

SATAKUNNAN AMMATTIKORKEAKOULU



Lauri Holm

2008

TUULILASIN PUSSIMITTAUSOHJELMISTO

Tekniikka Rauma

Tietotekniikan koulutusohjelma

## TUULILASIN PUSSIMITTAUSOHJELMISTO

Holm, Lauri

Satakunnan ammattikorkeakoulu

Tekniikka Rauma

Tietotekniikan koulutusohjelma

Yritys: Pilkington Automotive Finland Oy

Työn valvoja: Petri Meriläinen

Maaliskuu 2008

Ohjaaja: Yliopettaja Yrjö Auramo

UDK-luokka: 681.2

Asiasanat: lasiteollisuus, mittauslaitteet, ohjelmointi, Visual Basic.NET

Tämä opinnäytetyö tehtiin Pilkington Automotive Finland Oy:lle. Työn tarkoitus oli kehittää paranneltu ratkaisu jo osittain olemassa olevasta järjestelmästä, joka mittaa uunissa taivutetun lasin pinnasta ns. pussiarvon. Pussiarvo tarkoittaa lasin matalimman ja kahden korkeimman kohdan erotusta lasin poikkileikkauksessa.

Toteutettu järjestelmä koostuu laseranturista, mittausyksiköstä ja kannettavasta tietokoneesta. Opinnäytetyön keskeisin ja pitkäkestoisin osa oli mittausohjelmiston ohjelmoiminen Visual Studio 2005 Express edition -työkalulla. Toteutettu mittausohjelmisto kerää arvoja anturilta havaitessaan lasin sen alapuolella, laskee pussi- eli syvyysarvon ja tallentaa tiedot Microsoft Access -tietokantaan. Lisäksi ohjelman ominaisuuksiin kuuluvat kerättyjen mittapisteiden kuvaajan piirto sekä vanhojen mittausten tarkastelu tietokantaa lukemalla.

Järjestelmästä pyrittiin tekemään helposti siirrettävä ja kevyt, jotta sitä voitaisiin soveltaa mahdollisimman monella eri uunilla. Lisäksi järjestelmän komponentit valittiin niin että järjestelmä olisi mahdollista laajentaa myöhemmin.

Ohjelmiston keskeiset osat ovat lasin mittapisteiden keräys, tulosten muunnos millimetreiksi, pussiarvon laskeminen sekä tulosten tallennus tietokantaan. Nämä toiminnot ovat automaattisesti käytössä ohjelmiston ollessa mittaustilassa, jolloin mittaustulokset ovat nähtävissä suoraan monitorilta.

Ohjelmisto osaa etsiä mittauspisteiden joukosta anturin häiriöistä johtuvat virheelliset pisteet ja suodattaa ne pois tulosta laskettaessa sekä kuvaajaa piirrettäessä. Loppullinen mittaustulos selvitetään etsimällä taulukosta piste, jonka etäisyys taulukon alku- ja loppupisteiden lävistämästä suorasta on suurin.

Opinnäytetyön kirjallinen osa keskittyy mittausjärjestelmän toiminnan kuvailemiseen ja saatujen tulosten tarkastelemiseen. Mittausjärjestelmän eri toiminnot ja mittausten vaiheet kuvailaan tehdyn ohjelmointityön pohjalta. Lisäksi perustellaan ohjelmointiratkaisuja ja syitä, miksi tiettyihin ratkaisuihin päädyttiin.

## MEASURING SOFTWARE OF WINDSHIELD DEPTH

Holm, Lauri

Satakunta University of Applied Sciences

School of Technology Rauma

Information Technology

Commissioned by Pilkington Automotive Finland Oy

Supervisor: Petri Meriläinen, Maintenance Team Leader

March 2008

Tutor: Yrjö Auramo, Principal Lecturer

UDC: 681.2

Keywords: glass industry, measuring equipment, programming, Visual Basic.NET

This thesis was made for Pilkington Automotive Finland Oy. The purpose of this thesis was to develop an enhanced version of the existing system. It is used to measure the so called bagging value of a windshield bended in a furnace. The bagging value means a value calculated from the difference of the lowest and two highest points of the windshield.

The implemented system consists of a laser sensor, a measuring unit and a computer. The most central and time-consuming part of the thesis was the programming of the measuring software with Visual Studio 2005 Express edition. Additional features of the software include drawing a graph from the measured data and an analysis of the past measurements by reading the database.

The central parts of the software are collection of the measurement points, millimeter conversion of the results, calculating the bagging value and saving the results in the database. These functions are automatically in use when the software is in the measurement state, when the results of the measurement can be seen directly from the screen.

The software is able to search false values from the measurement data and filter them off when calculating the result and drawing a graph. The false values can occur when the sensor has interference. The literary part of the thesis focuses on describing the functionality of the measuring system and analyzing the received results.

## SISÄLLYS

### TIIVISTELMÄ

### ABSTRACT

1 JOHDANTO.....	5
2 TARKOITUS JA TEHTÄVÄ.....	6
3 TAUSTATIEDOT .....	6
3.1 Pilkington Automotive Finland Oy.....	6
3.2 Lasin mittaaminen.....	7
4 KÄYTETYT TYÖKALUT JA MENETELMÄT .....	8
4.1 Anturi .....	8
4.2 Mittausyksikkö.....	9
4.3 Mittausohjelmisto.....	9
4.3.1 Mittapisteiden kerääminen anturilta ohjelmaan.....	13
4.3.2 Mittauksen vaiheet yksityiskohtaisesti.....	13
4.3.3 Liittymä tietokantaan.....	17
4.3.4 Kuvaajan piirto.....	17
4.3.5 Tuotenimen haku.....	19
4.3.6 Asetukset-sivu.....	19
4.3.7 Tuote-selain .....	22
5 TULOKSET .....	23
5.1 Projektin kulku.....	24
5.1.1 Hankinnat.....	24
5.1.2 Suunnittelu .....	24
5.1.3 Ohjelmiston parantelu ja testaus .....	25
5.2 Työn tekijän havainnot.....	25
5.2.1 Tuloksia .....	26
5.3 Parannusehdotukset.....	26
LÄHTEET.....	28

# 1 JOHDANTO

Tämä opinnäytetyö käsittelee taivutetun lasin pussimittausjärjestelmää, joka kehitettiin projektina Pilkington Automotive Finland Oy:lle. Projektin toteutus koostui mittausanturista, mittausyksiköstä sekä niitä hyödyntävästä ohjelmistosta Windows XP -ympäristössä. Työn raportointi painottuu ohjelmiston toiminnan selostamiseen ja kuvailemiseen, koska ohjelmisto ja sen tekeminen oli keskeisessä asemassa työn suorituksessa.

Työn tilaajalla oli tarvetta tällaiselle projektille, koska käytössä oleva vastaavanlainen järjestelmä oli ominaisuuksiltaan rajoittunut sekä vaikeasti siirrettävä. Sillä ei myöskään voinut erinäisistä syistä mitata aivan kaikkia tuotteita, joten osa tuotteista mitattiin käsimitalla. Mitatut lasit kirjattiin käsin paperille ja arkistoitiin. Tässä työssä toteutettu ohjelmisto on siis paranneltu versio jo olevassa olevasta järjestelmästä. Uusia ominaisuuksia ovat mm. mittaus tiedon tallentuminen Microsoft Access -tietokantaan, graafinen tulostus mittauspisteistä ja tuotteen historiasta sekä mittausjärjestelmän helppo siirrettävyys.

