

TAMPEREEN AMMATTIKORKEAKOULU  
Tietotekniikan koulutusohjelma  
Tietokonetekniikan suuntautumisvaihtoehto

Opinnäytetyö

Jussi Saarijoki

**GPS-LOGGERI**

Työn valvoja      Yliopettaja Kai Poutanen  
Tampere 2009

<b>Tekijä:</b>	Jussi Saarijoki
<b>Työn nimi:</b>	GPS-loggeri
<b>Päivämäärä:</b>	6.4.2009
<b>Sivumäärä:</b>	31 sivua ja 8 liitesivua
<b>Hakusanat:</b>	GPS, logger, AVR
<b>Koulutusohjelma:</b>	Tietotekniikka
<b>Suuntautumisvaihtoehto:</b>	Sulautetut järjestelmät
<b>Työn valvoja:</b>	Yliopettaja Kai Poutanen
<p>Työn tavoitteena oli suunnitella ja valmistaa laite, joka vastaanottaa GPS-järjestelmän tarjoamaa paikkadataa, käsittelee sen haluttuun muotoon ja tallentaa massamuistissa sijaitsevaan FAT-yhteensopivaan tiedostoon.</p> <p>Tarkoituksena oli toteuttaa harrastuskäyttöön soveltuva avoimen järjestelmän GPS-loggeri ja toisaalta myös pohtia vastaavaan kaupalliseen versioon tulevia ominaisuuksia nyt valmistettavan laitteen pohjalta.</p> <p>Laitteisto koostuu Atmelin 8-bittisestä AVR-mikrokontrollerista, GPS-laitteesta, massamuistista ja oheiskomponenteista, joita ovat virtalähde tai akku, näyttö ja näppäimistö. Laitteen ohjelmisto on kirjoitettu C-kielellä ja käännetty AVR-GCC-kääntäjällä. Ohjelmistoa kirjoitettaessa käytettiin standardikirjastojen sisältämien funktioiden lisäksi hyödyksi lainattua FAT-yhteensopivaa tiedostojärjestelmää ja massamuistin I/O-rutiineja.</p> <p>Tehty prototyyppi antoi osviittaa järjestelmän vähimmäisvaatimuksista, mutta myös helpotti huomattavasti lopullisen tuotteen vaatimusten sekä niiden toteutuksen hahmottamista.</p> <p>Työn tuloksena tuotetun prototyypin pohjalta voidaan helposti rakentaa useita eri sovelluksia paikannustekniikkaa ja koordinaattidataa hyödyntämällä.</p>	

<b>Author:</b>	Jussi Saarijoki
<b>Title:</b>	GPS – loggeri
<b>Date:</b>	6.4.2009
<b>Number of pages:</b>	31 pages, 8 appendix pages
<b>Keywords:</b>	GPS, logger, AVR
<b>Program:</b>	Information Technology
<b>Specialisation:</b>	Embedded systems
<b>Thesis supervisor:</b>	Senior Lecturer Kai Poutanen
<p>The main object was to design and compose a device, that would collect, modify and store or otherwise utilize the positioning data provided by the GPS-system.</p> <p>The objective was to implement a non-commercial, hobby-intended system and to ponder the possible properties and difficulties upon the design and research with similar, commercially available products.</p> <p>The hardware of the device itself consists of Atmel's 8-bit AVR microcontroller, GPS-device, mass storage and peripheral devices like a display, power supply and a keyboard. The software of the device is written in C-language and compiled with the AVR-GCC-compiler. The written software implements a borrowed FAT-compatible file system and disk routines and also functions included in the standard libraries and the libraries that came with the compiler.</p> <p>Prototyping the device gave a good hint about the minimum requirements to achieve such a system but also eased the resource costs of a similar, commercial system.</p> <p>The prototype conceived seems an excellent base system for a multitude of applications by simply using the information and positioning data available.</p>	

## **ALKUSANAT**

Olen aina halunnut hyödyntää taivaalta saatavaa informaatiota ja opetella uusien mikro-kontrollerien käyttöä.

Haluan kiittää annetusta tuesta avovaimoani ja TAMKIn opettajista erityisesti Ari Rantalaa, Mauri Inhaa ja Kai Poutasta.

Tampereella 25. maaliskuuta 2009

---

Jussi Saarijoki

## SISÄLLYSLUETTELO

TIIVISTELMÄ .....	ii
ABSTRACT .....	iii
ALKUSANAT .....	iv
SISÄLLYSLUETTELO.....	v
KÄYTETYT TERMIT JA LYHENTEET .....	vii
1 JOHDANTO .....	1
2 JÄRJESTELMÄN MÄÄRITTELY .....	2
2.1 GPS-järjestelmä ja sen hyödyntäminen työssä .....	2
2.2 FAT ja sen tuomat edut .....	3
2.3 Prototyypilaitteiston ja sen toiminnan määrittely .....	3
3 LAITTEISTON SUUNNITTELU JA TOTEUTUS .....	4
3.1 ATmega128-mikrokontrolleri.....	4
3.2 MT-128-kehitysalusta .....	4
3.3 AVR-USB-JTAG .....	5
3.4 AVR Studio 4, WINAVR .....	6
3.5 SD-muistikortti.....	7
3.6 GPS-moduuli.....	8
3.7 Prototyypin kytkentä .....	9
4 OHJELMISTON SUUNNITTELU JA TOTEUTUS .....	10
4.1 Käytetyt kirjastot ja funktiot .....	11
4.2 GPS-data näytölle .....	12
4.3 Paikkadatan tallennus SD-kortille.....	12
4.4 Paikkadatan tallennus KML-muodossa.....	13
5 LAITTEEN TESTAUS JA KEHITYSMAHDOLLISUUDET .....	14
5.1 Ohjelmiston testaus .....	15
5.1.1 Käyttöliittymätestaus .....	15
5.1.2 Toiminnallisuustestaus.....	16
5.2 Laitteen testaus.....	17
5.2.1 Virrankulutus ja akkutoiminta .....	17
5.4 GPS-lokin tarkkuus .....	18
5.5 Lopullinen tuote .....	19
5.6 Kehitys- ja laajennusmahdollisuuksista .....	21

6 YHTEENVETO .....	21
LÄHTEET .....	23
LIITTEET .....	24

## **KÄYTETYT TERMIT JA LYHENTEET**

AVR	Yleisnimitys Atmelin 8-bittiselle mikrokontrolleriperheelle
GPS	Global Positioning System. Yhdysvaltain puolustusministeriön kehittämä ja rahoittama satelliittipaikannusjärjestelmä. Nykyään vapaassa siviilikäytössä.
JTAG	Joint Test Action Group. Piirivalmistajien yhteistyönä kehitetty testausväylä.
SPI	Serial Peripheral Interface Bus.
KML	Keyhole Markup Language
NMEA	National Marine Electronics Association. Lyhennettä käytetään myös NMEA 0183 -standardista.
USART	Universal Synchronous/Asynchronous Receiver Transmitter. Tässä dokumentissa tällä tarkoitetaan yleisesti mikrokontrollerin sarjaporttia.
DGPS	Differentiaalinen GPS. Tunnetun kiintopisteen avulla korjattu GPS-järjestelmä.

## 1 JOHDANTO

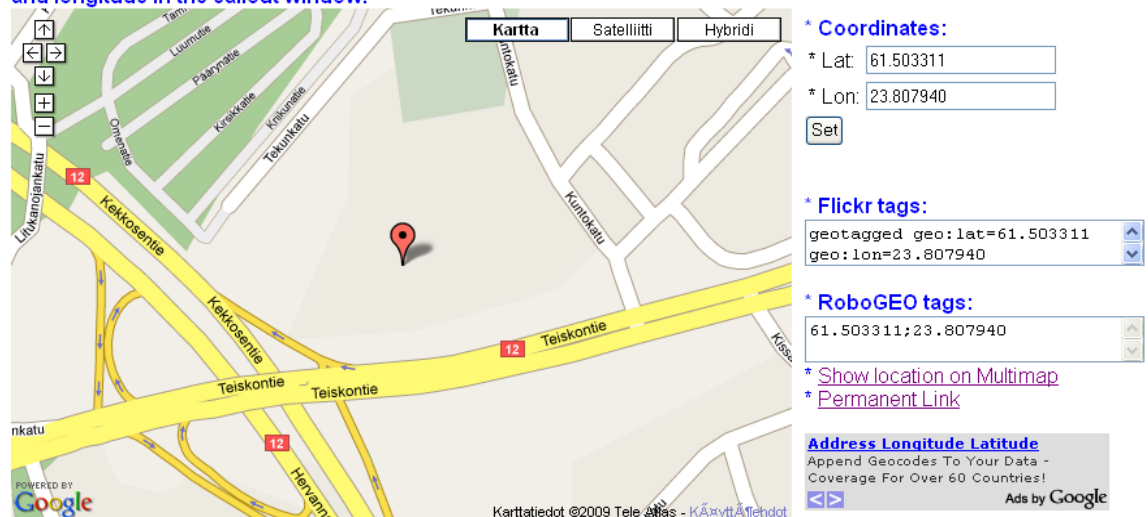
GPS-järjestelmän SA-häirinnästä luopumisen (2000-luvun alussa) seurauksena on erilaisten paikannusta hyödyntävien sovellusten yleistymisen ollut räjähdysmäistä. Signaalia hyödyntävien piirien ja näiden ympärille rakennettujen moduulien halpeneminen, fyysisten mittojen kutistuminen sekä virrankulutuksen pieneneminen on edesauttanut järjestelmän tuomista jokapäiväiseen käyttöön.

Hyvin lyhyessä ajassa suuri osa arkipäiväisiä toimintoja on muodostunut riippuvaiseksi satelliittipaikannusjärjestelmän tarkkuudesta ja saatavuudesta mutta samalla myös järjestelmän huvi- ja vapaa-ajankäyttö on lisääntynyt räjähdysmäisesti.

Tässä työssä on määritelty, suunniteltu ja toteutettu yksinkertainen GPS-paikannusdatan tallennusjärjestelmä, jonka pohjalta on myös mietitty vastaavan kaupallisen järjestelmän toteuttamisen sudenkuoppia.

**Google Maps Latitude, Longitude Popup**

Simply click on the map on a location and it will provide you with the latitude and longitude in the callout window.



The screenshot shows a Google Maps interface with a red location pin on a street. The popup window displays the following information:

- \* Coordinates:**
  - \* Lat: 61.503311
  - \* Lon: 23.807940
  - Set
- \* Flickr tags:**
  - geotagged geo:lat=61.503311
  - geo:lon=23.807940
- \* RoboGEO tags:**
  - 61.503311;23.807940
- [\\* Show location on Multimap](#)
- [\\* Permanent Link](#)
- Address Longitude Latitude**
  - Append Geocodes To Your Data - Coverage For Over 60 Countries!
  - Ads by Google

Address lookup has been removed because it violated the ToS of Google Maps.

Based on code taken from [this website](#) and [this website](#).  
All other errors are caused by code written by [Pierre Gorissen](#).

Kuva 1. Luentotilan A3-14 sijainti Google Maps -ohjelmassa esitettynä.



## 2 JÄRJESTELMÄN MÄÄRITTELY

Laitteen suunnittelussa lähdettiin liikkeelle yksinkertaisista perusasioista, joista tutkimesta laajennettiin, -ehkä turhankin laajalti, erilaisiin ominaisuuksiin, jotka toisivat laitteen käyttöön lisäarvoa.

### 2.1 GPS-järjestelmä ja sen hyödyntäminen työssä

GPS on globaalisti toimiva satelliittiperustainen paikannusjärjestelmä, jossa maanpäälliset vastaanottimet pystyvät laskemaan tarkan sijaintinsa satelliittien lähettämien signaalien kulku-aikerojen perusteella.

Vastaanottimista yleisin malli on kannettavaan käyttöön suunniteltu moduuli, joka sisältää usein samaan IC-koteloon integroituna pienikohinaisen vahvistinasteen, RF-osan sekä mikrokontrollerin, jolla itse laskenta suoritetaan.

Nämä moduulit käyttävät ulostulorajapinnoissaan monenlaisia eri protokollia, mutta ehdottomasti yleisin on NMEA0183-yhteensopiva ASCII-pohjainen, sarjamuotoinen, asynkroninen data. Tämän spesifikaation mukaisia lausetyyppejä on saatavilla ulos myös työssä käytetystä moduulista (Trimble DR+GPS-järjestelmän käyttöohje, sivu 39 alkaen). Kuitenkin käytössä ovat vain GGA- ja RMC-tyypin lauseet.

Standardin mukaan lauseet esitetään ASCII-riveinä, joissa rivi aloitetaan dollarimerkillä sekä sitä seuraavalla lausetyyppitunnuksella (esim. \$GPGGA GGA-lauseen alussa), jonka jälkeen saatavilla oleva informaatio on eroteltu pilkkujen avulla omiin kenttiinsä. Mikäli johonkin kenttään tulevan arvon informaatio puuttuu, jää kenttä tyhjäksi (Trimble DR+GPS -järjestelmän käyttöohje, sivu 43).

Yleisimmin tämän tyyppisestä datasta puhutaan yleisesti CSV:nä (Comma Separated Values, tiedostomuoto, jossa data on erotettu pilkuilla).

## **2.2 FAT ja sen tuomat edut**

FAT on Microsoftin kehittämä tiedostojärjestelmä. Se on yli 30 vuotta vanha järjestelmä joka on tällä hetkellä ylivoimaisesti yleisin siirrettävässä mediassa käytössä oleva tiedostojärjestelmä. Sitä tukevat myös kaikki yleisimmät käyttöjärjestelmät sekä erilaiset sulautetut sovellukset.

Nyt toteutettua järjestelmää helpompaa sekä huomattavasti vähemmän resursseja vaativa tapa olisi toteuttaa levyrutiinit yksinkertaisella kirjoitusoperaatiolla ilman varsinaista tiedostojärjestelmää, mutta tämä vaatisi erityisiä järjestelyjä tallennettua informaatiota luettaessa ja järjestelmän edelleenkehitys sekä laajennettavuus heikkenisivät merkittävästi.

## **2.3 Prototyyppilaitteiston ja sen toiminnan määrittely**

Laitteesta haluttiin ennen kaikkea helppolukuinen ja -käyttöinen. FAT-yhteensopiva tiedostojärjestelmä sekä hyvin laitteessa toimiva massamuisti yhdessä yksinkertaisten näppäintoimintojen kanssa olivat ne pääominaisuudet, joista työn suunnittelu aloitettiin.

Laitteen haluttiin tallentavan dataa vähintään kahtena eri lokitiedostotyyppinä: NMEA-lauseita tavalliseen tekstitiedostoon sekä XML-merkkaukieleen pohjautuvaa KML-syntaksia omaan tiedostoonsa.

Lisäominaisuuksina mietittiin mm. matriisinäyttöä, pitkälle vietyä virranhallintaa, monitasoista informaatiota, rajapintaa mobiililaitteisiin jne. Lisäominaisuuksia kuitenkin vain suunniteltiin ja esitettiin, jotta alkuperäisestä määrittelystä ei poikettaisi liiaksi.

### **3 LAITTEISTON SUUNNITTELU JA TOTEUTUS**

Tässä luvussa esitellään laitteen pääkomponentit, käytetyt ohjelmistot sekä prototyypin kytkentä. Vaikka prototyypin kytkentä eroaa lopullisesta versiosta, ei järjestelmän pääasiallinen toiminta tai toiminnan tulos muutu olennaisesti.

Laitteisto yksinkertaisimmillaan koostuu kolmesta pääosasta: GPS-moduuli, mikrokontrolleri sekä massamuisti. Työssä on käytetty näiden lisäksi mm. mikrokontrolleriin liitettyä näyttöä, jonka avulla laitteen käyttö ja kehitys helpottuvat huomattavasti.

#### **3.1 ATmega128-mikrokontrolleri**

ATmega128 kuuluu Atmelin 8-bittisten mikrokontrollerien perheeseen. Sen ominaisuuksia ovat mm. 20 megahertsin maksimikellotaajuus, 128 kt:n ohjelmamuisti, 4 kt:n RAM, 10-bittinen AD-muunnin, 53 ohjelmoitavaa I/O:ta, SPI jne.

Laitteistoa testattiin aluksi myös muilla prosessorityypeillä. Muiden muassa Texas Instrumentsin MSP430-sarja ja ATmega169 todettiin huonoiksi vaihtoehdoiksi.

#### **3.2 MT-128-kehitysalusta**

MT-128 on bulgarialaisen Olimex Ltd:n myymä kehitysalusta ATmega128-kontrollerille. Se sisältää kontrollerin lisäksi I/O-liityntöjä helppokäyttöisinä nastoina sekä mm. kaksirivisen, 16-merkkisen LCD-näytön, RS232-standardin mukaisen sarjaliitännän DB9S-naarasliittimellä, viisi painonäppäintä, 230 V releen ja piezokaiuttimen. Alusta valittiin työhön pienen kokonsa, suuren I/O-määrän ja halvan hintansa vuoksi.

### **3.3 AVR-USB-JTAG**

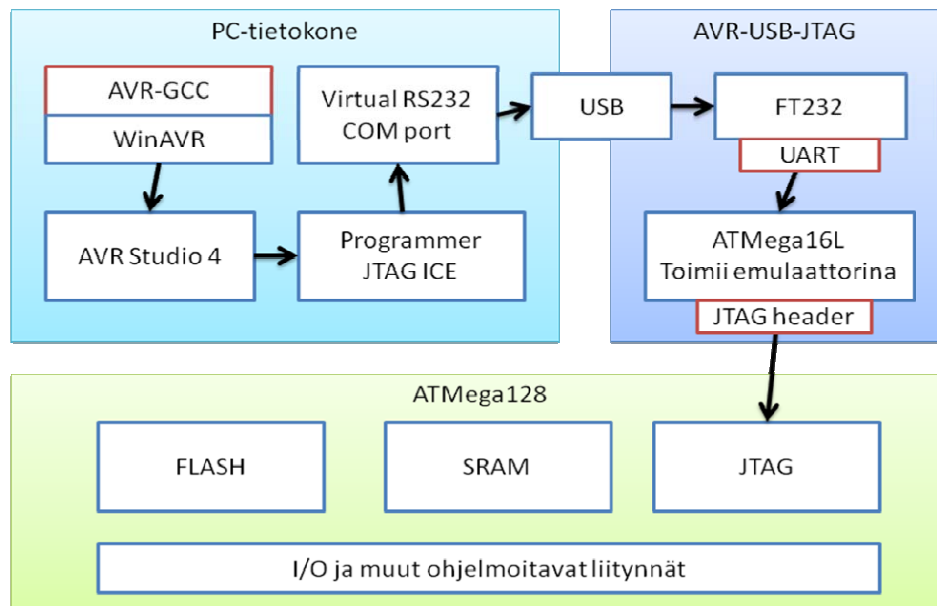
Olimex Ltd valmistaa myös AVR-mikrokontrollerien ohjelmointiin ja debuggaukseen soveltuvia työkaluja. Yksi näistä on AVR-USB-JTAG, joka perustuu Atmelin JTAG ICE -työkaluun ja on tämän karsittu kloonin. Kloonauksen takia ohjelmointityökalu sisältää pääosin samat ominaisuudet kuin sen esikuva.

Työn loppuvaiheessa ohjelmointiin ja debuggaukseen käytettiin myös Atmelin JTAG ICE MK II -työkalua, jossa mm. ”aidon” USB-yhteyden ja kahden ATMega128-prosessorin ansiosta on toiminta kloonina huomattavasti nopeampaa ja varmempaa.

### **3.4 AVR Studio 4, WINAVR**

Sulautetun järjestelmän ohjelmoinnissa pääosan muodostavat kehitysympäristö ja kääntäjä. Kehitysympäristön muodostavat kokonaisuus, jonka pääosia ovat integroitu ohjelmistoympäristö AVR Studio 4, sekä siihen sulautettu open-source-kääntäjäpaketti WinAVR.

WinAVR sisältää periaatteessa kaikki tarvittavat komponentit lähdekoodin kääntämiseen ja kohdelaitteen ohjelmoimiseen (AVR-GCC ja make), mutta AVR Studio graafisena ohjelmointiympäristönä helpottaa työn kulkua huomattavasti. AVR Studio sisältää mm. hyvät debuggausominaisuudet ja liittynät Atmelin tarjoamiin työkaluihin. Näihin työkaluihin kuuluu myös JTAG-ICE, joka on RS232-liittymällä varustettu JTAG-debuggaukseen kykenevä ohjelmointityökalu. AVR-USB-JTAG käyttää hyväkseen tälle työkalulle AVR Studioon luotua rajapintaa.



Kuva 2. Työkalujen liityntä kehitysympäristöön.

AVR-USB-JTAG ei sisällä aitoa USB-liitäntää, vaan USB on toteutettu FTDIChip-yrityksen valmistaman FT232-piirin avulla. Erillisen käyttöjärjestelmäajurin kanssa FT232 muodostaa virtuaalisen COM-portin niin, että sarjaliikenne USB-väylän yli toteutuu RS232-standardin mukaisesti. Aidon JTAG ICE -laitteen sijasta USB-JTAG sisältää aitoa työkalua jäljittelevän, emulointiohjelmalla varustetun ATmega16L-mikrokontrollerin. Käytännössä tämä rajoittaa hieman työkalun nopeutta, mutta ottaen huomioon hintaeron aidon USB-liitäntällä varustetun JTAG-työkalun ja AVR-USB-JTAG:n välillä, on nopeusrajoitus varsin siedettävä.

### 3.5 SD-muistikortti

Secure Digital (lyh. SD) on de facto-standardi yleisestä muistikorttityypistä. Sen ominaisuuksiin kuuluu suuri kapasiteetti pienessä koossa, 2,7-3,6 V:n käyttö- ja signaalijännitealue sekä kopiointi- ja ylikirjoitussuojaus. Korttityypistä on olemassa kolme eri kokoa jotka suurimmasta pienimpään ovat lueteltuna seuraavat: SD, MiniSD ja MicroSD.



Kuva 3. Työssä käytetyt muistikortit. Vasemmalta: MicroSD, MicroSD-adapteri, SD.

GPS-loggeriin SD-kortti valikoitui ennen kaikkea helpon liitettävyytensä perusteella. Verrattuna vastaaviin saatavilla oleviin, halpoihin, siirrettäviin mediatyyppeihin, SD-kortti sisältää monia ominaisuuksia, jotka tekevät siitä houkuttelevan, elleivät jopa ohittamattoman muistityypin harrastajan käytössä.

SD-kortti tukee kommunikointia SPI-väylän avulla, mutta väylän hyödyntämiseksi täytyy kortille viestittää halu käyttää SPI-tilaa. Tämä tapahtuu standardissa määritellyllä komentosarjalla ja voi kestää useita satoja millisekunteja, joten kortin käyttöönottoon tulee varata riittävästi aikaa. (SD-korttistandardi, luku 7)

Kortin virrantarve on hetkellisesti kohtuullisen suuri, jopa useita kymmeniä milliampeereja, joten jos kortin virransaanti on järjestetty suoraan samasta lähteestä mikrokontrollerin kanssa, on huolimattomalla kytkennän järjestelyllä vaarana käyttöjännitteen notkahdus tai värähtely (Chan). Tästä puolestaan voi aiheutua esimerkiksi mikrokontrollerin resetoituminen (ns. Brown Out Detection level) tai datan korruptoitumista. Tämä ongelma voidaan kuitenkin kiertää liittämällä oikein mitoitetut kondensaattorit lähdejännitteen puolelle. MT-128-kehitysalusta sisältää kuitenkin riittävän määrän vakavointikondensaattoreita, joten tätä ongelmaa ei havaittu.

SD-standardi määrittelee myös ylös- ja alaspäin suuntaiset viivakoodit kaikkiiin niihin signaalijohtimiin, joiden tilaa käyttämättömänä ei voida määritellä. Prototyypissä määriteltiin vain sisäiset ylösvedot mikro-ohjaimen datalehden mukaisesti ja todettiin järjestelmän toimivan näennäisesti moitteetta. Yleisesti SD-kortin liitännästä löytyy ristiriitaista tietoa, joten nykyistä kattavampi selvitys on paikallaan, kun sovellusta aletaan kehittää kaupalliseksi.

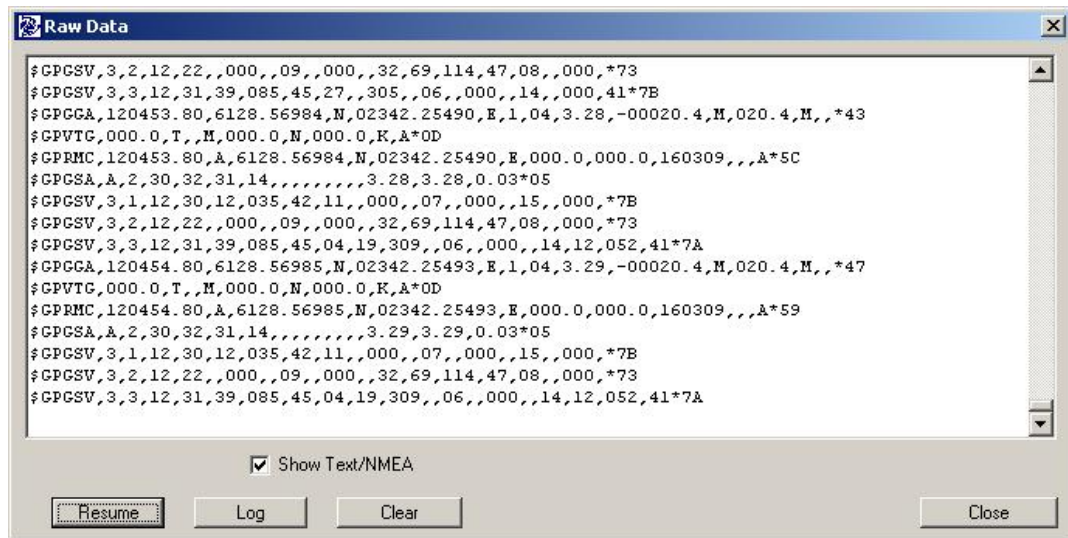
si tuotteeksi. Prototyypissä käytetty MT-128-kehitysalusta käyttää kuitenkin 5 V:n jännitettä kun puolestaan SD-kortin maksimijännitteet ovat 3,5 V ja normaali jännite 3,3 V. Komponenttien välisten jännite-erojen sovitus tehtiin toteuttamalla jännitteenjako 1,8 k $\Omega$ :n ja 3,3 k $\Omega$ :n vastuksilla, joten varsinaista ylösvetoa ei päästy toteuttamaan tai keilemaan.

Prototyypissä käytetty muistikorttityyppi on SD ja lopullisen version korttityyppi on SDmicro. Korttityyppien välillä virrankulutuksessa ei ole lopullisen laitteen kannalta merkittävää eroa, mutta fyysinen kokoero on merkittävä.

### 3.6 GPS-moduuli

Prototyypissä käytetty GPS-moduuli on Trimblen valmistama Lassen DR+GPS. Sen ominaisuuksiin kuuluvat mm. NMEA-muotoinen ASCII-riveinä toteutettu paikkadata sekä HIPPO-formaatti. Moduulissa on myös runsaasti erilaisia muokattavissa olevia lisäominaisuuksia, mutta tässä sovelluksessa siihen ei ole tarvetta. GPS-moduulin signaalointi- ja käyttöjännite on 3,3 V, mutta jännitteensovitusta prototyypin ja moduulin välillä ei tarvittu, koska moduuli on kytkettynä suoraan kehitysalustan toiseen USAR-Tiin ja signaalointi tapahtuu moduulista mikro-ohjaimen suuntaan. Myös signaalitasot ovat hyvin datalehden ilmoittamien minimi- ja maksimiarvojen sisäpuolella.

Kuvassa 4 on esitetty moduulilta saatava informaatio silloin, kun paikannusdata on saatavilla ja vastaanotettu signaali on hyvä. Esimerkkinä voidaan katsoa RMC-lause, joka kuvassa alkaa \$GPRMC-syntaksilla ja jatkuu pilkuilla erotetuilla kentillä, joissa tarvittava informaatio esiintyy järjestyksessä aika (UTC), status (A = active, V = void), latitudi, eteläisen/pohjoisen leveyden ilmaisin (N = pohjoista, S = eteläistä), longitudi, itäisen/läntisen pituuden ilmaisin (E = itäistä, W = läntistä), nopeus, kulma, päivämäärä.



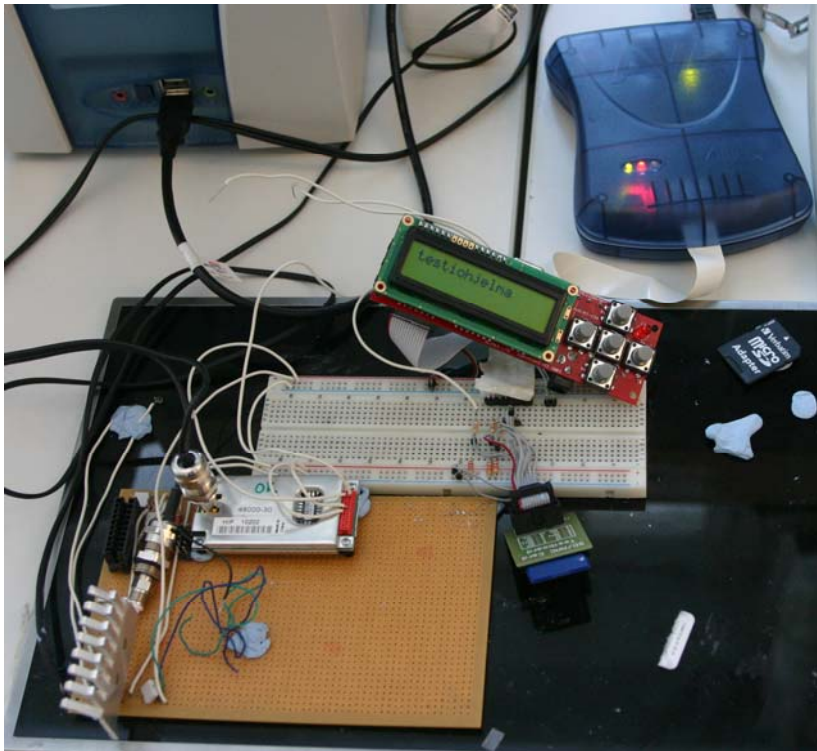
Kuva 4. Moduulilta saatava informaatio.

### 3.7 Prototyypin kytkentä

Kuvassa 5 on esillä varsinaisen prototyypin koekytkentä. Kuvassa näkyy mm. JTAG-työkalu (tällä kertaa JTAG ICE mkII), PC-tietokoneen USB-yhteys sekä kehitysalusta, SD-kortti jännitteenjakokytkentöineen ja GPS-moduuli. Kuvasta puuttuu GPS-moduuliin kytketty antenni.

Liitteessä 1 on esitetty kytkentä ilman näyttöä ja muita kehitysalustan komponentteja. Periaatteessa esitetyllä kytkennällä aikaansaadaan kaikki haluttu toiminta. Näyttö toki helpottaa laitteen käyttöä, mutta ei ole pakollinen.





Kuva 5. Prototyypin kytkentä.

Mikäli prosessorityyppi vaihdetaan 3,3 V versioon ja virransyöttö toteutetaan akuilla, voidaan kytkennästä jättää pois jännitejakovastukset, kaksi diodia ja regulaattori. Akkukennojen lisäksi tarvitaan välttämättömimpinä vain muutama komponentti virranvaihteluiden tasaamiseen. Kuitenkin tulee huomioida aiemmin mainittu SD-kortin aktiivoinnin mahdollisesti aiheuttama käyttöjännitteen notkahdus ja varautua sen esiintymiseen.

#### 4 OHJELMISTON SUUNNITTELU JA TOTEUTUS

Tässä kappaleessa esitellään ohjelmiston toimintaa pääpiirteissään. Tarkemmin ohjelmiston toimintaa on selvitetty liitteisiin lisätyn ohjelmakoodin kommentteissa (liite 1).

Laitteen ohjelmistoon suunniteltiin kolme eri aliohjelmaa ja yksinkertainen pääohjelmasilmutikka. Pääohjelma koostuu ikuisesta silmukasta, jossa pollataan A-portteihin kytkettyjä näppäimiä. Havaittaessa näppäinpainallus tulostetaan näytölle sanoma aliohjel-

masuoritukseen siirtymisestä, kulutetaan aikaa pienen viiveen aikaansaamiseksi ja siirrytään tämän jälkeen itse aliohjelmaan.

Näppäimiä on viisi kappaletta, joista käytössä on neljä. Näppäimistä kolmelle on annettu toiminnaksi pääohjelmasilmuksista aliohjelmasuoritukseen siirtyminen ja neljännellä palataan takaisin pääohjelmasilmuksaan.

Aliohjelmaan siirtymisen jälkeen järjestelmä odottaa, kunnes saadaan kelvollinen GPS-paikkatieto. Tämä tarkoittaa sitä, että mikro-ohjaimen vastaanottama lause sisältää jonkinlaisen koordinaattidatan. Tässä luotetaan GPS-moduulin toimintaan ja tarkkuuteen, eikä oteta kantaa siihen, mikäli koordinaattidata ei pidä paikkaansa, se puuttuu kokonaan tai vastaanotettu data on korruptoitunutta.

#### **4.1 Käytetyt kirjastot ja funktiot**

Varsinainen ohjelmisto tukeutuu pitkälle valmiisiin kirjastoihin sekä erilliseen FAT-tuella varustettuun levyrutiinikirjastoon. FAT-tiedostojärjestelmää on alkanut esiintyä entistä enemmän projekteissa, joiden kirjastot ovat vapaasti käytettävissä.

Tässä työssä on käytetty valmista, kehitysalustalle kirjoitettua ohjainkirjastoa sekä microsoft FAT -yhteensopivaa FatFS-tiedostojärjestelmän karsittua versiota, Tiny-FatFS:ää, joka on saatavilla vapaasti ei-kaupallisiin tarkoituksiin. ([http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)) Myös monia muita vapaasti saatavilla olevia kirjastoja kokeiltiin, mutta niiden toiminta oli epävakaa ja ominaisuudet hyvin rajalliset. Tiny-FatFS on myös kehitysvaiheessa oleva järjestelmä, josta työn edetessä löydettiin virheitä. Esimerkiksi 512 Mt muistikortit eivät jostain syystä toimi järjestelmässä, vaan tunnistuvat väärin.

#### 4.1 GPS-data näytölle

Aliohjelma `NMEA_on_display()` tulostaa näytölle GPS-moduulilta paikkadataa WGS84 datum -perustaisessa desimaalimuodossa. Näppäinkomennon jälkeen näytölle tulostetaan viesti ”Koordinaatit näyttöön” ja siirrytään aliohjelmaan. Aliohjelmassa odotetaan riviä, joka sisältää RMC-lauseen, noukitaan lauseesta koordinaattitieto, joka muunnetaan desimaalimuotoon. Tämän jälkeen desimaalimuotoiset liukulukumuuttujat muunnetaan merkkijonoiksi, jotka yhdistetään merkkijonoksi, joka tulostetaan näytölle.



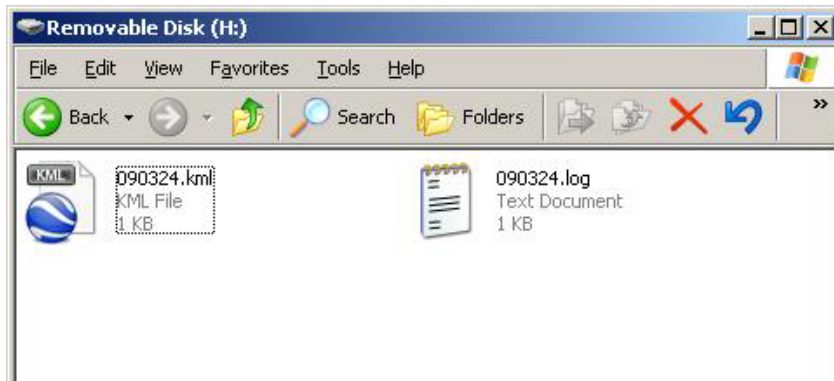
Kuva 6. GPS-koordinaatit näytöllä.

#### 4.2 Paikkadatan tallennus SD-kortille

Aliohjelma `NMEA_on_sd()` tallentaa kortille riveittäin ”raakaa” NMEA-lausedataa CSV-muodossa. Näppäinkomennon jälkeen näytölle tulostetaan viesti ”NMEA-tallennus” ja siirrytään itse aliohjelmaan. Aliohjelmassa alustetaan tiedostojärjestelmä, odotetaan riviä, joka sisältää RMC-lauseen, noukitaan RMC-lauseesta päiväystieto ja avataan tiedosto päiväystiedon perusteella. Mikäli tiedosto on jo olemassa, jatketaan tallennusta sen lopusta. Mikäli korttia ei ole, sitä ei voida alustaa tai sille ei voida kirjoittaa, ohjelmasuoritus havaitsee virheen, tulostaa näytölle virhesanoman ja palaa pääohjelmasilmuksaan.

Ohjelmakoodissa voidaan erikseen määrittellä, mitä NMEA-lauseita laitteen halutaan tallentavan. Laite voi tämän määrittelyn perusteella myös tallettaa useampaa lausetta kerrallaan. Lopulliseen tuotteeseen sisällytetty moduuli tietysti voi käyttää hyväkseen kaikkia mahdollisia lauseita ja lauseiden tallennus voidaan muuttaa ohjelmakoodiin niin, että haluttu lause voidaan valita näppäinkomennolla.

Paikkadata tallentuu kortille FAT-yhteensopivassa muodossa tekstitiedostona, jolloin se voidaan lukea miltei missä tahansa PC- tai Mac-tietokoneessa ja tulkita halutulla tavalla. Tietyissä tapauksissa data voidaan myös siirtää muihin laitteisiin jatkokäyttöä varten (plotterit, navigointitietokoneet, digikamerat).



Kuva 7. Paikkadata kortilla (tiedosto avoinna Windows XP -käyttöjärjestelmässä)

### 4.3 Paikkadatan tallennus KML-muodossa

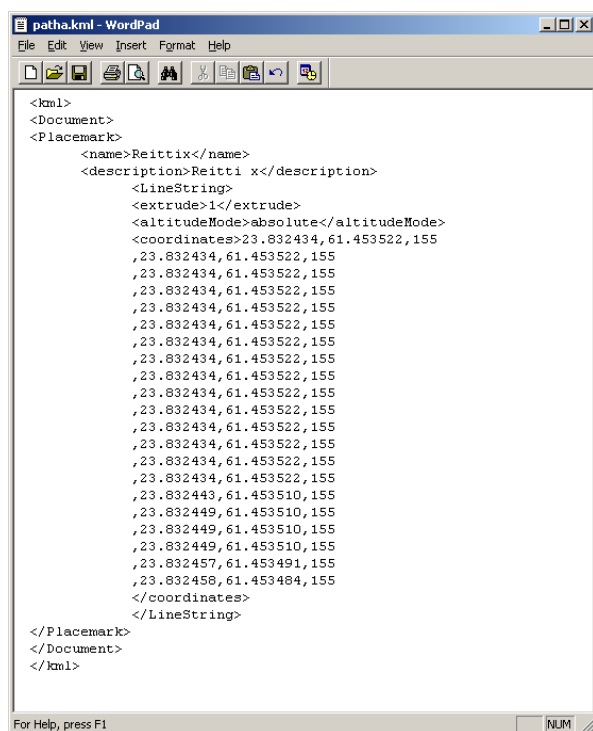
Aliohjelma KML\_on\_sd() on ikään kuin kahden aiemmin esitetyn aliohjelman hybridi. Siinä vastaanotettu paikkadata muunnetaan desimaalimuotoon, minkä jälkeen se tallennetaan SD-kortille XML-kuvauskielen lauseina KML-yhteensopivaan tiedostoon.

KML-merkkaukieli pitää sisällään keskeisinä osina graafisen esityksen ja paikkadatan. KML-tiedostolla voidaan kuvata mm. rakennusten ulkolinjoja, kuljettuja reittejä, maastonmuotoja, kiinnostavia kohteita kartalla ja näiden ominaisuuksia. Tämän kielen ehkä käytettävvin ominaisuus kuitenkin harrastajan kannalta lienee eri paikkojen ja reittien merkkauksmahdollisuus yleisesti käytetyissä ohjelmissa sekä World Wide Webissä.

Ohjelmistoon valittiin KML-merkkaukieli nimenomaan sen helppokäyttöisyyden ja monikäyttöisyyden vuoksi. Käytännössä ohjelmisto vain kirjoittaa kortille suoraan tiedostotyyppin kuvauskehukset ja tulostaa paikkadatan koordinaattipisteitä peräkkäin niin, että tiedosto voidaan tulkita kuljettuna reittinä. Kun tiedosto puretaan kortilta luettavak-

si, voidaan sitä käyttää lukemattomissa eri sovelluksissa, jotka automaattisesti lukevat ja muuntavat kuvauksen halutulla tavalla.

Kuvassa 8 on esitys laitteiston tallentamasta KML-tiedostosta. Ohjelma tuottaa nykyisessä versiossaan KML-kuvauksia, jotka esittävät reittejä absoluuttisella korkeusmerkinnällä. Reitti määritellään tulostiedostossa sijoittamalla pilkuilla erotettuja, desimaalimuotoisia koordinaattipisteitä <LineString> ja <coordinates>-elementtien väliin.



Kuva 8. Laitteiston tallentaman KML-tiedoston rakenne.

## 5 LAITTEEN TESTAUS JA KEHITYSMAHDOLLISUUDET

Tässä luvussa käsitellään laitteen kehityksen onnistuneisuutta ja mitataan sen suorituskykyä sekä arvioidaan eri kehitysvaihtoehtoja, joilla laitteen suorituskykyä ja muita ominaisuuksia voidaan parantaa.

## 5.1 Ohjelmiston testaus

Ohjelmiston kehitysvaiheessa suoritettiin luonnollisesti suurin osa ohjelmiston varsinaisesta vianetsinnästä. JTAG-debuggerin avulla käytiin ohjelmistoa läpi rivi riviltä tai vaihtoehtoisesti ajettiin ohjelmasuoritus virhetilanteeseen, minkä jälkeen voitiin arvioida tarvittava muutos virhetilasta toipumiseksi tai sen poistamiseksi. Kuitenkin tietyssä pisteessä ohjelmistokehitystä tulee aina vaihe, jolloin ohjelmiston arvioidaan olevan käyttövalmis vaikka järjestelmä itsessään ei olisi täydellinen.

### 5.1.1 Käyttöliittymättestaus

Laitteen käyttöliittymä jättää hieman toivomisen varaa. Näppäinkomentojen vasteet ovat melko hitaat, jolloin nopea painallus ei tuota toivottua tulosta. Hitauteen vaikuttaa mm. se, että SD-kortin toiminnat vaativat oman aikansa ja järjestelmä etenee ohjelman mukaisesti eli rivi kerrallaan. Mikäli järjestelmä olisi reaaliaikainen tai ohjelmassa olisi haarautumisen sallittu, voisi tilanne olla toinen ja käyttö ainakin näennäisesti nykyistä sujuvampaa.

Näyttö on myös hitaanlainen, mutta kuitenkin helppolukuinen, vaikkakin mm. eteläisen ja pohjoisen erottelu ja käytetty koordinaattiesitystapa on jätetty pois ja laitteeseen perehtymätön käyttäjä saa arvailla koordinaattien oikeellisuutta. Näytöllä ei myöskään kerrota NMEA-lauseiden vastaanoton tai kortille kirjoittamisen onnistumisesta ja jälleen käyttäjä jää arvailemaan, onko laitteen käyttötapa oikea. Lyhyen käytön jälkeen näppäimistön ja näytön hitauteen kuitenkin tottuu ja käyttö helpottuu, mutta lopullisesta tuotteesta nämä ominaisuudet tulisi karsia pois kokonaan.

Käyttöliittymä on havainnollinen, ja pelkällä näppäinten merkinnällä tai jopa ilman sitä laitteesta saadaan intuitiivinen käyttöä nykyisellään. Käyttöliittymän muutokset voidaan helpohkosti toteuttaa vastaavalla näppäinjärjestelyllä ja sen laajennettavuus on hyvä.

### 5.1.2 Toiminnallisuustestaus

Laite ilmaisee suurimman osan mahdollisista virhetilanteista ja niistä toipuminen on kohtuullista, joskaan ei pomminvarmaa. Kun kyse on prototyypistä, pienet jumiutumiset varsinaisen toiminnan todentamisen ohessa ovat pikkuseikkoja.

Suurin toiminnallisuuteen vaikuttava seikka on signaalin saatavuus. Ohjelmistossa ei ole käytännössä mitenkään varmistettu signaalin saatavuutta, ja mikäli joko GPS-signaali tai moduulilta saatava sarjaliikenne katkeaa, tulee ohjelmassa virhesuoritus.

Testattaessa selvisi, että näytölle tulostuu tällöin pelkkiä nollia ja korteille tallennus alkaa vasta, kun moduulilta saatu koordinaattidata sisältää informaatiota (ts. CSV-lauseissa latitudi- ja longitudi-kentissä on informaatiota).

Tämän lisäksi tallennuksen onnistuminen on käytännössä käyttäjistä kiinni, sillä ohjelmisto sulkee avoinna olevan tiedoston vasta, kun aliohjelmasta siirrytään takaisin pääohjelmasilmukkaan. Tämä tarkoittaa sitä, että ohjelmisto saattaa hukata tallennettua dataa, mikäli SD-kortti poistetaan telineestään ennen pääohjelmasilmukkaan palaamista tai laitteen virransaanti katkaistaan kesken tallennuksen.

Aliohjelmasta toiseen siirtyminen ei näin ole sujuvaa ja periaatteessa turha päävalikossa vierailu kuluttaa aikaa ja käyttäjän hermoja.

## 5.2 Laitteen testaus

Itse laitetta testattiin prototyypiversiona käyttämällä sitä mm. kävellen ja autoillessa. Laitteen ominaisuuksia testattiin myös ns. laboratorio-olosuhteissa, kun haluttiin selvittää sen virrankulutus ja arvioida akkutoiminnan mahdollisuutta.

GPS-moduulin käyttöohjeessa kerrotaan sen tarvitsevan pienikohinaisen vahvistimen sekä aktiivisen antennin toimiakseen oikein. Käytetty antenni on REEL Reinheimer Elektronik GmbH:n valmistama k60tam-tyypin magneettikiinnitteinen, pyöreä, korkean gainin omaava GPS+GSM-yhdistelmäantenni.

Havaittiin, että kyseisellä antennilla ei päästä moduulin vaatimaan signaalitasoon, mutta esteettömän taivasnäkyvän vallitessa, ei varsinaisessa paikannustarkkuudessa tai -nopeudessa havaittu merkittäviä muutoksia. Tampereen Teknillisen Yliopiston opiskelijoiden elektroniikkakerho TELOKissa on todettu moduulin toimivan jopa täysin passiivisella antennilla.

### 5.2.1 Virrankulutus ja akkutoiminta

Laitteiston virrankulutusta vain arvioitiin mitattavien kohteiden vaihtelevan virrankulutuksen vuoksi. Mittalaitteina käytettiin Fluke 75 III -yleismittaria sekä Fluke 3384B -oskilloskooppia. Laitteistosta mitattiin yksi osa kerrallaan yhdistämällä mittalaitteet vain tutkittavan osan virransyöttöön. Oskilloskoopilla varmistettiin toiminta ja yleismittarin näytöltä luettiin osan virrankulutus.

GPS-moduulin virrankulutus mitattiin kohtuullisen tasaiseksi n. 50 mA mikä ei riipu esim. antennin sijoituksesta. Myös antennin virrankulutus oli tasainen 22 mA, eivätkä ulkoiset tekijät muuttaneet kulutusta.

Kehitysalustan virrankulutus mitattiin n. 40 mA:n suuruiseksi, mutta tulee huomioida, että kehitysalusta sisältää näennäisesti turhia, virtaa vieviä komponentteja, joilla ei ole laitteen toiminnan kannalta olennaista merkitystä ja joita ei lopullisessa tuotteessa tarvi-



ta. Myös kehitysalustan prosessorityyppi ei ole vähävirtainen versio, vaan 5 V käyttäjännitteellä varustettu ”perusversio”.

Kun SD-kortin virrankulutukseksi arvioidaan muutamia milliampeereita, saadaan prototyypissä käytetyn laitteiston tyypilliseksi virrankulutukseksi tilasta riippuen 112 - 142 mA. Tästä voidaan karkeasti arvioida esim. 1100 mAh Li-ion-akulla laitteen toiminta-ajaksi noin kahdeksan tuntia.

Virrankulutukseen voidaan vaikuttaa jonkin verran eri komponenttivalinnoilla sekä kelloaajuutta laskemalla. Markkinoilla on esim. suhteellisen huokeita GPS-moduleita, joissa mikrokontrolleri on ohjelmoitavissa ja virrankulutus on huomattavasti pienempi kuin nyt käytetyssä.

#### **5.4 GPS-lokin tarkkuus**

Järjestelmän tarkkuutta mitattiin ajamalla summittain tunnettuja, karttoihin hyvin merkittyjä reittejä useaan kertaan edestakaisin, jolloin saatiin osviittaa laitteen ilmoittaman paikkadatan hystereesistä. Tätä hystereesiä yritettiin selvittää myös pysähtymällä pitkiksi ajanjaksoiksi paikoilleen tunnettujen maamerkkien läheisyyteen ja seuraamalla laitteen käyttäytymistä ottamalla koordinaatit näytölle ja myöhemmin lokitiedostoja analysoimalla.



Kuva 9. Hervannan katu Google Earth-ohjelmassa.

Kuvassa 9 on esitetty Tampereen Hervannan kaupunginosassa sijaitsevan liikenneympyrän ylitys muutamia sekunteja GPS-moduulin päällekytkennän jälkeen. Lopullinen virhe pienenee hieman ajan kuluessa.

Mittauksissa havaittiin, että prototyypin tallentamissa koordinaateissa on virhe pienempi kuin  $\pm 7$  metriä. Laitteen tarkkuus on siis kohtuullinen, joskin uusilla moduleilla tai käytämällä tarjolla olevia korjausmuotoja, kuten DGPS, saataisiin tarkkuutta parannettua huomattavasti.

## 5.5 Lopullinen tuote

Alkujaankin harrastustarkoituksiin määritellyn laitteen toteutuksen pohjalta olisi turhaa yrittää luoda lopullinen, spesifinen laite, jonka laajennettavuus haluttaisiin säilyttää maksimaalisena. Sen sijaan voidaan esittää toimivan prototyypin ja sen ohjelmakoodin perustalta hypoteettisia oletuksia lopullisesta versiosta, joka ei eroa varsinaiselta toiminnaltaan työssä toteutetusta prototyypistä miltei lainkaan, vaikka se muuten olisi-kin ominaisuuksiltaan erilainen.

Toiminta-aika, koko, tarkkuus, käytettävyys ja ennen kaikkea hinta ovat sellaisia ominaisuuksia, joita laitteelta odotetaan kaupallisessa tuotannossa. Lopullisen tuotteen laitteistoon ja ohjelmistoon tulisi sisällyttää mm. seuraavia parannuksia:

- AD-muunnos, jolla tarkkaillaan akkujen tilaa ja tarvittaessa keskeytetään levyrutiinit, jotta tiedostoja ei menetetä.
- Näytön ja GPS-moduulin virranhallinta logic-level-FETillä ja virransäästöominaisuus itse mikro-ohjaimen niin, että pitkän paikallaanolon jälkeen laite siirtyy automaattisesti lepotilaan.
- SD-kortin tms. massamuistin ”hot plugging”-mahdollisuus niin, ettei datan korruptoitumisesta tarvitse huolehtia, vaan kortti voidaan poistaa kesken kirjoituksen.
- Järjestelmä tulisi toteuttaa joko reaaliaikakäyttöjärjestelmän tyyliä tai vaihtoehtoisesti vähintään näppäinkomennot tulisi toteuttaa keskeytyspohjaisesti niin, että ainakin näennäinen toiminta näppäimen painamisen jälkeen olisi välitöntä, eikä vasteaika olisi riippuvainen aliohjelmasilmuksen kierrosta.

Laitteiston kehitystä silmällä pitäen testattiin eri komponentteja ja ohjelmistomuutoksia, mm. 132x132 pikselin värinäyttöä, jonka virrankulutus on alle 50 mA sekä vain muutamien millimetrien kokoista pintaliitettävää GPS-moduulia, jonka virrankulutus jää alle 36 mA:n.

Myös ohjelmiston supistamista yritettiin, mutta nykyisillä koodikirjastoilla ja halutuilla ominaisuuksilla osoittautui mahdottomaksi pienentää ohjelmiston RAM-käyttöä 1 kt pienemmäksi ilman vakavia ongelmia. Tämän muistirajoituksen asettaa pääasiassa ensisijaisesti käytetyn FAT-tiedostojärjestelmän vaatiman ohjelmakoodin minimimuistivaatimus (Chan). Uudelleenkirjoittamalla FAT-rutiinit, käyttämällä 32-bittistä prosessoria tai käyttämällä kokonaan eri kirjastoa, voitaisiin järjestelmän kustannuksia pienentää olennaisesti.

Laitteen fyysisen koon ja virrankulutuksen pienentäminen sekä käytettyjen komponenttien vähentäminen on mahdollista mm. prosessoryyppiä ja GPS-moduulia vaihtamalla tai käyttämällä mahdollisesti moduulia, johon ohjelmistokoodi sekä oheislaitteet (massamuisti ja muut I/O-laitteet) voitaisiin suoraan sisällyttää.

Tuotteen kaupallisessa tuotannossa tulee huomioida, että NMEA-standardi ja SD-korttistandardi ovat maksullisia. Myös Tiny-FatFS sekä FatFS ovat tekijänsä omaisuutta eivätkä kehitysasteella oleva tiedostojärjestelmä sekä massamuistirajapinta sovellu täydellisen tai edes hyvän kaupallisen tuotteen valmistukseen.

## **5.6 Kehitys- ja laajennusmahdollisuuksista**

Kehitys- ja laajennusmahdollisuuksia paikannussovellusten alalla on lähes rajaton määrä yksinkertaisenkin järjestelmän kohdalla. Jokainen järjestelmä toimii saman perusolettamuksen alla ja saman paikannusinformaation pohjalta, jolloin erilaisten sovellusten sulauttaminen on helppoa. Myös vapaita kirjastoja eri sovelluksiin on hyvin saatavilla, joten sulauttaminen tapahtuu myös ohjelmoinnin puolesta halutessa korkealla tasolla (C, C++).

Laitteen ohjelmiston yksinkertaisuus sekä käytetyn mikrokontrollerin hyvä laajennettavuus (vasta puolet SRAMista ja kymmeniä I/O-liityntöjä käyttämättä) antavat hyvän alustan jatkokehitykselle.

## **6 YHTEENVETO**

Koska etukäteen tiedettiin, mitä laitteelta haluttiin, olisi myös luullut, että itse laitteen toteuttaminen ja suunnittelu olisi helppoa. Kuitenkin laitteen määrittely muuttui hieman työn edetessä, koska siihen haluttiin pakata liikaa ominaisuuksia. Tämä johti siihen, että haluttuja lisäominaisuuksia ei saatu toteutettua esittelykelpoisiksi, mutta myös itse laitteen toteutus ja ohjelmiston päärunko kärsivät, koska suunnittelu- ja määrittelyvaiheisiin jouduttiin palaamaan aina uudelleen.

Ohjelman suunnitteluvaiheessa kiinnitettiin huomiota haluttuihin ominaisuuksiin, mutta jätettiin liikaa varaa laajennettavuudelle ja ohjelmiston varsinainen tarkka määrittely sivuutettiin sillä periaatteella, että vältettäisiin mahdolliset umpikujat itse ohjelmaa kirjoitettaessa. Ohjelman kirjoitusvaiheessa ohjelmaa testattiin aina uuden toiminnallisuus-

den lisäämisen jälkeen ja huomattiin, että suunniteltua ohjelmankulkua tai funktioita ei saatu toteutettua joko jonkin tuntemattoman virheen seurauksena tai parametrikäsittelyn tms. epäonistuttua. Ohjelmasta kirjoitettiin 12 eri versiota, joista neljässä käytettiin hyväksi eri näyttöä ja muita lisäkomponentteja.

Laitteen fyysiseen toteutukseen ei aikaa mennyt juurikaan, mutta komponenttien kanssa törmättiin moniin eri ongelmiin ja itse laitteen toiminnan ymmärtäminen vaati hyvinkin suuren panoksen ajallisesti. Vaikkakin esim. ohjelmiston kirjastot eivät ole itse kirjoitettuja, tuli niiden toiminta ymmärtää, jotta niitä voitiin käyttää, sillä liitynnät, ajastukset yms. tuli sovittaa erikseen ja virheet korjata käytetystä koodista.

Myös lyhyt vertailu kaupalliseen tuotteeseen suoritettiin (myLOGGER). Työn tuloksena kehitetyn järjestelmän suorituskyky eroaa tästä hieman. Kaupallinen tuote on 60 euron hintainen, joka vastaa hyvin odotuksia. Siinä on 230 mAh:n akku, jolla sen toiminta-ajaksi luvataan neljä tuntia, joka on huomattavasti parempi suorituskyky, kuin työssä esitellyllä prototyypillä. Toisaalta taas mikäli prototyypin muistityyppi vaihdettaisiin, näyttö poistettaisiin ja moduuli vaihdettaisiin antenneineen vähävirtaisempaan malliin, päästäisiin suunnilleen samoihin suorituslukemiin. Kaupallisen tuotteen TTFF (Time To First Fix) eli paikannusaika käynnistyksestä on hieman suurempi kuin prototyypiversiön vastaava.

Työn lopputuloksena viivästyksistä ja epäonnistumisista huolimatta saatiin valmistettua halutulla tavalla toimiva laite aiemmin täysin vieraista komponenteista.

## LÄHTEET

Trimble DR+GPS-järjestelmän käyttöohje.

[online] [viitattu 25.3.2009]

[http://trl.trimble.com/dscgi/ds.py/Get/File-362863/DR\\_GPS\\_Manual\\_58059-00\\_RevA.pdf](http://trl.trimble.com/dscgi/ds.py/Get/File-362863/DR_GPS_Manual_58059-00_RevA.pdf)

ATMega128-mikrokontrollerin datalehti.

[online] [viitattu 26.3.2009]

[http://www.atmel.com/dyn/resources/prod\\_documents/doc2467.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf)

SD Association 25.9.2006. SD-kortin spesifikaation yksinkertaistettu versio 2.0.

[online] [viitattu 25.3.2009]

[http://www.sdcard.org/developers/tech/sdcard/pls/Simplified\\_Physical\\_Layer\\_Spec.pdf](http://www.sdcard.org/developers/tech/sdcard/pls/Simplified_Physical_Layer_Spec.pdf)

“Chan”. Electronic Lives Mfg. Tiny-FatFS-tiedostojärjestelmä sekä rajapintakirjasto.

[www-sivu] [viitattu 25.3.2009]

[http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)

“Chan”. Electronic Lives Mfg. Tiny-FatFS ominaisuuksia, mm. muistinkäyttö.

[www-sivu] [viitattu 25.3.2009]

<http://elm-chan.org/fsw/ff/en/appnote.html>

“Chan”. Electronic Lives Mfg. SD-kortin käyttö sekä rajapinta.

[www-sivu] [viitattu 25.3.2009]

[http://elm-chan.org/docs/mmc/mmc\\_e.html](http://elm-chan.org/docs/mmc/mmc_e.html)

myLOGGER. Kaupallisen GPS-loggerin tekniset tiedot.

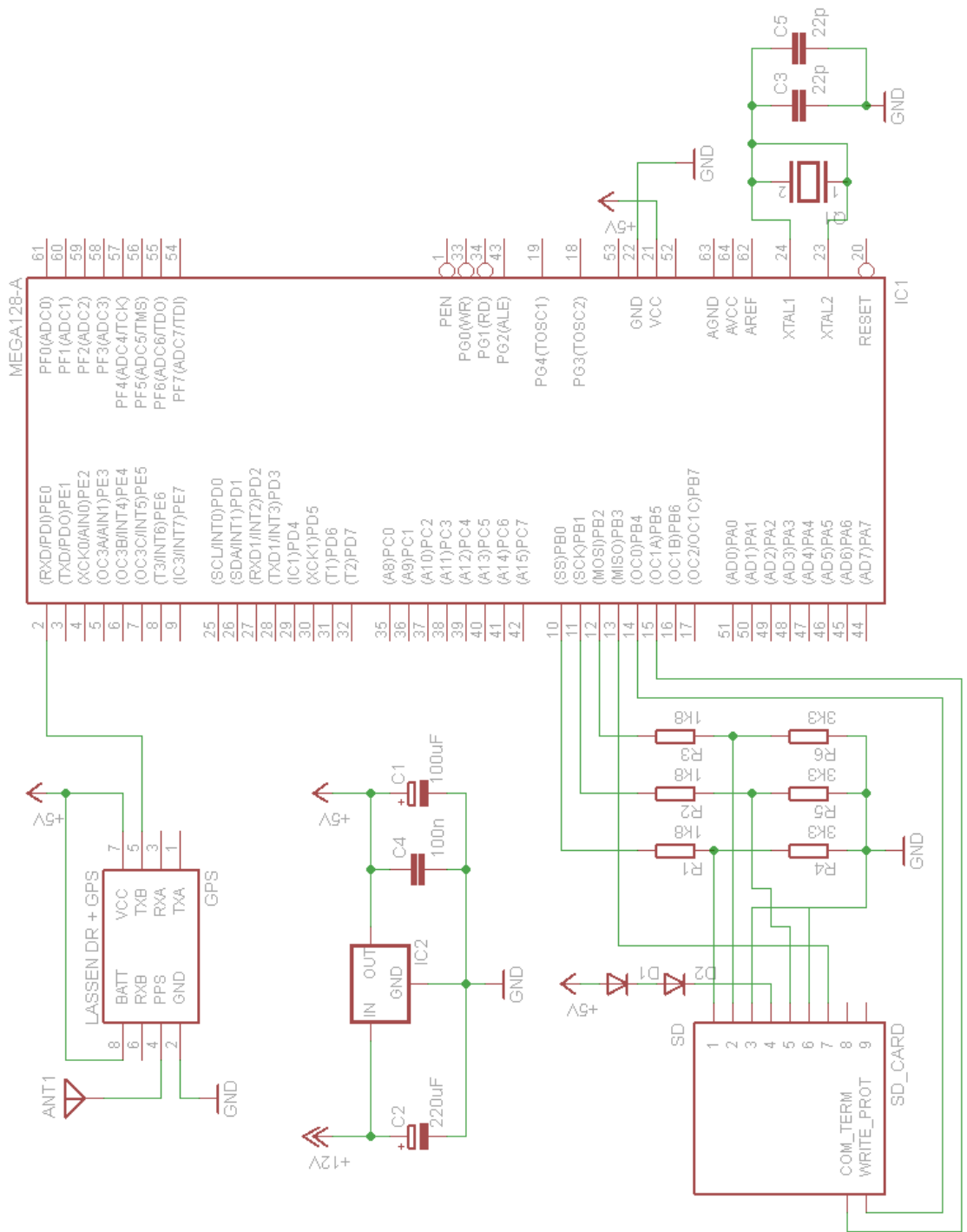
[online] [viitattu 26.3.2009]

[http://www.mylogger.fi/myLOGGER\\_Tekniset\\_tiedot.pdf](http://www.mylogger.fi/myLOGGER_Tekniset_tiedot.pdf)

## LIITTEET

- Liite 1. Kuva prototyypin koekytkenästä
- Liite 2. Ohjelmiston lähdekoodi

# LIITE 1. Laitteen kytkentäkaavio ilman näyttöä ja näppäimiä. 1(1)





```

/*GPS-loggeri */
/*Jussi Saarijoki, 2009*/

/* käytetyt kirjastot */

#include <avr/io.h>
#include <avr/pgmspace.h>
#include <avr/interrupt.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

/* sarjaliikenteen ja levyrutiinien kirjastot */

#include "uart.h"
#include "tff.h"
#include "diskio.h"

/* näytön ja näppäimistön includet */

#include "bits.h"
#include "lcd.h"
#include "delay.h"

/* näppäimistön määrittely*/

#define B1 (PINA&BIT0) //ylös
#define B2 (PINA&BIT1) //vasen
#define B3 (PINA&BIT2) //keskinäppäin
#define B4 (PINA&BIT3) //oikea
#define B5 (PINA&BIT4) //alas

/* globaalit */

FATFS fatts; /* Tiedostojärjestelmä loogiselle asemalle */
BYTE Buff[100]; /* Työpuskuri */

char* endptr;
double Latitud;
double Longitud;
unsigned int altitude;

char *Alt = NULL;
char *Latitude = NULL;
char *Longitude = NULL;
char *North = NULL;
char *West = NULL;
char *Fix = NULL;
double minutes;
double degrees;
char Lat[30];
char Long[12];

BYTE b, *p = NULL;
WORD s;
FIL pointFile;
FIL pathFile;

```

```
/* I/O - väylien alustus */
```

```
static
void lolnit ()
{
    PORTA = 0b00000000; // Port A - Liitynät näppäimille, PA0-PA4
    DDRA = 0b00000000; // Pin 7 ulos (koekytkentäkortin rele) muut sisääntuloiksi

    PORTB = 0b10110000; // Port B - Sisältää SPI-väylän sekä levyn kirjoitusuojat
    DDRB = 0b11000000;

    PORTC = 0b00000000; // Port C - Liitynät koekytkentäkortin näytölle
    DDRC = 0b11110111;

    PORTD = 0b11111111; // Port D - Käyttämättä

    PORTE = 0b11110010; // Port E - Sarjaväylä GPS:lle, RXD0 ja TXD0
    DDRE = 0b10000010;

    PORTF = 0b11111111; // Port F - Käyttämättä

    PORTG = 0b111111; // Port G - Käyttämättä

    uart_init(); // Alustetaan UART
    sei(); // Keskeytysten sallinta
}

```

```
/* Rivivertailu. Saa parametreiksi merkkijono-osoitteen ja vertailtavan merkkijonon.*/
/* Mikäli parametrit täsmäävät, palauttaa nollan.*/
```

```
static
BYTE gp_comp (BYTE *str1, const prog_uint8_t *str2)
{
    BYTE c;
    do {
        c = pgm_read_byte(str2++);
    } while (c && c == *str1++);
    return c;
}

```

```
/* Sarake-erottelu. Saa parametreina lause-osoittimen ja sarakkeen järjestyslunun.*/
/* Palauttaa järjestyslunun paikkaa vastaavan sarakkeen sisällön ja nollan, mikäli sarake on tyhjä.*/
```

```
static
BYTE* gp_col (const BYTE* buf, BYTE col) {
    BYTE c;
    while (col) {
        do {
            c = *buf++;
            if (c <= ' ') return NULL;
        } while (c != ',');
        col--;
    }
    return (BYTE*)buf;
}
/* Rivin haku USART:ilta. Palauttaa tavujen lukumäärän ja nollan, jos tapahtuu virheellinen suoritus. */
```

```
static
BYTE get_line (void)
```

```

{
  BYTE c, i = 0;
  for (;;) {
    c = uart_get();
    if (!c || (i == 0 && c != '$')) continue; // Jos ei riviä tai rivin alussa muu merkki kuin dollari
    Buff[i++] = c; // Tallentaa indeksoidun merkin puskuriin
  (Buff[100])
    if (c == '\n') break; // Jos rivi loppuu, tallennus lopetetaan
    if (i >= sizeof(Buff)) i = 0;
  }
  return i;
}

```

```

/* Aliohjelma, jolla tulostetaan desimaalikoordinaatit näytölle */
/*****

```

```

void NMEA_on_display(void){

```

```

    SEND_CMD(CLR_DISP); // Tyhjennetään näyttö

```

```

    while(get_line()){ // Niin kauan, kun saadaan kelvollisia rivejä..
        if (!gp_comp(Buff, PSTR("$GPRMC"))) // Jos vastaanotetaan RMC-lause
        {

```

```

            Latitude= gp_col(Buff, 3); // RMC-lauseen neljäs kenttä sisältää latitudin
            Longitude = gp_col(Buff, 5); // RMC-lauseen kuudes kenttä sisältää longitudin

```

```

            North = gp_col(Buff, 4); // Pallonpuolisko-indikaattorit
            West = gp_col(Buff, 6);

```

```

            Longitud = strtod(Longitude, &endptr); //Muutetaan merkkijono desimaaliluvuksi
            minutes = modf(Longitud/100, &degrees); //Eristetään desimaaliosa
            Longitud = degrees + (minutes*100)/60; //Pistetään takaisin "kasaan"

```

```

            Latitud = strtod(Latitude, &endptr); //Sama tehdään latitudille
            minutes = modf(Latitud/100, &degrees);
            Latitud = degrees + (minutes*100)/60;

```

```

            if(North == 'S') Latitud = -Latitud; //eteläinen vai pohjoinen pallonpuolisko?

```

```

            if(West == 'W') Longitud = -Longitud; //itäinen vai läntinen pallonpuolisko?

```

```

            sprintf(Lat, "%f", Latitud); // Näillä riveillä liukuluvut tulostetaan takaisin
            sprintf(Long, "%f", Longitud); // merkkijonoon
            strncat(Lat, " ", strlen(Lat) + 6);
            strncat(Lat, Long, strlen(Lat) + strlen(Long) - 1);

```

```

            SEND_STR(Lat); // Merkkijono tulostetaan näytölle
        }

```

```

        if(B5 == 0){ // Jos kesken ohjelmasuorituksen painetaan näppäintä,
            SEND_CMD(CLR_DISP); // palaa ohjelmasuoritus päävalikkoon
            break;
        }
    }
}

```

```

/* Koordinaattidatan tallennus SD-kortille */
/*****

```

```

int NMEA_on_sd(void){

```

```

disk_initialize(0); // Levyn ja tiedostojärjestelmän alustus
f_mount(0, &fatfs);

do{

    get_line(); // Odotetaan lausetta, josta saadaan poimittua päivämäärä

}while(gp_comp(Buff, PSTR("$GPRMC")));

    p = gp_col(Buff,9); // Poimitaan kymmenennessä kentässä sijaitseva päivämäärä

    strcpy_P(&Buff[6], PSTR(".log")); // Avataan tiedosto nimellä "päiväys.log"

    if (f_open(&pointFile, Buff, FA_OPEN_ALWAYS | FA_WRITE)
        || f_lseek(&pointFile, pointFile.fsize)) {

        SEND_STR("AVAUSVIRHE!"); // Mikäli tiedoston avaus epäonnistuu,
        delay_ms(50); // virheilmoitus ja palataan pääohjelmaan
        return 0;
    }

// Ohjelma tallentaa tässä määritellyt lausetyypit.

while ((b = get_line()) > 0){
    if ( !gp_comp(Buff, PSTR("$GPGGA"))
        || !gp_comp(Buff, PSTR("$GPRMC"))
        // || !gp_comp(Buff, PSTR("$GPGSA"))
        // || !gp_comp(Buff, PSTR("$GPGLL"))
        // || !gp_comp(Buff, PSTR("$GPGSV"))
        // || !gp_comp(Buff, PSTR("$GPZDA"))
        // || !gp_comp(Buff, PSTR("$GPVTG"))
        ) {
        if (f_write(&pointFile, Buff, b, &s) || b != s){

            SEND_CMD(CLR_DISP);
            SEND_STR("KIRJOITUSVIRHE!"); // Mikäli kirjoitus epäonnistuu,
            return 0; // virheilmoitus ja palataan pääohjelmaan

        }
    }

    if (B5==0){ // Kun painetaan näppäintä, ohjelma sulkee tiedoston ja palaa
                // pääohjelmaan

        f_close(&pointFile);
        SEND_CMD(CLR_DISP);
        break;
    }
}

return 1;
}

```

```

/* Koordinaattien tallennus SD-kortille KML-formaatissa */
/*****/

int KML_on_sd(void){

    disk_initialize(0);          // Levyn ja tiedostojärjestelmän alustus
    f_mount(0, &fatfs);

    do{
        get_line();             // Odotetaan lausetta, josta saadaan poimitua päivämäärä

    }while(gp_comp(Buff, PSTR("$GPRMC")));

    p = gp_col(Buff,9);          // Avataan tiedosto nimellä "päiväys.kml"
    strcpy_P(&Buff[6], PSTR(".kml"));

    if (f_open(&pathFile, Buff, FA_OPEN_ALWAYS | FA_WRITE)
        || f_lseek(&pathFile, pathFile.fsize) ) {

        SEND_STR("AVAUSVIRHE!"); // Jos virhe, palataan pääohjelmaan
        delay_ms(50);
        return 0;
    }

    // Kirjoitetaan KML-formaatin vaatimat kuvaukset muistikortille

    f_printf(&pathFile, "<kml>\n");
    f_printf(&pathFile, "\t<Document>\n");
    f_printf(&pathFile, "<Placemark>\n");
    f_printf(&pathFile, "\t<name>Reittix</name>\n");
    f_printf(&pathFile, "\t<description>");
    f_printf(&pathFile, gp_col(Buff,9));
    f_printf(&pathFile, "\t</description>\n");
    f_printf(&pathFile, "\t<LineString>\n");
    f_printf(&pathFile, "\t\t<extrude>1</extrude>\n");
    f_printf(&pathFile, "\t\t<altitudeMode>absolute</altitudeMode>\n");
    f_printf(&pathFile, "\t\t<coordinates>");

    // Tulostetaan koordinaattitiedot kyseisestä reitistä.

    while (get_line()) {

        if (!gp_comp(Buff, PSTR("$GPGGA")))
        {
            Latitude = gp_col(Buff, 2); // Poimitaan paikkadata GGA-lauseesta
            Longitude = gp_col(Buff, 4);

            North = gp_col(Buff, 3);
            West = gp_col(Buff, 5);

            Alt = gp_col(Buff, 9);

            // Muutetaan koordinaatit desimaalimuotoon

            Longitud = strtod(Longitude, &endptr);
            minutes = modf(Longitud/100, &degrees);
            Longitud = degrees + (minutes*100)/60;

            Latitud = strtod(Latitude, &endptr);
            minutes = modf(Latitud/100, &degrees);
            Latitud = degrees + (minutes*100)/60;
        }
    }
}

```

```

if(North == 'S') Latitud = -Latitud;
if(West == 'W') Longitud = -Longitud;

sprintf(Latitude, "%f", Latitud);
sprintf(Longitude, "%f", Longitud);

altitude = strtod(Alt, &endptr);

sprintf(Alt, "%i", altitude);

// Tulostetaan koordinaatit keskelle KML-tiedostoa

f_printf(&pathFile, "%s,%s,%s,",
Longitude,
Latitude,
Alt);

f_sync(&pathFile); // Tiedoston synkronointi ja puskurin tyhjennys
f_truncate(&pathFile);
}

if (B5==0){ // Mikäli painetaan nappia, tallennus keskeytyy
SEND_CMD(CLR_DISP);
break;
}
}

// Tulostetaan KML-tiedoston vaatimat kuvaukset

f_printf(&pathFile, "</coordinates>\n");
f_printf(&pathFile, "\t</LineString>\n");
f_printf(&pathFile, "</Placemark>\n");
f_printf(&pathFile, "\t</Document>\n");
f_printf(&pathFile, "</kml>\n");

f_close(&pathFile); // Suljetaan tiedosto

SEND_CMD(CLR_DISP); // Tyhjennetään näyttö

return 1;
}

```

```

/* Pääohjelmasilmutta */
/*****/

int main(void)
{

    lolnit();    // IO-alustukset ja keskeytysten sallinta
    LCD_Ini();  // LCD-näytön alustukset

    SEND_CMD(DISP_ON); // Näytön käynnistys
    SEND_CMD(CLR_DISP); // Näytön tyhjäys

    SEND_STR("GPSlogger v.0.4 Jussi Saarijoki"); // Tervetulosanoma
    delay_ms(50);

    SEND_CMD(CLR_DISP); // Näytön tyhjäys

    while (1) {

        // Tulostetaan näytölle päävalikkoteksti ja aloitetaan näppäimistön pollaus ikuisessa silmu-
        kassa

        SEND_STR("-- Paavalikko --");

        // YLÖS
        if (B1==0) {

            SEND_CMD(CLR_DISP);
            SEND_CMD(DD_RAM_ADDR);
            SEND_STR("Koordinaatit nayttoon");
            delay_ms(50);
            NMEA_on_display();
        }

        // VASEN
        if (B2==0) {

            SEND_CMD(CLR_DISP);
            SEND_CMD(DD_RAM_ADDR);
            SEND_STR("NMEA-tallennus");
            delay_ms(50);
            NMEA_on_sd();
        }

        // OIKEA
        if (B4==0) {

            SEND_CMD(CLR_DISP);
            SEND_CMD(DD_RAM_ADDR);
            SEND_STR("KML-tallennus");
            delay_ms(50);
            KML_on_sd();
        }
    }
}

```