

Automaatiotoimilaitteiden yhteisen ohjauksen suunnittelu

Alexi Niemi

OPINNÄYTETYÖ
Toukokuu 2024

Autotekniikka
Älykkäät koneet

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Autotekniikka
Älykkäät koneet

NIEMI, ALEKSI:
Automaatiotoimilaitteiden yhteisen ohjauksen suunnittelu

Opinnäytetyö 37 sivua, joista liitteitä 1 sivua
Toukokuu 2024

Opinnäytetyössä käsitellään useamman aktuaattorin yhteisen ohjauksen suunnittelua. Työn tavoitteena oli luoda pohja prosessille, jossa siirrettäisiin kappale paikasta A paikkaan B hyödyntäen useaa aktuaattoria. Työn toimeksiantajana toimii BrainNova Oy. Työssä käsitellään tarvittavia komponentteja ohjauksen toteuttamiseen sekä logiikkaohjelmointia Siemens S7-1200 -logiikkayksiköllä.

Työssä esitellään komponentteja, jotka mahdollistavat QR-koodin luvun automaatioprosessissa. Tämän lisäksi esitellään vaihtoehtoja, joilla saadaan toteutettua QR-luku yhdessä Siemens -logiikkayksikön kanssa. Muita esiteltykomponentteja ovat aktuaattorien ohjaukseen tarvittavat komponentit, esimerkiksi H-silta ajuri.

Toiminnallisena osuutena opinnäytetyössä toimii Siemensin TIA Portal -ohjelmistolla tehty logiikkaohjelmointi, jonka avulla simuloitiin automaatioprosessin toimintaa. Logiikkaohjelmoinnilla rakennettiin tarvittavat ominaisuudet automaatioprosessin toimintaa ajatellen.

Opinnäytetyö toimii perustana automaatioprosessin kehitykselle ja siten helpottaa projektin mahdollista fyysisen toteutuksen tekemistä. Työtä voisi kehittää eteenpäin ottamalla siihen osaksi fyysiset kytkennät ja väylän kautta tehdyn tietoliikenne viestinnän.

Asiasanat: plc, aktuaattori, logiikkaohjelmointi, siemens

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Automotive Engineering
Intelligent Machines

NIEMI, ALEKSI:
Designing Common Controls for Automation Actuators.

Bachelor's thesis 37 pages, appendices 1 pages
May 2024

This thesis addressed the designing of common controls for multiple actuators. The goal of this thesis was to create a basis for a process where and object would be transferred from point A to point B, utilizing multiple linear actuators. This thesis commissioned by a client, BrainNova Oy. This thesis covered the necessary components for implementing controls and logic programming with the Siemens S7-1200 logic controller.

The thesis introduced components that enable QR code reading capabilities in the automation process. In addition, it presented options for implementing QR reading together with the Siemens logic controller. Other components that were introduced include those required for controlling the actuators, such as the H-Bridge driver.

The functional part of the thesis involved logic programming done with the Siemens TIA Portal software which was used to build the program itself and to simulate the operation of the automation process. The logic programming provided the necessary features for the operation of the automation process.

This thesis serves as a foundation for the development of the automation process and thus facilitates the potential physical implementation of the project. The work could be further developed by including physical connections and communication via a bus system.

Key words: plc, actuator, logic programming, siemens

SISÄLLYS

1	JOHDANTO	5
2	AKTUAATTORIEN TOIMINTA JA TAUSTAT	6
2.1	Lineaariaktuaattorit.....	6
2.2	Ohjausyksiköt.....	9
2.3	Delta ASDA Moottorien ajurit	11
2.3.1	QR-lukija ja Rasberry Pi	13
2.3.2	H-silta	15
3	Ohjauksen toteuttamiseen vaadittavat komponentit	16
3.1	Virransaanti	16
3.2	Kytkenät	17
4	Menetelmät	19
4.1	Lineaariaktuaattoreiden ja ohjausjärjestelmien valinta.	19
4.2	Ohjelmistojen ja laitteiden asennus, sekä konfigurointi	20
5	Toteutus	21
5.1	TIA Portal ohjelmointi	21
5.1.1	Konfigurointi	21
5.1.2	PLC TAGS.....	23
5.2	Ohjelman luominen	25
5.3	Testaus	33
6	PÄÄTELMÄT	35
	LÄHTEET	37

1 JOHDANTO

Useimmille tutuimmat automaatiotoimilaitteet ovat yksinkertaisimmillaan sähkömoottorit. Sähkömoottoreita käyttämällä voidaan valmistaa aktuaattoreita, joilla pystytään liikuttamaan massaa haluttuun suuntaan. Yhdistämällä useampia aktuaattoreita ja toimilaitteita, saadaan luotua automaatioprosessi.

Tässä opinnäytetyössä käsitellään erilaisten automaatiotoimilaitteiden yhteisen ohjauksen suunnittelua. Tarkoituksena on saada luotua yhtenäinen prosessi, jossa siirretään kappale paikasta A paikkaan B. Toimeksiantajana on BrainNova Oy, joka on teknologia alan startup-yritys. Osa opinnäytetyön sisällöstä on jätetty pois salassapitosopimuksen takia. Opinnäytetyössä ei käsitellä laitteiston mahdollisia käyttötarkoituksia, vaan todetaan prosessin toimivuutta suunnittelutasolla. Opinnäytetyön pohjalta voidaan jatkojalostaa teknologiateollisuuteen käytettävää automaatioprosessia.

Teollisia automaatioprosesseja on rakennettu loputtomasti, mutta lähes kaikki ovat yksilöityjä ratkaisuja, joita ei voi muissa prosesseissa hyödyntää. Siitä syystä tässäkin opinnäytetyössä käsitellään ohjauksen suunnittelua eri toimilaitteille, sillä vastaavaa prosessia ei ole tehty tai siitä ei ole julkisesti tietoa saatavilla.

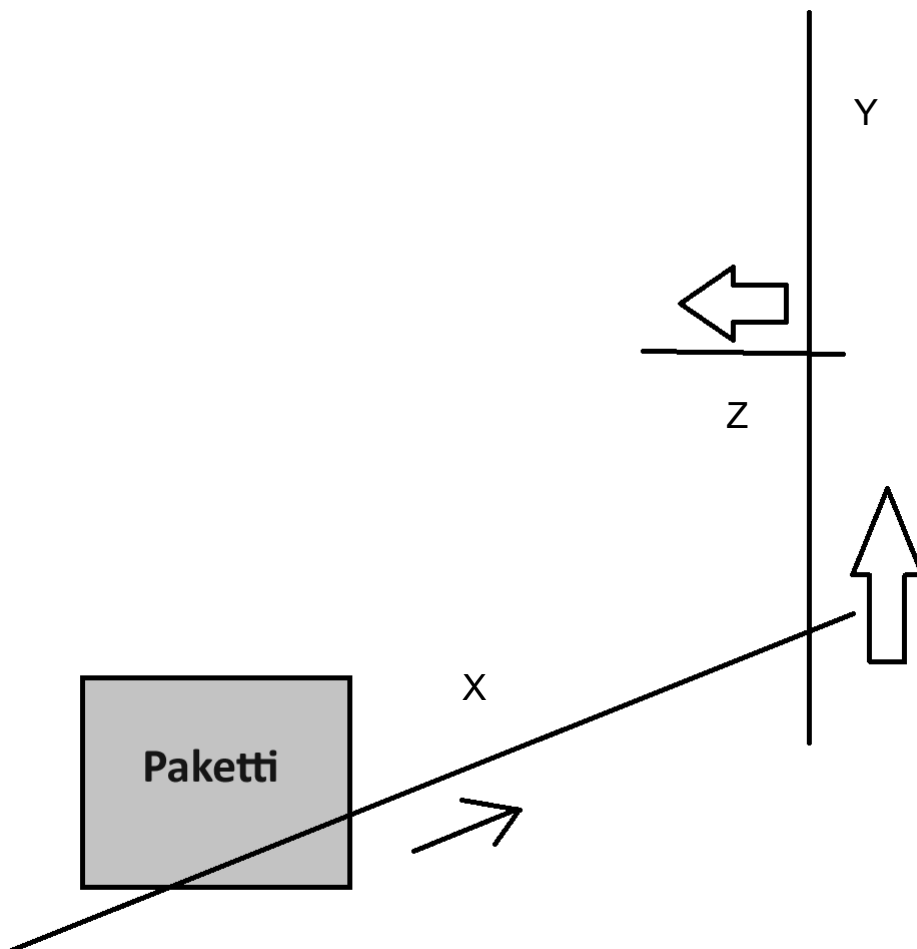
Työssä käsitellään, miten saadaan erilaiset automaatiotoimilaitteet keskustelemaan keskenään. Tätä varten tulee tietää tarvittavat kytkennät kaikille toimilaitteille, sekä niiden yhteensopivat kommunikaatioväylät. Ohjaus perustuu Siemensin S7-1200 sarjan logiikkayksikköön. Kytkentää varten tulee tehdä myös kytkentäkaavio.

Työ sisältää myös laitteiston ohjelmointia, jotta saadaan prosessi suoritettua halutulla tavalla. Ohjelmointi suoritetaan TIA Portal -ohjelmistolla.

2 AKTUAATTORIEN TOIMINTA JA TAUSTAT

2.1 Lineaariaktuaattorit

Tässä työssä käytetään kolmea erilaista lineaariaktuaattoria: ”Rod” tyyppin aktuaattori, ”Belt Driven motion guide” ja ”Rack and pinion motion guide”. Työn selkeyttämiseksi käytetään ”Rod” tyyppin aktuaattorista nimitystä RD aktuaattori, ”Belt Driven motion guide” on BD aktuaattori ja ”Rack and pinion motion guide” on RP aktuaattori. BD aktuaattori liikkuu X suunnassa, RP liikkuu Y suunnassa ja RD liikkuu suunnassa Z (KUVIO 1).



KUVIO 1. Paketin liikkeet hahmoteltuna (Kuvio: Aleks Niemi 2024).

”Rod” tyyppin aktuaattorissa” eli sauva-aktuaattorissa on yleensä sähkömoottori, joka on yhdistetty jonkinlaiseen ratastukseen (KUVA 1). Ratas pyörittää sauvaa, jossa on kierteet, joka puolestaan liikuttaa sauvaa ulos tai sisäänpäin. Useimmissa sauva-aktuaattoreissa on vain johdot virran syötölle, jolloin paikkatietoa ei voida saada aktuaattorista suoraan. Joissain malleissa on myös saatavilla paikkatietoanturointi. (Timotion, 2022.)



KUVA 1. Rod tyyppin aktuaattori (Kuva: Aleksi Niemi 2024).

Kytettäessä aktuaattorin positiivinen ja negatiivinen pää virran lähteeseen, alkaa aktuaattorin sähkömoottori pyörimään, jolloin sauva työntyy ulospäin koteloista. Jotta sauva voidaan vetää takaisin aktuaattoriin, tarvitsee johtojen polaarisuus kääntää. Tämän voi tehdä manuaalisesti vaihtamalla johdot tai voidaan käyttää H-silta moottorin ohjainta, jolla pystytään polaarisuus vaihtamaan. Sauvatyyppin aktuaattoria käytetään monessa eri käyttötarkoituksessa esimerkiksi robotiikassa, elintarviketeollisuudessa ja sähköpöydissä. Niiden ollessa yksinkertaisia ja edullisia, käytetään sauvatyyppin aktuaattoreita myös kotitekoisissa projekteissa. (Timotion, 2022.)

”Belt Driven motion guide” eli hihnäkäyttöinen aktuaattori toimii sauvan sijasta hihnalla (KUVA 2). Sähkömoottori pyörittää jälleen rattaita, jotka on yhdistetty suljetun kierron hihnaan. Hihnaan on osittain kiinnitetty kappale, joka liikkuu aktuaattorin iskupituuden verran. Hihnäkäyttöiset aktuaattorit ovat yleisesti kalliimpia, sillä ne ovat tehokkaampia, tarkempia ja nopeampia. Useimmat hihnäkäyttöiset

aktuaattorit ovat myös monimutkaisempia ja vaativat sisäänrakennetun anturoinnin. (Isotechnic, 2023.)

