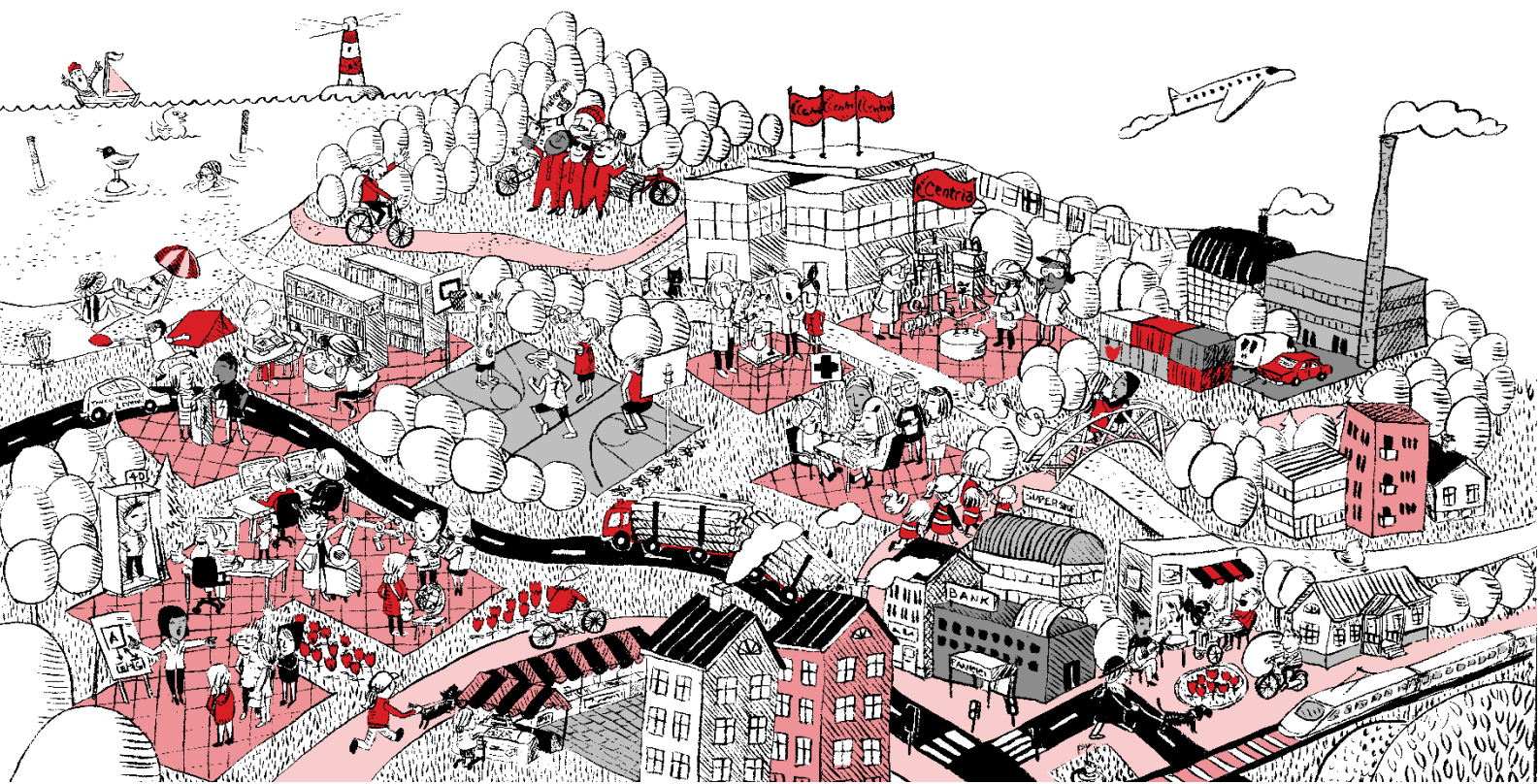


**Toni Niemi**

# **VIRTAUKSEN PARANTAMINEN TUOTANTOAUTOMAATIOILLA**

**Piirikorttien testausprosessin vakauden kehittäminen**

**Opinnäytetyö  
CENTRIA-AMMATTIKORKEAKOULU  
Tuotantotalouden koulutus  
Joulukuu 2020**



**TIIVISTELMÄ OPINNÄYTETYÖSTÄ**

<b>Centria-ammattikorkeakoulu</b>	<b>Aika</b> Joulukuu 2020	<b>Tekijä/tekijät</b> Toni Niemi
<b>Koulutus</b> Tuotantotalous		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
<b>Työn nimi</b> VIRTAUKSEN PARANTAMINEN TUOTANTOAUTOMAATIOILLA  Piirikorttien testausprosessin vakauden kehittäminen		
<b>Työn ohjaajat</b> Sakari Pieskä/Jari Kaarela	<b>Sivumäärä</b> 51 + 14	
<b>Työelämäohjaaja</b> Markku Kalliosaari		
<p>Tämä insinöörityö on tehty Sirico Electronics Oy:lle, joka on Pietarsaassa toimiva sähköteollisuuden toimialalle sijoittuva elektroniikkateollisuuden sopimusvalmistaja. Suomessa toimivassa elektroniikkateollisuudessa kustannustehokkuuden kehittäminen on keskiössä kilpailtaessa maiden kanssa, joissa tuotantokustannukset ovat edullisempia. Kustannustehokkuuden parantaminen edellyttää standardisoituja toimintamalleja, joissa tuotantoprosessit on dokumentoitu ja joissa kaikki kustannuksia aiheuttavat tuotantovaiheet ovat mitattavissa.</p> <p>Tuotantoketjut koostuvat monista eri prosesseista, jotka kaikki osaltaan vaikuttavat läpimeno-aikaan ja sitä kautta kustannustehokkuuteen. Tavoitteena oli tutkia yhden tuoteryhmän testausprosessia kvantitatiivisesti ja parantaa prosessia Lean-filosofian edellyttämällä tavalla. Tavoitteena tuoteryhmän keskeneräisten varastojen pienentäminen ja virtauksen parantaminen. Arvovirtamallin mukaisesti voidaan edetä kustannustehokkaaseen prosessiin ja standardisoimalla kehityskohteet voidaan tuoda pysyvästi lisäarvoa asiakkaalle.</p> <p>Työn tavoitteena oli kehittää LED-valaisintuotteiden testausprosessin vakautta. Tämä parantaa koko kyseisen tuotantoketjun virtausta, kun eliminoidaan turha odottelu testausprosessissa. Opinnäytetyön tutkimusosa painottui elektromekaniikan suunnitteluun ja konenäköratkaisujen kehittämiseen. Tutkimusvaiheessa kehitettiin keväällä 2019 käyttöönotettua automaattista testeriä. Testausjärjestelmän laitteita ohjataan graafisen LabVIEW-ohjelmistoympäristön kautta käyttäen TestStand-ohjelmaa testisegmenttien ajamiseen. Insinöörityössäni tutkin myös testausprosessin luotettavuuden parantamista konenäköratkaisun avulla.</p> <p>Lähtötilanteessa testausjärjestelmä oli suunniteltu täyttämään ainoastaan asiakasvaatimukset, mutta sitä ei ollut suunniteltu massatuotannon tarpeisiin. Tuotekohtaisten testausjigien avulla testausaika saatiin pudotettua neljännekseen lähtötilanteesta. Parannuksista huolimatta testausprosessi kerryttää edelleen keskeneräistä varastoa. Testausjärjestelmän jatkokehitysmahdollisuuksia tutkittiin tavoitteeksi asetetun testausajan saavuttamiseksi ja keskeneräisen varaston hallinnan parantamiseksi.</p>		
<b>Asiasanat</b> Automatisointi, konenäkö, LabVIEW, Lean		

## ABSTRACT

<b>Centria-ammattikorkeakoulu</b>	<b>Date</b> December 2020	<b>Author</b> Toni Niemi
<b>Degree programme</b> Industrial Management		
<b>Name of thesis</b> IMPROVING PRODUCTION FLOW WITH AUTOMATION  Developing the stability of circuit boards testing process		
<b>Instructor</b> Sakari Pieskä/Jari Kaarela		<b>Pages</b> 51 + 14
<b>Supervisor</b> Markku Kalliosaari		
<p>This engineering study was commissioned by Sirico Electronics Oy, a contract manufacturer from Pietarsaari, Finland, for electronics companies in the electricity sector. In Finland's electronics industry, improving cost effectiveness is the key as the companies in the sector compete with countries in which production costs are lower. Improving cost effectiveness requires standardized operating models with documented production processes.</p> <p>The production chains consist of several different processes, which all contribute to the manufacturing lead time, therefore affecting cost effectiveness. The purpose was to carry out a quantitative examination of the testing process of one product group and to improve the process in accordance with the Lean ideology. The aim was to reduce incomplete stocks in the examined product group and to improve the related flow. Value stream mapping enables development into a cost-effective process, and the standardization of development targets allows creating permanent additional value to the client.</p> <p>The work had the goal of developing the stability of a testing process for LED light products. This enhances the flow of the entire production chain as unnecessary waiting time is eliminated from the testing process. The research section of this study focused on electromechanical planning and the development of machine vision solutions. The research stage involved developing an automatic testing system deployed in the spring 2019. The testing system equipment is controlled via the LabVIEW graphical programming environment using the TestStand software for running test segments. The study also involved examining how the reliability of the testing process could be improved with a machine vision solution.</p> <p>At the start of the research process, the testing system had been designed to only meet customer requirements, and it had not been intended for mass production. Using product-specific test jigs enabled cutting the testing time to one quarter of the initial status. Despite the improvements, the testing process continues to accumulate incomplete stocks at the company. Potential further development areas in the testing system were investigated for accomplishing the testing time set as an objective and improving incomplete stock management.</p>		
<b>Key words</b> Automation, LabVIEW, Lean, machine vision		

## KÄSITTEIDEN MÄÄRITTELY

ATE	Automattinen testauslaitteisto (Automated Test Equipment)
CP	Prosessin kyvykkyys (Process Capability)
CT	Tuotantoaika, minkä tuote on työnalla (Cycle Time)
DAQ	Mittausdatan keräämiseen tarkoitettu järjestelmä (Data Acquisition)
Epästabiili prosessi	Prosessin ajautumista rakenteelliseen ylä- tai alarajaan ei voida ennustaa
FBD	Funktionaalinen lohkokaavio (Function Block Diagram)
FCT	Funktionaalinen testaus (Functional Testing)
HMI	Käyttöliittymä koneen ja ihmisen välillä (Human-Machine Interface)
Käyttöaste	Prosessin kuormitus
LCL	Alempi tilastollinen kontrolliraja (Lower Control Limit)
LED	Valoa säteilevä elektroninen komponentti (Light-Emitting Diode)
NI LabVIEW	National Instrumentin kehittämä sovelluskehitin
NI TestStand	National Instrumentin kehittämä sekvenssimoottori
PCB	Piirilevy (Printed Circuit Board)
PCBa	Kalustettu piirilevy (Printed Circuit Board assembly)

PLC	Ohjelmoitava logiikka (Programmable Logic Controller)
Pullonkaulaprosessi	Prosessin hitain vaihe, jonka käyttöaste on kovin
Sigma, $\sigma$	Perusjoukon keskihajonta
Stabiili prosessi	Tasapainossa oleva prosessi, jossa häiriöt pysyvät samana
SubVI	LabVIEW-aliohjelma
Teensy 3.0	Amerikkalaisen PJRC:n harrastelijakäyttöön valmistama kehitys alusta 32-bittisellä ARM-mikrokontrollerilla ja USB-liitännällä
Testausjigi	Mittatilaustyönä tehty asetti testattavalle tuotteelle, jonka tarkoitus on helpottaa liittimien asettelua
Testauspedit	Testattavan tuotteen ylä- ja/tai alapuolelle tuleva asetit, jotka kohdistavat vastaliittimet tuotteen liitinrajapintaan
TH	Valmistuvien tuotteiden määrä tunnissa (Throughput)
UCL	Ylempi tilastollinen kontrolliraja (Upper Control Limit)
UUT	Testattava laite (Unit Under Test)
VI	LabVIEW visuaalinen instrumentti (Virtual Instrument)
VISA	LabVIEW virtuaalinen rajapinta ohjelman ja mittalaitteen välillä (Virtual Instrument Software Architecture)
VSM	Arvovirtakuvaus (Value Stream Mapping)
WIP	Keskeneräinen tuotanto (Work in Process)

**TIIVISTELMÄ  
ABSTRACT  
KÄSITTEIDEN MÄÄRITTELY**

**SISÄLLYS**

<b>1 JOHDANTO.....</b>	<b>1</b>
<b>2 TUOTANNON LAADUNVARMISTUS.....</b>	<b>4</b>
2.1Lean-tuotantomenetelmä.....	5
2.2Vaatimustenmukaisuus.....	6
2.3Elektroniikkatuotanto.....	7
2.4Visuaalinen tarkastus.....	8
2.5Funktionaalinen testaus.....	9
2.6NI LabVIEW.....	10
2.7Konenäkö.....	13
<b>3 TESTAUSJÄRJESTELMÄ.....</b>	<b>15</b>
3.1Testauslaitteisto.....	16
3.2Chroma 62006P-100-25 DC Power.....	17
3.3NI Teststand.....	18
<b>4 TESTAUSJÄRJESTELMÄN KEHITTÄMINEN.....</b>	<b>20</b>
4.1Testausajan optimointi.....	20
4.2Tuotekohtaiset ratkaisut.....	22
4.2.1 Testerin ohjaus.....	26
4.2.2 PLC-ohjaus.....	26
4.2.3 Testerin rakennus.....	27
4.2.4 Testipetien suunnittelu ja valmistus.....	32
4.2.5 Jännitehäviöt.....	34
4.2.6 Vuotovirrat.....	35
4.3Luotettavuuden kehittäminen konenäön avulla.....	37
4.3.1 NI Vision Acquisition Software (VAS).....	37
4.3.2 IMAQdx.....	38
4.3.3 NI LabVIEW Vision Assistant Express VI.....	39
<b>5 JATKOKEHITTÄMINEN.....</b>	<b>43</b>
5.1Tasajännitelähde.....	44
5.2Data Acquisition DAQ.....	45
<b>6 TULOKSET.....</b>	<b>46</b>
<b>7 JOHTOPÄÄTÖKSET JA POHDINTA.....</b>	<b>49</b>
<b>LÄHTEET.....</b>	<b>50</b>
<b>LIITTEET</b>	

## KUVIOT

KUVIO 1. LabVIEW Block Diagram .....	12
KUVIO 2. LabVIEW VISA Block Diagram .....	18
KUVIO 3. NI TestStand sekvenssi hierarkia .....	19
KUVIO 4. 3x4 aihion johdotus .....	24
KUVIO 5. 6x2 aihion johdotus .....	24
KUVIO 6. Eriväristen LED-ryhmien ohjauksen kytkentä .....	25
KUVIO 7. Tuotteen ottama virta jännitteen funktiona .....	35
KUVIO 8. Vuotovirrat .....	36
KUVIO 9. NI Vision Assistant -lohkokaavio .....	40
KUVIO 10. Jatkokehitys version lohkokaavio .....	44
KUVIO 11. SQL-koodi .....	46

## KUVAT

KUVA 1. LabVIEW Front Panel .....	12
KUVA 2. Testauslaitteisto .....	17
KUVA 3. Yksittäisen virranmittaussegmentin viive .....	21
KUVA 4. Aihionvaihtopulssi IO-porttiin ennen tarran tulostusta .....	22
KUVA 5. PLC-ohjain .....	23
KUVA 6. PLC-ohjaimen käyttöliittymä HMI .....	27
KUVA 7. Contact Finger .....	28
KUVA 8. Tuotekohtainen testijigi .....	29
KUVA 9. Molex-liitin .....	29
KUVA 10. Ensimmäinen suunnitelma testerirungosta .....	29
KUVA 11. Ensimmäinen versio Molex-vastaliittimestä .....	30
KUVA 12. Testerirungon saranalukkomekanismi .....	31
KUVA 13. Testerirungon alumiiniosat työstettynä .....	31
KUVA 14. Testerirungon kokoonpanopiirustus .....	32
KUVA 15. Yläpetin asennettavan vastaliittimen CAD-kuva .....	33
KUVA 16. Tuotekohtainen testerirunkoon tuleva yläpeti .....	33
KUVA 17. Vision Assistant Express .....	41
KUVA 18. LED- tarkastus VI:n testaus .....	41
KUVA 19. LED check stepin asetukset TestStand:ssa .....	42
KUVA 20. LED check stepin Expression -asetukset .....	42
KUVA 21. I-MR-Ohjauskortti .....	47

## 1 JOHDANTO

Yrityksen, joka toimii valmistavan teollisuuden sopimusvalmistajana, on vastattava asiakastarpeisiin tuottamalla lisäarvoa asiakkaalle. Asiakkaalle lisäarvoa tuottavat ne palvelut, joista on sovittu. Kaikki muu on resurssien tuhlausta. Tosiasia on kuitenkin, että resursseja ei pystytä koskaan täysin hyödyntämään, sillä valmistusprosesseissa tapahtuvaa vaihtelua ei voida täysin eliminoida. Valmistusprosessin vakautta kehittämällä parannetaan prosessinkyvykkyyttä ja sitä kautta myös kustannustehokkuutta. (George, Rowlands & Kastle 2004, 14 - 15.)

Elektroniikkateollisuudessa hyödynnetään tyypillisesti massatuotantoa, jossa pyritään hyödyntämään kaikki resurssit mahdollisimman tehokkaasti, jotta kalliista tuotantokoneista saataisiin maksimaalinen hyöty. Massatuotanto edellyttää kuitenkin, ettei tuotantoketju katkea missään vaiheessa. Massatuotannossa puolivalmistetuotteita joudutaan varastoimaan eri prosessinvaiheiden väliin. Puolivalmiiden tuotteiden varastointi aiheuttaa kustannuksia, kun pääomaa on sidottu keskeneräiseen varastoon (Work in Process, WIP). Lean-filosofia ohjaa tuotantoa suuntaan, jossa kassavirta olisi koko ajan positiivinen. Tämä tarkoittaa käytännössä, toimitettavien tuotteiden myyntisaamisten realisoitumista, ennen ostolaskujen erääntymistä. Kaikista välivarastoista ei pystytä kuitenkaan täysin pääsemään eroon, sillä prosesseissa tapahtuvaa luonnollista vaihtelua ei pystytä täysin eliminoimaan. (Goldsby 2009.)

Kehitysprojektin onnistuminen edellyttää prosessiarkkitehtuurin tasojen tuntemista, sillä yksittäisen prosessin kehittäminen voi johtaa tilanteeseen, jossa tuhlataan kapasiteettia turhaan kehittämiseen (Aunola 2017). Asiakkaan tarpeisiin perustuvan arvovirtakuvauksen (Value Stream Map, VMS) luominen auttaa hahmottamaan, mikä prosessin vaihe kuluttaa tarpeettomasti resursseja (Wilson 2010, 246). Innovatiivinen prosessin parantaminen on aina parempi vaihtoehto kuin reagoiva parantaminen. Reagoivassa parantamisessa voidaan olla jo myöhässä, ja menetetyt mahdollisuudet saattavat olla merkittäviä. Tämän opinnäytetyön tavoitteena olikin parantaa olemassa olevaa testausprosessia innovatiivisesti. Parannusprojektissa hyödynnettiin jo valmiiksi rakennettua testausympäristöä. Johtuen työn luonteesta opinnäytetyössä ei käydä läpi yksityiskohtia testattavasta tuotteesta (Unit Under Test, UUT).

Elektroniikkavalmistuksen vaativat prosessit ovat pitkälti automatisoituja, koska tuotteet ovat rakenteeltaan yhä monimutkaisempia. Kun pyritään toimimaan keskeytyksettä ja nostamaan



koneiden käyttöaste mahdollisimman korkeaksi, luodaan samalla haasteita laadunvarmistukselle. Pienikin virhe prosessin alkupäässä voi nimittäin aiheuttaa kohtalokkaita seurauksia tuotantoketjun loppupäässä, mikäli virheelliset tuotteet jatkavat tuotannosta eteenpäin. Tästä syystä laadunvarmistus näyttelee merkittävää osaa elektroniikkavalmistuksessa, pitäen sisälleen valmistusprosessin aikaisen laadunvarmistuksen otannalla tehtävine tarkastuksineen.

On kuitenkin muistettava, että yksittäisen prosessin kehittäminen ei ole Lean-filosofian mukaista, vaan on syytä tarkastella koko tuotantoketjua ja sitä kautta tunnistaa, mikä prosessi vaihe on niin sanottu pullonkaulaprosessi. Pullonkaulaprosessin tunnistaminen edellyttää arvovirtamallia, jossa jokaisen tuotantoketjun eri vaiheen kustannukset ovat mitattavissa ja analysoitavissa. (Rother 2011, 24.) Tämä insinööri työ rajattiin funktionaalisen testausprosessin parantamiseen, jonka on huomattu kerryttävän ylimääräistä keskeneräisten tuotteiden varastoa (WIP) ja aiheuttavan tuotantokustannuksia, joista asiakas ei ole valmis maksamaan. Näin ollen asiakkaan näkökulmasta tämä pullonkaulaprosessi aiheuttaa pidentyviä toimitusaikoja ja valmistajan näkökulmasta virtauksen kapeikon, joka sitoo resursseja.

Epästabiili prosessi kerää keskeneräistä varastoa tuotantoprosessin väliin ja muodostaa näin kapeikon tuotantoon. Ideaalitilanteessa kassavirta pysyisi kaiken aikaa positiivisena, jolloin myyntisaamiset realisoituisivat ennen ostovelkojen erääntymistä. Littlen lain mukaan läpimenoaika korreloi suoraan keskeneräisten töiden määrään. Valmistuksessa yksi prosessi on aina pullonkaula, joka määrittää tuotantoketjun läpimenoajan. John Littlen mukaan keskeneräisten yksiköiden määrä on  $L = \lambda W$ , jossa  $\lambda$  on keskimääräinen jonoon saapumisnopeus aikayksikössä ja  $W$  on keskimääräinen jonotusaika. (Wilson 2010, 46.) Kaavasta käytetään nykyään muotoa:

$$WIP = TH * CT \quad (1)$$

tai

$$CT = \frac{WIP}{TH} \quad (2)$$

Jossa WIP on kuvaa keskeneräistä tuotantoa, jaksonaika (Cycle Time, CT) ja läpimeno (Throughput, TH) tuotetta tunnissa. Kaavan käyttäminen edellyttää, että prosessi on stabiili. CT kuvaa tuotteen keskeneräisessä tilassa olevaa aikaa WIP, sen aloituskirjauksesta valmistamiseen (Uitto 2015).

Tavoitteena oli funktionaalisen testauksen (Functional Testing, FCT) automatisointi ja konenäön hyödyntäminen ledien syttymisen toteamisessa. Tuotteiden erilaisuus toi haasteita testauksen rakentamiselle, sillä jokaiselle tuotteelle tuli rakentaa oma asetti testeriin (testausjigi). Tässä tuoteryhmässä nimittäin erilaisia testattavia tuotteita on noin kaksikymmentä. Työn lopullinen tavoite olikin testauksen saattaminen osaksi tuotantolinjaa, niin että testausjärjestelmä kykenee käsittelemään ja identifioimaan testattavat komponenttilevyt automatisoidusti. Ensin kuitenkin oli testauksen laadulliset tekijät saatettava samalle viivalle, jotta prosessi olisi stabiili.

Tämä työ on rajattu LED-valaisintuotteiden komponenttilevyjen (PCBa) funktionaalisen testausprosessin (FCT) kehittämiseen. Testaukseen kuuluu piirikortin sähköisen testauksen lisäksi ledien syttymisen visuaalinen tarkastus. Tutkimuksen kohteena olevien tuotteiden valmistus jakautuu tyypillisesti kolmeen pääprosessiin, joita ovat koneladonta, testaus ja pakkaus. Testattavat komponenttilevyt valmistetaan aihioissa, joissa voi olla 6-16 samanlaista komponenttilevyä eli tuotetta. Valmistuserät vaihtelevat 500 ja 5000 tuotteen välillä.

Työn luonteen takia tässä tutkimuksessa ei kuvata yksityiskohtaisesti hankittua NI Teststand -testiympäristöön rakennettua testausjärjestelmän toteutusta. Tavoitteena oli jatkokehittää olemassa olevaa testeriä sekä luoda samalla testausympäristöön käyttöohje insinööriyön muodossa, jota asiakasyritys voi hyödyntää tulevaisuuden tarpeisiinsa. Minulla ei ollut aiempaa kokemusta NI LabVIEW -ohjelmoinnista, joten aloitin tutustumalla National Instrumentsin graafiseen ohjelmointityökaluun. Tutkimustyön myötä tutustuin myös NI Teststand -kehitysympäristöön. Tavoitteena oli luoda virtuaalinen LabVIEW-konenäköinstrumentti, jota voidaan kutsua olemassa olevassa NI Teststand -sekvenssimoottorilla.

Käyn tässä opinnäytetyössä läpi komponenttilevyn (PCBa) valmistusprosessin, niin että lukija kykenee muodostamaan käsityksen testausprosessin kehittämisen tarpeellisuudesta. Tämän insinööriyön tutkimusosa toteutettiin kvalitatiivisena tutkimuksena, jossa pyrittiin tutkimaan ja ymmärtämään tutkimuskohteen käyttäytymistä sekä kuvaamaan sitä (Heikkilä 2014, 15).

## 2 TUOTANNON LAADUNVARMISTUS

Elektroniikkateollisuus on sähkötekniikkaan sisältyvä teollisuuden toimiala, joka suunnittelee ja valmistaa erilaisia elektronisia tuotteita ja laitekoonpanoja. Elektroniikkateollisuuden kasvu on seurausta innovaatioista, sillä yritykset käyttävät merkittäviä määriä tutkimukseen ja kehitykseen parempien tuotteiden valmistamiseksi ja tuotantoprosessin tehostamiseksi. Elektroniikkateollisuus tuottaa elektroniikkalaitteita ja kulutuselektroniikkaa sekä valmistaa sähkökomponentteja erilaisille tuotteille. (Beers 2020.) Erikoistuminen ja innovatiivisuus ovat tunnusomaista elektroniikantoolialalle, missä nopea teknologian soveltaminen ja vaativa asiakaskohdaisen piensarjavalmistus on ominaista (Peltola 2019).

