



Karelia-ammattikorkeakoulu  
Insinööri (AMK)  
Talotekniikka

# Säätöventtiilien ja jäätymisvaaratermostaatin liittäminen kenttäväylään

Joa Lepikkö ja Juho Muikku

Opinnäytetyö, Lokakuu 2022

[www.karelia.fi](http://www.karelia.fi)



**OPINNÄYTETYÖ**  
**Syyskuu 2022**  
**Talotekniikka**

Tikkarinne 9  
80200 JOENSUU  
+358 13 260 600

**Tekijä(t)**

Joa Lepikkö ja Juho Muikku

**Nimeke**

Säätöventtiilien ja jäätymisvaaratermostaatin liittäminen kenttäväylään

**Toimeksiantaja**

Karelia-ammattikorkeakoulu Oy

**Tiivistelmä**

Tässä opinnäytetyössä luotiin kaksi käytännön laboratorioharjoitusta toimeksiantona Karelia-ammattikorkeakoulu Oy:lle. Ensimmäisen harjoituksen aikana tehdään virtausmittauksia, kun venttiileiden asentoa on muutettu TA-Slider toimilaitteella. Työn toisessa harjoituksessa käsitellään säätöventtiilien sekä jäätymisvaaratermostaatin liittämistä Modbus-kenttäväylään. Molempia harjoituksia varten luotiin työskentelyohjeet, joita seuraamalla tulevat ammattikorkeakoulun opiskelijat voivat toistaa tehtävät harjoitukset.

Opinnäytetyön harjoitukset tehtiin Karelia-ammattikorkeakoulu Oy:n tiloissa ja laitteilla. Harjoitusten tekemiseen hyödynnetään IMI Hydronic Engineering Oy:n laitteistoja sekä Distech Controls Oy:n Eclipse-logiikkaa sekä EC-gfxProgram-ohjelmistoa. Opinnäytetyö sisältää väyläprotokollin sekä kenttäväyliin liittyvää yleistä teoretietoa. Opinnäytetyöhön on myös kerätty tietoa laboratorioharjoituksessa käytetyistä laitteista ja niiden käyttötar-koituksista sekä ominaisuuksista.

**Kieli**  
suomi

Sivuja 52  
Liitteet 1  
Liitesivumäärä 1

**Asiasanat**

Modbus, rakennusautomaatio, ohjelmointi, säätöventtiili



**THESIS**  
**September 2022**  
**Degree Programme in Building Services Engineering**

Tikkarinne 9  
80200 JOENSUU  
FINLAND  
+ 358 13 260 600

**Author (s)**  
Joa Lepikkö ja Juho Muikku

**Title**  
Connecting the Control Valves and Frost Protection Thermostat to the Field Bus

**Commissioned by**  
Karelia-ammattikorkeakoulu Oy

**Abstract**

In this thesis, two practical laboratory exercises were created as an assignment for Karelia-ammattikorkeakoulu Oy. In the first practical exercise flow measurements are made when the position of the valves have been changed with the TA-Slider actuator. In the second laboratory exercise the control valves and the frost protection thermostat are connected to the Modbus fieldbus. Working instructions were created for both exercises, which future students at the University of Applied Sciences can follow to repeat the exercises.

The thesis exercises were carried out in the facilities and with the equipment of Karelia-ammattikorkeakoulu Oy. IMI Hydronic Engineering Ltd's hardware and Distech Controls Inc's Eclipse logic and EC-gfxProgram software are used for the exercises. The thesis contains general theoretical knowledge related to bus protocols and fieldbuses. For the thesis, information has also been gathered about the equipment used in the laboratory exercise and their purposes and features.

**Language**  
Finnish

Pages 52  
Appendices 1  
Pages of Appendices 1

**Keywords**  
Modbus, building automation, programming, control valve

## Sisältö

1	Johdanto .....	5
2	Opinnäytetyön tietoperusta .....	6
2.1	OSI- ja TCP/IP mallit.....	6
2.2	Ethernet-kehys.....	9
2.3	Väyläprotokollat ja kenttäväylät .....	10
2.4	RS-485 .....	11
2.5	Modbus.....	12
2.6	BACnet .....	16
3	Työssä käytetyt laitteet .....	18
3.1	TA-Slider 160.....	18
3.2	TA-Slider 160 Modbus/BACnet -liityntä .....	19
3.3	TA-Dongle.....	21
3.4	Säätöventtiili TBV-C.....	22
3.5	TA-Scope.....	23
3.6	Jäätymisvaaratermostaatti JVA 24 .....	24
3.7	Lämpötila-anturit TEP NTC 10 ja TEP PT 1000.....	25
3.8	Eclipse-logiikka .....	26
3.9	EC-gfxProgram-ohjelmisto.....	27
4	Toteutus.....	27
4.1	Bluetooth ja TA-Scope- harjoitus .....	27
4.1.1	Mittaukset .....	28
4.1.2	Verkoston ensimmäinen mittaus.....	28
4.1.3	Verkoston toinen mittaus .....	30
4.1.4	Verkoston kolmas mittaus.....	30
4.1.5	Verkoston neljäs mittaus.....	31
4.2	TA-Sliderin ja jäätymisvaaratermostaatin liittäminen kenttäväylään ..	32
4.2.1	Alkutoimet.....	32
4.2.2	Ohjelmointiblokkien lisääminen.....	33
4.2.3	Tulojen lisääminen.....	33
4.2.4	Lämpötila-ohjelmarivin luominen .....	37
4.2.5	Modbus Devicen lisääminen .....	38
4.2.6	Ohjelmarivin luominen hälytystä varten .....	42
4.2.7	Kytkenät.....	43
4.2.8	TA-Sliderin asetusten muuttaminen.....	45
4.2.9	Valmiin ohjelman lataaminen logiikkaan .....	46
4.2.10	Valmiin ohjelman testaaminen .....	47
5	Pohdinta.....	49
6	Lähteet.....	51

# 1 Johdanto

Tämän opinnäytetyön tavoitteena oli luoda kaksi käytännön laboratorioharjoitusta sekä niihin työhjeet. Harjoitukset luotiin pääasiassa talotekniikan opiskelijoita varten. Kyseessä on toiminnallinen opinnäytetyö, joka saatiin toimeksiantona Karelia-ammattikorkeakoulu Oy:ltä.

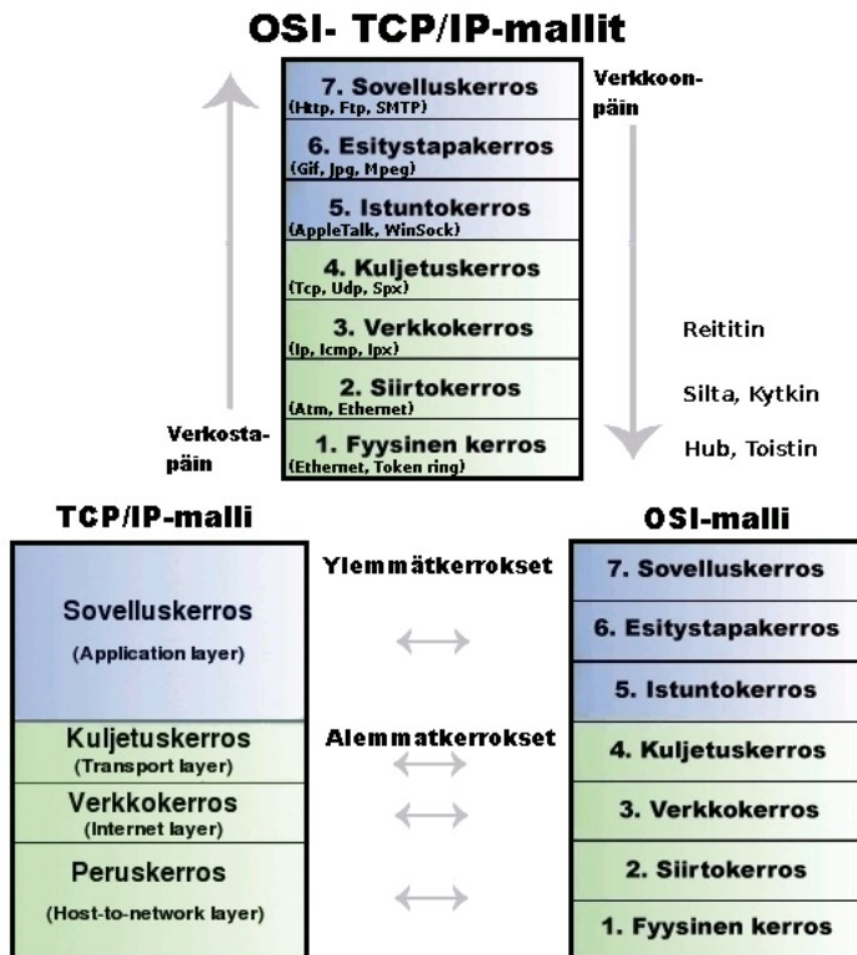
Ensimmäisessä laboratorioharjoituksessa suoritetaan virtausmittauksia TA-Scope mittausvälineistöllä, kun venttiileiden asentoa on muutettu TA-Slider toimilaitteen avulla. Ensimmäisen harjoituksen tavoitteena on saada suoritettua virtausmittaukset harjoitusverkoston kolmesta eri kohdasta. Harjoituksessa myös asetetaan TA-Slidereille asetukset HyTune-sovelluksen avulla, jotta virtausmittaukset voitaisiin suorittaa luotettavasti.

Toisessa laboratorioharjoituksessa liitetään jäätymisvaaratermostaatti ja TBV-C säätöventtiilit Modbus-kenttäväylään TA-Slideria hyödyntäen. Ensimmäisessä harjoituksessa TA-Slidereita ohjataan HyTune-puhelinapplikaation avulla, kun taas toisessa harjoituksessa TA-Slidereita ohjataan Modbus-kenttäväylän kautta. Toisen harjoituksen aikana luodaan uusi ohjelma EC-gfxProgram-sovelluksen avulla. Toisessa harjoituksessa tavoitteena on ohjata TA-Sliderin asentoa jäätymisvaaratermostaatile tulevan hälytystiedon avulla. TA-Slider sekä jäätymisvaaratermostaatti kytketään osaksi Eclipse-logiikkaa, johon ladataan harjoituksen aikana luotu ohjelma. Ohjelmaan määritetään parametrejä, joiden avulla määritetään esimerkiksi TA-Sliderin vakioasento, kun jäätymisvaaratermostaatin hälytys laukeaa.

## 2 Opinnäytetyön tietoperusta

### 2.1 OSI- ja TCP/IP mallit

**OSI-malli (Open Systems Interconnection)** on käsitteellinen viitekehys, jolla visualisoidaan verkko- tai tietoliikennejärjestelmiä seitsemällä eri kerroksella, (kuva 1) jossa jokaisella on oma tehtävänsä [1]. Nämä kerrokset rakentuvat aina edellisen kerroksen päälle, joista ensimmäinen kerros on fyysinen kerros [2].



Kuva 1. OSI- ja TCP/IP- mallit [3].

**Fyysinen kerros** on OSI-viitemallin alin kerros. Se vastaa fyysisen tiedonsiirtoyhteyden muodostamisesta kahden laitteen välille. Fyysisen kerroksen tyyppi määrittää myös järjestelmän topologian. [2.]

**Siirtokerros** huolehtii tiedonsiirrosta fyysistä yhteyttä pitkin pakkaamalla lähetettävän tiedon käytössä olevaan tiedonsiirtokehykseen. Siirtokerros kykenee uudelleenlähettämään datapaketteja virhetilanteissa. Siirtokerros lisää myös datapakettiin lähettäjän ja vastaanottajan fyysiset osoitteet eli MAC-osoitteet. [2.]

**Verkkokerros** vastaa tiedonsiirrosta verkossa olevien isäntälaitteiden välillä. Sen tehtävänä on myös reitittää datapaketti valitsemalla lyhyimmän reitin useista mahdollisista vaihtoehdoista. Verkkokerros lisää lähettäjän sekä vastaanottajan IP-osoitteet datapakettiin. [2.]

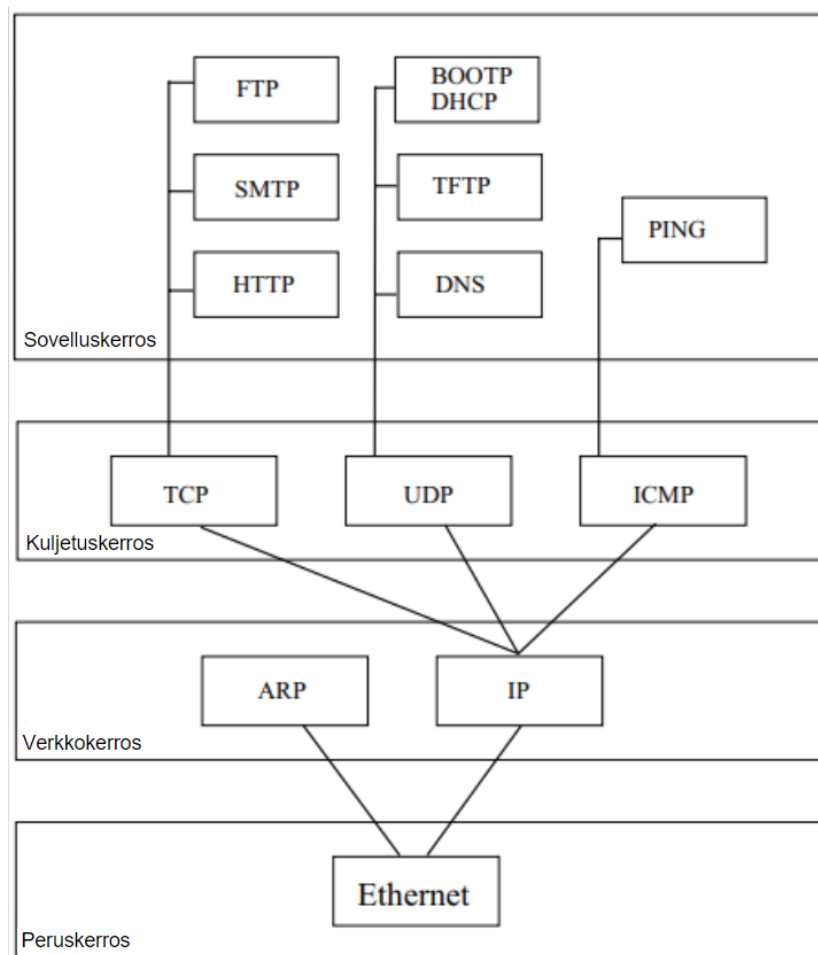
**Kuljetuskerros** vastaanottaa viestin istuntotasolta ja jakaa sen pienempiin osiin. Kuljetuskerros varmistaa, että viesti toimitetaan oikealle prosessille antamalla tämän osoitteen. Vastaanottajan kuljetuskerros kokoaa viestin uudelleen. Segmentit ovat kuljetuskerroksen tietoyksiköitä. Se on vastuussa koko viestin toimittamisesta alusta loppuun. Jos tapahtumassa havaitaan virhe, kuljetuskerros kuittaa tiedonsiirron onnistuneeksi ja lähettää tiedot uudelleen. [2.]

