

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Thomas Johann Seebeki elektroonikainstituut

IEE40LT

Rauno Lainemäe 120968 IAEB

**GLOBALSEL
SATELLITNAVIGATSIOONISÜSTEEMIL
BASEERUV SPORDIKOMPUUTER**

Bakalaureusetöö

Juhendaja: Reeno Reeder
doktorikraad
lektor

Tallinn 2016

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Rauno Lainemäe

06.06.2016

Annotatsioon

Selle bakalaureusetöö eesmärk on luua globaalsel satelliitnavigatsioonisüsteemil baseeruv spordikompuuter. Spordikompuuter on mõeldud kasutamiseks sporditreeningutel ja –võistlustel (eelkõige jalgrattaspordis), et näha oma saavutusi displeil ning salvestada need mälukaardile. Töö sisaldab loodavale seadmele sobivate komponentide välja valimist, elektriskeemi koostamist ning trükkplaadi disainimist.

Töö tulemuseks valmis spordikompuuter, mille põhikomponentideks on Atmel ATmega328P mikrokontroller [1], u-blox NEO-M8N GNSS moodul [2], Heltec 1.3“ OLED displei [3] ning mis saab toite laetavast Li-ioon akust. Spordikompuutri elektroonilise osa maksumuseks kujunes ümardatult 34€.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 35 leheküljel, 4 peatükki, 16 joonist, 2 tabelit.

Abstract

Sports Computer Based on Global Navigation Satellite System

The aim of this thesis is to develop a sports computer based on GNSS, which is used for sports training and competitions (primarily in cycling) to get feedback about users progress during and after the training. This development consists of picking the suitable components, composing the electrical schematic and designing a PCB.

The result of this thesis is a sports computer, which contains Atmel ATmega328P [1] microcontroller, u-blox NEO-M8N [2] GNSS module, Heltec 1.3“ OLED display [3] and rechargeable Li-ion battery as a power supply.

This device receives signals from satellites with GNSS module, calculates its position and sends necessary data to the microcontroller. The microcontroller is used to send the data to display driver and also to save the data to memory card. Display driver shows the data on the OLED display.

The usage of this device is broad as it can be used in different sports activities (for example cycling, running, skiing) and various other applications that require speed and/or altitude measurement and precise timing (for example car speedometer, digital clock).

Compared to alternative products this sports computer is capable of receiving signals from wider range of satellites, has a display that can be read in any ambient light condition, expected battery life is over 12 hours and the cost of the electronics is relatively low 34€.

The thesis is in Estonian language and contains 35 pages of text, 4 chapters, 16 figures, 2 tables.

Lühendite sõnastik

GNSS	<i>Global Navigation Satellite System</i> , globaalne satelliitnavigatsioonisüsteem
OLED	<i>Organic light-emitting diode</i> , orgaaniline valgusdiod
PCB	<i>Printed circuit board</i> , trükkplaat
LDO	<i>Low-dropout regulator</i> , pingeregulaator
USB	<i>Universal Serial Bus</i> , universaalne jadasiin
ADC	<i>Analog-to-digital converter</i> , analoog-digitaalmuundur
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i> , Elektriliselt eemaldatav programmeeritav lugemiskaitstud mälu
RAM	<i>Random-access memory</i> , suvapöördusmälu
PWM	<i>Pulse-width modulation</i> , pulsilaiusmodulatsioon
LED	<i>Light-emitting diode</i> , valgusdiod
EMI	<i>Electromagnetic interference</i> , Elektromagneetiline interferents
RTC	<i>Real-time clock</i> , reaajakell
Li	<i>Lithium</i> , liitium – keemiline element
I ² C	<i>Inter-Integrated Circuit</i> , jadaliidese suhtlusprotokoll
SPI	<i>Serial Peripheral Interface</i> , jadaliidese suhtlusprotokoll
USART	<i>Universal Synchronous/Asynchronous Receiver/Transmitter</i> , universaalne sünkroonne/asünkroonne saatja/vastuvõtja
TX	<i>Transmitter</i> , saatja, saatja liin
RX	<i>Receiver</i> , vastuvõtja, vastuvõtja liin
SD	<i>Secure Digital</i> , väikmälukaardi formaat
IDE	<i>Integrated development environment</i> , integreeritud arenduskeskkond

