

KARELIA-AMMATTIKORKEAKOULU  
Anssi Tuovinen

Anssi Tuovinen

ANTURIDATAN KÄSITTELY JA SOVELTUVUUS TYÖ-  
OLOSUHTEIDEN TARKKAILUUN

Opinnäytetyö  
Toukokuu 2016



**OPINNÄYTETYÖ**  
**Toukokuu 2016**  
**Tietojenkäsittelyn koulutusohjelma**

Tikkarinne 9  
80220 JOENSUU

Tekijä  
Anssi Tuovinen

Nimeke  
Anturidatan käsittely ja soveltuvuus työolosuhteiden tarkkailuun

**Tiivistelmä**

Opinnäytetyössä tarkasteltiin anturidatan käyttömahdollisuuksia ja soveltuvuutta toimistotyön työolosuhteiden tarkkailuun ja mittaamiseen. Työtä varten rakennettiin yksinkertainen Arduino-elektroniikka-alustaan pohjautuvan anturilaitteen prototyyppi sekä tarvittavat tietokantayhteydet. Opinnäytetyössä kuvataan anturilaitteen rakenne sekä anturidatan käsittelyn ja tietokantaan tallennuksen periaatteita.

Työssä tarkasteltiin työolosuhteita työympäristön fysikaalisten riskitekijöiden näkökulmasta. Työympäristönä oli perinteinen toimistotyöympäristö, jonka keskeiset fysikaaliset riskitekijät esitellään tässä raportissa. Lisäksi esitellään myös niiden mittaamiseksi saatavilla olevia peruselektroniikan komponentteja, jotka ovat yhteensopivia Arduino-alustan kanssa.

Anturidatan keräämistä testattiin toimistotyöolosuhteissa eräässä avotoimiston työpisteessä. Tuloksena saatiin selville, että anturilaitteella toteutetulla jatkuvalla työolosuhdeseurannalla voidaan havaita muutoksia, joita työntekijät eivät itse välttämättä havaitsisi. Työn loppuosassa tarkastellaan mahdollisia jatkokehityskohteita, joilla anturilaitteen prototyyppistä voitaisiin kehittää julkaisu- ja toteutuskelppoinen tuote.

Kieli

Sivuja 34

suomi

Liitteet 1

Asiasanat

työympäristö, anturit, elektroniikka, esineiden internet



**THESIS**  
**May 2016**  
**Degree Programme in Business Information Technology**

Tikkarinne 9  
80220 JOENSUU

Author  
Anssi Tuovinen

Title  
Processing of Sensor Data and Suitability for Monitoring Working Conditions

**Abstract**

In this thesis the possibilities of sensor data gathering to monitor surrounding environment and working conditions were investigated. An Arduino based prototype of a sensor datalogger device and database system were built for testing and monitoring purposes. The structure of the device and the data processing methods are also described in this report.

The conditions observed were limited to physical factors during the ordinary indoor office work. The most essential physical factors are introduced in this report as are also Arduino compatible basic electronic components, which may be used to measure the environmental conditions.

A testing session was held to gather sensor data in authentic environment. Results showed that sensor monitoring has a vast potential in noticing changes in conditions which would not be noticeable by human senses. In the latter part of this thesis, the possible development targets to develop the prototype into a fully functional final product are being examined.

Language

Pages 34

Finnish

Appendices 1

Keywords

working environment, sensors, electronics, The Internet of Things

# Sisältö

1	Johdanto.....	5
2	Fysikaaliset riskitekijät toimistotyössä .....	6
2.1	Melu.....	7
2.2	Lämpöolot.....	8
2.3	Valaistus .....	9
2.4	Säteily.....	10
3	Arduino .....	11
3.1	Uno.....	12
3.2	Mega .....	13
3.3	Yún .....	13
3.4	Ohjelmointi.....	14
3.5	Arduinon tulevaisuus .....	15
4	Anturilaitteen toteutus .....	16
4.1	Tekninen rakenne .....	16
4.2	Lämpötilan ja kosteuden mittaaminen.....	17
4.3	Valaistuksen havainnointi .....	19
4.4	Muut mittausmahdollisuudet .....	20
4.5	Verkkoyhteys .....	22
5	Tietokanta.....	24
6	Laitteen testaus .....	26
6.1	Testiympäristö .....	26
6.2	Luotettavuus ja käytettävyys.....	27
6.3	Datan hyödyntäminen .....	29
7	Pohdinta .....	31
	Lähteet .....	35

## Liitteet

Liite 1      Arduinon ohjelmakoodi

## 1 Johdanto

Tiedonsiirron nopeutumisen ja yleisen teknologisen kehityksen seurauksena yhä useammat laitteet ja koneet kytketään lähes poikkeuksetta internetverkkoon. Puhutaan paljon käsitteestä The Internet of Things (suom. esineiden internet), joka tarkoittaa sitä, että eri laitteet voivat verkon välityksellä lähettää ja vastaanottaa tietoja sekä reagoida eri tilanteisiin halutulla tavalla. Tämä mahdollistaa muun muassa langattomien anturiverkkojen kehityksen, jossa vaihteleva määrä erilaisia antureita mittaa esimerkiksi nousuveden korkeutta rantaviivalla tai tulivuoren purkauksen aiheuttamia värähtelyitä maaperässä. Antureiden määrää rajoittaa käytännössä vain tekniset esteet, kuten esimerkiksi laitteiden käytettävissä oleva muisti, mikropiirien kellotaajuus tai käytettävissä oleva energia. Kehitys on kuitenkin osoittanut, että mikään näistä tekijöistä ei ole este vaan pikemminkin hidaste kehityksen tiellä.

Yhtä nopeasti kuin tiedonsiirron kehittymisen seurauksen mahdollisuudet, on myös tietoisuus työhyvinvoinnin vaikutuksesta kasvanut kiihtyvällä vauhdilla koko 2000-luvun alun. Pidämme jo itsestänselvyytenä, että kun työntekijöitä kohdellaan hyvin, työolosuhteet ovat turvallisia ja työntekijä kokee itsensä arvokkaaksi ja halutuksi yritykselle, on sillä suuri vaikutus alkaen aina yksittäisen työntekijän työssä viihtymisen kannalta aina kansantaloudellisiin vaikutuksiin saakka. Työturvallisuuden ja esimerkiksi sisäilman laadun osalta lainsäädäntö, rakennustekniset ohjeet ja määräykset ovat vaikuttaneet työhyvinvointiin lisäämällä turvallisuutta ja kestävyyttä. Valitettavan usein arkipäiväisten olosuhteiden vaikutus havaitaan vasta silloin, kun hyvinvoinnin laiminlyönnin riskit realisoituvat, ja työntekijä joutuu jäämään esimerkiksi varhaiseläkkeelle.

Opinnäytetyö koostuu sekä käytännöllisestä että teoreettisesta osiosta, joten kyseessä on toiminnallinen opinnäytetyö. Raportin tieto-osiota varten tietoa kerättiin pääsääntöisesti internetlähteistä. Raportin teoreettisessa osiossa kuvataan luvussa kaksi työympäristön fyysiset riskitekijät, joiden mittaamista varten opinnäytetyössä rakennettiin yksinkertaisen anturilaitteen prototyyppi ja tietokantayh-

teydet. Laite perustuu Arduino-elektroniikka-alustaan, jonka yleisesittely on luvussa kolme. Laitteen kehitystyössä noudatettiin perinteisen vesiputousmallin toimintamallia.

Rakennetun anturilaitteen prototyypin ja järjestelmän rakenne ja toiminnallisuudet on kuvattu neljännessä ja viidennessä luvussa. Työssä pohditaan myös mahdollisuuksia kerätyn tiedon hyödyntämiseen työolosuhteiden ennustamisessa ja riskienhallinnassa. Luvussa kuusi on kuvattu arviointia varten perustetut testijärjestelyt sekä arvioitu anturidatan käyttökelpoisuutta kyseisessä testauskohteessa. Viimeisessä luvussa on yhteenveto laitteen ja järjestelmän toteutuksesta sekä opinnäytetyön prosessista yleisesti.

## **2 Fysikaaliset riskitekijät toimistotyössä**

Työhyvinvointi on mittava kokonaisuus, jonka ylläpitämisestä ja kehittämisestä vastuu on ensisijaisesti työnantajalla. Työturvallisuuslaki (738/2003) velvoittaa työnantajan huolehtimaan työskentelyolosuhteiden turvallisuudesta ja terveellisyydestä. Lain velvoituksen lisäksi yritykset ovat havahtuneet siihen, että työhyvinvointi vaikuttaa niin yksittäisen työntekijän elämään kuin myös yrityksen talouteen. Työhyvinvointiin on alettu kiinnittää erityistä huomiota etenkin 2000-luvulla, ja on tiedostettu, että työhyvinvoinnin – tai pikemminkin sen puutteiden – merkitys on merkittävä myös kansantaloudelle.

Viime vuosina etenkin Työterveyslaitos on tutkinut työhyvinvoinnin taloudellista merkitystä. Vuonna 2013 julkaistussa Työ ja terveys Suomessa 2012 -tutkimuksen lopputuloksena todetaan yksiselitteisesti, että työhyvinvoinnin merkitys niin kansantalouden kuin yksittäisten yritysten talouden kannalta on merkittävä. Tutkimuksen mukaan työhyvinvointipanostusten hyötypotentiaali olisi kansantalouden tasolla 35 miljardia ja yritysten talouden kannalta 4,5 miljardia euroa vuodessa. (Työterveyslaitos 2012.)

Huolimatta selvästä potentiaalisesti saavutettavasta hyödystä ja työhyvinvoinnin tilan tiedostamisesta, on työhyvinvointiin panostettu yrityksissä kuitenkin suhteellisen vähän. Vuosina 2009–2012 suomalaiset yritykset käyttivät vain noin 2,1 miljardia euroa työhyvinvoinnin kehittämiseen. Esimerkiksi ennenaikaisen eläkkeelle jäämisen kustannukset ovat kuitenkin vuosittain yhteensä jopa 24 miljardia euroa (Aura, Ahonen & Ilmarinen 2012.)

Yksi merkittävä yksittäinen työhyvinvointiin vaikuttava tekijä on työympäristössä vallitsevat fyysiset olosuhteet. Etenkin tietotyössä, jossa toimisto- ja näyttöpäätetyön osuus on suuri, on sisäilman laadulla, valaistuksella, lämpötilalla ja kosteudella merkittävä vaikutus työssä ja työympäristössä viihtymiseen. Tässä opinnäytetyössä keskitytään työympäristön fyysisiin riskitekijöihin, joita voidaan mitata.

## 2.1 Melu

Meluvammat ovat ihotautien ohella suurin ammattitautiryhmä Suomessa, kun mitataan Työperäisten sairauksien rekisteriin (TPSR)<sup>1</sup> ilmoitettujen meluvammojen määrää. Vuonna 2013 havaittuja meluvammoja oli 1040, joka on noin 22 % kaikista ammattitautiryhmien vammoista. Viimeisen vuosikymmenen aikana suhde on vuositasolla vaihdellut 22 % ja 26 % välillä. (Työterveyslaitos.) Meluvammariski on suurin teollisuustyössä, jossa 85 dB(A):n Euroopan unionin meludirektiivin 2003/10/EY mukainen raja-arvo kuulonsuojauksen järjestämisen tarpeelle ylittyy.