Tehokkaammat aktuaattorit tarvitsevat oman ohjausyksikkönsä eli driven. Drive eli ajuri ohjaa aktuaattorin virransyöttöä sekä kommunikoi aktuaattorin anturien kanssa. Ajuria tarvitsee myös ohjata ja siihen yleensä liitetään vielä ohjausjärjestelmä esim. Siemens logiikkayksikkö.



KUVA 2. Hihnakäyttöinen aktuaattori (Kuva: Aleksi Niemi 2024).

Näiden lisäksi on myös "Rack and pinion" eli hammaspyörällinen aktuaattori (KUVA 3). Siinä toimintaperiaatteena on sähkömoottorilla pyöritettävä hammaspyörä, jolla "kiivetään" hammaskiskoa ylös tai alaspäin. Tämä on kaikista kolmesta aktuaattorista suurikokoisin sekä suurimmalle kuormalle tarkoitettu. Tässä

projektissa sitä hyödynnetään pystysuuntaisessa liikkeessä. Kyseinen aktuaattori toimii samalla periaatteella kuin hihnakäyttöinen, eli sitä ohjataan sen oman ajurin kautta. (blog.ivesequipment n.d.)



KUVA 3. Hammaspyörällinen aktuaattori (Kuva: Aleks Niemi 2024).

2.2 Ohjausyksiköt

Toimilaitteiden kokonaisuutta ohjattaisiin yhden Siemens S7-1200 -ohjausyksikön avulla (KUVA 4). Siemensin ohjausyksikkö toimii logiikkaohjelmoinnilla. Logiikkaohjelmoinnissa käytetään logiikan soveltamista halutun toiminnan saavuttamiseksi. Ohjelmaan määritetään sääntöjä ja loogisia ehtoja, joiden pohjalta ohjelma suorittaa toimintoja.

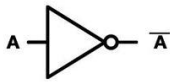
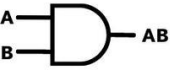
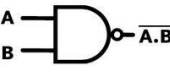
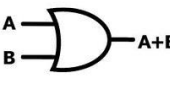
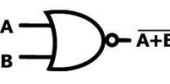
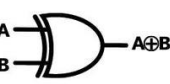
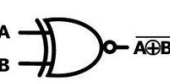


KUVA 4. Siemens S7-1200 PLC (<https://fi.rsdelivers.com/product/siemens/6es7211-1be40-0xb0/siemens-simatic-s7-1200-series-plc-cpu-for-use-s7/8624455>).

Siemensin S7-1200 -ohjausyksikköä voidaan ohjelmoida TIA Portal nimisen ohjelman avulla. Kyseessä on lohkopohjainen ohjelmointikieli. Ohjelmassa yksinkertaisimmillaan määritetään tulot ja lähdöt. Kun ohjelmaa käytetään oikeaoppisesti, vastaavat nämä tulot ja lähdöt logiikkayksikön fyysisiä kytkentöjä. (Virtusa, n.d.)

Logiikkaohjelmoinnissa käytetään lohkoja. Näiden avulla saadaan luotua erilaisia haluttuja toimintoja. TIA Portal sisältää useita eri moduuleja, mutta tässä työssä keskitytään STEP 7 moduulin. Se tarjoaa käyttäjälle graafisen ympäristön suorittaa ohjelmointia. Ohjelmointia voidaan tehdä eri ohjelmointi kielillä esimerkiksi Ladderilla. (Virtusa n.d.)

Esimerkiksi ”AND” portti syöttää positiivisen signaalin eteenpäin, kun sen molemmat sisääntulot ovat positiivisia. Jos niistä toinen on negatiivinen, on lähtö myös negatiivinen. Vaihtoehtoja porteista on useita ja jokaiseen käyttötarkoitukseen löytyy sopiva.

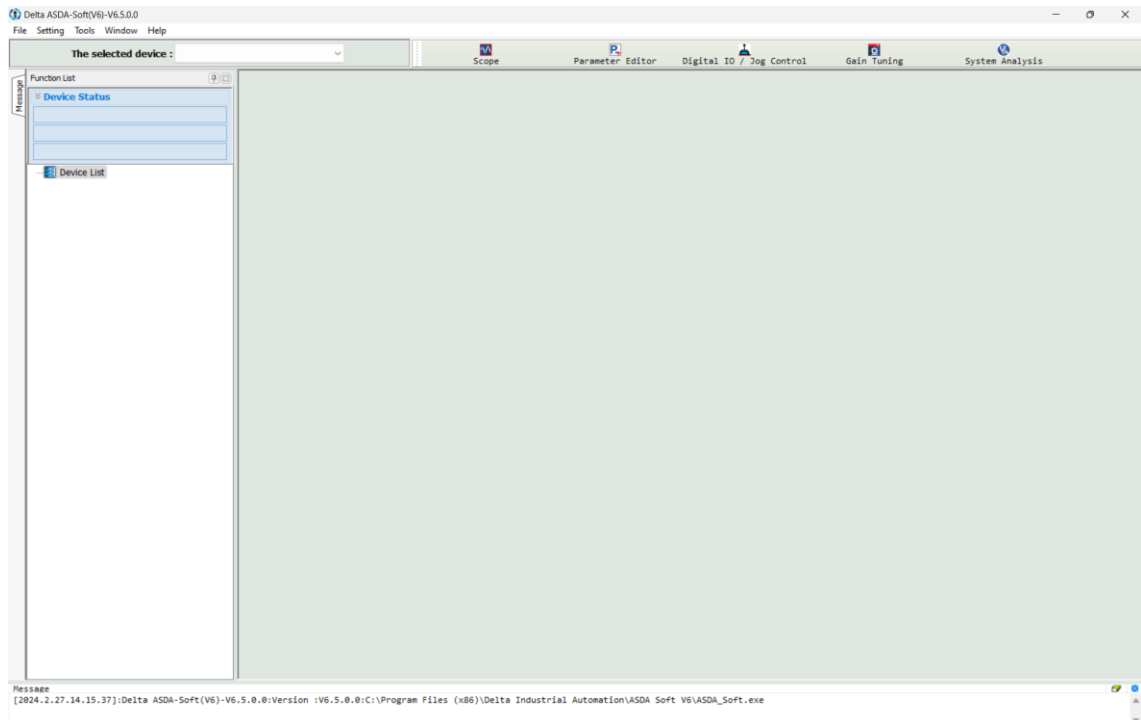
Name	Symbol & notation	Explanation
NOT		The inverter NOT simply accepts an input and outputs the opposite .
AND		All inputs must be positive (1) before the output is positive (1 or ON)
NAND <small>*Not AND</small>		Same as AND, but the outcome is the inverse (NOT) . So, perform AND first, then apply NOT to the output.
OR		At least one input must be positive (1) to give a positive output (1 or ON). All inputs could also be positive.
NOR <small>*Not OR</small>		Same as OR, but the outcome is the inverse (NOT) . So, perform OR first, then apply NOT to the output.
XOR <small>*Exclusive OR</small>		Only one input can be positive (1) to give a positive output (1 or ON). If both are positive, the output is negative (0 or OFF)
XNOR <small>*Exclusive Not OR</small>		All inputs must be the same (either high or low) for a positive output (1). Otherwise, the output is negative (0 or OFF)

KUVA 5. Logiikkaportit (<https://computerengineeringforbabies.com/en-eu/blogs/engineering/logic-gate>).

TIA Portal ohjelma sisältää myös muita hyödyllisiä työkaluja ohjelmointiin esimerkiksi ohjelman simuloimiseen, diagnostiikkaa ja erilaisia vianmäärityksen työkaluja.

2.3 Delta ASDA Moottorien ajurit

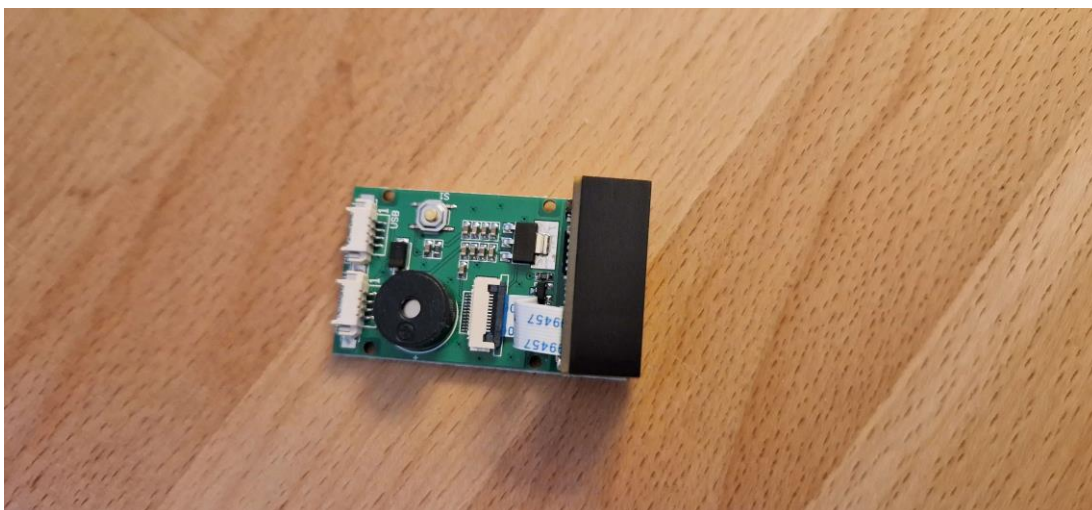
Vaikka suurin osa toimintojen ohjauksesta tapahtuu Siemensin logiikkayksiköllä, on moottorien ohjausta varten niille omat ajurit (KUVA 6). Moottorit sisältävät myös omat anturit ja näiden avulla saadaan ajurilla suoraan ohjelmoitua moottorille tapahtumaketju. Siemensin logiikkayksikön toimiessa pääkeskuksena, josta



KUVA 7. Delta ASDA-Soft ohjelmiston käyttöliittymä (Kuva: Aleksi Niemi 2024).

2.3.1 QR-lukija ja Raspberry Pi

Osana automaatioprosessia on QR-koodin lukija. Lukijana toimii Grow GM67 (KUVA 8). Tällä halutaan lukea QR-koodi, jonka ollessa vastaava kuin ennalta määritetty koodi, antaisi se signaalin logiikkayksikölle jatkaa prosessia. QR-koodin lukija tuottaa TTL-signaalia, joka tarvitsee saada luettavaan muotoon Logiikalle. QR-lukijan data voidaan välittää helposti USB:n välityksellä. Vaihtoehtoisesti voidaan käyttää logiikan ja QR-lukijan välissä esimerkiksi Raspberry Pi:tä, jotta saadaan data luettua ja lähetettyä signaali eteenpäin logiikalle.



KUVA 8. Grow GM67 QR-lukija (Kuva: Aleks Niemi 2024).

Raspberry Pi on yhden piirilevyn tietokone, jota käytetään paljon erilaisissa projekteissa kotiautomaatiosta, ohjelmointiin ja robotiikkaan (KUVA 9). Sen ominaisuudet ovat oma linux-pohjainen käyttöjärjestelmä ja lukuisat kytkentämahdollisuudet. (Raspberrypi, 2024.)

Raspberry Pi sisältää useamman USB portin, jolloin QR-lukija saadaan kytkettyä suoraan siihen. Raspberry Pi:llä luotaisiin koodi, jossa QR-koodit tarkastetaan ja koodin ollessa oikea, lähetetään signaali eteenpäin logiikkayksikölle. (Raspberrypi, 2024.)



KUVA 9. Raspberry Pi 5 (verkkokauppa.com).