**Istuntokerros** ylläpitää yhteyksiä useiden laitteiden tiedonsiirrossa mahdollisten laitteiden yhdenaikaisen lähettämisen ja vastaanottamisen laitteiden välillä. Lisäksi istuntokerros auttaa synkronoimaan tiedonsiirron havaitsemalla virheitä tiedonsiirrossa lisäämällä "tarkistuspisteitä" siirrettävään dataan vähentäen datan menetystä tiedonsiirron aikana. [2.]

**Esitystapakerros** toimii kääntäjänä sovelluskerrokselle tulossa ja menossa olevalle datalle. Esitystapakerros muuntaa esimerkiksi sovelluskerrokselle saapuvan tiedon binäärikoodista sovelluksen ymmärrettäväksi koodikieleksi kuten ASCII-koodiksi (American Standard Code for Information Interchange) ja toisinpäin. Esitystapakerros muuttaa siis käsittelemänsä tiedon esitystapaa toisen kerroksen tarvitsemaan muotoon. [2.]

**Sovelluskerros** on OSI-mallin seitsemäs ja ylin kerros. Sovellukset luovat dataa, jota pitää lähettää verkon yli. Tämä kerros toimii käyttäjälle ikkunana, jossa sovellukset voivat ottaa yhteyden verkkoon ja näyttää vastaanotetut tiedot käyttäjälle. Sovelluskerroksen avulla käyttäjä voi esimerkiksi muodostaa yhteyden etäisäntään verkon virtuaalisen päätelaitteen kautta. [2.]

**TCP/IP (Transmission Control Protocol / Internet Protocol)** on protokollapaketti, joka tarvitaan laitteiden väliseen kommunikaatioon Ethernet-verkossa. Kuvassa 2 on havainnollistettu TCP/IP-protokollan rakenne. [4.]



Kuva 2. TCP/IP protokolla [5, s. 20].



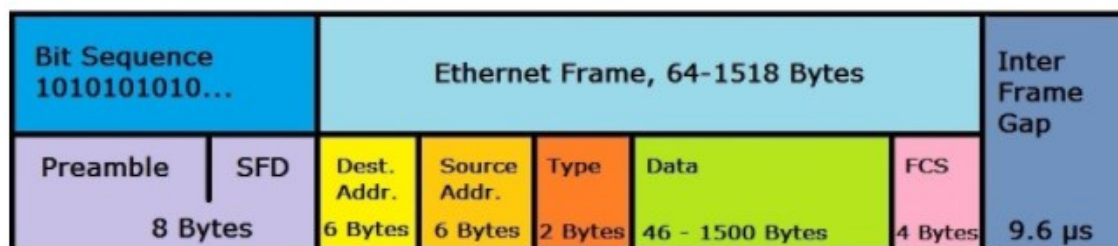
IP määrittää osoitteen laitteelle, joka on riippumaton laitteen sisältämästä MAC-osoitteesta. IP-osoite sisältää esimerkiksi sijaintitietoja ja se mahdollistaa laitteiden viestinnän keskenään. Toimiakseen Internet vaatii keinon erottaa laitteet ja verkkosivut toisistaan, mihin IP-osoitteita hyödynnetään. [6.]

IP muodostaa yksilöllisen 32-bittisen numeron jokaiselle verkon laitteelle. Jokainen internetin kautta lähetetty paketti sisältää sen lähettäjän että vastaanottajan IP-osoitteen. IP tarjoaa viestinnän laitteiden välille erityyppisissä verkoissa kuten Ethernet. Reitityksen tehostamisen vuoksi IP-osoitetta tarkastellaan kahdessa osassa, sen etuliitettä, joka tunnistaa osoitteen fyysisestä verkosta ja päätettä, joka tunnistaa verkossa olevan laitteen. Eli käytännössä IP-osoite on neljän numeron numerosarja esimerkiksi 192.168.1.10. Nämä numerosarjojen numerot ovat väliltä 0-255. [6.]

TCP on tiedonsiirtoprotokolla kahden laitteen väliseen tiedonsiirtoon. TCP-yhteydellä laitteet siirtävät dataa tavuina luotettavasti, sillä TCP protokolla pitää huolen, että lähetetyt paketit saapuvat vastaanottavalle laitteelle oikeassa järjestyksessä ja ilman virheitä. TCP protokolla kykenee huomaamaan virheitä kuten kadonnutta dataa ja on silloin kykenevä lähettämään tiedon uudelleen. Tämä ominaisuus takaa virheettömän tiedonsiirron kahden laitteen välillä. [5, s. 23.]

## **2.2 Ethernet-kehys**

Ethernet-verkossa lähetetty viesti (kuva 3) koostuu 8 tavun kokoisesta alkuosasta (eng. Preamble), Ethernet-kehyksestä (eng. Ethernet Frame) sekä viestien välisestä tauosta (eng. Inter Frame Gap). Alkuosaan kuuluu alkuosa eli Preamble sekä SFD (Start Frame Delimiter). Preamble kertoo vastaanottajalle tulevasta datakehyksestä. [7.] SFD on yhden 1 tavun kokoinen bittisarja, jonka tehtävänä on erottaa alkuosa ja itse Ethernet-kehys toisistaan [8].



Kuva 3. Ethernet-viestin kehysrakenne [7].

Ethernet-kehys alkaa viestin vastaanottajan laitteen yksilöllisellä osoitetiedolla eli kohteen MAC-osoitteella (eng. Destination Address). Tämän jälkeen tulee lähettäjän MAC-osoitetieto (eng. Source Address). [8.] Seuraavaksi viestissä tulee kahden tavun mittainen osio, joka kertoo vastaanottajalle datan pituuden ja tyyppin (eng. Type). Datatyyppin määrittämisen jälkeen tulee viesti eli se data, mitä haluttiin viestillä lähettää. Data voi olla pituudeltaan 46–1500 tavua pitkä. Viestin lopussa on virheentarkistuskoodi FCS (Frame Check Sequence). [7.] FCS:n avulla vastaanottaja saa vahvistuksen siitä, että viesti ei ole korruptoitunut [8]. Ennen seuraavaa viestiä on määritelty viestikehysten välinen tahdistettu tauko eli Inter Frame Gap [7].

### 2.3 Väyläprotokollat ja kenttäväylät

Kiinteistöautomaatiossa kenttäväylä on digitaalinen ja kaksisuuntainen tiedonsiirtotapa erilaisille väyläliitynnöille. Kenttäväyliä hyödynnetään siten, että niiden avulla voidaan yhdistää useita automaation komponentteja toisiinsa. Automaatiojärjestelmään yhdistetyt komponentit saavat jatkuvasti tietoa muilta järjestelmään yhdistetyistä laitteista ja tarkkailevat omaa tilatietoaan, kuten virtausnopeutta tai painetta. Eri komponenttien kaksisuuntaisen digitaalisen tiedonsiirron avulla voidaan ennakoida prosessin tapahtumia. [9, s. 28.]

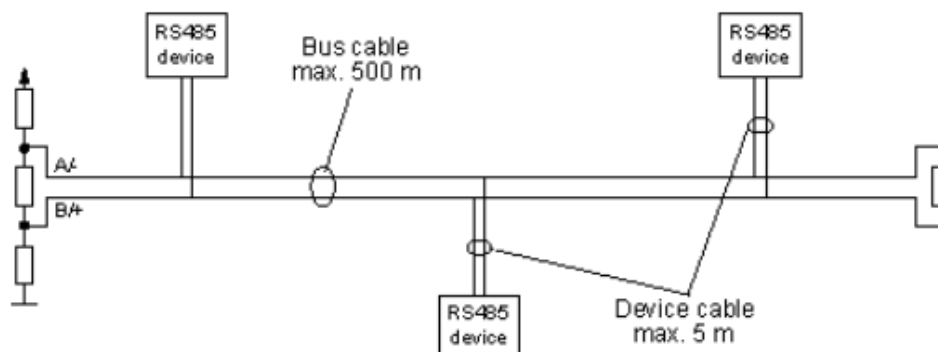
Automaattinen säätö on hyvin yleistä nykyajan automaatiojärjestelmissä aina rakennusautomaatiosta tehdastasolle asti. Automaattisella säädöllä tarkoitetaan järjestelmän jatkuvaan säätöön tarkoitettuja myötä- ja takaisinkytkentöjä.

[9, s. 30.] Automaatiojärjestelmien eri osien välillä kulkee paljon tietoa ja käskyjä eri valmistajien laitteilta toisille. Laitteiden kommunikaation vuoksi niiden on käytettävä samaa viestintäprotokollaa, jotta ne voisivat keskustella toistensa kanssa. Protokolla eli yhteyskäytäntö on standardoitu malli siitä, miten viestit muodostetaan laitteiden välisessä viestinnässä ja miten eri viesteihin kuuluu vastata. [9, s. 39.] Markkinoilla on ollut useita eri käyttötarkoitukseen sopivia väyläratkaisuja, mutta niitä tulee ja lähtee jatkuvasti markkinoilta. Yleisimpiä nykyään rakennusautomaatiossa käytössä olevia tiedonsiirtoprotokollia ovat muun muassa BACnet, Modbus, M-Bus, DALI (Digital Addressable Lighting Interface) sekä KNX. [9, s. 119.]

BACnet-protokolla on erityisesti LVI-tekniikan ohjaamiseen suunniteltu väylä [9, s. 125]. Modbus on avoin tiedonsiirtoprotokolla, jonka vahvuus on sen yhteensopivuus muiden valmistajien laitteiden kanssa [10]. M-Bus on erityisesti mittari- ja anturitietojen lukemiseen kehitetty väyläprotokolla, joka perustuu useisiin eurooppalaisiin standardeihin [9, s. 125]. DALI on valaistuksen ohjaamiseen suunniteltu digitaalinen järjestelmä, jossa jokaisella laitteella on oma yksilöity osoite. DALI-järjestelmään voidaan liittää eri valaisimia valmistajasta riippumatta. [9, s. 147.] KNX on kiinteistöautomaatiostandardi, jonka avulla voidaan ohjata esim. valaistusta, lämmitys- ja jäähdytysjärjestelmiä, kulunvalvontaan liittyviä ratkaisuja sekä energiakulutuksen seurantaa [11].

## **2.4 RS-485**

RS-485 on kaksiparinen johdinjärjestelmä, joka suunniteltiin teollisuuden datansiirtoon, mutta sille löytyy useita sovelluksia alkuperäisen tarkoituksen ulkopuolelta mm. talotekniikasta. RS-485 järjestelmä mahdollistaa aikaisempia järjestelmiä nopeamman tiedonsiirron ja pidempien välimatkojen eron laitteiden välillä. RS-485 järjestelmät voivat olla hyvinkin monimutkaisia, sillä järjestelmään voidaan liittää 32 eri osoitteellista laitetta. RS-485 järjestelmän kytkentäperiaate on esitetty kuvassa 4. [9, s. 49.]

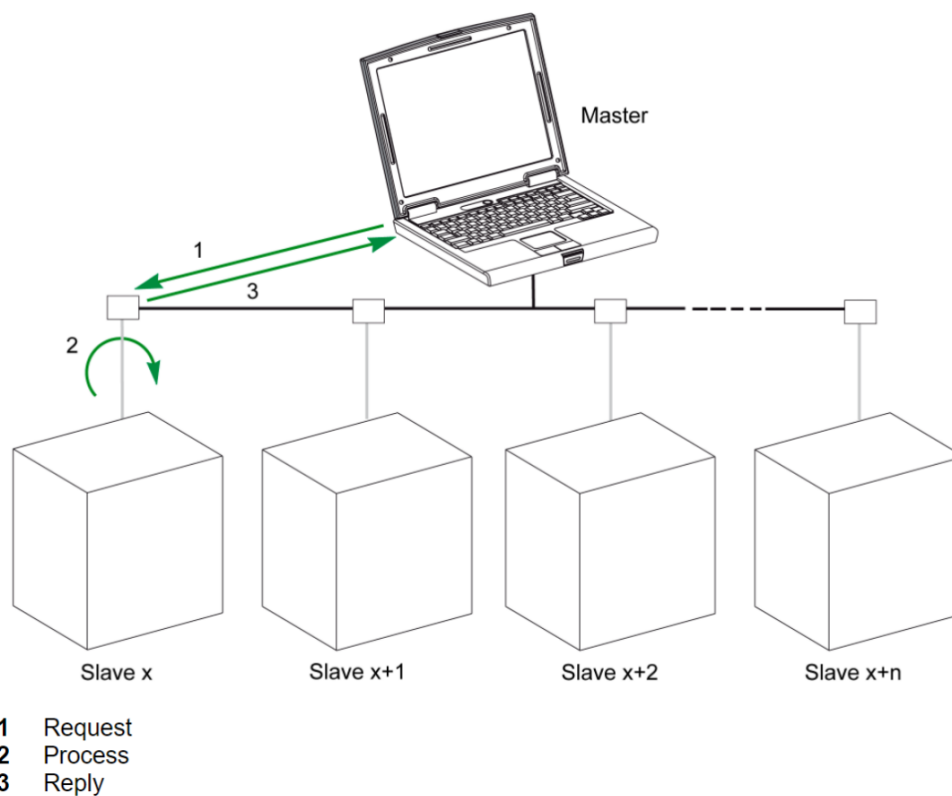


Kuva 4. RS-485 kytkentäperiaate [12].

## 2.5 Modbus

Modbus on avoin tiedonsiirtoprotokolla, joka kehitettiin vuonna 1979 Modiconin toimesta sen ohjelmoitavia logiikkoja varten. Modbusin suosion tärkeimmät tekijät ovat sen avoimuus, yhteensopivuus eri valmistajien laitteiden kanssa sekä sen ilmainen lisenssi. Modbusin yhteensopivuuden ja maksuttomuuden ansiosta siitä on kehittynyt alan standardi. [10.]