Perinteisessä toimistotyössä tasainen melu harvoin ylittää äänenvoimakkuuden tasoa, jolla olisi vaikutusta kuulon alenemiseen. Maailman terveysjärjestön (WHO) mukaan liiallinen ympäristön melu kuitenkin myös häiritsee ihmisten päivittäistä elämää niin työssä kuin vapaa-ajallakin. Perinteisten kuulovaurioiden ja kuulonalenemisen lisäksi melusaaste vaikuttaa ihmisten sosiaaliseen käyttäytymiseen ja laskee suorituskykyä häiriten esimerkiksi keskittymistä ja tavanomaista keskustelua. Toimistotyöolosuhteissa häiritseviä äänilähteitä voi olla esimerkiksi

---

<sup>1</sup> Työperäisten sairauksien rekisteri (TPSR) (ent. Suomen ammattitautirekisteri) on Työterveyslaitoksen (TTL) ylläpitämä tutkimusrekisteri, jonka tietoja voidaan hyödyntää mm. ammattitautien ja muiden työstä johtuvien sairauksien tutkimisessa ja ehkäisyssä (TTL 2016).

ilmastointilaitteet, tietokoneiden tuulettimet ja yleisesti huono äänieristys, jolloin esimerkiksi liikenteen äänet kantautuvat häiritsevästi muihin tiloihin. Myös työympäristön yleinen akustiikka, huonekalut ja niiden sijoittelu sekä rakennusmateriaalit vaikuttavat merkittävästi siihen, miten työntekijät kokevat ympäröivät äänet. (World Health Organization 1999.)

Toimistotyössä häiritseväksi ja haitalliseksi voidaan kokea myös niin sanottu impulssimelu. Työturvallisuuskeskuksen määritelmän mukaan melu on impulssimelua, jos ” – – se sisältää alle sekunnin kestäviä meluhuippuja, jotka ovat yli 15 desibeliä taustamelua voimakkaampia”. Etenkin keskittymistä vaativissa työtehtävissä taustamelun läpi kuuluva yllättävä ääni vaikuttaa työn tekemisen mielekkyyteen. Toimistotyössä impulssimelua voivat aiheuttaa rikkiäiset talo- ja rakennustekniikkaan liittyvät laitteet tai sitä voi kantautua työtilaan esimerkiksi muista sisä- tai ulkotiloista.

## **2.2 Lämpöolot**

Ilman lämpötila, kosteus ja ilmanvaihto muodostavat lämpöolojen kokonaisuuden. Työterveyslaitoksen mukaan lämpöoloiltaan viihtyisiksi luokiteltavissa sisätiloissa työskentelee Suomessa noin 1,7 miljoonaa työntekijää. Vaikka tilassa saavutettaisiinkin ihanteelliset Suomen rakentamismääräyskokoelman ohjeavojen mukaiset olosuhteet, on 10 % työntekijöistä edelleen tyytymättömiä sisäilman lämpötilaan. Sisäilmaa pidetään tyydyttävänä silloin, kun tyytymättömiä työntekijöitä on alle 30 %. (Työterveyslaitos 2014.)

Toisin kuin teollisuusympäristössä ei toimistotyössä ilmastonin tehtävänä yleensä ole suodattaa hengitysilma epäpuhtauksia tai myrkyllisiä kaasuja. Toimistoissa ilmastoinnilla on suuri vaikutus ilmanvaihtoon, joka vaikuttaa suoraan viihtyvyyteen. Toimistotyössä lämpötilan, kosteuden ja ilmanpaineen olisi työssä viihtymisen kannalta hyvä olla mahdollisimman tasainen. Esimerkiksi ilman lämpötilan suuri vaihtelu saa aikaan ylimääräistä ilman liikettä, jonka työntekijä voi aistia epämiellyttävänä vedon tunteena, johon vaikuttaa myös tulevan ja poistuvan ilmamäärien suhde.



Työterveyslaitoksen tutkijat ovat myös tutkineet lämpötilan vaikutusta työsuoriutumiseen toimisto-olosuhteissa. Tutkimuksen lopputuloksena oli, että lämpötila vaikuttaa suoriutumisen lisäksi myös motivaatioon ja energiatasoon. Tutkimuksen mukaan työsuoriutumisen kannalta optimaalisin lämpötila olisi välillä 21–25 °C, joka on myös työsuojeluhallinnon lämpötilasuositus kevyelle istumatyölle (Työsuojeluhallinto 2015a). Merkittävää tutkimuksessa oli myös, että testitilanteen yhtenäisestä vaatuksesta huolimatta miesten ja naisten välillä oli merkittäviä eroja lämpöviihtyvyydessä. Tutkimuksen vielä ei ole pystytty selvittämään sitä mikä eron sukupuolten välillä aiheuttaa. (Hägglom, Haapakangas, Hongisto, Koskela 2011.)

Ilmanvaihdolla voidaan vaikuttaa suuresti myös sisäilman kosteuteen. Hengityselimen mukaan sopiva suhteellinen kosteus huoneilmassa olisi 20–40 % (Hengityselinliitto). Riittämätön ilmanvaihto ei pysty poistamaan huoneilmasta liiallista kosteutta, joka ajan kuluessa voi johtaa homevaurioihin rakennuksen rakenteissa. Homeille altistumisen tavanomaisimpia oireita ovat päänsärky ja väsymys, jotka vaikuttavat merkittävästi työtehokkuuteen ja työssä viihtymiseen. Liian kuiva sisäilma taas voi lisätä hengitysteiden, limakalvojen ja ihon ärsytysoireita sekä vaikuttaa influenssavirusten leviämiseen (Lowen ym. 2007).

### **2.3 Valaistus**

Työtilojen valaistusvaatimuksista puhuttaessa mainitaan usein näkömukavuus, näkötehokkuus ja turvallisuus. Riittävä ja häiritsemätön yleisvalaistus vaikuttaa yleiseen viihtyvyyteen ja työssä olon mukavuuteen, ja vaikuttaa siten epäsuorasti myös työn tehokkuuteen ja tuottavuuteen. Näkötehokkuus tarkoittaa sitä, että yleis- ja kohdevalaistus on toteutettu siten, että työntekijä voi suoriutua työstään. Tämä vaikuttaa vahvasti suoraan myös työn turvalliseen suorittamiseen. Lainsäädäntö ei työturvallisuutta koskevissa säädöksissä ota kantaa työympäristön valaistukselle. Suositukset hyvästä valaistuskäytännöstä on mainittu sisävalaistusstandardissa SFS 12464-1.

Työsuojeluhallinto mainitsee hyvän valaistuksen tuntomerkeiksi 200 luksin yleisvalaistuksen jatkuvan työskentelyn tiloissa, jonka lisäksi työpiste on valaistava tarpeen vaatimalla tavalla. Valon tulosuuntaan on myös kiinnitettävä huomiota, jolloin vältetään häiritseviltä heijastuksilta. (Työsuojeluhallinto 2015b.)

## 2.4 Säteily

Ionisoiva säteily (esimerkiksi radioaktiiviset aineet, radon) eivät perinteisessä toimistotyössä muodosta riskitekijää. Suomessa kuitenkin noin 12000 työntekijää työskentelee sellaisissa olosuhteissa, joissa on riski altistua ionisoivalle säteilylle. Kaikki henkilöt kuuluvat säteilyvalvonnan piiriin ja säteilyn käyttöä valvoo Säteilyturvakeskus (STUK).

Toimistotyössä tavallisempaa on pienienerginen ionisoimaton säteily, joka jaetaan sähkömagneettiseen säteilyyn ja optiseen säteilyyn. Optinen säteily on keinoitekoisten valonlähteiden aiheuttamaa säteilyä ja se liittyy osittain myös valaistuksen turvallisuuteen. Esimerkiksi viallinen loisteputkivalaisin voi säteillä UV-säteilyä, jolla voi olla vaikutusta ihon tai silmien terveydelle. Valtioneuvosto velvoittaa asetuksessaan (Valtioneuvoston asetus työntekijöiden suojelemiseksi optiselle säteilylle altistumisesta aiheutuville vaaroilta 146/2010) työnantajan selvittämään altistuvatko työntekijät optiselle säteilylle ja tarvittaessa ryhtymään toimenpiteisiin säteilylähteen poistamiseksi ja säteilytasojen palauttamiseksi turvalliselle tasolle.

Sähkömagneettisia kenttiä syntyy toimistotyössä sähkölaitteista esimerkiksi tietokoneista, näytöistä ja tulostimista. Työturvallisuuskeskuksen mukaan esimerkiksi näyttöjen aiheuttamien sähkömagneettisten kenttien ei ole todettu aiheuttavan terveyshaittoja. Työntekijä voi kuitenkin kokea saavansa oireita, jolloin kyseessä on sähköliherkkyys. Vaikka sähköliherkkyiden tieteellinen näyttö onkin vähäistä, voi kyseessä olla yksilön kannalta merkittävä työhyvinvointia alentava tekijä. Tällöin työnantajan on hyvä pystyä todistamaan, että työolosuhteet ovat myös sähkömagneettisen säteilyn osalta turvalliset.

### 3 Arduino

Kehityksen myötä voimme esimerkiksi maanjäristysten ja gravitaatioaaltojen lisäksi mitata ja havaita myös arkipäiväisiäkin tilanteita tehokkaasti, luotettavasti ja reaaliaikaisesti. Useat kaupalliset ratkaisut anturidatan hyödyntämiseksi ovat kuitenkin joko äärimmäisen yksinkertaisia suljettuja järjestelmiä (esimerkiksi tavallinen langaton ulkolämpömittari) tai kalliita ammattikäyttöön tarkoitettuja asiakkaan tarpeisiin räätälöitäviä ratkaisuja. Aiheesta kiinnostuneet harrastelijat ovat tietenkin aina kehittäneet omia ratkaisujaan ja laitteitaan, mutta elektroniikkaharrastuksen aloittamisen kynnyks on korkea. Mikro-ohjaimellisen piirin rakentaminen ja ohjelmointi vaativat paljon perehtymistä ja osaamista, jotta yksinkertaisenkin mittauslaitteen toteuttaminen onnistuisi. Omalta osaltaan tähän ongelmaan vastasi italialainen instituutti *Interaction Design Institute Ivrea*, jonka tilaamana suunnittelija Hernando Barragan rakensi Arduinon edeltäjänä pidetyn Wiring-mikro-ohjaimen, josta nykyisin Arduino-nimellä<sup>2</sup> tunnettu projektin sai alkunsa. Jatkokehityksen tarkoituksena oli kehittää opetuskäyttöön edullinen kehitysalusta, joka madaltaisi kynnyksiä kokeilla ja kehittää erilaisia etenkin ympäristön kanssa interaktiivisia laitteita. (Arduino 2015a.)

Tässä opinnäytetyössä rakennettu anturilaitte pohjautuu Arduino-alustaan, joka on mikro-ohjaimen perustuva elektroniikka-alusta. Konsepti on jo hyvin varhaisessa kehitysvaiheessa suunniteltu toteuttamaan avoimen laitteiston ideologiaa. Alustojen rakenne on täysin dokumentoitu, ja alustoissa käytettävät komponentit ovat yleisesti saatavilla olevia elektroniikan peruskomponentteja, joten jokainen voisi rakentaa vastaavanlaisen laitteen itse. Virallisten Arduino-laitteiden rinnalla onkin valmistettu kymmeniä yhteensopivia laitteita, kuten esimerkiksi Teensy,

---

<sup>2</sup> Vuodesta 2015 alkaen Arduino-laitteet myydään Yhdysvaltojen ulkopuolella Genuino-nimellä. Kyseessä ovat täysin identtiset laitteet, mutta lakitekniikan syiden vuoksi yhtiön oli pakotettu muodostamaan Yhdysvaltojen ulkopuolisia markkinoita varten oma brändi. (Arduino 2015d.) Tässä esityksessä käytetään vielä Arduino-nimeä sen vakiintuneen aseman vuoksi.

JeeNode tai Richduino. Open source -hengessä virallinen Arduino-yhteisö kerääkin tietoja rinnakkaisprojekteista auttaakseen harrastajia omien laitteiden kehityksessä. (Arduino 2015b.)

Arduino-alustoja on lukuisia erilaisia. Eri alustat eroavat toisistaan muun muassa mikro-ohjaimen mallin, käytössä olevien liitäntöjen määrän ja tyyppin sekä fyysisen koon mukaan, joten eri käyttötarkoituksiin on saatavilla erilaisia valmiita kokoonpanoja. Alkuperäisissä Arduinoissa käytetään yhdysvaltalaisen puolijohdevalmistaja Atmelin AVR-mikro-ohjainperheen 8-bittisiä ohjaimia. (Arduino 2015c.)

