



BECKHOFF-LOGIIKAN KÄYTTÖ JÄRJESTELMIEN VÄLISENÄ RAJAPINTANA

Lappeenrannan–Lahden teknillinen yliopisto LUT

Sähkö- ja automaatiotekniikan diplomityö

2024

Samuli Murtoniemi

Tarkastajat: Professori Pertti Silventoinen
DI Janne Jäppinen

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT
LUT School of Energy Systems
Sähkötekniikka

Samuli Murtoniemi

Beckhoff-logiikan käyttö järjestelmien välisenä rajapintana

Sähkö- ja automaatiotekniikan diplomityö 2024

46 sivua, 11 kuvaa ja 1 taulukko

Tarkastajat: Professori Pertti Silventoinen ja nuorempi tutkija Janne Jäppinen

Avainsanat: logiikka, rajapinta, automaatio, integraatio

Tämän diplomityön tavoitteena oli toteuttaa ratkaisu kiinteistön eri automaatiojärjestelmien väliseen kommunikointiin. Tarkoituksena oli saada kiinteistön teatteritekniikka, valaistusohjaus sekä kiinteistöautomaation ohjaama ilmanvaihtojärjestelmä keskustelemaan saumattomasti keskenään, sekä ohjattua näitä kaikkia järjestelmiä yhdestä pisteestä kootusti. Kaikki nämä järjestelmät käyttävät omaan sisäiseen tiedonsiirtoonsa protokollia, jota toinen järjestelmä ei tue. Useiden eri kommunikointiprotokollia käyttävien järjestelmien väliseen tiedonsiirtoon ei ole olemassa valmiita ratkaisuja, vaan tällaisen tarpeen ratkaiseminen vaatii lähes aina kohde- ja tapauskohtaisen suunnitelman sekä rajapintaan sopivan laitteen valinnan. Kommunikointirajapinnan tulee tukea kaikkia kiinteistön eri järjestelmissä käytettäviä kommunikointiprotokollia molempiin suuntiin. Työ on tehty yhteistyössä LSK Talotekniikka Oy:n sekä PKS Talotekniikka Oy:n kanssa. Työ on rajattu koskemaan järjestelmien välisen rajapinnan toteutusta ja ohjelmointia.

ABSTRACT

Lappeenranta–Lahti University of Technology LUT
School of Engineering Science
Electrical Engineering

Samuli Murtoniemi

Beckhoff logic as an interface between operating systems

Electrical and Automation Engineering Diploma Thesis 2024

45 pages, 11 images and 1 table

Inspectors: Professor Pertti Silventoinen and junior researcher Janne Jäppinen

Keywords: logic, interface, automation, integration

The goal of this master's thesis was to implement a solution for communication between different automation systems in a property. The aim was to enable communication between the property's theater technology, lighting control, and the ventilation system controlled by building automation, and to control all these systems from a single point. Each of these systems uses its own internal communication protocol, which is not supported by the other systems. There are no ready-made solutions for communication between systems using different communication protocols. Solving such a need requires a target and case-specific plan, as well as the selection of a device suitable for the interface. The communication interface should support all communication protocols used in the different systems of the property in both directions. The work was done in collaboration with LSK Talotekniikka Oy and PKS Talotekniikka Oy. The scope of the work is limited to the implementation and programming of the interface between the systems.

LYHENNELUETTELO

Lyhenteet

BACNet	tiedonsiirtoprotokolla automaatiojärjestelmien väliseen kommunikointiin (Building Automation and Control Networks)
DMX	standardoitu digitaalinen sarjaliikenneprotokolla, käytetään yleensä valaistusohjaukseen (Digital Multiplex)
EDE	BACnet-protokollan tiedosto, joka sisältää laitteen nimen, tyyppitiedot, sen tukemat objektit, kommunikointiparametrit sekä muut tekniset tiedot. (Engineering Data Exchange)
EtherCAT	automaatiojärjestelmien verkkotekniikka sekä kenttäväyläprotokolla (Ethernet for Control Automation Technology)
IP	perusprotokolla, joka huolehtii IP-pakettien toimittamisesta pakettikytkentäisessä tietoverkoissa (Internet Protocol)
Kbps	mittayksikkö tiedonsiirtonopeudelle, kilobittia sekunnissa (kilobits per second)
KNX	tiedonsiirto- ja verkkoprotokolla automaatiojärjestelmän sisäiseen kommunikointiin (Konnex)
MS/TP	viestintäteknikka, joka perustuu määräävään laitteeseen sekä tiedonlähetyslupaan (Master-Slave/Token-Passing)
OSI	viitekehys tietoverkkojen keskinäiseen kommunikointiin (Open Systems Interconnection)
PLC	
RGB	värimalli, jossa eri värit luodaan yhdistämällä eri määriä punaista, vihreää ja sinistä valoa. (Red, Green, Blue)
TCP	IP-pohjainen protokolla, joka toimii kuljetuskerroksella yhdessä IP:n kanssa. Sarjamuotoinen protokolla. (Transmission Control Protocol)
TP	kierretty parikaapeli (Twisted Pair)
UART	sarjapohjainen tietoliikenneprotokolla UART (Universal Asynchronous Receiver/Transmitter)
UDP	kuuluu IP-pohjaisiin protokolleihin ja toimii kuljetuskerroksella yhdessä IP:n kanssa. Pakettimuotoinen protokolla (User Datagram Protocol)
VLAN	virtuaalinen paikallinen aliverkko (Virtual Local Area Network)

Sisällysluettelo

Tiivistelmä

Abstract

Lyhenneluettelo

Kuvaluettelo

Taulukkuuettelo

1	Johdanto.....	1
1.1	Erilaisia kommunikointiprotokollia käyttävien järjestelmien yhdistäminen	2
1.2	Menetelmät ja tutkimusasetelma	3
1.3	Järjestelmän rakenne.....	4
2	Rajapinnassa käytettävät väyläprotokollat	5
2.1	EtherCATista yleisesti	5
2.2	DMX-väylä ja protokolla.....	6
2.3	BACnet	7
2.4	KNX-järjestelmä.....	9
3	Väyläprotokollien käsittely logiikassa	13
3.1	DMX512-protokollan lukeminen Beckhoffilla	13
3.2	DMX-väylälle kirjoittaminen	18
3.3	KNX-protokollan käsittely Beckhoff-logiikalla	20
3.4	BACnet/IP integraatio Beckhoffilla.....	29
3.5	Järjestelmän tietoturva	32
3.6	Graafinen käyttöliittymä	34
4	Johtopäätökset	35
4.1	Rajapintajärjestelmän toiminta	35
4.2	KNX-ohjauksen käytettävyys	36
4.3	DMX ohjauksen toiminta.....	37
4.4	BACnet-liityntä.....	37

Kuvaluettelo

Kuva 1 Järjestelmän rakenne	4
Kuva 2 BACnet/IP kerrokset	8
Kuva 3 KNX-topologia	11
Kuva 4 Beckhoff liitännät DMX-universumeihin.	14
Kuva 5 DMX-kehys	15
Kuva 6 Tulevan DMX-signaalin käsittelyyn liittyvät toiminnot.....	17
Kuva 7 Lähtevän DMX-signaalin käsittelyyn liittyvät toiminnot.....	20
Kuva 8 KNX-järjestelmän luentaan käytettävät toiminnot.....	23
Kuva 9 KNX-väylälle kirjoittamiseen käytettävät toiminnot	27
Kuva 10 BACnet/IP EDE-tiedostosta saadut I/O-laitteet	31
Kuva 11 Rajapintajärjestelmän graafisen käyttöliittymän etusivu.....	34

Taulukkoluetelo

Taulukko 1 BACnet objektit	9
----------------------------------	---

1 Johdanto

Kiinteistöissä on nykyään useita eri automaatiojärjestelmiä, jotka ohjaavat tyypillisesti vain jotain tiettyä järjestelmää kiinteistössä olevista monista erilaisista järjestelmistä, tai vain osaa jonkin järjestelmän toiminnoista. Esimerkiksi kiinteistön valaistusta ja ilmanvaihtoa ohjataan usein eri järjestelmällä, koska valaistusohjaukseen on olemassa omia siihen erittäin hyvin soveltuvia järjestelmiään, jotka eivät sovellu juurikaan ilmanvaihdon ohjaamiseen, ja ilmanvaihdon ohjaamiseen taas omiaan, jotka eivät sovellu niin hyvin valaistusohjaukseen. Tällainen ratkaisu on tavanomainen siksi, että kummankin järjestelmän ohjaukseen on olemassa juuri kyseistä tarkoitusta varten suunniteltuja ohjausjärjestelmiä, joiden avulla tarvittavat ohjaukset saadaan toteutettua kohtuullisen helposti sekä kustannustehokkaasti.

Erilaiset järjestelmät saadaan kyllä ohjaamaan monenlaisia järjestelmiä. Esimerkiksi tavanomainen lämmitystä ja ilmanvaihtoa ohjaava kiinteistöautomaatiojärjestelmä kykenee ohjaamaan myös valaistusta, mutta monien ohjauslaitteiden, kuten liiketunnistimien ja ohjauspainikkeiden liittäminen tällaiseen järjestelmään on voi olla hyvin työlästä sekä taloudellisesti kannattamatonta työtä. Valaistusohjausjärjestelmät taas tukevat näitä liitettäviä kenttälaitteita suoraan, mutta näiden mahdollisuudet ohjata esimerkiksi lämmitystä ovat rajalliset. Kiinteistön omistajan kannalta järjestelmäkohtaisilla ohjausjärjestelmillä saadaan usein parempi lopputulos kuin sillä, että yksi järjestelmä työstettäisiin ohjaamaan kaikkia erilaisia ohjaustarpeita tarvitsevia järjestelmiä.

Eri ohjausjärjestelmiä käytettäessä esiintyy kuitenkin yleensä myös tarvetta saada jokin ohjausjärjestelmä keskustelemaan jonkin toisen ohjausjärjestelmän kanssa. Esimerkiksi valaistusohjauksesta saatetaan haluta läsnäolotieto ilmanvaihdon ohjaukseen. Tällainen yksisuuntainen ja yksinkertainen tieto voidaan yleensä siirtää eri järjestelmien välillä hyvinkin helposti, esimerkiksi kärkeitietona, jota tällaisessa käytetäänkin usein, mutta jos tietoja täytyy siirtää molempiin suuntiin ja useamman eri järjestelmän välillä, ei tällaisia ratkaisuja ole järkevää käyttää, vaan eri protokollia käyttävät järjestelmät kannattaa saada jollain muulla tavalla keskustelemaan keskenään.

Tietojen vaihtamiseen eri järjestelmien kesken on olemassa valmiita ratkaisuja, jotka muuttavat tyypillisesti vain yhden kommunikointiprotokollan toiseen, jolloin saadaan nämä kahta eri protokollaa käyttävät järjestelmät keskustelemaan keskenään, ja yleensä tämä onkin riittävä ratkaisu. Joskus on kuitenkin tarpeita saada useampi järjestelmä keskustelemaan keskenään. Ongelmaksi tässä muodostuu se, että tällaisia valmiita monia protokollia tukevia ratkaisuja ei olekaan tarjolla. Tämä johtuu siitä, että tarpeet tämänkaltaiseen ratkaisuun ovat hyvin harvinaisia, koska useimmiten kiinteistöissä ei ole tarvetta monille eri automaatiojärjestelmille, vaan tarvittavat toiminnot saadaan toteutettua yhdellä tai kahdella järjestelmällä. Joissain tapauksissa on kuitenkin tarpeen käyttää kullekin eri järjestelmälle juuri siihen tarkoitukseen suunniteltua ja valmistettua ohjausjärjestelmää.

Näissä tapauksissa, joissa järjestelmien välistä tiedonsiirtoa tarvitaan myös, täytyy valmiiden ratkaisujen puuttuessa käyttää jotakin rajapintaa hoitamaan eri järjestelmien välistä kommunikointia. Markkinoilla on kyllä erilaisia rajapintoina toimivia laitteita, mutta ne eivät täytä tämänkaltaisen tarpeen edellytyksiä, koska ne pääsääntöisesti keskustelevat vain kahden järjestelmän välillä. Näiden monipuolisempaa kehitystä rajoittavat muun muassa lukuisat eri standardit, jotka määrittelevät hyvinkin tarkasti kussakin eri kommunikointiympäristössä toimivien laitteiden ominaisuuksia. Lisäksi tarpeet monen eri järjestelmän yhdistämiseen ovat kuitenkin melko harvinaisia, ja tämä toimii osaltaan myös kehitystä rajoittavana tekijänä.

1.1 Erilaisia kommunikointiprotokollia käyttävien järjestelmien yhdistäminen

Eri järjestelmien välisen kommunikoinnin tarpeet ovat kohdekohtaisia ja useimmissa tapauksissa valmiit ratkaisut riittävät täyttämään kohteen tarpeet. Mikäli valmista ratkaisua ei kuitenkaan ole tarjolla, joudutaan usein tyytymään siihen, ettei kaikkia haluttuja toimintoja saada toteutettua suunnitellulla tavalla. Tässä työssä haetaan ratkaisua tilanteeseen, jossa Digital Multiplex (DMX) -protokollaa käyttävän teatteritekniikan halutaan ohjaavan ja kuuntelevan ohjauskomentoja kiinteistön Konnex (KNX) -pohjaisesta valaistusohjausjärjestelmästä, lisäksi molemmat järjestelmät keskustelevat myös kiinteistön ilmanvaihtoa ja lämmitystä ohjaavan kiinteistöautomaatiojärjestelmän kanssa Building Automation and Control Networks (BACnet) -protokollaa käyttäen. Näiden järjestelmien välisenä rajapintana toimivan teollisuuslogiikan myötä kokonaisuuteen tulee mukaan vielä Ethernet for Control

Automation Technology (EtherCAT) -protokolla, johon nämä kaikki mainitut protokollat tulee saada käännettyä.

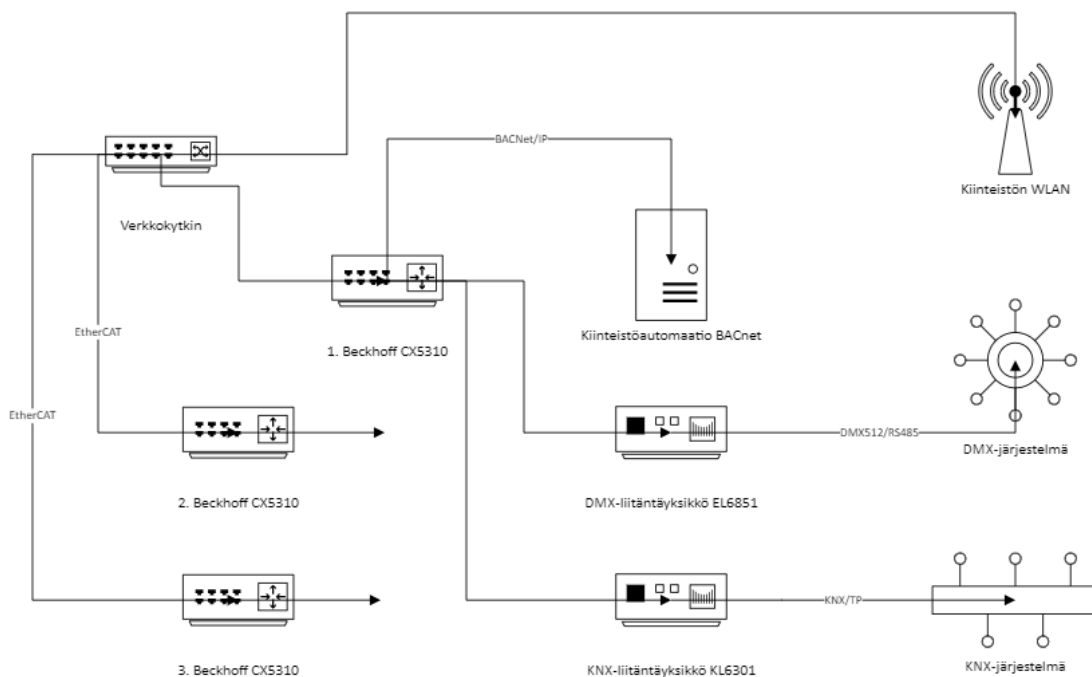
1.2 Menetelmät ja tutkimusasetelma

Tämä työ käsittelee eri kommunikointiprotokollilla varustettujen automaatiojärjestelmien keskinäistä tiedonsiirtoa ja siinä esiintyvien haasteiden ja ongelmien ratkaisemista. Menetelmänä tässä käytetään tapaustutkimusta. Tavoitteena on saada nämä eri kommunikointiprotokollia käyttävät järjestelmät vaihtamaan kaikkea tarvittavaa tietoa siten, että rajapintana toimivan logiikan toimintaa ei käytännössä havaittaisi lainkaan. Kohteessa on tarpeena saada välitettyä kaikkien teatteritekniikkaa käyttävien tilojen kaikki kyseiseen tilaan vaikuttavat eri järjestelmien ohjaukset välitettyä järjestelmästä toiseen, jotta eri ohjauksia varten ei tarvitse käyttää montaa erilaista järjestelmää, vaan kaikki tarvittavat ohjaukset voidaan toteuttaa yhdestä pisteestä. Tässä työssä selvitetään voiko tällaisen rajapintana toimivan järjestelmän tehdä Beckhoffin ohjelmoitavalla logiikalla sekä voiko tästä eri järjestelmien taustalla toimivasta järjestelmästä tehdä käyttäjän kannalta mahdollisimman huomaamattoman. Tutkimuksessa selvitettäviä asioita ovat:

1. Voidaanko Beckhoffin ohjelmoitavalla logiikalla toteuttaa rajapintana toimiva järjestelmä eri järjestelmien välille?
2. Mitkä ovat Beckhoffin ohjelmoitavan logiikan rajoitukset ja mahdollisuudet rajapintajärjestelmän toteuttamisessa?
3. Kuinka huomaamaton käyttäjäkokemus voidaan saavuttaa rajapintajärjestelmässä, joka yhdistää eri järjestelmiä?
4. Mitä haasteita voi ilmetä eri kommunikaatioprotokollilla varustettujen järjestelmien yhteensovittamisessa Beckhoffin logiikan avulla?

