

Opinnäytetyö (AMK)  
Tietotekniikka  
Ohjelmistoliiketoiminta  
2015

Tomi Akrén

# LASERMERKKAUS- JA EROTTELULAITTEEN KÄYTTÖLIITTYMÄ



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Insinööri (AMK) | Ohjelmistoliiketoiminta

2015 | 35 sivua

Reijo Kärkäs, Janne Roslöf

Tomi Akren

## LASERMERKKAUS- JA EROTTELULAITTEN KÄYTTÖLIITTYMÄ

Tämä opinnäytetyö toteutettiin Murata Electronics Oy:lle, joka toimii MEMS-anturien valmistajana. MEMS-anturit ovat kiihtyvyyttä ja kulmanopeutta mittaavia antureita, joita käytetään autoteollisuudessa mm. ajonhallinan toteutukseen.

Opinnäytetyön aiheena oli suunnitella ja toteuttaa käyttöliittymä lasermerkkkaus- ja erottelulaitteelle LabVIEW-ohjelmointirajapintaa käyttäen. Ohjelmiston tuli hoitaa logiikan ja ohjelman välinen kommunikointi, vastaanotettujen tietojen analysointi sekä analyysin pohjalta komentaa logiikkaa.

Opinnäytetyössä suunniteltiin ja ohjelmoitiin LabVIEW-ohjelmistokokonaisuus, joka kommunikoi lasermerkkkaus- ja erottelulaitteen logiikan kanssa. Ohjelmistokokonaisuus sisältää TCP/IP-kommunikoinnin ja viestijonojärjestelmän toteutuksen. Ohjelmisto mallinnettiin logiikkakaavioihin joista saadaan selkeä kuva ohjelmiston kokonaistoiminnasta

Uusi lasermerkkkaus- ja erottelulaite on korvannut tuotannon vanhat laitteet. Toimintavarmuus on parantunut, tuotantotehokkuus kasvanut sekä operaattorikustannukset ovat lähes puolittuneet vanhoihin laitteisiin verrattuna.

ASIASANAT:

MEMS, TCP/IP, User Interface, UI Design

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor (Engineering) | Software business

2015 | 35 pages

Reijo Kärkäs, Janne Roslöf

Tomi Akrén

## LASER MARKING AND SEPARATION SYSTEM - USER INTERFACE

This thesis was commissioned for Murata Electronics Oy, which functions in the field of MEMS sensor production. MEMS sensors measure acceleration and angle speed, and they are mainly used in automotive industry e.g. in stability control systems.

The purpose of this thesis was to design and program user interface software for laser marking and separation system. The software should handle the communication to logic, analyze the received data and respond to logic accordingly.

A LabVIEW software was designed to communicate with the laser marking and separation system. The software package includes TCP/IP and queue communication. The software package was designed with a flow chart and diagram software to obtain a better view of the cross-functionalities.

The new laser marking and separation system has fully replaced the old systems. The reliability has improved, production efficiency is higher and operator expenses compared to old system are almost halved.

KEYWORDS:

MEMS, TCP/IP, User Interface, UI Design

# SISÄLTÖ

<b>SANASTO</b>	<b>6</b>
<b>1 JOHDANTO</b>	<b>8</b>
<b>2 LABVIEW</b>	<b>10</b>
2.1 LabVIEW	10
2.2 LabVIEW-ohjelmointi	10
<b>3 MERKKAUS- JA EROTTTELULAITE</b>	<b>12</b>
<b>4 SUUNNITTELU</b>	<b>14</b>
4.1 Toimintarakenne	14
4.2 Ohjelma	15
4.2.1 Ohjelmarakenne	15
4.2.2 Tapahtumakuuntelija	16
4.2.3 Taustaprosessit	17
4.2.4 Virheen käsittely	17
4.3 Käyttöliittymärakenne	18
<b>5 TOTEUTUS</b>	<b>20</b>
5.1 Käyttöliittymä	20
5.1.1 Alkumäärittelyt	20
5.1.2 Tuotantosivu	22
5.1.3 Huoltosivu	22
5.1.4 Saantoikkuna	23
5.1.5 Tapahtumaindikointi	24
5.1.6 Virheilmoitukset	24
5.2 Taustaprosessit	25
5.2.1 Viestijärjestelmä	26
5.2.2 TCP/IP-taustaprosessi	27
5.2.3 Makasiinin tarkistus	29
5.2.4 Siirtokehyksen tarkistus	29
5.2.5 Merkkauksen tarkistus ja erottelu	30
5.2.6 Laserin tehontarkastaja	31
5.2.7 Muut komennot	32

**6 YHTEENVETO****33****LÄHTEET****35**

## SANASTO

DUT	Device under test, testattava yksittäinen tuote.
Indikaattori	Tyypimääritely muistipaikka, joka toimii myös indikointiin (osana käyttöliittymää).
Klusteri	Ennalta määritely joukko eri- tai samantyyppisiä kontrolleja ja indikaattoreita, joille on määritely sama muistireferenssi. Klusteri on myös itsessään kontrolli ja/tai indikaattori.
Kontrolli	Tyypimääritely muistialue, joka toimii myös toiminnallisena painikkeena tai muuna osana (esim. taulukko) käyttöliittymää.
Makasiini	Alumiininen suojakehikko, jossa siirtokehukset kulkevat pitkin testauslinjaa. Yhdessä makasiinissa on 20 siirtokehystä.
MEMS	Micro Electro Mechanical System, Mikroelektromekaaninen järjestelmä.
Operaattori	Laitteen pääasiallinen käyttäjä.
Pacman	Tetrikseen liitetty kone, joka hoitaa tuotteiden sähköisen testauksen, 3D-mittaukset ja pakkaamisen kelalle.
Referenssi	LabVIEW-ohjelmiston yksilöllinen kohdistusnumero tiettyyn kontrolliin, indikaattoriin, tiedostoon tai tietokantayhteyteen.
Tetris	Laser-merkkäus- ja erottelukoneen nimi, johon käyttöliittymäohjelmisto on suunniteltu.

Siirtokehys	Metallinen kehys jossa tuotteet ovat. Pienin yksikkö, jossa tuotteet liikkuvat ennen erottelua. Tässä tapauksessa 16 tuotetta per siirtokehys.
Taustaprosessi	Ohjelmiston käynnistyksessä kutsuttava taustaohjelma, joka hoitaa tiettyjä ohjelmiston toiminnallisuuteen liittyviä asioita, kuten esimerkiksi TCP/IP-kommunikoinnin.

# 1 JOHDANTO

Murata Electronics Oy on maailman johtava passiivisten keraamisten sähkökomponenttien valmistaja. Suomen yksikkö on erikoistunut MEMS-tekniikan eteenpäinviemiseen ja Murata Electronics Oy:n valmistamat MEMS-anturit voit löytää yli puolesta maailman nykyaikaisista autoista. Murata Electronicsilla on yhteensä 48,288 työntekijää 23:ssa maassa. Suomen yksikössä (Murata Electronics Oy) on työntekijöitä yli 800 henkilöä [1].

MEMS-anturi (Micro Electro Mechanical System) on autojen ajonhallintajärjestelmän tärkein komponentti. Anturit tunnistavat auton kiihtyvyyksiä tai kulmanopeutta elementtityypistä riippuen. Tällä saadaan autosta tietoa, esimerkiksi ajonhallintaa tai mäkilähtöavustinta varten.

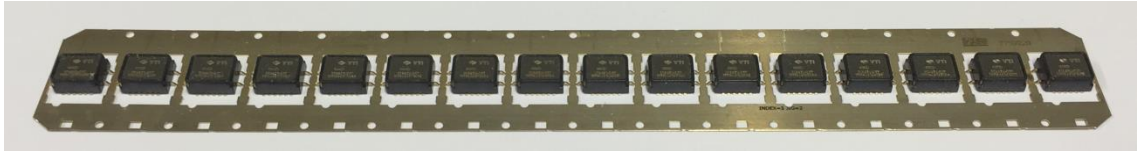
Anturien tekeminen vaatii erittäin tarkkaan hallittuja prosesseja. Anturit testataan kymmeniä kertoja ennen toimitusta asiakkaalle. Ne käyvät lävitse erilaisia testauksia, jotta voidaan olla varmoja siitä, että asiakkaalle päätyvät osat ovat aina toimivia ja luotettavia.

Tuotteet koostuvat anturielementistä ja integroidusta ASIC-piiristä kustannustehokkaassa, esivaletussa muovisessa kotelossa (Kuva 1). Anturit on suojattu ympäristövaikutuksien minimoimiseksi silikonigeelillä [1].



Kuva 1. Muratan Electronics Oy:n valmistama tuote.