Arduino-alustoja kehitetään myös silmällä pitäen jo ennalta määrättyjä käyttötarkoituksia. Nimensä mukaisesti, tosin nyt jo vanhentunut ja tuote-elinkaarensa loppupäähän ehtinyt, Arduino Robot soveltui pienen pyöreän muotonsa ja integroitujen pyörien ansiosta robottirakennelmien alustaksi. Älyvaatteita varten LilyPad-alustat on kehitetty niin pieniksi, että ne ovat ommeltavissa suoraan kankaaseen. (Arduino 2015c.)

Suosituimmat Arduino-alustat ovat aloittelijaystävälliset Uno ja edistyneempiä ominaisuuksia tarjoava Mega. Nykyisin myös valmiin WiFi-piirin sisältävä Yún on erittäin suosittu valinta IoT-sovelluksiin. Lisäksi muiden innovaatioiden lisäksi kehitetty Arduino Materia 101 on ensimmäinen virallinen Arduino-pohjainen 3D-tuostin. (Arduino 2015c.)

### **3.1 Uno**

Edullisen hintansa ja suhteellisen pienen kokonsa ansiosta monelle aloittelevalle kehittäjälle ensimmäinen Arduino-alusta on Arduino Uno. Alusta sisältää USB-liitännän ohjelmointia ja virransyöttöä varten sekä myös erillisen virtaliitännän esimerkiksi paristojen, akkujen tai aurinkopaneelin käyttöä varten. Jänniteregulaattorin ansiosta kehittäjän tarvitsee huolehtia vain, että virransyöttö pysyy suosituksen mukaisissa jännitearvoissa, eli välillä 7–12 V. (Arduino 2015e.)

Arduino Uno sisältää ATmega328P-mikrokontrollerin, ja sisältää siten 2 kB SRAM- ja 1 kB EEPROM-muistia. Flash-muistia Unossa on 32 kB, josta 0,5 kB on varattu mikro-ohjaimen alkulatausohjelman (engl. bootloader) käyttöön. Alkulatausohjelma voidaan myös poistaa, mutta tällöin laitteen ohjelmointi ei onnistu enää USB-liitännän kautta, vaan on käytettävä suoraan ICSP-porttia tarkoitukseen sopivalla ohjelmointilaitteella. (Arduino 2015e.)

Alustassa on käytävissä yhteensä 14 digitaalista ja 6 analogista I/O-porttia. Digitaalisista porteista kuusi tukee pulssinleveysmodulaatiota (PWM), joilla saadaan toteutettua analogisia signaalitasoja digitaalisesti. (Arduino 2015e.)

### **3.2 Mega**

Muun muassa arvostettu elektroniikka-alan toimija ja myyjä SparkFun on todennut, että Arduino Mega on Unon isovelji (B\_E\_N, SparkFun). Megassa digitaalisia I/O-liitäntöjä on 54 ja analogisia liitäntöjä 16 kappaletta. On siis selvää, että Mega soveltuu projekteihin, joissa alustaan liitetään huomattavasti enemmän esimerkiksi antureita ja muita komponentteja. (Arduino 2015f.)

Arduino Megan mikro-ohjain on myös askel suurempaan Unon ATmega328P-kontrollerista. ATmega1280-ohjaimen myötä käytävissä 8 kB SRAM- ja 4 kB EEPROM-muistia. Flash-muistia Megassa on neljä kertaa enemmän kuin Unossa. (Arduino 2015f.)

### **3.3 Yún**

Arduino-projekti kertoo sivuillaan Arduino Yún -alustasta, että "Arduino YÚN is the perfect board to use when designing connected devices and, more in general, Internet of Things projects". Vaikkakin I/O-liitäntöjen ja muistin määrällä mitattuna alusta sijoittuisi vertailussa lähemmäksi Unoa, tukee väitettä ehdottomasti se, että alusta sisältää käyttövalmiin WiFi-moduulin sekä OpenWrt-Linuxiin pohjautuvan OpenWrt-Yun käyttöjärjestelmän. (Arduino 2015g.)

OpenWrt on alun perin lähinnä tietoliikennereitittimelle suunniteltu käyttöjärjestelmä, mutta nykyisin se on löytänyt tiensä myös erilaisiin sulautettuihin järjestelmiin. Käytännössä tämä tekee Arduino Yún -alustasta jo itsessään eräänlaisen itsenäisen tietokoneen, jolla voidaan hyödyntää Linux-järjestelmistä tuttuja shell- ja Python-skriptejä. (Arduino 2015g.)

### 3.4 Ohjelmointi

Arduino-projekti on jo varhaisessa vaiheessa pyrkinyt tekemään mikro-ohjaimen ohjelmoinnista mahdollisimman helposti lähestyttävää. Erillisen ohjelmointilaitteen sijaan useimmat Arduino-alustat voidaan liittää tietokoneeseen USB-liitännän avulla. Arduino-projektin julkaiseman ja ylläpitämän kehitysympäristö Arduino Softwaren avulla lähdekoodi voidaan ladata suoraan alustan mikro-ohjaimelle.

Arduinon ymmärtämä ohjelmointikieli on yksinkertaistettu versio C++-ohjelmointikielestä. Ohjelmien rakenne toteuttaa wiring-prototyypin rakennetta, eli ohjelma rakennetaan kahden funktion ympärille. Käynnistyessään Arduino suorittaa `setup()`-funktiossa määritetyt toimenpiteet, josta tyypillisin esimerkki on I/O-liitännöjen asettaminen joko *input*- tai *output*-tilaan. Ohjelman suoritus jatkuu `setup()`-funktion jälkeen `loop()`-funktiossa, joka on käytännössä ikuinen jatkuva toistorakenne.

Esimerkiksi jos Arduino-alustan 7. digitaaliseen liitännään on liitetty äänisummerin positiivinen johdin (ja negatiivinen johdin maahan), voidaan Arduino asettaa toistamaan toistuvaa äänimerkkiä esimerkiksi seuraavalla koodilla:

```
// Määritetään summeriPin muuttuja.  
int summeriPin = 7;  
  
void setup() {  
    // Asetetaan liitännä ulostuloksi
```

```
pinMode(summeriPin, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(summeriPin, HIGH); // Ääni soi  
  delay(250); // Odotetaan 250 millisekuntia  
  digitalWrite(summeriPin, LOW); // Ääni ei soi  
  delay(1000); // Odotetaan yksi sekunti  
}
```

### 3.5 Arduinon tulevaisuus

Lienee itsestään selvää, että keskustelu ja mielenkiinto Internet of Things'n ympärillä ei ole laantumassa. Tulevaisuudessa esineiden ja asioiden lisäksi kyttemme myös itsemme entistä vahvemmin verkkoon. Jo nyt esimerkiksi mitaamme päivittäisiä askeliamme älypuhelimillamme, joka lataa tiedot pilvipalveluun. Ei tarvinne odottaa kauaakaan, että pienet anturit tarkkailevat muita elintointojamme, esimerkiksi sykettä, verenpainetta ja verensokeria reaaliajassa.

Tulevaisuuden visioita maalaillessa sekä ohjelmoinnin osaamisen suosio että IoT-viritysten rakentelu lisääntyvät vauhdilla. Markkinoille tuodaan jatkuvasti saataville uudenlaisia valmiita ratkaisuja, joiden avulla jokainen voi rakentaa itselleen niin sanotun älykodin älykkäine kodinkoneineen (mm. Telldus Technologies AB), tai vaikkapa opettaa auton ymmärtämään puhetta ja soittamaan suosikkikappaleita pelkällä ääniohjauksella (mm. Android Auto). Myös Arduino haluaa pysyä edelleen kehityksessä mukana. Tästä kertoo Intelin kanssa yhteistyössä kehitetty Arduino 101 -alusta, johon on jo valmiiksi liitetty ominaisuuksia (mm. gyroskooppi, bluetooth jne), jotka tekevät laajempienkin projektien aloittamisesta helpompaa, eikä erillisiä moduuleja ja antureita tarvitse hankkia tai asentaa erikseen. (Arduino 2015h.)

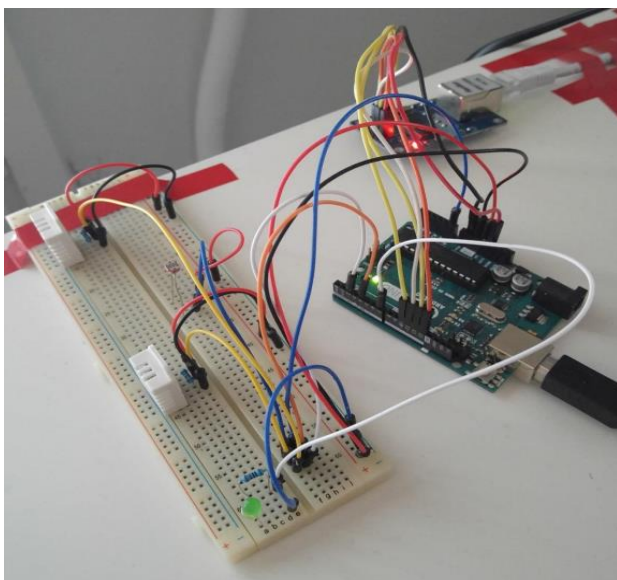
## 4 Anturilaitteen toteutus

Tässä opinnäytetyössä rakennettu anturilaitte pohjautuu Arduino Uno -alustan kolmanteen (R3) versioon. Alustan valintaan vaikuttivat laitteen saatavuus, hinta ja riittävän monipuoliset liitännävaihtoehdot. Toinen hyvä ja käyttötarkoitukseen sopiva vaihtoehto olisi ollut esimerkiksi Arduino Nano.

Seuraavissa luvuissa kuvataan anturilaitteen rakenne ja ominaisuudet. Arduinon ladattu ohjelmakoodi on kokonaisuudessaan liitteessä 1 (liite 1 Arduinon ohjelmakoodi).

### 4.1 Tekninen rakenne

Anturilaitteen käyttötarkoitus on mitata lämpötilaa, kosteutta ja valaistuksen tasoa. Laitte myös yhdistetään reitittimeen RJ45-verkkokaapelilla. Reitittimeen voitaisiin asettaa porttiosjauksen, että laite olisi näkyvä myös julkiseen internetiin, mutta tässä opinnäytetyössä laite on tietokantasovelluksen käytettävissä vain samassa lähiverkossa. Laitteen komponentit ovat nähtävillä kuvassa 1.



Kuva 1. Anturilaitteen rakenne (Kuva: Anssi Tuovinen).



Laitetta suunniteltaessa ei kiinnitetty huomiota laitteen fyysiseen kokoon. Merkittävään kokoon vaikuttava tekijä on koekytkentäalustan (breadboard, ns. leipälauta) käyttö, joka toisaalta nimensä mukaisesti helpottaa kytkentöjen kokeilua ja laitteen testausta. Seuraava luonnollinen askel prototyypin kehittämisessä olisi suunnitella laitteelle oma piirilevy.

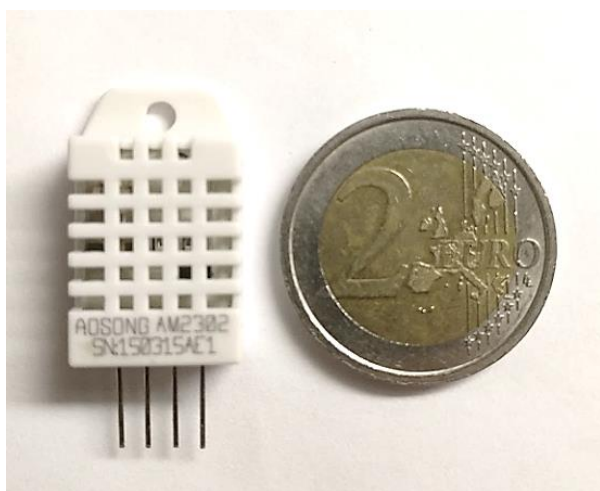
Koska laitteen tarkoitus tämän opinnäytetyön yhteydessä on tuottaa mittausdataa, ei laitetta ole myöskään suunniteltu toimimaan langattomasti. On kuitenkin selvää, että käyttökelpoisen tarkoituksenmukaisen laitteen on oltava kooltaan pieni ja huomaamaton sekä kommunikoitava tietokannan kanssa langattomasti. Pienillä muutoksilla laitteistoon ja ohjelmakoodiin tämä olisi saavutettavissa myös tämän laitteen osalta esimerkiksi käyttämällä WiFi-moduulia ja paristoja.