1.3 Järjestelmän rakenne

Rajapintana toimiva järjestelmä koostuu kolmesta Beckhoff CX5310-logiikasta, jotka ovat yhdistettynä toisiinsa EtherCAT-kytkimen kautta EtherCAT-väylällä. Kullakin logiikkayksiköllä on omat liityntänsä kiinteistön teatteritekniikan DMX-väyliin ja valaistusohtausjärjestelmänä toimivaan KNX-järjestelmään ja lisäksi yhdestä logiikasta on liityntä kiinteistön ilmanvaihtoa ohjaavaan järjestelmään BACnet/IP yhteyden kautta.



Kuva 1 Järjestelmän rakenne

Jokaisessa logiikkayksikössä on myös http-palvelin järjestelmägraafiikan välittämiseen ja ne on liitetty kiinteistön henkilökunnan langattomaan verkkoon verkkokytkimen kautta. Sama verkkokytkin hoitaa sekä Ethernet- että EtherCat verkkoliikenteen.

2 Rajapinnassa käytettävät väyläprotokollat

Väyläprotokollat ovat olennainen osa tietoliikennejärjestelmää, joka mahdollistaa eri laitteiden ja komponenttien, kuten antureiden, ohjainten ja säätölaitteiden, kommunikoinnin keskenään automaatio- ja ohjausjärjestelmissä. Ne ovat eräänlaisia sääntöjä ja menettelytapoja, jotka määrittävät, miten tietoa vaihdetaan ja kuinka laitteiden, ohjelmistojen ja järjestelmien välisissä tietoverkoissa tai muissa kommunikaatioympäristöissä kommunikoidaan. Ne tarjoavat rakenteen ja formaatin tiedonsiirrolle, jotta eri laitteet tai ohjelmistot voivat ymmärtää toisiaan ja kommunikoida keskenään. Protokollalla määritetään tiedonsiirron formaatti, se määrittelee esimerkiksi, miten tiedot esitetään ja koodataan siirron aikana, kuten bittijärjestyksen, datan muodon sekä käytettävät merkit. Joissakin tapauksissa protokolla voi sisältää pakkauksen tai koodauksen sääntöjä, jotta tiedonsiirto olisi vähemmän resursseja vaativaa. Se myös määrittelee, miten tiedonsiirto ajoitetaan ja synkronoidaan, jotta vastaanottava laite voi tulkita siirretyn tiedon oikein. Jotkut protokollat sisältävät myös mekanismeja virheiden havaitsemiseen ja korjaamiseen siirron aikana.

Useimmat väyläprotokollat noudattavat kerrosmallia, esimerkiksi Open Systems Interconnection (OSI) -mallia ja toimivat kerroksellisesti osana laajempaa protokollapinoa. Jokainen kerros tarjoaa tiettyjä palveluita seuraavalle kerrokselle ja hyödyntää edellisen kerroksen palveluita. Esimerkiksi fyysinen kerros käsittelee bittien siirtämistä, kun taas sovelluskerros tarjoaa sovelluksille pääsyn verkkopalveluihin. Näitä kerroksien toimintaa avataan tarkemmin tässä työssä käsiteltäessä kerroksia hyödyntävien protokollien toimintaa.

2.1 EtherCATista yleisesti

EtherCAT, Ethernet for Control Automation Technology, on teollisuusautomaation ja automaatiojärjestelmien Ethernetiin pohjautuva verkkotekniikka sekä kenttäväyläprotokolla, joka on suunniteltu erityisesti reaaliaikaisiin ohjaussovelluksiin. Se mahdollistaa eri laitteiden kommunikoinnin keskenään erittäin nopeasti ja tarkasti, ja sitä käytetäänkin sovelluksissa, joissa tarvitaan tällaista nopeaa ja tarkkaa reaaliaikaista tiedonsiirtoa. EtherCATin sykliajat ovat Ethernet-pohjaisista ratkaisuista nopeimpia. Yksi EtherCAT-lohko muodostuu ketjuttamalla laitteet yhdellä kaapelilla. Tämä yksinkertainen fyysinen rakenne mahdollistaa nopean ja helpon verkkojen rakentamisen. EtherCAT on avoin standardi, mikä tarkoittaa,

että useat laitevalmistajat voivat tuottaa EtherCAT-yhteensopivia laitteita. Verkkoa voidaan kohtuullisen helposti laajentaa lisäämällä solmuja ja/tai uusia laitteita. (Beckhoff, EtherCAT System Description, 2023)

EtherCATin avoimen standardin ansiosta useat laitevalmistajat voivat tuottaa EtherCAT-yhteensopivia laitteita. Tämän vuoksi järjestelmän laajentaminen uusilla laitteilla tai solmuilla on suhteellisen helppoa. Tämä ominaisuus EtherCATista joustavan ratkaisun, joka soveltuu monenlaisiin sovelluksiin ja ympäristöihin.

EtherCAT-verkon konfigurointi voi kuitenkin olla monimutkaisia, etenkin suurissa ja monimutkaisissa järjestelmissä. EtherCAT-järjestelmän diagnostiikka ja vianmääritys voivat olla haastavia verrattuna joihinkin muihin protokolliin verrattuna, johtuen siitä, että EtherCAT on hyvin monipuolinen protokolla. Vaikka EtherCAT onkin avoin standardi, se ei silti välttämättä ole yhteensopiva kaikkien laitevalmistajien kanssa. Yhteensopivuuden kanssa voi siis tulla ongelmia, jos käytetään eri valmistajien laitteita samassa järjestelmässä, joten yhteensopivuusongelmien välttämiseksi EtherCAT-ympäristössä käytetään kuitenkin yleensä vain yhden valmistajan tuotteita yhdessä järjestelmässä. (ETG, 2023)

2.2 DMX-väylä ja protokolla

DMX, Digital Multiplex, on digitaalinen ohjausprotokolla, jota käytetään yleisimmin valaistustekniikassa, erityisesti viihde- ja näyttämövalaistuksen ohjauksessa, mutta sitä voidaan käyttää muunlaiseen valaistusohjaukseen. Protokollaa voidaan käyttää myös muiden viihdealan laitteiden ohjaukseen, kuten valaisimien liikkeen, savukoneiden, laserien ja äänijärjestelmien ohjaukseen. DMX protokolla on standardoitu ja hyvin dokumentoitu, mikä lisää eri valmistajien laitteiden yhteensopivuutta. Myös protokollan yksinkertaisuus mahdollistaa osaltaan yhteensopivuuden erilaisten laitteiden kesken. DMX-protokolla on alun perin suunniteltu valaistuksen ohjaamiseen, ja se tarjoaa rajoitetun määrän ohjaustietoja melko hitaalla tiedonsiirtonopeudella, eikä sen reaaliaikainen suorituskyky riitä nopeutta vaativiin sovelluksiin. Tästä syystä sen käyttö rajoittuu yleensä pelkästään valaistusohjaukseen, jossa se on kuitenkin yksi nopeimmista siihen tarkoitukseen käytettävistä protokollista. DMX-protokollan etuina muihin valaistusohjaukseen käytettäviin protokolliin verrattuna onkin sen nopeus ja yksinkertaisuus sekä helppo käyttöönotto. Se on kuitenkin hyvin yksinkertainen protokolla, joka lähettää pelkkiä numeroarvoja, eikä se tarjoa monimutkaisia tietorakenteita tai

metatietoa. Tämä rajoittaa monimutkaisempien ohjaustoimintojen mahdollisuuksia ilman lisäprotokollien tai -ratkaisujen käyttöä. (ESTA, 2004)

DMX-ohjausprotokollan päälle on tästä syystä kehitetty myös Remote Device Management (RDM) protokolla, joka mahdollistaa monipuolisemman tiedonsiirron laitteiden välillä. RDM on DMX-protokollan päälle integroitu lisäominaisuus, joka mahdollistaa kaksisuuntaisen kommunikation DMX-väylällä. Se tarjoaa mahdollisuuden etäohjata ja hallita DMX-laitteita. (ESTA, 2011)

DMX512 käyttää RS-485-sarjaliikennettä tiedonsiirtoon 250 kilobittiä sekunnissa (kbps) nopeudella. DMX512 käyttää 8-bittisiä tietopaketteja ohjaustiedon välittämiseen. Nimensä mukaisesti DMX512 mahdollistaa jopa 512 kanavan ohjaamisen yhdessä DMX-universumissa. DMX-universumi on eräänlainen virtuaalinen kanavaryhmä, jonka alla yksi fyysinen laitekokonaisuus sijaitsee. Kukin DMX-kanava voi ottaa vastaan arvoja välillä 0–255, joiden avulla määritellään liitettyjen laitteiden tilat ja toiminnot. Esimerkiksi yhdessä DMX-universumissa voi ohjata 512 valaisinta niin, että jokainen kanava ohjaa yhtä valaisinta, ja kanavan arvo määrää valaisimen kirkkauden. Ohjauskanavan rajallisen tietomäärän vuoksi enemmän ohjausdataa vaativilla laitteilla käytetään useaa kanavaa ohjaukseen. Tällaisessa tapauksessa vastaanottavalle laitteelle määritetään niin sanottu aloituskanava, ja se kuuntelee ohjaustietoa tästä kanavanumerosta eteenpäin. Esimerkiksi RGB- värivalaisimen, jossa ohjataan kolmea värikanavaa: punaista (Red), vihreää (Green) ja sinistä (Blue) valonlähdettä omilla ohjausarvoillaan, kohdalla yhteensä kolmea kanavaa saadakseen kaikille R, G ja B väreille ohjausarvot. (ESTA, 2004)

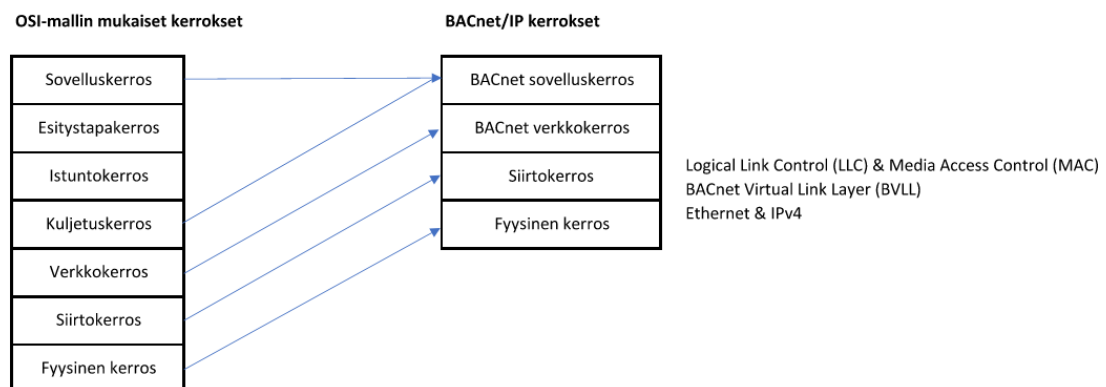
2.3 BACnet

BACnet on standardoitu protokolla, joka on suunniteltu rakennusautomaatiojärjestelmien hallintaan ja valvontaan. Se mahdollistaa eri laitteiden ja järjestelmien, kuten ilmastoinnin, lämmityksen, valaistuksen, turvajärjestelmien ja muiden rakennusautomaation komponenttien, yhteistoiminnan ja tiedonsiirron. BACnet on avoin ja laajasti hyväksytty standardi, mikä tarkoittaa sitä, että useat eri valmistajat voivat tuottaa BACnet-yhteensopivia laitteita ja järjestelmiä. Tämä edistää yhteensopivuutta ja valinnanvaraa rakennusautomaation laitteiden hankinnassa. BACnet tukee erilaisia tiedonsiirtotyyppejä, mikä mahdollistaa monipuoliset asennusvaihtoehdot sekä skaalautuvuuden. Näistä yleisimmät ovat BACnet Master

Slave Token Passing (MS/TP) ja BACnet/Internet Protocol (IP). BACnet tukee myös reaaliaikaista tiedonsiirtoa ja ohjausta.

BACnet MS/TP on yleisin BACnetin käyttämä tiedonsiirtotyyppi. Se perustuu RS-485-sarjaliikenteeseen ja käyttää master/slave-topologiaa sekä token-pohjaista tiedonsiirtotapaa. Token-pohjainen tiedonsiirtotapa on viestintäprotokolla, joka perustuu token -käsitteeseen. Token on eräänlainen ohjaussanoma tai -merkki, joka kiertää verkkoa ennalta määritetyssä järjestyksessä. Laitteet voivat lähettää tietoa vain silloin, kun niillä on token hallussaan.

BACnet/IP tai BACnet over Ethernet on BACnetin Internet-protokollaan perustuva versio. Se mahdollistaa BACnet-tiedon lähettämisen ja vastaanottamisen IP-pohjaisissa tietoverkoissa.