Modbus toimii isäntä-orja-protokollalla, jossa yhteen isäntään voidaan kytkeä 247 orjaa [9, s. 140]. Esimerkiksi tällaisessa tilanteessa rakennusautomaation alakeskus lähettää kyselyn orjalaitteelle, esimerkiksi huoneen lämpötilamittarille, joka vastaa takaisin isännälle halutulla tiedolla (kuva 5). Tällä tiedolla alakeskus lähettää tarvittaessa käskyn lämmittää tai jäähdyttää huonetta sitä vastaavalle orjalaitteelle. [9, s. 49.]



Kuva 5. Isäntä-orja toimintaperiaate [13].

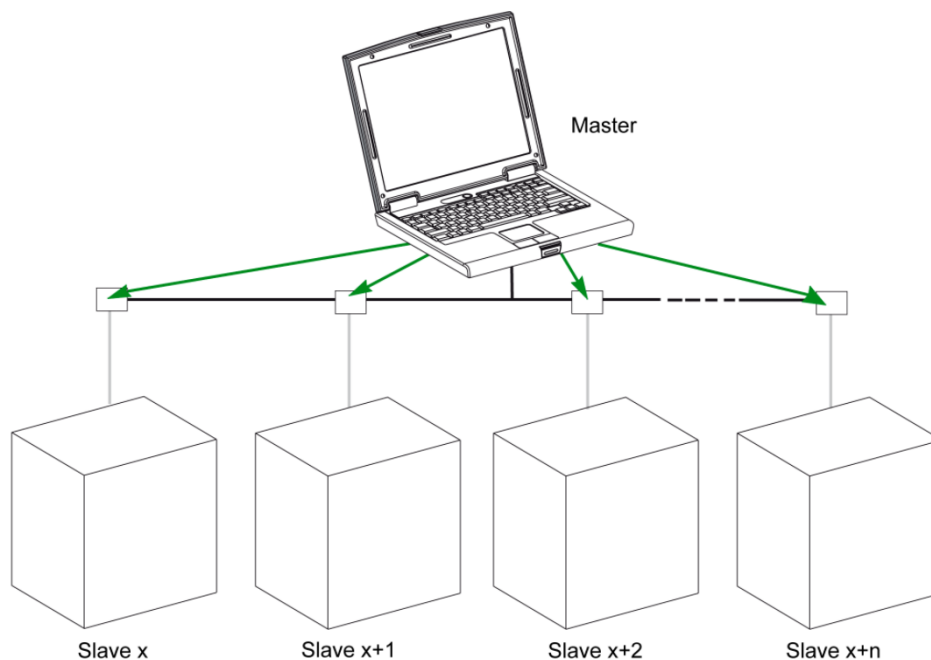
Modbus-protokollaa pystytään käyttämään usean erilaisen fyysisen yhteyden kanssa, ilman että sovelluskerroksen tarvitsee muuttua. Modbus-laitteet voivat käyttää useita eri tiedonsiirtonopeuksia väylässä viestimiseen. Modbus kehyksiä on useita: Modbus RTU (Remote Terminal Unit), Modbus ASCII ja Modbus over TCP/IP. Näistä versioista TCP/IP-versiota käytetään Ethernet-liitännöissä ja muiden perinteisten sarjaväylien kuten RS-485-väylän kanssa. [9, s. 140.]

Modbus RTU-viesti (kuva 6) koostuu aina osoitteesta, toiminnosta, datasta ja tarkistussummasta. Osoite kertoo sen orjalaitteen osoitteen, jolle viesti on tarkoitettu tai joka sen lähettää. Toiminnot ovat orjalaitteelle osoitettuja käskyjä, joita ovat esimerkiksi tiedon lukeminen tai kirjoittaminen. Viestiin kuuluvan datan koko voi olla pituudeltaan maksimissaan 252 tavua. [9, s. 141.]

Aloitus	Osoite	Toiminto	Data	Tarkistus- summa, CRC	Lopetus
vähintään 3,5 tavua pitkä tauko	1 tavu	1 tavu	0–252 tavua	2 tavua	vähintään 3,5 tavua pitkä tauko

Kuva 6. Modbus RTU-viestin perusrakenne [9, s. 140].

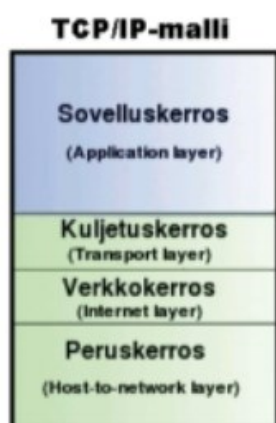
Isännän orjalle lähettämä viesti koostuu esimerkiksi seuraavasti: orjan osoite, toimintona “Lue tietoa”, viestin datana on tarkennus siitä, minkä tiedon isäntä haluaa ja lopuksi tarkistussumma. Orjan vastauksessa osoitteena on orjan oma osoitenumero, jotta isäntä tietää miltä orjalta viesti tulee. Isäntä pystyy lähettämän viestin myös osoitteella 0, jolloin viesti välittyy kaikille orjille. Tällöin kuitenkin orjat eivät vastaa isännälle (kuva 7). [9, s. 141.]



Kuva 7. Yleislähetys isännältä orjille [13].

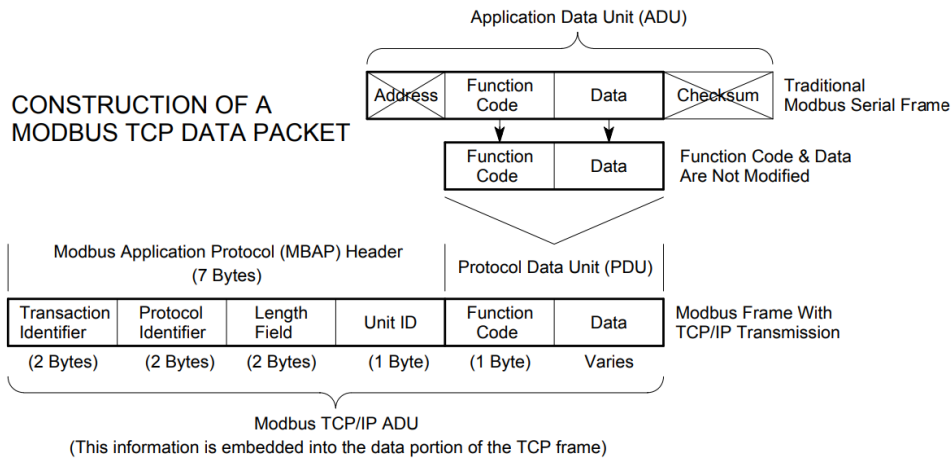
Modbus TCP/IP on käytännössä normaali Modbus RTU protokolla, joka hyödyntää TCP/IP viestintäprotokollaa viestin kuljettamiseen Ethernet-yhteyttä pitkin esim. tietokoneelta toiselle. Modbus RTU:n ja TCP/IP:n ero on siis siinä, minkälaisista väylästä pitkin ne voivat viestiä. [14, s. 4.] Modbus TCP/IP protokolla asettuu TCP/IP-mallin kerroksiin (kuva 8) siten, että Modbus toimii sovelluskerroksella, kun taas TCP ja IP täyttävät kuljetus- sekä verkkokerrokset. Peruskerroksena toimii TCP/IP protokollan käyttämä Ethernet yhteys.

[14, s. 10.]

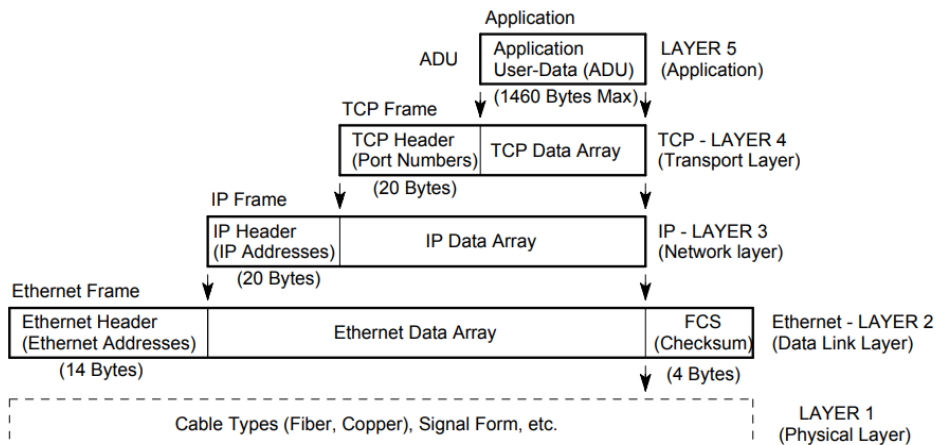


Kuva 8. TCP/IP-mallin kerrokset [3].

Modbus TCP/IP viestiprotokollassa perinteinen Modbus-viesti muokataan TCP/IP protokollan mukaiseksi, jotta viesti voidaan kuljettaa Ethernet-väylää pitkin. Kuvassa 9 näkyy, miten Modbus-viesti muutetaan Modbus TCP/IP ADU:ksi (Application Data Unit). Kuvasta 10 näkyy, miten tähän ADU-osaan liitetään TCP/IP-protokollan käyttämät osat, jotta niistä muodostuu kokonainen Modbus TCP/IP viesti. [14, s. 4.]



Kuva 9. Modbus TCP/IP-datapaketin muodostaminen [14].



Kuva 10. Modbus TCP/IP- viestin rakenne [14].

## 2.6 BACnet

BACnet on avoimen protokollan viestintästandardi, jonka on luonut ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineers). BACnet on kansallinen standardi yli 30 eri maassa, mukaan lukien Eurooppa ja Amerikka. BACnet viestintäprotokollalla on myös maailmanlaajuinen ISO- standardi. [15.] BACnet mahdollistaa eri valmistajien valmistamien tuotteiden yhteensopivuuden rakennusautomaation ohjausjärjestelmissä [16]. BACnet on nykyään yksi käytetyimmistä väyläprotokollista rakennusautomaatiossa [9, s. 125].



Rakennusautomaation laitteet liitetään BACnet-verkkoon, jossa ne mallinnetaan objekteina. BACnet määrittelee vakiojoukon objekteja, jotka koostuvat erilaisista ominaisuuksista. Eri objektityypeille on määritetty oma joukko näitä ominaisuuksia. Ominaisuuksien avulla valvotaan ja ohjataan objektien toimintaa. Laitteessa olevat ominaisuudet määräytyvät objektin tyyppin mukaan. Objekteja ovat esimerkiksi Analog Input, jota käytetään esimerkiksi anturin lukemiseen sekä Schedule, jonka avulla voidaan aikatauluttaa toimintoja. Jokaisella objektilla on BACnet-verkossa oma Object ID eli objektitunnus, joka toimii laitteen osoitteena verkossa. Nämä objektitunnukset ovat 32-bittisiä binäärilukuja, jotka koostuvat objektin tyyppistä ja sen yksilöllisestä numerosta. [16.]

BACnet TCP/IP voi hyödyntää kiinteistön Ethernet-verkkoa. TCP/IP:tä käytettäessä laitteille määritetään oma IP-osoite. BACnetin toiminta perustuu neljään toimintakerrokseen. BACnetin toimintakerroksia ovat OSI-mallin mukaisesti fyysinen-, siirtoyhteys-, verkko- ja sovelluskerros. Nämä neljä toimintakerrosta on esitetty kuvassa 11. BACnet käyttää tiedonsiirtomediana IEEE 802.3 sekä RS-232 ja RS-485- liityntärajapintoihin perustuvia ratkaisuja. RS-485-fyysisellä liittymällä on mahdollista käyttää siirtonopeuksia 9,6 kbps (kilobittiä sekunnissa), 19,2 kbps, 38,4 kbps ja 76,8 kbps. [9, s. 125-128.]

BACnet kerrokset					Vastaavat OSI kerrokset
BACnet sovelluskerros					Sovellus
BACnet verkkokerros					Verkko
ISO 8802-2 (IEEE 802.3)	Tyyppi 1	MS/TP	PTP	LonTALK	Siirtoyhteys
ISO 8802-3 (IEEE 802.3)	ARCNET	EIA-485	EIA-232		Fyysinen

Kuva 11. BACnet-kerrokset ja –rajapinnat [9, s. 127].

Tyypillisen BACnet/IP-paketin koostumus on esitetty kuvassa 12. Kuvaa kannattaa lukea sisältä ulospäin, sillä keskeisin laatikko (BVLL, NPDU, APDU, data) sisältää sen, mitä dataa halutaan lähetettävän. Kuvassa kehyksen osat ovat rajattu laatikoiden sisälle.

BVLL (BACnet Virtual Link Layer) on kokonaisuus, joka sisältää BVLC:n (BACnet Virtual Link Control). BVLC kertoo datan käyttötarkoituksen eli funktion. Datan käyttötarkoitus selventää, onko data NPDU (Network Protocol Data Unit), APDU (Application Protocol Data Unit) vai jokin muu. NPDU on BACnet verkkokerrokselle tarkoitettu datapaketti ja APDU on sovelluskerrokselle tarkoitettu datapaketti (kuva 12). [17, s. 5.]



Kuva 12. BACnet- kehyksen tyypillinen rakenne [17, s. 5].

Ethernet-kenttä sisältää alku- ja lopputiedon (eng. Checksum), mitkä kertovat viestin lukemistavan ja sen, onko viesti tullut kokonaisuena perille. Alkutieto kertoo protokollan, jonka mukaan paketti (eng. Padding) rakentuu. Ethernet- kerros sisältää IPv4+UDP- kerrokset. IPv4+UDP (User Datagram Protocol) sisältää datan vastaanottajan sekä lähettäjän IP-osoitteet. [17, s. 5.]

### 3 Työssä käytetyt laitteet

#### 3.1 TA-Slider 160

TA-Slider 160 (kuva 13) on digitaalisesti konfiguroitava toimilaitte nestepitoisten putkistojen venttiileille. TA-Slider 160:stä on useita versioita, joista tässä työssä tullaan käyttämään Modbus RTU:n sekä Bluetoothin avulla ohjattavaa versiota. TA-Sliderissa on ohjelmoitava tulo, rele sekä erilaisia muokattavia parametreja. Yhdessä nämä ominaisuudet tekevät laitteen soveltuvaksi monenlaisiin erilaisiin järjestelmiin ja tehtäviin, kuten virtaamien tasapainotukseen ja säätöön. [18.]