## 4.2 Lämpötilan ja kosteuden mittaaminen

DHT22 (esiintyy myös nimillä RHT03 tai AM2302) on edullinen ja yleisesti saatavilla oleva pienikokoinen (kuva 2) komponentti, joka mittaa sekä ilman suhteellista kosteutta että lämpötilaa. Anturin käyttöjännite ( $V_{cc}$ ) on 3,3–5,5 V, joten se soveltuu hyvin Arduino-alustan käyttöön. Anturin mittaustarkkuus suhteelliselle kosteudelle asteikolla 0–100 % on 2–5 % ja lämpötilalle +/- 0,5 °C. Anturi soveltuu myös ulkokäyttöön, sillä sen mittausskaala lämpötilalle on -40 – +80 °C. DHT22-anturin ilmoitettu mittaustaajuus on 0,5 Hz, eli käytännössä mittausta voidaan suorittaa kerran kahdessa sekunnissa. (Aosong Electronics Co., Ltd. 2013a.) Testaustarkoituksen vuoksi tämän opinnäytetyön prototyypissä molemmat DHT-22-anturit on kytketty samaan koekytkentäalustaan. Todellisessa käyttötilanteessa anturit olisivat omina erillisinä moduuleinaan (node), jotka kommunikoisivat Arduino Unon (gateway) kanssa langattomasti käyttäen esimerkiksi bluetooth-yhteyttä, WLAN-verkkoa tai RF-signaaleja. Tällä tavoin useampia antureita voitaisiin sijoittaa eri sijanteihin esimerkiksi toimistossa työpisteen eri alueille.

Toinen suosittu vastaava anturi on DHT11, joka pienemmän kokonsa ja edullisemmän hintansa ansiosta on usein yleisempi valinta harrastusprojekteihin kuin DHT22. Kuten kuvasta 2 näkee, on DHT22-anturi kooltaan kuitenkin vain kahden

euron kolikon kokoinen. Anturi häviää DHT22:lle kuitenkin etenkin lämpötilan mitausalueessa ja tarkkuudessa: käyttöalue on 0–50 °C ja tarkkuus +/- 2 °C. (Aosong Electronics Co., Ltd. 2013b.) Tähän opinnäytetyöhön valittiin DHT22-anturi nimenomaan paremman tarkkuuden ansiosta.



Kuva 2. DHT22-anturi (Kuva: Anssi Tuovinen).

DHT-22-anturissa on neljä liittintä, joiden kautta anturi liitetään anturilaitteeseen. Yksi näistä liittimistä ei ole käytössä. Taulukossa 1 on esitetty liittimien käyttötarkoitus ja kytkennät. Koska anturissa on vain yksi linja datan välittämiseen, täytyy sen toimia kahteen suuntaan. Tämän vuoksi anturin käyttöjännitteen ja data-linjan väliin on kytkettävä niin sanottu ylösvetovastus (engl. pull-up), joka varmistaa, että tuloliitännän signaali on aina oikea silloin kun signaalin suunta vaihtuu. (Aosong Electronics Co., Ltd.)

Taulukko 1. DHT-22-anturin liittimet

Selite	Liitin
V <sub>cc</sub>	1
DATA	2
Ei käytössä	3
GND	4

Tässä prototyypissä anturin mittaaman datan lukemiseksi käytetään Rob Tillaartin DHTlib-kirjastoa, jonka avulla voidaan lukea DHT11-, DHT22-, DHT33- ja DHT44-anturien lämpötila- ja kosteusarvot. Käytetty versio kirjastosta on 0.1.13.

Kirjasto tekee anturin arvojen lukemisen helposti: esimerkiksi DHT22-anturin arvojen lukeminen onnistuu helposti `read22()`-metodilla, jonka parametriksi annetaan Arduinon sen portin numero, johon DHT22-anturin datalinja on kytketty. Mahdollisessa virhetilanteessa anturi palauttaa virhekoodin. Tässä prototyypissä anturi palauttaa merkin "N" mikäli arvoa ei voida lukea. Näin käy esimerkiksi, jos anturi on viallinen, irrotettu koekytkentälevyltä tai anturin arvoja yritetään lukea liian nopeasti. (Tillaart 2015.)

### 4.3 Valaistuksen havainnointi

Valovastus (engl. *Light Dependent Resistor*, LDR) (kuva 3) on elektroniikan peruskomponentti. Se on vastus, jonka materiaali toimii eristeenä pimeässä. Mitä enemmän valoa vastuksen pinnalle pääsee, sitä vähemmän vastus vastustaa virran kulkua.



Kuva 3. LDR-vastus (Kuva: Anssi Tuovinen).

Tämän opinnäytetyön anturilaitteessa valovastus on kytketty Arduinon analogiseen sisääntuloon. Käyttöjännite on 5 V ja se saadaan suoraan Arduinon jännitelähteestä, ja valovastuksen ja maan välille on kytketty 10 kilo-ohmin vastus. Arduinon prosessorin 10-bittinen analogia-digitaalimuunnin muuntaa sisään tulevan jännitteen<sup>3</sup> arvoksi väliltä 0–1023. Esimerkiksi jos mitattu arvo on 500, on

---

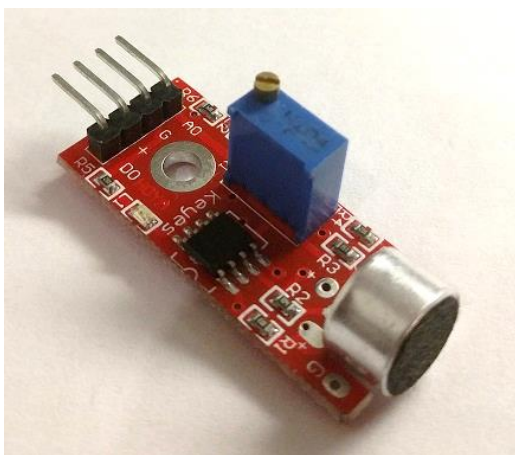
<sup>3</sup> Referenssijännite on Arduino Uno -alustassa oletuksena 0-5 V. Tätä voidaan muuttaa Arduinon `analogReference()`-metodilla.

sisään tuleva jännite tällöin n. 2,44 V. (Arduino 2015i.) Luonnollisesti myös valovastuksen ja maan välille kytketyn vastuksen vastusarvo vaikuttaa lukemaan.

Valovastuksen arvot eivät kuitenkaan ole lineaarisia, joten näin ollen valaistusvoimakkuuden laskeminen lukseina ei onnistu tarkasti ilman kalibrointia. Tämän vuoksi esimerkiksi fotodiodi olisikin parempi komponentti valaistuksen mittaamiseen, mutta esimerkiksi valaistuksen tason äkillisen muutoksen havaitsemiseksi valovastus on riittävä.

#### 4.4 Muut mittausmahdollisuudet

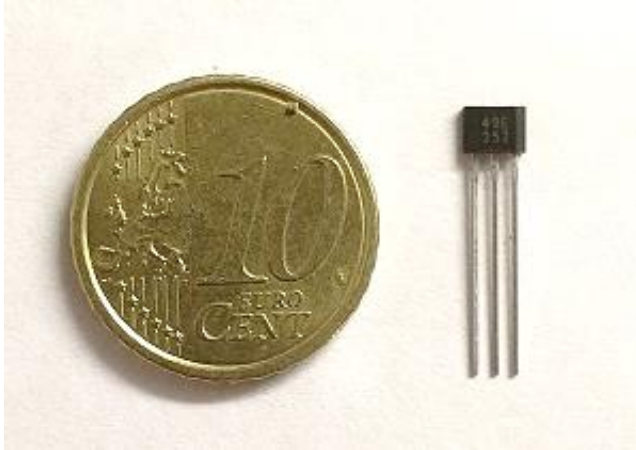
KY-038-anturia voidaan käyttää äänen havaitsemiseen. Anturissa on sekä digitaalinen että analoginen ulostulo. Digitaalinen ulostulo tekee anturista käytännössä kytkimen, joka aktivoituu määrätyn äänenvoimakkuuden ylitettyä. Anturin potentiometrillä voidaan säätää herkkyystasoa. Analogisen ulostulon kautta äänen tasoa voidaan havaita. (Tkkrlab 2015). Useimmat KY-038-anturit sisältävät potentiometrin, jolla digitaalisen ulostulon kynnyksarvoa voidaan säätää (kuva 4).



Kuva 4. KY-038-anturi (Kuva: Anssi Tuovinen).

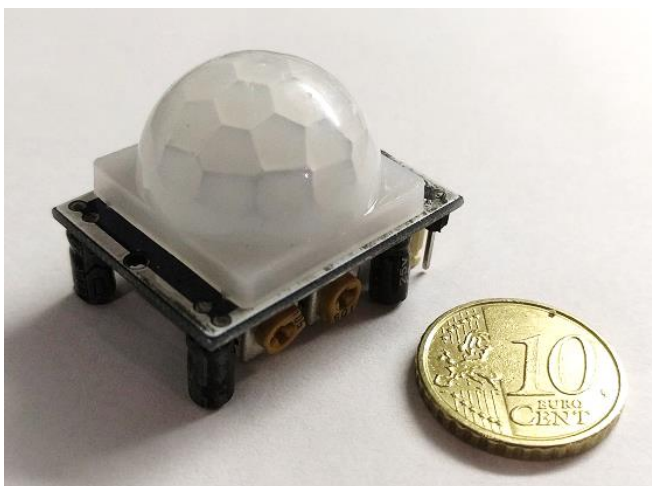
Hall-antureilla (kuva 5) voidaan muun muassa havaita ja mitata magneettikenttien voimakkuutta. Yksi merkittävä ja yleinen hall-anturin käyttökohde on autoissa, joissa sitä käytetään esimerkiksi nokka- ja kampiakselin asennon tunnistamiseen. Anturin toiminta liittyy hall-ilmiöön: magneettikentän vaikutuksesta an-

turiin syntyy jännite, jonka suuruus on verrannollinen magneettikentän suuruuteen. Hall-anturin merkittävä hyöty onkin siinä, että se ei perustu induktioon, jolloin sillä voidaan tunnistaa myös magneettikenttiä, jotka eivät ole liikkeessä. (SEC Electronics Inc 2008).



Kuva 5. Lineaarinen hall-anturi SS495 (Kuva: Anssi Tuovinen).

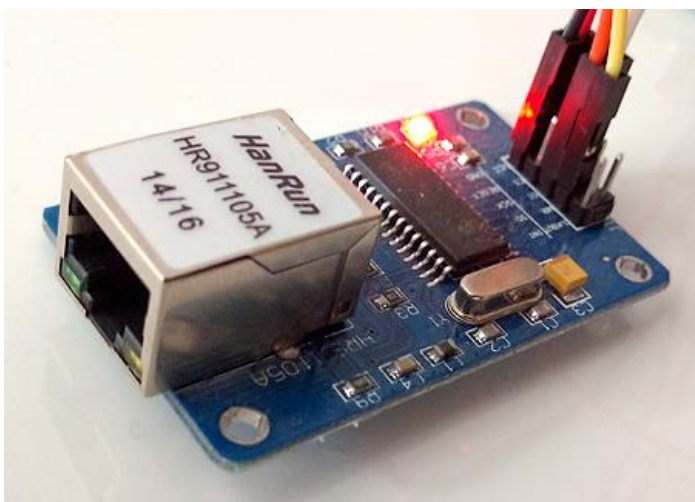
Toimistoympäristössä voitaisiin suhteellisen helposti mitata myös esimerkiksi sitä, että kuinka pitkään työntekijä on työpisteellään. Tähän tarkoitukseen voitaisiin käyttää erilaisia painetunnistimia tai -kytkimiä, mutta myös liiketunnistinta. Yleisin liiketunnistin on niin sanottu passiivitoiminen infrapunaanturi (engl. Passive Infrared, PIR). Digitaalinen anturi on tilassa 1 kun liikettä tunnistetaan ja tilassa 0 silloin, kun liikettä ei havaita. Herkkyyttä voidaan useissa tapauksissa säätää itse anturista (kuva 6). Käytännössä anturi pystyy tunnistamaan liikkeen noin 6 metrin päästä.



