



**LAHDEN AMMATTIKORKEAKOULU**  
*Lahti University of Applied Sciences*

# KÄYTTÖLIITTYMÄTESTERI

LAHDEN  
AMMATTIKORKEAKOULU  
Tekniikan ala  
Tietotekniikka  
Tietokone-elektronikka  
Opinnäytetyö  
Kevät 2014  
Ville Viitala

Lahden ammattikorkeakoulu  
Tietotekniikka

VIITALA, VILLE:

Käyttöliittymätesteri

Tietokone-elektroniikan opinnäytetyö, 60 sivua, 14 liitesivua

Kevät 2014

TIIVISTELMÄ

---

Tämä opinnäytetyö käsittelee hitsauskoneiden käyttöliittymätesterin suunnittelua ja toteutusta. Työ toteutettiin Kemppi Oy:n toimeksiannosta. Tarkoituksena oli suunnitella ja toteuttaa testerilaitte, joka kykenee sähköisesti lukemaan hitsauskoneiden käyttöliittymässä näkyvät tiedot ja välittämään ne tietokoneelle. Laitteen tulee myös kyetä simuloimaan käyttäjän vuorovaikutusta käyttöliittymään manipuloimalla painonappien ja muiden käyttöliittymäkomponenttien sähköisiä signaaleja. Laitetta tullaan hyödyntämään testausautomaatioissa.

Työ aloitettiin tutkimalla hitsauslaitteistojen käyttöliittymäpaneeleja ja niiden sisältämien komponenttien toimintaa. Näistä saatujen tietojen perusteella pystyttiin suunnittelemaan testerissä käytettävät testausliittymät sekä toteuttamaan testerin toiminnalliset ominaisuudet.

Testattavat paneelit tullaan säilyttämään mahdollisimman alkuperäisinä, jotta ne vastaavat loppukäyttäjälle meneviä tuotteita. Tämän takia paneelien liittäminen testeriin tullaan hoitamaan itse tehdyllä neulapedillä.

Työn lopputuloksena saatiin toteutettua alkuperäiset vaatimukset täyttävä testerilaitte. Laitteeseen jätettiin laajennusmahdollisuuksia, jotka mahdollistavat erilaisten lisätoimintojen lisäämisen tulevaisuudessa.

Asiasanat: testeri, käyttöliittymä, testiautomaatio, elektroniikka, hitsauskone

Lahti University of Applied Sciences  
Degree Programme in Information Technology

VIITALA, VILLE:

User interface panel tester

Bachelor's Thesis in computer electronics, 60 pages, 14 pages of appendices

Spring 2014

ABSTRACT

---

The purpose of this Bachelor's thesis was to design and implement a user interface tester for welding machine systems. The device was created for Kemppi Oy. The idea was to design and create a tester device that can electrically read the information displayed in the user interface and send the data to a computer. The device must also be capable of simulating human interaction with the user interface panel by manipulating electrical signals of the user interface components. The tester device is going to be a part of a test automation system.

The work with the thesis started by examining user interface panels of welding machine systems and the operation of the electronic components involved with them. Based on the information gathered from the user interfaces, the test interface connections and the functions of the tester's microcontroller were implemented.

The user interface panels to be tested are going to be kept as standard as possible to ensure that they match the customers' end products. This is why the panels are going to be interconnected to the tester by a self-made fixture.

As a result of this thesis, a tester device fulfilling the requirements was successfully created. The device was implemented so that it can be expanded in the future and by doing so add new features to the device.

Key words: tester, user interface, test automation, electronics, welding machine

## SANASTO

AD	Analogia- digitaalimuunnos
CA	(Common Anode) Yhteisanodi
CC	(Common Cathode) Yhteiskatodi
DA	Digitaali- analogiamuunnos
ESD	(Electrical Static Discharge) Kipinäpurkaus
GPIO	(General Purpose Input Output) Yleiskäyttöinen IO-liityntä
IO	(Input Output) Sisääntulo- ja ulostuloliityntä
MAG	(Metal Active Gas) Kaarihitsausmenetelmä, jossa suojakaasu reagoi aktiivisesti hitsisulan kanssa
Mb/s	Megabittiä per sekunti
MIG	(Metal Inertia Gas) Kaarihitsausmenetelmä, jossa suojakaasu ei reagoi hitsisulan kanssa
MMA	(Manual Metal Arc) Kaarihitsausmenetelmä, jossa valokaari muodostuu hitsauspuikon ja hitsattavan kappaleen väliin, ns. ”puikkohitsausta”
PWM	Pulssinleveysmodulaatio
SPI	(Serial Peripheral Interface) Sarjamuotoinen tiedonsiirtoväylä
TIG	(Tungsten Inertia Gas) Kaarihitsausmenetelmä, jossa valokaari ei kuluta hitsauselektrodiä

## SISÄLLYS

1	JOHDANTO	1
2	OHJELMISTOTESTAUS	3
3	JÄRJESTELMÄHITSAUSLAITTEISTOT	6
4	VAATIMUKSIEN MÄÄRITTELY	10
4.1	Testausympäristö	10
4.2	Testerin ja käyttöliittymien vuorovaikutustoiminnot	10
4.3	Testausliittynät ja niiden kytkeminen	11
4.4	Testerin liittäminen PC-tietokoneeseen	12
4.5	Muut työlle asetettavat vaatimukset	13
5	TESTAUKSEEN LIITTYVÄT KOMPONENTIT	14
5.1	Rotaatioenkooderi	14
5.2	Alfanumeerinen LCD-näyttö	16
5.3	Näppäinmatriisi	20
5.4	Seitsensegmenttinäytöt ja ledit	22
6	ELEKTRONIIKKAOSUUDEN SUUNNITTELU	24
6.1	Testausliityntöjen määrän kartoitus	24
6.2	Mikro-ohjaimen valinta	25
6.3	Testausliityntä laajennuksen toteuttaminen	26
6.3.1	Vaihtoehtona sarjaliitettävät IO-laajennuspiirit	26
6.3.2	Valittu toteutustapa, rinnakkaisväylälaajennus	28
6.4	Teholähdeosio	33
6.4.1	Sähköiset arvot	34
6.4.2	Teholähteen rakenne	35
6.5	USB-liityntä	38
7	TESTERIN TOTEUTTAMINEN	42
8	TESTERIN LIITTÄMINEN PANEELIHIIN	44
9	TESTERIN LAITEOHJELMA	46
9.1	LCD-näytön lukeminen	47
9.2	Seitsensegmenttinäyttöjen ja ledien luku	49
9.3	Enkooderien pyöriksen simulointi	50
9.4	Näppäinpainalluksien simulointi	51

10	LOPPUTULOS JA JATKOKEHITYS	54
11	YHTEENVETO	56
	LÄHTEET	58
	LIITTEET	60

# 1 JOHDANTO

Koska ohjelmistot lisääntyvät erilaisissa laitteissa, niiden toiminnallisuudet monimutkaistuvat ja tuotteista tehdyt variaatiot lisääntyvät, testauksen merkitys tuotekehityksessä lisääntyy jatkuvasti. Testaus, ja etenkin ohjelmistotestaus, on merkittävä virstanpylväs tuotteen kehityksessä, joka oikein toteutettuna säästää myös rahaa pidemmällä aikavälillä. Riippuen tuotteen laadusta ja laajuudesta testausta voidaan tehdä monin eri tavoin. Ohjelmistojen testauksessa hyödynnetään usein myös automaattista testausta, jossa esimerkiksi PC-tietokone tekee ennalta ohjelmoidut testirutiinit ja kirjaa niissä esiintyvät poikkeamat.

Tämä opinnäytetyö käsittelee hitsauskoneiden käyttöliittymätesterin suunnittelua testiautomaatiokäyttöön. Työ toteutetaan hitsauskonevalmistaja Kemppi Oy:n toimeksiannosta. Testerillä tullaan testaamaan järjestelmähitsauskoneiden käyttöliittymäpaneeleja. Tarkoituksena on suunnitella PC-tietokoneeseen liitettävä testerilaite, joka nopeuttaa käyttöliittymäpaneeleissa käytettävien laiteohjelmistojen testausprosessia. Laitteen tarkoituksena on auttaa automatisoimaan toistuvia rutiininomaisia testejä, joita toistetaan useita kertoja testattavalle kohteelle, keventäen testaajien manuaalista työkuormaa. Työ koostuu laitteen elektroniikan ja piirilevyn suunnittelusta, laitteen sisältämän mikro-ohjaimen laiteohjelman kehittämisestä sekä Windows-käyttöjärjestelmälle kehitetystä komento-ohjelmasta, jolla testeriä ohjataan. Komento-ohjelman kehittämistä ei kuitenkaan tässä raportissa käsitellä, koska sen koettiin sisältävän liikaa ei-julkista tietoa.

Työn tekemistä tukevat omat kokemukset Kempillä tehtävästä ohjelmistotestauksesta, koska työskentelin kesän 2013 yrityksen tuotekehityksen ohjelmistoryhmässä. Tuona aikana pääsin mm. itse testaamaan erilaisia hitsauslaitteita. Tämän takia tiesin, millaisia testejä laitteille tehtäisiin. Olin kesällä myös kehittämässä testiautomaatiota laiteohjelmistotestaukseen, joka paransi ymmärrystäni testiautomaation toiminnasta ja rakenteesta.

Kemppi Oy on yksi maailman suurimpia ja johtavia kaarihitsauslaite valmistajia. Yritys tarjoaa myös erilaisia hitsausratkaisuja ja palveluja hitsausprosessin kehittämiseen. Ratkaisut sisältävät erilaisia hitsauskoneita eri käyttötarkoituksiin, hitsausprosessin automatisointiin liittyviä ratkaisuja sekä eritasoisia hitsauksen laa-

dunvalvontajärjestelmiä. Kempillä on kaksi tuotantolaitosta Suomessa ja yksi Intiassa. Toimipisteitä yrityksellä on 15 eri maassa. Yritys työllistää noin 630 henkilöä. Kemppi kehitti 1970-luvulla maailman ensimmäisen hitsausinvertterin, jossa laitteen toimintataajuutta on nostettu, jolloin laitteen sisältämä hitsausmuuntaja saadaan pienempään kokoon. Nykypäivänä tämä tekniikka on jo itsestäänselvyys lähes kaikissa hitsauskoneissa. Lisäksi Kemppi oli maailman ensimmäinen hitsauskonevalmistaja, joka siirtyi hitsauskoneissaan analogiaohjauksesta käyttämään digitaali ohjausta. (Kemppi Oy 2014b.)

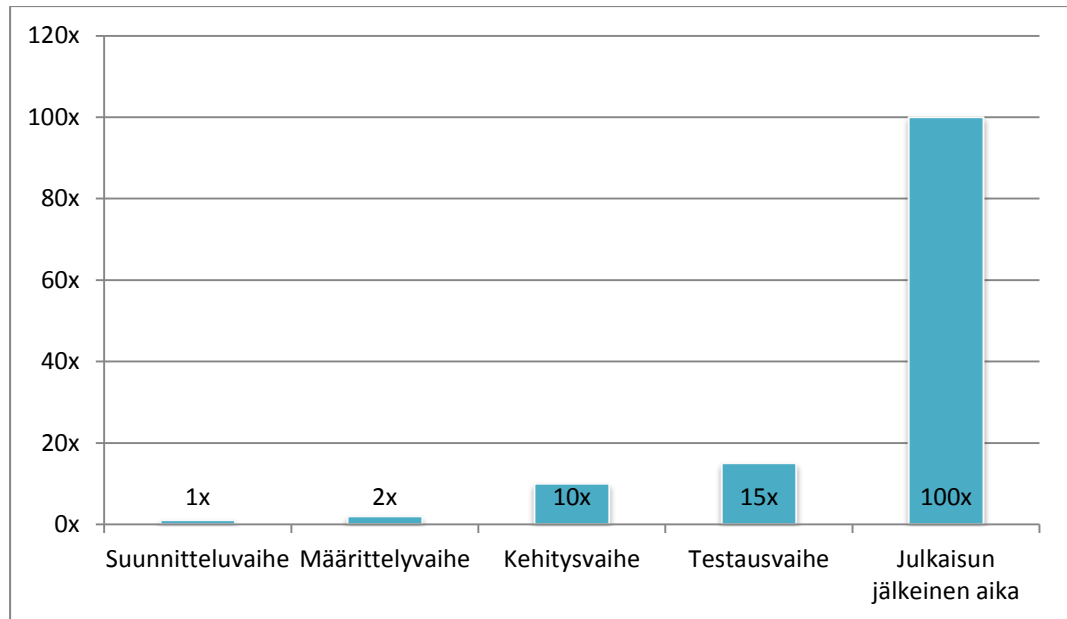


## 2 OHJELMISTOTESTAUS

Nykyisin ohjelmistotuotannon yksi merkittävä osa on ohjelmistotestaus, jolla varmistetaan tuotettavien ohjelmistojen korkea laatu. Ohjelmistotestausta tehdään erilaisille sovellusohjelmistoille kuin myös sulautettujen järjestelmien laiteohjelmistoille. Ohjelmistotestaus voidaan määritellä seuraavasti: se on vertailevaa työskentelyä, jossa testauksella varmistetaan, että ohjelmisto ja sen toiminnot vastaavat suunniteltuja määritteitä. (Kasurinen 2013, 10.)

Kaikki ohjelmistoja tuottavat yritykset testaavat nykyisin ohjelmistoja, mutta testauksen taso vaihtelee ohjelmiston laajuuden ja tärkeyden mukaan. Ohjelmistojen monimutkaistuessa ja niiden toimintojen lisääntyessä ohjelmistotestauksen merkitys on erittäin tärkeä. Riippuen ohjelmistosta ja sen käyttökohteesta virheellisen toiminnan seuraukset voivat vaihdella laidasta laitaan. Esimerkiksi joulukuussa 1999 Marsiin lähetetty NASA:n ”Mars Polar Lander” -avaruusluotaimen menetettiin yhteys sen oletetun laskeutumisen jälkeen. Tutkimuksissa ilmeni, että luotain on todennäköisesti tuhoutunut törmätessään planeetan pintaan laskeutumisvaiheessa. Varmaa syytä törmäykselle ei tiedetä, mutta NASA:n mukaan todennäköisin syy oli luotaimessa ollut ohjelmistovirhe. Luotaimen ohjelmisto tulkitsi virheellisesti anturien välittämää tietoa laskeutumisvaiheessa, minkä takia se luuli laskeutuneensa vaikka todellisuudessa se oli edelleen noin 40 m:n korkeudessa. Tämän takia luotain sammutti moottorinsa ja törmäsi planeetan pintaan suurella nopeudella. (NASA 2001.)

Ohjelmistoja testataan kehitysvaiheessa myös rahallisista syistä. Virheen korjaaminen ohjelmistosta kehitysvaiheessa on paljon halvempaa kuin silloin, kun ohjelmisto on jo julkaistu. Kuvio 1 esittää ohjelmistoissa korjattujen virheiden aiheuttamia kustannuksia periaatteellisesti. Kuvioista nähdään, että aikaisessa suunnitteluvaiheessa tunnistettujen ongelmien korjaaminen maksaa vain noin sadasosan siitä, mitä niiden korjaaminen maksaa ohjelmiston julkaisun jälkeen. (Kasurinen 2013, 17.)



KUVIO 1. Ohjelmistossa olevien virheiden suhteelliset korjauskustannukset eri vaiheissa (Kasurinen 2013, 18).

Ohjelmia voidaan testata käsitestauksena tai testausautomaatiolla. Manuaalisessa testauksessa testaaja tekee testitapauksen määrittelemän testin käsin ja kirjaa testissä esiintyvät poikkeamat ylös. Manuaalinen testaus soveltuu erityisesti testeille, jotka sisältävät liian monimutkaisia toimintoja ja muuttuvia tekijöitä, joiden takia testiä on hankala automatisoida. Lisäksi jos testiä ei tehdä kovin usein, tai testi on kertaluontoisesti suoritettava, se jätetään usein käsin testattavaksi. Käsin tehtävien testien on oltava suhteellisen lyhyitä, eikä niissä ei saa olla liikaa toistoja, jotta ne voidaan mahdollisen virhetilanteen jälkeen testata uudestaan.

Testausautomaatio mahdollistaa testien automatisoinnin, joka vähentää testaajien manuaalista työkuormaa. Testien automatisointi on erittäin kustannustehokasta silloin, kun testi sisältää helposti automatisoitavia toimintoja, runsaasti toistettavia kohtia tai testistä halutaan mahdollisimman tarkasti toistettava. Esimerkiksi testausautomaatio mahdollistaa sulautetun järjestelmän laiteohjelmiston perustestauksen uuden ohjelmistoversion ilmestyessä. Tällöin testausjärjestelmä voi käynnistyä automaattisesti ja tehdä joukon perustestejä uudelle ohjelmaversiolle. Testausautomaatiota voidaan hyödyntää myös testien osittaisessa automatisoinnissa. Tällöin vain osa testistä suoritetaan automaattisesti, mutta testaaja suorittaa käsin

tiettyjä kohtia. Vaikka testiautomaation käyttö vähentää testaajien työkuormaa ja tätä myöten tuo usein rahallisia säästöjä, sen käyttöönotto vaatii kuitenkin alkuinvestointeja. Riippuen siitä onko testattava kohde yksittäinen PC:n ohjelmisto vai kokonainen hitsauskonejärjestelmä testausympäristön toteuttaminen voi vaatia merkittäviä investointeja. PC-ohjelmiston tapauksessa usein riittää, että tietokoneelle asennetaan testausohjelmisto, jossa testausta suoritetaan. Taas testattaessa oikeaa hitsauskonejärjestelmää tarvitaan keino yhdistää hitsauslaitteisto tietokoneeseen, joka vaatii fyysistä laitteistoa. Testiautomaatiossa kaikki käytettävät testitapaukset on suunniteltava tarkkaan ja luotava käsin testijärjestelmään. Suunnittelussa on pyrittävä ottamaan huomioon mahdolliset virhetilanteet, jotta testijärjestelmä pystyy toipumaan niistä. Lisäksi testitapauksissa on mietittävä miten testattavan ominaisuuden tai toiminnon voidaan varmentaa toimivan oikeellisesti. Testiautomaatio on kuitenkin suurilla testausmäärillä tehokas keino vähentää kustannuksia tuotekehitysprosessissa.

### 3 JÄRJESTELMÄHITSAUSLAITTEISTOT

Kempin valmistamat hitsauskoneet voidaan jakaa kolmeen ryhmään koneiden käyttöympäristöjen ja koon perusteella. Ryhmät ovat kannettavat, kompaktit hitsauskoneet ja järjestelmähitsauslaitteistot. Kannettaviin hitsauskoneisiin kuuluvat esimerkiksi Minarc-tuoteperheen hitsauskoneet, jotka ovat pienikokoisia ja kevyitä, näin ollen helppoja kuljettaa. Kompakteihin koneisiin kuuluvat esimerkiksi Kempact-tuoteperheen koneet. Ne ovat kannettavia hitsauskoneita tehokkaampia ja sisältävät monipuolisempia hitsausominaisuuksia, mutta ovat kuitenkin kompakteja sisältäen kaiken yhdessä laitteessa. Suurempien hitsaustehojen takia kompaktit hitsauskoneet voivat sisältää myös mahdollisuuden hitsauspistoolin vesijäähdytykseen.

Järjestelmähitsauslaitteistot ovat modulaarisia hitsauslaiteratkaisuja, jotka on tarkoitettu raskaaseen teollisuuskäyttöön, kuten laivatelakoille. Järjestelmähitsauslaitteistoissa modulaarisuudella tarkoitetaan sitä, että hitsausvirtalähde ja -langansyöttölaite, jota käytetään MIG- ja MAG-hitsauksessa, ovat erillisiä yksiköitä. Ne voidaan sijoittaa pitkienkin matkojen päähän toisistaan. Tämä mahdollistaa pelkän langansyöttöyksikön sijoittamisen lähelle hitsattavaa kohdetta, kun taas virtalähde voidaan sijoittaa esimerkiksi sateelta suojattuun paikkaan. Lisäksi pelkän langansyöttöyksikön kuljettaminen on helpompaa sen pienemmän koon ja painon takia. Järjestelmähitsauslaitteistojen hitsaustehot ovat kompakteja hitsauskoneita suurempia, yltäen aina 500 ampeeriin. Suurempien hitsaustehojen ja pidempien hitsausaikojen takia järjestelmähitsauskoneet sisältävät lähes aina jäähdytyslaitteen hitsauspistoolille. Myös koneiden hitsausominaisuudet ovat loppuun asti hiottuja. Ne on optimoituja erityisesti teollisuustarkoituksiin, ja ne nopeuttavat ja parantavat hitsausprosessin laatua. Järjestelmähitsauslaitteistot ovat räätälöitävissä asiakkaan tarpeiden mukaan, jolloin asiakas voi valita haluamansa hitsausominaisuudet ja laitteet hitsausjärjestelmään. Lisäksi järjestelmää voidaan laajentaa jälkikäteen hankkimalla esimerkiksi toinen langansyöttölaite.



KUVIO 2. Kemppi FastMig X -järjestelmähitsauslaitteisto (Kemppi Oy 2014a.)

Kuviossa 2 on esitetty Kempin valmistama FastMig X -järjestelmähitsauslaitteisto, joka kuuluu Kempin teollisuuskäyttöön tarkoitettuihin hitsauslaitejärjestelmiin. Laitteiston perusta on moniprosessivirtalähde, joka mahdollistaa hitsauksen erilaisilla hitsausprosesseilla. Hitsausprosessit voivat olla tavallisia MIG- ja MAG-prosesseja tai esimerkiksi pulssitettuja erikoisprosesseja, joissa hitsausvirtaa ja langansyöttönopeutta vaihdellaan nopeassa tahdissa. Virtalähteellä voidaan tehdä myös MMA-hitsausta, jolloin langansyöttölaitetta ei tarvitse kytkeä. Virtalähdettä voidaan käyttää myös vakiojännite tai -virtalähteenä muille ulkoisille laitteille.

FastMig X tukee kolmea langansyöttölaitetta mahdollistaen erilaisilla hitsauslisäainelangoilla hitsaamisen ilman lankakelan vaihtamista kesken hitsauksen. Lisäksi virtalähteeseen voidaan liittää myös muita hitsauslaitteita, jotka saavat sähkönsä hitsausvirtaliitännöistä, kuten kuviossa kaksi vasemmalla näkyvä tuoteperheeseen kuulumaton MasterTig LT250 DC TIG -hitsauslaite. Kytkemällä erillinen TIG-hitsauslaite hitsausjärjestelmään mahdollistaa se myös tasavirralla tapahtuvan TIG-hitsauksen. Laitteisto tukee suurta määrää erilaisia lisälaitteita, kuten ArcQuality-hitsauslaadun valvontajärjestelmää, älykästä R30-hitsauskaukosäädintä ja SuperSnake-hitsauslanganvälisyöttäjää pitkille etäisyyksille. Laitteistoon on mah-

dollista ostaa uusia hitsauskäyriä ja -ominaisuuksia, jotka voidaan asentaa hitsauslaitteistoon DataGun-laitteella. DataGun:illa on mahdollista myös päivittää hitsausjärjestelmän laiteohjelmistot uusien laiteohjelmien ilmestyttyä.



KUVIO 3. FastMig X – X37-asetuskäyttöliittymä (Kemppi Oy 2014a).

Kuviossa 3 esitetyllä X37-asetuskäyttöliittymällä voidaan asettaa hitsauslaitteiston eri asetukset, tarkastella hitsauslaitteiston tietoja ja luoda hitsausmuistikanavat, joihin tallennetaan hitsauksessa käytettävät asetukset. Näin hitsauksessa käytettäviä asetuksia voidaan vaihtaa nopeasti ilman, että arvoja täytyy säätää käsin uudestaan. Muistikanaville voidaan luoda myös säätörajat, jotka rajaavat hitsauskäyttöliittymästä tehtävien hienosäätöjen suuruutta. Muistikanavia on mahdollista luoda 10 kappaletta yhdelle käyttäjälle. Käyttäjää on mahdollista olla 10. Lisäksi LCD-näytöllä näkyy yksityiskohtaisia tietoja hitsauksen aikana, kuten valokaaren todellinen jännite ja virta sekä langansyöttönopeus.



KUVIO 4. FastMig X – XF37-hitsauskäyttöliittymä (Kemppi Oy 2014a).

Kuviossa 4 on esitetty hitsauskäyttöliittymäpaneeli XF37. Tämä käyttöliittymä sijaitsee langansyöttölaitteessa. Paneelista voidaan esimerkiksi valita aktiivinen muistikkanava sekä säätää kanavan arvoja säätörajojen puitteissa, mikä mahdollistaa hitsausarvojen nopean hienosäädön.

