

Satu Oikarinen

KONDENSAATTORIN LOPPUTARKASTUSLAITTEEN
TIEDONSIIRTO- JA RAPORTOINTIOHJELMA

Insinööryö
Kajaanin ammattikorkeakoulu
Tekniikan ja liikenteen ala
Tietotekniikan koulutusohjelma
Kevät 2005



Osasto Tekniikka	Koulutusohjelma Tietotekniikka
Tekijä(t) Satu Oikarinen	
Työn nimi Kondensaattorin lopputarkastuslaitteen tiedonsiirto- ja raportointiohjelma	
Vaihtoehtoiset ammattiopinnot Elektroniikan testaussuunnittelu	Ohjaaja(t) Pentti Romppainen Esa Matero
Aika Maaliskuu 2005	Sivumäärä 46
Tiivistelmä <p>Insinööriyön tavoitteena oli suunnitella ja toteuttaa Evox Rifa Oy:lle tiedonsiirto- ja raportointiohjelma kondensaattorien lopputarkastukseen. Mittaustulosten lukeminen testaus/lajittelukoneen mittasillalta tietokoneelle tuli toteuttaa RS232-väylän kautta.</p> <p>Työssä suunniteltiin ja toteutettiin tilaajan vaatimusten mukaan tiedonsiirto- ja raportointiohjelma. Työssä tutkittiin mittasillan ominaisuuksia ja sen lähettämää merkkijonoa. Ohjelma toteutettiin National Instrumentsin LabVIEW-ohjelmointikielellä.</p> <p>Insinööriyön tuloksena saatiin tarkastusohjelma, joka lukee RS232-väylän avulla mittasillan tulokset. Ohjelma erottelee tarvittavat mittaustulokset merkkijonosta ja tallentaa mittaustulokset tiedostoon. Ohjelma muodostaa käsittelemistään tuloksista tarkastusraportin. Tehtaalla suoritettussa ohjelman testauksessa saatiin luettua tiedot mittasillalta ja luotua tarkastusraportti. Ohjelman exe-versio asennettiin tehtaan omalle tietokoneelle.</p>	
Luottamuksellinen Kyllä Ei X	
Hakusanat LabVIEW, kondensaattori, lopputarkastus	
Säilytyspaikka Evox Rifa Suomussalmen tehdas Kajaanin ammattikorkeakoulun kirjasto	



**Kajaanin
ammattikorkeakoulu**

Kajaani Polytechnic

**ABSTRACT
THESIS**

Faculty Engineering	Degree programme Information Technology
Author(s) Satu Oikarinen	
Title A Data Transfer and Report Program for the Final Inspection Device of a Capacitor	
Optional professional studies Design for Testability	Instructor(s) / Supervisor(s) Pentti Romppainen Esa Matero
Date March 2005	Total number of pages 46
Abstract <p>The purpose of this Bachelor's thesis was to design and implement a data transfer and report program for the final inspection device of a capacitor for Evox Rifa Oy. The reading of the measurement results from the measurement bridge of the testing/sorting device to the computer was to be implemented via the RS232 bus.</p> <p>A data transfer and report program was designed and implemented according to the commissioner's requirements. The properties of the measurement bridge and its output format were investigated. The program was implemented with the National Instrument's programming language LabVIEW.</p> <p>The final outcome of the thesis was an inspection program which reads the results of the measurement bridge via the RS232 bus. The program separates the required measurement results from the output and saves them to the file. The program creates the inspection report from the processed measurement results. The program was tested in the factory. The output of the measurement bridge was read and the inspection report was created by the program. The executable version of the program was installed to the computer of the factory.</p>	
Confidential Yes No X	
Keywords LabVIEW, capacitor, final inspection	
Deposited at Evox Rifa Suomussalmi factory Kajaani Polytechnic Library	

KÄYTETYT TERMIT

A	Pinta-ala
C	Kapasitanssi
C_p	Prosessin maksimisuorituskyky
C_{pk}	Prosessin suorituskyky
d	Välimatka
D	Dissipation factor, häviökerroin
DUT	Device Under Test, testattava laite
IR	Insulation Resistance, eristysvastus
kHz	Kilohertsi
MHz	Megahertsi
Q	Kondensaattorin sähkövaraus
U	Jännite
VI	Virtual Instrument

SISÄLLYSLUETTELO

1	JOHDANTO	6
2	KONDENSAATTORIEN LOPPUTARKASTUS	8
2.1	Kondensaattori	8
2.2	Kondensaattorin valmistus	9
2.3	Lopputarkastus Evox Rifan Suomussalmen tehtaalla	14
3	LOPPUTARKASTUSOHJELMAN SUUNNITTELU JA TOTEUTUS	21
3.1	Ohjelmalle asetetut vaatimukset	21
3.2	Lopputarkastusohjelman toteuttaminen ja LabVIEW	22
3.3	Lopputarkastuksen laitejärjestely	24
3.4	Ohjelman rakenne ja toiminta	25
3.4.1.	Etupaneelin toiminnat	25
3.4.2.	Diagrammi-ikkunan toiminta	28
4	TESTAUS	38
4.1	Testausjärjestelyt	38
4.2	Mittaustulokset	40
5	TULOSTEN TARKASTELO	41
5.1	Testauksen tulokset	41
5.2	Ohjelman käyttöönotto	42
5.3	Ohjelman kehittäminen	42
6	YHTEENVETO	44
	LÄHDELUETTELO	46

LIITTEET

1 JOHDANTO

Tämä insinööriyö on tehty Evox Rifa Oy:lle, joka on toiminut Suomussalmella vuodesta 1979 lähtien. Yritys valmistaa muovikalvoeristeisiä pintaliitos- ja radiaalikondensaattoreita.

Evox Rifan juuret ulottuvat 1940-luvulle asti. Ruotsalainen Rifa perustettiin vuonna 1942 ja suomalainen Evox vuonna 1947. Evox Rifa syntyi vuonna 1988, kun Finvest Oy yhdisti 1980-luvulla omistukseensa hankkimansa Evoxin ja Rifan. Evox Rifa -konserni listautui Helsingin pörssiin syksyllä 2000.

Evox Rifa -konserni valmistaa elektrolyytti-, muovikalvo- ja paperikondensaattoreita. Yrityksen suurimmat asiakasryhmät ovat teollisuus-, auto- ja kulutuselektroniikka sekä valaisinteollisuus. Evox Rifa -konsernin pääkonttori on Suomessa, ja sen tuotantolaitokset sijaitsevat Indonesiassa, Iso-Britanniassa, Kiinassa, Ruotsissa ja Suomessa.

Kondensaattorin valmistaminen on monivaiheinen prosessi. Jokaisen työvaiheen jälkeen kondensaattorit tarkastetaan. Kondensaattorien lopputarkastus suoritetaan siten, että tarkastettavat komponentit asetellaan käsin kuljettimelle. Tämän jälkeen mittapää ottaa kuljettimelta tarkastettavan komponentin leukoihinsa ja suorittaa tarkastuksen, jossa mitataan kapasitanssin toleranssi, häviökerroin ja eristysvastus. Lajittelutiedon perusteella mittapää pudottaa komponentit tiettyihin lokeroihin. Laskuri laskee jokaisen lokeron komponentit.

Tämän insinööriyön tavoitteena oli toteuttaa tiedonsiirto- ja raportointiohjelma kondensaattorien lopputarkastukseen. Mittaustulosten lukeminen testaus/lajittelukoneen mittasillalta tietokoneelle tuli toteuttaa RS232-väylän kautta.

Aluksi tutustuttiin mittalaitteen toimintaan ja tutkittiin sen lähettämää merkkijonoa. Erityisesti täytyi tutkia mittalaitteen lähettämää merkkijonoa mittapään ollessa tyhjä. Laadittavan ohjelman tuli automaattisesti lukea käyttäjän mitta-

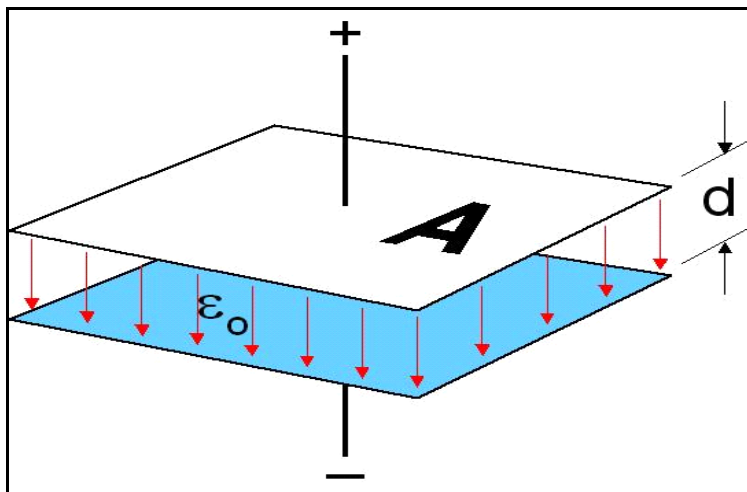
sillalle asettamat arvot, kuten kapasitanssin nimellisarvo, mittaustaajuudet sekä häviökertoimille ja kapasitanssipoikkeamille asetetut raja-arvot.

Yksittäiset mittaustulokset tuli tallentaa tiedostoon. Tuloksista täytyi muodostaa graafisia kuvaajia ja laskea prosessille tyypillisiä tunnuslukuja. Lopuksi ohjelman tuli muodostaa laskemistaan tunnusluvuista ja kuvaajista tarkastusraportti. Tietokoneohjelman tuli olla myös helppokäyttöinen. Laadittu ohjelma tuli testata ja käyttöönottaa tuotannossa. Mittaustulosten tiedonkeruuohjelma toteutettiin LabVIEW'illa.

2 KONDENSAATTORIEN LOPPUTARKASTUS

2.1 Kondensaattori

Kondensaattori muodostuu kahdesta sähköä johtavasta levystä, joiden välissä on eristemateriaali. Kokonaisuudella on kyky varastoida sähköenergiaa (kuva 1). Levyjen välissä olevan eristeaineen ominaisuudet vaikuttavat komponentin käyttösovelluksiin.



Kuva 1. Kondensaattori [1]

Kondensaattorin varautumiskyky eli kapasitanssi C lasketaan perusyhtälöstä

$$C = \frac{Q}{U}, \quad (1)$$

missä Q on kondensaattorin sähkövaraus ja U jännite.

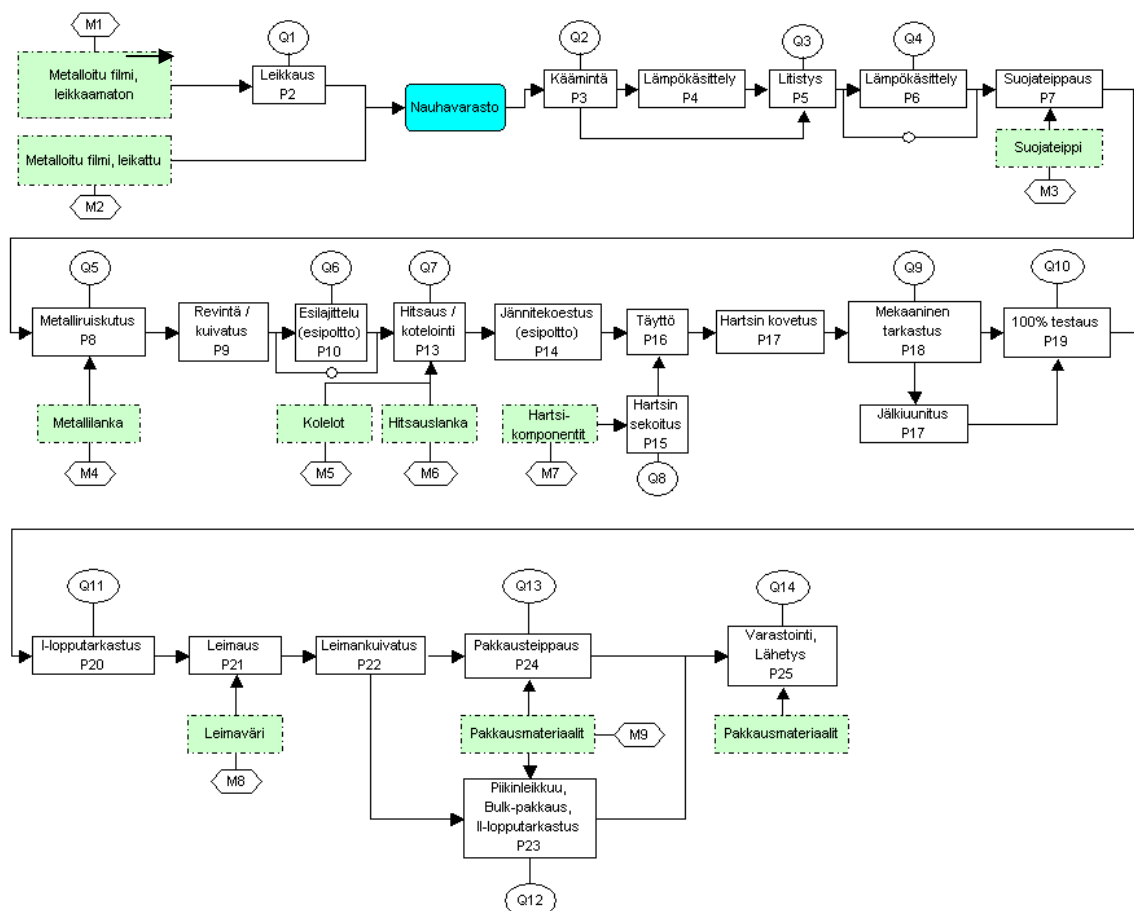
Levykondensaattorissa (kuva 1), jossa kummankin levyn pinta-ala on A , levyjen välinen etäisyys d , kondensaattorin kapasitanssi C lasketaan yhtälöllä

$$C = \varepsilon_r \varepsilon_0 \frac{A}{d}, \quad (2)$$

missä ε_r on väliaineen suhteellinen permittiivisyys ja ε_0 on tyhjiön permittiivisyys.[2.]