2.3.2 H-silta

RD lineeariaktuaattori sisältää ainoastaan johdot virransyötölle, jolloin jos suuntaa halutaan vaihtaa, tulee johtojen polariteetti vaihtaa. Tätä varten käytetään H-silta nimistä ajuria (KUVA 10), jolla voidaan polariteetti vaihtaa ilman fyysisten johtojen vaihtamista. Ajurin malli on EM-241C DC-MOTOR CONTROLLER 12-24V 15A. H-siltaa ohjattaisiin Siemensin logiikalla.



KUVA 10. H-silta ajuri (electromen.com).

3 Ohjauksen toteuttamiseen vaadittavat komponentit

3.1 Virransaanti

BD ja RP aktuaattorit ovat tehoiltaan 400 W ja 750 W. Ne toimivat myös 230 V verkkojännitteellä. Kyseisten aktuaattorien moottoreille saadaan virrat suoraan niiden omilta ajureilta. Ajurit ovat kytkettynä verkkovirtaan, jota varten tarvitaan liitäntäjohtoa.

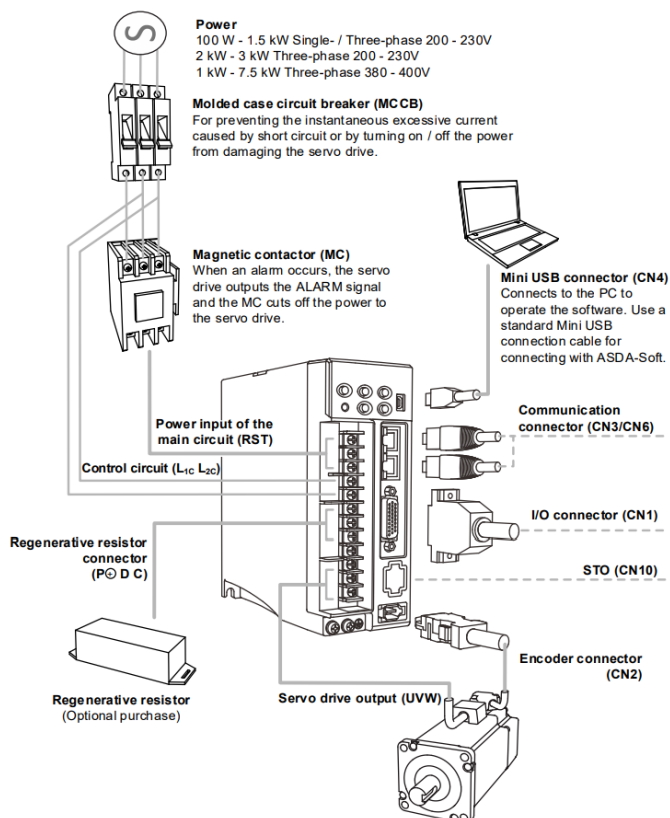
Verkkovirtaa ei voida kuitenkaan suoraan kytkeä ajureille vaan tarvitaan releet (KUVA 11), joiden kautta virtaa voidaan syöttää ajureille. Tarvitaan johdonsuojakatkaisija, kuvasta katsottuna ”Molded case circuit breaker”, jonka lisäksi tarvitaan magneettinen kontaktori eli ”Magnetic Contactor”. Sillä pystytään ohjaamaan virran päälle ja pois kytkentää. Molemmat releistä olisivat 10A nimellisvirralla.

Myös muut komponentit tarvitsevat erilliset virtalähteet. Siemens -logiikkayksikölle tarvitaan 24 V 200 W virtalähde, joiden perusteella valittiin käyttöön ”MEAN WELL LRS-200-24 - Hakkurivirtalähde 211W 24V 8.8A”.

H-silta tarvitsee erillisen virransyötön, sillä logiikasta ei saada tarpeeksi virtaa ulos, jotta sillä voitaisiin ohjata suoraan RD aktuaattoria. Sitä varten tarvitaan vielä yksi erillinen virtalähde, joka on ”MEAN WELL LRS-100-12 - Hakkurivirtalähde 102W 12V 8.5A”.

Logiikkayksikkö syöttää virran antureille.

3.1.1 Connecting to peripheral devices (connecting to Delta communication type servo motor)



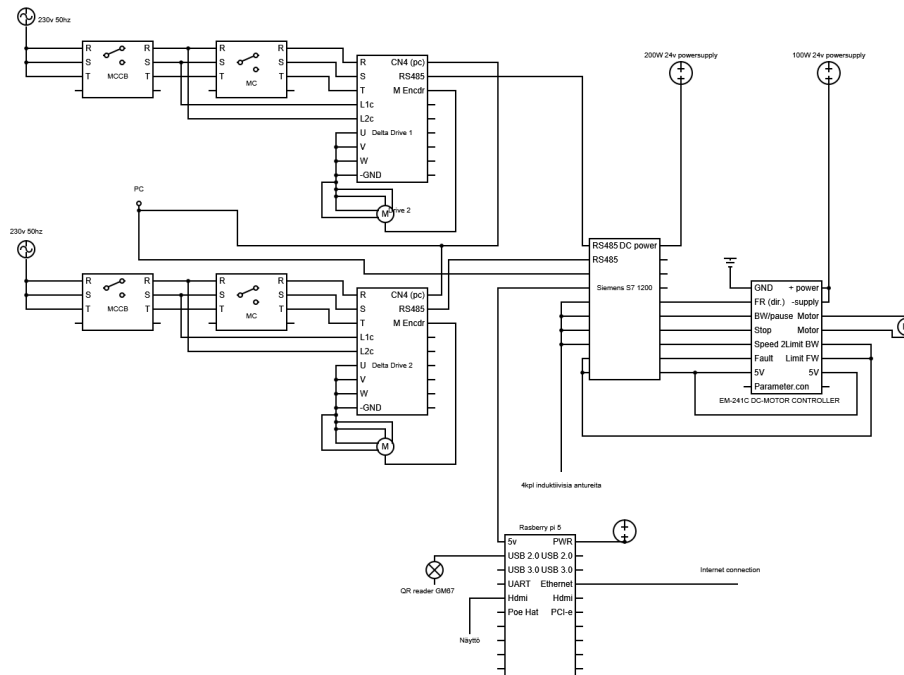
KUVA 11. Delta ajurien kytkentä (Kuva: Aleks Niemi 2024).

3.2 Kytkennät

Alla nähdään kuva kytkentäkaaviosta. Vasemmalta aloittaessa ensimmäisenä tulee moottorien ajurien kytkennät vastaan. Ne on kytketty suoraan verkkovirtaa, jonka jälkeen on kytketty 10 ampeerin johdonsuojakatkaisijat MCCB ja 10 ampeerin magneettiset kontaktorit MC. Moottorien ajurit tarvitsevat yhteensä viiden johdon kytkennät virransaannille. Moottorien ajurit ovat kuvassa Delta Drive nimellä. Niiden moottorit on merkitty palloilla, joissa on sisällä M kirjain.

Moottorien ajureilta on yhteys tietokoneeseen, sekä Siemens logiikka yksikköön. Ajureita ohjataan väylän avulla, jolloin ei tarvita suuria määriä kytkentöjä PLC:n puolelta. Logiikkayksiköllä on oma virtalähde, joka on merkattu kuvaan 200W 24V Powersupply nimellä. Eniten kytkentöjä vaatii H-silta moottoriajuri, johon tulee kuusi johtoa logiikalta. Ajurilla on myös oma virtalähde, sekä moottori, jotka on merkattu kuvaan.

Viimeisenä komponenttina piirikaaviosta löytyy Raspberry Pi 5 yhden piirilevyntietokone. Se sisältää oman virtalähteensä ja ethernet -yhteyden. QR-lukija on merkitty lampun symbolilla kuvaan. Raspberry lähettää yhden signaalin eteenpäin loogikalle johtoa pitkin.



KUVA 12. Projektin kytkentäkaavio (Kuva: Aleks Niemi 2024).

Kytkentöihin käytettäisiin 18 AWG johtoa ja 14 AWG johtoa. Muut johdot hankitaan valmiina, jolloin niiden paksuuksia ei tarvitse valita.

4 Menetelmät

4.1 Lineaariaktuaattoreiden ja ohjausjärjestelmien valinta.

Tämän työn toimilaitteet valikoituivat niiden toimintojen, käytettävyyden, sekä hinnan perusteella. Suuremmilta aktuaattoreilta tarvittiin kyky liikuttaa noin 30 kg ja 50 kg kuormia ilman ongelmia.

Hihnavetoinen aktuaattori on mitoitettu vetämään liikuttamaan 30 kilon kuormaa vaakatasossa nopeudella 1 m/s. Oli myös tarpeellista, että aktuaattori on tarpeeksi pitkä. BD aktuaattori on noin 2,3 metriä pituudeltaan. Aktuaattorit sisältää kolme omaa anturia, joita hyödynnetään projektissa. Molemmat aktuaattorit sisäl-sivät omat ajurinsa, jotka helpottavat niiden ohjausta.

Isommalta RP aktuaattorilta haluttiin kyky nostaa 50 kg pystysuunnassa nopeudella 1 m/s. Kyseinen aktuaattori on noin 1,8 metriä korkea. Sen ohjaus toimii samalla periaatteella kuin hihnavetoisen ja molemmat aktuaattorit ovat samalta toimittajalta.

RD aktuaattorilta vaadittiin iskupituudeksi 300 millimetriä, sekä sen piti olla kompakti myös muissa mitoissa. Tämän lisäksi tarvittiin suuri työntövoima. Valitun aktuaattorin työntövoima on 800 Newtonia ja nopeus on 28 mm/s. Hinta määrittä aktuaattorin muut ominaisuudet, jolloin päädyttiin yksinkertaiseen kaksi johtoi-seen ratkaisuun ilman paikkatietoa.

Ohjauslaitteet valikoituivat saatavuuden ja käytettävyyden perusteella. Siemens -logiikkayksikkö saatiin oppilaitoksen kautta projektiin lainaan. Siemensin logii-koita on myös käytetty opintojen eri vaiheissa logiikkaohjelmoinnin yhteydessä. Tämän takia oli helppo valita se ohjausjärjestelmäksi.

Muihin ohjausjärjestelmiin lasketaan Raspberry Pi. Se valikoitui sen saatavuuden, alhaisen hinnan ja helpon käytettävyyden takia. Raspberry Pi on todella suosittu alusta erilaisille projekteille, jolloin sen käyttöön löytyy paljon monipuolista ohjeis-tusta.

4.2 Ohjelmistojen ja laitteiden asennus, sekä konfigurointi

Tämän projektin ohjauksen toteuttamiseksi tarvittiin kolmen eri ohjelmiston asennusta. Pääohjelmistona toimii Siemensin TIA Portal -ohjelmisto, jolla logiikkayksikköä ohjelmoidaan ja valvotaan. Se saadaan ladattua Siemensin verkkosivuilta. Sen käyttöön tarvitaan lisenssi, joka tässä tapauksessa saadaan koulun puolesta.

Toinen ohjelmisto on Deltan ajurien ohjausta varten. Delta ASDA Soft saadaan ladattua ilmaiseksi Deltan sivuilta. Sen käyttöön ei tarvita erillistä lisenssiä. Ajurien ohjaus ja ohjelmointi tapahtuu USB:n välityksellä.

Kolmantena ohjelmistona toimii Raspberry Pi:n oma käyttöliittymä, johon on sisäänrakennetut ohjelmointityökalut. Se saadaan ladattua Raspberry Pi:n nettisivuilta ja tulee asentaa esimerkiksi muistikortilta laitteelle. Raspberry Pi:n ollessa oma tietokoneensa, ei sen ohjelmointiin tarvita erillistä tietokonetta. Raspberry Pi:hin kytketään näppäimistö, hiiri ja näyttö sen inputeista ja sitä voidaan käyttää normaalin tietokoneen lailla. Sen ohjelmointi tapahtuu Python ohjelmointikielellä.

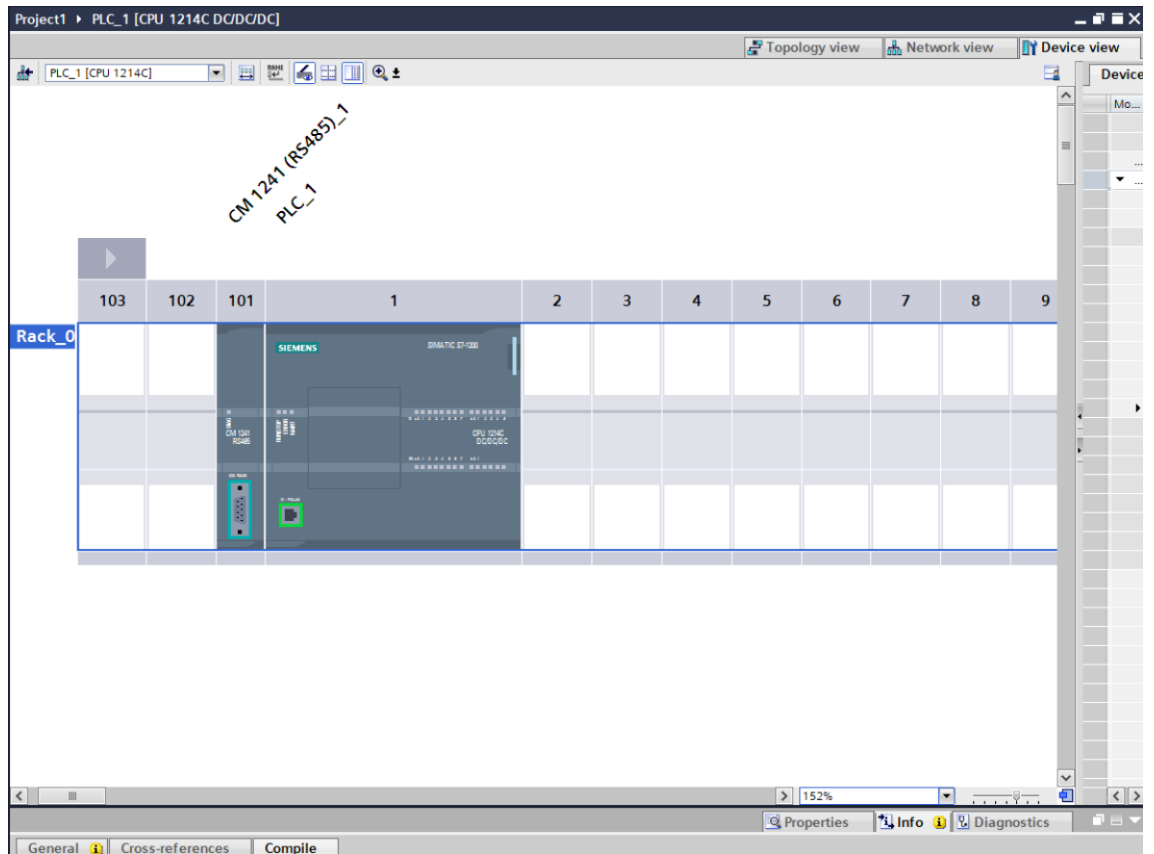
5 Toteutus

5.1 TIA Portal ohjelmointi

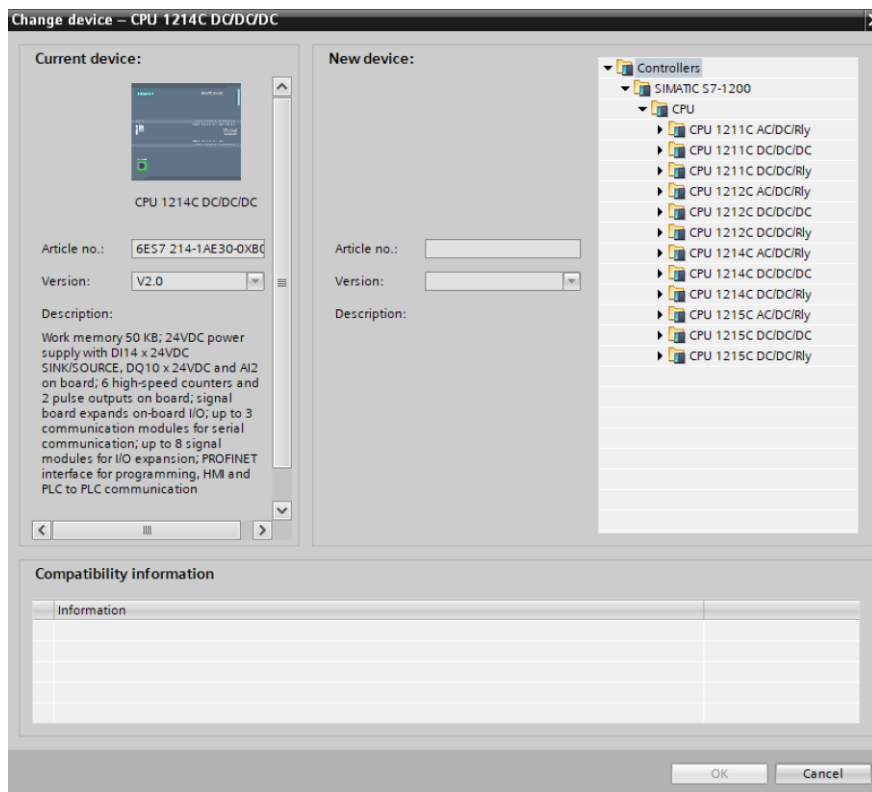
5.1.1 Konfigurointi

Tämän työn toteutus perustuu ohjelmoinnin suunnitteluun. Tällä tarkoitetaan ohjelman luomista, jolla teoreettisesti pystyttäisiin ohjaamaan aiemmin mainittua laitteistoa.

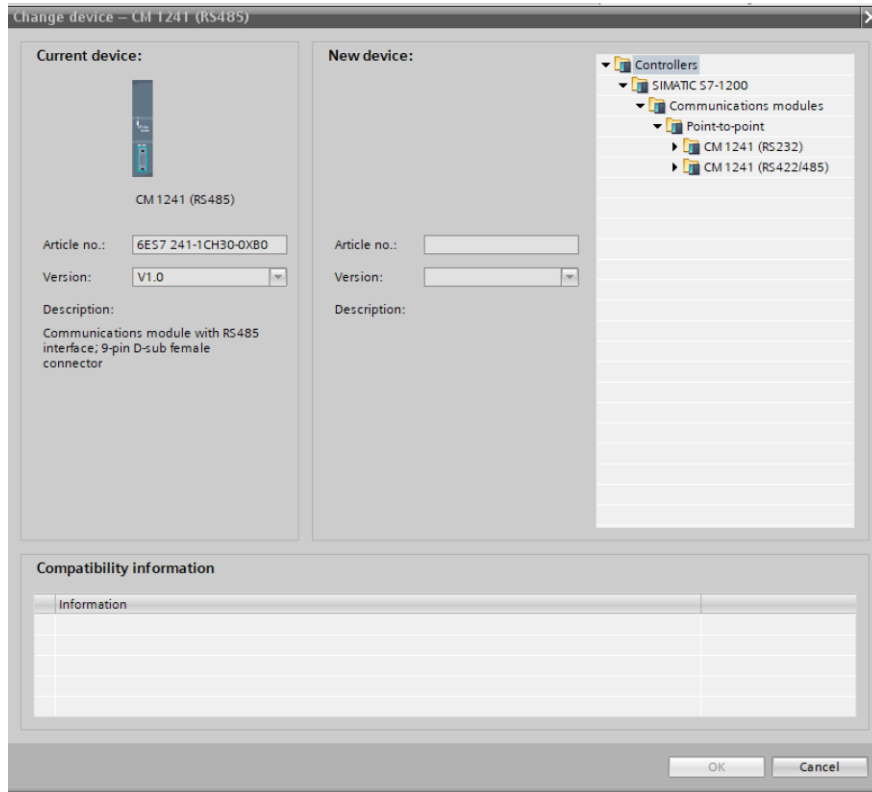
Ohjelmointi aloitetaan valitsemalla oikeat komponentit ohjelmointiympäristöön (KUVA 13). Tässä tapauksessa valittiin PLC:ksi S7 1200 CPU 1214 C DC/DC/DC (KUVA 14). Valinta tulee suorittaa käytettävän laitteiston pohjalta. PLC:n lisäksi lisätään väyläyksiköksi moduuli CM 1241 (RS-485) (KUVA 17). Tätä tarvitaan aktuaattorien ajurien välisessä tiedonsiirrossa. Kun valitaan oikeat komponentit ohjelmaan, vastaavat ne asentaessa laitteiston kytkentöjä, jolloin asennus on tehokasta.



KUVA 13. TIA Portal device view (Kuva: Aleksii Niemi 2024).



KUVA 14. TIA Portal PLC:n valinta (Kuva: Aleksii Niemi 2024).



KUVA 15. TIA Portal väylämoduulin valinta (Kuva: Aleksi Niemi 2024).

5.1.2 PLC TAGS

Kun komponentit on määritetty, tulee sisään- ja ulostulot määrittää ohjelmaan eli tagit. Se onnistuu PLC Tags -sivulta. Sivulta aukeaa ruudukko, johon merkataan laitteiston kytkennät. Sisääntulot merkataan I0.0 alkuisesti ja ulostulot Q1.0 alkuisesti. I0.0 ja Q1.0 ovat osoitteita, jotka vastaavat PLC:ssä fyysistä porttia. Portille annetaan nimi esimerkiksi B1, S3 tai Q2 ja kommenttikenttään on hyvä selostaa, mitä kyseinen portti ohjaa. Data tyyppinä käytetään Booleania, jossa käyttää true ja false arvoja.

Package simulation ▶ PLC_1 [CPU 1512C-1 PN] ▶ PLC tags ▶ Sensors and actuators [45]

Sensors and actuators									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervis...	Comment
1	B1	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Rasberry signal to start process
2	B2	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor checking that a Package is present
3	S1	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Start intake button
4	S2	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Stop button
5	B4	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor confirms position of package on pull-in.
6	B5	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor stopping the scissor mechanism pushing under the scoot
7	B6	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor stopping the pushing movement of the scissor actuator, when
8	B7	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor stopping compacting of scissor mechanism
9	Q1	Bool	%Q1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Output to H-Bridge controller
10	Q2	Bool	%Q1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		FW H-Bridge
11	Q3	Bool	%Q1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		REV H-Bridge
12	Q4	Bool	%Q1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Stop H-Bridge
13	Q5	Bool	%Q1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Low Speed H-Bridge
14	Q6	Bool	%Q1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Fault H-Bridge
15	Q7	Bool	%Q1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to pull-in actuator for position 2
16	Q8	Bool	%Q1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to lift motor Drive for position 2
17	B8	Bool	%I2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from pull-in motor Drive that position 2 is complete
18	B9	Bool	%I2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from lift motor Drive that position 2 is complete
19	Q9	Bool	%Q2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to pull-in actuator for position 3
20	B10	Bool	%I2.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from pull-in actuator that position 3 is complete
21	Q10	Bool	%Q2.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to lift motor drive to lift
22	B11	Bool	%I2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from lift motor drive that the lift is complete
23	B12	Bool	%I2.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Position sensor 1
24	B13	Bool	%I2.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Positions sensor 2
25	B14	Bool	%I3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Positions sensor 3
26	B15	Bool	%I3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Positions sensor 4
27	B16	Bool	%I3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Positions sensor 5
28	Q11	Bool	%Q2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Position sensor 1 OUTPUT
29	Q12	Bool	%Q2.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Position sensor 2 OUTPUT
30	Q13	Bool	%Q2.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Position sensor 3 OUTPUT
31	Q14	Bool	%Q3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Position sensor 4 OUTPUT
32	Q15	Bool	%Q3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Position sensor 5 OUTPUT
33	Q16	Bool	%Q3.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to lift motor drive to lower
34	B17	Bool	%I3.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from lift motor drive that package has been placed
35	B18	Bool	%I3.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor that tells when the scissor mechanism is pulled together
35	B18	Bool	%I3.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor that tells when the scissor mechanism is pulled together
36	Q17	Bool	%Q3.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to lift motor to go to normal position
37	B19	Bool	%I3.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from lift motor that its in normal pos.
38	S3	Bool	%I3.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Start scoot output
39	Q18	Bool	%Q4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to start package pickup procedure
40	Q19	Bool	%Q4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to lift motor to lift package from rack
41	B20	Bool	%I4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from lift motor that package has been lifted from rack
42	Q20	Bool	%Q4.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal to pull-in actuator for position 1
43	B21	Bool	%I4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signal from pull-in actuator that position 1 is confirmed
44	Q21	Bool	%Q4.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Light that package is ready for pick-up
45	S4	Bool	%I4.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button confirming pickup
46	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

KUVA 16. PLC tags Anturit ja aktuaattorit (Kuva: Aleksi Niemi 2024).

Kuvasta 16 nähdään kaikki projektissa käytettävät liitännät. Osa sisään- ja ulostuloista käyttäisi RS485 väylää, mutta tässä suunnitteluvaiheessa, väylän viestinnät on merkattu omina tuloina. Ohjelma sisältää neljä fyysistä painiketta ja 11 anturia. Ohjelmaan on lisätty myös valo merkitsemään prosessin tilaa. Suurin osa ulostuloista olisi väyläviestintää, mutta yhtä aktuaattoria ohjataan H-sillan kautta, jolloin sitä ohjataan logiikkayksikön analog signaaleilla. Kyseiset liitännät ovat Q1.1-Q1.5.

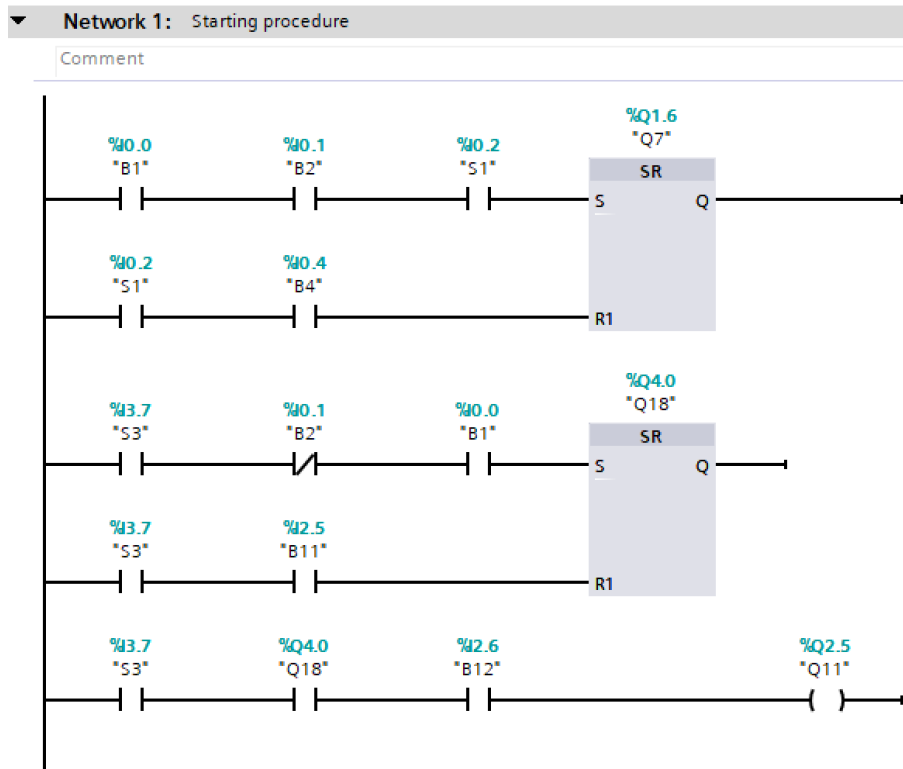
5.2 Ohjelman luominen

Ohjelmointi on jaettu eri Networkeille, eli verkoille, mitkä toteuttavat aina tietyt toiminnot ohjelmassa. Ohjelmointi aloitettiin luomalla aloituskäytäntö, jossa valitaan, halutaanko noutaa paketti vai palauttaa se. Painetaan painiketta S1, jos paketti halutaan palauttaa ja noutaessa painetaan painiketta S3. Molemmat napit jäisivät päälle ohjelman suorituksen ajaksi, jolloin voidaan varmistaa, etteivät ohjelman eri toiminnot aktivoidu samanaikaisesti.

Kun palautus aloitetaan (KUVA 17) ja painetaan painiketta S1, asetetaan paketti laitteeseen. Laite lukee paketin QR-koodin, joka käsitellään Raspberry Pi:llä. Koodin ollessa oikea, lähettää Raspberry Pi signaalin B1 on tosi. Anturi B2 tunnistaa myös paketin olemassaolon telineessä. Kun nämä kolme ehtoa toteutuvat, aloitetaan sisäänottoprosessi.

Q7 ulostulo aktivoituu ja lähettää signaalin hihnakäyttöiselle aktuaattorille siirtyä positioon kaksi. Anturi B4 nolaa SR-kiikun, kun se havaitsee paketin positiossa kaksi. SR-kiikku eli set-reset toiminto jää päälle aktivoituessaan, kunnes se nolataan reset toiminnolla.

Valittaessa nouto, painetaan S3 painiketta, jolloin ehtoina aloitukseen kielteinen signaali anturilta B2, sekä signaali Raspberry Pi:ltä. Tällöin Q18 lähtö aktivoituu, jolloin ohjelma alkaa etsimään, missä viidestä paikasta on paketti saatavilla.



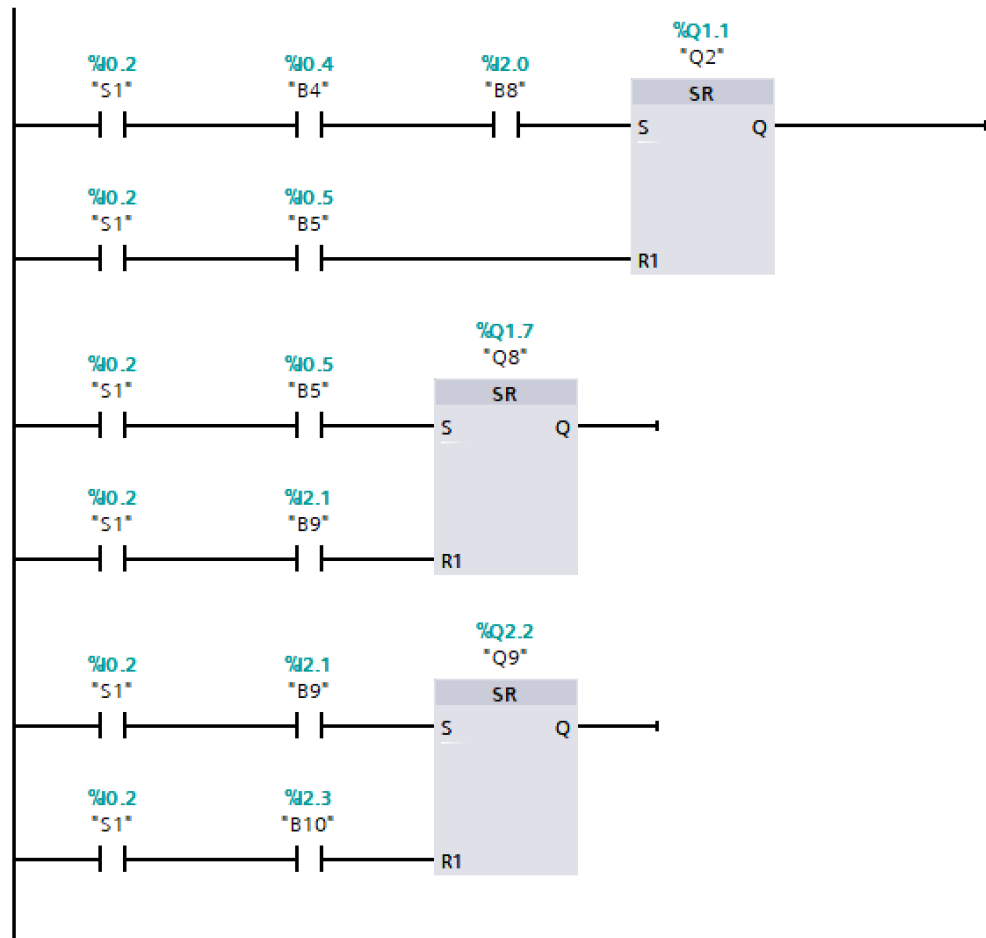
KUVA 17. Prosessin aloitus (Kuva: Aleksi Niemi 2024).

Seuraavaksi sisäänottoprosessissa (KUVA 18) hihnäkäyttöisen aktuaattorin ollessa positiossa kaksi, viestittää aktuaattorin ajuri (B8) PLC yksikölle siirron onnistuneen. Tämän varmistaa anturi B4. Tämä aktivoi Q2 tulon, joka on Forward signaali H-sillalle ja sen aktuaattorille. Aktuaattori työntyy paketin alle, kunnes anturi B5 lopettaa työntön. B5 anturin aktivoitua, nostetaan paketti hihnäkäyttöisen aktuaattorin telineestä ilmaan hammaspyörällisellä aktuaattorilla (Q8). Kun nosto on suoritettu, lähettää aktuaattori signaalin B9, jolloin siirretään hihnäkäyttöinen aktuaattori positioon kolme. Tämä tehdään siitä syystä, ettei kyseinen toimilaite ole noston tiellä.

Kun hihnäkäyttöisen aktuaattorin teline on siirretty pois alta (B10), voidaan hammaspyörällisellä aktuaattorilla nostaa paketti haluttuun tasoon.

▼ **Network 3:** Stopping pull-in and lifting the package

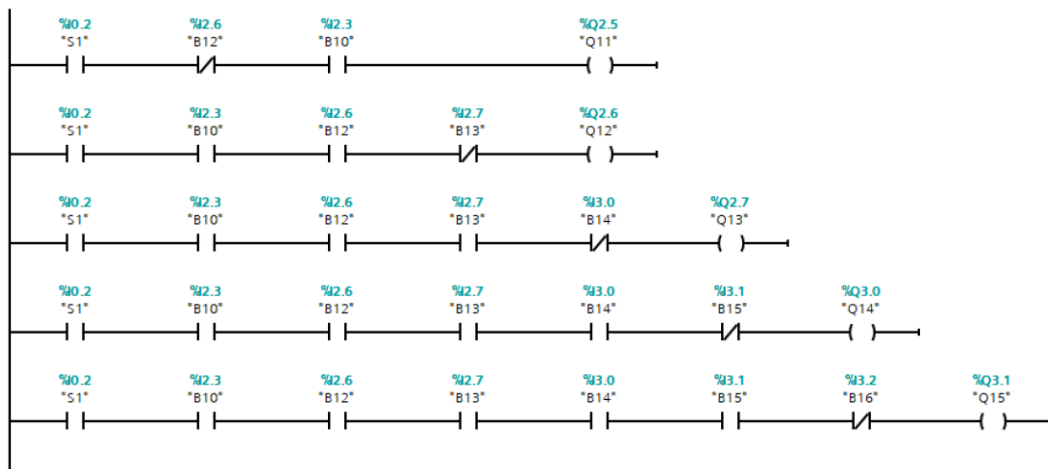
Comment



KUVA 18. Paketin sisäänotto ja telineeseen nosto. (Kuva: Aleks Niemi 2024).

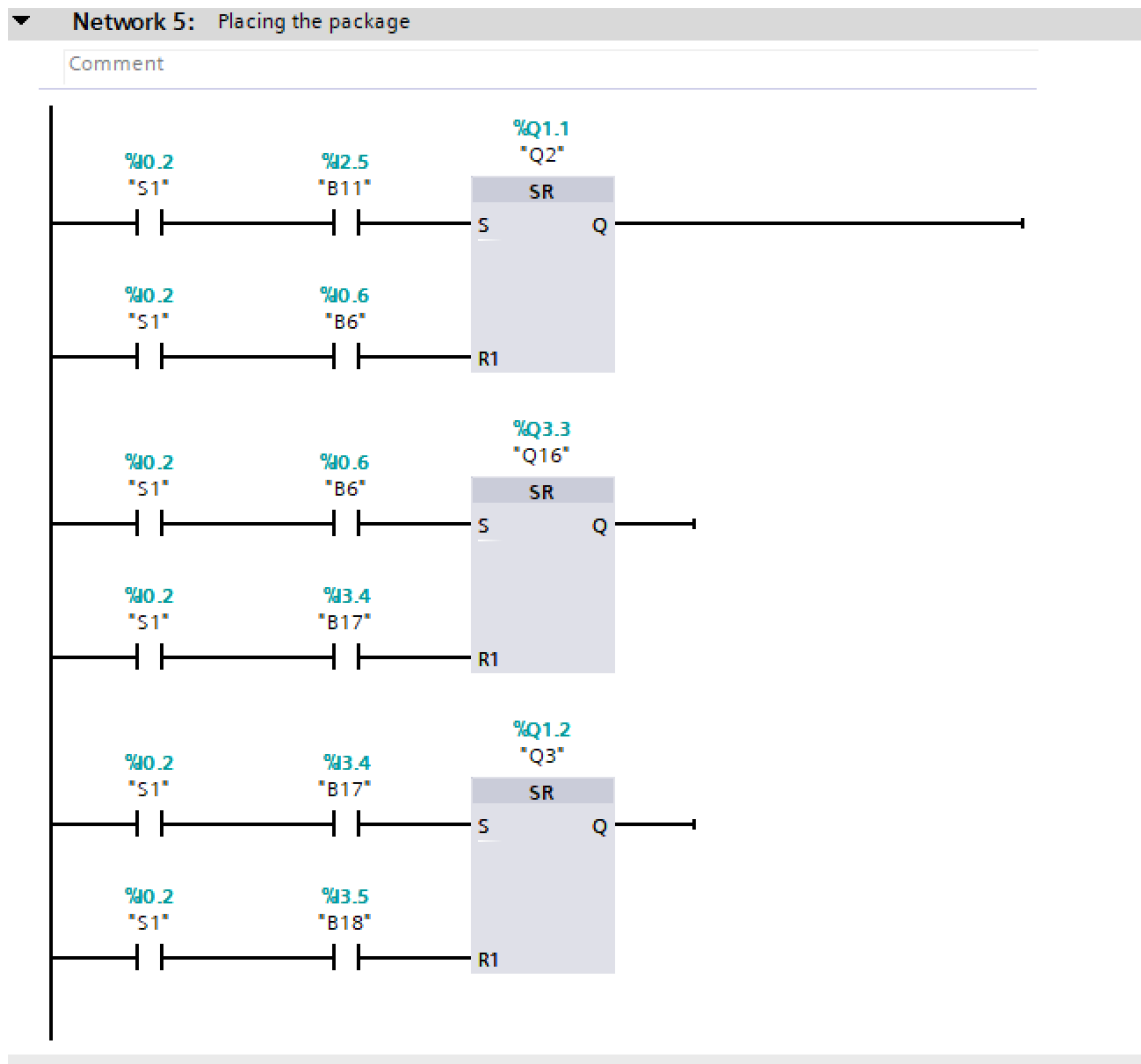
▼ **Network 4:** Lifting the package to the storage

Comment



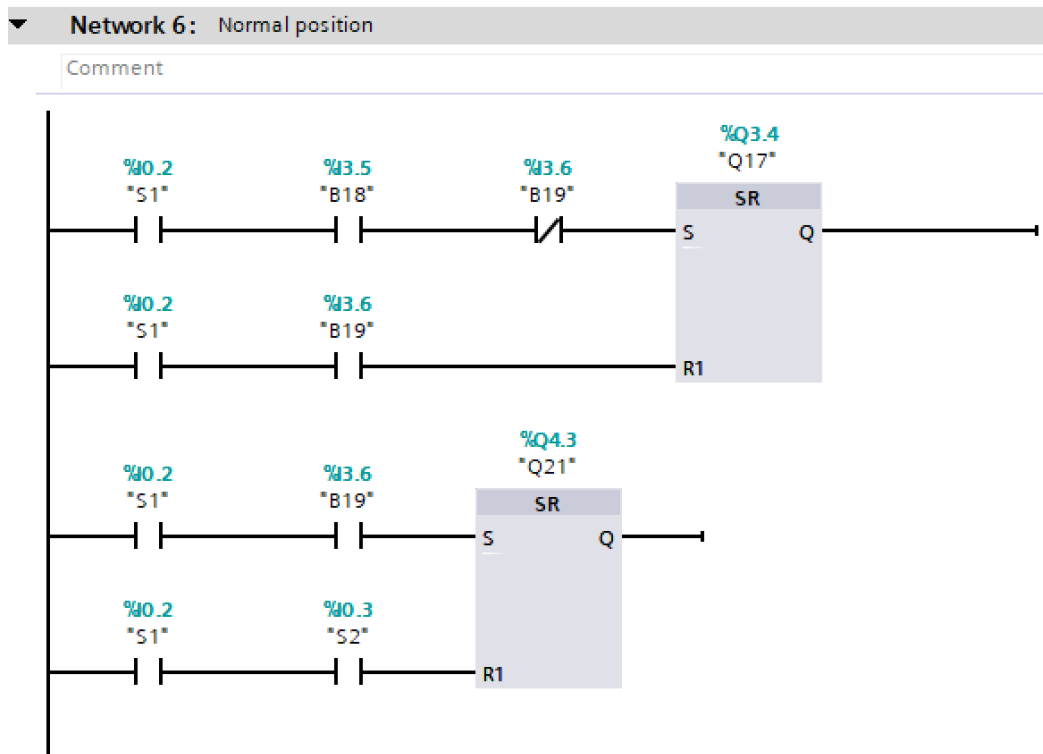
KUVA 19. Telineen paikan valikoituimen (Kuva: Aleks Niemi 2024).

Telineessä, johon paketti asetetaan, on viisi eri tasoa (KUVA 19). Jokaisella tasolla on anturi (B12-B16), joka havaitsee, onko paikalla pakettia. Ohjelma on toteutettu siten, että käytetään aina alimmaista paikkaa telineessä, jos se on tyhjänä. Anturien lähettäessä signaalin pakettin ollessa paikallaan, on ohjelmassa käännetty niiden signaali. Tällöin telineen ollessa tyhjä lähtee signaali hammaspyörälliselle aktuaattorille nostaa paketti tasoon, joka Q11 ilmoittaa.



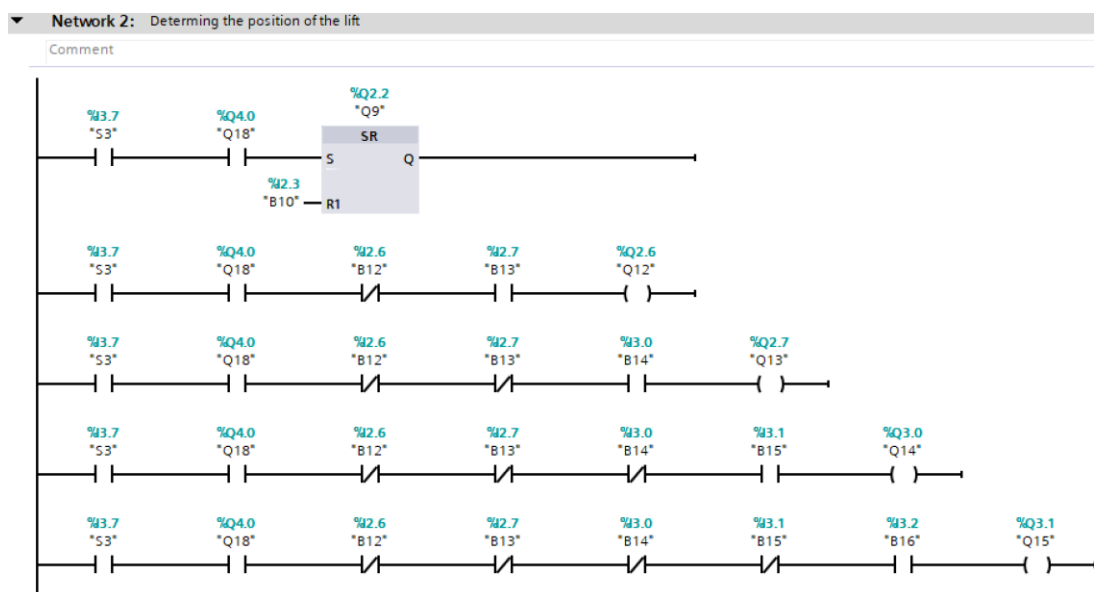
KUVA 20. Pakettin asettaminen telineeseen (Kuva: Aleks Niemi 2024).

B11 signaali aktuaattorin ajurilta ilmoittaa noston onnistuneen (KUVA 20). Pakettin asettaminen telineeseen aloitetaan työntämällä pakettia telinettä kohti (Q2). Työnnön pysäyttää anturi B6. Sitten hammaspyörällistä aktuaattoria lasketaan, jolloin paketti asettuu telineeseen (Q16). B17 on signaali pakettin laskemisesta telineeseen aktuaattorin ajurilta. Sen jälkeen Q3 aktivoituu, joka on Reverse signaali H-sillalle. Eli sillä vedetään aktuaattori kasaan. Tämän pysäyttää anturi B18.



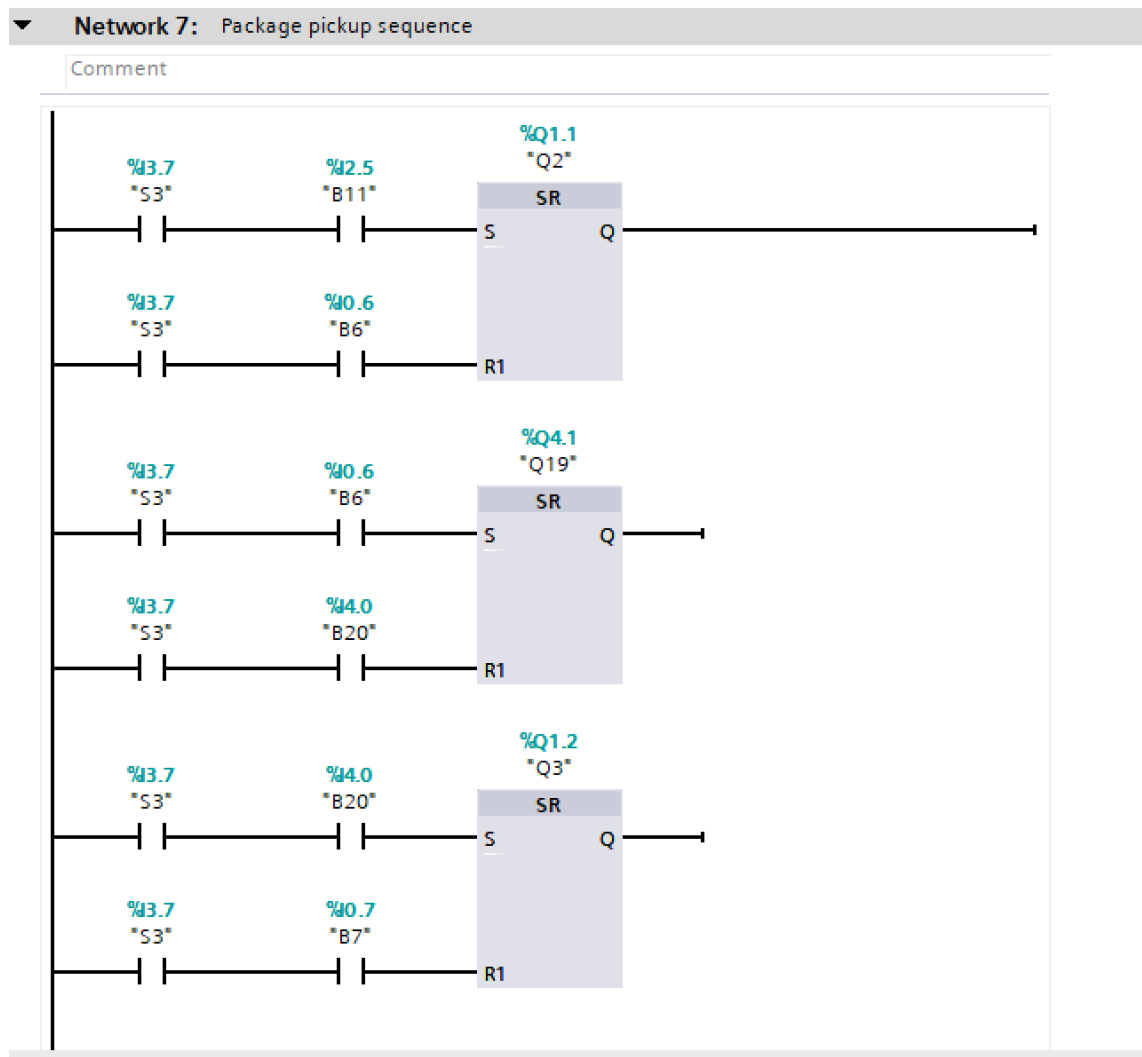
KUVA 21. Järjestelmän normaali positio (Kuva: Aleksi Niemi 2024).

Kun paketti on telineessä, palaa järjestelmä normaaliin positioon (KUVA 21). Hammaskäyttöinen aktuaattori ajetaan alas (Q17), hinnakäyttöinen aktuaattori on positiossa kolme ja työntävä aktuaattori on kasassa. Kun paketti on kasassa, syttyy valo Q21, joka kuitataan painikkeella S2.



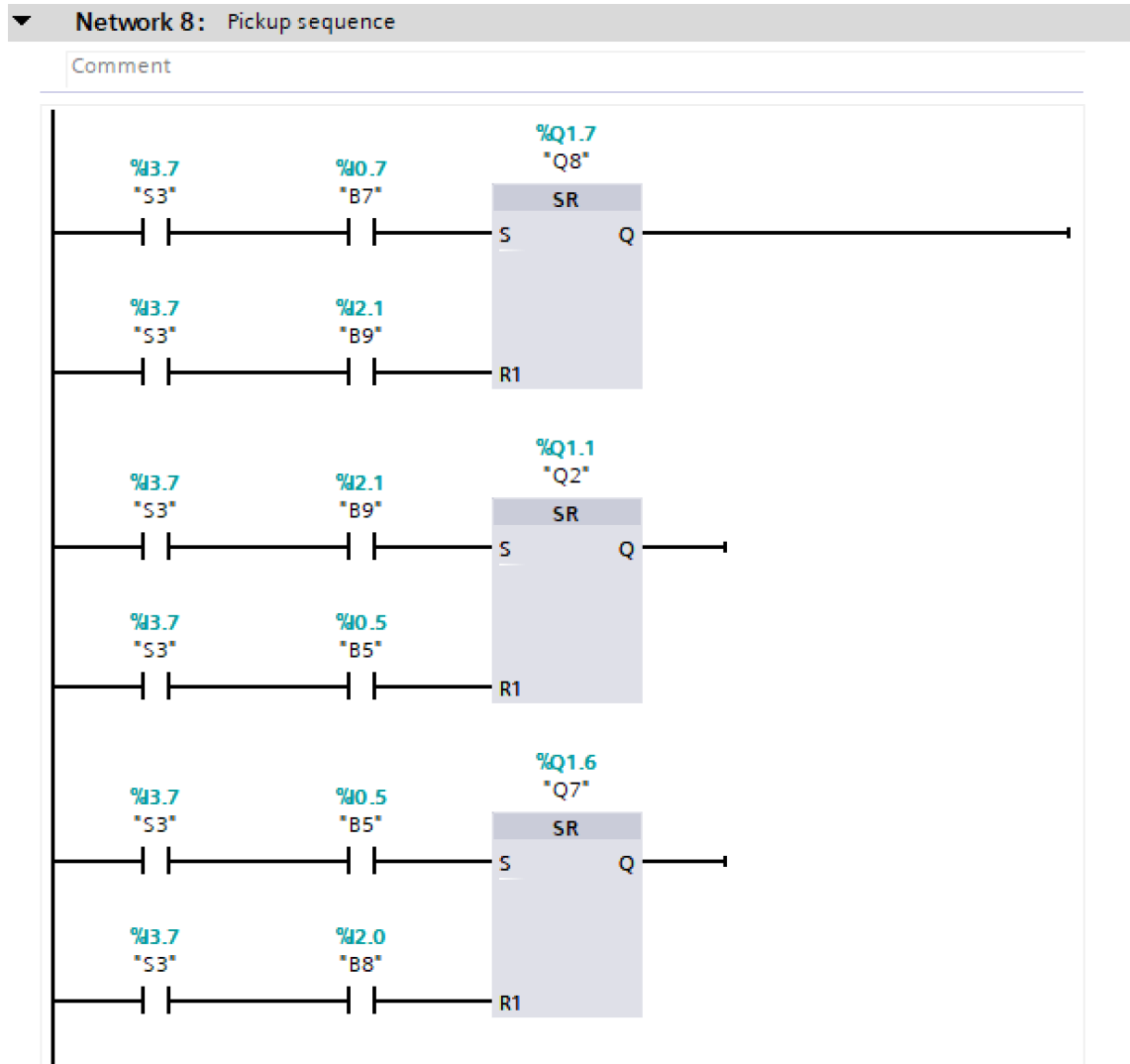
KUVA 22. Noudon position määrittäminen (Kuva: Aleksi Niemi 2024).

Paketin noutoprosessi toimii hyvin vastaavalla tavalla kuin palautus, mutta käänteisesti. Kun Q18 aktivoituu, ajetaan Q9 lähdöllä hihnakäyttöinen aktuaattori positioon kolme (KUVA 22). Samanaikaisesti määritetään mistä tasolta paketti nostetaan telineestä. Tämä toimii käänteisesti palautukseen verrattuna, jolloin jos ensimmäisessä on paketti, on Q11 aktiivinen. Jos ensimmäinen paikka on tyhjä, otetaan paketti seuraavasta tasosta eli tässä tapauksessa Q12. Ohjelma ei anna kahden tason aktivoitua samanaikaisesti.



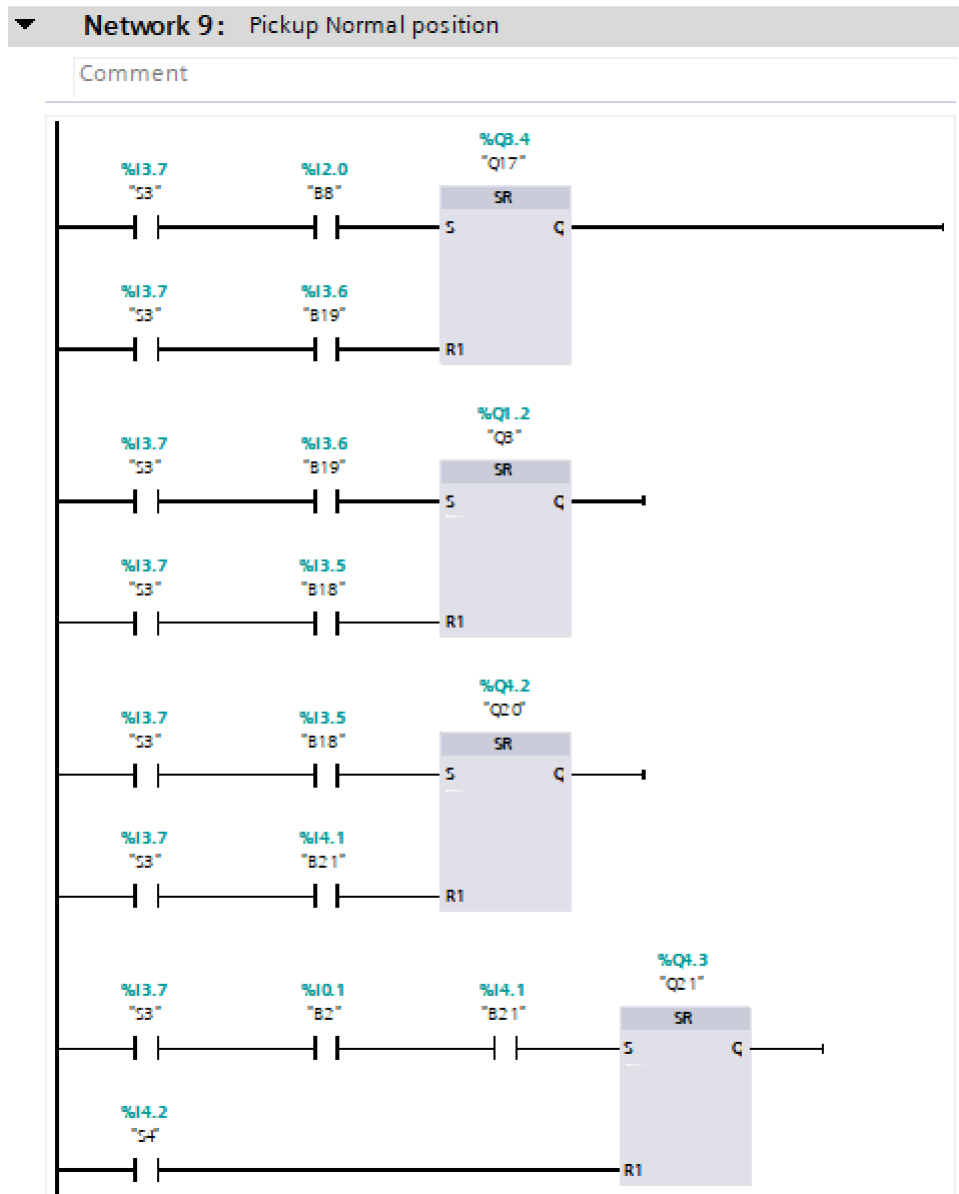
KUVA 23. Paketin nosto telineestä (Kuva: Aleks Niemi 2024).

Aktuaattorin ajuri ilmoittaa (B11) noston onnistuneen, jolloin Q2 työntö aloitetaan. Sen pysäyttää taas anturi B6 (KUVA 23). Tämän jälkeen hammaspyörälliselle aktuaattorille lähetetään signaali nostaa paketti telineestä. Ajuri ilmoittaa onnistuneesta nostosta B20 signaalilla, jolloin Q3 aktivoituu ja paketti vedetään kasaan. B7 anturi pysäyttää aktuaattorin vedon.



KUVA 24. Paketin laskeminen hihnakäyttöisen aktuaattorin telineeseen (Kuva: Aleksi Niemi 2024).

B7 aktivoitua saa hammaspyörällinen aktuaattori signaalin Q8, jolloin se laskeutuu positioon kaksi (KUVA 24). Ajuri lähettää tiedon B9 logiikkayksikölle, kun aktuaattori on siirtymän suorittanut. B9 aktivoi myös Q2 signaalin ja paketti työnnetään samaan tasoon hihnakäyttöisen aktuaattorin kanssa. B5 anturi lopettaa työntöä ja aktivoi Q7 signaalin, joka ohjaa hihnakäyttöisen aktuaattorin positioon kaksi, jolloin se tarttuu paketista. B8 on signaali ajurilta siirron onnistumisesta.



KUVA 25. Noudon toteutuminen ja normaali positio (Kuva: Aleksi Niemi 2024).

B8 signaali aktivoi Q17 lähdön, joka laskee hammaspyörällistä aktuaattoria, jolloin se irtaantuu paketista (KUVA 25). Tämän jälkeen Q3 vedetään työntävä aktuaattori kasaan. B18 pysäyttää työntävän aktuaattorin, kun se on kasassa. Q20 aktivoituu, joka ohjaa hihnakäyttöisen aktuaattorin positioon yksi. Anturi B21 huomaa paketin läsnäolon, jolloin kun QR-koodi luetaan Raspberry Pi:llä, aktivoituu B2 ja valo Q21 syttyy. Valo nollataan kuittaamalla nouto S4 painikkeesta.

5.3 Testaus

Ohjelmaa pystyttiin simuloimaan käyttäen Tia Portalin PLCSIM -ohjelmaa. Simuloinnissa pystytään vaihtamaan tulojen true ja false tilaa. Näin pystyttiin testaamaan toimiko ohjelmoitu toiminto halutulla tavalla.

Parempi tapa testaukseen on kuitenkin PLCSIMIn Sequence -toiminto. Siinä pystytään määrittämään aikamääreillä, milloin mikäkin tulo menee päälle tai pois päältä. Kuvissa 26 ja 27 on noudo sekä sisäänoton sekvenssi.

Time	Name	Address	Display format	Action	Action parameter	Comment
0.000				Trigger condition	"S3" = TRUE	
0.000	*B2*.P	%I0.1:P	Bool	Set to value	FALSE	
0.000	*B12*.P	%I2.6:P	Bool	Set to value	TRUE	
0.000	*B1*.P	%I0.0:P	Bool	Set to value	TRUE	
0.500	*B1*.P	%I0.0:P	Bool	Set to value	FALSE	
2.000	*B11*.P	%I2.5:P	Bool	Set to value	TRUE	
3.000	*B10*.P	%I2.3:P	Bool	Set to value	TRUE	
3.000	*B1*.P	%I0.0:P	Bool	Set to value	FALSE	
3.000	*B11*.P	%I2.5:P	Bool	Set to value	FALSE	
5.000	*B10*.P	%I2.3:P	Bool	Set to value	FALSE	
7.000	*B6*.P	%I0.6:P	Bool	Set to value	TRUE	
8.000	*B6*.P	%I0.6:P	Bool	Set to value	FALSE	
9.000	*B20*.P	%I4.0:P	Bool	Set to value	TRUE	
10.000	*B20*.P	%I4.0:P	Bool	Set to value	FALSE	
14.000	*B7*.P	%I0.7:P	Bool	Set to value	TRUE	
15.000	*B7*.P	%I0.7:P	Bool	Set to value	FALSE	
16.000	*B9*.P	%I2.1:P	Bool	Set to value	TRUE	
17.000	*B9*.P	%I2.1:P	Bool	Set to value	FALSE	
18.000	*B5*.P	%I0.5:P	Bool	Set to value	TRUE	
19.000	*B5*.P	%I0.5:P	Bool	Set to value	FALSE	
20.000	*B8*.P	%I2.0:P	Bool	Set to value	TRUE	
20.000	*B12*.P	%I2.6:P	Bool	Set to value	FALSE	
21.000	*B8*.P	%I2.0:P	Bool	Set to value	FALSE	
22.000	*B19*.P	%I3.6:P	Bool	Set to value	TRUE	
23.000	*B19*.P	%I3.6:P	Bool	Set to value	FALSE	
24.000	*B18*.P	%I3.5:P	Bool	Set to value	TRUE	
25.000	*B18*.P	%I3.5:P	Bool	Set to value	FALSE	
26.000	*B21*.P	%I4.1:P	Bool	Set to value	TRUE	
27.000	*B2*.P	%I0.1:P	Bool	Set to value	TRUE	
28.000	*B21*.P	%I4.1:P	Bool	Set to value	FALSE	
30.000	*S3*.P	%I3.7:P	Bool	Set to value	FALSE	
31.000	*S4*.P	%I4.2:P	Bool	Set to value	TRUE	
34.000		DEC		Set to value	0	
34.005				Stop sequence		

KUVA 26. Noudon simuloitu sekvenssi (Kuva: Aleks Niemi 2024).

Ajamalla sekvenssi läpi, simuloitiin todellista teollista prosessia. Anturit aktivoituivat halutulla tavalla ja ohjelma pystyttiin ajamaan loppuun asti halutulla tavalla. Sekvenssin simulointi todistaa ohjelman toimivuuden sen ehtojen puitteissa.

Simulointi ei korvaa tosimaailman testausta, mutta sillä päästään lähelle ohjelman perusrakenteessa.

Sisäänotto Sequence

Default interval 1.000 s seconds Execution time -----

Time	Name	Address	Display format	Action	Action parameter	Comment
0.000				Trigger condit...	"S1" = TRUE	
1.000	*B1*:P	%I0.0:P	Bool	Set to value	TRUE	
2.000	*B2*:P	%I0.1:P	Bool	Set to value	TRUE	
4.000	*B4*:P	%I0.4:P	Bool	Set to value	TRUE	
5.000	*B1*:P	%I0.0:P	Bool	Set to value	FALSE	
5.000	*B2*:P	%I0.1:P	Bool	Set to value	FALSE	
4.000	*B8*:P	%I2.0:P	Bool	Set to value	TRUE	
5.000	*B8*:P	%I2.0:P	Bool	Set to value	FALSE	
7.000	*B5*:P	%I0.5:P	Bool	Set to value	TRUE	
8.000	*B9*:P	%I2.1:P	Bool	Set to value	TRUE	
9.000	*B5*:P	%I0.5:P	Bool	Set to value	FALSE	
10.000	*B9*:P	%I2.1:P	Bool	Set to value	FALSE	
11.000	*B10*:P	%I2.3:P	Bool	Set to value	TRUE	
12.000	*B10*:P	%I2.3:P	Bool	Set to value	FALSE	
16.000	*B11*:P	%I2.5:P	Bool	Set to value	TRUE	
17.000	*B11*:P	%I2.5:P	Bool	Set to value	FALSE	
17.000	*B6*:P	%I0.6:P	Bool	Set to value	TRUE	
20.000	*B17*:P	%I3.4:P	Bool	Set to value	TRUE	
19.000	*B6*:P	%I0.6:P	Bool	Set to value	FALSE	
21.000	*B17*:P	%I3.4:P	Bool	Set to value	FALSE	
23.000	*B18*:P	%I3.5:P	Bool	Set to value	TRUE	
25.000	*B19*:P	%I3.6:P	Bool	Set to value	TRUE	
24.000	*S2*:P	%I0.3:P	Bool	Set to value	TRUE	
26.000	*S1*:P	%I0.2:P	Bool	Set to value	FALSE	
27.000	*S2*:P	%I0.3:P	Bool	Set to value	FALSE	
28.000		DEC		Set to value	0	
28.005				Stop sequence		

KUVA 27. Sisäänotton simuloitu sekvenssi (Kuva: Aleks Niemi 2024).

6 PÄÄTELMÄT

Opinnäytetyön tavoitteena oli suunnitella prosessi BrainNova Oy:lle, jossa käyttäen useampaa aktuaattoria, siirretään paketti paikasta A paikkaan B. Tavoitteena oli valita tarvittavat komponentit työtä varten, sekä tehdä ohjelma TIA Portal -ohjelmistolla. Näihin tavoitteisiin päästiin onnistuneesti ja rajoitusten puitteissa, pystyttiin valitsemaan tarvittavat komponentit ja luotua toimiva ohjelma TIA Portalilla, jota voidaan käyttää prosessin ajamiseen.

Työssä ei onnistuttu hyödyntämään TIA Portalin RS485 -väylän lisäosaa ja sen sisällyttämistä ohjelmointiin puuttuvan kokemuksen takia. Kyseinen aihe oli myös erittäin spesifi, jolloin siitä ei löytynyt juurikaan ohjeistusta. Tältä osin suunnittelu jäi vajaaksi, mutta sopivan toimijan kanssa saataisiin varmasti myös väylän toimivaksi.

Tässä opinnäytetyössä saatiin kasattua kattava kokonaisuus, joka toimii hyvänä pohjana useamman aktuaattorin yhteisen ohjauksen konkreettiselle toteutukselle. Komponenttien valinnassa tulee käyttää harkintaa, jotta kaikki komponentit saadaan toimimaan toistensa kanssa. Tässä työssä käytetty tapa on vain yksi lukuisista eri tavoista toteuttaa useamman aktuaattorin ohjausta. Kun kyseessä on ainoastaan toteutuksen suunnittelu, jää joitain asioita huomioimatta, jotka tulisivat vastaan fyysisessä toteutuksessa. Tässä opinnäytetyössä ei päätetty käyttää aikaa ASDA ohjelmiston luomiseen, sillä se olisi vaatinut yhteyden fyysiseen toimilaitteeseen. Sama pätee Raspberry Pi:hin, jonka ohjelmointia ei voitu tähän työhön sisällyttää, koska kyseistä tuotetta ei ollut saatavilla opinnäytetyön teko hetkellä.

Tehdystä työstä jatkojalostetaan toimiva prosessi todennäköisesti ulkopuolisen tahon toimesta, jossa opinnäytetyö toimii hyvänä ohjeistuksena. Työssä käydään selkeästi läpi haluttuja ominaisuuksia, joita mahdollisessa toteutuksessa tulisi hyödyntää. Jos opinnäytetyössä tehdylle TIA Portal ohjelmalle on käyttöä koulun ulkopuolisella taholla, tulee kyseinen ohjelma tehdä uudestaan ulkopuolisen tahon hankkimalla lisenssillä. Opinnäytetyössä käytetty lisenssi kattaa vain opintoihin liittyvän ohjelmointityön.

Koen opinnäytetyön ohjelmoinnin osuuden onnistuneeksi ja pidän sitä myös opinnäytetyön tärkeimpänä osuutena. Testauksen ja simuloinnin avulla pystyttiin todentamaan ohjelman toimivuus, mikä oli tärkein tavoite työtä tehdessä. Komponenttien valinnan tekeminen ja niiden yhteensopivuuden selvittäminen oli toinen tärkeä osa tätä opinnäytetyötä. Projektin jatkoa ajatellen ei tarvitse käyttää aikaa komponenttien selvittämiseen, vaan voidaan keskittyä laitteistojen yhteensopivuuden parissa työskentelyyn.

LÄHTEET

Isotechnic. 2023. BELT-DRIVEN VERSUS BALL SCREW ACTUATOR: WHICH IS THE BEST CHOICE FOR YOUR APPLICATION? Verkkosivu. Viitattu 15.4.2024

<https://www.isotechinc.com/belt-driven-versus-ball-screw-actuators/>

Rasberrypi. 2024. Hardware, Software. Verkkosivu. Viitattu 20.4.2024

<https://www.rasberrypi.com/>

The Ives Equipment Process Engineering & Control Blog. n.d. What Are Rack and Pinion Actuators? Verkkosivu. Viitattu 15.4.2024

<https://blog.ivesequipment.com/2018/06/what-are-rack-and-pinion-actuators.html>

Timotion. 2022. What Is An Electric Linear Actuator? Verkkosivu. Viitattu 15.4.2024.

<https://www.timotion.com/en/news-and-articles/part-1-what-is-an-electric-linear-actuator-and-how-to-choose-it>

Virtusa. n.d. Logic Programming. Verkkosivu. Viitattu 18.4.2024

<https://www.virtusa.com/digital-themes/logic-programming>