# LAAJENNUSKORTIT JA OHJELMISTOPÄIVITYS RACEDAC-TIEDONKERUULAITTEESEEN

Jari Korhonen

Opinnäytetyö
joulukuu 2011

Ohjelmistotekniikan koulutusohjelma
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

| Tekijä(t) | Julkaisun laji | Päivämäärä |
|---|---|---|
| KORHONEN, Jari | Opinnäytetyö | 08.12.2011 |
| | Sivumäärä<br>40 | Julkaisun kieli<br>Suomi |
| | Luottamuksellisuus<br>( ) saakka | Verkkojulkaisulupa myönnetty<br>( X ) |

Työn nimi
LAAJENNUSKORTIT JA OHJELMISTOPÄIVITYS RACEDAC-TIEDONKERUULAITTEESEEN

Koulutusohjelma
Ohjelmistotekniikka

Työn ohjaaja(t)
PIETIKÄINEN, Kalevi

Toimeksiantaja(t)
KORKOLAINEN, Tuomo

Tiivistelmä

Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa kaksi laajennuskorttia RaceDAC-tiedonkeruulaitteeseen. Toteutettavat laajennukset olivat SD-muistikorttituki ja GPS-paikannus. Opinnäytetyöhön kuului myös RaceDAC-laitteen ohjelmiston päivitys uusia laajennuskortteja tukevaksi. Opinnäytetyössä selvitettiin myös, kuinka kiihtyvyysanturia voitaisiin käyttää laitteessa ja vertailtiin edullisia kiihtyvyysantureita ja niiden sopivuutta RaceDAC-laitteessa käytettäväksi.

Opinnäytetyössä perehdyttiin myös SD-muistikortin käyttämään SPI-väylään, FAT-tiedostojärjestelmään, MTK-komentoihin ja kiihtyvyysantureiden ominaisuuksiin.

Työssä käsitellään tuote- ja ohjelmistosuunnittelussa vastaan tulleita ongelmia ja niiden ratkaisuja. SD-muistikorttilaajennus saatiin toimimaan vaatimusten mukaisesti 50 Hz:n tiedonkeruutaajuudella. GPS-laajennus toimi vaaditulla tavalla sisätiloissa silloin, kun moduuli ei ollut yhteydessä satelliittiin. Opinnäytetyössä saatiin selville, että ainakin teoriassa halvimman hintaluokan kiihtyvyysantureista löytyy sopivia vaihtoehtoja RaceDAC-laitteessa käytettäväksi.

Avainsanat (asiasanat)
GPS, SPI-väylä, SD-muistikortti, kiihtyvyysanturi, DSPIC33F, tiedonkeruu

Muut tiedot

Author(s)
KORHONEN, Jari

Type of publication
Bachelor´s Thesis

Date
08.12.2011

Pages
40

Language
Finnish

Confidential

( ) Until

Permission for web publication
( X )

Title
EXTENSION CARDS ANS SOFTWARE UPDARE FOR RACEDAC DATA ACQUISITION DEVICE

Degree Programme
Software Engineering

Tutor(s)
PIETIKÄINEN, Kalevi

Assigned by
KORKOLAINEN, Tuomo

Abstract

The purpose of this bachelor's thesis was to develop two extension cards for RaceDAC data acquisition module. Updating the software of the device was also a part of this thesis. The first extension was a support for an SD-memory card and the second extension was a support for a GPS-module.

Additional tasks in this bachelor's thesis were to choose the GPS-module, resolve the usage of an accelometer in a device and compare moderately priced accelometers suitable for this product.

The thesis also clarifies techniques that are used and needed for developing extension cards including information about SPI-bus technique, FAT filesystem and the usage of MTK commands for configuring GPS-modules. The thesis also contains information about problems that showed up while developing the extension cards and the solutions for these problems.

Keywords
GPS, SPI-bus, SD-memory card, accelometer, DSPIC33F, data acquisition

Miscellaneous

# SISÄLTÖ

# KUVIOT

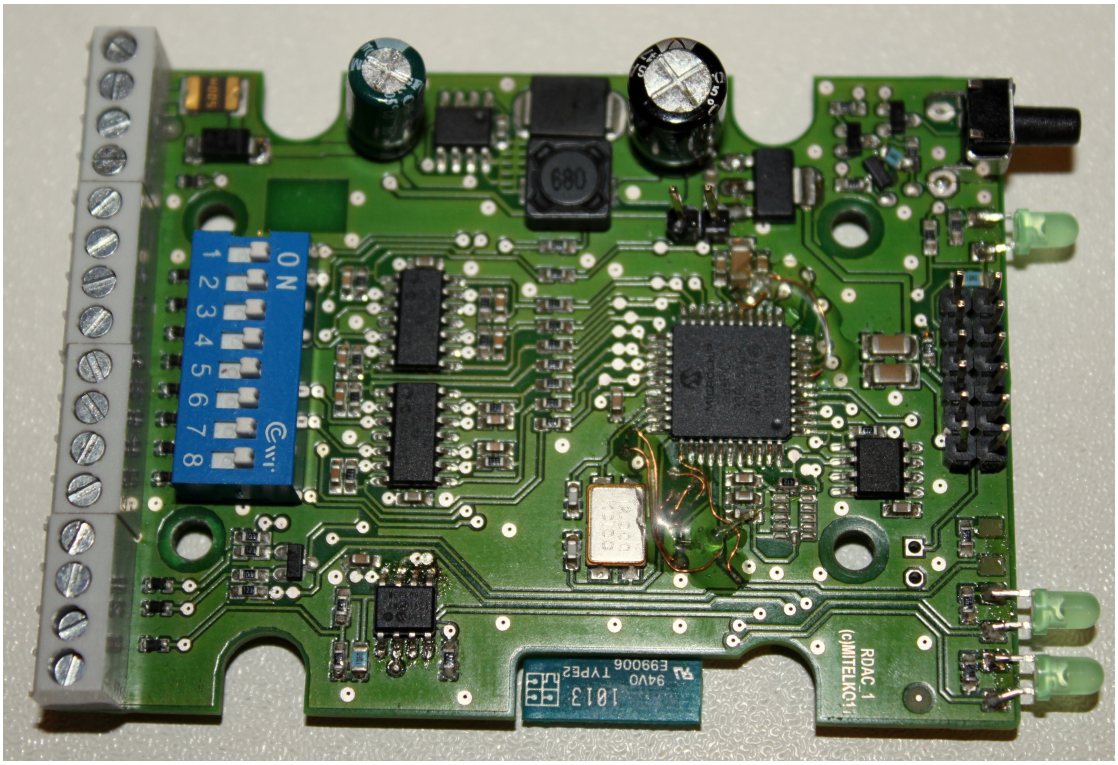


# TAULUKOT

# 1 ASIAKASVAATIMUKSET

## *1.1 RaceDAC-tiedonkeruulaite*

RaceDAC on moottoriurheilussa käytettävä tiedonkeruulaite, johon voi liittää auton tai muun moottorikäyttöisen kulkuneuvon moottorin tai alustan antureilta tulevia analogisia tai digitaalisia signaaleja ja kerätä tietoa niistä. Laitteeseen on mahdollista liittää kahdeksan analogista ja kaksi digitaalista tulosignaalia samanaikaisesti. Laitteen keräämää tietoa voi lukea bluetoothin kautta PC:llä, puhelimella tai millä tahansa laitteella, jossa on standardin mukainen bluetooth-yhteys käytössä. RaceDAC on suunniteltu toimimaan myös RaceChrono mobiilisovelluksen kanssa. Kuvio 1 esittää RaceDAC-laitteen prototyyppiä, jolle lisäkortit ja ohjelmistopäivitys on opinnäytetyössä tehty.




KUVIO 1. RaceDAC-tiedonkeruulaite

Laite rakentuu Microchipin 16-bittisen dsPIC33FJ64GP204 -mikrokontrollerin ympärille. Kontrollerissa on 13 analogista sisääntuloa, joista käytössä on 10. Kahdeksan käytössä olevista on tarkoitettu yhdistettäväksi moottoriajoneuvosta löytyviin anturei-hin ja loput kaksi on kortin sisäistä valvontaa varten. Mikrokontrollerissa on tuki kahteen sarjaporttiliitäntään. Näistä toinen on käytetty bluetooth-moduulin datan lukemiseen ja lähettämiseen. Kontrollerin kahdesta SPI-moduulista ei ole alunperin käytössä kumpikaan.

## *1.2 RaceChrono*

RaceChrono on tiedonkeruusovellus, jota käytetään moottoriurheilussa laskemaan kierrosaikoja ja mittaamaan suorituskykyä. RaceChrono-sovellus käyttää kierrosaikojen ja suorituskyvyn mittauksiin paikannustietoja GPS-moduulilta ja signaalitietoja auton tai minkä tahansa muun moottoriajoneuvon moottorinohjausyksiköltä. Sovellus ottaa vastaan tietyn muotoisia ASCII-merkkirivejä, joihin GPS-tiedot ja moottorinohjausyksikön signaalit on kirjoitettu. Sovellus osaa ottaa datarivit vastaan bluetoothin kautta tai tiedostona, jos halutaan tarkastella aikaisemmin kerättyjä tietoja. (RaceChrono. 2011)

Opinnäytetyössä RaceDAC-tiedonkeruulaitteeseen tehtiin laajennuskortit, joiden avulla RaceChrono-sovellukselle voitiin lähettää GPS-paikannustiedot ja tallentaa kerätyt tiedot muistikortille myöhempää tarkastelua varten. RaceChronon kotisivuilta löytyi dataformaatti, jossa tiedot täytyy olla, että sovellus osaisi käyttää niitä.

## *1.3 Laajennuskorttien tarvemäärittely*

Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa kaksi laajennuskorttia RaceDAC-laitteeseen. Toteutettavat laajennukset olivat ulkoinen muistikortti ja ulkoinen

tai sisäinen GPS-moduuli. Myös kiihtyvyysanturin käyttöä laitteessa on selvitetty osana opinnäytetyötä. Jokainen laajennus tuli suunnitella siten, että laajennuskortti kiinnitettiin aina samoihin mikrokontrollerin pinneihin, ja ohjelmallisesti pääteltiin, mikä laajennus oli laitteeseen liitetty. Laajennuskorttien suunnittelua rajoitti se, että RaceDAC-kortissa oli laajennuskäyttöön tarkoitettu vain neljä mikrokontrollerin pinniä. Tämän takia kaikkia laajennusominaisuuksia ei voinut suunnitella samalle piirikortille. Tarvittaessa oli luvallista ottaa laajennuskäyttöön mukaan myös laitteessa olevan in-circuit-ohjelmointiliittimen kaksi pinniä. Suunnittelua rajoitti myös RaceDAC-laitteen kotelo. Laajennuskortit tuli suunnitella siten, että ne mahtuivat samaan koteloon RaceDAC-kortin kanssa.

Ulkoisena muistina laajennuskortissa tuli käyttää SD-muistikorttia (tai mikroSD-muistikorttia), joka liitetään prosessorin SPI-väylään. SD-muistikorttiliitäntää varten käytettävien I/O -pinnien minimoimiseksi muistikortin paikallaolo ja kirjoitussuojaus täytyi toteuttaa ohjelmallisesti. Muistikortin tiedostojärjestelmänä tuli olla FAT16, joten tuki tulisi maksimissaan kahden gigatavun muistikortille.

Opinnäytetyössä osatehtävänä oli sopivan GPS-moduulin valinta laitteeseen. GPS-moduulin liittämiseksi tuli käyttää sarjaporttiliitäntää. Liitäntätavan sai päättää itse moduulin valinnan jälkeen. Jos GPS-moduuliksi valittaisiin ulkoinen ratkaisu, täytyi sille suunnitella laajennuskorttiin myös fyysinen liitin. Laajennuskorttiin tuli myös suunnitella tarvittavat tehonsiirtopiirit GPS-moduulia varten.

Ennen kiihtyvyysanturin valintaa tuli vertailla edullisia 3-akselisia kiihtyvyysantureita niiden toiminta-alueen, virrankulutuksen, liitännän ja muiden ominaisuuksien perusteella. RaceDAC-kortissa on valmiina paikka ADXL345 -anturille, mutta kyseisen anturin huonon saatavuuden takia sille oli löydettävä korvaava komponentti. Kiihtyvyysanturin sai suunnitella liitettäväksi joko RaceDACin pääpiirikorttiin tai opinnäytetyössä suunniteltuun laajennuskorttiin.

### 1.4 Ohjelmistopäivitys

RaceDAC-laitteessa olevalla mikrokontrollerilla on ohjelma, joka on kehitetty Microchipin MPLAB-ympäristössä C-kielellä. Tämän ohjelman tarkoitus on kerätä tiedot laitteeseen liitetyiltä antureilta ja lähettää ne bluetooth-yhteyden kautta halutulle laitteelle. Opinnäytetyössä tehtävä ohjelmistopäivitys tuli tehdä kyseiseen tiedonkeruuohjelmaan. Päivitetyssä ohjelmassa tuli olla tuki SD-muistikortille ja GPS-moduulille eli molempien laajennuskorttien ominaisuuksille. Ohjelman tuli myös osata automaattisesti tunnistaa, onko laitteeseen kiinnitetty laajennuskorttia vai ei. Ilman laajennusta laitteen tuli toimia samoin kuin ennen ohjelmistopäivitystä.