Alalle tyypillisten lyhyen elinkaaren omaavien tuotteiden vuoksi organisaation prosessien ja palveluiden jatkuva kehittäminen ovat avainasemassa kilpailuedun saavuttamisessa. Elektroniikan sopimusvalmistajana yrityksen toiminta perustuu hybridistrategiaan, jossa tilaus- ja toimitusprosessi on asiakaslähtöistä imuohjausta, joka yhdistää ketteryyden (agile) ja kustannustehokkaan sarjatuotannon (Lean). Asiakasyrityksen toimintaprosessit jakautuvat seuraavasti:

1. Myyntiprosessi
2. Ostoprosessi
3. Toimittajareklamaatiot, Return to Vendor (RTV)
4. Uusien tuotteiden valmistusprosessi, New Product Introduction (NPI)
5. Muutostenhallintaprosessi, Engineering Change Order (ECO)
6. Tilaus- ja toimitusprosessi
7. Valmistusprosessi
8. Palautus- ja reklamaatioprosessi, Return Merchandise Authorization (RMA)
9. Ramp down -prosessi

Yrityksen ydinprosesseina ovat myynti ja valmistusprosessi. Valmistusprosessia palvelevina prosesseina toimivat osto- ja NPI-prosessi. Tukiprosesseina toimivat toimittajareklamaatiot (RTV), muutostenhallinta (ECO) ja palautus- ja reklamaatioprosessi (RMA). (Sirico Electronics 2018.)

Kohdeyrityksessä jatkuvan parantamisen työkaluja ovat prosessien mittaaminen, auditoinnit, reklamaatiot sekä asiakaspalautteet. Poikkeaman korjaavat toimenpiteet pyritään kohdistamaan juurisyyn. Mikäli korjaavat toimenpiteet eivät onnistu eliminoimaan poikkeamaa, juurisyitä analysoidaan uudelleen ja määritellään uudet korjaavat ja ehkäisevät toimenpiteet. Toimivaksi todettu parannusehdotus dokumentoidaan tuotteiden valmistusdokumentteihin. Samalla varmistetaan, ettei sama virhe pääse toistumaan muissa tuotteissa.

## 2.1 Lean-tuotantomenetelmä

Lean-tuotantomenetelmällä keskitytään arvotuottamiseen asiakkaalle mahdollisimman tehokkaalla resurssien käytöllä. Valmistavassa teollisuudessa pyritään lähtökohtaisesti siihen, että tuotanto on jatkuvaa eikä keskeytyksiä tule. Massatuotannossa tavoitteena on pitää kalliiden laitteiden tuottavuus mahdollisimman korkeana korkean käyttöasteen kautta. Tämä on tyypillistä massatuotannolle, mutta monesti unohtuu se tosiasia, että resursseja hukataan, kun tuotteita joudutaan varastoimaan prosessivaiheiden välillä.

Lean-ajattelun mukaan tuotantoprosessi on niin vahva kuin sen heikon lenkki. Tämä tarkoittaa käytännössä sitä, että Lean-tuotannosta pyritään saamaan niin luotettava, että ylimääräistä välivarastointia ei tarvita. Lean-mallin mukainen ratkaisu lisää kustannustehokkuutta, mutta edellyttää myös, että prosessiosat ovat stabiileja ja prosessissa ilmenevien ongelmien korjaamiseksi on standardisoitu käytännöt, jolla häiriöt saadaan eliminoitua ja niistä pystytään oppimaan. Vuosina 1891-1967 eläneen fyysikko Walter A. Shewhartin mukaan vaihtelu voidaan jakaa ilmenemismuodon mukaan yleiseen (Common Cause Variation) ja erityisyysvaihteluun (Special Cause Variation). Yleisen syyn vaihtelu on luonnollinen osa stabiilia prosessia. Eriyisyyistä johtuva vaihtelu on odottamatonta, ja se johtuu epätavallisista tapahtumista, joita ei ole kyetty ennustamaan. Ei-stabiilin prosessin tunnistaa erityisyistä johtuvista poikkeamista. (Shewhart 1924.)

Stabiilissa prosessissa vaihtelu on pientä ja vaihtelu jakautuu odotetusti, kun taas epästabiilissa prosessissa vaihtelu on epätarkkaa ja epätasaisesti jakautunutta. On tärkeää pyrkiä eliminoimaan erityisyistä johtuvaa vaihtelua, jota esiintyy epästabiilissa prosessissa. Yleisen ja erityisyysvaihtelun erottaminen vaatii asiantuntemusta, sillä erityisyysvaihtelun eliminointi parantaa prosessin kyvykkyyttä. Eliminoimalla yleisyysvaihtelua voidaan tilannetta viedä

huonompaan suuntaan. Luonnollista prosessivaihtelua säätämällä vaihtelu todennäköisesti ennemminkin lisääntyy kuin pienenee. Tätä kutsutaan ylikorjaukseksi. Epästabiliin vaihtelun tunnistaa parhaiten tilastollisilla menetelmillä, missä vaihtelusta piirrettävään kuvaajaan asetetaan ohjausrajat ja erityissyytauluun kirjataan ohjauskortista havaitun ohjausrajan ylittävät havainnot analysointia varten. (Karjalainen & Karjalainen 2002.)

Lähtötilanteessa tutkittavaa testausprosessia ei ole suunniteltu massatuotantoa varten ja sen jaksonajassa tiedetään olevan eniten vaihtelevuutta ja sen tiedetään kerryttävän keskeneräistä tuotantoa. Näin ollen sitä pidetäänkin pullonkaulaprosessina kyseisessä tuotantoketjussa. Kyseinen testausprosessi sitoo yhden työntekijän ja on työnluonteeltaan yksipuolista, joten sen vaikutus virtaukseen on ollut ilmeistä. Mittaamalla ja analysoimalla kokonaisläpimenoaika ja keskeneräisten varastojen muodostumista voidaan osoittaa, että kyseisen prosessin kehittämisellä voidaan parantaa tuotannonvirtausta.

## **2.2 Vaatimustenmukaisuus**

Prosessien kehittämisellä ja myytävien palveluiden standardisoinnilla luodaan myytäviä palvelukokonaisuuksia. Dokumentoidun laatupolitiikan sekä laatujärjestelmädokumenttien avulla laatujärjestelmä tukee vahvasti organisaation johtamista ja kehittämistä sekä tuotteiden ja palveluiden jatkuvaa parantamista. Kohdeyrityksen laatujärjestelmä pohjautuu ISO9001:2015-standardiin, joka antaa pohjan arvioida organisaation kykyä täyttää asiakasvaatimukset ja lakisääteiset vaatimukset. Tuotteet valmistetaan asiakkaan toimittamien valmistusdokumenttien mukaan ja poikkeama katsotaan olevan silloin, jos se on vastoin laatujärjestelmässä kuvattua toimintamallia tai itse tuote tai tuotteen valmistusmenetelmä ei vastaa sille määritellyjä vähimmäisvaatimuksia.

Association Connecting Electronics Industries (IPC) on maailmanlaajuinen organisaatio, joka tarjoaa standardeja valmistavaan elektroniikkateollisuuteen kilpailukyvyyn ja taloudellista menestyksen takaamiseksi. IPC-A-610 on hyväksymisstandardi, joka määrittää visuaalisen tarkastuksen vaatimukset komponenttien liitoksille. J-STD-001 määrittelee vaatimukset juotoksille ja prosesseille. IPC-A-610 standardi ottaa kantaa ainoastaan valmiin tuotteen visuaalisen tarkastuksen vaatimukseen. Vaatimukset jaetaan IPC-standardissa kolmeen luokkaan:

**Class 1** – Yleiset elektroniikkalaitteet

Tuotteet, joiden tärkein vaatimus on valmiin kokoonpanon toiminta. Häiriötön jatkuva toiminta ja pitkä käyttöikä eivät ole näillä tuotteilla vaatimuksena.

**Class 2** – Erikoispalveluihin tarkoitetut elektroniset tuotteet

Tuotteet, joissa tarvitaan jatkuvaa suorituskykyä ja pitkää käyttöikää. Keskeytymätöntä toiminta on vaatimuksena, mutta se ei kriittistä. Tyypillisesti loppukäyttöympäristö ei aiheuttaisi vikatilanteita.

**Class 3** – Suuren suorituskyvyn elektroniikkalaitteet

Tuotteet, joiden jatkuva toiminta on vaatimuksena. Loppukäyttöympäristö voi olla epätavallisen vaativa ja toimintahäiriöitä ei sallita, kuten elintoimintoja ylläpitävä laitteisto tai ilmailu- ja sotilaselektroniikka.

Vaikka tuote täyttäisikin luokan 3 (Class 3) visuaaliset vaatimukset, ei se automaattisesti ole luokan 3 tuote, sillä tuotteen tulee täyttää IPC-laatuoluokitukset piirilevyn suunnittelusta materiaaleihin. (IPC-A-610F. 2014.) Laatuoluokan määrittely on asiakkaan vastuulla.

## 2.3 Elektroniikkatuotanto

Komponenttitasolla elektroniikan valmistus alkaa ohutlevykalvon kasvatuksella atomikerroksittain tai optisen litografian avulla, jossa johdinkuvio siirretään esim. piikiekon pintaan (Mack 2010). Komponentit testataan ja pakataan keloihin tai putkiin, joissa ne toimitetaan komponenttilevyjä valmistavaan yritykseen.

Piirilevyt (Printed Circuit Board, PCB) valmistetaan yleensä lasikuituvahvistetusta epoksilaminoidusta materiaalista, jonka pintaan on kiinnitetty kuparifolio epoksiliimalla. Piirilevy toimii alustana komponenteille, yhdistäen komponentit kuparifolioiden avulla. Yleisesti käytetyin piirilevy tyyppi on FR4, jossa FR viittaa NEMA-luokiteltuun paloluokkaan. Joustavat piirilevyt valmistetaan polyamidista. Alumiinista valmistettu piirilevy johtaa tehokkaasti lämmön piirilevyyn juotettujen komponenttien alta. Piirilevy voi olla yksi- tai kaksipuoleinen, ja piirilevyissä voi olla lisäksi jopa 40 johdotettua välikerrosta.

Piirilevylle asennettavat komponentit ladotaan ja juotetaan paikoilleen tuotantolinjassa. Pintaliitoslinjaan kuuluvat tyypillisesti pastanpaino- ja ladontakoneet sekä juotosuuni. Komponentit juotetaan reflow-pastajuotosmenetelmällä, jota käytetään pääasiassa pintaliitoskomponenttien pehmytjuotosten tekemiseen. Juotospasta on juotosjuoksutetta sisältävä metallijauheseos, joka muuttuu juoksevaan tinamuotoon 200-250 °C:een lämpötilassa. Juotospasta levitetään piirilevyn pintaan stensiilin avulla. Stensiili on tuotekohtainen laserleikattu 100-150um paksu teräslevy, jota käytetään elektroniikkatuotannossa pastanpainoon. Stensiilissä on neliskanttisia reikiä piirilevyn juotospädiä kohdalla. Stensiili voi olla osittain ohennettu etsaamalla siten, että tietyt komponentit saavat tarvittaessa paksumman kerroksen juotospastaa. Ladontarobotti lataa komponentit pastan päälle. Ladontarobotin nopeus vaihtelee 15000-60000 komponenttiin tunnissa. Piirilevy voidaan suunnitella myös siten, että toisen puolen pintaliitettävät komponentit juotetaan yhdessä läpiladottavien komponenttien kanssa aaltojuotosprosessissa. Tällöin pintaladottavat komponentit kiinnitetään reflow-uunissa kovettuvan liiman avulla omille paikoilleen.

Komponenttilevyn tarkastuksen jälkeen mahdolliset läpiladottavat komponentit ladotaan paikoilleen ja juotetaan selektiivisesti tai aaltojuotosprosessissa. Aaltojuotoksessa sula juotosmetalli pumpataan suuttimesta ylöspäin piirikortin kulkiessa tina-aallon yli. Kapillaari-ilmiön seurauksena juotosmetalli nousee läpikuparoitujen reikiä kautta myös piirilevyn pintapuolen juotosalueille. Valmis komponenttilevy testataan asiakasvaatimusten mukaisesti ja lopuksi yksittäiset komponenttilevyt irrotetaan aihioista ja pakataan asiakasvaatimusten mukaisesti.

Asiakkaan niin halutessa komponenttilevy suojapinnoitetaan testauksen jälkeen. Suojapinnoitus voidaan tehdä kastamalla, ruiskuttamalla tai levittämällä. Suojapinnoitusaine ruiskutetaan piirikortin pintaan käsin tai pinnoitusrobotilla. Selektiivisessä pinnoituksessa pinnoitusmateriaalia levitetään ainoastaan halutuille alueille. Pääsääntöisesti piirikortin kiinnityspisteet ja liittimet jätetään vapaaksi pinnoitteesta. Tämän lisäksi on huomioitava asiakkaan erityisvaatimukset.

## **2.4 Visuaalinen tarkastus**

Kalustettu komponenttilevy tarkastetaan visuaalisesti, jotta se täyttää asiakasvaatimukset. Tarkastajan tulee tietää, mikä IPC-laatuokka on vaadittu. Lisäksi tarkastuksessa on todettava,

että oikeat komponentit ovat oikeilla paikoillaan ja oikein päin. Löydetyt poikkeavuudet dokumentoidaan ja korjataan. Puhutaan tarkistuksesta, kun korjaus tapahtuu tarkastuksen yhteydessä. Kun tarkastettavia tuotteita on tuhansia, on ihmissilmällä mahdotonta tarkastaa jokaisen komponentin oikeellisuus. Visuaaliseen tarkastukseen käytetään pääsääntöisesti kahta menetelmää, jotka ovat:

**MVI** – Manual Visual Inspection.

IPC-standardi määrittelee suurentavien optisten laitteiden käyttöä koskevat vaatimukset sekä juotoksien ja komponenttien asemointia koskevat vaatimukset. Tarkastuksen apuna voidaan käyttää kameraa tai skanneria, jolloin kuvaa on helpompi analysoida tietokoneen näytöltä. Kuvan analysointiin on olemassa ohjelmistoja, joilla otettua kuvaa voi verrata referenssikuvaan.

**AOI** – Automated Optical Inspection.

Automaattisessa tarkastuslaitteistossa otettu kuva tarkastetaan automatisoidusti. Automattinen tarkastuskone voi olla linjatyyppinen, jolloin se voidaan kytkeä osaksi tuotantolinjaa.

Jos asiakasvaatimuksena on, että juotoksien täytyy täyttää luokan 3 (Class 3) vaatimukset, joudutaan joissain tapauksissa suorittamaan röntgentarkastus, jotta nähdään juotoksien, komponenttien ja piirilevyn sisään. Kalustetun komponenttilevyn vaatimustenmukaisuus tarkastetaan First Article Inspection -menettelyllä (FAI), jossa valmistuserän virheettömyys todetaan ensimmäisistä tuotantolinjalta valmistuneista tuotteista. Tällä varmistetaan, että tuotantolinjaan on asetettu oikeat komponentit ja ladontaohjelmat ennen varsinaisen sarjatuotannon aloittamista.

## **2.5 Funktionaalinen testaus**

Funktionaalilla testauksella pyritään mallintamaan reaali maailman olosuhteita, mitattaessa samanaikaisesti eri suureita. Funktionaaliossa testauksessa tuotteen toimintofunktiota simuloidaan ja pyritään selvittämään, mitä tuotteen sisällä tapahtuu normaali- ja häiriötilanteissa. Funktionaalista testausta varten piirilevyssä saattaa olla mittapisteitä ja/tai liittimiä, joita käytetään ainoastaan funktionaalista testausta varten. Funktionaalisen testin ajaksi tuotteeseen voidaan ladata erillinen testausohjelma, jolla pystytään paremmin mittaamaan ja simuloimaan eri toimintoja.



Kun tarvittavat piirilevyn sisäiset toiminnot on todettu olevan kunnossa, suoritetaan toiminnallinen testaus. Tämä tapahtuu tuotteen todellisen liitäntärajapinnan kautta. Toiminnalliseen testaukseen saattaa kuulua muitakin puolivalmisteita, joiden kuuluu toimia yhdessä testattavan tuotteen kanssa. Toiminnallinen testaus voi pitää sisällään myös olosuhdemittaukset, jossa tuote altistetaan ympäristönvaihteluille, kuten lämpötiloille, minkä jälkeen testaus suoritetaan uudelleen. Toiminnallinen testaus on yleensä osittain tai kokonaan automatisoitu (Automated Test Equipment, ATE). Testausympäristö on rakennettava niin, että työntekijän on suoritettava testaus määrättyssä järjestyksessä. Näin kaikki testattavat toiminnot saadaan verifioitua. Testin aikana työntekijän on seurattava ohjeita ja tehtävä tarvittavat toimenpiteet ohjeiden mukaan. Tyypillisesti tuote identifioidaan verifioinnin jälkeen samalla kun vaaditut mittaustulokset tallentuvat tietokantaan.

Monimutkaisen tuotteiden kohdalla ei ole useinkaan taloudellisesti kannattavaa testata kaikkia mahdollisia toimintoja tuotteesta. Jos esimerkiksi eri toiminnot käyttävät samoja liitäntärajapintoja, riittää, että valitaan yksi toiminto, jolla kyseisen liitännän toiminta varmistetaan. On myös muistettava, että jos toiminnalliseen testaukseen kuuluu paljon peräkkäisiä työntekijän vastuulla olevia tarkastuksia, ei voida olettaa ihmisen keskittymiskyvyn riittävän kaikkien poikkeavuuksien havaitsemiseen. Testausympäristöön voidaan toiminnallisen testauksen tueksi rakentaa erilaisia konenäkö ratkaisuja, joilla varmistetaan, että poikkeamat huomataan.

Toiminnallinen testaus mittaa piirikortin ottaman virran ja vertaa niitä asiakkaan antamiin ohjearvoihin. Asiakasvaatimuksena on, että jokainen komponenttilevy identifioidaan omalla QR-koodin sisältävällä tarralla, josta selviää tuotteen sarjanumero ja muut asiakkaan vaatimat tiedot. Jokainen komponenttilevy testataan asiakkaan määrittelemillä jännitteillä, ja on tarkistettava, pysyykö virta-arvo sallituissa rajoissa. NI Teststand tallentaa mittaustulokset tietokantaan asiakkaan vaatimalla tavalla.

## **2.6 NI LabVIEW**

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) on National Instrumentin kehittämä sovelluskehitin, jonka avulla graafisesta koodista käännetään konekielinen ohjelma. LabVIEW-ohjelma muodostuu sovelluksista, joita kutsutaan virtuaalisiksi instrumenteiksi

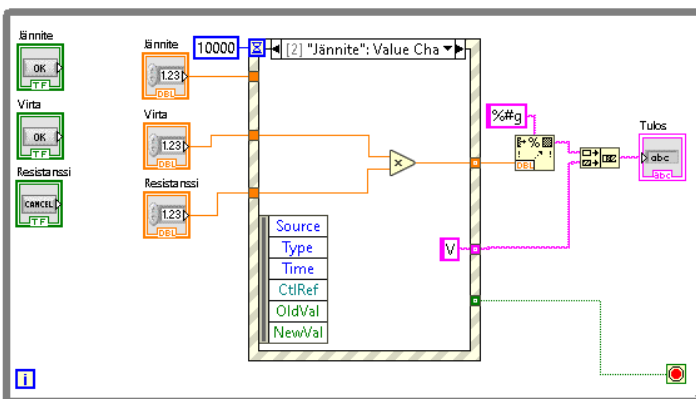
(VI). Virtuaaliset instrumentit ovat ohjelmallisesti luotuja mittalaitteita, joilla voidaan ohjata todellisia reaali maailman mittalaitteita tietoliikenneväylän kautta. Virtuaalinen instrumentti voi suorittaa myös pelkkiä datankäsittelyoperaatioita, ohjaamatta mitään ulkoista laitetta. LabVIEW-ohjelmistossa on myös työkalut konenäköratkaisuja varten, jolloin järjestelmään kytketty kamera voi välittää kuvan testattavasta tuotteesta ja ohjelma analysoi kuvasta havaittuja piirteitä. Virtuaalisia instrumentteja voidaan käyttää itsenäisinä aliohjelmina tai todellisten reaali maailman laitteiden kontrollointiin, datan keräämiseen sekä kerätyn datan analysointiin.

Tutkimuksen kohteena olevalle tuoteryhmälle asiakasvaatimuksena on, että tuotteet testataan ja identifioidaan tuotekohtaisten määrittelyiden mukaisesti (LIITE 1). Insinööryöni aloitushetkellä oli otettu käyttöön LabVIEW-ohjelmiston ympärille rakennettu testauslaitteisto, joka kykenee mittaamaan tuotteen asiakasvaatimusten mukaisesti. Työssä tutustuin National Instrumentin LabVIEW-ohjelmistoon ja TestStand-sekvenssimoottoriin. TestStand hyödyntää LabVIEW-ohjelmalla luotuja virtuaalisia instrumentteja (VI).

Virtuaalinen instrumentti toimii samaan tapaan kuin mikä tahansa Labview ympäristöön kytketty mittalaite. Virtual instrumentille annetaan tarvittavat parametrit ja tavoitearvot tuotekohtaisessa testissä. Parametrejä ovat erilaiset kuvaan tehtävät suodatukset, ja tavoitearvo on syttyvien ledien määrä. Virtuaaliset instrumentit ovat kuin ohjelman sisäisiä mittalaitteita, joita pyydetään tekemään tietty toiminto ja pyydetessä ne palauttavat mitatun muuttujan arvon. VI:llä pystytään mittaamaan reaali maailman suureita ja tulokset saadaan numeeriseen muotoon. VI:t tekevät ohjelmoinnista suoraviivaista, koska kerran toimivaksi todettuja VI:tä voidaan kutsua missä tahansa kohtaa ohjelmaa. VI-nimi tulee siitä, että instrumenttien ulkonäkö ja toiminnot on rakennettu mukailemaan mahdollisimman tarkasti reaali maailman laitetta. (National Instruments 2006.)

LabView VI koostuu kahdesta objektista: lohkokaaviosta (block diagram) ja etupaneelistä (front panel). Etupaneelistä näkee visuaalisesti, mitä lohkokaaviossa tapahtuu. VI koostuu kolmesta osasta, jotka ovat: etupaneeli, lohkokaavio, liityntäpaneeli. LabVIEW ohjelmointi jakautuu kahden rinnakkain toimivaan visuaalisen käyttöliittymään. Ohjelmointi koostuu etupaneelistä ja G-kielisestä lohkokaaviosta sekä liityntäpaneelistä. Käyttöpaneeliin lisättävät ohjaimet ilmestyvät diagrammiin, jossa niitä voidaan johdottaa ja lisätä erilaisia operaatioita signaalin käsittelyyn. Lohkokaaviossa käytettävät objektit ovat terminaaleja (terminals), noodeja (nodes) ja funktioita (functions). Objektit yhdistetään toisiinsa johtimille (wires).

Tutustuakseni LabVIEW-ohjelmointiin suunnittelin yksinkertaisen sähköisten perussuureiden laskurin. Laskuri on toteutettu EVENT-rakenteella, joka antaa tuloksen (number) ja mittayksikön merkkijonon (string) muodossa. Timeout-tapahtumasta johdotetaan Boolean-arvo while loop-silmukan toistoehtoon. Event-rakenteessa on jokaiselle tapahtumalle (case) oma numeerinen laskutoimitus sekä suureen määrittelevä merkkijono (string). Format Into String -funktiolla syötetään merkkijonon tyyppi (format String). Syntaksin (%#g) avulla merkkijonosta poistetaan ylimääräiset nollat desimaalista (KUVIO 1).



KUVIO 1. LabVIEW Block Diagram

Etupaneeliin lisätyt terminaalit ilmestyvät automaattisesti lohkokaaevioon. Lisäksi lohkokaaevioon voidaan lisätä vakio(constant), jotka eivät näy etupaneelissa. Lohkokaaevio sisältää etupaneelin graafisen esityksen lähdekoodin. Etupaneelin graafisten hallintalaitteiden avulla käyttäjä voi syöttää tietoja käynnissä olevaan virtuaaliseen instrumenttiin tai poimia tietoja siitä (KUVA 1). Etupaneeli voi toimia myös ohjelmallisena käyttöliittymänä. Siten virtuaalinen instrumentti voidaan joko ajaa omana sovelluksenaan tai lisätä se noodina toiseen lohkokaaevioon, jolloin käytetään nimitystä SubVI. (NI 2020.)



KUVA 1. LabVIEW Front Panel

LabVIEW:ssä muuttujat jaetaan paikallisiin (local) ja globaaleihin (global) muuttujiin ja niitä voidaan lukea tai kirjoittaa. Paikallisia muuttujia käytetään VI:n sisällä, kun taas globaaleja muuttujia VI:en tiedostojen väliseen tiedonsiirtoon. Yksi LabVIEW:n etu muihin kehitysympäristöihin verrattuna on laaja tuki instrumentointilaitteiden käytölle. LabVIEW pystyy tallentamaan tulokset melkein pä mihin tahansa muotoon, joista tyypillisimpiä on tekstitiedosto, taulukko ja tietokanta.

## 2.7 Konenäkö

Konenäköjärjestelmässä kameralla kuvattua kohdetta analysoidaan kuvankäsittelyohjelman avulla. Järjestelmän pääkomponentit ovat kamera, valonlähde ja tietokone. Kuvankäsittelyohjelman päätöksentekoa algoritmin avulla konenäköjärjestelmä kykenee tekemään itsenäisiä päätöksiä. Konenäön avulla pyritään vähentämään tarpeetonta laadunvaihtelua rutiininomaisissa tarkastustehtävissä. Sen avulla voidaan myös parantaa prosessin läpimenoaikaa laadusta tinkimättä.

Valaistus ja kameran optiikka valitaan siten, että tarkastettavat piirteet erottuvat kuvasta mahdollisimman selkeästi. Kuvankäsittelyn tarkoitus on suodattaa kuvasta pois kaikki sellaiset ominaisuudet, joita ei tarvita kuvan analysointia varten, ja toisaalta korostaa analysoitavia piirteitä kuvasta. Kuvankäsittelyohjelman avulla etsitään kuvasta raja-arvoja, joille on määritelty tietyt kynnsarvot. Kynnsarvojen avulla voidaan erottaa tummien ja vaaleiden pikselien rajakohdat, joita voidaan käyttää kuvioiden etsintään tai mittaamiseen. Kuvioiden tunnistamisen avulla voidaan lukea tekstiä ja viivakoodeja kohteesta. Kuvankäsittelyn avulla löydettyjä muotoja voidaan verrata referenssikuvaan tai tavoitemittoihin. Löydettyjä muotoja voidaan suodattaa siten, että analysoidaan vain tietyn kokoisia ja muotoisia kuvioita.

