OHJAUSKORTIN SUUNNITTELU JA TOTEUTUS

Juho Vahteri

Opinnäytetyö
Huhtikuu 2011

Elektroniikka
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

Tekijä(t) VAHTERI, Juho	Julkaisun laji Opinnäytetyö	Päivämäärä 18.05.2011
	Sivumäärä 52	Julkaisun kieli Suomi
	Luottamuksellisuus ( ) saakka	Verkkojulkaisulupa myönnetty (X)
Työn nimi OHJAUSKORTIN SUUNNITTELU JA TOTEUTUS		
Koulutusohjelma Elektroniikka		
Työn ohjaaja(t) PIETIKÄINEN, Kalevi		
Toimeksiantaja(t) Mikro-Väylä Oy		

Tiivistelmä

Opinnäytetyön lähtökohtana oli suunnitella ja toteuttaa Mikro-Väylä Oy:n uusiutuvaan lainausautomaattijärjestelmään nykyistä ohjauspiirikorttia paremmin sopiva ohjauspiirikortti. Piirikortin suunnittelussa tärkeysjärjestyksessä korkeimmalla oli palautusautomaatin ohjaus, mutta myös lainausautomaatin ohjaukseen haluttiin parannuksia, kuten entistä integroidumpi tiedonkulku kortilta tietokoneelle ja oheislaitteiden ohjaus.

Päätavoitteet ja suurimmat muutokset vanhaan ohjauspiirikorttiin olivat kompaktimpi ulkomuoto ja kotelointi, selkeästi erotellut input- ja output-väylät sensoreiden tulkintaan ja moottoreiden ohjaukseen. PC:n ja mahdollisesti toisen ulkopuolisen laitteen kanssa kommunikointi RS232-väylän kautta yhtäaikaisesti ja piirikorttien mahdollisuus keskustella toistensa kanssa samassa väylässä.

Toimeksianto oli tarkkaan määritelty, mutta piirikorttien väliseen kommunikointiin soveltuvien väyläratkaisuvaihtoehtojen seulominen ja parhaan vaihtoehdon valitseminen jäivät opinnäytetyössä ratkottavaksi. Lisäksi opinnäytetyöhön kuului piirilevysuunnittelu, komponenttien valinta, sekä piirilevyn osa-alueiden testaaminen.

Opinnäytetyön asetetut asiakastoivomukset saavutettiin ja suoritettu testausosio osoitti, että ohjauskortti toimi suunnitellulla tavalla vaikkakin ohjauskortista jouduttiin tekemään toinen prototyyppi testausosion loppuun viemiseksi.

Avainsanat (asiasanat)
Atmega-mikroprosessori, CAN-väylä, I/O-väylä, lainaus- ja palautusautomaatti, RS232-väylä, RS485-väylä, piirilevysuunnittelu, tuotetestaus,

Muut tiedot

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

DESCRIPTION

Author(s)	Type of publication	Date
VAHTERI, Juho	Bachelor´s Thesis	18052011
	Pages 52	Language Finnish
	Confidential ( ) Until	Permission for web publication (X)

Title
DESIGN AND IMPLEMENTATION OF AN I/O-CARD

Degree Programme
Electronics

Tutor(s)
PIETIKÄINEN, Kalevi

Assigned by
Mikro-Väylä Ltd.

Abstract

The subject of this Bachelor's Thesis was to design and produce a new I/O-card to control Mikro-Väylä Ltd.'s new self-designed automatic book-returning system for the library environments. The new I/O-card will replace the old I/O-card which is incompetent to control bigger assemblies. In designing the new I/O-card the book-returning system was the highest priority but the new I/O-card should also handle the book-lending automatic better than the old I/O-card. Compared to the old I/O-card in book-lending environments the improvements needed were more integrated data transmission and more diverse control of peripheral devices.

Most noticeable changes compared to the old I/O-card were more compact physical size, possibility to communicate with several I/O-cards at the same bus and better electromagnetic compatibility. Distinctly separated inputs and outputs instead of using shift registers and possibility to communicate simultaneously with PC and other peripheral device via RS232 bus.

The customer requirements were specified but finding all feasible bus-solutions between I/O-cards, comparing those bus-solutions with each other and choosing the best bus-solution was decided by the researcher in this Bachelor's thesis. The other goals of the Bachelor's thesis was to design and produce the circuit board, choose the electronic components needed and test the circuit board to work as intended.

All the customer requirements set to this Bachelor's thesis by Mikro-väylä Ltd. were achieved and numerous tests showed that the new I/O-card worked as planned even though it required a little re-designing and a 2nd prototype.

Keywords
Atmega.microprocessor, automatic book-lending and -returning system, CAN-bus, I/O-bus, RS232-bus, RS485-bus, PCB designing, product testing

Miscellaneous

SISÄLTÖ

KUVIOT

SANASTO JA LYHENTEET

Bitti ***B**inary dig**it*** on binäärijärjestelmä, eli yksi bitti voi pitää sisällään joko arvon 1 tai 0. Bittien kasvaessa vaihtoehtoisten tilojen määrä kasvaa eksponentiaalisesti muodossa 2^n, jossa n = bittien määrä.

bps **B**its-**P**er-**S**econd on lyhenne sarjaliikenneväylissä käytetyistä nopeuksista. Esimerkiksi 9600 bps tarkoittaa, että bittejä liikkuu 9600 bitin sekuntinopeudella.

CAN-väylä ***C**ontroller **A**rea **N**etwork* on lähinnä teollisuudessa käytetty sarjaväylä, joka käyttää kommunikointiin kahta johdinta.

CTC **C**lear **T**imer on **C**ompare on Atmelin käyttämä keskeytysominaisuus, joka laukaisee keskeyttäjän kun joku mikroprosessorin sisäinen kello saavuttaa ennalta määritetyn arvon.

Datasheet Komponenttien ja mikropiirien piirilevysuunnittelun kannalta oleelliset tiedot sisältävä manuaali.

Debug Koodauksessa käytetty termi, joka mahdollistaa pysäyttää koodin tiettyyn kohtaan ja etenemään koodissa rivi kerrallaan. Helpottaa loogisten virheiden etsimistä.

Decal Kuva tietystä komponentista piirilevysuunnittelun pohjapiirustuspuolella. Kertoo piirilevysuunnittelijalle ja -valmistajalle komponentin fyysisen koon.

EMC ***E**lectro**m**agnetic **C**ompatibility*:lla tarkoitetaan tietyn elektronisen komponentin tai piirilevyn elektromagneettisen häiriön sietokykyä tai sitä, kuinka paljon se lähettää elektromagneettista häiriötä.

Header Ajuritiedosto koodausohjelmissa, joka määrittelee tiettyjä funktioita, joita voi käyttää, kun header on lisätty koodiin. Helpottaa ja nopeuttaa koodaamista.

IIC-väylä ***I**nter-**I**ntegrated **C**ircuit* on nimensä mukaisesti lähinnä piirikortin sisäiseen liikenteeseen tarkoitettu kaksisuuntainen tiedonsiirtoväylä.

Input Käytetään puhuttaessa mikroprosessorille päin tulevasta liikenteestä, eli toisen elektronisen komponentin lähettämä jännitteenmuutos, jonka mikroprosessori kykenee tulkitsemaan etukäteen asetettujen arvojen perusteella.

I/O-väylä Yleinen nimitys väylälle, joka kykenee reagoimaan jännitteenmuutokseen tai tuottamaan jännitteenmuutoksen. Toiminto määritetään erikseen ohjelmallisesti. Mikroprosessoreilla valtaosa pinneistä on I/O-pinnejä.

JTAG ***J**oint **T**est **A**ction **G**roup* on mikroprosessorien ohjelmointiin ja testaukseen tarkoitettu väylä.

LSU110 Mikro-Väylä Oy:n nykyinen palautus- ja lainausautomaattien ohjauskortti, jonka tässä opinnäytetyössä valmistettavan kortin tulisi korvata.

Master Laite, joka ohjaa samassa väylässä olevia muita laitteita ja pystyy kuuntelemaan niitä protokollasta riippuen.

Multi-master

On väyläprotokolla, joka mahdollistaa useamman masterin samaan väylään.

Node Nimitystä käytetään samassa väylässä olevista laitteista. Node tarkoittaa solmukohtaa, eli yksi laite jossain kohden väylää on node.

Output Käytetään, kun puhutaan mikroprosessorilta ulospäin lähtevästä signaalista. Yleensä ohjataan toista elektronista komponenttia.

RFID **R**adio **F**requency **Id**entification on radiotaajuuksilla toimiva tiedon etälukumenetelmä. Kirjastoissa käytettävät RFID-tunnisteet ovat passiivisia antenneita joista RFID-lukijat pystyvät lukemaan tarvittavat tiedot.

RS232 ***R****ecommended* ***S****tandard* ***232*** on asynkroninen sarjaliikenneväylä kahden RS-232 –laitteen väliseen kommunikointiin. Toimii yhdellä signaalilla molempiin suuntiin toisin kuin synkroninen sarjaliikenneväylä.

RS485 ***R****ecommended* ***S****tandard* ***485*** on differentiaalipohjainen kaksi- tai yksisuuntainen sarjaväylä, johon voi liittää enimmillään 256 väylälaitetta kommunikoimaan keskenään kolmi- tai viisijohtimisessa väylässä.

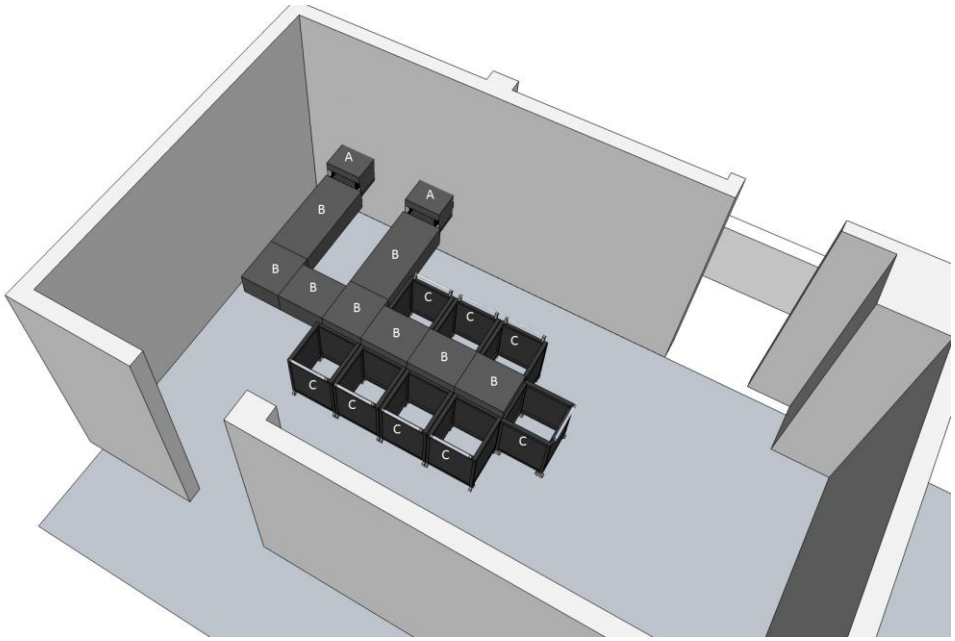
Slave Laite, joka saa käskynsä masterilta, mutta pystyy myös lähettämään pyydettäessä masterille informaatiota.

SPI-väylä Eli ***S****erial* ***P****eripheral* ***I****nterface* on synkroninen Motorolan standardoima kaksisuuntainen tiedonsiirtoväylä.

USART ***U****niversal* ***S****ynchronous/****A****synchronous* ***R****eceiver-****T****ransmitter* on tarkoitettu sarjaliikenneväylien rinnakkais- ja sarjadatan käsittelyyn. Löytyy yleensä integroituna mikropiireistä.

1 LÄHTÖASETELMA JA ASIAKASVAATIMUKSET

Mikro-Väylä Oy on Suomen suurimpia kirjastoympäristöihin tarkoitettujen lainaus- ja palautusautomaattien valmistaja ja myyjä. Sillä on yli 250 asiakasta Suomessa ja Virossa. Kirjastojen kasvava kiinnostus suurempia palautusautomaatteja kohtaan on tuonut esiin nykyisen palautusautomaatteja ohjaavan logiikkakortin LSU100:n ongelmat. Palautusautomaatin toiminta perustuu pitkään hihnaan, joka rakentuu moottoreilla varustetuista moduuleista, jotka ohjaavat kirjoja pitkin palautusautomaatin linjastoa siirtäen kirjan sen RFID-tunnisteen tai viivakoodin tietojen perusteella ennalta määrättyyn kärryyn. Asiakkaat syöttävät kirjoja palautusautomaattiin asiakassyöttöpäistä (ks. kuvio 1). Nykyinen palautusautomaatteja ohjaava kortti pystyy ohjaamaan enintään kuutta lajittelumoduulia eli jakamaan kirjoja 13:een eri kärryyn. Tämän kokoisista palautusautomaateista on tullut jo kyselyitä kirjastoilta ja odotettavissa on, että automaattien koot yhä kasvavat.



KUVIO 1. Palautusautomaatti. A: Asiakassyöttöpää, B: Ohjausmoduuli, C: palautuskärry

1.1 Havaitut ja mahdollisesti tulevat ongelmat

Suurempien automaattien myötä kirjojen asiakassyöttöpäät ovat lisääntyneet ja palautusvolyymi on kasvanut, mikä on nostanut esiin ruuhkautumisesta syntyneen ongelman. LSU110 hoitaa liikenteen syöttöpään etupäässä olevan tietokoneen kanssa, käsittelee kirjan etenemistä indikoivien antureiden tiedon ja ohjaa kirjaa kuljettavia hihnoja siirtorekistereiden avulla. Tämä tarkoittaa, että palautussektoreiden, eli asiakassyöttöpäiden, ja palautusmäärän kasvaessa ohjauskortille tuleva tiedon määrä kasvaa niin suureksi, että ohjainkortin prosessori ei ehdi suorittaa annettuja käskyjä, tai informaatiota saattaa jäädä huomioimatta.

Tulevaisuudessa palautusautomaattien automaation lisääntyessä tarve eri tapahtumia seuraaville antureille kasvaa ja näin ollen pelkästään yhden moduulin seurantaan voidaan tarvita useampia antureita. Mikro-Väylä Oy:llä on jo suunnittelutasolla kaavailtu lajittelukärryn täyttymistä indikoivaa anturia, joka kärryn täyttyessä ilmoittaa kirjaston tiskille tyhjennystä vaativasta kärrystä. Lajittelukärryjä on yhtä moduulia kohden joko yksi, kaksi tai kolme, joiden täyttymistä seuraavat anturit vaatisivat jokainen oman input-väylänsä. LSU110-ohjainkortissa input-väyliä on kuitenkin vain kuusi kappaletta, mikä rajaa yhdessä moduulissa käytettävien antureiden määrän yhteen. Tällä hetkellä jokaisella moduulilla on jo yksi anturi joka seuraa kirjan etenemistä linjastolla ja kertoo LSU110-ohjainkortille, mitä moottoreita pyöritetään ja minne suuntaan.

