia ongelmia toi verkon toimimattomuus. Näin ollen ei testin avulla saatu tuloksia tietokantaan. Skriptien toimivuus on silti testattu asentamalla vastaavat ohjelmat Vaasan ammattikorkeakoulun kannettavaan. Kannettava on sijoitettu yhteen Suvilahteen rakennetuista kerrostaloista, jossa se mittaa tietoa tälläkin hetkellä. Tuloksia voi käydä katsomassa osoitteesta <http://matalaenergia.puv.fi/index.php?page=kosteusmittaustulokset>.