Tuotteiden muovinen esivalettu kotelo on kiinnitetty kultapinnoitettuun siirtokehukseen (Kuva 2), jossa tuotteet siirtyvät koko testauksen läpi. Siirtokehuksesta irrotus tapahtuu merkkauk- ja erottelukoneessa. Siirtokehuksesta irrottamisen jälkeen tuotteita ei voida enää testata uudelleen.



Kuva 2. Siirtokehys

Analogisten MEMS-tuotteiden tuotantoa varten haettiin uutta lasermerkkauk- ja erottelulaitetta, koska vanhojen erillisten manuaalisten lasermerkkauk- ja erottelulaitteiden toimintavarmuus ja kapasiteetti huomattiin olevan tulevaisuudessa riittämätön tuotannon vaatimuksiin. Kilpailutuksen jälkeen uutta laitetta alettiin suunnittelemaan kuopiolaisen Eidtech Oy:n kanssa.

Laitteen ensimmäiset testit ajettiin alkuvuonna 2013 ja laite tuli tuotantotiloihin kesällä 2014. Tuotantoon käyttöönotto tapahtui heinäkuussa 2014.

Laitteen nimeksi tuli Tetris. Tetris-projektissa lähdettiin rakentamaan täysin tyhjältä pöydältä käyttöliittymän perusrakennetta. Vanhoissa tuotannon käyttöliittymissä oli erittäin raskaita ja epäloogisia näyttöjä, käyttöliittymät eivät olleet itseään ohjaavia, vaan käyttäjien piti lukea manuaaleista käyttöohjeet.

Työssä käydään läpi laitteen ja ohjelmiston välinen toimintarakenne, sen määrittely ja tapahtumat. Seuraavaksi määritellään käyttöliittymän toiminnalliset osiot ja selvennetään logiikkakaaviolla ohjelmiston sisäiset kommunikointirakenteet. Tämän jälkeen kuvataan käyttöliittymä, sen toiminnallisuudet käynnistyksestä virheen käsittelyyn, sekä taustaprosesseina toimivat ohjelmat.

## 2 LABVIEW

### 2.1 LabVIEW

LabVIEW on National Instrumentsin luoma ohjelmointiympäristö, joka perustuu G-kieleen. LabVIEW-ohjelmointi on suunnattu teollisuuden mittaus- ja testaussovelluksiin, mutta sitä voidaan myös käyttää yleisohjelmointikielenä hyvin pitkän kehityskaaren ja sen mukana tulleen monipuolisuuden tuloksena [2].

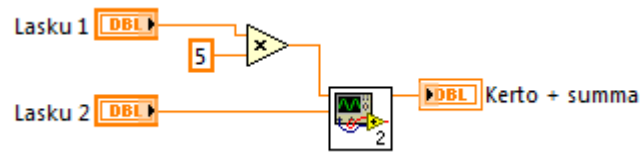
Ohjelmointikielenä LabVIEW on monipuolinen ja helppo oppia. Käytettävissä on muun muassa varmistetut jonot, tapahtumat, TCP/IP-tuki, ActiveX-komponentit, OpenGL-grafiikka ja erittäin laaja aliohjelmakirjasto. Myös avoimen lähdekoodin aliohjelmakirjasto (OpenG), on erittäin kattava ja laajentaa LabVIEW'n omaa aliohjelmakirjastoa sekä vähentää ohjelmointityötä huomattavasti.

### 2.2 LabVIEW-ohjelmointi

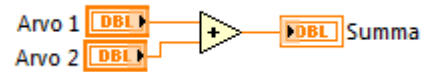
Ohjelmoinnissa on kaksi ikkunaa, lohkoikkuna (Kuva 3) ja etupaneeli-ikkuna (Kuva 5).

Ohjelmointi perustuu graafisessa muodossa esitettäviin laatikoihin, jotka ovat aliohjelmaa (Sub-VI) tai toimintalohkoja (summaaminen, vähentäminen, kertominen ...). Aliohjelmissa on myös aliohjelmaa ja toimintalohkoja. Aliohjelmiin vedetään "naruja" sisään ja ulos, jotka kuvaavat sisään ja ulospäin menevää datavirtaa. (Kuva 4)

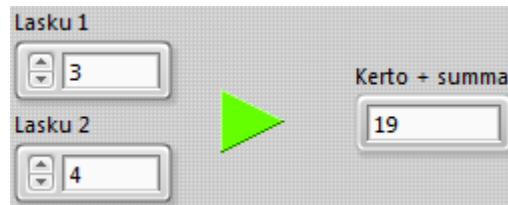
Datavirroilla on erivärisiä ja eripaksuisia "naruja". Datavirta menee oikealta vasemmalle ja narujen vedot päättävät datavirran kulun. Ohjelmaa voi kommentoida laatikoiden ja toimintalohkojen päälle nimilippuun sekä erillisenä lippuna, jolloin koodin lukeminen helpottuu.



Kuva 3. Aliohjelma pääohjelmassa



Kuva 4. Aliohjelman sisältö



Kuva 5. Etupaneeli

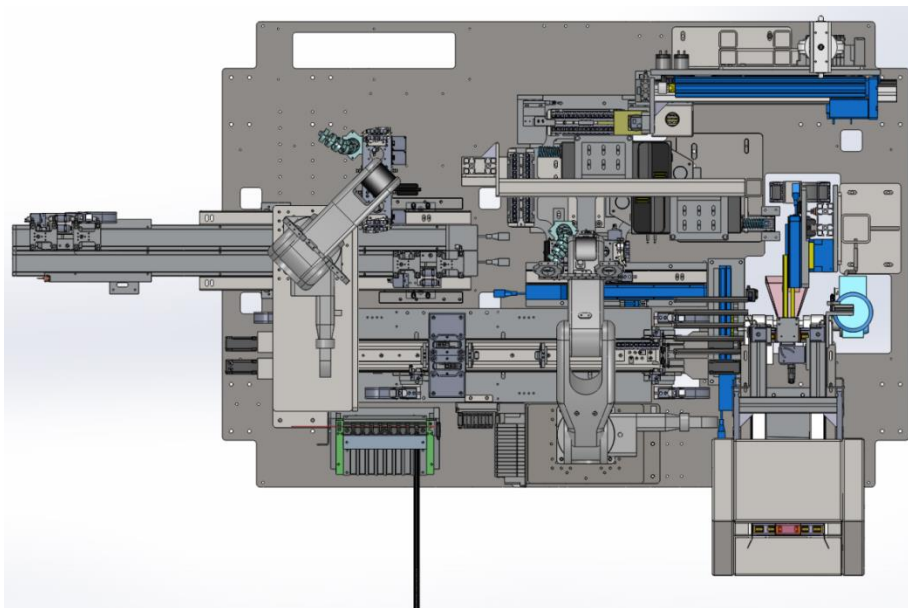
LabVIEW valittiin ohjelmointiympäristöksi tähän projektiin, koska kaikkien tuotantomme laitteiden operaattorille suunnatut ohjelmat on tehty kyseisellä ohjelmointiympäristöllä, joten yrityksellä on tarvittavat lisenssit ohjelmien tekemiseen ja tietotaitoa ylläpitämiseen ja mahdollisiin muutostöihin.

### 3 MERKKAUS- JA EROTTELULAITE

Laite sijoittuu tuotannon viimeiseen vaiheeseen, jossa tuotteen tulee olla testattu jokaisessa tarpeellisessa laitteessa ja mittapisteessä. Kun tuote erotellaan siirtokehyksestä, ei tuotetta enää voida testata uudelleen. Testauslaitteet hyväksyvät testaukseen ainoastaan siirtokehyksessä olevia tuotteita. Juuri ennen erottelua tuote on kalleimmillaan, eli tuote on jo käynyt läpi koontivaiheen sekä kaikki mahdolliset testaukset, jolloin hyvän tuotteen ennenaikainen erottelu on todella kallista yritykselle.

Erottelyn jälkeen tuote siirretään putkiin tai tuotekelalle. Tämän jälkeen tuotteet siirtyvät välivarastoon, josta ne menevät asiakkaalle tai mahdollisiin laboratoriotesteihin.

Merkkaus- ja erottelijalaite (Tetris) on Eidtech Oy:n Murata Electronics Oy:lle suunnittelema ja valmistama mittatilauslaite (Kuva 6).



Kuva 6. Lasermerkkäus- ja erottelulaite suunnittelukuva

Laitteessa on 2 Denso-robottikättä, jotka siirtelevät tuotteita tarvittaviin pisteisiin. Merkkauksen ja erottelun jälkeen laite siirtää tuotteet siirtokelkkaa pitkin pakkauskoneelle, joka hoitaa sähköisen mittauksen, 3D-kuvauksen sekä pakkaamisen tuotekelalle.