Sisukord

1 Sissejuhatus	10
2 Tehnoloogia valik	12
2.1 Kasutatud komponendid	12
2.1.1 Mikrokontroller ATmega328P	12
2.1.2 GNSS Moodul u-blox NEO-M8N	13
2.1.3 Cirocomm GNSS antenn	14
2.1.4 Heltec 1,3" OLED displei	15
2.1.5 Top Power TP4056 laadija	16
2.1.6 Seadme toiteahel	16
2.2 Kommunikatsioon	18
2.2.1 SPI	18
2.2.2 I ² C	18
2.2.3 USART	19
2.3 Programmeerimine	19
2.3.1 Mikrokontroller	19
2.3.2 GNSS moodul	20
2.4 Prototüüpseade	20
2.4.1 Prototüüpseadme moodulid	20
2.4.2 Prototüüpseadme testimine	23
3 Elektriskeem ja trükkplaat	24
3.1 Elektriskeemi koostamine	24
3.1.1 Mikrokontroller	24
3.1.2 GNSS moodul	25
3.1.3 Pingeregulaator	27
3.1.4 Aku laadimine	27
3.1.5 Mälukaardi ühendamise	28
3.1.6 Pistikud ja lülitid	29
3.2 Trükkplaadi disain	30
4 Kokkuvõte	32

Kasutatud kirjandus	33
Lisa 1 – Spordikompuutri olekumudel	36
Lisa 2 – Heltec OLED displei elektriskeem	37
Lisa 3 – Prototüüpseadme tarkvara lähtekood.....	38

Jooniste loetelu

Joonis 2.1 u-blox GNSS moodulite seeriad [15].	13
Joonis 2.2 u-blox GNSS moodulite omaduste võrdlus[15].	14
Joonis 2.3 Liitiumaku pinge ja mahtuvuse graafik tühjakslaadimisel [20].	17
Joonis 2.4 Arduino Pro Mini prototüüpplaat [28].	21
Joonis 2.5 Korpuse ja ühendusjuhtmetega GNSS moodul prototüüpmodeli jaoks.	21
Joonis 2.6 Laadimismoodul TP4056 baasil.	22
Joonis 2.7 Heltec OLED displei[3].	22
Joonis 2.8 LC Technology microSD mälukaardi ühendamise moodul [30].	23
Joonis 2.9 Spordikompuutri prototüüpmodul.	24
Joonis 3.1 Mikrokontrolleri skeem.	25
Joonis 3.2 GNSS mooduli skeem.	26
Joonis 3.3 Pingeregulaatori skeem.	27
Joonis 3.4 Aku laadimiseks vajalik skeem.	28
Joonis 3.5 Mälukaardi pesa skeem.	28
Joonis 3.6 Pistikute ja lülitite skeem.	29
Joonis 3.7 Spordikompuutri trükkplaadi disain programmis CadSoft Eagle.	31

Tabelite loetelu

Tabel 2.1 Atmel ATmega328P mikrokontrolleri põhiandmed.....	12
Tabel 2.2 Põhikomponentide keskmine teoreetiline voolutarve [1], [2], [20].	17

1 Sissejuhatus

Töö teema arenes välja vajadusest saada sporditreeningul, eelkõige rattasõidul, infot oma liikumiskiiruse ja läbitud distantsi kohta ning analüüsida tulemusi pärast treeningut. Turul leidub mitmeid alternatiivseid tooteid, mis omavad selliseid funktsioone, kuid nendel leidub negatiivseid külgi.

Treeningtulemuste levinumateks mõõteviisideks võib lugeda traditsioonilist keelreleed ja magnetit kasutavat rattaspidomeetrit ning kiirendusanduril põhinevat jalaandurit. Esimene variant on küllaltki usaldusväärne, kuid seda saab kasutada vaid rattaga sõites ning vajab seadistamist vastavalt rattale. Lisaks koosneb see mitmest osast, mis on vaja ratta külge kinnitada ning see teeb paigalduse ja kasutamise ebamugavamaks. Jalaandurit saab kasutada vaid kõndimisel ja joostes, seega on ka selle kasutusala vägagi piiratud. Jalaandur vajab enne kasutamist kalibreerimist, et mõõtetulemus oleks võimalikult täpne. Sellegipoolest jääb jalaanduri tulemuste täpsus tavaliselt alla GNSS-il põhineva kompuutri tulemuste täpsusele, eriti pikemate distantside läbimisel. Mõõtetulemuste nägemiseks vajab jalaandur lisaks näiteks mobiiltelefoni või spordikella ehk ka see mõõteviis koosneb mitmest osast.