Kameran tavoitteena on digitoida kuva, joten kameraksi käy periaatteessa mikä tahansa edullinen digitaalinen järjestelmäkamera tai web-kamera, jossa on suora kuvansiirto-ominaisuus (direct-show compliant). Täytyy kuitenkin muistaa, että teollisuuskamerat on suunniteltu kuvadatan analysointia varten, kun taas tavanomaiset digikamerat kuvien tallentamista varten. Teollisuuskamerat omaavat suuremman kuvanottonopeuden sekä ovat rakenteeltaan vankempia. Niillä toistettavuus on omaa luokkaansa ja tekninen tuki on parempi. Teollisuuskameroissa

kuvapikseleitä on vain tarvittava määrä ja monissa sovelluksissa riittää, että kuva on mustavalkoinen. Tämä mahdollistaa jopa 60 kuvan ottamisen ja analysoinnin sekunnissa.

Älykaderoissa kamera ja tietokone on yhteen rakennettu, mikä yksinkertaistaa järjestelmän rakennetta. Älykamerat ovat kompakteja ja sopivat hyvin teolliseen ympäristöön, jossa mittaukset ja ohjattavat sovellukset ovat tarpeeksi yksinkertaisia. Älykaderoilla voi tunnistaa muotoja sekä lukea tekstiä ja viivakoodeja. Älykaderoissa on digitaalisia lähtöjä sekä liitäntä ohjelmointia varten. PC-pohjaisessa systeemissä kuvan analysointi tapahtuu tietokonesovelluksessa. PC-pohjaisella konenäköjärjestelmällä pystytään tekemään vaativampia analyyssejä kuville, ja liitännät muihin digitaalisiin sovelluksiin ovat paremmat. Ohjelmointi on monesti vaikeampaa. Insinööriyön konenäkö tutkimuksissa käytin Microsoftin 720p HD Web -kameraa, joka liitetään tietokoneeseen USB 2.0 -väylän kautta. Kameran resoluutio on 1920 X 1080 ja kuvanottotaajuus on 30 kuvaa sekunnissa.

### 3 TESTAUSJÄRJESTELMÄ

Vuoden 2019 keväällä hankittua NI LabVIEW -testausjärjestelmää hankittaessa kiinnitimme suurimman huomiota siihen, että se täyttää asiakasvaatimukset testauksen osalta. Testausjärjestelmän rakenteesta johtuvista syistä se oli hidas ja testaaminen työlästä, sillä tuotteet täytyi testata yksitellen mittajohtoja käyttäen. Testausympäristö on rakennettu LabVIEW-ohjelmiston ympärille, joka tallentaa testituloksen tietokantaan identifiointitarran tulostuksen jälkeen. Asiakasvaatimuksena on, että tuotteet testataan määrättyillä testijännitteillä ja tuotteen ottama virta mitataan ja tallennetaan vastaavalla tuotteen sarjanumerolla tietokantaan. Lisäksi asiakasvaatimuksena on, että tuotteessa olevien ledien syttyminen tarkastetaan tietyillä nominaalitestijännitteillä, jotka ovat tyypillisesti 12, 24 ja 48 voltia.

Asiakasvaatimuksena tutkittavalle testausprosessille oli, että jokaisella testijännitteellä piirikorttien ottama virta on ohjearvojen välissä. Asiakas oli toimittanut jokaisen tuotteelle virtarajat. Testijännitteet vaihtelevat 9-110 voltin tasajännitteen (DC) välillä ja virtarajat välillä 0-10 ampeeria (A). Testausdokumentista ilmenee myös identifiointitarran sijainti ja jännitteet, joilla ledien syttyminen on tarkastettava. Asiakasvaatimuksena on lisäksi, että ID-tarra tulee liimata ainoastaan hyväksytysti testattuihin tuotteisiin. Lisäksi on määritelty, että mittaustulokset tulee tallentaa tietokantaan, ne on säilytettävä tietyn ajan ja niiden on oltava asiakkaan saatavilla pyydettyäessä. Liitteenä esimerkki testausdokumentista (LIITE 1). Mittausparametrit tarkastuksessa noudatetaan kaavaa:

$$I_{low} \leq I_{meas} \leq I_{high} \quad (3)$$

Jossa  $I_{low}$  on alaraja- ja  $I_{high}$  on yläraja tuotteen ottamalle virralle. Virranmittaus algoritmin tulee antaa hyväksyttäväarvo (passed), kun mitattu virta-arvo jää määrättyjen virtarajojen väliin tai on yhtä suuri kuin määritelty low- tai high-virtaraja. Labview TestStand -ympäristössä numeerinen vertailu, jossa mitatun suuretta vertaillaan annettuihin low- ja high- rajojen väliin tapahtuu GELE (>=<=)-tyyppisellä vertailuoperaatio-syntaksilla.

Lisäksi jokaiselle tuotteelle on määritelty estosuuntainen virta, kun jännitteen polariteetti on kytketty vastasuuntaiseksi. Tuote on suunniteltu siten, että estosuunnassa se ei saa ottaa enempää virtaa kuin mitä itse tuotteen suojaapiirin ottama virta on. Lisäksi teholähde tarvitsee

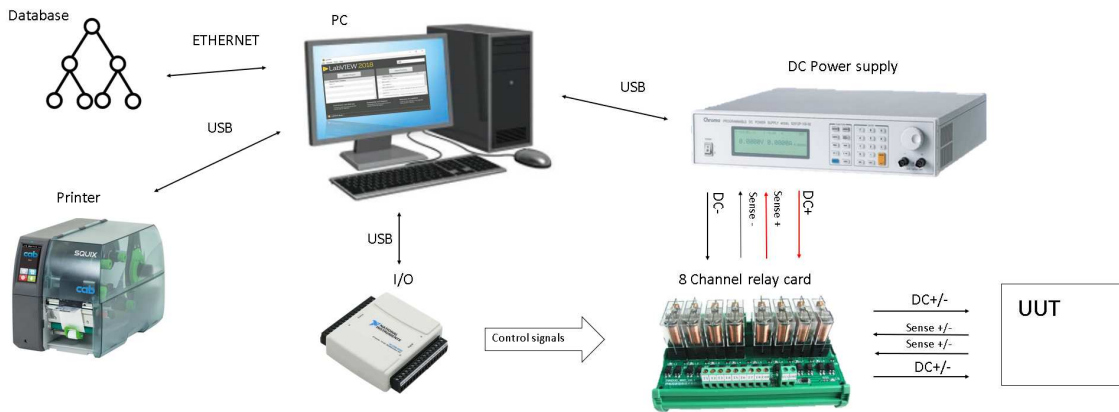
virtarajan (Current Limit), joka rajoittaa tuotteen ottaman maksimivirran. Maksimivirranrajoitus on tärkeä asettaa oikealle tasolle, jottei vialliseen tuotteeseen tule vakavia vaurioita. Viallisessa tuotteessa saattaa olla osittainen tai täydellinen oikosulku, jolloin tuotteen ottamaa virtaa rajoittaa teholähteen virranrajoitus. Virranrajoituksen arvoa asetettaessa on kuitenkin otettava huomioon, että tuotteen käynnistyshetkellä ottama hetkellinen virta-arvo voi olla moninkertainen tavoitevirtaan nähden. Mittasimme tuotteen käynnistyshetkellä ottaman käynnistysvirran ja päätimme, että virtaraja tulee asettaa 1-2 ampeeria korkeammaksi kuin virtarajan suurin arvo.

Vuoden 2019 keväällä käyttöönotetussa testauslaitteistossa ledien tarkastus tapahtuu silmämääräisesti ja mittapäät asetettiin yhteen tuotteeseen kerrallaan testin ajaksi. Syksyllä 2019 valmistettujen tuotekohtaisten testausjigien avulla saavutettiin tilanne, jossa koko piirikorttiaiho pystyttiin testaamaan ilman mittajohtojen kytkemistä jokaiseen testattavaan tuotteeseen erikseen. Testijännitteiden ohjaaminen tuotteelta toiselle toteutettiin modulaarisenlogiikan avulla. Logiikka siirtää testijännitteen seuraavalle piirikortille hyväksytyin testin jälkeen, kun logiikka saa ohjaussignaalin LabView IO -kortin kautta.

Laitteiston toimittajan järjesti käyttökoulutuksen keväällä 2019 uusien testausohjelmien luomiseen. Koulutus käsitti ainoastaan jännite- ja virtaparametrien päivittämisen uusiin ohjelmiin, jotka luodaan tallentamalla \*.seq-päätteinen tiedosto uudella ohjelmanimellä. Uuden ohjelman tekoa varten laadimme yksinkertaisen ohjeen, joka opastaa työntekijän antamaan tarvittavat jännite- ja virta-arvot oikeisiin kohtiin ohjelmassa.

### **3.1 Testauslaitteisto**

Rakennettu testauslaitteisto koostuu tietokoneesta, tarratulostimesta ja Chroma 62000 DC-ohjelmoitavasta virtalähteestä. Lisäksi laitteistoon kuuluu switch-box, joka kääntää jännitteen polariteetin testin aikana. Switch-boxia ohjaa NI LabVIEW I/O-kortti, joka on kytketty USB-väylän kautta tietokoneeseen (KUVA 2).



KUVA 2. Testauslaitteisto

Tietokone on kytketty työpaikan verkkoon, mikä mahdollistaa kerätyn mittausdatan tallentamisen suoraan serverille. Kaikki testauslaitteet on kytketty USB-väylän kautta tietokoneeseen. Laitteistoon kuuluva Chroma DC-virtalähde ei pysty kääntämään jännitteen napaisuutta, joten se on toteutettu erillisellä releohjauksella. Kahdeksan releen avulla kytketään erikseen mitta-jännitteet DC- ja DC+, sekä sense-signaalit. NI TestStand -polariteetinvaihtosegmentin kautta ohjataan releitä siten, että jännitteen ja sense-signaalin polariteetti saadaan käännettyä testattavana olevalle tuotteelle (UUT).

### 3.2 Chroma 62006P-100-25 DC Power

Teholähteeksi valittiin Chroma 62006P-100-25, joka on 600 wattinen (W) ohjelmoitava malli. Teholähteen valinnassa otettiin huomioon, että samasta laitteesta on saatava kaikki tarvittavat testijännitteet 0-100 VDC. Teholähde kykenee antamaan 25 A:n virran aina 24 V:n jännitteeseen saakka ja 100 V maksimijännitteellä vielä 6 A:n virran. Tämä riittää mainiosti kyseiseen käyttötarkoitukseen.

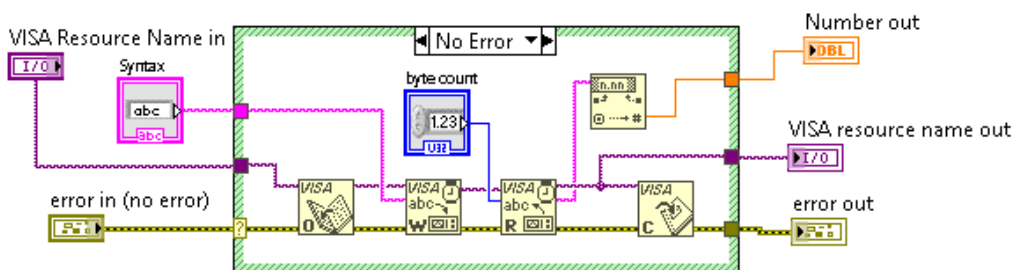
Mittajohdot on kytketty teholähteen taakse, josta löytyy myös sense-signaalit. Sense- ja sense+-signaalit muodostavat takaisinkytkennän johtimien jännitehäviöiden eliminoimiseksi. Sense-signaalit tulisi viedä mittajohdojen päihin asti. Näin teholähde pystyy kompensoimaan lähtöjännitteen, ja mittapäissä vaikuttava jännite on asetusarvon mukainen.

Chroma-teholähteen ohjaamiseen tarvitaan joukko virtuaalisia instrumentteja asetusarvojen syöttämiseen teholähteeseen ja mitattujen arvojen lukemiseen teholähteestä. Virtual



Instrument Software Architecture (VISA) on virtuaalinen rajapinta, joka on kehitetty parantamaan yhteentoimivuutta eri valmistajien mittalaitteiden välillä. VISA:a on mahdollista käyttää kommunikointiin eri instrumentointiväylien kanssa kuten Ethernet, GPIB, VXI, PXI tai sarjaportti. VISA-funktioita ovat esimerkiksi VISA Read ja VISA Write, joiden avulla tieto saadaan kulkemaan molempiin suuntiin tietokoneen ja laitteen välillä. VISA-funktioita ohjelmoitaessa ei tarvitse olla vielä tiedossa, mitä liitännäväylää tullaan käyttämään laitteen ja tietokoneen välissä, sillä se voidaan syöttää myöhemmin annettavalla parametrilla.

Mitatun virtarvon lukemiseksi VISA-resurssilta täytyy VISA Write funktiolla syöttää VISA Resource Name, joka sisältää laitteen osoitteen sekä tiedon siitä, miten se on kytketty tietokoneeseen. Lisäksi syötetään syntaksi, jota laitteen väylään halutaan kirjoittaa. VISA resource name- ja error-tiedot tulee kulkea jokaisen VISA-funktion läpi, sillä ohjelma pysytyisi, jos error-funktion tieto katkeaisi ja jossakin funktiossa tapahtuisi virhe (KUVIO 2).



KUVIO 2. LabVIEW VISA Block Diagram

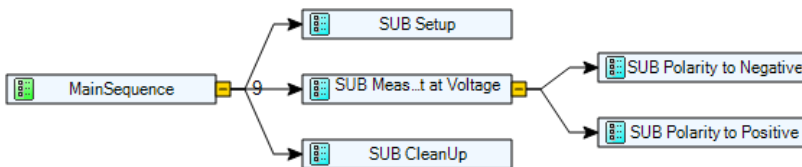
VISA sessio -tunnus luodaan VISA open -funktiolla. Chroma 6200P -mittalaitteelta virta-arvon lukeminen tapahtuu (FETCh:CURRent?) -syntaksilla, joka annetaan parametrinä VISA Write -funktiolle. Ja kun VISA-väylään on annettu tieto siitä, mitä halutaan lukea, voidaan data lukea VISA Read -funktion avulla. VISA Read -funktio palauttaa luetun datan string-muodossa, joten se täytyy muuttaa liukuluku (DBL) muotoon String To Number -funktiolla.

### 3.3 NI Teststand

Testausympäristö on rakennettu TestStand-käyttöliittymän ympärille, joka on National Instrumentsin valmistama testisekvenssimoottori. TestStand tarjoaa työkalut tuloksien raportointiin sekä mahdollistaa rinnakkaiset tai monisäikeiset testit. TestStandin vahvuuksista ovat valmiiksi

rakennetut toiminnallisuudet, kuten sarjanumerointi ja raportointi, joita on helppo muokata eri käyttötarkoituksiin.

Sekvenssit (sequence) ja subsekvenssit (subsequence) sisältävät joukon askeleita (steps), jotka suorittavat kukin tietyn toiminnon ja voivat kutsua virtuaalisia instrumentteja VI. TestStand tallentaa saadut data-arvot muuttujiin, jotka voivat olla globaaleja (global) sekvenssitiedostolle tai paikallisia (local) vain tietyille sekvenssille. TestStand ylläpitää globaalien muuttujien sekvenssitiedostossa (SequenceContext object), (KUVIO 3). Näin globaalit muuttujat voivat olla myös testiasemalle (station) yhteisiä niin, että ne tallentuvat kyseiseen main.seg-tiedostoon. Globaaleja muuttujia hyödynnetään muun muassa testausjärjestelmässä tuotteiden sarjanumeroinnissa.



KUVIO 3. NI TestStand -sekvenssi hierarkia

## 4 TESTAUSJÄRJESTELMÄN KEHITTÄMINEN

Tavoitteena oli että kaikki tutkimuksen kohteena olevan tuoteryhmän tuotteet testataan ja identifioidaan vaatimusten mukaisesti. Poikkeaminen tästä tekee testausprosessista epästabiilin, vaikka se muilta osin olisikin täysin vakaa. Kun mitataan koko testausprosessin suorituskykyä, koko valmistusprosessin suorituskyky on suurimmillaankin niin suuri kuin sen heikoimman lenkin suorituskyky. Tästä syystä ensimmäisenä parannetaan sitä osaa prosessista, jossa on suurin hajonta. Keväällä 2019 vain osa tuotteista testattiin vaatimustenmukaisesti, joten ensimmäinen tavoite oli kehittää testeriä niin, että kaikki tuotteet voidaan testata ja identifioida standardoidusti.

Testilaitteistoa suunniteltaessa tavoitteeksi asetettiin yhden komponenttilevyn testausajaksi 10 sekuntia. Testin suorittaminen mittajohtoja käyttäen vei kuitenkin lähes minuutin. Puolet ajasta kului mittajohtojen asetteluun ja puolet testiajoon. Vaikka tässä vaiheessa minulla ei ollut vähäkään tuntemusta NI TestStand -ohjelmoinnista, onnistuin löytämään NI TestStand -ohjelmasta parametrit, joilla onnistuin pudottamaan yhden tuotteen testiajon ottaman ajan kolmasosaan alkuperäisestä. Vaikka sain pudotettua testerin ottama testausajan kymmenen sekunnin tuntumaan, kului kokonaisen piirilevyaihion testaamiseen edelleen nelinkertainen aika tavoitteeseen nähden, sillä mittajohdot piti siirtää piirikortilta toiselle käsin.

Testausprosessin on nähty muodostavan kapeikon tuotannonvirtaukselle, varsinkin uuden testerin käyttöönoton myötä. Ensimmäisessä laadunparannusvaiheessa on huomioitavaa, että funktionaalista testausta ei tulla heti täysin automatisoimaan, joten testausprosessin kehittäminen tiedettiin muodostavan kapeikon tuotannonvirtaukselle. Automatisointiprojekti etenee vaiheittain siten, että ensimmäisenä keskityn laadullisiin asioihin, sen jälkeen siirryn analysoimaan ja parantamaan testauksen jaksonaikaa, ja viimeisessä vaiheessa parantamaan koko valmistusprosessin läpimenoaikaa.

### 4.1 Testausajan optimointi

Tuotteiden testaaminen yksittäin, mittajohtoja käyttäen, vei aluksi aikaa yli minuutin. Mittausjännitteen polariteetinvaihtosegmentin releiden kytkennän välissä olevan viiveen muutin

yhdestä sekunnista 0,2 sekuntiin ja jokaisen testijännitteen kytkennän jälkeisen viiveen muutin kolmesta sekunnista 0,2 sekuntiin. (KUVA 3). Näillä muutoksilla sain yhden tuotteen testiajon ottaman ajan vähenemään minuutista, kummenen sekunnin tuntumaan.

Parameter Name	Type	Log	Default	Value
Voltage	Number	<input type="checkbox"/>	<input type="checkbox"/>	7
CurrentLimit	Number	<input type="checkbox"/>	<input type="checkbox"/>	2
<u>WaitTimeS</u>	Number	<input type="checkbox"/>	<input type="checkbox"/>	<u>3</u>
MeasLow	Number	<input type="checkbox"/>	<input type="checkbox"/>	0.3
MeasHigh	Number	<input type="checkbox"/>	<input type="checkbox"/>	0.9

KUVA 3. Yksittäisen virranmittaussegmentin viive

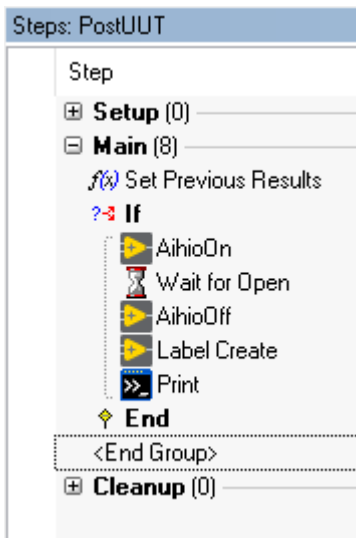
Jännitteen polariteettia käännettäessä ohjelmaan asetetulla viiveellä varmistetaan, että toiset mittajohtimien väliin kytketyt kuusi relettä kerkeävät avautua ennen kuin napaisuus käännetään kytkemällä toiset kuusi relettä. Releiden datalehdessä on ilmoitettu kontaktien avautumisajaksi 7-10 ms, joten 200 ms:n viive riittää hyvin varmistamaan sen, että releet ovat kerenneet avautua ennen toisten releiden kytketymistä. Polariteetin vaihdon aikana virtalähteen jännite on asetettu nolnaan volttiin, joten riski siitä, että releiden kontaktit kerkeäisivät oikosulkemaan virtalähteen jännitteen, on vähäinen.

Toinen turhan pitkä viive oli asetettu virtamittauksen yhteyteen. Viiveen tarkoitus on varmistaa, että tuotteen ottama virta on stabiili, jotta se saadaan mitattua luotettavasti. Testattavat LED-valaisintuotteet käynnistyvät kuitenkin hyvin nopeasti, joten 200 ms riittää mainiosti mitausajaksi. Joidenkin tuotteiden kohdalla huomasin tietyillä mittajännitteillä ledien syttyvän hitaammin, joten jouduimme nostamaan näissä tapauksissa viiveen 200 ms:sta 300 ms:iin.

Testausaikojen optimoinnin yhteydessä huomasin, että jos estosuuntaisen virran mittaus on testiajon lopussa, mittajohtojen liian aikainen irti kytkeminen ei aiheuta hylättyä testiä. Tämä johtuu siitä, että estosuunnassa virtaraja on 0-10 mA, joten mittajohtojen ollessa irti kytkettynä testerin mittaama virta on hyväksyttävät 0 A. Mittauksen luotettavuuden varmistamiseksi päätin, että kaikki sellaiset segmentit, joissa tuotteen ei tule ottaa virtaa, eivät saa olla testiajon alku- ja loppupäässä.

## 4.2 Tuotekohtaiset ratkaisut

Aloitin tuotekohtaisten neulapetien suunnittelun keväällä 2019, heti testausjärjestelmän käyttöönoton jälkeen. Tuotekohtainen neulapeti mahdollistaa kaikkien aihiossa olevien piirikorttien testaamisen yksittäin, ilman mittajohtojen siirtelyä kortilta toiselle. Lisäksi neulapedit tarvitsevan releohjauksen, jolla mittajännite ohjataan yhden tuotteen piirikorttiin kerrallaan. Onneksenne testilaitteiston IO-kortissa oli vapaita lähtöjä, joita hyödynsimme testipetien ohjaamiseen. NI TestStand -ohjelmaan tuli lisätä oma segmentti, jolla ohjaussignaali saatiin tuotua liitántärajapintaan. Jokaisen hyväksytysti suoritettun testin jälkeen NI LabVIEW -ohjelmaan liitettiin segmentti, joka ohjaa logiikan siirtymään automaattisesti seuraavaan piirikorttiin (KUVA 4).



KUVA 4. Aihionvaihtopulssi IO-porttiin ennen tarran tulostusta

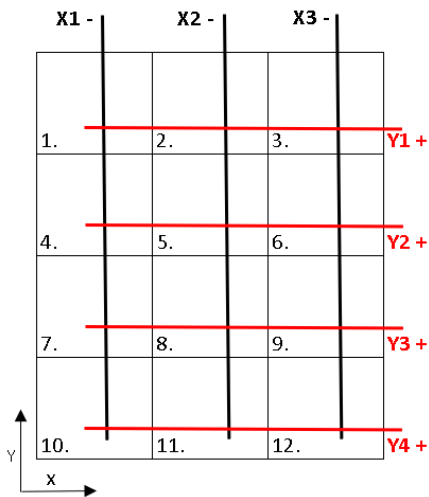
Lähtökohtana suunnittelulle oli, että työntekijän ei tarvitse tehdä ylimääräisiä kytkentöjä tai asetuksia tuotteen vaihdon yhteydessä. Tuote saattaa vaihtua useita kertoja päivässä, joten tuotteen vaihto täytyy tehdä helpoksi. Toteutin mittajännitteen ohjaamisen oikeaan piirikorttiin aihiossa Mitsubishin PLC-logiikalla (KUVA 5). Logiikka siirtää mittajännitteen automaattisesti seuraavaan korttiin hyväksytyyn mittaukseen jälkeen. Lisäksi liittimen kautta välittyy piirilevyn koko ja muoto 4-johtimen kautta (LIITE 2). Mahdollisia piirilevyn kokoja ovat: 2x1, 2x2, 2x3, 3x2, 4x2, 3x3, 3x4, 4x3 ja 4x4.

Aihion koko valitaan liitinkotelon sisään asetettavilla hyppylangoilla. Liittimen nastat 11-14 on kytketty PLC-logiikan sisääntulonastoihin. Hyppylangan kytkeminen valitun liittimen nastan ja 24VDC:n väliin aktivoi kyseisen sisääntulon.



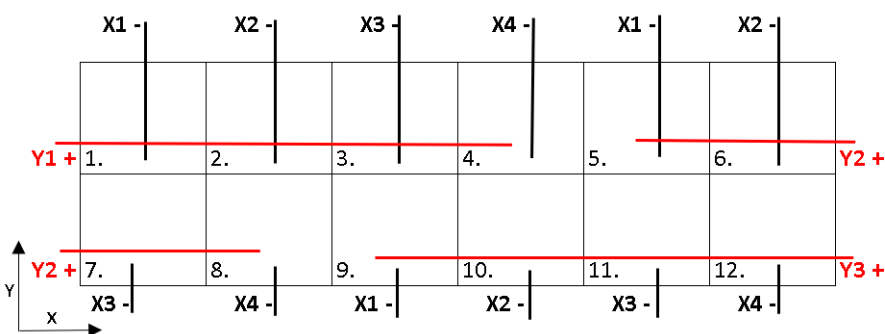
KUVA 5. PLC-ohjain

Mitsubishi AL2-24MR-D-Alpha XL -logiikkaohjaimen relelähdöistä 1-4 on tarkoitettu 8 A:n kuormalle ja lähdot 5-9 2 A:n kuormalle. Tästä syystä jouduin lisäämään neljä erillistä 8 A virran kestävä relettä, joita ohjataan lähtöjen 5-8 kautta (LIITE 3). Mitsubishi AL2 -logiikassa lähtöjen vasteaika on 10 ms, ja niiden perään kytkettyjen releiden vasteaika on 5 ms, mikä on huomioitu lisäämällä ohjelmaan releiden kytkennän jälkeen 200 ms:n viive. Jännitesyötön ohjaus testattaville tuotteille toteutettiin matriisikytkennällä, missä demultipleksaamalla syöttöjännite saadaan ohjattua siten, että jokainen aihiossa Y-suunnassa oleva piirikortti saa miinusjännitteen saman aikaisesti ja aihiossa X-suunnassa olevat piirikortin plusjännitteen. Tällä tavoin tuotekohtaisten jigien kaapelointi onnistuttiin hoitamaan enimmillään kahdeksaa virtakaapelia käyttäen.