Kuva 2 BACnet/IP kerrokset

BACnet-protokolla on rakenteeltaan monikerroksinen ja se noudattaa Open Systems Interconnection -mallin (OSI) periaatteita, koostuen useista kerroksista, joista jokainen suorittaa tietyn tehtävän. BACnet-protokollassa käytetään kuitenkin vain osaa OSI-mallin kerroksista. Fyysinen kerros määrittelee laitteiston, signaalin ja jännitteen, jonka avulla tieto siirtyy. BACnet voi toimia useilla eri fyysisillä tiedonsiirtotekniikoilla, esimerkiksi edellä mainitut RS-485 sekä Ethernet. Siirtokerros vastaa tiedon siirtämisestä laitteiden välillä sekä tiedon jakamisesta eri verkkojen ja verkkolaitteiden kesken. Tässä kerroksessa BACnet käyttää esimerkiksi Ethernet-verkkopohjaista tiedonsiirtoa. Verkkokerros määrittelee, miten

tietopaketit ohjataan ja reititetään eri laitteiden välillä verkon yli, se sisältää tietopakettien osoitteistuksen ja reitityksen. OSI-mallin mukainen kuljetuskerros hallitsee yleensä tiedon siirtoa laitteiden välillä ja varmistaa, että tieto saapuu perille. BACnetissä ei ole kuitenkaan määritetty varsinaista kuljetuskerrosta, vaan sen tehtäviä hoitaa osittain sovelluskerros. Sovelluskerros määrittelee BACnet-protokollassa käytetyt viestityypit, kuten tiedon rakenteen ja viestintäkäsyt. BACnet käyttää User Datagram Protocol -protokollaa, (UDP)-protokolla tässä kerroksessa. OSI-mallista poiketen BACnetin sovelluskerroksessa on myös ratkaisut tiedonsiirron varmistamiseen ja virheenkorjaukseen. BACnet/IP käyttää näissä menetelmänä uudelleenlähetystä UDP-protokollan vahvistus, eli Acknowledgment-viestin, (AKC) avulla. Mikäli tämä kuittaus jää saamatta, lähetetään paketti uudelleen. (ASHRAE, 2020)

BACnet/IP-protokollassa käytetään asiakas/palvelin -mallia, joka tarkoittaa sitä, että laitteet toimivat järjestelmässä joko asiakkaana tai palvelimena. Palvelimena toimiva laite tarjoaa tietoja ja palveluita muille laitteille. Palvelin vastaa asiakkaiden pyyntöihin ja tarjoaa niille tarvittavat tiedot. Asiakas lähettää kyselyitä, ohjaukomentoja ja vastaanottaa tietoa palvelimelta. BACnet-standardi määrittelee tiedonsiirtoon 26 erilaista objektiä, joilla tietoja siirretään asiakkaan ja palvelimen välillä. (ASHRAE, 2020) Tässä työssä käytetyt objektit on esitetty taulukossa 1.

Taulukko 1 BACnet objektit

Tietopisteen tyyppi/ BACnet-objektin tyyppi	Objektin tunnistus	Tietopisteen kuvaus
Multi-state Input (13) - BACnetMultiStateInput	13	Monitilainen tulo, mittaa ja lukee tuloa, joilla on useampia kuin kaksi toimintatila.
Multi-state Output (14) - BACnetMultiStateOutput	14	Monitilainen lähtö, lähettää ohjaukomentoja laitteille, joilla on useampia kuin kaksi toimintatila.
Multi-state Value (19) - BACnetMultiStateValue	19	Monitilainen arvo, tallentaa ja hallinnoi monitilaisia arvoja, jotka eivät ole suoraan liitetty fyysiseen laitteeseen.

2.4 KNX-järjestelmä

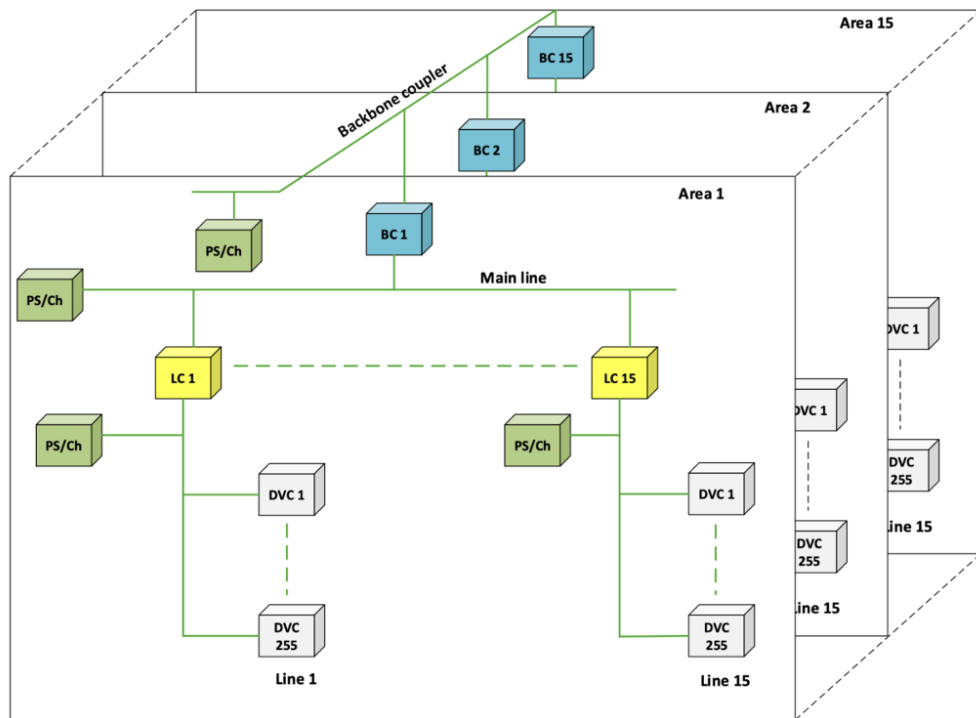
KNX on avoin ja standardoitu protokolla sekä viestintäteknikka, jota käytetään tavallisesti ohjaamaan kiinteistön valaistusta, ilmanvaihtoa tai lämmitystä. KNX voi toimia erilaisilla fyysisillä medioilla, kuten kierrettyillä parikaapeleilla (TP), radioverkkotekniikalla (RF) ja IP-verkoilla. Kierrettyjä parikaapeleita käytetään yleisimmin, ja ne tarjoavat luotettavan, häiriöttömän ja vakaan siirtotien tietojen välittämiseen.

KNX protokolla on monikerroksinen protokolla. Sen väyläkerros vastaa tiedonsiirron hallinnasta ja ohjauksesta. Se varmistaa, että tiedot siirtyvät oikein väylällä. KNX käyttää

kolmea viestimuo- toa: unicast (yksisuuntainen viesti yhdestä laitteesta toiseen), multicast (viesti useammalle vastaanottajalle) ja broadcast (viesti kaikille väylän laitteille). Verkko- kerros mahdollistaa laitteiden tunnistamisen ja osoittamisen väylällä. Jokaiselle KNX-laitteelle annetaan yksilöllinen osoite, joka mahdollistaa viestien ohjaamisen oikeille vastaanottajille. Kommunikaatiokerros määrittelee viestien rakenteen ja formaatin. KNX-protokolla käyttää erityyppisiä viestejä, kuten ohjaus- ja tiedusteluviestejä laitteiden väliseen tiedon- siirtoon.

KNX-protokolla noudattaa hajautettua arkkitehtuuria, jossa ei ole yhtä ohjaavaa laitetta, vaan jokainen laite sisältää kaikki tarvittavat toiminnot kommunikointiin muiden KNX-laitteiden kanssa. KNX-järjestelmä koostuu väylästä, linjoista ja laitteista. Minimissään KNX-järjestelmä voi koostua yhdestä väylästä ja kahdesta keskenään keskustelevasta laitteesta ja maksimissaan 15 eri alueesta, joissa kussakin voi olla 15 linjaa ja kussakin linjassa 64 laitetta. Tämä mahdollistaa hierarkkisen rakenteen ja järjestelmän skaalautuvuuden. KNX-laitteet noudattavat tietyntyyppisiä laiteprofiileja, jotka määrittelevät, mitä toimintoja kukin laite voi suorittaa. Laiteprofiilien käyttö helpottaa yhteensopivuutta ja integrointia eri valmistajien laitteiden välillä. (KNX Association, 2023)

KNX TP-väylän nopeus on 9600 bps ja dataa lähetetään sarjamuotoisena tavu kerrallaan tahdistamattomalla tiedonsiirrolla, jossa vastaanottaja kuittaa jokaisen sanoman vastaanotetuksi. KNX TP-sanoma koostuu yhden tavun kontrollikentästä, yhteensä neljän tavun sekä yhden bitin suuruisesta osoitekentästä, ja 1–16 tavun suuruisesta datakentästä. Lisäksi sanoman loppuun tulee yhden tavun tarkistussummakenttä, jolla varmistetaan, että sanoma on lähetetty ja vastaanotettu oikein.



Kuva 3 KNX-topologia

KNX-järjestelmässä käytetään kahta osoiteavaruutta, laite- ja ryhmäosoitetta. Laiteosoite on aina yksilöllinen. Osoite on malliltaan esimerkiksi 1.2.3. Yksilöllisen osoitteen rakenne sisältää kolme tasoa. Ensimmäinen taso, eli osoitteen ensimmäinen numero, on alueosoite. Alueosoite määrittelee alueen, jossa linjat sijaitsevat. Alueosoitteella erotetaan tyypillisesti fyysiset alueet tai osat rakennuksessa, kuten rakennukset eri siivet. Näitä alueosoitteita voi olla viisitoista kappaletta. Toinen taso, eli osoitteen toinen numero, on linjaosoite. Kunkin alueosoitteen alla voi olla viisitoista linjaosoitetta. Linjaosoitteella määritetään, missä fyysisessä linjassa tai verkon osassa laite sijaitsee. Viimeinen taso, eli osoitteen kolmas numero, on laiteosoite, joka määrittelee, kuinka laite sijoittuu linjalle. Laiteosoitteita voi olla yhden linjan alla 255 kappaletta. Laiteosoitteen avulla kullekin laitteelle osataan lähettää ohjelmointivaiheessa oikeat tiedot. Laiteosoitetta hyödynnetään myös järjestelmän diagnosointiin, muuttamiseen ja uudelleenohjelmointiin. Laiteosoitetta ei käytetä järjestelmän sisäisessä tiedonsiirrossa lainkaan, vaan laitteiden välinen kommunikointi tapahtuu ryhmäosoitteita käyttäen. Järjestelmän toimiessa normaalisti laiteosoitetta ei siis käytetä mihinkään. (KNX Association, 2023)

KNX-ryhmäosoite on järjestelmän sisäisen tiedonsiirron osoite. Osoite on kolmitasoinen ja se voi olla esimerkiksi 0/1/1, jossa osoitteen ensimmäinen numero on pääryhmä. Pääryhmiä järjestelmässä voi olla 32 kappaletta, osoitteet 0–31. Osoitteen toinen taso, eli toinen

numero, on keskiryhmä. Kukin pääryhmä voi sisältää kahdeksan keskiryhmää, osoitteet 0–7. Alin taso, osoitteen viimeisin numero, on aliryhmä, näitä osoitteita voi olla kunkin keskiryhmän alla 256 kappaletta, osoitteet 0–255. Ryhmäosoite on kokonaisuus, kaikki tasot täytyy määrittää ja lähetetty sanoma sisältää aina nämä kaikki kolme tasoa. Ohjelmointiohjelmassa laitteiden kommunikointiobjektit sijoittuvat kuitenkin aliryhmän alle. Laitteilla voi olla useita kommunikointiobjekteja, joiden kautta laitteet vaihtavat tietoa keskenään. Yksittäisen laitteen yksi kommunikointiobjekti voi lähettää tietoa vain yhteen ryhmäosoitteeseen ja järjestelmän kaikki laitteet voivat kuunnella tätä ryhmäosoitetta. (KNX Association, 2023) Rajapintajärjestelmässä tällaista rajoitusta ei ole, vaan se voidaan ohjelmoida lähettämään useampi sanoma väylälle yhden toiminnon aikana. KNX-järjestelmän ohjelmoinnissa rajapintajärjestelmää ei tarvitse huomioida muulla tavalla, kuin varmistamalla, että kaikki rajapintajärjestelmään haluttavat ryhmäosoitteet pääsevät linjayhdistimien läpi rajapintalaitteelle asti. Tämä tapahtuu yksinkertaisimmillaan lisäämällä jokaiseen haluttuun ryhmäosoitteeseen rajapintalaitteen kommunikointiobjekti, jolloin KNX-ohjelmointiohjelma osaa itse luoda linjayhdistimille oikeanlaisen reitityksen sanomille.

3 Väyläprotokollien käsittely logiikassa

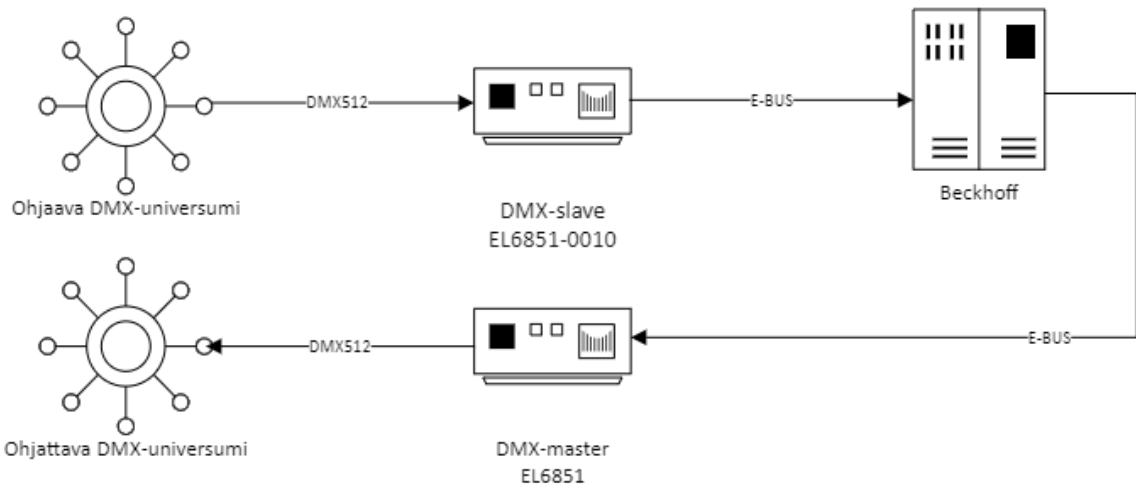
Beckhoffin logiikka tukee fyysisesti monia eri väyläprotokollia erilaisten väyläliityntäyksiköiden kautta. Väyläliityntäyksikkö on Beckhoff-järjestelmän sisäiseen väylään liitettävä fyysinen laite, joilla liitynnät eri järjestelmiin tehdään. Näiden väyläliityntäyksiköiden avulla saadaan liitettyä eri protokollia käyttävät väylät EtherCAT-ympäristöön sekä myös takaisin alkuperäiseen protokollaan. Beckhoff-logiikan sisäinen kommunikointi tapahtuu EtherCAT-protokollaa käyttäen. (Beckhoff, EtherCAT System Description, 2023) Fyysinen järjestelmäväylä liitetään väyläliityntäyksikköön, jolle on valmis laitevalmistajan ohjelmisto, jonka avulla kunkin väylätyypin tai -protokollan sisältämää tietoa voidaan lukea ja/tai kirjoittaa. Jotkin väyläliityntäyksiköt ovat yksisuuntaisia, eli lukemista ja kirjoittamista varten tarvitaan oma yksikkönsä. Esimerkiksi DMX-väylän kommunikointiyksiköt ovat tällaisia. KNX-kommunikointiyksikkö taas on kaksisuuntainen laite, eli samalla fyysisellä laitteella voidaan sekä lukea väylää että kirjoittaa sinne. Jotkin väylätyypit voidaan taas liittää suoraan logiikan ethernet-porttiin, kuten tässä työssä käytetty BACnet/IP. (Beckhoff Information System, 2023)

Beckhoff-järjestelmä on ohjelmoitava teollisuuslogiikka, jossa ohjelmakoodilla saadaan käsiteltyä kaikkea väyläliityntäyksiköiden kautta järjestelmään saatua tietoa. Vapaa ohjelmoitavuus mahdollistaa kaiken väylistä saatavan tiedon hyödyntämisen järjestelmän sisällä ja myös ulos siitä. Ohjelmoinnissa pyrittiin tekemään kaikille eri väyläprotokollien luenta- ja kirjoitustoiminnolle omat funktioblokkinsa, jolloin niiden toimintoja on mahdollista käyttää missä tahansa kohtaa ohjelmakoodia kyseistä blokkia kutsumalla. Väylistä saadut tiedot tallennettiin logiikan globaaleihin muuttujiin, jolloin ne olivat helposti kaikkien eri ohjelmien ja funktioblokkien käytettävissä.