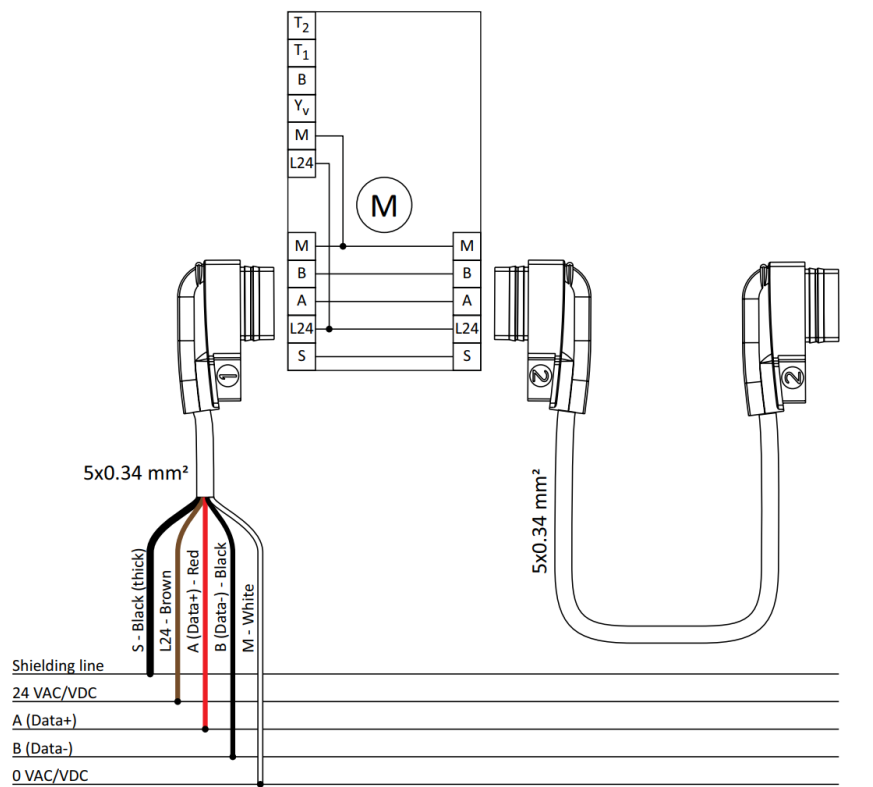


Kuva 13. TA-Slider 160 [18].

TA-Slider 160 sisältää useita ominaisuuksia, joita ovat esimerkiksi suhteellinen säätö, itse säätyvä sulkuvoima sekä venttiilin tukkeutumis- ja juuttumissuoja. Nämä mahdollistavat laitteen pitkän käyttöiän ja varman toiminnan. Lisäksi laitteessa on tietojenkeruuominaisuus sekä turva-asento mahdollisia häiriötilanteita varten. [18.]

### 3.2 TA-Slider 160 Modbus/BACnet -liityntä

TA-Slider liitetään Modbus-väylään laitteen oman BUS-kaapelin avulla. Tämän kaapelin toinen pää yhdistetään Sliderin liitäntään. Kaapelissa olevat avoimet johtimet yhdistetään logiikkaan kytkentäkaavion (kuva 14) mukaisesti. Johtimien selitykset löytyvät taulukosta 1.



Kuva 14. TA-Slider Modbus-kytkentäkaavio [19, s. 6].

Liitântä Kuvaus

S	Suojaus, linja on kytkettävä toisesta päästä erityiseen suojapäätteeseen, joka puolestaan on kytketty maahan.
L24	Virran syöttö 24 VAC/VDC
M	Nollajännite 24 VAC/VDC virransyöttöön ja viesteihin.
A (Data+)	Data+ (RS 485)
B (Data-)	Data- (RS 485)
Yv	Suhteellisen säädön 0(2) -10 VDC, 47 kΩ tuloviesti
B	Potentiaalivapaa liitântä (esim. avoimen ikkunan tunnistus) maks. 100 Ω, maks. 10 m kaapeli tai suojattu kaapeli
T1	Liitântä Pt1000 lämpötila-anturille, liitettäväksi välille T1 ja M. Maks. 10 m kaapelin kokonaispituus toimilaitteen ja anturin välillä.

Taulukko 1. TA-Sliderin johdinmerkinnät [19, s. 6].

### 3.3 TA-Dongle

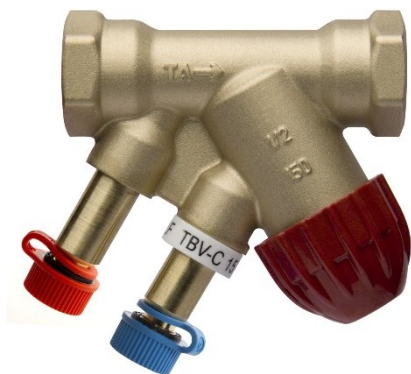
TA-Sliderin asetuksia voidaan muokata TA-Donglen (kuva 15) ja Bluetoothin kautta toimivan älypuhelinsovelluksen HyTune avulla. HyTune-sovelluksen avulla voidaan tehdä laitekohtaisia konfiguraatioita toimilaitteille esimerkiksi muuttamalla niiden laiteosoitteita, kun taas pelkän Donglen avulla voidaan tehdä yksinkertaisia ohjaustoimintoja, kuten pakottamaan Slider auki- tai kiinni- asentoon. HyTune- sovelluksen ja TA-Slidereiden välisen yhteyden luomiseksi vaaditaan TA-Dongle, jonka Bluetooth-yhteyden avulla puhelin ja Slider kommunikoivat keskenään. TA-Dongle yhdistetään Slideriin laitteessa olevan Micro-USB-B-liitännän avulla, jotta tieto siirtyy Donglen ja Sliderin välillä. Donglen akun lataamiseen käytetään Micro-USB-B kaapelia. Donglen akku mahdollistaa Slidereiden asetusten muuttamisen, vaikka niillä ei olisi omaa virtalähdettä. [20.]



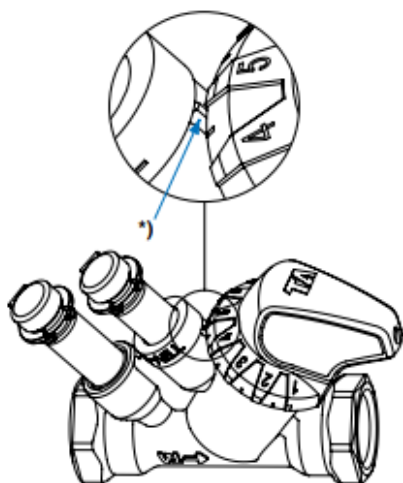
Kuva 15. TA-Dongle [21].

### 3.4 Säätöventtiili TBV-C

TBV-C on IMI Hydronic Engineering Oy:n valmistama säätöventtiili (kuva 16) nestekiertoisiin putkistojärjestelmiin. TBV-C on valmistettu sinkkikatoa kestävästä metalliseoksesta, mikä pidentää sen käyttöikää ja vähentää mahdollisten vuotojen riskiä. TBV-C:n ominaisuuksiin kuuluu virtauksen maksimirajoitus, säätö ja sulkeminen. Nämä säädöt voidaan tehdä käsin esisäätötyökalun (kuva 17) avulla tai ohjattavissa olevan päätelaitteen avulla. Venttiilissä on myös kaksi itsetiivistyvää mittayhdettä, joiden avulla voidaan seurata järjestelmässä kiertävän nesteen tilaa. [28.]



Kuva 16. TBV-C säätöventtiili [28].



Kuva 17. Esisäätötyökalu [28].

### 3.5 TA-Scope

TA-SCOPE (kuva 18) on IMI Hydronic Engineering Oy:n valmistama mittalaite lämmitys- ja jäähdytysjärjestelmien mittauksiin ja tasapainotukseen. Laitteen avulla voidaan mitata virtaamia, paine-eroa, tehoa sekä lämpötilaa, lisäksi mitaustulokset voidaan tallentaa myöhempää käyttöä varten. [29.] Laite sisältää muitakin ominaisuuksia mittaustoimintojen lisäksi, kuten tasapainotus, vianetsintä, tietojen keruu ja nopea mittaus. Lisäksi laitteella voidaan määrittää mitattava neste, mikäli mitattava järjestelmä sisältää muita nesteitä kuin vesi. TA-SCOPE sisältää hydronisen laskimen, jolla voidaan tehdä laskutoimituksia paine-eron, virtaaman, Kv-arvon, lämpötilaeron ja tehon avulla tehtäviin laskutoimituksiin. [30.]



Kuva 18. TA-SCOPE- mittaus ja säätölaite [30].

### 3.6 Jäätymisvaaratermostaatti JVA 24

JVA 24 (kuva 19) on Produal Oy:n valmistama jäätymisvaaratermostaatti, jonka avulla voidaan valvoa ja säätää ilmanvaihtokoneen patterin paluuveden lämpötilaa. Lämpötilaa säätämällä halutaan estää verkoston jäätyminen. JVA 24 toimii siten, että kun lämpötila alittaa säädetyn hälytyspisteen esimerkiksi  $+3^{\circ}\text{C}$  se alkaa korjaamaan säätöventtiilille menevää viestiä. Mikäli lämpötila alittaa asetetun hälytyspisteen, JVA 24 pysäyttää ilmanvaihtokoneen toiminnan ja lähettää hälytyksen. Hälytys tulee kuitata JVA:n kuittauspainikkeesta tai katkaisemalla sen syöttöjännite. [22.]

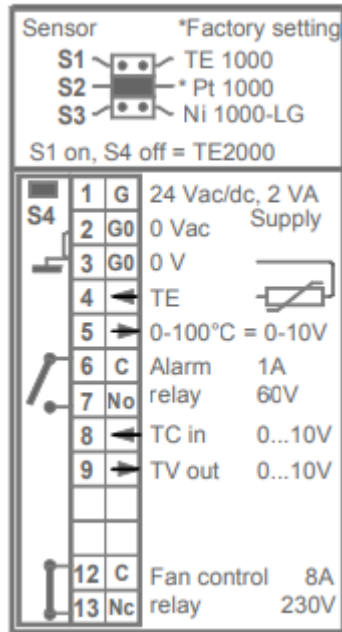


Kuva 19. Jäätymisvaaratermostaatti JVA 24 [22].

Jäätymisvaaratermostaatti JVA 24 (kuva 20) sisältää seuraavat kytkentämahdollisuudet:

- 1-2: laitteen käyttöjännite
- 3-4: lämpötila-anturin vastuksen tilatieto
- 5: logiikalle menevä lämpötilatieto jänniteviestinä
- 6-7: hälytystilatieto
- 8: logiikalta JVA 24:lle tuleva ohjausviesti
- 9: JVA 24:ltä lähtevä ohjausviesti
- 12: puhallinohjaus
- 13: rele





Kuva 20. JVA 24- kytkentäkuva [22].

### 3.7 Lämpötila-anturit TEP NTC 10 ja TEP PT 1000

TEP NTC 10 (Negative Temperature Coefficient) sekä TEP PT 1000 (Positive temperature) ovat Pro dual Oy:n tuottamia pintalämpötila-antureita, jotka on kehitetty LVI-järjestelmien patteriverkostojen lämpötilamittauksiin. Anturien kotelot ovat lämmönkestävää muovia ja ne voidaan asentaa putken pinnalle metallipannan avulla. Antureiden mitta-alue on  $-50^{\circ}\text{C} \dots +120^{\circ}\text{C}$ , mikä mahdollistaa anturin toimimisen putkiston eri osissa ja eri käyttötarkoituksissa. [23. ; 31.]

TEP NTC 10-anturi mittaa lämpötilaa NTC 10-anturielementillä, jonka vastus muuttuu suhteessa lämpötilaan. NTC-10 elementin nimellisvastus  $25^{\circ}\text{C}$  asteen lämpötilassa on  $10\text{k}\Omega$ , kun taas PT 1000 (kuva 21) lämpötila-anturielementin nimellisvastus  $0^{\circ}\text{C}$  asteen lämpötilassa on  $1000\Omega$ . NTC-antureissa lämpötilan noustessa elementin vastus pienenee, kun taas PTC-antureissa (Positive Temperature Coefficient) elementin vastus kasvaa lämpötilan noustessa. [23. ; 31.]



Kuva 21. Lämpötila-anturi TEP PT 1000 [31].

### 3.8 Eclipse-logiikka

ECLYPSE logiikka (kuva 22) on helposti muokattavissa oleva alusta, jota käytetään useiden LVI-sovellusten yhteydessä [24]. Logiikka tukee usean eri protokollan viestintää, kuten BACnet MS/TP, Modbus RTU, Modbus TCP ja M-Bus protokollia. ECLYPSE-logiikka koostuu virtalähde-, ohjaus-, yhteyspalvelin-, automaatio- sekä I/O- laajennusmoduuleista. [25.] Ohjelmoitavan logiikan ominaisuuksiin kuuluu mm. mukautettava ohjauslogiikka, nettipohjainen suunnittelu- ja visuaalinen käyttöliittymä, tapahtumaloki, hälytykset sekä aikataulutukset [24]. RESTful API ominaisuuden avulla mahdollistetaan datan näyttäminen eri applikaatioissa kuten puhelinsovelluksissa tai seurantanäytöissä [25]. Logiikka tukee EC-gfxProgram-ohjelmointityökalua, jonka avulla voidaan luoda mukautettuja ohjauksia ECLYPSE-logiikalle [24].



Kuva 22. ECLYPSE- logiikka [24].

### 3.9 EC-gfxProgram-ohjelmisto

Distech Controlsin EC-gfxProgram-ohjelmisto on graafinen ohjelmointityökalu BACnet ECB- ja LONWORKS ECL-, sekä ECLYPSE sarjan ohjelmitaville logiikoille [26] Ohjelmiston visuaalisten työkalujen avulla voi koota mukautettuja ohjausjaksoja erilaisiin LVI- ja rakennusautomaation tarkoituksiin [27]. Ohjelmiston mukana tulee laaja kirjasto valmiiksi suunniteltuja LVI- sovelluksia, joita hyödyntämällä voi säästää ohjelmointiin kuluvaan aikaan. Ohjelmistossa ECL LONWORKS, ECB BACnet- tai ECLYPSE-sarjan logiikoille luodut ohjelmistot voidaan muuntaa protokollasta tai tuotesarjasta riippumatta uudelleenkäytettäväksi tulevia projekteja varten. [26.]