Työn toteutus suoritettiin ohjelmoimalla mittausohjelmisto tilaajalle Microsoft Visual studio 2005 Express edition -työkalulla. Tämä versio Visual studiosta on ilmainen, ja koodatun ohjelman kaikki oikeudet jäävät ohjelmoijalle. Työn tilaaja toimitti tietokoneen, mittausanturin ja mittausyksikön, johon sisältyi mittausmoduuli, sekä sarjaportti/USB -adapterin.

## 2 TARKOITUS JA TEHTÄVÄ

Tämän projektin tarkoitus on parantaa lasin mittauksen tehokkuutta ja samalla laadunvalvontaa lasinvalmistuksen aikana. Raakalasilvyt käyvät läpi pitkän matkan valmiiksi tuulilasiksi, mutta tämä mittaus keskittyy taivutusvaiheen laadunvalvontaan. Tarkoituksena on myös helpottaa lasiuunilla työskentelevien henkilöiden työtä, kun mittaustuloksia ei enää tarvitse kirjata muistiin paperille ja arkistoida, vaan tulokset tallentuvat tietokantaan automaattisesti mittauksen jälkeen.

Pussimittaus on kriittinen mittaus, jonka arvon perusteella työntekijät joko asettavat tuotteen jigille eli sovitussuoralle tai suoraan lasinkuljetuspukkiin, jolla lasit siirretään prosessin seuraavaan vaiheeseen. Jigillä tuotteen koko ja syvyys mitataan tarkemmin.

Työn tekijän tehtäväksi jäi tarkoitukseen sopivien mittauslaitteiden valinta ja hankinta sekä ohjelmiston ohjelmoiminen. Ohjelmiston tuli olla luotettava, helposti siirrettävä ja mahdollisesti laajennettava. Mittauslaitteena kokeiltiin ensin ultraäänianturia, joka kuitenkin osoittautui liian häiriöalttiiksi mm. uunista tulevan kuumen ilman väräilylle. Toimiessaan ultraäänianturi olisi ollut laseranturia huokeampi vaihtoehto. Lopulta päädyttiin käyttämään laseranturia, johon lämpöväreily ei vaikuta.

## 3 TAUSTATIEDOT

### 3.1 Pilkington Automotive Finland Oy

Pilkington on perustettu vuonna 1826 Iso-Britanniassa. Vuonna 2006 Pilkington siirtyi japanilaista alkuperää olevan NSG Groupin omistukseen ja on siitä asti ollut osa yhdestä suurimmista rakennus- ja autoteollisuuden lasinvalmistajista maailmassa. La-

sinvalmistusta harjoitetaan yhteensä 26 eri maassa, omistuksessa tai osittaisomistuksessa on 46 lasitehdasta ja työntekijöitä on yhteensä noin 36 000.

Pilkington Automotive Finland Oy:llä on kolme ajoneuvonlasitehdasta: Ylöjärven tehdas valmistaa henkilö- ja kuorma-autojen tuulilaseja, Tampereen tehdas karkaistuja laseja ajoneuvoteollisuudelle ja Laitilan tehdas pääasiassa linja-autojen tuulilaseja. Lisäksi Espoossa toimii varaosalasien tukkuliike ja Laitilassa laivojen vaativia lasituksia toimittava yksikkö Pilkington Marine. Lisäksi Pilkingtonilla on Suomessa floatlasia valmistava Pilkington Lahden Lasitehdas Oy ja eristyslaseja rakennusteollisuudelle valmistava Pilkington Nivala Oy. Vuonna 2008 avataan uusi rakennuslasitehdas Forssaan. Pilkingtonin Suomen yhtiöiden liikevaihto viime tilikaudella oli 171 miljoonaa euroa. Pilkingtonilla työskentelee Suomessa 1200 henkilöä. /1/

### 3.2 Lasin mittaaminen

Linja-autojen ja työkonien tuulilasien valmistusprosessissa yksi vaihe on oikeanmuotoiseksi leikatun lasin taivuttaminen kaarelle sitä voimakkaasti kuumentamalla. Tämä tapahtuu uunissa, jossa on ohjelmoitavia lämpövastuksia. Vastusten lämpötilaa ja etäisyyttä lasiin muokkaamalla luodaan lasille tuotekohtainen lämmityskuvio. Uunin työntekijät lataavat oikeat lämmityskuviot järjestelmään vaihdettuaan tuotteen uuniin. Lasi käytetään uunissa tuotekohtaisella muotilla, ja lämmityskuvion sekä muotin muodon mukaan lasi taipuu haluttuun muotoonsa.

Lasin pussimittaus tarkoittaa sitä mittausta, joka taivutetun lasin reunojen ja eniten uunissa vajonneen kohdan väliin jää. Tämä mitta ilmoitetaan millimetreinä ja tulos vaihtelee tuotteesta riippuen noin 5–100 millimetriä. Toki myös tästä mittavälisestä poikkeavia tuotteita löytyy. (Kuva 1.)



Kuva 1. Lasin poikkileikkaus muotilla. Pussimitta on merkitty kuvassa nuolella.

Lasi mitataan aina, kun se ajetaan uunista ulos. Tässä tapauksessa mittalaite sijaitsee uunin oven päällä siten, että lasi ajetaan ulos sen alta. Tällöin lasin poikkileikkauksesta saadaan mittapistedataa, jonka avulla pussimitta lasketaan.

Laserin pettäessä tai antaessa laaturajoista poikkeavia tuloksia lasi voidaan mitata vielä käsimitalla. Sitä ei kuitenkaan voida pitää absoluuttisen tarkkana, koska käsimitaus suoritetaan joka kerta käsivaralta ja jokainen mittauskerta on hieman erilainen. Itse lasintaivutusprosessista ei juuri löydy kirjallisuutta, koska menetelmä on tiilajan itsensä kehittämä, eikä sellaisenaan muiden käytössä.

## 4 KÄYTETYT TYÖKALUT JA MENETELMÄT

Käytetyt työkalut olivat mittausanturi, mittausyksikkö sekä kannettavalle tietokoneelle kehitetty ohjelmisto. Mittaustulosten vertailussa ja ohjelman testaamisessa käytettiin apuna käsimittaa sekä vanhaa olemassa olevaa lasermittausjärjestelmää.

### 4.1 Anturi

Anturiksi valittiin lopulta laser-anturi, OMRON ZS-L -tuotesarjan mittari. Ensin kokeiltu huokeampi ultraäänianturi osoittautui liian epätarkaksi ja häiriöalttiiksi. ZS-LD200 -anturin mittausrajat ovat asetettavissa välille 250–500 millimetriä, joten anturi sopii hyvin käytettäväksi tämän järjestelmän kanssa. Anturin näytteenottonopeus on yksi näyte 110 mikrosekunnissa, ja ohjelmisto käyttää anturia tällä nopeudella. Liitännänä mittausyksikköön toimii luotettava RS-485-sarjaväylä.

Anturilla on oma mittausyksikkönsä, josta säädetään mittausrajoja sekä anturin nopeutta ja herkkyyttä. ZS-LD200 olisi ollut mahdollista liittää myös suoraan tietokoneeseen siinä olevan USB-liitännän avulla, mutta tätä ei kokeiltu, koska ohjelmisto oli jo



säädetty toimimaan ADAM-4017-mittausyksikön kanssa, joka oli välttämätön ultraäänianturille. ADAM-4017-mittausyksikön ollessa liitettynä järjestelmään anturia on mahdollista vaihtaa ilman koodimuutoksia, ja pienellä lisäohjelmoimisella järjestelmä saadaan tallentamaan monelta anturilta yhtä aikaa.