3.1 DMX512-protokollan lukeminen Beckhoffilla

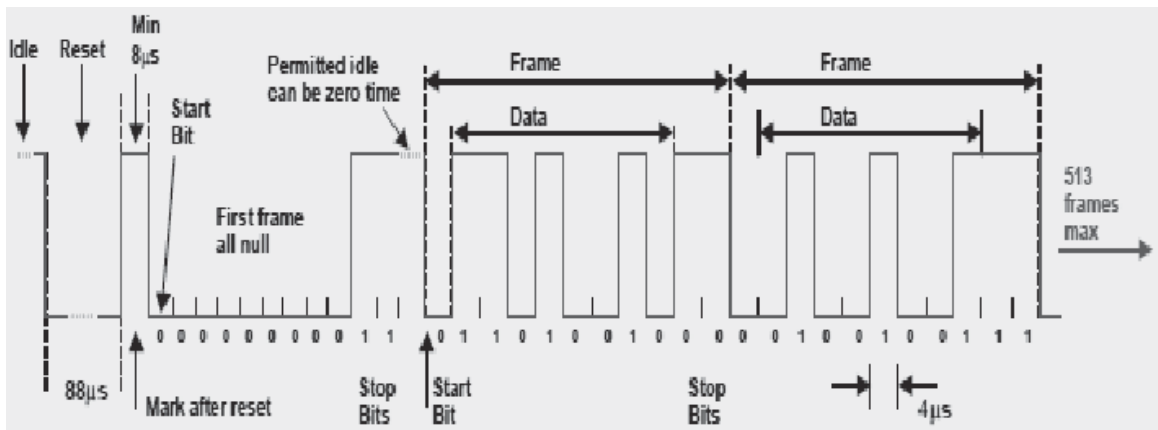
DMX512-laitteet liitetään Beckhoff-järjestelmään 1-kanavaisen DMX kommunikointiyksikön avulla. Käytettävissä on kaksi eri laitetta, joko DMX-master EL6851 tai DMX-slave EL6851-0001, näitä erilaisia liityntätarpeita varten. DMX-ympäristössä master-laitteet ovat käskyjä antavia laitteita ja slave-laitteet käskyjä vastaanottavia laitteita, eikä yksi laite voi hoitaa molempia rooleja. Perus DMX-signaali on yksisuuntainen signaali, joka tarvitsee aina

tällaisen lähettäjä-vastaanottaja-parin, jossa master-laite ohjaa ja slave-laitteet toteuttavat ohjaussignaalin mukaiset toiminnot. Tässä tapauksessa järjestelmä voidaan liittää kuuntelemaan yhtä DMX-universumia ja ohjaamaan toista DMX-universumia. (ESTA, 2004) Ohjaava DMX-universumi on kohteessa teatteritekniikan valopöytä ja ohjattava universumi käsittelee kohteen DMX-valaisimet.



Kuva 4 Beckhoff liitynnät DMX-universumeihin.

Beckhoff EL6851 ja EL6851-0001 ovat käytännössä RS485 half-duplex sarjaliikenneportteja ja ne voivat joko lähettää tai vastaanottaa dataa maksimissaan 250 kbps nopeudella UART-tekniikkaan pohjautuen. Half-duplex laitteet voivat lähettää ja vastaanottaa tietoa, mutta tieto voi liikkua vain yhteen suuntaan kerrallaan. DMX-järjestelmässä tällä ei ole merkitystä, koska laitteet ovat aina vain joko lähettäviä tai vastaanottavia. Beckhoff-järjestelmässä sama fyysinen laite ei voi kuitenkaan olla lähettävänä tai vastaanottavana laitteena vaan näihin tarkoituksiin on kumpaankin oma laitteensa. (Beckhoff, EL6851DMX Master/Slave terminal, 2023)



Kuva 5 DMX-kehys

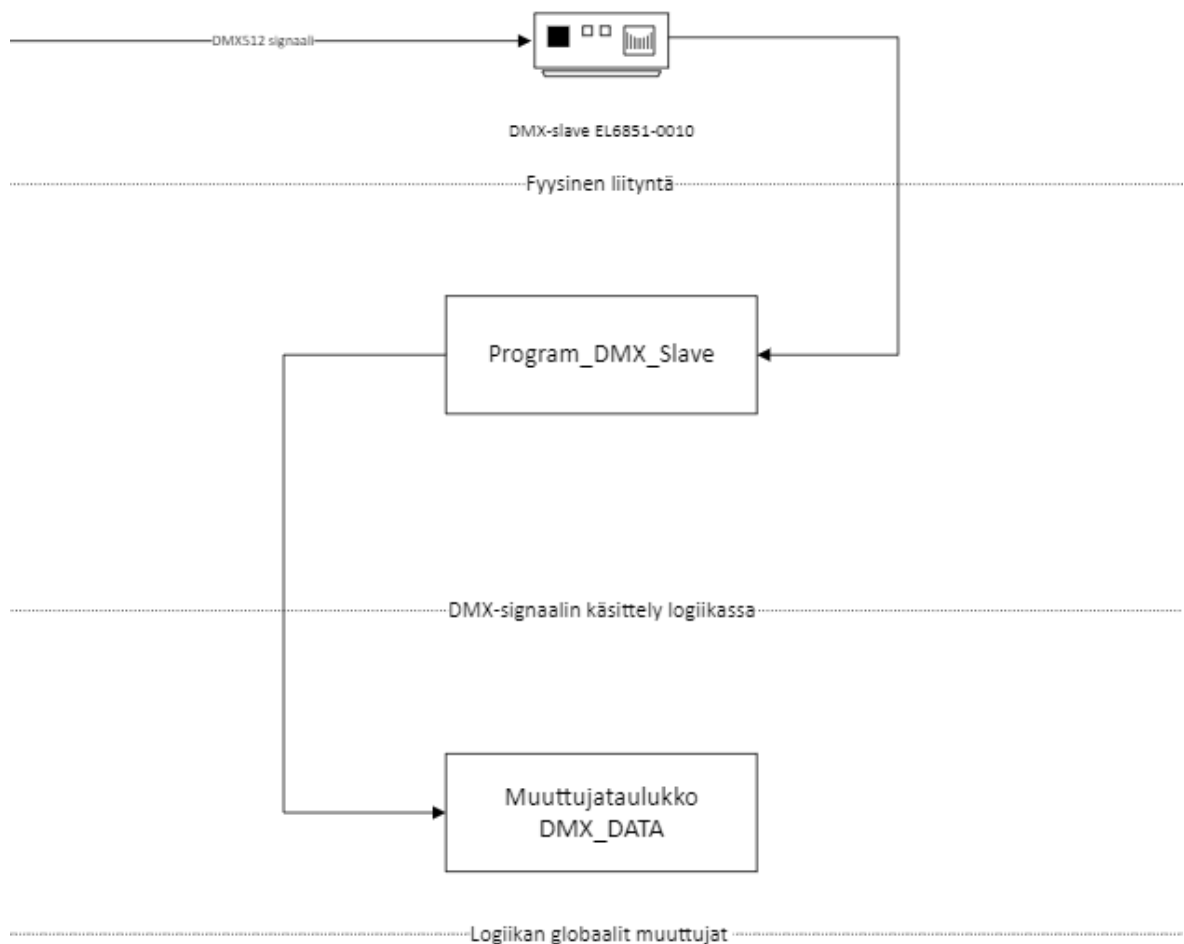
DMX-protokollassa dataa siirretään maksimissaan 513 slotin suuruisissa DMX-kehyksissä. Yksi DMX-kehys on sarja tavuja, DMX-slotteja, joista kukin sisältää kahdeksan databittiä ja lisäksi ohjausbitteinä on yksi start-bitti ja kaksi stop-bittiä. Yhdessä DMX-kehyksessä voi olla näitä slotteja 513 kappaletta, joista ensimmäinen alustaa sanoman ja loput 512 kappaletta sisältävät ohjausdataa. Kehyksessä on seuraavanlainen datarakenne:

1. Tauko, 92 mikrosekuntia matalan tason signaalia.
2. Merkki tauon jälkeen, 12 mikrosekuntia korkean tason signaalia.
3. Ensimmäisen slotin aloitus, START Code 0, määrittää datapaketin sisällön
 - a. Start-bitti
 - b. Start Code (8-bittinen arvo sanoman sisällön määrittämiseksi)
 - c. Ensimmäinen stop-bitti
 - d. Toinen stop-bitti
4. Tauko ennen ensimmäistä datapakettia, 2 mikrosekuntia
5. Maksimissaan 512 slottia ohjausdataa, toistuu n-kertaa.
 - a. Start-bitti
 - b. 8-bittinen slotin arvo
 - c. Ensimmäinen stop-bitti
 - d. Toinen stop-bitti
 - e. Tauko ennen seuraavaa datapakettia, 2 mikrosekuntia

Start-bittiä käytetään osoittamaan, että tietoliikennepaketti alkaa ja varmistamaan, että vastaanottava laite tunnistaa datansiirron alkamisen. DMX Start Code arvo kertoo

järjestelmälle, millainen sanoma väylälle on tulossa. DMX-standardissa on varattu käyttö-tarkoitus kaikille 255 arvolle, jotka 8-bittinen sanoma voi sisältää. (Professional Lightning and Sound Association, 2023) Yleisin väylälle lähetettävä Start Coden arvo on heksadesimaaliluku 00h, joka osoittaa sanoman sisältävän valaisimien ohjaukseen tarkoitettua dataa. Suurin osa ohjaavista ja vastaanottavista laitteista tunnistaa ainoastaan tämän arvon, eivätkä ne reagoi mitenkään muihin Start Code arvoihin. Stop-bitit kuuluvat olennaisena osana sarjapohjaisiin protokolliin, ja niiden tarkoituksena on ilmoittaa tiedonsiirron lopetuksesta ja varmistaa, että vastaanottava laite tunnistaa sen. DMX-master lähettää aina uuden kehyksen edellisen kehyksen lähetyksen päätyttyä, vaikka kehyksen sisältö ei muuttuisikaan edellisestä lähetyksestä. (Fairhurst, 2020)

Beckhoffilla on ohjelmakirjasto Tc2_DMX, EL6851-laitteen alustamiseksi ja käyttämiseksi, ja se sisältää toiminnot DMX-väylän kommunikointiin sekä DMX-sanoman kääntämisen EtherCAT ympäristöön ja siitä takaisin DMX-protokollaan. Jokainen järjestelmään liitetty EL6851 master -laite tarvitsee oman kutsunsa ohjelmakirjastossa olevalta funktioblokilta FB_EL6851CommunicationEx. Slave -laitteen tulot määritellään sanomaa lukevassa ohjelmakoodissa tulomuuttujiksi ja nämä muuttujat linkitetään EL6851-0001 laitteen tuloihin, jonka jälkeen slave -laite voi vastaanottaa sanomia DMX-järjestelmästä. Beckhoffissa vastaanotettu 512 tavun kokoinen DMX-sanoma jaetaan EL6851-0001 laitteen ominaisuuksien vuoksi 64 tavun suuruisiin osiin, ja kutakin näistä 64 tavun suuruista osista täytyy käsitellä omalla ohjelmakoodin osuudellaan.



Kuva 6 Tulevan DMX-signaalin käsittelyyn liittyvät toiminnot

Tulevan sanoman alku tunnustetaan EL6851-0001-laitteen Receive Toggle -tulon avulla. Receive Toggle saa arvon 1 aina, kun vastaanotetaan DMX-kehyksen Start Code -osuus. Sanoman sisältö luetaan tällöin EL6851-laitteen sisäiseen rekisteriin ja siirretään ohjelmakoodin MEMCPY-käskyllä laitteen rekisteristä logiikan muistiin. Alla olevassa ohjelmakoodissa kopiointi tapahtuu 200 millisekunnin välein Receive Toggle -bitin ohjaamana. Ohjelmalla DMX-Slave luetaan vastaanotetusta DMX-sanomasta ensimmäiset 64 kanavaa ja siirretään niistä saatu tieto globaaliin muuttujataulukkoon DMX_DATA. Tämä muuttujataulukko on kaikkien logiikan ohjelmien ja funktioiden käytettävissä, ja siitä saadaan siirrettyä tietoa kaikkiin logiikkaan liitettyihin järjestelmiin.

Ohjelma 1 DMX järjestelmästä lukeminen

```

PROGRAM DMX_Slave
VAR
    DMX_Sl_LEN      AT %I*      : INT;
    DMX_Sl_1_64    AT %I*      : ARRAY [1..64] OF BYTE;
    DMX_Sl_Toggle  AT %I*      : BOOL;
    result          : UDINT;
    DMX_DATA        : ARRAY [1..64] OF BYTE;
    fbMonitorToggleBit : DMX_Monitor_Toggle_Bit;
    bGetData        : BOOL;
END_VAR

fbMonitorToggleBit(bToggleBit := DMX_Sl_Toggle, tWatchdogTime := T#200MS, bWatchdog=> );

IF ((DMX_Sl_LEN = 64) AND (NOT fbMonitorToggleBit.bWatchdog)) THEN
    result := MEMCPY(ADR(DMX_DATA), ADR(DMX_Sl_1_64), 64);
    bGetData :=TRUE;
ELSE
    result := MEMSET(ADR(DMX_DATA), 0, 64);
    bGetData :=FALSE;
END_IF

```

Taulukkomuotoisessa tietorakenteessa solu 1 sisältää yhden DMX universumin kanavalle 1 osoitetun datan, solu 2 DMX-kanavan 2 datan ja niin edelleen. Mikäli lähetetty sanoma ei sisällä kaikille 512 solulle arvoa, tallentuu näiden loppujen solujen arvoksi 0 (Beckhoff Information System, 2023). Jos DMX lähetys syystä tai toisesta keskeytyy, täytetään taulukko 0-arvoilla, jotta vältetään virheellisiltä ohjauskomennoilta muihin järjestelmiin. Tämä toteutettiin EL6851-laitteen Receive Toggle -tulon avulla, sekä lukemalla DMX-kanavaa 64, joka saa tässä tapauksessa ohjausarvonsa teatterin valopöydän ollessa toiminnassa. Tätä kanavaa käytettiin myös valopöydästä tulevan master -ohjauskomentona. Valopöydästä lähetetään kanavaan 64 yli 0 olevaa arvoa, kun ohjaukset DMX-puolelta ovat käytössä. Mikäli tämän arvo menee nolleen, ja Receive Toggle ei vastaanota tietoa vähintään 200 millisekunnin välein, niin DMX-sanomien välitys keskeytetään rajapintajärjestelmästä eteenpäin.

3.2 DMX-väylälle kirjoittaminen

Väylälle kirjoittamisessa käytetään ohjelmakirjastossa olevaa funktioblokkia FB_EL6851CommunicationEx. Funktioblokki käyttää globaaleissa muuttujissa olevaa taulukkomuotoista muuttujaa arrProcessData, jonka sisältö kirjoitetaan EL6851 laitteelle jokaisessa logiikan syklissä.