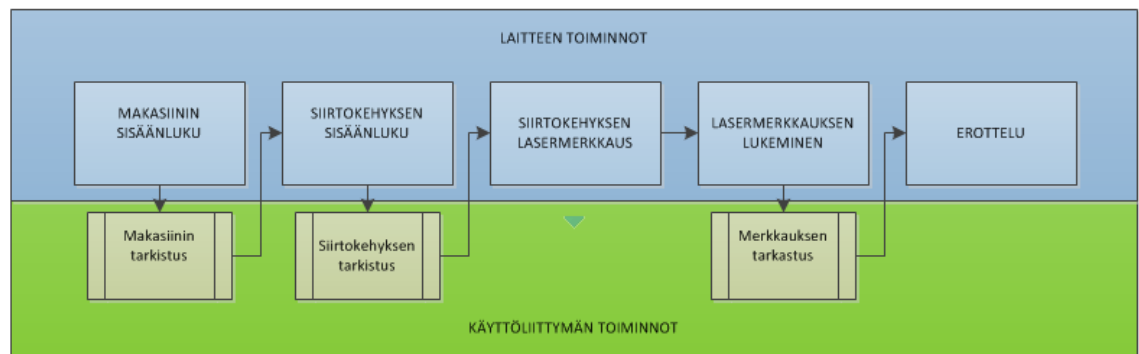
Käyttöliittymän ja laitteen välillä kommunikoidaan TCP/IP-protokollalla. Laitteen toiminta voidaan kuvata viidellä vaiheella:

1. Laite ottaa sisäänsä makasiinin, lukee viivakoodin ja lähettää viivakooditiedon käyttöliittymälle.
2. Laite ottaa makasiinista siirtokehysten ja lukee siitä 2D-matriisikoodin, lähettää sen käyttöliittymälle.
3. Laite merkkää tuotteen.
4. Laite tarkastaa kameralla lasermerkkäusjäljen ja lähettää tuloksen käyttöliittymälle.
5. Laite erottelee tuotteen.

## 4 SUUNNITTELU

### 4.1 Toimintarakenne

Laitteen ja käyttöliittymän välinen toiminta on peruseriaatteeltaan suhteellisen yksinkertainen. Yksinkertaistettuna toimintavuokaavio on seuraavanlainen (Kuva 7):

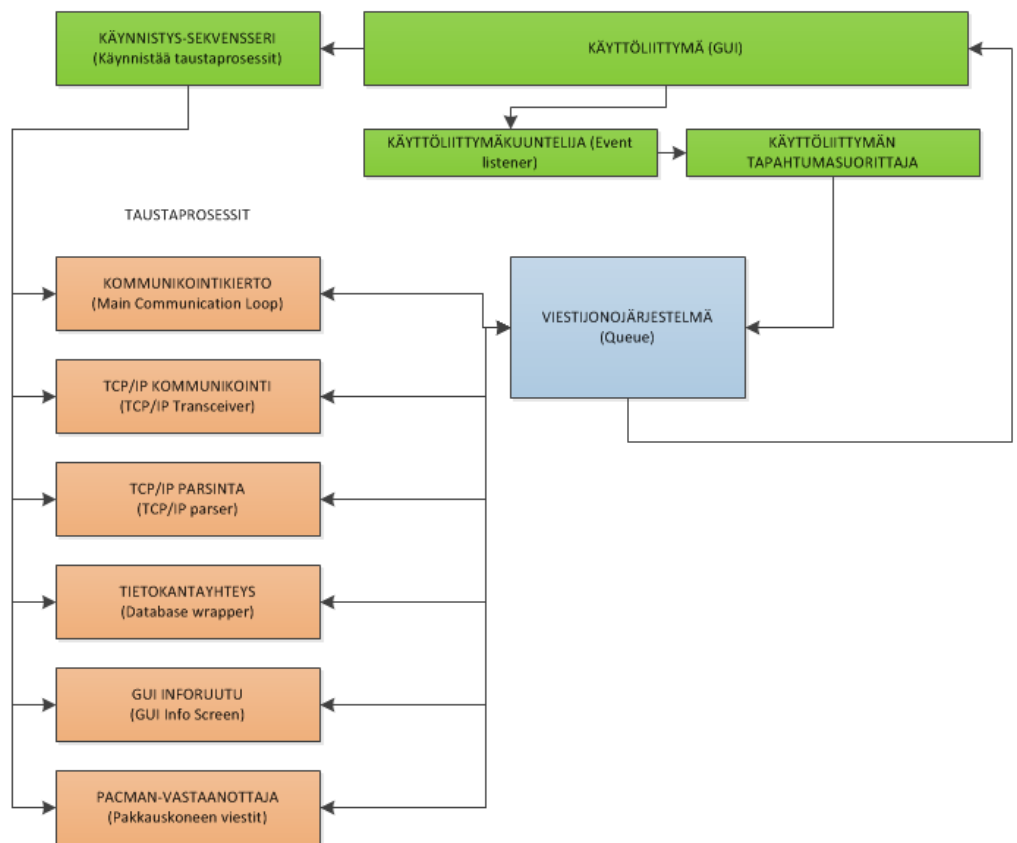


Kuva 7. Toimintarakenne

## 4.2 Ohjelma

### 4.2.1 Ohjelmarakenne

Ohjelma koostuu kolmesta pääosasta (Kuva 8): Käyttöliittymä, viestijonojärjestelmä ja taustaprosessit. Käyttöliittymässä ohjataan laitteen toimintoja painikkeiden eli kontrollien avulla ja saadaan käyttöliittymästä tietoa indikaattoreiden eli näyttöjen kautta. Nämä tiedot päivitetään ja saadaan taustaprosesseilta. Käyttöliittymän ja taustaprosessien välissä on viestijonojärjestelmä.



Kuva 8. Ohjelmarakenne

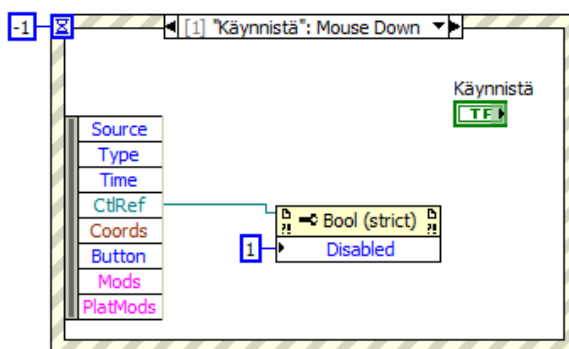
#### 4.2.2 Tapahtumakuuntelija

Käyttöliittymän ohjelma sisältää aina käyttäjän ja ohjelman välistä interaktiota. Interaktio yleensä sisältää käyttöliittymän painikkeiden, hiiren ja näppäimistön välityksellä tapahtuvaa tiedonsyöttöä, joita kutsutaan yleisesti tapahtumiksi [3].

Tapahtumakuuntelija on tämän ohjelmointikielen erikoisuus. Muissa ohjelmointikielissä tehdään erilliset kuuntelijat jokaiselle toiminnolle ja tehdään niiden mukaan tapahtumia. LabVIEW:ssä tehdään ainoastaan yksi kuuntelija, joka hoitaa kaikki tapahtumiin liittyvän kuuntelun automaattisesti ilman lisämäärittysten tarvetta.

Tapahtumakuuntelija kuuntelee käyttöjärjestelmän tapahtumia ja jos tapahtuu jokin halutuista tapahtumista, niin toteutetaan kyseiselle ehdolle määritetyt toiminnot. Kyseiselle toiminnolle ei tarvitse määrittää muuta kuin painike jota kuunnellaan ja sen jälkeen siihen liittyvän tapahtuman ohjelmointi voidaan suorittaa. Tapahtumakuuntelu on tehokas tapa liipaista käyttöliittymän toimintoja ja samalla minimoida resurssien käyttöä [3].

Tapahtumakuuntelun esimerkissä (Kuva 9) käynnistä-painike otetaan pois käytöstä, jos käyttöjärjestelmä ilmoittaa, että hiiren painike on painettu alas käynnistä-painikkeen päällä.



Kuva 9. Painikkeen tapahtumakuuntelu

### 4.2.3 Taustaprosessit

Ohjelman monimuotoisuuden takia päätettiin ohjelman kommunikointi ja käsittely rakentaa taustaprosesseina. Tällä ratkaisulla saadaan minimoitua vasteaikoja TCP/IP-kommunikoinnista logiikalta saataviin vastauksiin sekä pienennettyä käyttöliittymän vasteaikaa.