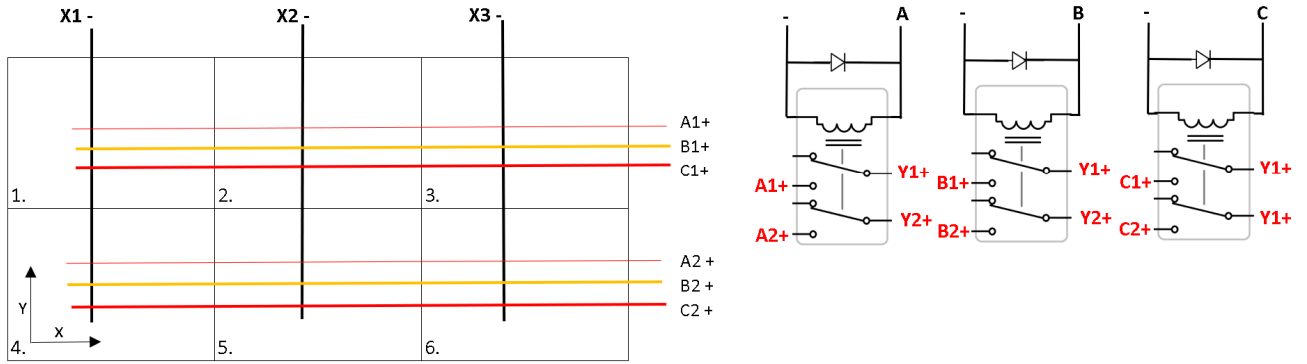
KUVIO 4. 3x4 aihion johdotus

Miinusjännite ohjataan aihion kaikkiin pystysuoran rivin tuotteisiin yksi kerrallaan ja plusjännite vaakasuoran rivin tuotteisiin. Näin saadaan testijännite syötettyä yhteen tuotteeseen kerrallaan (KUVIO 4).



KUVIO 5. 6x2 aihion johdotus

Yhden aihion koko on 6x2 piirikorttia. Virtasyötön matriisi on lähtökohtaisesti suunniteltu ensimmäisellä 4x4 muotoiselle aihiolle, joten matriisikytkentä täytyi tämän tuotteen osalta kytkeä eri tavalla (KUVIO 5.) Mukana on myös yksi tuote, joissa on useampi erivärinen LED-ryhmä. Näiden tuotteiden ohjaamiseen rakennettiin jigi. Jigissä matriisiohjauksen perässä on releohjaus, joka syöttää jännitteen tiettyihin nastoihin testiohjelman eri vaiheissa (KUVIO 6). Jokaisella värillä on määritelty omat virtarajansa. Värin valitseminen toteutettiin lisäämällä NI TestStand-ohjelmaan oma segmentti jokaiselle värille (LIITE 4).



KUVIO 6. Eriväristen LED-ryhmien ohjauksen kytkentä



### 4.2.1 Testerin ohjaus

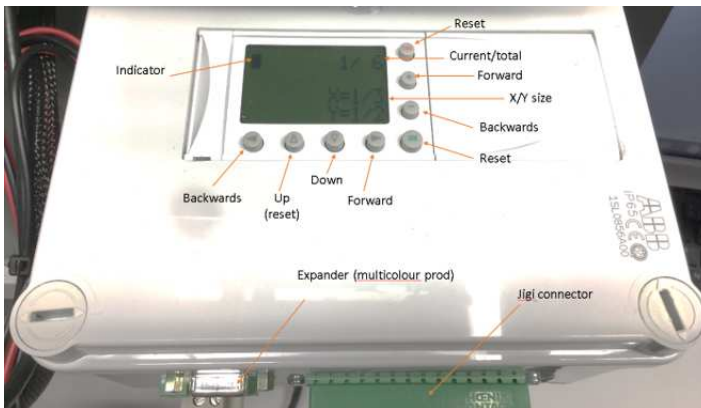
Tavoitteena oli luoda ohjauslogiikka mittajännitteiden ohjaamiseen yhden piirikortin rajapintaan kerrallaan. Ensimmäisen version ohjauslogiikasta toteutin Philipsin 74HC174 Hex D-type flip-flop -laskureilla ja L293E Push-Pull -ajureilla, jossa X- ja Y-suuntaisia rajottimet huomioiden signaali ohjataan tuotekohtaisen liitännän rajapintaan. Yksinkertaisuudessaan laskurit U1- ja U2-lähdöt ohjaavat lähdöt X1 – ja Y1 + (LIITE 5). Logiikkapiireillä toteutettu kytkentä ei kuitenkaan ollut täysin luotettava. Ongelmat johtuivat todennäköisesti piirien sisäisistä viiveistä, jotka vaihtelivat valmistajakohtaisesti. Luotettavuus ongelmista johtuen päätettiin, että tämä ohjaustapa korvataan ohjelmoitavalla PLC-logiikalla.

### 4.2.2 PLC-ohjaus

Programmable Logic Controller (PLC) on uudelleenohjelmoitava mikrokontrolleriohjattu laite, jossa on joko modulaarisia tai integroituja tulo- ja lähtöliitäntöjä. PLC-logiikan valintaan vaikutti ensimmäisen vuoden opinnoissa saatu kokemus Mitsubishi Alpha 2 -kompaktilogiikan ohjelmoinnista. Pohjana suunnitellulle logiikkaohjelmoinnille toimi aiemmin logiikkapiireillä toteutettu kytkentä, missä piirilevyaihion koko on määrittely hyppylangoilla. Mitsubishi Alpha 2 -logiikkaohjaimen näyttö tarjoaa käyttöliittymän (Human-Machine Interface, HMI), joka mahdollisuuden ohjaimen tilan seurannan aihion testauksen aikana (KUVA 7). Näytöltä voidaan lukea, missä kohtaa aihiota testi on milloinkin menossa. PLC-logiikan ohjelmoinnin lähtökohtana oli, että testaajan ei tarvitse tehdä logiikkaan asetuksia vaan logiikan tarvitsema tieto aihiossa olevien piirikorttien määräst ja aihion muodosta tulisi käytettävästä testausjigistä. Tuotekohtaisissa testipedeissä on 14-napainen Phoenix-liitin, jonka kautta voidaan ohjata testausjännite enimmillään 16 testattavalle piirikortille (LIITE 2).

Mitshubishi Alphan ohjelmointi pohjautuu toimilohkokaavio-tyyppiseen (Function block diagram, FBD) visuaaliseen ohjelmointiin. Ensiksi asensin tietokoneelle Alpha Programming -sovelluksen, joka on ilmaiseksi ladattavissa Mitshubishin internetsivuilla. Logiikan ohjelmointiin tarvitaan lisäksi erikseen hankittava ohjelmintikaapeli, jolla ohjelma saadaan siirrettyä tietokoneesta logiikan muistiin. FBD-ohjelmointi ei vaadi aiempaa kokemusta logiikkaohjelmoinnista, sillä ohjelmointitapa on erittäin suoraviivainen ja helposti ymmärrettävä.

Kääntöpuolena on, että ohjelmasta tulee helposti sekava ja johtimet kulkevat ristiinrastiin, joten ohjelman kommentointi ja dokumentointi on ehdottoman tärkeää.



KUVA 6. PLC-ohjaimen käyttöliittymä HMI

Testauslaitteiston IO-kortista saatavaa pulssia hyödynnettiin PLC-logiikan ohjaamiseen. Tätä varten jokaisen testin loppuun lisäitiin segmentti, joka antaa pulssin PLC-logiikalle. IO-kortilta saatava pulssin toin logiikalle analogiseen tuloon, koska LabVIEW IO -kortista tulevan signaalin jännitetaso ei ollut 0 V tai 5 V. AL2-logiikassa on kahdeksan 8bitin resoluutiolla olevaa sisääntuloa. Gain-operaatiolla sain analogisen signaalin vahvistettua ja Schmitt trigger -operaatiolla määrittelin liipaisutasot, joilla sain analogisen signaalin muutettua digitaaliseen muotoon.

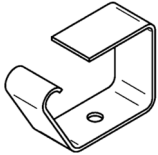
Logiikan ohjelmoinnissa käytin hyväkseni signaalina analogisia olo- ja asetusarvoja. Logiikan relelähtöjen ohjaamiseen käytin compare-toimilohkoa, jossa analogista signaalia ohjataan tilan perusteella oikeaan lähtöön ohjelman eri vaiheissa. Aihion koon valitsimen toteutin Boolean toimintolohkoa käyttäen. Boolean loogiset operaatiot määrittelin niin, että liitinrajapinnan liitäntöjen tilat ohjataan X- ja Y-suunnan laskureille, josta niiden asetusarvot saadaan analogisessa muodossa ulos. Liitteenä ohjaimen graafinen ohjelmointi kokonaisuudessaan (LIITE 6).

#### 4.2.3 Testerin rakennus

Vuoden 2019 syksyllä rakensin tuotteisiin ensimmäiset testipedit, joissa virtaliitin menee piirikortin läpi alhaalta päin. Testipetejä suunniteltaessa oli tavoitteena luoda mahdollisimman avoin ja yksinkertainen rakenne, jotta huoltaminen ja vastaliittimien vaihtaminen olisi

mahdollisimman helppoa. Rakenne on sandwich-tyyppinen, eli kaapelointi on kahden CEM1-levyn välissä (LIITE 9).

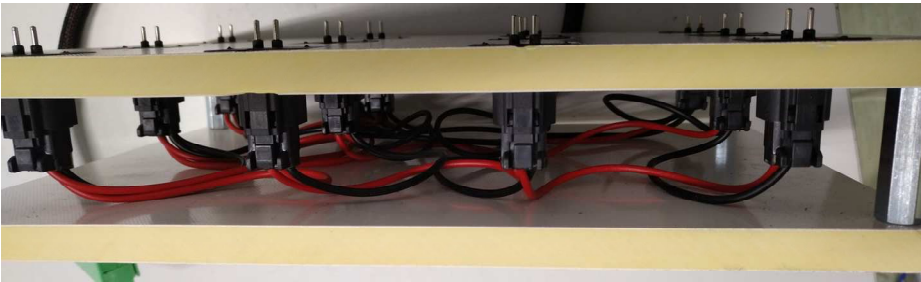
Suurimmassa osassa testattavista tuotteita virtaliitin työntyy alakautta piirilevyissä olevista au-koista. Näille tuotteille riittivät yksinkertaiset tuotekohtaiset testausjigit, joissa vastaliittimet on upotettu alustaan, jonka päälle testattava tuote asetetaan (KUVA 8).



KUVA 7. Contact Finger

Tuotekohtaiset testipedit jyrsin 8-10mm:n CEM1-kerroslaminoidusta materiaalista, joka vastaa ominaisuuksiltaan piirilevyissä käytettävää FR4 materiaalia. Ensimmäiseksi laadimme listan testattavista tuotteista Excelliin. Taulukkoon kerättiin tiedot piirilevyjen ja aihoiden koosta sekä käytettävän liittimen tyypit. Dimensiot saatiin tuotteiden gerber-tiedostoista. Taulukko laadittaessa huomiottiin myös mahdolliset ramp down -tuotteet, jotka ovat poistumassa tuotannosta. Keräsin samaan taulukkoon gerber-tiedostoista kaikki tarvittavat mitat, jotka sain suoraan tuotua SolidWorksiin konfiguraatiotiedostona. Suunnittelin SolidWorks-ohjelmalla jigien, jossa oli viittaukset Excel-taulukosta löytyviin mittoihin. Konfiguraatio toiminnon hyödyntäminen säästi paljon aikaa tuotekohtaisten jigien suunnittelussa, kun mitoitusta ei tarvinnut tehdä jokaiselle tuotteelle erikseen (LIITE 10).

CEM1 EP 105 AS on piirilevymateriaalia vastaava kerroslaminoitu materiaali. Levymateriaalissa ydin on epoksihartsikövetettyä paperia ja pintakerrokset ovat epoksilasikuitulaminaattia. Pienten alle 1mm:n reikien poraaminen epoksilasikuitulaminoituun pintakerrokseen on hankalaa, koska pieni terä ei keskity kunnolla kovalla pinnalla. Huomasin, että ennen pienen reiän poraamista kannattaa pintakerrokseen tehdä halkaisijaltaan 1,5-2 mm leveä potero, joka ulottuu ainoastaan pintakerroksen läpi (KUVA 14).



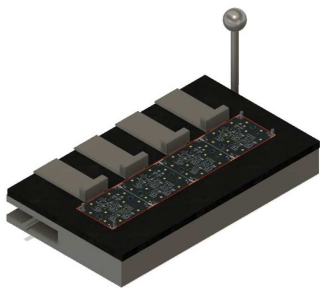
KUVA 8. Tuotekohtainen testijigi

Toisentyypiset virtaliittimet omaavissa piirikorteissa vastaliitin tuodaan piirikortin sivusta, joten näitä tuotteita varten joudun rakentamaan enemmän mekaniikka sisältävä testerin (KUVA 9).



KUVA 9. Molex-liitin

Testerirungon ensimmäisessä versiossa piirikortit olisi pitänyt osittain irrottaa aihioista ennen testausta. Ideana oli, että Molex-vastaliittimet olisi kiinnitetty liikemekaniikkaan, joka työntäisi ne liittimiin testin ajaksi (KUVA 10).

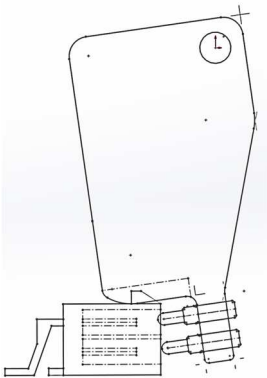


KUVA 10. Ensimmäinen suunnitelma testerirungosta

Suunnitelmana oli, että sama testerin kävisi kaikille tämän tyyppisille tuotteille. Tarkoitus oli, että komponenttilevyn alla oleva asetti määrittäisi liittimen sijainnin ohjaamalla liikkuvat osat oikealle etäisyydelle toisistaan. Tuotteille olisi kuitenkin pitänyt tehdä suojapinnoitus niiden ollessa kokonaisessa aihiossa, joten tuotteet olisi testattu vasta pinnoituksen jälkeen. Tämä vaihtoehto

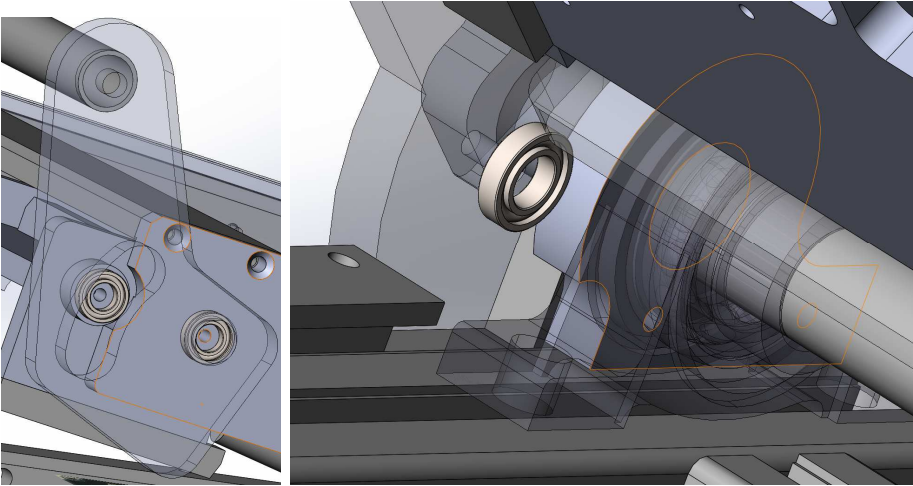
hylättiin. Niinpä ryhdyin suunnittelemaan testeriä, jossa koko piirikorttiahion testauksen voisi suorittaa ilman piirikorttien irrottamista aihioista.

Aluksi oli tarkoitus rakentaa vastaliitin, joka ei vaadi ympärilleen enempää mekaniikkaa kuin avattavan kannen, jossa vastaliittimet olisivat kiinnitettyinä. Suunnittelin SolidWorks-ohjelmalla vastaliittimen, joka jousikuormitettuna kääntyy liittimen sisään laskettaessa se liittimen päälle (KUVA 11). Tulostin 3D-pöytätulostimella muovisen vastaliittimen, jotta sain kokeiltua käytännössä, miten tämän tyyppinen ratkaisu toimisi. En kuitenkaan ollut täysin tyytyväinen tähän ratkaisuun, sillä kun vastaliittimen nosti nopeasti pois liittimen päältä, niin siinä olevat testi-neulat osuivat liittimen runkoon. Hylkäsin tämän ratkaisun, koska se ei tuntunut luotettavalta tuotantokäyttöön.



KUVA 11. Ensimmäinen versio Molex-vastaliittimestä

Suunnittelin SolidWorks-ohjelmalla testerirungon, joka soveltuu kaikkien Molex-tyyppisen liittimen omaavien tuotteiden testaukseen. Suunnittelin testerinrungon siten, että vastaliittimet työntyvät piirikorteissa oleviin Molex-liittimiin, kun testerin luukun kahva käännetään alas. Haastavinta oli suunnitella mekaniikka siten, että kahvan kääntäminen luukun ollessa auki on estetty (KUVA 12). Lukkomekaniikan huolellinen suunnittelu oli tärkeää, sillä jos kannen avaus olisi sallittu vastaliittimien ollessa komponenttilevyssä olevan liittimen sisässä, aiheuttaisi se vaurioita liittimiin. (LIITE 11).



KUVA 12. Testerirungon saranalukkomekanismi

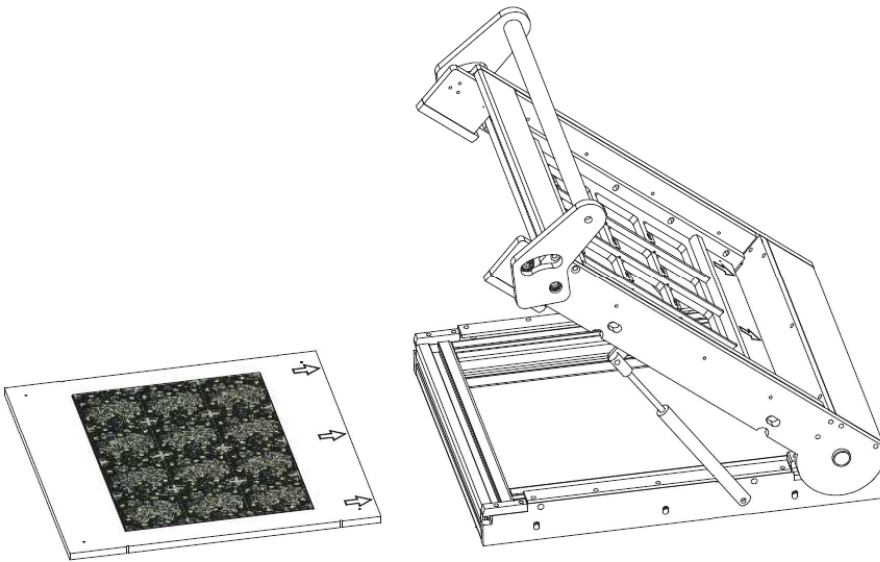
Lukkomekaniikan toteutuksessa hyödynsin kuulalaukereita, jotka on sijoitettu siten, että suunniteltu geometria ei salli kiellettyjä liikkeitä. Kahvan alas kääntäminen siirtää kannessa olevan tuotekohtaisen neulapedin kannen saranaa kohti ainoastaan silloin, kun kansi on suljettuna. Lukitusliikkeen pituus on säädettävissä 0-7mm:n välillä ja tuotekohtaisen neulapedin paksuus on säädettävissä 7-11mm:n välillä.

Suunnittelin SolidWorks-ohjelmalla kokoonpanon ja jyrättäviin osien CAM-ohjelmoinnin tein Autodesk Fusion 360 -ohjelmalla (LIITE 12). Tarvittavat alumiiniosat työstin autotallassani omalla CNC-koneella (KUVA 13). Numeerisesti ohjattu kone on omavalmisteinen, kolmeakselinen portaalirobotti. Alumiiniosissa käytin EN AW-6063-alumiiniseosta, joka on helposti työstettävää.



KUVA 13. Testerirungon alumiiniosat työstetty

Testerirungon suunnittelun osalta toteutui sanonta ”hyvin suunniteltu on kuin puoliksi tehty”. Mallinsin kokoonpanon ennen osien työstöä Solidworksin kokoonpanotyökalulla. Kokoonpano onnistui ongelmitta ja kaikki osat kohdistuivat suoraan paikoilleen. Pohjarunko on valmistettu 40x40-kokoisesta alumiiniprofiilista, joka on kiinnitetty 10mm paksun teräslevyn päälle testerin painon lisäämiseksi (KUVA 14).



KUVA 14. Testerirungon kokoonpanopiirustus

Kovimmalle rasitukselle joutuvat lukkomekaniikan osat vaihdettiin kulutusteräksestä (OPT650) laserleikattuihin osiin. Tällä ennakoin kulumisesta johtuvat toimintahäiriöt lukkomekaniikan toiminnassa. Kokeilimme kahta eri vaimennuksen omaavaa kaasujousta, ja valitsimme 80N:n vaimennuksen omaavat 15-6 L 236-100-80N tyyppiset kaasujouset. Kaasujousten valmistaja Lesjöfors tarjoaa myös laskentakaavan oikeantyyppisen kaasujousten valintaan, mutta yksittäistä laitetta suunniteltaessa riittää, että käytettävyyden ja testerin turvallisen käytön on testattu käytännössä.

#### 4.2.4 Testipetien suunnittelu ja valmistus

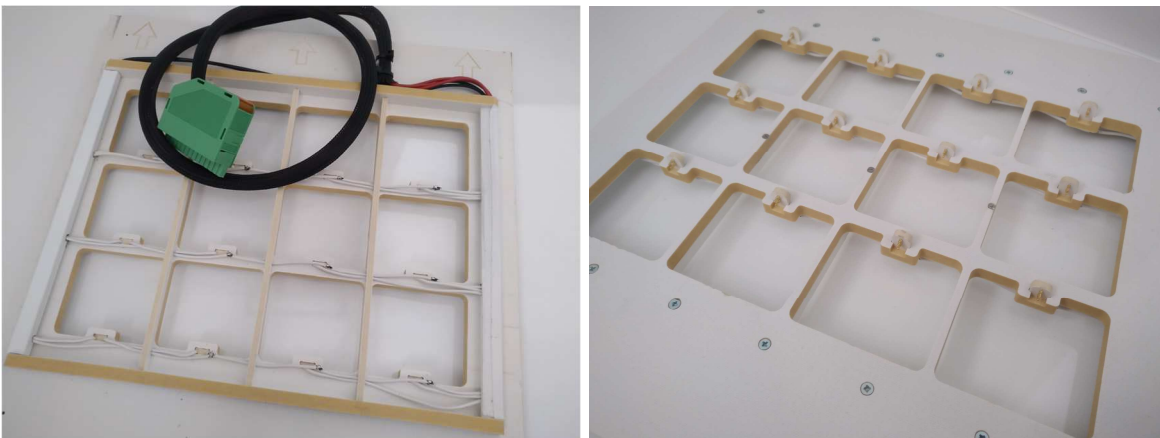
Testerirunkoon tulevien testipetien suunnittelu oli huomattavasti haastavampaa, koska ylä- ja alapedin täytyy kohdistua testin aikana, jotta vastaliitin osuisi mahdollisimman hyvin tuotteen liittimeen. Alapetiin tuli tuotteen asemointiin kohdistustapit ja yläpetiin vastaliittimet (LIITE 18). Testineulaksi valitsen Smiths Connectorin valmistaman 101190-testineulan. Tärkeä kriteeri

neulaa valittaessa oli virran kesto, joka kyseissä testineulassa on 15 A. Testineulan jousivoima on 74 g ja liikerata 1,7 mm, mikä vaikutti sopivalta tähän projektiin. Testinaulat juotetaan paikolleen, joten ne ovat helposti vaihdettavissa. Testineuloja vaihdettaessa on huomioitava, että itse neulaan ei saa koskea pihdeillä liikkumapintojen vaurioitumisen estämiseksi (KUVA 15).



KUVA 15. Yläpetiin asennettavan vastaliittimen CAD-kuva

Piirilevyjen paksuudet vaihtelevat tuotteittain 1,6 mm-2,4 mm:n välillä, joten yläpedissä olevat vastaliittimet suunnittelin siten, että niiden korkeutta pystyy säätämään. Tuotekohtaiset jigat valmistin 8 mm:n CEM1-materiaalista ja vastaliittimien osat 4mm:n CEM1-materiaalista (KUVA 16). Vastaliittiniosan suunnittelin siten, että se painaa samalla piirilevyn pohjapetiä vasten. Testerirunko on suunniteltu siten, että ala- ja yläpeti jäävät 8 mm:n etäisyydelle toisistaan. Vastaliittimiä varten yläpetiin suunnittelin 4,1 mm\*10,1 mm-kokoiset suorakulmion muotoiset aukot, joihin vastaliittimet asennetaan. Aukkojen sovitus on tarkoituksella tiukka, jottei vastaliittimiä tarvitse erikseen lukita (LIITE 13).