Järjestelmähitsauslaitteiston monipuolisten ominaisuuksien takia lähes kaikki laitteet ja käyttöliittymät sisältävät oman mikroprosessorin tai mikro-ohjaimen. Laitteet kommunikoivat toistensa kanssa jatkuvasti, jolloin esimerkiksi asetuskäyttöliittymästä tehdyt muutokset päivittyvät välittömästi hitsauskäyttöliittymään sekä muille laitteille. Jotta järjestelmä kykenee laadukkaasiin hitsausominaisuuksiin, on hitsauksen aikana tapahtuvat eri toiminnot jaettu eri mikrotietokoneille. Näiden mikrotietokoneiden on toimittava synkronisesti, jotta hitsausominaisuudet toimivat laadukkaasti. Tämän takia järjestelmän ohjelmistotestaus on erittäin tärkeää. Jos jokin järjestelmän mikrotietokoneista ei toimi oikein, tarkoittaa se usein hitsausominaisuuksien huononemista. Järjestelmähitsauslaitteiston ohjelmistotestaus on erittäin tärkeää laitteiston myös suuren variaatiomäärän takia. Laitteisto voi sisältää eri määrän langansyöttölaitteita, erilaisia yhdistelmiä lisälaitteita ja joissakin tapauksissa myös muiden tuoteperheiden laitteita. Laitteistossa voi olla eri ohjelmaversioita, minkä takia myös eri versioiden yhteensopivuustestaus on tärkeää.

## 4 VAATIMUKSIEN MÄÄRITTELY

Paneeliteri suunniteltiin alustavasti Kempin FastMig X -järjestelmähitsauslaitteiden asetuskäyttöliittymä- ja hitsauskäyttöliittymäpaneeleille. Pääasiallinen vaatimusmäärittely testerilaitteelle tehtiin kyseisen laiteperheen vaatimusten perusteella. Aiemmin kuviossa 2 esitetty FastMig X -laitteisto sisälsi tuoteperheeseen kuulumattoman MasterTig LT250 TIG -hitsauslaitteen, jota ei kyseisen seikan takia huomioitu erityisemmin vaatimusten määrittelyssä. Testeria suunniteltaessa kuitenkin pyrittiin ottamaan myös sen yleiskäyttöisyys huomioon, jotta jatkossa sitä voitaisiin käyttää myös muiden hitsauslaiteperheiden käyttöliittymien testaamiseen, mutta pääasiallinen painoarvo oli FastMig X -tuotteiden tarpeilla.

### 4.1 Testausympäristö

Testattavat käyttöliittymäpaneelit tulevat olemaan kytkettyinä testeissä käytettävään toimivaan järjestelmähitsauslaitteistoon. Paneelit eivät toimi yksinään, jolloin on välttämätöntä, että testausympäristö laitteineen toimii kuten oikea järjestelmähitsauslaitteisto. Testausympäristön tulee vastata muiltakin osin oikeaa käyttöympäristöä, jotta testaustulokset ovat vertailukelpoisia oikean käyttöympäristön ja testausympäristön välillä. Tämän vuoksi paneeleihin ei myöskään tehdä mitään muutoksia testausta varten. Paneelit säilytetään sellaisina, että ne vastaavat täysin loppukäyttäjälle menevää tuotetta. Paneelit saavat käyttösähkensä testauksessa käytetystä hitsauslaitteistosta, jolloin testerin tehonlähteen ei tarvitse syöttää käyttöjännitteitä paneeleille. Tämä pienentää virranantovaatimusta testerin tehonlähteelle.

### 4.2 Testerin ja käyttöliittymien vuorovaikutustoiminnot

Käyttöliittymätesterin haluttiin mukailevan ihmiskäyttäjää, jolloin se kykenisi painelemaan sähköisesti käyttöliittymien painikkeita, sekä säätämään asetusarvoja rotaatioenkoodereilla. Jotta painikkeiden painelemisen ja asetusarvojen muuttamisen toiminnallisuudet käyttöliittymästä voitaisiin tarkistaa, tuli testerin sisältää myös ominaisuudet, joilla voidaan lukea käyttöliittymässä näkyvät tiedot. Näkyviin tietoihin lukeutuvat mm. näytöillä olevat tiedot ja paneeleissa palavat ledit.



Vaatimusmäärittelyä tehtäessä konenäön käyttämistä tietojen tarkistusta varten mietittiin yhtenä vaihtoehtona. Konenäössä tietokoneeseen liitetty kamera kuvaa tarkasteltavan kohteen ja yksinkertaisimmillaan vertaa ottamaansa kuvaa aiemmin otettuihin referenssikuviiin. Tällöin tietokone ilmoittaa, mikäli kuvat eroavat toisistaan. Edistyneemmät konenäkösovellukset osaavat erottaa kuvissa näkyvät tekstit, jolloin referenssikuvia ei tarvitse luoda jokaiselle tekstivaihtoehdolle erikseen. Kuitenkin konenäköidea hylättiin lähes saman tien sen esittämisen jälkeen. Syitä idean hylkäämiseksi olivat mm. suuri epävarmuus toimivuudesta ja toteuttamisen hankaluus. Etenkin ympäristön valaistuksen olisi täytynyt olla varsin stabiili, jotta kameran ottamat kuvat olisivat vertailukelpoisia. Samoin vertailukuvien luominen olisi ollut työlästä. Tämän takia tietojen takaisin lukeminen käyttöliittymistä tulee toteuttaa tutkimalla ja lukemalla käyttöliittymäkomponentteihin kirjoitettavia tietoja sähköisesti. Näin ollen valaistus tai muut ympäristöseikat eivät vaikuta testaukseen.

#### 4.3 Testausliittynät ja niiden kytkeminen

Järjestelmähitsauslaitteissa eri laitteet ja käyttöliittymäpaneelit kommunikoivat reaaliaikaisesti toistensa kanssa, jolloin esimerkiksi asetuspaneelista tehdyt asetusmuutokset päivittyvät välittömästi hitsauskäyttöliittymään ja langansyöttölaitteelle. Järjestelmähitsauslaitteiden käyttöliittymien ohjelmistotestaus suoritetaan usein kahden tai useamman käyttöliittymälaitteen ryhmänä, jotta varmistutaan eri laitteiden välisen tiedonsiirron toimivuudesta ja siitä, että tiedonsiirrossa ei synny ristiriitoja. Tämän takia testerin yhdeksi vaatimusmääritteeksi kirjattiin, että testerillä tulee olla mahdollista testata käyttöliittymäpaneeleita pareittain. Myöhemmin tässä dokumentissa yhteen käyttöliittymäpaneeliin kohdistuvaa testaussignaalien ryhmää kutsutaan testauskanavaksi.

Testausliittymäntöjen määrää ei määriteltä laitteen vaatimuksia kartoitettaessa, vaan määrittely jätettiin elektroniikan suunnitteluvaiheeseen. Testausliittymäntöjen määrä riippuu ratkaisevasti testattavasta paneelista ja sen komponenteista. Testattavan paneelin kytkeminen testeriin voidaan toteuttaa monella eri tavalla, mikä puolestaan vaikuttaa käytettävien testausliittymäntöjen määrään. Ennen testausliittymäntöjen määrän päättämistä tuli selvittää käyttöliittymäkomponenttien toimintatavat. Tie-

dettiin kuitenkin, että kaikki käyttöliittymäkomponentit paneeleissa toimivat digitaalisilla signaaleilla. Tämän takia testaussignaalitkin ovat puhtaasti digitaalisia, mikä helpottaa niiden toteuttamista. Testeriin voitaisiin myös toteuttaa analogisia tuloja ja lähtöjä, mikäli ne olisivat helposti toteutettavissa pääasiallisten vaatimusten täyttymisen jälkeen.

Testerin yleiskäyttöisyysvaatimus vaikutti omalta osaltaan testausliityntöjen sähköisiin ominaisuuksiin. Erilaiset testattavat käyttöliittymäpaneelit voivat sisältää erilaisia digitaalisia jännitetasoja. Tämän takia testausliityntöjen tulisi olla sellaisia, että ne kykenevät toimimaan erilaisilla jännitetasoilla. Ominaisuus toimisi samalla lievänä ylijännitesuojana testerille, mikäli jokin testausliitynnöistä kytkettäisiin vahingossa väärään signaaliin paneelissa. Paneelien erilaisten jännitetasojen takia, testausliitynnöissä käytettyjen lähtöjännitteiden tulisi olla myös valettavissa.

#### 4.4 Testerin liittäminen PC-tietokoneeseen

Nykyisin PC-tietokoneista löytyvät ulkoisten lisälaitteiden liittämistä varten lähes ainoastaan USB-liitynnät. Tämän takia tämä oli luonnollinen valinta paneelitesterin ja PC:n väliseen tiedonsiirtoon. USB-liityntä tarvitsee aina erikseen laitekoh- taisen ajurin PC-tietokoneelle, minkä takia liityntä päätettiin toteuttaa käyttämällä FTDI:n valmistamaa USB-UART-muunninpiiriä. Tällöin paneelitesterin USB-liityntä näkyy PC:lle virtuaalisena sarjaporttina, eikä PC:lle tarvitse erikseen tehdä laitekohtaista ajuria, vaan valmiin ajurin muunninpiirille saa FTDI:n kotisivuilta. FTDI:n valmistamat muunninpiirit ovat yleistyneet siinä määrin, että nykyisin esimerkiksi Microsoft Windows -käyttöjärjestelmät osaavat usein itse etsiä tarvit- tavan laiteajurin Windows Update -palvelusta. Myös itse testerin komento- ohjelman kehittäminen helpottuu, koska sarjaporttia voidaan käyttää helposti C++ ohjelmointikielessä. (FTDI 2009a, 5.)

Koska testeri liittyy käynnissä olevaan hitsauslaitteistoon, haluttiin PC galvaani- sestä erottaa testerilaitteesta vikatilanteiden varalta. Mikäli testauksen aikana hit- sausvirta pääsisi vikatilanteesta johtuen kulkeutumaan testeriin, galvaaninen ero- tus estäisi sitä rikkomasta PC-tietokonetta. Paneelitesteri tulee kuitenkin olemaan

paljon halvempi, kuin PC-tietokone, minkä takia testerin muuta elektroniikkaa ei tarvitse suojata tällaisten vikatilanteiden varalta.

#### 4.5 Muut työlle asetettavat vaatimukset

Elektroniikkasuunnittelun kannalta eräs tärkeä vaatimus oli, että laitteen suunnittelussa käytetään vain sillä hetkellä Kempin elektroniikkatuotannossa olevia komponentteja. Tämä oli myös oma henkilökohtainen vaatimus, sillä suunnittelu-aikaa ei nyt tarvitsisi käyttää uusien komponenttien mallien luomiseen piirilevyohjelmistoon. Tämä mahdollistaa testerin kasaamisen Kempin omalla elektroniikkatehtaalla ja nopeuttaa tuotantoa, koska kaikki komponentit löytyvät tuotannon varastosta. Samoin esimerkiksi SMD-ladontakoneet on ohjelmoitu tuotannossa oleville komponenteille. Pelkästään tuotannossa käytettyjen komponenttien käyttö suunnittelussa oli myös erinomainen testi nuorelle insinöörille. Siinä mitattiin osaamista, sopeutumista rajoitettuun komponenttivalikoimaan ja siitä johtuvaa innovointikykyä.

## 5 TESTAUKSEEN LIITTYVÄT KOMPONENTIT

Ennen laitteen elektroniikan suunnittelua tuli kartoittaa, mitä komponentteja testerilaitteen pitäisi testauksen yhteydessä käsitellä. Komponenttien toiminta- ja kytkentätavat asettivat myös tiettyjä vaatimuksia laitteen elektroniikalle, jolloin niiden toiminnot testaustilanteessa oli syytä tietää. Näiden tietojen perusteella testiin pystyttiin luomaan funktiot, joilla komponenteilta voidaan lukea tietoja tai niiden signaaleja voidaan manipuloida.

FastMig X -perheen XF37-hitsauskäyttöliittymäpaneeli (esitetty kuviossa 4) sisältää seuraavat käyttöliittymäkomponentit:

- 7 kpl seitsensegmenttinäyttöjä multipleksattuna
- 2 kpl (rotaatio)enkoodereita
- 9 kpl painonäppäimiä kytkinmatriisissa
- 21 kpl ledejä multipleksattuna.

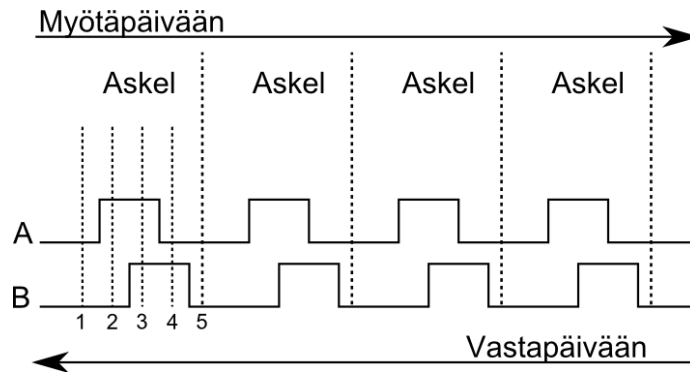
FastMig X -perheen X37-asetuskäyttöliittymäpaneeli (esitetty kuviossa 3) sisältää seuraavat käyttöliittymäkomponentit

- seitsensegmenttinäyttö
- alfanumeerinen LCD-näyttö
- 11 kpl painonäppäimiä kytkinmatriisissa
- (rotaatio)enkooderi.

### 5.1 Rotaatioenkooderi

Rotaatioenkooderi (nähtävissä kuviossa 4), lyhyemmin enkooderi, on sähkömekaaninen komponentti, joka muuttaa akselinsa pyöritysliikkeen digitaalseksi tiedoksi. Enkoodereilla on monia käyttötarkoituksia. Niitä voidaan käyttää moottoreissa kertomaan akselin asento ja pyörimisnopeus, tietokoneen hiirissä tai juurikin käyttöliittymissä. Enkooderin tuottama tieto voi olla absoluuttista, jolloin siitä käyvät ilmi akselin sen hetkinen täsmällinen asento, nopeus ja suunta. Tieto voi olla myös inkrementaalista, jolloin siitä käyvät ilmi vain akselin pyörimissuunta ja -nopeus. Käyttöliittymäpaneeleissa käytetyt enkooderit ovat inkrementaalityyppisiä ja malliltaan tarkoitettuja juurikin käyttöliittymissä käytettäväksi. Niiden

sisällä on kaksi erillistä kytkinkontaktia, jotka sulkeutuvat ja avautuvat akselia pyöritettäessä. Sulkeutuessaan ne kytkeytyvät komponentissa olevaan yhteiseen COM-nastaan. COM-nasta voi olla maadoitettu tai kytketty johonkin jännitteeseen, riippuen komponentille halutusta toiminnasta. Kytkinkontaktien liittynät on nimetty komponentissa liittynöiksi A ja B. Ne tuottavat akseli pyöritettäessä kaksi eri vaiheista digitaalisignaalia. Näistä erivaiheisista digitaalisignaaleista on mahdollista päätellä akselin pyörimissuunta ja -nopeus. Akselin absoluuttista asentoa tiedoista ei voi päätellä, ellei aiempi akselin asento ole tiedossa. Kuviossa 5 on esitetty inkrementaalisen enkooderin tuottama digitaali-tieto sen akselia pyöritettäessä. Kuviossa käyvät ilmi liittyntöjen eri vaiheet, kun akselia pyöritetään myötä- sekä vastapäivään. Kuviossa signaalit eivät edusta täsmällisiä jännitteitä, vaan sisäisten kytkinkontaktien sulkeutumista ja avautumista.



KUVIO 5. Enkooderin liittyntöjen periaatteelliset tilat akselia pyöritettäessä

Enkooderit voivat olla askellettuja, jolloin niitä pyöritettäessä askeleen kohdalla tuntuu selvä pykälä, johon akseli ikään kuin lukittuu. Yksi kierros voi komponentin mallista riippuen sisältää muutamasta aina satoihin askeliin. Komponentin mallista riippuen siinä ei välttämättä ole lukittuvia askelia, vaan komponentin akseli pyörii täysin vapaasti.

Yhden askeleen tuottamat signaalit voidaan jakaa tässä tapauksessa viiteen eri vaiheeseen, jotka on merkitty kuviossa 5 vasempaan laitaan numeroin. Vaiheessa yksi komponentin akselia ei ole vielä pyöritetty, jolloin A ja B signaalit ovat lepo-tilassa. Vaiheessa kaksi akselia on alettu pyörittämään myötäpäivään ja A-liittynnän kontakti kytkeytyy yhteen COM-nastan kanssa. Vaiheessa kolme myös

B-liitynnän kontakti kytkeytyy COM-nastaan. Vaiheessa neljä A-liitynnän kontakti avautuu kytkeytyen irti COM-nastasta ja vaiheessa viisi B-liitynnän kontakti avautuu myös irtautuen COM-nastasta. Tässä tilanteessa enkooderin akseli on pyörähtänyt yhden askeleen myötöpäivään. Jos akselia pyöritettäisiin vastapäivään, luettaisiin kuviossa 5 esitettyä kaaviota oikealta, jolloin vaiheiden järjestys olisi 5, 4, 3, 2 ja 1.

## 5.2 Alfanumeerinen LCD-näyttö

Alfanumeerinen LCD-näyttö (nähtävissä kuviossa 3) on tarkoitettu tiedon esittämiseen pääasiassa numeroin ja kirjaimin. Kyseinen näyttötyyppi soveltuu huonosti grafiikan esittämiseen, sillä se koostuu erillisistä merkkilohkoista, jotka sisältävät yhden näytettävän merkin. Tyypillisesti yksi merkkilohko koostuu 5x8 pikselin kokoisesta matriisista. Näytöstä riippuen yhdelle riville mahtuu 8, 16, 20, 24, 40 tai 80 merkkiä. Rivejä näytössä on tyypillisesti 1, 2 tai 4. (Koskinen 2004, 233.)

Tämän tyyppiset näyttömoduulit sisältävät usein yksinkertaisen näytönohjaimen, näyttömuistin ja merkkimuistin. Näytönohjain nimensä mukaisesti hoitaa näytön ohjaamisen. Sitä voidaan komentaa esimerkiksi liikuttamaan kursori tiettyyn kohtaan näyttöä tai vaikka tyhjentämään näyttö. Näyttömuisti sisältää näyttöön tulostuvan datan. Näyttömuistiin kirjoittaminen hoidetaan datarekisterin kautta merkki kerrallaan. Merkkimuisti voidaan vielä jakaa kiinteään ja erikoismerkkimuistiin. Kiinteä merkkimuisti sisältää numerot, pienet ja suuret kirjaimet sekä joitain valmiita erikoismerkkejä, jotka voidaan esittää näytöllä. Erikoismerkkimuistiin on mahdollista luoda omia merkkejä, joita ei ole valmiina kiinteässä merkkimuistissa. Esimerkiksi lämpömittarisovelluksessa celsiusasteen merkki voitaisiin luoda omaksi erikoismerkiksi, mikäli sitä ei ole valmiina kiinteässä merkkimuistissa. (Koskinen 2004, 233.)

Näytölle tulostettavat merkit annetaan moduulille usein standardeina ASCII-koodeina, mutta jotkin erikoismerkit, kuten skandinaaviset kirjaimet, annetaan poikkeavilla koodeilla. Näytöt tukevat usein vain englanninkielessä käytettäviä merkkejä ASCII-koodeina. Skandinaavisten merkkien käyttö vaatii myös sen, että ne ovat yleensäkin määritetty näytönohjaimen tukemiin merkkeihin. Näytönohjain

tulkitsen näyttömuistissa olevien merkkien koodit ja tulostaa näytölle merkkimuis-  
tin koodia vastaavan muistipaikan sisältämän merkin. (Koskinen 2004, 233.)

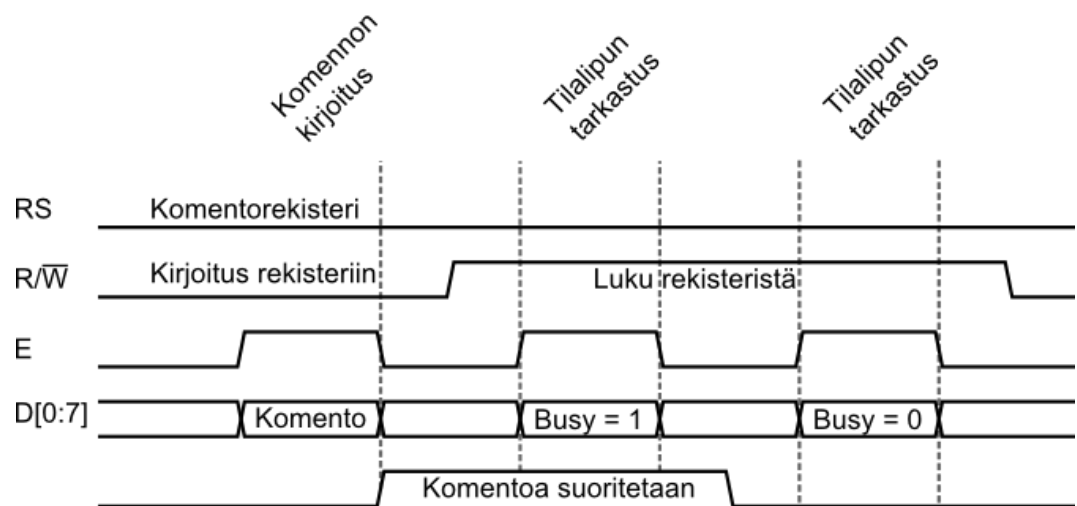
Suuri osa alfanumeerisista LCD-näyttömoduuleista käyttää liityntärajapintaa, joka on alun perin tarkoitettu kytkettäväksi suoraan moduulia ohjaavan mikroprosessorin osoite- ja dataväyliin, helpottaen moduulin kytkentää ja käyttöä. Kuitenkin näyttö voidaan kytkeä mihin tahansa IO-liityntöihin. Liityntärajapinta koostuu kolmesta yksisuuntaisesta ohjaussignaalista ja kahdeksasta kaksisuuntaisesta datasi-  
gnaalista. Ohjaussignaaleja ovat E, RS ja RW. E-signaali on valintasi-  
gnaali, jonka laskevalla reunalla data kirjoittuu näyttömoduulille. RS-signaalilla valitaan kohdistuuko suoritettava operaatio komento- vai datarekisteriin. RW-signaalilla kerrotaan, onko operaatio kirjoitus vai luku. Liityntärajapinnassa on myös kahdeksan kaksisuuntaista datasi-  
gnaalia D0 - D7. Näillä hoidetaan moduulin ja sitä ohjaavan mikroprosessorin tai -ohjaimen välinen tiedonsiirto. Datan siirtäminen voidaan tehdä joko kahdeksan bitin siirtämisenä kerralla tai kahdessa neljän bitin palassa. Toimintatila valitaan toimintarekisteristä, kun näyttöä alustetaan sähköjen kytkettyä. Kahdeksan bitin kertakirjoituksessa data siirretään sellaisenaan yhdellä operaatiolla. Neljä-bittisessä toimintatilassa datasi-  
gnaaleista käytetään vain D4 - D7 signaaleja. Tämä vähentää kytkettävien IO-liityntöjen määrää, koska loput datalinjat voidaan jättää kytkemättä. (Koskinen 2004, 233-235.)

Neljäbittisessä toimintatilassa kahdeksanbittinen data siirretään kahdessa osassa. Ensin siirretään puolikas, joka sisältää neljä eniten merkitsevää bittiä. Tämän jälkeen siirretään jäljelle jäänyt puolikas, joka sisältää neljä vähiten merkitsevää bittiä. Näytölle siirrettävien datan ja komentojen ajastuskaaviot ovat esitetty myöhemmin kuvioissa 6 ja 7, joissa näyttö on kahdeksan bittisessä toimintatilassa. (Hitachi 1999, 33.)

Edellä esitelty liityntä on jo muodostunut nykypäivänä yleiseksi standardiksi, jota suuri osa näyttöjen valmistajista käyttää. Tästä liitynnästä käytetään usein, etenkin harrastelijapiireissä, epävirallista nimitystä ”HD44780-liityntä”. Tämä nimitys on peräisin paljon alfanumeerisissa LCD-näytöissä käytetystä Hitachin HD44780-näytönohjainpiiristä. Nimityksellä usein viitataan myös siihen, että liityntää käyttävä näyttö on myös komentojensa puolesta yhteensopiva HD44780-näytönohjaimen kanssa. Myös monissa LCD-näyttöjen datalehdissä on maininta

HD44780 yhteensopivuudesta, jos näytössä käytettävä näyttöohjain ei ole sama kuin edellä mainittu, mutta toimii samalla tavalla.

Kuvion 6 aikakaaviossa on esitetty tyypillinen komennon kirjoitustapahtuma. Tässä tapauksessa näytölle kirjoitetaan komento, jota näyttö alkaa suorittaa. Komennon kirjoittamisen jälkeen sen loppumista tarkkaillaan lukemalla komentorekisteriä, koska näyttöohjain ei voi suorittaa kuin yhden komennon kerrallaan. Komentorekisterissä eniten merkitsevä bitti on näyttöohjaimen tilalippu. Se ilmaisee, mikäli näyttöohjain suorittaa vielä edellistä komentoa. Mikäli näyttöohjaimella on komennon suoritus kesken, on tilalippu asettunut ykköstitilaan. Kun näyttöohjain saa suoritettua sille annetun komennon, tilalippu nollautuu. (Hitachi 1999, 24, 29, 32.)



KUVIO 6. Komennon kirjoittaminen ja tilalipun luku

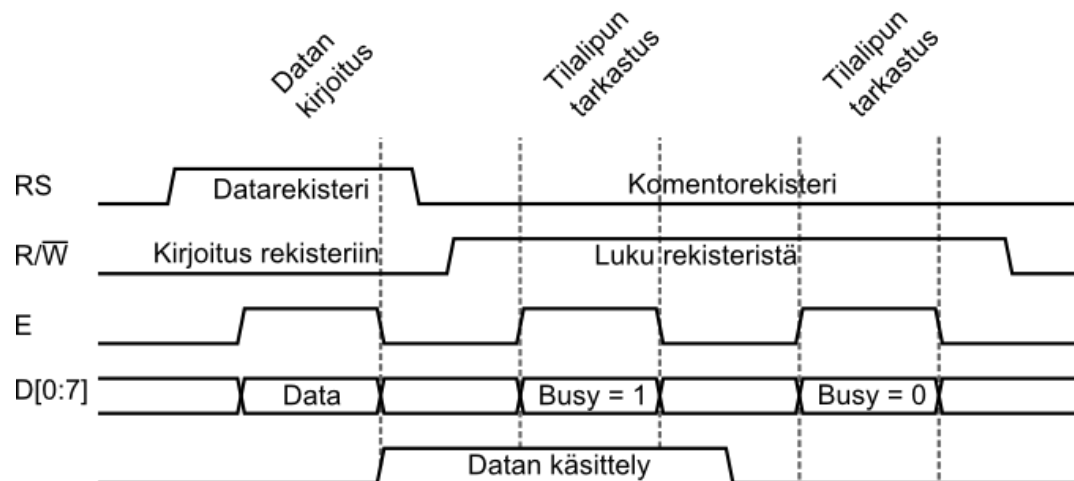
Näyttöön tulostettavan datan kirjoittaminen etenee lähes samalla tavalla, kuin komennon kirjoittaminen. Tyypillinen kirjoitustapahtuma on esitetty kuviossa 7. Poiketen komennon kirjoitustapahtumasta dataa kirjoitettaessa RS-signaali on ylätilassa.

Datan oikein tulostumisen edellytyksenä on, että näyttöohjaimen osoitelaskuri on asetettu haluttuun alkupisteeseen. Tämä tapahtuu esimerkiksi kirjoittamalla näytölle ”Set DD RAM address” -komento, joka kirjoittaa näyttöohjaimen osoitelaskuriin halutun arvon. Komennon bitit D0-D6 ovat haluttu osoitelaskurin arvo.



Osoitelaskurin arvo voidaan myös lukea komentorekisteristä samalla tavalla kuin tilalippu. Luettaessa komentorekisteriä näyttönohjain kirjoittaa nykyisen osoitelaskurin arvon bitteihin D0-D6. Näytölle voidaan myös antaa ”Clear display” -komento, joka tyhjentää näytön ja asettaa osoitelaskurin osoittamaan näyttömuistin ensimmäiseen muistipaikkaan. Kyseinen komento on näyttönohjaimelle melko hidas, minkä takia sitä tulee käyttää harkitusti. Jos kyseistä komentoa kutsutaan liian tiheään tahtiin, se voi aiheuttaa näytön välkkymistä häiritsevästi. (Hitachi 1999, 29 31.)

Sen jälkeen kun kursori on asetettu oikeaan paikkaan, voidaan data kirjoittaa näytölle. Datan kirjoitus tehdään merkki kerrallaan. Merkkien kirjoitusten välillä osoitelaskuria ei tarvitse erikseen muuttaa, koska näyttönohjain hoitaa sen itse. Komennolla ”Entry Mode Set” voidaan valita, lisääkö vai vähentääkö näyttönohjain osoitelaskuria muistiin kohdistuneen kirjoitus- tai lukuoperaation. Komennolla voidaan myös valita, siirretäänkö samalla näytöllä näkyviä merkkejä vai pelkistään kursoria. Mikäli dataa kirjoitetaan ei-peräkkäisiin muistiosoitteisiin, täytyy siinä tapauksessa osoitelaskurin arvot asettaa käsin. (Hitachi 1999, 26.)



KUVIO 7. Yksittäisen datan (merkin) kirjoittaminen ja tilalipun luku

Kuten edellä mainittiin, näyttö täytyy alustaa ennen sen käyttöä jokaisen sähköjen kytkemiskerran jälkeen. Näyttömoduulit sisältävät myös sisäisen reset-kytkennän, joka alustaa näytön ns. tehdasasetuksilla. Tällöin näytön asetukset ovat usein riittävät ”peruskäyttöä” varten. Tehdasasetuksilla näyttö on kuitenkin pois päältä,

jolloin tulostettavat merkit eivät ole näkyvissä. Näytön asettaminen päälle tapahtuu ”Display on/off” -komennolla. Kyseisellä komennolla voidaan myös vaikuttaa näytön kursorin näkyvyyteen ja vilkkumiseen.

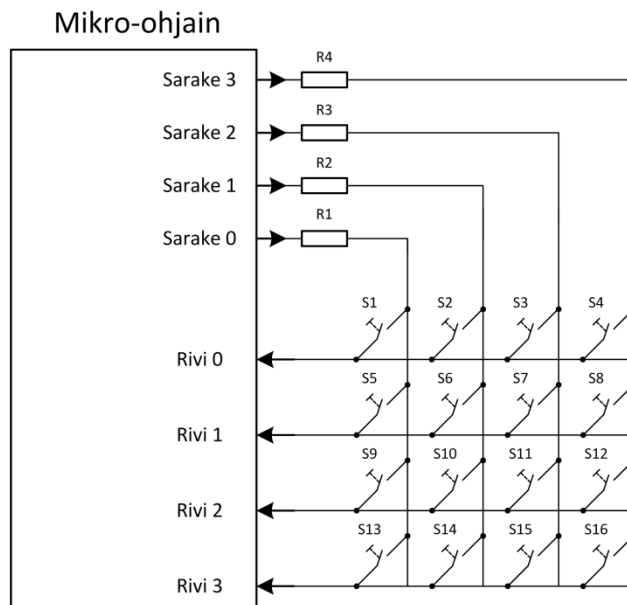
Näytön toiminta on kuitenkin luotettavampaa, jos alustusrutiinit suoritetaan käsin. Lisäksi jos näyttö halutaan asettaa 4-bittiseen toimintatilaan, on alustusrutiini suoritettava joka tapauksessa. Alustettaessa näytön komentorekisteriin kirjoitetaan ns. ”Function Set” -komento. Komennon kirjoitus noudattaa kuviossa 6 esitettyä tapaa, pois lukien tilalipun lukemisen. Alustuskomennon bittien 2, 3 ja 4 arvot määräytyvät näytölle halutun toimintatilan mukaan. Biteillä 2 ja 3 määritellään näytön rivien määrä, ja käytettävien merkkien koko. Bitti 4 määrittelee, käyttääkö näyttö kahdeksaa vai neljää datasiignaalia. Kyseinen komento on siitä erityinen, että se on oltava ensimmäinen komento, joka näytölle kirjoitetaan, jos sisäinen reset-kytkentä ei kykene alustamaan näyttöä. Samoin kyseinen komento on annettava kahdesti peräkkäin, eikä sen suorituksen päättymistä voida todeta tilarekisteristä tilalippua lukemalla. Tämän alustuskomennon jälkeen näytölle voidaan asettaa halutut asetukset, kuten kursorin näkyvyys ja sen vilkkuminen. Tietysti näyttö täytyy jossain vaiheessa asettaa päälle, jotta tulostettavat merkit näkyisivät näytöllä. (Hitachi 1999, 27.)

### 5.3 Näppäinmatriisi

Matriisikytkentä on paljon käytetty kytkentäteknikka, jolla pyritään minimoimaan tarvittavien IO-liityntöjen määrä. Esimerkiksi kytkettäessä 36 näppäintä yksitellen suoraan mikro-ohjaimen, vaatisi kytkeminen 36 IO-liityntää mikro-ohjaimesta. Sama määrä näppäimiä voidaan kytkeä matriisikytkennällä, jolloin tarvittavien IO-liityntöjen määrä vähenee 12:een. Tällöin puhutaan näppäin- tai näppäimistömatrisista. (Koskinen 2004, 216.)

Näppäinmatriisin signaalit voidaan jakaa sarake- ja riviliityntöihin. Näppäimien lukeminen perustuu siihen, että toisia näistä liitynnöistä luetaan ja toisia liityntöjä kytkentään yksi kerrallaan aktiiviseksi. Se, mitä aktiivinen tila edustaa, voi tarkoittaa, että liityntä kytketään jännitteeseen tai maahan riippuen käytettävästä kytkennästä. Kuviossa 8 on esitetty yksinkertainen esimerkki näppäinmatriisikytkennästä. Kyseisessä näppäinmatriisissa on neljä rivi- ja neljä sarakeliityntää.

Näppäinmatriisi on esimerkissä kytketty suoraan mikro-ohjaimen IO-liityntöihin yksinkertaisuuden vuoksi. Kuviossa esitetyssä kytkennässä näppäimien liitynnät kytkeytyvät aktiivisessa tilassa jännitteeseen, jonka mikro-ohjain syöttää sarakeliityntöihin. Näppäimien tilatiedot luetaan riviliitynnöistä, jotka asetetaan sisäänuloiksi. Kytkentään on lisätty sarjavastukset vähentämään näppäimien läpi kulkevaa virtaa, mikäli riviliitynnät asetetaan ulostuloiksi ja aiheuttavat näppäintä painettaessa mahdollisen oikosulkutilanteen.



KUVIO 8. Esimerkki kuudentoista näppäimen matriisikytkennästä

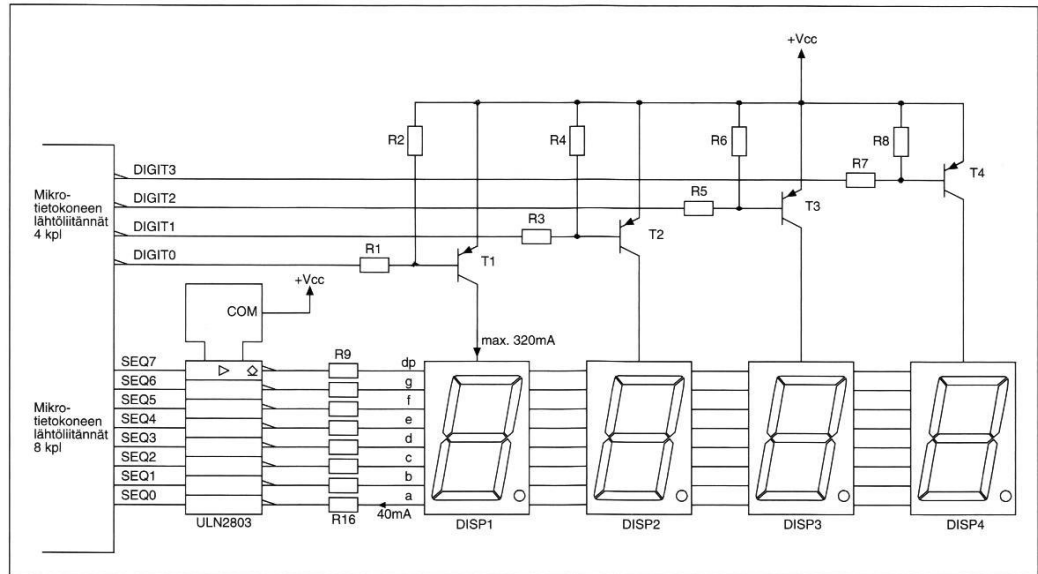
Esimerkin näppäimistön lukeminen alkaa siten, että mikro-ohjain asettaa ”sarake 0” -liitynnän ylätilaan. Tämän jälkeen se lukee riviliityntöjen tilat. Mikäli jokin riviliitynnöistä on ylätilassa eli kytkeytyneenä jännitteeseen, on jotakin sarakkeen 0 näppäimistä painettu. Esimerkiksi, jos ”rivi 1” -liityntä on ylätilassa, näppäintä S5 on painettu. Riviliityntöjen lukemisen jälkeen mikro-ohjain asettaa ”sarake 0” -liitynnän alatilaa ja asettaa ”sarake 1” -liitynnän ylätilaan. Mikro-ohjain lukee taas riviliityntöjen tilat ja jos jokin riviliitynnöistä on ylätilassa, on sarake yhden jotain näppäintä painettu. Mikro-ohjain toistaa tätä, kunnes se on käynyt kaikki sarakeliitynnät läpi, minkä jälkeen lukeminen alkaa alusta.

#### 5.4 Seitsensegmenttinäytöt ja ledit

Aiemmin näppäinmatriisin yhteydessä esitettyä kytkentäteknikkaa voidaan hyödyntää myös muille komponenteille, kuten ledeille ja seitsensegmenttinäytöille. Molemmat edellä mainitut komponentit ovat optisia komponentteja ja jakavat saman toimintaidean. Optisten komponenttien yhteydessä kytkentään voidaan viitata myös multipleksaus-nimellä. Tässä kappaleessa keskitytään seitsensegmenttinäyttöihin, mutta kaikki esille tulevat asiat pätevät myös yksittäisille ledeille.

Seitsensegmenttinäytöissä yksittäinen segmentti sisältää tyypillisesti yhden ledin. Kytkettäessä tietyt segmentit päälle näytölle voidaan muodostaa numeroita ja joi-takin kirjaimia. Yksittäinen seitsensegmenttinäyttö sisältää vähintään seitsemän liityntää segmenteille ja mikäli näytössä olevaa desimaalipistettä käytetään, liityntöjä on yhteensä kahdeksan. Tämän takia useampia näyttöjä tapauksissa kytkennän käyttö on järkevää. Kyseistä tekniikkaa on käytetty myös XF37-hitsauskäyttöliittymässä, jotta käytettyjen IO-liityntöjen määrä on saatu minimoitua. Hitsauskäyttöliittymä sisältää seitsemän kappaletta seitsensegmenttinäyttöjä, jolloin yksittäisessä näyttöjen kytkemisessä liityntöjä tarvittaisiin 56 kappaletta.

Kuviossa 9 on esitetty esimerkki neljän seitsensegmenttinäytön kytkemisestä multipleksaamalla. Kuviossa käytetyt näytöt ovat CA-tyyppisiä, eli kaikkien segmenttien yhteinen liityntä on anodi. Kytkennän toiminta perustuu siihen, että näyttöjä vaihdellaan nopeassa tahdissa vuorotellen aktiiviseksi. Vain yksi näyttö on kerrallaan aktiivisena, ja vain aktiivisessa näytössä näytetään siihen kuuluva numero tai merkki. Muut näytöt ovat sen aikaa sammuksissa. Näyttöjen vuorottelutaajuus niin suuri, että ihmissilmä ei kerkeä erottamaan yksittäisten näyttöjen syttymistä ja sammumista, vaan kaikki näytöt näyttävät palavan yhtä aikaa. Kaikki kytkennässä käytetyt näytöt jakavat segmenttiliitynnät toistensa kanssa, koska vain yksi näytöistä on aktiivinen kerrallaan.



KUVIO 9. Neljän seitsensegmenttinäytön kytkentä (Koskinen 2004, 224).

Näyttöjen aktivoiminen tapahtuu kuviossa 9 näkyvillä PNP-transistoreilla. Haluttaessa jokin näyttö aktiiviseksi, asetetaan kyseiseen näyttöön liittyvä DIGIT-signaali alatilaaan. Tällöin transistori alkaa johtaa ja jännite pääsee kytkeytymään näyttölle. Kuviossa näkyy myös kahdeksanporttinen ULN2803-puskuripiiri näyttöjen segmenttilinjoille. Tämän piirin tarkoitus on ohjata näyttöjen segmenttejä päälle kytkemällä haluttu segmentti maahan. Segmenttilinjojen virrat ovat sen verran suuria, että pelkkä mikro-ohjaimen IO-liityntä ei kestäisi sitä ilman puskurointia.

## 6 ELEKTRONIIKKAOSUUDEN SUUNNITTELU

Tämä osuus käsittelee testerin elektroniikan suunnittelua. Osuudessa on käyty suunnittelun vaiheet läpi siinä järjestyksessä, kuin ne toteutettiin. Suunnittelun ensimmäinen vaihe oli miettiä, millaiset sähköiset testausliitännät laite tarvitsee paneeleihin liittymiseen, ja kartoittaa niiden lukumäärä. Tämä vaihe vaikutti oleellisesti testerissä käytettävän mikro-ohjaimen valintaan, koska testausliitintöjen toteutustapa ja lukumäärä määrittelevät mikro-ohjaimelta vaadittujen ohjaussignaalien lukumäärän.

Testerilaitteen kytkentäkaavio löytyy liitteestä 1. Kyseisestä kytkentäkaaviosta on poistettu joitain toiminnallisia lohkoja, joita ei haluttu julkaista. Kyseiset lohkot eivät vaikuta testerin perustoimintaan, eivätkä ne kuuluneet opinnäytetyön vaatimuksiin, vaan ne toivat tiettyjä lisätoiminnallisuuksia laitteeseen.

### 6.1 Testausliitintöjen määrän kartoitus

X37-paneelissa olevien käyttöliittymäkomponenttien takia testausliitintöjä tarvitaan ainakin 29 kappaletta kyseiselle paneelille. XF37-hitsauskäyttöliittymää varten testausliitintöjä tarvitaan 24 kappaletta. Osa testausliitinnöistä voi olla puhtaasti pelkkiä sisääntuloja, joista luetaan tietoa. Esimerkiksi testauksessa LCD-näytön tietojen lukeminen riittää, eikä sen tietoja ole tarvetta manipuloida. Puolestaan esimerkiksi rotaatioenkooderien ja näppäinmatriisien tietoja on tarve manipuloida näppäinpainalluksia simuloitaessa, minkä takia näiden testausliitintöjen tulisi olla ulostuloja. Nämä testausliitinnät päätettiin toteuttaa siten, että ne voivat toimia sekä ulostuloina että sisääntuloina.

Aiemmin mainittujen testausliitintävaatimusten takia testausliitintöjä päätettiin toteuttaa yhteensä 32 kappaletta yhtä testauskanavaa kohden. Nämä testausliitinnät jakautuisivat siten, että 16 liitintää olisi tulotyyppisiä, joilla voidaan vain lukea tietoa. Loput 16 liitintää olisivat kaksisuuntaisia liitintöjä, joita voidaan myös lukea, mutta niitä voidaan käyttää digitaalisina lähtöinä. Tämä tekee testeristä yleiskäyttöisemmän, ja sillä voidaan testata jatkossa myös muita käyttöliittymäpaneeleita. Näin toteutettuna testaussignaaleja testerissä on kaikkiaan 64 kappaletta.

ta. Testerissä oleva mikro-ohjain tulisi lukemaan ja manipuloimaan näiden testausliityntöihin kytkeytyviä signaaleja.

## 6.2 Mikro-ohjaimen valinta

Testausliityntöjen määrän kartoituksen jälkeen suunnittelussa edettiin mikro-ohjaimen valintaan. Mikro-ohjaimen ensimmäinen valintakriteeri oli sen sisältämien yleiskäyttöisten IO-liityntöjen määrä. IO-liityntöjä tulisi olla riittävästi, jotta testausliitynnät voitaisiin toteuttaa. Kuitenkin mikro-ohjaimen sisältämä IO-määrä tulisi olla testausliityntöjen määrää suurempi, koska testausliityntöjen määrään ei sisälly esimerkiksi PC:n kanssa kommunikointiin käytetty sarjaliikenneliityntä.

Mikro-ohjain lukisi testattavan käyttöliittymäpaneelin komponenttien tiedot sekä hoitaisi näppäinpainalluksia simuloitaessa tietojen manipuloimisen, minkä takia sen tulisi olla myös nopea suorittamaan erilaisia käskyjä ja operaatioita. Etenkin LCD-näytölle kirjoitettavien tietojen lukemisessa mikro-ohjaimen on oltava nopea, koska näytölle kirjoitettavan datan kirjoitustaajuus voi olla useita megahertsejä. Samoin näppäinpainallusten manipulointitietojen tahdistaminen näppäinmatriisin signaaleihin vaatii nopeita toimintoja, jotta testattava paneeli tulkitsee simuloitua näppäinpainallukset oikein.

Näiden valintakriteerien perusteella mikro-ohjaimeksi valittiin NXP:n LPC2387 32-bittinen ARM7-mikro-ohjain. Kyseisen mikro-ohjaimen prosessoriydin on ARM7TDMI-S, jonka suurin toimintataajuus on 72 MHz. Ohjelmamuistia mikro-ohjain sisältää 512 kt. Mikro-ohjain sisältää 70 kappaletta yleiskäyttöisiä GPIO-liityntöjä (General Purpose Input Output), joita voidaan käyttää sekä tuloina että lähtöinä. Nämä liitynnät voivat myös toimia mikro-ohjaimen sisäisten oheislaitteiden toimintaliityntöinä. Yhteen liityntänastaan voi kytkeytyä useita sisäisiä oheislaitteita, mutta vain yksi oheislaitte voidaan valita liitännään. Liityntöjen toimintatila valitaan ohjelmallisesti PINSEL-rekistereillä.

Muita mikro-ohjaimen merkittäviä yksiköitä testerin jatkokehitystä ajatellen ovat mm. 10-bittinen AD-muunnin, joka voidaan kytkeä kuuteen eri ulkoiseen liityntään. Tämä yksikkö mahdollistaisi testerille analogisen jännitteen mittaamisen. Mikro-ohjain sisältää myös 10-bittisen DA-muuntimen sekä kuusi PWM-lähtöä.

Näillä voisi tarvittaessa tuottaa myös analogisia jännitteitä, joita voitaisiin käyttää esimerkiksi analogisen hitsauskaukosäätimen signaaleja manipuloitaessa.

### 6.3 Testausliityntä laajennuksen toteuttaminen

Mikro-ohjaimen valinnan yhteydessä kävi ilmi, että ei ole järkevää yrittää toteuttaa testausliityntöjä siten, että yksittäiset testausliitynnät kytketään suoraan mikro-ohjaimen GPIO-liityntöihin. Testausliityntöjä on tarkoitus toteuttaa 64 kappaletta, jolloin LPC2387:n GPIO-liitynnöistä jäisi vain kuusi vapaaksi muuhun käyttöön. Samalla myös testausliitynnät hajaantuisivat mikro-ohjaimen eri portteihin, jolloin eri GPIO-liitynnöille suoritettavat operaatiot hankaloituisivat. Tällöin liityntöjen tilat jouduttaisiin kirjoittamaan ja lukemaan eri rekistereistä. Lukuoperaatioissa testausliityntöjen tiloja jouduttaisiin yhdistelemään eri rekisterien arvoista, jolloin testerin suorituskyky heikkenisi.

Vaikka kyseisessä mikro-ohjaimessa GPIO-portit ovat nimellisesti 32-bittisiä, eivät ne välttämättä sisällä 32:a liityntää. Esimerkiksi kyseisen mikro-ohjaimen GPIO-portissa P0, ei ole käytettävissä liityntöjä P0.12, P0.13, P0.14 ja P0.31, minkä takia P0-portissa onkin vain 28 liityntää. Lisäksi LPC2387-mikro-ohjaimessa GPIO-liityntöjen keskeytykset ovat käytettävissä vain porteissa P0 ja P2. Tällöin P1-portissa olevista testausliitynnöistä puuttuisi mahdollisuus käyttää esimerkiksi signaalissa olevan nousevan- tai laskevanreunan aiheuttamaa keskeytystä, mikäli testausliitynnät kytkettäisiin suoraan mikro-ohjaimen portteihin. Aikakriittisissä operaatioissa ja erilaisissa tahdistuksissa keskeytysten käyttö on erittäin suotavaa, minkä takia testausliityntöjen tulisi olla sellaisia, että ne voidaan tarvittaessa asettaa antamaan keskeytys. Keskeytyksiä ei kuitenkaan ole tarve käyttää kaikkia yhtä aikaa, mutta niiden tulisi tarpeen tullen olla käytettävissä kaikissa testausliitynnöissä.

#### 6.3.1 Vaihtoehtona sarjaliitettävät IO-laajennuspiirit

Yksi toteutustavoista olisi ollut käyttää erillisiä sarjaliitettäviä IO-laajennuspiirejä, jotka liittyisivät mikro-ohjaimen esimerkiksi SPI-väylällä. Yksi tällainen laajennuspiiri sisältää useita yleiskäyttöisiä IO-liityntöjä, jotka voidaan ohjelmoida tuloiksi tai lähdöiksi. Esimerkiksi Maxim Integratedin valmistama MAX7301 on



16-bittinen SPI-liitettävä IO-laajennuspiiri, joka sisältää 20 tai 28 yleiskäyttöistä IO-liityntää riippuen piirin kotelovaihtoehdosta. 16-bittisyys piirin yhteydessä tarkoittaa sitä, että piirin sisäinen yhdistetty osoite- ja datasiirtorekisteri on 16 bittiä pitkä, josta 8 bittiä on varattu piirin rekisteriosoitteelle ja 8 bittiä rekisteriin kirjoitettavalle datalle. Näin ollen piirin rekistereihin kirjoitettaessa väylän yli siirrettävän datan pituus on 16 bittiä. (Maxim Integrated 2006.)

Luettaessa piiriltä jonkin rekisterin sisältö siirretään piirille ensin 8-bittinen osoitetieto, jossa R/W-bitti on asetettu ylätilaan. Tämän jälkeen piirille siirretään 8-bittinen täytedata, joka voi olla mitä hyvänsä, koska piiri ei käytä tätä dataa mihinkään. Tämän jälkeen piirin valintasiignaali asetetaan lepotilaan, jolloin piirin sisäisen rekisterin data siirtyy siirtorekisteriin. Tämän jälkeen piirin valintasiignaali aktivoidaan ja piirille kelloitetaan mitä hyvänsä dataa 16 bitin verran. Tämän datan kelloituksen aikana rekisterin sisältämä kahdeksanbittinen tieto voidaan lukea sarjamuotoisena piiriltä kahdeksan ensimmäisen kellopulsin jälkeen. (Maxim Integrated 2006.)

SPI-liitettäviä piirejä käytettäessä piirit tarvitsisivat SPI-väylän kolme jaettua signaalia MISO, MOSI ja SCK, sekä yhden oman aktivointisignaalin, jolla kerrotaan, minkä väylällä olevan piirin kanssa kommunikoidaan. Tällöin esimerkiksi neljää 16 IO-liityntäistä laajennuspiiriä käyttäen, 64 liittymän muodostaminen vaatisi vain 11 liityntää. Näistä 11:sta IO-liitynnästä kolme olisi SPI-väylän signaalit ja loput kahdeksan piirien aktivointisignaaleja. Mikäli piirit tukevat ns. ”Daisy chain” -kytkentätapaa, jossa piirit kytketään ketjumaisesti toisiinsa, IO-liityntöjä tarvittaisiin kaikkiaan vain 4 kappaletta. Tässä tapauksessa väylälle kirjoitettavan datan pituus riippuu kytkettyjen piirien määrästä.

Vaikka sarjaliitettävien laajennuspiirien kytkemisen helppous ja tarvittavien IO-liityntöjen määrä olisivat suuria etuja, toisivat ne mukanaan myös haittapuolia. Verrattuna mikro-ohjaimen omien GPIO-liityntöjen luku- ja kirjoitusoperaatioihin, ovat sarjaliitettävät piirit todella hitaita käsitellä. Sarjaliitettävien piirien yhteydessä ilmoitettava suurin väylätaajuus tarkoittaa väylän kellosignaalin taajuutta. Kellosignaalin yhdellä kellopulsilla voidaan siirtää vain yksi bitti tietoa, jolloin suurinta väylätaajuutta voidaan karkeasti kutsua yhden bitin siirtonopeudeksi. Tämän vuoksi todellinen tiedonsiirtonopeus piiriltä on aina maksimissaan väylä-

taajuus jaettuna väylällä liikutettavan tiedon bittimäärällä. Esimerkiksi kirjoitettaessa aiemmin esitellyn MAX7301-piirin rekistereihin kahdeksanbittinen data, vaatii operaatio 16 kellopulssia, koska kirjoitettava tieto sisältää myös rekisterin osoitteen. Jos väylän kelloaajuus on esimerkiksi 10 MHz, tarkoittaa se, että todellinen tiedonsiirtonopeus on enimmillään 625 kHz. Lukuoperaatio kyseisellä piirillä on vieläkin hitaampi, koska aiemmin selostetun lukuoperaation mukaisesti piirin lukuoperaatio täytyy hoitaa kahdessa osuudessa. Lukuoperaation tapauksessa kellopulsseja tarvitaan yhteensä 32 kappaletta, jolloin ideaalinen lukunopeus 10 MHz:llä on 312,5 kHz. Todellisuudessa piirin lukunopeus olisi tätäkin arvoa pienempi, koska tässä lukuarvossa ei ole huomioitu lukuoperaation eri osuuksien välissä olevaa piirin valintasiignaalin lepoaikaa. Mikro-ohjaimen omat GPIO-liitynnät kirjoitetaan ja luetaan rinnakkaismuotoisena yhdellä toimintasyklillä, minkä takia se on sarjamuotoista tiedonsiirtoa todella paljon nopeampaa.

### 6.3.2 Valittu toteutustapa, rinnakkaisväylälaajennus

Edellä mainittujen seikkojen perusteella päätettiin testausliitynnät toteuttaa rinnakkaistyyppisellä väyläratkaisulla. Testerin mikro-ohjaimen GPIO-liitynnöistä muodostetaan kummallekin testauskanavalle oma kahdeksan bittiä leveä rinnakkaisväylä, joka laajennetaan 32-bittiseksi. Tällöin testausliitynnät jakautuvat ikään kuin neljäksi tavuksi. Rajoitteena tämä toteutustapa aiheuttaa sen, että ainoastaan yhtä testausliityntälohkoa voidaan lukea kerrallaan. Vaikka tiedon käsittely testausliityntöjen osalta tapahtuukin kahdeksan bitin paloissa, on tämä monin kerroin nopeampaa kuin sarjamuotoisen IO-laajennuksen käyttäminen. Samalla tämä laajennustapa mahdollistaa testausliityntöjen tietoväylän kytkemisen mikro-ohjaimen portteihin siten, että testausliitynnöissä on käytettävissä keskeytystoiminnot. Tosin keskeytyksiä voidaan käyttää vain siinä testausliityntätavussa, joka on kytketty väylälle. Myöhemmin tähän väylään viitataan selkeyden vuoksi nimellä testausväylä.

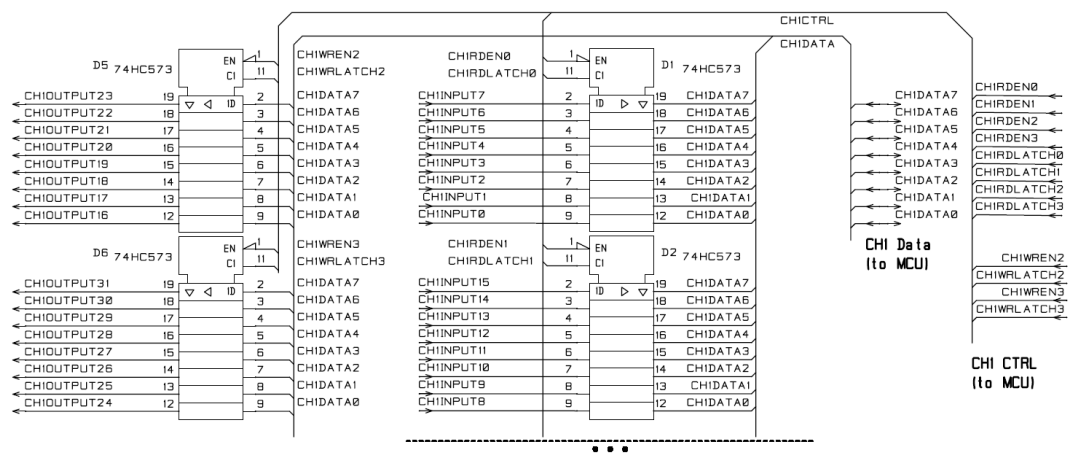
Laajennus päätettiin toteuttaa käyttämällä diskreettejä 74HC573 -logiikkapiirejä, jotka ovat 8-bittisiä läpinäkyviä D-kiikkuja. Niiden lähdöt ovat kolmitilalähtöjä ja kellotulo staattinen. Läpinäkyvyys piirin yhteydessä tarkoittaa sitä, että kiikun kellotulon ollessa aktiivisessa tilassa, tässä tapauksessa ylätilassa, siirtyy piirin

tuloissa oleva tieto sellaisenaan piirin lähtöihin. Mikäli piirin kellotulo laitetaan ei-aktiiviseen tilaan, sillä hetkellä tuloissa oleva data lukittuu piirin lähtöihin jopa silloin, kun niitä ei ole kytketty aktiivisiksi. (NXP 2012a.)

Tällöin tuloissa ollut data voidaan käydä myöhemmin lukemassa piiriltä kytkemällä lähdöt aktiivisiksi. Tämä on erittäin hyödyllinen ominaisuus nopeiden signaalien tutkimisissa, kuten esimerkiksi LCD-näytölle kirjoitettavan datan kaappaamisessa. Tällöin LCD-näytön ohjaussignaaleja tarkkaillaan aktiivisesti tietyn tapahtuman varalta. Mikäli ohjaussignaaleissa havaitaan kirjoitustapahtuma, lukitaan niiden testausliityntöjen tilat, jotka kytkeytyvät LCD-näytön datalinjoihin. Tämän jälkeen datalinjoilla ollut data voidaan käydä rauhassa lukemassa lukitusta piiristä ilman epävarmuutta siitä, että se olisi ehtinyt muuttua ennen sen lukemista.

Kuviossa 10 on esitetty osittainen testausliityntöjen laajennuskytkentä testauskanavalle 1. Kuvioista käy ilmi, miten kahdeksanbittisellä CHIDATA-testausväylällä on toteutettu tulo- ja lähtöominaisuudet testausliityntöihin.

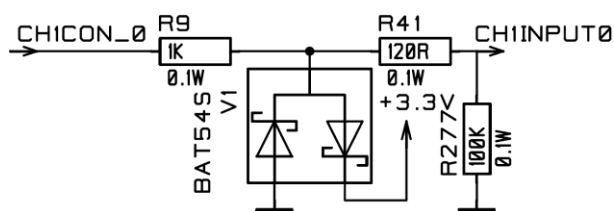
CH1INPUTx-signaalit ovat testausliityntöjen sisääntulosignaaleja, joiden tilat voidaan lukea. Signaalit CH1OUTPUTx ovat manipuloitavien testausliityntöjen ohjaussignaaleja, joihin voidaan kirjoittaa tietoa. Nämä aiemmin mainitut signaalit eivät vielä kytkeydy suoraan testausliityntäliittimeen, vaan niitä edeltävät vielä liityntöjen tulosuojaukset ja lähtöpuskurit, jotka on esitetty myöhemmin kuvioissa 11 ja 12.



KUVIO 10. Osa testauskanava 1:n laajennuskytkennästä

Yhden testauskanavan 32 testausliitynnän tulo-ominaisuus toteutettiin käyttämällä neljää kiikkupiiriä. Näistä piireistä vain kaksi, D1 ja D2, on esitetty kuviossa. Loppujen testausliityntöjen tulokiikkupiirit kytkettiin samaa periaatetta noudattaen. 74HC573-piirien lähdöt ovat kolmitilatyypisiä, jolloin ne voidaan kytkeä suoraan testausväylälle ilman sarjavastuksia. Kuitenkin tällöin testerin laiteohjelmassa tulee varmistaa se, että vain yksi piiri on kytkettynä väylälle kerrallaan. Mikäli useampi tulo-ominaisuuden toteuttava piiri on yhtäaikaisesti kytkeytyneenä testausväylälle ja piirien lähtöjen tilat eroavat, piirit voivat rikkoutua oikosulkuutilanteen takia. D-kiikut ovat yksisuuntaisia komponentteja, minkä takia tieto voi kulkea niissä vain tuloista lähtöihin. Tämän takia kytkentään lisättiin vielä kaksi kiikkupiiriä, D5 ja D6, jotka on kytketty tuloista poiketen toisinpäin. Näin ollen niihin testausliityntöihin, jotka sisältävät lähtöominaisuuden, voidaan kirjoittaa tietoa näiden piirien avulla. Piirien signaalit on järjestetty hieman erikoisesti kytkentäkaaviossa, jotta piirilevyllä signaalit kulkevat mahdollisimman suoraan testausliityntöihin.

Hetkelliseksi ongelmaksi muodostui tässä vaiheessa D-kiikkupiirien yksinkertaisuus. Kiikkupiireissä ei ole mahdollisuutta aktivoida vain osaa lähtöliitynnöistä, vaan kaikki kiikkujen lähtöliitynnät aktivoituvat samaan aikaan. Jos lähtöpiirit olisi kytketty suoraan käytettyihin testausliityntöihin, asettuisivat kaikki yhteen piiriin liittyvät testausliitynnät pakolla siihen tilaan, jossa testausväylän sen hetkinen tieto olisi. Lisäksi toinen osa ongelmaa oli testausliityntöihin tuotavat jännite- tasot. Testausliityntöihin kytkeytyvät signaalit voivat käyttää tulopiirien käyttöjännitteestä poikkeavia jänniteasoja. 74HC573-piirin datalehdessä mainitaan, että piirille kytkettävät signaalit eivät saisi ylittää piirin käyttöjännitettä. Tämän vuoksi testausliityntöihin lisättiin signaalien tulopuskurit, jotka leikkaavat liityntään kytkettävän jännitteen turvalliselle tasolle, ennen tulopiireille kytkemistä. Kytkentä on esitetty kuviossa 11.



KUVIO 11. Yksittäisen testausliittynnän tulopuskuri

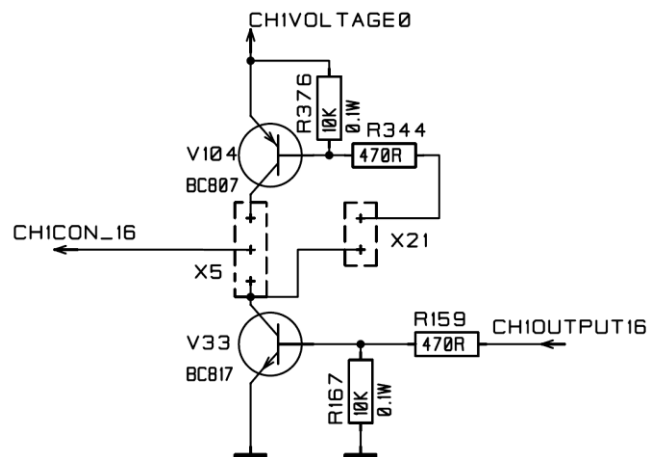
Tulopuskuri kytkeytyy varsinaisen testausliittynnän ja aiemmin esitettyjen tulopiirien väliin. Testausliittyntää edustaa signaali CH1CON ja puskuroitua signaalia edustaa CH1INPUT, jonka kytkeytyminen tulopiireille voidaan nähdä aiemmassa kuviossa 10. Tulopuskurissa käytettiin BAT54S-kaksoisschottkydiodia leikkaamaan testausliittyntään kytkeytyvä jännite, mikäli se ylittää turvalliset jännitetasot. Riippuen testausliittynnässä vallitsevasta jännitteestä, diodi kytkee yli menevän jännitteen testerin 3,3 V:n käyttöjännitteeseen tai maahan. Käytännössä maahan kytkeminen tapahtuu aina, jos testausliittynnän jännitetaso on negatiivinen.

BAT54S:n sisältämän yksittäisen diodin kynnyksijännite on noin 400 mV 10 mA:n virralla (NXP 2012b). Tämä tarkoittaa karkeasti sitä, että noin 3,7 V:n ylittävät jännitteet leikkautuvat tähän jännitearvoon. Kuviossa esitetyn vastuksen R9 tarkoitus on rajoittaa leikkaustilanteessa diodien läpi menevää virtaa. Kyseinen vastus lisää samalla testausliittynnän sisääntuloimpedanssia, jotta tulopiiri ei kuormittaisi testattavia signaaleja. Vastuksen R41 tarkoituksena on estää kiikkupiirin sisäisten ESD-suojadiodien kuormittuminen. Vastus R277 toimii tulopiirin sisääntulon alavetovastuksena, jotta tulopiirin sisääntuloliittyntä on vakaassa tilassa testausliittynnän ollessa irti.

Yksi vaatimuksista oli, että testausliittyntöjen lähtöjännitteiden tulisi olla valittavissa paneelien erilaisten logiikkajännitteiden takia. 74HC573-piirien tuloliittyntöjen loogistentilojen jänniterajat muuttuvat piirin käyttöjännitteen mukaan. Tulon ylätilaksi tunnistettavan jännitteen alaraja on noin 75 % piirin käyttöjännitteestä, jolloin esimerkiksi 3,3 V käyttöjännitteellä, ylätilan alaraja on noin 2,5 voltia. Mikäli lähtöpiirejä käytettäisiin 5 V käyttöjännitteellä, olisi ylätilaksi tunnistettavuuden alaraja 3,75 voltia. Testerin mikro-ohjain toimii 3,3 V:n käyttöjännitteellä, jolloin se ei kykenisi enää suoraan ohjaamaan ulostulopiirejä. Tällöin mikro-

ohjaimen lähtöjännite osuisi jo aiemmin mainitulle kielletylle alueelle, koska 74HC573-piirit eivät sisällä schmitt trigger -ominaisuutta. (NXP 2012a.)

Näiden syiden vuoksi lähtöpiirejä päätettiin käyttää 3,3 V:n jännitteellä ja niiden lähdöt puskuroitaisiin erillisellä lähtökytkennällä, joka mahdollistaa erilaisten jännitetasojen käytön signaaleja manipuloitaessa. KytKentä poistaa myös ongelman, joka liittyy kaikkien lähtöliityntöjen yhtäaikaiseen aktivoitumiseen. Lähtöpuskurointikytkentä on esitetty kuviossa 12.



KUVIO 12. Yksittäisen testausliitynnän lähtöpuskuri

Testausliitynnän lähtöpuskuri on eräänlainen toteemipaalulähtö. Toteemipaalulähtöjä käytetään usein mikro-ohjaimien sisällä niiden IO-lähdöissä. KytKentää on kuitenkin yksinkertaistettu, jotta kytKentää voidaan käyttää yhdellä IO-linjalla. Kuvion 12 lähtöpuskurin ohjaus tapahtuu signaalilla CHIOUTPUT16, joka näkyy myös kuviossa 10. Testerin haluttiin mukailevan ihmiskäyttäjää, jolloin se kykenee sähköisiä signaaleja manipuloimalla mukailemaan näppäimien painalluksia. Näppäinpainalluksien mukaileminen tapahtuu kytKemällä käyttöliitymän näppäinsignaaleja niiden aktiiviseen tilaan. Tämän takia testerin tarvitsee vain kyetä pakottamaan yksittäinen signaali joko ylä- tai alatilaan, riippuen manipuloitavan signaalin aktiivisesta tilasta. Kuviossa komponentit X5 ja X21 ovat jumbpereita, joiden siltauksilla valitaan manipuloitavan signaalin aktiivinen tila.

Mikäli manipuloitava signaali on alatila-aktiivinen, eli esimerkiksi näppäintä painettaessa se kytkeytyy maahan, sillataan jumbperi X5 kuvion 12 mukaisesti ”ala-

asentoon” ja X21 jätetään avonaiseksi. Tällöin manipuloitava signaali kytkeytyy suoraan transistorin V33-kollektorille, joka on NPN-transistori. Testeri voi tällöin manipuloida signaalin alatilaan kytkemällä transistorin kannalle jännitteen, jolloin se alkaa johtaa ja maadoittaa signaalin. Manipuloitava signaali voidaan vapauttaa takaisin omaan tilaansa katkaisemalla transistorin V33-kannan jännite.

Mikäli manipuloitava signaali on ylätila-aktiivinen, sillataan jumpperi X5 ”ylä-asentoon”, ja oikosuljetaan jumpperi X21. Nyt manipuloitava signaali kytkeytyy transistorin V104-kollektorille, joka on PNP-transistori. Tässä tapauksessa tuomalla jännite transistorin V33-kannalle se alkaa nyt johtaa maadoittaen transistorin V104-kannan. Tällöin transistori V104 alkaa johtaa, jolloin testausliityntäryhmälle valittu manipulointijännite, CH1VOLTAGE0, kytkeytyy manipuloitavaan signaaliin pakottaen sen ylätilaan.

Manipulointiominaisuutta käytettäessä on huomioitava kohdesignaalin impedanssi. Testausliityntöjen lähtöpuskurit kytkeytyvät suoraan testausliittimiin, eivätkä ne sisällä virranrajoitusominaisuutta. Tämän takia manipuloitavan signaalin tulee sisältää sen verran impedanssia, jotta manipulointivirta ei ylitä transistoreille asetettua 500 mA:n maksimivirtaa.

Kirjoitus tapahtuu asettamalla mikro-ohjaimen testausväylän liitynnät lähdeksi ja kirjoittamalla haluttu tieto testausväylälle. Jos tieto halutaan siirtää suoraan testausliityntöihin, aktivoidaan haluttu lähtöpiiri asettamalla CH1WREN-signaali alatilaan, jolloin tieto siirtyy välittömästi testausliityntöihin. Kirjoitettava tieto on myös mahdollista lukita lähtöpiiriin asettamalla piirin CH1WRLATCH- signaali alatilaan. Tällöin piirille siirretty data lukittuu, eikä tietoa tarvitse enää säilyttää testausväylällä. Tämä ominaisuus mahdollistaa esimerkiksi näppäinmatriisin helpon manipuloinnin, joka on selostettu myöhemmin testerin laiteohjelman kehityksen yhteydessä.

#### 6.4 Teholähdeosio

Testausliityntöjen suunnittelun jälkeen työssä edettiin testerin teholähdeosion suunnitteluun. Suunnittelu aloitettiin määrittämällä sähköiset arvot, joiden pohjalta alettiin miettiä teholähteen toteutusta.

#### 6.4.1 Sähköiset arvot

Teholähdeosiota suunniteltaessa lähdettiin liikkeelle kartoittamalla testerin tarvitseman pienimmän jännitteen ja sisääntulojännitteen erotusta. USB-liityntä tulisi saamaan sähkönsä PC:ltä, jotta se on galvaanisesti erotettu testeristä. Tämän takia USB-liitynnän ja sen oheiskomponenttien sähköisiä vaatimuksia ei tarvinnut huomioida teholähteen suunnittelussa.

Testerit on osa testiautomaatiojärjestelmää, mutta tarvittaessa sitä tulisi olla mahdollisuus käyttää itsenäisenä laitteena testauksessa. Tämän takia laitetta tulisi syöttämään erillisellä yleismallisella 12 V:n tasajännitelähteellä, jonka virranantokyky on 1,5 A. Tällaisia tasajännitelähteitä käytetään erittäin paljon kuluttajille tarkoitettussa elektroniikassa, kuten esimerkiksi ulkoisissa kiintolevyissä ja ADSL-modeemeissa, joissa niitä kutsutaan yleisnimityksellä verkkolaite.

Testerissä oleva pienin jännite ei määräydy pelkästään testerin sisäisten ominaisuuksien perusteella, vaan myös testattavien paneelien perusteella. Esimerkiksi näppäinpainalluksia simuloitaessa, testattavasta käyttöliittymäpaneelistä riippuen, testerit joutuu kytkemään paneelin näppäinsignaaleja maahan tai tiettyyn jännitetasoon. Tämän vuoksi on erittäin tärkeää, että testerin käyttämä jännite simuloinnissa vastaa paneelin käyttämiä jännitetasoja, jotta paneelin komponentit eivät vaurioidu esimerkiksi ylijännitteestä.

Myös liian alhainen simulointijännite voi aiheuttaa ongelmia. Esimerkiksi simulointijännitteen jäädessä näppäinsignaalien loogistenjännitetasojen väliin ns. kielletylle alueelle, voi testattava paneeli tulkita näppäin painallukset väärin, mikäli näppäintuloissa ei ole schmitt trigger -ominaisuutta. Schmitt trigger -ominaisuus tarkoittaa, että tulossa on hystereesiä, joka eliminoi ns. kielletyn alueen jännitetasojen välistä (Cockrill 2011).

Kuten testerin mikro-ohjain, myös monet muut mikro-ohjaimet ja digitaaliset piirit käyttävät 3,3 V:n jännitettä, minkä takia tämä jännite on ehdottomasti oltava käytettävissä manipulointia varten. Myös 5 V:n jännitetasot ovat yleisiä digitaali-laitteissa, joten myös tämä jännite tulisi olla käytettävissä. Mikäli testerissä olisi tarvetta jollekin erikoisemmalle manipulointijännitteelle, se voitaisiin tarvittaessa



”ryöstää” sitä käyttävästä käyttöliittymäpaneelistä ja kytkeä tarvittaviin testausliittymälähtöihin.

Kuten vaatimusmäärittelyssä todettiin, testerin ei tarvitse syöttää käyttöjännitteitä testattaville paneeleille. Testattavien paneelien käyttöjännitteet tulevat testauksen aikana hitsauslaitteistosta, johon paneeli on kytketty. Näin ollen teholähteen virranantokyvyssä ei tarvitse huomioida paneelien tarvitsemaa virtaa. Testerin tehonlähteen virranantokyvyksi määritettiin 1 A, joka on hieman pienempi kuin testerinä syöttävän tasajännitelähteen virranantokyky. Kyseinen virta-arvo valittiin sillä perusteella, että tällöin testerin tehonlähdettä voitaisiin käyttää tulevaisuudessa syöttämään myös mahdollisia laajennustoimintoja. Vaikka tehonlähde mitoitettiin 1 A:n virralle, rajattiin todellinen käyttövirta noin 0,5 A:iin sulakkeella. Tässä vaiheessa testerin käyttämä maksimivirta määräytyy karkeasti testausliittymästä lähtevien virtojen perusteella. Esimerkiksi näppäinpainalluksia simuloitaessa, yksittäisestä testausliittymästä lähtevän virran suuruus on maksimissaan noin pari milliampeeria. Tällöin teoreettisesti, jos kaikista testerin 32 lähtötyyppisestä testausliittymästä lähtisi noin 5 mA:n virta, olisi kokonaisvirta vain noin 160 mA. Rajoittamalla laitteen kokonaisvirran 0,5 A:iin, voidaan kaikista lähtötyyppisistä testausliittymästä ottaa silti noin 10 mA:n virta yhtäaikaaisesti, jolloin kokonaisvirta olisi noin 320 mA. Todellisuudessa tällaista tilannetta ei koskaan testauksessa tule ilmenemään, mutta tämä mahdollistaa suuremman hetkellisen virranantokyvyn yksittäisille testausliittymäille. Testerin muut virrat, kuten mikro-ohjaimen käyttämä virta, ovat niin pieniä, ettei niitä tarvinnut erityisemmin huomioida tehonlähteen mitoituksessa tai sulakkeen valinnassa.

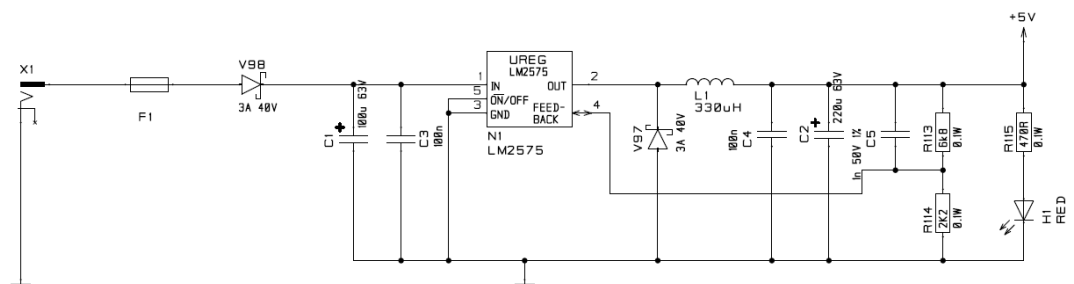
#### 6.4.2 Tehonlähteen rakenne

Tehonlähde päätettiin toteuttaa siten, että se toteuttaisi portaittaisen jännitteiden tiputuksen. Testerin sisääntulojännite tiputettaisiin ensin 5 V:iin hakkurikytkennällä, josta se tiputettaisiin edelleen 3,3 V:n jännitteeksi lineaariregulaattorilla. Tämä mahdollistaa tehonlähteessä tapahtuvien tehohäviöiden jakamisen eri komponenteille, jolloin yksittäiset komponentit eivät rasitu liikaa. Jos 3,3 V:n jännite muodostettaisiin suoraan 12 V:n jännitteestä lineaariregulaattorilla, syntyisi regulaattorissa jo 100 mA:n virralla noin 0,9 W tehohäviötä. Hakkurikytkennällä ky-

seinen jännite voitaisiin tuottaa suoraan sisääntulojännitteestä ja mikäli kytkennässä käytettäisiin sopivaa hakkuriregulaattoria, se osaisi sopeutua myös korkeampiin sisääntulojännitteisiin.

Syy miksi lineaariregulaattoria kuitenkin haluttiin käyttää tuottamaan 3,3 V:n jännite oli, että testeri ei käytä pitkäaikaisesti suuria virtamääriä kyseisellä jännitteellä. Tällöin tehollähde voitaisiin pitää mahdollisimman yksinkertaisena, koska hakkurikytkentä vaatii aina lineaariregulaattoria enemmän komponentteja. Lisäksi testeriin jouduttaisiin joka tapauksessa luomaan 5 V:n jännite signaalien manipuloimista varten, jolloin oli luontevaa muodostaa 3,3 V:n jännite tästä jännitteestä. Tällöin myös tehohäviöt jakautuvat tasaisemmin, mikä lisää mahdollisuuden kuormittaa 3,3 V:n jännitettä enemmän. Lineaariregulaattorilla on myös tehokas ”ripple rejection” -ominaisuus, jolla se vaimentaa sisääntulon rippelijännitettä. Tällöin 3,3 V:n vakavoimiseen käytetty lineaariregulaattori vaimentaa 5 V:n hakkuritehollähteen tuottamaa rippelijännitettä, muodostaen puhtaan käyttöjännitteen mikro-ohjaimelle.

Tehollähteen toteutus aloitettiin 5 V:n hakkurikytkennästä, joka on esitetty kuviossa 13. Hakkuri on step-down -tyyppinen, eli jännitettä pudottava. Kytkentään valittiin Texas Instrumentsin LM2575-ADJ -säädettyä hakkuriregulaattori. Kyseinen komponentti vaatii melko vähän ulkoisia komponentteja toimiakseen ja sille on kuvattu tarkka mitoitusproseduuri, jolla oheiskomponenttien arvot on helppo määrittää. Regulaattorilla toteutetun hakkurin lähtöjännite määrätään jännitejaolla muodostetulla takaisinkytkennällä. Regulaattorissa on lisäksi virranrajoitustoiminto sekä automaattinen sammutustoiminto, jotka suojaavat komponenttia mahdollisessa vikatilanteessa. (TI 2013a.)



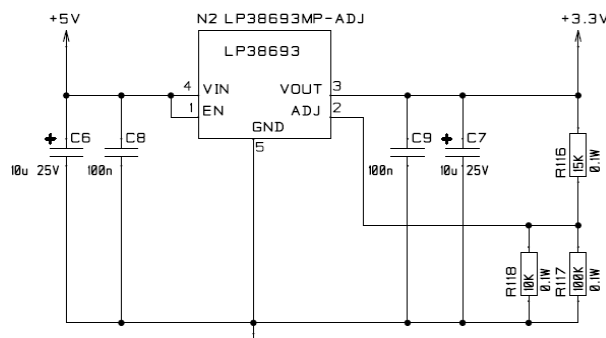
KUVIO 13. Hakkurikytkentä LM2575-ADJ -regulaattorilla

Kuviossa esitetyn hakkurikytkennän lähtöjännite määräytyy vastusten R113 ja R114 muodostamalla jännitteenjakoketjulla. Lähtöjännite määräytyy kaavan 1 mukaisesti, jossa  $V_{REF}$  vastaa piirin sisäistä 1,23 V:n referenssijännitettä. Kaavaan on merkitty myös testerissä käytettyjen vastusten tunnuksat selkeyden vuoksi.

$$V_{OUT} = V_{REF} \left( 1 + \frac{R2}{R1} \right) = 1,23 \text{ V} \left( 1 + \frac{R113}{R114} \right)$$

KAAVA 1. LM2575-hakkuriregulaattorin lähtöjännite (TI 2013a).

LM2575 regulaattorista on olemassa versioita, joissa komponentin lähtöjännite on asetettu kiinteäksi, eikä vastusjakoketjua tällöin tarvita. Suunnitteluhetkellä tällaisia komponentteja ei ollut käytettävissä, joten kytkennässä käytettiin säädettävää mallia. Kuviossa näkyy myös testeriä suojaava sulake F1. Teholähteeseen lisättiin myös verkkolaitteen väärältä napaisuudelta suojaava diodi, joka näkyy kuviossa tunnuksella V98.



KUVIO 14. 3,3 V:n muodostava lineaariregulaattorikytkentä

Hakkuriosion jälkeen toteutettiin 3,3 V:n tuottava regulaattorikytkentä, joka on esitetty kuviossa 14. Tähän kytkentään valittiin lineaariregulaattoriksi Texas Instrumentsin valmistama LP38693-ADJ, joka on myös säädettävä. Kuten kuviosta käy ilmi, lineaariregulaattori vaatii vähemmän oheiskomponentteja kuin hakkuriregulaattori. Kyseisen regulaattorin lähtöjännite määrätään hakkuriregulaattorin tavoin myös jännitteen jakoketjulla, jonka muodostavat vastukset R116, R117 ja R118. LP38693:n tapauksessa kaavassa käytettävä säätöjännite on 1,25 V. Lisäksi regulaattorien datalehdissä esitettyjen vastusten tunnuksat ovat vaihdettu keske-

nään, mutta tämä on huomioitu kaavassakin. LP38693:lle sopiva lähtöjännitekaava on esitetty kaavana 2, johon on myös merkitty testerissä käytettyjen vastusten tunnuksat.

$$V_{OUT} = V_{ADJ} \left( 1 + \frac{R1}{R2} \right) = 1,25 V \left( 1 + \frac{R116}{(R117 || R118)} \right)$$

KAAVA 2. LP38693-ADJ lähtöjännitteen kaava (TI 2013b).

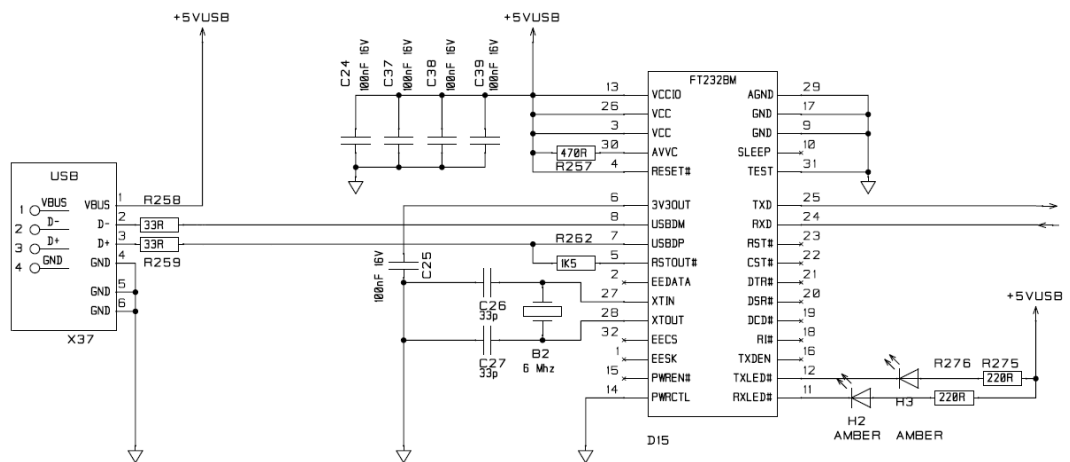
LP38693-ADJ -regulaattorin datalehdessä mainitaan jakoketjun maahan päin kytkeytyvästä vastuksesta, että sen tulisi olla 12 kΩ tai pienempi. Tämän vaatimuksen takia sopivia yksittäisiä vastusarvoja ei löytynyt pintaliitosmallisina, joilla vastusten suhde olisi ollut haluttu. Jotta lähtöjännite saatiin määritettyä 3,3 V:iin, jännitteenjakoketjussa täytyi muodostaa rinnankytkentä vastuksilla R117 ja R118, ja tämä muodosti noin 9 kΩ resistanssin. Tällöin näiden vastusten rinnankytkentä yhdessä 15 kΩ:n vastuksen R116 kanssa asettivat lähtöjännitteen noin 3,3 V:iin. (TI 2013b.)

## 6.5 USB-liityntä

Työn vaatimusmäärittelyn mukaisesti testeri kommunikoi tietokoneen kanssa USB:llä, joka näkyy tietokoneen käyttöjärjestelmälle virtuaalisena sarjaporttina. Toteutus tehtiin käyttämällä FTDI:n valmistamaa FT232BL USB-UART -muunninpiiriä, vaikka testeriin valittu NXP:n LPC2387 mikro-ohjain sisältää myös itsessään USB-liitynnän. Syy miksi erillistä USB-UART-muunninpiiriä päädyttiin käyttämään, johtui täysin vaatimuksesta, että PC:n ja testerin tulee olla galvanisesti erotettu. USB-liityntä on differentiaalinen liityntä, jossa tieto kulkee molempiin suuntiin samoissa johtimissa. Tämän takia sitä ei voida erottaa yksinkertaisilla peruskomponenteilla, vaan tätä varten tarvitaan erityisiä USB-erotuspiirejä. Suunnitteluhetkellä tämän kaltaisia piirejä ei ollut tuotannossa käytettävissä. Tämän takia päädyttiin ratkaisuun, jossa USB-liityntä muunnetaan ensin asynkroniseksi sarjaportiksi, joka muunnoksen jälkeen erotetaan muusta testeristä. Sarjaportissa tietoa siirretään ainoastaan yhteen suuntaan yhdessä johtimessa, mikä mahdollistaa helposti toteutettavan erotuskytkennän.

Testerissä oleva muunninpiiri saa sähkönsä PC:ltä USB:n kautta, jotta se ei olisi galvaanisessa yhteydessä testeriin. Tämä oli yksinkertainen ja varmin ratkaisu. Lisäksi USB-liitynnän komponentit eivät tässä tapauksessa tarvitse kovin paljoa virtaa, minkä takia niiden sähkönsyöttö voidaan ottaa USB:stä. Jos USB-liitynnän sähkönsyöttö olisi otettu testerin teholähteestä, olisi teholähteen joutunut muuntamaan flyback-tyyppiseksi, jotta galvaaninen erotus USB-osion ja testerin välillä säilyisi. Tämä olisi tarkoittanut nykyisen hakkuriregulaattoriin vaihtamista toiseen ja flyback-muuntajan käyttämistä.

Itse USB-UART-muunninpiiri vaatii melko vähän oheiskomponentteja toimiakseen. Piiri on suunniteltu mahdollisimman helppokäyttöiseksi ja sitä varten on kattavat esimerkkikytkennät. Kytkennän toteutuksessa hyödynnettiin FTDI:n omaa esimerkkikytkentää, joka on esitetty piirin datalehdessä. Testeriin toteutettu USB-kytkentä on esitetty kuviossa 15.



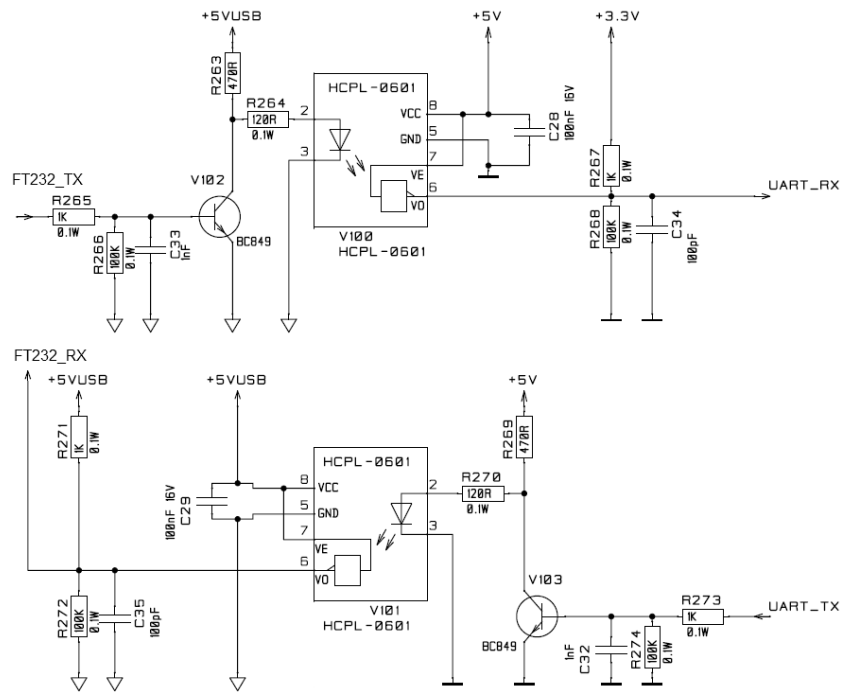
KUVIO 15. USB-UART muunnos FT232BL-muunninpiirillä

Kytkennässä itse USB-liityntään liittyvät tärkeimmät oheiskomponentit ovat muunninpiiriin kellokide B2, ja vastukset R258, R259 ja R262. Näistä komponenteista sarjavastuksien R258 ja R259 tehtävänä on sovittaa USB-kaapelin impedanssi, jotta datasiinaaleissa ei esiinny heijastumia. Kellokiteen B2 avulla muunninpiirille luodaan sen tarvitsema kellosignaali. Vastuksen R262 tehtävä liittyy laitteen USB-nopeusluokan tunnistukseen.

USB-laitteet voivat toimia kolmessa eri nopeusluokassa low-speed, full-speed ja high-speed. Nopeusluokan tunnistaminen perustuu siihen, että kytkettäessä lisälaitetta isäntälaitteeseen, esimerkiksi PC:hen, isäntälaitte tutkii, onko toinen USB:n datalinjoista ylätilassa. Mikäli USB:n D- -linja on ylätilassa, laite on low-speed -tyyppinen. Mikäli D+ -linja on ylätilassa kytkettäessä, laite on full-speed -tyyppinen. High-speed -luokan laitteet toimivat aluksi samalla tavalla kuin full-speed luokan laitteet, mutta siirtyvät isäntälaitteen tekemän USB-väylä resetin yhteydessä high-speed -tilaan. Low-speed -luokan laitteiden tiedonsiirtonopeus on 1,5 Mb/s, full-speed laitteiden 12 Mb/s ja high-speed laitteiden 480 Mb/s. (USB-IF 2000.)

Vastuksen R262 tarkoitus on kertoa isäntälaitteelle, tässä tapauksessa PC:lle, muunninpiirin olevan full-speed -tyyppinen USB-laite. Virallisen USB 2.0 spesifikaation mukaisesti ylösveto tulee poistaa käytöstä silloin, kun isäntälaitteen USB-liityntä ei ole aktiivinen. Tämä toiminnallisuus toteutuu FT232BL-piirin RSTOUT-liitynnällä, joka asettuu alatilaa silloin, kun USB-liityntä on resettilassa.

USB-sarjaportti -muunnoksen jälkeen seuraava vaihe oli toteuttaa testerin mikroohjaimen ja USB-osion galvaaninen erotus. Testerissä käytetty sarjaliikenne on asynkronista, ja se kulkee kahdessa eri johtimessa. Yksittäisessä johtimessa liikennöinti tapahtuu vain toiseen suuntaan. Tämän johdosta erotus oli helppo toteuttaa optoerottimilla. Erotuskytkentä on esitetty kuviossa 16.



KUVIO 16. Sarjaliikenneliitynnän galvaaninen erotus

Tuotannosta löytyi sopivat pintaliitosmalliset optoerotimet. Erottimien käyttöjännite on 5 V ja niiden lähdöt ovat avokollektorityyppisiä. Erottimen ledi vaatii vähintään noin 7 mA virtaa toimiakseen luotettavasti. Lisäksi avokollektorilähdön takia optoerotin toimii invertoidusti. Tämän takia erottimille tuotu sarjaliikennesignaali on puskuroitu NPN-transistoreilla.

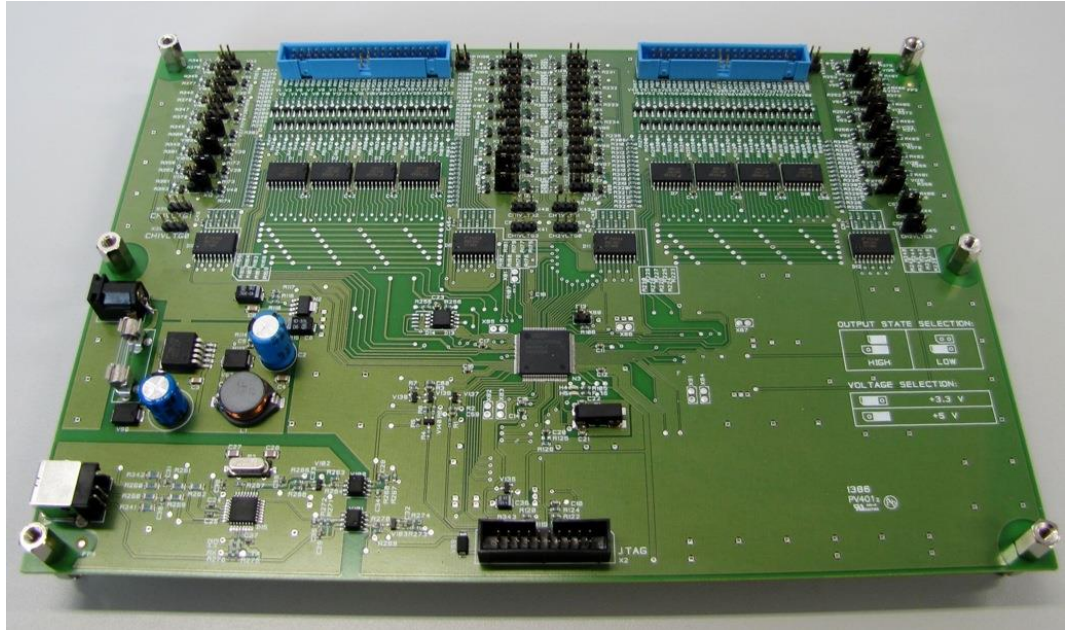
## 7 TESTERIN TOTEUTTAMINEN

Elektroniikan suunnittelun ja piirikaavion piirtämisen jälkeen aloitettiin testerin piirilevyn suunnittelu. Tämä oli työskentelyssä kaikista aikaa vievin työvaihe. Piirilevy päätettiin suunnitella kaksipuoliseksi, ja siihen sovellettiin Kempillä käytössä olevia suunnittelusääntöjä.

Suurin osa levyllä käytettävistä komponenteista on pintaliitoskomponentteja. Palakomponenttien pääasiallinen kotelokoko on ns. 0603, mutta esimerkiksi joitain vastuksia ei löytynyt tästä koteloluokasta, jolloin ne korvattiin 1206 kotelokoon komponenteilla. Liittimet, jumpperien piikkirimat ja jotkin elektrolyyttikondensaattorit ovat läpijuotettavia, koska vastaavia pintaliitososia ei ollut saatavilla. Liittimien tapauksissa läpijuotettavat komponentit ovat kuitenkin usein mekaanisesti kestävämpiä kuin pintaliitosmallit, mikä lisää laitteen elinikää.

Testeri tuotetaan Kempin omalla elektroniikkatehtaalla, minkä takia piirilevysuunnittelussa tuli huomioida tiettyjä tuotannollisia seikkoja. Piirilevyllä kaikki komponentit ovat sijoitettu top-kerrokselle. Tämän johdosta piirilevyä ei tarvitse ajaa kahta kertaa komponenttiladontalinjan läpi, mikä nopeuttaa laitteen kokoonpanoa. Lisäksi piirilevyn ääri rajoissa on otettu huomioon tuotantolinjan vaatimukset. Tuotantolinjalla käytetään muutamaa eri leveyksistä kuljetinta. Tämän takia piirilevyn toisen sivun ulkomitan tuli vastata kuljettimen leveyttä. Piirilevyille lisättiin myös testipisteet kaikkiin signaalivetoihin, jotta kalustetulle piirilevyille voidaan suorittaa IC-testaus. IC-testauksella havaitaan, jos levyllä on muodostunut oikosulkuja tuotannon juotosvaiheen jälkeen, levyllä puuttuu komponentteja tai ne on asennettu väärin. Piirilevyn molemmille puolille tehtiin maadoitettu kuparitäyttö häiriöiden minimoimiseksi. Levyn eripuolien maatäytöt yhdistettiin toisiinsa useista eri kohdista läpivientien avulla, jotta niiden välille ei syntyisi potentiaalieroja.





KUVIO 17. Valmis testerilaite

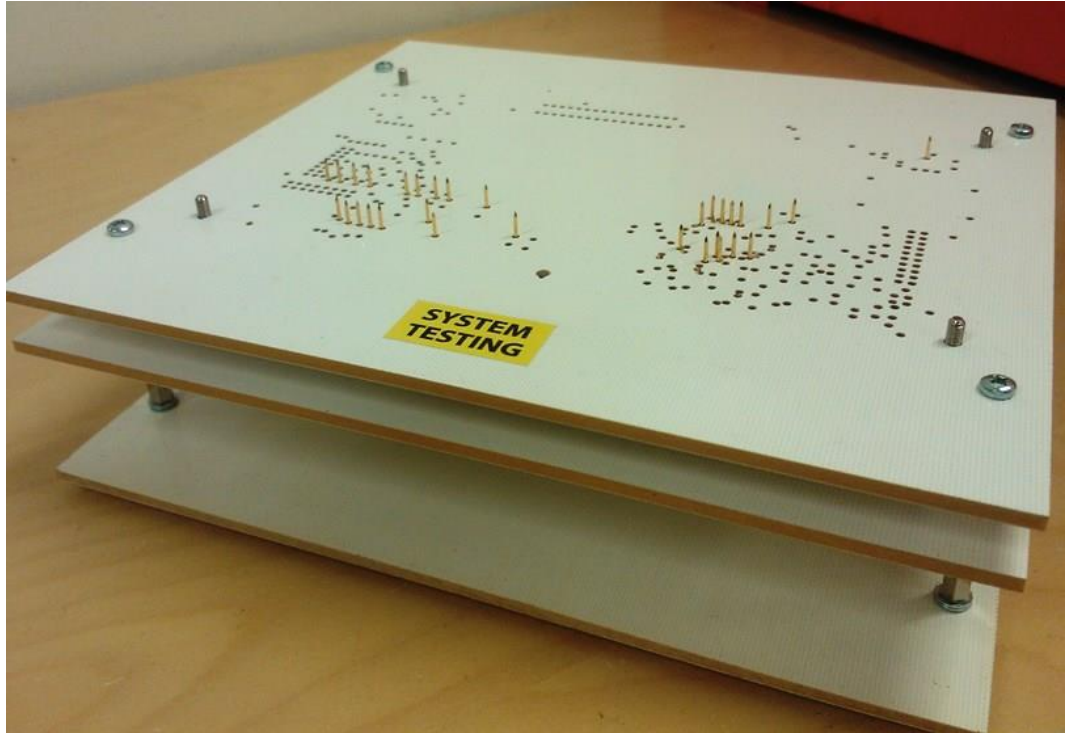
Valmis kalustettu tester on esitetty kuviossa 17. Kuviossa näkyy hyvin testerin hakkuriosio, USB-osio sekä testauskanavat ja niiden komponentit. Kuvion vasemmassa alakulmassa näkyy testerin USB-osio, joka on erotettu muusta testeristä. USB-osion maatäyttö on myös täysin erotettu muusta testeristä. Teholähteen hakkuriosio on nähtävissä USB-osion yläpuolella. Osion erottaa kahdesta pyöreästä elektrolyyttikondensaattorista. Hakkuri on eristetty omaan maasaarekkeeseen, joka on maadoitettu vain yhdestä pisteestä muuhun testeriin. Yksipistemaadoituksen tarkoituksena on estää ei-haluttujen virtasilmukoiden muodostuminen. Näin hakkurin tulo- ja paluuvirrat kulkevat vain sen omassa maasaarekkeessa. Tämä estää hakkurin tuottamien häiriöiden leviämisen maatäyttöä pitkin muualle testeeriin. Testausliityntöjen nauhakaapeliliittimien ympärillä näkyvät piikkirimat jumppereille lähtöpuskureiden toimintatilojen asettamista varten. Musta liitin kuvion alakulmassa on mikro-ohjaimen JTAG-liityntä, jonka kautta mikro-ohjain voidaan ohjelmoida.

## 8 TESTERIN LIITTÄMINEN PANEELIIN

Paneelien liittäminen testeriin haluttiin olevan mahdollisimman helppoa ja nopeaa. Tulevaisuudessa sillä voitaisiin testata myös muiden hitsauslaiteperheiden paneeleja, minkä takia etenkin paneelien vaihtaminen tulisi onnistua nopeasti. Johtojen juottaminen paneeliin on aikaa vievää ja mekaanisesti huono ratkaisu. Tämä havaittiin jo laitteen kehitysvaiheessa, kun testerin ominaisuuksia testattiin käyttöliittymäpaneelilla, joihin oli juotettu kytkentäjohdot. Lisäksi johtojen kytkemisessä oli suurempi todennäköisyys virheelliseen kytkentään. Tämän takia testausta varten paneeleille tuli kehittää liityntätapa, jolla paneeli voidaan helposti ja luotettavasti liittää testeriin.

Testerin liittäminen testattavaan paneeliin tapahtuu käyttämällä itse tehtyä neulapetiä, jollainen valmistetaan jokaiselle käyttöliittymäpaneelimalle. Testeriä varten tehty neulapeti sisältää käyttöliittymän piirilevyn layoutin mukaan sijoitettuja neuloja, jotka kytkeytyvät piirilevyllä kulkeviin signaaleihin. Neulapeti kytkeytyy testattavan kohteen piirilevyllä olevien läpivientien, testipisteiden ja komponentin padien kautta. Teollisuudessa, kuten Kemppi Oy:llä, neulapetejä käytetään elektroniikan IC-testauksessa (In-Circuit testing). IC-testauksessa elektroniikkakortin sisältämät komponentit voidaan testata niiden ollessa jo kalustettuina. Kortille voidaan tehdä myös toiminnallisia testejä, jolloin testilaitteisto kytkee sähkötkortille ja mittaa sen toimintoja. Neulapetiä kutsutaan usein teollisuudessa englanninkielisellä termillä fixture, jolla tarkoitetaan testausalustaa.

Neulat ovat jousikuormitettuja, jolloin ne säätävät aina oikeaan korkeuteen. Neulat asennetaan niille tarkoitettuihin neulaholkkeihin, jolloin ne voidaan helposti vaihtaa tarvittaessa. Neuloja on saatavilla myös erimallisilla päillä ja erilaisilla jousivoimilla olevia. Testerin neulapeteissä neulat on sijoitettu aina tietyn paneelimalle mukaan, minkä takia jokaiselle paneelimalle tehdään oma neulapeti. Valmis X37-asetuskäyttöliittymäpaneelia varten toteutettu neulapeti on esitetty kuviossa 18.



KUVIO 18. Neulapeti, johon testattava X37-asetuskäyttöliittymä kiinnitetään

Testerää varten tehty neulapeti koostuu kolmesta eristävästä levystä, jotka ovat kiinnitetty toisiinsa. Alimmaisina levy toimii vain suojana neuloja pitäville holkeille ja neulapedin johtimille. Kahteen päällimmäisimpään levyyn on porattu reiät paneelin läpivientien, testipisteiden ja läpijuotettavien komponenttien jalkojen kohdalle. Keskimmäinen levy toimii kiinnitysalustana neuloja pitäville holkeille. Jotta neulat ovat varmasti tarkasti oikeilla kohdilla ja päällekkäiset levyt kohdistuvat oikein, on reikien poraus tehty tietokoneohjatulla jyrsimellä. Testattava paneeli kiinnitetään neulapetiin kiristysruuveilla alhaaltapäin. Kiristysruuvit näkyvät kuviossa esitetyn neulapedin kulmissa.

## 9 TESTERIN LAITEOHJELMA

Tässä osuudessa esitellään testeriin toteutettuja toimintoja yleisellä tasolla. Toimintojen vuokaaviot löytyvät liitteestä 2. Testerin toiminnallisuus perustuu sen mikro-ohjaimen suorittamiin toimintoihin. Testerin laiteohjelma kehitettiin alusta asti itse, koska samalla haluttiin oman ammattitaidon kehittämisen merkeissä tustua paremmin laitteessa käytettyyn LPC2387-mikro-ohjaimen. Laiteohjelma toteutettiin tilakonetyyppisenä ratkaisuna. Tämä mahdollistaa uusien ominaisuuksien lisäämisen helposti ohjelmaan. Ohjelmasta on esitetty vain testerin ominaisuuksiin liittyvät funktiot.

Mikro-ohjaimen laiteohjelmaan on toteutettu toimintafunktiot, jotka mahdollistavat datan kaappaamisen ja signaalien manipuloinnin. Suurin osa funktioista on pystytty toteuttamaan siten, että ne soveltuvat erilaisille käyttöliittymäpaneelleille. Osa kuitenkin on jouduttu toteuttamaan siten, että ne ovat tarkoitettu vain tietyille käyttöliittymätyypille. Esimerkiksi LCD-näytön tietojen lukemiseen tarkoitettu funktio jouduttiin toteuttamaan tällä tavalla. Syynä tähän oli se, että funktiosta tehtiin liian yleiskäyttöinen, ja se hidastui, mikä aiheutti virheitä tietojen lukemiseen.

Testeri suorittaa vain kaikkein välttämättömimmät toimenpiteet datan kaappauksen osalta. Se ei ala tulkitsemaan lukemaansa dataa, vaan välittää sen sellaisenaan PC:lle. PC:llä oleva testerin komento-ohjelma hoitaa datan käsittelyn ja muuntamisen tarvittavaan muotoon. Datan käsittely on huomattavasti nopeampaa ja helpompaa suorittaa PC:llä, jossa on huomattavasti enemmän muistia ja tehoa kuin pienessä mikro-ohjaimessa. Lisäksi komento-ohjelman muokkaaminen tarvittaessa on nopeampaa kuin mikro-ohjaimen laiteohjelman muokkaaminen.

## 9.1 LCD-näytön lukeminen

```

1  CH1R_ENABLE = X37_LCD_CTRLBYTE;
2  while(1) // Wait until LCD cursors returns home
3  {
4      if( !(CH1DATA & lcdconn.rs_mask) && !(CH1DATA & lcdconn.rw_mask) )
5      {
6          while( !(CH1DATA & lcdconn.e_mask) ); // Wait for rising edge or high on E
7          while( CH1DATA & lcdconn.e_mask ); // Wait for falling edge on E
8
9          CH1R_LOCK = X37_LCD_DATALATCH; // Lock data
10         CH1R_DISABLE= X37_LCD_CTRLBYTE; // Disable LCD control signal reading
11         CH1R_ENABLE = X37_LCD_DATAABYTE; // Enable LCD data signal reading
12         lcdcmd = CH1DATA; // Read written command
13
14         CH1R_DISABLE= X37_LCD_DATAABYTE; // Disable LCD data signal reading
15         CH1R_ENABLE = X37_LCD_CTRLBYTE; // Enable LCD control signal reading
16         CH1R_UNLOCK = X37_LCD_DATALATCH; // Release data lock
17
18         if(lcdcmd == LCD_CURSOR_HOME_CMD) // Was the given command "Cursor home"?
19         {
20             break; // Yes, start capturing data
21         }
22     }
23 }
24 while( buffctr < 80 ) // Repeat reading (for 20x4 display)
25 {
26     while( !(CH1DATA & lcdconn.rs_mask) ); // Wait for RS to go high
27
28     while( !(CH1DATA & lcdconn.e_mask) ); // Wait for rising edge or high on E
29     while( CH1DATA & lcdconn.e_mask ); // Wait for falling edge on E
30
31     CH1R_LOCK = X37_LCD_DATALATCH; // Lock data
32     CH1R_DISABLE= X37_LCD_CTRLBYTE; // Disable LCD control signal reading
33     CH1R_ENABLE = X37_LCD_DATAABYTE; // Enable LCD data signal reading
34     lcdbuffer[buffctr++] = CH1DATA; // Read data to buffer
35
36     CH1R_DISABLE= X37_LCD_DATAABYTE; // Disable LCD data signal reading
37     CH1R_ENABLE = X37_LCD_CTRLBYTE; // Enable LCD control signal reading
38     CH1R_UNLOCK = X37_LCD_DATALATCH; // Release data lock
39
40     while( CH1DATA & lcdconn.rs_mask ); // Wait for RS to go low
41 }
42 CH1R_DISABLE = X37_LCD_CTRLBYTE; // Disable test-connections
43 CH1R_DISABLE = X37_LCD_DATAABYTE;

```

### KUVIO 19. LCD-näytön lukeminen testauskanavasta 1

Kuviossa 19 on esitetty LCD-näytön lukufunktio testauskanavalle yksi. Kyseinen funktio on pyritty optimoimaan mahdollisimman nopeaksi, minkä takia funktiossa on käytetty paljon ohjelmakääntäjän vakiomäärittelyjä. Esimerkiksi CH1R\_ENABLE on vakiomäärittely. Kyseinen vakio on määritetty tarkoittamaan samaa asiaa, kuin testauskanavan yksi tulopiirien EN-signaaleja ohjaavien GPIO:iden -nollausrekisteri FIO1CLR. Testerin kytkennässä käytetyt 74HC573-kiikkupiirit ovat alatila-aktiivisia. Tällöin, kun jokin tulopiireistä halutaan aktiiviseksi, sen EN-signaali asetetaan alatilaan. Sama pätee myös kiikkupiireissä olevan C1-signaaliin, jota voidaan hyödyntää datan lukitsemiseen. Nollausrekisterin käyttö mahdollistaa sen, että GPIO-liityntä voidaan nollata täsmälleen samalla datalla kuin se voidaan asettaa ylätilaan. Ylätilaan asettaminen voidaan tehdä

käyttämällä joko nollausrekisterin kaltaista asetusrekisteriä FIO1SET, tai kirjoittamalla suoraan GPIO-liityntöjen yleiseen ohjausrekisteriin FIO1PIN.

Funktio alkaa sillä, että ne testausliitynnät, jotka liittyvät LCD-näytön ohjaussignaaleihin, kytketään aktiivisiksi testausväylälle lukua varten. Tämän jälkeen funktio alkaa odottaa LCD-näytölle kirjoitettavaa ”cursor home” -komentoa. Kyseisen komennon käyttö tarkoittaa, että näytölle aletaan kirjoittaa uudelleen alusta. Käyttöliittymäkomponentteihin tutustumisen yhteydessä esiteltiin LCD-näytön komentojen ja datan kirjoittamisen ajastuskaaviot. Niistä käy ilmi, että komennon kirjoituksessa LCD-näytölle, näytön ohjaussignaalit RS ja RW ovat molemmat alatilassa. Funktio tarkkailee tätä toiminnallisuutta if-lauseella rivillä 4. Mikäli komennonkirjoitustapahtuma on havaittu, funktio siirtyy kaappaamaan kirjoitettavan komennon. Se odottaa näytön E-signaalin laskevaa reunaa, jolloin se lukitsee LCD-näytön datalinjoilla olevan komennon. Komento käydään tämän jälkeen lukemassa ja tarkastetaan, onko kyseessä haluttu komento. Mikäli komento oli haluttu, funktion suoritus etenee näytöllä esitettävän datan kaappaukseen. Näytöllä esitettävän datan kaappaus etenee lähes samalla tavalla kuin aiempi komennon kaappaus. Periaatteessa ainoa ero edelliseen on se, että rivillä 26 funktio odottaa ylätilaa RS-signaalissa. RS-signaalin ollessa ylätilassa LCD-näytön toiminnot kohdistuvat sen näyttömuistiin. Tämän jälkeen suoritus etenee samalla tavalla kuin komennonkaappaus, mutta kaapattu data tallennetaan taulukkoon. Lopuksi käytetyt testausliitynnät kytketään pois testausväylältä.

Testauskanavan kaksi funktio on täsmälleen samanlainen toiminnaltaan, mutta ohjaukset tapahtuvat testauskanavan kaksi ohjaussignaaleille. Funktion toiminnallinen vuokaavio on esitetty liitteessä 2.

## 9.2 Seitsensegmenttinäyttöjen ja ledien luku

```

1  RLatch(channel, segmentconn.columnbyte, UNLOCK_DATA);           // Disable column latch if active
2
3  memset( segmentdata, 0, sizeof( segmentdata ) );               // Clear previous segment data
4  RByteEnable( channel, segmentconn.columnbyte, ENABLE_BYTE );   // Enable column reading
5  columnctr = segmentconn.columnpadding;                          // Set start point
6
7  while( columnctr < (segmentconn.n_columns+segmentconn.columnpadding) )
8  {
9      while( ( ReadData(channel) & (1 << columnctr) ) );         // Wait for low state
10     while(! ( ReadData(channel) & (1 << columnctr) ) );         // Wait for columns rising edge
11     BlockingDelay_us( 100 );                                     // Wait for while (100us) to get valid data
12
13     RLatch(channel, segmentconn.segmentbyte, LOCK_DATA);        // Lock segments
14     RByteEnable( channel, segmentconn.columnbyte, DISABLE_BYTE); // Disable columns
15     RByteEnable( channel, segmentconn.segmentbyte, ENABLE_BYTE); // Enable segments reading
16
17     segmentdata[columnctr] = (ReadData(channel) & segmentconn.segmentmask); // Read segments to array
18
19     RByteEnable( channel, segmentconn.segmentbyte, DISABLE_BYTE); // Disable segment reading
20     RByteEnable( channel, segmentconn.columnbyte, ENABLE_BYTE); // Enable column reading
21     RLatch(channel, segmentconn.segmentbyte, UNLOCK_DATA);      // Unlock segments
22
23     while( (ReadData(channel) & (1<<columnctr) ) );             // Wait until high state is over
24     columnctr++;
25 }
26 RByteEnable( channel, segmentconn.columnbyte, DISABLE_BYTE ); // Disable column reading

```

### KUVIO 20. Segmenttinäyttöjen lukufunktio

Kuviossa 20 on esitetty seitsensegmenttinäyttöjen lukemiseen käytetty funktio. Funktiota käytetään myös ledien tilatietojen lukemiseen. Ledien ja seitsensegmenttinäyttöjen päivitystahti on vain muutamia satoja hertsejä, minkä takia funktiosta on voitu tehdä yleiskäyttöisempi kuin esimerkiksi LCD-näytön lukufunktiosta. Tämän takia funktiossa on käytetty apufunktioita, jotka kykenevät suoraan ohjaamaan eri testauskanavia.

Funktion toiminta perustuu siihen, että se aluksi tarkkailee ensimmäisen seitsensegmenttinäytön sarakesignaalia ja odottaa, että siinä esiintyy nouseva reuna. Sarakesignaali vastaa aiemmin kuviossa 9 esitettyä DIGIT-signaalia, jolla yksittäinen näyttö aktivoidaan. Havaitessaan nousevan reunan sarakesignaalisissa, funktio odottaa noin 100 µs, minkä jälkeen se kaappaa näytölle menevien segmenttilinjojen datan. Odottamisella varmistetaan, että segmenttiliityntöjen data on mahdollisimman validia. Data käydään tämän jälkeen lukemassa lukituista liittynöistä ja tallennetaan taulukkoon. Toimintaa toistetaan kaikille näytöille, jotka on määritetty ”segmentconn” -tietorakenteessa. Luettuaan kaikkien näyttöjen tiedot taulukkoon testeri sammuttaa testausliittynät ja välittää luetut tiedot PC:lle.

### 9.3 Enkooderien pyöryksen simulointi

```

1  if( ppotcommand.direction == CW_DIRECTION )           // Clockwise rotation?
2  {
3      encoderphaseA = targetpot.potLineA;
4      encoderphaseB = targetpot.potLineB;
5  }
6  else if( ppotcommand.direction == CCW_DIRECTION )    // Counter-clockwise?
7  {
8      encoderphaseA = targetpot.potLineB;
9      encoderphaseB = targetpot.potLineA;
10 }
11
12 ChannelToOutput( ch );                               // Set testchannel to output
13 for( ctr = 0; ctr < ppotcommand.steps; ctr++ )      // Perform desired number of steps
14 {
15     uart_PutC('.');                                   // Send keep-alive signal to PC
16     WriteData(ch, encoderphaseA);                    // Put first rotation phase to bus
17     Wlatch(ch, targetpot.potByte, LOCK_DATA);       // Lock it, but don't output it yet
18     WByteEnable(ch, targetpot.potByte, ENABLE_BYTE); // Output the data to pulsed lines
19     BlockingDelay_us( 1000 );
20
21     WriteData(ch, encoderphaseA | encoderphaseB);    // Set next phase to bus
22     Wlatch(ch, targetpot.potByte, UNLOCK_DATA);     // unlock previous data
23     Wlatch(ch, targetpot.potByte, LOCK_DATA);       // lock new data
24     BlockingDelay_us( 1000 );
25
26     WriteData(ch, encoderphaseB);                    // And the next phase to bus
27     Wlatch(ch, targetpot.potByte, UNLOCK_DATA);     // unlock previous data
28     Wlatch(ch, targetpot.potByte, LOCK_DATA);       // lock new data
29     BlockingDelay_us( 1000 );
30
31     WByteEnable(ch, targetpot.potByte, DISABLE_BYTE); // Disable data output
32     Wlatch(ch, targetpot.potByte, UNLOCK_DATA);     // Release data from outputs
33
34     BlockingDelay_us( ppotcommand.tbw_steps );      // Wait for time between individual steps
35 }
36 ChannelToInput( ch );                               // Set channel back to input
37 uart_PutC('+');                                     // Send end of simulation
38 }

```

#### KUVIO 21. Enkooderian pyöryksen simuloiminen

Enkooderian pyöryksen simuloiminen on melko suoraviivaista. Aiemmin esitetystä kuviosta 5 käy ilmi enkooderian liityntöjen tilat pyöryksen eri vaiheissa. Kuviossa näkyy, myös kuinka liityntöjen tilojen järjestys muuttuu pyöryssuunnan vaihtuessa.

Funktion alussa tarkastetaan mihin suuntaan enkooderia tullaan pyörittämään.

Tämän perusteella valitaan enkooderian liityntöjen tilat ensimmäiseen vaiheeseen.

Liityntöjen tilat siirretään testausväylän ulostuloihin ensimmäisessä vaiheessa, joka alkaa riviltä 16 ja kestää riville 19. Ensimmäistä vaihetta pidetään yllä noin 1 millisekunnin ajan. Tämän jälkeen muodostetaan toinen vaihe, jossa enkooderian liitynnät ovat yhtä aikaa samassa tilassa. Kolmannessa vaiheessa, alkaen riviltä 26, vain toinen enkooderian liitynnöistä on enää aktiivisena. Neljäs vaihe on, kun liitynnät palaavat takaisin ei-aktiivisiin tiloihin. Tässä kohtaa odotetaan enkooderian pyöryksenopeutta vastaavan viiveen verran. Tätä toimintorakennetta toistetaan



niin monta kertaa, kuin simuloitavia askeleita halutaan tehdä. Funktiossa näkyy riveillä 15 ja 37 sarjaliikenteeseen liittyvä merkin lähettämiskomento. Kommento-ohjelmassa on toiminto, joka katkaisee komennon suorittamisen, jos testeri ei vastaa PC:lle takaisin noin 10 sekunnin kuluessa. Tämä estää komento-ohjelman jumittumisen, mikäli testerissä tapahtuu jokin häiriö. Riippuen enkooderin pyörittämisnopeudesta ja askelten määrästä koko simulointi voi kestää jopa useita kymmeniä sekunteja. Tämän takia testeri lähettää aina yhden askeleen alussa tiedon PC:lle, että simulointi on päällä. Näin ollen komento-ohjelma tietää, että testeri ei ole jumittunut.

#### 9.4 Näppäinpainalluksien simulointi

```

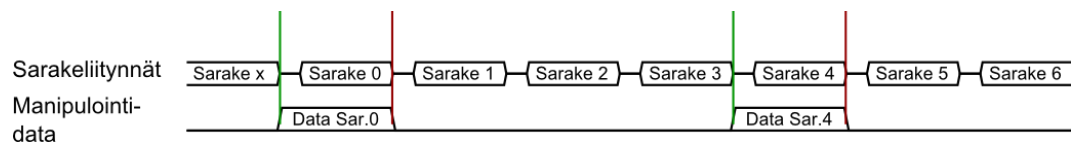
1 // Perform output clearing
2 RByteEnable ( channel, btnmatrixinfo.columnbyte, DISABLE_BYTE);
3 ChannelToOutput( channel );
4 WLatch      ( channel, btnmatrixinfo.rowbyte,  UNLOCK_DATA );
5
6 WriteData( channel , 0x00 );
7
8 WLatch      ( channel, btnmatrixinfo.rowbyte , LOCK_DATA) ;
9 ChannelToInput( channel );
10 RByteEnable ( channel, btnmatrixinfo.columnbyte, ENABLE_BYTE);
11
12 reft = TITC; // Get reference time
13 VICINTENABLE |= ( ulong32 ) ( 1<<VIC_EINT3 ); // Enable button simulation
14
15 RByteEnable( channel, btnmatrixinfo.columnbyte, ENABLE_BYTE ); // Enable column signals
16 WByteEnable( channel, btnmatrixinfo.rowbyte, ENABLE_BYTE ); // Enable outputs
17
18
19 while( curholdtime < data.holdtime ) // Stay here while button simu. is active
20 {
21     uart_PutC( '.' ); // Send keep-alive signal to PC
22     curholdtime = GetTimeDiff( reft ); // Get elapsed time from timer
23 }
24 VICINTENCLEAR |= (ulong32) (1 << VIC_EINT3); // Stop simulating
25
26 WByteEnable( channel, 3, DISABLE_BYTE ); // Disable outputs
27 uart_PutC( '*' ); // Inform PC that button simulation is over
28 RByteEnable( channel, btnmatrixinfo.columnbyte, DISABLE_BYTE ); // Disable column signals
29 WLatch      ( channel, 3, UNLOCK_DATA ); // Disable output latch if active

```

#### KUVIO 22. Osa näppäinmatriisin manipuloinnin pääfunktioista

Näppäinpainalluksen simuloiminen käyttöliittymien näppäinmatriiseille on kaksiosainen. Toiminnallisuus koostuu kahdesta eri funktiosta. Kuviossa 22 on esitetty näppäinpainalluksen simuloinnin pääfunktion tärkeimmät osuudet. Testeri on ennen kuviossa esitettyä funktio-osuutta muuntanut sille lähetetyt näppäintiedot manipuloitidataksi, jota se hyödyntää näppäinpainalluksia simuloitaessa. Se on myös asettanut näppäinmatriisien sarakesignaalien kohdalle laskevan reunan keskeytykset käyttöön. Laskevan reunan tarkoituksena on ennakoida seuraava aktiivinen

sarake, jotta testeri ehtii ajoittamaan manipulointidatan oikeisiin signaaleihin. Näppäinmatriisi luetaan käyttöliittymissä pian sen jälkeen, kun jokin sarakesignaaleista on asetettu aktiiviseksi. Tämän takia näppäinmatriisin manipulointidatan tulisi olla luettavissa heti, kun sarakesignaali aktivoidaan. Jos manipulointidata ajoittuu väärin suhteessa sarakesignaaleihin, käyttöliittymäpaneeli tulkitsee näppäinpainalluksen väärin. Sarakesignaalien aktivoitumisten välissä on pieni viive, jonka aika manipulointidata voidaan aktivoida ennakkoon. Tällöin data on välittömästi validi sarakesignaalin aktivoituessa. Kuviossa 23 on esitetty manipulointidatan ajoitus näppäinmatriisin sarakesignaaleihin. Esimerkissä sarakkeiden 0 ja 4 näppäinsignaaleja manipuloidaan, muiden sarakkeiden kohdalla manipulointia ei tehdä.



KUVIO 23. Manipulointidatan ajoittaminen sarakesignaaleihin

Ennen varsinaisen painallussimuloinnin aloittamista funktiossa tyhjennetään näppäinmatriisin rivisignaalien lähtevä data. Rivillä 13 sallitaan mikro-ohjaimen GPIO-liityntöihin liittyvät keskeytykset, mikä mahdollistaa kuviossa 24 esitetyn keskeytysfunktion automaattisen kutsumisen. Itse pääfunktio ei tee simuloinnin aikana muuta kuin tarkkailee näppäimelle määritettyä painallusaikaa ja lähettää PC:lle tiedon simuloinnin käynnissä olosta. Se jää odottamaan niin kauaksi aikaa, kunnes näppäimen pitoaika on kulunut, minkä jälkeen se estää uusien GPIO-keskeytysten aktivoitumisen ja sammuttaa testausliittynät. Todellisen näppäinpainalluksen simuloimisen hoitaa keskeytysfunktio.

```

1 void BtnMatrixCH1SimulationInt(void)
2 {
3     // Check what column was active
4     for( bitcounter = 0 ; bitcounter < btnmatrixinfo.n_columns + btnmatrixinfo.columnpadding ; bitcounter++ )
5     {
6         if( (IO0INTSTATF >> 16) & (1 << bitcounter) ) // CH1DATA: P0.16 ... P0.23
7         {
8             break;
9         }
10    }
11
12    bitcounter -= (btnmatrixinfo.columnpadding);
13
14    if( bitcounter < btnmatrixinfo.n_columns ) // Valid column signal?
15    { // Yes, perform manipulation data load to outputs
16
17        RByteEnableCH1(btnmatrixinfo.columnbyte, DISABLE_BYTE); // Disable column reading
18        CH1TOOUTPUT( ); // Testbus to output
19        WLatchCH1(btnmatrixinfo.rowbyte, UNLOCK_DATA); // Release previous data
20        CH1DATA = manipulationdata[ ( bitcounter+1 ) * btnmatrixinfo.n_columns ]; // Write next columns manipulation data
21        WLatchCH1(btnmatrixinfo.rowbyte, LOCK_DATA); // Lock new data to outputs
22        CH1TOINPUT( ); // Testbus to input
23        RByteEnableCH1(btnmatrixinfo.columnbyte, ENABLE_BYTE); // Enable column reading
24    }
25    IO0INTCLR = (ulong32) ( btnmatrixinfo.columnmask << 16 ); // Clear any active interrupt
26 }

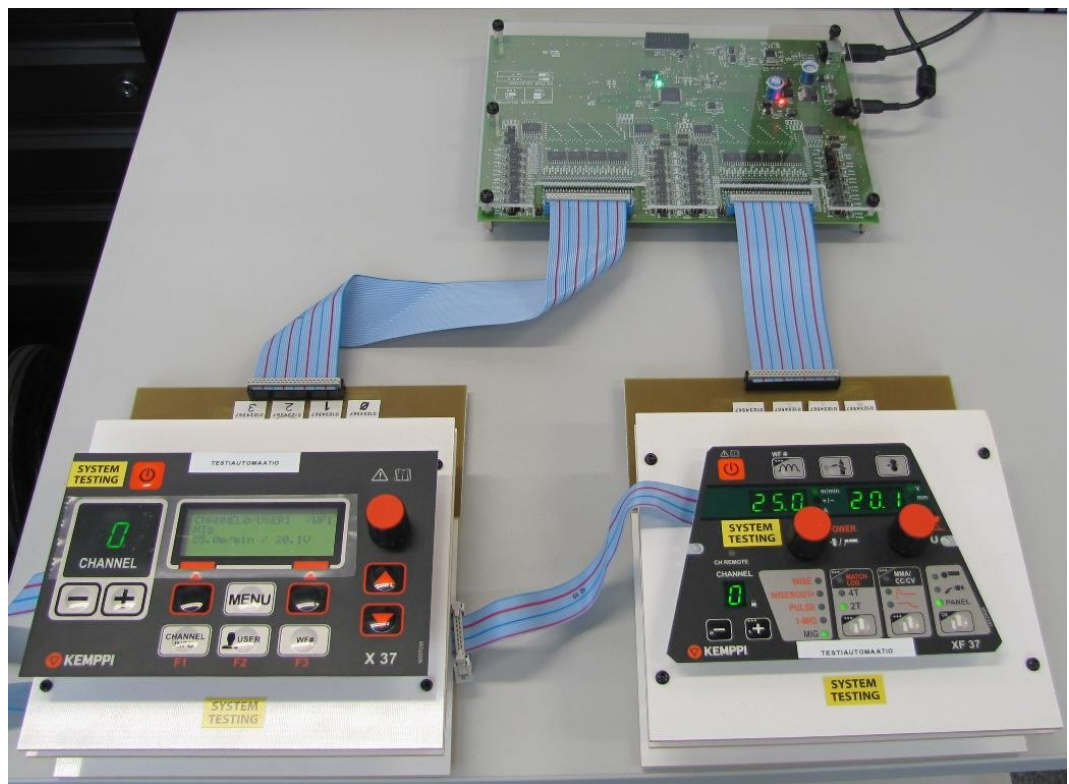
```

## KUVIO 24. Näppäinmatriisin manipuloinnin keskeytysfunktio

Keskeytysfunktioa kutsutaan aina, kun näppäimien sarakesignaaleissa esiintyy laskeva reuna. Keskeytysfunktio tarkistaa ensimmäiseksi, mistä liittynästä keskeytys tuli. Tämän jälkeen se lataa manipuloitaviin näppäinliityntöihin seuraavaksi aktivoituvan sarakesignaalin manipuloitidatan. Mikäli manipulointia ei tehdä jonkin sarakesignaalin kohdalla, sen sarakkeen manipuloitidataksi on merkitty nolladata, jotta lähtöpuskurit eivät aktivoidu. Tämän jälkeen keskeytysfunktio nollaa keskeytyslipun.

## 10 LOPPUTULOS JA JATKOKEHITYS

Työn lopputuloksena saatiin toteutettua alkuperäiset vaatimukset täyttävä testeri-laite. Toiminnassa oleva testeri on nähtävissä kuviossa 25, jossa näkyy myös testeriin lisätty läpinäkyvä muovisuojus. Laitteen ominaisuuksia testattaessa testeri on osoittautunut varsin luotettavaksi. Esimerkiksi testerillä tehtiin yli 1000 peräkkäistä lukuoperaatiota X37-asetuskäyttöliittymän LCD-näytölle, eikä testeri kertaakaan lukenut tietoja virheellisesti. Samantapaiset testit suoritettiin myös muille testerin ominaisuuksille. Testerin on myös havaittu olevan merkittävä apu ohjelmistotestauksessa, ja sitä päästiin hyödyntämään heti laitteen valmistumisen jälkeen oikeissa testitilanteissa.



KUVIO 25. Valmis käyttöliittymätesteri toiminnassa

Testerä suunniteltaessa sille ideoitii myös erilaisia lisätoiminnallisuksia, joita ei toteutettu tässä opinnäytetyössä. Ominaisuudet kuitenkin huomioitiin jatkokehitystä ajatellen, minkä takia joillekin testerin mikro-ohjaimen kytkemättömille liittynöille jätettiin juotospisteet, joista liittynät voidaan ottaa haluttaessa käyttöön. Testerin mikro-ohjain sisältää useita AD-muunninkanavia, joita voitaisiin käyttää

erilaisten jännitteiden mittaukseen. Näillä voitaisiin esimerkiksi mitata hitsausko-neissa käytettävien analogisten kaukosäätimien jännitteitä. Mikro-ohjaimen PWM ja DA-liitynnöillä voitaisiin puolestaan simuloida analogisten kaukosäätimien toimintaa.

Testeriin voitaisiin myös lisätä ominaisuus, että se tunnistaisi automaattisesti siihen liitetyn neulapedin ja sitä myöten testattavan käyttöliittymäpaneelin. Siten testerisi voisi automaattisesti valita oikeat toiminnot kyseiselle käyttöliittymäpaneelille ja ilmoittaa virheestä, mikäli käyttäjä yrittää toteuttaa väärää toimintoa väärälle käyttöliittymälle. Tällä hetkellä käyttäjän pitää kertoa testerille siihen liitetyn käyttöliittymän tyyppi komento-ohjelman kautta, tai asettamalla tiedot asetustiedostoon. Komento-ohjelma tarkistaa, onko toiminto sallittu kyseiselle paneelille, mutta se ei voi estää toiminnon toteuttamista väärin ilmoitetun paneelityypin kohdalla.

## 11 YHTEENVETO

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa käyttöliittymätesteri järjestelmähitsauskoneille. Työ tehtiin Kemppi Oy:n toimeksiannosta. Laitteen tarkoitus oli mahdollistaa toistuvien rutiininomaisten testien automatisointi, jolla vähennetään ohjelmistotestauksessa testaajien manuaalista työkuormaa. Testeri liitetään tulevaisuudessa osaksi suurempaa testiautomaatiojärjestelmää.

Työ aloitettiin kartoittamalla laitteen vaatimukset. Vaatimuksiin kuuluivat liitettävyyden PC:hen USB:llä, testauskanavien määrä ja vaatimukset testerin toiminnallisuuksista. Työskentelyn kannalta todennäköisesti merkittävin vaatimus oli, että suunnittelussa oli käytettävä vain Kempin tuotantolinjalla käytössä olevia komponentteja. Vaatimus mahdollisti sen, että laite voidaan tuottaa Kempin omalla elektroniikkatehtaalla. Ennen laitteen varsinaisen elektroniikan suunnittelua tutustuttiin Kempin FastMig X -hitsauslaitteiston käyttöliittymäpaneeleihin ja niiden sisältämien komponenttien toimintaan. Näiden käyttöliittymäkomponenttien toiminnan perusteella pystyttiin kartoittamaan testausliittymien määrä sekä niiden sähköisiä ominaisuuksia.

Elektroniikan suunnittelu aloitettiin toteuttamalla varsinaiset testausliittymät. Testausliittymien toteutuksessa hyödynnettiin väylälaajennusta, jossa kahdeksanbitoinen testausväylä laajennettiin 32 testausliittymäksi. Näistä liittymäistä 16 on puhtaasti sisääntuloja ja 16 sisääntuloja sekä ulostuloja. Testauskanavia laitteeseen toteutettiin kaksi, jolloin testausliittymä on yhteensä 64 kappaletta. Testerin teholähde toteutettiin sekä hakkuri-, että lineaarityyppisenä. Laitteeseen toteutettiin myös galvaanisesti erotettu USB-liittymä, jonka kautta testeriä voidaan ohjata.

Lopputuloksena saatiin toteutettua toimiva ja alkuperäiset vaatimukset täyttävä testeri. Laite on osoittautunut käytännössä erittäin luotettavaksi. Sitä kyetään jatkossa hyödyntämään ohjelmistotestauksessa.

Työ oli monipuolinen ja haastava. Se sisälsi monia erilaisia työalueita, kuten elektroniikkaa, piirilevysuunnittelua, sulautetun järjestelmän ohjelmointia, Windows -ohjelmointia ja mekaanista suunnittelua. Kaikkein aikaa vievin osuus työssä oli piirilevyn suunnittelu. Raportissa ei kuitenkaan kuvattu kaikkia mahdollisia työvaiheita. Testerin Windows komento-ohjelman kehittäminen jätettiin raportista

kokonaan pois, koska sen koettiin sisältävän liikaa sellaista tietoa, jota ei haluttu julkiseksi. Samoin testerin mikro-ohjaimen laiteohjelmaa ei kuvattu tarkasti, vaan siitä esiteltiin lähinnä testaustoimintojen funktioiden pääasiallinen toiminta.

Työ kuvastaa mielestäni erinomaisesti insinöörin osaamistaitoja erilaisilla osa-alueilla. Lisäksi sopeutuminen rajoitettuun komponenttivalikoimaan oli mielenkiintoinen haaste. Aiemmin omien henkilökohtaisten elektroniikkaprojektien kohdalla olin tottunut, että sopiva komponentti voidaan etsiä komponenttitoimittajalta ja tilata välittömästi. Tässä työssä kuitenkin jouduin keksimään ratkaisun eri ongelmiin käyttämällä vain niitä komponentteja, joita Kempin tuotannolla oli käytössä. Olen erittäin tyytyväinen saavutettuun lopputulokseen.

## LÄHTEET

Cockrill, C. 2011. Understanding Schmitt Triggers. Texas Instruments [viitattu 25.2.2014]. Saatavissa: <http://www.ti.com/lit/an/scea046/scea046.pdf>

Fairchild. 2013. HCPL06XX. Fairchild semiconductor [viitattu 29.3.2014]. Saatavissa: <http://www.fairchildsemi.com/ds/HC/HCPL0611.pdf>

FTDI. 2011. FT232BL/BQ USB UART IC. Future Technology Devices International Limited [viitattu 8.3.2014]. Saatavissa: [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232BL\\_BQ.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232BL_BQ.pdf)

FTDI. 2009. FTDI Drivers Installation guide for Windows 7. Future Technology Devices International Limited [viitattu 17.2.2014]. Saatavissa: [http://www.ftdichip.com/Support/Documents/AppNotes/AN\\_119\\_FTDI\\_Drivers\\_Installation\\_Guide\\_for\\_Windows7.pdf](http://www.ftdichip.com/Support/Documents/AppNotes/AN_119_FTDI_Drivers_Installation_Guide_for_Windows7.pdf)

Hitachi. 1999. HD44780U Dot Matrix Liquid Crystal Display Controller/Driver. Hitachi [viitattu 27.2.2014]. Saatavissa: <http://www.datasheetarchive.com/dl/Datasheet-011/DSA00197146.pdf>

Kasurinen, P. 2013. Ohjelmistotestauksen käsikirja. 1. painos. Jyväskylä: Docendo.

Kemppi Oy. 2014a. FastMig X valokuvat. Intranet materiaalipankki.

Kemppi Oy. 2014b. Kemppi: Kemppi yrityksenä. Kemppi Oy [viitattu 14.1.2014]. Saatavissa: [http://www.kemppi.com/inet/kemppi/fi/akp.nsf/frameset/frameset?OpenDocument&left=/inet/kemppi/fi/akp.nsf/WEB\\_Nav?OpenView&navcat=Kemppi&main=/inet/kemppi/contman.nsf/0/95C80EAA5FCDB5A2C22571950036CC4D?opendo&top=/inet/kemppi/fi/akp.nsf/Top?ReadForm&topcat=Kemppi](http://www.kemppi.com/inet/kemppi/fi/akp.nsf/frameset/frameset?OpenDocument&left=/inet/kemppi/fi/akp.nsf/WEB_Nav?OpenView&navcat=Kemppi&main=/inet/kemppi/contman.nsf/0/95C80EAA5FCDB5A2C22571950036CC4D?opendo&top=/inet/kemppi/fi/akp.nsf/Top?ReadForm&topcat=Kemppi)

Koskinen, J. 2004. Mikrotietokonetekniikka. Sulautetut järjestelmät. 1. uudistettu painos. Helsinki: Otava.



Maxim Integrated. 2006. MAX7301 Datasheet. Maxim Integrated [viitattu 24.2.2014]. Saatavissa:

<http://datasheets.maximintegrated.com/en/ds/MAX7301.pdf>

NASA. 2001. Public Lessons Learned Entry: 0938. National Aeronautics and Space Administration [viitattu 8.3.2014]. Saatavissa:

<http://www.nasa.gov/offices/oce/llis/0938.html>

NXP. 2012a. 74HC573; 74HCT573 Octal D-type transparent latch; 3-state. NXP [viitattu 22.3.2014]. Saatavissa:

[http://www.nxp.com/documents/data\\_sheet/74HC\\_HCT573.pdf](http://www.nxp.com/documents/data_sheet/74HC_HCT573.pdf)

NXP. 2012b. BAT54 series schottky barrier diodes. NXP [viitattu 22.3.2014].

Saatavissa: [http://www.nxp.com/documents/data\\_sheet/BAT54\\_SER.pdf](http://www.nxp.com/documents/data_sheet/BAT54_SER.pdf)

TI. 2013a. LM1575/LM2575/LM2575HV. Texas Instruments [viitattu 27.3.2014].

Saatavissa:

<http://www.ti.com.cn/general/cn/docs/lit/getliterature.tsp?genericPartNumber=lm1575&fileType=pdf>

TI. 2013b. LP38691, LP38693. Texas Instruments [viitattu 29.3.2014]. Saatavissa:

<http://www.ti.com/lit/ds/symlink/lp38693-adj.pdf>

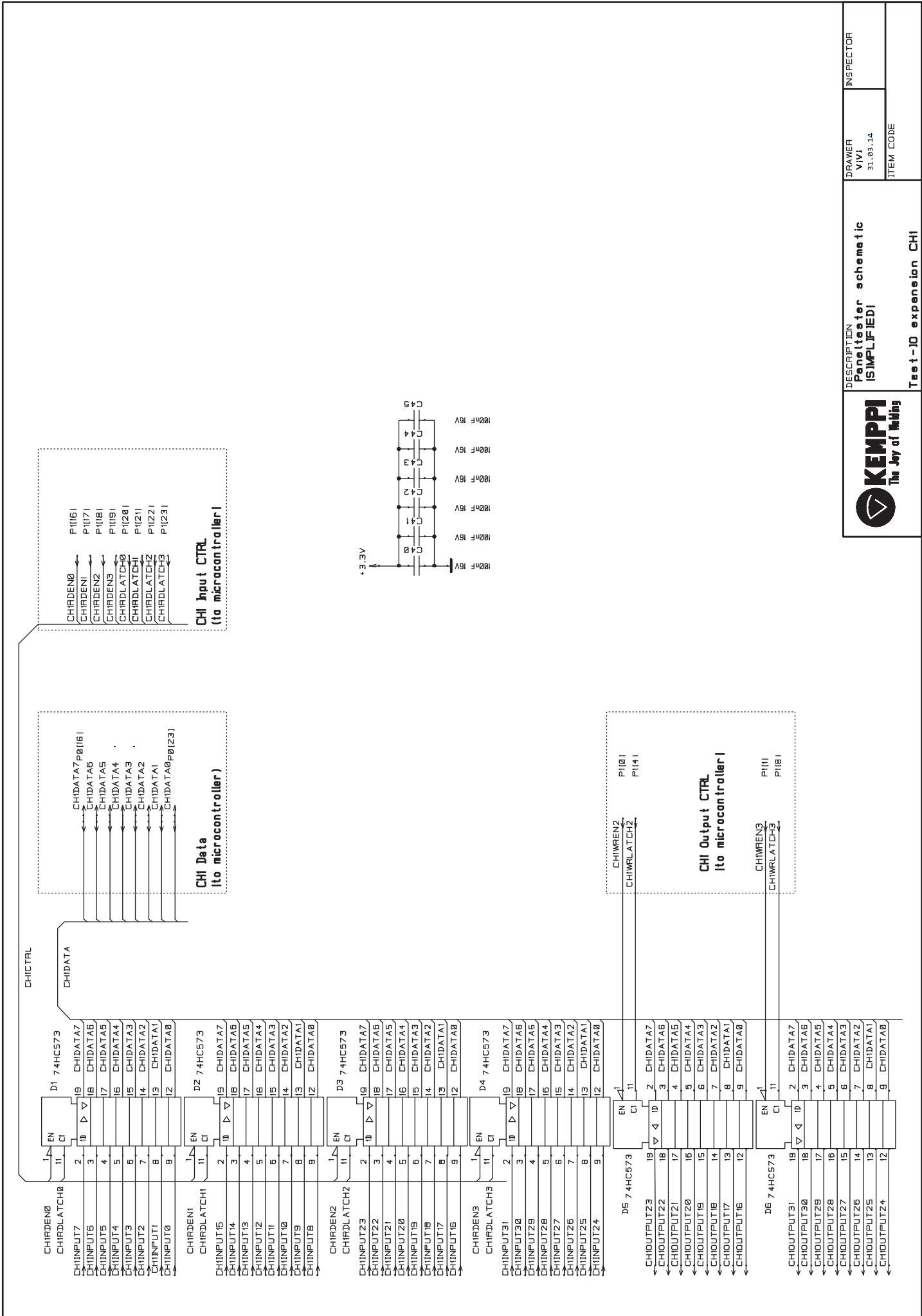
USB-IF. 2000. USB 2.0 Specification. USB Implementers Forum, Inc [viitattu 29.3.2014]. Saatavissa:

[http://www.usb.org/developers/docs/usb20\\_docs/usb\\_20\\_031114.zip](http://www.usb.org/developers/docs/usb20_docs/usb_20_031114.zip)

## LIITTEET

Liite 1. Käyttöliittymätesterin yksinkertaistettu kytkentäkaavio

Liite 2. Käyttöliittymätesterin testausfunktioiden vuokaaviot



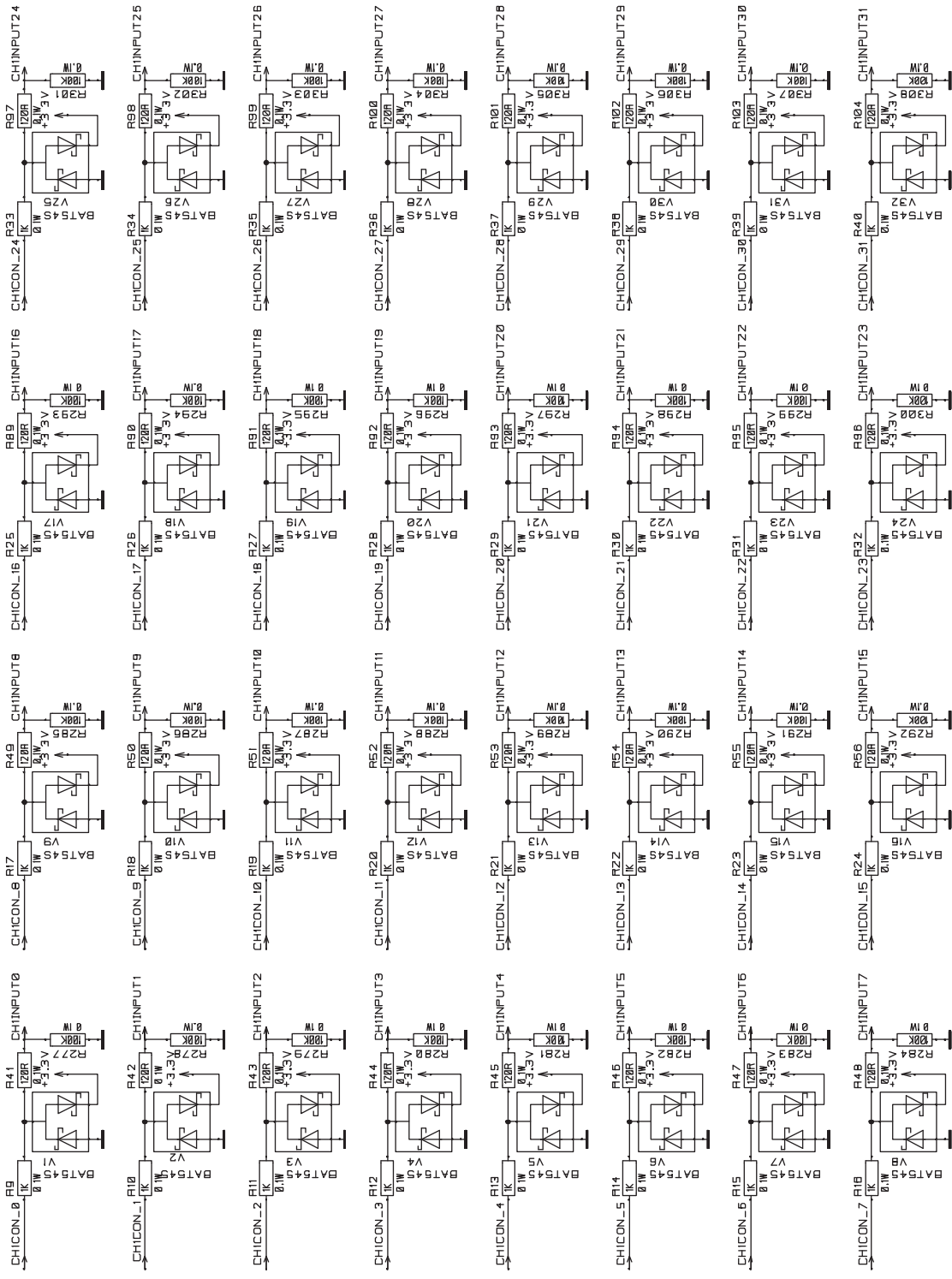
DESCRIPTION  
Paneltester schematic  
ISIMPLIFIED

Test-ID expansion CHI

DRAWER  
VIWI  
31.03.14

INSPECTOR

ITEM CODE



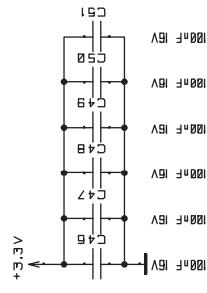
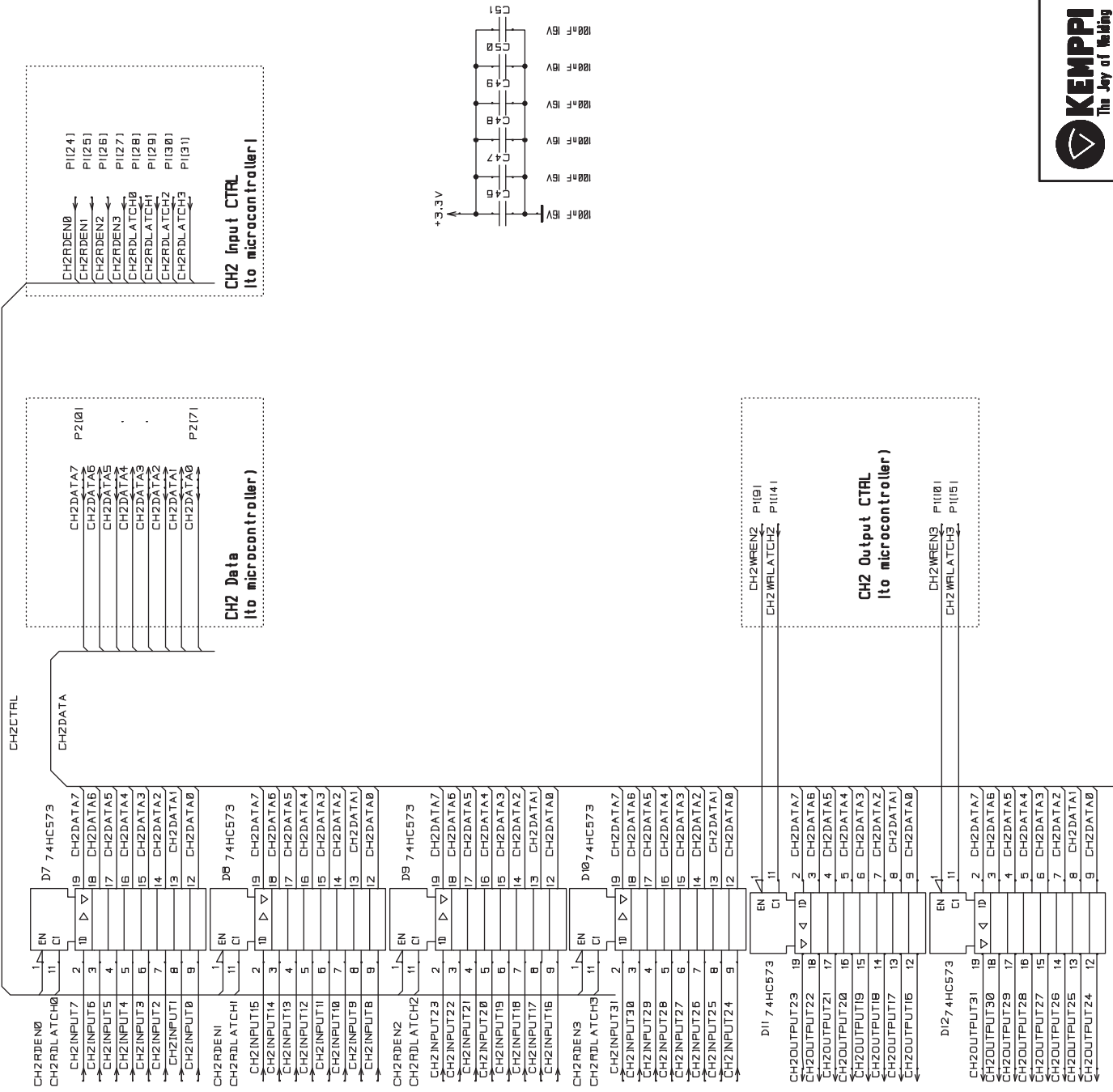
DESCRIPTION  
**Penetration tester schematic (SIMPLIFIED)**

INSPECTOR  
DRAWER  
VIVI  
31.03.14

ITEM CODE



Test - IO input clamps CHI



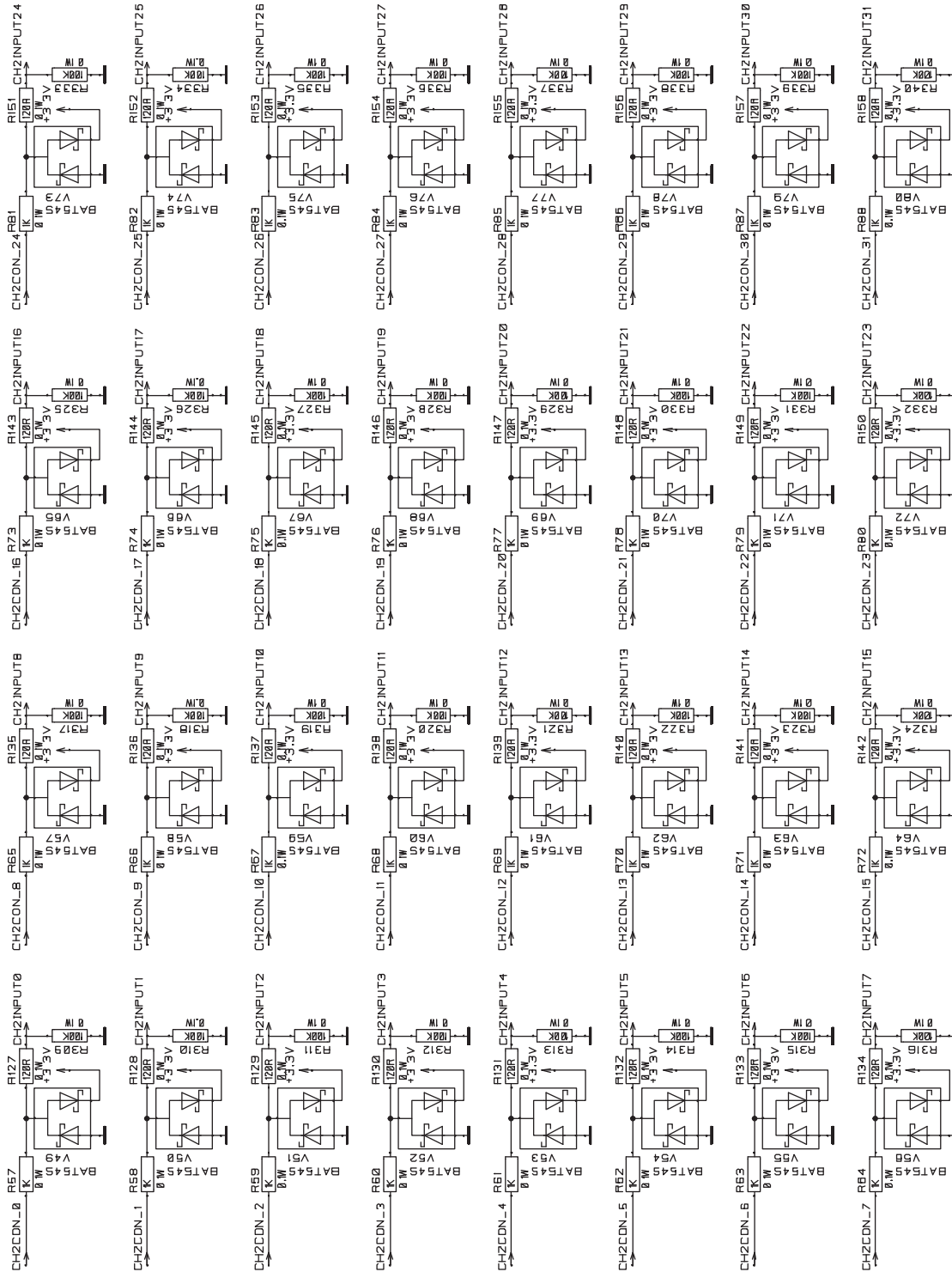
DESCRIPTION  
**Perilletester schematic**  
**ISIMPLIFIED!**

DRAWER  
VIVI  
31.03.14

INSPECTOR

ITEM CODE

Test-ID expansion CH2



**KEMPPI**  
The Joy of Making

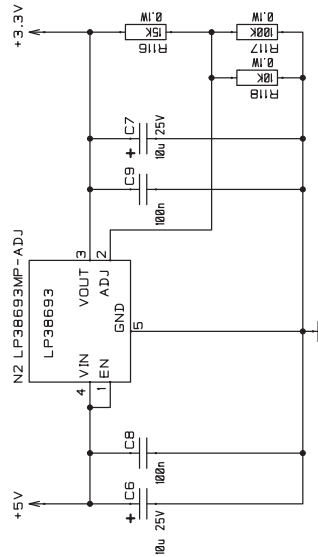
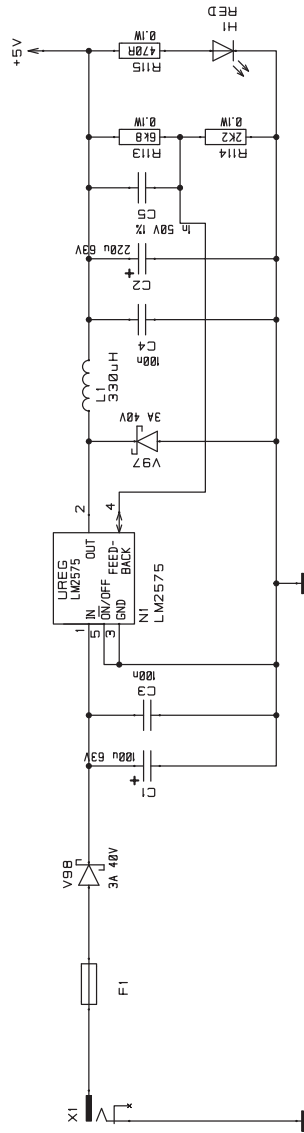
DESCRIPTION  
**Paneltester schematic**  
ISIMPLIFIED!

DRAWER  
VIVI  
31.03.14

INSPECTOR

ITEM CODE

Test-10 input clamps CH2

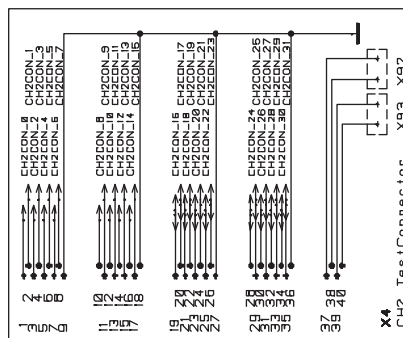
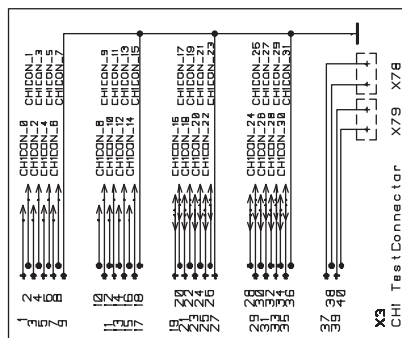


DESCRIPTION Paneltester schematic (SIMPLIFIED)	INSPECTOR
	POWER SUPPLY
DRAWER VVI 31.03.14	ITEM CODE









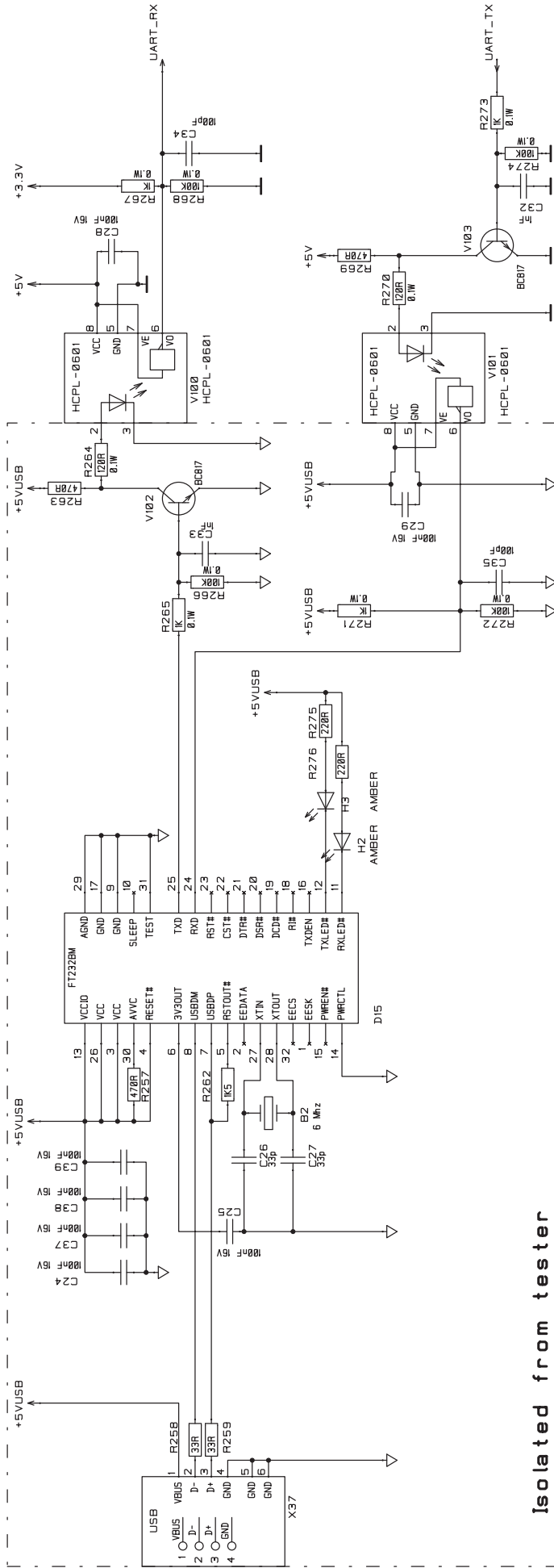
DESCRIPTION  
Paneltester schematic  
ISIMPLIFIED

DRAWER  
VW:  
31.03.14


INSPECTOR

ITEM CODE

Connectors



Isolated from tester



**KEMPPFI**  
The Joy of Welding

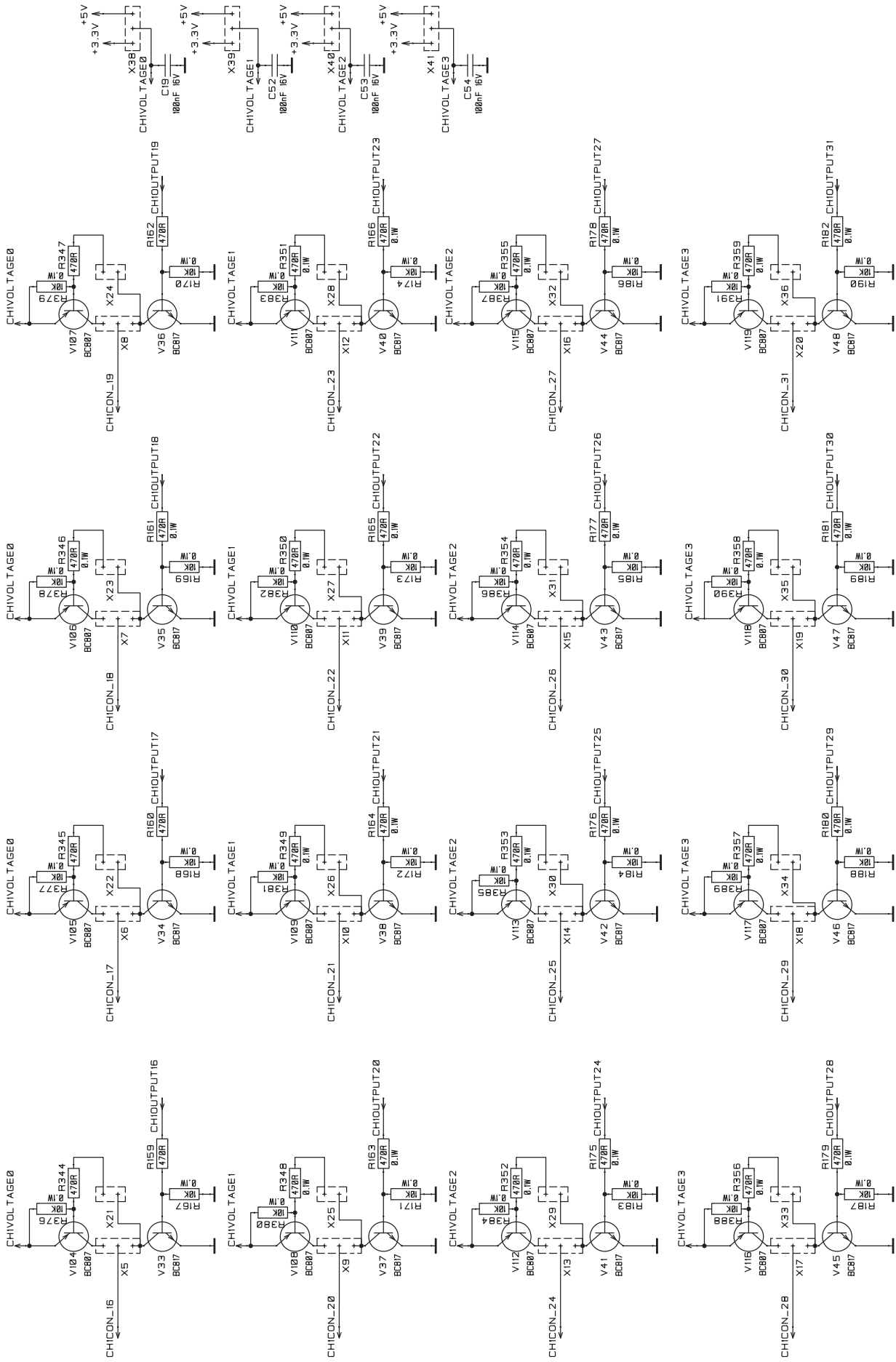
DESCRIPTION  
Panel tester schematic  
(SIMPLIFIED)

USB - UART & Isolation

DRAWER  
VW1  
31.03.14

INSPECTOR

ITEM CODE



**KEMPPI**  
The Joy of Welding

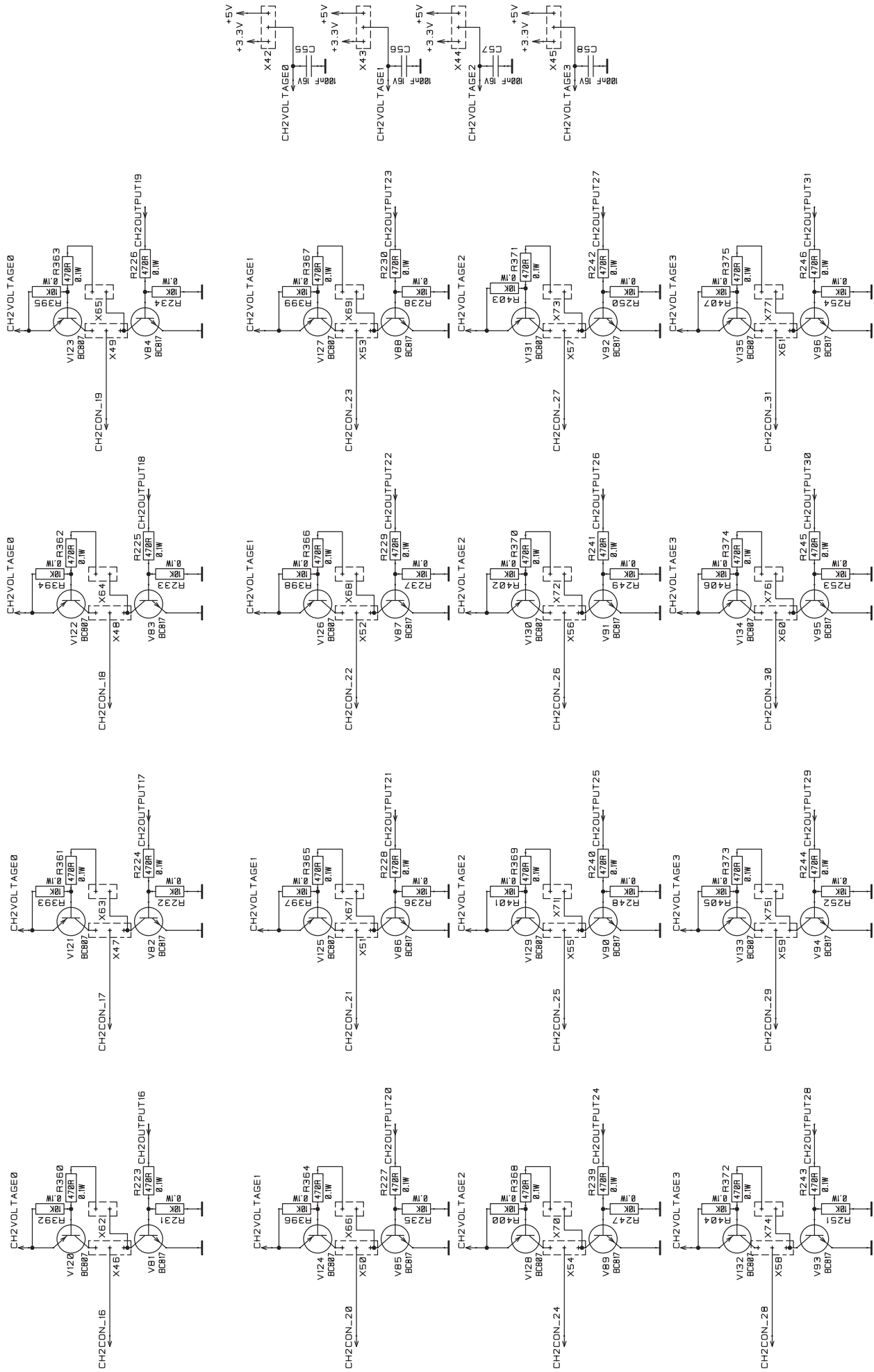
DESCRIPTION  
Paneltester schematic  
(SIMPLIFIED)

Test-IO outputs CHI

DRAWER  
VW1  
31.03.14

INSPECTOR

ITEM CODE



**KEMPPI**  
The Joy of Welding

DESCRIPTION  
Panel (tester schematic  
(SIMP LIFIED))

Test-IO outputs CH2

DRAWER VW1 31.03.14	INSPECTOR
ITEM CODE	