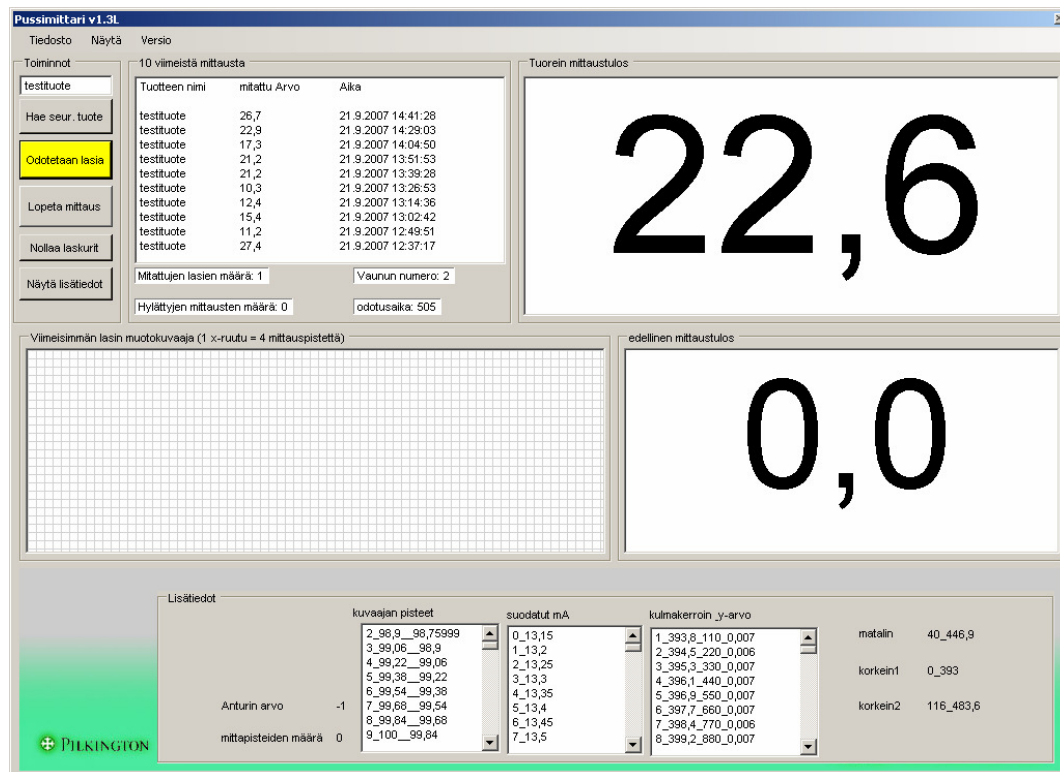
#### 4.2 Mittausyksikkö

Mittausyksikkö koostuu ADAM-4017-tiedonhankintamoduulista, sekä ADAM-4561 RS232/422/485-USB -adapterista. Tiedonhankintamoduuli lähettää tietonsa RS-485-liitännän kautta, joten kannettavaan tietokoneeseen liittämiseksi ADAM-4651-adapteri oli aluksi ultraäänianturin takia pakollinen, koska kannettavassa tietokoneessa ei ole RS-485-sarjaliitintää. Lisäksi mittausyksikössä on muuntaja, josta mittausmoduuli ottaa käyttövirtansa. Adapterimoduuli ottaa virtansa tietokoneen USB-väylästä.

ADAM-4017-moduulissa on kahdeksan kanavaa, eli siihen saisi liitettyä kahdeksan eri anturia. Tässä projektissa tarvittiin vain yhtä kanavaa, mutta laajennusmahdollisuus on olemassa. Monen anturin yhtäaikainen lasinmittaus erilaisilla antureilla on mahdollista tulevaisuudessa, jos ohjelmistoa kehitetään eteenpäin. ADAM-4017 on ohjelmoitavissa sarjaporttiin lähetettävillä käskyillä, joilla saadaan säädettyä monia eri ominaisuuksia.

#### 4.3 Mittausohjelmisto

Mittausohjelmiston käyttö tapahtuu suurimmaksi osaksi pääikkunan kautta. Pääikkunassa on painikkeet mittauksen aloitukselle ja lopetukselle sekä muille toiminnoille. (Kuva 2.) Pääikkunan hallitsevin osa on mittaustuloksen näyttöruutu, joka esitetään suurella fontilla selkeyden vuoksi. Näin ohjelman toimintaa voi seurata hieman kauempeakin. Pääikkunassa näytetään myös mitatun lasin kuvaaja, edellinen mittaustulos sekä historiatietoja kyseiseltä tuotteelta. Nämä historiatiedot haetaan MS Access -tietokannasta, jonne mittaustulokset aina tallennetaan.