## 4 Toteutus

Opinnäytetyön aluksi tutustuttiin käytössä olevaan laitteistoon sekä ohjelmistoihin. Laitteiston toimintoja selvitettiin niiden datalehtien avulla ja tutkittiin sitä, miten laitteita voitaisiin käyttää harjoituksissa. TA-Slider toimilaitteen ohjekirjan avulla selvitettiin oikeaoppinen kytkentätapa Modbus-väylälle. Harjoituksia varten tehtiin pieniä testejä, joilla haluttiin kokeilla TA-Sliderin toimintaa. Samalla tutustuttiin HyTune-sovellukseen, jonka avulla pystyttiin asettamaan Modbus-väylään sopivia liikennöintiasetuksia TA-Sliderille. EC-gfxProgram-ohjelmiston käytössä päästiin alkuun Caverionin järjestämän koulutuksen avulla, jossa opetettiin ohjelmiston peruskäyttöä. Tiedonkeruun sekä harjoittelun jälkeen aloitettiin suunnittelemaan varsinaisten laboratorioharjoitusten kulkua.

### 4.1 Bluetooth ja TA-Scope- harjoitus

Ensimmäisessä harjoituksessa tavoitteena on yhdistää Bluetooth-sliderit maksimirajoitusventtiileihin, sekä ohjata niitä HyTune-puhelinsovelluksen ja TA-

Donglen avulla. Harjoituksessa myös mitataan venttiilin virtaaman ja paineen muutoksia TA-Scope mittausvälineistön avulla, kun maksimirajoitusventtiilin asentoa muutetaan TA-Sliderin avulla. Laboratorioharjoitus koostuu neljästä eri vaiheesta, joissa suoritetaan TA-Scope mittaukset verkoston kolmesta eri kohdasta. Maksimirajoitusventtiilin asentoa säädetään TA-Sliderin avulla mittausten välissä. Tällä harjoituksella havainnollistetaan sitä, miten TA-Sliderin asennon muutos vaikuttaa verkoston vesivirtaamiin sekä paine-eroihin.

#### **4.1.1 Mittaukset**

Harjoituksia varten säädettiin maksimirajoitusventtiilit verkoston (LIITE 1) kolmesta eri haarasta esisäätötyökalun avulla asentoon 10, jolloin venttiilit on säädetty täysin auki. Venttiilien säätämisen jälkeen suoritettiin pikamittaukset TA-Scopella harjoitusverkoston kolmesta eri haarasta. Verkoston haarat ovat nimetty niissä olevien linjasäätöventtiilien mukaisesti. Verkoston ylin haara on LSV2, keskimäinen LSV3 sekä alin LSV4.

#### **4.1.2 Verkoston ensimmäinen mittaus**

Ensimmäisessä vaiheessa valitaan TA-Scopesta pikamittaus-toiminto ja asetettiin siihen seuraavat asetukset mittauksia varten (kuva 23). Kiinnitettiin TA-Scopen anturiyksikkö verkoston alimmassa haarassa olevaan maksimirajoitusventtiiliin, sen mittapäätteen värien mukaisesti (kuva 24). Aloitimme pikamittauksen, jolloin näyttöön ilmestyi mitattu virtaama ja paine-ero. Mittauksien jälkeen kirjattiin ylös mitatut arvot, jonka jälkeen mittaus toistettiin verkoston kahdelle muulle haaralle.



Kuva 23. TA-Scopen asetetut venttiilin asetukset



Kuva 24. TA-Scopen anturiyksikkö kytkettynä venttiiliin.

### 4.1.3 Verkoston toinen mittaus

Harjoituksen toisessa vaiheessa tavoitteena on yhdistää TA-Slider sekä TA-Dongle toisiinsa ja ohjata TA-Slider-toimilaitetta Bluetooth-yhteydellä HyTune-sovelluksessa. Harjoituksessa TA-Slider-toimilaite on kiinnitettyä maksimirajoitusventtiiliin. Harjoituksessa suoritetaan samat pikamittaukset kuin ensimmäisessä osiossa. TA-Dongle yhdistettiin TA-Sliderin kanssa, jotka sitten yhdistetään HyTune-sovellukseen Bluetooth-yhteyden avulla. Puhelinsovelluksen avulla määritetään venttiilin tyyppi. Sovelluksessa pystyttiin seuraamaan TA-Sliderin asentoa väliltä 0-10. Sovelluksessa ajetaan TA-Slider haluttuun kierroslukumäärään, jonka jälkeen suoritetaan pikamittaukset uudelleen.

Kun suoritetaan mittauksia pitää muistaa asettaa TA-Scopeen eri kierrosmäärät venttiileille. Kaksi ylempää haaraa (LIITE 1) pysyy samassa kierrosmäärässä kuin aikaisemmin eli asennossa 10, mutta alimman haaran venttiilistä tehtävään mittaukseen tulee asettaa sovelluksessa näkyvä asento, johon TA-Slider on ajettuna. Tässä harjoituksessa se ajettiin asentoon 5/10. Mittauksien jälkeen kirjattiin ylös saadut mittaustulokset kolmesta eri verkoston haarasta.

### 4.1.4 Verkoston kolmas mittaus

Harjoituksen kolmannessa vaiheessa tarkastellaan virtaamien ja paine-erojen muutoksia suhteessa edellisiin mittauksiin, kun kaksi TA-Slideria on ajettuna noin asentoon 5/10. Kolmannessa vaiheessa suoritetaan samat pikamittaukset kuin edellisissä osiossa.

Irrotettiin TA-Dongle (jätetään TA-Slider paikoilleen) ja kiinnitettiin se verkoston keskimmäisen haaran TA-Slideriin. Seuraavaksi määritettiin venttiilin tyyppi oikeaksi samalla tavalla kuin harjoituksessa kaksi. Kiinnitettiin TA-Slider venttiiliin

samalla tavalla kuin vaiheessa kaksi (TA-Slider tulee olla ajettuna täysin auki- asentoon 10/10). Ajettiin TA-Slider asentoon 5/10. Suoritettiin pikamittaukset uudelleen venttiileille LSV2,3,4. Asetettiin TA-Scopeen kierrosmäärät seuraavasti: LSV2 10. kierrosta, LSV3 asennoksi 5. kierrosta sekä LSV4:lle se asento, mihin TA-Slider jätettiin edellisessä harjoituksessa eli 5. kierrosta. Mittauksien jälkeen kirjattiin saadut tulokset ylös.

#### 4.1.5 Verkoston neljäs mittaus

Harjoituksen neljännessä vaiheessa tarkastellaan virtaamien ja paine-erojen muutoksia suhteessa edellisiin mittauksiin, kun kaikki kolme TA-Slideria on ajettuna asentoon 5/10. Neljännessä vaiheessa suoritetaan samat pikamittaukset kuin edellisissä osiossa. Harjoituksen aikana kaikki kolme TA-Slideria oli kiinnitetty LSV2, LSV3 ja LSV4 maksimirajoitusventtiileihin. TA-Sliderit ajettiin HyTune-sovelluksen avulla asentoon 5/10, jonka jälkeen jokaisesta venttiilistä suoritettiin pikamittaus TA-Scopella. Mittauksien jälkeen kirjattiin saadut tulokset ylös. Saaduista mittaustuloksista (taulukko 2) voitiin seurata verkoston virtauksen ja paine-erojen muutoksia.

#### Mittaustulokset:

Mittauskerta:	LSV2 ylin	LSV3 keskimäinen	LSV4 alin
1. Kaikki asennossa 10/10	200 l/h 1250 Pa	250 l/h 1900 Pa	255 l/h 2000 Pa
2. LSV4 asennossa 5/10	207 l/h 1330 Pa	250 l/h 2000 Pa	88 l/h 2200 Pa
3. LSV3 ja 4 asennossa 5/10	207 l/h 1340 Pa	85 l/h 1950 Pa	90 l/h 2300 Pa
4. Kaikki asennossa 5/10	70 l/h 1400 Pa	88 l/h 2150 Pa	90 l/h 2250 Pa

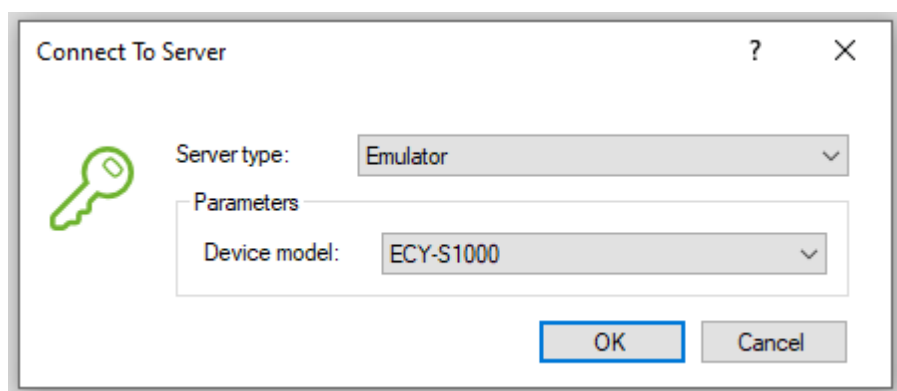
Taulukko 2. TA-Scope mittaustulokset

## 4.2 TA-Sliderin ja jäätymisvaaratermostaatin liittäminen kenttäväylään

Toisessa harjoituksessa tavoitteena on liittää TA-Slider 160 (Modbus) sekä jäätymisvaaratermostaatti JVA 24 Modbus- kenttäväylään. Työn aikana luodaan ohjelma, jossa TA-Sliderin toimintaa ohjataan jäätymisvaaratermostaatilet saattavan lämpötila- ja hälytystiedon avulla. Työssä hyödynnetään Caverionin Eclipse-logiikkaboksia sekä logiikalle suunniteltua EC-gfxProgram-ohjelmistoa. Harjoituksen aikana tehdään tarvittavat kytkennät laitteiden ja logiikan välille. Harjoituksen tavoitteena on kytkeä TA-Slider Modbus-kenttäväylään ja ohjata sen toimintaa jäätymisvaaratermostaatin tilatiedon avulla.

### 4.2.1 Alkutoimet

Avattiin tietokoneelta ohjelmisto EC-gfxProgram. Tällöin aukesi Connect to Server ikkuna (kuva 25), josta valittiin "Server type" valikosta "Emulator" sekä "Device model" "ECY-S1000". Nyt voitiin luoda uutta ohjelmistoa, vaikka logiikkaa ei ollut vielä kytketty.



Kuva 25. Connect to Server "Emulator"



Seuraavaksi aukesi "Select Device" ikkuna, josta valittiin "ECY-S1000". Laitteen valitsemisen jälkeen painettiin "OK". Ohjelman aukeamisen jälkeen avattiin vasemmasta yläkulmasta "File" ja valittiin "New Project". Tämän jälkeen uudelleennimettiin "New Programming Sheet" välilehti "Rename" työkalulla. Välilehden nimeksi annettiin "Common".

#### 4.2.2 Ohjelmointiblokkien lisääminen

Seuraavaksi lisättiin edellisessä vaiheessa luodulle välilehdelle "project\_base" ohjelmointilaatikko. Tämän jälkeen luotiin uusi välilehti, johon lisättiin MIO\_LABRA ohjelmointilaatikko. Seuraavaksi käytettiin "Replace Object" komentoa, jonka avulla muutimme tiedostossa olevat MIOx tekstit MIO1:een.

Seuraavaksi lisättiin uusi välilehti "Project Explorer" valikosta. Annoimme välilehdelle nimeksi "Harjoitus 1". Tämän jälkeen välilehdelle lisättiin "CustomBlock\_library\_model" blokki. Vaihdettiin Custom Blockin nimi Properites valikosta "Name" kohdasta. Annettiin blokille nimeksi "jaatymissuoja"

#### 4.2.3 Tulojen lisääminen

Tässä vaiheessa lisättiin ohjelmointialueelle lämpötilalle sekä hälytystiedolle tarvittavat tulot. Hälytystiedolle asetettiin sen tarvitsemat parametrit. Lisäksi ohjelmassa luodut digitaaliset sekä analogiset tulot kytkettiin logiikan fyysisiin tulo-pisteisiin. Tässä vaiheessa myös poistettiin ylimääräiseksi jäävät tulot. Seuraavaksi lisättiin "Analog Value"-laatikko, johon asetettiin kuvan 26 mukaiset tiedot.

The image shows a configuration window for a temperature object. It is divided into several sections:

- General:** Object name: Lampotila; Description: lampotilatieto; Controller specific: ; Persist value 16: .
- BACnet Properties:** Object type: Analog Value; Object instance: 1; Broadcast mode: None; COV min send time: 5 s; COV max send time: 300 s; COV increment: 1 %; Units: %; Default value: 0 %.
- Metadata:** No element +.
- Alarm:** Enabled: ; Inhibit binary value: None; Limits enabled: High limit:  Low limit: .

At the bottom right, there are buttons for 'OK' and 'Cancel'.

Kuva 26. Lämpötila- objekti

Seuraavaksi lisättiin "Binary Value" laatikko hakemalla se Toolboxista samalla tavalla kuin aikaisemmin. Tuplaklikattiin lisätty laatikko auki ja tehtiin sille kuvan 27 mukaiset muokkaukset. Seuraavaksi painettiin sinisestä Configure linkistä ja asetettiin alla olevan kuvan 28 tiedot.

The screenshot shows a configuration window with three main sections:

- General:**
  - Object name: halytys
  - Description: jaatymisvaara
  - Controller specific:
  - Persist value 16:
  - Persist binary statistics:
- BACnet Properties:**
  - Object type: Binary Value
  - Object instance: 3
  - Broadcast mode: None
  - COV min send time: 5 s
  - COV max send time: 300 s
  - Default value: OK (with a [Configure...](#) link)
  - Minimum on time: 0 s
  - Minimum off time: 0 s
- Metadata:**
  - No element

Kuva 27. Hälytys- objekti

The screenshot shows the 'halytys Format Editor' dialog box with the following settings:

- Format type: Boolean
- Parameters:
  - Quick select: (empty dropdown)
  - True text: HÄLYTYS
  - False text: OK
- Buttons: OK, Cancel

Kuva 28. Format Editor-ikkuna

Muokattiin seuraavat tiedot Alarm-osiossa (kuva 29). Hälytys-toiminnolla voitiin määrittellä hälytysviestit tarkemmin, jotta hälytyksestä saadaan tarkempi tieto sen aiheuttajasta. Message-tiedot saatiin määritettyä nopeasti oikein, kun klikattiin "Alarm" otsikon tasalla olevasta kolmesta pisteestä. Aukeavasta valikosta valittiin "Bind to Existing Configuration Variable Group". Auenneesta ikkunasta

valittiin "Alarm", josta "Binary\_Alm". Lopuksi painettiin ok. Lisättyä hälytystoimintoa hyödynnettiin myöhemmässä vaiheessa harjoitusta.

**Alarm**

Enabled:

Inhibit binary value: None

Alarm value: HÄLYTYS

Time delay: 0 s [0 ... 65 535]

Time delay (Normal): 0 s [0 ... 65 535]

Notify type: Alarm

Events enabled: To offnormal:  To fault:  To normal:

To offnormal message: {Description} hälytys Insert variable...

To fault message: {Description} laitevika Insert variable...

To normal message: {Description} hälytys Insert variable...

Notify class: None + Configure...

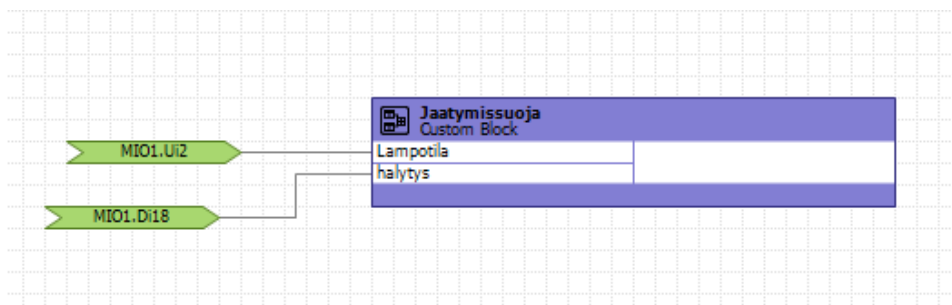
**Trending**

Generate trend logs:

OK Cancel

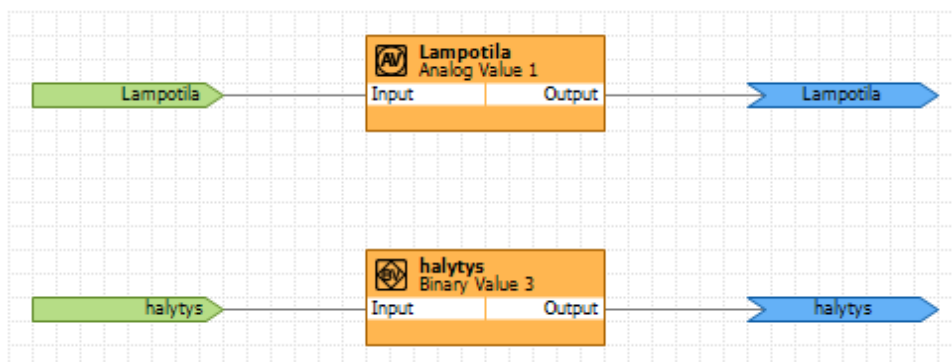
Kuva 29. Hälytys-tiedot

Klikattiin hiiren oikealla painikkeella "Lampotila" Analog Value laatikon input-sanan päällä. Tällöin aukesi valikko, josta valittiin "Link To" komento, jonka avulla määritettiin logiikan tulo ohjelmaan "Exported Input". Tällöin ilmestyi vihreä nuoli linkattuna Lampotila-laatikkoon. Tämä toimenpide toistettiin alemmalle "halytys" Binary Value laatikolle. Tämän jälkeen palattiin harjoitus 1 välilehdelle, jossa lisäsimme lämpotila ja hälytys tulot jäätymissuojan "Custom Blockiin". Nyt nämä ohjelman tulot vastasivat logiikan analogista (Ui2) ja binääristä (Di18) tuloa. Logiikan kytkennät löytyvät myöhemmästä osiosta "Kytkenät". Ohjelmablokkiin oli nyt määritelty tulot kuvan 30 mukaisesti.



Kuva 30. Jäätymissuojan tulot

Tämän jälkeen palattiin “Harjoitus1/Jaatymissuoja” välilehdelle. Valittiin lampotila-laatikon “Output” kohdan päällä hiiren oikealla ”Link To” ja kohteeksi ”Reference Hub”. Sama toimenpide toistettiin halytys-laatikon “Output” kohdalle. Input laatikot näyttivät nyt kuvan 31 mukaiselta.

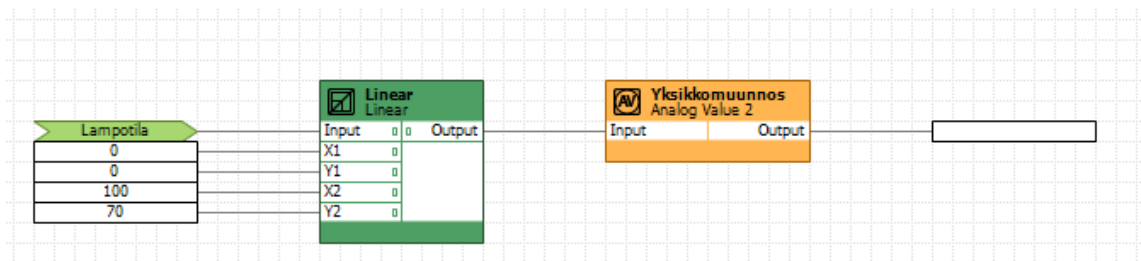


Kuva 31. Input laatikot

#### 4.2.4 Lämpötila-ohjelmarivin luominen

Tässä vaiheessa lisättiin lineaarinen muutostyökalu, jonka avulla muunnetaan jäätymisvaaratermostautilta saatu lämpötilatieto. Näin saatiin muunnettua lämpötilatieto jänniteviestistä luettavaan muotoon asteiksi. Lämpötilaa pystyttiin seuraamaan reaaliajassa monitorointilaatikon avulla.

Ohjelmarivin luominen aloitettiin lisäämällä "Linear" blokki Toolboxista. Seuraavaksi valittiin laatikolle tulo eli Input, joka yhdistettiin "Link To" komennolla "Reference Target" lämpötilatietoon "Lampotila". Seuraavaksi lisättiin Linear-laatikolle muuttujien X ja Y- arvot. Linear työkalun muuttujat skaalaavat arvot lämpötila-anturin mittavälille, muuttaen jänniteviestin lämpötilatiedoksi. Valittiin "Linear" laatikolle lähtö eli "Output", joka yhdistettiin "Link To" komennolla "Analog Value" laatikkoon. Analog Value laatikolle annettiin nimeksi "Yksikkomuunnos". Seuraavaksi vielä vaihdettiin "Units" kohdasta "Temperature" celsiusasteiksi °C. Näin saatiin Yksikkomuunnos-laatikon lämpötilatieto näkymään celsiusasteina. Lopuksi lisätään ohjelmarivin loppuun monitorointilaatikko, jonka avulla lämpötilan muuttumista on helppo seurata. Valmis ohjelmarivi näytti kuvan 32 mukaiselta.



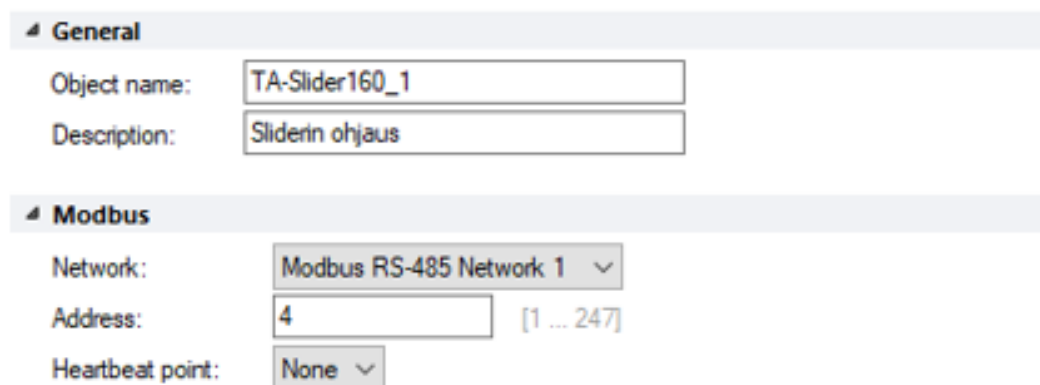
Kuva 32. Lämpötila-ohjelmarivi

#### 4.2.5 Modbus Devicen lisääminen

Tässä vaiheessa luotiin Modbus Device-ohjelmointiblokki, jonka avulla voidaan ohjata Modbus-kenttäväylässä olevia laitteita. Modbus Device-ohjelmointiblokkille määritetään Modbus Pointit hyödyntäen TA-Slider 160 laitteen Modbus RTU kirjastoa (Modbus holding registers).

Ensimmäiseksi valittiin "Toolbox" ja haettiin ModbusDevice. Hakemalla löytyi yksi ModbusDevice-blokki, joka lisättiin Logic-otsikon ohjelmointialueelle

“Harjoitus 1/MyCustomBlock” välilehdelle. Avattiin Modbus Device blokki, jolloin aukesi “Resources Configuration” ikkuna. Tässä ikkunassa asetettiin “Network” Modbus RS-485 Network 1:ksi. Laitteelle annettiin osoite 4. Osoitteet voitiin asettaa välille 2-247, koska 1 on jo logiikan käytössä. Asetettiin “MD2: Modbus Device 2” asetukset kuvan 33 mukaisesti.



General	
Object name:	TA-Slider160_1
Description:	Sliderin ohjaus

Modbus	
Network:	Modbus RS-485 Network 1
Address:	4 [1 ... 247]
Heartbeat point:	None

Kuva 33. MD2: Modbus Device 2

Seuraavaksi valittiin “MD2” osiosta “Points” valikko. Lisättiin uusi “Modbus Point” TA-Slider toimilaitteelle. Seuraavaksi muutettiin Modbus Pointin asetukset kuvan 34 mukaiseksi käyttäen apuna valmistajan verkkosivulta löytyvää TA-Slider 160 rekisteridokumenttia (TA-Slider 160/500 Modbus RTU). “InPos” toiminnon avulla TA-Sliderille voidaan määrittää tavoiteasento, johon se pyrkii ajamaan itsensä.

<b>General</b>	
Name:	<input type="text" value="InPos"/>
<b>Address</b>	
Address type:	<input type="text" value="Decimal"/>
Address:	<input type="text" value="0"/> [0 ... 65 535]
<b>Type Definition</b>	
Register type:	<input type="text" value="Analog Output Holding Register"/>
Data type:	<input type="text" value="Unsigned 16 bits"/>
Access type:	<input type="text" value="Read Write"/>
Default value:	<input type="text" value="0"/> [0 ... 65 535]
<b>Timing</b>	
Update interval:	<input type="text" value="10"/> s

Kuva 34. InPos- ikkuna

Kun ensimmäinen Modbus Point saatiin valmiiksi niin seuraavaksi lisättiin uusi Modbus Point samalla tavalla kuin aikaisemmin ja muokattiin sen tiedot kuvan 35 mukaisesti. CurrentPos toiminto kertoo käyttäjälle TA-Sliderin nykyisen asennon. Asetettu "Address" 12 saatiin TA-Slider 160 Modbus RTU kirjastosta.



The image shows a configuration window for a Modbus Point named "CurrentPos". The window is divided into several sections:

- General:** Name: CurrentPos
- Address:** Address type: Decimal, Address: 12 [0 ... 65 535]
- Type Definition:** Register type: Analog Output Holding Register, Data type: Unsigned 16 bits, Data Units: no-unit, Access type: Read Only, Default value: 0 [0 ... 65 535]
- Timing:** Update interval: 10 s
- Programming Value:** Units: no-unit, Scale: 1, Offset: 0

Kuva 35. CurrentPos- ikkuna

Seuraavaksi luotiin uusi Modbus Point samalle laitteelle. Asetettiin Modbus Pointin nimeksi SerNro. Etsittiin Modbus RTU dokumentista MBSN (Modbus Serial Number) kohta, josta löytyy asetettava osoite 10, joka asetettiin "Address" kohtaan. Lisäsimme loput tiedot kuvan 36 mukaisesti. Kun Modbus pointit olivat valmiit, suljimme muokkausikkunan painamalla "OK".

<b>General</b>	
Name:	SerNro
<b>Address</b>	
Address type:	Decimal
Address:	10 [0 ... 65 535]
<b>Type Definition</b>	
Register type:	Analog Output Holding Register
Data type:	Single Precision Floating Point
Data Units:	no-unit
Access type:	Read Only
Default value:	0
<b>Timing</b>	
Update interval:	10 s
<b>Programming Value</b>	
Units:	no-unit
Scale:	1
Offset:	0
<b>Resource Binding</b>	
Enabled:	<input type="checkbox"/>
Resource type:	
Resource instance:	Configure...

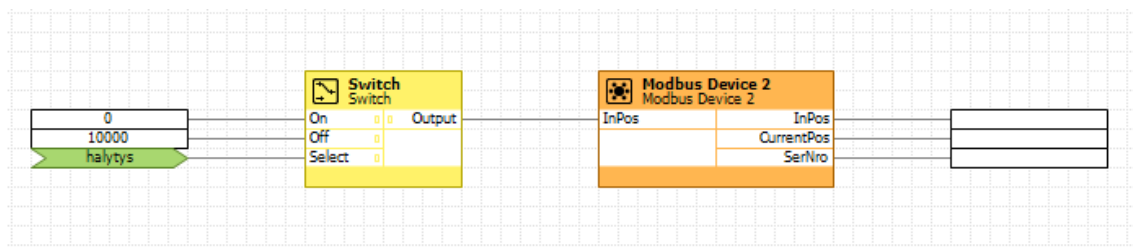
Kuva 36. SerNro- ikkuna

#### 4.2.6 Ohjelmarivin luominen hälytystä varten

Tässä vaiheessa luodaan "Switch" laatikko, joka yhdistetään aikaisemmin luotuun Modbus Device-laatikkoon. "Switch" laatikon avulla voidaan määritellä mitä numeraalisia arvoja jäätymisvaaratermostautilta tulevan hälytystiedon ON ja OFF tilat vastaavat. Hälytystiedon muuttaminen numeroiksi on tärkeää, sillä TA-Sliderin asentoa voidaan ohjata numeraalisten arvojen avulla väliltä 0 - 10 000.

Ensimmäiseksi haettiin Toolboxista "Switch" laatikko ja raahattiin se luodun Modbus Device laatikon vasemmalle puolelle. Tämän jälkeen lisättiin Switch-

laatikon tuloille “On” ja “Off” arvot Internal Constant toiminnolla. Arvoiksi toiminnolle asetettiin “On” 0 ja “Off” 10 000. Lopuksi asetettiin “Select” toiminnolle tulo yhdistämällä se “Link To” toiminnolla hälytystietoon. Seuraavaksi lisättiin “Switch” laatikolle lähdeksi “Modbus Device 2”. Lopuksi Modbus Device-laatikolle lisättiin “Monitor All” toiminto. Valmis ohjelmarivi näkyy kuvassa 37.



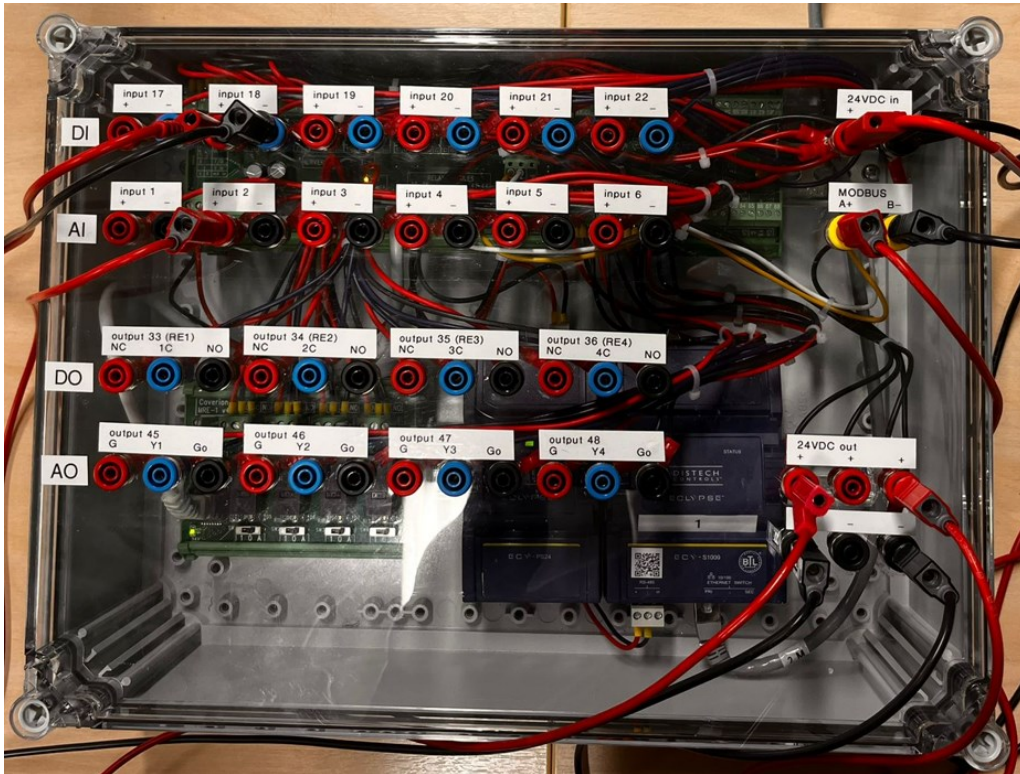
Kuva 37. Modbus- ohjelmarivi

#### 4.2.7 Kytkennät

Caverionin logiikan komponentit ovat koteloitu valmiiksi laboratorioskäyttöä varten. Logiikkaan on nimetty valmiiksi tulot ja lähdöt käytön helpottamista varten. Lisäksi näihin on lisätty liittimet, joiden avulla voidaan tehdä kytkentämuutoksia nopeasti. Logiikan kotelointi on näkyvässä kuvassa 38.

TA-Sliderin kytkennät logiikkaan tehtiin seuraavalla tavalla:

- TA-Sliderin vasempaan liittäntäporttiin yhdistettiin A-tyyppin BUS- kaapeli.
- BUS-kaapelin johtimet liitettiin WAGO- liittimien avulla logiikkaan johtimien värien perusteella.
- ruskea johdin logiikan 24 VDC out + kohtaan.
- valkea johdin logiikan 24 VDC out – kohtaan.
- punainen johdin MODBUS A+ kohtaan.
- musta johdin MODBUS B- kohtaan.

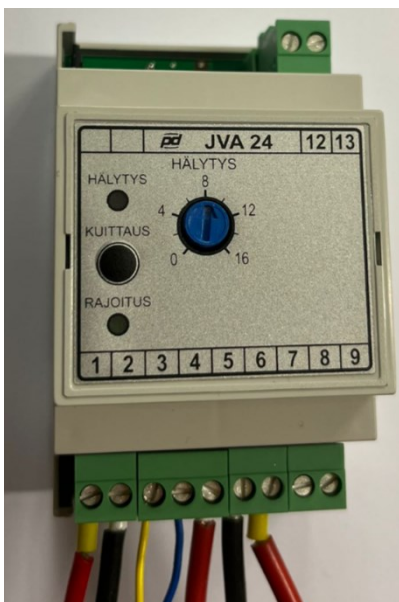


Kuva 38. Logiikkaboksin kytkennät

JVA 24:ssa on numeroitu portit 1-9, jotka näkyvät laitteen etupuolella vasemmalta oikealle lukien (kuva 39). JVA 24:n oikealla kyljellä on myös tarra, jossa on tarkemmin selitetty porttien tehtävä (esim. Lämpötilatieto). Kun johdotukset olivat valmiit niin seuraavaksi yhdistettiin logiikan Ethernet-kaapeli koulun “Experimental” verkkoon. Lopuksi kytkettiin logiikan virtajohto 24 VDC in + kohtaan punainen johdin ja 24 VDC in – kohtaan musta johdin.

JVA 24:n kytkeminen logiikkaan sekä lämpötila-anturiin toteutettiin seuraavasti:

- portti 1 logiikan 24 VDC out + kohtaan.
- portti 2 logiikan 24 VDC out – kohtaan.
- portti 3 lämpötila-anturin TEK PT 1000:n keltainen johdin.
- portti 4 lämpötila-anturin TEK PT 1000:n sininen johdin.
- portti 5 logiikan AI input 2 + kohtaan.
- portti 6 DI input 18 – kohtaan.
- portti 7 DI input 18 + kohtaan.



Kuva 39. JVA 24- kytkennät

#### 4.2.8 TA-Sliderin asetusten muuttaminen

TA-Sliderin asetusten muuttamiseen tarvittiin TA-Dongle, TA-Slider, HyTune sovellus sekä USB-microB liitin. TA-Dongle sekä HyTune sovellus kommunikoivat Bluetooth-yhteyden kautta. Sovelluksen avauduttua etsittiin laitteita, jolloin löytyi TA-Dongle nimellä DGL13AD. Seuraavaksi valittiin löytynyt laite painamalla löytyneen laitteen kuvasta, jolloin aukesi laitteen tiedot, joita voitiin applikaatiossa muokata. Ensimmäiseksi mentiin valikossa alaspäin ja avattiin asetuksista "Väylä" Modbus RTU. Tästä valikosta löytyvät seuraavat muokattavat tiedonsiirtoasetukset: Siirtonopeus, Pariteetti, Slave ID ym. TA-Sliderille pitää asettaa samat tiedonsiirtoasetukset kuin Eclipse-logiikalle, jotta ne voisivat kommunikoida keskenään ongelmitta. Kommunikaatioasetusten jälkeen määritettiin TA-Sliderille venttiilin tyyppi ja malli sekä iskunpituus.

TA-Sliderin liikennöintiasetuksista määritettiin siirtonopeudeksi (baudinopeus) 38400 Bd/s. Pariteetiksi asetettiin "Ei mitään" eli None. Slave ID:ksi asetettiin

sama osoite mikä aiemmin määritettiin "Address" kohtaan EC-gfxProgram-ohjelmassa. Mikäli tässä vaiheessa asettaa väärän osoitteen, tieto ei siirry oikealle laitteelle.

Asetusten muokkaamisen jälkeen siirryttiin edelliseen valikkoon, jossa siirryttiin alaspäin "Venttiili"-valikkoon, josta aktivoitiin "IMI hydronicilta" asetus päälle. Tämän jälkeen asetettiin venttiin malliksi TA-Modulator ja venttiiliksi TA-Modulator 15. Seuraavaksi kohdasta "Iskunpituus" laitettiin iskunpituuden rajoitus päälle. Lopuksi määritettiin iskunpituudeksi 4.0 mm.

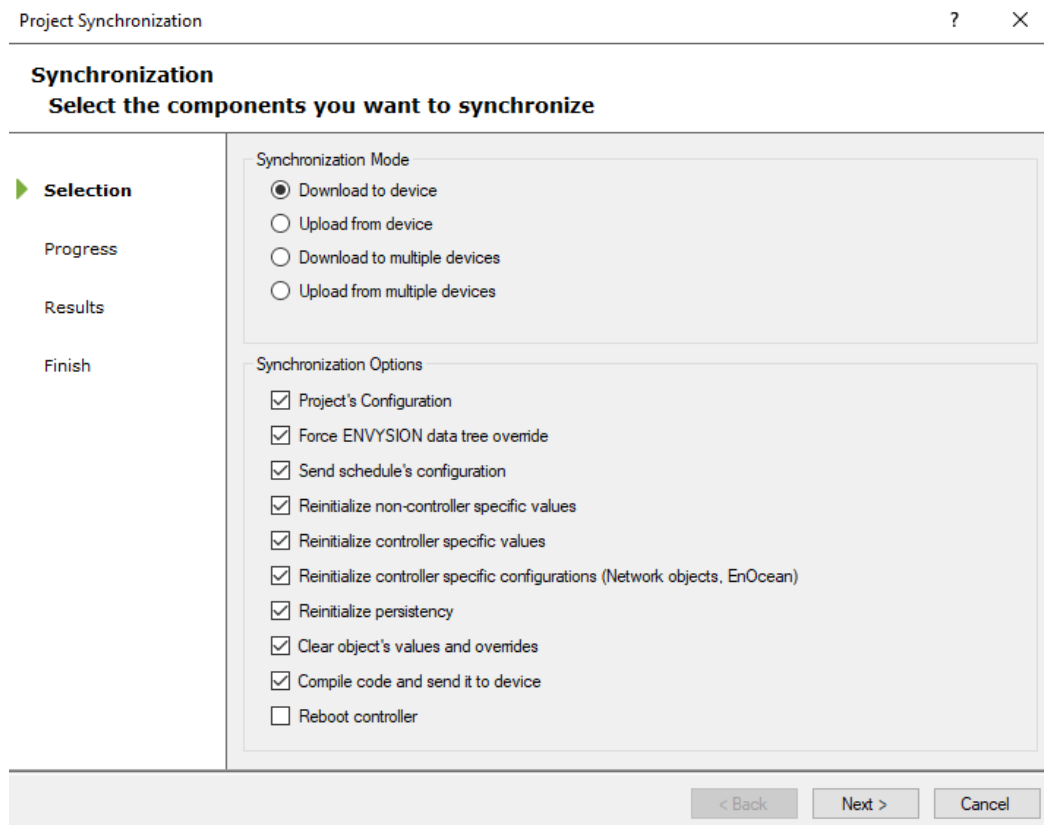
#### **4.2.9 Valmiin ohjelman lataaminen logiikkaan**

Ohjelman luomisen, laitteiston kytkentöjen ja TA-Sliderin asetusten muuttamisen jälkeen jäljellä oli enää ohjelman lataaminen logiikkaan. Tämän vaiheen suorittamisen jälkeen pääsimme kokeilemaan ohjelman toimivuutta käytännössä. Ennen ohjelman lataamista täytyi kuitenkin vielä tarkistaa tietokoneen ja logiikan välisen yhteyden toimivuus. Myös EC-gfxProgram-ohjelmisto täytyi käynnistää eri tilassa kuin aikaisemmin.

Ensimmäiseksi pingattiin logiikkakan IP-osoitetta, jotta saatiin testattua yhteyden toimivuus. Avattiin komentorivi kirjoittamalla "CMD" Windowsin hakuriville. Komentoriville kirjoitettiin "ping" komento ja sen perään logiikkakan IP-osoite. Yhteyden toimiessa komentorivi antoi vastauksen: Packets Sent = 4, Received = 4 0% Loss.

Käynnistettiin EC-gfxProgram uudelleen, jolloin aukesi Connect to Server ikkuna. Asetettiin Server tyypiksi "REST", sekä osoitteeksi logiikkakan IP-osoite. Ikkunan hyväksymisen jälkeen aukesi "Select Device" ikkuna, josta valittiin löytnyt ECY-S1000-logiikka ja painettiin "OK". Seuraavaksi tuotiin tiedostoista

aiemmin luotu ohjelma. Ohjelman tuomisen jälkeen se tarkistettiin virheiltä “Build” toiminnolla. Kun virheitä ei löytynyt, voitiin aloittaa ohjelman synkronointi logiikkaan “Synchronize” toiminnolla. Tällöin aukesi “Project Synchronization” valikko, josta valittiin Synchronization Mode “Download to device”. Tämän jälkeen valittiin “Synchronization Options” listasta kaikki muut vaihtoehdot paitsi “Reboot controller” kuvan 40 mukaisesti. Synkronointiasetusten valitsemisen jälkeen painettiin “Next” painikkeesta, jolloin ohjelma alkoi latautumaan logiikkaan. Latauksen jälkeen aukesi “Result” ikkuna, jossa luki “SUCCESS”. Tämä ikkuna suljettiin painamalla “Finish” painikkeesta.

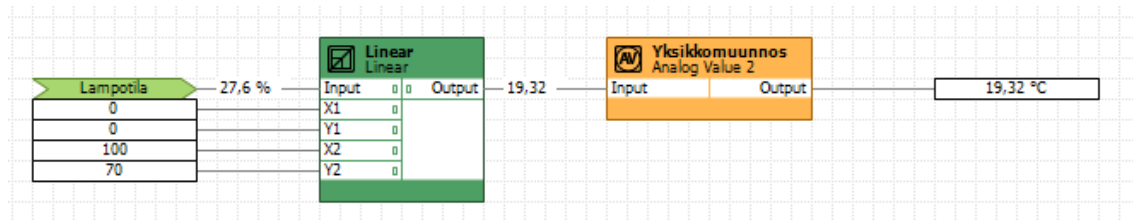


Kuva 40. Project Synchronization- ikkuna

#### 4.2.10 Valmiin ohjelman testaaminen

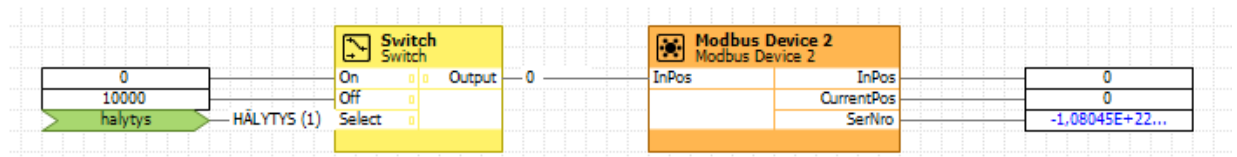
Ohjelman lataamisen jälkeen testattiin ohjelman toimivuus. Hälytyksen tapahtuessa TA-Sliderin pitäisi liikkua ja kun hälytys kuitataan JVA:n

kuittauspainikkeesta, niin Sliderin pitäisi palata takaisin aloitusasentoon. Ensimmäiseksi painettiin yläpalkista “Start”. Nyt ohjelmassa nähtiin jatkuvasti päivittyvä lämpötilatieto (kuva 41).



Kuva 41. Lämpötila-ohjelman toiminta

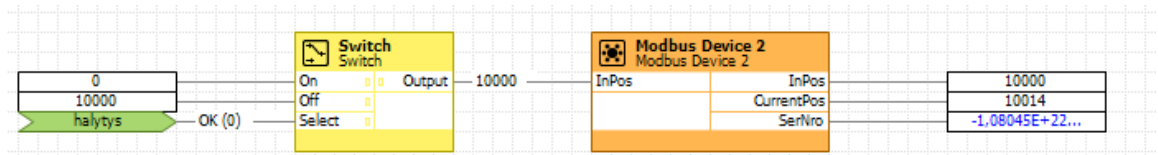
Seuraavaksi kokeiltiin JVA 24:n hälytyksen toimivuutta. JVA 24 laukaisee hälytyksen lämpötilan laskettua alle asetetun arvon, tai kun JVA 24:n ja lämpötila-anturin välille tulee virhe esimerkiksi johtimen irrotessa. Tässä harjoituksessa hälytys aiheutettiin irrottamalla toinen lämpötila-anturin johtimista. Hälytyksen lauetessa Slider ajoi itsensä kiinni eli asentoon 0 (kuva 42).



Kuva 42. Hälytys

Seuraavaksi kiinnitettiin johdin takaisin paikalleen ja kuitattiin hälytys JVA 24:n päällä olevasta kuittauspainikkeesta. Kuittaamisen jälkeen Slider ajoi itsensä takaisin asetettuun lähtöarvoon 10000 (kuva 43). Testaamisen jälkeen voitiin pysäyttää ohjelma painamalla yläpalkista “Stop” painikkeesta.





Kuva 43. Hälytyksen kuittaus

## 5 Pohdinta

Tämä opinnäytetyö toteutettiin Karelia-ammattikorkeakoulun laboratoriotyötilassa ja sen tavoitteena oli saada aikaan laboratorioharjoitus, jota hyödynnetään rakennusautomaatiokoulutuksessa. Opinnäytetyön lopputuloksena syntyi kaksi erillistä harjoitusta. Ensimmäisessä harjoituksessa opetellaan suorittamaan perusmittauksia TA-Scope mittausvälineistöllä, kun maksimirajoitusventtiiliin on liitetty TA-Slider toimilaite. Toisessa harjoituksessa TA-Slider kiinnitetään Modbus-väylään, jota ohjataan jäätymisvaaratermostaatin hälytystiedon avulla. Harjoituksessa luodaan EC-gfxProgram-ohjelmiston avulla oma ohjelma, jonka avulla hälytystieto muunnetaan Modbus-kenttäväylään sopivaksi.

Opinnäytetyön tekeminen oli välillä haastavaa ja tarkkaa työtä, sillä siinä olevat harjoitukset tuli tehdä vaihe vaiheelta selkeäksi ja toistettavaksi. Harjoitukset oli tehtävä tarkasti selostaen, jotta opiskelijat voisivat toistaa harjoitukset. Harjoitusten toistamiseen ei vaadita aiempaa kokemusta käytetyistä laitteista ja ohjelmistoista. Opinnäytetyön tekeminen oli haasteellista, koska emme olleet aiemmin käyttäneet EC-gfxProgram ohjelmistoa ja TA-Slidereita. Työn suorittaminen vaati paljon tiedonhakua ja laitteistoon tutustumista.

Harjoitusta suunniteltaessa tuli ongelma sen kanssa, kun mietittiin, miten TA-Sliderin ajaminen vaikuttaa TA-Scopella suoritettaviin mittauksiin. HyTune-sovellus näytti TA-Sliderin asennon millimetreissä, joita oli hankala hyödyntää tarkoissa mittauksissa. Haluttiin selvittää, kuinka TA-Sliderin asennosta voitaisiin päätellä TA-Scopen asetettava venttiilin esisäätöarvo. Otettiin yhteyttä

tuotteen valmistajaan, IMI Hydronic Engineering Oy:öön. Kysyttiin ratkaisua asiakaspalvelun kautta siihen, miten voitaisiin luotettavasti saada TA-Scopella mittausarvot, kun käytettiin TA-Sliderilla ohjattavaa maksimirajoitusventtiiliä. Saatiin heiltä ohjeeksi asettaa HyTune-sovelluksessa venttiilin malliksi TA-Modulator ja venttiilin kooksi TA-Modulator 15. Näiden asetusten avulla HyTune-sovelluksesta päästiin näkemään TA-Slideria ajettaessa mitä venttiilin esisäätöarvoa nykyinen TA-Sliderin asento täsmää.

Opinnäytetyötä tehdessämme opimme rakennusautomaation laitteistoista, ohjelmistoista sekä muusta opinnäytetyöhön liittyvästä teoriasta. Saimme hyviä vinkkejä työssä käytettävään EC-gfxProgram ohjelmistoon koululla järjestetystä Caverionin koulutuksesta. Lisäksi IMI Hydronic Engineering Oy:n yhteyshenkilö antoi meille neuvoja HyTune-sovelluksen avulla TA-Sliderille määritettäviin asetuksiin. Suurin osa työn aikana vastaan tulleista asioista tuli aivan uusina.

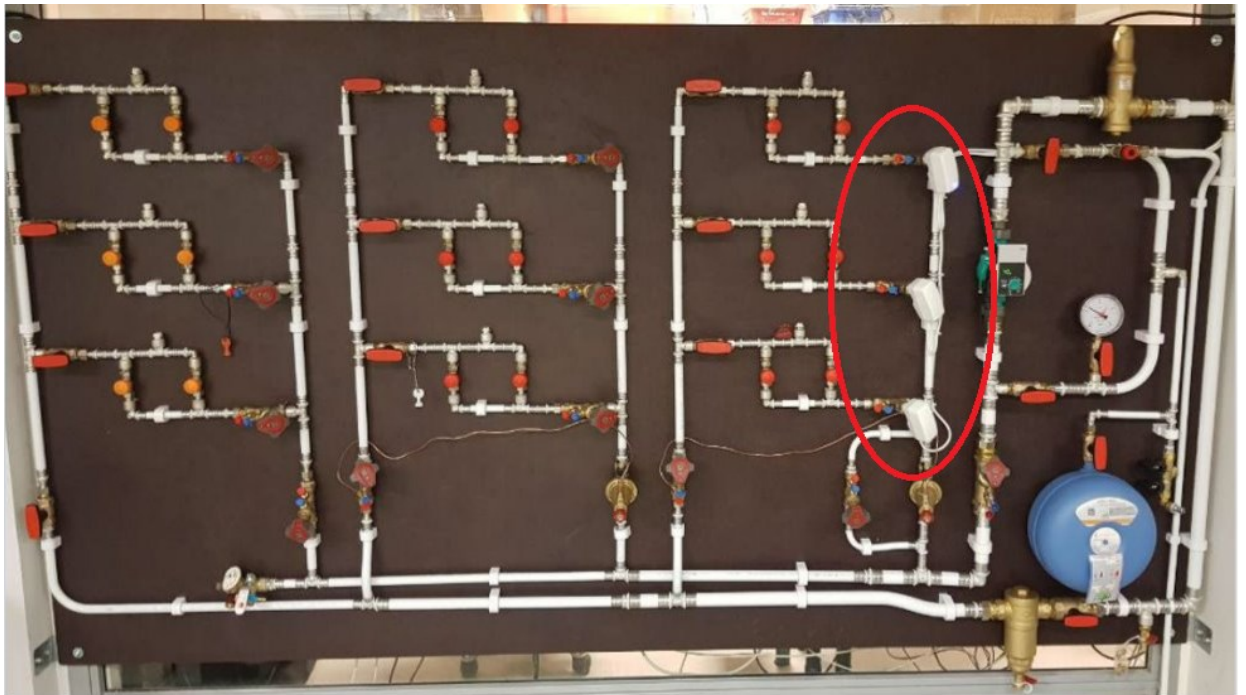
Opinnäytetyötä voidaan hyödyntää harjoituksena talotekniikan laboratorio-työympäristössä sekä ohjeena TA-Slidereiden liittämiseen Modbus-kenttäväylään. Luotuja laboratorioharjoituksia suoritettaessa opiskelijat tutustuvat rakennusautomaatiossa käytettäviin ohjelmistoihin ja laitteisiin sekä oppivat käyttämään niitä. Lisäksi he harjaantuvat tekemään niiden vaatimia kytkentöjä.

## 6 Lähteet

1. Shaw, K. 2022. The OSI model explained and how to easily remember its 7 layers. Networkworld. <https://urly.fi/2Q0G> (luettu 22.9.2022).
2. Geeksforgeeks. 2022. This is exactly why we still use the OSI model when we have TCP/IP Model. <https://urly.fi/2Q0J> (luettu 22.9.2022)
3. Krimaka. Osi- ja tcp/ip mallit <http://www.krimaka.net/tietotekniikka/verkko-ja-ethernet/osi-ja-tcp-ip-mallit.html> (luettu 22.9.2022)
4. Rinaldi, J. 2016. EtherNet/IP and TCP/IP. RTAutomation. <https://urly.fi/2Q0N> (luettu 22.9.2022)
5. Jameco. 2001. An Introduction to TCP/IP. <https://urly.fi/2Q0P> (luettu 22.9.2022)
6. Kaspersky. IP-osoite – määritelmä ja selitys. <https://urly.fi/2Q0R> (luettu 22.9.2022)
7. Ionos. 2020. Ethernet frame: definition and variants of the frame format. <https://urly.fi/2Q0T> (luettu 22.9.2022)
8. Geeksforgeeks. 2022. Ethernet Frame Format. <https://www.geeksforgeeks.org/ethernet-frame-format/> (luettu 22.9.2022)
9. Veijo Piikkilä ja Toivo Sahlsten 2017. Kiinteistöjen tiedonsiirtoväylät. ST-käsikirja 21. Tietotekniset järjestelmät.
10. Modbus. Modbus FAQ: About The Modbus Organization. <https://modbus.org/faq.php> (luettu 22.9.2022)
11. KNX. KNX - ratkaisusi nykyaikaiseen rakentamiseen ja ohjauksiin. <https://www.knx.fi/> (luettu 22.9.2022)
12. WuT. RS485 Bus Systems. <https://www.wut.de/e-6www-11-apus-000.php> (luettu 22.9.2022)
13. Schneider Electric. Modbus Master-Slave Principle. <https://urly.fi/2NkD> (luettu 22.9.2022)
14. Acromag. 2005. Introduction to Modbus TCP/IP. [https://www.prosoft-technology.com/kb/assets/intro\\_modbustcp.pdf](https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf) (luettu 22.9.2022)
15. Bacnet. Frequently Asked Questions (FAQs). <https://bacnet.org/faq/> (luettu 22.9.2022)

16. Automated Buildings. 2008. Object Modeling a Physical BACnet Device. <https://urly.fi/2Q0Z> (luettu 22.9.2022)
17. Tonejc J. 2015. Visualizing BACnet Data to Facilitate Humans in Building-Security Decision-Making. ResearchGate. <https://urly.fi/2NkU> (luettu 22.9.2022)
18. IMI Hydronic Engineering. TA-Slider 160 BACnet/Modbus. <https://urly.fi/2Q6M> (luettu 22.9.2022)
19. IMI Hydronic Engineering. TA-Slider 160 <https://urly.fi/2NkX> (luettu 22.9.2022)
20. IMI Hydronic Engineering. 2019. TA-Dongle Quick Guide. <https://urly.fi/2Q6R> (luettu 22.9.2022)
21. IMI Hydronic Engineering. TA-Slider 160. [TA-Slider 160 \(imi-hydronic.com\)](https://urly.fi/2NkX) (luettu 22.9.2022)
22. Produl. 2014. Jäätymisvaaratermostaatti JVA 24. <https://urly.fi/2NI3> (luettu 22.9.2022)
23. Produl. 2019. TEP NTC 10. <https://urly.fi/2NI5> (luettu 22.9.2022)
24. Distech Controls. ECLYPSE Connected System Controller. <https://urly.fi/2NIa> (luettu 22.9.2022)
25. Distech Controls. ECY-S1000 Series Spec Sheet. <https://urly.fi/2NIb> (luettu 22.9.2022)
26. Distech Controls. EC-gfxProgram Graphical Programming Interface. <https://urly.fi/2NIc> (luettu 22.9.2022)
27. Distech Controls. 2019. EC-gfxProgram Spec Sheet. [https://docs.distech-controls.com/bundle/EC-gfxProgram\\_SP](https://docs.distech-controls.com/bundle/EC-gfxProgram_SP) (luettu 22.9.2022)
28. IMI Hydronic Engineering. TBV-C. <https://urly.fi/2Q6W> (luettu 22.9.2022)
29. IMI Hydronic Engineering. TA-Scope. [TA-SCOPE \(imi-hydronic.com\)](https://urly.fi/2Q6W) (luettu 22.9.2022)
30. IMI Hydronic Engineering. 2020. TA-SCOPE-FI-MAIN.pdf <https://urly.fi/2Q6T> (luettu 22.9.2022)
31. Produl. 2019. TEP PT 1000 - Strap-on temperature sensor. <https://urly.fi/2NIh> (luettu 22.9.2022)

# Liitteet



Liite 1. Demoverkoston mittauskohdat.