GNSS-põhine spordikompuuter omab mitmeid eeliseid alternatiivsete mõõteviiside ees, mis on ka põhjuseks, miks sellised seadmed on üha rohkem sporditegijate seas levinud. Suurimaks eeliseks alternatiivide ees on kasutusmugavus: mõõtetulemuste saamiseks on kõik vajalik ühes väiksemõõtmelises korpuses ning seade ei vaja eelnevat kasutajapoolset seadistamist. Lisaks on sellise seadme kasutusala väga lai, kuid seda ei saa enamasti kasutada siseruumides või muudes kehva GNSS signaaliga kohtades (näiteks suurlinnad, tunnelid, tihedad metsad).

Tänapäeval on turul saadaval lai valik satelliitsüsteemil põhinevaid spordikompuutreid. Põhiliseks miinuseks nende puhul on seadme kõrge hind. Lisaks sellele võtavad paljud seadmed vastu vaid GPS satelliitsignaale ehk väiksem valik signaale võrreldes GPS/Galileo/GLONASS vastuvõtjatega ning tõenäolisem positsiooni fikseeringu

kaotamine kehvema signaaliga kohtades. Oluliseks aspektiks on ka seadme displei, mis paljudel spordikompuutritel pole ereda päikese käes või ööpimeduses loetav.

Lõputöös keskendutakse satelliitsüsteemil baseeruva spordikompuutri nõudmiste sõnastamisele, tehnoloogia valikule ning elektriskeemi ja trükkplaadi disainile.

2 Tehnoloogia valik

Antud peatükk annab ülevaate seadmes kasutatavatest komponentidest, tutvustab nende funktsioone ning põhjendab nende valikuid. Valitud põhikomponentideks on ATmega328P mikrokontroller [1], u-blox NEO-M8N GNSS moodul [2], Cirotech GNSS antenn [4], Heltec 1.3" OLED displei [3], Top Power TP4056 [5] laadija, Microchip MCP1700 [6] LDO ning 1.0Ah Li-ioon aku. Lisaks selgitab peatükk komponentide omavahelist suhtlemist, mikrokontrolleri ja GNSS mooduli programmeerimisviise ning tutvustab prototüüpmodelit.

2.1 Kasutatud komponendid

2.1.1 Mikrokontroller ATmega328P

Seadme keskmeks ehk mikrokontrolleriks on valitud Atmel ATmega328P. Peamiseks valiku põhjuseks võib lugeda seda, et antud mikrokontroller on kasutusel prototüüpplatvormil nimega Arduino [7], millel on suurepärane kasutajatugi ning avatud lähtekoodiga riistvara- ja tarkvaraplatvorm. Tänu sellele on seda mikrokontrollerit lihtne programmeerida ning kasutada prototüüpimisel.

Tabel 2.1 Atmel ATmega328P mikrokontrolleri põhiandmed

Atmel ATmega328P	
Protsessor	8-bit AVR
Maksimaalne taktsagedus	20 MHz
Programmimälu (Flash)	32 KB (16K x 16)
EEPROM mälu	1 KB
RAM mälu	2 KB
Toitepinge	1.8 V- 5.5 V
Sisendite ja väljundite arv	23
Korpus	TQFP-32

Lisapõhjuseks, miks nimetatud mikrokontroller valitud sai, on mikrokontrolleri riistvaraline võimekus ning erinevate liideste olemasolu. ATmega328P põhineb RISC

arhitektuuril, mis on loodud olema kiire, lihtsate käsuriadega ning riistvaraliselt lihtsa ehitusega [8]. ATmega328P on 8-bitine 32-kilobaidise programmeeritava välkmäluga mikrokontroller, millel on kaks 8-bitist taimerit, üks 16-bitine taimer, kuus PWM kanalit, kaheksa 10-bitist ADC-d, programmeeritav USART [9], SPI liides [10], I²C liides [11]. Lisaandmed on toodud tabelis 2.1.