Taustaprosessien kanssa kommunikointi ratkaistiin viestijonojärjestelmän tuomisella ohjelmaan. Viestijonojärjestelmällä voidaan siirtää dataa taustaprosessien välillä. Viestijono (Queue) nimetään jokaiselle yksilöllisesti, jotta jokainen viesti menee oikeaan paikkaan. Network Stream -viestijonolla voidaan myös lähettää viestejä tarvittaessa koneelta toiselle TCP/IP-verkkoprotokollan yli [4]. Tätä protokollaa käytin kommunikoidakseni pakkauskoneen ja erottelukoneen välillä.

TCP/IP-protokollan kehitti alunperin DARPA (The Defense Advance Research Projects Agency) yhdistääkseen erinäisiä puolustusvoimien tietokoneverkkoja.

TCP/IP (Transmission Control Protocol / Internet Protocol) on tietokoneiden verkkomalli ja määrittely kommunikointiprotokollille, joita käytetään internetissä ja muissa samantyyppisissä verkoissa [5]. TCP/IP-protokolla koostuu eri tasoista (applikaatiotaso, siirtotaso, verkkotaso ja datalinkkitaso) [6]. Käytän tässä projektissa siirtotasoa (TCP), kommunikointiin erottelukoneen logiikan ja käyttöliittymän välillä.

### 4.2.4 Virheenkäsittely

Virheenkäsittely on vaadittua testausjärjestelmissämme, jotta voidaan varmistua ohjelmamme oikeasta toiminnasta tuotteiden käsittelyssä. Virheet pitää voida jäljittää aikaleimoilla tiettyihin ajankohtiin, jossa mahdollisia tuoteongelmia on löytynyt. Virheenkäsittely parantaa käyttäjän kokemusta käyttöliittymästä ja parantaa ohjelman toimintaa. Virheenkäsittelyssä ennakoidaan, havaitaan ja palaututaan erinäisistä ohjelmointi-, kommunikointi- ja applikaativirheistä [7].

Virheenkäsittelyä suunniteltaessa tuli ottaa huomioon virheiden jäljitettävyyden kannalta tärkeitä asioita:

- virhetiedoston koko
- virhetiedoston nimeämiskäytäntö
- virhetiedostojen jatkokäsittely
- virhetiedostojen luettavuus ja käsittely Microsoft Office Excelissä
- virhetiedoston sisältämän ajan yhteneväisyys tietokannasta löytyvän tuotekohtaisen tiedon kanssa.

Tiedostot nimettiin päiväyksellä omaan virhelokihakemistoonsa. Kirjattavia virheitä tulee harvoin, joten tiedostojen koko ei muodostu ongelmaksi. Tiedostomuodoksi valittiin tabuloinnilla jaotellun tekstiformaatin, jonka saa helposti luettua Exceliin soluihin jaoteltuna.

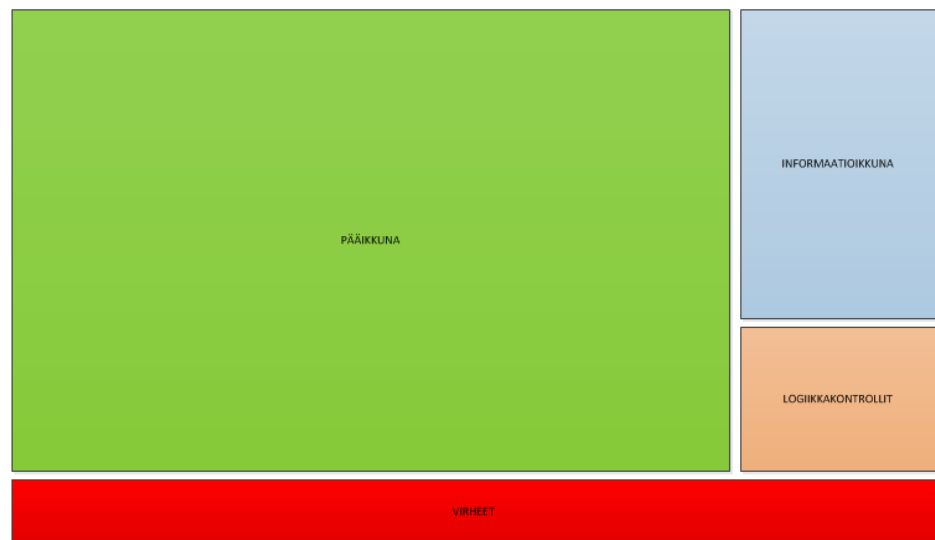
Ajan yhteneväisyys varmistetaan tietokannan kellonaikakyselyllä. Tällä poistetaan riski, jossa tietokoneen kello ja tietokanta eivät ole samassa ajassa ja virhetiedostojen tiedot eivät olisi vertailukelpoisia keskenään toisten laitteiden kanssa.

#### 4.3 Käyttöliittymärakenne

Käyttöliittymä suunnitellaan käytettävyyden ehdoilla. Käyttöliittymän tulee näyttää tilansa, olla yhdenmukainen, minimoida virheilmoitukset ja auttaa käyttäjää tekemään oikeat valinnat [8].

Rakenteet suunniteltiin niin, että käyttäjän katse kiinnittyy aina oikealle puolelle käyttöliittymää ja operaattori saa tarvitsemansa tiedon yhdellä vilkaisulla laitteen tilasta (Kuva 10). Pääikkuna sisältää ohjaavaa tietoa, jolla voidaan määrittää laitteen siirtyminen haluttuun tilaan logiikkakontrollien käyttöä varten. Pääikkuna vaihtuu dynaamisesti riippuen laitteen tilasta ja operaattorin tekemistä määrittämisistä. Kun laite on valmis ajettavaksi, käyttöliittymä indikoi tämän selvästi operaattorille.

Logiikkakontroleilla käynnistetään, pysäytetään ja tyhjennetään laite. Kontrollit vaihtuvat dynaamisesti laitteen tilasta riippuen. Informaatioikkunasta näkee kaiken tarvittavan tiedon nopealla laitteen käyttöä ajatellen (Saanto, tuotetyyppi, versionumero, saantotaulu). Virheet ilmaantuvat käyttöliittymän alaosaan, josta operaattori näkee selvästi uuden ikkunan avautumisen. Virhetekstit ovat muista käyttöliittymän teksteistä poiketen punaisia, jotta virhetilanne käy mahdollisimman selväksi.



Kuva 10. Käyttöliittymärakenne

## 5 TOTEUTUS

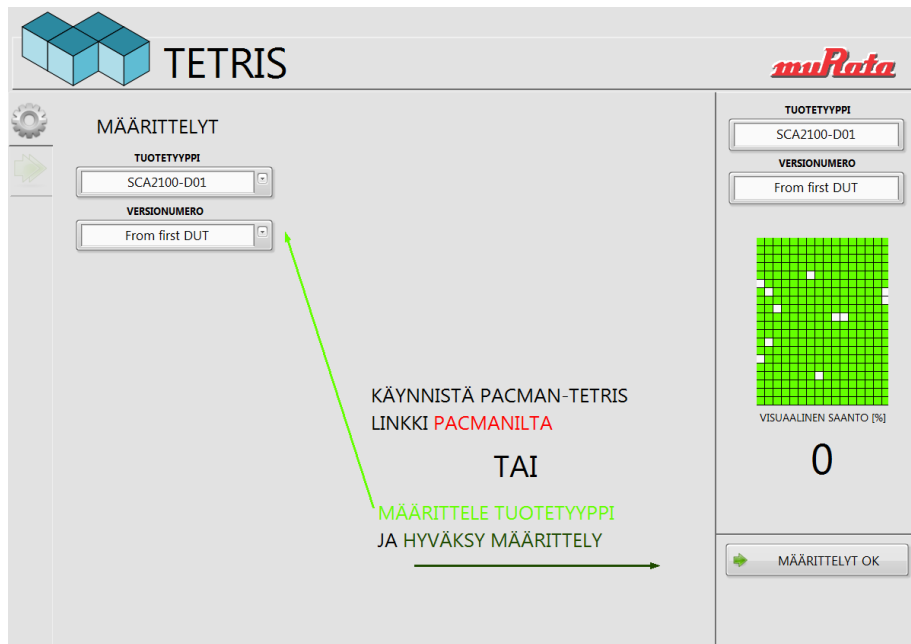
### 5.1 Käyttöliittymä

Käyttöliittymän rakenteesta tuli lähes alkuperäisen suunnitelman mukainen (Kuva 8). Käyttöliittymä ohjaa käyttäjää värein ja tekstein tekemään oikeat valinnat tuotannon aloittamiseksi.

