

OPINNÄYTETYÖ
Tuomas Virjonen 2012

RALLIAUTON GPS-LOGGERI



Rovaniemen
ammattikorkeakoulu
University of Applied Sciences

TIETOTEKNIIKAN KOULUTUSOHJELMA

ROVANIEMEN AMMATTIKORKEAKOULU

TEKNIikka JA LIIKENNE

Tietotekniikan koulutusohjelma

Opinnäytetyö

RALLIAUTON GPS-LOGGERI

Tuomas Virjonen

2012

Ohjaaja Matti Rahkala

Tekijä	Tuomas Virjonen	Vuosi	2012
Työn nimi	Ralliauton GPS-loggeri		
Sivumäärä	43 + 3		

Työssä suunniteltiin ja toteutettiin GPS-sijaintitietoa tallentava laite, jonka tarkoituksena on mahdollistaa kuljetun reitin tarkasteleminen rallikilpailun jälkeen. Laite tallentaa GPS-moduulilta saadut sijaintitiedot sekä nopeuden massamuistiin, Google Earth -karttasovelluksen ymmärtämään muotoon, jolloin reitti saadaan piirrettyä suoraan sovelluksen karttapohjalle.

Laitteen suunnittelussa pyrittiin ottamaan huomioon, että laitetta tullaan käyttämään kilpa-ajoneuvossa kilpailun aikana, joten laitteen käyttöliittymästä haluttiin mahdollisimman helppokäyttöinen ja selkeä. Laite koostuu kolmesta pääosasta: GPS-moduulista, Atmel AVR -mikrokontrollerista sekä SD-muistikortista.

Laitteen ohjelmisto toteutettiin AVR Studio -ohjelmalla ja kirjoitettiin C-kielellä. Lähdekoodin kääntämiseen käytettiin AVR-GCC-kääntäjää. Ohjelmistossa käytettiin standardikirjastojen lisäksi vapaasti saatavilla olevaa FAT-tiedostojärjestelmää muistikortin luku- ja kirjoitusrutiineihin.

Työssä saatiin toteutettua alkuperäisten vaatimusten mukainen ja toimiva laite.

Author	Tuomas Virjonen	Year	2012
Subject of thesis	Rally Car GPS Data Logger		
Number of pages	43 + 3		

The aim of this final year study was to design and implement a device which collects GPS positioning data and allows the user to examine the route after the race. The device records positioning data and speed from GPS to a mass storage device in a format that is compatible with Google Earth application so that the route can be presented directly on the application's map.

The device was designed to be installed in a car and to be used during the race, so the user interface was designed to be as simple and clear as possible. The device consists of three main components: GPS-module, Atmel's AVR-microcontroller and SD-card as a mass storage device. The software was developed on AVR Studio development software and was written in C-language. The source code was compiled by AVR-GCC compiler. The software implements standard C-libraries and free FAT-file system for disk read and write routines.

As a result of this study the initial requirements for the device were fulfilled and the device operates as designed.

Key words

AVR, GPS, KML, microcontroller, NMEA, SD-card

SISÄLTÖ

KUVIOLUETTELO	1
KÄSITTEET JA LYHENTEET	2
1 JOHDANTO	4
2 GPS-JÄRJESTELMÄN HISTORIA JA KEHITYSKAARI	6
2.1 TRANSIT SYSTEM.....	6
2.2 TIMATION JA PROJECT 621B.....	7
2.4 NYKYAIKAINEN NAVSTAR GPS	9
3 GPS-JÄRJESTELMÄN OSAT JA TOIMINTAPERIAATE	11
3.1 JÄRJESTELMÄN OSAT	11
3.2 GPS-JÄRJESTELMÄN TOIMINTAPERIAATE	12
3.3 GPS-SIGNAALIN VIRHEET	15
3.4 SA-HÄIRINTÄ	17
3.5 GPS-JÄRJESTELMÄN VIRHEENKORJAUSMENETELMÄT	18
4 GPS-LOGGERI	20
4.1 LAITTEEN MÄÄRITTELY	20
4.2.1 Trimble GPS -moduuli ja NMEA 0138 -lausekkeet	20
4.1.2 Secure Digital (SD) -muistikortti ja sen liittäminen SPI-väylään	23
4.1.3 AVR Atmega644P -mikrokontrolleri	26
4.2 LAITTEEN FYYSINEN SUUNNITTELU JA TOTEUTUS.....	27
4.3 OHJELMISTON SUUNNITTELU JA TOIMINTA	29
4.3.1 Ohjelmiston suunnittelu ja sen ominaisuudet.....	29
4.3.2 FAT-tiedostojärjestelmä ja FatFS.....	31
4.3.3 Ohjelmiston toiminta.....	32
5 LAITTEEN TESTAAMINEN JA KEHITYSMAHDOLLISUUDET	34
5.1 LAITTEEN TESTAAMINEN.....	34
5.2 KEHITYSMAHDOLLISUUDET	35
5.2.1 Havaittujen vikojen ja puutteiden korjaaminen.....	35
5.2.2 Lisää ominaisuuksia	36
6 YHTEENVETO	38
LÄHTEET	40
LIITTEET	44

KUVIOLUETTELO

KUVIO 1. PIIRROS TRANSIT-SATELLIITISTA TOIMINTAKUNNOSSA	7
KUVIO 2. TIMATION I -SATELLIITTI	8
KUVIO 3. NTS-1-SATELLIITTI	9
KUVIO 4. GPS-JÄRJESTELMÄN OSAT	11
KUVIO 5. GPS-SIGNAALIN RAKENNE	14
KUVIO 6. KATVEALUE.....	15
KUVIO 7. MONITIEHEIJASTUMINEN.	15
KUVIO 8. ILMAKEHÄN VAIKUTUS SIGNAALIIN.....	16
KUVIO 9. GPS-JÄRJESTELMÄN TARKKUUS ENNEN JA JÄLKEEN SA-HÄIRINNÄN POISTAMISEN	18
KUVIO 10. DIFFERENTIAALISEN GPS:N TOIMINTAPERIAATE.....	19
KUVIO 11. EGNOS- JA WAAS-JÄRJESTELMIEN TOIMINTAPERIAATE.....	19
KUVIO 12. TRIMBLE LASSEN DR+GPS -MODUULI JA AKTIIVINEN GPS-ANTENNI	21
KUVIO 13. SD-MUISTIKORTTITYYPIT JA KÄYTTÖSOVELLUKSIA	24
KUVIO 14. SPI-VÄYLÄN LIITÄNTÄTAPA.....	24
KUVIO 15. MUISTIKORTIN KYTKENTÄ MIKROKONTROLLERIIN	25
KUVIO 16. LOPULLINEN PIIRILEVYVERSIO LAITEKOTELOSSAAN.....	27
KUVIO 17. KÄYTTÖJÄNNITTEEN REGULAATTOREIDEN KYTKENNÄT	28
KUVIO 18. GPS-LOGGERIN KÄYTTÖLIITTYMÄ	29
KUVIO 19. VALMIIN LAITTEEN TALLENTAMA LYHYT REITTI GOOGLE EARTH -SOVELLUKSESSA.....	30
KUVIO 20. LAITTEEN TALLENTAMAN KML-TIEDOSTON RAKENNE XML-KUVAUKSINEEN	31
KUVIO 21. 7-SEGMENTTINÄYTTÖRYHMÄN KYTKENTÄ.....	37
KUVIO 22. TYÖSSÄ VALMISTETTU KILPA-AJONEUVON GPS-LOGGERI.	39

KÄSITTEET JA LYHENTEET

AVR	Atmelin valmistamista 8-bittisistä mikrokontrollereista käytetty yleisnimike.
DGPS	Differentiaalinen GPS. GPS-järjestelmän tunnettuun kiintopisteeseen perustuva häiriönkorjausmenetelmä.
Doppler-ilmiö	Aaltoliikkeen taajuudessa, aallonpituudessa tai vaiheessa aiheutuva muutos, joka johtuu lähettimen tai vastaanottimen liikkeestä toisiinsa nähden.
GPS	Global Positioning System. Yhdysvaltain puolustusministeriön kehittämä ja ylläpitämä satelliittipaikannusjärjestelmä.
GPS Fix	GPS-laitteen yhteyden tilaa kuvaava korjaustieto.
KML	Keyhole Markup Language. Mm. Googlen karttasovelluksissaan käyttämä tiedostomuoto.
Latitudi	Leveyspiiri. Sijainti etelä-pohjoissuunnassa päiväntasaajaan nähden.
Longitudi	Pituuspiiri. Sijainti itä-länsisuunnassa Greenwichin pituuspiiriin nähden.
NMEA	National Marine Electronics Association. Käytetään myös yleisnimityksenä NMEA 0183 -standardista, joka on yleisesti GPS-laitteissa käytetty sijaintitietoja ilmoittava standardi ja jota kyseinen organisaatio kehittää ja ylläpitää.
PRN-koodi	Pseudo Random Noise. GPS-signaalissa tunnetuilla algoritmeilla luotu näennäissatunnainen signaali.
SPI	Serial Peripheral Interface Bus. Standardi tiedonsiirtoväylä kahden laitteen välillä.

UHF	Ultra High Frequency. Suoraan etenevät, taajuusalueeltaan 0,3-3 GHz:ä olevat radiotaajuudet. Käytetään mm. satelliiteissa ja televisiokanavien lähetyksessä. Vaatii käytännössä suoran näköyhteyden lähettimeen.
USART	Universal Synchronous/Asynchronous Receiver Transmitter. Sarjamuotoisen dataliikenteen lähetys ja vastaanottopiiri.

1 JOHDANTO

GPS-loggerin suunnittelu sai alkunsa mielenkiinnosta seurata jälkeinpäin rallikilpailun kuljettua reittiä ja vertailla nopeuksia erikoiskokeilla ja niiden eri kohdissa. Koska reaaliaikainen seuranta ja auton ulkopuolelle lähetettävä tieto on Suomen autourheilun kattojärjestön, AKK-motorsportsin, alaisten kilpailujen säännöissä kielletty, jäi ainoaksi vaihtoehdoksi hankkia sijaintitietoa tallentava laite, josta tiedot voitaisiin lukea jälkikäteen ja piirtää tiedoista selkeä ja luettava reitti. Koska käyttötarkoitukseen, eli kilpailukäyttöön, myytävät laitteet ovat ominaisuuksiinsa nähden varsin kalliita, päätettiin vastaava omiin käyttötarkoituksiin räätälöity laite rakentaa itse. Samalla heräsi into perehtyä GPS-järjestelmän toimintaperiaatteeseen hieman tarkemmin, sillä järjestelmän käyttö on kasvanut viime vuosina räjähdysmäisesti monissa eri sovelluksissa. GPS-järjestelmän toimintaperiaate ja häiriöt ovat insinöörielle perustietoa, mutta tiedonrippeitä tuntuu joutuvan keräämään useista lähteistä. Siispä työn edetessä tuli haalittua tiivis paketti järjestelmän ominaisuuksista.

Työssä haluttiin tallentaa kuljettu reitti, jolloin melko luonnollisesti syntyi idea käyttää vapaasti saatavilla olevia karttasovelluksia, kuten Google Earth. Googlen karttasovellukset mahdollistavat reittien merkitsemisen karttasovelluksiinsa kml-formaatissa, joka on xml-merkkaukielinen tiedostomuoto, ja tiedostot ovat suoraan yhteensopiva Google Earth -karttasovelluksen kanssa.

AVR-mikrokontrolleri allekirjoittaneelle oli työtä aloitettaessa suhteellisen uusi alusta, jonka käytöstä oli kokemusta ainoastaan äärimmäisen alkeellisissa ja pienissä sovelluksissa. AVR:n käyttöönottoa tuki valmistajalta saatavilla oleva selkeä dokumentointi ja Internetistä löytyvät tuhannet harrastajien projektit, joita tutkimalla ja seuraamalla AVR-mikrokontrolleri käyttömahdollisuuksineen ja ominaisuuksineen alkoi pikkuhiljaa käydä tutuksi. Työn edetessä olikin iloa huomata oman osaamisen lisääntyessä kymmenet ja taas kymmenet käyttökohteet ja ideat jatkojalostaa työssä rakennettua laitetta.

Työn suunnitteluvaiheessa oli tiedon tallennusvaihtoehtoina SD-muistikortti tai tallennus suoraan laitteen muistiin. Jälkimmäinen vaihtoehto olisi vaatinut tieto-

koneen sarjaliikenneyhteydellä käytettäväksi tiedon purkamiseksi laitteesta, joten se vaihtoehto hylättiin ja päädyttiin loppukäyttäjän kannalta helppokäyttöisempään SD-muistikorttiin, joka voitaisiin vain poistaa laitteesta ja liittää myöhemmin tietokoneeseen. SD-muistikortin käyttö mikrokontrollerin SPI-väylän kautta ja toiminta tallennusmediaana ei tulisi kuitenkaan olemaan se helpoin vaihtoehto. Laite koostuisi yksinkertaisuudessaan kolmesta pääosasta: GPS-moduulista, SD-muistikortista, sekä mikrokontrollerista.

C-ohjelmointikielen perusteilla ei pitkälle pötkitä, mutta täytyy todeta vanhan sanonnan: ”kyllä työ tekijäänsä opettaa”, pitävän enemmän kuin paikkansa. Työn tekeminen alkoikin pitkälti palauttamalla mieleen vähäiset C-kielestä mieleen jääneet kikat ja niksit, joita pyörittelemällä AVR-kehitysalustan pienissä sovelluksissa päästiin melkoisen nopeasti laajempien kokonaisuuksien ja funktioiden kimppuun.

Laite haluttiin alusta asti rakentaa itsenäiseksi ja AVR-kehitysalustoista riippumattomaksi ”standalone”-laitteeksi, eivätkä ajoneuvokäytössä monenlaiset hypylangoin viritellyt laitekoonpanot tulleet kysymykseenkään. Laitteiston suunnittelun ensimmäinen lähtökohta oli siis helppokäyttöisyys ja selkeys kilpailutilanteessa, jolloin erikoiskokeelle valmistautuessa ei tarvitsisi kiinnittää erityistä huomiota laitteen päälle kytkemiseen tai käyttöön. Tällöin huomioon otettavia seikkoja olivat loppukäyttäjälle tärkeimmät käyttöliittymä ja laitteen fyysinen koko, kaikki ylimääräinen tulisi karsia pois. Milloin laite on toimintakunnossa, sekä milloin ja miten laite tallentaa tietoa, olivat huomioon otettavat seikat, jotka haluttiin selkeästi ja yksinkertaisesti esittää. Laitteen ja sen painikkeiden koko on ratkaisevassa asemassa helppokäyttöisyyden kannalta kilpailutilanteessa, joten laitteessa päädyttiin käyttämään auton kojelautaan kiinnitettävää laitekoteloa suurilla ja selkeillä painikkeilla, ainoastaan muutaman led-valon indikoidessa laitteen tilaa.

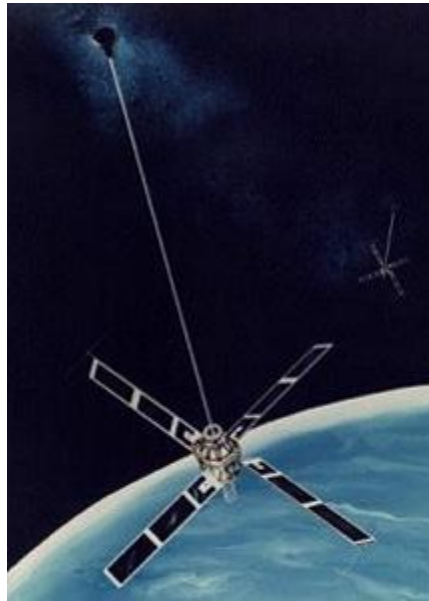
2 GPS-JÄRJESTELMÄN HISTORIA JA KEHITYSKAARI

2.1 TRANSIT System

GPS-järjestelmä juontaa juurensa 1940-luvulta, jolloin sitä ideoitiin lento- ja meriliikenteen kulun helpottamiseksi, mutta jo ennen tätä 1910-luvulla tunnettiin tekniikka, jossa taivaalla olevia satelliitteja käytettäisiin eräänlaisina radiomajakoina laivoille. Majakat määrittäisivät sijainnin laskemalla radiomajakakan lähettämän radiosignaalin matka-ajan. 1950-luvun alkupuolella toisen maailmansodan jälkeen ja kylmän sodan kiristyessä huomattiin nykyisenkaltaisen GPS-järjestelmän tarpeellisuus varsinkin globaalimmassa mittakaavassa. Idea ensimmäiseen nykyaikaisen kaltaiseen järjestelmään syntyi, kun amerikkalaiset tiedemiehet huomasivat, että Neuvostoliiton Sputnik-satelliittia pystyttiin seuraamaan sieppaamalla sen lähettämiä radiosignaaleja ja signaalien perusteella määrittämään satelliitin sijainti ja nopeus taivaalla. (John Hopkins University APL 1998; 2008; 2010.)

Tekniikan kehittyessä ensimmäiset isot askeleet tulevan järjestelmän kehityksessä otettiin vuonna 1960, kun TRANSIT System -satelliittipaikannusjärjestelmän ensimmäinen satelliitti "Transit 1B" saatiin onnistuneesti taivaalle 15. huhtikuuta 1960, edeltävän "Transit 1A":n tuhouduttua laukaisussa puolta vuotta aikaisemmin syyskuussa 1959. Yhdysvaltain merivoimien kehittämä TRANSIT otettiin käyttöön vuonna 1964 ja tunnetaan myös nimellä NAVSAT (Navy Navigation Satellite System). TRANSIT-järjestelmän päätarkoitus oli antaa Yhdysvaltain ydinsukellusveneille sijaintitieto nopeasti aluksen noustua pintaan, ennen kuin vihollinen pystyisi tutkillaan aluksen havaitsemaan. Järjestelmään kuului viisi satelliittia ja viisi varasatelliittia, jotka kiersivät 1100 kilometrin korkeudella maan pinnalta, maan kierron kestäessä 106 minuuttia. Paikannustarkkuus järjestelmällä oli myöhempien parannusten myötä nykymittapuulla vaatimatonta 200 metriä, eikä pieni määrä matalalla kiertäviä satelliitteja mahdollistanut järjestelmän käyttöä tauotta joka puolella maapalloa, vaan toiminta oli katkonaista ja käytettävissä vain lyhyitä aikoja kerrallaan katkojen kestäessä päivän-tasaajan korkeudella jopa useita tunteja. Transit-satelliitti lähetti kahdella UHF-lähettimellään ratatietonsa kahden minuutin välein, joten sukellusveneiden pai-

kannustiedon vastaanottaminen kesti vastaavan ajan. Vastaanotin pystyi laskemaan sijaintinsa satelliitilta saatujen ratatietojen lisäksi Doppler-ilmiötä hyväksi käyttäen, koska jokaisella sijainnilla Doppler-ilmiön aikaansaama muutos signaalissa on uniikki. Satelliitin kello korjattiin kaksi kertaa vuorokaudessa neljältä maanpäälliseltä laivaston seuranta-asemalta. TRANSIT-järjestelmä oli käytössä aina vuoteen 1991 asti, kunnes se tuli nykyaikaisen Navstar GPS:n syrjäyttämäksi. (John Hopkins University APL 1998a; 1998b; 2008; 2010, NASA 2010a, Spacecraft encyclopedia 2004a.)



Kuvio 1. Piirros Transit-satelliitista toimintakunnossa (Navy NRL 2010)

2.2 TIMATION ja Project 621B

Yhdysvaltain laivaston Naval Research Laboratory (NRL) aloitti seuraavan navigaatioprojektin, TIMATION-järjestelmän kehityksen jo TRANSIT Systemin käyttöönoton aikoihin vuonna 1964 (TIME navigATION). TIMATION jatkoi siitä, mihin 60-luvun alussa toiminut SECOR-projekti jäi. SECOR-satelliitti laski neljän maan aseman lähettämän signaalin perusteella sijaintinsa tai yhden tuntemattoman aseman sijainnin maan kamaralla. TIMATION oli täysin kokeellinen hanke, jonka tarkoituksena oli saada taivaalle tarkasti aikansa pitävillä kelloilla varustettuja satelliitteja, joiden tuottamalla taajuudella etäisyyden mittaaminen olisi huomattavasti aikaisempaa tarkempaa. TIMATION-järjestelmä koostui alkuvaiheen kokeiluissa kolmesta satelliitista ja näiden ohjaukseen tarkoitetuista seitsemästä

maa-asemasta. Satelliiteista ensimmäinen laukaistiin napoja kiertävälle radalle 800 kilometrin korkeuteen vuonna 1967. Ensimmäinen satelliitti, korkealuokkaisella kvartsioskillaattorilla varustettu Timation I teki lukuisia tuloksekkaita mittauksia mittaamalla pienten laivojen, lentokoneiden ja kuorma-autojen etäisyyksiä ns. passiivisena etäisyysmittauksena Timation-satelliitin ollessa ainoa osa mittausjärjestelmää. (Navy NRL 1986; 2010.)



Kuvio 2. Timation I -satelliitti (Navy NRL 2010)

Vuonna 1969 laukaistu Timation II -satelliitti oli jo ensimmäistä versiota huomattavasti kehittyneempi laite, jonka ominaisuuksiin kuului niin ikään kvartsikello, mutta lisäksi mm. kaksi eri taajuudella toimivaa lähetintä. Näiden kahden 150 MHz:n ja 400 MHz:n taajuudella lähetettyjen signaalien avulla saatiin järjestelmälle laskettua ionosfäärin heijastumisen aiheuttaman virheen korjaus aikaisempaa tarkemmin ja täten paikannustarkkuus kasvoi huomattavasti. NTS-1:ksi nimetty Timation III -satelliitti taas laukaistiin lukuisine parannuksineen tuomaan TRANSIT- ja TIMATION-järjestelmiä lähemmäs toisiaan, jolloin kaksi eri järjestelmää sulautui pikkuhiljaa yhteen paikannustarkkuuden edelleen kasvaessa. TIMATION III oli ensimmäinen satelliitti, joka sisälsi nykyäänkin käytettävän atomikellon. Se laukaistiin vuonna 1974, ja kaiken kaikkiaan TIMATION-järjestelmä koostui lopulta 28 satelliitista. (Navy NRL 1986; 2010; NASA 2010b.)



Kuvio 3. NTS-1-satelliitti (Navy NRL 2010)

U.S. Air Force aloitti TIMATION-järjestelmän kanssa samoihin aikoihin oman tutkimustyön tutkiakseen mahdollisuutta luoda omiin käyttötarkoituksiinsa sopivampi aikaisempaa laivaston TRANSIT-järjestelmää kehittyneemmän paikannusjärjestelmän. Hanke sai nimen Project 621B. Projektin tavoite oli kehittää nopea, reaaliaikainen ja tarkka järjestelmä, koska laivaston käyttämä TRANSIT oli auttamatta liian hidas ja epätarkka nopeiden hävittäjäkoneiden ja ohjusten paikantamiseen. Projektin työryhmä keksi, että neljän tai viiden Transit-satelliitin ryppään, maa-aseman ja vastaanottimen avulla saataisiin aikaan nopea ja tarkka, kolmiulotteinen paikannus. Järjestelmä ei käyttänyt satelliittien atomikelloja, vaan atomikellojen aika lähetettiin seitsemästä maa-asemasta käyttäen PRN-signaalia (Pseudo Random Noise), joka on taajuusmuotona käytössä myös nykyisessä GPS-järjestelmässä. (Richard Easton 2006; Yhdysvaltain puolustusministeriö 1996.)

2.4 Nykyaikainen Navstar GPS

Vuonna 1973 tehtiin varsinainen harppaus kohti nykyaikaista GPS-järjestelmää, kun Yhdysvaltain puolustusministeriö päätti kehittää uuden Navstar GPS -järjestelmän, joka syntyy sanoista Navigation System Using Timing And Ranging Global Positioning System, eli globaali ajanmääritykseen ja etäisyyden mittauksen perustuva paikannusjärjestelmä. Myöhemmin käyttöön on vakiintunut terminä pelkkä GPS. Navstar GPS luotiin ottamalla Yhdysvaltojen meri- ja ilma-voimien käytössä olleiden Project 621B-, TRANSIT-, TIMATION- ja SECOR-järjestelmien parhaat puolet käyttöön yhtenä kokonaisuutena. Järjestelmä osoitautui mielenkiintoiseksi myös siksi, että sen suunnittelun yhdeksi lähtökohdaksi

tuli mukaan sotilaskäytön lisäksi myös siviilikäyttö vuoden 1982 tuhoisan lentokoneturman jälkeen, pääpainon ollessa kuitenkin sotilaskäytössä ja täten järjestelmän luotettavuudessa vihollisen ja luonnon aiheuttamia häiriötä vastaan sekä paikannustarkkuudessa (NAVCEN 1995; 1996.)

Ensimmäinen Navstar GPS -järjestelmän ensimmäisen sukupolven Block I -satelliitti ”Navstar 1” laukaistiin kiertoradalle 22. helmikuuta 1978. Toisen sukupolven Block II -satelliitit olivat tilauksessa vuosi Navstar 1 -satelliitin laukaisun jälkeen jo vuonna 1979, mutta ensimmäinen toimiva Block II -satelliitti ”Navstar 2-01” saatiin toimintakuntoon ja taivaalle vasta helmikuussa 1989. (NAVCEN 1995; 1996, Spacecraft encyclopedia 2004b.)

U.S. Air Force Space Command (AFSC) ilmoitti Navstar GPS -järjestelmän olevan operatiivisessa valmiustilassa joulukuussa 1993, jolloin järjestelmä koostui 24:stä toimivasta ensimmäisen sukupolven Block I sekä toisen sukupolven Block II ja Block IIA satelliiteista. Huhtikuussa 1995 järjestelmä julistettiin täysin valmiiksi ja toimintakykyiseksi tauotta maailmanlaajuisesti. Tällöin Block I -sukupolven satelliitit oli jo poistettu kokonaan käytöstä ja järjestelmä koostui ainoastaan toisen sukupolven Block II ja IIA satelliiteista. (NAVCEN 1995; 1996, Spacecraft encyclopedia 2004b.)

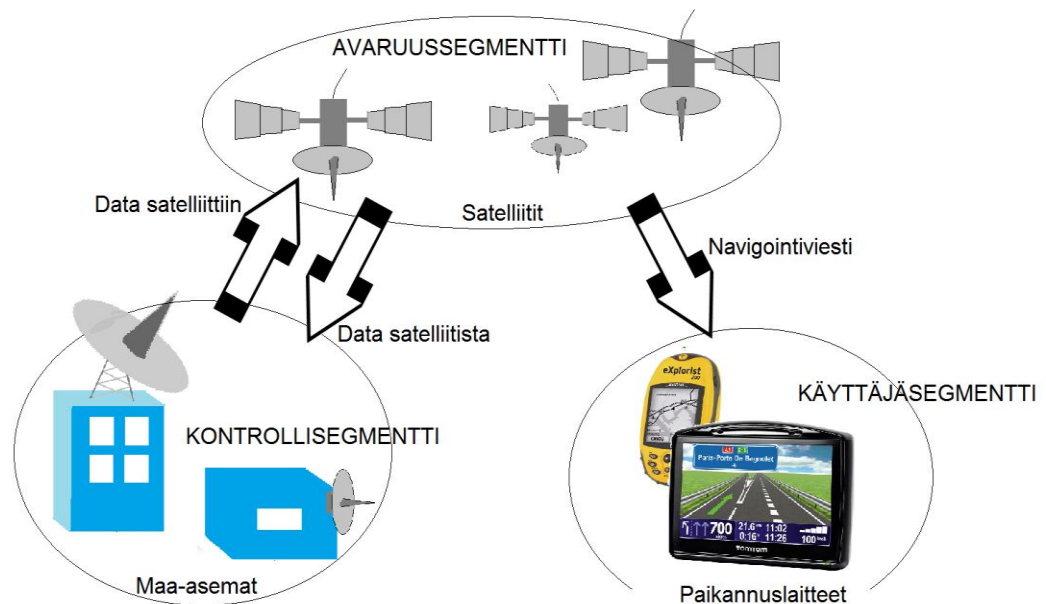
Vuoteen 1985 mennessä järjestelmään kuuluvia Block I -sukupolven satelliitteja laukaistiin onnistuneesti kaiken kaikkiaan yhdeksän kappaletta ja ne palvelivat käyttötarkoituksessaan aina vuoteen 1995 asti. Nykyään järjestelmä koostuu 30:stä Block II -sukupolven satelliitista. Lähivuosina ollaan siirtymässä seuraavaan, kolmanteen Block-sukupolveen. (NAVCEN 1995; 1996.)

3 GPS-JÄRJESTELMÄN OSAT JA TOIMINTAPERIAATE

3.1 Järjestelmän osat

GPS-järjestelmä koostuu kolmesta eri segmentistä:

- Avaruussegmentistä (Space Segment, SS), satelliitit
- Kontrollisegmentistä (Control Segment, CS), maa-asemat
- Käyttäjässegmentistä (User Segment, US), paikannuslaitteet.



Kuvio 4. GPS-järjestelmän osat

Avaruussegmentti koostuu kolmestakymmenestä satelliitista, jotka kiertävät maata kuudella eri kiertoradalla keskimäärin 20200 km:n korkeudella maan pinnalta. Jokaisella kiertoradalla kiertää neljä satelliittia peräkkäin. Jokaisen satelliitin kiertoaika maan ympäri on 12 tuntia, saman kiertoradan satelliittien välisen etäisyyden ollessa 60° . Satelliitteja kiertää toiminnassa yhtäaikaaisesti 24 kappaletta, jolloin loput satelliitit ovat käytettävissä paikkaamaan rikkoutunut satelliitti. Näin pystytään varmistamaan, että joka paikassa maapallolla, jokaisena hetkenä vuorokaudesta on käytettävissä paikannukseen tarvittava minimimäärä, 3 kpl, satelliitteja näkyvällä taivaan osalla. (NAVCEN 1995; 1996; 2010).

Kontrollisegmentti koostuu neljästä maa-asemasta (Monitor Station, MS), jotka on sijoitettu ekvaattorin läheisyyteen sekä näiden kaikkien valvonnasta ja ohja-

uksesta vastaavasta keskusvalvonta-asemasta (Master Control Station, MCS). Nämä neljä maa-asemaa sijaitsevat Tyynessä valtameressä Marshallsaarilla, Kwajaleinin atollilla Havaijilla, Diego Garcian atollilla Intian valtameressä, sekä Ascension Islandilla Atlantin valtameressä. Keskusvalvonta-asema sijaitsee Colorado Springsissä Yhdysvalloissa Coloradon osavaltiossa. (NAVCEN 1995; 1996.)

Käyttäjäsegmentti koostuu käyttäjien GPS-paikannuslaitteista, jollainen tämänkin opinnäytetyön aikana rakennettiin. Paikannuslaitteen tehtävänä on laskea sijainti satelliitilta saatujen arvojen perusteella ja viestiä tieto käyttötarkoituksen mukaan joko reaaliajassa käyttäjälle tai kolmannelle osapuolelle tai/ja tallentaa tieto myöhempää käyttöä varten. Paikannuslaitteita löytyy nykyään moniin käyttötarkoituksiin, joko karttapohjaisia kuten auto- ja venenavigaattorit tai pelkästään sijainnin, suunnan ja kuljetun etäisyyden näyttäviä esimerkiksi suunnistus-käyttöön sopivia laitteita. (NAVCEN 1995; 1996.)

3.2 GPS-järjestelmän toimintaperiaate

Järjestelmän toiminta perustuu kolmiomittaukseen, jossa avaruussegmentin satelliitit sekä käyttäjäsegmentin päätelaite muodostavat mittauspisteet. Sijaintitiedon muodostamiseksi käyttäjäsegmentin päätelaite mittaa satelliitin lähettämän radiosignaalin matkan kestoa satelliitilta päätelaitteeseen. Tämän mahdollistaa se, että sekä päätelaitteessa, että satelliitissa muodostetaan samaa PRN-koodia, jolloin vertailemalla kahden samanlaisen PRN-koodin sekvenssejä saadaan syntyneen vaihe-eron perusteella laskettua signaalin vastaanottamisen ja lähettämisen välillä tapahtunut viive. Mitattua tulosta kutsutaan pseudoetäisyydeksi. Tällöin on myös tiedettävä, milloin satelliitti on koodin lähettänyt, joten päätelaitteen kello tulee synkronoida satelliitin kellojen kanssa täsmälleen oikeaksi. Synkronointi tapahtuu vertailemalla kahden eri mittauspisteen ja satelliitin lähettämän alkuperäisen tiedon erotussuureita keskenään. Sijaintitieto voidaan siis laskea minimissään kolmella satelliitilla, jolloin pystytään muodostamaan kaksiulotteinen paikannustieto sekä kellovirheen korjaus, mutta ei korkeustietoa. Kaksiulotteista eli 2D-sijaintitietoa voidaan hyödyntää esimerkiksi lentokoneissa, joissa koneen omat mittarit määrittävät korkeuden tai laivoissa, joissa korkeus

meren pinnasta ei ole tarpeellinen tieto. 3D-sijaintitiedon määrittämiseen tarvitaan siis vähintään neljältä satelliitilta saadusta datasta muodostettua neljää eri muuttujaa, X , Y , Z ja kellokorjaus. Tällöin voidaan muodostaa muutamien metrien virhemarginaalilla myös korkeustiedon sisältävä sijaintitieto lausekkeesta:

$$p = \sqrt{(x - X)^2 + (y - Y)^2 + (z - Z)^2} + c(\Delta T_{vast} - \Delta T_{sat})$$

Kaava 1. Sijaintitiedon laskukaava

jossa tunnettuja muuttujia ovat

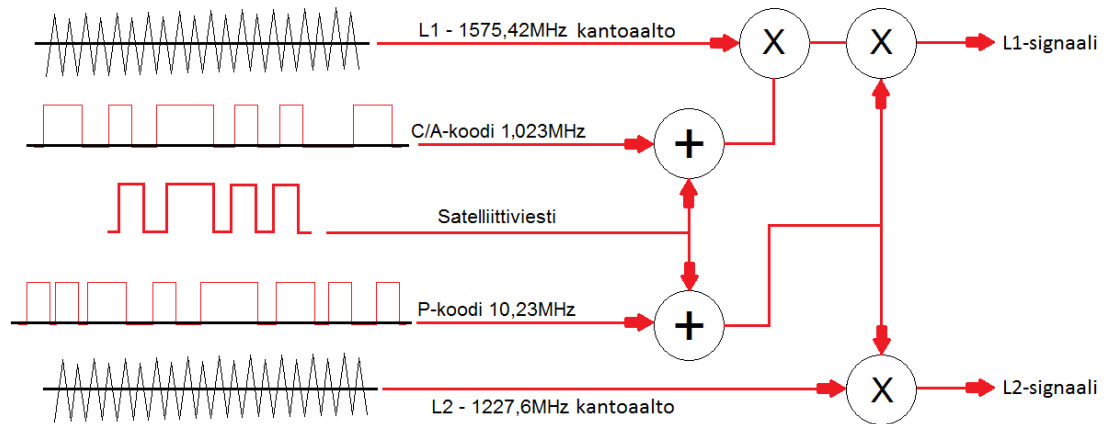
- X , Y , Z = satelliittien sijainti avaruudessa, eli ratatiedot
- ΔT_{sat} = satelliitin lähettämä kellovirhe

joista saadaan laskettua tuntemattomat

- x , y , z = vastaanottimen sijaintitiedot
- ΔT_{vast} = vastaanottimen kellovirhe
- p = kulkuajan perusteella mitattu pseudoetäisyys

Voidaan siis sanoa, että päätelaitteeseen yhteydessä olevien satelliittien ja mittauspisteiden määrän kasvaessa myös mittausvirhe ajan myötä pienenee. Virhettä ei kuitenkaan voida edes matemaattisesti täysin poistaa, josta kerrotaan lisää luvussa 3.3. ”GPS-signaalin virheet”. (NAVCEN 1995; 1996.)

GPS-satelliitit tuottavat cecium-atomikelloillaan 10,23 MHz:n perustaajuutta, johon sisältyy C/A- ja P-koodit. C/A-koodi toimii 1,023 MHz:n taajuudella, joka sisältää siviilikäyttöön tarkoitetun julkisen navigointidatan (Coarse Acquisition). P-koodiin on taas sisällytetty sotilaskäyttöön tarkoitettu salattu navigointidata ja se käyttää 10,23 MHz:n taajuutta. Perustaajuus moduloidaan kahteen kantaaltoon L1 1517,42 MHz ja L2 1227,60 MHz, joiden taajuudet saadaan kertomalla perustaajuus 154:llä ja 120:lla. L1-kantaalto sisältää sekä julkisen C/A-koodin, että P-koodin, kun L2-koodi sisältää ainoastaan sotilaskäyttöön tarkoitetun P-koodin. 10,23 MHz:n perustaajuuteen moduloidaan myös itse 50 MHz:n satelliitiviesti.

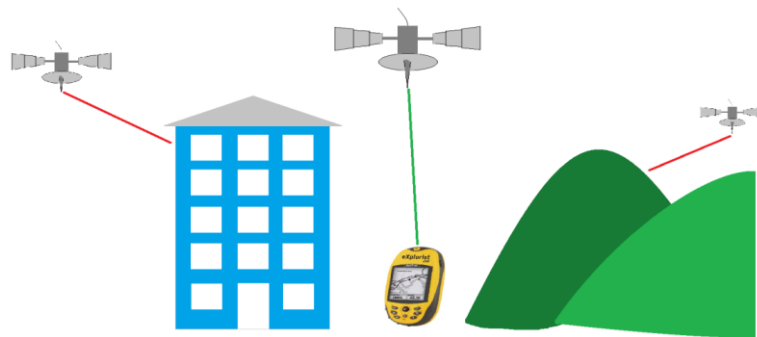


Kuvio 5. GPS-signaalin rakenne

GPS-järjestelmä tuottaa siis jatkuvasti kahta palvelua: Standard Positioning Service ja Precise Positioning Service. SPS-palvelu on kaikille GPS:n käyttäjille avoin palvelu sisältäen ainoastaan C/A-koodin. PPS on sotilaskäyttöön tarkoitettu palvelu, joka muodostetaan sekä C/A-, että P-koodista ja se käyttää toimiakseen molempia L1- ja L2-signaaleja. Näin sotilaskäyttöön tarkoitettua palvelusta saadaan tarkempi ja häiriösuojatumpi SPS-palveluun verrattuna. Lisäksi järjestelmä tuottaa myös L3- ja L4-signaaleja, joista L3 toimii 1381,05 MHz:n taajuudella ja on ns. NUDET-signaali, jota käytetään paikantamaan ydinaseiden laukaisuja tai muita korkean energian omaavia, voimakasta infrapunasaäteilyä tuottavia tapahtumia. L4-signaalia käytetään tutkimuksellisessa käytössä, jonka tarkoituksena on mm. löytää vaihtoehtoinen ionosfäärin aiheuttaman virheen korjausmenetelmä. L4-signaali käyttää 1379,913 MHz:n taajuutta. Vuoden 2010 aikana laukaistava ensimmäinen Block IIF -sukupolven satelliitti tarjoaa myös 1176,45 MHz:n L5-signaalia, joka tullaan valjastamaan siviilikäyttöön SoL-palveluun (Safety of Life). SoL-palvelun tarkoituksena on toimia täydentävänä palveluna esimerkiksi merenkulussa ja ilmailussa, jossa äärimmäisen tarkka ja häiriötön signaali voi tietyissä olosuhteissa olla ihmishenkiä pelastava. (DLR 2007, NAVCEN 1995; 1996.)

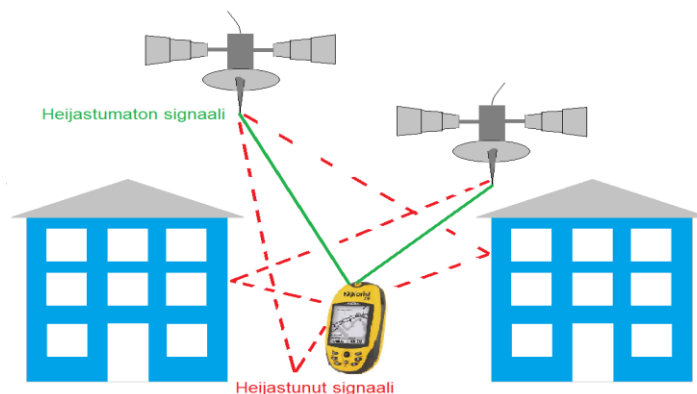
3.3 GPS-signaalin virheet

GPS-järjestelmä ei vielä nykyisilläkään menetelmillä ole täysin aukoton ja virheitä aiheuttaa niin luonto kuin käyttäjätkin. Tyypillisimpiä tavalliselle käyttäjälle aiheutuvia virhetilanteita ovat katvealueet. Katvealue tulee monesti eteen korkeiden rakennusten taajamissa, ahtailla kujilla, tunneleissa, tiheissä metsissä tai muuten vaikeissa maasto-olosuhteissa. On olemassa GPS-laitteita, jotka ajoneuvoon asennetun takometrin tai gyroskoopin avulla pystyvät laskemaan signaalin katkeamisen jälkeenkin laitteen sijainnin. Tällaisia laitteita käytetään esimerkiksi kaivostyömailla, jolloin päätelaite voi toimia syvällä kallion sisällä signaalin katkettua ilman ongelmia.



Kuvio 6. Katvealue

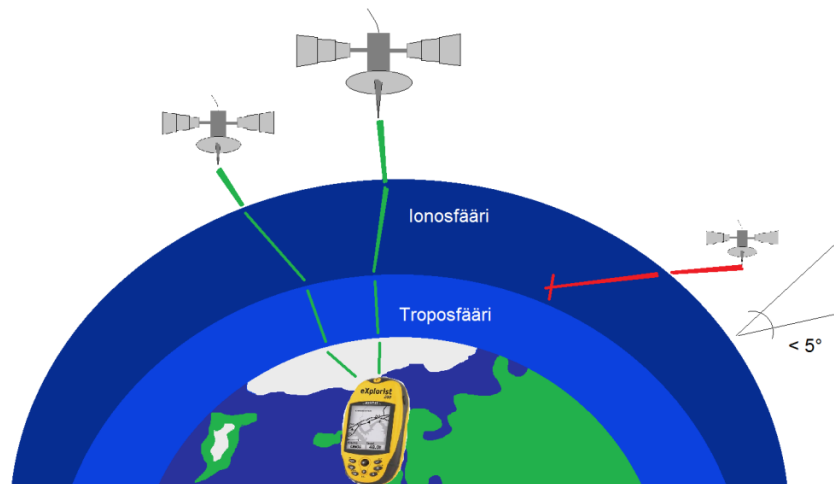
Toinen yleinen virheenaiheuttaja on ns. monitieheijastuminen. Vastaanottimen ympäriltä heijastuu antenniin signaaleja, jotka vääristävät oikeaa signaalia tai laite havaitsee heijastuneen ja viivästyneen signaalin. Monitieheijastumisen voi aiheuttaa esimerkiksi rakennukset, veden pinnan heijastukset, valkea lumihanki tai ikkunat.



Kuvio 7. Monitieheijastuminen

Käyttäjistä tai hänen sijainnistaan riippumattomia virheitä on mm. ratavirhe. Ratavirhe johtuu maapallon vetovoiman tai aurinkotuulien aiheuttamista muutoksista satelliittien kiertoradoissa. Ratavirheitä korjataan jatkuvasti maa-asemilta, kiertoradan muutoksen kestäessä vuorokauden, joten ratavirheestä johtuvan virheen suuruus voi ajoittain hieman vaihdella.

Ilmakehän vaikutus signaalin kulkuun johtuu auringon aktiivisuuden aiheuttamista maapallon ilmakehän ionosfäärin elektronitasojen muutoksista, troposfäärin vesihöyryjen ja muiden tiheiden kaasujen, sekä ilmakehän epäpuhtauksien koostumuksista. Tyhjiössä radioaallot kulkevat valon nopeudella, mutta hidastuvat heti ilmakehän kohdattuaan, joten häiriötä signaalissa ilmenee aina. Troposfäärin olosuhteiden ja täten signaaliin kohdistuvan vaikutuksen ennustettavuus on mahdotonta, joten troposfäärissä aiheutuneita virheitä ei pystytä täysin korjaamaan. Ionosfäärin elektroniset ominaisuudet tunnetaan paremmin ja sen sähköisten ominaisuuksien aiheuttamia virheitä pystytään ehkäisemään. Virheen vaikutusta voidaan pienentää ionosfääri- ja troposfäärimallinnuksilla, jolloin ilmakehän vaikutus paikannustarkkuuden heikkenemiseen saadaan yleensä rajattua muutamaan metriin. Lisäksi GPS-järjestelmä on rakennettu siten, että päätelaite ei huomioi alle viiden asteen kulmassa horisontissa olevia satelliitteja, estäen loivassa kulmassa ilmakehään ja rajusti vääristyvien signaalien käytön.



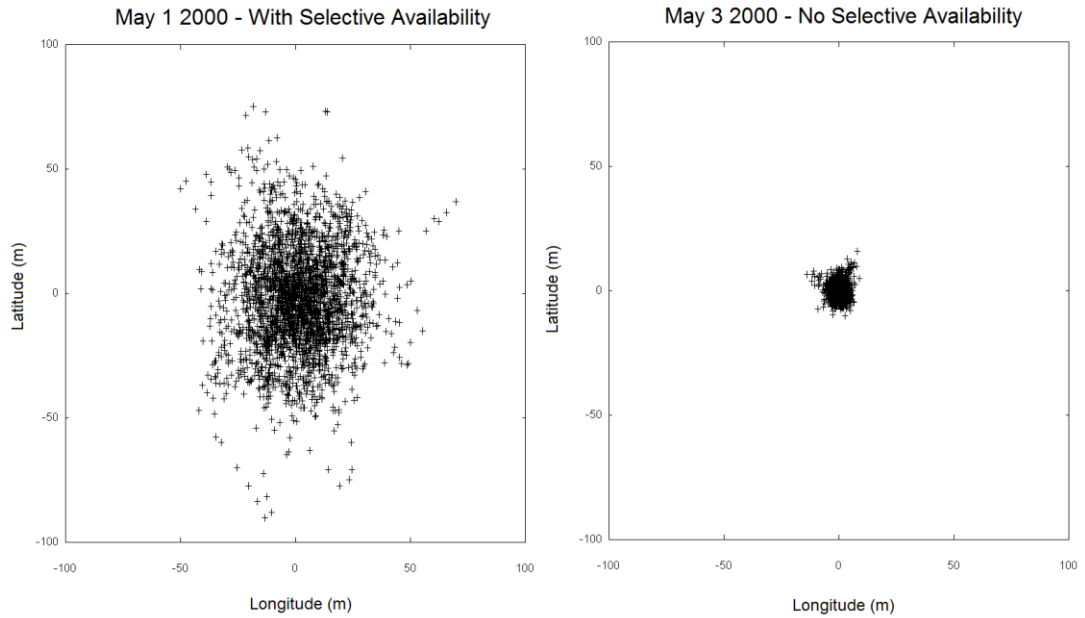
Kuvio 8. Ilmakehän vaikutus signaaliin

Huono satelliittigeometria aiheuttaa myös virheitä paikannustarkkuuteen. Satelliitit voivat tietyissä tilanteissa olla kiertoradoillaan liian lähellä toisiaan päätelaitteeseen nähden, jolloin paikannustarkkuus pienenee. Ongelmaan ei ole tarvetta pyrkiä löytämään ratkaisua, koska virhe on yleensä hetkellinen ja korjaantuu, kun satelliitit liikkuvat radoillaan tai päätelaite vaihtaa sijaintiaan.

Kellovirheet johtuvat satelliittien cesium-atomikellojen alttiudesta pienille muutoksille. Muutokset kelloissa ovat yleensä niin mitättömiä, ettei niiden toimintaan tarvitse puuttua maa-asemalta, mutta tarpeen tullen kellojen toimintaa voidaan säädellä. Kellovirheen osuus paikannustarkkuuden heikentymisessä on reilusti alle metrin luokkaa, joten suuresta vääristymisestä ei yleensä ole kyse.

3.4 SA-häirintä

Vuoteen 2000 asti Yhdysvaltain puolustusministeriö (DoD, Department of Defence) tarkoituksellisesti häiritsi GPS:n C/A-koodia alentaakseen paikannustarkkuutta. Menetelmää kutsuttiin SA-häirinnäksi (Selective Availability). Menetelmän tarkoituksena oli ehkäistä liian tarkka paikannustarkkuus vapaasti saatavasta C/A-koodista, jota voitaisiin käyttää kriisitilanteissa ja sotilaallisissa operaatioissa Yhdysvaltoja vastaan. Häirintä tapahtui yksinkertaisesti lisäämällä ylimääräisiä kertoimia satelliittien lähettämiin ratatietoihin tai lisäämällä ylimääräistä kohinaa PRN-signaaliin, jolloin paikannustarkkuus putosi 100 metriin. Toukokuun 1. vuonna 2000 Yhdysvaltain presidentti Bill Clinton julkaisi tiedotteen, jossa Yhdysvaltain puolustusministeriö luopuu SA-häirinnästä, eli ns. asettaa kertoimet noltaan jo seuraavan puolen yön aikaan. Tästä hetkestä alkoi GPS-laitteiden yleistyminen siviilikäytössä paikannusvirheen pudotessa ”yhdes- sä yössä” kymmenesosaan häirittyyn signaaliin verrattuna. Syyskuussa 2007 DoD tiedotti, että seuraavan sukupolven satelliiteista SA-häirinnän mahdollista- vat ominaisuudet on poistettu pysyvästi. (PNT 2010; Valkoisen talon tiedote 2000.)

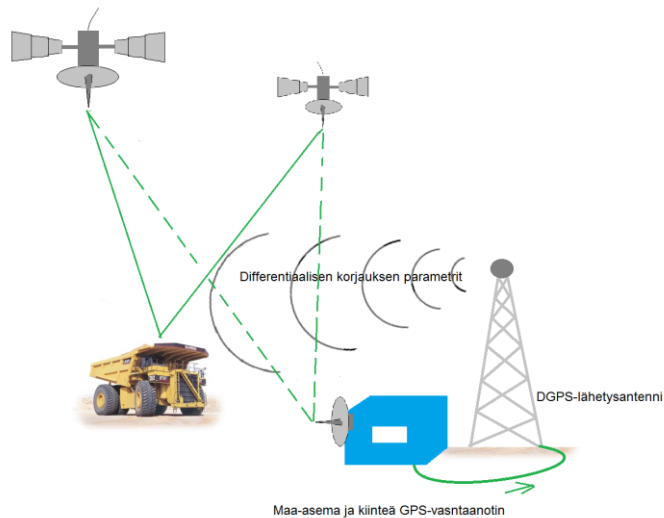


Kuvio 9. GPS-järjestelmän tarkkuus ennen ja jälkeen SA-häirinnän poistamisen (PNT 2010)

3.5 GPS-järjestelmän virheenkorjausmenetelmät

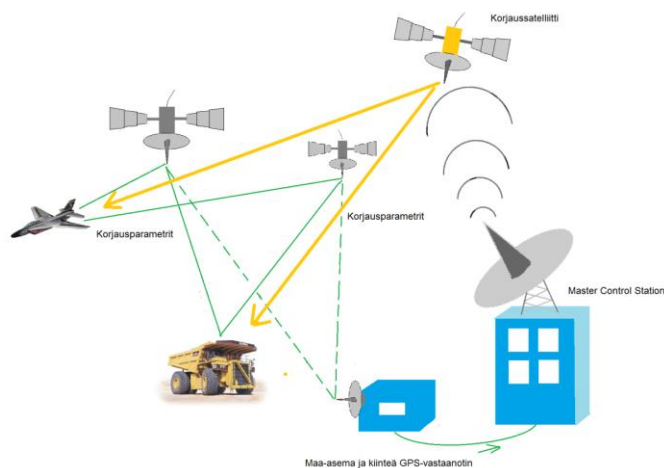
DGPS:ssä eli differentiaalisessa GPS:ssä järjestelmän osaksi on liitetty maa-asemia, joihin on yhdistetty tarkan sijaintinsa tietäviä kiinteitä GPS-vastaanottimia. Vastaanottimet vastaanottavat jatkuvasti sijaintitietoa satelliiteilta, vertailevat sitä omaan tarkkaan sijaintiinsa, laskevat satelliitilta saadusta signaalista kunkin satelliitin hetkellisen virheen, jonka jälkeen differentiaalinen korjaustieto lähetetään maa-asemien kautta radioteitse käyttäjän GPS-vastaanottimeen. DGPS-menetelmällä päästään 1-3 metrin paikannustarkkuuteen ja ajoittain jopa muutamien kymmenien senttimetrien tarkkuuteen. (AMSA 2007.)

Suomessa DGPS-palvelua tarjoaa kaksi tahoa, joista Digitan jakeluverkkoa hyödyntävä Indagon Oy:n FOKUS-palvelu on tarkoitettu maalla tapahtuvaan navigointiin, sekä Merenkululaitoksen tarjoama DGNSS merenkulun käyttöön. (Indagon Oy 2010; Merenkululaitos 2010.)



Kuvio 10. Differentiaalisen GPS:n toimintaperiaate

EGNOS (European Geostationary Navigation Overlay System) ja WAAS (Wide Area Augmentation System) järjestelmät toimivat pitkälti samalla periaatteella kuin DGPS, mutta näissä järjestelmistä korjaustietoa ei lähetetä käyttäjälle suoraan radioteitse, vaan korjaustiedot lähetetään kontrolliaseman (MCS) välityksellä korjaussatelliittiin, joka lähettää tiedon eteenpäin käyttäjälle. Tällöin useita lähetyverkkoja ei tarvita, vaan satelliitti pystyy lähettämään korjaustiedot laajalle alueelle ja ennen kaikkea myös lentoliikenteelle. EGNOS on Euroopan kattava järjestelmä ja WAAS Pohjois-Amerikan kattava järjestelmä. Japanilla on oma vastaava MSAS-järjestelmä (Multifunctional Transport Satellite-based Augmentation System). (ESA 2009; 2010, FAA 2010.)



Kuvio 11. EGNOS- ja WAAS-järjestelmien toimintaperiaate

4 GPS-LOGGERI

4.1 Laitteen määrittely

Lähtökohtaisesti haluttiin valmistaa kilpailukäyttöön soveltuva laite, joka tallentaa kilpailun aikana kuljetun reitin ja nopeuden myöhempää tarkastelua varten. GPS-paikkadatan tulisi olla luettavissa Windows PC:llä jälkikäteen, koska reaaliaikainen tiedonkeruu on ralliautoilussa kielletty.

Laitteen käyttöliittymän haluttiin olevan selkeä ja yksinkertainen, mutta samalla sen tulisi kertoa käyttäjälle laitteen tila. Käyttöliittymän suunnitteluvaiheessa tulisi ottaa huomioon laitteen käyttötarkoitus, jolloin laitteen käytön ei tulisi häiritä suoritusta. Laite tulisi olemaan kiinteästi autoon kiinnitettävä ja sen käyttöjännite saataisiin auton 12 V:n sähköjärjestelmästä.

Käytännössä laite tulisi siis koostumaan kolmesta eri osa-alueesta, GPS-vastaanottimesta, tallennusmediasta ja AVR-mikrokontrollerista. PC-puolella ohjelmistot ja laitetuki olisi valmiina ja laite tultaisiin rakentamaan näiden valmiina olevien sovellusten ehdoilla. GPS-vastaanottimen tulisi näyttää sijaintitieto helposti mikrokontrollerilla luettavassa ja muokattavassa muodossa, jolloin tärkeimpänä vaatimuksena vastaanottimen valinnalle oli NMEA 0138 -standardin mukaista sarjamuotoista dataa tarjoava moduuli. Reitti haluttiin tallentaa Google Earth -sovelluksen käyttämään kml-tiedostomuotoon, jolloin reitti saadaan piirrettyä suoraan sovelluksen karttapohjalle tai satelliittikuvan päälle (Google kml-tutorial 2010). Massamuistiksi valittiin SD-muistikortti, joka tarvitsisi myös FAT-tiedostojärjestelmän. Tiedostojärjestelmän luku- ja kirjoitusrutiineita hoitamaan valittiin vapaasti saatavilla oleva, pieniin, ei-kaupallisiin sovelluksiin tarkoitettu FatFS-kirjasto.

4.2.1 Trimble GPS -moduuli ja NMEA 0138 -lausekkeet

Työssä käytetty GPS-moduuli on Trimblen valmistama, ajoneuvokäyttöön ja erityisesti vaativiin olosuhteisiin suunniteltu Lassen DR+GPS. Moduulin tärkeimpiin ominaisuuksiin kuuluvat:

- tuki NMEA 0183-, HIPPO-, sekä Trimble binary -protokollille

- CMOS-tasoinen sarjaliikenneväylä @ 38400 baud
- 3,3 Vdc \pm 0,2 V käyttöjännite
- 5 Vdc käyttöjännite aktiiviselle antennille
- sijainnin päivitysnopeus 1Hz tai 5Hz
- horisontaalinen tarkkuus < 5 metriä
- korkeustarkkuus < 10 metriä
- liiketarkkuus < 0,02 m/s
- tuki analogiselle gyroskoopille sekä liiketunnistimelle.

Lisäksi hyvää tuotekehittelyä ja GPS-tekniikkaan tutustumiskäyttöä tukee monipuoliset valmistajan tarjoavat Windows-pohjaiset moduulin konfigurointityökalut, joilla laitteen käyttämiä ominaisuuksia voidaan muokata käyttötarpeen mukaan. (Trimble 2010.)

Työssä moduuli asetettiin tilaan, jossa käytetään NMEA 0138 -protokollaa ja sen RMC-lauseita, joista saadaan kaikki tarvittava tieto reitin ja nopeuden tallentamista varten. Moduulin päivitysnopeus asetettiin 1Hz:iin eli laite tallentaa sijaintipisteen sekunnin välein ja muodostaa pisteistä kuljetun reitin.



Kuvio 12. Trimble Lassen DR+GPS -moduuli ja aktiivinen GPS-antenni

NMEA 0138 on sarjamuotoista, asynkronista ASCII-pohjaista dataa, jossa standardinmukaiset NMEA-lausekkeet aloitetaan aina dollarisymbolilla ja laite tunnisteella "\$GP"-muodossa sekä lausekkeiden eri solut erotellaan pilkuilla omiin kenttiinsä (ns. Comma Separated Values, CSV). Vaikka moduuli ei pystyisi kentän dataa näyttämäänäkään, on lauseessa silti tyhjä pilkuilla eroteltu kenttä, joka mahdollistaa rakennetussa laitteessa ohjelmallisesti määrittämään laitteen yhteyden tilan. NMEA 0138 -standardi määrittelee kaiken kaikkiaan seitsemän eri

lausetyyppejä, joilla voidaan mahdollistaa periaatteessa kaikenlainen navigaatio ja paikannus niin merellä, maalla kuin ilmassakin. Esimerkiksi työssä käytetty RMC-lause (Recommended Minimum Specific GPS/TRANSIT Data) NMEA 0138 -muodossa soluittain näyttää seuraavalta:

```
$GPRMC,hhmmss.ss,A,IIII.II,a,x.x,x.x,ddmmyy,x.x,a*hh
```

\$	= NMEA-lauseen aloitusmerkki
GP	= laitetunniste (GPS)
RMC	= lausetyyppi
1	= UTC-aika
2	= yhteyden tila, A(automatic), V(void / system warning)
3	= latitudi
4	= pohjoista vai eteläistä leveyttä (N tai S)
5	= longitudi
6	= itäistä vai läntistä pituutta (E tai W)
7	= nopeus solmuissa
8	= suunta
9	= UT-päiväys
10	= variaatio
11	= E tai W
12	= tarkistussumma

Moduuli testattiin toimivaksi yhdistämällä moduuli sarjaliikenneväylän kautta kannettavaan tietokoneeseen ja valmistajan tarjoamia testaussovelluksia käyttämällä tallennettiin lyhyt autolla ajettu reitti, jolloin varmistettiin moduulin paikannustarkkuus raja-arvojen sisällä.

Signaalitasot mitattiin oskilloskoopilla, eikä niissä huomattu virheen aiheuttavia poikkeavuuksia. GPS-moduuli käyttää TTL-tasoista sarjaliikennettä eli arvot vaihtelevat 0–3 V:n välillä.

Moduulin käyttökuntoon asettumisen, kylmä- ja kuumakäynnistymisen eli TTFF,

Hot Start, ja Hot Reset -ajat tarkistettiin Trimblen ohjelmistoilla. Tarkistuksen perusteella saatiin tietoon, että kun laitteeseen kytketään virta, on moduuli saanut ensimmäisen sijaintitiedon noin minuutissa (Time To First Fix), sekä lyhyen virtakatkoksen jälkeen sijaintitieto saadaan uudelleen muutamassa sekunnissa (Hot Reset).

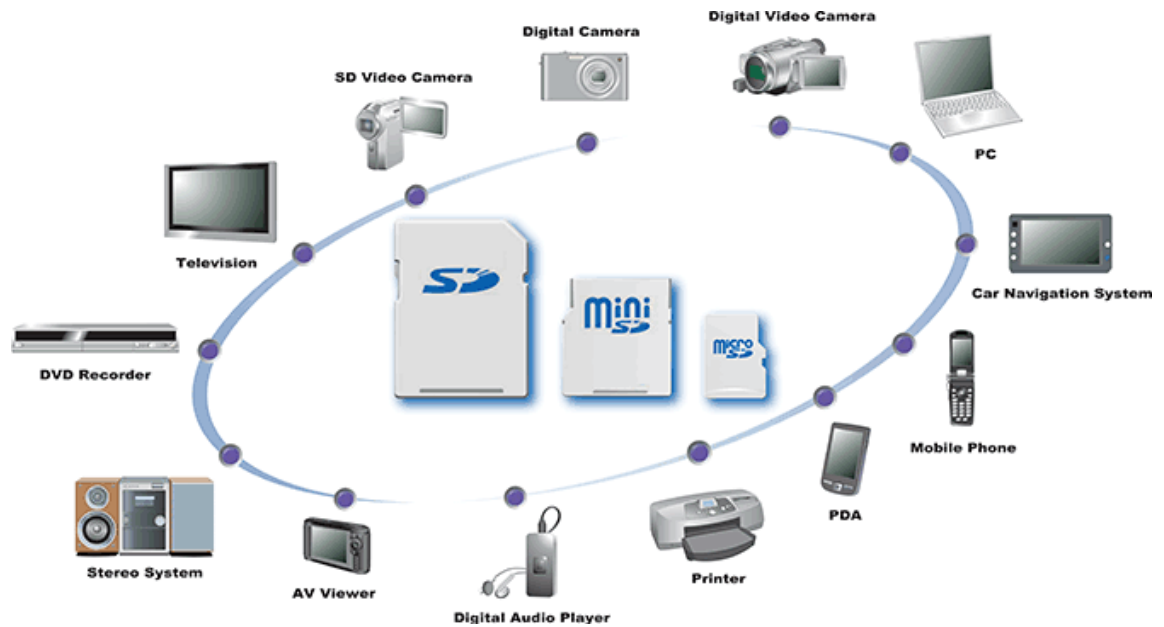
Koska moduulin konfigurointi on tehty ohjelmallisesti jo ennen työssä valmistettuun laitteeseen asentamista, ei mikrokontrollerilta tarvitse lähettää komentoja GPS-moduulille, eikä täten tarvittu mikrokontrollerin ja moduulin välille signaalitason muokkaamista 5 V:sta 3,3 V:iin, vaan moduuli kytkettiin ainoastaan TxD-linjan (Transmit Data) osalta mikrokontrollerin toisen USART-väylän RxD-linjaan (Receive Data) siis liikenne tapahtuu ainoastaan moduulista mikrokontrolleriin.

4.1.2 Secure Digital (SD) -muistikortti ja sen liittäminen SPI-väylään

SD-muistikortti on yleisesti kannettavissa laitteissa käytetty non-volatile eli jatkuvasta käyttöjännitteestä riippumaton muistityyppi. Muistityypin ovat kehittäneet yhteistyössä Panasonic, SanDisk, sekä Toshiba. SD-muistityyppiä käytetään laajalti mm. digikameroissa, matkapuhelimissa, viihdelaitteissa sekä GPS-laitteissa. Nykyään SD-muistikorttityyppiä kehittää ja hallinnoi em. yritysten perustama SD Card Association.

SD-kortin tärkeimpiä ominaisuuksia ovat pieni virrankulutus ja suuri tallennuskapasiteetti pienessä koossa. SD-muistikortit toimivat 2,7–3,7 V:n käyttöjännitteellä, virrankulutuksen ollessa ainoastaan muutamia milliampeereja. Korttien tallennuskapasiteetit lähtevät aina 1MB:stä alkaen, vuoden 2009 aikana saavutettuun jopa 32GB:n kapasiteettiin asti. Vuonna 2009 julkistettujen tutkimustulosten myötä on lähivuosina odotettavissa jopa 2TB:n kapasiteetilla varustettuja SD-muistikortteja (SD Card Association 2010).

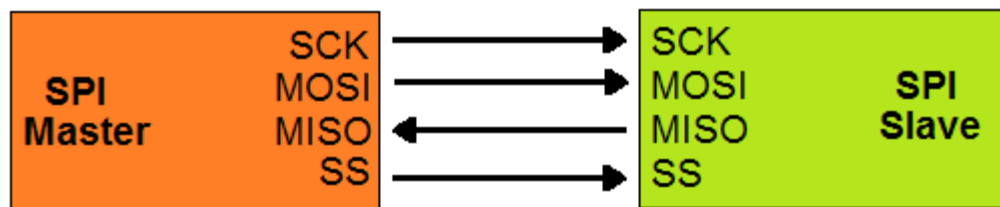
SD-muistikorttityyppejä on olemassa kolmea fyysiseltä kooltaan erilaista mallia SD-Card, MiniSD-Card ja MicroSD-Card. Toiminnaltaan tai virrankulutukseltaan eri tyypit eivät toisistaan eroa, mutta muistikapasiteetti pienimmissä malleissa on hieman ”normaalia” SD-Cardia pienempi jääden yleisesti maksimissaan 4–8 GB:n luokkaan (SD Card Association 2010).



Kuvio 13. SD-muistikorttityypit ja käyttösovelluksia (SD Association 2010)

Tässä työssä päädyttiin käyttämään SD-Card -tyyppistä muistikorttia, koska pienempään korttiin ei käyttötarkoituksen ja helppokäyttöisyyden säilyttämisen vuoksi nähty tarvetta.

Laitteessa kortti on liitetty standardin mukaisesti mikrokontrollerin SPI-väylään, jossa Atmega644P-mikrokontrolleri toimii master- ja SD-muistikortti slave-tilassa. SPI-tiedonsiirto vaatii aina neljä käytettyä signaalitasoa: SCK (kellopuls- si), MOSI (Master Out / Slave In), MISO (Master In / Slave Out) ja SS(Slave Se- lect).

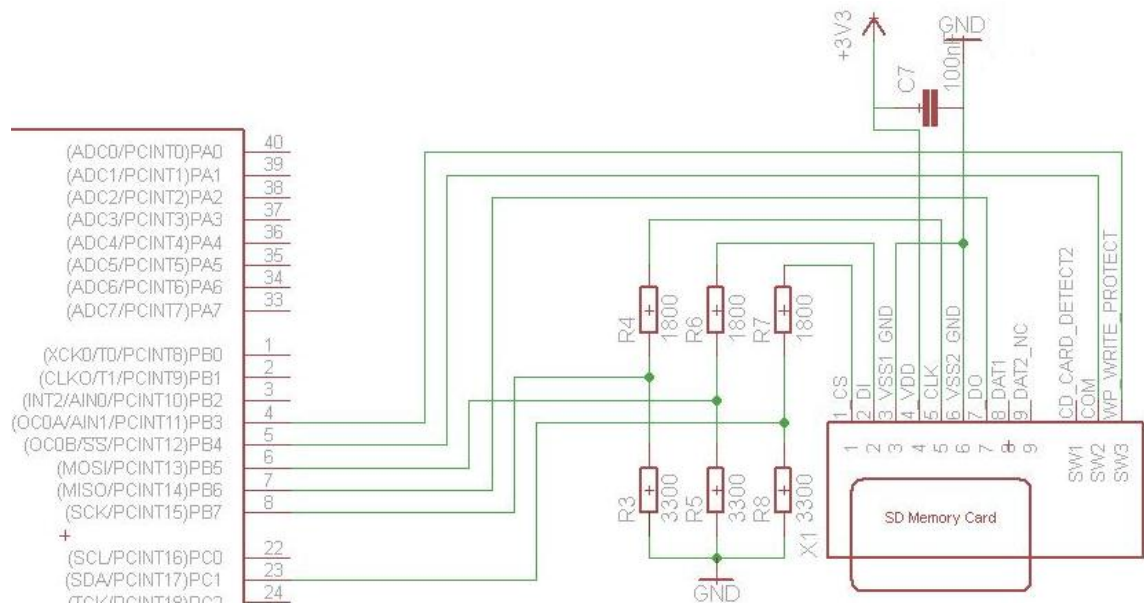


Kuvio 14. SPI-väylän liitântätapa

Master-laite määrittää tiedonsiirtoparametrit sekä pakottaa muistikortin slave-tilaan (SS). Master-laitteen lähetettyä SCK-kellopuls- sin slave-laite osaa valmistautua datan vastaanottamiseen. Tällöin mitään eri osoitteistoja ei tarvitse käyttää, vaan tiedonsiirto tapahtuu ainoastaan master- ja slave-laitteiden välillä. SPI-väylä toimii Full Duplex -tilassa, jolloin master-laitteen MOSI-väylää pitkin lähet-

tämä signaali lähetetään takaisin MISO-väylää pitkin master-laitteelle, joka vertailee lähetetyn ja vastaanotetun datan oikeellisuutta, joten Full Duplex -moodia käyttämällä varmistetaan siitä, että mikrokontrollerin lähettämä data on tahdistettu kellopulssin mukaan oikein eikä data korruptoidu. (Microchip Technology Inc. 2010.)

Muistikortin kytkennässä mikrokontrolleriin tuli ottaa huomioon laitteiden käyttöjännite. Mikrokontrollerin käyttäessä 5 V:n jännitettä tuli MISO-, MOSI- ja SCK -väylien jännitettä pudottaa SD-korttistandardin sallimiin rajoihin eli 2,7–3,5 V:iin. Tämä toteutettiin asettamalla signaaliväyliin 1,8 k Ω ja 3,3 k Ω jännitteenjakovastukset, jolloin signaaliväylien jännite saatiin asetettua 3,25 V:iin. Muistikortin 3,3 V:n käyttöjännite otetaan suoraan laitteen virransyötön LM3940-regulaattorilta, mutta samaa jännitteenjakomenetelmää käyttäen muistikortin käyttöjännite olisi voitu ottaa yhdestä mikrokontrollerin vapaasta pinnistä, mikä taas olisi mahdollistanut SPI-väylän ja muistikortin lepotilan ohjaamisen mikrokontrollerilla ja näin kyetty hieman pienentämään virrankulutusta.



Kuvio 15. Muistikortin kytkentä mikrokontrolleriin

4.1.3 AVR Atmega644P -mikrokontrolleri

AVR Atmega644P -mikrokontrolleri kuuluu Atmelin 8-bittiseen RISC-ytimelliseen mikrokontrolleriperheeseen. AVR-mikrokontrollereille tyypillisiä ominaisuuksia ovat useat I/O-väylät, UART/USART/SPI-liitännät, Flash-ohjelmamuisti aina 256 Kb:iin asti, SRAM-käyttömuisti, PWM-pulssinleveysmodulaattori, sisäänrakennetut A/D-muuntimet, keskeytysajastimet sekä sisäinen kello-oskillaattori. Työssä käytetyn Atmega644P:n tärkeimpiä ominaisuuksia ovat:

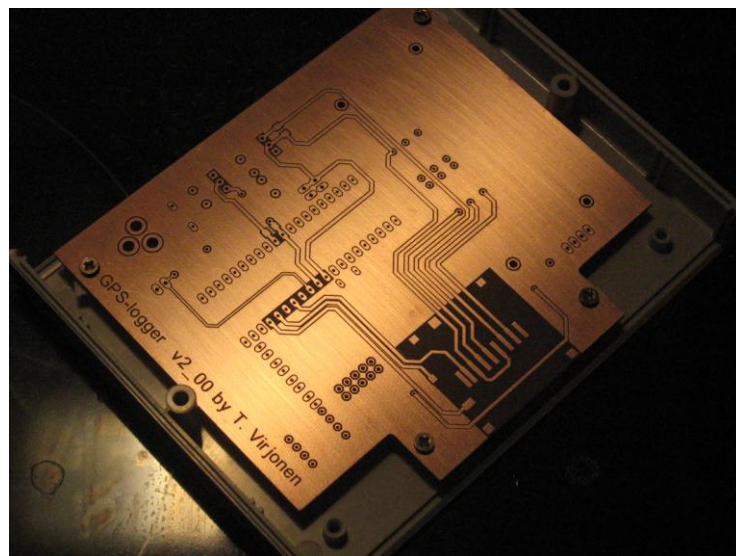
- Maksimissaan 20 MHz:n kellotaajuus
- 64 Kb:n flash-ohjelmamuisti
- 2 Kb:n EEPROM-muisti
- 4 Kb:n SRAM-käyttömuisti
- 32 ohjelmoitavaa I/O-väylää
- 8-kanavainen, 10-bittinen A/D-muunnin
- 2 ohjelmoitavaa USART-väylää
- SPI-väylä
- 2,7–5,5 V:n käyttöjännite.

Tässä työssä Atmega644P on kytketty toimimaan 5 V:n käyttöjännitteellä ja 8 MHz:n ulkoisen oskillaattorin tuottamalla kellotaajuudella. Mikrokontrollerista käytetään toista USART-väylää GPS-moduulilta saatavan datan vastaanottamiseen, SPI-väylää muistikortin liittämiseen sekä kuutta I/O-väylää laitteen käyttöliittymää eli painikkeita ja led-merkkivaloja varten. Atmega644P on ominaisuuksiltaan varsin järeä tähän työhön, mutta alun perin käytettäväksi alunperin suunnitellun Atmega32-mikrokontrollerin 32 Kb:n ohjelmamuistin koko tuli FatFS-tiedostojärjestelmää käytettäessä nopeasti vastaan, joten päädyttiin hieman yllimitoitetuista ominaisuuksista huolimatta nyt käytettyyn kontrolleriin. (Atmel 2010.)

4.2 Laitteen fyysinen suunnittelu ja toteutus

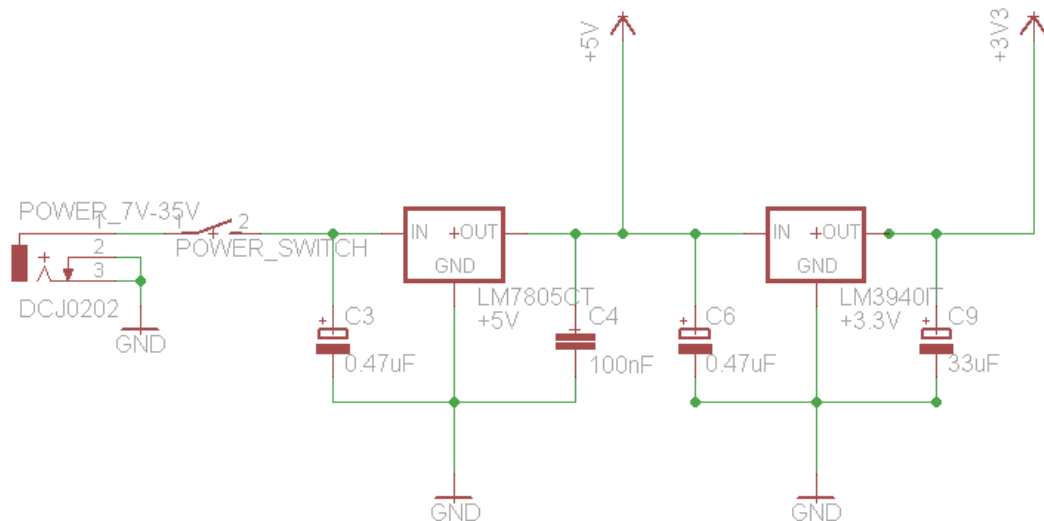
Suunnittelussa pyrittiin ottamaan huomioon yleisesti käytetyt piirilevy suunnittelun säännöt, kuten linjojen veto tarpeeksi etäällä toisistaan sekä välttämään 90 asteen kulmia ylimääräisen signaalikohinan välttämiseksi. Alusta asti pyrittiin huomioimaan myös kokoamisvaihetta helpottavat seikat sekä valmiin laitteen kannalta järkevät ratkaisut komponenttien sijoittelussa jakamalla esimerkiksi virransyöttö, käyttöliittymän merkkivalot ja painikkeet, sekä GPS-moduuli liitäntöineen selkeästi omiin ryhmiinsä.

Laitteen kehityksen aikana valmistettiin kaiken kaikkiaan kolme eri piirilevyversiota. Ensimmäinen versio oli täysin prototyyppiasteella oleva malli tarkoitukseen ainoastaan mahdollistaa laitteen ohjelmiston ja toiminnan testaaminen. Toiseen versioon korjattiin kaikki prototyyppivaiheessa huomatu puutteet ja virheet sekä optimoitiin komponenttien sijoittelua. Laite saatiinkin jo tässä vaiheessa täysin toimintakuntoiseksi. Kolmanteen eli 2.0-versioon piirilevy muokattiin lopulliseen fyysiseen kokoonsa lopullisen laitekotelon määrätessä mitat. Laitekoteloksi valittiin Multicompin valmistama ABS-muovinen, mitoiltaan 140 mm x 110 mm x 35 mm (L x S x K) laitekotelo, johon valmistettiin etu- ja takapaneeli jyrsimällä ne mittoihinsa piirilevystä. Tarkemmin piirilevyn kytkennöistä liitteessä 2 ”GPS-loggerin kytkentäkaavio”.



Kuvio 16. Lopullinen piirilevyversio laitekotelossaan

Laitteen virransyötön haluttiin kohtalaisen laaja käyttöjännitealue, jotta laitetta voitaisiin käyttää tarpeen vaatiessa muunkin tyyppisissä ajoneuvoissa tai paristokäyttöisenä. Laitteen komponentit käyttävät, sekä 5 V:n että 3,3 V:n käyttöjännitteitä, joten virransyötön rakentaminen päätettiin toteuttaa kahdella jänniteregulaattorilla. Regulaattoreista ensimmäinen, LM7085, reguloi ajoneuvon sähköjärjestelmästä otettavan jännitteen 5 V:iin ja LM7085:ltä käyttöjännitteen ottava toinen jänniteregulaattori, LM3940 3,3 V:iin. LM7085-regulaattorin spesifikaation mukaan laitteelle voidaan syöttää 7-35VDC jännitettä, jotta molemmat tarvittavat käyttöjännitteet saadaan reguloitua. Tasaisen käyttöjännitteen varmistukseksi asetettiin, sekä mikrokontrollerin, GPS-moduulin että SD-muistikortin jännitelinjoihin 100nF kondensaattorit. Varsinkin SD-muistikortin käynnistysvaiheessa muistikortti voi haukata hetkellisesti useammankin kymmenen milliampeerin virran normaalin muutaman milliampeerin käyttövirrasta poiketen, jonka seurauksena mahdollisesti aiheutuvan DC-käyttöjännitteen notkahduksen aiheuttama mikrokontrollerin tai GPS-moduulin resetoituminen pystyttiin eliminoimaan.



Kuvio 17. Käyttöjännitteen regulaattoreiden kytkennät

Laitteen käyttöliittymä koostuu kahdesta painikkeesta ja neljästä merkkiledistä. Vihreä ledi on merkinä virran ollessa kytkettynä. Sininen "No GPS Fix" -ledi indikoi GPS-yhteyden tilaa. Merkkivalon palaessa laitteella ei ole "fixiä" eli sijaintitietoa. "DISK ERROR" -ledi indikoi levyvirhettä, joka syntyy jos tiedoston luomisessa, avaamisessa tai muistikortin alustuksessa tapahtuu virhe. Tallennusledi keskellä olevan muistikorttipaikan vieressä syttyy kun tallennus alkaa ja sam-

muu vilkkuen kun tallennus päättyy. Painikkeista punainen aloittaa ja musta lopettaa tallennuksen.



Kuvio 18. GPS-loggerin käyttöliittymä

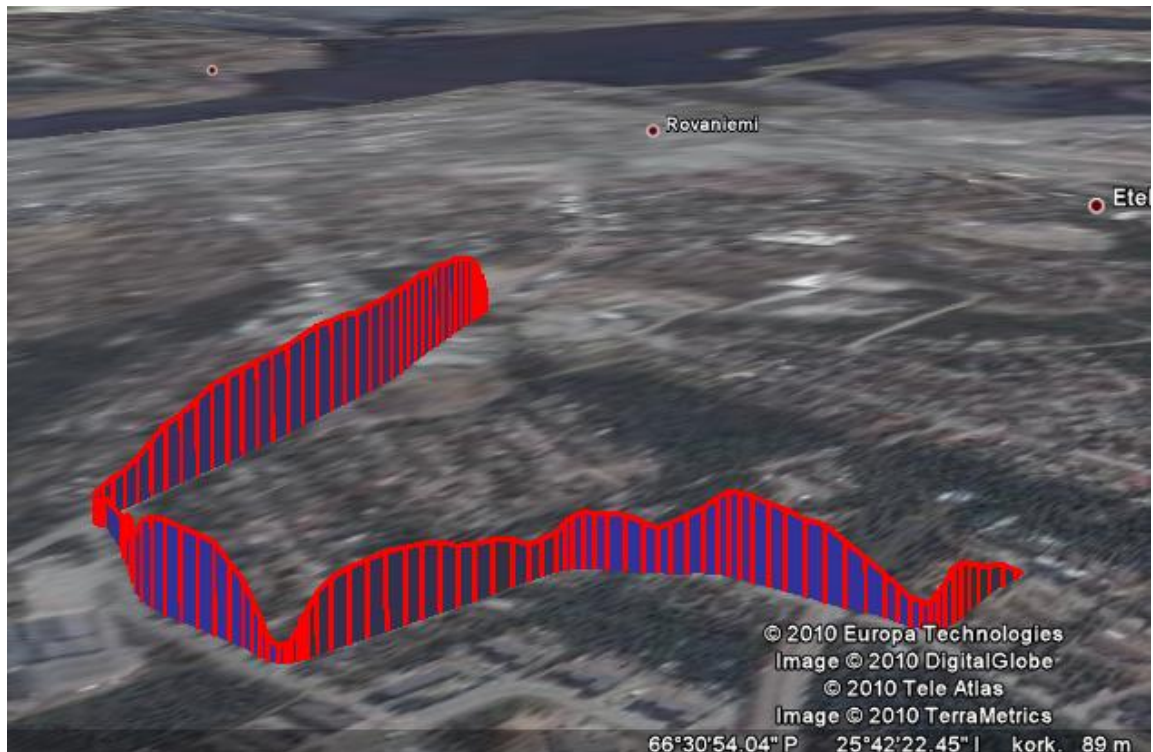
4.3 Ohjelmiston suunnittelu ja toiminta

4.3.1 Ohjelmiston suunnittelu ja sen ominaisuudet

Ohjelmistoa alettiin suunnitella päämääränä saada tallennettua sijaintitiedot sekä nopeus helposti luettavaan muotoon. Tähän tarkoitukseen käyttökelpoisimpia lienee nykypäivänä ilmaiseksi saatavilla olevat karttasovellukset, kuten Google Earth, johon reitti saadaan piirrettyä suoraan kartan tai satelliittikuvan päälle kml-formaatissa. Kml on Google Earth -karttasovelluksen käyttämä xml-merkkaukielinen tiedostomuoto, jonka monipuolisilla merkkausalgoritmeilla onnistuu reittien, rakennusten, mielenkiintoisten kohteiden, korkeuserojen yms. monipuolinen koordinaattipohjainen piirtäminen Googlen karttasovelluksissa.

Käytännössä lähdekoodia alettiin rakentaa kahdesta tärkeimmästä osaluueesta, sarjaliikenteisen NMEA-viestin lukemisesta ja tiedon tallentamisesta muistikortille. Reitin muodostamiseen tarvittavat koordinaatit ja nopeustieto parsroidaan GPS-moduulin lähettämästä sarjaliikennemuotoisesta NMEA-datasta. Koordinaattien tallentamiseen fyysiselle medialle tarvitaan pelkän muistikortin lisäksi myös FAT-tiedostojärjestelmä hoitamaan tarvittavat rutiinit tiedon tallentamista varten. Levyrutiinien hoitamiseen valittiin harrastelijapiireissä yleisesti käytetty FatFS-moduuli, joka koostuu useista kirjastokokonaisuuksista ja se on muokattavissa monille eri valmistajien alustoille. Ohjelman kirjoittamiseen käytettiin Atmelin AVR Studio 4 -ohjelmistoa, joka on tarkoitettu 8-bittisten AVR pohjaisten laitteiden suunnitteluun ja kehittämiseen. Koodi kirjoitettiin C-kielellä ja kääntäjänä käytettiin AVR Studion avr-gcc -lisäosaa.

Tässä työssä GPS-moduulista saatavista koordinaattipisteistä muodostetaan kuljettu reitti Google Earth -sovellukseen. Samalla kuljettuun reittiin lisätään jokaisessa koordinaattipisteessä ollut nopeus, joka ”huijataan” näkymään Google Earth -sovelluksessa absoluuttisena korkeutena maan pinnasta eli reitti ja nopeus näkyvät alla olevan kuvan mukaisesti palkin korkeuden maan pinnasta metreissä myötäillessä nopeutta kilometreissä tunnissa.



Kuvio 19. Valmiin laitteen tallentama lyhyt reitti Google Earth -sovelluksessa

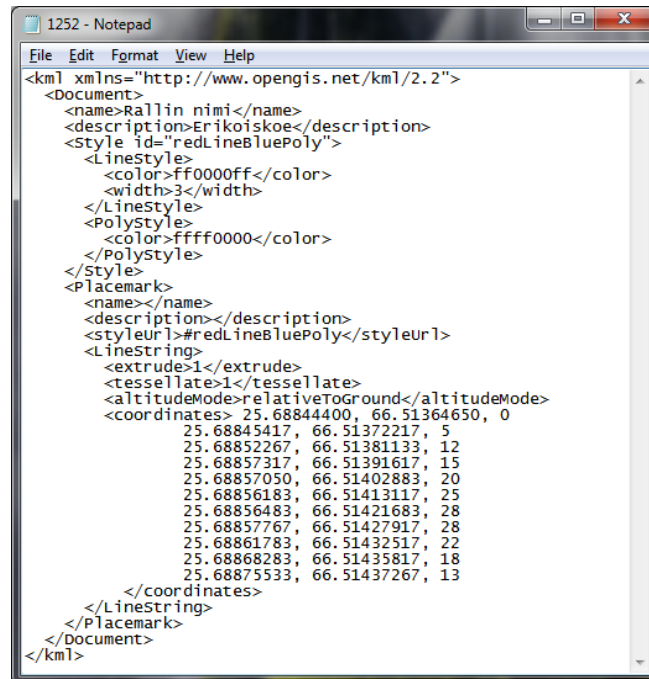
Laite luo kml-tiedoston GPS:ltä saadun kellonajan perusteella. Alla olevan kuvan tiedostonimi on 1252.kml, eli tallennus on aloitettu klo. 12.52 UTC. Kuvasta näkyy myös kml-tiedoston rakenne, joka koostuu standardin mukaisista tyyliasetuksista, sekä tallennetusta sijaintitiedosta.

Tiedoston alkuun määritellään merkkäuskieli ja tyyliasetusten lähde sekä reitille annetaan nimi <name> ja kuvaus <description>. Laite antaa reitille oletuksena nimen ”Rallin nimi” ja reitin kuvaukseksi ”Erikoiskoe”, jotka voidaan muokata jälkikäteen sopiviksi.

Tyyliasetuksissa, <style></style>, on määriteltä reitti näytettäväksi punaisena viivana ja sinisinä polygoneina, mutta <color>-kentistä voidaan värejä muuttaa

tyylin nimestä huolimatta.

Oletuksena valittu ”redLineBluePoly”-tyyli näyttää reitin <coordinates> </coordinates> kenttään kirjoitettujen koordinaattien sekä korkeustiedon perusteella, mutta tässä korkeustieto on korvattu nopeudella joka näytetään maan pintaan suhteutettuna, ”relativeToGround”, oletuksena olevan korkeuden meren pinnasta sijaan. <coordinates>-kenttä koostuu siis desimaalimuotoisista sijaintitiedoista longitudi, latitudi sekä nopeudesta kilometreinä tunneissa.



```

1252 - Notepad
File Edit Format View Help
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Rallin nimi</name>
    <description>Erikoiskoe</description>
    <style id="redLineBluePoly">
      <LineStyle>
        <color>ff0000ff</color>
        <width>3</width>
      </LineStyle>
      <Polystyle>
        <color>ffff0000</color>
      </Polystyle>
    </style>
    <Placemark>
      <name></name>
      <description></description>
      <styleurl>#redLineBluePoly</styleurl>
      <LineString>
        <extrude>1</extrude>
        <tessellate>1</tessellate>
        <altitudeMode>relativeToGround</altitudeMode>
        <coordinates> 25.68844400, 66.51364650, 0
          25.68845417, 66.51372217, 5
          25.68852267, 66.51381133, 12
          25.68857317, 66.51391617, 15
          25.68857050, 66.51402883, 20
          25.68856183, 66.51413117, 25
          25.68856483, 66.51421683, 28
          25.68857767, 66.51427917, 28
          25.68861783, 66.51432517, 22
          25.68868283, 66.51435817, 18
          25.68875533, 66.51437267, 13
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>

```

Kuvio 20. Laitteen tallentaman kml-tiedoston rakenne xml-kuvauksineen

4.3.2 FAT-tiedostojärjestelmä ja FatFS

File Allocation Table eli FAT on Bill Gatesin ja Marc McDonaldin (myöh. Microsoft) vuosina 1976–77 luoma tiedostojärjestelmä, joka on vielä nykyäänkin käytössä lähinnä siirrettävien medioiden flash-muisteissa kuten SD-muistikortissa ja muistitikuissa. FAT on poikkeuksetta tuettu kaiken tyyppisissä tietokoneissa yksinkertaisuutensa vuoksi ja se soveltuukin hyvin kohtalaisen pienten tiedostojen, kuten digitaalisten valokuvien ja musiikkitiedostojen varastoimiseen ja siirtämiseen. FAT-tiedostojärjestelmän heikkoudet tulevatkin esiin kun tiedostoja on valtava määrä, jolloin tiedostojärjestelmän kyky pakata dataa tiiviisti tallennusmediaan sekä datan kirjoitus- ja lukunopeus tippuu huomatta-

vasti. FAT-tiedostojärjestelmän suurin heikkous nykypäivän tekniikassa onkin luotettavuus, koska tiedostoja on paljon ja ne ovat huomattavasti aikaisempaa suurempia. FAT:n suurin tuettu tiedostokoko on $2^{32} - 1$ tavua, eli yhtä tavua vaille 4 Gb, joka ei tämän päivän tiedostoko'oissa ole mitenkään harvinaista verrattuna 10–15 vuotta sitten vallinneeseen tilanteeseen, jolloin FAT-tiedostojärjestelmän kehitys oli vielä kuumimmillaan, vuonna 2000 julkaistun Windows ME:n ollessa viimeinen puhtaasti FAT-tiedostojärjestelmän perustalle rakennettu käyttöjärjestelmä. (Microsoft MSDN 2010.)

FatFS on taas pieniin sulautettuihin järjestelmiin suunnattu geneerinen FAT-tiedostojärjestelmä. FatFS:n ominaisuuksiin kuuluu Windows-yhteensopivuus, sekä rautatason riippumattomuus, joten se voidaan liittää ilman muutoksia osaksi myös muiden kuin Atmelin AVR-tuoteperheen laitteita. FatFS-moduuli on vapaasti saatavilla ja muokattavissa opetuksellisiin, tutkimuksellisiin sekä eikaupallisiin tarkoituksiin (ELM-ChaN 2010). FatFS-moduulista käytettiin työssä tekohetkellä tuoreinta R0.07e-versiota. Vaikka FatFS-moduuli onkin periaatteessa suoraan yhteensopiva, täytyy kaikki levykirjoitusrutiinien ajastukset ja porttiasetukset ohjelmoida oman mikrokontrollerin kanssa yhteensopiviksi. Tässä vaiheessa työtä tulee käyttäjän ymmärtää käyttämänsä mikrokontrollerin ajastimien toimintaperiaate ja käyttöönotto. Käytettävän mikrokontrollerin datalehti on tärkein apuväline moduulin käyttöönotossa, josta selviää kaikkien tarpeellisten ajastimien osoitteet ja toimintaperiaatteet.

4.3.3 Ohjelmiston toiminta

Laitteen toiminta koostuu käyttäjälle annetuista laitteen tilaa kuvaavista merkinannoista ja pääohjelmasilmuksista, jonka etenemistä rajoittaa GPS-laitteella saatu Fix-tieto. Kun laite kytketään päälle ohjelma alkaa haistelemaan sarjaliikenneväylästä GPS-moduulin lähettämää RMC-lauseketta heti käynnistyttyään ja lukee lausekkeen toisesta kentästä saatavaa GPS-status -arvoa. Arvon ollessa muu kuin A (Automatic) ohjelma ei etene silmukassaan, koska Fix-tietoa ei GPS-moduulilla vielä ole. Tästä merkinä käyttäjälle etupaneelissa oleva ”No GPS Fix” -merkkivalo palaa. Jos yhteyden tila on A, ”No GPS Fix” -merkkivalo sammuu merkiksi, että järjestelmä vastaanottaa sijaintitietoja satelliiteilta ja oh-

jelma jää silmukassaan odottamaan käyttäjän toimenpiteitä tallennuksen aloittamiseksi eli Save-painikkeen painamista.

Save-painikkeen painamisen jälkeen ohjelma siirtyy tilaan, jossa `disk_initialize()`-funktio alustaa SPI-väylän ja pakottaa SD-kortin slave-tilaan. Jos muistikortin alustus ei onnistu johtuen viallisesta muistikortista, kirjoitussuojasta, väärästä tiedostojärjestelmästä tai ettei muistikorttia ole lainkaan, laite vilkuttaa "DISK ERROR" -merkkivaloa viestiksi levyvirheestä, jonka jälkeen ohjelma palaa silmukan alkuun tarkistamaan GPS:n tilan ja odottamaan seuraavaa Save-painikkeen painallusta. Tässä vaiheessa käyttäjän tulee korjata muistikortin alustuksen estävä virhe.

Kun levyn alustus tapahtuu onnistuneesti, ohjelma alkaa vastaanottaa sijaintitietoja GPS-moduulista saatavasta RMC-lauseesta sekä tallentaa sijaintitietoja muistikortille. "Saving"-merkkivalo syttyy muistikortin vieressä merkiksi tallennuksen olevan käynnissä.

Tiedon tallennus aloitetaan poimimalla RMC-lauseesta kellonaika, jonka perusteella ohjelma luo muistikortille tiedoston nimeltä "hhmm.kml", eli tiedostojen tarkastelua jälkikäteen helpottamiseksi tiedostojen nimet ovat kellonajan mukaan aina järjestyksessä.

Tiedoston luomisen jälkeen tiedoston sisälle kirjoitetaan kml-tiedostomuodon vaatimat xml-kuvaukset, jonka jälkeen ohjelma parseroi tarvittavat sijaintitiedot NMEA-lausekkeista, muuttaa koordinaatit desimaalimuotoon sekä maanopeuden solmuista kilometreiksi tunneissa, jolloin saadut arvot tallennetaan `<coordinates>` -tagin perään.

Tallennus jatkuu silmukassaan, kunnes käyttäjä painaa Stop-painiketta. Tällöin ohjelma lopettaa NMEA-lausekkeiden parseroinnin ja kirjoittaa loput xml-kuvaukset tiedostoon, synkronoi ja sulkee tiedoston. Ohjelma vilkuttaa Saving-LEDiä merkiksi tallennuksen päättymisestä, jonka jälkeen palataan silmukan alkuun ja uusi tallennus voi alkaa. Laitteen ohjelmiston toimintaa kuvaava lohko-kaavio liitteessä 1. "Ohjelmiston toimintaperiaate lohko-kaaviona" sekä lähdekoodi liitteessä 2. "Ohjelmiston lähdekoodi".

5 LAITTEEN TESTAAMINEN JA KEHITYSMAHDOLLISUUDET

5.1 Laitteen testaaminen

Laite testattiin laboratorio-olosuhteissa sekä kiinnitettynä ajoneuvoon. Laboratoriossa mitattiin mm. laitteen maksimi virrankulutus kaikkien toimintojen ollessa päällä yhtä aikaa, jolloin virrankulutukseksi saatiin noin 130 milliampeeria. Mittaustuloksen perusteella voidaan laskea, että jo tavallisella palovaroittimistakin löytyvällä 9 voltin (~500mAh) PP3 ”neppariparistolla” laitteen käyttöaika on noin neljä tuntia.

Laitteen toiminta testattiin ennen kilpa-ajoneuvoon asentamista siviiliajoneuvossa, jolla ajettiin reittejä tallentaen dataa sekä valmistetulla laitteella, että tallentamalla toisella GPS-moduulilla sama reitti raakana datana suoraan PC:lle. Testauksen perusteella ei huomattu eroavaisuutta kahden laitteen välillä mittaus-tarkkuudessa.

Laitteen toiminnassa itsessään huomattiin puutteita siinä tapauksessa, että GPS-moduulin yhteys satelliittiin katkeaa esimerkiksi katvealueen vuoksi, jolloin laite ei kesken tallennuksen virhetilasta ilmoita mitenkään, vaan jatkaa tallentamalla pelkkiä nolliä koordinaattien sijasta. Lisäksi moduuli saattaa lähettää fix-tietoa ”A” parikin minuuttia vielä sen jälkeen, kun antenni on irrotettu ja yhteys satelliitteihin estetty.

FatFS-moduulin huomattiin olevan valikoiva erilaisten muistikorttien suhteen, sillä jotkut jopa saman valmistajan eri muistikortit eivät tuntemattomasta syystä tunnistuneet. Vaikka huomattavan suuri osa kokeilluista muistikorteista toimikin oikein on laitteen käyttöönoton yhteydessä suotavaa varmistaa, että laite tunnistaa kortin ja tarvittaessa vaihtaa korttia.

Ongelmia huomattiin myös painikkeiden toiminnassa, johtuen todennäköisesti painikkeiden fyysisestä rakenteesta. Painikkeita tuleekin painaa rauhallisesti ”pohjaan saakka”, sillä nopea painallus ei välttämättä rekisteröidy laitteeseen.

5.2 Kehitysmahdollisuudet

5.2.1 Havaittujen vikojen ja puutteiden korjaaminen

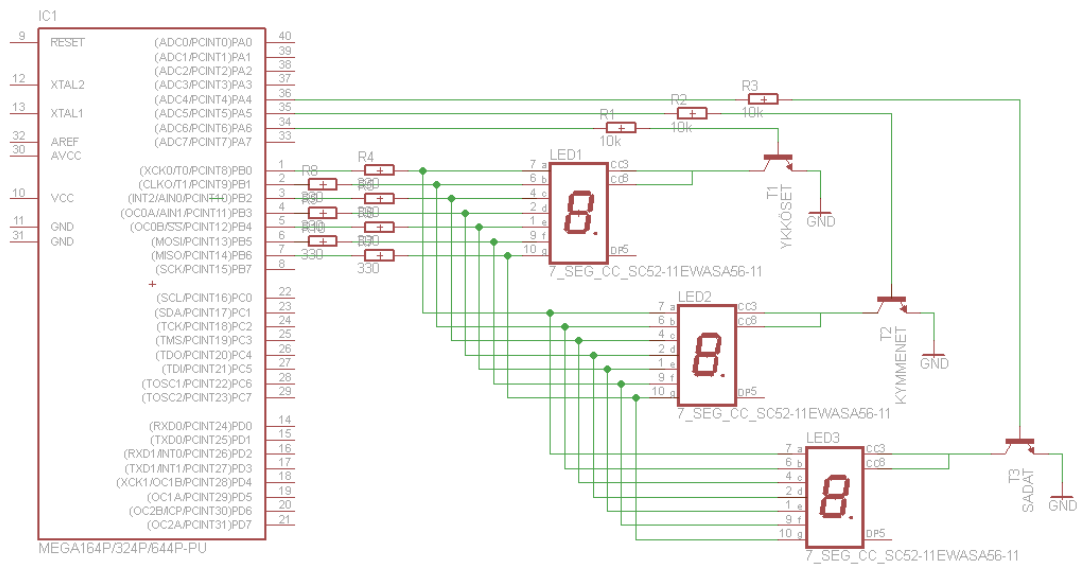
Testausvaiheessa esiin tullee pahimpaan ongelmaan, eli laitteen reagoimattomuuteen satelliittiyhteyden katketessa, voisi puuttua ohjelmallisesti lisäämällä tallennusvaiheeseen ohjelmasilmukan, jossa vertailtaisiin NMEA-lauseen koordinaattikenttiä. Kun laite kadottaa yhteyden satelliitteihin, ei laite reagoi katkokseen, vaan jatkaa datan tallentamista GPS-moduulista saatujen koordinaattikenttien ollessa NMEA-standardin mukaisesti tyhjiä. Tällöin tallennusvaiheessa voitaisiin tyhjän koordinaattikentän tapauksessa syyttää ”No GPS Fix” -merkkivalo käyttäjälle ja käyttäjän tehtäväksi jäisi joko lopettaa tallennus tai jatkaa tallennusta virhetilasta huolimatta. Katkos havaittaisiin nimenomaan NMEA-lausekkeen koordinaattikentästä, koska GPS-yhteyden tilaa esittävän kentän todettiin näyttävän yhteyden olevan kunnossa vielä minuutteja yhteyden katkeamisen jälkeenkin. Katkoksen tarkan sijaintitiedon tallennuksessa voitaisiin ehkäistä käyttämällä GPS-moduulin tukemaa gyroskooppiä tai ajoneuvon nopeusanturilta saatavaa pulssia. Tällöin myös paikannustarkkuutta saataisiin kasvatettua, eikä laite olisi riippuvainen jatkuvasta satelliittiyhteydestä.

Vaikka alkuperäisen käyttötarkoituksen mukaan laite on kaikin puolin alkupe-
räissuunnitelman mukainen ja sinällään tarkoitukseensa toimiva, voisi sen fyysiseen kokoon ja virrankulutukseen puuttua, jotta laitetta voitaisiin käyttää esimerkiksi paristokäyttöisenä. Laitteen fyysistä kokoa voitaisiin pienentää nykyisestään jo nyt käytetyillä komponenteilla karsimatta toimintoja. Huomattavasti enemmän kokoon pystyttäisiin vaikuttamaan muuttamalla mikrokontrollerin kotelotyyppiä TQFP44 malliseen, joka on kooltaan vain 10 mm x 10 mm nykyisen 40PDIP koteloidun ollessa 52 mm x 12 mm. Tilan tarvetta piirilevyllä voisi pienentää myös vaihtamalla muistikorttityyppiä micro-SD-malliin ja korvaamalla GPS-moduuli saatavilla olevalla fyysisesti pienemmällä mallilla. Näillä muutoksilla laite voitaisiin rakentaa muutamien senttimetrin kokoiseksi. Monipuolisten käyttötarkoitusten mahdollistamiseksi laitteeseen voisi lisätä myös paristokäytömahdollisuuden, jonka tukemiseksi myös komponenttien käyttöjännitettä ja virrankulutusta tulisi pienentää mahdollisimman pitkän käyttöajan saavuttami-

seksi. Mikrokontrollerin voisi asettaa toimimaan pienemmällä kellotaajuudella ja sen voisi korvata kokonaan vähävirtaisemmalla Atmega644A -mallilla, jolloin käyttöjännitteen voisi pudottaa 3,3 V:iin ja alle. Samalla GPS-moduulin vaatima aktiivinen antenni pitäisi korvata pienemmän käyttöjännitteen tarvitsevalla mallilla.

5.2.2 Lisää ominaisuuksia

Käyttötarkoituksessaan käytännöllinen lisä laitteeseen olisi nopeusnäyttö. Nopeusnäyttö tulitaisiin toteuttamaan ns. multipleksaamalla eli kolmella transistorilla ohjataan kolmea eri 7-segmenttinäyttöä GPS-moduulilta saadun nopeustiedon näyttämiseksi reaaliajassa. Multipleksaaminen mahdollistaa näyttöjen käyttämisen pienemmällä määrällä mikrokontrollerin pinnejä, jolloin kaikkien kolmen 7-segmenttinäytön pinnit voitaisiin kytkeä samaan mikrokontrollerin porttiin ja josta näyttöä ohjaisi yksi transistori, jolloin tarvittavia mikrokontrollerin pinnejä olisi vain kymmenen (7 näytöille + 3 transistoreille) suoraan kytkettyjen näyttöjen 21:n sijaan (7 pinniä/näyttö). Ideana multipleksaamisessa on syöttää useampi-numeroiselle näyttöryhmälle numero "vilkuttelemalla" erittäin nopeaan tahtiin kunkin numeron näyttämiseen tarvittavaa näyttöä, jolloin ihmissilmä ei erota vilkkumista, vaan näyttöryhmä näyttää toimivan sulavasti ilman katkoksia. Ohjelmallisesti mikrokontrollerin porttiin, johon näytön segmentit ovat kytketty, ajetaisiin jatkuvasti GPS-moduulista saatua 1–3 numeroista nopeustietoa (0–999km/h) numero kerrallaan, samalla kun 10–15 millisekuntiin ajastetut kolme transistoripinniä määräisivät mikä näytöistä näyttää porttiin ajatun numeron, ykköset–kymmenet–sadat-periaatteella. Samalla GPS-status ja muita tietoja voitaisiin kertoa käyttäjälle heksadesimaalikirjaimina näyttöryhmässä. Vaihtoehtoisesti laitteeseen voisi liittää LCD-tyyppisen näytön, jolla voitaisiin esittää myös sijaintitiedot koordinaatteina, suunnan ja muuta GPS-moduulista saatava informaatiota.



Kuvio 21. 7-segmentinäyttöryhmän kytkentä

Laitteeseen voisi liittää DB9 sarjaliikenneportin PC-kytkentää varten, joka vaatisi esim. MAX232-ohjaimen muokkaamaan GPS-moduulin TTL-signaalitasot PC-yhteensopiviksi. Tällöin GPS-laitetta pystyttäisiin käyttämään erillisenä ”GPS-mokkulana” PC:n karttasovelluksen kanssa esimerkiksi veneessä navigointilaitteena, laitteen kuitenkin toimiessa normaalisti alkuperäisen käyttötarkoituksen mukaan. Samalla MAX232-piiri mahdollistaisi laitteen konfiguroinnin valmistajan työkaluilla suoraan moduulia laitteesta irrottamatta.

Eräs mielenkiintoinen ominaisuus olisi lisätä laitteeseen GSM-moduuli, jolla laitteesta pystyttäisiin tekemään esimerkiksi seurantalaitte ajoneuvoon, jolloin GSM-moduuli lähettäisi laitteen sijaintitietoa esimerkiksi SMS-viestinä toiseen matkapuhelimeen tai GPRS-yhteydellä internetiin.

Ohjelmiston puolelle lisäominaisuuksia toisi keskinopeutta, kuljettua matkaa ja matka-aikaa mittaavat funktiot, jotka toisivat tarpeellista lisäinformaatiota kuljetuun reittiin. Lisäksi tiedostojärjestelmän voisi korvata huomattavasti yksinkertaisemmalla, vähemmän muistia vievällä ja varmatoimisemmalla versiolla.

6 YHTEENVETO

Laitteen ollessa ideavaiheessa oli jo selvillä vaadittavat ominaisuudet, jotka haettiin pitää toimintoja karsimatta tai lisäämättä toteutuneen laitteen mukaisina. Alkuun monimutkaiselta tuntuneen ohjelmiston suunnittelu- ja toteutusvaiheen päästyä vauhtiin oli jopa vaikeuksia pitää paketti alkuperäisessä suunnitelmasaan. Nälkä alkoi kasvaa syödessä. AVR-tuoteperheen monipuoliset käyttösovellukset sekä kymmenet projektit, joihin työn aikana törmättiin loivat valtavasti mielenkiintoisia kehitysideoita, jopa lopulliseksi kaupalliseksi tuotteeksi asti. Vaikka AVR-mikrokontrollereiden käyttäminen oli entuudestaan uutta asiaa, muutamaa pientä harjoitussovellusta lukuun ottamatta, oli sovelluksen kehittäminen varsin helppoa valmistajan tarjoamilla työkaluilla ja dokumentoinnilla sekä alan verkkoyhteisöjen keskusteluja seuraamalla ja erilaisia pieniä sovellusesimerkkejä kokeilemalla.

Laitteen fyysinen toteutus oli osa-alueista ylivoimaisesti helpoin ja toteutukseen vähiten aikaa vievä. Prototyypiversiosta lopulliseen versioon matkaa olikin ainoastaan kaksi eri versiota. Suurimpana kompastuskivenä oli tällä kertaa piirilevyjen valmistusvaiheen venyminen laitteistovioista johtuen, mikä viivästytti kehitysprosessia huomattavasti. Yhtenä erittäin tärkeänä asiana esille tulikin saatavilla olevien kehitysalustojen tärkeys tällaisessa projektissa, jossa suunnittelu alkaa puhtaalta pöydältä. Koska sopivaa kehitysalustaa ei ollut saatavilla, jouduttiin laitteen ensimmäinen versio valmistamaan jo sitä varten, että ensimmäistä lähdekoodiversiota päästiin edes testaamaan. Monikäyttöinen kehitysalusta onkin ehdoton edellytys tuotteen jatkojalostukselle, jotta aikaa vievältä uuden fyysisen version valmistamiselta joka kerta ominaisuuksien lisääntyessä vältyttäisiin.

Ohjelmiston kehityksessä suurimmat ongelmat olivat FatFS-moduulin integroinnissa käytettyyn järjestelmään. Tässä vaiheessa olisi toivonutkin, että moduulin tuhatrivisessä lähdekoodissa olisi ollut tarkemmat selostukset eri osioiden vaikutuksesta toimintaan ja että mitä kaikkea tiedostojärjestelmän käyttöönotossa ja räätälöinnissä omaan kokoonpanoon tulisi ottaa huomioon. Loppujen lopuksi FatFS-moduuli saatiin toimintakuntoiseksi pitkälti ”kokeile ja erehdy” -

menetelmällä ja mikrokontrollerin datalehteä opiskelemalla, joiden myötä käytettyjen rutiinien, ajastimien, signaalien ja näiden ohjelmointimenetelmien toimintakin avautui loppujen lopuksi. SPI- sekä USART-väylien toimintakuntoon saattamisessa tuotetuki löytyi niin ikään valmistajan datalehdistä sekä kml-formaatin toimintaperiaate Googlen hyvin dokumentoiduista esimerkeistä, eikä siis näiltä osin pahoihin ongelmiin ajauduttu.

Lopputuloksena työssä saatiin valmistettua paitsi toimiva ja alkuperäiset vaatimukset täyttävä GPS-laite, myös paljon tietoa jatkuvasti eri sovelluksissa lisääntyvän GPS-järjestelmän toiminnasta sekä mikrokontrollereiden maailmasta pintaa syvemmmältä.



Kuvio 22. Työssä valmistettu kilpa-ajoneuvon GPS-loggeri

LÄHTEET

- AMSA 2007. Australian merivartiosto. DGPS Fact Sheet. Osoitteessa
http://www.amsa.gov.au/Publications/Fact_sheets/DGPS_Fact_Sheet.pdf 23.3.2010
- Atmel 2010. ATmega644P/V datasheet. Osoitteessa
http://www.atmel.com/dyn/resources/prod_documents/doc8011.pdf
19.3.2010
- DLR 2007. German Aerospace Center. Developments for Safety of Life Applications. Osoitteessa
<http://www.dlr.de/kn/portaldata/27/resources/dokumente/kn-flyer-2007-sol-e.pdf> 18.3.2010
- ELM-ChaN 2010. FAT File System Module. Osoitteessa http://elm-chan.org/fsw/ff/00index_e.html 20.3.2010
- ESA 2009. European Space Agency. EGNOS-järjestelmä. Osoitteessa
http://www.egnos-pro.esa.int/Publications/ESA_EGNOS_br284_2009.pdf 20.3.2010
- ESA 2010. European Space Agency. EGNOS fact sheet. Osoitteessa
http://www.mi.gov.pl/1/files/0/3141/fact_sheet_14.pdf 20.3.2010
- FAA 2010. Federal Aviation Administration. WAAS-järjestelmä. Osoitteessa
http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/waas/ 20.3.2010
- Google kml-tutorial 2010. Osoitteessa: http://code.google.com/intl/fi-FI/apis/kml/documentation/kml_tut.html 20.3.2010
- Indagon Oy 2010. FOKUS-palvelu. Osoitteessa
<http://www.indagon.com/fi/tuotteet-ja-palvelut/21-fokus-palvelu>
23.3.2010

- John Hopkins University APL 2008. Genesis of Satellite Navigation System
Osoitteessa <http://techdigest.jhuapl.edu/td1901/guier.pdf> 11.3.2010
- John Hopkins University APL 2010. Yleiskatsaus TRANSIT (NAVSTAR)-
järjestelmästä. Osoitteessa <http://sd-www.jhuapl.edu/Transit/>
11.3.2010
- John Hopkins University APL1998a. An overview of Transit Development. Osoitteessa <http://www.jhuapl.edu/techdigest/td/td1901/danchik.pdf>
11.3.2010
- John Hopkins University APL1998b. The Legacy of Transit: Guest Editor's Introduction. Osoitteessa
<http://www.jhuapl.edu/techdigest/td/td1901/pisacane.pdf> 19.3.2010
- Merenkululaitos 2010. DGNSS-palvelu. Osoitteessa
http://portal.fma.fi/sivu/www/fma_fi/merenkulun_palvelut/liikenteen_ohjaus/radionavigointi 23.3.2010
- Microchip Technology Inc 2010. Overview and Use of the PICmicro Serial Peripheral Interface. Osoitteessa
<http://ww1.microchip.com/downloads/en/DeviceDoc/spi.pdf>
26.3.2010
- NASA 2010a. Transit-satelliitit. Osoitteessa
<http://nssdc.gsfc.nasa.gov/nmc/masterCatalog.do?sc=TRAN1>
11.3.2010
- NASA 2010b. Timation-satelliitit. Osoitteessa
<http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1967-053E>
11.3.2010
- NAVCEN 1995. Yhdysvaltain rannikkovartioston navigaatiokeskus. Global Positioning System standard positioning service signal specification.
Osoitteessa
<http://www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf>

15.3.2010

NAVCEN 1996. Navstar GPS User Equipment Introduction. Osoitteessa
<http://www.navcen.uscg.gov/pubs/gps/gpsuser/gpsuser.pdf>

14.3.2010

NAVCEN 2010. Yhdysvaltain rannikkovartioston navigaatiokeskus. GPS-
 aikataulut. Osoitteessa

<http://www.navcen.uscg.gov/gps/almanacs.htm> 14.3.2010.

Navy NRL 1986. Yhdysvaltain laivasto. GPS Clock technology and the NAW
 PTTI programs at the U.S. Naval Research Laboratory. Osoitteessa

http://tycho.usno.navy.mil/ptti/1986/Vol%2018_03.pdf 11.3.2010.

Navy NRL 2010. Yhdysvaltain laivasto. Timation Project. Osoitteessa

<http://code8200.nrl.navy.mil/nts.html> 11.3.2010

PNT 2010. The National Executive Committee for Space-Based Positioning,
 Navigation, and Timing. SA-häirintä. Osoitteessa

<http://pnt.gov/public/sa/> 22.3.2010

Richard Easton 2006. Who invented the Global Positioning System? Osoittees-
 sa

<http://www.thespacereview.com/article/626/1> 23.3.2010

SD Card Association 2010. SD-muistikorttien teknisiä ominaisuuksia. Osoittees-
 sa :

<http://www.sdcard.org/developers/tech/> 18.3.2010

Spacecraft encyclopedia 2004a. Transit-satelliitit. Osoitteessa

<http://claudelafleur.qc.ca/Spacecrafts-1959.html#Transit-1A>

11.3.2010

Spacecraft encyclopedia 2004b. Navstar-satelliitit. Osoitteessa

<http://claudelafleur.qc.ca/Navstar.html> 14.3.2010

Trimble 2010. Lassen DR+GPS datasheet. Osoitteessa

http://www.dpie.com/datasheets/gps/022542-005A_Lassen-DR_DS_0907_A4_Ir.pdf 19.3.2010

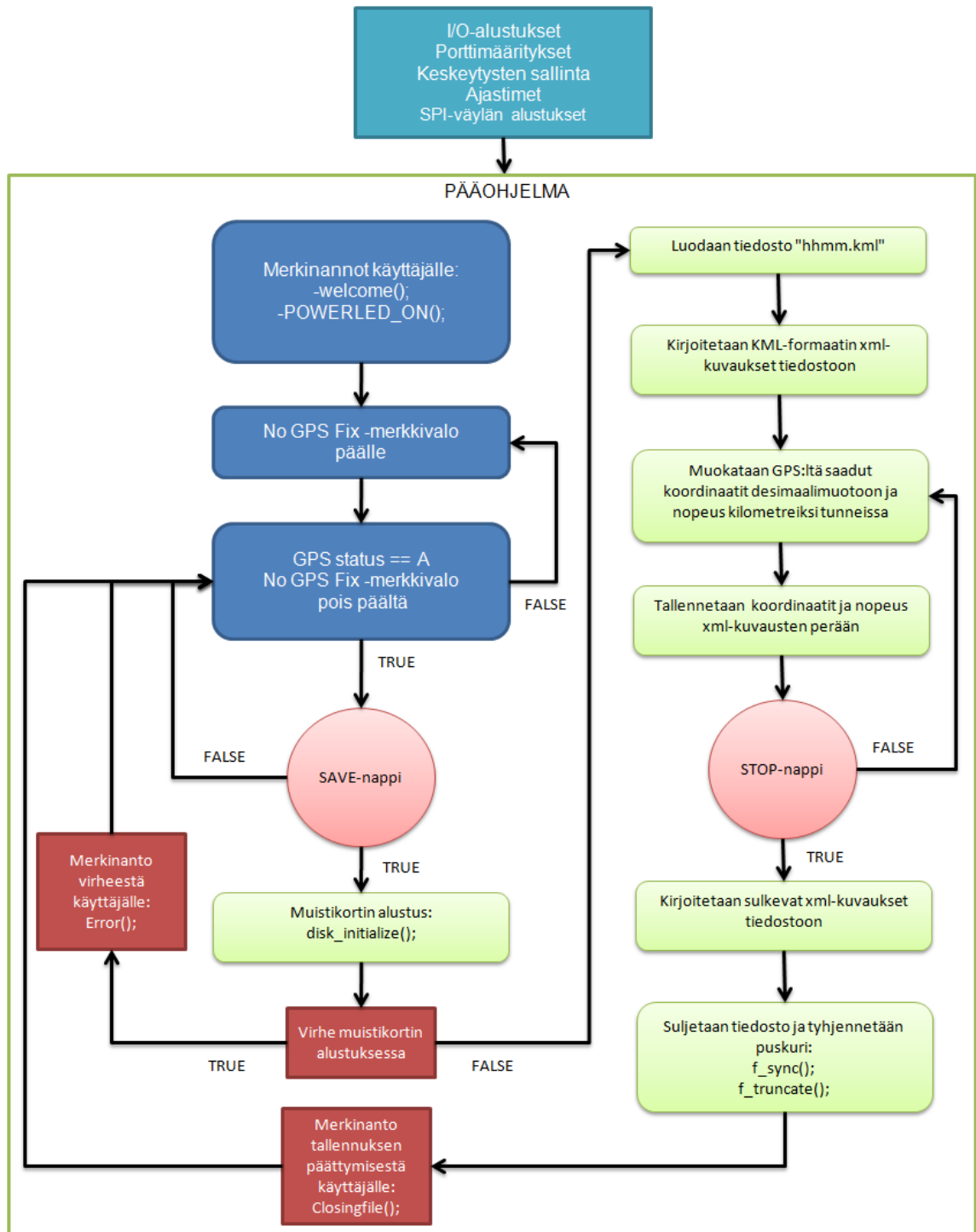
Valkoisen talon tiedote 2000. Presidentti Clintonin puhe SA-häirinnän poistamisesta. http://clinton3.nara.gov/WH/EOP/OSTP/html/0053_2.html
22.3.2010

LIITTEET

Ohjelmiston toimintaperiaate lohkokaaavana	Liite 1
Ohjelmiston lähdekoodi	Liite 2
GPS-loggerin kytkentäkaavio	Liite 3

Ohjelmiston toimintaperiaate lohkokaaviona

Liite1



Ohjelmiston lähdekoodi

Liite 2 (1/7)

```

/*****
Project : gps_logger.c
Hardware: ATmega 644P (8Mhz)
Software: AVRStudio 4.17.0.655 + WinAVR-20090313
Date: 22.3.2010
Author: Tuomas Virjonen
Comments: Picks NMEA-sentences from USART0, parses coordinates and ground speed from
RMC-sentence, writes data on SD-memory card to kml-file format.
*****/

/*Kirjastot*/

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <util/delay.h>
#include "uart.h"
#include "xitoa.h"          //
#include "ff.h"            //      FAT-tiedostojärjestelmä
#include "diskio.h"        //
#include "rtc.h"           //

FATFS Fatfs;              // Tiedostojärjestelmän määrittely
FIL pointFile;
FIL pathFile;
BYTE str[100];           // Työpuskuri. NMEA-lauseiden pituus noin 80-90 merkkiä
BYTE *p = NULL;

/*Painikkeet ja merkkivalot*/

#define    SAVELED_ON()      PORTC |= _BV(4)
#define    SAVELED_OFF()    PORTC &= ~_BV(4)
#define    POWERLED_ON()    PORTC |= _BV(5)
#define    POWERLED_OFF()   PORTC &= ~_BV(5)
#define    ERRORLED_ON()    PORTC |= _BV(6)
#define    ERRORLED_OFF()   PORTC &= ~_BV(6)
#define    FIXLED_ON()      PORTC |= _BV(7)
#define    FIXLED_OFF()     PORTC &= ~_BV(7)
#define    BUTTON1          bit_is_clear(PINC, 2)      //Save-painike
#define    BUTTON2          bit_is_clear(PINC, 3)      //Stop-painike

/*Muuttujat*/

unsigned int speed_kmh; double lat_dec, lon_dec, minutes, degrees;
char lat_tmp[20], lon_tmp[20];
char Fix, latitude[15], longitude[15], east_or_west[1], north_or_south[1], speed[5];

```

Ohjelmiston lähdekoodi

Liite 2 (2/7)

```
/*I/O-asetukset*/
```

```
static
void ioinit()
{
    PORTA = 0xFF;           //Ei käytetty
    PORTB = 0xB0;         // SPI-väylä
    DDRB = 0xC0;
    PORTC = 0x0F;         //Napit PC2&PC3, Ledit PC4-PC7, Slave Select PC1
    DDRC = 0xFF;
    PORTD = 0xFE;         //USART-väylä, PD0 RXD0 -käytössä (vastaanotto)
    DDRD = 0x83;
    OCR2A = 90-1;         // 100Hz ajastin (OC2). Kutsuu muistikorttia 10ms välein
    TCCR2A = 0b00000010;
    TCCR2B = 0b00000101;
    TIMSK2 = 0b00000010; // TC2.oc keskeytyksen sallinta
    uart_init();          // USART alustus
    sei();                 // yleinen keskeytysten sallinta
    ISR (USART0_RX_vect); // USART0 vastaanoton overflow keskeytys
}

```

```
/* Kelvollisen NMEA-lauseen haku USART:lta */
```

```
static
BYTE get_line(void)
{
    BYTE c, i = 0;
    for (;;) {
        c = uart_get(); // NMEA lauseen ensimmäinen merkki on '$'
                        // luetaan rivi työpuskuriin.
        if (!c || (i == 0 && c != '$')) continue; // Jos rivi loppuu
        str[i++] = c; // tai puskuri tulee täyteen,
                    // tyhjennetään puskuri
        if (c == '\n') break; // ja jääetään odottamaan uutta
        if (i >= sizeof(str)) i = 0; // riviä.
    }
    return i;
}

```

```
/* Tietyn NMEA-lauseen haku USART:lta */
```

```
static
BYTE compare_str (const char *str1, const prog_char *str2)
{
    BYTE c;
    do {
        c = pgm_read_byte(str2++); // Vertailee määriteltyä merkkijonoa
                                // ja GPS:ltä saatua merkkijonoa
    } // esim. "$GPRMC".
}

```

Ohjelmiston lähdekoodi

Liite 2 (3/7)

```

    while (c && c == *str1++);           // Jos merkit täsmäävät
    return c;                           // palautetaan rivin arvo
}

/* Sarakkeiden erottelu NMEA-lauseesta */

static
BYTE* get_column (const char* buf, BYTE col)
{
    BYTE c;
    while (col) {
        do {
            c = *buf++;                 // Luetaan merkkejä puskuriin.
            if (c <= ' ') return NULL;  // Jos tyhjä sarake, palautetaan
        } while (c != ',');            // nolla. Jos pilkku, sarakkeen
        col--;                          // järjestysluku kasvaa.
    }
    return (BYTE*)buf;
}

/*Merkinannot käyttäjälle*/

typedef unsigned int u_int;

int welcome(u_int n)                   // Päälle kytkettäessä kaikkia ledejä
{                                       //vilkutetaan
    ERRORLED_ON();
    FIXLED_ON();
    POWERLED_ON();
    SAVELED_ON();
    _delay_ms(150);
    ERRORLED_OFF();
    FIXLED_OFF();
    POWERLED_OFF();
    SAVELED_OFF();
    _delay_ms(150);
    n--;}
}

int Error(u_int n) {                   // Virhe muistikortin alustuksessa tai kortille
    while(n > 0) {                     //kirjoittamisessa
        ERRORLED_ON();
        _delay_ms(50);
        ERRORLED_OFF();
        _delay_ms(30);
        n--;}
}

```

Ohjelmiston lähdekoodi

Liite 2 (4/7)

```

int Closingfile(u_int n) {
    while(n > 0) {
        SAVELED_ON();
        _delay_ms(150);
        SAVELED_OFF();
        _delay_ms(100);
        n--;
    }
}

/*PÄÄOHJELMA*/

int main(void)
{
    ioinit(); // I/O-alustukset
    welcome(10);
    _delay_ms(200);
    POWERLED_ON(); // Power -ledi päälle

    while(1); {
        do{
            get_line(); //Haetaan rivi USART:lta
        }

        while(compare_str(str, PSTR("$GPRMC"))); { // haetaan RMC-lause
            p = get_column (str,2); //poimitaan Fix-tieto kentästä 3
            memcpy(&Fix, p, 1); //A = automatic, V= system warning

            if (Fix != 'A') { // jos erisuuri kuin A
                FIXLED_ON(); // No GPS Fix -merkkivalo päälle
            }

            if(Fix == A) { //Fix == A
                FIXLED_OFF(); //No GPS Fix -merkkivalo pois päältä
                if(BUTTON1){ // Fix == A ja painetaan SAVE-nappia
                    //aloitetaan tallennus

                disk_initialize(0); //Muistikortin alustus
                f_mount(0, &Fatfs); // Tiedostojärjestelmä -> Fat
            }
        }
    }
}

```

Ohjelmiston lähdekoodi

Liite 2 (5/7)

```

do {
    get_line();
}

while(compare_str (str, PSTR("$GPRMC"))); // jos RMC-lause löytyy
p = get_column(str,1); // poimitaan kellonaika "hhmmss.ss"
memcpy(&str[0], p, 4); // huomioidaan ainoastaan tunnit ja minuutit
strcpy_P(&str[4], PSTR(".kml")); // luodaan tiedosto "hhmm.kml"

if (f_open(&pathFile, str, FA_OPEN_ALWAYS | FA_WRITE) || f_lseek(&pathFile, pathFile.fsize)){
    Error(7); // jos tiedoston luomisessa tapahtuu virhe
    SAVELED_OFF(); break; // annetaan merkki virheestä käyttäjälle
} // ja palataan pääohjelmaan...

SAVELED_ON(); //...muulloin tallennuksen merkkivalo päälle
// ja kirjoitetaan xml-kuvaukset luotuun kml-tiedostoon

f_printf(&pathFile, "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
f_printf(&pathFile, "<kml>\n");
f_printf(&pathFile, "\t<Document>\n");
f_printf(&pathFile, "\t<name> Rally </name>\n");
f_printf(&pathFile, "\t<description> </description>\n");
f_printf(&pathFile, "\t<Style id=\"redLineBluePoly\">\n");
f_printf(&pathFile, "\t<LineStyle>\n");
f_printf(&pathFile, "\t\t<color>ffff0000</color>\n");
f_printf(&pathFile, "\t\t<width>3</width>\n");
f_printf(&pathFile, "\t</LineStyle>\n");
f_printf(&pathFile, "\t<PolyStyle>\n");
f_printf(&pathFile, "\t\t<color>ffff0000</color>\n");
f_printf(&pathFile, "\t</PolyStyle>\n");
f_printf(&pathFile, "\t</Style>\n");
f_printf(&pathFile, "\t<Placemark>\n");
f_printf(&pathFile, "\t<name></name>\n");
f_printf(&pathFile, "\t<description></description>\n");
f_printf(&pathFile, "\t<styleUrl>#redLineBluePoly</styleUrl>\n");
f_printf(&pathFile, "\t\t<LineString>\n");
f_printf(&pathFile, "\t\t\t<extrude>1</extrude>\n");
f_printf(&pathFile, "\t\t\t<tessellate>1</tessellate>\n");
f_printf(&pathFile, "\t\t\t<altitudeMode>relativeToGround</altitudeMode>\n");
f_printf(&pathFile, "\t\t\t\t<coordinates>\n");

```

Ohjelmiston lähdekoodi

Liite 2 (6/7)

```

while(get_line()) {           //jatketaan tietojen poimimista RMC-lauseesta
if(!compare_str (str, PSTR("$GPRMC"))) {
    char* endptr;

    p = get_column (str, 3);           // kenttä 4, latitudi
    memcpy(&lat_temp, p, 10);
    lat_dec = strtod(lat_temp, &endptr);           // muutetaan string → float
    minutes = modf(lat_dec/100, &degrees);           // irrotetaan desimaalit
    lat_dec = degrees + (minutes*100)/60;           // lasketaan takaisin koordinaatiksi

    p = get_column (str,4);           // kenttä 5, leveyspiiri
    memcpy = (&north_or_south, p, 1);
    if (north_or_south == 'S')           // jos eteläinen:
    lat_dec = -lat_dec;           // latitudi on -latitudi

    p = get_column (str, 5);           // kenttä 6: longitudi
    memcpy(&lon_temp, p, 11);
    lon_dec = strtod(lon_temp, &endptr);           // muutetaan string → float
    minutes = modf(lon_dec/100, &degrees);           // irrotetaan desimaalit
    lon_dec = degrees + (minutes*100)/60;           // lasketaan takaisin koordinaatiksi

    p = get_column (str, 6);           //kenttä 7: pituuspiiri
    memcpy(&east_or_west, p, 1);
    if(east_or_west == 'W')           //jos läntinen:
    lon_dec = -lon_dec;           // longitudi on -longitudi

    p = get_column (str,7);           //kenttä 8: maanopeus
    memcpy(&speed, p, 5);           //muutetaan maanopeus solmuista → km/h
    speed_kmh = strtod(speed, &endptr)*1.851+0.5;
    //pyöristetään ylöspäin
    // ja jätetään desimaalit huomioimatta

    sprintf(latitude, "%f", lat_dec);           // muutetaan saadut arvot
    sprintf(longitude, "%f", lon_dec);           // takaisin merkkijonoiksi
    sprintf(speed, "%i", speed_kmh);
    // tulostetaan desimaalimuotoisiksi koodrinaateiksi ja kilometreiksi tunnissa
    // muutetut merkkijonot tiedostoon
    f_printf(&File1, "\t\t\t%s, %s, %s \n", longitude, latitude, speed);

    f_sync(&File1);           // tiedoston synkronointi
    f_truncate(&File1);           // kirjoituspuskurin tyhjentäminen

```


GPS-loggerin kytkentäkaavio

Liite 3

