

Sahatavaran pituusmittaus LiDar-anturilla

LAB-ammattikorkeakoulu

Insinööri (AMK), Konetekniikka

2022

Marko Tuomikoski

Tiivistelmä

Tekijä(t) Tuomikoski, Marko	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika 2022
	Sivumäärä 29	
Työn nimi Sahatavaran pituusmittaus LiDar-anturilla		
Tutkinto Insinööri (AMK)		
Toimeksiantajan nimi, titteli ja organisaatio Raimo Seppänen, Hallituksen puheenjohtaja, Pronor Control Oy		
Tiivistelmä <p>Työn aihe on LiDar-teknologian hyödyntäminen sahateollisuudessa kappalemittaukseen. Toteutus antaa mahdollisuuden mitata kappaleita tarkemmin. Työn aikana kehitettiin logiikkaohjelmisto. Työn tilaajana toimi Pronor Control Oy. Tavoitteena oli tuottaa toimiva logiikkaohjelma, jota pystytään hyödyntämään sahateollisuudessa.</p> <p>Työ toteutettiin tilaajan tiloissa Heinolassa. Työhön kuului selvitys siitä, kuinka logiikkalaitteisto ja LiDar-anturi saadaan yhdistettyä toisiinsa, lisäksi tuotettiin työhön tarvittu logiikkaohjelma.</p> <p>Työn aikana kehitettiin logiikkaohjelma, jota tulevaisuudessa käytetään vastaavissa toteutuksissa. Prototyypin testaaminen päätettiin toteuttaa tilaajayrityksen tiloissa. Työn onnistuminen mitattiin siitä, saadaanko anturilla mitattua toleranssien sisälle todellisesta kappaleen pituudesta.</p> <p>Tehtyä työtä hyödynnetään myöhemmin erilaisissa kehitysprojekteissa sahateollisuudessa. Työn tilaajayritys voi hyödyntää tuotettua ohjelmistoa markkinoimalla sitä sahalaiteille. Ohjelmistoa kehitetään eteenpäin, jotta mittaustarkkuutta saadaan kasvatettua. Tarkastelematta jäi, mihin muuhun teknologiaa voitaisiin hyödyntää. Tulevaisuudessa tulisi kehittää mittausprosessia pidemmälle, jotta voitaisiin mitata esimerkiksi useampaa dimensiota.</p>		
Asiasanat LiDar, Sahateollisuus, Laser		

Abstract

Author(s) Tuomikoski, Marko	Type of Publication Thesis, UAS	Published 2022
	Number of Pages 29	
Title of Publication Measuring lumber length with LiDar Sensor		
Name of Degree Engineer		
Name, title and organization of the client Raimo Seppänen, Chairman of the board, Pronor Control Oy		
Abstract <p>Subject of this thesis was LiDar-technology and how it could be used to measure pieces in sawmill industry. The implementation of this project allows for more accurate measurements of lumber. During thesis a program was produced. This thesis was commissioned by Pronor Control ltd. The goal was to produce a program that could be used in sawmill industry.</p> <p>The project was done at commissioner's facilities in Heinola. Part of the project was to re-search how to operate LiDar sensor with a logic controller. Additionally, a logic program was produced.</p> <p>During the making of the thesis a logic program was produced, which would be used in similar implementations. The testing of program was done at commissioner's facilities. The success of the implementation was measured by how accurately the program performed.</p> <p>The produced program will be later used in variety of projects for sawmill industry. The commissioner can utilize the program by marketing it as an alternative method of measurement to sawmills. Further development will focus on increasing the accuracy of the measurements further. How this technology could be used to measure multiple dimensions should be investigated.</p>		
Keywords LiDar, sawmill industry, Laser		

Sisällys

1	Johdanto.....	1
1.1	Toimeksiantaja	1
1.2	Opinnäytetyön tavoite	1
2	Sahausprosessi	2
2.1	Tukkien lajittelu.....	2
2.2	Tukkien lajittelulaitokset.....	3
2.3	Kuorinta	3
2.4	Sahaus prosessi	4
2.5	Sahaustavat.....	4
2.6	Dimensiolajittelu	5
2.7	Rimoitus ja kuivaus.....	5
2.8	Kuivan sahatavaran lajittelu	7
3	LiDar-anturitekniikka	8
3.1	LiDar-teknologia	8
3.2	Anturin valinta.....	9
3.2.1	Riegl	9
3.2.2	Pepperl+Fuchs	10
3.2.3	Sick	10
4	Laitteisto ja ohjelmisto.....	12
4.1	2D LiDar-anturi	12
4.2	Ohjelmoitava logiikka.....	12
4.3	SOPAS Engineering Tool	12
4.4	Tia Portal	15
5	Sovellusohjelmointi	18
5.1	Anturiin yhdistäminen	18
5.2	Anturin ohjaaminen ja datan lukeminen	19
5.3	Datan prosessointi ja tallentaminen	21
5.4	Laskenta.....	23
6	Testaus.....	26
7	Käyttöönottosuunnitelma	28
8	Yhteenveto	29
	Lähteet	30

1 Johdanto

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana oli heinolalainen Pronor Control Oy. Pronor Control on vuonna 2005 perustettu sähkö- ja automaatioalan yritys, joka on erikoistunut sahateollisuuden automaatio suunnitteluun ja ohjausjärjestelmien toimitukseen. Pronor Control toimittaa sähkö- ja automaatio suunnittelua, automaatio-ohjelmointia sekä käyttöliittymien suunnittelua ja ohjauspaneelien toteutuksia. Asiakaskunta koostuu sahateollisuuden johtavista yrityksistä, joihin kuuluu mm. HaSa Group ja Versowood Oy. Asiakaskuntaan kuuluu myös sahateollisuuden koneita ja laitteita toimittavia yrityksiä. (Pronor Control Oy)

1.2 Opinnäytetyön tavoite

Sahalaitoksissa sahattujen kappaleiden mittaus toteutetaan usein käyttäen valokennoja, jotka on sijoitettu linjastolle riviin poikkisuuntaisesti kuljetinlinjaan nähden tietyn etäisyyden päähän toisistaan. Tämä on toimiva ratkaisu, mutta se voi olla epäkäytännöllinen tai kallis tapa toteuttaa mittaus. Jos esimerkiksi jokin valokennoista hajoaa sitä ei välttämättä huomata välittömästi. Tämä voi johtaa kappaleiden virheelliseen mittaukseen.

Opinnäytetyön tavoitteena on selvittää, kuinka LiDar-teknologiaa voitaisi hyödyntää sahateollisuuden kappaleiden pituusmittauksessa. Työhön kuuluu selvitystyön lisäksi logiikkaohjelman toteuttaminen valitulle ohjelmoitavalle logiikalle.

2 Sahausprosessi

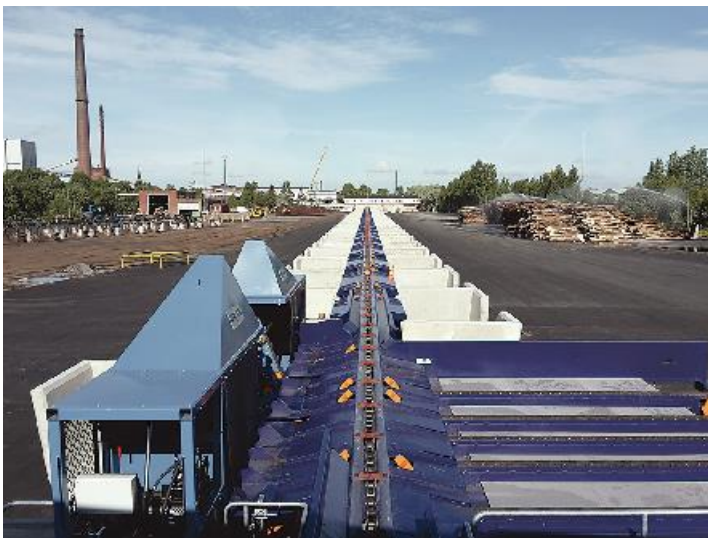
2.1 Tukkien lajittelu

Sahausprosessi alkaa tukkilajittelusta. Lajittelussa mitataan latva- ja tyviläpimitta, pituus, kartiokkuus, lenkous, soikeus ja tilavuus. Tämän jälkeen tukit ohjataan tukkilokeroon erilaisten lajitteluehtojen perusteella. (Varis 2017, 47.) Lajittelu ehtoina voivat olla esimerkiksi tukin halkaisija, pituus tai suoruus. Myös muita lajitteluun vaikuttavia tekijöitä on olemassa, esimerkiksi homevauriot tai hyönteistuhot. Kuvassa 1 on lajittelulinjaston alussa sijaitseva tukkimittari, jonka perusteella mittaus ja lajittelu tapahtuvat.



Kuva 1. Tukkilajittelijan alussa oleva tukkimittari (Varis 2017, 50)

Lokeroita on yleensä 30 lokerosta 80 lokeroon, mutta uusimmissa laitoksissa niitä voi olla jopa 120. Lokeroiden suuri lukumäärä mahdollistaa lajittelun suuremmalla tarkkuudella. Lajittelu voidaan toteuttaa samanaikaisesti esimerkiksi normaalikoisille ja pienille tukeille. Myös mittausteknologian kehitys ja asiakkaiden tarpeiden erikoistuminen vaativat kykyä lajitella tukit useampaan eri luokkaan. Erilaiset asiakas tarpeet lisäävät lajittelu lokeroiden tarvetta entisestään. (Varis 2017, 47.) Kuvassa 2 nähdään tukkilinjasto ja sen varrella olevat tukkilokerot.



Kuva 2. Tukin lajittelulinja UPM:n Seikun sahalla Porissa (Varis 2017, 49)

2.2 Tukkien lajittelulaitokset

Lajittelulaitoksen on suunniteltu siten, että tukit tuodaan sahan lajittelulaitoksen tukkipöydälle. Tästä ne siirtyvät lajittelu radalle, jossa ne laadutetaan ja tämän jälkeen pudotetaan yhteen monista lajittelulokeroista. (Varis 2017, 49.)