## 2 TARVITTAVAT TEKNIIKAT

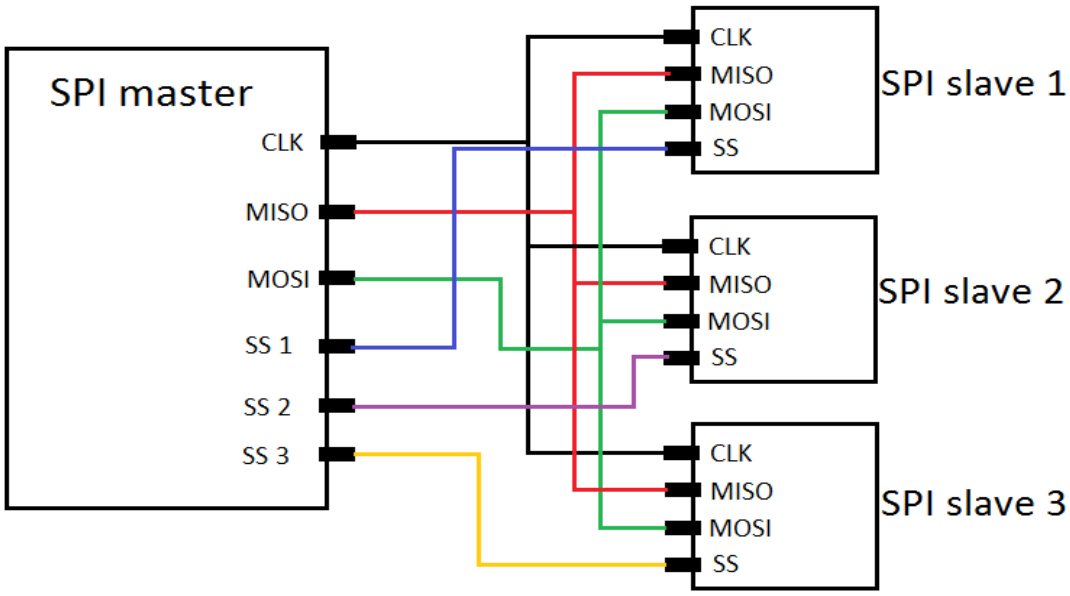
### 2.1 SPI-väylä

SPI-väylä on tiedonsiirtoväylä, jossa data liikkuu sarjamuotoisena. SPI-lyhenne tulee englannin kielen sanoista serial peripheral interface. Väylään voi liittää yhden masterlaitteen ja useita slave-laitteita, mutta master voi kommunikoida ainoastaan yhden slave-laitteen kanssa samanaikaisesti. SPI-väylässä laitteiden välillä on full duplex-yhteys, eli molemmat laitteet voivat sekä lähettää että vastaanottaa dataa samanaikaisesti. (Serial peripheral interface bus. 2011)

SPI-väylän peruskonfiguraatiossa on neljä signaalijohdinta: MISO, MOSI, CLK ja SS. MISO-signaali on slave-laitteen lähettämä signaali master-laitteelle (Master Input Slave Output), ja MOSI-signaali on puolestaan master-laitteen lähettämä signaali slave-laitteelle (Master Output Slave Input). CLK on masterin tuottama kellosignaali, jonka mukaan väylässä kulkeva tieto liikkuu. SS-signaalilla (slave select) valitaan, minkä slave-laitteen kanssa halutaan muodostaa yhteys, silloin kun väylässä on useita

slave-laitteita samanaikaisesti. (Serial peripheral interface bus. 2011)

Jotta SPI-väylään voitaisiin liittää useita slave-laitteita, täytyy jokaisen slave-laitteen tukea kolmitasoista ulostuloa (tri-state). Tämä tarkoittaa sitä, että slave-laitteen MISO-pinnin impedanssi kasvaa niin suureksi, että se on käytännössä ”poikki”. Useat SPI-väylään liitettävät laitteet tukevat tätä ominaisuutta, ja se aktivoituu, kun SS-signaali asetetaan ylätilaan (laite ei ole valittuna väylällä). Kuvio 2 kuvaa kolmen slave-laitteen kytkemistä yhteiseen master-laitteeseen SPI-väylällä.




KUVIO 2. SPI-väylällä yksi master ja kolme slave-laitetta

Kun yhteys masterin ja slave-laitteen välille muodostetaan, master alkaa lähettää CLK-linjalla kellopulssia. Tämän jälkeen master asettaa sen slave-laitteen SS-pinnin alatilaan, jonka kanssa se haluaa muodostaa yhteyden. Seuraavaksi data alkaa liikkua väylällä siten, että master lähettää bitin MOSI-linjalle ja slave lukee tämän bitin samalta linjalta. Tämän jälkeen slave lähettää bitin MISO-linjalla ja master lukee sen. Nämä neljä operaatiota tapahtuvat jokaisella CLK-linjalla kulkevan pulssin aikana riippumatta siitä, halutaanko niiden tapahtuvan vai ei. (Serial peripheral interface bus. 2011)

## *2.2 Massamuisti*

### 2.2.1 Yleistä

Massamuisti on haihtumatonta tietokoneen käyttämää muistia. Tietokoneen kiintolevy, USB-muisti, muistikortit, CD-ja DVD-levyt ovat esimerkkejä massamuistista. Termi massamuisti tarkoittaa siis kestomuistia, eikä se ota kantaa esimerkiksi toteutustekniikkaan. Massamuisti voi olla sähköistä, magneettista tai vaikka mekaanista häviämätöntä muistia.

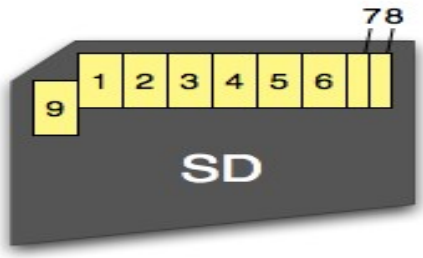
### 2.2.2 SD-muistikortti

Lyhenne SD tulee sanoista Secure Digital, ja se on muistikorttityyppi, joka on erityisesti suunniteltu käytettäväksi pienissä kannettavissa laitteissa kuten matkapuhelimissa, digikameroissa ja muissa sulautetuissa järjestelmissä. Tämän muistikorttitekniikan kehitti SD Card Association (SDA). (SecureDigital. 2011.)

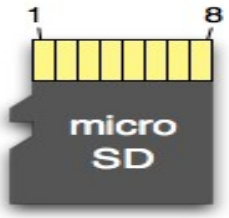
SD-muistikortit on jaettu kolmeen tyyppiin fyysisen koon mukaan ja muistikapasiteetin mukaan. Fyysisen koon mukaan SD-kortit on jaettu Standard size (SDSC,SDHC, SDXC ja SDIO), Mini size (miniSD, miniSDHC ja miniSDIO) ja Micro size (microSD, microSDHC ja microSDXC) -kortteihin. Muistikapasiteetin mukaan kortit on jaettu SDSC (<4 GB), SDHC (4-32 GB) ja SDXC (32 GB-2 TB) -tyyppeihin. Standardin mukaan SDSC-kortteihin kuuluvat vain kortit 2 GB:n asti. On kuitenkin olemassa 4 GB:n SDSC-kortteja, joten niissä saattaa olla yhteensopivuusongelmia niiden laitteiden kanssa, jotka tukevat vain SDSC-tyyppisiä muistikortteja. (SecureDigital. 2011.)

SD-muistikortit tukevat kahdenlaista tapaa sen hallitsemiseen. Muistikorttia voi käyttää joko SPI-tilassa tai SD-tilassa. SD-tilassa dataväylällä voi lähettää tai vastaanottaa samanaikaisesti neljä bittiä yhdellä kellojaksolla, kun taas SPI-tilassa yhdellä kellojak-

solla voi lähettää ja vastaanottaa yhden bitin. SD-tila vaatii kuitenkin isäntälaitteelta kuusi IO-pinniä, jos halutaan käyttää tehokkainta 4 bitin dataväylää. Tästä syystä opinnäytetyössä käyttöön otettiin SPI-väylä, joka tarvitsee minimissään kolme IO-pinniä (CLK,MOSI,MISO). Yleensä käytössä on SPI-tilassa kuitenkin 4 pinniä, koska väylällä ei voi olla useampia laitteita ilman neljättä CS-pinniä (chip select). Kuviossa 3 näkyy SD-muistikortin pinnit sekä SPI-tilassa että SD-tilassa. (Elastic Sheep. 2010; Kingmax Digital Inc. 2006.)




| Pin | SD | SPI |
|---|---|---|
| 1 | CD/DAT3 | CS |
| 2 | CMD | DI |
| 3 | VSS1 | VSS1 |
| 4 | VDD | VDD |
| 5 | CLK | SCLK |
| 6 | VSS2 | VSS2 |
| 7 | DAT0 | DO |
| 8 | DAT1 | X |
| 9 | DAT2 | X |




| Pin | SD | SPI |
|---|---|---|
| 1 | DAT2 | X |
| 2 | CD/DAT3 | CS |
| 3 | CMD | DI |
| 4 | VDD | VDD |
| 5 | CLK | SCLK |
| 6 | VSS | VSS |
| 7 | DAT0 | DO |
| 8 | DAT1 | X |

KUVIO 3. SD-muistikortin ja microSD-muistikortin pinnit ja toiminnot

### 2.2.3 FAT-tiedostojärjestelmä

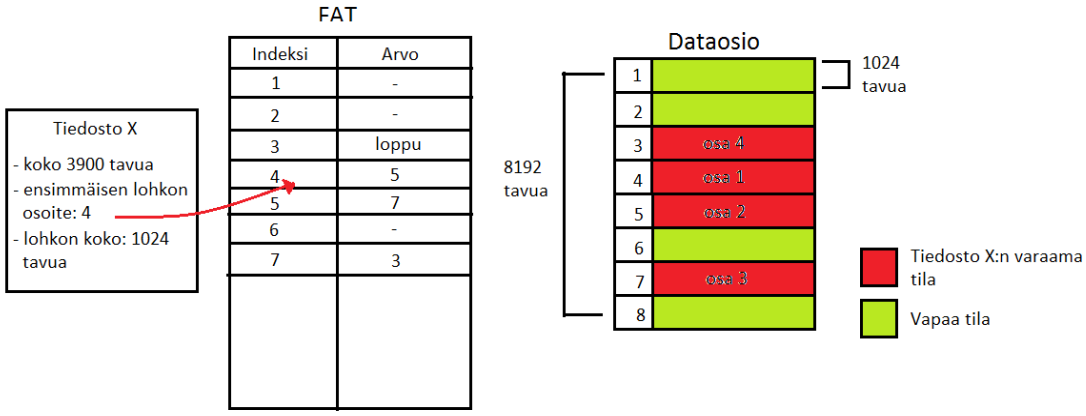
FAT on tiedostojärjestelmä, jota käytetään nykyään muistikorttien, USB-muistitikkujen ja erilaisten pienten kannettavien laitteiden tiedostojärjestelmänä. Lyhenne FAT tulee englannin kielen sanoista file allocation table. FAT-tiedostojärjestelmä oli ennen myös käytössä DOS-käyttöjärjestelmässä ja Windows 9x -sarjan käyttöjärjestelmissä. Uudemmissa käyttöjärjestelmissä FAT-tiedostojärjestelmästä luovuttiin sen heikon tietoturvan ja suuren hukkatilan vuoksi. (FAT filesystem. 2011)

FAT-tiedostojärjestelmät on nimetty käytetyn taulukon bittimäärän mukaan. On olemassa kolme eri versiota: FAT12 (12 bittiä), FAT16 (16 bittiä) ja FAT32 (32 bittiä). Bittien lukumäärä määrää, kuinka suurta tallennuskapasiteettia tiedostojärjestelmä kykenee tukemaan. Opinnäytetyössä FAT16-tiedostojärjestelmän takia laite tukee muistikortteja 2 GB:n asti.

FAT-tiedostojärjestelmässä levy jaetaan 512 tavun sektoreihin. Näitä sektoreita voidaan ryhmitellä lohkoihin, ja lohkon koko riippuu siitä, kuinka suuri levykapasiteetti on käytössä. Koko on kuitenkin aina 512 tavun monikerta ja tämä tiedonvarausyksikkö on pienin levyllä osoitettavissa oleva yksikkö. Vaikka tiedoston koko olisi kolme tavua, se vie levyllä tilaa yhden lohkon koon verran. (Räty, J. 2007)

Osion ensimmäinen sektrori on aina boot-sektori, joka sisältää perustiedot tiedostojärjestelmän tyypistä. Tässä käynnistyslohkossa on tiedot lohkon koosta, juurihakemiston ominaisuuksista ja FAT:ien ominaisuuksista. Käynnistyslohkossa on myös osoittimia, jotka osoittavat muistia tiedostojärjestelmän osion tärkeisiin tietoihin. (Reen, P. & Mohaanswamy, N. 2010)

Käynnistyslohkon jälkeen sijaitsevat FAT:it (File Allocation Table). FAT on taulukko, joka sisältää tiedot siitä, kuinka lohkot on jaettu data-osiossa. Näitä FAT:ejä on aina varmuuden vuoksi kaksi kappaletta ja ne sijaitsevat peräkkän levyllä. Taulukon alkioon on tallennettuna aina seuraavan tiedoston varaaman lohkon osoite. Kuviossa 4 on esitetty, kuinka tiedoston varaamat lohkot näkyvät FAT-taulussa.




KUVIO 4. File Allocation Table

Kuviossa 4 on esitetty, kuinka 3900 tavun tiedosto varaa tilaa levyltä ja kuinka FAT-taulussa tiedoston eri lohkojen osoitteet on määritelty. Tiedostojärjestelmän juurihakemistossa on jokaisen levyllä sijaitsevan tiedoston tiedot. Näistä tiedoista näkyy esimerkiksi tiedoston varaaman ensimmäisen lohkon osoite. Tiedoston varaamat lohkot eivät välttämättä ole aina peräkkäiset levyn lohkot. Tämän takia on olemassa FAT-taulu, jonka alkion arvo kertoo seuraavan lohkon osoitteen, joka on varattu tiedoston käyttöön. FAT-taulussa kulkee mukana myös tieto viimeisestä lohkosta, joka sisältää tiedoston dataa. Kuvion 4 esimerkissä näkyy myös, että 3900 tavun tiedosto varaa levytilaa 4096 tavua. Dataosiossa tiedoston eri osat eivät myöskään ole järjestyksessä, eli FAT-taulun tuhoutuessa ei voida enää tietää, mitkä lohkot kuuluvat millekkin levyllä olevista tiedostoista.