Ohjelma 2 DMX ohjaus

```

PROGRAM DMX_Control

IF DMX_Slave_1.bGetData AND DMX_Slave_1.DMX_DATA[64]>0 THEN
    TableControl :=TRUE;
ELSE
    TableControl :=FALSE;
END_IF

IF TableControl THEN
    GVL_DMX.arrProcessData0[79]:=DMX_Slave_1.DMX_DATA[1];
    GVL_DMX.arrProcessData0[80]:=DMX_Slave_1.DMX_DATA[2];
    GVL_DMX.arrProcessData0[81]:=DMX_Slave_1.DMX_DATA[3];
    GVL_DMX.arrProcessData0[82]:=DMX_Slave_1.DMX_DATA[4];
    byWhite:=DMX_Slave_1.DMX_DATA[44];

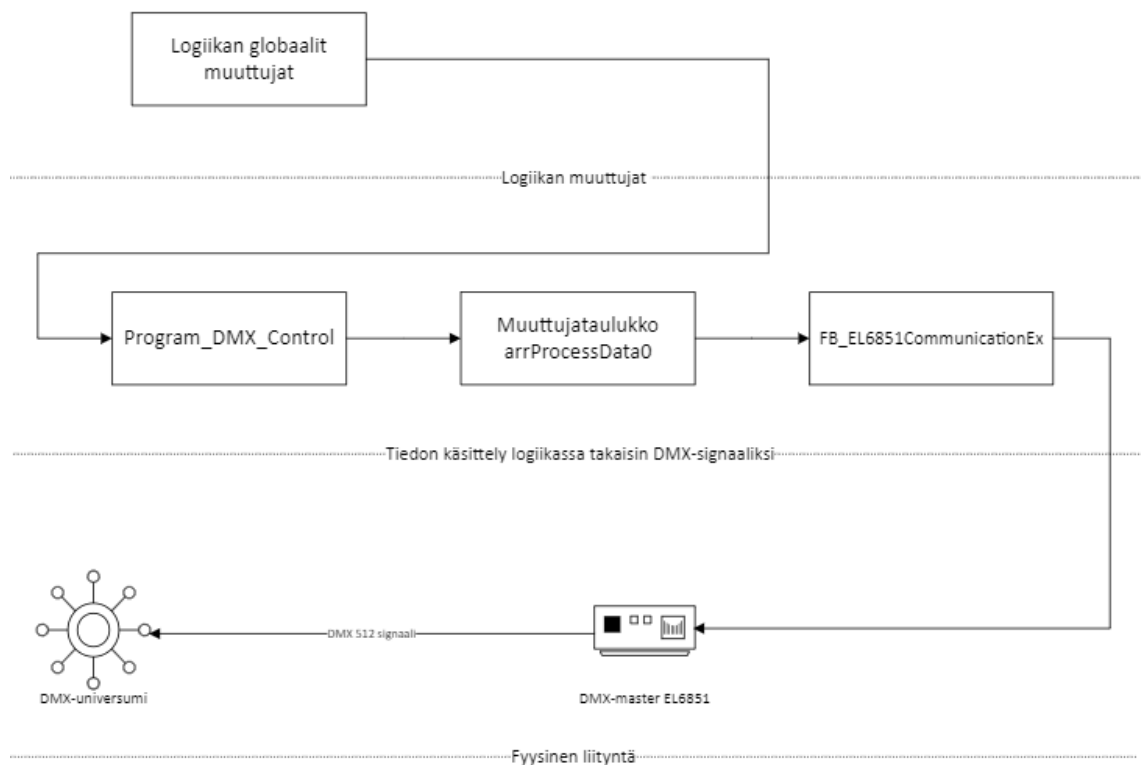
ELSE

    FOR counter0:=1 TO 511 BY 1 DO
        GVL_DMX.arrProcessData0[iChannel0] := byKatsomo;
        GVL_DMX.arrProcessData1[iChannel0] := byKatsomo;
        GVL_DMX.arrProcessData2[iChannel0] := byKatsomo;
        GVL_DMX.arrProcessData3[iChannel0] := byKatsomo;
        iChannel0:= iChannel0+1;
    END_FOR

END_IF

```

Esimerkki väylälle kirjoittamisessa käytetystä ohjelmakoodista on yllä olevassa ohjelmassa 2. Tässä TableControl-muuttujaa ohjataan DMX-kanavalla 64 sekä bGetdata-muuttujalla. Mikäli näille annetut ehdot toteutuvat, voi valopöytä ohjata kaikkia järjestelmään liitettyjä laitteita. Yllä olevassa ohjelmakoodissa ohjataan katsomon DMX-valaisimia valopöydästä tulevilla ohjauskomennoina. DMX_Slave1-laitteen tulon arvot siirretään globaalien muuttujataulukoiden kautta DMX-master laitteen lähtöön 0. Mikäli TableControl-muuttujan arvo on epätosi, täytetään DMX-master laitteen kaikki ohjauskanavat ennalta määrättyyn arvoon byKatsomo-muuttujan avulla, kyseinen muuttuja saa arvonsa sekä KNX-järjestelmästä, että Beckhoff-järjestelmän ohjauspaneelilta. TableControl muuttujaa käytettiin myös master-ohjauskomentona, jolloin valopöydästä tuleva ohjauskomento ohittaa tässä esimerkkitapauksessa kaikista muista järjestelmistä kyseisiin valaisimiin tulevat ohjauskomennot.



Kuva 7 Lähtevän DMX-signaalin käsittelyyn liittyvät toiminnot

Logiikan globaaleista muuttujista voidaan ottaa mitä tahansa tietoa ohjelman DMX_Control käyttöön. Tällä ohjelmalla siirretään halutut tiedot globaaleista muuttujista muuttujatauluk-
koon arrProcessData0. Tämä muuttujataulukko on DMX-masterin lähdön muuttuja. Ja se
liittyy suoraan funktioblokkiin FB_EL6851CommunicatioEx. Kyseisen muuttujataulukon
sisältö kirjoitetaan DMX-väylälle jokaisessa logiikan syklistä.

3.3 KNX-protokollan käsittely Beckhoff-logiikalla

KNX-väylä liitetään Beckhoff-logiikkaan KL6301-väyläsovittimen avulla. Väyläsovitin toi-
mii KNX TP-topologialla ja se liitetään osaksi KNX-järjestelmän väyläkokonaisuutta. Toi-
sin kuin DMX-puolella, jossa slave-laite voi vain kuunnella suoraan väylällä liikku-
vaa tietoliikennettä, KNX-väyläsovittimen lisääminen järjestelmään vaatii sen huomioimista
myös KNX-ohjelmoinnin puolella. Ohjaava KNX-sanoma sisältää enemmän tietoa ja se on
kohdennettu aina jollekin laitteelle, joten sen lukeminen väylältä vaatii tietoa KNX-järjes-
telmän ohjelmoinnista ja toimintojen ryhmäosoitteista, ja myös väyläsovitin on huomioitava

KNX-ohjelmoinnissa. Beckhoffissa KNX-väylän liityntää ohjelmapuolella käytetään Tc2_EIB – ohjelmakirjastoa, joka sisältää tarvittavat toiminnot KNX-järjestelmän kanssa kommunikointiin. Kommunikoinnin perustana on ohjelmakirjastossa oleva, suoraan sovittinta ohjaava KL6301-funktioblokki. Tätä KL6301-funktiota pitää kutsua yhdelle järjestelmään liitettyllä sovittimelle vain kerran PLC-syklin aikana, tämä on otettava huomioon ohjelmointia tehdessä.

Ohjelma 3 KL6031 laitteen alustus

```

PROGRAM KNX_Communication
VAR
    i                : INT;
    KL6301_0         : KL6301;
    EIB_BIT_REC_0    : EIB_BIT_REC;
    KNX_Control      : KNX_Control;
    stGroupaddress   : EIB_GROUP_ADDR;

END_VAR

CASE i OF
0:
    KL6301_0.EIB_PHYS_ADDR.Area:=1;
    KL6301_0.EIB_PHYS_ADDR.Line:=1;
    KL6301_0.EIB_PHYS_ADDR.Device:=21;
    KL6301_0.EIB_GROUP_FILTER[1].GROUP_ADDR.MAIN:=0;
    KL6301_0.EIB_GROUP_FILTER[1].GROUP_ADDR.SUB_MAIN:=0;
    KL6301_0.EIB_GROUP_FILTER[1].GROUP_ADDR.NUMBER:=1;
    KL6301_0.EIB_GROUP_FILTER[1].GROUP_LEN:=63;

    KL6301_0.bActivate:=TRUE;

    IF KL6301_0.bReady THEN
        IF NOT KL6301_0.bError THEN
            i:=1;
        ELSE
            i:=2;
        END_IF
    END_IF
1:
    ;
2:
    ;
END_CASE

KL6301_0(
    idx:=1,
    iMode:=0,
    KL6301_IN:= GVL.KL6301_IN_0,
    KL6301_OUT:= GVL.KL6301_OUT_0,
    str_Data_Rec => GVL.stDataRec );

IF i = 1 THEN
    KNX_Control();
END_IF

```

Yllä olevalla koodilla alustetaan KL6301-sovitin ja asetetaan se kuuntelemaan KNX-ryhmäosoitteita 0.0.1–0.0.63 sekä 0.1.1–0.1.63. Koko käytettävä osoiteavaruus on määritettävä sovittimen alustusvaiheessa. Kun sovitin on käynnistynyt, se ottaa tietoa vastaan ainoastaan

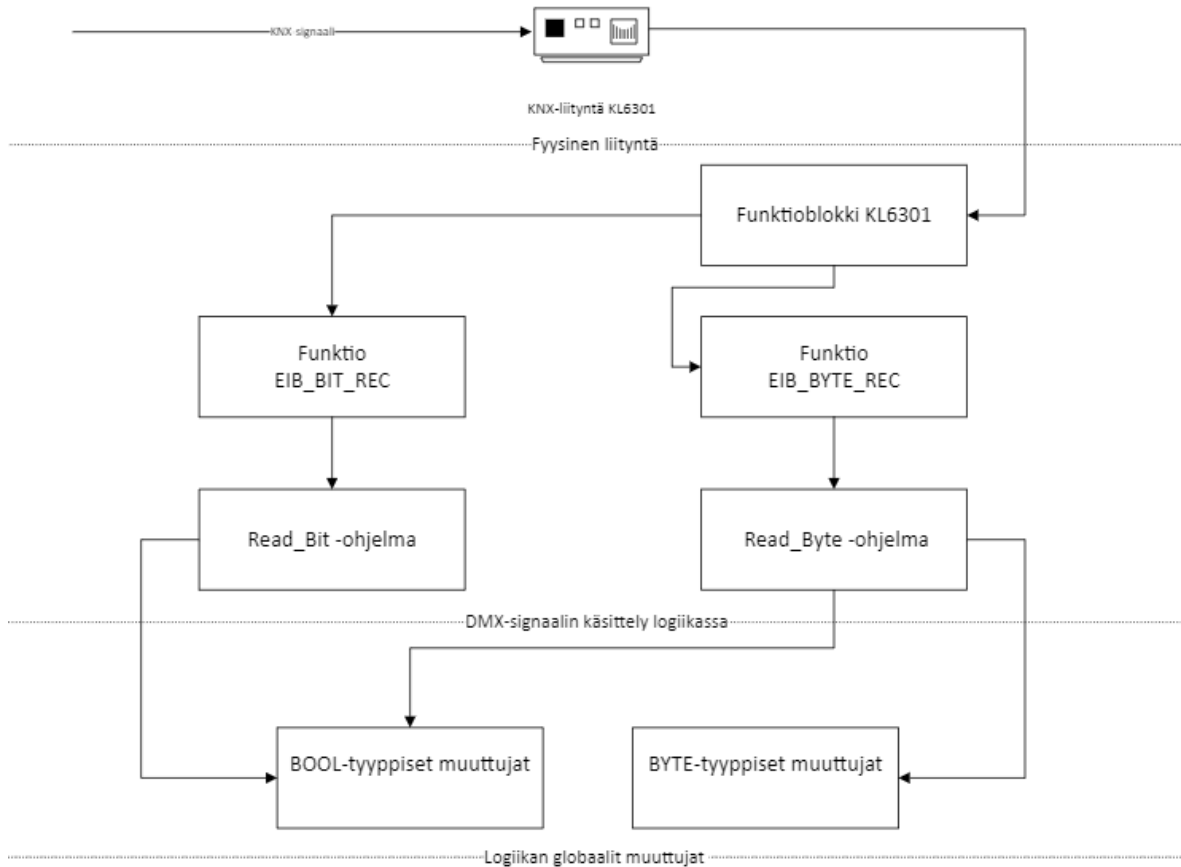
näistä ryhmäosoitteista. Yksi sovitin voi lukea 256 KNX-ryhmäosoitetta. Sama sovitin voi myös lähettää tietoa KNX-väylään, mutta tiedon lähettämistä varten osoitevaruutta ei tarvitse määritellä etukäteen, vaan tietoa voidaan lähettää mihin tahansa ryhmäosoitteeseen. (Beckhoff Information System, 2023).

Ohjelmakoodin CASE-OF-rakenteella varmistetaan, että KL6301-blokki aktivoidaan vain kerran PLC-syklin aikana, ja että aktivointi onnistuu ennen kuin dataa yritetään siirtää. Aktivointi tapahtuu `KL6301_0.bActivate:=TRUE` käskyllä. Mikäli laite aktivoituu oikein ja siinä on KNX-väylä kytkettynä, se ilmoittaa olevansa valmis lähettämään ja vastaanottamaan dataa `bReady`-bitin avulla. Kun järjestelmässä käytetään useampaa kuin yhtä KL6301-sovitinta, kullekin tulee antaa oma yksilöllinen `idx`-numero, kuvan 3 tapauksessa `idx`-numeroksi on määritetty 1. Laiteelle myös määritetään käytettävä ryhmäosoiteavaruus `iMode`-tulon avulla. Tuloon voidaan asettaa arvot 0, 1, 2 tai 100. Arvolla 0 laite voi kuunnella 256 ryhmäosoitetta neljästä eri keskiryhmästä, eli näiden keskiryhmien 64 ensimmäistä ryhmäosoitetta. Arvoilla 1 ja 2 laite voi kuunnella 256 ryhmäosoitetta kahdeksasta keskiryhmästä, eli 32 ryhmäosoitetta kustakin keskiryhmästä. Arvolla 100 laite menee monitorointitilaan ja kuuntelee kaikkea väylällä liikkuvaa dataa, mutta ei kuittaa mitään sanomaa vastaanotetuksi, tässä moodissa laite ei voi myöskään lähettää väylälle mitään. Tässä tapauksessa arvoksi on valittu 0, koska on tarkoitus lukea kahden keskiryhmän kaikkia 64 osoitetta, eivätkä moodit 1 ja 2 mahdollista kuin 32 ryhmäosoitteen lukemisen yhdestä keskiryhmästä.

Tämän kaltainen valinta täytyy tehdä aina järjestelmäkohtaisesti ja se on riippuvainen KNX-järjestelmän ohjelmoinnista. Koska tämä rajoittaa järjestelmän käsittelemien ryhmäosoitteiden määrää, niin mikäli käsiteltäviä ryhmäosoitteita on yhden KL6301-sovittimen tukemaa määrää enemmän, tarvitsee järjestelmään liittää useampia sovittimia. (Beckhoff Information System, 2023)

Tässä kohteessa käytettiin tästä syystä yhteensä neljä KL6301-sovitinta. Koska kutakin sovitinta voi ja täytyy kutsua kerran yhden PLC-syklin aikana, aiheutti tämä pieniä haasteita lukemisen ja kirjoittamisen ajoitusten kanssa. Sovittimista saatavan tiedon lukemisessa tämä ei aiheuttanut erityisemmin ongelmia, mutta väylälle kirjoittaminen epäonnistui toisinaan, kun laite oli mahdollisesti varattuna väylän lukemiseen. Kirjoittamiseen olisi voinut, ja mahdollisesti olisi myös pitänyt, käyttää omaa sovitinta, mutta ongelmaa kierrettiin sillä, että KNX-järjestelmä ohjelmoitiin kuittaamaan järjestelmän kannalta oleelliset ohjaukset omalla sanomallaan. Mikäli `bReady`-bitti saadaan, eikä sovitin ole muusta syystä

virhetilassa saadaan i-muuttujan arvoksi 1 ja ohjelmakoodi käynnistää KNX-väylän luenta-toiminnot.



Kuva 8 KNX-järjestelmän luentaan käytettävät toiminnot

Logiikassa käytetään kaikkien erilaisten KNX-sanomamuotojen lukemiseen aina omaa funktioblokkiaan, jolla voidaan lukea vain kyseistä yhdentyyppistä sanomaa, jonka lukemiseen se on tehty. Järjestelmä ei kykene lukemaan KNX-väylältä niin sanottua raakadataa, vaan on aina tiedettävä ja valittava käyttöön tarvittavien sanomatyyppien mukaiset funktioblokit. Erilaiset KNX-sanomat sisältävät eri määrän tietoa, ja kun tiedetään mitä tietoa halutaan lukea ja missä muodossa se esiintyy, ei ohjelmakoodissa tarvitsekaan varautua vastaanottamaan kaikkia mahdollisia sisältöjä, vaan funktioblokin sisältö on voitu tehdä vastaanottamaan vain se tietyn tyyppinen sanomarakenne, tämä yksinkertaistaa funktioblokin toimintaa huomattavasti, mutta on toisaalta myös kankeampi ratkaisu, kuin sellainen, jossa voitaisiin lukea suoraan kaikkea KNX-väylällä liikkuvaa dataa, ja erotella sieltä tarvittavat tiedot. Tähän vaikuttaa osaltaan se, että Beckhoff-järjestelmässä kutakin tietotyyppiä varten

tarvitaan omantyyppisensä muuttuja, eikä kaikkiin muuttujatyyppeihin voida tallentaa kaikkea väylältä saatavaa erilaista tietoa.