Tukkien lajittelussa hyödynnetään paljon modernia teknologiaa, 3D-mittausta ja röntgeniä. Se mahdollistaa tukkien lajittelun suuremmalla tarkkuudella. Lajitteluprosessi on usein pitkälle automatisoitu. Tämä mahdollistaa sen, että operaattori voi keskittyä paremmin prosessin yleiskuvaan ja visuaaliseen lajitteluun.

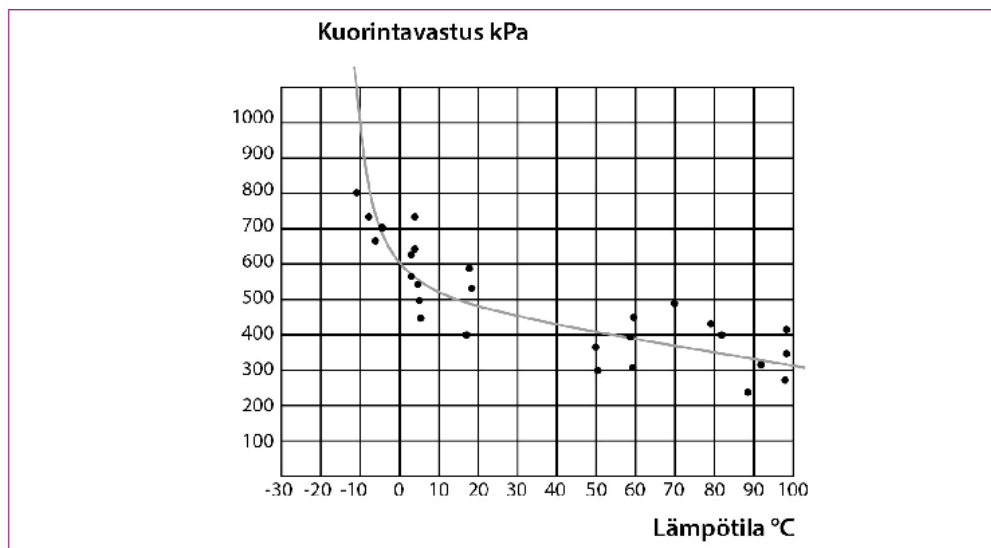
Visuaalisessa lajittelussa operaattori arvioi tukin silmämääräisesti. Tämä visuaalinen lajittelutieto yhdistetään automaattisesti tuotettuun lajitteludataan, jonka perusteella tehdään lopullinen lajittelupäätös. Operaattori tarkistaa visuaalisesti lajitteluun vaikuttavia tekijöitä, joita automaattinen lajittelu ei havaitse kuten, sinistymät ja kovalaho. (Varis 2017, 51.)

2.3 Kuorinta

Tukit kuoritaan puhtaaksi ennen sahausprosessia, jotta sahatavara pysyisi puhtaana. Kuorinnan tehtävänä on erottaa kuori puumateriaalista kohdistamalla riittävän suuri voima puuta ja kuorta yhdessä pitävään jälsikerrokseen.

Puun kuoren määrään vaikuttava muun muassa puun ikä, koko, laji ja maantieteellinen sijainti. Puun kasvaessa myös kuoren paksuus kasvaa, mutta kuoren suhde hyödynnettävän puun määrään pienenee. (Varis 2017, 55.)

Puun ja kuoren välisen sidoksen lujuus vaihtelee huomattavasti puulajien ja olosuhteiden mukaan. Esimerkiksi kuivassa puussa kuori on kiinni huomattavasti voimakkaammin kuin märässä puussa. Myös lämpötilalla on vaikutus siihen, kuinka vahvasti kuori on kiinni puussa. Jäätäneessä puussa kuori on erittäin voimakkaasti kiinni puussa. (Varis 2017, 57.) Kuviossa 1 nähdään, että lämpötilan kasvaessa kuorintavastus pienenee, varsinkin kun lämpötila ylittää 0 °C.



Kuvio 1. Lämpötilan vaikutus kuorinta vastukseen (Varis 2017, 58)

2.4 Sahaus prosessi

Sahausprosessissa on sahaamisen ja särmäyksen avulla tarkoitus saada mahdollisimman suuri määrä laadukasta sahatavaraa. Tähän pyritään erityyppisillä laitteistoilla ja sahaus-tekniikoilla. (Varis 2017, 65.)

Käytössä olevia sahaustapoja on monia, esimerkiksi: nelisahaus, läpisahaus, tähtisahaus, ympärisahaus, trapetsisahaus, epäsymmetrinen sahaus ja komponentti sahaus. Osa niistä käytetään yleisesti kaiken kokoisilla sahoilla, mutta osa soveltuu paremmin pienemmillä toimijoille.

2.5 Sahaustavat

Nelisahaus on pohjoismaissa yleisin käytössä oleva sahaustapa. Siinä tukista haketetaan kaksi sivua ja erotetaan näiltä sivuilta ennalta määrätty määrä sivu- ja pintalautoja. Tämän jälkeen syntynyt pelkka käännetään 90 astetta ja suoritetaan käsittelemättömien sivujen haketus ja jakosahaus. Näin saadaan eroteltua pinta- ja sivulaudat ja keskitavara. (Varis 2017, 65.)

Läpisaauksessa tukista halutaan mahdollisimman paljon raakasahatavaraa, esimerkiksi komponenttiteollisuuden tarpeisiin. Tässä menetelmässä kappaleita ei särmätä, vaan ne lajitellaan ja kuivataan särmäämättömänä. Kuivauksen jälkeen sahatavara jalostetaan, jolloin siitä sahataan tarpeen mukaan oksatonta tai terveoksaista raaka-ainetta muun puuteollisuuden käyttöön. (Varis 2017, 67.)

Ympärisahaus on pohjoismaissa lähinnä pienten yrittäjien käyttämä sahausmenetelmä. Sahaauksessa tukista sahataan yksi kappale kerrallaan. Tätä menetelmää käytetään eteläisessä Euroopassa suurien tukkidimensioiden sahaaukseen. Näin saadaan kalliista raaka-aineesta mahdollisimman suuri osuus hyötykäyttöön, vaikka työnopeus onkin alhainen. (Varis 2017, 68.)

2.6 Dimensiolajittelu

Tukin sahaamisen jälkeen prosessin seuraava vaihe on lajittelu. Tuorelajittelu, jota myös dimensiolajitteluksi kutsutaan, on lajitteluprosessi, jossa sahatavara lajitellaan sen mittojen perusteella. Nykyään lajitteluun voi dimension lisäksi vaikuttaa myös puutavaran laatu jo tuorelajittelun aikana. Tämä tapahtuu usein automaattisesti konenäköjärjestelmiä käyttäen. (Varis 2017, 98.)

Dimensiolajittelussa sahatavara lajitellaan lokeroihin sen paksuuden ja leveyden perusteella. Prosessoitavan sahatavara määrän ja eri dimensioiden takia tarvitaan useita lajittelulokeroita. Suuri käsiteltävän sahatavaran määrä kasvattaa sahalaitoksen kokoa. Kuvassa 3 nähdään osa lajitteluun käytetyistä varastolokeroista.



Kuva 3. Dimensiolajittelulaitoksen vinolokerosto (Varis 2017, 100)

2.7 Rimoitus ja kuivaus

Rimoituksessa sahatavara kasataan tasakokoiseen pakettiin, jossa jokaisen kerroksen väliin lisätään poikittain välirimoja erottelemaan kerrokset toisistaan. Tämä mahdollistaa ilman

kulkemisen paketin läpi. Kuivaamo kuorman tekeminen mahdollistaa näin halutun kuivaustuloksen saavuttamisen tehokkaasti koneellisessa kuivauksessa. Kuvassa 4 nähdään valmis rimapaketti.



Kuva 4. Rimoitettu kuorma valmiina kuivaukseen (Varis 2017, 106)

Anturi sijoitetaan prosessissa kuljettimen yläpuolelle ennen rimoituksen kappaleidenannostelija. Anturia käytetään mittaamaan rimakuorman kappaleiden pituus. Mittatietoa hyödynnetään muun muassa rimapaketin kuutioiden laskentaan.

Puun kuivaamiseen tarkoituksena on poistaa vettä sahatavarasta mahdollisimman pienellä raakaa-aineen hävikillä ja kuivaus kustannuksella. Sahatavara kuivataan pääsääntöisesti ilmaa kierrättämällä. Kuivaamoissa lämmön siirtäminen toteutetaan puhaltamalla ilmaa rimakuorman läpi.

Puun kosteuspuoisuus ilmoitetaan prosentteina ja sillä tarkoitetaan puussa olevan veden painon suhdetta puun absoluuttiseen kuivapainoon. Vasta sahatun puun kosteus on yleensä 40–200 %. Normaalikäytössä puun kosteus vaihtelee 8–25 painoprosentin välillä ilman suhteellisen kosteuden mukaan. (Varis 2017, 107)

Korkean lämpötilan kuivaus tarkoittaa prosessia, jossa lämpötila on yli 100 °C. Lämmön siirron väliaineena toimii höyry. Tyhjiökuivaus perustuu veden kiehumiseen alipaineessa alhaisemmassa lämpötilassa. Tämä nopea kuivaustapa sopii puulajeille, joita on vaikea kuivata aiheuttamatta värivikoja. Muun muassa tammi, pyökki ja koivu kuuluvat näihin puulajeihin. Kuivauslämpötila vaihtelee 50 Celsius asteesta 80 Celsius asteeseen. (Varis 2017, 109.)

Lauhdekuivaus on variaatio ilmankiertoon perustuvasta kamarikuivauksesta. Prosessissa vesihöyryä ei johdeta pois kuivaamosta, vaan se puhalletaan lauhduttimen lämpöpumpulle, jonka jälkeen höyrystyslämpö palautetaan takaisin prosessiin. (Varis 2017, 109.)