Jokaisesta levyllä olevasta tiedostosta ja kansiosta on olemassa 32 tavun kokoinen tietue tiedostojärjestelmän juurihakemistossa. Jokainen tietue sisältää tiedoston nimen, koon, luontiajan, muokkausajan ja ensimmäisen lohkon osoitteen. Kun tiedosto avataan, haetaan ensin ensimmäisen lohkon data osoitteen osoittamasta paikasta. Tämän jälkeen katsotaan FAT-taulusta seuraavan tiedoston datalohkon osoite ja tätä jatketaan niin kauan kuin FAT-taulussa löydetään tieto, joka kertoo lohkon olevan viimeinen tiedostolle kuuluva datalohko. (Reen, P. 2010)

## *2.3 Kiihtyvyysanturi*

Kiihtyvyysanturi on komponentti, joka mittaa kiihtyvyyttä vapaaseen pudotukseen verrattuna. Tämä tarkoittaa siis samaa kiihtyvyyttä, kuin minkä ihminen voi tuntea esimerkiksi auton kiihdyttäessä vauhtia. Ollessaan levossa maanpintaan nähden kiihtyvyysanturi näyttää tulosta 1 g ylöspäin.

Kiihtyvyysanturin voi toteuttaa usealla eri tavalla. On olemassa esimerkiksi pietsosähköiseen ilmiöön perustuvia kiihtyvyysantureita. Niiden toiminta perustuu siihen, että kiihtyvässä liikkeessä ollessaan pietsosähköinen materiaali muodostaa päidensä välille siihen kohdistuvaa voimaa vastaavan jännitteen. Toinen esimerkki kiihtyvyysanturin toteutuksesta on kapasitanssin muutokseen perustuvat kiihtyvyysanturit. Kahden johtavan materiaalin välillä on jonkin suuruinen kapasitanssi, joka muuttuu niiden välisen välimatkan muuttuessa. Kapasitanssin muutokseen perustuvissa kiihtyvyysantureissa kiihdytysvoima kasvattaa tai lyhentää johtavien materiaalien välimatkaa ja kapasitanssin suuruudesta voidaan päätellä voiman suuruus. (Dimension Engineering LLC. 2011)

Kiihtyvyysantureita on olemassa analogisia ja digitaalisia. Analogiset anturit antavat ulostulossa jännitteen, joka vastaa tiettyä kiihtyvyyttä. Esimerkiksi maanpinnalla lepotilassa anturi voisi näyttää 2,5 V, joka vastaisi 1 g:n suuruista voimaa. Digitaaliset kiihtyvyysanturit antavat ulostulona kanttiaaltoa ja ne liitetään joko SPI-väylään tai I2C-väylään. Kanttiaallon taajuus kertoo tällöin kiihtyvyyden. Analogiset anturit ovat yleisemmin käytettyjä, koska niiden lukeminen mikrokontrollerin avulla on yksinkertaisempaa. Mikrokontrollereissa on lähes aina analogisisääntulo, johon kiihtyvyysanturin voi liittää. Kontrolleri muuntaa analogisen jännitetason digitaalisesti luettavaksi ja ohjemassa voidaan tämän jälkeen analysoida tulosta. Digitaalisen anturin käyttö vaatii enemmän sitä lukevalta ohjelmalta, joten usein päädytään käyttämään analogisia kiihtyvyysantureita. Analoginen anturi saattaa joissakin tapauksissa vaatia kuitenkin enemmän oheiselektroniikkaa toimiakseen oikein. Tämä riippuu käytetyn mikrokontrollerin analogikanavien ominaisuuksista. Joidenkin mikro-ohjainten analogikanavat vaativat niihin liitetyltä anturilta alle 10 K:n ulostuloimpedanssin. Useilla analogisilla

kiihtyvyysantureilla ulostuloimpedanssi on kuitenkin yli 10000 ohmia, joten tarvitaan puskurivahvistin alentamaan ulostuloimpedanssi mikrokontrollerin analogisisääntulolle sopivan pieneksi. (Dimension Engineering LLC. 2011)

Kiihtyvyysanturilla on seuraavia ominaisuuksia, jotka vaikuttavat anturin valintaan: herkkyys, maksimikiihtyvyys, päivitystaajuus ja akseleiden määrä. Herkkyys tarkoittaa tässä tapauksessa sitä, kuinka paljon anturin ulostulosignaalin taso muuttuu, kun kiihtyvyys kasvaa yhden g:n verran. Digitaalisissa antureissa herkkyys voidaan ilmoittaa esimerkiksi yksikkönä g/digit tai counts/g. Yksikkö g/digit tarkoittaa kuinka monta g:tä kiihtyvyyden täytyy muuttua, että anturin ulostulossa voidaan nähdä muutos. Yksikkö counts/g tarkoittaa kuinka monta eri arvoa anturi pystyy näyttämään kiihtyvyyden muuttuessa yhden g:n verran. Analogisissa antureissa herkkyys ilmoitetaan usein yksikkönä V/g, joka tarkoittaa jännitteen muutosta kiihtyvyyden muuttuessa 1 g:n verran. Maksimikiihtyvyys kertoo kuinka suuria kiihtyvyyksiä anturi kykenee mittaamaan menemättä rikki. Usein kiihtyvyysantureissa on mahdollista vaihtaa maksimikiihtyvyyttä, mutta se luonnollisesti johtaa huonompaan herkkyyteen. Päivitystaajuus määrää sen, kuinka usein anturilta voidaan saada luotettavasti luettua kiihtyvyys. Tämä voi aiheuttaa ongelmia lähinnä analogisissa kiihtyvyysantureissa, koska ne näyttävät jotakin tulosta kokoaikaisesti kun taas digitaaliset anturit lähettävät mittaustuloksen vain sen ollessa mahdollista. Akseleiden määrä kertoo kuinka monesta suunnasta kiihtyvyysanturi voi mitata kiihtyvyyden samanaikaisesti. On olemassa kaksiakselisia ja kolmiakselisia versioita. Jos kiihtyvyysanturin sisältävän laitteen asento halutaan saada selville, käytetään kolmiaksilista kiihtyvyysanturia, joka kykenee mittaamaan kiihtyvyyden kaikista kolmesta suunnasta. Kolmiakselisia kiihtyvyysantureita kutsutaan myös 3D-kiihtyvyysantureiksi.

# 3 GPS-MODUULIN VALINTA

## *3.1 Valintaperusteet*

Opinnäytetyön yksi osatehtävä oli sopivan GPS-moduulin valinta laajennuskorttia varten. Valitun moduulin tuli olla kykenevä päivittämään paikannustiedot 10 Hz taajuudella ja olla mahdollisimman edullinen. Moduulin oli myös oltava tilattavissa yksittäiskappaleena.

Sopivan paikannusmoduulin löytäminen oli hankalaa, koska suurin osa moduuleista oli tilattavissa vain suoraan valmistajalta, suurissa erissä ja ilman hintatietoja. Opinnäytetyötä tehtäessä ei kuitenkaan pyydetty tarjouksia valmistajilta, koska yksittäisten moduulien tilaaminen ei olisi luultavasti ollut mahdollista. Halvempia malleja oli tilattavissa verkkokaupoista, mutta kaikissa niissä paikannustietojen päivitystaajuus oli liian pieni. Kaikissa halvimmissa moduuleissa oli mahdollista saada paikannustiedot luettua vain 1 sekunnin välein eli 1 Hz:n päivitystaajuudella.

Toimeksiantajalla oli jo opinnäytetyön aloitusvaiheessa yksi vaihtoehto GPS-moduuliksi. Kyseisessä moduulissa päivitystaajuuden sai asetettua 10 Hz:iin ja sen hinta jäi hieman alle 20 euroon. Tämä moduuli valittiin opinnäytetyössä paikannusominaisuuden sisältävään laajennuskorttiin.

## *3.2 GlobalTop FGPMMOPA6B GPS-moduuli*

GlobalTop Technologyn PA-sarjan PA6B-moduuli perustuu MediaTek Technologyn kehittämään GPS-paikannustekniikkaan. Myös muut GlobalTopin GPS-moduulit on toteutettu yhteistyössä MediaTekin kanssa. PA6B-moduuli on pieni, edullinen ja se ei

tarvitse toimiakseen paljon oheiskomponentteja. Tämän takia kyseinen moduuli oli hyvä valinta opinnäytetyöhön. (GlobalTop. 2011)

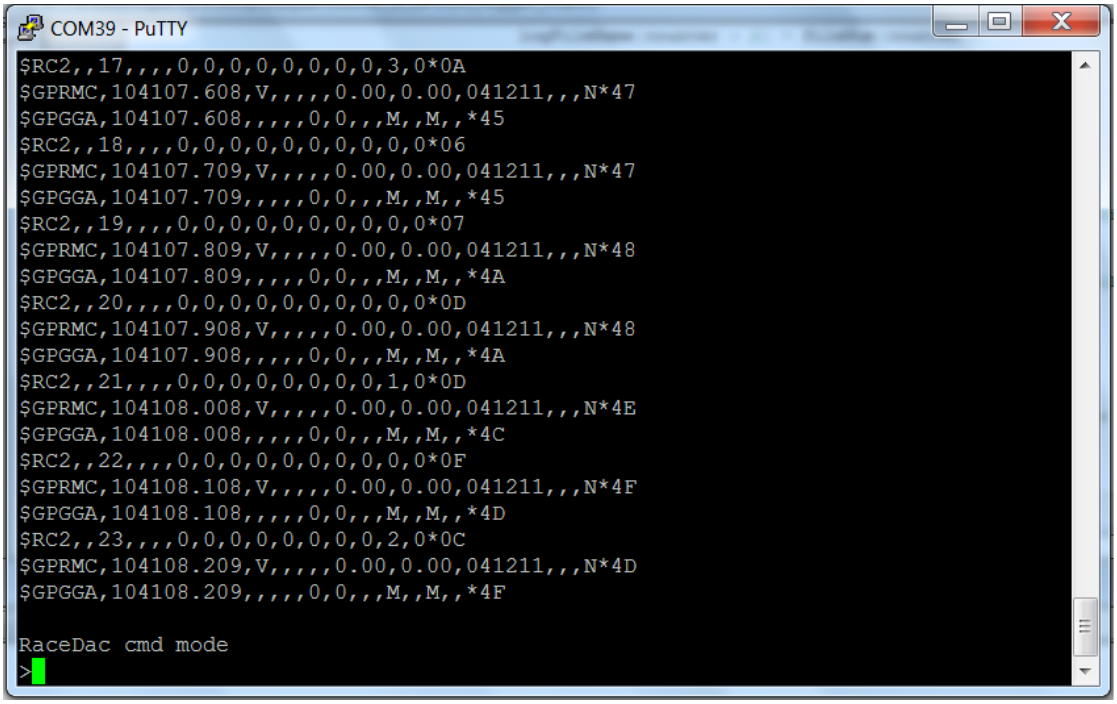
PA6B-moduuli päivittää paikannustiedot 10 Hz:n taajuudella, joka oli vaatimuksena opinnäytetyössä. Moduuli kykenee noin 3 metrin paikannustarkkuuteen, joka on riittävä RaceDAC-laitteessa käytettäväksi. Lisäksi moduuli kykenee myös mittaamaan nopeutta (max 515 m/s), korkeutta (max 18 km) ja kiihtyvyyttä (max 4 G). Koska RaceDAC-laitetta käytetään autoissa, veneissä tai moottoripyörissä, moduuli soveltuu varmasti tarvittaessa myös nopeuden mittaukseen ja kiihtyvyyden mittaukseen.

## *3.3 NMEA-viestit ja MTK-komennot*

NMEA tulee sanoista National Marine Electronics Association. NMEA on määrittely, joka on kehitetty merenkulussa käytettyjen laitteiden väliseen kommunikointiin. Tällaisia laitteita on esimerkiksi kaikuluotain, tuulimittari ja GPS-vastaanotin. Kaikki nämä laitteet lähettävät tietyn ohjeen mukaisia ascii-merkkirivejä.

Jokainen NMEA-viesti alkaa $-merkillä, jota seuraa viisi merkkiä. Näistä merkeistä kaksi ensimmäistä kertoo, mikä laite on lähettämässä riviä ja seuraavat kolme kertovat lähetettävän viestin tyypin. Tämän jälkeen tulee viestin varsinainen dataosuus, joka voi koostua useasta eri tiedosta. Kaikkien erityyppisien rivien sisältö on ennalta määrätty, joten jonkin tiedon puuttuessa se jätetään tyhjäksi. Kaikki lähetettävät tiedot erotetaan kuitenkin pilkulla riippumatta siitä, onko kyseinen tieto saatavissa vai ei. Dataosuuden jälkeen rivillä on *-merkki, jos tarkistussumma on käytössä. Muussa tapauksessa rivi loppuu merkkeihin <CR><LF>. Tarkistussumman ollessa käytössä, *-merkin jälkeen tulee tarkistussumma 8-bittisenä heksalukuna. Myös tarkistussumman sisältävät NMEA-viestirivit päättyvät lopetusmerkkeihin <CR><LF>. Kuviossa 5 nä-

kyy esimerkkinä GPS-moduulin lähettämät GGA- ja RMC-tiedot. Muut rivit ovat RaceDAC-laitteeseen kiinnitettyjen signaaleiden tietoja.




KUVIO 5. RaceDAC-laitteen lähettämät logirivit ja paikannustietorivit

Kuviossa 5 GPS-moduuli ei ole saanut yhteyttä satelliittiin, joten paikannustiedot eivät näy. Ainoastaan tarkistussumma- ja aikatiedot muuttuvat. Myös RC2-rivin tiedot näyttävät nollaa, koska laitteeseen ei ole liitetty mitään mitattavia signaaleja.