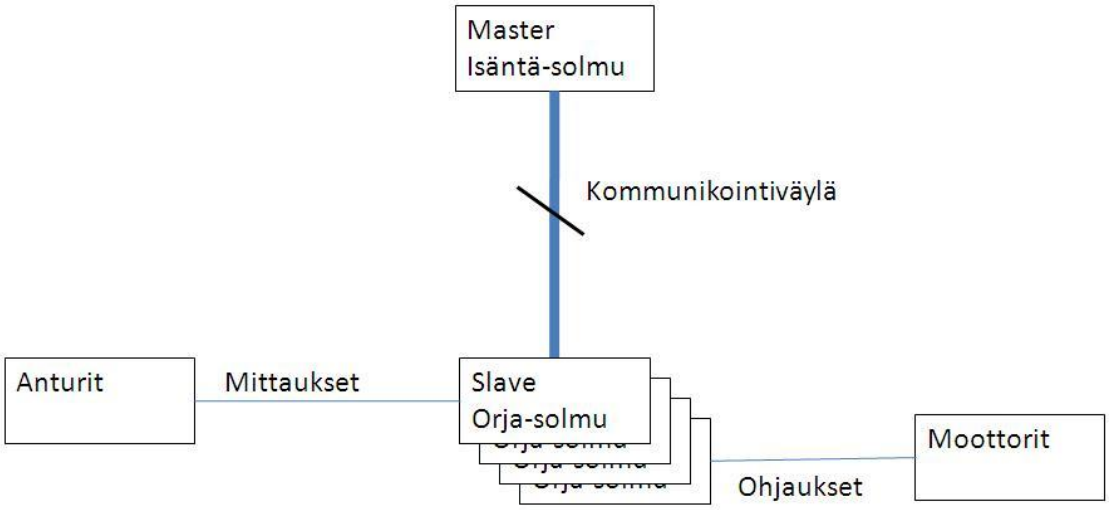
Palautusautomaatit rakentuvat yhdestä kuuteen ohjausmoduulista joiden jokaisen ohjaukseen tarvitaan neljä ohjauspinniä, jotta moduulin hihnoja voidaan pyörittää eteen, taakse, vasemmalle sekä oikealle. Kuuden moottorin ohjaukseen tarvitaan siis 24 ohjaus-pinniä. LSU110-ohjauskortissa kuuden moottorin ohjaus yhdellä kortilla on ratkaistu laittamalla SPI-väylään kuusi siirtorekisteriä, jotka ohjaavat moottoreita mikroprosessorilta lähtevien ohjauskäskyjen mukaan. Teoreettisella tasolla ja pääosin käytännössäkin ratkaisu on järkevä ja mahdollistaa suurten kokonaisuuksien ohjauksen tuhlaamatta mikroprosessorin I/O-pinnejä. Siirtorekisteri on kuitenkin aiheuttanut ongelmia palautusautomaateissa heikon häiriönsiedon takia. Palautusautomaatin jokaisessa moduulissa käytetään kahta sähkömoottoria ja kahta solenoidia jotka ovat kortin läheisyydessä ja näyttävät muuttuvilla magneettikentillään sekoittavan siirtorekistereille lähetettävää signaalia. Häiriöistä vääristynyt ohjaussignaali saattaa sekoittaa

siirtorekisterin ja aiheuttaa epäloogista toimintaa tai siirtorekisterin hajoamisen. Yksi epälooginen komento siirtorekisterissä voi aiheuttaa kirjan joutumisen väärään paikkaan, ohjelman sekoamisen tai muun vastaavan vikatilan.

Lainausautomaatissa ohjelman sekoamista tai epäloogisuutta ei ole ilmennyt, mikä selkeästi kielii palautusautomaatin ongelmien johtuvan häiriöllisestä ympäristöstä. LSU110-ohjainkortti ei silti ole täysin optimaalinen lainausautomaatinkaan ohjaukseen. Nykytilanteessa sekä kortinlukija, LSU110-ohjainkortti että numeronäppäimistö syöttävät tiedon lainausautomaatin PC:lle joko omaa RS232-väylää tai USB-väylää käyttäen. Kolmen erillisen portin hallinta PC:llä pyörivällä Mikro-Väylä Oy:n omalla käyttöliittymällä tuo turhaa monimutkaisuutta ohjelmarakenteeseen, vaatii PC:ltä suuren määrän liitäntöjä ja lisää automaatin kustannusta.

1.2 Halutut muutokset

Opinnäytetyössä suunniteltavan uuden I/O-ohjauskortin haluttiin yksinkertaistavan ja jämäköittävän palautusautomaatin toimintaa laittamalla yksi ohjainkortti ohjaamaan jokaista lajittelumoduulia (ks. kuvio 2). Tämä ratkaisu vähentää huomattavasti prosessorissa käsiteltävän tiedon määrää ja mahdollistaa lisääntyneiden input-väylien ansiosta useamman palautusprosessia seuraavan anturin käyttämistä. Liukuhihnojen ohjauksessa ongelmia aiheuttavasta siirtorekisteristä halutaan päästä myös eroon. Uudessa I/O-ohjauskortissa suunniteltiin moottoreiden ohjauksessa käytettävän elektronisia releitä joita ohjataan yksittäisillä mikroprosessorin output-pinneillä. Tämä mahdollistaa selkeämmän ja toimintavarmemman ohjaustavan, koska linja ei ole yhtä herkkä häiriösignaaleille kuin siirtorekisteri. Jotta uusilla korteilla pystyttäisiin ohjaamaan selkeästi suurempia palautusautomaatteja, täytyy korttien väliseen yhteyteen löytää riittävän vakaa väyläprotokolla, johon pystyy liittämään tarvittavan määrän kortteja. Varteenotettavia väyläprotokollia löytyy useita, ja opinnäytetyön ensimmäinen tehtävä oli vertailla eri väylien ominaisuuksia keskenään ja valita parhaiten tarkoitusperää tukeva väyläratkaisu.



KUVIO 2. Kytkentä- ja väylätopologia

Palautusautomaatin lisäksi uuteen ohjauskorttiin haluttiin muutoksia myös lainausautomaattien puolelle. Lainausautomaatin tiedonsiirtoa ja toimintoja haluttiin integroida ja yksinkertaistaa kortinlukijan ja numeronäppäimistön osalta sekä ohjainkortin mahdollisuuksia haluttiin paremmin hyödyntää. Uuteen ohjauskorttiin haluttiin kaksi sarjaliikenneväylää, jotta sarjaliikennettä käyttävän kortinlukijan pystyy liittämään suoraan uuteen ohjauskorttiin. Näin ohjauskortti pystyy ohjaamaan kortinlukijaa lainausautomaatin PC:n antamien käskyjen mukaan, käsittelemään kortinlukijalta lähtevän tiedon ja lähettämään tarpeellisen osan yhtä linjaa pitkin takaisin PC:lle. Numeronäppäimistö on normaali matriisinäppäimistö, jonka ohjaus onnistuu kahdeksalla input-väylällä. Kahdeksan input-väylää oli vaatimuksena jo palautusautomaatin suhteen, joten numeronäppäimistön ohjaus ei lisämuutoksia kaivannut.

2 VÄYLÄPROTOKOLLIEN ESITTELY JA VALINTA

Mikroprosessorien väliseen keskusteluun ja ohjaamiseen löytyy monta eri käyttötarkoituksiin löytyvää väyläprotokollaa. Jotta useat eri vaihtoehdot pystyttiin rajaamaan lähempää tarkastelua varten, oli alussa käytettävä karkeampaa rajaamista (Hirsjärvi, S. Remes, P. & Sajavaara, P. 1997, 75-77). Karkeampaan rajaamiseen käytettiin tiedossa jo olevia vaatimuksia kuten vähimmäistiedonsiirtonopeus väylien välillä, tiedonsiirtoväylään kytkettävien laitteiden vähimmäismäärä, tiedonsiirtoväylän mahdollinen fyysinen enimmäispituus ja tarvittava häiriönsietokyky. Linjat kulkevat solenoidien ja sähkömoottoreiden läheisyydessä.

2.1 Väylät

Näitä rajauksia noudattaen ja yleisimmät väyläprotokollat läpi käyneenä päädyttiin ottamaan lähempään tarkasteluun CAN-, RS485-, SPI- sekä IIC-väylät. Kaikki väylät mahtuivat karkean rajauksen sisään, mutta eroja löytyi niin nopeudesta, tiedonsiirron varmuudesta, häiriönsiedosta kuin linjan maksimipituudestakin.

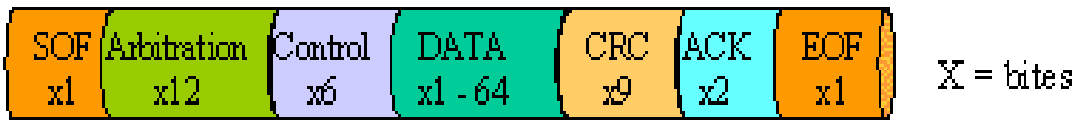
2.1.1 CAN-väylä

Controller-area network eli CAN-väylä kehitettiin elektroniikkavalmistaja Boschin toimesta alun perin autoteollisuuteen yksinkertaistamaan ja varmentamaan jatkuvasti lisääntyneiden antureiden ja sähkölaitteiden välistä kommunikointia. Hyvä häiriön sieto ja jopa yhden kilometrin pituiset väylävedot teki CAN-väylästä suositun väyläratkaisun teollisuusautomaatiossa, työkoneissa ja sairaalaympäristössä. (CAN history 2011).

CAN-väylässä tiedonsiirto tapahtuu kierretyllä parikaapelilla, jossa high- ja low-johtojen jännitetaso määrää, luetaanko arvo ykkösenä vai nollana. CAN-väylän vähimmäistiedonsiirronnopeudeksi on määritelty 20 kilobittiä sekunnissa, jolloin väylän pituudet voivat olla enintään 1000 metriä ja enimmäistiedonsiirtonopeudeksi 10 megabittiä sekunnissa, jolloin väylän pituudet voivat olla enintään 40 metriä. (CAN Bus 2006).

CAN-väylän etuja on, että jokainen väylän node kykenee lähettämään viestin ja kuuntelemaan jokaisen ketjussa olevan muun noden viestin, eli kyseessä on niin sanottu multi-master-väylä. Kahden noden yhtäaikaisesta viestin lähettämisestä syntyvä törmäys on estetty viestin alussa esiintyvällä sovittelu-osiolla, joka määrää viestin prioriteetin. Matalamman prioriteetin viestin lähettänyt node näin ollen odottaa, kunnes korkeamman prioriteetin viestin lähettänyt node on saanut viestin lähetettyä. Jokaisella viestillä täytyy olla oma prioriteettiarvo, koska saman prioriteettiarvon omaavat viestit yhtä aikaa lähetettyinä jatkaisivat viestinsä loppuun yhtä aikaa aiheuttaen virheviestejä linjaan. Sovittelukoodin osuus viestistä on 12 bittiä, joten siihen siis mahtuu 4096 viestiä eri prioriteettiarvoilla (ks. kuvio 3).

Erinäköisten ongelmien välttämiseksi CAN-väylässä kulkevat viestit on tiukasti jaoteltu tiettyyn formaattiin (CAN Bus Description 2011). Yksi viesti muodostuu 32-95 bitistä. SOF eli start of frame on merkki viestin aloitukselle. Arbitration eli sovittelu määrää viestin prioriteetin. Control eli ohjauskenttä määrää, mille nodelle viesti on tarkoitettu. DATA eli tieto pitää sisällään itse informaation, mitä viestissä halutaan lähettää. CRC eli Cyclic Redundancy Code on tarkistuskoodi, jolla pystytään tarkistamaan, että haluttu informaatio meni oikein perille. ACK eli acknowledgement error checks -osiolla hyväksytään tarkistuskoodin oikeellisuus. EOF eli end of frame lopettaa viestin. (ks. Can dictionary 2008).



KUVIO 3. CAN-väylän viestirakenne (www.interface.com 2010). X tarkoittaa bittien määrää.

CAN-väylällä varustettuja mikroprosessoreja on vähänlaisesti tarjolla, mutta CAN-väylään löytyy omia ulkoisia kontrollereita, jotka keskustelevat mikroprosessorin SPI-väylässä.

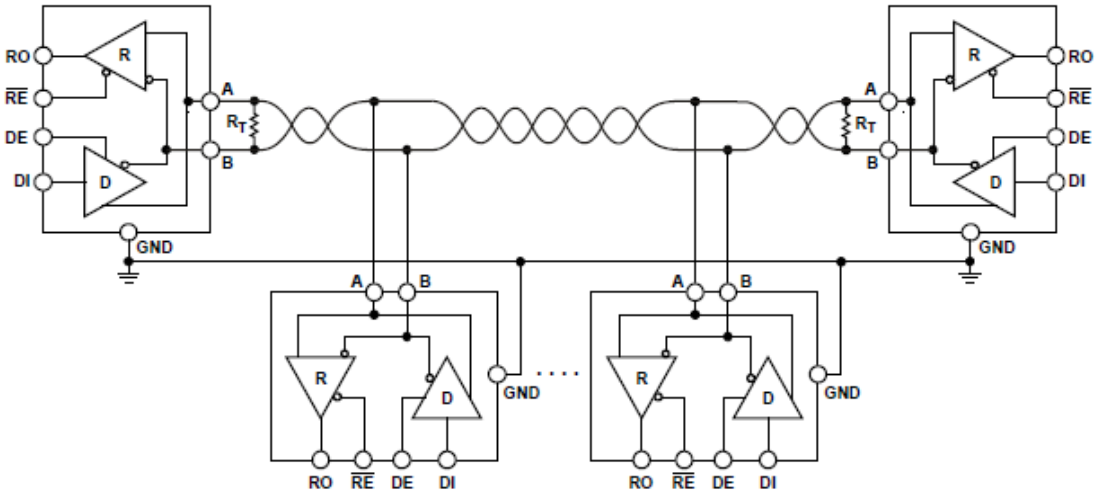
2.1.2 RS485-väylä

Recommended Standard 485 -väylä on Telecommunications Industry Association/Electronic Industries Alliancen (TIA/EIA) määrittelemä sarjaliikenneväylä. CAN-väylän tyyliin se on myös käytössä monissa automaatiosovelluksissa ja tehdasalueilla hyvän häiriönsiedon ja CAN-väylää kevyemmän protokollan ansiosta(ks. RS-485 Interface). Vaikka RS485-väylää käytetään yleisesti kaupallisissa tuotteissa, varsinaisia nopeuksia, formaattia ja protokollaa ei ole erikseen määritelty TIA/EIA:n toimesta. RS485-väylälle löytyy tosin useita kolmannen osapuolen protokollia, joista tunnetuimmat lienevät Modbus ja Profibus protokollat.

RS485 käyttää tiedon siirtoon CAN-väylän tyyliin yhtä kierrettyä parikaapelia, jos kyseessä on half-duplex eli yksisuuntainen väylä. Full-duplex eli kaksisuuntainen väylä käyttää kahta kierrettyä parikaapelia. Tämän lisäksi RS485-väylä vaatii yhteisen referenssi-maan, joten johtojen määrä on suuntaisuudesta riippuen kolme tai viisi. Nodien enimmäismäärä samassa väylässä on yleisesti ottaen 32, mutta Modbus-protokollassa on 8-bittinen osoiterekisteri, mikä teoriassa mahdollistaa 254 nodea samassa väylässä. (RS-422 and RS-485 Application Note 2006). 32 nodea on tosin opinnäytetyötä silmällä pitäen riittävä.