KNX-järjestelmässä on kullekin sanomalle määritetty oma tietotyyppinsä, DPT, data point type, ja sanoman sisältö tunnustetaan tämän tietotyypin perusteella. Tämä vaikuttaa erityisesti ohjaustiedoltaan yhden tavun ja sitä suurempien sanomien sisällön tulkintaan. Esimerkiksi yhden tavun sanoman sisältö voi olla ohjausarvo välillä 0–225 tai -127–128, ilman DPT-määritystä sanoman sisällöstä ei tiedettäisi, että sanoman sisällön tulee skaalautua myös negatiivisen asteikon puolelle.

Ohjelma 4 Bittimuotoisen tiedon lukeminen KNX-järjestelmästä

```

FUNCTION_BLOCK KNX_Read_Bit
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
    EIB_BIT_REC_0          : EIB_BIT_REC;
    stGroupaddress        : EIB_GROUP_ADDR;
    bDataReceive           : BOOL;
    counter                : INT;
    bData                  : BOOL;
END_VAR

stGroupaddress.MAIN:=0;
stGroupaddress.SUB_MAIN:=0;
FOR counter:=1 TO 10 BY 1 DO
    stGroupaddress.NUMBER:=INT_TO_BYTE(counter);
    EIB_BIT_REC_0(Group_Address:=stGroupaddress,
        strData_Rec:= GVL.stDataRec, bDataReceive=> bDataReceive,
        bData=>bData);
    IF bDataReceive = FALSE THEN;
        CASE counter OF
            1: GVL.KNX001_data :=bData;
            2: GVL.KNX002_data :=bData;
            3: GVL.KNX003_data :=bData;
            4: GVL.KNX004_data :=bData;
            5: GVL.KNX005_data :=bData;
            6: GVL.KNX006_data :=bData;
            7: GVL.KNX007_data :=bData;
            8: GVL.KNX008_data :=bData;
            9: GVL.KNX009_data :=bData;
            10: GVL.KNX0010_data :=bData;
        END_CASE
    END_IF
END_FOR

```

Bittimuotoisen tiedon lukemiseen käytetään valmista EIB_BIT_REC funktioblokkia. Blokin tulomuuttujiin määritetään KNX-ryhmäosoiteavaruus, jota halutaan kuunnella. Tässä voidaan käyttää vain KL6301-laitteen alustamisen yhteydessä määritettyä osoiteavaruutta. Yllä olevassa ohjelmakoodissa pääryhmäksi on määritetty 1, keskiryhmäksi 0 ja keskiryhmästä luetaan osoitteita yhdestä kolmeentoista. KL6301-sovitin voi vastaanottaa vain yhden

sanoman yhden PLC-syklin aikana. Kun laite vastaanottaa KNX-sanoman, sen lähtömuuttuja bDataReceive saa arvon FALSE yhden PLC-syklin ajaksi. Vastaanotetun sanoman informaatio tallentuu bData-lähtömuuttujaan myös vain yhden PLC-syklin ajaksi ja tieto täytyy saada siirrettyä pysyvämpään muuttujaan tämän kyseisen syklin aikana. (Beckhoff Information System, 2023)

Ohjelman 4 koodissa tiedon tallentaminen tapahtuu FOR-silmukan ja CASE OF -rakenteen avulla. FOR-silmukalla luetaan jatkuvasti ryhmäosoitteita 1–10, ja mikäli näistä vastaanotetaan sanoma sekä bDatareceive saa arvon FALSE, tallennetaan bData-muuttujan tieto globaaleihin muuttujiin CASE-OF-rakenteella. Koska KNX-väylä on suhteellisen hidas, ehditään tällä menetelmällä tallentamaan kaikki väylällä siirtyvä haluttu data, kun käytetään nopeaa, esimerkiksi 5 millisekunnin PLC-sykliäikää.

Ohjelma 5 Tavumuotoisen tiedon lukeminen KNX-järjestelmästä

```

FUNCTION_BLOCK KNX_Read_Byte
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
    EIB_3bit_cont           : EIB_3BIT_CONTROL_REC;
    stGroupaddress         : EIB_GROUP_ADDR;
    bDataReceive           : BOOL;
    counter                : INT;
    byRange                : BYTE;
    bControl               : BOOL;
END_VAR
stGroupaddress.MAIN:=0;
stGroupaddress.SUB_MAIN:=1;
FOR counter:=1 TO 3 BY 1 DO
    stGroupaddress.NUMBER:=INT_TO_BYTE(counter);
    EIB_3bit_cont(Group_Address:= stGroupad-
        dress, strData_Rec:= GVL.stDataRec, bDataReceive=>
        bDataReceive, bControl=> bControl,byRange=> byRange );
    IF bDataReceive = FALSE THEN;
        CASE counter OF
            1: GVL.KNX011_data :=byRange;
              GVL.KNX011_data_bool :=bControl;
              END_CASE
        END_IF
    END_FOR
END_BLOCK

```

KNX-järjestelmässä esimerkiksi valaistuksen tason säätämiseen käytetään yleensä 4-bittiä ohjausdataa sisältävä sanomatyyppejä. Tässä sanomassa on varattuna 3-bittiä säädön suuruudelle ja yksi bitti suunnalle. Yllä olevalla ohjelmalla luetaan tällaista sanomaa. Toimintaperiaate on hyvin samanlainen, kuin yhden bitin sanomaa luettaessa. Lukemiseen käytetään valmista funktioblokkia EIB_3BIT_CONTROL_REC. Kolmen ohjausarvoa sisältävän bitin

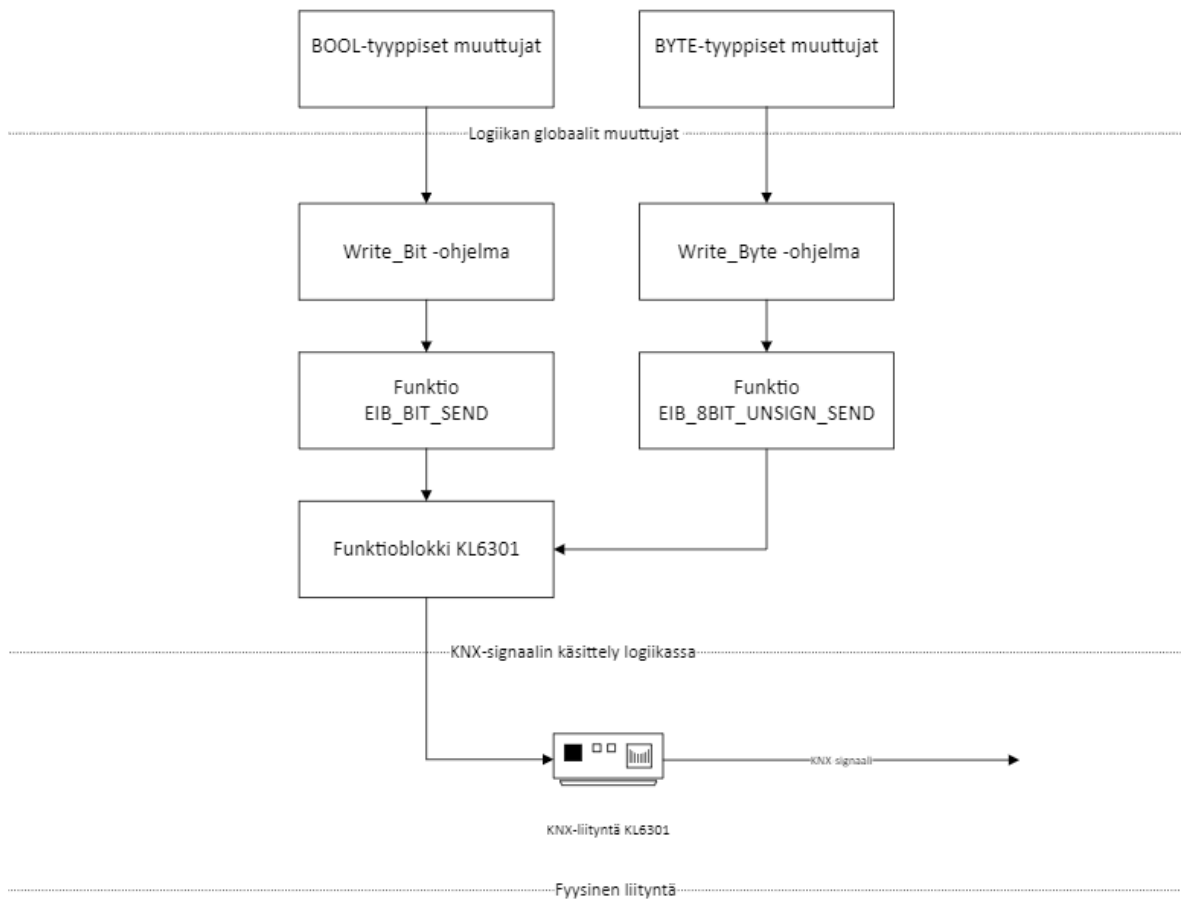
arvot saadaan tallennettua byRange- ja suunta bControl-muuttujan avulla. Tallentamiseen käytetään globaaleja muuttujia, jotta arvot ovat helposti käytettävissä kaikissa eri aliohjelmassa. Muuttujan nimeen on sisällytetty KNX-ryhmäosoitteen arvo, tällä menetelmällä pyydytään hyvin selvillä siitä, mitä tietoa mikäkin muuttuja Beckhoff-järjestelmässä sisältää, ja helpotetaan tiedonvaihtoa sekä Beckhoff- että KNX-ohjelmoinnin puolella.

Ohjelma 6 Ohjaus KNX-järjestelmästä DMX järjestelmään

```
F_trig1(CLK:=GVL.KNX004_data); // Kaikki katsomon tyoivalot Off
F_trig5(CLK:=bKatsomoval);
    IF F_trig1.Q AND GVL.KNX_Enabled OR F_trig5.q THEN
        bKatsomo1val:=FALSE;
        bKatsomo3val:=FALSE;
        bKatsomo4val:=FALSE;
        bKatsomo5val:=FALSE;
        bPortaalival:=FALSE;
    END_IF

R_trig1(CLK:=GVL.KNX004_data); // Kaikki katsomon tyoivalot On
R_trig5(CLK:=bKatsomoval);
    IF R_trig1.Q AND GVL.KNX_Enabled OR R_trig5.Q THEN
        bKatsomo1val:=TRUE;
        bKatsomo3val:=TRUE;
        bKatsomo4val:=TRUE;
        bKatsomo5val:=TRUE;
        bPortaalival:=TRUE;
    END_IF
```

Kun KNX-väylästä luettu tieto on saatu tuotua Beckhoff-järjestelmän sisäisiin muuttujiin, voidaan tietoa hyödyntää suoraan kirjoittamisessa kaikkiin liitettyihin järjestelmiin. Yllä olevassa ohjelmakoodin pätkässä KNX-järjestelmästä saadulla bittimuotoisella tiedolla ohjataan samoja valaisimia, kuin aiemmin esitettyä DMX-järjestelmästä saadulla tiedolla. Ohjauksissa käytetään nousevan ja laskevan reunan liipaisufunktiota, jotta kustakin järjestelmästä huomioidaan vain viimeisin saatu ohjaustieto.



Kuva 9 KNX-väylälle kirjoittamiseen käytettävät toiminnot

KNX-väylälle kirjoittamiseen on valmistajan puolelta suoraan oma funktioblokkinsa, jota kutsumalla KL6301 laite kirjoittaa pyydetyn mallisen sanoman KNX-väylälle. Lähettämistä varten tarvitsee määritellä vain ryhmäosoite sekä sanoman sisältö. Alla olevalla koodilla lähetetään yksibittistä sanomaa, eli arvoja 0 tai 1, KNX pääryhmän 0, keskiryhmän 1 ja vapaavalintaisen alaryhmän osoitteeseen. Funktiota kutsuessa määritellään BNumber sekä bData muuttujille arvo, jotka EIB_BIT_SEND funktioblokki sitten lähettää eteenpäin. Lähetys voi tapahtua aina vain yhden kerran yhdessä logiikan syklissä.

Ohjelma 7 KNX-järjestelmään kirjoittaminen

```

FUNCTION_BLOCK KNX_Write_Bit
VAR_INPUT
    BNumber      : BYTE;
    bData        : BOOL;
END_VAR
VAR_OUTPUT
END_VAR
VAR
    EIB_BIT_SEND_0      : EIB_BIT_SEND;
    stGroupaddress      : EIB_GROUP_ADDR;
    iErrorID            : Tc2_EIB.EIB_ERROR_CODE;
    bError              : BOOL;
END_VAR

stGroupaddress.MAIN:=0;
stGroupaddress.SUB_MAIN:=1;
stGroupaddress.NUMBER:=BNumber;
EIB_BIT_SEND_0(Group_Address:=stGroupaddress, bData:=bData,
str_Rec:=GVL.stDataRec,bError=>bError,          iErrorID =>iErrorID);

```

Lähetysten onnistumista ei voida mitenkään monitoroida Beckhoffin päästä, vaan tiedon siirtyminen täytyy varmistaa monitoroimalla KNX-järjestelmää, mikäli halutaan varmistua tiedon siirtymisestä. Tämä aiheutti ohjelmoinnin ja käyttöönoton aikana hieman ylimääräistä työtä, kun tiedonsiirron varmistamiseen täytyi käyttää kahta järjestelmää. Myös lopputuloksessa jouduttiin käyttämään KNX-järjestelmästä luettavaa kuittausta, siitä saatiinko joku tärkeäksi luokiteltu tieto siirrettyä kerralla, vai yritettiinkö sitä Beckhoffin puolelta uudelleen. Mikäli kuittausta ei saatu KNX-järjestelmää luettaessa, lähetystä yritettiin uudelleen.

Myös tässä tiedon lähettämisessä KNX-järjestelmään on huomioitava, että KL6301 laitetta voidaan kutsua vain kerran PLC-syklin aikana, ja sen aktivoituminen tulee varmistaa ennen lähetystä. Tämä on sisällytetty aiemmin esitellyn KNX_Control funktion alle. Tämä funktio ei voi käynnistyä, ellei KL6301-laite ole ilmoittanut olevansa valmis keskustelemaan KNX-väylän kanssa. Käytännössä ainoa este KL6301 -laitteen käynnistymiseen, on virhe Beckhoffin puolella ohjelmoinnissa, tai syystä tai toisesta puuttuva KNX-väylä, joten tilanteet, joissa laite ei käynnistyisi, ovat hyvin harvinaisia.

3.4 BACnet/IP integraatio Beckhoffilla

Beckhoffissa BACnet/IP otetaan käyttöön lisäämällä ohjelmaan ohjelmakirjasto TcBACnet-Rev12.lib. Tämän käyttöönoton lisää automaattisesti järjestelmän I/O puolelle BACnet/IP-oletussovittimen sekä BACnetClientCalls -ohjelman, johon sitten lisätään tarvittavia funktioblokkeja eri toimintojen käyttöönottamiseksi. Tämä ominaisuus aiheuttaa hieman sekavuutta, koska järjestelmään täytyy lisätä BACnetClientCalls -ohjelmaan koodipätkiä, joilla ei ole vaikutusta olevan muuta käyttöä, kuin osoittaa järjestelmän BACnet laitteelle tuettavia toimintoja. On kuitenkin tärkeää ymmärtää, että nämä BACnetClientCalls -ohjelman koodinpätkät eivät ole turhia, vaan ne määrittävät järjestelmän BACnet-laitteille tuettavat toiminnot. Vaikka tämä saattaa tuntua sekavalta, se on välttämätöntä järjestelmän oikean toiminnan kannalta. Alla olevan ohjelmakoodin esimerkissä on otettu käyttöön BACnetin MultiState Value ja Output -objektien lukeminen ja kirjoittaminen. Tässä tapauksessa Multistate Output -objektilla siirretään tietoa palvelimen suuntaan ja MultiState Value -objektilla vastaanotetaan tietoa palvelimelta. Tällä objektilla voidaan välittää arvot väliltä 1–256, joiden ominaisuudet ohjaavassa järjestelmässä voidaan määrittellä vapaasti.

