



jamk.fi

GPS-paikannin

Juhana Suhonen

Opinnäytetyö

Toukokuu 2016

Tekniikan ja liikenteen ala

Ohjelmistotekniikka

Jyväskylän ammattikorkeakoulu

JAMK University of Applied Sciences

Tekijä(t) Suhonen, Juhana	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2016
	Sivumäärä 69	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi GPS-paikannin		
Tutkinto-ohjelma Ohjelmistotekniikan koulutusohjelma		
Työn ohjaaja(t) Kotkansalo Jouko		
Toimeksiantaja(t)		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli luoda kevyt, käsikäyttöinen ja energiatehokas GPS-paikannin harrasteprojektina. GPS-paikannin vastaanottaa GPS-satelliittien radioaallot ja näyttää lasketun paikkatiedon käyttäjälle laitteen omalla LCD-näytöllä tai USB-kaapelin kautta kytkettynä tietokoneeseen kolmansien osapuolien tarjoamilla GPS-ohjelmistoilla. GPS-paikantimeen on toteutettu automaattinen virrankatkaisujärjestelmä, sekä käyttäjäpainikkeet käyttöliittymää varten.</p> <p>Opinnäytetyö sisältää laajan kuvauksen GPS-paikantimen kehitysprosessista, mukaan lukien kehitystyön alussa tehdyt komponenttivalinnat ja loppuvaiheen piirilevy suunnittelu. Toteutus aloitettiin etsimällä sopivat kehitysalustat ja ohjelmistotyökalut testikytkentää varten. Onnistuneen testikytkennän jälkeen opinnäytetyössä käsitellään piirilevyn suunnittelun vaiheet kytkentäkaaviosta valmiiseen piirilevyyn asti. Opinnäytetyössä lopuksi käsitellään ohjelmistokehitystä ja testausta valmistuneelle GPS-paikantimelle.</p> <p>Lopputuloksena saatiin toimiva akkukäyttöinen GPS-paikannin hyödyntäen avoimen lähdekoodin kehitystyökaluja sulautettuihin järjestelmiin. Lopputuote toimii harrasteprojekteissa valmiina kehitysalustana ominaisuuksien laajentamiselle ja muihin GPS-paikannukseen perustuviin sulautettuihin järjestelmiin.</p>		
Avainsanat (asiasanat) Sulautetut järjestelmät, Arduino, GPS, C, NMEA		
Muut tiedot		

Author(s) Suhonen, Juhana	Type of publication Bachelor's thesis	Date May 2016 Language of publication: Finnish
	Number of pages 69	Permission for web publication: x
Title of publication Handheld GPS device		
Degree programme Software Engineering		
Supervisor(s) Kotkansalo Jouko		
Assigned by		
Abstract <p>The goal of this thesis was to create a lightweight, handheld and energy-efficient GPS receiver as a hobby project. The GPS receives radio waves from satellites and displays the calculated location information on the GPS receiver's screen or via USB cable on a computer with third party software. The GPS receiver has been integrated with an automatic shutdown system and buttons for user interaction.</p> <p>The thesis includes an extensive description of the development process for the GPS receiver. The implementation began with the search for suitable development platforms and software tools to test the overall connection. After a successful test of the connection the thesis introduces basic steps to create a working printed circuit board for the GPS receiver. The printed circuit board was designed based on the wiring diagram from the test connections. Finally, the thesis deals with software development and testing for hardware ready GPS receiver.</p> <p>The end result was a functional battery-powered GPS receiver making use of open source development tools for embedded systems. The final product is ready for further feature expansions and other embedded system projects in the future.</p>		
Keywords/tags (subjectshttp://vesa.lib.helsinki.fi/) Embedded systems, Arduino, GPS, C, NMEA		
Miscellaneous		

Sisältö

Sanasto	6
1 Työn lähtökohdat	12
2 Kuinka GPS toimii	12
3 Työn aloitus	13
3.1 Tavoitteet	13
3.2 Toteutuksen hahmotelma	13
4 Ensimmäinen kehitysversio	14
4.1 Kehitysalustan valinta	14
4.2 GPS-sirun valinta	16
4.3 LCD-näytön valinta	18
4.4 Muiden komponenttien valinta	20
4.5 Kokonaisuus	22
4.6 Firmwaren kehitys	23
4.7 Arduino IDE	24
4.8 Kirjastot	25
4.8.1 GPS-kirjastot	25
4.8.2 LCD-kirjastot	26
5 Komponenttien käyttöönotto ja testit	26
5.1 GPS kokeilu	26
5.2 GPS integrointi	27
5.3 LCD:n kokeilu	28
5.4 Kokonaisuuden integrointi	29
5.5 Testaus ja tulokset	30
5.5.1 Ongelman selvitys	30
5.6 Vian korjaaminen	30
6 Toinen kehitysversio	31
6.1 Uuden alustan valinta	31

	2
6.2	Uuden alustan käyttöönotto ja testaus.....33
6.3	Akkuvahti.....35
6.3.1	Akkuvahdin testaus.....36
6.3.2	ADC-ongelma ja selvitystyö36
6.4	Virtakytkimen toteutus37
6.4.1	Automaattinen sammutus.....38
6.5	Käyttäjäpainikkeet.....38
6.5.1	Bounce-efekti.....40
6.5.2	Ohjelmatoteutus.....42
6.5.3	Painikkeiden testaus.....42
6.5.4	Ohjelmallinen debounce korjaus.....43
7	Piirilevyn kehitysprosessi43
7.1	CadSoft Eagle.....44
7.2	KytKentäkaavio44
7.2.1	Kehitysalustan kytKentäkaavio44
7.2.2	Muutokset kehitysalustan kytKentään44
7.2.3	LCD-näytön kytKentä45
7.2.4	GPS-sirun kytKentä46
7.2.5	Virtakytkimen kytKentä47
7.2.6	Painikkeiden kytKentä.....48
7.2.7	Vikojen tarkistus49
7.3	BOM.....49
7.4	Levyn suunnittelu50
7.4.1	Kerrokset ja rakenne.....50
7.4.2	Valmistajan etsiminen50
7.4.3	Komponenttien asettelu.....51
7.4.4	LCD-näytön vaatimat muutokset levyllä53

7.4.5	DRC	54
7.4.6	Silkscreen	54
7.5	Valmistajan vaihtaminen.....	55
7.5.1	CAD-kuvien muutokset uudelle valmistajalle.....	55
8	Prototyypin kasaaminen	56
8.1	Kasausprosessi.....	56
8.2	Firmwaren kirjoitus	57
8.3	Ensimmäinen testaus	58
9	Ominaisuuksien laajentaminen.....	59
9.1	Käyttöliittymä	60
9.2	Näytettävät paikannustiedot	60
9.3	Laajennettujen ominaisuuksien testaaminen	60
10	Tulevaisuuden optiot	62
11	Arvio lopputuloksesta	62
	Lähteet.....	64
	Liitteet	66
	Liite 1. Kytkenäkaavio	66
	Liite 2. BOM	67

Kuviot

Kuvio 1. Arduino Pro Mini ja FTDI-piiri.....	16
Kuvio 2. Adafruit Ultimate GPS Breakout.....	18
Kuvio 3. Nokia 5110:n LCD-näyttö LPH7366	20
Kuvio 4. On/Off kytkin	22
Kuvio 5. Ensimmäinen hahmotelma GPS-paikantimesta.....	23
Kuvio 6. Arduino IDE aloitusnäky.....	25
Kuvio 7. Lohkokaavio ensimmäisestä GPS-piirin testauksesta	27
Kuvio 8. Lohkokaavio GPS-paikantimen kytkennästä Arduino Pro Minin kanssa.....	28
Kuvio 9. Lohkokaavio LCD-näytön kytkennästä Arduino Pro Minin kanssa.....	28
Kuvio 10. Lohkokaavio GPS-piirin ja LCD-näytön yhteistoiminnasta Arduino Pro Minin kanssa	29
Kuvio 11. Sparkfun Fio V3.....	32
Kuvio 12. Lohkokaavio Sparkfun Fio V3 koekytkennästä.....	33
Kuvio 13. Fio V3:n alustatestaus	34
Kuvio 14. Lohkokaavio akun kytkennästä nykyiseen toteutukseen.....	35
Kuvio 15. Akun jännitteen näyttäminen näytöllä	37
Kuvio 16. Lohkokaavio virtakytkimen liittamisestä kytkentään.....	38
Kuvio 17. Lohkokaavio painonappien liittamisestä kokonaiskytkentään	39
Kuvio 18. Napin vapautuessa tapahtunut bounce-efekti	40
Kuvio 19. Mittauksissa ollut nappi ei antanut painettaessa bounce-efektiä.....	41
Kuvio 20. Painikkeen toteutuksen testaus	43
Kuvio 21. Uusi hahmotelma kokonaistoteutuksesta	45
Kuvio 22. LCD-näytön kytkentäkaavio lopullisella levyllä	46
Kuvio 23. GPS-piirin kytkentä lopullisella levyllä.....	47
Kuvio 24. Virtakytkimen kytkentä lopullisella levyllä.....	48
Kuvio 25. Painonappien kytkentä lopullisella levyllä.	49
Kuvio 26. Komponenttiasettelu piirilevyn päällä	52
Kuvio 27. Komponenttiasettelu piirilevyn alapuolella	53
Kuvio 28. Yksi LCD-näytön kiinnikkeitä varten tehty muutos	54
Kuvio 29. Piirilevyn yläreunan silkscreen muutokset.....	55

Kuvio 30. Valmistajan antama huomio piirilevyn Gerber-tiedostoissa.	56
Kuvio 31. Brandnerin toimittama valmis piirilevy ENIG-pinnoitteella	57
Kuvio 32. Testausfirmware ladattu onnistuneesti GPS-paikantimeen	59
Kuvio 33. Päivitetty käyttöliittymä GPS-paikantimella	61
Kuvio 34. VisualGPSView testaus GPS-paikantimen kanssa	62

Taulukot

Taulukko 1. Vaihtoehtoiset kehitysalustat	15
Taulukko 2. Vaihtoehtoiset GPS-alustat	17
Taulukko 3. Vaihtoehtoiset LCD-näytöt	19
Taulukko 4. Lista edullisista piirilevynvalmistajista	51

Sanasto

ADC

ADC (Analog to Digital Converter) on lyhenne laitteesta, joka muuttaa analogisen jännitetason digitaaliseen muotoon. Digitaalinen muoto on yleensä binäärinen arvo AD-muuntimen tarkkuudella. Esimerkiksi 10-bittiä.

Arduino

Arduino italialaista alkuperää oleva avoimen lähdekoodin kehitysympäristö sulautetuille järjestelmille. Arduinoa tukevat sulautetut järjestelmät ovat pääsääntöisesti AVR:än valmistamia 8-bittisiä mikrokontrollereita. Mukaan on myös viime aikoina tullut ARM pohjaisia 32-bittisiä kehitysalustoja, sekä intellin x86 arkkitehtuuriin perustuva Edison alusta.

AT

AT-komennot ovat yleinen kommunikointiprotokolla sarjaliikenteen ylitse modeemia tai vastaavaa edustavalle laitteelle. AT-komennoilla voidaan käskyttää, muuttaa tai lukea parametrejä päätelaitteelta. AT tulee englannin kielisestä sanasta "Attention" suomennettuna "huomio". Tunnetaan myös nimellä Hayes command set

AVR

AVR on Atmel valmistajan 8-bittisten mikrokontrollereiden tuoteperhe. AVR tuoteperhe jaetaan kolmeen päähaaraan pohjautuen muistin ja ominaisuuksien määrään: ATiny, AT90x ja ATmega.

BOM

BOM (Bill Of Materials) on luettelo kaikista elektronisen kytkennän osista. BOM yleensä sisältää komponentin kotelotyyppin, arvon ja referenssinimen kytkennässä. Esimerkiksi: R1, 4,7 k Ω , 0603. Tarkemmat BOM:it sisältävät myös komponentin valmistajan, tilauskoodin ja hinnan.

COM

COM (Communication port) on alkuperäinen nimi sarjaliikenneportille. COM-portteja voidaan toteuttaa virtuaalisena USB:n tai Bluetoothin ylitse. Käytetään hyvin paljon kommunikointiin tietokoneen ja sulautetun järjestelmän kanssa.

DRC

DRC (Design Rule Checking) on sääntötarkastus piirilevyvalmistajan tuotantokyvyyden mukaan. DRC pitää sisällään esimerkiksi pienimmät poranhalkaisijat, johdinleveydet ja eristevälit.

EEPROM

EEPROM (Electrically Erasable Programmable Read-Only Memory) on haihtumaton muistityyppi, joka on yleensä osana mikrokontrolleria. EEPROM:lle voidaan tallentaa muuttujia, jotka halutaan säilyvän myös virrankatkaisun yli.

ENIG

ENIG (Electroless Nickel Immersion Gold) on lyijytön pinnoitemateriaali piirilevyjen johtimilla ja kontakteilla. ENIG on uudemman tyyppinen pinnoitemateriaali ja monin tavoin parempi, kuin HASL, mutta kalliimpi.

Firmware

Firmware on mikrokontrollerin FLASH-muistille kirjoitettava ohjelman suorituskoodi, jonka mukaan mikrokontrolleri toimii.

FTDI

FTDI (Future Technologies Devices International) on Skotlantilainen yritys, joka on erikoistunut USB-RS232/TTL sarjaliikennemuunnin piireihin. Yleensä FTDI-piirillä viitataan USB to TTL-serial piireihin.

Gerber

Gerber on ASCII pohjainen 2-ulotteinen vektori tiedosto, johon on tallennettu piirilevyn eri kerrokset tuotantoa varten. Esimerkiksi: Ylin johdin kerros, Juotteenestopinnoite ja Silkscreen.

GPIO

GPIO (General Purpose Input Output) on portti mikrokontrollerissa tai mikroprosessorissa, joka voidaan ohjelmoida joko loogiseksi sisääntuloksi tai ulostuloksi. Portilla voidaan ohjata oheislaitteita tai lukea dataa oheislaitteilta.

GPS

GPS (Global Positioning System) on satelliittipohjainen paikannusjärjestelmä, jonka on rahoittanut Yhdysvaltojen puolustusministeriö. Sen kehitys alkoi 1970-luvulla.

HASL

HASL (Hot Air Solder Leveling) on pinnoitemateriaali piirilevyn johtimilla ja kontakteilla. HASL:ia on saatavilla lyijyllisenä sekä lyijyttömänä. HASL ei sovellu kaikista pienimmille johdinleveyksille.

IDE

IDE (Integrated Development Environment) tarkoittaa kehitysympäristöä, joka yleensä sisältää ohjelmointityökalut, debuggaustyökalut sekä käännöstyökalut. Esimerkkinä Arduino IDE.

ISP

ISP (In System Programmer) on mikrokontrollerin uudelleenohjelmointityökalu, jolla voidaan uudelleen kirjoittaa mikrokontrollerin firmware.

LCD

LCD (Liquid Crystal Display) eli nestekidenäyttö. LCD perustuu polarisoivaan filteriin ja nesteeseen, joka muuttaa järjestystään polarisaation suuntaiseksi sähkökentässä.

LED

LED (Light Emitting Diode) on diodi, joka emittoi valoa, kun diodin kynnysjännite ylitetään ja se alkaa johtamaan virtaa. LED:it ovat nykyään tulleet suureen suosioon valaistuksessa, niiden pienen virrankulutuksen ansiosta.

Li-Po

Li-Po (Lithium-Polymer) on akkutyyppi, joka on tunnettu keveytensä ja energiatiheydensä ansiosta. Li-Po-akut ovat hyvin tarkkoja latausvirrastaan, eikä niitä saa ladata väärin.

MCU

MCU (Micro Controller Unit) eli mikrokontrolleri. Mikrokontrolleri on pieni ohjelmoitava piiri, joka sisältää erinäisiä integroituja kytkentöjä, järjestelmiä ja muisteja.

MOSFET

MOSFET (Metal-Oxide-Semiconductor Field-effect Transistor) on transistorityyppi, joka on jänniteohjattava. MOSFET:ejä on P- ja N-kanavaisia.

NAND

NAND (Negative-AND) on looginen piiri, jolla on sama totuustaulu kuin negatiivisella ja logiikalla. Piiri palauttaa loogisen 1:sen kun kanavista A tai B ei kumpikaan ole tosi tai vain toinen on tosi.

NMEA

NMEA 0183 (National Marine Electronics Association) on spesifikaatio paikannusdalle sarjaliikenteessä. NMEA-lause perustuu ASCII-merkistöön ja alkaa aina "\$" merkillä. Seuraavat viisi merkkiä kuvaavat lauseen sisällön. Loput merkeistä on paikannusdataa kuvauksen mukaan.

QFN

QFN (Quad Flat No-Leads) on mikropiirin kotelotyyppi. QFN nimikkeen perässä on aina lukema, joka kuvaa kontaktien määrää piirillä. Esimerkiksi QFN-44.

RAM

RAM (Random Access Memory) on nopea ja haihtuva muistityyppi, johon voidaan tallentaa väliaikaista tietoa.

Silkscreen

Silkscreen on piirilevyn pinnalla oleva kuvituskerros, johon voidaan merkata tietoja levystä. Tiedot voivat esimerkiksi olla komponenttien ulkoreunat, komponenttien arvot, komponenttien nimet, levyn nimi, levyn versio ja yrityksen logo. Silkscreen on yleensä valkoinen.

SMD

SMD (Surface Mount Device) on aktiivi- tai passiivikomponentti, joka juotetaan suoraan piirilevyn pintaan. SMD-komponentit ovat korvanneet suurimmaksi osaksi läpiladottavat komponentit kokonsa ja tehokkaan kasaustekniikan ansiosta.

Soldermask

Soldermask eli juotteenestopinnoite. Suomeksi lyhennettynä JEP. Juotteenestopinnoitteen ideana on estää piirilevyn juotoskontaktin pinnalla olevan juotostinan tai juotospastan leviäminen ympäri piirilevyä juotosoperaatiossa. Toimii myös osittain eristeenä.

SOT

SOT (Small-Outline Transistor) on puolijohteen kotelointimalli. SOT nimikkeen perässä on aina lukema, joka kuvaa komponentin jalkajärjestyksen ja asettelun. Esimerkiksi SOT23-3. SOT Koteloituja komponentteja ovat yleensä transistorit, diodit ja regulaattorit.

SPI

SPI (Serial Peripheral Interface) on sarjaliikenneväylä, johon voi kytkeä rinnan useamman SPI protokollaa käyttävän laitteen. SPI kommunikaatiossa on yksi käskyttävä laite ja vähintään yksi tai useampi orjalaite, joka valitaan ohjaukseen erillisellä I/O linjalla ajamalla se maatasoon. Linjoihin kuuluu myös kello, käskyttävän laitteen sarjaliikenne, orjalaitteen sarjaliikenne sekä reset linja.

TQFP

TQFP (Thin Quad Flat Package) on mikropiirin kotelotyyppi. TQFP nimikkeen perässä on aina lukema, joka kuvaa kontaktien määrää piirillä. Esimerkiksi TQFP-44

UART

UART (Universal Asynchronous Receiver Transmitter) on sarjaliikennepiiri, joka on yleensä integroitu osa mikrokontrolleria tai muuta mikropiiriä. Sen avulla voidaan kommunikoida, joko tietokoneen kanssa tai toisen mikropiirin kanssa.

USB

USB (Universal Serial Bus) on sarjaliikenneväylä tietokoneen ja sen oheislaitteiden välillä.

VIA

VIA (Vertical Interconnect Access) on kontakti, joka yhdistää piirilevyllä johtimen eri kerrosten välillä.

1 Työn lähtökohdat

GPS alunperin kehitettiin Yhdysvaltojen sotilaskäyttöön, mutta viime vuosina se on päätenyt vahvasti myös siviilikäyttöön. GPS-paikannus on lisääntynyt nykyaikana huomattavissa määrin. Älypuhelimet ja aktiivisuutta seuraavat urheilulisälaitteet keräävät nykyään paikkatietoa, josta käyttäjät voivat tarkkailla toimintaansa tai saada lisäpalveluja.

Tämän opinnäytetyön tarkoituksena on kehittää toimiva prototyyppi GPS-vastaanottimesta. GPS-vastaanotin itsessään on sulautettu järjestelmä, joka sisältää joukon erilaisia piirejä jotka on saatava toimimaan kokonaisuutena. Vastaavan prototyypin luomiseen tarvitaan niin ohjelmistokehitystä kuin elektroniikan suunnittelua. Toteutuksessa ohjelmointi sekä elektroninen suunnittelu tulee kulkemaan hyvin vahvasti rinnakkain pitkin kehitysprosessia.

2 Kuinka GPS toimii

GPS perustuu vähintään 24:ään maata kiertävään satelliittiin, joista kukin sisältää hyvin tarkan atomikellon. Nämä satelliitit kiertävät maata 20350 km korkeudella maanpinnasta. Jokainen satelliiteista kiertää maan kahdesti vuorokauden aikana, jotta saavutetaan koko maan kattava peittävyys. Satelliitit kiertävät maata kiertoradoillaan siten, että käytännössä puolet niistä ovat tasaisesti toisella puolella maapalloa.

Jokainen satelliitti lähettää radiosignaalillaan oman atomikellonsa ajan sekä satelliitin oman fyysisen sijainnin kolmiulotteisessa avaruudessa. Tämä signaali saavuttaa GPS-vastaanottimen valonnopeudella, joka on vakio. GPS-vastaanotin vertaa satelliitista saatua atomikellon aikaa ja vertaa sitä omaan sisäiseen kelloonsa. Vertailun avulla saadaan laskettua satelliitin etäisyys vastaanottimesta. Koska GPS-vastaanottimen kello on paljon epätarkempi kuin satelliitin atomikello, joudutaan radiosignaali vastaanottamaan vähintään neljästä GPS-satelliitista, jotta voidaan laskea tarkka kolmiulotteinen sijainti. Sijaintitarkkuus kasvaa, mitä enemmän satelliitteja GPS-vastaanotin havaitsee. (GPS Educational Poster 2013.)

3 Työn aloitus

3.1 Tavoitteet

Ilman tavoitteita ei voi saavuttaa selkeää lopputulosta. Työtä aloittaessa tulee tehdä selväksi, mitä halutaan. Kun tiedetään, mitä halutaan, tulee miettiä, että kuinka tavoitteet voidaan toteuttaa. Projekti itsessään tulee vaatimaan monta vaihetta, joten tavoitteet tulee pitää realistisina.

Yksi tavoitteista oli tehdä GPS-paikantimesta mahdollisimman energiatehokas. Hyvä käyttöaika parantaa tuotteen käyttömukavuutta. Energiatehokkuutta voi parantaa GPS-paikantimen automaattisella sammuttamisella, jos laite on jäänyt käyttämättömäksi.

Pää tavoitteista oli tietenkin saada oikea paikannusdata näkymään näytöllä. Paikannusdatan lisäksi tavoitteena oli GPS-paikantimessa nopeuden sekä mahdollisten kuu- luvien satelliittien lukumäärän näyttäminen. GPS-piiriltä tulevien NMEA-lauseiden ohjaus tietokoneelle laajentaisi tuotteen käyttökohteita huomattavasti, minkä takia sekin toimi yhtenä tavoitteista.

3.2 Toteutuksen hahmotelma

Työ alkaa sopivien komponenttien valinnoilla. Kaikki rakentuu oikean kehitysalustan ympärille. Tämä on yksi tärkeimmistä komponenteista, koska tämä osio tulee olemaan se, joka toimii kaiken aivoina. Tämän ympärille tulee valita sopiva näyttö, GPS-siru, virransyöttö ja oheiskomponentit. Jokainen näistä ratkaisee huomattavasti mitä ja miten MCU ohjelmoidaan kehitysalustalla. Komponentit implementoidaan ja testataan kokonaisuudessa toimivuuden takia. Kun kokonaisytkentä on kasassa ja testattu ohjelmallisesti voidaan aloittaa prototyypin valmistaminen. Kokonaisytkentä hahmotellaan piirilevyn suunnitteluohjelmistolla, josta se sitten mallinnetaan piirilevyn CAD-kuviksi. Nämä CAD-kuvat lähetetään piirilevyn valmistajalle, joka toimittaa mittatilatut piirilevyt prototyyppiin. Kun piirilevylle on juotettu komponentit ja testattu toimivaksi, voidaan alustalle kehittää lisää ohjelmapohjaista toiminnallisuutta.

4 Ensimmäinen kehitysversio

4.1 Kehitysalustan valinta

Kehitysalustassa ratkaisevin valinta on siinä oleva MCU. Kehitysalustoja on markkinoilla hyvin kattava määrä, mutta valtaosassa käytetään vain tiettyjä MCU:ita. Tämä on erittäin keskeistä varsinkin 8-bittisten MCU:ita sisältävien kehitysalustojen kanssa. MCU valikoima on paljon laajempi 32-bittisten kehitysalustojen kanssa. Yksinkertainen GPS-prototyyppi ei kuitenkaan tarvitse 32-bittistä laskentatehoa. Tästä syystä pyrin pitämään alustavalikoiman 8-bittisissä.

8-bittisissä kehitysalustoissa valikoimaa erityisesti edustaa Atmelin AVR MCU valikoima. Tunnetuimpia alustoja ovat Arduinon omat kehittämät alustat. Yleisimmät 8-bittiset MCU:t Arduinon alustoissa ovat Atmega 328, Atmega 168, Atmega 32u4 sekä Atmega 2560. (Arduino Products n.d.) Koska Arduino perustuu avoimeen lähdekoodiin, ovat muut valmistajat tehneet arduinon yhteensopivia kehitysalustoja. Nämä ulkopuoliset kehitysalustat poikkeavat arduinon omista alustoista pääsääntöisesti alustan lisälaitteilla ja komponenteilla.

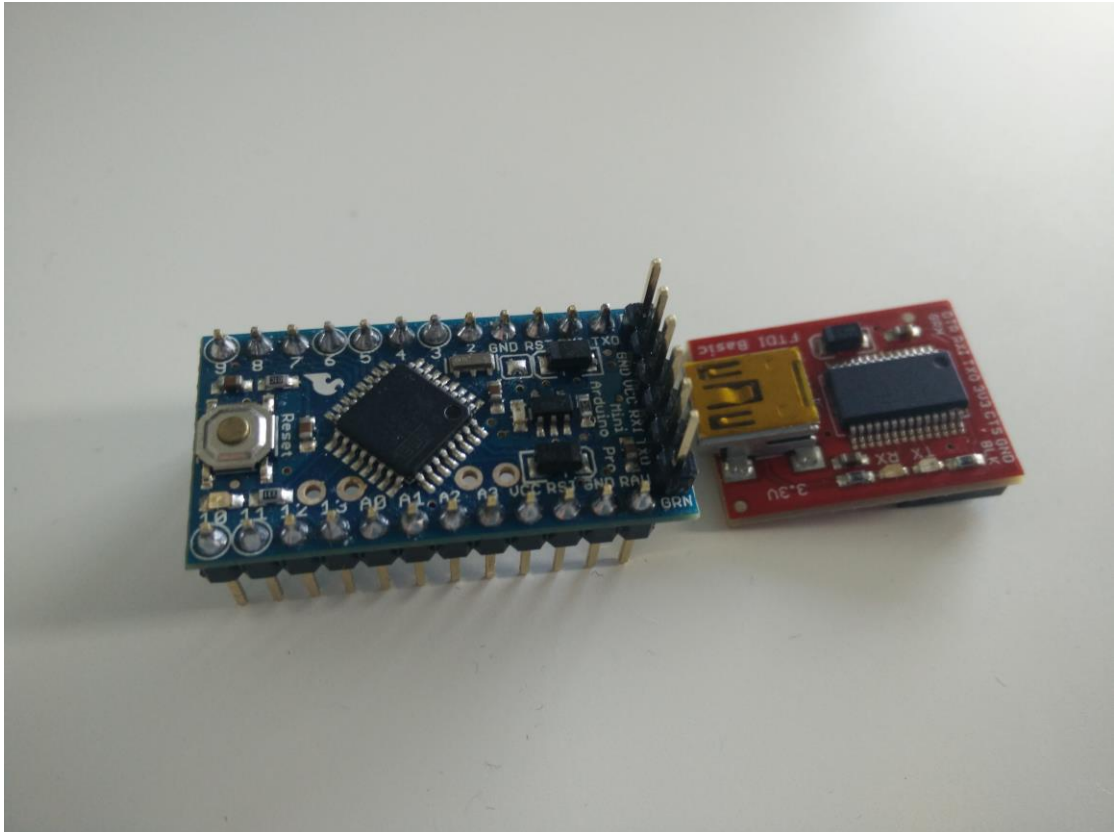
Taulukkoon 1 on listattu vaihtoehtoisia kehitysalustoja projektiin. Alustan tulisi mahdollistaa vaatimusten mukainen toiminta. Tärkeimpänä huomiona matala käyttöjännite mobiiliin ratkaisuun sekä riittävästi GPIO-portteja.

Taulukko 1. Vaihtoehtoiset kehitysalustat

Kehitysalusta	Plussat	Miinukset	Muuta
Arduino Nano (328)	-Integroitu FTDI piiri -ISP kytkentä -Pieni	-Korkeahko käyttöjännite	
Arduino Micro (32u4)	-Natiivi USB tuki -ISP kytkentä -Pieni -Enemmän ulkoisia interrupt portteja	-Korkeahko käyttöjännite -Kallis MCU	Ei FTDI piirin tarvetta.
Arduino Pro Mini (328)	-Matalan käyttöjännitteen versio saatavilla -Vähän oheiskomponentteja -Pienin	-Ei ISP kytkentää	Vaatii ulkoisen FTDI piirin hankinnan

Arduino Nanon sekä -Micron tapauksessa tuli erityisesti ottaa huomioon, että korkea 5 V käyttöjännite ei ole optimaalinen mobiiliin käyttöön. Näiden alustojen käyttö ei onnistu pelkästään 2-3:lla sormiparistolla tai Li-Po-akulla. Nanon ja Micron tapauksessa tulisi rakentaa step-up jännitemuunnin ja se lisäisi kytkennän monimutkaisuutta. Arduino Micro tarjoaisi natiivin USB-tuen, jolloin firmwaren ohjelmointi ei vaatisi FTDI-piirin hankintaa, mutta on itsessään kallis MCU hankittavaksi. Arduino Pro Mini on alustoista karsituin. Se kuitenkin tarjoaisi mahdollisuuden käyttää 3,3 V käyttöjännitettä, jolloin Li-Po-akun tai kolmen sormipariston käyttö onnistuisi.

Näistä kolmesta optimaalisin ratkaisu oli Arduino Pro Mini. (Ks. Kuvio 1.) Syynä valintaan oli matala käyttöjännite ja sen tuomat edut virransyötöön. Lisäksi alustalla oleva MCU tarjosi tarvittavan määrän muistia ja GPIO-portteja.



Kuvio 1. Arduino Pro Mini ja FTDI-piiri

4.2 GPS-sirun valinta

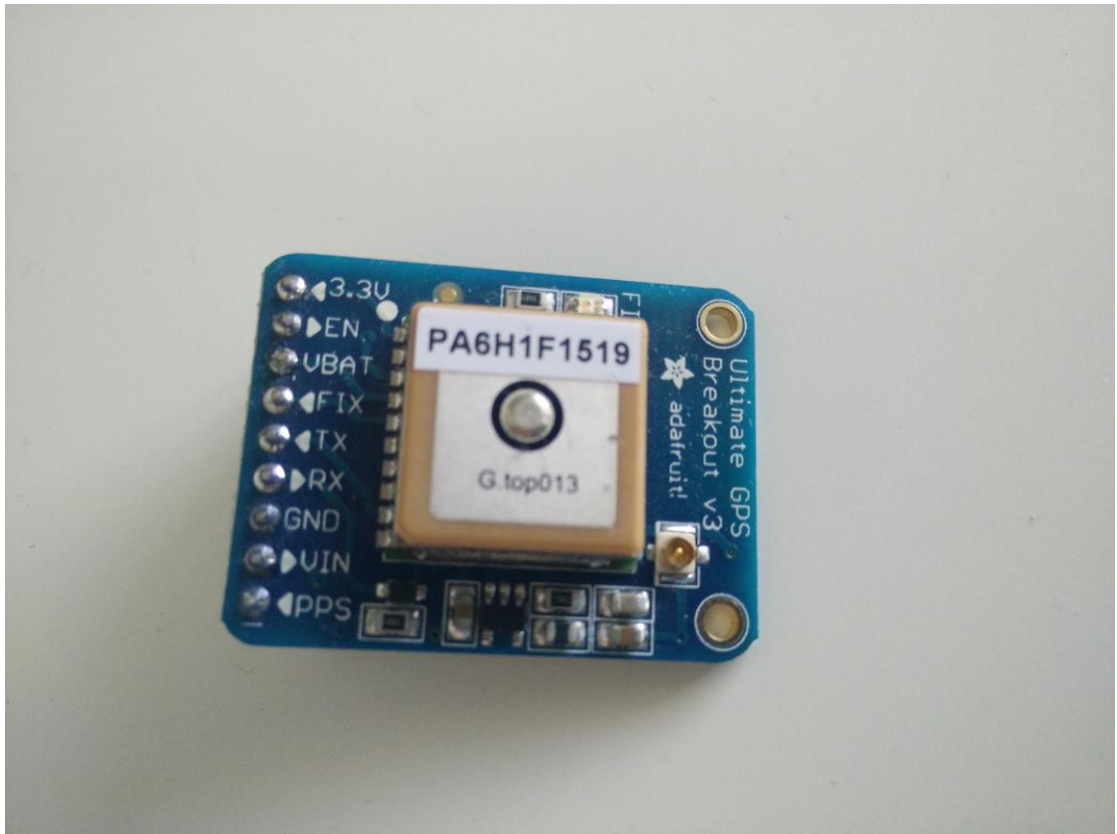
Koko prototyypin toiminta tuli pohjautumaan GPS-piiriin, joka lähettää NMEA-paikkatietoa MCU:lle. GPS-piirejä ei ollut markkinoilla niin paljoa, mitä kehitysalustoja oli. Piiriä valittaessa tuli ottaa huomioon myös se, että siitä on saatava valmis referenssikytkenä levyllä. Tämä rajasi huomattavasti valinnanvaraa. Referenssikytkenä pitää testata kehitysalustan kanssa ennen kuin toteutetaan lopullinen kokonaisuus. Taulukkoon 2 on listattu vaihtoehtoisia GPS-alustoja projektiin.

Taulukko 2. Vaihtoehtoiset GPS-alustat

GPS	Plussat	Miinukset	Muuta
Ublox Neo6M	-Halvin -Helppo juottaa	-Referenssilevyn huono dokumentaatio	-Ulkoinen antenni
Sparkfun Venus (Venus638FLPx)	-Tehokkain -Dataloggeri integroitu	-Kallein -Suurin virrankulutus -Piirin kotelo hankala juottaa käsin.	-Ulkoinen antenni -Sparkfun tutoriaali
Adafruit Ultimate GPS Breakout (MTK3339)	-Integroitu antenni -Dataloggeri integroitu -Helppo juottaa.	-Kallis	-Adafruit tutoriaali

Ublox on GPS-piirien valmistajana listalta tunnetuin. Kuitenkin harrastajille saatavia valmiita alustoja oli niukasti tarjolla. Kyseinen Ublox Neo6M piiriin perustuva levy löytyi ebaysta hyvin rajatulla dokumentaatiolla. Sparkfunin Venus GPS-piiriin perustuva alusta oli vaihtoehtoista ehdottomasti tehokkain. Kuitenkin piiri kuluttaa huomattavasti enemmän virtaa kuin muut kilpailijansa. Tämä syö toiminta-aikaa radikaalisti. Sparkfunin alustasta löytyi kuitenkin hyvin dokumentaatiota, mikä auttaisi lopullisen prototyypin kytkennän ja ohjelman tekemisessä. Adafruitin alusta oli melkein samalla tasolla paikannusherkkyyden kanssa kuin Sparkfunin alusta, mutta alle kolmasosan virrankulutuksella. Lisäksi Adafruitin GPS-piiriin oli integroitu oma antenni. Tämä helpottaisi lopullista suunnittelua. GPS-irtopiiri oli Adafruitin ratkaisussa yllättävän kallis, mutta sen ympärille on helppo tehdä lopullinen toteutus. Adafruit myös tarjosi omalle levylleen erittäin kattavan dokumentaation sekä piiriä varten erityisesti suunnitellun Arduino kirjaston.

Näistä vaihtoehtoista päätin valita Adafruitin ratkaisun. (Ks. Kuvio 2.) Suurimpina syinä oli juurikin loistava dokumentaatio sekä integroitu antenni. Levy sisältää oman regulaattorin, jota en kytkennässä tarvinnut, mutta sen voi kuitenkin ohittaa.



Kuvio 2. Adafruit Ultimate GPS Breakout

4.3 LCD-näytön valinta

Näytön valitseminen ratkaisee, millainen käyttäjäkokemus saavutetaan prototyypille. Näyttöjä oli paljon suurempi valikoima kuin GPS-piirejä, mutta erot näyttöjen välillä ovat huomattavat. Kaikki tavoitteiden mukaiset ominaisuudet tuli näkyä ja mahdollista toteuttaa käyttäjäystävällisesti. Värinäytöt karsin valinnoista pois, koska niiden ohjaaminen vie hyvin paljon MCU:n RAM-muistia. Taulukkoon 3 on listattu vaihtoehtoisia LCD-näyttöjä.

Taulukko 3. Vaihtoehtoiset LCD-näytöt

LCD	Plussat	Miinukset	Muuta
Generic 8x2 ASCII LCD	-Yksinkertainen -Paljon tuettuja kirjastoja.	-Näyttää vain tekstiä -Ei välttämättä käyttäjäystävällinen	
NOKIA 5110 LCD (LPH-7366)	-Pieni -Halpa -Paljon tuettuja kirjastoja.	-Pieni resoluutio	
Generic 128x64 Graphic LCD	-Iso pinta-ala -Iso resoluutio -Paljon tuettuja kirjastoja	-Kallis	

8x2 ASCII merkinäyttö on ratkaisuna hyvin geneerinen ja yksinkertainen. Kyseinen ratkaisu ei kuitenkaan mahdollista kovin kaunista käyttöliittymää. Ominaisuuksien laajentaminen tulevaisuudessa voi karsiutua, jos olisin päätenyt kyseiseen ratkaisuun. Nokian vanha 5110 puhelimen näyttö on graafinen ja mahdollistaa täten erilaisten kuvioiden piirtämisen näytölle. Graafisuuden avulla uusien ominaisuuksien jatkokehitys olisi luontevampaa. Kyseinen näyttö oli myös edullinen. Isompi 128x64:n pikselin näyttö on jo reilusti isompi kuin Nokian 5110:n LCD-näyttö. Se oli kuitenkin hyvin kallis ja suurikokoinen.

Vaihtoehtoista helpoksi valinnaksi jäi Nokia 5110:n näyttö. Se kattaa perus graafiset ominaisuudet sekä on edullinen ratkaisu. Näyttö on kooltaan pieni, joten sillä voidaan saavuttaa fyysisiltä mitoiltaan pieni GPS-paikannin. (Ks. kuvio 3)



Kuvio 3. Nokia 5110:n LCD-näyttö LPH7366

4.4 Muiden komponenttien valinta

MCU:n, GPS-piirin ja LCD-näytön lisäksi tarvitsin muita komponentteja lopulliseen prototyyppiin. Suurimmat puuttuvat osat ovat virransyöttö sekä painonapit. Virransyötön vaihtoehtoina ovat Li-Po-akku tai sormiparistot. Sormiparistot olisivat virransyötön kannalta helpoin ratkaisu, mutta ne veisivät huomattavasti enemmän tilaa kuin ohut Li-Po-akkukenno. Li-Po-akku kuitenkin vaatii oman laturipiirin, jos toteutuksen haluaa tehdä fiksusti.

Valitsin vaihtoehdoista Li-Po-akkuversion, koska sormiparistoilla kokonaisuus voi tulla liian painavaksi ja kookkaaksi. Akkukäyttöisiä GPS-paikantimia on markkinoilla vähemmän kuin paristokäyttöisiä, joten projektissa päätin kokeilla valtavirrasta poikkeavaa ratkaisua. Valinnan takia joudun valitsemaan sopivan laturipiirin Li-Po-akuille. Laturipiirin valinnan pohjasin vahvasti EEVBlogin Li-Po-latauksen oppaaseen. (EE-

Vblog #176 - Lithium Ion/Polymer Battery Charging Tutorial 2011.) Oppaassa käydään läpi MCP73831 latauspiiri Microchipiltä, joka on täysin riittävä lataamaan GPS-paikantimen Li-Po-akku.

Pelkkä Li-Po-akku ei ole toimiva ratkaisu antamaan vakiojännitettä MCU:lle ja oheiskomponenteille. Tätä varten tarvitsin regulaattorin, joka tiputtaisi Li-Po-akun jännitteen alaspäin kiinteäksi 3,3 V:ksi. Arduino Pro Mini kehitysalustalla oli valmiina MIC5205 3,3 V regulaattori juotettuna. Kyseinen regulaattori kykenee 150mA virtaan. (Micrel MIC5205 2006.) Jos kyseinen regulaattori kestää kokonaiskytkennän kuormituksen, niin sen voi ottaa mukaan lopulliseen prototyyppiin.

Akun ja regulaattorin väliin piti tehdä virtakytkin, muuten GPS-paikannin olisi koko ajan päällä. Koska automaattinen virrankatkaisu oli mainittu tavoitteissa, tuli virtakytkimen rakenne toteuttaa niin, että se on mahdollista. Tämä rajasi vaihtoehtoista yksinkertaiset liukukytkimet, koska niitä ei voi ohjelmallisesti käskyttää MCU:lta. Markkinoilla oli kuitenkin vain yksi valmiiksi saatavilla oleva kytkentälevy Adafruitilta, joka mahdollisti virtapiirin avaamisen etänä MCU:lta. (Adafruit Push-button Power Switch Breakout 2013.) (Ks. Kuvio 4.) Kyseinen kytkentä koostui yhdestä NAND-piiristä, kondensaattoreista, vastuksista, P-kanavaisesta MOSFET:istä, transistorista, sekä painonapista.



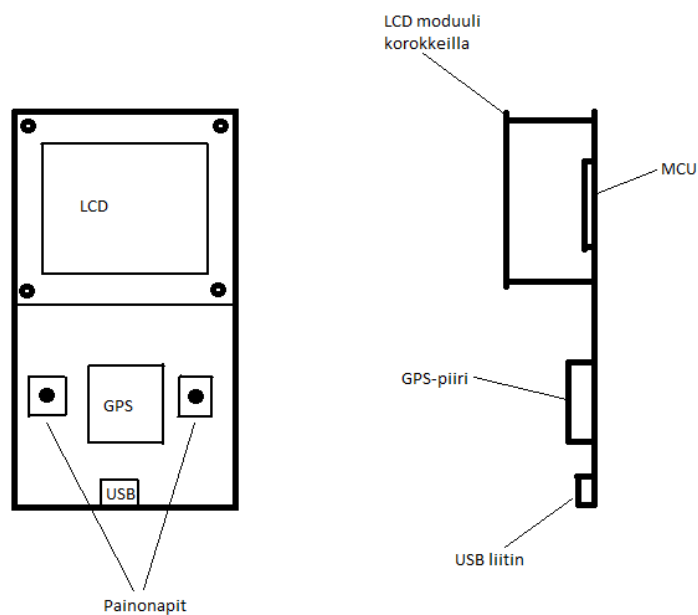
Kuvio 4. On/Off kytkin

GPS tarvitsi graafisen käyttöliittymän tueksi myös painonapit. Käyttäjän on pystyttävä vuorovaikuttamaan GPS-paikantimen kanssa. Järkevän toteutuksen kannalta napit tuli kytkeä GPIO-portteihin, jotka tukevat keskeytyksiä. Valitulla MCU:lla oli vapaana kaksi porttia, jotka tukevat keskeytyksiä. Tämä oli juuri riittävä määrä saavuttamaan jonkin tasoisen käyttöliittymäkokemuksen. Kehityskäyttöön valitsin yleiskäyttöisen läpiladottavan tactile painonapin, mutta lopulliseen versioon valitsin myöhemmin vastaavan pintaliitosmallin.

4.5 Kokonaisuus

Kun pääkomponentit oli valittu, tämän pohjalta pystyin toteuttamaan karkean hahmotelman loppukokonaisuudesta. Kokonaisuudessa tuli erityisesti ottaa huomioon se, että GPS-piirin antenni vaatii mahdollisimman hyvän näkyvyyden taivaalle. Piiri ei siis saanut olla esimerkiksi laitteen selkäpuolella osoittamassa kohti maata. Lisäksi

ajatuksena oli käyttää LCD-näytön omaa moduulia sellaisenaan kytkennässä. Näihin asioihin pohjautuen päädyin lopputulokseen, jossa LCD-näyttö olisi korokkeilla MCU:n päällä ja näytön alapuolelta paljastuisi GPS-piiri sekä painonapit. Näyttö olisi kytkettynä piikkirimalla päälevyyyn. (Ks. Kuvio 5.)



Kuvio 5. Ensimmäinen hahmotelma GPS-paikantimesta

4.6 Firmwären kehitys

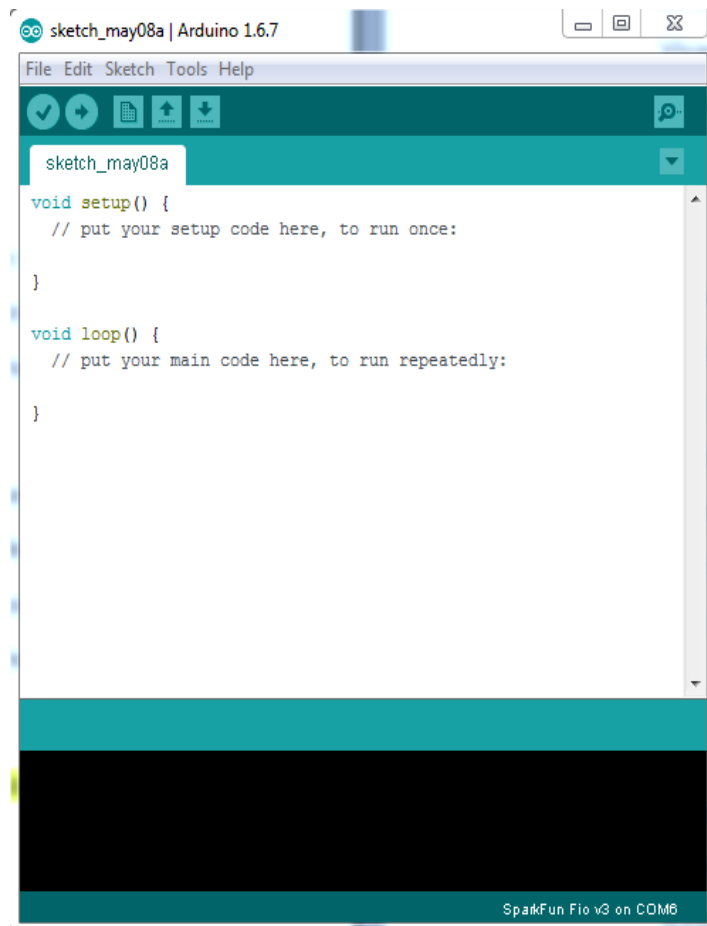
MCU ei itsessään toimi automaattisesti, vaan se tarvitsee firmwären toimiakseen halutulla tavalla. 8-bittisille MCU:lle firmware ohjelmoidaan yleensä C-kielellä tai assembler-kielellä. Kun haluttu toiminnallisuus on ohjelmoitu, se tulee kääntää muotoon, jonka MCU ymmärtää. Käännetty firmware ohjelmoidaan MCU:n muistiin, mikä tapahtuu AVR:än tapauksessa ISP-protokollalla. Tämä vaatii erityisen ohjelmoijakortin, joita on markkinoilla useita versioita. (Ganssle 2015.) Erillinen ohjelmoijakortti

voidaan kiertää, jos MCU:lle on ohjelmoitu bootloader. Bootloader on hyvin yleinen varsinkin Arduino kehitysympäristössä. Jos MCU:lle on ohjelmoitu bootloader, voi sille ohjelmoida firmwären käyttäen UART:ia tai joissain tapauksissa MCU:n natiivina USB protokollaa. 8-bittisten MCU:iden kanssa yleisin vaihtoehto on käyttää UART:ia. Tällöin kokonaisuus vaatii FTDI-piirin käyttöä, joka muuttaa USB-sarjaliikenteen UART yhteensopivaksi. (SparkFun USB to serial UART Boards Hookup Guide n.d.)

4.7 Arduino IDE

Arduino IDE on kehitysympäristö, jossa GPS-paikantimen firmware tehdään. (Ks. Ku-
vio 6.) IDE kattaa hyvin kaikki tarpeelliset kehitystyökalut kuten editorin, kääntäjän,
sarjaporttimonitorin, kirjastojen hallinnan, kehitysalustojen hallinnan ja bootloaderin
ohjelmoinnin. (Arduino Software 2015.)

Arduino IDE on helppokäyttöinen ja siihen löytyy runsaasti dokumentaatiota. Ardui-
non vakiokirjastoille löytyy valmiina esimerkkikoodit jokaiselle ominaisuudelle. Kirjas-
tonhallinnan kautta pystyy lataamaan useita eri kolmannen osapuolen kirjastoja.
Suurin osa näistäkin sisältää esimerkkikoodit. Näitä esimerkkikoodeja hyödyntämällä
voidaan kehittää ominaisuuksia hyvinkin nopeasti sulautettuihin järjestelmiin.



Kuvio 6. Arduino IDE aloitusnäky

4.8 Kirjastot

Projektissa oli Arduinon ulkopuolisia komponentteja, joten näille isommille komponenteille tuli etsiä kirjastot. Kirjastojen avulla yksinkertaistettiin itse pääkoodia ja tehtiin siitä samalla luettavampi. Projektissa tarvittiin ulkopuoliset kirjastot LCD-näytölle sekä GPS-piirille.

4.8.1 GPS-kirjastot

GPS-kirjastot ovat yleensä toiminnaltaan NMEA-parsereita, jotka erittelevät tietoa NMEA-lauseista, jotka sisältävät kaiken tarvittavan paikannustiedon. Valitulle GPS-piirille on olemassa Adafruitin oma GPS-kirjasto. (Adafruit_GPS n.d.) Kyseinen kirjasto on suunnattu nimenomaan MTK3339 piirisarjan GPS-paikantimille, joka oli valittu

projektiin. Kirjasto löytyi myös Arduinon omasta kirjastojen hallinnasta, josta pystyi lataamaan uusimman version. Adafruitin kirjasto oli siis luonnollinen valinta tässä projektissa.

4.8.2 LCD-kirjastot

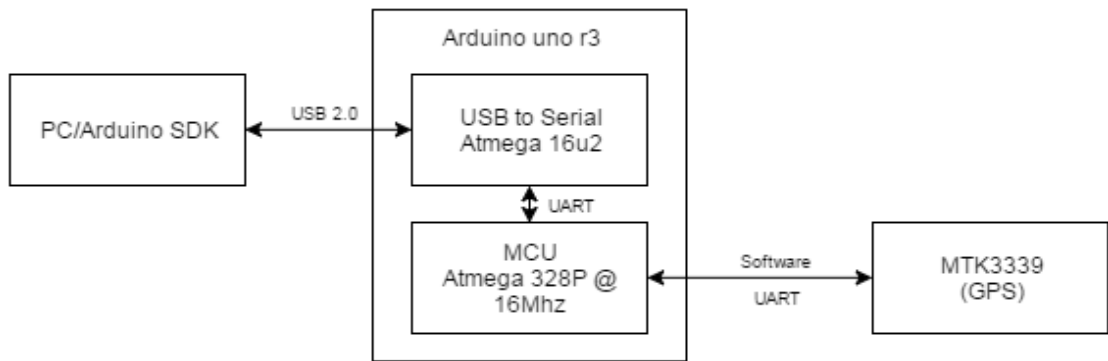
LCD-näytölle oli tarjolla useampia kirjastovaihtoehtoja. Esimerkiksi Sparkfun ja Adafruit tarjosivat kumpikin kyseiselle näytölle omat kirjastonsa. (Adafruit-PCD8544-Nokia-5110-LCD-library n.d.; Graphic LCD Hookup Guide n.d.) Adafruitin kirjasto oli yksinkertaisempi ja se oli saatavilla Arduinon omasta kirjastojenhallinnasta. Adafruitin kirjaston päälle oli saatavissa oma graafinen kirjasto Adafruit-GFX, joka tarjosi useita graafiseen piirtoon tarkoitettuja funktioita, esimerkiksi kirjaimet, viivat, neliöt, pisteet ja bittikuvien piirtäminen näytölle. (Adafruit-GFX-Library n.d.) Adafruitin kirjastot olivat siis helpoin valinta projektiin.

5 Komponenttien käyttöönotto ja testit

Jokainen komponenteista oli testattava erikseen ennen kuin ne voitiin liittää selkeästi yhdeksi kokonaisuudeksi. Näin saatiin selville, kuinka hyvin komponentti toimi. Kun komponentit testattiin, voitiin ne integroida yhdeksi kokonaisuudeksi ja kokeilla yhteistoimintaa. Kun yhteistoiminta oli onnistunut, pystyin aloittamaan piirilevyn kehittämisen.

5.1 GPS kokeilu

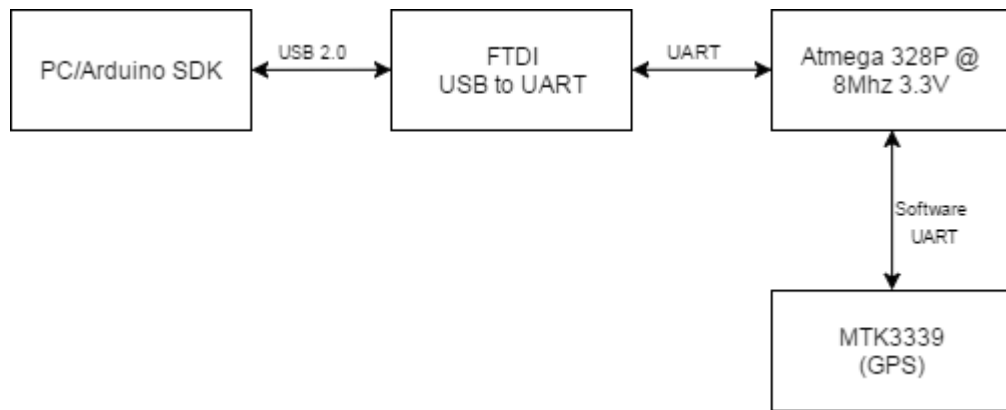
GPS-piirin ensikokeilua en suorittanut Arduino Pro Mini alustalla vaan Arduino Uno alustalla, joka toimii 5V käyttäjännitteellä. Tämän Unon kanssa oli nopea koestaa esimerkikoodi Adafruitin kirjastosta. Esimerkkikoodissa GPS-piiri keskusteli UART:in ylitse Arduinolle, joka sitten lähetti NMEA-lauseet tietokoneelle virtuaalisarjaporttiin parsittuna. (Ks. Kuvio 7.)



Kuvio 7. Lohkokaavio ensimmäisestä GPS-piirin testauksesta

5.2 GPS integrointi

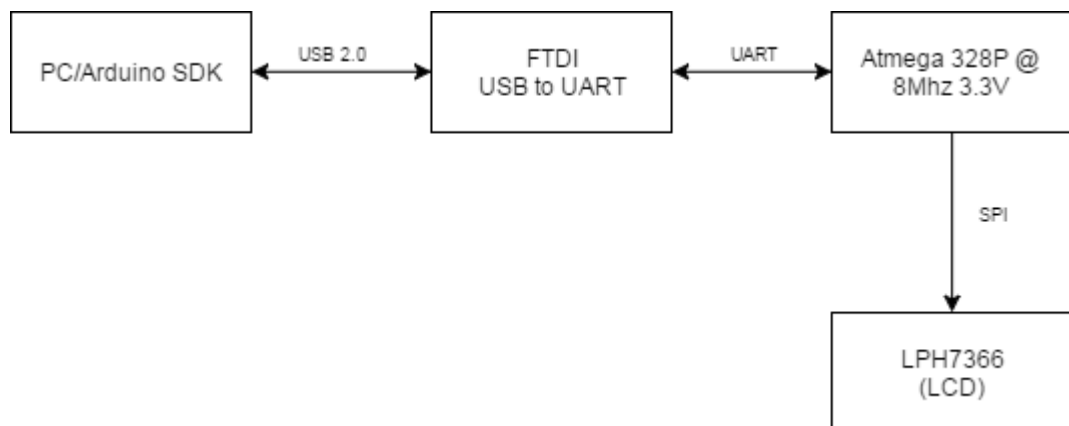
GPS todettiin toimivaksi Arduino Unon kanssa ja seuraavaksi se tuli testata Arduino Pro Minin kanssa. Tämän kanssa ei suuria muutoksia tarvittu. Arduino Uno sekä Pro Mini käyttävät tismalleen samaa MCU:ta, mutta käyttöjännite ja kellotaajuus ovat matalammat. MCU:n ja tietokoneen välissä oleva sarjaliikennemuunnin piiri on vaihtunut Atmega 16u2:sta FTDI-piiriin. Muutoksilla ei pitäisi olla vaikutusta kokonaistoimintaan. GPS-piiri edelleen kommunikoi UART:in ylitse Arduinolle. UART on toteutettu Arduinossa ohjelmallisesti, koska FTDI-piiri kommunikoi Arduinon oman kiinteän UART:in kanssa ja välittää informaatiota sarjaliikenteellä tietokoneelle. (Ks. kuvio 8.) Testaus tehtiin täysin samalla esimerkkikoodilla. MCU:n ja GPS-piirin pinnijärjestystä ei tarvitse vaihtaa.



Kuvio 8. Lohkokaavio GPS-paikantimen kytkennästä Arduino Pro Minin kanssa

5.3 LCD:n kokeilu

LCD-näytön testasin suoraan Arduino Pro Miniin kytkettynä, koska näyttö ei kestä viiden voltin käyttöjännitettä. Pro Minin 3,3V käyttöjännite on ideaalinen näytölle ja tämän takia ei tarvinnut asentaa SPI-väylään virtaa rajoittavia vastuksia tai logiikkajännitteen muunninta. (Ks. Kuvio 9.) Testatessani LCD-moduulia huomasin kahden kolmesta levystä olevan viallisia ja näytölle ei ilmestynyt kuvaa. Yhdessä viallisessa näytön taustavalon kaikki ledit eivät palaneet. Onneksi yksi levyistä toimi.

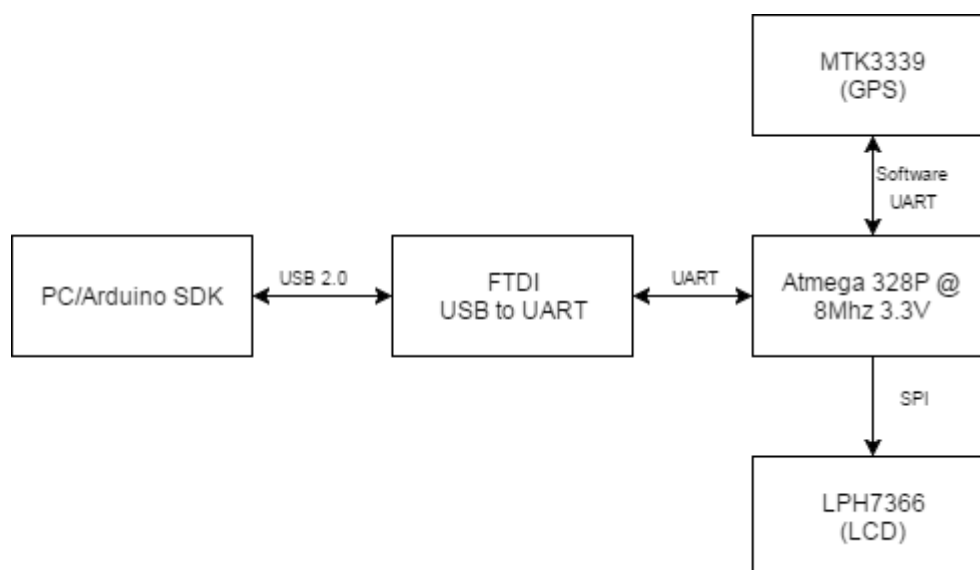


Kuvio 9. Lohkokaavio LCD-näytön kytkennästä Arduino Pro Minin kanssa

Ohjelmallisessa testauksessa käytin Adafruitin kirjaston mukana tulevaa LCD-test esimerkkikoodia. Kyseinen esimerkki näyttää Adafruit GFX-kirjaston perusfunktiot lävitse. Koodiesimerkki tulostaa LCD-näytölle bittikuvioita, neliöitä, kolmioita, viivoja ja tekstiä useina kappaleina. Havaitsin myös, että näytön päivitysnopeus on yllättävän ripeä. Vähän ajan kuluttua näyttö alkoi tummua ja vaikutti siltä, ettei mekaanisesti viimeinenkään näyttö ollut aivan kunnossa. Tästä syystä päätin tilata muutaman eri valmistajan levyn samalla näytöltä varmuuden vuoksi. Muuten näyttö oli osoittautunut toimivaksi.

5.4 Kokonaisuuden integrointi

Pääkomponenttien erillistestauksen jälkeen oli aika kokeilla komponenttien yhteistoimintaa. GPS-kirjaston esimerkin päälle muutin paikkatiedon ulostulon LCD-näytölle käyttäen LCD-kirjaston tekstin kirjoitusfunktioita. Tämä prosessi ei vaatinut kovin suuria muutoksia kokonaiskoodiin. GPS-piirin ja LCD-näytön kytkentöjä ei tarvinnut muuttaa, koska ne käyttävät kumpikin aivan eri kommunikaatioväylää. (Ks. Kuvio 10.)



Kuvio 10. Lohkokaavio GPS-piirin ja LCD-näytön yhteistoiminnasta Arduino Pro Minin kanssa

5.5 Testaus ja tulokset

Tässä vaiheessa toteutin testausta useamman kierroksen. LCD-näyttö antaa tällä hetkellä paikkatiedot koordinaatteina sekä korkeuden merenpinnasta metreinä. Jonkin ajan kuluttua havaitsin, että koordinaatit menivät sekaisin. Esimerkiksi toinen tai molemmat koordinaateista saattoi muuttua 0:ksi tai satunnaisnumeroksi. Välillä koordinaatit näyttivät oikein. Kyseessä oli siis selkeä ongelma, joka täytyi selvittää.

5.5.1 Ongelman selvitys

Vian selvittäminen alkoi vaihtamalla testausympäristöä sisältä ulkotiloihin. Sisällä on tunnetusti huono GPS-kuuluvuus. Ulkona kuitenkin paikkatieto alkoi radikaalisti hajoata samalla tavalla kuin sisätiloissa. Tämä viittasi, että GPS-paikannin vaikuttaisi vastaanottavan satelliittien radioaalto aivan oikein.

Toinen lähtökohta vian selvitykselle oli tutkia raakaa NMEA-dataa USB-sarjaportin ylitse tietokoneella. Sarjaportin ylitse tuli NMEA-data sisältäen koordinaatit aivan oikein. Havaitsin kuitenkin, että välillä NMEA-lause saattaa jäädä samalle riville edellisen lauseen kanssa, mutta koordinaatit ovat kuitenkin oikein.

Kolmannella kerralla palautin GPS-kirjaston parserin toimimaan USB-sarjaportille, jotta voin tarkastella tulevatko koordinaatit vielä oikein. Koordinaateissa oli tällä kertaa havaittavissa sama ongelma, mikä ilmeni LCD-näytölle tulevissa koordinaateissa. Tämä vaikuttaa siltä, että NMEA-parseri ei toimi tässä kokoonpanossa oikein.

Lähdin tutkimaan asiaa foorumeilta, että olisiko muilla ilmennyt samaa. Löysin vastaavan ongelman ilmenneen Adafruitin foorumilla. (Ultimate GPS breakout: Problems with Arduino Fio 2012.) Ongelma ilmeisesti tulee esille, kun käytetään ohjelmallista UART:ia ja MCU käy 8 Mhz kellotaajuudella.

5.6 Vian korjaaminen

Ensimmäinen korjauskokeiluni perustui UART:in Baud-raten muuttamiselle. MTK3339 kommunikoi tehdasasetuksilla 9600 Baudin nopeudella. Päätin puolittaa Baud raten GPS-piirille 4800 Baudiin. Tämä muutos tapahtuu lähettämällä AT-komentoja GPS-piirille alkuperäisellä nopeudella UART:ia pitkin. (GlobalTop PMTK command packet

2012.) Tämä prosessi kyllä muutti baud-raten oikein, mutta se ei kuitenkaan korjannut asiaa.

Seuraava lähestymistapa oli vaihtaa koko Adafruitin GPS-kirjasto toiseen kirjastoon. Adafruitin kirjasto ei tarjonnut NMEA-lauseen tarkistusta vaan sylkee ulos sen, mitä parseri vain pilkkoo NMEA-lauseesta. Yleisimpiä Arduinolle tehtyjä GPS-kirjastoja on TinyGPS. (Hart 2013.) Kyseinen kirjasto tarjosi tarkistussumman tarkistuksen NMEA-lauseesta eli jos tarkistussumma ei osu oikeaan, niin koordinaatteja ei hyväksytä lauseesta. Tämä estäisi sen, että LCD-näytöllä olevat koordinaatit eivät olisi satunnaisarvoja vaan koordinaatit viimeisimmästä onnistuneesta parsinnasta.

TinyGPS kirjasto korjasi testeissä näytöllä näkyvät vialliset koordinaatit, mutta päätin kuitenkin syventyä ongelmaan ja ohjelmoin LCD-näytölle kirjaston tarjoamat diagnostiikka arvot saapuneista NMEA-lauseista sekä lukumäärät hylätyistä NMEA-lauseista. Tässä testissä havaitsin, että hyväksytyt NMEA-lauseet olivat vain 20 % kaikista NMEA-lauseista. Tämä tilanne oli epästabiili ja tuloksena oli liian epätarkka GPS-paikannin käytettäväksi, koska paikkatieto ei päivity enään säännöllisesti.

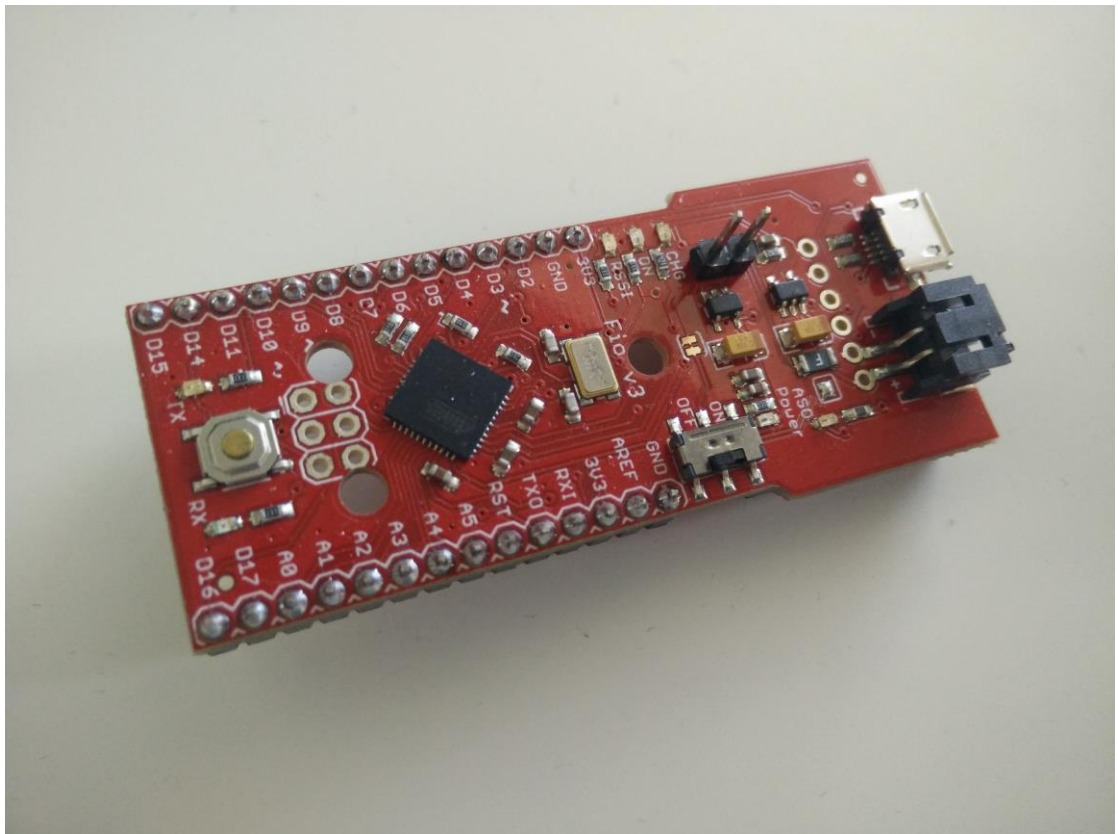
Päädyin lopputulokseen, että ohjelmallinen UART ei ole kyllin pätevä vastaanottamaan vakaasti NMEA-dataa. Tämä vaatii sen, että Atmega 328 MCU tulee vaihtaa kyvykkäämpään MCU:hun, joka sisältäisi tietokonekommunikaation lisäksi yhden vapaan UART-väylän GPS-paikantimelle.

6 Toinen kehitysversio

6.1 Uuden alustan valinta

Koska Arduino Pro Minin MCU ei kyennyt pysymään GPS:n mukana, jouduin vaihtamaan koko kehitysalustan sopivampaan. Alustaa valitessa tuli siis jättää kaikki ne vaihtoehdot jotka käyttävät Atmega 328:a tai pienempiä MCU:ita. Yksi vaihtoehtoista oli löytää alusta, joka käyttäisi Atmega 32u4:ää 3,3 V käyttöjännitteellä. Kyseisenlaista kehitysalustaa ei tällä hetkellä ollut saatavilla Arduinon natiivista kehitysalustavalikoimasta. Tämän takia alusta tuli etsiä kolmannen osapuolien kehitysalustavali-koimista, kuten Adafruit tai Sparkfun.

Alustavalikoimaa selatessani törmäsin Sparkfunin Fio V3 alustaan. (Ks. Kuvio 11.) (Fio v3 - ATmega32u4 n.d.) Kyseinen alusta käyttää Atmega 32u4 MCU:ta ja käy 3,3 V käyttöjännitteellä. Lisäksi alustaan on integroitu valmiiksi edellä valittu Li-Po laturipiiri sekä 3,3 V regulaattori. Fio v3 tarjoaa myös mahdollisuuden käyttää ulkopuolista virtakytkintä, joka on ideaalinen Adafruitin virtakytkimen testaamiseen. Alusta siis tarjosi jo valmiiksi suurimman osan kytkennästäni toteutettuna ja Sparkfun tarjosi samalla levystä valmiin dokumentaation omiin toteutuksiin. Tämä alusta olisi pitänyt valita ensisijaisesti ennen Arduino Pro Miniä.



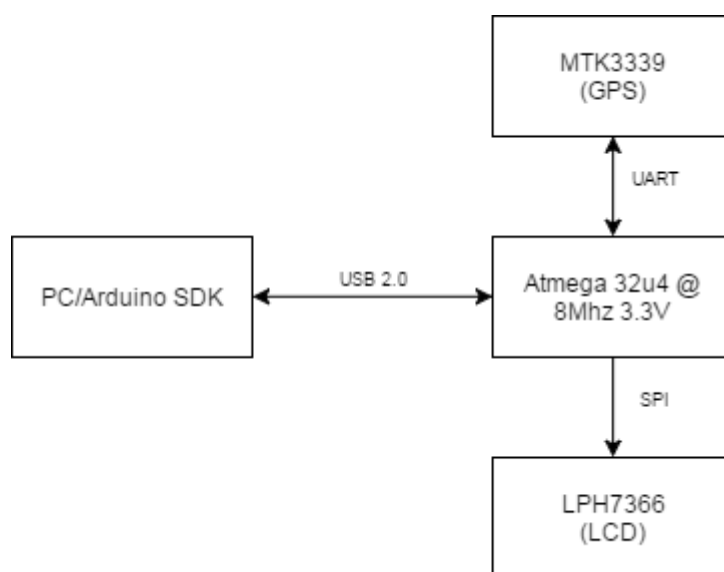
Kuvio 11. Sparkfun Fio V3

6.2 Uuden alustan käyttöönotto ja testaus

Sparkfun oli toteuttanut käyttöönotto-ohjeet sivuillaan Fio V3 alustalle. (Pro Micro & Fio V3 hookup Guide n.d.) Ohjeet kertoivat levyn lisäämisohjeet Arduino IDE:en sekä ajurien asentamisen. Käyttöönottosivusto kertoi, myös Fio V3:n USB-bootloaderin toiminnan.

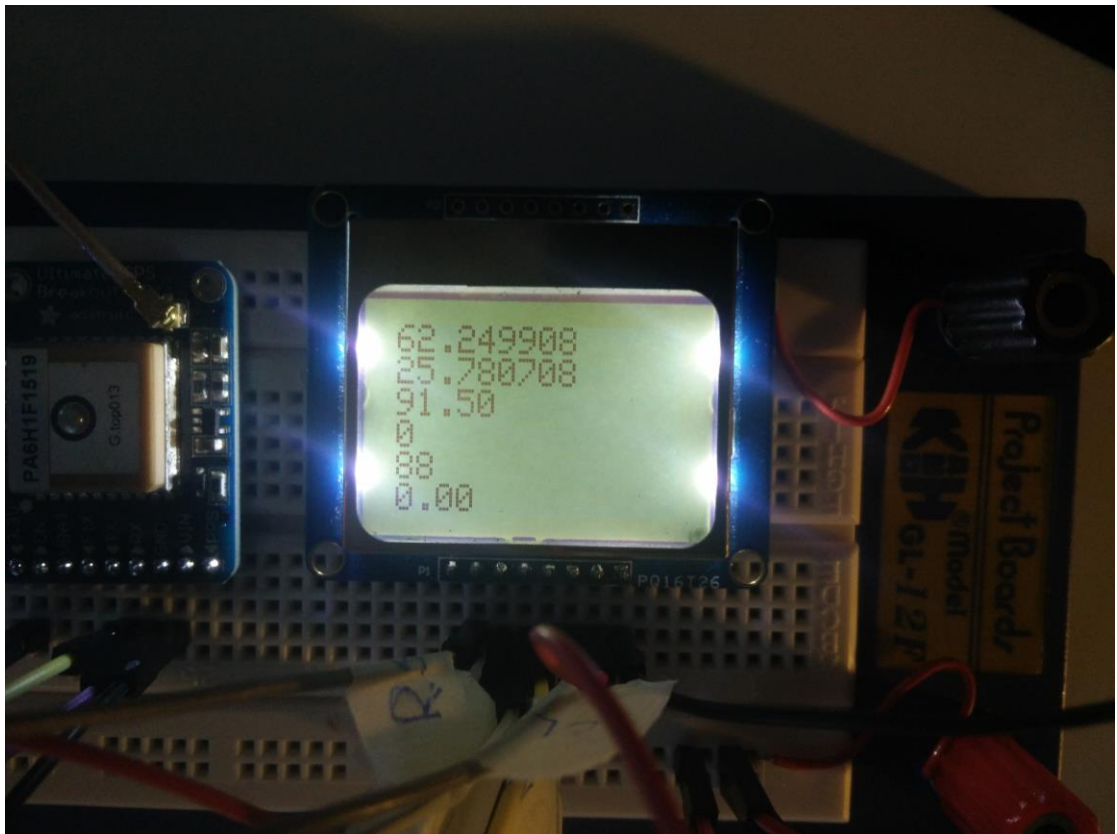
Levyn onnistuneen asennuksen jälkeen oli aika muokata alkuperäinen ohjelmakoodi toimimaan uuden MCU:n kanssa. Alkuperäisessä ohjelmakoodissa GPS-UART-linja oli ohjelmapohjaisella toteutuksella. Nyt uudella alustalla UART voitiin vaihtaa MCU:n sisäiseen aitoon UART-linjaan. Tämä tarkoitti, että ohjelmakoodissa SoftwareSerial toteutus vaihtui HardwareSerial toteutukseksi. Tämä oli varsin yksinkertainen toimenpide, koska HardwareSerial ei vaatinut pinnimäärittelyjä Atmega 32u4:n tapauksessa.

LCD-näytön tapauksessa muutoksia ei tullut, koska Arduino Pro Minin sekä Fio v3:n GPIO pinniviittaukset ohjelmakoodissa olivat käytännössä samat. Tästä johtuen tarvitsi vain huolehtia, että LCD-näyttö kytketään samoihin GPIO-portteihin. (Ks. Kuvio 12)



Kuvio 12. Lohkokaavio Sparkfun Fio V3 koekytkenästä

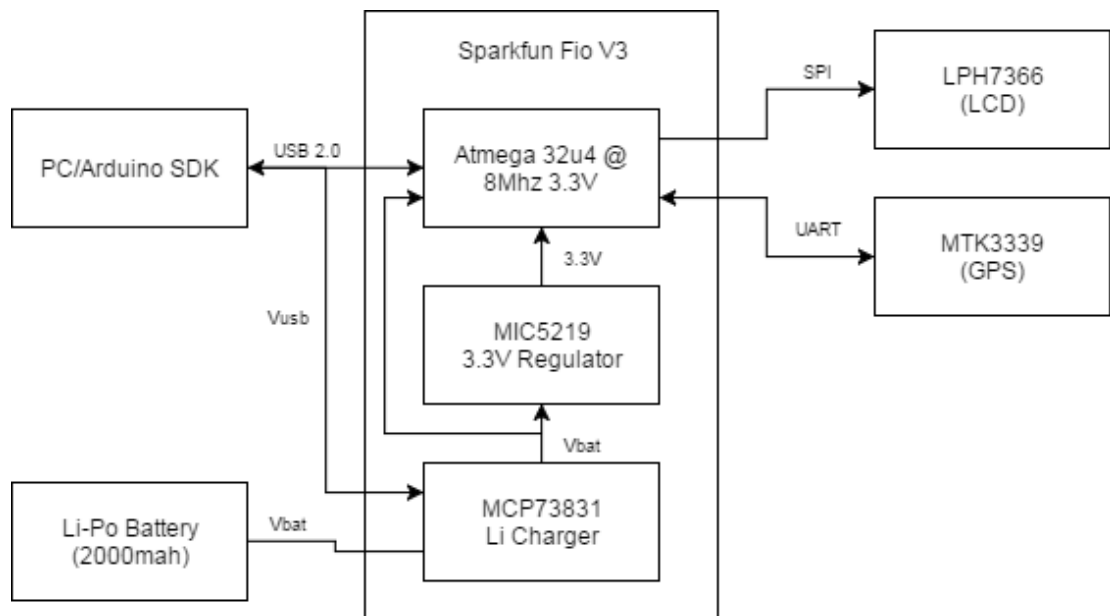
Käyttöönottestissä käytin edelleen TinyGPS kirjastoa apunani ja sen tuomia diagnostiikka muuttujia. Tämän avulla pystyin varmistamaan sen, että vika oli korjaantunut uudella MCU:lla. Noin puolentunnin käyttötestissä ei tullut yhtäkään hylättyä NMEA-lausetta. Kuviossa 13 on havainnollistettu vastaava testi muutaman minuutin suoritusajalla. Ylhäältä alaspäin olevat arvojen merkitykset: Lat koordinaatti, Lon koordinaatti, korkeus, hylätyt NMEA-lauseet, vastaanotetut NMEA-lauseet ja hylkäämisprosentti. Kuvioista nähdään, että vastaanotettuja NMEA-lauseita on 88 kpl ja hylkäämisprosentti on 0 %. Tämä tarkoittaa sitä, että alustanvaihto korjasi ongelman ja prototyypin kehitystä voitiin jatkaa eteenpäin.



Kuvio 13. Fio V3:n alustatestaus

6.3 Akkuahti

Sparkfun Fio V3 alustalla oli valmiina kytkentä, jonka avulla voidaan mitata alustaan kytketyn Li-Po-akun jännite. Kytkennässä oli tehty akulta jännitteenjako kahdella 1 k Ω vastuksella. (Ahoranta 2006, 99; Arduino-Fio_v31 2014.) Jännitteenjaolla puolitettu akun jännite oli kytketty MCU:n ADC-kanavaan. Tämä oli tehty siksi, koska Li-Po-akun täysi jännite 4,2 V ylittää MCU:n käyttöjännitteen 3,3 V. MCU käytti ADC kanavalla jännitereferenssiä, joka tuli suoraan käyttöjännitteestä. Jos akun käyttöjännite olisi kytketty suoraan ADC-kanavalle niin ADC digitaalinen arvo alkaisi muuttumaan vasta akun jännitteen tiputtua alle 3,3 V:in. Ohjelma toteutuksessa tuli siis huolehtia, että ADC-kanavan arvo tulee kaksinkertaistaa, jotta saadaan oikea akun jännite. Kuviossa 14 on hahmoteltuna nykyinen toteutus.



Kuvio 14. Lohkokaavio akun kytkennästä nykyiseen toteutukseen

6.3.1 Akkuvahdin testaus

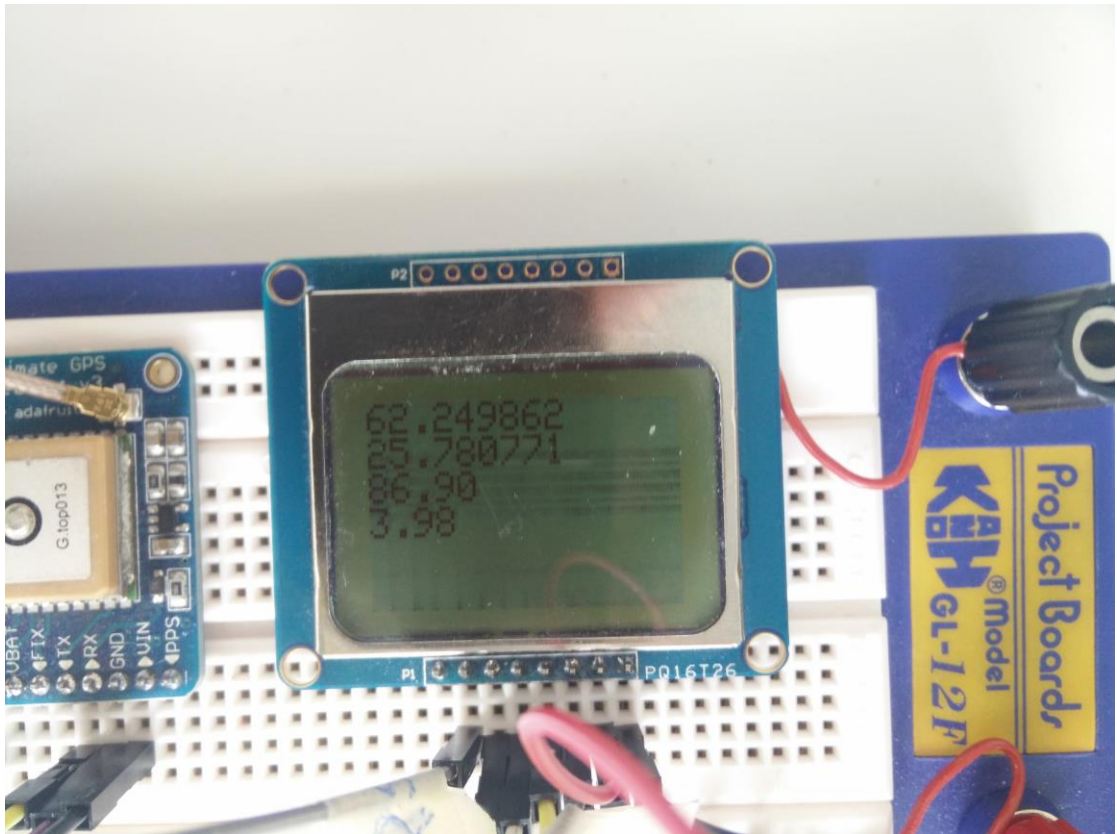
Akkuvahdin testauksessa poistin NMEA-diagnostiikkamuuttujat LCD-näytöltä ja ohjelmoin tilalle muuttujan, joka näyttää todellisen akun jännitteen voltteina. Testeissä huomasin, että LCD-näytöllä oleva jännite on karkeasti 0,3 V matalampi, mitä akun jännite on kuormassa yleismittarilla mitattuna. Jotain oli siis mahdollisesti hieman pielessä akun jännitteen mittauksessa.

6.3.2 ADC-ongelma ja selvitystyö

Pieni jännitteen notkahdus voidaan korjata ohjelmallisesti lisäämällä lopulliseen arvoon niin sanottu "offset" arvo, joka korjaisi lukeman näyttämään oikealta. Halusin kuitenkin saada asiaan täyden selvyden, miksi poikkeamaa oli. Asiaa kokeilin korjata liittämällä AREF ("analog reference") portin käyttöjännitteeseen 3,3 V, jotta referenssijännite olisi varmasti oikein. Tämä ei auttanut ongelmaan. Seuraavaksi kokeilin auttaa ADC:tä liittämällä kondensaattorin rinnan maatasoon akkua mittaavan ADC-portin kanssa. Tämäkään toimenpide ei auttanut korjaamaan poikkeamaa. Tämän jälkeen päätin kokeeksi tehdä mittauksen toisella ADC-kanavalla. Tein toisen muuttujan, joka tulostui myös LCD-näytölle jännitteenä. Kanavalle syötin jännitteen säädettävältä jännitelähteeltä ja lopputulos oli sama kuin akkua mittaavalla portilla. Lopulta ratkaiseva vihje löytyi, kun vaihdoin LCD-näytölle tulevan muuttujan ADC:n raakaan digitaaliseen arvoon. Käsin laskettuna 3,3 V referenssistä tulokset olivat oikein. Tämä tarkoitti sitä, että todellisen jännitteen laskeminen tapahtui ohjelmallisesti väärin.

```
batteryVoltage = battery * 2 * (3.3 / 1023.0);
```

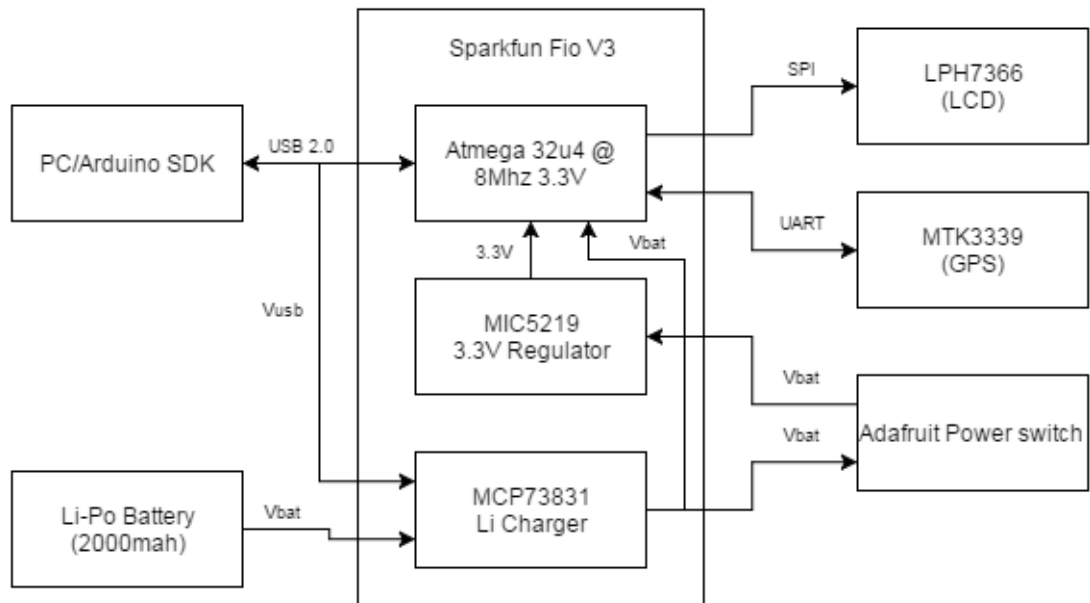
Ohjelmarivillä laskennallinen referenssijännite oli kirjoitettu muotoon: "3, 3". Tämä ilmeisesti aiheutti sen, että laskutoimituksessa referenssijännitteenä käytettiin arvoa 3 V. Korjaamalla referenssiarvo muotoon "3 . 3" kyseinen vika korjaantui ja LCD-näytöllä näkyvä akun jännite on nyt oikein. (Ks. Kuvio 15.)



Kuvio 15. Akun jännitteen näyttäminen näytöllä

6.4 Virtakytkimen toteutus

Virtakytkimeksi olin edellä valinnut Adafruitin On/Off kytkimen. Tämän kytkennän testaaminen ei ollut monimutkainen toimenpide. Sparkfunin Fio V3 alustassa oli piikkirima paikka ulkoiselle virtakytkimelle, jonka voi havaita kuvioista 11. Fio V3 alustalla tuli vain kytkeä valmiiksi juotettu virtakytkin off-asentoon, jotta MCU ja oheiskomponentit eivät kytkeytyneet päälle sitä kautta. Kokonaiskytkennässä tämä kytkin tuli Li-Po akun ja regulaattorin väliin. (Ks. Kuvio 16.) Testeissä havaitsin, että On/Off kytkin ei ole niin stabiili päälle kytkemisessä tai sammuttamisessa, mutta tarpeeksi vakaaksi omaan käyttöön.



Kuvio 16. Lohkokaavio virtakytkimen liittämistä kytkentään

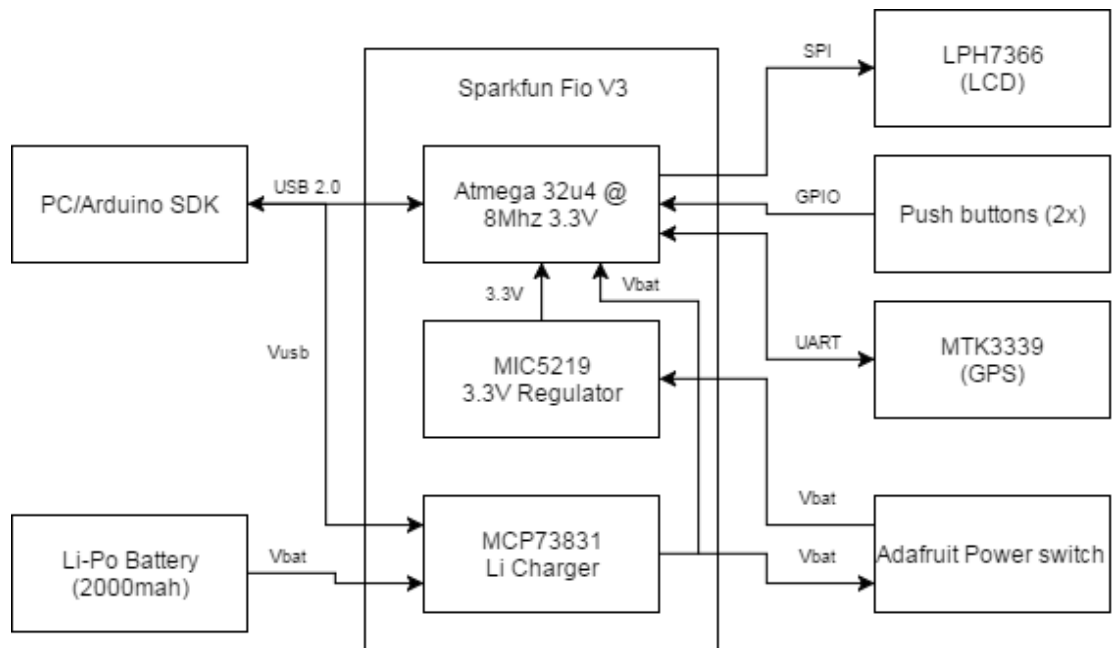
6.4.1 Automaattinen sammutus

Adafruitin On/Off kytkin tuki ulkoista sammutusta. Tämä toimenpide toimii siten, että kytkimen ulkopuolelta ohjataan virtaa kytkimen off-nastaan. (Adafruit Push-button Power Switch Breakout 2013.) Toteutuksessa ohjattiin MCU:n GPIO-portti loogiseksi 1:ksi, joka oli kytketty kyseiseen off-pinniin virtakytkimellä. Tällä toteutuksella voitiin toteuttaa automaattinen sammutus. Tämä ominaisuus estää akkua tyhjenemästä turhaan, jos laite on ollut kauan käyttämättömänä päällä. Toinen käyttötapa on esimerkiksi tarkastella akun jännitettä ja sammuttaa laite automaattisesti, jos akku käy liian tyhjäksi. Testauksessa ohjelmoitiin GPS-paikantimen sammuttamaan itsensä minuutin jälkeen, kun laite oli saanut ensimmäisen paikkatiedon. Tämä toiminto toimi ilman ongelmia.

6.5 Käyttäjäpainikkeet

Jotta GPS-paikanninta voitaisiin ohjata ja käyttää, tarvitaan tapoja, joilla käyttäjä voi vuorovaikuttaa laitteen kanssa. Käyttäjälle yksinkertaisin tapa on tehdä toteutus pai-

nonapeilla. Ilman monimutkaisempia kytkentöjä tämänhetkiseen toteutukseen voitiin lisätä kaksi painonappia. Painonapit kytkettiin MCU:n keskeytyksiä tukeviin GPIO-portteihin. Fio V3 alustalla kyseiset portit olivat D2 ja D3. (Ks. Kuvio 17.)

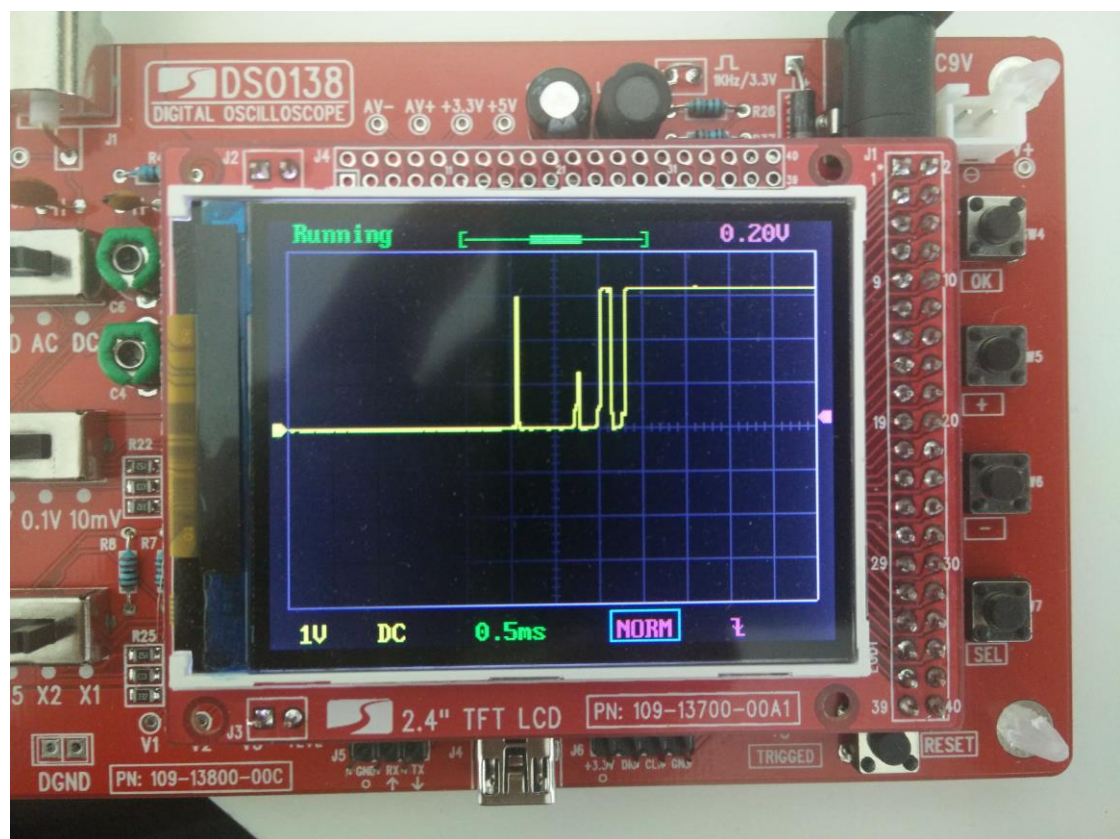


Kuvio 17. Lohkokaavio painonappien liittämisestä kokonaiskytkentään

Portit ohjattiin loogisesti ylös MCU:n sisäisillä ylösvetovastuksilla. (Digital Pins n.d.) Painonappien toinen puoli vietiin maatasoon. Tällä kytkennällä mahdollistettiin looginen muutos nappia painettaessa keskeytysporttiin. Kun nappi painetaan pohjaan, portin looginen taso siirtyy maatasoksi eli loogiseksi 0:ksi. Ohjelmallisesti ulkoisia keskeytyksiä seurataan portin nousevalla tilalla (0 -> 1) tai tippuvalla tilalla (1 -> 0). Tässä toteutuksessa käytettiin tippuvaa reunaa eli keskeytysohjelma ajetaan, kun nappi painetaan pohjaan. Nousevassa tapauksessa ohjelma ajettaisiin, kun nappi vapautetaan.

6.5.1 Bounce-efekti

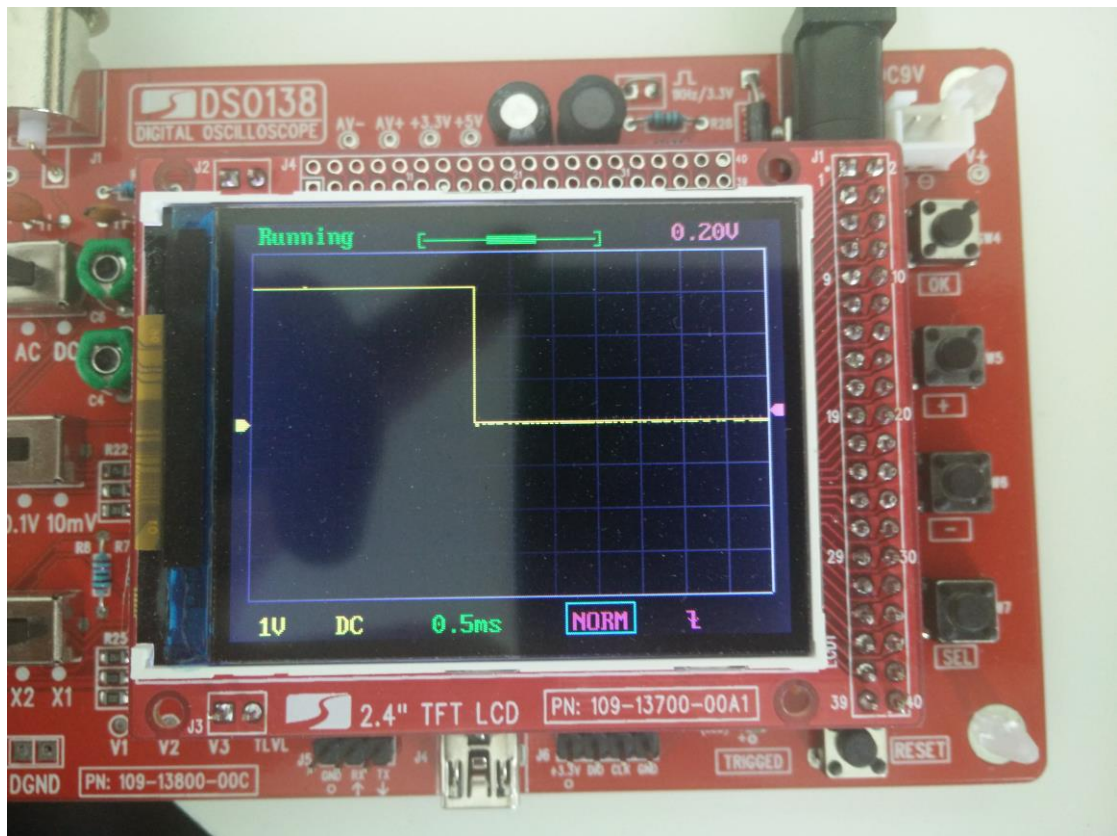
Tavallinen painonappi tai kytkin ei kuitenkaan ole niin yksinkertainen komponentti kuin voisi kuvitella. Ideaalisella tasolla tila vaihtuu nolasta ykköseksi suoraan. Oikeassa elämässä tulee vastaan sähköän hyppiminen ja tämän seurauksena tilan vaihtuminen voi napin painamishetkellä tai vapauttamishetkellä tapahtua useamman kerran lyhyen ajan sisällä. (Ks. Kuvio 18.)



Kuvio 18. Napin vapautuessa tapahtunut bounce-efekti

Bounce-efektin voi kytkennässä suodattaa pois kytkemällä napin rinnalle kondensaattorin. Kondensaattorin suuruuden voi laskea käyttämällä kondensaattorin aikavakion laskukaavaa RC-kytkennässä (Resistor-Capacitor), jossa vastuksena toimi MCU:n sisäinen ylösvetovastus, joka on kokoluokkaa 20 k Ω - 50 k Ω . (Gammon 2013;

ATmega16U4/ATmega32U4 2015, 384.) Nappia painettaessa varautunut kondensaattori peittää alleen tulevat bounce-efektit ja napin vapautuessa kondensaattori varautuu käyttöjännitteen tasolle suodattaen bounce-efektin. Kuviosta 18 nähdään bounce-efektin olevan noin 1,5 ms. Bounce aika vaihtelee aina ja tämän takia tuli valita kondensaattori, joka peittäisi alleen vähän pidemmänkin bounce-efektin. Bounce-efekti tulee kasvamaan napin käyttöiän myötä ja tämä tuli ottaa huomioon. Mittauksissa oli käytössä aivan uusi nappi, joka ei juuri ollenkaan antanut bounce-efektiä, kun nappi painettiin pohjaan. (Ks. Kuvio 19.)



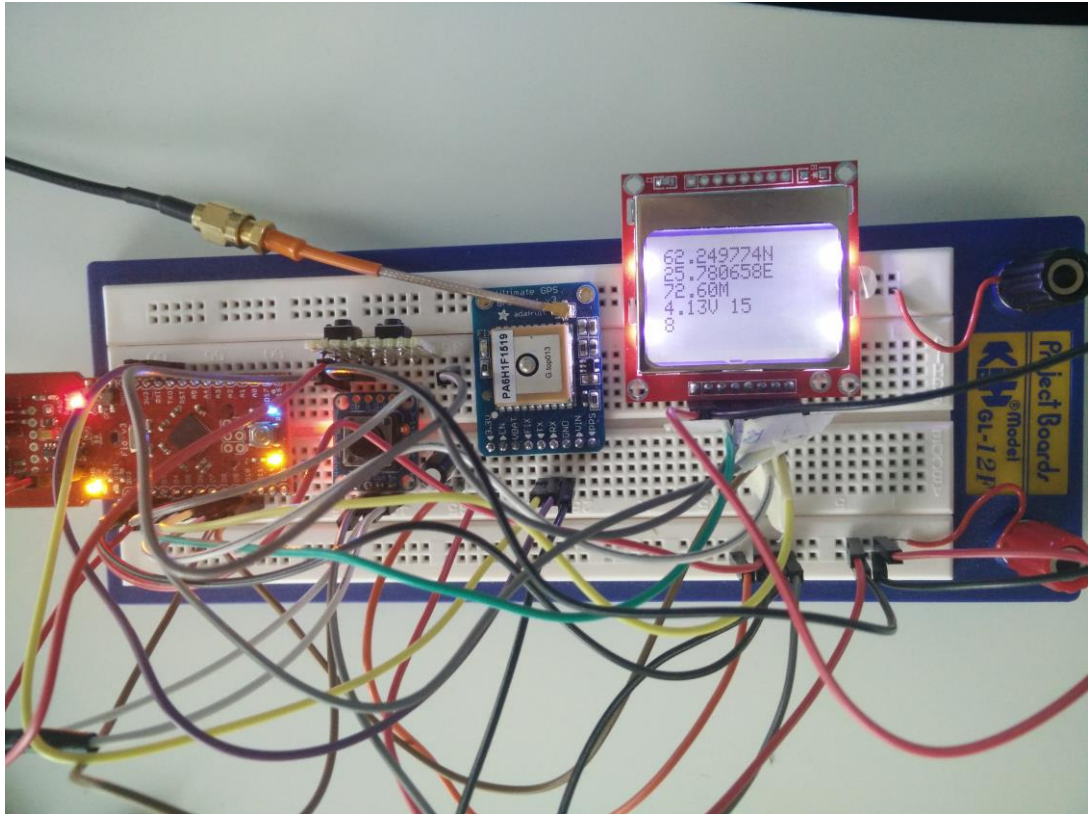
Kuvio 19. Mittauksissa ollut nappi ei antanut painettaessa bounce-efektiä

6.5.2 Ohjelmatoteutus

Kuten edellä mainittu, painonappien ohjelmatoteutus perustui keskeytyksiin. Keskeytykseen liitetty aliohjelma ajetaan välittömästi keskeytyksen tapahtuessa. Ohjelman pääsilmukassa tapahtuva toteutus pysähtyy keskeytyksen ajaksi ja jatkaa keskeytyksen jälkeen siitä mihin alkuperäinen suoritus jäi. (Digital Pins With Interrupts n.d.) Keskeytyksillä vältytään pääohjelmassa jatkuvalta GPIO-portin tilan kyselyltä. Jatkuvaan kyselyyn perustuva toteutus saattaa joskus olla huomaamatta painalluksen, jos pääohjelman kokonaistoteutus on raskas ja napin painaminen tapahtuu portin tilan kyselyn välissä.

6.5.3 Painikkeiden testaus

Testaukseen ohjelmoin keskeytysaliohjelmaan laskurin, joka tulostaa muuttujansa LCD-näytölle. Tällä tavalla näin tapahtuneet painallukset ja mahdolliset bounce efektit. Debounce kondensaattoriksi kytkin $0,47 \mu\text{F}$ elektrolyyttikondensaattorin. (Ks. Kuvio 20.) Tämän pitäisi olla teoreettisesti riittävä, mutta bounce efektiä oli havaittavissa. Tämä tarkoitti, että bounce-efektiä oli korjattava mahdollisesti myös ohjelmatoteutuksessa. Lopulliseen kokoonpanoon kytkin debounce kondensaattoriksi $1 \mu\text{F}$ kondensaattorin.



Kuvio 20. Painikkeen toteutuksen testaus

6.5.4 Ohjelmallinen debounce korjaus

Ohjelmallisesti päätin korjata bouncea sisältämällä ehtolauseen keskeytysaliohjelmaan. Ehtolauseessa tarkasteltiin edellisestä keskeytyksestä kulunutta aikaa. Jos edellisestä keskeytyksestä oli alle 200 ms niin keskeytysaliohjelman sisältöä ei ajettu. Tämä toteutus ei kuitenkaan korjannut tilannetta, jossa nappia painettiin pitkään ja vapautuksessa ilmentynyt debounce ajautui kuitenkin läpi. Toteutus oli kuitenkin riittävä toteamaan napin toiminnallisuuden onnistuneen.

7 Piirilevyn kehitysprosessi

Kun kokonaisytkentä oli saatu toimimaan kehitysalustoilla, voitiin aloittaa lopullisen prototyypin fyysinen toteutus. Tämä prosessi sisälsi kokonaisytkennän mallintamisen ja sen pohjalta pääpiirilevyn suunnittelun. Lopuksi kun pääpiirilevy oli suunnitel-

tu, voitiin teetättää tämä levy prototyypilevyksi. Näin saataisiin oikeaoppinen ja siisti toteutus prototyypille.

7.1 CadSoft Eagle

Piirilevyn suunnitteluprosessi toteutettiin CadSoft Eagle piirilevy CAD-ohjelmistolla. Ohjelmisto koostui kolmesta osasta: Autoroute, kytkentäkaavio ja levynpiirros(Layout). Eaglesta oli tarjolla rajoitettu ilmainen versio, joka tarjoaisi riittävät ominaisuudet tämän projektin toteutukseen. Ilmaisversio rajoitti piirilevyn kokoa, piirilevyn kerrosten määrää sekä kytkentäkaavioita, joita voi olla vain yksi projektia kohden. (What is EAGLE 2011.)

7.2 Kytkentäkaavio

Levyn toteuttaminen alkoi kytkentäkaavion tekemisellä. Siinä tuli koostaa kaikki käytetyt komponentit yhteen, jotka olivat olleet kehitysalustoissa kiinni. Työssä tuli myös karsia ylimääräisiä komponentteja pois, koska osa komponenteista kehityslautalla ja referenssilevyillä eivät olleet tarpeellisia. Tästä on hyvä esimerkki GPS-levyllä olevasta omasta regulaattorista, joka oli tässä kytkennässä turha.

7.2.1 Kehitysalustan kytkentäkaavio

Sparkfun tarjosi Fio V3 kehitysalustastaan valmiin Eagle kytkentäkaaviotiedoston. (Arduino-Fio_v31 2014.) Tämä sisälsi kaikki komponentit valmiiksi valittuna ja oikein kytkettynä. Otin tämän kytkentäkaavion suoraan eagle projektiin, jonka päälle tein muutokset ja muut kytkennät.

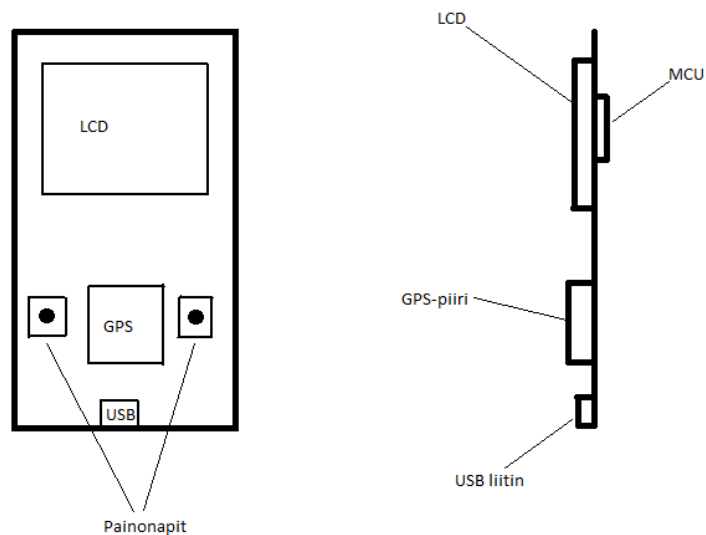
7.2.2 Muutokset kehitysalustan kytkentään

Suurimmat muutokset koko kytkennässä vaati kehitysalusta. Ensimmäinen muutos oli poistaa kehitysalustan piikkirimat ja xBeen kytkennät. Lopullisessa prototyypissä ei tarvita jokaista MCU:n porttia viedä ulos erilliselle liittimelle. Seuraava muutos oli poistaa originaali virtakytkin sekä piikkirima ulkoiselle kytkimelle. Tämän korvasin Adafruitin tarjoamalla On/Off kytkennällä. Kolmas isompi muutos oli vaihtaa MCU:n kotelotyyppi QFN:stä TQFP:hen, jotta se olisi helpompi käsin juottaa levyille. Päätin

samalla myös vaihtaa kellokiteen kotelotyyppiä helpommin juotettavaksi. Tähän etsin 8 Mhz kellokiteen 5032-koteloinnilla.

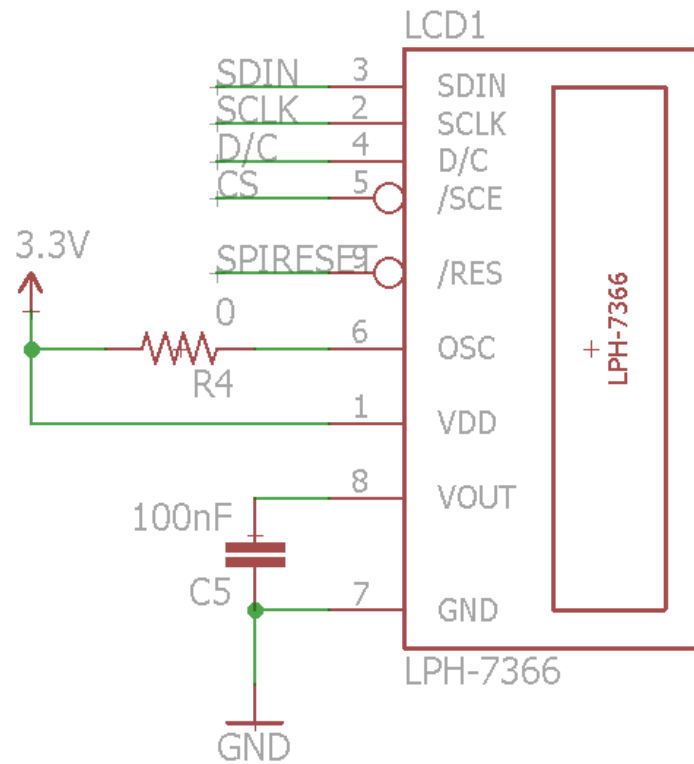
7.2.3 LCD-näytön kytkentä

LCD-näyttöön löysin valmiin komponentin, joka mahdollisti siistimmän kokonaistoteutuksen. (lph-7366.lbr 2007.) (Ks. Kuvio 21.)



Kuvio 21. Uusi hahmotelma kokonaistoteutuksesta

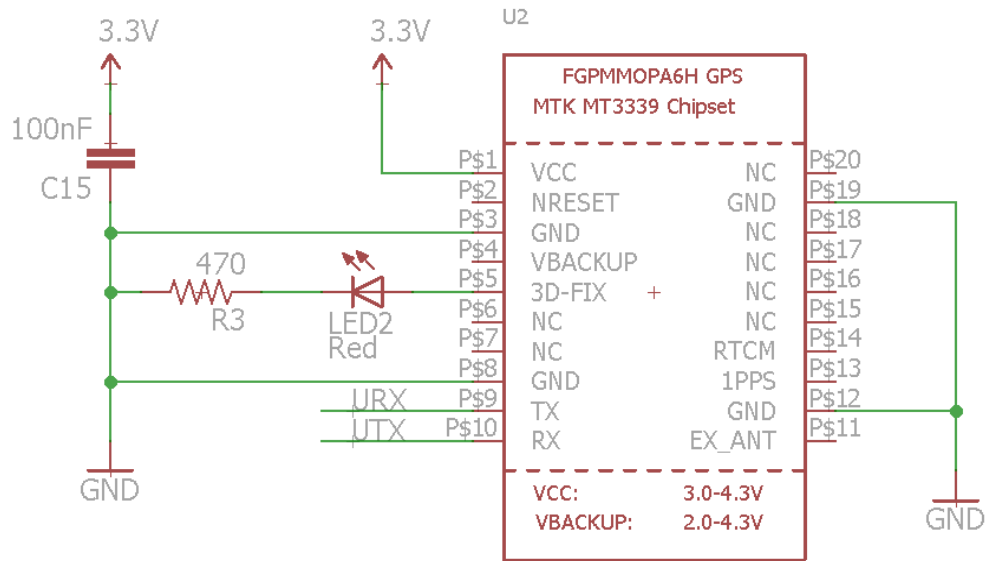
Alkuperäinen suunnitelma oli hyödyntää valmista näytön kehitysmoduulia, joka tulisi piikkirimalla päälevyn päälle. Tämä olisi vaatinut kuitenkin näytölle oman komponentin tekemistä itse. Valmiin komponentin myötä piti tehdä pieni tutkimus oikeaoppisesta kytkennästä, johon löysin lopulta toimivan ratkaisun. (pcd8544-based displays 2005.) (Ks. Kuvio 22.)



Kuvio 22. LCD-näytön kytkentäkaavio lopullisella levyllä

7.2.4 GPS-sirun kytkentä

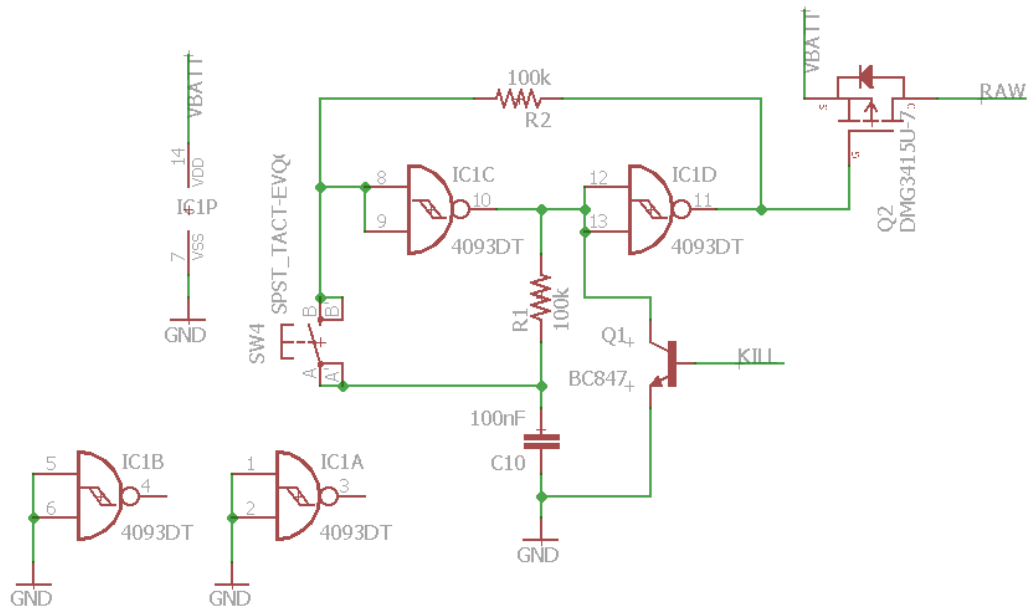
GPS-sirun kytkentään Adafruit tarjosi kytkentäkaavion. (GPS_FGPMMA6H_v0.3 2014.) Muutoksia tämä kytkentä vaati omaan projektiin huomattavasti. Päätin mennä mahdollisimman yksinkertaisella kytkennällä. Adafruitin kytkennästä poistin regulaattorin, piikkiriman, ulkoisen antennin ja kellopariston. Tarpeellisena pidin kytkennässä vain UART-linjan sekä FIX-LED:in. (Ks. Kuvio 23.)



Kuvio 23. GPS-piirin kytkentä lopullisella levyllä

7.2.5 Virtakytkimen kytkentä

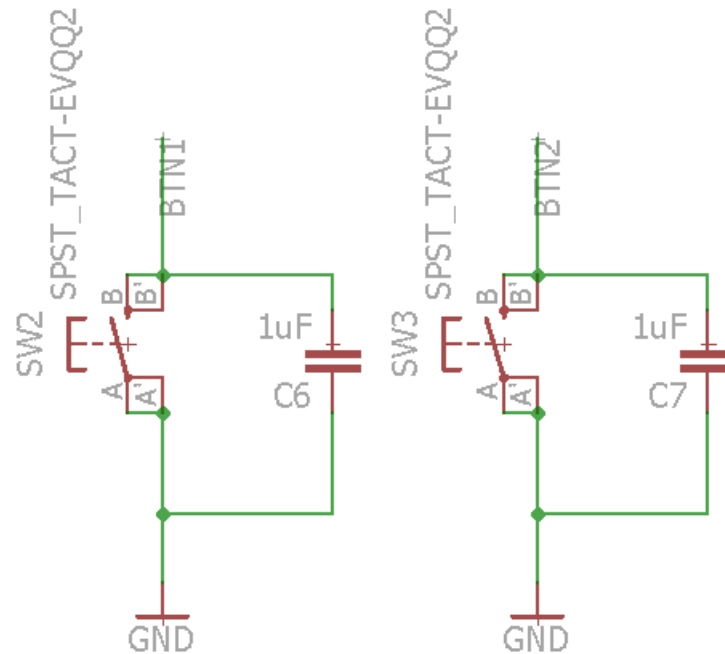
Adafruitin tapaan myös On/Off-kytkimestä oli tarjolla kytkentäkaavio. (Powerswitch v1 2013.) Tästä kytkennästä ei tarvinnut karsia muita komponentteja kuin alustan piikkirimat sekä LED. Virransyötössä oli jo valmiina virtaa indikoiva LED kytkettynä, eikä lopulliseen tuotokseen tarvittu kahta LED:iä näyttämään samaa asiaa. Alkuperäisessä kytkennässä oli SOT-23 koteloitu 2N2222 NPN-transistori, jota ei ollut enää yleisesti saatavilla kyseiseen koteloon. Tämä transistori piti vaihtaa toiseen vastaavaan NPN-tyyppiseen transistoriin. Päädyn valitsemaan korvaajaksi yleisen BC-sarjan pienitaajuustransistorin BC-847. (Johnsson 2004, 93) Myös virtanapin vaihdoin vakio-kokoiseksi 6 x 6 mm SMD-painonapiksi. (Ks. Kuvio 24.)



Kuvio 24. Virtakytkimen kytkentä lopullisella levyllä

7.2.6 Painikkeiden kytkentä

Painikkeiden kytkeminen oli hyvin suoraviivainen prosessi. Kytkentä toteutettiin samalla tavalla kuin testeissä oli toteutettu, paitsi debounce kondensaattorin vaihdoin $1\ \mu\text{F}$ kokoiseksi. Nappien toinen puoli kytkettiin GPIO-portteihin D2 ja D3. Painonapeiksi valitsin saman napin, minkä valitsin myös virtakytkimelle. (Ks. Kuvio 25.)



Kuvio 25. Painonappien kytkentä lopullisella levyllä.

7.2.7 Vikojen tarkistus

Kokonaisytkentä tuli valmistuessaan tarkistaa. (Liite 1.) Valmista levyä on vaikea muuttaa jälkikäteen ja lopputulos voi olla karu. Tämän takia kytkennän tarkastaminen on tärkein operaatio ennen kuin siirrytään levyn piirtämiseen ja mallintamiseen. Eaglessa on ominaisuus, joka tarkastaa vikakohtat ja epäselvyydet. Esimerkiksi jos jokin johdin ei ollut kytkentäkaaviolla kytketty komponenttiin tai johtimien yhdistämispiste puuttui. Eaglen-tarkistin ei kuitenkaan huomaa suunnittelijan tekemiä virheitä, kuten vääristä komponenteista tai väärästä kytkennästä.

7.3 BOM

Kun kytkentä oli valmis ja tarkastettu voitiin generoida BOM valmiista kytkennästä. (Liite 2.) Eaglessa on valmis työkalu tälle operaatiolle. BOM-dokumentin pohjalta voin etsiä tilattavat komponentit Farnell:ilta ja lisätä ne jokaisen komponentin perään

taulukkaan. Prosessi oli hyvin aikaa vievä, mutta tärkeä. Erilliskomponenttien valitseminen huolella voi säästää projektin hankintakuluja huomattavasti.

7.4 Levyn suunnittelu

Komponenttien valitsemisen ja tilaamisen jälkeen pystyin aloittamaan piirilevyn mallintamisen. Eaglessa piirilevytiedosto generoidaan kytkentäkaavion pohjalta. Piirilevyn mallintamisessa eagle osaa näyttää automaattisesti, mitkä kontaktit ovat yhteydessä toisiinsa. Tämän avulla tiedetään tehdä johdotukset piirilevyllä oikein. Suunnittelussa tuli erityisesti huomioida komponenttien asettelu, datalinjojen johdotus, maa-alueet ja virransyötön vaatimat suuremmat johdinleveydet.

7.4.1 Kerrokset ja rakenne

Piirilevyn suunnittelussa käytetään useita erityyppisiä kerroksia. Käyttämäni Eagle lisenssi tarjosi vain kaksi kuparikerrosta. Jouduin tyytymään niihin, mutta se ei ollut este vielä tämän kokoisessa työssä. Useammat kuparikerrokset lisäävät toki piirilevyn valmistuskustannuksia. Kuparikerrosten päälle tulee soldermask. Tämä kerros on eriste, joka estää juotettaessa tinan valumisen kontakteilta johtimiin. Soldermask myös määrittelee, minkä värinen piirilevy on. Toinen näkyvä kerros on silkscreen. Se on informaatiota antava tuloste soldermaskin päällä. Suurin suunnittelutyö tuli tapahtumaan levyn kuparikerroksilla ylä- ja alapuolella. Eaglessa komponentit osaavat huolehtia automaattisesti soldermaskin poistamisesta kontakteilta. Silkscreenin toteuttaminen on loppuvaiheen hienosäätöä.

7.4.2 Valmistajan etsiminen

Prototyypipiirilevyjen valmistajan valinta oli tässä vaiheessa suhteellisen tärkeää. Eri tehtaat kykenevät tekemään piirilevyjä tiettyjen sääntöjen mukaan ja tämä tuli ottaa huomioon levyä mallintaessa. Taulukossa 4 on listattu vaihtoehtoisia valmistajia piirilevyille. Valinnassa tuli ottaa huomioon, että piirilevy tulee tehdä näytön kiinnikkeiden takia 1 mm paksuiselle levyille. Tämä voi joissain tapauksissa olla kalliimpi kuin perinteinen 1.6 mm paksuinen piirilevy.

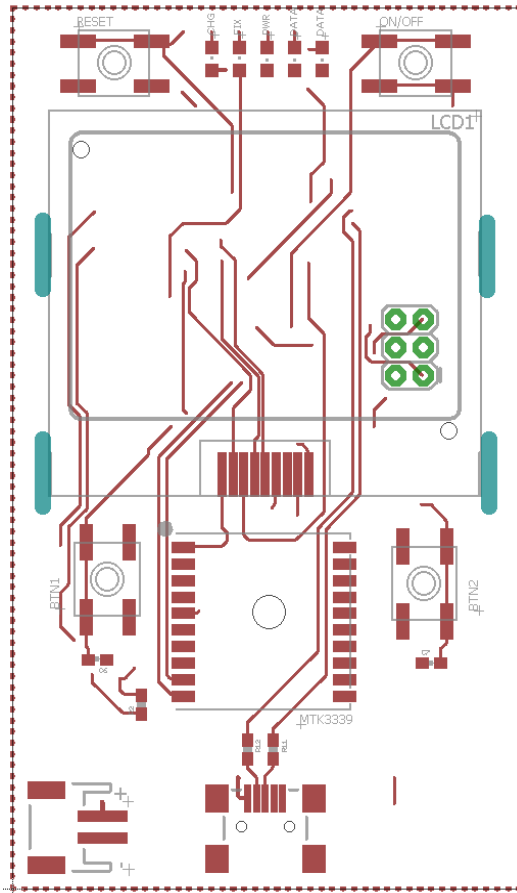
Taulukko 4. Lista edullisista piirilevynvalmistajista

Valmistaja	Plussat	Miinukset	Hinta-arvio
Seed PCB	-Hinta	-Toimitusaika	15€+toim
Dirty PCBs	-Mahdollisuus valita nopeampi toimitus	-Kalliimpi -Toimitusaika voi silti vaihdella	-23€ (halvempi toim) -37€ (nopeampi toim)

Hinnoittelun yhdenvertaistamiseksi taulukossa 4 valitut levynkustannukset sisältävät vähintään 5 piirilevyä, pinnoitteena valittu halvempi HASL ja soldermaskin väriksi valittu vihreä. Aikataulullisista syistä päätin mennä aavistuksen kalliimmalla Dirty PCB:llä.

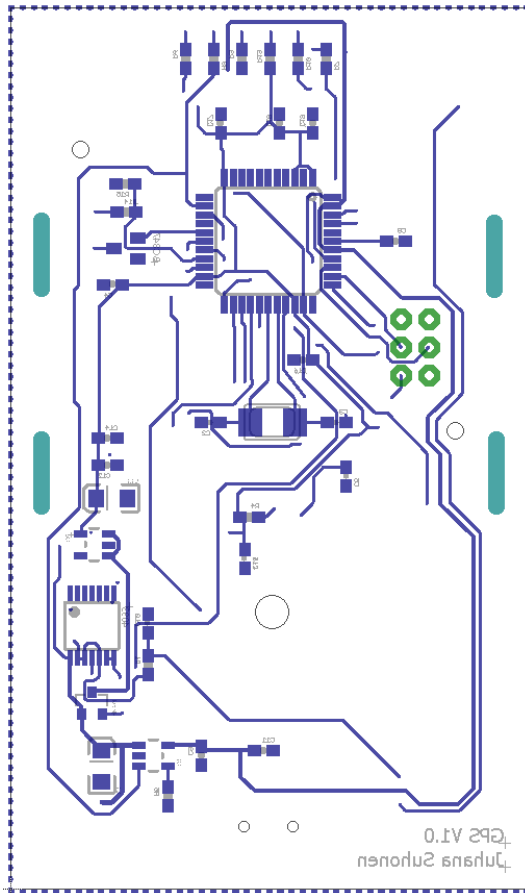
7.4.3 Komponenttien asettelu

Komponenttien asettelussa seurasin johdonmukaisesti uusittua hahmotelmaa levyn asettelusta kuviossa 21. Hahmotelmasta kuitenkin puuttui paikat akun liittimelle, virtanapille sekä reset-napille. Kuviosta 26 näkyy komponenttien asettelu piirilevyn päällä. Reset- ja virtanappi on sijoitettu näytön yläpuolelle levyn yläreunaan. Indikaatio-LED:it lataukselle, GPS:lle, virralle ja USB-datalle oli sijoitettu reset- ja virtanapin väliin. Näytön alapuolella on käyttäjäpainikkeet sekä GPS-piiri sijoiteltuna, kuten hahmotelmassa oli. Piirilevyn vasempaan alareunaan päätin sijoittaa akun liittimen. Siellä se olisi mahdollisimman vähän tiellä. GPS-piirin alapuolella on suunnitelman mukaisesti mini-USB liitin. LCD-näytön alapuolelle päätin sijoittaa ISP-ohjelmointiliittimen.



Kuvio 26. Komponenttiasettelu piirilevyn päällä

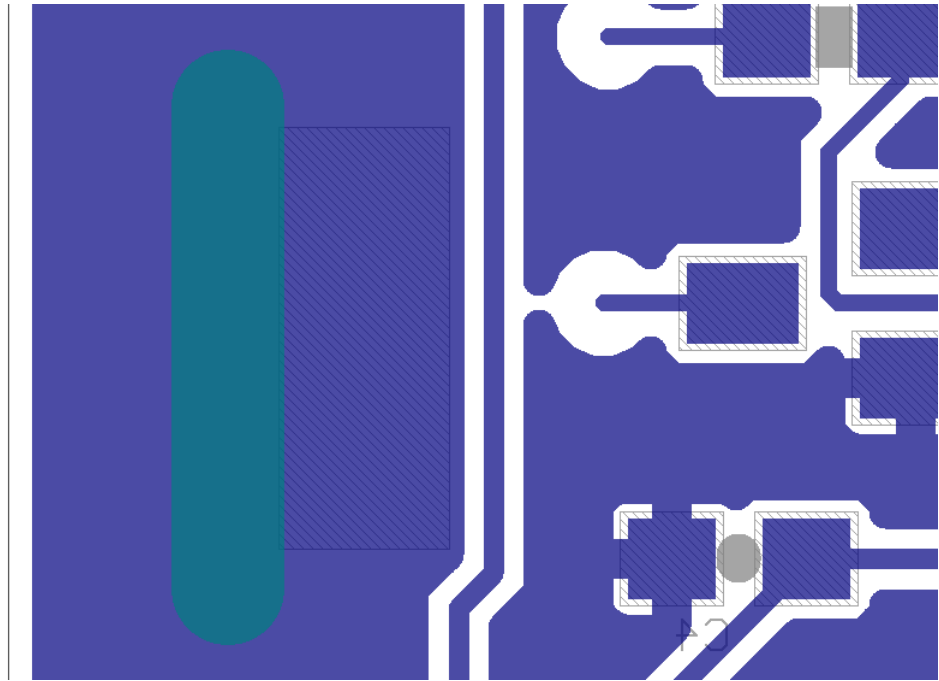
Piirilevyn alapuolelle sijoitin kaikki pienemmät komponentit. (Ks. Kuvio 27.) Alapuolelta löytyy MCU, kellokide, virtakytkimen NAND-piiri, regulaattori sekä Li-Po-latauspiiri. MCU on sijoitettuna piirilevyn alapuolella hahmotelman mukaisesti LCD-näytön vastakkaiselle puolelle. Indikaatio LED:ien alapuolella on sijoitettuna niiden sarjavastukset. Myöskin ISP-ohjelmointiliitin näkyy piirilevyn alapuolella, koska se on läpirei'itetty.



Kuvio 27. Komponenttiasettelu piirilevyn alapuolella

7.4.4 LCD-näytön vaatimat muutokset levyllä

LCD-näytön komponentti ei itsessään sisältänyt kiinnityksen toteutusta. Tätä varten piti tehdä käsin pieniä muutoksia piirilevyllä. Näytön metallisen kehikon reunoilla on kiinnikekorvat, jotka kiinnittyvät piirilevyn vastakkaiselle puolelle. Näille kiinnikkeille piti toteuttaa 1.2 mm leveät kolot, jotka jyrättäisiin piirilevyttä pois. Lisäksi näytön metallinen kehikko tuli saada maadoitettua, joten piirilevyn alapuolelta piti poistaa soldermask-kerrosta. Muutoksen avulla kiinnikkeet ottaisivat näin kontaktin maatasoon piirilevyn alapuolelta. (Ks. Kuvio 28.)



Kuvio 28. Yksi LCD-näytön kiinnikkeitä varten tehty muutos

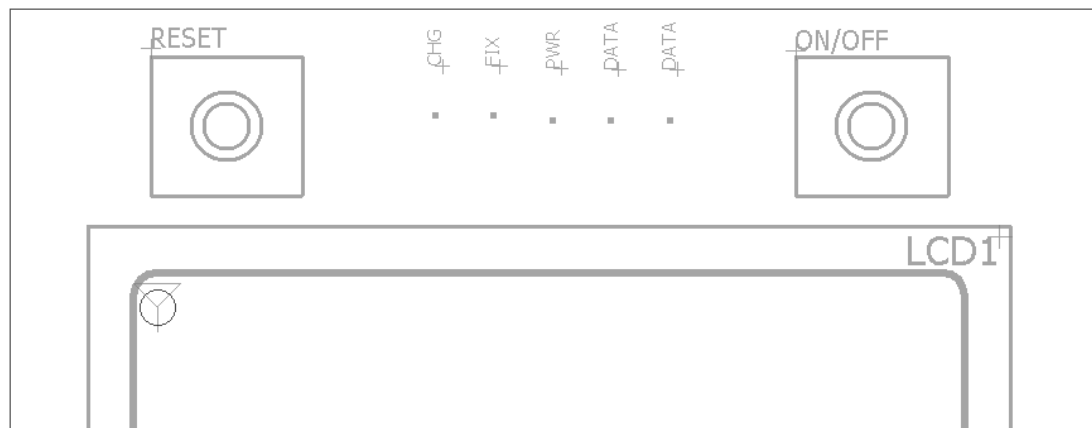
7.4.5 DRC

Jotta piirilevyn valmistaja voi valmistaa levyn, tulee sen olla suunniteltu heidän sääntöjensä mukaisesti. Nämä säännöt piti tarkistaa ennen Gerber tiedostojen tekemistä Eaglen omalla DRC-työkalulla. Dirty PCBs tarjosi Eaglelle oman DRC-tiedoston, jonka mukaan tarkastus tapahtui varmasti valmistajan ohjeiden mukaan. Tarkastuksen jälkeen tuli yleensä esille eristevälien ongelmat. Vastaavissa ongelmissa tuli vain johdottaa levyä uusiksi tai siirtää johtimia ja VIA-reikiä.

7.4.6 Silkscreen

Silkscreen koostui projektissa komponenttien nimistä, ulkoreunoista sekä omista lisäteksteistä. Valtaosa komponenteista asettaa vakiona komponentin nimen huonoon kohtaan ja yleensä toistensa päälle. Tätä varten eaglessa on SMASH-työkalu, joka irrottaa komponentin omiin pienempiin palasiin, joita voi nyt liikutella tai poistaa. Pällekkäin menevät tekstit tuli siirtää järkevämmiin, jotta silkscreenistä ei tullut sekava. Suurimmat muutokset tein piirilevyn yläreunaan indikaatio LED:eille ja pai-

nonapeille. Poistin jokaisen LED:in nimen ja vaihdoin tilalle oman tekstin, jossa lukee LED:in käyttökohde. Saman toimenpiteen toteutin myös piirilevyn jokaiselle painonapille. (Ks. Kuvio 29.)



Kuvio 29. Piirilevyn yläreunan silkscreen muutokset

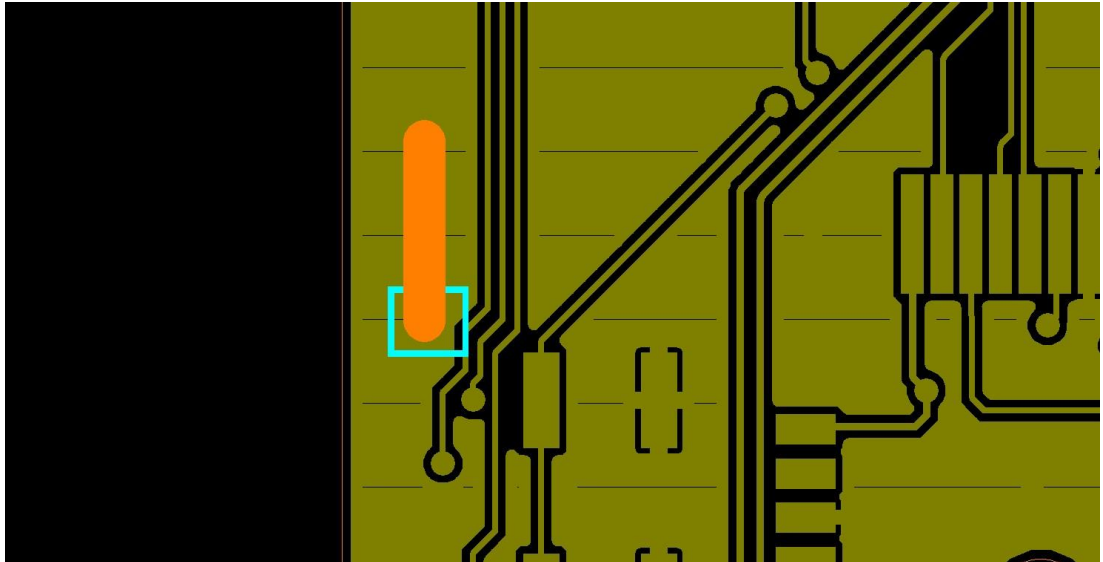
7.5 Valmistajan vaihtaminen

Aikataulullisista syistä päädyin vaihtamaan piirilevyn valmistajan aasialaisesta eurooppalaiseen valmistajaan. Näin piirilevy saataisiin nopeammin kasattavaksi lopulliseen kokoonpanoon. Vaihtaminen kuitenkin nosti valmistuskustannuksia moninkertaisesti edullisiin aasialaisiin verrattuna. Vertailussa oli mukana kaksi keskieuropplaista, kaksi suomalaista sekä yksi virolainen valmistaja, joista päädyin virolaiseen Brandneriin. Brandner toimitti piirilevyt nopeasti ja edullisemmin kuin suomalaiset valmistajat.

7.5.1 CAD-kuvien muutokset uudelle valmistajalle

Ensimmäiset toimittamani Gerber-tiedostot eivät kelvanneet valmistajalle, koska yksi johtimista oli liian lähellä LCD-näytön kiinnikkeelle tehtyä reikää. (Ks. kuvio 30.) Ongelma pystyttiin kiertämään vaihtamalla pinnoitemateriaali HASL:sta ENIG-

pinnoitteeseen tai siirtämällä johdinta kauemmaksi. Päätin kuitenkin mennä molemmilla vaihtoehdoilla, koska ENIG-pinnoite ei maksanut kamalasti, ja mahdollisti paremman lopputuloksen. (Liiva 2016.)

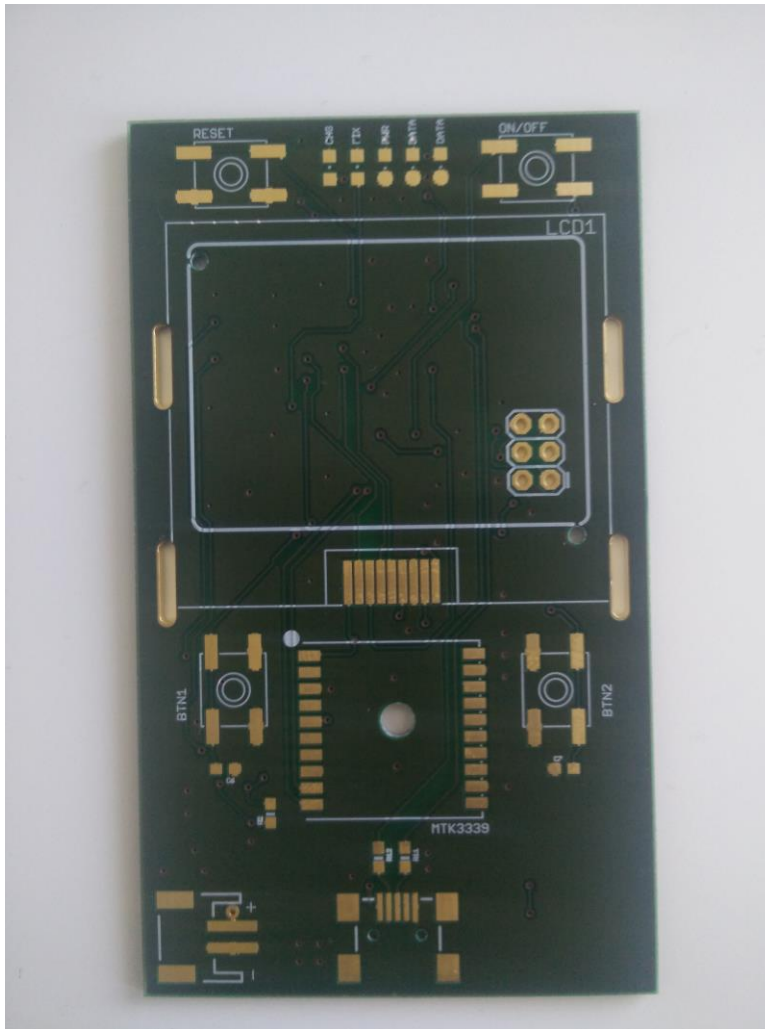


Kuvio 30. Valmistajan antama huomio piirilevyn Gerber-tiedostoissa.

8 Prototyypin kasaaminen

8.1 Kasausprosessi

Piirilevyn valmistuttua ja komponenttien saavuttua pystyin aloittamaan kokonaisuuden kasaamisen. (Ks. Kuvio 31.) Prosessiin tarvittiin juotoskolvi, juotostinaa, juoksetta, pinsetit ja mikroskooppi. Valtaosa komponenteista oli pieniä ja juotoksista tuli saada mahdollisimman siistejä. Lisäämällä juoksetta saadaan juotoksen lopputuloksesta huomattavasti parempi kuin pelkästään juotostinan omalla sisäisellä juoksetteella. Juottaminen aloitettiin pienistä komponenteista kuten esimerkiksi vastuksista ja kondensaattoreista. Lopuksi juotettiin isot ja tilaa vievät komponentit kuten napit ja GPS-piiri.



Kuvio 31. Brandnerin toimittama valmis piirilevy ENIG-pinnoitteella

8.2 Firmwaren kirjoitus

Arduino IDE:n kautta voi kirjoittaa MCU:lle bootloaderin käyttäen erilaisia tuettuja programmereita. Alkuperäinen tarkoitus oli kirjoittaa bootloader Arduino IDE:ä ja USBASP V2 programmeria käyttäen, mutta tällä yhdistelmällä tämä prosessi ei jostain syystä onnistunut. Arduino IDE:en integroitu avrduide ohjelmointityökalu ei onnistunut muuttamaan MCU:n ohjelmoitavia sulakkeita. Näitä sulakkeita muuttamalla voidaan esimerkiksi käskyttää MCU käyttämään ulkoista kellokidettä sisäisen resonattorin sijaan. Bootloaderin kirjoittamisprosessi onnistui vasta käyttämällä Atmel studiota sekä Atmelin omaa AVRISP MKII programmeria. Kun bootloader oli kirjoitettu

programmerilla ISP-väylää pitkin, voitiin ohjelmointityökalu vaihtaa takaisin Arduinoon. Bootloaderin avulla MCU:n firmware voidaan kirjoittaa USB-väylää pitkin. Kirjoittaminen aktivoidaan painamalla reset-nappia kahdesti lyhyen ajan sisällä. (Pro Micro & Fio V3 hookup Guide n.d.)

8.3 Ensimmäinen testaus

Levyä testasin käyttämällä samaa firmwarea, mikä oli viimeisenä käytössä ennen piirilevyn suunnitteluprosessia. Firmwaressa LCD-näytölle tulostettiin paikkatiedot, korkeus, akun jännite, automaattisen virrankatkaisun laskuri ja toisen käyttäjäpainikkeen tunnistava laskuri. (Ks. Kuvio 32.) GPS-paikannin lähti pyörimään samalla tavalla kuin koekytkentälevyllä. Paikkatiedot sekä korkeus olivat samoissa lukemissa kuin ennenkin. Vasemmanpuolimmainen painike nosti painikelaskurin arvoa yhdellä eikä bounce-efektiä ollut havaittavissa. 1 μ F kondensaattori bouncen suodattimena vaikutti oikealta. Akun jännite myös näytti oikein referoituna yleismittarilla saatuun tulokseen akun liittimestä. GPS-paikannin myös sammui 1 minuutin kuluttua. Tässä vaiheessa myös onnistuneesti testasin NMEA-ulostuloa USB:n kautta käyttämällä Arduino IDE:ssä olevaa sarjaporttimonitoria. Lisäksi kaikki indikaatio LED:it toimivat sekä Li-Po laturi. Näistä tuloksista voitiin todeta, että GPS-paikantimen piirilevy oli onnistunut.



Kuvio 32. Testausfirmware ladattu onnistuneesti GPS-paikantimeen

9 Ominaisuuksien laajentaminen

Pelkkä testaukseen tarkoitettu firmware ei riittänyt kattamaan kaikkia tavoiteltuja ominaisuuksia. Nykyinen firmware ei hyödyntänyt käyttäjäpainikkeita oikeastaan yhtään lisäksi GPS-paikannin sammui automaattisesti aina 1 minuutin kuluttua ensimmäisestä paikkatiedosta lähtien. GPS-paikantimen käyttöliittymää tuli parantaa, sekä ohjelmoida näkymään lisää paikannusdataa.

9.1 Käyttöliittymä

Käyttöliittymän rakenne perustui tilakoneeseen. (Rouse 2005.) Kahdella käyttäjäpainikkeella siirryttiin eteenpäin tai taaksepäin käyttöliittymän tilassa. Riippuen käyttöliittymän tilasta näytölle piirrettiin eri informaatiota paikannustiedoista.

Akun jännitettä ei enää näytetty jännitteenä vaan sekin rakennettiin pienen tilakoneen taakse. Näytön oikeaan reunaan piirrettiin akun varausta kuvaava palkki. Palkki tippuu, mitä tyhjempi akun varaus oli. Viimeisessä tilassa kun akun jännite oli alle 3.4 V ohjattiin GPS-paikannin sammuttamaan itsensä.

GPS-paikantimen automaattisen sammutuksen laskuri nollautui jokaisesta näppäinpainalluksesta. Näin laitetta voitiin käyttää pidemmän aikaa käynnistämättä laitetta aina uudestaan.

9.2 Näytettävät paikannustiedot

Ensimmäisessä käyttöliittymän tilassa näytölle piirrettiin testausfirmwaresta tutut paikkatiedot sekä korkeus. Seuraavassa tilassa näkymään vaihtui pelkästään nopeuden näyttäminen. Nopeuden yksiköksi valitsin tinyGPS kirjaston valikoimasta metriä sekunnissa. Kirjasto tarjosi myös kaikki muut pääsääntöiset nopeuden yksiköt käytettäväksi. Kolmannessa tilassa näytettiin yhteydessä olevien satelliittien lukumäärä.

9.3 Laajennettujen ominaisuuksien testaaminen

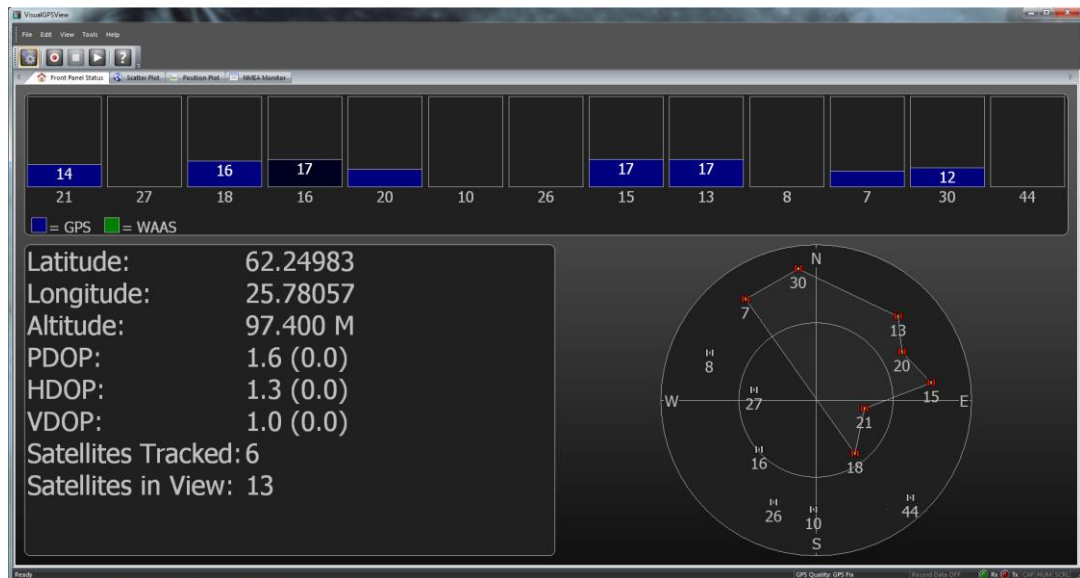
Paikkatiedot sekä korkeuden näyttäminen toimivat kuten ennenkin. Laajennettujen ominaisuuksien testaaminen tapahtui osittain eri firmware versioiden välillä. Painikkeet vaihtavat käyttöliittymän tilaa oikein ja kiertävät omassa silmukassaan. Satelliittien lukumäärää en pystynyt tarkasti omilla työkaluillani testaamaan ja joudun luottamaan GPS-paikantimen antamaan tulokseen. (Ks. Kuvio 33.)



Kuvio 33. Päivitetty käyttöliittymä GPS-paikantimella

Nopeusnäytön testaaminen tapahtui avustuksella. Testissä ajettiin autolla vakionopeudella 40 km/h ja vertaamalla vakionopeutta GPS-paikantimen antamaan arvoon. GPS-paikannin ilmoitti nopeudeksi 11 - 12 m/s testin aikana, joka on noin 40 km/h.

NMEA-ulostuloa testasin vielä lopullisessa toteutuksessa ulkopuolisella ohjelmistolla. Ohjelmistona toimi VisualGPSView. Tietokoneella GPS-paikannin näkyi COM-porttina, joka valittiin GPS-laitteeksi VisualGPSViewissä. Tämäkin testi sujui ongelmitta. Tulokset olivat identtiset GPS-paikantimen näytöllä kuin visualGPSView ohjelmassa. Kuviossa 34 näkyy VisualGPSViewin käyttöliittymä, joka näyttää nykyiset koordinaatit, sekä satelliittien tiedot suoraan GPS-paikantimelta tulevilta NMEA-lauseista.



Kuvio 34. VisualGPSView testaus GPS-paikantimen kanssa

10 Tulevaisuuden optiot

TinyGPS-kirjasto tarjoaa laajalti ominaisuuksia, joita saadaan irti GPS:n NMEA-datasta. Tulevaisuudessa mahdollisia toteutuksia kirjaston avulla voi olla esimerkiksi EEPROM:ille kirjoitettujen kohdekoordinaatteihin paikantaminen. Etäisyyden mittaaminen kahden koordinaatin välillä on myöskin mahdollista toteuttaa. MTK3339 GPS-piirillä on mahdollista ottaa käyttöön reittipisteiden tallentaminen. Kuitenkin yksi suurimmista puutteista työssä on vielä koteloinnin uupuminen. Tämä vaatisi 3D-mallintamista ja 3D-tulostamista. Ominaisuuksia voi laajentaa vielä paljon, sillä nykyinen firmware vie tilaa ohjelmamuistilta alle 65 %.

11 Arvio lopputuloksesta

Lopputulos oli odotettua parempi. Piirilevy onnistui kytkennältään heti ensimmäisellä yrittämällä. Usein vastaavissa projekteissa ensimmäinen piirilevy ei välttämättä ole se lopullinen, vaan erheitä voi tapahtua paljon. Seuraavaan mahdolliseen piirilevyversioon tekisin kuitenkin vielä parannuksia. Esimerkiksi LCD-näytön kontaktit tulevat näytön alta esille niin, että ne melkein koskettavat näytön metallista runkoa.

Tästä syystä lyhentäisin hieman näytön kontakteja piirilevyllä. Oikeanpuoleinen painonappi on jäänyt aivan aavistuksen alemmaksi kuin vasemmanpuoleinen painonappi. Tämän kuitenkin pystyy nykyisellä levyllä kiertämään juottamalla oikeanpuoleisen painonapin hiukan ylemmäksi. Kaiken lisäksi yksi virransyötön johdin menee hyvin läheltä MCU:n yhtä maapinniä. Siinä on pieni oikosulun vaara.

Suurin takaisku projektissa oli alkuperäisen kehitysalustan kyvyttömyys hallita GPS-piiriltä tulevia NMEA-lauseita. Tämän takia meni aikaa uusien kehitysalustojen etsimiseen sekä ohjelmamuutoksia uuden alustan käyttöönottoon. Uusi alusta kuitenkin saattoi edistää virransyötön sekä virtakytkimen kehitystä valmiilla kytkennällä.

Ohjelmallisesti erheitä ei suurempia ollut poislukien akun jännitteen mittaamiseen tullut kirjoitusvirhe. Lopputulokseen olen erittäin tyytyväinen. Perustoiminnallisuus toimii kuten pitääkin ja toiminnallisuutta pystyy hiomaan ja lisäämään reilusti jälkikäteenkin.

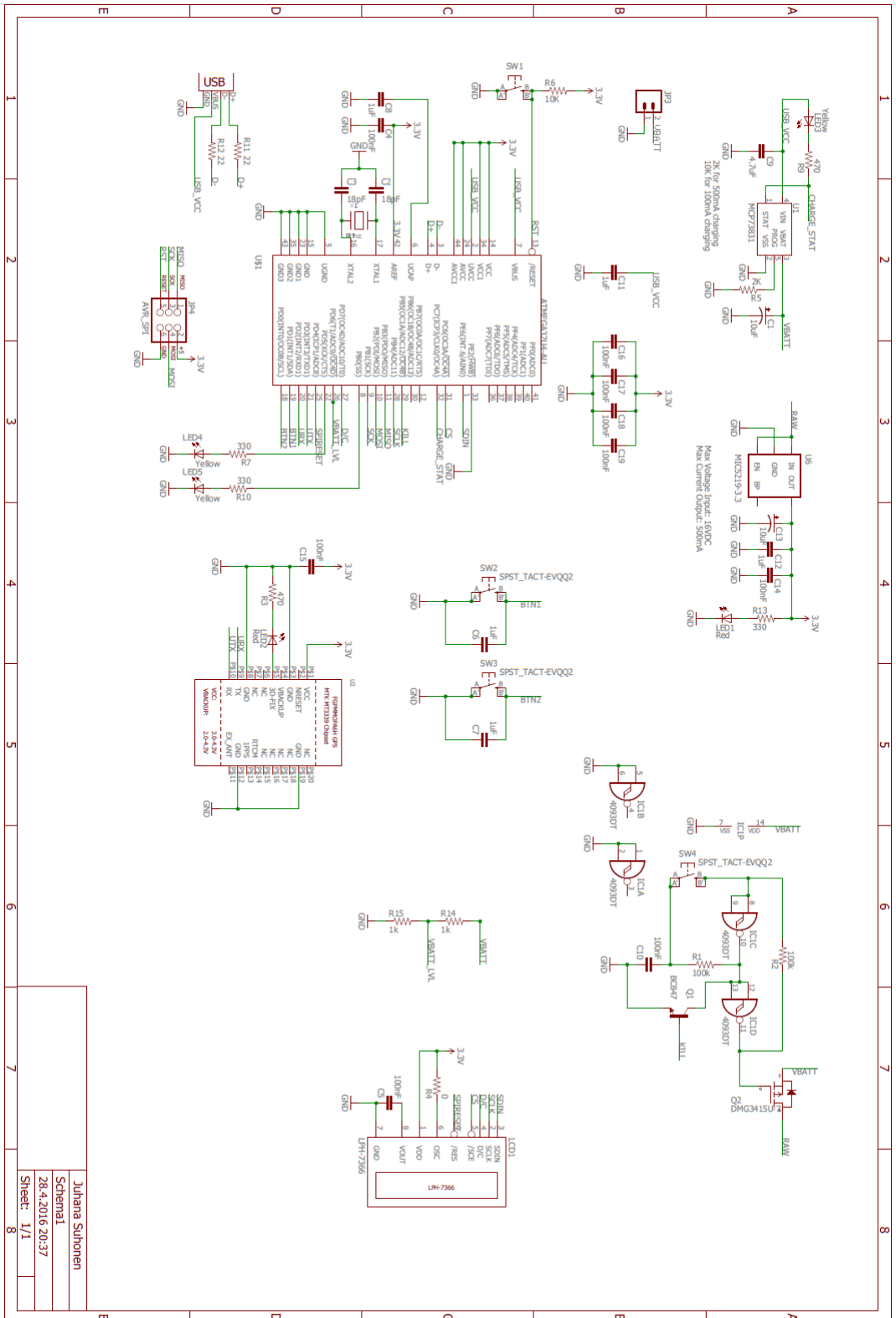
Lähteet

- Adafruit-GFX-Library. N.d. Grafiikkakirjaston lähdekoodi Github sivustolla. Viitattu 2.12.2015. <https://github.com/adafruit/Adafruit-GFX-Library>.
- Adafruit_GPS. N.d. Adafruit GPS-kirjaston lähdekoodi Github sivustolla. Viitattu 1.12.2015. https://github.com/adafruit/Adafruit_GPS.
- Adafruit-PCD8544-Nokia-5110-LCD-library. N.d. Nokia 5110:n LCD-näytön kirjaston lähdekoodi Github sivustolla. Viitattu 2.12.2015. <https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library>.
- Adafruit Push-button Power Switch Breakout. 2013. Virtakytkin Adafruit sivustolla. Viitattu 15.12.2015. <https://www.adafruit.com/product/1400>.
- Ahoranta, J. 2006. Sähkötekniikka. 8.p. Helsinki: WSOY.
- Arduino-Fio_v31. 2014. Fio v3 kehitysalustan kytkentäkaavio. Viitattu 1.3.2016. https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/Arduino-Fio_v31.pdf.
- Arduino Products. N.d. Tuoteluettelo Arduino sivustolla. Viitattu 13.11.2015. <https://www.arduino.cc/en/Main/Products>.
- Arduino Software. 2015. Arduino IDE:n esittely Arduino-sivustolla. Viitattu 1.10.2015. <https://www.arduino.cc/en/Guide/Environment#toc5>.
- ATmega16U4/ATmega32U4. 2015. ATmega 32U4 datalehti. Viitattu 10.3.2016. http://www.atmel.com/images/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf.
- Digital Pins. N.d. Digitaalisten porttien esittely Arduino sivustolla. Viitattu 5.3.2016. <https://www.arduino.cc/en/Tutorial/DigitalPins>.
- Digital Pins With Interrupts. N.d. Keskeytysten opas Arduino sivustolla. Viitattu 11.3.2016. <https://www.arduino.cc/en/Reference/AttachInterrupt>.
- EEVblog #176 - Lithium Ion/Polymer Battery Charging Tutorial. 2011. Videoblogi Youtube-sivustolla. Viitattu 7.12.2015. https://youtu.be/A6mKd5_-abk.
- Fio v3 - ATmega32U4. N.d. Fio v3 kehitysalustan kuvaus Sparkfun sivustolla. Viitattu 1.2.2016. <https://www.sparkfun.com/products/11520>.
- Gammon, N. 2013. Switches tutorial. Painonappien kytkentäohje Arduino kehitysalustaan Gammon sivustolla. Viitattu 10.3.2016. <http://www.gammon.com.au/switches>.
- Ganssle, J. 2015. How to Become an Embedded Systems Geek. Esittely sulautettujen järjestelmien kehittämisestä The Ganssle Group sivustolla. Viitattu 15.12.2015. <http://www.ganssle.com/startinges.htm>.

- GlobalTop PMTK command packet. 2012. AT-komennot GPS-piirille. Viitattu 14.1.2016. https://cdn-shop.adafruit.com/datasheets/PMTK_A11.pdf.
- GPS_FGPMMPA6H_v0.3. 2014. GPS-piirin kytkentäkaavio. Viitattu 5.4.2016. <https://cdn-learn.adafruit.com/assets/assets/000/022/494/original/gpssch.png?1421536475>.
- GPS Educational Poster. 2013. Juliste GPS.gov sivustolla. Viitattu 8.11.2015. <http://www.gps.gov/multimedia/poster/>.
- Graphic LCD Hookup Guide. N.d. Nokia 5110:n LCD-näytön opas Sparkfun sivustolla. Viitattu 2.12.2015. <https://learn.sparkfun.com/tutorials/graphic-lcd-hookup-guide>.
- Hart, M. 2013. TinyGPS. TinyGPS kirjasto sivustolla Arduiniana. Viitattu 20.1.2016. <http://arduiniana.org/libraries/tinygps/>.
- Johnsson, B. 2004. Elektroniikka. Iisalmi: IS-VET.
- Liiva, R. 2016. Greetings from Brandner PCB. Sähköpostiviesti 28.4.2016. Vastaanottaja J.Suhonen. Keskustelua tarvittavista muutoksista piirilevyille.
- lph-7366.lbr. 2007. LPH-7366 LCD-näytön Eagle komponentti CadSoft USA sivustolla. Viitattu 1.4.2016. <http://www.cadsoftusa.com/downloads/file/lph-7366.lbr>.
- Micrel MIC5205. 2006. Microchip regulaattori datalehti. Viitattu 7.12.2015. <http://ww1.microchip.com/downloads/en/DeviceDoc/mic5205.pdf>.
- pcd8544-based Displays. 2005. PCD8544 pohjaisten LCD-näyttöjen kytkentäohje Sourceforge sivustolla. Viitattu 2.4.2016. <http://serdisplib.sourceforge.net/ser/pcd8544.html#preface>.
- Powerswitch v1. 2013. Virtakytkimen kytkentäkaavio. Viitattu 7.4.2016. <https://cdn-shop.adafruit.com/datasheets/1400sch.png>.
- Pro Micro & Fio V3 Hookup Guide. N.d. Fio v3 kehitysalustan käyttöönotto-ohje Sparkfun sivustolla. Viitattu 12.2.2016. <https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide>.
- Rouse, M. 2005. State machine. Määritelmä termistä tilakone sivustolla whatis. Viitattu 12.5.2016. <http://whatis.techtarget.com/definition/state-machine>.
- SparkFun USB to serial UART Boards Hookup Guide. N.d. FTDI-piirin käyttöohje Sparkfun sivustolla. Viitattu 10.1.2016. https://learn.sparkfun.com/tutorials/sparkfun-usb-to-serial-uart-boards-hookup-guide?_ga=1.94799900.120289709.1456849755.
- Ultimate GPS breakout: Problems with Arduino Fio. 2012. Foorumi adafruitin GPS-kytkentälevystä. Viitattu 12.1.2016. <https://forums.adafruit.com/viewtopic.php?f=19&t=28438>.
- What is EAGLE. 2011. EAGLE suunnitteluohjelmiston esittely CadSoft USA sivustolla. Viitattu 25.3.2016. <http://www.cadsoftusa.com/eagle-pcb-design-software/about-eagle/>.

Liitteet

Liite 1. Kytentäkaavio



Juhana Sihonen
 Schemati
 28.4.2016 20:37
 Sheet: 1/1

Liite 2. BOM

Part	Value	Device	Package	Description	Farnell
C1	10uF	CAP_POL 1206	EIA3216	Capacitor Polarized	x
C2	18pF	CAP0603- CAP	0603-CAP	Capacitor	x
C3	18pF	CAP0603- CAP	0603-CAP	Capacitor	x
C4	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C5	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C6	1uF	CAP0603- CAP	0603-CAP	Capacitor	x
C7	1uF	CAP0603- CAP	0603-CAP	Capacitor	x
C8	1uF	CAP0603- CAP	0603-CAP	Capacitor	x
C9	4.7uF	CAP0603- CAP	0603-CAP	Capacitor	x
C10	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C11	1uF	CAP0603- CAP	0603-CAP	Capacitor	x
C12	1uF	CAP0603- CAP	0603-CAP	Capacitor	x
C13	10uF	CAP_POL 1206	EIA3216	Capacitor Polarized	x
C14	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C15	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C16	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C17	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C18	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
C19	100nF	CAP0603- CAP	0603-CAP	Capacitor	x
IC1	4093DT	4093DT	MC14093 B- TSSOP14	Quad 2- input NAND schmitt trigger	x

JP3		M02-JST- 2MM-SMT AVR_SPI_ PRG_6PT	JST-2- SMD	Header 2	x
JP4	AVR_SPI	H	2X3	AVR ISP 6 Pin	
LCD1	LPH-7366	LPH-7366	LPH-7366		x
LED1	Red	LEDCHIP LED_0805	CHIPLD _0805	LED	x
LED2	Red	LEDCHIP LED_0805	CHIPLD _0805	LED	x
LED3	Yellow	LEDCHIP LED_0805	CHIPLD _0805	LED	x
LED4	Yellow	LEDCHIP LED_0805	CHIPLD _0805	LED	x
LED5	Yellow	LEDCHIP LED_0805	CHIPLD _0805	LED NPN TRANSIS	x
Q1	BC847	BC847	SOT23	TOR	x
Q2	DMG3415	P- CHANNEL		P-Channel MOSFET Mode Field Effect Transistor	x
R1	100k	RESISTOF	0603-RES	Resistor	x
R2	100k	RESISTOF	0603-RES	Resistor	x
R3	470	RESISTOF	0603-RES	Resistor	x
R4	0	RESISTOF	0603-RES	Resistor	x
R5	2K	RESISTOF	0603-RES	Resistor	x
R6	10K	RESISTOF	0603-RES	Resistor	x
R7	330	RESISTOF	0603-RES	Resistor	x
R9	470	RESISTOF	0603-RES	Resistor	x
R10	330	RESISTOF	0603-RES	Resistor	x
R11	22	RESISTOF	0603-RES	Resistor	x
R12	22	RESISTOF	0603-RES	Resistor	x
R13	330	RESISTOF	0603-RES	Resistor	x
R14	1k	RESISTOF	0603-RES	Resistor	x
R15	1k	RESISTOF	0603-RES	Resistor	x

SW1	SPST_TA CT- EVQQ2	SPST_TA CT- EVQQ2	EVQ-Q2	SMT 6mm switch, EVQQ2 series	x
SW2	SPST_TA CT- EVQQ2	SPST_TA CT- EVQQ2	EVQ-Q2	SMT 6mm switch, EVQQ2 series	x
SW3	SPST_TA CT- EVQQ2	SPST_TA CT- EVQQ2	EVQ-Q2	SMT 6mm switch, EVQQ2 series	x
SW4	SPST_TA CT- EVQQ2	SPST_TA CT- EVQQ2	EVQ-Q2	SMT 6mm switch, EVQQ2 series	x
U\$1	ATMEGA 32U4-AU	ATMEGA 32U4-AU	TQFP44	http://www .ladyada.n et/library/e agle Microchip' s	x
U1	MCP7383 1	MCP7383 1	SOT23-5	MCP7383 1	x
U2		GPS_FGP MMOPA6 H	FGPMMO PA6H	GPS Module - MTK MT3339 Chipset, - 165dBm sensitivity, 22 channels, 10Hz, auto- select external antenna options	x
U6	MIC5219- 3.3	V_REG_L DOSMD	SOT23-5	Voltage Regulator LDO	x
X1		XTALNX5 032	NX5032	Low cost SMT crystals, no capacitors included	x
X2	USBSMD	USBSMD	USB- MINIB	USB Connector s	x