2.2 Kondensaattorin valmistus

Kondensaattorin valmistus sisältää monia eri valmistusvaiheita. Kuvassa 2 on esitetty Evox Rifan Suomussalmen tehtaalla käytetty radiaalikonkondensaattorin valmistusprosessi. Pintaliitoskomponentin tuotantoketju ei poikkea paljon radiaalikonkondensaattorin valmistuksesta.



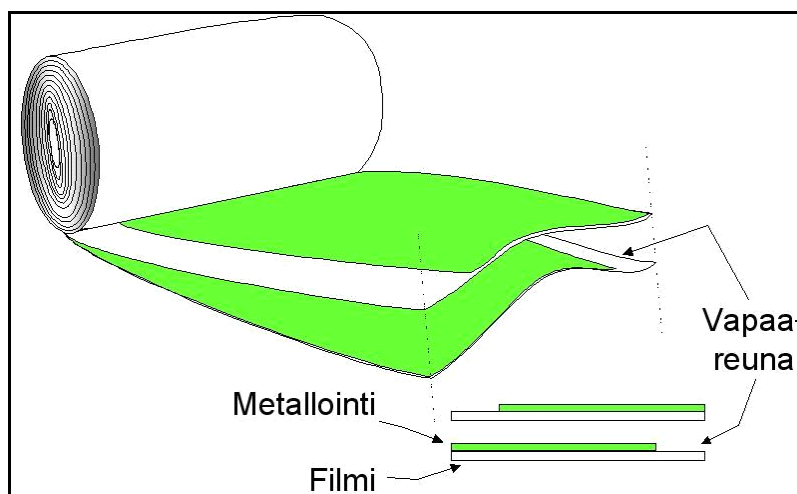
Kuva 2. Radiaalikonkondensaattorin valmistusprosessin lohkokaavio [3]

Kuvassa esiintyvät symbolit Q1-Q14 tarkoittavat laaduntarkastuspisteitä ja M1-M9 materiaalien hyväksymispisteitä. Lohkokaaviossa esiintyviä valmistusvaiheita selitetään tarkemmin seuraavassa.

Kondensaattorin valmistus aloitetaan tehtaalle valmiiksi toimitetusta raaka-aineesta, ohuesta muovikalvosta, jonka pinnalle on höyrystetty metallipinnoite. Ohuin Suomussalmen tehtaalla käytettävistä muovikalvoista on paksuudeltaan 1 μm . Muovikalvon paksuudet riippuvat tarvittavasta jännitekestosta, ja ne vaihtelevat sen mukaan. Suurin osa käytettävistä muovikalvoista on 1 μm :n ja 10 μm :n välillä. Jokaisesta materiaalin tuloerästä otetaan näyte, josta vastaanottotarkastuksessa mitataan neliövastus, vapaareuna ja leveys [4]. Raaka-ainetta kutsutaan filmiksi.

Käämintä

Kääminnässä metalloidut ja oikeaan leveyteen leikatut filmit rullataan käämeiksi. Käämikierroksien lukumäärä vaikuttaa pinta-alan suuruuteen ja sen seurauksena kapasitanssiarvoon. Myös käämintäfilmityyppi vaikuttaa kondensaattorin ominaisuuksiin.[1] Kondensaattorikäämi on esitetty kuvassa 3.



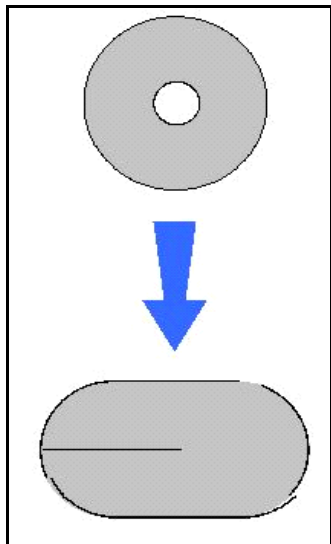
Kuva 3. Kondensaattorikäämi [1]

Vapaareunan leveys ja eristekalvon tyyppi ja vahvuus vaikuttavat kondensaattorin jännitteen keston.

Kääminnän laaduntarkastuksessa mitataan käämin pituus, kapasitanssi sekä kondensaattorin jännitekesto kuvaava eristysvastus. Jatkuvasti tarkkaillaan visuaalisesti käämin laatua. Lisäksi seurataan käämin kovuutta eli sitä, kuinka tiukaksi paketiksi käämintäkone on rullannut käämin. [4.]

Litistys

Litistuksen tarkoituksena on muotoilla pyöreästä kondensaattorikämmistä litteä, muuttumaton rakenne, joka saadaan mahtumaan haluttuun koteloon. Litistäminen vaikuttaa myös kondensaattorin sähköisiin ominaisuuksiin. Suomussalmen tehtaalla on käytössä lämpö- ja kylmlitistys sekä näiden yhdistelmä, jota kutsutaan tornilitistykseksi. Kuvassa 4 on esitetty litistetty käämi.



Kuva 4. Litistetty käämi [1]

Litistuksen jälkeen käämeille suoritetaan uunissa lämpökäsittely, jonka tarkoituksena on vakiinnuttaa niiden muoto ja kapasitanssiarvo. Laatua

tarkkaillaan visuaalisesti sekä mittaamalla erillisen tuotanto-ohjeen mukaisesti kondensaattorin kapasitanssiarvo ja eristysvastus [4].

Metalliruiskutus

Ennen metalliruiskutusta litistettyjen käämien rungot suojataan suojateipillä siten, että käämin päädyt jäävät näkyviin. Metalliruiskutuksen tarkoituksena on muodostaa kondensaattorikäämien päätyihin tietyn paksuinen metallikerros, johon filmin päällä oleva metalli saa kontaktin. Filmillä oleva vapaareuna estää oikosulut.

Laatua tarkkaillaan mm. mittaamalla tietyin väliajoin ruiskutepinnan paksuutta. Kuitenkin mittaus suoritetaan aina ruiskutemateriaalin vaihtuessa. Lisäksi tarkkaillaan visuaalisesti ruiskutepinnan tasaisuutta. [4.]

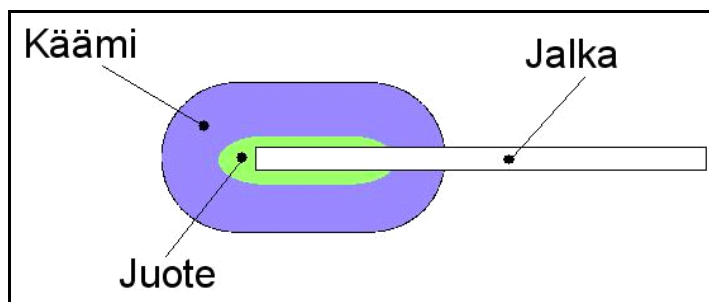
Revintä

Revinnän tarkoituksena on irrottaa päätyruiskutetut käämit suojateippauksesta. Tämän jälkeen kondensaattorit rummutetaan, jonka tarkoituksena on poistaa niihin ruiskutuksen ja revinnän jälkeen mahdollisesti jääneet ruiskutusreunat ja paperit.

Laatua seurataan visuaalisesti mm. tutkimalla käämin päässä olevan ruiskutepinnan tasaisuutta. Rummutuksen jälkeen tarkastetaan, että käämit ovat puhtaita. [4.]

Hitsaus ja kotelointi

Hitsausprosessissa metalloituihin käämipäätyihin juotetaan kondensaattorin jalat ja pintaliitosmallien juotosraina. Tällöin aikaansaadaan sähköä johtava kontakti muovikalvon metallikerrokseen. Jalallinen kondensaattori on esitetty kuvassa 5.



Kuva 5. Jalallinen kondensaattori[1]

Hitsauksen jälkeen kondensaattori laitetaan muoviseen koteloon ja siihen jäävä tyhjä tila täytetään hartsilla. Täytön tarkoituksena on suojata kondensaattori-käämi ympäristöolosuhteiden rasituksilta. Tämän jälkeen kondensaattorit laitetaan uuniin, jonka tarkoituksena on nopeuttaa hartsin kovettumista.

Työvaiheessa tarkkaillaan mm. hartsin laatua (seossuhde, kovuus) ja määrää. Lisäksi mitataan erillisen tuotanto-ohjeen mukaisesti kondensaattorien jalkojen väli (rasterimittaus), lankojen pituus ja vetolujuus. Visuaalisesti tarkkaillaan myös langan muotoa ja suoruutta. [4.]

Mekaaninen tarkastus

Kaikille kondensaattorille suoritetaan mekaanisten ominaisuuksien tarkastus. Tällöin tutkitaan mm. hartsin, rasterin (komponentin jalkojen väli) ja kotelo. Lisäksi tarkastetaan komponentin jalat ja niiden vinous, pituus ja asemointi. [4.]

Testaus

Kondensaattorien sähköisten ominaisuuksien testauksen tarkoituksena on lajitella edellisistä valmistusvaiheista tulevat hyvät ja huonot komponentit toisistaan. Hyvät komponentit lajitellaan myös eri toleransseihin. Nk. sata-prosenttisessä testauksessa mitataan jokaisen kondensaattorin kapasitanssin toleranssi (C), häviökerroin (D) ja kondensaattorin jännitekesto kuvaava eristysvastus (IR).

2.3 Lopputarkastus Evox Rifan Suomussalmen tehtaalla

Lopputarkastus on testauksen jälkeinen tuotantovaihe, jolla varmistetaan, että testaus on onnistunut. Evox Rifan Suomussalmen tehtaalla kondensaattorien lopputarkastus suoritetaan näytteenottomenetelmällä. Jokaisesta testauksessa toleransseihin lajitelluista tuotanto- eli valmistuseristä otetaan näyte, jolle tehdään sekä mekaanisen että sähköisen laaduntarkastus. Näytteet otetaan valmistuserän viimeisimmäksi testatuista tuotteista. Tarkastettavan näyte-erän suuruus määräytyy valmistuserän suuruuden mukaan.

Mekaanisessa lopputarkastuksessa mitataan kondensaattorin kotelon mekaaniset mitat, rasterimitta (jalkojen välinen etäisyys), lankojen pituus ja paksuus. Lisäksi tarkastetaan visuaalisesti kotelonastat, hartsivalun määrä ja sen sijainti. Tarkastuksen tuotanto-ohjeissa on määritelty tuotteiden tarkastustuloksen raja-arvot.

Sähköisen laadun lopputarkastuksessa näyte-erän kondensaattoreista mitataan kapasitanssin toleranssi (C), häviökerroin (D) ja eristysvastus (IR). Tarkastustuloksen perusteella koko valmistuserä joko hyväksytään tai hylätään. Tällöin näyte-erässä ei saa olla yhtään viallista kondensaattoria. Mikäli tarkastuserästä löytyy yksikin viallinen komponentti, koko valmistuserä palautetaan uudelleen testattavaksi. Tämän jälkeen suoritetaan toinen lopputarkastus ja mikäli näyte-erästä ei löydy viallisia komponentteja, erä hyväksytään. Muutoin koko valmistuserä toimitetaan tutkittavaksi laadunvarmistukseen.