2.1.2 GNSS Moodul u-blox NEO-M8N

GNSS moodul on selle seadme tähtsaim komponent, sest see mõjutab mõõtetulemuste täpsust kõige rohkem. Valikukriteeriumiteks on:

- GPS [12], Galileo [13] ja GLONASS [14] satelliitide signaali samaaegne vastuvõtt ja signaalitöötlus
- Vähemalt 2 Hz uuenduskiirus
- Sisseehitatud välkmälu
- Odomeetri funktsioon
- Kõrge signaalitundlikus ja hea positsioneerimistäpsus, et seade toimiks ka kehvema signaaliga kohtades
- Töötamine madala pingega ja voolutarbega, kuna seade on mobiilne ja ressursid piiratud
- Madal tootehind
- Saadavus ka eraisikule ning väikestes kogustes

Valitud GNSS mooduliks on firma u-blox poolt toodetav NEO-M8N, mis vastab kõikidele eelnimetatud nõuetele. Turul leidub ka teisi kriteeriumitele vastavaid mooduleid, kuid eelistatud valik oli firma u-blox toodang, sest töö autoril on nendega olemas varasem kokkupuude ning neid müüakse ka prototüüpimiseks sobival kujul. Firma u-blox GNSS moodulite tootevalik on jaotatud moodulite mõõtmete põhjal kuute seeriasse (joonis 2.1) [15].



Joonis 2.1 u-blox GNSS moodulite seeriad [15].

Kõige sobilikumaks osutus NEO seeria moodul mudelinumbriga M8N. NEO seeria moodul on suuruselt keskmiste seas ja mõõtmete poolest probleeme ei tekita (joonis 2.1). Väiksemad ehk EVA ja MAX seeria moodulid on ilma Galileo satelliitide toeta ning neil puudub sisseehitatud välkälu, andmelogi ning mõnedel juhtudel ka RTC. Ainsad moodulid, mis võimaldavad kasutada korraga kolme satelliitsüsteemi ning sisaldavad välkälu ja andmelogi, on NEO-M8N, NEO-M8L ning NEO-M8U (joonis 2.2). Kuna M8U pole hetkeseisuga veel saadaval ning M8L mooduli hind [16] on neli korda suurem M8N hinnast, siis jääb ainuõigeks valikuks just NEO-M8N.

Model	Category				GNSS				Interfaces				Features							Grade						
	Standard Precision GNSS	High Precision GNSS	Dead Reckoning	Timing	GPS / QZSS	GLONASS	Galileo	BeiDou	Number of Concurrent GNSS	UART	USB	SPI	DDC (I/C compliant)	Programmable (Flash)	Data logging	Additional SAW	Additional LNA	RTC crystal	Oscillator	Built-in antenna	Built-in sensor	Built-in antenna supply and supervisor	Timepulse	Standard	Professional	Automotive
EVA-M8M	•				•	•	m	•	w	•	•	•	•	m	m		•	C								
EVA-8M	•				•	•			1	•	•	•	•		m			•	C							
LEA-M8S	•				•	•	•	•	2	•	•	•	•			•		•	T		•					
MAX-M8C	•				•	•	•	•	2	•	•	•	•					•	C							
MAX-M8Q	•				•	•	•	•	2	•	•	•	•					•	T							
MAX-M8W	•				•	•	•	•	2	•	•	•	•					•	T		•					
MAX-8C	•				•	•	•	•	1	•	•	•	•					•	C							
MAX-8Q	•				•	•	•	•	1	•	•	•	•					•	T							
NEO-M8N	•				•	•	•	•	3	•	•	•	•	•	•	•	•	•	T							
NEO-M8Q	•				•	•	•	•	2	•	•	•	•		•	•	•	•	T							
NEO-M8M	•				•	•	•	•	2	•	•	•	•					•	C							
NEO-8Q	•				•	•	•	•	1	•	•	•	•		•	•	•	•	T							
CAM-M8Q	•				•	•	•	•	2	•	•	•	•		•	•	•	•	T							
CAM-M8C	•				•	•	•	•	2	•	•	•	•		•	•	•	•	C		•					
PAM-7Q	•				•	•	•	•	1	•	•	•	•					•	T		•					
NEO-M8L		ADR			•	•	•	•	3	•	•	•	•	•	•	•	•	•	C							
NEO-M8U		UDR			•	•	•	•	3	•	•	•	•	•	•	•	•	•	C							

Joonis 2.2 u-blox GNSS moodulite omaduste võrdlus[15].

NEO-M8N on 72 kanaliga GNSS moodul, millel on GPS & GLONASS režiimis maksimaalne tundlikkus -167 dBm ning maksimaalne uuenduskiirus 5 Hz. Horisontaalne positsioonitäpsus ligikaudu 2,5 m [2].

2.1.3 Cirotech GNSS antenn

GNSS moodul ei toimi ilma õige antennita. Antenni valikul juhenduti u-blox GPS antennide kasutussoovituste andmelehest [17].

