

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Elektroniikka

Tutkintotyö

Juha Silén

***PERUUTUSTUTKAN SUUNNITTELU JA PROTOTYYPIN RA-
KENTAMINEN***

Työn ohjaaja: DI Ilkka Tervaoja
Tampere 2006

Tietotekniikka

Elektroniikka

Silén, Juha

Peruutustutkan suunnittelu ja prototyypin valmistus

Tutkintotyö

31 sivua + 17 liitesivua

Työn ohjaaja

DI Ilkka Tervaoja

Huhtikuu 2006

Tiivistelmä

Tässä insinööriyössä selvitetään ultraäänellä toimivan auton peruutustutkan toimintaa ja suunnittelua. Tässä tutkassa etäisyyden mittaamiseen käytetään ultraääntä. Ultraääni ei ole nykyään ainoa vaihtoehto peruutustutkan toimintatavaksi, mutta tällä hetkellä se on yleisin. Ultraääntä käyttävän tutkan ajatus on hyvin yksinkertainen. Pieni kaiutin lähettää taajuutta, joka on tässä tapauksessa 40kHz. Taajuutta lähetetään noin kymmenen jaksoa. Ääni kulkee tiettyä nopeutta (n. 300m/s) eteenpäin ja kimpoaa esteestä takaisinpäin. Sama ääni tulee takaisin kaiuttimelle, joka toimii mikrofonina. Näiden tapahtumien välisestä ajasta lasketaan matka. Suurimpana erona tässä tutkassa verrattuna kaupallisiin tutkiin on se, että lisäominaisuutena tutkaan on lisätty ns. kaistanvaihtotutka. Kaistanvaihtotutkan tehtävä on ”katsoa”, onko auton vieressä toista autoa, kun auto on ns. kuolleessa kulmassa.

Tästä työstä selviävät peruutustutkan toiminta ja rakenne. Toimintaa selvitetään kytkennän ja mikrokontrollerin ohjelman lohkokaavioin. Mikrokontrollerin ohjelman osuus on suuri osa työtä, kytkentä on varsin yksinkertainen.

Peruutustutkan prototyyppi on tarkoitus asentaa autoon, mutta sitä ei vielä tämän työn kirjoittamisen aikana tehty, joten käytännön tuloksia tässä työssä ei tutkasta ole.

Tutkaan on jätetty mahdollisuus lisätä kaksi anturia. Nämä anturit asennettaisiin auton etukulmiin, ja niiden tarkoituksena olisi helpottaa auton etukulmien hahmottamista. Tutkaan on tehty myös liityntä mahdollisille anturien lämmityselementeille. Lämmityksen tarpeellisuudesta ei ole vielä tietoa, sen näkee vasta, kun laitteisto on kytketty autoon. On kuitenkin hyvin mahdollista, että jää anturien päällä heikentää niiden toimintaa, jollei kokonaan estä sitä, ja talvioloissa tutkan toiminta on ehkä tärkeämpää kuin kesällä. Kaistanvaihtotutkan tarpeellisuus on vielä arvoitus. Voi hyvin olla, että siitä ei tulla koskaan saamaan sitä hyötyä, kuin on kuviteltu.

Computer science

Electronics

Silén, Juha

Parking radar design and prototype making

Engineering Thesis

31 pages + 17 appendices

Thesis Supervisor

Ilkka Tervaoja (MSc)

April 2006

Abstract

This engineering thesis is about parking radar and shows main principles about how it works. Because of new car design cars are bigger and more aerodynamics. That makes driving backwards harder. That is one reason why new cars are often equipped with parking radar. This parking radar uses ultrasonic for measurement. Ultrasonic is the most used technology commercially. This parking radar has four ultrasonic transducer to rear and two to sideways. Four rear radars give a good back coverage. Two sideway radars are designed to protect cars driving lane change. Lane changing protection is not yet commercially used. This thesis focuses on design of the electronics and describes software to build a prototype. Prototype is build and testing is started during this project. This thesis doesn't include information about testing and use experiments of the equipment. Appendices include drawings of printed circuit board (PCB) and full schematics also source codes written on c-language.

Alkusanat

Tämän työn tarkoitus on kertoa autonperuutustutkista ja esittää suunnitelma peruutustutkasta. Tässä työssä ei ole paneuduttu kovin syvällisesti ultraäänen fysikaalisiin ominaisuuksiin, vaan on pyritty pysymään sellaisessa tietomäärässä, joka riittää peruutustutkan toiminnan ymmärtämiseen.

Peruutustutka on omaa suunnittelua. En sitten tiedä, kuinka paljon se eroaa kaupallisista malleista. Tiettyjä kytkentöjä en ole itse keksinyt, vaan ne ovat yleisesti tiedossa olevia. Tämä työ sisältää täydelliset kytkennät, piirilevy kuvat ja ohjelman lähdekoodin liitteenä. Liitteenä siis ovat täydelliset tiedot, miten tämä laite toimii ja kuinka sellaisen voi tehdä.

Minulle oli tämän työn kirjoittamisen aikana hyvä onni, kun pääsin töihin sellaiseen yritykseen, jossa oli meneillään juuri ultraääniprojekti. Tällä minun omalla pienellä ultraääniprojektillani saattoi olla jopa merkitystä sille, että yleensä pääsin sinne töihin. Voisi ehkä jopa sanoa, että minulle päättötyön teosta saattoi olla jopa hyötyä.

Kiitos kaikille, jotka ovat tämän työn valmistumista edistäneet ja peruutustutkaa apuna ideoineet.

Tampereella 28. huhtikuuta 2006

Juha Silén

Käytetyt termit ja lyhenteet

LED: Light-Emitting Diode. Siinä puolijohteen P-N-rajapinta alkaa hehkua, kun siihen johdetaan sähkövirtaa. Käytetään jollain tavalla lähes jokaisessa elektroniikkalaitteessa, jossa on merkkivalo. Se on alkanut korvata hehkulamppuja niin taloissa kuin autoissakin. /1/

PIC: Microchipin valmistama mikrokontrolleri. Peripheral Interface Controller. Alun perin PIC oli lyhenne sanoista Programmable Intelligent Computer. Tässä työssä puhutaan PIC:stä mikrokontrollerina, mikä tarkoittaa mikropiiriä, johon on yhdistetty mikroprosessori ja muistit. PIC:eissä on tämän lisäksi myös paljon muita ominaisuuksia kuten AD-muuntimia ja erilaisia sisäisiä tiedonsiirtoprotokollia. /2/

Pietzosähköisyys: Pietzosähköisen elementin molekyylit kääntyvät ulkoisen sähkövirran suuntaisiksi ja tämä aiheuttaa elementin muodonmuutoksen (kaiutin). Sama toimii myös toisinpäin. Kun elementti saadaan tärisemään tai sitä puristetaan, se synnyttää jännitteen elementin napoihin (mikrofoni). Näitä kumpaakin ominaisuutta käytetään tässä työssä.

Assembly: Ns. konekielinen ohjelmointikieli, jossa käytetään suoraan prosessorin ymmärtämiä käskyjä eli lähinnä muistiin osoituksia ja hyppykäskyjä. Assembly-kieliset käskyt vaihtelevat eri prosessoreissa. Tässä työssä käytetyssä PIC:ssä on 35 assemblerkäskyä. Assembly-kielen kääntäjää kutsutaan assebleriksi.

Dobbler-ilmiö: Dobbler-ilmiöksi sanotaan taajuuden muuttumista, mikäli kuuntelija ja taajuuslähde liikkuvat eri nopeudella. Tämän ilmiön huomaa käytännössä silloin, kun auto ajaa ohitse, lähestyvän auton ääni on korkeampi kuin loitontuvan. Paikallaan olevan auton lähettämän äänen taajuus paikallaan on näiden keskiarvo. Tähän perustuvat poliisien nopeuden valvontatutkat. Laskukaava alla (Kaava 1 Dobbler-ilmiö).

$$\frac{f_L}{v + v_L} = \frac{f_H}{v + v_H} \quad \begin{array}{l} f_L, f_H = \text{Lähetetty, Vastaanotettu taajuus} \\ v_L, v_H = \text{Lähetetty, Vastaanotettu nopeus} \\ v = \text{aaltoliikkeen nopeus} \end{array}$$

Kaava 1 Dobbler-ilmiö

A/D-muunnos: A/D-muuntaminen tarkoittaa toimenpidettä, jossa analoginen signaali muunnetaan digitaaliseksi ”0”:ksi ja ”1”:ksi. Tässä työssä käytetään 10-bittistä muunnosta eli 10:n ”0”:n tai ”1”:n sarjaa. Näin ollen 5 voltin jännite voidaan jakaa 2^{10} :n osaan, desimaaleina määrä on 1024. Yksi askel muunnoksessa vastaa jännitteenä 4,9mV.

MOSFET: Metal Oxide Silicon Field Effect Transistor. Voidaan käyttää myös nimitystä IGFET eli eristehilakanavatransistori. MOSFET on tällä hetkellä eniten käytetty transistori, lähes kaikki mikropiirit sisältävät MOSFET-piirejä (CMOS). MOSFET:eja käytetään tässä työssä kyt-

kiminä, johon ne soveltuvat hyvin pienen sisäisen vastuksensa takia. Vastus vaihtelee 10Ω :sta muutamaan milliohmiin. Tässä työssä käytetään avaustyyppin N- ja P-tyypin MOSFET:ejä. N-kanavainen MOSFET avautuu silloin, kun ohjausjännite (hilajännite) on suurempi kuin ohjattava (kollektori) jännite. P-kanavainen MOSFET toimii päinvastoin.

MSB / LSB Most / least significant bit. Eniten / vähiten merkitsevä bitti. Ne kertovat, miten useamman kuin yhden bitin luvut on muodostettu.

SPI Serial Peripheral Interface. Yksinkertainen synkronoitu (kellotettu) sarjaliityntä, jossa on isäntälaitte (master) ja yksi tai useampi passiivinen laite (slave). Siirto tehdään isäntälaitteen kellotuksen tahdissa. SPI-väylässä voidaan tietoa siirtää molempiin suuntiin samanaikaisesti tosin erilinjaa pitkin. SPI-väylää käyttävät useimmat AD / DA-muuntimet ja sillä on helppo ohjata siirtorekisteripiirejä.

Sisällysluettelo

TIIVISTELMÄ	I
ABSTRACT	II
ALKUSANAT	III
KÄYTETYT TERMIT JA LYHENTEET	IV
SISÄLLYSLUETTELO	VI
1 JOHDANTO	1
2 YLEISTÄ	2
2.1 C-kääntäjä	2
2.1 Peruutustutkat	2
3 MITTAAMINEN	3
3.1 Erilaisia mittausmenetelmiä	3
3.2 Ultraääni.....	3
3.3 Äänennopeus	4
4 PERUUTUSTUTKA	5
4.1 Ominaisuudet	5
4.2 Peruutustutkan lohkokaavio.....	6
4.3 Tutkamoduuli.....	7
4.3.1 UÄ-anturien liityntä	8
4.3.1.1 Anturin valinta	8
4.3.1.2 Lähetyksen päteaste	8
4.3.1.3 Kaiun vastaanotto	9
4.3.2 Mikrokontrolleri	11
4.4 Näyttömoduuli.....	12
4.4.1 LCD-näyttö	13
4.4.1.1 Toiminta	13
4.4.2 Sarjaliikenneprotokolla	14
4.5 Lämpötilan mittaus.....	14
5 OHJELMA	16
5.2 Ohjelman rakenne	16
5.3 Tutka	18
5.3.1 Peruutustutka.....	18
5.3.2. Kaistanvaihtotutka	19
5.3.2 Matkan mittaaminen.....	20
5.4 Näyttö	21
5.4.1 Näytönohjauksen toimintakaavio	21
5.4.2 Näytön alustus	22
5.4.3 Merkin tai ohjauskäskyn lähetys	23
5.5 Lämpötilan laskenta	23
6 KÄYTTÖLIITTYMÄ	25

7 RAKENTAMINEN	26
7.1 Piirilevy	26
7.2 Komponentit.....	26
7.3 Kotelointi	26
7.4 Liitännät	26
8 ASENNUS.....	27
8.1 Mekaaninen asennus.....	27
8.1.1 Anturin asennus.....	29
8.1.2 Näyttömoduulin asennus	29
8.2 Sähköasennus	29
8.3 Asennus autoon	29
9 YHTEENVETO	30
LÄHDELUETTELO	31
LIITTEET	A
<i>Liite 1 CD4051BC datalehti</i>	<i>a</i>
<i>Liite 2 TS925 datalehti.....</i>	<i>b</i>
<i>Liite 3 AD5220 datalehti.....</i>	<i>c</i>
<i>Liite 4 74HC595 sisäinen kytkentä</i>	<i>d</i>
<i>Liite 5 Lämpötilanmittaus taulukko</i>	<i>e</i>
<i>Liite 7 Tutkamoduulin kytkentäkaavio</i>	<i>g</i>
<i>Liite 8 Näyttömoduulin kytkentäkaavio.....</i>	<i>h</i>
<i>Liite 9 Piirilevyjen osasijoittelukuvat</i>	<i>i</i>
<i>Liite 10 Piirilevyn syövytyskuva yläpuoli.....</i>	<i>j</i>
<i>Liite 11 Piirilevyn syövytyskuvat alapuoli</i>	<i>k</i>
<i>Liite 12 Komponenttilistaus</i>	<i>l</i>
<i>Liite 13 Tutkamoduulin kotelo</i>	<i>m</i>
<i>Liite 14 Näyttömoduulin kotelo.....</i>	<i>n</i>
<i>Liite 15 Tutkamoduuli</i>	<i>o</i>
<i>Liite 16 Peruutustutkan lähdekoodi</i>	<i>p</i>

1 Johdanto

Nyky aikaisten autojen muotoilun ja koon takia on autoihin alettu lisätä peruutustutkia, jotka kertovat takana olevista esteistä. Peruutustutkia näkee uusissa autoissa yhä enemmän, etenkin niin sanotuissa paremman luokan autoissa, kuten katumaasureissa. Peruutustutka ei ole uusi keksintö, niitä on ollut autoihin saatavissa jo runsaat kymmenen vuotta. Peruutustutka on järkevä hankinta auton lisävarusteeksi, koska auton korjaaminen on kallista ja peruutusvahinko sattuu helposti. Peruutustutka maksaa uuteen autoon noin 200€ ja sen saa lähes joka autoon. Peruutustutkia saa myös tarvikkeina, silloin ne maksavat 50 - 300€ ja lisäksi tulee asennuskulut.

Peruutustutkaa kutsutaan myös parkkitutkaksi, tämä nimi kertoo tutkan tärkeimmän käyttötarkoituksen. Peruutustutka ostetaan lähinnä helpottamaan ns. ”taskuun” pysäköimistä. Isoista autoista ja myös joistain pienemmistäkin on todella vaikea arvioida matkaa takana olevaan autoon ja juuri tämän matkan mittaamiseen tutkaa käytetään. Tutka kertoo matkan äänenä, jolloin kuljettaja voi keskittyä auton ajoon ja muihin suuntiin katsomiseen. Useissa kalliin hintaluokan autoissa, joissa on peruutustutka, on myös vastaava tutka eteenpäin. Tällöin tutkien ultraäänianturit on sijoitettu eritavoin. Ulompi pari on asennettu aivan etukulmiin, se helpottaa taskuun pysäköinnin suoristusvaihetta.

Tässä peruutustutkassa on myös näyttö, joka näyttää tarkan matkan esteeseen jokaisesta anturista. Sen tarkoitus on kertoa kuljettajalle, millä puolella autoa este on. Tätä toimintoa ei yleensä peruutustutkissa ole. Useissa tutkissa on jonkinlainen näyttö. Joskus se on LED-valopalkki tai hienommissa tutkissa näyttö on rakennettu taustapeiliin. Tämänkin laitteen näyttö on suunniteltu asennettavaksi lähelle taustapeiliä, jotta samalla silmäyksellä näkee niin peilin kuin mittaustuloksen.

Tämän peruutustutkan erikoisuus on kaistanvaihtotutka, joka kaistaa vaihdettaessa kertoo, jos ns. kuollessa kulmassa on auto. Tätä toimintoa ei juuri ole missään autossa vielä, mutta lähitulevaisuuden autosta tämäkin todennäköisesti löytyy.

2 Yleistä

2.1 C-kääntäjä

Tässä työssä käytetään mikrokontrollerina Microchipin valmistamaa PIC-mikrokontrolleria. Kääntäjän tarkoitus on ”kääntää” ohjelmakoodi merkeiksi, joita mikrokontrolleri ymmärtää. PIC:lle voidaan kirjoittaa ohjelmat konekielellä eli assemblylla. 16-sarjan Pci:ssä ei ole kuin 35 assembly-käskyä, joten assembly-ohjelmointi ei ole aivan mahdollisimman vaikeata. Mikäli ohjelma ei ole erityisen nopeuskriittinen, on syytä käyttää ns. korkeamman tason ohjelmointikieliä. Pic:lle löytyy kääntäjä lähes kaikista perusohjelmointikielistä. Eniten löytyy C- ja Basic-kääntäjiä, niitä löytyy useilta eri valmistajilta. Valitsemani kääntäjä on CCS:n valmistama C-kääntäjä. Tämän kääntäjän valinnan syyt olivat lähinnä kääntäjän edullisuus verrattuna muihin ja monipuolisuus. CCS valmistaa myös erilaisia testialustoja, ohjelmointi- ja debuggaus-laitteita. Korkeamman tason kielet eivät kokonaan poista assemblerikielen osaamisen tarvetta, jotkin asiat on helpompi tehdä suoraan assemblerina. Myös äärimmäistä nopeutta vaativat asiat kirjoitetaan usein assemblerina.

2.1 Peruutustutkat

Ultraääniantureita käyttävä peruutustutka ei ole ainoa mahdollinen vaihtoehto peruutustutkan toimintatavaksi. Toinen käytetty tapa on kamera. Kameran käyttö on yleistynyt GPS-laitteiston mukana. GPS-paikannin tarvitsee autoon suhteellisen suurikokoisen näytön ja samaan näyttöön saa usein liitettyä kameramoduulin. Kameramoduulissa on tietyt hyvät ja huonot puolet verrattaessa sitä ultraäänellä toteutettuun tutkaan. Seuraavassa on listattu kameramoduulin hyvät ja huonot puolet suhteessa ultraäänitutkaan.

Hyviä puolia:

Tiedon luotettavuus:

Kuljettaja näkee konkreettisesti esteen ja voi siitä arvioida esteen laadun kuten korkeuden, esim. kadun reunakynnyksen, josta ultraäänitutka voi jo hälyttää.

Optimaalinen ajolinja:

Käytössä lähinnä taskuun pysäköinnissä. Näytössä voi olla ajolinja, jota noudattaen auto menee hyvin kadun reunaan.

Näyttömoduulin muu käyttö:

Kameran kuvaa voidaan käyttää esimerkiksi silloin, kun halutaan ajaa vetokoukku mahdollisimman lähellä peräkärryn aisaa.

Huonoja puolia:

Likaantuminen:

Kameramoduuli on paljon herkempi lialle kuin ultraäänianturi. Sen vuoksi kameramoduuli asennetaan mahdollisimman korkealle. Jäätynäänä kumpikaan anturi ei toimi oikein, mutta kun ultraääniantureita on useita niin voi olla, että jokin niistä toimii.

Käyttö:

Näytön katselu lisää kuljettajan tehtäviä, koska pelkän kuvan mukaan on vaikeata peruuttaa. Peruutustutkassa voi olla hahmontunnistus ominaisuus, joka sitten havaitsee esteen ja hälyttää siitä. Se kuitenkin tekee tutkasta monimutkaisen ja kalliin.

3 Mittaaminen

Tässä kappaleessa käydään läpi erilaiset tavat mitata etäisyyksiä sähköisesti, ja tutustutaan ultraääneen.

3.1 Erilaisia mittausmenetelmiä

Erilaisia etäisyyden mittausmenetelmiä vertailtaessa menetelmän valinta perustuu siihen, mikä on haluttu mitattava matka. Menetelmät perustuvat radio-, ääni- tai valoaltojen mittaamiseen. Parhaisiin tuloksiin päästään, mikäli pystytään yhdistämään eri menetelmät.

Radioaalto:

Radioaaltomittaus on eniten käytetty menetelmä, johon tutkat perustuvat. Radiolähetin lähettää hyvin suunnatulla antennilla radiopulssin ja mittaa sitten ajan, joka on lähetyksestä kaikuun. Matka voidaan mitata myös vastaanotetun kaiun vaiheensiirtona. Sitten yleensä lähetin kääntyy vähän ja lähettää uuden pulssin. Tällaista menetelmää käytetään laivoissa ja esim. lentokentillä. Radioaaltomittausta käytetään myös poliisitutkissa, jotka mittaavat nopeuksia. Tosin silloin ei lasketa etäisyyksiä vaan **Dobbler-ilmioita**. Radioaalloilla käytettyä mittausta voidaan käyttää vain pitkillä etäisyyksillä, koska lyhyet etäisyydet vaatisivat äärettömän korkeat prosessointinopeudet.

Valoaalto:

Valolla etäisyyttä mitattaessa käytetään lähinnä infrapunaa tai laseria. Etäisyydenmittaus perustuu kuluaikaan ja / tai vaiheensiirtoon. Mitattava etäisyys voi olla senttimetristä aina useisiin kilometreihin. Infrapunaa käytetään yleisesti erilaisissa läheisyysmittauksissa, joissa mitataan aivan lyhyitä matkoja ja mittausmenetelmä perustuu lähinnä kolmiomittaukseen. Infrapuna-purske lähettää tietystä kulmassa ja valo törmää esteeseen ja kimpoaa takaisin. Vastaanottimen asettelulla säädetään haluttava mittausmatka. Tällä ei saada kovinkaan hyvää resoluutiota mittaukseen, resoluutio paranee vastaanottimia lisäämällä. Lasermittaus perustuu hyvin samaan asiaan kuin radioaaltomittauskin. Lasermittausta käytetään lähinnä maanmittauksessa ja rakentamisessa sen hyvän suunnattavuuden takia.