Kamarikuivaus soveltuu kaikenlaisen puutavaran kuivaukseen. Kuivaus laitos voidaan mitoitaa mille tahansa sahatavara paksuudelle. Kamarikuivaus soveltuu erityisesti pienten erien ja halkeamisherkkien, järeiden sahatavaroiden kuivaamiseen. (Varis 2017, 109.)

2.8 Kuivan sahatavaran lajittelu

Kuivauksen jälkeen sahatavara puretaan lajittelulaitoksen vastaanotto kuljettimelle. Lajittelulaitoksessa kuivattu sahatavara lajitellaan laadun perusteella, katkaistaan lopulliseen mittaan ja ohjataan paketointiin.

Kuivalajittelulaitoksella sahatavaran päät tasataan joko vakio- tai asiakaspituuteen. Kuivan sahatavaran lajittelu prosessi alkaa rimakuorman purkamisesta. Kuorma syötetään purkuhissille, joka nostaa ja kallistaa kuormaa vasteita vasten. Ylin kerros valuu paketin päältä syöttöpöydälle välirimojen pudotessa omalle kuljettimelleen. Laudat viedään lajittelu pöydällä tavaravirtaa tasaavien laitteiden läpi suuren kappalemäärän takia. Näihin laitteeseen kuuluvat muun muassa porrasnostimet ja kiramot. Porras nostin nostaa pienen erän lautoja kiramo edeltävälle syvennykselle, josta kiramo nostaa laudan kerrallaan arvostelupöydälle. (Varis 2017, 123–124)

Laitoksen alkupään tarkoitus on erotella sahatavara kappaleet toisistaan, jotta jokainen kappale voidaan yksitellen laaduttaa. Jotta tämä olisi mahdollista tarvitaan kappaleannostelua. Sahatavara kappale tulee annostella omaan tilaansa kolaväliin ketjukuljettimelle. Ketjukuljettimella kappaleet on erotettu toisistaan ketjussa olevilla kiinteillä kolilla. (Varis 2017, 124)

Kappaleannostelun jälkeen sahatavara tasataan. Tasaamalla vastaanottokuljettimien jälkeen kappaleiden päät tasataan samaan linjaan. Tähän käytetään linjan kulkusuunnan suuntaisi rullia. Ne pyörivät linjan kulkusuuntaa vasten ja siirtävät kappaleet samaan linjaan kuljettimen reunalla sijaitsevaa estettä vasten. (Varis 2017, 124)

Tämän jälkeen sahatavara kappaleet laadutetaan ja päätetään mistä kohtaa se tasataan. Tasaamalla voidaan kappaleen päädyistä poistaa virheitä.

Tasaamisen jälkeen sahatavara lajitellaan ja varastoidaan siihen varattuihin lokeroihin laadutuksen perusteella, esimerkiksi dimension perusteella. Tämän jälkeen sahatavara puretaan lokero kerrallaan ja kasataan valmiiseen pakettiin ja paketoidaan valmiiksi tuotteeksi.

3 LiDar-anturitekniikka

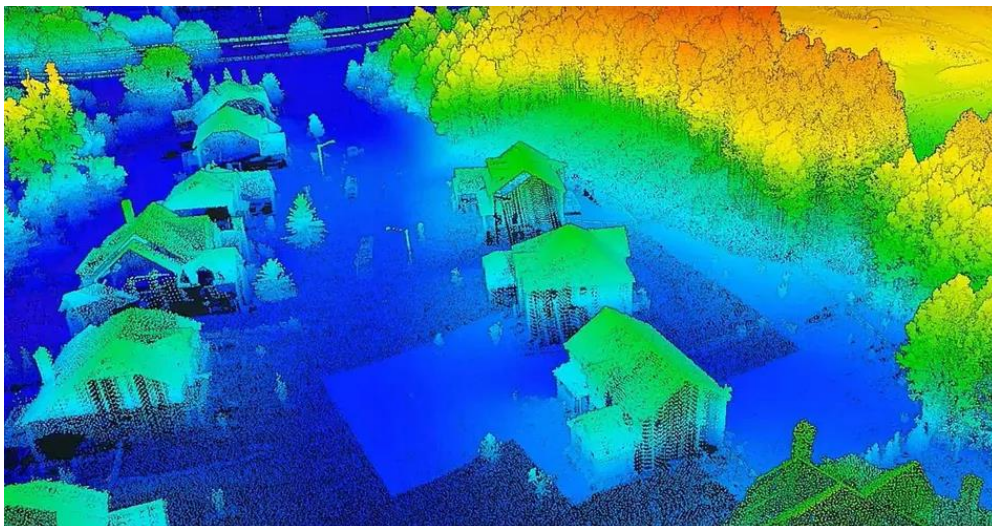
3.1 LiDar-teknologia

LiDar on etäisyyden mittaamiseen käytetty teknologia, jonka toiminta perustuu valon heijastumiseen mitattavasta kohteesta. Kuviossa 2 on kuvattu LiDar-teknologian toiminta periaate, kuinka etäisyys voidaan laskea ajan ja valon nopeuden avulla.



Kuvio 2. LiDar-teknologian toimintaperiaate (Notaro 2021)

Termi LiDar tulee englanninkielisistä sanoista Light Detection and Ranging. Teknologian yleisin käyttötarkoitus on suuri resoluutioisten karttojen tuottaminen. Myös autoteollisuus käyttää LiDar-teknologiaa autonomisten autojen ohjaamiseen. Teollisuudessa LiDar-teknologiaa käytetään esimerkiksi kappaleiden sijainnin tunnistukseen. Kuvassa 5 näkyy LiDar-anturin tuottama ilmakehän kuva, jossa korkeammat kohteet näkyvät kirkkaammalla värillä.



Kuva 5. LiDar-anturin tuottama ilmakehän kuva (Gjorgjioska 2018)

LiDar on laservaloon perustuva anturi teknologia, anturi lähettää laser valoa mitattavaan kohteeseen ja laskee kuinka kauan valolta kestää heijastua takaisin kohteesta. (GeoSlam, 2022)

LiDar, eli 3D-laserskannaus toimii kuten tutka. Radioaaltojen sijasta anturi lähettää laser pulsseja nopealla taajuudella, jopa 160 000 pulssia sekunnissa kohti mitattavaa kohdetta. Etäisyys kohteeseen voidaan laskea kaavalla (Kaava 1). (P.Kumar, 2022.)

$$S = c / t / 2 \quad (1)$$

3.2 Anturin valinta

Työhön valittavan anturin on täytettävä tietyt ehdot: mitta-alueen leveyden on oltava vähintään 3 metriä, mittaustarkkuuden on oltava tarpeeksi suuri, jotta kappaleen pituus mittalolisi tarpeeksi tarkasti luettavissa, mittaussnopeuden on oltava tarpeeksi nopea, jotta kappaleita pystytään prosessoimaan toivotulla nopeudella. Myös laitteen vaivaton ohjelmointi käyttökohteesta riippumatta on tärkeä tekijä. Laitteen tuottama mittausdata on myös saatava talteen sellaisessa muodossa, jota käytössä oleva logiikka pystyy tulkitsemaan.

Koska sovelluksessa halutaan tarkkailla vain yhtä kohtaa linjastosta, suunnattu mitta-alue on parempi. Suunnattu mitta-alue mahdollistaa myös sen, ettei laitteen käyttämä laservalo pääse heijastumaan linjaston ulkopuolelle, esimerkiksi operaattoreiden silmiin.

3.2.1 Riegl

Yhtenä vaihtoehtoisena sovelluksen anturointiin tarkastettiin Riegl VZ-200-tyyppistä anturia. Kuvassa 6 näkyy tarkasteltavana oleva anturi.



Kuva 6. RIEGL VZ-200 anturi (Riegl 2019a)

Anturin skannaussnopeus on 550 000 kertaa sekunnissa. Mittausetäisyys on 1.5 metristä 390 metriin ja mittausssektori on 110 astetta. Kulmaresoluutio on 0.2 ja 0.0007 asteen välillä.

Anturin käyttämä kommunikaatio protokolla on ohjelmoinnissa ja datan palauttamisessa TCP/IP. (Riegl, 2019b.)

Anturin mittausnopeus kohteeseen on riittävä. Mitattavan kappaleen etäisyys on myös riittävä. Kulmaresoluutio mahdollistaa 70 asteen alueelta 350 mittauspisteestä 100 000 mittauspisteeseen.

3.2.2 Pepperl+Fuchs

Toisena vaihtoehtona sovelluksen anturointiin tarkasteltiin Pepperl+Fuchs OMD10M-R2000-B23 anturia. Kuvassa 7 näkyy tarkasteltavana oleva anturi.



Kuva 7. Pepperl+Fuchs OMD10M-R2000-B23 anturi (Pepperl+Fuchs 2016, 9)

Anturin skannausnopeus on 250 000 mittausta sekunnissa. Skannattavan kohteen etäisyys voi olla 0.2 metristä 3 metriin. Mittausalue kyseisellä anturilla on 360 astetta. Kulmaresoluutio anturilla on 0.014 astetta. Anturin käyttämä kommunikaatio protokolla on ohjelmoinnissa ja datan palauttamisessa TCP/IP, HTTP tai UDP/IP. (Pepperl+Fuchs 2016, 29–32).

Anturin mittausnopeus on kohteeseen riittävän nopea. Mitattavan kappaleen etäisyys ja mitta-alue ovat myös riittävät. Kohteessa on tarve vain yksisuuntaiseen mittaukseen. Täten tarvetta 360 asteen mittaus alueelle ei ole. Anturin kulmaresoluution mahdollistaa 70 asteen alueelta 5 000 mittaus pisteteen saamisen.