MTK-komentoja käytetään MediaTekin valmistamien GPS-paikannuspiirien konfigurointiin. Komentojen muoto on lähes samanlainen kuin NMEA-viesteissä eli ne alkavat $-merkillä, jonka jälkeen tulee seitsemän merkkiä. Näistä merkeistä ensimmäiset neljä kertoo, että komento on MTK-komento ja loput merkit kertovat komennon numeron. Näiden seitsemän merkin jälkeen tulee komennon sisältö ja sitä seuraa *-merkki ja tarkistussumma heksalukuna. MTK-komennoilla voidaan vaihtaa esimerkiksi GPS-moduulin paikannustietojen lähetysnopeutta (baudrate). Muita konfigurointeja on moduulin lähettämien NMEA-rivien valinta, lähetyksen päivitystaajuus, moduulin

resetointi tehdasasetuksille, moduulin oletusarvojen uudelleenkirjoitus ja tallennettujen sen hetkisten asetusten luku. Alla on RaceDAC-laitteen päivitetystä ohjelmasta otettu esimerkki MTK-komentojen lähetyksestä. MTK-komentojen lähetysrivit on lihavoituna.

```
if(U2STAbits.URXDA == 1 || U2STAbits.OERR == 1)
{
        IEC1bits.U2RXIE =0;
        putsUART2("$PMTK251,115200*1F\r\n");  //baud change to 115200
        __delay_ms(500);
        CloseUART2();
        __delay_ms(500);
        OpenUART2(U2MODEVALUE, U2STAVALUE, U2BAUDRATE);
        __delay_ms(300);
        putsUART2("$PMTK220,100*2F\r\n");  //update rate to 10Hz
        putsUART2("$PMTK314,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*28\r\n");
        IEC1bits.U2RXIE = 1;
        IFS1bits.U2RXIF = 0;
        U2STAbits.OERR = 0;
        return 1;
}
else return 0;
```

Ensimmäisenä annettiin komento, joka muutti GPS-moduulin lähetysbaudraten suuremmaksi. Tämän jälkeen mikrokontrollerin UART2-moduuli suljettiin ja avattiin uudelleen muutetulla baudratella. Kun baudrate on muutettu, GPS-moduulille voidaan antaa komento, joka muuttaa sen päivitystaajudeksi 10 Hz. Tämän jälkeen koodissa on vielä valittu GPS-moduulin lähetettäviksi NMEA-riveiksi vain GGA- ja RMC-rivit.

# 4 KIIHTYVYYSANTURIN KÄYTTÖ LAITTEESSA

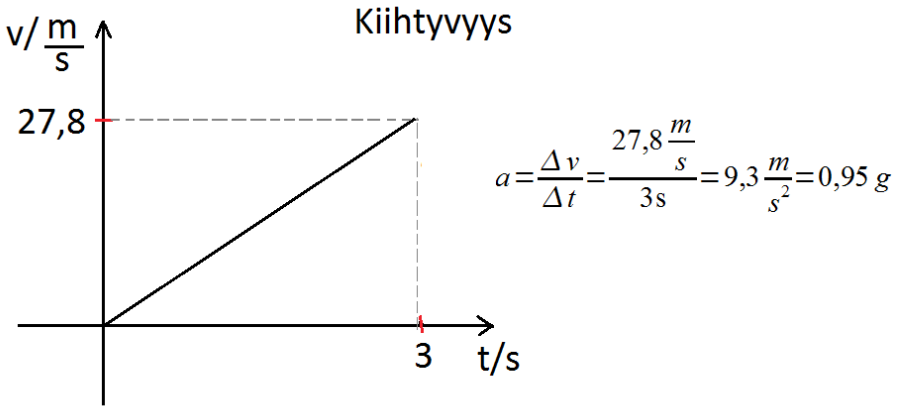
## *4.1 Valintaperusteet*

Valittaessa sopivaa kiihtyvyysanturia RaceDAC-laitteeseen täytyi miettiä mihin sitä

käytettäisi. Moottoriurheilussa käyttökohteina voi olla esimerkiksi kiihtyvyyden ja nopeuden mittaaminen. Joissakin tapauksissa myös asennon mittaus saattaa olla haluttu mittaustieto. Moottoripyörää ohjataan kallistamalla ajoneuvoa ja voisi olla mielekästä analysoida näitä tietoja, jos RaceDAC-laite kiinnitetään moottoripyörään. Jos asennon mittaustietoja halutaan luotettavasti käyttää, se tuo rajoituksia laitteen asennukseen ja kiinnitykseen. Laitteen täytyy pysyä samassa kohdassa liikkumatta, jotta mittaustulosta voidaan pitää riittävän tarkkana.

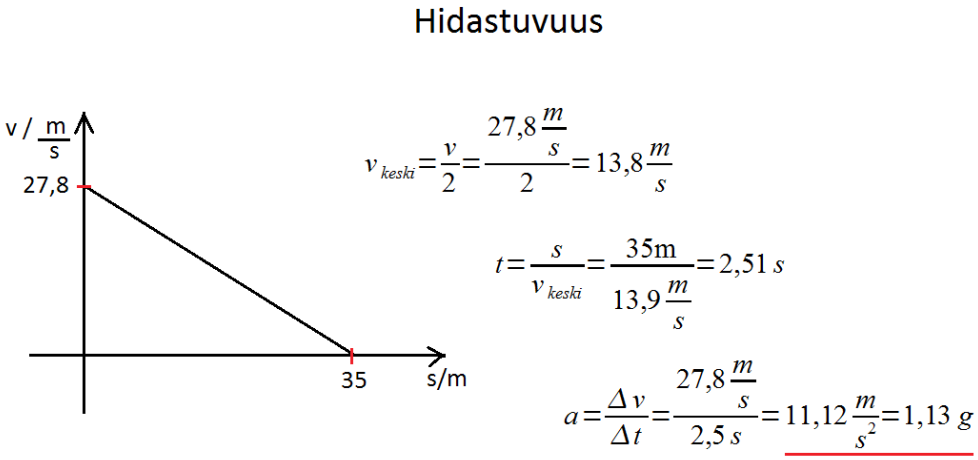
Moottoriajoneuvojen huippunopeudet jäävät melko varmasti alle 300 Km/h, joten nopeus tuskin on tässä tapauksessa rajoite kiihtyvyysanturin valinnassa. Kiihdytyskilpailuissa käytettyjen kiihdytysajoneuvojen maksimikiihtyvyys voi olla jopa 5 g:tä. RaceDAC-laitteen on kuitenkin tarkoitus olla käytössä tavallisissa rata-autoissa, moottoripyörissä ja veneissä. Näissä suurin kiihtyvyys lienee moottoripyörissä ja se voi olla suoralla kiihdytettäessä noin 1 g. Kaarteissa voimat ovat maksimissaan samaa luokkaa kuin suoralla kiihdytettäessä. Myös hidastuvuus nykyisillä moottoripyörillä ja autoilla on hieman yli 1 g.

Kuvioissa 6, 7 ja 8 näkyy laskukaavat, joilla perustellaan edellisiä kiihtyvyyksiä. Laskuissa on pyritty saamaan suurimmat mahdolliset arvot, jotta niitä voisi pitää rajaarvoina kiihtyvyysanturin valinnassa. Kaavoissa on myös pidetty kiihtyvyyttä aina tasaisena, joka ei vastaa todellisuutta.




KUVIO 6. Ajoneuvon kiihtyvyys suoralla tiellä

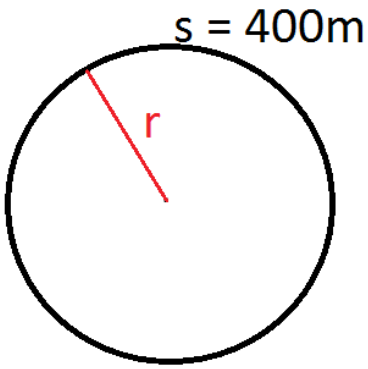
Kuviossa 6 on laskettu ajoneuvon kiihtyvyys suoralla tiellä noupeudesta 0 Km/h nopeuteen 100 Km/h (27,8 m/s). Nykyiset moottoripyörät kiihtyvät 0-100 Km/h noin 3 sekuntiin, joten kiihtyvyydeksi on saatu alle 1 g. Hidastuvuus lasketaan täsmälleen samalla tavalla. Kuvio 7 esittää hidastuvuuden laskemisen, kun jarrutusmatka on tiedossa. Jarrutusmatka nykyisillä moottoriajoneuvoilla on 35 metrin luokkaa.




KUVIO 7. Ajoneuvon hidastuvuus jarrutuksessa

Moottoripyörä pysähtyy 100-0 Km/h noin 35 metrin matkalla, joten pysähtymisaika on n. 2,5s. Hidastuvuudeksi saadaan laskettua 1,13 g kuten kuviossa 7 näkyy. Hidastuvuus kasvaisi todella suureksi, jos ajoneuvo törmäisi seinään. Tätä ei tarvitse kuitenkaan ottaa huomioon, koska RaceDAC-laitetta ei ole tarkoitettu toimimaan sellaisissa tilanteissa.

Ajoneuvossa tunnetaan kiihtyvyyttä myös, kun ajetaan kaarteeseen. Kyseisen kiihtyvyyden aiheuttaa keskipakoisvoima ja Kuvio 8 esittää keskipakoisvoiman aiheuttamaa kiihtyvyyttä ja sen laskemista.




$$r = \frac{s}{2\pi} = \frac{400\text{m}}{2\pi} = 63{,}7\,m$$

$$a = \frac{v^2}{r} = \frac{(27{,}8\,\frac{m}{s})^2}{63{,}7\,m} = 12{,}13\,\frac{m}{s^2} = 1{,}24\,g$$

KUVIO 8. Ajoneuvon kiihtyvyys kaarteessa

Kuviossa 8 on laskettu ajoneuvvoon keskipakoisvoiman aiheuttama kiihtyvyys. Kiihtyvyys on laskettu tarkoituksella mahdottomilla arvoilla, jotta saataisiin käsitys kaarteissa ilmenevistä kiihtyvyyksistä. Käytännössä ei olisi mahdollista ajaa ympyrärataa 100 Km/h, kun radan pituus on 400m.

Laskuesimerkeissä kiihtyvyyksien on oletettu olevan täysin tasaisia. Näin ei kuitenkaan todellisuudessa ole, joten ajoneuvon hetkellinen kiihtyvyys vaihtelee ja saattaa olla paljon suurempi, kuin esimerkkilaskuissa. Tästä syystä kiihtyvyysanturiksi ei kannata valita anturia, jonka maksimikiihtyvyys on 1,5 g. Riittävä maksimikiihtyvyys voisi olla 2 g:tä. Akseleiden määrä valittavassa anturissa tulee olla kolme, jotta olisi mahdollista saada tietoa laitteen ja ajoneuvon asennosta. Päivitystaajuus ei tässä sovelluksessa tule olemaan valinnan kannalta ratkaiseva, koska lähes kaikissa antureissa näyttää olevan säädettävä ulostulotaajuus. RaceDAC-laitteessa taajuudeksi riittäisi 50 Hz, koska myöskään muut laajennuskortit eivät kykene logirivien lähettämiseen/tallentamiseen sen nopeammin. Herkkyyden valinnassa voidaan ajatella, että riittävä tarkkuus kiihtyvyysanturille olisi silloin, kun anturi kykenee antamaan 150 eri arvoa kun kiihtyvyys muuttuu 0-1 g. Tämä tarkkuus on riittävä siksi, että RaceDAC-laite kerää tietoja 50 Hz:n taajuudella. Moottoripyörällä kiihdytettäessä 0-100 Km/h aikaa menee 3s ja RaceDac-laite kerkeää muodostamaan ja lähettämään 150 logiriviä. RaceDAC-laitteessa kannattaa käyttää digitaalista kiihtyvyysanturia, koska laitteen mikrokontrollerin ADC-moduuli pystyy vain 10 bitin tarkkuuteen ja silloin analogisen

anturin herkkyys tulisi olla 500mV/g. Nämä anturit eivät enää ole edullisimmassa hintaluokassa, joten kannattaa valita digitaalinen kiihtyvyysanturi. Digitaalisessa anturissa tarkkuudeksi sopisi 6mV/digit, koska 1g / 150 = 6,6mV.

## *4.2 MMA8450Q- ja LIS331DLH-kiihtyvyysanturit*

Tässä kappaleessa kerrotut anturien ominaisuudet on löydetty niiden datalehdiltä. Freescalen valmistama MMA8450Q-kiihtyvyysanturi on edullisin Farnellin varastosta löytyvä 3-akselinen ja riittävän tarkka anturi käytettäväksi RaceDaC-laitteessa. Taulukossa 1 näkyy kyseisen anturin tässä sovelluksessa merkittäviä ominaisuuksia. Taulukko 2 puolestaan esittelee tuplasti kalliimman STMicroelectronicsin valmistaman LIS331DLH-anturin ominaisuuksia. Vertailua ei tehdä analogisen ja digitaalisen anturin välillä, koska opinnäytetyössä päädyttiin jo aiemmin kiihtyvyysanturien yleisten ominaisuuksien tutkimisen aikana digitaaliseen anturiin. Tästä syystä valittiin siis kaksi melko edullista, mutta hintaeroltaan erilaista digitaalista anturia vertailuun.

