

Heikki Kivilahti

FOCALSPECIN MITTAPÄÄN KÄYTTÖ LABVIEW-YMPÄRIS- TÖSSÄ

FOCALSPECIN MITTAPÄÄN KÄYTTÖ LABVIEW-YMPÄRIS- TÖSSÄ

Heikki Kivilahti
Opinnäytetyö
Syksy 2016
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, langattomat laitteet

Tekijä: Heikki Kivilahti
Opinnäytetyön nimi: FocalSpecin mittapään käyttö LabVIEW-ympäristössä
Työn ohjaajat: Karri Niemelä, Timo Vainio
Työn valmistumislukukausi ja -vuosi: Syksy 2016
Sivumäärä: 29

Tämän opinnäytetyön tavoitteena oli luoda uusi esimerkkisovellus FocalSpec Oy:n optiselle mittalaitteelle käyttäen National Instrumentsin LabVIEW-ohjelmointiympäristöä. FocalSpec Oy on erikoistunut erilaisten materiaalien ja pintojen 3D -mittauksiin.

LabVIEW oli valittu uudeksi sovellusympäristöksi sen monipuolisuuden ja nopean sovelluskehityksen vuoksi. LabVIEW on suosittu ohjelmointiympäristö erilaisissa teollisuuden ja laadunvalvonnan kohteissa. FocalSpec Oy:llä oli ennestään käytössä C++- ja C#-kielillä tehtyjä sovelluksia, joten uusi ohjelmisto oli tärkeä lisä myös markkinoinnin kannalta.

Ohjelmalla oli pystyttävä asettamaan tiettyjä perusparametreja ennen mittausta-pahtumaa ja esittämään mittausdata käyttäjälle kaksiulotteisena sivuprofiilina, josta ilmenee pinnan korkeuserot.

Ohjelma toteutettiin kesän 2016 aikana ja työssä päästiin sen alussa asetettuihin tavoitteisiin. Ohjelman rakenne toteutettiin tilakoneena, joten siihen voidaan myöhemmin luoda lisää ominaisuuksia ilman suuria muutoksia.

Asiasanat: optinen 3D -mittaus, LabVIEW

ABSTRACT

Oulu University of Applied Sciences
Information technology, wireless devices

Author: Heikki Kivilahti

Title of thesis: Using FocalSpec's sensor in LabVIEW environment

Supervisors: Karri Niemelä, Timo Vainio

Term and year when the thesis was submitted: Autumn 2016

Pages: 29

The goal of this thesis was to develop a new application for FocalSpec Oy's optical sensor using National Instrument's LabVIEW software. FocalSpec is specialized in optical measurement of different surfaces and materials using 3D-scanning technology.

LabVIEW was chosen as a new software environment because of its fast and versatile abilities in software development. All previous measurement applications in FocalSpec have been made using C++ and C#-languages. LabVIEW offers a new alternative for development and marketing purposes.

In the application the user had to be able to configure some basic parameters before scanning the sample and after the measurement, the application shows 2D-profile of the target's surface. A state machine was used as a backbone to the structure so that adding more features in the future would be easy without larger changes to the code.

The application was developed during summer 2016 and the goals that were set in the beginning of the thesis were reached.

Keywords: optical 3D -measurement, LabVIEW

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
1 JOHDANTO	6
2 FOCALSPEC YRITYKSENÄ	7
2.1 Teknologia	7
2.2 Mittalaitteet	8
3 LABVIEW	9
3.1 Front Panel	10
3.2 Block Diagram	11
3.3 Connector Pane	11
4 TYÖN TAVOITTEET	12
5 TOTEUTUS	13
5.1 LabVIEW-ajuri	14
5.2 Esimerkkisovellus	16
5.3 Ohjelman toiminta	17
5.4 Lohkokaavio eli ohjelmakoodi	19
5.5 Käyttöliittymä	22
5.6 Versiohallinta	24
6 TULOKSET	25
7 YHTEENVETO	28
LÄHTEET	29

1 JOHDANTO

Opinnäytetyö tehtiin oululaisen FocalSpec Oy:n tilauksena. Työn tavoitteena oli luoda mittaussovellus FocalSpecin mittalaitteille käyttäen National Instrumentsin LabVIEW-ohjelmointiympäristöä.

LabVIEW oli valittu uudeksi ohjelmointiympäristöksi nopean sovelluskehityksen vuoksi. Se on laajasti käytössä teollisuudessa, joten samassa ympäristössä toimivan mittalaitteen lisääminen asiakkaan omiin järjestelmiin on helpompaa kuin erillisen ohjelmiston luominen tiettyihin mittauksiin. LabVIEW on myös erinomainen ohjelmisto mittalaitteiden toiminnan esittelyyn, sillä ohjelmalla voidaan helposti esitellä ominaisuuksia ja luoda nopeasti uusia sovelluksia käyttäen valmiita funktioita.

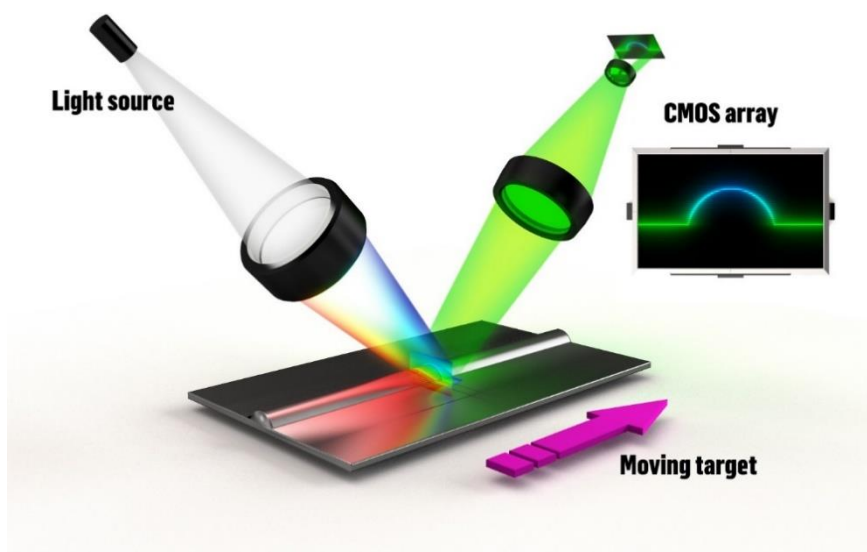
LabVIEW-ohjelman graafisessa käyttöliittymässä oli pystyttävä esittämään mitattavan kohteen pinnan kaksiulotteinen profiili (X, Z)-koordinaatistossa reaaliajassa sekä asettamaan tiettyjä perusparametreja mittalaitteelle.

2 FOCALSPEC YRITYKSENÄ

FocalSpec on oululainen mittaustekniikan yritys, joka on erikoistunut optisiin 3D-mittalaitteisiin. Focalspec on perustettu vuonna 2009, jolloin yrityksen käyttämää tekniikkaa kehitettiin mm. VTT:n VALMIT -projektissa.

2.1 Teknologia

Mittalaitteissa käytettävä teknologia perustuu väriaberraatioon. Mitattavaa kohdetta valaistetaan ja takaisin heijastunut valo mitataan. Tekniikassa käytetyn valkoisen valon eri värit eli aallonpituudet tarkentuvat eri syvyyksille ja takaisin heijastuneen valon spektri mitataan spektrografilla. Kohteen etäisyys määritellään analysoimalla pinnasta heijastuneen valon spektri. (Kuva 1.) Mittaustapa soveltuu moniin eri pintoihin kuten kiiltävään, mattaan ja läpinäkyvään. Myös eri materiaalien pintojen mittaus onnistuu samalla tekniikalla. (1, s. 4.)



KUVA 1. Mittausperiaate (2)

Kyseisellä mittaustekniikalla saavutetaan mikrometritason erottelukyky, joka on tärkeää erilaisissa teollisuuden sovelluksissa, kuten metallipintojen karheusmittauksissa.

Itse signaalinkäsittely tapahtuu mittalaitteessa. Tämä vähentää taakkaa esimerkiksi mittausdataa esittävältä ohjelmalta ja mahdollistaa helpomman integraation muihin mittausjärjestelmiin tai linjastoihin.

Mittausdatasta saadaan myös laskettua pinnan kiiltoaste käyttämällä signaalin intensiteettitietoa. Pääasiallisena ohjelmistona toimivat C++- ja C#- sovellukset, joilla mittausdataa esitetään käyttäjälle PC:llä.

2.2 Mittalaitteet

FocalSpec valmistaa erilaisia mittapäitä eri tarkkuuksia vaativiin kohteisiin. Mittalaitteilla voidaan mitata pintoja mm. elektroniikasta, metalleista, muoveista ja kaapeleista. Mittalaite mittaa 2048 3D-pistettä 4,5 mm – 16,4 mm matkalta malista riippuen ja ohjelma esittää mittaustuloksen PC:llä käyttäjälle.

3 LABVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) on amerikkalaisen National Instrumentsin kehittämä ja ylläpitämä graafinen ohjelmointiympäristö. LabVIEW on suosiossa teollisuusautomaatiossa ja erilaisissa datan keräykseen ja käsittelyyn liittyvissä sovelluskohteissa.

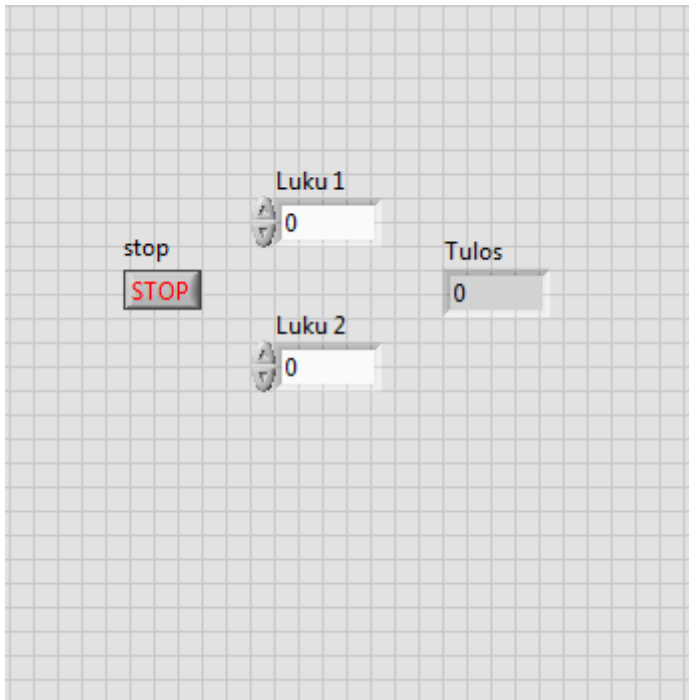
National Instruments käyttää LabVIEW'n ohjelmointiin graafista G-kieltä, joka on dataflow-ohjelmointia. Dataflow-ohjelmointi perustuu tuloihin ja lähtöihin, jotka yhdistävät operaatioita. Operaatiot käynnistyvät vasta, kun kaikki tulot ovat toisia. Graafinen G-kieli mahdollistaa helpon lähestymistavan ohjelmointiin käyttämällä valmiita virtuaalisia mittalaitteita ja ominaisuuksia, jotka voivat olla käyttäjälle ennestään tuttuja oikeista mittalaitteista. LabVIEW'n G-kieli on National Instrumentsin omistama ja sitä ei valvota tai standardisoida minkään järjestön puolesta kuten esimerkiksi C-kieltä. G-kieli yksinkertaistaa ohjelmointia, mutta monimutkaisten ohjelmien luominen on haastavaa. (3.)

LabVIEW'n ohjelmia kutsutaan VI:ksi, Virtual Instrument. Jokainen VI koostuu kolmesta osasta, front panel, block diagram ja connector pane. Front panel eli etupaneeli toimii ohjelman graafisena käyttöliittymänä, jonka kautta käytetään ohjelmaa ja myös lisätään erilaisia indikaattoreita, jotka toimivat ulostuloina block diagramissa eli lohkokaaviossa tapahtuville operaatioille. VI voi toimia itsenäisenä ohjelmana front panelin toimiessa käyttöliittymänä tai VI voidaan sijoittaa osaksi toisen VI:n block diagramia, jolloin se toimii aliohjelman tavoin. Tämä mahdollistaa yksittäisen VI:n testaamisen ennen isomman kokonaisuuden rakentamista. (3.)

3.1 Front Panel

Front Panelia eli etupaneelia muokkaamalla luodaan VI:n graafinen käyttöliittymä. Käyttöliittymässä voi olla painikkeita ja säätimiä, kuten oikeassa mittalaitteessa. Myös mitattu tai kerätty data esitetään etupaneelissa ohjelman käyttäjälle ymmärrettävässä muodossa, esim. taulukoina, grafiikkana, lukuarvoina tai muina ilmaisuina.

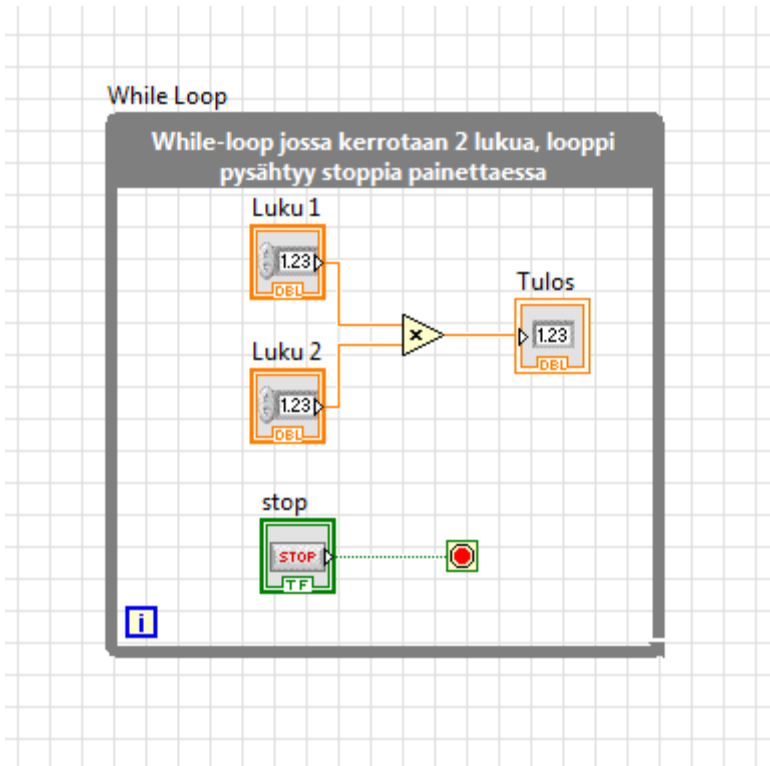
Esimerkkinä kuvassa 2 on yksinkertainen sovellus, jossa käyttäjä valitsee kaksi lukua kertolaskua varten. Aina kun käyttäjä muuttaa lukuja, ohjelma päivittää tuloksen. Ohjelma sammuu STOP-painikkeesta.



KUVA 2. Etupaneeli kertolaskulle

3.2 Block Diagram

Kuvassa 3 näkyvässä block diagrammissa, eli lohkokaaviossa luodaan ohjelman lähdekoodi ja suuri osa ohjelman toiminnallisuudesta. Ohjelma toimii nimensä mukaisesti lohkokaaavion tavoin. Koodin eri operaatiot yhdistetään toisiinsa johdotuksilla, eikä seuraava toiminto käynnisty, ennen kuin kaikki siinä olevat tulot ovat valmiita.



KUVA 3. Kertolaskun lohkokaavio

3.3 Connector Pane

Connector Panea tarvitaan, kun VI:tä halutaan käyttää aliohjelmana eli subVI:nä. Connector Pane määrittelee VI:n tulot ja lähdöt, joita voidaan yhdistää VI:hin, jotta sitä voidaan käyttää subVI:nä.

4 TYÖN TAVOITTEET

Opinnäytetyön tavoitteena oli luoda uusi mittaussovellus, jolla voitaisiin esitellä FocalSpecin mittapäätä LabVIEW-ohjelmointiympäristössä käyttämällä FocalSpecin C++- API:n (Application programming interface), eli ohjelmointirajapinnan avulla luotua LabVIEW-ajuria. Käyttöliittymästä tehtiin samankaltainen kuin nyt esimerkkinä toimivasta mittaussovelluksesta. LabVIEW:ssa on myös helppo myöhemmin muokata käyttöliittymää, jos sille tulee tarvetta.

Ohjelmalla oli saatava selkeästi esitettyä reaaliaikaista mittausdataa (X, Z)-koordinaatistossa 2D-profiilina sekä pystyttävä asettamaan perusparametreja mittausta varten ja tutkimaan mittaustuloksia.

Sovellusesimerkiksi valittiin pinnan korkeuseron mittaaminen (englanniksi step height measurement). Tämä tilanne on yleinen optisissa mittauksissa ja soveltuu siksi hyvin esimerkikse LabVIEW-sovellukseen. Sovelluksen oli pystyttävä vertaamaan mitattavan kohteen eri tasoja ja esitettävä tasojen välinen korkeusero käyttäjälle mikrometreinä.

5 TOTEUTUS

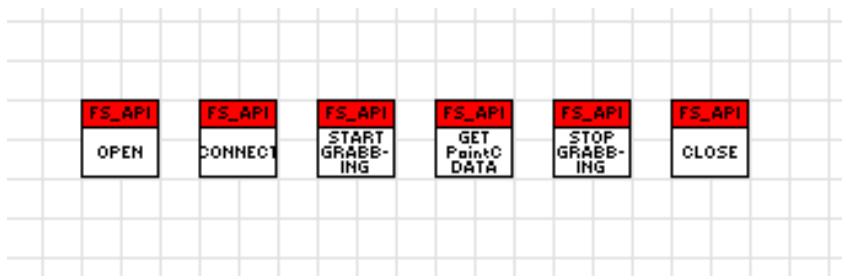
Työ aloitettiin tutustumalla FocalSpecin C++- API:iin ja sen funktioihin, jotta saataisiin käsitys LabVIEW-ajurin toiminnasta, vaikka LabVIEW'ta käyttäessä ei juuri tarvittu C++-kieltä. Ajurin toteutus oli tilattu ulkopuoliselta yritykseltä, ja tässä työssä on keskitytty vain käyttämään ajuria ja luomaan sen avulla LabVIEW-sovellus. Ennen työn alkamista olin mukana palaverissa, joissa käsiteltiin LabVIEW-ajurin hankintaa. Näin sain hyvän kuvan tulevasta työstä ja sen vaatimuksista.

Ohjelmaa luodessa C++-kielellä API:n toiminta seuraa tiettyä järjestystä. Sama järjestys pätee myös LabVIEW'ssa.

Eri vaihteet ovat yksinkertaistettuna seuraavat (kuva 4):

1. Avataan API
2. Initialisoidaan kamera
3. Aloitetaan datan vastaanotto
4. Tehdään datalle jotain
5. Datat vastaanottaminen lopetetaan
6. Suljetaan API.

LabVIEW ajurin avulla API:n funktiot ovat Block Diagrammissa aseteltavia nodeja ja toimivat samalla tavalla kuten C++:lla ohjelmoitaessa.



KUVA 4. Vaiheet 1-6, vasemmalta oikealle LabVIEW-nodeina

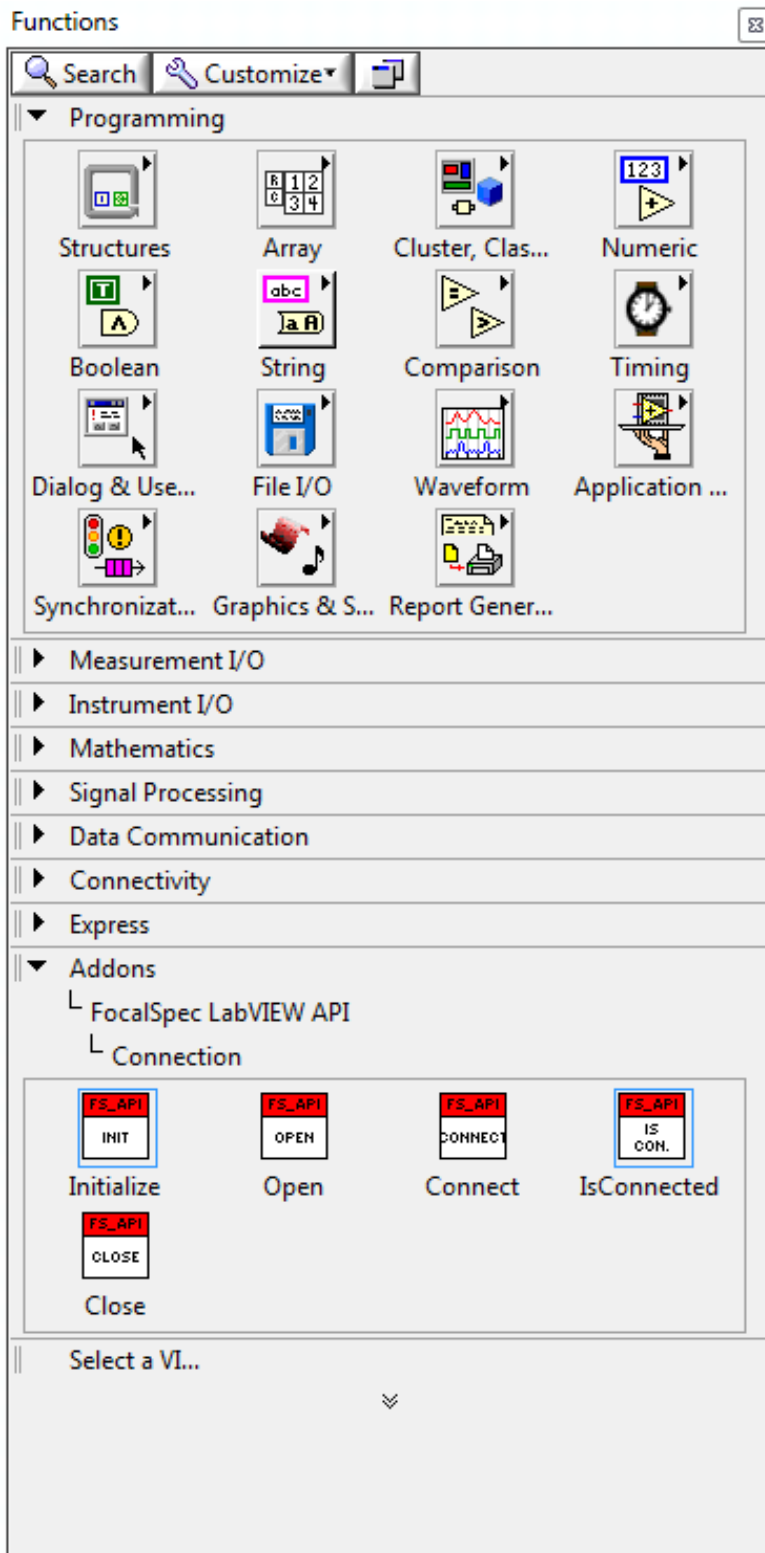
Ohjelmaa tehtäessä oli kuitenkin lisättävä muitakin toimintoja, jotta voidaan tarkemmin määritellä datan käsittelyä ja mittauksessa olevia parametreja. Työssä tärkeimmät parametrit olivat valotusaika, skannauspulssin taajuus ja mittalaitteen kalibrointitiedostot, jotta mittaustulos voidaan esittää mikrometreinä.

Alussa tutustuttiin myös nyt käytössä olevaan C++- ja C#-kielillä toteutettuun mittaussovellusesimerkkiin, jotta mm. käyttöliittymästä tulisi helposti käytettävä. Tämä helpotti myöhemmin toteuttavan uuden sovelluksen luontia.

5.1 LabVIEW-ajuri

Ennen LabVIEW-ohjelman luomista oli asennettava LabVIEW-ajuri, jotta mittalaitte saadaan LabVIEW-yhteensopivaksi. Ajuri mahdollistaa FocalSpecin API:n ja DLL-tiedostojen käyttämisen LabVIEW'illa. Ilman ajuria API:ssa olevia C++-funktioita ei voitaisi käyttää LabVIEW'illa mittalaitteen ohjaamiseen tai datan käsittelyyn. Ajuri toimitettiin FocalSpecille niin sanottuna VI-pakettina, jonka asentamiseen käytettiin JKI VI Package Manager -ohjelmaa.

Ajurin asentaminen tapahtuu automaattisesti JKI VI Package Managerin avulla normaalin Windows-sovelluksen tapaan. Managerissa valitaan LabVIEW-versio, johon ajuri halutaan asentaa ja seurataan ohjelman ikkunoita. Kyseisen LabVIEW'n version on oltava käynnissä ajuria asennettaessa. Asennuksen jälkeen API:ssa olevat C++-funktiot ovat käytettävissä LabVIEW'n lohkokaaviossa muiden funktioiden tavoin (kuva 5). Funktiot oli luotu yksittäisiksi subVI:ksi, joita voidaan käyttää nodejen tavoin.

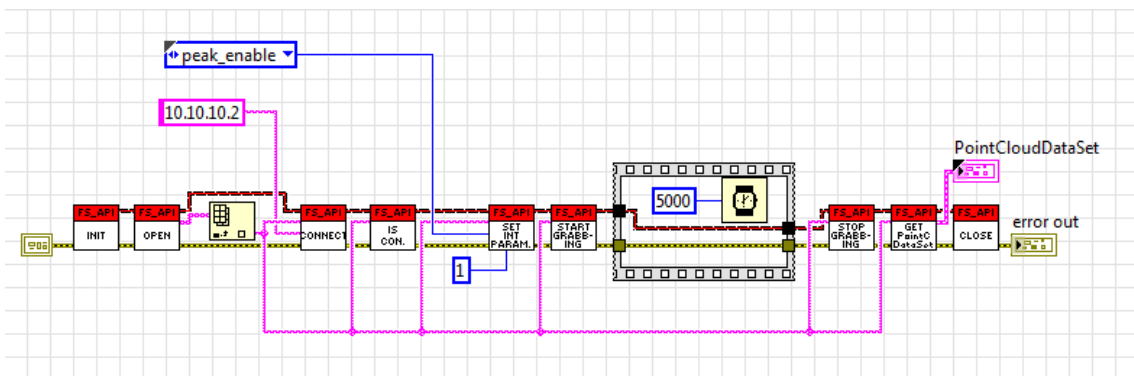


KUVA 5. FocalSpecin LabVIEW-ajurin funktioita LabVIEW:ssa

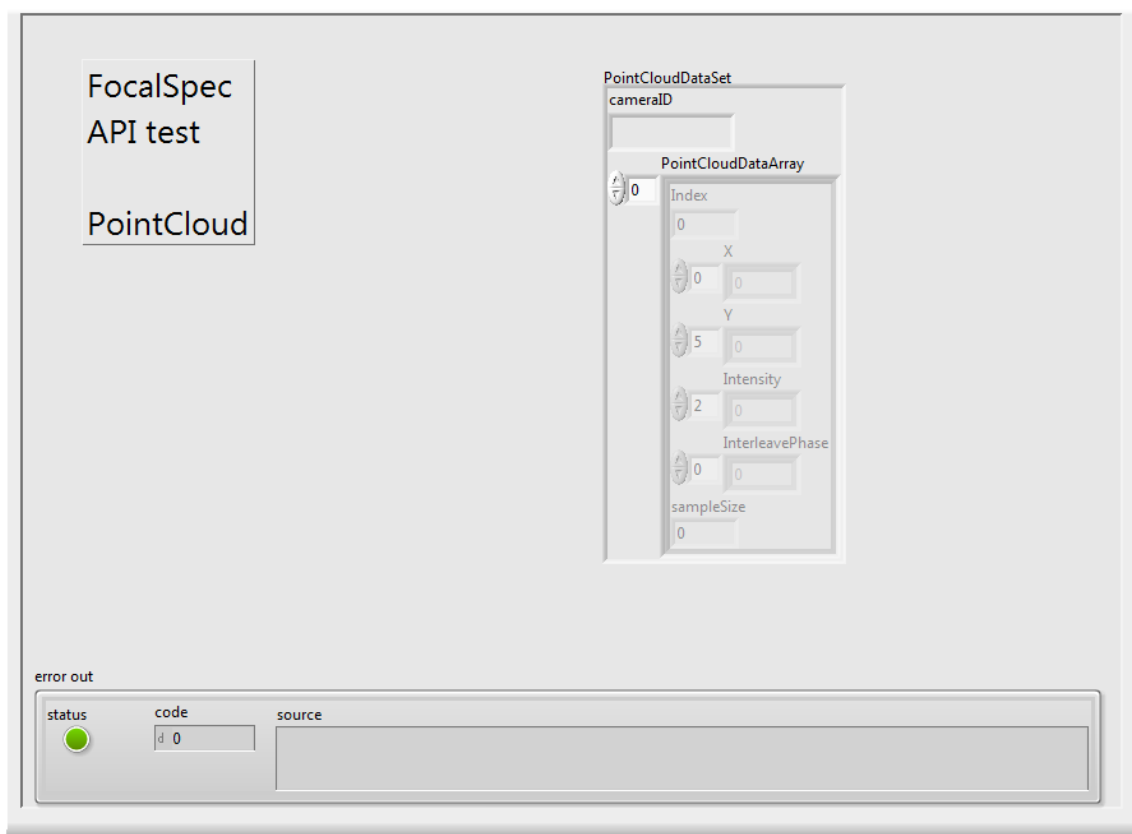
5.2 Esimerkkisovellus

Ajurin asentamisen jälkeen voitiin aloittaa itse LabVIEW'n käyttäminen. Alussa tutustuttiin nyt LabVIEW:ssa toimivan API:n funktioihin ja esimerkkisovellukseen, joka oli toimitettu ajurin mukana. Esimerkissä muodostettiin yhteys mittalaitteeseen ja paluuarvoina saatiin tietoa mittalaitteesta ja mittausdataa lukuarvoina. Tämän pohjalta aloitettiin luomaan laajempaa ja käytännöllisempää sovellusta.

Esimerkkisovelluksen (kuva 6) rakenne on aikaisemmin mainitun kaltainen, muutamilla muutoksilla. Ohjelmalle kerrotaan mittalaitteen IP-osoite yhteyden muodostamista varten ja se määritetään vastaanottamaan peak-data mittalaitteelta käyttämällä Set Int Parameter -nodea, jonka avulla määritellään integertyyppisiä parametreja. Sen jälkeen aloitetaan datan vastaanottaminen Start grabbing -nodella, joka kestää 5000 millisekuntia. Tämän jälkeen datan vastaanottaminen lopetetaan Stop grabbing -nodella, ja se esitetään käyttäjälle front panelin taulukossa (kuva 7).



KUVA 6. Esimerkkisovelluksen lohkokaavio



KUVA 7. Esimerkkisovelluksen etupaneeli eli käyttöliittymä

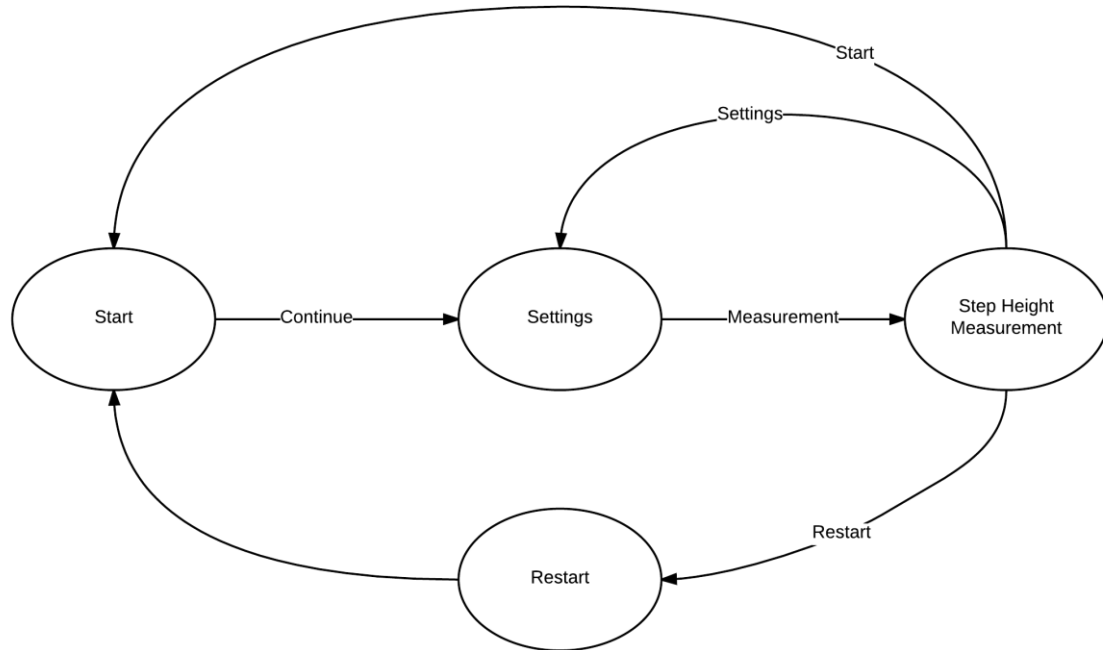
5.3 Ohjelman toiminta

Työn alussa oli tutkittava eri mahdollisuuksia datan esittämiseen ja reaaliaikaisuuteen, erityisesti pitäisikö jokin ominaisuus luoda vai käyttää LabVIEW'n valmiita funktioita. Myös käyttöliittymään kiinnitettiin huomiota jo tässä vaiheessa, koska suurin osa siitä tuli olemaan (X, Z) -koordinaatistoa. Työssä päädyttiin käyttämään valmista funktiota koordinaatiston luomiseen työn nopeuttamiseksi.

Tärkeimmässä osassa ohjelmaa oli reaaliaikainen datan esittäminen ja asetusten muuttaminen, myöhemmässä vaiheessa myös oikeiden mittayksiköiden, mikrometrien, käyttäminen pikseleiden sijaan mittaustuloksessa.

Ohjelman perusrakenteeksi päätettiin luoda tilakone (kuva 8) rakenteen selkeyden ja mahdollisten lisäominaisuuksien luomisen vuoksi (4). Tilakoneeseen on yksinkertaista lisätä uusia ominaisuuksia lisäämällä ominaisuutta varten uusi tila. Tilakone toimii neljässä eri tilassa ohjelman eri vaiheissa. Ohjelmassa käyttäjällä voi olla näkyvissä kolme eri tilaa riippuen siitä, missä vaiheessa ohjelman

käyttämistä ollaan. Käyttöliittymä seuraa tilakoneen tiloja vaihtamalla välilehteä sitä mukaa, kun ohjelman tila vaihtuu.



KUVA 8. Tilakaavio ohjelman toiminnasta

Ohjelman käynnistyessä ollaan aloitussivulla Start-tilassa, jossa käyttäjän on määritettävä tietokoneen ja kameran yhteysasetukset yhteyden muodostamista varten. Start-tilasta voidaan edetä vain Settings-tilaan tai sammuttaa ohjelma kokonaan. Settings-tilassa käyttäjä voi edetä Step height measurement -tilaan, jossa itse mittausdata esittäminen tapahtuu. Mittaustapahtuman jälkeen voidaan joko sammuttaa ohjelma, käynnistää se uudelleen tai palata muokkaamaan asetuksia Settings-tilaan. Restart-tilalle ei ole omaa näkymää käyttöliittymässä, vaan se käynnistää tilakoneen uudelleen Start-tilaan ja siirtää käyttäjän oikealle välilehdelle automaattisesti.

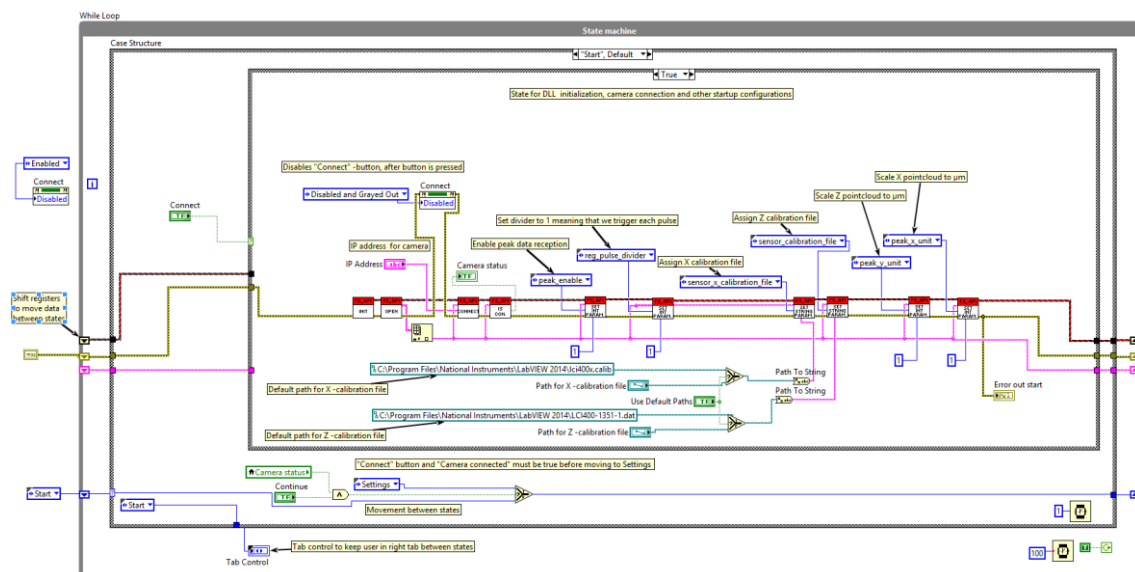
Eri tilat on jaettu eri välilehdille ja siirtyminen eri välilehdillä ja tiloissa tapahtuu samalla, kun käyttäjä vaihtaa näkymää. Tällä estetään se, että itse ohjelmakoodi ja käyttöliittymä olisivat eri tiloissa. Tilakone pysyy samassa tilassa niin

kauan, kun käyttäjä päättää edetä käyttöliittymässä. Käyttäjä ei myöskään voi vaihtaa tilaa ilman, että käyttöliittymän näkymä ei vaihtuisi mukana.

5.4 Lohkokaavio eli ohjelmakoodi

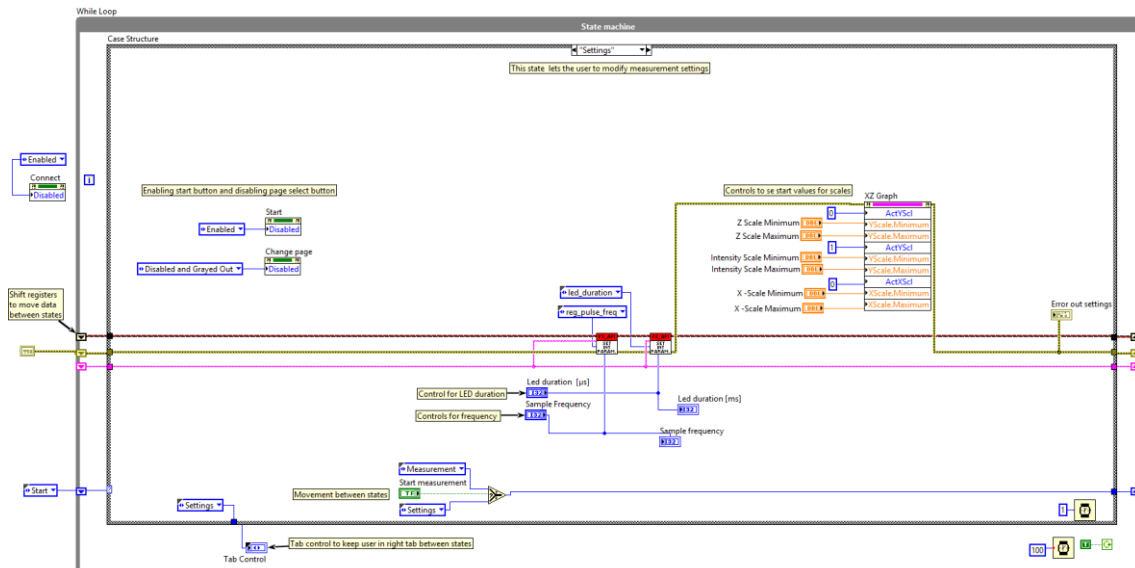
Tilakoneen eri tilat rakennettiin while-silmukan sisään case-rakenteen avulla. Ohjelman toiminta alkaa samalla logiikalla kuin esimerkissä ja aiemmissa FocalSpecin sovelluksissa, eli yhteyden avaamisella ohjelmasta mittalaitteeseen. Datan siirtäminen tilakoneen tilojen välillä toteutettiin siirtorekistereillä, jolloin se on täysin automaattista tilojen välillä.

Kuvassa 9 on lohkokaaavion näkymä Start-tilasta. Case-rakenteen ensimmäisessä tilassa käydään läpi aiemmin mainitut vaiheet yhteyden avaamisesta, oikean tyyppisen datan vastaanottoon ja kalibroitiedostojen asettamiseen. Yhteyden muodostaminen aloitetaan INIT-, OPEN- ja CONNECT-nodeilla ottamalla käyttöön DLL-tiedostot ja asettamalla IP-osoite mittalaitteelle. SET INT PARAM -nodeilla määritellään int-tyyppisten muuttujien avulla ohjelma vastaanottamaan oikeantyyppistä dataa ja rekisteröimään kaikki mittauspulssit. Tämän jälkeen määritellään kalibroitiedostojen sijainnit string-tyyppisinä muuttujina SET STRING PARAM -nodeilla ja asetetaan mittayksiköt mikrometreiksi SET INT PARAM -nodeilla.



KUVA 9. Start- eli aloitustilan lohkokaaavio

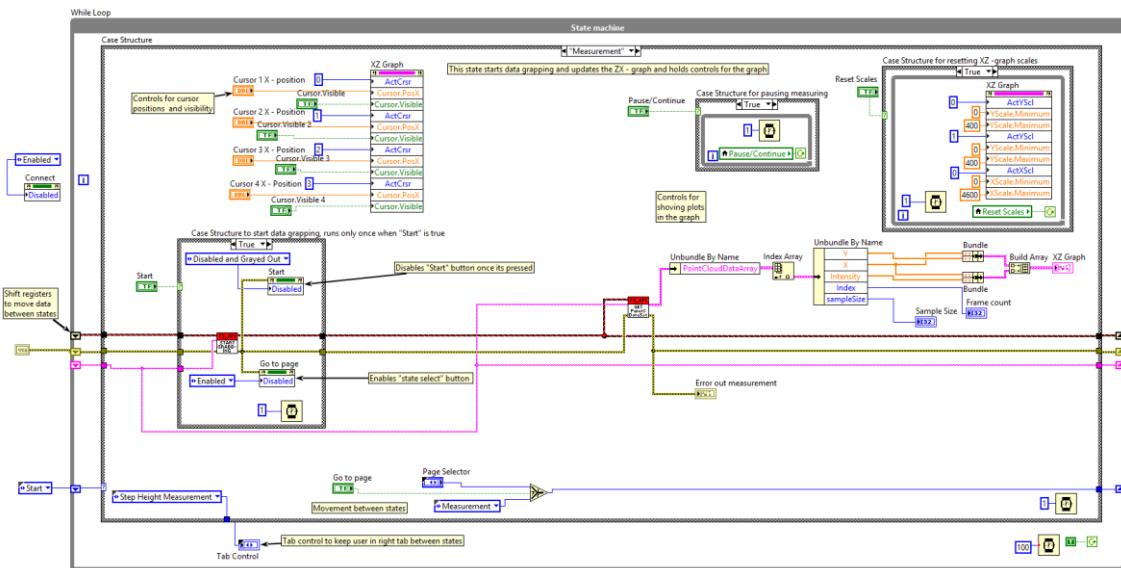
Kuvassa 10 on lohkokaavion näkymä Settings-tilasta, jossa mahdollistetaan valotusajan ja mittaustajauuden muokkaaminen SET INT PARAM -nodeilla. Myös (X, Z) -graafin skaalojen asettaminen on toteutettu kyseisessä tilassa.



KUVA 10. Settings- eli asetustilan lohkokaavio

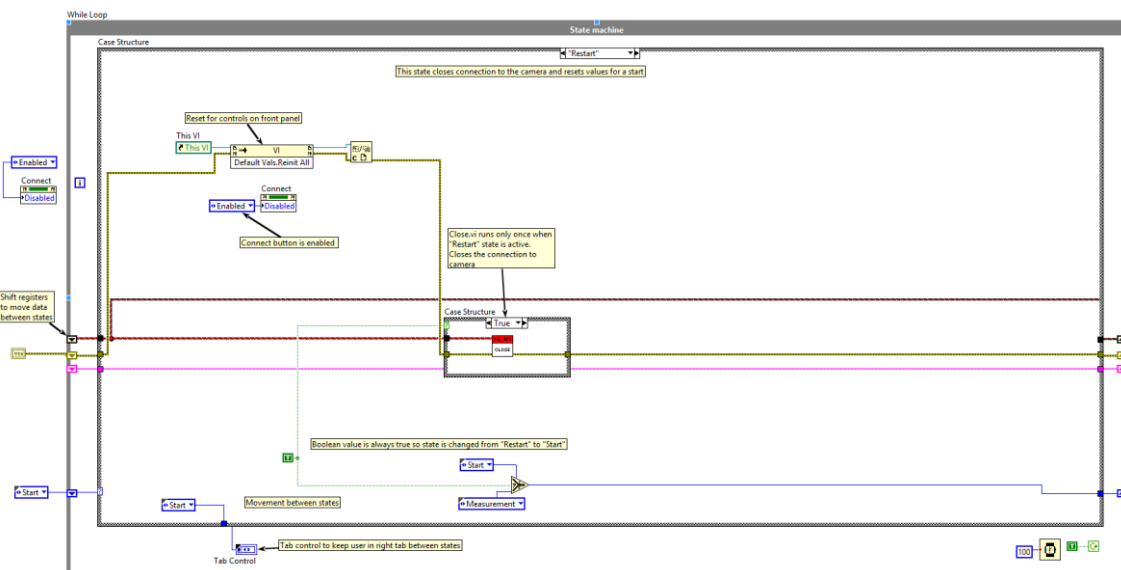
Kuvassa 11 on Measurement-tila, jossa itse mittausdatan vastaanottaminen ja esittäminen tapahtuvat. Tässä tilassa on myös kursorien siirtämisen mahdollistavat ominaisuudet. Datnan vastaanottaminen aloitetaan START GRABBING -nodella, joka on sijoitettu oman case-rakenteen sisälle, joka käynnistyy vain, kun Front Panelissa olevaa Start-näppäintä on painettu. GET PointCloudDataSet -nodella vastaanotetaan pistepilvi-tyyppistä mittausdataa, josta erotellaan jokaisen mittauspisteen X- ja Z-koordinaatit. Datasta saadaan selville myös mitattavan kohteen intensiteetti, näytteen koko ja se kuinka mones näyte on kyseessä. Nämä tiedot esitetään käyttäjälle käyttöliittymässä.

Tässä tilassa on myös kontrollit mittausdatan esittämisen pysäyttämiseksi ja jatkamiseksi sekä (X, Z) -graafin skaalojen asettamiseksi oletusarvoihin.



KUVA 11. Measurement- eli mittauksen tilan lohko-kaavio

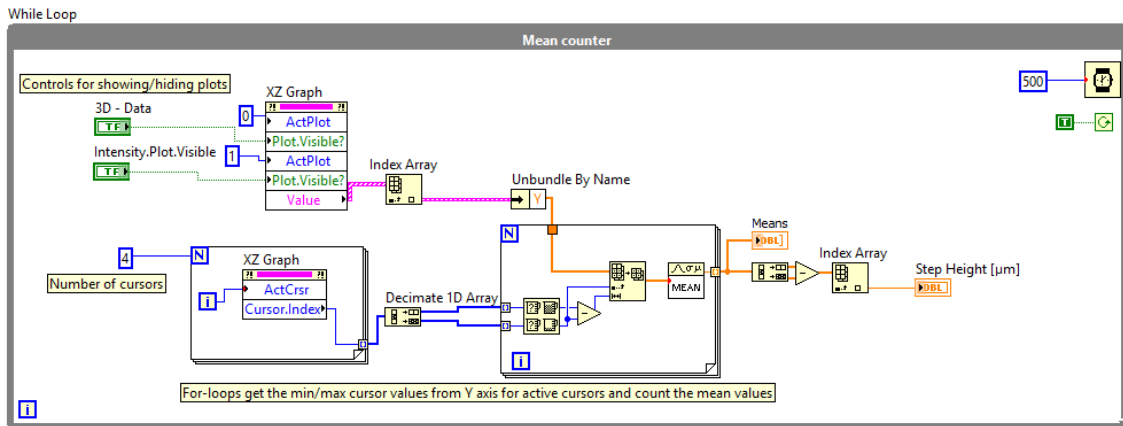
Kuvassa 12. on Reset-tilan case-rakenne, jossa suljetaan yhteys mittalaitteelle CLOSE-nodella ja alustetaan kaikki asetukset ennen palaamista Start-tilaan.



KUVA 12. Reset- eli uudelleenkäynnistystilan lohko-kaavio

Kuvassa 13. on erillinen while-silmukka kursorilla rajattavien mittauspisteiden keskiarvojen laskemiselle ja vertailulle. Erillinen silmukka mahdollistaa tilanteen,

jossa mittausdatan vastaanottaminen pysäytetään, jos tiettyä kohtaa datasta halutaan tutkia ilman, että keskiarvoistaminen pysähtyy.

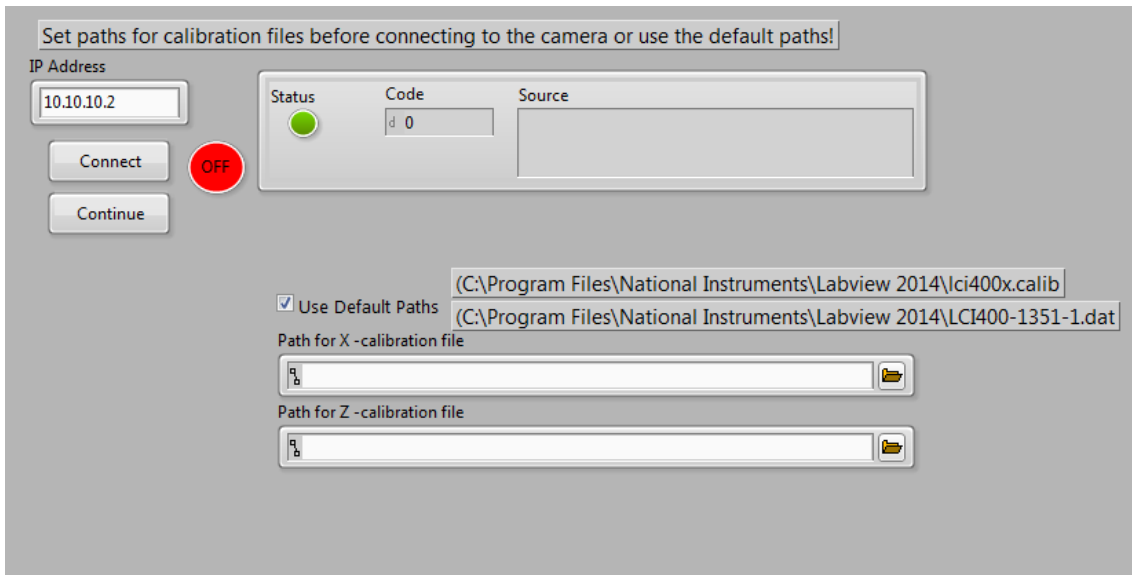


KUVA 13. Lohkokaavio erilliselle while-silmukalle mittausalueiden keskiarvoistamiselle ja vertailulle. (5)

5.5 Käyttöliittymä

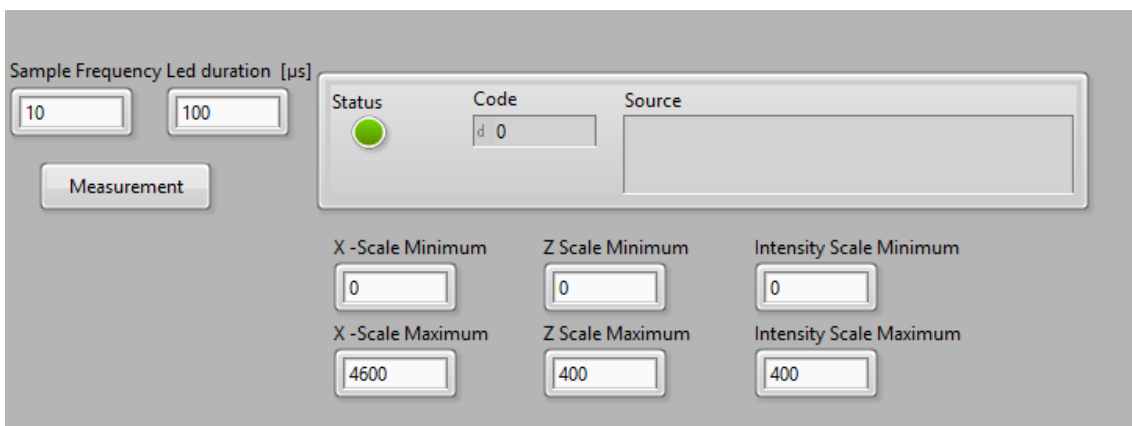
Käyttöliittymässä haettiin yksinkertaista näkymää jokaiselle välilehdelle, jotta ohjelma sopisi demokäyttöön. Jokaisella välilehdellä on oma funktio ja käyttäjä näkee selkeästi sen tarkoituksen.

Aloituvälilehdellä (kuva 14) Start käyttäjä asettaa kameran IP-osoitteen johon yhdistetään ja asettaa mittauksia varten kalibrointitiedostojen hakemistopolut. Kalibrointitiedostot mahdollistavat mittauksien esittämisen mikrometreinä, jos polkuja ei aseteta ohjelma käyttää oletuspolkuja, jotka ovat osana ohjelma-koodia. Käyttäjällä on valittavana kaksi painiketta, Connect ja Continue. Kun IP-osoite on asetettu oikein ja yhteys mittalaitteeseen on muodostettu painamalla Connect-painiketta, vaihtuu punainen valo vihreäksi ja käyttäjä voi painaa Continue-painiketta siirtyäkseen asetusvälilehdelle.



KUVA 14. Start eli aloitusnäkö käyttöliittymässä

Asetusvälilehdellä (kuva 15.), Settings, käyttäjällä on valittavana kuvantamistajuus, sensorin valotusaika ja mittaustulosta esittävän koordinaatiston akseleiden maksimi- ja minimiarvot. Kuvantamistajuuden yksikkönä toimii hertsi, eli kuinka monta kertaa sekunnissa otetaan kuva, valotusajan yksikkönä on millisekunti ja koordinaatiston akseleiden mittayksikkönä ovat mikrometrit. Käyttäjä voi muuttaa X-, Z- ja intensiteetti akseleiden maksimi- ja minimiarvoja. Koordinaatistoa voi muuttaa myös mittauksen aikana, mutta sen oletusarvot asetetaan Settings-välilehdellä.



KUVA 15. Settings eli asetusvälilehti

Mittausvälilehdellä käyttäjälle esitetään mitattavan kohteen kaksiulotteinen profiili reaaliajassa. Mitattavan kohteen korkeusero lasketaan keskiarvoistamalla mittausdataa tietyltä matkalta ja vertaamalla tulosta toiseen samanlaiseen näytteeseen. Keskiarvoistettavat näytteet valitaan liikuteltavien kursorien avulla. Ohjelma laskee automaattisesti kursorien välissä olevien mittauspisteiden korkeuden koordinaatiston nolatasosta ja vertaa tulosta toisen kursoriryhmän samalla tavalla mitattuun tulokseen. Ohjelma esittää kyseisten tulosten erotuksen käyttäjälle mikrometreinä.

5.6 Versiohallinta

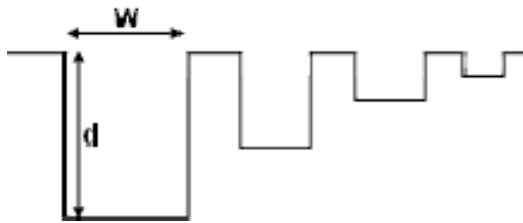
Ohjelmoinnissa on tärkeää käyttää versiohallintaa ohjelman kehitysvaiheessa ja sitä muokatessa, jotta kaikki muutokset tulee dokumentoitua. Myös koodin vanhempiin versioihin on helpompi palata muutosten jälkeen, jos koodin toiminta muuttuu rajusti tai se toimi muutosten jälkeen. Versiohallintaan käytettiin Gitiä ja SourceTree -ohjelmaa. Git on hyvin nopea ohjelmisto ja SourceTreen puurakenne luo selkeyttä ohjelman eri versioiden hahmottamiseen. Versiohallinnan käyttäminen oli minulle uutta, mutta lopulta aika yksinkertaista.

6 TULOKSET

Ohjelman toimintaa testatessa esimerkkimittauksessa kohteena käytettiin Rubert & Co Ltd:n 513X -referenssikappaletta, joka on ISO 5436-1- standardin mukainen pinnankarheusmittauksissa käytettävä kalibrointikappale. Kappaleen sivuprofiili on kuvan 16 mukainen. Kappaleen urien mitoiksi valmistaja ilmoittaa seuraavat mitat, lueteltuna vasemmalta oikealle (6):

$d = 1000 \mu\text{m}, 500 \mu\text{m}, 200 \mu\text{m}$ ja $30 \mu\text{m}$

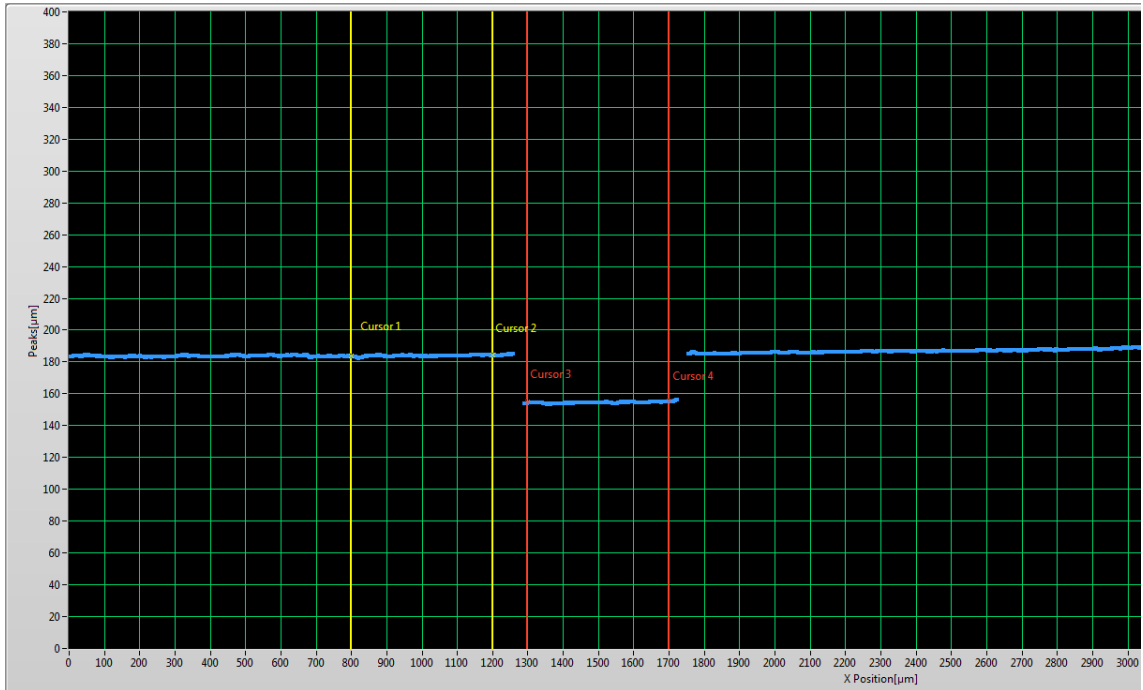
$w = 3 \text{ mm}, 2 \text{ mm}, 2 \text{ mm}$ ja $0,5 \text{ mm}$.



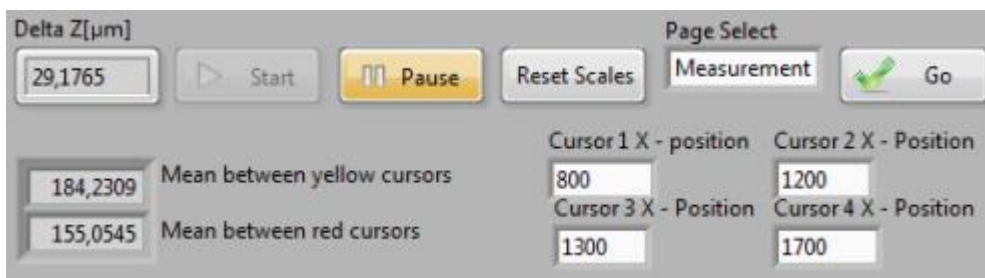
KUVA 16. Referenssikappaleen sivuprofiili (6)

Testimittauksissa käytössä oli FocalSpecin LCI 400 -mittalaite, jonka mittausprofiilin pituus on 4,50 mm (7).

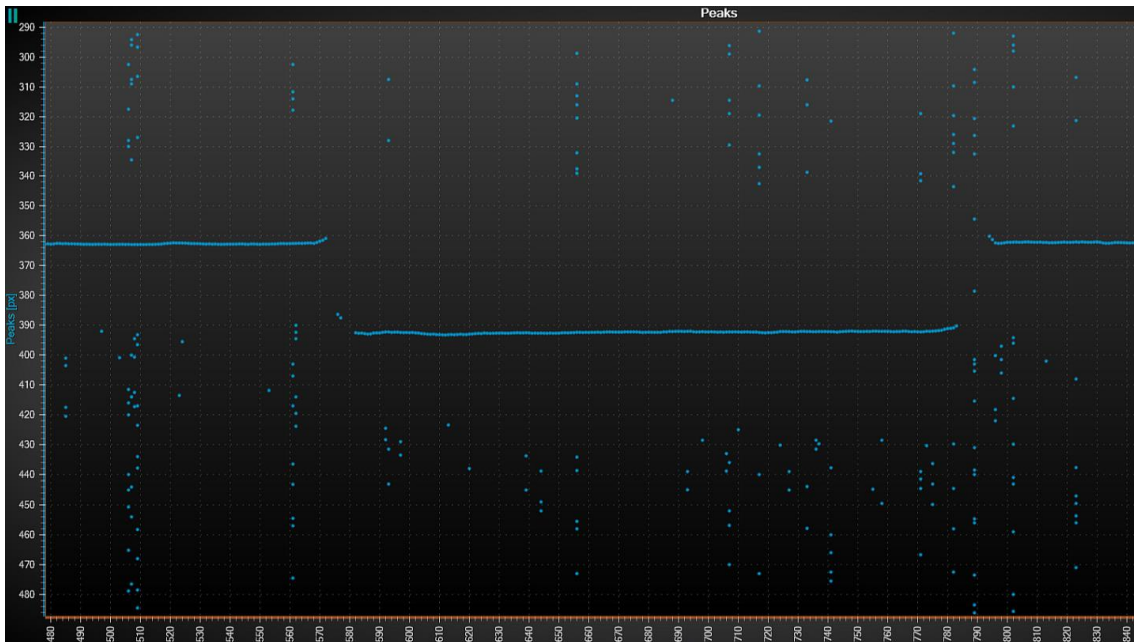
Kuvassa 17. on tarkennettu näkymä mittaustuloksesta LabVIEW'n mittauskäyttöliittymässä. Kuvassa 18. on ohjelmalla lasketut mittaustulokset, jotka ovat hyvin lähellä ilmoitettua arvoa. Uran syvyydeksi saatiin $29,2 \mu\text{m}$. Kuvassa 18. on näkyvillä myös kursoreilla rajattujen mittauspisteiden keskiarvot, joita vertaamalla mittaustulos laskettiin. Saatua arvoa verrattiin myös FocalSpecin jo käytössä olevalla esimerkkiohjelmalla mitattuun arvoon, joka vastasi saatuja tuloksia (kuva 19).



KUVA 17. Tarkennettu näkymä mittaustuloksesta LabVIEW-ohjelmalla

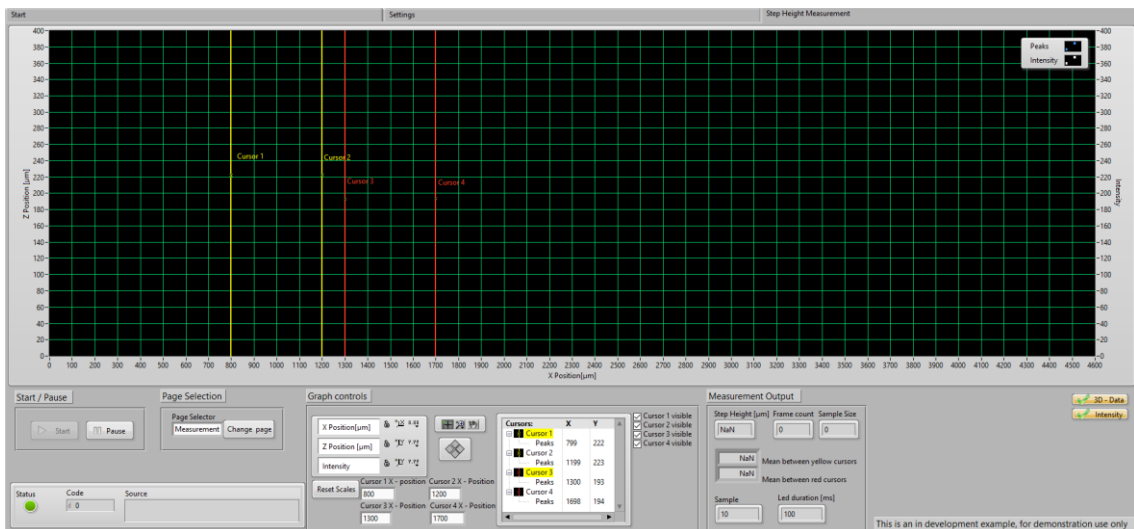


KUVA 18. Mittaustulokset



KUVA 19. Tarkennettu näkymä mittaustuloksesta Sensor Client Example -ohjelmalla

Mittausten suorittamisen jälkeen käyttöliittymän rakennetta järjesteltiin selkeämmäksi (kuva 20). Itse mittausdataa esittävä graafi pysyi samanlaisena, mutta käytön helpottamiseksi eri asetukset ja kontrollit sijoitettiin omille kohdilleen.



KUVA 20. Lopullinen käyttöliittymä mittaukselle

7 YHTEENVETO

Työn tarkoituksena oli luoda toimiva mittalaitesovellus FocalSpec Oy:n mittalaitteelle käyttäen National Instrumentsin LabVIEW-ohjelmointiympäristöä. FocalSpecin mittalaitteita ei ollut käytetty LabVIEW-ympäristössä ennen tätä opinnäytetyötä, joten työssä oli haasteita myös siltä kannalta.

Työn alkuvaiheessa määriteltiin tulevan ohjelman ominaisuudet ja hahmoteltiin käyttöliittymän rakenne. Tarvittavat ominaisuudet rajattiin tarkasti, jotta ne voitiin toteuttaa vaaditussa ajassa. Käyttöliittymän suunnittelussa haettiin yksinkertaista näkymää, jotta sen käyttö olisi helposti opittavissa. Myös LabVIEW'n käyttö oli alussa hieman kankeaa, koska aiemmasta käyttökerrasta oli kulunut aikaa. LabVIEW'n oma Help-toiminto ja National Instrumentsin keskustelufoorumit toimivat kuitenkin erinomaisena apuna ongelmatilanteissa. Itse mittalaitteen kanssa esiin tulleista ongelmista oli helppo päästä eroon työkavereiden opastuksella.

Työssä toteutettiin vaaditut ominaisuudet, jotka olivat lopulta hyvin suoraviivaisesti luotavissa LabVIEW'n funktioiden ja FocalSpecin LabVIEW-ajurin avulla. Opin työssä paljon uutta ohjelmistokehityksestä LabVIEW'lla ja myös optisesta mittauksesta tietotekniikan haarana.

Ohjelman toteutus eteni työn alussa sovitussa aikataulussa kesän 2016 aikana ja ohjelman jatkokehitysmahdollisuudet ovat hyvin laajat. LabVIEW'lla toteutettu ohjelma on mahdollista sulauttaa osaksi jo olemassa olevaa ympäristöä ja näin nopeuttaa mittalaitteen käyttöönottoa huomattavasti.

Opinnäytetyön lopputuloksena yritykselle jäi käyttöön uusi esimerkkisovellus, jota voidaan käyttää mittalaitteiden esittelyssä.

LÄHTEET

1. Niemelä, Karri 2010. VALMIT WP1. Kromaattisen aberraation tekniikka pinnan mittauksessa. Tutkimusraportti VTT-R-03201-10. Raportti 1. Oulu: VTT.
2. Technology. 2016. FocalSpec. Saatavissa: <http://www.focalspec.com/technology/>. Hakupäivä 25.11.2016.
3. LabVIEW. 2016. Wikipedia. Saatavissa <https://en.wikipedia.org/wiki/LabVIEW>. Hakupäivä 27.11.2016.
4. Application Design Patterns: State Machines. 2011. White Paper. National Instruments. Saatavissa <http://www.ni.com/white-paper/3024/en/>. Hakupäivä 25.11.2016.
5. Finding the average between 2 cursors. 2008. Keskusteluketju. National Instruments Community, Discussion Forums. Saatavissa: <http://forums.ni.com/t5/LabVIEW/finding-the-average-between-2-cursors/td-p/806010>. Hakupäivä 25.11.2016.
6. The Home Of Surface Measurement. 2016. Rubert & Co Ltd. Saatavissa: <http://www.rubert.co.uk/reference-specimens/>. Hakupäivä 26.11.2016.
7. Line Confocal Sensors for Industrial Applications. Esite. FocalSpec. Saatavissa: http://www.focalspec.com/wp-content/uploads/2016/08/LineConfocalSensors_A4_24082016_for_web.pdf. Hakupäivä 25.11.2016.