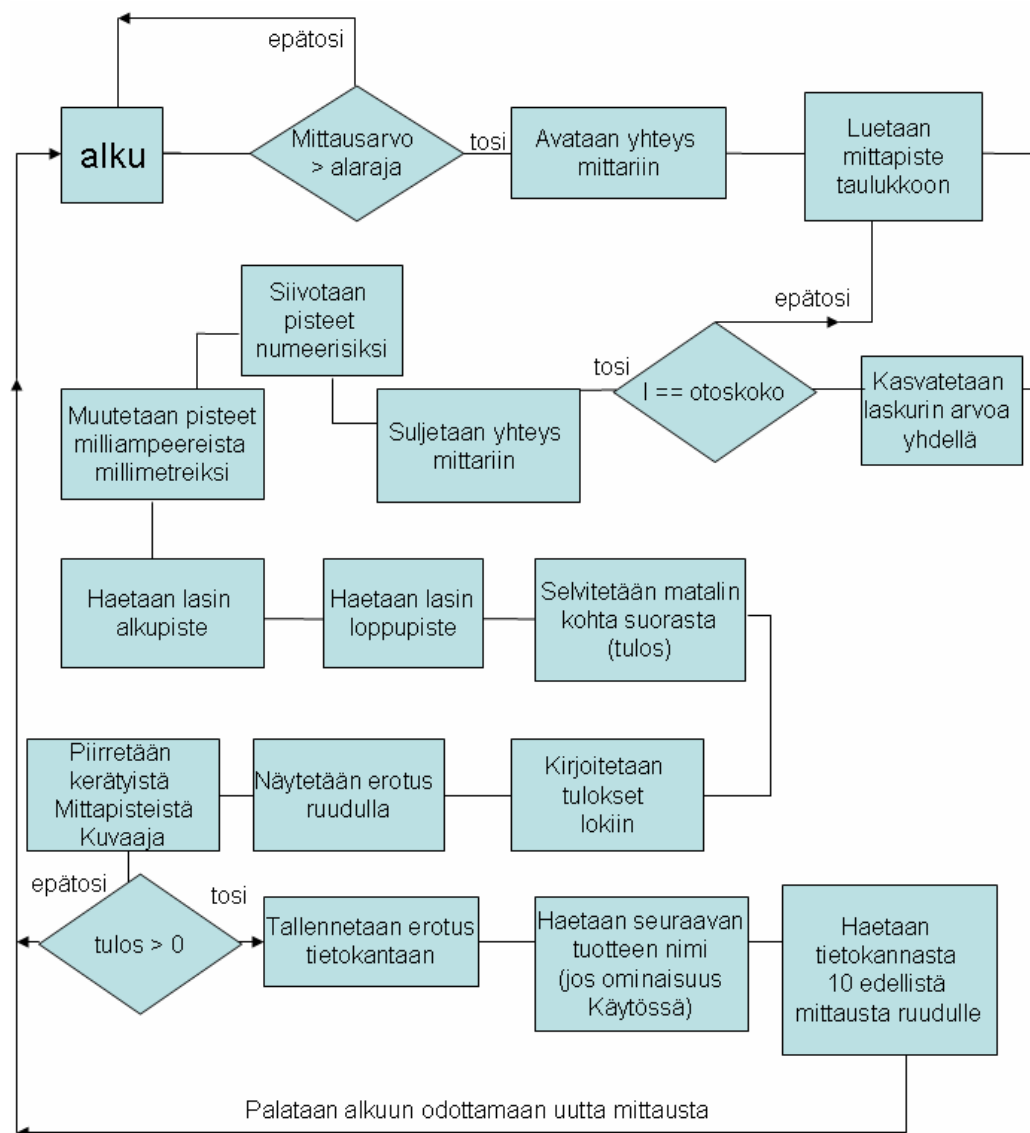


Kuva 2. Ohjelman pääikkuna. Pääikkunan kentät vasemmalta oikealle: painikkeet, tuotehistoria, viimeisin mittaus, käyrä lasin pinnasta, edellinen mittaustulos sekä lisätietoja mittapisteistä.

Mittausohjelmisto toteutettiin Microsoft Visual Studio 2005 Express edition -työkalulla Visual Basic.NET -kieltä käyttäen. Visual Basic.NET oli hyvä valinta projektin toteutukseen, koska ohjelmointikieli oli tekijälle entuudestaan tuttu ja työkalu ilmainen. Visual Basic.NET on korkean tason ohjelmointikieli. Tämä tarkoittaa käytännöllisesti katsoen sitä, että laitetason on vaikea päästä suoraan kiinni ja kielen jotkin ilmaisut ovat hieman rajoittuneita. Se tuo yksinkertaisuutta ja selkeyttä monien abstraktien asioiden käsittelyyn. Visual Basic.NET on teoriassa täysin Windows-riippuvainen ohjelmointikieli, sillä tehtyjen ohjelmien ajaminen vaatii Microsoftin www-sivuilta ladattavan .NET Framework -kirjastokokoelman asentamista. Nykyään joidenkin Visual Basic.NET -kielellä tehtyjen ohjelmien ajaminen on kuitenkin mahdollista esim. Linuxista Windows-emulaattoria käyttäen.

Ohjelmiston pääominaisuudet ovat tuotteen mittaus, mittapisteiden graafinen esitys, pussituksen laskeminen sekä tuotehistorian esittäminen. Lisäksi ohjelmistoon sisältyy

asetukset-sivu, josta käyttäjä voi säätää ohjelman muuttujia. Mittauksen sekä pisteiden laskemisen voi parhaiten havainnollistaa kaaviona. (Kuva 3.)



Kuva 3. Yksinkertaistettu kaavio mittausohjelman toiminnasta.

Kaaviosta on jätetty selkeyden vuoksi pois osa pienemmistä tarkistuksista ja laskureista, jotka ohjaavat ohjelman toimintaa. Mittauspisteiden käsittely ja vertailupisteiden haku sekä analysointi ovat hieman kaaviossa kuvattua monimutkaisempia tapahtumia. Lasin korkeimpien kohtien (kaaviossa alku ja loppupisteet) avulla tapahtuva matalimman kohdan haku tapahtuu käyttämällä pisteen etäisyys suoralta -menetelmää. /2/

Mittausprosessi voidaan jakaa kolmeen tilaan tai vaiheeseen. Odotustilassa laskuri laskee sekunteja seuraavaan tilaan siirtymiseen ja mittausta ei tapahdu. Kun ohjelma siirtyy mittaustilaan, alkaa mittapisteiden keräys, jonka jälkeen tutkimustilassa suoritetaan tulosten analysointi ja esitys.

Alla on esitetty ohjelman toiminnallisen ytimen eli Mittaaja2-aliohjelman koodi. Mittaaja2-aliohjelma käynnistyy ”Aloita mittaus” -painikkeesta. Ohjelma pysyy ikuisessa silmukassa, jollei sitä erikseen pysäytetä ja lukee mittausdataa, kunnes mittausarvo asettuu rajojen väliin. Tällöin kutsutaan aliohjelmaa TaulukonKeräys, jonka suoritus päättyy, kun lasi on tullut kokonaan uunista ulos. Seuraavaksi ajetaan aliohjelma TaulukonKäsittely, joka hakee taulukosta kriittiset arvot ja suorittaa niiden perusteella pussimittauksen. Kyseinen aliohjelma käydään tarkemmin läpi omassa kappaleessaan. Aliohjelma Pussikuva piirtää käyrän lasin muodosta, jonka jälkeen tallennetaan tulos tietokantaan aliohjelmalla TallennusKantaan. Lopuksi Odottaja-niminen aliohjelma jää laskemaan odotusaikaa, jonka jälkeen ohjelma palaa alkutilaan mittaamaan rajoja.

```
Public Sub mittaja2()
```

```
Control.CheckForIllegalCrossThreadCalls = False
```

```
Dim mitArvo As Single = 0
```

```
Dim i As Integer = 1
```

```
Dim laskuri As Integer = 0
```

```
alaraja = AseTaulukko(3)
```

```
yläraja = AseTaulukko(7)
```

```
otoskoko = AseTaulukko(2)
```

```
Do While i = 1
```

```
mitArvo = (Trim(sendAndRead()))
```

```
Label6.Text = mitArvo
```

```
If mitArvo > alaraja And mitArvo < yläraja Then
```

```
taulukonKeräys()
```

```
taulukonKäsittely()
```

```

Pussikuva()
TallennusKantaan()
odottaja()

ElseIf mitArvo <= 0 Then
    trd.Abort()
    MsgBox("ei toimi")
End If
ReDim pisteTaulukko(0 To otoskoko - 1)
Loop
End Sub

```

#### 4.3.1 Mittapisteiden kerääminen anturilta ohjelmaan

Laseranturi syöttää jatkuvasti mittadataa ADAM-mittausyksikölle, joka tulkitsee tietoa milliampeeriasteikolla 4–20 milliampeeria. Ohjelmisto hakee mittausyksiköltä aina senhetkisen arvon funktiolla `SendAndRead()` ja tallentaa sen taulukkoon.

`SendAndRead()` toimii siten, että se lähettää mittausyksikölle kutsun hakea dataa kanavasta 1, johon laseranturi on kytketty. Sen jälkeen ohjelmisto lukee arvon mittausyksiköltä käyttäen Visual Basicin sarjaportinlukumetodia. Dataa luetaan tavu kerrallaan lopetusmerkkiin asti, jonka jälkeen palautetaan kerätty merkkijono. Myöhemmissä vaiheissa mittausta siivotaan datasta tarpeettomat merkit pois ja muunnetaan se single-tietotyyppiseksi.

#### 4.3.2 Mittauksen vaiheet yksityiskohtaisesti

Anturin sensorin havaitessa rajan ylityksen ohjelma siirtyy odotustilasta mittaustilaan, jossa se tallentaa taulukkoon halutun määrän mittapisteitä. Mittaus lopetetaan vasta, kun haluttu määrä mittapisteitä on saatu, eli ohjelma mittaa lattiaa lasin mentyä ohi. Näin on tehtävä, jotta anturin äkillinen häiriö ei katkaisisi mittausta. Jos sensorin arvo ylittää rajan liian nopeasti eli mittausta käynnistyy anturin häiriöstä eikä lasia ole, mittausta hylätään ja siirrytään takaisin odotustilaan.

