TUOTANNON TESTILAITE

R100-teholähteelle

LAHDEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Tietokone-elektroniikka
Opinnäytetyö
Kevät 2009
Hannu Kopsa

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

KOPSA, HANNU:

Tuotannon testilaite
R100-teholähteelle

Tietokone-elektroniikan opinnäytetyö, 36 sivua, 50 liitesivua

Kevät 2009

TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena oli suunnitella ja rakentaa tuotannon testilaite MEG Power Oy:n valmistamalle R100-hakkuriteholähteelle. MEG Power Oy suunnittelee ja valmistaa hakkuriteholähteitä ja niihin liittyviä laitteita. R100 on asiakaskohtainen hakkuriteholähde, joka poikkeaa huomattavasti muista tuotteista, joten tavanomaiset teholähteiden testauslaitteet eivät sovi sen testaamiseen. R100-laitteen testaaminen on toteutettu aikaisemmin manuaalisesti käyttämällä useita mittalaitteita. Manuaalinen testaaminen on hyvin monimutkaista ja hidasta. Tästä syystä päätettiin suunnitella R100-laitteelle oma testilaite, joka on nopea ja helppokäyttöinen.

Työ aloitettiin tutustumalla testattavaan laitteeseen ja sen manuaaliseen testausmenetelmään. Tämän jälkeen selvitettiin testauksessa tarvittavien toimenpiteiden sopivuus automaattiseen testiin. Automaattisen testauksen kannalta monimutkaiset testivaiheet pyrittiin yksinkertaistamaan niin, että niiden toteutus vaatii mahdollisimman vähän elektroniikkaa.

Testilaitteen suunnittelussa käytettiin sekä analogia- ja digitaalielektroniikkaa. Kaikkien testilaitteen toimintojen ohjaamiseen käytettiin PIC16F887-mikro-ohjainta. Mikro-ohjaimen ohjelma kirjoitettiin C-kielellä. Testilaitteen käyttöliittymä toteutettiin käyttämällä mikro-ohjaimeen kytkettävää LCD-näyttöä ja kolmea painiketta.

Testilaitteen suunnittelu aloitettiin elektroniikkasuunnittelusta. Ohjelmakehitys aloitettiin vasta elektroniikkakorttien valmistumisen jälkeen. Näin voitiin elektroniikan ja ohjelman toimivuutta testata sitä mukaan, kun ohjelman kirjoitus eteni. Testilaitteen toimintojen testausta ja ohjelman kirjoittamista tehtiin samaan aikaan, koska elektroniikan testausta varten tarvittiin erilaisia testiohjelman pätkiä.

Avainsanat: tuotannon testilaite, mikro-ohjain, C-kieli

Lahti University of Applied Sciences
Degree Programme in Information Technology

KOPSA, HANNU: Production tester for the R100 power supply

Bachelor´s Thesis in Computer Electronics, 36 pages, 50 appendixes

Spring 2009

ABSTRACT

The purpose of this thesis was to design and build a test device for a customized power supply, R100. Test device will be used in production at MEG Power Oy. The company designs and produces switched mode power supplies and other related products. R100 is a customized power supply which differs significantly from standard products. Therefore R100 cannot be tested by commonly used power supply test devices. Earlier, testing of R100 has been performed manually by using several different test devices. However, manual testing is very complicated and time-consuming. The decision was made to design and build a special test device for R100, which is easy to use and testing time is short.

The project started by investigating the function of power supply R100 and the manual testing method. The aim was to simplify the complicated test phases in order to be able to use simple circuitry. The test device was designed by using both analog and digital electronics. Controlling of all functions of the test device was handled by a PIC16F887 micro controller. Programming was done by the C programming language. User interface was created using an LCD display and a panel with three buttons.

The first step was to design hardware for the test device. Programming was started after the printed circuit boards had been made. Functionality testing of the designed test device was done step by step during software development. Testing software was needed to test the functionality of the electronics of the test device.

Keywords: test device, micro controller, C programming language

SISÄLLYS

1 JOHDANTO

Tämän opinnäytetyön aiheena on asiakaskohtaisen hakkuriteholähteen testilaitteen suunnittelu ja rakentaminen. Työ on tehty MEG Power Oy -nimiselle yritykselle, joka on hakkuriteholähteitä suunnitteleva ja valmistava yritys. Yritys valmistaa pääasiassa yleiskäyttöisiä standardituotteita, mutta myös asiakaskohtaisia erikoismalleja.

Kaikki yrityksen valmistamat laitteet testataan ennen toimitusta. Standardituotteiden testaaminen on helppoa, koska eri laitteiden liitännät ovat samankaltaisia, eivätkä testiohjelmat poikkea paljon toisistaan. Asiakaskohtaisien tuotteiden testaus on usein vaikeampaa, koska ne sisältävät usein asiakkaan tarvitsemia erityisominaisuuksia.

Tässä työssä käsiteltävänä oleva testilaite on tarkoitettu teholähteelle, joka on osa suurempaa laitekokonaisuutta. Teholähteessä on useita lähtöliitäntöjä, joita ohjataan useilla eri tuloliitännöillä. Työhön kuuluu kolme suurta osa-aluetta, jotka ovat elektroniikkasuunnittelu, ohjelmointi ja mekaniikkasuunnittelu.

2 TESTILAITTEEN OMINAISUUDET

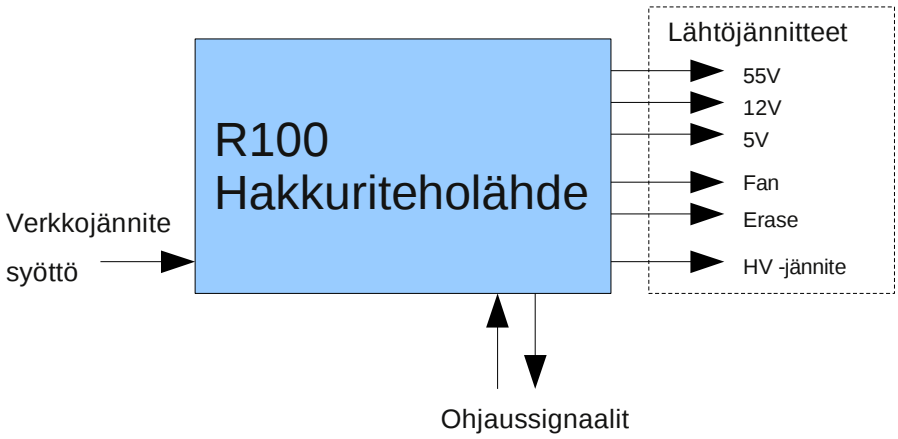
Ennen testilaitteen suunnittelun aloittamista oli tarpeellista tutustua testattavan laitteen toimintaan ja kaikkiin ominaisuuksiin. Tämä tehtiin tarkastelemalla R100-laitteen piirikaaviota ja kokeilemalla laitteen toimintaa laboratorion pöydällä. Apuna oli myös aikaisemmin käytetyn manuaalisen testauksen työohje, jota käytettiin pohjana testausmenetelmien suunnittelussa.

2.1 Testattava laite

Testilaite suunniteltiin MEG Power Oy:n tuotanto-osaston käyttöön. Testilaitteella piti voida testata R100-tyyppisten asiakaskohtaisten erikoisteholähteiden toimintaa. Koska R-100 on asiakaskohtainen tuote, tässä työssä ei voida paljastaa laitteen kaikkia yksityiskohtia ja toimintoja. R-100 teholähteen käyttökohdetta ei myöskään voida kertoa.

2.1.1 Testattavan laitteen ominaisuudet

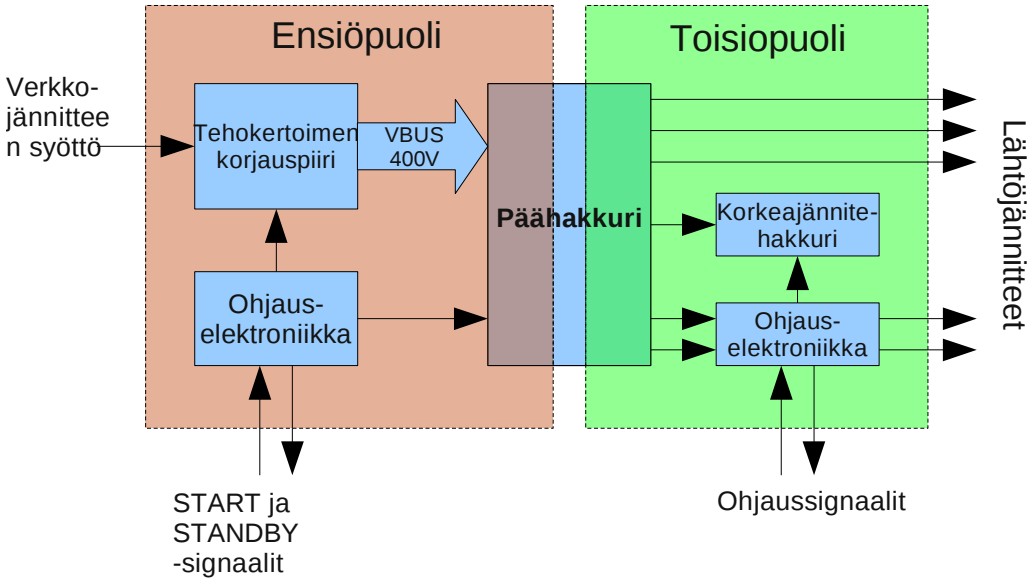
R-100 on ensisijaisesti hakkuriteholähde, joka muuntaa verkkojännitteestä useita DC -ulostulojännitteitä. Laite sisältää myös muuta toiminnallisuutta, kuten jänniteulostulojen ohjauselektroniikkaa. Laite on toteutettu lähes kokonaan analogiatekniikalla joitakin digitaalisia ohjaussignaaleja lukuun ottamatta. Kuviossa 1 on havainnollistettu testattavan laitteen liitäntöjä. Kaikki liitännät on toteutettu käyttämällä jousivoimaliittimiä. Testattavan laitteen metallikuoressa on aukot, joista piirilevylle sijoitettuihin liittimiin voidaan kytkeä johtosarja.



KUVIO 1. R100-hakkuriteholähteen liitännät.

Testattavan laitteen lähtöliitännät on erotettu galvaanisesti tuloliitännöistä. Laitteessa on myös tehokertoimen korjaustoiminto, jota kutsutaan lyhenteellä PFC (Power Factor Correction). Tehokertoimen korjauspiiri tekee tasasuunnatusta syöttöjännitteestä 400V:n tasajännitteen hakkuriteholähteen välijännitepiiriin. Tässä työssä tätä jännitettä kutsutaan VBUS -jännitteeksi (kuvio 2).

Testattavan laitteen päähakkuri tekee VBUS -jännitteestä useita galvaanisesti erotettuja lähtöjännitteitä. Kuviossa 2 on on esitetty lohkokaavio, joka havainnollistaa eri signaalien sijoittumisen ensiö- ja toisiopuolelle.



KUVIO 2. Lohkokaavio testattavan laitteen signaalien sijoittumisesta.

Päähakkuri siirtää sähköenergian ensiöpuolelta toisiopuolelle muuntajan välityksellä. Muuntaja erottaa ensiö- ja toisiojännitteet toisistaan ja muuntaa toisiopuolen jännitetasot sopivaksi. Toisiopuolen lähtöjännitteillä on pääsääntöisesti yhteinen maataso. Tämä helpottaa mittauksien tekemistä, koska melkein kaikkien lähtöliittimien maasignaalit voidaan kytkeä testilaitteessa yhteen. Poikkeuksena tähän on kuitenkin korkeajännitelähdön liitäntä, jonka mittauskytkentä täytyy olla erotettu muista lähtöliitännöistä.

2.2 Testattavat toiminnot ja mittaustavat

R100-teholähteestä testattavat toiminnot ja mittaustavat valittiin siten, että voidaan olla varmoja teholähteen oikeasta toiminnasta sen lopullisessa toimintaympäristössä. Tarkempi kuvaus kaikista testeistä löytyy testiohjelman C -kielisestä lähdekoodista (liite 12), mutta seuraavassa luetelmassa on tiivistelmä testattavalle laitteelle tehtävistä testeistä:

- apujännitelähteen käynnistyminen ja standby -ledi
- lämpösuojauksen ja start -signaalin toiminta
- VBUS-jännitteen säätäminen
- päähakkurin käynnistymisjännite
- lähtöjännitteiden tarkistus
- virtarajojen säätäminen
- lähtöjännitteiden tarkistus kuormituksen kanssa
- ohjaussignaalien toiminta
- 5V lähtöjännitteen hienosäätö
- korkeajännitelähdön toiminta ja hienosäätö
- korkeajännitteen tarkistussignaalin toiminta
- 5V lähtöjännitteen vakaus kuormituksen muuttuessa nopeasti

2.3 Käyttöliittymä ja laitteen käytettävyys

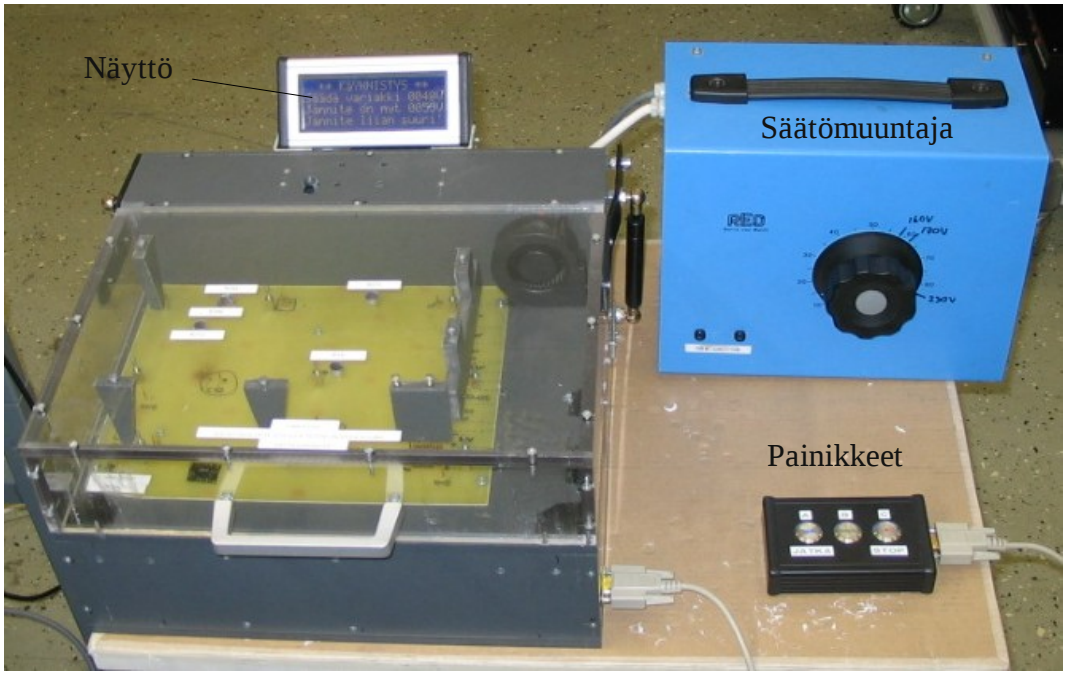
Testilaitteesta haluttiin tehdä niin helppokäyttöinen ja käyttäjäystävällinen, että kuka tahansa tuotannon työntekijä voi käyttää sitä ilman pitkäkestoista opettelua ja jatkuvaa käyttöohjeiden lukemista. Lisäksi testilaite pyrittiin suunnittelemaan niin, että testaaja ei pysty helposti vaikuttamaan testituloksiin.

Testilaitteen käyttöliittymä toteutettiin kolmella painonapilla ja 4x20 merkin LCD -näytöllä. Näyttö on kiinnitetty kiinteästi laitteen kotelon päälle. Näyttö koteloitiin pieneen alumiinikoteloon, ja näytön eteen on asennettiin kirkas polykarbonaattilevy suojaamaan näyttöä kolhuilta. Näytölle tulostetaan testiohjelman aikana ohjeita siitä, mitä testaajan pitää seuraavaksi tehdä. Kun testin aikana käyttäjän täytyy säätää testattavan laitteen trimmereitä, näytölle voidaan tulostaa ohje siitä, mihin suuntaan trimmeriä täytyy säätää (kuvio 3). Kun säätö on kohdallaan, se kuitataan painamalla testilaitteen sinistä painonappia. Punaisella painonapilla voi keskeyttää testin, jos se on jostain syystä tarpeellista. Painonapit sijoitettiin erilliseen kaukosäätimeen, joka on liitetty johdolla testilaitteeseen (kuvio 4). Painonapeiksi valittiin vandalismin kestävät mallit, koska tuotannossa voi käyttö olla välillä kovaakin.

KUVIO 3. Testilaitteeseen valittu LCD -näyttö

KUVIO 4. Testilaitteen painonapit

Testilaitteesta pyrittiin tekemään mahdollisimman kompakti laite, mikä tarkoittaa sitä, ettei laitteeseen tarvitse liittää turhan paljon ulkoisia lisälaitteita. Koska testiohjelma vaatii tulojännitteen muuttamista, testilaitteeseen täytyy kuitenkin kytkeä ulkoinen säätömuuntaja, jolla testilaitteelle tulevaa verkkojännitettä voidaan säätää. Kuviossa 5 näkyy koko testaamiseen tarvittava laitteisto.



KUVIO 5. Testilaite käyttövalmiina

2.4 Turvallisuus

Yksi tärkeä laitteen suunnittelussa huomioon otettava seikka on testauksen turvallisuus testaajan kannalta. R100-laitteessa on testin aikana hengenvaarallisia jännitteitä, kuten laitteelle tuleva syöttöjännite, joka on maksimissaan 230V. Lisäksi laitteen piirilevyllä on toiminnan aikana muun muassa 400V:n VBUS -jännite ja korkeajännitelähtö, jonka jännite voi olla jopa 1250V.

Suunnittelussa täytyi ottaa huomioon, ettei näistä jännitteistä saa koitua testaajalle missään tilanteessa mitään vaaraa. Sähköturvallisuudesta omalta osaltaan huolehtii polykarbonaatista valmistettu suojakansi ja kannen magneettiset turvakytkimet. Kansi päätettiin valmistaa 10mm vahvuisesta polykarbonaattilevystä, jotta se olisi tukeva ja kestäisi hyvin. Jos kansi avataan kesken testin, magneettiset turvakytkimet ohjaavat suojausreleitä, jotka katkaisevat testattavan laitteen sähkönsyötön. Testattavan laitteen ensiöpuolella olevan suurikapasitanssisen elektrolyyttikondensaattorin vaarallinen 400V:n jännite päätettiin purkaa automaattisesti ja nopeasti aina, kun kansi avataan.

Testilaitteen kansi päätettiin lukita kiinni kahdella sähkömagneetilla. Lukitusmagneettien ohjaus toteutettiin mikro-ohjaimella, joka avaa kannen automaattisesti sen jälkeen, kun testi on päättynyt tai se on oikealla tavalla keskeytetty. Kansi pysyy testin aikana kiinni sähkömagneettien pitovoimalla. Vaikka testaaja vääntäisi kannen väkisin auki testin aikana, turvakytkimet avautuvat ja turvapiiri katkaisee jännitteen syötön.

3 TESTILAITTEEN SUUNNITTELU

Testilaitteen suunnitteluun kuului kolme osa-aluetta: elektroniikkasuunnittelu, mekaniikkasuunnittelu ja ohjelmointi. Alun perin ajateltiin, että suurin osa suunnitteluajasta kuluu elektroniikkasuunnitteluun ja muut osa-alueet on nopeasti hoidettu. Työn edetessä ohjelmointi osoittautui kuitenkin vähintään yhtä suuritöiseksi kuin elektroniikkasuunnittelukin. Myös mekaniikkasuunnittelu osoittautui odotettua suuremmaksi haasteeksi.

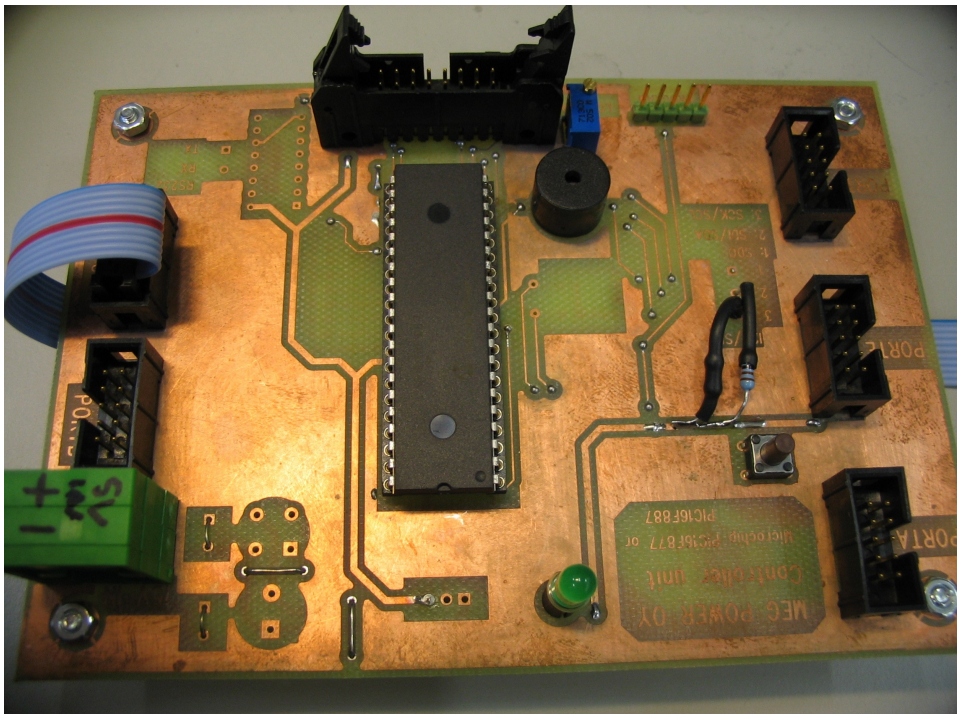
3.1 Elektroniikkasuunnittelu

Elektroniikkasuunnittelussa käytettiin PADS -suunnitteluohjelmistoa. Testilaitteen kaikki piirikaaviot suunniteltiin PADS Logic -ohjelmalla. Kun piirikaaviot oli tarkistettu ja viimeistelty, niistä suunniteltiin piirilevyt PADS Layout -ohjelmalla. Piirilevyt valmistettiin MEG Power Oy:n omalla piirilevyjyrsimellä. Ne myös kalustettiin itse juottamalla kaikki komponentit käsin paikoilleen.

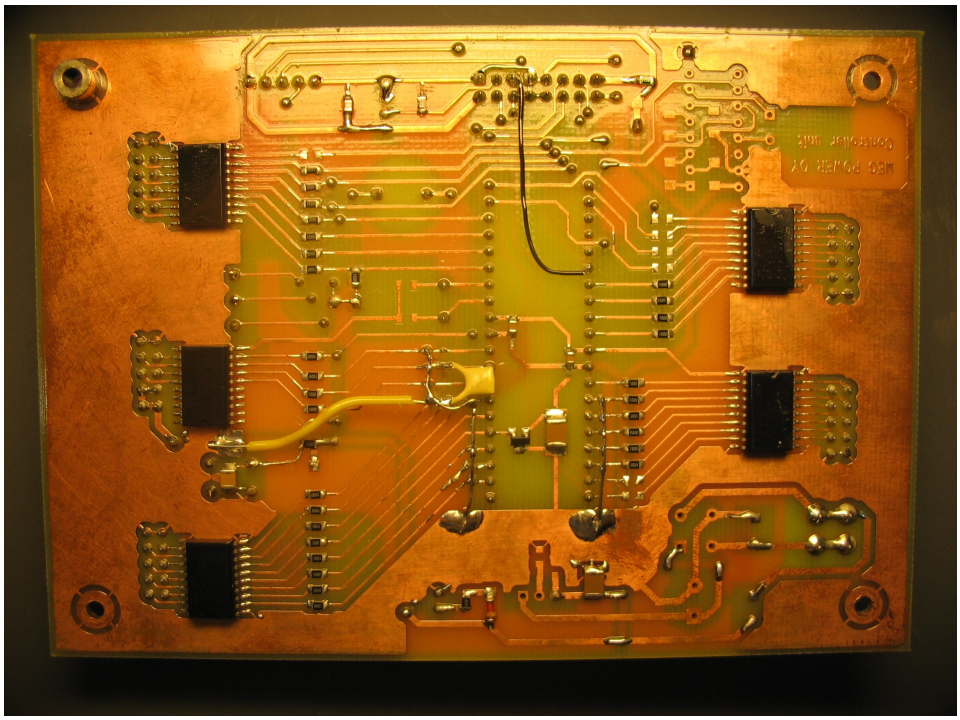
Testilaitteeseen suunniteltiin kaksi piirikorttia, joista toinen on yleiskäyttöinen ohjainkortti. Ohjainkorttia voidaan tarvittaessa käyttää muihinkin sovelluksiin. Toinen kortti on mittauskortti, joka suunniteltiin vain tähän testilaitteeseen, eikä siinä ole mitään ylimääräisiä ominaisuuksia.

3.1.1 Yleiskäyttöinen ohjainkortti

Koska testilaitteen toiminta perustuu mikro-ohjaimeen, päätettiin mikro-ohjainta varten suunnitella ohjainkortti, jota voi tarvittaessa helposti käyttää muissakin sovelluksissa. Yleiskäyttöisyyden vuoksi kaikki mikro-ohjaimen I/O -liitännät kytkettiin ohjainkortin liittimiin. Näistä liittimistä voidaan ottaa käyttöön kaikki tarvittavat mikro-ohjaimen toiminnot. Kuvioissa 6 ja 7 on valokuvat ohjainkortista. Ohjainkortin piirikaavio on liitteessä 1.



KUVIO 6. Ohjainkortti yläpuolelta.



KUVIO 7. Ohjainkortti alapuolelta.

Ohjainkortin tärkein yksittäinen komponentti on Microchip:n valmistama mikro-ohjain PIC16F887 (liite 1). PIC16F887 valittiin sen monipuolisten ominaisuuksien vuoksi. MEG Power Oy:llä oli myös valmiina kehitystyökaluja PIC-mikro-ohjainperheen ohjelmointia varten. Ohjelmakoodin siirtämiseen mikro-ohjaimen sisälle käytettiin sarjamuotoista In-Circuit Serial Programming™ -liitäntää. Tätä liitäntää varten ohjainkortille sijoitettiin oma liitin (liite 1). Ohjelmointilaitteena käytettiin ICD2 -tyyppistä laitetta. (PIC16F887 Datasheet 2008.)

Ohjainkortille voidaan asentaa joitakin muitakin PIC-perheen mikro-ohjaintyyppejä. Esimerkiksi monet PIC16F887:n rinnakkaismallit ovat sopivia, muttta yhteensopivuus täytyy kuitenkin aina tarkistaa tapauskohtaisesti. PIC16F887-mikro-ohjaimessa on sisäinen kello-oskillaattori, minkä vuoksi oskillaattorikide Y1 ja kondensaattorit C4 ja C5 ovat turhia tätä mikro-ohjainta käytettäessä. Niille on kuitenkin hyvä jättää paikat, koska esimerkiksi PIC16F778 ei sisällä oskillaattoria. (PIC16F887 Datasheet 2008.)

PIC16F887 mikro-ohjaimessa on muun muassa seuraavat tässä työssä tarvittavat ominaisuudet (PIC16F887 Datasheet 2008):

- 2,0V – 5,5V käyttöjännitealue
- Sisäinen kello-oskillaattori (8MHz)
- 35 konfiguroitavaa I/O linjaa
- Analoginen komparaattorimoduuli
- 14-kanavainen A/D-muunnin 10-bittisellä resoluutiolla
- kaksi 8-bittistä ajastinmoduulia
- 16-bittinen ajastinmoduuli
- kaksi PWM-moduulia
- USART-sarjaportti (RS232:lle optio ohjainkortilla)
- In-Circuit Serial Programming™ -ohjelmointiliitäntä
- I2C-sarjaväylä
- Ulkoinen keskeytystulo
- 40-pinninen PDIP-kotelo.

Ohjainkortin käyttöjännitteen syöttöön suunniteltiin kaksi vaihtoehtoa; Yleiskäyttöinen vaihtoehto otetaan käyttöön asentamalla kaikki piirikaavion vasemmassa ylänurkassa näkyvät käyttöjännitepiirin komponentit (liite 1). Yleiskäyt-

töinen käyttöjännitesyöttö mahdollistaa ohjainkortin käyttämisen noin 10V-30V tasa- tai vaihtojännitteellä. Käyttöjännitekytkennän jänniteregulaattori U2 (LM317) tekee mikro-ohjaimelle tasaisen viiden voltin käyttöjännitteen. Jännitetaso on asetettu vastuksilla R1 ja R2 (liite 1). (LM317 Datasheet 2008.)

Koska testilaitteen mittauskortille päätettiin suunnitella vakavoidut jännitesyötöt kaikille tarvittaville käyttöjännitteelle, tässä testilaitteessa otettiin ohjainkortilla käyttöön yksinkertaisempi kytkentä, joka vaatii valmiiksi vakavoidun ja suodatetun viiden voltin jännitesyötön. Tässä kytkentätavassa voidaan piirikortilta jättää pois piirikaavion vasemmassa ylänurkassa näkyvät komponentit D1, C2, C10, U2, R1 ja R2 (liite 1). Liittimiin JP1 ja JP2 täytyy tässä tapauksessa kytkeä oikosulkupalat (liite 1).

Ohjainkortille sijoitettiin HD44780 yhteensopivaa LCD -näyttöä varten oma liitin, johon kytkettiin kaikki näytön ohjaukseen tarvittavat I/O -liitännät. Lisäksi näytön liittimeen kytkettiin käyttöjännitesyöttö ja taustavalon jännitesyöttö. Liittimen pinnijärjestys suunniteltiin niin, että siihen on helppo kytkeä LCD -näyttö yhdellä lattakaapelilla (liite 1). (Specification for BTHQ42008VSS-SMN-LEDwhite, 7.)

Äänimerkkien antamista varten ohjainkortille sijoitettiin summeri H1. Summeri on kytketty mikro-ohjaimen pinniin RC0, jota voidaan käyttää ajastinmoduuli Timer 1:n lähtönä. Ajastinmoduulilla voidaan tuottaa eritaajuisia signaaleja summerille. Tässä testilaitteessa erilaisille äänitaajuuksille ei kuitenkaan ollut tarvetta, joten summerin tyypiksi valittiin vakiotaajuudella toimiva malli. Tämän tyyppinen summeri ei tarvitse valmista taajuussignaalia, vaan summerille syötettiin pelkkää tasajännitettä. Tasajänniteohjauksen vuoksi kondensaattori C12 jouduttiin poistamaan piirikortilta ja korvaamaan hyppylangalla (liite 1).

Mikro-ohjainkortille lisättiin A/D -muunninta varten jännitereferenssi U9 (liite 1). REF3140-tyyppinen referenssikomponentti tuottaa erittäin tarkan 4,096 voltin referenssijännitteen. PIC16F887-mikro-ohjaimessa on myös ominaisuuksia, joilla olisi ollut mahdollista toteuttaa sisäinenkin jännitereferenssi, mutta sen tarkkuus ei olisi ollut riittävä. Jännitereferenssiksi valittiin erittäin tarkka komponentti, jotta A/D -muuntimen tulokset olisivat mahdollisimman luotettavia. REF3140:n

datalehdessä luvataan referenssijännitteen tarkkuudeksi 0,2% ja lämpötilariippuvuudeksi korkeintaan 15ppm/°C (REF3140 Datasheet 2006).

Ohjainkortilla on paikka RS232 -sarjaväylän tasonmuunnospiirille U8 (liite 1). Tätä ominaisuutta ei tässä työssä tarvittu, joten piiriä U8 ei asennettu piirikortille. RS232 on kuitenkin hyvin yleisesti käytetty sarjaväylä mikro-ohjainsovelluksissa, ja tasonmuuntopiirin avulla se on helppo ottaa tarvittaessa käyttöön. Myös I2C ja SPI -sarjaväyliä varten lisättiin oma liitin (liite 1). Työssä käytetty mikro-ohjain tukee kaikkia näitä sarjaväyliä, mutta I2C ja SPI -väyliä ei voi ottaa käyttöön samaan aikaan. Tässä työssä on käytetty I2C -väylää ohjaamaan I/O -laajennuspiirejä. (PIC16F887 Datasheet 2008.)

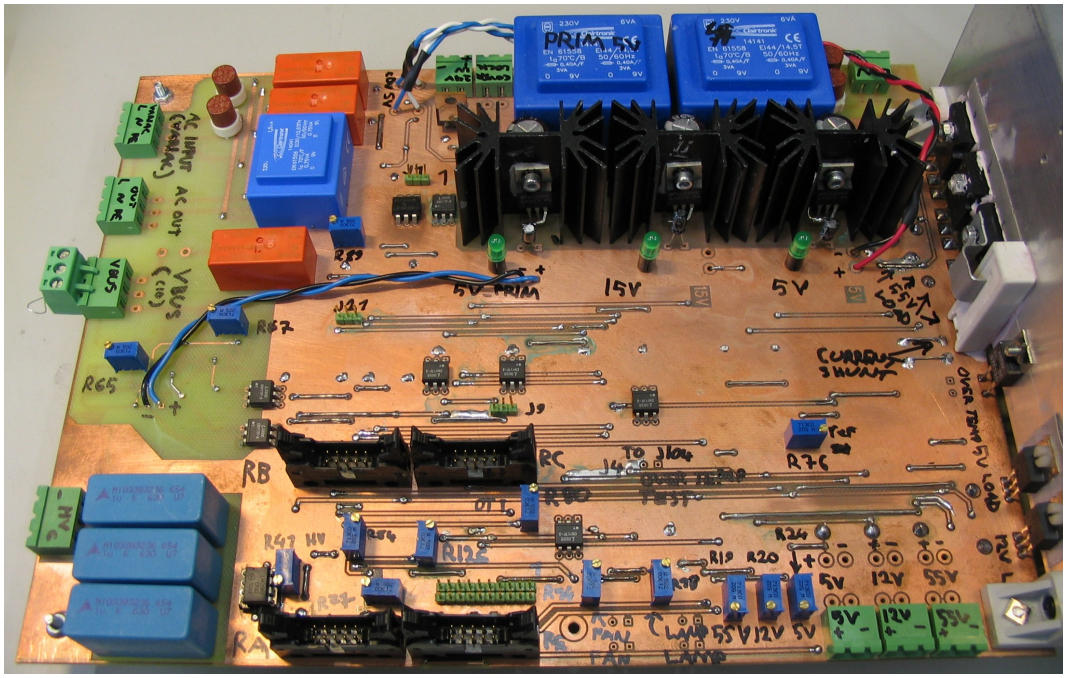
Mikropiirit U3 – U7 (liite 1) suojaavat mikro-ohjaimen I/O -linjoja ylijännitepiikeiltä. Nämä ITA6V5B3 piirit eivät ole laitteen toiminnan kannalta välttämättömiä, mutta ne estävät laitteen vikaantumista esimerkiksi staattisen sähkön purkautuessa laitteeseen. I/O -linjoihin kytketyt sadan ohmin vastukset suojaavat mikro-ohjainta oikosululta. Nämä suojauskytkennät on lisätty ohjainkorttiin mahdollista tulevaa käyttöä ajatellen. (ITA6V5B3 Datasheet 2007.)

3.1.2 Mittauskortti

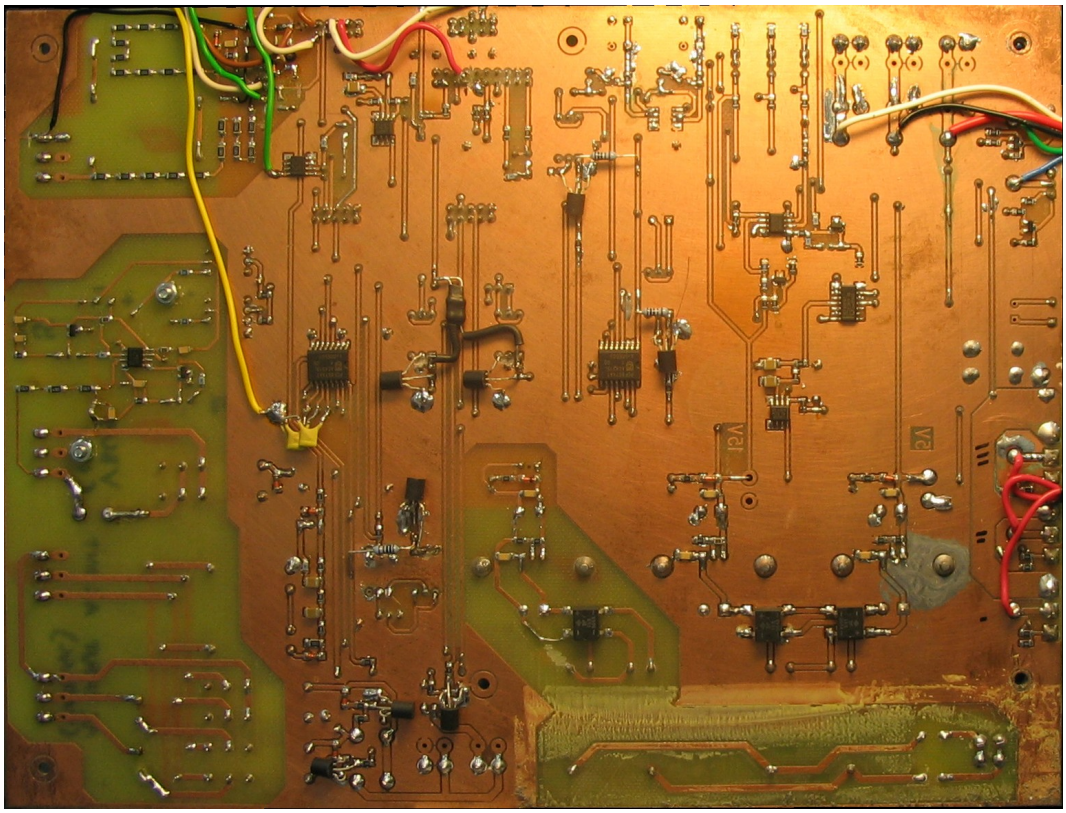
Suurin osa testilaitteen elektroniikasta sijoitettiin mittauskortille. Mittauskortista tuli melko suurikokoinen (kuviot 8 ja 9). Suuri koko johtuu suuresta määrästä liittimiä, jotka haluttiin sijoittaa kortin reuna-alueille. Suurijännitteiset kytkennät erotettiin muusta elektroniikasta turvallisuussyistä omiin osastoihinsa. Näiden kytkentöjen turvavälit kasvattivat myös kortin kokoa. Ohjainkortti ja mittauskortti kytkettiin toisiinsa käyttämällä lattakaapeleita ja niihin puristettavia liittimiä.

Mittauskortti sisältää seuraavia ominaisuuksia:

- käyttöjännitelähde
- AC -syöttöjännitteen mittaus-, kytkentä- ja turvapiirit
- VBUS -jännitteen purkupiiri
- kannen lukitusmagneettien ohjauskytkentä
- R100 -laitteen ohjaussignaalien mittaus- ja ohjauskytkennät
- I/O -signaalien laajennuspiirit
- jännitemittauksien skaalauskytkennät
- 5V jännitelinjan jännitekuoppien tunnistuspiiri
- korkeajännitelähdön tasajännitteen mittauspiiri
- korkeajännitelähdön häiriötason mittauspiiri
- korkeajännitelähdön mittauspiirien erotuspiiri
- PWM/DC -muunnospiiri
- jännitereferenssit
- PWM -signaalilla säädettävä vakiovirtakuormituspiiri
- vastuskuormien kytkentäpiirit
- VBUS -jännitteen ylä- ja alarajan tunnistuspiiri



KUVIO 8. Mittauskortti yläpuolelta.

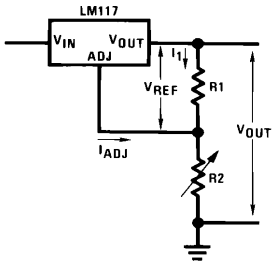


KUVIO 9. Mittauskortti alapuolelta

Käyttöjännitelähde

Mittauskortin elektroniikka tarvitsee toimiakseen useita käyttöjännitteitä. Tämän vuoksi mittauskortille suunniteltiin käyttöjännitelähde, joka toteutettiin käyttämällä yksinkertaista analogiatekniikkaa, jossa käytettiin verkkomuuntajia ja lineaariregulaattoreita. Verkkomuuntajiksi valittiin piirilevylle asennettavat mallit, joiden toisiopuolen vaihtojännite tasasuunnattiin diodisillalla ja suodatettiin kondensaattoreilla. Jänniteregulaattorit säätävät lähtöjännitteen asetettuun jännitearvoon. (liite 2, LM317 Datasheet 2008.)

Ohjainkortin ja mittauskortin digitaalipiirit toimivat viiden voltin jännitteellä, mutta osa analogiakytkennöistä tarvitsee kuitenkin toimiakseen korkeamman 15 voltin jännitteen. Regulaattoreiden lähtöjännitetaso asetettiin vastuksien jännitejaolla kuviossa 10 esitetyn esimerkkikytkennän mukaan. Tämän kytkennän lähtöjännite voidaan mitoittaa kaavan 1 mukaan.



KUVIO 10. LM317 jänniteregulaattorin kytkentäesimerkki (LM317 Datasheet 2008).

$$V_{OUT} = 1{,}25\,V\left(1+\frac{R2}{R1}\right)$$

KAAVA 1. Regulaattorin lähtöjännitteen laskeminen

R100 -laitteen VBUS -jännite sijaitsee laitteen ensiöpuolella (kuvio 2). Tämän vuoksi kyseisen jännitteen mittauskytkentä täytyy olla sähköisesti erotettu muista kytkennöistä. VBUS -mittauskytkennälle tehtiin oma käyttöjännite, jonka suuruus on viisi volttia (Liite 2). Jo suunnitteluvaiheessa todettiin, että tälle jännitteelle omistettu muuntaja ja suodatuskytkentä oli tehonkestoltaan huomattavasti ylimitoitettu, mutta näin voitiin kytkennöissä käyttää samoja komponentteja. Toisaalta ylimitoituksesta ei ollut mitään haittaa.

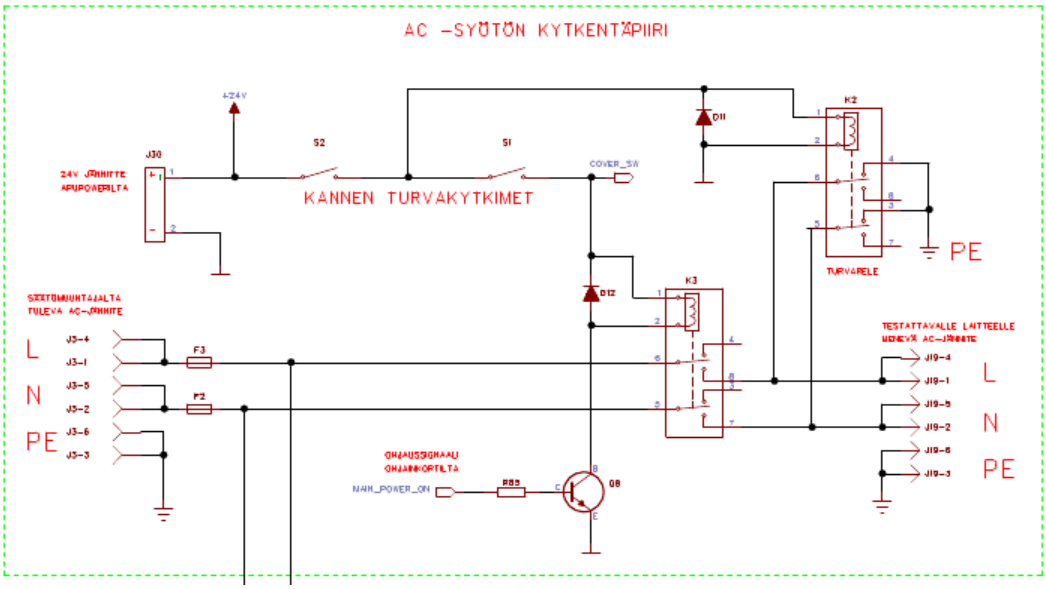
Yksi vaihtoehto käyttöjännitteiden luomiseen olisi ollut hakkuritekniikan käyttäminen, mutta se olisi tuottanut lisää työtä suunnitteluun ja testaamiseen. Hakkuritekniikkaa käyttämällä käyttöjännitelähteen koko olisi pienentynyt huomattavasti, mutta toisaalta se olisi ehkä voinut aiheuttaa häiriöitä mittauksiin. Hakkuritekniikan oletettiin myös olevan monimutkaisuudestaan johtuen vähemmän luotettavaa ja vikatilanteissa vaikeammin korjattavaa tekniikkaa kuin työhön valittu tekniikka, joten valinta oli kaikenkaikkiaan hyvä.

AC -syöttöjännitteen kytkentä- ja mittauspiiri

Testattava laite tarvitsee testauksen aikana erisuuruisia tulojännitteitä, jotka syötetään piirilevylle neulapedin kautta. Jännite on vaihtojännitettä, jonka suuruus vaihtelee testin aikana välillä 0-230V. Jännitteen suuruus asetetaan ulkoisella säätömuuntajalla (variakilla), jota säädetään käsikäyttöisellä säätönupilla. Jotta neulapedin neuloille ei tulisi vaarallista jännitettä silloin, kun suojakansi on auki, jännitteen syöttöä varten suunniteltiin releillä ja turvakytkimillä toteutettu kytkentä- ja turvapiiri (Kuvio 11, Liite 2).

Jotta neulapedin neuloille voi tulla jännitettä, täytyy seuraavien ehtojen täyttyä:

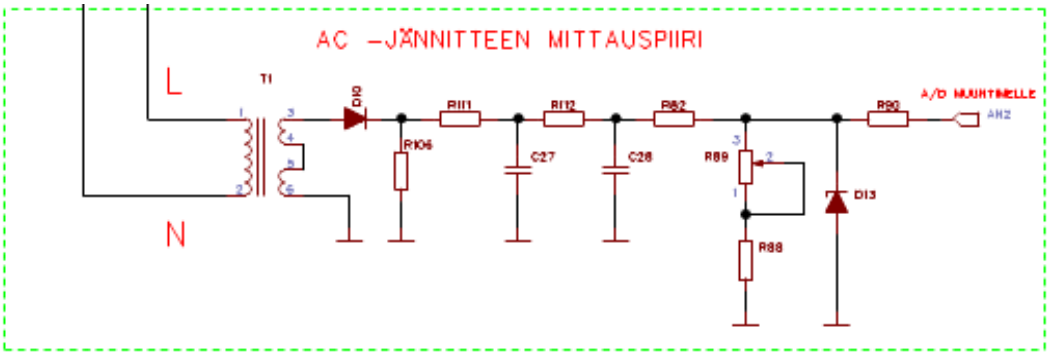
1. Turvakytkin S1 on suljettu, jolloin rele K2 poistaa oikosulun L:n N:n ja PE:n väliltä (Kuvio 11).
2. Turvakytkin S2 on suljettu, jolloin rele K3 saa käämilleen jännitteen (Kuvio 11).
3. Mikro-ohjain on aktivoinut ohjaussignaalin MAIN_POWER_ON, jolloin transistori Q8 sallii releen K3 toimimisen (Kuvio 11).



KUVIO 11. AC-syöttöjännitteen kytkentä ja turvapiiri

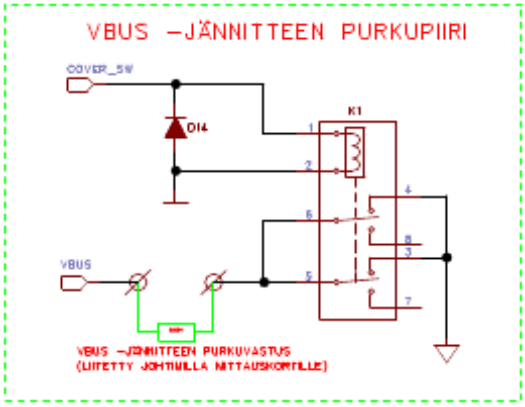
Testauksen aikana kannen ollessa suljettuna mikro-ohjain ohjaa jännitteen päälle ja pois kytkemistä testiohjelman mukaisesti. Jos kansi kuitenkin avautuisi testin aikana, turvakytkimet estäisivät jännitteen syötön. Koska syötettävää jännitettä täytyy säätää testauksen aikana, päätettiin testilaitteeseen lisätä mittauskytkentä,

joka mahdollistaa jännitteen mittaamisen mikro-ohjaimen A/D -muuntimella. Mittauskytkentä näkyy kuviossa 12 ja liitteessä 2. Mitattava jännite erotetaan ja muunnetaan pienemmäksi muuntajalla T1. Diodi D10 tekee muuntajalta tulevalle vaihtojännitteelle puoliaaltotasasuuntauksen, minkä jälkeen kaksi peräkkäistä RC -piiriä muodostaa tästä jännitteestä keskiarvon. Lopuksi keskiarvojännite vielä skaalataan A/D -muuntimelle sopivaan tasoon vastuksien jännitejaolla, jossa esiintyvä trimmeri R89 lisättiin kytkentään kalibrointia varten (Kuvio 11). Koska A/D -muuntimelle menevä mittaustieto on muuntajalla ja jännitteenjaolla skaalattu hyvin pieneksi jännitteeksi, täytyy lopullinen tulos korjata ohjelmallisesti mikro-ohjaimessa. Jännitteen skaalaamisessa käytettiin suhdetta 1/64.



KUVIO 12. AC -syöttöjännitteen mittauspiiri

R100-laitteen ensiöpuolella on PFC-korjaimen jälkeen VBUS-jännitteeksi kutsuttu 400V:n suuruinen tasajännite, joka varastoituu suureen elektrolyyttikondensaattoriin. R100-laitteen toiminnan jälkeen voi elektrolyyttikondensaattoriin jäädä vaarallisen suuri jännite. Turvallisuuden parantamiseksi tälle jännitteelle päätettiin suunnitella purkupiiri, joka tyhjentää elektrolyyttikondensaattorin varauksen erittäin nopeasti, aina kun testilaitteen kansi avataan (Kuvio 12).



KUVIO 12. VBUS -jännitteen purkupiiri

Jännitteen purkuun käytettiin suurta lankavastusta, joka kestää erittäin suuria hetkellisiä tehopulsseja. Purkuvastuksena käytetyn lankavastuksen resistanssi oli 100Ω. Ohmin lain mukaan 400V:n jännitteen purkaminen tällaisella resistanssilla aiheuttaa vastukseen neljän ampeerin hetkellisen virran ja 1600W:n hetkellisen tehon (Kaava 2). Lankavastukseksi valittiin 50W:n jatkuvaa tehoa kestävä tyyppi. Purkamiseen kuluva aika on kuitenkin niin lyhyt hetki, että lankavastus kestää tällaisen tehopulssin hyvin. R100:n suurijännitteisen elektrolyyttikondensaattorin kapasitanssi on 330µF. Tämä elektrolyyttikondensaattori ja purkuvastus muodostavat RC -piirin, jonka aikavakioksi τ saadaan kaavan 3 mukaan 33ms. Tämän ajan kuluttua kondensaattorin jännite on siis laskeutunut noin 37%:iin eli 148V:iin. 165ms:n kuluttua (5τ) kondensaattorin jännite on käytännöllisesti katsoen täysin purkautunut (Volotinen 1998, 119).

$$I = \frac{U}{R}$$

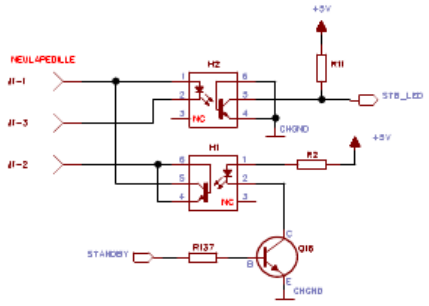
KAAVA 2. Ohmin laki

$$\tau = R \cdot C \qquad \tau = aikavakio$$

KAAVA 3. RC -aikavakion laskeminen

R100 -laitteen ohjaussignaalien testauskytkennät

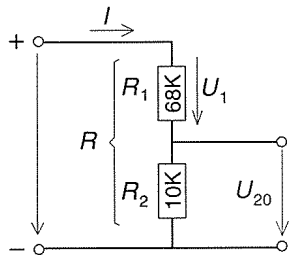
R100-laitteessa on useita digitaalisia ohjaussignaaleja, joista osa on lähtöjä ja osa tuloja. Ohjaussignaalien käsittely päätettiin toteuttaa optoerottimien välityksellä. Optoerottimet eristävät testattavan laitteen sähköisesti testilaitteesta, suojaten testilaitteen elektroniikkaa mahdollisissa vikatilantissa. Kuviossa 13 on esimerkki ohjaussignaalien käsittelystä optoerottimien välityksellä. Liittimeen J1 kytketään kaksi ohjaussignaalia, joista toinen on tulo ja toinen lähtö. Optoerotinta H2 käytetään testaamaan LED -merkkivalon ohjaussignaalia. (Kuvio 13.)



KUVIO 13. Esimerkki ohjaussignaalien lukemisesta ja ohjaamisesta optoerottimen välityksellä

Mitattavien jännitteiden skaalaaminen

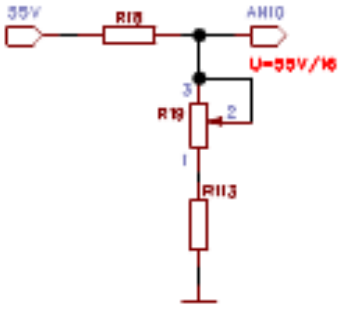
Jännitteiden mittaamiseen käytettiin PIC16F887-mikro-ohjaimen sisäistä A/D -muunninta, jonka referenssijännitetuloon syötettiin 4,096V:n jännite. Näin ollen A/D-muuntimen mittausalue on 0-4,096V. Jotta A/D muuntimella voidaan mitata myös tätä suurempia jännitteitä, käytettiin mittauskortilla skaalauskytkentöjä jännitteiden muuttamiseen mittausalueelle. Jännitteiden skaalauksessa käytettiin kuvion 14 tyyppistä jännitteenjakokytkentää. Jännitejaon tulevan ja lähtevän jännitteen suhde mitoitettiin kaavan 4 mukaan niin, että lähtöjännitetaso sopii A/D -muuntimelle. Suhdelukuina käytettiin luvun kaksi potensseja, esimerkiksi kuviossa 14 Uin / Uout = 2^4. Tällä oli tarkoitus helpottaa mikro-ohjaimessa tarvittavaa ohjelmallisesti toteutettua jännitearvon skaalaamista ylöspäin. Mikro-ohjaimessa jännitearvot oli helppo kertoa käyttämällä bittien siirto-operaatiota.



KUVIO 14. Esimerkki jännitteenjakokytkennästä (Ahoranta 1998, 99)

$$U_{20} = \frac{R_2}{R} \cdot U$$

KAAVA 4. Jännitteenjaon laskukaava (Ahoranta 1998, 99)



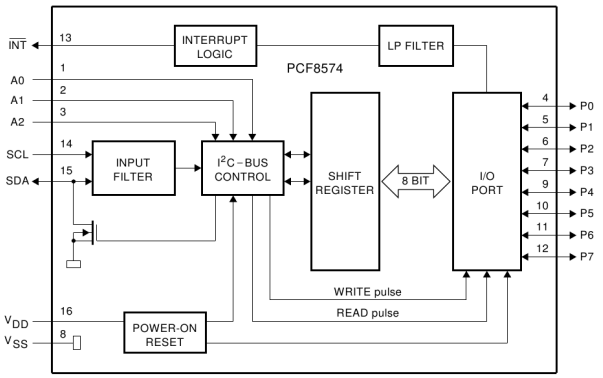
KUVIO 15. Mitattavien jännitteiden skaalaaminen

Tässä työssä jännitteenjakokytkentöihin lisättiin trimmerit kuvan 15 mukaisesti. Trimmereillä voidaan jännitejakoa hienosäätää, ja niiden avulla kaikki jännite-mittauskanavat on helppo kalibroida tarkasti kohdalleen toisistaan riippumatta.

I/O signaalien lisääminen sarjaväylän avulla

Vaikka PIC16F887-mikro-ohjaimessa on runsaasti I/O-linjoja, ne eivät riittäneet kaikkien testilaitteen toimintojen ohjaamiseen. I/O-linjojen lisäämiseksi mittaus-

kortille sijoitettiin kaksi PCF8574-tyyppistä mikropiiriä U8 ja U9. Näitä mikropiirejä ohjataan I2C-sarjaväylän kautta, ja niiden 8-bittiset väylät voivat toimia joko tuloina tai lähtöinä (PCF8574 Data Sheet 2002, 3). Kuviossa 16 on PCF8574:n lohkokaavio.



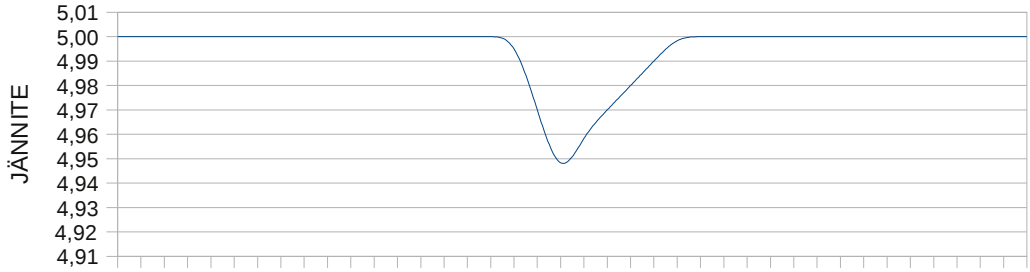
KUVIO 16. PCF8574 -mikropiirin lohkokaavio (PCF8574 Data Sheet 2002, 4)

Tässä työssä piirin U8 I/O-linjoja käytettiin tuloina ja U9:n linjoja lähtöinä. Piirien väylien suunta asetetaan I2C-väylän kautta luku- tai kirjoitusoperaation aikana. Piirejä voi kytkeä samaan I2C-sarjaväylään useita, mutta niille täytyy määrittää eri osoitteet. Osoitteet määritettiin asettelemalla piirien kolmebittisiin osoiteväyliin osoitetta vastaava binääriluku (PCF8574 Data Sheet, 10). U8:n osoitteeksi määritettiin 000 ja U9:n 001. Piirien kytkennät näkyvät mittauskortin piirikaaviossa sivulla 7 (Liite 2).

PIC16F887-mikro-ohjaimen ohjelmakoodiin kirjoitettiin ajuriohjelma PCF854-piirien ohjaamista varten. Ajuriohjelma sisältää mikro-ohjaimen I2C-moduulin käsittelytoiminnot ja PCF854-piirien ohjausfunktiot. PCF854-piirejä ohjataan lähettämällä halutun piirin osoitteen sisältävä käsky sarjaväylälle. Jos käsky on kirjoitusoperaatio, piirin I/O-linjat asettuvat käskyn sisältämän bittijonon mukaiseen tilaan (PCF8574 Data Sheet, 10). Jos käsky on lukuoperaatio, niin piiri lukee I/O-linjojen tilan ja palauttaa sen mukaisen bittijonon sarjamuotoisena mikro-ohjaimelle (PCF8574 Data Sheet 2002, 11).

5V jännitelinjan jännitekuoppien tunnistus

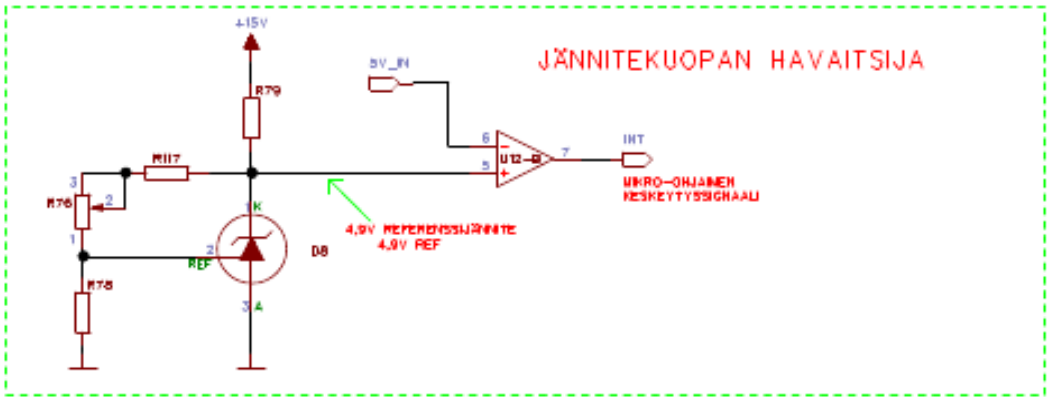
Yksi testilaitteessa tarvittava toiminto oli R100-laitteen viiden voltin jännitelähdössä mahdollisesti esiintyvien jännitekuoppien havaitseminen. Viiden voltin jännitelähdössä voi esiintyä normaalitoiminnassa kuvion 17 tyyppinen pieni jännitekuoppa silloin, kun laitteen kuormitus lisääntyy äkillisesti. Laitteelle on kuitenkin määritetty alhaisimmaksi viiden voltin jännitelähdön jännitetasoksi 4,90V, jota ei saa alittaa.



KUVIO 17. Tyypillinen jännitekuoppa R100 laitteen 5V -lähdössä äkillisen kuormituksen muutoksen aikana

Liian suuren jännitekuopan havaitsemista varten mittauskortille suunniteltiin kuvion 18 mukainen jännitteiden vertailukytkentä, joka kykeni testien mukaan havaitsemaan 50µs pituiset ja sitä pidemmät jännitekuopat, joiden taso alittaa referenssijännitteen. Referenssijännite säädettiin tarkasti 4,90 voltin tasoon trimmerillä R76, jota käytetään kalibrointiin.

Kytkennässä käytetään hyödyksi mikro-ohjaimen keskeytystuloa, jota jännitevertailijan lähtö ohjaa jännitekuopan havaitessaan. Testiohjelmaan on kirjoitettu toiminto, jossa R100-laitteen kuormitusta kasvatetaan äkillisesti sytyttämällä 50W tehoinen halogeenilamppu. Jos kuormituksen muutoksesta johtuva jännitekuoppa on liian suuri, mikro-ohjaimelle tulee keskeytyssignaali, joka käynnistää keskeytysohjelman. Keskeytysohjelma ilmoittaa testiohjelmalle edellä mainitulla tavalla havaitusta virheestä muuttamalla tietyn muistipaikan bitin arvon ykköseksi.



KUVIO 18. Jännitekuoppien havaitsija

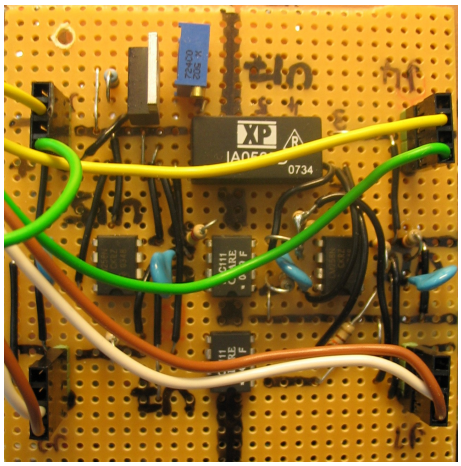
Korkeajännitelähdön mittaukset

R100-laitteessa on säädettävä korkeajännitelähtö (HV), jonka jännitealue on 400-1250V. Tämän jännitteen taso säädetään analogisella ohjaussignaalilla (HV_REF). Testiohjelman suorituksen aikana korkeajännitelähtö täytyy kalibroida siten, että ohjaussignaalin ollessa 5,0V, korkeajännitelähdön jännite on 1250V (R100 tuotannontestausohje, 8). 1250V:n suuruisen jännitteen mittaamisessa täytyi käyttää hyvin suurta skaalaussuhdetta 1/512. Skaalaaminen tehtiin jännitteenjakoperiaatteella, mutta koska kytkennässä käytettyjen pintaliitosvastuksien jännitteenkesto on vain 200V, jännitteenjaossa täytyi käyttää pitkää vastusketjua. Korkeajännitelähdön mittauskytkentä on mittauskortin piirikaavion sivulla 5 liitteessä 2.

Korkeajännitelähdön mittaukseen tehtiin kaksi kytkentää, joista toinen mittaa korkeajännitteen tasajännitetason ja toinen jännitteessä esiintyvän vaihtojännitetyyppisen häiriösignaalin voimakkuuden. R100-laitteen korkeajännitelähdön häiriösignaalin voimakkuudelle on määritelty raja-arvoksi 20mV, jota laite ei saa ylittää (R100 tuotannontestausohje, 8). Häiriösignaalin mittaamiseen suunniteltiin kytkentä, jossa vaihtojännitekomponentti erotetaan tasajännitteestä kondensaattoreilla. Kondensaattoreita kytkettiin kolme kappaletta sarjaan riittävän jännitteenkeston saavuttamiseksi. Kytkentään lisättiin kondensaattoreiden kanssa sarjaan kytketty vastusketju ja suojadiodit, jotka estävät mittauskytkennän rikkoontumisen nopean jännitteen muutoksen aikana. (Liite 2, 5.)

Alkuperäisen suunnitelmaan mukaan korkeajännitelähdön ja mittauskortin maatasot kytkettiin yhteen. Testausvaiheessa kuitenkin selvisi, että maatasojen yhteen kytkeminen aiheutti häiriöitä mittauksiin, eikä tällä menetelmällä saatu oikeita tuloksia. Mittausmenetelmää muutettiin siten, että korkeajännitelähdön mittauskytkentä ja mittauskortin muu elektroniikka erotettiin toisistaan käyttämällä mittauskortin piirikaaviossa sivulla 5 (Liite 2) näkyvää erotuspiiriä.

Erotuspiirin kytkentä lisättiin piirikaavioon, mutta mittauskorttia ei haluttu testauksen tässä vaiheessa kokonaan uusia. Sen sijaan erotuspiiri lisättiin testilaitteeseen rakentamalla se koekytkentälevylle ja muokkaamalla alkuperäistä mittauskorttia niin, että mittaussignaalit kulkevat erotuspiirikortin kautta. Kuviossa 19 on valokuva koekytkentälevylle rakennetusta erotuspiiristä.

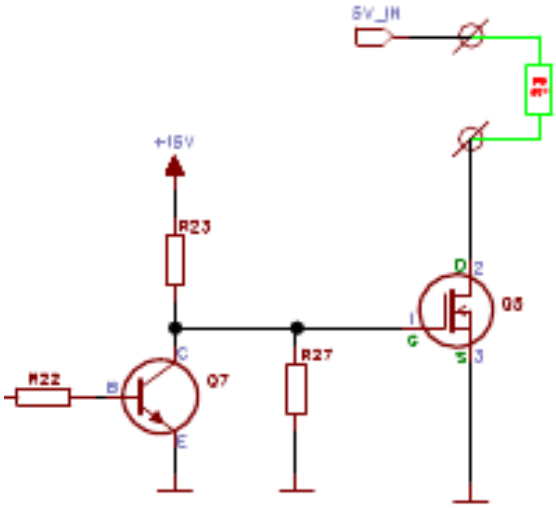


KUVIO 19. Koekytkentälevylle rakennettu erotuspiiri

R100-laitteen kuormittaminen

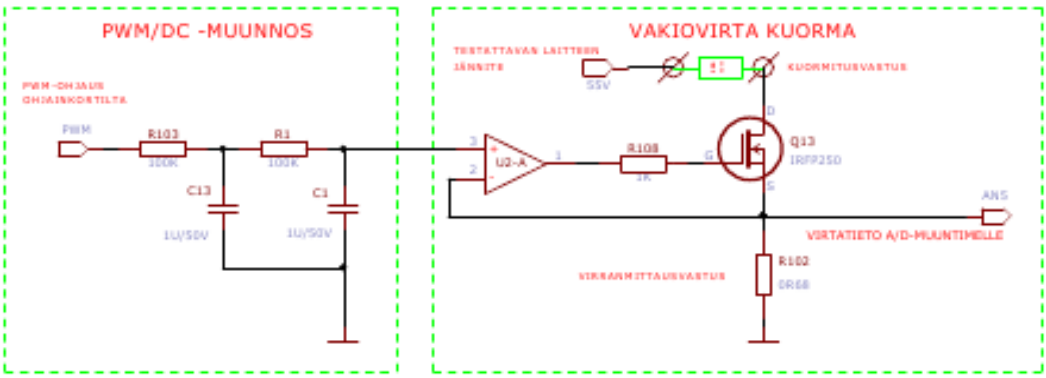
R100:n lähtöjen kuormittamista varten mittauskortille suunniteltiin kuormituskytkennät. 5V- ja 12V-lähtöjen kuormittamiseen käytettiin tehovastuksia, jotka kytketään päälle ja pois MOSFET-transistoreilla, joiden hilaa mikro-ohjain ohjaa signaalitransistoreiden välityksellä. Kuviossa 20 on 5V lähdön kuormituskytkentä. 12V lähdön kuorma toimii samalla periaatteella, mutta siinä on kaksi tehovas-

tusta sarjaan kytkettynä. Kaikkien kuormien kytkennät ovat mittauskortin piirikaavion sivulla 6 (Liite 2).



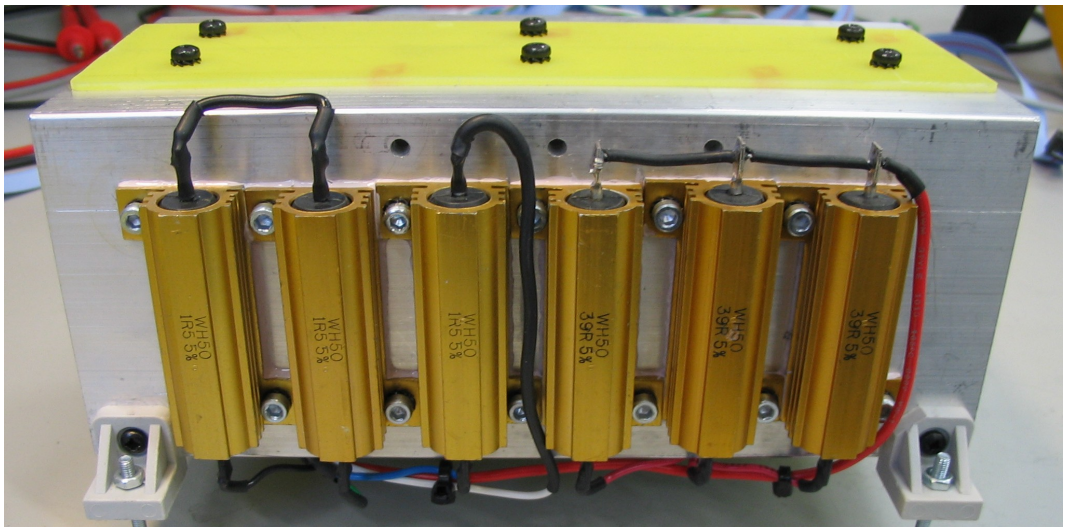
KUVIO 20. Vastuskuorman kytkentä

55V-lähdön kuormittamista varten suunniteltiin säädettävä vakiovirtakuorma, jota mikro-ohjain ohjaa PWM-signaalilla. Kuviossa 21 on yksinkertaistettu kytkentä, josta selviää kuorman toimintaperiaate. Mikro-ohjaimelta tuleva PWM-signaali muutetaan tasajännitteeksi kahdella peräkkäin kytketyllä RC-alipäästösuotimella. Tätä jännitettä käytetään operaatiovahvistimen referenssinä, jonka mukaan operaatiovahvistin säätää kuormitusvastuksien ja FET-transistorien läpi kulkevan virran suuruutta. Virranmittausvastukselta saatavaa jännitetietoa käytetään takaisinkytkennässä ja mikro-ohjaimen A/D -muuntimen tulosignaalina. Kuorman kytkentä on kokonaisuudessaan mittauskortin piirikaavion sivulla 6 (Liite 2).



KUVIO 21. PWM-signaalilla säädettävä vakiovirtakuorma yksinkertaistettuna

R100:n lähtöteho on testin aikana suurimmillaan noin 200W kaikkien kuormien ollessa päällä. Tämä teho muutetaan kuormissa lämmöksi, minkä vuoksi kuormien tehovastukset ja FET -transistorit kiinnitettiin suureen jäähdytyselementtiin (Kuvio 22). Kuormissa syntyvä lämpö siirretään ulos kotelosta takaseinään asennetun tuulettimen avulla, joka imee ilmaa jäähdytyselementin kautta ja puhaltaa lämmenneen ilman ulos kotelosta. Korvausilma virtaa koteloon sisään vastakkaisella puolella sijaitsevasta aukosta.

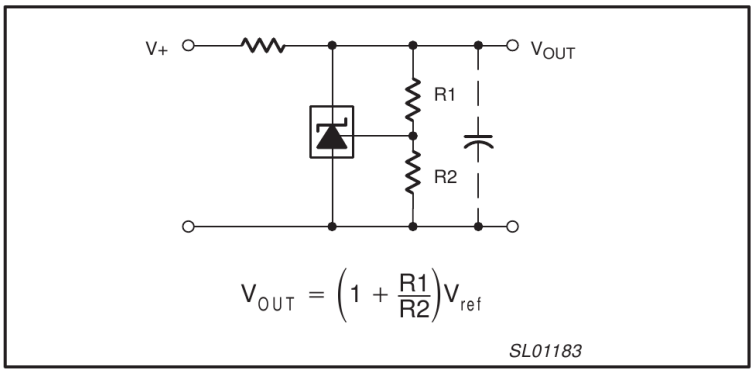


Kuvio 22. Jäähdytysprofiili ja kuormitusvastukset.

VBUS-jännitteen mittaus

R100:n ensiöpuolella sijaitseva VBUS-jännite täytyy säätää oikeaan arvoon testin aikana (R100 tuotannontestausohje, 2). Tämän jännitteen mittaamiseen päätettiin käyttää analogiaelektroniikalla toteutettua kytkentää, joka vertailee VBUS-jännitettä trimmereillä asetettuihin ylä- ja alarajoihin. Kytkentä ei anna mikro-ohjaimelle analogista tietoa jännitteen arvosta, vaan ilmoittaa optoerottimien välityksellä digitaalisen tiedon jännitevertailun tuloksesta. Mikro-ohjaimen ohjelmaan kirjoitettiin koodinpätkä, joka päättelee, onko jännite liian pieni tai suuri vai onko se sallituissa rajoissa. Mittauskytkentä on mittauskortin piirikaavion sivulla 4 (Liite 2).

VBUS-jännitteen mittauskytkennässä käytettiin TL431-tyyppisiä jännitereferenssikomponentteja, joiden avulla on helppo tehdä säädettäviä referenssijännitteitä (TL431 Data Sheet 2002, 2). Jännitereferenssien kytkennät mitoitettiin kuviossa 23 näkyvän kaavan mukaan. Kytkentään lisättiin trimmeri referenssijännitteen hienosäätöä varten.



KUVIO 23. TL431 jännitereferenssin mitoittaminen (TL431 Data Sheet 2002)

3.2 Mekaniikkasuunnittelu

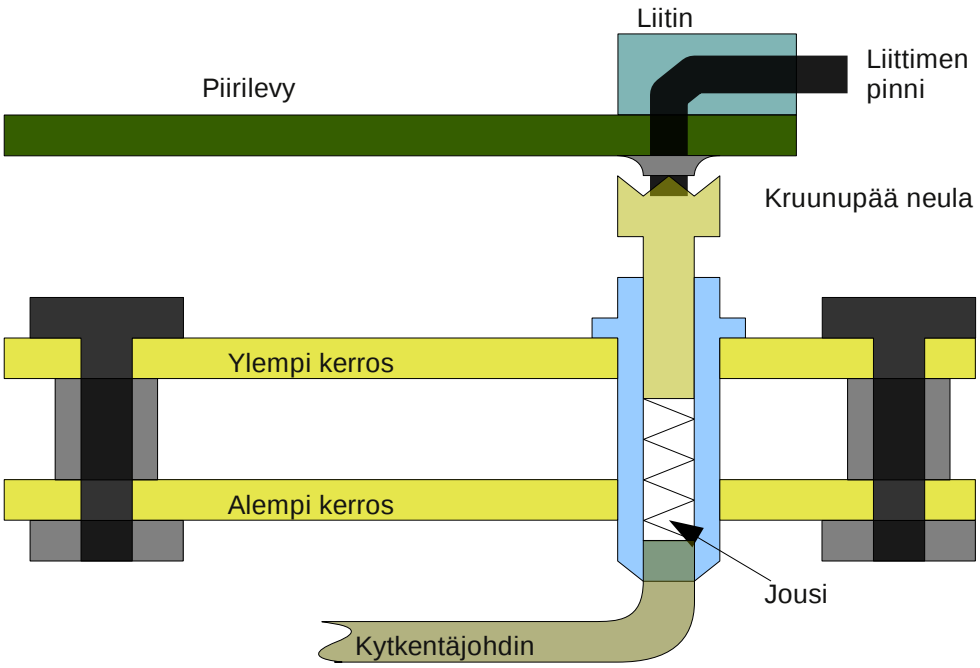
Yksi tärkeä osa-alue testilaitteen suunnittelussa oli mekaniikkasuunnittelu. Mekaniikkasuunnittelussa luotiin laitteen ulkokuori, joka kätkee suurimman osan elektroniikasta sisäänsä. Mekaniikkasuunnittelussa täytyi ottaa huomioon testilaitteen kestävyys tuotannon olosuhteissa. Lisäksi mekaniikkasuunnittelussa huomioitiin laitteen turvallisuus ja testauksen helppous. Mekaniikkasuunnitteluun kuului testilaitteen kotelon lisäksi neulapeti, jonka päälle testattava laite asetetaan testauksen ajaksi.

3.2.1 Neulapeti

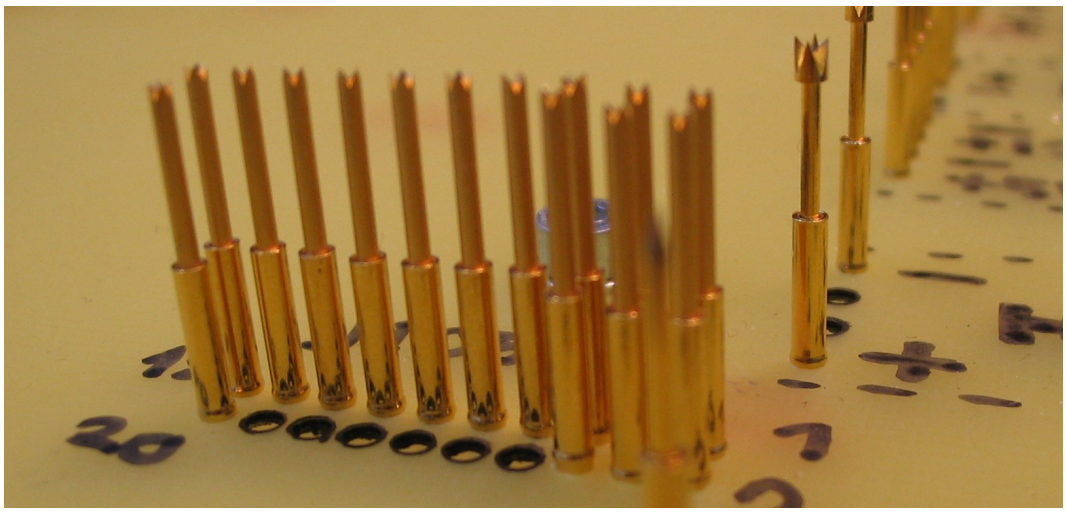
Testattavassa laitteessa on paljon liitäntöjä, jotka täytyy testauksen ajaksi kytkeä testilaitteeseen. Testauksen helpottamiseksi ja nopeuttamiseksi testilaitteeseen suunniteltiin neulapeti, jonka jousilla varustetut neulat kytkevät testattavan laitteen testilaitteeseen.

Neulapedin suunnittelussa käytettiin apuna testattavan laitteen piirikorttikuvaa, jonka päälle oli AutoCad -suunnitteluohjelmalla helppo sijoitella paikat testineuloille. Neulojen paikat valittiin siten, että testattavan laitteen piirilevylle ei tarvinnut tehdä muutoksia. Tämän vuoksi testipisteinä käytettiin liittimien juotospisteitä. Kun testineulojen paikat olivat selvillä, piirikortin kuva poistettiin suunnitelmasta, ja jäljelle jäi vain testineulojen sijoittelukuva.

Neulapeti valmistettiin piirilevyjen jyrsintään tarkoitetulla LPKF ProtoMat -jyrsinlaitteella. Kahteen kuparittomaan piirilevyaihioon jyrsittiin reiät testineuloille sijoittelukuvan mukaisesti. Piirilevyaihiosta valmistetut neulapedin levyt asetettiin päällekkäin noin 10 millimetrin etäisyydelle toisistaan käyttäen apuna piirilevyjen kiinnitykseen tarkoitettuja korotusholkkeja. Kahdesta päällekkäisestä kerroksesta saatiin neulojen kiinnitykseen tukeva rakenne. Yhtä kerrosta käytettäessä neulat olisivat pyrkineet vääntymään kieroon, mutta kahden kerroksen rakenne toimi hyvin. Kuvioissa 23 ja 24 on havainnollistettu neulapedin toimintaa.



KUVIO 23. Neulapedin toiminta



KUVIO 24. Valokuva neulapedin neuloista

Testineulat kuluvat käytössä, ja saattavat vääntyä kieroon. Tämän vuoksi neulapedissä käytettiin asennusholkkeja, joihin varsinainen testineula on helppo asentaa ja tarvittaessa nopeasti vaihtaa. Neulatyypit valittiin IDI:n valmistamasta ICT-100 -sarjasta. Lähes kaikissa testipisteissä kontaktipintana oli piirilevylle juotettu liittimen jalka. Paras sähköinen kontakti tällaisiin testipisteisiin saatiin käyttämällä kruunutyyppistä neulaa.

Testineulojen alapäähän juotettiin johtimet, joilla neulat yhdistettiin mittauskortille. R100:n suurivirtaisissa lähdöissä käytettiin useita neuloja rinnankytkettynä, koska näin haluttiin varmistaa kunnollinen kontakti ja estää yhden neulan neulan läpi kulkevan virran nouseminen liian suureksi. Tästä saadaan myös etua silloin, jos yksi neuloista vioittuu esimerkiksi jumittumalla pohjaan. Tällöin virta voi vielä kulkea muiden neulojen kautta.

3.2.2 Kotelointi

Testilaitetta varten yritettiin etsiä valmista kotelointiratkaisua, mutta koska kaikki löydetyt valmiit kotelovaihtoehdot olisivat vaatineet suuria muutoksia, päätettiin laitteelle suunnitella oma kotelointiratkaisu. Kotelosta päätettiin tehdä kaksiosainen, jossa testilaitteen elektroniikalle on oma suljettu osa, ja testattavalle

laitteelle läpinäkyvällä kannella varustettu osa. Kannen läpinäkyvyys oli tärkeää, koska testaajan täytyy nähdä testattava laite testin aikana, jotta laitteen trimmereitä voisi säätää. Toisaalta kansi täytyi olla sähköturvallisuuden vuoksi hyvin suljettu. Testattavan laitteen trimmereiden säätämistä varten kanteen porattiin kuitenkin muutamia reikiä, joista pitkän ruuvimeisselin voi työntää kannen alle. Sähköturvallisuuden vuoksi tähän täytyy käyttää eristettyä ruuvimeisseliä.

Koteloinnin suunnittelussa käytettiin apuna AutoCAD-ohjelmaa. Kotelon alaosa päätettiin valmistaa PVC-muovista, koska se on kestävää, ja sitä on helppo työstää. Suojakannen täytyi olla läpinäkyvää materiaalina, joten siihen käytettiin polykarbonaattilevyä. Kaikki kotelon ja suojakannen osat tilattiin valmiiksi oikeaan mittaan sahattuina Kalustemuovi Virtala Oy:stä. Kokoonpano tehtiin itse MEG Power Oy:llä. Liitteessä 3 on kotelon ja suojakannen AutoCad kuva.

3.3 Ohjelmointi

Yksi suuritöinen ja tärkeä testilaiteprojektin osa-alue oli mikro-ohjaimen ohjelmointi. Testilaitteen koko toiminta perustuu mikro-ohjaimen ohjelmakoodiin kirjoitettuun testiohjelmaan. Ohjelmakoodi ohjaa kaikkia muita testilaitteen toimintoja paitsi jännitteensyötön suojapiiriä, jota ohjataan suoraan testilaitteen kannen turvakytkimillä.

3.3.1 Ohjelmointikieli

Mikro-ohjaimen ohjelmointikieleksi valittiin C-kieli. Valinta oli helppo C-kielen helppouden ja hyvän kääntäjätuen vuoksi. Myös runsas koodiesimerkkien määrä ja valmiiden ohjelmapätkien helppo siirrettävyys muihin sovelluksiin vaikutti valintaan.

Ohjelmakehitys aloitettiin käyttämällä HI-TECH PICC -kääntäjää ja graafista HI-TIDE-ohjelmointiympäristöä. Kyseisen ohjelmiston demoversion rajoitukset estivät kuitenkin pian ohjelmakehityksen jatkamisen. Demoversiossa on rajoitettu suurin sallittu ohjelmamuistin käyttö kahteen kilotavuun. Tätä ei osattu ottaa

huomioon ennen ohjelmoinnin aloitusta, koska ohjelmamuistin tarpeen ei arvattu kasvavan näin suureksi. Maksullisen version hankinnan tullessa ajankohtaiseksi löysimme vaihtoehtoisen kääntäjän, jonka hinta oli vain murto-osa HI-TECH PICC -kääntäjän hinnasta. Ohjelmakehitystä päätettiin siis jatkaa SourceBoost Technologies -nimisen yrityksen tekemällä BoostC-kääntäjällä. Tämä kääntäjä osoittautuikin hinta/laatusuhteeltaan erittäin hyväksi valinnaksi. Ohjelmakoodin siirtäminen kääntäjältä toiselle tuotti hiukan lisätyötä, mutta C-kielen luontaisesti helpon siirrettävyyden ansiosta se onnistui kuitenkin helposti. Vain joitakin epästandardeja laitteistokohtaisia bittien ja porttien käsittelyfunktioita ja makroja piti vähän muuttaa.

3.3.2 Laitteistoajurit

Ohjelmakehitys aloitettiin laitteiston ajurien kirjoittamisesta. Ajurit pyrittiin tekemään modulaariseksi, jotta ne voidaan tarvittaessa helposti kopioida uusiin projekteihin. Ajurit ovat hyvin yksinkertaisia, mutta niiden helppokäyttöiset käsittelyfunktiot helpottivat ja nopeuttivat varsinaisen testiohjelman kirjoittamista huomattavasti. Selkeät ja helposti ymmärrettävät laitteistonkäsittelyfunktiot tekivät testiohjelmasta myös helposti luettavaa ja loogista koodia.

Ensimmäinen ohjelmointitehtävä oli LCD-näytön ajurin ja käyttöliittymän kirjoittaminen, minkä jälkeen näyttöä voitiin käyttää laitteiston ja ohjelmiston testaamisessa ja virheiden etsimisessä. LCD-näytön ajurin pohjana käytettiin internetistä löydettyä Andy Kunz:n LCD-näyttöajuria, jota muokattiin hieman tähän testilaitteeseen sopivaksi. Lähdekoodi on liitteessä 6.

A/D-muuntimen ajurin kirjoittamisessa käytettiin apuna mikro-ohjaimen datalehteä. Alustusfunktion rekisterit ja asetukset selvitettiin datalehden taulukoista. Datalehdessä oli hyvät ohjeet A/D-muunnoksen tekemistä varten, joissa selitettiin vaihe vaiheelta muunnoksen eteneminen ja tuloksen tallentaminen. C-kielinen ajurin lähdekoodi on liitteessä 7.

I/O-laajennuspiirin PCF8574 ohjaus tapahtui I2C-sarjaväylän kautta, joten sen ajurissa käytettiin pohjana internetistä löytynyttä I2C-väylän ajuria, jonka oli kirjoittanut Michael Alon. Tästä ajurista poimittiin vain I2C-väylän käsittely-funktiot. Loput osat ajurista täytyi kirjoittaa itse. PCF8574-piirin ohjaus täytyi selvittää tarkasti datalehden ajastuskaavioista. Niiden avulla kirjoitettiin piirin ohjausfunktiot. Ajurin C-kielinen lähdekoodi on liitteessä 9.

Myös mikro-ohjaimen sisäisille PWM-toiminnoille kirjoitettiin oma ajuritiedosto. PWM-lähtöjen ohjaus selvitettiin mikro-ohjaimen datalehdestä ja ohjelmakoodi kirjoitettiin datalehden tietoja soveltamalla. PWM-arvon muuttaminen oli hyvin yksinkertainen toimenpide, mutta PWM-lähtöjen sopivien asetusten tekeminen vaati hieman enemmän selvittelyä ja testaamista. Ajurin C-kielinen lähdekoodi on liitteessä 10.

3.3.3 Testiohjelma

Testiohjelman lähdekoodi kirjoitettiin omaan C-kieliseen tiedostoon. Lähdekoodi löytyy liitteestä 12. Ohjelman kulku etenee suoraviivaisesti ylhäältä alas. Testiohjelman kirjoittamisessa kiinnitettiin erityistä huomiota helppoon luettavuuteen, jotta testiohjelmaa on tarvittaessa helppo muuttaa jälkeenpäin. Testiohjelma koostuu pääasiassa laiteajurien ja LCD -näytön ohjauskomennoista. Ohjelmassa on yksi silmukka, joka käydään läpi kertaalleen jokaisen testattavan laitteen kohdalla. Mitään testidataa ei säilytetä, vaan kaikki muuttujat nollataan aina silmukan alussa. Näin varmistetaan, että testi on samanlainen jokaiselle testattavalle laitteelle.

3.3.4 Ohjelman testaus

Ohjelmistokehityksen aikana tehtiin usein väliaikaisia pieniä ohjelman pätkiä, jotka esimerkiksi kirjoittivat erilaista informaatiota näytölle tai ohjasivat jotain I/O-linjaa. Näin voitiin varmistaa kaikkien toimintojen toimivuus ennen varsinaisen testiohjelman kirjoittamista. Tästä esimerkkinä on seuraava A/D-muuntimen kanavien testaukseen ja kalibrointiin tarkoitettu ohjelman katkelma:

```
// Testiohjelma ad-muuntimen testaukseen
// 12 kanavainen "jännitemittari"
void voltagemeter(void)
{
            lcd_clear();
            lcd_gotoxy(1,1);
            lcd_printf("A ");
            lcd_gotoxy(1,8);
            lcd_printf("B ");
            lcd_gotoxy(1,15);
            lcd_printf("C ");
            lcd_gotoxy(2,1);
            lcd_printf("D ");
            lcd_gotoxy(2,8);
            lcd_printf("E ");
            lcd_gotoxy(2,15);
            lcd_printf("F ");
            lcd_gotoxy(3,1);
            lcd_printf("G ");
            lcd_gotoxy(3,8);
            lcd_printf("H ");
            lcd_gotoxy(3,15);
            lcd_printf("I ");
            lcd_gotoxy(4,1);
            lcd_printf("J ");
            lcd_gotoxy(4,8);
            lcd_printf("K ");
            lcd_gotoxy(4,15);
            lcd_printf("L ");


            while(1)
            {
                        lcd_gotoxy(1,3);
                        lcd_printdec(Read_AD(0));
                        lcd_gotoxy(1,10);
                        lcd_printdec(Read_AD(1));
                        lcd_gotoxy(1,17);
                        lcd_printdec(Read_AD(2));
                        lcd_gotoxy(2,3);
                        lcd_printdec(Read_AD(4) * 4);
                        lcd_gotoxy(2,10);
                        lcd_printdec(Read_AD(5));
                        lcd_gotoxy(2,17);
                        lcd_printdec(Read_AD(6) * 2);
                        lcd_gotoxy(3,3);
                        lcd_printdec(Read_AD(7));
                        lcd_gotoxy(3,10);
                        lcd_printdec(Read_AD(8));
                        lcd_gotoxy(3,17);
                        lcd_printdec(Read_AD(9));
                        lcd_gotoxy(4,3);
                        lcd_printdec(Read_AD(10) * 16);
                        lcd_gotoxy(4,10);
                        lcd_printdec(Read_AD(11));
                        lcd_gotoxy(4,17);
                        lcd_printdec(Read_AD(13));
                        delay_ms(200);
            }
}
```

4 YHTEENVETO

Tämän työn tarkoitus oli suunnitella ja valmistaa testilaite R100-tyyppisen hakkuriteholähteen testaamista varten. Tässä onnistuttiin, ja laite on tämän työn kirjoittamishetkellä MEG Power Oy:n tuotanto-osastolla toimintavalmiina. Laitteen suunnittelussa törmättiin monta kertaa erilaisiin ongelmiin, jotka kaikki saatiin ratkaistua tavalla tai toisella. Tämä oli erinomainen tilaisuus tutustua elektroniikkasuunnitteluun ja uuden laitteen tuotekehitykseen yleensä.

Mielestäni tämä projekti oli äärimmäisen hyvä opinnäytetyön aiheeksi, koska se oli niin mielenkiintoinen ja monipuolinen. Työ opetti ongelman ratkaisua, tiedon hankintaa ja sen soveltamista käytäntöön. Työssä käsiteltiin monia koulussa opiskeltuja aiheita erityisesti mikro-ohjaimen ohjelmoinnin yhteydessä. Ohjelmointityö olisi ollut huomattavasti vaikeampaa ilman koulussa C-kielen kursseilla tehtyjä ohjelmointiharjoituksia. Tämän työn ohjelmointiosuus oli kuitenkin monin verroin vaativampi kuin yksikään koulussa tehty harjoitus, mistä kertoo huima lähdekoodirivien määrä.

Työn laajuutta ei aluksi osattu arvioida oikein ja testilaitteen arvioitiin valmistuvan alle neljässä kuukaudessa. Todellinen aika oli kuitenkin huomattavasti pidempi ja testilaite oli valmis vasta noin kahdeksan kuukauden kuluttua suunnittelun aloittamisesta. Työtä ei tosin tehty kokoaikaa täysipäiväisesti, vaan sitä tehtiin muun opiskelun ohessa osa-aikaisesti. Jälkeenpäin ajateltuna näin pitkän ajan kuluminen ei ole ihme, kun ottaa huomioon työn laajuuden ja tekijän kokemattomuuden. Kokemusta kuitenkin karttui työn aikana siinä määrin, että tämän työn jälkeen oli huomattavasti helpompaa osallistua muihin tuotekehitysprojekteihin.

Jo testilaitteen testausvaiheessa tuli mieleen monia parannusideoita. Näitä ei kuitenkaan alettu toteuttamaan, koska sillä menetelmällä laite ei olisi tullut koskaan valmiiksi. Yksi idea oli käyttää modulaarisempaa ratkaisua. A/D-muuntimena voisi käyttää mikro-ohjaimen sisäisen muuntimen sijasta sarjaväylälle kytkettävää mallia. Tällöin mittauksiin olisi voinut käyttää useita A/D-muuntimia, jotka olisi sijoitettu hyvin lähelle mitattavaa kohdetta, vaikkapa neulapedin alle mittausneulojen läheisyyteen. Sarjaväylälle kytkettävät A/D-muunninmoduulit olisi

helppo erottaa sarjaväylän kohdalta tähän tarkoitetuilla komponenteilla. Myös kaikki I/O -signaalit olisi voitu tehdä sarjaväylään kytkettävillä I/O-laajennuspiireillä, jolloin olisi selvitty huomattavasti vähemmällä kaapelimäärällä ohjainkortin ja mittauskortin välillä.

Mielestäni yksi tärkein työssä oppimani asia on se, että suunnittelutyössä täytyy uskoa itseensä. Suunnittelijan ei saa jatkuvasti epäröidä eikä koko ajan miettiä miten joku toinen jonkun asian tekisi. Täytyy uskaltaa tuoda omat ideat ja näkemykset esille ja kokeilla niitä käytännössä.

LÄHTEET

Ahoranta, J. 1998. Sähkötekniikka. 1.-3. painos. Porvoo: WSOY

STMicroelectronics. ITA6V5B3 datasheet. 2007. Datalehti [viitattu 13.4.2009]. Saatavissa: http://www.st.com/stonline/products/literature/ds/11085/itaxxb3.pdf

National Semiconductor Corporation. LM317 datasheet. 2008. Datalehti [viitattu 13.4.2009]. Saatavissa: http://www.national.com/ds/LM/LM117.pdf

Philips Semiconductor. PCF8574 datasheet. 2002. Datalehti [viitattu 13.4.2009]. Saatavissa: http://www.nxp.com/acrobat/datasheets/PCF8574_4.pdf

Microchip Technology Inc. PIC16F882/883/884/886/887 Data Sheet. 2008. Datalehti [viitattu 13.4.2009]. Saatavissa: http://ww1.microchip.com/downloads/en/DeviceDoc/41291E.pdf

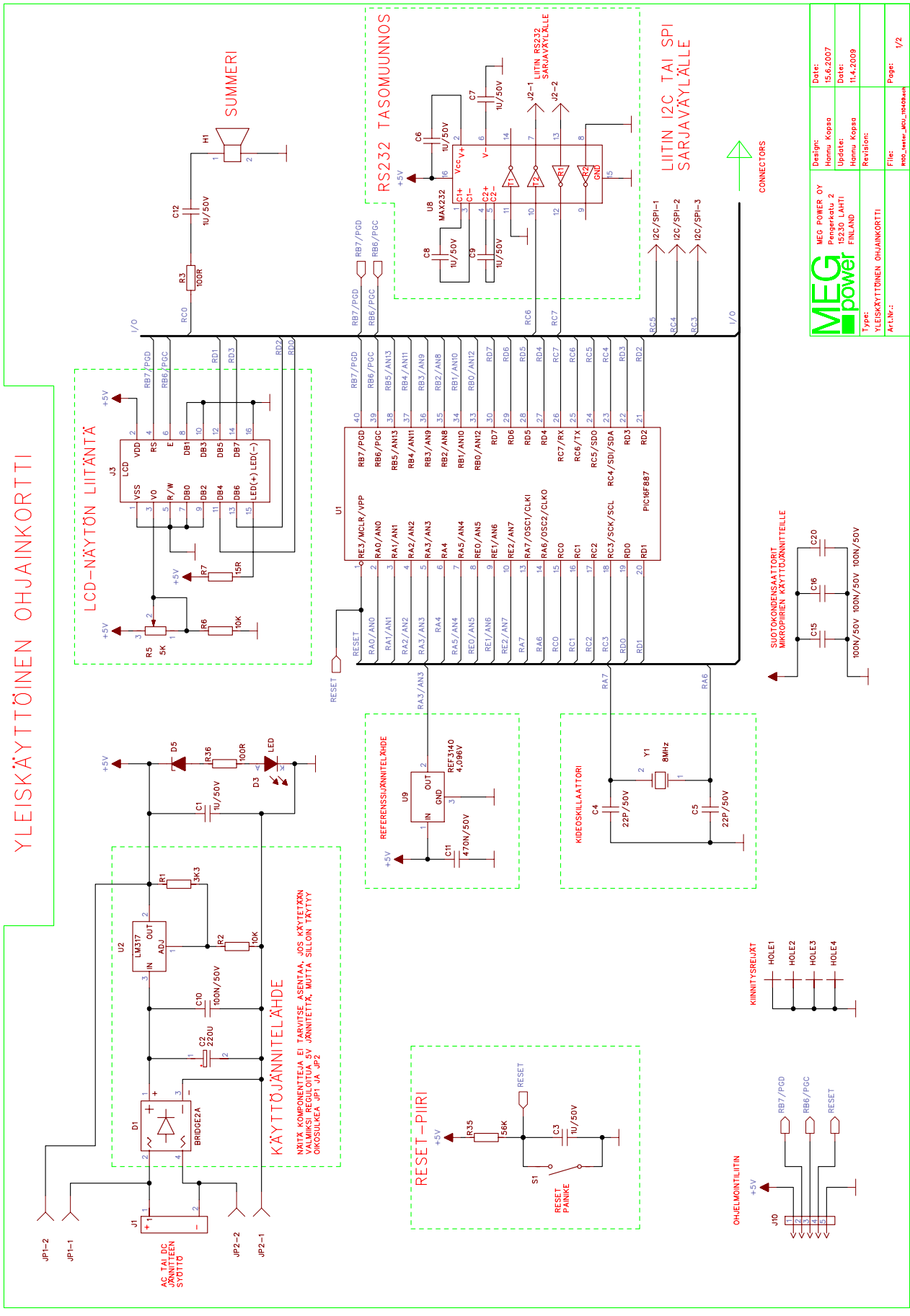
Texas Instruments Incorporated. REF3140 datasheet. 2006. Datalehti [viitattu 13.4.2009]. Saatavissa: http://www.ti.com/lit/gpn/ref3140

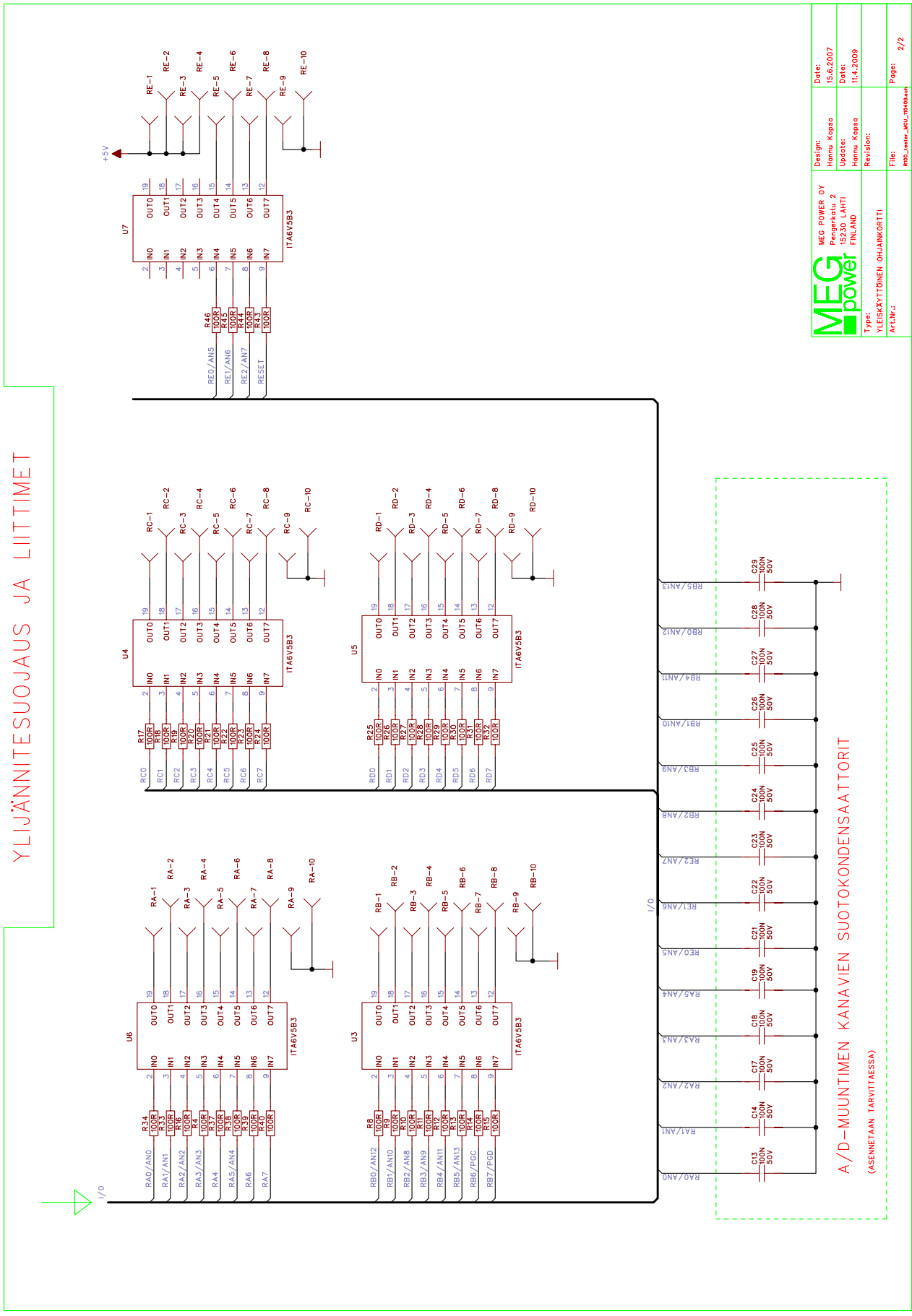
Data Modul AG. Specification for BTHQ 42008VSS-SMN-LEDwhite. 2003. Datalehti [viitattu 13.4.2009]. Saatavissa: http://www.datamodul.com/us/page/popup_pages/Standard%20monochromatic%20alpha/alphanumeric/BT%2042008/BTHQ%2042008VSS-SMN-LEDwhite.pdf

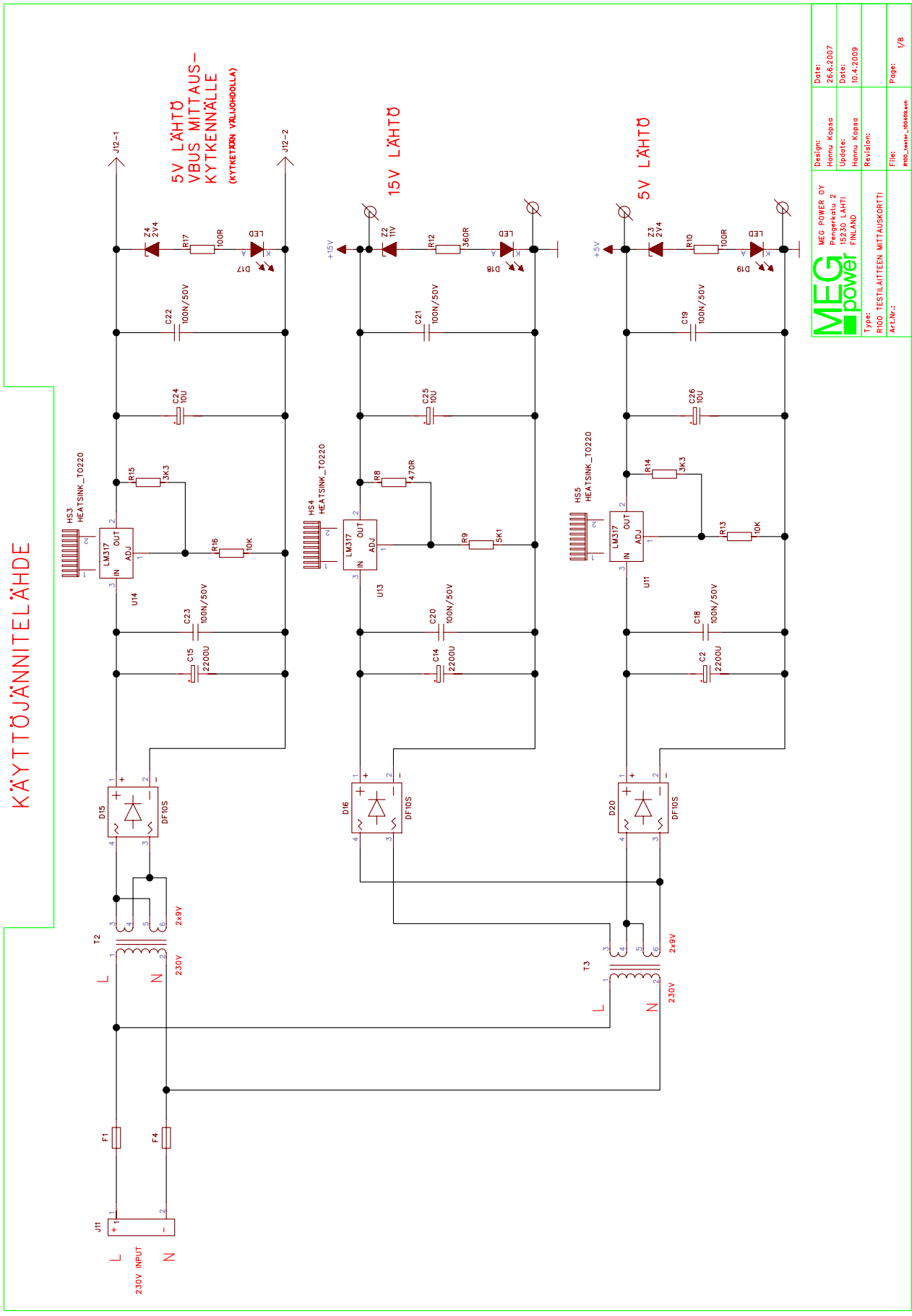
Philips Semiconductor. TL431 datasheet. 2002. Datalehti [viitattu 13.4.2009]. Saatavissa: http://www.nxp.com/acrobat/datasheets/TL431C_AC_I_AI_LM431AC_2.pdf

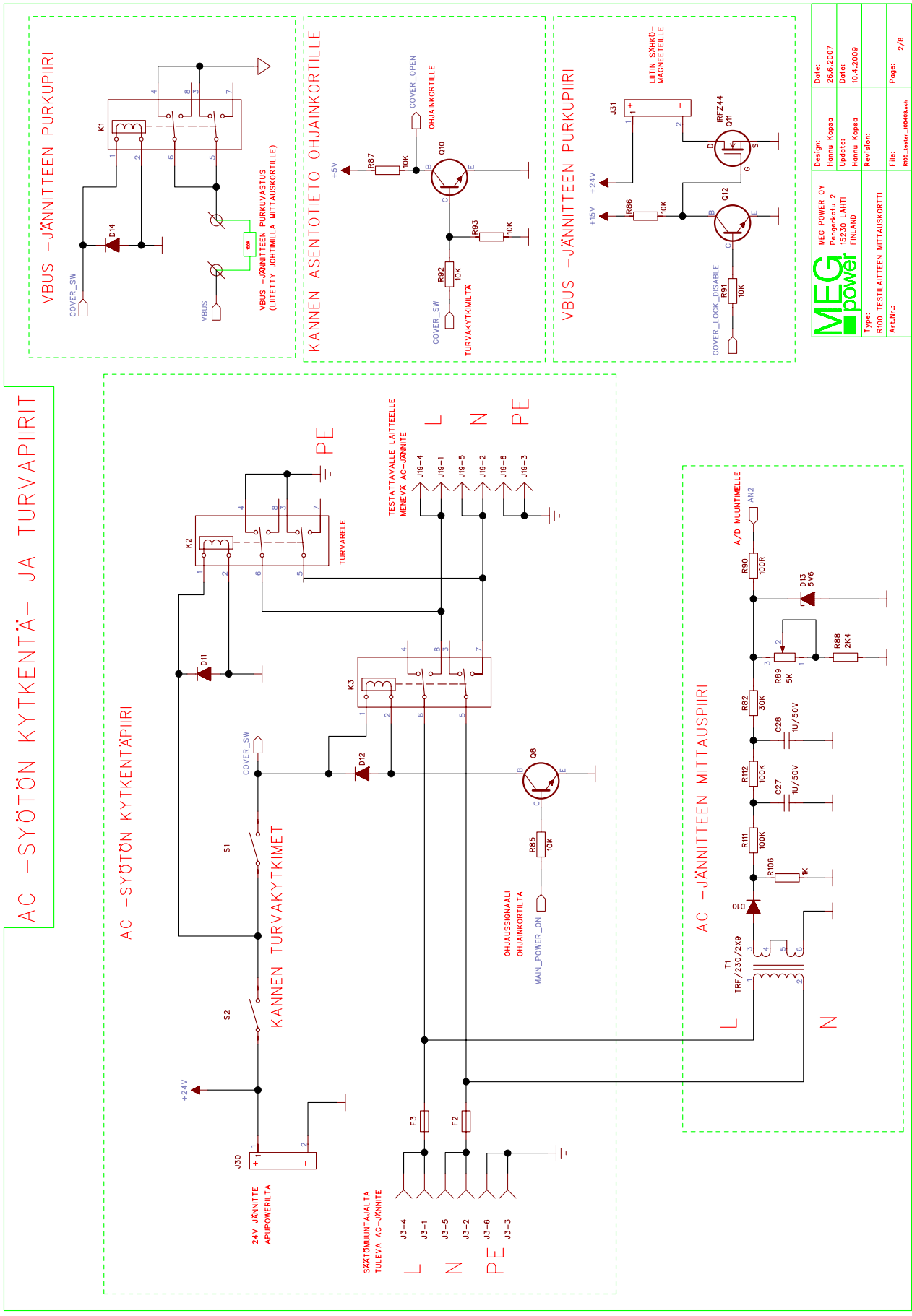
Volotinen V. 1997. Analoginen elektroniikka. 1.-2. painos. Porvoo: WSOY

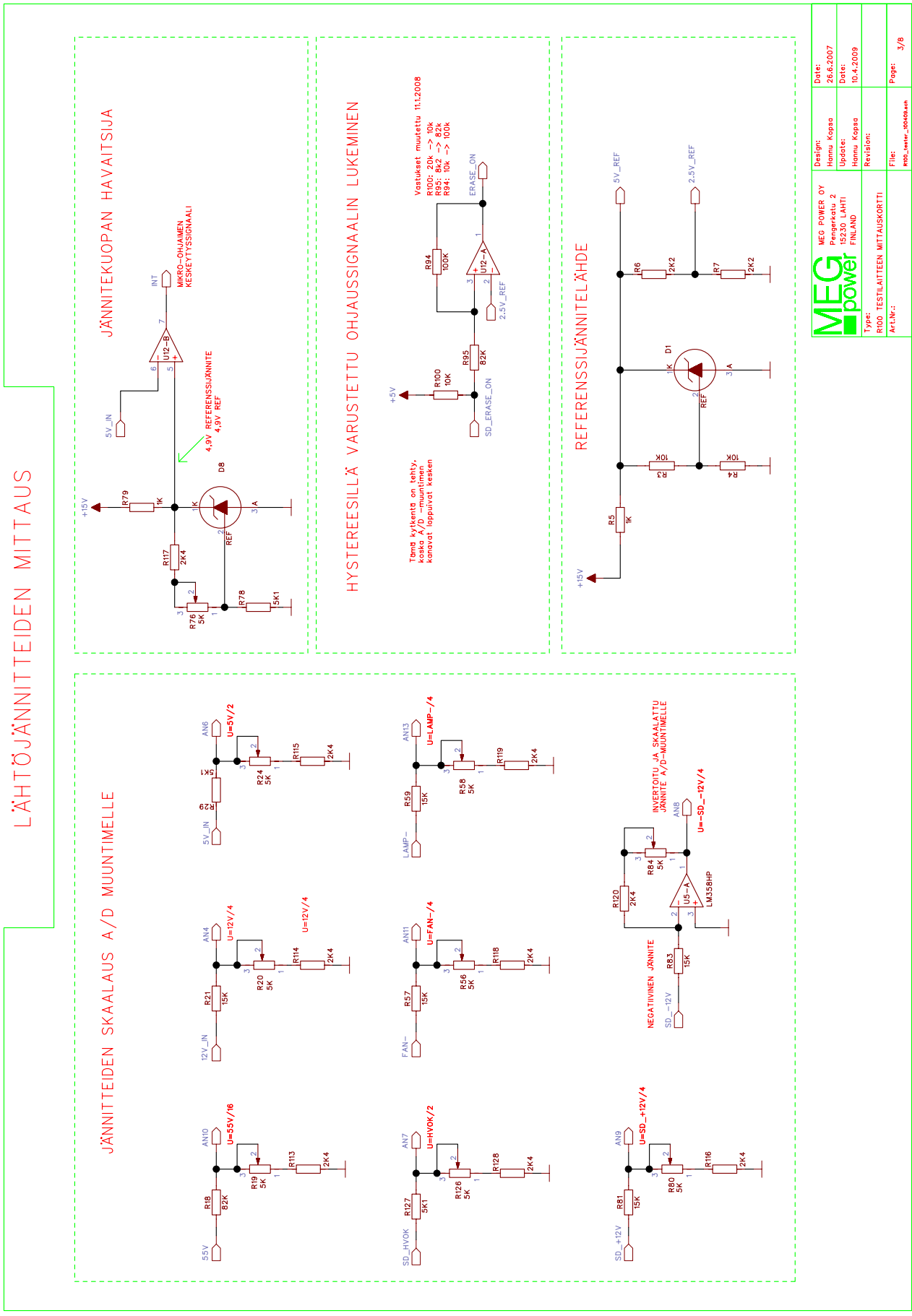
LIITTEET

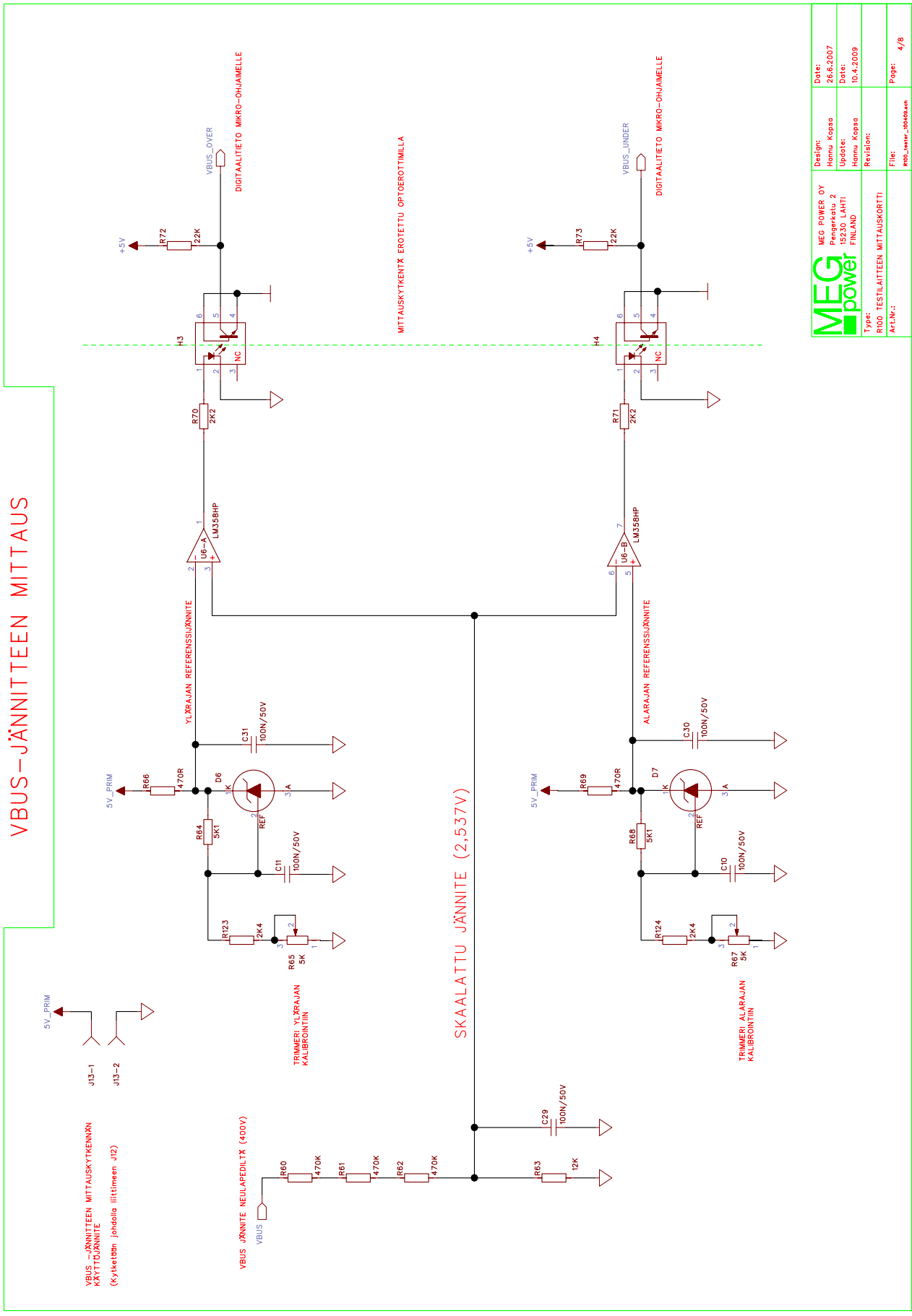

YLEISKÄYTTÖINEN OHJAINKORTTI
LCD-NÄYTÖN LIITÄNTÄ
SUMMERI
RS232 TASOMUUNNOS
LIITIN I2C TAI SPI SARJAVÄYLÄLLE
CONNECTORS
SUOJAKONDENSAATTORIT MIKROPIIRIEN KÄYTTÖJÄNNITTEILLE
REFERENSSIJÄNNITELÄHDE
KIDEOSKILLAATTORI
KÄYTTÖJÄNNITELÄHDE
RESET-PIIRI
KIINNITYSREIJÄT
OHJELMOINTILIITIN
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Type: YLEISKÄYTTÖINEN OHJAINKORTTI
Date: 15.6.2007
Date: 11.4.2009
Page: 1/2

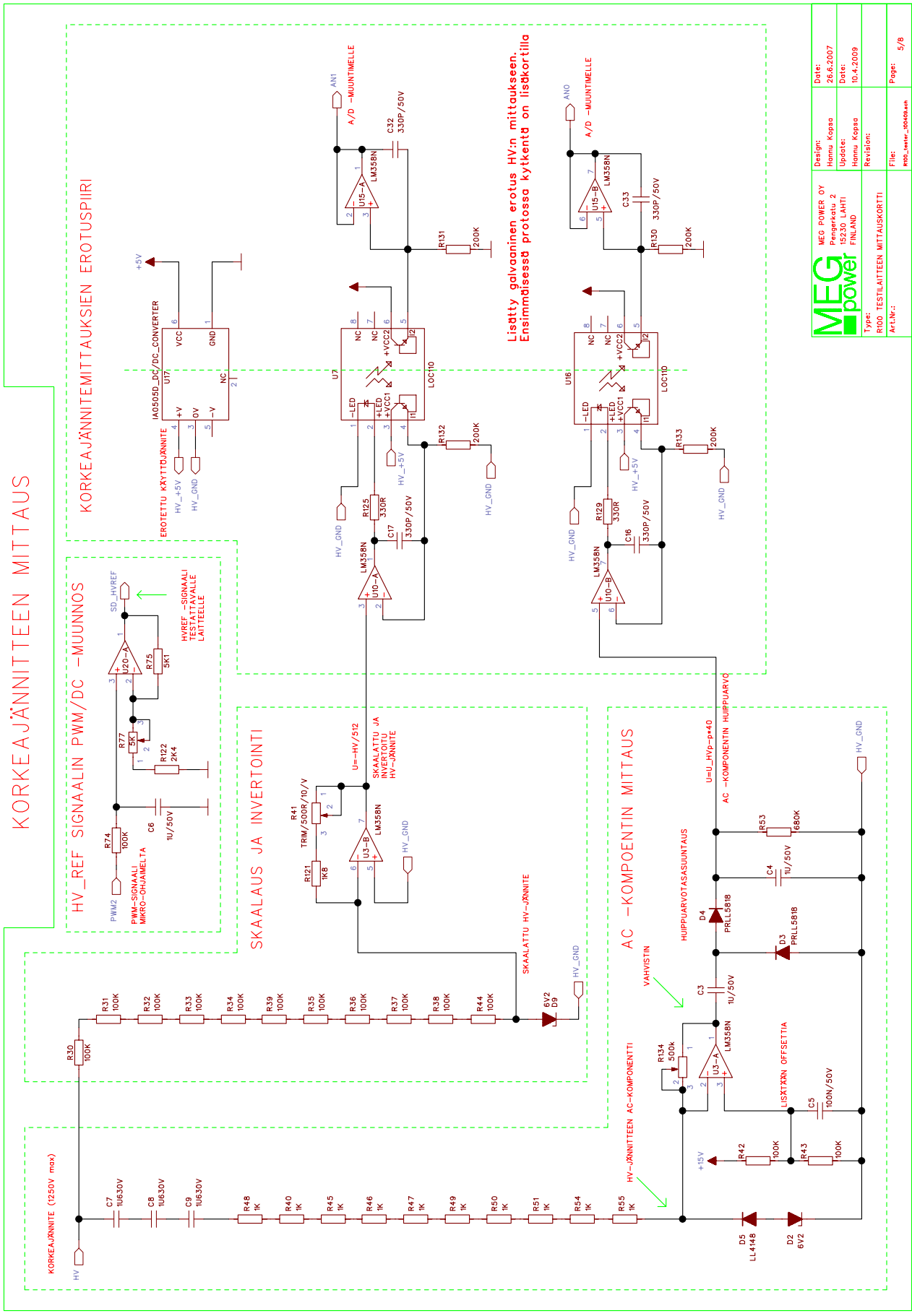

YLIJÄNNITESUOJAUS JA LIITTIMET
A/D-MUUNTIMEN KANAVIEN SUOTOKONDENSAATTORIT
(ASENNETAAN TARVITTAESSA)
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Date: 15.6.2007
Date: 11.4.2009
Page: 2/2
Type: YLEISKÄYTTÖINEN OHJAINKORTTI
ArtiNr.:
File:
U3
U4
U5
U6
U7
ITA6V5B3
I/O
+5V
RESET

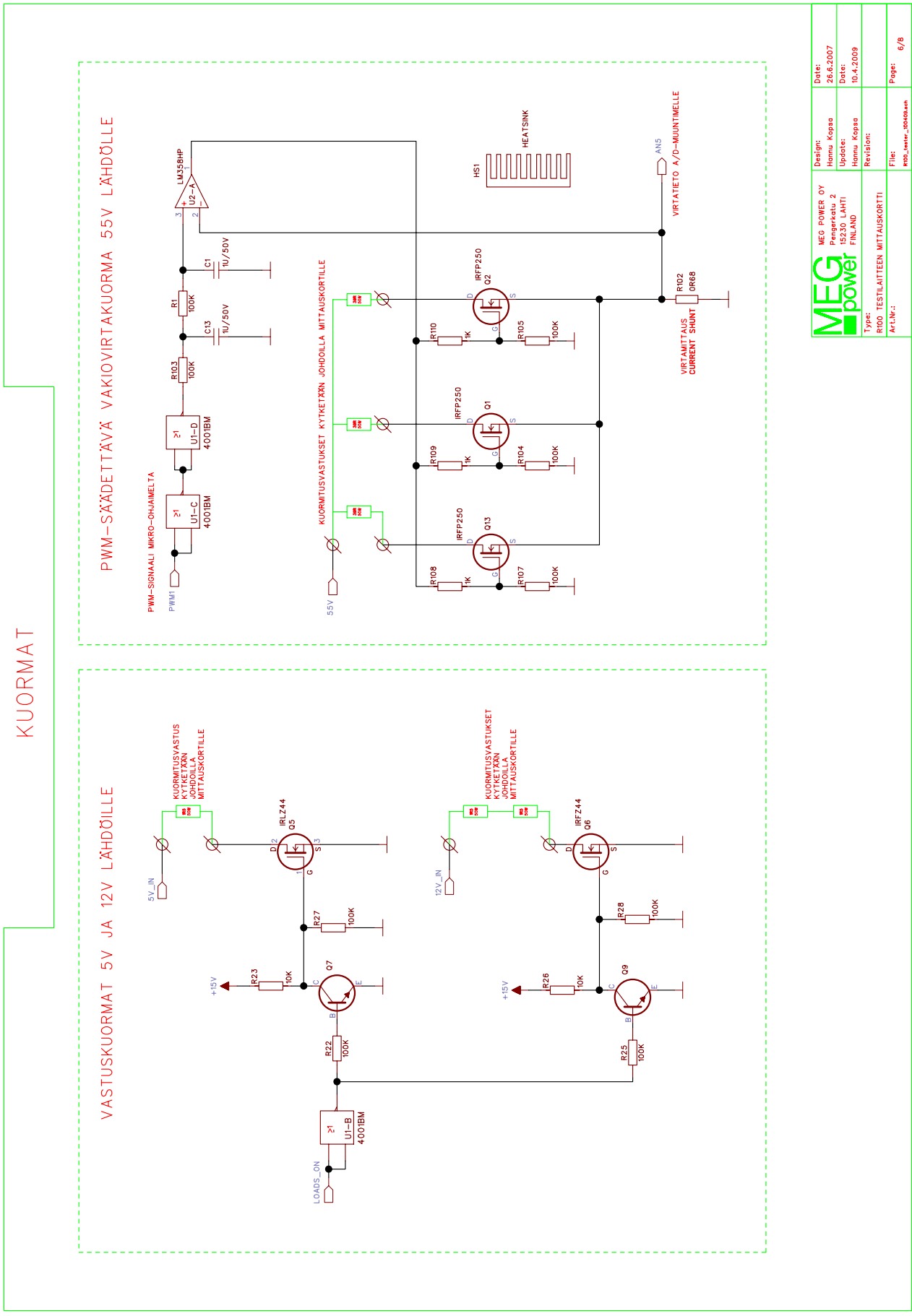

KÄYTTÖJÄNNITELÄHDE
5V LÄHTÖ
VBUS MITTAUS-
KYTKENNÄLLE
(KYTKETÄÄN VÄLIJOHDOLLA)
15V LÄHTÖ
5V LÄHTÖ
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Type: R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:
Date: 26.6.2007
Date: 10.4.2009
Page: 1/8
File: R100_tester_300409.sch
230V INPUT
L
N
F1
F4
J11
T2
T3
D15
D16
D20
DF10S
U14
U13
U11
LM317
HS3
HS4
HS5
HEATSINK_TO220
J12-1
J12-2
Z4 2V4
Z2 11V
Z3 2V4
D17
D18
D19
R17 100R
R12 360R
R10 100R
R15 3K3
R16 10K
R8 470R
R9 5K1
R14 3K3
R13 10K
C22 100N/50V
C21 100N/50V
C19 100N/50V
C24 10U
C25 10U
C26 10U
C23 100N/50V
C20 100N/50V
C18 100N/50V
C15 2200U
C14 2200U
C2 2200U
2x9V
230V

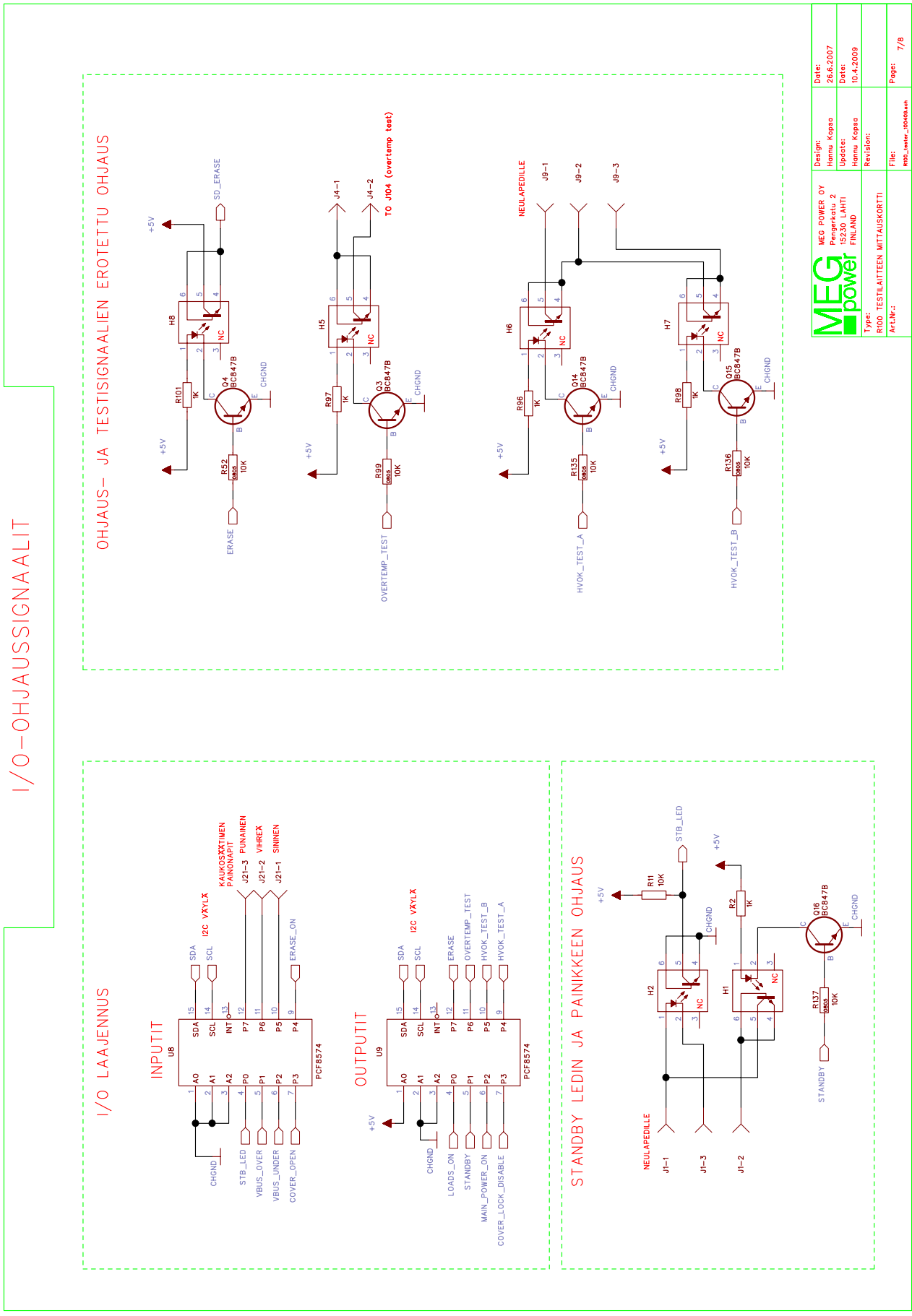

AC -SYÖTÖN KYTKENTÄ- JA TURVAPIIRIT
AC -SYÖTÖN KYTKENTÄPIIRI
KANNEN TURVAKYTKIMET
AC -JÄNNITTEEN MITTAUSPIIRI
VBUS -JÄNNITTEEN PURKUPIIRI
KANNEN ASENTOTIETO OHJAINKORTILLE
VBUS -JÄNNITTEEN PURKUPIIRI
MEG power
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Date: 26.6.2007
Date: 10.4.2009
Page: 2/8
Type: R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:
File: R100_tester_00409.sch
K1
K2
K3
D14
D11
D12
D10
D13 5V6
S1
S2
J30
J31
J3-4
J3-1
J3-5
J3-2
J3-6
J3-3
J19-4
J19-1
J19-5
J19-2
J19-6
J19-3
TESTATTAVALLE LAITTEELLE MENEVÄ AC-JÄNNITE
TURVARELE
SÄÄTÖMUUNTAJALTA TULEVA AC-JÄNNITE
24V JÄNNITTE APUPOWERILTA
OHJAUSSIGNAALI OHJAINKORTILTA
MAIN_POWER_ON
COVER_SW
COVER_OPEN
OHJAINKORTILLE
TURVAKYTKIMILTÄ
COVER_LOCK_DISABLE
VBUS
VBUS -JÄNNITTEEN PURKUVASTUS (LIITETTY JOHTIMILLA MITTAUSKORTILLE)
LIITIN SÄHKÖ-MAGNEETEILLE
A/D MUUNTIMELLE
AN2
T1
TRF/230/2X9
IRFZ44
Q8
Q10
Q11
Q12
R85 10K
R86 10K
R87 10K
R88 2K4
R89 5K
R90 100R
R91 10K
R92 10K
R93 10K
R82 30K
R106 1K
R111 100K
R112 100K
C27 1U/50V
C28 1U/50V
F2
F3
+5V
+15V
+24V

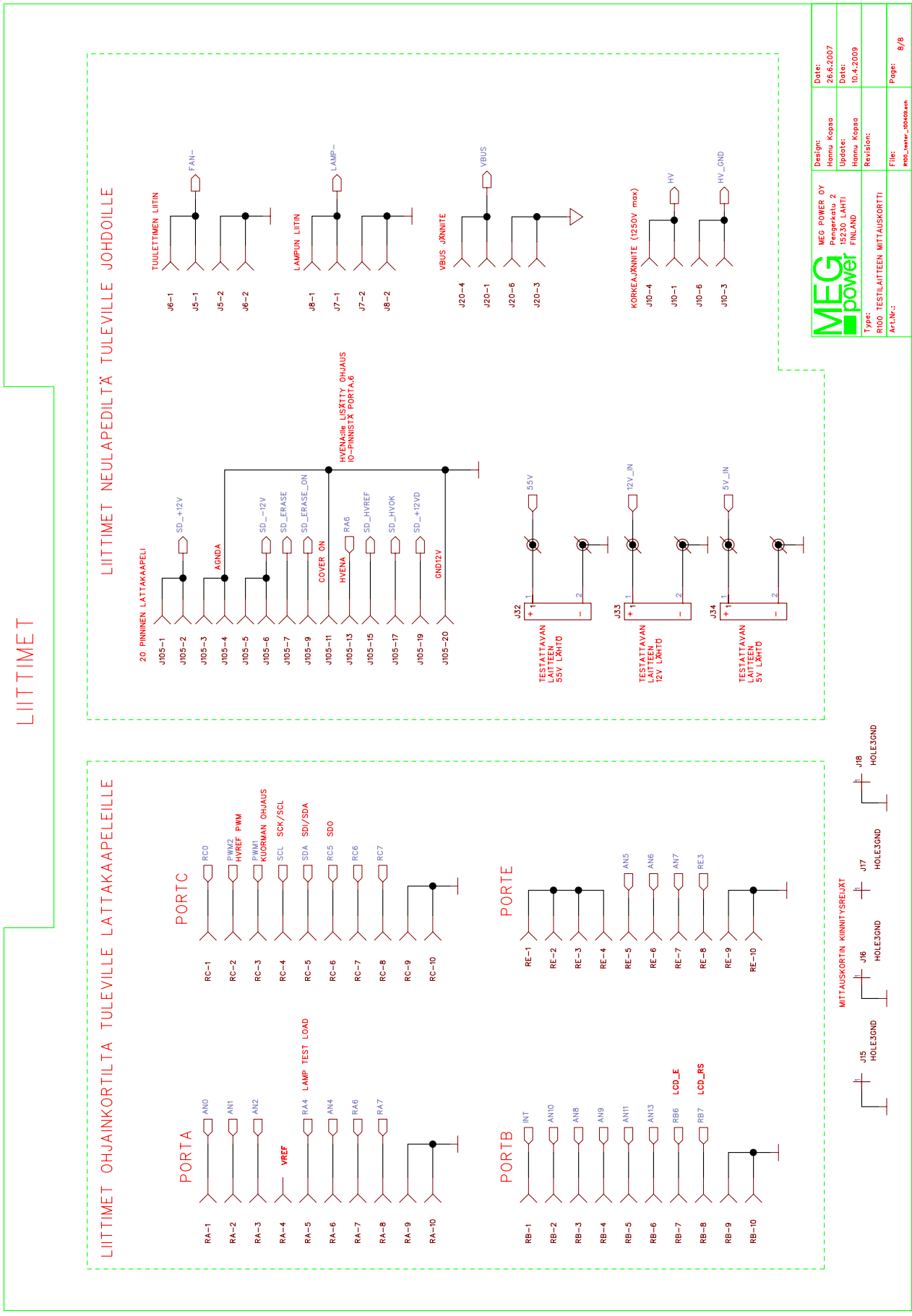

LÄHTÖJÄNNITTEIDEN MITTAUS
JÄNNITTEIDEN SKAALAUS A/D MUUNTIMELLE
JÄNNITEKUOPAN HAVAITSIJA
HYSTEREESILLÄ VARUSTETTU OHJAUSSIGNAALIN LUKEMINEN
REFERENSSIJÄNNITELÄHDE
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Date: 26.6.2007
Date: 10.4.2009
Page: 3/8
Type: R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:
File: R100_tester_300409.sch


VBUS–JÄNNITTEEN MITTAUS
MITTAUSKYTKENTÄ EROTETTU OPTOEROTTIMILLA
SKAALATTU JÄNNITE (2,537V)
VBUS_OVER
VBUS_UNDER
DIGITAALITIETO MIKRO-OHJAIMELLE
+5V
R72 22K
R73 22K
H3
H4
NC
R70 2K2
R71 2K2
U6-A LM358HP
U6-B LM358HP
YLÄRAJAN REFERENSSIJÄNNITE
ALARAJAN REFERENSSIJÄNNITE
5V_PRIM
R66 470R
R69 470R
D6
D7
REF
R64 5K1
R68 5K1
C31 100N/50V
C30 100N/50V
C11 100N/50V
C10 100N/50V
R123 2K4
R124 2K4
R65 5K
R67 5K
TRIMMERI YLÄRAJAN KALIBROINTIIN
TRIMMERI ALARAJAN KALIBROINTIIN
VBUS JÄNNITE NEULAPEDILTÄ (400V)
VBUS
R60 470K
R61 470K
R62 470K
R63 12K
C29 100N/50V
J13-1
J13-2
VBUS –JÄNNITTEEN MITTAUSKYTKENNÄN KÄYTTÖJÄNNITE
(Kytketään johdolla liittimeen J12)
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Date: 26.6.2007
Date: 10.4.2009
Type: R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:
File: R100_tester_300409.sch
Page: 4/8

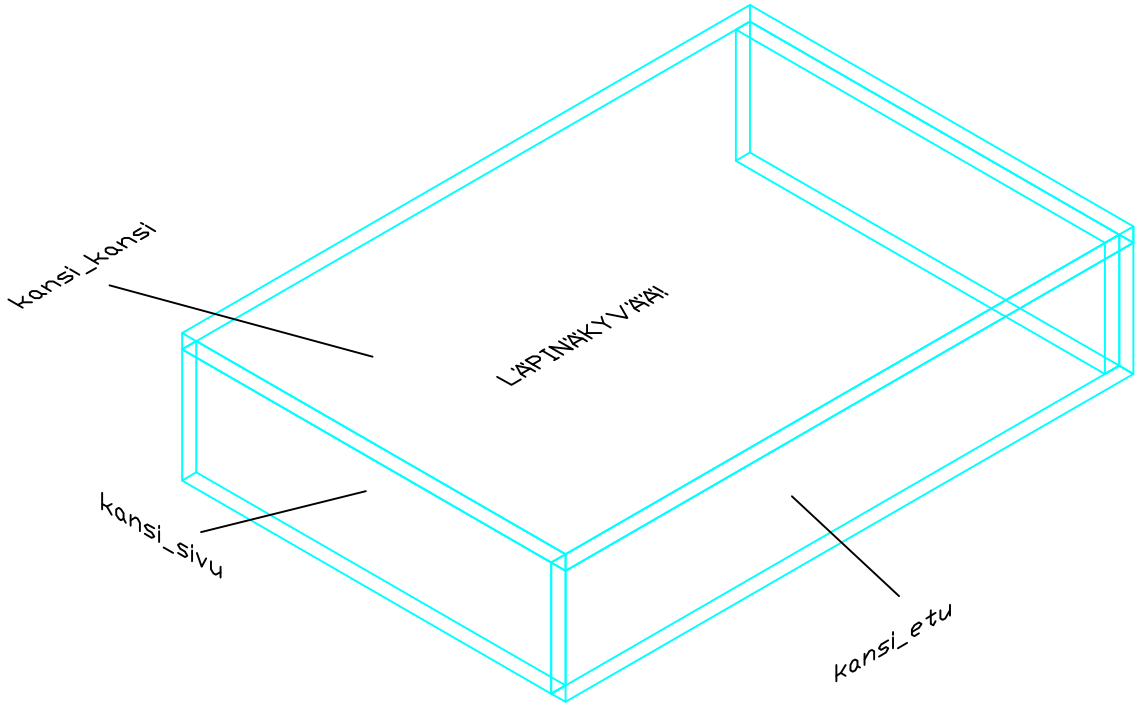

KORKEAJÄNNITTEEN MITTAUS
HV_REF SIGNAALIN PWM/DC -MUUNNOS
KORKEAJÄNNITEMITTAUKSIEN EROTUSPIIRI
SKAALAUS JA INVERTOINTI
AC -KOMPONENTIN MITTAUS
Lisätty galvaaninen erotus HV:n mittaukseen.
Ensimmäisessä protossa kytkentä on lisäkortilla
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Date: 26.6.2007
Date: 10.4.2009
Type: R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:
File: R100_tester_300409.sch
Page: 5/8

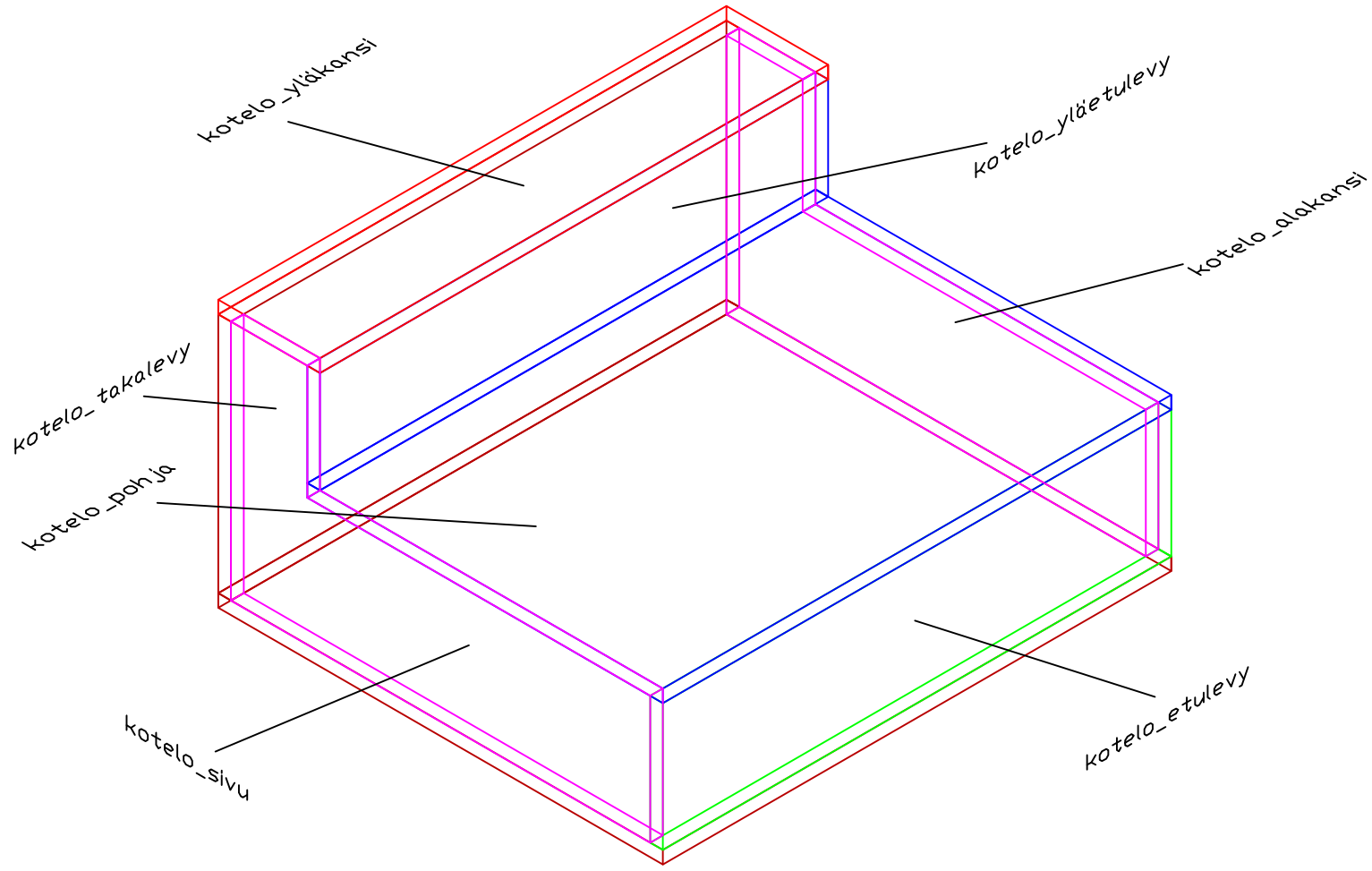

KUORMAT
PWM-SÄÄDETTÄVÄ VAKIOVIRTAKUORMA 55V LÄHDÖLLE
VASTUSKUORMAT 5V JA 12V LÄHDÖLLE
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Date: 26.6.2007
Date: 10.4.2009
Page: 6/8
Type: R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:
File: R100_tester_300409.sch
LM358HP
U2-A
HEATSINK
HS1
VIRTATIETO A/D-MUUNTIMELLE
VIRTAMITTAUS CURRENT SHUNT
R102 OR68
IRFP250
Q2
Q1
Q13
R110 1K
R109 1K
R108 1K
R105 100K
R104 100K
R107 100K
KUORMITUSVASTUKSET KYTKETÄÄN JOHDOILLA MITTAUSKORTILLE
PWM-SIGNAALI MIKRO-OHJAIMELTA
PWM1
55V
R1 100K
R103 100K
C1 1U/50V
C13 1U/50V
U1-D 4001BM
U1-C 4001BM
KUORMITUSVASTUS KYTKETÄÄN JOHDOILLA MITTAUSKORTILLE
IRLZ44
Q5
Q6
R27 100K
R28 100K
R23 10K
R26 10K
Q7
Q9
R22 100K
R25 100K
U1-B 4001BM
5V_IN
12V_IN
+15V
LOADS_ON


I/O-OHJAUSSIGNAALIT
I/O LAAJENNUS
INPUTIT
OUTPUTIT
OHJAUS- JA TESTISIGNAALIEN EROTETTU OHJAUS
STANDBY LEDIN JA PAINIKKEEN OHJAUS
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design: Hannu Kopsa
Update: Hannu Kopsa
Revision:
Date: 26.6.2007
Date: 10.4.2009
Page: 7/8
Type: R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:
File: R100_tester_100409.sch
PCF8574
I2C VÄYLÄ
SD_ERASE
TO J104 (overtemp test)
NEULAPEDILLE
STB_LED
ERASE
OVERTEMP_TEST
HVOK_TEST_A
HVOK_TEST_B
STANDBY
CHGND
+5V


LIITTIMET
LIITTIMET NEULAPEDILTÄ TULEVILLE JOHDOILLE
20 PINNINEN LATTAKAAPELI
J105-1
J105-2
J105-3
J105-4
J105-5
J105-6
J105-7
J105-9
J105-11
J105-13
J105-15
J105-17
J105-19
J105-20
SD_+12V
AGNDA
SD_-12V
SD_ERASE
SD_ERASE_ON
COVER ON
HVENA
RA6
SD_HVREF
SD_HVOK
SD_+12VD
GND12V
HVENA:lle LISÄTTY OHJAUS IO-PINNISTÄ PORTA.6
TUULETTIMEN LIITIN
J6-1
J5-1
J5-2
J6-2
FAN+
FAN-
LAMPUN LIITIN
J8-1
J7-1
J7-2
J8-2
LAMP+
LAMP-
VBUS JÄNNITE
J20-4
J20-1
J20-6
J20-3
VBUS
KORKEAJÄNNITE (1250V max)
J10-4
J10-1
J10-6
J10-3
HV
HV_GND
J32
J33
J34
55V
12V_IN
5V_IN
TESTATTAVAN LAITTEEN 55V LÄHTÖ
TESTATTAVAN LAITTEEN 12V LÄHTÖ
TESTATTAVAN LAITTEEN 5V LÄHTÖ
LIITTIMET OHJAINKORTILTA TULEVILLE LATTAKAAPELEILLE
PORTA
RA-1
RA-2
RA-3
RA-4
RA-5
RA-6
RA-7
RA-8
RA-9
RA-10
AN0
AN1
AN2
VREF
RA4
AN4
RA6
RA7
LAMP TEST LOAD
PORTB
RB-1
RB-2
RB-3
RB-4
RB-5
RB-6
RB-7
RB-8
RB-9
RB-10
INT
AN10
AN8
AN9
AN11
AN13
RB6
RB7
LCD_E
LCD_RS
PORTC
RC-1
RC-2
RC-3
RC-4
RC-5
RC-6
RC-7
RC-8
RC-9
RC-10
RC0
PWM2
HVREF PWM
PWM1
KUORMAN OHJAUS
SCL
SCK/SCL
SDA
SDI/SDA
RC5
SDO
RC6
RC7
PORTE
RE-1
RE-2
RE-3
RE-4
RE-5
RE-6
RE-7
RE-8
RE-9
RE-10
AN5
AN6
AN7
RE3
MITTAUSKORTIN KIINNITYSREIJÄT
J15
HOLE3GND
J16
HOLE3GND
J17
HOLE3GND
J18
HOLE3GND
MEG power
MEG POWER OY
Pengerkatu 2
15230 LAHTI
FINLAND
Design:
Hannu Kopsa
Update:
Hannu Kopsa
Revision:
File:
R100_tester_100409.sch
Date:
26.6.2007
Date:
10.4.2009
Page:
8/8
Type:
R100 TESTILAITTEEN MITTAUSKORTTI
Art.Nr.:

LIITE 3


kansi_kansi
kansi_sivu
kansi_etu


kotelo_yläkansi
kotelo_yläetulevy
kotelo_alakansi
kotelo_takalevy
kotelo_pohja
kotelo_sivu
kotelo_etulevy

```c
/*****************************************
 * FILE         : main.c                 *
 * PROJECT      : 857728 tester          *
 * DATE         : 31.8.2007              *
 * AUTHOR       : Hannu Kopsa            *
 * ***************************************
 *                                       *
 * main function for 857728 power supply *
 * tester                                *
 *                                       *
 *****************************************/

#include <system.h>
#include "init.h"

void initializations( void );

extern void start_testing(void);
extern void lcd_init(void);
extern void adc_init(void);
extern void i2c_init(void);
extern void PWM_init(void);



void main(void)
{
        // **********   INITIALIZATIONS   ************

        init();                         // auto generated initialization code (HI-TIDE wizard)
        lcd_init();                     // initialize lcd-display
        adc_init();                     // initialize analog to digital converter
        PWM_init();                     // initialize pulse width modulation unit
        i2c_init();                     // initialize i2c serial bus unit

        start_testing();                // start the test program

}

//      END OF FILE

```

```
/*****************************************
 * FILE         : init.c                 *
 * PROJECT      : 857728 tester          *
 * DATE         : 31.8.2007              *
 * AUTHOR       : Hannu Kopsa            *
 * COMPILER     : HI-TECH PIC C          *
 * ***************************************
 *                                       *
 * Auto generated initialization code    *
 *                                       *
 *****************************************/

#include <system.h>

/* Program device configuration word
 * Oscillator = Internal RC Clockout
 * Watchdog Timer = Off
 * Power Up Timer = Off
 * Master Clear Enable = /MCLR is external
 * Code Protect = Off
 * Data EE Read Protect = Off
 * Brown Out Detect = BOD and SBOREN disabled
 * Internal External Switch Over Mode = Disabled
 * Monitor Clock Fail-safe = Disabled
 * Low Voltage Program = Disabled
 */
#pragma DATA _CONFIG1,_DEBUG_OFF & _INTRC_OSC_NOCLKOUT & _WDT_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOR_OFF & _IESO_OFF & _FCMEN_OFF & _LVP_OFF;

/* Program second configuration word
 * Self Write Enable = disable
 * Brown Out Reset Sel Bit = Brown out at 2.1V
 */
#pragma DATA _CONFIG2,_WRT_OFF & _BOR21V;

//Set clock frequency
#pragma CLOCK_FREQ      8000000

// Peripheral initialization function
void init(void){
        /***** Common Code ****
         *  Timer 1 interrupt disabled.
         *  Timer 2 interrupt disabled.
         *  CCP Module 1 interrupt disabled.
         *  Usart TX interrupt disabled.
         *  Usart RX interrupt disabled.
         */
        pie1    = 0b00000000;
        /*
         *  Portbit7:4 interrupt-on-change disabled
         *  Timer 0 interrupt enabled.
         *  Global interrupt disabled during initialization
         */
        intcon  = 0b00100000;
        /*
         *  CCP Module 2 interrupt disabled.
         *  EEPROM interrupt disabled.
         */
        pie2    = 0b00000000;
        /*
         *  Timer 0 is prescaled by 1:8
         *  Timer 0 is clocked internally.
         *  Weak pullup on PORT disabled
         */
        option_reg      = 0b10000010;
        /*
         *  Port directions: 1=input, 0=output
         */
        trisc   = 0b11111000;

        /***** 16F887 Code ****
         *  Internal oscillator set to 8MHz
         */
        osccon  = 0b01110000;

        /***** CCP Module 1 Code ****
         *  CCP Module 1 disabled
         *  CCP Module 1 in PWM mode
         *  Output cleared on match
         */
        ccp1con = 0b00101001;
        /*
         *  DutyCycle set to 0.125 uSec
         */
        ccpr1l  = 0b00000010;

        /***** CCP Module 2 Code ****
         *  CCP Module 2 disabled
```

```
 88          *  CCP Module 2 in PWM mode
 89          *  Output cleared on match
 90          */
 91         ccp2con = 0b00101001;
 92         /*
 93          *  DutyCycle set to 0.125 uSec
 94          */
 95         ccpr2l  = 0b00000010;
 96
 97         /***** PortA Code ****
 98          *  Port directions: 1=input, 0=output
 99          */
100         trisa   = 0b11111111;
101         clear_bit(trisa,4);
102         clear_bit(trisa,6);
103
104         /***** PortB Code ****
105          *  Port directions: 1=input, 0=output
106          */
107         trisb   = 0b00111111;
108
109         /***** PortD Code ****
110          *  Port directions: 1=input, 0=output
111          */
112         trisd   = 0b00000000;
113
114         /***** PortE Code ****
115          *  Port directions: 1=input, 0=output
116          */
117         trise   = 0b00001111;
118
119         /***** Timer 1 Code ****
120          *  Timer is inactive
121          *  Timer derived from system clock
122          *  External clock input is not synchronized
123          *  T1 oscillator circuit (for external input) is inactive
124          *  Timer prescaler is 1:1
125          */
126         t1con   = 0b00000100;
127
128         /***** Timer 2 Code ****
129          *  Prescale ratio is 1:1
130          *  Timer 2 is suspended
131          *  Postscale ratio set to 1:1
132          */
133         t2con   = 0b00000000;
134         /*
135          *  Period register set to 0xFF
136          */
137         pr2     = 0b11111111;
138
139         /***** Usart Code ****
140          *  Usart in Synchronous mode
141          *  Usart in transmission mode
142          *  TX in eight bit format
143          *  Usart in synchronous slave mode
144          */
145         txsta   = 0b00110000;
146         /*
147          *  RX in eight bit format
148          *  Usart module disabled
149          */
150         rcsta   = 0b00000000;
151         /*
152          *  Baud rate is 110
153          */
154         spbrg   = 0b00000000;
155
156         /****************************
157          *  EXTERNAL INTERRUPT CODE *
158          * *************************/
159
160         //Enable interrupt
161         set_bit(intcon, INTE);
162         set_bit(option_reg, INTEDG);
163
164         set_bit(intcon,GIE);// Global interrupts enabled
165
166 }
167
168 //      END OF FILE
169
```

```
/*****************************************
 * FILE         : lcd_driver.c           *
 * PROJECT      : 857728 tester          *
 * DATE         : 21.3.2009              *
 * AUTHOR       : Hannu Kopsa            *
 * COMPILER     : BOOST C                *
 * ***************************************
 * Driver file of HD44780-based LCD display.*
 * This driver works with 4*20 displays.    *
 * Modified from Andy Kunz's LCD drivers.   *
 *****************************************/

#include <system.h>
#include "lcd_driver.h"

const unsigned char const LCD_ROW_ADDRESS[] =   // Row/Column information for lcd_gotoxy()
{
        0x00,
        0x40,
        0x14,
        0x54
};

void lcd_init (void)                            // Reset display from software
{

        clear_bit(portb,LCD_E);                 // Set up control pin I/O
        clear_bit(trisb,LCD_TRIS_E);
        clear_bit(portc,LCD_RW);                // Write mode
        clear_bit(trisc,LCD_TRIS_RW);
        clear_bit(portb,LCD_RS);                // Command mode
        clear_bit(trisb,LCD_TRIS_RS);

        LCD_TRIS_PORT &= ~LCD_TRIS_DATAMASK;    // Set data bus to output

        clear_bit(portb,LCD_E);                 // Start talking to LCD
        delay_ms (200);                         // Wait a little while

        lcd_putnybble (0b0010);                 // Select 4-bit mode
        delay_ms (10);                          // Spec calls for 4.1 mS
        lcd_putnybble (0b0010);                 // Do it again
        delay_ms (10);
        lcd_putnybble (0b0010);
        delay_ms (10);
        lcd_putnybble (0b0010);
        delay_ms (10);

        lcd_putbyte(0b00101000);                //function set
        lcd_putbyte(0b00101000);                //function set
        lcd_putbyte(0b00101000);                //function set
        lcd_putbyte(0b00101000);                //function set

        lcd_putbyte(0b00001000);                //display off
        lcd_putbyte(0b00000001);                //display clear
        lcd_putbyte(0b00000110);                //entry mode set
        lcd_putbyte(0b00001100);                //display on

}


void lcd_putbyte (unsigned char c)              // Write byte to port in current RS mode
{
        unsigned char   RS_Status;

        RS_Status = portb.LCD_RS;               // Get old pin state
        clear_bit(portb,LCD_RS);                // Force into command mode to read state
        delay_ms(2);
        if (RS_Status)
                set_bit(portb,LCD_RS);          // Restore RS to old state
        asm                                     //delay
        {
                nop
                nop
        }
        clear_bit(portc,LCD_RW);                // Set to write mode
        asm                                     //delay
        {
                nop
                nop
        }
        lcd_putnybble (c >> 4);                 // Send the character out
        lcd_putnybble (c);
        set_bit(portb,LCD_E);
        asm                                     //delay
        {
                nop
                nop
        }
```

```
 89         clear_bit(portb,LCD_E);
 90 }
 91
 92 void lcd_command (unsigned char c)             // Send command to LCD port
 93 {
 94         clear_bit(portb,LCD_RS);
 95         lcd_putbyte (c);
 96 }
 97
 98 void lcd_putnybble (unsigned char c)           // Write nybble to port in current RS mode
 99 {
100         c &= LCD_TRIS_DATAMASK;                // Remove any extraneous bits
101         LCD_DATA_PORT = (LCD_DATA_PORT & ~LCD_TRIS_DATAMASK) | c;       // Write data bits to port
102         asm                                    // delay
103         {
104                 nop
105                 nop
106         }
107         set_bit(portb,LCD_E);                  // Start to write it
108         asm                                    // delay
109         {
110                 nop
111                 nop
112         }
113         clear_bit(portb,LCD_E);                // Finish write cycle
114 }
115
116 void lcd_printf (const char* message)          // Write message to LCD (C string type)
117         {
118         while (*message)                       //      Look for end of string
119                 lcd_putc (*message++);         //      Show and bump
120         }
121
122 void lcd_clear (void)                          // Clear LCD screen
123         {
124         lcd_command (LCD_COMMAND_CLEAR);
125         delay_ms(20);
126         }
127
128 void lcd_putc (unsigned char c)                // Write character to LCD
129         {
130         unsigned char   CursAddr;
131         switch (c)
132                 {
133                 case '\f':                     // Form Feed (clear screen)?
134                         lcd_command (LCD_COMMAND_CLEAR); // Erase screen
135                         lcd_gotoxy (1,1);      // Position cursor to top of screen
136                         break;
137                 case 0xE4:                     // ä?
138                         set_bit(portb,LCD_RS);
139                         lcd_putbyte (0xE1);
140                         break;
141                 case 0xC4:                     // Ä?
142                         set_bit(portb,LCD_RS);
143                         lcd_putbyte (0xE1);
144                         break;
145                 case 0xF6:                     // ö?
146                         set_bit(portb,LCD_RS);
147                         lcd_putbyte (0xEF);
148                         break;
149                 case 0xD6:                     // Ö?
150                         set_bit(portb,LCD_RS);
151                         lcd_putbyte (0xEF);
152                         break;
153                 default:                       // Printable?
154                         set_bit(portb,LCD_RS);
155                         lcd_putbyte (c);       // Send character out
156
157                 }
158         }
159
160
161 unsigned char lcd_lineof (unsigned char CursorAddress)  // Calculate cursor row from it's address
162         {
163         CursorAddress &= 0x50;                 //      Strips out uniquely the address bits
164         switch (CursorAddress)
165                 {
166                 case 0x00:                     // Note - this handles all cases except for some
167                         CursorAddress = 1;     //        of those unsupported displays listed in
168                 case 0x40:                     //        lcd.h file.
169                         CursorAddress = 2;
170                 case 0x10:
171                         CursorAddress = 3;
172                 case 0x50:
173                         CursorAddress = 4;
174                 default:
175                         CursorAddress = 1;
176                 }
```

```
177         return (CursorAddress);
178         }
179
180 void lcd_gotoxy (unsigned char row, unsigned char col)  // Position cursor
181 {
182         if (row > LCD_MAXROWS)                  // Range limit
183                 row = LCD_MAXROWS;
184         if (col > LCD_MAXCOLS)
185                 col = LCD_MAXCOLS;
186         row = LCD_ROW_ADDRESS[row-1];           // Get address of first byte on desired row
187         row += col - 1;
188
189         lcd_command (0x80 | row);               // Write new cursor address
190         }
191
192 void lcd_printdec(unsigned int x)               // Print a value in decimal to the LCD screen
193 {
194   unsigned int count;
195
196   if (x > 9999)
197   {
198         /* how many 10 thousands? */
199         count=0;
200         while(x > 9999) {
201                 x = x - 10000;
202                 count++;
203         }
204   lcd_putc('0' + count);
205   }
206
207   /* how many thousands? */
208   count=0;
209   while(x > 999) {
210     x = x - 1000;
211     count++;
212   }
213   lcd_putc('0' + count);
214
215   /* how many hundreds? */
216   count=0;
217   while(x > 99) {
218     x = x - 100;
219     count++;
220   }
221   lcd_putc('0' + count);
222
223   /* how many tens? */
224   count=0;
225   while(x > 9) {
226     x = x - 10;
227     count++;
228   }
229   lcd_putc('0' + count);
230
231   /* and finally units */
232   lcd_putc('0' + x);
233 }
234
235 // END OF FILE
236
```

```
/*****************************************
 * FILE         : adc.c                  *
 * PROJECT      : 857728 tester          *
 * DATE         : 21.9.2007              *
 * AUTHOR       : Hannu Kopsa            *
 * COMPILER     : BOOST C                *
 * ***************************************
 *                                       *
 * Driver code for analog to digital     *
 * converter.                            *
 *                                       *
 *****************************************/

#include "adc.h"
#include <system.h>

int ad_offset;

/***** ADC init *********************************
   ADC is enabled
   Enabled AD-channels: AN0, AN1, AN2, AN4, AN5, AN6, AN7, AN8, AN9, AN10, AN11, AN13
   AD clock = Fosc/32
 ***********************************************/

void adc_init (void)
{
        ansel  = 00001000;              // enable AD channels
        anselh = 00010000;
        adcon1 = 10010000;              // Right justified, Vref+, Vss
        adcon0 = 10000001;              // set clock, enable ADC
}

//****** AD conversion code *****

unsigned int Read_AD(unsigned char ad_channel)
{
        unsigned int result;
        unsigned int average;
        unsigned char i;

        adcon0 = (adcon0 & 0b11000011) | (ad_channel << 2);     // Set ad channel
        delay_us(6);                            // Acquisition delay

        adcon0 |= 0b00000010;                   // Start conversion
        while (adcon0 & 0b00000010);            // Is conversion done?
        average = adresh << 8;
        average = average | adresl;

        for(i=0; i < 63; i++)
        {
                adcon0 |= 0b00000010;           // Start conversion
                while (adcon0 & 0b00000010);    // Is conversion done?
                result = adresh << 8;
                result = result | adresl;
                average = average + result;
        }

        return average / 16;                    // return ad value (millivolts)
}

//      END OF FILE

```

```
1  /*****************************************
2   * FILE          : beeb.c                *
3   * PROJECT       : 857728 tester         *
4   * DATE          : 21.3.2009             *
5   * AUTHOR        : Hannu Kopsa           *
6   * COMPILER      : BOOST C               *
7   * ***************************************
8   *                                       *
9   * Driver software for BUZZER            *
10  * Buzzer is connected to RC0 pin        *
11  *                                       *
12  *****************************************/
13
14 #include "beeb.h"
15
16 void beeb(unsigned char repeat, unsigned int delay)
17 {
18         while(repeat > 0)
19         {
20                 set_bit(portc,0);       // buzzer on
21                 delay_ms(delay);
22                 clear_bit(portc,0);     // buzzer off
23                 delay_ms(delay);
24                 --repeat;
25         }
26 }
27
28 // END OF FILE
29
```

```
/*********************************************
 * FILE         : IO_exp.c                   *
 * PROJECT      : 857728 tester              *
 * DATE         : 15.11.2007                 *
 * AUTHOR       : Hannu Kopsa                *
 * COMPILER     : BOOST C                    *
 *********************************************
 * DRIVER CODE FOR PHILIPS PCF8574 I/O EXPANDER *
 *                                           *
 * Modified from Michael Alon's I2C lib file    *
 *                                           *
 *********************************************/

// i2c_init     - initialize I2C functions
// i2c_start    - issue Start condition
// i2c_repStart - issue Repeated Start condition
// i2c_stop     - issue Stop condition
// i2c_read(x)  - receive unsigned char - x=0, don't acknowledge - x=1, acknowledge
// i2c_write    - write unsigned char - returns ACK

#include <system.h>
#include "IO_exp.h"
#include "delay.h"

void i2c_init(void);
void i2c_waitForIdle(void);
void i2c_start(void);
void i2c_repStart(void);
void i2c_stop(void);

void write_io_exp(unsigned char data)
 {
   i2c_start();
   i2c_write(ADDR_OUT);
   i2c_write(data);
   i2c_stop();
   delay_ms(11);
}

/*****************************************************************************************/

unsigned char read_io_exp(void)
{
   unsigned char data;
   i2c_waitForIdle();
   i2c_start();
   i2c_waitForIdle();
   i2c_write(ADDR_IN);
   i2c_waitForIdle();
   set_bit(sspcon2,RCEN);
   while(sspcon2.RCEN);
   data = sspbuf;
   clear_bit(sspcon2,ACKDT);
   i2c_waitForIdle();
   i2c_stop();

   return(data);
}

/*****************************************************************************************/

void i2c_init()
{
        set_bit(trisc,3);               // set SCL and SDA pins as inputs
        set_bit(trisc,4);
        portc  = 0;
        sspadd = 20;                    // 100k at 8Mhz clock
        set_bit(sspstat,SMP);           // standard speed mode
        sspcon = 0x8;                   // I2C mode
        sspcon2= 0;
        set_bit(sspcon,SSPEN);          // enable ssp
}

/*****************************************************************************************/

void i2c_waitForIdle()
{
while (( sspcon2 & 0x1F ) | sspstat.R_W ); // wait for idle and not writing
}

/*****************************************************************************************/

void i2c_start()
{
 i2c_waitForIdle();
 set_bit(sspcon2,SEN);
}

```

```
 89 /*****************************************************************************************/
 90
 91 void i2c_stop()
 92 {
 93  i2c_waitForIdle();
 94  set_bit(sspcon2,PEN);
 95 }
 96
 97 /*****************************************************************************************/
 98
 99 unsigned char i2c_write( unsigned char i2cWriteData )
100 {
101  i2c_waitForIdle();
102  sspbuf = i2cWriteData;
103
104  return ( ! sspcon2.ACKSTAT  );     // function returns '1' if transmission is acknowledged
105 }
106
107 // END OF FILE
108
```

```
/*****************************************
 * FILE         : pwm.c                  *
 * PROJECT      : 857728 tester          *
 * DATE         : 12.9.2007              *
 * AUTHOR       : Hannu Kopsa            *
 * ***************************************
 *                                       *
 * main function for 857728 power supply *
 * tester                                *
 *                                       *
 *****************************************/

#include "pwm.h"
#include <system.h>
#include "global.h"

void PWM_init (void)
{

        // Timer2 init

        t2con = 0b00000100;             // Prescaler = 1, Tmer2 is on.
        pr2 = 0xFF;                     // Set Timer2 period.


        // CCP1 init...

        ccp1con = 0b00001100;           // Set CPP1 control register. PWM mode, Single output.
        ccpr1l  = 0;                    // Duty cycle is 0


        // CCP2 init...

        ccp2con = 0b00001100;           // Set CPP2 control register. PWM mode, Single output.
        ccpr2l  = 0;                    // Duty cycle is 0

}


void Set_PWM1_Duty_Cycle(unsigned char dutycycle)
{

        ccpr1l  = dutycycle;            // Set duty cycle.

}


void Set_PWM2_Duty_Cycle(unsigned char dutycycle)
{

        ccpr2l  = dutycycle;            // Set duty cycle.

}

```

```
/*************************************
 * FILE         : isr.c                *
 * PROJECT      : 857728 tester        *
 * DATE         : 16.10.2007           *
 * AUTHOR       : Hannu Kopsa          *
 * COMPILER     : BOOST C              *
 * ***********************************
 *                                     *
 * Interrupt service routines          *
 *                                     *
 ***************************************/

#include <system.h>

char delay_ms_flag;
char EXT_INT;

void interrupt (void)
{

        /***** Timer 0 interrupt code *****/
        /* used in delays */
        if( intcon & (1<<T0IF) )
        {
                delay_ms_flag = 1;              // set millisecond flag

                clear_bit( intcon, T0IF );      // clear event flag
        }


        /***** External interrupt code *****/
        /* "5V low" comparator connected to external interrupt pin RB0 */
        /* makes interrupt if 5v-line goes too low */
        if( intcon & (1<<INTF) )
        {
                EXT_INT = 1;                    // set external interrupt flag

                clear_bit( intcon, INTF );      // clear event flag
        }

}

// END OF FILE

```

```
/*****************************************
 * FILE         : test_program.c         *
 * PROJECT      : 857728 tester          *
 * DATE         : 14.3.2009              *
 * AUTHOR       : Hannu Kopsa            *
 * COMPILER     : BoostC                 *
 * ***************************************
 *                                       *
 * TEST_PROGRAM FOR 857728 POWER SUPPLY  *
 *                                       *
 *****************************************/

// MICROCONTROLLER:     PIC16f887

#include <system.h>
#include "test_program.h"

void start_testing(void)
{

        //testiohjelma alkaa

        lcd_gotoxy(1,5);
        lcd_printf("MEG POWER OY");
        lcd_gotoxy(3,3);
        lcd_printf("POWERIN 857728");
        lcd_gotoxy(4,4);
        lcd_printf("TESTAUSLAITE");
        delay_s(2);

        while(1)
        {
                r_button_rising_edge = false;
                g_button_rising_edge = false;
                b_button_rising_edge = false;
                stop_test            = false;
                error_code           = 0;
                hv_ripple_error      = false;


// Asetetaan laitteisto alkutilaan

                main_power_on(false);                   // sähköt pois!!
                write_io_exp(io_exp_stat);              // i/o expander init
                leds(1,1,1);                            // switch on push buttons leds
                start_button(false);                    // switch off emulated start button
                erase(false);                           // disable erase signal
                hv_ok_test_a(false);                    // disable hv_ok signal tests
                hv_ok_test_b(false);                    //
                loads_on(false);                        // loads off
                lamp_test_loads_on(false);              // 5V, 1.3A and 12V, 0.5A loads off.
                set_55v_load(0);                        // current is 0A
                hv_enable(false);                       // hv_enable is disabled
                set_hvref(0);                           // HV- reference voltage = 0V

// ALKUVALIKKO

                lcd_clear();
                lcd_gotoxy(1,2);
                lcd_printf("ALOITA TESTI");
                lcd_gotoxy(2,2);
                lcd_printf("PAINAMALLA A-");
                lcd_gotoxy(3,2);
                lcd_printf("PAINIKETTA");

// Helppi poistettu käytöstä ohjelmamuistin loputtua

//              lcd_gotoxy(4,10);
//              lcd_printf("C = HELP");
                cover_lock_disable(true);                                       // Kansi auki!
                while(!r_button_rising_edge && !g_button_rising_edge && !b_button_rising_edge)  // Odotetaan napin painallusta
                {
                        read_buttons();
                }

                if(b_button_rising_edge)                                        // jos painettiin sinistä nappia
                {                                                               // aloitetaan testiohjelma
// APUPOWERIN KÄYNNISTYS JA STANDBY LEDI

                        lcd_clear();
                        b_button_rising_edge = 0;
                        cover_lock_disable(false);
                        lcd_gotoxy(1,3);
                        lcd_printf("** KÄYNNISTYS **");
                        set_main_voltage(40);
                        if (!stop_test)                                         //jos ei keskeytetty, jatketaan eteenpäin...
                        {
                                main_power_on(true);                            //sähköt päälle!!
```

```
89                              delay_s(1);
90                              if(stb_led())                                   //jos ledi syttyi...
91                              {
92                                      main_power_on(false);                   //sähköt pois
93                                      lcd_clear();
94                                      lcd_printf("APUPOWERI OK!");
95                                      set_main_voltage(120);
96                              }
97                              else                                            //jos ei syttynyt... fail!!!
98                              {
99                                      error_code = 1;                         // ledi V1 ei syty, kun variakki 40V
                                        (apupoweri ei käynnisty?)
100                                     stop_test = true;
101                             }
102                     }
103                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan eteenpäin...
104                     {
105
106 // LÄMPÖSUOJAUKSEN JA START -SIGNAALIN TOIMINTA
107
108                             main_power_on(true);
109                             delay_s(2);
110                             lcd_clear();
111                             lcd_printf("OVERTEMP TEST...");
112                             start_button(true);                             //"painetaan" START kytkin pohjaan
113                             delay_s(2);
114                             start_button(false);                            //"vapautetaan" START kytkin
115                             if (!stb_led())                                 //sammuiko ledi?
116                             {                                               //jos sammui (hyvä!),...
117                                     overtemp_test(true);                    //...tehdään lämpösuojatesti
118                                     delay_s(1);
119                                     overtemp_test(false);
120                                     if(!stb_led())                          //jos ledi ei syttynyt (poweri ei sammunut)...
121                                     {
122                                             stop_test = true;               //lopetetaaan testaus
123                                             error_code = 2;                 //Ylilämpösuojaus ei toiminut, kun
                                                oikosuljettiin J19
124                                     }
125                             }
126                             else
127                             {
128                                     stop_test = true;                       //lopetetaan testaus
129                                     error_code = 3;                         //Poweri ei käynnisty (ledi V1 ei sammu), kun
                                        painetaan START-kytkintä
130                             }
131
132                     }
133                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan eteenpäin...
134                     {
135
136 // VBUS -JÄNNITTEEN SÄÄTÖ
137
138                             lcd_clear();
139                             lcd_gotoxy(1,1);
140                             lcd_printf("Säädä VBUS (400V)");
141                             lcd_gotoxy(2,1);
142                             lcd_printf("trimmerillä R212");
143
144                             main_power_on(true);
145                             delay_s(1);
146                             start_button(true);                             //"painetaan" START kytkin pohjaan
147                             delay_s(1);
148                             start_button(false);                            //"vapautetaan" START kytkin
149
150                             r_button_rising_edge = false;
151                             while(1)
152                             {
153                                     b_button_rising_edge = false;
154                                     read_buttons();
155
156                                     //HUOM! ylä- ja alaraja säädetään testilaitteen trimmereillä R65 ja R67.
157                                     //Niihin ei siis voi vaikuttaa ohjelmallisesti.
158
159                                     if (vbus_under())                       //jos VBUS < alaraja (esim. 398V ?)
160                                     {
161                                             lcd_gotoxy(3,3);
162                                             lcd_printf("VBUS LIIAN PIENI");
163                                             lcd_gotoxy(4,1);
164                                             lcd_printf("pyöritä myötäpäivään");
165                                     }
166                                     else if (vbus_over())                   //jos VBUS > yläraja (esim. 402V ?)
167                                     {
168                                             lcd_gotoxy(3,3);
169                                             lcd_printf("VBUS LIIAN SUURI");
170                                             lcd_gotoxy(4,1);
171                                             lcd_printf("pyöritä vastapäivään");
172                                     }
173                                     else
```

```
174                                     {
175                                             lcd_gotoxy(3,3);
176                                             lcd_printf("     VBUS OK!     ");
177                                             lcd_gotoxy(4,1);
178                                             lcd_printf("            PAINA A ");
179                                             if(b_button_rising_edge)
180                                             {
181                                                     beeb(1,50);
182                                                     break;
183                                             }
184                                     }
185                                     if(r_button_rising_edge)                //jos painettiin punaista...
186                                     {
187                                             beeb(1,500);
188                                             stop_test = true;               //keskeytetään testaus, ja mennään alkuun
189                                             error_code = 4;                 // VBUS jännitettä ei säädetty/ei voitu säätää
190                                             break;
191                                     }
192                             }
193                     }
194                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan eteenpäin...
195                     {
196
197 // OIKOSULKUPALAN LISÄÄMINEN
198
199                             main_power_on(false);                           // sähköt pois kannen aukaisun ajaksi
200                             lcd_clear();
201                             lcd_gotoxy(2,1);
202                             lcd_printf("Laita oikosulkupala");
203                             lcd_gotoxy(3,1);
204                             lcd_printf("X1 ja paina A");
205                             cover_lock_disable(true);
206                             delay_s(1);
207                             cover_lock_disable(false);
208                             b_button_rising_edge = false;
209                             while(!b_button_rising_edge)
210                             {
211                                     read_buttons();
212                             }
213                             beeb(1,50);
214                     }
215                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan eteenpäin...
216                     {
217
218 // PÄÄHAKKURIN KÄYNNISTYSJÄNNITE
219
220                             u_start_ok = false;
221                             while(!u_start_ok && !stop_test)
222                             {
223                                     lcd_clear();
224                                     main_power_on(false);
225                                     set_main_voltage(70);
226                                     lcd_clear();
227                                     main_power_on(true);
228                                     delay_s(2);
229                                     start_button(true);                     // "painetaan START -nappia"
230                                     delay_s(2);
231                                     start_button(false);
232                                     lcd_gotoxy(1,1);
233                                     lcd_printf("Säädä variakki");
234                                     lcd_gotoxy(2,1);
235                                     lcd_printf("HITAASTI! 100V");
236                                     while(1)
237                                     {
238                                             lcd_gotoxy(3,1);
239                                             lcd_printf("Jännite on nyt ");
240                                             u_variakki_v = read_main_voltage();     // read main voltage
241                                             lcd_printdec(u_variakki_v);
242                                             lcd_putc('V');
243                                             lcd_gotoxy(4,1);
244                                             read_buttons();
245
246                                             if (read_ad_5v() > 4000)
247                                             {
248                                                     beeb(1,200);
249                                                     lcd_clear();
250                                                     lcd_gotoxy(1,5);
251                                                     lcd_printf("STOP!!");
252                                                     lcd_gotoxy(3,1);
253                                                     lcd_printf("Käynnistysjännite on");
254                                                     lcd_gotoxy(4,1);
255                                                     lcd_printdec(u_variakki_v);
256                                                     if(u_variakki_v < 86)
257                                                     {
258                                                             lcd_printf("V. Se on OK!");
259                                                             u_start_ok = true;
260                                                             delay_s(3);
261                                                             break;
```

```
262                                             }
263                                             else
264                                             {
265                                                     lcd_printf("V. EI HYVÄ!!!");
266                                                     delay_s(3);
267                                                     break;
268                                             }
269                                     }
270                                     if(r_button_rising_edge)                //jos painettiin punaista...
271                                     {
272                                             beeb(1,500);
273                                             stop_test = true;               //keskeytetään testaus, ja mennään
                                                alkuun
274                                             error_code = 5;                 // Käynnistysjännite liian suuri!!
275                                             break;
276                                     }
277                                     delay_ms(100);
278                             }
279                     }
280             }
281             if (!stop_test)         //jos ei keskeytetty, jatketaan eteenpäin...
282             {
283
284 // 5V LÄHDÖN KARKEA SÄÄTÖ
285
286                     lcd_clear();
287                     r_button_rising_edge = false;
288                     while(1)
289                     {
290                             read_buttons();
291                             lcd_gotoxy(1,1);
292                             lcd_printf("Säädä trimmeriä R211");
293                             lcd_gotoxy(2,1);
294                             u_5v_mv = read_ad_5v();
295                             lcd_printdec(u_5v_mv);
296                             lcd_printf("  -> ->  5000");
297                             if(u_5v_mv < 4980)                              // jos 5V:n lähdön jännite on alle
                                4,99V:n...
298                             {
299                                     lcd_gotoxy(3,3);
300                                     lcd_printf("MYÖTÄPÄIVÄÄN");
301                             }
302                             else if(u_5v_mv > 5020)                         // jos 5V:n lähdön jännite on yli
                                5.01V...
303                             {
304                                     lcd_gotoxy(3,3);
305                                     lcd_printf("VASTAPÄIVÄÄN");
306                             }
307                             else                                            // jos 5V:n lähtöjännite on välillä
                                4.99V ja 5.01V (se olisi tarkoitus)...
308                             {
309                                     {
310                                             lcd_gotoxy(3,1);
311                                             lcd_printf("5V:n lähtöjännite OK");
312                                             beeb(1,1000);
313                                             break;
314                                     }
315                             }
316                             if(r_button_rising_edge)                        // Jos painetaan punaista nappia...
317                             {
318                                     stop_test = true;                       // keskeytetään testaus, ja mennään
                                        alkuun
319                                     if(error_code == 0)
320                                     error_code = 30;                        // Käyttäjä keskeytti 5V:n jännitteen
                                        säätämisen trimmerillä 211. Jännitettä ei ehkä voitu säätää kohdalleen.
321                                     break;
322                             }
323                             delay_ms(200);                                  // hidastetaan vähän luupissa
                                kiertovauhtia
324                     }
325             }
326             if (!stop_test)         //jos ei keskeytetty, jatketaan eteenpäin...
327             {
328
329 // LÄHTÖJÄNNITTEIDEN TARKISTUS
330
331                     main_power_on(false);
332                     lcd_clear();
333                     set_main_voltage(100);
334                     main_power_on(true);
335                     delay_s(2);
336                     start_button(true);                                     //"painetaan" START kytkin pohjaan
337                     delay_s(2);
338                     start_button(false);                                    //"vapautetaan" START kytkin
339
340                     // lähtöjännitteiden tarkistus
341                     u_5v_mv = read_ad_5v();                                 // mitataan 5V
342                     u_12v_mv = read_ad_12v();                               // mitataan 12V
```

```
343                             u_55v_mv = read_ad_55v();                               // mitataan 55V
344                             u_miinus12v_mv = read_ad_miinus12v();                   // mitataan -12V
345                             u_plus12v_mv = read_ad_plus12v();                       // mitataan +12V
346
347                             if( u_5v_mv > 5500 )
348                             {
349                                     stop_test = true;
350                                     error_code =6;                                  // 5V yli 5,5V (ilman kuormaa)
351                             }
352                             if( u_5v_mv < 4500 )
353                             {
354                                     stop_test = true;
355                                     error_code =7;                                  // 5V alle 4,5V (ilman kuormaa)
356                             }
357                             if( u_12v_mv > 15000 )
358                             {
359                                     stop_test = true;
360                                     error_code =8;                                  // 12V yli 15V (ilman kuormaa)
361                             }
362                             if( u_12v_mv < 12000 )
363                             {
364                                     stop_test = true;
365                                     error_code =9;                                  // 12V alle 12V (ilman kuormaa)
366                             }
367                             if( u_55v_mv > 60000 )
368                             {
369                                     stop_test = true;
370                                     error_code =10;                                 // 55V yli 60V (ilman kuormaa)
371                             }
372                             if( u_55v_mv < 50000 )
373                             {
374                                     stop_test = true;
375                                     error_code =11;                                 // 55V alle 50V (ilman kuormaa)
376                             }
377                             if( u_miinus12v_mv > 12600 )
378                             {
379                                     stop_test = true;
380                                     error_code =12;                                 // -12V yli -12.15V (ilman kuormaa)
381                             }
382                             if( u_miinus12v_mv < 11400 )
383                             {
384                                     stop_test = true;
385                                     error_code =13;                                 // -12V alle -11.85V (ilman kuormaa)
386                             }
387                             if( u_miinus12v_mv > 12600 )
388                             {
389                                     stop_test = true;
390                                     error_code =14;                                 // +12V yli 12.15V (ilman kuormaa)
391                             }
392                             if( u_miinus12v_mv < 11400 )
393                             {
394                                     stop_test = true;
395                                     error_code =15;                                 // +12V alle 11.85V (ilman kuormaa)
396                             }
397                     }
398                     if (!stop_test)         //jos ei keskeytetty, jatketaan eteenpäin...
399                     {
400
401 // VIRTARAJOJEN SÄÄTÖ
402
403
404                             // Säädetään virtarajat lähelle maksimia, jotta poweri jaksaa syöttää tarpeeksi virtaa.
405
406                             lcd_clear();
407                             lcd_printf("Pyöritä trimmereitä");
408                             lcd_gotoxy(2,1);
409                             lcd_printf("R209 ja R244 noin");
410                             lcd_gotoxy(3,1);
411                             lcd_printf("10kierr myötäpäivään");
412                             lcd_gotoxy(4,1);
413                             lcd_printf("Paina A, kun valmis.");
414                             b_button_rising_edge = false;
415                             while(!b_button_rising_edge)
416                                     read_buttons();
417                     }
418                     if (!stop_test)         //jos ei keskeytetty, jatketaan eteenpäin...
419                     {
420
421 // PFC:n virtarajan säätö...
422
423                             lcd_clear();
424                             lcd_printf("   ODOTA HETKI...");
425                             loads_on(true);                                         // 5v ja 12V kuormat päälle
426                             set_55v_load(4050);                                     // 55v kuorma 4,2A (tällä virralla
                                virtarajan pitäisi ylittyä ja lähtöjännitteiden tippua)
427                             lcd_clear();
428                             r_button_rising_edge = false;
429                             while(1)
```

```
430                             {
431                                     read_buttons();
432                                     lcd_gotoxy(1,1);
433                                     lcd_printf("Säädä trimmeriä R244");
434                                     lcd_gotoxy(2,1);
435                                     u_5v_mv = read_ad_5v();
436                                     u_5v_mv += 75;                          // lisätään johtimien resistanssista
                                        johtuva jännitehäviö 75mV
437                                     lcd_printdec(u_5v_mv);
438                                     lcd_printf("  -> ->  4800");
439                                     if(u_5v_mv < 4790)                      // jos 5V:n lähdön jännite on tippunut
                                        alle 4,795V:n...
440                                     {
441                                             lcd_gotoxy(3,3);
442                                             lcd_printf("MYÖTÄPÄIVÄÄN");
443                                     }
444                                     else if(u_5v_mv > 4810)                 // jos 5V:n lähdön jännite on yli
                                        4,805V...
445                                     {
446                                             lcd_gotoxy(3,3);
447                                             lcd_printf("VASTAPÄIVÄÄN");
448                                     }
449                                     else                                    // jos 5V:n lähtöjännite on välillä
                                        4,4V ja 4,6V (se olisi tarkoitus)...
450                                     {
451                                             {
452                                                     lcd_gotoxy(3,2);
453                                                     lcd_printf("Virtaraja OK!!!!!!");
454                                                     lcd_gotoxy(4,2);
455                                                     lcd_printf("   ODOTA HETKI...");
456                                                     beeb(1,1000);
457                                                     break;
458                                             }
459                                     }
460                                     if(r_button_rising_edge)                // Jos painetaan punaista nappia...
461                                     {
462                                             stop_test = true;               // keskeytetään testaus, ja mennään
                                                alkuun
463                                             error_code = 16;                // Käyttäjä keskeytti PFC:n virtarajan
                                                säätämisen. Virtarajaa ei ehkä voitu säätää kohdalleen.
464                                             loads_on(false);                // 5v ja 12V kuormat pois päältä
465                                             set_55v_load(0);                // 55v kuorma nollaan
466                                             beeb(1,1000);
467                                             break;
468                                     }
469                                     delay_ms(200);                          // hidastetaan vähän luupissa
                                        kiertovauhtia
470                             }
471                     }
472                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan
                        eteenpäin...
473                     {
474 // Päähakkurin virtarajan säätöä varten >150VAC
475
476                             lcd_clear();
477                             set_main_voltage(150);
478                     }
479                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan
                        eteenpäin...
480                     {
481
482 // PÄÄHAKKURIN VIRTARAJAN SÄÄTÖ
483
484                             lcd_clear();
485                             lcd_printf("ODOTA HETKI...");
486                             r_button_rising_edge = false;
487                             set_55v_load(4000);                             // 55v kuorma 4,0A (tällä virralla
                                virtarajan pitäisi ylittyä ja lähtöjännitteiden tippua)
488                             while(1)
489                             {
490                                     read_buttons();
491                                     lcd_gotoxy(1,1);
492                                     lcd_printf("Säädä trimmeriä R209");
493                                     lcd_gotoxy(2,1);
494                                     u_5v_mv = read_ad_5v();
495                                     u_5v_mv += 75;                          // lisätään johtimien resistanssista
                                        johtuva jännitehäviö 75mV
496                                     lcd_printdec(u_5v_mv);
497                                     lcd_printf("  -> ->  4800");
498                                     if(u_5v_mv < 4790)                      // jos 5V:n lähdön jännite on tippunut
                                        alle 4,795V:n...
499                                     {
500                                             lcd_gotoxy(3,3);
501                                             lcd_printf("MYÖTÄPÄIVÄÄN");
502                                     }
503                                     else if(u_5v_mv > 4810)                 // jos 5V:n lähdön jännite on yli
                                        4,805V...
504                                     {
```

```
505                                             lcd_gotoxy(3,3);
506                                             lcd_printf("VASTAPÄIVÄÄN");
507                                     }
508                                     else                                    // jos 5V:n lähtöjännite on välillä
                                        4,4V ja 4,6V (se olisi tarkoitus)...
509                                     {
510                                             {
511                                                     lcd_gotoxy(3,1);
512                                                     lcd_printf("Hakkuri virtaraja OK");
513                                                     beeb(1,1000);
514                                                     break;
515                                             }
516                                     }
517                                     if(r_button_rising_edge)                // Jos painetaan punaista nappia...
518                                     {
519                                             stop_test = true;               // keskeytetään testaus, ja mennään
                                                alkuun
520                                             error_code = 17;                // Käyttäjä keskeytti hakkurin
                                                virtarajan säätämisen. Virtarajaa ei ehkä voitu säätää kohdalleen.
521                                             loads_on(false);                // 5v ja 12V kuormat pois päältä
522                                             set_55v_load(0);                // 55v kuorma nollaan
523                                             break;
524                                     }
525                                     delay_ms(200);                          // hidastetaan vähän luupissa
                                        kiertovauhtia
526                             }
527                     }
528                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan
                        eteenpäin...
529                     {
530                             lcd_clear();
531                             lcd_printf("Hakkuri virtaraja OK");
532                             lcd_gotoxy(3,2);
533                             lcd_printf("   ODOTA HETKI...");
534                             loads_on(true);                                 // 5v ja 12V kuormat päälle (4A ja 3A)
535                             set_55v_load(2500);                             // 55v kuorma päälle, virta 2,5A
536
537 // lähtöjännitteiden tarkistus kuorman kanssa
538                             u_5v_mv = read_ad_5v();                         // mitataan 5V
539                             u_5v_mv += 75;                                  // lisätään johtimien resistanssista
                                johtuva jännitehäviö 75mV
540                             u_12v_mv = read_ad_12v();                       // mitataan 12V
541                             u_55v_mv = read_ad_55v();                       // mitataan 55V
542
543                             if( u_5v_mv > 5200 )
544                             {
545                                     stop_test = true;
546                                     error_code =18;                         // 5V yli 5,1V (3A kuorma)
547                             }
548                             if( u_5v_mv < 4800 )
549                             {
550                                     stop_test = true;
551                                     error_code =19;                         // 5V alle 4,9V (3A kuorma)
552                             }
553                             if( u_12v_mv > 15000 )
554                             {
555                                     stop_test = true;
556                                     error_code =20;                         // 12V yli 15V (4A kuorma)
557                             }
558                             if( u_12v_mv < 13000 )
559                             {
560                                     stop_test = true;
561                                     error_code =21;                         // 12V alle 13V (4A kuorma)
562                             }
563                             if( u_55v_mv > 58000 )
564                             {
565                                     stop_test = true;
566                                     error_code =22;                         // 55V yli 58V (2,5A kuorma)
567                             }
568                             if( u_55v_mv < 52000 )
569                             {
570                                     stop_test = true;
571                                     error_code =23;                         // 55V alle 52V (2,5A kuorma)
572                             }
573
574                             loads_on(false);                                // 5v ja 12V kuormat pois päältä
575                     }
576                     if (!stop_test)         //jos ei keskeytetty, jatketaan eteenpäin...
577                     {
578                             loads_on(false);                                // 5v ja 12V kuormat pois päältä
579
580                             main_power_on(false);
581
582                             set_main_voltage(230);
583
584                             main_power_on(true);
585                             delay_s(1);
586                             start_button(true);                             //"painetaan" START kytkin pohjaan
```

```
587                             delay_s(1);
588                             start_button(false);                                    //"vapautetaan" START kytkin
589
590
591
592 // Lampun, erase -signaalin, erase_ON -signaalin ja tuulettimen testaus.
593
594                             lcd_clear();
595                             lcd_printf(" LAMPPU JA TUULETIN");
596                             lcd_gotoxy(3,1);
597                             lcd_printf("Tarkista että lamppu");
598                             lcd_gotoxy(4,1);
599                             lcd_printf("varmasti syttyy!!!!!");
600                             lamp_test_loads_on(true);                               // 5V, 1.3A ja 12V, 0.5A kuormat
                                päälle
601                             set_55v_load(2800);                                     // 55v kuorma 2.8A
602                             delay_s(3);
603                             if (erase_on())                                         // onko Erase ON jännite yli 0.1V?
604                             {
605                                     stop_test = true;
606                                     if(error_code == 0)
607                                     error_code = 24;                                // "erase ON" jännite yli 0.1V, kun
                                        pitäisi olla 0V
608                             }
609
610                             u_12v_mv = read_ad_12v();
611                             u_fan_mv = read_ad_fan();
612                             u_lamp_mv = read_ad_lamp();
613
614                             if(u_12v_mv > u_fan_mv)
615                                     u_fan_mv = u_12v_mv - u_fan_mv;
616                             else
617                                     u_fan_mv = 0;
618                             if(u_12v_mv > u_lamp_mv)
619                                     u_lamp_mv = u_12v_mv - u_lamp_mv;
620                             else
621                                     u_lamp_mv = 0;
622
623                             if (u_fan_mv > 12500)                                   // onko tuulettimen jännite yli 12V?
624                             {
625                                     stop_test = true;
626                                     if(error_code == 0)
627                                     error_code = 25;                                // Tuulettimen jännite yli 12V, kun
                                        pitäisi olla 10-12V
628                             }
629                             if (u_fan_mv < 9000)                                    // onko tuulettimen jännite alle 9V?
630                             {
631                                     stop_test = true;
632                                     if(error_code == 0)
633                                     error_code = 26;                                // Tuulettimen jännite alle 9V, kun
                                        pitäisi olla 10-12V
634                             }
635                             if (u_lamp_mv > 200)                                    // onko lampun jännite yli 0.2V?
636                             {
637                                     stop_test = true;
638                                     if(error_code == 0)
639                                     error_code = 27;                                // lampun jännite yli 0.2V, kun lampun
                                        pitäisi olla sammuksissa. Tarkista onko lamppu palanut tai huono kontakti.
640                             }
641
642                             erase(true);                                            // erase päälle
643                             delay_s(3);                                             // odotellaan lampun syttymistä
644
645                             if(u_12v_mv > u_fan_mv)
646                                     u_fan_mv = u_12v_mv - u_fan_mv;
647                             else
648                                     u_fan_mv = 0;
649                             if(u_12v_mv > u_lamp_mv)
650                                     u_lamp_mv = u_12v_mv - u_lamp_mv;
651                             else
652                                     u_lamp_mv = 0;
653
654
655                             if (!erase_on())                                        // onko Erase ON jännite alle 4.5V?
656                             {
657                                     stop_test = true;
658                                     if(error_code == 0)
659                                     error_code = 28;                                // "erase ON" jännite alle 4.5V, kun
                                        pitäisi olla 4.5V-5V ja lampun pitäisi palaa. (Onko lamppu palanut??)
660                             }
661                             if (u_lamp_mv < 11900)                                  // onko lampun jännite alle 12V?
662                             {
663                                     stop_test = true;
664                                     if(error_code == 0)
665                                     error_code = 29;                                // lampun jännite alle 12V, kun lampun
                                        pitäisi loistaa. Lamppu ei luultavasti syty.
666                             }
667
```

```
668                             lcd_clear();
669                             r_button_rising_edge = false;
670
671 // 5V-jännitteen tarkka säätö lampun loistaessa
672
673                             while(1)
674                             {
675                                     read_buttons();
676                                     lcd_gotoxy(1,1);
677                                     lcd_printf("Säädä trimmeriä R211");
678                                     lcd_gotoxy(2,1);
679                                     u_5v_mv = read_ad_5v();
680                                     u_5v_mv += 33;                          // lisätään johtimien resistanssista
                                        johtuva jännitehäviö 75mV
681                                     lcd_printdec(u_5v_mv);
682                                     lcd_printf("  -> ->  5000");
683                                     if(u_5v_mv < 4999)                      // jos 5V:n lähdön jännite on alle
                                        4,99V:n...
684                                     {
685                                             lcd_gotoxy(3,3);
686                                             lcd_printf("MYÖTÄPÄIVÄÄN");
687                                     }
688                                     else if(u_5v_mv > 5001)                 // jos 5V:n lähdön jännite on yli
                                        5.01V...
689                                     {
690                                             lcd_gotoxy(3,3);
691                                             lcd_printf("VASTAPÄIVÄÄN");
692                                     }
693                                     else                                    // jos 5V:n lähtöjännite on välillä
                                        4.99V ja 5.01V (se olisi tarkoitus)...
694                                     {
695                                             {
696                                                     lcd_gotoxy(3,1);
697                                                     lcd_printf("5V:n lähtöjännite OK");
698                                                     beeb(1,1000);
699                                                     break;
700                                             }
701                                     }
702                                     if(r_button_rising_edge)                // Jos painetaan punaista nappia...
703                                     {
704                                             stop_test = true;               // keskeytetään testaus, ja mennään
                                                alkuun
705                                             if(error_code == 0)
706                                             error_code = 30;                // Käyttäjä keskeytti 5V:n jännitteen
                                                säätämisen trimmerillä 211. Jännitettä ei ehkä voitu säätää kohdalleen.
707                                             break;
708                                     }
709                                     delay_ms(200);                          // hidastetaan vähän luupissa
                                        kiertovauhtia
710                             }
711
712                             erase(false);                                   // erase pois päältä
713                             lamp_test_loads_on(false);                      // kuormat pois päältä
714                             set_55v_load(0);                                // 55v kuorma 0A
715                     }
716                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan
                        eteenpäin...
717                     {
718
719 // HV -jännitteen säätö
720
721                             lcd_clear();
722                             lcd_printf("HV jännitteen säätö");
723                             lcd_gotoxy(3,2);
724                             lcd_printf("   ODOTA HETKI...");
725
726                             hv_enable(true);
727                             set_hvref(5000);                                // asetetaan hvref 5.0V
728                             delay_s(3);
729
730                             lcd_clear();
731                             r_button_rising_edge = false;
732                             while(1)
733                             {
734                                     read_buttons();
735                                     lcd_gotoxy(1,1);
736                                     lcd_printf("Säädä trimmeriä R19");
737                                     lcd_gotoxy(2,1);
738                                     u_hv_voltage_v = read_hv();             // read HV voltage
739                                     lcd_printdec(u_hv_voltage_v);
740                                     lcd_printf("  -> ->  1250");
741                                     if(u_hv_voltage_v < 1245)               // jos HV:n lähdön jännite on alle
                                        1245V:n...
742                                     {
743                                             lcd_gotoxy(3,3);
744                                             lcd_printf("MYÖTÄPÄIVÄÄN");
745                                     }
746                                     else if(u_hv_voltage_v > 1255)          // jos HV:n lähdön jännite on yli
```

```
746                                     1255V...
747                                     {
748                                             lcd_gotoxy(3,3);
749                                             lcd_printf("VASTAPÄIVÄÄN");
750                                     }
751                                     else                                    // jos HV:n lähtöjännite on välillä
                                        1245V ja 1255V (se olisi tarkoitus)...
752                                     {
753                                             {
754                                                     lcd_gotoxy(3,2);
755                                                     lcd_printf("HV:n säätö OK");
756                                                     lcd_gotoxy(4,2);
757                                                     lcd_printf("ODOTA HETKI!");
758                                                     beeb(1,1000);
759                                                     break;
760                                             }
761                                     }
762                                     if(r_button_rising_edge)                // Jos painetaan punaista nappia...
763                                     {
764                                             stop_test = true;               // keskeytetään testaus, ja mennään
                                                alkuun
765                                             error_code = 31;                // Käyttäjä keskeytti HV:n
                                                huippujännitteen säätämisen trimmerillä R19. Jännitettä ei ehkä voitu säätää
                                                kohdalleen.
766                                             break;
767                                     }
768                                     delay_ms(200);                          // hidastetaan vähän luupissa
                                        kiertovauhtia
769                             }
770                     }
771                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan
                        eteenpäin...
772                     {
773
774 // HV rippelin mittaus
775
776                             lcd_clear();
777                             lcd_printf("HV rippelin mittaus");
778                             lcd_gotoxy(3,1);
779                             lcd_printf("Odota hetki...");
780                             lcd_gotoxy(4,1);
781
782                             set_hvref(0);                                   // HV-jännite nollaksi
783                             unsigned char i=0;
784                             unsigned int sweep = 400;                       // aluksi asetetaan 400V
785                             set_hv_voltage(sweep);
786                             while(i < 9)                                    // 9 kierrosta
787                             {
788                                     set_hv_voltage(sweep);                  // käydään läpi 400V - 1200V
789                                     delay_s(5);
790                                     if(read_hv_ripple() > 2000)             // Jos rippeli > 20mV...
791                                     {
792                                             hv_ripple_error = true;
793                                             lcd_gotoxy(1,1);
794                                             lcd_printf("HV:ssa RIPPELIÄ!!!!! Jännitteellä ");
795                                             lcd_printdec(sweep);
796                                             lcd_printf("V");
797                                             beeb(2,300);
798                                             b_button_rising_edge = false;
799                                     }
800                                     lcd_printf("**");
801                                     sweep += 100;
802                                     i++;
803                             }
804                     }
805                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan
                        eteenpäin...
806                     {
807
808 // HV OK -signaalin tarkistus
809
810                             lcd_clear();
811                             lcd_printf("HV OK -signaali...");
812                             lcd_gotoxy(3,1);
813                             lcd_printf("Odota hetki...");
814                             lcd_gotoxy(4,1);
815
816                             if(read_hvok() > 200)
817                             {
818                                     stop_test = true;                       // keskeytetään testaus, ja mennään
                                        alkuun
819                                     if(error_code == 0)
820                                     error_code = 33;                        // HV OK on yli 0.2V, kun pitäisi olla
                                        0V!
821                             }
822                             hv_ok_test_a(true);
823                             delay_s(2);
824                             if(read_hvok() > 6600)
```

```
825                             {
826                                     stop_test = true;                       // keskeytetään testaus, ja mennään
                                        alkuun
827                                     if(error_code == 0)
828                                     error_code = 34;                        // HV OK on yli 6.6V, kun pitäisi olla
                                        6.2V (J18.1 ja J18.2 yhdistetty)
829                             }
830                             if(read_hvok() < 5800)
831                             {
832                                     stop_test = true;                       // keskeytetään testaus, ja mennään
                                        alkuun
833                                     if(error_code == 0)
834                                     error_code = 35;                        // HV OK on alle 5.8V, kun pitäisi
                                        olla 6.2V (J18.1 ja J18.2 yhdistetty)
835                             }
836                             hv_ok_test_a(false);
837                             hv_ok_test_b(true);
838                             delay_s(2);
839                             if(read_hvok() > 6600)
840                             {
841                                     stop_test = true;                       // keskeytetään testaus, ja mennään
                                        alkuun
842                                     if(error_code == 0)
843                                     error_code = 36;                        // HV OK on yli 6.6V, kun pitäisi olla
                                        6.2V (J18.2 ja J18.3 yhdistetty)
844                             }
845                             if(read_hvok() < 5800)
846                             {
847                                     stop_test = true;                       // keskeytetään testaus, ja mennään
                                        alkuun
848                                     if(error_code == 0)
849                                     error_code = 37;                        // HV OK on alle 5.8V, kun pitäisi
                                        olla 6.2V (J18.2 ja J18.3 yhdistetty)
850                             }
851                             hv_ok_test_b(false);
852
853                             set_hvref(0);                                   // HV-jännite nollaksi
854                             hv_enable(false);
855
856                     }
857                     if (!stop_test)                                         //jos ei keskeytetty, jatketaan
                        eteenpäin...
858                     {
859
860 // Testataan notkahtaako 5V:n jännite, kun lamppu sytytetään.
861 // 5V:n jännite ei saa mennä alle 4,900V!!
862
863                             lcd_clear();
864                             lcd_printf(" Odotellaan lampun");
865                             lcd_gotoxy(2,2);
866                             lcd_printf("jäähtymistä...");
867                             char secs = 5;
868                             while(secs > 0)
869                             {
870                                     lcd_gotoxy(4,2);
871                                     lcd_printdec(secs);
872                                     delay_s(1);
873                                     secs--;
874                             }
875
876                             loads_on(false);                                // 5v ja 12V kuormat pois päältä
877                             lamp_test_loads_on(true);                       // 5V, 1.3A ja 12V, 0.5A kuormat
                                päälle
878                             set_55v_load(2800);                             // 55v kuorma 2.8A
879                             delay_s(2);
880                             EXT_INT = 0;                                    // Nollataan external interrupt -
                                lippu, joka menee ykköseksi aina,
881                                                                             // kun 5V laskee alle 4,90V
                                                                                (kalibroidaan trimmerillä R76)
882                             erase(true);                                    // sytytetään lamppu.
883                             delay_s(1);
884
885                             if(EXT_INT)                                     // jos tuli keskeytys niin 5V notkahti
                                liikaa.
886                             {
887                                     stop_test = true;                       // keskeytetään testaus, ja mennään
                                        alkuun
888                                     error_code = 38;                        // 5V:n jännite notkahtaa liikaa, kun
                                        lamppu sytytetään. (5V alle 4,90V)
889                             }
890
891                     }
892
893 // TESTAUS LOPETETAAN JA KERROTAAN KÄYTTÄJÄLLELLE TESTIN TULOKSET...
894                     main_power_on(false);                                   //sähköt pois!!
895
896                     if (stop_test && !error_code)                           //jos testi keskeytettiin...
897                     {
```

```
898                             lcd_clear();
899                             lcd_gotoxy(2,1);
900                             lcd_printf("TESTI KESKEYTETTIIN!");
901                             beeb(3,200);
902                             delay_s(3);
903                     }
904                     else                                                    //jos ei keskeytetty...
905                     {
906                             if (error_code)                                 //jos tuli virheitä...
907                             {
908                                     lcd_clear();
909                                     lcd_gotoxy(1,5);
910                                     lcd_printf("Vikoja löytyi!!!");
911                                     lcd_gotoxy(3,2);
912                                     lcd_printf("VIKAKOODI = ");
913                                     lcd_printdec(error_code);
914                                     beeb(6,100);
915                             }
916                             if (hv_ripple_error)                            //jos HV:ssa oli rippeliä...
917                             {
918                                     b_button_rising_edge = 0;
919                                     if(error_code)
920                                     {
921                                             while(!b_button_rising_edge)    //odotetaan napin painallusta
922                                             {
923                                                     read_buttons();
924                                             }
925                                     }
926                                     lcd_clear();
927                                     lcd_gotoxy(1,5);
928                                     lcd_printf("Vikoja löytyi!!!");
929                                     lcd_gotoxy(3,2);
930                                     lcd_printf("HV:ssa oli rippeliä!");
931                                     b_button_rising_edge = 0;
932                             }
933                             if (!error_code && !hv_ripple_error)            //jos ei tullut virheitä
934                             {
935                                     lcd_clear();
936                                     lcd_gotoxy(1,5);
937                                     lcd_printf("TESTI LÄPI!");
938                                     lcd_gotoxy(3,5);
939                                     lcd_printf("KAIKKI OK!");
940                             }
941                             b_button_rising_edge = 0;
942                             cover_lock_disable(true);
943                             while(!b_button_rising_edge)                    //odotetaan napin painallusta
944                             {
945                                     read_buttons();
946                             }
947                     }
948                     r_button_rising_edge = 0;
949                     g_button_rising_edge = 0;
950             }
951
952
953             ////////////////////////////////////////////////////////////////////////////////////////////////////////////
954             ////////////////////////////////////////////////////////////////////////////////////////////////////////////
955             ////////////////////////////////////////////////////////////////////////////////////////////////////////////
956
957
958
959             // HUOLTOTILA!!!!!!!!!!
960
961             if(g_button_rising_edge)                        //alkuvalikossa vihreällä pääsee huoltotilaan
962             {
963                     lcd_clear();
964                     lcd_gotoxy(1,1);
965                     lcd_printf("*****HUOLTOTILA*****");
966                     lcd_gotoxy(2,1);
967                     lcd_printf("Lue kalibrointiohje!");
968                     lcd_gotoxy(3,1);
969                     lcd_printf("(laitteen sisällä)");
970                     lcd_gotoxy(4,1);
971                     lcd_printf("A = JATKA C = POISTU");
972                     r_button_rising_edge = 0;
973                     g_button_rising_edge = 0;
974                     b_button_rising_edge = 0;
975                     while(!b_button_rising_edge && !r_button_rising_edge && !g_button_rising_edge)  //odotetaan napin
                        painallusta
976                     {
977                             read_buttons();
978                     }
979                     if (b_button_rising_edge || r_button_rising_edge)       //jos painettiin punaista tai sinistä...
980                     {
981                             lcd_clear();
```

```
982                             lcd_gotoxy(1,5);
983                             lcd_printf("Poistutaan");
984                             lcd_gotoxy(2,5);
985                             lcd_printf("huoltotilasta");
986                             delay_s(3);
987                     }
988                     if (g_button_rising_edge)               // vihreällä painikkeella kalibrointiin
989                     {
990                             lamp_test_loads_on(true);
991                             lcd_clear();
992                             lcd_gotoxy(1,3);
993                             lcd_printf("Kalibrointi!!!");
994                             delay_s(3);
995                             lcd_clear();
996                             lcd_printf("5V jännite:");
997                             b_button_rising_edge = false;
998                             while(!b_button_rising_edge)
999                             {
1000                                    lcd_gotoxy(3,4);
1001                                    lcd_printdec(read_ad_5v() + 33);
1002                                    lcd_printf("mV");
1003                                    b_button_rising_edge = false;
1004                                    read_buttons();
1005                                    delay_ms(100);
1006                            }
1007                            lamp_test_loads_on(false);
1008                            lcd_clear();
1009                            lcd_printf("12V jännite:");
1010                            b_button_rising_edge = false;
1011                            while(!b_button_rising_edge)
1012                            {
1013                                    lcd_gotoxy(3,4);
1014                                    lcd_printdec(read_ad_12v());
1015                                    lcd_printf("mV");
1016                                    b_button_rising_edge = false;
1017                                    read_buttons();
1018                                    delay_ms(100);
1019                            }
1020                            lcd_clear();
1021                            lcd_printf("55V jännite:");
1022                            b_button_rising_edge = false;
1023                            while(!b_button_rising_edge)
1024                            {
1025                                    lcd_gotoxy(3,4);
1026                                    lcd_printdec(read_ad_55v());
1027                                    lcd_printf("mV");
1028                                    b_button_rising_edge = false;
1029                                    read_buttons();
1030                                    delay_ms(100);
1031                            }
1032                            lcd_clear();
1033                            lcd_printf("+12V jännite:");
1034                            b_button_rising_edge = false;
1035                            while(!b_button_rising_edge)
1036                            {
1037                                    lcd_gotoxy(3,4);
1038                                    lcd_printdec(read_ad_plus12v());
1039                                    lcd_printf("mV");
1040                                    b_button_rising_edge = false;
1041                                    read_buttons();
1042                                    delay_ms(100);
1043                            }
1044                            lcd_clear();
1045                            lcd_printf("-12V jännite:");
1046                            b_button_rising_edge = false;
1047                            while(!b_button_rising_edge)
1048                            {
1049                                    lcd_gotoxy(3,4);
1050                                    lcd_printdec(read_ad_miinus12v());
1051                                    lcd_printf("mV");
1052                                    b_button_rising_edge = false;
1053                                    read_buttons();
1054                                    delay_ms(100);
1055                            }
1056                            lcd_clear();
1057                            lcd_printf("Variakin jännite");
1058                            b_button_rising_edge = false;
1059                            while(!b_button_rising_edge)
1060                            {
1061                                    lcd_gotoxy(3,4);
1062                                    lcd_printdec(read_main_voltage());
1063                                    lcd_printf("V");
1064                                    b_button_rising_edge = false;
1065                                    read_buttons();
1066                                    delay_ms(100);
1067                            }
1068                            lcd_clear();
1069                            lcd_printf("VBUS alaraja:");
```

```
1070                            b_button_rising_edge = false;
1071                            while(!b_button_rising_edge)
1072                            {
1073                                    lcd_gotoxy(2,1);
1074                                    lcd_printf("VBUS-jännite on ");
1075                                    if(vbus_under())
1076                                    lcd_printf("alle");
1077                                    else
1078                                    lcd_printf("yli ");
1079                                    lcd_gotoxy(3,1);
1080                                    lcd_printf(" alarajan");
1081                                    read_buttons();
1082                                    delay_ms(200);
1083                            }
1084                            lcd_clear();
1085                            lcd_printf("VBUS yläraja:");
1086                            b_button_rising_edge = false;
1087                            while(!b_button_rising_edge)
1088                            {
1089                                    lcd_gotoxy(2,1);
1090                                    lcd_printf("VBUS-jännite on ");
1091                                    if(vbus_over())
1092                                    lcd_printf("yli ");
1093                                    else
1094                                    lcd_printf("alle");
1095                                    lcd_gotoxy(3,1);
1096                                    lcd_printf(" ylärajan");
1097                                    read_buttons();
1098                                    delay_ms(200);
1099                            }
1100                            lcd_clear();
1101                            lcd_printf("Tuuletin:");
1102                            b_button_rising_edge = false;
1103                            while(!b_button_rising_edge)
1104                            {
1105                                    lcd_gotoxy(3,4);
1106                                    lcd_printdec(read_ad_fan());
1107                                    lcd_printf("mV");
1108                                    b_button_rising_edge = false;
1109                                    read_buttons();
1110                                    delay_ms(100);
1111                            }
1112                            lcd_clear();
1113                            lcd_printf("Lamppu:");
1114                            b_button_rising_edge = false;
1115                            while(!b_button_rising_edge)
1116                            {
1117                                    lcd_gotoxy(3,4);
1118                                    lcd_printdec(read_ad_lamp());
1119                                    lcd_printf("mV");
1120                                    b_button_rising_edge = false;
1121                                    read_buttons();
1122                                    delay_ms(100);
1123                            }
1124                            lcd_clear();
1125                            lcd_printf("5V:n notkahdus:");
1126                            b_button_rising_edge = false;
1127                            EXT_INT = 0;                            // Nollataan external interrupt -lippu, joka
                                menee ykköseksi aina,
1128                                                                    // kun 5V laskee alle 4,90V (kalibroidaan
                                                                        trimmerillä R76)
1129                            while(!b_button_rising_edge)
1130                            {
1131                                    if(EXT_INT)                     // jos tuli keskeytys niin 5V notkahti liikaa.
1132                                    {
1133                                            EXT_INT = 0;            // Nollataan external interrupt -lippu, joka
                                                menee ykköseksi aina,
1134                                                                    // kun 5V laskee alle 4,90V (kalibroidaan
                                                                        trimmerillä R76)
1135                                            beeb(10,5);
1136                                    }
1137                                    read_buttons();
1138                            }
1139                            lcd_clear();
1140                            lcd_printf("HV jännite:");
1141                            b_button_rising_edge = false;
1142                            while(!b_button_rising_edge)
1143                            {
1144                                    lcd_gotoxy(3,4);
1145                                    lcd_printdec(read_hv());
1146                                    lcd_printf("V");
1147                                    b_button_rising_edge = false;
1148                                    read_buttons();
1149                                    delay_ms(100);
1150                            }
1151                            lcd_clear();
1152                            lcd_printf("HV RIPPELI:");
1153                            b_button_rising_edge = false;
```

```
1154                                while(!b_button_rising_edge)
1155                                {
1156                                        lcd_gotoxy(3,4);
1157                                        lcd_printdec(read_hv_ripple());
1158                                        lcd_printf("mV");
1159                                        b_button_rising_edge = false;
1160                                        read_buttons();
1161                                        delay_ms(100);
1162                                }
1163                                lcd_clear();
1164                                lcd_printf("HVREF:");
1165                                lcd_gotoxy(2,2);
1166                                lcd_printf("Trimmeri R77");
1167                                lcd_gotoxy(3,2);
1168                                lcd_printf("J105.7 <-> J105.20");
1169                                lcd_gotoxy(4,4);
1170                                lcd_printf("--> 5.00V");
1171                                set_hvref(5000);                                // asetetaan hvref 5.0V
1172                                b_button_rising_edge = false;
1173                                while(!b_button_rising_edge)
1174                                {
1175                                        read_buttons();
1176                                }
1177
1178
1179
1180
1181
1182                                lcd_clear();
1183                                lcd_printf("KALIBROINTI VALMIS!!");
1184
1185                        }
1186                        r_button_rising_edge = 0;
1187                        g_button_rising_edge = 0;
1188
1189                }
1190
1191                ////////////////////////////////////////////////////////////////////////////////
1192
1193                if(r_button_rising_edge)        // alkuvalikossa punainen nappi heittää ohjeeseen
1194                {
1195
1196 ////////////////////////////// DEBUGGAUSTA!! KÄYTETÄÄN VAIN TESTAAMISEEN!!!!!!!!!!!!!!
1197 /*
1198                    erase(false);               // erase pois päältä
1199                    loads_on(false);            // 5v ja 12V kuormat pois päältä
1200                    lamp_test_loads_on(true);   // 5V, 1.3A ja 12V, 0.5A kuormat päälle
1201                    main_power_on(true);
1202                    delay_s(2);
1203                    lcd_clear();
1204                    start_button(true);         //"painetaan" START kytkin pohjaan
1205                    delay_s(2);
1206                    start_button(false);        //"vapautetaan" START kytkin
1207                    delay_s(1);
1208
1209                    set_55v_load(2800);         // 55v kuorma 2.7A
1210                    delay_s(1);
1211
1212
1213                    while(1)
1214                    {
1215                            EXT_INT = 0;        // Nollataan external interrupt -lippu, joka menee ykköseksi aina,
1216                                                // kun 5V laskee alle 4,90V (kalibroidaan trimmerillä R76)
1217                            erase(true);        // sytytettään lamppu.
1218                            delay_ms(500);
1219
1220                            if(EXT_INT)         // jos tuli keskeytys niin 5V notkahti liikaa.
1221                            {
1222                                    beeb(2,100);
1223                                    lcd_clear();
1224                                    lcd_printf("Notkahti!!!");
1225                            }
1226
1227                            erase(false);       // erase pois päältä
1228                            delay_s(1);
1229                            lcd_clear();
1230                            delay_s(10);
1231                    }
1232 */
1233 ////////////////////////////////////////////////////////////////////////////////
1234
1235 // OHJE!!!!!!!!!!!!!!!!!!!
1236
1237 // Ohje poistettiin käytöstä ohjelmakoodin loputtua kesken.
1238
1239 /*                      r_button_rising_edge = 0;
1240                         lcd_clear();
1241                         lcd_gotoxy(1,1);
```

```
1242                        lcd_printf("*******OHJEET*******");
1243                        lcd_gotoxy(2,1);
1244                        lcd_printf("Nappi A jatkaa tes-");
1245                        lcd_gotoxy(3,1);
1246                        lcd_printf("tiohjelman suoritus-");
1247                        lcd_gotoxy(4,1);
1248                        lcd_printf("ta.        A = JATKA");
1249                        b_button_rising_edge = 0;
1250                        while(!b_button_rising_edge)
1251                        {
1252                                read_buttons();
1253                        }
1254                        lcd_clear();
1255                        lcd_gotoxy(1,1);
1256                        lcd_printf("Nappi B keskeyttää");
1257                        lcd_gotoxy(2,1);
1258                        lcd_printf("testaamisen");
1259                        lcd_gotoxy(3,1);
1260                        lcd_printf(" ");
1261                        lcd_gotoxy(4,1);
1262                        lcd_printf("A = JATKA");
1263                        b_button_rising_edge = 0;
1264                        while(!b_button_rising_edge)
1265                        {
1266                                read_buttons();
1267                        }
1268                        lcd_clear();
1269                        lcd_gotoxy(1,1);
1270                        lcd_printf("Huoltotilaan pääsee");
1271                        lcd_gotoxy(2,1);
1272                        lcd_printf("painamalla alussa");
1273                        lcd_gotoxy(3,1);
1274                        lcd_printf("vihreää nappia");
1275                        lcd_gotoxy(4,1);
1276                        lcd_printf("A = JATKA");
1277                        b_button_rising_edge = 0;
1278                        while(!b_button_rising_edge)
1279                        {
1280                                read_buttons();
1281                        }
1282                        r_button_rising_edge = 0;
1283                        g_button_rising_edge = 0;
1284 */                 }
1285         }
1286
1287 }
1288
1289
1290
1291
1292
1293 void read_buttons(void) //nappien luku funktio
1294 {
1295         char buttons;
1296         buttons = read_io_exp();
1297
1298 //***RED BUTTON****************
1299         if (buttons & 0b00100000)
1300                 r_button = 1;
1301         else
1302                 r_button = 0;
1303         if (r_button && !r_button_old)
1304         {
1305                 r_button_rising_edge = 1;
1306         }
1307         r_button_old = r_button;
1308
1309 //***GREEN BUTTON**************
1310         if (buttons & 0b01000000)
1311                 g_button = 1;
1312         else
1313                 g_button = 0;
1314         if (g_button && !g_button_old)
1315         {
1316                 g_button_rising_edge = 1;
1317         }
1318         g_button_old = g_button;
1319
1320 //***BLUE BUTTON***************
1321         if (buttons & 0b10000000)
1322                 b_button = 1;
1323         else
1324                 b_button = 0;
1325         if (b_button && !b_button_old)
1326         {
1327                 b_button_rising_edge = 1;
1328         }
1329         b_button_old = b_button;
```

```
1330 }
1331
1332
1333 void leds(bit red, bit green, bit blue) // ledien asetusfunktio
1334         {
1335                 if(red)
1336                         clear_bit(portd,5);
1337                 else
1338                         set_bit(portd,5);
1339                 if(green)
1340                         clear_bit(portd,6);
1341                 else
1342                         set_bit(portd,6);
1343                 if(blue)
1344                         clear_bit(portd,7);
1345                 else
1346                         set_bit(portd,7);
1347         }
1348
1349 ////////////////////////////////////////////////////////////////////////////////////////////////
1350
1351 //      INPUTS!!!!!!!!!!!!!!!!!!!
1352
1353 unsigned char stb_led(void)
1354 {
1355         unsigned char io_exp_value;
1356         io_exp_value = read_io_exp();
1357         return !io_exp_value.0;
1358 }
1359 unsigned char vbus_over(void)
1360 {
1361         unsigned char io_exp_value;
1362         io_exp_value = read_io_exp();
1363         return !io_exp_value.1;
1364 }
1365 unsigned char vbus_under(void)
1366 {
1367         unsigned char io_exp_value;
1368         io_exp_value = read_io_exp();
1369         return !io_exp_value.2;
1370 }
1371 unsigned char cover_open(void)
1372 {
1373         unsigned char io_exp_value;
1374         io_exp_value = read_io_exp();
1375         return io_exp_value.3;
1376 }
1377 unsigned char erase_on(void)
1378 {
1379         unsigned char io_exp_value;
1380         io_exp_value = read_io_exp();
1381         return io_exp_value.4;
1382 }
1383
1384 ////////////////////////////////////////////////////////////////////////////////////////////////
1385
1386 //      OUTPUTS!!!!!!!!!!!!!!!!!!!
1387
1388 void loads_on(bool state)
1389 {
1390         if (state)
1391                 set_bit(io_exp_stat,0);
1392         else
1393                 clear_bit(io_exp_stat,0);
1394
1395         write_io_exp(io_exp_stat);
1396 }
1397 void start_button(bool state)
1398 {
1399         if (state)
1400                 set_bit(io_exp_stat,1);
1401         else
1402                 clear_bit(io_exp_stat,1);
1403
1404         write_io_exp(io_exp_stat);
1405 }
1406 void main_power_on(bool state)
1407 {
1408         if (state)
1409                 set_bit(io_exp_stat,2);
1410         else
1411                 clear_bit(io_exp_stat,2);
1412
1413         write_io_exp(io_exp_stat);
1414 }
1415 void cover_lock_disable(bool state)
1416 {
1417         if (state)
```

```
1418                    set_bit(io_exp_stat,3);
1419            else
1420                    clear_bit(io_exp_stat,3);
1421
1422            write_io_exp(io_exp_stat);
1423    }
1424    void hv_ok_test_a(bool state)
1425    {
1426            if (state)
1427                    set_bit(io_exp_stat,4);
1428            else
1429                    clear_bit(io_exp_stat,4);
1430
1431            write_io_exp(io_exp_stat);
1432    }
1433    void hv_ok_test_b(bool state)
1434    {
1435            if (state)
1436                    set_bit(io_exp_stat,5);
1437            else
1438                    clear_bit(io_exp_stat,5);
1439
1440            write_io_exp(io_exp_stat);
1441    }
1442    void overtemp_test(bool state)
1443    {
1444            if (state)
1445                    set_bit(io_exp_stat,6);
1446            else
1447                    clear_bit(io_exp_stat,6);
1448
1449            write_io_exp(io_exp_stat);
1450    }
1451    void erase(bool state)
1452    {
1453            if (!state)
1454                    set_bit(io_exp_stat,7);
1455            else
1456                    clear_bit(io_exp_stat,7);
1457
1458            write_io_exp(io_exp_stat);
1459    }
1460
1461    ////// ADC-channels ////////////////////////////////////
1462
1463    unsigned int read_main_voltage(void)
1464    {
1465            int ad_result;
1466            ad_result = Read_AD(2);
1467            unsigned char i;
1468            for(i=0; i < 3; i++)
1469            {
1470                    ad_result += Read_AD(2);
1471            }
1472            ad_result /= 4;
1473            ad_result += 18;
1474            return ad_result;
1475    }
1476
1477    unsigned int read_ad_5v(void)
1478    {
1479            int ad_result;
1480            ad_result =  Read_AD(6);
1481            unsigned char i;
1482            for(i=0; i < 7; i++)
1483            {
1484                    ad_result += Read_AD(6);
1485            }
1486            return ad_result / 4;
1487    }
1488
1489    unsigned int read_ad_12v(void)
1490    {
1491            int ad_result;
1492            ad_result = Read_AD(4);
1493            unsigned char i;
1494            for(i=0; i < 3; i++)
1495            {
1496                    ad_result += Read_AD(4);
1497            }
1498            return ad_result;
1499    }
1500
1501    unsigned int read_ad_55v(void)
1502    {
1503            int ad_result;
1504            ad_result = Read_AD(10);
1505            ad_result *= 16;
```

```
1506         return ad_result;
1507 }
1508
1509 unsigned int read_ad_plus12v(void)
1510 {
1511         int ad_result;
1512         ad_result = Read_AD(9);
1513         ad_result *= 4;
1514         return ad_result;
1515 }
1516
1517 unsigned int read_ad_miinus12v(void)
1518 {
1519         int ad_result;
1520         ad_result = Read_AD(8);
1521         ad_result *= 4;
1522         return ad_result;
1523 }
1524
1525 unsigned int read_ad_fan(void)
1526 {
1527         int ad_result;
1528         ad_result = Read_AD(11);
1529         unsigned char i;
1530         for(i=0; i < 3; i++)
1531         {
1532                 ad_result += Read_AD(11);
1533         }
1534         return ad_result;
1535 }
1536 unsigned int read_ad_lamp(void)
1537 {
1538         int ad_result;
1539         ad_result = Read_AD(13);
1540         ad_result *= 4;
1541         return ad_result;
1542 }
1543
1544 unsigned int read_hv(void)
1545 {
1546         int ad_result;
1547         ad_result = Read_AD(1);
1548         unsigned char i;
1549         for(i=0; i < 3; i++)
1550         {
1551                 ad_result += Read_AD(1);
1552         }
1553         return ad_result / 8;                           // returns volts
1554 }
1555
1556 unsigned int read_hv_ripple(void)
1557 {
1558         int ad_result;
1559         ad_result = Read_AD(0);
1560         unsigned char i;
1561         for(i=0; i < 7; i++)
1562         {
1563                 ad_result += Read_AD(0);
1564         }       return ad_result / 8;
1565 }
1566 unsigned int read_hvok(void)
1567 {
1568         int ad_result;
1569         ad_result = Read_AD(7);
1570         return ad_result * 2;                           // returns millivolts
1571 }
1572
1573 void set_55v_load(unsigned int new_current)             //current = virta (mA)
1574 {
1575         int kierros = 0;
1576         long current_now;                               // tämän hetkinen virta-arvo
1577         if (new_current == 0)
1578                 ccpr1l = 0;
1579         else
1580         {
1581                 while(1)
1582                 {
1583                         current_now = Read_AD(5);       // luetaan tämän hetkinen virta-arvo (sunttivastuksen jännite)
1584
1585                         // I = U / R
1586                         // U = Read_ad(5)
1587                         // R = 0.68R
1588                         // => I = U*1.47 => I = (U*(1.47*128))/128
1589
1590                         current_now *= 188;     // =1.47*128
1591                         current_now /= 128 ;    // /128
1592
1593                         delay_ms(100);
```

```
1594                        if (new_current > (current_now + 1))
1595                                {
1596                                        if(ccpr1l < 255)
1597                                                ++ccpr1l;       // Increment duty cycle.
1598                                }
1599                        else if (new_current < (current_now - 1))
1600                                {
1601                                        if(ccpr1l > 0)
1602                                                --ccpr1l;       // Decrement duty cycle.
1603                                }
1604                        else
1605                                {
1606                                                break;
1607                                }
1608
1609                        ++kierros;
1610                        if(kierros == 400)              // jos virtaa ei saatu kohdalleen...
1611                        {
1612                                stop_test = true;       // ERROR!
1613                                error_code = 100;       // Virtaa ei saatu kohdalleen tai
1614                                break;                  // poweri ei jaksanut syöttää tarpeeksi virtaa
1615                        }
1616                }
1617        }
1618 }
1619
1620 void lamp_test_loads_on(bool state)
1621 {
1622         if (state)
1623                 set_bit(porta,4);                       //loads on
1624         else
1625                 clear_bit(porta,4);                     //Loads off
1626 }
1627
1628 void hv_enable(bool state)
1629 {
1630         if (state)
1631                 clear_bit(porta,6);                     //HV enable
1632         else
1633                 set_bit(porta,6);                       //HV disable
1634 }
1635
1636 void set_main_voltage(unsigned char main_voltage)
1637 {
1638         unsigned int u_variakki_v;                      // AD2
1639         while(1)
1640         {
1641                 b_button_rising_edge = false;
1642                 if(cover_open())
1643                 {
1644                         lcd_clear();
1645                         lcd_gotoxy(2,3);
1646                         lcd_printf("SULJE KANSI!!!");
1647                         while(cover_open());
1648                         lcd_clear();
1649                 }
1650                 lcd_gotoxy(2,1);
1651                 lcd_printf("Säädä variakki ");
1652                 lcd_printdec(main_voltage);
1653                 lcd_putc('V');
1654                 lcd_gotoxy(3,1);
1655                 lcd_printf("Jännite on nyt ");
1656                 u_variakki_v = read_main_voltage();     // read main voltage
1657                 lcd_printdec(u_variakki_v);
1658                 lcd_putc('V');
1659                 lcd_gotoxy(4,1);
1660                 read_buttons();
1661                 if ((u_variakki_v > (main_voltage - 3)) && (u_variakki_v < main_voltage + 3))
1662                 {
1663                         lcd_printf("Jännite OK!! PAINA A");     //testausta voi jatkaa, jos jännite on ok
1664                         if(b_button_rising_edge)
1665                                 {
1666                                         beeb(1,50);
1667                                         break;
1668                                 }
1669                 }
1670                 else if (u_variakki_v > (main_voltage + 2))
1671                         lcd_printf("Jännite liian suuri!");
1672                 else
1673                         lcd_printf("Jännite liian pieni!");
1674
1675                 if(r_button_rising_edge)                //jos painettiin punaista...
1676                 {
1677                         beeb(2,50);
1678                         stop_test = true;               //keskeytetään testaus, ja mennään alkuun
1679                         break;
1680                 }
1681                 delay_ms(100);
```

```
1682         }
1683 }
1684
1685 void set_hv_voltage(unsigned int new_hv_voltage)
1686 {
1687         if(new_hv_voltage == 0)
1688                 ccpr2l = 0;
1689         int kierros = 0;
1690         unsigned int hv_voltage_now;                    // tämän hetkinen jännite
1691         while(1)
1692         {
1693                 hv_voltage_now = read_hv();             // luetaan tämän hetkinen HV-jännite
1694                 delay_ms(120);
1695
1696                 if (new_hv_voltage > (hv_voltage_now + 10))
1697                         {
1698                                 if(ccpr2l < 127)
1699                                         ++ccpr2l;       // Increment duty cycle.
1700                         }
1701                 else if (new_hv_voltage < (hv_voltage_now - 10))
1702                         {
1703                                 if(ccpr2l > 0)
1704                                         --ccpr2l;       // Decrement duty cycle.
1705                         }
1706                 else
1707                         {
1708                                         break;
1709                         }
1710
1711                 ++kierros;
1712                 if(kierros == 300)                      // jos virtaa ei saatu kohdalleen...
1713                 {
1714                         stop_test = true;               // ERROR!
1715                         error_code = 200;               // HV jännitettä ei saatu kohdalleen
1716                         break;
1717                 }
1718         }
1719
1720 }
1721
1722 // HVREF-jännitteen asetus ////////////////////
1723
1724 void set_hvref(int hvref)                               // haluttu hvref-jännite annetaan millivoltteina
1725 {
1726         hvref /= 40;
1727         ccpr2l  = hvref;                                // Set duty cycle.
1728 }
1729
1730 ////////////////////////////////////////////////////////////////////////////////////////////////////////
1731 // END OF FILE
```