TAULUKKO 1. MMA8450Q-kiihtyvyysanturin ominaisuuksia

| **Ominaisuus** | **Arvo** | **Lisätietoja** |
|---|---|---|
| Herkkyys | 0,98 mg/digit<br>1,953 mg/digit<br>3,906 mg/digit | ±2g kiihtyvyysalue<br>±4g kiihtyvyysalue<br>±8g kiihtyvyysalue |
| Kiihtyvyysalue | ±2g<br>±4g<br>±8g | Valittavissa |
| Liitäntä | I2C | |
| Resoluutio | 8 bittiä<br>12 bittiä | Valittavissa |
| Päivitystaajuus | 1,563-400Hz | Valittavissa |
| Kohina | 375 µV/√Hz | |

TAULUKKO 2. LIS331DLH-kiihtyvyysanturin ominaisuuksia

| **Ominaisuus** | **Arvo** | **Lisätietoja** |
|---|---|---|
| Herkkyys | 1 mg/digit<br>2 mg/digit<br>3,9 mg/digit | ±2g kiihtyvyysalue, 12 bittinen ulostulo<br>±4g kiihtyvyysalue, 12 bittinen ulostulo<br>±8g kiihtyvyysalue, 12 bittinen ulostulo |
| Kiihtyvyysalue | ±2g<br>±4g<br>±8g | Valittavissa |
| Liitäntä | I2C ja SPI | Valittavissa |
| Resoluutio | 16 bittiä | |
| Päivitystaajuus | 0,5-1000Hz | Valittavissa |
| Kohina | 218 µV/√Hz | |

Molemmat yllä esitellyistä antureista sopisivat ominaisuuksiensa puolesta käytettäväksi RaceDAC-laitteessa. MMA8450Q-kiihtyvyysanturin hinta yksittäiskappaleena tilattuna Farnellilta oli 2,86€. LIS331DLH-anturi oli hinnaltaan 5,36€. Jälkimmäinen anturi on kalliimpi hieman parempien ja laajemmin valittavissa olevien ominaisuuksiensa vuoksi. Molemmat anturit ovat digitaalisia ja liitettävissä I2C-väylään. Kalliimmassa anturissa liitäntämahdollisuutena on myös SPI-väylä. Molemmat näistä liitännöistä ovat mahdollisia RaceDACissa käytettäväksi, koska mikrokontrollerin kahdesta SPI-moduulista on käytössä tällä hetkellä vain toinen. Kalliimpi anturi tarjoaa myös 16-bittisen ulostulon, mutta opinnäytetyön sovelluksessa 12 bittiä riittää hyvin. 12-bittinen ulostulo pystyy näyttämään 4096 eri arvoa, joka ±2 g:n kiihtyvyysalueella tarkoittaa 1024 arvoa yhtä g:tä kohti. RaceDAC-laitteessa 50Hz tiedonkeruutaajuudella 1g:n kiihtyvyyden muutoksessa keretään saada noin 150 arvoa, joten molempien antureiden pitäisi tämänkin puolesta sopia käytettäväksi laitteessa. Päivitystaajuus on riittävä jo halvemmassa anturissa. Jo 100 Hz:n taajuus olisi luultavasti riittävä tässä sovelluksessa.

Kahden hintaluokan antureiden suurimmat erot olivat päivitystaajuudessa. Tämä ominaisuus on kuitenkin tärkeämpi sovelluksissa joissa mitataan esimerkiksi moottorin tärinää tai muita nopeasti vaihtelevia liikkeitä. Kohinan osalta kalliimpi anturi oli myös

jonkin verran parempi. RaceDAC-laitteessa käytettäväksi anturiksi voi kuitenkin hyvin valita pienemmän hintaluokan anturin sen riittävän hyvien ominaisuuksien vuoksi.

# 5 TOTEUTUS

## *5.1 Laitteiston suunnittelu*

### 5.1.1 SD-muistikortti

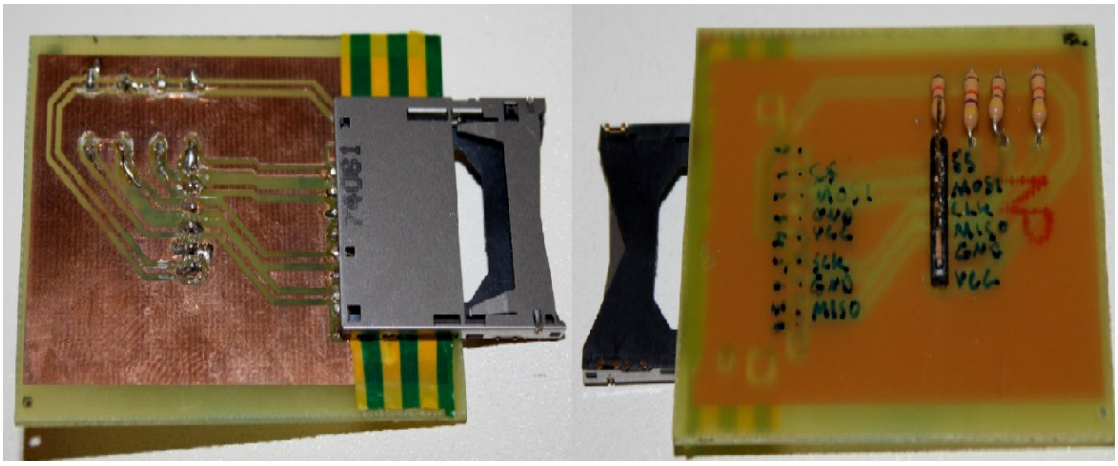
Opinnäytetyössä tärkeimpänä laajennuksena toteutettiin tuki ulkoiselle SD-muistikortille. Kaikki moottorinohjausyksiköltä tulevat anturitiedot oli tarkoitus tallentaa muistikortille silloin, kun laajennuskortti on kiinnitetty laitteeseen. Muulloin anturien tiedot lähetetään bluetoothin kautta eteenpäin niin kuin ennenkin. Muistikortin lukuun ja kirjoitukseen käytettiin SPI-väylää. SPI-väylään tuli opinnäytetyössä vain yksi slave-laite (SD-muistikortti), joten SPI-väylä oltaisiin voitu minimissään toteuttaa kolmea mikrokontrollerin pinniä käyttämällä. SS-signaalia ei välttämättä tarvita, jos väylällä on vain yksi laite. Slave-laitteen SS-pinnin voi maadoittaa, jolloin kyseinen laite on aina valittuna väylällä. Päädyttiin kuitenkin käyttämään neljän pinnin konfiguraatiota, koska jotkin SD-muistikortit tarvitsevat valitsemisvaiheessa SS-signaalin tilanvaihdon, ennen kuin laite tietää olevansa valittuna väylällä.

Opinnäytetyössä muistikortin tiedostojärjestelmäksi valittiin FAT16. Microchipin kotisivuilta löytyy kirjasto, jossa on tuki myös FAT32-tiedostojärjestelmään, mutta ohjelmakoodin pitämiseksi mahdollisimman pienenä tuki FAT32:een karsittiin ajurista pois. Kirjasto ei tarjonnut suoraan tukea RaceDAC-laitteessa olevalle dspic64gp204-mikrokontrollerille, joten ensimmäinen tehtävä oli saada käännettyä kirjasto kyseiselle

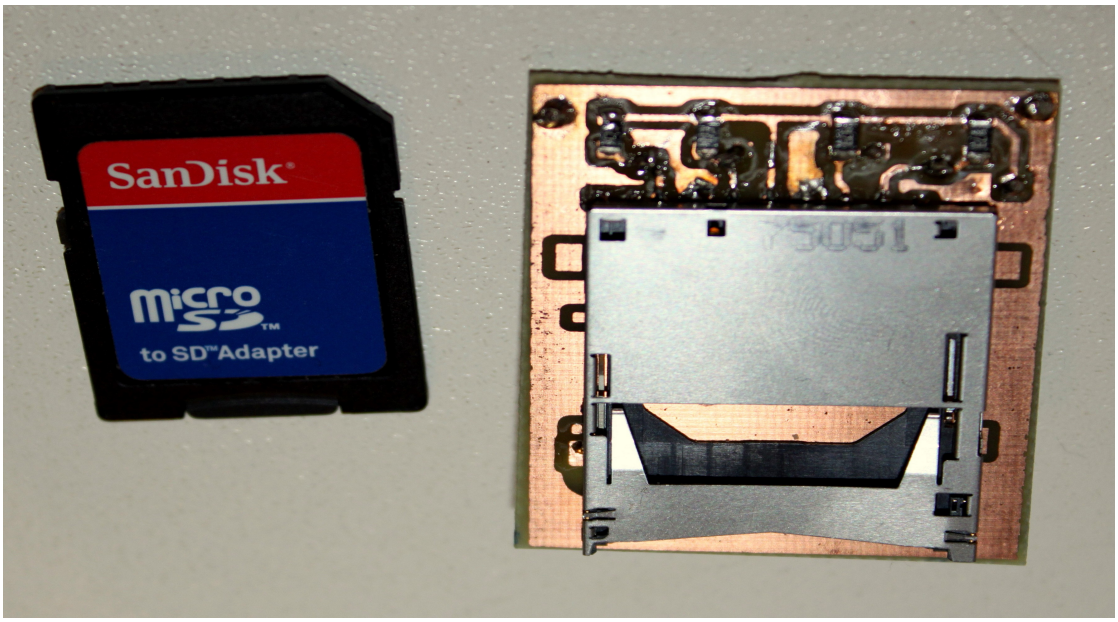
kontrollerille. Pieniä ongelmia aiheutti se, että mikrokontrollerissa ei ollut valmiita pinnejä SPI-väylälle, vaan kaikki kontrollerin pinnit on asetettavissa mihin tehtävään tahansa (remappable pheripheral). Ensin täytyi ottaa selvää siitä, mitä tulee ottaa huomioon, kun asetetaan tietyt pinnit jonkin moduulin käyttöön. Microchipin kotisivuilta löytyy dsPIC33F Family Reference Manual, jonka luku 30 käsittelee aihetta. (Microchip Technology inc. 2010)

Kun pinnit saatiin asetettua SPI-väylän signaaleille oikeiksi, uudeksi ongelmaksi muodostui SPI-väylän alustaminen. Kellopulssi ei näkynyt oskilloskoopilla ollenkaan, eikä tieto väylällä liikkunut. Kääntäjä ei löytänyt virhettä, joka oli mikrokontrollerin lukkobitti pinnien ominaisuuksien uudelleenasettamiselle. Lukkobitin tilaa piti siis vaihtaa, ennen kuin pinnien asettaminen uuteen tehtävään onnistui. Tämänkään jälkeen kellopulssia ei näkynyt, mutta se johtui mikrokontrollerin sisäisen kellotaajuuden asetusrekisterien arvoista. Datalehteä lukemalla valittiin sopivat arvot rekistereille ja saatiin data väylällä kulkemaan.

Tässä vaiheessa oli saatu SD-muistikortin käsittelyyn tarvittava kirjasto käännettyä ja SPI-väylän asetukset oikeiksi käytössä olevalle mikrokontrollerille. Toimeksiantaja antoi aloituspalaverissa käyttöön pienen kehitysalustan dspic33fj64gp204-kontrollerilla varustettuna, ohjelmointilaitteen kyseiselle mikro-ohjaimelle ja Arduinon kanssa yhteensopivan laajennuslevyn SD-muistille. Kirjaston toimintaa kokeiltiin ensin liittämällä laajennuslevy kehitysalustaan ja lataamalla käännetty ohjelma kehitysalustalle. Muistikortille ei kuitenkaan pystynyt tallentamaan mitään vaikka ohjelman oli tarkoitus luoda uusi tiedosto ja kirjoittaa rivi tekstiä siihen jokaisella käynnistyskerralla. Tämä ongelma johtui siitä, että Arduinoon tarkoitetut laajennuskortit toimivat 5 V:n jännitteellä ja kehitysalustassa oleva mikrokontrolleri antaa IO-pinneistään ulos vain 3,3 V. Arduinon SD-muistikorttilevyllä oli SD-muistikortille menevillä signaalijohtimilla etuvastukset, jotka pienensivät jännitettä niin paljon, että kortille kirjoitus ja luku eivät onnistuneet. Täytyi suunnitella oma piirilevy, jotta toiminta voitiin testata. Levy toimi suoraan 3,3 V:n jännitteellä ja siinä oli paikka SD-muistikortille. Itse tehty piirikortti SD-muistikortin testaamiseen näkyy kuviossa 9. Kun uusi levy kiinnitettiin kehitysalustaan, muistikortille kirjoitus ja sieltä lukeminen alkoivat toimia. Kuviossa 10 on vertailun vuoksi kuva valmiista prototyypistä.



KUVIO 9. Itse tehty piirikortti SD-muistikortin testaamiseen




KUVIO 10. SD-muistikorttilaajennuksen prototyyppi ja SD-muistikortti

### 5.1.2 GPS-moduuli

GPS-moduuliksi opinnäytetyössä valittiin Globaltop FGPMMOPA6B-moduuli. Kyseinen GPS-moduuli kykenee lähettämään paikannustietoa 10 Hz:n taajuudella, joka oli vähimmäisvaatimus opinnäytetyöhön suunniteltavassa laajennuskortissa. PA6B-moduuli oli myös kaikkein edullisin 10 Hz:n päivitystaajuuden omaavista vaihtoehdoista.

Suunnittelu alkoi testaamalla moduulin toiminta siitä löytyneellä USB-rajapinnalla. USB:n kautta paikannustiedot saatiin näkyviin heti, joten alettiin kokeilla moduulin sarjaliikennerajapintaa, jolla se yhdistetään RaceDAC-laitteeseen. Sarjaliikenneväylän kokeilu tehtiin ensin dspic33fj64gp204-mikrokontrollerin sisältävällä demokortilla, jotta testiohjelmasta saatiin tarpeeksi yksinkertainen mahdollisten virheiden löytämiseksi ja korjaamiseksi.