Kun mittapisteet on kerätty, siirrytään mittaustilasta tutkimustilaan, jossa taulukko suodatetaan eri funktioiden läpi. Tässä kohdassa aletaan käyttää struktuuria, johon säilötään yhtä mittapisteitä kohden kaksi arvoa: Y-akselin arvo eli edellisessä aliohjelmassa kerätty arvo sekä tässä kohdassa ohjelman luoma x-akselin arvo, joka on indeksin kerrannainen 110:stä. Tämä luku johtuu anturin nopeudesta koska yksi arvo mitataan noin kerran 110 millisekunnissa. Struktuuria käydään läpi merkiten selvästi väärät, anturin häiriöistä johtuvat arvot.

Väärien arvojen merkitseminen tapahtuu kulmakerrointekniikkaa käyttäen. Jos kahden peräkkäisen pisteen kulmakertoimet eroavat toisistaan liikaa, jälkimmäinen merkitään nollaksi, jotta se on helppo ohittaa tulosten käsittelyssä. Alla esitellään osa TaulukonKäsittely-aliohjelmasta esimerkin vuoksi. Koko ohjelman esittäminen koodina ei ole tarkoituksenmukaista.

```
'lasketaan silmukassa kahden peräkkäisen pisteen kulmakertoimia.
    For i = 1 To pisteet.Length - 1
        y1 = pisteet(i).y
        y2 = pisteet(i - 1).y
        x1 = pisteet(i).x
        x2 = pisteet(i - 1).x
        kulmak = (y2 - y1) / (x2 - x1)

        'vertaillaan kulmakertoimia viitearvoihin, jos on liian suuri tai
        pieni niin edellinen piste
    'muutetaan nollaksi jotta se on helppo ohittaa myöhemmin.
        If (kulmak >= 0.1 Or kulmak <= -0.1) Then
            pisteet(i - 1).y = 0
        End If
        TextBox6.Text = TextBox6.Text & i & "_" &
Math.Round(pisteet(i).y, 1) & "_" & pisteet(i).x & "_" &
Math.Round(kulmak, 3) & vbNewLine
        'kerätään samalla viimeinen piste jota käytetään suoran
        loppupisteenä
        If pisteet(i).y <> 0 Then
            viim_y = pisteet(i).y
            viim_x = pisteet(i).x
            viim_indeksi = i
        End If
    Next
```

```

'sitten kerätään eka kelpo piste. HUOM lopusta alkuun!
For i = pisteet.Length - 1 To 0 Step -1
    If pisteet(i).y <> 0 Then
        eka_y = pisteet(i).y
        eka_x = pisteet(i).x
        eka_indeksi = i
    End If
Next

'näytetään eka ja viimeinen piste ruudulla (vain y-
koordinaatti)
Label9.Text = eka_indeksi & "_" & eka_y
Label10.Text = viim_indeksi & "_" & viim_y

'lasketaan pussimitta "kaukaisin piste suoralta" menetelmällä
(distancepointline-funktio)esitetään tulos ruudulla.

For i = eka_indeksi To viim_indeksi
    If pisteet(i).y <> 0 And pisteet(i).y > alaraja And
        pisteet(i).y < yläraja Then
        etäisyystaulukko(i) = DistancePointLine(pisteet(i).x,
            pisteet(i).y, eka_x, eka_y, viim_x, viim_y)
    End If
    If i > 0 Then
        If etäisyystaulukko(i) > pussitus Then
            pussitus = etäisyystaulukko(i)

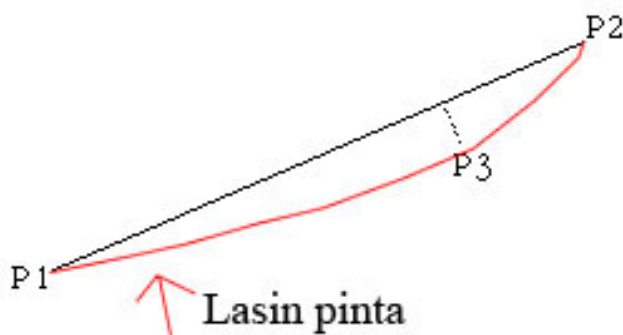
            matalin_i = i
            Label26.Text = matalin_i & "_" & pisteet(i).y
        End If
    End If
Next

```

Suodatetut mittapisteet esitetään käyränä, joka muodostaa suuntaa antavan kuvan lasin poikkileikkauksesta. X-akselilla on struktuurin x-arvo, eli se kuvaa aikaa, ja y-akselilla on pisteen struktuurin y-arvo mitattuna kuvaajaikkunan yläreunasta. Suurempi mittauspisteen arvo tarkoittaa lasin pinnan suurempaa etäisyyttä anturista. Kuvaaja näyttää lasin muodon ilman mittarin häiriöiden tuottamia virhearvoja, eli se on

melko lähellä todellisen lasin muotoa. Aiemmin nollassi merkityt virheelliset taulukon arvot siis ohitetaan piirroksessa. Tämä selvitetään tarkemmin omassa luvussaan.

Lasin syvin piste eli lopullinen mittauksen tulos haetaan käyttämällä menetelmää, jossa etsitään pistejoukosta kaukaisin piste(P3) verrattuna pistejoukon alku-(P1) ja loppupisteen(P2) lävistävään suoraan. (Kuva 4.) Tällöin ei ole väliä, onko lasimuotti vinossa anturiin nähden, eli esim. lasin etupää saa olla matalammalla kuin takapäätä, ja matalin piste löytyy silti oikein. /2/



Kuva 4. Pisteiden etäisyys suorasta.

Onnistuneen mittauksen ja pussituslaskun jälkeen ohjelma tallentaa pussimitan, tuotteen nimen ja aikaleiman Microsoft Access -tietokantaan ja näyttää ruudulla omassa ikkunassaan kymmenen viimeistä mittaustulosta, jotka on tallennettu kyseiselle tuotteelle. Tämän jälkeen ohjelma siirtyy odottamaan seuraavaa mittausta. Koska myös lasin lastaaminen uuniin laukaisee sensorin rajan, onnistuneen mittauksen jälkeen ohjelma siirtyy määritellyn mittaiseen odotustilaan, jossa ei tapahdu mitään. Odotustila estää useimmat ylimääräiset anturin häiriöistä johtuvat mittaukset. Odotustilan odotusajan voi määrittää ohjelman asetuksista haluamukseen. Ajan tulee olla tarpeeksi pitkä, jotta seuraava lasi ehditään lastata uuniin ongelmitta, mutta tarpeeksi lyhyt, jotta mittausohjelma on ajoissa valmiina mittaamaan seuraavan lasin. Logiikan pettäessä anturin häiriön takia voidaan seuraava lasi määrätä mitattavaksi manuaalisesti.



### 4.3.3 Liittymä tietokantaan

Liittymä Microsoft Access -tietokantaan toteutettiin käyttämällä Dataset-tekniikkaa, joka toimii rajapintana Accessin ja Visual Basic-ohjelman välillä. Ohjelmaa voi käyttää, vaikkei aktiivista yhteyttä Accessin tietokantaan olisikaan koko ajan saatavilla. Muuttuneet tiedot päivitetään Access-kantaan, kun yhteys on taas käytettävissä. Dataset-tekniikka sisältää DataTable-luokan, johon kannan tiedot tallentuvat.

DataTable-luokkaa ohjataan Visual Basicissa TableAdapter-luokalla, jonka metodeilla muokataan ja esitetään tietokannan tietoja ohjelmassa. Tableadapter-luokka toimii SQL-lauseiden avulla, halutuista operaatioista tietokantaan luodaan oma kysely TableAdapteriin. Haluttaessa mittaustuloksia voi tutkia tuotekohtaisesti normaalien SQL-kyselyjen avulla Accessista.

Tietokannan rakenne on yksinkertainen. Kanta sisältää yhden taulun, jossa on kolme tietuetta. Nämä ovat tuotenimi, mittausrvo ja aikaleima. Aikaleima toimii kannan pääavaimena, ja se generoidaan jokaiselle kantaan tallennetulle mittausrvolle erikseen. Tuotenimi-tietue tallennetaan aikaleiman ja mittausrvon rinnalle, jotta tiedetään, mistä lasista on kysymys.

### 4.3.4 Kuvaajan piirto

Kuvaajan piirto tapahtuu aliohjelman Pussikuva avulla. Kuvaaja lasin muodosta piirretään Visual Basic -kielessä olevan System.Drawing.Graphics-luokan ja siihen kuuluvien Drawing.Pen- ja Drawing.Drawline-metodien avulla. Jotta struktuurin pisteet näkyisivät oikeassa kohdassa ruudulla, niiden arvoja muutetaan jakamalla pisteet viidellä. Tämä muutos vaikuttaa ainoastaan kuvaajan sijaintiin näytöllä.

Drawline-metodilla suoritetaan käyrän piirto. Metodin argumentteina ovat Pen, x1,y1,x2 ja y2. Pen tarkoittaa ennalta määritettyä kokoelmaa, jonka ominaisuuksia ovat muun muassa viivan väri sekä paksuus. Pisteet x1 ja y1 merkitsevät viivan alkupisteen ja x2 ja y2 loppupisteen.

### Piirtäminen tapahtuu seuraavasti:

'piirretään kuvaajaa. jos pisteen y-koordinaatti on rajojen ulkopuolella hypätään seuraavaan

```

For i = 2 To pisteet.Length - 1
    TextBox9.Text = TextBox9.Text & i & "_" & pisteet(i).y &
    "_" & pisteet(i - 1).y & vbCrLf

    If pisteet(i).y < araja Or pisteet(i).y > yraja2 Then
        'jos piste rajojen ulkopuolella, kelataan eteenpäin
        kunnes piste on taas rajoissa ja jatketaan piirtoa
        siitä.'näin seulotaan peräkkäisiä virheitä.
        While (pisteet(i).y < araja Or pisteet(i).y > yraja2)
            If i < pisteet.Length - 1 Then
                i = i + 1
            Else
                Exit Sub
            End If
        End While
        viiva.DrawLine(k, pisteet(ok).x, pisteet(ok).y,
        pisteet(i).x, pisteet(i).y)
    Else
        'tässä normaalin pisteen piirto, edellisestä nykyiseen
        viiva.DrawLine(m, pisteet(i - 1).x, pisteet(i - 1).y,
        pisteet(i).x, pisteet(i).y)
        ok = i
    End If
Next

```

Viivaa piirretään siten, että aloituspisteenä on aina edellinen mittauspiste, pisteet(i-1) y ja x, jolloin viivasta tulee jatkuva. X-akselilla pisteen muutos on vakio, eli seuraava piste on aina 110 millisekuntia edempänä. Jos struktuurin y-arvo ei ole rajojen sisällä, aletaan tarkastella arvoja siitä eteenpäin, ja kun vastaan tulee taas kelvollinen arvo, jatketaan piirtoa. Näin lasin kuva säilyy melko tarkasti oikean lasin pintaa muistuttavana ja pituus säilyy samana, vaikka pisteitä hylätäänkin välistä.

#### 4.3.5 Tuotenimen haku

Jotta järjestelmä olisi käytössä täysin automaattinen, tulee mitattavan tuotteen nimi hakea verkon kautta uunilta, jossa taivutusohjelmat sijaitsevat. Tämä on toteutettu Microsoftin OLE automation -rajapinnan avulla. Rajapinta on tarkoitettu tiedon lukemiseen ja kirjoittamiseen eri ohjelmista, kuten Microsoft Excelistä tai Microsoft Accessista. OleDb-luokan metodien avulla annetaan ohjelmalle tarkat käskyt, mistä ja minkä tyyppisestä tiedostosta hakea juuri tietyn solun sisältämä data. Tässä ohjelmassa tuotenimitaulukko on mahdollista hakea esimerkiksi verkon yli Microsoft Excel-tilukosta, jota erillinen ohjelma päivittää uunin ohjausjärjestelmästä.

Koska uuni ei välttämättä aina ole täynnä, uunissa on resepti myös tyhjälle lasivau-nulle. Ohjelma tunnistaa uunilta saadun tuotenimen TYHJA, ja sellaisen havaitessaan hyppää automaattisesti seuraavaan soluun, koska tyhjää vaunua ei oteta ulos uunista turhaan.

Seuraavan tuotenimen haku suoritetaan ohjelmassa aina, kun edellisen tuotteen pus-simitta on onnistuneesti tallennettu tietokantaan. Tämän ominaisuuden saa tarvittaes-sa pois päältä, jos mittari siirretään sellaiselle uunille, jossa mitataan vain yhtä tuotet-ta.

#### 4.3.6 Asetukset-sivu

Asetukset-sivulla voidaan muuttaa monia muuttujia, joita tarvitaan ohjelman käytös-sä. Lokitiedoston kirjoittaminen ja tiedoston haluttu polku ovat asetettavia ominai-suuksia, kuten myös automaattinen tuotenimen haku ja Tuotenimi-tilukon polku. (Kuva 5.)

Kuva 5. Asetukset-sivu

Käytettävän COM-portin voi valita, sillä Windows XP -käyttöjärjestelmä hakee mitauslaitteelle uuden virtuaalisen COM-portin joka kerta, kun tietokone käynnistetään uudelleen. Tämän portin nimi voi siis vaihdella. Mittalaitteelle lähetettävällä käskyllä voi muuttaa esimerkiksi haluamaansa kanavaa mittalaitteelta, tai muita mittalaitteen asetuksia. Mittalaitteelle lähetettävistä komennoista kertova listaus löytyy ADAM-4017-manuaalista, joka toimitetaan mittalaitteen mukana.

Milliampeeri/millimetri muunnosta varten tarvittavan kaavan arvoja voi muuttaa mittarin kalibrointi -kohdassa. Tämä saattaa olla tarpeen, jos mittausanturi vaihdetaan tai sen asetuksia muokataan. Kaavan  $mmArvo = (((mAarvo - a) * b) + c)$  johdetaan edempänä olevan mallin mukaan.

Lasermittarin asetukset on säädetty siten, että sen tarkkuusalue 4 milliampeeria – 20 milliampeeria vastaa millimetreissä arvoja 250–500. Tällä milliampeerivälillä mittarin arvot vastaavat edellä mainittuja millimetriarvoja lineaarisesti, ja näiden tietojen avulla johdetaan muunnoksessa käytettävä kaava. Suoran yhtälön kaavan  $y = kx + b$  avulla ratkaistaan seuraava yhtälöpari:

$$\{ 250 = k * 4 + b$$

$$\{ 500 = k * 20 + b$$

joka voidaan kirjoittaa muotoon

$$\{ 4k + b = 250$$

$$\{ 20k + b = 500$$

Seuraavaksi vähennetään yhtälöt toisistaan k:n ratkaisemista varten

$$4k + b - (20k + b) = 250 - 500$$

$$- 16k = -250 \quad |:-16$$

$$k = 15.625$$

Sen jälkeen toinen muuttuja ratkaistaan sijoittamalla saatu k ensimmäiseen yhtälöön

$$4 * 15.625 + b = 250$$

$$4*15.625 -250 = b$$

$$b = 187,5$$

Lopuksi sijoitetaan k ja b suoran yhtälöön  $y = kx + b$

$$y = 15,625*x + 187,5$$

Tämä kaava toimii siis x:n arvojen ollessa välillä 4–20. Y:n arvoksi muodostuu siten piste väliltä 250 – 500. Itse ohjelmassa sama kaava on kirjoitettu muotoon  $mArvo = ((mArvo * 15.625) + 250) - 62.5$  eli  $((mArvo * a) + b) - c$ , jotta kaikki arvot ovat vapaasti muokattavissa, jos mittarin rajoja muutetaan tai mittaria vaihdetaan.