Alhaalla oikealla olevat toiminnalliset painikkeet vaihtuvat dynaamisesti laitteen logiikalta saatujen tietojen mukaan. Virheilmoitukset tulevat erilliseen kelluvaan ikkunaan näkyville käyttöliittymän alaosaan.

#### 5.1.1 Alkumäärittelyt

Alkumäärittelysivun tunnistaa vasemmalla välilehti-valinnoissa olevasta ratas-logosta, sekä välilehden määrittelyt-nimikkeestä (Kuva 11). Ohjelma käynnistyy aina tähän tilaan ja odottaa käyttäjältä määrittäksiä.



Kuva 11. Alkumäärittelyt

Putkiin erottelulla tarkoitetaan tuotteiden merkkausta ja erottelua niille tarkoitettuihin putkipaikkoihin laitteessa. Putkipaikkoja on yhteensä 8 kappaletta ja niille on määritelty erilaisia toimintoja. Testauksessa hyväksytyt ja hylätyt tuotteet jaetaan eri putkiin, jotta hylätyt tuotteet eivät mene pakkausvaiheeseen ja hidasta pakkauskoneen toimintaa.

Alkumäärittelyissä määritetään ajettavan tuotteen tyyppi sekä haluttaessa versionumero. Versionumeron voi myös määrittellä "From First DUT", eli ensimmäisestä laitteeseen tulevasta osasta määriteltäväksi (Kuva 11).

Ketjuajolla tarkoitetaan merkkauk- ja erottelulaitteelta tulevien tuotteiden suoraa siirtoa siirtokelkoilla pakkauskoneeseen. Tämä Master-moodi on laitteiden perustoimintatapa, jolloin päästään tarvittavaan suorituskykyyn.

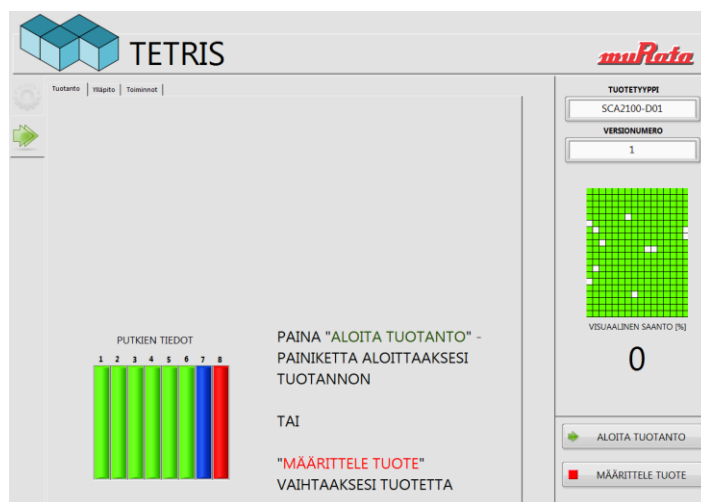
Jos Tetris on yhteydessä pakkauskoneeseen, aloitussivun yläriivin teksti "Tetris" (Kuva 11) muuttuu tekstiksi "Tetris + Pacman" indikoimaan, että pakkauskone on ottanut Tetriksen komentoonsa (eli Master-moodi).

Pakkauskone Pacman on suoraan kiinni laitteessa siirtokelkkojen välityksellä. Tästä pystytään siirtämään merkatut ja erotellut tuotteet suoraan pakkauskoneelle ilman putkitusvälivaihetta. Tällöin operaattori ei pysty määrittelemään Tetriksestä tuotetyyppejä, vaan määrittely tulee hoitaa pakkauskoneella. Määrittely tehdään samalla tavalla kuin Tetriksessä.

### 5.1.2 Tuotantosivu

Tuotantosivulle (Kuva 12) siirrytään, jos laitteelle on määritetty tuote ja määrittelyt hyväksytyt tai jos pakkauskone on Master-moodissa, eli määrää merkkaukoneen toiminnoista. Tuotantosivu ohjaa operaattoria tekstillä ja värillä tekemään oikeat valinnat. Sivulta voi aloittaa tuotannon tai määrittellä tuotteen uudelleen. Jos laite ei ole oikeassa tilassa käynnistystä varten (esimerkiksi laite ei vastaa kommunikointipyyntöihin) on tuotannon aloituspainike harmaana.

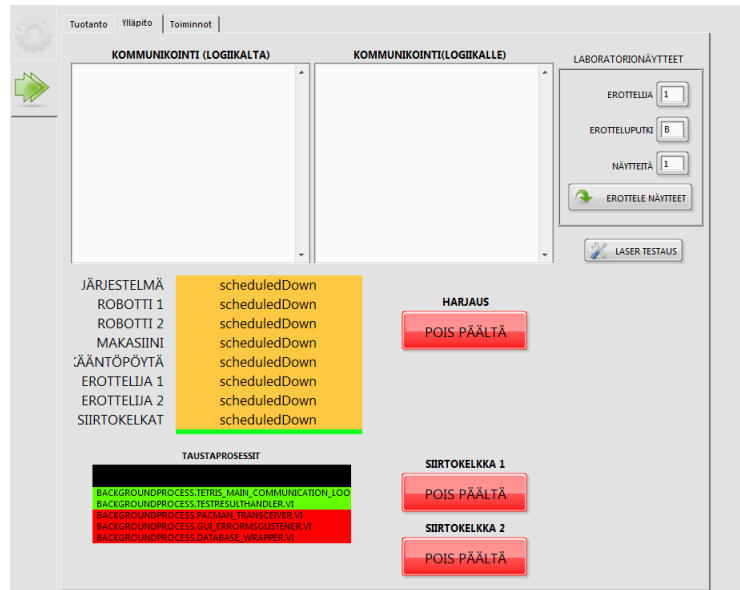
Erotteluputkien tiedot ja värit näkyvät selvästi oikealla alakulmassa. Nämä värit ovat verrannollisia saantoikkunassa oikealla näytettäviin väreihin testien tuloksista.



Kuva 12. Tuotantosivu

### 5.1.3 Huoltosivu

Huoltosivulla (Kuva 13) huoltomiehet ja tuotannon tukihenkilöt pystyvät tekemään toimintoja, joita ei ole tarkoitettu operaattoreille. Tällä sivulla voidaan tarkastella laitteen tila, käynnissä olevat taustaprosessit ja niiden tila, erinäisten laitteiden käyttöönotto ja -poisto, testata laserin toimivuus sekä ottaa laboratorionäytteet. Myös laitteen logiikan ja käyttöliittymän välinen kommunikointi on näkyvässä tällä sivulla.



Kuva 13. Huoltosivu

#### 5.1.4 Saantoikkuna

Saantoikkuna on käyttöliittymässä (Kuva 12) oikealla oleva 2D-taulukko. Taulukko indikoi testattavien siirtokehysten visuaalisen testauksen ja tietokantatarkistuksen yhteistuloksen: Punainen on hylky ja sininen on väärä tuotetyyppi. Vihreät ovat hyviä tuotteita, joilla merkkkaus on onnistunut ja kaikki tuotteen testitulokset ovat sallituissa rajoissa. Musta ja valkoinen ovat aikaisemmissa testeissä hylättyjä tuotteita

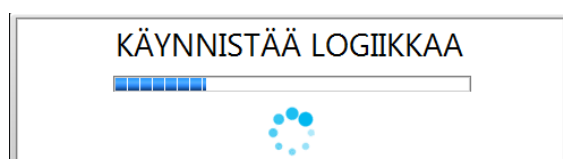
Jokaisesta siirtokehyksellä olevasta tuotteesta saa lisätietoa viemällä hiiri taulukossa olevan neliön päälle. Lisätiedot sisältävät mm. siirtokehysten numeron, tuotteen sarjanumeron ja tietokantatestien tuloksen.

Saantoikkunasta operaattori pystyy nopeasti näkemään laitteessa näkyvät ongelmat. Jos saantoikkunassa on punaisia tai sinisiä neliöitä, tulee operaattorin kiinnittää asiaan huomiota. Visuaalinen saanto ilmoitetaan myös taulukon alapuolella prosentteina. Ohjelma ilmoittaa operaattorille, jos saanto menee alle asetetun minimirajan (95 %).

### 5.1.5 Tapahtumaindikointi

Käyttöliittymän tapahtuma käynnistyy, kun operaattori painaa jotain painiketta. Jos painikkeen toimintoon liittyy laitteeseen kohdistettuja toimintoja, jotka kestävät kauemmin kuin yhden sekunnin, näytetään operaattorille tapahtumasta indikointi (Kuva 14). Indikoinnissa kuvataan tekstillä mitä ollaan tekemässä ja palkilla mahdollisen monivaiheisen tapahtuman edistyminen.