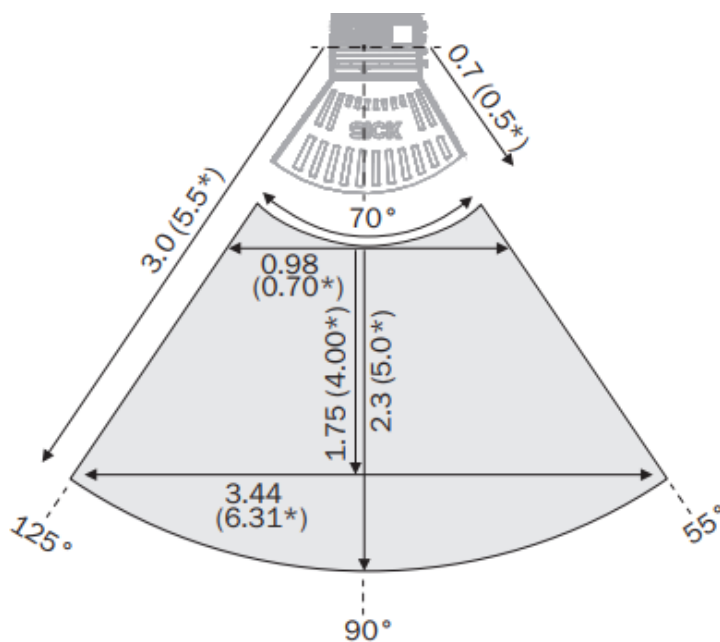
3.2.3 Sick

Kolmantena vaihtoehtona käytettäväksi anturiksi oli Sick LMS4111R-13000 anturi. Kuvassa 8 tarkasteltavana oleva anturi.



Kuva 8. Sick LMS4111R-13000 anturi (Sick 2022)

Anturin skannausnopeus on 600 mittausta sekunnissa. Skannattavan kohteen etäisyys laitteesta voi olla 0.7 metristä 3.0 metriin ja mittaussektori on suurimmillaan 70 astetta. Anturin käyttämä kommunikaatio protokolla on ohjelmoinnissa ja datan palauttamisessa TCP/IP tai UDP/IP. (Sick 2022.) Kuviossa 3 nähdään alue jolta anturi pystyy kappaleita havaitsemaan.



Kuvio 3. Anturin mittaus alue (Sick 2022)

Kyseisellä kulma-alueella ja maksimi etäisyydellä saadaan suurimmaksi mitattavaksi pituudeksi 3.4 metriä. Skanneri tuottaa mittapisteitä $1/12^\circ$ välein tuottaen täten 841 mittapistettä 70° alueelta.

Anturin mittausnopeus ja tarkkuus riittävät tutkittavaan sovellukseen. Myös anturin vain yhteen suuntaan kohdistuva 70° mittaus alue on käyttökohteeseen sopiva. Mahdollisuus anturin asetusten muuttamisen käyttäen erillistä sovellusta on merkittävä tekijä.

4 Laitteisto ja ohjelmisto

4.1 2D LiDar-anturi

Työssä käytettävä anturi on Sickin LMS4000 2D LiDar-anturi. Anturilla pystyttiin saavuttamaan tarvittava 3 metrin mittausalue. Myös 841 mittauspisteen maksimitarkkuus tuolta alueelta todettiin riittäväksi kyseiseen tarkoitukseen.

Anturin ohjelmoiminen onnistuu sekä käyttäen Sickin tarjoamaa ohjelmistoa että komennoilla suoraan logiikalta. Tämä oli merkittävä tekijä anturin valinnassa. Esimerkiksi se kuinka monta mittausa yhdestä laudasta suoritetaan, voidaan määrittää suoraan logiikalta.

4.2 Ohjelmoitava logiikka

Projektissa käytetty logiikka on Siemensin 1517TF-3 PLC. Kyseinen logiikka on tarpeeksi tehokas pystyäkseen prosessoimaan anturilta tuleva suurikin datamäärä nopeasti.

Kyseisen logiikan käyttö oli myös asiakasyrityksen toive. Yrityksen edeltävä kokemus Siemensin logiikkojen käytöstä on olennainen syy kyseisen logiikan käyttöön. Kuvassa 9 on projektiin käyttöön tuleva logiikkaohjain.



Kuva 9: Siemens 1517 CPU (Siemens 2022)

4.3 SOPAS Engineering Tool

Sopas Engineering Tool on Sickin tarjoama ohjelmisto, joka on suunniteltu anturien ohjaimiseen ja parametroiintiin. Sovelluksen kautta voidaan määrittellä mittausalue, mitattavien pisteiden määrä, lähetettävät tiedot ja datan mahdolliset suodatukset.

Kuviossa 4 näkyy anturin mittaukseen vaikuttavat asetukset. Muun muassa mittauksen aloitus- ja lopetus kulma. Mittauksen pienin aloituskulma on 55 astetta ja suurin lopetuskulma 125 astetta mahdollistaen suurimmaksi mittaus alueeksi 70 astetta.

▼ Filter

Start angle	<input type="text" value="55.00"/>	↑ ↓	°	
Stop angle	<input type="text" value="125.00"/>	↑ ↓	°	
Scan range			70.00 °	
Cubic filter active	<input type="checkbox"/>			?
Edge filter active	<input type="checkbox"/>			?
Median filter active	<input type="checkbox"/>			?
Gloss compensation	<input type="checkbox"/>			?
Mean filter active	<input checked="" type="checkbox"/>			?
Number of Scans	<input type="text" value="25"/>	↑ ↓		
Angular resolution	<input type="text" value="0.25"/>	▼		?

▼ Output data format

Distance	<input checked="" type="checkbox"/>	Device name	<input type="checkbox"/>	?
Angle	<input checked="" type="checkbox"/>	Quality	<input type="checkbox"/>	
Encoder ticks	<input type="checkbox"/>	RSSI	<input type="checkbox"/>	
Timestamp	<input type="checkbox"/>			Digits ▼
Output interval	<input type="text" value="1"/>			
Data output during laser off	<input type="checkbox"/>			?

Kuvio 4. Anturin skannaus määrittely

Myös laitteen datankeruuseen liittymättömät parametrit kuten IP-osoite, laitteen nimi ja kommunikaatio tyyppi määritellään ohjelmiston kautta. Kuviossa 5 näkyy laitteen IP:n määrittelyyn käytetty sivu. Myös laitteen käyttämä kommunikaatio portti ja protokolla valitaan tältä sivulta.

▼ General

Address mode: ?

IP address:

Subnet mask:

Default gateway:

MAC address:

Negotiated speed:

▼ Host port (parametrisation)

CoLa dialect: ?

Port:

Kuvio 5. Anturin IP-osoitteen ja data virran asetus

Anturia on mahdollista ohjata ja sen asetuksia muuttaa myös Sopas Engineering Toolin konsolin tai Tia Portalin kautta lähettämällä sille tavuista koostuva datavirta, joka sisältää tiedon siitä mitä laitteen toivotaan tekevän tai mitä arvoja halutaan muuttaa. Data lähetetään heksadesimaali merkeinä. Kuviossa 6 näkyy Sopas Engineering Toolin konsoli, josta anturin toimintaa voidaan ohjata ja testata.

SICK

LMS41xxx (SN 21480184)

nput format: ASCII Hex 🟢 🔴 🟢 🔴

<02>...<00> | <00> ▼

No.	Com	Length	Data
1	🔌		Device connected: LMS41xxx (SN 21480184)
2	⬇️	25	<02><02><02><02><00><00><00><10><73><4d><4e><20><4c><4d><43><73><74><61><72><74><6d><65><61><73><68>
3	⬆️	27	<02><02><02><02><00><00><00><12><73><41><4e><20><4c><4d><43><73><74><61><72><74><6d><65><61><73><20><00><44>

Kuvio 6. SOPAS Terminal: Esimerkki heksadesimaali muodossa olevasta ohjauskäskystä ja anturin vastauksesta

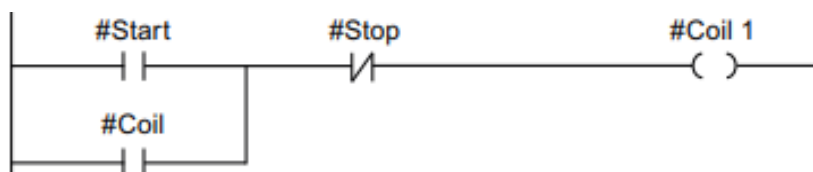
Tähän projektiin anturi asetettiin mittaamaan 70 ° alueelta 0.25 ° välein. Näin saatiin 281 mittaus pistettä koko mittausalueelta. Laitteelta haluttiin tallentaa vain etäisyystieto eri mittapisteisiin. Data luettiin PLC:lle heksadesimaaliarvoina ja muunnettiin logiikassa desimaalilukuarvoksi.

4.4 Tia Portal

Tia Portal tulee sanoista Totally Integrated Automation Portal. Tia Portal on Siemensin tarjoama ohjelmisto, jota käytetään tuottamaan ohjelmistoja Siemensin ohjelmoitaviin logiikoihin. Tia Portaliin on integroituna useita Siemensin ohjelmistoja, STEP 7, SIMATIC WinCC ja SINAMICS Startdrive.

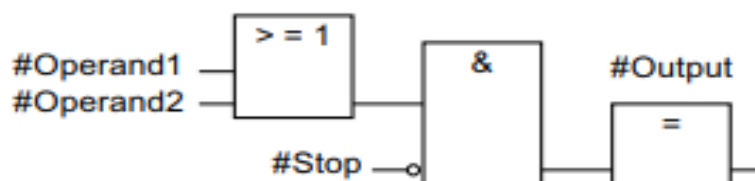
STEP 7 ohjelmaa käytetään PLC-ohjelmointiin. Ohjelmointiin voidaan käyttötarpeen mukaan valita yksi tai useampi seuraavista ohjelmointikielistä.