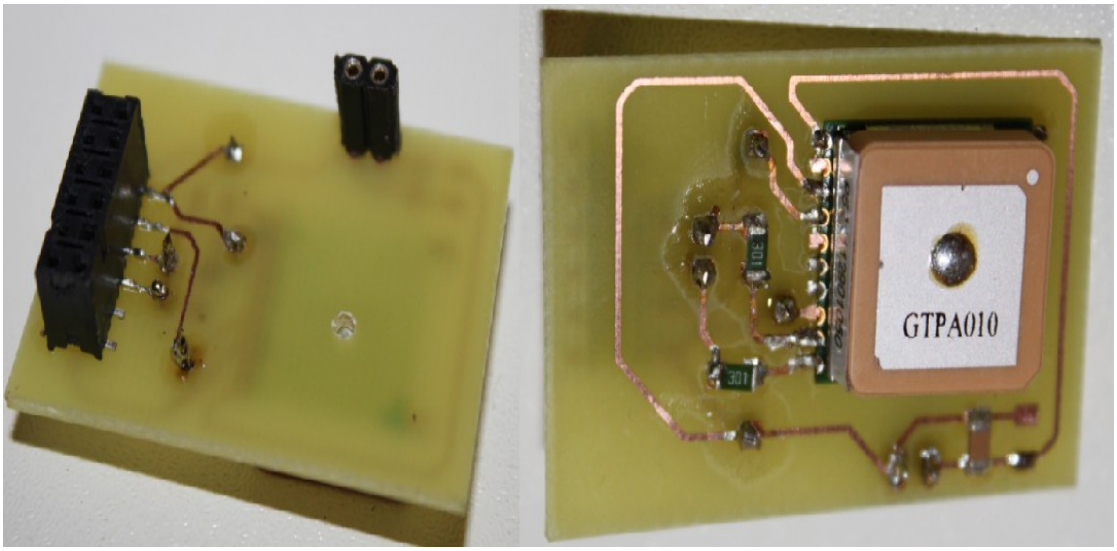
Moduulin lukeminen UART-väylän kautta vaati mikrokontrollerin UART-väylän alustamisen ja keskeytysaliohjelman kirjoituksen. UART-väylä alustettiin GPS-moduulin tehdasasetuksia vastaaviksi: tiedonsiirtonopeus 9600 bps, 8 data bittiä, 1 lopetusbitti ja ilman varmistusbittiä. Asetusten jälkeen GPS-moduulin lähettämät paikannustietorivit eivät näyttäneet oikeilta. Vikaa etsittiin ohjelmasta ja kytkennässä käytetyistä oheiskomponenteista. Kun ongelmaan ei löytynyt ratkaisua, kokeiltiin moduuli liittää Arduino Duemilanove-kehitysalustaan GPS-moduulin oikean toiminnan varmistamiseksi. Arduino-kehitysalustalla luettuna paikannustietorivit näyttivät kuitenkin aivan oikeilta, joten vian täytyi johtua dspic-mikrokontrollerin UART-väylän rekisterien virheellisistä asetuksista. Lopulta vika löytyikin UART-väylän tiedonsiirtonopeuden asetusrekisterin väärästä arvosta. Testiohjelmassa kyseisen rekisterin arvo oli laskettu kokoinaislukuja käyttäen, jolloin mahdollinen jakojäännös jää huomioimatta kokonaan. Rekisterin arvo laskettiin mikrokontrollerin datalehdeltä löytyneellä kaavalla:

$$rekisterin\ arvo = \frac{FCY}{16 \times baudrate} - 1 = \frac{3685000}{16 \times 9600} - 1 = 22{,}99$$

FCY tarkoittaa mikrokontrollerin kellotaajuutta, joka kyseisessä testiohjelmassa oli 3,685 Mhz. Ohjelmassa rekisterin arvoksi tuli 22, koska kokonaisluvuilla laskettaessa jakojäännös jäi kokonaan pois. Tästä tuli niin suuri virhe tiedonsiirtonopeuteen, että mikrokontrollerin lukemat merkit eivät olleet oikeita. Ongelma saatiin ratkaistua asettamalla rekisterin arvoksi suoraan 23, joka on riittävän lähellä 22,99:ää.

Kun tiedonsiirto GPS-moduulin ja dspic-kontrollerin välillä saatiin toimimaan, täytyi

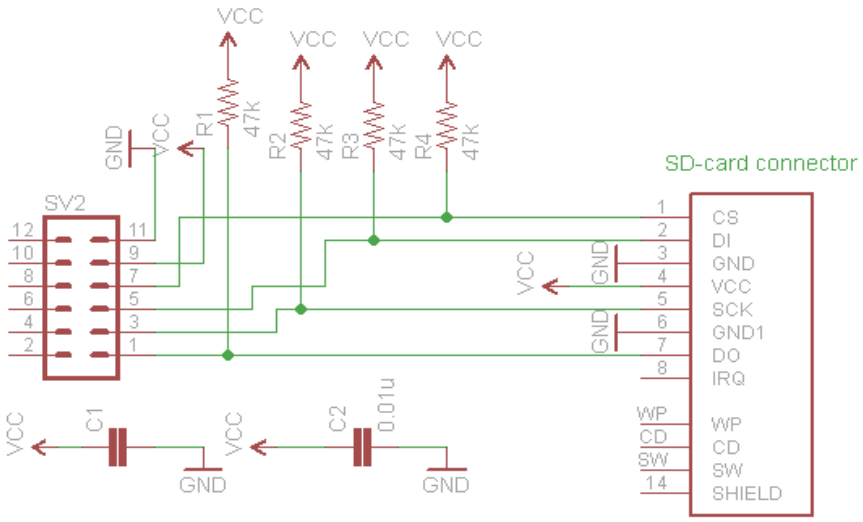
saada luettua talteen GPS-moduulin lähettämät rivit. Opinnäytetyössä ei tarvittu läheskään kaikkia datarivejä, jotka moduuli lähetti sarjaportin kautta mikrokontrollerille. Tässä vaiheessa kirjoitettiin keskeytysaliohjelma, joka tarkasti jokaisen vastaanotetun rivin ja tallensi sen erilliseen muuttujaan, jos rivi oli yksi halutuista. Myöhemmin kyseisen rivien suodatuksen todettiin olevan tarpeetonta, kun alettiin konfiguroida GPS-moduulia toimimaan vaaditulla 10 Hz:n päivitys-taajuudella. Moduulia oli mahdollista konfiguroida omiin tarpeisiin sopivaksi lähettämällä sille MTK-komentoja UART-väylän kautta. Etsittäessä oikeaa komentoa taajuuden muuttamiseksi löytyi myös komento, jolla pystyi valitsemaan, mitä NMEA-rivejä moduulin haluttiin lähettävän. Kuviossa 11 on kuva valmiista GPS-laajennuskortin prototyyppilevystä.




KUVIO 11. GPS-laajennuskortin prototyyppi ala- ja yläpuolelta

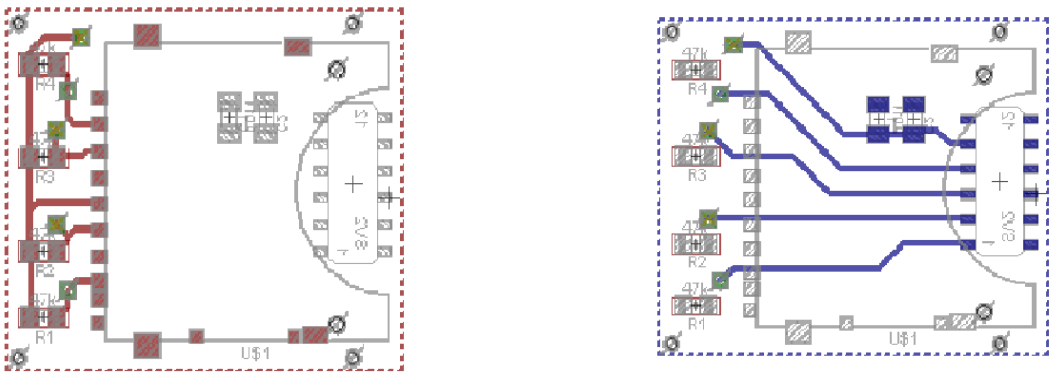
### 5.1.3 Piirilevysuunnittelu

Lopullinen prototyyppilevy SD-muistikorttilaajennuksesta tehtiin sitten, kun testikortin avulla oli saatu ohjelmistopäivitys tehtyä RaceDAC-laitteeseen. Lopullinen levy erosi testilevystä lähinnä kooltaan ja tarkemmin mietityltä pinnijärjestykseltään. Kuviossa 12 näkyy lopullisen laajennuskortin piirikaavio.



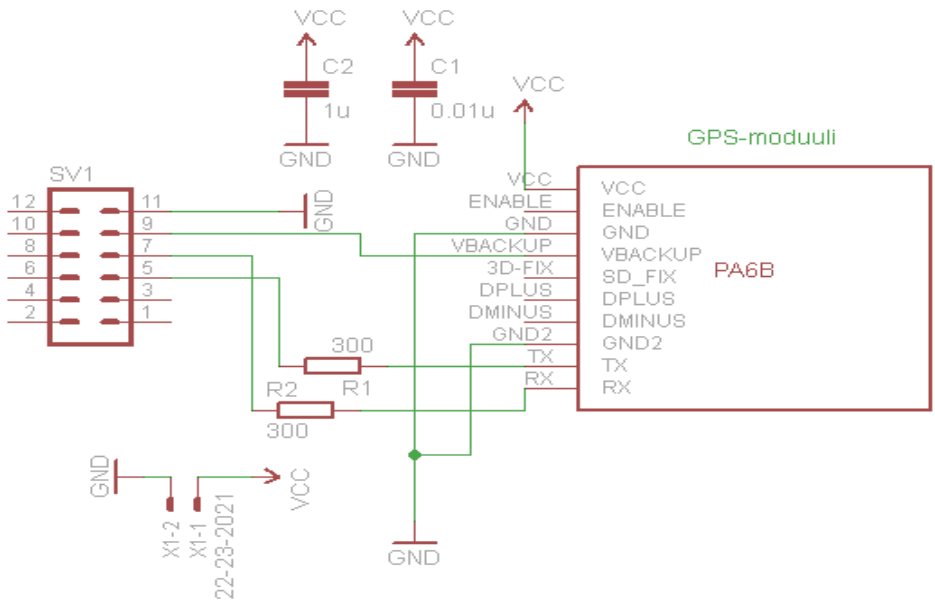

KUVIO 12. SD-muistikorttilaajennuksen prototyypin piirikaavio

Prototyypin layout on suunniteltu siten, että piirilevy on mahdollista valmistaa syövyttämällä. Tarkoitus oli, että lopullinen layout sisältäisi molemmat laajennukset ja ladonnalla valittaisiin kumpi laajennus otetaan käyttöön. Tämä vähentää kustannuksia, kun levyt tilataan valmistajalta. Opinnäytetyössä tehtiin kuitenkin molemmille laajennuksille omat layoutit, koska lopullista piirilevyä tuskin olisi voinut valmistaa itse syövyttämällä. Prototyyppilevyn layoutin suunnittelua rajoitti se, että syövyttämällä tehdyssä levyssä tulee johdinpaksuuden ja komponenteille tarkoitettujen padien olla paljon suurempi kuin piirilevyvalmistajalta tilatuissa levyissä. Myös komponenttien asettelu on rajoitetumpaa, jotta johdinvetojen välit syöpyvät kunnolla, eikä niihin jää oikosulkuja. Kuviossa 13 on esitetty muistikorttilaajennuslevyn layout ylä- ja alapuolelta.

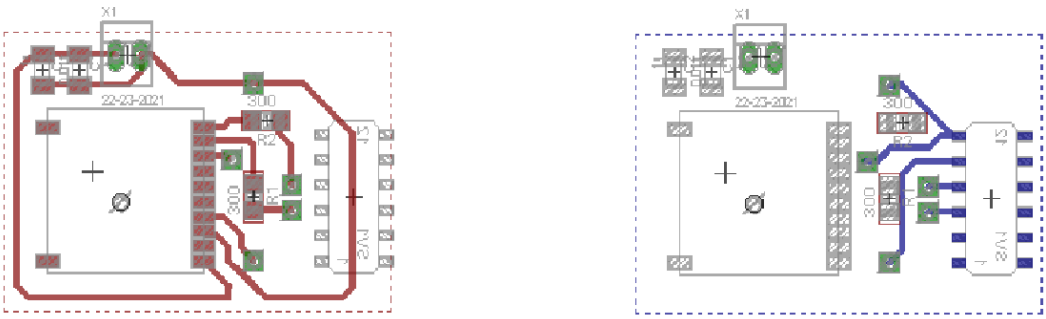


KUVIO 13. Muistikorttilaajennuslevyn layout

GPS-laajennuskortin piirilevysuunnittelua rajoittivat samat asiat kuin muistikorttilaajennuslevyn suunnitteluakin. GPS-moduulin datalehdeltä löytyi esimerkkipiirikaavio, jolla mooduulin sai otettua käyttöön. Piirikaaviossa tarvittiin vain vastukset TX- ja RX-linjoille ja kaksi häiriönpoistokondensaattoria käyttöjännitelinjalle. Datalehdellä oli myös muutamia rajoituksia layoutin suunnitteluun. Moduulin alla tuli piirilevyssä olla reikä siihen integroidulle antennille ja suoraan moduulin alapuolella ei saanut olla signaalijohtimia eikä läpivientejä. Nämä rajoitukset eivät suuresti hankaloittaneet layoutin suunnittelua, koska kytkentään käytettävien komponenttien määrä oli hyvin pieni siihen tilaan nähden, joka oli varattu laajennuskortin käyttöön RaceDAC-laitteessa. Kuviossa 14 näkyy GPS-laajennuskortin piirikaavio ja kuviossa 15 sen layout ala- ja yläpuolelta.




KUVIO 14. GPS-laajennuskortin piirikaavio



KUVIO 15. GPS-laajennuskortin layout

## 5.2 Ohjelmistosuunnittelu

### 5.2.1 Alkutilanne

RaceDAC-laitteen alkuperäinen ohjelma toimi siten, että käynnistyessään laite lukee EEPROM-muistista asetukset, joilla laite oli toiminnassa edellisellä käynnistyskerralla. Tämän jälkeen ohjelmassa alustettiin tarvittavat moduulit kuten UART1 bluetooth-datan siirtoa varten, timerit tiedonkeräystaajuuden laskemiseksi, ADC-moduuli AD-muunnoksiin ja I2C-väylä EEPROM-muistin lukuun ja kirjoitukseen.

Itse pääsilmukka ohjelmassa luki jokaisella ajokerralla ensin digitaali-sisääntulokanavat, jonka jälkeen luettiin mahdolliset bluetooth-yhteyden kautta tulleet komennot. Seuraavaksi luettiin analogikanavien arvot ja sitten muodostettiin logirivi, joka viimeiseksi lähetettiin bluetoothin kautta eteenpäin. Pääsilmukassa hoidettiin myös merkkiledin tilanvaihdot ja laitteen sammutusnapin pohjassaoloajan luku.