KUVA 16. Tuotekohtainen testerirunkoon tuleva yläpeti



#### 4.2.5 Jännitehäviöt

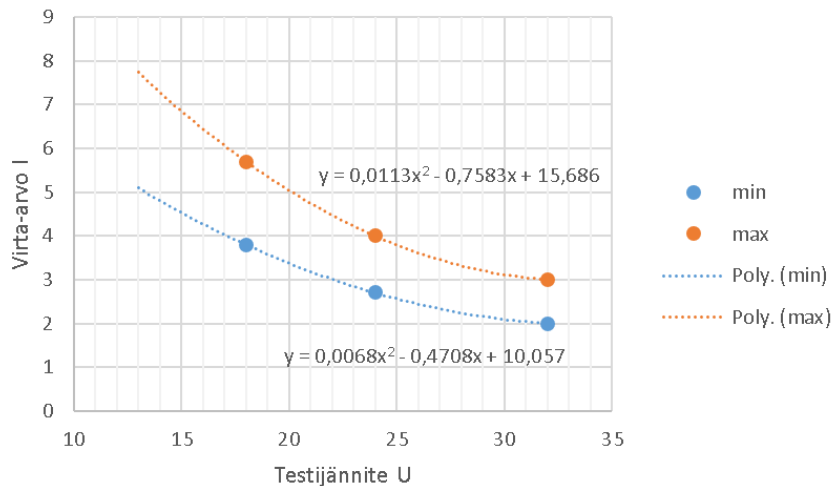
Testerinkaapelointia suunniteltaessa oli huomioitava kaapeleissa syntyvät jännitehäviöt ja niiden vaikutus mittaustuloksiin. Kun teholähteen sense- ja sense+ -singnaaleja ei ole tuotu tuotteen liitinrajapintaan asti, vaan sense-signaalit katkeavat polariteetin käännön jälkeen. Käytännössä tuote ottaa enemmän virtaa, kun mittajohtojen resistanssin aiheuttama jännitehäviö laskee testattavan tuotteen napajännitettä. Ohmin lain mukaan sähköjohtimen aiheuttamalla jännitehäviöllä tarkoitetaan johtimen resistanssin ( $R$ ) ja sen läpi kulkevan sähkövirran ( $I$ ) tuloa. Johtimen resistanssi saadaan kaavalla:

$$R = \frac{\rho l}{A} \quad (4)$$

Jossa ( $\rho$ ) on johdinaineen ominaisresistanssi, ( $l$ ) johtimen pituus ja ( $A$ ) poikkipintala. Ominaisresistanssi on ilmoitettu 20 °C lämpötilassa, ja metalleilla ominaisresistanssi kasvaa lämpötilan kasvassa materiaalin lämpötilakertoimesta riippuen.

Rakennetuissa testijigeissä vastaliittimien sähköjohdot ovat kuparia ja poikkipinta-alaltaan  $1\text{mm}^2$ , mikä riittää hyvin tuotteen maksimissaan 7 ampeerin ( $A$ ) virransyöttöön. Johtimessa on suojaeristys 500V:n jännitteeseen asti, koska testijännite ylittää ajoneuvokäyttöön tarkoitetun 75V:n jännitteen. Kaapeli on tyypiltään MKEM-HF monisäikeinen halogeeniton kuparijohdin. Testerikäyttöön valitaan yleensä robottikaapeli, joka on suunniteltu kestävästi jatkuvaa taivuttelua. Tässä laitteistossa kuitenkin sellaista rasitusta ei ole. Yhteensä mittajohdon pituudeksi tulee enimmillään kaksi metriä, joten muodostuva jännitehäviö on kahdella johtimella mitattaessa  $\frac{0,0168\ \Omega}{1\ \text{mm}^2} * 2\text{m} * 7\text{A} = 0,24\text{V}$ , jossa  $0,0168\ \Omega$  on johtimen ominaisresistanssi. Tämä tarkoittaa, että jos tuote ottaa 7A:n virran testijännitteen ollessa 12 Volttia (V), vaikuttaa tällöin tuotteen virtaliittimiin 11.76 V:n napajännite (KUVIO 7).

Otin esimerkkituotteen ottamat virrat excel-taulukkoon ja piirsin pisteiden kautta kulkevan trendiviivan. Virtarajat muuttuvat kyseisen tuotteen jännitteen funktiona seuraavasti:



KUVIO 7. Tuotteen ottama virta jännitteen funktiona

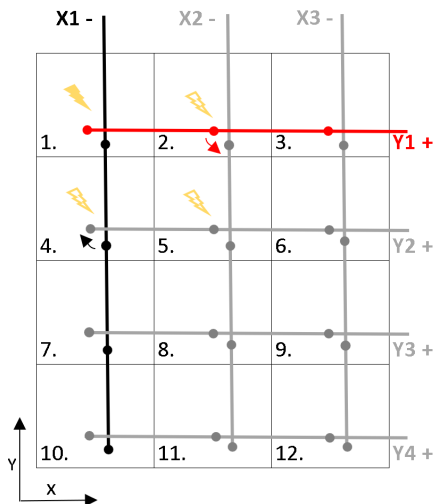
$$I_{min} = 0,0068U^2 - 0,4708U + 10,057 \quad (6)$$

$$I_{max} = 0,0113U^2 - 0,7583U + 15,68 \quad (7)$$

Missä  $U$  on tuotteen napoihin vaikuttava jännite testin aikana. Esimerkkitapauksessa 18 V:n testijännitteellä virtaliittimen napoihin vaikuttaisivat 17,8-17,87 V:n jännitteet. Tuntemillamme arvoilla voidaan laskea tuotteen ottama maksimi- ja minimivirta. Sijoitetaan nämät jännitearvot kaavoihin 6 ja 7. Minimivirta olisi  $0,0068 * 17,87^2 - 0,4708 * 17,87 + 10,057 = 3,82 \text{ A}$  ja maksimivirta olisi  $0,0113 * 17,8^2 - 0,7583 * 17,8 + 15,68 = 5,77 \text{ A}$ . Kun verrataan saatuja minimi- ja maksimivirta-arvoja, voidaan huomata virheen kasvavan virran kasvaessa. Mittausvirhe on 20 mA-70 mA, kun mitataan 3,8 A-5,7 A:n virtoja, eikä näin ollen ole merkittävä testin kannalta.

#### 4.2.6 Vuotovirrat

Kahden tuotteen testausjigin käyttöönoton yhteydessä havaittiin, että matriisiin demultipleksattu syöttöjännite vuotaa aihion piirikorteille, jotka eivät ole testattavana (UUT). Tämä ongelma ilmenee etenkin 80-100 V:n testijännitteillä ja osalla testattavista piirikorttiaihioista. Kyseisten aihioiden testaus onnistui vain pilkkomalla aihio pienempiin osiin ja testaamalla aihio useammassa pienemmässä erässä (KUVIO 8).



KUVIO 8. Vuotovirrat

Ongelman ratkaisemiseksi ei ollut muuta vaihtoehtoa kuin suunnitella releohjaus, jolla testijänite ohjataan yhdelle tuotteelle kerrallaan. Suunnittelin logiikan ja tuotekohtaisen testausjigin väliin tulevan releohjaimen, jota voidaan ohjata olemassa olevalla logiikkaohjaimella niin, että työntekijä pystyy kontrolloimaan testausta edelleen logiikkaohjaimen käyttöliittymästä (HMI).

Mitsubishi AL2-24MR-D-Alpha XL -logiikkaohjaimessa on yhdeksän relälähtöä, joista kahdeksan on jo varattu matriisikytkentää varten. Lisälähtöjä varten täytyi hankkia siis AL2-4EYR-laajennusmoduuli, joka asetetaan logiikkaohjaimen päällä olevaan liitäntään. Logiikkaohjaimen ohjelmaan täytyi myös lisätä ohjaus, jolla neljästä ulostulosta saatiin 16-kanavainen ulostulo binäärimuodossa. Desimaaliluvun muuttaminen binäärimuotoon on haasteellista Mitsubishiin toimilohko-tyyppisessä ohjelmointikielessä, josta puuttuu tuki eksponentti laskutoimituksille. Vähiten ja eniten merkitsevän bitin ohjelmointi kävi suoraviivaisesti, mutta toisen ja kolmannen bitin osalta jouduttiin tekemään useita vertailuoperaatioita.

Binäärimuotoisen datan muuttaminen desimaalimuotoon onnistuisi suoraan logiikkapiirillä demultipleksaamalla 4-bittinen signaali 16 kanavaan. En kuitenkaan löytänyt sopivaa läpiladottavaa logiikkapiiriä, jolla tämän olisi voinut toteuttaa yhtä komponenttia käyttäen, joten toteutin demultipleksauksen Teensy 3.2 -mikrokontrollerilla. Joka on 32-bittinen ARM-mikrokontrollerilla varustettu kehitysalusta, jossa on 34 IO-kanavaa. Teensy mahdollistaa myös kontrollerin ohjaamisen USB-väylän kautta, joka tarjoaa hyvät jatkokehitysmahdollisuudet. Ohjelmakoodi on työssä liitteenä (LIITE 7).

Teensy:n antama 3.3 V:n lähtöjännite ei riitä sellaisenaan releiden ohjaamiseen. Käytin ohjaussignaalin vahvistamiseen neljää L293E Push-Pull-ajuria, joilla 3.3 V:n jännite saatiin vahvistettua releiden tarvitsemaksi 24 V:n jännitteeksi. Lisäksi jokaisen releen kelan rinnalle täytyi asettaa diodit oikosulkemaan kelan itseinduktion aiheuttamat jännitepiikit. Jotta jännitteen syöttäminen onnistuisi jokaiselle tuotteelle erikseen, täytyi testijigin kaapelointi muuttaa siten, että kaikilla aihiossa olevilla komponenttilevyillä olisi yhteinen maa (Common Ground), sekä testijännite johdetaan releiden kautta yhteen tuotteeseen kerrallaan (LIITE 8). Kytkenän toteutin reikälevylle, johon komponentit juotetaan kiinni ja johdotukset tehdään hyppylankoja käyttäen.

### **4.3 Luotettavuuden kehittäminen konenäön avulla**

Testin aikana työntekijän tulee havaita, että kaikki piirikortissa olevat ledit syttyvät vaaditulla testijännitteillä. Inhimillisistä tekijöistä johtuen tämä on käytännössä kuitenkin mahdotonta. Konenäön avulla voidaan saavuttaa lähes 100% luotettavuus ledien tarkastukseen, kun inhimilliset tekijät suljetaan pois. Tavoitteena oli, että testiohjelma laskee syttyneiden valopisteiden määrän, samaan tapaan kuin virtamittaus mittaa sähkövirtaa. Konenäköratkaisun tavoitteena oli, että TestStand kutsuu tekemääni Virtual Instrumenttia (LEDtest.vi) ja käyttää sitä ledien määrän laskemiseen. Tutustuessani NI LabVIEW -ohjelmointiin suunnittelin kaksi eri tavoin toteutettua konenäkösovellusta, jotka molemmat muodostavat oman virtuaalisen instrumentin VI. Toinen on Vision Assistant Express ja toinen IMAQdx-funktioiden hyödyntäminen. Näin tustuin samalla molempiin tapoihin liittämään kamera osaksi LabVIEW-sovellusta.

#### **4.3.1 NI Vision Acquisition Software (VAS)**

Tarkastettavat ledit ovat erittäin kirkkaita, luokkaa 4000-10000lm. Tämän vuoksi kuva palaa helposti puhki kameran valotuksen ollessa liian korkea. Puhki palaneesta kuvasta ei voi erottaa ledejä toisistaan, vaan kuvassa on vain yksi iso valkoinen valoalue. Kokeilin asettaa tummentavan akryylimuovin kameran ja testattavan tuotteen väliin, mutta se sai aikaan heijastuksia, joita olisi joutunut suodattamaan esimerkiksi polarisoivalla linssillä. Web-kameraan suodattimia on vaikea asentaa. Etsin sopivat arvot kirkkaus- ja valotusarvolle niin, että ledit erottuivat selvästi.

NI LabVIEW perusasennuspaketti ei pidä sisällään konenäköön liittyviä ohjelmistoja eikä ajureita. NI Vision Acquisition Software (VAS)-työkalupaketti sisältää kolme kuvankäsittelyajuria, jotka ovat:

**NI-IMAQ** – tiedostojen näyttö-, hankkimis- ja tallennustoiminnot

**NI-IMAQdx** – kamera ajurit, GigE Vision, IEEE-1394, USB ja IP-kamerat

**NI-IMAQ I/O** – FPGA (Field Programmable Gate Array)

Nämä ajurit sisältävät joukon konenäön kannalta hyödyllisiä, valmiita virtuaalisia instrumentteja kuvan hankkimiseen, näyttämiseen sekä tallentamiseen. VAS työkalupaketti sisältää myös Vision Assistant (VA)-ohjelman, jonka avulla kuvan analysointi algoritmi voidaan suoraan kääntää (\*.vi)-päätteiseksi tiedostoksi. Tämä toimii suoraan virtuaalisena instrumenttina LabVIEW-ohjelmassa. (Knowledge NI 2020.)

#### 4.3.2 IMAQdx

Ensin suunnittelin sovelluksen, jossa hyödynsin LabVIEW-noodien lisäksi IMAQ-noodia (LIITE 14). Sovellus koostuu while-silmukasta ja noodista, jotka suoritetaan ennen ja jälkeen while-silmukan. Sovelluksen sisään tulevia syötteitä ovat "Session in" ja "Error in". Session in -parametrillä valitaan haluttu kamera. Omassa tapauksessa se oli USB-portin kautta kannettavaan tietokoneeseen liitetty "cam2". Sovelluksessa hyödynnetään valmiita virtuaalisia instrumentteja, joista ensimmäisenä ovat IMAQdx Open camera ja IMAQdx Configure Grab, joiden välissä on Property-noodi. IMAQdx Property -noodissa kameran asetusten attribuuteille annetaan haluttu arvo. LED-tarkastuksen kannalta tärkeää on määrittellä kameran asetuksista kirkkaus (brightness), valotus (exposure) sekä tarkennus (focus).

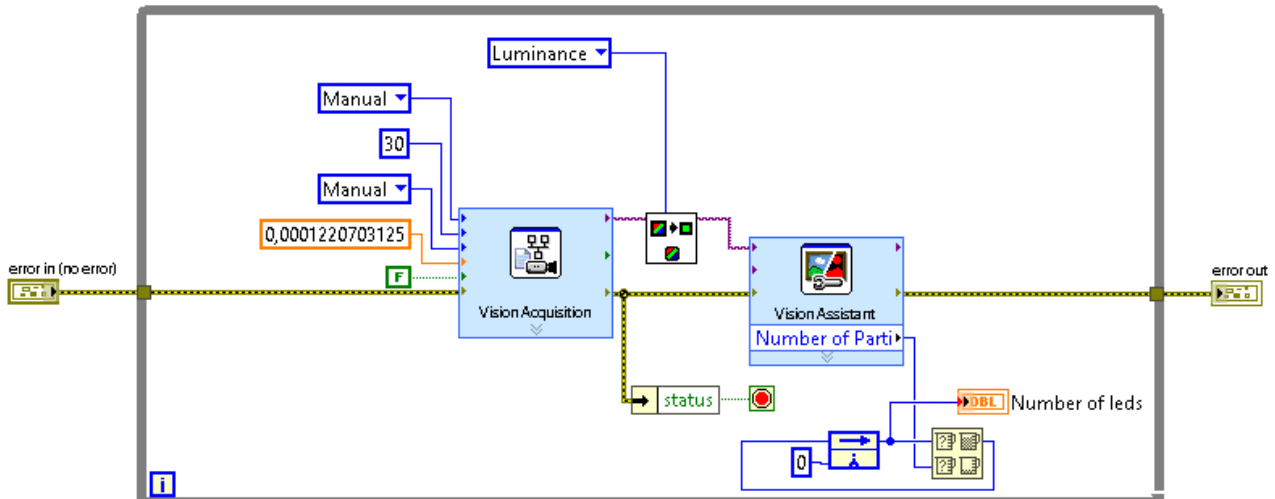
IMAQ Create VI muodostaa kuvalle maskin, johon Grab VI -noodissa upotetaan kameralta saatu digitalisoitu kuva. IMAQ Count Objects -noodia ennen kuva täytyy muuttaa mustavalkoiseksi luminance color plane -toiminnon avulla. IMAQ Count Objects 2 VI:ssä on settings-liityntärajapinta, mihin on annettava parametrit kuvan analysointi algoritmin suorittamista varten. Ensimmäinen parametri settings-ikkunassa on etsittävien objektien tyyppi: kirkas (bright) tai tumma (dark). Toinen parametri on kynnyisarvo harmaasävyn intensiteetille, jolla algoritmi kykenee erottelemaan valopisteiden rajat. Kolmannen parametrin ollessa aktiivisena kuva-

alueen reunoihin osuvat objektit hylätään. Neljännen parametrin ollessa aktiivinen objektien keskellä olevat aukot täydennetään. Seuraavat neljä parametriä määrittelee havaittujen objektien koolle suodattimen (filter). Filtrien avulla suodatetaan pois sellaiset objektit, jotka ovat kooltaan poikkeavan pieniä. Viimeiset kolme parametria määrittelee, millä tavoin löydettyt objektit esitetään kuvassa.

IMAQ Count Objects 2 VI antaa uloslöydettyjen objektien määrän numerotietona. While-silmukassa oleva objektien laskenta iteroidaan 30 ms:n syklillä, joten löydettyjen valopisteiden määrä vaihtelee johtuen erilaisista häiriötekijöistä. Toteutin maksimiarvon etsinnän Min & Max-komparaattorilla, joka vertailee annettuja arvoja ja luokittelee ne sen mukaan. Suurimman arvonetsinnän toteutin johtamalla Min & Max-noodin ulostulo takaisin sen sisääntuloon Feedback-noodi kautta. Feedback-noodi muistaa edellisen silmukan iteroinnin tuloksen Shift-rekisterin tavoin. Näin ollen Feedback-noodin palauttama arvo edustaa tässä sovelluksessa suurinta mitattua arvoa. Kun ohjelmaa kutsutaan NI Teststand -ohjelmassa, saatu maksimiarvo vastaa todellista syttyneiden ledien määrää.

#### **4.3.3NI LabVIEW Vision Assistant Express VI**

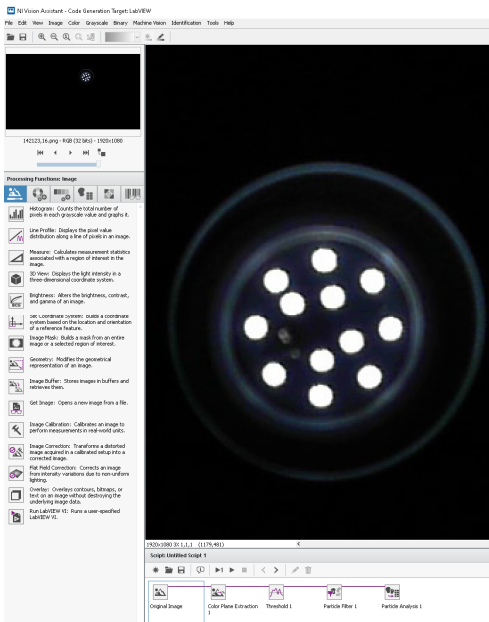
NI Vision Assistant Express VI:tä on kaksi eri funktiota, joita ovat Vision Acquisition ja Vision Assistant. Express VI:ta voidaan käyttää yksin tai yhdessä IMAQdx-noodien kanssa. Tarkemmin asiaa tutkiessa selviää, että Express VI:t muodostuvat IMAQ-noodeista. Express-työkalu tekee ohjelmoinnista kuitenkin huomattavasti suoraviivaisempaa. Kaksoisklikkaamalla lohko-kaavioon lisättyä Vision Acquisition -kuvaketta aukeaa Vision Acquisition Express -ikkuna. Jokaisella välisivulla on omat valinnat Vision Acquisition -algoritmin suorittamiseen (KUVIO 9).



KUVIO 9. NI Vision Assistant -lohkokaavio

Ensimmäisellä välisivulla valitaan kamera ja toisella välisivulla valitaan Express-toiminnolla luotavan ohjelman rakenne. Valittavissa on neljä eri tyyppiä: yksittäinen (single), jatkuva (continuous), rajallinen prosessoinnilla (finite with inline processing) ja rajallinen jälkiprosessoinnilla (finite post processing). Käytännössä nämä toiminnot luovat neljä erilaista rakennetta IMAQ-noodeista, joita ovat: ei lainkaan silmukkaa (single), while -silmukka (continuous), for -silmukka (finite with inline processing) ja for -silmukka, joka ottaa kuvan vain silmukan iteroituessa (finite post processing). Kolmannella välisivulla ovat kameran asetukset ja neljännellä sivulla kuvan kaappaukseen liittyvät valinnat. Viimeisellä välisivulla tehdään kontrollointi- ja indikointiasetukset, joilla määritellään, mitä liitäntäpisteitä luotavaan noodiin halutaan. (NI Vision 2011.)

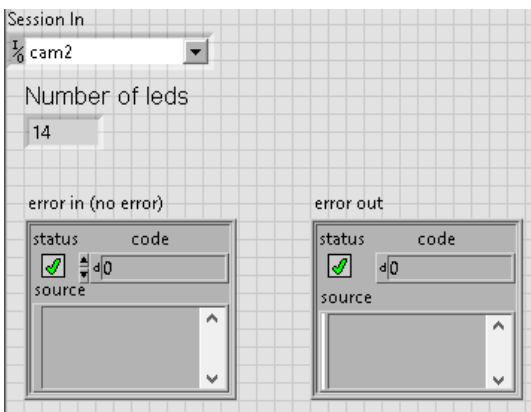
Käytännössä nämä kaksi eri tavalla toteutettua Visual Instrumenttia tuottavat saman lopputuloksen. Tarkemmin tutkiessani huomasin kuitenkin, että Vision Acquisition Express -toiminnalla toteutettu on for-silmukassa ja IMAQdx-noodeilla toteutetussa sovelluksessa Vision Acquisition on while-silmukan ulkopuolella (KUVA 17). Tämä johtuu siitä, että konenäköön liittyvä kuvan analysointi tapahtuu silmukan sisällä, toisin kuin Express-toiminnolla tehdyssä sovelluksessa, jossa kuvan analysointi tapahtuu uudessa silmukassa.



KUVA 17. Vision Assistant Express

Kaksoisklikkaamalla Vision Assistant -noodia, avautuu NI Vision Assistant -sovellus. Sovelluksessa on laaja valikoima kuvan prosessointiin ja analysointiin liittyviä työkaluja. Tässä sovelluksessa kuvan prosessointi etenee samaan tapaan kuin IMAQdx-noodeilla, joissa ensimmäisenä kuva muunnetaan mustavalkoiseksi Luminance Color Plane -toiminnon avulla ja seuraavaksi määritellään kynnyksarvot ja objektien suodattimet. Viimeisessä vaiheessa tehdään Particle Analysis, jossa lasketaan tunnistetut objektit.

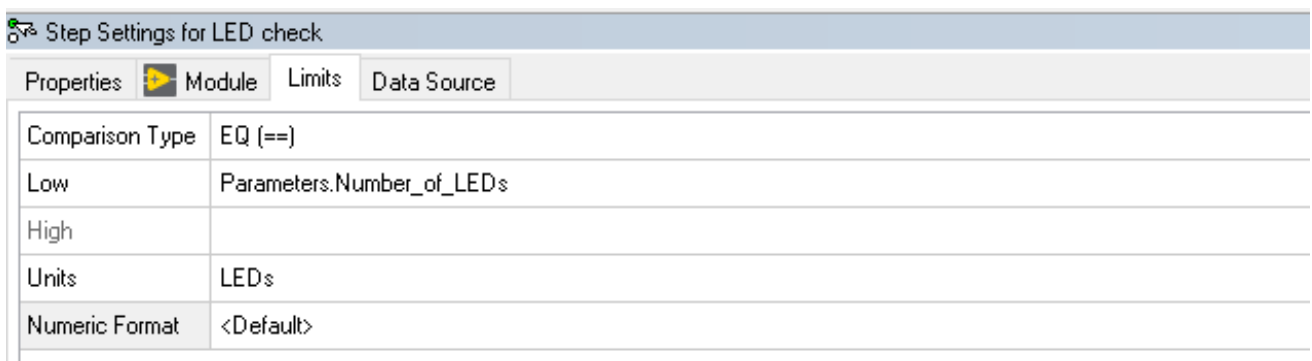
Ennen Vision Assistant -sovellusikkunan sulkemista on määriteltävä kontrollointi- ja indikointirajapintaan tulevat liitännät, jotka näkyvät lohkokaaviossa. Tässä sovelluksessa riittää, että näemme vain löydettyjen partikkeleiden määrän (Number of Particles) (KUVA 18).



KUVA 18. LED-tarkastus VI:n testaus



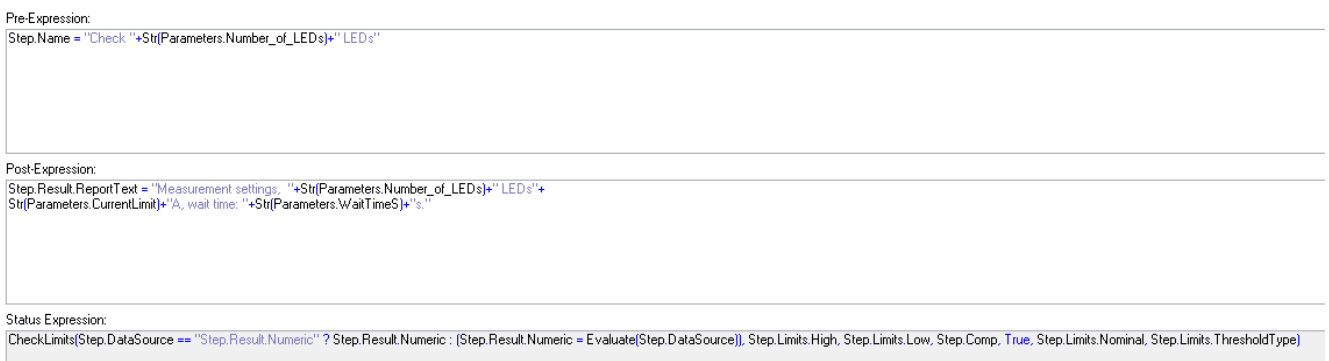
IMAQdx-noodeilla toteutetun VI:n toimivuutta kokeilin myös TestStand-toimintoympäristössä luomalla yksinkertaisen subsekvenssin. Subsekvenssi kutsuu tekemääni VI:tä ja vertaa löytyneiden ledien määrää parametriin "Number\_of\_LEDs" määriteltyyn arvoon. Parametriin "Number\_of\_LEDs" annetaan tuotekohtainen ledien määrä subsekvenssin kutsun yhteydessä TestStand main -ohjelmassa (KUVA 19). LED check step -toimintoa ennen on viive, jolla varmistetaan, että ledit ovat syttyneet, ennen konenäkö VI:n käynnistämistä (KUVA 19). TestStand Comparison Type EQ (==) vertaa mitattua (Result.Numeric) -arvoa parametriin "Number of Particles" ja antaa vertailun tuloksena "Pass" tai "Fail" stepin palautusarvoksi (Callback).