1. LAD eli Ladder Logic on graafinen kuvaus STEP 7-ohjelmointikielestä. Se perustuu piirikaavioon. Virtakisko on vasemmalla, tätä seuraa johdin, jolle voidaan asettaa erilaisia objekteja, kuten suljettuja tai avoimia kontakteja tai monimutkaisempia funktioita. (Siemens 2021, 7857.) Kuviossa 7 nähdään LAD-kielisen ohjelmalohkon rakenne yksinkertaisimmillaan.



Kuvio 7. LAD-kielen esimerkki (Siemens 2021, 7857)

2. FBD eli Function Block Diagram on graafinen ohjelmointikieli. Se perustuu piirikaavioon. Toiminta perustuu Boolean algebraan. (Siemens 2021, 7919.) Kuviossa 8 on FBD-kielillä muodostettu ohjelma.



Kuvio 8. FBD-kielen esimerkki (Siemens 2021, 7919)

3. STL eli Statement List on teksti pohjainen ohjelmointikieli. Sitä luetaan ylhäältä alaspäin rivi kerrallaan. Yhdellä rivillä voi olla vain yksi komento ja muuttuja. (Siemens 2021, 7975) Kuviossa 9 on kolmen rivin mittainen esimerkki ohjelma toteutettuna statement list-kielillä.

STL	Explanation
A "Tag_Input_1"	// Check whether the signal state of the operand is "1" and AND with current RLO
A "Tag_Input_2"	// Check whether the signal state of the operand is "1" and AND with current RLO
S "Tag_Output"	// Sets the operand to "1" if the RLO is "1"

Kuvio 9. STL-kielen esimerkki (Siemens 2021, 7975)

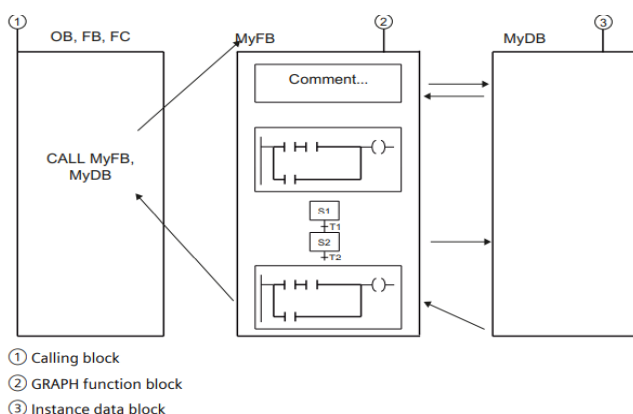
4. SCL eli Structured Control Language on tekstipohjainen PASCAL-ohjelmointikielen perustuva korkean tason ohjelmointikieli. Se mahdollistaa muun muassa erilaiset toistorakenteet ja matemaattiset funktiot. (Siemens 2021, 8023) Kuviossa 10 on yhden rivin mittainen esimerkki SCL-kielen koodista. Kyseinen koodi lukee kahden muistipaikan arvon ja kertoo ne yhteen, tallentaen ne kolmanteen muistipaikkaan.

SCL

```
"MyTag1" := "MyTag2" * "MyTag3";
```

Kuvio 10. SCL kielen esimerkki (Siemens 2021, 8027)

5. GRAPH on graafinen ohjelmointikieli, jolla luodaan sekvenssiohjelmaa. Sekvenssit suoritetaan järjestyksessä eli seuraavaan askeleeseen ei siirrytä ennen kuin edellisen askeleen määrittämät ehdot täyttyvät. (Siemens 2021, 8097.) Kuviossa 11 on esimerkki GRAPH-kielellä toteutetusta ohjelmasta.



Kuvio 11. GRAPH-kielen esimerkki (Siemens 2021, 8098)

6. CEM eli Cause-Effect-Matrix, syyseuraus-matriisi on syyseuraussuhteisiin perustuva ohjelmointikieli. (Siemens 2021, 8219.) Kuviossa 12 on esimerkki syyseurausmatriisi ohjelmasta.



Kuvio 12. CEM kielen esimerkki (Siemens 2021, 8220)

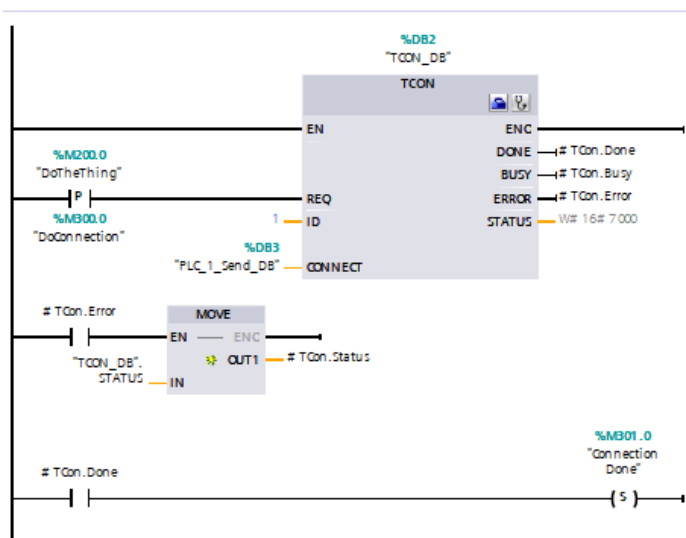
5 Sovellusohjelmointi

Projektissa käytetty logiikkaohjelma pyrittiin tekemään mahdollisimman pitkälti käyttäen SCL-ohjelmointikieltä sen kompaktin rakenteen ja helpon hallittavuuden takia. Myös toistosilmukoiden käyttötarve vaikutti ohjelmointikielen valintaan. Joitakin toimintoja, joita projekti vaati ei ollut tehokasta toteuttaa muuten, esimerkiksi tallennuslohkon läpikäynti alkio kerrallaan ja alkiodien arvojen hyödyntäminen laskennassa. Joitakin toiminnoista ei tosin pystytty tehokkaasti ja luotettavasti toteuttamaan SCL-kielillä, näissä sovelluksen osissa käytettiin LAD-ohjelmointikieltä. Tällaisia toimintoja olivat esimerkiksi anturiin yhdistäminen ja yhteyden katkaisu.

Tuotettu sovellus liitetään osaksi sahalaitoksen logiikkaa joko muokkaamalla sahalla olemassa olevaa ohjelmistoa tai lisäämällä laitokselle oma logiikkaohjain, joka hoitaa mittauksen ja anturin ohjaamisen. Tämä erillinen logiikka lähettää datan halutussa muodossa muun sahalaitoksen käytettäväksi, esimerkiksi suoraan laitoksesta vastaavalle logiikalle tai tietokantaan.

5.1 Anturiin yhdistäminen

Anturi yhdistetään logiikkaan ohjausjärjestelmän TCON-komennolla. Komennolle määritetään anturin ja logiikan IP-osoite, portti, jota anturin halutaan käyttävän tiedonsiirtoon, tiedonsiirron tyyppi ja yhteydelle yksilöllinen tunnistenumero. Nämä määrittelyt tehdään data-lohkoon, joka osoitetaan muuttujana komennolle "CONNECT". Kuviossa 13 näkyy yhdistämiseen käytettävän ohjelmalohkon käyttö logiikkaohjelmassa. Kuviossa näkyy myös ohjelmallinen kosketin, joka käynnistää yhdistämisprosessin. Kun yhteys on muodostettu ohjelma saa siitä tiedon, jonka jälkeen mittausprosessi alkaa.



Kuvio 13. Anturiin yhdistämiseksi käytetty komento LAD-kielillä

Ohjelmalle on kerrottava, kuinka yhteys anturiin muodostetaan. Se tehdään osoittamalla TCON palikalle datalohko, joka sisältää tiedot muun muassa anturin IP-osoitteesta, käytettävästä yhteys porteista sekä yhteys tyypistä. Tällaisia datalohkoja voi yhdessä ohjelmassa olla useampia kuin yksi, siksi yhteydelle on annettava uniikki tunniste numero. Tämän numeron perusteella ohjelma pystyy erottelemaan eri yhteydet toisistaan ja päättämään mihin niistä tulisi yhdistää ja mihin lähettää tai mistä lukea tietoa. Kuviossa 14 on esimerkki käytettävästä datalohkosta. Data lohkon on muodostuttava juuri näin, jotta se toimisi yhteyden muodostamiseen.

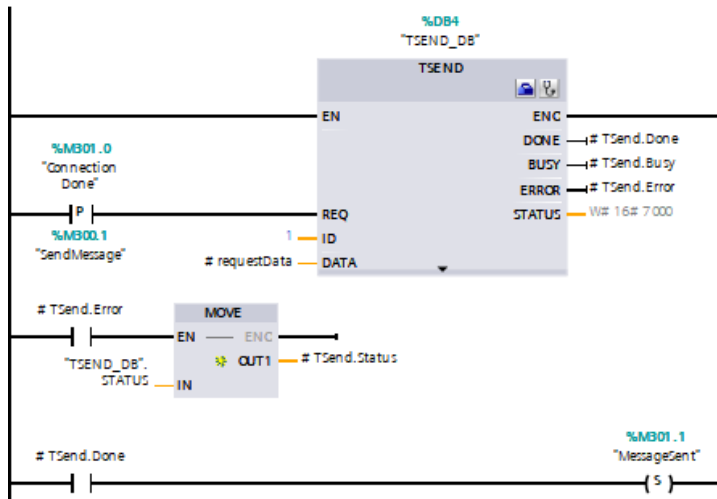
PLC_1_Connection_DB									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Si
	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Interfaceld	HW_ANY	72	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	ID	CONN_OUC	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	ConnectionType	Byte	16#0B	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	ActiveEstablished	Bool	true	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	RemoteAddress	IP_V4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	ADDR	Array[1..4] of Byte		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	ADDR[1]	Byte	192	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	ADDR[2]	Byte	168	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
0	ADDR[3]	Byte	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1	ADDR[4]	Byte	133	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	RemotePort	UInt	2112	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	LocalPort	UInt	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Kuvio 14. Anturin yhteyden määrittelyyn käytetty data lohko

Jokaisen mittauksen jälkeen yhteys anturiin katkaistaan ja muodostetaan uudelleen, kun seuraava kappale on saapumassa mittaus alueelle. Tämä tehdään, jotta voidaan varmistua datan tallentumisesta oikein.