### 5.2.2 Tuki SD-muistikortille

Kun muistikortille saatiin luotua tiedosto ja kirjoitettua siihen tekstiä demokorttiin liitettynä, alkoi varsinaisen ohjelmistopäivityksen kehitys RaceDAC-laitteen ohjelmaan. Aluksi oli tarpeen karsia SD-muistikorttituen sisältävästä kirjastosta kaikki tässä laitteessa tarpeettomat ominaisuudet pois. Tällaisia toimintoja kirjastossa olivat tuki FAT32-tiedostojärjestelmälle, pitkien tiedostonimien mahdollisuus, kansioiden luonti ja muistikortin uudelleen alustaminen. Seuraava tehtävä oli ohjelman toimintaperiaatteen selvitys, jonka jälkeen alettiin tehdä ohjelmistoon tukea SD-muistikortille.

RaceDAC-laitteessa olevaa bluetooth-ominaisuutta ei ollut tarkoitus käyttää logirivien lähetykseen silloin, kun laajennuskortti muistikorttiladonnalla oli kiinnitetty laitteeseen. Tällöin laitteen keräämät anturitiedot tallennettiin ainoastaan muistikortille.

Bluetooth-yhteys tuli kuitenkin olla toiminnassa, jos laitteelle haluttiin tehdä uusia asetuksia kuten esimerkiksi tiedonkeruutaajuuden vaihto. Ohjelman bluetooth-datan lähettämiseen tehdyssä aliohjelmassa oli valmiiksi muodostettu merkkijono, joka sisälsi kaikki tiedot, jotka oli tarkoitus tallentaa muistikortille. Tämä helpotti huomattavasti muistikorttituen ohjelmointia ja testausta, koska tietojen saattoi luottaa olevan oikein.

Microchipin kotisivuilta löytyvä kirjasto sisältää funktiot tiedostojärjestelmän alustamiseen, muistikortilta lukemiseen ja kirjoittamiseen. Tiedoston voi avata lukutilassa tai kirjoitustilassa. Opinnäytetyötä tehtäessä ilmeni ongelma saman tiedoston avaamisessa peräkkäin ensin lukutilassa ja sitten kirjoitustilassa. Tiedostosta lukemisen jälkeen tiedoston avaus täytyi tehdä useita kertoja, kun haluttiin avata sama tiedosto kirjoitusta varten. Molempien operaatioiden jälkeen tiedosto oli oikeaoppisesti suljettu, joten tiedoston avaamisen olisi pitänyt onnistua normaalisti. Ongelma ratkaistiin avaamalla tiedostoa silmukassa niin monta kertaa, että se onnistui. Myös logitiedostoon kirjoittaminen alkoi vasta neljänneltä riviltä ennen kuin tiedosto avattiin silmukassa.

Jos ohjelmassa on havaittu SD-muistikortin olevan liitettynä laitteeseen, ohjelma kokeilee avata RDAC.ini -tiedoston muistikortilta. Tiedoston löytyessä ja avautuessa normaalisti, tiedostosta luetaan sen hetkisen olemassa olevan logitiedoston järjestysnumero. Jos tiedostoa taas ei löydy, se luodaan ja sen jälkeen oletetaan ettei muistikortilla ole olemassa vielä yhtään logitiedostoa. Laitteen on jokaisella käynnistyskerralla tarkoitus luoda uusi logitiedosto, jonka nimi on muotoa Rdxx.txt. X-kirjainten tilalla on siis aina tiedoston järjestysnumero. Alla on ohjelmistopäivityksen yhteydessä tehty koodi ini-tiedoston luontiin, lukuun ja päivittämiseen.

```
if(ExtensionCardID ==1)  //read ini file
{
                int counter = 0;
                fileNumPtr = &fileNum[0];//&myData;
                pIniFile = FSfopen(iniFileName, "r+"); //try to open existing ini file
        if(pIniFile == NULL) //file does not exist, create one
            {
                FSfclose(pIniFile); //make sure that file is closed
                fileNumber = 1;
                ileNumPtr = Uint_to_numArr(fileNumber, fileNumPtr); //convert to char array
                /*create file name for first file on memorycard*/
```

```
        logFileName[0] = 'R';
        logFileName[1] = 'D';
        while(fileNum[counter] != '!')
        {
        logFileName[counter + 2] = fileNum[counter];
        counter++;
        }

        logFileName[counter + 2] = '.';
        logFileName[counter + 3] = 't';
        logFileName[counter + 4] = 'x';
        logFileName[counter + 5] = 't';

        pIniFile = FSfopen(iniFileName, "W"); //create inifile and open it in write mode
        if (pIniFile != NULL) //file creating succeed?
        {
        //write fileNumber to ini file so that it can be readlater and new log file can be created
          FSfwrite ((void *)fileNum,1,9,pIniFile);
        }
        FSfclose(pIniFile); //close the file
}
else //ini file exists, read latest file number and create name for new log file
{
        FSfread( (void *)fileNum, 1,5, pIniFile ); //read file number
        FSfclose (pIniFile);
        fileNumber = cTableToUint(fileNum); //convert char array to unsigned int
        fileNumber++;
        //file number for new file
        fileNumPtr = Uint_to_numArr(fileNumber, fileNumPtr); //convert to char array
         // open ini file for writing (file does not open first time for unknown reason)
        pIniFile = FSfopen(iniFileName, "W");
        while(pIniFile == NULL) // file must exist because data was recently read from it
        {
        // try to open it until opening is succesfull
        pIniFile = FSfopen(iniFileName, "W");
        }
        if(pIniFile != NULL)
        {
        FSfwrite ((void *)fileNum, 1,9,pIniFile);//write new file number to ini file
        }
        FSfclose (pIniFile);
        /*create filename for new log file*/

        logFileName[0] = 'R';
        logFileName[1] = 'D';
        while(fileNum[counter] != '!')
        {
           logFileName[counter + 2] = fileNum[counter];
           counter++;
        }

        logFileName[counter + 2] = '.';
```

```
                logFileName[counter + 3] = 't';
                logFileName[counter + 4] = 'x';
                logFileName[counter + 5] = 't';

                }

        }
```

### 5.2.3 Tuki GPS-moduulille

Kun valitun GPS-moduulin toiminta oli testattu PicKit3-demokortin avulla, alettiin kirjoittaa RaceDAC-laitteen ohjelmaan tukea GPS-laajennuskortille. Alussa törmättiin ongelmaan, kun yritettiin asettaa GPS-moduulin lukemiseen tarkoitetun UART2-väylän rekisteriarvoja kohdilleen. Sarjaliikenneväylän asetus ei tälläkään kertaa toiminut suoraan. Tämä johtui siitä, että opinnäytetyötä tehtäessä käytössä oli RaceDAC-laitteesta prototyyppi, jossa oli käytössä 12 Mhz:n kide. Ohjelmistossa oli asetukset prototyypille, joka toimi 11.0592 Mhz:n kiteellä. Uart-tiedonsiirto GPS-moduulin ja mikrokontrollerin välille saatiin toimimaan, kun ohjelmakoodiin tehtiin muutokset käytössä olevalle 12 Mhz:n kiteelle sopiviksi. Kun tiedonsiirto GPS-moduulin ja mikrokontrollerin välille saatiin toimimaan, oli moduulin lähettämää dataa karsittava, koska moduuli lähetti RaceChrono-sovellukselle tarpeettomia rivejä ja bluetooth-moduuli tuskin olisi kyennyt lähettämään kaikkia rivejä eteenpäin jokaisella lähetyskerralla. RaceChrono osaa käsitellä RMC- ja GGA-tiedot, jotka täytyi löytää muiden GPS-moduulin lähettämien rivien joukosta. Opinnäytetyössä päädyttiin ratkaisuun, jossa jokaisen kokonaisen uuden rivin lukemisen jälkeen tarkastettiin, onko se toinen halutuista riveistä. Tarkistus tehtiin UART2-moduulin keskeytysaliohjelmassa.

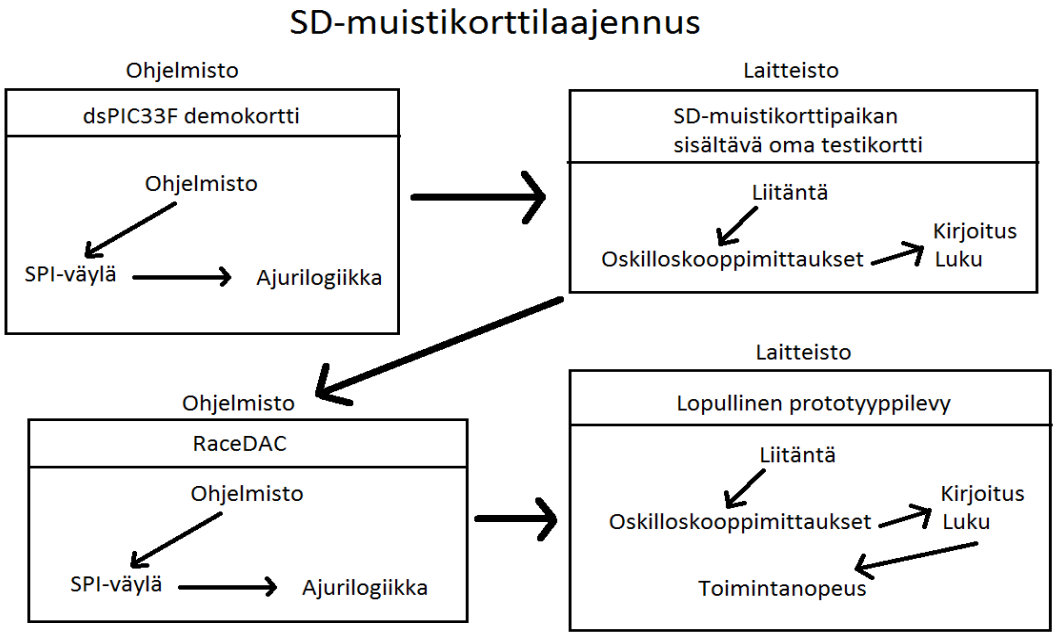
Tässä vaiheessa ohjelmistosuunnittelua täytyi alkaa huomioida se, että laitteen käynnistyessä täytyi selvittää, onko laitteeseen kiinitetty laajennuskortti vai ei. Opinnäytetyössä päädyttiin ratkaisuun, jossa kokeillaan järjestyksessä alustaa kaikki laajennukset. Ensimmäisenä alustettiin UART2-väylä ja koodissa asetettiin pieni viive, jonka aikana UART2-moduulin vastaanottopuskuriin on tullut dataa, jos laajennuskortti GPS-moduulilla on kiinnitettynä laitteeseen. Jos vastaanottopuskuri taas on tyhjä, kokeillaan alustaa RaceDAC-laitteen laajennuskortille tarkoitetut pinnit SPI-väylän käyttöön. Tämän jälkeen kokeillaan suorittaa FSInit-funktio, joka yrittää alustaa muistikortin. Kyseinen funktio palauttaa NULL-arvon, jos muistikorttia ei löydy ja ohjelmassa

oletetaan, että tällöin SD-muistikorttilaajennus ei ole kiinnitetty laitteeseen. Jos kumpikaan edellisistä kokeiluista ei onnistu, ohjelma toimii kuten ennen ohjelmistopäivitystä ja lähettää keräämänsä tiedot bluetoothin kautta puhelimeen tai tietokoneeseen.

# 6 TESTAUS

## *6.1 Testaussuunnitelma*

Opinnäytetyössä pyrittiin kehittämään laajennukset siten, että jokainen osuus jaettiin ensin pienempiin osiin ja testattiin niiden toiminta ennen käyttämistä varsinaisessa laajennuksessa. Kaikki ohjelmistopäivityksessä tarvittavat mikrokontrollerin ominaisuudet testattiin ensin demokortilla, johon oli juotettu kiinni sama mikrokontrolleri, kuin RaceDAC-laitteessa. Samaa käytäntöä pyrittiin käyttämään myös laitesuunnittelussa. Tällä tavalla tehtäessä jouduttiin testaamaan useaan kertaan samoja asioita, mutta jokaisen ominaisuuden toimimaan saanti oli helpompaa demokortilla, jolle kirjoitettu ohjelma oli yksinkertainen ja sitä kautta vikojen etsintä/korjaaminen olivat helpompia. Jos laajennuskortteja ja niiden ominaisuuksia olisi lähdetty kehittämään suoraan RaceDAC-laitteeseen, olisi vastaan tulleiden ongelmien etsintä ja ratkominen ollut työlästä ja aikaa vievää. Kuvio 16 esittää kuinka laitteistotestausta ja ohjelmistotestausta on tehty SD-muistikorttilaajennusta kehitettäessä.




KUVIO 16. SD-muistikorttilaajennuksessa käytettyjä osatestejä

Kuvio 16 esittää kuinka testaussuunnitelmaa on pyritty käyttämään opinnäytetyössä. Laajennuskortit vaativat aina ohjauksen RaceDAC-laitteelta, joten aina ensin tehtiin demokortille ohjelma, jonka tarkoitus oli simuloida RaceDAC-laitetta. Kuvion esimerkissä muistikorttilaajennuksessa ensin tehtiin ohjelmaan SPI-väylän tarvitsemat asetukset, jonka jälkeen SPI-väylän toiminta testattiin liittämällä SD-muistikortin sisältävä levy demokorttiin. Kun testauksessa havaitut ongelmat saatiin korjattua, siirryttiin seuraavaan vaiheeseen. Aina kun laajennuskortissa tarvittavat eri osat oli saatu testattua demokortin avulla, siirryttiin tekemään samat asiat RaceDAC-laitteeseen.