Step Settings for LED check			
Properties	Module	Limits	Data Source
Comparison Type	EQ (==)		
Low	Parameters.Number_of_LEDs		
High			
Units	LEDs		
Numeric Format	<Default>		

KUVA 19. LED check-stepin asetukset TestStand:ssa

Expression-välilähdellä määritellään toiminnot, jotka tehdään ennen ja jälkeen stepin suorittamisen. Funktiolla "Str()" saadaan tulostettua muuttujan arvo ymmärrettävässä muodossa testausraporttiin (KUVA 20).



Pre-Expression:	Step.Name = "Check "+Str(Parameters.Number_of_LEDs)+"LEDs"
Post-Expression:	Step.Result.ReportText = "Measurement settings: "+Str(Parameters.Number_of_LEDs)+"LEDs"+ Str(Parameters.CurrentLimit)+"A, wait time: "+Str(Parameters.WaitTime)+"s."
Status Expression:	CheckLimits(Step.DataSource == "Step.Result.Numeric" ? Step.Result.Numeric : [Step.Result.Numeric = Evaluate(Step.DataSource)], Step.Limits.High, Step.Limits.Low, Step.Comp. True, Step.Limits.Nominal, Step.Limits.ThresholdType)

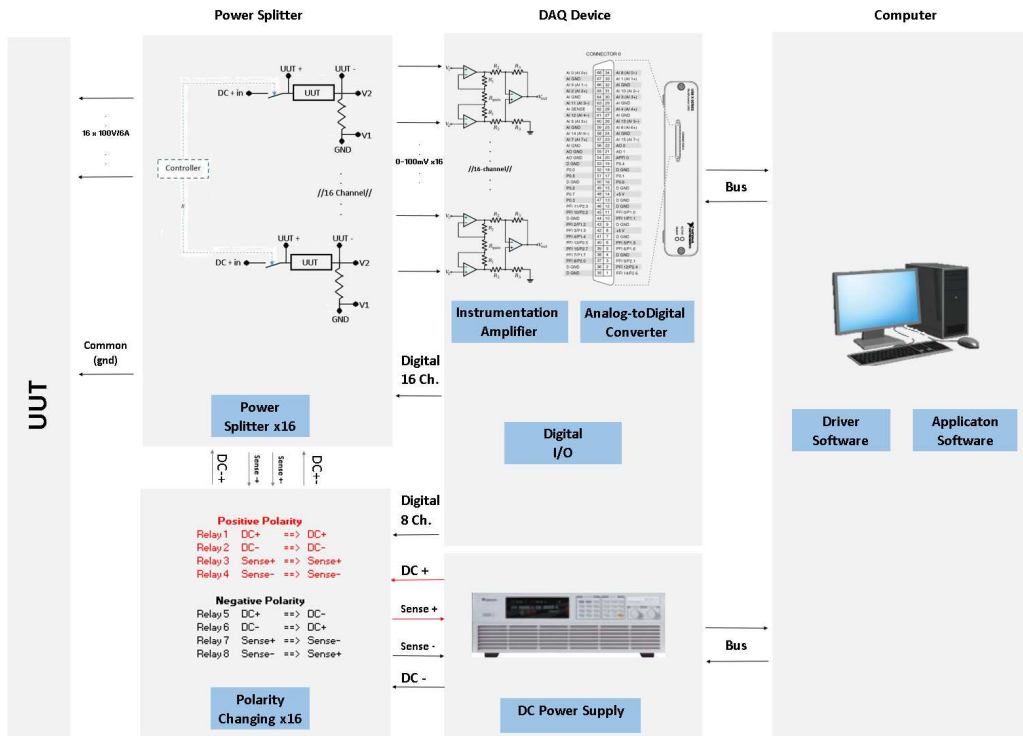
KUVA 20. LED check stepin Expression -asetukset

## 5 JATKOKEHITTÄMINEN

Laatua ja luotettavuutta kehittämällä saavutettiin tasaisempi jaksonaika prosessille, joten voidaan ajatella, että prosessin stabiloinnissa onnistuttiin prosessin standardisoinnin kautta. Alusta lähtien kaikille oli kuitenkin selvää, että koko aihio olisi pystyttävä testaamaan samanaikaisesti. Rakennetun testauslaitteiston rajallisuuden takia on testaus kuitenkin suoritettava yksi piirilevy kerrallaan. Kaikkien tuotteiden testaus samanaikaisesti vaatisi vähintään 10kW:n tehonsyötön, sekä 16-kanavaisen virranmittauksen. NI TestStand -testiympäristön päivittämiseksi kysyttiin tarjous järjestelmäntoimittajalta syksyllä 2020. Luottamuksellisista syistä en käy tarjouksen sisältöä läpi hinnoittelun osalta tässä insinööriyössäni.

Chroman valikoimassa on 10 kW DC-teholähteitä, jolla tehonsyöttö onnistuisi kaikkiin aihiossa oleviin tuotteisiin samanaikaisesti (KUVIO 10). Virranmittauksen voisi toteuttaa kanavakohtaisilla shunttivastuksilla, joista muodostuva jännitehäviö kerättäisiin NI DAQ -moduulin kautta tietokoneelle. Tällä järjestelyllä tuotekohtainen jaksonaika saataisiin ajallisesti samalle tasolle kuin mitä nyt kuluu koko piirilevyaihion testaamiseen. Maksimikoko aihiolle on 16 piirikorttia piirilevyaihiota kohden, joten yhden tuotteen testaamiseen kuluva aika saataisiin laskemaan teoriassa 16 sekunnista yhteen sekuntiin. Käytännössä jouduttaisiin lisäämään joitain uusia testisegmenttejä, jotta voitaisiin varmistaa, että jokainen tuote toimii halutulla tavalla. Tällä testauslaitteiston päivityksellä saavutettaisiin kuitenkin merkittävä parannus jaksoaikaan ja samalla voitaisiin hyödyntää jo rakennettuja tuotekohtaisia testijigejä.

Rakennettuihin testausjigeihin jouduttaisiin tekemään kaapeloinnin osalta muutoksia, sillä yhteisen (common ground) virran noustessa 50-120 A:iin tulisi maajohtimen poikkipinta-alaksi valita 10 – 16 mm<sup>2</sup>.



KUVIO 10. Jatkokehitys version lohkokaavio

## 5.1 Tasajännitelähde

Chroma 62000H -mallisarjaan kuuluu kolme varteenotettavaa DC-virtalähdettä, joiden ulostuloteho (Output Power) vaihtelevat välillä 10 kW-11,25 kW. Chroma 62100H-30 ja 62100H-40 tarjoavat enimmillään 30 V:n ja 40 V:n DC-ulostulojännitteen (Output Voltage), joten ne eivät sovellu tähän sovellukseen. Chroma 62100H-100P pystyy antamaan 0-100 VDC-jännitteen ja 0-250 A:n virran, mikä riittää 16 tuotteen rinnakkaisajoon. Chroma 62100H-100P DC-virtalähteen ulostuloteho on 10 kW. Tämä tarkoittaa, että 100V jännitteellä se kykenee syöttämään vielä 100 A:n virran testattavalle piirikorttialhiolle, sillä  $100 V * 100 A = 10 kW(VA)$ . Virta jakautuisi testattavien tuotteiden (UUT) kesken, kun testaus tapahtuisi rinnakkaisajossa, jolloin yhdelle tuotteelle syöttövirta olisi keskimäärin 100V/16piirikortilla, eli 6,25 A ja 24 V jännitteellä

teellä  $\frac{10 kW}{24 V} = 26$  Ampeeria. Vertailun vuoksi mainittakoon, että aiemmin käyttöön otetussa testausjärjestelmässä oleva Chroma 62006P-100-25 -teholähde kykenee tarjoamaan 25 A:n virran aina 24 V:n jännitteeseen saakka ja 100V:n maksimijännitteellä vielä 6 A:n virran.

## 5.2 Data Acquisition DAQ

National Instrument:n valikoimassa on USB-6361 DAQ, jossa analogisia sisääntuloja on 8 kelluvaa (differential)- tai vaihtoehtoisesti 16 yksipäistä (Single Ended) sisääntuloa. Differentiaalissa mittauksessa jännite-ero on "kelluva", mikä tarkoittaa, että ulkopuolelta tulevat häiriöt saadaan eliminoitua niiden kumotessa toisensa. Single-Ended-mittauksessa häiriöt ovat vain positiivisessa johtimessa, joten häiriöitä saattaa ilmetä. Tästä syystä instrumenttivahvistimen sijoittaminen ennen A/D -muunninta on oleellista, jotta mittauksen sisääntuloimpedanssi saadaan mahdollisimman alas samalla, kun jännite vahvistetaan A/D-muunninta varten. Virran mittausta varten jokaiselle kanavalle tarvitaan pieniohminen shunttivastus, jonka yli muodostuvasta potentiaalierosta (jännite) saadaan ohjelmallisesta laskettua virta-arvo vastaava jännite.

Rinnakkaisajon aikana testattavien tuotteiden ottama virta jakautuu niiden muodostaman kuorman mukaan. Nykyisessä testausjärjestelmässä teholähteen virtaraja (Current Limit) rajoittaa testattavan tuotteen saamaa virtaa vikatilanteessa, mutta suunniteltaessa rinnakkaisajoa tulee virranrajoitus toteuttaa ohjelmallisesti kanavakohtaisen virranmittauksen kautta. Pelkkä kanavakohtainen virranmittaus ei kuitenkaan riitä, vaan jokainen kanava on pystyttävä kytkemään pois virtarajan ylittyessä.

## 6 TULOKSET

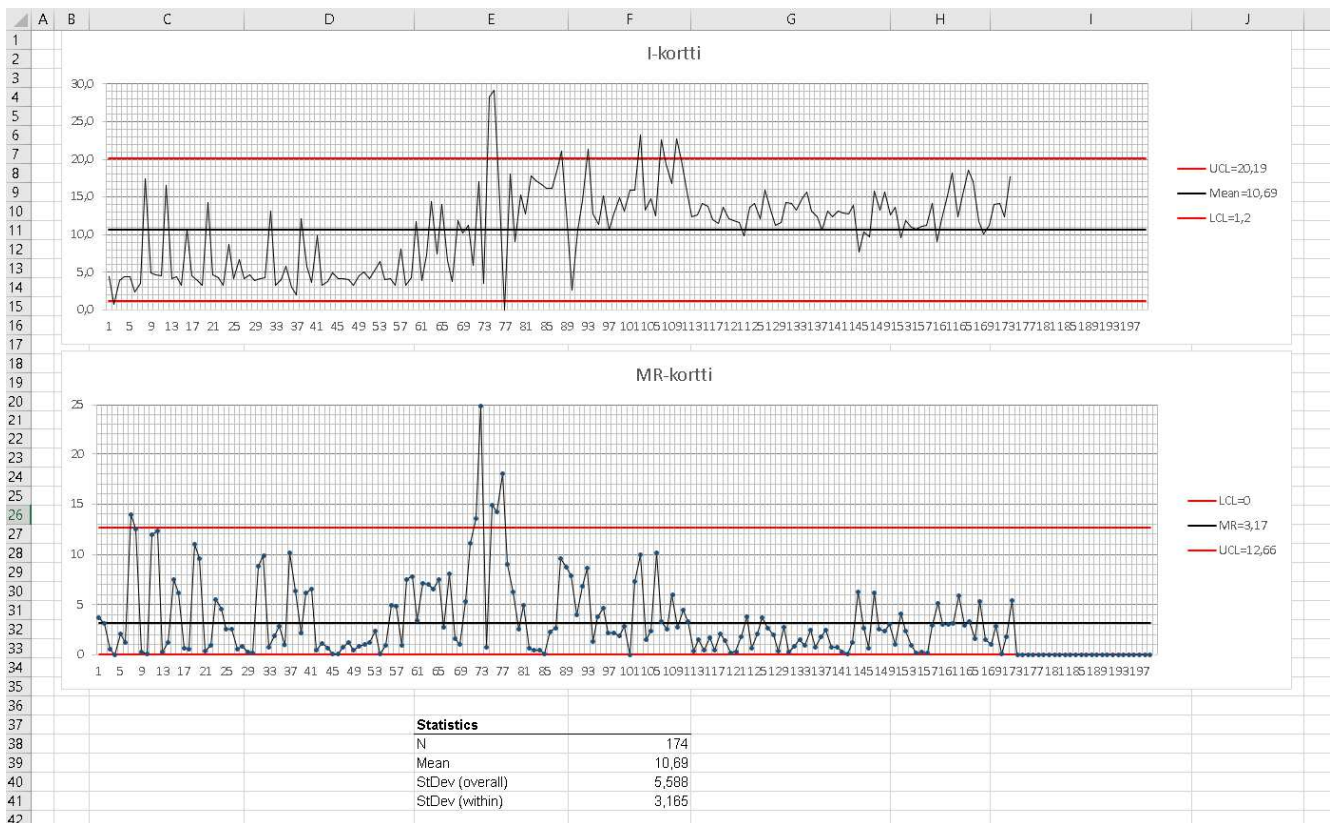
Rakennettu testauslaitteisto otettiin käyttöön vuosien 2019-2020 aikana vaiheittain. Testauslaitteistoa on ongelmien ilmetessä kehitetty ja kehitystyö jatkuu edelleen. Tavoitteena oli, että testeri olisi koko ajan käytettävissä ja tähän tavoitteeseen päästiin. Tuotekohtaisten testausjien ansiosta tuotteiden testaus on nopeutunut merkittävästi ja asiakasvaatimusten mukainen testaus on saatu standardisoitua osaksi testausprosessia. Testijigit ovat osoittautuneet helpokäyttöisiksi, eivätkä uudet työntekijät tarvitse pitkää käyttökoulutusta. Testauslaitteisto onnistuttiin suunnittelemaan niin, että työntekijän ei tarvitse omata erityisosaamista eikä tietämystä elektronisen laitteen toiminnasta.

Pysyvien tulosten takaamiseksi laadin I-MR-ohjauskortin Excel-tilukkolaskentaohjelmalla. Excel-tilukko hakee mittausdatan suoraan Oracle-tietokannasta (KUVIO 11), jonne kaikki toteutuneet työtunnit kirjataan ERP-järjestelmän kautta. Oracle-tietokantahaku on toteutettu seuraavasti:

```
SELECT SUM(tunnit)*3600/valmkpl AS TestausAika
FROM jrivi
INNER JOIN jtyo ON jtrivi.tyonro=jtyo.tyonro
GROUP BY tyonro
HAVING nimtunnus LIKE 'LED%' AND vaihekdi='100' AND krtunnus='5'
AND valmkpl > '1000' AND suvlmvko > '2018311'
ORDER BY suvlmvko
```

### KUVIO 11. SQL-koodi

SQL-tietokantahaku laskee yhden tuotteen testausajan ja tulostaa sen taulun "TestausAika" -sarakkeeseen. Haku tehdään kyseisen kuormitusryhmän valmiiksi kirjatuilta töiltä, joista on rajattu alle 1000kpl:een valmistuserät pois sekä ennen kesää 2018 valmiiksi kirjatut työt (KUVA 21). Pienemmät alle 1000kpl:een aloitussarjat on rajattu pois, jotta ne eivät vääristäisi mittaus-tuloksia. Havaintoarvoja voidaan pitää otoksena, sillä se edustaa testausprosessin vaihtelua 2018 kesältä lokakuuhun 2020.



KUVA 21. I-MR-Ohjauskortti

Upper Control Limit (UCL) ja Lower Control Limit (LCL)-kontrollirajat on laskettu seuraavasti:

$$UCL = \text{mean} + (3 * \sigma) \quad (8)$$

$$LCL = \text{mean} - (3 * \sigma) \quad (9)$$

Missä  $\sigma$  on keskihajonta, joka kuvaa mittausarvojen ryhmittymistä keskiarvonsa ympärille. Keskihajonnan laskentaan käytin Excelin funktiota STDEV.S, joka laskee soluun havaintoarvojen keskihajonnan. Keskihajonta kertoo siis, kuinka paljon joukossa on vaihtelua (YHTÄLÖ 10).

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}} \quad (10)$$

Missä  $\bar{X}$  on keskiarvo,  $X_i - \bar{X}$  on poikkeama ja  $n$  havaintojen lukumäärä (Karjalainen & Karjalainen 2002, 135).

Lisäsin Excel-taulukon välisivulle erityisyytaulun, johon ohjauskortista havaitun ohjausrajoiden ylittävät havainnot on kirjattava analysointia varten. Kontrollirajojen ylitys tarkoittaa, että

havainto ylittää  $6\sigma$  (Six Sigma) rajan ja vaatii tarkempaa analysointia. Ohjauskortista voidaan havaita ensimmäinen UCL-ohjausrajan ylitys keväällä 2019, jolloin uusi testausjärjestelmä otettiin ensimmäistä kertaa käyttöön. I-kortista voidaan myös havaita, että keskimääräinen testausaika on noussut uuden testerin myötä 5 sekunnista 15 sekuntiin. MR-kortista voidaan nähdä myös, miten samalla aikavälillä hajonta on vähentynyt uuden testerin käyttöönoton myötä, ja tämä kertoo osaltaan standardisoinnin onnistumisesta. Testausprosessi kerryttää kuitenkin edelleen keskeneräistä varastoa, sillä testaus vie edelleen enemmän aikaa kuin muut prosessin vaiheet.

Konenäön osalta en kerennyt saamaan tuotannosta tuloksia tähän insinööriyöhön. Suunnittelemani VI:n integrointi osaksi todellista testiympäristöä vaatii NI Vision Acquisition Software-lisenssin testiympäristön tietokoneelle sekä VI:n integroinnin testisegmenttiin. Saimme tarjouksen NI Vision Acquisition Software -lisenssistä sekä VI:n integroinnista nykyiseltä järjestelmän-toimittajalta, ja tutkimukset jatkuvat konenäön osalta edelleen tämän insinööriyön jälkeen.

## 7 JOHTOPÄÄTÖKSET JA POHDINTA

Insinööriyön tavoitteena oli parantaa testausprosessin kustannustehokkuutta, joka edellyttää vakautta kaikissa testausprosessin vaiheissa. Tämän insinööriyön tuloksena luotiin edellytykset kaikkien tuotteiden testaamiseksi asiakasvaatimusten mukaisesti. Tuotekohtaisten testausjigien ansiosta pystytään nyt testaamaan kaikki tuotteet samaan tapaan, joten testausprosessin standardisoinnin osalta onnistuttiin hyvin. Pysyviä parannuksia testausprosessiin onnistuttiin luomaan, ja I-MR-ohjauskortin avulla prosessinkyvykkyyden analysointi tarjoaa hyvät mahdollisuudet testausjärjestelmän jatkokehitykselle.

Tavoitteena oli myös kehittää olemassa olevaa järjestelmää ja luoda virtuaalinen instrumentti, joka analysoi testattavaan tuotteeseen syttyneiden ledien määrää toiminnallisen testauksen aikana. Tätä ei kuitenkaan päästy integroimaan osaksi valmistusprosessia, mutta LabView-ohjelmoinnin osalta onnistuttiin luomaan virtuaalinen instrumentti, joka voidaan myöhemmin integroida osaksi olemassa olevaa testausohjelmaa. Konenäköratkaisun kehittämistä vaikeutti NI Vision Acquisition Software -lisenssin puuttuminen ja myös osaltaan se, että testausjärjestelmä on päivittäisessä käytössä, enkä halunnut ottaa tutkimustyöstä johtuvaa riskiä tuotannon keskeytykselle.

Insinööriyön aikana sain syventää insinöörikoulutuksessa opittuja asioita, liittyen digitaalisiin ohjausjärjestelmiin, LabView -ohjelmointiin sekä konenäköratkaisuihin. Opinnäytetyöni on painottunut paljon tutkimushankkeen toteutukseen ja suurin haasteeni olikin kirjallisen tuotoksen kanssa. Tämän insinööriyön kirjoitus on jakautunut pitkälle aikavälille, kuten myös tutkimushankkeen, mikä on tehnyt työn rajaamisesta haasteellista.

Tutkimusaineistosta syntyi kattava käyttöohje testausjärjestelmän ylläpitoa ja vianetsintää varten. Kysytty tarjous testausjärjestelmän jatkokehittämiseksi yhdessä tämän insinööriyön kanssa antaa työkalut analysoida investoinnin kannattavuutta ja tarpeellisuutta. Tässä insinööriyössä en uuden testauslaitteiston tarjousta käynyt yksityiskohtaisesti läpi, mutta uskon, että laitteiston päivittämisellä yhdessä konenäköintegraation kanssa päästäisiin tavoitteeksi asetetun 10 sekunnin testattavaa tuotetta kohti. Teoriassa testausajassa voitaisiin päästä jopa muutamana sekunnin testausaikaan yhtä tuotetta kohden, kun kaikki piirilevyaihiassa olevat tuotteet voitaisiin testata samanaikaisesti.



## LÄHTEET

- Aunola, S. 2017. Arter: Blogi-kirjoitus. Prosessiarkkitehtuuri arvontuotannon apuna. Saatavissa: <https://www.arter.fi/prosessiarkkitehtuuri/>. Viitattu 22.10.2020.
- Beers, B. 2020. Fundamental analysis sectors & industries analysis, Electronics Sector. Saatavissa: [www.investopedia.com](http://www.investopedia.com). Viitattu 29.9.2020.
- George, M., Rowlands, D. & Kastle, B. 2004. What is Lean Six Sigma. United States, New York: George Group. ISBN10: 007142668X.
- Goldsby, T.J. 2009. Process Excellence Network: Six Sigma and the Seven Drivers of Cash Flow Saatavissa. <https://www.processexcellencenetwork.com/business-process-management-bpm/articles/six-sigma-and-the-seven-drivers-of-cash-flow>. Viitattu 23.10.2020.
- Heikkilä, T. 2014. Tilastollinen tutkimus. Helsinki: Edita Publishing OY.
- IPC-A-610F. 2014. Acceptability of Electronic Assemblies, Revision F. United States, Bannockburn, Illinois : IPC.
- Karjalainen, T., Karjalainen, E. 2002. Six Sigma: Uuden sukupolven johtamis- ja laatumenetelmä. Hollola: Quality Knowhow Karjalainen Oy.
- Knowledge NI 2020. Differences Between NI Vision Acquisition Software, NI Vision Builder, and the NI Vision Development Module. Saatavissa: <https://knowLEDge.ni.com/KnowLEDgeArticleDetails?id=kA00Z000000P6OASA0&l=fi-FI>. Viitattu 10.9.2020.
- Mack C.A. 2010. Semiconductor Lithography - The Basic Process. Saatavissa: [www.lithoguru.com/scientist/lithobasics.html](http://www.lithoguru.com/scientist/lithobasics.html). Viitattu 25.9.2020.
- National Instruments 2006. LabVIEW Environment Basics. Saatavissa: <http://www.ni.com/getting-started/labview-basics/environment>. Viitattu 27.8.2020.
- NI Vision 2011. NI Vision Assistant tutorial. Saatavissa: <http://www.ni.com/pdf/manuals/372228m.pdf>. Viitattu 18.9.2020.
- NI 2020. LabVIEW SubVIs Explained. Saatavissa: <http://www.ni.com/white-paper/7593/en>. Viitattu 23.1.2020.
- Peltola, V. 2019. Elektroniikka- ja sähköteollisuus, teknologiateollisuus. Saatavissa: <http://www.teknologiateollisuus.fi/fi/elektroniikka-ja-sahkoteollisuus>. Viitattu 27.9.2020.

Rother, M. 2011. Toyota Kata. Porvoo: A Bonnier Group Company.

Shewhart, W. A. 1924. Walter A Shewhart and the Hawthorne factory. Saatavissa: [www.ncbi.nlm.nih.gov/pmc/articles/PMC2464836/](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2464836/). Viitattu 29.9.2020.

Sirico Electronics 2018. Laatukäsikirja. Vain sisäiseen käyttöön. Viitattu 24.9.2020.

Tuominen 2010. Lean - Tehoa ja laatua siisteyden ja järjestyksen kehittämiseen - 5S  
ISBN: 9789522203083.

Uitto, J. 2015. Miten keskeneräinen tuotanto vaikuttaa? Saatavissa: <https://jesseuitto.fi/miten-keskenerainen-tuotanto-vaikuttaa/>. Viitattu 22.10.2020.

Wilson, L. 2015. How To Implement Lean Manufacturing. United States, New York City:  
McGraw-Hill LLC. ISBN: 978007183573.

