

Jyrki Heikkinen

KAIUTINELEMENTIN IMPEDANSSIMITTARI

Insinöörityö
Kajaanin ammattikorkeakoulu
Tekniikan ja liikenteen ala
Tietotekniikan koulutusohjelma
Kevät 2007



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Tekniikka	Koulutusohjelma Tietotekniikka
Tekijä(t) Jyrki Heikkinen	
Työn nimi Kaiutinelementin impedanssimittari	
Vaihtoehtoiset ammattiopinnot Langaton tiedonsiirto	Ohjaaja(t) Jukka Heino Toimeksiantaja Ismo Talus
Aika Kevät 2007	Sivumäärä ja liitteet 28+9
<p>Insinööriyössä suunniteltiin ja toteutettiin impedanssimittari. Mittari on tarkoitettu erityisesti kaiutinelementin mittaamiseen, mutta muitakin kuormia voi toki mitata. Mittaria on tarkoitus hyödyntää kaiuttimien ominaisuuksien määrittämisessä. Impedanssi on taajuudesta riippuva, kompleksinen suure ja voi sisältää resistanssin lisäksi reaktanssia. Reaktanssin osuutta voidaan kuvata myös vaihekulmalla.</p> <p>Koska impedanssi on taajuudesta riippuva suure, laitteen on kyettävä mittaamaan eri taajuuksilla. Tavoitteena on saada impedanssi mitattua ainakin alueella 10–200 Hz. Laitteen tulisi mitata impedanssista itseisarvon lisäksi sen vaihekulma. Impedanssin mittausta varten kytketään mitattavan kohteen kanssa sarjaan tunnetunkokoinen vertailuvastus. Kun molempien läpi syötetään sinisignaalia, voidaan impedanssi laskea vertailemalla näiden yli vaikuttavia jännitteitä.</p> <p>Laite koostuu useasta osasta. Ohjauksesta huolehtii mikrokontrolleri johon on sarjaportin kautta yhdistetty PC. Kontrollerissa on C-kielinen ohjelma, joka ohjaa siniaaltogeneraattoria ja AD-muunninta. PC:llä on LabVIEW-ohjelma, joka käsittelee mittaustiedot ja esittää tulokset. Siniaaltogeneraattori toteutetaan käyttäen Walsh-funktioita. Walsh-funktioiden avulla saadaan haluttu aaltomuoto muodostettua digitaalisista signaaleista. Menetelmällä saadaan yksinkertaisesti muodostettua kohtalaisen laadukasta siniaaltoa. Siniaaltogeneraattorin taajuutta ohjataan kontrollerilla. Jännitteen mittausta suoritetaan AD-muuntimella.</p> <p>Laite testattiin mittaamalla bassoelementin impedanssia ja vertailemalla tuloksia teoreettisiin arvoihin. Testit osoittavat että laite toimii kuten pitää ja on tarpeeksi tarkka käytännön mittauksiin.</p>	
Kieli	Suomi
Asiasanat	Impedanssi, kaiutin, elementti
Säilytyspaikka	<input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School School of Engineering	Degree Programme Information Technology
Author(s) Jyrki Heikkinen	
Title A Device for Measuring Loudspeaker Impedance	
Optional Professional Studies Wireless Communication	Instructor(s) Jukka Heino
	Commissioned by Ismo Talus
Date Spring 2007	Total Number of Pages and Appendices 28+9
<p>The purpose of this Bachelor's thesis was to design and build a device that could measure impedance from a loudspeaker. Knowing impedance is useful because most of the speaker parameters can be determined from it. Impedance is a complex unit that consists of resistance and reactance. These can also be expressed as magnitude and a phase angle.</p> <p>Because impedance is a frequency depending unit, the device must be able to measure at different frequencies. The goal is to measure impedance at least from 10 to 200 Hz. The device must measure both parts of impedance, its magnitude and phase angle. Impedance is measured by connecting the known reference resistor in series with the loudspeaker, feeding sinusoidal current through them, and then comparing their voltages.</p> <p>The device consists of three main parts: the control unit, signal generator unit and measuring unit. The control unit includes a microcontroller and PC that are connected together via RS232. The controller uses a C language program to control the signal generator and measuring units. The PC has a LabVIEW program that collects data as well as calculates and displays the results. The signal generator feeds sinusoidal current to the load. A sine wave is produced with the Walsh functions. That is a method for creating an arbitrary waveform from the digital signals. The output frequency of the generator can be varied. The measuring unit is just an AD converter that is used to measure the voltage over the reference resistor and loudspeaker.</p> <p>The device was tested by measuring the subwoofer driver impedance. The test results show that the device works as planned, maybe even better.</p>	
Language of Thesis	Finnish
Keywords	Impedance, loudspeaker
Deposited at	<input checked="" type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Idea insinööriyöhön on tullut kiinnostuksesta kaiuttimien ominaisuuksiin ja niiden mittamiseen. Kaiutinelementin impedanssin perusteella on mahdollista määrittää elementin ominaisuudet.

Haluan kiittää työn ohjaajaa Jukka Heinoa sekä työn tilaajaa Ismo Talusta. Lisäksi kiitän Eero Soinista ja Kaisu Korhosta työn kielellisestä ohjauksesta.

Kajaanissa 12.4.2007

Jyrki Heikkinen

SISÄLLYS

1 JOHDANTO	1
2 KAIUTINELEMENTIN OMINAISUUDET	2
2.1 Yleistä	2
2.2 Impedanssi	2
2.3 Parametrit	4
3 SIGNAALIN MUODOSTAMINEN	7
3.1 Walsh-funktiot	7
3.2 Korrelaatio	10
3.3 Interpolaatio ja ekstrapolaatio	11
4 OHJAUSYKSIKÖN TOTEUTUS	13
4.1 Mikrokontrollerin toimintaan asettaminen	13
4.2 Mikrokontrollerin ohjelmointi	14
4.3 LabVIEW-ohjelma	15
5 SINIAALTOGENERAATTORIN TOTEUTTAMINEN	17
5.1 Siniaaltoapproksimaatio Walsh-funktioilla	17
5.2 Suodattimen toteuttaminen	20
5.3 Vahvistimen toteuttaminen	21
6 MITTAUKSEN TOTEUTTAMINEN	22
6.1 Mittauskytkentä	22
6.2 AD-muunnin	23
7 LAITTEEN TESTAUS	24
8 LOPPUANALYYSI	26
9 YHTEENVETO	27
LÄHTEET	28
LIITTEET	

1 JOHDANTO

Insinööriyön tarkoituksena on suunnitella ja toteuttaa kaiutinelementin mittaamiseen soveltuva impedanssimittari. Idea työhön syntyi mielenkiinnosta kaiutinelementtien ominaisuuksiin ja niiden mittaamiseen. Tarkoituksena on toteuttaa edullinen mutta tarpeeksi tarkka laite, jotta sitä voi käyttää elementin ominaisuuksien määrittämiseen.

Impedanssi on taajuudesta riippuva suure, joka koostuu resistanssin lisäksi reaktanssista. Reaktanssi voi olla joko kapasitiivista tai induktiivista. Reaktanssin osuutta kuvataan usein vaihekulmalla. Mittaria on tarkoitus käyttää nimenomaan bassoelementin mittaamiseen, joten sen täytyy toimia matalilla taajuuksilla. Mittarin mittausalueeksi halutaan vähintään 10–200 Hz ja 1–100 Ω . Mittarin tulisi selvittää kuormasta impedanssin itseisarvon lisäksi vaihekulma. Impedanssi on tarkoitus mitata syöttämällä sinimuotoista signaalia tunnetunkokoisen vertailuvastuksen ja kuorman läpi sekä vertailemalla näiden välisiä jännitteitä.

Impedanssin mittausta varten työhön täytyy toteuttaa siniaaltogeneraattori ja jännitemittari. Koska impedanssi on taajuudesta riippuva suure, täytyy siniaaltogeneraattorin taajuutta pystyä säätämään. Siniaaltogeneraattorin ja jännitemittarin ohjaus toteutetaan mikrokontrollerilla. Koska mittaustulosten käsittely mikrokontrollerilla olisi hankalaa, täytyy ne siirtää PC:lle. PC:llä mittaustulokset voidaan tehokkaasti käsitellä käyttäen LabVIEW-ohjelmointia.

Siniaaltogeneraattori toteutetaan käyttämällä Walsh-funktioita. Walsh-funktioiden avulla saadaan luotua mikä tahansa aaltomuoto summaamalla digitaalisia signaaleja keskenään. Menetelmällä saadaan yksinkertaisesti toteutettua varsin laadukasta siniaaltoja. Jännitemittaus toteutetaan AD-muuntimella. Muuntimella otetaan näytteitä vertailuvastuksen ja mitattavan kuorman jännitteistä.

2 KAIUTINELEMENTIN OMINAISUUDET

2.1 Yleistä

Suurin osa nykyään käytössä olevista kaiuttimista on toimintaperiaatteeltaan dynaamisia kartiokaiutinelementtejä. Niissä signaali tuodaan voimakkaaseen magneettikenttään sijoitetulle puhekelalle, joka on kiinnitetty kartionmuotoiseen värähtimeen. Puhekelalle tuotu sähköinen signaali aiheuttaa kartion edestakaisen liikkeen, jonka pituus riippuu signaalin voimakkuudesta ja taajuudesta. Iskunpituuden ja kartion pinta-alan tulo on iskutilavuus. Mitä korkeampi on signaalin taajuus, sen pienempi iskutilavuus riittää määrätyn äänenpaineen saavuttamiseksi. Taajuuden madaltuessa täytyy kartion liikepoikkeaman kasvaa neliöllisesti, jotta äänen-voimakkuus pysyisi vakiona. [1, s. 54–55.]

Kaiutinelementissä puhekela liikkuu kapeassa ilmaraossa. Elementin kyky liikepoikkeamaan onkin suoraan verrannollinen magneettiraon ja puhekelan mekaanisiin mittoihin. Mitä pidempi puhekela on, sitä laajempi on lineaarisena pysyvä liikepoikkeama. Pitkä puhekela huonontaa elementin hyötysuhdetta sillä paras hyötysuhde saavutetaan kun mahdollisimman suuri osa puhekelaa on magneettiraossa. [2, s. 168.]

Jos tavallinen kaiutinelementti saa olla vapaassa ilmatilassa, kumoavat kaiuttimen edestä ja takaa lähtevät vastakkaisvaiheiset signaalit toisiaan. Kumoutuminen on sitä tehokkaampaa, mitä pidempi on aallonpituus, eli matalampi taajuus. Tätä ilmiötä kutsutaan akustiseksi oikosuluksi. Kun aallonpituus on huomattavasti kartion halkaisijaa pienempi, ei kumoutumaa suoraan edestä mitattuna enää havaita. Akustisesta oikosulusta johtuen on varsinkin matalia taajuuksia toistamaan tarkoitettu elementti syytä koteloida. [1, s. 55.]

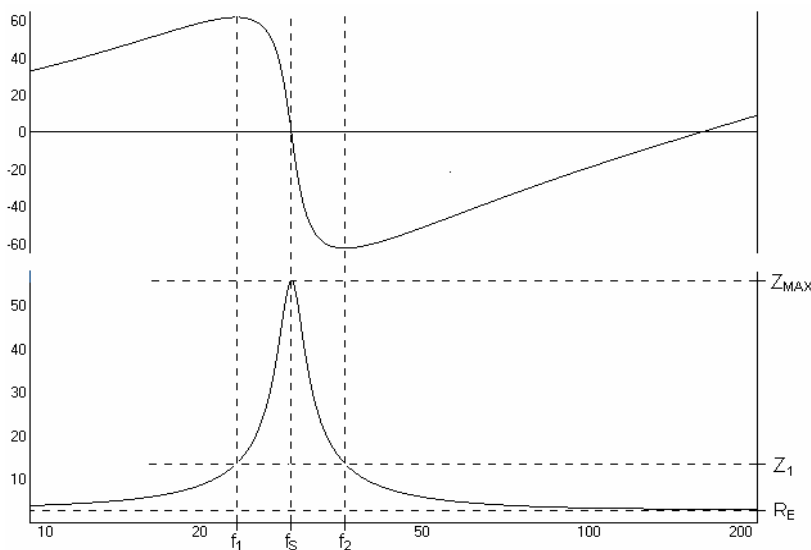
2.2 Impedanssi

Kaiuttimille ja kaiutinelementeille määritellään yleensä nimellisimpedanssi, joka on usein 4 tai 8 ohmia. Käytännön kaiutin ei kuitenkaan edusta ihanteellista, resistiivistä kuormaa, vaan se muodostuu resistanssin lisäksi kapasitanssista ja induktanssista. Tällaista kuormaa sanotaan reaktiiviseksi, ja tällaiselle kuormalle on luonteenomaista sen riippuminen taajuudesta. [1, s. 58.]

Tavanomaisen koteloimattoman bassoelementin impedanssikäyrässä on maksimi resonanssitaajuudella. Tämän jälkeen impedanssi laskee minimiarvoonsa, joka on usein pienempi kuin nimellisimpedanssi. Minimien jälkeen impedanssi jälleen nousee, mikä johtuu puhekelan induktanssista. Kaiuttimen koteloitintapa näkyy myös aina impedanssissa. Lisäksi kaiuttimen puhekela lämpenee, kun siihen syötetään tehoa. Lämpenemisen seurauksena impedanssi nousee. [1, s. 58.]

Yleensä impedanssista puhuttaessa tarkoitetaan impedanssin itseisarvoa. Impedanssiin kuuluu kuitenkin lisäksi vaihekulma, jolla ilmaistaan impedanssin reaktiivista osuutta. Induktiivisella reaktanssilla on positiivinen vaihekulma ja kapasitiivisella reaktanssilla negatiivinen. Kaiuttimessa vaihekulman suuruus vaihtelee taajuuden mukaan riippuen sekä elementistä että kotelosta. Vaihekulmalla on merkitystä koska se saattaa vaikuttaa vahvistimen kykyyn ohjata kaiutinta säröytymättä. [3.]

Kuvassa 1 on esitetty tyypillinen koteloimattoman bassoelementin impedanssikäyttäytyminen. Kuva on muodostettu käyttäen apuna Juha Hartikaisen WinISD-ohjelmaa. Kuvan yläosassa on esitetty impedanssin vaihe asteina ja alaosassa impedanssin itseisarvo ohmeina. Kuvasta nähdään että resonanssitaajuudella f_s impedanssi on maksimissaan ja tällä taajuudella vaihekulma on nolla. Resonanssitaajuuden molemmiin puolin impedanssin vaihe muuttuu rajusti, jonka jälkeen se alkaa taas hitaasti lähestyä nollaa. Resonanssitaajuuden molemmilta puolilta löytyvät taajuudet, joilla impedanssin vaihe käy maksimissaan. Nämä taajuudet on merkitty kuvaan taajuuksiksi f_1 ja f_2 .



Kuva 1. Tyypillinen koteloimattoman bassoelementin impedanssi.

2.3 Parametrit

Kaiutinelementin ominaisuuksiin vaikuttavia tekijöitä ovat esimerkiksi ripustusten jäykkyys, liikkuvien osien massa, puhekelan rakenne sekä magneetin voimakkuus. Kaikkien ominaisuuksien tulee olla mielekkäässä suhteessa toisiinsa, jotta elementin ominaisuudet olisivat tasapainoiset. Tietty sarja näitä ominaisuuksia on määritelty parametreina. Parametrit yhteismitallistavat eri valmistajien elementit ja ovat kotelosuunnittelun laskennan peruste. [2, s. 169.]

Pääasiassa kaiutinelementin piensignaalioston määräävät muutamat parametrit. Näitä ovat R_E , B_l , S_D , C_{MS} , M_{MS} ja R_{MS} . Näistä R_E tarkoittaa puhekelan resistanssia. B_l kuvaa magneetin ja puhekelan tuottamaa voimakeroa. S_D tarkoittaa säteilijän tehollista pinta-alaa ja C_{MS} kuvaa elementin ripustusten joustavuutta. M_{MS} on liikkuvien osien massa mukaan luettuna ilman aiheuttama lisämassa. R_{MS} taas kuvaa ripustusten häviöiden aiheuttamaa vastusta. [4.]

Edellä mainituista parametreista on muodostettu yleisemmin käytössä olevat ns. Thiele/Small-parametrit. Parametrit on nimetty kahden australialaisen tutkijan, Richard H. Smallin ja A. N. Thielen mukaan. Parametrit ovat suhteessa edellisessä kappaleessa mainittuihin mutta ne ovat yksinkertaisempia käsitellä ja niiden kanssa on helpompaa toimia. Nämä parametrit ovat f_s , V_{AS} , Q_{MS} ja Q_{ES} . Näistä f_s on elementin resonanssitaajuus vapaassa tilassa. V_{AS} kuvaa elementin ripustusten joustavuutta ilmoitettuna sitä vastaavana ilmatilavuutena. Q_{MS} ja Q_{ES} tarkoittavat elementin mekaanista ja sähköistä hyvyyslukua resonanssitaajuudella. [4.] [5.]

Parametrien määrittäminen

Kaiutinelementin parametrien määrittämiseen on olemassa erilaisia tekniikoita. Eräs näistä on määrittää parametrit elementin impedanssin perusteella. Esimerkki elementin impedanssivasteesta on esitetty aiemmin kuvassa 1.

Elementin resonanssitaajuus f_s on helppo löytää, sillä se on taajuus jossa impedanssi on maksimissaan. Muiden parametrien laskeminen aloitetaan määrittämällä r_0 maksimi impedanssin ja puhekelan resistanssin suhteeksi eli

$$r_0 = \frac{Z_{MAX}}{R_E}. \quad (1)$$

Tämän jälkeen etsitään resonanssitaajuuden molemmilta puolilta taajuudet f_1 ja f_2 , joilla impedanssi on

$$Z_1 = \sqrt{r_0} * R_E. \quad (2)$$

Taajuuksien kesken tulisi päteä kaava

$$f_s = \sqrt{f_1 f_2}. \quad (3)$$

Nyt voidaan laskea hyvyysluvut Q_{MS} ja Q_{ES} kaavoilla

$$Q_{MS} = \frac{f_s \sqrt{r_0}}{f_2 - f_1} \text{ ja} \quad (4)$$

$$Q_{ES} = \frac{Q_{MS}}{r_0 - 1}. \quad (5)$$

Nämä voidaan yhdistää kokonaishyvyysluvuksi kaavalla

$$Q_{TS} = \frac{Q_{ES} Q_{MS}}{Q_{ES} + Q_{MS}}. \quad (6)$$

Ekvivalenttitilavuus, eli V_{AS} , voidaan määrittää käyttäen massan lisäys -tekniikkaa. Tässä täytyy elementin kartioon lisätä tunnetunkokoinen lisämassa, ja mitata resonanssitaajuus uudelleen. Tämän jälkeen lasketaan liikkuvien osien massa kaavalla

$$M_{MS} = \frac{M_{ADD}}{\left(\frac{f_S}{f_S'}\right)^2 - 1}, \quad (7)$$

jossa M_{ADD} on lisätty massa, f_S on elementin resonanssitaajuus ilman lisämassaa ja f_S' lisämassan kanssa. Lisämassa aiheuttaa resonanssitaajuuden alenemisen. Tämän jälkeen voidaan määrittää C_{MS} kaavalla

$$C_{MS} = \frac{1}{(2\pi f_S)^2 M_{MS}}. \quad (8)$$

Ennen ekvivalenttitilavuuden laskemista, täytyy selvittää säteilijän tehollinen pinta-ala S_D . Tämä voidaan laskea, kun on arvioitu säteilijän tehollinen halkaisija. Yleensä halkaisijaan luetaan kartion lisäksi noin puolet ripustuksen leveydestä. V_{AS} voidaan laskea kaavalla

$$V_{AS} = \rho_0 c^2 C_{MS} S_D^2, \quad (9)$$

jossa ρ_0 on ilman tiheys ($1,21 \text{ kg/m}^3$) ja c on äänennopeus ilmassa (344 m/s). [4.] [5.]

3 SIGNAALIN MUODOSTAMINEN

3.1 Walsh-funktiot

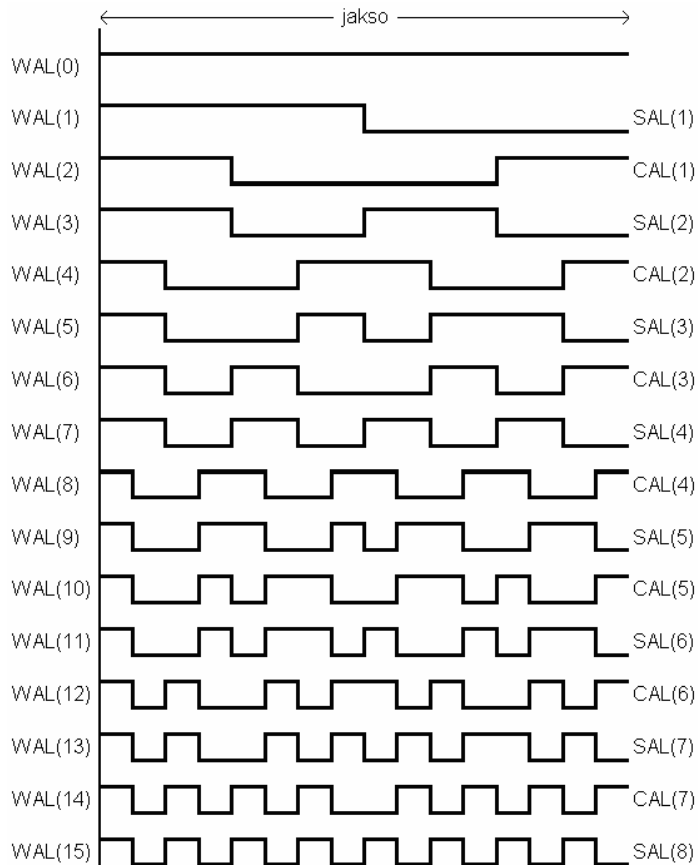
Seuraavassa asiakokonaisuudessa on käytetty lähteenä Benjamin Jacobyn tekstiä [6].

Fourier-sarjojen avulla aaltomuoto luodaan summaamalla tietyn taajuisia sini- ja kosiniaaltoja keskenään. Tästä seuraa herkästi kysymys: onko mahdollista luoda monimutkaisia aaltomuotoja käyttäen digitaalisia, teräväreunaisia aaltoja? Tähän kysymykseen löytyy vastaus Walsh-funktioista, joiden voisi sanoa olevan digitaalinen vastine Fourier-sarjoissa käytetyille sini- ja kosiniaalloille. Walsh-funktioiden avulla voidaan siis muodostaa mikä tahansa jaksollinen aaltomuoto digitaalisten signaalien avulla.

Walsh-funktiot ovat saaneet alkunsa vuonna 1922, jolloin kaksi herraa, J. L. Walsh ja H. Rademacher esittelivät omat kehitelmänsä. Heidän työnsä yhdistettiin vuonna 1929, josta seurasi että Rademacher-funktiot ovat osa Walsh-funktioita, ja jos kaikki funktiot ovat käytössä, mikä tahansa jaksollinen aaltomuoto voidaan muodostaa näitä summaamalla.

Rademacher-funktiot ovat sarja normaaleja kanttiaaltoja. Sarja koostuu perustaajuisesta kanttiaallosta sekä sen harmonisista kanttiaalloista taajuuksilla $2x$, $4x$, $8x$, $16x$, $32x$ ja niin edelleen. Loput Walsh-funktiot ovat hieman monimutkaisempia digitaalisia signaaleja, jotka saadaan muodostettua Rademacher-funktioista xor-logiikan avulla.

Walsh-funktiot ymmärtää helpoimmin katsomalla kuvaa aaltomuodoista. Kuvassa 2 on esitetty 16 ensimmäistä Walsh-funktiota. Kuvasta nähdään, että $WAL(0)$ on vain tasajännitetaso, joten usein se jätetään käyttämättä. Funktiot $WAL(1)$, $WAL(3)$, $WAL(7)$ ja $WAL(15)$ ovat Rademacher-funktioita, eli normaaleja kanttiaaltoja. Muut funktiot ovat enemmän tai vähemmän erikoisia aaltomuotoja. Kannattaa myös huomioida, että funktiot on lisäksi merkitty joko CAL tai SAL . Nämä merkinnät tulevat sanoista ”cosine walsh” ja ”sine walsh”.

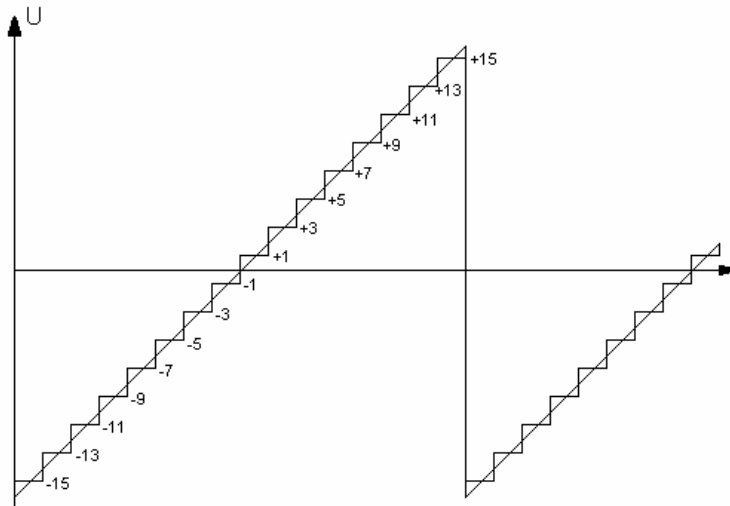


Kuva 2. Ensimmäiset 16 Walsh-funktiota.

Funktiot WAL(1), WAL(3), WAL(7) ja WAL(15) saadaan luotua helposti, sillä ne ovat vain normaaleja kanttiaaltoja. Ne saadaan muodostettua yksinkertaisesti esimerkiksi kiikkupiireillä toteutetulla jakajalla. Muut funktiot ovat monimutkaisempia ja niiden luominen on hieman hankalampaa, mutta muutaman säännön avulla nekin onnistuvat. Säännöt on esitetty liitteissä 1 ja 2. Liitteessä 1 on esitetty sääntöjä ja ohjeita jotka auttavat päättämään mitä funktioita tarvitaan halutun aaltomuodon saavuttamiseksi. Liitteessä 2 on neuvottu kuinka tarvittava Walsh-funktio saadaan luotua helpoiten.

Kun tarvittavat funktiot on saatu luotua, täytyy ne summata yhteen oikeilla painoarvoilla. Painoarvo saadaan käyttämällä summaimessa funktioille laskettua summainvastusta. Laskutoimituksissa kannattaa huomioida, että jos aaltomuoto on parillinen tai pariton, summat täytyy laskea vain signaalin ensimmäiselle puolijaksolle. Tämä johtuu siitä, että summa on tällöin tasan puolet kokoajon summasta.

Kunkin Walsh-funktion painoarvon laskeminen on ehkä helpoiten ymmärrettävissä esimerkistä. Jos halutaan luoda kuvan 3 mukainen lineaarinen ramppi 16 pisteen tarkkuudella, täytyy aalto jakaa 16 osaan. Signaalin amplitudit katsotaan jokaisen kohdan keskeltä.



Kuva 3. Lineaarinen ramppi jaettuna 16 osaan.

Esimerkiksi ensimmäisen osan keskellä signaalin arvo on -15 ja seuraavan osan keskellä -13 jne. Näistä arvoista muodostuu taulukon 1 ylin rivi. Koska tavoiteltava aalto on pariton, voidaan käyttää sääntöä 5, jonka mukaan tarvitsee käyttää vain SAL-funktioita. Tarvittavien funktioiden kohdalle taulukkoon merkitään P tai N, riippuen onko funktio positiivinen vai negatiivinen kyseisessä kohdassa aaltoa. Katsomalla kuvaa 2 havaitaan esimerkiksi että funktio SAL(1) on positiivinen ensimmäisen puolijakson, ja negatiivinen toisen puolijakson. Komponenttien painoarvot ovat laskettuna taulukossa 1.

Taulukko 1. Funktioiden painoarvot.

Funktio	-15	-13	-11	-9	-7	-5	-3	-1	1	3	5	7	9	11	13	15	Summa	Normalisoitu
																		Summa
SAL(1)	P	P	P	P	P	P	P	P	N	N	N	N	N	N	N	N	-128	1
SAL(2)	P	P	P	P	N	N	N	N	P	P	P	P	N	N	N	N	-64	0,5
SAL(3)	P	P	N	N	N	N	P	P	N	N	P	P	P	P	N	N	0	0
SAL(4)	P	P	N	N	P	P	N	N	P	P	N	N	P	P	N	N	-32	0,25
SAL(5)	P	N	N	P	P	N	N	P	N	P	N	P	N	N	P	P	0	0
SAL(6)	P	N	N	P	N	P	P	N	P	N	N	P	N	P	P	N	0	0
SAL(7)	P	N	P	N	N	P	N	P	N	P	N	P	P	N	P	N	0	0
SAL(8)	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	-16	0,125

←————— jakso —————→

Taulukon summat on normalisoitu suurimman mukaan, joten suurin saa arvon 1, ja muut alle 1. Tästä on helppoa laskea funktioille tarvittavat summainvastukset. Jos valitaan ensimmäiselle funktioille esimerkiksi 10 kΩ vastus, saadaan muiden arvot jakamalla tämä kunkin normalisoidulla summalla. Esimerkiksi funktioille SAL(2) ja SAL(4) saadaan summainvastuksien arvoiksi 20 kΩ ja 40 kΩ. Termit joiden summa on nolla, jätetään käyttämättä.

3.2 Korrelaatio

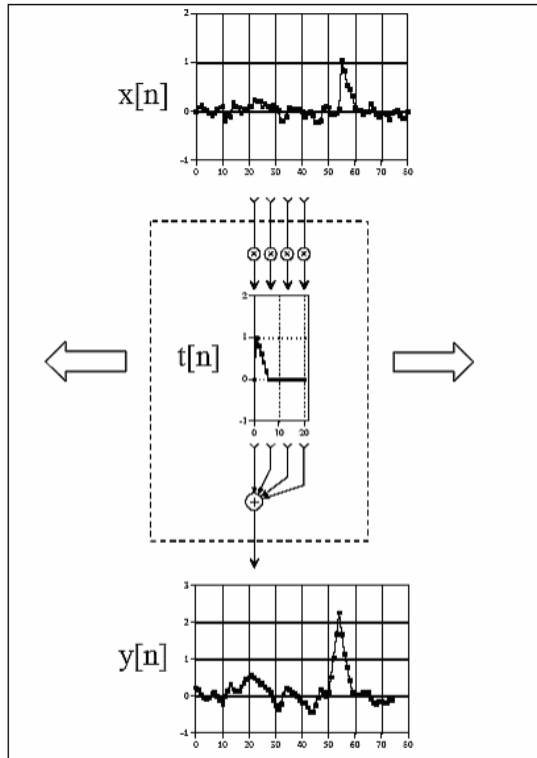
Korrelaatio kuvaa kahden signaalin samankaltaisuutta. Jos kyseessä on kahden eri signaalin korrelaatio, puhutaan ristikorrelaatiosta. Jos taas signaali korreloidaan itsellään, on kyseessä autokorrelaatio. Ristikorrelaation avulla voidaan esimerkiksi etsiä tietyn funktion piirteitä toisesta signaalista. Autokorrelaatiota voidaan käyttää esimerkiksi etsittäessä jaksollisia signaaleja kohinan seasta. [7.]

Korrelaatio on samankaltainen matemaattinen operaatio kuin konvoluutio. Näissä molemmissa muodostetaan kahdesta signaalista kolmas signaali. Kahden signaalin ristikorrelaatio voidaan määrittää yhtälöllä

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x[n]y[n-l],$$

missä $l = 0, \pm 1, \pm 2, \dots$ [9.]

Esimerkki korrelaatiokoneen toiminnasta on esitetty kuvassa 4. Siinä käytetään korrelaatiota etsimään kohdesignaalia vastaanotetusta signaalista. Kuvan esimerkissä vastaanotettu signaali $x[n]$ ja ristikorrelaatio signaali $y[n]$ pidetään kiinteästi paikoillaan. Haettava signaali eli kohde $t[n]$ sijaitsee korrelaatiokoneen sisällä. Jokainen näyte ristikorrelaatio signaalissa lasketaan liikkuttamalla korrelaatiokonetta oikealle tai vasemmalle, kunnes se osoittaa laskettavaa näytettä. Seuraavaksi näytteet vastaanotetusta signaalista menevät korrelaatiokoneeseen, jossa ne kerrotaan vastaavilla näytteillä kohdesignaalin kanssa. Näiden tulojen summa siirtyy kyseiseen kohtaan ristikorrelaatio signaalista. [8.]



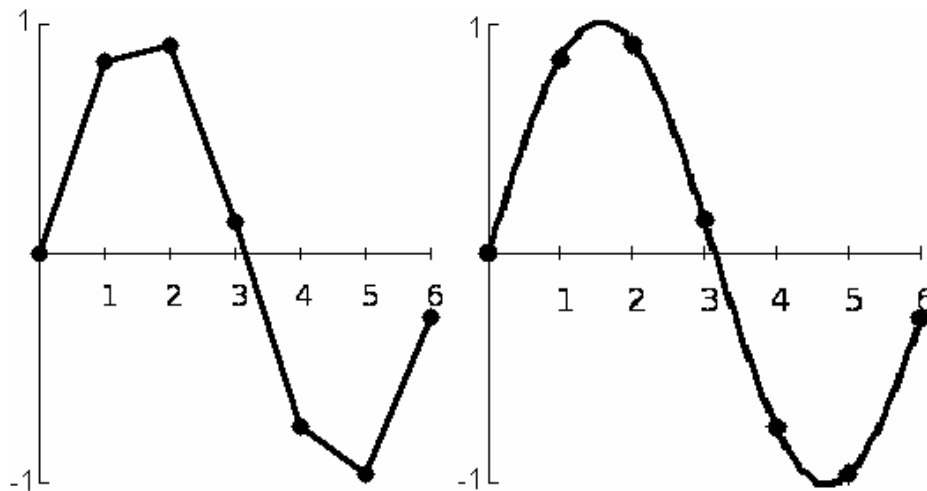
Kuva 4. Korrelaatiokone. [8]

Ristikorrelaatio-signaalissa jokaisen näytteen voimakkuus siis ilmaisee, kuinka hyvin vastaanotettu signaali vastaa kohdesignaalia siinä kohdassa. Tämä tarkoittaa, että ristikorrelaatio-signaalissa on huippu siinä kohdassa, jossa vastaanotettu signaali vastaa parhaiten kohdesignaalia. Ristikorrelaatio-signaalin piikki on symmetrinen, lukuun ottamatta kohinaa. Ristikorrelaatio-signaali ei siis muistuta kohdesignaalia, se vain ilmaisee mistä kohtaa se löytyy. Menetelmä toimii vaikka kohdesignaali sisältäisi negatiivisia arvoja. Korrelaatio on optimaalinen menetelmä etsiessä tunnettua aaltomuotoa kohinan seasta. [8.]

3.3 Interpolaatio ja ekstrapolaatio

Interpoloinnilla ja ekstrapoloinnilla tarkoitetaan matemaattista mittauspisteiden lisäämistä. Interpoloinnissa pisteitä lasketaan jo olemassa olevien pisteiden väliin. Ekstrapoloinnissa pisteitä lasketaan lisää olemassa olevan pistesarjan ulkopuolelle. Ekstrapolointi on periaatteessa matemaattista ennustamista.

Näytteiden lisäämiseen on kehitetty erilaisia algoritmeja. Sopiva algoritmi riippuu signaalista. Yksinkertaisin on lineaarinen interpolointi. Siinä oletetaan että kahden vierekkäisen näytteen välillä muutos on ollut lineaarinen, eli näytteet yhdistetään suoralla. Polynomisen interpolointi on myös yleinen menetelmä. Siinä etsitään pisteisiin vastaava polynomi ja lasketaan halutut pisteet polynomien avulla. Tämän menetelmän huonona puolena on että pistemäärän kasvaessa myös polynomien laskeminen monimutkaistuu. Polynomisen interpoloinnin yksi kehittely on splini-algoritmi, jossa vierekkäiset pisteet yhdistetään myös polynomilla. Kuva 5 on verrattuna lineaarinen interpolointi ja splini-interpolointi. [11.]



Kuva 5. Lineaarinen interpolointi ja splini-interpolointi. [11]

Kuvan molemmissa funktioissa on 7 mitattua pistettä. Pisteiden arvot ovat $f(0)=0$, $f(1)=0,8415$, $f(2)=0,9093$, $f(3)=0,1411$, $f(4)=-0,7568$, $f(5)=-0,9589$ ja $f(6)=-0,2794$. Jos esimerkiksi halutaan tietää funktion arvo kohdassa 2,5, täytyy turvautua interpolointiin. Lineaarista interpolointia käyttäen saadaan laskettua funktion arvo kahden pisteen välillä kaavalla

$$y(x) = y_A + \frac{(x - x_A)(y_B - y_A)}{x_B - x_A}.$$

Eli funktion arvoksi kohdassa 2,5 saadaan

$$y(2,5) = 0,9093 + \frac{(2,5 - 2)(0,1411 - 0,9093)}{3 - 2} = 0,5252. [11.]$$

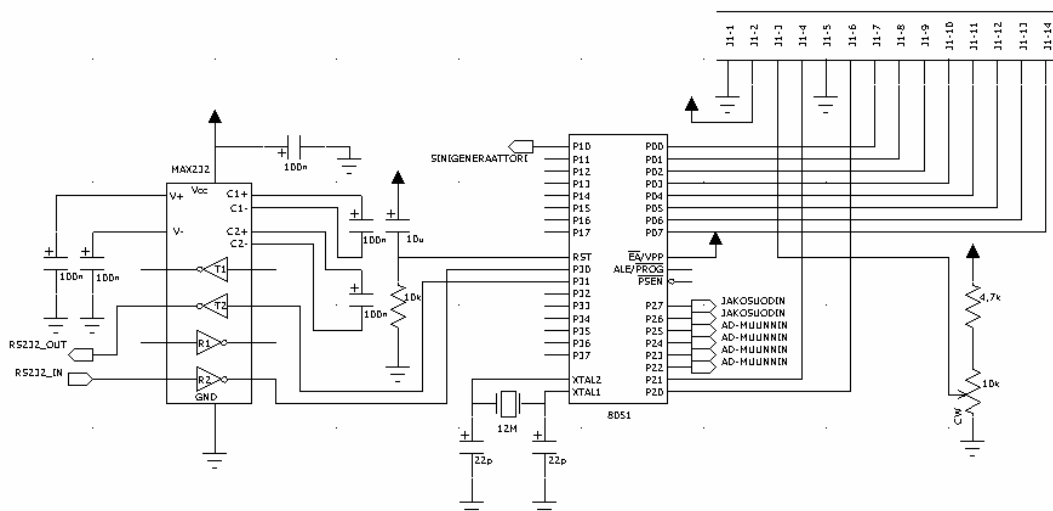
4 OHJAUSYKSIKÖN TOTEUTUS

Laitteen ohjausyksikköön kuuluu mikrokontrollerin lisäksi siihen sarjaportin välityksellä kytketty PC-tietokone. Kontrolleriin on laadittu C-kielinen ohjelma, joka ohjaa siniaalto-generaattoria ja AD-muunninta. PC:llä on LabVIEW-ohjelma, joka syöttää ja lukee tietoa kontrollerilta sekä käsittelee mittaustiedot ja esittää tulokset.

4.1 Mikrokontrollerin toimintaan asettaminen

Mikrokontrollerin toimintaan asettaminen on hyvin yksinkertaista, tarvitaan vain käyttöjännitteet, kide ja muutama oheiskomponentti. Kontrollerin käyttöjännitteet otetaan 5 voltin regulaattorista. Kontrolleriin kytketään 12 MHz kide ja kideä kuormittamaan kaksi 22 pF kondensaattoria. Lisäksi reset-nastaan kytketään RC-piiri, jotta mikrokontrolleri saa lyhyen reset-pulssin aina, kun jännitteet kytketään.

Näyttö kytketään mikrokontrollerin porttiin 0. Näyttö on rinnakkaisväyläinen ja vaatii kahdeksan dataväylää sekä kolme ohjausväylää. Tässä tapauksessa näytön Rd-nasta kytketään suoraan maihin, sillä näytöltä ei tarvitse pystyä lukemaan tietoa. Näytölle kytketään sopiva kontrastijännite trimmerin avulla. Näytön liittäminen mikrokontrolleriin on esitetty kuvassa 6. Näyttö liitetään kuvassa näkyvään 14-pinniseen liittimeen.



Kuva 6. Näytön ja sarjaliikennepiirin kytkeminen kontrolleriin.

Sarjaliikenne toteutetaan RS-232-sarjaportin välityksellä. Käytetyssä kontrollerissa on valmiina sarjaliikennettä varten tarvittavat lähetys- ja vastaanottoportit sekä puskurit ja keskeytykset. Sarjaliikenteen nopeus asetetaan ohjelmallisesti laskurin avulla.

RS-232:n mukaiset jännitteet poikkeavat TTL-tasosta, joten mikrokontrollerin ja sarjaväylän väliin täytyy laittaa jännitetasojen sovitinpiiri. Piirinä käytetään MAX232-piiriä joka muodostaa tarvittavat ± 10 voltin jännitteet 5 voltin käyttöjännitteestä. Piiri tarvitsee toimiakseen käyttöjännitteiden lisäksi viisi ulkoista kondensaattoria. Sarjaliikenteen toteutus, ja sovitinpiirin kytkentä näkyvät kuvassa 6.

4.2 Mikrokontrollerin ohjelmointi

Mikrokontrolleriin laadittiin C-kielinen ohjelma. Ohjelman kuuluu ottaa PC:ltä vastaan mittauksiin liittyvää tietoa, suorittaa mittaus ja lähettää mittaustiedot PC:lle. Ohjelmointi toteutetaan IAR Embedded Workbench –kääntäjällä.

Ohjelman toiminta

Ohjelman alussa alustetaan tarvittavat komponentit kuten LCD-näyttö, sarjaportti sekä puskurit ja keskeytykset. LCD-näyttöä käytetään ilmaisemaan mikä vaihe mittauksessa on menossa. Laskuri 2 alustetaan toimimaan kanttiaaltogeneraattorina. Alustuksien jälkeen jäädytään odottamaan kunnes saadaan PC:ltä tarvittavat tiedot ensimmäisen mittauksen suorittamista varten. Ensimmäisenä otetaan vastaan tieto halutusta jakotaajuudesta. Jakotaajuuksiahan on valittavissa neljä kappaletta ja se valitaan ohjaamalla analogikytkimiä. Kun haluttu jakotaajuus on saatu PC:ltä, asetetaan portit P2.6 ja P2.7 oikeisiin tiloihin, jotta haluttu jakotaajuus saavutetaan.

Seuraavaksi otetaan PC:ltä vastaan tieto halutusta mittaustaajuudesta. Haluttu taajuus saadaan aikaan kontrollerin sisäisen laskurin avulla. Koska laskuri on alussa alustettu toimimaan taajuusgeneraattorina, tarvitsee siihen syöttää vain haluttua taajuutta vastaavat arvot. PC:ltä otetaan vastaan kaksi tavua, jotka syötetään laskurin 2 rekistereihin. Tämän jälkeen laskuri käynnistetään, jonka jälkeen kyseinen taajuus saadaan ulos portista 1.0.

Koska näytteenottotaajuutta täytyy pystyä muuttamaan mittaustaajuudesta riippuen, täytyy PC:ltä saada vielä tieto halutusta viiveestä näytteiden välillä. PC:ltä otetaan vastaan kaksi tavua joilla viive asetetaan.

Seuraavaksi ryhdytään mittaamaan jännitettä AD-muuntimen avulla. Muunnin käyttää sarjamuotoista tiedonsiirtoa, joten lukeminen ja kirjoittaminen täytyy tehdä bitti kerrallaan. Muuntimelle on kytketty kontrollerilta neljä porttia eli kaksi datalinjaa, kello sekä piirinvalinta. Muuntimelle syötetään aluksi tietoa halutusta mittaustavasta sekä käytetty kanava. Nämä tiedot syötetään asettamalla haluttu tieto muuntimen D_{IN} nastaan ja käyttämällä kellonastaa 1-tilassa.

Kun tarvittavat tiedot on syötetty muuntimelle, aloitetaan mittaustiedon vastaanotto. Muunninhan on 12-bittinen, joten täytyy vastaanottaa 12-bittiiä. Bitit luetaan yksi kerrallaan kontrollerin muistiin. Tässä käytetään hyväksi kontrollerissa olevaa bittiosoitettavaa muistialuetta, jonne kukin bitti voidaan asettaa erikseen. Aluksi luetaan 4 ensimmäistä bittiiä. Näiden bittien arvo luetaan muuttujaan, jonka jälkeen luetaan loput 8 bittiiä. Kun kaikki bitit on luettu, lasketaan koko luvun arvo ja asetetaan se muistiin.

Jokaisen mittauksen välissä on taajuudesta riippuva viive. Tämä toteutetaan asettamalla laskurin 0 rekistereihin PC:ltä saadut tavut, käynnistämällä laskuri ja odottamalla kunnes laskuri on käynyt loppuun. Joka mittauksen jälkeen vaihdetaan kanavaa, eli näytteitä otetaan vuoroitellen kanavista 1 ja 4. Mittauksia suoritetaan jokaisella taajuudella 64 kappaletta ja kaikki nämä tulokset kerätään kontrollerin muistiin. Mittaustuloksista 4 ensimmäistä jätetään käyttämättä. Mittauksen jälkeen tulokset lähetetään sarjaportin kautta PC:lle, jonka jälkeen jää odottamaan valmiina seuraavan taajuuden mittausta varten. Ohjelmakoodi on esitetty liitteessä 3.

4.3 LabVIEW-ohjelma

Mittaustulokset käsitellään PC:llä käyttäen LabVIEW-ohjelmointia. Ennen jokaista mittausta ohjelma syöttää mikrokontrollerille sarjaportin kautta mittaukseen tarvittavat tiedot. Näitä ovat laskureiden arvot, joilla määräytyy suodin, taajuus sekä näytteiden välinen viive.

Ohjelman toiminta

Ohjelman alussa luetaan mittaukseen liittyvät tiedot tiedostosta. Tiedosto sisältää jokaiselle mitattavalle taajuudelle oikeat laskureiden arvot. Tiedot ovat tiedostossa, koska tarvittavat laskureiden arvot kullekin taajuudelle on helpointa laskea taulukkolaskennan avulla. Kaikki tiedoston tiedot luetaan kerralla taulukkoon. Tämän jälkeen alustetaan kaikki tarvittavat taulukot ja muuttujat oikeisiin alkuarvoihin. Seuraavaksi luetaan taulukosta ensimmäiseen mittaukseen liittyvät arvot, ja lähetetään ne vuorotellen kontrollerille. Kun kaikki tiedot on lähetetty, odotellaan kunnes kontrolleri lähettää mittaustulokset. Nämä tulokset luetaan taulukkoon.

Koska näytteiden määrä on kohtalaisen alhainen, joudutaan niitä lisäämään laskennallisesti, interpoloimalla. Tätä varten on LabVIEW:ssä valmis funktio. Interpolointi voidaan suorittaa monella eri algoritmilla. Tässä käytetään splini-algoritmia, joka sopii hyvin siniaallolle. Interpoloimalla näytteiden määrä kymmenkertaistetaan.

Seuraavaksi suoritetaan impedanssin itseisarvon laskeminen. Tätä varten täytyy mittaustuloksista selvittää molempien signaalien huipusta huippuun -arvo. Tämä onnistuu etsimällä jännitteen maksimi-arvo ja vähentämällä siitä minimiarvo. Impedanssin itseisarvo lasketaan vertailuvastuksen jännitteen ja kuorman jännitteen huipusta huippuun -arvojen suhteesta kaavalla

$$|Z| = \frac{U_{LOAD} * R_{REF}}{U_{REF}}$$

Impedanssin vaihekulma saadaan selville vertaamalla kuorman ja vertailuvastuksen jännitteen välistä vaihe-eroa. Ensimmäisenä vertailuvastuksen jännitteestä etsitään maksimi- ja minimikohta. Näiden kohtien välimatka on 180 astetta. Tämän jälkeen selvitetään signaalien maksimien välinen etäisyys. Impedanssin vaihekulman laskeminen suoritetaan kaavalla

$$\phi = \frac{(x_{REFMAX} - x_{LOADMAX}) * 180^\circ}{|x_{REFMAX} - x_{REFMIN}|}$$

5 SINIAALTOGENERAATTORIN TOTEUTTAMINEN

Työssä käytetään siniaallon muodostamiseen Walsh-funktioita. Tällä tekniikalla muodostetaan siniaaltoaprosimaatio kanttiaalloista. Saatu signaali syötetään sen jälkeen jakosuotiin, jolla aalto saadaan enemmän muistuttamaan siniaaltoja. Tämän jälkeen signaali vahvistetaan ennen kuormaan syöttämistä.

5.1 Siniaaltoaprosimaatio Walsh-funktioilla

Siniaallon muodostus päätettiin toteuttaa kosiniaallolla ja tarkkuudeksi valittiin 16 osaa. Kosiniaaltohan eroaa siniaallosta vain vaiheessa, joten se käy työhön hyvin.

Aloitetaan suunnittelu käyttämällä hyväksi aaltomuodon parillisuuteen liittyviä sääntöjä. Koska kosiniaalto on parillinen $\frac{1}{2}$ osasta aaltoa katsottuna ja pariton $\frac{1}{4}$ osasta aaltoa katsottuna, voidaan käyttää hyväksi sääntöä 7. Säännön 7 mukaan tällaisen aaltomuodon muodostamiseen tarvitsee käyttää vain parittomia CAL-funktioita. Koska tarkkuudeksi on valittu 16 osaa, käytetään vain 16:ta ensimmäistä Walsh-funktiota. Eli 16-osaisen kosiniaallon muodostamiseen tarvitaan funktiot CAL(1), CAL(3), CAL(5) ja CAL(7).

Seuraavaksi katsotaan liitteessä 2 olevasta taulukosta tarvittavien funktioiden Gray-koodit. Gray-koodia käytetään apuna selvittämään, mitä Rademacher-funktioita tarvitaan kunkin funktion muodostamiseen. Taulukosta havaitaan, että CAL(1)-funktiota varten tarvitaan Rademacher-funktioita WAL(1) ja WAL(3), eli kanttiaallot taajuuksilla 1 x perustaajuus ja 2 x perustaajuus. CAL(3)-funktion muodostamiseen puolestaan vaaditaan Rademacher-funktiot WAL(1) ja WAL(7), eli kanttiaallot taajuuksilla 1 x perustaajuus ja 4 x perustaajuus. CAL(5) funktion muodostamiseen vaaditaan kaikki neljä käytössä olevaa Rademacher-funktiota, ja CAL(7) vaatii funktiot WAL(1) ja WAL(15). Tarvittavat Walsh-funktiot saadaan siis muodostettua yhdistämällä normaaleja kanttiaaltoja toisiinsa xor-logiikalla.

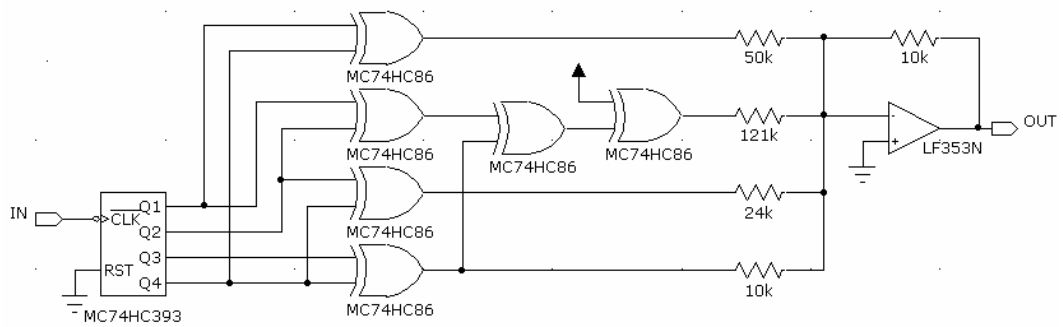
Seuraavaksi täytyy kullekin funktiolle laskea oikea painoarvo. Painoarvo määrää, minkä kokoisen vastuksen kautta funktio yhdistetään summaimeen. Painoarvojen laskemista varten muodostetaan taulukko. Koska kosini on parillinen funktio, tarvitsee painoarvon laskemiseen käyttää vain ensimmäistä puolijaksoa. Aalto haluttiin siis jakaa 16 osaan, joten yhden osan leveys on $22,5^\circ$. Signaalin arvo lasketaan aina kunkin osan puolivälistä, joten ensimmäinen kohta on $11,25^\circ$, toinen $33,75^\circ$ jne. Seuraavaksi tutkitaan kunkin funktionaaltoa ja merkitään taulukkoon, onko funktion arvo kussakin osassa negatiivinen vai positiivinen. Jos arvo on positiivinen, lisätään sen kohdan arvo summaan. Jos se on negatiivinen, niin vähennetään. Taulukossa 2 on esitetty laskutoimitukset.

Taulukko 2. Kosiniaallolle lasketut funktioiden painoarvot.

Funktio	1/2 jakso								Summa	Normalisoitu
	$\cos(11,25) = 0,9808$	$\cos(33,75) = 0,8315$	$\cos(56,25) = 0,5556$	$\cos(78,75) = 0,1951$	$\cos(101,25) = -0,1951$	$\cos(123,25) = -0,5556$	$\cos(146,25) = -0,8315$	$\cos(168,75) = -0,9808$		
CAL(1)	P	P	P	P	N	N	N	N	5,1260	1,0000
CAL(3)	P	P	N	N	P	P	N	N	2,1232	0,4142
CAL(5)	P	N	N	P	N	P	P	N	-0,4224	-0,0824
CAL(7)	P	N	P	N	P	N	P	N	1,0196	0,1989

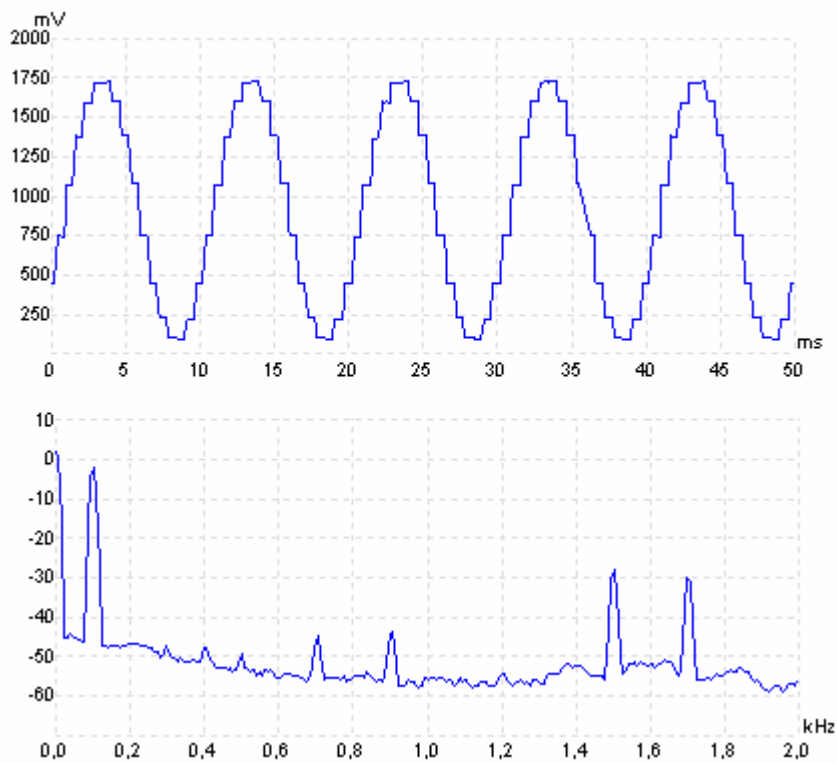
Taulukon 2 perusteella voidaan laskea oikeankokoiset summainvastukset eri funktioille. Valitaan CAL(1)-funktion vastusarvoksi $10 \text{ k}\Omega$ ja suhteutetaan muut vastukset tähän arvoon taulukon normalisoidun summan mukaisesti. Eli arvoiksi saadaan $10,00 \text{ k}\Omega$, $24,14 \text{ k}\Omega$, $121,4 \text{ k}\Omega$ ja $50,28 \text{ k}\Omega$. Taulukosta näkyy että CAL(5)-funktion painoarvo on negatiivinen, joten se täytyy summata käänteisenä muihin verrattuna. Funktio saadaan invertoitua xor-portilla asettamalla toinen tulo 1-tilaan.

Kuvassa 7 on esitetty kytkentä, josta näkyy funktioiden toteutus ja summaaminen. Siinä $16x$ perustaaajuinen kanttiaalto tuodaan 4-bittiseen jakajaan, joka jakaa taajuuden 2, 4, 8 ja 16 osaan. Kuvassa seuraavana on xor-logiikka, jolla toteutetaan halutut funktiot. Funktiot summataan operaatiovahvistimella, johon funktiot kytketään oikeankokoisilla vastuksilla. Kukin vastus on kasattu 1-2 vastuksesta ja arvot on mitattu resistanssimittarilla, jotta arvo saataisiin mahdollisimman tarkaksi.



Kuva 7. Xor-logiikka ja summain.

Kytkenästä ulos saatava aaltomuoto ja sen spektri on esitetty kuvassa 8.

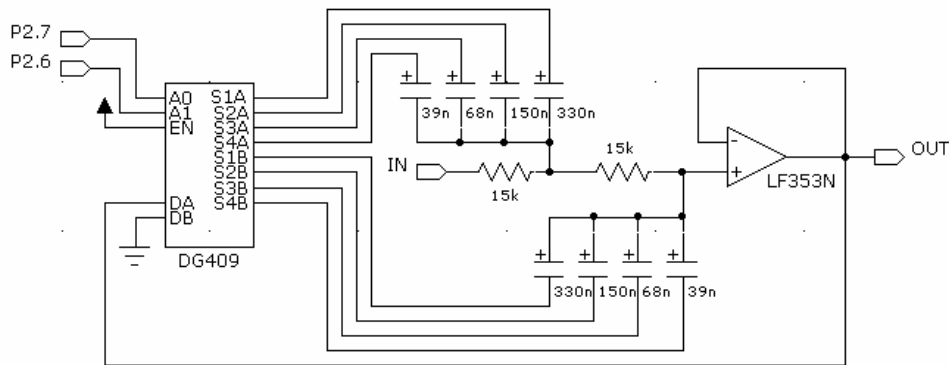


Kuva 8. Siniaalto ja sen spektri.

Summaimelta ulos saatava aalto on varsin hyvää siniaaltoa, kuten kuvan 8 spektriesityksestä nähdään. Silti aalto halutaan muokata muistuttamaan siniaaltoa vielä paremmin. Tämä siksi että myöhemmässä vaiheessa eteen tuleva mittaustulosten käsittely helpottuisi. Tätä varten toteutetaan alipäästösuodatin.

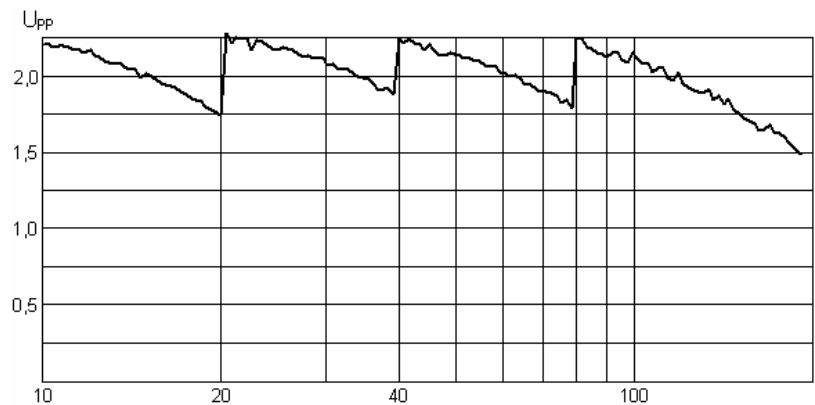
5.2 Suodattimen toteuttaminen

Koska summaimelta saatava siniaaltoaprosimaatio on kantikasta, se syötetään seuraavaksi ohjattavaan jakosuotimeen, jolla aalto suodatetaan muistuttamaan enemmän siniaaltoa. Jakosuotimena käytetään operaatiovahvistimella toteutettua aktiivista 2. asteen suodinta. Suotimen jakotaajuus määräytyy käytetyistä vastuksista ja kondensaattoreista. Käyttämällä kondensaattorin kytkemiseen mikrokontrollerilla ohjattavia analogikytkimiä saadaan jakotaajuutta vaihdettua ohjelmallisesti. Kytkennässä käytetään neljää eri kondensaattoriparia, joten käytävissä on neljä eri jakotaajuutta. Jakosuodin kytkentä on esitetty kuvassa 9.



Kuva 9. Jakosuodin, jossa kondensaattorit valitaan analogikytkimellä taajuuden mukaan.

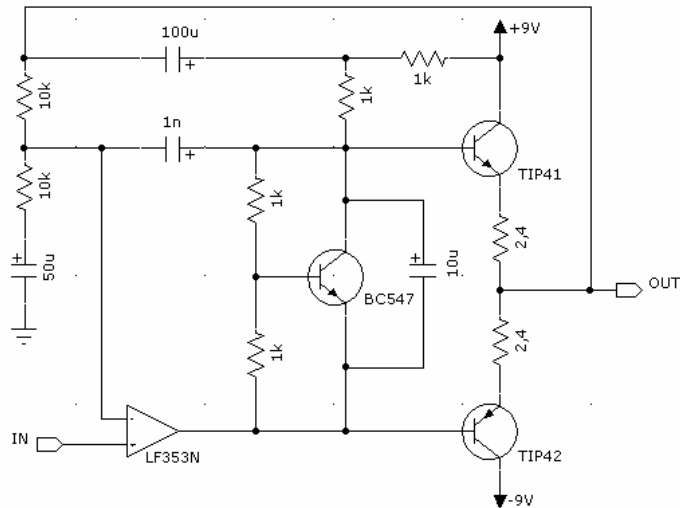
Jakosuotimen kondensaattorit on mitoitettu niin, että suurin kondensaattori on kytkettynä taajuuksilla 10–20 Hz, seuraava kondensaattori taajuuksilla 20–40 Hz, sitten 40–80 Hz ja pienin kondensaattori 80 Hz:stä ylöspäin. Kuvassa 10 on esitetty mittausjännitteen muutos taajuuden suhteen. Kuvasta näkee kuinka suotimen vaihdot vaikuttavat jännitteeseen.



Kuva 10. Jakosuotimen vaikutus mittausjännitteeseen.

5.3 Vahvistimen toteuttaminen

Päätevahvistimeksi toteutettiin yksinkertainen ab-luokan vahvistin. Päätevahvistin tarvitaan, koska operaatiovahvistin ei itse suoraan kykene ohjaamaan kyllin matalaimpedanssista kuormaa. Vahvistimen suunnittelussa on käytetty apuna Harri Honkasen kotisivuja [10]. Kytkentä on esitetty kuvassa 11.



Kuva 11. Vahvistimen kytkentä.

Vahvistimessa päätetransistoreita ohjataan operaatiovahvistimella. PNP-transistoria ohjataan suoraan operaatiovahvistimen lähdöllä. NPN-transistorin ohjausta varten jännitettä nostetaan transistorikytkennän avulla. Vahvistimen lähdöstä on takaisinkytkentä operaatiovahvistimen kääntävään tuloon. Tämä aiheuttaa sen, että lähdön jännite pysyy sisääntulosignaalin mukaisena kuormituksesta huolimatta. Takaisinkytkennän vastuksien suhteella määritetään kytkennän vahvistus, joka on tässä kytkennässä 2.

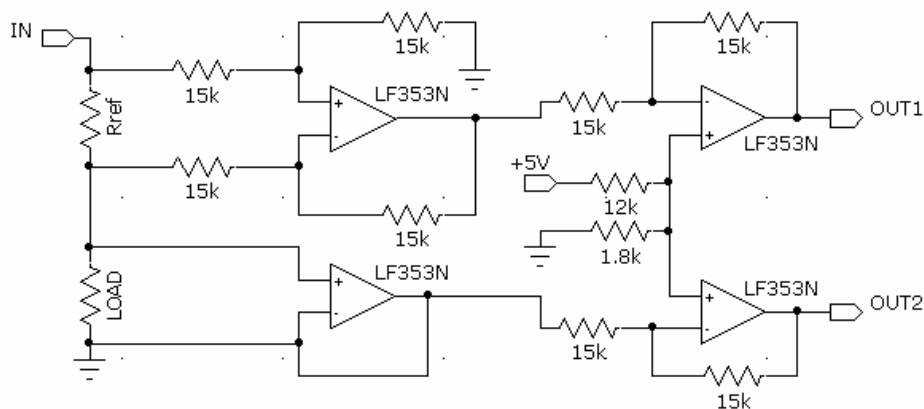
6 MITTAUKSEN TOTEUTTAMINEN

Jännitemittaus suoritetaan 12-bittisellä AD-muuntimella, jota ohjataan mikrokontrollerilla. Muuntimella mitataan jännitettä sekä vertailuvastuksen että mitattavan kuorman yli. Näiden jännitteiden voimakkuutta vertaamalla voidaan laskea impedanssin itseisarvo. Impedanssin vaihekulma saadaan selvitettyä vertailuvastuksen- ja kuorman jännitteen välisestä vaiheerosta

6.1 Mittauskytkentä

Jotta saataisiin selville jännite vertailuvastuksen yli, täytyy käyttää ero- eli differentiaalivahvistinta. Tällainen saadaan toteutettua helposti operaatiovahvistimella. Kytkennässä vertailuvastuksen päissä vaikuttavat jännitteet kytketään samankokoisten vastusten kautta differentiaalivahvistimen sisääntuloihin. Vahvistimen vahvistus asetetaan yhdeksi käyttämällä myös samankokoista takaisinkytkentävastusta. Kuorman yli olevan jännitteen ”poimimiseen” tarvitaan vain normaali operaatiovahvistimella toteutettu puskuri.

Koska vastusten yli vaikuttaa vaihtojännite, ja AD-muuntimen sisääntuloon halutaan vain positiivista jännitettä, täytyy jännitteet nostaa kokonaan positiiviselle puolelle. Tämä onnistuu kytkemällä kääntävän vahvistimen positiiviseen sisäänmenoon tasajännite. Tarvittavat kytkennät on esitetty kuvassa 12.

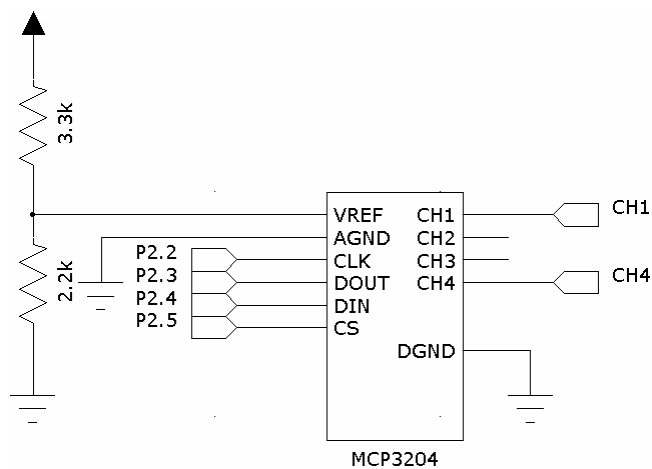


Kuva 12. Jännitteen puskurointi ja tasajännitepisteen nosto.

6.2 AD-muunnin

AD-muuntimena käytetään Microchipin MCP3204-muunninta. Tämä on 12-bittinen ja 4-kanavainen muunnin, jossa on sisäänrakennetut sample/hold-piirit. Muuntimelta tarvitsee kytkeä kontrolleriin 4 väylää. Nämä ovat data sisään, data ulos, kello sekä piirin valinta väylät. Väylät kytetään kontrollerin vapaisiin portteihin. Muunninta ohjataan kontrollerilla ja sillä otetaan näytteitä vuorotellen vertailuvastuksen ja kuorman jännitteistä.

Näytteenottotaajuus on maksimissaan noin 7000 näytettä sekunnissa. Tämä johtuu mikrokontrollerin hitaudesta ohjata AD-muunninta ja kerätä sieltä dataa. Tämä on kuitenkin riittävä nopeus mitattaville taajuuksille. Näytteenottotaajuutta joudutaan hidastamaan matalimmilla taajuuksilla. Tämä johtuu siitä että mikrokontrollerin muisti on rajallinen ja sinne saadaan kerrallaan vain 60 näytteen arvot. Jotta mitattavasta signaalista saataisiin näytteitä ainakin 1 jakson ajalta, on alle 120 Hz:n taajuuksilla näytteenottoa hidastettava. Tämä tapahtuu ohjelmallisesti asettamalla joka näytteenoton väliin taajuudesta riippuva viive. AD-muuntimen kytkeminen on esitetty kuvassa 13.



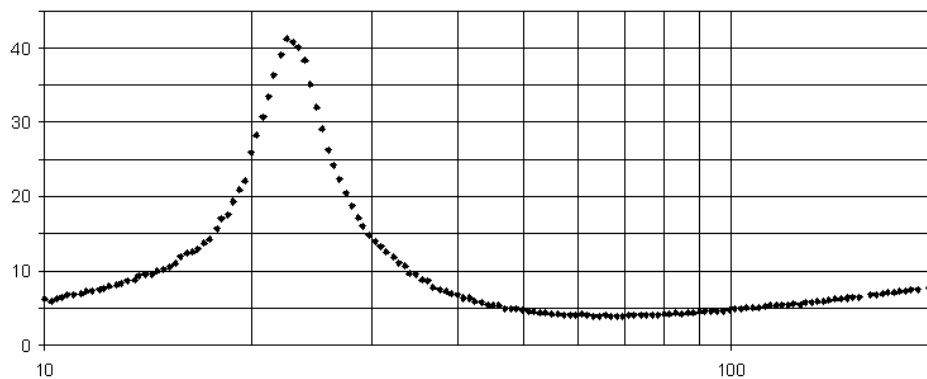
Kuva 13. AD-muunnin kytetään kontrollerin portteihin.

Muuntimella siis otetaan kullakin taajuudella yhteensä 60 näytettä vuorotellen referenssivastuksen ja kuorman yli. Jännitteet on kytketty kanaviin 1 ja 4. Tämän jälkeen ne siirretään sarjaportin välityksellä PC:lle, jossa varsinainen impedanssin laskenta tapahtuu.

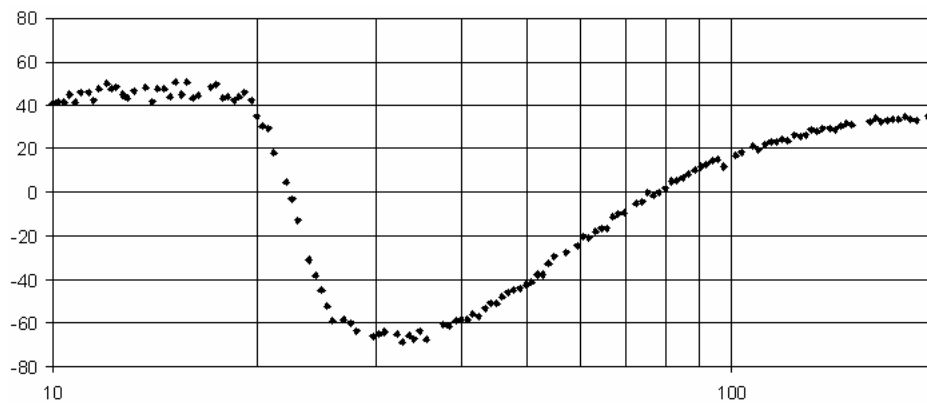
7 LAITTEEN TESTAUS

Laite testattiin mittaamalla subwoofer-elementin impedanssia. Testauksessa käytettiin Focus Acoustics Black 10D2 –elementtiä. Tämä on 10-tuumainen autokäyttöön suunniteltu kaksi-puhekelainen malli. Molemmat puhekelat ovat nimellisimpedanssiltaan 2 ohmia. Testissä puhekelat kytkettiin sarjaan, joten nimellisimpedanssi on 4 ohmia. Elementille ilmoitetut parametrit ovat $f_s = 27$ Hz, $V_{AS} = 32$ l, $Q_{MS} = 6,2$, $Q_{ES} = 0,33$ ja $R_E = 2,7$ Ω .

Testausta varten elementti tuettiin pöydälle vaakatasoon. Testissä mittaustaajuuudet on valittu niin, että taajuutta nostetaan aina 2 prosenttia edelliseen verrattuna. Eli kun ensimmäinen taajuus on 10,0 Hz, on seuraava taajuus 10,2 Hz jne. Taajuudet mitataan sekalaisessa järjestyksessä, eli vierekkäisiä mittaustaajuuksia ei ole mitattu peräkkäin. Mittauksissa käytettiin 20 Ω vertailuvastusta ja mittausjännite on keskimäärin noin 2 Vpp, kuten kuvasta 10 voi todeta. Mitattu impedanssi on esitetty kuvissa 14 ja 15.



Kuva 14. Mitattu impedanssin itseisarvo.



Kuva 15. Mitattu impedanssin vaihe.

Vertaamalla mittaustuloksia teoreettisiin käyriin kuvassa 1, havaitaan että mitattu impedanssi on täysin teorian mukainen. Kuvasta 14 havaitaan kuinka impedanssi käy resonanssitaajuudella huipussaan, laskee sen jälkeen nopeasti ja alkaa tämän jälkeen hitaasti nousemaan. Tämä hidas impedanssin nousu taajuuden kasvaessa johtuu puhekelan induktanssista. Kyseiselle elementille puhekelan induktanssiksi mitattiin 2,8 mH. Kuvista myös havaitaan kuinka impedanssin vaihe muuttuu resonanssitaajuuden lähistössä voimakkaasti. Mittaustuloksien mukaan impedanssin huippuarvo on noin 42Ω taajuudella 23,0 Hz, eli tämä on resonanssitaajuus. Kuvista näkee myös että resonanssitaajuudella impedanssin vaihe on hyvin lähellä nollaa, kuten teorian mukaan kuuluukin.

Testitulosten perusteella voidaan laskemalla arvioida elementin parametrit. Elementin puhekelan resistanssin arvoksi saadaan mittaamalla $2,7 \Omega$, kuten oli ilmoitettukin. Aluksi lasketaan impedanssin maksimiarvon ja puhekelan resistanssin suhde kaavalla 1.

$$r_0 = \frac{Z_{MAX}}{R_E} = \frac{42\Omega}{2,7\Omega} = 15,6.$$

Tämän jälkeen lasketaan impedanssi Z_1 kaavalla 2.

$$Z_1 = \sqrt{r_0} * R_E = \sqrt{15,6} * 2,7\Omega = 10,7\Omega.$$

Mittaustuloksista etsitään seuraavaksi resonanssitaajuuden molemmilta puolilta taajuudet, joilla impedanssi on lähimpänä arvoa Z_1 . Taajuudet ovat $f_1 = 11,5 \text{ Hz}$ ja $f_2 = 33,5 \text{ Hz}$. Tarkastetaan vielä että taajuuksien kesken pätee kaava 3.

$$f_s = \sqrt{f_1 f_2} = \sqrt{11,5 \text{ Hz} * 33,5 \text{ Hz}} = 22,8 \text{ Hz}.$$

Tämän jälkeen saadaan hyvyysluvut laskettua kaavoilla 4, 5 ja 6.

$$Q_{MS} = \frac{f_s \sqrt{r_0}}{f_2 - f_1} = \frac{23 \sqrt{15,6}}{33,5 \text{ Hz} - 11,5 \text{ Hz}} = 5,0.$$

$$Q_{ES} = \frac{Q_{MS}}{r_0 - 1} = \frac{5,0}{15,6 - 1} = 0,34.$$

$$Q_{TS} = \frac{Q_{ES} Q_{MS}}{Q_{ES} + Q_{MS}} = \frac{5,0 * 0,34}{5,0 + 0,34} = 0,32.$$

8 LOPPUANALYYSI

Työssä saatiin toteutettua laite, jolla pystytään mittaamaan impedanssia. Mittarin tarkkuus saatiin riittäväksi, jotta sitä voidaan hyödyntää käytännön mittauksissa. Mittarin taajuusalueeksi saatiin tavoiteltu 10–200 Hz. mikään ei kuitenkaan estä mittaamasta hieman korkeampiakin taajuuksia, tarkkuus tosin kärsii. Tavoitteena oli pystyä mittaamaan impedanssi ainakin 100 Ω asti. Tämä onnistuu helposti, sillä impedanssille ei periaatteessa ole ylärajaa, tosin vertailuvastusta joutuu kasvattamaan kun kuorman impedanssi kasvaa..

Insinööriyön suunnittelu ja toimintaan saaminen oli kohtuullisen haastavaa. Ylitsepääsemättömiä ongelmia ei kuitenkaan ilmennyt, joten työ saatiin toteutettua. Työ sisälsi pääasiassa elektroniikan suunnittelua ja ohjelmointia. Elektroniikan puolella aikaa vaati erityisesti signaaligeneraattorin toteuttaminen. Aluksi siniaallon tuottaminen yritettiin toteuttaa PLL-silmukan avulla, mutta havaittiin että näin matalilla taajuuksilla tekniikkaa on hankala saada toimivaksi. Myös ohjelmoimiseen kului paljon aikaa. Erityisen aikaa vievää oli toteuttaa AD-muuntimen ohjaus mikrokontrollerilla, sillä muunnin käyttää sarjaliikennettä. Haastavaa siinä oli saada ohjelma kyllin nopeaksi suorittaa mikrokontrollerilla, jotta näytteenottonopeus olisi riittävä. Kontrollerin ohjelmoinnissa voitiin kuitenkin suurelta osin käyttää hyödyksi aikaisemmin saatuja laboraatiokokemuksia. PC:lle toteutettu LabVIEW-ohjelma oli myös haasteellinen vähäisen LabVIEW kokemuksen vuoksi. Ohjelma saatiin kuitenkin varsin hyvin toimivaksi.

Työtä tehdessä havaittiin että kaiutinelementin impedanssin mittaaminen on haastavaa. Impedanssi nimittäin riippuu taajuuden ohella mm. mittaussignaalin voimakkuudesta. Esimerkiksi testissä olleen elementin resonanssitaajuus vaihteli käytössä olevan mittausjännitteen mukaan. Käytettäessä mittausjännitettä 1 Vpp taajuudeksi saatiin noin 24 Hz ja jännitteellä 4 Vpp noin 22 Hz. Toteutetussa laitteessa virhettä mahdollisesti aiheuttaa ainakin se, ettei mittausvirta pysy vakiona. Lisäksi testivaiheessa havaittiin että mitattaessa suurikelaista elementtiä, vahvistin alkoi toisinaan värähtelemään. Laitteessa on muutenkin vielä paljon paranneltavaa, mutta se toimii. Insinööriyössä laite toteutettiin vain kytkentäalustalle. Tarkoituksena on kuitenkin lähitulevaisuudessa kasata laite piirilevyille ja koteloida se. Laitetta on tarkoitus pyrkiä hyödyntämään kaiutinkoteloiden suunnittelussa ja kaiuttimien testaamisessa. Impedanssin vaiheen mittausta on myös tarkoitus kehittää tarkemmaksi ja mahdollisesti muokata laitetta niin että mittaussignaalin voimakkuutta voi säätää.

9 YHTEENVETO

Insinööriyössä suunniteltiin ja toteutettiin laite impedanssin mittaamiseen. Mittari on suunniteltu erityisesti kaiuttimelementin impedanssin mittaamiseen, mutta toki sillä voi mitata muunlaisiakin kuormia. Laitteella saadaan mitattua impedanssia taajuusalueella 10–200 Hz, kuten oli tavoitteenakin. Periaatteessa hieman suurempienkin taajuuksien mittaaminen onnistuu, mutta tarkkuus kärsii. Laitteella saadaan mitattua impedanssista itseisarvon lisäksi vaihekulma. Mittaus suoritetaan syöttämällä sinimuotoista signaalia tunnetun kokoisen vertailuvastuksen ja kuorman läpi. Tämän jälkeen saadaan kuorman impedanssi pääteltyä vertailemalla näiden välisiä jännitteitä.

Laitte koostuu ohjausyksiköstä, johon kuuluu mikrokontrolleri sekä PC. Mikrokontrollerina käytetään Atmel AT89S52 –kontrolleria, johon toteutetaan C-kielinen ohjelma. Kontrolleri ohjaa AD-muunninta sekä siniaaltogeneraattoria. Kontrolleri yhdistetään sarjaportin kautta PC:hen, jossa mittaustulokset käsitellään LabVIEW-ohjelmalla. Siniaaltogeneraattori toteutettiin Walsh-funktioilla. Tämä on menetelmä tuottaa haluttua aaltomuotoa digitaalisista signaaleista. Menetelmällä saatiin luotua laadukasta siniaaltoa yksinkertaisesti. Siniaaltogeneraattoriin kuuluu myös ohjattava jakosuodin sekä pienitehoinen päätevahvistin. Jännitemittaus suoritetaan käyttäen 12-bittistä AD-muunninta.

Laitte testattiin mittaamalla subwoofer-elementin impedanssia. Vertaamalla testituloksia teoreettisiin arvoihin, huomataan että teoria ja käytäntö yhtyvät. Testituloksista voidaan päätellä että laite toimii kuten pitää ja se on kyllin tarkka käytännön mittauksiin. Laitteessa on silti vielä paljon paranneltavaa ja kehitettävää.

LÄHTEET

- 1 Esa Blomberg, Ari Lepoluoto, Audiokirja: Audiotekniikkaa ammattilaisille ja kehittyneille harrastajille. Forssan Kirjapaino Oy, 1993, ISBN 951-96379-1-5
- 2 Johan Sandholm, Äänentoistojärjestelmät, Tammer-Paino Oy, Espoo 1994, ISBN 952-9756-10-0
- 3 Pekka Tuomela, Kaiutinelementin parametrit, artikkeli HIFI-lehti 9/93, s. 20–22.
- 4 Richard H. Small, Direct-radiator loudspeaker system analysis. 1971 luettu 17.2.2007 [pdf-dokumentti] <http://ieeexplore.ieee.org>
- 5 Brian E. Anderson, Derivation of moving-coil loudspeaker parameters using plane wave tube techniques. 2003 luettu 8.3.2007 [pdf-dokumentti] <http://contentdm.lib.byu.edu/ETD/image/etd330.pdf>
- 6 Benjamin Jacoby, Walsh Functions: A Digital Fourier Series [pdf-dokumentti]
- 7 Petri Kärhä, Luento 5: Signaalit ja järjestelmät aika ja taajuusalueissa, Helsinki University of Technology, Metrology Research Institute, luettu 10.4.2007 [pdf-dokumentti] http://metrology.hut.fi/courses/s108-180/Materiaali2004/aikataaj_2004.pdf
- 8 DSP guide -verkkokirja, Chapter 7: Properties of convolution, luettu 5.3.2007 [www-dokumentti] <http://www.dspguide.com/ch7/3.htm>
- 9 Lappeenrannan teknillinen yliopisto, sähkötekniikan osasto, Diskreettiaikaisten signaalien korrelaatio luettu 10.4.2007 [pdf-dokumentti] <http://www.ee.lut.fi/courses/080563000/luennot/luento10.pdf>
- 10 Harri Honkanen, AB-luokan audiovahvistimen suunnitteluohjeita. [pdf-dokumentti] <http://gallia.kajak.fi/opmateriaalit/yleinen/honhar/ma>
- 11 Wikipedia, the free encyclopedia, Interpolation luettu 8.4.2007 [www-dokumentti] <http://en.wikipedia.org/wiki/interpolation>

LIITEIDEN LUETTELO

1 OHJEET TARVITTAVIEN WALSH-FUNKTIOIDEN MÄÄRITTÄMISEEN

2 OHJEET WALSH-FUNKTIOIDEN MUODOSTAMISEEN

3 C-KIELINEN OHJELMAKOODI

Ohjeet tarvittavien Walsh-funktioiden määrittämiseen:

1. Tavoiteltavan aaltomuodon täytyy olla toistuva.
2. On suositeltavaa käyttää 2^n määrä askelia, sillä silloin vaadittavien harmonisten määrä on automaattisesti optimi.
3. Jos lasketaan komponenttien arvot suuremmalle tarkkuudelle ja riittää käyttää heikompa tarkkuutta, voidaan tarpeettomat funktiot vain kytkeä pois käytöstä. Tämä johtuu siitä, että painoarvot säilyvät silti oikeina. Mitä symmetrisempää tavoiteltava aaltomuoto on, sitä yksinkertaisempia laskutoimitukset ja kytkennät ovat.
4. Jos tavoiteltava aaltomuoto on parillinen, eli signaalin arvo on sama mentäessä keskipisteestä vasemmalle tai oikealle sama matka, tarvitsee käyttää vain CAL-funktioita.
5. Jos tavoiteltava aaltomuoto on pariton, eli signaalin arvo on käänteinen mentäessä keskipisteestä vasemmalle tai oikealle sama matka, tarvitsee käyttää vain SAL-funktioita
6. Jos aaltomuoto on parillinen ja lisäksi se on parillinen myös $\frac{1}{4}$ kohdasta aaltoa katsottuna, tarvitsee käyttää vain parillisia CAL-funktioita.
7. Jos aaltomuoto on parillinen ja lisäksi pariton $\frac{1}{4}$ kohdasta aaltoa katsottuna, tarvitsee käyttää vain parittomia CAL-funktioita.
8. Jos aaltomuoto on pariton ja lisäksi parillinen $\frac{1}{4}$ kohdasta aaltoa katsottuna, tarvitsee käyttää vain parillisia SAL-funktioita.
9. Jos aaltomuoto on pariton ja lisäksi pariton $\frac{1}{4}$ kohdasta aaltoa katsottuna, tarvitsee käyttää vain parittomia SAL-funktioita.

Ohjeet Walsh-funktioiden luontiin:

1. Funktion $WAL(n)$ luomista varten kirjoitetaan ensin numero n Gray-koodattuna. Gray-koodi on muokattua binäärikoodia, jossa vain yksi bitti muuttuu vierekkäisten lukujen välillä. Gray-koodit 16 ensimmäiselle Walsh-funktiolle on esitetty taulukossa.
2. Määritetään jokaiselle Gray-koodin bitille Rademacher-funktio eli normaali kanttiaalto. Aloitetaan määrittämällä vähiten merkitsevälle bitille $WAL(1)$, seuraavalle $WAL(3)$, seuraavalle $WAL(7)$ jne. Bittien Rademacher-funktiot näkyvät taulukon yläosassa.
3. Jätetään käyttämättä Rademacher-funktiot, joiden Gray-koodin bitti on 0. Ne funktiot, joiden bitti on 1, yhdistetään toisiinsa xor-logiikalla.
4. Kaikkien Walsh-funktioiden tulee alkaa positiivisena, joten joitain funktioita voi joutua invertoimaan.

Taulukko. Gray-koodit 16 ensimmäiselle Walsh-funktiolle.

Funktio	WAL(15)	WAL(7)	WAL(3)	WAL(1)	Funktio
WAL(0)	0	0	0	0	
WAL(1)	0	0	0	1	SAL(1)
WAL(2)	0	0	1	1	CAL(1)
WAL(3)	0	0	1	0	SAL(2)
WAL(4)	0	1	1	0	CAL(2)
WAL(5)	0	1	1	1	SAL(3)
WAL(6)	0	1	0	1	CAL(3)
WAL(7)	0	1	0	0	SAL(4)
WAL(8)	1	1	0	0	CAL(4)
WAL(9)	1	1	0	1	SAL(5)
WAL(10)	1	1	1	1	CAL(5)
WAL(11)	1	1	1	0	SAL(6)
WAL(12)	1	0	1	0	CAL(6)
WAL(13)	1	0	1	1	SAL(7)
WAL(14)	1	0	0	1	CAL(7)
WAL(15)	1	0	0	0	SAL(8)
	Gray-koodi				

```

/* Jyrki Heikkinen, Kaiutinelementin impedanssimittarin ohjelmakoodi*/

/*Käytetyt kirjastot*/
#include<io_a89.h>
#include<string.h>
#include<stdio.h>

*/Määrittelyt*/
#define LCD_data P0          /*Määritetään LCD_data porttiin 0*/
#define EN P2.0             /*Määritetään näytön EN-nasta porttiin 2.0*/
#define RS P2.1            /*Määritetään näytön RS-nasta porttiin 2.1 */
#define CLK P2.2           /*Määritetään muuntimen CLK-nasta porttiin 2.2*/
#define Dout P2.3         /*Määritetään muuntimen Dout-nasta porttiin 2.3*/
#define Din P2.4          /*Määritetään muuntimen Din-nasta porttiin 2.4 */
#define CS P2.5           /*Määritetään muuntimen CS-nasta porttiin 2.5 */
#define SV1 P2.6          /* Määritetään Suotimen Valinta 1 porttiin 2.7*/
#define SV2 P2.7          /* Määritetään Suotimen Valinta 2 porttiin 2.8*/

*/Aliohjelmat*/
void viive(int pituus);      /*1ms viive aliohjelma*/
void LCD_alustus();         /*LCD_alustus aliohjelma */
void LCD_tulostus(char merkit[]); /*LCD_tulostus aliohjelma*/
void sarjaportti_tulostus(char merkit[]); /*sarjaportti_tulostus aliohjelma */
void nayton_nollaus();     /*Näytön nollaus aliohjelma */
void sarjaportin_alustus(); /*Sarjaportin alustus aliohjelma*/
void laskurin_alustus();   /*Laskureiden alustus aliohjelma*/
void jannite_mittaus();    /*jännite_mittaus aliohjelma*/
void mittausten_kasittely(); /*mittausten_kasittely aliohjelma*/

*/Globaalit muuttujat*/
bdata char mittaus;        /*määritetään muuttuja bittiosoitettavaksi*/
int Tulokset[64];         /* Taulukko mittaustuloksille*/
char merkit[8];           /*Merkkitaulukko*/
char lippu;
/*****/
void main()                /*PÄÄOHJELMA*/
{
    laskurin_alustus();    /*Alustetaan laskurit*/
    LCD_alustus();        /*Alustetaan LCD-näyttö*/
    sarjaportin_alustus(); /*Alustetaan sarjaportti*/

    for(;;)                /*Ikuinen silmukka*/
    {
        nayton_nollaus(); /*Nollataan LCD-näyttö */
        sprintf(merkit, "suodin"); /*Kirjoitetaan merkit muuttujaan tekstiä*/
        LCD_tulostus(merkit); /*Tulostetaan teksti LCD-näytölle*/

        /*otetaan seuraavaksi vastaan sarjaportilta jakosuotimen valinta tavu*/
        REN=1;             /* sallitaan vastaanotto */
        ES=1;              /* sallitaan sarjaporttikeskeytys */
    }
}

```

```

lippu=0x00; /*Nollataan lippu*/
while(lippu==0x00); /*odotetaan kunnes lippu muuttuu*/

/*Asetetaan jakosuotimen valinta -portit oikeaan tilaan*/
if(lippu==0x05)
{
    SV1=0;
    SV2=0;
}
else if(lippu==0x10)
{
    SV1=1;
    SV2=0;
}
else if(lippu==0x15)
{
    SV1=0;
    SV2=1;
}
else
{
    SV1=1;
    SV2=1;
}
}
TR2=0; /*Sammutetaan laskuri 2*/
nayton_nollaus(); /*Nollataan LCD-näyttö*/
sprintf(merkit, "taajuusH"); /*Kirjoitetaan merkit muuttujaan tekstiä*/
LCD_tulostus(merkit); /*Tulostetaan teksti LCD-näytölle*/

lippu=0x00; /*Nollataan lippu*/
while(lippu==0x00); /*odotetaan kunnes lippu muuttuu*/
RCAP2H=lippu; /*Asetetaan lippu laskuri 2:sen rekisteriin*/

nayton_nollaus(); /*Nollataan LCD-näyttö*/
sprintf(merkit, "taajuusL"); /*Kirjoitetaan merkit muuttujaan tekstiä*/
LCD_tulostus(merkit); /*Tulostetaan teksti LCD-näytölle*/

lippu=0x00; /*Nollataan lippu*/
while(lippu==0x00); /*odotetaan kunnes lippu muuttuu*/
RCAP2L=lippu; /*Asetetaan lippu laskuri 2:sen rekisteriin*/

TR2=1; /*Käynnistetään laskuri 2*/
ES=0; /*Estetään sarjaporttikeskeytys*/
jannite_mittaus(); /*Siirrytään aliohjelman*/
mittausten_kasittely(); /*Siirrytään aliohjelman*/
}
}
/*****
void mitausten_kasittely()
{

```

```

int i;

printf(merkit,"Alku");          /*Kirjoitetaan merkit muuttujaan tekstiä*/
sarjaportti_tulostus(merkit);  /*Tulostetaan teksti sarjaporttiin*/

for( i=4;i<64;i++ )          /*Silmukassa pyöritään 60 kertaa*/
{
    printf(merkit,"%d", Tulokset[i]); /*Kirjoitetaan merkit muuttujaan tulos*/
    sarjaportti_tulostus(merkit);    /*Tulostetaan tulos sarjaporttiin*/
    Tulokset[i]=0;                 /*Nollataan tulos*/
}

printf(merkit,"Loppu");       /*kirjoitetaan merkit muuttujaan tekstiä*/
sarjaportti_tulostus(merkit);  /*Tulostetaan teksti sarjaporttiin*/
}
/*****/
void jannite_mittaus()
{
    int i;
    int Temp=0;
    int b=0;
    char laskuri_H, laskuri_L;

    ES=1;                       /*Sallitaan sarjaportti keskeytys*/
    nayton_nollaus();           /*Nollataan LCD-näyttö*/
    printf(merkit, "laskuriH");  /*Kirjoitetaan merkit muuttujaan tekstiä*/
    LCD_tulostus(merkit);      /*Tulostetaan teksti LCD-näytölle*/
    lippu=0x00;                 /*Nollataan lippu*/
    while(lippu==0x00);        /*odotetaan kunnes lippu muuttuu*/
    laskuri_H=lippu;           /*Asetetaan lippu muuttujaan*/

    nayton_nollaus();          /* Nollataan LCD-näyttö */
    printf(merkit, "laskuriL"); /* Kirjoitetaan merkit muuttujaan tekstiä */
    LCD_tulostus(merkit);     /* Tulostetaan teksti LCD-näytölle */
    lippu=0x00;               /*Nollataan lippu*/
    while(lippu==0x00);      /*odotetaan kunnes lippu muuttuu*/
    laskuri_L=lippu;         /*Asetetaan lippu muuttujaan*/

    nayton_nollaus();        /* Nollataan LCD-näyttö */
    printf(merkit, "mittaa"); /* Kirjoitetaan merkit muuttujaan tekstiä */
    LCD_tulostus(merkit);   /* Tulostetaan teksti LCD-näytölle */

    ES=0;                   /*Estetään sarjaportti keskeytys*/
    CS=1;                   /*Asetetaan CS-nasta 1-tilaan*/
    CLK=0;                  /*Asetetaan CLK-nasta 0-tilaan*/
    Din=0;                  /*Asetetaan Din-nasta 0-tilaan*/
    viive(1);               /*1ms viive*/

    /*Aloitetaan AD-muuntimella mittaus*/
    for(i=0;i<64;i++)      /*Silmukassa pyöritään 64 kertaa*/

```



```

TMOD=0x21;                                /* Laskureiden toimintamuodon valinta */

/* Alustetaan laskuri2:n rekisterit T2CON ja T2MOD */
T2CON=0x00;
T2MOD=0x02;

/*Baudinopeuden asettaminen, 0xF3 = 2400 baud*/
TH1=0xF3;
TR1=1;                                    /* Käynnistetään laskuri 1 */
}
/*****/
void viive(int pituus)
{
    int i;
    for(i=0;i<pituus;i++)
    {
        TL0=0x18;                          /* Laskurin LSB-bittien alustaminen */
        TH0=0xFC;                          /* Laskurin MSB-bittien alustaminen */
        TR0=1;                              /* Laskurin käynnistäminen */
        while(TF0==0);                     /* Odotetaan että laskurissa tulee ylivuoto */
        TR0=0;                              /* Laskurin Sammutus */
        TF0=0;                              /* Ylivuotolipun nollaus */
    }
}
/*****/
void LCD_alustus()
{
    int i;

    /*LCD-näytölle kirjoitettavat 11 alustustavua*/
    code static char alustus[]={0x30,0x30,0x30,0x38,0x08,0x01,0x07,0x01,0x0F,0x06,0x01};
    viive(16);                              /*16ms viive*/
    for(i=0;i<11;i++)                       /* for-silmukka, jossa pyöritään 11 kertaa*/
    {
        viive(5);                          /*5ms viive*/
        RS=0;                              /*Nollataan RS-nasta*/
        EN=1;                              /*EN-nasta 1-tilaan*/
        LCD_data=alustus[i];               /*Kirjoitetaan kaikki 11 alustusarvoa vuorotellen*/
        viive(1);                          /*1ms viive*/
        EN=0;                              /*Nollataan EN-nasta*/
    }
}
/*****/
void LCD_tulostus(char merkit[])
{
    int i;
    for(i=0;i<strlen(merkit);i++)
    {
        viive(1);                          /*1ms viive*/
        RS=1;                              /*Asetetaan näytön RS-nasta*/
    }
}

```

```

    EN=1; /*EN-nasta 1-tilaan*/
    LCD_data=merkit[i]; / /*Kirjotetaan LCD-näytölle muuttujan sisältö */
    viive(1); /*1ms viive*/
    EN=0; /*Nollataan EN-nasta*/
    if(i==7) /*Tarkistetaan onko näytölle tulostettu 8 merkkiä */
    {
        viive(1); /*1ms viive*/
        RS=0; /*Nollataan näytön RS-nasta*/
        EN=1; /*EN-nasta 1-tilaan*/
        LCD_data=0xC0; /* Kirjoitetaan näytölle rivinvaihtomerkki */
        viive(1); /*1ms viive*/
        EN=0; /*Nollataan EN-nasta*/
    }
}
}
/*****/
void nayton_nollaus()
{
    viive(1); /*1ms viive*/
    RS=0; /* Nollataan näytön RS-nasta*/
    EN=1; /*EN-nasta 1-tilaan*/
    LCD_data=0x01; /* Komento 0x01 tyhjentää LCD-näytön*/
    viive(1); /*1ms viive*/
    EN=0; /*Nollataan EN-nasta*/
}
/*****/
void sarjaportti_tulostus(char merkit[])
{
    int i;
    for(i=0;i<strlen(merkit);i++) /*Pyöritään kunnes kaikki merkit käyty*/
    {
        viive(5); /*5ms viive*/
        SBUF=merkit[i]; /*Lähetetään merkki sarjaportti puskuriin*/
    }
    viive(5); /*5ms viive*/
    SBUF=0x0A; /*Lähetetään Linefeed */
}
/*****/
interrupt void SCON_int() /* Sarjaportti keskeytys aliohjelma */
{
    if(RI==1) /* Tarkastetaan onko vastaanoton keskeytysbitti */
    {
        lippu=SBUF; /*Luetaan puskurin sisältö lippuun*/
        RI=0; /*Nollataan vastaanoton keskeytys bitti*/
    }
    if(TI==1) /* Tarkastetaan onko lähetyksen keskeytysbitti */
    {
        TI=0; /*Nollataan lähetyksen keskeytysbitti*/
    }
}
}

```