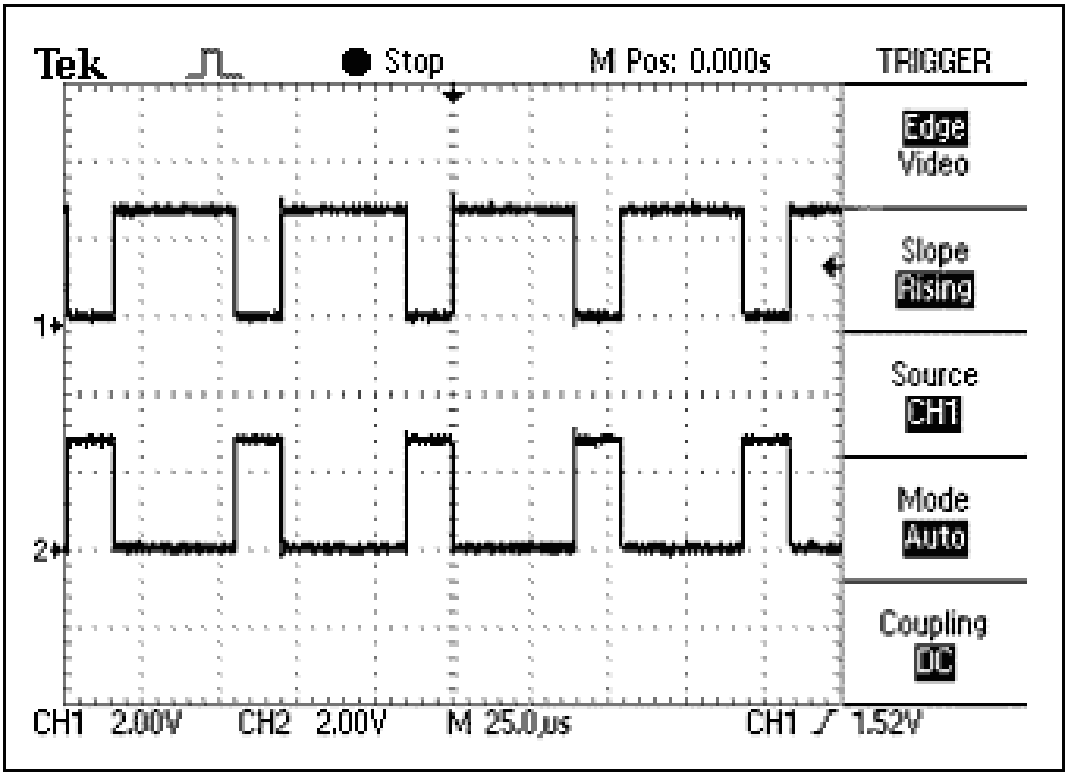
RS485-väylän tiedonsiirron enimmäisnopeus voittaa CAN-väylän kolminkertaisesti nousemalla 32 megabittiin sekunnissa ja väylän enimmäispituus on 1200 metriä 100 kilobitin sekuntivauhdilla. Yleisesti hyväksyttynä sääntönä pidetään, että nopeus bitteinä sekunnissa kerrottuna linjan pituus metreinä ei saisi olla yli 10^8. (EIA-422/485 Bus Description 2011). Esimerkiksi 100 metrin linjan maksiminopeus saisi olla enintään 1 Megabitti sekunnissa, koska 100 (m) * 10^6 (bit/s) = 10^8.

RS485:n väylätopologiaksi suositellaan rinnankytkentää tähti- tai kimppukytkentöjen sijaan. Rinnankytkennässä väylä täytyy terminoida kummastakin päästä laittamalla 120 Ω vastus ensimmäisen ja viimeisen noden lähtö- ja tulolinjan väliin (ks. kuvio 4). Tällä tavoin estetään tehokkaasti viestien mahdolliset heijastussignaalit. Tämän takia RS485-väylässä suositellaankin käytettävän rinnankytkentää, koska vain niin voidaan varmasti terminoida molemmat päät ja estää takaisin heijastuneet jännitteet. (RS-422 and RS-485 Application Note 2006).



KUVIO 4. Oikeaoppinen väylätopologia (www.neural-eng.com 2011), Rt = terminointivastus

Vaikka RS485 käyttää samalla tavalla parikaapelia kuin CAN-väylä, tiedon kulku vaihtelee suuresti. RS485-väylä käyttää differentiaalista tiedonsiirtoa, mikä tarkoittaa, että parikaapelin piuhoihin syötetään peilikuva toisen jännitteestä (ks. kuvio 5). Jännitetaso indikoi joko ykköstä tai nollaa sillä periaatteella, kumpaan syötetään negatiivinen ja kumpaan positiivinen jännite. Häiriönsiedoltaan differentiaalinen tiedonsiirto on erinomainen siitä syystä, että johtuvat häiriöt näkyvät kummassakin signaalissa samansuuntaisina. Tämä tarkoittaa, että kun signaalia vastaanottava node lisää negatiivisen jännitteen vastaluvun toisen johtimen positiiviseen jännitteeseen, alkuperäiset informaatiosignaalit muuttuvat peilikuvasta samansuuntaisiksi ja samansuuntaiset häiriösignaalit muuttuvat peilikuviksi ja kumoavat toisensa. Siksi RS485-väylässä ei haittaa, vaikka johtuva häiriösignaali olisi suurempi kuin itse informaatiosignaali, koska häiriöt kumoutuvat joka tilanteessa pois.



KUVIO 5. RS485-väylän parikaapelissa kulkevat signaalit (Explanation of Maxim RS-485 Features 2000)

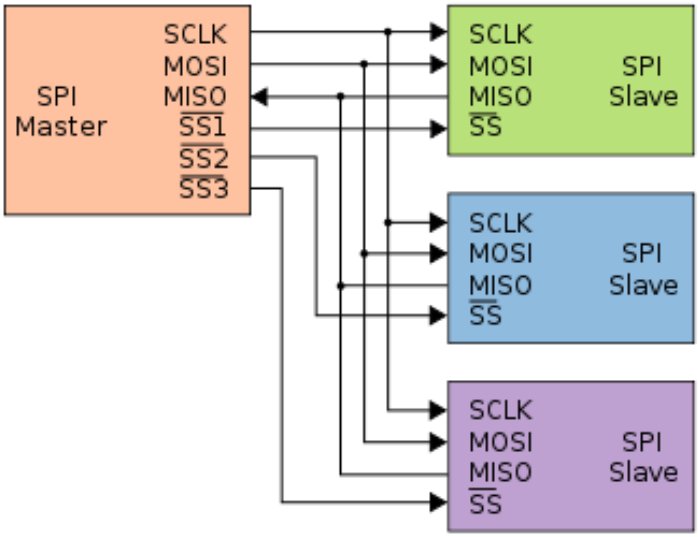
RS485-väylässä käytettävä kaksisuuntainen eli full-duplex –tiedonsiirto mahdollistaa, että useampi laite samassa väylässä kykenee esittämään kyselyitä yhtä aikaa. Tämä tosin saattaa aiheuttaa törmäyksiä, koska RS485-protokollassa ei käytetä CAN-protokollasta tuttua ”sovittelu-osoitetta”, mikä määrää viestin prioriteetin. Viestien törmäykset väylässä aiheuttavat viestien sotkeutumista tai estävät perille pääsyn kokonaan. RS485-väylä voi olla siis protokollasta riippuen joko multi-master tai pelkkä master-slave. (RS-422 and RS-485 Application Note 2006).

2.1.3 SPI-väylä

Serial Peripheral Interface -väylä on Motorolan standardoima synkroninen kaksisuuntainen väylästandardi, joka on ensisijaisesti suunniteltu piirikortin sisäiseen liikenteeseen, mutta voidaan tarvittaessa tuoda lyhyillä vedoilla piirikortilta toiselle. Sitä käytetään usein antureiden, kauko-ohjauksen, Flash- ja EEPROM-muistien, reaaliaikakello-

jen, lcd-näyttöjen ja kortinlukijoiden kommunikointiin mikroprosessorin kanssa. (Understanding the SPI Bus with NI LabVIEW 2011).

SPI-väylä käyttää yleensä neljää johtoa kommunikointiin, mutta jos päälaitteella on vain yksi slave-laite tai jos päälaitteen ei tarvitse erikseen valita, kenelle slavelle haluaa viestin lähettää, voi SPI-väylää käyttää myös kolmella johdolla (ks. Introduction to Serial Peripheral Interface 2002). Tiedonsiirtoon tarvitaan siis vähintään SCK eli Serial Clock, joka synkronoi väylässä olevat muut laitteet päälaitteen kellotaajuudelle(ks. kuvio 6). SDI eli serial data in ja SDO eli serial data out -piuhoja käytetään SPI-väylän muodostamiseen. SDI-piuha lähtee masterista slaveen ja SDO taas slavesta seuraavan slaven SDI-porttiin ja niin edelleen. Näiden lisäksi voi halutessaan käyttää CS eli chip select -piuhaa, jolla masteri voi valita, mille slavelle haluttu viesti menee. (ks. SPI Interface Specification 2005).



KUVIO 6. SPI-väylän kytkentätopologia (www.wikimedia.org 2011)

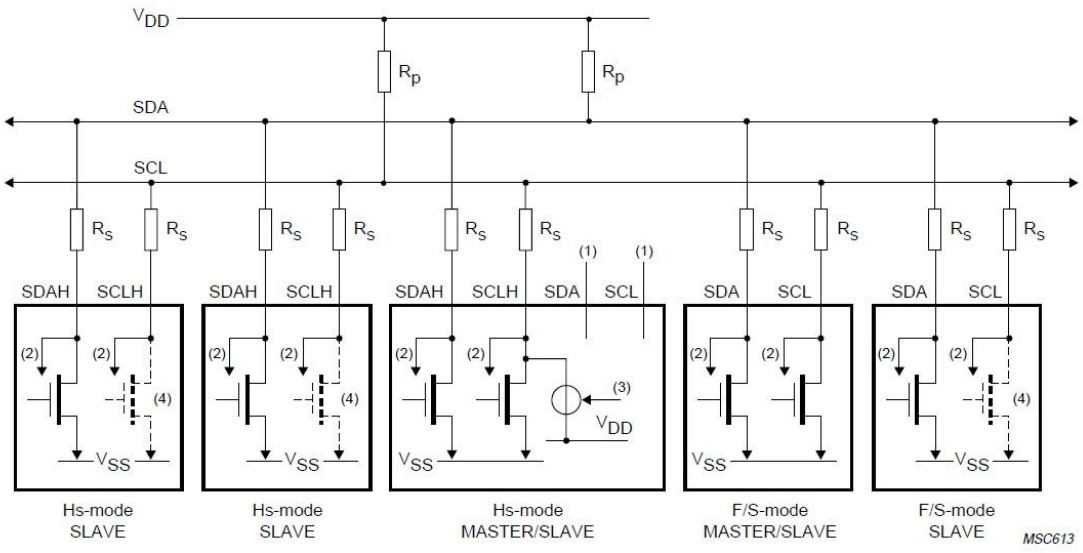
SPI-väylän suurimpia etuja on, että master-laite jakaa väylään yhteisen kellotaajuuden, eli väylässä olevissa laitteissa ei tarvitse olla mikroprosessoria tai omaa kellopulssia. Tämä tarkoittaa, että väylässä voi olla esimerkiksi pelkkä anturipiiri tai ad/da-muunnin ilman omaa kellopulssia, mikä tekee ratkaisusta halvan ja yksinkertaisen. SPI-väylän huonoja puolia on mahdollisuus vain yhteen master-laitteeseen ja väylän

pituus on erittäin rajoittunut. Tämän lisäksi häiriön sieto on kovin vaatimaton. SPI-väylään ei löydy myöskään mitään protokollaa, joka tukisi minkäänlaista virhetarkistusta. Viestin oikeellisuudesta ei siis voi saada varmaa taetta. Enimmäisnopeudet ovat myös varsin rajoittuneita ja asia huononee entisestään vetoja pidentämällä.

2.1.4 IIC-väylä

Inter-Integrated Circuit -väylä on Philipsin standardoima väyläprotokolla, joka nimensä mukaisesti, ja SPI-väylän tapaan, on piirikortin sisäiseen liikenteeseen tarkoitettu standardi. Käyttökohteet ovat myös hyvin lähellä SPI-väylää, mutta viestin virheentarkistus, priorisoidut viestit ja multi-master -ominaisuus tekee IIC-väylästä huomattavasti ISP-väylää monipuolisemman. (ks. I^2C-bus specification and user manual 2007).

IIC-väylässä tiedonsiirto tapahtuu kahden johdon välityksellä. SDA eli serial data line -piuha välittää viestin seuraavalle laitteelle, ja SCL eli serial clock -piuha välittää yhteisen kellopulssin väylään ja synkronoi viestit (ks. kuvio 7). Tämän lisäksi sekä SDA että SCL vaatii ylösvetovastukset.



KUVIO 7. IIC-väylän topologia (www.esacademy.com 2011)

IIC-väylä tukee CAN-väylän tyylistä viestikonstruktiota, jossa master ilmoittaa ensin viestin alkavan (Start), sen jälkeen viestissä ilmoitetaan arbitration-arvo eli viestin prioriteetti, joka määrää, kenen viesti saa jatkua, jos useampi master lähettää viestin yhtä aikaa. Viestin prioriteetin jälkeen viestijonossa ilmoitetaan slaven osoite, kenelle viesti on tarkoitettu. Tämän jälkeen määritellään kirjoitus- tai lukutila, annetaan tarkastusluku ja syötetään itse viesti, mikä halutaan slavelle toimittaa. Tämän jälkeen tulee vielä toinen tarkistusluku, jonka jälkeen viesti lopetetaan P (Stop)-signaalilla. (I^2C-bus specification and user manual 2007).

2.2 Väylävertailu ja valinta

Positiivisia ja negatiivisia puolia miettiessä on otettu huomioon vain opinnäytetyön kannalta positiiviset tai negatiiviset asiat, joten vertailu ei tapahdu siis yleisellä tasolla. IIC-väylän ero SPI-väylään oli niin pieni, että opinnäytetyötä ajatellen SPI-väylän hyvät ja huonot puolet ovat täysin valideja IIC-väylän suhteen. Väylävertailussa SPI-väylästä puhuttaessa kaikki argumentit pätevät myös IIC-väylän kanssa, ellei toisin erikseen mainita.

SPI-väylän tarkempi tutkiminen osoitti varsin nopeasti, että väyläratkaisussa on monia heikkoja puolia opinnäytetyötä silmällä pitäen. Master-slave-väyläprotokolla esimerkiksi tarkoittaisi sitä, että jos palautusautomaattiin tulee kaksi tai yli syöttöpäätä, täytyisi niiltä lähettää informaatio SPI-väylän master-kortille eri väylää pitkin ja tämän jälkeen master-kortti jakaisi tarvittavan tiedon SPI-väylää pitkin slave-nodeille. IIC-väylän multi-master-väyläprotokolla ei vaikuta tilanteeseen, koska matka etuhihnalta masterille ylittää väylän enimmäispituuden. Kahden väyläratkaisun käyttö saattaisi vielä jossain määrin toimia, mutta sitten saavutaan toiseen rajoittavaan tekijään eli väyläratkaisun fyysiseen pituuteen. SPI-väylää ei suositella yli puoli metriä pitkiin vetoihin, ja väylän pituusrajakin tulee vastaan jo reilussa 1,5 metrissä. Palautusautomaatissa kortit sijaitsevat yli puolen metrin päässä toisistaan, mikä tarkoittaa, että yksi SPI-väylä kuluisi pelkästään kahden laitteen väliseen matkaan. Tämänkin pystyisi vielä kiertämään käyttämällä toistopiiriä, joka lähettää saamansa informaation eteenpäin tai tekemällä ensimmäisen SPI-väylän slave-kortille uuden SPI-väylän, joka näin ollen muuttuisi master-kortiksi ja sitä seuraavasta kortista tulisi slave ja niin edelleen. Näillä menetelmillä ketjusta voisi saada jossain määrin toimivan, jos ympäristö olisi

täysin häiriötön. Palautusautomaatin ympäristö on tosin solenoidien ja sähkömoottorien takia varsin häiriöaltis alue, ja automaatin toiminnan kannalta on äärimmäisen tärkeää häiriötä sietävä tiedonkulku. Myös viestin tarkistusosio, joka ISP-protokollasta puuttuu, on välttämätön.

IIC-väyläprotokollasta tarkistusviestit löytyvät, mutta miinuspuolet ovat niin suuria, että ISP-/ IIC-väylä ei tule kysymykseenkään opinnäytetyön suhteen. Tästä syystä väylien hyvien puolien vertaileminen enää RS485- ja CAN-väylän kanssa on melko irrelevanttia. Jäljelle jääneet RS485- ja CAN-väylä olivat toki jo opinnäytetyön aloitusvaiheessa selkeät ennakkosuosikit, mutta ISP- ja IIC-väylät oli hyvä huomioida vertailussa ja sulkea ne selkeästi pois vaihtoehdoista.

RS485 ja CAN ovat päällisin puolin ja toiminnaltaan hyvin lähellä toisiaan. Väylien häiriön sieto, tiedonsiirtonopeus ja maksimipituus ovat kummassakin täysin riittävät palautusautomaatin tarpeisiin, joten eroja on lähdettävä etsimään eri suunnasta. RS485-väylän etuna voidaan todeta protokollan kevyempi ja helpompi toteutus ohjelmatasolla ja se, että vanhassa logiikkakortissa hyödynnetään myös RS485-väylää lähinnä useamman syöttöpään palautusautomaateissa. Tämä voidaan laskea varsin suureksi eduksi, koska siirtyminen kokonaan RS485-väylään olisi näin ollen pienempi muutos, koska protokolla ja ympäristö ovat entuudestaan tuttuja.

CAN-väylän suurimpia etuja on viestitopologiasta löytyvä priorisointijärjestelmä. Multi-master-tilanteessa RS485-väylässä voi protokollasta riippumatta tapahtua törmäyksiä, koska viestien tai masterien prioriteettia ei voida määrätä. Törmäys on kierrettävissä vielä nykyisissä palautusautomaateissa, mutta tulevaisuuden eri konfiguraatioissa tilanne voi olla toinen. SPI-väylää hyödyntävä CAN-controlleri on kevyempi suunnitella rautatasolla mikä kasvattaa valittavien mikroprosessorien valikoimaa. RS485-väylän tilanteessa täytyy turvautua prosessoreihin, mitkä on varustettu kahdella USART-väylällä, koska RS485-väylä käyttää kommunikointiin USART-väylää ja palautuspään kortit tarvitsevat samalla myös RS232-väylää kommunikoidakseen palautuspään pc:n kanssa. Kahden USART-väylän prosessoreja löytyy tosin Atmelilta varsin kiitettävästi ja sopivilla ominaisuuksilla tähän käyttökohteeseen, eikä tilanne muuttuisi CAN-väylää käytettäessäkään, koska kahta USART-väylää tullaan tarvitse-

maan lainausautomaattien kanssa joka tapauksessa. Näin ollen todettiin, että rauta-puolella ei eroja väylille tule, joskin CAN-controller on hivenen halvempi toteuttaa piirilevytasolla ja väyläratkaisu vaatii muutenkin vähemmän komponentteja.

CAN- ja RS485-väylien tasaisuus kääntyi lopulta RS485-väylän puolelle, koska Mikro-Väylä Oy on käyttänyt RS485-väylää aikaisemmissa sovelluksissaan. Ominaisuuksien valossa CAN-väylä ei noussut palautusautomaatin osalta juurikaan RS485-väylän yläpuolelle, joten oli Mikro-Väylä Oy:n kannalta viisaampaa pysyä valmiiksi tutussa ympäristössä. Mahdolliset viestien törmäykset ja niiden aiheuttamien ongelmien suuruus palautusautomaateissa on vielä tässä vaihein kuitenkin hankala ennustaa. Tästä syystä muutaman kappaleen prototyyppisarjaan päätettiin laittaa paikat CAN-väylälle, jotta väylää pystyttäisiin testaamaan ja tutkimaan, jos RS485-väylän huomataan aiheuttavan ongelmia.

3 KOMPONENTTIVALINNAT

3.1 Mikroprosessori

Spesifikaatioiden selvittyä pystyttiin aloittamaan piirilevyn prototyypin suunnittelu ja komponenttien valinta. Aluksi aloitettiin tärkeimmästä, eli mikroprosessorista. Vaatimukset mikroprosessorin suhteen olivat riittävä I/O-väylien määrä, kaksi USART-liitäntää RS232- ja RS485-väylille ja riittävästi keskusmuistia ohjelmalle. I/O-väylien tarve nousi lopulta 34:n, mutta osaa väylistä käytettiin vain palautusautomaateissa ja osaa lainausautomaateissa. Tästä syystä muutaman väylän pystyi yhdistämään, jolloin väylän valinta tehdään fyysisellä kytkimellä riippuen siitä, tuleeko kortti palautus- vai lainausautomaattiin. Kortin kompakti koko oli pidettävä mielessä, joten olisi ollut turhaa muuttaa prosessorikanta 44TQFP-kotelosta 64TQFP-koteloon ylimääräisten I/O-väylien takia.

Mikroprosessoria rajattaessa aluksi valmistajien mukaan käytettiin rajauskriteerinä kahta USART-väylää, koska se rajasi eniten vaihtoehtoja. Freescale Semiconductor ei tarjonnut yhtään 8-bittistä mikroprosessoria USART-väylällä ja Texas Instrument on lopettanut kokonaan 8-bittisten mikroprosessorien valmistuksen. Jäljelle jäivät Microchipin PIC-tuoteperhe ja Atmelin Atmega-tuoteperhe. Kummaltakin valmistajalta löytyi 44TQFP-kotelossa mikroprosessoreja, jotka täyttivät kriteerit niin muistin, nopeuden kuin sarjaliikenneväylienkin suhteen. Mikro-Väylä Oy on käyttänyt aiemmissa ohjauskorteissaan pääosin juuri Microchipin mikroprosessoreja, joten olisi sinänsä luonteva valinta jatkaa samassa kehitysympäristössä. Toiveena kuitenkin oli, että Microchipin vanhentuneesta kääntäjästä päästäisiin vähitellen eroon sen epäloogisuuksien ja epävakauden vuoksi. Tämän takia Atmelin Atmega nousi Microchipin edelle. Atmelin etuja on myös vahva harrastajakunta ja aktiivinen foorumi, mistä löytää tietoa ja opastusta, jos suurempia ongelmia ilmestyy. Mikro-väylältä löytyy jo valmiiksi myös Atmelin ohjelmointialusta debug-ominaisuudella, mikä helpottaa testausta ja nopeuttaa ongelmakohtien löytämistä koodista.

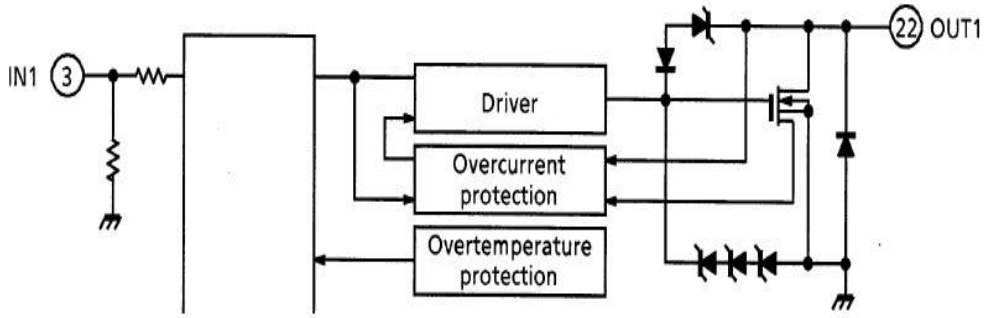
Mikroprosessorin valmistajan selvittyä tehtiin vertailua vielä Atmega-tuoteperheiden kriteerit täyttävien mikroprosessorien välillä. Hinnan, kotelotyypin, I/O-porttien, Flash-, eeprom- sekä sram-muistin osalta ATmega324-mikroprosessori vastasi parhaiten haettuja ominaisuuksia. Kaikki kriteerit täsmäsivät ja 32kb ohjelmointimuisti vai-

kutti riittävältä. Jos muisti kuitenkin jossain vaiheessa käy riittämättömäksi, niin Atmelilta löytyy samalla kotelotyypillä ja pinnikonfiguraatiolla myös ATmega644 ja ATmega1284, joiden ohjelmointimuistit ovat 64kb ja 128kb.

3.2 Mikropiirit ja muut komponentit

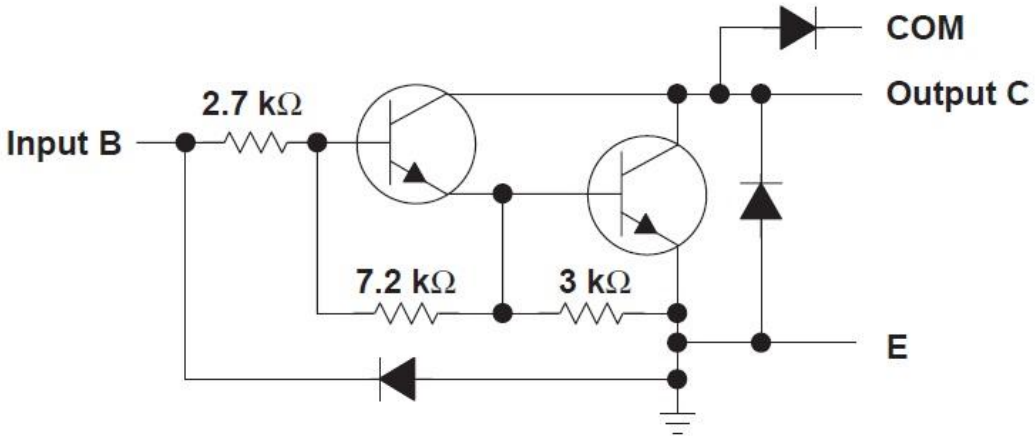
RS232-väylien ohjaukseen valittiin kaksikanavainen Maxim Integratedin valmistama MAX202-piiri, joita on käytetty Mikro-Väylä Oy:n muissa ohjainkorteissa. RS485-väylän ohjaukseen tuli kaksi Maxim Integratedin MAX485-piiriä. RS485-väylä päätettiin erottaa muun piirilevyn käyttöjännitteestä optoerottimilla häiriön siedon parantamiseksi. Kuuden linjan erottamiseen ei löytynyt kuusikanavaista optoerotinta, joten siihen valittiin Avago Technologiesin valmistamat neli- ja kaksikanavaiset optoerottimet. RS485-väylän maat ja 5 voltin käyttöjännite erotettiin Perelin dc-dc-muuntimella.

Output-väylä suunniteltiin LSU110-kortin tavoin maihin vetäväksi, jotta moottoriohjauksen lisäksi myös RGB-ledit saataisiin toimimaan vanhan kortin tapaan. Spesifikaatioina käytettiin aikaisemmin output-ohjauksessa käytetyn siirtorekisteripiirin vastaavia tietoja. 500 milliampeerin ja 50 voltin kanavakohtaisen virran- sekä jännitteenkestävyyden ja sisäänrakennetun virta- sekä ylikuumenemissuojan takia output-linjan ohjaukseen valittiin Toshiban kahdeksankanavaisen TPD2007F-low-side-MOSFET. TPD2007-MOSFET:n toimintaperiaate on vetää output-linjaan kytketty kelluva signaali maihin, kun TPD2007-MOSFET:n input-linjaan syötetään mikroprosessorilta viiden voltin jännite (ks. kuvio 8).



KUVIO 8. TPD2007 MOSFET:n yksittäisen kanavan rakenne(TPD2007F datasheet 2006)

Input-väylään haluttiin LSU110-kortin toimintaperiaatteella suunniteltu Input-ratkaisu, jossa mikroprosessorin input-väylä on ylösvetovastuksilla tilassa yksi, kun Input-väylään ei syötetä jännitettä. Input-liittimien ja mikroprosessorin input-väylien välissä on kahdeksankanavainen Texas Instrumentin valmistama ULN2803-NPN-transistori. Yhteen tai useampaan Input-liittimeen syötettäessä 3,5 voltin - 30 voltin jännite, transistori vetää mikroprosessorin ylösvedetyn input-pinnin maihin, muuttaen tilan nollaksi(ks. kuvio 9). Tätä ratkaisua kutsutaan nolla-aktiiviseksi input-väyläksi.



KUVIO 9. ULN2803-transistorin yksittäisen kanavan rakenne(ULN2803A datasheet 2006)

CAN-väylän kohtalo oli vielä prototyyppivaiheessa hämärän peitossa, joten se päätettiin ottaa vielä mukaan mahdollisia tulevia käyttökohteita ajatellen. Itse CAN-väylä on integroitu moneen mikroprosessoriin valmiiksi, mutta koska väylän tarpeesta ei ollut mitään takeita, päätettiin väylä toteuttaa ulkoisilla piireillä. Microchipin MCP2515 CAN-ohjain ja MCP2551 CAN-lähetinvastaanotin sopivat käyttötarkoitukseen hyvin halvan hinnan ja pienen koteloratkaisun puolesta.

4 PIIRILEVYSUUNNITTELU

Komponenttien valinnan jälkeen prototyypin suunnittelu jatkuu piirilevysuunnittelulla, jonka voi jakaa kahteen osioon. Lohkokaaviopuolen suunnittelulla määritellään levyn kytkentätopologia ja varsinainen toimintamalli luomalla suunniteltavan piirilevyn jokaisesta komponentista lohkokaaviokomponentti ja yhdistämällä komponentit haluttuun kytkentään. Lohkokaaviopuolella määritellyt kytkennät ja raja-arvot pitävät huolen pohjapiirustuspuolelle siirryttäessä, että vikakytkentöjä ei synny ja näin ollen pohjapiirustuspuolella pystytään täysin keskittymään linjojen ideaalisiin reitteihin ja komponenttisijoitteluun. (ks. Kärnä 2009a). Piirilevyn suunnitteluun on useita erilaisia ohjelmia, mutta tämän levyn suunnitteluun käytettiin työpaikalta löytyvää Mentor Graphicsin PADS-ohjelmaa, joka on tällä hetkellä yleisin piirilevysuunnitteluohjelma maailmalla.

4.1 Lohkokaaviosuunnittelu

Piirilevyn suunnittelu alkaa aina lohkokaaviosuunnittelusta ja vaikka opinnäytetyössä käytiin ensiksi komponenttivalinnat lävitse, niin usein lohkokaaviosuunnittelu aloitetaan samaan aikaan ja komponenttien valinta etenee piirikortin loogisen kytkentätopologian hahmottuessa. Lohkokaaviopuoli vaatii tarkkaa suunnittelua ja huolellisuutta, koska mikropiireistä täytyy lohkokaaviokomponentit tehdä itse mikropiirien datasheetin mukaan ja tämän jälkeen yhdistää komponenttien pinnit vastaamaan suunniteltua kytkentää. Prototyyppiä suunniteltaessa kaikki peruskomponentit löytyivät valmiina kirjastoista. Mikroprosessori ja valtaosa mikropiireistä jäävät usein piirilevysuunnittelijan itse tehtäviksi, kuten tässäkin opinnäytetyössä. Lohkokaaviokomponenttien jälkeen lohkokaaviosuunnittelu jatkui kytkentäkaavion suunnittelulla. Kytkentäkaaviota rakentaessa on hyvä jakaa eri osa-alueet osiin ja rakentaa erillään toisistaan, minkä jälkeen yhdistää osa-alueet yhdeksi kokonaisuudeksi. Opinnäytetyössä erotellut osa-alueet olivat I/O-väylä, virransyöttö, CAN-väylä, RS232/RS485-väylä sekä viimeisenä mikroprosessori ja siihen liittyvät komponentit.