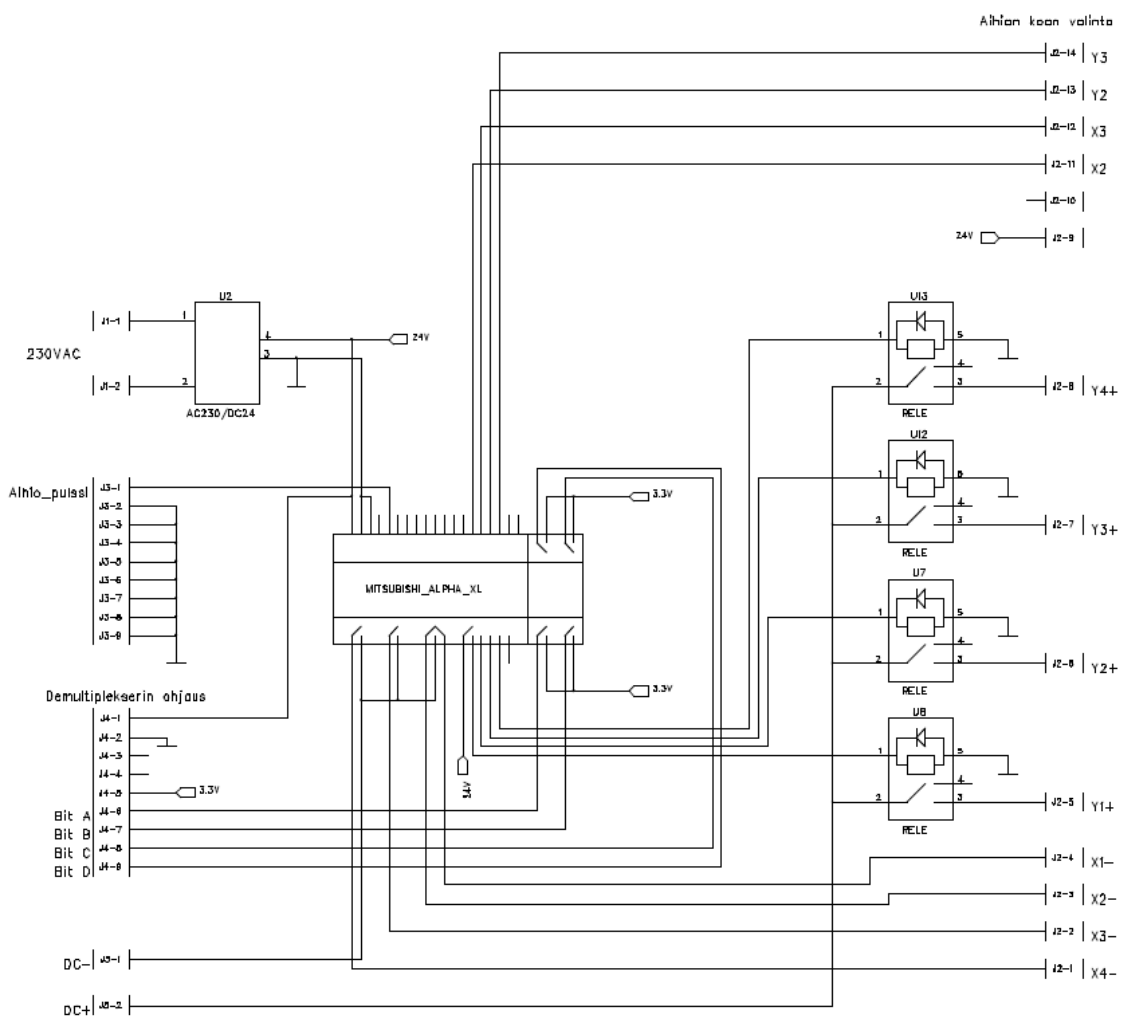
Liite salainen

## Testausjigin liittimen nastajärjestys

Littin	Toiminto
1.	X1 -
2.	X2 -
3.	X3 -
4.	X4 -
5.	Y1 +
6.	Y2 +
7.	Y3 +
8.	Y4 +
9.	24VDC
10.	
11.	X2
12.	X2
13.	Y2
14.	Y3

Aihio	X2	X3	Y2	Y3
2x2	<del></del>		<del></del>	
2x3	<del></del>			<del></del>
3x2		<del></del>	<del></del>	
4x2			<del></del>	
3x3		<del></del>		<del></del>
3x4		<del></del>		
4x3				<del></del>
4x4				
2x1			<del></del>	<del></del>

Logiikkaohjaimen kytkentäkaavio



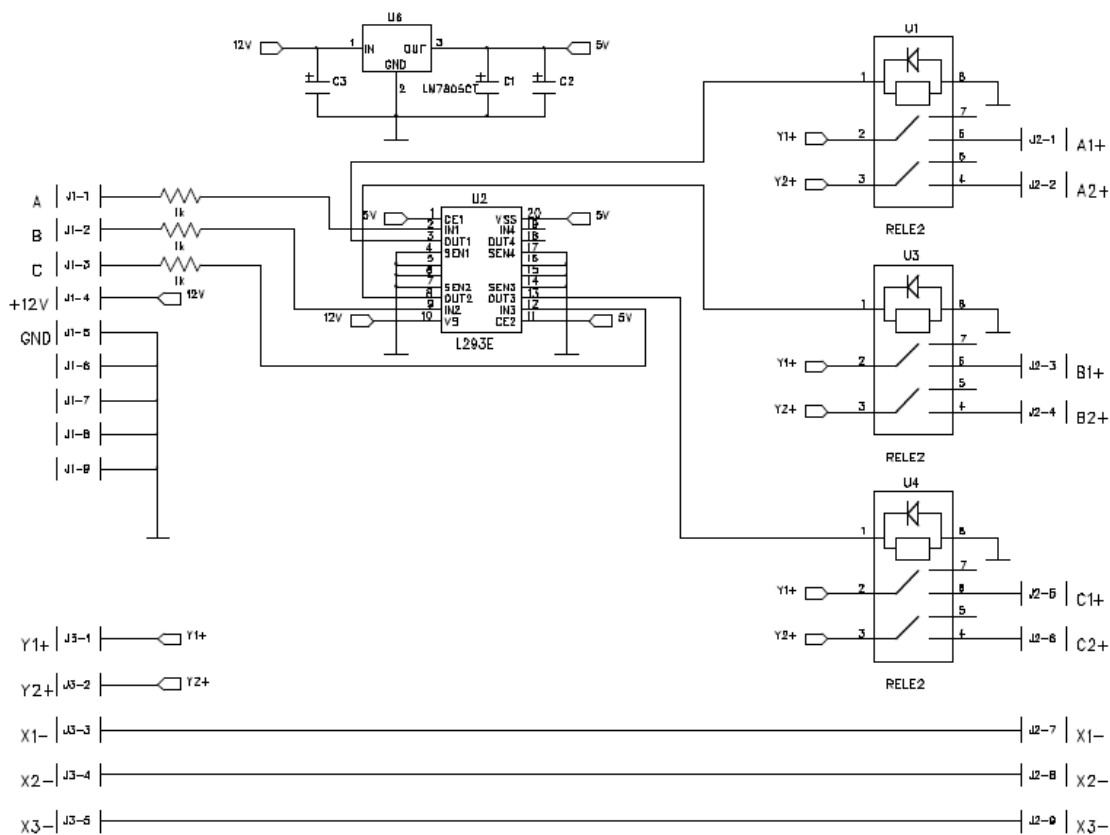
C

B

A

DRAWN: Toni Niemi		DATED: 20.6.2020		COMPANY: Toni Niemi			
CHECKED:		DATED:		TITLE: Mitsubishi Alpha XL logiikkaohjous			
QUALITY CONTROL:		DATED:		CODE:	SIZE:	DRAWING NO:	REV:
RELEASED:		DATED:		SCALE:		SHEET: OF	

LED-ryhmän ohjaus



Y1+ | J3-1 | Y1+

Y2+ | J3-2 | Y2+

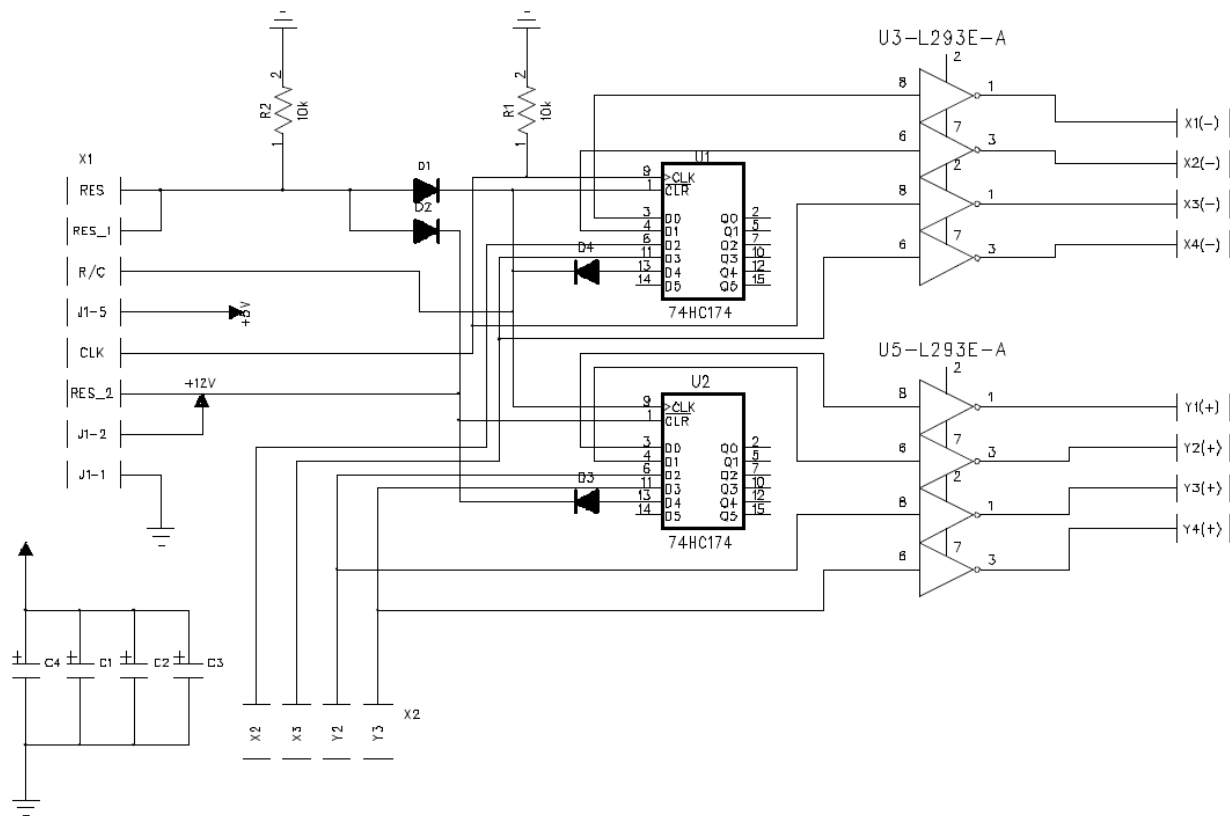
X1- | J3-3 | J2-7 | X1-

X2- | J3-4 | J2-8 | X2-

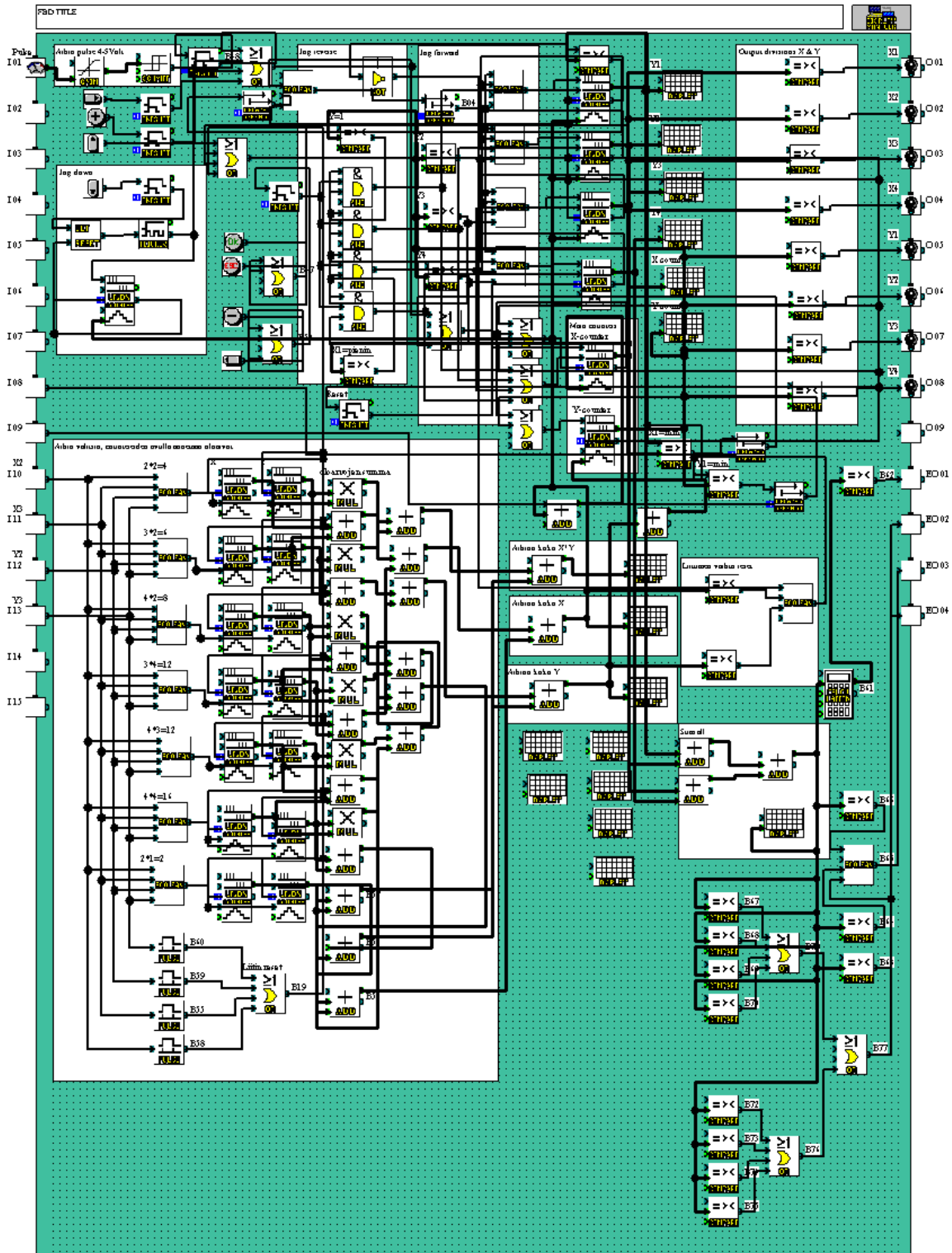
X3- | J3-5 | J2-9 | X3-

COMPANY: Toni Niemi			
TITLE: LED-ryhmän ohjaus			
DRAWN: Toni Niemi	DATED: 24.10.2020	CODE:	SIZE:
CHECKED:	DATED:	DRAWING NO:	REV:
QUALITY CONTROL:	DATED:		
RELEASED:	DATED:	SCALE:	SHEET: DF

Ensimmäinen version kytkentäkaavio



Mitsubishi Alpha Programming -toimilohkokaavio





## Teensy-ohjelmakoodi

```

/* Ohjelman alussa on luotu bittimuotoiset taulukot kaikille käytettävälle ulostuloporteille, joka mahdollistaa demultiplekserin tilan ohjaamisen poikkeavassakin järjestyksessä. Ennen varsinaista ohjelmakoodia asetetaan portit D, B ja C:n 2 viimeistä bittiä lähdeiksi sekä C-portin muut bitit tuloiksi. Seuraavaksi luodaan kaksi funktiota, joilla kirjoitetaan ja luetaan IO-portteja. Lähdön tila asetetaan ulostuloporttiin setPin-funktiossa parametrin luku pohjalta. ReadTila-funktiossa käydään läpi kaikki neljä sisääntuloporttia ja tila- muuttujaa kasvatetaan sisääntulobitin merkitsevyyden mukaan. Pääohjelma silmukassa (loop) kutsutaan vuorotellen luku- ja kirjoitusfunktiota ja jokaisen iteroinnin välissä on 200 ms:n viive.
*/

```

```

byte portD[] = {B00000000,
                B00000001,
                B00000010,
                B00000100,
                B00001000,
                B00010000,
                B00100000,
                B01000000,
                B10000000}; //D-portin bitit
byte portB[] = {B00000000,
                B00000001,
                B00000010,
                B00000100,
                B00001000,
                B00010000,
                B00100000}; //B-portin bitit
byte portC[] = {B00000000,
                B00000001,
                B00000010}; // C-portin bitit

int tila=0;
void setup() //Porttien alustukset, C-port 6-7 Output, 0-5 Input
{
  DDRD = B11111111;
  DDRB = B11111111;
  DDRC = B00000011;
}
void setPin(int luku) //Demultipleksaus kolmeen porttiin, tila muuttuja nollataan
{
  if(luku<8) {PORTD= portD[luku+1]; PORTC= portC[0]; PORTB= portB[0];}
}

```

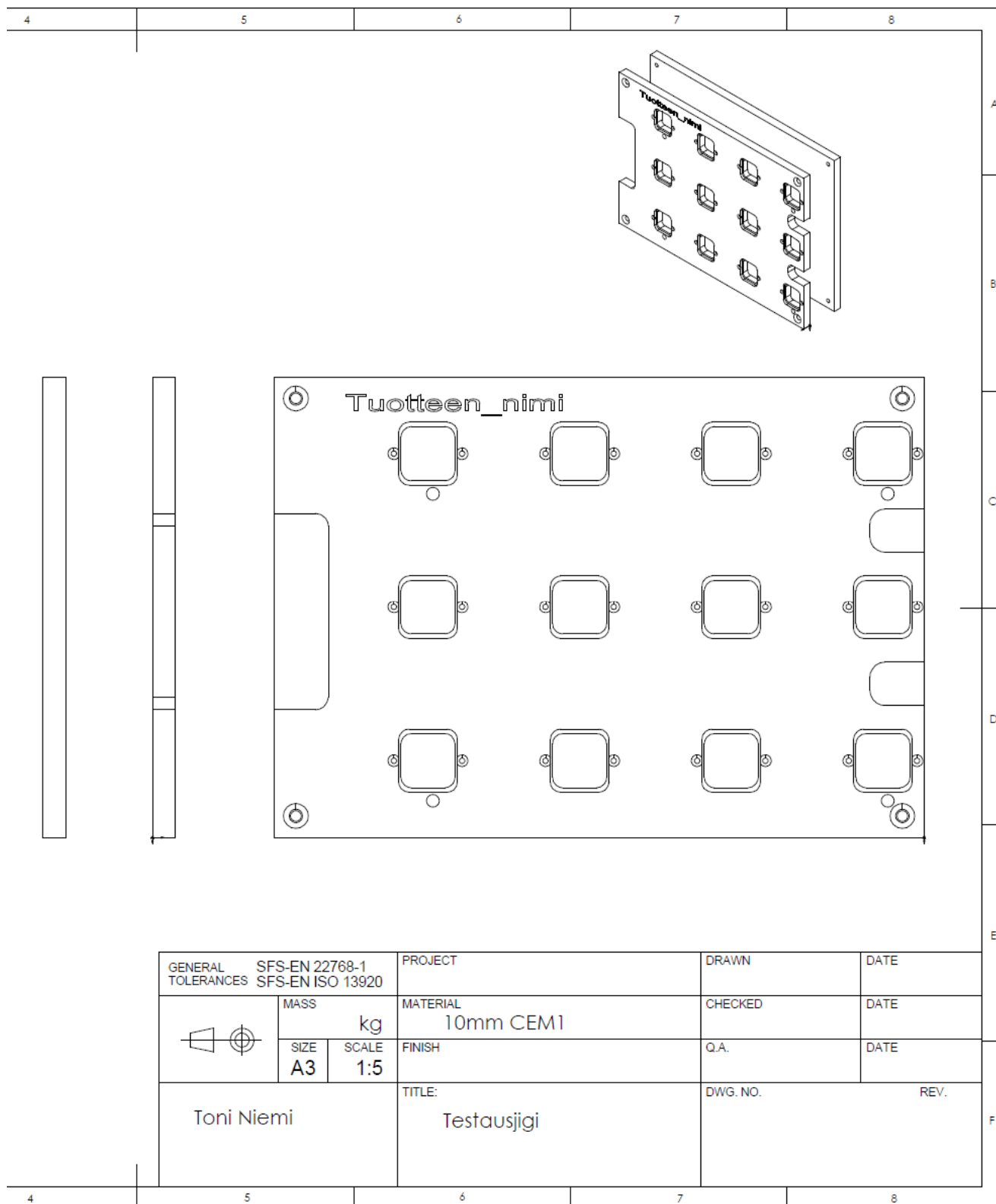
```
    if(luku>=8&&luku<=14) {PORTD= portD[0]; PORTC= portC[0]; PORTB=
portB[luku-7];}
    if(luku>=14) {PORTD= portD[0]; PORTC= portC[luku-13]; PORTB=
portB[0];}
    tila=0;
}

void readTila() //Binääri muutetaan decimaaliluvuksi tila muuttuun
{
    if(digitalRead(19)) tila += 1;
    if(digitalRead(18)) tila += 2;
    if(digitalRead(17)) tila += 4;
    if(digitalRead(16)) tila += 8;
}

void loop() //Luetaan ja kirjoitetaan vuorotellen
{
    readTila();
    setPin(tila);
    delay(200);
}
```



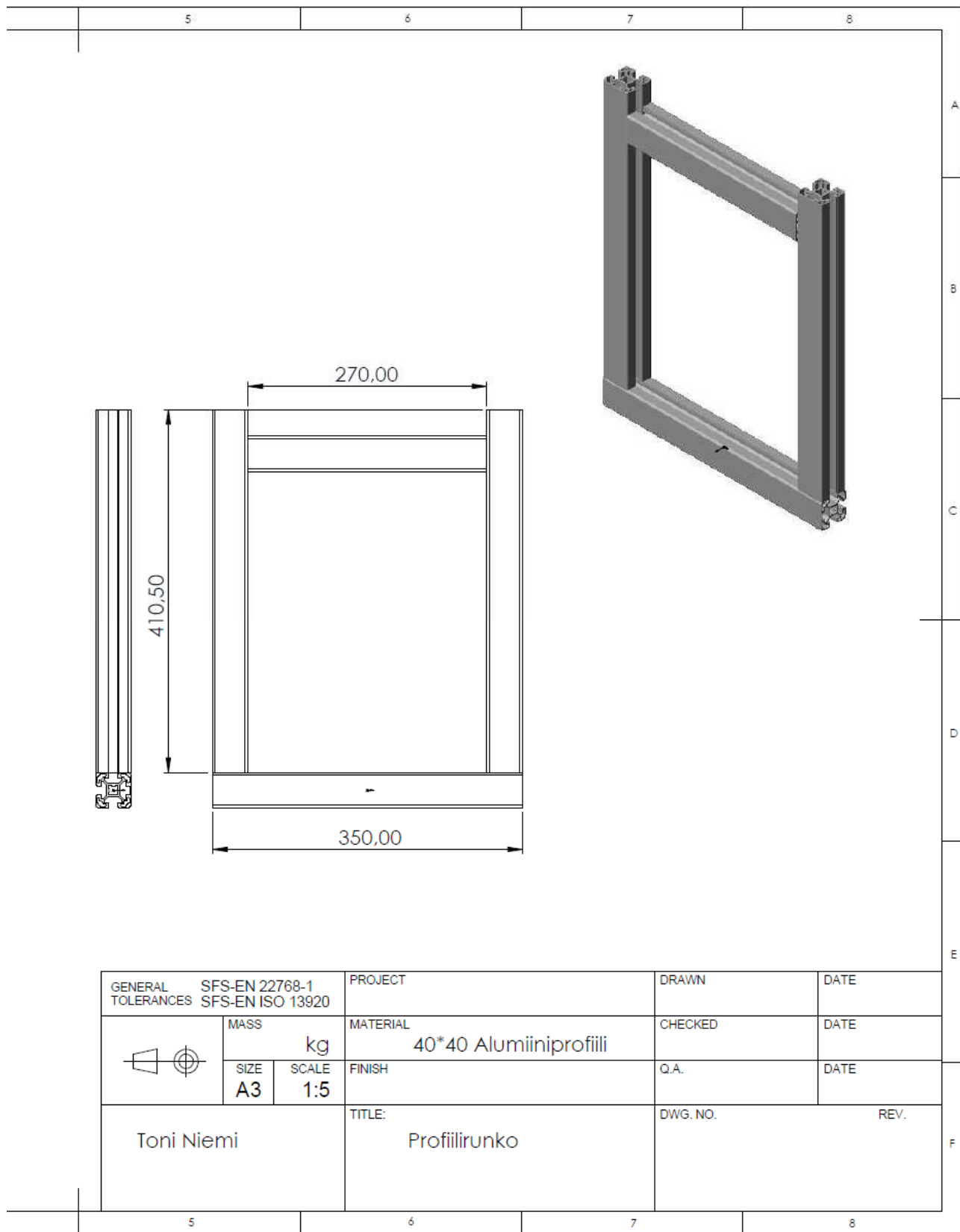
Testijigi



Liite salainen

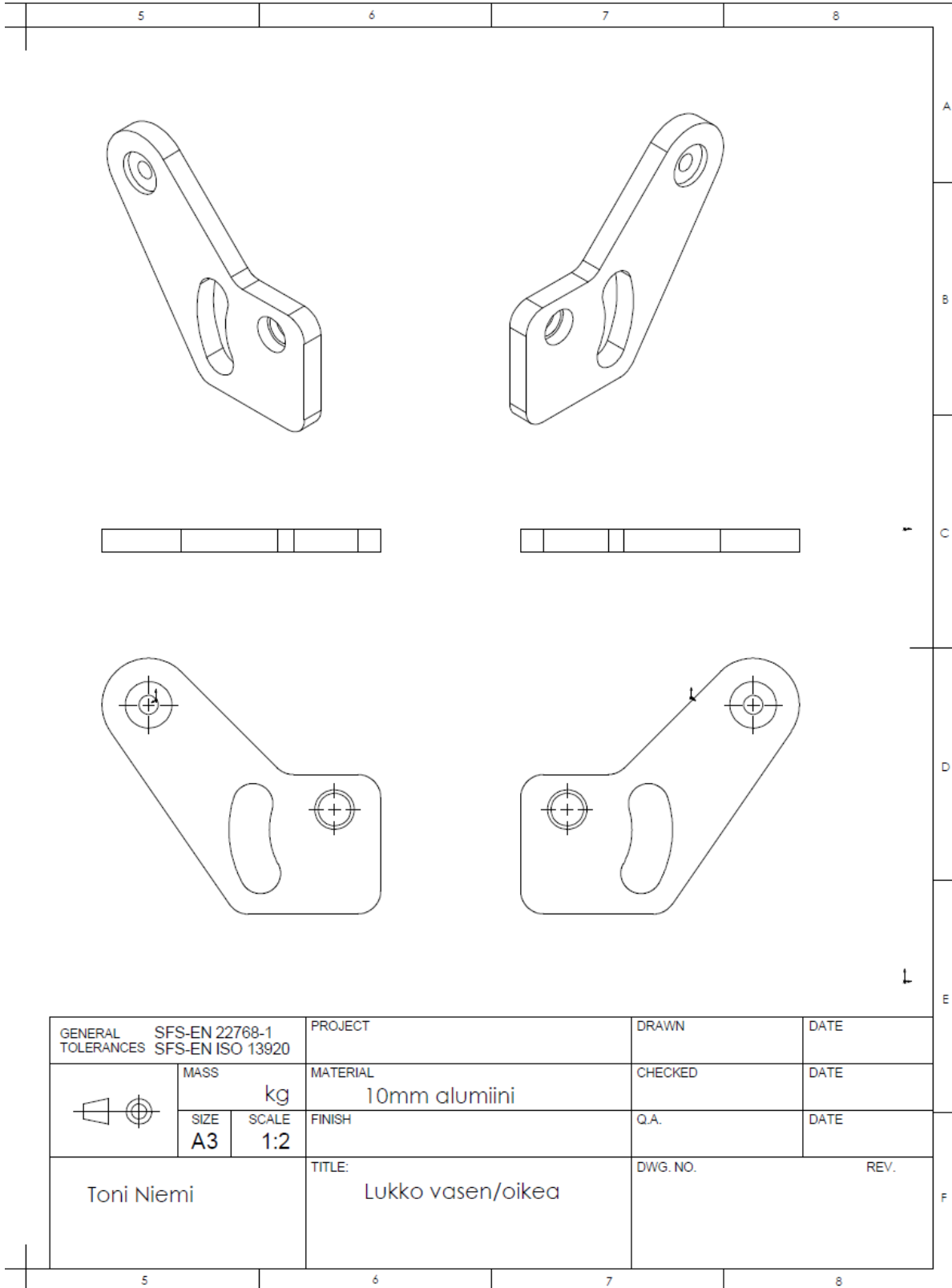
Liite salainen

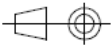
Profilirunko



GENERAL SFS-EN 22768-1 TOLERANCES SFS-EN ISO 13920		PROJECT		DRAWN	DATE
	MASS	MATERIAL		CHECKED	DATE
	kg	40*40 Alumiiniprofilii			
SIZE	SCALE	FINISH		Q.A.	DATE
A3	1:5				
Toni Niemi		TITLE: Profilirunko		DWG. NO.	REV.

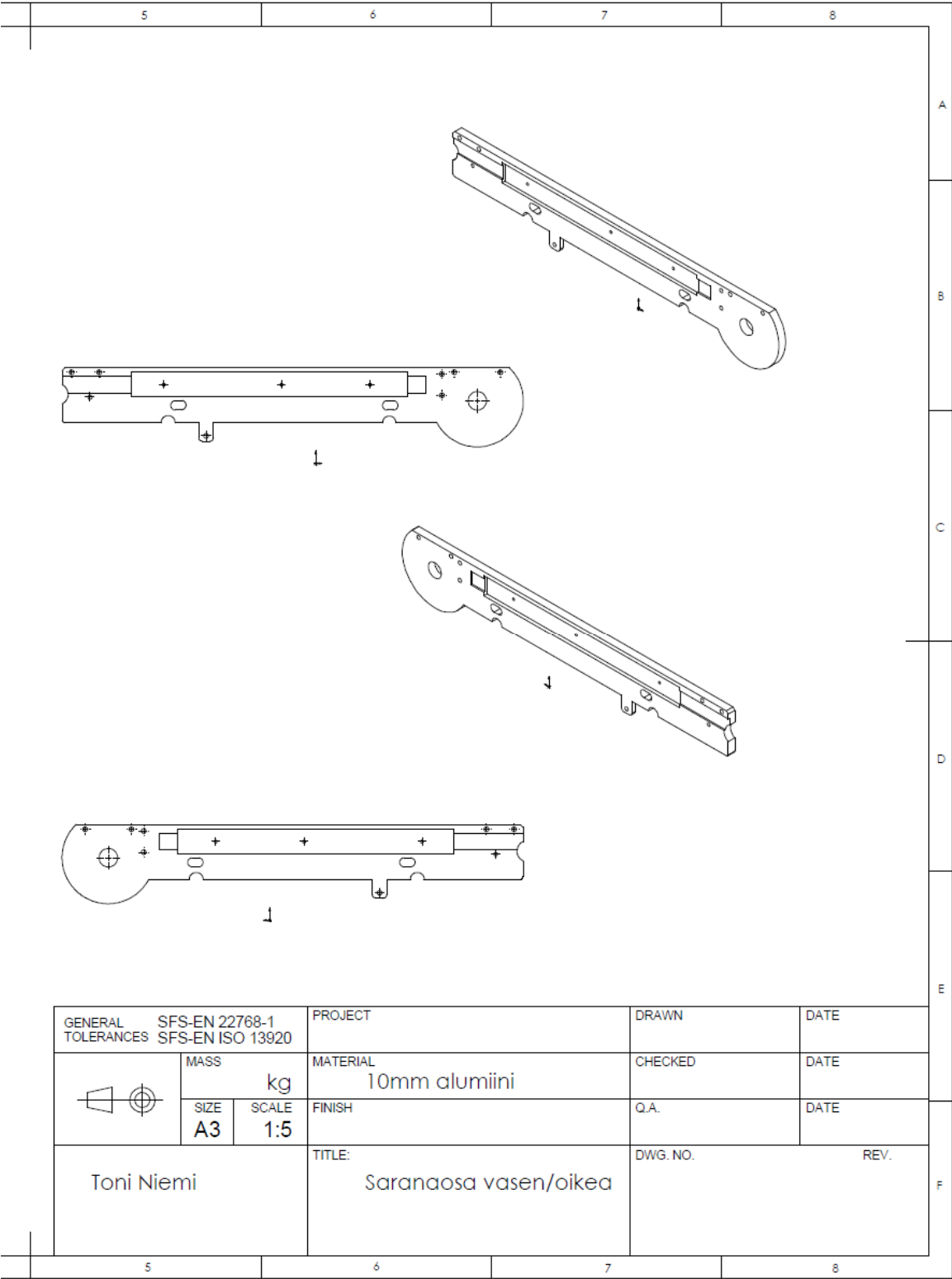
# Lukko



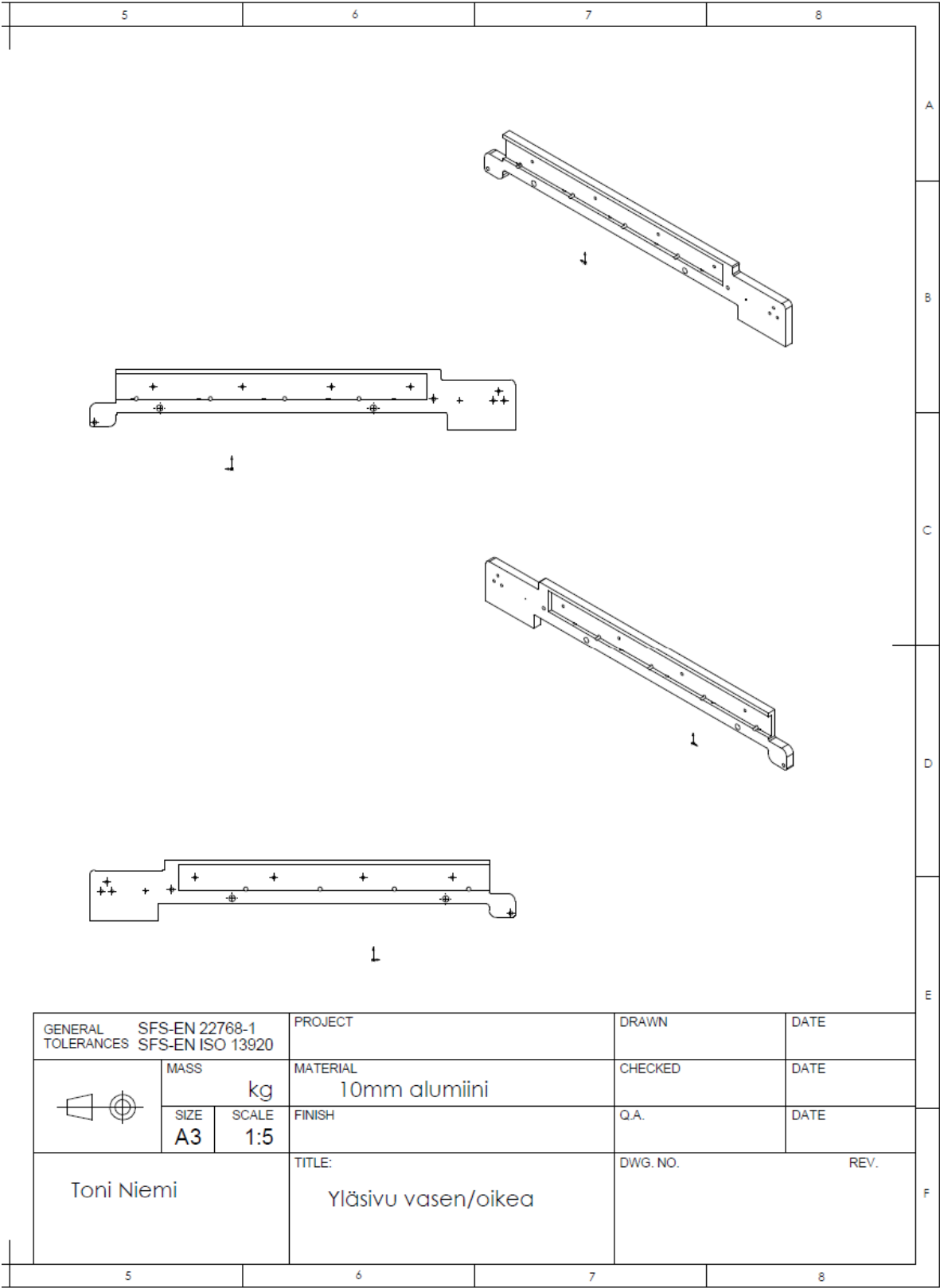
GENERAL SFS-EN 22768-1 TOLERANCES SFS-EN ISO 13920		PROJECT		DRAWN	DATE
	MASS	MATERIAL		CHECKED	DATE
	kg	10mm alumiini			
SIZE	SCALE	FINISH		Q.A.	DATE
A3	1:2				
Toni Niemi		TITLE: Lukko vasen/oikea		DWG. NO.	REV.



Saranaosa



Yläsivu



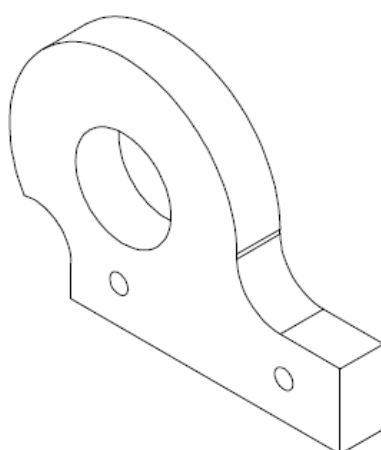
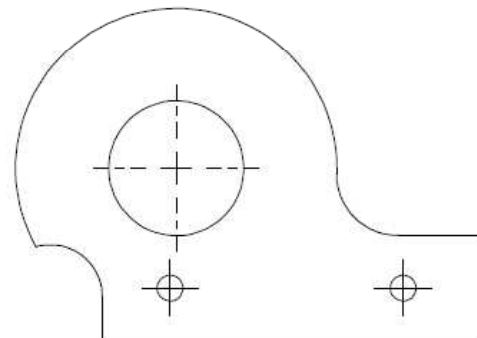
GENERAL TOLERANCES SFS-EN 22768-1 SFS-EN ISO 13920		PROJECT		DRAWN	DATE
	MASS kg		MATERIAL 10mm alumiini		CHECKED
	SIZE A3	SCALE 1:5	FINISH		DATE
Toni Niemi		TITLE: Yläsivu vasen/oikea		DWG. NO.	REV.

Saranalukko

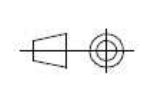
	5	6	7	8	
					A
					B
					C
					D
					E
					F
	5	6	7	8	

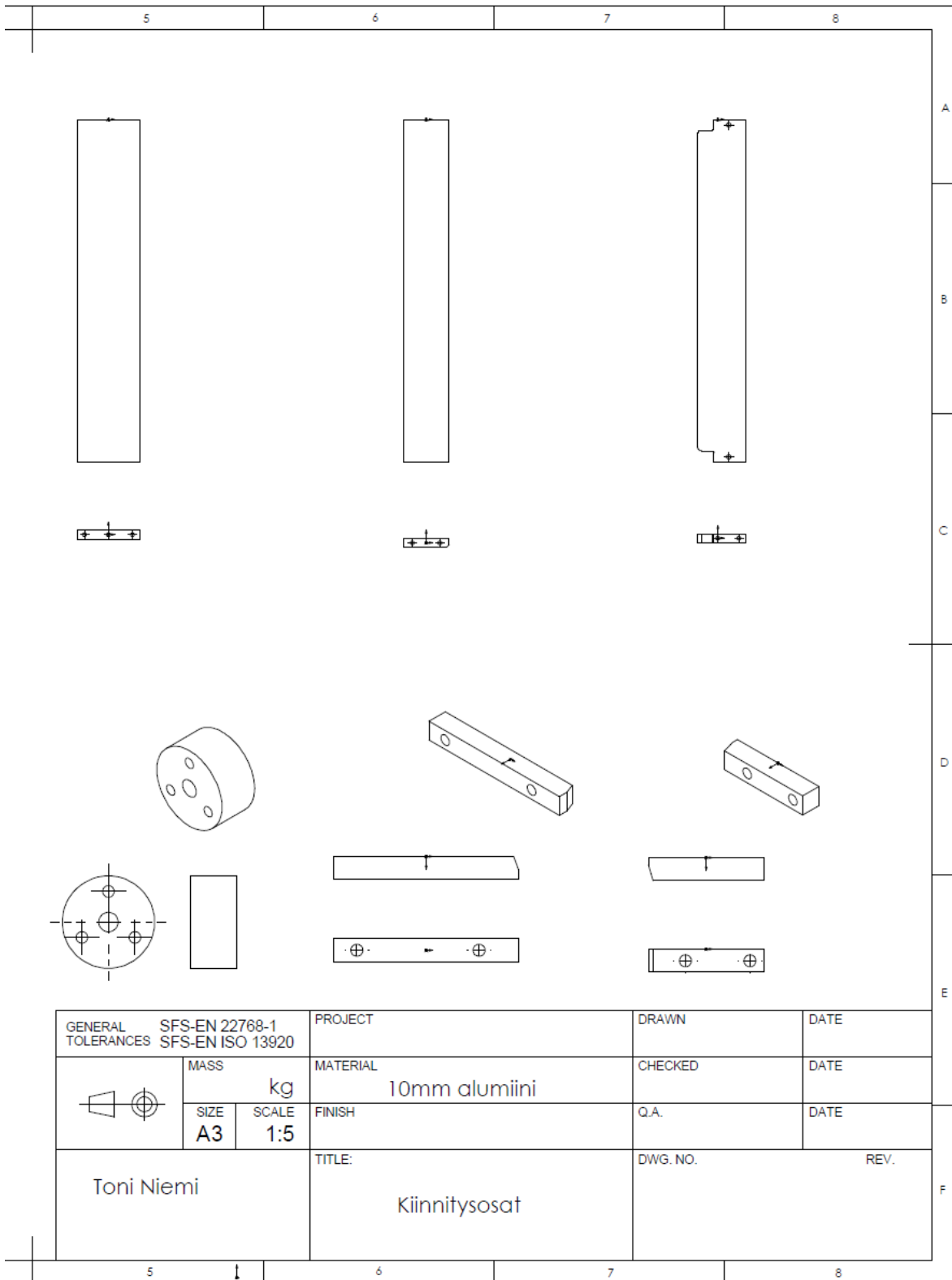
L

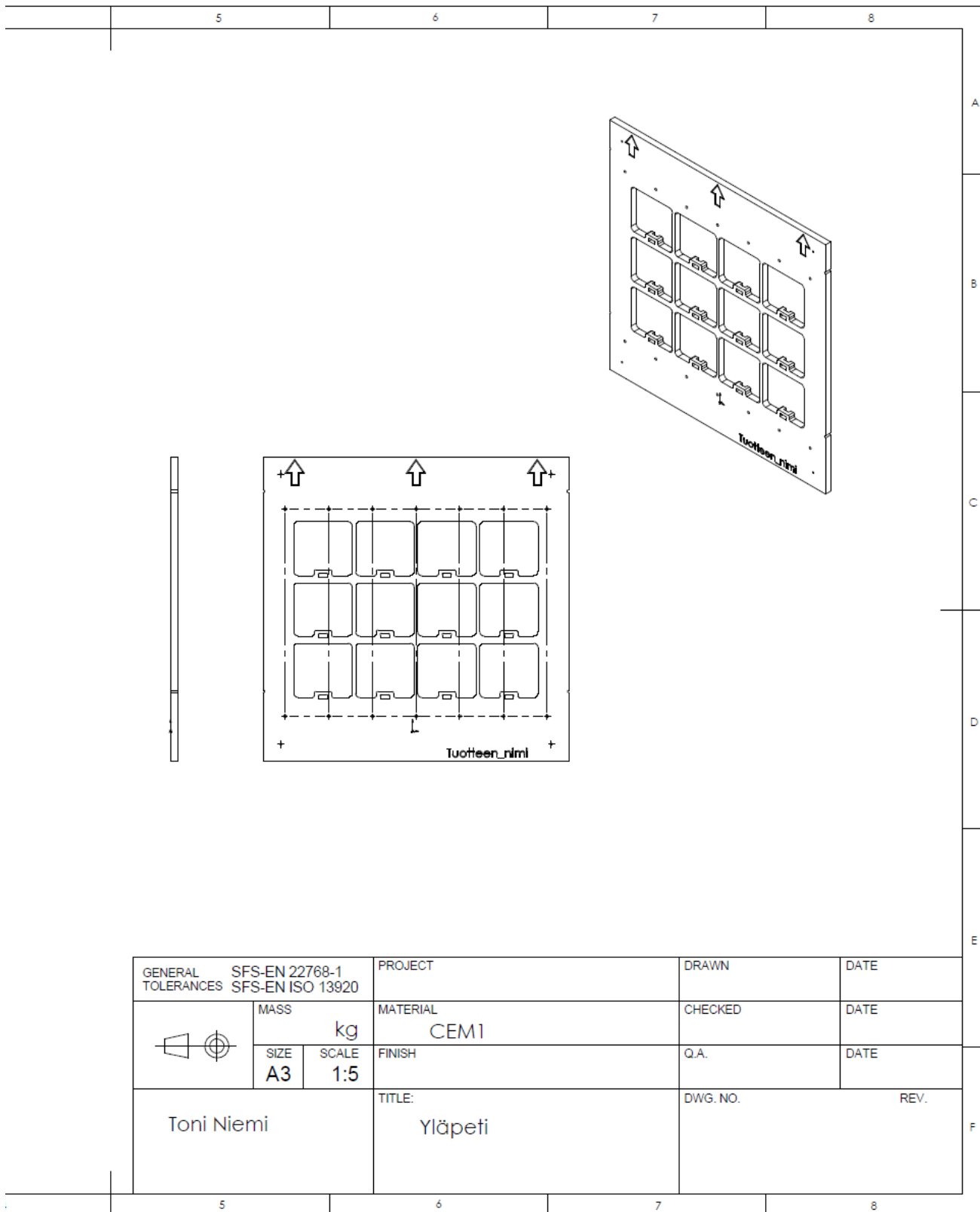
  

GENERAL SFS-EN 22768-1 TOLERANCES SFS-EN ISO 13920		PROJECT		DRAWN	DATE
	MASS	MATERIAL		CHECKED	DATE
	kg	10mm alumiini			
SIZE	SCALE	FINISH		Q.A.	DATE
A3	1:1				
Toni Niemi		TITLE:		DWG. NO.	REV.
		Saranalukko			

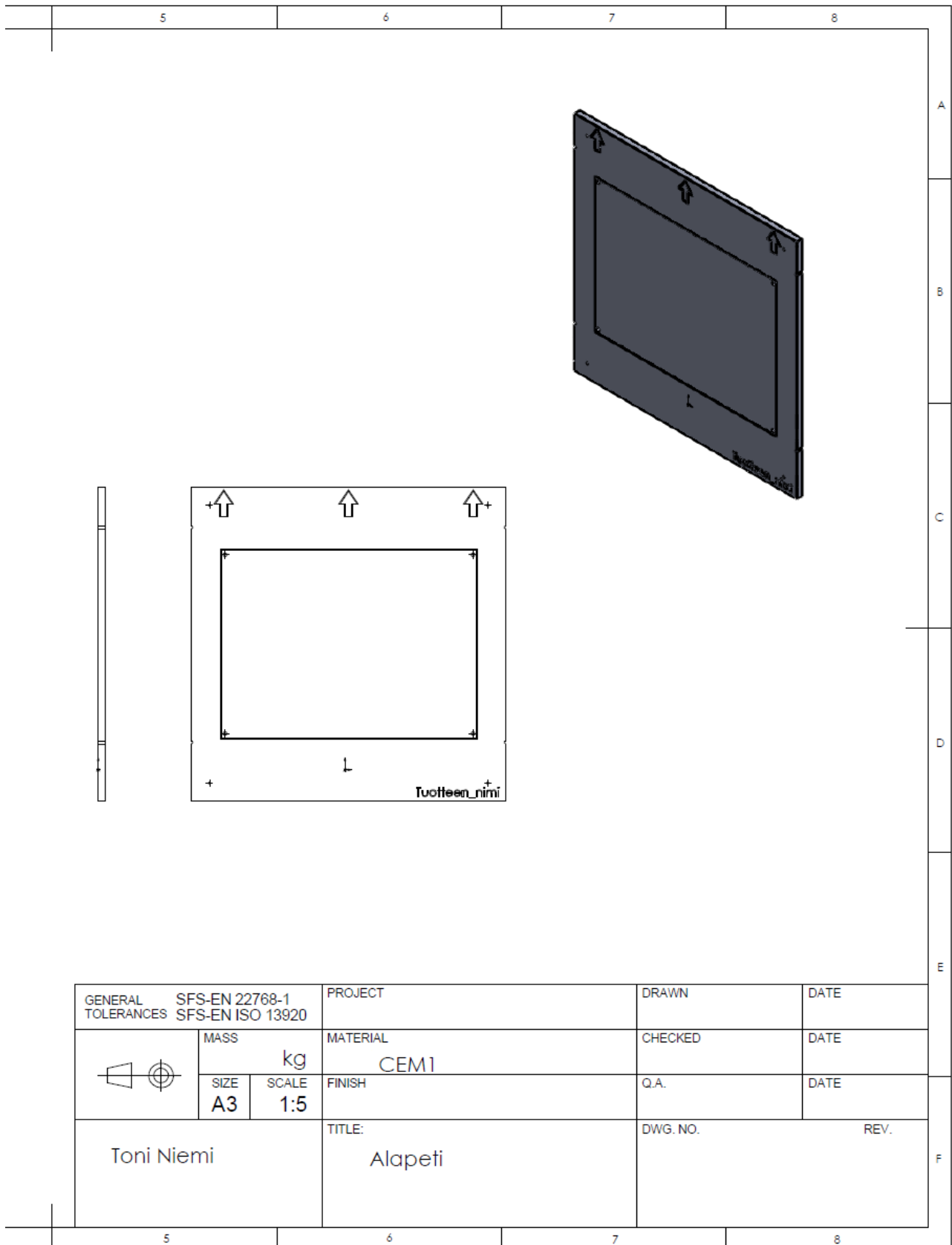
Kiinnitysosat



Yläpeti

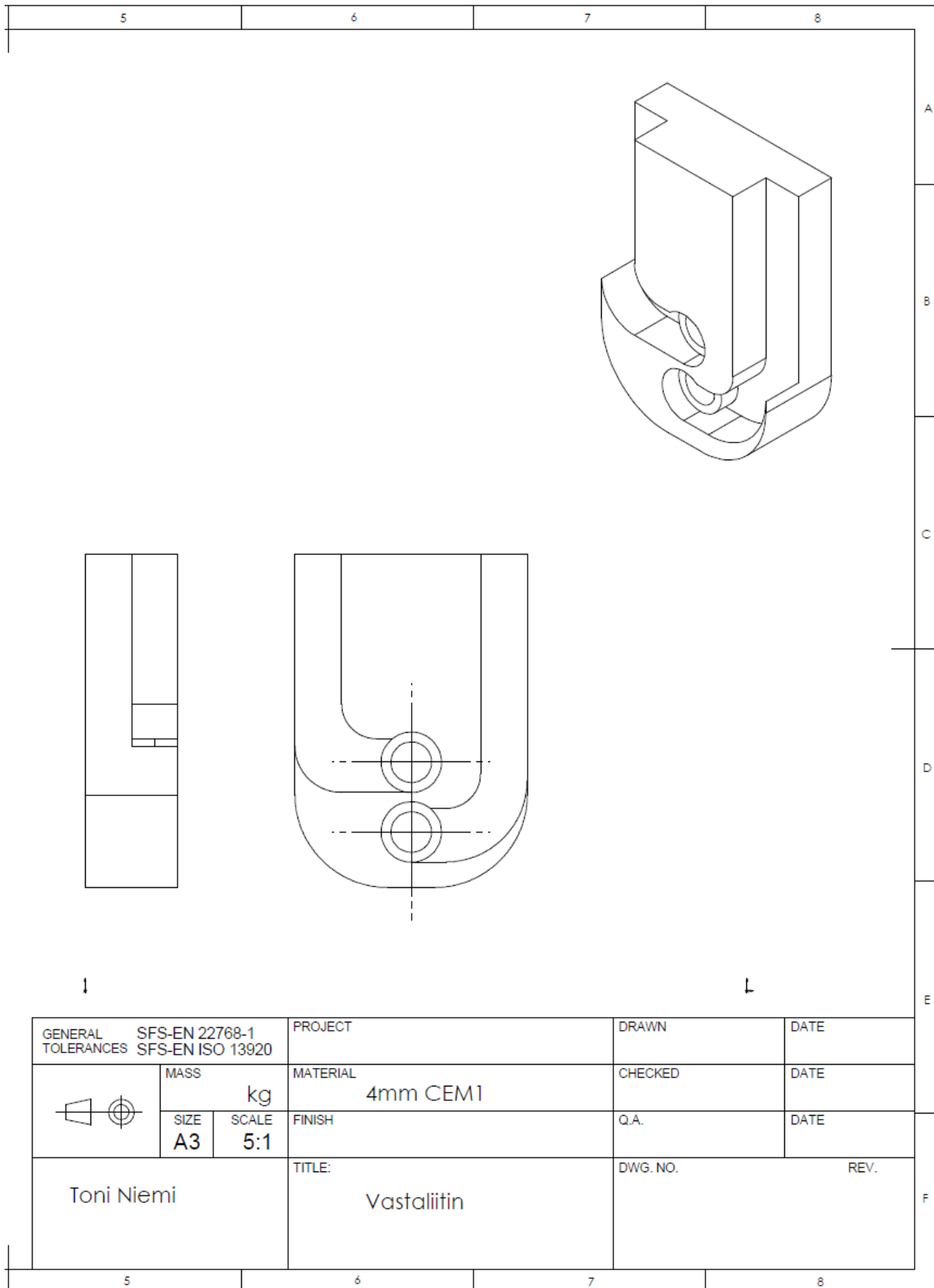


Alapeti



GENERAL TOLERANCES		SFS-EN 22768-1 SFS-EN ISO 13920		PROJECT		DRAWN		DATE		
	MASS		kg		MATERIAL		CHECKED		DATE	
	SIZE	SCALE		FINISH		Q.A.		DATE		
A3		1:5		CEM1						
Toni Niemi				TITLE:		DWG. NO.		REV.		
				Alapeti						

Vastaliitin



NI-IMAQdx -ajureilla toteutettu VI