Kuva 6. PIR-anturi (Kuva: Anssi Tuovinen).

## 4.5 Verkkoysteys

Arduino-alusta voidaan yhdistää verkkoon joko langattomasti tai langallisesti. Markkinoilta on saatavilla sekä WLAN- että Ethernet-moduuleita. Tähän työhön valittiin edullinen HanRun Electronics Company Limitedin valmistama ENC28J60-siruun perustuva ethernet-moduuli (kuva 7).



Kuva 7. ENC28J60-Ethernet moduuli (Kuva: Anssi Tuovinen).

Tässä prototyypissä moduulin käyttämiseksi käytetään saksalaisen Norbert Truchsessin kehittämää UIPEthernet-kirjastoa, joka on välikerros ja tulkki Adam Dunkelsin uIP TCP/IP-pinolle (Truchsess 2015). Käytetyn kirjaston versionumero on versiota 1.09. Avoimen lähdekoodin uIP on kehitetty juurikin pitäen silmällä sulautettuja järjestelmiä, joten se käyttää todella vähän RAM-muistia (Dunkels 2013).

Moduuli liitetään Arduinoon käyttöjännitteen ja maan lisäksi neljällä johtimella. Alustassa on SPI-väylä (engl. Serial Peripheral Interface) ja SPI-protokollan mukaisesti jokainen johdin välittää neljää loogista signaalia, jotka ovat väylän kello-signaali (engl. serial clock, SCK), orjalaitteen sisään- ja ulostulosignaali SI ja SO (engl. slave in, slave out) sekä orjalaitteen valintesignaali (engl. slave select, SS/CS). Taulukossa 2 on listattu ne Arduinon portit, joita UIPEthernet-kirjasto signaaleille tarvitsee.

Taulukko 2. ENC28J60-kytkentä Arduino Uno -alustaan.

ENC28J60	Arduino
V <sub>CC</sub>	3,3 V
GND	GND
SCK	Pin 13
SI	Pin 12
SO	Pin 11
CS	Pin 8

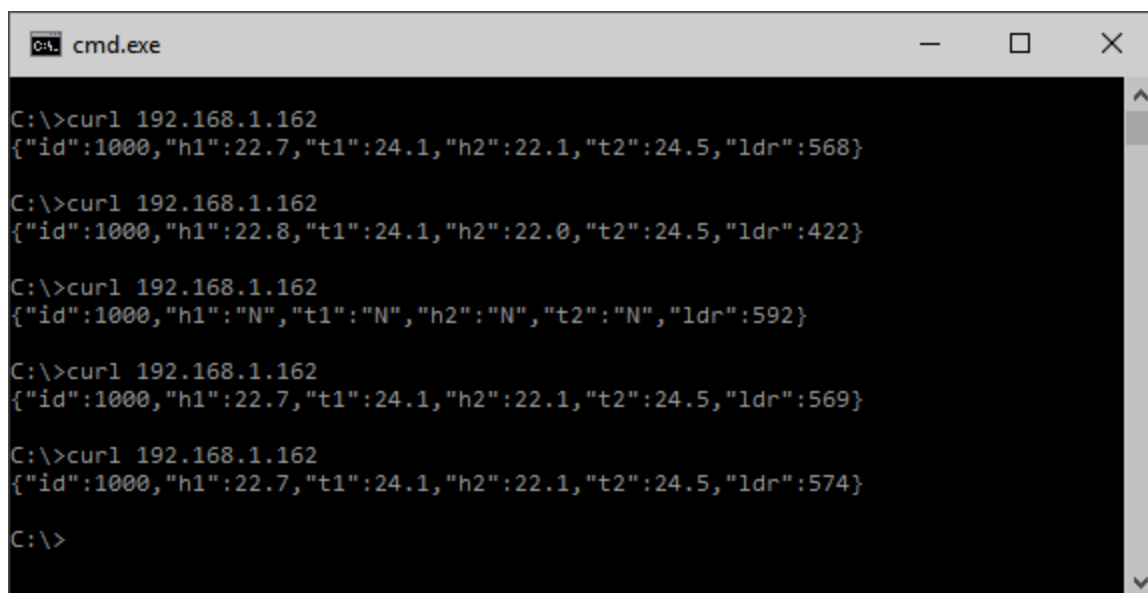
Arduinon ohjelmakoodissa UIPEthernet-kirjaston avulla rakennetaan TCP-palvelin, joka asetetaan kuuntelemaan tulevia yhteyksiä. Moduulille voitaisiin määrittää IP-osoite kiinteästi, mutta tässä tapauksessa lähiverkon reitittimen DHCP-palvelin myöntää laitteelle kiinteän IP-osoitteen 192.168.1.162. Kun laitteeseen lähetetään HTTP-protokollan mukainen GET-kutsu, hakee laite anturien tiedot ja lähettää ne takaisin JSON-muotoisena merkkijonona HTTP-vastauksena. Vastauksen parametrit on kuvattu taulukossa 3.

Taulukko 3. Laitteen vastausviestin parametrit.

Parametri	Kuvaus
id	Laitteen tunniste.
h1	Ensimmäisen DHT22-anturin mittaama ilman suhteellinen kosteus.
h2	Toisen DHT22-anturin mittaama ilman suhteellinen kosteus.
t1	Ensimmäisen DHT22-anturin mittaama ilman lämpötila.
t2	Toisen DHT22-anturi. mittaama ilman lämpötila.
ldr	Valovastuksen mittaama arvo.

Kuvan 8 esimerkissä näkyy neljä cURL-ohjelmalla lähetettyä GET-pyyntöä Arduinon IP-osoitteeseen ja JSON-vastaukset Arduinolta. DHT22-anturit sijaitsevat toisistaan vain n. 15 cm päässä, joten anturien mittaamien lämpötilojen ja kosteuksien erot johtuvat anturien tarkkuudesta, joka vaihtelee suuresti eri anturiyksilöiden välillä, mutta pysyy kuitenkin ilmoitetussa 2–5 %:n tarkkuudessa. Tämä on otettava huomioon kalibroimalla arvot ja tehtävä korjaukset joko Arduinon ohjelmakoodissa tai tietoja tietokantaan tallennettaessa.

Esimerkistä ilmenee myös selvästi se, miten valovastuksen arvo vaihtelee herkästi vain hieman anturin asentoa muuttamalla. Valovastuksen rakenteen vuoksi tulevan valon suunta vaikuttaa merkittävästi lukemaan, joten luotettavan valaistuksen tason mittaamiseksi tarvittaisiin useampia valovastuksia, joiden keskiarvosta voitaisiin päätellä valaistuksen taso. Esimerkkitapauksesta näkee myös, että liian nopeasti edellisen pyynnön jälkeen lähetetty uusi pyyntö palauttaa DHT22-anturin lukemille arvon "N".



```
cmd.exe
C:\>curl 192.168.1.162
{"id":1000,"h1":22.7,"t1":24.1,"h2":22.1,"t2":24.5,"ldr":568}
C:\>curl 192.168.1.162
{"id":1000,"h1":22.8,"t1":24.1,"h2":22.0,"t2":24.5,"ldr":422}
C:\>curl 192.168.1.162
{"id":1000,"h1":"N","t1":"N","h2":"N","t2":"N","ldr":592}
C:\>curl 192.168.1.162
{"id":1000,"h1":22.7,"t1":24.1,"h2":22.1,"t2":24.5,"ldr":569}
C:\>curl 192.168.1.162
{"id":1000,"h1":22.7,"t1":24.1,"h2":22.1,"t2":24.5,"ldr":574}
C:\>
```

Kuva 8. GET-pyyntöt Arduinolle cURL-ohjelmalla ja JSON-vastaukset. (Kuva: Anssi Tuovinen).

## 5 Tietokanta

Esineiden internet (engl. The Internet of Things, IoT) on pakottanu tietokantaratkaisuiden toimittajat pohtimaan omaa tuotetarjontaansa ja osoittamaan kyvykkyytensä myös IoT-ratkaisuiden parissa. Esimerkiksi markkinoiden suosituimmat<sup>4</sup> tietokannat Oracle ja MySQL kertovat verkkosivuillaan, miten heidän alustansa

<sup>4</sup> Tietokantojen suosiota mittaava itävaltalainen IT-yritys Solid IT ylläpitää db-engines.com sivustoa, jonka mukaan huhtikuussa 2016 kolme suosituinta tietokantaa olivat Oracle, MySQL ja Microsoft SQL Server (Solid IT 2016a). Suosion mittaaminen perustuu esimerkiksi Googlen hakutrendeihin ja tietokannan mainintojen määrään eri verkkosivuilla, työpaikkailmoituksissa tai Twitter-twiiteissä (Solid IT 2016b).



soveltuvat IoT-ympäristön tietokannaksi. Väittely SQL-tietokantojen – joita Oracle ja MySQL siis ovat – ja NoSQL-tietokantojen soveltuvuuden paremmuudesta suuren datan hallitsemiseksi on jatkunut jo vuosia.

Tässä opinnäytetyössä näiden eri ratkaisuiden ominaisuuksia ja kyvykkyyttä ei vertailla. Vaikka kumpikaan ratkaisu ei tarjoakaan täysin ongelmattonta ratkaisua, selvää on kuitenkin se, että NoSQL-tietokantojen suosio on nousussa (Solid IT 2016c).

### **Tietojen tallentaminen**

Tämän opinnäytetyön prototyyppilaitteiston tietokannaksi valittiin MongoDB, joka muun muassa MySQL:stä ja Oraclesta poiketen on niin sanottu NoSQL-tietokanta. DB-Engines-sivuston mukaan suosituin NoSQL-tietokanta on MongoDB Inc.:n kehittämä avoimen lähdekoodin dokumenttirakenteinen MongoDB-tietokanta (Solid IT, 2016a). Suomalainen tieto- ja viestintätekniikan merkittävä julkaisu Tietoviikko mainitsee MongoDB:n suosion syyksi tietokannan käytön aloittamisen helppouden ja edullisuuden sekä vahvasti vakiintuneen JSON-formaatin käyttämisen tietojen tallentamiseksi. Tietoviikko mainitsee yhdeksi todennäköiseksi syyksi myös sen, että MongoDB käyttää tietojen tallentamiseen JSON-formaattia. (Hovi 2015.)

Tiedot tallennetaan MongoDB-tietokantaan käyttäen Python-ohjelmaa. Ohjelma hakee tiedot anturilaitteen IP-osoitteesta käyttäen Pythonin Requests-kirjastoa.

### **Rajapinnat**

Asiakasohjelmat noutavat tiedot tietokannasta käyttäen rajapintaa, joka toteutettiin Pyramid-verkko-ohjelmistokehyksellä (engl. Web Framework). Pyramid on pieni, nopea ja yksinkertainen kehys (The Pylons Project 2016a.)

Asiakasohjelmat käyttävät HTTP-standardin mukaisia GET-pyyntöjä noutaakseen tietoja tietokannasta. Kutsut on rakennettu Pyramid-kehyksellä siten, että tietoja voidaan hakea mittauksen ajankohdan mukaan, joka määritellään GET-pyyntöissä URL-osoitteen polkuna. Pyynnön formaatti on:

http://[server:port]/api/[selector]/[year]/[month]/[day]/[hour]/[minute]/[second]

Hakuehdoista vain vuoden määrittely on pakollista. Muut aikamääritteet tarkentavat hakutulosta. Selector-parametrille annetaan arvoksi *one* tai *all* sen mukaan, halutaanko tietokannasta hakea kaikki kyseisellä hakuehdolla löytyvät tulokset vai ainoastaan ensimmäinen hakuehtoa vastaava tulos. Taulukossa 4 on kuvattu ja selitetty muutamia hakuesimerkkejä.

Taulukko 4. Rajapintakutsujen toiminta.

Kutsu	Kuvaus
.../all/2016/4	Kaikki mittaustulokset huhtikuussa 2016
.../all/2016/4/9/12	Kaikki mittaustulokset 9.4.2016 kello 12:00–12:59.
.../one/2016/4/9	Ensimmäinen mittaustulos 9.4.2016
.../one/2016/4/9/12	Ensimmäinen mittaustulos 9.4.2016 ajalta 12:00–12:59.

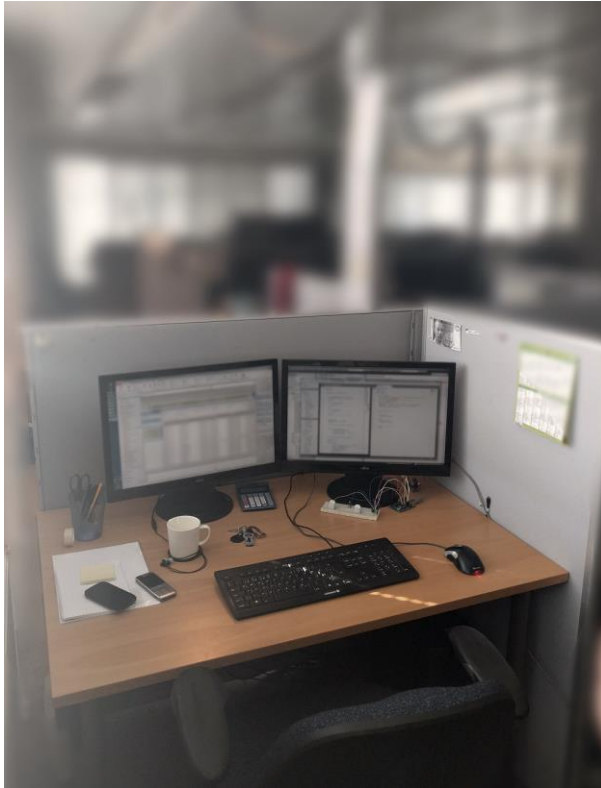
Rajapinnan toiminnassa on otettava myös tietoturva huomioon, mutta sen toteutus jätettiin tämän opinnäytetyön ulkopuolelle. Pyramid-kehyksessä on valmiit kirjastot ja logiikka käyttäjän tunnistamiselle (authentication) ja tunnistetun käyttäjän haetun sisällön käyttöoikeuksien tarkistamiselle (authorization) (The Pylons Project 2016b.)

## 6 Laitteen testaus

### 6.1 Testiympäristö

Laitteen toimivuutta testattiin henkilöstöpalveluyrityksen palvelukeskuksessa Joensuussa. Testiympäristönä oli niin sanottu perinteinen avotoimistoympäristö, jonka yhdelle työpisteistä laite sijoitettiin. Työpisteet on sijoitettu toimistossa ryhmiin ja ne on erotettu toisistaan kevytrakenteisilla välisermeillä.

Kuten kuvasta 9 käy osittain ilmi, on tilan yleisvalaistus koko tilaa ympäröivien suurten ikkunoiden ja loisteputkivalaisinrivistöjen vuoksi hyvä. Testikohteena oleva työpiste sijaitsee välittömästi ikkunan vieressä, joten ulkoa ikkunasta tuleva valaistus voi vaikuttaa merkittävästi työpisteen valaistukseen.



Kuva 9. Avotoimiston työpiste testiympäristönä (Kuva: Anssi Tuovinen).

Jo lähtökohtaisesti oli selvää, että laitteen prototyyppi on riittämätön mittaamaan lämpötilaa ja kosteutta tarpeeksi suurelta työpistettä ympäröivältä alueelta, jotta mittaustuloksista voitaisiin tehdä luotettavia päätelmiä työtilan lämpöoloista. Tämä vaatisi useamman lämpötila- ja kosteusanturien sijoittamisen eri puolille työpistettä, jotta esimerkiksi näyttöjen säteilemä lämpö vaikuttaisi tulokseen liikaa. Testauksen varsinaisena tarkoituksena olikin testata laitteen toimivuutta ja anturitiedon keräämistä mahdollisimman autenttisessa ympäristössä.

## 6.2 Luotettavuus ja käytettävyys

Laitteen käytettävyydessä suurimmaksi ongelmaksi nousee laitteen fyysinen koko, joka vaikeuttaa laitteen sijoittelua työpisteellä. Koekytkeäalustalle rakennettua prototyyppiä ei kuitenkaan ole vielä tässä vaiheessa tarkoitettukaan saada fyysisiltä mitoiltaan mahdollisimman pieneksi, eikä se kuulu tämän opinnäytetyön piiriin. Laitteen jalostus jätetään jatkokehityksen tehtäväksi.

Laitteen testaus aloitettiin noin kello 8:45 ja lopetettiin noin kello 14:00. Tiedot haettiin laitteelta sekunnin välein<sup>5</sup>, joten tietokantaan tallentui testauksen aikana yhteensä 15447 dokumenttia.

Ensimmäinen ja viimeinen dokumentti tietokannassa ovat:

```
{
  "_id" : ObjectId("570896d5a0777c079a292dca"),
  "h2" : 23.5,
  "timestamp" : "2016-04-09T08:44:53.939323",
  "h1" : 23.9,
  "t2" : 22.7,
  "t1" : 22.5,
  "ldr" : 673,
  "id" : 1000
}
{
  "_id" : ObjectId("5708dffba0777c079a296a20"),
  "h2" : 18.6,
  "timestamp" : "2016-04-09T13:56:59.932286",
  "h1" : 18.8,
  "t2" : 24.4,
  "t1" : 24.3,
  "ldr" : 657,
  "id" : 1000
}
```

Tietokannan koko on testauksen ajalta MongoDB:n *stats()*-funktion mukaan 1945 kilotavua.

```
> db.data.stats( { scale: 1024 } )

...
"count" : 15447,
"size" : 1945
```

<sup>5</sup> Sekunnin viiveen aikaansaamiseksi käytettyiin Python 3.4:n *time.sleep()*-funktioita.

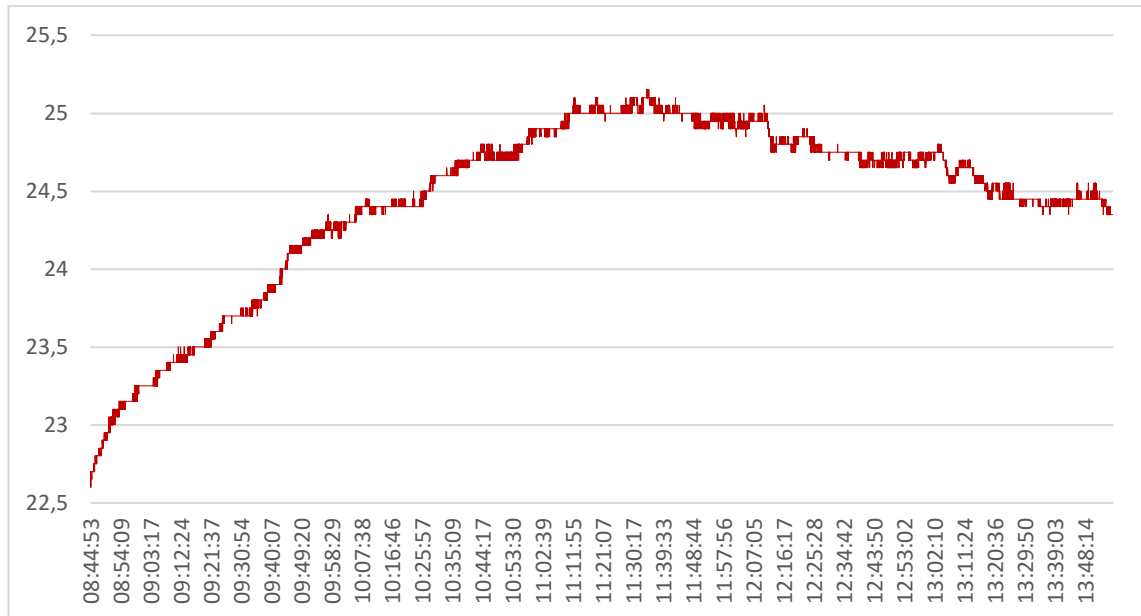
...

Laite toimi koko testausajan moitteetta, eikä ongelmia tietokannan toiminnan kanssa ollut. Toiminnan arvioinnissa on kuitenkin otettava huomioon se, että testauksen aikana yksikään asiakasohjelma ei hakenut tietoja rajapintojen kautta, joka saattaisi todellisessa käyttötilanteessa aiheuttaa merkittäväkin kuormitusta tietokantapalvelimelle.

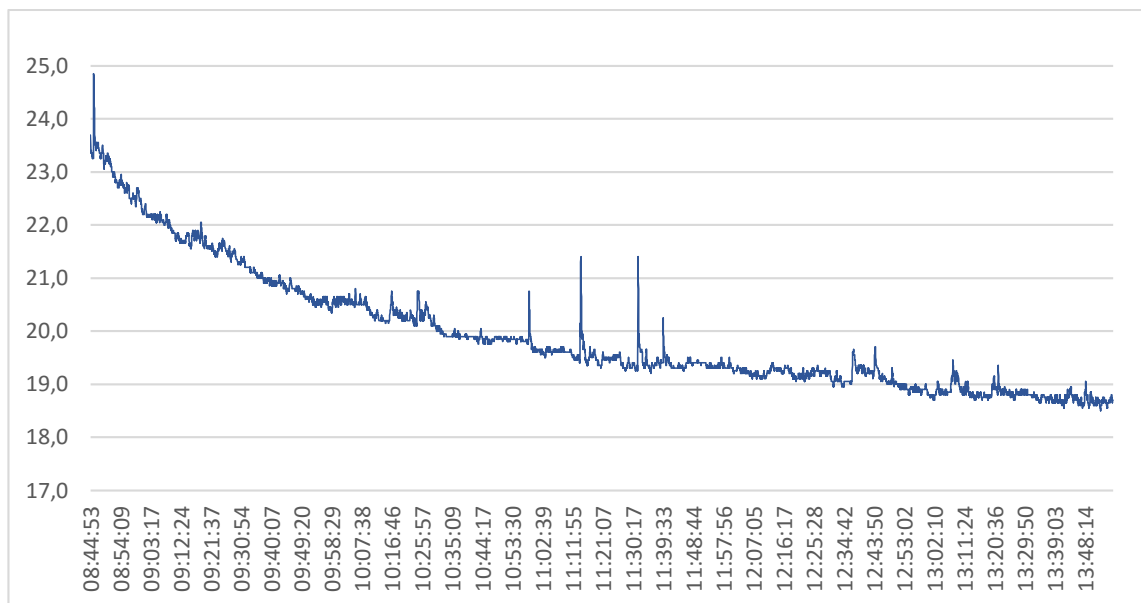
### 6.3 Datan hyödyntäminen

Testauksen aikana kävi nopeasti selväksi, että yhden sekunnin intervalli mittaus- tulosten hakemissa on tarpeettoman lyhyt toimistotyöympäristössä. Testiympäristössä on toiminnassa tavanomainen ilmastointi, joka ylläpitää tasaista tilan lämpötilaa, joten suuria äkillisiä ja aaltomaisia lämpötilanvaihteluita ei testausai- kana tapahtunut (kuvio 1). Todellisessa käyttötilanteessa toimistotyössä yhden celsiusasteen murto-osan muutosta ei työntekijä todennäköisesti edes havaitsisi. Näin ollen mittausintervalli voisi olla moninkertainen sekä tallennus tietokantaan olisi järkevää tehdä vain silloin, kun mitattu arvo eroaa merkittävästi esimerkiksi viimeisen tunnin keskiarvolämpötilasta.

Mittaustuloksista huomataan, että vaikka lämpötilassa ei tapahtunut äkillisiä muu- toksia, on lämpötila työpäivän aikana kuitenkin kohonnut merkittävästi. Lämpötila on työpisteellä jopa noin kaksi astetta korkeampi iltapäivällä kuin aamulla (kuvio 1). Tämä laskee luonnollisesti myös ilman suhteellista kosteutta (kuvio 2), joka jo lähtökohtaisesti oli alle Hengitysliiton 20 %:n suositusprosentin.



Kuvio 1. Lämpötila (°C) testiympäristössä testausaikana.

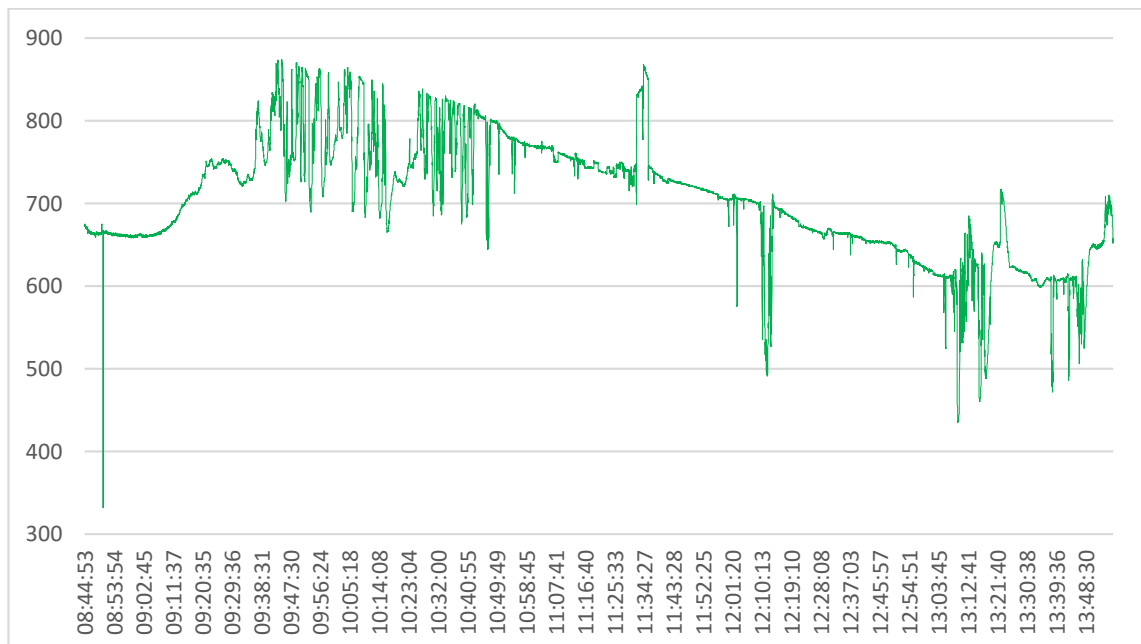


Kuvio 2. Ilman suhteellinen kosteus testiympäristössä testausaikana.

Ihminen sopeutuu nopeasti ympärillä oleviin muutoksiin. Tämä huomattiin myös testitilanteessa, sillä olosuhteiden muutosta ei havaittu muuten kuin anturidataa tarkkailemalla. Näin ollen voidaan olettaa, että olosuhdeseurannalla voidaan saavuttaa työhyvinvointia pitkällä aikavälillä parantavia tuloksia. On kuitenkin otettava huomioon, että tämän testisession mittaustulokset voivat selittyä ainakin osittain laitteen sijainnilla työpisteen kulmassa, jossa työpisteiden väliseiniä vuoksi etenkin sähkölaitteista johtuva lämpö ei välttämättä poistu työpisteeltä

yhtä tehokkaasti kuin muista työpisteen osista. Luotettavampi arviointi vaatisi uuden testijakson, jossa työpisteen lämpöoloja havainnoitaisiin ja mitattaisiin tarkemmin. Lämpötila pysyi kuitenkin pääsääntöisesti koko testijakson ajan Työsuojeluhallinnon suositusarvojen (Työsuojeluhallinto 2015a) 21–25 °C sisällä.

Valaistuksen tason mittaamiseksi käytetty valovastus osoittautui erittäin herkäksi pienillekin valaistuksen muutoksille. Kuvioista 3 on nähtävissä, miten taso vaihtelee hetkittäin merkittävästikin, sillä valovastus reagoi herkästi pieniinkin paikallisiin varjoihin ja heijastuksiin, joita aiheutui esimerkiksi työntekijöistä ja tietokoneen näytöistä. Useamman valovastuksen käytöllä ja niiden huolellisemmin suunnitellulla sijoittelulla saataisiin luotettavampia mittaustuloksia. Valovastuksen arvoa ei ole kalibroitu kertomaan valaistusvoimakkuuden suuruutta.



Kuvio 3. Valovastus valaistuksen tason indikaattorina.

## 7 Pohdinta

Rakennetun laitteen vaatimuksena oli olla toimiva kokonaisuus, joka sisältää tarvittavat komponentit ja ratkaisut todellisenkaltaista käyttötilannetta varten. Laitteen avulla piti pystyä mittaamaan luotettavasti ilman lämpötilaa ja suhteellista

kosteutta sekä arvioimaan ympäristön valaistuksen tasoa. Laitteen täytyi pystyä kommunikoidaan tietokantapalvelimen kanssa laitteen sijainnista riippumatta. Laitteen rakentaminen ja Arduinon ohjelmointi veivät suuren osan opinnäytetyön työajasta. Laitetta olisi ollut suhteellisen suoraviivaista laajentaa käyttämään muitakin hyödyllisiä antureita esimerkiksi melun ja hengitysilman laadun ja esimerkiksi siitepölyn havaitsemiseksi. Laitteen ominaisuuksia oli kuitenkin rajoitettava aikataulullisista ja taloudellisista syistä. Antureiden lisääminen olisi monimutkistanut Arduinon ohjelmakoodia, jolloin olisi ollut pohdittava enemmän muistia sisältävän alustan hankkimista tai työlästä ohjelmakoodin optimointia muistinkäytön osalta. Valmis prototyyppi oli kokonaisuudessaan puutteistaan huolimatta käyttökelpoinen kokonaisuus, jolla mittaustulokset saatiin kerättyä. Laitteen puutteiden korjaaminen jätetään jatkokehityksen tehtäväksi.

Havaintoni vastaavan laitteen haasteista ovat merkittävässä määrin yhtenevät jo vuonna 2011 julkaistun Beira Interior -yliopiston tutkimuksen kanssa, jossa tutkittiin langattomien anturiverkkojen käyttömahdollisuuksia. Tutkimuksessa merkittäviksi anturiverkkojen käyttökelpoisuuden haasteiksi mainitaan muun muassa virranhallinta ja -saatavuus, käytettävyyys ja fyysinen koko. (Oliveira & Rodrigues 2011, 148–149). Laitteen jatkokehityksen ensimmäinen selkein askel olisikin mielestäni laitteen muuttaminen langattomaksi, joka parantaisi laitteen käytettävyyttä huomattavasti. Eri paristosta virtansa saavat anturit voisivat keskustella Arduino UNO:n kanssa RF-signaaleihin, ja tietokantapalvelin hakisi tiedot Arduinolta WLAN-yhteyden kautta. Tämä lisäisi laitteen käyttömahdollisuuksia ja joustavuutta sekä mahdollistaisi anturien vapaamman sijoittelun, joka parantaisi merkittävästi myös mittaustulosten hallinnointia ja luotettavuutta.

Laitteen fyysinen koko sekä koekytkentäalustalle tehdyt kytkennät rajoittavat ja vaikeuttavat laitteen käyttöä. Koon pienentämiseksi helpoin tapa olisi suunnitella ja rakentaa antureille oma piirilevy, jonka voisi kytkeä niin sanotun *breakout boardin* tavoin suoraan Arduino UNO:n liitäntöihin. Koska Arduino toteuttaa avointen laitteistojen periaatetta, on kaikki Arduinon vaatimat komponentit vapaasti saatavilla. Näin ollen laite saataisiin vielä pienemmäksi, jos sille suunniteltaisiin kokonaan oma piirilevy aivan Arduino-alustasta saakka. Tällöin voitaisiin komponenttivalinnoilla vaikuttaa myös virrankulutukseen.



Merkittävä käytettävyyteen vaikuttava haaste, jonka ratkaisemisella olisi anturi-verkkoratkaisuiden tulevaisuuden kannalta suuri vaikutus, on standardisoinnin puute, johon myös Beira Interior -yliopiston tutkimuksessa viitataan (Oliveira & Rodrigues 2011, 149). Viime vuosina markkinoille onkin tullut saataville lukuisia erilaisia ja eri käyttötarkoitukseen tarkoitettuja laitekokonaisuuksia, mutta usein edelleen laitteiden ongelmana on eri komponenttien ja toisten ratkaisuiden liitettävyys, muokattavuus ja toimivuus eri laitteiden ja antureiden kesken. Tähän ongelmaan pyrkii vastaamaan esimerkiksi suomalainen vuonna 2013 perustettu Cozify Oy, jonka järjestelmä mahdollistaa eri valmistajien laitteiden yhdistämisen ja hallinnan Cozifyn tarjoaman sovelluksen kautta.

Esineiden internet on kiinnostanut minua jo kauan, joten teemaan liittyvää opinnäytetyötä oli erittäin antoisaa ja mielenkiintoista tehdä. Arduino oli minulle jo ennestään tuttu, mutta tämän työn yhteydessä pääsin toteuttamaan kokonaisuuden, jonka yhteydessä opin myös paljon uutta. Vaikka laitteessa käytetyt komponentit ja kytkennät ovatkin pääsääntöisesti peruselektroniikka, oli laitteen rakentaminen erittäin mielenkiintoinen prosessi. Etenkin ENC28J60-moduulin kytkennän ja ohjelmoinnin onnistuminen edellytti perehtymistä muun muassa SPI-protokollan toimintaan ja yksinkertaisen TCP-palvelimen logiikkaan. Yksittäisten asioiden kuten Arduinon tarkemman rakenteen ja Python-kielen säikeistyksen lisäksi sain myös osittaisen kuvan yksinkertaisen IoT-tuotteen toteuttamisesta, vaikka kyseessä olikin vain yksinkertainen laitteen prototyyppi.

Opinnäytetyön kokonaisuuden hallinta onnistui mielestäni hyvin, sillä työ eteni suunnitelman mukaan, havaitut ongelmat ratkaistiin nopeasti ja tehokkaasti. Tähän vaikutti myös työlle varattu aikatauluni, joka tarvittaessa antoi joustoa muiden opiskelu- ja työelämän haasteiden kohtaamista varten. Työn projektimaisen luonteen vuoksi ketterät kehitysmenelmät olisivat voineet soveltua työn tekemiseen ja hallintaan erinomaisesti, mutta lopulta perinteinen vesiputousmalli osoittautui selkeämmiksi projektin hallinnan kannalta erityisesti siitä syystä, että aikataulullisia haasteita ja ehdottomia aikataulumääritteitä ei ollut. Työn yhteydessä tutustuin kuitenkin myös useisiin erilaisiin projektinhallinta- ja tehtävälisätyökaluihin,

joiden käytön kuvaus jätettiin luonnollisesti tämän opinnäytetyön aiheen ulkopuolelle. Uskon kuitenkin, että myös itse opinnäytetyön työstämisen kokemuksesta – kaikkine asiaan kuuluvine ja kuulumattomine tekijöineen – on minulle hyötyä myöhemmissä projekteissani niin työssä kuin vapaa-ajallakin.

## Lähteet

- Aosong Electronics Co., Ltd. 2013a. AM2302.  
<http://www.aosong.com/en/products/details.asp?id=117>. 6.3.2016.
- Aosong Electronics Co., Ltd. 2013b. DHT11.  
<http://www.aosong.com/en/products/details.asp?id=109>. 6.3.2016.
- Arduino. 2015a. Introduction. <https://www.arduino.cc/en/Guide/Introduction>.  
9.11.2015.
- Arduino. 2015b. SimilarBoards.  
<http://playground.arduino.cc/Main/SimilarBoards>. 28.10.2015.
- Arduino. 2015c. Products. <https://www.arduino.cc/en/Main/Products>.  
28.10.2015.
- Arduino. 2015e. ArduinoBoardUno.  
<https://www.arduino.cc/en/Main/ArduinoBoardUno>. 9.11.2015.
- Arduino. 2015d. GenuinoBrand. <https://www.arduino.cc/en/Main/GenuinoBrand>.  
28.10.2015.
- Arduino. 2015e. ArduinoBoardUno.  
<https://www.arduino.cc/en/Main/ArduinoBoardUno>. 9.11.2015.
- Arduino. 2015f. arduinoBoardMega.  
<https://www.arduino.cc/en/Main/arduinoBoardMega>. 28.10.2015.
- Arduino. 2015g. ArduinoBoardYun.  
<https://www.arduino.cc/en/Main/ArduinoBoardYun>. 28.10.2015.
- Arduino. 2015h. Arduino 101.  
<https://www.arduino.cc/en/Main/ArduinoBoard101>. 2.11.2015.
- Arduino. 2015i. AnalogRead.  
<https://www.arduino.cc/en/Reference/AnalogRead>. 7.3.2016.
- Aura, O., Ahonen, G., Ilmarinen, J. 2009–2012. Strategisen hyvinvoinnin tila Suomessa. Helsinki: Excenta Oy, Pohjola Op.
- B\_E\_N. SparkFun Tutorials, What is an Arduino?  
<https://learn.sparkfun.com/tutorials/what-is-an-arduino/the-arduino-family>. 28.10.2015.
- Dunkels, A. 2013. uIP. <https://github.com/adamdunkels/uip>. 7.3.2016.
- Hägglom, H., Haapakangas, A., Hongisto, V. & Koskela, H. 2011. LÄMPÖTILAN VAIKUTUS TYÖSUORIUTUMISEEN JA VIIHTYVYYTEEN - LABORATORIOTUTKIMUS TOIMISTOOLOSUHTEISSA.  
[http://www.ttl.fi/fi/tutkimus/hankkeet/toti/Documents/hagglom\\_sisailma\\_stoseminaari\\_2011.pdf](http://www.ttl.fi/fi/tutkimus/hankkeet/toti/Documents/hagglom_sisailma_stoseminaari_2011.pdf). 14.2.2016.
- Hovi, A. 2015. MongoDB haastaa relaatiokantoja. Tietoviikko 2015/10 s. 50–51.
- Lowen, A., Mubareka, S., Steel, J., Palese, P. 2007. Influenza Virus Transmission Is Dependent on Relative Humidity and Temperature. PLoS Pathog.
- Oliveira, L., Rodrigues, J. 2011. Wireless Sensor Networks: a Survey on Environmental Monitoring. Journal of Communications, vol. 6 nro. 2.
- SEC Electronics Inc. 2008. SS49E Linear Hall Effect Sensor.  
[https://dscl.lcsr.jhu.edu/main/images/3/31/SS49e\\_Hall\\_Sensor\\_Datasheet.pdf](https://dscl.lcsr.jhu.edu/main/images/3/31/SS49e_Hall_Sensor_Datasheet.pdf). 7.3.2016
- Solid IT. 2016a. DB-Engines Ranking. <http://db-engines.com/en/ranking>.  
2.4.2016

- Solid IT. 2016b. Method of calculating the scores of the DB-Engines Ranking.  
[http://db-engines.com/en/ranking\\_definition](http://db-engines.com/en/ranking_definition). 2.4.2016.
- Solid IT. 2016c. DBMS popularity broken down by database model. [http://db-engines.com/en/ranking\\_categories](http://db-engines.com/en/ranking_categories). 2.4.2016
- The Pylons Project. 2016a. The Pyramid Web Framework.  
<http://docs.pylonsproject.org/projects/pyramid/en/latest/index.html>.  
22.4.2016.
- The Pylons Project. 2016b. Security.  
<http://docs.pylonsproject.org/projects/pyramid/en/latest/narr/security.html>. 22.4.2016.
- Tillaart, R. 2015. DHTlib.  
<https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib>.  
6.3.2016.
- Tkkrlab. 2015. Arduino KY-038 Microphone sound sensor module .  
[https://tkkrlab.nl/wiki/Arduino\\_KY-038\\_Microphone\\_sound\\_sensor\\_module](https://tkkrlab.nl/wiki/Arduino_KY-038_Microphone_sound_sensor_module). 7.3.2016
- Truchsess, N. 2015. Arduino\_uip. [https://github.com/ntruchsess/arduino\\_uip](https://github.com/ntruchsess/arduino_uip).  
7.3.2016.
- Työsuojeluhallinto. 2015a. Lämpöolot.  
<http://www.tyosuojelu.fi/tyoolot/fysikaaliset-tekijat/lampoolot>. 14.2.2015
- Työsuojeluhallinto. 2015b. Valaistus.  
<http://www.tyosuojelu.fi/tyoolot/fysikaaliset-tekijat/valaistus>. 10.4.2016.
- Työterveyslaitos. Työllisten ammattitaudit tautiryhmittäin ja vuosittain.  
[http://www.ttl.fi/fi/tilastot/tyotaturmat\\_ammattitaudit\\_ja\\_sairauspoissaolot/Sivut/ammattitaudit\\_tautiryhmittain\\_ja\\_vuosittain.aspx](http://www.ttl.fi/fi/tilastot/tyotaturmat_ammattitaudit_ja_sairauspoissaolot/Sivut/ammattitaudit_tautiryhmittain_ja_vuosittain.aspx). 14.2.2016.
- Työterveyslaitos. 2013. Työ ja terveys Suomessa 2012.
- Työterveyslaitos. 2014. Lämpöviihtyvyys.  
[http://www.ttl.fi/fi/tyoymparisto/lampoolot/lampoviihtyisat\\_olot/Sivut/default.aspx](http://www.ttl.fi/fi/tyoymparisto/lampoolot/lampoviihtyisat_olot/Sivut/default.aspx). 10.4.2016.
- Työturvallisuuslaki 738/2002.
- Valtioneuvoston asetus työntekijöiden suojelemiseksi optiselle säteilylle altistumisesta aiheutuvilta vaaroilta 146/2010.
- World Health Organization. 1999. Guidelines For Community Noise.

## Arduinon ohjelmakoodi

```
#include <UIPEthernet.h>
#include <dht.h>

// DEVICE SETTINGS
#define DEVICE_ID 1000
#define DEBUG_LED_PIN 7 // Vihreä led

// SENSOR SETTINGS
#define DHT22_PIN1 2
#define DHT22_PIN2 4
#define LDR_PIN A0

// Struct DHT22-antureille
typedef struct dht22 {
    dht DHT;
    String h;
    String t;
    byte lastStatus;
} DHT22;

DHT22 dht22_1; // 1. DHT-22
DHT22 dht22_2; // 2. DHT-22

// ETHERNET SETTINGS
// Reitittimen DHCP jakaa tietyn IP:n MAC:n perusteella.
byte mac[] = { 0x54, 0x34, 0x41, 0x30, 0x30, 0x31 };
EthernetServer server(80);

// SETUP() AND LOOP()

// Arduino suorittaa tämän ensimmäiseksi
void setup() {
    // Pin output-tilaan ja oletuksena LOW
    pinMode(DEBUG_LED_PIN, OUTPUT);
    digitalWrite(DEBUG_LED_PIN, LOW);

    // Palvelin päälle
    Ethernet.begin(mac);
    server.begin();

    //Serial, debuggausta varten
    //Serial.begin(9600);
}

// Arduinon mainloop
void loop() {

    // Asettaa Arduinon kuuntelemaan tulevia yhteyksiä
    EthernetClient client = server.available();

    if (client)
    {
        // Apumuuttuja HTTP-pyyntöön päättymisen tulkitsemista varten.
        // HTTP-pyyntö päättyy tyhjään riviin...
        boolean lineIsBlank = true;
```

```
// Kun kutsu saapuu...
while (client.connected())
{
  // client.available() palauttaa palvelimen pyynnön koon,
  // joten pyyntö käsitellään.
  if (client.available())
  {
    // Luetaan pyynnön sisältö
    char c = client.read();

    // Tulostaa Serial Monitoriin pyynnön sisällön.
    // Debuggausta varten vain, koska vie muistia!
    // Serial.write(c);

    // HTTP-kutsu on päättynyt, jos rivi on tyhjä.
    // Luetaan anturit ja muodostetaan vastaus.
    if (c == '\n' && lineIsBlank)
    {
      // LED päälle
      digitalWrite(DEBUG_LED_PIN, HIGH);

      // READ DHT-22 DATA
      // read22() palauttaa 0 mikäli anturin luku onnistuu
      dht22_1.lastStatus = dht22_1.DHT.read22(DHT22_PIN1);
      dht22_2.lastStatus = dht22_2.DHT.read22(DHT22_PIN2);
      //Serial.println(dht22_1.lastStatus);
      //Serial.println(dht22_2.lastStatus);

      char tempb[5]; // Väliaikainen bufferi dtostrf()-funktiolle
      String jsonData; // JSON-dataa varten. String vie enemmän muistia,
      mutta helpompi käyttää...

      // 1. DHT-22-anturin arvojen käsittely
      if (dht22_1.lastStatus != 0) {
        dht22_1.h = "\"N\"";
        dht22_1.t = "\"N\"";
      }
      else {
        dtostrf(dht22_1.DHT.humidity, 3, 1, tempb);
        dht22_1.h = String(tempb);
        dtostrf(dht22_1.DHT.temperature, 3, 1, tempb);
        dht22_1.t = String(tempb);
      }

      // 2. DHT-22-anturin arvojen käsittely
      if (dht22_2.lastStatus != 0) {
        dht22_2.h = "\"N\"";
        dht22_2.t = "\"N\"";
      }
      else {
        dtostrf(dht22_2.DHT.humidity, 3, 1, tempb);
        dht22_2.h = String(tempb);
        dtostrf(dht22_2.DHT.temperature, 3, 1, tempb);
        dht22_2.t = String(tempb);
      }

      // LDR-anturin arvojen käsittely
      int ldr_value = analogRead(LDR_PIN);

      // Vastauksen muodostus (JSON)
```

```
jsonData =
  "{\"id\":" +
  String(DEVICE_ID) +
  "\",\"h1\":" +
  dht22_1.h +
  "\",\"t1\":" +
  dht22_1.t +
  "\",\"h2\":" +
  dht22_2.h +
  "\",\"t2\":" +
  dht22_2.t +
  "\",\"ldr\":" +
  ldr_value +
  "\"}";

// HTTP-RESPONSE
client.println(F("HTTP/1.1 200 OK"));
client.println(F("Content-Type: application/json;charset=utf-8"));
client.println(F("Connection: close"));
client.println();
client.println(jsonData);
break;
}

if (c == '\n') {
  // Jos rivinvaihto, mutta kutsussa ei tyhjää riviä => pyyntö saattaa
  jatkaa...
  lineIsBlank = true;
}

else if (c != '\r')
{
  // Kutsu jatkuu
  lineIsBlank = false;
}
}
}
}
delay(10);

// Suljetaan yhteys ja sammutetaan led
client.stop();
digitalWrite(DEBUG_LED_PIN, LOW);
}
```