4.1.1 I/O-väylän suunnittelu

I/O-väylän rakentaminen oli I/O-väylän yksinkertaisen ja aiemmista piirilevysuunnitteluista tutun kytkentätopologian ansiosta varsin helppo osa-alue lohkokaaviosuunnittelussa. Output-väylässä käytettiin kahdeksankanavaista TPD2007-low-side-MOSFET:a. MOSFET ohjaa output-liittimeen liitetyn vedon maihin, jonka ohjauspuoli kytkettiin mikroprosessoriin ja väliin laitettiin testauspisteet. vedettävältä puolelta vedin vastaavat vedot suoraan output-liittimeen. Väliin laitoin led-valon jokaiseen väylään liittimen ja TPD2007:n välille indikoimaan linjan toimimista. Led-valot asetettiin TPD2007:n maihin vedettävälle puolelle, että samalla pystytään helposti toteamaan TPD2007:n toiminta ilman oskilloskooppia. Input-väylän suunnittelu vastasi hyvin paljon output-väylää sillä erotuksella, että nyt väylän toimivuutta indikoivat ledit olivat ylösvetovastuksella, ULN2803-NPN-transistorin ja mikroprosessorille menevien testipisteiden välissä. Näin voidaan todeta samalla transistorin toiminta ja led-valot saadaan suojattua ulkoisilta häiriöiltä. I/O-väylän suunnittelu oli sitä myöten valmis ja väylien veto mikroprosessorille tulee kysymykseen lohkokaaviosuunnittelun mikroprosessori-osiossa.

4.1.2 Virransyötön suunnittelu

Käyttöjännitteenä on suunniteltu käytettävän samaa joko +12 voltin tai +24 voltin DC-muuntajaa. Piirilevyllä kaikki mikropiirit ja itse mikroprosessori käyttää +5 voltin jännitettä ja liittimistä saa haluttaessa sisään syötettävän +12 volttia tai +24 volttia. Sama käyttöjännite kaikkialla yksinkertaisti virransyötön suunnittelua, koska asiasta selvittiin yhdellä lineaarisella regulaattorilla. National semiconductorin LM340T-regulaattori antaa ulostulosta +5 voltin jännitteen ja maksimissaan 1 ampeerin virran. Regulaattorin sisääntulojännitteenä voi käyttää 7,5 volttia - 35 volttia, joten ongelmaa ei tule, vaikka käyttäisi +12 voltin tai +24 voltin muuntajaa. Regulaattorin sisääntulon puolelle sijoitettiin vielä 220 mikrofaradin tasauskondensaattorin ja 100 nanofaradin suotokondensaattorin ja ulostulopuolelle yksi 100 nanofaradin suotokondensaattori, minkä jälkeen virransyöttö oli valmis.

4.1.3 CAN-väylän suunnittelu

CAN-väylä oli myös suunnittelun puolesta varsin yksinkertainen, koska se sisältää vain CAN-controllerin, joka keskustelee mikropiirin SPI-väylässä ja CAN-lähetinvastaanottimen, joka muuntaa tulevan ja lähtevän signaalin oikeaan muotoon. Yhdistettyä CAN-controllerin CAN-lähetinvastaanottimeen ja CAN-vastaanottimen CAN-liittimeen täytyi linjojen välille lisätä paikka vielä väylän terminointivastukselle ja suotokondensaattorit CAN-piireille.

4.1.4 RS232/RS485-väylän suunnittelu

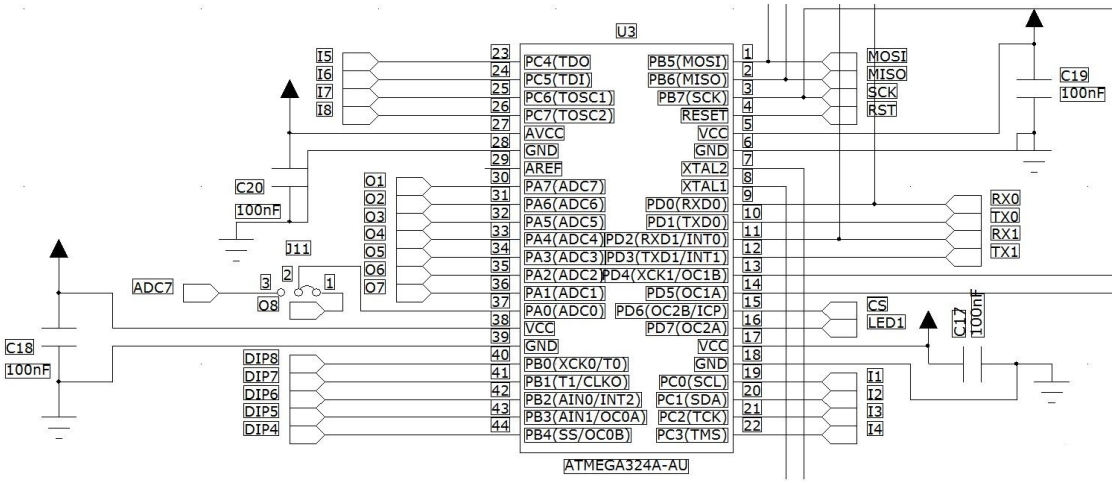
Sarjaliikenneväylien suunnittelu oli konstruktioltaan selkeästi monimutkaisin ja vaati eniten huolellisuutta. RS232-väylä itsessään vaati vain yhden kaksikanavaisen MAXIM202-piirin neljällä 100 nanofaradin suotokondenssaattorilla hoitamaan signaalit oikeaan muotoon mikroprosessorille ja taas mikroprosessorilta RS232-väylään. RS485-väylä jakaa mikroprosessorilta samat USART-väylät RS232-väylän kanssa, joten mikroprosessorin USART-väylän vastaanotto- eli RX-pinnit piti laittaa mikrokytkimen taakse, josta halutessaan voidaan valita käyttääkö piirilevy RS232-väylää vai RS485-väylää.

RS485-väylän suunnittelussa jännitteiden eristäminen ja ns. kättelyssä tapahtuvan lähetys- ja vastaanottolinjan sulkeminen ja avaaminen vaatii huomattavasti monimutkaisempaa suunnittelua. RS485-väylä eristettiin DC-DC-muuntimella muun piirilevyn käyttöjännitteestä ja paluuvirrasta, ja lähetys- sekä vastaanottolinjat eristetään optoerottimilla mikroprosessorin pinneistä, joilla hoidetaan myös linjojen sulkeminen ja avaaminen. DC-DC-muuntimeksi valittiin Perelin valmistama jo aiemmin Mikro-Väylä Oy:n tuotteissa käytettyä DC-DC-muunninta. Lähetys- ja vastaanottolinjojen erottamiseen vaadittiin kuusi optoerotettua linjaa. Tilan säästämiseksi piirikorttiin valittiin neljä- ja kaksikanavaiset Avago techgnologien valmistamiin optoerottimet. Niiden idea on jatkaa mikroprosessorilta saapuva lähetettävä signaali ja RS485-väylästä saapuva vastaanotettava signaali eteenpäin häiriöttömänä. Optoerottimien jälkeen laitettiin kumpaankin RS485-väylään MAX485-lähetinvastaanottimet, jotka hoitavat liikenteen RS485-väylästä optoerottimien kautta mikroprosessorille ja mikroprosessorilta lähtevän liikenteen optoerottimien kautta takaisin RS485-väylään. Kyt-

kentöjen jälkeen lisättiin mikropiirien tarvitsemat käyttöjännitteet ja suotokondensaattorit sekä lähetys-, vastaanotto- ja kättelyväylään punaiset, vihreät sekä keltaiset led-valot indikoimaan lähtevää ja saapuvaa signaalia. Tämän jälkeen sarjaliikenneväylä oli kokonaisuudessaan valmis.

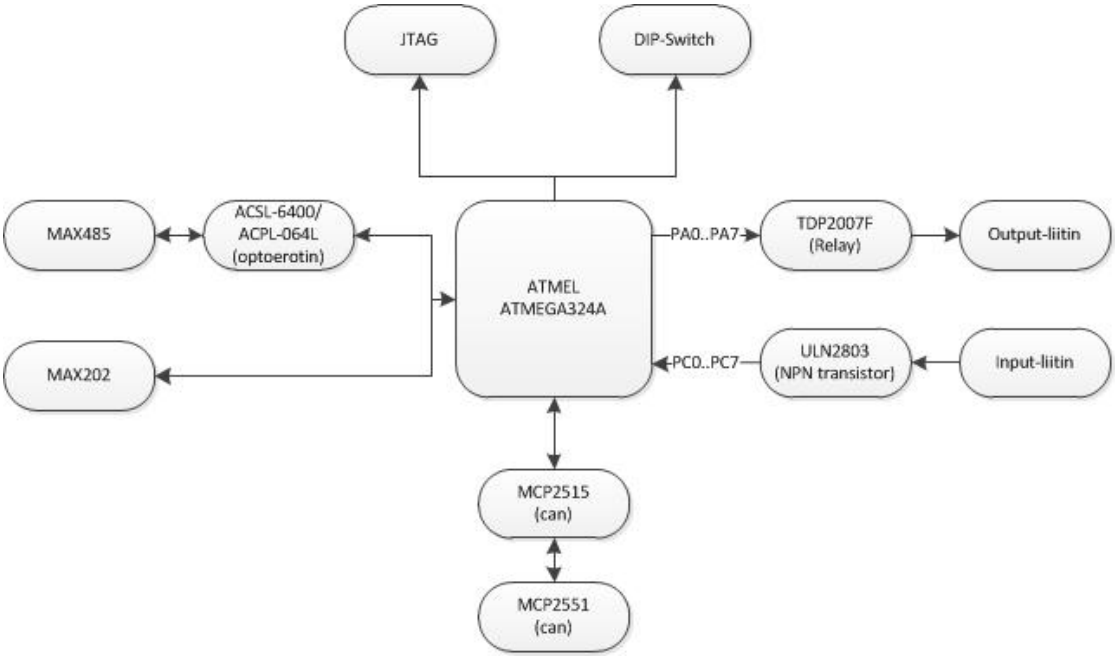
4.1.5 Mikroprosessorin suunnittelu ja lohkokaaviosuunnittelun viimeistely

Mikroprosessorin I/O-pinnit jaetaan lähes valmistajasta riippumatta kahdeksan, kuuden tai neljän pinnin nippuihin, joita kutsutaan I/O-porteiksi. ATmega324A:ssa on käytössä A-, B-, C- ja D-portit ja ne kaikki ovat kahdeksan pinnin portteja. Yleensä voi melko vapaasti valita, mitä porttia ja minkä portin pinnejä käyttää mihinkin väylään, mutta rajoituksiakin löytyy. Esimerkiksi USART-väylät ovat tarkkaan määrätty D-portin neljään ensimmäiseen pinniin, mikroprosessorin ohjelmointipinnit B-portin kolmeen viimeisempään pinniin ja A/D-muuntimen pinnit A-portin jokaiseen pinniin. Näitä portteja voi silti käyttää normaalina I/O-portteina, mutta jos edellä mainittuja ominaisuuksia haluaa käyttää, niin ainoa vaihtoehto on silloin käyttää ennalta määrättyjä pinnejä (Atmega 324 datasheet s. 7-8). Nyt kun sekä B- että D-portit oli hajotettu USART- ja ohjelmointiväyliksi, niin selkeyden takia C-portin pinnit valittiin input-väylään ja A-portin pinnit output-väylään. A/D-muuntimelle ei tämän I/O-kortin sovelluksissa ole tarvetta, joten A-porttia ei tarvinnut varata mihinkään erityiseen. Kahdeksankanavaisen Dip-switch-kytkimen laitoin B-porttiin osittain rinnan ohjelmointiväylän kanssa, koska ohjelmointiväylä varaa pinnit vain ohjelmoinnin aikana, minkä jälkeen ne vapautuvat vapaasti käytettäviksi. Tämän jälkeen ainoat vapaat I/O-pinnit olivat D-portin neljäs, viides, kuudes ja seitsemäs pinni, mistä neljäs ja viides käytettiin RS485-väylän kättelyyn ja kuudes CAN-väylän kättelyyn. D-portin seitsemäs pinni jätettiin led-valolle, joka indikoi ohjelman toimivuutta vilkuttamalla tilaansa tietyllä taajuudella. ATmega324A:sta löytyy ennalta määriteltyjä pinnejä, joiden tilaa voi ohjata päälle, pois tai vaihtaa tilaa rautatasolla ilman keskeyttäjiä (Atmega 324 datasheet s. 89, 141-142). PD7 oli yksi kyseistä ominaisuutta tukeva pinni, joten led-valo päätettiin lisätä siihen (ks. kuvio 10). Näin koodia pystytään yksinkertaistamaan ja mikroprosessorin kuormaa vähentämään, kun mikroprosessori itse hoitaa led-valon tilan vaihdon koodiin merkatulla taajuudella (ks. liite 2).



KUVIO 10. Mikroprosessorin kytkentätopologia piirikaaviopuolelta

Lohkokaaviopiirustuksesta on varsin hankala hahmottaa koko piirilevyn toimintaperiaatetta, koska lohkokaavio jakautuu useampaan sivuun ja sisältää erittäin monta yleisnäkemyksen kannalta turhaa komponenttia. Tein parempaa havainnollistusta varten yksinkertaistetun lohkokaavion (ks. kuvio 11) prototyypin konstruktiosta jättäen virransyötön ynnä muut toimintaperiaatteen kannalta epäolennaiset komponentit pois.

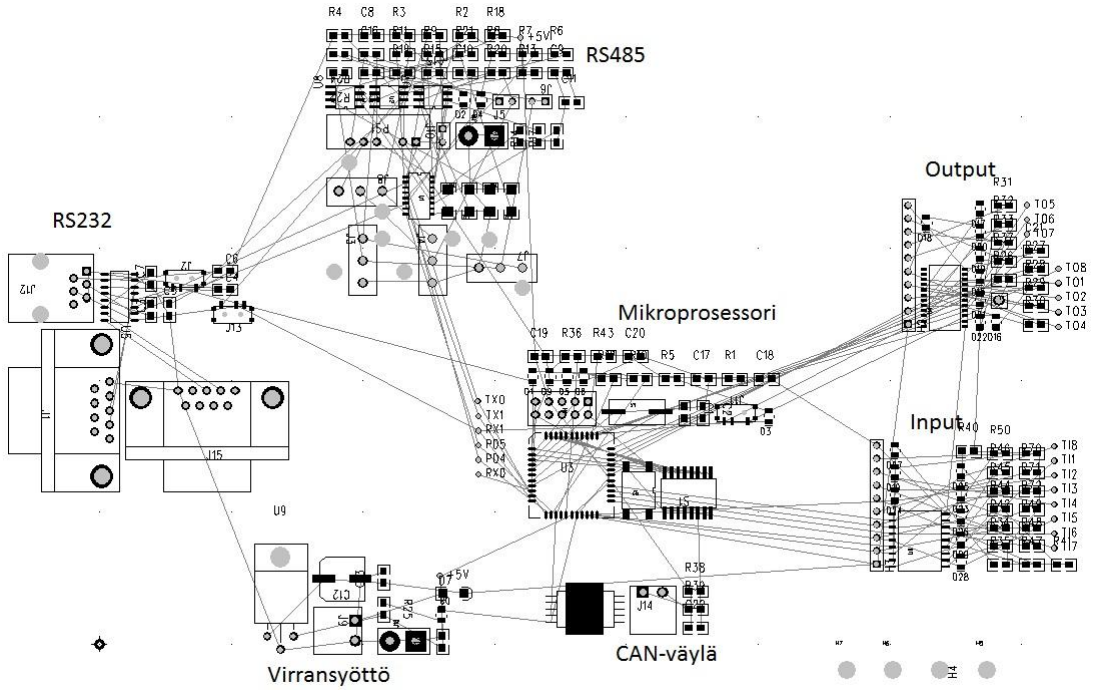


KUVIO 11. Prototyypin lohkokaaviosuunnittelun pääkonstruktio

4.2 Pohjapiirrustussuunnittelu

Lohkokaavion valmistuttua synkronoidaan PADS Logic:n (lohkokaaviosuunnittelun puoli) PADS Layout:n (pohjapiirustussuunnittelun puoli) kanssa, mikä lukitsee lohkokaaviopuolella määritellyt vedot ja raja-arvot ja siirtää komponenttien oikeat decalit suoraan pohjapiirustuspuolelle. Synkronoinnin jälkeen joko piirretään ennalta määritellyt piirikortin ulkomitat tai aletaan suoraan sommittelemaan komponenttien paikkoja ja piirikortin mitat piirretään jälkikäteen. Tässä tilanteessa tehtiin osittain jälkimmäisen kaavan mukaan, koska kyseessä on vielä prototyyppi ja tärkeämpää on saada kaikki komponentit yhdelle puolelle piirilevyä ja useiden testipisteiden kanssa, kuin puristaa komponentit mahdollisimman tiiviiseen pakettiin. Tästä huolimatta piti pitää mielessä, että tavoitellut ulkomitat lopulliselle piirikortille olisi noin 10cm*10cm. Se on myös nopeampaa ja helpompaa puristaa prototyyppi mahdollisimman kompaktiin muotoon tuotantopiirikorttia varten, jos valmiiksi ottaa huomioon optimaalisen komponenttien sijoittelun ja tilan tarpeen minimoinnin. (ks. Kärnä 2009a).

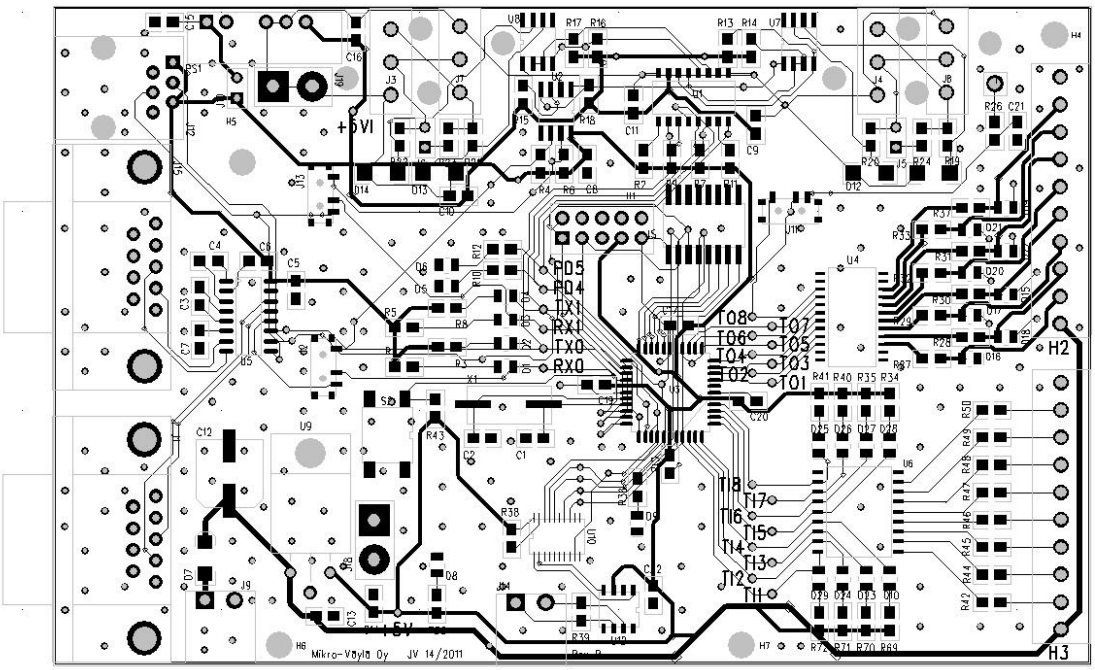
Komponenttien sijoittelu aloitettiin osioiden ensiksi molempien sarjaliikenneväylien komponentit, CAN-väylän komponentit ja input- sekä output-väylän komponentit omiin ryhmiin ja sen jälkeen sijoitellen ne ryhminä mikroprosessorin ympärille (ks. Kuvio 12). Vaikka tässä vaiheessa tilanne näyttää ehkä hieman epäselvältä, niin kyseinen toimintamalli on todettu olevan helpoin tapa hahmottaa kokonaisuuksia, mikä vähentää selkeästi uudelleensuunnittelua ja -sijoittelua.



KUVIO 12. Pohjapiirustussuunnitelman alkuasetelma

Tämän jälkeen jokaisen eri kokonaisuuden komponentit järjesteltiin toimivaan järjestykseen, jolloin niiden asettelu lohkoina lähemmäksi mikroprosessoria oli helpompaa. Nyt myös tilan optimointi vaati vähemmän vaivaa ja aikaa, kun jokainen lohko oli valmiiksi jo jossain määrin järkevässä muodossa. Komponenttien sijoittelun jälkeen täytyi vielä jokainen komponentti yhdistää lohkokaaviosuunnittelua vastaavaksi vetämällä linjat komponenttien pinnien välille niin, ettei mikään veto sivua toista missään tilanteessa. Ohjelmasta löytyy automaattinen ominaisuus mikä pitää huolen, etteivät väärät vedot voi sivuta toisiaan tai mennä lähemmäksi toisiaan, mitä lohkokaaviopuolella on määritelty. Linjojen vetämisen aikana komponenttisijoittelu eli vähäsen, koska vetojen tilan tarvetta on vaikea tarkkaan arvioida, mutta silti ajan säästö kasvaa suoraan verrannollisesti esisijoitteluun käytetyn ajan suhteen. Kun kaikki komponentit oli yhdistetty oikein ja komponenttien sijoittelu tyydytti, oli vuorossa täyttökuparoinnin luominen. Tällä menetelmällä yhdistetään jokaisen piirin maihin menevä pinni ja pidetään huoli, että paluuvirralla on mahdollisimman suora tie, eikä synny saarekkeita, mistä paluuvirta ei voi edetä.(Kärnä 2009a). Elektromagneettisten ongelmien välttämiseksi on ensisijaisen tärkeää, että paluusignaali pääsee etenemään suorinta tietä,

eikä aiheuta potentiaalieroja ja häiriöitä muissa piireissä kiertämällä pidempiä reittejä (ks. Kärnä 2009b). Lopulta luotiin gerber-tiedostot, joista löytyy tarvittavat tiedot piirilevyn valmistukseen. Kuviossa 13 nähdään piirikortin lopullinen ulkoasu.



KUVIO 13. Prototyyppilevyn pohjapiirustuskuva valmiina

4.3 Elektromagneettinen yhteensopivuus

Komponenttien sijoittelun ja piirilevysuunnittelun alkuvaiheilla ja sen edetessä on hyvä pitää mielessä elektromagneettinen yhteensopivuus ja elektromagneettiset häiriöt. Yksi piirilevysuunnittelun suurimpia virheitä on jättää EMC-suunnittelu piirilevysuunnittelun loppuvaiheeseen, koska silloin mahdolliset muutokset häiriöiden poistamiseen voi vaatia suuria ja aikaa vieviä muutoksia piirilevysuunnittelu. Vielä suurempi virhe on toki jättää EMC kokonaan huomioimatta, jolloin ongelmat todennäköisesti tulevat esiin piirilevyn epävakaassa toiminnassa tai viimeistään EMC-testeissä. Tässä vaiheessa häiriöiden poistaminen ja korjaaminen on usein erittäin kallista ja aikaa vievää touhua eikä välttämättä tule saavuttamaan alusta asti hyvin suunnitellun kortin EMC-tasoa silloinkaan. (ks. Armstrong & Clough 2007a).

Ongelmallisimpia sisäistä elektromagneettista häiriötä aiheuttavia komponentteja ovat AC/DC-muuntimet, hakkurit ja DC/AC-muuntimet, jotka aiheuttavat varsinkin maatasoon suuria potentiaalieroja ja häiritsevät häiriöherkkien mikropiirien, kuten mikroprosessorien yms. toimintaa. Häiriöitä ilmenee myös yleensä käyttöjännitteen puolella. (ks. Armstrong & Clough 2007a). Tarpeeksi suuri notkahdus käyttöjännitteessä saattaa sammuttaa mikroprosessorin tai logiikkapiirin, jona aikana piiri ei toimi odotetulla tavalla ja ongelmia alkaa syntyä. Tarpeeksi suuri piikki käyttöjännitteessä taas saattaa rikkoa mikropiirin kokonaan.

Toinen sisäisten EMC-häiriöiden kannalta ongelmallinen ryhmä on erittäin korkeataajuiset signaalit. Johtimet, missä kulkee korkeita taajuuksia, toimii tahattomina antenneina, jos johtimien pituus on yli kymmenesosa taajuuden aallonpituudesta. EMC:n kannalta suositeltavaa tosin olisi pyrkiä vetoihin, mitkä ovat alle kahdeskymmenesosan taajuuden aallonpituudesta. (Väänänen 2011.) Tämä tarkoittaa, että jo yhden GHz:n taajuudella johtimen ehdoton maksimipituus saisi olla 3 cm, mutta enintään 1,5 cm olisi suositeltava (Williams 2010).

Opinnäytetyössä suunniteltavassa piirikortissa ei AC/DC-muuntimia, DC/AC-muuntimia, hakkureita, korkeilla taajuuksilla toimivia mikropiirejä tai muita kriittisiä komponentteja juuri ole, mikä helpotti osaltaan suunnittelua. Varsinkin korkeataajuuskomponenttien puuttuminen mahdollisti komponenttien ja liittimien sijoittelussa

keskittymään loogiseen ryhmittelyyn, koska vetojen pituudet eivät piirikortin koon takia näillä taajuuksilla voi kasvaa niin pitkiksi, että ne alkaisivat aiheuttaa ongelmia. Oikeastaan piirikortissa suotokondensaattoreiden asettaminen mikropiirien käyttöjännitteiden ja maiden välille, output- ja input-linjojen eristäminen mikroprosessorilta, jokaisen piirin ja komponentin paluuvirralle useita eri reittejä ja RS485-väylän eristäminen piirikortin muusta käyttöjännitteestä ja maasta olivat ainoat sisäisen EMC-suunnittelun kannalta tehdyt ratkaisut. (ks. Armstrong & Clough 2007b).

Ulkoiset häiriöt olivat alusta alkaen projektin EMC-suunnittelussa suuremmin esillä, koska ohjainkortit tullaan sijoittamaan hihnamoduulin alle. Kortin lähietäisyydellä tulee olemaan siis sähkömoottori ja solenoidi, jotka luovat ympäristöön paljon nopeasti muuttuvia magneettikenttiä. Sähkömoottorin ja solenoidin magneettikentät johtuvat induktiivisesti lähiympäristössä oleviin johteisiin. Pelkkä magneettikentän johtuminen ei vielä ongelmia aiheuta, mutta nopeamuotoinen magneettivuon muutos aiheuttaa ongelmia ja häiriöitä.(Väänänen 2011). Ulkoisia häiriöitä on hankala estää piirilevysuunnittelun puolella, mutta mitä vähemmän piirilevy säteilee häiriötä ulospäin, sitä parempi on myös piirilevyn immuniteetti ulkoisia häiriöitä kohtaan. Turvallisin ratkaisu estää magneettikentän pääsyä piirikortin johteisiin on käyttää johtavaa, metallipitoista koteloa ja yhdistää piirilevyn maa koteloon.

5 TESTAUSSUUNNITTELU

Piirilevysuunnittelun jälkeen levyn gerber-tiedostot lähetettiin Brandner Oy:n valmistettavaksi Viroon. Prototyyppejä tilattiin 4 kappaletta, että kaikki ominaisuudet, kuten ketjutettu tiedonkulku RS485- ja CAN-väylän kautta voidaan testata. Opinnäytetyön kannalta testaus keskittyi pelkästään itse piirilevyn toimivuuden, kuten kytkentöjen ja komponenttivalintojen testaamiseen. Kaikkien osa-alueiden testaaminen ja toimivuuden todentaminen vaatii kuitenkin jokainen oman testiohjelmansa. Tästä syystä testaussuunnitteluosio tulee sisältämään myös paljon ohjelmointia. C-koodin kirjoittamiseen, kääntämiseen ja testikortille ajamiseen käytettiin Atmelin valmistamaa ilmaista AVR-Studio 4 -ohjelmaa. Ohjelma ajetaan mikroprosessorille Atmelin JTAGICE mkII -ohjelmointityökalulla, joka tukee mikroprosessoritason debug-ominaisuutta.

5.1 Piirilevyn testaus

Piirilevyn saavuttua aloitettiin kortin testaus systemaattisesti eri osa-alueet yksitellen testaten. Piirilevy kalustettiin ensin vain mikroprosessori, +5 voltin regulaattori ja niitä koskevat piirit juottaen. Juotosten jälkeen piirikorttiin kytkettiin virta jotta pystyttiin toteamaan ensimmäisen vaiheen toimivuus. +5 voltin jännitettä indikoiva led-valo syttyi palamaan, minkä jälkeen täytyi vielä testata, että +5 voltin linja menee kaikkialle minne sen oli tarkoitus mennä.

5.1.1 Ohjelman polttaminen mikroprosessorille

Koodin sisäänajo mikroprosessorille oli vuorossa seuraavaksi, joten ensimmäisen testikoodin idea oli olla vain mahdollisimman yksinkertainen. Piirilevylle laitettu ohjelman toimivuutta indikoiva led-valo sopi yksinkertaisen koodin toteutukseen hyvin. Ensimmäinen testiohjelman idea oli vilkuttaa led-valoa 500 ms syklillä. Kyseinen koodi oli nopeasti tehty peruskoodi ja AVR-Studion ominaisuuksiin kuuluu oikeiden porttien automaattiset määrittämiset ja kertoa vitaalit komennot valitun mikroprosessorin ohjaukseen. Jäljelle jäi vain avata D-portin yksi pinni, lisätä tarvittavat headerit, delay.h sekä io.h ja kirjoittaa led-valon vilkuttamisen hoitava koodi (ks. kuvio 14).

```
/************************************************************
Project : Led-toggle.c
Hardware: ATMega324 (16 MHz) +
Software: WinAVR 20100406
Date    : 25.03.2011
Author  : Juho Vahteri
Comments: LED-toggle
************************************************************/
#include <avr/io.h>
#include <util/delay.h>


int main (void)
{


DDRD = 0x80; //LED-pin to output

   for (;;) // Loop forever
    {
    _delay_ms(4000); // LED-toggle delay 4000ms/8 = 500ms
    PORTD ^= (7 << 6); // Switching PORTD PIN 7 to 0 and 1
    }
}

```

KUVIO 14. LED-toggle-koodi

Koodin kirjoituksen jälkeen koodi käännetään mikroprosessorin vaatimaan hex-muotoon ja ajetaan koodi kortille. Ehkä hieman yllättäen koodi meni suoraan toisella yrityksellä sisälle. Ensimmäisellä kerralla mikroprosessorin reset-pinni oli jäänyt vetämättä alas, mikä estää ohjelman polttamisen. Ohjelman poltettua led-valo alkoi vilkkua halutulla taajuudella, joten testauksessa voitiin siirtyä seuraavaan osaan.

5.1.2 I/O-väylän testaus

I/O-väylän komponenttien kalustuksen jälkeen oli suunniteltava testikoodi, millä saisi helposti testattua, että mikroprosessori reagoi input-liittimien jännitteen muutokseen ja ohjaa niitten mukaan output-liittimien tilaa. Testikoodin funktio on ohjata kahdeksan-pinnisen dip-switch-kytkimen neljännellä ja viidennellä pinnillä output-väylän ensimmäistä neljää ja viimeistä neljää pinniä. Samalla mikroprosessorin tulisi vetää ensimmäinen output-väylän pinni maihin, jos input-väylän ensimmäiseen pinniin syöte-

tään 3,5 V - 30 V jännite. Kaikkien pinnien erillinen testaus ei ole välttämätöntä, koska kytkentä on identtinen ja yhden pinnin testaus takaa muidenkin toiminnan. Kyseinen I/O-testikoodi kirjoitettiin edellisen testikoodin perään, koska ohjelman toimintaa indikoiva led-valo oli tarkoitus pitää mukana jokaisessa testikoodissa. I/O-testi ei vaatinut uusia headereita, joten koodiin tarvitsi vain määrittää muuttujiin input-liittimen ensimmäinen pinni, dip-switch-kytkimen neljäs ja viides pinni, määritellä porttien ja pinnien suunnat joko output- tai input-tilaan ja kirjoittaa itse koodi, joka määrittelee mikroprosessorille testikoodin mukaiset ehdot toimia (ks. Kuvio 15).

```
/*************************************************************
Project : I/O-test.c
Hardware: Oma ATMega324 (16 MHz) +
Software: WinAVR 20100406
Date    : 26.03.2011
Author  : Juho Vahteri
Comments: I/O-test
*************************************************************/
#include <avr/io.h>
#include <util/delay.h>

void main()
{
    int DIP4;
    int DIP5;
    int DIP6;
    int IN1;

    DDRA = 0xFF; //PORTA to output
    DDRB = 0x00; //PORTB to input
    PORTB = 0xFF; // PORTB pull-up on
    DDRC = 0x00; // PORTC to input
    DDRD = 0x80; // PORTD Pin 7 to output


for ( ;; )
    {
    _delay_ms(4000); //LED-toggle delay 400ms/8 = 500ms
    PORTD ^= (7 << 6); //Switching PORTD PIN 7 to 0 and 1

    DIP4 = (PINB & 0x08);
    DIP5 = (PINB & 0x04);
    DIP6 = (PINB & 0x02);
    IN1  = (PINC & 0x01);

    if(IN1 == 1)
    {
    PORTA = 0x01;
    }
    else
    {
    PORTA = 0x00;
    }

    if(DIP4 == 0)
        {
        PORTA = 0x0F;
        }
        else
        {
        PORTA = 0x00;
        }
    if(DIP5 == 0)
        {
        PORTA = 0xF0;
        }
        else
        {
        PORTA = 0x00;
        }
    }
}
```

KUVIO 15. I/O-väylän testauskoodi

Koodin käännettyä ja piirilevylle ajettua, ohjelma pyörähti oikein ja output-väylä reagoi oikein dip-switch-kytkimen tilan muutoksiin ja input-liittimeen syötettyyn 12 V jännitteeseen.

5.1.3 RS232-väylien testaus

Tässä prototyypissä päätettiin ottaa käyttöön kaksi RS232-väylää, eikä ole vielä täysin varma, jääkö toinen väylä lopulliseen tuotteeseen, mutta silti oli hyvä testata nyt kummatkin RS232-väylät samassa koodissa.

Aloin rakentamaan koodia taas led-toggle-koodin päälle joten aivan alussa täytyi vanhaan lisätä vain interrupt.h-headerin, koska led-valon vilkutus oli helpompi toteuttaa keskeyttäjillä samalla, kun käsittelee RS232-väylän tietoa. Headerin lisättyä vuorossa oli määrittää USART-väylän nopeus. 9600bps on yleisimpiä RS232-nopeuksia, ja se jakautuu hyvin käyttämäni kellotaajuuden kanssa, joten käytän sitä tässä koodissa. USART-väylän nopeuden ja kellotaajuuden määritteleminen onnistuu kätevästi valmiilla lyhyillä komennoilla, mitkä io.h-header määrittelee jälkikäteen tarkemmin. Nopeudet määritettyäni täytyy USART0- sekä USART1-pinnit kytkeä päälle, jotta mikroprosessori tietää käyttää niitä sarjaliikenneväylänä, eikä normaalina I/O-pinninä. AVR-Studiosta löytyy kaikkien Atmelin mikroprosessorien tiedot, millä komennolla ja millä bittijärjestyksellä saa kunkin portin ja toiminnon avattua, joten halutut komennot löytyivät vaivattomasti (Atmega 324 datasheet). UCSR0B ja UCSR1B -komennot hoitivat USART:n aktivoinnin. Seuraavaksi määrättiin viestin pituus kahdeksaan bittiin USCR0C ja USCR1C -komennolla ja viestin jako ala- ja ylärekisteriin UBBR0L/H ja UBBR1L/H -komennoilla. Vielä piti sallia USART-väylien lupa käyttää keskeyttäjää UCSR0B ja UCSR1B -komennoilla, jotta led-valon vilkkuminen voidaan keskeyttää viestin vastaanottamisen, käsittelyn ja lähetyksen ajaksi.

Tämän testiohjelman pääohjelma on vilkuttaa led-valoa samalla 500 ms taajuudella. Mikäli mikroprosessori saa viestin joko USART0 tai USART1 -väylään, ohjelma keskeyttää päätapahtuman ja käsittelee saamansa viestin, tallentaa sen UDR0 tai UDR1 -rekisteriin ja lähettää saman viestin takaisin, tässä tapauksessa tietokoneelle. Edellä mainittu ohjelmarakenne on nopea ja helppo tapa testata sarjaliikenneväylän toimivuus (ks. kuvio 16). Jos piirilevy palauttaa saman komennon, mikä sille on lähetetty,

niin väylä toimii oikein. Testaus näytti varsin nopeasti, että koodi ja kytkentä toimivat oikein. Tietokoneelta lähetetyt viestit palautuivat oikein sekä ensimmäisestä että toisesta sarjaliikenneväylästä. Piirilevyyn laitettiin USART0 sekä USART1 -porttien väliin vihreä led-valo indikoimaan saapunutta liikennettä ja punainen led-valo indikoimaan lähtevää liikennettä. Tällä tavalla kytkennän oikeellisuuden ja viestien liikkumisen pystyy toteamaan suoraan, eikä tarvitse seurata signaaleita oskilloskooppia käyttäen.

5.1.4 RS485-väylän testaus

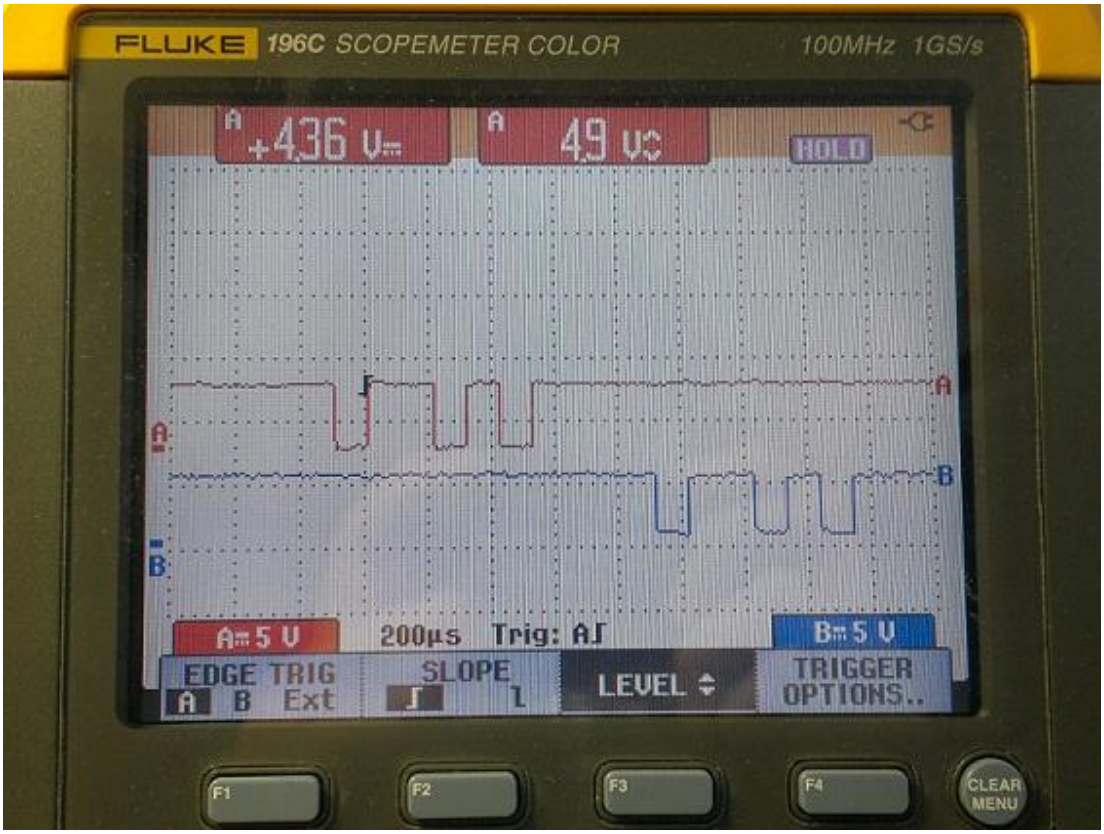
Kortista oli vielä tällä hetkellä testaamatta RS485- sekä CAN-väylä. Mikro-Väylä Oy:n toiveena oli, että CAN-väylän testaaminen ja tarkempi tutkiminen jätetään vielä tässä vaiheessa pois, koska akuuttia tarvetta sille ei ole. Opinnäytetyön piirilevy tulee ohjaamaan heinäkuun lopulla asennettavaa Tampereen kirjaston lainausautomaattia. Vaikka aikaa on useita kuukausia, niin nykyisen moduuliohjausohjelman kääntämisen ja väylävalinnasta seuranneiden muutosten lisäksi edessä on vielä käytännön testaus, jossa kortin toiminta testataan perusteellisesti, mikä voi viedä viikkoja. Mikro-Väylä Oy:lle on tässä vaihein ensisijaisen tärkeää, että niin kortista kuin ohjelmastakin saadaan toimiva versio siihen mennessä, ja vasta tämän jälkeen aletaan tutkia, kannattaako RS485-väylää korvata CAN-väylällä.

RS485-väylän testaamiseen pystyttiin käyttämään samaa koodia samoihin (ks. kuvio 16), millä testattiin RS232-väylän toimivuus, koska kummatkin väylät ovat kytketty mikroprosessorilla samaan väylään. Tässä koodissa ei ole otettu huomioon RS485-väylän tarvitsemaa kättelyitä tai ajoituksia, koska ideana on vain todeta signaalin menevän oikeassa muodossa mikroprosessorin USART-väyliin. Nopein ja helpoin tapa tämä on todeta katsomalla, että USART-väylän punainen sekä vihreä led-valo välähtää ja tämän jälkeen oskilloskoopilla todeta, että kulkevat signaalit ovat jännitetasoltaan halutut. RS485-väylän komponentit kalustettua ohjainkortti kytkettiin Mikro-Väylä Oy:n valmistamaan RS232-RS485-muuntimeen. Sen kautta syötettiin tietokoneelta RS232-väylältä dataa, joka sitten jatkoi datan RS485-väylän mukaisesti piirilevylleni. USART-väylän led-valot välähtivät dataa lähettäessä ja oskilloskoopilla pys-

tyi tarkastamaan, että tuleva signaali näyttää järkevältä ja lähtevä signaali vastaa saapunutta signaalia (ks. Kuvio 17).

```
/************************************************************
Project : RS232.c
Hardware: Oma ATMega324 (16 MHz) +
Software: WinAVR 20100406
Date    : 30.03.2011
Author  : Juho Vahteri
Comments: RS232-Echo-Interrupt
************************************************************/
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

#define USART_BAUDRATE 9600 // Set baudrate to 9600
#define BAUD_PRESCALE (((8000000 / (USART_BAUDRATE * 16UL))) - 1) /*prescale baudrate to match
the used freqruency */



int main (void)
{


DDRD = 0x80; //PORTD PIN 7 to output

//USART0
UCSR0B |= (1 << RXEN0) | (1 << TXEN0); //Turn on the transmission and recxeption circuitry
UCSR0C |= (3 << UCSZ00); //Use 8-bit character sizes

//USART1
UCSR1B |= (1 << RXEN1) | (1 << TXEN1);
UCSR1C |= (3 << UCSZ10);

//USART0
UBRR0L = BAUD_PRESCALE; /* Load lower 8-bits of the baud rate value into
the low byte of the UBRR register */
UBRR0H = (BAUD_PRESCALE >> 8); /* Load upper 8-bits of the baud rate value into
the high byte of the UBRR register */

//USART1
UBRR1L = BAUD_PRESCALE;
UBRR1H = (BAUD_PRESCALE >> 8);

//USART0
UCSR0B |= (1 << RXCIE0); // Enable the USART Recieve Complete interrupt (USART_RXC)

//USART1
UCSR1B |= (1 << RXCIE1);

sei(); // Enable the Global Interrupt Enable flag so that interrupts can be processed

   for (;;) // Loop forever
    {
    _delay_ms(4000); //1500ms/8 = 500ms
    PORTD ^= (7 << 6); //PORTD

    }
}

ISR(USART0_RX_vect)
{
   char ReceivedByte;
   ReceivedByte = UDR0; // Fetch the recieved byte value into the variable "ByteReceived"
   UDR0 = ReceivedByte; // Echo back the received byte back to the computer
}
ISR(USART1_RX_vect)
{
    char ReceivedByte2;
    ReceivedByte2 = UDR1;
    UDR1 = ReceivedByte2;
}
```

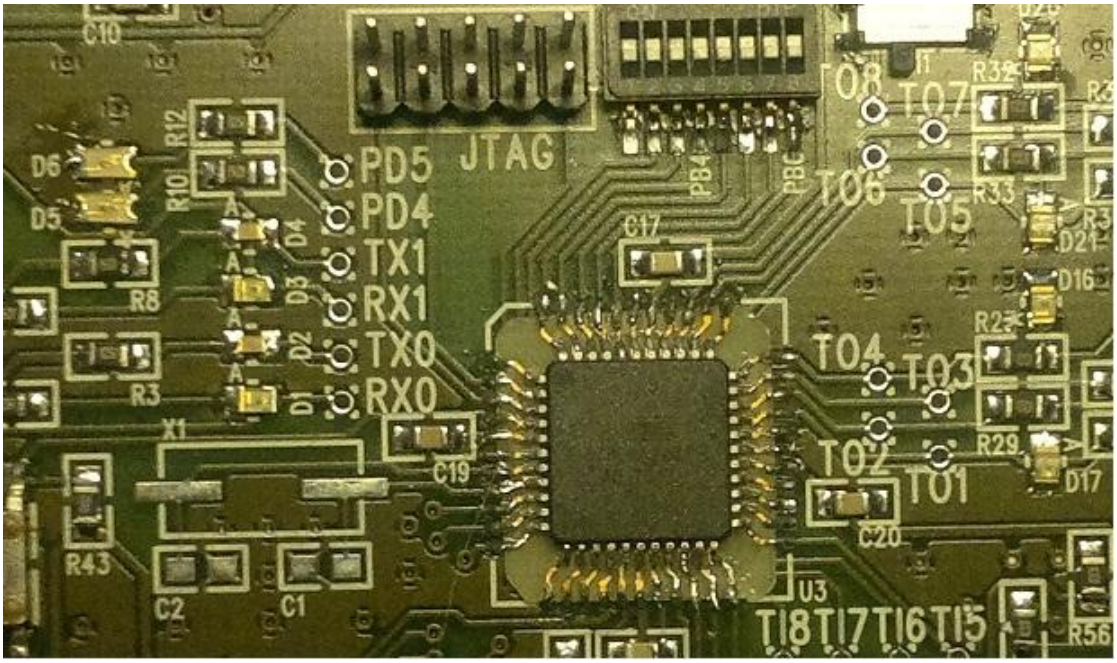
KUVIO 16. RS232-väylän testauskoodi



KUVIO 17. RS485-väylän tulo- ja lähtösignaali.

5.2 Testauksessa ilmenneet ongelmat ja niiden korjaukset

Testausvaiheen edetessä piirilevysuunnittelussa tapahtuneita virheitä nousi esiin muutama kappale. Ensimmäinen ongelma oli mikroprosessorin liian suuri decali piirilevyllä. Piirilevysuunnittelua tehtiin sekä koti- että työkoneella ja toisella koneella tehdyt decalit eivät päivittyneet oikein, jos unohti siirtää komponenttikirjastot muiden tiedostojen ohella. Tässä tapauksessa kotikoneella tehty oikea TQFP44-decali jäi kotikoneelle ja PADS korvasi työkoneella TQFP44-tiedoston vanhalla, padsista löytyvästä, decalilla. Ongelma saatiin ratkaistua koulussa tehdyllä apupiirilevyllä(ks. kuvio 18), jotta prototyyppipiirilevyn testausta päästiin jatkamaan.



KUVIO 18. Mikroprosessorin apupiirilevy

RS232-väylän testauksessa toisen RS232-väylän lähetys- ja vastaanottolinjat olivat yhdistetty väärinpäin lohkokaaviosuunnittelussa, mikä esti viestin kulun toista väylää pitkin. Ongelma korjattiin nostamalla MAX202-piirin virheellisesti kytketyt jalat ylös ja liittämällä ne prosessorille oikein hyppylangoilla, minkä jälkeen kummatkin väylät toimivat oikein

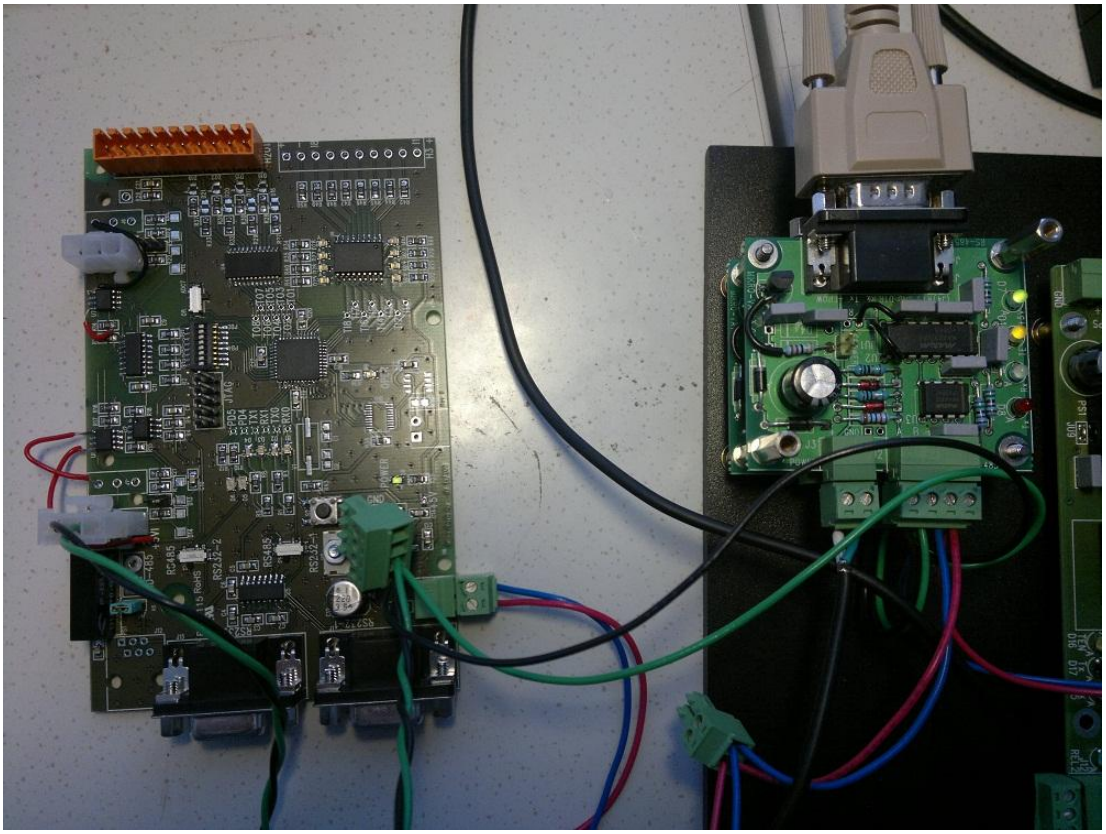
RS485-väylän testauksessa Avago Technologiesin nelikanavaisen optoerottimen datasheetistä löytyneet mikropiirin mitat eivät täsmänneet millään tapaa oikeaan mikropiiriin. Tästä syystä optoerottimen juottaminen piirilevylle oli mahdotonta, ja piirilevystä jouduttiin tilaamaan toinen prototyyppi, mihin kaikki havaitut ongelmat korjattiin. Optoerottimen decali saatiin tehtyä oikean kokoiseksi tulostamalla paperille erikokoisia decaleita ja testaamalla, mikä sopii kyseiselle mikropiirille. Datasheetin virheellisyydestä ilmoitettiin www.Farnell.com-sivustolle, mistä datasheet ladattiin.

Toinen prototyyppi jouduttiin tilaamaan, ennen kuin RS485-väylää oli päästy testaamaan, joten kolmanteen prototyyppiin täytyi varautua. Toisen prototyypin saavuttua se kalustettiin täysin ja testattiin kaikki osa-alueet uudelleen. Kaikki toimi kuten piti ja

havaitut ongelmat oli korjattu oikein. RS485-väylää päästiin näin ollen myös testaamaan, koska nyt optoerottimen decali osui kohdalleen.

RS485-väylää testatessa ensimmäinen väylä toimi oikein, mutta toinen väylä tuntui olevan mykkänä. Lyhyen testirupeaman ja datasheetien tarkastelun jälkeen selvisi, että RS485-väylässä oli lohkokaaviopuolella jäänyt lisäämättä käyttöjännite ja paluuvirta MAX485-piireille ja Avago Technologiesin kaksikanavaisen optoerottimen toisen kanavan anodi ja katodi olivat väärin päin. Nämä olivat selkeitä huolimattomuusvirheitä lohkokaaviosuunnittelussa, mutta korjattavissa helposti hyppylangoilla (ks. kuvio 19), joten piirilevyn uudelleensuunnittelua ja kolmatta prototyyppiä ei tarvittu.

Elektromagneettisten häiriöiden testaus täytyi jättää opinnäytetyöstä pois opinnäytetyön asettamien aikarajojen takia. Testausympäristönä piti toimia Mikro-Väylä Oy:n palautusautomaatin yksi ohjausmoduuli, mutta ylimääräistä moduulia testausta varten ei ollut opinnäytetyön tekohetkellä saatavilla ja suunnitteilla oleva ohjausmoduuli ei kerinnyt siihen vaiheeseen opinnäytetyön aikana, että sitä olisi voinut käyttää testausympäristönä.



KUVIO 19. Mikro-Väylä Oy:n valmistama RS232-RS485-muunnin (oik.) ja piirilevyn 2. prototyyppi (vas.).

6 YHTEENVETO

Opinnäytetyön tavoitteena oli kehittää Mikro-Väylä Oy:n vaatimuksia vastaava ohjauskortti Mikro-Väylä Oy:n osittain uusiutuvaan palautusautomaatiojärjestelmään. Opinnäytetyön osa-alueet ja tavoitteet oli tutkia eri väylävaihtoehtoja ohjauskorttien väliseen kommunikointiin ja valita kriteerit täyttävä vaihtoehto, suunnitella ja toteuttaa vaatimuksia vastaava piirilevy, valita asiaan parhaiten sopivat komponentit hinnan, saatavuuden ja ominaisuuksienkin suhteen sekä testata suunniteltu piirilevy ja sen kaikki osa-alueet, jotta ne vastaisivat asetettuja ominaisuuksia.

Tiedossa olleet kriteerit auttoivat rajaamaan eri väyläratkaisuista suuren osan pois, ja lopulta lähempään tarkasteluun päätyi SPI-, IIC-, CAN- ja RS485-väylä. SPI- ja IIC-väylät todettiin häiriönsiedoltaan riittämättömäksi ja tiedonsiirron varmuus puutteelliseksi. CAN- ja RS485-väylä sopivat kummatkin kriteerien puolesta väyläratkaisuksi hyvin, mutta RS485-väylään päädyttiin Mikro-Väylä Oy:n pyynnöstä. CAN-väylälle lisättiin paikka prototyyppipiirilevyyn, että väylän testaus ja tutkiminen onnistuu, jos RS485-väylä tulevaisuudessa tuo ongelmia.

Piirikortin suunnittelussa kytkentäkaavioiden suunnittelu, komponenttivalinnat ja haluttujen toimintojen toteutus onnistui siinä mielessä hyvin, että ainoat ongelmat olivat huolimattomuusvirheistä johtuneita väärinkytköksiä ja väärän kokoisia decaleita. Näiden virheiden takia jouduttiin prototyypistä tekemään toinen versio, minkä jälkeen piirilevy saatiin toimimaan halutulla tavalla.

Testausvaiheessa selvisi, että piirilevysuunnittelussa tehty kytkentäkaavio ja komponenttivalinnat menivät suunnitelmien mukaan. Piirikortti toimi asiakasvaatimusten mukaisella tavalla ja täytti vaaditut arvot. Piirikortin elektromagneettisten häiriöiden testaaminen sille tarkoitetussa ympäristössä ei onnistunut opinnäytetyön asettamissa aikarajoissa, koska Mikro-Väylä Oy:n uudistunut palautusautomaatin moduuli ei saapunut tarpeeksi ajoissa.

LÄHTEET

Armstrong, K. Clough, C. Design Techniques for EMC. Part5-1 – Printed Circuit Board (PCB) Design and Layout. The EMC journal September 2007.

Armstrong, K. Clough, C. Design Techniques for EMC. Part 5-2 – Printed Circuit Board (PCB) Design and Layout. The EMC journal November 2007.

Atmega 324 datasheet. 2010. Atmel co. Viitattu 24.03.2011. http://www.atmel.com/dyn/resources/prod_documents/doc8272.pdf.

CAN Bus 2006. Viitattu 11.04.2011. http://intranet.daiict.ac.in/~ranjan/esp2006/presentation/CAN%20Bus.pdf.

CAN Bus Description 2011. Viitattu 11.04.2011. http://www.interfacebus.com/CAN-Bus-Description-Vendors-Canbus-Protocol.html.

CAN dictionary May 2011. Viitattu 10.04.2011 http://www.can-cia.de/fileadmin/cia/pdfs/CANdictionary-v4_fi.pdf.

CAN history 2011. Viitattu 12.04.2011. http://www.can-cia.de/index.php?id=161#c138.

EIA-422/485 Bus Description. Viitattu 12.04.2011. http://www.interfacebus.com/Design_Connector_RS485.html.

Explanation of Maxim RS-485 Features 01.12.2000. Viitattu 13.04.2011. http://pdfserv.maxim-ic.com/en/an/AN367.pdf.

Hirsjärvi, S. Remes, P. & Sajavaara, P. 1997. Tutki ja Kirjoita. 10. osin uud. p. Jyväskylä: Gummerus.

I^2C-bus specification and user manual 19.07.2007. Viitattu 13.04.2011. http://www.nxp.com/documents/user_manual/UM10204.pdf.

Introduction to Serial Peripheral Interface January 2002. Viitattu 12.04.2011. http://www.eetimes.com/discussion/beginner-s-corner/4023908/Introduction-to-Serial-Peripheral-Interface.

Kärnä, M. 2009a. Piirilevyn suunnittelu, osa 1: Piirilevy on kriittinen komponentti. Prosessori nro 6-7. S. 49-51.

Kärnä, M. 2009b. Piirilevyn suunnittelu, osa 2: Reititä signaalit järkevästi. Prosessori nro 9. S. 38-39.

RS-422 and RS-485 Application Note June 2006. Viitattu 12.04.2011. http://www.bb-elec.com/bb-elec/literature/tech/485appnote.pdf.

TPD2007F datasheet. 31.10.2006. Toshiba. Viitattu 15.04.2011. http://www.farnell.com/datasheets/391889.pdf.

ULN2803A datasheet. July 2006. Darlington Transistor Array. Viitattu 15.04.2011. http://www.farnell.com/datasheets/87988.pdf.

Understanding the SPI Bus with NI LabVIEW 12.05.2011. Viitattu 15.05.2011. http://sine.ni.com/nipdfgenerator/nipdfgenerator?pageURL=http://zone.ni.com/devzone/cda/tut/p/id/9119&clientAppName=dz&dotsPerPixel=&dotsPerPoint=.

SPI Interface Specification 19.09.2005. Viitattu 14.04.2011. http://www.vtitechnologies.jp/documents/data_sheet/TN15_SPI_Interface_Specification.pdf.

Väänänen, O. 2011. EMC-suunnittelu ja -testaus. Jyväskylän ammattikorkeakoulu, EMC-suunnittelu, luentomateriaali.

Williams, T. 11.02.2010. Designing for EMC. Viitattu 20.04.2011 http://www.elmac.co.uk/pdfs/desfremc.pdf.

www.esacademy.com. 2011. Viitattu 13.04.2011. http://www.esacademy.com/assets/images/hsbus.jpg.

www.interfacebus.com. 2010. Viitattu 11.04.2011. http://www.interfacebus.com/CANbus-Message-Frame-Format-Protocol.png.

www.neural-eng.com. 2011. Viitattu 13.04.2011. http://www.neural-eng.com/A7/images/stories/Doc_Pictures/RS485Halfduplex.png.

www.wikimedia.org. 2011. Viitattu 15.04.2011. http://upload.wikimedia.org/wikipedia/commons/thumb/f/fc/SPI_three_slaves.svg/350px-SPI_three_slaves.svg.png.

LIITTEET

Liite 1. Led-valon tilan käyttäen CTC-keskeyttäjää ja sisäistä 16-bittistä kelloa.

```
/************************************************************
Project : Interrupt-timer.c
Hardware: ATMega324 (16 MHz)
Software: WinAVR 20100406
Date    : 14.04.2011
Author  : Juho Vahteri
Comments: Flashing led by using intterrupt
************************************************************/
#include <avr/io.h>
#include <avr/interrupt.h>

int main (void)
{
    DDRD = 0xB0; // LED as output

    TCCR1B |= (1 << WGM12); // Timer 1 for CTC mode

    TIMSK1 |= (1 << OCIE1A); //Enable CTC interrupt

    sei(); //Enable global interrupt

    OCR1A = 15624; // Set CTC compare value to 1Hz at 1MHz AVR clock, with a prescaler of 64

    TCCR1B |= ((1 << CS10) | (1 << CS11)); //Start timer at prescaler of 64



    for(;;)
    {

    }

}

ISR(TIMER1_COMPA_vect)
{
    PORTD ^= (7 << 6); //Toggle led
}
```

Liite 2. Led-valon tilan vaihtaminen käyttäen CTC-keskeyttäjää ja ATmegan Toggle OC2A(PD7) on Compare Match -toimintoa.

```
/************************************************************
Project : Led-toggle-hardware.c
Hardware: ATMega324 (16 MHz)
Software: WinAVR 20100406
Date    : 16.04.2011
Author  : Juho Vahteri
Comments: Flashing led using only hardware CTC
************************************************************/
#include <avr/io.h>
#include <avr/interrupt.h>

int main (void)
{
    DDRD = 0xB0; // LED as output

    TCCR2B |= (1 << WGM22); // Timer 2 for CTC mode

    TCCR2A |= (1 << COM2A0); // Enable timer 2 Compare Output channel A in toggle mode

    OCR2A = 244; // Set CTC compare value ~1/4Hz at 1Mhz AVR clock

    TCCR2B |= ((1 << CS20 | 1 << CS21 | 1 << CS22)); //Start timer with a prescaler of 1024

    for(;;)
    {

    }

}
```