Hyväksytyjen erien tarkastustulokset (mekaaninen ja sähköinen laatu) kirjataan lopputarkastus- ja eräseurantaraaporttiin. Tulokset syötetään myös yrityksen tietokantaan. Mikäli näyte-erästä on löytynyt viallisia komponentteja kirjataan niiden kappalemäärät ja vikatyypit eräseurantaraaporttiin. Lisäksi täytetään erillinen palautusraportti ja syötetään tiedot tietokantaan. ATK:lle syötettyjä lopputarkastustietoja käytetään tilastolliseen tarkasteluun.

Lopputarkastuslaitteisto

Lopputarkastuslaitteisto muodostuu tärastinkuljettimesta, mittausyksiköstä sekä testiräkistä, johon mittalaite ja sen tarvitsema teholähde on sijoitettu. Spieck-merkkisellä testaus- eli tarkastuslaitteella mitataan kondensaattorin sähköiset ominaisuudet. Lopputarkastuslaitteisto on esitetty kuvassa 6.



Kuva 6. Lopputarkastuslaitteisto

Kuvassa olevaa laitteistoa käytetään radiaalikonkondensaattorien lopputarkastuksessa. Spieck-testauslaite on kuvassa etualalla ja testiräkkiin sijoitettu mittalaite oikealla. Tärastinkuljetin (kuvassa vasemmalla) sisältää erikokoisia uria, jotka on jaoteltu kotelotyyppien mukaan. Testiräkissä olevaan mittalaitteeseen asetetaan raja-arvot (toleranssirajat, häviökertoimet, mittaustaajuudet) ja nimellisarvo. Tämän jälkeen mitattavat kondensaattorit ladotaan käsin tietyille kuljettimen uralle. Tärinän vaikutuksesta komponentit asettuvat oikeaan asentoon ja kuljetin kuljettaa ne testauslaitteelle.

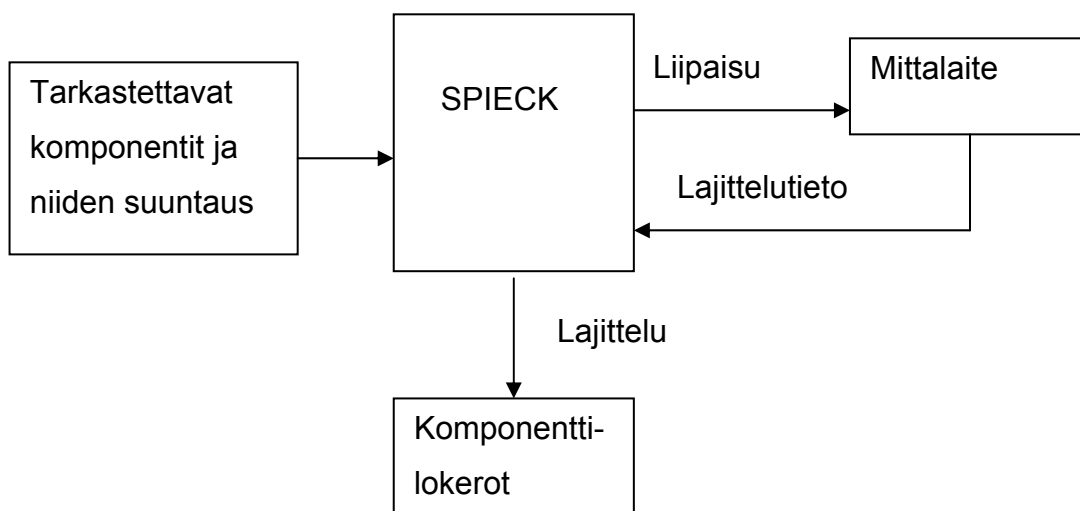
Testauslaitteessa (Spieck) olevassa mittapyörässä on 30 kappaletta leukapareja, joilla tartutaan tarkastettavaan komponenttiin. Kondensaattoreita ladataan ja puretaan halutulla koestusjännitteellä.

Mittauksia, joita kondensaattorille suoritetaan, ovat kapasitanssipoikkeama-, eristysvastus-, häviökerroin- ja oikosulkumittaukset. Tarkastuksen jälkeen leukapari pudottaa kondensaattorin tiettyyn lokeroon mittalaitteelta saamansa lajittelutiedon mukaan. Seuraavassa on esitelty laitteen toiminta ja sen ominaisuuksia.

Mittalaite

Suomussalmen tehtaalla mittauksissa käytetään tanskalaisen Danbridgen valmistamia laitteita malleiltaan DB230 ja CT30. Mittareiden toimintaperiaatteena on ns. siltamittaus. Tällöin itse silta sijaitsee joko laitteen sisällä tai sen ulkopuolella. Tyyppien DB230 ja CT30 mittasillat ovat ulkoisia. Lisäksi tehtaalla käytetään käsin suoritettavissa tarkastuksissa saman valmistajan pöytämallisia DB231-laitteita, joissa silta on sisäinen.

Tarkastustapahtuman lohkokaavio on esitetty kuvassa 7.

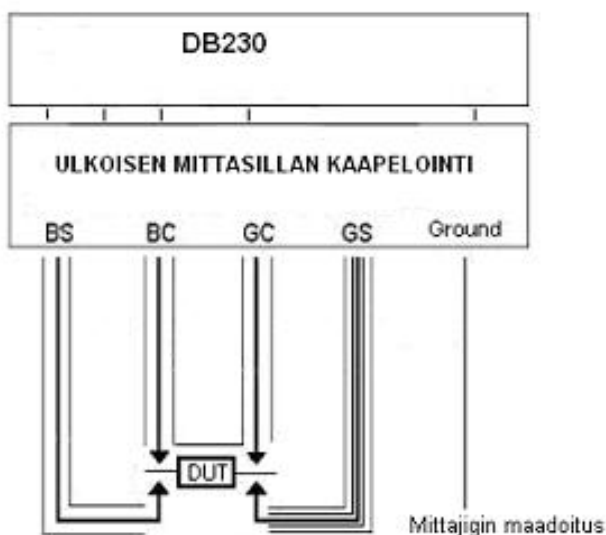


Kuva 7. Mittaustapahtuman lohkokaavio

Tarkastettavat komponentit ladotaan kuljettimelle, joka asettaa ne oikeaan asentoon ja kuljettaa testauslaitteelle (Spieck). Mittalaite muodostaa tuloksen testauslaitteen mittapäässä olevalle kondensaattorille ja antaa nimellisarvon perusteella lajittelutiedon mittapäälle. Saamansa tiedon perusteella testauslaitteen leukapari pudottaa komponentin määrättyyn lokeroon. Mittausyksikössä oleva lajittelija avaa sähköisesti sen lokeron luukun, mihin komponentti on lajiteltu. Testauslaitteen laskuri ilmoittaa eri toleranssialueille lajiteltujen komponenttien ja oikosulkujen kappalemäärät.

Mittalaitteen DB230 ominaisuuksia

DB230 on RLC-mittasilta, joka on suunniteltu nopeisiin ja tarkkoihin mittauksiin useilla taajuuksilla. Laite soveltuu ympäristöihin, joissa vaaditaan komponenttien automaattista lajittelua ja mittausta. Taajuudet, joilla testaus voidaan suorittaa, ovat 1 kHz, 10 kHz, 100 kHz ja 1 MHz. Yksittäinen mittaus voidaan suorittaa joko yhdellä taajuudella tai yhdistämällä em. taajuuksia tai kaikilla em. taajuuksilla. Mittasillan ulkoinen kaapelointi on esitetty kuvassa 8.



Kuva 8. Mittasillan kaapelointi [5, s. 13]

Mittajohtimet ovat koaksiaalikaapelia ja BC- ja GC-kaapelien välille muodostuu virtasilmukka. Vasteen eli jännitteen mittausta tapahtuu BS- ja GS-kaapelien avulla. Yllä olevassa kuvassa DUT on tarkastettava komponentti.

Suomussalmen tehtaalla mittausta tapahtuu kahdella taajuudella joko 1 kHz:n ja 10 kHz:n tai 1 kHz:n ja 100 kHz:n taajuuksilla. Ennen mittaustapahtuman aloittamista käyttäjä määrittelee mittalaitteelle tarkastettavan tuotteen mukana olevasta eräseurantareportista tiettyjä arvoja. Mittarille asetetaan kondensaattorin nimellisarvo, toleranssialueet sekä mittaustaajuuksien ja niiden häviökertoimien raja-arvot.

Mittalaitteen kauko-ohjaus

Mittalaitteessa olevat IEEE488 (GPIB)- ja RS232-väyläliitynnät mahdollistavat laitteen kaikkien toimintojen kauko-ohjaamisen tietokoneelta. Väyläliityntöjen kautta on myös mahdollista kerätä tuloksia mittaustapahtuman aikana tietokoneelle. Kauko-ohjauksen aikana mittarin oma näppäimistö ei toimi lukuun ottamatta menu-näppäintä. Etupaneelin REMOTE-ledi palaa laitteen ollessa kauko-ohjauksessa.

Ohjattaessa mittalaitetta joko IEEE488- tai RS232-väylän kautta mittarin väyläasetukset tulee asettaa kohdalleen laitteen alt-menu-valikosta. Mm. tiedon siirron nopeus, databittien lukumäärä ja pariteettibitti (tarkistusbitti) tulee määrittää. Molempien väylien ohjauskäskyt ovat samanlaisia, mutta RS232-väylää varten on muutamia lisäkäskyjä. RS232-väylän kautta lähetetty syöttödata täytyy päättää LF (line feed) -komentoon. Kaikki tulostusdata on päätetty CR LF -komentoon (carriage return line feed). [5, s. 40].

Mittalaitteen lähettämä merkkijono

Yhden mittalaitteen lähettämän merkkijonon pituus on 44 merkkiä. Laitteen lähettämä merkkijono on taulukossa 1 esitetyn kaavion mukainen.

Taulukko 1. Mittasillan lähettämä merkkijono [5, s. 44]

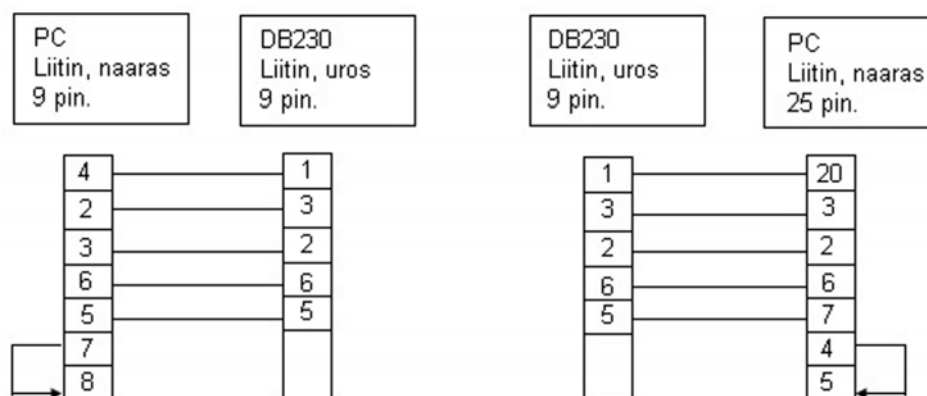
CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10	CH11	
F		1	0	0	K	H	Z		C		
CH12	CH13	CH14	CH15	CH16	CH17	CH18	CH19	CH20	CH21	CH22	
X	.	X	X	X	X	P	F		,		
CH23	CH24	CH25	CH26	CH27	CH28	CH29	CH30	CH31	CH32		
D		0	.	X	X	X	X		;		
CH33	CH34	CH35	CH36	CH37	CH38	CH39	CH40	CH41	CH42	CH43	CH44
B	I	N	X	X	,	B	I	N	X	CR	LF

Taulukossa olevat lyhenteet CH1 – CH44 tarkoittavat lähetettävän merkin numeroa ja merkki X tarkoittaa numeroa välillä 0 - 9. Alarivillä oleva ensimmäinen BIN-luku merkitsee primääristä lajittelua eli se pitää sisällään kapasitanssipoikkeamaa koskevan lajittelutiedon. Toinen BIN-luku on ns. sekundaarinen lajittelu, joka kertoo häviökertoimen lajittelutiedon. Mittalaitteen lähettämä merkkijono tulostuu esimerkiksi seuraavanlaisena:

F 1KHZ C -1.16% ,D 0.0041; BIN 3,BIN 0

Sarjaliikenne

Tiedonsiirto on mahdollista toteuttaa 9- tai 25-nastaisella kaapelilla. Molempien kaapeliliittimien nastajärjestys on esitetty kuvassa 9.



Kuva 9. RS232-kaapelin nastajärjestys [5, s. 59]

RS232-liitännän tiedonsiirtonopeus (baud rate) on määritelty 300:sta 9600:aan. Databittien lukumäärä on 7 tai 8. Lisäksi on mahdollista käyttää parillista tai paritonta pariteettibittiä (tarkistusbitti).

3 LOPPUTARKASTUSOHJELMAN SUUNNITTELU JA TOTEUTUS

Taustan lopputarkastusohjelman suunnittelulle asettivat työn tilaajan vaatimukset ja käytettävät laitteet.

3.1 Ohjelmalle asetetut vaatimukset

Käytettävälle ohjelmalle asetettu vaatimus oli mm. sen helppokäyttöisyys. Mittaustulosten tiedonsiirron testaus/lajittelukoneen mittasillalta tietokoneelle tuli toteuttaa RS232-väylän kautta. Ohjelman tuli erotella tarvittavat mittaustulokset merkkijonosta. Lisäksi oli huomioitava tyhjän mittapään aiheuttama merkkijono ja sen erottelu oikeista tuloksista.

Ohjelman tuli automaattisesti lukea käyttäjän mittalaitteelle määrittelemät asetukset ja raja-arvot. Näitä arvoja ovat kapasitanssiin nimellisarvo, mittaustaajuudet sekä häviökertoimille ja kapasitanssipoikkeamille asetetut arvot. Kyseisiä lukuja käytetään ohjelmassa mm. laskettaessa eri toleranssialueelle sijoittuneiden komponenttien kappalemääriä.

Mittaustulosten tallennus tuli toteuttaa siten, että ohjelma nimeää tiedoston valmiiksi käyttäjän antamien tuotetietojen mukaan. Käyttäjä määrää vain, minne kohtaan yrityksen tietokantaan tiedosto tallennetaan. Tulosten lisäksi tiedostossa täytyi myös näkyä käyttäjän mittalaitteelle asettamat raja-arvot.

Ohjelman tuli luoda automaattisesti mittaustuloksista tarkastusraportti. Raportissa täytyi olla mm. kuvaajat kapasitanssipoikkeama- ja häviökerroinjakaumista. Lisäksi siinä tuli olla päivämäärä, tuotetiedot (tuotekoodi, eränumero), nimellisarvo ja erilaisia laskettuja tunnuslukuja (keskiarvo, keskijajonta, C_p ja C_{pk}). Ohjelman oli nimettävä myös raportin tallennusnimi automaattisesti. Sekä mittaustulokset että tarkastusraportti tuli tallentaa Excel-tilukkuun. Valmis ohjelma tuli testata ja käyttöönottaa tuotannossa.

3.2 Lopputarkastusohjelman toteuttaminen ja LabVIEW

Tulosten lukeminen mittalaitteelta voidaan toteuttaa monella erilaisella ohjelmistolla. Lopputarkastusohjelma päätettiin toteuttaa LabVIEW'illa, jota käytetään tietokonepohjaisten mittaus- ja testaussovellusten ohjelmoinnissa. Kyseistä ohjelmaa on Suomussalmen tehtaalla käytetty aikaisemmin mm. prosessitietojen keruussa. Käytetty ohjelman versio oli LabVIEW 7.0.

LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) on National Instrumentsin kehittämä graafinen ohjelmointikieli, joka tuli markkinoille vuonna 1986. Perinteisistä rivipohjaisista ohjelmointikielistä poiketen LabVIEW-ohjelma luodaan kuvakkeiden avulla käyttäen vuokaavioesitystä. Graafisesta esityksestä LabVIEW kääntää konekielisen ohjelman. [6.]

LabVIEW tukee esimerkiksi tiedonkeruukortteja, RS232/422-porttia ja kuvankäsittelykortteja. LabVIEW-ohjelmia kutsutaan virtuaali-instrumenteiksi (Virtual Instrument, VI). [6.]

Avattaessa LabVIEW-ohjelma avautuu näyttöön kaksi ikkunaa. Vasemmanpuoleisesta harmaapohjaisesta ikkunasta käytetään nimitystä etupaneeli. Oikeanpuoleista vaaleapohjaista ikkunaa kutsutaan vuokaavio- tai diagrammi-ikkunaksi. Ohjelmaa laadittaessa tarvitaan molempia ikkunoita. [6.]

Etupaneeli toimii ohjelman käyttöliittymänä, johon sijoitetaan kontrollit (Control) ja indikaattorit (Indicator). Kontrollien avulla siirretään data ohjelmaan ja indikaattorin avulla esitetään ohjelman tuottama data. Vuokaavioikkuna koostuu solmuista (ikonit), kontrollien ja indikaattorien terminaaleista ja langoista. Ikonit vastaavat rivipohjaisen ohjelman lausekkeita, funktioita ja aliohjelmia. Terminaalit vastaavat parametreja ja vakioita. [6.]

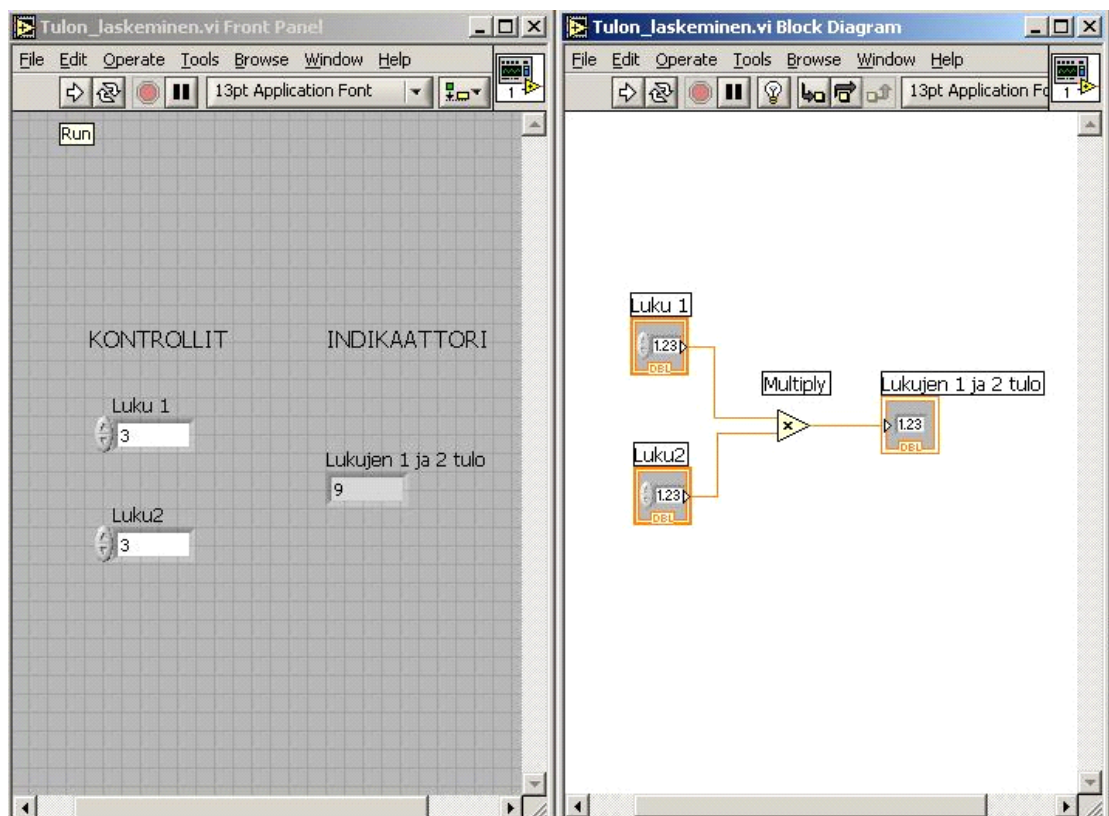
Itse ohjelman tekeminen tapahtuu vuokaavioikkunassa, jossa langoitetaan yhteen kontrollien ja indikaattorien terminaaleja sekä ohjelmassa käytettävien

funktioiden terminaaleja. Terminaalien kautta tapahtuu datan siirto. Kun kontrolli tai indikaattori luodaan etupaneeliin, muodostuu sitä vastaava terminaali automaattisesti vuokaavioikkunaan.

Aliohjelmien avulla ohjelmista voidaan tehdä modulaarisia. Valmiiksi tehtyjä ohjelmia voidaan käyttää aliohjelmina toisissa LabVIEW-sovelluksissa. Aliohjelma on mahdollista testata myös omana kokonaisuutenaan. Aliohjelmien käyttäminen on suositeltavaa, sillä virheiden etsiminen ja ohjelman ylläpito helpottuu.

Esimerkkiohjelma LabVIEW-ohjelmoinnista

LabVIEW-ohjelmointia on havainnollistettu kuvan 10 mukaisella esimerkillä. Ohjelma laskee kahden luvun tulon ja tulostaa saadun arvon.

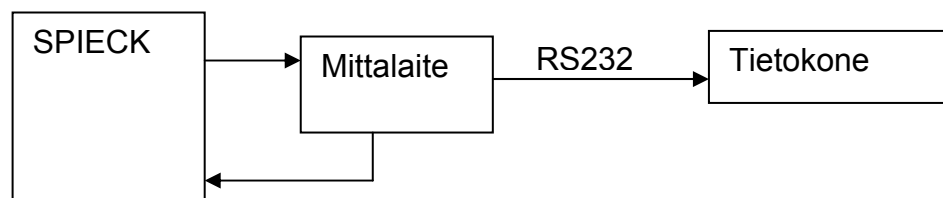


Kuva 10. Esimerkkiohjelma (*Tulon_laskeminen.vi*), etupaneeli-ikkuna ja vuokaavioikkuna

Kuvassa 10 on esitetty vasemmalla puolella etupaneeli-ikkuna ja oikealla puolella vuokaavioikkuna. Etupaneeli-ikkunaan on sijoitettu kaksi kontrollia (Luku 1 ja Luku 2), joiden kautta ohjelmaan syötetään tarvittavat lukuarvot. Kertolaskun tulos tulostuu etupaneelissa olevaan indikaattoriin (Lukujen 1 ja 2 tulo). Vuokaavioikkunassa tapahtuu ohjelmakoodin tekeminen. Luotaessa tarvittavat indikaattorit ja kontrollit etupaneeli-ikkunaan ilmestyvät niitä vastaavat terminaalit automaattisesti vuokaavio-ikkunaan. Jotta kaksi syötettävää lukua voidaan kertoa keskenään, täytyy vuokaavioikkunaan lisätä Multiply-funktio. Sitten terminaalit langoitetaan langoitustyökalulla. Valmis ohjelma voidaan käynnistää joko etupaneeli-ikkunassa tai vuokaavioikkunassa olevasta valkoisesta nuolinäppäimestä.

3.3 Lopputarkastuksen laitejärjestely

Tietokone liitetään RS232-kaapelilla tarkastuksessa käytettävän mittalaitteen takana olevaan liittimeen. Laitejärjestely on esitetty kuvassa 11.

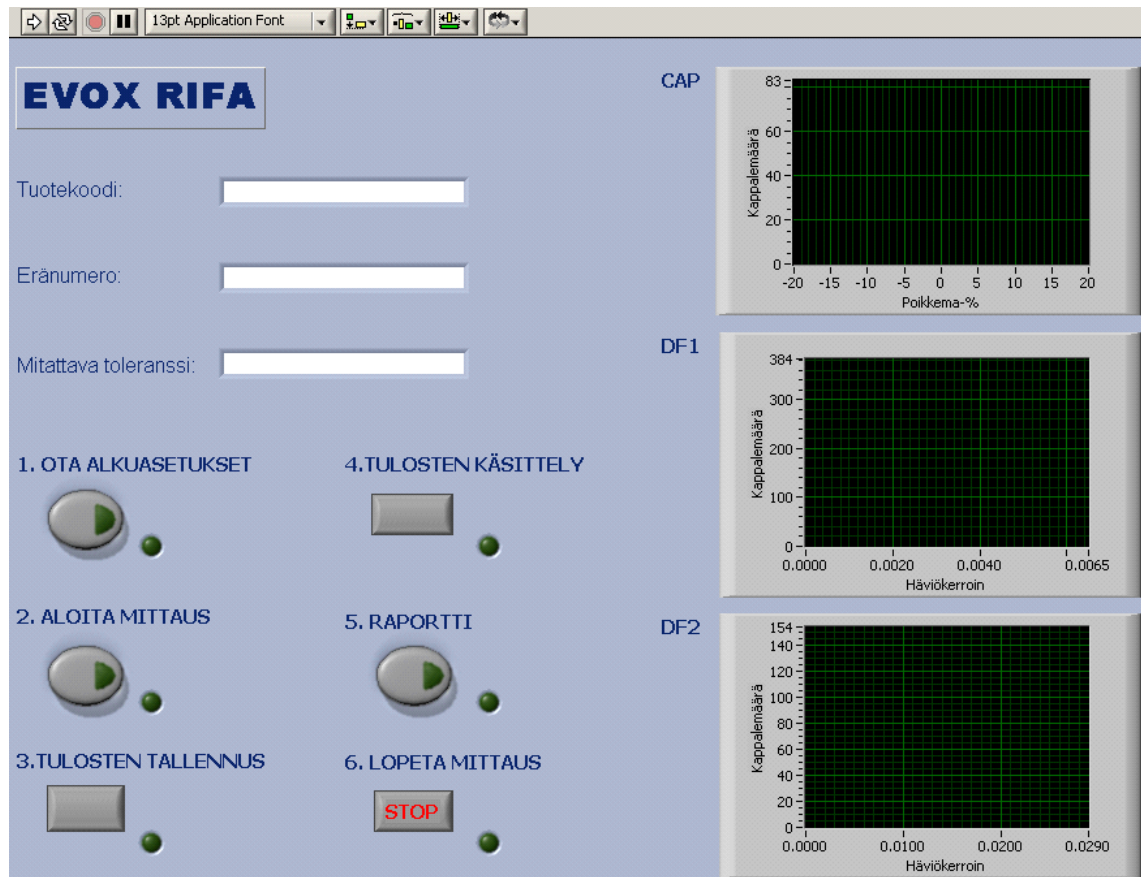


Kuva 11. Laitejärjestely

Käyttäjä määrittelee mittalaitteelle erän mukana tulevasta eräseurantareportista vaaditut raja-arvot. Tämän jälkeen ohjelman käyttöliittymään kirjataan tuotetiedot (tuotekoodi, eränumero, mitattava toleranssi). Sen jälkeen käynnistetään ohjelma. Tarkastuslaitteen (Spieck) käynnistämisen jälkeen ohjelma aloittaa mittaustulosten lukemisen RS232-väylän kautta. Tarkastuslaitteen pysäyttämisen jälkeen ohjelma tallentaa mittaustulokset ja tarkastusraportin yrityksen tietokantaan.

3.4 Ohjelman rakenne ja toiminta

Ohjelmalle asetettujen vaatimusten mukaisesti laaditun sovelluksen käyttöliittymä on esitetty kuvassa 12.



Kuva 12. Ohjelman käyttöliittymä.

Ohjelman päätoiminnot on toteutettu Event-rakenteen avulla. Käyttäjä huolehtii näppäimien painalluksella ohjelman suoritusjärjestyksestä. Kunkin näppäimen vieressä oleva ledimerkkivalo ilmaisee käyttäjälle, että kyseessä oleva Event-kehys on suoritettu loppuun.

3.4.1. Etupaneelin toiminnot

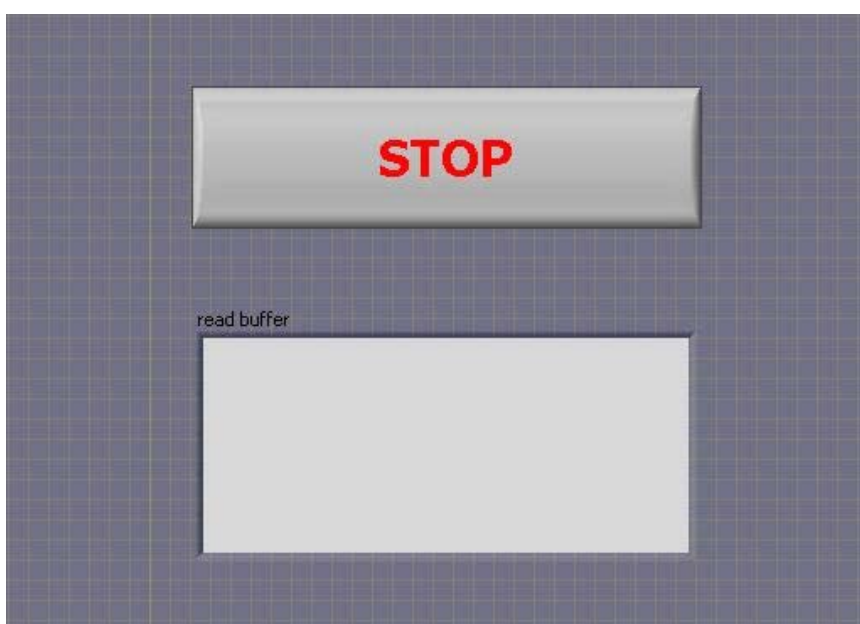
Kuvan 12 mukainen etupaneeli ilmestyy näyttöön, kun LabVIEW-sovellus avataan. Ennen ohjelman käynnistämistä käyttäjä kirjoittaa etupaneelissa

oleviin kohtiin tuotetta koskevia tietoja. Ohjelma muodostaa automaattisesti tallennettavan tiedoston nimen käyttäjän näppäilemien tuotetietojen perusteella. Tarkastusraportille muodostuvan tiedostonimen eteen ohjelma lisää sanan raportti, jotta siitä muodostuisi erillinen tiedosto. Sekä mittaustulokset että raportti tallentuvat Excel-taulukkoon.

Käyttäjän antamien tuotetietojen jälkeen ohjelma käynnistetään runnäppäimellä. Etupaneelissa olevien näppäimien avulla määrätään ohjelman suorituserjestys. Näppäimen vieressä oleva ledi muuttuu keltaiseksi, kun kyseessä oleva ohjelmakehys on suoritettu.

Näppäintä 1 (OTA ALKUASETUKSET) painettaessa ohjelma lukee RS232-väylän kautta mittalaitteelta käyttäjän asettamat arvot (nimellisarvo, mittaustaajuudet, toleranssipoikkeamien sekä häviökertoimien raja-arvot).

Näppäintä 2 (ALOITA MITTAUS) painettaessa ohjelma suorittaa aliohjelman, jossa tapahtuu mittaustulosten lukeminen RS232-väylän kautta. Kun näppäintä 2 on painettu, niin etupaneelin näkymä muuttuu kuvassa 13 esitetyn mukaiseksi.



Kuva 13. Aliohjelman etupaneeli

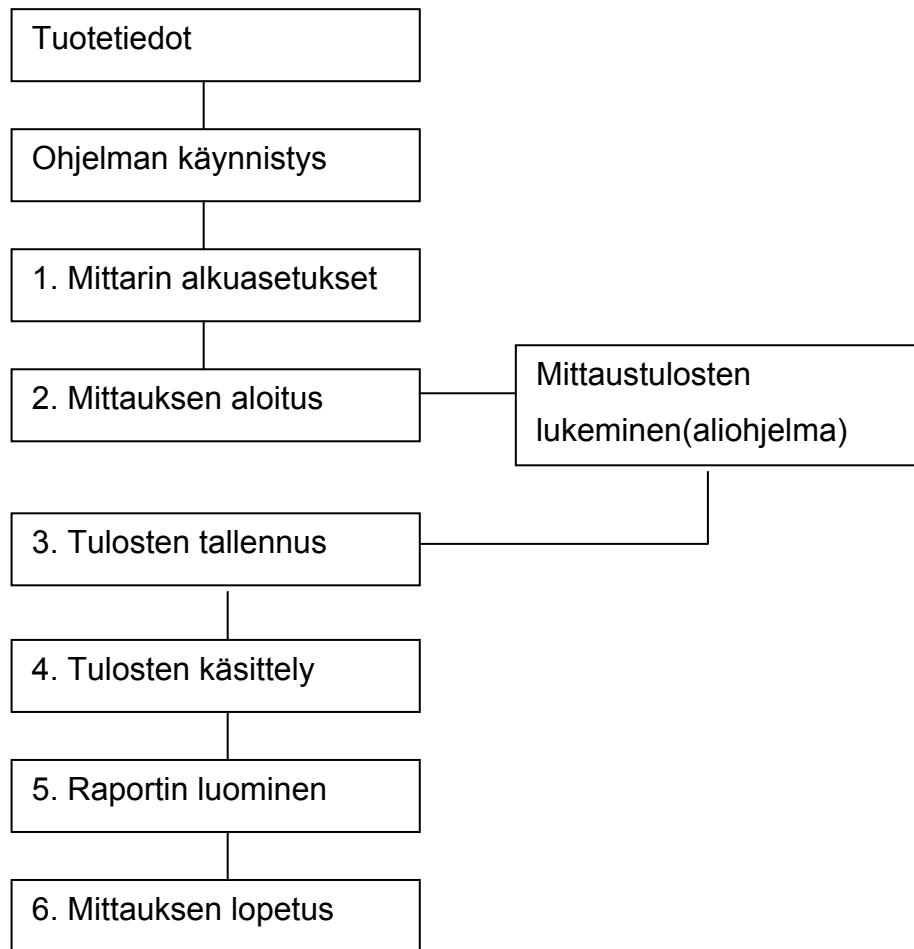
Tämän jälkeen käynnistetään tarkastuslaitteisto (Spieck). Kuvassa 13 olevassa käyttöliittymässä (read buffer) näkyy kulloinkin luettavana oleva merkkijono. Kun tarkastuslaitteisto katkaistaan ja käyttäjä painaa etupaneelissa olevaa STOP-näppäintä, vaihtuu näkymä takaisin kuvan 12 mukaiseksi.

Näppäintä 3 (TULOSTEN TALLENNUS) painettaessa ohjelma tallentaa mittaustulokset Excel-taulukkoon. Tallennettavan tiedoston nimi määräytyy annettujen tuotetietojen mukaisesti.

Näppäintä 4 (TULOSTEN KÄSITTELY) painettaessa ohjelma käsittelee mittaustuloksia tarkastusraporttia varten. Se laskee mm. eri toleranssialueille sijoittuneiden komponenttien kappalemäärät. Lisäksi ohjelma luo tuloksista etupaneeliin ilmestyvät kolme histogrammia. Samat kuvaajat esiintyvät myös tarkastusraportissa.

Näppäintä 5 (RAPORTTI) painettaessa ohjelma luo mittaustuloksista raportin. Lopuksi näppäintä 6 (LOPETA MITTAUS) painettaessa ohjelman suoritus pysähtyy.

Työn tilaajan vaatimusten mukaan toteuttavan ohjelman ratkaisun pohjaksi laadittiin kuvan 14 mukainen lohkokaavio.



Kuva 14. Ohjelman lohkokaavio

Toteutettu ohjelma sisältää pääohjelman lisäksi aliohjelman, jossa luetaan merkkijono ja erotellaan tarvittavat mittaustulokset jonosta. Ohjelmasta laadittu lohkokaavio loi pohjan toteutettavalle ohjelmalle.

3.4.2. Diagrammi-ikkunan toiminta

Perinteisissä rivipohjaisissa ohjelmointikielissä käskyt suoritetaan siinä järjestyksessä kuin ne on kirjoitettu. LabVIEW:ssä solmu suoritetaan, kun data on kaikissa vaadituissa tuloissa. Mikäli halutaan määrittää ohjelman suoritusjärjestystä, on LabVIEW-ohjelmoinnissa käytettävissä erilaisia rakenteita.

While-silmukka toistaa ohjelmakoodia, kunnes sen ehtotermiäliin kytkeyty data on tosi tai epätosi. For-silmukka eroaa while-silmukasta siten, että sitä toistetaan tietty määrä kertoja.

Sekvenssirakenteen (Sequence Structure) avulla voidaan määrätä LabVIEW-koodin suoritusjärjestystä. Rakenteessa ensin suoritettava diagrammi asetetaan ensimmäiseen sekvenssiin toinen toiseen ja niin edelleen. Yksittäistä sekvenssiä kutsutaan sekvenssikehykseksi.

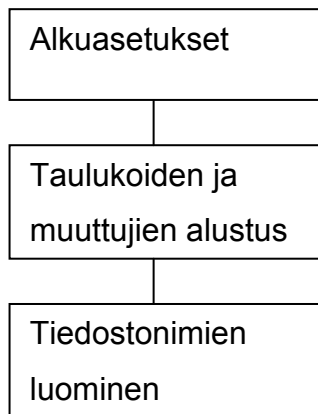
Case-rakenteen (Case Structure) eli tapauskehysten avulla haarautetaan ohjelman suoritusta. Rakenne koostuu useista päällekkäisistä kehyksistä, joista yksi suoritetaan rakenteeseen langoitetun ehdon perusteella.

Event-rakenne (Event Structure) koostuu Event-kehyksistä. Käytettäessä Event-rakennetta kehyksien suorittaminen määrätään etupaneelissa olevien näppäimien avulla.

Tarkastusohjelman diagrammi-ikkuna koostuu Event-rakenteesta, joka on while-silmukan sisällä. Event-rakenne sisältää kuusi erilaista kehystä. Mikäli mitään etupaneelissa olevaa näppäintä ei ole painettu, niin ohjelma ei tee mitään. Kun jotakin näppäintä painetaan, niin ohjelma suorittaa kyseisen kehyksen. Esimerkiksi kun painetaan etupaneelin näppäintä yksi, ohjelma lukee mittalaitteen alkuasetukset. Kun painetaan näppäintä kuusi, ohjelman suoritus pysähtyy.