Koemittaus-painikkeella voidaan testata yhteys mittauslaitteelle hakemalla anturilta yksi mittauspiste eli anturin senhetkinen arvo. Tulos näytetään sekä millimetreinä että milliampeereina.

Mittauspisteiden määrää eli otoskoko voidaan muokata. Otokoko tulee mitoittaa siten, että mittauspisteitä olisi tarpeeksi pisimmällekkin lasille, mutta kuitenkin mahdollisimman vähän, ettei kone mittaa tyhjää lasin ulosajon jälkeen turhan kauan. Suositeltava mittauspisteiden määrä on välillä 170–220 pistettä, riippuen uunissa olevasta tuotevalikoimasta.

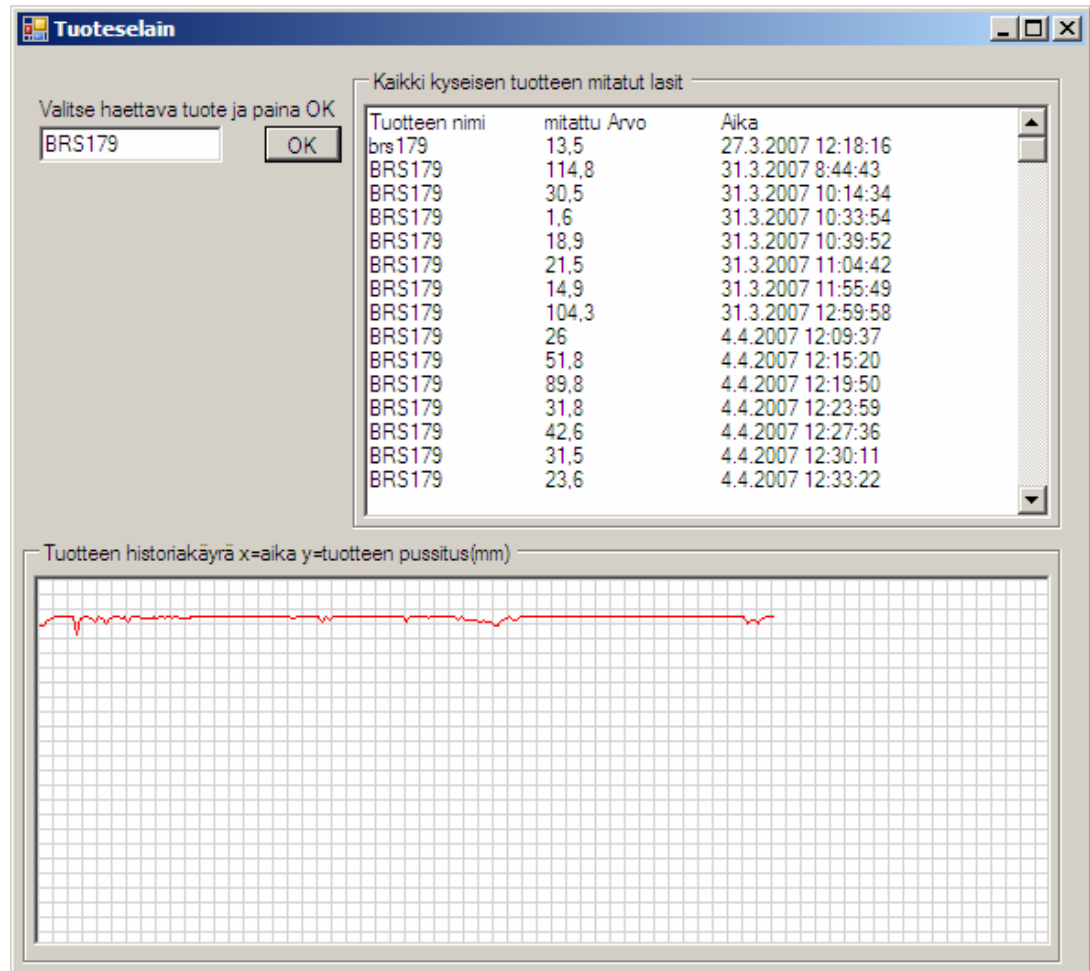
Odotusaika eli aika, jonka ohjelma käyttää odottamiseen onnistuneen mittauksen jälkeen, annetaan sekunteina. Uunista riippuen odotusaika voi olla 5-15 minuuttia. Odotusajan kuluessa mittaus ei ole toiminnassa eikä ohjelma ole tällöin altis häiriöille.

Mittausrajat ovat hyvin tärkeä asetus mittausten onnistumisen kannalta. Rajat määrittävät sen alueen, missä mittaus aloitetaan, ja missä rajoissa mittapisteiden tulee pysyä. Rajat määritellään asettamalla ylä- ja alaraja. Näiden rajojen väliin jäävä mitta-alue on niin sanottua aktiivista aluetta, johon osuttuaan mittausanturi aloittaa mittausprosessin.

Esimerkiksi oletusarvoina olevat n. 251 mm ja n. 499 mm kertovat, että anturin havaitessa arvon, joka sopii välille mittarista 251 millimetriä lattian pintaa kohti ja siitä n. 250 millimetriä eteenpäin, aloitetaan mittaus. Ohjelma tarkkailee jatkuvasti, että anturin arvo on yhä annettujen rajojen sisäpuolella. Rajoja käytetään myös taulukon käsittelyssä mittapisteitä laittomia mittapisteitä siivottaessa. Sivulta poistutaan ja asetukset tallennetaan tekstitiedostoon ja ladataan käyttöön Tallenna-painikkeella.

#### 4.3.7 Tuote-selain

Tuote-selain on pieni lisätyökalu mittausohjelmistoon, joka tulostaa ruudulle listauksen käyttäjän syöttämän tuotenimen perusteella sekä piirtää kuvaajan tuotehistoriasta. Tiedot haetaan MS Access -tietokannasta samaan tapaan kuin mittausikkunassakin. Tällä kertaa vain näytetään kaikki tulokset eikä ainoastaan kymmentä viimeistä. (Kuva 6.)



Kuva 6. Tuote-selain-ikkuna

Kuvaaja piirretään myös täysin samaa tekniikkaa käyttäen kuin mittapisteistä muodostettu käyrä ohjelman pääikkunassa. Piirtoon käytetään valmiita mittaustuloksia, ja normaalista mittauksesta poikkeavat arvot erottuvat kuvaajassa piikkeinä joko ylös- tai alaspäin.

## 5 TULOKSET

Työn tulokset on eritelty projektiin käytetyn ajan kuvailemiseen sekä omiin havaintoihin ja parannusehdotuksiin. Työn luonteesta johtuen varsinaisia tuloksia saadaan vasta, kun järjestelmä on ollut jonkin aikaa käytössä. Toisin sanoen, työn arvo on mi-

tattavissa vasta, kun käytäntö on osoittanut, kuinka paljon järjestelmä säästää työntekijöiden aikaa ja helpottaa lasin mittausta.