5.2 Anturin ohjaaminen ja datan lukeminen

Anturille lähetetään käskyjä Tia Portalin sisäisellä TSEND-ohjelmalohkolla. Ohjelmalohkolle on kerrottava käytettävän yhteyden tunnistenumero ja se mitä halutaan lähettää. Ohjelmalohkolla voidaan lähettää muun muassa tieto input- ja outputpisteiden tilasta, bittimuisti tai kokonainen muistilohko. Koska anturi ottaa komennot vastaan tietyssä muodossa lähetetään sille muistilohko, johon suoritettava komento on tallennettu heksadesimaalimuodossa. Lähetettävän viestin pituus määräytyy lähetettävän komennon perusteella. Kuviossa 15 on esiteltynä TSEND-ohjelmalohko rakenne ja se, kuinka sitä ohjelmassa hyödynnetään.



Kuvio 15. Ohjaukskäskyn anturille käytetty ohjelmalohko

Anturille lähetettävä komento muodostuu sarjasta heksadesimaalimerkkejä. Merkit määräytyvät sanoman sisällön mukaan. Jokainen sanoma alkaa ”STX”-merkkijonolla, joka on käännetty vastaaviksi heksadesimaaliarvoiksi. Tämä osuus sanomasta kertoo anturille sanoman alkavan. Tätä seuraa tieto siitä, kuinka monta merkkiparia kyseinen sanoma sisältää. Tämän jälkeen anturille kerrotaan minkä tyyppinen sanoma on, eli halutaanko anturia esimerkiksi ohjata päälle vai halutaanko anturilta vastaus, esimerkiksi viimeisin mittaus tulos. Tämän jälkeen merkkijonossa sijaitsee varsinainen komento-osuus sanomasta, eli mitä tehdään. Tämän osuuden pituus ja sisältö vaihtelee käskyjen välillä. Jokainen sanoma päättyy sanoman lopettavaan kahden merkin mittaiseen tarkistusarvoon. Tämän perusteella anturi voi olla varma siitä, että saatu käsky on oikeanlainen.

Anturille komentoja voidaan toimittaa ja tietoa anturilta ottaa vastaan kahta kanavaa pitkin CoLa A ja CoLa B. CoLa A ottaa vastaan käskyjä ASCII ja heksadesimaali muodossa, kun taas CoLa B ottaa niitä vastaan binaarimuodossa. Anturin vastaukset voidaan ottaa luotettavasti vastaan vain binaarimuotoisena. Kuviossa 16 näkyy komennon rakenne ASCII merkkeinä, heksadesimaali muodossa ja binaarimuodossa.

CoLa A	ASCII	<STX>sMN[SPC]SetAccessMode[SPC]03[SPC]F4724744<ETX>
	Hex	02 73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 30 33 20 46 34 37 32 34 37 34 34 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 03 F4 72 47 44 B3

Kuvio 16. Esimerkki anturiin sisään kirjautumiskäskystä.

Osa komennoista vaatii kirjautumisen laitteelle. Myös tämä tehdään samalla tavalla kuin muutkin komennot. Ainoana erona on, että tämä komento sisältää myös määrittelyn siitä millä tilillä halutaan kirjautua ja kyseisen tilin salasanan. Alla olevassa kuviossa 17 on esillä

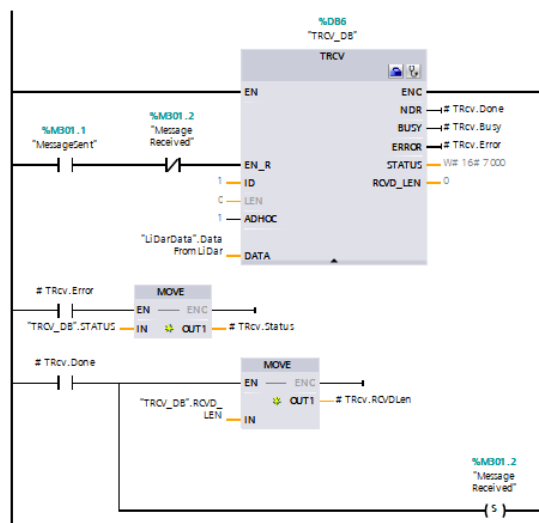
anturille lähetettävän komennon 24 ensimmäistä merkkiä, kyseinen komento pyytää anturia lähettämään viimeisimmän mittauksen datan.

requestData	Array[0..23] of Byte	
requestData[0]	Byte	16#02
requestData[1]	Byte	16#02
requestData[2]	Byte	16#02
requestData[3]	Byte	16#02
requestData[4]	Byte	16#00
requestData[5]	Byte	16#00
requestData[6]	Byte	16#00
requestData[7]	Byte	16#0F
requestData[8]	Byte	16#73
requestData[9]	Byte	16#52
requestData[10]	Byte	16#4E
requestData[11]	Byte	16#20
requestData[12]	Byte	16#4C
requestData[13]	Byte	16#4D
requestData[14]	Byte	16#44
requestData[15]	Byte	16#73
requestData[16]	Byte	16#63
requestData[17]	Byte	16#61
requestData[18]	Byte	16#6E
requestData[19]	Byte	16#64
requestData[20]	Byte	16#61
requestData[21]	Byte	16#74
requestData[22]	Byte	16#61
requestData[23]	Byte	16#05

Kuvio 17. Osa anturin vastaanottamasta ohjaus käskystä

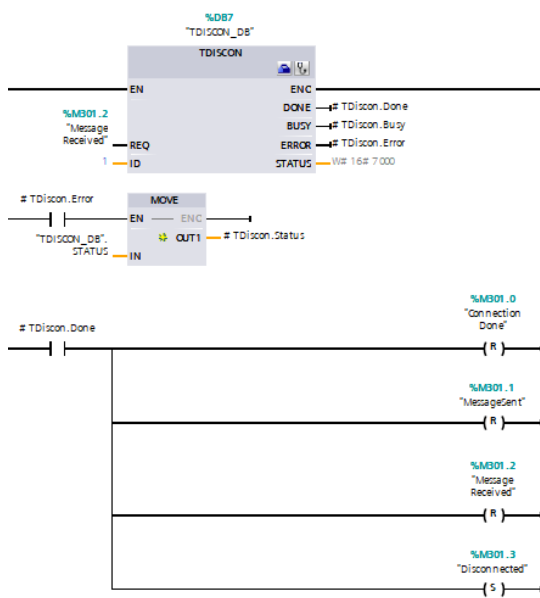
5.3 Datan prosessointi ja tallentaminen

Kun anturilta on pyydetty mittaustieto, se otetaan vastaan Tia Portalin sisäisellä TRCV-komennolla. Komennolle on määritettävä käytettävän yhteyden tunniste, jotta se tunnistaa, mistä IP-osoitteesta ottaa tietoa vastaan. Tämän lisäksi komennolle on annettava osoitin muistipaikkaan, johon vastaanotettava tieto tallennetaan. Kuviossa 18 näkyy, kuinka TRCV-ohjelmalohkoa käytetään opinnäytetyön aikana tuotetussa ohjelmassa.



Kuvio 18. Datan lukemiseen anturilta käytetty ohjelmalohko

Kun anturilta on saatu vastaus kyselyyn ja se on tallennettu, yhteys anturiin katkaistaan. Tämä tehdään TDISCON-ohjelmalohkon avulla. Kuten muissakin anturin ohjaukseen liittyvissä ohjelmalohkoissa tällekin on kerrottava mikä yhteys halutaan katkaista. Ohjelmalohkon käyttö näkyy kuviossa 19.



Kuvio 19. Yhteyden katkaisuun käytetty ohjelmalohko

Anturin lähettämä data sisältää muutakin kuin projektin kannalta tärkeän etäisyystiedon. Anturin viesti sisältää tietoja muun muassa siitä millaiseen komentoon kyseinen vastaus liittyy, kuinka paljon dataa vastaus sisältää, anturin nimen ja kellonajan. Vastauksen sisältämän tiedon määrää on mahdollista rajata, mutta yksinkertaisimmillaankin se sisältää mittauksen kannalta turhaa tietoa. Tämän takia tallennetusta tiedosta on eroteltava prosessille tärkeät tiedot. Tämä on mahdollista koska anturi lähettää datan aina samassa muodossa ja saman pituisena viestinä.

Kun prosessin kannalta tärkeä data on kerätty talteen se pitää vielä prosessoida, jotta sitä voidaan käyttää laskennassa. Anturilta saatu tieto tulee merkkijonona, joka koostuu kahden merkin mittaisista heksadesimaali arvoista. Yksi heksadesimaali koostuu 4 bitistä. Yksi etäisyys tieto koostuu neljästä heksadesimaalimerkistä, joten oli tarpeen yhdistää kaksi merkkisarjaa Tia Portalin käyttämäksi wordiksi, joka koostuu neljästä heksadesimaalimerkistä eli 16 bitistä. Yhdistämisen lisäksi merkkien järjestys piti vaihtaa oikeaksi, sillä sanomassa ne tulivat käänteisessä järjestyksessä.

