LAPPEENRANNAN TEKNILLINEN YLIOPISTO
LUT School of Energy Systems
LUT Kone
BK10A0401 Kandidaatintyö ja seminaari

CODESYS HARJOITUSTYÖYMPÄRISTÖN KEHITTÄMINEN JA HARJOITUSTYÖN SUUNNITTELU

DEVELOPING A CODESYS EXERCISE ENVIRONMENT AND DESIGNIN AN ASSIGNMENT

Lappeenrannassa 10.5.2018

Tuomas Huuskonen
Ohjaaja & tarkastaja: TkT Lauri Luostarinen

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
LUT Energiajärjestelmät
LUT Kone

Tuomas Huuskonen

Codesys-harjoitustyöympäristön kehittäminen ja harjoitustyön suunnittelu

Kandidaatintyö

2018

37 sivua, 15 kuvaa ja 4 liitettä

Ohjaaja & tarkastaja: TkT Lauri Luostarinen

Hakusanat: PLC, ohjelmoitava logiikka, mekatroniikka, Codesys

Lappeenrannan teknillisen yliopiston Mekatroniikan kurssilla ei ole ollut laitteistoa mikroluokissa PLC-harjoitustöiden testaamiseen. Harjoitustöitä on testattu tietokonesimulaatioilla, jotka eivät anna realistista kuvaa siitä, kuinka harjoituksen asiat käytännössä toimivat. Testausta on myös tehty laboratoriossa, mutta suurien ryhmäkokojen vuoksi laboratoriotestausaika on jäänyt pieneksi opiskelijaa kohti. Ratkaisuksi kehiteltiin ohjelmointiasema, joka mahdollistaa PLC-harjoitustöiden testaamisen mikroluokissa, näin tehostaen oppimista.

Tässä kandidaatintyössä on suunniteltu ohjelmointiasema ja Mekatroniikan kurssin ensimmäinen uutta laitteistoa hyödyntävä harjoitustyö malliratkaisuineen. Työ jakautuu kahteen osaan, joista ensimmäinen kuvaa ohjelmointiaseman suunnittelua ja kokoonpanoa. Myös tietoturva on otettu huomioon, sillä asemat kytketään yliopiston sisäverkkoon. Toisessa osassa on selostettu harjoitustyön suunnittelu ja malliratkaisu käytännön kytkentöineen.

Ohjelmointiasemalla tavoiteltu vapaus harjoituksien testaamiseen paikasta riippumatta saavutettiin ja harjoitustyöstä tuli oppimista hyvin edistävä. Komponenttien rajallisen määrän takia, ohjelmointiasemasta ei tullut monipuolisuudeltaan parhain mahdollinen, mutta sen katsottiin olevan riittävän hyvä ensimmäiselle harjoitustyölle Mekatroniikan kurssilla. Ohjelmointiasema suunniteltiin modulaariseksi, jotta valmius lisäosille ja laajennuksille säilyi tulevia harjoituksia varten.

ABSTRACT

Lappeenranta University of Technology
LUT School of Energy Systems
LUT Mechanical Engineering

Tuomas Huuskonen

Developing a Codesys exercise environment and designing an assignment

Bachelor's thesis

2018

37 pages, 15 figures and 4 appendices

Supervisor & examiner: D. Sc. (Tech.) Lauri Luostarinen

Keywords: PLC, mechatronics, Codesys

The Mechatronics course in Lappeenranta University of Technology did not have the hardware for testing the practical assignments. The PLC-assignments have been tested with computer simulations that do not give a realistic picture of how things work in practice. Testing has also been conducted in the laboratory, but due to large group sizes testing time per student has been low. As a solution, a programming station was designed, which will enable the testing of the PLC-assignment codes in the computer classrooms.

This Bachelor's thesis describes the developing process of the programming station and designing of the first assignment using the new hardware for the Mechatronics course. This report is divided into two parts, first of which describes the development and assembling of the programming station. Cyber security was also addressed since the programming station will be connected to the local area network of the university. The second part describes the designing of the first assignment of the course. An example solution for the assignment was also created including the code and instructions for making the required connections.

The programming station allowed the freedom wanted for the testing of the assignments regardless of the location, therefore aiding the learning results. Due to the limited number of components in the programming station, it did not turn out most versatile, but was considered good enough for the first assignment. Station was designed to be modular, to keep the possibility of future addons and extension parts.

SISÄLLYSLUETTELO

LIITE III: Kytkentäkaavio

LIITE IV: Ohjelmointiaseman käyttöohje

LYHENNELUETTELO

CAD	Computer Aided Design – tietokone avusteinen suunnittelu
GPIO	General Purpose IO – ohjelmoitava portti signaalin vastaanottamiseen tai lähettämiseen
FBD	Function Block Diagram – lohkokaavio-ohjelmointitapa
IL	Instruction List – ohjelistaohjelmointitapa
LD	Ladder Diagram - tikapuukuvaajaohjelmointitapa
PLC	Programmable logic controller – ohjelmoitava logiikka
SFC	Sequential Function Charts – juokseva funktiokaavio -ohjelmointitapa
SSH	Secure Shell – salattu tietoliikenne protokolla
ST	Structured Text – rakenteellinen teksti -ohjelmointitapa

1 JOHDANTO

Lappeenrannan teknillisen yliopiston Mekatroniikan kurssilla on viime vuosina tehty harjoitustyöt pääosin virtuaalisen PLC:n ohjelmoinnilla. Tämä on melko toimiva ratkaisu, mutta mitään konkreettista kosketuspintaa harjoitukseen ei tule. Kaikki tapahtuu ruudulla tekstinä, tai esimerkiksi erilaisina väreinä. Konkreettisen työaseman ja oikeiden kytkentöjen tekemisen lisäksi, ohjelmointiasema mahdollistaisi todella monipuoliset mahdollisuudet harjoituksien sisällölle. Esimerkiksi sähkömoottori, pumppu tai jokin muu sähköinen toimilaite voitaisiin kytkeä ohjelmointiasemaan, mutta tietokoneen näytöllä sen simuloiminen olisi vaikeaa ja keskittyminen päätyisi kauas harjoituksen tavoitteista.

Harjoitustöitä on testattu harjoitusryhmä kerrallaan laboratoriossa. Tämä on kuitenkin tehoton ja aikaa vievä tapa, sillä harjoitusoppituntien lisäksi on vielä pidettävä jokaisen ryhmän kanssa erillinen harjoitus laboratoriossa. Laboratorioharjoitukset ovat hyviä ja käytännön läheisiä, mutta ajankäyttöongelmat ja yhden laboratorioharjoituksen riittämättömyys opetuksen tehokkaaseen välittymiseen luovat tarpeen jollekin uudelle ratkaisulle. Parempi olisi, jos jokaisella harjoitusoppitunnilla voitaisiin harjoitella kokonaisvaltaisesti meneillään olevan harjoituksen asioita.

1.1 Tutkimusongelma

Tämä kandidaatintyö käsittelee kymmenen PLC-ohjelmoinnin harjoituslaitteiston suunnittelua, kokoonpanoa ja testausta Lappeenrannan teknillisen yliopiston Mekatroniikan kurssia varten. Työhön sisältyy myös kurssin ensimmäisen harjoitustyön suunnittelu ja sen malliratkaisun rakentaminen. Tutkimusongelma on seuraava:

- Mekatroniikan peruskurssilla ei ole kompaktia ja liikuteltavaa PLC-laitteistoa.

1.2 Tavoitteet ja tutkimuskysymykset

Ongelman ratkaisuksi ideoitiin kannettava PLC-ohjelmointiasema, jonka harjoituksenpitäjä voisi viedä oppitunnille mukanaan, mahdollistaen näin käytännön testaamisen jo oppituntien aikana. Tavoitteena tässä kandidaatintyössä on kehittää mahdollisimman kompakti, monipuolinen ja tarpeeksi helppokäyttöinen ohjelmointiasema. Ensimmäisestä

harjoituksesta on tavoitteena saada mahdollisimman mielekäs, mutta kuitenkin haastava ja informatiivinen. Tutkimuskysymykset ovat seuraavat:

- Kuinka toteuttaa kompakti ja helposti liikuteltava ohjelmointiasema?
- Kuinka tehdä kyseisestä yksiköstä tarpeeksi tietoturvallinen liitettäväksi yliopiston verkkoon?

1.3 Laitteisto

Seuraavat käsitteet ovat tärkeitä tätä työtä lukiessa:

- Codesys on 3S-Smart Software Solutionsin kehittämä IEC 61131-3 -normin mukainen ohjelmointiympäristö. Codesys kattaa teollisuusautomaation eri lajit yhdessä ohjelmistossa.
- Codesys Runtime on ohjelmisto, joka on saatavilla lähes jokaiselle älykkäälle laitteelle. Runtime kykenee ajamaan Codesys koodia ja ohjaamaan toimilaitteita koodin käskyjen mukaan.
- EtherCAT on Beckhoffin kehittämä automaatiossa käytettävä tehokas väylätekniikka. EtherCAT-ohjain kykenee ohjaamaan useita EtherCAT-laitteita.

Ohjelmointiympäristön ensimmäinen versio tulee sisältämään Raspberry Pi yhden piirikortin tietokoneen, Beckhoffin EtherCAT-portin, sekä digitaalisia ja analogisia sisään- ja ulostuloja. Koteloon tulee myös 24-voltin virtalähde EtherCATille ja I/O-moduleille. Myös Raspberry tarvitsee oman virtalähteensä, joten koteloon on myös asennettava toinen 5-voltin virtalähde. Raspberryyn asennetaan Codesys Runtime, joka pystyy ajamaan Codesysillä ohjelmoitua koodia. Codesysiä on tässä kandidaatintyössä ohjelmoitu ladder-logicilla sen yksinkertaisuuden ja selkeyden takia. Opiskelijoille jää kuitenkin vapaus käyttää parhaaksi näkemäänsä ohjelmointitapaa.

Koodien testaaminen ohjelmointiasemalla vähentää huomattavasti harjoituksiin kuluvaa kokonaisaikaa, joten harjoituksiin voitaisiin keskittyä syvällisemmin ja tehokkaammin. Näin jokaisessa harjoituksessa mahdollistettaisiin maksimaalinen oppiminen. Lisäksi opiskelijat saisivat arvokasta kokemusta Beckhoffin I/O-moduuleista, jotka toimivat vastaavalla tavalla kuin kaikki teollisuudessa käytettävät I/O-moduulit. Näin opiskelijat valmistautuisivat jo opiskeluaikanaan tulevan työelämän laitteisiin ja laitteistoihin.

1.4 Rajaukset ja käytetyt ohjelmistot

Suurimmat rajaukset olivat komponenttien vähäinen määrä ohjelmointiasemassa ja sovittu kymmenen yksikön kokoonpaneminen. Harjoitustyön suunnittelu rajattiin siten, että tehtävänannon lisäksi oli suunniteltava myös malliratkaisu. Myös selkeät ja lyhyehköt ohjeet ohjelmointiaseman käyttöönottoa ja käyttämistä varten tuli kehittää. Itse laitteiston piti olla Codesys-yhteensopiva ja soveltua PLC-ohjelmoinnin ja komponenttien kytkennän harjoitteluun. Lisäksi laitteiston piti olla liitettävissä yliopiston verkkoon. Tässä kandidaatintyössä käytetyt ohjelmistot ovat esitelty alla.

1.4.1 Codesys V3.5 SP12

3S-Smart Software Solutionsin kehittämä IEC 61131-3 normin mukainen ohjelmointiympäristö. Codesys on tarkoitettu vaativaan automaatiosuunnitteluun ja ohjelmointiin teollisuuden tarpeisiin. (CODESYS Runtime 2018a.)

IEC 61131-3 on standardi ohjelmoitaville logiikoille ja sen kolmas osa, joka on julkaistu 2013, määrittää eri ohjelmointikielien käyttöä ja ominaisuuksia. Graafisia ohjelmointikieliä on kolme: Ladder Diagram, Sequential Function Charts ja Function Block Diagram. Tekstikieliä on kaksi: Instruction List ja Structured Text. (IEC 61131-3 2013, s.14.)

1.4.2 Solidworks 2017 Student Edition

Solidworks on Dassault Systémesin kehittämä laaja 3D-mallinnusohjelmisto, jolla mallinnettiin ohjelmointiaseman runkoon tehtäviä koneistuksia. Ohjelmiston laajennusosilla voi tehdä laajan skaalan eri asioita. Esimerkiksi virtausanalyysiä ja rasitusanalyysiä. (Solidworks 2018.)

Solidworks on Lappeenrannan teknillisessä yliopistossa käytössä oleva CAD-ohjelmisto, joten sitä on käytetty tämän kandidaatintyön tekoprosessissa. Solidworks on laajalti käytössä myös teollisuudessa sen helppokäyttöisyytensä ja yksinkertaisuutensa takia. Tästä huolimatta Solidworksillä voi suunnitella mitä monimutkaisempia rakenteita ja kokoonpanoja.

2 KIRJALLISUUSKATSAUS

Mekatroninen järjestelmä koostuu määritelmänsä mukaan mekaanisesta osasta, joka tekee määrättyä liikettä, sekä elektronisesta osasta, joka tekee järjestelmästä älykkään (Bishop 2008, osa 2 s. 3). Nykypäivänä lähes kaikki laitteet sisältävät mekatroniikkaa. Ajoneuvon moottori, vaihteisto ja esimerkiksi tuulilasinpyyhkijä muodostavat ohjausjärjestelmänsä kanssa mekatronisen kokonaisuuden Näin ollen mekatroniikkaa on todella tärkeää opettaa konetekniikan opiskelijoille.

2.1 PLC – Ohjelmoitava logiikka

PLC, eli ohjelmoitava logiikka on 1970-luvulta lähtien vallannut alaa relepohjaisten ohjausratkaisujen seuraajana. Se on mikroprosessoripohjainen ohjainlaite, jonka ohjelmointitapa muokattiin releiden ladder logic -kuvaajien mukaisiksi, jotta suunnittelijoiden ja rakentajien olisi helpompi siirtyä uuden ohjaustavan pariin. (Bishop 2008, osa 25 s. 3.)

PLC-laitteita kutsutaan vapaasti ohjelmoitaviksi logiikoiksi, koska ohjelman kirjoitusjärjestys on vapaa, eikä vaikuta ohjelman toimintajärjestykseen. Ohjelmaa siis skannataan jatkuvasti rivi kerrallaan. Ohjelman käynnistyessä, ensin luetaan tulojen ja lähtöjen tila ja tallennetaan ne I/O-muistiin. Tämän jälkeen ajetaan ohjelmamuistissa olevat ohjelmarivit. PÄÄLLE- ja POIS-komennot toteutetaan vasta kun ohjelmakierros on luettu END-käskylle asti. (Keinänen et al. 2001, s244.)

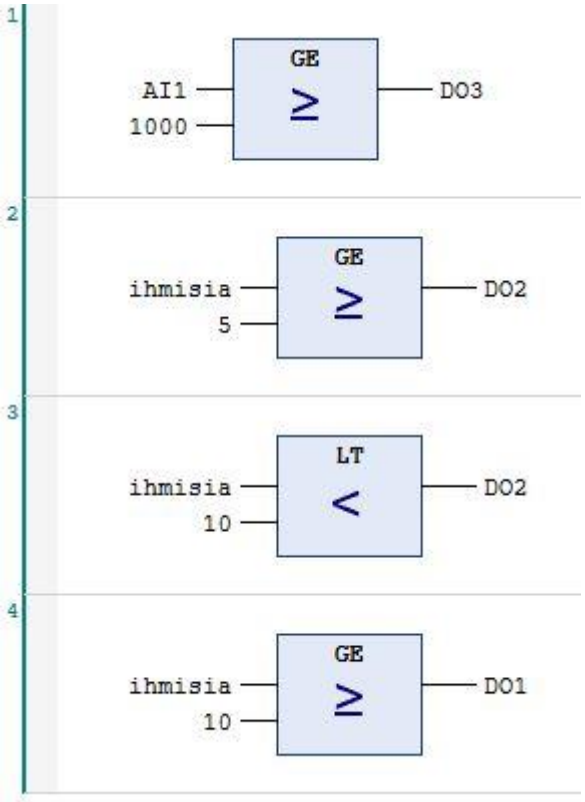
IEC 61131-3 on standardi, joka on kehitetty yhdistämään ohjelmoitavaa logiikkaa. Standardi määrittää viisi erilaista ohjelmointitapaa: Ladder Diagram(LD), Instruction List(IL), Structured Text(ST), Sequential Function Charts(SFC) ja Function Block Diagrams(FBD). Ohjelmointitavat on suunniteltu toimimaan keskenään, mahdollistaen näin ohjausjärjestelmän suunnittelemisen ja toteuttamisen kahta tai jopa useampaa tapaa käyttäen. (Bishop 2008, osa 25 s. 3.)

2.2 Ohjelmoiminen

Tekstipohjaisia kieliä on siis kaksi. Nämä sopivat ohjelmoimaan tottuneille ihmisille, sillä erityisesti ST-ohjelmointitapa muistuttaa todella paljon C-ohjelmointikieltä. IL, eli Instruction List -ohjelmointitapa on laajasti käytetty niin sanottuna välikielenä. Muita tekstipohjaisia ja graafisia ohjelmointitapoja käännetään IL-kielelle. (John & Tiegelkamp 2010, s100.) Kuten ohjelmointitavan nimestä voi päätellä, on koodi lista ohjeita, joita PLC suorittaa. ST, eli Structured Text taas tarjoaa laajan skaalan erilaisia monimutkaisia tehtävälauseita, jotka on pakattu tiiviiksi käskyiksi (John et al. 2010, s116).

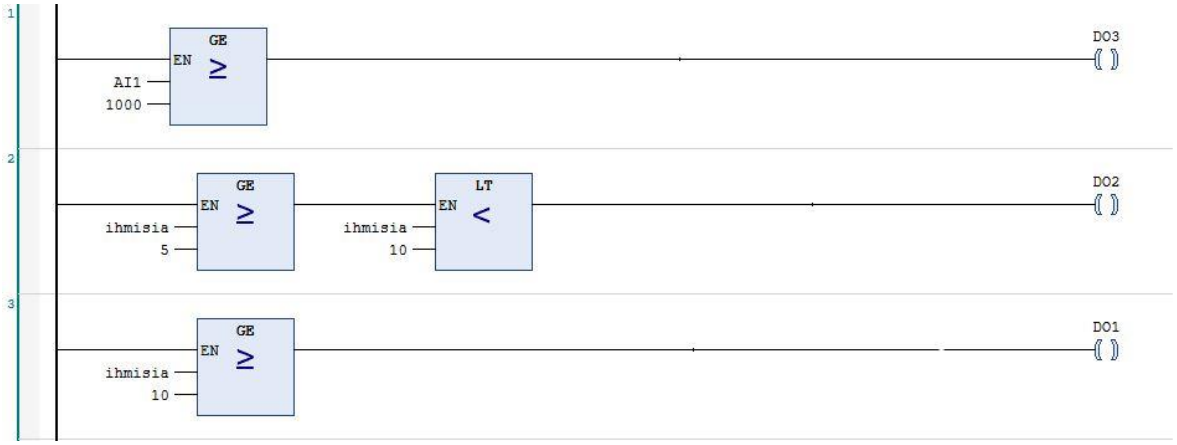
Graafisia ohjelmointitapoja on kolme. Nämä on helpompia ymmärtää, mikäli ei ole aiempaa kokemusta ohjelmoimisesta. Graafiset ohjelmointitavat muistuttavat kytkentäkaavioita, joihin vedetään hiirellä erilaisia operaattoreita, jotka sitten suorittavat niille määritettyjä tehtäviä. FBD ja LD -tavoilla ohjelmoitaessa, on käytössä network-rakenne. Jokaisessa networkissä on erillinen osa koodista ja ne ajetaan joko ylhäältä alas, tai ohjelmoijan määrittämässä järjestyksessä. Networkin sisältö jakautuu kolmeen eri kategoriaan: graafiseen objektiin, graafisiin elementteihin ja yhteysviivoihin. (John et al. 2010, s135.) Objekti on itse funktio, joka tekee muun muassa laskutoimituksia tai laskee esimerkiksi anturitietoa. Elementit ovat tämän funktion osia, esimerkiksi muuttujia, antureiden tietoja tai muita funktioon tarvittavia sisääntuloja. Yhteysviivat taas toimivat kytkentäkaavion tavoin yhdistäen eri funktioita ja reitittäen logiikkaa ja käskyjä eteenpäin.

FBD-ohjelmointitapa on alun perin signaalien käsittely alalta, jossa integer- ja floating point-arvot ovat tärkeitä. Siitä on kuitenkin vakiintunut ajan saatossa tärkeä universaali ohjelmointitapa teollisuusohjainten ohjelmoimiseen. (John et al. 2010, s135.) FBD-ohjelmoiminen muistuttaa nykyään hyvin paljon LD-ohjelmoimista, sillä IEC 61131-3 standardin myötä ohjelmointitavat ovat integroituneet toisiinsa melko paljon. Kuvassa 1 on rakennettu yksinkertainen koodi FBD-menetelmällä.



Kuva 1. Esimerkki FBD-ohjelmointitavalla rakennetusta koodista.

Kuten kappaleessa 2.1 sanottiin, perustuu Ladder Diagram -ohjelmointitapa ohjelmoitavaa logiikkaa edeltävään relepohjaiseen ohjaukseen. Tämän takia se soveltuu pääasiassa PÄÄLLÄ- ja POIS-käskyjen suorittamiseen (John et al. 2010, s147). Kuitenkin koska IEC 61131-3 -standardi määrittää ohjelmointitavat keskenään yhteensopiviksi ja ne on sellaisiksi suunniteltu, voidaan LD-ohjelmaan lisätä esimerkiksi ST- ja FBD-ohjelmien osia. Tällöin koodista voidaan tehdä huomattavasti monimutkaisempi ja siihen voidaan lisätä erilaisia toiminnallisuuksia. Kuvassa 2 on ensimmäisen kuvan koodin toiminnot toteutettu LD-ohjelmointitavalla.



Kuva 2. Sama esimerkki koodi kuin kuvassa 1, mutta ohjelmoituna LD-ohjelmointitavalla.

IEC 61131-3 -standardissa on määritetty, että SFC-ohjelmointitavalla saadaan suuria ja monimutkaisia ohjelmia osioitua pienemmiksi helpommin hallittaviksi kokonaisuuksiksi. Tällä tavalla on myös mahdollista suunnitella päällekkäisiä ja peräkkäisiä prosesseja. Prosessit joissa eteneminen on askeltavaa, tai prosesseissa on selkeitä vaiheita, ovat hyvin soveltuvia SFC-ohjelmoinnilla toteutetulle ohjaimelle. (John et al. 2010, s147.)

2.3 Tietoturvaselvitys

Raspberry Pin tietoturva on internetissä puhuttava aihe. Avoimenlähdekoodin käyttöjärjestelmä, oletuskäyttäjänimet ja -salasanat, sekä yksinkertainen käyttöjärjestelmäarkkitehtuuri tekevät laitteesta helposti murrettavan. Sopivia toimenpiteitä noudattaen nämä tietoturva-aukot saadaan kuitenkin korjattua ja Raspberry Pistä saadaan turvallinen laite.

Oletuksena Raspberryn käyttäjänimi ja salasana ovat aina samat: Pi ja raspberry. Näiden vaihtaminen estää hakkeria, joka laitteeseen yrittää murtautua, kirjautumasta sisään oletustiedoilla. Näin voidaan varmistaa, ettei kukaan jolla ei ole tarvittavia tietoja sisäänkirjautumista varten, pääse kirjautumaan ja mahdollisesti väärinkäyttämään Raspberry Pi -tietokonetta. Palomuurin asetukset on myös syytä käydä läpi ja varmistaa, ettei luvatonta käyttöä pääse tapahtumaan niiden puolesta. Laiteen käyttöjärjestelmä on syytä pitää ajan tasalla lataamalla aina viimeisimmät päivitykset, jotta löydetyt tietoturva-aukot ja muut haavoittuvuudet saadaan paikattua. Verkkoprotokollat, joita ei käytetä, on järkevää pitää pois päältä, jotta niiden kautta ei tapahdu väärinkäyttöä. (Klein Keane & Van De Viele 2017.)

Edellä mainittua väärinkäyttöä voisi olla esimerkiksi kryptovaluutan louhiminen, palvelunestohyökkäykset tai roskapostin lähettäminen. Mikäli väärinkäyttöä pääsisi tapahtumaan, voisi LUT:n sähköpostipalvelin joutua mustalle listalle yhteistyöyritysten sähköpostijärjestelmissä ja näin edes asialliset sähköpostit eivät reitittyisi vastaanottajille. Myös juurikäyttäjä, eli eräänlainen pääkäyttäjä, tulee turvata vaihtamalla salasana. Tietoturvasyistä näitä salasanoja ja käyttäjä nimiä ei tässä työssä mainita, työn ollessa julkinen.

3 OHJELMOINTIASEMAN SUUNNITTELU JA KOKOONPANO

Ohjelmointiaseman fyysinen kokoonpano oli huomattavasti helpompaa kuin ohjelmallisen puolen toimintakuntoon saaminen. Ohjelmallisella puolella ongelmia aiheutti dokumentaationpuute siitä, kuinka Beckhoffin I/O-moduuleita käytetään Codesysin kanssa. Tässä luvussa on esitelty kuinka sekä fyysinen että ohjelmallinen puoli saatiin lopulta onnistuneesti toimimaan.

3.1 Raspberry Pi 3 Model B

Käytössä oleva Raspberry Pi 3 Model B on ensimmäinen kolmannen sukupolven Raspberry tietokone. Se korvasi Raspberry Pi 2 Model B:n alkuvuonna 2016. (Raspberry Pi 2018a.)

Tekniset tiedot ovat melko maltilliset, mutta laitteen kokoon nähden todella vakuuttavat (Raspberry Pi 2018a):

- 1,2 GHz neliydinprosessori
- 1 Gt RAM-muistia
- Wlan ja Bluetooth
- 40-nastainen GPIO
- 4 kappaletta USB-2 –portteja
- Täysikokoinen HDMI-liitäntä
- CSI-kameraliitäntä Raspberry Pi kameralle
- DSI-kuvaliitäntä Raspberry Pi kosketusnäytölle
- Micro SD -portti käyttöjärjestelmää ja tiedostojen säilömistä varten
- Micro USB -virtalähde 2,5 A:in asti

Ostetun pakkauksen mukana tuli myös 16 Gt muistikortti, jolle oli asennettu NOOBS käyttöjärjestelmämanageri. Lyhenne NOOBS tulee sanoista New Out Of Box Software (Raspberry Pi 2018b). Käyttöjärjestelmän asennus on näin tehty todella helpoksi. Valitaan käyttöjärjestelmä, jota halutaan käyttää ja asennusohjelma suorittaa asennuksen muutamassa minuutissa. NOOBS sisältää fyysisesti muistikortilla vain Raspbian-käyttöjärjestelmän, mutta verkkoyhteydellä vaihtoehtojen määrä kasvaa yhdeksällä eri käyttöjärjestelmällä (Raspberry Pi 2018b).

Ohjelmointiasemaan valittiin Raspbian, koska Codesys Runtime on suunniteltu toimimaan sillä (CODESYS 2018b). Asennus oli helppo ja nopea toimenpide, jonka jälkeen Raspberry Pi käynnistyikin jo käyttöjärjestelmän oletuskäyttäjän työpöydälle.

3.2 Tietoturvan takaaminen

Raspberry Pi tietokoneesta tuli tehdä tietoturvallinen Lappeenrannan teknillisen yliopiston verkkoon liittämistä varten. LUT:n tietohallinto antoi ohjeistuksen tähän ja käski seurata Justin Klein Keanen ja Tom Van De Vielen kirjoittamaan artikkelia. Ensimmäinen askel on vaihtaa oletuskäyttäjänimi ja -salasana, jotta hakkerit eivät pääse väärinkäyttämään laitetta.

Klein Keanen ja Van De Vielen mukaan alla olevilla käskyillä tehdään uusi käyttäjä ja luodaan sille uusi salasana Rasbianin komentorivillä (Klein Keane et al. 2017):

- sudo /usr/sbin/useradd --groups sudo -m KÄYTTÄJÄNIMI
- sudo passwd KÄYTTÄJÄNIMI

Seuraavilla käskyillä poistetaan oletuskäyttäjä pi ja vaihdetaan juurikäyttäjän root salasana (Klein Keane et al. 2017):

- sudo userdel pi
- sudo passwd root

Seuraava askel on laitteen palomuurin asetuksien kiristäminen niin, että vain oikeutetut henkilöt pääsevät yhdistämään Raspberry Pihin. Iptables on Raspbianin palomuurisovelluksen nimi. Käyttäen komentoriviä ja alla olevaa käskyä, varmistetaan että Iptables on asennettu:

- sudo apt-get install iptables iptables-persistent

Iptables tallentaa asetukset rules.v4 -nimiseen tiedostoon. Tämä tiedosto tulee luoda ennen kuin sinne voi kirjoittaa asetuksia, sillä oletuksena se on tyhjä. Seuraavalla käskyillä oletusasetustiedosto luodaan (Klein Keane et al. 2017):

- sudo /sbin/iptables -L
- sudo /sbin/iptables-save > /etc/iptables/rules.v4

Tiedostoon tulee lisätä kuvassa 3 esitetyt asetukset. Rivit 7 ja 8 sallivat silmukka lo0 liikenteen portilla 127/8 ja hylkäävät kaiken liikenteen, joka ei käytä lo0 silmukkaa. Rivi 11 sallii todennetun saapuvan liikenteen. Rivillä 19 sallitaan salattu SSH-liikenne portilla 22.

Tämä on välttämätöntä sallia, sillä Codesys kommunikoi käyttäen SSH-protokollaa. Rivillä 22 asetetaan estettyjen yhteyksien tallentaminen päälle, jotta estetyistä yhteyksistä jää merkintä ja voidaan tarvittaessa huomata väärinkäyttöyritykset. Rivit 25 ja 26 hylkäävät kaiken liikenteen, jota ei ole erikseen sallittu.

```
1  # Generated by iptables-save v1.4.21 on Tue Feb 27 12:12:21 2018
2  *filter
3  :INPUT ACCEPT [0:0]
4  :FORWARD ACCEPT [0:0]
5  :OUTPUT ACCEPT [0:0]
6  # Allows all loopback (lo0) traffic and drop all traffic to 127/8 that doesn't use lo0
7  -A INPUT -i lo -j ACCEPT
8  -A INPUT ! -i lo -d 127.0.0.0/8 -j REJECT
9
10 # Accepts all established inbound connections
11 -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
12
13 # Allows all outbound traffic
14 # You could modify this to only allow certain traffic
15 -A OUTPUT -j ACCEPT
16
17 # Allows SSH connections
18 # The --dport number is the same as in /etc/ssh/sshd_config
19 -A INPUT -p tcp -m state --state NEW --dport 22 -j ACCEPT
20
21 # log iptables denied calls (access via 'dmesg' command)
22 -A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7
23
24 # Reject all other inbound - default deny unless explicitly allowed policy:
25 -A INPUT -j REJECT
26 -A FORWARD -j REJECT
27
28 COMMIT
29 # Completed on Tue Feb 27 12:12:21 2018
30
```

Kuva 3. Palomuuriin syötetyt asetukset. Perustuu soveltaen Klein Keanen kirjoittamaan ohjeeseen.

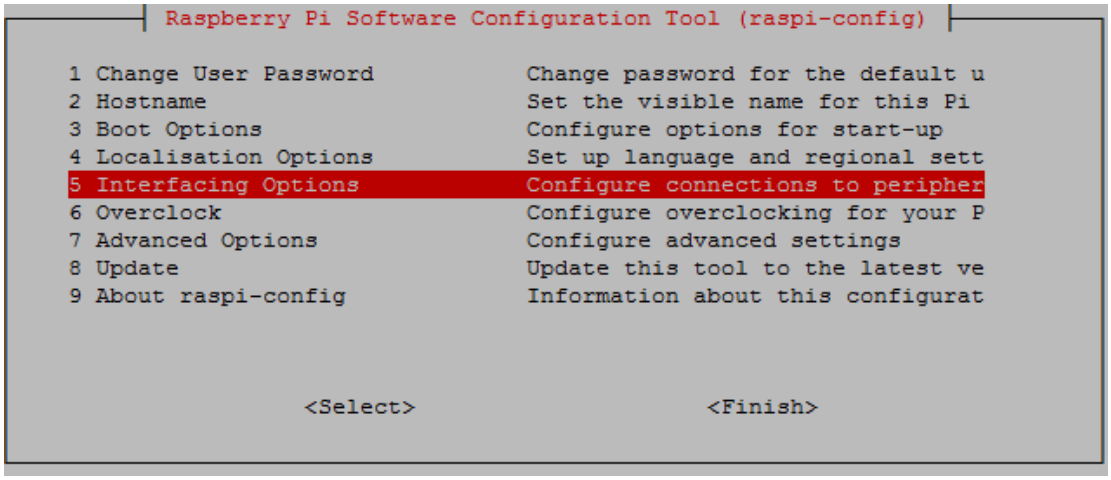
Lopulta on vielä pantava asetukset täytäntöön seuraavalla käskyllä (Klein Keane et al. 2017):

- sudo /usr/sbin/iptables-apply /etc/iptables/rules.v4

Tietoturvaan vaikuttaa myös esimerkiksi käyttöjärjestelmän päivitysten lataaminen ja siten käyttöjärjestelmän pitäminen ajan tasalla. Rasbianissa on Windows-tyylinen päivittäjä, mutta se on oletuksena pois päältä. Koska ohjelmointiaseman Raspberry Pi:t eivät tule olemaan yhteydessä internettiin, tehdään päivitykset manuaalisesti. Päivittäminen onnistuu komentorivin kautta, käyttäen käskyä:

- sudo raspi-config

Käsky avaa kuvassa 4 esitetyn valikon. Valikossa valitaan kohta 8. ja päivitetään käyttöjärjestelmä.



Kuva 4. Sudo raspi-config käskyllä aukeava valikko, josta saa päivitettyä käyttöjärjestelmän.

3.3 Codesys Runtimen asennus

Kuten johdannossa kirjoitettiin:

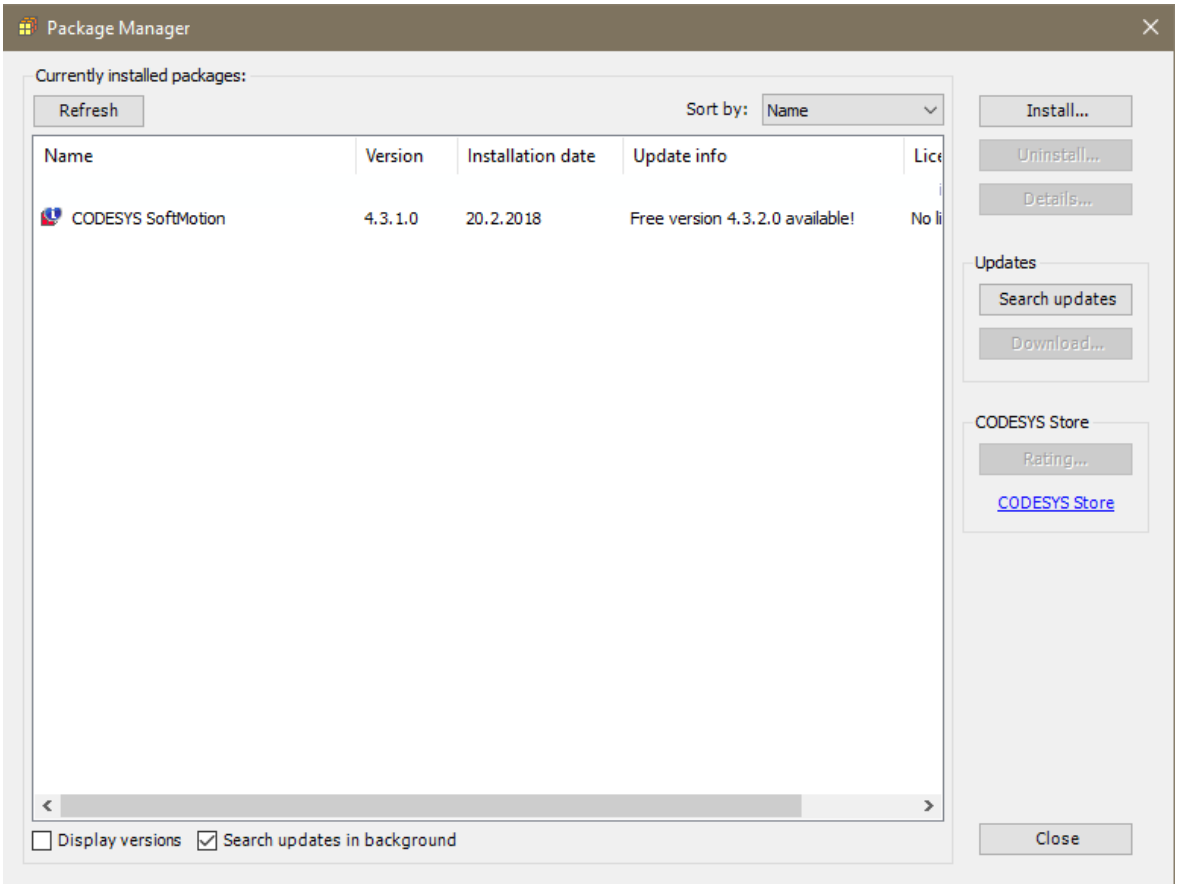
- Codesys Runtime on ohjelmisto, joka on saatavilla lähes jokaiselle älykkäälle laitteelle. Runtime kykenee ajamaan Codesys-koodia ja ohjaamaan toimilaitteita koodin käskyjen mukaan.

Codesys Runtimen voi ladata Codesysin verkkokaupasta, mutta ilman lisenssiä Runtime toimii yhtäjaksoisesti vain kaksi tuntia. Tällä estetään ilmaisen version käyttö teollisuudessa ja mahdollistetaan ilmaisversion opiskelija- ja harjoittelukäyttö.

Kun Codesys Runtime -paketti on ladattu, on aika avata Codesys V3.5. Ensimmäisenä on asennettava ladattu paketti. Tämä onnistuu seuraamalla alla olevaa valikkorakennetta:

- Tools > Package Manager

Kuvassa 5 on Paketti Manageri, jolla asennetaan Raspberry Runtime -paketti Codesysiin. Painamalla Install ja navigoimalla ladatun paketin kansioon saadaan Runtime asennettua. Tämän jälkeen Runtime on vielä asennettava itse Raspberry Pi:lle.



Kuva 5. Paketti manageri jolla asennetaan Runtime Codesysiin.

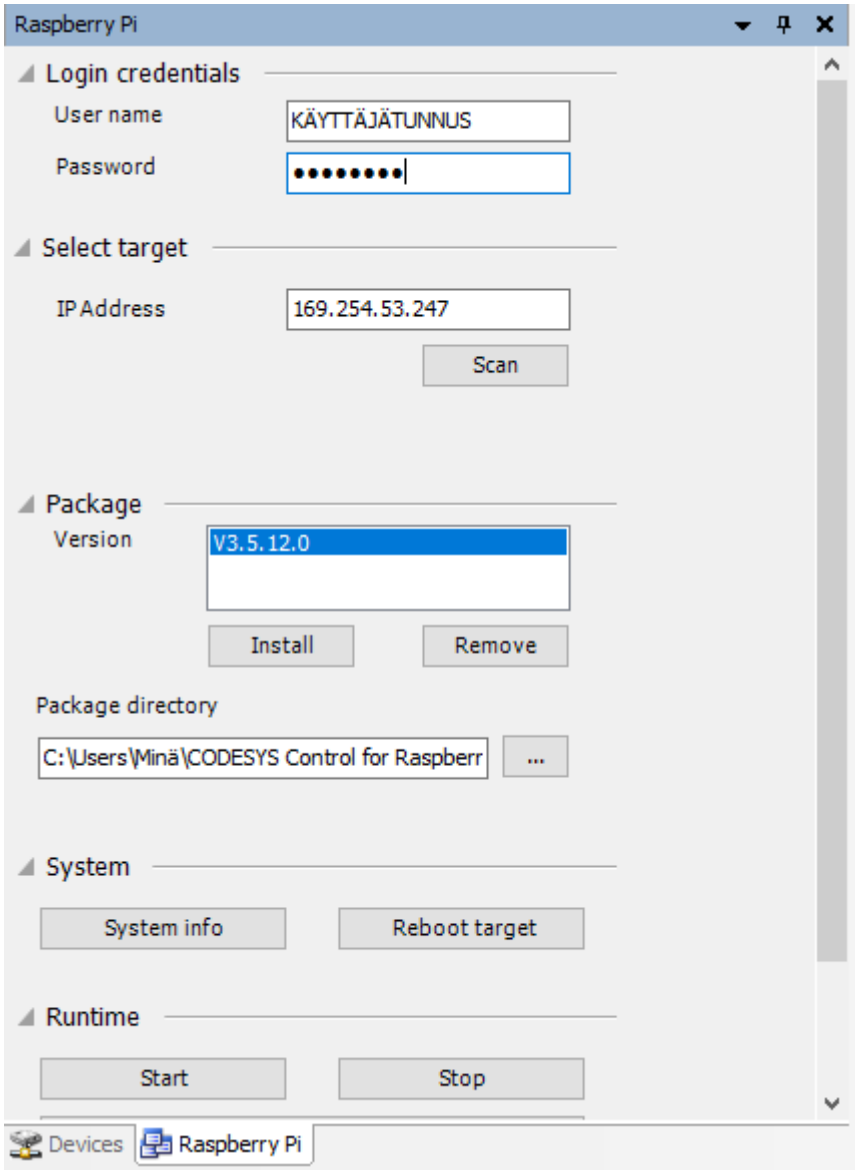
Ennen asennusta on SSH-protokolla kytkettävä päälle Raspberrystä jotta Codesys saa yhteyden Raspberryyn. Tämä onnistuu komentorivin kautta käyttäen käskyä:

- sudo raspi-config

Kuvassa 4 näkyvästä valikosta valitaan kohta 5. Interfacing Options. Tämän jälkeen valitaan kohta SSH ja kytketään se päälle. Raspberry Pi:n IP-osoite on vielä selvitettävä ennen kuin Runtimen asennus voidaan suorittaa. Tämä onnistuu sekä komentoriviltä, että graafisesta käyttöjärjestelmästä. Tällä kertaa käytettiin graafista käyttöliittymää sen helppouden takia. Oikeassa yläkulmassa on Wlan-logo, jonka päällä pitämällä hiirtä näkee eri verkkoyhteyksien osoitteet. Raspberryn Ethernet-portin on oltava kytkettynä tietokoneen Ethernet-porttiin, jotta eth0 verkon IP-osoite näkyy.

Seuraavaksi avataan Raspberry Pi työkalupalkki Codesysin kohdasta:

- Tools > Update Raspberry Pi



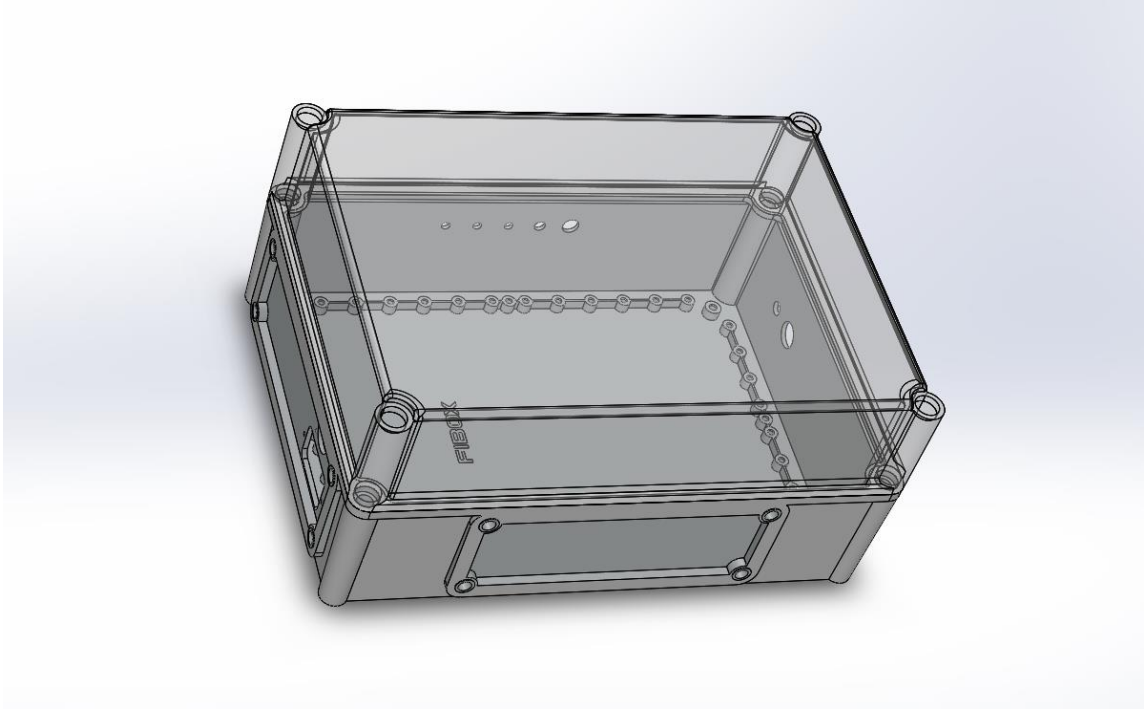
Kuva 6. Raspberry Pi työkalupalkki, johon annetaan Raspbianin käyttäjätunnus, salasana ja IP-osoite.

Käyttäjätunnuksen, salasanan ja IP-osoitteen antamisen jälkeen, alkaa asennus painamalla Install-painiketta. Mikäli virheitä ei tule, antaa Codesys alapalkkiin viestin onnistuneesta asennuksesta. Runtimen toimivuus on vielä testattava, painamalla kuvassa 6. näkyvää Start-painiketta Runtime-otsikon alapuolella. Painikkeen painamisen jälkeen alapalkkiin pitäisi tulla viesti onnistuneesta käynnistyksestä ja Runtimen toimivuudesta.

3.4 Ohjelmointiaseman kokoonpano

Ohjelmointiasemaan tulleet komponentit olivat melko isoja, joten tilanpuute ja ohjelmointiaseman lämpötila mietityttivät kokoonpanovaiheessa. Koko aseman runkona

toimii Fiboxin EKPE asennuskotelo 380x280x100 mm koossa. Koteloihin asennetaan sormiruuvein irrotettava läpinäkyvä kansi, asennuksen siistimmän ulkomuodon, sekä komponenttien suojaamisen varmistamiseksi. Asennuskotelon 3D-malli kuvassa 7. Kannen kanssa koko ohjelmointiaseman ulkomitoiksi tulee 380x280x180mm.



Kuva 7. Ohjelmointiaseman rungon 3D-malli. Fibox EKPE 380x280x100mm asennuskotelo kirkkaalla 80 mm korkealla kirkkaalla kannella.

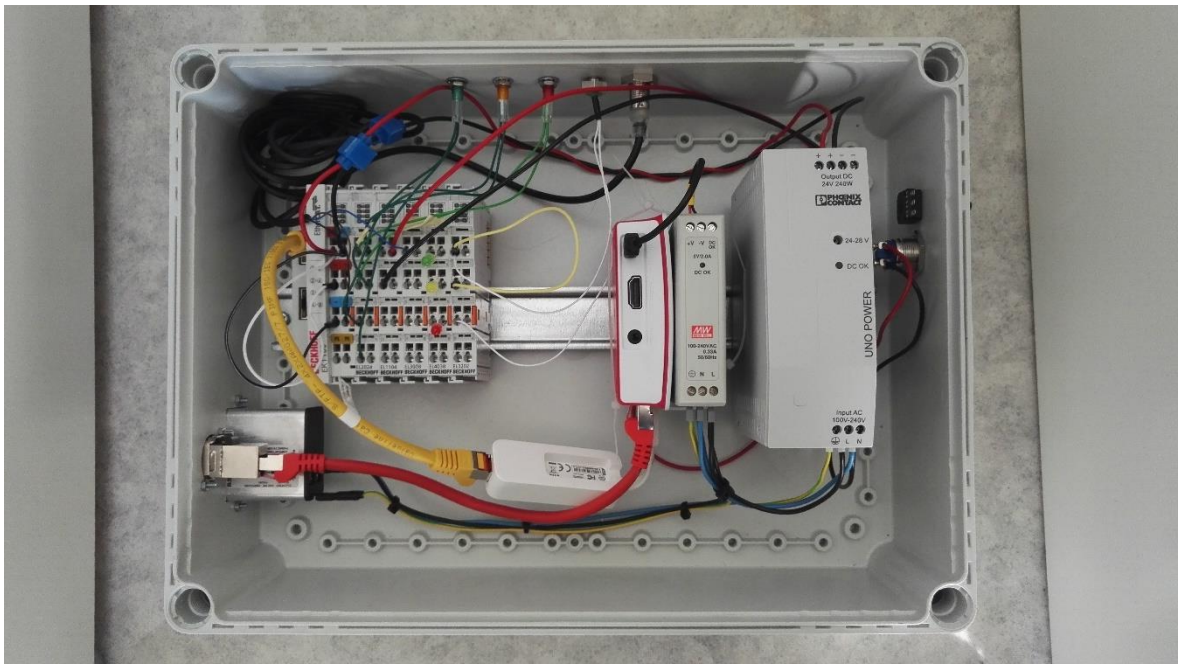
Kuten kuvassa 7. näkyy, on yläreunassa ja molemmilla sivuilla läpiviennit erilaisia komponentteja varten. Tämä mahdollistaa harjoitustyön testaamisen kansi kiinni, jolloin kytkennät on huomattavasti helpompi tarkistaa, kun johdoista I/O-moduuleissa roikkuvia komponentteja ei ole. Läpiviennit koneistettiin CNC-koneistuskeskuksella liitteessä I olevan valmistuspiirroksen mukaisesti. 3D-mallinnus ja valmistuspiirros ovat tehty Solidworks 2017 Student editionilla.

Laatikon sisään tulevat komponentit ovat seuraavat:

- Beckhoff EK1100 EtherCAT master
- Beckhoff EL2024 nelikanavainen digitaaliulostulo
- Beckhoff EL1104 nelikanavainen digitaalisisääntulo

- Beckhoff EL3008 kahdeksankanavainen analoogisisääntulo
- Beckhoff EL4038 kahdeksankanavainen analoogiulostulo
- Beckhoff EL3202 kaksikanavainen sisääntulo PT100 anturille
- Kolme hehkulamppua
- PT100-lämpötila-anturi
- Induktiivinen etäisyysanturi
- Yksitoiminen painike
- Potentiometri
- 24-volttinen virtalähde EtherCATille
- 5-volttinen virtalähde Raspberry Pi -tietokoneelle
- Raspberry Pi -tietokone

Komponenttien sijoittelu koteloon oli melko pitkällinen prosessi, jonka aikana komponentit olivat laatikossa monin eri tavoin. Lopulta päädyttiin sopivaan ratkaisuun, joka on kuvattu kuvassa 8. Komponentit ovat yhdellä DIN-kiskolla, joka on sijoitettu 11,5 cm kotelon yläreunasta. Vasemmalla puolella koteloa ovat I/O-moduulit ja oikealla molemmat virtalähteet, sekä Raspberry Pi. Näin on saatu tilaa uusien I/O-moduulien asentamiseen ja ohjelmointiaseman yleisilme on siisti ja järjestelmällinen.



Kuva 8. Lopullinen ohjelmointiaseman sisällön sijoitteluratkaisu.

Toimilaitteiden sijoittelu on tehty seuraavasti: Vasemmalla reunalla laatikkoa ovat virransyöttö- ja Ethernet-pistokkeet. Virransyöttöpistokkeelta haarautuu molemmille virtalähteille 230V kaapelit, jotka mahdollistavat aseman käytön verkkovirralla. Ethernet-pistokkeelta kulkee Ethernet kaapeli Raspberry:lle. Tätä johtoa pitkin ajetaan koodi Codesysistä Raspberryssä olevaan Runtimeen, joka ohjaa EtherCATia. Laatikon yläreunaan tulee reiät kolmelle hehkulampulle, lämpötila-anturille ja etäisyysanturille (Liite I). Näiden johdotus on helppo tehdä, sillä I/O-moduulit ovat lähes tulkoon läpivientien alla. Oikeassa reunassa olevat kaksi reikää ovat potentiometrille sekä painikkeelle. Näiden johdotus I/O-moduuleille vaatii pitkät kaapelit, mutta tämä oli laitteen selkeyden kannalta paras ratkaisu painikkeiden ja muiden käyttäjän kosketeltavana olevien komponenttien paikaksi.

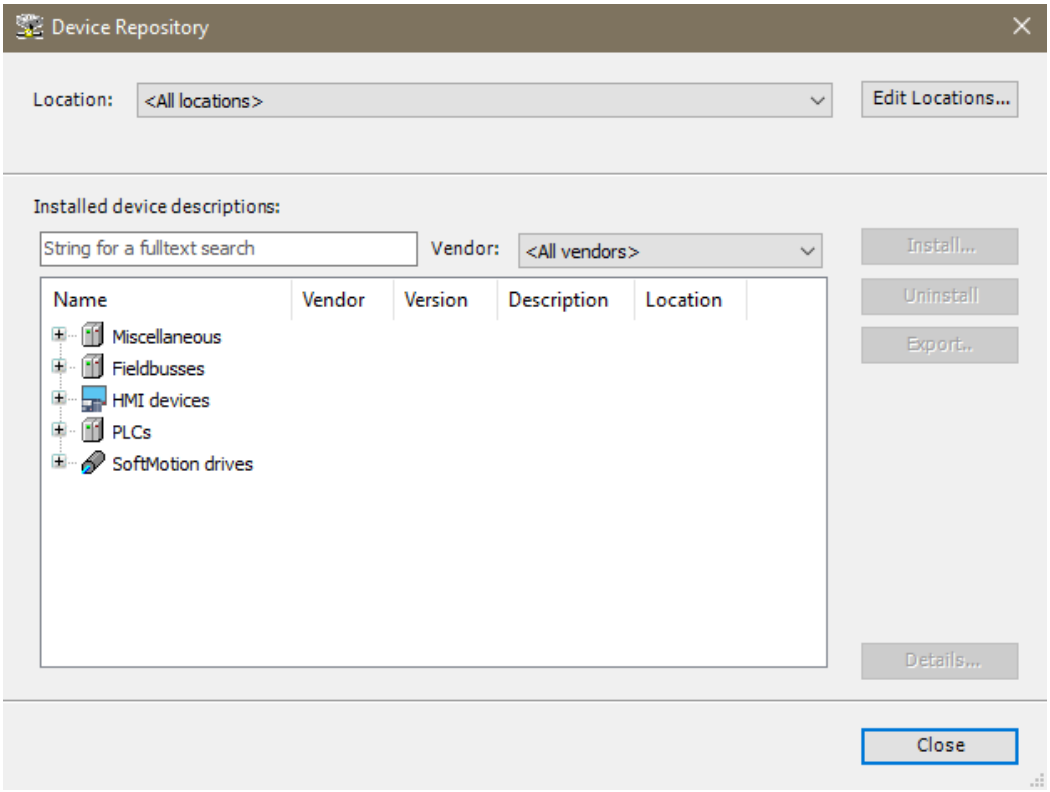
Komponenttien kytkemiseen ei tässä luvussa oteta kantaa, sillä riippuen harjoituksesta, niitä voidaan ja pitääkin kytkeä eri tavoin. Luvussa 4.4 on kerrottu, minkälaisia kytkentöjä tässä kandidaatintyössä kehitetyssä harjoituksessa voi esimerkiksi käyttää.

4 HARJOITUSTYÖN SUUNNITELU

Koska ohjelmointiasemassa on melko vähäinen määrä komponentteja, ei harjoitustyöstä voinut tehdä kovin monimutkaista. Ensimmäiseksi harjoitukseksi mekatroniikan opintoihin tämän kandidaatintyön tuotos on mitä luultavimmin sopiva. Erilaisia lähtötasoja totta kai on, mutta tämän harjoituksen ei pitäisi olla mahdoton edes asiasta mitään ymmärtämättömälle opiskelijalle. Ensimmäisenä oli kuitenkin saatava I/O-moduulit toimimaan halutulla tavalla.

4.1 I/O-moduulien laitekuvaus-tiedostot

I/O-moduulien toimiminen vaati Beckhoffin internetsivulta (Beckhoff 2018) XML laitteenkuvaus tiedostojen lataamisen, jotta Codesys ymmärtää minkälaisia laitteita on kytkettynä. Ladattava .zip -tiedoston koko oli 10Mt, mutta mikäli kaikkien laitteiden kuvaukset asentaa, kuluu levytilaa yli 10Gt. Tästä syystä asennettiin vain tarvittavien laitteiden tiedot, eli luvussa 3.4 listatut moduulit.



Kuva 9. Device Repository jossa on Codesysin oletuslaitteet, sekä mahdollisuus asentaa uusia XML tiedostoja.

Kuvassa 9 näkyvän Install-napin takaa pääsee valitsemaan asennettavan laitteen XML-tiedoston. Kun tiedosto on valittu ja painetaan asenna, kuluu hetki, jonka jälkeen laite on asennettu ja valmis käytettäväksi Codesysissä.

4.2 Harjoitustyön runko

Harjoitustyötä varten ajatuksena oli tehdä jonkinlainen kulunvalvontajärjestelmä, joka sisältäisi indikaattorivaloja tietyille toiminnoille, sekä anturitietoja etäisyys- ja lämpötila-antureilta. Koska komponentteja ei ole kovin montaa, harjoituksesta oli tehtävä yksinkertainen. Jotta harjoituksesta tuli mielekäs, oli siihen lisättävä visualisointi demonstroimaan sähköisiä laitteita. Tehtävänannon (Liite II) mukaan käytetään sähkölukkoa ja -summeria, mutta mikäli opiskelija kokee mielekkäämmäksi ajatella visualisoinnit näiden laitteiden indikaattorivaloina, on tämä totta kai yksi vaihtoehto.

Kulunvalvontajärjestelmä, joka yksilöisi sisään ja ulos kulkevat henkilöt, oli näillä komponenteilla mahdoton tehdä. Pienen ideoimisen jälkeen, päädyttiin siihen, että kulunvalvontajärjestelmä tulee huvipuiston kauhujen taloon, jossa saa paloturvallisuusmääräysten takia olla yhtä aikaa vain 10 ihmistä sisällä. Sisään- ja ulosmeno-ovien valvominen oli ajatuksena toteuttaa induktiivisilla etäisyysantureilla, mutta niiden rajallisen määrän takia ulosmeno-ovi oli tehtävä nappikäyttöiseksi, jotta myös ulosmenijät voidaan laskea. Induktiivinen anturi havaitsee vain ferriittisen materiaalin liikkeet, joten seuranta on toteutettava ovessa olevan metallin havaitsemisen avulla. Jotta sisäänmeno-ovi saatiin lukkoon suurimman sallitun ihmismäärän täyttyessä, oli siihen laitettava sähkötoiminen lukko. Lukkoa ei itseasiassa ole olemassa, joten sen tilalle voidaan Codesysin Visualisation-välilehdellä lisätä jonkinlainen visualisointi. Sähköinen ovilukko toimisi oikeassa elämässä 230 V jännitteellä, mutta koska käytetään analogi ulostuloa, ei jännitettä saada 10 volttia enempää.

Lämpötila-anturin tarpeellisuutta oli hieman mietittävä, mutta paloturvallisuusmääräyksistä kirjoittaminen antoi idean myös talon sisälämpötilan monitoroimisesta. Kesähelteellä kauhujen talon sisälämpötila voi helposti ylittää tehtävänannossa määrätyn 32°C, mutta harjoituksen vuoksi oli lämpötila saatava nousemaan helposti määritellyn rajan yli. 32°C on saavutettavissa anturiin puhaltamalla, tai esimerkiksi pitämällä siinä sormea jonkin aikaa. Mikäli tulipalo tai jonkin laitteen vaarallinen ylikuumeneminen pääsisi kauhujen talossa

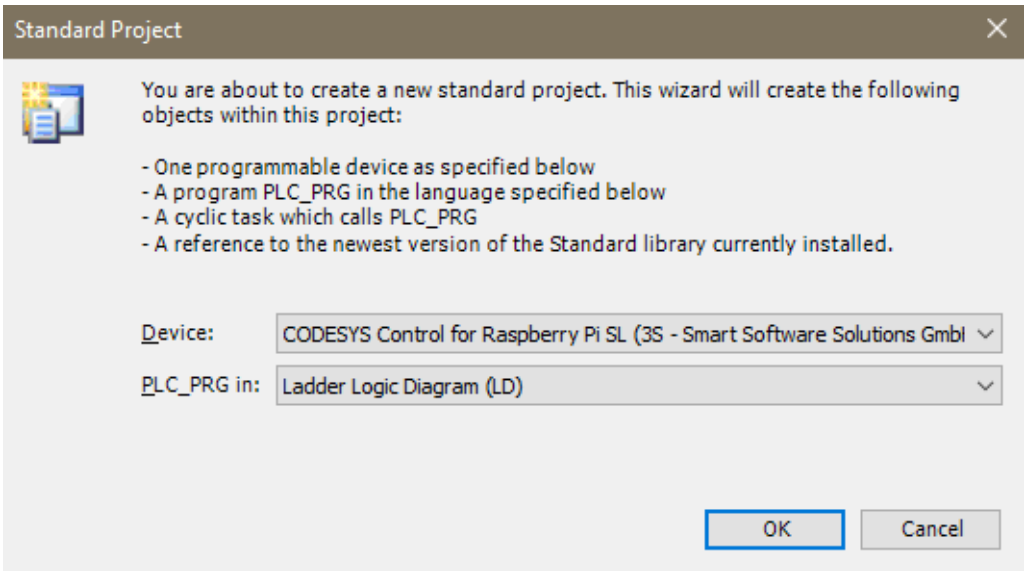
sattumaan, oli tietysti vielä lisättävä hälytyssummeri, joka ilmoittaa talon sisällä olijoille välittömästä poistumisen tarpeesta. Summeria ei kuitenkaan tässä tapauksessa ollut saatavilla, joten myös summerille tehdään visualisointi Codesysiin.

Kauhujen talon valvomoon on myös välitettävä tieto tilanteesta talon sisällä. Liikennevalotyyppinen järjestelmä, joka kertoo, onko talossa alle viisi, yli viisi vai 10 ihmistä, on tässä harjoituksessa riittävä. Jatkossa, mikäli ohjelmointiasemaan ostetaan esimerkiksi LED-näyttö, voisi valvomoon lähettää myös tarkan tiedon siitä, kuinka monta ihmistä sisällä on.

4.3 Harjoitustyön malliratkaisu

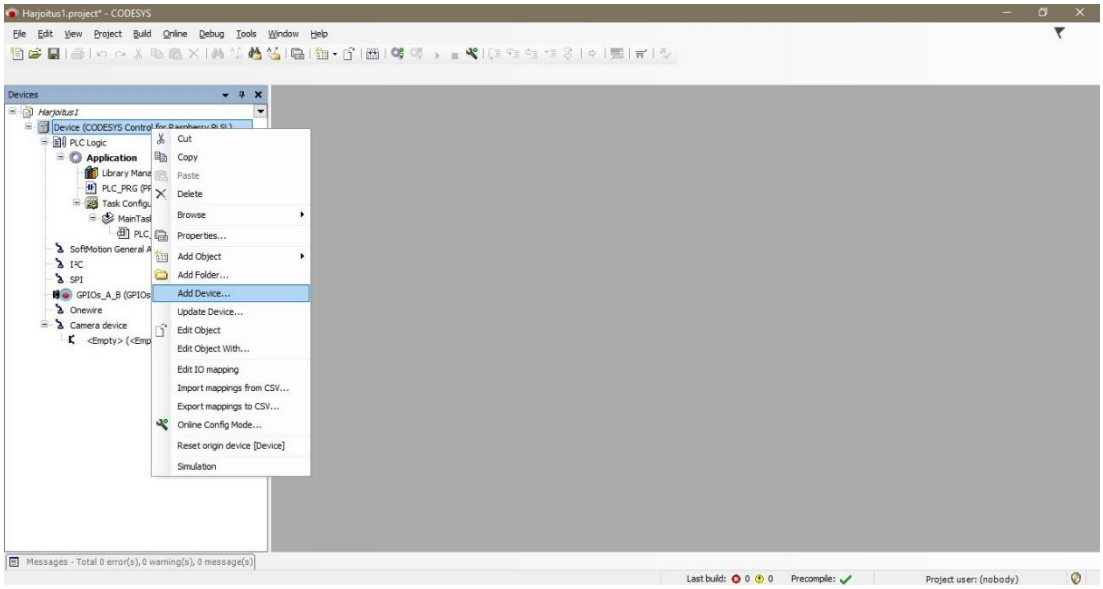
Beckhoffin internetsivuilla on laajalti dokumentaatiota, kuinka I/O-moduulit toimivat ja kuinka niitä käytetään. Ongelmaksi meinasi kuitenkin muodostua se, ettei Codesysin käyttöön moduulien ohjaimena meinannut löytyä juurikaan ohjeita tai dokumentaatiota. Moduulien käyttöönotto perustui siis Beckhoffin dokumentaatioon, sekä puhtaaseen päättelemiseen ja kokeilemiseen.

Codesysissä ensimmäinen askel on aloittaa uusi projekti, johon I/O-moduulit voidaan lisätä. Projektin asetuksissa tulee valita ohjelmointitavaksi Ladder Logic Diagram ja laitteeksi Codesys Control for Raspberry Pi SL (Kuva 10).



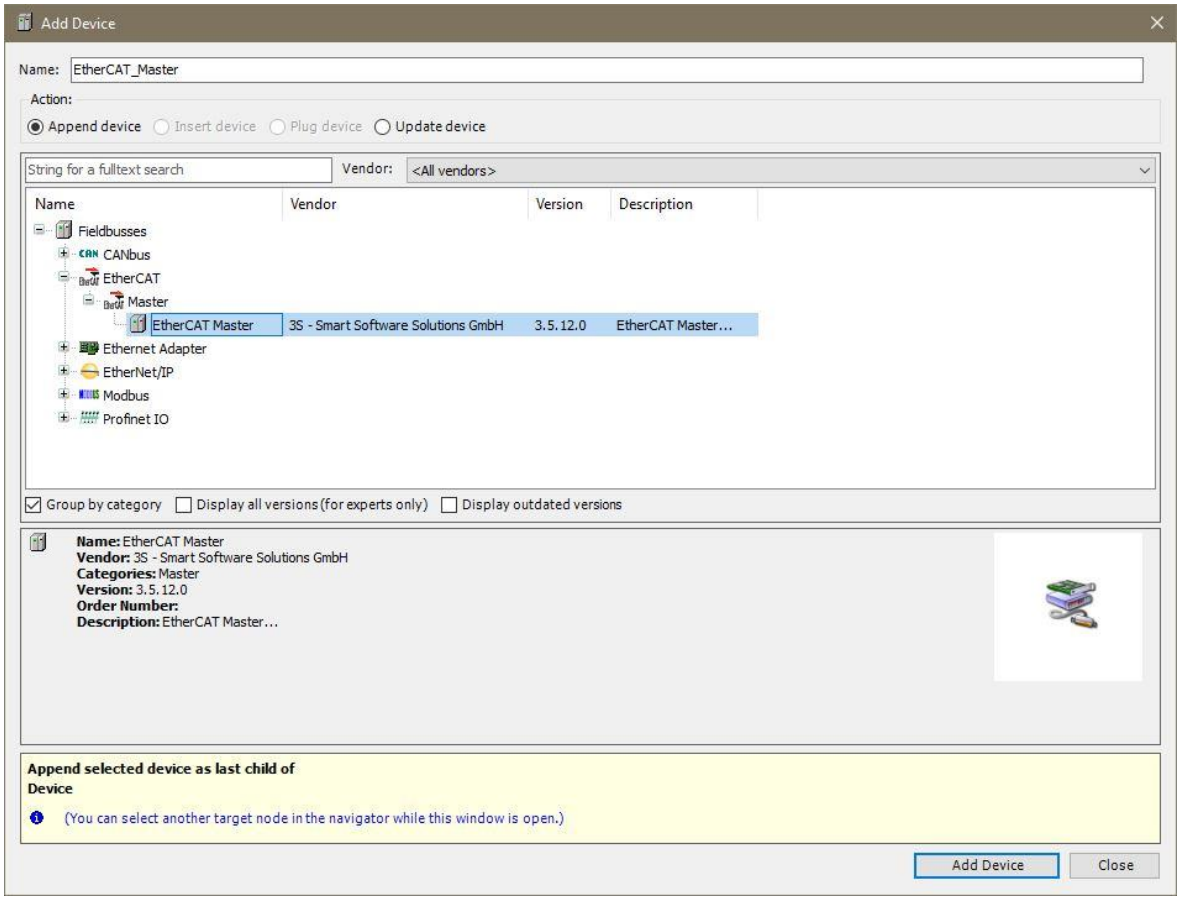
Kuva 10. Harjoitustyön ohjelmointi tapa ja PLC-laite.

Seuraavaksi on lisättävä EtherCAT master ja siihen kiinnitetyt I/O-moduulit projektiin. Tämä onnistuu klikkaamalla hiiren oikealla painikkeella Device-otsikkoa rakennepuusta Codesys-ikkunan vasemmasta reunasta ja valitsemalla Add device (Kuva 11). Kun laite on lisätty, on sitä vielä kaksoisklikattava rakennepuussa ja määritettävä laitteen MAC-osoite. MAC-osoite löytyy samalla tavalla Raspberryssä, kuin IP-osoite luvussa 3.4.



Kuva 11. EtherCAT masterin määritysvaihe.

EtherCAT masterin lisäämisen jälkeen, voidaan lisätä itse I/O-moduulit. Tämä tapahtuu klikkaamalla rakennepuusta edellisessä vaiheessa lisättyä EtherCAT masteria ja valitsemalla Add device. Nyt voidaan valita käytössä oleva EtherCAT moduuli, joka on tässä tapauksessa Beckhoff EK1100-0030. Tämän EtherCATin alle voidaan lisätä loput I/O-moduulit samaan tasoon. Hiiren oikealla painikkeella klikataan EtherCAT moduulin kohdalta jälleen Add device ja valitaan loput luvussa 3.4 mainitut moduulit.



Kuva 12. EtherCAT masterin mallin määritysvaihe.

Moduulien lisäämisen jälkeen voidaan alkaa pohtimaan itse koodia, eli kuinka I/O-moduuleita saadaan ohjattua liitteessä II halutulla tavalla. Tehtävänannon ensimmäinen kohta:

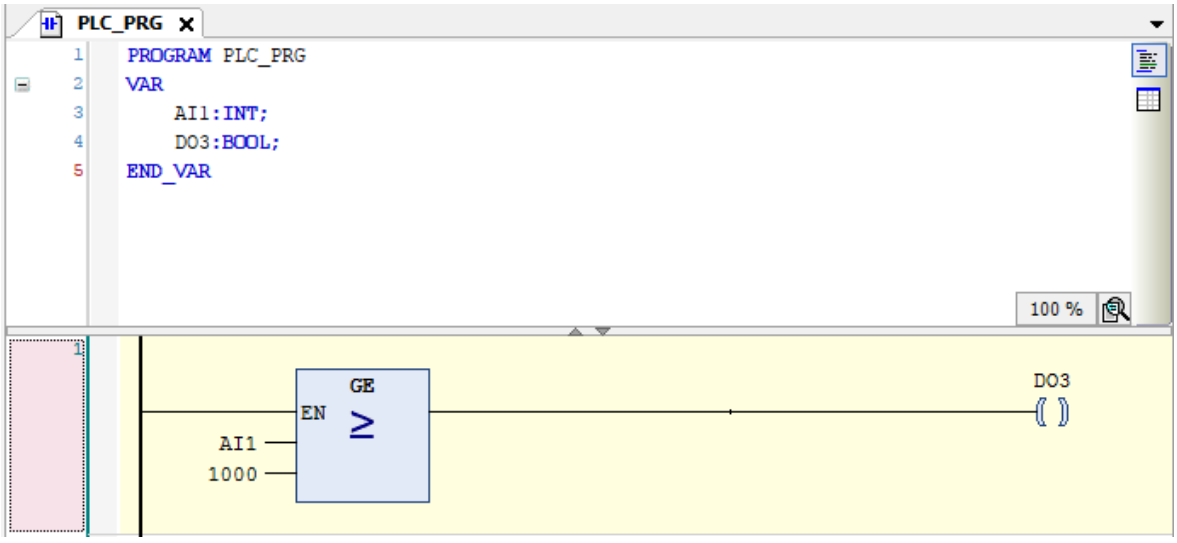
- Vihreän valon palaessa sisäänmeno-ovi on auki.

Oven liikkeitä seurataan yksitoimisella induktiivisella anturilla, joka antaa 0 voltin jännitettä oven ollessa kiinni ja 24 voltin jännitettä oven ollessa auki. Vihreän valon voisi siis kytkeä suoraan anturiin, mutta tässä harjoituksessa se ei olisi mielekästä, sillä painopiste on ohjelmallisella puolella.

Anturin voi kytkeä joko digitaaliseen tai analogiseen sisääntuloon. Digitaalinen sisääntulo voidaan linkittää ainoastaan boolean-muuttujaan, joka voi saada arvoja 0 ja 1. 24-volttisessa digitaaliseen sisääntuloon linkitetty muuttuja on siis nolla siihen asti, että sisääntuloon kohdistuu 24 V jännite. Analoginen sisääntulo taas toimii kuin jännitemittari. Siihen linkitetty muuttuja on integer-mallinen, eli saa kokonaislukuarvoja. Tässä tapauksessa 12-

bittinen EL3008-moduuli saa arvoja 2^{12}, eli 4096. Koska moduulin jännitealue on +-10 volttia, jakautuvat arvot nollan molemmin puolin, eli +-2048. Koodin testausvaiheessa huomattiin kuitenkin, että jostain syystä moduuli toimii 16-bittisillä arvoilla, vaikka se on kaikessa dokumentaatiossa 12-bittinen. Näin ollen oli käytettävä 2^{16} eli +-32768 kokonaislukualuetta.

On siis tehtävä induktiiviselle anturille oma muuttuja, joka kytketään tässä tapauksessa analogiseen sisääntuloon. Olkoon se *AI1*. Vihreä lamppu taas kytketään tässä työssä digitaaliseen ulostuloon. Lamppu kytkettiin digitaalisen ulostulon kolmanteen porttiin, eli sen muuttuja olkoon *DO3*. Jotta lamppu syttyy, kun ovi avataan, on tehtävä uusi network, johon lisätään GE-operaattori(Greater or Equal), joka sytyttää lampun, kun anturilta tuleva jännite ylittää määritetyn rajan. Oven ollessa kiinni jännitteen integer-arvo hyppi mittaustarkkuuden takia noin 0-20 välillä, maksimin ollessa jo edellä mainittu 32768. Arvoksi käy siis mikä vain arvo 20-32768 väliltä. Esimerkkikoodissa on käytetty arvoa 1000 joka on noin 0,3 volttia. Kuvassa 13 on esitetty koodi, jolla valo syttyy oven auetessa.



Kuva 13. Koodi jolla muuttujan *AI1* ylittäessä arvon 1000 muuttujan *DO3* arvo muuttuu 0:sta 1:si.

Seuraavaksi muuttujat on kytkettävä fyysisiin sisään- ja ulostuloihin. Tämä on melko yksinkertaista. Klikataan hiiren oikealla painikkeella EK1100-moduulia ja valitaan Edit IO mapping, josta sitten valitaan *DO3* muuttujalle digitalisen ulostulon 3. portti ja *AI1*

muuttujalle analogisen ulostulon ensimmäinen Value muuttuja, joka on tässä tapauksessa osoitteessa %IW2.

Harjoitustehtävän seuraavat kohdat ovat (Liite II):

- Keltaisen valon palaessa sisällä on ainakin 5 ihmistä.
- Punaisen valon palaessa sisällä on 10 ihmistä ja sisäänmeno-ovi menee lukkoon 3 voltin sähkölukolla. Kun ihminen poistuu talosta, sammuu valo ja sisäänmeno-ovi aukeaa lukosta.

Jotta sisään menijöitä voidaan laskea, on tehtävä network, joka laskee kuinka monta kertaa vihreä valo syttyy. On tehtävä uusi muuttuja, joka nimetään selkeyden vuoksi nimellä: *ihmisia*. Muuttujan tyypiksi kirjoitetaan integer, sillä ihmiset ja oven avauskerrat ovat kokonaislukuja. Tähän muuttujaan on lisättävä jokaisella oven avauskerralla numero yksi. Koska ulosmeno-ovi avataan napilla, pitää muuttujasta *ihmisia* vähentää numero yksi, kun ihminen poistuu talosta. Näin muuttuja *ihmisia* pysyy koko ajan samana kuin ihmisten määrä talossa. Codesysissä on matemaattisia funktiolaatikoita, sekä Function Blocks -otsikon alapuolella on myös laskurilaatikoita, jotka tekevät käytännössä saman asian. Molempia voi käyttää tehtävän ratkaisemiseen, mutta tässä ratkaisumallissa on käytetty matemaattisia laatikoita. Uuteen networkiin on siis vedettävä ADD-operaattori, jonka eteen on vedettävä anturia varten jo ensimmäisessä kohdassa GE-operaattori. Näin saadaan anturin liikkeet linkitettyä *ihmisia*-muuttujan lisäyslaatikkoon. Jotta muuttujan arvo kasvaa vain yhdellä oven auetessa, on lisättävä edge detection ennen ADD-operaattoria. Näin laskuri lisää muuttujaan yhden vain, kun arvossa tapahtuu muutos, eikä laske aikaperusteisesti muuttujaan ykkösiä lisää koko siltä ajalta, kun ovi on auki.

SUB-operaattori, eli vähennysoperaattori vähentää muuttujan arvosta yhden. On siis tehtävä uusi network, johon lisätään SUB-operaattori. Tätä operaattoria ohjataan ulosmeno-oven napilla, joka määritetään nyt *DI2*-nimiseksi integer-muuttujaksi. SUB-operaattorin eteen on lisättävä contact-operaattori, jonka muuttujaksi määritetään juuri luotu *DI2*-muuttuja. *DI2*-muuttuja on vielä määritettävä digitaalisen ulostulon porttiin numero 2 joka on osoitteessa *%IX0.1* Edit IO mapping -välilehdellä.

Keltainen valo voidaan tehdä lähestulkoon samalla kaavalla kuin vihreä valo tehtiin. Tehdään uusi integer-muuttuja valolle: *DO2*. Ero vihreään valoon on siinä, että networkiin on vedettävä kaksi funktiolaatikkoa. Toinen sytyttää valon, jos sisällä on viisi tai enemmän ihmisiä. Tämä onnistuu GE-operaattorilla. Toinen funktiolaatikko sammuttaa valon, jos talossa on 10 ihmistä. Tämä tehdään LE-operaattorilla(Less or Equal). Operaattorien jälkeen lisätään vielä yläpalkista coil-operaattori, jonka muuttujaksi määritetään lampun *DO2*-muuttuja. Muuttuja tulee vielä määrittää digitaaliulostuloon osoitteeseen *%QX0.1*. Jotta punainen valo saadaan sytytettyä, kun ihmisiä on talon sisällä 10, on käytettävä vielä toista GE-funktiolaatikkoa. Eli kun muuttujan *ihmisia* arvo on suurempi tai yhtä suuri kuin kymmenen, sytytetään punainen lamppu, joka määritetään boolean-muuttujaan *DO1*. Tämä muuttuja linkitetään digitaaliulostulon osoitteeseen *%QX0.0*.

Sähkölukon käyttäminen tapahtuu tässä harjoituksella analogiulostulolla, sekä visualisoinnilla Codesysissä, sillä sähkölukkoa ei ollut saatavilla. Tässä kohtaa päästään tutustumaan myös toiseen ohjelmointitapaan, eli Structured textiin. Lukon käyttämiseen olisi myös muita tapoja, mutta harjoituksen monipuolisuuden vuoksi tässä kandidaatintyössä käytetään Structured textillä kirjoitettua koodia, jotta opiskelijat tutustuvat myös tähän ohjelmointitapaan. Koodia varten on tehtävä uusi network, johon lisätään execute-operaattori. Operaattorissa on tekstikenttä, johon voi kirjoittaa Structured text -kielellä koodia. Tähän käyttötarkoitukseen kirjoitetaan seuraavassa kuvassa esitetty koodi:

```
1   IF DO1=1
2       THEN
3           AO1:=10000;
4       ELSE
5           AO1:=0;
6   END_IF
7
```

Kuva 14. Lukkoa kontrolloiva Structured text -koodi.

Koodissa on yksinkertainen if-lause, joka asettaa analogiulostulon arvoksi 10000, eli noin 3 volttia, kun muuttujan *DO1* arvo on yksi, eli punainen valo palaa. ELSE-käsky määrää analogiulostulon arvoksi nolla, mikäli punainen valo ei pala. Näin lukko on auki, kun talossa

on alle 10 ihmistä ja lukitus menee päälle, kun kymmenes ihminen astuu taloon. Edellisessä kappaleessa koodattu laskuri myös sammuttaa punaisen valon ja sitä kautta avaa lukon, kun kymmenes ihminen poistuu talosta.

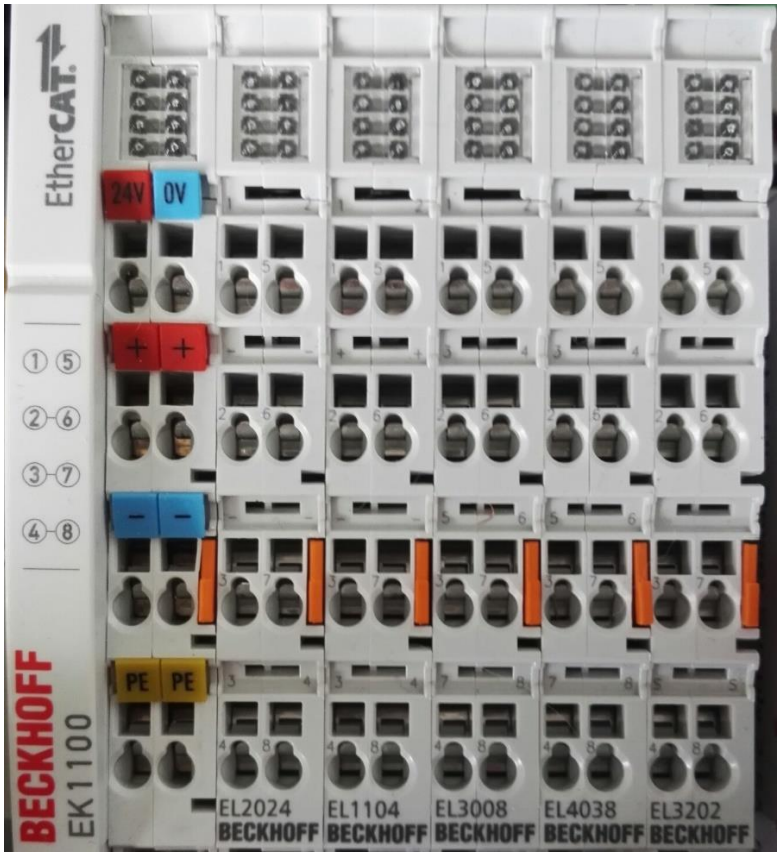
Viimeinen kohta harjoitustyössä on lisätä lämpötilan seuranta taloon käyttäen PT100-lämpötila-anturia. Tämä on tulipalon huomaamista varten, joten oikeasti lämpötilan kriittiseksi rajaksi tulisi asettaa paljon korkeampi luku kuin tässä harjoituksessa asetetaan. Jotta anturi saadaan hälyttämään esimerkiksi puhaltamalla siihen, tai pitämällä sitä kämmenen lämmössä, asetetaan rajaksi 32°C. Lämpötila-anturi kytketään sille tarkoitettuun moduuliin, EL3202, joka näyttää lämpötilan integer-muuttujana kymmenkertaisina celsiusasteina. Esitystapa on siksi näin, että integer-muuttujat voivat saada vain kokonaislukuarvoja, joten kymmenkertaisena saadaan mittatarkkuudeksi 0.1°C. Tämä on tämän harjoituksen sovellutukseen riittävä. Tehdään anturille integer-muuttuja, jonka nimeksi tulee tässä tapauksessa *lampotila*. Summerina toimii sähkölukon visualisointi, koska itse lukkoa ei näillä näkymin saada hankittua. visualisoinnille tehdään analogiulostuloon sopiva muuttuja, *AO3*.
Eri ohjelmointitapojen harjoittelun vuoksi, tehdään lämpötilanseurantajärjestelmä Structured text -ohjelmointikielellä. Uuteen networkiin siis lisätään jo tutuksi tullut execute-operaattori, johon voi ohjelmoida tekstillä. Koodi on helpointa toteuttaa if-lauseella, vaikka muitakin tapoja toki olisi. Tässä harjoituksessa käytetään if-lausetta, sillä se tukee myös esimerkiksi Ohjelmoinnin perusteet -kurssin opintotavoitteita.

Koodi on melko yksiselitteinen: jos lämpötila nousee yli 32°C:n *AO3*, eli summeri saa 3 voltin jännitteen ja alkaa soimaan. *AO1* kytketään nollaksi, eli sisäänmeno-oven lukko on auki, jotta sitä kautta pääsee myös ihmisiä pois tulipalon keskeltä. *DO1, DO2 ja DO3.* -muuttujien arvoksi annetaan 1, jotta myös valvomossa huomataan tulipalo, kaikkien valojen syttyessä. Muuttuja *ihmisia* nollataan, jotta väärän hälytyksen jälkeen voidaan taas jatkaa kauhujen talon normaalia toimintaa. Mikäli lämpötila on alle 32°C, on summerin muuttujan arvo 0 ja muut muuttujat noudattavat aikaisempaa koodia. Muuttujat *AO3* ja *lampotila* ovat kytkettävä vielä fyysisille I/O-moduuleille. Codesysin Edit IO mapping -välilehdeltä asetetaan lämpötila-anturin muuttuja osoitteeseen *%IW18* moduulille EL3202. *AO3*, eli summerin ulostulo on määritettävä analogiulostulon osoitteeseen *%QW2*. Visualisation-välilehdelle on lisättävä visualisoinnit muuttujille *AO1* ja *AO3.* Tämä tapahtuu vetämällä

haluttu visualisointi, esimerkiksi laskuri tai lamppu, Visualisation-ikkunaan ja kytkemällä muuttujan tähän objektiin.

4.4 Ohjelmointiaseman kytkentä harjoitustyötä varten

Kun harjoitustyö on ohjelmallisesti valmis, on jäljellä enää asiaan kuuluvien kytkentöjen tekeminen ohjelmointiasemaan. Kytkentöjä voi tietenkin varioida, riippuen mille ulos- tai sisääntulolle tietty muuttuja on määritetty. Tässä kappaleessa on selitetty edellisen kappaleen koodille ja muuttujamäärityksille oikeat kytkennät. Kytkentäkaavio on esitetty liitteessä III. Tässä kappaleessa on selitetty, kuinka kytkennät tulee tehdä.



Kuva 15. I/O-moduulien kytkentätaulut.

Kuten kuvasta 15 näkee, on kytkentätaulussa numeroidut koskettimet. Kytkentöjen tekeminen selitettään alkaen vasemmalta ja loppuen oikealla olevaan moduuliin. Ensimmäiseen moduuliin, eli EtherCAT masteriin, on tuotava 24 V jännite virtalähteeltä. Positiiviset ja negatiiviset johdot kytketään koskettimiin 1 ja 5. Jotta muille moduuleille kytkeytyy virta, ovat koskettimet 1 ja 2, sekä koskettimet 5 ja7 kytkettävä yhteen. Näin

EtherCAT masterilla ja kaikilla moduuleilla on toimintaansa tarvittava virta. Koskettimilta 6 ja 3 otetaan virta induktiiviselle etäisyysanturille.

Seuraavana kytkettävänä moduulina on EL2024 eli digitaalinen ulostulo. Yksi ulostulo tarvitsee positiivisen ja negatiivisen kontaktin, joten Beckhoff on määrittänyt koskettimet pareiksi seuraavalla tavalla: 1&2, 3&4, 5&6 ja 7&8. Jokaisen moduulin kyljessä on kytkentäkaavio, joka kertoo mikä kosketin on positiivinen ja mikä negatiivinen. Punainen lamppu on kytkettävä 1 ja 2 koskettimiin, jotta se toimii Codesysiin määritettynä *DO1* -muuttujana. Keltainen lamppu kytketään samalla periaatteella koskettimiin 5 ja 6. Edelleen samaa periaatetta käyttäen, kytketään vihreä lamppu koskettimiin 3 ja 4. Digitaaliseen sisääntuloon, EL1100; kytketään ainoastaan nappi. Digitaalisen sisääntulon koskettimet ovat samanlaisessa järjestyksessä kuin digitaalisessa ulostulossa, joten *DI2*-muuttujan nappi kytketään koskettimiin 5 ja 6.

Analogiset sisään- ja ulostulot ovat 8 kanavaisia, eli jokaisella koskettimella on oma osoitteensa Codesysissä. EL3008-moduuliin kytketään ainoastaan etäisyysanturin signaalijohto. Koska muuttuja on *AI1* ja määritetty ensimmäiseen ulostuloon, kytketään johto koskettimeen 1. Analogiseen ulostuloon halutaan lukko ja summeri muuttujilla *AO1* ja *AO3*. Koska jokainen kosketin on oma ulostulonsa, voidaan ledit kytkeä siten, että negatiivinen johto laitetaan käyttämättömänä olevaan koskettimeen. Niiden jännite on 0 V ja käytössä olevan koskettimen jännite muuttuu Codesysin käskyjen mukaan, joten siihen kytketty laite toimii, kun koodi niin käskee. Laitteiden positiiviset johdot ovat kuitenkin kytkettävä oikeisiin koskettimiin, eli 1. ja 3. Koska tähän harjoitukseen ei itse sähköisiä laitteita saatu järjestettyä, jätetään analogi ulostulo ilman kytkentöjä.

Viimeinen kytkentä on EL3202-moduulin kytkeminen. PT100-anturin kytkeminen tapahtuu niin, että anturin johdot kytketään koskettimiin 5 ja 7. Koska moduuli on tarkoitettu kolmi- tai nelijohtoiselle anturille, on koskettimet 5 ja 6 kytkettävä yhteen, jotta anturi toimii.

5 JOHTOPÄÄTÖKSET

Ohjelmointiaseman suunnittelu oli onnistunut projekti. Mekatroniikan kurssin harjoitustöiden testaaminen saatiin siirrettyä mikroluokkiin ja harjoituksiin saatiin konkreettinen rajapinta.

Tavoitteina oli tehdä helposti mukana liikkuva, kompakti PLC-ohjelmointiasema ja tässä onnistuttiin hyvin. Ohjelmointiaseman toiminnallisuuteen jäi kuitenkin vielä parantamisen varaa, lähinnä komponenttien rajallisen määrän takia. Mikäli asemaan olisi saanut esimerkiksi sähkömoottorin ja summerin, olisi ensimmäisestä harjoituksesta voinut tehdä monipuolisemman. Asemaa tullaan jatkossa laajentamaan toimilaitteilla ja lisäkomponenteilla, joten sen toiminnallisuus ja monipuolisuus tulee varmasti paranemaan. Kuitenkin, verrattuna vanhaan käytäntöön, jossa harjoitustöitä testattiin yhden kerran laboratoriossa, antaa kehitetty ohjelmointiasema vapauden testata harjoitustyötä lähestulkoon missä luokassa tahansa, sekä käytännön harjoittelukertoja kertyy enemmän.

Suunnitellusta harjoitustyöstä tuli myös onnistunut. Opiskelijat saavat tehtävän, joka vaikeutuu harjoituksen edetessä, pitäen motivaation yllä ja antaen onnistumisen tunteita, kun jokin asia onnistuu. Tehtävä voi olla haastava, mikäli yhtään kokemusta mekatroniikasta ei ole, mutta helpommat alkutehtävät valmentavat opiskelijaa harjoituksen aikana myöhemmin tuleviin vaikeampiin osioihin. Malliratkaisu ja esimerkkikoodi on suunnattu harjoitustenpitäjälle, jonka työtä tulee varmasti helpottamaan alusta loppuun asti suunniteltu harjoitustyö. Koska asemaa tullaan laajentamaan lisälaitteilla, saadaan kurssille tulevaisuudessa todella monipuolisia harjoituksia. Nykyisillä komponenteilla saa myös suunniteltua erilaisia harjoitustöitä.

LÄHTEET

Bishop, R. 2008. Mechatronic system control, logic and data acquisition. Boca Raton, Yhdysvallat: CRC Press.

CODESYS. 2018a. [Codesysin verkkosivu]. [viitattu 19.4.2018]. Saatavissa: https://www.codesys.com/#_

CODESYS Runtime. 2018b. [Codesysin verkkosivu]. [Viitattu 8.5.2018]. Saatavissa: https://www.codesys.com/products/codesys-runtime.html

IEC 61131-3. 2013. International standard for programmable controller programming languages. Second edition. International Electrotechnical Commission. [viitattu 19.4.2018]

John, K. & Tiegelkamp, M. 2010. IEC 61131-3: Programming Industrial Automation Systems. Heidelberg, Saksa: Springer Verlag GmbH. 390 s.

Keinänen, T., Kärkkäinen, P., Metso, T. & Putkonen, K. 2001. Koneautomaatio 2: Logiikat ja ohjausjärjestelmät. 1. painos. Vantaa: Tummavuoren Kirjapaino Oy. 410 s.

Klein Keane & Van De Viele. 2017. Take These Steps to Secure Your Raspberry Pi Against Attackers. Verkkodokumentti. [viitattu 15.4.2018]. Saatavissa: https://makezine.com/2017/09/07/secure-your-raspberry-pi-against-attackers/

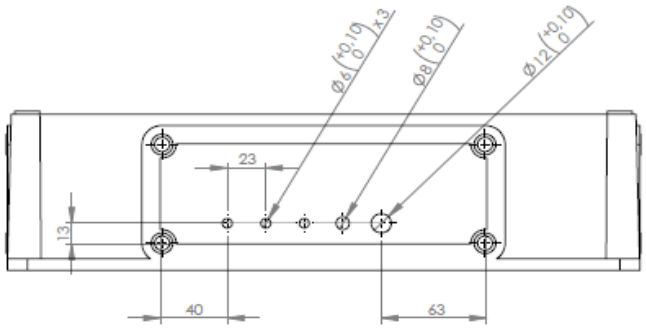
Raspberry Pi. 2018a. [Raspberry Pin verkkosivu]. [viitattu 15.4.2018]. Saatavissa: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

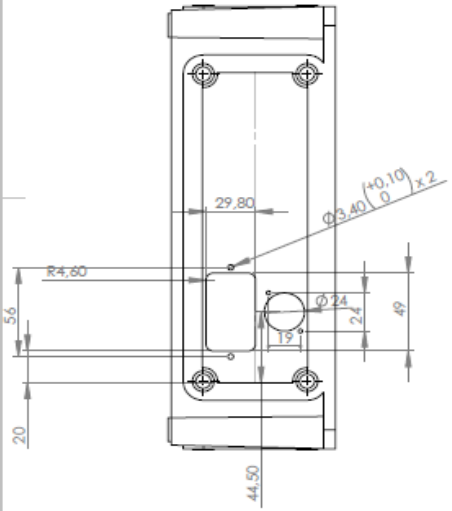
Raspberry Pi. 2018b. [Raspberry Pin verkkosivu]. [viitattu 15.4.2018]. Saatavissa: https://www.raspberrypi.org/documentation/installation/noobs.md

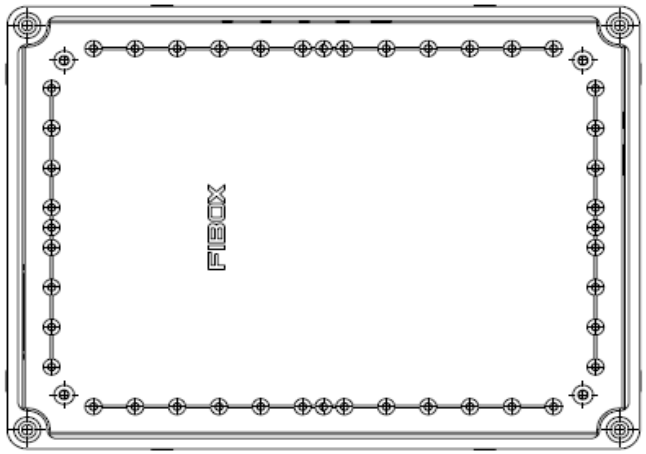
Tietoja SOLIDWORKISTA. Solidworks. 2018. [Solidworksin verkkosivu]. [viitattu 19.4.2018]. Saatavissa: http://www.solidworks.fi/sw/6453_SVF_HTML.htm

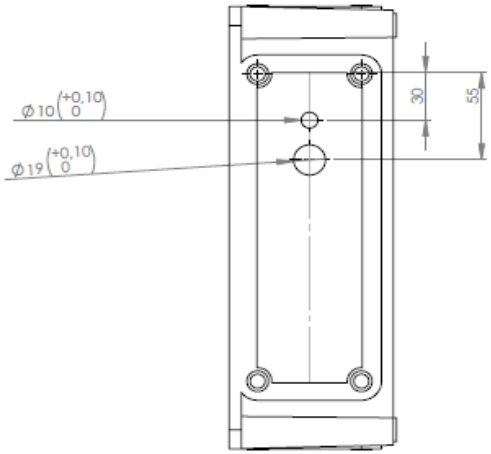
XML Device Description. Beckhoff. 2018. [Beckhoffin verkkosivu]. [viitattu 19.4.2018]. Saatavissa: https://beckhoff.fi/english/download/elconfg.htm?id=19839196423490140

LIITE I









UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION
	NAME	SIGNATURE	DATE			TITLE: Ohjelmointiympäristön laatikko
DRAWN	T.Huuskonen		20.3.2018			
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL: PC		DWG NO. Fibox_EKPE / A3
				WEIGHT:		SCALE:1:3 / SHEET 1 OF 1

Harjoitustyö 1

Eräässä huvipuistossa on kauhujen talo, jossa ei paloturvallisuusmääräysten takia saa olla kuin 10 ihmistä kerrallaan.

Huvipuiston johtaja haluaa vähentää työntekijöistä koituvia kustannuksia ja aikoo korvata ihmisiä sisään ja ulos laskevat työntekijät PLC-pohjaisella ohjausjärjestelmällä, jonka valvomiseen riittää yksi työntekijä. Valvomoon tulisi saada seuraavanlainen valvontajärjestelmä:

- Vihreän valon palaessa sisäänmeno-ovi on auki.
- Keltaisen valon palaessa sisällä on ainakin 5 ihmistä.
- Punaisen valon palaessa sisällä on 10 ihmistä ja sisäänmeno-ovi menee lukkoon 3 voltin sähkölukolla. Kun ihminen poistuu talosta, sammuu valo ja sisäänmeno-ovi aukeaa lukosta.
- Sisään meneviä ihmisiä lasketaan oven liikkeiden perusteella induktiivisella etäisyysanturilla, joka kertoo onko ovi auki vai kiinni.
- Ulosmeno-ovi aukeaa napilla, jotta ulosmenijät voidaan laskea.
- Mikäli lämpötila talossa ylittää 32 astetta, syttyvät kaikki valot, sisäänmeno-ovi ei ole lukossa ja 3 voltin summeri alkaa soimaan käskien ihmiset ulos. Lämpötilan laskettua laskuri alkaa taas laskemaan nollasta ylöspäin.

Käytössä on seuraavat komponentit:

Beckhoff I/O-moduulit:

- EK1100 EtherCAT master
- EL2024 4-kanavainen digitaali-input
- EL1104 4-kanavainen digitaalioutput
- EL3008 8-kanavainen analogi-input
- EL4038 8-kanavainen analogioutput
- EL3202 2-kanavainen analogi-input PT100-anturille

Komponentit

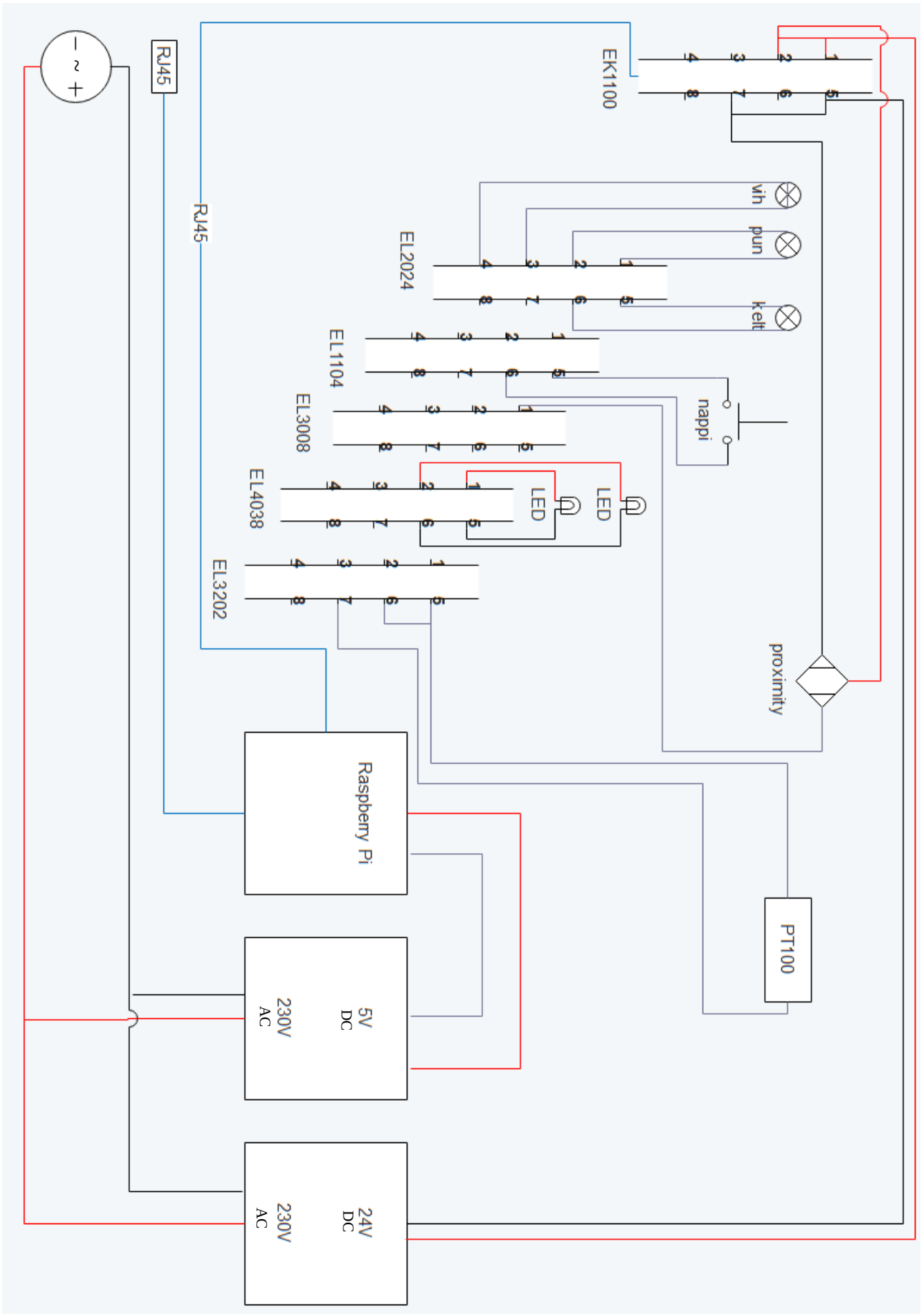
- Kolme hehkulamppua
- PT100 Lämpötila-anturi
- Induktiivinen etäisyysanturi

- 2 LED-lamppua visualisoimaan summeria ja lukkoa
- Nappi

Vinkkejä:

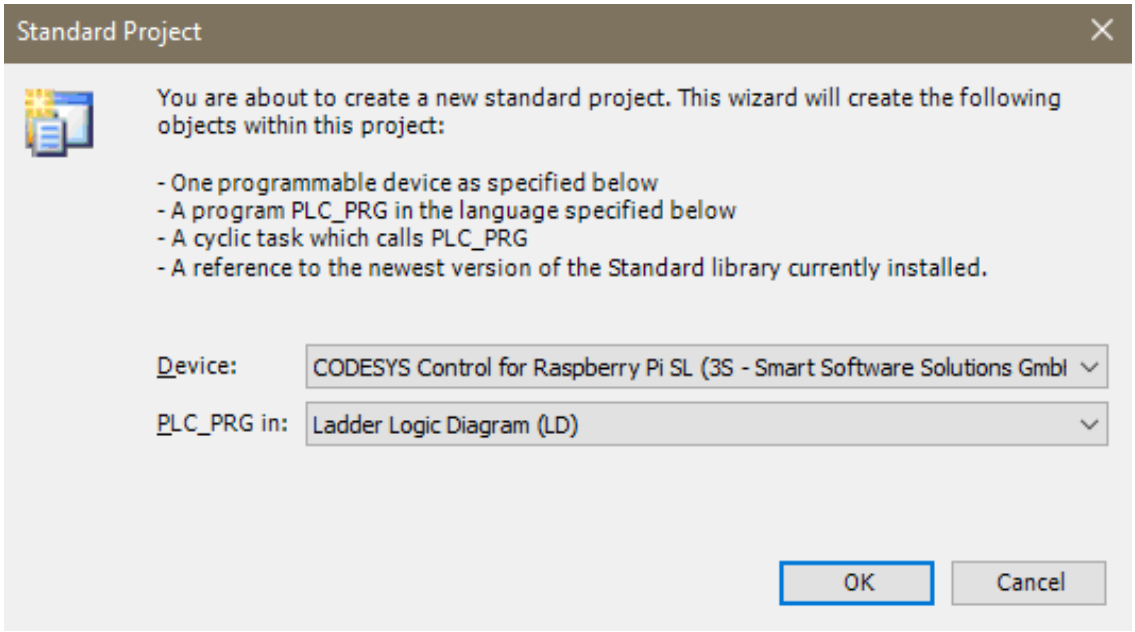
- Ohjelmointi suositeltavaa tehdä Ladder-Logicilla, mutta valinta on opiskelijalla itsellään.
- Etäisyysanturin voi kytkeä joko digitaaliseen tai analogiseen inputtiin.
- Lämpötila-anturi on kytkettävä sille tarkoitettuun EL3202-moduuliin.
- Analogi-outputit toimivat jostain syystä 16-bittisen arvoilla, vaikka moduuleissa lukee 12-bit
- Beckhoffin internetsivuilta löytyy todella hyvät ohjeet I/O-moduulien käyttöön sekä esimerkiksi kytkentäkaavioita ja esimerkkejä.

LIITE III


EK1100
RJ45
RJ45
EL2024
EL1104
EL3008
EL4038
EL3202
vih
pun
kelt
nappi
LED
LED
proximity
Raspberry Pi
PT100
5V
DC
230V
AC
24V
DC
230V
AC

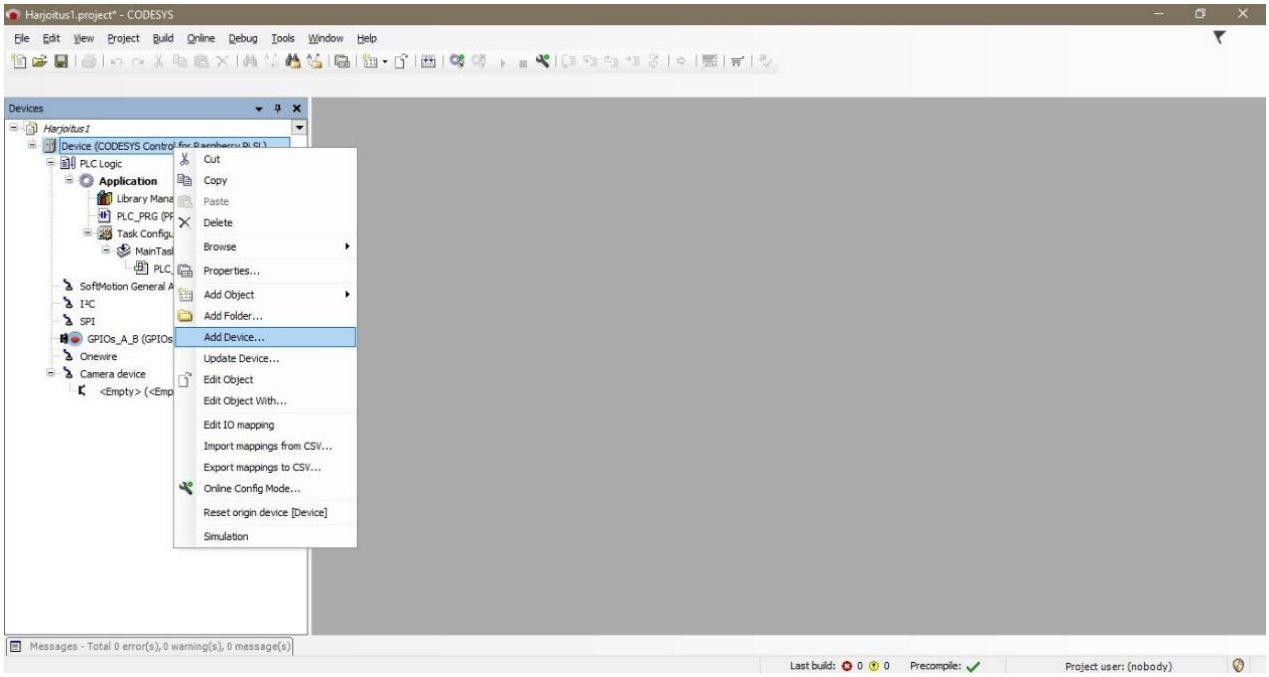
Codesys ohjelmointiaseman käyttöönotto on fyysisiltä kytkennöiltään helppo. Virtajohto kytketään seinään ja ethernet-johto kiinni tietokoneeseen. Laitteen kyljessä olevasta ethernet-läpiviennistä kuuluu aseman sisällä olevan johdon olla kytketty Raspberryn omaan ethernet-porttiin. Raspberrystä taas kuuluu lähteä usb-adapterilla toinen ethernet-johto EtherCATin inputporttiin. Raspberry lähtee päälle painamalla virtajohdon liittimestä nappi ON-asentoon.

Codesysissä ensimmäinen askel on aloittaa uusi projekti, johon I/O-moduulit voidaan lisätä. Projektin asetuksissa tulee valita ohjelmointitavaksi Ladder Logic Diagram ja laitteeksi Codesys Control for Raspberry Pi SL(Kuva 1).



Kuva 1. Harjoitustyön ohjelmointi tapa ja PLC-laite.

Seuraavaksi on lisättävä EtherCAT master ja siihen kiinnitetyt I/O-moduulit projektiin. Tämä onnistuu klikkaamalla hiiren oikealla painikkeella Device-otsikkoa rakennepuusta Codesys-ikkunan vasemmasta reunasta ja valitsemalla Add device(Kuva 2). Kun laite on lisätty, on sitä vielä kaksoisklikattava rakennepuussa ja määritettävä laitteen MAC-osoite. MAC-osoite löytyy samalla tavalla Raspberryssä, kuin IP-osoite luvussa 2.4.



Kuva 2. EtherCAT masterin määritysvaihe.

EtherCAT masterin lisäämisen jälkeen, voidaan lisätä itse I/O-moduulit. Tämä tapahtuu klikkaamalla rakennepuusta edellisessä vaiheessa lisättyä EtherCAT masteria ja valitsemalla Add device. Nyt voidaan valita käytössä oleva EtherCAT-moduuli, joka on tässä tapauksessa Beckhoff EK1100-0030. Tämän EtherCATin alle voidaan lisätä loput I/O-moduulit samaan tasoon. Hiiren oikealla painikkeella klikataan EtherCAT-moduulin kohdalta jälleen Add device ja valitaan käytössä olevat moduulit.

Kun moduulit on lisätty, otetaan yhteys EtherCATiin kaksoisklikkaamalla rakennepuusta otsikkoa Device ja kirjoittamalla Raspberryn IP-osoite oikean puoleiseen IP-osoitekenttään. Enter-painiketta painamalla Codesys ottaa yhteyden laitteeseen.

Näin ohjelmointiasema on valmis ottamaan vastaan ajettavaa koodia.