Event-kehys 1 (Ota alkuasetukset)

Kun painetaan näppäintä 1, ohjelma suorittaa kuvassa 15 esitetyn lohkokaaavion mukaiset toiminnot.



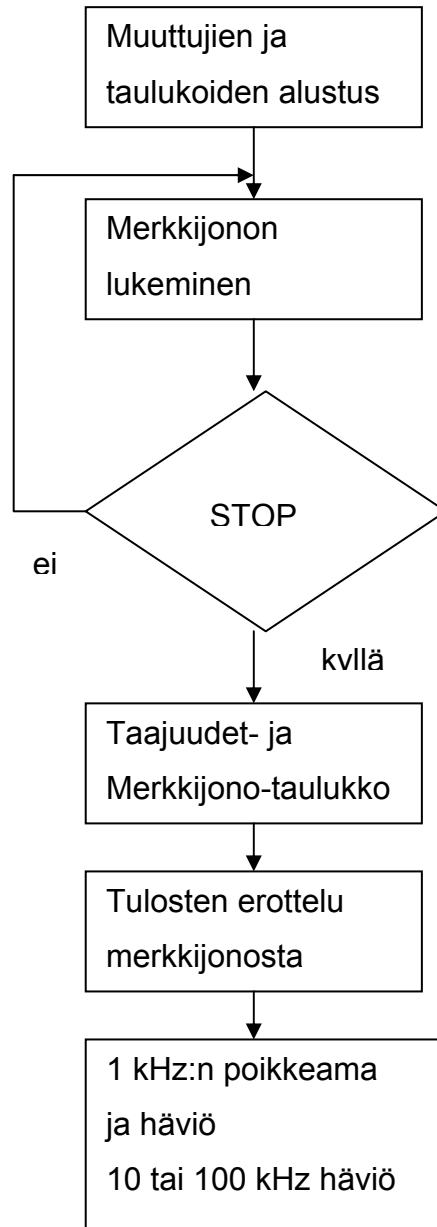
Kuva 15. Event-kehys 1, lohkokaavio

Tapauskehyksessä ohjelma lukee mittarilta käyttäjän määrittelemät alkuasetukset (nimellisarvo, mittaustaajuudet, kapasitanssipoikkeama- sekä häviökertoimien raja-arvot). Tällöin mittalaite siirtyy kauko-ohjaukseen. Samassa kehyksessä mittarilta luettuja arvoja muokataan siten, että niitä voidaan myöhemmin käyttää ohjelmassa.

Kuvassa 15 esitetyt kaksi alimmaista lohkoa suoritetaan sekvenssirakenteessa. Ensimmäisessä sekvenssikehyksessä alustetaan ohjelmassa käytettävät muuttujat ja taulukot. Toisessa sekvenssikehyksessä luodaan tiedoston tallennusnimi. Sekä mittaustuloksille että raportille muodostetaan oma tiedostonimi.

Event-kehys 2 (Aloita mittaus)

Painettaessa näppäintä 2 ohjelma siirtyy toiseen Event-kehykseen. Se koostuu sekvenssirakenteesta, jossa on kaksi sekvenssikehystä. Ensimmäisessä niistä ohjelma siirtyy suorittamaan aliohjelman (E_merkkijonoerottelu.vi), jossa tapahtuu mittaustulosten lukeminen ja niiden erottelu merkkijonosta. Aliohjelman vuokaavio on esitetty kuvassa 16.



Kuva 16. Aliohjelman vuokaavio

Ensin aliohjelmassa alustetaan sekvenssikehyksessä tarvittavat muuttujat ja taulukot. Merkkijonon lukeminen tapahtuu while-silmukassa. Kun kaikki tarvittavat mittaustulokset on luettu, käyttäjä sammuttaa tarkastuslaitteen (Spieck) ja painaa ohjelman käyttöliittymässä olevaa STOP-näppäintä. Tämän jälkeen mittalaite siirtyy paikallisohjaukseen, jolloin laitteen oma näppäimistö toimii.

Luetut merkit tallentuvat kahteen eri taulukkoon. *Taajuudet* niminen -taulukko sisältää pelkän taajuustiedon (0, 1, 10 tai 100). *Merkkijono* nimiseen taulukkoon

tallentuu merkkijonossa taajuuden jälkeen esiintyvät merkit ja numerot (esimerkiksi C -1.16% ,D 0.0041; BIN 3,BIN 09).

Ohjelma erottaa merkkijonosta tarvittavat mittaustulokset for-silmukassa olevan tapausrakenteen avulla. Suoritettava tapauskehys määräytyy taajuustiedon mukaan. Esimerkiksi 1 kHz:n mittaustuloksen sisältämä merkkijono käsitellään numerolla 1 erotetussa tapauskehyksessä. Merkkijonon erottelussa käytetään funktiota (*Match Pattern*), joka etsii tietyn merkin merkkijonosta.

Tyhjän mittapään aiheuttaman merkkijonon (-.---E+-; BIN 12,BIN 1) käsittely tapahtuu numerolla 0 nimetyssä tapauskehyksessä eli nk. oletuskehyksessä. Muita tapauksia ovat 1, 10 ja 100. Merkkijonosta erotellut tarvittavat mittaustulokset päivittyvät kolmeen eri taulukkoon (*1 kHz:n poikkeama*, *1.häviö*, *2.häviö*).

1 kHz:n poikkeama -niminen taulukko sisältää kapasitanssipoikkeamat. *1.häviö* -niminen taulukko sisältää 1 kHz:n häviökertoimet. *2.häviö* -niminen taulukko sisältää joko 10 kHz:n tai 100 kHz:n mittaustulokset riippuen mittaus-taajuudesta. Ennen pääohjelmaan siirtämistä aliohjelmassa tutkitaan valinta-funktion (*Select*) avulla taulukoiden (10 kHz ja 100 kHz) kokoa. Esimerkiksi jos 10 kHz:n taulukko on yhtä suuri kuin nolla, viedään 100 kHz:n häviökerroin-tulokset pääohjelmaan. Toimenpide suoritetaan, jotta pääohjelmaan ei siirrettäisi tyhjää taulukkoa.

Kun aliohjelma on suoritettu loppuun, siirtyy ohjelman suoritus pääohjelmaan. Aliohjelmassa luetut merkkijonot ja siitä erotellut mittaustulokset muodostavat sekvenssikehyksessä kolme erilaista taulukkoa (*Tod.mittaustulokset*, *Tod.mittaustulokset 2*, *Tod.mittaustulokset 3*).

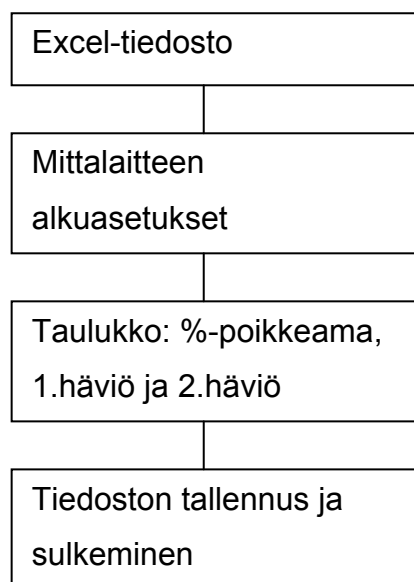
Taulukoiden muodostamisen jälkeen ohjelma siirtyy toiseen sekvenssi-kehukseen, jossa poistetaan aiheettomat mittaustulokset. Joissakin tapauksissa tyhjä mittapää saattaa aiheuttaa merkkijonon, joka muistuttaa normaalista mittauksesta syntyvää merkkijonoa. Merkkijono on esimerkiksi muotoa F 1KHZ

C -100.00% ,D 0.0188; BIN 0,BIN 1. Kyseinen tyhjän mittapään aiheuttama tulos poistetaan for-silmukassa.

Silmukassa olevan tapauskehyksen avulla tutkitaan, onko kapasitanssi-poikkeama -100. Mikäli kapasitanssi-poikkeaman mittaustulos (*Tod.mittaustulokset*) on -100, poistetaan se ja sitä vastaavat häviökertoimet (1 kHz:n ja 10 tai 100 kHz:n) tuloksista. Aiheettomien tulosten poistamisen jälkeen oikeat mittaustulokset muodostavat kolme taulukkoa (*kapasitanssi*, *häviök1* ja *häviök2*). *Kapasitanssi*-taulukko sisältää 1 kHz:n kapasitanssi-poikkeamat ja *häviök1*-taulukko 1 kHz:n häviökertoimet. *Häviök2*-taulukko muodostuu joko 10 kHz:n tai 100 kHz:n häviökertoimista riippuen mittaustaajuudesta.

Event-kehys 3 (Tulosten tallennus)

Kun painetaan näppäintä 3, ohjelma suorittaa kuvassa 17 esitetyn lohkokaaavion mukaiset toiminnot.



Kuva 17. Event-kehys 3, lohkokaavio

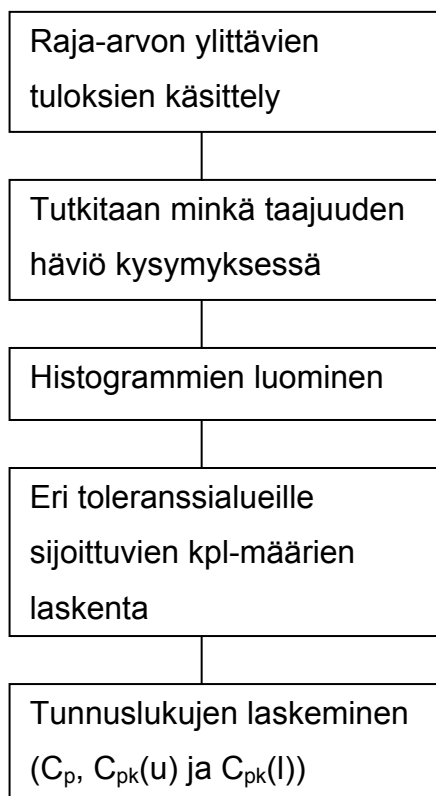
Mittaustulosten tallennus tapahtuu sekvenssikehyksessä. Ohjelmassa määritellään tallennettava tiedosto Excel-tyypiseksi. Muodostuvan Excel-tiedoston yläosaan ohjelma lisää mittalaitteelta luetut alkuasetukset (nimellis-

arvo, mittaustaajuudet, kapasitanssipoikkeamien sekä häviökertoimien raja-arvot).

Mittaustulokset tallentuvat taulukossa rinnakkain kolmeen eri sarakkeeseen. Sarakkeet on otsikoitu valmiiksi ohjelmassa. Ensimmäiseen sarakkeeseen (*%-poikkeama*) tallentuvat 1 kHz:n poikkeamat. Toiseen sarakkeeseen (*1.häviö*) tallentuvat 1 kHz:n häviökertoimet ja kolmanteen (*2.häviö*) joko 10 tai 100 kHz:n häviökertoimet riippuen mittaustaajuudesta. Ohjelma ehdottaa tiedostolle tallennusnimeä, jonka se on muodostanut ensimmäisessä Event-kehyksessä. Kun tiedoston tallennusnimi on hyväksytty, tiedosto sulkeutuu automaattisesti.

Event-kehys 4 (Tulosten käsittely)

Kun painetaan näppäintä 4, ohjelma suorittaa kuvassa 18 esitetyn lohko-kaavion mukaiset toiminnot. Event-kehys muodostuu sekvenssirakenteesta, joka sisältää viisi erilaista sekvenssikehystä.



Kuva 18. Event-kehys 4, lohko-kaavio

Ensimmäinen sekvenssikehyks sisältää kaksiosaisen sekvenssikehyksen, jossa erotellaan tuloksista ne arvot, jotka ylittävät mittalaitteelle asetetut raja-arvot. Lisäksi siinä tutkitaan, minkä taajuuden häviö on kysymyksessä (1 kHz vai 10 kHz tai 100 kHz riippuen mittaustaajuudesta). Tutkiminen tapahtuu for-silmukassa olevien tapauskehysten avulla.

Sekvenssikehyksessä verrataan tulosta (*1.häviök*) mittalaitteelle asetettuun raja-arvoon. Mikäli se on suurempi kuin raja-arvo, luetaan kyseinen tulos 1 kHz:n häviöksi. Jos tulos on taas pienempi kuin raja-arvo, verrataan toisen häviökertoimen (*2.hävik*) tulosta sille asetettuun raja-arvoon. Mikäli tulos ylittää raja-arvon, luetaan se joko 10 kHz:n tai 100 kHz:n häviöksi riippuen mittaustaajuudesta. Tulokset, jotka ylittävät raja-arvot ilmoitetaan raportissa vain kappalemäärinä.