## 5.1 Projektin kulku

Projektisopimus kirjoitettiin lokakuussa 2006, ja sen jälkeen alkoi suunnitteluvaihe. Tarvittavat laitteet oli määritettävä ja hankittava sekä tämän jälkeen ohjelmisto koodattava ja testattava toimivaksi käytössä.

### 5.1.1 Hankinnat

Mittausjärjestelmä päätettiin toteuttaa Advantechin valmistamilla ADAM- moduuleilla, koska muilta valmistajilta ei löytynyt yhtä tehokasta ja taloudellista ratkaisua USB-liitäntäiseen monikanavaiseen mittausjärjestelmään. Saman tuoteperheen linkitettävillä moduuleilla taattiin yhteensopivuus. Ylimääräinen laseranturi löytyi yrityksestä jo valmiiksi, eikä sitä niin ollen tarvinnut hankkia erikseen.

Kannettavan tietokoneen valinta ja hankinta jäi työn tilaajan hoidettavaksi, koska sille ei ollut muita kriteerejä kuin Microsoft Windows XP -käyttöjärjestelmä ja USB-liitännät. Nämä löytyvät lähes jokaisesta markkinoilla olevasta kannettavasta tietokoneesta. Niinpä yritykseen hankittiin HP Compaq nx6310 -mallinen kannettava tietokone.

### 5.1.2 Suunnittelu

Kun tarvittavat laitteet oli hankittu, siirryttiin ohjelmiston tarkempaan suunnitteluun. Koska aihe oli tekijälle uusi ja vieras, kuten myös työkalut, tutkimiseen, tutustumiseen ja testiohjelmien laatimiseen kului yllättävän paljon aikaa. Varsinaisen ohjelmiston rungon suunnittelu ja toteutus alkoivat vasta vuoden 2007 alussa.



Järjestelmän tarvitsemat ominaisuudet, kuten yhteys virtuaaliseen sarjaporttiin, liittymä tietokantaan ja mittauspisteiden suodatus, ohjelmoitiin pala palalta yhdeksi kokonaisuudeksi. Toisinaan ohjelmiston työstäminen vaati testausta yrityksen tiloissa, lasiuunilla. Samalla kerättiin työntekijöiltä ehdotuksia ja parannuksia ohjelmaan, koska heidän työtään projektissa on tarkoitus helpottaa.

### 5.1.3 Ohjelmiston parantelu ja testaus

Ohjelmavirheiden etsimiseen ja korjaukseen sekä koodin optimointiin ja siistimiseen käytettiin paljon aikaa. Tämä oli tarpeellista, koska työkalut olivat projektin alussa tekijälle melko vieraita, ja Visual Basic kielen opettelu myötä myös vanhoja osia koodista oli tehtävä uudestaan ja paremmin, koska tietämys ja taito karttuivat ohjelmoimissa.

Testausta suoritettiin yrityksessä kevään 2007 ajan, ja mittauslogiikkaa hiottiin toimivammaksi. Projektin aikataulu ylitettiin johtuen pitkästä suunnittelu- ja hankintavaiheesta, sekä ohjelmointivaiheessa uusien asioiden ja työkalujen käytön opettelusta.

Testauksessa eniten aikaa vei mittausrajojen etsintä sekä mittapisteiden oikeiden arvojen haku pussilaskutoimitusta varten. Ohjelman testaaminen oli melko hidasta, koska uunista tulee uusi lasi noin 15 minuutin välein. Aina koodin muokkaamisen jälkeen piti siis odottaa kyseinen aika.

### 5.2 Työn tekijän havainnot

Projekti osoittautui odotettua vaikeammaksi ja pitkäkestoisemmaksi. Havaittiin kuitenkin, että Visual Basic.NET oli hyvinkin sopiva työkalu tällaisen ohjelmiston luomiseen helpon omaksuttavuutensa ja näennäisen yksinkertaisuutensa takia.

Ohjelman toteutuksessa tuli ilmi monta erilaista ongelmaa, jotka kuitenkin ajan myötä voitettiin. Mittauspisteiden kerääminen ja varsinkin pisteiden käsittely sekä kuvaajan piirtäminen mietittiin uudelleen monta kertaa. Mittarin herkkyydestä johtuvat häi-

riöt tuottivat paljon ongelmia, jotka pitkäjänteisellä testauksella ja anturin vaihtamisella kuitenkin saatiin selvitettyä.

Valmis ohjelmisto mittaa lähes kaikki tuotteet oikein, muutamia poikkeuksia lukuun ottamatta. Samoja tuotteita ei vanha mittausjärjestelmäkään mitannut, esimerkiksi joillakin tuotteilla oli esteenä muotin fyysinen muoto. Nämä poikkeustuotteet joudutaan yhä mittaamaan käsin.

### 5.2.1 Tuloksia

Opinnäytetyön valmistumisen hetkellä ohjelmisto on testikäytössä ja sen kehitystä jatketaan vielä opinnäytetyön valmistuttua. Ennen lopullista käyttöönottoa järjestelmälle on tehtävä perusteelliset testit tarkkuuden määrittämiseksi ja sen liittämiseksi osaksi laatuja järjestelmää. Jos tarkkuus on riittävä, järjestelmä jää osaksi laadunvalvonnaprosessia, ja mahdollisesti se otetaan käyttöön myös Ylöjärven tehtaalla.

### 5.3 Parannusehdotukset

Parannusehdotuksia on monenlaisia. Laseranturin ja sen oman mittausyksikön ominaisuuksia voisi hyödyntää paremmin, esimerkiksi kytkemällä mittausyksikön suoraan tietokoneeseen USB-liitännän kautta. Tällöin ohjelmaa tulisi muokata vastaanottamaan suoraan millimetriarvoja ja ohjelmisto olisi hieman kevyempi, jos muunnosta millimetreistä milliampeereiksi ei tarvitsisi laskea. Lisäksi laitteistosta tulisi fyysisesti kevyempi, kun mittausjärjestelmästä jäisivät kokonaan pois ADAM-4017-yksikkö, sen virtalähde sekä USB-adaptteri. Toisaalta ADAM-yksikön mukanaolo mahdollistaa anturin helpon vaihdettavuuden tarvittaessa.

Ohjelmistoon voisi lisätä ominaisuuksia, varsinkin tuoteselain on hyvin rajoittunut. Mittausohjelmassa ei ole mahdollista tulostaa tietoja paperille, joskin Access-ohjelmistossa sellainen mahdollisuus kuitenkin löytyy. Verkko-ominaisuuksia voisi myös parantaa, ensimmäisissä suunnitelmissa olikin vielä web-liitäntä mukana. Se kuitenkin jäi toteuttamatta ajanpuutteen vuoksi. Web-liitännän avulla mittaustuloksia

voisi seurata yrityksen intranetin avulla toimistosta koneelta, ja puuttua jonkin tietyn tuotteen valmistukseen, jos mittausarvot ovat toistuvasti liian pieniä tai suuria.

## LÄHTEET

1. Pilkingtonin Suomen kotisivut. [Verkkodokumentti] [Viitattu 2.1.2008]. Saatavissa: <http://www.pilkington.com/europe/finland/finnish/about+pilkington/default.htm>
2. Bourke, Paul, 1988. Minimum Distance between a Point and a Line [Verkkodokumentti] [Viitattu 31.1.2008]. Saatavissa: <http://local.wasp.uwa.edu.au/~pbourke/geometry/pointline/>