

# TIETOKONEEN VIRTALÄHTEEN JÄNNITEMITTARI

LAHDEN AMMATTIKORKEAKOULU  
Tekniikan ala  
Tietotekniikan koulutusohjelma  
Tietokone-elektronikka  
Opinnäytetyö  
Kevät 2010  
Antti Lakanen

Lahden ammattikorkeakoulu  
Tietokonetekniikan koulutusohjelma

LAKANEN, ANTTI: Tietokoneen virtalähteen jännitemittari

Tietokone-elektroniikan opinnäytetyö, 34 sivua, 6 liitesivua

Kevät 2010

## TIIVISTELMÄ

---

Tässä opinnäytetyössä tutustutaan tietokoneen virtalähteisiin sekä suunnitellaan toimiva jännitemittari tietokoneen virtalähteelle. Laitteen tarkoituksena oli huoltotöiden nopeuttaminen tietokoneita korjaavissa yrityksissä sekä helpottaa työntekijöiden työtaakkaa.

Teoriaosuudessa tutustutaan hakkuritehonlähteiden teoriaan, toimintaperiaatteen, ohjaukseen ja hakkuritehonlähteiden rooliin tietokoneissa. Samalla kerrotaan kaikki työlle asetetut vaatimukset ja lähtötiedot, joiden pohjalta laitteen suunnittelu aloitettiin. Suunnitteluun liittyen esitellään työssä käytettävä mikrokontrolleri. Mikrokontrolleriluvussa tutustutaan mikrokontrolleriin ja opinnäytetyössä käytettyihin ominaisuuksiin.

Seuraavaksi keskitytään laitteen piirikaavion ja -levyn suunnitteluun. Suunnittelu-luvussa esitellään käytetty ohjelma sekä selostetaan, kuinka ohjelmalla suunnitellaan valmis piirilevy. Piirilevyn suunnittelua seurasi laitteen kokoaminen. Samassa yhteydessä tutustutaan myös valmistelevista työvaiheista ennen juottamisen aloittamista.

Lopuksi tutustutaan laitteen ohjelman suunnitteluun ja toteutukseen. Ohjelman kirjoittaminen ja eri komentojen käyttö sekä rekistereiden alustukset selitetään teoriassa. Ohjelmoinnin seuraavassa vaiheessa esitellään ohjelma yksityiskohtaisesti ja selostetaan jokainen koodirivi, jotta lukija ymmärtää asian hyvin.

Yhteenvedossa arvioidaan työn onnistumista. Kommentoidaan projektin hyviä ja huonoja puolia sekä mitä voisi tehdä paremmin seuraavalla kerralla.

Avainsanat: virtalähde, suunnittelu, piirikaavio, piirilevy, ohjelmointi

Lahti University of Applied Sciences  
Degree Programme in Information Technology

LAKANEN, ANTTI: Voltage meter for the power supply unit of a computer

Bachelor's Thesis in Computer Electronics, 34 pages, 6 appendices

Spring 2010

## ABSTRACT

---

The object of this Bachelor's thesis is to get acquainted with computer power supplies and to design a working voltage meter for it. The main function of the device to be designed is to speed up the maintenance procedure and decrease the workload for the workers in companies that specialize in computer maintenance.

The thesis describes the theory, operation principles, controlling and the role of the switch-mode power supplies in computers. It also presents the standards and startup information of the project, based on which the planning was started. It is justified why certain solutions were made. Along with the planning process, the micro controller unit and its properties are described.

The next chapter focuses on planning of the circuit diagram and circuit board. The used program is presented and the whole design phase is explained from the beginning. Following the planning stage is the assembly of the device. The chapter also tells what preparations need to be done before soldering.

The last chapter is about the planning and the execution of the program. Writing the program and the usage of the commands and also the formatting of the registers is explained in theory. The program is explained in detail and every single line in the code is explained in great detail.

In the summary section the project is evaluated as a whole. The successes and failures of the project are commented and there is speculation about what could have been done better.

Key words: power supply unit, designing, circuit graph, circuit board, programming

## SISÄLTÖ

1	JOHDANTO	1
2	HAKKURITEHOLÄHDE	2
2.1	Hakkuriteholähde teoriassa	2
2.1.1	Hakkuriteholähteen toimintaperiaate	4
2.1.2	Pulssinleveysmodulaatio hakkuriteholähteessä	5
2.2	Hakkuriteholähde tietokoneissa	6
3	LAITTEEN SUUNNITTELU JA OHJELMAN TEKEMINEN	8
3.1	Suunnittelu	8
3.1.1	Mikrokontrolleri AT90USB1287	14
3.1.2	Piirikaavio ja piirilevy	16
3.1.3	Piirilevyn kokoaminen	21
3.2	Ohjelmointi	23
3.2.1	Ohjelmoinnin suunnittelu	24
3.2.2	Ohjelma	28
4	YHTEENVETO	30
4.1	Onnistumiset, epäonnistumiset ja korjaukset	30
4.2	Opinnäytetyö kokonaisuutena	32
	LÄHTEET	33
	LIITTEET	35

# 1 JOHDANTO

Tietokoneet ovat suuressa roolissa nyky-yhteiskunnassa. Tietokoneiden määrä lisääntyy jatkuvasti ja näin ollen myös tietokoneisiin liittyvät viat ja ongelmat. Näiden syiden vuoksi tietokoneita huoltavien yritysten työtaakka lisääntyy. Suomessa, etenkin kesäisin, työtaakkaa eniten lisäävät ukonilmoista johtuvat virtalähteiden hajoamiset. Hyvin yksinkertainen vika, mutta suuressa määrin kuormittaa ja hidastaa huoltotoimintaa huomattavasti. Opinnäytetyössäni perehdytään huoltotoimintaa ongelmaa helpottavaan ratkaisuun.

Tietokoneen virtalähteen jännitemittarilla yritetään nopeuttaa huoltotoimia ja näin välttää huoltojen ruuhkautumista ja huoltoaikojen pidentymistä. Jännitemittari kytketään virtalähteeseen, näin vältetään tietokoneen komponenttien irrottamista, mikä nopeuttaa huoltoa, ja laite antaa yksiselitteisen vastauksen virtalähteen tilasta.

Tämän opinnäytetyön tarkoitus on suunnitella ja rakentaa toimiva tietokoneen virtalähteen jännitemittari, jota olisi mahdollista käyttää apuna tietokoneiden huoltamisessa. Samalla tutustua eri työvaiheisiin, joita laitteiden suunnittelussa tulee huomioida.

Opinnäytetyössä tutustutaan aluksi hakkuriteholähteiden teoriaan ja käyttökohteisiin. Tämän jälkeen tutustutaan kaikkiin työvaiheisiin: laitteen suunnitteluun, piirikaavion ja -levyn suunnitteluun, piirilevyn valmistukseen ja ohjelman kirjoittamiseen.

## 2 HAKKURITEHOLÄHDE

Tässä pääluvussa selvitetään hakkuritehonlähteen teoria, ominaisuudet ja käyttökohteet. Luvussa myös verrataan hakkuriteholähdettä vanhemmalla tekniikalla toteutettuihin ratkaisuihin sekä kerrotaan, kuinka hakkuriteholähde muuttaa seinästä tulevan verkkovirran tasasähkökomponenteille ja laitteille sopiviksi. Tämän lisäksi selitetään hakkuritehonlähteiden toiminta tietokoneissa ja perehdytään tehonlähteen jännitteenhallintatekniikkaan.

### 2.1 Hakkuriteholähde teoriassa

Hakkuritehonlähteet ovat sähköisiä tehonlähteitä, joita käytetään muuntamaan esimerkiksi verkkovirta sopivaksi jollekin kodinkoneelle. Hakkuritehonlähteellä voidaan helposti muuttaa suurempi tulojännite pienemmäksi tai päinvastoin, näin ollen myös tehohäviö jää hyvin pieneksi hyötysuhteen ollessa n. 60–70 %, mutta hyvän suunnittelun avulla voidaan saavuttaa jopa 95 %:n hyötysuhde. (Switched-mode power supply 2009.)

Lineaaritehonlähde pitää tietyn lähtöjännitteen muuttamalla ylimääräisen sähkötehon lämmöksi. Tämän takia lineaaritehonlähteellä voidaan saavuttaa ainoastaan noin 50 %:n hyötysuhde. Verrattuna lähtöjännitteen hallintaa hakkuritehonlähteessä, jossa vaihdetaan kondensaattoreiden ja kelojen tilaa johtavan ja johtamattoman tilan välillä, lähtöjännitteen hallitsemiseksi. Ideaalitapauksessa transistoreja käytetään niiden aktiivialueen ulkopuolella, jolloin niillä oikosulussa ei ole resistanssia eivätkä ne avoimna kuljeta virtaa, mikä teoriassa mahdollistaisi 100 % hyötysuhteen eli kaikki tuloteho saadaan suoraan ohjattua kuormalle. Lähtöjännite riippuu sisään tulevasta tehosta, valituista komponenteista ja niiden välisestä topologiasta sekä hakkurin ohjauksesta. Yleensä hakkuritehonlähteet toimivat suurella taajuudella. Tästä aiheutuva rippelijännite voidaan suodattaa pois sopivalla kela-kondensaattorikytkennällä. Taulukossa 1 on käyty läpi muita etuja ja haittoja hakkuritehonlähteen ja lineaaritehonlähteen välillä. (Switched-mode power supply 2009.)

Suurin etu hakkuritehonlähteen käytössä on sen alhainen hukkateho. Suurtaajuudella ohjattu transistori kuluttaa vain vähän virtaa sen ollessa aktiivialueensa ulkopuolella. Muita etuja ovat kevyt paino ja pienempi koko. Edellä mainitut ominaisuudet on saavutettu siirtymällä pientaajuusmuuntajista suurtaajuusmuuntajiin, jolloin tarvittavan kelan koko ja paino on pienempi. Pienempi lämmöntuotto voidaan myös laskea eduksi. (Switched-mode power supply 2009.)

Hakkuritehonlähteessä on myös haittoja. Näihin voidaan laskea sen monimutkaisuus verrattuna vanhoihin pientaajuusmuuntajiin sekä suurten taajuuksien sivutuotteena tulevat sähkömagneettiset häiriöt, joiden suodattamiseen tarvitaan ali-päästösuodatin. Muita haittoja ovat suuri vaihteluväli jännitteessä sekä rippeli-jännitekytkimen ohjaustaajuus ja sen aiheuttamat harmoniset värähtelyt. (Switched-mode power supply 2009.)

TAULUKKO 1. Lineaarisen virtalähteen edut ja haitat hakkurivirtalähteeseen (Switched-mode power supply 2009.)

	Lineaarinen virtalähde	Hakkurivirtalähde	Merkille pantavaa
Koko ja paino	Jos käytössä on pienmuuntaja, on laite ulkoisilta mitoiltaan suurikokoinen.	Laite on pienikokoinen, jos kytkentätaajuus on suuri.	-
Hyötysuhde, lämpö -ja tehohäviö	Jos reguloitu, suurin osa ylimääräisestä tehosta haihtuu lämmöksi näin ollen hyötysuhde tyypillisesti 30–40%.	Käyttää tehoa vain sen verran kuin kuorma tarvitsee.	Ainoa lämpö syntyy, kun komponentit toimivat ideaalisen alueensa ulkopuolella.
Radiotaajuus-häiriöt	Syntyy heikkoa korkeataajuuksista häiriötä, kun kondensaattorit ovat suuren virtakuorman alla.	Nopeasti muuttuva virran tila aiheuttaa elektromagneettista ja radiotaajuushäiriötä. Tämän takia EMI suodattimia ja RF suojausta tarvitaan hillitsemään häiriötä.	Pitkät vedot heikentävät suodatusta tuloissa ja lähdoissä

### 2.1.1 Hakkuriteholähteen toimintaperiaate

Hakkuriteholähteellä muutetaan pistorasiasta tuleva verkkojännite matalammaksi tasajännitteeksi. Jännitettä voidaan sen jälkeen käyttää tasajännitettä käyttävien laitteiden sisäisten komponenttien käyttöjännitteenä. Ensimmäinen vaihe hakkuriteholähteen muunnosoperaatiossa on tasasuuntaus. Tällöin muutetaan pistorasiasta tuleva vaihtojännite tasajännitteeksi. Tämän jälkeen tasajännite menee suureen suodatuskondensaattoriin.

Sähköverkosta otettu virta ilmenee lyhyinä pulsseina vaihtojännitepiikkien ympärillä. Pulssit ovat korkeataajuusenergiaa. Virta on hyvin epälineaarista pulsseissa. Hakkurivirtalähteessä käytetään sisääntulon jännitealueen valintakytkintä. Sisään tulevan jännitteen ollessa pientä eli 120 VAC käytetään tasasuuntaajaa jännitekerrotojana, joka kaksinkertaistaa jännitteen. Muussa tapauksessa sisään tuleva jännite on suurta eli 230/240VAC, ja silloin käytetään tasasuuntaajaa suoraan. Jännitealueen valinta kytkimen puuttuessa käytetään kokoaaltotasasuuntausta. (Switched-mode power supply 2009.)

Seuraavassa vaiheessa tasasuuntauksesta saatu tasajännite käännetään takaisin vaihtojännitteeksi käyttämällä teho-oskillaattoria, jossa on pieni lähtöjännitemuuntaja. Tässä vaiheessa tasajännite muutetaan suurtaajuusvaihtojännitteeksi, jotta siitä voidaan seuraavassa vaiheessa muuntajalla saada halutut jännitteet ulos. Taajuus valitaan yleensä korkeammaksi kuin 20 kHz, jotta se ei ole kuultavissa ihmiskorvalla. Taajuus on kymmenistä satoihin kilohertzeihin. Tilanvaihto on toteutettu monivaiheisella mosfetilla. Koska ainoastaan viimeisessä vaiheessa on suuri tehtäväjakso, voidaan aikaisemmat vaiheet toteuttaa bipolaarisilla transistoreilla tehokkuuden kuitenkin kärsimättä. Toiseksi viimeisessä vaiheessa käytetään vastavuoroista mallia, jossa toinen transistori lataa viimeisen mosfetin ja toinen transistori purkaa varauksen. Tehokkuus laskisi jos toiseksi viimeinen vaihe tehtäisiin vastuskytkennän avulla. (Switched-mode power supply 2009.)



Seuraavaksi vaihtojännitettä johdetaan lähtömuuntajan ensiökelaan, joka sitten pienentää tai suurentaa jännitettä toisiokelalle, jotta saadaan halutut lähtöjännitteet kuormalle. Jos kuorma vaatii tasajännitettä, tarvitsee lähtöjännite tasasuunnata muuntajan jälkeen. Yli 10 voltin lähtöjännitteiden tasasuuntaukseen voidaan käyttää piidiodeja mutta alle kymmenen voltin lähtöjännitteiden luomiseen käytetään joko schottky-diodeja tai mosfetteja. Komponenteilla on pienempi elpymisaika kuin piidiodeilla, mikä mahdollistaa pienemmät häviöt suurilla taajuuksilla. Tasasuuntauksen jälkeen lähtöjännite suodatetaan kela-kondensaattorikytkennän avulla. (Switched-mode power supply 2009.)

Viimeinen osa hakkuritehonlähteessä on takaisinkytkentäpiiri. Piiri vertaa lähtöjännitettä joko ennalta annettuun vertailuarvoon tai referenssijännitteeseen. Jos vertailussa lähtöjännite poikkeaa halutusta jännitteestä, kompensoi piiri sitä muuttamalla mosfettien tilanvaihtojen ajoitusta. Riippuen hakkuritehonlähteen kuormasta sovelluksessa saattaa olla käytössä optinen erotus, jolla saadaan estettyä mahdollisten jännite- ja virtapiikkien pääsy kuormaan. (Switched-mode power supply 2009.)

### 2.1.2 Pulssinleveysmodulaatio hakkuriteholähteessä

Pulssinleveysmodulaatio eli PWM on tehokas keino hallita digitaalisesti analogisia lähtöjä. Pulssinleveysmodulaatiolla on mahdollista luoda mikä tahansa lähtöjännite maan ja tulojännitteen väliltä. Tämä vaatii ainoastaan hieman resursseja pulssinleveysmodulaatiota ohjaavalta mikrokontrollerilta. Tehtäväjaksolla (duty cycle) tarkoitetaan aikaa, jonka PWM on aktiivinen eli tulojännite ohjataan suoraan kuormalle. Esimerkiksi yhdeksän voltin tulojännite muuttuu, 30 % tehtäväjaksolla, 2,7 voltin jännitteeksi modulaatiotaajuuden ollessa riittävän suuri, ettei se vaikuta kuormaan. Yleensä modulaatiotaajuus on 1–200 kHz riippuen kuormasta. Pulssinleveysmodulaation etuna on se, että ohjaussignaali pysyy digitaalisena mikrokontrollerilta ohjattavaan komponenttiin tai systeemiin. Tämä vähentää häiriöiden vaikutusta ohjaussignaaliin, ne vaikuttavat ainoastaan, jos häiriöt ovat niin

vahvoja, että ne pystyvät muuttamaan loogisen nollan ykköseksi tai päinvastoin. PWM on soveltuva käytettäväksi melkein missä tahansa elektronisessa sovelluksessa. Tyypillisimpiä käyttökohteita ovat lamppujen himmentimet, hakkuriteholähteet ja portaattomat tuulettimien nopeuden säätimet. (Barr 2001, 101–103.)

## 2.2 Hakkuriteholähde tietokoneissa

Virtalähteellä tietokoneissa tarkoitetaan laitetta, jolla jaetaan virtaa eteenpäin muille tietokoneen komponenteille. Virtalähteellä muutetaan pistorasiasta tuleva vaihtosähkö pienjännitteiseksi tasasähköksi, jolla saadaan virtaa tietokoneen sisällä oleville komponenteille. Yleisin tietokoneiden virtalähdestandardi on ATX. Standardissa määritellään virtalähteiden ulkomitat sekä sähköiset ominaisuudet, jotta ne ovat yhteensopivia muiden ATX-standardia noudattavien komponenttien ja laitteiden kanssa. ATX-standardin uusin versio on 2.31, joka julkaistiin kesällä 2008. (Power supply unit (computer) 2009.)

1980-luvulla virtalähteet olivat vielä hyvin painavia ja isokokoisia, niissä saattoi olla virvoitusjuomatölkin kokoisia kondensaattoreita ja painavia muuntajia muuttamassa pistorasiasta tulevan 230 voltin vaihtosähkön viiden ja kahdentoista voltin jännitteeksi. Nykyään virtalähteet ovat edellisessä kappaleessa esitettyjä hakkuriteholähteitä. Nykyaikaisen tietokoneen virtalähteen antoteho on 110 watista aina 2 kilowattiin asti. (Power supply unit (computer) 2009.)

ATX-standardivirtalähteestä löytyy seitsemän erilaista liitäntää tietokoneen sisäisille komponenteille ja laitteille. Liittimet on myös muotoiltu sillä tavalla, että ne on vaikeaa kytkeä väärin paikkoihin tai väärin päin. Liittimien johdot ovat myös värikoodatut. Tärkeimmät liittimet ovat seuraavat: Päävirta (P1), prosessorin lisävirta (P4), molex ja PCI-express lisävirtaliitin. P1-liitintä on olemassa kahta eri versiota, 20 ja 24 pinnin. Ensimmäisenä mainittu löytyy hieman vanhemmista tietokoneista ja jälkimmäinen uudemmista. Päävirtaliittimen kautta saadaan virta emolevyn komponenteille ja eri liitäntäväylille. Päävirtaliitimestä saadaan 12 V, 5 V, 3.3 V:n jännitteet sekä harvemmin käytetyt negatiiviset jännitteet -12 V ja -5 V. Kuviossa 1 nähdään päävirtaliittimen johdotus, värikoodit ja johtimien määrät.

P4-liitännästä on olemassa myös kaksi erilaista versiota neljän ja kahdeksan pinniset. Uusimmat tietokoneet käyttävät jälkimmäistä liitännää, koska se mahdollistaa paremman virransyötön.

Molex on tietokoneiden yleisin liitäntä. Liitännällä usein saadaan virtaa kaikkiin oheislaitteisiin tietokoneen sisällä, kuten dvd-asemat ja kiintolevyt. Molex-liittimen monipuolisen käytön mahdollistaa viiden ja kahdentoista voltin jännitelähdöt. SerialATA-virtaliitin on hiljaksen korvaamassa molex virtaliittintä tietokoneissa. SerialATA-virtaliitin tarjoaa molexin tarjoamien jännitteiden lisäksi myös 3.3 V:n jännitteen. (Power supply unit (computer) 2009.)

Color	Signal	Pin	Pin	Signal	Color
Orange	+3.3 V	1	13	+3.3 V	Orange
				+3.3 V sense	Brown
Orange	+3.3 V	2	14	-12 V	Blue
Black	Ground	3	15	Ground	Black
Red	+5 V	4	16	Power on	Green
Black	Ground	5	17	Ground	Black
Red	+5 V	6	18	Ground	Black
Black	Ground	7	19	Ground	Black
Grey	Power good	8	20	No connection	
Purple	+5 V standby	9	21	+5 V	Red
Yellow	+12 V	10	22	+5 V	Red
Yellow	+12 V	11	23	+5 V	Red
Orange	+3.3 V	12	24	Ground	Black

KUVIO 1. Päävirtaliittimen johdotus. (Power supply unit (computer) 2009.)

### 3 LAITTEEN SUUNNITTELU JA OHJELMAN TEKEMINEN

Tässä luvussa käsitellään kaikki ne asiat, jotka piti ottaa huomioon laitetta suunniteltaessa. Lisäksi selitetään suunnitteluprosessi alusta loppuun ja käydään läpi laitteen ohjelmointi sekä itse ohjelman kirjoittaminen. Luvussa kerrotaan myös suunnittelussa ja ohjelmoinnissa käytetyistä ohjelmista.

#### 3.1 Suunnittelu

Aloitin suunnittelun tutkimalla ja keräämällä tietoa tietokoneenvirtalähteistä ja hakkuritehonlähteistä, joihin ne perustuvat. Tutustuin tämän päivän tietokoneiden tekniikkaan ja niiden jännitelähteille asettamiin vaatimuksiin, kuten edellisestä pääluvusta selvisi. Näiden vaatimuksien pohjalta päätin, että mittaan kolmea eri jännitettä, 12 V, 5 V ja 3,3 V, neljästä eri liitimestä. Koska nykyään tietotekniikassa käytetään harvemmin - 5 V ja -12 V:n tulojännitteitä, päätin jättää ne huomioimatta tässä laitteessa. Mitattaviksi liittimiksi valitsin edellisessä pääluvussa kerrotut tärkeimmät liittimet.

Mitattavat kohteet selville saatuani aloin miettiä, mitä muuta laitteessa pitäisi olla. Aiemmissa projekteissa laitteen ohjelmointi on ollut erittäin hankalaa, joten päätin heti aluksi kiinnittää huomioni siihen. Aloin tutkia eri liitännämahdollisuuksia, joita voisin hyödyntää ohjelman lataamiseen mikrokontrollerille. Lopulta päädyin siihen lopputulokseen, että kannattavin vaihtoehto tähän olisi käyttää USB-liitännää, koska se on yleistyvä liitäntä erilaisissa sovelluksissa ja sen avulla on mahdollista ohjelmoida laite kotona. (Atmel 2009.)

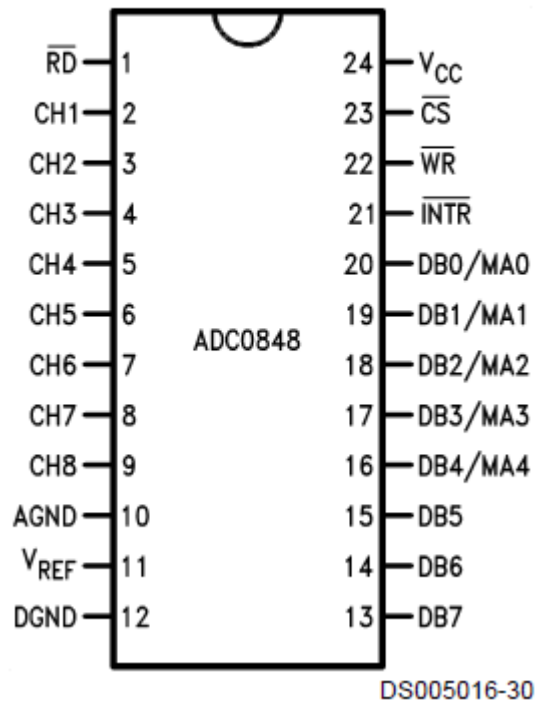
Tutkin eri valmistajien mikrokontrollereita, jotka mahdollistaisivat liitännän käytön, ja päädyin harkinnan jälkeen Atmel AT90USB1287 -mikrokontrolleriin. Valintaan vaikuttivat edellisissä ohjelmissa ja harjoitustöissä tutustuminen Atmelin muihin mikrokontrollereihin ja kyseisen piirin tehtaalla tehty oletuskoodaus, joka mahdollistaa ohjelmiston lataamisen USB-liitännän kautta hyödyntämällä Atmel flip -nimistä ohjelmaa. Yksi syy oli myös, että kurssikaverini oli saanut toimivan koodin latausliitännän tehtyä kyseisellä piirillä ja ohjelmalla edellisellä työkurssillamme, joten olin luottavainen sen toimivuuteen. Piiriltä löytyi myös riittävästi

ominaisuuksia työn muita käyttötarkoituksia varten, mutta USB -liitännän mahdollistaminen oli ehdottomasti tärkein ominaisuus. Kerron mikrokontrollerin ominaisuuksia omassa luvussaan vielä tarkemmin. (Atmel. 2009.)

Mikrokontrollerin valinnan jälkeen pohdin, mitä kautta mittaustulokset voitaisiin lukea. Päädyin siihen ratkaisuun, että laitteeseen tuli pieni LCD-näyttö, josta tulokset olisivat luettavissa. Tarkasteltuani eri vaihtoehtoja päätin valita työhöni 16 x 1 matriisinäytön. 16 x 1 tarkoittaa siis 16 merkkiä yhdellä rivillä. Valintaan vaihtui näytöstä itsestään löytyvä merkkirekisteri, josta löytyi kaikki tarvittavat merkit. Tämä mahdollistaisi helpomman ohjelmoinnin verrattuna aiemmin ollut vaihtoehto koodata jokainen kirjain pisteittäin. Näytön yhteyteen suunniteltiin myös potentiometri säätämään näytön kontrastia. Potentiometrillä voidaan säätää vastus sopivaksi, jotta teksti saadaan näkyväksi näytöllä. Mikrokontrolleriin kytkettiin näytöstä neljä dataväylää sekä R/W (read and write), RS (register select) ja E (enable). (Powertrip 2010.)

Pääkomponenttien valinnan jälkeen mietin itse mittauksen suorittamista. Mittauksen voi suorittaa yksinkertaisella analogia-digitaalimuunnoksella. Tätä varten katsoin erillisiä A/D-muuntimia, jotka pystyisivät muuntamaan analogiajännitteen digitaalimuotoon. Perehdyttyäni asiaan uskoin löytäneeni sopivan komponentin nimeltään ADC0848. Piiristä oli kahdeksan analogista sisääntuloa ja sisäinen kello muunnokselle. Sisääntulojen ns. kuuntelua hallittaisiin kuviossa 2. näkyvillä MA0 – MA5 -pinneillä, CS-, RD- ja WR-pinneillä. (National Semiconductor 2008.)

## Dual-In-Line Package



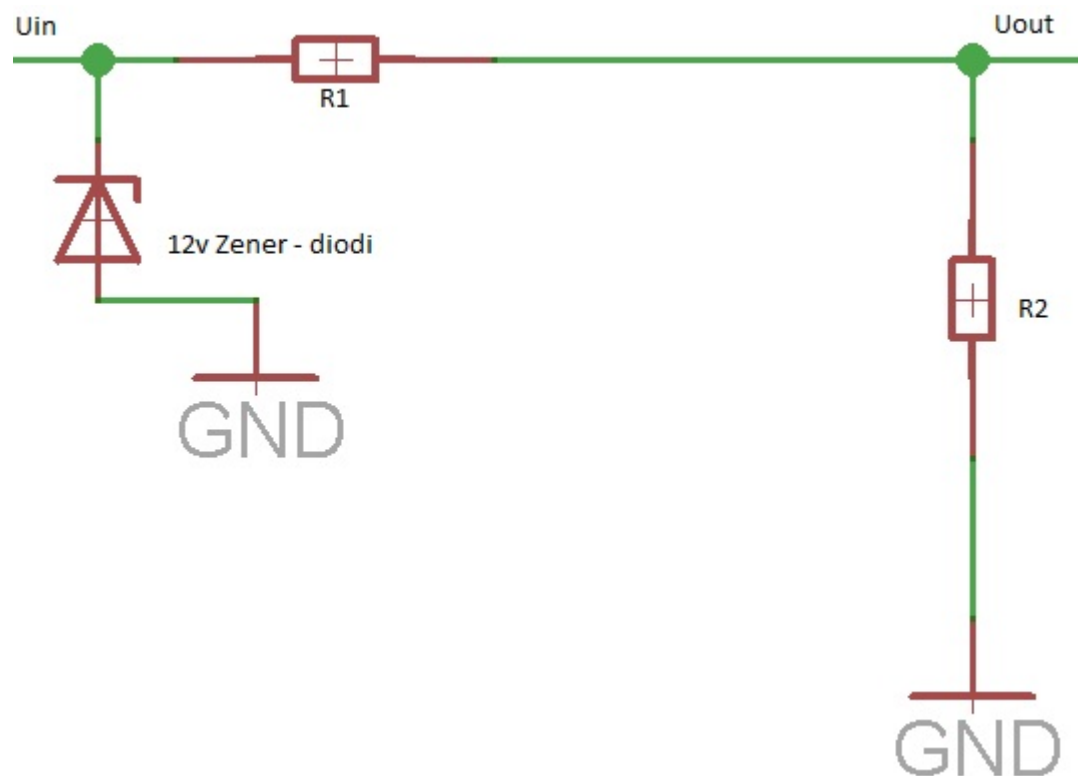
KUVIO 2. ADC0848 analogia-digitaalimuuntimen pakkauskuvat ja pinnien järjestys (National Semiconductor 2008.)

Ongelmaksi muodostuivat 12 V:n jännitteet, jotka ylittäisivät analogiselle muunnettavalle jännitteelle annetun arvon ja näin ollen eivät olisi mahdollisia. Aloin etsiä ongelmaan ratkaisua Internetistä ja konsultoin myös opettajia. Ongelmaan löytyi ratkaisu. Jännitteen alentaminen viiteen volttiin tehtiin jännitejaon avulla. Noudattamalla tätä suunnitelmaa jännitteen alentamiseen, jännitejaolla ja tarkkuusvastuksien avulla, pystyttäisiin säilyttämään mittaustuloksissa tarkkuus. Resistanssit mitoitettiin seuraavan kaavan mukaan:

$$U_{out} = \frac{R_2}{R_1 + R_2} * U_{in}$$

Kaavassa  $U_{in}$  kuvastaa tulojännitettä eli 12:sta volttia,  $R_1$  ensimmäistä vastusta arvoltaan 1.4 k $\Omega$ .  $R_2$  toista vastusta arvoltaan 1 k $\Omega$ .  $U_{out}$  tarkoittaa lähtöjännitettä eli tässä tapauksessa viittä volttia. Näillä arvoilla saatiin 12 v laskettua A/D-

muuntimelle sopivaan muotoon. Myöhemmin ohjelmoinnin yhteydessä kerrotaan A/D-muunnoksen tulos sopivalla suhdeluvulla, joka tässä tapauksessa on 2,4. Jännitejaon yhteyteen piirille suunniteltiin myös zener-diodi-kytkentä, jolla leikataan kaikki yli 12 V:n jännitepiikit pois sisään tulevasta jännitteestä. Tämä tehtiin kuvion 3 mukaisella kytkennällä, jossa sisään tuleva jännite kytketään zener diodin läpi maahan. (Kontra 2009.)



KUVIO 3. Jännitejako sekä zener – diodi kytkentä (Kontra 2009.)

Alle viiden voltin jännitetulot voidaan kytkeä analogia-digitaalimuuntimen tuloihin ilman jännitteiden laskua. Kaiken kaikkiaan mikrokontrolleriin kytkettiin neljä 12 voltin tuloa, kaksi 5 voltin tuloa ja yksi 3,3 voltin tulo.

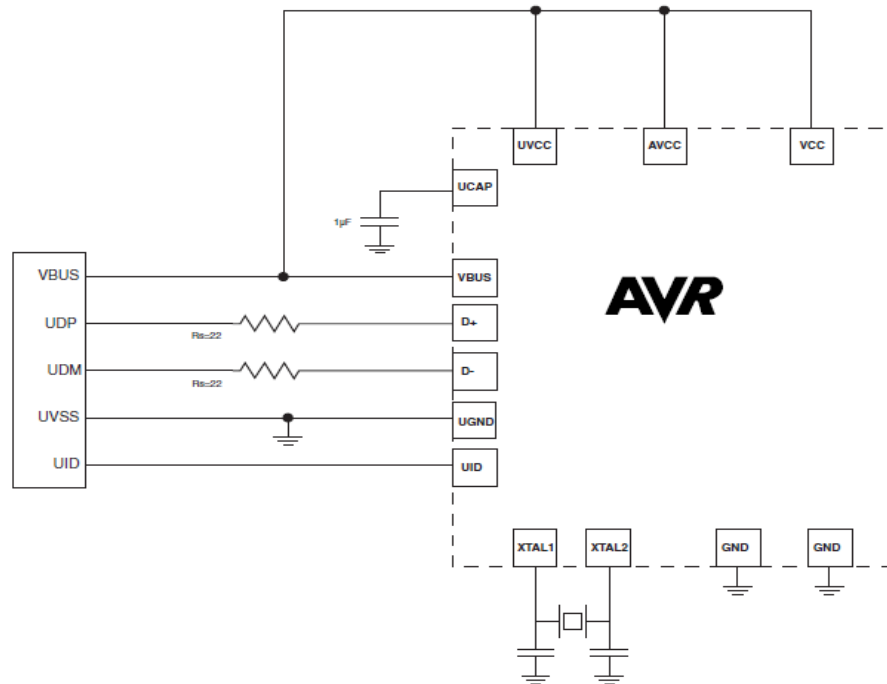
Opinnäytetyön myöhemmässä vaiheessa, kun tuli aika tilata komponentteja työtä varten, muuttui suunnitelma vielä kertaalleen. Tutkittuani useampien komponenttien jälleenmyyjien varastoja minulle selvisi, että työhön suunnittelemani analogia-digitaalimuunnin oli hyvin hankalasti saatavissa. Mietittyäni erilaisia ratkaisuja ja konsultoituani opettajaa päädyin käyttämään mikrokontrollerissa olevaa analogia-digitaalimuunninta. Olin aiemmin hylännyt tämän vaihtoehdon koska ajattelin, että pienimmätkin jännitepiikit saattaisivat rikkoa koko mikrokontrollerin, mutta kun jännitejako toteutettiin tarkkuusvastuksilla, epäilykseni väistyivät. Mikrokontrolleri-luvussa on kerrottu tarkemmin piirin omasta A/D-muuntimesta.

Laitteeseen piti suunnitella myös virtalähteen käynnistys. Kytkentä suunniteltiin virtalähteen ns. normaalia käynnistysprotokollaa hyväksi käyttäen, eli tehdään kytkentä samalla tavalla kuin se on toteutettu tämänpäiväisissä tietokoneiden emolevyillä.

Kaikessa yksinkertaisuudessaan päävirtaliittimen pinni numero 16 oikosuljetaan napin kautta maahan hetkeksi ja tämä käynnistää virtalähteen. Samaan yhteyteen suunniteltiin myös merkkivalo, joka ilmoittaa koska virtalähteen jännitteet ovat tasoittuneet riittävästi, että ne ovat mittauskelpoisia. Tämä tehtiin laittamalla virtalähteen ”powergood”- pinniin led-valo,  $140 \Omega$  etuvastuksen kanssa ja kytkettiin ledin toinen jalka maahan. Virtalähde antaa viiden voltin pulssin, kun jännitteet ovat tasoittuneet ja tämä sytyttää ledin, minkä jälkeen voidaan aloittaa jännitteen mittausta. (Power supply unit (computer) 2010.)

Laitteeseen tuleva USB-kytkentä vaatii ulkoisen oskillaattorin toimiakseen, koska piirin sisäinen kello ei ole soveltuvainen tähän tarkoitukseen. Tämän takia laitteeseen kytkettiin 8 MHz:n ulkoinen oskillaattori. Kytkentä tehtiin datakirjan ohjeiden ja kuvion 4 mukaan. Oskillaattori kytkettiin pinneihin xtal1 ja xtal2 välille sekä molemmista jaloistaan 15 pF:n kondensaattoreihin ja niiden kautta maahan. mikrokontrolleriin (Atmel 2009.)





KUVIO 4. Usb-liittimen kytkeminen AT90USB1287 mikrokontrolleriin (*Atmel 2009.*)

Ohjelman yhteydessä olevan mahdollisen virheen varalle laitteeseen suunniteltiin koodin nollauskytkentä ja tehdasasetuksien palautus. Kytkentä suunniteltiin datakirjojen ohjeiden ja vaatimuksien mukaan. 47 Ω:n vastus ja 220 nF:n kondensaattori kytkettiin rinnan: toinen pää kytkettiin käyttöjännitteeseen ja toinen pää reset-pinniin sekä painonapin kautta maahan. Napin painaminen aiheuttaisi laitteen nollauksen. Samanlainen kytkentä tehtiin myös hwb-pinniin. Hwb on ns. hardware bootloader, jota painetaan samaan aikaan reset napin kanssa. Tällä tavoin pakotetaan piirille sen oletusohjelmointi ja päästään eroon virheellisestä ohjelmasta. (*Atmel 2009.*)

Lisäksi laitteeseen suunniteltiin neljäporttinen kytkin, jolla valitaan mitä A/D-muuntimen kanavaa ”kuunnellaan” sekä minkä jännitetulon mittausrvo näkyy LCD-näytöllä. Kytkimen toiset jalat kytkettiin mikrokontrollerin neljään vapaaseen I/O-pinniin. Toiset jalat kytkettiin maahan. Eri nappiyhdistelmät käydään läpi ohjelmoinnissa, jossa selvitetään miten laitetta ohjataan.

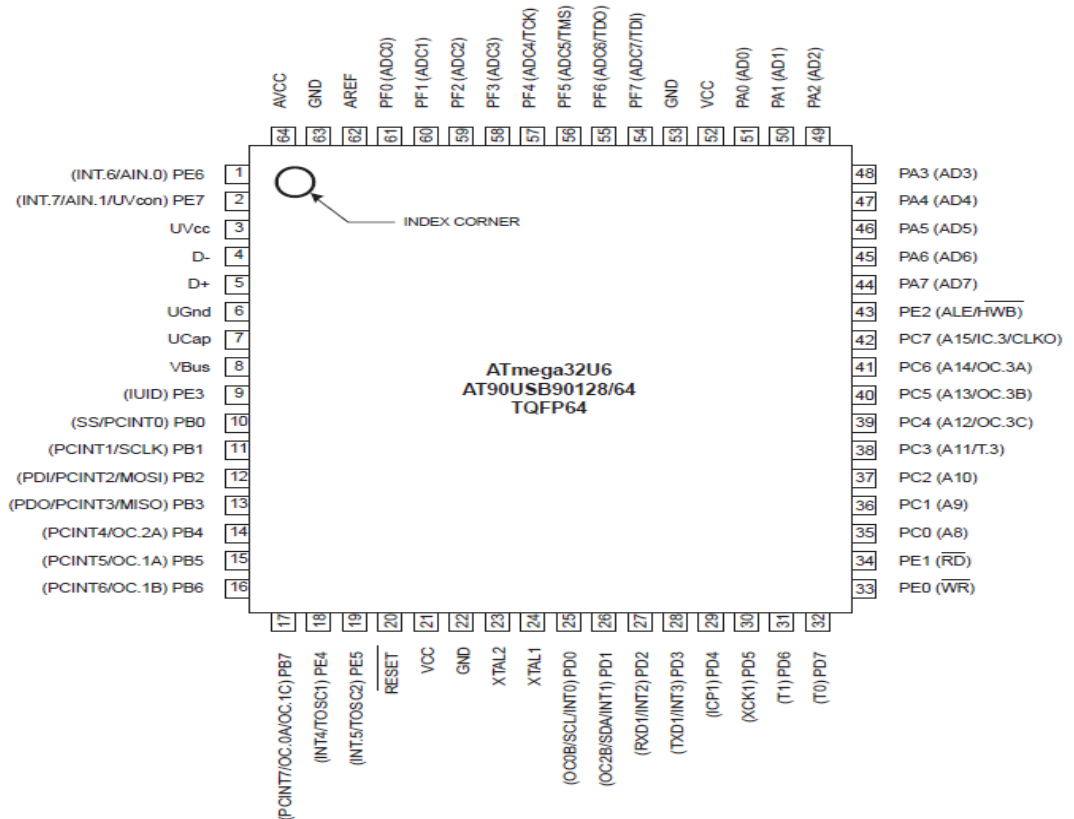
Kaikki tässä luvussa mainitut asiat suunniteltiin ja tutkittiin osittain samaan aikaan, kun laitteelle suunniteltiin piirikaaviota, jonka pohjalta suunniteltaisiin itse piirilevy.

### 3.1.1 Mikrokontrolleri AT90USB1287

AT90USB1287 on Atmelin valmistama 8-bittinen mikrokontrolleri. Kuten lähes kaikki Atmel AVR -mikrokontrollerit, myös AT90USB1287 on suunniteltu RISC-arkkitehtuurin pohjalta. Mikrokontrollerissa on 32 yleisrekisteriä, jotka on kaikki kytketty piirin aritmeettisloogiseen yksikköön. Tämä mahdollistaa kahden yksittäisen rekisterin käytön yhden yksittäisen käskyn aikana yhden kellojakson aikana. Arkkitehtuuri on tämän ansiosta huomattavasti kooditehokkaampi ja kymmeniä kertoja CISC-arkkitehtuuriin perustuvia, mikrokontrollereita nopeampi. Kuviossa 5 on kuvattu mikrokontrollerin TQFP64-kotelo ja pinnit.

Mikrokontrollerista löytyy useita hyödyllisiä ominaisuuksia, jotka mahdollistavat mikrokontrollerin käytön useissa eri sovelluksissa. Ominaisuuksia ovat mm.

- A/D-muunnin
- 48 ohjelmoitavaa I/O-pinniä
- kolme erilaista ohjelma- ja datamuistia
- 2.0 USB, SPI ja UART-liitäntämahdollisuus
- analoginen komparaattori
- pulssinleveysmodulaatio
- kuusi erilaista virransäästötilaa.



Kuvio 5. AT90USB1287 pinnien järjestys TQFP64 kotelossa. (Atmel. 2009.)

Tutustutaan tässä opinnäytetyössä tarvittaviin ominaisuuksiin tarkemmin. Ensimmäisenä otetaan käsittelyyn ADC eli analogia-digitaalimuunnin. Sitä käytetään muuntamaan analoginen signaali bittimuotoon. ADC koostuu kahdeksasta kanavasta, joilla on mahdollista muuttaa differentiaalisia ja positiivisia jännitteitä. Suurin mahdollinen muutettava jännite on riippuvainen piirin omasta käyttöjännitteestä, eli käytännössä suurin mahdollinen muutettava jännite on noin viisi voltia. Tulojännitteen ollessa suurempi kuin vertailujännite saattaa piiri kärsiä suuremmasta jännitteestä ja vahingoittua. Koska piiri käyttää 10-bittistä resoluutiota on maksimiarvo bitteinä ilmoitettuna 111111111 eli 1023. Kerron A/D-muunnoksesta tarkemmin ja yksityiskohtaisemmin ohjelmoinnin yhteydessä. (Atmel. 2009.)

Kuten edellä on mainittu, mikrokontrollerissa on 48 ohjelmitavaa I/O-pinniä. Tämä tarkoittaa käytännössä sitä, että porttien tilaa voidaan vaihtaa toisistaan riippumatta, esimerkiksi tuloiksi asetettujen pinnien sisäinen ylösveto voidaan kytkeä

päälle tai pois päältä sen vaikuttamatta mihinkään muuhun. Käyttöjännite ja maataso on jokaisessa pinnissä suojattu diodilla. Suurimmalla osalla porteista on myös vaihtoehtoinen käyttötarkoitus, tämä näkyy hyvin kuviossa 5. Ohjelmoinnin yhteydessä selvitän tarkemmin porttien käytön tässä opinnäytetyössä. (Atmel. 2009.)

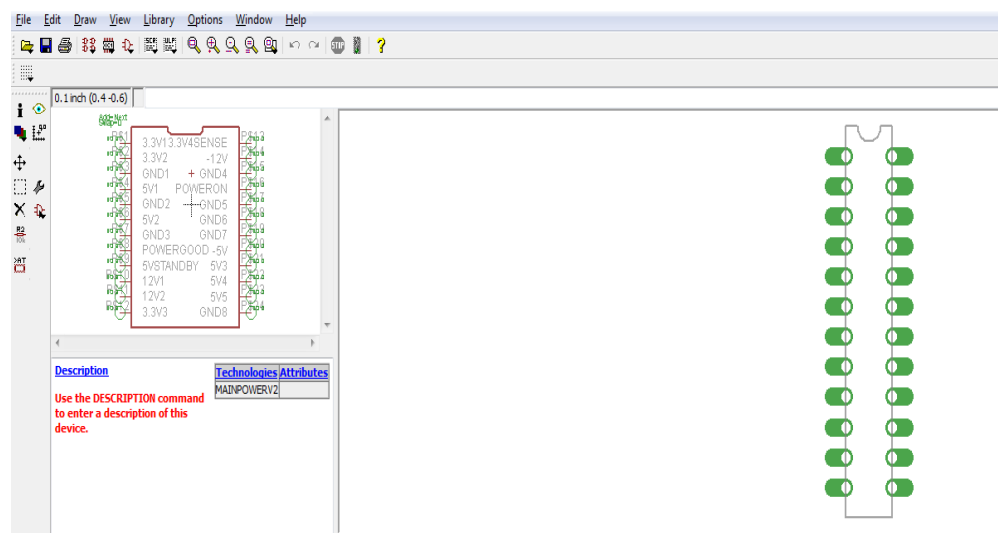
Tärkein ominaisuus opinnäytetyötäni varten on mikrokontrollerissa oleva USB 2.0 liitettävyyys. USB:lle on tehtaalla piirille ohjelmoitu boot-koodia latausta varten ohjelma oletuksena. USB-liitäntää voi käyttää lähes mihin sovellukseen tahansa, vaikka sitä tässä työssä käytetäänkin ainoastaan koodin lataamiseen piirille. Liitäntä mahdollistaa esimerkiksi tietokoneen hiiren valmistamisen kallistusantureilla ja sopivalla rajapinnalla. (Atmel. 2009.)

### 3.1.2 Piirikaavio ja piirilevy

Opinnäytetyön käytännön osuutta varten suunniteltiin piirikaavio ja -levy. Piirikaavio suunniteltiin suunnitelmaluvussa käytyjen vaatimuksien pohjalta. Piirikaavio ollessa valmis, suunniteltaisiin sen pohjalta piirilevy, jossa sijoitetaan komponentit paikoilleen ja luotaisiin kuparivedot komponenttien välille. Piirilevy-suunnitelma siirrettäisiin sitten piirilevyille, joka toimisi käytännön osuuden pohjana.

Piirikaavion ja piirilevyn suunnittelemiseen valitsin ohjelmaksi EAGLE:n (Easily Applicable Graphical Layout Editor). Suunnittelun olisi myös voinut tehdä PADS-ohjelmalla, jota käytimme oppikursseilla. Mielestäni PADS on paljon monimutkaisempi käyttää ja opastus ohjelmaan tutustuttaessa, oli ollut varsin puutteellista. Vaikka en ollut aiemmin käyttänyt EAGLE-ohjelmistoa, opin sen ohjelman oman opastuskirjan kautta nopeasti. Ohjelman valintaan vaikutti myös laajat komponenttikirjastot sekä Internetistä saatavissa olevat lisäkirjastot. Ohjelma on yksi harvoja, jota saa käyttää ilmaiseksi harrastuskäyttöön, vaikka piirilevyalue on rajattu 8 x10 cm:n alueeseen. Ohjelmassa on erilliset osiot sekä piirikaavion että piirilevyn suunnittelemiseen sekä myös osio, jossa piirilevynsuunnitelma voitaisiin koodata erillisille piirilevyjyrsimille.

Piirikaavion suunnittelu aloitettiin kartoittamalla mitä komponentteja EAGLE:n kirjastoista löytyi. Kaikkia komponentteja ei tietenkään löytynyt valmiina, joten suurin osa niistä jouduttiin luomaan itse. Ensimmäisenä piti luoda oma kirjasto omille komponenteille. Uuden komponentin luonti aloitettiin sähköisen piirto-merkin tekemisellä, jossa olisi oikea määrä pinnejä. Piirrosmerkin koolla ja laitteen todellisilla mitoilla ei ollut niin paljon väliä tässä vaiheessa. Tämän jälkeen komponentille suunniteltiin piirilevy-suunnitelmassa käytettävä kuva. Tässä vaiheessa luin komponentin datakirjasta tarkat mittasuhteet, kuten komponentin ulkomitat ja pinnien etäisyys toisistaan millimetrin tarkkuutta käyttäen. Kun molemmat piirrosmerkit olivat valmiita, yhdistettiin ne yhdeksi komponentiksi.

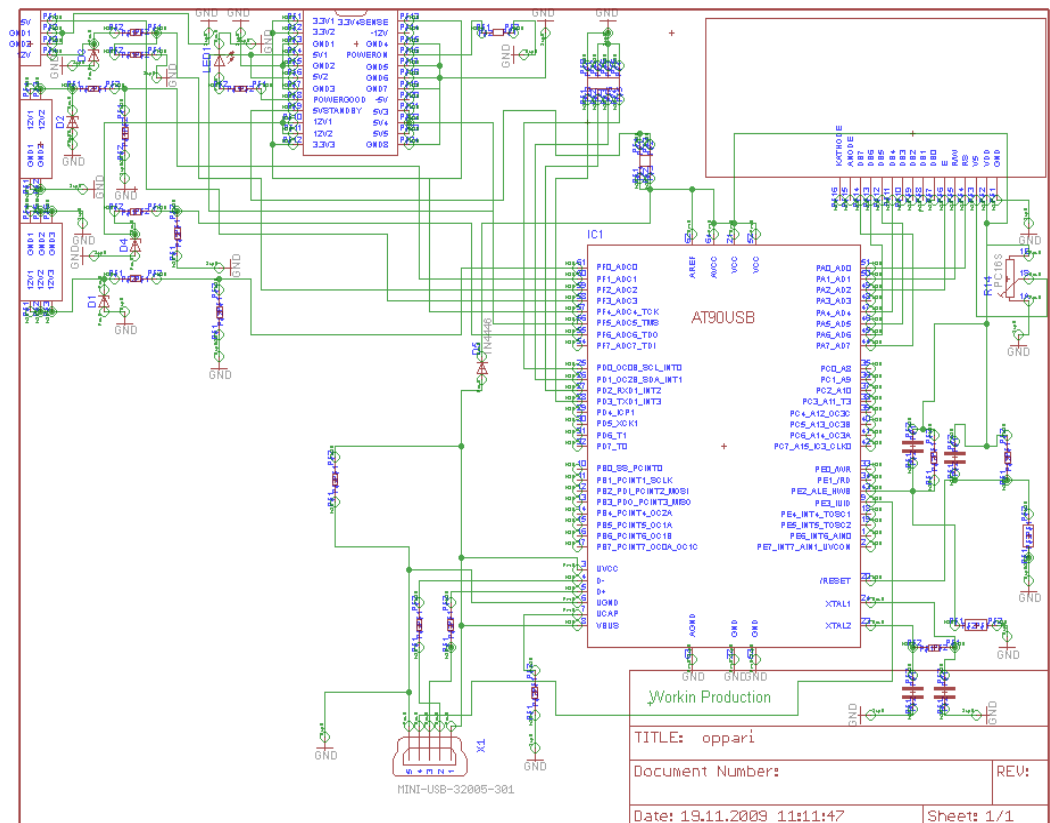


## KUVIO 6.. Komponentin luominen EAGLE:lla

Kuviossa 6 näkyy piirikaavioille luotu piirrosmerkki vasemmassa yläkulmassa ja kuvan oikealla puolella näkyy piirilevyä varten tehty piirrosmerkki. Näiden vaiheiden jälkeen komponenttia voitiin käyttää opinnäytetyön suunnittelussa. Kaiken kaikkiaan minun piti luoda kymmenen erilaista komponenttia, jotta ne olisivat fyysiseltä kooltaan samanlaisia. Jalallisille komponenteille suunniteltiin 0,8 mm:n poraukset.

Piirikaavio tehtiin suunnitteluluvussa esiteltyjen vaatimuksien pohjalta. Piirikaavion tekemisen aloitin sijoittamalla kaikki suurimmat komponentit kaaviolle. Tämän jälkeen piirsin niiden välisiä yhteyksiä ja lisäsin niiden vaatimat pienemmät komponentit sitä mukaan kun huomasin, että niitä tarvittaisiin. USB-liitäntä suunniteltiin toisen laitteen datakirjan ohjeiden mukaisesti; siinä oli käytössä sama mikrokontrolleri, joten suunnittelu oli helppoa. Käytännössä kytkennässä laitettiin samannimiset johtimet yhteen käyttämällä muutamaa pienempää komponenttia avuksi.

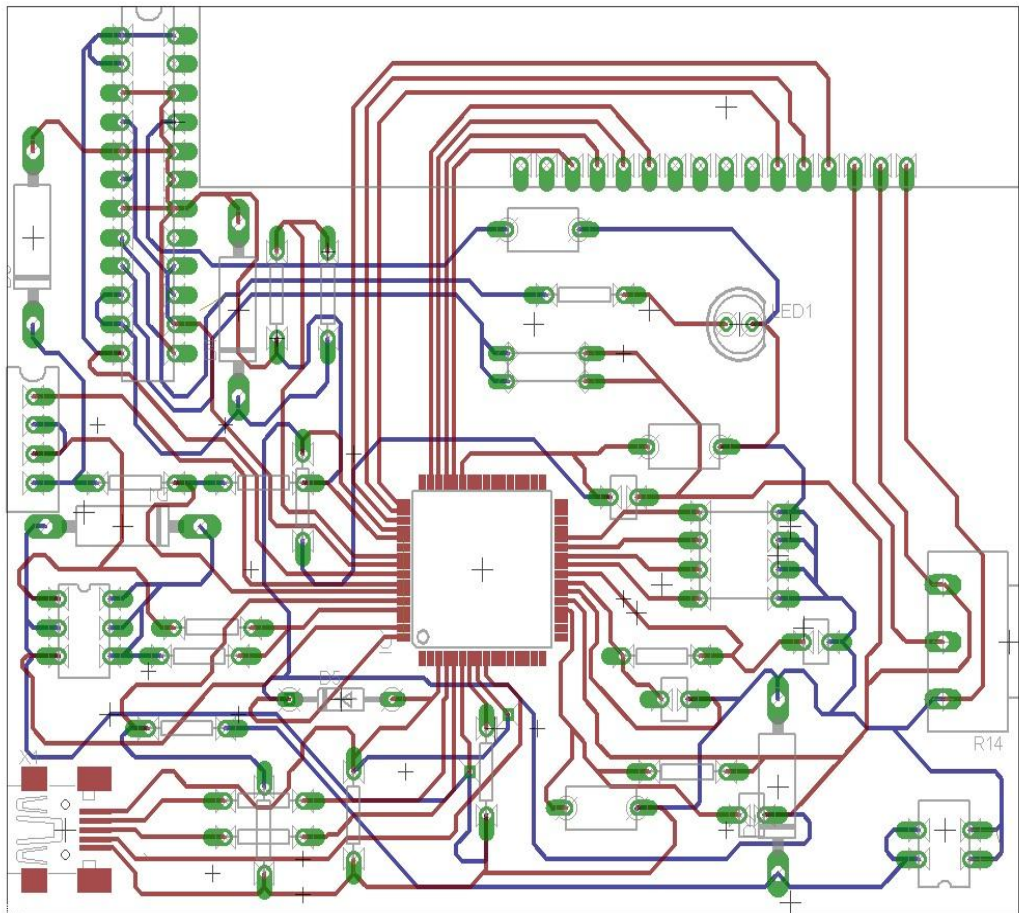
Aiemmin mainitun AD0848-piirin hankalan hankinnan takia jouduin korjaamaan piirikaaviota jälkeinpäin. Piirilevysuunnitelman yhteydessä saatoin myös vielä muuttaa joidenkin komponenttien paikkoja I/O-pinneissä, jotta niille olisi helpompi tehdä kuparivedot.



KUVIO 7. Tietokoneen virtalähteen jännitemittarin piirikaavio EAGLE-ohjelmalla suunniteltuna

Piirilevysuunnittelu aloitettiin kuviossa 7 näkyvän piirikaavion pohjalta. Komponentit sijoitettiin rajatulle alueelle, joka EAGLE Light -versiossa on 8 x 10 cm. Komponentteja pyörittelin ympäriinsä, jotta saisin mahdollisimman helpot vedot tehtyä. Osien sijoittelua ei meinannut saada valmiiksi millään, koska aina tuntui, että komponentit oli jatkuvasti huonosti sijoiteltu. Yritin korjata tätä vaihtamalla osien paikkaa I/O-pinneissä ja sainkin hieman apua tällä tavoin. Lopulta ajattelin vain alkaa tehdä vetoja valmiiksi, että saisin työn etenemään ja pikkuhiljaa työ alkoi näyttää hieman valmiimmalta. Levystä suunniteltiin alusta alkaen kaksipuolinen, mutta koska mikrokontrolleri ja USB-liitin olivat pintaliitoskomponentteja, tarvitsi kaikki niistä lähtevät vedot vetää samalle puolelle, mikä aiheutti suurimmat ongelmat USB-liittimen sijoituksen suunnittelussa. Asiassa ei myöskään auttanut se, että olin alussa sijoittanut liittimen hyvin ahtaaseen tilaan ja jouduin lähes valmiin piirilevyn kanssa ongelmiin ja tekemään kytkennät hyvin pieneen

tilaan. Piirilevyn kuparivedot tehtiin 16 mils:n levyisiksi. Lisäksi suunnittelussa pyrittiin noudattamaan yleisiä suunnitteluohjeita, joista esimerkkeinä ovat 90° kulmien välttäminen kuparivedoissa, liittimien sijoittaminen piirilevyn reunoille, vältettiin pitkien yhdensuuntaisten vetojen, hyppylankojen ja läpivetojen vetämistä. Kuviossa 8 on kuvattu valmis piirilevyn vedos, jossa punaisella värillä on kuvattu yläpuoliset vedot ja sinisellä alapuoliset vedot.



KUVIO 8. EAGLE:lla tehty piirilevy tietokoneenjännitemittarille

Työn seuraavassa vaiheessa tehtiin piirilevyn molemmista puolista erilliset tulosteet, joissa näkyivät eri komponenttien padit ja niiden väliset vedot. Tulosteet tehtiin läpinäkyvälle kalvolle, minkä jälkeen ne kohdistettiin keskenään reunaviivojen mukaisesti ja teipattiin yhteen yhdestä reunasta. Koulutuskeskus Salpauksessa



opettaja Jarmo Salon opastuksella, syövytettiin piirilevy. Siellä kalvojen väliin asetettiin ultraviolettivaloon reagoiva piirilevy. Piirilevy altistettiin tämän jälkeen noin kahdeksi minuutiksi ultraviolettivalolle. Kaikki se alue, johon ultraviolettivalo pääsi vaikuttamaan, tulisi syövytyksen yhteydessä syöpymään pois piirilevyltä pois.

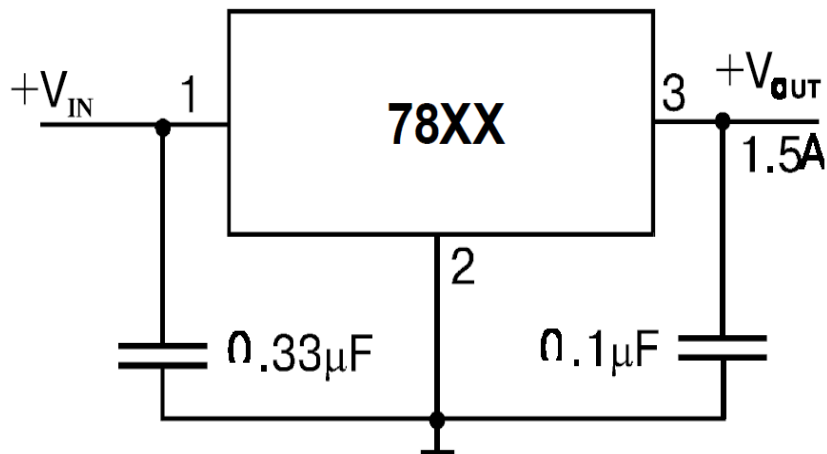
Syövytyksessä havaittiin myös ongelma. Noin 40 minuutin syövytyksen jälkeen piiri näytti syöpyvän erittäin hitaasti. Yritimme selvittää tähän syytä ja hetken tutinnan jälkeen havaitsimme, että syövytysreaktiota nopeuttava hapon lämmitin oli pois päältä. Tämän vuoksi piiri oli nyt jo ollut liian kauan hapossa ja tämä aiheuttaisi myöhemmässä vaiheessa ongelmia.

### 3.1.3 Piirilevyn kokoaminen

Opinnäytetyön kokoaminen koostui useammasta eri vaiheesta. Ensimmäisenä piti piirilevylle porata reiät, jalallisia komponentteja varten. Reiät tein 0,8 mm:n poranterällä, kuten ne olin aikaisemmin piirilevynsuunnittelussa mitoittanut. Sovitin muutamalla mukana olleilla komponenteilla, että reiät olisivat riittävän suuret. Porauksien jälkeen piiri pyyhittiin asetonilla, jotta kuparin hapettumista vastaan suojaava lakka saataisiin poistettua ja komponenttien juottamisen yhteydessä tina tarttuisi kuparipintoihin.

Piirilevyn komponenttien asettelun aloitin jalallisista komponenteista ja etenin osaryhmä kerrallaan eli samanarvoiset kondensaattorit samaan aikaan ja niin edelleen. Komponenttien juottamisen yhteydessä huomasin, että kaikille jalallisille komponenteille eivät 0,8 mm:n reiät riittäneet. Esimerkiksi virtalähteen liittimien kiinnitykset olisivat vaatineet 6 mm porauksen. Ratkaisuksi koetin etsiä liittimiä piirilevylle, johon 6 mm:n johdot olisivat mahtuneet. Liittimiä löytyi monta sopivaa, mutta yhdessäkään ei sopinut rasteri kohdalleen, joten lopulta päätin suurentaa kaikille puuttuville komponenteille poratut reiät edellä mainittuun 6 mm:n kokoon.

Laitteen kokoamisen yhteydessä ongelmia muodostui aikaisemmassa luvussa mainitun, syövytyksessä tapahtuneen virheen takia. Kuparoinnit porauskohdissa olivat erittäin heikosti kiinni piirilevyssä ja kuparointeja irtosikin huomattava määrä ja näitä yritin korjailla parhaani mukaan hyppylangoilla. Suunnitteluvaiheessa oli tapahtunut muitakin virheitä, jotka tulivat nyt kokoamisen yhteydessä esille. Suurin virhe oli käyttöjännitteen laitteelle tuovan vedon säätämättä jättäminen. Mitattavasta laitteesta otettava käyttöjännite pitää siis säätää lineaariregulaattorilla tasan viiteen volttiin, jotta se olisi hyväksyttävä. Kuviossa 9 on nähtävissä tarvittava kytkentä. Tässä tapauksessa  $V_{IN}$  on virtalähteeltä tuleva hieman yli viiden voltin jännite. Kytkentää täydennetään 330 nF:n ohituskondensaattorilla sekä muunnosta vakauttavalla 100 nF:n kondensaattorilla. Lineaariregulaattori muuttaa jännite-eroa vastaavan tehon lämmöksi. Käyttöjännitteen säätäminen tarkasti viiteen volttiin on tärkeätä, koska analogi-digitaalimuunnin käyttää tätä arvoa muuttaessaan mitattavia jännitteitä digitaaliseen muotoon sekä tietenkin laitteen toimivuuden kannalta. (Estek electronics. 2010.)



KUVIO 9. Käyttöjännitteelle tasaamiseen suunniteltu lineaariregulaattori kytkentä. (Estek electronics. 2010)

Osien hankinnan yhteydessä en onnistunut löytämään sopivaa liitintä CPU-lisävirtakytkentään. Näin ollen jouduin sen nyt jättämään pois työstä. Kytkin mo-

lex-liittimestä yhden maavedon kiinni kohtaan, jossa olisi pitänyt olla lisävirtaliittimen maa, ettei se vaikuttaisi muihin laitteen komponentteihin, jotka hyödynsivät liittimen ominaisuuksia. Ohitin myös kaksi mikrokontrollerin lähellä olevaa läpivetoa hyppylangoilla.

Piirilevyn työstäminen oli kaiken kaikkiaan aikaa kuluttavaa ja työlästä. Sain kuitenkin piirilevyn kasattua jonkin näköiseksi kokonaisuudeksi, vaikka se ei ollutkaan yhtään sen näköinen kuin olin sen kuvitellut näyttävän. Yhteenvedo-luvussa kerron enemmän laitteen käyttöönotosta ja toiminnasta.

### 3.2 Ohjelmointi

Kirjoitin opinnäytetyöni ohjelman mahdollisimman aikaisessa vaiheessa eli käytännössä heti, kun olin selvittänyt ohjelmointiin vaikuttavat komponentit. Minulla on opintojeni aikana ollut vaikeuksia kirjoittaa harjoitustöihini ohjelmakoodia, koska aihealue on ollut minulle vaikea. Koulussa minulla oli vielä uudelleen suoritettavana Sulautetun järjestelmän ohjelmointi -niminen kurssi, jonka aikana otin tavoitteekseni ymmärtää asian niin hyvin, että voisin kirjoittaa opinnäytetyöhöni ohjelmakoodin samalla kertaa. Kurssin jälkeen ymmärsin sulatetun järjestelmän ohjelmointia paremmin. Omaksi yllätyksekseni osasin kirjoittaa opinnäytetyöhön sopivan ohjelman, jonka teoriassa pitäisi toimia.

Kerron ensin ohjelman kirjoittamiseen vaadituista asioista teoriatasolla ja sen jälkeen käyn ohjelmakoodin yksityiskohtaisesti läpi sekä selitän jokaisen työvaiheen.

### 3.2.1 Ohjelmoinnin suunnittelu

Ohjelmoinnin suunnittelun aloitin etsimällä tietoa Internetistä LCD-näytöistä sekä siitä kuinka analogia-digitaalimuunnos koodattaisiin ohjelmaan. Käytin samoja foorumeita kuin olin käyttänyt koulussa ohjelmointikursseilla apuna. Löytämälläni harrastelijasivustolla oli selitetty yksityiskohtaisesti teoria analogia-digitaalimuunnoksen taustalla sekä löysin ajurin LCD-näytölle, joka helpottaa tekstin kirjoittamista sille huomattavasti. Samalta sivustoilta löytyi muitakin oppaita Atmelin mikrokontrollereista, joista opin erittäin paljon opinnäytetyötä varten.

Atmelin mikrokontrollerin sisäinen A/D-piiri muuttaa tulevan jännitteen digitaalseksi 10 bitin resoluutiolla. Opinnäytetyössäni 10 bitin resoluution avulla saavutetaan sadasosien tarkkuus mittauksiin. Digitaalimuunnoksesta saadaan arvoja 0-1023, jotka sitten ohjelmassa muutetaan vastaamaan jälleen oikeita jännitearvoja. (*Atmel. 2009.*)

Kuten aiemmin on kerrottu, suurin mahdollinen tulojännite voi olla maksimissaan viisi voltia. Jos tulojännite on suurempi kuin vertailujännite, saattaa mikrokontrolleri rikkoutua tai joka tapauksessa mittaustulos vääristyy, koska laite antaa kaikista vertailujännitetä vastaavista ja suuremmista arvoista vastaukseksi saman digitaalimuunnoksen arvon. Samalla periaatteella, jos tulojännite on esimerkiksi 2,5 voltia, on digitaalimuunnoksen tulos 512. (*Extreme Electronics 2008b.*)

ADC:tä voidaan käyttää joko jatkuva-ajo- tai yksittäismuunnos-tilassa. Jatkuva-ajo-tilassa ADC aloittaa uuden muunnoksen heti edellisen loputtua. Nimen mukaisesti yksittäismuunnos-tilassa mikrokontrolleri tekee yhden muunnoksen ja sitten pysähtyy. Opinnäytetyössäni toteutin muunnoksen yksi kerrallaan, koska jatkuva-ajo-tilassa en pystyisi muuttamaan ”kuunneltavaa” ADC:n kanavaa. Muunnosta varten ADC:lle määritettiin esijakaja, jonka avulla ADC:lle määritetään kello. Suurella kellotaajuudella muunnos olisi nopea ja hitaalla erittäin tarkka. ADC:n kellotaajuus tulisi olla 50–200 kHz. Ohjelmassa valitsin tietysti mah-

dollisimman pienen kellotaajuuden, jotta laite pystyisi hyvään mittaustarkkuuteen eli tässä tapauksessa valitsin esijakajaksi 128. Mikrokontrollerin kellotaajuuden ollessa 8 MHz saadaan A/D-muunnoksen kellotaajuudeksi 62,5 kHz. . (Extreme Electronics 2008b.)

ADC-muunnoksen käynnistämiseksi pitää mikrokontrollerille alustaa ADMUX- ja ADCSRA-rekisterit. ADMUX-rekisteristä määritetään referenssijännite, johon laite vertaa tulojännitettä. Määrittäminen kuvion 10 esiintyvien bittien REFS0 ja REFS1 avulla. Opinnäytetyössä kirjoitin REFS0 loogiseksi ykköseksi, josta johtuen työssä käytetään AV<sub>CC</sub>:hen tulevaa jännitettä, vertailujännitteenä. (Extreme Electronics 2008b.)

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AV <sub>CC</sub> with external capacitor on AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor on AREF pin

KUVIO 10. ADC:n vertailujännitteen valitseminen. (Atmel. 2009.)

Edellä mainittu esijakaja asetetaan ADCSRA-rekisteristä. Kuviosta 11 nähdään, että tässä tapauksessa bitit ADPS0–ADPS2 kirjoitetaan loogisiksi ykkösiksi, jotta saadaan luotua esijakajaksi 128.

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

KUVIO 11. ADC:n esijakajan määrittäminen (Atmel 2009.)

Muunnoksen aloittamiseksi pitää ADCSRA-rekisteriin myös muuttaa bittien ADEN ja ADSC tilaa. ADEN-bitillä hallitaan analogiadigitaalimuuntimen päällä oloa, eli aina kun muunnos tehdään, pitää ADEN-bitti olla looginen yksi. Kaikki lähtötiedot ja rekisterit alustettuani kirjoitin ADSC-bittiin loogisen ykköseni, jotta muunnos alkaa. (Extreme Electronics 2008b.)

LCD-kytkennän suunnitteluun löysin Internetistä erittäin hyödyllisen oppaan, jossa neuvottiin, kuinka samankaltaisten näyttöjen ohjelmointi ja kytkentä pitäisi tehdä, kun käytössä on Atmelin mikrokontrolleri. Tämänkin vaiheen aloitin opiskelemalla teoriaosuuden kytkentöjen takana ja tutkimalla samankaltaisen näytön datalehteä, jonka olin omaan projektiini hankkinut.

Opinnäytetyöhöni olin valinnut näytöksi yksinkertaisen 16 x 1 matriisinäytön, jolle voidaan ohjelmoida näkyviin 16 merkkiä yhdelle riville. Näyttöni käyttää samaa piiriä, joka on käytössä useissa erilaisissa matriisinäytöissä, joten kaikkia kuviossa 12 esiintyviä pinnejä ei tarvinnut tässä opinnäytetyössä kytkeä. 16-merkkinen näyttö tarvitsee ainoastaan neljä datapinniä, joten jätin pinnit 7–10 kytkemättä. Pinnit 15 ja 16 jätettiin kytkemättä, koska minun näytössäni ei ollut tukea taustavalolle. Tutkittuani matriisinäytöistä tehtyä yleistä opasta löysin niiden liitteenä LCD-näytöille sopivan ajurin, jonka avulla ohjelmointi sujui huomattavasti yksinkertaisemmin. Tällä tavoin koodista saatiin tiiviimpi ja välttyttiin monimutkaiselta merkkikirjastoiden käytöltä. Muokkasin ajurin header-tiedostoa omaan projektiin sopivaksi ja kirjoitin ohjelman, joka hyödyntäisi ajuria kirjoitta-

essaan analogidigitaalimuunnoksen tuloksen näytölle. Header-tiedostoon muokattiin pinnit, joihin näyttö kytkettäisiin kiinni mikrokontrolleriin, koska kyseessä oli yleiseen käyttöön sopiva ajuri eivätkä valmiiksi määritetyt pinnit olleet vapaina tai sopineet yhteen jo valmiiksi suunnitellun piirikaavion kanssa. (*Extreme Electronics 2008a.*)

PIN ASSIGNMENT		
Pin no.	Symbol	Function
1	Vss	Power supply(GND)
2	Vdd	Power supply(+)
3	Vo	Contrast Adjust
4	RS	Register select signal
5	R $\bar{W}$	Data read / write
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for LED B/L (+)
16	K	Power supply for LED B/L (-)

KUVIO 12. 16 x 1 matriisi näytön pinnit. (*Powertrip. 2010.*)

Näytölle kirjoittaminen onnistui muutamalla yksinkertaisella komennolla. LCDWriteStringXY:n avulla voidaan kirjoittaa mihin tahansa näytöllä sijaitseviin koordinaatteihin. LCDWriteintXY:llä saadaan näytölle numeroita haluttuihin koordinaatteihin. Esimerkiksi jos halutaan kirjoittaa numero kahdeksan näytön keskelle, kirjoitetaan koodiksi LCDWriteIntXY(8,8), tällöin näytön keskelle kirjoitetaan 00000008. Tekstin kirjoittaminen ilman nollia vaatii kursorin siirtämisen

ensiksi haluttuun paikkaan ennen kirjoitusta. Komennolla LCDgotoXY saadaan kursori siirretty haluttuihin koordinaatteihin ja tämän jälkeen voidaan kirjoittaa ilman, että näytölle ilmestyy täyttömerkkejä. Näillä yksinkertaisilla komennoilla pystyttiin kirjoittamaan kaikki tarvittu teksti näytölle. Selvitän seuraavassa luvussa tarkemmin, kuinka niitä käytetään opinnäytetyössäni. (*Extreme Electronics 2008a.*)

### 3.2.2 Ohjelma

Tässä luvussa käsittelen ohjelmakoodin kohta kohdalta sekä kerron ohjelman sisällöstä. Ohjelma tehtiin osaksi AVR Studio 4:llä ja osaksi CodevisionAVR:llä. Jälkimmäistä tosin käytettiin vain koodin ensimmäiseen versioon. Ensimmäisestä koodista ei lopulliseen ohjelmaan juurikaan mitään säilynyt. Kyseinen ohjelmakoodi löytyy opinnäytetyön liitteenä yksi.

Ohjelman alussa esitellään eri kirjastot, joita työssäni olen käyttänyt. Tämän lisäksi asetetaan mikrokontrollerin kellotaajuus 8 MHz:iin ja esitellään globaali muuttuja ”tila”, jolla ohjataan pääohjelman sisällä nimensä mukaisesti ohjelman tiloja. Tässä kohdassa hypätään välissä olevien apuohjelmien ylitse ja palataan niihin, kun pääohjelmassa niitä tarvitaan. Pääohjelma aloitetaan esittelemällä suhdeluvut yksi ja kaksi. Muuttujia käytetään palauttamaan jännitejaon läpi tulevat jännitteet vastaamaan todellisia arvoja. Samalla alustetaan portit, joihin on kiinnitetty kanavanvalitsinkytkin. Pääohjelmassa esitellään muuttuja, johon luetaan A/D-muunnoksen tulos. Vielä ennen ensimmäiseen apuohjelmaan pääsyä alustetaan LCD-näyttö ja sen kursori sekä tyhjätyö näyttö.

ADCalustus on ensimmäinen koodissa olevista apuohjelmista. Ohjelma alustaa nimensä mukaisesti A/D-piirin ja sen vaatimat rekisterit edellisessä luvussa mainittujen määritteiden mukaisesti. Rekistereistä valitaan vertailujännite sekä asetetaan kellotaajuuden esijakaja.

Alustuksen jälkeen ohjelma siirtyy tilanvalintaa. Kanavanvalitsimen kytkinten ollessa johtavassa tilassa ohjelma pysähtyy odottamaan. Ohjelma jatkaa suoritus- taan vasta, kun kytkimet on asetettu johtamattomaan tilaan.



Seuraavassa vaiheessa ohjelma lukee jälleen kanavanvalitsinta, mutta tällä kertaa haetaan tiloja, jossa yksi tai useampi kytkin on johtavassa tilassa. Esimerkiksi kytkin yksi on johtavassa tilassa eli porttiin D tulee arvoksi 1110. Ohjelma huomaa tämän ja kirjoittaa näytölle valmiiksi PCI- E -tekstin. Tämän jälkeen koodi alkaa lukea `adc_tulos`-muuttujaan analogiadigitaalimuunnoksen arvoa, apuohjelmasta `LueADC`.

`LueADC` on toinen koodissa olevista apuohjelmista. Apuohjelman sisällä suoritetaan A/D-muunnos. Pääohjelmassa annetaan `LueADC`-apuohjelman muuttujalle `ch` arvo väliltä 0 - 6. Luvulla valitaan apuohjelmassa mitä ADC-kanavaa ”kuunnellaan”. Kanavan valinnan jälkeen `ADCSRA`-rekisteriin kirjoitetaan `ADSC` loogiseksi ykköseksi, joka käynnistää analogiadigitaalimuunnoksen. Ohjelman suoritus pysähtyy odotuskomentoon. Se pysyy paikallaan tässä niin kauan, että mikrokontrolleri kirjoittaa `ADIF`-bitin ykköseksi, joka tarkoittaa muunnoksen valmistumista. `ADIF`-bitti nollataan ennen apuohjelmasta poistumista sekä palautetaan analogiadigitaalimuunnoksen arvo `adc_tulos`-muuttujaan.

Pääohjelmassa mitattu tulos kerrotaan suhdeluvuilla, jotta muunnos vastaisi oikeata arvoaan. Tämän jälkeen näytölle kirjoitetaan tulos oikeilla komennoilla ja lisätään perään oikea yksikkö. Tämän jälkeen päästään kolmanteen apuohjelmaan, nimeltään odotus. Tässä apuohjelmassa luodaan viivettä, jotta teksti pysyisi kauemmin näytöllä. Viiveen mentyä laite siirtyy tila kahteen, jossa LCD-näyttö tyhjennetään ja laite siirretään tilaan yksi.

Ohjelmaan on kirjattu jokaiselle mitattavalle kanavalle oma osio. Kävin tässä esimerkissä ainoastaan yhden niistä läpi.

## 4 YHTEENVETO

Opinnäytetyöni viimeisessä pääluvussa käsittelen työtä kokonaisuutena. Kerron mitkä asiat mielestäni onnistuivat ja mitkä eivät ja ennen kaikkea miksi. Yritän myös miettiä, mitä asioita olisi voinut tehdä erilailla, jotta lopputulos olisi ollut parempi.

### 4.1 Onnistumiset, epäonnistumiset ja korjaukset

Opinnäytetyössä oli useampia alueita, joissa onnistuin yllättävänkin hyvin, ja muutama sellainen, joista olisin halunnut suoriutua paremmin. Positiivinen asia kuitenkin on, että useimpiin epäonnistuneisiin asioihin olisi olemassa helpohko korjaus ja toiset virheet voitaisiin välttää uudemmalla yrityskerralla. Kerron jokaisesta osa-alueesta omassa kappaleessaan, jotta niistä saa hyvän kuvan.

Onnistumisia tuli kaikissa projektin vaiheissa, vaikka lähes kaikissa opinnäytetyön vaiheissa oli myös jotain ongelmia. Laitteen suunnittelun alussa mielestäni onnistuin ongelmien ratkomisessa hyvin ja kaikkiin eteen tulleisiin ongelmiin löysin ratkaisun. Suurin saavutukseni ja onnistumiseni oli projektin loppuun saattaminen, vaikkakin ainoastaan teoriassa. En ole kertaakaan ammattikorkeakouluopintojeni aikana saanut suunniteltua laitetta alusta loppuun. Aina on tullut vastaan jokin ylitsepääsemätön ongelma, jonka jälkeen en ole pystynyt etenemään työssäni. Joten tällä kertaa oli erittäin palkitsevaa, että asiat etenivät ja valmistuivat takerteluista huolimatta.

Toinen mielestäni onnistunut asia oli ohjelma, jonka laitteelle kirjoitin. Opinnäytetyön suunnitteluvaiheessa stressasin ohjelman kirjoittamisesta, koska aikaisemmat projektit olivat siltä osa-alueelta epäonnistuneet.. Yllätyin positiivisesti, kun osasin kirjoittaa laitteelle ohjelman ja vielä sillä tavoin, että itse ymmärsin jokaisen ohjelman vaiheen.

Piirilevyn suunnittelu alusta asti ohjelmalla, jota en ollut aiemmin käyttänyt, luotakoon onnistumiseksi. Huolimatta siitä että, lopulliseen piirilevyyn jäi useita virheitä, jotka olisi pitänyt ottaa huomioon piirilevysuunnittelun onnistumiseksi.

Tässä oli lueteltuna kaikki maininnan arvoiset onnistumiset. Opinnäytetyön aikana oli toki muitakin onnistumisia, jotka olivat pieniä voittoja pitkässä kamppailussa projektin valmistumisessa.

Epäonnistumisia ja pettymyksiä tapahtui työn edetessä. Suurimman osan vastaan tulleista ongelmista pystyin ratkaisemaan projektin yhteydessä, mutta muutama ongelma esti ensimmäisen versiota toiminnan. Laitteen ensimmäisen version toimimattomuus on useamman asian summa.

Piirilevyn suunnittelussa tapahtui pari suurempaa virhettä. Ensimmäinen näistä oli käyttöjännitteen regulointikytkennän pois jättäminen, joka esiteltiin luvussa 3.1.3. Kytkentää on lähes mahdoton tehdä tiiviisti suunnitellulle piirilevylle jälkikäteen. Toinen virhe oli tilankäytön suunnittelu ja joidenkin komponenttien asettelu. Nyt jälkeenpäin ajateltuna olisin sijoitellut komponentit väljemmin.

Ongelmat ilmenivät vasta laitteen kokoamisen yhteydessä. En osannut hahmottaa sijoitteluongelmia, ennen kuin näin ongelman fyysisesti edessäni. Ennen piirilevyn valmistamista olisi pitänyt odottaa, että kaikki tilatut komponentit ovat saapuneet. Näin olisi voitu ottaa huomioon komponenttien mukanaan tuomat muutokset piirilevysuunnitelmassa.

Toinen epäonnistuminen oli syövytyksessä tapahtunut piirilevyn liiallinen hapolle altistuminen. Levylle on lähes mahdoton juottaa komponentteja, koska ohuet 16 milsin levyiset vedot irtosivat piirilevyltä altistuttuaan liian kauan hapolle. Tämä teki laitteen kokoamisvaiheesta erittäin vaikean. Asiaa ei helpottanut yhtään vähäinen kokemukseni juotoskolvin ääressä.

Nämä syyt olivat suurimpia laitteen fyysisen osuuden toimimattomuuteen. Opinnäytetyössä oli muitakin vaikeuksia matkan varrella, mutta koska onnistuin ne ratkaisemaan, en laske niitä epäonnistumisiksi.

Lähes kaikki edellisessä kappaleessa mainitut epäonnistumiset olisi mahdollista korjata, jos laitteen rakentaisi uudestaan alusta alkaen. Laitteen komponenttien sijoitteluun käyttäisin enemmän aikaa ja mieltäisin komponenttien vaikutusta ko-

koonpanon yhteydessä. Lisäisin myös lineaariregulaattori kytkennän levyille sekä poistaisin tässä suunnitelmassa turhiksi osoittautuneet komponentit.

Tämän lisäksi paksuntaisin ohuimpia kuparivetoja välttääkseni niiden irtoamisen. Yrittäisin välttää piirilevyn liialle hapolle altistumisen, mutta asia ei olisi minun käsissäni, sillä syövytyksen tekisi joku muu kuin minä. Piirilevyn juotoksiin en ikävä kyllä voisi vaikuttaa muuten kuin palkkaamalla jonkun ulkopuolisen tekemään juotokset puolestani. Muita työtä helpottavia korjauksia olisivat liittimille sopivilla rastereilla olevat paikat sekä LCD – näytölle sopiva piikkiriva.

#### 4.2 Opinnäytetyö kokonaisuutena

Opinnäytetyön tekeminen oli pitkä projekti, johon mahtui mukaan myötä- ja vastoinkäymisiä. Alkuperäiset suunnitelmat muuttuivat useaan kertaan kohdatessani ongelmia, joita ei voinut ohittaa ilman suunnitelman muutosta. Opinnäytetyön tekemisen yhteydessä opin ja ymmärsin paremmin, mitä kaikkea laitteen suunnittelussa pitää ottaa huomioon. Suunniteltaessa pitäisi pystyä miettimään millainen laite tulisi olemaan valmiina ja miten kukin komponentti vaikuttaisi laitteen fyysiseen kokoon, muotoon ja toimintaan. Unohtamatta sitä, että kaupallisen laitteen suunnittelussa pitää ottaa huomioon laitteen häiriöntuotto ja muiden laitteiden aiheuttama häiriönesto. Projektin aikana opin myös paljon lisää sulautetun järjestelmän ohjelmoinnista.

Suunnitelmien muutoksista huolimatta onnistuin suunnittelemaan toimivan laitteen, joka tässä opinnäytetyössä oli tavoitteena. Olettaen että edellisessä luvussa mainittu lineaariregulaattorikytkentä lisättäisiin piirilevyille. Samalla yrittäisin parantaa laitetta muilta osin sekä varmistaa sen toimivuuden. Edellä mainittujen tavoitteiden saavuttamisen ansiosta pidän opinnäytetyötä onnistuneena, huolimatta siitä etten saanut laitetta toimimaan fyysisesti.

## LÄHTEET

Atmel. 2009. 8-bit AVR microcontroller with 64/128K bytes of ISP flash and USB [viitattu 21.12.2009].

Saatavissa: [http://www.atmel.com/dyn/resources/prod\\_documents/doc7593.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc7593.pdf)

Barr, M. 2001. Pulse Width Modulation, Embedded Systems Programming 9/2001, 101–103.

Estek electronics. 2010. 78xx [viitattu 28.3.2010].

Saatavissa: <http://pdf1.alldatasheet.com/datasheet-pdf/view/222818/ESTEK/78XX.html>

Extreme Electronics. 2008a. Using LCD module with AVR's [viitattu 11.4.2010].

Saatavissa: <http://extremeelectronics.co.in/avr-tutorials/using-lcd-module-with-avrs/>

Extreme Electronics. 2008b. Using the analog to digital converter [viitattu 10.4.2010] .

Saatavissa: <http://extremeelectronics.co.in/avr-tutorials/using-the-analog-to-digital-converter/>

Kontra, V. 2009. Tietokone-elektroniikan yliopettaja. Lahden ammattikorkeakoulu. Haastattelu 22.10.2009.

National semiconductor. 2008. ADC0844/ADC0848 8-bit  $\mu$ P compatible A/D converters with multiplexer [viitattu 25.11.2009].

Saatavissa: <http://www.national.com/ds/DC/ADC0844.pdf>

Powertrip. 2010. PC 1601-A [viitattu 4.4.2010].

Saatavissa: <http://www.farnell.com/datasheets/355520.pdf>

Power supply unit (computer). 2009. Wikipedia [viitattu 14.12.2009]

Saatavissa: [http://en.wikipedia.org/wiki/Power\\_supply\\_unit\\_%28computer%29](http://en.wikipedia.org/wiki/Power_supply_unit_%28computer%29)

Switched-mode power supply. 2009. Wikipedia [viitattu 10.10.2009]

Saatavissa: [http://en.wikipedia.org/wiki/Switch-mode\\_power\\_supply](http://en.wikipedia.org/wiki/Switch-mode_power_supply))

## LIITTEET

### LIITE 1: Ohjelman koodi

/\*\*\*\*\*\*

Project : 1.0

Version :

Date : 5.1.2010

Author : Antti Lakanen

Company :

Comments:

Chip type : AT90USB1287

Program type : Application

AVR Core Clock frequency: 8,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 2048

\*\*\*\*\*/

```
#include <avr/io.h>
```

```
#include <avr/interrupt.h>
```

```
#define F_CPU 8000000UL
```

```
#include <util/delay.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include "lcd.h"
```

```
unsigned char tila = 0;
```

LIITE 1.( jatkuu)

```
void ADCalustus()
{
ADMUX=(1<<REFS0);
ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
}
uint16_t LueADC(uint8_t ch)
{
//Valitaan kanava 0-6
ch=ch&0b00000111;
ADMUX|=ch;

//Aloitetaan muunnos
ADCSRA|=(1<<ADSC);

//Odotetaan muunnoksen valmistumista
while(!(ADCSRA & (1<<ADIF)));

//Nollataan ADIF seuraavaa muunnosta varten, kirjoittamalla siihen looginen
ykkönen

ADCSRA|=(1<<ADIF);

return(ADC);
}
void odotus()
{
uint8_t i;
for(i=0;i<20;i++)
_delay_loop_2(0);
}
```



LIITE 1 .(jatkuu)

```
void main()
{
    unsigned int suhdeluku;
    suhdeluku = 5/1023;
    unsigned int suhdeluku2 = 12/5;
    PORTD = 0x0f;
    uint16_t adc_tulos;
    InitLCD(LS_BLINK);
    LCDClear();
    ADCalustus();
    while(1)
    {
        switch (tila)
        {
            case 0:
                if (PIND & 0b1111)
                    tila = 1;
            case 1:

                if (PIND & 0b1110)
                {
                    LCDWriteString("PCI-E:");
                    adc_tulos=LueADC(0);
                    adc_tulos = adc_tulos * suhdeluku * suhdeluku2;
                    LCDWriteIntXY(7,0,adc_tulos,3);
                    LCDWriteStringXY(10,0,"v");
                    odotus();
                    tila = 2;
                }
        }
    }
}
```

LIITE 1.( jatkuu)

```
if (PIND & 0b1101)
```

```
{
```

```
    LCDWriteString("CPU:");
```

```
    adc_tulos=LueADC(1);
```

```
    adc_tulos = adc_tulos * suhdeluku * suhdeluku2;
```

```
    LCDWriteIntXY(5,0,adc_tulos,3);
```

```
    LCDWriteStringXY(8,0,"v");
```

```
    odotus();
```

```
    tila = 2;
```

```
}
```

```
if (PIND & 0b1011)
```

```
{
```

```
    LCDWriteString("Molex12:");
```

```
    adc_tulos=LueADC(2);
```

```
    adc_tulos = adc_tulos * suhdeluku * suhdeluku2;
```

```
    LCDWriteIntXY(9,0,adc_tulos,3);
```

```
    LCDWriteStringXY(12,0,"v");
```

```
    odotus();
```

```
    tila = 2;}
```

```
if (PIND & 0b0111)
```

```
{
```

```
    LCDWriteString("Molex5:");
```

```
    adc_tulos=LueADC(3);
```

```
    adc_tulos = adc_tulos * suhdeluku;
```

```
    LCDWriteIntXY(8,0,adc_tulos,3);
```

```
    LCDWriteStringXY(11,0,"v");
```

```
    odotus();
```

```
    tila = 2;
```

```
}
```

LIITE 1.( jatkuu)

```
if (PIND & 0b1010)
```

```
{
```

```
    LCDWriteString("Main3.3:");  
    adc_tulos=LueADC(4);  
    adc_tulos = adc_tulos * suhdeluku;  
    LCDWriteIntXY(9,0,adc_tulos,3);  
    LCDWriteStringXY(12,0,"v");  
    odotus();  
    tila = 2;
```

```
}
```

```
if (PIND & 0b0101)
```

```
{
```

```
    LCDWriteString("Main5:");  
    adc_tulos=LueADC(5);  
    adc_tulos = adc_tulos * suhdeluku;  
    LCDWriteIntXY(7,0,adc_tulos,3);  
    LCDWriteStringXY(10,0,"v");  
    odotus();  
    tila = 2;
```

```
}
```

```
if (PIND & 0b0000)
```

```
{
```

```
    LCDWriteString("Main12:");  
    adc_tulos=LueADC(6);  
    adc_tulos = adc_tulos * suhdeluku * suhdeluku2;  
    LCDWriteIntXY(8,0,adc_tulos,3);  
    LCDWriteStringXY(11,0,"v");  
    odotus();  
    tila = 2;
```

```
}
```

```
    break;
```

## LIITE 1.( jatkuu)

case 2:

```
LCDClear();
```

```
tila = 1;
```

```
break;
```

```
}
```

```
}
```

```
}
```