

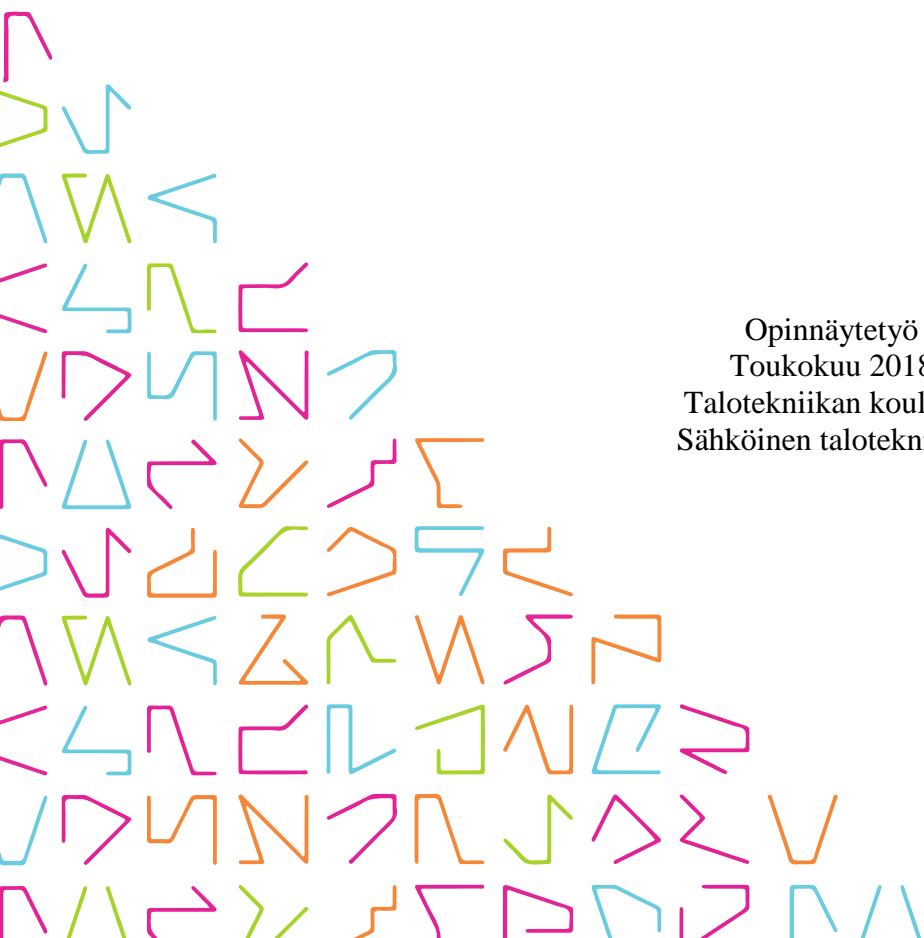


TAMPEREEN
AMMATTIKORKEAKOULU

RAKENNUSAUTOMAATIO- JA IoT-OPPIMISYMPÄRISTÖN DEMONSTRAATION TOTEUTUS

Juha-Petteri Sormunen

Opinnäytetyö
Toukokuu 2018
Talotekniikan koulutus
Sähköinen talotekniikka



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Talotekniikan koulutus
Sähköinen talotekniikka

SORMUNEN, JUHA-PETTERI:

Rakennusautomaatio- ja IoT-oppimisympäristön demonstraation toteutus

Opinnäytetyö 43 sivua, joista liitteitä 3 sivua
Toukokuu 2018

Tässä työssä esitellään IoT- ja rakennusautomaatoratkaisut, jotka kehitettiin Tampereen ammattikorkeakoulun talotekniikan oppimisympäristön kehittämistä varten. Talotekniikan oppimisympäristön tarkoituksena on esittää mittauksilla sisäilmaston olosuhteissa tapahtuvia muutoksia, kun olosuhteisiin vaikuttavia prosesseja säädetään eri tavoilla.

Demonstraatioympäristön käyttöjärjestelmä toteutettiin Wapice Oy:n IoT-Ticket -alustalla. IoT-Ticketiin luotiin näkymät erilaisille mittauksille ja tilaa palvelevan ilmanvaihtokoneen mittaus- ja säätötiedoille. Mittausdatan keräämisessä käytettiin IoT-Ticketin WRM 247+ -laitetta, johon liitettiin kentälaitteita Modbus RTU -väylällä.

Opinnäytetyön tuloksena syntyi ratkaisu, jonka pohjalta oppimisympäristöön voidaan toteuttaa mittauksien tiedonkeruu. IoT-Ticketiin luotiin mittausnäyttöikkunat, joiden avulla voidaan toteuttaa lopullinen käyttöliittymä, kun kaikki tarvittavat kentälaitteet on saatu asennettua. Lisäksi työstä laaditun dokumentoinnin avulla oppimisympäristöä voidaan jatkojalostaa esimerkiksi uusien opinnäytetöiden avulla.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Building Services Engineering
Electrical Building Services

SORMUNEN, JUHA-PETTERI:

A Demonstration of the Learning Environment of Building Automation and IoT

Bachelor's thesis 43 pages, appendices 3 pages
May 2018

The objective of this thesis was to develop a demo environment for the learning of building automation and the Internet of Things. The learning environment was meant to be used for the teaching purposes in Building Services Engineering at Tampere University of Applied Sciences. The learning environment will be used for measuring and presenting the quality of indoor climate conditions when heating, ventilation and air conditioning (HVAC) processes are regulated in different ways.

The user interface of the demo environment was engineered with the IoT-Ticket platform developed by Wapice Ltd. Different kind of dashboards were made for measurements and for the measurement and regulation information of the air handling unit. The data was collected with the IoT-Ticket WRM 247+ device via the Modbus RTU protocol.

The demo environment produced a solution, which will be used for the data acquisition in the final execution of the learning environment. The demo dashboards will be used for the base of the final user interface. In addition, the notes that were collected during the study will assist in the further development of the learning environment.

Key words: building automation, Internet of Things, learning environment

SISÄLLYS

1	JOHDANTO.....	7
2	TALOTEKNIIKAN OPPIMISSYMPÄRISTÖ	8
2.1	Oppimisympäristön kuvaus	8
2.2	Tavoitteet oppimisympäristölle	9
2.2.1	Painesuhdemittaukset.....	9
2.2.2	Lämpötilamittaukset.....	9
2.2.3	Ilmanlaatumittaukset.....	10
2.2.4	Energiamittaukset.....	10
2.2.5	Valaistuksen ohjaus.....	11
2.3	Opinnäytetyön tavoitteet ja tarkoitus	11
3	PILVIPALVELUIDEN KÄYTTÖ TALOTEKNIKASSA	12
3.1	Talotekniikan IoT	12
3.2	IoT:n hyödyntäminen talotekniikassa	13
4	IoT-TICKET -ALUSTA.....	14
4.1	IoT-Ticketin yleiskuvaus	14
4.2	IoT-Ticketin käyttöliittymä.....	14
4.2.1	Interface Designer -työkalu.....	15
4.2.2	Dataflow Editor -työkalu.....	16
4.3	Tiedonkeruu IoT-Ticketillä.....	17
4.3.1	IoT-Ticket API -rajapinnat.....	17
4.3.2	WRM 247+ -laite	17
5	MODBUS-PROTOKOLLA	19
5.1	Modbus-väyläprotokollan yleiskuvaus	19
5.2	Modbus-väylän rakenne.....	19
5.3	Sarjaliikenne Modbus-väylässä	20
5.4	Isäntä/orja-protokolla.....	21
5.5	Modbus-rekisterien käyttö	21
6	DEMOYMPÄRISTÖN FYYNINEN TOTEUTUS.....	23
6.1	Demoympäristön järjestelmäkuvaus.....	23
6.1.1	Demoympäristön järjestelmäkaavio.....	23
6.1.2	Demoympäristön laitteet	24
6.1.3	Demoympäristön sähkökytkennät.....	24
6.1.4	Demoympäristön Modbus-väylä.....	25
6.2	WRM 247+ -laitteen liittäminen ja konfigurointi.....	25
6.3	Solmujen lisääminen IoT-Ticketiin	26
7	DEMOYMPÄRISTÖN KÄYTTÖLIITTYMIEN RAKENTAMINEN	27

7.1	Paine-eron ja pintalämpötilan käyttöliittymä.....	27
7.1.1	Paine-eron ja pintalämpötilan mittarit.....	27
7.1.2	Aikaleiman esittäminen.....	28
7.2	Olosuhdemittausten käyttöliittymä.....	29
7.2.1	Olosuhdemittausten mittarit.....	29
7.2.2	Olosuhdemittausten trendiseurannat.....	30
7.2.3	Rajojen lisääminen hiilidioksidimittaukseen.....	30
7.3	Ilmankäsittelykoneen käyttöliittymä.....	31
7.3.1	Ilmankäsittelykoneen tietojen esitys.....	32
7.3.2	Hälytysten indikoinnit.....	33
7.3.3	Tilatiedon indikointi.....	34
8	TYÖN AIKANA SELVITETTYJÄ ONGELMATILANTEITA.....	36
8.1	Modbus-väylän ongelmat.....	36
8.1.1	Laitteen ID-numero 1 ei toimi.....	36
8.1.2	Mittaustieto ei näytä oikein.....	36
8.2	Ethernet-verkon ongelmat.....	37
8.3	Mittaustieto ei päivittynyt käyttöliittymään.....	37
9	POHDINTA.....	38
	LÄHTEET.....	40
	LIITTEET.....	41
	Liite 1. Oppimisympäristön järjestelmäkaavio.....	41
	Liite 2. Demoympäristön johdotuskaavio.....	42
	Liite 3. Demoympäristön väyläkaapelointi.....	43

ERITYISSANASTO

API	ohjelmointirajapinta, jolla eri ohjelmat voivat keskustella keskenään (Application programming interface)
IoT	esineiden internet (Internet of Things)
ID	laitteen yksilöllinen tunniste (identifier)
logata	tallentaa mittaustietoa
Modbus RTU	sarjaliikenneprotokolla, jonka tiedonsiirto perustuu isäntä/orja -kommunikaatioon
protokolla	standardoitu tapa, jolla samaa tiedonsiirtotapaa noudattavat laitteet tai ohjelmistot keskustelevat keskenään
sarjaliikenne	kommunikaatitapa siirtää tietoa tietokoneiden ja elektronisten laitteiden välillä
solmu	tietoverkon itsenäinen perusosa, esimerkiksi laite tai datapiste, joka sisältää dataa (node)

1 JOHDANTO

Tässä työssä esitellään IoT- ja rakennusautomaatioratkaisut, jotka kehitettiin Tampereen ammattikorkeakoulun talotekniikan oppimisympäristön kehittämistä varten. Talotekniikan oppimisympäristön tarkoituksena on esittää mittauksilla sisäilmaston olosuhteissa tapahtuvia muutoksia, kun olosuhteisiin vaikuttavia prosesseja säädetään eri tavoilla. Oppimisympäristössä on tarkoitus teettää opiskelijoilla erilaisia talotekniikan prosesseihin ja ilmiöihin liittyviä harjoituksia.

Työn ensimmäisessä osiossa esitellään työn lähtötiedot talotekniikan opetuksen tarpeiden näkökulmasta. Siinä kerrotaan mitä oppimisympäristössä halutaan opettaa ja mitä prosesseja halutaan tutkia mittauksien avulla, sekä millä ratkaisulla tiedot hankitaan. Lähtötietojen ja aiheen laajuuden perusteella opinnäytetyön aihe rajattiin käsittelemään demonstraatioympäristöä, jonka pohjalta voidaan toteuttaa lopullinen oppimisympäristö. Työn tavoitteeksi muodostui keskeisimpien laitteiden käyttöönotto ja liittäminen oppimisympäristön käyttöliittymään sekä käyttöliittymän luonti.

Toisessa osiossa kerrotaan pilvipalveluiden ja esineiden internetin (IoT) merkityksestä nykypäivän talotekniikassa. Osiossa esitellään, miten pilvipalveluita voidaan hyödyntää nykyään taloteknisten järjestelmien ohjaamiseen ja analysointiin. Lisäksi kolmannessa osiossa esitellään työssä käytettyä Wapice Oy:n IoT-Ticket alustaa.