## *6.2 Laitetestaus*

Laitetestaukseen kuului molempien lopullisten laajennuskorttien toiminnan testauksen lisäksi GPS-moduulin testaaminen käytännössä. GPS-moduuli ei saanut yhteyttä satelliittiin sisätiloissa, joten sen toiminta täytyi todeta liittämällä RaceDAC-laite autoon. Ulkona testattaessa GPS-moduuli sai luultavasti yhteyden satelliittiin, mutta RaceDAC-laite resetoitui aina samalla hetkellä. Ongelmaan ei keretty löytää varmaa

ratkaisua, mutta ongelma johtui luultavasti GPS-moduulin ottaman virran äkillisestä muutoksesta satelliittiyhteyden muodostuessa.

## 6.3 Ohjelmistotestaus

Ohjelmistotestauksen rooli opinnäytetyössä jäi hyvin vähäiseksi, koska käyttäjä ei voi vaikuttaa laajennuskorttien käyttöön kovin monella tapaa. Käyttäjä voi antaa laitteelle bluetooth-yhteyden kautta komentoja, joilla voidaan esimerkiksi asettaa laitteelle uusi näytteenottotaajuus, asettaa kalibrointiarvot AD-muunnoskanaville, vaihtaa logirivin tyyppiä tai asettaa uuden ohjelmistopäivitysviiveen laitteen käynnistykseen. Nämä kaikki ominaisuudet on kuitenkin testattu ennen laajennuskorttien suunnittelua ja laajennukset eivät vaikuta niiden toimintaan mitenkään. Ainoastaan näytteenottotaajuuden vaihto vaikutti laitteen toimintaan laajennuskorttien kanssa. RaceDAC-laitteelle voidaan asettaa näytteenottotaajuudeksi 1, 2, 5, 10, 20, 50 tai 100 Hz. Laitteen toimintaa testattiin kaikilla taajuuksilla ja molemmilla laajennuskorteilla. Molempien laajennuskorttien käytössä 100 Hz:n taajuudella tuli ongelmia. Laite ei kyennyt lähettämään logirivejä kokonaisina bluetoothin kautta silloin, kun GPS-laajennuskortti oli kiinnitetty laitteeseen. SD-muistikorttilaajennuksen ollessa kiinnitettynä laitteeseen, laite ei kyennyt kirjoittamaan jokaista riviä muistikortille. Noin joka kuudes rivi jäi kirjoittamatta, kun näytteenottotaajuus oli 100 Hz. Muilta osin ohjelmiston toiminta voitiin todeta oikeaksi, kun katsottiin laitteen lähettämien logirivien olevan oikean muotoisia molemmilla laajennuskorteilla varustettuna ja bluetooth-yhteyden olevan toimiva.

RaceDAC-laite on suunniteltu käytettäväksi RaceChrono-sovelluksen kanssa, joka puolestaan on kehitetty käytettäväksi GPS-moduulin sisältävän laitteen kanssa. Nykyään GPS-moduulit eivät normaalisti pysty päivittämään paikannustietojansa 10 Hz:n taajuutta nopeammin, joten RaceDAC-laitteen ei välttämättä tarvitse pystyä lähettämään logirivejä 100 Hz:n taajuudella. Tästä syystä toimeksiantajan kanssa päätettiin, että kummankaan laajennuskortin ei tarvitse kyetä toimimaan suurimmalla näytteenottotaajuudella, vaan 50 Hz:n taajuus on riittävä.

# 7 POHDINTA

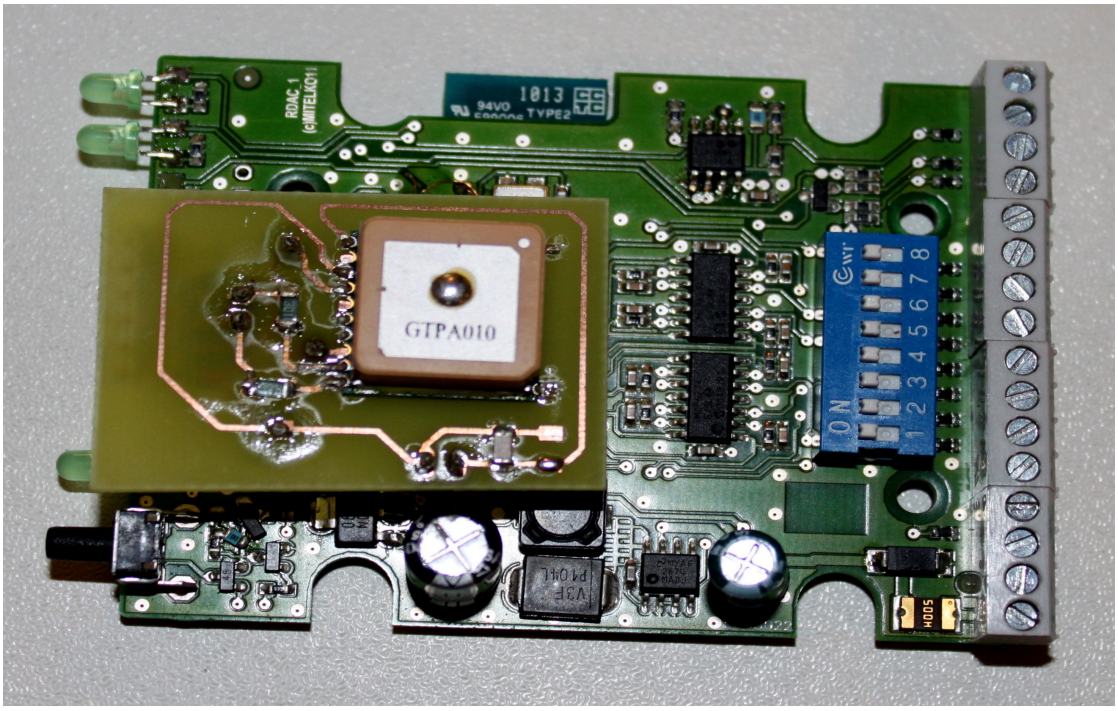
Opinnäytetyössä saavutettiin melko hyvin kaikki tavoitteet. Ainoat suuremmat puutteet, mitkä työhön jäivät olivat laajennuskorttien toiminta suurimmalla mahdollisella 100 Hz:n päivitystaajuudella ja GPS-moduulin satelliittiyhteyden jälkeinen RaceDACin resetoituminen. Päivitystaajuusongelmaan ei kuitenkaan olisi pystytty saamaan ratkaisua ilman koko RaceDAC-laitteen päivittämistä. Tämä ongelma ei myöskään vaikuta ratkaisevasti laitteen käyttöön, koska laitetta ei yleensä käytetä kuin 10 Hz:n päivitystaajuudella. Tästä syystä voidaan sanoa tavoitteiden täyttyneen päivitystaajuuden osalta, sillä laajennuskortit saatiin toimimaan moitteettomasti vielä 50 Hz:n taajuudella. GPS-moduulin satelliittiyhteyttä alettiin testaamaan niin myöhään, että aika ei ongelman havaitsemisen jälkeen enää riittänyt sen korjaamiseen. GPS-laajennuskortin toimimaan saanti sisätiloissa laittoi ajattelemaan, ettei ulkona toiminta muuttuisi ainakaan niin radikaalisti.

Opinnäytetyössä ei tullut ongelmaa vähäisestä laajennuskortille varatusta pinnimäärästä, vaikka alussa ajateltiin sen muodostuvan ongelmaksi. Molemmat opinnäytetyön aikana toteutetut laajennukset saataisiin luultavasti suunniteltua samalle piirikortille, jos otettaisiin käyttöön osa in-circuit-ohjelmointiliittimen pinneistä.

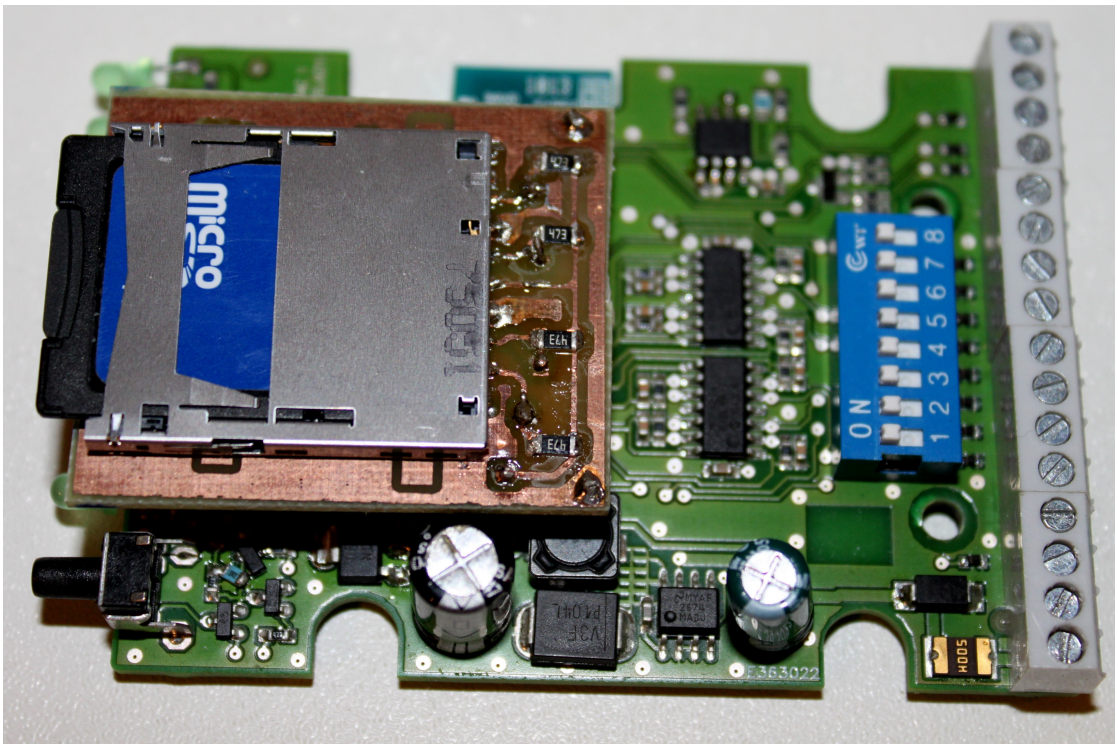
Ohjelmistopäivityksen osalta työssä onnistuttiin hyvin. Joitakin asioita olisi varmasti voinut tehdä hieman järkevämmin. Esimerkiksi muistikorttilaajennuksen yhteydessä ini-tiedoston avaaminen ei koskaan onnistunut ensimmäisellä kerralla vaan tiedostoa täytyi avata silmukassa, kunnes se aukesi virheettömästi. Tällaisten silmukoiden käyttö voi joissain tilanteessa olla melko riskialtista, koska ohjelma voi päätyä ikisilmukkaan ja resetoitua. Tällaiseen silmukkaan voitaisiin päätyä opinnäytetyössä tehdyssä ohjelmassa, jos tiedostoa yritettäisiin avata silloin, kun sitä ei ole olemassa. Tämän ei pitäisi koskaan tapahtua, koska tiedoston tiedetään olevan onnistuneesti luotu juuri ennen kuin ohjelmassa mennään silmukkaan.

Piirilevysuunnittelussa olisi voinut miettiä tarkemmin laitteen kotelon mittoja, mutta opinnäytetyön tarkoituksena oli laajennuskorttien prototyyppien teko, eikä niistä var-

maankaan tule myytäviä versioita. Tästä syystä kumpikaan laajennuskortti ei mahtunut kotelon sisälle yläkannen ollessa paikallaan. Prototyyppien valmistuksessa päädyttiin tähän karkeaan ratkaisuun, jotta saatiin käytettyä mahdollisimman paljon valmiiksi koululta löytyviä komponentteja. SD-muistikorttilaajennuksessa käytettiin prototyyppilevyssä SD-muistikortille sopivaa korttipaikkaa. SD-muistikortti on kuitenkin niin kookas laitteeseen verrattuna, että lopullisessa versiossa tullaan luultavasti käyttämään microSD-muistikorttipaikkaa. Kuvioissa 17 ja 18 näkyvät laajennuskortit kiinnitettyinä RaceDAC-laitteeseen.




KUVIO 17. GPS-laajennus RaceDAC-laitteessa




KUVIO 18. SD-muistikorttilaajennus RaceDAC-laitteessa

# LÄHTEET

Dimension Engineering LLC. 2011. A beginner's guide to accelometers. Viitattu 28.11.2011 http://www.dimensionengineering.com/accelerometers.htm.

Elastic Sheep. 2010. Harrastelijoiden elektroniikkaprojekteja esittelevä sivusto. Viitattu 21.11.2011 http://elasticsheep.com/2010/01/reading-an-sd-card-with-an-atmega168/.

FAT filesystem. Wikipedian sivusto. 2011. Viitattu 13.10.2011 http://en.wikipedia.org/wiki/FAT_filesystem.

GlobalTop. 2011. GPS-moduulien valmistaja. Viitattu 1.11.2011 http://www.gtop-tech.com/jsf/index.jsf.

Kingmax Digital Inc. 2006. Secure Digital Memory Card. Viitattu 20.11.2011 http://downloads.amilda.org/MODs/SDCard/SD.pdf.

Microchip Technology inc. 2010. dsPIC33F/PIC24H Family Reference Manual

application notes for dspic. Viitattu 17.11.2011.

RaceChrono. 2011. GPS-paikannukseen perustuva kierrosaikojen seurantaohjelma. Viitattu 6.11.2011 http://www.racechrono.com/about/.

Reen, P. & Mohaanswamy, N. 2010. Application note AN1045. File I/O Functions Using Microchip's Memory Disk Drive File System Library. Viitattu 8.10.2011. http://www.microchip.com/downloads/en/AppNotes/01045b.pdf.

Räty, J. 2007. PC-Tekniikka osa IV. Viitattu 10.11.2011 http://www.ratol.fi/opensource/pctekniikka/3/kirjat/tilanvarausjarjestelmat.pdf.

SecureDigital. 2011. Wikipedian sivusto. Viitattu 15.11.2011. http://en.wikipedia.org/wiki/Secure_digital.

Serial peripheral interface bus. 2011. Wikipedian sivusto. Viitattu 15.10.2011 http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus.