



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Teemu Kortelainen

# Ilmamääräsäätimen integroiminen rakennusautomaatiojärjestelmään Modbus-sarjaliikenneprotokollalla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

22.9.2019

Tekijä Otsikko Sivumäärä Aika	Teemu Kortelainen Ilmamääräsäätimen integroiminen rakennusautomaatiojärjestelmään Modbus-sarjaliikenneprotokollalla 48 sivua + 2 liitettä 22.9.2019
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine	Automaatiotekniikka
Ohjaajat	Osastopäällikkö Tero Wendelin Lehtori Reijo Leinonen
<p>Insinöörytyössä integroitiin ultraäänitekniikalla mittaava ilmamääräsäädin rakennusautomaatiojärjestelmään Modbus RTU -sarjaliikenneprotokollalla. Työ tehtiin rakennusautomaatiojärjestelmiä toimittavalle Fidelix Oy:lle.</p> <p>Integraatio testattiin oikeilla laitteilla realistisessa testiympäristössä, jotta mahdolliset virheet saatiin eliminoidua jo varhaisessa vaiheessa – ennen kuin integraatio toteutettiin isomassa mittakaavassa.</p> <p>Työn tuotteena syntyi valmis ohjelmakirjasto projektipäälliköiden käyttöön. Projektipäälliköt pystyvät kopioimaan ohjelmakirjastoa omiin projekteihinsa sopivaksi, sillä ohjelmakirjastoon sisällytettiin kaikki tärkeät rekisterit, jotka olivat laitevalmistajan mukaan saatavilla.</p> <p>Ohjelmakirjaston käytölle luotiin selkeät ohjeet koodiin, jotta projektipäälliköt saisivat mahdollisimman pienellä vaivalla luotettavan integraation omiin projekteihinsa.</p> <p>Lisäksi työssä havaittiin, että ultraäänellä ilmavirtaa mittaavat ilmamääräsäätimet ovat luotettavampia, huoltovapaampia ja tarkempia tietyillä ilmavirroilla kuin ilmamääräsäätimet, jotka mittaavat ilmavirtaa perinteisillä mittayhteillä. Perinteiset mittayhteet aiheuttavat painehäviötä ja ne likaantuvat ajan mittaan. Kun ilmavirtaa mitataan ultraäänellä suoraan kanavan pinnasta, ei kanaviston likaantuminenkaan aiheuta painehäviötä mittayhteiden välille.</p>	
Avainsanat	kenttäväylät, rakennusautomaatio, ultraääni-ilmamääräsäädin

Author Title Number of Pages Date	Teemu Kortelainen Integration of Variable Air Volume Unit into Building Automation System with Modbus Serial Communication Protocol 48 pages + 2 appendices 22 September 2019
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Engineering
Instructors	Tero Wendelin, Head of Department Reijo Leinonen, Senior Lecturer
<p>This bachelor's thesis deals with integration of ultrasonic variable air volume unit into a building automation system with Modbus serial communication protocol. The work was commissioned by Fidelix Oy.</p> <p>The integration was done with real devices in a realistic test environment, so that any potential errors could be corrected at an early stage before the integration was done at a larger scale.</p> <p>As a result, fully completed program library was created for the project managers. Project managers can now copy the created program library to their own projects. The code was commented clearly, so that project managers can do their own integrations with as little effort as possible.</p> <p>In addition, it was found that ultrasonic air volume controllers are more reliable and more accurate in a certain air flow than older air volume controllers. Ultrasonic air volume controllers have no protruding parts, so when the air duct gets dirty, that will not interfere with measurement accuracy.</p>	
Keywords	fieldbuses, building automation, variable air volume unit

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Rakennusautomaatiojärjestelmä	1
2.1	Järjestelmän rakenne ja kaapelointi	2
2.2	Kenttälaitteet	3
2.2.1	Anturit	4
2.2.2	Toimilaitteet	7
2.3	Järjestelmälaitteet	9
2.3.1	Alakeskukset	9
2.3.2	I/O-kortit	11
2.3.3	Yksikkösäätimet	13
2.4	Valvomo	13
3	Kenttäväylät	14
3.1	Verkkotopologiat	15
3.2	Yleisimmät kenttäväyläratkaisut rakennusautomaatiojärjestelmissä	17
4	Ilmanvaihtojärjestelmät	21
4.1	Ilmanvaihtokone	22
4.2	Ilmamääräsäädin	23
5	Käytännön työosuus	24
5.1	Vaatimusmäärittelyt	24
5.2	Työn suunnittelu ja laitteen käyttöönotto	26
5.3	Human-Machine Interface	32
5.4	Pisteiden luominen	33
5.5	Ohjelmointi	36
5.6	Integraation testaus ja lopputulokset	40
6	Yhteenveto	46

Liitteet

Liite 1. Ilmamääräsäätimen Modbus-rekisterilista

Liite 2. Työn tuloksena syntynyt ohjelmakirjasto

## Lyhenteet

AI/AO	<i>Analog Input/Analog Output.</i> I/O-kortilla olevat analogiset tulo- ja lähtöpisteet.
ASCII	<i>American Standard Code for Information Interchange.</i> 7-bittinen, 128:n merkin laajuinen tietokonemerkitö.
BACnet	<i>Building Automation and Control Networks.</i> Standardoitu tiedonsiirtoprotokolla, jota käytetään yleisesti rakennusautomaatiojärjestelmissä.
CPU	<i>Central Processing Unit.</i> Suoritin/proessori, joka käsittelee tietokoneohjelman sisältämiä konekielisiä käskyjä.
DALI	<i>Digital Addressable Lightning Interface.</i> Valaistusohjaukseen suunniteltu tiedonsiirtoprotokolla.
DI/DO	<i>Digital Input/Digital Output.</i> I/O-kortilla olevat digitaaliset tulo- ja lähtöpisteet.
GND	<i>Ground.</i> Maadoittava liitin.
HMI	<i>Human-Machine Interface.</i> Ihmisen ja ohjelmoitavan logiikan välisen kommunikaation mahdollistava käyttöliittymä.
I/O-kortti	Tulo- ja lähtökortti, joka on suorassa yhteydessä kenttälaitteisiin.
LVIS	Lämpö, vesi, ilmanvaihto ja sähkö. Lyhennettä käytetään yleisesti rakennusalan suunnitelmissa ja muissa dokumenteissa.
M-Bus	<i>Meter-Bus.</i> Sarjaliikenneprotokolla mittaustietojen välitykseen.
Modbus	Sarjaliikenneprotokolla, jota käytetään väylälaitteiden integroinneissa.
RS-485	Standardi sarjaliikenneväylälle.

RTU	<i>Remote Terminal Unit.</i> Yksi kolmesta Modbus-kehyksistä.
TCP/IP	<i>Transmission Control Protocol/Internet protocol.</i> Usean tietoliikenneprotokollan yhdistelmä.
VAK	<i>Valvonta-alakeskus.</i> Automaatiojärjestelmien keskuskaappi.
VPN	<i>Virtual Private Network.</i> Virtuaalinen erillisverkko.

## 1 Johdanto

Insinöörityön tavoite on tehdä toimiva väyläintegraatio Fidelix-rakennusautomaatiojärjestelmän sekä ultraäänitekniikalla ilmavirtaa mittaavan Lindab UltraLink FTCU -ilmamääräsäätimen välille. Tavoitteena on tehdä toimiva ohjelmakirjasto, jota Fidelix Oy:n projektipäälliköt voisivat käyttää omissa projekteissaan. Ohjelmakirjaston toiminta sekä väyläintegraatio testataan oikeilla laitteilla realistisessa testiympäristössä. Ohjelmakirjasto tulee käyttöön Fidelix Oy:n urakoimissa kohteissa, joissa kyseinen ilmamääräsäädin liitetään rakennusautomaatiojärjestelmään Modbus-väylällä.

Ultraäänitekniikka on yleistynyt huomattavasti lähiaikoina, ja ultraäänitekniikalla mitattavien ilmamääräsäätimien väitetäänkin olevan vanhemmalla mittaustekniikalla mitattavia ilmamääräsäätimiä luotettavampi, tarkempi sekä huoltovapaampi, mikä on saanut monet yritykset mukaan tähän kilpailuun. Vanhempaa mittaustekniikkaa edustaa ilmamääräsäädin, jonka toimilaite mittaa kanavan ilmavirtaa paineyhteiden avulla. Tarkoituksena on vertailla näiden kahden tekniikan ominaisuuksia.

## 2 Rakennusautomaatiojärjestelmä

Rakennusautomaatio (toiselta nimeltään myös kiinteistöautomaatio) on olennainen osa kiinteistön LVIS-prosesseja. Rakennusautomaatiojärjestelmä ohjaa, säätää ja valvoo automaattisesti kiinteistön ilmanvaihtoa sekä lämmönjakoa muuttuvien olosuhteiden mukaisesti niin, että sisälämpötila, ilmamäärä sekä ilmanlaatu vastaavat kiinteistön olosuhdevaatimuksia sekä yleisiä määräyksiä.

Rakennusautomaatiojärjestelmän keskusyksikkönä käytetään eräänlaista tietokonetta, jonka tietokoneohjelmisto suorittaa siihen ohjelmoidut käskyt ja toiminnot I/O-korteilta saamiensa tietojen perusteella. Esimerkiksi käyttöveden menolämpötilaa säädetään käyttövesipiiriin asennetun lämpötila-anturin perusteella. Käyttöveden lämpötila-anturin mittaustieto saa käyttövesipiirin venttiilimoottorin avautumaan ja sulkeutumaan sen mukaan, kuinka paljon mittausarvo poikkeaa asetusravosta. Asetusravo syötetään suoraan rakennusautomaatiojärjestelmän käyttöliittymään.



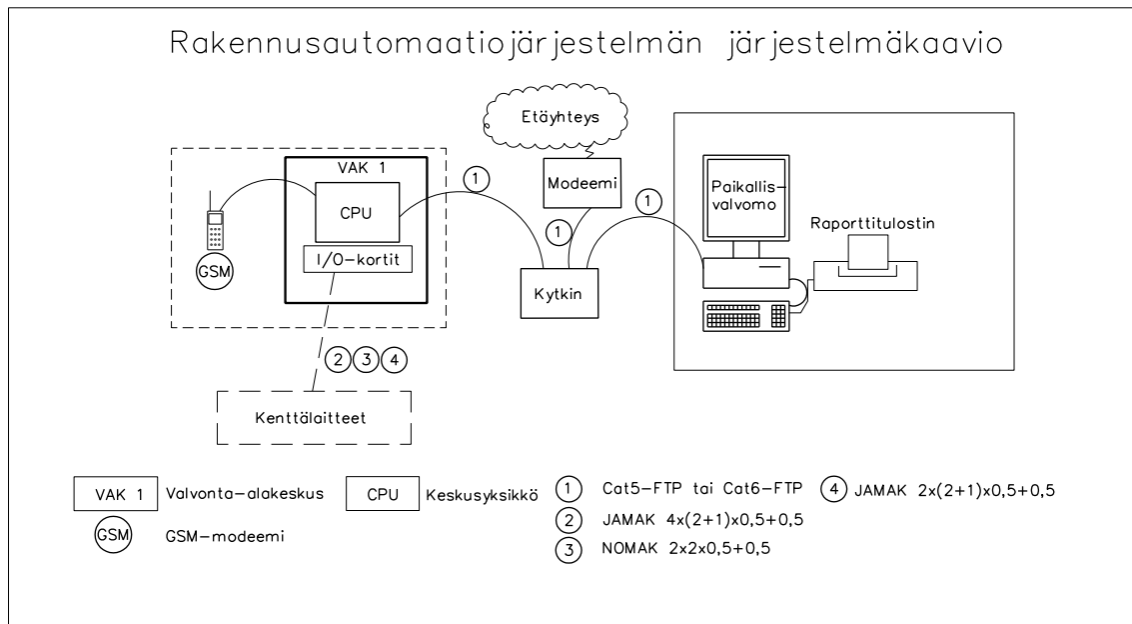
Järjestelmässä on yleensä myös lukuisia erillisohjauksia, joiden toiminnot räätälöidään jokaiselle kohteelle erikseen tarpeen mukaisesti. Yleisimpiä erillisohjauksia ovat valaistus-, sulatus- ja erillispoisto-ohjaukset.

Rakennusautomaatiojärjestelmä lisää kiinteistön viihtyvyyttä ja raportoi kiinteistön huololle kaikista mahdollisista poikkeamatapauksista, jotta kiinteistölle aiheutuisi mahdollisimman vähän vahinkoa. Järjestelmä voi esimerkiksi epäillä vesivuotoa vesimittarin kulutuksen perusteella tietynä ajanjaksona, hälyttää liiallisesta hiilidioksidipitoisuudesta sisätiloissa tai raportoida palopeltien toimimattomuudesta testijakson jälkeen.

Kiinteistön energia-, sähkö- ja vesimittaukset liitetään usein rakennusautomaatiojärjestelmään, jossa lukemia seurataan ja raportoidaan kuukausitasolla ja josta kiinteistön omistaja pystyy seuraamaan lukemia myös reaaliajassa. Rakennusautomaation avulla saavutetaan merkittäviä energiansäästöjä kiinteistölle, joten järjestelmästä saatavat energiaraportit ovat arvokasta dataa kiinteistön omistajalle.

## 2.1 Järjestelmän rakenne ja kaapelointi

Tavallisin rakennusautomaatiojärjestelmä koostuu valvonta-alakeskuksista, I/O-korkeista, antureista, toimilaitteista, valvomosta, hälytyksensiirtolaitteesta sekä kaapeleista, joiden avulla kaikki liitetään yhtenäiseksi kokonaisuudeksi. Kuvassa 1 on esitetty tavallisen rakennusautomaatiojärjestelmän järjestelmäkaavio. Järjestelmässä voi myös olla langattomia antureita ja toistimia, mutta niin sanottu langallinen järjestelmä on vielä huomattavasti yleisempi ja yleisesti ottaen myös luotettavampi.



Kuva 1. Rakennusautomaatiojärjestelmän järjestelmäkaavio.

Mittaavien antureiden kaapelit ovat tyypiltään parikierrettyjä, sillä kierrettyjen pariensa ansiosta ulkopuolisen magneettikentän aiheuttamat häiriövirrat kumoutuvat tehokkaasti, eikä pieni häiriövirta pääse vaikuttamaan mittaustulokseen. Häiriövirtaa voi aiheuttaa esimerkiksi anturikaapelia lähellä oleva voimakaapeli, jossa kulkee suuria virtoja ajoittain tai jatkuvasti. Normaalisti nämä kaapelit pyritään pitämään erillään kaapelihyllyllä, mutta risteyskohdissa kaapelit saattavat kohdata. Kaapeleissa on lisäksi häiriösuojalanka, joka kytketään vain valvonta-alakeskuksen päästä erilliseen häiriösuojaliittimeen.

## 2.2 Kenttälaitteet

Kenttälaitteiksi kutsutaan rakennusautomaatiojärjestelmään liitettyjä mitta- ja toimilaitteita. Mittalaitteita voivat olla esimerkiksi anturit, jotka mittaavat lämpötilaa, painetta, kosteutta, hiilidioksidipitoisuutta tai valoisuutta.

Toimilaitteet voivat olla pelti- tai venttiilimoottoreita, magneettiventtiileitä tai vahamoottoreita.

### 2.2.1 Anturit

Anturit ovat mittalaitteita, joiden avulla voidaan mitata erilaisia fysikaalisia suureita. Anturit antavat järjestelmälle aistit, ja niiden lähettämän mittaustiedon perusteella järjestelmä havainnoi prosessin tiloja ja toimintoja [1, s. 8]. Yleisimmät rakennusautomaatiojärjestelmässä käytetyt anturit ovat tyypiltään lämpötila-, paine-, kosteus- ja hiilidioksidiantureita sekä valoisuutta mittaava valoisuusanturi.

Lämpötila-antureita (kuva 2) asennetaan putkistoihin, ilmanvaihtokanaviin sekä huonetiloihin. Lämpötila-anturit voivat olla tyypiltään joko säätäviä tai informoivia. Säätävien lämpötila-antureiden taustalle ohjelmoidaan yleensä PI-säädin, joka laskee mitattavan lämpötila-anturin lukeman sekä asetusravon perusteella toimilaitteelle arvon. Yleisimmin toimilaitte saa aikaan liikkeen, joka lisää tai vähentää lämpöä esimerkiksi patterilla tai lämmönsiirtimellä. Esimerkiksi ilmanvaihtokoneessa on lämmityspatteri, johon on liitetty lämmönjakokeskuksesta tulevat putket. Tässä tapauksessa toimilaitte on venttiilimoottori, joka venttiilirunkoineen on asennettu patterin paluupuolen putkeen. Venttiilimoottorin avautuessa patteri lämpenee ja näin ollen myös ilmanvaihtokanavassa patterin läpi liikkuva ilma lämpenee.

Informatiiviset lämpötila-anturit eivät ole säätäviä. Näiden lämpötila-antureiden dataa voidaan kuitenkin tallentaa historiatrendeihin, joista saadaan lämpötilatietoja pidemmältäkin ajanjaksolta.



Kuva 2. ProDualin kanavalämpötila-anturi. [2.]

Paineanturit (kuva 3) mittaavat useimmiten joko ilmanvaihtokanavan tai vesipiirin painetta. Vesipiirissä olevat paine-eroanturit ovat usein tyypiltään säätäviä, ja ne säätävät

piirissä olevan pumpun tai pumppujen kierroslukua. Vesipiirin liitetyt paineanturit voivat olla myös informatiivisia, jolloin paineantureille asetetaan ala- ja ylärajahälytykset.

Tulo- ja poistoilmanvaihtokanavissa olevat paineanturit mittaavat kanavan painetta ja ovat lähes aina säätäviä antureita. Paineanturit säätävät puhaltimen kierroslukua painemittauksen perusteella pitäen paineen aina tasaisena ja asetusarvonsa mukaisena. Ilmanvaihtokoneeseen asennetaan yleensä myös paine-eroanturit mittaamaan suodattimien paine-eroa. Mitä likaisempi suodatin, sitä suuremman paine-eron paineanturi mittaa suodattimen yli. Paineanturille asetetaan yleensä ylärajahälytys, mikä kertoo suodattimen likaisuudesta ja vaihtotarpeesta.



Kuva 3. Paine-eroanturi. [3.]

Hiilidioksidianturit (kuva 4) mittaavat joko huoneilman tai poistoilmakanavaan asennetun hiilidioksidianturin perusteella ilman hiilidioksidipitoisuutta. Yleensä nämä mittaukset ohjelmoidaan säätäviksi mittauksiksi. Mittaustuloksen perusteella säädetään tilan ilmamäärää joko portaattomasti muunnostaulukon avulla tai yhdellä raja-arvolla on/off-tyyppisesti.

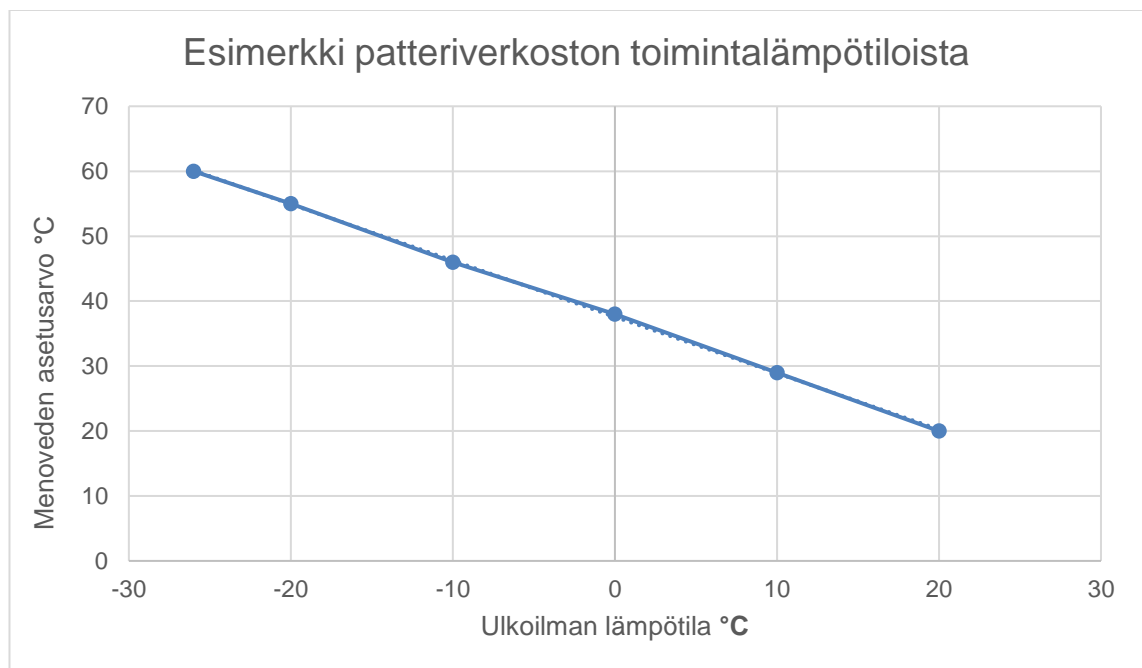
Hiilidioksidipitoisuus nousee, mikäli suuri määrä ihmisiä on yhdessä ja samassa tilassa. Tämä tilanne saa aikaan hiilidioksidipitoisuuden nousun ja näin ollen myös ilmanlaadun heikkenemisen kyseisessä tilassa. Ilmanlaadun parantamiseksi hiilidioksidipitoisuuden

nousu saa aikaan ilmamäärän tehostamisen kyseisessä tilassa – joko ilmamääräsäätimen tai ilmanvaihtokoneen puhaltimen avulla.



Kuva 4. Hiilidioksidianturi. [4.]

Rakennuksen pohjoisseinälle asennettava ulkolämpötila- ja valoisuusanturi on koko rakennusautomaatiojärjestelmän tärkein anturi. Anturin mittaama ulkolämpötila säätää lämmitysverkostojen menoveden asetusarvoja ulkolämpötilaan verrannollisen käyrän mukaisesti. Kuvassa 5 on esitetty esimerkkikohteen patteriverkoston menoveden säätökäyrä ulkolämpötilan mukaan.



Kuva 5. Patteriverkoston toimintalämpötilat esimerkkikohteessa.

Anturin tärkeyden vuoksi asennuspaikkaan on kiinnitettävä erityistä huomiota. Anturi asennetaan pohjoisseinälle siitä syystä, ettei aurinko pääsisi paistamaan suoraan anturiin. [5, s. 1.] Anturin ylle voidaan myös asentaa erillinen lippa, mikäli anturin lukema häiriintyy auringonpaisteesta kohtuuttoman paljon.

Anturin valoisuusmittausta käytetään ulkovalaistusryhmien ohjaamiseen. Valaistushajauksille ohjelmoidaan syttymis- ja sammumisrajat, joiden mukaan valaistukset käyvät. Anturi tulisi asentaa kauas ulkovaloista, ettei niiden aiheuttama valo heijastuisi suoraan anturiin. Tämä aiheuttaa muuten ilmiön, jossa valot syttyvät ja sammuvat mielivaltaisesti, vaikka ulkona olisi pimeää.

### 2.2.2 Toimilaitteet

Toimilaite voi olla esimerkiksi venttiilimoottori, pumppu, peltimoottori tai ilmamääräsäädin, joka säätyy säätimen tai muun ohjelmakoodin seurauksena [6, s. 8]. Toimilaitteita kutsutaan automaatiojärjestelmän ”lihaksiksi”. Ne aiheuttavat toimiessaan yleensä liikettä, minkä seurauksena ilmamäärä, lämpötila tai virtausnopeus muuttuu – joko suhteellisesti, eli portaattomasti, tai auki/kiinni-tyyppisesti [7, s. 6].

Käyttövesipiirin venttiilimoottorin (kuva 6) on oltava nopea toiminnaltaan, jotta verkoston lämpötilamuutokset saadaan tasattua nopeasti. Normaalin käyttövesipiirin venttiilimoottorin ajoaika on 9–15 s.



Kuva 6. Belimon venttiilimoottori LRQ24A-SR, jossa ajoaika 90°:n kulmalle on 9 sekuntia. [8.]

Pumput (kuva 7) siirtävät nestettä paikasta toiseen. Neste virtaa putkistoissa matalammasta paineesta korkeampaan paineeseen päin. [9, s. 7.] Pumppuja ohjataan rakennusautomaatiojärjestelmästä yleensä on/off-periaatteella, eli ohjaus on joko päällä tai pois päältä. Mikäli pumpun kierroslukua ohjataan suhteellisesti 0–100 %:n säädöllä, se säätyy yleensä putkistoon asennetun paine-eroanturin mittaustuloksen perusteella.



Kuva 7. Grundfosin MAGNA3-kiertovesipumppu. [10.]

Peltimoottorit (kuva 8) ovat moottoreita, jotka asennetaan ilmanvaihtokanavaan asennetun ilmanvaihtopellin akseliin kiinni. Pelti voi olla joko ilmanvaihtokoneen sisällä tai palvelualueella kiinteistössä. Pelti estää sulkeutuessaan ilman liikkeen joko osittain tai kokonaan. Pellillä voidaan myös ohjata lisäilmanvaihtoa tilaan.



Kuva 8. Belimon jousipalautteinen SF24A-peltimoottori. [11.]

Ilmanvaihtokoneessa käytettävät sulkupeltimoottorit ovat tyypiltään aina jousipalautteisia, mikä tarkoittaa sitä, että jos peltimoottorilta katkeaa jännite syystä tai toisesta, peltimoottori palautuu jousivoimalla kiinni. Peltimoottoreita voidaan ohjata joko on/off-tyyppisesti tai suhteellisesti käyttötarkoituksen mukaan.

## 2.3 Järjestelmälaitteet

Järjestelmälaitteet muodostavat keskenään koko automaatiojärjestelmän perustan. Järjestelmälaitteet käsittelevät antureilta mitatut tiedot ja tulostavat ne järjestelmän käyttöliittymään, josta käyttäjä voi niitä tarkastella.

### 2.3.1 Alakeskukset

Automaatiojärjestelmässä voi olla yksi tai useampia alakeskuksia. Alakeskuskaappiin tuodaan antureiden ja toimilaitteiden kaapelit, jotka kytketään lopuksi I/O-kortteille. Alakeskuksessa on automaatiojärjestelmän keskusyksikkö (kuva 9), johon on myös yleensä rakennettu käyttöliittymä käyttäjää varten.

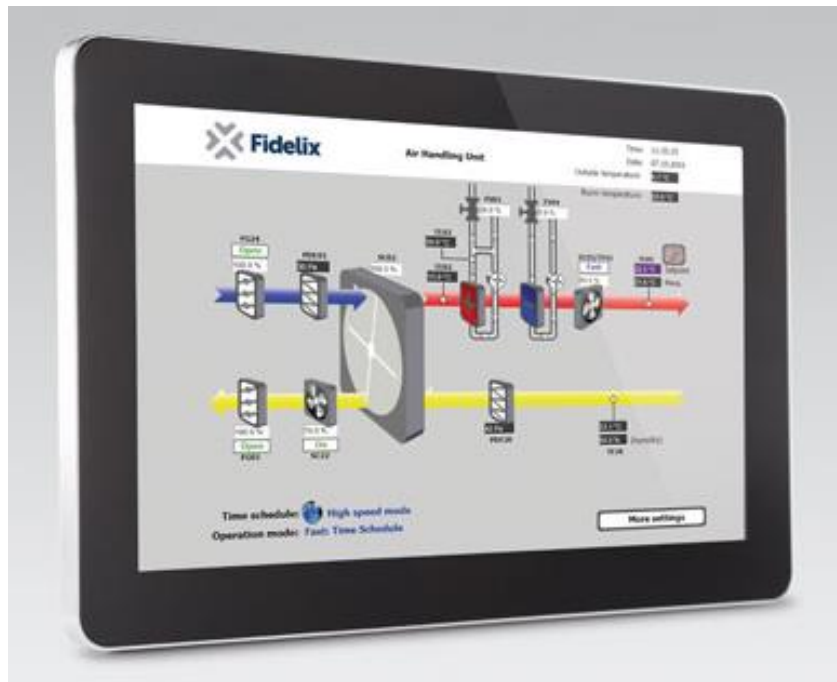




Kuva 9. Fidelixin automaatiojärjestelmän keskusyksikkö FX-3000C, johon liitetään tässä työssä mainittu ilmamääräsäädin. [12.]

Keskusyksikköä pidetään automaatiojärjestelmän ”aivoina”, sillä kaikki loogiset komennot suoritetaan täällä (ellei automaatiojärjestelmässä ole säätimiä, jolloin osa toiminnoista on voitu siirtää säätimen omaan logiikkaan).

Keskusyksiköihin on voitu integroida paikallinen näyttö käyttöliittymää varten, tai vaihtoehtoisesti näyttö voi olla esimerkiksi ulkoinen kosketusnäyttö, joka on liitetty keskusyksikköön. Kuvassa 10 on esitetty Fidelix-automatiojärjestelmän keskusyksikön FX-3000C:n käyttämistä varten tarkoitettu kosketusnäyttö Visio-15-C.



Kuva 10. 15,6-tuumainen kosketusnäyttölinen ja Android-pohjainen käyttöpaneeli Fidelix-automatiojärjestelmän käyttöä varten. [13.]

Keskusyksiköt ovat tavallisesti IP-pohjaisia laitteita, ja ne voidaan liittää internetiin etäkäyttöä varten. Tällöin on aina huomioitava tarpeeksi vahva tietoturvaso hyökkäyksiä varten. Nykyisin markkinoilla on tarjolla suojatulla VPN-yhteydellä varustettuja etäyhteyksratkaisuja, joiden avulla etäyhteyden muodostaminen onnistuu helposti ja turvallisesti.

### 2.3.2 I/O-kortit

Kaikki I/O-kenttälaitteet kytketään valvonta-alakeskuksissa I/O-korteille (kuva 11). I/O-kortit keskustelevat keskusyksikön kanssa yleensä väylän avulla.



Kuva 11. Fidelix Oy:n Compact-sarjan I/O-moduulit. [14.]

Digitaaliset tulopisteet (DI-pisteet) lukevat laitteiden kosketintietoja, jotka ovat tyypeiltään pääosin tilatietoja ja hälytyksiä. Esimerkiksi pumpulta voidaan ottaa käyntitilatiedon lisäksi hälytyspiste, jotka ovat molemmat DI-pisteitä.

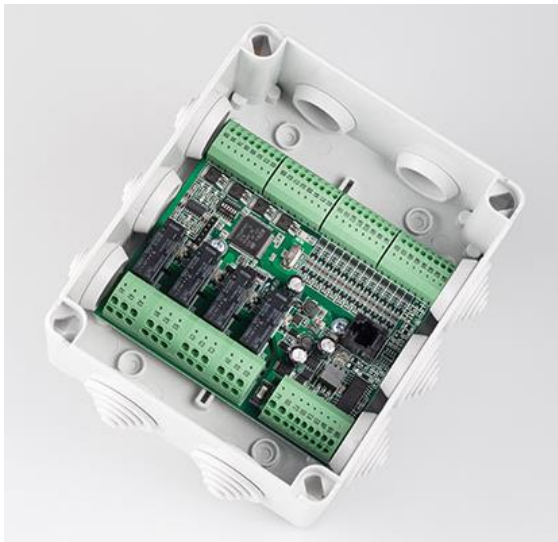
Digitaaliset lähtöpisteet (DO-pisteet) ohjaavat toimilaitteita ja erillispisteitä on/off-tyyppisesti. Erillispisteohjaus voi olla tyyppiltään esimerkiksi ulkovalo-ohjaus, jota ohjataan ryhmäkeskuksen kautta. Ryhmäkeskuksen ulkovaloryhmässä on automaattiasento, jonka pitää olla päällä, jotta automaatiojärjestelmä voi ohjata valoja.

Analogisilla tulopisteillä (AI-pisteet) erilaisten antureiden tietoja. Anturi voi olla tyyppiltään joko aktiivinen tai passiivinen. Aktiivisiin antureihin tuodaan automaatiojärjestelmästä käyttöjännite. Anturi lähettää automaatiojärjestelmään takaisinkytkentätietona 0–10 VDC:n tai 4–20 mA:n lähtöviestin mitatusta mittaustuloksesta. Mittaustulos skaalataan keskusyksiköllä anturin datalehden mukaisesti. Esimerkiksi paineanturin mittaustulos 0–10 VDC voidaan skaalata anturilla 0–1000 Pa:n asteikolle. Tällöin anturin lähettämä 5 VDC:n mittaustulos tarkoittaisi 500 Pa:n painetta, joka pitäisi näkyä automaatiojärjestelmän käyttöliittymässä.

Analogisilla lähtöpisteillä (AO-pisteet) ohjataan eri prosessien toimilaitteita, kuten esimerkiksi venttiili- ja peltimoottoreita portaattomasti. Useat eri prosessit tarvitsevat toimiakseen tarkan portaattoman säädön, mikä onnistuu analogisten pisteiden ja soveltuvien laitteiden avulla. Analogisilla lähtöpisteillä voidaan ohjata myös puhaltimien ja pumppujen kierroslukua.

### 2.3.3 Yksikkösäätimet

Yksikkösäädin (kuva 12) on laite, joka suorittaa siihen ohjelmoituja käskyjä itsenäisesti, ilman automaatiojärjestelmän keskusyksikön tukea. Yksikkösäätimiä voidaan käyttää pienemmissä sovelluksissa, kuten esimerkiksi kaukolämpöratkaisussa, hotelli- ja asuinhuoneistoissa, sekä ilmanvaihtojärjestelmissä [15, s. 1].



Kuva 12. Multi-24-yksikkösäädin. [15.]

Säätimet voidaan liittää automaatiojärjestelmään väylän avulla, jonka kautta säädin voi kertoa automaatiojärjestelmän keskusyksikölle siihen liitettyjen laitteiden tiloista, tai vaihtoehtoisesti automaatiojärjestelmä voi käskyttää yksikkösäädintä ja siihen liitettyjä laitteita yksikkösäätimen ”ohi”.

## 2.4 Valvomo

Valvomot voidaan katsoa kuuluvaksi automaatiojärjestelmien kolmannelle eli ylimmälle tasolle hierarkianäkökulmasta. Valvomot toimivat rajapintana ihmisen ja järjestelmän välillä. Tätä kutsutaan HMI:ksi (Human Machine Interface). [16, s. 13.] Paikallisvalvomoon liitetään kiinteistön kaikki valvonta-alakeskukset kootusti. Paikallisvalvomon grafiikkakuviin tehdään koontikuvia eri järjestelmistä. Koontikuvaan voidaan liittää esimerkiksi kiinteistön kaikki lämpötilamittaukset, poistopuhaltimet, ilmanvaihtokoneet, erillispisteet tai

valo-ohjaukset eri alakeskuksilta. Tämä helpottaa huomattavasti kiinteistöhoitajan työtä, sillä kaikki on nähtävillä yhdellä sivulla nopeasti. Paikallisvalvomo on yleensä normaali PC, joka on sijoitettu kiinteistön valvomohuoneeseen. Täältä kiinteistöhoitaja voi tarkistaa järjestelmän toiminnot esimerkiksi heti aamusta.

Keskitetty järjestelmä valvoo useiden rakennusten automaatiojärjestelmien alakeskuksia ja niiden toimintoja. Valvomo on voitu asentaa esimerkiksi kiinteistöhoitoyrityksen tiloihin, kaupungin varikolle tai muuhun keskusvalvomotilaan. [17, s. 169.] Nykyisin valvomo-ohjelmisto voidaan asentaa suoraan ”pilveen”, eli tässä tapauksessa kaikki data tallennetaan sinne oikean valvomo-PC laitteiston sijasta. Kun ohjelmisto on pilvessä, ei laiterikkoja tarvitse miettiä.

### 3 Kenttäväylät

Kenttäväyläratkaisut mahdollistavat suuremman tiedonsiirtomäärän pienemmällä kaapeloinnilla verrattuna perinteiseen I/O-tekniikkaan. Kenttäväyliin liitetään älykkäitä antureita, ohjauslaitteita sekä näyttöjä, jotka yhdessä ohjauskeskusten kanssa muodostavat kaksisuuntaisen, digitaalisen tiedonsiirtoverkon. [18, s. 32.] Älykkäät väyläpohjaiset kenttälaitteet osaavat kertoa mahdollisesta huoltotarpeesta tai viasta tarkasti eri rekistereiden avulla. Lisäksi toimilaitteita saadaan ohjattua monipuolisilla menetelmillä.

Erään arvion mukaan väyläpohjainen ratkaisu vähentäisi kaapelointia ja kytkentäpisteiden sekä -liityntöjen määrää jopa 60–70 %. Tämä vähentää huomattavasti kytkentävirheiden määrää sekä asennuskuluja. [18, s. 33.] Kenttäväylälaitteiden ohjelmointi ja käyttöönotto on kuitenkin hieman työläämpää, ellei laitteen ja järjestelmän välille ole jo tehty valmista integraatiota.

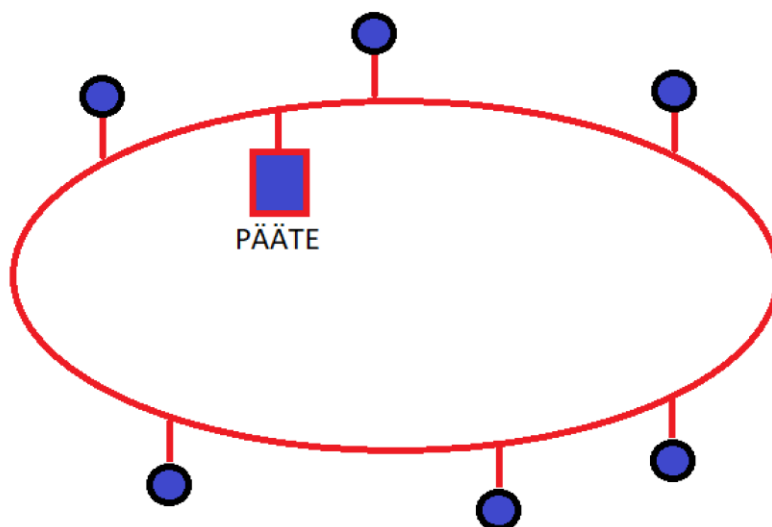
Rakennusautomaatiojärjestelmien toimittajat ohjelmoivat uuden kenttäväylälaitteen valmistajan lähettämän dokumentaation mukaisesti. Dokumenteissa on mainittu tiedonsiirtoasetukset, kuten väylänopeus (baud rate), laitteen osoite, viestien käsittelytapa väylässä ja muut kenttäväyläprotokollaan oleellisesti liittyvät asiat.

### 3.1 Verkkotopologiat

Verkkotopologiat kuvaavat eri kaapelointitapoja. Kaapelit ovat osa verkon fyysistä rakennetta, joissa kulkee tietoa laitteelta laitteelle. Verkon suunnittelussa on huomioitava väylän taloudellisuus, tehokkuus ja toimivuus. Kaapelia on voitava vetää mahdollisimman vähän, mutta riittävästi, jotta järjestelmä toimisi ilman häiriöitä. Yleisimpiä verkkotopologioita ovat rengas-, väylä- ja tähtitopologiat. [18, s. 71.]

#### Rengastopologia

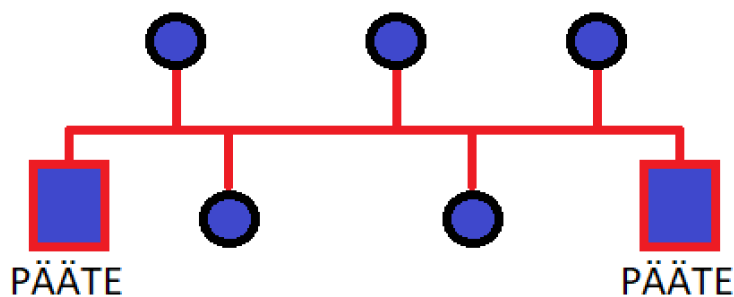
Jos lähdetään seuraamaan kaapelia pisteestä A, ja päädytään pian takaisin samaan pisteeseen, on verkkotopologia tällöin rengastopologia (kuva 13). Tällöin rengastopologiassa verkon runkokaapeli muodostaa rengasmaisen kaapeloinnin laitteiden välille. Pääte lähettää sanoman, mikä kulkee asemalta toiselle, eli sanoma tavoittaa aina kerrallaan vain yhden aseman. Rengastopologia on varsin yleinen verkkotopologia, mutta sen suurena heikkoutena on järjestelmän haavoittuvuus. Mikäli jokin asemista rikkoonuu, koko väylän kommunikaatio saattaa pysähtyä täysin, sillä jokainen asema toimii samalla signaalitoistimena, eli asema lukee kehuksen verkosta ja lähettää sen eteenpäin. Mikäli siis yksi asema rikkoonuu, ei sanoma jatkakaan enää kulkuaan renkaassa eteenpäin. [18, s. 73–74.]



Kuva 13. Rengastopologian periaatekaavio.

## Väylätopologia

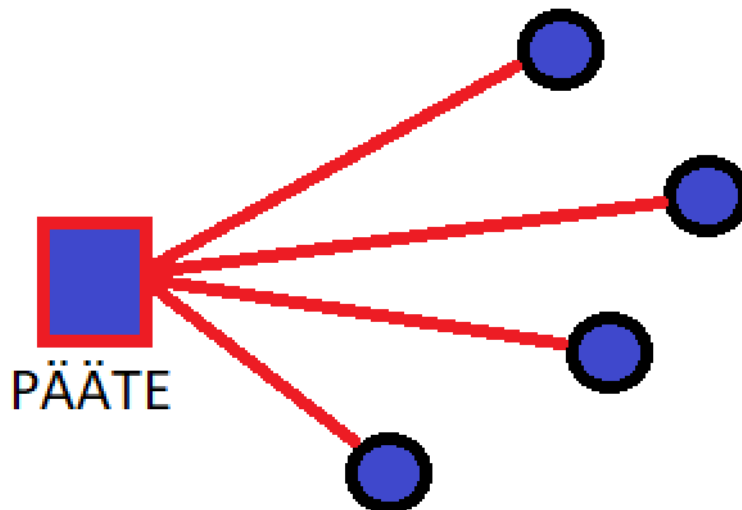
Väylätopologiassa (kuva 14) kaikki asemat on kiinnitetty samaan kaapeliin, ja kaapeli-tyyppinä käytetään yleisimmin kierrettyä parikaapelia. Väylän päihin kytketään aina pääte-  
tevastukset, jotka oikein mitoittamalla saadaan eliminoitua lähes kokonaan signaalin heijastumat. Signaalin heijastumat heikentävät, tai pahimmillaan kaatavat koko väylän toiminnan. Jokaisella asemalla on omat osoitteensa, joiden ansiosta kommunikointi voi tapahtua luotettavasti. Mikäli asema A haluaa lähettää sanoman asemalle B, se odottaa lupaa lähettää sanoma verkkoon. Tämän jälkeen se lähettää B:n osoitteella varustetun sanoman verkkoon, josta asema B voi kyseenomaisen sanoman poimia. Asemat C tai D voivat kuunnella sanomaa, mutta eivät lukea sitä, sillä sanomaa ei ollut osoitettu niille. Sanomat ovat tyyppiltään paketteja, ns. kehyksiä, jotka lähetetään verkkoon ja joita vastaanottava asema voi lukea. [18, s. 72–73.]



Kuva 14. Väylätopologian periaatekaavio.

## Tähtitopologia

Tähtitopologian (kuva 15) rakenne koostuu asemista, jotka kytketään tähden keskusosan kytkentäkeskukseen. Kaapelina voidaan käyttää kierrettyä parikaapelia, optista kuitua tai koaksiaalikaapelia. Tähtitopologian yksi suurimmista heikkouksista on kytkentäkeskuksen vikaantumisen aiheuttama vahinko, joka pahimmillaan voi tehdä koko verkon toimintakyvyttömäksi. Ongelmaa voidaan kuitenkin lieventää lisäämällä kytkentäkeskus verkkoon, mutta se lisää kustannuksia. Tästä huolimatta tähtitopologia on erittäin yleinen ja paljon käytetty eri ratkaisuisissa. [18, s. 74–75.]



Kuva 15. Tähtitopologian periaatekaavio.

### 3.2 Yleisimmät kenttäväyläratkaisut rakennusautomaatiojärjestelmissä

Maailmassa on olemassa lukuisia eri kenttäväyläratkaisuja, joita voidaan hyödyntää eri tilanteissa. Kiinteistöt ja laitteet ovat erilaisia, ja eri kenttäväyläratkaisut sopivat eri tilanteisiin. Esimerkiksi kiinteistön veden- ja sähkönkulutuksen lukemiseen voidaan käyttää M-Bus-väylää, jota käytetään yleisesti mittariluentajärjestelmissä. Modbus-väylän avulla voidaan ohjata ilmanvaihtokonetta tai vain lukea sen lähettämiä mittaustietoja tai muita parametreja. DALI-väylä taas soveltuu parhaiten valaistusratkaisuihin.

#### Modbus

Modbus-tiedonsiirtoprotokolla perustettiin vuonna 1979. Modbus on avoimeen arkkitehtuuriin perustuva tiedonsiirtoprotokolla ohjelmoitaville logikoille, eikä se maksa käyttäjälle tai toimittajalle mitään. Modbus-protokolla on yleinen teollisuuden sovelluksissa, rakennusautomaatiossa sekä pitkän matkan tiedonsiirrossa, ja se soveltuu myös etävalvontaratkaisuihin. [18, s. 243.]

Modbus perustuu avoimeen isäntä-renki protokollaan, jota voidaan hyödyntää helposti sarjaliitännöissä. Yhdellä isännällä voi olla maksimissaan 247 renkiä. OSI-mallit ovat taso 1 (fyysinen), taso 2 (siirtoyhteys) ja taso 3 (sovellus). Modbus RTU (Remote



Terminal Unit), Modbus ASCII (American Standard Code for Information Interchange) ja Modbus over TCP/IP (Transmission Control Protocol/Internet Protocol), ovat Modbus-kehyksiä. Modbus over TCP/IP -versiota käytetään Ethernetin liitännöissä, kun taas RTU- ja ASCII-versiot soveltuvat perinteisille sarjaväylille (esim. RS-485). [18, s. 244.]

ASCII-versiossa sanoman rakenteena on 8-bittinen tavu, joka lähetetään kahtena ASCII-merkkinä. RTU:n modbus-sanoman perusrakenne on 8-bittinen tavu, joka lähetetään kahtena 4-bittisenä heksamerkkinä (taulukko 1). [18, s. 244.]

Taulukko 1. Modbus-sanoman perusrakenne.

Aloitus	Osoite	Toiminto	Data	Tarkistus-summa	Lopetus
4 merkkiä	1 tavu	1 tavu	1 tavu	2 tavua	4 merkkiä

Modbus-protokollan perusversiossa voi olla vain yksi isäntälaitte. Tiedonsiirrossa isäntälaitte lähettää renkilaitteelle käskyn, jossa isäntälaitte pyytää renkilaitetta palauttamaan tietoa halutusta rekisteristä. Funktiot ovat Modbus-tietoliikenteen perusta – niillä saadaan määriteltyä, käytetäänkö rekistereitä lukemiseen, kirjoittamiseen vai kenties molempiin. Modbus-versiosta riippumatta isäntälaitte lähettää renkilaitteelleen funktiokoodin ja sen parametrin, jossa on mainittu muun muassa kirjoittamisen alkuosoite, kirjoitettavien sanojen määrä ja kirjoitettava tieto. Näiden lisäksi isäntälaitte lähettää renkilaitteelleen Modbus-RTU-versiossa renkilaitteen osoitteen, funktiokoodin, funktion datan ja CRC-tarkistussumman (16 bittiä). Tässä versiossa luvut lähetetään heksalukuina sellaisenaan, kun ASCII-versiossa heksaluvut hajotetaan kahdeksi ASCII-merkkiksi. Kun renkilaitte on saanut käskyn isännältään, renkilaitte toteuttaa sen. Toiminnon onnistuessa renkilaitte palauttaa isännälle vastauskehyksen, joka sisältää isäntälaitteen lähettämän käskyn mukaiset tiedot. [18, s. 244–245.]

Topologiana voidaan käyttää joko kaksipisteyhteyttä (Point-To-Point), joka toimii sekä RS-232 että RS-422-järjestelmissä, tai monipisteyhteyttä, jota käytetään vain RS-485-

järjestelmissä. RS-485-järjestelmän isännällä voi olla 31 renkiä, kun taas RS-232-järjestelmän isännällä voi olla vain yksi renki. RS-485-järjestelmän kaapelin maksimipituus voi olla jopa 1,2 kilometriä, kun taas RS-232-järjestelmässä kaapelin pituus voi olla maksimissaan vain 15 metriä. RS-485-väylässä suositellaan käytettävän suojattua parikaapelia, jonka molempiin päihin on kytketty asianmukainen päätevastus. [18, s. 245.]

## M-Bus

M-Bus on standardoitu avoin protokolla, jota käytetään erityisesti mittaustietojen siirtämiseen. Järjestelmä koostuu mittalaitteista (esimerkiksi vesimittarit), sekä keskuslaitteesta. Keskuslaitteeseen kerätään mittalaitteilta saatu data, jota voidaan keskuslaitteen liitännämahdollisuuksista riippuen lukea esimerkiksi etäyhteyden välityksellä. Mittalaitteita kutsutaan väyläympäristössä usein myös päätelaitteiksi. M-Bus-järjestelmässä tavallisin päätelaite on vesi-, sähkö- tai energiamittari. [19, s. 2.]

Vaikka M-Bus-protokolla ei sovellu suoranaisesti päätelaitteiden lähettämien hälytysten siirtämiseen, voidaan keskuslaitteella kääntää hälytyskoodi ymmärrettävään muotoon. Tätä ei siis suositella, mutta se on silti mahdollista. [19, s. 2.]

M-Bus-protokollan suurimpia etuja ovat sen toimintavarmuus, avoimuus ja helppokäyttöisyys. Kaapelointi voidaan tehdä yleisimmillä parikierrityillä kaapelityypeillä. Väylä toimii lähes kaikilla verkkorakenteilla, kuten puu-, väylä- ja tähtirakenteilla sekä näiden yhdistelmillä. [19, s. 2.]

Tiedonsiirto tapahtuu aina yksisuuntaisena. Keskuslaite lähettää päätelaitteille pyynnön datan siirtoa varten, jonka jälkeen päätelaite vastaa keskuslaitteelle kyseiseen pyyntöön. Keskuslaite voi pyytää vaikkapa vesimittarilta vedenkulutuksen lukemaa, jolloin vesimittari vastaa keskuslaitteelle mitatun vedenkulutuksensa. [19, s. 3.]

M-Bus-järjestelmän keskuslaitteen väyläliitynnän tyhjäkäyntijännite tulee olla noin 39 VDC. Jotta väyläliikenne toimisi luotettavasti, on keskuslaitteelta katsottuna kaukaisimman päätelaitteen väyläjännite oltava vähintään 24 VDC. [19, s. 4.]

Väylän laajuuteen ja suorituskykyyn vaikuttaa monta eri tekijää. Väylän pituus voi olla jopa useita kilometrejä, mutta tällöin tiedonsiirtonopeutta joudutaan laskemaan (Taulukko 2).

Taulukko 2. Verkon maksimipituuksia eri tiedonsiirtonopeuksilla. [19, s. 4.]

Liikennöinti nopeus (baudia)	Verkon maksimi kokonaispituus (parikapasitanssi 150 nF/km)
9600	1 km
2400	4 km
300	12 km

Suuria kokonaisuuksia suunniteltaessa on huomioitava järjestelmän tiedonsiirtonopeus, mikä lyhenee huomattavasti isommissa kokonaisuuksissa. Viestin pituus voi kasvaa pahimmillaan tunteihin, eikä sellaista voida sallia esimerkiksi tuntiraportointijärjestelmissä. Tiedonsiirtonopeuksia pystytään kasvattamaan esimerkiksi keskuslaitteita lisäämällä, jolloin väylän pituus lyhenee ja päätelaitteiden määrä vähenee.

## BACnet

BACnet-verkossa olevat laitteet ovat objekteja (objects) ja ne koostuvat joukosta ominaisuuksia (properties). Objekteiksi voidaan luokitella esimerkiksi aikaohjelmat, asetusravot, järjestelmäpisteet ja kalenteriohjelmat. Laitetoimittajien pitää toimittaa tuoteseloste BACnet-ominaisuuksista järjestelmän hallintaa varten, sillä se on välttämätön dokumentti järjestelmän hallinnassa. Tuoteseloste tunnetaan myös nimellä PICS (Protocol Implementation Conformance Statement). [19, s. 226.]

BACnet-tiedonsiirtoprotokolla on suunniteltu pääasiassa rakennusautomaatiojärjestelmien tarpeita varten. Liityttäessä BACnet-rajapintaan, rakennusautomaatiojärjestel-

mässä täytyy olla tuki BACnetin standardoimalle funktioiden esitystavalle. Järjestelmän tulee täyttää vaatimukset binäärisille ja analogisille tuloille ja lähdöille, jotta kommunikatio toimisi oikein. Esimerkiksi analogisen tulon tärkeimpiä ominaisuuksia ovat anturin tyyppi, sijainti ja hälytysrajat. Kaikki vaatimukset eivät kuitenkaan ole pakollisia, vaan osa niistä on vapaaehtoisia, eli niitä voidaan käyttää tarvittaessa. [20.]

## DALI

DALI on valaistuksen ohjaukseen suunniteltu tiedonsiirtoprotokolla. DALI on lyhenne sanoista ”Digital Addressable Lightning Interface”. Se on suunniteltu yksinkertaiseksi ja toimintavarmaksi tiedonsiirtoprotokollaksi. DALI-väylää käytetään tavallisimmin silloin, kun on tarve säätää ja ohjata esimerkiksi yksittäisiä valaisimia, joita on työlästä ohjata vanhanaikaisilla analogisilla ohjaimilla. [21.]

## 4 Ilmanvaihtojärjestelmät

Ilmanvaihdolla on tarkoitus poistaa kiinteistön sisäilmasta epäpuhtauksia ja kosteutta ja tarjota tilalle raikasta ja sopivan lämpöistä korvausilmaa. Ilmanvaihto voi olla toteutettu täysin painovoimaan perustuvalla tavalla, jossa kiinteistön ilmanvaihto tapahtuu ulko- ja sisäilman lämpötilaerojen ja tuulen aiheuttaman paine-eron avulla. Ilmanvaihtojärjestelmä voi olla toteutettu myös koneellisella poistoilmanvaihdolla, jossa kiinteistössä sijaitsevat huippuimurit tehostavat ilmanvaihtoa poistaen kiinteistöstä ilmaa ulospäin koneellisesti. Nykyaikana yleisin järjestelmä on kuitenkin koneellinen tulo- ja poistoilmanvaihto, jossa koko kiinteistön ilmanvaihto toteutetaan koneellisesti tehokkaan ilmanvaihdon aikaansaamiseksi. Koneellisen tulo- ja poistoilmanvaihdon suurimpina etuina ovat energiatehokkuus ja tasaisen ilmanvaihdon jatkuva saanti. Lämmöntalteenoton avulla jo valmiiksi lämmitetty sisäilman lämpötila (poistoilma) saadaan talteen ja puhallettua takaisin tuloilmaan. [22.]

#### 4.1 Ilmanvaihtokone

Ilmanvaihtokone kierrättää ilmaa kiinteistössä. Useimmissa kiinteistöissä on useampi ilmanvaihtokone, joista yksi voi hoitaa asuntojen ilmanvaihdon ja toinen yleisten tilojen ilmanvaihdon. Ilmanvaihtokoneessa on yleensä kaksi puhallinta, tulo- ja poistopuhallin.

Jokaista nykyaikaista ilmanvaihtokonetta ohjataan automaatiojärjestelmän avulla. Puhaltimet ohjautuvat tavallisimmin kanavaan asennetun paineanturin lukeman ja automaatiojärjestelmään asetetun paineen asetusarvon mukaisesti. Asetusarvo seuraa aktiivisesti paineen mittausta kanavasta ja säätää puhaltimen kierroslukua painemittauksen ja -asetusarvon mukaisesti. Ilmanvaihtokoneessa on ainakin kaksi suodatinta, joista yksi sijaitsee raitisilmassa ja toinen poistoilmassa. Suodattimet poistavat ilman epäpuhtauksia ja näin estävät ilmanvaihtokoneen ja kanavan likaantumisen sekä ilmanlaadun parantamisen. Jokainen ilmanvaihtokone on voitava pysäyttää hätätilanteessa. Tämä voidaan toteuttaa joko releillä tai ohjelmallisesti automaatiojärjestelmästä.

Ilmanvaihtokone huolehtii ilmanlaadun lisäksi myös ilman lämmittämisestä. Etenkin talvella tämä on erityisen tärkeää, sillä pelkästä lämmöntalteenotosta saatava lämpö ei yleensä riitä lämmittämään tuloilmaa halutulle tasolle – varsinkaan kovilla pakkasilla. Tästä syystä ilmanvaihtokoneisiin asennetaan jälkilämmityspatteri tuloilmaan lämmöntalteenoton jälkeen, jonka avulla tuloilman lämpötila saadaan sopivaksi talvellakin. Jälkilämmityspatteri voi olla joko sähköinen tai nesteen avulla lämmitettävä. Jälkilämmityspatterin ollessa nestekäyttöinen, pitää patterin paluulinjastoon asentaa jäätymisvaara-anturi jäätyksen estämiseksi. Patteri jäätyy helposti, mikäli lämmönjaossa tapahtuu häiriöitä. Jäätymisvaara-anturi kytketään jäätymisvaaratermostaattiin, joka hoitaa koneen pysäyttämisen katkaisemalla siitä sähköt ja näin estäen patterin jäätyksen.

Ilmanvaihtokoneeseen asennetaan useita lämpötila-antureita sekä painemittauksia seuraamaan ilmanvaihtokoneen tilaa. Anturit liitetään automaatiojärjestelmään, johon ohjelmoidaan fiktiivisiä hälytyspisteitä antureiden lähettämien tietojen perusteella. Esimerkiksi jos tuloilman lämpötila laskee alle asetetun raja-arvon, aktivoituu automaatiojärjestelmässä hälytys, joka tulostetaan useimmiten järjestelmän grafiikkakuvulle samaan paikkaan missä ilmanvaihtokoneen prosessikaaviokin sijaitsee. Hälytys voidaan tarvittaessa lähettää myös eteenpäin huoltoyhtiöön.

Älykkään automaatiojärjestelmän avulla ilmanvaihtokoneen tilaa voidaan seurata myös etäyhteyden välityksellä, josta nähdään reaaliaikaiset mittaustulokset ja koneen käyntitilat. Konetta voidaan myös tehostaa tai se voidaan pysäyttää etäyhteyden välityksellä.

#### 4.2 Ilmamääräsäädin

Ilmamääräsäätimien avulla voidaan säätää tarkasti tietyn tilan ilmanvaihtoa tietyissä tilanteissa. Esimerkiksi juhlien aikana ilmanvaihtoa tarvitaan enemmän, joten ilmamääräsäätimelle voidaan tällöin antaa asetusravon avulla tehostuskäsky, kun taas tilan ollessa vähemmällä käytöllä ilmamääräsäätimet voidaan asettaa minimi asetusravon mukaiselle teholle. Ilmamääräsäädintä (kuva 16) voidaan ohjata automaatiojärjestelmän avulla tavallisimmin joko analogiaviestillä 0–10 V, 2–10 V tai väylätekniikan avulla, esimerkiksi Modbusilla.



Kuva 16. Lindab Ultralink -ilmavirtasäädin. [23.]

Ultraäänitekniikka edustaa uutta tyyliä ilmavirtasäätimissä. Se mahdollistaa erittäin tarkan ilmavirtamittauksen ja -säädön ilman likaantuvia mittayhteitä. Mittaus tapahtuu yleensä suoraan kanavan pinnasta, joten kanaviston likaantuminen ei vaikuta

mittauksen tai säädön tarkkuuteen. Vanhemman mallisissa ilmamääräsäätimissä perinteiset mittayhteet aiheuttavat painehäviötä sekä likaantuvat helposti, mikä näkyy painesuhteiden muutoksina aiheuttaen muun muassa säädön epätarkkuutta. [23.]

## 5 Käytännön työosuus

Käytännön työosuus sisälsi tiedonhakuja, ohjelmointia, grafiikkakuvan piirtämistä ja rajapinnan toimivuuden testaamista sekä ohjelmakirjaston julkaisemisen. Ohjelmakirjasto julkaistiin yrityksen verkkosivuille, josta projektipäälliköt voivat sen hakea omiin projekteihinsa käytettäväksi.

Liitettäessä Modbus-laitetta rakennusautomaatiojärjestelmään pitää rajapinnan ohjelmoijan tietää, missä rekisterissä mikäkin tieto sijaitsee liitettävällä laitteella. Laitteen valmistajalta saadaan ohjelmointia varten rekisterilista, josta selviävät kaikki laitteen väylää pitkin saatavat tiedot. Rekisterilistaa muutetaan usein, joten on tärkeää hakea itse tai pyytää valmistajalta uusinta rekisterilistaa ohjelmointia varten.

Rekisterilistalta löytyy monenlaista tietoa, mikä ei välttämättä ole tärkeää juuri omassa projektissa. Ohjelmoija voi itse päättää listalta tärkeimmät tiedot, joita projektissa tarvitaan. Tässä työssä rekisterilistalta ohjelmoitiin lähes kaikki saatavilla olevat tiedot, sillä ohjelmakirjaston tarkoitus on olla mahdollisimman laaja, ainakin tässä tapauksessa.

Tämän työn Modbus-rajapinta testattiin oikealla laitteella, jonka laitevalmistaja lähetti rajapinnan tekemistä ja testaamista varten. Tämänlaisissa tapauksissa rajapinnan onnistumiselle on huomattavasti paremmat edellytykset, kuin tapauksissa, joissa rajapinnan tekeminen tapahtuu pelkän rekisterilistan avulla. Oikealla laitteella testattu rajapinta on varmasti toimiva, ja sitä voidaan käyttää helposti tiukkojenkin aikataulujen projekteissa.

### 5.1 Vaatimusmäärittelyt

Tavoitteena oli tehdä toimiva ohjelmakirjasto ja rajapinta Fidelix-järjestelmän alakeskukseen sekä Lindabin UltraLink-ilmamääräsäätimen välille. Lisäksi luotiin yleismallinen

grafiikkakuva ja väyläpisteet, jotta liitynnän tekeminen jatkossa eri kohteisiin olisi mahdollisimman vaivatonta.

Seuraavassa tarkennetaan rekistereitä, jotka piti ohjelmoida rajapintaan. Alussa oleva tekstijono tarkoittaa rekisterin tyyppiä ja osoitetta, esimerkiksi tulorekisteri 3x013 on rekisterissä 13. Suluissa olevat lukuarvot kertovat selkokielitekstinä, mitä laite tekee, jos tietyssä rekisterissä on tietty arvo. Näiden tietojen avulla ohjelmoija voi tehdä automaatiojärjestelmään valmiit tilatekstit jokaiselle arvolle, esimerkiksi jos edellä oleva tulorekisteri 3x013 olisi arvossa 3, laitteessa olisi jokin vika, ja se voitaisiin ilmaista automaatiojärjestelmän grafiikkakuvassa tekstinä "Vika", jotta automaatiojärjestelmän käyttäjä tietäisi nopealla silmäyksellä mitä laite tekee ja onko siinä jokin vika.

#### Tulorekisterit (luettavat tiedot)

- 3x013, Laitteen tilatieto (0 = Normaali, 1 = Virtauksen paikantaminen, 2 = Ohituksen ohjaus, 3 = Virhe, 4 = Ohjaussilmukan säätö, 5 = Kulma-anturin kalibrointi)
- 3x154, Ilmavirta (l/s)
- 3x200, Lämpötilatieto (°C)
- 3x251, Kertoo säätöpellin aukioloprosentin
- 3x252, Kertoo säätöpellin moottorin toiminnasta (0 = Moottori pysäytetty, 1 = Moottorin avautuminen, 2 = Moottorin sulkeutuminen)

#### Hälytykset (luettavat tiedot)

- 3x400, Hälytyksen rekisteri (1 = Moottori ei toimi, 2 = Kulma-anturi ei toimi oikein, 3 = Virtauksen asetuspistettä ei saavutettu, 4 = Virtauksen mittausongelma, 5 = Säätöpelti säätää, 6 = Ei käytössä, 7 – 31 = Varattu tulevaa käyttöä varten, 32 = Tehtaan tiedot turmeltuneet)

#### Järjestelmän määrytykset (kirjoitettavat ja luettavat tiedot)

- 4x070, Määrittää, kuinka peltiä säädetään (0 = Säädin pois päältä, 1 = Säätää pellin asentoa, 2 = Säätää virtausta)
- 4x071, Säätöpellin ohjaustavan määrittäminen (0 = Modbus tai Pascal, 1 = Analogiatulo)
- 4x082, Tehdasasetusten palautus (0 = Ei tee mitään, 1 = Palauttaa tehdasasetukset)



- 4x083, Uudelleenkäynnistys (0 = Ei tee mitään, 1 = Käynnistää laitteen uudelleen)

Määrittysten ohitus (kirjoitettavat ja luettavat tiedot)

- 4x151, Säätopellin ohitustilan määrittys (0 = Normaali tila, 1 = Ohituksen ohjaus maksimi auki, 2 = Ohituksen ohjaus min. auki, 3 = Ohituksen ohjaus 100 % auki, 4 = Ohituksen ohjaus 100 % kiinni)

Säätopellin rekisterit (kirjoitettavat ja luettavat tiedot)

- 4x300, Asennon kalibrointi (0 = Ei tee mitään, 1 = Käynnistää asento-anturin uudelleenkalloinnin, 2 = Käynnistää uudelleenkalloinnin laitteen käynnistyksen yhteydessä)
- 4x302, Asennon asetuspiste % (Käytetään vain, mikäli 4x070 on asetuksessa 1)
- 4x314, Virtauksen asetuspiste l/s (Käytetään vain, mikäli 4x070 on asetuksessa 2) [24.]

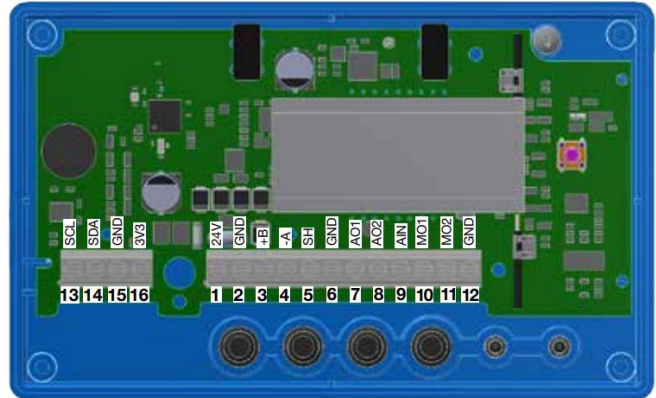
## 5.2 Työn suunnittelu ja laitteen käyttöönotto

Työ alkoi rekisterilistaan ja ilmamääräsäätimen tuotelehteen perehtymisellä. Jännite- sekä väyläkytkennät sijaitsevat ilmamääräsäätimen kotelon sisällä. Liittimiin 1 (24 V) sekä 2 (GND) kytketään laitteen käyttöjännite. Modbus-väyläliityntä kytketään liittimiin 3 (+B) sekä 4 (-A) (kuva 17).

## Connections

Connect the Controller to a remote terminal unit using RS485 or analog terminals. Connections are made in the terminal board which can be accessed when the lid of the display unit is removed. In the back of the lid there is a picture with a list of the terminals.

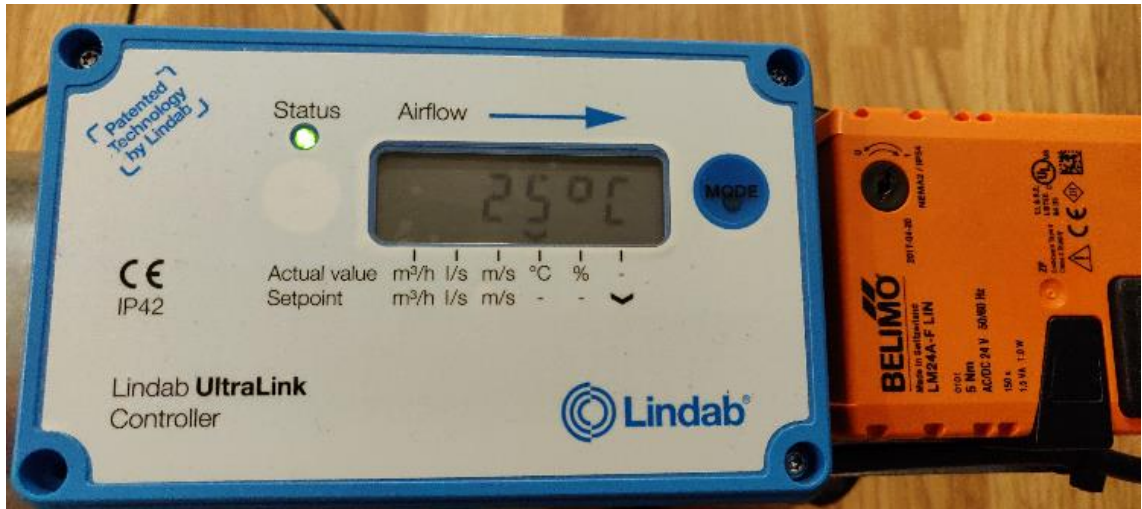
1. **24V**, power supply (AC G, DC +) \*
2. **GND**, power supply (AC G0, DC -) \*
3. **+B**, connection for Modbus via RS485
4. **-A**, connection for Modbus via RS485
5. **SH**, shield
6. **GND**, ground (system neutral)
7. **AO1**, analog output
8. **AO2**, analog output
9. **AIN**, analog input
10. **MO1**, connection for motor
11. **MO2**, connection for motor
12. **GND**, ground (system neutral)
13. **SCL**, not used
14. **SDA**, not used
15. **GND**, ground (system neutral)
16. **3V3**, not used (in case of biasing)



\*) When using AC terminal 1 (G) should have system potential and terminal 2 (G0) should be system neutral.

Kuva 17. Lindab Ultralink -kytkennät. [24.]

Laitteen käyttöjännitteen ja väyläkytkentöjen jälkeen (kuva 18) laite käynnistettiin ensimmäistä kertaa. Näytössä näkyy yksikön lämpötila.



Kuva 18. Laite käynnissä.

Seuraavaksi piti selvittää ilmamääräsäätimen väyläparametrit eli käytettävä protokolla, baudinopeus, pysäytysbitit sekä pariteetti. Ilmamääräsäätimessä on vain yksi painike, jonka avulla navigoidaan valikoissa. Määritysvalikkoon pääsee painamalla näytön

tilapainiketta vähintään viisi sekuntia pohjassa, kunnes esiin tulee valikko, jossa on kolme eri vaihtoehtoa:

- Con.Set (Tiedonsiirtoasetukset)
- AIn.Set (Analogisen ohjaustavan asetukset)
- Cancel (Peruutus ja palaaminen infonäyttöön) [24.]

Valittiin Con.Set, ja painettiin tilapainiketta pitkään pohjassa, jotta päästiin tiedonsiirtoasetuksiin. Käyttöliittymään ilmestyi teksti Pr, joka tarkoittaa protokollaa. Valittiin vaihtoehto Pr.Mod painamalla painiketta kerran lyhyesti. Pr.Mod tarkoittaa Modbus-protokollan valitsemista. Seuraavaan valikkoon päästiin painamalla tilapainiketta pitkään pohjassa. Valikkoon ilmestyi merkki b, joka tarkoittaa baudinopeutta. Vaihtoehtoista valittiin 19 200. Siirryttiin valikossa eteenpäin ja vaihdoimme bit (pysäytysbitit) kohdan arvoksi 1, eli yhden pysäytysbitin. Viimeisenä väyläasetuksena aseteltiin kohdan P eli pariteetin arvoon Odd (pariton pariteetti). Modbus-osoite löytyi valikon kohdasta Id. Tämä on tärkeä tieto väylän käyttöönottoa varten. Ilmamääräsäätimen ohjaustavan voi valita joko samasta valikosta tai Modbus-väylän kautta. Mikäli haluaa ohjata ilmamääräsäädintä väylän kautta, pitää valikon kohdasta Cnt. (ohjaustapa) valita vaihtoehto Cnt.bus. Mikäli haluaa lukea laitteelta pelkkiä tietoja ja ohjata laitetta analogialähdöllä, pitää valita samasta valikosta kohta Cnt.AIn. Tässä tapauksessa myös laitteen liittimeen 9 (AIN) pitää kytkeä analogialähtö (0–10 V). Laitteen väyläasetusten valikkorakenne löytyy kuvasta 19.

Menu tag	Description	Options	Description
• Pr.	Protocol	Pr.PAS Pr.Mod	Pascal protocol Modbus
• b.	Baud rate	b.9600 b.19200 b.38400 b.76800	Baud rate 9600 Baud rate 19200 Baud rate 38400 Baud rate 76800
• bit.	Stop bits	bit.1 bit.2	1 stop bits 2 stop bits
• P.	Parity	P.odd P.even P.none	Odd parity Even parity Parity none
• Id.	Modbus Id	Id.x	Modbus id (x = value) *)
• PLA.	PLA address for Pascal	PLA.x	PLA address (x = value) *)
• ELA.	ELA address for Pascal	ELA.x	ELA address (x = value) *)
• Pi.	Pin-code	Pi.xxxx	Default: xxxx = 1111
• Cnt.	Control by bus	Cnt.bus Cnt.Aln	Control by bus Control by analog in
• Store	Store changes		Stores changes on long press
• Cancel	Cancel		Cancel and ignore changes on long press

Kuva 19. Väyläasetusten valikkorakenne. [24.]

Laitteen Modbus-tietoliikenteen rekistereiden testaamiseen käytettiin apuna Modbus Poll -tietokoneohjelmaa. Ohjelmaan saa kolmenkymmenen päivän ilmaisen kokeilujakson, jonka jälkeen ohjelman käyttö on maksullista. Modbus Poll on nopea ja helppo tapa testata laitteen Modbus-rajapinta ilman työläitä järjestelmäkonfiguraatioita. Tässä työssä käytetty ohjelmaversio oli 4.3.4.

Modbus Poll liitettiin ilmamääräsäätimeen USB-sarjaporttimuuntimen avulla. Sarjaporttimuuntimen toisessa päässä oli kaksi johdinta, jotka kytkettiin suoraan ilmamääräsäätimen väyläliittimiin. Toinen puoli muuntimesta kytkettiin tietokoneen USB-porttiin.

Tietokoneen laitehallinnasta valittiin tietokoneen valitsema portti (tässä tapauksessa portti numero 9), josta muutettiin väyläasetukset (kts. ylempää) samanlaisiksi kuin ilmamääräsäätimessä. Tämän jälkeen avattiin Modbus Poll, valittiin Connection ja sen jälkeen Connect. Ruutuun ilmestyi Connection Setup -valikko, josta valittiin samat väyläasetukset kuin ilmamääräsäätimessä (kuva 20). Modeksi valittiin RTU ja Connection-typiksi Serial Port.

Connection Setup

Connection  
 Serial Port  TCP/IP

Port 9  
19200 Baud  
8 Data bits  
Odd Parity  
1 Stop Bit

Mode  
 RTU  ASCII

Response Timeout  
1000 [ms]

Delay Between Polls  
10 [ms]

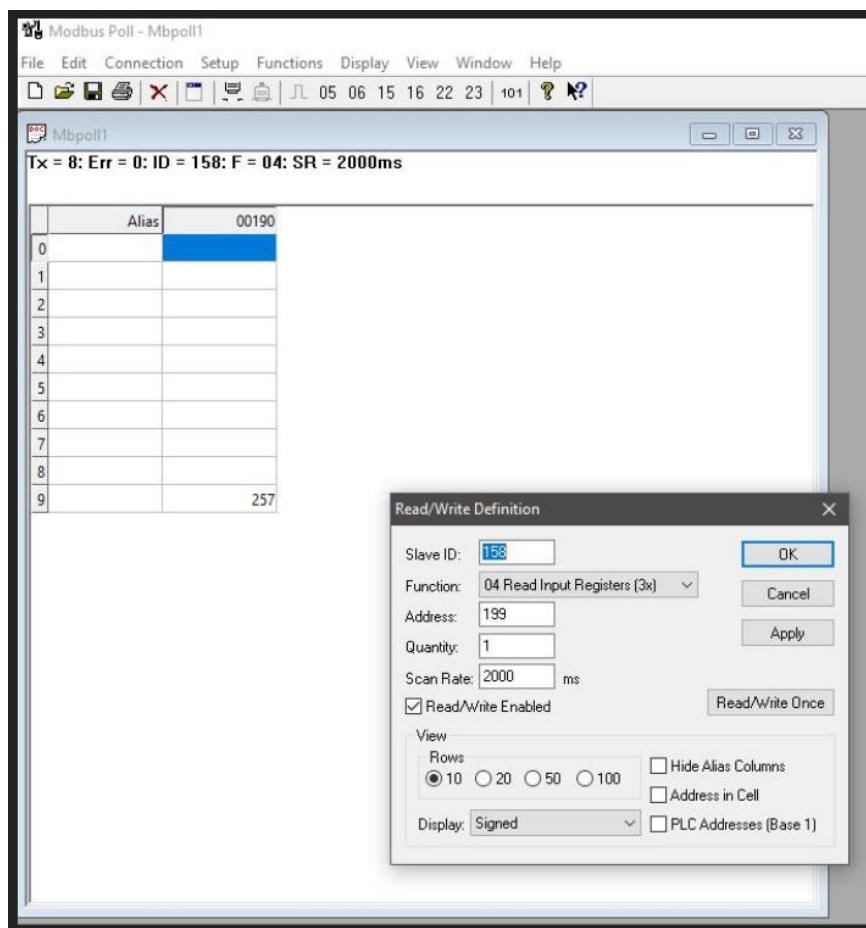
Advanced...

Remote Server  
IP Address: 0.0.0.0  
Port: 502  
Connect Timeout: 3000 [ms]

OK  
Cancel

Kuva 20. Connection Setup -valikko.

Painamalla OK saatiin ohjelman käyttöliittymään näkymä, joka kertoo, kommunikoidko laite ohjelman kanssa. Vielä tässä vaiheessa laitteen ei pitäisi kommunikoida ohjelman kanssa, vaan käyttöliittymän pitäisi varoittaa kommunikaatiovirheestä. Porttiasetusten jälkeen pitää kysellä haluttua rekisteriä eli esimerkiksi yksikön lämpötilaa ja löytää rekisterilistasta sen rekisteri ja syöttää se ohjelmaan. Tähän valikkoon päästään kohdasta Setup ja painamalla linkistä Read/Write Definition. Käyttöliittymään avautuu Read/Write Definition -valikko, josta valitaan haluttu rekisterialue tai vain yksittäinen rekisteri, laitteen osoite ja rekisterin tyyppi (kuva 21).



Kuva 21. Read/Write Definition -valikko.

Kohtaan Slave ID laitettiin yksikön osoite. Function-kohtaan valittiin rekisterin tyyppi, joka oli tässä tapauksessa rekisterilistan mukaisesti Read Input Registers, sillä tarkoitus oli lukea yksikön lämpötilatieto. Kohtaan Address syötettiin rekisterin aloitusosoite, joka näkyi rekisterilistassa. Rekisterilistassa lämpötilan osoitetieto on 200, mutta ohjelmasta pystyy valitsemaan, alkaako osoitejärjestelmä osoitteesta 0 vai osoitteesta 1. Tämän pystyy vaihtamaan Read/Write Definition -valikon oikeasta alareunasta. Quantity tarkoittaa rekisterien määrää, eli miltä rekisterialueelta halutaan lukea/kirjoittaa. Tässä tapauksessa valittiin luku 1, sillä rekisterilistan mukaan lämpötilatieto näkyy vain yhdessä rekisterissä.

Kuvan 21 yläreunassa näkyy kohta Tx = 8, joka kertoo, että yhteensä kahdeksan sanaa on mennyt onnistuneesti perille. Sarakkeen 9 kohdassa, jossa lukee arvo 257, on yksikön lämpötilatieto. Tämä tieto pitää jakaa 10:llä (kts. kuva 22), jotta oikea luku

saadaan selville. Yksikön oikea lämpötila on siis 25,7°C, joka saatiin onnistuneesti luet-  
tua ohjelman avulla.

Address	UltraLink®		Name	Description	Data type	Unit	Div	Default	Min	Max	Access
	Controller	Monitor									
<b>INPUT REGISTERS</b>											
3x008	X	X	Product Nominal Size	Nominal diameter of duct	16bit	mm					RO
3x013	X	X	Unit Status	Current unit status: 0 = Normal mode; 1 = Locating flow; 2 = Override control; 3 = Error; 4 = Control loop regulating; 5 = Angle sensor calibrating	16bit						RO
<b>Flow info</b>											
3x150	X	X	Velocity in m/s	Velocity in m/s	Float	m/s					RO
3x152	X	X	Air flow in m³/h	Air flow in m³/h	Float	m³/h					RO
3x154	X	X	Air flow in l/s	Air flow in l/s	Float	l/s					RO
<b>Temperature info</b>											
3x200	X	X	Current temperature in °C	Temperature in degree celcius.	16bit	°C	10				RO

Kuva 22. Ote ilmamääräsäätimen rekisterilistasta, jossa alimmaisena yksikön lämpötilatieto. [24.]

Laitteen muut tärkeät rekisterit testattiin samalla tavalla, ja ne todettiin toimiviksi ennen automaatiojärjestelmän ohjelmointityön aloittamista. Liitteessä 1 on esitelty kaikki lait-  
teelta saatavilla olevat rekisterit.

### 5.3 Human-Machine Interface

Grafiikkakuva (myös HMI) tarkoittaa ihmisen ja automaatiojärjestelmän välisen kommu-  
nikaation mahdollistavaa ohjelmistoa. Työstä luotiin valmis grafiikkakuva, jotta pisteiden  
ja symbolien kopioiminen omaan projektiin olisi mahdollisimman vaivatonta. Grafiikka-  
kuvaan liitettiin kaikki vaatimusmäärittelyissä mainitut rekisterit.

Grafiikkakuvan piirtäminen tapahtui Fidelix:n FX-Editor:ssa (versio 1.7.0) olevan grafiik-  
katyökalun avulla. Grafiikkatyökaluun pääsee FX-Editor:n Graphics-välilehdeltä (kuva  
23). Tekstiä saa tehtyä kuvan vasemmassa reunassa olevan työkalupalkin Add static or  
dynamic text -linkistä. Symbolit tuodaan kuvaan saman työkalupalkin Add symbol -lin-  
kistä. Tässä työssä tekstiä käytettiin ilmaisemaan ilmamääräsäätimen eri tiloja ja ohjauk-  
sia. Symbolit luotiin ainoastaan hälytyksille.

**Fidelix** Esimerkkikohde VAK  
Esimerkkitie IMS  
Esimerkkikaupunki

Selite	Luku väylältä	Selite	Luku & kirjoitus
Virtausnopeus	???	Virtauksen asetusarvo	???
Ilmavirta m3/h	???	Virtauksen asetusarvon minimi	???
Ilmavirta l/s	???	Virtauksen asetusarvon maksimi	???
Tila	Normaali	Säätötyyppi	Modbus
Moottorin tila	Moottori seis	Säätötapa	Säätö kytketty pois päältä
Yksikön lämpötila	???	Ohjauksen ohitus	Normaali ohjaus
Pellin asento asteina	???	Kulma-anturin kalibrointi	Ei aktiivinen
Pellin aukioalo prosentteina	???	RS485-protokolla	Modbus
Nimelliskoko	???	Modbusosoite	???
Moottori ei toimi		Väylänopeus	9600
Kulma-anturi ei toimi oikein		Pariteetti	Odd
Virtauksen asetuspistettä ei saavutettu		Stop-biitti	1
Virtauksen mittaongelma		Uudelleenkäynnistys	Ei aktiivinen
Säätöpelti säätää			
Modbus-kommunikaatiohälytys			

grafiiikkakuva.htm

X: 1036 Y: 354 Selected: 1 Selected X/Y: 210/132 Zoom: 200% Snap: none Symbol path: C:\ProgramData\FX-Editor\Symbol\Default\

Event log  
Elapsed time reading included files 127ms  
Project ready

Offline CommTask: 0/0 Pointcount: 38 Selected points: 1 PLC-IP: 10.10.14.193 Local-IP: 192.168.10.41

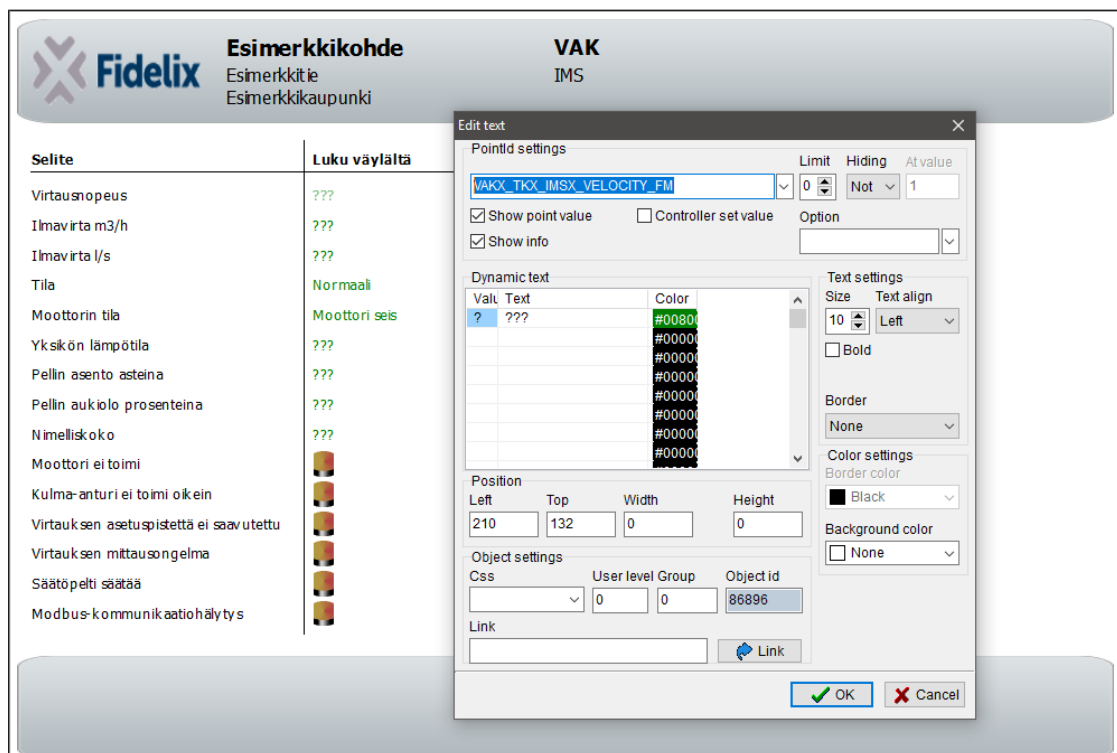
Kuva 23. Valmis grafiikkakuva piirrettynä FX-Editorin grafiikkatyökalulla.

#### 5.4 Pisteiden luominen

Grafiikkakuvaan lisättyihin teksteihin ja hälytyssymboleihin luotiin pistetunnukset. Pistetunnus pitää määrittellä jokaiselle fyysiselle tai fiktiiviselle pisteelle, sillä sitä käytetään pisteen tyyppin määrittelemiseen, kytkentäkuvien tekemiseen ja ohjelmointiin. Esimerkiksi huoneen lämpötilan pistetunnus voisi olla VAK1\_TKPK301\_TE20\_M, jonka alkuosasta saadaan valvonta-alakeskuksen tunnus, toisesta osasta ilmanvaihtokoneen tunnus ja kolmannesta osasta huoneanturin tunnus. Loppupääte \_M määrittelee, minkä tyyppinen piste on (mittaus, säätö, ohjaus, indikointi, hälytys tai pulssi).

Pistetunnus luodaan painamalla grafiikkatyökalussa symbolia tai tekstiä kaksi kertaa hiiren vasemmalla näppäimellä. Pistetunnus kirjoitetaan kuvassa 24 näkyvään laatikkoon, jossa pistetunnus on maalattuna.





Kuva 24. Pistetunnuksen luominen.

Pistetunnuksen luomisen jälkeen painetaan OK-painiketta, jonka jälkeen pitää valita välilehti Fidelix FX-Editorin näkymästä. Valitsemalla kansion www ja painamalla sen jälkeen hiiren oikealla painikkeella vasta luotua grafiikkakuvaa ja valitsemalla avautuvasta ikkunasta Include file -linkin saadaan pisteet muokattavaksi ohjelmaan. Points-välilehdellä (kuva 25) näkyvät kaikki pisteet – sekä työkaluun tuodut että tuomattomat. Työkaluun tuodut pisteet näkyvät kuvan vasemmassa reunassa, listalla Pointlist, ja tuomattomat pisteet kuvan oikeassa reunassa, listalla Uncleared points. Pisteet pitää ”suodattaa” ennen niiden tuomista ohjelmaan. Tämä tarkoittaa sitä, että suodatin antaa pistetunnukselle jo valmiiksi tietyt parametrit, kuten pisteen selkokielitekstin, yksikön, tilatekstin ja paljon muuta. Tässä työssä tehtiin niin vähän pisteitä, että sitä varten ei luotu omaa suodatinta, vaan kaikki pisteet konfiguroitiin yksitellen.

Pointname	Text	Type	I/O	Changed
VAKO_TIO_IMSX_STATUS_FI	Ilmämääräsäädin, Tila, FiktivinenIndkohti	Indication	00.0000:00	3.9.2018 18.45.10
VAKO_TIO_IMSX_FH	Ilmämääräsäädin, FiktivinenHälytys	Alarm	00.0000:00	3.9.2018 18.45.28
VAKO_TIO_IMSX_PARITY_FI	Ilmämääräsäädin, Pariteetti, FiktivinenIndkohti	Indication	00.0000:00	3.9.2018 18.45.19
VAKO_TIO_IMSX_STOPBIT_FI	Ilmämääräsäädin, Stop-bitit, FiktivinenIndkohti	Indication	00.0000:00	3.9.2018 18.45.24
VAKO_TIO_IMSX_PROTOCOL_FI	Ilmämääräsäädin, Protokolla, FiktivinenIndkohti	Indication	00.0000:00	3.9.2018 18.45.27
VAKO_TIO_IMSX_REGULATION_FI	Ilmämääräsäädin, Säätötapa, FiktivinenIndkohti	Indication	00.0000:00	3.9.2018 18.45.31
VAKO_TIO_IMSX_OVERRIDE_FI	Ilmämääräsäädin, Ohjauksen ohitus, FiktivinenIndkohti	Indication	00.0000:00	3.9.2018 18.45.50

Kuva 25. Tuodut ja tuomattomat pisteet.

Suodatin valitaan Uncleared points -listan yläpuolella olevasta pudotusvalikosta, jossa on jo valmiiksi luotuja suodattimia. Suodattimen valinnan jälkeen painetaan pistetunnusta hiiren oikealla painikkeella, jonka jälkeen valitaan avautuvasta ikkunasta kohta Import Points. Tämän jälkeen piste siirtyy vasemman puoliselle listalle, josta sitä voidaan alkaa työstämään. Pisteen konfigurointi aloitetaan painamalla pistetunnusta vasemman puolisella listalla kaksi kertaa. Ohjelmaan avautuu ikkuna (kuva 26), josta pistettä voidaan muokata.

Alarm point properties

OK Apply Cancel

**Point**

Pointname: VAKO\_TIO\_IMSX\_FH View: 0 Manual: 0 Progr.: 0

Text: Ilmämääräsäädin, FiktivinenHälytys

Picture:  Auto  History point

**Point link**

Port.Module: Point:

Global point

**FX-Editor properties**

Include point to label list  Physical point  Multi24

**Alarm group**

Alarm group: B-LUOKKA Priority: 0 Alerta: 0

Name of overriding point:  Override to off state: 0

Help of alarm: From group

Save to log

**State Text**

State Text: HÄLYTYS

**Alarm input settings**

Start delay (sec): 0 Normally

Stop delay (sec): 0

**Alarm settings**

Mode: Normal

1 point is selected for editing

Kuva 26. Hälytyspisteen muokkaaminen.

Hälytyspisteelle luodaan hälytysluokka, joka kertoo hälytyksen kiireellisyyden ja sen, lähetetäänkö hälytys esimerkiksi huoltoyhtiön puhelimeen tekstiviestinä. Lisäksi valitaan tilateksti, joka kertoo hälytyksen tilan selkokielitekstinä. Hälytys voi olla esimerkiksi tilassa ”Normaali” tai ”Hälytys”. Jokaiselle pisteelle ja pistetyypille luodaan omat ominaiset asetuksensa, joiden avulla käyttäjä tietää laitteesta tai järjestelmästä mahdollisimman paljon ja joiden avulla käyttöliittymästä tehdään mahdollisimman selkeä.

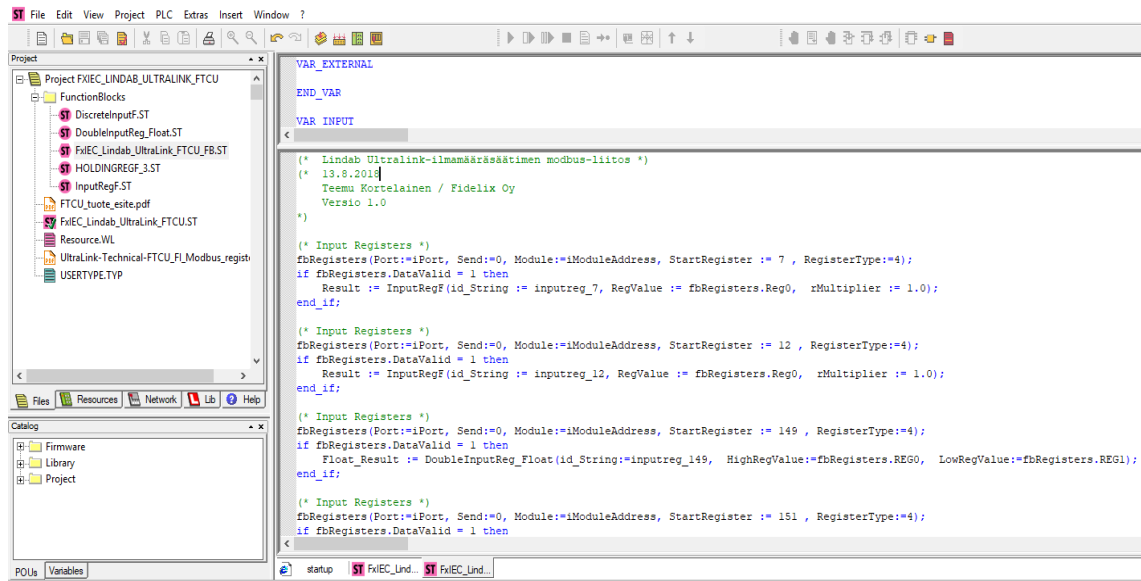
Tämän työn pistetunnusten runko tehtiin VAKX\_TKX\_IMS-tyyppiseksi (kuva 27), jotta pistetunnusten tekstin korvaaminen olisi helpompaa ja nopeampaa. Projektipäälliköt voivat korvata x-kirjaimen omaan projektiinsa sopivaksi tekstiksi ja/tai numeroksi.

Pointname	Text	Type	I/D	Changed	Pointst
VAKX_TKX_IMS_STATUS_FI	Ilmääränsäädin, Tila, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:10	
VAKX_TKX_IMS_FH	Ilmääränsäädin, FiktiivinenHälytys	Alarm	00,000:00	3.9.2018 18:48:28	
VAKX_TKX_IMS_BAUD_FI	Ilmääränsäädin, Väylänopeus, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:14	
VAKX_TKX_IMS_PARITY_FI	Ilmääränsäädin, Pariteetti, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:19	
VAKX_TKX_IMS_STOPBIT_FI	Ilmääränsäädin, Stop-bitit, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:24	
VAKX_TKX_IMS_PROTOCOL_FI	Ilmääränsäädin, Protokolla, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:27	
VAKX_TKX_IMS_REGULATION_FI	Ilmääränsäädin, Säätötapa, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:31	
VAKX_TKX_IMS_REBOOT_FI	Ilmääränsäädin, Uudelleenkäynnistys, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:37	
VAKX_TKX_IMS_OVERRIDE_FI	Ilmääränsäädin, Ohjauksen ohitus, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:45:50	
VAKX_TKX_IMS_CALB_FI	Ilmääränsäädin, Kuluva-anturi kalibrointi, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:46:26	
VAKX_TKX_IMS_MOCBUS_FH	Ilmääränsäädin, Modbus-kommunikaatio, FiktiivinenHälytys	Alarm	00,000:00	3.9.2018 18:46:43	
VAKX_TKX_IMS_MOTOR_FI	Ilmääränsäädin, Moottorin tila, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:47:05	
VAKX_TKX_IMS_FI	Ilmääränsäädin, Hälytys, FiktiivinenIndikointi	Indication	00,000:00	3.9.2018 18:48:15	
VAKX_TKX_IMS_SIZE_FM	Ilmääränsäädin, Nimelliskoko, FiktiivinenMittaus	Analog in	00,000:00	4.9.2018 19:00:28	
VAKX_TKX_IMS_TEMP_FM	Ilmääränsäädin, Lämpötila, FiktiivinenMittaus	Analog in	00,000:00	3.9.2018 18:51:58	
VAKX_TKX_IMS_POS_FM	Ilmääränsäädin, Pellin asento astena, FiktiivinenMittaus	Analog in	00,000:00	3.9.2018 18:51:48	
VAKX_TKX_IMS_PERCEN_FM	Ilmääränsäädin, Pellin asento prosentteina, FiktiivinenMittaus	Analog in	00,000:00	3.9.2018 18:51:43	
VAKX_TKX_IMS_MOCBUS_FM	Ilmääränsäädin, Modbus-osotte, FiktiivinenMittaus	Analog in	00,000:00	3.9.2018 18:51:39	
VAKX_TKX_IMS_FLOW_SET_FM	Ilmääränsäädin, Virtauksen asetusarvo, FiktiivinenMittaus	Analog in	00,000:00	7.9.2018 15:18:30	
VAKX_TKX_IMS_FLOW_SET_MIN_FM	Ilmääränsäädin, Virtauksen asetusarvon minimi, FiktiivinenMittaus	Analog in	00,000:00	3.9.2018 18:53:33	
VAKX_TKX_IMS_FLOW_SET_MAX_FM	Ilmääränsäädin, Virtauksen asetusarvon maksimi, FiktiivinenMittaus	Analog in	00,000:00	3.9.2018 18:53:29	
VAKX_TKX_IMS_VELOCITY_FM	Ilmääränsäädin, Virtausnopeus, FiktiivinenMittaus	Analog in	00,000:00	7.9.2018 16:07:20	
VAKX_TKX_IMS_AIRFLOW_MID_FM	Ilmääränsäädin, Ilmavirta m3/s, FiktiivinenMittaus	Analog in	00,000:00	7.9.2018 16:07:41	
VAKX_TKX_IMS_AIRFLOW_LS_FM	Ilmääränsäädin, Ilmavirta l/s, FiktiivinenMittaus	Analog in	00,000:00	7.9.2018 16:07:51	
VAKX_TKX_IMS_INPUT_FO	Ilmääränsäädin, Säätötyyppi, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:35:58	
VAKX_TKX_IMS_REGULATION_FO	Ilmääränsäädin, Säätötapa, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:36:59	
VAKX_TKX_IMS_OVERRIDE_FO	Ilmääränsäädin, Ohjauksen ohitus, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:36:10	
VAKX_TKX_IMS_CALB_FO	Ilmääränsäädin, Kuluva-anturi kalibrointi, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:36:15	
VAKX_TKX_IMS_PROTOCOL_FO	Ilmääränsäädin, Protokolla, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:36:21	
VAKX_TKX_IMS_BAUD_FO	Ilmääränsäädin, Väylänopeus, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:36:31	
VAKX_TKX_IMS_PARITY_FO	Ilmääränsäädin, Pariteetti, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:36:36	
VAKX_TKX_IMS_STOPBIT_FO	Ilmääränsäädin, Stop-bitit, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:40:15	
VAKX_TKX_IMS_REBOOT_FO	Ilmääränsäädin, Uudelleenkäynnistys, FiktiivinenOhjaus	Digital out	00,000:00	3.9.2018 19:36:56	
VAKX_TKX_IMS_MOTOR_ALARM_FH	Ilmääränsäädin, Moottori ei toimi, FiktiivinenHälytys	Alarm	00,000:00	6.9.2018 19:15:21	
VAKX_TKX_IMS_SENSOR_FH	Ilmääränsäädin, Kuluva-anturi ei toimi oikein, FiktiivinenHälytys	Alarm	00,000:00	6.9.2018 19:15:21	
VAKX_TKX_IMS_FLOW_SET_FH	Ilmääränsäädin, Virtauksen asetusarvo ei saavutettu, FiktiivinenHälytys	Alarm	00,000:00	6.9.2018 19:15:21	
VAKX_TKX_IMS_FLOW_PROB_FH	Ilmääränsäädin, Virtauksen mittausongelma, FiktiivinenHälytys	Alarm	00,000:00	6.9.2018 19:15:21	
VAKX_TKX_IMS_DAMPER_FH	Ilmääränsäädin, Säätöpelin säätö, FiktiivinenHälytys	Alarm	00,000:00	6.9.2018 19:15:21	

Kuva 27. Valmis pistelista.

## 5.5 Ohjelmointi

Ohjelmointi tehtiin OpenPCS-ohjelmalla, jonka versio tekohetkellä oli 6.7. Ohjelmoinnissa on välttämätöntä käyttää laitevalmistajalta saatua rekisterilistaa. Ohjelmointi alkaa luomalla uuden projektin ohjelman käyttöliittymästä (File->New->Projects->Empty project). Ohjelman vasemmassa reunassa näkyy navigointipalkki, jossa on luotuna kaikki ohjelmat ja niiden kansiorakenne (kuva 28).



Kuva 28. Ohjelman käyttöliittymä.

Ohjelmat tehtiin ST-kielellä (Structured Text). Ohjelmakirjasto luotiin Function Blockiin (File->New->POU->Function Block). Ohjelmointi aloitettiin input-rekistereiden luomisesta. Virtausnopeuden Modbus-osoite on rekisterilistan mukaan 3x150. Tämä ohjelmoitiin koodin 1 mukaisesti.

```
(* Input Registers *)
```

```
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress,
StartRegister := 149, RegisterType:=4);
```

```
if fbRegisters.DataValid = 1 then
Float_Result := DoubleInputReg_Float(id_String:=inputreg_149,
HighRegValue:=fbRegisters.REG0, LowRegValue:=fbRegisters.REG1);
end_if;
```

Esimerkkikoodi 1. Virtausnopeuden rekisterin ohjelmointi.

Koodissa määritellään käytettävä fyysinen portti (iPort-muuttuja), Modbus-laitteen osoite sekä rekisterin tyyppi, joka koodin tapauksessa oli 4 eli input-rekisteri. Datavalid-komennolla tarkistetaan, onko ala-asema saanut luettua laitteelta kysytyn datan onnistuneesti. Rekisterilista määrittelee virtausnopeuden rekisterin datatyyppiä Float. Float-datatyyppi on kahdessa perättäisessä rekisterissä, ja se on 32-bittinen. Tästä syystä koodissa käytetään eräänlaista valmista ohjelmakirjastoa, DoubleInputReg\_Float-ohjelmakirjastoa, joka mahdollistaa tämän datatyyppin luennan oikein. HighRegValue- ja LowRegValue-muuttujiin tallennetaan rekisterin arvo. Molempiin tallennetaan yksi 16-bittinen arvo.

Ohjelmakirjasto muuntaa nämä kaksi 16-bittistä tietoa helposti luettavaan muotoon grafiikalle, esimerkiksi 200 l/s.

```
(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress,
StartRegister := 399 , RegisterType:=4);

if fbRegisters.DataValid = 1 then
    Result := DiscreteInputF(id_String := inputreg_399_1, RegValue :=
fbRegisters.Reg1, BitMask := 16#0001);
    Result := DiscreteInputF(id_String := inputreg_399_2, RegValue :=
fbRegisters.Reg1, BitMask := 16#0002);
    Result := DiscreteInputF(id_String := inputreg_399_3, RegValue :=
fbRegisters.Reg1, BitMask := 16#0004);
    Result := DiscreteInputF(id_String := inputreg_399_4, RegValue :=
fbRegisters.Reg1, BitMask := 16#0008);
    Result := DiscreteInputF(id_String := inputreg_399_5, RegValue :=
fbRegisters.Reg1, BitMask := 16#0010);
end_if;
```

Esimerkkikoodi 2. Hälytysrekistereiden ohjelmointi.

Hälytysrekisteri sijaitsee rekisterilistan mukaan osoitteessa 400, ja se on datatyypiltään 32-bittinen. Hälytykset on tallennettu biteittäin kuvan 29 mukaisesti.

Hälytykset										
3x400	X	X	X	Hälytyksen rekisteri 1	Hälytykset 1-32-bittiset: 1 = Moottori ei toimi. 2 = Kulma-anturi ei toimi oikein. 3 = Virtauksen asetuspiistettä ei saavutettu. 4 = Virtauksen mittausongelma. 5 = Säätöpelti säätää. 6 = Ei käytössä. 7 - 31 = Varattu tulevaa käyttöä varten. 32 = Tehtaan tiedot turmeiltuneet.	32-bittinen				RO
PITOREKISTERIT										

Kuva 29. Hälytykset rekisterilistalla.

Jokaisella hälytyksellä on oma bittinsä rekisterissä. Oikea hälytysbitti löytyi BitMaskin avulla. BitMask ”maskaa” yhden bitin koko rekisteristä, joka mahdollistaa hyvin yksityiskohtaisen rekisterin tarkastelun. Tämän tyyppisessä rekisterissä voi olla myös indikointitietoja, jotka kertovat laitteen tilasta.

Seuraavaksi ohjelmoitiin Holding-rekisterit. Holding-rekistereitä voidaan lukea ja kirjoittaa. Niihin voidaan tallentaa esimerkiksi asetusarvoja ja mittaustietoja. Koodin 3 mukaisesti ohjelmoitiin asetusarvon tallennus sekä minimi- ja maksimiarvot asetusarvolle.

```
(* Holding Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress,
StartRegister := 313 , RegisterType:=3);

if fbRegisters.DataValid = 1 then
fbRegisters.Reg0 := HoldingRegF_3(id_String := holdreg_313, RegValue :=
fbRegisters.Reg0, rMultiplier:=1.0, Force := holdreg_313_force,
ReleaseOverride := holdreg_313_releaseoverride);

fbRegisters.Reg1 := HoldingRegF_3(id_String := holdreg_314, RegValue :=
fbRegisters.Reg1, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

fbRegisters.Reg2 := HoldingRegF_3(id_String := holdreg_315, RegValue :=
fbRegisters.Reg2, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

fbRegisters (Send:=2);
end_if;
```

Esimerkkikoodi 3. Asetusarvo sekä minimi- ja maksimirajat asetusrvolle.

Rekisterin tyyppi määriteltiin numero 3, joka tarkoittaa Holding-rekistereitä. rMultiplier kertoo, millä arvolla väylältä luettu arvo jaetaan ennen pisteeseen kirjoittamista. Mikäli arvo myös kirjoitetaan rekisteriin, se kerrotaan samalla luvulla ennen rekisteriin kirjoittamista. Rekisterilistan mukaan arvoa ei kerrota ennen väylälle kirjoittamista, joten rMultiplier-arvo on tässä tapauksessa 1.0.

Force- ja ReleaseOverride-muuttujien arvot riippuvat toisistaan. Mikäli Force-muuttuja määritellään TRUEksi ja ReleaseOverride-muuttuja FALSEksi, ohjelma kirjoittaa aina pisteen mukaista arvoa väylälle, mutta piste voi olla myös käsiohjattuna. Esimerkiksi tämän työn ilmamääräsäätimelle voidaan tehdä IEC-ohjelma, joka tehostaa ilmamääräsäätimen asetusrvoa tietyssä tilanteessa, esimerkiksi tilan hiilidioksiditason nousussa, tai vaikka vain päivisin aikaohjelman mukaan. Mikäli Force määritellään FALSEksi ja ReleaseOverride TRUEksi, asetusrvoa muutetaan vain käsiohjauksen avulla. Kun piste ohjataan esimerkiksi arvosta 200 l/s arvoon 100 l/s, piste ohjautuu käsin arvoon 100 l/s ja vapautuu automaatille Modbus-viestin mennessä onnistuneesti perille.

Lopuksi tarkistetaan kommunikaation tila ja aktivoidaan hälytys, mikäli laite ei vastaa sanoiin (esimerkkikoodi 4).

```
(* Modulealarm *)
Result := SetDigitalPointF( Value:=BOOL_TO_INT(GetSystemStatusF( Mode:=1, iParameter:=iModuleAddress, rParameter:=INT_TO_REAL(iPort)) > 30.0), LockState:=1, Name:=id_ModuleAlarm );
```

Esimerkkikoodi 4. Kommunikaatiohälytyksen koodi.

Ohjelmakirjaston ollessa valmis luotiin kirjastolle ”kutsu” (liite 2, sivut 5–7), johon määriteltiin aiemmin tehdyt pistetunnukset. Kutsun avulla voidaan ohjelmoida monia laitteita saman ohjelmakirjaston taakse, mikä säästää huomattavan määrän vaivaa ja ohjelmointia.

Kutsuun ohjelmoitiin valmiiksi esimerkkipisteet, jotka voidaan korvata helposti ohjelmassa olevan korvaustoiminnon avulla. Muuttujat `inputreg_x`, jossa `x:n` kohdalla on rekisterilistan mukainen osoite, menee ohjelmakirjastoon käsiteltäväksi, jonka jälkeen ohjelmakirjasto tallentaa ”-merkkien välille asetettuun pistetunnukseen arvon, tai väylälle kirjoitettaessa pistetunnuksen arvo kirjoitetaan laitteelle.

Lopuksi ohjelma käännettiin ja ladattiin ala-asemalle.

## 5.6 Integraation testaus ja lopputulokset

Jokainen rekisteri testattiin erikseen Modbus Poll -ohjelmalla, ennen kuin ne ohjelmoitiin automaatiojärjestelmään. Tällä tavalla virheet löytyivät nopeasti, ja rekistereiden toimivuus oli varmistettu. Kun rekisterit oli testattu ja automaatiojärjestelmän ohjelmointi oli valmis, alkoi integraation todellinen testi, jossa kaikkia ohjelmoituja rekistereitä käytettiin samanaikaisesti.

Integraation todellinen toimivuus todettiin lataamalla pisteet, grafiikkakuva, modbus-laitteet sekä ohjelmat ala-asemaan. Ilmamääräsäätimen toiseen päähän asetettiin tuuletin, jonka tarkoitus oli simuloida ilmavirtaa kanavassa. Tämän avulla saatiin testattua säädinparametrit ja ilmamääräsäätimen todellinen toiminta.

Aluksi testattiin hälytykset. Kuvassa 30 moottori on vikatilassa. Teksti grafiikkakuvalla muuttui punaiseksi ja hälytyssymboli kertoo tarkemmat tiedot, eli tässä tapauksessa ”virtauksen asetuspistettä ei saavutettu”. Tämä voidaan todeta helposti oikeassa reunassa olevasta asetusarvosta, joka on sinisellä värillä, ja arvoltaan 15 l/s. Kuvan vasemmassa reunassa todellinen mittausarvo on vain 2,67 l/s, minkä takia hälytys onkin aktiivinen.

Fidelix		Esimerkkikohde		VAK	
		Esimerkkitie		IMS	
		Esimerkkikaupunki			
Selite	Luku väylältä	Selite	Luku & kirjoitus		
Keskimääräinen nopeus	0.22 m/s	Virtauksen asetusarvo	15 l/s		
Keskimääräinen ilmavirta m <sup>3</sup> /h	9.60 m <sup>3</sup> /h	Virtauksen asetusarvon minimi	0 l/s		
Keskimääräinen ilmavirta l/s	2.67 l/s	Virtauksen asetusarvon maksimi	1000 l/s		
Tila	Vika	Säätötyyppi	Modbus		
Moottorin tila	Moottori seis	Säätötapa	Säätö virtauksen mukaan		
Yksikön lämpötila	25.1 °C	Ohjauksen ohitus	Normaali ohjaus		
Pellin asento asteina	90 °	Kulma-anturin kalibrointi	Ei aktiivinen		
Pellin aukioolo prosentteina	100 %	RS485-protokolla	Modbus		
Nimelliskoko	125 mm	Modbusosoite	47		
Moottori ei toimi		Väylänopeus	19200		
Kulma-anturi ei toimi oikein		Pariteetti	Odd		
Virtauksen asetuspistettä ei saavutettu		Stop-bitit	1		
Virtauksen mittausongelma		Uudelleenkäynnistys	Ei aktiivinen		
Säätöpelti säätää					
Modbus-kommunikaatiohälytys					

Kuva 30. Hälytyksen simulointi.

Seuraavaksi testattiin kulma-anturin hälytyspisteen toiminta. Toiminta testattiin ottamalla moottorin säätöviesti pois, mikä saa moottorin liikkumaan. Kuvassa 31 nähdään, kuinka hälytyspiste on aktiivinen.

Fidelix		Esimerkkikohde		VAK	
		Esimerkkitie		IMS	
		Esimerkkikaupunki			
Selite	Luku väylältä	Selite	Luku & kirjoitus		
Keskimääräinen nopeus	0.02 m/s	Virtauksen asetusarvo	1 l/s		
Keskimääräinen ilmavirta m <sup>3</sup> /h	0.94 m <sup>3</sup> /h	Virtauksen asetusarvon minimi	0 l/s		
Keskimääräinen ilmavirta l/s	0.26 l/s	Virtauksen asetusarvon maksimi	1000 l/s		
Tila	Vika	Säätötyyppi	Modbus		
Moottorin tila	Moottori seis	Säätötapa	Säätö virtauksen mukaan		
Yksikön lämpötila	25.6 °C	Ohjauksen ohitus	Normaali ohjaus		
Pellin asento asteina	41 °	Kulma-anturin kalibrointi	Ei aktiivinen		
Pellin aukioolo prosentteina	46 %	RS485-protokolla	Modbus		
Nimelliskoko	125 mm	Modbusosoite	47		
Moottori ei toimi		Väylänopeus	19200		
Kulma-anturi ei toimi oikein		Pariteetti	Odd		
Virtauksen asetuspistettä ei saavutettu		Stop-bitit	1		
Virtauksen mittausongelma		Uudelleenkäynnistys	Ei aktiivinen		
Säätöpelti säätää					
Modbus-kommunikaatiohälytys					

Kuva 31. Kulma-anturin hälytyspisteen testaus.



Hälytyspisteiden testauksen jälkeen siirryttiin testaamaan pellin säätöominaisuuksia. Kuvassa 32 nähdään, kuinka säädin säätää peltiä moottorin avulla.

Fidelix Esimerkkikohde Esimerkkitie Esimerkkikaupunki		VAK IMS	
Selite	Luku väylältä	Selite	Luku & kirjoitus
Keskimääräinen nopeus	0.22 m/s	Virtauksen asetusarvo	0 l/s
Keskimääräinen ilmavirta m <sup>3</sup> /h	9.74 m <sup>3</sup> /h	Virtauksen asetusarvon minimi	0 l/s
Keskimääräinen ilmavirta l/s	2.71 l/s	Virtauksen asetusarvon maksimi	1000 l/s
Tila	Säätö käynnissä	Säätötyyppi	Modbus
Moottorin tila	Moottori sulkemassa peltiä	Säätötapa	Säätö virtauksen mukaan
Yksikön lämpötila	25.1 °C	Ohjauksen ohitus	Normaali ohjaus
Pellin asento asteina	89 °	Kulma-anturin kalibrointi	Ei aktiivinen
Pellin aukiolo prosentteina	99 %	RS485-protokolla	Modbus
Nimelliskoko	125 mm	Modbusosoite	47
Moottori ei toimi		Väylänopeus	19200
Kulma-anturi ei toimi oikein		Pariteetti	Odd
Virtauksen asetuspistettä ei saavutettu		Stop-bitit	1
Virtauksen mittausongelma		Uudelleenkäynnistys	Ei aktiivinen
Säätöpelti säätää			
Modbus-kommunikaatiohälytys			







Kuva 32. Säätö käynnissä.

Seuraavaksi testattiin ohjauksen ohituksen toiminto, 100 %:n auki-ohjaus. Kuvassa 33 nähdään, kuinka grafiikkakuvan vasemmassa reunassa säätimen tila muuttui ”ohjauksen ohitukseksi” ja moottorin tila oli ”moottori avaamassa peltiä”.

Fidelix Esimerkkikohde Esimerkkitie Esimerkkikaupunki		VAK IMS	
Selite	Luku väylältä	Selite	Luku & kirjoitus
Keskimääräinen nopeus	0.06 m/s	Virtauksen asetusarvo	0 l/s
Keskimääräinen ilmavirta m <sup>3</sup> /h	2.58 m <sup>3</sup> /h	Virtauksen asetusarvon minimi	0 l/s
Keskimääräinen ilmavirta l/s	0.74 l/s	Virtauksen asetusarvon maksimi	1000 l/s
Tila	Ohjauksen ohitus	Säätötyyppi	Modbus
Moottorin tila	Moottori avaamassa peltiä	Säätötapa	Säätö virtauksen mukaan
Yksikön lämpötila	25.0 °C	Ohjauksen ohitus	100% auki
Pellin asento asteina	71 °	Kulma-anturin kalibrointi	Ei aktiivinen
Pellin aukiolo prosentteina	79 %	RS485-protokolla	Modbus
Nimelliskoko	125 mm	Modbusosoite	47
Moottori ei toimi		Väylänopeus	19200
Kulma-anturi ei toimi oikein		Pariteetti	Odd
Virtauksen asetuspistettä ei saavutettu		Stop-bitit	1
Virtauksen mittausongelma		Uudelleenkäynnistys	Ei aktiivinen
Säätöpelti säätää			
Modbus-kommunikaatiohälytys			

Kuva 33. Ohjauksen ohituksen testaus.

Kulma-anturin kalibroinnin testaus nähdään kuvassa 34. Kuvan vasemmassa reunassa säätimen tila muuttui ”kulma-anturin kalibroinniksi”. Kuvan oikeasta reunasta aseteltiin sitä ennen kulma-anturin kalibrointi tilaan ”käynnistä kalibrointi”.

Fidelix		Esimerkkikohde		VAK	
		Esimerkkite		IMS	
		Esimerkki kaupunki			
Selite	Luku väylältä	Selite	Luku & kirjoitus		
Keskimääräinen nopeus	0.05 m/s	Virtauksen asetusarvo	1 l/s		
Keskimääräinen ilmavirta m <sup>3</sup> /h	2.01 m <sup>3</sup> /h	Virtauksen asetusarvon minimi	0 l/s		
Keskimääräinen ilmavirta l/s	0.56 l/s	Virtauksen asetusarvon maksimi	1000 l/s		
Tila	Kulma-anturin kalibrointi	Säätötyyppi	Modbus		
Moottorin tila	Moottori avaaassa peltiä	Säätötapa	SÄÄTÖ virtauksen mukaan		
Yksikön lämpötila	25.5 °C	Ohjauksen ohitus	Normaali ohjaus		
Pellin asento asteina	0 °	Kulma-anturin kalibrointi	Käynnistä kalibrointi		
Pellin aukio prosentteina	0 %	RS485-protokolla	Modbus		
Nimelliskoko	125 mm	Modbusosoite	47		
Moottori ei toimi		Väylänopeus	19200		
Kulma-anturi ei toimi oikein		Pariteetti	Odd		
Virtauksen asetuspistettä ei saavutettu		Stop-bitit	1		
Virtauksen mittausero		Uudelleenkäynnistys	Ei aktiivinen		
Säätöpelti säätää					
Modbus-kommunikaatiohälytys					

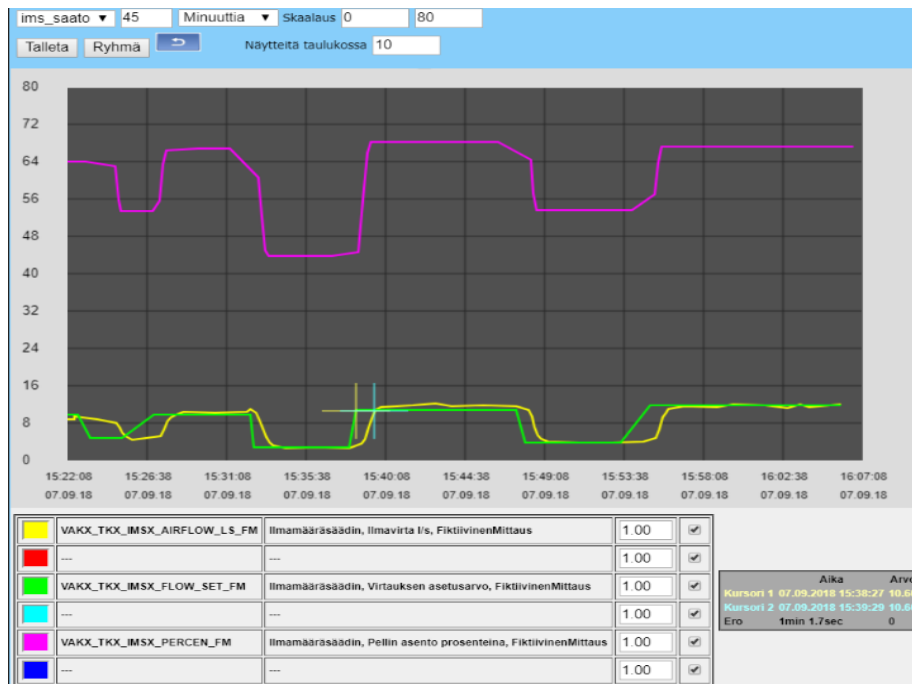
Kuva 34. Kulma-anturin kalibroinnin testaus.

Seuraavana testattiin ilmamääräsäätimen ”normaali” tilanne, jossa mittaus on lähellä asetettua arvoa, eikä hälytyksiä ole. Kuvassa 35 nähdään ilmamääräsäätimen normaali tila.

Fidelix Esimerkkikohde Esimerkkite Esimerkkikaupunki		VAK IMS	
Selite	Luku väyläitä	Selite	Luku & kirjoitus
Keskimääräinen nopeus	1.00 m/s	Virtauksen asetusarvo	12 l/s
Keskimääräinen ilmavirta m <sup>3</sup> /h	43.97 m <sup>3</sup> /h	Virtauksen asetusarvon minimi	0 l/s
Keskimääräinen ilmavirta l/s	12.22 l/s	Virtauksen asetusarvon maksimi	1000 l/s
Tila	Normaali	Säätötyyppi	Modbus
Moottorin tila	Moottori seis	Säätötapa	Säätö virtauksen mukaan
Yksikön lämpötila	25.6 °C	Ohjauksen ohitus	Normaali ohjaus
Pellin asento asteina	61 °	Kulma-anturin kalibrointi	Ei aktiivinen
Pellin aukioalo prosentteina	68 %	RS485-protokolla	Modbus
Nimelliskoko	125 mm	Modbusosoite	47
Moottori ei toimi		Väylänopeus	19200
Kulma-anturi ei toimi oikein		Pariteetti	Odd
Virtauksen asetuspistettä ei saavutettu		Stop-bitit	1
Virtauksen mittausongelma		Uudelleenkäynnistys	Ei aktiivinen
Säätöpelti säätää			
Modbus-kommunikaatiohälytys			

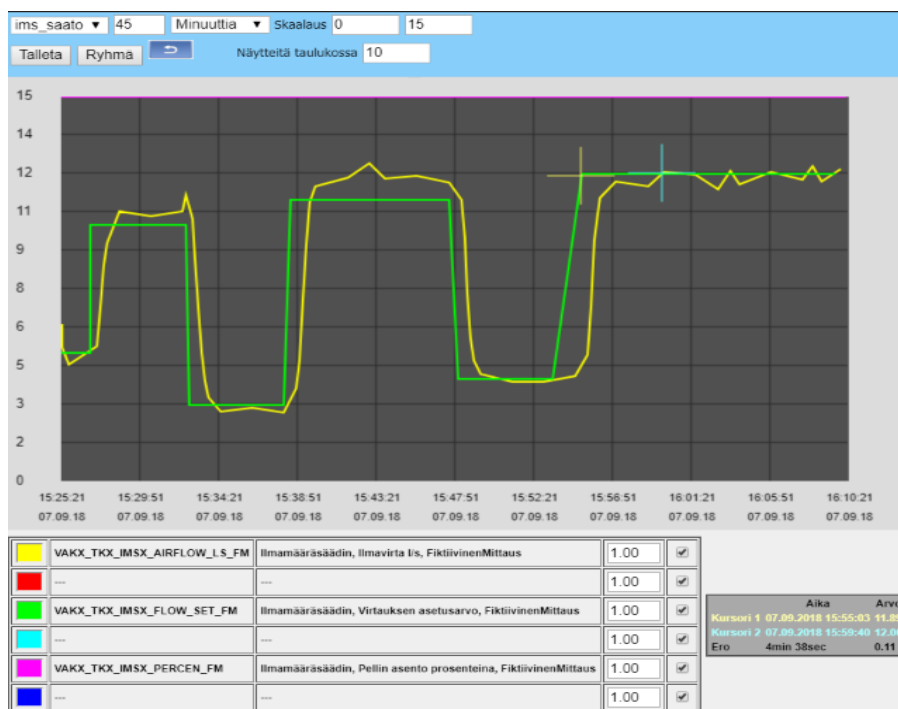
Kuva 35. Ilmamääräsäätimen normaali tila, jossa ei ole lainkaan hälytyksiä, ja todellinen mittaus on lähellä asetettua arvoa.

Lopuksi testattiin ilmamääräsäätimen säädinparametreja, eli miten tehokkaasti ilmamääräsäädin reagoi asetusarvon muutokseen ja onko säätö normaalikäytössä stabiili. Kuvan 36 käyrästä voidaan päätellä, että ilmamääräsäädin toimii hyvin ja se reagoi asetusarvon muutoksiin tehokkaasti. Asetusarvo näkyy käyrässä vihreällä värillä, ja mittaus näkyy keltaisella värillä. Violetti väri kertoo pellin asennon prosentteina. Eropisteistä voidaan myös havaita, että asetusarvon muutos ylöspäin noin 8 l/s saavutettiin hieman yli minuutissa, jonka jälkeen mittaus stabiloitui. Asetusarvon laskeminen aiheutti samantapaisen muutoksen käyrään.



Kuva 36. Ilmamääräsäätimen säädinparametrien testaus.

Kuvassa 37 testattiin hieman isompaa asetusarvon muutosta.



Kuva 37. Säädinparametrien testausta suuremmalla asetusarvon muutoksella.

Kuvasta 37 nähdään, että säädin pystyy toteuttamaan suuremmankin asetusarvon muutoksen siedettävässä ajassa.

Näiden tuloksien varjolla voidaan todeta, että integraatiosta tuli toimiva ja tätä ohjelmakirjastoa voidaan käyttää muissakin projekteissa.

## 6 Yhteenveto

Insinööriyössä käsiteltiin uudenlaisen ultraääni-ilmamääräsäätimen teknisiä ominaisuuksia, sen heikkouksia ja vahvuuksia sekä tehtiin täydellinen väyläintegraatio rakennusautomaatiojärjestelmään Modbus-sarjaliikenneprotokollan avulla. Havaittiin, että ultraääni-ilmamääräsäätimet ovat osaltaan luotettavampia ja huoltovapaampia vanhaan malliin nähden – johtuen niiden uudenlaisesta mittaustavasta, minkä ansiosta kanaviston likaantuminenkaan ei häiritse mittaustarkkuutta.

Väyläintegraation sekä ilmamääräsäätimen säädinparametrien tarkistus toi lupaavia tuloksia, eikä ollut merkkiäkään siitä, että väylän toimivuus olisi ollut heikkoa tai epäluotettavaa. Ilmamääräsäädin reagoi odotusten mukaisesti asetusarvon muutokseen, ja mitauskin pysyi stabiilina. Nämä tulokset mahdollistavat hankalimpienkin projektien toteuttamista erilaisissa ympäristöissä.

Väyläintegraation toimivuus oli merkittävä paalu, joka saavutettiin projektin edetessä. Tulevaisuudessa eri projektipäälliköt voivat hyödyntää tätä täydellistä ohjelmakirjastoa, joka varmasti toimii ja on luotettava. Tämä säästää huomattavan määrän aikaa monelta eri ihmiseltä. Ohjelmakirjastoa voidaan muokata helposti tulevaisuudessakin, mikäli ilmamääräsäätimen rekisterit tai muut ominaisuudet muuttuvat. Koodia kommentointiin niin, että se oli helppolukuista ja helposti muokattavaa sekä kopioitavaa.

Vain tulevaisuus näyttää, miten ultraäänellä mittaavat ilmamääräsäätimet oikeasti toimivat. Kärsiikö mittaustarkkuus laitteen vanhentuessa? Ilmaantuuko laitteisiin muita ominaisvikoja, joita ei alun perin osattu ennustaa?

## Lähteet

- 1 Koskinen, Kari. 2018. Automaation historia, nykytila ja tulevaisuus. Automaation verkkolehti 1/2018, s. 8. Luettu 3.3.2019.
- 2 TEK NTC10. 2006. Verkkoaineisto. Pro dual.  
<[http://www.produal.com/fi/shop/web\\_duct\\_sensors/sku-1175040#dataSheet](http://www.produal.com/fi/shop/web_duct_sensors/sku-1175040#dataSheet)>. Luettu 3.3.2019.
- 3 DPT-R8. 2017. Verkkoaineisto. HK Instruments.  
<<http://hkstruments.fi/fi/tuotteet/paine-erolahettimet-ilmalle/dpt-r8/>>. Luettu 3.3.2019.
- 4 CDT2000. 2017. Verkkoaineisto. HK Instruments.  
<<http://hkstruments.fi/fi/tuotteet/hiilidioksidilahettimet/cdt2000/>>. Luettu 3.3.2019.
- 5 Lux 34 ja Lux 34-100 valoisuus- ja lämpötilalähetin. 2010. Verkkoaineisto. Pro dual.  
<[http://www.produal.com/fi/shop/by\\_valoisuus/sku-1133310#dataSheet](http://www.produal.com/fi/shop/by_valoisuus/sku-1133310#dataSheet)>. Luettu 3.3.2019.
- 6 Harju, Timo. 2000. Sääätötekniikan koulutusmateriaali. Espoo: Automaatiosäätiö. Luettu 3.3.2019
- 7 Jylkkä, Henri. 2014. Rakennusautomaatiokirjasto ja suunnitteluohje asuinrakennusten rakennusautomaatiosuunnitteluun. Opinnäytetyö.  
<<https://www.theseus.fi/handle/10024/81769>>. Luettu 3.3.2019.
- 8 LRQ24A-SR. 2009. Verkkoaineisto. Belimo.  
<[https://www.belimo.fi/pdf/fi/LRQ24A-SR\\_datasheet\\_fi-fi.pdf](https://www.belimo.fi/pdf/fi/LRQ24A-SR_datasheet_fi-fi.pdf)>. Luettu 3.3.2019.
- 9 Keskinen, Jarkko. 2014. Keskipakopumpun ja hammaspyöräpumpppujen rakenteet ja toimintaperiaatteet. Opinnäytetyö.  
<<https://www.theseus.fi/handle/10024/78819>>. Luettu 4.3.2019.
- 10 MAGNA 3. 2019. Verkkoaineisto. Grundfos.  
<<https://fi.grundfos.com/tuotteet/etsi-tuote/magna3.html#overview>>. Luettu 4.3.2019.
- 11 SF24A. 2016. Verkkoaineisto. Belimo.  
<[https://www.belimo.fi/pdf/e/SF24A\\_datasheet\\_en-gb.pdf](https://www.belimo.fi/pdf/e/SF24A_datasheet_en-gb.pdf)>. Luettu 4.3.2019.

- 12 FDX Compact FX-3000 C. 2019. Verkkoaineisto. Fidelix.  
<[https://www.fidelix.fi/wp-content/uploads/FX-3000-C\\_FI.pdf](https://www.fidelix.fi/wp-content/uploads/FX-3000-C_FI.pdf)>. Luettu 4.3.2019.
- 13 FDX Compact VISIO-15-C. 2019. Verkkoaineisto. Fidelix.  
<[https://www.fidelix.fi/wp-content/uploads/VISIO-15-C\\_FI.pdf](https://www.fidelix.fi/wp-content/uploads/VISIO-15-C_FI.pdf)>. Luettu 4.3.2019.
- 14 FDX Compact-tuotesarja. 2019. Verkkoaineisto. Fidelix.  
<<https://www.fidelix.fi/wp-content/uploads/Fidelix-FdxCompact-modules-FI.pdf>>. Luettu 4.3.2019.
- 15 Multi-24. Verkkoaineisto. Fidelix Oy.  
<[https://www.fidelix.fi/wp-content/uploads/MULTI24\\_FI.pdf](https://www.fidelix.fi/wp-content/uploads/MULTI24_FI.pdf)>. Luettu 5.3.2019.
- 16 Piikkilä, Veijo. 2008. ST-Käsikirja 22, Kiinteistöjen valvomojärjestelmät. Tampere: Sähköinfo Oy.
- 17 Värjä, Pertti. 1999. Uusi kiinteistöautomaatio: Automaatio ja säätötekniikka. Elämäki: Korian kirjapaino Ky.
- 18 Piikkilä, Veijo. 2006. ST-Käsikirja 21, Kiinteistöjen tiedonsiirtoväylät. Tampere: Sähköinfo Oy.
- 19 M-Bus-etäluennan suunnitteluohje. 2018. Verkkoaineisto. Saint-Gobain Finland Oy/Pam.  
<<https://www.pamline.fi/ratkaisut/veden-ja-energianmittaus>>. Luettu 11.6.2019.
- 20 BACnet: Answers to Frequently Asked Questions. 1998. Verkkoaineisto. American Society of Heating.  
<<http://www.bacnet.org/FAQ/HPAC-3-97.html>>. Luettu 23.7.2019.
- 21 DALI manual. 2013. Verkkoaineisto. Tridonic.  
<[http://www.tridonic.se/it/download/technical/DALI-manual\\_en.pdf](http://www.tridonic.se/it/download/technical/DALI-manual_en.pdf)>. Luettu 23.7.2019.
- 22 Ilmanvaihtojärjestelmät. 2019. Verkkoaineisto. Hengitysliitto.  
<<https://www.hengitysliitto.fi/fi/sisailma/ilmanvaihto/ilmanvaihtojarjestelmat>>. Luettu 25.7.2019.
- 23 Lindab UltraLink. 2019. Verkkoaineisto. Lindab. <<http://www.lindab.com/fi/pro/Pages/ultralink.aspx>>. Luettu 25.7.2019.
- 24 Lindab UltraLink-ilmavirtasäädin. 2019. Tekniset tiedot -tuotelehti. Lindab. Luettu 13.8.2019.

## Ilmamääräsäätimen Modbus-rekisterilista

Address :	Modbus register address (3x indicates Input & 4x indicates Holding)
UltraLink® :	Type of UltraLink® where the register is available (Indicated by “x”)
Name:	Name of register
Description:	Short description of register.
Data type:	Data type for register (16bit contained in one register, 32bit and float in two consecutive registers).
Unit:	Unit for register value (if any).
Div:	Scale factor for stored value (divide register value with “div” to get correct value).
Default:	Default setting.
Min:	Minimum value allowed for the register.
Max:	Maximum value allowed for the register.
Access:	RO for read only (Input registers) and RW for read and write (Holding registers).

Address	UltraLink®		Name	Description	Data type	Unit	Div	Default	Min	Max	Access
	Controller	Monitor									
<b>INPUT REGISTERS</b>											
3x008	X	X	Product Nominal Size	Nominal diameter of duct	16bit	mm					RO
3x013	X	X	Unit Status	Current unit status: 0 = Normal mode; 1 = Locating flow; 2 = Override control; 3 = Error; 4 = Control loop regulating; 5 = Angle sensor calibrating	16bit						RO
<b>Flow info</b>											
3x150	X	X	Velocity in m/s	Velocity in m/s	Float	m/s					RO
3x152	X	X	Air flow in m³/h	Air flow in m³/h	Float	m³/h					RO
3x154	X	X	Air flow in l/s	Air flow in l/s	Float	l/s					RO
<b>Temperature info</b>											
3x200	X	X	Current temperature in °C	Temperature in degree celcius.	16bit	°C	10				RO
<b>Damper info</b>											
3x251	X		Damper open in %	Damper actual position in percentage open.	16bit	%	10				RO
3x252	X		Damper motor action	Damper motor action: 0 = Motor stopped. 1 = Motor opening damper 2 = Motor closing damper	16bit						RO
<b>Alarms</b>											
3x400	X	X	Alarm Register 1	Alarms 1-32 - bitwise: 1 = Motor not working. 2 = Angle sensor not working correctly. 3 = Flow setpoint not reached. 4 = Flow measure problems. 5 = Damper is regulating. 6 = Not used. 7 - 31 = Reserved for future use. 32 = Factory data is corrupted.	32bit						RO
<b>Other</b>											
3x500	X	X	Signal amplification	Current signal amplification	16bit			0	3	20	RO



Address	UltraLink®		Name	Description	Data type	Unit	Div	Default	Min	Max	Access
	Controller	Monitor									
<b>HOLDING REGISTERS</b>											
<b>Communication settings</b>											
4x001	X	X	Communication id	Modbus address	16bit				1	239	RW
4x002	X	X	RS485 Baud Rate Conf.	Baudrate: 0 = 9600 1 = 19200 2 = 38400 3 = 76800	16bit			1	0	3	RW
4x003	X	X	RS485 Parity Conf.	Parity: 0 = Odd; 1 = Even; 2 = None	16bit			0	0	2	RW
4x004	X	X	RS485 Stop Bit Conf.	Number of stopbits: 1 or 2.	16bit			1	1	2	RW
4x005	X	X	RS485 Protocol Conf.	Protocol: 0 = Modbus; 1 = Not used; 2 = Pascal;	16bit			0	0	2	RW
4x006	X	X	Bluetooth Password	Password which must be provided to pair Bluetooth devices. This password can always be changed from wired connection. From wireless it can only be changed when connection is established using current password.	16bit			1111	0000	9999	RW
4x007	X	X	Bluetooth Enable	Enable Bluetooth Communication 0 = Bluetooth turned off; 1 = Bluetooth turned on;	16bit			1	0	2	RW
4x008	X	X	PLA	ID used for Pascal	16bit				1	239	RW
4x009	X	X	ELA	ID used for Pascal	16bit				1	239	RW
4x010	X	X	Bluetooth TX Power Level	Configure TX Power Level dBm. Accepted values: -40, -20, -16, -12, -8, -4, 0, 2, 3, 4, 5, 6, 7, 8, 9	16bit			0	-40	9	RW
<b>System configuration</b>											
4x070	X		Damper Regulation Conf.	Specifies how damper is regulated: 0 = Regulator turned off 1 = Regulate damper angle 2 = Regulate flow	16bit			2	0	2	RW
4x071	X		Damper Input Conf.	Specifies input to control damper: 0 = Modbus or Pascal 1 = Analog input	16bit			1	0	1	RW
4x082	X	X	Execute Factory Reset	Factory reset of all parameters. Unit will restart 0 = Do nothing; 1 = Factory Reset	16bit			0	0	1	RW
4x083	X	X	Execute Reboot	Reboot the unit 0 = Do nothing; 1 = Reboot the unit;	16bit			0	0	1	RW
<b>Override configuration</b>											
4x150	X		Damper Override Timeout	Time before returning to normal mode	16bit	min		120	0	600	RW
4x151	X		Damper Override Conf.	0 = Normal mode; 1 = Override control - Max open; 2 = Override control - Min open; 3 = Override control - 100% open; 4 = Override control - 100% closed	16bit			0	0	4	RW
<b>Damper</b>											
4x300	X		Execute Angle Calibration	0 = Do nothing; 1 = Start recalibration of the angle sensor; 2 = Start recalibration when starting up;	16bit			0	0	2	RW
4x302	X		Angle Set Point	Angle setpoint used in normal mode. (Only relevant when 4x070 is set to 1 )	16bit	%		0	0	100	RW
4x314	X		Flow Set Point	Flow setpoint used in normal mode. (Only relevant when 4x070 is set to 2 )	16bit	l/s		*	0	4700	RW

\* = the value depends on the dimension of the product.

Address	UltraLink®		Name	Description	Data type	Unit	Div	Default	Min	Max	Access
	Controller	Monitor									
4x315	X		Flow Set Point Minimum	Flow setpoint min.	16bit	l/s		*	0	4700	RW
4x316	X		Flow Set Point Maximum	Flow setpoint max.	16bit	l/s		*	0	4700	RW
<b>Analog output</b>											
4x400	X	X	Analog Output 1 Level Conf.	Analog output config: 0 = 0-10 V, 1 = 10-0 V, 2 = 2-10 V, 3 = 10-2 V.	16bit			2	0	3	RW
4x401	X	X	Analog Output 1 Unit Conf.	Show: 0 = Flow; 1 = Temperature; 2 = Angle;	16bit			0	0	2	RW
4x402	X	X	Analog Output 1 Temp. Min.	Min temperature shown = Min output voltage (Only relevant when 4x401 is set to 1)	16bit	°C		0	-40	50	RW
4x403	X	X	Analog Output 1 Temp. Max.	Max temperature shown = Max output voltage (Only relevant when 4x401 is set to 1)	16bit	°C		50	-40	50	RW
4x404	X	X	Analog Output 1 Flow Min.	Min flow shown = Min output voltage (Only relevant when 4x401 is set to 0)	16bit	l/s		0	-4700	4700	RW
4x406	X	X	Analog Output 1 Flow Max.	Max flow shown = Max output voltage (Only relevant when 4x401 is set to 0)	16bit	l/s		*	-4700	4700	RW
4x408	X		Analog Output 1 % Open Min.	Min open % shown = Min output voltage (Only relevant when 4x401 is set to 2)	16bit	%	10	0	0	1000	RW
4x409	X		Analog Output 1 % Open Max.	Max open % shown = Max output voltage (Only relevant when 4x401 is set to 2)	16bit	%	10	1000	0	1000	RW
4x430	X	X	Analog Output 2 Level Conf.	Analog output config: 0 = 0-10 V, 1 = 10-0 V, 2 = 2-10 V, 3 = 10-2 V.	16bit			2	0	3	RW
4x431	X	X	Analog Output 2 Unit Conf.	Show: 0 = Flow 1 = Temperature 2 = Angle	16bit			2	0	2	RW
4x432	X	X	Analog Output 2 Temp. Min.	Min temperature shown = Min output voltage (Only relevant when 4x431 is set to 1)	16bit	°C		0	-40	50	RW
4x433	X	X	Analog Output 2 Temp. Max.	Max temperature shown = Max output voltage (Only relevant when 4x431 is set to 1)	16bit	°C		50	-40	50	RW
4x434	X	X	Analog Output 2 Flow Min.	Min flow shown = Min output voltage (Only relevant when 4x431 is set to 0)	16bit	l/s		0	-4700	4700	RW
4x436	X	X	Analog Output 2 Flow Max.	Max flow shown = Max output voltage (Only relevant when 4x431 is set to 0)	16bit	l/s		*	-4700	4700	RW
4x438	X		Analog Output 2 % Open Min.	Min open % shown = Min output voltage (Only relevant when 4x431 is set to 2)	16bit	%	10	0	0	1000	RW
4x439	X		Analog Output 2 % Open Max.	Max open % shown = Max output voltage (Only relevant when 4x431 is set to 2)	16bit	%	10	1000	0	1000	RW

Address	UltraLink®		Name	Description	Data type	Unit	Div	Default	Min	Max	Access
	Controller	Monitor									
<b>Analog input (Settings below are only relevant when register 4x071 is set to 1)</b>											
4x500	X		Analog In Level Conf.	Analog input: 0 = 0-10 V, 1 = 10-0 V, 2 = 2-10 V, 3 = 10-2 V.	16bit			2	0	3	RW
4x501	X		Analog In Angle Minimum	Min angle = min voltage	16bit	%		0	0	100	RW
4x502	X		Analog In Angle Maximum	Max = max voltage	16bit	%		100	0	100	RW
4x503	X		Analog In Flow Minimum	Min flow = min voltage (Must be equal or higher than register 4x315)	16bit	l/s		0	0	4700	RW
4x504	X		Analog In Flow Maximum	Max flow = max voltage (Must be equal or lower than register 4x316)	16bit	l/s		*	0	4700	RW
4x510	X		Analog In Override Low Trigger Min.	Lowest voltage level to activate 1st Override level (Only relevant when 4x500 is set to 2 or 3)	16bit	V	10	0	0	20	RW
4x511	X		Analog In Override Low Trigger Max.	Highest voltage level to activate 1st Override level (Only relevant when 4x500 is set to 2 or 3)	16bit	V	10	8	0	20	RW

## Työn tuloksena syntynyt ohjelmakirjasto

```
FUNCTION_BLOCK FxIEC_Lindab_UltraLink_FTCU_FB
VAR_EXTERNAL

END_VAR

VAR_INPUT
    iModuleAddress          : INT;
    iPort                   : INT;
    id_ModuleAlarm          : STRING(30);

    inputreg_7 : string(30);
    inputreg_12 : string(30);
    inputreg_149 : string(30);
    inputreg_151 : string(30);
    inputreg_153 : string(30);
    inputreg_199 : string(30);
    inputreg_249 : string(30);
    inputreg_250 : string(30);
    inputreg_251 : string(30);
    inputreg_399_1 : string(30);
    inputreg_399_2 : string(30);
    inputreg_399_3 : string(30);
    inputreg_399_4 : string(30);
    inputreg_399_5 : string(30);
    inputreg_399_6 : string(30);

    holdreg_0 : string(30);
    holdreg_1 : string(30);
    holdreg_2 : string(30);
    holdreg_3 : string(30);
    holdreg_4 : string(30);
    holdreg_69 : string(30);
    holdreg_70 : string(30);
    holdreg_82 : string(30);
    holdreg_149 : string(30);
    holdreg_150 : string(30);
    holdreg_299 : string(30);
    holdreg_313 : string(30);
    holdreg_313_force : BOOL;
    holdreg_313_releaseoverride : BOOL;
    holdreg_314 : string(30);
    holdreg_315 : string(30);

END_VAR

VAR_OUTPUT

END_VAR

VAR
    fbRegisters          :GenericModbus642FB;
    Result                :int;
    uintn_Result         :uint;
    Float_Result         :real;
```

END\_VAR

```
(* Lindab Ultralink-ilmamääräsäätimen modbus-liitos *)
(* 13.8.2018
    Teemu Kortelainen / Fidelix Oy
    Versio 1.0
*)
```

```
(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 7 ,
RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Result := InputRegF(id_String := inputreg_7, RegValue := fbRegisters.Reg0, rMultiplier := 1.0);
end_if;
```

```
(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 12
, RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Result := InputRegF(id_String := inputreg_12, RegValue := fbRegisters.Reg0, rMultiplier := 1.0);
end_if;
```

```
(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 149
, RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Float_Result := DoubleInputReg_Float(id_String:=inputreg_149, HighRegValue:=fbRegisters.REG0, LowRegValue:=fbRegisters.REG1);
end_if;
```

```
(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 151
, RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Float_Result := DoubleInputReg_Float(id_String:=inputreg_151, HighRegValue:=fbRegisters.REG0, LowRegValue:=fbRegisters.REG1);
end_if;
```

```
(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 153
, RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Float_Result := DoubleInputReg_Float(id_String:=inputreg_153, HighRegValue:=fbRegisters.REG0, LowRegValue:=fbRegisters.REG1);
end_if;
```

```
(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 199
, RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Result := InputRegF(id_String := inputreg_199, RegValue := fbRegisters.Reg0, rMultiplier := 10.0);
end_if;
```

```

(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 249
, RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Result := InputRegF(id_String := inputreg_249, RegValue := fbRegisters.Reg0, rMultiplier := 10.0);
    Result := InputRegF(id_String := inputreg_250, RegValue := fbRegisters.Reg1, rMultiplier := 10.0);
    Result := InputRegF(id_String := inputreg_251, RegValue := fbRegisters.Reg2, rMultiplier := 1.0);
end_if;

(* Input Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 399
, RegisterType:=4);
if fbRegisters.DataValid = 1 then
    Result := DiscreteInputF(id_String := inputreg_399_1, RegValue := fbRegisters.Reg1, BitMask := 16#0001);
    Result := DiscreteInputF(id_String := inputreg_399_2, RegValue := fbRegisters.Reg1, BitMask := 16#0002);
    Result := DiscreteInputF(id_String := inputreg_399_3, RegValue := fbRegisters.Reg1, BitMask := 16#0004);
    Result := DiscreteInputF(id_String := inputreg_399_4, RegValue := fbRegisters.Reg1, BitMask := 16#0008);
    Result := DiscreteInputF(id_String := inputreg_399_5, RegValue := fbRegisters.Reg1, BitMask := 16#0010);
end_if;

(* Holding Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 0
, RegisterType:=3);
if fbRegisters.DataValid = 1 then
    fbRegisters.Reg0 := HoldingRegF_3(id_String := holdreg_0, RegValue := fbRegisters.Reg0, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);
    fbRegisters.Reg1 := HoldingRegF_3(id_String := holdreg_1, RegValue := fbRegisters.Reg1, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);
    fbRegisters.Reg2 := HoldingRegF_3(id_String := holdreg_2, RegValue := fbRegisters.Reg2, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);
    fbRegisters.Reg3 := HoldingRegF_3(id_String := holdreg_3, RegValue := fbRegisters.Reg3, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);
    fbRegisters.Reg4 := HoldingRegF_3(id_String := holdreg_4, RegValue := fbRegisters.Reg4, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

    fbRegisters (Send:=2);
end_if;

(* Holding Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 69
, RegisterType:=3);
if fbRegisters.DataValid = 1 then
    fbRegisters.Reg0 := HoldingRegF_3(id_String := holdreg_69, RegValue := fbRegisters.Reg0, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);
    fbRegisters.Reg1 := HoldingRegF_3(id_String := holdreg_70, RegValue := fbRegisters.Reg1, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

    fbRegisters (Send:=2);
end_if;

```

```
(* Holding Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 82
, RegisterType:=3);
if fbRegisters.DataValid = 1 then
    fbRegisters.Reg0 := HoldingRegF_3(id_String := holdreg_82, RegValue :=
fbRegisters.Reg0, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

    fbRegisters (Send:=2);
end_if;

(* Holding Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 149
, RegisterType:=3);
if fbRegisters.DataValid = 1 then
    fbRegisters.Reg0 := HoldingRegF_3(id_String := holdreg_149, RegValue :=
fbRegisters.Reg0, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);
    fbRegisters.Reg1 := HoldingRegF_3(id_String := holdreg_150, RegValue :=
fbRegisters.Reg1, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

    fbRegisters (Send:=2);
end_if;

(* Holding Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 299
, RegisterType:=3);
if fbRegisters.DataValid = 1 then
    fbRegisters.Reg0 := HoldingRegF_3(id_String := holdreg_299, RegValue :=
fbRegisters.Reg0, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

    fbRegisters (Send:=2);
end_if;

(* Holding Registers *)
fbRegisters(Port:=iPort, Send:=0, Module:=iModuleAddress, StartRegister := 313
, RegisterType:=3);
if fbRegisters.DataValid = 1 then
    fbRegisters.Reg0 := HoldingRegF_3(id_String := holdreg_313, RegValue :=
fbRegisters.Reg0, rMultiplier:=1.0, Force := holdreg_313_force, ReleaseOver-
ride := holdreg_313_releaseoverride);
    fbRegisters.Reg1 := HoldingRegF_3(id_String := holdreg_314, RegValue :=
fbRegisters.Reg1, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);
    fbRegisters.Reg2 := HoldingRegF_3(id_String := holdreg_315, RegValue :=
fbRegisters.Reg2, rMultiplier:=1.0, Force := FALSE, ReleaseOverride := TRUE);

    fbRegisters (Send:=2);
end_if;

(* Modulealarm *)
Result := SetDigitalPointF( Value:=BOOL_TO_INT(GetSystemStatusF( Mode:=1, iPa-
rameter:=iModuleAddress, rParameter:=INT_TO_REAL(iPort)) > 30.0), Lock-
State:=1, Name:=id_ModuleAlarm );

END_FUNCTION_BLOCK
```

```
PROGRAM FxIEC_Lindab_UltraLink_FTCU
VAR_EXTERNAL
```

```
END_VAR
```

```
VAR_GLOBAL
```

```
END_VAR
```

```
VAR
```

```
fb_ims : FxIEC_Lindab_UltraLink_FTCU_FB;
```

```
END_VAR
```

```
(* Lindab Ultralink-ilmamääräsäätimen modbus-liitos *)
```

```
(* 13.8.2018
```

```
Teemu Kortelainen / Fidelix Oy
```

```
Modbus-osoite: Laitteen sarjanumeron kolme viimeistä numeroa (näkyvä myös  
laitteen näytössä)
```

```
Baudrate: 19200 (9600, 19200, 38400, 76800)
```

```
Databits: 8
```

```
Parity: Odd (Even, None)
```

```
Stopbits: 1 (2)
```

```
Modbuslaitteet:
```

Tyyppi	Aloitus	Määrä
Input regs	7	1
Input regs	12	1
Input regs	149	2
Input regs	151	2
Input regs	153	2
Input regs	199	1
Input regs	249	3
Input regs	399	2
Holding regs	0	5
Holding regs	69	2
Holding regs	82	1
Holding regs	149	2
Holding regs	299	1
Holding regs	313	3

\*)

```
fb_ims(
```

```
  iModuleAddress :=47,
```

```
  iPort :=3,
```

```
(* Read only *)
```

```
  inputreg_7 := 'VAKX_TKX_IMSX_SIZE_FM', (* Product nominal size: Nominal  
  diameter of duct (Unit = mm) *)
```

```
  inputreg_12 := 'VAKX_TKX_IMSX_STATUS_FI', (* Unit Status: Current unit  
  status: 0= Normal mode; 1 = Locating flow; 2 = Override control; 3 = Error; 4  
  = Control loop regulating; 5 = Angle sensor calibrating *)
```

```
  inputreg_149 := 'VAKX_TKX_IMSX_VELOCITY_FM', (* Average velocity in m/s  
  (Unit = m/s) *)
```

```
  inputreg_151 := 'VAKX_TKX_IMSX_AIRFLOW_M3H_FM', (* Average air flow in  
  m3/hs (Unit = m3/h) *)
```

```
  inputreg_153 := 'VAKX_TKX_IMSX_AIRFLOW_LS_FM', (* Average air flow in l/s  
  (Unit = l/s) *)
```

```
  inputreg_199 := 'VAKX_TKX_IMSX_TEMP_FM', (* Current temperature in °C:  
  Temperature in degree celcius (Unit = °C) *)
```



```

    inputreg_249 := 'VAKX_TKX_IMSX_POS_FM', (* Current damper position:
Damper actual position in degrees (Unit = deg) *)
    inputreg_250 := 'VAKX_TKX_IMSX_PERCEN_FM', (* Damper open in %: Damper
actual position in percentage open (Unit = %) *)
    inputreg_251 := 'VAKX_TKX_IMSX_MOTOR_FI', (* Damper motor action: 0 =
Motor stopped; 1 = Motor opening damper; 2 = Motor closing damper *)
    inputreg_399_1 := 'VAKX_TKX_IMSX_MOTOR_ALARM_FH', (* Alarm: Motor not
working correctly *)
    inputreg_399_2 := 'VAKX_TKX_IMSX_SENSOR_FH', (* Alarm: Angle sensor not
working correctly *)
    inputreg_399_3 := 'VAKX_TKX_IMSX_FLOW_SET_FH', (* Alarm: Flow setpoint
not reached *)
    inputreg_399_4 := 'VAKX_TKX_IMSX_FLOW_PROB_FH', (* Alarm: Flow measure
problems *)
    inputreg_399_5 := 'VAKX_TKX_IMSX_DAMPER_FH', (* Alarm: Damper is regu-
lating *)

(* Read & write *)
    holdreg_0 := 'VAKX_TKX_IMSX_MODBUS_FM', (* Communication id: Modbus ad-
dress *)
    holdreg_1 := 'VAKX_TKX_IMSX_BAUD_FO', (* RS485 Baudrate conf. Baudrate:
0 = 9600; 1 = 19200; 2 = 38400; 3 = 76800 *)
    holdreg_2 := 'VAKX_TKX_IMSX_PARITY_FO', (* RS485 Parity conf. Parity: 0
= Odd; 1 = Even; 2 = None *)
    holdreg_3 := 'VAKX_TKX_IMSX_STOPBIT_FO', (* RS485 Stop bit conf. Number
of stopbits: 1 or 2 *)
    holdreg_4 := 'VAKX_TKX_IMSX_PROTOCOL_FO', (* RS485 Protocol conf. Proto-
col: 0 = Modbus; 1 = Not used; 3 = Pascal *)
    holdreg_69 := 'VAKX_TKX_IMSX_REGULATION_FO', (* Damper regulation conf.
Specifies how damper is regulated: 0 = Regulator turned off; 1 = Regulate
damper angle; 2 = Regulate flow *)
    holdreg_70 := 'VAKX_TKX_IMSX_INPUT_FO', (* Damper input conf. Specifies
input to control damper: 0 = Modbus or Pascal; 1 = Analog input *)
    holdreg_82 := 'VAKX_TKX_IMSX_REBOOT_FO', (* Execute reboot: Reboot the
unit: 0 = Do nothing; 1 = Reboot the unit *)
    holdreg_150 := 'VAKX_TKX_IMSX_OVERRIDE_FO', (* Damper override conf. 0 =
Normal mode; 1 = Override control - Max open; 2 = Override control - Min open;
3 = Override control - 100% open; 4 = Override control - 100% closed *)
    holdreg_299 := 'VAKX_TKX_IMSX_CALIB_FO', (* Execute angle calibration: 0
= Do nothing; 1 = Start recalibration of the angle sensor; 2 = Start recal-
ibrating when starting up *)
    holdreg_313 := 'VAKX_TKX_IMSX_FLOW_SET_FM', holdreg_313_force := TRUE,
holdreg_313_releaseoverride := FALSE, (* Flow setpoint: Flow setpoint used in
normal mode (only relevant when reg. 69 is set to 2) (Unit = l/s) *)
    holdreg_314 := 'VAKX_TKX_IMSX_FLOW_SET_MIN_FM', (* Flow setpoint mini-
mum: Flow setpoint min. (Unit = l/s) *)
    holdreg_315 := 'VAKX_TKX_IMSX_FLOW_SET_MAX_FM', (* Flow setpoint maxi-
mum: Flow setpoint max. (Unit = l/s) *)

    id_ModuleAlarm := 'VAKX_TKX_IMSX_MODBUS_FH' (* Modbus-kommu-
nikaatiohälytys *)
);

(*** Ohje forcelle ja releaseoverridelle ***)

(*
    holdreg_x_force:
    TRUE: Mikäli syötät holdreg_x_forceen TRUE-tekstin, voit ohjata pistettä
IEC-ohjelman kautta.
    FALSE: Mikäli syötät holdreg_x_forceen FALSE-tekstin, pisteen tieto menee
IMS:lle

```

vasta pisteen ollessa käsiohjauksessa.

holdreg\_x\_releaseoverride:

TRUE: Mikäli syötät holdreg\_x\_releaseoverrideen TRUE-tekstin, pisteen tietoa EI kirjoiteta IMS:lle

mikäli se on käsiohjattuna ja väylältä luettu arvo on eri kuin käsiohjauksipisteessä.

FALSE: Mikäli syötät holdreg\_x\_releaseoverrideen FALSE-tekstin, piste saa olla käsiohjattuna.

ESIMERKKI: Mikäli kirjoitat IMS:n asetusarvoa ohjelman kautta ja piste saa myös olla käsiohjattuna,

asettele holdreg\_x\_force TRUE:ksi ja holdreg\_x\_releaseoverride FALSE:ksi.

\*)

END\_PROGRAM