Neljännessä osassa tarkastellaan Modbus-sarjaliikenneprotokollaa. Osiossa esitetään perustiedot Modbus RTU kommunikoinnista, sen tiedonsiirtotavasta ja siitä miten tieto liikkuu Modbus RTU –väylässä ja miten tietoa käsitellään Modbus-laitteessa.

Teorian jälkeen tarkastellaan työssä rakennettua fyysistä ja virtuaalista talotekniikan demoympäristöä, joka toteutettiin IoT-Ticketin alustalla ja siihen liitetyllä WRM 247+ -laitteella. Osiossa selvitetään työssä tuotetut ratkaisut ja käytetyt laitteet. Lisäksi esitellään työnaikana esiin tulleita ongelmia, jotka saatiin ratkaistua. Työn liitteissä on kaavioita demoympäristöstä ja sen rakenteesta.

2 TALOTEKNIIKAN OPPIMISSYMPÄRISTÖ

2.1 Oppimisympäristön kuvaus

Talotekniikan oppimisympäristö käsittää laboratoriotilat, sinne sijoitettavat talotekniset mittaus- ja ohjauslaitteiston, tiedonsiirtotavat ja IoT-Ticketin avulla toteutettavan käyttöliittymän. Oppimisympäristön luokkatila on nähtävissä kuvassa 1.



KUVA 1. Oppimisympäristön luokkatila

2.2 Tavoitteet oppimisympäristölle

Oppimisympäristön tavoitteet määräytyvät TATE-labran kehitystyöryhmän yhteisten pohdintojen pohjalta. Laboratorion ensisijainen tavoite on palvella talotekniikan koulutusta ja opintojaksoihin liittyviä harjoituksia.

Laboratoriossa halutaan mallintaa ja mitata tilan olosuhteita sekä mitata ja säätää LVIS-järjestelmiä. Opintokokonaisuuksista tilaa tulevat hyödyntämään Energiatehokkuus ja elinkaaritalous sekä LVI-talotekniikka toimitiloissa.

2.2.1 Painesuhdemittaukset

Mittauksissa tarkastellaan laboratoriotilan paine-eroja ympäröiviin tiloihin nähden. Paine-eroja on pystyttävä loggaamaan eli tallentamaan jatkuvasti ja niitä on voitava seurata reaaliaikaisesti. Tilojen järjestelmien toimintaa on pystyttävä muuttamaan ja muutoksien vaikutusta seurataan paine-eroihin. Muutoksia ovat esimerkiksi ilmanvaihdon ilmavirtojen muutos ja ovien tiivistysten muutokset. Lisäksi halutaan havainnollistaa tilan ilmavirtareittejä ja selvittää tilan ilmatase. (TATE-labran kehitystyöryhmä 2017.)

Tilaan tarvitaan paine-eromittaukset kaikkiin ympäröiviin tiloihin nähden. Nykyisen ilmanvaihdon kanaviin tarvitaan käsipellit, jotta yleisilmanvaihto voidaan sulkea mittausharjoitusten ajaksi. Tilaan tarvitaan erillinen IV-kone ja sen tulo- ja poistokanaviin tarvitaan loggaavat mittaukset. (TATE-labran kehitystyöryhmä 2017.)

2.2.2 Lämpötilamittaukset

Tilasta halutaan tietää kuivalämpötila, märkälämpötila, operatiivinen lämpötila ja väliseinien pintalämpötilat. Tilaa ympäröivistä huoneista tarvitaan lämpötilamittaukset. (TATE-labran kehitystyöryhmä 2017.)

Tilasta halutaan myös olosuhdeselvitys, jossa selvitetään olosuhteiden muuttuminen järjestelmää muutettaessa, esimerkiksi ilmanvaihdon ja lämpökuorman muuttuessa. Lisäksi tarkastellaan lämmön siirtymistä viereisiin huoneisiin. (TATE-labran kehitystyöryhmä 2017.)

Mittauksia varten tarvitaan loggaava lämpötilamittausjärjestelmä, jolla on verkkopohjainen käyttöliittymä. Mittauksia on voitava tehdä eri korkeuksilla ja mittalaitteita on pystyttävä siirtelemään. (TATE-labran kehitystyöryhmä 2017.)

2.2.3 Ilmanlaatumittaukset

Ilmanlaadusta halutaan tietää tilan hiilidioksidi (CO₂) ja haihtuvien orgaanisten yhdisteiden (VOC) pitoisuudet. Myös suhteellisen kosteuden (RH) arvo pitää pystyä mittaamaan. Mittauksissa tarkastellaan ilmanvaihdon sekä tarpeenmukaisen ohjauksen vaikutuksia ilmanlaatumittauksiin. (TATE-labran kehitystyöryhmä 2017.)

Mittausten toteutukseen tarvitaan loggaavat mittaukset hiilidioksidista, VOC-päästöistä ja suhteellisesta kosteudesta. Lisäksi halutaan tarkastella tilan eri kohtien mittaustuloksia, joten antureita tarvitaan useita ja niiden paikkoja pitää pystyä siirtelemään helposti. (TATE-labran kehitystyöryhmä 2017.)

2.2.4 Energiamittaukset

Energiamittauksissa halutaan määrittää tilan energiatase ja vertailla sitä IDA-ICE-ohjelmalla tehtävään malliin tilan energian käytöstä. Energiataseeseen kuuluu pattereiden tuottama energia, sähkölaitteiden lämpökuorma, ilmanvaihdon osuus, jäähdytysenergia, lämpöpölväviöt, aurinkokuorma ja ihmisten lämpökuorma. (TATE-labran kehitystyöryhmä 2017.)

Tavoitteiden saavuttamiseksi tilaan tarvitaan jäähdytys- lämmitys- ja ilmanvaihdon energiamittaukset. Lisäksi pitää pystyä mittaamaan valaistuksen ja muiden sähkölaitteiden

sähköenergia. Ulkoilman olosuhteista tarvitaan tieto lämpötilasta ja valoisuustasosta. Ihmisten lämpökuormaa varten tarvitaan tieto mittausten aikaisesta käyttäjämäärästä. (TATE-labran kehitystyöryhmä 2017.)

2.2.5 Valaistuksen ohjaus

Valaistusta tutkittaessa tarkastellaan valaistuksen ohjaustapojen vaikutusta energian kulutukseen. Lisäksi tilassa halutaan mitata valaistustasoa ja seurata sen muutoksia. (TATE-labran kehitystyöryhmä 2017.)

Tavoitteiden toteutumista varten tilan valaistus on uusittava DALI-valaistusjärjestelmäksi ja siihen on liitettävä loggaava valaistusmittausjärjestelmä. (TATE-labran kehitystyöryhmä 2017.)

2.3 Opinnäytetyön tavoitteet ja tarkoitus

Opinnäytetyön tarkoituksena oli luoda toimiva demoympäristö, jota hyödyntämällä voidaan toteuttaa talotekniikan oppimisympäristön jatkokehitys. Työssä ei ollut tavoitteena luoda valmista ratkaisua, jossa on toteutettu kaikki edellä esitetyt oppimisympäristön tavoitteet.

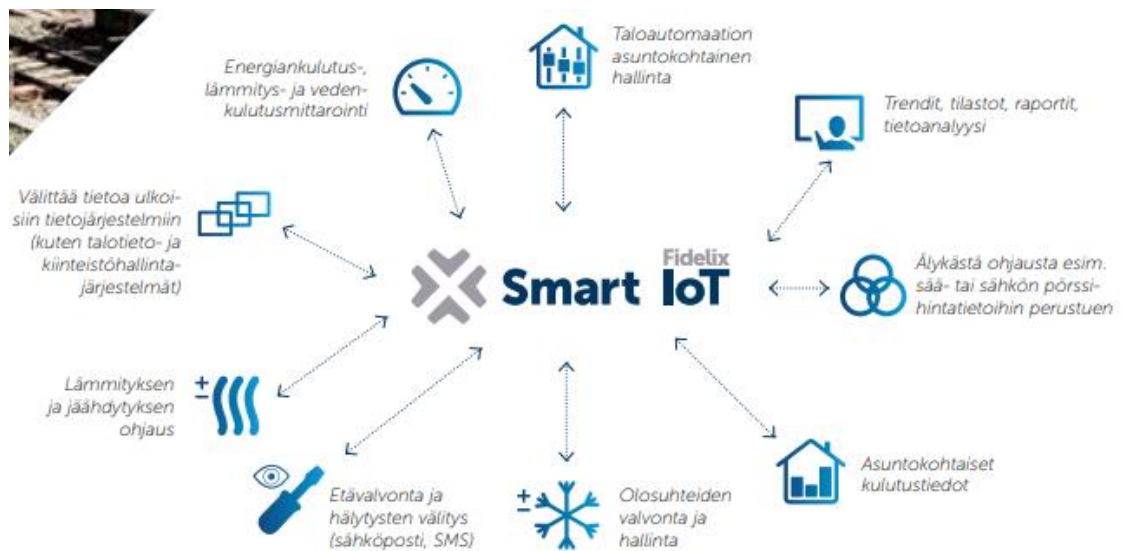
Työ rajattiin IoT-Ticketin talotekniikan valvomo-alustan demon luomiseen, keskeisimpien laitteiden liittämiseen IoT-Ticketiin ja tehtyjen asioiden dokumentointiin. Näiden summana saatiin kerättyä tiedot IoT-Ticketin konfiguroinnista ja käytöstä.

3 PILVIPALVELUIDEN KÄYTTÖ TALOTEKNIKASSA

3.1 Talotekniikan IoT

IoT-ratkaisuissa talotekniikan koneet ja laitteet liitetään internet-yhteydellä verkkoon, jolloin ne ovat ohjattavissa internetin välityksellä. IoT mahdollistaa laajan laitteistokannan reaaliaikaisen seurannan yhdestä valvomosta käsin. Keskitettyyn valvomoon voi olla liitettynä useita kiinteistöjä, jolloin kiinteistömassan hallinta tehostuu ja pystytään vertailemaan kiinteistöjä keskenään. Automaatiojärjestelmän toimittaja voi tarjota valvomoratkaisuaan palveluna, jolloin tietoturva ja ylläpito ovat palveluntarjoajan vastuulla. (Fidelix Oy, IoT-esite.)

Keskeisenä käsitteenä IoT:ssa on, että automaatiolaitteet ovat liitettynä jonkin automaatiojärjestelmätoimittajan pilvialustaan. Esimerkkejä näistä ratkaisuista ovat Fidelix Oy:n Smart IoT ja Schneider Electricin EcoStruxure. Kuvassa 2 on esitettyä Fidelix Oy:n ratkaisuesimerkki talotekniikan IoT:sta ja miten sillä voidaan ohjata ja seurata kiinteistön toimintaa.



KUVA 2. Esimerkki talotekniikan IoT-ratkaisusta (Fidelix Oy, IoT-esite)

3.2 IoT:n hyödyntäminen talotekniikassa

Pilviratkaisu mahdollistaa käyttöliittymien räätälöinnin eri käyttäjille. Esimerkiksi asukkaalle, isännöitsijälle ja huoltoyhtiölle voidaan kullekin räätälöidä käyttöliittymä, jossa on huomioitu kunkin käyttäjän tarpeet. (Fidelix Oy, Smart IoT.) Pilveen liitetyn rakennusautomaatiojärjestelmän käyttö on mahdollista periaatteessa kaikilla internetyhteyden omaavilla päätelaitteilla, kuten tietokoneilla, tableteilla ja älypuhelimilla.

Pilvipalveluiden suurin hyöty saavutetaan kiinteistöistä kerätyn datan hyödyntämisellä ja analysoinnilla. Se mahdollistaa esimerkiksi reaaliaikaisen olosuhdeseurannan, ja pitkän aikavälin kulutustietojen keräämisen. Tietojen avulla voidaan muodostaa raportteja kiinteistön taloteknisten prosessien tehokkuudesta. Raporttien avulla voidaan tehdä esimerkiksi päätöksiä kiinteistön järjestelmien kunnosta tai saneeraustarpeista.

Keinoälyn avulla voidaan luoda kerätystä datasta matemaattisia malleja, jotka yhdistettynä reaaliaikaisiin sääennustemalleihin, mahdollistavat ennakoivan prosessien säädön esimerkiksi lämmitykseen, jäähdytykseen ja ilmanvaihtoon. Mallit huomioivat niin energiatehokkuuden kuin sisäilmaston olosuhteet. Keinoäly mahdollistaa myös oppivien järjestelmien rakentamisen, jolloin voidaan esimerkiksi huomioida miten rakennus reagoi vallitseviin olosuhteisiin. (Fidelix Oy, Ecosmart.)

4 IoT-TICKET -ALUSTA

4.1 IoT-Ticketin yleiskuvaus

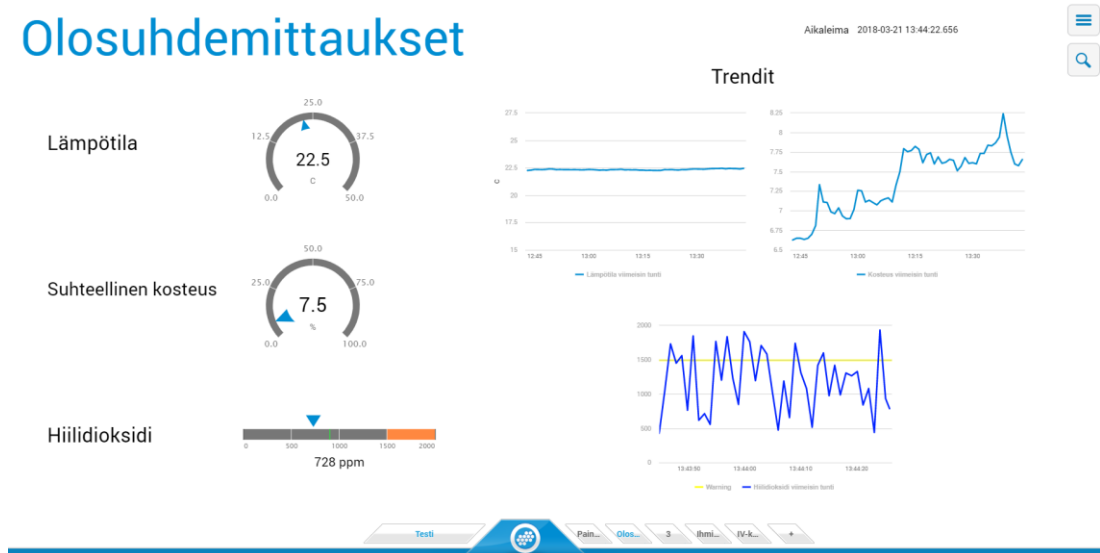
Wapice Oy on kehittänyt tuotteen nimeltä IoT-Ticket, jolla on mahdollista yhdistää pilvipalvelu, palvelin ja elektroniikka integroiduksi ratkaisuksi käyttäjän tarpeiden mukaan. IoT-Ticketiä voidaan käyttää käyttäjän omalta palvelimelta tai Wapicen tarjoamana palveluna. Sen avulla voidaan tehdä etäseurantaa ja hallintaa selainpohjaisen käyttöliittymän avulla. IoT-Ticket mahdollistaa lisäksi kerätyn tiedon analysoinnin ja raportoinnin. (IoT-Ticket. Alusta.)

IoT-Ticketillä on suomenkielinen tuotesivusto osoitteessa <https://iot-ticket.com/fi/>. Lisäksi alustalla on oma IoT-Ticket.com –YouTube-kanava, jossa esitellään esimerkiksi IoT-Ticketin käyttöä ja käyttöliittymien luomisessa käytettäviä toimintoja.

4.2 IoT-Ticketin käyttöliittymä

IoT-Ticketin käyttöliittymää kutsutaan Dashboardiksi. Se on verkkopohjainen käyttöliittymä, joka on liitetty turvatulla yhteydellä etälaitteisiin. Dashboardilla voi esimerkiksi selailla raportteja, tarkastella laitteiden tilatietoja tai tuottaa erilaisia sisältöjä käyttäjän tarpeisiin. (IoT-Ticket. Alusta.) Kuvassa 3 on esitettyä IoT-Ticketillä luotu käyttöliittymä.

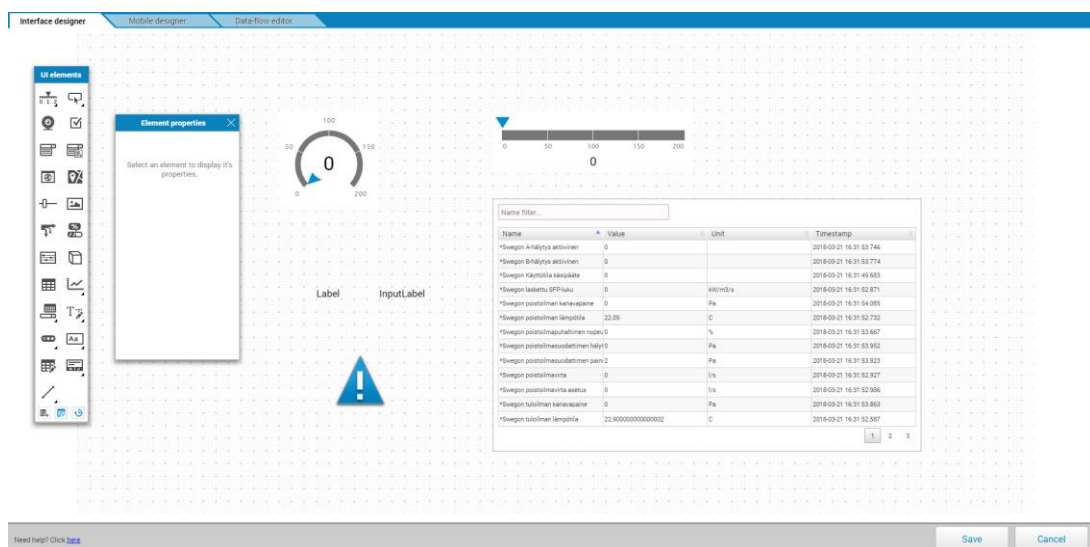
Olosuhdemittaukset



KUVA 3. IoT-Ticketillä luotu käyttöliittymä

4.2.1 Interface Designer -työkalu

Dashboardilla esitettävää tietoa luodaan ja muokataan Interface Designer -työkalulla. Sillä luodaan Dashboardin käyttöliittymä graafisesti. Se hyödyntää ”vedä ja pudota” tekniikkaa ja sillä voidaan luoda muun muassa erilaisia kaavioita, mittareita, taulukoita ja diagrammeja. Myös painikkeiden, pudotusvalikoiden, karttojen, valintaruutujen, tekstin, kuvien ja tunnisteiden luonti on mahdollista. (IoT-Ticket. Alusta.) Kuvassa 4 on esitettynä Interface Designerin näkymä ja erilaisia käytössä olevia käyttöliittymän elementtejä.

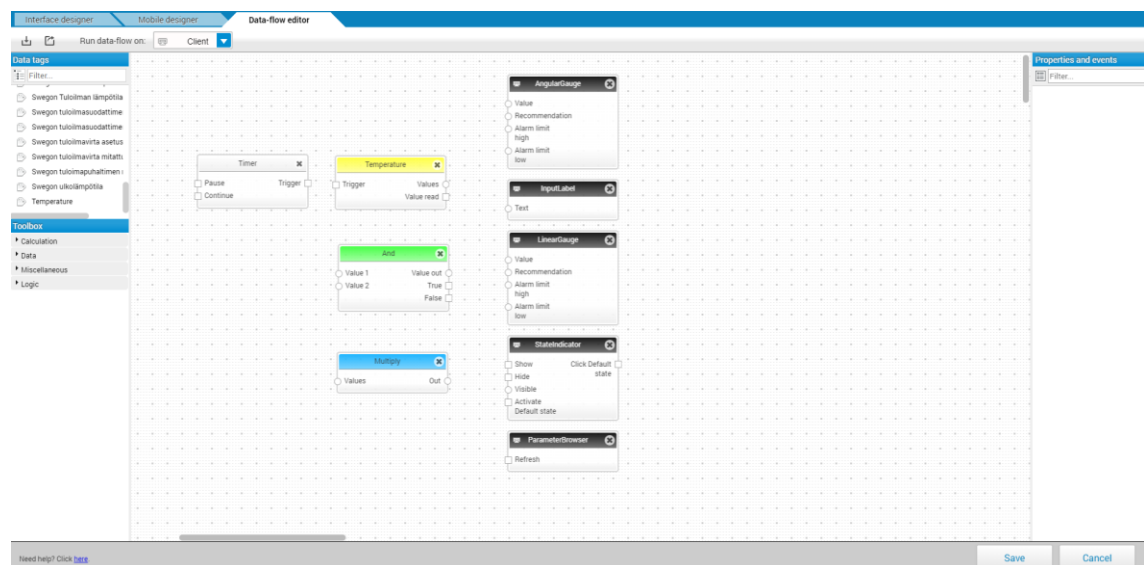


KUVA 4. IoT-Ticketin Interface Designer -näkymä

Kuvassa 4 näkyy vasemmalla valikkoruutu käyttöliittymäelementeille. Elementit liitetään käyttöliittymään vetämällä se hiirellä haluttuun kohtaan ja pudottamalla se paikalleen. Elementit ovat muokattavissa käyttäjän tarpeiden mukaan. Esimerkiksi elementin kokoa ja värimaailmaa voidaan muokata.

4.2.2 Dataflow Editor -työkalu

Logiikkatoimintojen toteutus tapahtuu Interface Designeriin liitettyllä Dataflow Editorilla. Interface Designerilla luodut käyttöliittymäelementit näkyvät Editorin näytöllä datablokkeina, joihin voidaan liittää tiedonkulku graafisesti ”vedä ja pudota” -tekniikalla. Tiedonkulkuun on mahdollista liittää esimerkiksi erilaisia lohkoja, ajastimia, logiikoita ja matemaattisia lausekkeita sekä IoT-Ticketiin liitettyjä datapisteitä. (IoT-Ticket. Alusta.) Kuvan 4 Interface Designer –näkyvää vastaava Dataflow Editor -näkyvä on nähtävissä kuvassa 5.



KUVA 5. Dataflow Editor -näkyvä

Kuvassa 5 harmaa blokki (Timer) on kello, jotka ajoittavat datan keräystä. Keltainen blokki on solmu eli datapiste, josta tieto kerätään, vihreä on looginen portti (AND-portti), sininen on matemaattinen blokki ja mustat blokit ovat Interface Designerissä liitettyjä käyttöliittymässä näkyviä tiedonesityslokkeja. Tiedonkulku blokkien välillä tehdään esimerkiksi hiirellä vetämällä yhteydet blokkien tulo- ja lähtöpisteiden välillä.

4.3 Tiedonkeruu IoT-Ticketillä

Tiedonkeruussa voidaan käyttää elektroniikka, jolla integroidaan datan tuloportti IoT-Ticketin Big-Data palvelimelle. Toinen tapa on liittää data ohjelmiston avulla IoT-Ticket API-rajapintaan. (IoT-Ticket. Alusta.)

4.3.1 IoT-Ticket API -rajapinnat

Käyttäjällä on mahdollista liittää oma ohjelmistonsa IoT-Ticket API -rajapintaan. IoT-Ticket mahdollistaa ohjelmoinnin, esimerkiksi Java, Python, Linux C++, ja C# -ohjelmointikielillä. (IoT-Ticket. Alusta.)

4.3.2 WRM 247+ -laite

Wapice on kehittänyt oman WRM 247+ -laitteen, joka toimii rajapintana laitteiden ja IoT-Ticket -alustan välillä (kuva 6). Valmistajan mukaan laite on suunniteltu täyttämään teollisuuden olosuhteiden vaatimukset. (IoT-Ticket. Alusta.)



KUVA 6. WRM 247+ -laite (IoT-Ticket. Alusta)

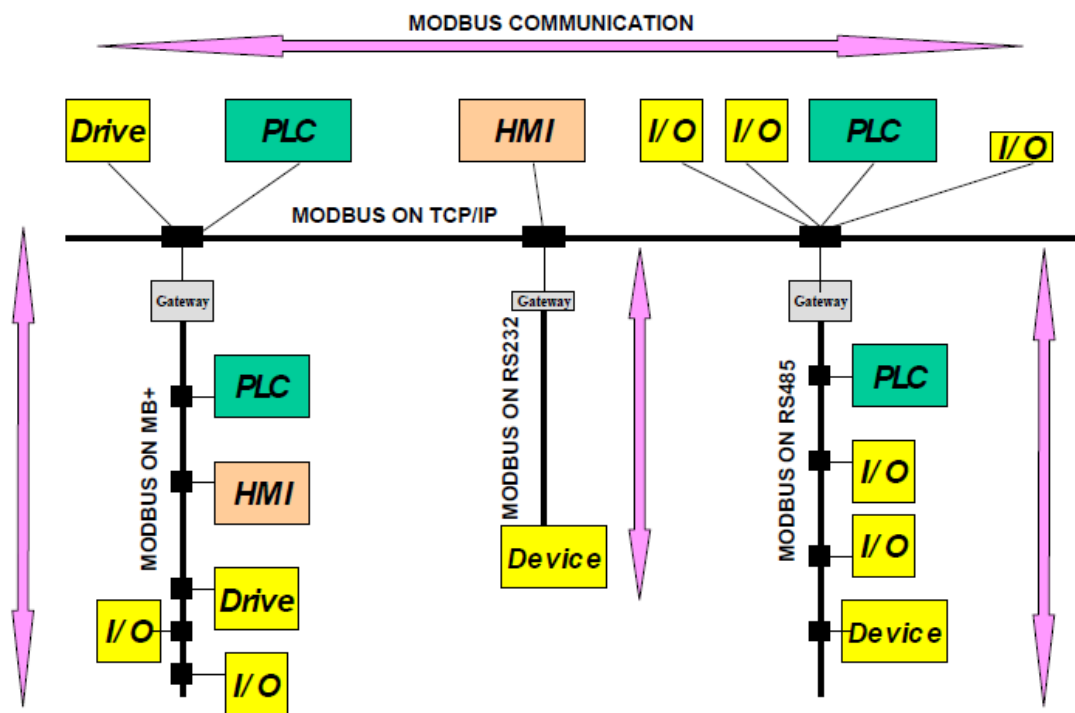
WRM 247+ -laitteella on oma prosessori sekä RAM- ja FLASH-muistia. Langallisesti laitteeseen voidaan liittyä kahdeksan liittimen kautta, joille jokaiselle on määritetty oma tiedonsiirtotapa. Laite tukee CAN ja Modbus-väyliä ja siinä on kaksi Ethernet-porttia. Siihen voidaan liittää digitaalisia I/O pisteitä ja yhdessä liittimessä on analoginen virta/jännite tulo, johon voidaan liittää 0-22 mA tai 0-30 V viestillä kommunikoivia laitteita. (IoT-Ticket. WRM 247+.)

Langattomasti laite tukee GSM-, GPRS-, 3G-, WLAN- ja Bluetooth-standardeja ja palveluita. Laitteessa on lisäksi sisäänrakennettu kiihtyvyyssanturi. (IoT-Ticket. WRM 247+.)

5 MODBUS-PROTOKOLLA

5.1 Modbus-väyläprotokollan yleiskuvaus

Modbus on sarjaliikenneprotokolla, joka on nykyään hyvin käytetty menetelmä laitteiden ja tietoverkkojen yhdistämiseen esimerkiksi teollisuudessa ja rakennuksissa. Yksinkertaistettuna sillä välitetään informaatiota sähköisten laitteiden välillä sarjaväylän yli. Modbus-protokolla on määritetty sarjaportteille (Modbus RTU ja Modbus ASCII) sekä Ethernetille (Modbus TCP). (Schneider Electric.) Kuvassa 7 on esitettyä eri tapoja liittää laitteita Modbus-protokollalla.



KUVA 7. Esimerkit Modbus-väyläratkaisuista (Modbus Organization. V1.1b3, 3)

5.2 Modbus-väylän rakenne

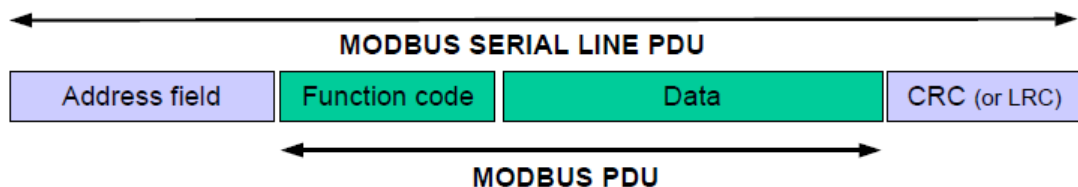
Modbus-väylässä orjalaitteet liitetään yhteen isäntälaitteeseen. Yksinkertaisimmillaan Modbus-viestintä tapahtuu yhden isäntälaitteen ja yhden orjalaitteen välillä eli RS-232-väylällä. Useampia laitteita liitettäessä käytetään RS-485-väylää. (Modbus Organization. V1.1b3, 2, 3.)

Data lähetetään ykkösinä ja nollina eli bitteinä. Nollat lähetetään positiivisina jännitteen arvoina ja ykköset negatiivisina jännitteinä. Viestintä on hyvin nopeaa. Yleisin käytetty viestintänopeus on 9600 baud eli 9600 bittiä sekunnissa. (Schneider Electric)

5.3 Sarjaliikenne Modbus-väylässä

Modbusista on olemassa kaksi erilaista muunnosta sarjamuotoiselle liikenteelle: Modbus RTU (Remote Terminal Unit) ja Modbus ASCII (American Standard Code for Information Interchange). (Modbus Organization. V1.02, 12) Molemmat noudattavat samaa protokollaa, mutta niiden viestirakenteessa on eroja. Esimerkiksi RTU välittää viestit binäärisessä muodossa ja ASCII-viestit lähetetään ASCII-merkistönä. Binääriset viestit ovat lyhyempiä, kuin ASCII-viestit ja siten ainakin teoriassa RTU on nopeampi viestintätapa. (Schneider Electric)

Väylässä lähetettäviä viestejä eli sanomia kutsutaan kehyksiksi (PDU, Protocol data unit). Modbusin kehysrakenne on esitetty kuvassa 8.



KUVA 8. Modbus-viestin kehyksen rakenne (Modbus Organization. V1.02, 8)

Kuvassa 8 ylempi kaksisuuntainen nuoli kuvaa sarjaliikenteen kehysrakennetta ja alempi kaksisuuntainen nuoli Ethernet-verkossa lähetettäviä viestejä (Modbus TCP). Esimerkiksi RTU:n kehys muodostuu orjalaitteen ID-numerosta (Address field), ohjauksesta eli toteutettavasta toiminnosta (Function code), datasta eli pyyntö- ja vastaus-parametreistä (Data) sekä virheentarkistuksesta (CRC). LRC-virheentarkistusta käytetään ASCII-viestinnässä. (Modbus Organization. V1.02, 8)

Yhteen sarjaväylään liitettujen laitteiden tulee tukea samaa sarjaliikennemuotoa. Samassa väylässä ei voi olla RTU- ja ASCII-laitteita. Kaikkien Modbus-laitteiden tulee vähintään toteuttaa RTU-muotoa. ASCII-muoto on optionaalinen tiedonvälitystapa. (Modbus Organization. V1.02, 12.)

5.4 Isäntä/orja-protokolla

Sarjaliikenteessä Modbus käyttää isäntä/orja-protokollaa. Kommunikointi väylässä tapahtuu aina isäntälaitteen aloitteesta. Isäntälaitte lähettää väylään viestin, joka välittyy oikealle orjalaitteelle viestissä olevan osoitteen eli ID:n perusteella. Jokaisella väylään liitettyllä orjalaitteella tulee olla yksilöllinen osoite. (Modbus Organization. V1.02, 7)

Isäntälaitte voi lähettää viestejä suoraan yhdelle orjalle (unicast) tai kaikille orjille (broadcast). Unicast-keskusteluun kuuluu aina kaksi osaa: pyyntö isäntälaitteelta ja vastaus orjalaitteelta. Broadcast-viestit ovat write-käskyjä ja orjalaitteet eivät vastaa siihen. (Modbus Organization. V1.02, 7)

5.5 Modbus-rekisterien käyttö

Orjalaitteen informaatio on tallennettu laitteen muistiin erilaisiin taulukoihin. Taulukot voivat olla tyyppiä read/write (taulukon arvoa luetaan ja sen päälle voidaan kirjoittaa) tai read only (taulukon arvoa voidaan vain lukea). Osa Modbusin rekistereistä on esitetty taulukossa 1. Laitteen valmistajat määrittävät mitä rekisterityyppejä kustakin Modbus-laitteessa on käytössä. Modbus-laitteen käyttöön tarvitaan rekisterien tiedot laitevalmistajalta.

Taulukko 1. Modbus-tilukset (Schneider Electric; Modbus Organization. V1.1b3, 5)

Funktiokoodi	Taulukko	Datan tyyppi	Toiminta
02	Discrete Input	1 bit	Read
01	Coils	1 bit	Read
05	Coils	1 bit	Write
04	Input Registers	16 bit	Read
03	Holding Registers	16 bit	Read
06	Holding Registers	16 bit	Write

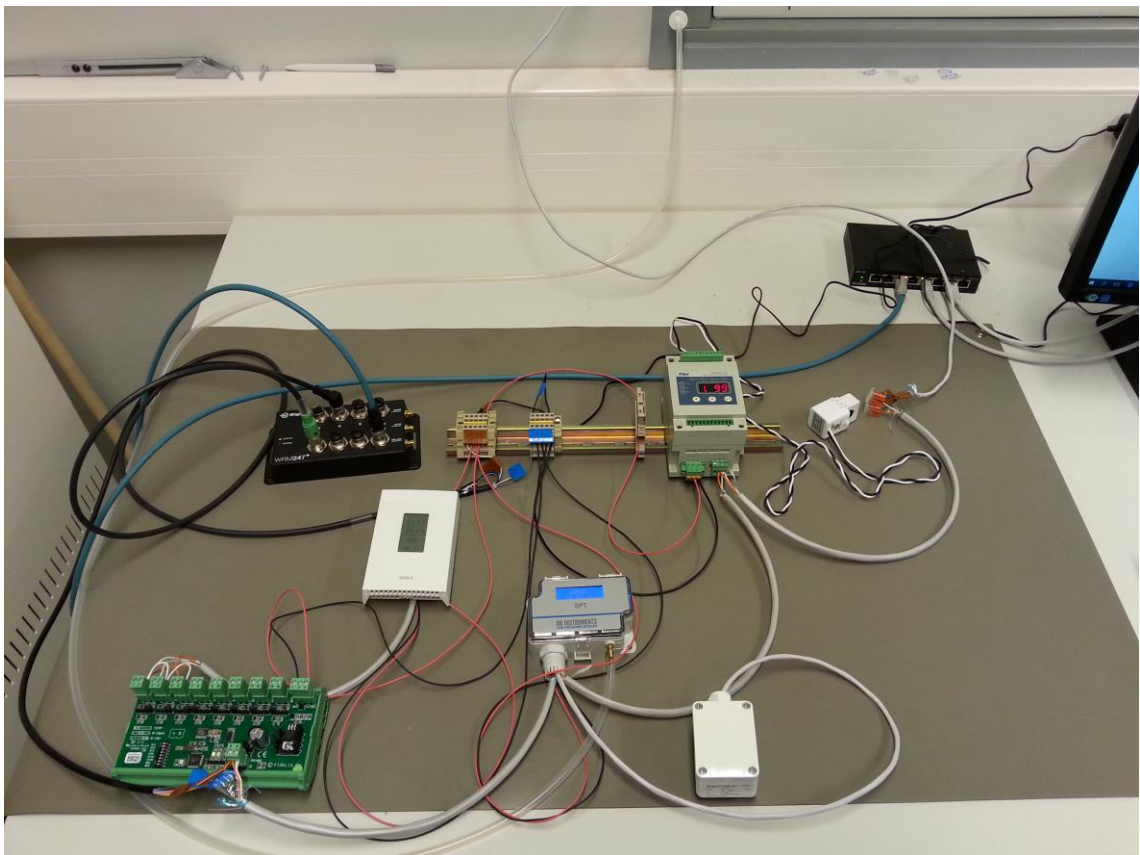
1 bittiset rekisterit voivat olla arvoltaan joko 0 tai 1 (pois/päällä). 16 bittisiin rekistereihin säilötään numeerisia arvoja. Kullakin taulukolla voi olla 9999 rekisteriä ja taulukot erotetaan toisistaan funktiokoodilla. (Schneider Electric.)

Rekisteristä luettu tulos voi tarvita muunnoskerroimen, jotta tulos saadaan näyttämään todellista arvoa. Muunnoskerroin voi olla esimerkiksi muodossa 0.1, 10, x100, tai /10. Esimerkiksi rekisteristä luetaan tulos 239, jonka muunnoskerroin on 0.1 (tai /10). Tällöin rekisterin todellinen mittaustulos on $239 \cdot 0,1 = 23,9$ (tai $239/10 = 23,9$).

6 DEMOYMPÄRISTÖN FYYSINEN TOTEUTUS

6.1 Demoympäristön järjestelmäkuvaus

Demoympäristön fyysinen toteutus käsitti tutkimustilan pöydälle rakennetun demojärjestelmän, johon liitettiin saatavilla olleet laitteet. Laitteille kytkettiin jännite ja ne liitettiin Modbus RTU -väylällä WRM 247+ -laitteeseen. Valokuva demoympäristöstä on nähtävissä kuvassa 9.



KUVA 9. Oppimisympäristön demoasennus

6.1.1 Demoympäristön järjestelmäkaavio

Työ aloitettiin piirtämällä tilasta järjestelmäkaavio annettujen reunaehtojen perusteella. Kaaviota varten etsittiin ja valittiin esimerkkityypit mittalaitteista, joiden tiedonsiirtotapa soveltuu liitettäväksi WRM 247+ -laitteeseen. Päätiedonsiirtotavaksi valittiin Modbus RTU. Järjestelmäkaavio on esitettyä liitteessä 1. Järjestelmäkaavio on viitteellinen, eikä siinä ole otettu huomiin esimerkiksi Modbus RTU -topologiaa.

6.1.2 Demoympäristön laitteet

Osa käytetyistä laitteista oli jo ennestään asennettuna tai koululle hankittuna. Ennalta hankittuja laitteita olivat ilmankäsittelykone, sähköenergian mittari, paine-erolähetin sekä hiilidioksidi-, kosteus- ja lämpötilalähetin. Näiden lisäksi hankittiin WRM 247+ -laite, Pt1000-anturi ja mittausmoduuli. Demoympäristö rakennettiin taulukon 2 laitteilla.

Taulukko 2. Demoympäristön laitteet

Laite	Valmistaja	Käyttötarkoitus	Tiedonsiirtotapa
WRM 247+	Wapice	Rajapinta/Master-laite	Ethernet - Modbus RTU
DPT-IO-MOD	HK Instruments	Paine-erolähetin tulosignaali	Modbus RTU
PMAC201-HW	Zhuhai Pilot Tech.	Monikanavainen sähköenergian mittari	Modbus RTU
GMW83DRP	Vaisala	Hiilidioksidi-, kosteus- ja lämpötilalähetin	0-10V -viesti
GOLD E	Swegon	Ilmankäsittelykone	Modbus RTU
AI-8	Fidelix	Mittausmoduuli Vaisalan lähettimelle	Modbus RTU
Pt1000-anturi	Thermokon	Vastuslämpötila-anturi liitettäväksi paine-erolähettimeen	Resistanssiarvo

Paine-erolähettimeen liitettiin PT1000-anturi, jolloin samalla laitteella pystyttiin mittaamaan paine-eroa ja pintalämpötilaa. Vaisalan lähetin liitettiin Modbus-mittausmoduuliin (AI-8), jolla voitiin muuntaa hiilidioksidin, suhteellisen kosteuden ja lämpötilan mittauksen analogiset jännitesignaalit Modbus RTU -viesteiksi.

Monikanavaisesta sähköenergian mittarista ei saatu luettua tietoja Modbusilla eli sitä ei saatu liitettyä osaksi demoympäristöä. Se on kuitenkin esitetty laiteluettelossa ja järjestelmäkaavioissa, koska sähköenergian mittaukset on suunniteltu tehtäväksi sen avulla.

6.1.3 Demoympäristön sähkökytkennät

Demoympäristössä käytetyille laitteille sopiva käyttöjännite oli 24VDC. Jännitelähteenä toimi kahden ampeerin pistorasiamuuntaja, jonka johto katkaistiin ja sen plus ja miinus johdot kytkettiin DIN-kiskoon asennettuihin riviliittimiin. Riviliittimiltä kytkettiin plus- ja miinus-johdot laitteille, jotka tarvitsivat käyttöjännitteen. Kytkennöissä käytettiin 0,5 mm² kaapeleita. Demoympäristön sähkökytkennän johdotuskaavio on esitetty liitteessä 2.

6.1.4 Demoympäristön Modbus-väylä

Demoympäristön Modbus-väylä toteutettiin liitteen 3 kaavion mukaisesti. Kaapeloinnissa käytettiin NOMAK 2x2x0,5+0,5 instrumentointikaapelia. Demoympäristön Modbus-väylässä poikettiin vaatimuksista siten, että väylästä tehtiin kelluva (maadoitusta ei tehty) ja päätevastuksia ei käytetty. Poikkeukset voitiin tehdä, koska johdinpituudet olivat lyhyitä ja käyttöympäristö ei ollut häiriöitä aiheuttava. Lopulliseen oppimisympäristön väylään tulee tehdä maadoitus ja terminointi.

6.2 WRM 247+ -laitteen liittäminen ja konfigurointi

WRM 247+ -laite liitettiin TAMKin IoT-Ticket-palvelimeen internet-yhteydellä. Liittäminen tapahtui WRM 247+ -laitteen paikallisessa käyttöliittymässä. Siellä asetettiin WRM-serverin osoitteeksi TAMKin palvelimen internetosoite. Tämän jälkeen laite liitettiin IoT-Ticketin hallintaohjelmalla oppimisympäristölle luotuun resurssiin.

Liittämisen jälkeen hallintaohjelmassa valittiin mitä moduulin tiedonsiirtotapoja haluttiin ottaa käyttöön. Laitteesta otettiin käyttöön Modbus RTU –väylä. Sen käyttöönotossa määritettiin tiedonsiirron määreet. Näitä olivat Modbusissa siirtonopeus, data bittien määrä, sanoman viive, pariteetti (kyllä/ei), sarjaväylä (RS-485/RS-232) ja lopetus bittien määrä. Määreet saatiin selville orjalaitteiden teknisistä tiedoista. Taulukossa 3 on esitettynä demoympäristön Modbus-väylän tiedot.

Taulukko 3. Demoympäristön Modbus-väylän määreet

Määre	Tiedot
Protokolla	Modbus RTU
Sarjaväylä	RS-485
Siirtonopeus (Baud)	9600
Databitit	8
Sanoman viive	3,5 merkkiä
Pariteetti	Ei pariteettia
Lopetusbittien määrä	1

6.3 Solmujen lisääminen IoT-Ticketiin

IoT-Ticketissä solmuja kutsutaan Data Nodeiksi. Niillä määritetään mitä tietoa liitetään IoT-Ticketiin. Esimerkkejä solmuista ovat Modbusin rekisterit sekä digitaaliset tai analogiset I/O-pisteet. Modbus-solmujen lisäämisessä valitaan ensin rekisterin tyyppi, esimerkiksi Holding Register, Input Register tai Coils. Tyypin valinnan jälkeen nimetään solmu, määritetään mitattavan suureen yksikkö, datatyyppi ja nimetään solmun tagi. Tagin avulla solmu tunnistetaan Dataflow Editorissa ja Interface Designerissa. Lisäksi määritetään ainakin I/O-asetuksiin rekisterin numero, tavujärjestys (endianness) ja orjalaitteen osoite. Tallennuksen jälkeen solmu ja sen mittausdata tulee näkyviin WRM 247+ -laitteen Data node -listaan.

7 DEMOYMPÄRISTÖN KÄYTTÖLIITTYMIEN RAKENTAMINEN

7.1 Paine-eron ja pintalämpötilan käyttöliittymä

Paine-erolle luotiin käyttöliittymä, jossa paine-erosäätimen mittaama paine-ero viereiseen tilaan esitettiin tekstinä. Paine-eromittariin liitetyn Pt1000-anturin lämpötilan mitauslukema esitettiin osoitinikonilla. Kuvassa 10 on esitettynä paine-eron ja lämpötilan käyttöliittymä.

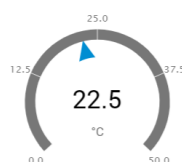
Paine-ero ja pintalämpötila

Aikaleima 2018-04-04 18:29:55.255

Paine-ero

-2 Pa

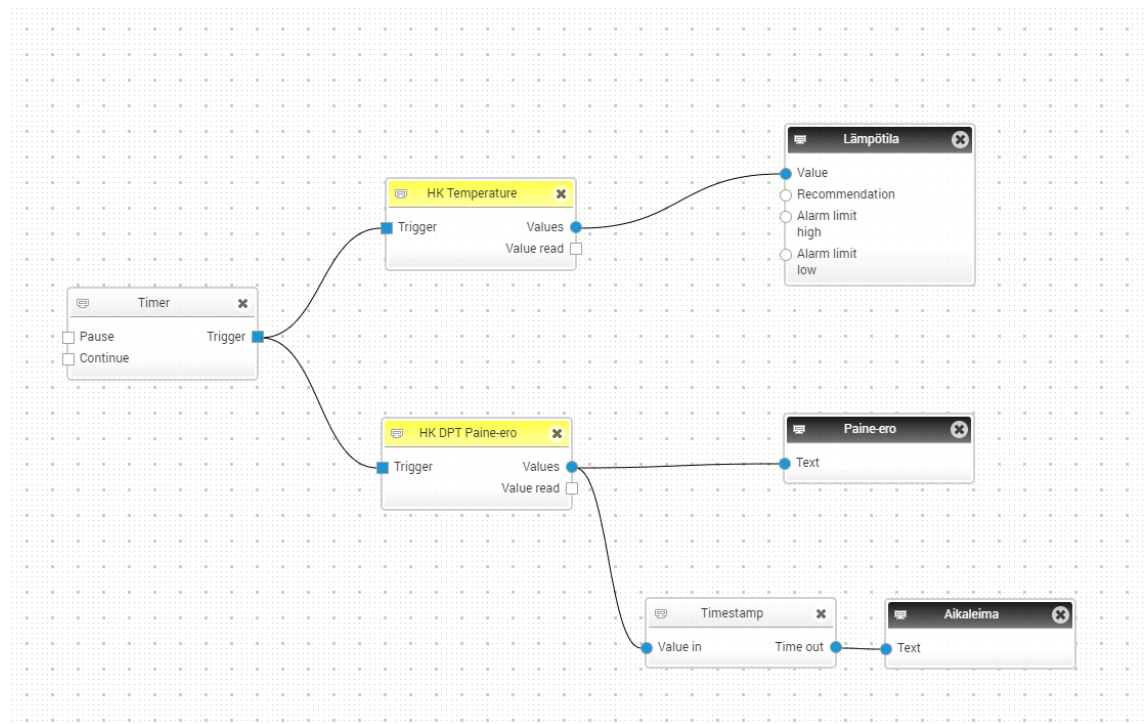
Pintalämpötila



KUVA 10. Paine-eron ja pintalämpötilan käyttöliittymä

7.1.1 Paine-eron ja pintalämpötilan mittarit

Interface Designerissa lisättiin paine-erolle Input label -blokki ja pintalämpötilalle osoitinkuvake. Dataflow Editorissa liitettiin paine-eromittarilta liitetyt solmut suunnittelualueelle. Mittausten ajastus tehtiin liittämällä Timer-blokki ja liittämällä sen Trigger-piste solmujen Trigger-pisteisiin. Kuvassa 11 on nähtävillä paine-eron ja lämpötilan käyttöliittymän Dataflow Editor -näkyvä.



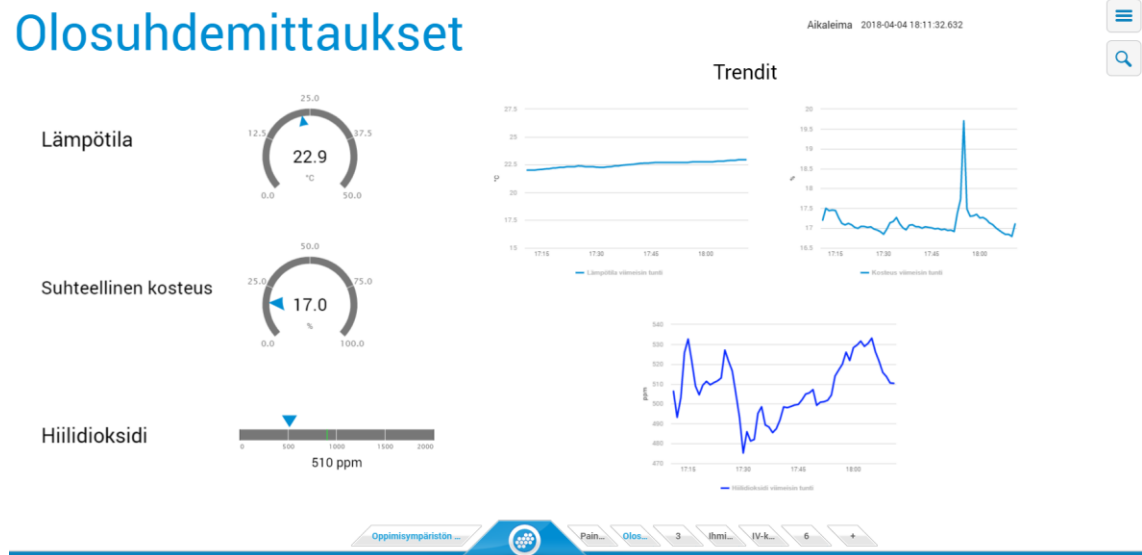
KUVA 11. Paine-eron ja pintalämpötilan Dataflow Editor -näkömä

7.1.2 Aikaleiman esittäminen

Käyttöliittymään liitettiin aikaleima, josta voit seurata, milloin käyttöliittymän mittausarvot on luettu viimeksi. Aikaleima saatiin lisäämällä Interface Designerissa Input label -blokki, johon liitettiin Timestamp-blokki. Timestamp-blokki liitettiin datasolmun values-pisteeseen. Tällöin datasolmun lähettäessä mittaus tuloksen, sen lähetysaika saatiin näkyviin aikaleiman Input labeliin. Samaan tapaan kaikkiin käyttöliittymäikkunoihin luotiin aikaleimatieto. Aikaleiman luonnin rakenne on esitettyä kuvassa 11.

7.2 Olosuhdemittausten käyttöliittymä

Olosuhdemittauksista tehtiin käyttöliittymä (kuva 12), jossa näkyy lämpötila, suhteellinen kosteus ja hiilidioksidi. Lisäksi jokaiselle mittaukselle tehtiin trendiseuranta. Trendiseurannan ajanjaksoksi määritettiin edellinen tunti.

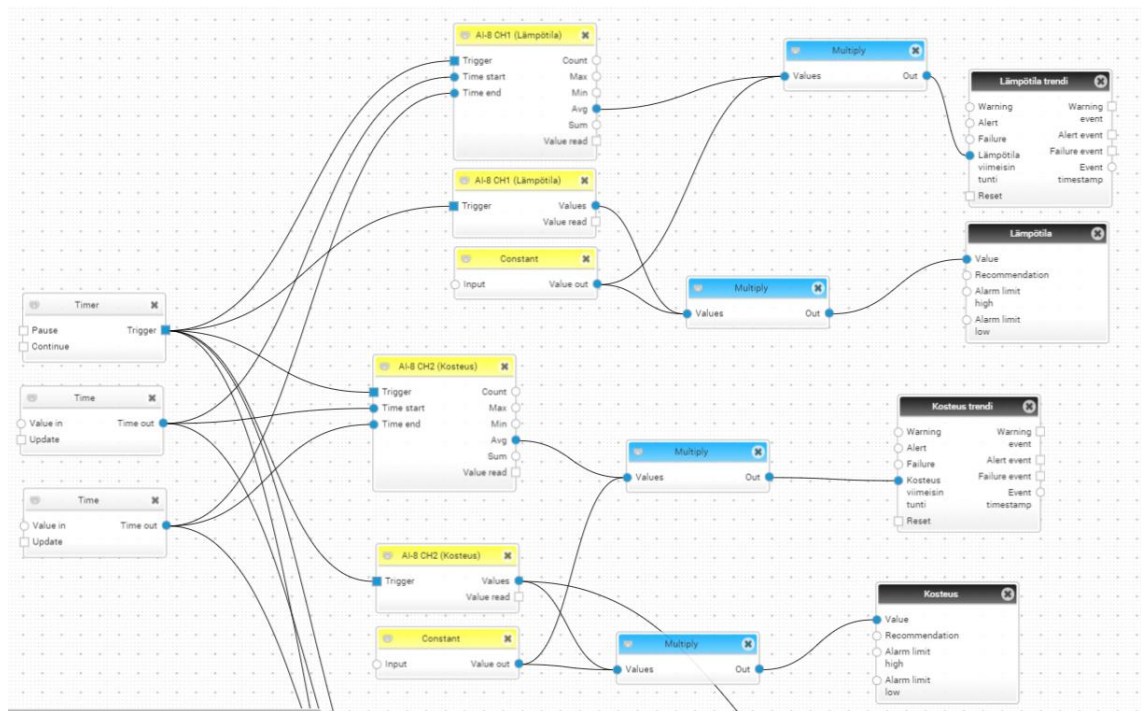


KUVA 12. Olosuhdemittausten käyttöliittymä

7.2.1 Olosuhdemittausten mittarit

Aluksi näyttöikkunaan liitettiin Interface Designerin kirjastosta mittarikuvakkeet. Tämän jälkeen siirryttiin Dataflow Editoriin, johon oli generoitunut luotujen mittarien datapiste-blokit. Editorissa lisättiin mittausten solmut ja liitettiin niihin ajastus lisäämällä Timer-blokki ja liittämällä sen Trigger-piste solmujen Trigger-pisteisiin. Solmujen ajanjaksoksi määritettiin nykyinen aika, jolloin mittari esittää reaaliaikaista tulosta. Tämän jälkeen muunnettiin solmun jännitesignaali esimerkiksi lämpötilaksi liittämällä siihen skaalaus.

Esimerkiksi lämpötila-anturin jännitesignaali oli 0-10 voltia. Mittarin mittaalue oli 0...+50 °C eli mittauksen ulostulojännite 0V vastasi 0°C ja 10 V vastasi +50 °C. Tästä saatiin kulmakertoimen avulla skaalaukseksi 5. Skaalaus tehtiin lisäämällä Multiply-ikonin ja liittämällä sen Values-pisteeseen datapisteen Values-piste ja Constant-ikonin Values-out-piste. Constant-ikonin arvoksi määritettiin luku viisi. Dataflow Editor -näkyminen osasta olosuhdemittauksista on esitettyä kuvassa 13.



KUVA 13. Dataflow Editor -näkyvä olosuhdemittausten ohjelmoinnista

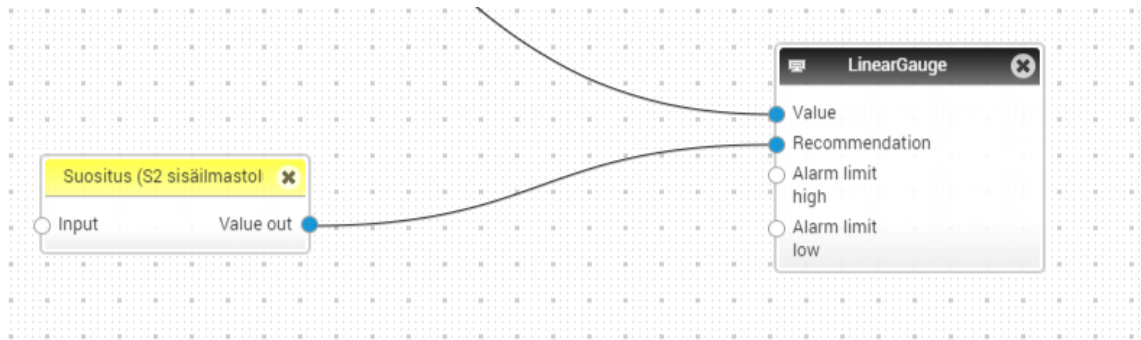
7.2.2 Olosuhdemittausten trendiseurannat

Trendiseurannan luomiseksi Interface Designeriin lisättiin taulukot kullekin mittaukselle. Tämän jälkeen liitettiin Dataflow Editoriin toiset kappaleet mittausten solmuja. Solmujen ajanjaksoksi vaihdettiin Custom, jolloin datablokkiin ilmestyi lisää datapisteitä. Trendiseurannan ajanjakso saatiin lisäämällä kaksi aika-blokkia, jotka liitettiin solmun Time start - ja Time end -datapisteisiin. Time start -blokin offsetiksi asetettiin -1 tunti ja Time end -blokkiin ei asetettu mitään muutoksia. Tällöin kunkin mittauksen taulukossa näkyi edeltävän tunnin mittaustulokset. Trendiseurannan liitokset ja rakenne Dataflow Editorissa ovat nähtävissä kuvassa 13

7.2.3 Rajojen lisääminen hiilidioksidimittaukseen

Hiilidioksidimittauksen mittarissa haluttiin esittää suositustaso hiilidioksidipitoisuudelle. Pitoisuuden raja-arvo valittiin sisäilmayhdistyksen suosituksen mukaisesti sisäilmastoluokkaa S2 (hyvä) vastaavaksi. S2 sisäilmastoluokituksen mukaan hiilidioksidipitoisuuden enimmäisarvo on 900 ppm. (Sisäilmayhdistys ry 2008.)

Suositusraja saatiin tehtyä lisäämällä Dataflow-editoriin Constant-blokki ja asettamalla sen lukuarvoksi 900. Constant-blokin Value out –datapiste liitettiin mittari-ikonin Recommendation-pisteeseen. Kuvasta 14 löytyy suositusrajan liittäminen mittari-ikoniin.



KUVA 14. Suositusrajan liittäminen mittariin

Asetettu raja-arvo näkyi käyttöliittymän mittari-ikonissa vihreänä pystyvivana. Hiilidioksidimittauksen mittari-ikoni on esitettyä kuvassa 15.

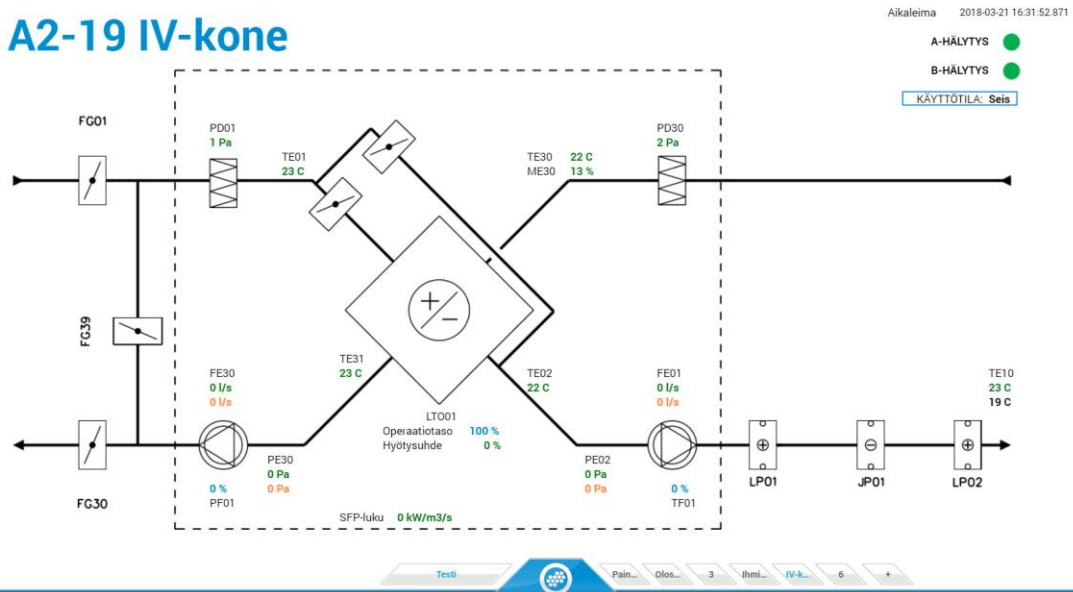


KUVA 15. Hiilidioksidimittauksen mittari-ikoni

7.3 Ilmankäsittelykoneen käyttöliittymä

Ilmankäsittelykoneesta piirrettiin Cads-ohjelmalla prosessikaavio, joka liitettiin Interface Designeriin kuvana. Kuvan päälle lisättiin prosessin mittaustiedot, jolloin niistä muodostui kuvassa 16 esitetty ilmavaihtokoneen käyttöliittymä.

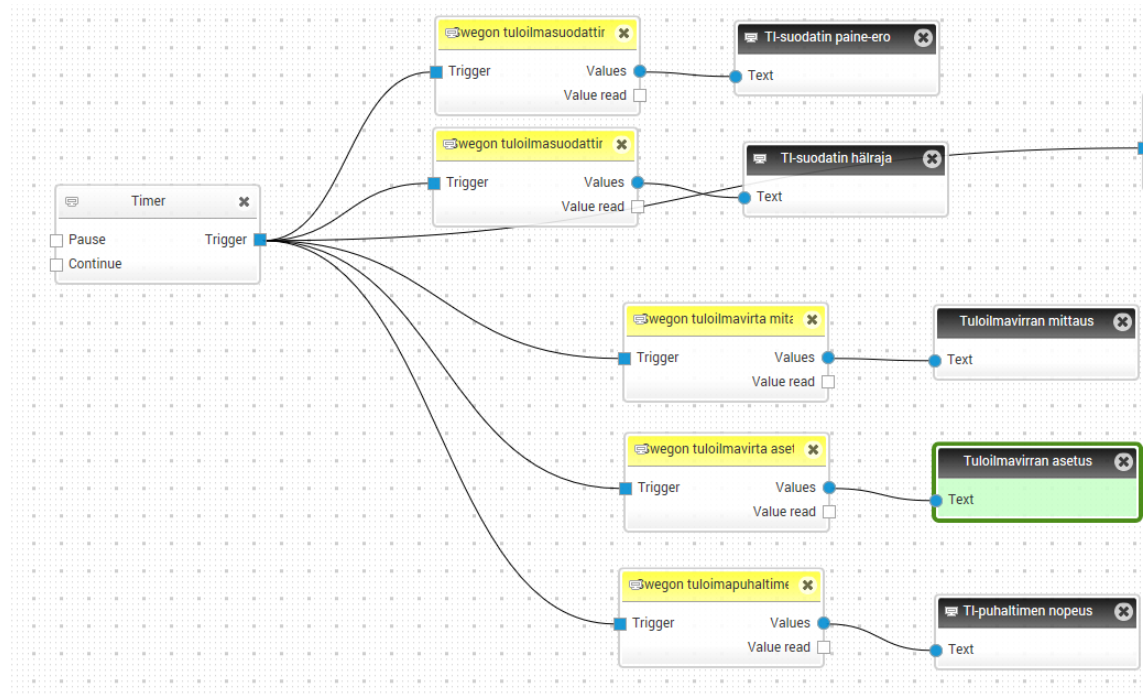
A2-19 IV-kone



KUVA 16. Ilmanvaihtokoneen käyttöliittymä

7.3.1 Ilmankäsittelykoneen tietojen esitys

Interface Designerissa prosessikaavion päälle lisättiin sopiviin kohtiin Input labelita ilmentämään ilmanvaihdon mittausten asetus- ja mittausrvoja. Mitattu arvo esitettiin vihreällä värillä, asetusarvo oranssilla ja säätöviesti sinisellä värillä. Input labelien viereen sijoitettiin tekstiruutuja, joilla nimettiin jokainen mittauspiste. Nimeäminen tehtiin Tampereen kaupungin suunnitteluohjeen mukaisesti (Tampereen kaupunki). Jokaiseen Input labeliin liitettiin sitä vastaava solmu ja solmuun liitettiin ajastus Timer-blokilla. Kuvassa 17 on esitettyä osa ilmanvaihtokoneen solmuista ja niiden linkitys Input labeleihin.



KUVA 17. Ilmankäsittelykoneen Dataflow Editor -näkömää

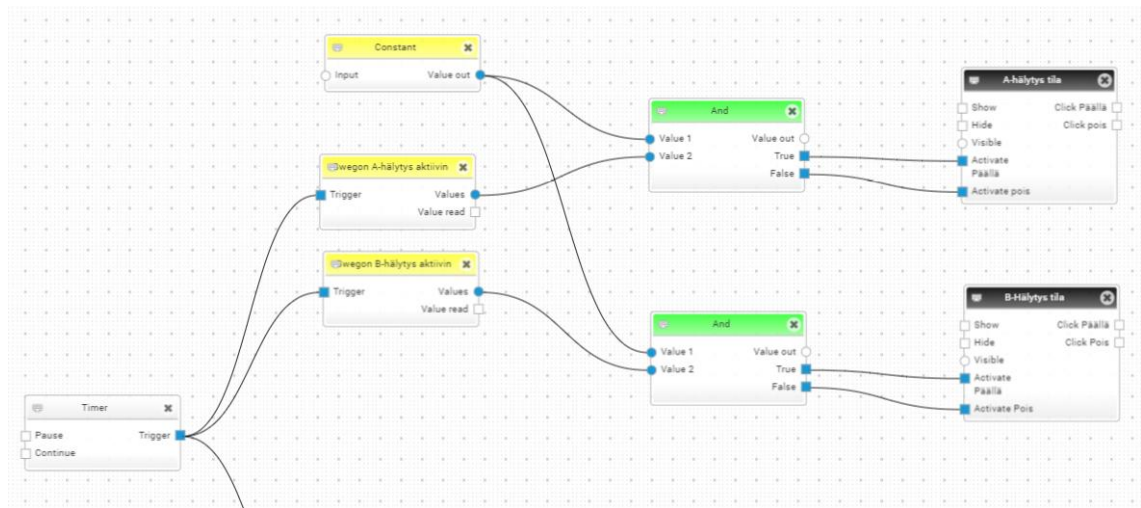
7.3.2 Hälytysten indikoinnit

Hälytystieto tuotiin käyttöliittymään luomalla Interface Designerissa StateIndicator-blokit A-hälytyksille ja B-hälytyksille. Indikoinnit ilmoittivat, että oliko jokin ilmanvaihtokoneen A- tai B-luokan hälytyksistä aktiivisena. Indikaattoriin luotiin kuvaikonit hälytyksen aktiiviselle (punainen pallo) ja passiiviselle tilalle (vihreä pallo). Ikonit ovat esitettyinä kuvassa 18.



KUVA 18 Hälytysten indikoinninikonit

Interface Designerissa liitettiin datapisteet luotuihin indikaattoreihin. Lisäksi tietoketjuun tarvittiin AND-logiikkablokki. Indikointien näkömää Interface Designerissa on esitettyinä kuvassa 19.



KUVA 19. Indikointien Dataflow Editor -näkömä

Hälytysten tilaindikointi tehtiin siten, että AND-blokin tulopisteisiin liitettiin Constant-blokki ja hälytyksen solmu. Constant-blokin vakioarvoksi annettiin luku 1. Hälytyksen solmu oli rekisterityyppiä Discrete Input ja se saa arvot 1 (hälytys päällä) tai 0 (hälytys pois päältä).

AND-blokki toimii siten, että se lähettää tiedon tosi, kun siihen liitetyt tulot saavat arvon yksi. Muissa tapauksissa AND-blokin lähtö on epätosi. AND-blokin tosi ja epätosi -lähdöt liitettiin indikaattorin tuloihin Activate Päällä ja Activate Pois. Hälytyksen aktiivinen ikoni (punainen pallo) syttyy siis aina, kun solmu lähettää arvon 1. Muussa tapauksessa hälytyksen ikoni on vihreä pallo (hälytys pois päältä).

7.3.3 Tilatiedon indikointi

Tilatiedon indikoinnissa esitettiin ilmapääntelykoneen käsipäänteen käyttötila. Tilatiedon solmu oli tyyppiä Read Input Register ja se sai arvot välillä 0-3. Käyttötilan indikointi on esitettyä kuvassa 20. Arvoja vastaavat käyttötilat ovat esitettyä taulukossa 4.

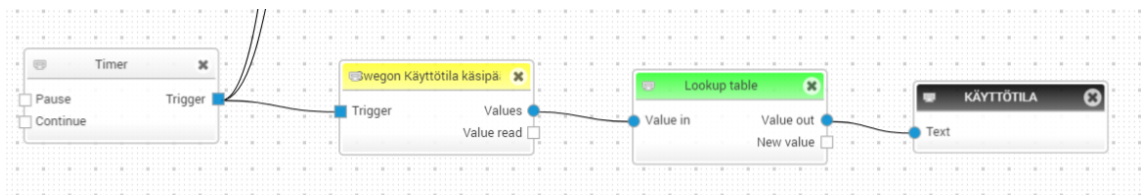
KÄYTTÖTILA: Seis

KUVA 20. Käyttötilan indikointi

Taulukko 4. Käyttötilan indikointi

Rekisterin tulo	Käyttötila
0	Seis
1	Auto
2	Käsin, pieni ilmavirta
3	Käsin, suuri ilmavirta

Käyttötilan indikointi luotiin liittämällä käyttötilan solmuun valintataulukko (Lookup table), johon syötettiin taulukossa 4 esitetyt tiedot tulolle ja lähdölle. Valintataulukon lähtöpiste liitettiin käyttötilan Input labeliin, joka indikoi solmun lähettämän tiedon käyttötilasta tekstinä. Käyttötilan indikoinnin Dataflow Editor -näkyvä on nähtävissä kuvassa 21.



KUVA 21. Käyttötilan indikoinnin Dataflow Editor -näkyvä

8 TYÖN AIKANA SELVITETTYJÄ ONGELMATILANTEITA

8.1 Modbus-väylän ongelmat

8.1.1 Laitteen ID-numero 1 ei toimi

Yhteen Modbus-väylään voidaan liittää laitteita, joilla kullakin tulee olla yksilöllinen osoite väliltä 1-247. Isäntälaitteella ei tarvitse olla yksilöllistä osoitetta. Tehtäessä ohjelmointia huomattiin, että laite, jonka osoitteeksi oli määritetty yksi, ei vastannut rekisterikyselyihin. Kun laitteelle määritettiin toinen osoite, se alkoi toimia oikein. Tästä pääteltiin, että isäntälaitte varasi osoitteen yksi. Se voitiin päätellä siitä, että yhdelläkään toisella väylään liitettyllä laitteella ei ollut osoitteena ykköstä.

8.1.2 Mittaustieto ei näytä oikein

Määrittäessä painelähttimen Modbus-rekisteriä, huomattiin että mittaustieto näytti väärin. Lukema oli 65534, kun sen olisi pitänyt olla laitteen näytön mukaan -2 (Pa). Tämä johtui siitä, että solmun datatyypiksi oli valittu väärin. Paine-ero mittari pystyi mittaamaan negatiivisia ja positiivisia paine-eron arvoja.

Valmistajan antama tieto datatyypistä oli 16 bit. Se ei ollut määrittänyt ohjeessaan onko sen tyyppi etumerkillinen vai etumerkitön (signed/unsigned) eli ovatko mitatut arvot positiivisia tai negatiivisia vai pelkästään positiivisia. Etumerkillisen tiedon datatyypiksi oli IoT-Ticketissä 16INT ja etumerkitön 16UINT. Alkujaan solmun datatyypiksi oli valittu 16UINT. Kun se vaihdettiin etumerkilliseksi eli tyyppiksi 16INT, laitteen mittaustulos saatiin luettua oikein.

8.2 Ethernet-verkon ongelmat

WRM 247+ -laitteen käyttöönotto ei aluksi onnistunut, vaikka laite oli konfiguroitu oikein laitteen paikallisesta käyttöliittymästä. Yhdessä TAMK:in IoT-Ticketin pääkäyttäjän kanssa tutkittiin asiaa ja huomattiin, että TAMK:in palomuurin estä liikenteen laitteen Ethernet-portista, josta se oli liitetty TAMK:in verkkoon. Kun palomuurin asetuksista sallittiin liikenne liitetystä portista, niin laite liittyi TAMK:in serveriin.

8.3 Mittaustieto ei päivittynyt käyttöliittymään

Yhdessä tilanteessa mittaustieto ei päivittynyt demoympäristön käyttöliittymään. IoT-Ticketin Enterprise Manager -työkalulla nähtiin, että WRM 247+ -laite oli online-tilassa, mutta se ei välittänyt tietoa käyttöliittymään. Tästä voitiin päätellä, että vika löytyy TAMK:in IoT-Ticketin servereiltä. Pääkäyttäjän selvittäessä tilannetta, huomattiin, että yksi servereistä oli kaatunut. Järjestelmä saatiin toimimaan, kun kaikki IoT-Ticketin serverit käynnistettiin uudelleen. Tämän jälkeen mittaustiedot päivittyivät käyttöliittymään myös katkoksen ajalta.

9 POHDINTA

Aihe opinnäytetyöhön tuli TAMKilta. Tarjotun työn aiheen alkuperäinen otsikointi, IoT ja rakennusautomaatio, herätti heti tekijän mielenkiinnon. Aihealue oli juuri sitä mistä tekijä halusi tehdä opinnäytetyönsä. Lisäksi motivaationa oli edesauttaa tulevien talotekniikan opiskelijoiden oppimismahdollisuuksia rakennusautomaatiosta.

Työ tehtiin pääosin vuoden 2018 tammikuun ja huhtikuun välisenä aikana. Tekijän alkuperäisenä ajatuksena oli, että työ toteutettaisiin asentamalla laitteita oppimisympäristöön ja suoraan monitoroitaviin prosesseihin. Opinnäytetyön aikataulusta johtuen työ rajattiin lopulta oppimisympäristöstä tehtäväksi demoympäristöksi. Demoympäristöön pyrittiin liittämään saatavilla olleet keskeiset laitteet ja erityisenä tavoitteena oli IoT-Ticket alustan käyttöönotto ja talotekniikan käyttöön soveltuvien käyttöliittymien luonti.

Opinnäytetyölle asetetut tavoitteet saavutettiin kokonaiskuvassa hyvin. Käyttöön saadut laitteet saatiin liitettyä osaksi demoympäristöä lukuun ottamatta sähköenergian mittauslaitetta. Muut laitteet saatiin liitettyksi ja niiden tärkeimmät mittaustiedot saatiin tuotua käyttöliittymiin.

Myös IoT-Ticketin käyttöönotossa onnistuttiin hyvin. IoT-Ticket alustan käyttö oli helppo omaksua suhteellisen nopeasti ja saatu käyttöopastus ja ohjeet olivat riittäviä, joten varsinaista käyttökoulutusta IoT-Ticketin käytöstä ei tarvittu. IoT-Ticketin työkalujen käyttö osoittautui helpoksi ja talotekniikan opetuksen käyttöön saatiin luotua toimivat käyttöliittymämallit. Käyttöliittymiin rakennettiin erilaisia tapoja esittää tietoja, jolloin niiden esimerkkien avulla olisi helppo toteuttaa lopulliset oppimisympäristön käyttöliittymät.

Keskeisenä teemana työssä oli Modbus RTU sarjaliikenneprotokolla ja siihen liittyvät asiat. Työn aikana saatiin konkreettista oppia esimerkiksi väylän rakenteesta ja tiedonsiirrosta Modbus RTU -kommunikoinnilla. Työn tekijällä ei ollut alkujaan osaamista Modbus RTU:sta. Myöskään pienjänniteasennuksista ei ollut kokemusta, muuten kuin opintojen aikana tehtyjen pienten asennusharjoitusten kautta. Opinnäytetyön edetessä osaaminen kummastakin kehittyi huomattavasti. Opinnäytetyön avulla tekijälle muodostui hyvä käsitys Modbus RTU:n perusteista ja Modbus-väylän toteuttamisesta.

Jatkokehitettävää työn jälkeen jäi runsaasti. Oikeastaan tämä työ toimi vasta alkusysäyksenä oppimisympäristön kehitystyölle. Lopullisen oppimisympäristön toteutus ja asennukset tulevat yksinään jo olemaan iso projekti. Lisäksi tässä työssä epäonnistunut energiamittauksen liittäminen vaatii jatkotyötä. Työn aikana ei saatu selville miksi energiamittaria ei saatu liitettyä demoympäristöön. Nähtäväksi jää soveltuuko käytössä ollut laitemalli energiamittaukseen vai joudutaanko käyttämään toista ratkaisua.

Lisäksi oppimisympäristön totuttamiseen tarvitaan myös muiden energiamittausten (lämmitys, jäähdytys, ilmanvaihto ja lämpökuormat) liittäminen ja niiden toteutusratkaisujen kehittäminen sekä esimerkiksi DALI-valaistuksen toteuttaminen tilaan. Näistä näkökulmista oppimisympäristön kehitys mahdollistaa opinnäytetyöaiheita tuleville talotekniikan ja mahdollisesti muidenkin opintoalojen opiskelijoille.

LÄHTEET

Fidelix Oy. Ecosmart. Luettu 20.3.2018. <https://www.fidelix.fi/ecosmart/>

Fidelix Oy. Smart IoT. Luettu 20.3.2018. <https://www.fidelix.fi/smart-iot/>

Fidelix Oy. IoT-esite. Viitattu 27.3.2018. <https://www.fidelix.fi/wp-content/uploads/Fidelix-Smart-IoT-web.pdf>)

IoT-Ticket. Alusta. Wapice Oy. Luettu 1.2.2018. <https://iot-ticket.com/fi/alusta>

IoT-Ticket. WRM 247+. Wapice Oy. Luettu 1.2.2018.
https://www.iot-ticket.com/images/Files/WRM247+_DataSheet_2015.pdf

Modbus Organization. MODBUS Application Protocol Specification V1.1b3. Luettu 5.2.2018. http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Modbus Organization. MODBUS over serial line specification and implementation guide V1.02. Luettu 5.2.2018. http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf

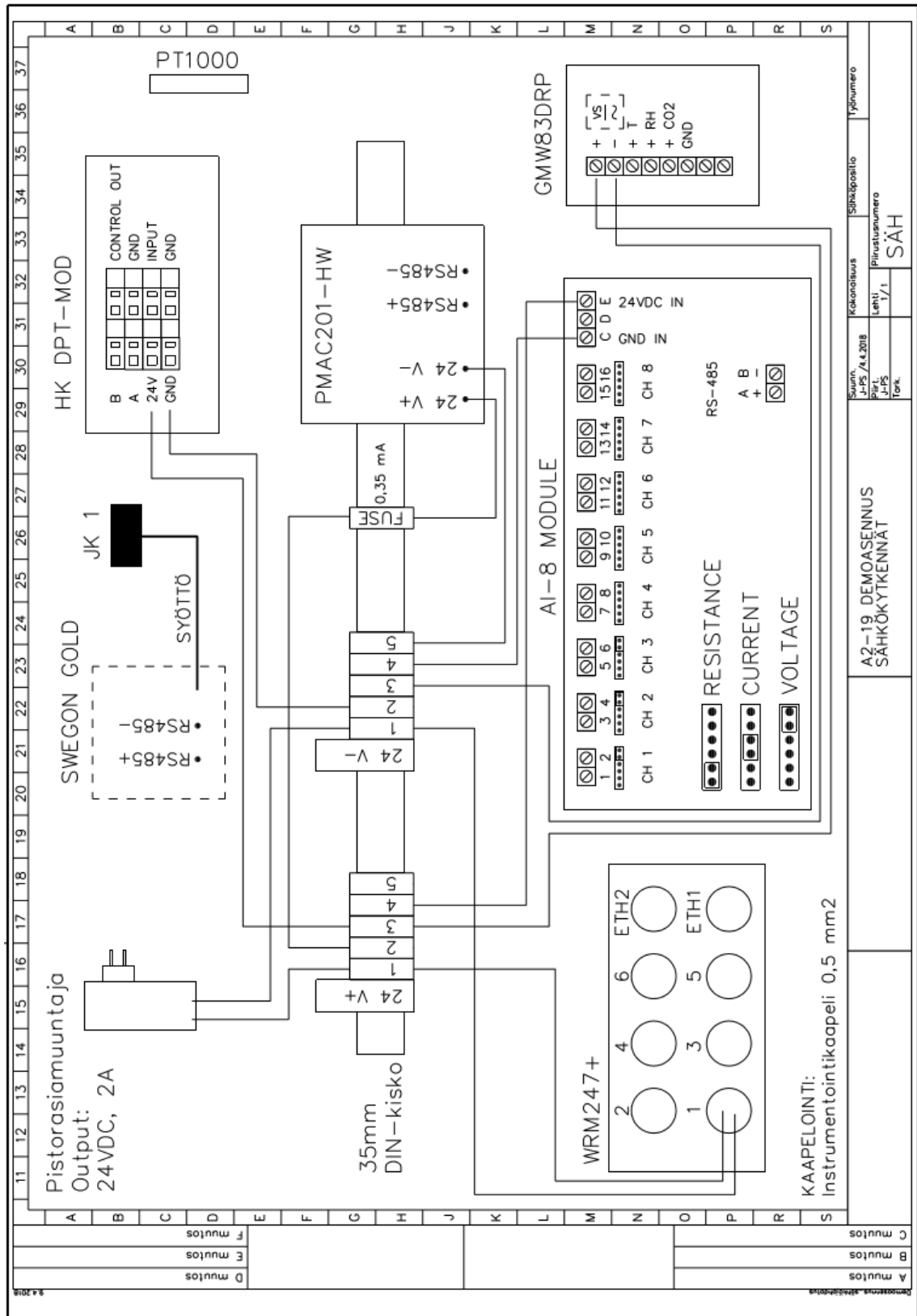
Schneider Electric. What is Modbus and How does it work? Luettu 28.2.2018.
<https://www.schneider-electric.com/en/faqs/FA168406/>

Sisäilmayhdistys ry. 2008. Kemialliset epäpuhtaudet. Luettu 27.3.2018. <http://www.sisailmayhdistys.fi/Terveelliset-tilat/Sisailmasto/Kemialliset-epapuhtaudet>

Tampereen kaupunki. Tampereen tilapalvelut Oy. Laitemerkintäjärjestelmäohje LVIASJ-suunnittelijalle. 6.5.2013. Luettu 27.3.2018. http://www.tampere.fi/tilapalvelut/material/Knw0CCvtq/Rakennusautomaatiojarjestelma_laitemerkintajarjestelma_060513.pdf

TATE-labran kehitystyöryhmä. 2017. Palaverimuistio.

Liite 2. Demoympäristön johdotuskaavio



Liite 3. Demoympäristön väyläkaapelointi