Ultraääni:

Ultraäänessä kuluaika on pitempi verrattuna näihin muihin vaihtoehtoihin. Tämä tarkoittaa sitä, että lähetys- ja vastaanottoelektroniikka on muita yksinkertaisempi. Ultraäänellä voidaan ilmassa mitata etäisyyksiä 1cm:stä...15 metriin. Ultraääni on tähän sovellukseen parhaiten soveltuva mittaus tapa. Ultraäänestä kerrotaan lisää seuraavassa luvussa. /3/

3.2 Ultraääni

Ultraääneksi sanotaan ääntä, jonka taajuus ylittää ihmiskorvan kuuloalueen eli 20 kHz ja korkeimmillaan se on 1 GHz:n luokkaa. Ultraääni, niin kuin ”tavallinenkin” ääni, tarvitsee aina väliaineen, toisin sanoen ääni ei kulje tyhjiössä. Ultraääni kulkee niin kaasussa, nesteessä kuin kiinteässäkin aineessa. Ultraääni etenee niin poikittain, pitkittäin kuin pinta-aaltonakin. Ultraäänien etenemistapa ja nopeus riippuu väliaineesta. Voisikin yleistää, että mitä tiheämpi aine, niin sitä nopeammin ääni

etenee. Kaasuissa ja nesteissä ultraääni etenee vain pitkittäisenä aaltona. Kiinteissä aineissa ääni etenee sekä pitkittäisenä että poikittaisena aaltona. Kiinteän aineen ja kaasun tai nesteen rajapinnassa etenee myös pinta-aalto. Ultraäänen nopeus riippuu niin väliaineesta kuin lämpötilastakin. Äänennopeudesta lisää seuraavassa kappaleessa.

Ultraääntä käytetään yllättävän monissa sovelluksissa. Ehkä yleisin sovelluskohde on sairaalassa. Ultraäänellä tutkitaan niin sikiötä kuin muitakin nestettä sisältäviä kudoksia. Vastaavaa laitetta käytetään myös veneissä ja laivoissa kaikuluotaukseen. Teollisuudessa ultraääntä käytetään laaduntarkkailussa. Siinä metallilevyyn johdetaan ultraääntä ja sitten vastaanoton erilaiset kaiut kertovat mahdollisista hiusmurtumista. Yksi käyttökohteita ultraäänelle on hitsaaminen. Tässä menetelmässä hitsattavat liitospinnat kuumenevat nopeasti ja sulavat yhteen. Tätä menetelmää voidaan käyttää metallin, keramiikan ja metallien hitsaamiseen. Myös voidaan hitsata eri metalleja keskenään. Teollisuudessa käytetään ultraääntä kaikenlaisiin etäisyyden mittauksiin lyhyillä matkoilla. Ultraääntä käytetään myös pesemiseen, jolloin ultraääni saa aikaan värähtelyä pestävässä kohteessa ja siten irrottaa lian. Luonnossa ultraääntä käyttävät lepakat suunnistamiseen. /4/



Kuva 1 Erilaisia ultraäänilähtettä

Kuvassa näkyy erityyppisiä ultraäänivastaanottimia ja lähtettä. Nämä ultraäänianturit perustuvat piezoelektriseen elementtiin. Kolme vasemmanpuoleista mallia ei voi käyttää lähetin-vastaanottimena hyvin tuloksin. Lähetintä ja vastaanotinta ei erota ulkokuoresta toisistaan. Oikeanpuoleinen, tässä työssä käytetty, anturi on niin lähetin kuin vastaanotinkin, ja se on suojattu lialta.

3.3 Äänennopeus

Äänennopeus ideaalikaasulle lasketaan alla olevalla kaavalla./5/

$$c = \sqrt{\kappa \cdot R \cdot T}$$

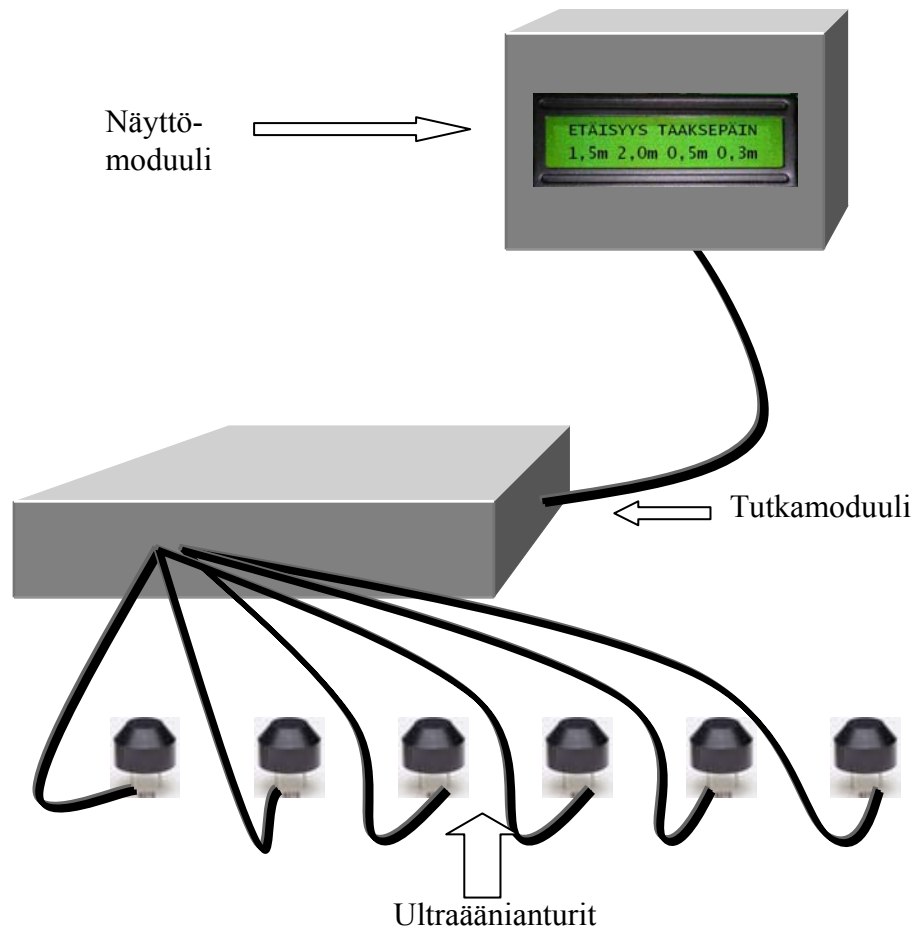
k = (kappa) adiabaattinen indeksi (1.402 ilmalle)
R = 287,05 J/(kgK) ilmalle, jonka moolimassa on noin 16
T = Lämpötila Kelvineinä
c = äänennopeus m/s

Äänennopeuksia eri väliaineissa:

- Ilmassa 343 m/s (20°C)
- Vedessä 1492 m/s (20°C)
- Kiinteissä aineissa 70-6000 m/s, pehmeässä kumissa 70 m/s ja lasissa 6000 m/s. Teräksessä 5100 m/s

4 Peruutustutka

Tässä kappaleessa kerrotaan peruutustutkan kytkennästä ja sen toiminnasta. Alla on piirretty kuva peruutustutkalaitteistosta.



Kuva 2 Peruutustutka

4.1 Ominaisuudet

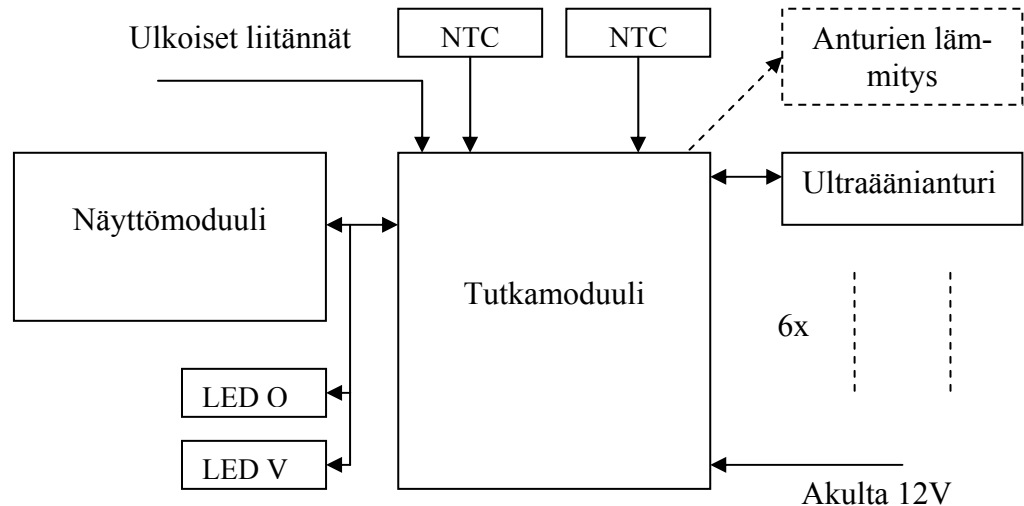
Tutkalaitteiston päätehtävä on mitata matkaa auton peruuttaessa. Tämä tehdään ulträäniantureilla, jotka pystyvät mittaamaan matkaa muutamasta sentistä noin kahdeksan metriin.

Kaistanvaihtotutka on lähes samalla tekniikalla toteutettu lisäsovelluskohde. Kaistanvaihtotutka mittaa sitä, onko toista autoa vieressä, ja vilkuttaa LED:iä tai hälyttää auton läheisyyden mukaan.

Lämpötilanmittaus tehdään lähinnä mahdollista anturien lämmitystä ja äänennopeuden lämpötilakorjausta varten. Lämpötila tieto tuo myös lisäarvoa näyttöön, joka pääosan ajasta olisi täysin tyhjä. Näytöllä näytetään lämpötilojen lisäksi myös auton akun jännite haluttaessa.

Anturien lämmitys on lisäominaisuus, jonka tarpeellisuuden näkee vasta peruutustutkan ollessa asennettu ja käytössä. Tähän ominaisuuteen on kytkennässä valmius, mutta itse lämmityselementtejä ei heti aluksi prototyyppiin rakenneta.

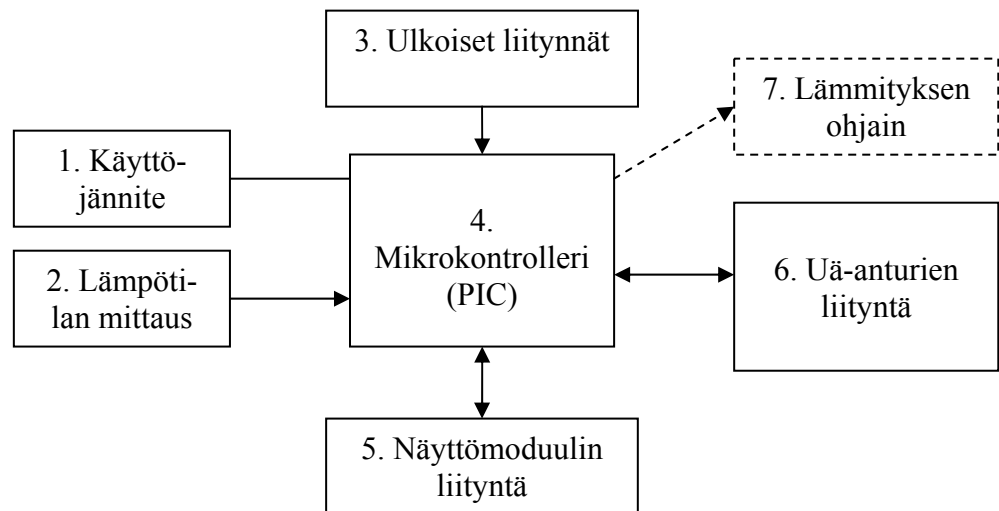
4.2 Peruutustutkan lohkokaavio



Kuva 3 Peruutustutkan lohkokaavio

Yllä on lohkokaavio peruutustutkan päälohkoista. Jokainen lohko on fyysisesti oma moduuli tai komponentti. Käyttöjännite tulee tutkamoduulille, jossa se sitten laskeetaan 5 volttiin, jota sitten käytetään mikropiireissä tutkamoduulissa ja näyttömoduulissa. Ulkoiset liitännät sisältävät peruutusvalon ja vilkkujen ohjaussignaalit. NTC-vastukset mittaavat lämpötilaa auton sisällä ja ulkona. Lämpötilatietoa käytetään lähinnä anturien lämmityksen ohjaukseen. Anturien lämmitys on lisäominaisuus, koska sen tarpeellisuudesta ei ole tietoa. Oikeata ja vasenta LED:iä käytetään kaistanvaihtotutkan varoituksen merkkivalona, ja ne sijoitetaan auton niin sanottuun a-pilariin. LED:it on kytketty samaan johtoon (liittimeen) kuin näyttömoduuli tutkamoduulin päästä. Kytkennässä on paikka kuudelle ultraäänianturille 4 peruutustutkaa ja 2 kaistanvaihtotutkaa varten. Tutkassa on valmius vielä 2:n anturin lisäämiseksi esimerkiksi auton etukulmiin.

4.3 Tutkamoduuli



Kuva 4 Tutkamoduulin lohkokaavio

1. KÄYTTÖJÄNNITE

Auton sähköjärjestelmästä saatava n.14V muunnetaan 10 ja 5 voltiksi yksinkertaisella regulaattorikytkennällä. 5V:n jännitettä käytetään mikropiirien ja näytön ja 10V:n jännitettä ultraäänilähettimien käytösähköinä.

2. LÄMPÖTILAN MITTAUS

Lämpötila mitataan NTC-vastuksella, ja NTC kytketään perinteiseen jännitteenjakokytkentään. Tällä tavoin ei saada erityisen hyvää mittaustulosta pienestä jännitteen vaihteluvälin vuoksi. Lämpötilan mittauksesta lisää kappaleessa 4.5 Lämpötilan mittaus

3. ULKOISET LIITYNNÄT

Ulkoisiin liitäntöihin kytketään ohjausjännite suoraan peruutusvalolta ja vilkuilta. Signaalit otetaan suoraan auton sähköjärjestelmästä. Sen takia signaaleille tehdään galvaaninen erotus optoeristimillä. Niin menetellään siksi, että mahdollinen autonsähköjärjestelmän vika ei rikkoisi peruutustutkaa. Liitäntöjen tuloimpedanssit ovat suuret, niin että laite ei kuormita vilkkujen relettä ja vaikuta niiden toimintaan.

4. MIKROKONTROLLERI

Mikrokontrolleri on kytkennän ydin, johon kaikki muut lohkot on kytketty. PIC:eissä on paljon toimintoja, joten oheiskomponentteja ei juurikaan tarvita. PIC:stä kerrotaan lisää **kappaleessa 4.2.2.**

5. NÄYTTÖMODUULIN LIITYNTÄ

Tämä lohko ei sisällä minkäänlaista kytkentää, koska näytönohjaus tulee suoraan PIC:ltä. Näyttömoduulin kaapeli liitetään tutkamoduuliin 8-napaisella RJ-45-liitimellä, samaa liitintä käytetään tietokoneiden ethernet-verkoissa (cat5).

6. UA-ANTURIEN LIITYNTÄ

Tämä lohko on koko kytkennän monimutkaisin ja siten suurin lohko, jos laskee komponenttimäärää. Tämä lohko on kuvattu komponenttitasolla **kappaleessa 4.2.1**.

7. LÄMMITYKSEN OHJAUS

Lämmitykselle on suunniteltu kytkentä piirilevyllä, vaikka sitä ei välttämättä koskaan oteta käyttöön, mutta lämmityselementit saa sitten helposti kytkettyä jos niitä tarvittaisiin. Lämmitys toimii yksinkertaisella ”alaspäin”-kytkennällä, joka on toteutettu N-MOSFET:illä, eli kytkentä kytkee lämmityksen pois päältä tai päälle, välimuotoa ei ole.

4.3.1 UÄ-anturien liityntä

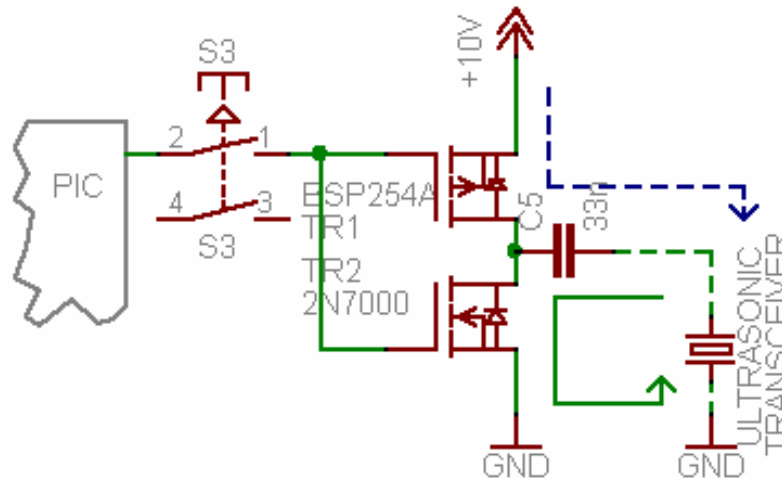
Ultraäänianturien liitintä voidaan jakaa kolmeen lohkoon, lohkot ovat: anturin valinta, lähetyksen pääteaste ja kaiun vastaanotto. Nämä lohkot on seuraavassa jaettu omiin alaotsikkoihin.

4.3.1.1 Anturin valinta

Anturin valinnan ajatuksena on se, että kahdella PIC:n lähtöpinnillä saadaan ohjattua 40 kHz:n taajuus kaikille antureille. Vastaavalla valinnalla johdetaan vastaanotettu kaiku vain yhteen vahvistinkytkentään.

4.3.1.2 Lähetyksen pääteaste

Alla olevassa kuvassa on ultraäänilähdön pääteaste. Kytkennän tarkoitus on muuttaa 10 voltin jännite kaksipuoliseksi jännitteeksi. Näin ollen ultraäänilähtettiin vaikuttaa +-10 voltia. S3-kytkin kuvaa analogiakytinmikropiiriä 4051 (datakirjan sivu liitteenä, Liite 1) eli anturinvalintalohkoa. Kytkin avaa yhteyden PIC:iltä MOSFET:ien hilalle. Kytkimen ollessa auki ”suuri-impedanssissa tilassa” kumpikin MOSFET pysyvät suljetussa tilassa, tämä sen takia että kondensaattori ei haittaisi vastaanottoa. Tällaisessa kytkennässä tulee huomioida käytettyjen MOSFET:ien sisäinen vastus. Mikäli sisäinen vastus olisi hyvin pieni, tällaista kytkentää ei voisi käyttää, vaan kumpaakin FET:iä pitäisi ohjata erikseen. Tämä johtuu siitä, että FET:ien sisäinen kondensaattori pitäisi kumpaakin FET:ejä hetken auki, tällöin kytkentä olisi oikosulussa. Tässä kytkennässä käytetyissä FET:eissä sisäinen vastus on 5 ja 10 ohmia.



Kuva 5 Ultraäänivahvistimen päätteaste

Kytkenän kondensaattori (kuvassa C5) lasketaan ylipäästöjakosuotimen kaavalla. Kaavan 120Ω , on mitattu syöttämällä 40 kHz :n taajuutta etuvastuksen läpi ultraäänianturiin, tästä on mitattu ja laskettu jännitehäviöt, mistä saa vastuksen.

$$C = \frac{1}{2\pi rf} \Rightarrow \frac{1}{2\pi * 120\ \Omega * 40\ \text{kHz}} = 33\ \text{nF}$$

Kaava 2 Ylipäästösuodin

KYTKENNÄN TOIMINTA:

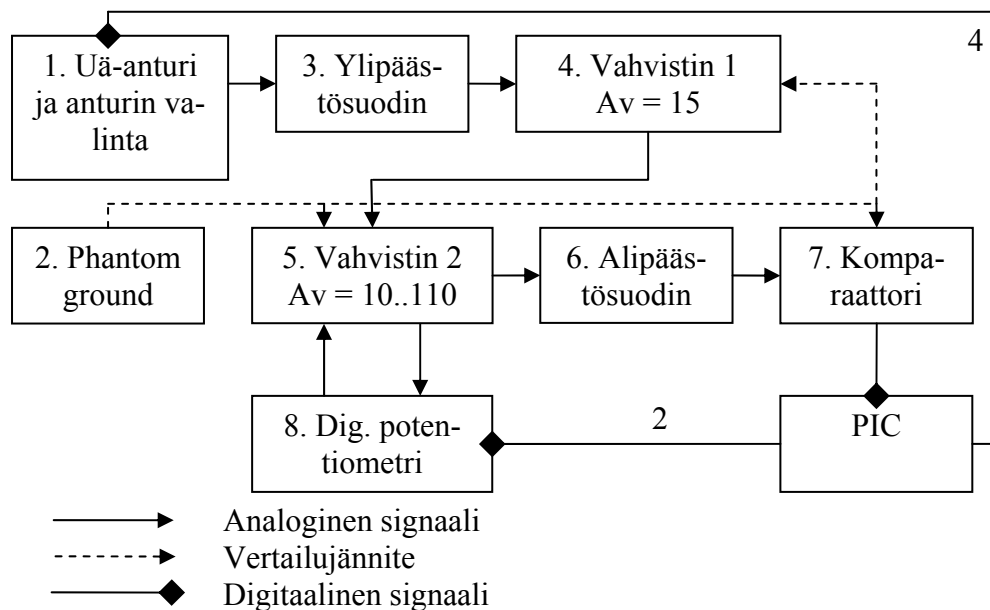
1. Ylempi p-kanavainen FET avataan nollaamalla hilajännite. Tällöin jännite FET:ien välissä nousee 10 V :iin, tällä hetkellä uä-lähttimen navoissa on sama 10 V :n jännite. Näiden läpi kulkevaa virtaa kuvataan katkoviiva nuolella. Tämä virta alkaa ladata kondensaattoria, mikä sitten laskee jännitettä uä-lähttimen navoissa. Edellisellä kaavalla laskettu kondensaattori ehtii melkein täyttyä kun FET:ien tila muuttuu.
2. P-kanavainen FET suljetaan 10 V :n hila jännitteellä. Tästä muutamien mikrosekunnin päästä avataan alempi N-MOSFET 10 V :n hila jännitteellä. Tässä tilanteessa kondensaattorin +-napa on maissa, ja tällöin virta lähtee kiertämään alemman nuolen suuntaan. Tämä tarkoittaa uä-anturin kannalta negatiivista jännitettä.

4.3.1.3 Kaiun vastaanotto

Ultraäänikaiku tekee muutaman millivoltin tasoisen jännitteen ultraäänianturin nas-toihin. Jännite syntyy siitä, että ultraäänianturin kalvo on piezosähköinen elementti. Tämä vain muutaman millivoltin jännite vahvistetaan noin 5 volttiin, jotta mikrokontrolleri voi tunnistaa kaiun tulleen. Ultraäänianturin taajuuskaista on kapea, tässä mallissa $1,5\text{ kHz}$. Sen vuoksi kytkentä ei tarvitse suoranaista kaistanpäästösuodinta, koska esimerkiksi ihmisen kuulemat äänet eivät juuri tee minkäänlaista jännitettä anturiin. Kuitenkin ilmassa on aina myös 40 kHz :kin ääntä. Tämä ongelma on kytkennässä ratkaistu automaattisesti säädettävällä vahvistuksella.

Kytkenän tärkein komponentti on operaatiovahvistin, niitä kytkennässä on 3. Operaatiovahvistimeksi valittiin TS 925. Alla on listattu kyseisen piirin tärkeimmät valintaan johtaneet ominaisuudet. Lisää ominaisuuksia löytyy liitteestä 2.

- Single supply: Toimii yksipuolisella käyttöjännitteellä.
- Low input offset voltage: 900uV, sisääntulojen pienin huomattava erojännite.
- Phantom ground: TS925:stä saa suoraan vertailujännitteen, joka on puolet käyttöjännitteestä.
- Quad channel: TS925:ssä on neljä operaatiovahvistinta yhdessä mikropiirissä. Yksi näistä on käyttämättä.



Kuva 6 Vastaanotin kytkennän lohkokaavio

Kytkenän komponenttien arvot on valittu laskemalla ja käytännössä kokeilemalla. Kokeilussa lähetettiin 40 kHz:in taajuutta funktiogeneraattorilla ja signaali kytkettiin ultraäänilähettimeen, kaiku otettiin vastaan katon kautta (etäisyys n 2m). Seuraavassa ovat vastaanotinkytkennän lohkojen toiminnan selvitykset.

1. Laitteessa on 6 uä-anturia, joista 4051-piirillä kytketään haluttu anturi vahvistimeen, Liite 1. 4051-piiriä ohjataan neljällä signaalilla, joista kolmea käytetään valitsemiseen, yksi on piirin valintaan.
2. Phantom ground on TS925:n sisäinen ominaisuus, jossa sisäisellä jännitteen jakokytkennällä käyttöjännite puolitetään ja puskuroidaan sisäisellä op-vahvistimella. Näin ollen piiristä saadaan 2,5 V:n referenssijännite, jota käytetään vahvistimissa ja komparaattorissa.
3. Ylipäästösuodin tekee karkean suodatuksen. Tarkoituksena on poistaa voimakkaat pienet taajuudet, joita tulee tuulesta ja alemptaajuisista äänistä.

4. Ensimmäinen vahvistusaste, jossa tuleva noin millivoltin vaihtojännite vahvistetaan n.15-kertaiseksi ja samalla jännitteen nollassa nostetaan 2,5 volttiin. Ulostulossa on siis 2,5 V:n jännite, jossa on verhoikäyränä 15 kertaa suurempi jännite kuin se, joka anturilta saadaan. Jos ensimmäisen vahvistinasteen vahvistus olisi suurempi, kytkennästä tulisi resonaattori, joka alkaisi itsekseen värähdellä.
5. Ensimmäisen ja toisen vahvistinasteen välissä on kondensaattori, joka poistaa tasasähkökomponentin. Toisen asteen vahvistus on 10 - 110, sitä muutetaan digitaalisella potentiometrillä. Vahvistimien yhteisvahvistus on 150 – 1650 eli 1mV. Se vahvistetaan noin 1 volttiin.
6. Alipäästösuodatuksella n. 40 kHz:n taajuudella sykkivästä 2,5 V:n tasa-jännitteestä suodatetaan näkyviin verhoikäyrä, joka on jännitteeltään yli 2,5 volttia, jos ultraääntä on tullut.
7. Komparaattori eli vertailija toimii siten, että jos op-vahvistimen plus-napaan tulee suurempi jännite kuin miinus-napaan, ulostulojännite nousee käyttöjännitteeseen eli 5 volttiin. Op-vahvistimen plus-napaan on lisätty myötäsuumainen diodi. Sen kynnyksijännite (0,6V) tekee kynnyksen, joka vahvistetun signaalin tulee ylittää, että se tulkittaisiin kaiuksi. Tästä voidaan laskea pienin ultraäänianturin jännite, joka tulkitaan kaiuksi.

$$\hat{U}_{anturi}_{\min} = U_{d_{kynnys}} / A_{v_{\max}} \Rightarrow 0,6V / 1650 = 0,3mV$$

8. 100kΩ:lla ja 128-portaisella digitaalisella potentiometrillä säädetään haluttu vahvistus toiseen vahvistimeen. Digitaaliseen potentiometriin säädetään haluttu arvo kahdella signaalilla up/down ja clk. Up/down-signaalilla valitaan muutoksen suunta ja clk:lla hyväksytään. Tarkempia tietoja ad5220-komponentista on liitteessä, Liite 3.

4.3.2 Mikrokontrolleri

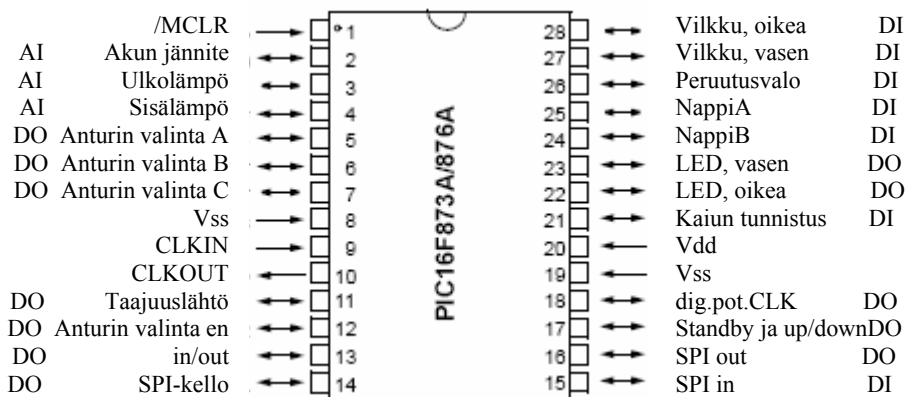
Peruutustutkan kytkennän sydämenä toimii Microchipin valmistama PIC-piiri (Programmable Integrate Circuit). Alla on taulukko PIC16f87XA-sarjan mikrokontrollereista ja niiden tärkeimmistä ominaisuuksista

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

Taulukko 1 PIC 16F87XA-mikrokontrollerit

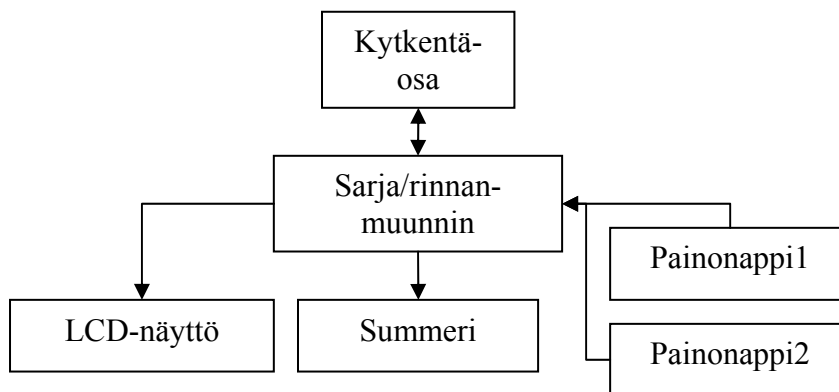
Tähän sovellukseen riittää ominaisuuksiltaan malli 873. Lähinnä valinnan määrää muistin määrä, I/O:n ja A/D-kanavien tarve. Koteloinniltaan riittää 28-pinninen dil-kotelo. Seuraavassa kuvassa on PIC18f873A:n pinnijärjestys ja se mihin signaaleihin se on kytketty.

MIKROKONTROLLERIN KYTKENTÄ:



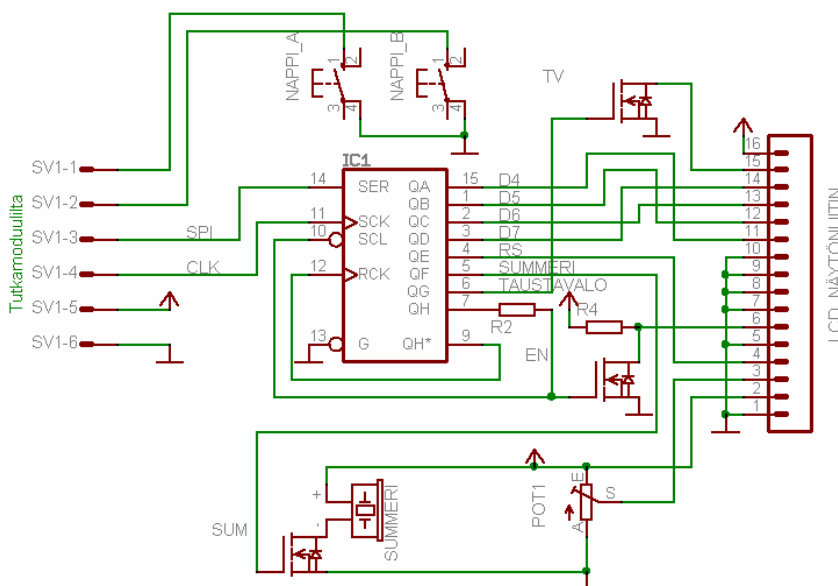
4.4 Näyttömoduuli

Näyttömoduulin lohkokaavio:



Kuva 7 Näyttömoduulin lohkokaavio

Yllä on jaettu päalohkoihin sinänsä yksinkertainen näyttömoduulin kytkentä. Oleellisin osa kytkentää on sarja-rinnanmuunnin. Muunnos on toteutettu 74HC595-piirillä. Painonappien tiedot lähetetään tutkamoduulille muista erillisinä johtoina. Seuraavassa on vielä tarkempi kuvaus siitä, kuinka sarja-rinnanmuunnos on toteutettu.



Kuva 8 Näyttömoduulin kytkentä

74HC595-piirin pinnien selvitys (Sisäinen kytkentä, Liite 4):

SER = SPI-datalinja, tämän pinnin arvo ladataan siirtorekisteriin

SCK = SPI-kello, tämän nousevalla reunalla siirtorekisteriin ladataan SER-pinnin arvo

SCL = ”Nollaa” siirtorekisterin, mutta ei ulostuloja

RCK = Lataa siirtorekisterin arvon ulostuloihin

G = Ulostulojen sallinta. Ulostulot pidetään aina päällä

QA,QH = Ulostulot LSB, MSB

QH' = Siirtorekisterin ylin bitti.

Sitten, kun 8-bittinen siirtokehys on siirtorekisteriin ladattu käyttäen SPI-väylää, QH' on silloin ”1” kehyksen MSB-bitin vakio-”1”-tilan vuoksi. Tämä tekee liipaisun pinniin RCK, joka lataa siirtorekisterin arvon ulostuloon. Nyt QH-pinni on ”1”. Tämä nollaa siirtorekisterin (pinni SCL) ja panee FET:n johtavaan tilaan, joka sitten liipaisee LCD-näyttöä ja lataa siirtorekisterin ulostulojen arvon LCD-näyttöön.

Muut näyttömoduulikytkennän toiminnot ovat hyvin suoraviivaisia. Yksi vetää summerin FET:n auki, joka sitten käynnistää summerin. Samanlainen on taustavalon käynnistys kytkentä. POT1:llä säädetään LCD-näytön kontrasti. Napit A ja B kytkyvät signaalit maahan, ylösvetovastus sijaitsee tutkamoduulissa.

4.4.1 LCD-näyttö

Peruutustutkassa on 16x2-LCD-näyttö, 16 tarkoittaa merkkejä rivillä ja 2 rivimäärää. Yksi merkkiruutu koostuu 8x5 pisteestä, merkit ovat pääasiassa 7x5 pisteen kokoisia. Näyttö on Hitachi 44720-yhteensopiva, niin kuin lähes kaikki muutkin samankaltaiset näytöt. Hitachi 44720-yhteensopivissa näytöissä on sisäinen muisti ja näytönohjauspiiri. Tämä helpottaa näytön ulkoista ohjaamista, näytölle siis lähetetään vain tieto siitä, mihin halutaan kirjoittaa ja mitä. /6/



Kuva 9 FM162 LCD-näyttö

4.4.1.1 Toiminta

Näytön toimintaa kuvataan tarkemmin kappaleessa 5.4.1 Näytönohjauksen toimintakaavio ja kappaleessa 5.4.2 Näytön alustus.

4.4.2 Sarjaliikenneprotokolla

Näyttö- ja tutkamoduulin välinen johdinmäärä on pyritty pitämään mahdollisimman pienenä. Tähän on syynä se, että itse johto olisi mahdollisimman ohut ja siten helposti taipuva. Taipuvuus on tärkeä, kun johto asennetaan autoon ja johto kiertelee piilossa auton kynnyskotelossa tai vastaavassa paikassa. Sarjaliikenteellä on saatu tarvittava tiedonsiirto kahteen johtimeen, kun siihen ilman sarjamuunnosta tarvittaisiin kahdeksan johdinta. Sarjaliikenteellä ohjataan kaikki näyttömoduuliin tulevat (input) signaalit. Lähtevät (output) signaalit käsitellään rinnakkaismuotoisina niiden vähyyden takia. Sarjaliikenne tarvitsee tiedonsiirtoon protokollan. Protokolla tarvitaan vastaanotossa tiedon purkua varten. Tällaisessa tapauksessa, jossa on yksi lähettäjä ja vastaanottaja, protokolla on yksinkertainen, vain pelkkä datakehys. Virheenkorjausbittejä ei ole, koska mahdollinen virhe ei ole erityisen merkittävä, kytkentä ja kehys on pyritty pitämään mahdollisimman yksinkertaisena. Alla datakehys, lähetys tehdään MSB edellä.

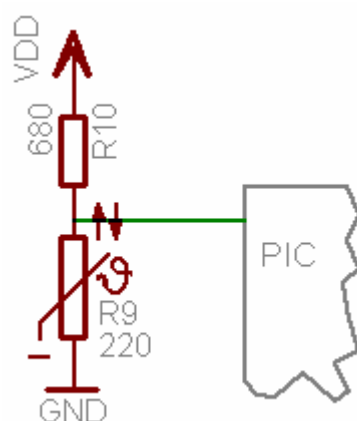
Data[3..0] (4bit)	RS (1bit)	Summeri (1bit)	Taustavalo (1bit)	Enable (1bit)
LSB				MSB

Kehyksen Enable-bitti on käytännössä aina aktiivinen eli ”yksi”, eli toisin sanoen jokaisella kehyksellä on vaikutusta näyttöön. Sarjaliikenteellä lähinnä ohjataan näyttöä, mutta yksi bitti on varattu hälytyssummerille.

Itse datan siirto on toteutettu SPI-protokollan mukaisesti eli aluksi asetetaan data-bitti haluttuun tilaan ja sitten lähetetään kellopulssin nouseva reuna. Vastaanottaja lukee sitten databitin kellon nousevalla reunalla. Tätä siirtoprotokollaa käytetään usein AD/DA-muuntimissa.

4.5 Lämpötilan mittaus

Lämpötilan mittaus on lisätoiminto, joka tuo lisäarvoa peruutustutkalaitteistoon. Lämpötilaa käytetään peruutustutkan tarvitseman äänennopeuden laskennassa, koska äänennopeus on riippuvainen vallitsevasta lämpötilasta. Tosin lämpötilan muutos ei vaikuta merkittävästi tuloksiin lyhyillä matkoilla.



Kuva 10 Lämpötilan mittaus

Lämpötilan mittaus kytkentä on yksinkertainen jännitejako kytkentä. Vastusten arvojen valinnassa tulee ottaa huomioon se että kokonaisvastus ei saa ylittää 2,5k Ω . Tämä raja on PIC:n AD-muunnoksen suurin sallittu tuloimpedanssi. Näillä valituilla arvoilla impedanssi ylittyy vasta -25 °C:ssa. Jännitteen jaon toinen vastus on valittu siten, että vastukset ovat samanarvoiset mitta-alueen puolivälissä eli noin 0 °C. NTC-vastuksen arvo 220 Ω on arvo 25 °C:n lämpötilassa. Tällä valinnalla saadaan kaikista suurin arvojen vaihteluväli. Jännite vaihtelee -30..+40 °C:n lämpötilassa 4:sta 0,7V:iin. Tämä tekee AD-muunnos arvoina 150 – 800, kun käytettävä alue 10-bittisenä muunnoksena on 0-1023, eli muunnos käyttää vain noin 65 % käytet-

tävissä olevasta muunnosalueesta. Kuitenkin näinkin tehtynä lämpötilan mittauksen tarkkuus on noin 0,1 °C, tosin se vaihtelee vähän eri lämpötilassa NTC-vastuksen epälineaarisuuden vuoksi.

Tämä NTC-vastuksen epälineaarisuus tekee lämpötilan laskuun vaikeuksia. NTC-vastuksen arvosta, lämpötilaksi laskemiseen on olemassa kaava, mutta siinä on potenssilaskuja ja logaritmeja, joten laskeminen PIC:llä olisi hidasta (kaava alla Kaava 3). Tämän vuoksi käytetään taulukkoa johon on laskettu 32 AD-muunnosarvoa 2 °C:n välein. Tämä 2 °C väli lasketaan, siten että väli oletetaan lineaarisiksi. Arvot on laskettu alla olevalla kaavalla. Arvot on taulukoituna liitteesä, Liite 5.

$$R(T) = R_{ref} \times e^{A + B/T + C/T^2 + D/T^3}$$

$$T(R) = \left(A_1 + B_1 \ln \frac{R}{R_{ref}} + C_1 \ln^2 \frac{R}{R_{ref}} + D_1 \ln^3 \frac{R}{R_{ref}} \right)^{-1}$$

Kaava 3 NTC-vastuksen laskenta

R(T) = Vastuksen arvo, tunnetussa lämpötilassa

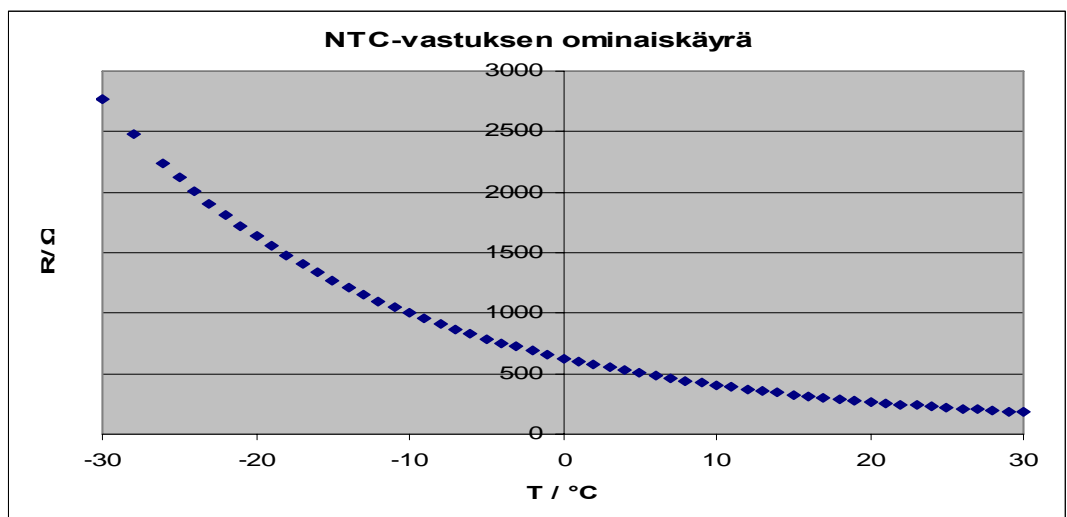
T(R) = Lämpötila, tunnetulla vastuksen arvolla

R_{ref} = NTC-vastuksen nimellisarvo eli tässä tapauksessa 220Ω (25 °C)

T = Lämpötila Kelvineinä

A,B,C,D,A₁,B₁,C₁,D₁ = NTC-vastuksen vakioita, Liite 6

Seuraavassa kaavio NTC-vastuksen ominaiskäyrästä, mikä on laskettu edellisellä kaavalla.



Taulukko 2 NTC-vastuksen ominaiskäyrä

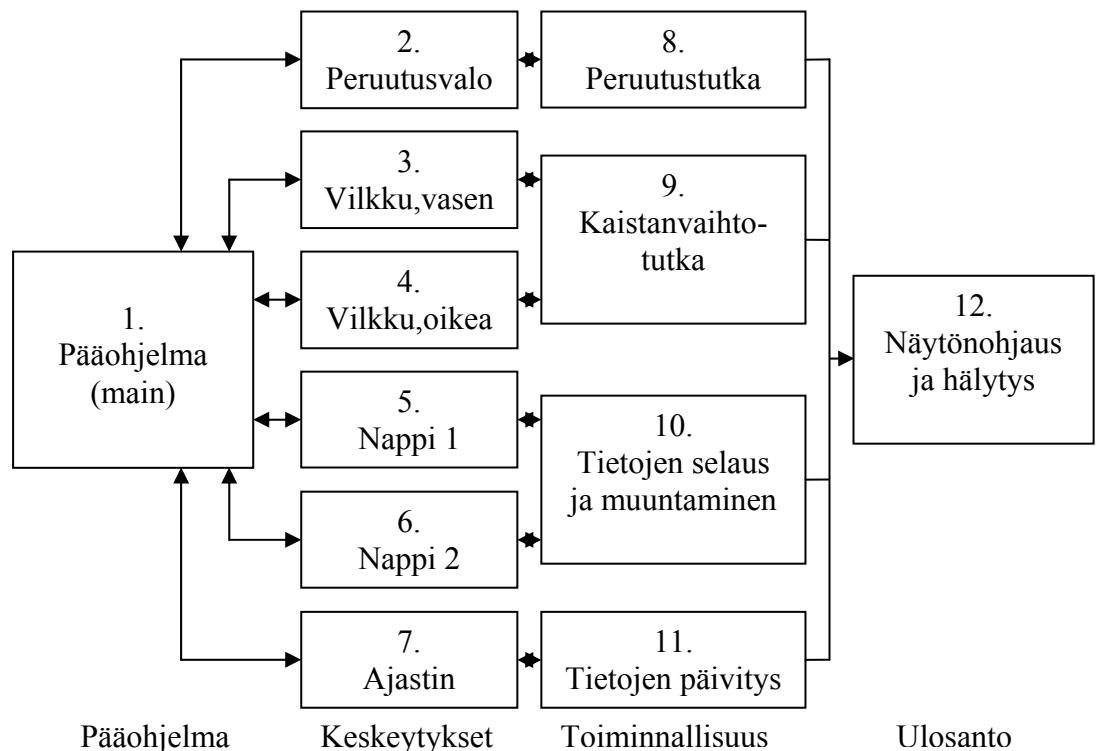
5 Ohjelma

Tällaisissa sulautetuissa järjestelmissä, mikrokontrollerin ohjelma on suuri osa järjestelmää. Ohjelmaa muuttamalla pystytään tekemään lisää ja parantelemaan vanhoja ominaisuuksia koskematta itse kytkentään.

5.2 Ohjelman rakenne

Sulautetuissa järjestelmissä ohjelman rakenne on usein sellainen, että toiminnot ovat pääohjelman ympärillä. Pääohjelmasta siirrytään keskeytyksen tullessa keskeytyksen käsittelyyn. Keskeytyksen käsittelyn jälkeen palataan takaisin pääohjelmaan odottamaan seuraavaa keskeytystä. Keskeytyksen käsittelijä on normaali aliohjelma, joka sitten kutsuu muita aliohjelmiä. Ohjelman rakennetta on helpointa kuvata lohkokaaaviolla.

LOHKOKAAVIO:



Kuva 11 Ohjelman lohkokaavio

LOHKOJEN SELITYS:

1. Pääohjelma

Pääohjelma on se, johon mennään kun laitteeseen kytketään virta. Pääohjelma nimetään yleensä nimellä main(). Pääohjelma ei juuri sisällä muuta kuin muuttujien alustuksia ja ns. ikuisen silmukan, jossa ohjelma pysyy odottaessaan keskeytyksiä.

2. Peruutusvalo (ulkoinen keskeytys)

Peruutusvalo tekee PIC:lle ulkoisen keskeytyksen. Tämä käynnistää peruutustutkan. Peruutustutka aliohjelma palaa jokaisen lähetyskerran jäl-

keen katsomaan, onko peruutusvalo vielä päällä. Jos peruutusvalo ei ole enää päällä, palataan takaisin pääohjelmaan.

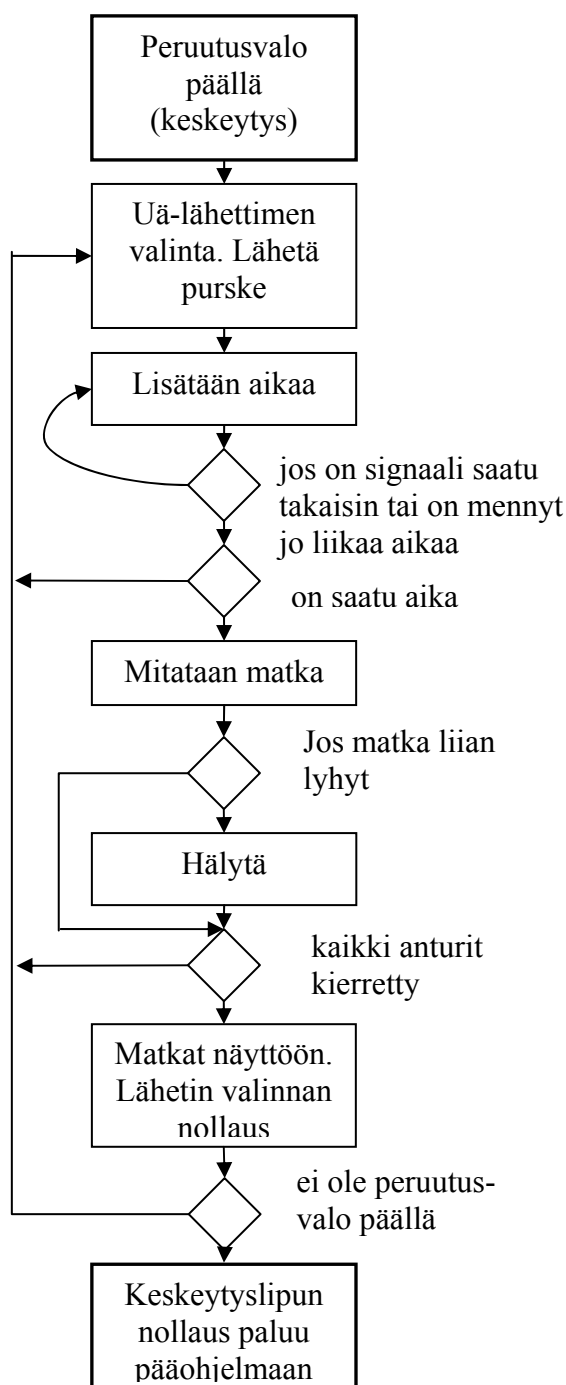
3. *Vilkut* (ulkoinen keskeytys)
4. Kumpikin vilkku tekee PIC:lle oman ulkoisenkeskeytyksen. Kumpikin vilkku käynnistää kaistanvaihtotutkan ja samalla tavoin kuin peruutusvallossakin palataan katsomaan onko vilkku vielä päällä, jos vilkku ei ole enää päällä niin palataan pääohjelmaan.
5. *Nappi 1* (ulkoinen keskeytys) *Nappi2* (pollaava)
6. Käyttäjän painama nappi 1 tekee PIC:lle keskeytyksen näyttömoduulista. Napeilla voidaan PIC:n ohjelman käyttäjälle näkyviä asetuksia katsella ja muuttaa. nappikeskeytys toimii samalla tavoin kuin vilkku keskeytyskin
7. *Ajastin* (sisäinen keskeytys)
Ajastin tekee sisäisen keskeytyksen PIC:lle noin yhden sekunnin välein. Tätä käytetään siihen että päivitetään tiettyjä tietoja kuten lämpötilat. Tietojen päivityksen jälkeen palataan takaisin pää ohjelmaan
8. *Peruutustutka*
Peruutustutka ei ole yksi aliohjelma vaan moduuli eli joukko aliohjelmaa. Peruutustutka moduuli kirjoittaa tietonsa näyttöohjaus/hälytys-moduuliin. Tämän jälkeen palaa takaisin keskeytys aliohjelmaan katsomaan onko keskeytys vielä päällä. Peruutustutkasta tarkemmin kappaleessa **5.3.1**
9. *Kaistanvaihtotutka*
Kaistanvaihtotutka on hyvin samanlainen kuin peruutustutka ja käyttävät pääasiassa samoja aliohjelmaa. Kaistanvaihtotutkasta tarkemmin kappaleessa **5.4.1**
10. *Tietojen selaus / muuntaminen*
Nappi A:sta voi selata eri näyttötiloja ja Nappi B:stä valita toimintoja / arvoja. Kummankin napin painallus johtaa tarpeeseen päivittää näyttö. Yhden päivittämisen jälkeen palataan keskeytys aliohjelmaan katsomaan onko nappi vielä päällä. Tätä tarvitaan joidenkin asetusten muuttamisessa, joissa nappia pidetään pitkään pohjassa.
11. *Tietojen päivittäminen*
Päivitetään tiedot, jotka näkyvät näytöllä aina kun tutkat tai muu toiminto ei ole päällä. Tällaisia ovat akun jännite ja lämpötilat. Eli siis tehdään AD-muunnokset, ja kirjoitetaan uudet tiedot näyttöön. Tämän jälkeen palataan ajastimen keskeytys aliohjelmaan nollaamaan ajastin.
12. *Näytönohjaus / hälytys*
Tätä moduulia käyttävät kaikki toiminnalliset moduulit. Tähän kuuluu lähinnä näytönohjauksen sarjaliikenteen datakehityksen luominen. Sama kehys ohjaa myös hälytintä ja näytöntaustavaloa. Kun näyttöön on kirjoitettu, palataan takaisin toiminnalliseen moduuliin. Tästä moduulista lisää kappaleessa **5.4.1**.

5.3 Tutka

Tässä peruutustutkassa on kaksi erilaista tutkatoimintoa peruutus- ja kaistanvaihtotutka, jotka ovat fyysisesti aivan samanlaisia. Tutkatoimintojen ero tehdään ohjelmassa ja ultraäänianturien erilaisena asetteluna.

5.3.1 Peruutustutka

Peruutustutkan perusidea on siis kertoa auton ajajalle onko auton takana estettä ja jos on niin kuinka kaukana se on. Esteen läheisyys voidaan kertoa eritavoin lukemana näytössä, valoina tai äänimerkkinä. Tärkein näistä on äänimerkki, joka löytyy jokaisesta peruutustutkasta. Yleensä äänentaajuus kertoo matkan esteeseen siten, että aluksi ääni tulee harvana ”piippauksena”. Esteen lähestyessä ”piippaus” tiihentyy kunnes se on jatkuvaa.



PERUUTUSTUTKAN toimintakaavio:

Peruutusvalo tekee keskeytyksen mikrokontrollerille.

40kHz taajuus ultraäänilähettimelle. Lähetetään kymmenen jaksoa kesto noin. 25uS.

Lisätään aikaan

Luetaan portista onko vastaanotettu ja katsotaan onko aikaa mennyt jo niin kauan, että signaali on hukunut.

onko saatu laskettava arvo

Mitataan matka, ohjelma laskee suoraan senttimetreiksi

Jos matka on vaarallisen lyhyt tai lyhenee nopeasti

Hälytetään niin kauan kun matka on alle 1,2m. Neliportaisesti

Kaikki neljä anturia kierretty.

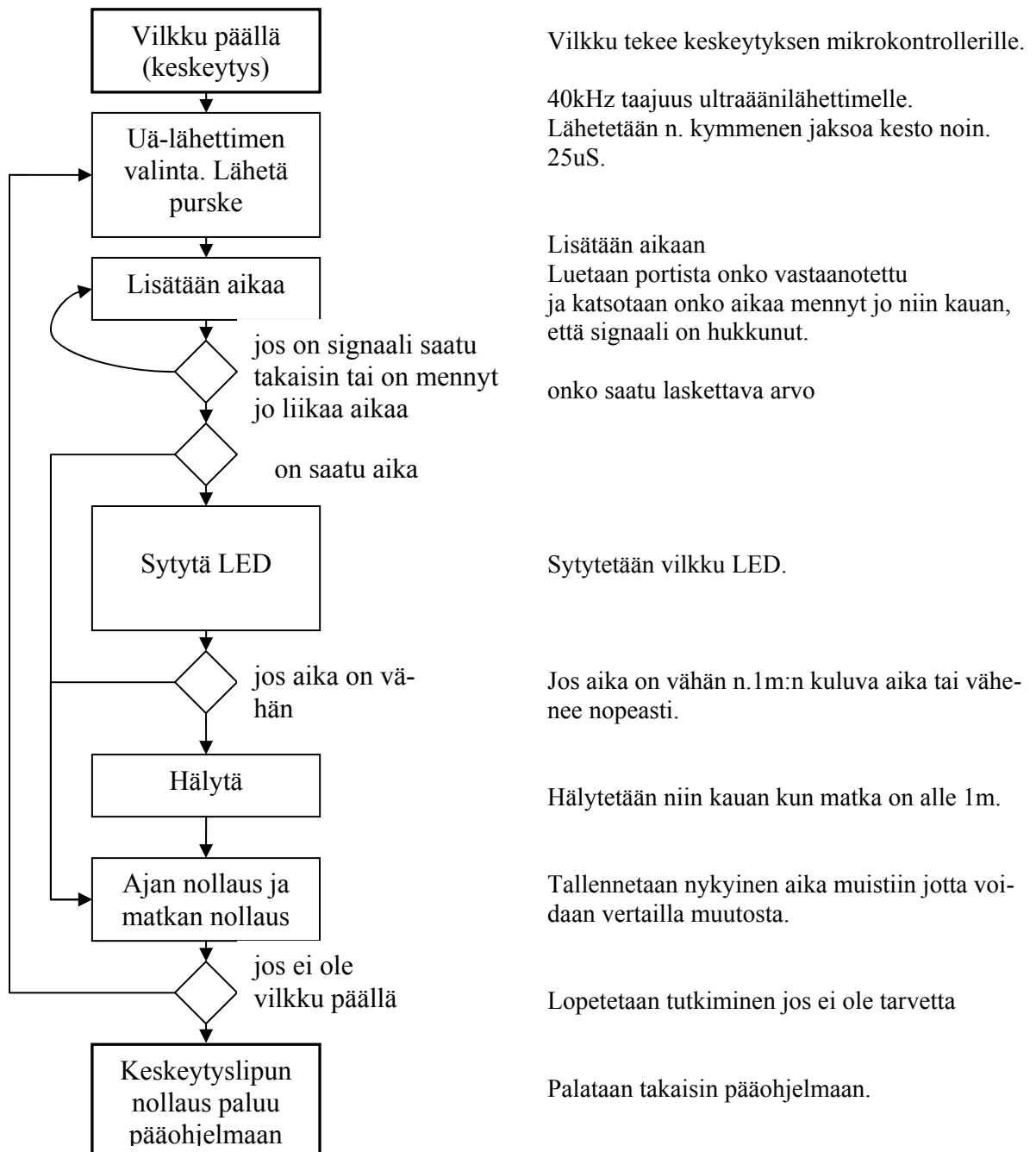
Tulostetaan matkat näyttöön.

Palataan takaisin pääohjelmaan.

5.3.2. Kaistanvaihtotutka

Kaistanvaihtotutka yksinkertaisuudessaan valvoo sitä, onko auton vieressä tilaa kun kaistaa vaihdetaan. Mikäli kaistaa vaihdetaan ja tutka huomaa vieressä toisen auton niin tämän järjestelmä kertoo kuljettajalle vilkuttamalla LEDiä, joka sijaitsee sivupeilin vieressä ns. A-pilarissa. Jos toinen auto on jo lähellä tai lähestyy nopeasti niin järjestelmä hälyttää.

KAISTANVAIHTOTUTKAN toimintakaavio:



5.3.2 Matkan mittaaminen

Matkan mittaamiseen tarvitaan äänenkulku aika esteeseen ja äänennopeus. Seuraavassa esitetään ohjelmakoodi, joka toteuttaa kappaleessa 5.3.1 esitellyn lohkokaaavion.

Alla oleva ohjelmakoodi laskee kulkuajan jokaiselta anturilta. Ohjelma panee tulokset 8-bittiseen muuttujaan. 255 senttimetriä on maksimietäisyys, mikä on mahdollista mitata. Yhden askeleen muutos muuttujassa tarkoittaa yhtä senttiä etäisyydessä. Odotusajan arvo on siis laskettu äänennopeudesta, ja on 2 senttimetriin kuluva aika, kaksinkertainen siksi kun ääni kulkee kumpaankin suuntaan. Äänen nopeus 2 senttimetrillä vaihtelee lämpötilan mukaan 64 μ S:sta 56 μ S:n (-30..+40 °C), Liite 5.

```
void Measure_distance()
{
    int tempdist;
    for(channel=0;channel < 4;channel++) //channel (global)
    {
        output_a(channel<<2 & 0b00011100); //anturin valinta
        output_low(Inhibit); //anturi valittu
        bit_set(INTCON,INTE); //Sallitaan kaiunvastaanotto
        Send_frequency(); //lähetetään 40kHz taajuus purske
        bit_clear(INTCON,INTF); //nollataan kaiunvastaanotto lippu
        while(tempdist < 256 && Distance[channel] == 0)
        {
            if(bit_test(INTCON,INTF)) //jos kaikuvastaanotettu
                Distance[channel] = tempdist; //tallennetaan matka
            delay_us(Time); //2 senttimetriin kuluva aika.
            tempdist++; //lisätään matkaa 1cm
        }
        tempdist = 0;
    }
}
```

Distance[]-taulukon arvot ovat siis suoraan senttimetriä. Näiden mittauksien jälkeen tulee tuloksen käsittely lähinnä summeria varten. Hälytyksien raja-arvot voisivat olla käyttäjän valittavissa, mutta siitä ei olisi varsinaista hyötyä. Raja-arvot on ajateltu niin, että autoa ei päästettäisi 30cm lähemmäksi estettä, eli tällöin summeri soi jatkuvasti. Peruutusjutkassa on neljä eri raja arvoa, josta neljäs on jatkuva ääni. Ensimmäinen raja-arvo on 1,2m toinen 0,9m kolmas 0,6m. Ääni voisi olla portaattomasti tihenevä, mutta tällöin kuljettajalla ei olisi selvää tietoa matkasta. Portaallisesti muuttuvana kuljettaja voi arvioida äänestä etäisyyden. Ohjelman rakenne on suhteellisen yksinkertainen ja siihen voidaan helposti lisätä ominaisuus vaikka siihen että nopeasta matkan lyhenemisestä hälytetään. Kaistanvaihtotutka tekee myös hälytyksen peruuttaessa, mutta yksiportainen hälytys tulee jos matka on lyhempi kuin 20 senttimetriä.

Kaistanvaihtotutka toimii lähes samalla tavoin käyttää samaa mittaus aliohjelmaa mikä esiteltiin edellä. Suurin ero kaistanvaihtotutkassa peruutusjutkaan on raja-arvoissa. Kaistanvaihtotutka vilkuttaa LED:iä jos tutka saa signaalia takaisin, eli jos 255 senttimetrin päästä. 1,5m tekee hidastajuksen hälytyksen ja 1m jatkuvan.

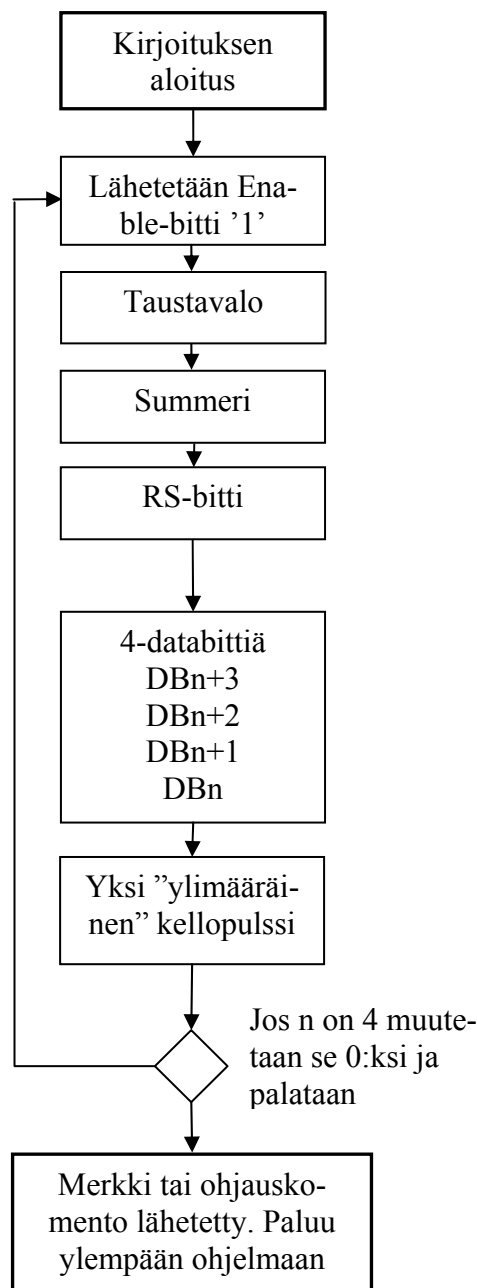
5.4 Näyttö

Useimmissa korkeamman tason PIC-kääntäjissä on suora tuki Hitachi 44720 yhteensopiville näytöille. Mutta tämä suora käsky toimii vain silloin kun näyttö kytketään suoraan PIC:n nastoihin. Tässä tapauksessa näyttöä ohjataan sarjamuotoisella datalla. Joten näyttöä ohjataan datakehyksellä, joka esiteltiin kappaleessa 4.3.1.

5.4.1 Näytönohjauksen toimintakaavio

Seuraavassa toimintokaavio siitä miten siirtokehys tehdään.

NÄYTÖNOHJAUKSEN toimintakaavio:



Kirjoitus aloitetaan kun tutkasta on saatu uusi tieto tai tietyn ajan välein.

Ensimmäisen "MSB-bitti" on aina "1" tämä "enable" näytön kun siirto on valmis.

Lähetetään tieto siitä onko taustavalo asetettu päälle.

Lähetetään tieto siitä onko summeri asetettu päälle.

Lähetetään RS-bitti, mikä kertoo onko lähetettävä data ohjaus vai merkki.

Databitit lähetetään kahdessa kehyksessä eli yksi komento tai merkki tarvitsee kaksi kehystä, ensimmäisenä lähetetään tavun yläosa ja sitten ala. Eli ensimmäisellä kierroksella n on 4.

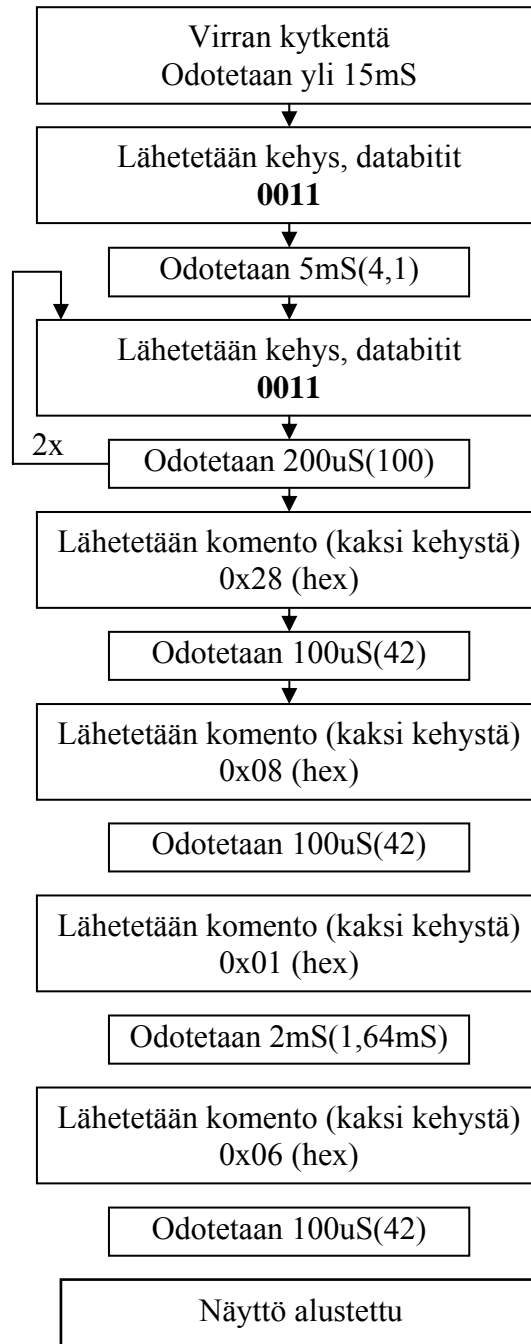
Lähetetään kello pulssi joka yhdessä enable-bitin kanssa virkistää näytön.

Jos lähetettiin ylempi tavu, palataan alkuun ja lähetetään alempi tavu.

Palataan takaisin ohjelmaan, joka sitten lähettää uuden merkin tai ohjauksen.

5.4.2 Näytön alustus

Ennen kuin näyttöön voidaan kirjoittaa, näyttö tarvitsee alustaa tietyllä rutiinilla. Rutiinissa tarvitsee odottaa näytöntoimintoja joten jokaisen datan lähetyksen jälkeen tulee odottaa hetki. RS-bitti on joka kohdassa "0".



Näyttöön kytketään virta sekunnin päästä siitä kun laitteeseen on kytketty virta.

Datakehukset lähetetään 8-bitin moodissa, mutta alimmat bitit ovat "0":ia joten yksi kehys riittää.

Kehys lähetetään kolme kertaa, toinen tapa olisi odottaa näytön lähettämää busyflag-bittiä. Tätä toimintoa ei käytetä, koska se vaatisi näytöltä lukua.

Lähetetään set funktion-komento, näyttö asetetaan 4-bittiseen tilaan, 2-riviseksi ja merkin kooksi 4*7 pikseliä. Tätä ei enää pysty muuttamaan ilman uudelleen käynnistystä.

Lähetetään display on/off-komento. Näyttö sammutetaan, kursori ja sen vilkunta pois.

Lähetetään display clear-komento. Näytön tyhjennys ja kursorin paikan nollaus.

Display entry mode-komento. Laitetaan kursori automaattisesti lisääväksi, ja näyttö ei "shiftaavaksi".

Näyttö on nyt alustettu, näytön saa päälle lähettämällä display on/off-komento 0x0C.

5.4.3 Merkin tai ohjaukaskäskyn lähetys

Alla oleva aliohjelma-koodi lähettää kaksi kehystä eli lähettää yhden komennon näytölle. Samalla aliohjelmalla voi lähettää niin merkin kuin ohjaukaskäskynkin. Aliohjelmalle annetaan parametrina lähetettävä data (8-bittinen **sdata**). **Clk_pulse**-aliohjelma lähettää aina yhdenpulssin, joka virkistää ja lataa datan siirtorekisteriin. **For** luupilla lähetetään 4-databittiä, **bit_test** aliohjelma ”poimii” **sdata**-muuttujasta indeksi **i** osoittaman bitin (**i** = 7..0).

```
void sendtolcd(byte sdata)
{
    int i,j;
    i = 7;
    while(i != 0) // niin kauan kuin i on erisuuri kuin 0
    { // eli 2 kierrosta
        Output_high(LCD_data); //Lähetetään enable
        clk_pulse(); //kello pulssi
        Output_bit(LCD_data, backlight); //lähetetään taustavalo
        clk_pulse();
        Output_bit(LCD_data, buzzer); //lähetetään sumneri
        clk_pulse();
        for(j=0;j<4;i--,j++) //Lähetetään 4 databittiä
        {
            Output_bit(LCD_data,bit_test(sdata,i));
            clk_pulse();
        }
        Output_high(LCD_data);
    }
}
```

5.5 Lämpötilan laskenta

Alla aliohjelma siitä miten lämpötilan laskenta on toteutettu. Ensiksi tehdään AD-muunnos. AD-muunnoksen arvoa verrataan taulukoituihin arvoihin ja etsitään se väli taulukosta jossa edellinen arvo on pienempi ja seuraava suurempi. Näiden taulukoiden arvojen välistä katsotaan mihin AD-muunnosarvo sijoittuu. Lämpötilan desimaali luku lasketaan siitä kohdasta mihin muunnosarvo sijoittuu. Jos muunnosarvo sijoittuu yli puolen välin niin lisätään kokonaislukuasteisiin yksi. Toiminnan selitys on aika vaikeita, mutta alla oleva koodi selittää toimintaa.

```
void Count_temperature(int channel,int temperature[3] )
{
    int AD_value; //temperature[0] = lämpötilan kokonaisluku [1] = desimaaliluku
    int i; // [2] 0 = pos / 1 = neg
    int difference; //taulukon arvojen väli
    set_adc_channel(channel); //mistä kanavasta otetaan AD
    AD_value = read_adc(); //luetaan arvo

    for(i=0;AD_value < ADTEMPERATURES[i];i++); //etsitään taulukosta ensin
    if( i > 32) //mäinen pienempi arvo
    {
        difference = ADTEMPERATURES[i+1] - ADTEMPERATURES[i];
        //taulukon arvojen väli
    }
}
```

```
        temperature[1] = AD_value - ADTEMPERATURES[i]; //muutetaan väli
                                                    //lineaarisesti tarkkailtavaksi
        temperature[1] = temperature[1] / difference * 20; //lasketaan desi-
    }                                                    //maalit 0,0... 2.0
temperature[0] = TEMPERATURES[i]; //katsotaan TEMPERATURE taulukosta
                                                    //lämpötila indeksillä i
if(i < 18) //onko negatiivinen
    temperature[2]++;
if(temperature[1] > 9) //lisätään lämpötilaan 1, jos dvalue 10 tai yli
    {
        temperature[0]++;
        temperature[1]-=10;
    }
}
```


6 Käyttöliittymä

Peruutustutka ei sinänsä tarvitsisi näyttöä, eikä sen takia minkäänlaista käyttöliittymää. Tähän tutkaan on suunniteltu lisätoimintoja ja laajennusvaraa. Eli tähän tutkaan pystyy rakentamaan lisälaitteita, jotka sitten käyttävät samaa näyttöä. Tällainen lisälaitte voisi olla vakionopeudensäädin. Mikrokontrollerin käyttöaste silloin kun ei peruuteta, on hyvin pieni tällöin, prosessointi aikaa on hyvin käytössä muuhun toimintaa. Erilaisia näyttötiloja voi helposti lisätä, tässä kappaleessa esitellään oleelliset.

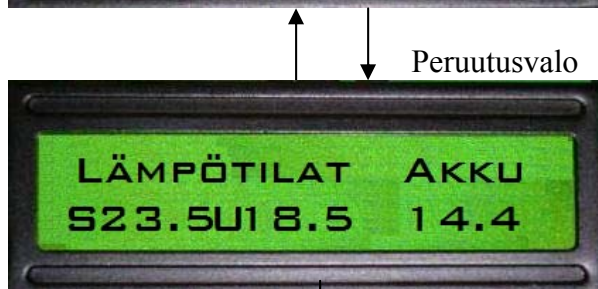


Kuva 12 Näyttöpaneeli

Kuva 12 Näyttöpaneeli. Piirros näyttöpaneelistä, jossa on kuvattu millainen peruutustutkan näyttöpaneeli tulee olemaan. Nappi A:lla selataan eri näyttötiloja ja nappi B:llä muutetaan asetuksia. Kuvan summeri kuvaa reikää, josta sitten summerin ääni pääsee ohjaamoon.



Peruutustutkan näyttö peruutustilassa. Lyhin matka on merkattu alleviivauksella.



Peruutusvalo

Peruutustutkan perusnäyttö, tämä näyttö on silloin kun mikään toiminto ei ole päällä. Toimintojen jälkeen tämä näyttö palaa takaisin.

Nappi A



Tämä näyttö kertoo taustakohinan määrän 6-bittisellä luvalla. Näyttö kertoo digitaalisen potentiometrin arvon, ei siis mitään absoluuttista lukua.

7 Rakentaminen

Laitteen rakentaminen tehtiin aivan itse, sen takia laite on pyritty pitämään yksinkertaisena. Laitteen oma valmistaminen ei ole taloudellisesti kannattavaa, tällainen laite tulee maksamaan noin 150€. Sillä saa jo lähes millaisen vaan kaupasta. Tämä luku on jaettu konkreettisiin eri rakennusvaiheista. Kuvia valmiista tutkamoduulista liitteissä,

7.1 Piirilevy

Peruutustutkan piirilevy on kaksipuolinen, koska kytkentä on kuitenkin sen verran monimutkainen, lähinnä suurimmaksi ongelmaksi tuli se kun PIC on keskuskomponentti mihin lähes kaikkien johtimien toinen pää on. Piirilevy piirrettiin samalla Eagle layout editor-ohjelmalla kuin kytkentäkin. Piirilevyn vetoihin käytettiin ohjelman niin sanottua Autorouter-toimintoa, joka johdottaa automaattisesti levyn. Osa johtimista korjattiin käsin. Piirilevy tehtiin perinteisellä syövytysmenetelmällä. Näyttömoduulin piirilevy on yksinkertainen kun siinä on vain muutama komponentti, sen piirilevy on yksipuoleinen. Piirilevykuvat ovat liitteenä Liite 9, Liite 10, Liite 11.

7.2 Komponentit

Komponentit on normaaleja läpijuotettavia. Mikään komponentti ei ole erityisen harvinainen, niitä saa lähes jokaisesta liikkeestä ainakin tilaamalla. Tähän prototyyppiin komponentit on ostettu Farnellista (www.farnell.fi) ja Elfasta (www.elfa.se/fi). Komponenttilistaus on liitteessä, Liite 12.

7.3 Kotelointi

Koteloiksi valittiin ABS-muoviset laatikot, syynä tähän valintaan oli lähinnä halpa-hinta koska koteloille ei ollut varsinaisesti mitään vaatimuksia. Näyttömoduulinkotelo haluttiin niin pienenä kuin mahdollista, että se mahtuu kattoverhoilun päälle. Tutkamoduulin laatikoksi valittiin mahdollisimman pieni kotelo, johon piirilevy mahtuu. Laatikossa ei tarvitse olla tilaa liitäntäjohdolle, koska ulkoisten liitäntien liitokset ovat johdoissa laitteen ulkopuolella. Koteloista mittatiedot ovat liitteissä Liite 13, Liite 14.

7.4 Liitännät

Peruutustutka liitetään auton sähköjärjestelmään perinteisillä Abico-liittimillä. Abicoiden valinta liittimeksi johtui lähinnä liittimen helposta saatavuudesta yksinkertaisuudesta. Tällaisen kytkentätapa sopii hyvin tähän sovellukseen, kun johdot lähtevät hyvin satunnaisesti suuntiin, eikä ulkovaipasta olisi hyötyä. Johtoon on myös saatu yksinkertaisesti pistettyä sulake. Näyttö ja anturit on liitetty tutkamoduuliin RJ-45 liittimellä, tämä on valittu pienen koon ja yksinkertaisen liitännän takia. Johdona käytetään perinteistä verkkokaapeli, näyttömoduulin ja tutkamoduulin yhdistämiseen voi siis käyttää cat5 verkkokaapeli. Näyttömoduulissa ei ole liitintä, vaan johto on määrämittainen ja tinataan kiinni piirilevyyn.

8 Asennus

Tässä luvussa käsitellään peruutustutkalaitteen asennus autoon ja auton sähköjärjestelmään. Asennus on suunniteltu tehtäväksi Ford Focusiin v.m.2001.

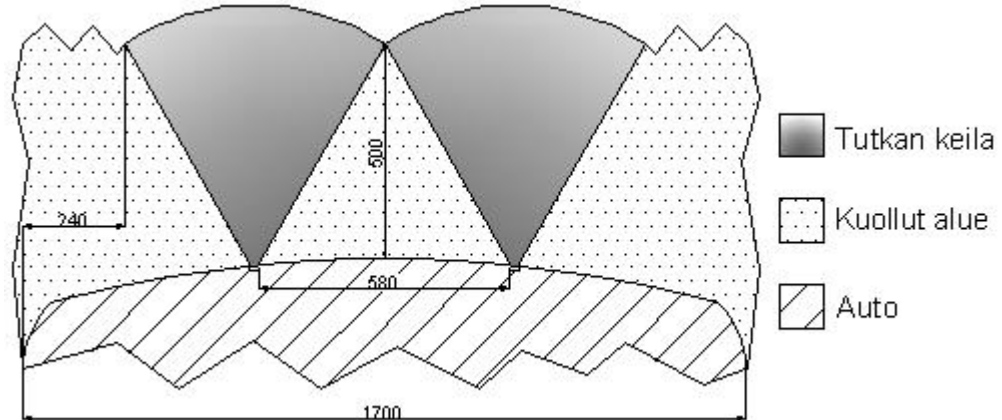
8.1 Mekaaninen asennus

Peruutustutka asennetaan autoon kiinteästi, sitä ei enää voida poistaa lähinnä tehtävien porauksien takia. Auton takapuskuriin pitää tehdä neljä reikää taaksepäin peruutustutkanantureita varten ja kaksi reikää puskurin reunoihin kaistanvaihtotutkanantureita varten. Tätä porausta ja johtojen vetoa varten puskuri tulee irrottaa autosta.

Peruutustutka anturi tulee sijoittaa siten, että koko perän ala tulee katettua, jokaisen anturin lähetyks- ja vastaanottokeila on 90 astetta. Neljällä anturilla saa hyvin katettua koko perän alueen ja ”kuollut alue” aivan perän takana jää pieneksi. Seuraavassa kuvissa siitä miten antureiden määrä vaikuttaa tutkien keilojen kattavuuteen ja millainen kuollut alue jää.

Alla olevat kuvat kuvaavat sitä, miten eri määrä ultraääniantureita kattavat perän alueen. Anturien rakenteesta johtuen keilat eivät levene rajatta vaan levenemiskulma pienenee. Tätä ei kuvista näe, koska on oletettu että n. puolen metrin matkalla leveneminen on vakioita.

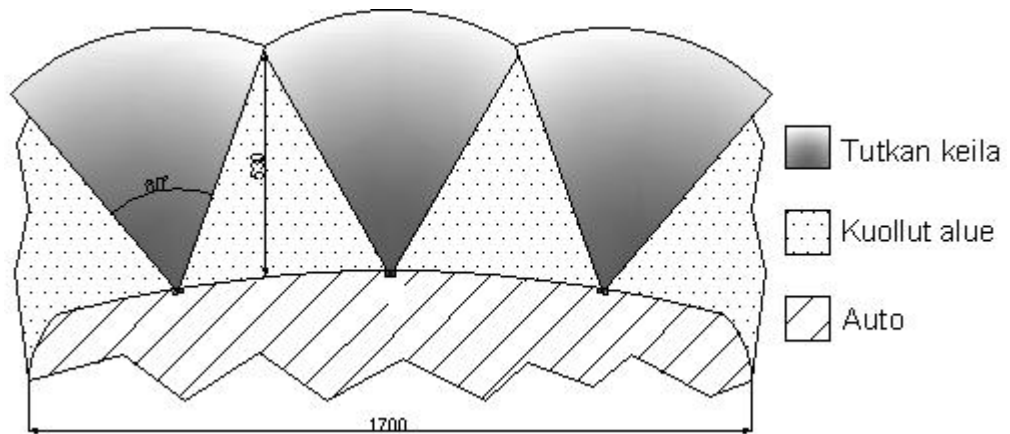
Kaksi anturia:



Kuva 13 Ultraäänen peittoalue 2-anturia

Kahta anturia käytettäessä koko perän aluetta ei pystytä valvomaan. Ja anturien väliin jää suuri ”kuollut alue”. Tällaista ratkaisua käytetään jossain halvimmista tarkvetutkissa.

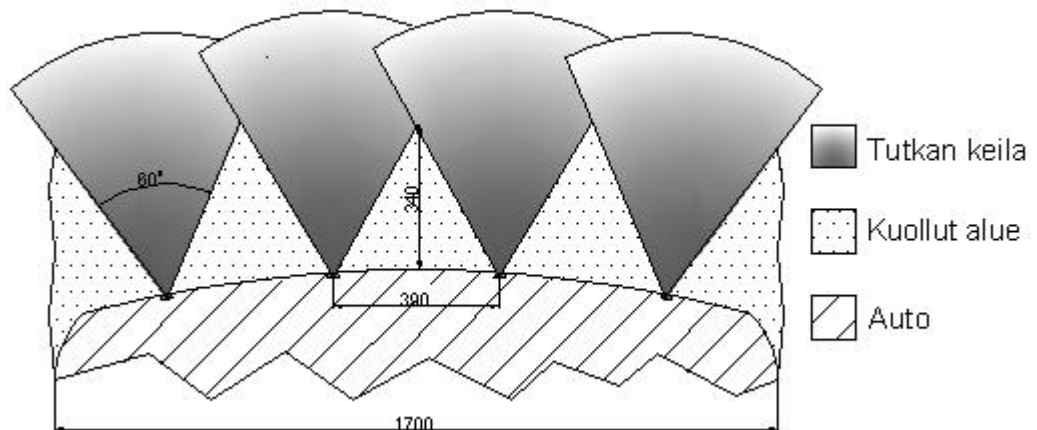
Kolme anturia:



Kuva 14 Ultraäänen peittoalue 3-anturia

Kolme anturia kattaa jo koko perän alueen. Anturien välinen ”kuollut alue” on aikalailla yhtä suuri, tosin se riippuu asettelusta. Kolme anturia on jo hyvin toimiva vaihtoehto, mutta tällaisen ratkaisun huonopuoli on että se ei sovi autoon jossa on kiinteä vetokoukku.

Neljä anturia:



Kuva 15 Ultraäänen peittoalue 4-anturia

Neljä anturia on kaikista yleisintä, koska perän kattavuus on hyvä. Kattavuus on ehkä liiankin hyvä koska on mahdollisuus, että tutka hälyttää myös ohi menevistä esteistä. Kuollut alue anturien välillä on noin 30cm syvä riippuen asettelusta. Anturit on aseteltu kuvassa tasavälisesti noin 40cm välein. Tässä kuvassa anturit on asennettu peränmuodon mukaan. Parhaaseen tulokseen pääsisi jos kaikki anturit osoittaisivat suoraan taaksepäin, mutta tämä tekee asennuksesta vaikeampaa.

Kaistanvaihtotutkien asennus tehdään puskurin sivuihin siten että anturien keila osoittaa vähän takaviistoon. Tämä sen takia, että varmistutaan siitä että auto todellakin mahtuu vaihtamaan kaistaa, eikä siten aja aivan ohitettavan auton eteen. Tällaisella asettelulla on myös sellainen hyöty, että jos kaistaa vaihdettaessa vieressä olevan auton nopeus on suurempi, niin tutka ehtii varoittamaan vähän nopeammin. Kaistanvaihtotutkasta ei luonnollisesti ole hyötyä, jos vierelle tulevan auton nopeus on selkeästi suurempi. Tällainen tilanne näkyy jo peileistä, tutkan tarkoitus siis ei ole se, että peilejä ei tarvitsisi enää käyttää vaan lähinnä siinä, että päätä kääntämällä ei tarvitsisi varmistaa vieressä oleva tila.

Anturit asennetaan korkeussuunnassa noin autonrenkaiden keskiöntasolle tai sen yläpuolelle puskuriin, pystysuoraan kohtaan. Ellei aivan pystysuoraa kohtaa ole niin siten että anturin osoittavat alaspäin. Tämä sen takia että yleensä kaikki esteet ovat maassa, mutta alaspäin suuntaus ei saa olla suuri. Jos anturit suunnataan liian paljon alaviistoon, niin on mahdollista että tutka alkaa mitata etäisyyksiä maassa olevista mahdollisista peilauskohdista.

8.1.1 Anturin asennus

Anturinasennus on suhteellisen ikävää hommaa, koska auton puskuriin pitää tehdä kuusi reikää ja reikien teko pitää onnistua kerrasta. Anturit asennetaan siten, että aluksi suunnitellaan ja mitataan anturien paikat. Sitten kun paikat on merkattu halkaisijaltaan 10mm reikä. Tämän jälkeen porataan senkkaus-terällä sisäpuolelta ison reikään siten, että sisäpuolen halkaisija on n.17mm. Anturi saa tulla vähän puskurin ulkopuolelle.



Kuva 16 UTEC4018F

UTEC4018F antureissa on pientä eroa niin sähköisessäkin värähtelytaajuudessa kuin ulkokuoressa. Ulkokuoren erot otetaan huomioon asennusvaiheessa. Anturien kiinnitys on helpointa tehdä liimalla kartiokohdasta.

8.1.2 Näyttömoduulin asennus

Näyttömoduuli on hyvä asentaa siten että se on mahdollisimman lähellä taustapeliä, tämä tarkoittaa lähinnä sitä, että näyttö asennetaan kattoon. Katto verhoiluun on siis tehtävä reikä näytölle. Kattoon tehtävä reikä on hiukan pienempi kuin näyttömoduulinlaatikko. Näyttömoduuli siis laitetaan kattoverhoilun päälle ja näyttömoduuli kiristetään kattoon etulevyllä.

8.2 Sähköasennus

Peruutustutka liitetään tietyiltä osin auton sähköjärjestelmään. Käyttövirta otetaan auton akkujännitteestä. Paras tapa olisi ottaa sähkö omalla johdollaan suoraan akusta, tällöin pitää laitteelle vetää erikseen herätejohto, joka kertoo laitteelle koska autossa on virta päällä. Peruutustutkalaite kuluttaa suhteellisen vähän virtaa, joten virta voidaan ottaa haaroittamalla esimerkiksi tupakansytyttimeltä tai muusta vastaavasta paikasta johon tulee virta vain auton virran ollessa päällä. Tällöin erillistä herätejohtoa ei tarvita. Käyttövirran lisäksi laite vaatii vielä tiedon siitä peruuttaako auto ja onko vilkut päällä. Nämä signaalit on yksinkertaisinta ottaa peräkoukun virranoton johdosta niin sanotuilla rosvoliitimillä. Nämä ohjaussignaalit eivät juuri tarvitse virtaa joten ne eivät vaikuta vilkkujen releiden toimintaan.

8.3 Asennus autoon

Tutka-moduuli asennetaan auton takaosaan paikkaan, jossa se ei ole alttiina lialle ja siihen on suhteellisen helppo päästä käsiksi. Tutka-moduuli asennetaan auton takaosaan lähinnä siksi, että tutkien johdot eivät olisi pitkät ja vilkkujen- ja takavalon signaalit ovat lähellä. Näyttö-moduulin kaapelista tulee pitkä, koska se kulkee auton kynnyskoteloa pitkin eteen ja ns. a-pilaria pitkin auton kattoverhoilun väliin. Seuraavassa kuva siitä miten kyseiset moduulit ja johdot on suunniteltu asennettavan.

9 Yhteenveto

Tämä peruutustutkaprojekti on paljon aluksi luultua suurempi projekti, työn olisi voinut oikeastaan jakaa kahdeksi työksi. Loogisin jako olisi ollut ohjelmisto ja itse laitteistonsuunnittelu. Siksi tästä työstä puuttuu joitakin asioita kuten itse laitteen rakentaminen ja testaus. Mikäli näihin olisi edes kohtuullisella tasolla perehtynyt, niin tämä työ olisi selkeästi pidempi. Työn kirjoitusvaiheessa täytyi määrätä ne asiat johon työssä paneudutaan muita tarkemmin. Nämä asiat ovat ohjelmisto ja kytkentä. Ohjelmisto ei suoraan todella tärkeässä osassa tällaisessa sulautetussa järjestelmässä. Kuten tämänkin laitteen kytkennästä huomaa itse kytkentä on varsin yksinkertainen.

Ultraäänen fysiikkaan ja muihin ei asioihin, jotka ei suoranaisesti liity peruutustutkan toiminnan ymmärtämiseen ei juurikaan syvennytty. Työ siis pyrittiin pitämään suhteellisen käytännönläheisenä. Kirjoituksessa huomaa välillä lähes opeuskirjamaisia kappaleita, tällaista on erityisesti luvussa 8 Asennus. Tämä johtuu siitä kun asennuksesta ei suoranaisesti ole tehty suunnitelmaa, koska se on hyvin auto kohtaista.

Peruutustutkalaitteistolla on paljon varaa laajentua. Tämä johtuu lähinnä siitä, että laitteen käyttöaika on hyvin pieni, PIC:llä on paljon aika tehdä kaikenlaista muuta eteenpäin mennessä. Yksi mahdollinen lisäominaisuus olisi vakionopeuden säädin, mihin PIC:n teho riittäisi hyvin, tai vaikka varashälytys. Yksinkertaisempi lisäominaisuus voisi olla ajovaloautomaatiikka, tämän lisääminen olisi helppoa koska laitteella on tieto siitä koska auto on käynnissä. Yksi ominaisuus joka tässä työssäkin mainittiin, on anturien lämmitys, tähän laitteessa on jo suora tuki mutta sen tarpeellisuus ei ole vielä selvillä.

Tämä työ on lähinnä suunnitelma ja itse laite ei ollut aivan valmis työn luovutusvaiheessa, joten laitteen toiminnasta ei ole täyttä varmuutta. Mutta laite kyllä rakennetaan loppuun ja toimivaksi. Liitteenä olevista lähdekoodeista ei saa toimivaa laitetta, johtuen c-kääntäjän ominaisuudesta, mutta siitä näkee tavat kuinka ohjelma toimii.

Toivottavasti tämä insinöörityö oli kiinnostavaa luettavaa ja antoi kuvan siitä miten ultraääninen peruutustutka toimii.

Lähdeluettelo

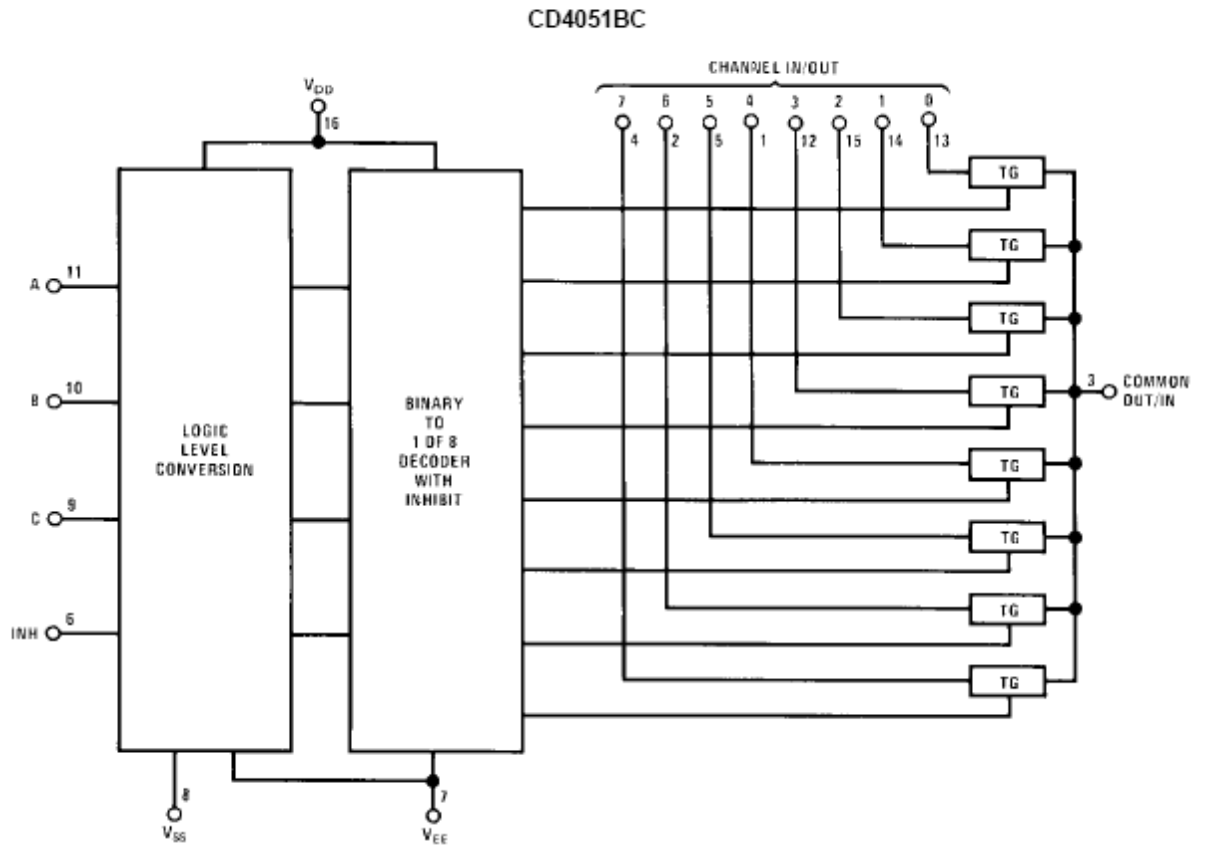
INTERNET

- /1/ <http://fi.wikipedia.org/wiki/LED>
- /2/ http://en.wikipedia.org/wiki/PIC_microcontroller
- /3/ http://www.machina.hut.fi/kurssit/41/190/Paikannus_ho.pdf
- /4/ <http://personal.inet.fi/yritys/creativesco/Alkuosa.htm>
- /5/ <http://fi.wikipedia.org/wiki/%C3%84%C3%A4nennopeus>
- /6/ <http://www.probyte.fi/info/lcd.htm>

+ Käytettyjen komponenttien datakirjat.

Liitteet

Liite 1 CD4051BC datalehti Logic Diagrams



Truth Table

INPUT STATES				"ON" CHANNELS		
INHIBIT	C	B	A	CD4051B	CD4052B	CD4053B
0	0	0	0	0	0X, 0Y	cx, bx, ax
0	0	0	1	1	1X, 1Y	cx, bx, ay
0	0	1	0	2	2X, 2Y	cx, by, ax
0	0	1	1	3	3X, 3Y	cx, by, ay
0	1	0	0	4		cy, bx, ax
0	1	0	1	5		cy, bx, ay
0	1	1	0	6		cy, by, ax
0	1	1	1	7		cy, by, ay
1	*	*	*	NONE	NONE	NONE

*Don't Care condition.

Liite 2 TS925 datalehti



TS925

RAIL TO RAIL HIGH OUTPUT CURRENT QUAD OPERATIONAL AMPLIFIER

- RAIL TO RAIL INPUT AND OUTPUT
- LOW NOISE : $9\text{nV}/\sqrt{\text{Hz}}$
- LOW DISTORTION
- HIGH OUTPUT CURRENT : 80mA
(able to drive 32Ω loads)
- HIGH SPEED : 4MHz , $1.3\text{V}/\mu\text{s}$
- OPERATING FROM 2.7V TO 12V
- LOW INPUT OFFSET VOLTAGE : $900\mu\text{V}$
max. (TS925A)

- ADJUSTABLE PHANTOM GROUND ($V_{CC}/2$)
- STANDBY MODE

- ESD INTERNAL PROTECTION : 2kV
- LATCH-UP IMMUNITY
- MACROMODEL INCLUDED IN THIS
SPECIFICATION

DESCRIPTION

The TS925 is a RAIL TO RAIL quad BiCMOS operational amplifier optimized and fully specified for 3V and 5V operation.

High output current allows low load impedances to be driven. An internal low impedance PHANTOM GROUND eliminates the need for an external reference voltage or biasing arrangement.

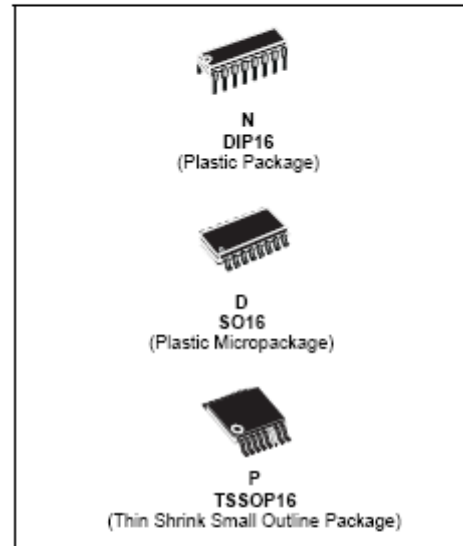
The TS925 exhibits a very low noise, low distortion and high output current making this device an excellent choice for high quality, low voltage or battery operated audio/telecom systems.

The device is stable for capacitive loads up to 500pF .

When the STANDBY mode is enabled, the total consumption drops to $6\mu\text{A}$ ($V_{CC} = 3\text{V}$).

APPLICATIONS

- Headphone amplifier
- Soundcard amplifier, piezoelectric speaker
- MPEG boards, multimedia systems, ...
- Cordless telephones and portable communication equipment
- Line driver, buffer
- Instrumentation with low noise as key factor

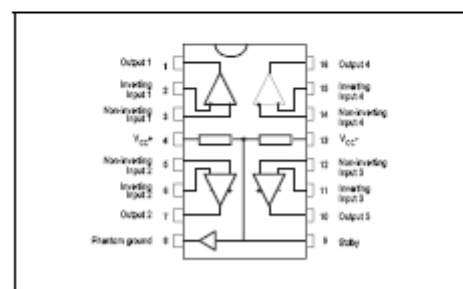


ORDER CODE

Part Number	Temperature Range	Package		
		N	D	P
TS925I	-40°C , $+125^{\circ}\text{C}$	*	*	*

N = Dual in Line Package (DIP)
D = Small Outline Package (SO) - also available in Tape & Reel (OT)
P = Thin Shrink Small Outline Package (TSSOP) - only available
in Tape & Reel (PT)

PIN CONNECTIONS (top view)



Liite 3 AD5220 datalehti



Increment/Decrement
Digital Potentiometer

AD5220

- FEATURES**
 128 Position
 Potentiometer Replacement
 10 kΩ, 50 kΩ, 100 kΩ
 Very Low Power: 40 μA Max
 Increment/Decrement Count Control

- APPLICATIONS**
 Mechanical Potentiometer Replacement
 Remote Incremental Adjustment Applications
 Instrumentation: Gain, Offset Adjustment
 Programmable Voltage-to-Current Conversion
 Programmable Filters, Delays, Time Constants
 Line Impedance Matching
 Power Supply Adjustment

GENERAL DESCRIPTION

The AD5220 provides a single channel, 128-position digitally controlled variable resistor (VR) device. This device performs the same electronic adjustment function as a potentiometer or variable resistor. These products were optimized for instrument and test equipment push-button applications. A choice between bandwidth or power dissipation are available as a result of the wide selection of end-to-end terminal resistance values.

The AD5220 contains a fixed resistor with a wiper contact that taps the fixed resistor value at a point determined by a digitally controlled UP/DOWN counter. The resistance between the wiper and either end point of the fixed resistor provides a constant resistance step size that is equal to the end-to-end resistance divided by the number of positions (e.g., $R_{STEP} = 10\text{ k}\Omega / 128 = 78\ \Omega$). The variable resistor offers a true adjustable value of resistance, between the A terminal and the wiper, or the B terminal and the wiper. The fixed A-to-B terminal resistance of 10 kΩ, 50 kΩ, or 100 kΩ has a nominal temperature coefficient of 800 ppm°C.

The chip select \overline{CS} , count CLK and $\overline{U/D}$ direction control inputs set the variable resistor position. These inputs that control the internal UP/DOWN counter can be easily generated with mechanical or push button switches (or other contact closure devices). External debounce circuitry is required for the negative-edge sensitive CLK pin. This simple digital interface eliminates the need for microcontrollers in front panel interface designs.

The AD5220 is available in both surface mount (SO-8) and the 8-lead plastic DIP package. For ultracompact solutions selected models are available in the thin μSOIC package. All parts are guaranteed to operate over the extended industrial temperature range of -40°C to +85°C. For 3-wire, SPI compatible interface applications, see the AD7376/AD8400/AD8402/AD8403 products.

FUNCTIONAL BLOCK DIAGRAM

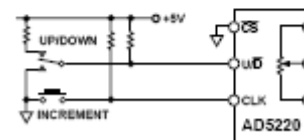
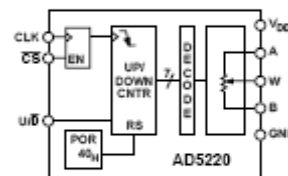


Figure 1. Typical Push-Button Control Application

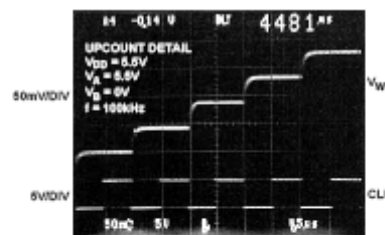


Figure 2a. Stair-Step Increment Output

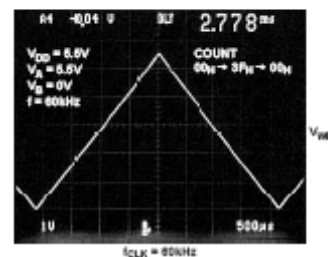
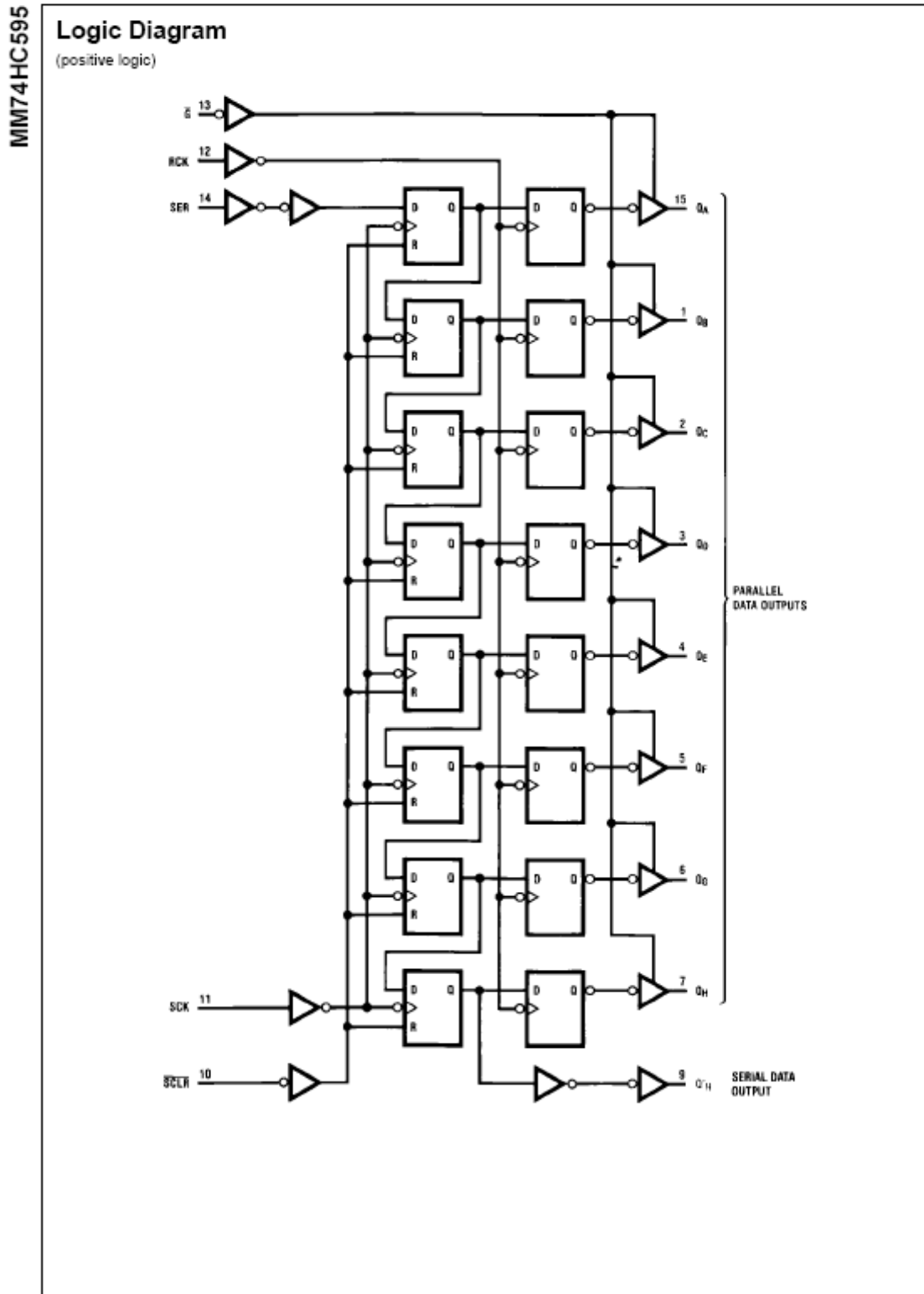


Figure 2b. Full-Scale Up/Down Count

Liite 4 74HC595 sisäinen kytkentä



Liite 5 Lämpötilanmittaus taulukko

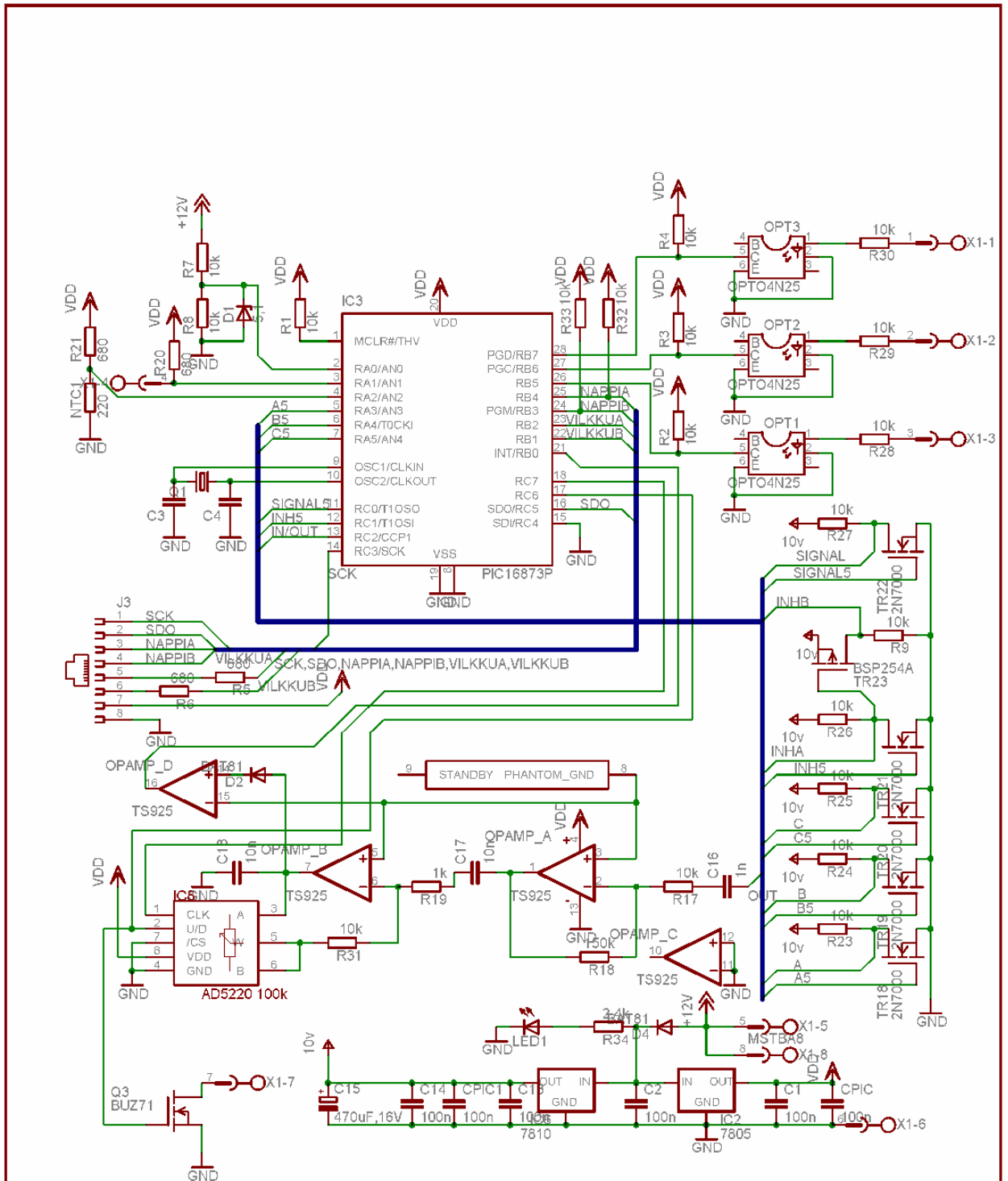
NTC /ohm	T / °C	AD-muunnos	AD-ero	Indeksi	Jännite	äänennopeus uS/2cm
126	40,0	160		1	0,78	56
150	35,0	186	26	2	0,91	57
181	30,1	215	30	3	1,05	57
189	29,0	222			1,09	57
196	28,0	229	13	4	1,12	57
204	27,0	236			1,15	58
212	26,0	243	14	5	1,19	58
220	25,0	250			1,22	58
229	24,0	258	14	6	1,26	58
238	23,0	266			1,30	58
248	22,0	273	16	7	1,33	58
258	21,0	282			1,37	58
269	20,0	290	17	8	1,42	58
280	19,0	299			1,46	58
291	18,0	307	17	9	1,50	58
303	17,0	315			1,54	59
315	16,0	324	17	10	1,58	59
328	15,0	333			1,63	59
343	14,0	343	19	11	1,68	59
358	13,0	353			1,72	59
372	12,0	362	19	12	1,77	59
388	11,0	372			1,82	59
404	10,1	382	20	13	1,86	59
424	9,0	393			1,92	59
442	8,0	403	21	14	1,97	59
461	7,0	414			2,02	60
482	6,0	425	22	15	2,07	60
503	5,0	435			2,13	60
526	4,0	446	22	16	2,18	60
550	3,0	458			2,24	60
576	2,0	470	23	17	2,29	60
602	1,0	481			2,35	60
628	0,0	492	22	18	2,40	60
657	-1,0	503			2,46	60
689	-2,0	515	24	19	2,52	61
721	-3,0	527			2,57	61
754	-4,0	538	23	20	2,63	61
789	-5,0	550			2,69	61
828	-6,0	562	24	21	2,75	61
867	-7,0	574			2,80	61
909	-8,0	586	23	22	2,86	61
954	-9,0	598			2,92	61
1000	-10,0	609	24	23	2,98	61
1048	-11,0	621			3,03	62
1100	-12,0	633	23	24	3,09	62
1155	-13,0	645			3,15	62
1213	-14,0	656	23	25	3,20	62
1275	-15,0	668			3,26	62
1339	-16,0	679	23	26	3,32	62
1407	-17,0	690			3,37	62
1479	-18,0	701	22	27	3,42	62
1556	-19,0	713			3,48	63
1637	-20,0	723	22	28	3,53	63
1724	-21,0	734			3,59	63
1815	-22,0	745	21	29	3,64	63
1909	-23,0	755			3,69	63
2012	-24,0	765	20	30	3,74	63
2119	-25,0	775			3,79	63
2236	-26,0	785	20	31	3,83	63
2485	-28,0	804			3,93	64
2766	-30,0	822	37	32	4,01	64

Liite 6 NTC-vastuksen laskentavakiot

R ₂₅ (Ω)	B _{25/85} -VALUE	CATALOGUE NUMBER 2322 640 6....				COLOUR CODE (see Fig.1 and note 1)		
		R ₂₅ ±2%	R ₂₅ ±3%	R ₂₅ ±5%	R ₂₅ ±10%	I	II	III
3.3	2880 K ±3%	4338	6338	3338	2338	orange	orange	gold
4.7	2880 K ±3%	4478	6478	3478	2478	yellow	violet	gold
6.8	2880 K ±3%	4688	6688	3688	2688	blue	grey	gold
10	2990 K ±3%	4109	6109	3109	2109	brown	black	black
15	3041 K ±3%	4159	6159	3159	2159	brown	green	black
22	3136 K ±3%	4229	6229	3229	2229	red	red	black
33	3390 K ±3%	4339	6339	3339	2339	orange	orange	black
47	3390 K ±3%	4479	6479	3479	2479	yellow	violet	black
68	3390 K ±3%	4689	6689	3689	2689	blue	grey	black
100	3560 K ±0.75%	4101	6101	3101	2101	brown	black	brown
150	3560 K ±0.75%	4151	6151	3151	2151	brown	green	brown
220	3560 K ±0.75%	4221	6221	3221	2221	red	red	brown
330	3560 K ±0.75%	4331	6331	3331	2331	orange	orange	brown
470	3560 K ±0.5%	4471	6471	3471	2471	yellow	violet	brown
680	3560 K ±0.5%	4681	6681	3681	2681	blue	grey	brown

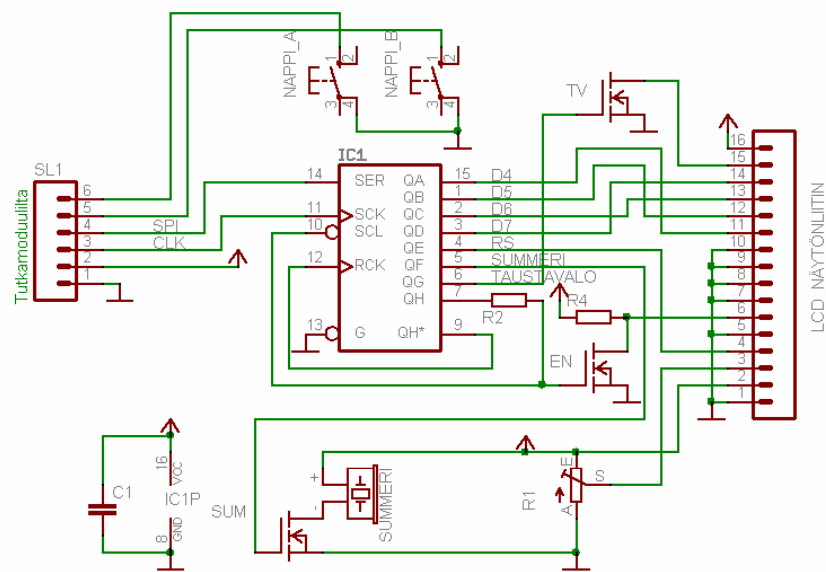
B _{25/85} -VALUE (K)	A	B (K)	C (10 ⁵ K ²)	D (10 ⁶ K ³)	A ₁ (10 ⁻³)	B ₁ (10 ⁻⁴ K ⁻¹)	C ₁ (10 ⁻⁶ K ⁻²)	D ₁ (10 ⁻⁷ K ⁻³)
2880	-9.094	2251.74	229098	-27.4482	3.354016	3.495020	2.095959	4.260615
2990	-10.2296	2887.62	132336	-25.0251	3.354016	3.415560	4.955455	4.364236
3041	-11.1334	3658.73	-102895	0.516652	3.354016	3.349290	3.683843	7.050455
3136	-12.4493	4702.74	-402687	31.96830	3.354016	3.243880	2.658012	-2.70156
3390	-12.6814	4391.97	-232807	15.09643	3.354016	2.993410	2.135133	-8.05672
3528 ⁽¹⁾	-12.0596	3687.667	-7617.13	-5914730	3.354016	2.909670	1.632136	0.719220
3528 ⁽²⁾	-21.0704	11903.95	-2504699	247033800	3.354016	2.933908	3.494314	-7.71269
3560	-13.0723	4190.574	-47158.4	-11992560.91	3.354016	2.884193	4.118032	1.786790
3740	-13.8973	4557.725	-98275	-7522357	3.354016	2.744032	3.666944	1.375492
3977	-14.6337	4791.842	-115334	-3730535	3.354016	2.569355	2.626311	0.675278
4090	-15.5322	5229.973	-160451	-5414091	3.354016	2.519107	3.510939	1.105179
4190	-16.0349	5459.339	-191141	-3328322	3.354016	2.460382	3.405377	1.034240
4370	-16.8717	5759.15	-194267	-6869149	3.354016	2.367720	3.585140	1.255349
4570	-17.6439	6022.726	-203157	-7183526	3.354016	2.264097	3.278184	1.097628

Liite 7 Tutkamoduulin kytkentäkaavio



Peruutustutka / Juha Silén	
TITLE: tutkamoduuli	
Document Number:	REV:
Date: 27.05.2006 20:33:29	Sheet: 1/2

Liite 8 Näyttömoduulin kytkentäkaavio



Insinööriyö / Juha Silén

TITLE: Näyttömoduuli

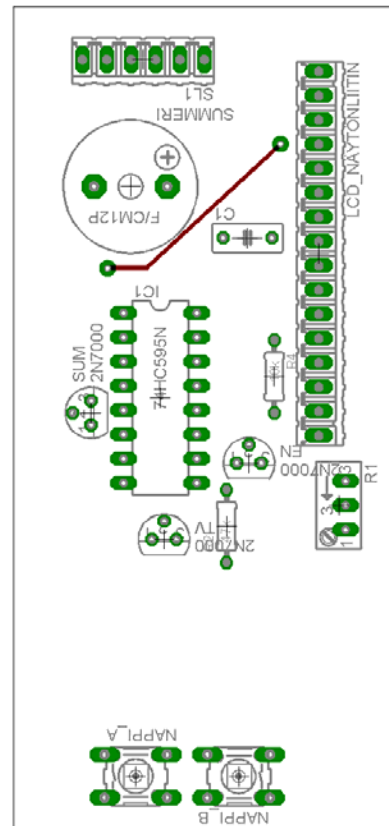
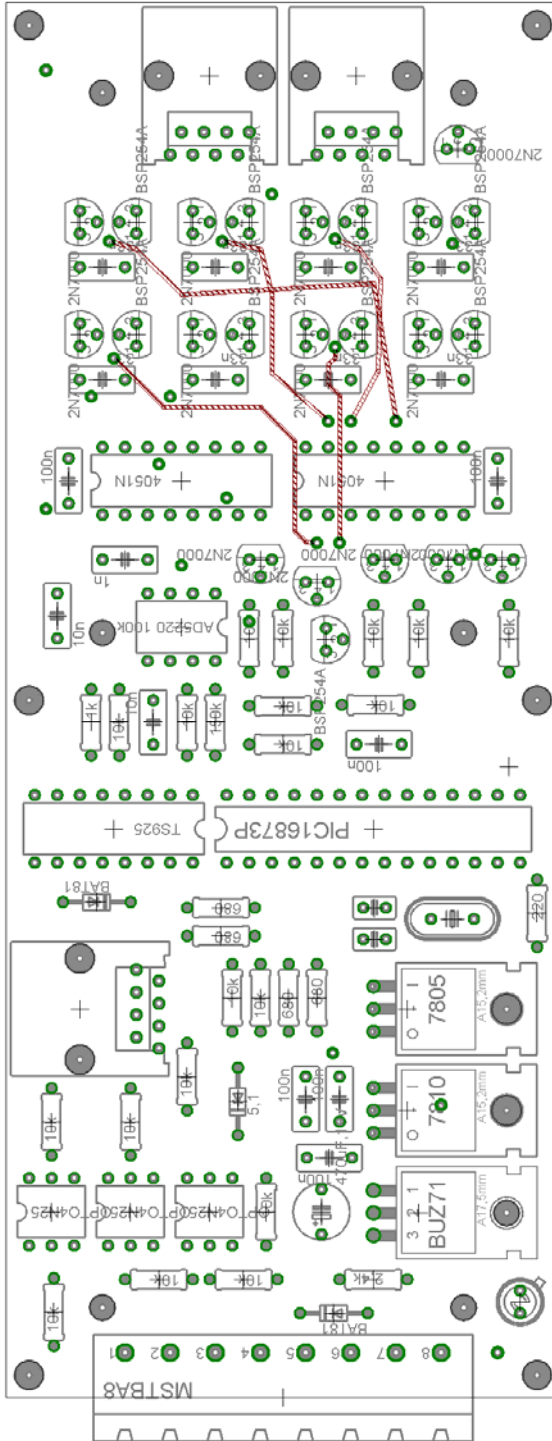
Document Number:

REV:

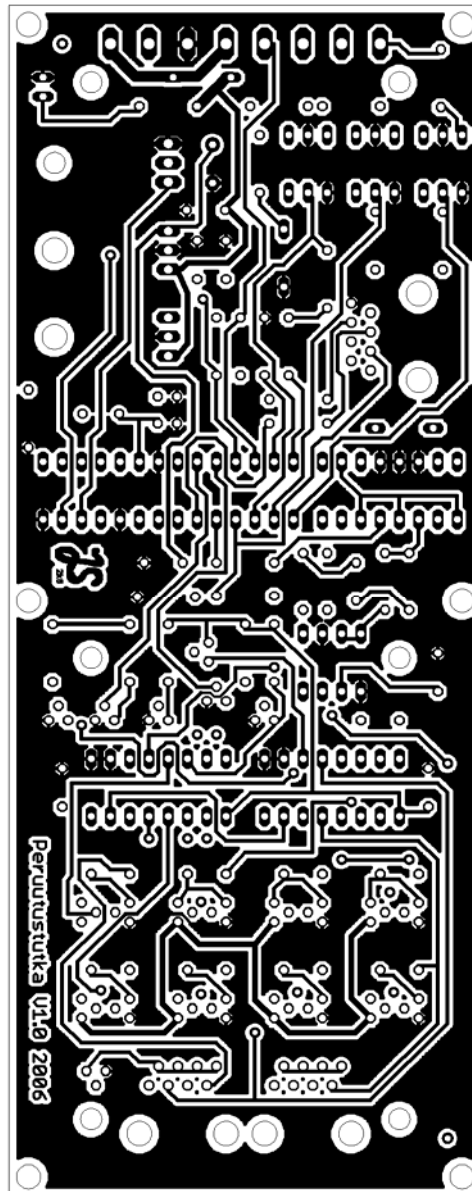
Date: 26.05.2006 22:29:24

Sheet: 1/1

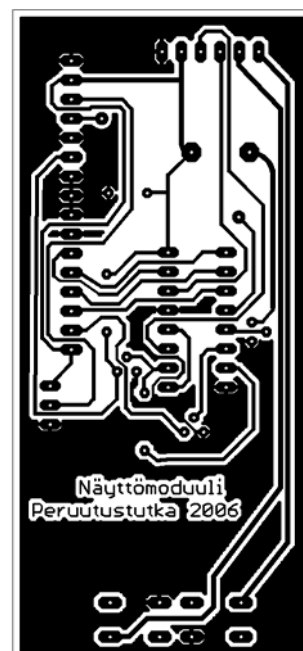
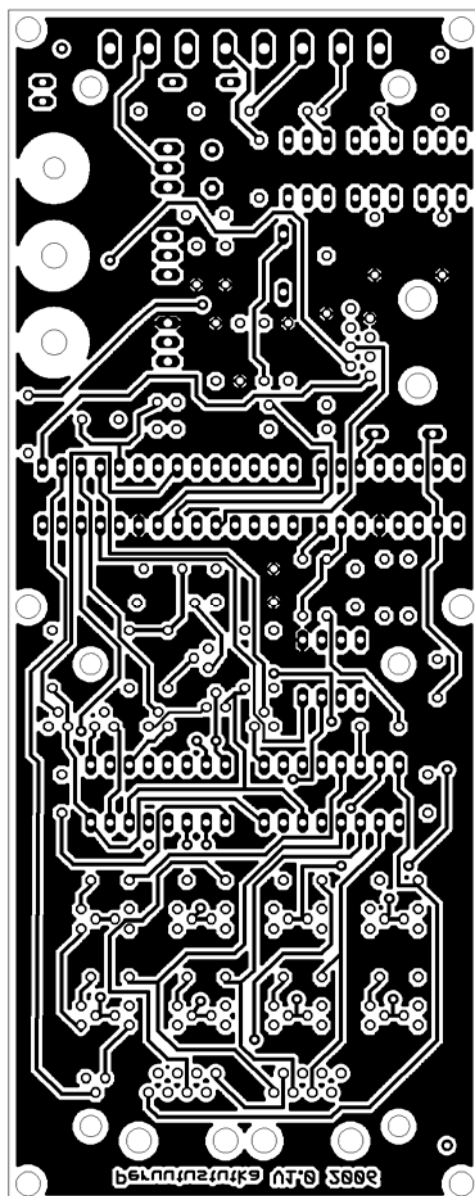
Liite 9 Piirilevyjen osasijoittelukuvat



Liite 10 Piirilevyn syövytyskuva yläpuoli



Liite 11 Piirilevyn syövytyuskuvat alapuoli



Liite 12 Komponenttilistaus

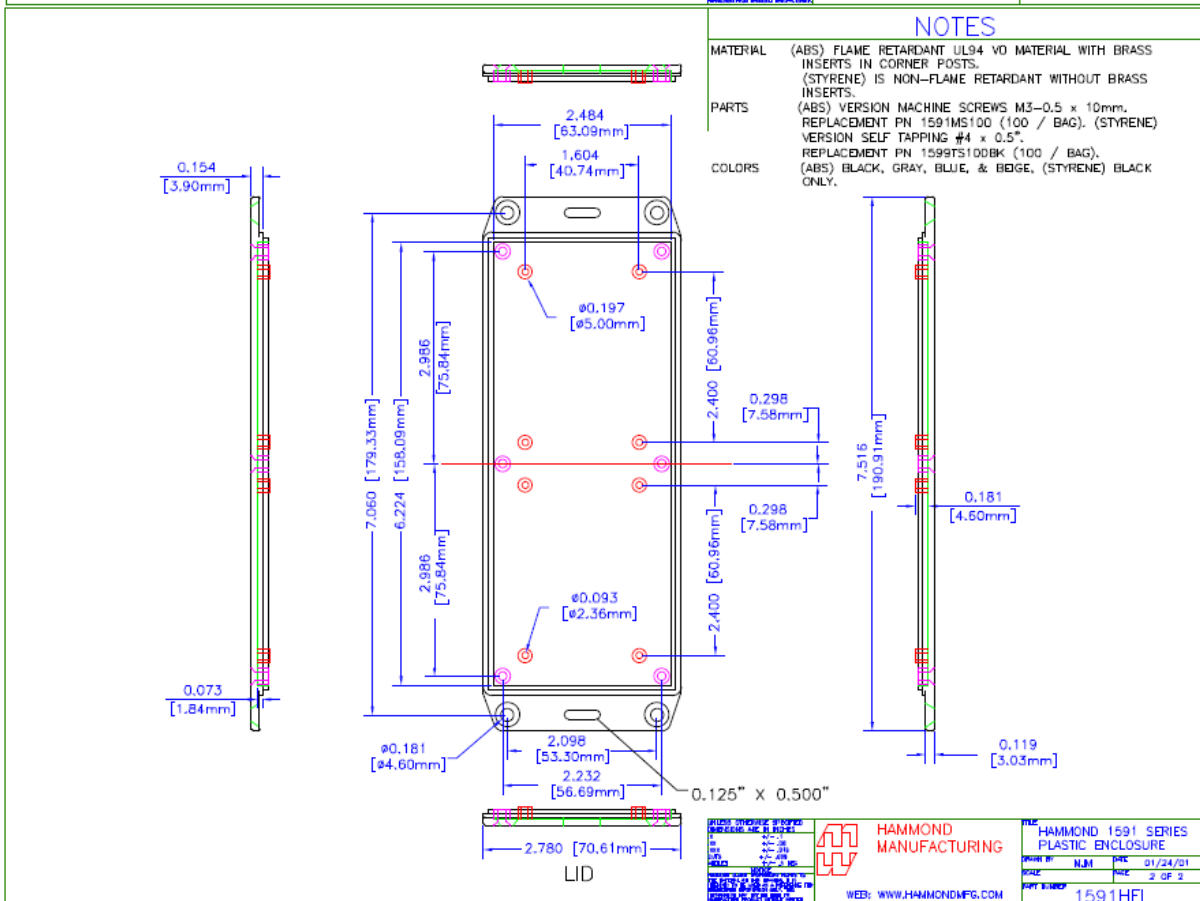
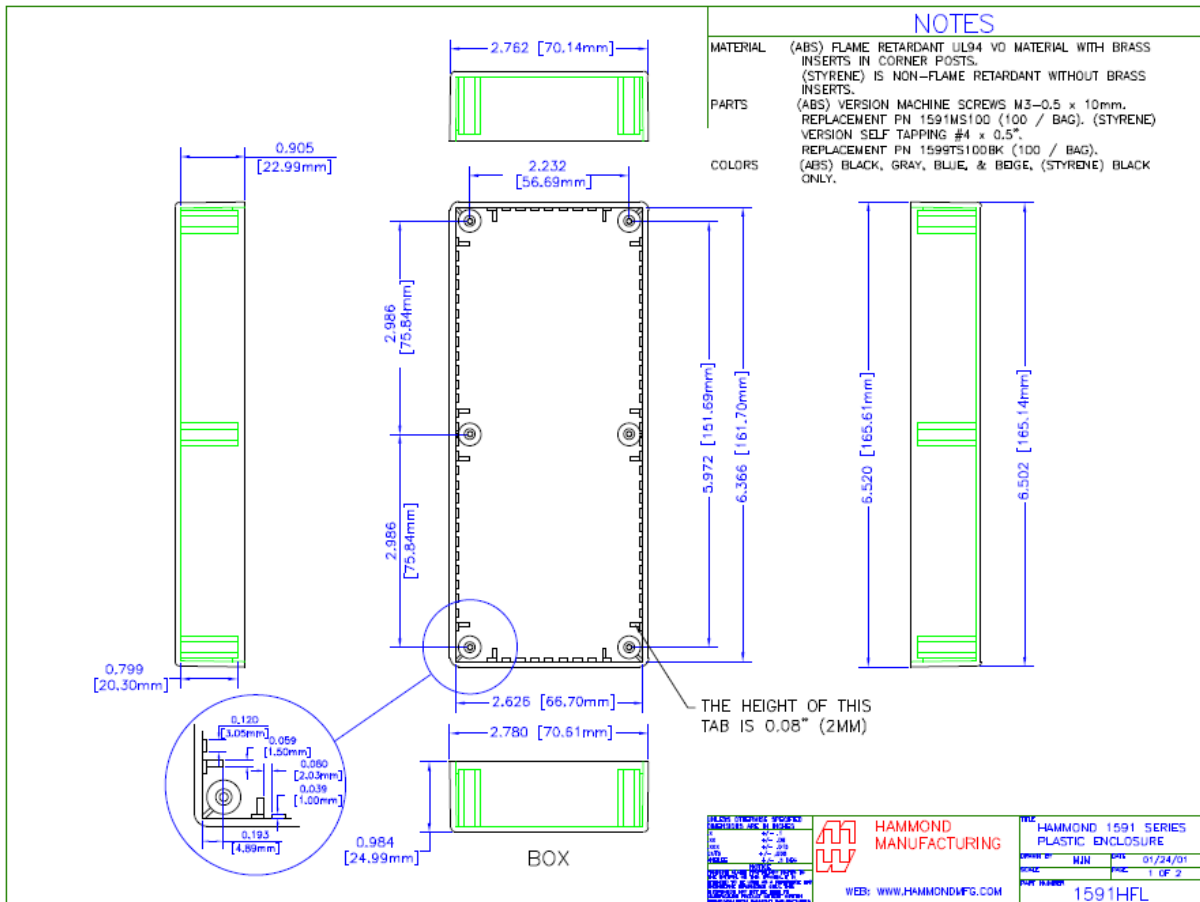
Tutkamoduuli

Qty	Value	Device	Parts
3	RJ45	215877-7	J1, J2, J3
2		C-EU025-025X050	C3, C4
1	10Mhz	CRYTALHC49U-V	Q1
1	green 3mm	LEDSFH482	LED1
1	1k	R-EU_0207/7	R19
1	1n	C-EU050-030X075	C16
1	2,4k	R-EU_0207/7	R34
14	2N7000	2N7000	TR2, TR4, TR6, TR8, TR10, TR12, TR14, TR16, TR17, TR18, TR19, TR20, TR21, TR22
1	5,1	ZENER-DIODEDO34-7	D1
2	220	R_NTC	NTC1, ulkolämpö
19	10k	R-EU_0207/7	R1, R2, R3, R4, R7, R8, R9, R17, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33
2	10n	C-EU050-030X075	C17, C18
8	33n	C-EU050-030X075	C5, C6, C7, C8, C9, C10, C11, C12
6	100n	C-EU050-030X075	C1, C2, C13, C14, CPIC, CPIC1
1	150k	R-EU_0207/7	R18
1	470uF,16V	CPOL-EUE5-6	C15
4	680	R-EU_0207/7	R5, R6, R20, R21
2	4051N	4051N	IC4, IC7
1	7805	78XXL	IC2
1	7810	78XXL	IC6
1	AD5220 100k	AD5220	IC8
2	BAT81	BAT81	D2, D4
9	BSP254A	BSP254A	TR1, TR3, TR5, TR7, TR9, TR11, TR13, TR15, TR23
1	BUZ71	BUZ71L	Q3
1	MSTBA8	MSTBA8	X1
3	OPTO4N25	OPTO4N25	OPT1, OPT2, OPT3
1	PIC16873P	PIC16873P	IC3
1	TS925	TS925ERIL	OPAMP_

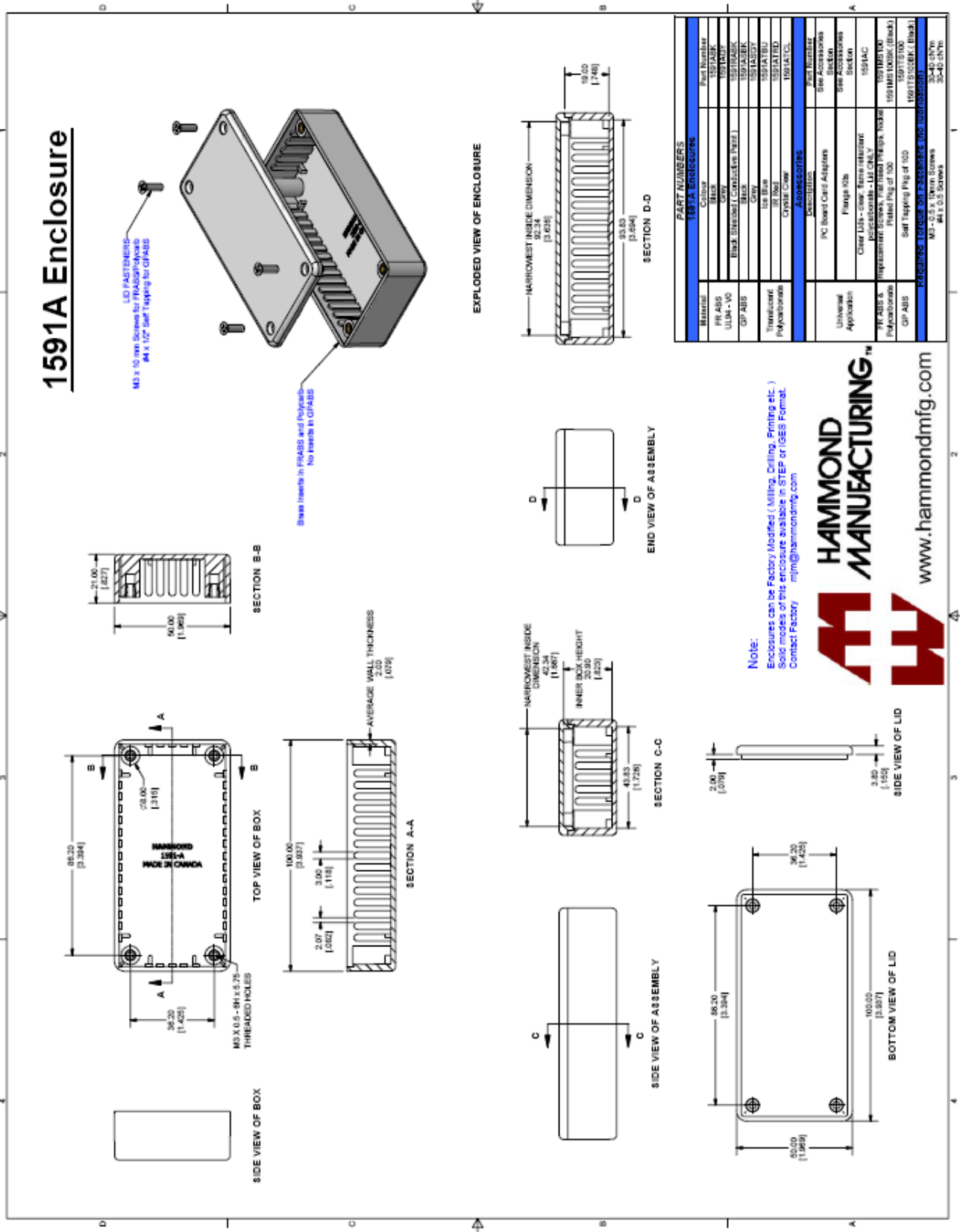
Näyttömoduuli

Qty	Value	Device	Parts
2	10-XX	SWITCH	NAPPI_A, NAPPI_B
1		C-EU050-030X075	C1
1		M06	SL1
1	4,7k	TRIM_EU-RJ9W	R1
3	2N7000	2N7000	EN, SUM, TV
1	10k	R-EU_0204/7	R4
1	47k	R-EU_0204/7	R2
1	74HC595N	74HC595N	IC1
1	F/CM12P	F/CM12P	SUMMERI
2	red 5mm	LED	LED-vasen, LED-oikea

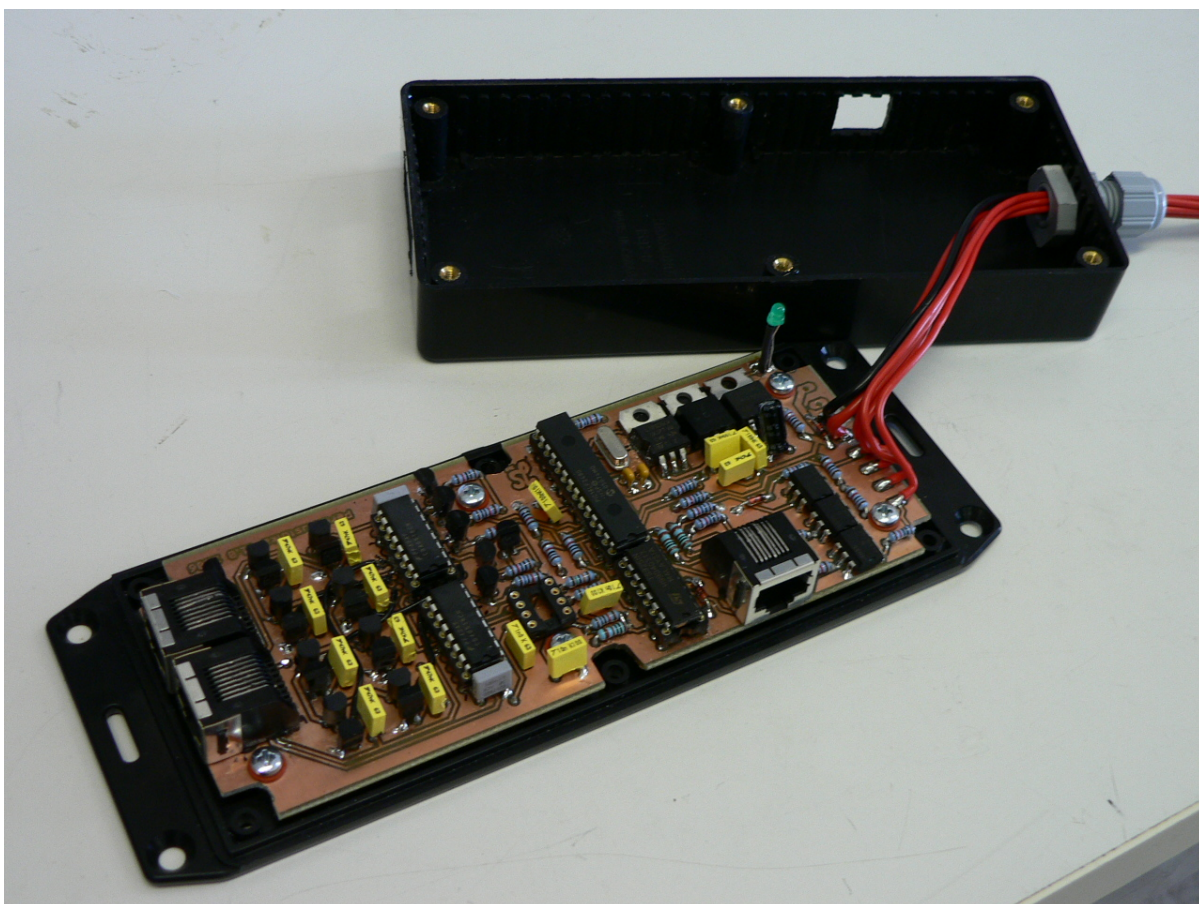
Liite 13 Tutkamoduulin kotelo



Liite 14 Näyttömoduulin kotelo



Liite 15 Tutkamoduuli



Liite 16 Peruutustutkan lähdekoodi

```
#include "C:\Program Files\PICC\peruutustutka\Parking_radar.h"

int16 CONST ADTEMPERATURES[32] =
{159,184,215,229,243,258,273,290,307,324,343,362,382
,403,425,446,470,492,515,538,562,586,609,633,656,679,701,723,745,765,785,822};
int CONST TEMPERATURES[33] = {40,35,30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0,
2,4,6,8,10,12,14,16,18,20,22,24,26,28,30};

int Distance[8];
int Time;

#int_RB
RB_isr()
{ //katsotaan mistä keskeytys tuli
  if(Back_signal == 1)
    Reverse_radar();
  else if(Left_signal == 1)
    Left_radar();
  else if(Right_signal == 1)
    Right_radar();
}

#int_timer0
Timer0()
{
  buzzer^=1;
  sendtolcd(buzzer << 5);
}

//peruutustutka
void Reverse_radar()
{
  while(Back_signal == 1) {
    Measure_distance(0,4); //mitataan matka taaksepäin 4 anturilta
    Set_alarm();
  }
}

//kaistanvaihtotutkat
void Left_radar()
{
  while(Left_signal == 1) {
    Measure_distance(6,1);
    Set_sidealarm(0);
  }
}

void Right_radar()
{
  while(Right_signal == 1) {
    Measure_distance(7,1);
    Set_sidealarm(1);
  }
}

//kaistanvaihtotutkan hälytys
void Set_sidealarm(int1 side) // 0 vasen 1 oikea
{
  side+=6;
  if(Distance[side] > 150) //jos matka on yli 1,5m
    if(side == 6)
      output_high(LED_left); //vilkutetaan lediä
    else
      output_high(LED_right); //vilkutetaan lediä
  else if(Distance[side] > 100) { //jos matka on yli 1,0m
    set_timer0(0x55);
    enable_interrupts(int_timer0);
  }
  else
  {
    buzzer = 1;
    sendtolcd(0b00100000);
  }
}

void clk_pulse() // lähettää kellopulssit näytölle
{
```

```
        Output_high(LCD_clk);
        Output_low(LCD_clk);
    }
    //lähettää yhden kehyksen näyttömoduulille
    void sendtolcd(byte sdata)
    {
        int i,j;
        i = 7;
        while(i != 0) // niin kauan kuin i on erisuuri kuin 0
        {
            // eli 2 kierrosta
            Output_high(LCD_data); //Lähetetään enable
            clk_pulse(); //kello pulssi
            Output_bit(LCD_data, backlight); //lähetetään taustavalo
            clk_pulse();
            Output_bit(LCD_data, buzzer); //lähetetään summeri
            clk_pulse();
            for(j=0;j<4;i--,j++) //Lähetetään 4 databittiä
            {
                Output_bit(LCD_data,bit_test(sdata,i));
                clk_pulse();
            }
            Output_high(LCD_data);
        }
    }

    //lämpötilan laskenta
    void Count_temperature(int channel,int temperature[3] )
    {
        int AD_value; //temperature[0] = lämpötilan kokonaisluku [1] = desimaaliluku
        [2] 0 = pos / 1 = neg
        int i;
        int difference; //taulukon arvojen väli
        set_adc_channel(channel); //mistä kanavasta otetaan AD
        AD_value = read_adc(); //luetaan arvo

        for(i=0;AD_value < ADTEMPERATURES[i];i++); //etsitään taulukosta ensimmäinen
        pienempi arvo
        if( i > 32)
        {
            difference = ADTEMPERATURES[i+1] - ADTEMPERATURES[i]; //taulukon arvojen
            väli
            temperature[1] = AD_value - ADTEMPERATURES[i]; //muutetaan väli lineaarises-
            ti tarkkailtavaksi
            temperature[1] = temperature[1] / difference * 20; //lasketaan desimaalit
            0,0... 2.0
        }
        temperature[0] = TEMPERATURES[i]; //katsotaan TEMPERATURE taulukosta lämpöti-
        la indeksillä i
        if(i < 18) //onko negatiivinen
            temperature[2]++;
        if(temperature[1] > 9) //lisätään lämpötilaan 1, jos dvalue
        10 tai yli
        {
            temperature[0]++;
            temperature[1]-=10;
        }
    }

    //Hälytysrajojen testaus
    void Set_alarm()
    {
        int i,temp;
        for(i=0;i<4;i++) {
            if(Distance[i] < Distance[i+1])
                temp = i;
            else
                temp = i+1;
        }
        if(Distance[temp] > 121) //jos matka on yli 1,2m
            buzzer = 0;
        else if(Distance[temp] < 31) //jos matka on alta 0,3m
            buzzer = 1;
        else if(Distance[temp] < 61) { //jos matka on alta 0,6m
            set_timer0(0xaa);
            enable_interrupts(int_timer0);
        }
        else if(Distance[temp] < 91) { //jos matka on alta 0,9m
```



```
        set_timer0(0x55);
        enable_interrupts(int_timer0);
    }
}

//Peruutustutkan matkan mittaus
void Measure_distance(int channel,int amount)
{
    int tempdist;
    amount+=channel;
    for(;channel < amount;channel++) //channel (global)
    {
        output_a(channel<<2 & 0b00011100); //anturin valinta
        output_low(Inhibit); //anturi valittu
        bit_set(INTCON,INTE); //Sallitaan kaiunvastaanotto
        disable_interrupts(int_timer0);
        Send_frequency(); //lähetetään 40kHz taajuus purske
        bit_clear(INTCON,INTF); //nollataan kaiunvastaanotto lippu
        while(tempdist < 256 && Distance[channel] == 0)
        {
            if(bit_test(INTCON,INTF))
                Distance[channel] = tempdist;
            delay_us(Time); //2 senttimetriin kuluva aika.
            tempdist++;
        }
        tempdist = 0;
    }
}
//lähettää 10 pulssia 40kHz taajuutta

void Send_frequency()
{
    int i;
    for(i=0;i<21;i++)
    {
        output_toggle(Freq);
        delay_us(12);
    }
    output_float(Freq);
}
//automaattinen vahvistuksen säätö

void Set_gain()
{
    int i;
    output_high(Up_down); //aluksi vahvistus suurimmalle
    for(i=0; i < 128;i++)
        output_toggle(Pot_clk);
    output_low(Up_down);
    while(input(Echo) == 1) //vahvistusta vähennetään siihen asti
        output_toggle(Pot_clk); //kun turhaa kaikua ei enää saada
}

void main() {

    setup_adc_ports(RA0_RA1_RA3_ANALOG);
    setup_adc(ADC_OFF);

    setup_counters(RTCC_INTERNAL,WDT_18MS);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    Set_Trис_A(0b00000111);
    Set_Trис_B(0b1111001);
    Set_Trис_C(0b00010000);
    enable_interrupts(INT_RB);
    enable_interrupts(global);
    Set_gain();

        while(true);

}
}
```