Tapahtumia joissa erillistä indikointia tarvitaan ovat muun muassa tuotannon lopetus, tuotannon aloitus, koneen tyhjennys ja lasermittausten suorittaminen.



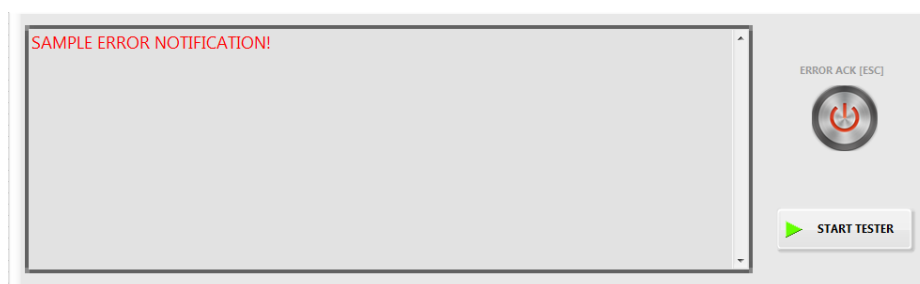
Kuva 14. Laitteen tapahtuma käynnissä

### 5.1.6 Virheilmoitukset

Laitevirheen sattuessa käyttäjälle ilmoitetaan asiasta isolla huomiota herättävällä virheikkunalla (Kuva 15). Virheikkuna tulee käyttöliittymän alalaidan alle kelluvana ilmoituksena päällimmäiseksi. Jos virhe vaatii operaattorilta muita toimintoja kuin pelkästään virheen kuittauksen, ilmoittaa virheikkuna myös siitä.

Virhe kuitataan "ERROR ACK"-painikkeesta ja tuotanto käynnistetään uudelleen "Start Tester"-painikkeella.

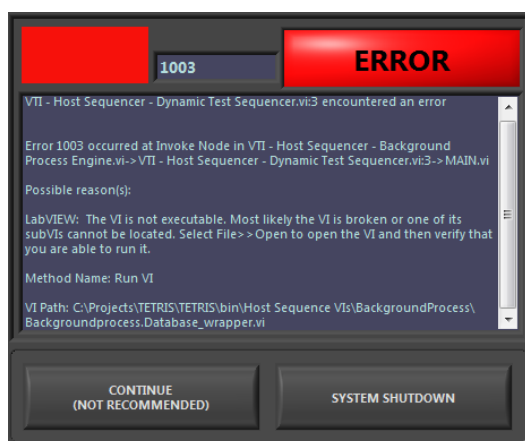
Virheet tallentuvat `C:\temp\YYYYMMDD_Tester_Err_Log.txt` tiedostoon, josta voidaan myöhemmin tarkastella mahdollisia ongelmia.



Kuva 15. Logiikan virheikkuna käyttöliittymässä

Käyttöliittymän virheet ovat aina kriittisiä ja tuotannon pysäyttäviä. Käyttäjälle indikoidaan erittäin selvästi, että on tapahtunut jokin hallitsematon virhe josta ei voida toipua (Kuva 16). "Continue"-painiketta painettaessa tuotanto pysähtyy ja käyttöliittymä siirtyy aloitustilaansa määritys-sivulle. "System shutdown"-painike sammuttaa koko ohjelman.

Virheet tallentuvat C:\lverrorlog\ -hakemistoon aikaleimalla ja virhekoodilla varustettuna, josta niitä voidaan myöhemmin käydä tarkastelemissa.



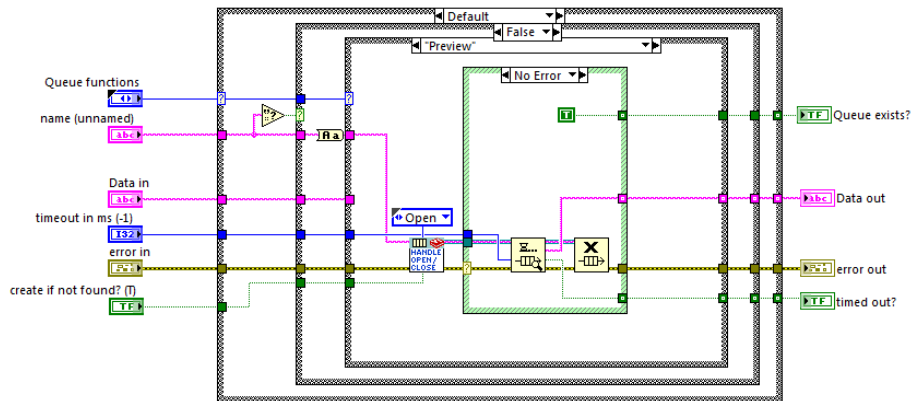
Kuva 16. Kriittinen ohjelmavirhe

## 5.2 Taustaprosessit

Taustaprosessit hoitavat poikkeuksetta kaikki tiedonsiirtoon liittyvät asiat. Taustaprosessit kommunikoivat TCP/IP-protokollan avulla käyttöliittymän, Tetris-logiikan ja Pacman-logiikan välillä. Taustaprosessit tallentavat kaikki tulokset, ylläpitävät testimuistia, tarkastavat laserien kunnan automaattisesti sekä näyttää mahdolliset virheet logiikalta.

### 5.2.1 Viestijärjestelmä

Taustaprosessit kommunikoivat keskenään jonojärjestelmällä (Kuva 17). Viestijärjestelmässä on jonon nimi (Vastaanottaja), datasisältö ja aikaviive. Viestin datasisältö riippuu vastaanottajasta ja tiedon muoto tulee olla tiedossa molemmilla kommunikoivilla taustaprosesseilla.

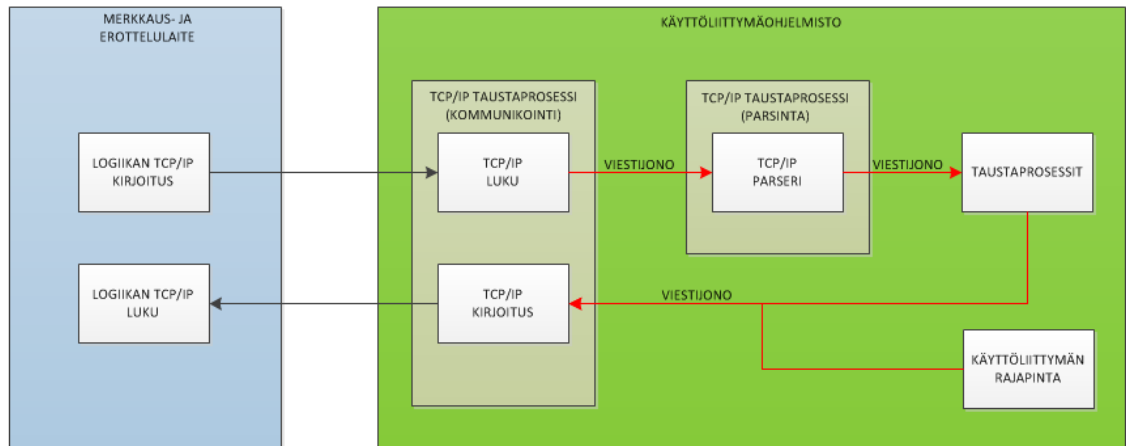


Kuva 17. Jonoon luku/kirjoitus

Jokaisella taustaprosessilla on vähintään yksi jono datan lukuun. Näin ne eivät joudu odottamaan muiden taustaprosessien komentojen suorittamista. Jono alustetaan pääohjelman käynnistyessä ja sen jälkeen tyhjentämisestä vastaavat taustaprosessit itse. Jos jono täyttyy, antaa LabVIEW'n oma muistihallintajärjestelmä virheilmoituksen.

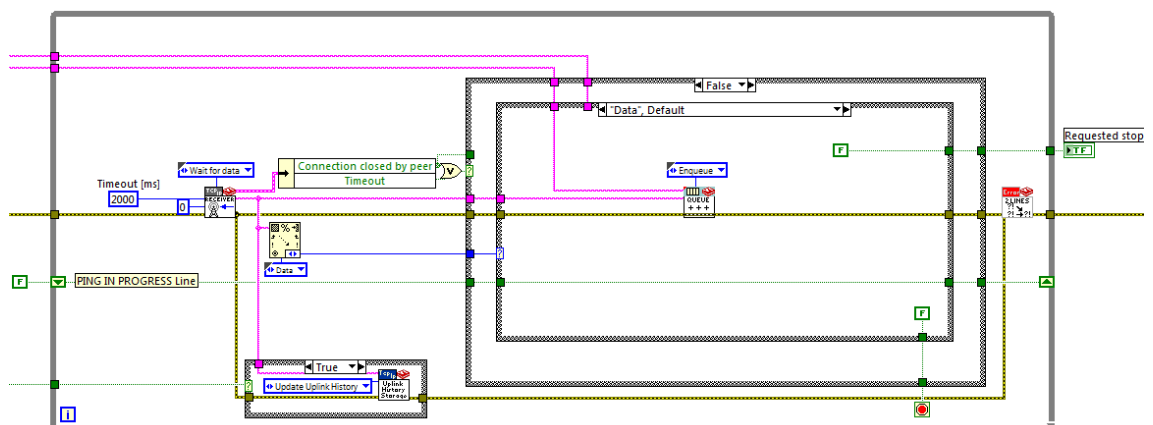
## 5.2.2 TCP/IP-taustaprosessi

Kommunikointi laitteen ja käyttöliittymän välillä hoidetaan TCP/IP-protokollan välityksellä. Kommunikointia hoitaa kaksi erillistä taustaprosessia, jotka ovat jaoteltu käyttötarkoituksen mukaan. Ensimmäinen vaihe on kommunikoinnin vastaanotto ja lähetys, toinen on vastaanotetun datan parsinta (Kuva 18).



Kuva 18. TCP/IP-kommunikointi

TCP/IP-kommunikoinnin taustaprosessi vastaanottaa komennot ja lähettää ne edelleen jonoon parsintaan (Kuva 19). Taustaprosessi lähettää edelleen viestijonosta tulevat komennot merkkauk- ja erottelulaitteelle.



Kuva 19. Vastaanotetun TCP/IP-datan lähetys viestijonoon

Parsinnassa jono puretaan viesti kerrallaan, jossa on ennalta määritetty sallitut komennot. Jos parsintaan tulee tuntematon komento, ilmoitetaan tästä virheilmoituksella. Jokaisen vastaanotetun TCP/IP-viestin lopetusmerkki on # ja komennon jälkeiset lisäparametrit on eroteltu pilkulla.

Esimerkki:

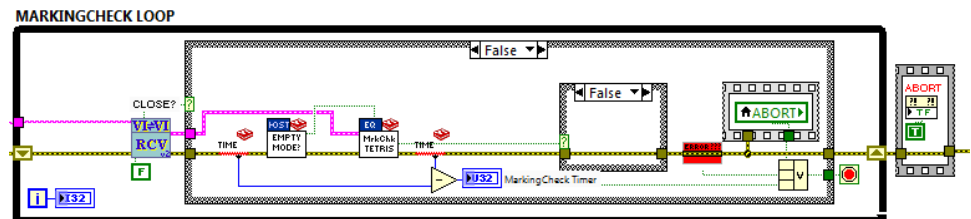
**LeadFrameID,1241512 #**

Komento kertoo, että koneeseen on otettu siirtokehys jonka koodi on 1241512.

Parsinta lähettää viestit edelleen seuraaviin jonoihin, jotka ovat kohdistettu taustaprosesseihin. Taustaprosessit kuuntelevat omia jonojaan ja tekevät toimintonsa viestien pohjalta ja lähettävät vastauksensa viestijonoon (Kuva 18).

Pääkommunikointi-taustaprosessi sisältää monta viestijonojen kuuntelijakiertoa, joista jokaiselle on määritelty oma viestijononsa. Jos yksikin viestijonoista suorittaa virheen jota ei voida käsitellä, suljetaan koko taustaprosessi. Tämä taustaprosessi hoitaa kaikki tuotteiden tarkastuksiin liittyvien tietokantakyselyiden eteenpäinviemisen ja logiikan vastausten lopullisen parsinnan, sekä päättää mitä tuotteille tehdään.

Taustaprosessi on jaettu osiin (Kuva 20), joiden suorituskyvyllä on merkitystä. Tietyille vastaanotetuille komennoille on tämän takia oma parsintakiertonsa, jottei syntyisi ylimääräisiä viiveitä laitteen toiminnassa.



Kuva 20. Merkkauksen tarkistus- ja kuuntelukierto

### 5.2.3 Makasiinin tarkistus

Logiikka lähettää makasiininumeron. Makasiinin numero tarkistetaan tietokannasta, kysellään pakkauskoneelta (Jos Master-moodi on käynnissä) ja katsotaan että kyseinen makasiini voidaan ottaa sisään koneeseen.

Jos makasiini läpäisee tarkastukset ja pakkauskone tarvitsee lisää pakattavaa, lähetetään koneelle "Makasiininumero, OK#", jos makasiinia ei voida hyväksyä laitteeseen, lähetetään "Makasiininumero, NG#". Jos pakkauskone ei tarvitse tällä hetkellä lisää pakattavaa, lähetetään vastaukseksi "Makasiininumero, WAIT#".

### 5.2.4 Siirtokehysten tarkistus

Siirtokehysten tarkistuksessa laite lukee siirtokehystä 2D-matriisin ja lähettää tiedon käyttöliittymälle. Käyttöliittymässä tarkastetaan siirtokehysten tiedot tietokannasta. Tietokannan tärkeitä tietoja ovat muun muassa oikea vaihe, testauksen tulos ja tuotetyyppi. Jos tuotteen ei kuuluisi vielä olla merkkauksen- ja erotteluvaiheessa, hylkää ohjelma tuotteen ja siirtokehys palautetaan takaisin makasiiniin.

Jos makasiinilta on palautettu neljä perättäistä siirtokehystä takaisin, lähetetään makasiini ulos koneesta. Jos luettu siirtokehys on hyvä, luetaan tietokannasta haetulle tuotetyypille merkkauksetiedoston nimi ja merkkauksetiedot (2D-matriisi, nuolet, logotyyppi, sarjanumeron näkyminen jne.) ja lähetetään ne vastausviestinä laitteelle.

### 5.2.5 Merkkauksen tarkistus ja erottelu

Logiikka lähettää datajoukon jokaiselle siirtokehyksessä olevalle tuotteelle (esimerkki yhden tuotteen viestistä):

```
"LEAD-
FRAMEID,Framecode=8633259,OK,Filename=SCA8x0,Mark
_1=YES,                                     Mu-
rataLogo_Mark_1=NO,VTILogo_Mark_1=YES,ProductNr_M
ark_1=YES,                                   ProductNr_Data_1=SCA2120-
D12,SerialNr_Mark_1=YES,SerialNr_Data_1=
0351512625M3F,Matrix2D_Mark_1=NO,Matrix2D_Data_1=
0351512625M3F,ArrowZ_Mark_1=YES,ArrowY_Mark_1=YES
,ArrowX_Mark_1=NO,FirstPinMark_1=NO"
```

Tarkastusvastausta verrataan siirtokehysten tarkastuksessa lähetettyyn merkaustiedoston sisältöön ja tarkistetaan, että kaikki oikeat logot, sarjanumero, tuotetyyppi ja muut merkinnät ovat täysin identtisiä.

Jos tuotteella ei ole 2D-matriisia, tehdään tuotteen sarjanumero luvulle ryhmätarkastus, jolla voidaan olla varmoja tuotteen sarjanumeron merkkauksesta. Tässä tarkastuksessa on vaatimuksena, että siirtokehyksellä on vähintään 5 tuotetta, joiden kannesta luettu numeraalisessa muodossa oleva sarjanumero täsmää tietokannasta haettuun. Jos näin ei ole, erotellaan siirtokehys hylkyputkeen.

Jos tuotteiden merkkaukset on hyväksyttävä, lähetetään tuotteet eteenpäin hyväksyttävien tuotteiden putkiin / pakkauskoneelle.

## 5.2.6 Laserin tehontarkastaja

Laite käyttää kahta Datalogic-laseria tuotteiden merkkaukseen. Kommunikointi lasereille hoidetaan suoraan logiikan kautta ja käyttöliittymäohjelmisto lähettää ainoastaan merkkaustiedoston ja tarvittavat parametrit logiikan edelleen lähetettäväksi.

Laserin tehontarkastaja pitää huolen, että merkkauksen laatu ei muutu tuotteiden välillä. Laserien tehoa tarkastellaan 6 tunnin välein 35 %:n, 65 %:n ja 100 %:n kuormilla ja tehomäärät (W) kirjataan ylös tiedostoon `c:\testresults\laser1.dat` ja `laser2.dat` tiedostoihin. Jos kummankaan laserin voima muuttuu yli 10 %, ilmoittaa taustaprosessi siitä käyttäjälle.

Virheilmoituksen jälkeen taustaprosessi avaa graafin, josta näkee miten laserien voima on kehittynyt ajan saatossa 35 %, 65 % ja 100 % kuormilla (Kuva 21).