Toisessa sekvenssikehyksessä luodaan käyttöliittymään ilmestyvät histogrammit (CAP, DF1 ja DF2). Kapasitanssipoikkeamia kuvaava histogrammi näyttää tulokset, jotka sijoittuvat -20 % ja +20 % väliselle alueelle. Häviökertoimista (DF1 ja DF2) muodostuviin kuvaajiin tulostuvat pelkästään raja-arvon suuruiset ja sitä pienemmät tulokset.

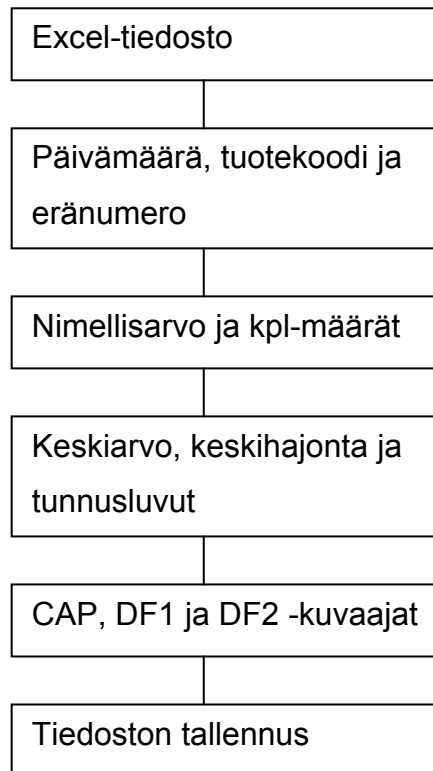
Kolmannessa sekvenssikehyksessä lasketaan eri toleranssialueille sijoittuvien komponenttien kappalemäärät, jotka esitetään raportissa. Kappalemäärien laskenta tapahtuu for-silmukassa olevien tapauskehysten avulla.

Neljännessä sekvenssikehyksessä mittaustuloksista saadut kapasitanssipoikkeamat (prosenttiarvot) muutetaan kapasitanssiarvoiksi. Saaduista arvoista lasketaan keskiarvo ja keskihajonta. Sekvenssikehyksessä lasketaan myös mitattujen kondensaattorien kokonaiskappalemäärä.

Viidennessä sekvenssikehyksessä lasketaan prosessille tyypillisiä tunnuslukuja. Näitä lukuja ovat mm. C_p , $C_{pk}(u)$ ja $C_{pk}(l)$. Laskemisessa käytetään edellisessä sekvenssikehyksessä saatua keskiarvoa ja keskihajontaa.

Event-kehys 5 (Raportti)

Kun painetaan näppäintä 5, ohjelma suorittaa kuvassa 19 esitetyn lohkokaaavion mukaiset toiminnot. Event-kehys sisältää sekvenssirakenteen, joka koostuu seitsemästä erilaisesta sekvenssikehyksestä. Rakenteessa käytetään valmiita LabVIEW'n Excel-kirjastofunktioita.



Kuva 19. Event-kehys 5, tarkastusraportin luominen

Ohjelma luo Excel-tyyppisen tarkastusraportin, johon tulostuu päivämäärä, kellonaika, tuotetiedot (tuotekoodi, eränumero, nimellisarvo) ja eri kapasitanssi-alueille jakaantuneiden tulosten kappalemäärät. Lisäksi siihen tulostuu pelkkänä kappalemääränä häviökertoimet, jotka ylittävät mittalaitteelle asetetut raja-arvot. Raportissa näkyvät myös mitattujen komponenttien kokonaiskappalemäärä sekä ohjelmassa lasketut keskiarvo, keskihajonta ja tunnusluvut (C_p , $C_{pk}(u)$ ja $C_{pk}(l)$). Mittaustuloksista käyttöliittymään muodostuvat kolme jakaumakuvaava (CAP, DF1 ja DF2) näkyvät myös tarkastusraportissa.

Ohjelma tallentaa raportin Excel-tiedostoon. Tallennettavan tiedoston nimi muodostuu käyttäjän antamien tuotetietojen (tuotekoodi, eränumero, mitattava toleranssi) perusteella. Ohjelma lisää tiedostonimen eteen sanan raportti, jotta tiedosto tallentuu eri nimelle kuin mittaustulokset. Tallentamisen jälkeen tiedosto jää avoimeksi, jotta tarkastusraportti voidaan tarvittaessa tulostaa.

Event-kehys 6 (Lopeta mittaus)

Kun painetaan näppäintä 6, ohjelman toiminta pysähtyy. Kyseisessä Event-kehyksessä alustetaan käyttöliittymässä olevat ledimerkkivalot.

4 TESTAUS

Valmis ohjelma testattiin Evox Rifan Suomussalmen tehtaalla. Laadittuun ohjelmaan lisättiin osio, jossa kaikki mittalaitteelta luetut merkit tallentuivat erilliseen tekstitiedostoon. Näin voitiin verrata, kuinka ohjelma oli erotellut ja käsitellyt merkkijonosta erottelemansa merkit. Testausilanteessa pyrittiin aiheuttamaan erilaisia vikatilanteita. Pysäytettiin mm. Spieck-testauslaite ja estettiin kondensaattorin pääsy mittapyörälle. Siten ohjelmaa saatiin testattua tilanteissa, joissa mittapää oli tyhjä.

Testauksen jälkeen ohjelman tuottamassa tarkastusraportissa olevia kappalemääriä verrattiin testauslaitteen laskemiin kappalemääriin. Ohjelman käsittelemiä ja tallentamia mittaustuloksia verrattiin mittalaitteelta luettuun merkkijonotiedostoon. Testausilanteessa mitattiin satunnainen määrä kondensaattoreita. Ohjelmakoodi testattiin siitä laaditulla ajo- eli exe-versiolla.

4.1 Testausjärjestelyt

Ohjelmasta laadittu exe-versio testattiin tehtaalla työvaiheessa, jossa kondensaattoreille suoritetaan sataprosenttinen sähköinen testaus. Laittejärjestelyt olivat kuvan 11 mukaiset (sivu 24).

Testauskomponentit

Testauskondensaattoreina käytettiin pintaliitoskomponentteja, joiden tuotekoodi oli SMCK331040050, eränumero 29923000 ja versio 0. Nimellisarvo oli 100 nF. Testauskomponenttien lukumäärää ei määritelty etukäteen, vaan ohjelmaa testattiin silloin, kun todellisen tuotantoerän sähköinen testaus oli käynnissä. Ohjelman testauksessa käytettävät kondensaattorit oli jo kertaalleen testattu. Ensimmäisessä testauksessa kyseiset komponentit oli lajiteltu IR-viallisiin. Siten ohjelman testiajon aikana testatut kondensaattorit poikkesivat normaalista näyte-erästä.

Mittalaitteen DB230 asetukset

Ohjelman testattiin liittämällä tietokone RS232-kaapelilla mittalaitteeseen DB230. Koska ohjelmaa testattiin kesken tuotantoerän testauksen, oli mittalaitteelle valmiiksi määritelty tarkastettavia kondensaattoreita koskevat tiedot (nimellisarvo, mittaustaajuudet, häviökertoimien ja kapasitanssipoikkeamien raja-arvot). Mittalaitteen väyläasetukset asetettiin RS232-väylälle sopiviksi ja säädettiin seuraavanlaisiksi:

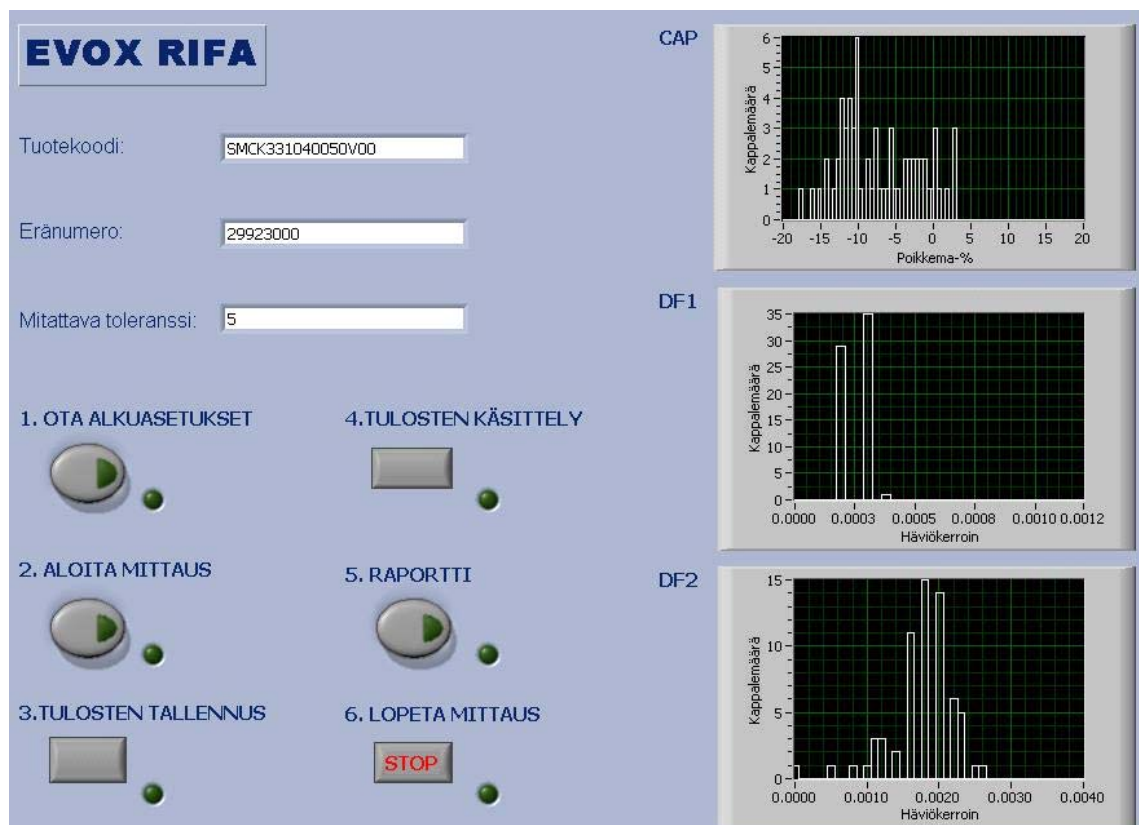
- Baud Rate 9600
- Data Bits 8
- No Parity
- Stop Bits 1

Testauslaitteen senhetkiset kappalemäärät kirjattiin, jotta niitä voitiin myöhemmin verrata ohjelman testauksessa muodostamaan tarkastusraportin arvoihin. Ohjelman käyttöliittymään kirjoitettiin tuotetta koskevia tietoja (tuotekoodi, eränumero, mitattava toleranssi), jotka saatiin valmistuserän mukana olevasta eräseurantareportista. Tämän jälkeen käynnistettiin ohjelma. Kun ohjelma oli lukenut mittalaitteen alkuasetukset, määriteltiin tiedostolle nimi, johon merkit tallentuivat. Tämän jälkeen käynnistettiin Spieck-testauslaite.

Mittaustapahtuman aikana pysäytettiin testauslaite. Epämääräisin väliajoin estettiin kondensaattorien eteneminen mittapyörälle, jotta väliin jäisi tyhjiä leukapareja. Kun testauslaite oli mitannut tarvittavan määrän komponentteja, se sammutettiin. Tämän jälkeen painettiin ohjelman käyttöliittymässä olevaa STOP-näppäintä, jonka seurauksena ohjelman suoritus siirtyi pääohjelmaan. Ohjelma suoritettiin loppuun käyttöliittymässä olevien näppäimien avulla.

4.2 Mittaustulokset

Ohjelma muodosti testaustapahtuman aikana mittalaitteelta lukemistaan merkkijonoista ja niistä erottelemistaan mittaustuloksista kuvassa 20 esitetyt kolme graafista kuvaajaa.



Kuva 20. Käyttöliittymä ohjelman testauksen jälkeen

Kuvassa esitetyt kapasitanssipoikkeamat (CAP) asettuvat -20% :n ja $+20\%$:n välille. Häviökertoimia (DF1 ja DF2) esittävässä kuvaajissa näkyvät pelkästään raja-arvon suuruiset tai sitä pienemmät arvot. Liitteessä A on esitetty ohjelman muodostama tarkastusraportti. Ohjelman testauksessa saatuja tarkempia mittaustuloksia ei ole tarpeellista esittää tässä työssä.

5 TULOSTEN TARKASTELU

5.1 Testauksen tulokset

Ohjelma muodosti testauksessa saaduista mittaustuloksista tarkastusraportin ja tallensi mittalaitteelta luetut merkit erilliseen tiedostoon. Tarvittavat mittaustulokset, jotka ohjelma oli erotellut lukemistaan merkeistä, tallentuivat myös omaan tiedostoon. Mittaustuloksista muodostuvan Excel-taulukon yläosaan oli ohjelma lisännyt mittalaitteen alkuasetukset (nimellisarvo, mittaustaajuudet, kapasitanssipoikkeamien- ja häviökertoimien raja-arvot).

Aliohjelmassa suoritettava tarvittavien merkkien erottelu mittalaitteen lähettämästä merkkijonosta toimi suunnitellulla tavalla. Aliohjelma erotteli oikein tyhjiä mittapyörän leukaparien aiheuttamat merkkijonot ja esti niiden pääsyn pääohjelmaan. Suoritettujen tarkastuslaskelmien perusteella ohjelma laski oikein keskiarvon, keskihajonnan ja prosessin tunnusluvut (C_p , $C_{pk}(u)$ ja $C_{pk}(l)$).

Käyttöliittymässä esiintyvät kolme graafista kuvaajaa oli ohjelma muodostanut suunnitellulla tavalla (kuva 20). Kapasitanssipoikkeamia (CAP) esittävässä kuvaajassa arvot asettuivat välille -20 % ja +20 %. Myös häviökertoimista muodostuvat jakaumakuvat (DF1 ja DF2) sisälsivät pelkästään mittalaitteelle asetetun raja-arvon ja sitä pienemmät arvot.

Verrattaessa testauslaitteen laskemia kappalemääriä ohjelman tuottaman tarkastusraportin kappalemääriin huomattiin, että kokonaiskappalemäärä oli sama. Pieniä eroavaisuuksia kuitenkin ilmeni verrattaessa eri toleranssialueille jakaantuneita kappalemääriä. Eroavaisuudet saattoivat aiheutua siitä, että testauslaitteen käsittelemät tulokset sisälsivät eristysvastusmittauksen (IR-mittaus). Mittalaitetta, joka mittaa kondensaattorin eristysvastuksen, ei huomioida laaditussa ohjelmassa.

5.2 Ohjelman käyttöönotto

Ohjelmasta laadittu exe-versio asennettiin tehtaan omalle tietokoneelle. Koska kyseisessä tietokoneessa ei ole LabVIEW-ohjelmistoa, jouduttiin siihen myös asentamaan LabVIEW'n runtime-ohjelma sekä visa301 runtime -ohjelma. Viimeksi mainittu ohjelma tarvittiin, koska ohjelmassa oli käytetty visa-funktioita luettaessa merkkejä RS232-portin kautta. Ohjelmasta laadittiin tehtaalle myös käyttöohje.

Kiireisen aikataulun vuoksi ohjelmalle ei voitu suorittaa enempää testausta. Ohjelman täydellinen käyttöönotto vaatii kuitenkin lisää testaamista. Erityisesti täytyy perehtyä tilanteisiin, joissa mittapyörässä oleva leukapari on tyhjä. Tällöin tulisi tutkia, että aiheuttaako tyhjä leukapari myös muunlaisia merkkijonoja, kuin mitä tässä työssä on esille tullut.

5.3 Ohjelman kehittäminen

Laadittu ohjelma vaatii jatkokehitystä, sillä se ei huomioi eikä käsittele lainkaan eritysvastusmittausta (IR-mittaus). Kyseinen mittaus tulisi lisätä ohjelmaan, koska eristysvastusviallisista kondensaattoreista aiheutuvia tuloksia voi sisältyä hyvien komponenttien lajittelutietoihin. Edellä mainitun lisäyksen jälkeen ohjelmaa olisi mahdollisuus hyödyntää myös testauksessa (sataprosenttinen testaus).

Käyttäjä näppäilee tarkastettavaa tuotetta koskevat alkuasetukset (nimellisarvo, mittaustaaajuudet, poikkeamien ja häviökertoimien raja-arvot) suoraan mittalaitteelle. Kyseisten arvojen määrittely voitaisiin toteuttaa myös ohjelman käyttöliittymästä, sillä mittalaitetta on mahdollisuus kauko-ohjata tietokoneella.

Ohjelman käyttöliittymän graafisilla kuvaajilla ei ole nyt käytännön merkitystä, sillä samat jakaumakuvat on heti nähtävissä tarkastusraportista. Ohjelmaa tulisi muuttaa siten, että mittaustulokset päivittyisivät reaaliaikaisesti ohjelman käyttöliittymässä oleviin histogrammeihin.

Ohjelma olisi toteutettava siten, että tarvittavien tulosten erottelu tulisi tehdä heti mittalaitteen lähettämän merkkijonon lukemisen jälkeen. Merkkien lukeminen ja mittaustulosten erottelu kannattaisi toteuttaa erillisessä aliohjelmassa. Tämän jälkeen toisessa aliohjelmassa tapahtuisi tuloksen käsittely. Kun tulokselle on tehty tarvittavat toimenpiteet, siirtyisi valmiiksi käsitelty mittaustulos pääohjelmaan. Pääohjelmassa tulos päivittyisi käyttöliittymän jakaumakuvaan. Siten mittaustulos olisi heti nähtävissä ja analysoitavissa. Käytettäessä aliohjelmaa vähenisi myös pääohjelmassa tarvittavan ohjelmakoodin määrä.

6 YHTEENVETO

Insinööriyön tavoitteena oli selvittää kondensaattorin lopputarkastuksessa käytettävän mittalaitteen ominaisuuksia ja tehdä tiedonsiirto- ja raportointi-ohjelma. Erityisesti täytyi tutkia mittalaitteen lähettämää merkkijonoa tilanteissa, jolloin testauslaitteen mittapyörän leukapari oli tyhjä. Tulosten lukeminen mittalaitteelta oli toteutettava RS232-väylän kautta. Laadittavan ohjelman tuli automaattisesti tallentaa tulokset ja muodostaa käsittelemistään tuloksista raportti. Ohjelman tuli olla myös helppokäyttöinen.

Työn suorittaminen alkoi tutustumalla käytettävään mittalaitteeseen ja sen käyttöympäristöön. Tutkittiin mittalaitteen lähettämää merkkijonoa ja erityisesti sitä, millaisia merkkejä tyhjä mittapää aiheuttaa. Lopputarkastusohjelma toteutettiin National Instrumentsin graafisella LabVIEW-ohjelmointikielellä. Ohjelman suunnitteluvaiheessa käytettiin apuna Danbridgen valmistamaa pöytämallista mittalaitetta. Ohjelma toteutettiin pienissä osissa. Kun jokin osa ohjelmasta oli saatu laadittua pöytämällisen mittalaitteen avulla, käytiin se testaamassa tehtaalla.

Merkkijonon lukemisen ja selvittämisen jälkeen suunniteltiin, kuinka tarvittavat mittaustulokset saadaan eroteltua merkeistä. Hankaluuksia aiheutti se, kuinka saadaan poistettua tyhjien mittapäiden aiheuttamat merkkijonot luetuista merkeistä. Edellä mainittujen merkkijonojen poistaminen toteutettiin siihen suunnitellun erillisen ratkaisun avulla.

Kun tarvittavat mittaustulokset oli saatu eroteltua merkkijonosta, toteutettiin tulosten käsitteleminen ohjelmassa. Tämän jälkeen suunniteltiin, millainen tarkastusraportin ulkoasu tulee olemaan ja mitä tietoja siinä tullaan esittämään. Raportin ohjelmallisessa laadinnassa ongelmia aiheutti se, kuinka mittaustuloksista saadaan muodostettua graafisia kuvaajia raporttiin. LabVIEW:ssa olevien valmiiden funktioiden avulla toteutettavat kuvaajat eivät vastanneet työn tilaajan vaatimuksia. Valmiilla funktioilla toteutettu graafinen kuvaaja ei ollut riittävän tarkka ja selkeä. Kyseinen ongelma ratkaistiin siten, että käyttö-

liittymään mittaustapahtuman jälkeen ilmestyvät histogrammit siirrettiin sellaisenaan tarkastusraporttiin.

Työn tilaajalla ei ollut erityisiä vaatimuksia ohjelman käyttöliittymää suunniteltaessa ja toteutettaessa. Käyttöliittymän ulkoasu pyrittiinkin laatimaan selkeäksi ja välttämään kirkkaiden värien käyttöä. Ohjelmasta laadittiin erillinen käyttöohje.

Ohjelman ajoversio asennettiin tehtaan omalle tietokoneelle. Koska ajoversio vaatii toimiakseen myös erilaisia runtime-ohjelmia, asennettiin myös ne kohde-koneelle. Ohjelmasta laadittu ajoversio toimi moitteettomasti.

Insinööri työn tekemisessä pääpaino oli lopputarkastusohjelman suunnittelussa ja toteuttamisessa. Tavoitteena oli toteuttaa helppokäyttöinen tiedonsiirto- ja raportointiohjelma kondensaattorin lopputarkastukseen. Työn tuloksena saatiin ohjelma, joka lukee mittaustulokset testaus/lajittelukoneen mittasillalta RS232-väylän kautta tietokoneelle. Ohjelma tallentaa merkkijonosta erotellut mittaus-tulokset tiedostoon ja muodostaa käsitellyistä tuloksista tarkastusraportin.

LÄHDELUETTELO

- 1 Evox Rifa, Suomussalmi. Kondensaattorin valmistusprosessi. Power Point esitys.
- 2 Mäkelä, M., Mäkelä, R., Siltanen, O. Insinöörikoulutuksen Fysiikka Kolmas painos. Jyväskylä: Gummerus, 1997. 308 s. ISBN 951-9004-55-6.
- 3 Evox Rifa, Suomussalmi. Tuotantoprosessin vuokaaviot. Excel-
taulukko.
- 4 Evox Rifa, Suomussalmi. Laadunvarmistussuunnitelma. Excel-
taulukko.
- 5 Instruction Manual Component Tester DB230. Elokuu 2002. 61 s.
- 6 National Instruments. LabVIEW Basics I. Introduction Course
Manual,2002.

EVOX RIFA TEST REPORT

Date: 7/3/2005 Time: 15/18

Tuotekoodi: SMCK331040050V00 Eränumero: 29923000

Nimellisarvo: 100.00NF

C1: -0.80% < C1 < 0.80%

C2: -4.80% < C2 < 4.80%

C3: -9.80% < C3 < 9.80%

OUT:<= -9.80% or >= 9.80%

DF1 at 1kHz : >= 0.0012

DF2 at 100kHz: >= 0.0040

Mitattujen lkm:

Kpl

6

23

36

29

2

5

72

Keskiarvo: 92.85

Keskihajonta: 5.2793

Cp: 0.62

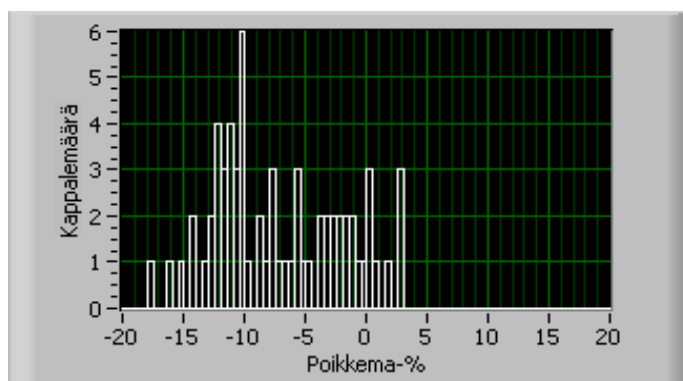
Cpk(u): 1.07

Cpk(l): 0.16

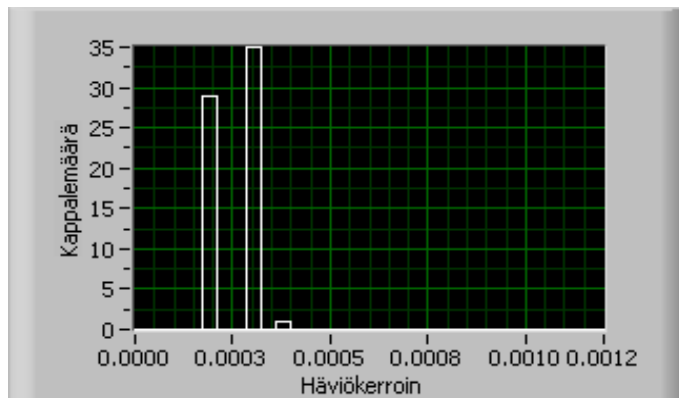
IR:

OIK:

CAP:



DF1:



DF2:

