

Opinnäytetyö (AMK)
Elektroniikan koulutusohjelma
Tietoliikennejärjestelmät
2013

Vitali Moilanen

LED-AJONEUVOVALOJEN TESTAUSLAITTEISTON SUUNNITTELU



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Elektroniikka | Tietoliikennejärjestelmät

2013 | 56 sivua

Ohjaajat: ins. (YAMK) Yngvar Wikström, ins. (AMK) Miikka Puomila

Vitali Moilanen

LED-AJOVALOJEN TESTAUSLAITTEISTON SUUNNITTELU

Tässä työssä laajennettiin valmiina olevaa LED-ajovalojen testauslaitteistoa suunnittelemalla uusi käyttöliittymä ja lisäämällä uusia hyödyllisiä ominaisuuksia.

Laitteeseen on lisätty tulosten esittelyä ja tarkastelua varten pieni 2-rivinen nestekidenäyttö ja USB:n avulla toteutettu yhteys tietokoneen kanssa. Lisäksi oli suunniteltu tietokoneohjelma Microsoft Visual Basicilla. Ohjelma on tarkoitettu laitteen hallintaan.

Testauslaitteen ytimenä toimii Atmelin atMega48PA-mikro-ohjain ja sen sisäinen ohjelma on tehty AVR Studio 5:n avulla. Ohjelmointikielenä on käytetty C-kieli.

Laite on todettu toimivaksi ja kelvolliseksi tuotantotestausta varten. Kehitettävät asiat liittyvät lähinnä laitteen fyysiseen kokoon ja ergonomiaan. Laitteen hallintaa varten suunniteltu graafisella käyttöliittymällä varustettu tietokoneohjelma on väitetty liian monimutkaiseksi, mutta toisaalta se antaa paljon mahdollisuuksia säätöön.

Laitteen avulla saadaan nopeammin testattua lopputestauksen ja vianetsinnän aikana useita tuotteita. Sen toimintoja voidaan säätää graafisen käyttöliittymän avulla kytkemällä laite USB-porttiin.

ASIASANAT:

atMega48, AVR, LED, RS-232, Testaus, UART, USB, Visual Basic

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics | Telecommunication systems

2013 | 56 pages

Instructor(s): M.Sc Yngvar Wikström, B.Sc Miikka Puomila

Vitali Moilanen

DEVELOPING A TESTING SYSTEM FOR AUTOMOTIVE LED DEVICES

The purpose of this thesis was to improve an existing testing system for automotive led devices. The purpose was achieved by designing a new interface and adding new features.

The testing equipment was improved with the addition of a small 2-row LCD -screen for viewing results and a computer connection with a USB-interface. A computer program written in Microsoft Visual Basic 6.0 was developed in order to manage the device.

The improved testing system uses an atMega48PA -microcontroller. The program in the microcontroller was built on AVR Studio 5.0 and the programming language is C.

The testing system was built and tested. The testing system works fine and can be used on the production floor. However, the system can be further developed by building in a smaller case and being simpler to use. The managing testing device can also be further improved by building a computer program with a graphical interface with Visual Basic.

The developed device can better and faster test led lights for automotive systems. The functions can be customized with the computer program via the USB -port.

KEYWORDS:

atMega48, AVR, LED, RS-232, Testing, UART, USB, Visual Basic

SISÄLTÖ

1 JOHDANTO	1
2 LÄHTÖKOHDAT	2
2.1 Yleisesti LED-tuotteista	2
2.2 LED-valot liikenneajoneuvoissa ja niiden käytön haasteet	3
2.3 Testattavat laitteet	4
2.4 Testausmenetelmä	4
2.5 General Tester 1.0	5
2.5.1 Mikro-ohjainpiirilevy	5
2.5.2 Toiminta kahdella jännitealueella ja profiilin muodostaminen	7
2.5.3 Profiilin valinta	7
2.5.4 Testauslaitteen muut ominaisuudet	8
2.5.5 Testauslaitteen laajennusportit	8
2.5.6 Testauslaitteen teholähde	9
2.6 Työn tavoitteet ja vaatimukset	10
3 TOIMINTAYMPÄRISTÖ, VÄLINEET JA LAITTEET	12
3.1 AVR -mikrokontrollerit	12
3.1.1 AVR -mikrokontrolleriperheet	13
3.1.2 AVR-mikrokontrollereiden rakenne	15
3.2 AVR -ohjelmointiympäristö	16
3.3 STK-500 - AVR -tuotekehitysalusta	16
3.3.1 STK500-alustan liitännät	17
3.3.2 Lyhyt käyttöohje STK500 alustalle	18
3.4 AVR Studio kehitysympäristönä	19
3.5 Microsoft Visual Basic 6.0 ohjelmointikieli	20
4 TESTAUSLAITTEEN KEHITYS	21
4.1 Työn alkuvaiheet ja hahmottelu	21
4.2 LCD-näyttö	22
4.3 Yhteyden rakentaminen tietokoneeseen	24
4.3.1 RS-232 -portti	25
4.3.2 USB-portti	25
4.3.3 Portin valinta testauslaitetta varten	26

4.3.4 CP2102 USB-UART -muunnin	26
4.4 EEPROM-muisti ja profiilien säilytys	29
4.5 Tietokoneohjelma profiilien luontia ja siirtoa varten	29
4.5.1 Datasiirto tietokoneen ja testauslaitteen välillä	30
4.5.2 Ohjelman toimintojen ja ominaisuuksien suunnittelu	32
4.5.3 Ohjelman graafisen käyttöliittymän suunnittelu	32
4.5.4 Ohjelman asetusten suunnittelu	34
4.5.5 Ohjelman virhetilanteet ja niiden käsittely	35
4.5.6 Ohjelman käyttöohjeen laatiminen	36
5 TESTAUSLAITTEEN LOPPUKOKOONPANO	37
6 KÄYTTÖÖNOTTO JA TESTAUS	39
7 PÄÄTELMÄT	41
8 YHTEENVETO	43
LÄHTEET	44

KÄYTETYT LYHENTEET (TAI SANASTO)

ADC	Analog-to-Digital Converter, analogi-digitaalimuunnin
ALU	Arithmetic Logic Unit, aritmettinen laskentayksikkö
CAN	Controller Area Network, automaatioväylä
DAQ	Data Acquisition systems, data-analysointijärjestelmä
DIP (DIL)	Dual In-Line Package, nelikulmainen komponenttikotelo, jossa liitokset kahdella puolella
DMA	Direct Memory Access, suora muistiyhteys
EPROM	Erasable Programmable Read-Only Memory, uudelleenkirjoitettava ohjelmoituva muisti
EEPROM	Electrically Erasable Programmable Read-Only Memory, sähköisesti uudelleenkirjoitettava ohjelmoituva muisti
IO	Input-Output, sisääntulo/ulosotto
ISP	In-System Programming, kytkennän sisäinen ohjelmointi
LED	Light Emitted Diode, valoa säteilevä diodi
LCD	Liquid Crystal Display, nestekidenäyttö
LPT	Line Print Terminal, rinnakkaisportti
PWM	Pulse-Width Modulation, pulssileveysmodulaatio
RAM	Random Access Memory, käyttömuisti
SRAM	Static Random Access Memory, staattinen käyttömuisti
TQFP	Thin Quad Flat Package, ohut nelikulmainen komponenttikotelo, jossa liitokset jokaisella puolella
TTL	Transistor-transistor logic, transistori-transistori logiikkataso

UART	Universal Asynchronous Receiver/Transmitter, universaali asynkroninen sarjamuotoinen vastaanotin-lähetin
USART	Universal Synchronous/Asynchronous Receiver/Transmitter, universaali synkroninen-asynkroninen sarjamuotoinen vastaanotin-lähetin
USB	Universal Serial Bus, universaali sarjaliikenneväylä

1 JOHDANTO

Monissa elektroniikkayrityksessä valmiiden tuotteiden laatu ja toimintakelpoisuus testataan erilaisten testauslaitteiden ja testausjärjestelmien avulla. Usein nämä testausjärjestelmät tarvitsevat lisää virittelyä, joka parantaa niiden käytettävyyttä sekä testausnopeutta. Tämä opinnäytetyön tarkoituksena on jatkokehittää olemassa oleva LED-ajoneuvovalojen testauslaitteisto. Opinnäytetyö on tehty Tepcomp Oy:lle, josta jatkossa puhutaan toimeksiantajayrityksenä. Testauslaitteen alkuperäisessä versiossa käytetään AVR-mikrokontrolleria atMega48PA:ta. Sen ohjelmointikielenä on ollut C-kieli.

Vastaavia opinnäytetöitä on tehty aiemminkin, Turun Ammattikorkeakoulun kirjastosta löytyy yli 10. Lähin vastaava opinnäytetyö on Jyri Kallion, joka on tehnyt vastaavan testauslaitteen aihiotestausta varten käyttäen samoja lähtökohtia. Hän on tehnyt myös tarkkuusparannuksen, mutta tässä työssä ei käytetä hänen parannettua kytkentäänsä, sillä työt oli tehty lähes samanaikaisesti. [1]

Työlle on asetettu seuraavat vaatimukset. Sen tarkoitus on olla vianetsintälaite, käsi-käyttöinen, mutta sen testausprofiileita pitäisi päästää helposti muokkaamaan tietokoneohjelman avulla. Tietokoneyhteyden pitäisi toteuttaa USB:n avulla. Testauslaitteiston täytyy olla mahdollisimman monipuolinen. Hyviä toimintoja saa lisätä esimiehen suosituksesta. Laitteessa on oltava LCD-näyttö testituloksien tarkastelua ja profiilin valintaa varten.

Tässä opinnäytetyössä ei panosteta mittaustuloksien epätarkkuuden korjaamiseen eri mittausten menetelmistä, vaikka ongelma on ollut alkuperäisessä versiossa. Tämän työn tarkoituksena on ollut tutkia tiedonsiirtoa tietokoneen avulla, näytön lisäämistä valmiina olevaan projektiin ja graafisella käyttöliittymällä varustetun ohjelman suunnitteluun. Tuloksista pitäisi tehdä selkeä dokumentaatio toimeksiantajalle. Tämän opinnäytetyön tarkoitus on avata mahdollisuudet rakentaa entistä parempia ja monipuolisempia kuin ennen, sekä tietokoneavusteisia testausjärjestelmiä toimeksiantoyrityksessä.

2 LÄHTÖKOHDAT

2.1 Yleisesti LED-tuotteista

LED on puolijohdekomponentti, joka tarkoittaa diodia, joka alkaa säteillä valoa, kun sen läpi kulkee sähkövirta. Mitä suurempi on LEDin läpi kulkeva virta, sitä suurempi on sen valoteho. LEDin yli vaikuttava jännite vaihtelee vain suhteellisen vähän silloin, kun se säteilee valoa. LEDit ovat siis virtaohjattavia. LEDin valotehon suuruus ei ole kuitenkaan lineaarisesti riippuvainen LEDin läpi kulkevasta virrasta ja vaihtelee LEDin tyypistä riippuen. LEDin pohjalle on rakennettu viime aikoina paljon uusia tuotteita. [2]

LED-tuotteet ovat viime aikoina alkaneet valloittaa uusia käyttöalueita. Niitä käytetään mm. valaistuksessa, merkkisignaalien antamisessa, valomainoksissa ja autoteollisuudessa. Lisäksi on olemassa LEDien avulla rakennettuja televisioita ja tietokonenäyttöjä. Kuvassa 1 on esimerkkisovellus, jossa käytetään LEDejä. Kyseessä on hehkulampun korvaaja.



Kuva 1. Hehkulampun korvaaja E27-kantaan [3]

LEDien etuna hehkulamppuihin verrattuna on pienempi virrankulutus, pienempi tilantarve, alhaiset toimintajännitteet, suurempi värivalikoima ja pidempi käyttöikä. LED:ien huonoja ja haastavia puolia ovat korkeampi hinta joissakin tapauksissa, suppea valaisukeila, säätöelektronikan ja virranrajoittimien tarve (eli LEDien virtalähteet sekä regulaattoriipiirit), värintoiston ongelmat, hieman erilainen, hieman epäluonnollinen valo

ja lämpötilaongelmat. Kuitenkin LED-tuotteiden kehitys ja parannus koko ajan jatkuu, niiden hinnat laskevat ja luotettavuus paranee [4].

2.2 LED-valot liikenneajoneuvoissa ja niiden käytön haasteet

Autoteollisuus on alkanut viime aikoina käyttää LED-valoja yhä useammin ajoneuvojen huomio-, vilkku-, jarru- sekä peruutusvaloina. Nykyisin LEDeja käytetään jopa ajovaloina, mm. Audi A8:ssa [5]. Niiden myötä auton valot kuluttavat vähemmän tehoa, mahtuvat pienempään tilaan sekä kestävät kauemmin. Ongelmana lähinnä on lämpö, koska LED-valot eivät siedä korkeita käyttöympäristön lämpötiloja, joten ne kuuluisi eristää lämpöä tuottavista paikoista [6]. Koska LED-valot tuottavat vähemmän lämpöä ympäristöön, ne saattavat talvella aiheuttaa toisen ikävän ilmiön. LED-valot eivät yleensä pysty kovinkaan hyvin sulattamaan päälle muodostunutta lumen tai jään pintaa, joten kuljettajan täytyy ottaa nämä asiat huomioon [4]. Toinen, vielä nykyisin ajankohtainen oleva ongelma, on LED-valojen korkeampi hinta, joten edullisiin autoihin niitä ei hankita vakiovarusteena.

Kolmas ongelma liittyy useimpien autojen suuntavilkkuautomaattikaan. Kun autossa on käytössä tavalliset polttimot suuntavaloina, auton sisäinen sähköjärjestelmä tarkistaa, että niiden lamput todella syttyvät, kun siihen syötetään jännite. Tapa, jolla voi tarkistaa hehkulampan syyntymisen voi olla vaikka valoanturin avulla tai kulkevan virran avulla. Valoantureista koostuva järjestelmä olisi liian monimutkainen ja vaatinut autoon lisää elektroniikkaa sekä sähköjohtoja. Sen takia käytetään yleensä jälkimäistä tapaa eli tarkistetaan kulkevaa virtaa. Kun tavallinen hehkulamppu syttyy, se kuluttaa hetkellisesti paljon virtaa, minkä jälkeen virran kulutus laskee ja tasoittuu. Tämä ilmiö on kaikissa tavallisissa hehkulampuissa ja halogeenilampuissa. Auton elektroniikkajärjestelmä on suunniteltu siten, että se tarkistaa juuri heti virransyötön yhteydessä olevan sähkövirtapulssin lampun läpi. Jos se on liian suuri (oikosulku) tai puuttuu kokonaan (katkos), järjestelmä ilmoittaa siitä kuljettajalle esimerkiksi ääni- tai valomerkin avulla ohjauspaneelissa. Monissa autoissa vikailmoitus on toteutettu vilkkujen käytön aikana siten, että ohjauspaneelissa vilkkuu nopeasti vilkkumerkki, kun normaalisti se vilkkuu hitaasti [7].

LED-valot kuluttavat paljon vähemmän virtaa kuin hehkulamput, Lisäksi ne eivät yleensä käyttäydy samalla tavalla kuten hehkulamput. Jos autojen valot vaihdetaan suoraan LED-valoihin, täytyy joko säätää auton elektroniikkajärjestelmää uudelleen tai lisätä

LED-valoihin sähköpiiri, joka osaa imitoida hehkulamppua syttymisen yhteydessä. Tämän jälkeen se käyttäytyy, kuten LED-lamppu. Toisin sanoen tarkoitus on, että LED-lamppu alkuvaiheessa kuluttaa hetkellisesti paljon virtaa ja sen jälkeen virrankulutus laskee.

2.3 Testattavat laitteet

Eräs asiakas on tehnyt toimeksiantoyritykselle tilauksen, joka käsitti useiden erilaisten autoihin tarkoitettujen LED-valojen tuotannon ja testauksen. Asiakas antoi tarkkaat kriteerit laitteiden kuluttamille virroille. Laitteet ovat yhdistelmävalaisimia, joissa on toteutettu LEDien avulla kolme toimintoa yhdessä: sivuvilkkuvalo (kirkas keltainen), jarruvalo (kirkas punainen) sekä takavalon valo (himmeä punainen).

Tuotteet on suunniteltu toimimaan laajalla käyttöalueella. Kuitenkin ajoneuvoissa on käytössä vain kaksi erisuuruista jännitettä. Nämä jännitteet ovat 12 V ja 24 V autotyypistä riippuen. Näin olleen kyseiset LED-valot ovat yleiskäyttöisiä eli sopivat sekä 12 V:n, että 24 V:n sähköjärjestelmän omaaviin autoihin. 24 V:n sähköjärjestelmä on yleensä käytössä vain raskaissa ajoneuvoissa.

Jokaisessa LED-ajoneuvovalossa täytyy testata vilkkuvalo, jarruvalo sekä käyttövalon sähkökulutusvirrat sekä sivuvilkkuvalon käynnistyksen aikana tapahtuva hetkellinen huippuvirrankulutus, josta kerrotaan luvussa 2.4. Maajohdin on yhteinen ja jokaista toimintoa varten on oma liitäntäjohtimensa. Näin olleen täydessä varustuksessa olevassa ajoneuvon takavalaisimessa, joka sisältää jarru-, taka- ja vilkkuvalon, on 4 liitäntää (3 toimintoliitäntää ja yhteinen maa).

2.4 Testausmenetelmä

Edellisessä luvussa mainittuja tuotteita pitää testata tuotannossa siten, että jokaiselle tehtaalla tehdylle tuotteelle tehdään tietyt testit. Kyseistä testausta on vaikea tehdä käsin, sillä se vie paljon aikaa ja riskinä on se, että työntekijä saattaa helposti erehtyä vertailemalla tuloksia, sillä arvoja on paljon (3 toimintoa \times 2 raja-arvoa = 6 arvoa) ja testit suoritetaan kahdella jännitteellä (6 arvoa \times 2 jännitettä = 12 arvoa), jolloin tarvitaan joko kaksi virtalähdettä tai ainakin säädettävä virtalähde. Silloin se olisi vasta vienyt pitkää testausaikaa, kun työntekijä joutuisi joka kerta säätämään oikea jännite.

Huippuvirran testaus olisi kaikista vaikein, koska olisi tarvetta rakentaa lisäkytkentä, esim. vastuksen ja digitaalisen oskilloskoopin avulla, jossa on liipaisuominaisuus (trigger).

Tuotannon kannalta sopiva ratkaisu olisi rakentaa automatisoitu sähköinen testauslaite, joka mahdollistaa yllämainittujen testien suorittamisen nopeasti kerrallaan. Työntekijän täytyisi vain kytkeä testattava laite testauslaitteeseen ja painaa testin aloituspainiketta. Testien jälkeen testauslaite näyttää työntekijälle tuloksen, onko tuote päässyt testeistä läpi vai ei. Jos tuote on päässyt testeistä läpi, se viedään pakattavaksi, jos ei – yritetään ottaa selvää, johtuuko työntekijän väärästä kytkennästä vai onko tuotteessa jokin vika. Jos on vika, otetaan selvää, voiko sitä korjata vai tehdäänkö korvaava uusi tuote.

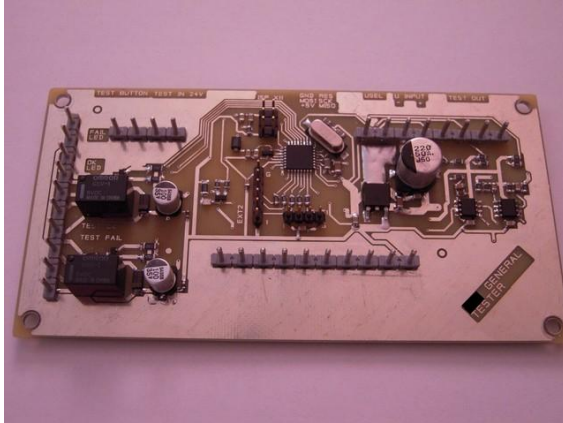
2.5 General Tester 1.0

Toimeksiantoyrityksellä oli projektina kehitetty General Tester -testauslaite, jota on käytetty runkona tässä opinnäytetyössä. Testauslaitteita kutustaan usein tuotantotiloissa testereiksi, vaikka nimitys on epävirallinen, tosin lyhyt ja selkeä. Testauslaitteen alkuperäinen suunnittelija on yrityksen elektroniikkasuunnittelijainsinööri. Vaatimukset testauslaitteelle oli asettanut asiakas, jonka tuotteita täytyy testata, ennen kuin ne toimitetaan. Testauslaitteella pystyy testaamaan erilaisia autojen LED-valoja, joissa on kolme toimintoa ja vilkkukanavan käynnistyksen huippuvirtaa. Tulokset testeistä saa varsin yksinkertaisessa muodossa: joko hyväksytty tai sitten ei. Tulosten esitystä varten on kaksi pientä LED-merkkivaloa, joista toinen on punainen (yksi tai kaikki testit epäonnistuivat) ja toinen vihreä (kaikki testit onnistuivat).

Alkuperäinen testauslaite koostuu seuraavista osista: mikro-ohjainpiirilevy, hakkurivirtalähde, muuntaja ja kotelo liittimiseen. Testauslaite on itsenäinen laite, eikä tarvitse toimiakseen muita erillisiä mittalaitteita tai jännitelähteitä. Se kytketään 230 V:n verkkojännitteeseen ja liittimiin kytketään erilliset sovittimet tuotteiden piirikortteja varten, joita sanotaan usein myös JIG:eiksi.

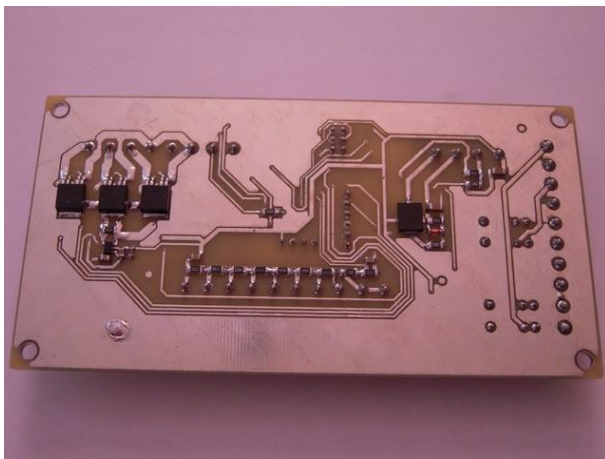
2.5.1 Mikro-ohjainpiirilevy

Kuvassa 2 on esitetty pääpiirilevyn yläosa. Siellä on suurin osa komponenteista ja kaikki sen liittimet. Piirilevyn materiaalina on käytetty FR4:a.



Kuva 2. Mikro-ohjainpiirilevyn yläpuoli

Kuvassa 3 on mikro-ohjainpiirilevyn alaosa. Alaosassa on vain pintaliitoskomponentit, joita oli rationaalista siirtää sinne tilan säästön ja lyhyemmän johdotuksen takia.



Kuva 3. Mikro-ohjainpiirilevyn alapuoli.

Kuten kuvista jo huomaa, piirilevyssä kaikki peruskomponentit ovat pintaliitostyyppisiä, pois lukien liittimet ja releet, jotka ovat läpiladottuja.

Piirilevyn reunoissa on paksummat piikkityyppiset liittimet, joiden kautta kytketään syöttöjännitteet, testauspainike, monivalintakytkin ja testattavan laitteen syöttöliittimet. Pienet liitinrimat keskellä on tarkoitettu laajennuskäyttöön ja mikro-ohjaimen käsittelyn ISP-tiedonsiirtoa varten.

2.5.2 Toiminta kahdella jännitealueella ja profiilin muodostaminen

Autojen LED-valojen toiminta kahdella jännitteellä on otettu huomioon myös testausvaiheessa. Autojen LED-valot testataan ensin 13,5 V:n suuruisella jännitteellä ja sen jälkeen 28 V:n jännitteellä. Nämä jännitteet on valittu siksi, koska akkujen jännite pienemmällä kuormalla voi olla nimellistä suurempi, joten LED-valoille täytyy taata toimivuus akkujen ylärajatoleransseja huomioon ottaen. Koska LED-valot kuluttavat eri jännitteillä toisistaan eroavat virrat, tarvitaan tuplasti enemmän vertailuarvoja (3 kanavaa \times 2 raja-arvoa (minimi-maksimi) + 3 kanavaa \times 2 raja-arvoa (minimi-maksimi) = 12 vertaisarvoa tuotetta kohti). Kun halutaan testata erilaisia LED-valojen tuoteperheitä, joiden arvot poikkeavat toisistaan, tarvitaan aina lisätä uusi testausprofiili, joka sisältää ainakin nuo kaikki 12 arvoa. Kaikki nuo arvot löytyivät mikro-ohjaimen ohjelmakoodista ja ne olivat sijoitettu suoraan vakioarvoiksi koodin sisällä.

General Testerin pitää siis pystyä toimimaan ja syöttämään testattavalle laitteelle sekä 13,5 V:n että 28 V:n jännitteen. Näinhän se on tehtykin. Koska mikro-ohjain ja muut komponentit kuluttavat vain vähän virtaa, joten ne saavat 5 V:n jännitteen pientehoisesta jänniteregulaattorista, joka on myös pääpiirilevyllä. Tässä tapauksessa pääpiirilevy pystyy toimimaan sekä 13,5 V:n että 28 V:n jännitteellä.

Koska jännite testaus hetkellä ei saa poiketa kuin $\pm 0,5$ V, ohjelmakoodissa on otettu huomioon jännitevalvonta. Näin on olemassa erilliset sallitut minimi- ja maksimiarvot sekä 13,5 V:n, että 28 V:n -jännitteitä varten. Huippuvirran mittaamisen yhteydessä mitataan myös jännite, joka vaikuttaa testattavan laitteen yli. Jos mitattu jännite poikkeaa enemmän kuin on sallittu, testitulokset hylätään (eli lopputulos on hylätty, vaikka kaikki virrat olisivat muuten kunnossa).

2.5.3 Profiilin valinta

Testauslaitteen alkuperäisversiossa oli otettu jo huomioon nuo yllä mainitut asiat, mutta ne olivat toteutettu siten, että testausprofiilit olivat kiinteitä (static) arvoja, joiden minimiarvo on 0 ja maksimiarvo on 255 eli 8-bittisiä lukuja. Valmiiksi oli tehty 8 erilaista profiilia, joita voi valita kiertokytkimen avulla. Kiertokytkinperiaatteella tarkoitetaan tässä tapauksessa sellaista menetelmää, jossa kiertokytkimen ja useiden vastuksien avulla valitaan tietty resistanssi, jonka suuruus asettaa mikro-ohjaimen ADC-tulolle tietyn jän-

nitearvon. Kun kyseistä jännitearvoa verrataan valmiiksi olevassa taulukossa oleviin arvoihin, niin sen perusteella saadaan tieto siitä, missä asennossa kiertokytkin on siinä vaiheessa. Kun kiertokytkin on asetettu tarvittavaan asentoon ja käyttäjä painaa testauslaitteessa olevaa Test -näppäintä, alkaa testaus valitulla profiililla.

2.5.4 Testauslaitteen muut ominaisuudet

Alkuperäisessä testauslaitteessa oli valmiiksi tehty erilaisia hyödyllisiä lisäominaisuuksia. Kaksi relettä, joita voi kytkeä päälle, kun testi onnistuu tai epäonnistuu. Niiden tarkoitus oli alun perin sytyttää Test-Ok tai Test-Fail merkkianto-LEDeja ja kytkeä virrat pois testattavasta laitteesta, kun testi on suoritettu.

Toinen alkuperäinen käyttömahdollisuus oli, että voidaan ohjata testauksen alkua Test-In liitännän kautta. Tämä on hyödyllistä esimerkiksi silloin, kun testauslaite on osa isompaa kokonaisuutta tai liitetty esimerkiksi automaatiojärjestelmään, jossa testausta on automatisoitu myös mekaanisesti.

Testauslaitteessa on myös seuraavia ohjelmallisesti toteutettuja ominaisuuksia: kanavan oikosulkutarkistus, testattavan jännitteen varmistus ja testattavan jännitteen valinta ohjaamalla hakkuriteholähdettä TTL-tason signaalilla.

2.5.5 Testauslaitteen laajennusportit

Testauslaitteen suunnittelija on tehnyt varsin hyvän ratkaisun, että tyhjänä jääneet mikro-ohjaimen portit on johdotettu erillisiin piikkirimaliittimiin. Tämä ratkaisu on todella hyvä, sillä voidaan laajentaa testauslaitteen toimintoja helposti, jopa ilman piirilevymuutoksia. 5 V:n ja maa-liitännät olivat myös johdotettu piikkirimaliittimiin, joten laajennusosiin voidaan syöttää käyttöjännitettä suoraan niiden kautta. Silloin täytyy vain varmistaa, että piirilevyllä oleva 7805-jänniteregulaattori kestävä ylijäävän tehon.

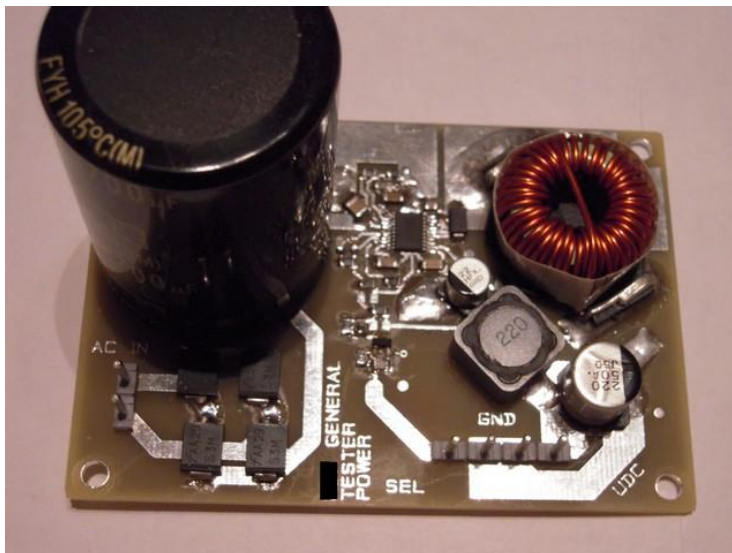
Yksi tärkeimmistä liittimistä oli UART/USART (Rx ja Tx), johon voidaan kytkeä sarjaliitännällä varustettu LCD-näytön sekä sovittimen, jonka avulla muodostetaan yhteys tietokoneeseen. Tämän liittimen yhteydessä on myös +5 V ja GND -nastat.

Toinen tärkeä piikkiliitin sisälsi portin *PORTC:n*, *PC3*, *PC4* ja *PC5* nastat, joiden avulla voidaan toteuttaa ohjauksen esimerkiksi painikkeilla. Tässäkin liittimissä on myös 5 V:n ja maaliitäntä.

Kolmas piikkirimaliitin oli tehty mikro-ohjaimen ISP-ohjelmointilaitetta varten, jolla olisi mahdollisuus siirtää mikro-ohjaimen uusi ohjelmakoodi tai kopioida vanha ohjelmakoodi mikro-ohjaimesta talteen. Tämä liitin oli tehty standardien mukaisesti ja sitä pystyy käyttämään monien tietokoneeseen liitettävien ISP-ohjelmointisovittimien kanssa. Tämän liittännän sijoittaminen mihin tahansa mikro-ohjaimella toteutettuun piirilevyyn on suositeltava, koska sen avulla voidaan korjata mahdolliset ohjelmavirheet sekä päivittää mikro-ohjaimessa olevaa dataa. Kulutuselektronikassa vastaava ratkaisu on tehty mm. RS-232 tai USB -liittännän avulla, esimerkiksi digitaalivastaanottimissa tai soittimissa. [7]

2.5.6 Testauslaitteen tehrolähde

Testauslaitteessa on oma tehrolähde. Se on tehty siten hieman erikoisella tavalla siten, että sen lähtöjännitettä pystyy valitsemaan joko 13,5 V:n tai 28 V:n suuruisena. Tehrolähteen piirilevy on kuvassa 4.



Kuva 4. Testauslaitteen hakkuritehrolähde.

Piirilevyn lisäksi tehrolähteen osaksi kuuluu myös muuntaja, joka on erillinen osa. Muuntaja on rengastyypinen, jossa on galvaanisesti erotetut ensiö- ja toisiokäämit.

Muuntajan teho on 50 VA, hyötysuhde on tuntematon, mutta ottaen huomioon rengas-sydänmuuntajien ominaisuudet, niiden hyötysuhteet ovat parempia kuin tavallisten pakkamuuntajien hyötysuhteet, eli suurusluokkaa 90–95 %.

Yleensä hakkurivirtalähteet on suunniteltu siten, että niihin syötetään suoraan sähköverkosta 230 V:n suuruinen jännite (joissakin maissa 110 V) ja ulos saadaan tarvittavat pienjännitteet, mutta tässä tapauksessa on tehty erikoinen ratkaisu, jolloin 230 V:n suuruinen vaihtojännite alennetaan 40 V:n suuruiseksi vaihtojännitteeksi, jota syötetään hakkuriosaan. Tämä on tehty lähinnä turvallisuussyistä, koska jos hakkuri olisi suunniteltu verkkokäyttöiseksi, tämä olisi vaatinut useampia komponentteja korkeajännitepuolella ja kyseisen tehonlähteen viritys sekä testaus aiheuttaisivat teknisesti vaaraa. Lisäksi sen olisi täytettävä useita sähköturvallisuusvaatimuksia.

2.6 Työn tavoitteet ja vaatimukset

Opinnäytetyön tavoitteena oli kehittää testauslaitetta niin, että siihen tulisi näyttö, josta voisi lukea mitattujen virtojen arvot. Myös laitteeseen pitäisi pystyä vaihtamaan mittausprofiileita, eli hyväksytyjen virtarajojen arvoja pitäisi pystyä muuttamaan. Ehdotuksena on se, että testauslaitetta pystyisi kytkemään tietokoneeseen ja oman tietokone-sovelluksen avulla muuttamaan virtarajojen arvoja. Asetusarvot pitäisi onnistua säilyttämään laitteen omassa muistissa. Ihanteellinen tilanne on se, että laitteessa pystyisi olemaan yhtä aikaa tallennettuna useita profiileja ja niiden vaihtaminen onnistuisi laitteen oman näytön ja painikkeiden avulla.

Testauslaite tulisi olla loppukäyttäjälle tarpeeksi selkeä ja helppokäyttöinen. Joka tapauksessa laitteeseen pitäisi liittää dokumentaatiota ja käyttöohjeita. Erikseen tulisi olla ohjeet perustestausta varten, joita ymmärtäisivät lähes kaikki käyttäjät ja toiset ohjeet edistyneille käyttäjille, joissa kerrotaan enemmän laitteen kaikista mahdollisuuksista ja profiilien teko-ohjeet.

Koska alkuperäinen testauslaite oli tarkoitettu sellaisia tuotteita varten, joissa on kolme kanavaa, niin haluttiin vielä tehdä muutokseen testattavien kanavien määrään. Lisäksi on ajateltu sitäkin, että voi tulla vastaan laitteita, jotka toimivat vain 12 V:n tai 24 V:n järjestelmissä. Profiilissa pitäisi siis onnistua määrittämään, mitkä kanavat testataan, minimi- ja maksimivirtojen raja-arvot sekä testausjännitteet. Esimerkiksi on olemassa tuotteita, joiden käyttöjännite on pelkästään 12 V, niin niitä ei testatakaan 24 V:n jännit-

teellä, koska muuten se voi jopa tuhoutua korkeamman jännitteen seurauksena. Lisäksi vilkkuvalon käynnistysvirran testausta pitäisi onnistua kytkemään pois käytöstä, koska kaikissa laiteissa sitä ei välttämättä ole (esim. on olemassa laiteita, joissa on vain peruutus- ja jarruvalot). Testauslaitteen alkuperäisessä versiossa sellaista ominaisuutta ei otettu huomioon, jolloin aina testattiin kaikki kanavat molemmilla jännitteillä.

Toimeksiantajan ehdotus on myös se, että testauslaite pystyisi havaitsemaan testattavan laitteen kytkemisen ja aloittamaan testaus ilman, että käyttäjän tarvitse painaa käynnistyspainikettä. Tämä ominaisuus olisi todella tärkeä tuotannon kannalta, sillä käyttäjä voisi suorita testit nopeammin ja käsitoimintaa vaaditaan silloin vähemmän. Lopuksi pitäisi sytyttää kaikki kanavat, että testaaaja voisi varmistaa himmennyskalvon tai suojalasien läpi, onko kaikki LEDit toimivia. Koska monissa laitteissa on sisäänrakennettu virtaohjain, ei voida tarkistaa varmuudella, ettei laitteessa ole LEDejä, jotka ovat oikosulussa. Myös joskus voi käydä niin, että kaikki LEDit ovat paikallaan, mutta osa on ladonnassa asettunut väärin, eli vinossa. Näiden laatuongelmien välttämiseksi täytyy testaajan varmistaa ainakin lopputestauksessa, että kaikki LEDit on asetettu oikein ja palaavat yhtä kirkkaasti. Isoissa erissä voidaan soveltaa myös optista, automaattista kameraa. Tässä tapauksessa kameraa ei käytetä, eli sen käyttöä ja ohjausta ei tarvitse ottaa huomioon.

Testauslaitteen on tarkoitus tulla tuotantotiloihin seuraaviin käyttötarkoituksiin: yksittäisten, aihioista irrotettujen korttien testaus, viallisten korttien vianetsintään sekä lopputestaukseen. Tarvittaessa laitetta voidaan käyttää myös tuotekehityksessä. Testauslaite liitetään erillisten sovittimien (JIG) avulla testattaviin kortteihin. Tavalliset yleisesti käytössä olevat banaaniliittimet, joiden kautta voidaan yhdistä mittausjohdot riittävät.

Ennen kaikkea on tärkeätä, että testauslaitteella pystyisi nopeasti testaamaan ja se olisi tarpeeksi helppokäyttöinen ja ergonominen. Testauslaitteen tulisi olla itsenäinen, koteloitu laite, jota voidaan suoraan liittää sähköverkkoon.

3 TOIMINTAYMPÄRISTÖ, VÄLINEET JA LAITTEET

Toimeksiantajaryitykseltä on saatu kaikki olemassa olevan testauslaitteen kytkentäkaaviot sekä ohjelmakoodi, joka on ollut testauslaitteen alkuperäisversion mikro-ohjaimen sisällä. Lisäksi toimeksiantajaryityksessä laadittiin spesifikaatiot ja työn tavoitteet niin suullisesti kuin kirjallisesti. Työtä varten on saatu myös mikro-ohjaimet atMega48 DIP -koteloissa, että niihin voi tutustua ja kokeilla ohjelmakoodia ja sähköisiä kytkentöjä, ennen kuin varsinainen muutostyö alkaa. Kuitenkin tilaus tehtiin vasta hiukan myöhemmin, joten aluksi AVR-ohjelmointia on kokeiltu atMega32:lla.

Alkutestausta ja AVR-ohjelmoinnin tutustumista varten oli valittu STK500-kehitysalusta, josta mainitaan luvussa 3.3. Sattumalta kyseistä alustaa käytetään jonkin verran tuotekehityksyksikössä toimeksiantajaryityksessä, joten ongelmatilanteissa on mahdollista saada tukea. Kyseessä on siis todella laadukas ja monipuolinen kehitysalusta, jolla onnistuu AVR -tuoteperheen mikro-ohjaimien testaus ja simulointi. Lisäksi käytettävissä on ollut analoginen oskilloskooppi, jos tarvitse joskus mitata tasoja ja jännitteitä kytkennässä tai kellopulssit. Johtuen siitä, että oskilloskooppi on analoginen, sillä ei onnistu signaalin tallennusta, josta voisi olla hyötyä digitaalipiirien analysoinnissa. Mikro-ohjaimen ohjelmointia ja koodin siirtoa varten on käytetty ilmaista Atmel AVR Studio 4.1 -ohjelmistokehitysovellusta. Siihen sisältyy myös STK500-alustan tuki, joten koodin siirto onnistuu helposti. Tietokoneelle tarkoitetun työn toisen puolen, eli hallintaohjelmiston suunnittelua varten on käytetty Microsoft Visual Basic 6.0 -ohjelmointisovellusta.

3.1 AVR -mikrokontrollerit

AVR -mikrokontrollerit ovat saaneet päivänvalon vuonna 1996, jolloin Atmel-yhtiö on julkaissut muokatun ns. Harvardin arkkitehtuurin pohjalta luodun RISC-mikrokontrollerin. Ideana oli luoda mikro-ohjain, jossa olisi flash-muisti ohjelmakoodia varten, silloin kun muut käyttivät vielä kertakäyttöisiä ROM -muisteja, EPROM ja EEPROM. Idean takana ovat kaksi norjalaista opiskelijaa, Alf-Egil Bogen ja Vegard Wollan. He olivat silloin olleet töissä nykyisessä Nordic Semiconductorissa, mutta myöhemmin siirtyivät Atmeliin ja jatkoivat idean kehitystä. Nimestä AVR ei ole vieläkään saatu selvää, mitä se oikein tarkoittaa ja mistä sanoista on peräisin. Yleinen tulkinta on "Advan-

ced Virtual RISK”, mutta toiset ovat sitä mieltä, että kyseessä onkin ”Alfin ja Vegardin RISK -mikrokontrolleri”. [9]

Kuten oli jo mainittu, AVR-mikrokontrollerit perustuvat Harvardin arkkitehtuuriin, joka tarkoittaa sitä, että ohjelmakoodi ja sen käsittelemä tieto sijaitsevat eri muistiosoiteavaruudessa. Käskykanta on RISK-ideologian läheinen, mutta poikkeaa siitä siten, että rekisterit eivät ole ortogonaalisia. AVR-mikrokontrollereissa on 32 8-bittistä yleisrekisteriä, jotka on yhdistetty yhteiseen rekisteritiedostoon. [9]

AVR-mikrokontrollerien käskykanta on varsin kehittynyt ja käsittää mallista riippuen 90–133 erilaista käskyä. Suurin osa käskyistä vie vain yhden muistisolun (16 bittiä). Suurin osa käskyistä suoritetaan yhden kellojakson aikana. [9]

3.1.1 AVR -mikrokontrolleriperheet

AVR-mikrokontrollerit jakautuu erilaisiin tuoteperheisiin, joilla on tietyt ominaisuudet, käyttökohteet ja hinta. Jokainen tuoteperhe sisältää useita erilaisia malleja.

AVR-tuoteperheitä ovat:

tinyAVR (mallimerkinnot ATtinyxxx)

- Maks. flash-muistin koko 16 kt, maks. käyttömuistin koko 512 tavua, maks. EEPROM 512 tavua
- Maks. IO-liitäntöjen määrä 4-18 (yhteensä liitäntöjen määrä maks. 6-32)
- Rajoitettu ulkoisten laitteiden kokoonpano

megaAVR (mallimerkinnot ATmegaxxx)

- Maks. flash-muistin koko 256 kt, maks. käyttömuistin koko 8 kt, maks. EEPROM-muistin koko 4 kt.
- IO-liitäntöjen määrä 23-86 (yhteensä liitäntöjen määrä 28-100)
- Laitteistopohjainen kertolasku moduuli
- Laajennettu käskykanta ja suurempi tuki ulkoisille laitteille.

XMEGA AVR (mallimerkinnot ATxmegaxxx)

- Flash-muisti koko jopa 384 kt:n asti, maksimi käyttömuistin koko (SRAM) 32 kt, maks. EEPROM-muistin koko 4 kt.

- 4-kanavainen DMA-ohjain
- Uudistunut tapahtumien tietojen käsittely [9]

Näiden yllä mainittujen tuoteperheiden lisäksi löytyy erikoismikrokontrollereita, jotka perustuu AVR-ominaisuuksiin, mutta sisältävät lisätoimintoja. Nämä ovat esimerkiksi mikrokontrollerit, joissa on laitteistotuki erilaisille tietoliikenneväylille (USB, CAN, LCD-ohjaimelle), sisäänrakennettu kaksisuuntainen radiomoduuli tai moottoriohjain. Eriksseen on vielä autoelektroniikkaa ja valaistusta varten suunnitellut mikrokontrollerit. [9]

Saman mallin mikrokontrollereita voi olla useampia versioita, jotka poikkeavat perusversiosta kotelon tyypillä, toimintajännitteellä tai poikkeavalla virrankulutuksella.

Tässä on lista erilaisista versioista ja niiden ominaisuuksista:

AT(mega/tiny)xxx — perusversio.

ATxxxL — pienemmällä jännitteellä toimivat versiot (2,7 V:n toimintajännite)

ATxxxV — pienemmällä jännitteellä toimivat versiot (1,8 V:n toimintajännite)

ATxxxP — pienemmän virrankulutuksen omaavat versiot (kulutusvirta jopa 100 nA Power-down tilassa).

ATxxxA — pienempi virrankulutus koko taajuus ja jännitealueella, sekä yhteensopivuus edellisten versioiden kanssa.

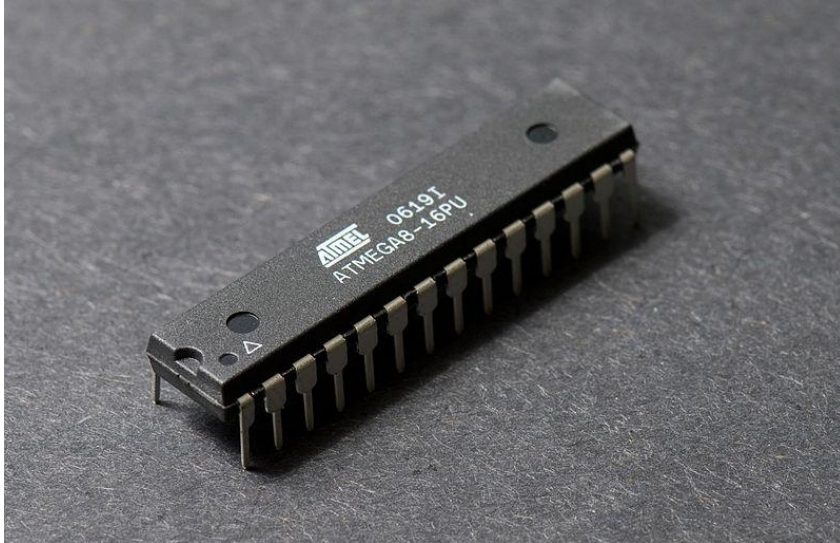
ATxxx-PI — DIP -kotelossa

ATxxx-PU — DIP -kotelossa, lyijytöntä juotosta varten

ATxxx-AI — TQFP -kotelo

ATxxx-AU — TQFP -kotelo, lyijytöntä juotosta varten [8]

Kuvassa 5 on esimerkki mikrokontrollerista. Kyseessä on ATmega8-16PU mikrokontrolleri, joka on DIP-koteloitu ja soveltuu lyijytöntä juotosta varten.



Kuva 5. ATMega8-mikrokontrolleri [10]

Mikrokontrollerin valinnassa täytyy ottaa huomioon näiden versioiden erot, koska väärin valittu mikrokontrollerin versio voi aiheuttaa ongelmia suunnittelussa, jatkosuunnittelussa ja toiminnassa. [11]

3.1.2 AVR-mikrokontrollereiden rakenne

AVR-mikrokontrollerit sisältävät ainakin ALU-yksikön, RAM-muistin, flash-muistin, EEPROM-muistin sekä IO-portit datasiirtoa varten. Lisäksi ne voivat sisältää muitakin toimintoja, kuten AD-muuntimet, analogiset komparaattorit, PWM -modulaattorit sekä tuet datasiirtoprotokollia varten. [9]

Sisäisen flash-muistin voi kirjoittaa uudelleen ainakin 10 000 kerta ja EEPROM-muistin kohdalla vastaava luku on jopa 100 000. [9]

Oskillaattorina voi valita sisäisen RC-piirin, ulkoisen kideoskillaattorin tai ulkoisen pulsiosihjauksen. Suurin toimintataajuus on yleensä 20 MHz, mutta AVR 32-bit perheen mikrokontrollerit toimivat jopa 150 MHz:n taajuuksilla. [9]

3.2 AVR -ohjelmointiympäristö

AVR -sovelluskehityksen ohjelmakoodin kirjoittamista voidaan tehdä vaikka millä tahansa tekstinkäsittelyohjelmalla, vaikkei tuo kuulostakaan kovin käytännölliseltä. Mutta varsinaisen ohjelman siirtämistä varten tarvitaan vielä vähintään kääntäjä ja flash-muistin kirjoittamista varten erillinen ohjelma. Laitteistosta tarvitaan lisäkytkentää, joka käsittää ohjelmointilaitetta. Asia on tehty viime aikoina helpommaksi, jolloin syntyi ohjelmointiympäristöjä, jotka käsittävät koodin tuottamista varten tarkkaan suunnitellun, helppokäyttöisen käyttöliittymän, kääntäjän ja flash-muistin kirjoittamista varten tarkoitettua liittymän.

Yleisimmin käytetyt ohjelmistot, joita käytetään AVR -ohjelmien kehityksessä, ovat seuraavat:

- AVR Studio (varsin monipuolinen, Atmelin oma)
- WinAVR
- Code::Blocks
- Bascom-avr (Basic-ohjelmointikielen tapainen)
- CodeVisionAVR
- IAR AVR
- Avrdude (ohjelman siirtoa varten)
- PonyProg (ohjelman siirtoa varten). [9]

Osa näistä ohjelmista on ilmaisia, osa maksullisia. Joistakin maksullisista ohjelmista saattaa löytyä ilmaisia, rajoitettuja versioita. Näin esimerkiksi Bascom-avr -ohjelmasta löytyy ilmainen versio, jossa ohjelmakoodin koko on rajoitettu 4 kt:uun. AVR-ohjelmien kehityksessä pääasiassa käytetään Assembly- tai C-kieltä. Yksi tämän ajan suosituimmista AVR-kehitysympäristöistä on Atmelin oma paketti nimeltä AVR Studio. [9]

Tässä työssä käytettiin siis AVR Studiota, kuten alkuperäinen General Testerin ohjelmistosuunnittelu.

3.3 STK-500 - AVR -tuotekehitysalusta

Kuten aikaisemmin mainittu, käytettiin STK500-testialustaa mikro-ohjaimen ohjelman suunnittelussa. Kyseisen tuotekehitysalustan on kehittänyt Atmel, eli se on yksi suosituimmista AVR-kehitysympäristöistä.

tuimmista alustoista AVR-tuoteperheen piirien ohjelmointiin. Alustan nimi on lyhenne sanoista Starter Kit. Alustan pakettiin kuuluu lisäksi seuraavat osat ja komponentit: RS-232 ohjelmointikaapeli, virransyöttösovitin (ilman teholaähdettä), ohjevihko, CD-ROM-levy, jolla on ohjelmistot ja sähköisessä muodossa oleva dokumentaatio, laattakaapelit ja 2-pin kaapelit sekä AT90S8515-mikro-ohjain. Kuva 6 esittää alustaa sekä oheistavikkeitä.



Kuva 6. STK500-kehitysalusta

Alustalla on lisäksi muitakin hyödyllisiä toimintoja ja laajennusmahdollisuuksia. Tietojen syöttöä varten on tarkoitettu 8 mikrokytkintä, joista jokainen on suojattu kontaktien värähtelystä. Tietojen lukua varten on tarkoitettu 8 pintaliitos-LEDiä, väriltään keltaiset. Sekä kytkimet että LEDit on sijoitettu riveittäin. Näin on helpompi esittää porttien tiloja binäärimuodossa. Lähes kaikki mikro-ohjaimen nastat on johdatettu levyllä oleviin pinneluihin, jotta niihin voidaan kytkeä erilaiset IO-laitteet ja ulkoiset kytkennät. [12]

3.3.1 STK500-alustan liitännät

STK500-alustalla on kannat useille yleisemmille läpiladottaville DIL-koteloissa oleville mikro-ohjaimille. Piirikantoja ovat 8-pinnisille, 20-pinnisille, 24-pinnisille ja 28-pinnisille soveltuvat. Jokaista kantatyyppiä levyllä on 2 kpl, joka on tehty sen takia, koska saman

määrän pinnejä omaavat mikro-ohjaimet sisältävät erilaisen kytkentäkannan. Levyn reunalla on sijoitettu kaksi laajennuskantaa, joita on tarkoitettu lisälaajennusmoduulien kytkemiseen, esim. STK501-laajennusmoduulia varten. [12]

Ohjelmointia varten alustalla käytetään RS-232 -liitäntä, eli sarjamuotoista tietokoneporttia. Alustan voi kytkeä myös USB/RS-232 -sovittimen kautta tietokoneeseen, esimerkiksi silloin, kun tietokoneessa ei ole RS-232:a. Kehitysalustalla on oma ISP-ohjelmointiosa, jonka avulla valmiiksi käännetty konekielinen ohjelma siirretään mikro-ohjaimen. Toinen ohjelmointivaihtoehto on High Voltage Serial Programming (HVSP), eli mikro-ohjaimen ohjelmointi korkealla jännitteellä. Käytännössä tämä tarkoittaa, että *RESET*-nastaa syötetään tavanomaista korkeampi jännitetaso (yleensä 11,5 - 12 V), jolloin pystytään ohjelmoimaan mikro-ohjainta normaalisti. [12]

Koska alustalla on valmiiksi suunniteltu useita erilaista piirikantaa, niin on myös useita ISP -tuloja niiden ohjelmointia varten. Jotta voidaan ohjelmoida tietyn mikro-ohjaimen, täytyy kytkeä oikeat ISP-tulopiikit yhteiseen ISP-lähtöön 6-pinnisen laattakaapelin avulla. Kyseinen kaapeli kuuluu alustan varusteisiin. Tarvittaessa voidaan käyttää ulkoista ISP -ohjelmointilaitetta, tuloliitännät ovat standardien mukaiset. [12]

STK500-kehitysalustalla on toinenkin RS-232 -portti, joka on yhdistetty levyssä olevaan MAX202-mikropiiriin. Kyseinen mikropiiriin tehtävä on muuntaa TTL-signaalit RS-232 tasoisiksi sekä päinvastoin. Toiminnot ovat lähes samanlaisia kuin hyvin suosituilla MAX232 -piirillä. Tämän osan tarkoitus on lähinnä sarjamuotoisen datan välittämiseen tietokoneen avulla mikro-ohjaimen testauksen aikana. Käytetään yleensä vain UART/USART -testauksessa. [13]

3.3.2 Lyhyt käyttöohje STK500 alustalle

STK500-kehitysalustassa saa olla kerrallaan vain yksi mikrokontrolleri. Alustassa on monta kantaa, joten ohjekirjasta täytyy tarkistaa, mihin kantaan valittu mikrokontrolleri soveltuu. Mikrokontrollerin kannan alla on värimerkintä, joka sitten myöhemmin määrää ohjelmointikaapelin liitäntäpaikan. Mikrokontrolleri asennetaan alustaan silloin, kun alusta on katkaistu jännite.

Kun mikrokontrolleri on asennettu, voidaan tehdä tiettyjä valintoja jumppereiden avulla, voidaan vaikuttaa mm. siihen, että käytetäänkö ulkoista kello-oskillaattoria, otetaanko *RESET*-nastaa käyttöön tai käytetäänkö syöttöjännitettä (5 V) AD-muuntimen referens-

siarvona (Aref). Ohjelmointia varten täytyy asentaa ohjeiden mukaisesti 6-pinnisen laattakaapelin *ISP6PIN* ja *SPROG(1-3)* liittimien väliin. Liitin on merkitty samanvärisenä kuin mikrokontrollerin kanta, johon se on asennettu.

Jos on tarvetta, voidaan kytkeä myös 10-pinnisillä laattakaapeleilla painikkeet ja/tai ledit. Myös on mahdollista kytkeä omia ulkopuolisia piirejä ja komponentteja. Silloin vain tarvitaan laattakaapeleita.

Kun kaikki liitokset on tehty on aika siirtyä ohjelman siirtoon (olettaen siis, että ohjelma on jo valmiiksi tehty). RS-232 -kaapelin avulla yhdistetään ohjelmointia varten tarkoitettu liitin tietokoneen sarjaporttiin. Jos sarjaporttia tietokoneessa ei ole, joudutaan käyttämään esim. USB-RS-232 -sovitinta. Alustaan voidaan tässä vaiheessa kytkeä käyttöjännite. Tietokoneessa avataan AVR Studio ja käynnistetään ohjelmansiirtotyökalun. Valitaan listasta saman mikrokontrollerin, joka on asennettu alustaan, tarkistetaan tai tehdään muutoksia FUSE-asetuksiin, valitaan HEX- ja/tai EEPROM-tiedostot ja painetaan Program-painikettä, kun ollaan valmiina. Vaihtoehtoisesti voidaan tarkistaa onko kontrollerin sisältö sama kuin tietokoneessa olevan tiedoston sisältö painamalla Verify. Voidaan myös lukea kontrollerissa olevaa tietoa painamalla Read-painikettä, jonka jälkeen ohjelma kysyy, mihin ja millä nimellä tiedostoa tallennetaan. [14]

3.4 AVR Studio kehitysympäristönä

Atmel on kehittänyt myös omia ohjelmointituotekehitysympäristöjä, joista tunnetuin on Atmel AVR Studio, josta tämän työn tekemisen aikana on tullut jo kuudes versio. Atmelin AVR Studiota käytetään mikro-ohjainten ohjelmien suunnittelussa. AVR Studio on hyvin monipuolinen ja sisältä valmiiksi useita lisätyökaluja, joista yksi tarjoaa tuen mm. Atmel STK500-alustalle. Nykyisin AVR Studio tarjoaa tuen myös ARM-prosessorille ja mikro-ohjaimille. Mutta kuitenkin alun perin AVR Studio on tarkoitettu vain AVR-tuoteperhettä varten.

AVR Studio tarjoaa mahdollisuuden ohjelmoida oletuksena kahdella ohjelmointikielellä, joista toinen on GNU C-kieli ja toinen on AVR Assembler. AVR Studio on ilmainen niin kotikäyttäjälle, kuin yrityksillekin. Tämä asia selittää jo aika paljon, miksi AVR on niin suosittu. AVR Studiosta löytyy valmiiksi rakennetut mahdollisuudet koodin kääntämiselle. Kehitysympäristö siis sisältää myös valmiiksi integroidun kääntäjän molemmille kielelle. Kääntämisen lisäksi on debuggaus-toiminnot, joiden avulla pystyy tarkkailemaan

ohjelman suoritusta mikro-ohjaimen sisällä. Debuggausta käytetään virheiden etsimisessä. [15]

AVR Studio sisältää täyden tuen STK500-kehitysalustalle. Mahdollisuus on tehty lisätyökalun ansiosta. Työkalu mahdollistaa hallitsemaan niin itse alustaa, kuin myös siinä olevaa mikro-ohjainten sisältöä. Valmiiksi tehdyn ja käännetyn ohjelman voidaan ajaa mikro-ohjaimen ohjelma-muistiin tai lukea sieltä. Lisäksi voidaan hallita EEPROM- ja flash-muistien sisältöä lukemalla tai ylikirjoittamalla ne. Käytännöllinen toiminto on vertaus, jolloin voidaan verrata mikrokontrollerin sisällä olevaa dataa tiedostoon olevan datan nähden. Näin voidaan esimerkiksi varmistaa, että mikro-ohjaimessa on tietty ohjelman versio. [15]

3.5 Microsoft Visual Basic 6.0 ohjelmointikieli

Microsoft on kehittänyt 1990-lukujen alussa graafisen, BASIC -pohjalta rakennetun ympäristön, jonka avulla on mahdollista suunnitella suhteellisen helposti graafisella käyttöliittymällä varustettuja sovelluksia. Kehitysympäristö sai tuolloin nimen Visual Basic. Vuosien aikana Visual Basicista oli tehty useampi versio, kunnes ohjelmointiympäristö on siirtynyt .NET -alustaan. Viimeisin ”ei-.NET” versio on 6.0. [16]

Valinta on kohdistunut juuri tuohon Visual Basic 6.0:aan, koska se on varsin helppo ja tehokas tapa luoda sovelluksia ja soveltuu myös sarjaporttien hallintaan. Lisäksi Visual Basicin avulla luodut sovellukset ovat suhteellisen kevyitä, ainakin jos verrataan .NET -sovellusalustaan. Toimeksiantoyrityksen tuotannossa olevissa koneissa ei ole nettiyh- teyttä, eikä niihin haluta asentaa .NET -alustan ajonaikaisia tiedostoja.

Visual Basic 6.0:n avulla pystytään tekemään yksinkertaisia, helppoja ja ajonkelpoisia ohjelmia. Visual Basic 6.0 sisältää tuen tietokannoille, sen avulla pysytään lukemaan ja kirjoittamaan tiedostoja, muuttamaan Windows-järjestelmän rekisteriä, toimia API-rajapinnassa yms. [17]

Visual Basicissa on hieman poikkeava funktioiden ja yleisesti ottaen koodin rakenne, jos verrataan esimerkiksi C-kieleen. Visual Basicissa isojen ja pienien kirjaimien välillä funktioiden ja muuttujien kirjoittamisessa ei ole merkitystä. Funktioiden rakenne ja syntaksi ovat erilaisia kuin C-kielessä. Toisaalta visuaaliset ominaisuudet ovat todella helppoja ja erilaisia ikkunoita voi tehdä jopa alle minuutissa. [18]

4 TESTAUSLAITTEEN KEHITYS

General Testeria oli tarkoitus kehittää niin, että sitä olisi helpompi käyttää työntekijän ja tuotantoprosessin kannalta. Tärkeimmät ominaisuudet asiassa on tarkemmat tiedot testattavasta laitteesta ja testausprofiilin valinnat sekä profiilien muokkaaminen. Tietojen esittämistä varten riittää pieni nestekidenäyttö, jonka tarkkuus on 2×16 merkkiä, eli tarkoittaa 2 riviä, kummassakin rivissä on 16 merkkiä. Näytössä täytyy olla taustavalo, joka helpottaa tietojen lukua hämärässä olosuhteessa. Mikro-ohjaimen suurin osa nastoista on jo käytössä ja ihanteellinen lopputulos olisi sellainen, jolloin ei tarvitse muuttaa pääpiirilevyä, ei ainakaan oleellisesti. Tästä syystä on valittu LCD -näyttö sarjaporttiohjauksella.

Toinen ongelma on tässä se, että UART-liitäntä on Atmega48PA -piireillä vain yksi. [19] Tässä työssä pitäisi olla kaksi UART-yhteyttä: toinen on näyttöä varten ja toinen on RS-232 -tietokoneyhteyttä varten. Onneksi tämä on helppo ratkaista yksinkertaisesti tarkentamalla, mitä työssä halutaan ja mikä on päämäärä. Näyttöä ja tietokoneyhteyttä tuskin tarvitsee käyttää yhtä aikaa. Jos lisätään uusi mittausprofiili, käytetään tietokonetta, eikä silloin katsota laitteen näyttöä. Testauslaite on silloin ns. orja-tilassa. Silloin kun mittausprofiili on muistissa ja ollaan valmiina suorittamaan mittauksia, voidaan taas käyttää testauslaitetta normaalisti. Eli täytyy tehdä niin, että tietokoneyhteyden aikana näyttö olisi pois päältä tai siinä voi olla teksti, joka kertoo, että tietokoneyhteys on muodostettu.

Tämän opinnäytetyön yhteydessä ei julkaista alkuperäisten, eikä muokatun ohjelmistojen koodia, eikä paljasteta sen toimintaa, johtuen toimeksiantoyrityksen salassapitomenettelyistä. Tietokoneohjelmasta esitellään vain graafinen ulkoasu.

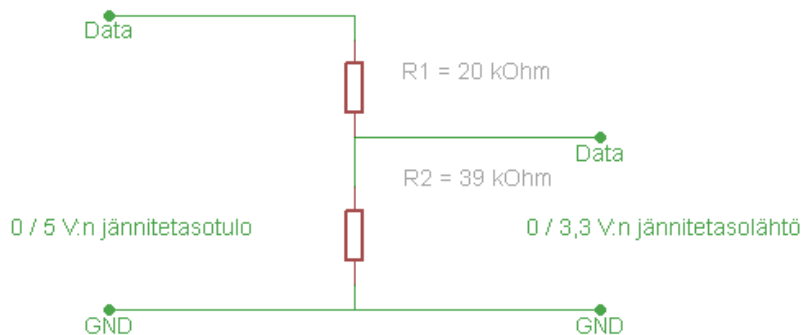
4.1 Työn alkuvaiheet ja hahmottelu

Työn alussa on selvitetty, mitä kaikkea pitäisi osata ja mitä pitäisi oppia ja harjoitella ennen varsinaisen työn aloittamista. Ennen kaikkea on tehty kuvio siitä, mitä halutaan saada aikaiseksi ja mitä kaikkea pitäisi ottaa huomioon. Lisäksi on piirretty paperille muutamia lohkokaavioita. Myös on käyty toimeksiantajayrityksessä neuvottelemassa työn edistämisestä projektinohjaajan kanssa.

Kun alkoi mikro-ohjaimen ohjelmasuunnittelu, on tehty sellaisen ratkaisu, että on kehitetty aluksi asia kerrallaan ja kun alkoi tulla valmiita koodilohkoja, niitä on yhdistetty yhteen. Näin on tehty sen takia, että voidaan rauhassa kokeilla erilaisia ratkaisuja, oppia niistä tarvittavat asiat ja tehdä päätöksiä jokaisen asian kohdalla erikseen. Aluksi on toteutettu ja koekäytetty EEPROM-muistin kirjoitus- ja lukutoiminnot, sen jälkeen UART-toiminta tietokoneen sarjaportin kautta ja lopulta LCD-moduulin toiminta. Apuna tässä oli STK500-kehitysalusta. Mikro-ohjaimena on valittu AtMega48-piirin DIL-koteloitu versio, jotta olisi helppo asettaa yllä mainittuun kehitysalustaan.

4.2 LCD-näyttö

LCD-näyttö pitää olla sarjaliitännäinen. Parhaiten sopisi LCD-näyttö, jossa on valmiina UART -liitäntä. Erästä nettikaupasta, nimeltä iteadstudio.com on löydetty UART-LCD näyttö, jonka resoluutio on 16 x 2 merkkiä. Kyseinen näyttö soveltuu tarpeeksi hyvin tähän projektiin ja sillä saisi näytettyä loppukäyttäjälle sekä useampi testausarvo (tässä tapauksessa virta-arvot), että yhteenvetotulokset tekstimuotoisina. Näyttö tilattiin suoran postikirjeenä tulemaan ja se on tullut kahdessa viikossa. Kustannukset näytölle olivat 22 USD (hinta) + 3 USD (toimituskulut postikirjeenä), eli yhteensä 25 USD. Yksi erittäin hyvä asia on ollut tässä näytössä se, että siinä on 4-pin -liitin, joten sen yhdistämiseen ei tarvita juottamista ja tilapäisesti se voidaan irrottaa testauslaitteesta (esimerkiksi testauslaitteen korjauksen tai muutoksen ajaksi). Toinen hyvä asia on se, että näytön kanssa kommunikointi tapahtuu yksinkertaisesti lähettämällä sinne merkkijonoja, jossa merkkijonon alku on käskyosa, keskiosa on käskyn parametrit ja loppupää on aina ';' -merkki, joka määrittää käskyn päättyvän. Käskykanta on yksinkertainen, mutta riittävä peruskäyttöä varten. Lisäksi käskyn suorittamisen jälkeen, LCD-näyttömoduuli lähettää takaisin kiittauksen, joka kertoo, onko käsky mennyt perille ja onnistuiko sen suorittaminen. Taulukossa 1 on LCD-näyttömoduulin käskykanta ja mahdolliset parametrit sekä esimerkki käytöstä. LCD-näyttömoduuli ei tarvitse erillistä näytönohjainta, sen voi liittää lähes mihin tahansa UART -väylään, kunhan vaan UART -väylän bittinopeus on säädetty 9 600 bit:ksi/s. Muita bittinopeuksia näyttö ei tue, eikä missään kerrottu, miten sen voi muuttaa. Koska atMega48PA tukee kyseistä nopeutta, tästä ei tule vastaavia ongelmia. UART -tulotasojen sisään- ja ulostulojännite on 3,3 V (minimi arvo on 2,7 V ja maksimi on 3,6 V). Tästä syystä jouduttiin tekemään jännitesovittimen kahdella vastuksella, jolla saatiin 5 V:n TTL -jännite muutettua 3,3 V jännitteeksi. Sovittimen kytkentä näkyy kuvassa 7.



Kuva 7. Jännitetason sovitin

Voidaan osoittaa, että jännitesovitin on suunniteltu oikein käyttäen Ohmin lakia, jolloin käytetään kaavaa:

$$U_{out} = \frac{U_{in}}{R1+R2} \times R2.$$

Koska mikrokontrolleri pystyy ohjamaan jopa 20 mA:n kuormaa, ei oteta huomioon sovittimen aiheuttamaa kuormitusta siihen. Jos sovittimeen syötetään 0 / 5 V:n tasoista datasiignaalia, ulos saadaan 0 / 3,30508 V:n tasoinen datasiignaali, joka on jo itsestään todella tarkkaa, varsinkin jos otetaan huomioon edellä mainitut sallitut rajat (2,7 V ja 3,6 V). Vastuksina on käytetty 1 % toleranssin vastuksia. Näytön *RXD*-nasta siis kytketään tämän sovittimen datalähtöön.

Näyttöä kokeiltiin ensin tietokoneen kanssa Visual Basicilla, jolloin STK500-alustan kautta oli lähetetty dataa. LCD-näyttömoduuli oli kytketty alustan *RXD* (Received Data) ja *TXD* (Transmitted Data) -pinneihin (jotka on tarkoitettu yleensä kytkettäväksi mikro-ohjaimiin) jännitesovittimen kautta ja samalla seurasin, mitä tapahtuu näytöllä. Oskilloskoopilla oli seurattu tilannetta, ettei tule ylimääräisiä häiriöitä ja jännitetasot ovat oikeita. Näyttö alkoi toimia heti. Ensimmäinen oli kokeiltu käskyä, jolla oli saatu pyyhittyä valmiiksi olevaa tekstiä näytössä ja sen jälkeen kokeiltiin lähettää näyttöön tekstiä "Toimii!". Tämäkin onnistui. Tässä vaiheessa ei ole vielä seurattu tilannetta, miten LCD-näyttömoduuli vastaa käskyihin. Seuraavaksi ruvettiin kokeilemaan näyttöä mikro-ohjaimen kanssa, joka on ollut käytössä jo muissa testeissä (ATmega48PA). Kokeilussa on käytetty edelleen STK-500 alustaa.

Kuten oli jo edellä mainittu, näyttö on sarjamuotoinen, joten sen ohjaamiseen tarvitaan sarjamuotoa tukeva käskykanta, jolloin näytölle lähetetään ensin käsky (eli mitä teh-

dään) ja sen jälkeen käskyn mahdolliset parametrit. Näytön valmistajan sivuilta löytyi käskykantalista, joka on esitetty taulukossa 1.

Taulukko 1. Näytön käskykanta

Käsky	Parametrit	Käskyn toiminnon kuvaus
sc	-	Näytön tyhjennys
sb	0/1	Tausta valo päälle / pois, 0-pois, 1-päälle
sd	X,Y	Koordinaatin asetus näytössä X=0...1;Y=0...15
ss	merkkijono	Merkkijonon lähetys, maksimipituus 16 merkkiä
su	0/1	Kursorin asetus päälle/pois; 0-pois,1-päälle
sf	0/1	Kursorin tyyppin valinta; 0-vilkkuminen päälle, 1-pois
sm	0/1	Kursorin siirto; 0-vasemmalle, 1-oikealle

Kun näyttö suorittaa käskyn, se palauttaa vastauksen, jossa kerrotaan, onko käsky suoritettu oikein vai virheellisesti. Palautus voi olla "O" (eli OK), joka tarkoittaa, että käsky on suoritettu oikein, tai "E" (eli error), joka tarkoittaa, että käskyn toteutus aiheutti virheen. Kaikki käskyt suoritetaan alle 50 ms:n aikana. [19]

4.3 Yhteyden rakentaminen tietokoneeseen

Kuten oli jo aiemmin mainittu, testauslaitteen kommunikointi tehdään UART-väylän avulla. Mutta UART-väylä käyttää TTL-logiikkatasoa, mikä tarkoittaa sitä, että tulotasot saavat olla TTL-spesifikaation mukaan 0...0,8 V ja 2,0...5,0 V, lähtötasot saavat olla vastaavasti 0...0.5 V ja 2,7...5,0 V. [20] Tietokoneyhteys UART:in kautta ei onnistu suoran tietokoneen kanssa, koska yleiset tietokoneportit eivät tue näitä tasoja natiivisti standardilla IO-porteilla. Tietokoneissa on yleisempiä digitaalisia IO-portteja ovat: USB, PS/2, LPT, COM, Firewire (IEEE1349) ja verkkoliitännät, kuten Ethernet (RJ-45) ja erilaiset modeemit. Koska tarvetta on sarjamoitoiselle siirtotielle molempiin suuntiin ja liitännöiden pitäisi olla yleisesti käytössä, järkevinä vaihtoehtoina jää lähinnä USB- ja COM -portit. Seuraavissa kappaleissa tutkitaan COM-porttien ja USB-porttien soveltuvuutta tähän työhön ja niiden tarkoitusta yleisesti.

4.3.1 RS-232 -portti

RS-232 eli COM-portti on jo hyvin vanha tietoliikenneväylä, joka on säilynyt tähänkin päivän asti ja on vielä käytössä useissa ratkaisuissa. Ensimmäinen standardi oli otettu käyttöön vuonna 1962 ja kolmas revisio oli standardisoitu vuonna 1969 (RS-232C) ja neljäs revisio vuonna 1987 (RS-232D). Viimeinen ja universaali revisio oli tehty vuonna 1991 (RS-232E), joka ei sisällä mitään uusia parannuksia, vaan takaa yhteensopivuuden vanhojen revisioiden kanssa. Yleisin liitäntä, jota käytetään COM-portin liittämisessä on DE-9, joka on lähes jokaisen pöytätietokoneen takana, jossa on vanhempi emolevy (yleensä valmistettu ennen v. 2004-2005). Joissakin tietokoneissa on jopa kaksi sellaista liitäntää, koska aiemmin (ennen v. 2000) oli vielä paljon sarjaporttilaitteita, kuten modeemit, hiiret ja muut ulkoiset laitteet. [21]

Sarjaportti on hidas isojen datamäärien tiedonsiirtoon ja nykypäivänä sitä käytetään yhä vähemmän ja vähemmän. Lisäksi se vaatii hyvin usein, että ulkoisella laitteella on oma jännitesyöttö. Tämä portti olisi muuten soveltunut tähän työhön, koska kyse on hyvin pienistä datamääristä ja laitteella on joka tapauksessa oma jännitesyöttö. Mutta toimeksiantajan mukaan kannattaa yrittää toteuttaa USB-yhteyden.

4.3.2 USB-portti

USB (eng. Universal Serial Bus) on myös sarjaliikenneväylä, joka viime aikoina on eniten suosittu sarjamuotoinen väylä tietotekniikassa. Ensimmäisen kerran USB oli julkaistu vuonna 1995. USB-laitteita on nykyään todella paljon ja USB-portteja on lähes jokaisessa pöytätietokoneessa ja kannettavassa tietokoneessa. USB-portit ovat syrjäyttäneet COM -portit viime vuosina todella tehokkaasti ja nykyään on jo paljon tietokoneen emolevyjä, joissa on jopa yli 10 USB-porttia, eikä ole yhtään COM-porttia. [22]

USB:stä on olemassa erilaisia versioita, joista tunnetaan ainakin 1,1, 2,0 ja 3,0. Niiden enimmäistiedonsiirtonopeudet ovat 12 Mbit/s, 480 Mbit/s ja 4,8 Gbit/s. USB-liitimissä on erilliset nastat jännitesyöttöä varten ja datasiirtoa varten. Esimerkiksi USB 1.1:ssä ja USB 2.0:ssa on neljä nastaa, jotka ovat: +5 V, GND, DATA+ ja DATA-. [22]

Koska tämän opinnäytetyön aihe ei ole keksitetty porttien tutkimiseen tämä tieto on vain yleisesittelyä ja portin historian esittelyä.

4.3.3 Portin valinta testauslaitetta varten

Oli annettu mahdollisuus valita itse ratkaisun, mutta toivottu tulos oli USB-portin kautta kommunikointi. Ei ollut merkitystä sillä, miten toteutetaan sen testauslaitteen sisällä ja mitä piirejä olisi käytössä, kunhan olisi edullinen ja toimiva ratkaisu. Aluksi oli käytetty COM-porttia STK500-alustan kanssa ja on saatu ohjelman toimimaan siltä osin, että yhteys onnistuu ja datansiirto toimii. Sen jälkeen piti siirtyä USB-porttiin. Oli useita ratkaisuja, kuten löytää valmiina oleva USB-muunnin COM-portiksi ja sitten MAX232-piirillä toteutettu tasomuunnin TTL-logiikkatasoiksi, mutta työpaikalla annettiin hyvä neuvo, että on olemassa USB/TTL muuntimia yhdessä mikropiirissä, joka olisi paras ratkaisu. Valitettavasti nämä piirit olivat kaikki pintaliitosmalleja. Onneksi lopulta löytyi hyvin edullinen ja valmis ratkaisu, kuten valmis sovitin, jonka avulla voidaan muuttaa USB-signaalit TTL-logiikkatasoon, joka kuitenkin näkyy tietokoneessa kuten virtuaalinen COM-portti. Tämä oli myös edullinen ratkaisu, sillä moduulin hinta on vain 10,95 e Suomessa. [23] Sovitin on hyvin kompakti, eikä tarvitse siihen juottaa mitään, moduulissa on valmiit liittimet. Sovitin näkyy kuvassa 8.



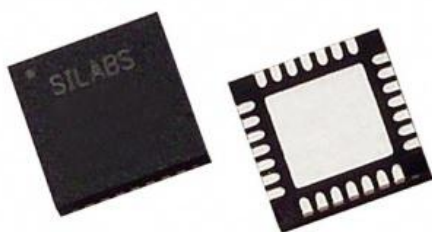
Kuva 8. USB-UART sovitin

Sovittimen koko on varsin pieni, kuten kuvasta huomataan, se on yhtä pieni kuin markkinoilla olevat USB-muistit. Koteloa tässä sovittimessa ei ole, eli pelkkä piirilevy-versio.

4.3.4 CP2102 USB-UART -muunnin

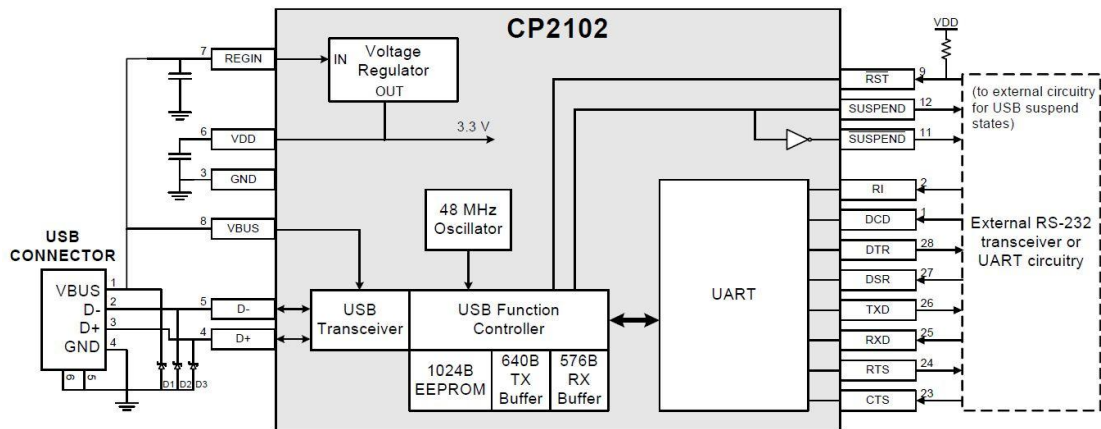
Edellisessä luvussa esitetty moduuli on siis USB/UART-sovitin, jonka kehittäjä on DFRobot-niminen yritys. Sovittimessa käytetään Silicon Labsin kehittämää piiriä, jonka nimi on CP2102. Samaa piiriä käyttävät myös monet muut USB-UART-sovittimien valmistajat, ja se on laajassa käytössä ammattilaisten sekä harrastajien laitteissa. Kysei-

nen mikropiiri vaatii toimiakseen ajureita. Ajurit kyseiseen sovittimeen löyvät Silicon Labsin sivuilta, sieltä löytyy myös piirin dokumentaatio sekä ohjelmapaketit, joiden avulla pystyy muokkaamaan ajureita ja piirin ID-tietoja halutuiksi. Tämä on tarpeellista silloin, kun kyseessä on tapaus, jolloin tämän piirin pohjalla halutaan valmistaa ”oma laite”. Kuvassa 9 on Silicon Labsin kehittämä mikropiiri, jota käytetään USB-UART sovitimissa.



Kuva 9. CP2102 -piiri [24]

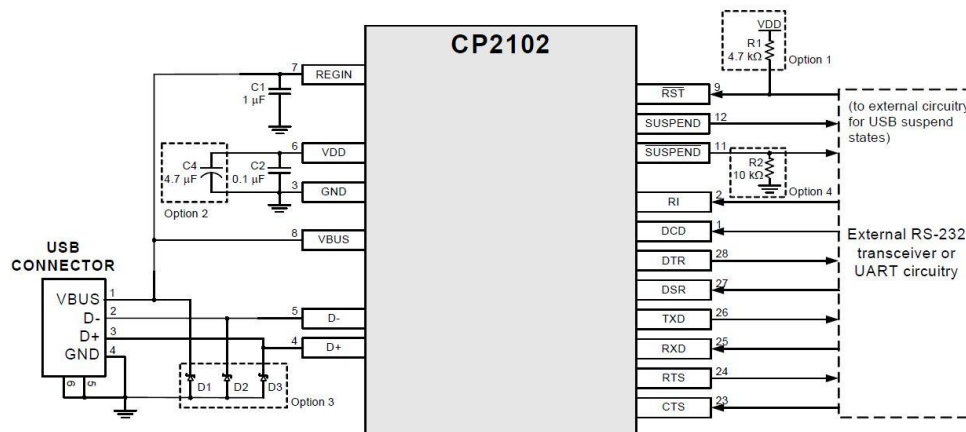
Kuten kuvasta voidaan huomata, se on QFN-28-koteloitu, SMD-ladontaa silmällä pitäen suunniteltu. On yritetty etsiä, löytyykö samasta piiristä DIL-koteloitu versio, muttei sitä ole löydetty. Jos tästä piiristä olisi DIL-koteloitu versio, sen avulla olisi voitu rakentaa kytkentä vaikka reikälevylle. Kuvassa 10 on lohko-kaaviokytkentä, josta näkee piirin sisäiset lohkot ja niiden tarkoituksen.



Kuva 10. CP2102 -piirin lohkokaavio [25]

Kuten tästä voidaan huomata, piirissä on sisäinen 3,3 V:n jänniteregulaattori, 48 MHz:n oskillaattori, USB-lähetin, sisäinen EEPROM-muisti, datapuskurit sekä itse USB-UART-muunnin. Tämän piirin avulla pystyy toteuttamaan lähes täydellisen RS-232-portin, koska löytyy kaikki tarvittavat nastat (*TXD*, *RXD*, *RTS*, *CTS*, *DTR*, *DSR*) lisäksi on myös *DCD* ja *RI*-nastat. [25]

Tämä mikropiiri vaatii todella vähäisen määrän ulkoisia komponentteja toimiakseen. Näin olleen sen kytkentä ei vie paljon piirilevytilaa. Jopa 3,3 V:n regulaattori on integroitu piiriin sisään. Kuvassa 11 on referenssikytkentä. [25]



Kuva 11. Referenssikytkentä CP2102 -mikropiiriä varten [25]

Kun oli tutkittu DFRobotin suunnittelemaa piirilevyä ja sen kytkentää, oli todettu, että se on lähes kokonaan referenssikytkennän mukainen. Ainoastaan kondensaattorit *C1* ja

$C2$ on jätetty, suojadiodeja $D1-D3$ ei ole, eikä muitakaan komponentteja. Kondensaattori $C4$ on suositeltavaa asentaa silloin, kun on tarkoitus syöttää muille osille 3,3 V:n jännitettä VDD -nastasta. Vastus $R1$ toimii ylösvetovastuksena, jos halutaan parantaa kytkennän luotettavuutta ja suojamaan kohinalta RST -nastaa. [25]

Jotta saadaan sovitin näkymään Windowsin laitehallinnassa oikealla nimellä, päädyttiin siihen, että täytyy muokata hieman piirin EEPROM-muistissa olevia nimiketietoja ja tietokoneen ajuritiedostoja. Tämä asia ei ollut välttämätön, mutta haluttu ominaisuus. Silicon Labsin Internet-sivuilta löytyi apuohjelmopaketti, jonka nimi on USB Xpress. Se sisältää mm. työkalun, jonka avulla voi muokata laitteen ID-numeroa, nimikettä ja muita tietoja. [26] ID-tunnukset on jätetty koskemattomina, mutta nimikekenttään kirjoitettu ”General Tester 1.0”-teksti. Näiden toimenpiteiden jälkeen laite on saatu näkymään Windows-käyttöjärjestelmän laitehallinnassa juuri sellaisella nimellä.

4.4 EEPROM-muisti ja profiilien säilytys

Kuten oli jo aiemmin mainittu, AtMega48PA sisältää oman EEPROM-muistin, jossa voidaan säilyttää dataa. EEPROM-muistin koko tässä mikro-ohjaimessa on 256 t, eli siihen voidaan kirjoittaa tai siitä lukea 256 kpl 8-bittistä arvoa. [18] Jokainen profiili vie 20 t tilaa. Profiilin rakenne on seuraava: 3 kanavaa, joista jokaisessa on minimi-maksimi arvot, joita voidaan mitata kahdella jännitteellä (tämä tekee $3 \times 2 \times 2$ arvoa = 12 arvoa) + testausjännitteen valinta-arvo + pulssitestaus valinta-arvo + 10 arvoa, jotka muodostavat profiilin nimen. Yhteensä nämä arvot vievät 25 t muistitilaa.

Jos jokainen arvo on 8-bittinen, tarvitaan vain yksi tavu jokaista arvoa koden. Ja jos EEPROM -muistin koko on 256 t, siihen voidaan tallentaa seuraavasti: $256 / 25 = 10,24$ kpl. EEPROM-muistiin mahtuu 10 kokonaista profiilia ja jäljelle jää vielä 6 t. Profiilin arvot tallennetaan EEPROM-muistiin peräkkäin, että niiden luku ja kirjoitus olisi helppoa (voidaan suorittaa helposti silmukka-menetelmällä lukemalla useita arvoja peräkkäin). Loput 6 t käytetään omien asetusten tallentamiseen.

4.5 Tietokoneohjelma profiilien luontia ja siirtoa varten

Testausprofiilien siirtämiseen tarvitaan tietokoneohjelmisto, jolla voidaan luoda, muokata ja poistaa testausprofiileja. Ohjelmistolla pitäisi päästä lukemaan testauslaitteessa

olevia arvoja ja siirtää uusia arvoja tilalle. Ohjelman tulisi olla suunnitelmien mukaan ergonominen, eli helppokäyttöinen ja selkeä. Kyseessä olisi ns. laitteen hallintaohjelmisto. Koska alun perin on suunniteltu niin, että käytetään RS-232-virtuaaliporttia USB:n kautta tietokoneyhteyttä varten, niin ohjelmiston pitäisi tukea datasiirtoa kyseisen portin kautta (siis RS-232). Optimaalinen ja helppo sovelluskehitysohjelmisto kyseisen ohjelman suunnittelua varten oli aiemmin mainittu Visual Basic 6. Sillä on paljon hyviä puolia: koodin tuottaminen on helppoa ja visuaalisen käyttöliittymän tekeminen vielä helpompaa. Toinen hyvä näkökanta siinä on mahdollisuus saada sillä tehtyä ohjelmaa toimimaan myös vanhemmissa käyttöjärjestelmissä ja tietokoneissa vaatimatonta kokoonpanolla.

4.5.1 Datasiirto tietokoneen ja testauslaitteen välillä

Testausprofiilin siirtoalgoritmi tietokoneesta mikro-ohjaimen:

- 1) Testauslaite sammutetaan ja kytketään tietokoneeseen USB-kaapelin avulla. Kun testauslaite sen jälkeen käynnistetään ja kun USB-UART-sovitin moduuli saa virran USB-väylästä, 10 kOhm:n vastuksen kautta positiivinen jatkuva 1-tila signaali siirtyy mikro-ohjaimen nastaan 12 (*Port C*), jolloin mikro-ohjaimessa oleva ohjelma heti käynnistyksen jälkeen tarkistaa kyseisen portin tilaa. Jos se on 1-tasossa, laite siirtyy tietokoneyhteystilaan (PC-Mode), jolloin voidaan tietokoneohjelman kautta muokata EEPROM-muistissa olevia testausprofiileita.
- 2) Käynnistetään testausprofiilien muokkausohjelmaa ja painetaan Yhdistä-painike.
- 3) Testataan, onko yhteys kunnossa. Tietokoneohjelma lähettää tietyn merkin (tässä tapauksessa "Z") ja mikro-ohjaimessa oleva ohjelma vastaa takaisin jollakin tietyllä merkillä (tässä tapauksessa "X"). Jos tietokoneohjelma on vastaanottanut merkin, yhteys on toimiva, jolloin tietokoneohjelma ilmoittaa asiasta. Jos yhteyden muodostaminen on epäonnistunut tai saatu merkki ei ole oikea, niin käyttäjä saa ilmoituksen epäonnistuneesta yhteydestä.
- 4) Jos yhteys on kunnossa, käyttäjälle avautuu mahdollisuus lukea ja muokata mikro-ohjaimen EEPROM-muistissa olevia testausprofiileita. Jos yhteyttä ei muodostettu, toimintoja ei voi käyttää ja käyttöliittymän tietyt painikkeet ja kentät poistettu käytöstä.

- 5) Lohkon lähetys tapahtuu näin: tietokoneohjelma muuttaa virta-arvot milliampeereista 0...255 arvoiseksi (eli 8-bittiseksi), jonka jälkeen tuo 8-bittinen numero muutetaan ascii-koodiksi, jolloin sitä pystyy lähettämään yhdellä merkillä. Näin ohitetaan ongelma, jos täytyy lähettää luku esimerkiksi 67, joka koostuu kahdesta merkistä 6 ja 7, joista kumpikin on 8-bittinen luku. Mutta kun muutetaan luku 67 ASCII-koodin merkiksi, saadaan arvoksi merkki C, joka on 8-bittinen. Mikro-ohjaimen kannalta sillä ei ole merkitystä, koska siinä käytetään 8-bittisiä lukuja ja "C"-merkki on sille sama kuin '67'-luku. Näin muodostetut merkit lähetetään RS-232 portin kautta (siis käytetään USB-USART-sovitinta), joka toimii USART-tilassa mikro-ohjaimen, jossa kyseinen merkki vastaanotetaan, tallennetaan EEPROM-muistiin, vastaanotettu merkki lähetetään takaisin tietokoneeseen.
- 6) Tietokoneohjelma tarkistaa, että lähetetty ja vastaanotettu arvo täsmää. Jos takaisin saatu arvo on sama, kuin lähetetty arvo, tietokoneohjelma kuittaa tapahtuman onnistuneeksi ja lähettää seuraavat arvot samalla periaatteella. Jos takaisin saatu arvo ei täsmää, tietokoneohjelma lähettää arvon uudelleen vielä kerran. Toisen yrityksen epäonnistumisen jälkeen, tietokoneohjelma ei enää yritä lähettää dataa mikro-ohjaimen, vaan näyttää käyttäjälle virheilmoituksen tapahtuman epäonnistumisesta.
- 7) Jos kaikkien arvojen lähetys onnistui, tietokoneohjelma ilmoittaa käyttäjälle, että profiilin lähetys onnistui.
- 8) Testausprofiilin siirron jälkeen käyttäjä voi siirtää toisen profiilin, lukea olemassa olevat profiilit tai poistua muokkaustilasta, jolloin mikro-ohjain alkaa suorittaa ohjelman normaalisti eteenpäin, eli laitetta ei tarvita sammuttaa profiilien muokkauksen jälkeen.

Testausprofiilin luku mikro-ohjaimesta:

- 1) Lähetetään kysely, ilmoitetaan mikro-ohjaimelle, että halutaan lukea sisältö.
- 2) Mikrokontrolleri vastaa kuittaukseen lähettämällä saman merkin takaisin.
- 3) Tietokoneohjelma lähettää numeron, joka tarkoittaa profiilin numeroa, jota halutaan lukea
- 4) Mikrokontrolleri vastaanottaa numeron ja alkaa lukea EEPROM-muistin sisältö, lähettäen jokaista arvoa yksitellen tietokoneeseen.
- 5) Tapahtuma kohdassa 4 toistetaan uudelleen.

- 6) Tietokoneohjelma vertaa ensimmäisen ja toisen vastaanotetun datan ja jos ne täsmäävät, arvot siirtyvät käyttöliittymän kenttiin
- 7) Jos toinen data eroaa ensimmäisestä, ilmoitetaan virheestä.

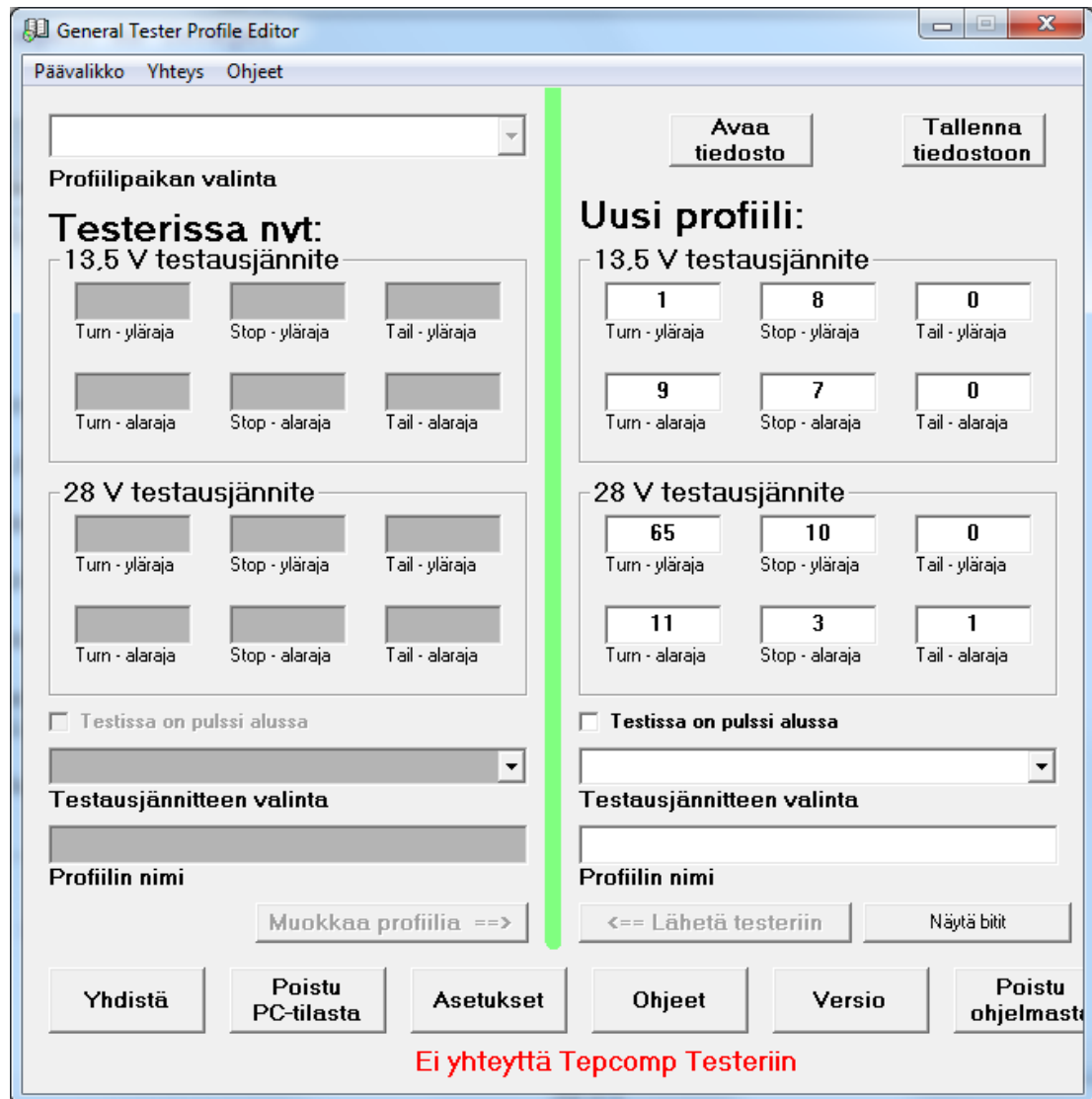
Kyseistä menetelmää ei voida pitää parhaana ratkaisuna ja onhan tässäkin pieni riski, että dataa voi mennä väärin samasta kohdasta uudestaan, joten ohjelma ei huomaa virhettä. Kuitenkin kyseinen ratkaisu oli pakko valita, koska jos mukaan haluttiin saada parempi virheen havaitseminen ohjelmallisesti, niin ohjelman koko kasvoi huomattavasti, eikä enää mahtunut mikrokontrollerin muistiin.

4.5.2 Ohjelman toimintojen ja ominaisuuksien suunnittelu

Tietokoneohjelman avulla pitäisi pystyä muokkaamaan laitteen EEPROM-muistissa olevia arvoja helposti ja ilman suurempaa vaivaa tai pitkää käytönopettelu. Myös ohjelman avulla pitäisi pystyä muokkaamaan testauslaitteen tietyjä asetuksia, tallentamaan profiilin arvoja tietokoneelle tiedostoon ja avaamaan niitä. Tietysti kannattaa ohjelmaan sijoittaa toiminnot, joiden avulla pystyy avaamaan tarvittaessa ohjetiedosto, sulkemaan yhteyden testauslaitteen kanssa ja uudestaan avaamaan sen tarvittaessa. Samalla on mietitty, kannattaako tehdä graafisessa käyttöliittymässä painikkeet vai valikot, joista toimintoja kutsutaan.

4.5.3 Ohjelman graafisen käyttöliittymän suunnittelu

Graafisen käyttöliittymän tulisi olla mahdollisimman helppokäyttöinen ja selkeä. Ohjelmassa pitää esittää yhtäaikaisesti kaikki arvot, jotka ovat testauslaitteen sisällä ja samalla myös pitää olla kenttiä, joihin kirjoitetaan uudet arvot. Ohjelmassa saisi olla erillinen Asetukset-ikkuna ja toimintoja varten painikkeet. Näiden tietojen pohjalta oli suunniteltu vaadittu käyttöliittymä, joka näkyy kuvassa 12.



Kuva 12. Tietokoneohjelman pääikkuna

Kuten kuvasta voidaan huomata, sen ikkuna on jaettu kahtia selkeällä paksulla vihreällä viivalla (Line-komponentti, jonka paksuusarvo on 10 pikseliä). Ohjelmaikkunan vasemmalla puolella on testauslaitteen sisältö ja ohjelman oikealla puolella on kentät uusia arvoja varten. Vasemmalla puolella ylhäällä on pudotuslistakomponentti (Combo-box-komponentti), jonka avulla valitaan tietty profiili testauslaitteiston muistista. Oikealla puolella on painikkeet kenttien arvojen tiedostojen avaamista ja tallennusta varten.

Vasemmalla puolella olevat komponentit pitäisi olla vain lukua varten, eli niiden kenttien arvot pitäisi olla kirjoitussuojattu. Oikealla puolella olevia arvoja täytyy pystyä muokkaamaan. Jos halutaan siirtää testausprofiili testauslaitteen EEPROM-muistiin, täytyy kirjoittaa kaikkiin kenttiin arvot ja painaa "Siirrä" -painikettä. Kun arvot siirtyneet

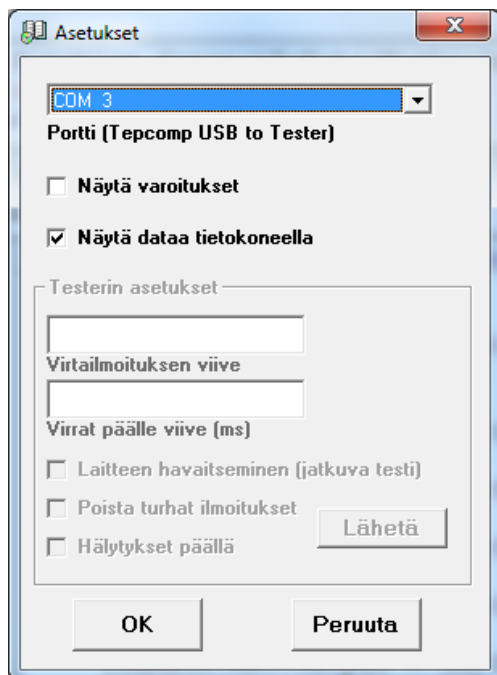
testauslaitteeseen, ne kutsutaan uudelleen vasemmalla oleviin kenttiin. Näin käyttäjä voisi itsekkin varmistaa, että testausprofiilitietojen arvot siirtyivät virheettömästi, eikä siirron aikana tapahtunut virheitä. Jos testauslaitteessa olevaa profiilia halutaan muokata siten, ettei tarvitse kirjoittaa arvoja uudelleen, vaan halutaan esim. muokata yhden tai useamman arvon, niin voidaan helposti siirtää se vasemmalta puolelta oikealle painamalla ”Muokkaa profiilia →” -painikettä. Sen jälkeen pystytään muokkaamaan arvoja ja lähettämään ne takaisin testauslaitteeseen tai tallentamaan tiedostoon.

Ohjelmaikkunan alaosassa on painikerivi, jossa on tärkeimmät toiminnot. Painikkeiden kautta pääsee mm. ohjelman asetuksiin, ohjeisiin, yhteyden avaamiseen ja ohjelman sulkemiseen.

Yleisesti ottaen, ohjelman käyttöliittymä on suhteellisen selkeä ja sujuvaa. Pientä opettelua ja tottumista se kuitenkin vaatii, joten täytyy myös laatia selkeät ja tarkat käyttöohjeet.

4.5.4 Ohjelman asetuksien suunnittelu

Ohjelmalle on tehty erillinen Asetukset-ikkuna, jonka avulla voidaan säätää niin ohjelman, kuin testauslaitteen omia asetuksia. Asetukset-ikkunan kautta voidaan säätää kuin ohjelman omia, että testauslaitteen asetuksia. Kuvassa 13 näkyy Asetukset-ikkuna.



Kuva 13. Asetukset-ikkuna

Kuten siitä voi huomata, siinä on erillinen kehys testauslaitteen asetuksille, joka aktivoituu, kun yhteys testauslaitteeseen on muodostettu. Seuraavia laitteen asetuksia pystyy muuttamaan: virtailmoituksen viive, testauksen lopuksi valojen syyttämisaika, testattavan laitteen havaitseminen, turhien ilmoitusten poisto, äänihälytysten asettaminen.

4.5.5 Ohjelman virhetilanteet ja niiden käsittely

Jokaisessa ohjelmassa pitäisi olla tietynlainen virheenkäsittely, joka suojaa tiedot käyttäjän omista virheistä sekä ilmoittaa ohjelman sisäisistä virheistä. Näin voidaan antaa käyttäjälle suuntaa-antavia tietoja, joiden perusteella hän voisi saada selvillä, mikä meni pieleen. Visual Basic 6.0 -ohjelmointikielessä on valmiiksi suunniteltu ajonaikaisen virheen havaitseminen ja ilmoitus käyttäjälle (Run-Time error). Valitettavasti ohjelma yleensä tämän jälkeen sulkeutuu, eli kaikki tiedot menetetään. Näitä virheitä voi tapahtua esim. jos ohjelma yrittää kirjoittaa tiedostoon, joka on kirjoitussuojattu tai lukea tiedosto, jota ei ole edes olemassa. Toinen tapaus on se, että käyttäjä itse voi aiheuttaa tiedon katoamisen, esim. poistumalla ohjelmasta tallentamatta käsiteltäviä tiedostoja. Silloin täytyy ohjelmoijan valmiiksi miettiä joukko lisätoimintoja, jotka varoittavat tai estävät virheiden syntymistä ilmoittamalla käyttäjälle, mitä hänen pitäisi tehdä.

Tässäkin testauslaitteen hallintaohjelmassa on otettu huomioon tietyt virheet ja lisätty joukko funktioita, jotka ilmoittavat niistä ja käsittelevät ohjelmia. Yksi esimerkki on yhteyden muodostaminen testauslaitteeseen. Jos testauslaite ei ole kytketty, on väärä porttivalinta asetuksissa tai testauslaite ei ole yhteysvalmiudessa, niin näyttöön tulee virheilmoitus, joka ilmoittaa käyttäjälle asiasta. Toimeksiantajan mukaan täydellistä virheselitystä ei tarvitse tehdä, sillä kyseessä on ohjelmisto tuotantoa varten, eikä sitä ole tarkoitus myydä.

Virheen havaitseminen on tehty varsin yksinkertaisella tavalla. Jos testauslaite ei kuitenkaan lähetettyä merkkiä, niin yhteyttä ei voida muodosta. Tätä riittää jo siihen, että käyttäjä saa viestin virheenilmoituksen yhteydessä, jossa ohjeistetaan tarkistamaan porttivalintaa, tarkistamaan testauslaitteen tilaa ja yhteyden.

4.5.6 Ohjelman käyttöohjeen laatiminen

Jokainen tietokoneohjelmisto pitäisi sisältää käyttöohjeen, jotta käyttäjän olisi helppoa ottaa ohjelmisto käyttöön ja hänelle olisi selvää, mitä hän on tekemässä. Pienet sovellukset eivät yleensä vaadi kovin tarkkoja ohjeita, joten ohjeen voidaan joskus upottaa ohjelman käyttöliittymään. Mutta kun kyse on isommasta ja tärkeämmästä sovelluksesta, tarvitaan erillinen tiedosto, josta löytyy selitykset kaikkiin toimintoihin ja ohjelman käyttöön.

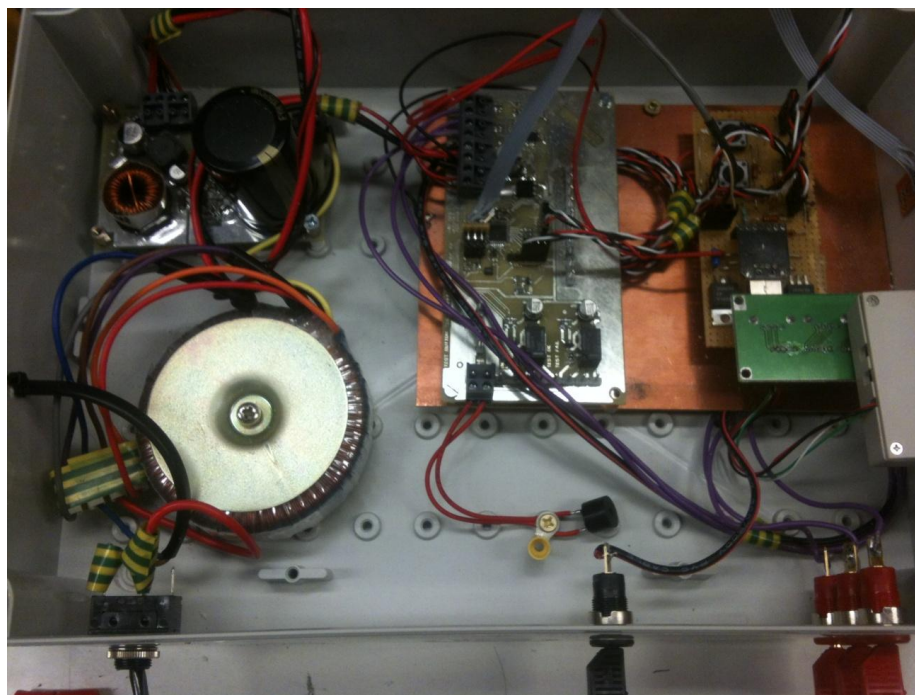
Tämän ohjelmiston mukana tulee myös erillinen ohje, joka on sähköinen pdf-tiedosto loppukäyttäjää varten. Tiedosto on myös tulostettu paperiversiota varten. Pdf-muodossa olevan ohjeen voidaan avata painamalla ohjelman pääikkunassa Ohjepainikettä. Tiedosto on oltava aina ohjelman kanssa samassa kansiossa.

5 TESTAUSLAITTEEN LOPPUKOKKOONPANO

Testauslaite lopulta koostui useasta piirilevystä, jotka olivat

- mikrokontrolleri piirilevy
- hakkurivirtalähde
- sovitinlevy
- USB-UART sovitin, joka on kiinnitetty sovitinlevyssä
- muut pienemmät komponentit.

Lisäksi testerissä oli tietysti painikkeet, joita oli kiinnitetty suoraan muoviseen runkoon ja USB-liitin, jota oli kiinnitetty pienen piirilevypalan avulla koteloon seinään. USB-liitimen tueksi on tehty muovinen kotelo, joka estää liitimen rikkoutumasta ja irtoamasta, kun siihen työnnetään USB-johto. Kuvassa 14 näkyy testauslaitteen sisältö kokoonpanovaiheessa.



Kuva 14. Testauslaitteen loppukokoonpano

Testerit oli rakennettu muovisen koteloon, jonka ulkomitat olivat 323 mm – 233 mm – 130 mm. Muovinen kotelo oli valittu siksi, että sitä on helppo käsitellä (mm. helppo po-

rata ja leikata) sekä se on hyvä eriste. Näyttö on kiinnitetty neljän ruuvien avulla yläkanteen. Oikealla kyljellä on virtakatkaisija. Virtakatkaisija sammuttaa siis laitteen kokonaan verkosta. Liitteissä 1-3 näkyy kuvat laitteen kyljestä, jossa on banaaniliittimet JIG:ejä varten, näytöstä kiinnitettynä kotelon etuosaan ja valmiiksi kootusta laitteesta.

Loppukokoonpanoon on mennyt n. 10 tuntia työaika. Lopuksi vielä oli tarkistettu laitteen turvallisuus, jossa erityistä huomiota kiinnitettiin muuntajan liitoksiin, kytkimen juotokseen eristämiseen ja sähköjohdon vedon poistoon.

6 KÄYTTÖÖNOTTO JA TESTAUS

Laite oli otettu käyttöön tuotekehitysryhmässä ja sillä on testattu erilaisia näytekappaleita, joilla oli tunnettu virrankulutus. Tässä vaiheessa laitetta ei ole kokeiltu tuotannossa, mutta sen olisi tarkoitus siirtyä sinne LED-ajoneuvovalojen lopputestauspaikkaan. Laitteeseen on tarkoitus rakentaa myöhemmin kytkeä ulkoiset sovittimet, joiden avulla voidaan kytkeä testattavat ajoneuvovalot. Kun tulee uusia tuotteita, joilla on poikkeava kytkentätapa, mutta sama toimintaperiaate, kuin nykyisin tuotannossa olevilla, täytyy tehdä vain uuden sovitiimen, eikä uutta testauslaitetta.

Laitteella oli testattu käyttöönottovaiheessa vain joitakin näytekappaleita ja prototyypppejä, joiden virrankulutusarvot oli jo ennestään tunnettuja. Virtoja oli tarkistettu kaikissa kolmessa kanavassa. Kun oli seurattu virtoja, huomattiin, että laitteen mittatarkkuus on suhteellisen hyvä, vaikka pienimmillä (< 15 mA) virroilla näkyi yli 10 %:n toleranssi. Kuitenkin tuotannossa on hyvin vähän sellaisia tuotteita, joilla olisi niin pienet virrat. Tämä ongelma korjattiin mikrokontrollerin ohjelmakoodia muuttamalla. Eli oli tehty tässä vaiheessa ohjelmistokorjaus pienimmille virroille. Ohjelmistokorjaus toimii sillä periaatteella, että kun mitattu virta on alle 20 mA mutta yli 12 mA, mittaustulos korjataan ylöspäin lisäämällä 2 mA näyttötulokseen. Kun mitattu virta on alle 12 mA mutta yli 5 mA, mittaustulos korjataan ylöspäin lisäämällä 3 mA näyttötulokseen. Kyseinen menetelytapa ei ole kovin tarkka, mutta on parempi kuin kokonaan vääristetty tulos.

Lisäksi myöhemmin voidaan sovittaa tarpeen mukaan Jyri Kallion kehittämää menetelmää tähän testauslaitteistoon virtojen korjaamista varten. Se vaatisi kuitenkin jonkin verran muutoksia myös komponenttitasolla.

Tietokoneyhteys ohjelmasta oli toiminut hyvin, tieto siirtyi testeriin virheettömästi, mutta hieman hitaasti (profiilin siirto kesti n. 5 s). Tämän takia oli optimoitu viiveitä, jonka jälkeen ohjelman tietojen siirtonopeus laitteen kanssa parani. Lopulta profiilin siirto testauslaitteeseen kesti vain 2,5 s. Ohjelman käyttöliittymää oli tarkasteltu projektin valvojan (esimiehen) kanssa ja tarkasteluun jälkeen oli saatu parannuskehotuksia. Näin piti muuttaa otsikot ja nimikkeet sopivimmaksi sekä korostaa tietyt asiat enemmän. Lisäksi oli tarkasteltu ohjeet ja muokattu hieman selkeämmäksi.

Kun laitetta oli viety tuotantoon, työntekijät eivät aluksi pitäneet laitteen koosta ja näytön sijainnista. Heillä oli käytössä tähän asti pienemmät testauslaitteet, mutta ne olivat

vaatineet ulkoisia virtalähteitä. Lisäksi testauslaitteet olivat yksinkertaisia ottaa käyttöön ja säätää virtojen rajat, mutta tässä sitten vaaditaan tietokoneohjelmien käyttöä. Useiden työntekijöiden mielestä tämä säätäminen on liian monimutkaista.

Lopulta tilanne päättyi siihen, että tämä testauslaite otettiin käyttöön loppukokoonpanossa ylimääräisenä itsenäisenä testauslaitteena LED-ajoneuvovaloja varten ja vuoden sisällä hankitaan tuotantoon LabVIEW:lla suunniteltu testausjärjestelmä, jossa käytetään ulkoisia USB-DAQ-sovittimia. Silloin olisi standardien noudattaminen ja ohjelmien muokkaus helpompaa. Lisäksi LabVIEW antaa enemmän mahdollisuuksia mitauksissa sekä ohjauksessa (pystyy säätämään ulkoisia laitteita, kommunikoimaan erilaisten protokollien avulla ja tehdä yhteenvetoja työstä).

LabVIEW on monessa suhteessa parempi ratkaisu, mutta silloin tarvitaan tietokoneita, DAQ-laitteita ja ohjelmistoja. Tämä sitten taas vaatisi jonkin verran enemmän investointeja. Toisaalta, pitemmällä aikavälillä se tulee olemaan edullisempi ratkaisu, sillä uusin testauslaitteiden rakentaminen ja uudelleensuunnittelu, kuten tämän opinnäytetyön kaltainen laite, vie enemmän aikaa ja rahaa.

7 PÄÄTELMÄT

Tässä opinnäytetyössä suunniteltiin uusi testauslaitteisto tuotantoa varten käyttäen valmiita pohjia. Laitte toimii hyvin, profiilien siirto onnistuu, asetuksien avulla voidaan hienosäätää laitteen toimintaa ja tehdä siitä miellyttävämpää. Tietokoneohjelma toimii hyvin, vaikka sen käyttöliittymä on hieman vanhanaikainen. Toisaalta ulkonäkö ei tässä tapauksessa merkitse paljon, sillä testauslaitteisto on vain tuotantoa ja tuotekehitystä varten, eikä ole tarkoitettu asiakkaalle tuotteeksi.

Taulukossa 2 on tehty vielä vertaus Jyri Kallion opinnäytetyön ja tämän opinnäytetyön välillä, josta selviää, kuinka erilaisia testauslaitteita on syntynyt samoista lähtökohdista.

Taulukko 2. Verrattavat perusteet

Verrattava peruste	Jyri Kallion opinnäytetyö [1]	Tämä opinnäytetyö
Tarkoitus	Aihiotestaus, 6 kpl aihiossa kerrallaan	Yksittäistestaus, 1 kpl irrotettuna aihiossa tai koteloituna kerrallaan
Tuloksen esittely	Korttikohtainen tulos: hyväksytty tai hylätty	Vian tarkempi määrittäminen
Tuloksen esittelytapa	LEDit	LCD -näyttö
Toimintajännite	230 VAC, sisäinen teholähde	230 VAC, sisäinen teholähde
Profiilin valintatapa	Kiertokytkin	Painikkeet
Profiilin muokkaus	Uudelleenohjelmointi	Tietokoneohjelman avulla
Testattavien laitteiden kytkentätapa	Valmiiksi integroitu testauspohja, jota ohjataan paineilmalla	Erilliset sovittimet, joita voidaan kytkeä laitteeseen banaanijohdoilla

Valitettavasti laitteen hieman suuri koko ja näytön sijoittelu keskelle epäonnistunui ergonomian suhteessa. Tietokoneohjelman käyttökään ei ole niin helppoa, että kaikki selviäisi ilman ohjeita. Kuitenkin tarkoitus ei ollut vain rakentaa testauslaitteita, vaan myös oppia uusia asioita, joita voisi käyttää muissakin tuotteissa ja testauslaitteissa. Opin-

näytetyössä on otettukin selvää AVR-mikrokontrollereista, sarjamuotoisesta datasiirrosta UART/USART-väylässä, tietokonerajapinnan ohjelmoinnista, tietokoneen sarjaporttidatasiirrosta ja sarjaliitännäisen näytön käyttöön ottamista. Kaiken lisäksi on tutustuttu paremmin Visual Basic 6.0 -ohjelmointiin, vaikka nykypäivänä tämä onkin jo tarpeeksi hyvin vanhentunut ohjelmistokehitysympäristö.

Vaikka toimeksiantaja on sitä mieltä, että laite toimii todella hyvin ja toivotut ominaisuudet löytyvät, silti olisi voitu parantaa laitetta ja luoda uuden ohjelmiston vanhan tilalle.

Yksi parannusehdotuksista on tehdä kokonaan uusi piirilevy, johon voidaan sijoittaa kaikki tarvittavat komponentit, jottei niitä tarvitse olla monessa paikassa ja useammalla levyllä, kuten nyt on. Mikrokontrolleri olisi voitu vaihtaa toiseen, esimerkiksi ATmega32:een, jossa on enemmän ohjelmamuistia, käyttömuistia, isompi EEPROM-muisti sekä enemmän IO-nastoja. Silloin olisi voitu käyttää tavallistakin, 4 bitin siirtokaistanäyttöä. Lisäksi olisi enemmän laajennusvaraa ja toimintojen määrä voisi kasvattaa huomattavasti.

Yhteyden suhteen ei ole muita erityisiä parannusehdotuksia. Valmiina olevaa sovitinadapteria USB-UART:ia ei olisi tarvittu, jos käytetään suoraan FT232 -tyyppistä mikropiiriä. Toinen idea on päästää kokonaan eroon USB-yhteydestä ja siirtyä langattomaan tekniikkaan esimerkiksi Bluetoothin tai 1-wire-protokollan avulla.

Tämä testauslaite olisi voitu myös toteuttaa täysin itsenäisenä laitteena, jossa on oma yksinkertainen näppäimistönsä ja isompi LCD-näyttö, jolloin ei olisi tarvittu tietokoneohjelmaakaan. Kuitenkin parhaaksi ratkaisuksi on paljastunut LabVIEW-ohjelmisto, DAQ-sovitin ja yksinkertainen lisäkytkentä, jolla olisi voitu paljon helpommin ja nopeammin rakentaa ammattimaisen testauslaitteen, jonka uudelleensuunnittelu ja monistaminen ei vie paljon aikaa.

8 YHTEENVETO

Tässä opinnäytetyössä on onnistuneesti rakennettu uusi testauslaite vanhan laitteen pohjalta. Työssä oli käsitelty testauslaitteen ja testaustavan toimintaperiaatetta, laitteen sulautetun ohjelman päivitystä, LCD-näytön lisäämistä, USB-yhteyden muodostamista laitteeseen USB-UART-sovittimen avulla ja graafisen hallintaohjelmiston luomista tietokonetta varten. Kehitysvaiheessa oli käytetty STK500-testausalustaa, AVR Studio -ohjelmistoa, Visual Basic 6.0 -ohjelmointiympäristöä, USB Xpress -ohjelmistoa sekä tutustuttu erilaisiin AVR-mikrokontollereihin.

Työ on osoittanut, että voidaan luoda laitteita, jotka sisältävät testausprofiileita EEPROM-muistissa ja joiden tietoa voidaan muuttaa USB-yhteyden ja tietokoneohjelman avulla. Testauslaitteeseen on tehty lukuisia parannuksia, jotka nopeuttavat testausta ja antavat tulokset selkeämmässä muodossa. Täten uusi testauslaite sisältää paremman vikakuvauksen kuin alkuperäinen testauslaite, jonka pohjalta on alkanut uuden laitteen suunnittelu. Tuotannon kannalta tämä on tärkeä asia, joka vaikuttaa tuotantoaikaan ja nopeampaan vikaetsintään.

Rajoitteina tässä laitteessa on lähinnä vain se, että laitteella pystyy testaamaan vain tiettyjä tuotteita, kuten LED-ajoneuvovaloja. Lisäksi laitteen hallintaa vaatii jonkin verran koulutusta ja opastusta. Laitteen ergonomiaan olisi kannattanut kiinnittää enemmän huomiota, koska runko on liian iso ja näytön asettelu onnistui huonosti.

Laite on otettu käyttöön tuotantotiloissa toimeksiantoyrityksessä ja laitteen toiminta on ollut erinomainen. Tämän työn tuloksia voidaan soveltaa sulautettujen testausjärjestelmien suunnittelussa.

LÄHTEET

[1] Kallio Jyri, ”LED -kortin tuotantotestin suunnittelu ja toteutus”. Opinnäytetyö. Turun Ammattikorkeakoulu. 2012.

[2] Sanoma News Oy, ”Vaihto LED-valaisimiin luo turvallisuutta ja tuo säästöjä”, [www-dokumentti]. Saatavilla: <http://www.omataloyhtio.fi/artikkelit/7708> (Luettu: 10.12.2012)

[3] Verkkokauppa.com, ”Favourlight Celesta LED Bulb - 7.9W pallolamppu E27-kantaan”, [www-dokumentti]. Saatavilla: <http://www.verkkokauppa.com/files/images/1/111719-500x500.jpeg>

[4] Wikipedia, ”LED”, [www-dokumentti]. Saatavilla: <http://fi.wikipedia.org/wiki/LED> (Luettu: 14.12.2012)

[5] Audi Finland, ”Audi 8. Tehokkuus vakiona”, [www-dokumentti]. Saatavilla: <http://www.audi.fi/brand/fi/models/a8/a8/efficiency.html> (Luettu: 7.12.2012)

[6] Valokas, ”Miksi jotkut LED-valot maksaa? Osa 2. Lämmönhallinta”, [www-dokumentti]. Saatavilla: <http://www.valokas.fi> (Luettu: 7.12.2012)

[7] Большая Энциклопедия Нефти Газа, ”Лампа - указатель – поворот”, [www-dokumentti]. Saatavilla: <http://www.ngpedia.ru/id124652p1.html> (Luettu: 3.2.2013)

[8] Wikipedia, ”Firmware”, [www-dokumentti]. Saatavilla: <http://en.wikipedia.org/wiki/Firmware> (Luettu: 3.2.2013)

[9] Wikipedia, ”AVR”, [www-dokumentti]. Saatavilla: <http://ru.wikipedia.org/wiki/AVR> (Luettu: 12.2.2013)

[10] Wikipedia, ”ATmega8 -mikrokontrolleri”, [www-dokumentti]. Saatavilla: http://upload.wikimedia.org/wikipedia/commons/thumb/a/a9/ATmega8_01_Pengo.jpg/800px-ATmega8_01_Pengo.jpg

[11] ”Торнадо Модульные Системы”. 14.01.1999 г, ”Выбор микроконтроллера”, [www-dokumentti]. Saatavilla: <http://www.gaw.ru/html.cgi/txt/publ/micros/micros.htm> (Luettu: 12.2.2013)

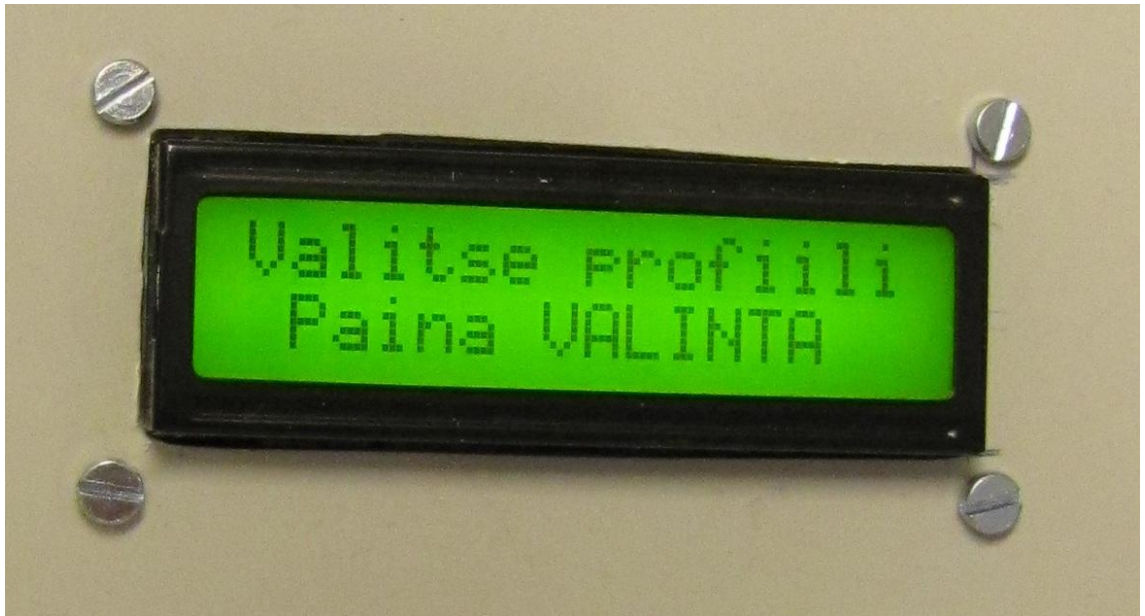
- [12] Atmel, "AVR STK-500 User Guide", [pdf-dokumentti]. Saatavilla: <http://www.atmel.com/Images/doc1925.pdf>
- [13] Maxim Integrated, "+5V, RS-232 Transceivers with 0,1uF External Capacitors", [pdf-dokumentti]. Saatavilla: <http://datasheets.maximintegrated.com/en/ds/MAX200-MAX213.pdf>
- [14] Atmel, "STK-500. Руководство пользователя", [www-dokumentti]. Saatavilla: <http://www.gaw.ru/html.cgi/txt/app/Atmel/micros/avr/stk500/5.htm> (Luettu 10.12.2012)
- [15] Atmel, "AVR Studio User Guide", [pdf-dokumentti]. Saatavilla: <https://instruct1.cit.cornell.edu/courses/ee476/AtmelStuff/doc1019.pdf>
- [16] Silicon Laboratories, "Single Chip USB to UART", [pdf-dokumentti]. Saatavilla: http://www.sparkfun.com/datasheets/IC/CP2101Rev1_5.pdf
- [17] Huru Erkki, Microsoft Press, "Ohjelmoijan käsikirja", IT Press 1999
- [18] Atmel, "ATmega48A/PA/88A/PA/168A/PA/328/P", [pdf-dokumentti]. Saatavilla: http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf
- [19] ITeadStudio, "Serial LCD Specification", [pdf-dokumentti]. Saatavilla: <http://www.vitmoi.eu/LCD.pdf>
- [20] Wikipedia, "Logic level", [www-dokumentti]. Saatavilla: http://en.wikipedia.org/wiki/Logic_level (Luettu 16.12.2012)
- [21] Wikipedia, "RS-232", [www-dokumentti]. Saatavilla: <http://en.wikipedia.org/wiki/RS-232> (Luettu 11.12.2012)
- [22] Wikipedia, "USB", [www-dokumentti]. Saatavilla: <http://fi.wikipedia.org/wiki/USB> (Luettu 11.12.2012)
- [23] DFRobot, "USB to TTL Converter", [www-dokumentti]. Saatavilla: <http://www.robotshop.com/dfrobot-usb-ttl-converto.html> (Luettu 15.12.2012)
- [24] CP2102 kuva, [www-dokumentti]. Saatavilla: <http://www.ic235.com/images/2007/2/21/SL-28-MLP.jpg>
- [25] Silicon Labs, "CP2102", [pdf-dokumentti]. Saatavilla: <https://www.silabs.com/Support%20Documents/TechnicalDocs/cp2102.pdf>

[26] Silicon Labs, "USBXpress Programmers Guide", [pdf-dokumentti]. Saatavilla: <http://www.silabs.com/Support%20Documents/TechnicalDocs/an169.pdf>

Kuva testauslaitteen kyljestä, jossa on liittimet.



Kuva laitteen näytöstä.



Kuva valmiista testauslaitteesta