Ohjelma 8 BACnet MultiState-objektien käyttöönotto ohjelmakoodilla.

```
(** Object1 (BACnet Client) calls **)
BN_MultistateOut(
bEnablePV:= , bNullPV:= , nPV:= , RemoteDevice:=BACnet_remote, bReady=> ,
nPresentValue=> , bOverridden=> , bOutOfService=> , bFault=> , bInAlarm=> ,
bNoOutput=> , bOpenLoop=> , bShortedLoop=> , bOtherFault=> , eEventState=> ,
nActivePrio=> , nNbrOfStates=> , tObjectID=> , bError=> , nErrorId=> );

BN_MultistateValue( bEnablePV:= , bNullPV:= , nPV:= , RemoteDevice:=BACnet_remote ,
bReady=> , nPresentValue=> , bOverridden=> , bOutOfService=> , bFault=> , bInAlarm=> ,
bMultiStateFault=> , bOtherFault=> , eEventState=> , nNbrOfStates=> , tObjectID=> ,
bError=> , nErrorId=> );
```

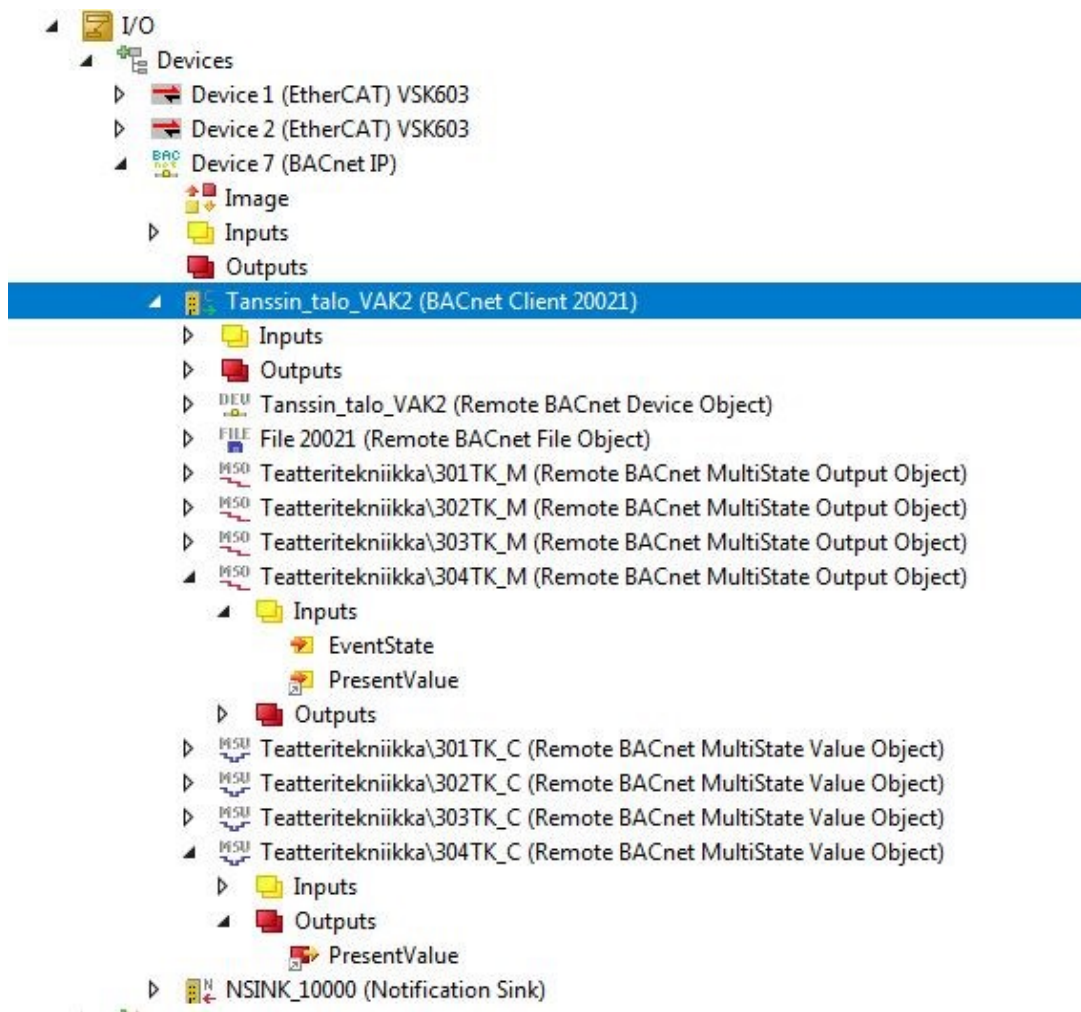
Toiseen BACnet-järjestelmään liityttäessä tulee tietää, liitytäänkö siihen palvelimena vai asiakkaana, järjestelmän konfiguraation rakentaminen riippuu tästä tiedosta. Valinta ei vaikuta järjestelmien välillä siirrettäviin tietoihin, kaikkia tietoja voidaan siirtää riippumatta siitä, onko liittyjä palvelin vai asiakas. Valinta vaikuttaa ainoastaan järjestelmän konfiguraatioon. Tässä tapauksessa kohteen kiinteistöautomaatiojärjestelmään liityttiin asiakkaana. BACnet-asiakkaaksi voidaan liittyä usealla tavalla. Palvelinjärjestelmästä voidaan tallentaa EDE-tiedosto, Engineering Data Exchange, jonka avulla palvelinjärjestelmän tekninen konfiguraatio voidaan siirtää asiakasympäristöön.

EDE-tiedoston käyttö on yksi helpoimmista tavoista liittyä BACnet-asiakkaaksi, koska se mahdollistaa teknisen konfiguraation siirtämisen suoraan palvelinjärjestelmästä asiakasympäristöön. Tämä säästää yleensä merkittävästi aikaa ja vaivaa, koska manuaalista konfigurointia ei tarvitse tehdä. EDE-tiedoston avulla voidaan siirtää kaikki järjestelmän tarvitsemat tekniset tiedot, kuten datapistetyypit, datapisteosoitteet ja datapisteiden esitystiedot standardoituissa muodoissa. Tämän ansiosta järjestelmän konfigurointi on erittäin yksinkertaista ja nopeaa.

EDE-tiedoston käyttö ei ole kuitenkaan ainoa tapa liittyä BACnet-asiakkaaksi. On olemassa myös muita menetelmiä, kuten manuaalinen konfigurointi tai erityisten integraatiotyökalujen käyttö. Valittu menetelmä riippuu useista tekijöistä, kuten järjestelmän monimutkaisuudesta, käytettävissä olevista resursseista ja erityisvaatimuksista.

Kun tässä kohteessa kiinteistöautomaatiojärjestelmään liityttiin asiakkaana, EDE-tiedoston käyttö oli helppo valinta, koska se oli suoraan saatavilla kiinteistöautomaatiojärjestelmän toimittajalta, ja sen käyttö mahdollistikin sujuvan integraation näiden BACnet-järjestelmien välille.

Beckhoff-järjestelmässä BACnet-asiakas lisätään BACnet/IP-laitteen alle, ja sille asetetaan oma yksilöllinen IP-osoite. EDE-tiedoston lataamisen jälkeen yhteys toiseen järjestelmään on parhaassa tapauksessa kunnossa. Kuvan 11 esimerkissä nähdään, kuinka EDE-tiedostoa käyttämällä on Beckhoff-järjestelmään lisätty teatteritekniikkaan vaikuttavat ilmanvaihtokoneet tarvittavine objekteineen kiinteistöautomaation BACnet-serverin konfiguraatiosta. Nämä BACnetin objektit ovat tämän jälkeen suoraan linkitettävissä Beckhoff-järjestelmän I/O-muuttujiin. Järjestelmä näyttää myös niiden reaaliaikaisen tilan suoraan näiden I/O pisteiden ominaisuuksia katsomalla. Niistä voidaan myös lukea sekä niihin voidaan kirjoittaa tietoa reaaliaikaisesti. Koska tässä tapauksessa BACnetin kohde, josta tietoa luetaan ja sitä ohjataan, toimii myös täysin itsenäisesti, täytyy tiedonsiirrossa huomioida tilatiedon välittäminen järjestelmästä toiseen.



Kuva 10 BACnet/IP EDE-tiedostosta saadut I/O-laitteet

MultiState Value- objektilla lähetetään tietoa ilmanvaihtokoneetta ohjaavaan kiinteistöautomaatioon, joka sitten ohjaa ilmanvaihtokoneen haluttuun tilaan. Alla olevassa ohjelmakoodin pätkässä MultiState Value-objektiin lähetetään lukuarvoja 1–7, joiden perusteella kiinteistöautomaatiojärjestelmä suorittaa toimintoja sinne ohjelmoitujen toimintojen perusteella.

Ohjelma 9 BACnet MultiState Value ohjaus

```

IF iMOB_302TK= 1 THEN st302TK :='IV-seis' ; b302_1_status:=TRUE;
ELSE b302_1_status:=FALSE; END_IF
IF iMOB_302TK= 2 THEN st302TK :='Esitys' ; b302_2_status:=TRUE;
ELSE b302_2_status:=FALSE; END_IF
IF iMOB_302TK= 3 THEN st302TK :='Lisäaika'; b302_3_status:=TRUE;
ELSE b302_3_status:=FALSE; END_IF
IF iMOB_302TK= 4 THEN st302TK :='Minimi' ; b302_4_status:=TRUE;
ELSE b302_4_status:=FALSE; END_IF
IF iMOB_302TK= 5 THEN st302TK :='Huuhtelu'; b302_5_status:=TRUE;
ELSE b302_5_status:=FALSE; END_IF
IF iMOB_302TK= 6 THEN st302TK :='Vasemmalle'; b302_6_status:=TRUE;
ELSE b302_6_status:=FALSE; END_IF
IF iMOB_302TK= 7 THEN st302TK :='Oikealle'; b302_7_status:=TRUE;
ELSE b302_7_status:=FALSE; END_IF

```

Kiinteistöautomaatiojärjestelmästä siis pyydetään esimerkiksi lukuarvolla 1 toimintoa 1, jolloin automaatiojärjestelmä ohjaa ilmanvaihtokoneen ennalta ohjelmoituun tilaan 1. MultiState Output- objektin kautta kiinteistöautomaatiojärjestelmä kuittaa pyydetyn tilan Beckhoff-järjestelmään. Beckhoff-järjestelmän iMOB_302TK muuttujaan tallennetaan kiinteistöautomaatiosta saatua tietoa ilmanvaihtokoneen 302 ohjauksesta. Kiinteistöautomaatiojärjestelmä vastaa samoilla lukuarvoilla, 1–7, vahvistaneensa ohjauskomennon. Trig302_X-muuttujia, jotka välittävät numeroarvot eteenpäin, voidaan ohjata mistä tahansa järjestelmään liitetystä laitteesta.

3.5 Järjestelmän tietoturva

Kohteen Beckhoff-logiikka on teollisuus-pc:hen pohjautuva avoin ratkaisu, joka käyttää käyttöjärjestelmänään Windows Embedded 7 versiota. Tämän version virallinen tuki on päättynyt, ja siihen liittyvät tietoturvaongelmat ovat yleisesti tiedossa, joten näitä tietoturva-aukkoja on mahdollista hyödyntää, mikäli ulkopuolinen taho pääsee jollain tavalla kiinni järjestelmään. Käyttöjärjestelmän ja pc-pohjaisen rakenteen vuoksi järjestelmä on siis hyvin altis erilaisille tietoturvauhille, mikäli se liitetään johonkin muuhun kuin täysin suljettuun verkkoon. Kohteen tapauksessa rajapintajärjestelmänä toimiva Beckhoff-logiikka liitettiin BACnet/IP:llä kiinteistöautomaatiojärjestelmään, joka taas on etäkäyttöominaisuuksien vuoksi yhteydessä internetiin. Automaatiojärjestelmän ja rajapintajärjestelmän väliseen liityntään käytettiin kiinteistön yhteistä tietoverkkoa.

BACnet/IP protokollassa ei ole salausta ja sen tietoliikenteeseen käyttämä portti on yleisesti hyvin tiedossa, joten pääsy kiinteistön tietoverkon kautta rajapintajärjestelmään on mahdollista, ja jopa hyvin yksinkertaista. Kiinteistön tietoverkon osalta tarvittiin siis lisäsuojauksia. Lisäsuojaus kiinteistön verkon sisällä toteutettiin palomuuria ja loogista aliverkkoa sekä Virtual Local Area Network, virtuaalinen aliverkko (VLAN) tekniikoita käyttämällä. Fyysisen verkon sisälle luotiin oma eristetty virtuaalinen aliverkko automaatiojärjestelmiä varten. Virtuaaliset aliverkot mahdollistavat fyysisen verkon jakamisen osiin, jolloin liikenne voidaan eristää ja rajoittaa tiettyihin segmentteihin. Tällä menetelmällä estetään ei-toivotun liikenteen leviäminen verkossa. Loogisessa aliverkossa on sallittu vain tietyt IP-osoitteet, jotka voivat kyseiseen verkon osaan liittyä. Loogisella aliverkolla varmistettiin myös se, että

BACnetin käyttämät broadcast-viestit pysyvät vain tässä yhdessä aliverkossa. Tästä aliverkosta ei sallittu mitään liikennettä muihin verkon osiin.

Näiden menetelmien keskeinen ero on se, että looginen aliverkko toimii OSI-mallin kolmannessa kerroksessa (verkkokerros) IP-osoitteiden perusteella, kun taas VLAN toimii toisessa kerroksessa (datalink-kerros) fyysisen verkon laitteiden, kuten kytkimien avulla. Looginen aliverkko perustuu IP-osoitteisiin, kun taas VLAN perustuu fyysisten laitteiden tai porttien sijaintiin verkossa. Yhdistettynä nämä voivat tarjota tehokkaan ja turvallisen tavan organisoida ja hallita verkkoa, koska VLANia voidaan käyttää fyysisen verkon segmentointiin ja loogisia aliverkkoja IP-osoitteiden jakamiseen näiden segmenttien sisällä.

Teknologia kehittyy jatkuvasti, joten myös protokollat kehittyvät ajan myötä vastaamaan muuttuvia vaatimuksia, kuten suurempaa datanopeutta tai parempaa tietoturvaa. Joissakin sovelluksissa käytetään silti yleisesti jo pitkään käytössä olleita ja muutoin hyväksi havaittuja protokollia, vaikka näiden tietoturvaominaisuudet eivät vastaakaan nykypäivän vaatimuksia. Näiden protokollien käyttäminen ei yleensä edellytäkään parempia tietoturvaominaisuuksia, koska niiden käyttö on paikallista, eikä niitä käyttäviä järjestelmiä liitetä tavallisesti suoraan yleisiin tietoverkkoihin.

Vaikka nämä vanhemmat protokollat eivät tarjoa nykypäivän tietoturvaominaisuuksia, niitä voidaan silti käyttää turvallisesti, kunhan niiden käyttöön liittyvät riskit tunnistetaan ja niihin varaudutaan asianmukaisesti. Jos tietoturvaltaan huonompaa protokollaa käytetään paikallisessa verkossa, joka on eristetty yleisestä tietoverkosta, sen tietoturvariskit ovat lähes olemattomat. Tällaisissa tapauksissa on kuitenkin tärkeää varmistaa, että paikallinen verkko on todella eristetty ja että siihen pääsy on rajoitettu vain luotettuihin laitteisiin ja käyttäjiin.

Lisäksi on olemassa useita tekniikoita ja menetelmiä, joilla voidaan parantaa vanhojen protokollien tietoturvaa. VPN-yhteyksillä voidaan suojata tietoliikennettä, tai voidaan käyttää palomureja rajoittamaan pääsyä järjestelmään. Myös järjestelmän säännöllinen päivittäminen ja tietoturvatarkastusten suorittaminen auttavat varmistamaan, että mahdolliset tietoturva-aukot havaitaan ja korjataan ajoissa.

On myös tärkeää muistaa, että tietoturva ei ole pelkästään tekninen, vaan myös organisatorinen ja inhimillinen kysymys. Henkilöstön kouluttaminen tietoturvakäytäntöihin ja -riskeihin on yhtä tärkeää kuin teknisten ratkaisujen käyttöönotto. Tietoturva on jatkuva prosessi, joka vaatii säännöllistä arviointia ja parannusta. On myös tärkeää havaita, että vaikka

tietoturva onkin tärkeä osa järjestelmän toimintaa, suunnittelua ja ylläpitoa, se ei saa estää järjestelmän tehokasta ja joustavaa käyttöä. Tietoturvaratkaisujen on oltava tasapainossa järjestelmän käytettävyyden sekä suorituskyvyn kanssa. Tämä vaatii jatkuvaa arviointia ja sopeutumista, sillä sekä teknologia että tietoturvat kehittyvät jatkuvasti.

3.6 Graafinen käyttöliittymä

Järjestelmään toteutettiin siis myös graafinen käyttöliittymä, josta nähdään järjestelmän reaaliaikainen tila käyttäjän toimiin liittyvien ohjauksien osalta, ja siitä voidaan myös ohjata useimpia järjestelmään liitettyjä laitteita. Graafista käyttöliittymää varten rajapintajärjestelmään tehtiin myös http-palvelin, jonka kautta käyttöliittymä jaettiin kiinteistön henkilökunnan käytettäväksi. Tämä jakaminen tehtiin henkilökunnan suojatun langattoman verkkoyhteyden kautta. Koska rajapintajärjestelmään ei ole kuitenkaan liitetty mitään turvallisuuteen vaikuttavaa tekijää, katsottiin nämä ratkaisut tietoturvan kannalta riittäväksi.



Kuva 11 Rajapintajärjestelmän graafisen käyttöliittymän etusivu

4 Johtopäätökset

4.1 Rajapintajärjestelmän toiminta

Beckhoff-logiikka rajapintajärjestelmänä täytti sille asetetut odotukset ja tavoitteet. Järjestelmän monipuoliset liitännämahdollisuudet erilaisiin väyliin, yhdistettynä vapaaseen toteutukseen ohjelmoimalla olivat tämänkaltaiseen rajapintana toimivaan järjestelmään sopivat. Erilaisten väyläprotokollien sisältämä ohjaustieto saatiin tuotua logiikan omiin muuttujiin sen kaltaisena, että sitä voitiin hyödyntää suoraan sekä logiikan sisällä, että muihin väyliin lähetettäessä. Rajapintajärjestelmästä saatiin myös tehtyä sellainen, että sen olemassaoloa ei normaalitilanteessa edes havaitse, joka oli yksi työn keskeisistä tavoitteista. Se myös keskusteli saumattomasti eri protokollien kanssa molempiin suuntiin, ja ohjaussignaalit järjestelmästä toiseen siirtyivät lähes viiveettä ja ilman virhetilanteita. Kun kunkin protokollan välittämä tieto on saatu luettua järjestelmän yhteensopiviin muuttujiin, ne ovat helposti käytettävissä kuhunkin järjestelmään kirjoittamisessa. Tiedonsiirto järjestelmästä toiseen ei ole kuitenkaan välttämättä reaaliaikaista kaikkien järjestelmien kesken, mutta nämä pienet, eri tekniikoista johtuvat viiveet, eivät tämänkaltaisessa ohjausratkaisussa vaikuta lainkaan.

Ohjelmoinnissa pyrittiin tekemään jokaiselle eri väyläprotokollan luenta- ja kirjoitustoiminnolle oma funktioblokkinsa, joka hoitaa vain yhtä tehtävää, jolloin näitä blokkeja voitiin hyödyntää monipuolisesti kaikkien protokollien kesken. Järjestelmistä luettu tieto taltioitiin logiikassa globaaleihin muuttujiin, jolloin ne olivat suoraan kaikkien toimintojen käytettävissä. Järjestelmästä ulos siirrettävien signaalien muuttujina käytettiin yhtä ja samaa muuttujaa kaikille järjestelmille, jolloin esimerkiksi DMX-järjestelmästä aiemmin tullut ohjauskomento voitiin tarvittaessa korvata KNX-järjestelmän ohjauskomennolla. Yhteisellä lähtömuuttujalla ristiriitaiset ohjauskomennot saatiin käsiteltyä, sillä tavalla, että viimeisin ohjauskomento on määräävä. Joissakin tietyissä toiminnoissa käytettiin myös ohjaustapaa, jossa jokin väyläprotokolla on määräävä. Esimerkiksi DMX-järjestelmän valopöydän ohjaus on tällainen. Valopöydästä tuleva ohjaus ohittaa kaikki muut järjestelmät siksi aikaa, kun valopöydän niin sanottu master-ohjauskomento on päällä. Koska tietojen siirto järjestelmästä toiseen perustuu yhteisessä käytössä oleviin muuttujiin, voidaan tällaiset rajoitukset toteuttaa helposti erilaisia ehtoja käyttämällä.

4.2 KNX-ohjauksen käytettävyys

KNX-väyläliityntäyksikön tekninen toteutus sekä siihen valmistajan puolelta tehdyt funktiot vaikuttivat toteutukseltaan hieman keskeneräisen oloisilta, ja KNX-väylän lukeminen sekä sinne kirjoittaminen aiheuttivat suurimpia haasteita kohteessa. Yhtä KNX-väylää jouduttiin laitteen ominaisuuksien vuoksi lukemaan usealla liityntäyksiköllä ja niiden keskinäisen toiminnan yhteensovittaminen aiheutti logiikan ohjelmakoodissa ajoitusongelmia. Väylälle kirjoittamisessa oli myös omat haasteensa, kun väylää luettiin jatkuvasti, niin samalla väyläliityntäyksiköllä kirjoittamisessa esiintyi sitä, että sanoma ei lähtenytäkään ensimmäisellä lähetykserralla liikkeelle. Tähän ongelmaan ei löydetty muuta ratkaisua, kuin varmistaa järjestelmän kannalta oleelliset ohjaukset siten, että mikäli KNX-järjestelmä ei kuittaa tiettyä ohjausta vastaanotetuksi, se lähetetään uudelleen seuraavassa PLC-sykliissä. Tähän saattoi myös vaikuttaa se, että sanomaa yritettiin lähettää, silloin kun väylä ei ollut vapaa, mutta tätä ei voitu diagnosoida Beckhoffin päästä, koska KL6301 laitteen reaaliaikaista tilaa ei ole mahdollista seurata mitenkään. Valmistajan toteutus laitteesta vaikuttaa siltä, että sillä on ennemminkin tarkoitus vastaanottaa ja lähettää yksittäisiä ohjauksia KNX-järjestelmään, kuin käyttää sitä yhtenä laajempuna ohjauspisteenä. Tähän vaikuttaa osaltaan se, että KNX-järjestelmässä voi liikkua hyvin monta erilaista datatyyppiä, eikä siitä voida lukea niin sanottua raakadataa tämänkaltaiseen järjestelmään, jossa erityyppiset tietomuodot täytyy saada tallennettua kukin omaan muuttujatyyppiinsä. Valmistajan ratkaisu lienee tämän vuoksi ollut se, että kukin datatyyppi tunnistetaan ja puretaan jo itse laitteessa Beckhoff-järjestelmään sopivaan muotoon. Myös KL6301 laitteen alustaminen ja kutsuminen jokaisessa PLC-sykliissä vaikutti ohjelmointiin. Beckhoff-järjestelmän KNX-järjestelmään liittyvät muuttujat eivät päivitty reaaliaikaisesti järjestelmän sisällä, vaan ne ovat riippuvaisia KL6301 laitteen tilasta, eikä niitä voi tästä syystä lukea tai kirjoittaa varmistumatta ensin siitä missä tilassa väyläsovitin kulloinkin on.

Tästä syystä tämänkaltaisessa rajapintana toimivassa järjestelmässä, jossa on tarkoitus pysyä lukemaan sekä kirjoittamaan moniin eri ryhmäosoitteisiin, kannattaisi mahdollisesti käyttää eri KL6301 laitetta KNX-sanomien lukemiseen ja kirjoittamiseen. Tämä havainto tuli tässä työssä esiin niin myöhäisessä vaiheessa, että sen toteuttamiseen ei ollut riittävästi aikaa kohteen valmistumisaikataulun puolesta. Tältä osin liityntää KNX-järjestelmään kannattaisi tutkia enemmän. Liityntäratkaisu, jossa tarvitaan sitä useampia komponentteja, mitä

useampaa tietoa jostain järjestelmästä tarvitaan, ei ole toteutuksen kannalta kovinkaan järkevä ratkaisu. KNX-järjestelmän osalta voitaisiin lisäksi tutkia myös mahdollisuutta kääntää KNX-protokolla johonkin toiseen protokollaan, esimerkiksi BACnetiin, jollain toisella sovitimmella ennen sen tuomista Beckhoff-järjestelmään.

4.3 DMX ohjauksen toiminta

DMX-järjestelmään liittyminen on Beckhoff-järjestelmässä tehty hyvin helpoksi. Kyseessä on yksinkertainen ja helposti käsiteltävä protokolla, joten sen toimintojen käsittely logiikassa on myös sen puolesta hyvin yksinkertaista. Beckhoff-järjestelmässä on omat laitteensa lähetykseen ja vastaanottoon, ja näistä saatavaa tietoa voidaan käsitellä suoraan lähes reaaliaikaisesti logiikassa. DMX-väylästä saatava tieto on hyödynnettävissä suoraan logiikan sisäisiin muuttujiin, eikä siinä tarvita erillisiä luku- tai kirjoitustoimintoja, kuten esimerkiksi KNX-protokollan käsittelyssä tarvitaan. DMX-slave -yksikköön tuleva tieto voidaan siirtää myös suoraan DMX-master -yksikköön, jos on tarvetta tällaiselle ohjaukselle. Kohteen tapauksessa osa valopöydästä tulevista DMX-ohjauskomennoista välitettiin suoraan valaisimille tätä menetelmää käyttäen. DMX-ohjauksessa Beckhoffiin saatava tieto tulee sovittimelta sisälle järjestelmään Receive Toggle tulon ohjaamana, eli aina kun ohjaava DMX-laite on liitettyä ja lähettävissä tilassa, tiedot saadaan reaaliaikaisena järjestelmän sisäisiin muuttujiin käytettäväksi. Toisaalta ohjauksen puuttuminen näkyy myös vain tässä Receive Toggle tulossa, joten ohjaavan signaalin puuttumista ei voi sen tarkemmin tulkita. Järjestelmässä päästään myös seuraamaan reaaliaikaisesti tulevaa signaalia EL6851 laitteen ominaisuuksien kautta. Laitteen tuloista nähdään suoraan DMX:n kanavakohtaiset arvot, mikä helpottaa esimerkiksi ohjelmointia ja mahdollista vianetsintää huomattavasti. Rajapintana toimivassa järjestelmässä tällaisesta diagnostiikkamahdollisuudesta on huomattavaa hyötyä.

4.4 BACnet-liityntä

Liityntä BACnet/IP-asiakkaaksi on Beckhoffissa tehty hyvin yksinkertaiseksi. Vaikka järjestelmien integrointi vaatiikin yleensä suunnittelua, konfigurointia ja tietoa järjestelmistä, kuten tässäkin tapauksessa, niin BACnet/IP liityntään tarvitaan käytännössä ainoastaan IP-osoite, johon liitytään, sekä toisen järjestelmän konfiguraatiosta tehty EDE-tiedosto.

Onnistunut konfiguraatio nähdään suoraan järjestelmän I/O-pisteiden reaaliaikaisesta tilasta. Tässä työssä käytetty MultiState Value ja Output -objektien käyttö on vain yksi esimerkki siitä, kuinka BACnet/IP:llä voidaan siirtää tietoa palvelimen ja asiakkaan välillä. Nämä objektit tarjoavat joustavan tavan välittää arvoja välillä 1–256, ja näiden arvojen ominaisuudet voidaan määritellä vapaasti ohjaavassa järjestelmässä. Tämä mahdollistaa yksinkertaisen tiedonsiirron, ja tämä ominaisuus tekee osaltaan BACnet/IP:stä erittäin joustavan protokollan tällaisessa käytössä. Vaikka BACnet/IP on tehokas ja joustava protokolla, sen käyttöönotto ja konfigurointi vaatii kuitenkin hieman asiantuntemusta ja kokemusta erilaisista järjestelmistä. BACnet/IP:n käyttöönottoon liittyvät haasteet, kuten oikean ohjelmakirjaston valinta ja konfigurointi, vaativat ymmärrystä protokollasta, sekä sen toiminnasta. BACnet/IP:n turvallinen käyttö edellyttää myös tietoturvan huomioimista kaikissa vaiheissa, mukaan lukien järjestelmän suunnittelu, toteutus ja ylläpito.

Lähteet

- ASHRAE. (2020). *ANSI/ASHRAE Standard 135-2020 BACnet A Data Communication Protocol for Building Automation and Control Networks*.
- Beckhoff. (4 2011). *Fast communication with the DMX protocol*. Noudettu osoitteesta <https://www.beckhoff.com/media/downloads/application-reports-downloads/2011/dk9222-0311-0029.pdf>
- Beckhoff. (10 2023). *EL6851DMX Master/Slave terminal*. Noudettu osoitteesta <https://download.beckhoff.com/download/Document/io/ethercat-terminals/el6851en.pdf>
- Beckhoff. (9 2023). *EtherCAT System Description*. Noudettu osoitteesta https://download.beckhoff.com/download/Document/io/ethercat-terminals/ethercatsystem_en.pdf
- Beckhoff Information System. (11 2023). *KL6301 EIB/KNX Bus Terminal*. Noudettu osoitteesta <https://download.beckhoff.com/download/document/io/bus-terminals/kl6301en.pdf>
- Beckhoff Information System. (11 2023). *TwinCAT 3 | PLC Library: Tc2_DMX*. Noudettu osoitteesta https://infosys.beckhoff.com/english.php?content=../content/1033/tcplclib_tc2_dmx/index.html&id=
- ESTA. (8 2004). *USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories*. Noudettu osoitteesta <https://tsp.esta.org/tsp/documents/docs/E1-11.pdf>
- ESTA. (1 2011). *E1.20 - 2010, Entertainment Technology-RDM-Remote Device Management over USITT DMX512 Networks*. Noudettu osoitteesta https://tsp.esta.org/tsp/documents/published_docs.php
- ETG. (9 2023). *EtherCAT Knowledge Base*. Noudettu osoitteesta https://www.ethercat.org/login.aspx?ReturnUrl=%2fmemberarea%2fen%2fknowledge_base.htm
- Fairhurst, G. (2020). *DMX Frame*. Noudettu osoitteesta <https://erg.abdn.ac.uk/users/gorry/eg3576/DMX-frame.html>
- KNX Association. (11 2023). *KNX Basic Course*. Noudettu osoitteesta http://knx.com.ua/attachments/article/132/KNX-basic_course_full.pdf
- Merz, H.; Hansemann, T.; & Hübner, C. (2009). *Building Automation: Communication systems with EIB/KNX, LON und BACnet*. Springer-Verlag Berlin Heidelberg.
- Professional Lightning and Sound Association. (10 2023). *DMX Start Codes*. Noudettu osoitteesta https://sitelec.org/download_page.php?filename=themes/dmx/start_code.pdf

Xueli Zhu, S. Z. (10 2015). *Research Gate*. Noudettu osoitteesta
https://www.researchgate.net/publication/314598165_Design_for_an_Interference-Suppressing_DM512_Protocol_Expansion_and_Repeater