Kuviossa 20 on koodi, jolla osa anturin lähettämästä datasta otetaan käsittelyyn ja muunnetaan neljästä heksadesimaali arvosta reaalityttö arvoksi. Tämä koodi suoritetaan 281 kertaa koska anturilta saadaan 281:n pisteen etäisyystieto.

```

REGION Raw_data_to_distance_from_board
  IF "Disconnected" AND "DoCalibration" = 0 THEN
    #startIndex := 85;
    REGION NormalOperation
      FOR #Count := 0 TO 280 DO
        #firstWord := "LiDarData".DataFromLiDar[#startIndex];
        #secondWord := "LiDarData".DataFromLiDar[#startIndex + 1];
        #firstWord := SHL(IN := #firstWord, N := 8);

        #Distance := #firstWord + #secondWord;
        "MeasurementData".DistanceFromBoardsReal[#Count] := #Distance * 0.1;
        #startIndex := #startIndex + 2;

      END_FOR;
    END_REGION
  END_IF;
END_REGION

```

Kuvio 20. Anturilta luetun datan muuntaminen reaalityluku arvoksi

5.4 Laskenta

Kun laskentaan tarvittava data on kerätty ja muunnettu käytettävissä olevaan muotoon eli pituusarvot heksadesimaaleista reaalityluvuiksi, on laskennan vuoro. Anturin mitaama pituus on suora etäisyys anturista kyseiseen mittapisteeseen. Tämä arvo ei kyseisessä projektissa riitä vaan sen avulla joudutaan laskemaan pisteen etäisyys anturin keskiliinjasta. Tämä tehdään ensin muuntamalla kyseessä olevan pisteen kulman arvo radiaaneiksi Tia Portalin laskentaa varten (Kaava 2).

$$\text{Radiaani} = \text{Kulma} * \frac{\pi}{180} \quad (2)$$

Jossa kulma on 90 ° ja mittauspisteen välinen kulma. Tämän jälkeen voidaan trigonometristä kaavaa käyttäen laskea pisteen etäisyys keskiliinjasta. Tämä tehdään jokaiselle mittapisteele. Kuviossa 21 on esiteltyä, kuinka mittapisteiden etäisyys anturin keskiliinjasta lasketaan.

```

IF "Disconnected" AND "DoCalibration" = 0 THEN
  FOR #ConversionIndex := 0 TO 280 DO

    #Angle := ((90 - (#AngleResolution * #ConversionIndex + #StartAngle)) * #Pii/180);
    CASE #ConversionIndex OF
      0..140:
        "MeasurementData".EachPointsDistanceFromCenterLine[#ConversionIndex] :=
          SIN(#Angle) * "MeasurementData".DistanceFromBoardsReal[#ConversionIndex];
      140..280:
        "MeasurementData".EachPointsDistanceFromCenterLine[#ConversionIndex] :=
          -SIN(#Angle) * "MeasurementData".DistanceFromBoardsReal[#ConversionIndex];
    END_CASE;
  END_FOR;

```

Kuvio 21. pisteen etäisyyden laskenta anturin keskiliinjasta

Kun pisteiden etäisyys keskilinjasta on saatu laskettua, voidaan selvittää, mikä näistä pisteistä vastaa mitattavan laudan päätä. Tämä tehdään vertailemalla pisteiden suoraa etäisyyttä anturista laudan ja kalibrointitiedon välillä. Niin kauan kuin mitattu etäisyysarvo eroaa kalibrointi-arvosta, mittapiste osuu lautaan. Kun löydetään viimeinen piste, jossa näin on, voidaan sen avulla laskea etäisyys laudan päähän ja tämän avulla laudan pituus. Mittapisteiden vertailuun käytetty koodi on esitelty kuviossa 22.

```

FOR #DifferenceCount := 0 TO 280 DO
  #CalibrationDistanceAtBeginning := TAN((90 - #StartAngle) * #Pii/180) * #BoardThickness;
  #CalibrationDistanceAtEnd := TAN((#AngleResolution * #DifferenceCount + #StartAngle - 90) * #Pii/180) * #BoardThickness;

  #Difference := ((#CalibrationData".DistanceFromCensorToBoard[#DifferenceCount] -
  "MeasurementData".DistanceFromBoardsReal[#DifferenceCount]) /
  "CalibrationData".DistanceFromCensorToBoard[#DifferenceCount] * 100);

  IF #Difference < (0.5) AND #DifferenceCount < 280 THEN
    .
    CASE #DifferenceCount OF
      0..140:
        #Angle := ((90 - (#AngleResolution * #DifferenceCount + #StartAngle)) * #Pii/180);
        "LengthOfPiece"."THIS"[#BoardNum] := "MeasurementData".EachPointsDistanceFromCenterLine[0] -
        "MeasurementData".EachPointsDistanceFromCenterLine[#DifferenceCount + 1] + #CalibrationDistanceAtBeginning;
      141..279:
        #Angle := ((#AngleResolution * #DifferenceCount + #StartAngle - 90) * #Pii/180);
        "LengthOfPiece"."THIS"[#BoardNum] := "MeasurementData".EachPointsDistanceFromCenterLine[0] +
        "MeasurementData".EachPointsDistanceFromCenterLine[#DifferenceCount] + #CalibrationDistanceAtBeginning;

    END_CASE;
    #IndexStoppedAt := #DifferenceCount;
    EXIT;
  ELSIF 0.5 < #Difference AND #DifferenceCount = 280 THEN
    "LengthOfPiece"."THIS"[#BoardNum] := 3000;
  END_IF;
END_FOR;

```

Kuvio 22. laudan pituuden laskenta

Jotta nykyinen ohjelma toimii oikein, on anturin käyttöönoton yhteydessä suoritettava kalibrointi mittausta linjastosta. Tässä mittauksessa laite on asennettuna siihen paikkaan, jossa se tulee olemaan normaalissa käytössä. Tämän jälkeen linjastolle, anturin mittausta alueen alle asetetaan kappale, joka on ohuempi kuin yksikään laitoksessa käsiteltävä kappale. Tämän jälkeen ajetaan kalibrointiohjelma. Esimerkki ohjelmasta löytyy kuviossa 23. Ohjelma on lähes identtinen ohjelmaan, jota anturi käyttää normaali tilanteessa, mutta sen tuottamat arvot tallennetaan erilliseen muistipaikkaan.

```

]REGION Raw_data_for_calibration
] IF "Disconnected" AND "DoCalibration" = 1 THEN
  #startIndex := 85;
]   REGION Calibration
]     FOR #Count := 0 TO 280 DO
]       #firstWord := "LiDarData".DataFromLiDar[#startIndex];
]       #secondWord := "LiDarData".DataFromLiDar[#startIndex + 1];
]       #firstWord := SHL(IN := #firstWord, N := 8);
]       #Distance := #firstWord + #secondWord;
]       "CalibrationData".DistanceFromCensor[#Count] := #Distance * 0.1;
]       #startIndex := #startIndex + 2;
]     END_FOR;
]   END_REGION
] END_IF;
]END_REGION
]REGION #Distance_of_each_point_from_center_line
] IF "Disconnected" AND "DoCalibration" = 1 THEN
]   FOR #ConversionIndex := 0 TO 280 DO
]     #Angle := ((90 - (#AngleResolution * #ConversionIndex + #StartAngle)) * #Pii/180);
]     CASE #ConversionIndex OF
]       0..140:
]         "CalibrationData".DistanceFromCenterLine[#ConversionIndex] :=
]           SIN(#Angle) * "CalibrationData".DistanceFromCensor[#ConversionIndex];
]       141..280:
]         "CalibrationData".DistanceFromCenterLine[#ConversionIndex] :=
]           - SIN(#Angle) * "CalibrationData".DistanceFromCensor[#ConversionIndex];
]     END_CASE;
]   END_FOR;
] END_IF;
]END_REGION

```

Kuvio 23. kalibrintidatan tallennus

6 Testaus

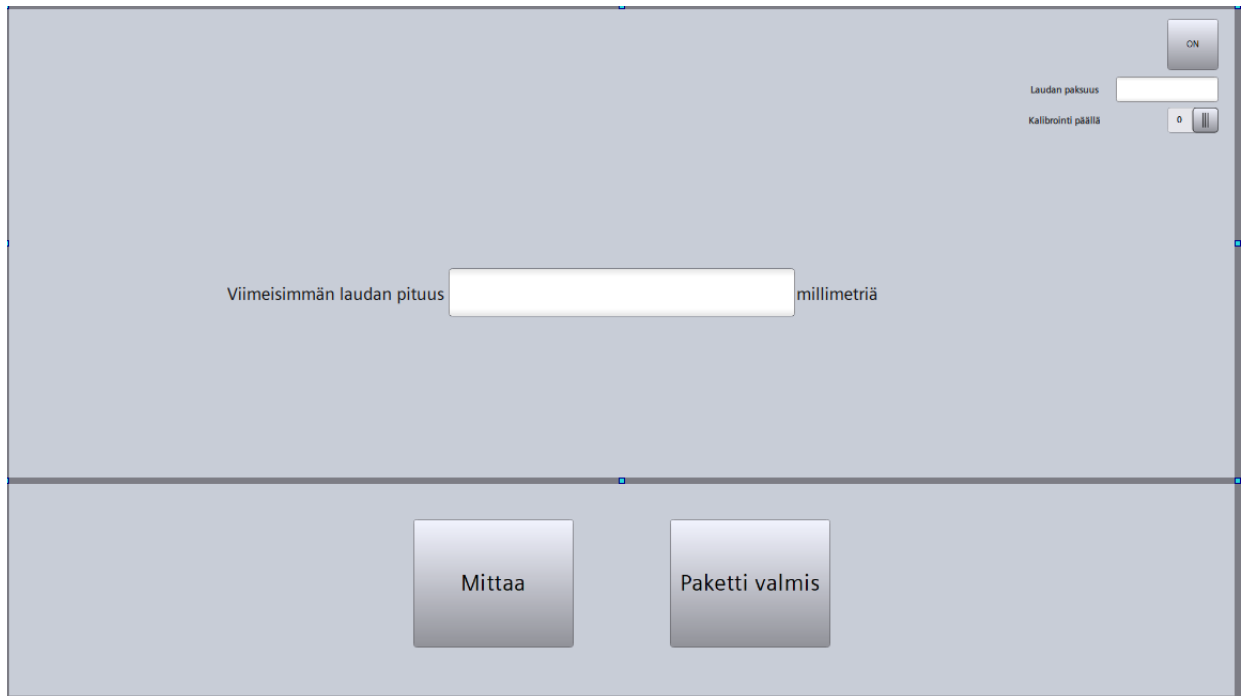
Laitteiston testaus varsinaisessa tuotantolaitoksessa johon anturi sijoitetaan ei ollut mahdollista. Tämän takia testaus suoritettiin Pronor Control oy:n tiloissa.

Mittaukset tuotettiin mittaamalla laitteella maantasossa olevia lautoja. Mittoja otettiin eri paksuisista ja pituisista laudoista. Tämä tehtiin, jotta voitiin varmistua siitä, ettei kappaleen paksuudella ole vaikutusta mittaustulokseen ja että koko mittausaluetta voitiin hyödyntää ja saada todellisuutta vastaavia tuloksia. Mittauksia otettiin jokaisesta kappaleesta useampia, sekä kappaleen ollessa paikoillaan, että sen liikkeessä mittausalueen ohi. Taulukossa 1 näkyy mittausten tulokset.

	Todellinen pituus (mm)	Mittaus paikoillaan (mm)	Ero (mm)	Ero (%)	Mittaus liikkeessä (mm)	Ero (mm)	Ero (%)
Lauta 1	710.0	696.2	13.8	1.9	697.3	12.7	1.8
Lauta 1	710.0	696.5	13.5	1.9	696.0	14.0	2.0
Lauta 1	710.0	696.3	13.7	1.9	700.9	9.1	1.3
Lauta 2	1300.0	1288.1	11.9	0.9	1283.0	17.0	1.3
Lauta 2	1300.0	1288.3	11.7	0.9	1288.1	11.9	0.9
Lauta 2	1300.0	1288.2	11.8	0.9	1288.1	11.9	0.9
Lauta 3	1448.0	1440.8	7.2	0.3	1431.0	17.0	0.7
Lauta 3	1448.0	1441.0	7.0	0.3	1431.0	16.1	0.7
Lauta 3	1448.0	1440.6	7.4	0.3	1431.6	16.4	0.7
Lauta 4	2400.0	2384.5	15.5	0.5	2383.5	16.5	0.6
Lauta 4	2400.0	2384.5	15.5	0.5	2389.0	11.0	0.4
Lauta 4	2400.0	2384.4	15.6	0.5	2384.4	15.6	0.5
Lauta 5	2900.0	2906.2	6.2	0.4	2886.4	13.6	0.9
Lauta 5	2900.0	2906.4	6.4	0.4	2887.5	12.5	0.9
Lauta 5	2900.0	2906.4	6.4	0.4	2885.9	14.1	1.0

Taulukko 1. Testimittauksen tulokset

Testausta varten toteutettiin erillinen WinCC Unified käyttöliittymä. Käyttöliittymästä muun muassa käynnistetään mittausprosessi ja määritellään mitattavan laudan paksuus. Myös kalibrointi-tila valitaan aktiiviseksi käyttöliittymästä. Kuviossa 24 nähdään käyttöliittymä, jota testaukseen käytettiin.



Kuvio 24. Testauksessa käytetty käyttöliittymänäyttö

Myöhemmässä vaiheessa, kun ohjelma ja laitteisto tulevat käyttöön tuotantolaitokselle mittausprosessi käynnistetään automaattisesti valitulla tavalla. Näitä tapoja voi olla esimerkiksi valokenno tai pulssianturi kuljettimessa, jonka alueelta anturi mittaa. Varsinainen käyttökohteeseen tuleva käyttöliittymä toteutetaan myöhemmin, kun asiakkaan tarpeet käyttöliittymästä selviävät. Myös logiikkaohjelmaa on laajennettava kattamaan asiakkaan muut tarpeet. Nykyinen ohjelma on suunniteltu ohjaamaan anturia ja laskemaan kappaleiden pituus.

7 Käyttöönottosuunnitelma

Anturin mekaanisessa asennuksessa tulee ottaa huomioon, että anturin mittauksiin käytämä valo on laservaloa. Asentaessa on varmistettava, ettei valo pääse heijastumaan linjastoa käyttävien tai alueella muuten liikkuvien ihmisten silmiin. Myös laitteen turvallisuus huoltotöiden yhteydessä on varmistettava.

Sähköisen asennuksen yhteydessä anturille on toimitettava käyttöjännite, joka on 24V. Tämän lisäksi on anturille tuotava langallinen yhteys laitoksen profinet verkkoon. Jotta anturia voidaan ohjata tekemään mittaus vain, kun mittausalueella on lauta, on linjastolle asennettava valokenno.

Anturin mekaanisen ja sähköisen asennuksen jälkeen sen IP-osoite täytyy määritellä samaan verkkoon kuin sitä ohjaava PLC. Tämän jälkeen Sickin ohjelmistoa käyttäen varmistetaan, että anturi on mekaanisesti oikeassa paikassa, jotta sillä pystytään mittaamaan haluttu alue. On myös varmistettava, että anturiin saadaan yhteys kaikkialta käytössä olevasta verkosta, myös logiikalta.

Logiikkaohjelmaan on muutettava anturiin yhteyden määrittävän datalohkon parametrit toden mukaisiksi. Tämän jälkeen on otettava kalibrointimittaus tyhjästä linjastosta. Näin saadut etäisyysarvot tallennetaan datalohkoon, jota käytetään myöhemmin varsinaisessa mittausprosessissa. Tämä tehdään, jotta logiikka pystyisi mittaustulosten erojen perusteella laskemaan kappaleen pituuden.

Anturia testatessa mitattavien kohteiden lukumäärää tietyllä aikavälillä kasvatetaan, kunnes saavutetaan tuotantomäärä. Anturia ja logiikkaohjelmaa säädetään tarvittaessa erien välillä, jotta tavoite nopeuksiin päästäisiin. Anturia ohjaavassa logiikkaohjelmistossa voidaan joutua säätämään muun muassa laskentaan käytettyjä kiinteitä arvoja, esimerkiksi anturin etäisyyttä linjastoon.

8 Yhteenveto

Opinnäytetyön toteutuksen aikana selvisi, että testattua anturia ja logiikka ohjelmistoa voidaan käyttää suunniteltuun käyttötarkoitukseen, joka on lautojen pituusmittaus rimoituslinjastolla. Anturilla testauksen aikana tuotetut mittaustulokset olivat hyväksyttävän tarkkuuden sisällä todellisista arvoista. Hyväksyttävä ero todellisen pituuden ja mitatun pituuden välillä on +- 50 mm.

Toteutuksen aikana tuli esille myös laitteen mekaanisen asennuksen tarpeet. Anturi on kiinnitettävä tukevasti linjaston yläpuolelle, oikeaan kulmaan jotta etäisyys linjaston pintaan on molemmissa mittauksen ääripäissä sama. Anturi on eristettävä linjastosta mekaanisesti, jotta sen värinä ei siirry anturiin ja vaikuta mittaustarkkuuteen.

Lähteet

GeoSlam. 2022. What is LiDar and how does it work. Viitattu 9.6.2022. Saatavissa <https://geoslam.com/what-is-lidar/>

Gjorgjoska, M. 2018. Meet LiDar, The Technology That Can Stop Wildfires Using AI. Viitattu 25.8.2020. Saatavissa [Meet LiDAR, The Technology That Can Stop Wildfires Using AI \(thethings.com\)](https://thethings.com/Meet-LiDAR-The-Technology-That-Can-Stop-Wildfires-Using-AI)

Kumar, P. 2022. What is LiDAR technology? How does LiDAR help in depth measurement? Viitattu 14.6.2022. Saatavissa: <https://www.e-consystems.com/blog/camera/technology/what-is-lidar-technology-how-does-lidar-help-in-depth-measurement/>

Notaro, von J. 2021. Was Lidar in Automotive-Anwendungen leisten kann. Viitattu 28.6.2022. Saatavissa <https://www.all-electronics.de/automotive-transportation/lidar-fuer-die-tiefenerfassung-in-automotive-anwendungen-119.html>

Pepperl+Fuchs. 2016. Product manual 2016. Viitattu 10.6.2022. Saatavissa https://files.pepperl-fuchs.com/webcat/navi/productInfo/doct/tdoct3124d_eng.pdf?v=20200320015040

Pronor Control Oy. Viitattu 2.9.2022. Saatavissa [Palvelut \(pronorcontrol.fi\)](https://www.pronorcontrol.fi/)

Riegl. 2019a. VZ-200 anturi. Viitattu 14.6.2022 Saatavissa [RIEGL - Produktdetail](https://www.riegl.com/produkte/produkte-detail/RIEGL-VZ-200)

Riegl. 2019b. DataSheet Riegl VZ-200. Viitattu 14.6.2022 saatavissa http://www.riegl.com/uploads/tx_pxriegl/downloads/DataSheet_RIEGL_VZ-200_2019-12-02.pdf

Sick. 2022. LMS4000 2D LiDar sensors. Viitattu 9.6.2022.

Saatavissa https://cdn.sick.com/media/docs/0/90/790/operating_instructions_lms4000_2d_lidar_sensors_en_im0079790.pdf

Siemens. 2021. SIMATIC STEP 7 BASIC/PROFESSIONAL V17 and SIMATIC WinCC V17. Viitattu 14.6.2022. Saatavissa <https://support.industry.siemens.com/cs/document/109798671/simatic-step-7-basic-professional-v17-and-simatic-wincc-v17?dti=0&lc=en-AE>

Siemens. 2022. CPU-1517-3 PN-DP Viitattu 16.6.2022. Saatavissa [Product Details - Industry Mall - Siemens WW](https://www.industry.siemens.com/products/details/1517-3-pn-dp)

Varis, R. 2017. Sahateollisuus. E-kirja. Kustannuspalvelut Kirjakaari Oy