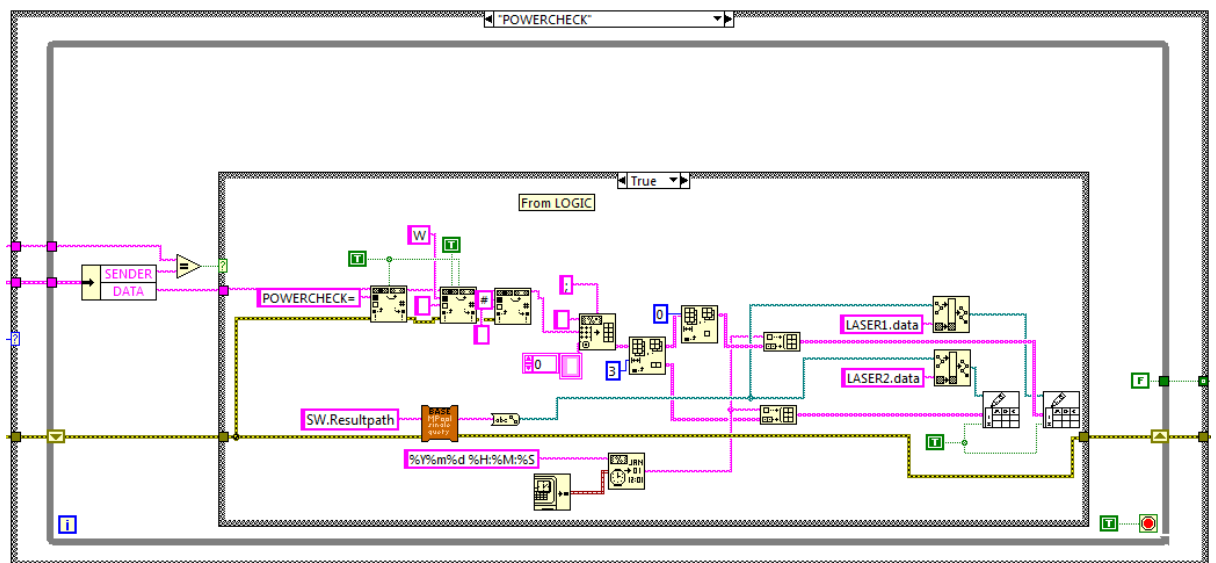
Kuva 21. Laserin tehonmittauskäyrät

### 5.2.7 Muut komennot

Komennot joilla ei ole omaa kuuntelukiertoa kommunikoinnin taustaprosessissa, ohjautuvat tähän kuuntelijakiertoon.

Tässä kierrossa parsitaan muun muassa:

- laitteen virheilmoitukset
- laserin tehomittauksen vastaukset (Kuva 22)
- laitteen sisällä olevien tuotteiden määrä
- laitteen virtojen tilan



Kuva 22. Laserin tehontarkastustulosten tallennus

## 6 YHTEENVETO

Tässä opinnäytetyössä suunniteltiin käyttöliittymä sekä kommunikointi logiikan ja pakkauskoneen välille. Ohjelmisto myös hoitaa merkkkaus-laserin tehojen valvonnan ja ilmoittaa mahdollisista logiikan ongelmista käyttäjälle.

Lähtiessäni projektiin minulla oli hyvä mielikuva siitä, minkälaisen käyttöliittymän ja operaattorinäkyvän halusin luoda. Käydessäni operaattoreiden kanssa keskustelua heidän haluamistaan toiminnoista, huomasin että operaattorit olivat ehkä tottuneet muissa käyttöliittymissä vanhanaikaisiin ratkaisuihin käytettävyyden osalta.

Käyttöliittymäsuunnittelua tarkastellessani, opin hyvien käyttöliittymien erilaisia suunnittelutapoja ja lähestymistyyliä. Omaan suunnittelutyyliin sopii yhdistelmä toiminnallista ja käyttäjäläheistä suunnittelua, jossa loogisuudella ja sijoittelun jatkuvuudella pidetään käyttäjän katse käyttöliittymän oikeassa osassa, jolloin virhepainalluksien ja ylimääräisten kysymysten esittäminen käyttöliittymän toiminnasta ja pitkien ohjeiden tarve minimoituu.

Palautteet operaattoreilta ovat olleet erittäin positiivisia. Tuotantoympäristössä pienikin toiminnan tehostus vaikuttaa yrityksen tulokseen positiivisesti ja kilpaillulla elektroniikan alalla kaikki säästö on hyväksi. Operaattoria ohjaavat tekstit ja nuolet, joilla vähennetään operaattorien koulutustarvetta ja arkuutta uuden laitteen käytössä lisäävät operaattorien tehokkuutta. Näin saadaan pienennettyä myös kouluttajan työkuormaa, kun ohjeet itsessään ovat yksinkertaiset ja käyttöliittymä osaa ohjata operaattorin toimintaa.

Myös Microsoft Vision käyttö on helpottanut loogisten rakenteiden ymmärtämistä ja nopeuttanut oman koodin tekemistä ja testausta. Dokumentointi tulee tällöin suoraan määrittelyvaiheessa ja helpottaa koodin korjausta, päivitystä ja uudelleenkäyttöä. Toinen ohjelmaan perehtymätön henkilö voi helposti nähdä loogisista kaavoista toimintaperiaatteet ja ohjelmien yhteydet toisiinsa, sekä löytää haluamansa ohjelmaosion jota haluaa päivittää.

Alkuperäisestä aikataulusta myöhästettiin lähes vuodella laitetoimittajan toimitusongelmien takia. Myös pitkä välimatka laitetoimittajan tiloihin vähensi testausmahdollisuuksia ennen laitteen toimitusta yrityksen tuotantotiloihin.

Kävin kahdesti laitetoimittajan valmistustiloissa vuonna 2014 testaamassa laitetta, joista ensimmäinen kerta oli lähinnä laitteen logiikan toiminnan testausta ja kommunikointien määrittystä laitetoimittajan ohjelmistovastaavan kanssa.

Määrittäsvaiheessa myös kommunikointiin jäi vielä puutteita. Laitteen saapuessa tuotantotiloihin piti laitteen komentolistoihin lisätä vielä yli 10 toimintoa komentojen kautta ja päivittää laitteen logiikkaohjelmisto toimintaongelmien takia useaan kertaan. Tulevaisuuden projekteihin aion kysyä enemmän laitteita aikaisemmin määriteltä vanhemmilta insinööreiltä neuvoja ja mahdollisia kompastuskiviä logiikkakaavioissa, komentotarpeissa sekä hyväksyntätestauksissa.

Tuotantoon käyttöönotossa virheilmoitusten vähyys aiheutti tuotantoviiveitä operaattoreiden toiminnassa ja huoltomiehien vianselvityksessä. Laite ei ilmoittanut, minkä takia tietty siirtokehys tai makasiini hylättiin. Toiminnan läpinäkyvyyden parantamiseksi lisättiin laitteeseen lisää tietoa siirtokehysien ja makasiinien tiloista, jotta operaattorit ja huoltomiehet voivat toimia paremmin tilanteissa, joissa ongelmat ilmenee.

## LÄHTEET

- [1] Murata Electronics Oy, "*Murata Electronics Oy*", [www-dokumentti], saatavilla: <http://www.muramems.fi/fi/murata/murata-electronics-oy>, (Luettu: 15.2.2015)
- [2] National Instruments, "*What Is LabVIEW?*", [www-dokumentti], saatavilla: <http://www.ni.com/newsletter/51141/en/>, (Luettu: 15.2.2015)
- [3] National Instruments, "*A Powerful New Tool for UI Programming*", [www-dokumentti], saatavilla: <http://www.ni.com/white-paper/2962/en/#toc2>, (Luettu: 15.2.2015)
- [4] ViewPoint Systems Inc., "*Communicating between parallel loops*". [www-dokumentti], saatavilla: <http://www.viewpointusa.com/resource/view/newsletters/communicating-between-parallel-loops/>, (Luettu: 15.2.2015)
- [5] Protocols.com, "*TCP / IP Protocols*", [www-dokumentti], saatavilla: <http://www.protocols.com/pbook/tcpip1.htm>, (Luettu: 3.3.2015)
- [6] International Business Machines Inc., "*IBM. Redbooks*", [pdf-dokumentti], saatavilla: <http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>, (Luettu: 3.3.2015)
- [7] National Instruments, "*LabVIEW Error Handling Best Practices*", [www-kuunnelma], saatavilla: <http://www.ni.com/webcast/2711/en/>, (Kuunneltu: 1.2.2015)
- [8]. Jacob Nielsen, "*10 Heuristics for User Interface Design*", [www-dokumentti], saatavilla: <http://www.nngroup.com/articles/ten-usability-heuristics/>, (Luettu: 3.3.2015)