Spordikompuuter peab olema ühes tükis, kompaktne ning küllaltki väikeste mõõtmetega, seega peab antenn olema seadme sees ning väiksemõõtmeline. Kõige sobivamaks ja optimaalsemaks antennitüübiks sai valitud planaarrantenn [17].

Valitud antenn on passiivne, sest aktiivne antenn tarbib voolu ja seega pole mõistlik seda kasutada akutoitega seadmes. Passiivse antenni miinuseks võrreldes aktiivse

antenniga on suurem müratundlikkus ning seega peab suuremat tähelepanu pöörama trükkplaadi disainile [17].

Antenni suurus on piiratud eelkõige spordikompuutri mõõtmetega. Kuna antenn joodetakse trükkplaadile, siis peab antenn olema piisavalt väike, et selle ümber jääks ruumi antenni maandusalale ning teistele komponentidele. Valitud antenni mõõtmed on 25x25x4mm, sest selline antenn mahub mõõtmetelt trükkplaadile, on kuluefektiivne ning optimaalse suutlikkusega [17].

Valitud antenniks on tootja Cirocomm poolt pakutav 25x25x4 mm GPS+GLONASS planaarrantenn, mis on sobivate omadustega ja soodsa hinnaga. Tegemist on PIN-tüüpi antenniga ehk ühendustüübiks on läbiaugumontaaž. Antenni signaalide vastuvõtmise sagedusulatus on sobiv GPS-i 1575.42 ± 10.23 MHz ja GLONASS-i 1602 ± 5 MHz jaoks [4].

2.1.4 Heltec 1,3” OLED displei

Displei on spordikompuutri ainus komponent, mis annab treeningu ajal kasutajale tagasisidet. Displeile kuvatakse liikumiskiirus, kellaaeg, altituud ning töörežiim. Nõuded displeile on järgmised:

- Loetavus muutuvates valgustingimustes ehk nii ereda päikse käes kui ka õhtupimeduses.
- Suur vaatenurk
- Madal voolutarve
- Displei diagonaal vähemalt 1,3 tolli ning resolutsioon minimaalselt 128x64 pikslit, et displeile mahuks kogu vajalik informatsioon ja oleks loetav treeningu ajal. Samas peab olema piisavalt väike, et see ei mõjutaks seadme üldmõõtmeid
- Temperatuuritaluvus -10...+40 kraadi

Antud nõudmiste täitmiseks sobib hästi OLED tüüpi displei. Võrreldes LCD tehnoloogiaga displeiga on OLED parema pildikvaliteediga, eredam, suurema vaatenurgaga, kiirema värskendusajaga, madalama voolutarbega, lihtsama ehitusega (mis tagab õhema ehituse) ning parema temperatuuritaluvusega. OLED displei miinusteks LCD kõrval on selle suurem hind ning võimalik lühem eluiga [18].

Valitud displei on Heltec HTDS-WI13, mille displeidiagonaal on 1,3 tolli ning resolutsioon 128x64 pikslit. Displeil on suur 160-kraadine maksimaalne vaatenurk, sobib kasutamiseks temperatuurivahemikus -30...+70 °C, töötab ka 3,3 V pingega ning voolutarve on sisselülitatud olekus keskmiselt 18 mA. Displei draiveri mudeliks on SH1106, mille jaoks on Arduino tarkvarateek displei lihtsaks kasutamiseks [19]. Heltec displei sai valitud tema kompaktsuse, odava hinna ning mugava kommunikatsiooniprotokolli olemasolu tõttu [3].

Helteci displei on joodetud eraldiseisvale trükkplaadile koos tööks vajalike passiivsete komponentidega. See on loodud lihtsaks kasutamiseks näiteks Arduino [7] platvormiga ehk sisaldab kogu vajalikku riistvara displei kasutamiseks I²C protokolliga.

2.1.5 Top Power TP4056 laadija

Kogu spordikompuuter saab oma toite liitiumioonakult ja kuna aku laeng on piiratud, siis vajab see aeg-ajalt laadimist. Laadimiseks on valitud TP4056 lineaarne Li-ioon aku laadimiskiip, mis võimaldab laadimisvoolu kuni 1 A [5]. TP4056-l on soodne hind, seda on lihtne skeemi integreerida ning saadaval on seda kiipi kasutatav prototüüp moodul.

2.1.6 Seadme toiteahel

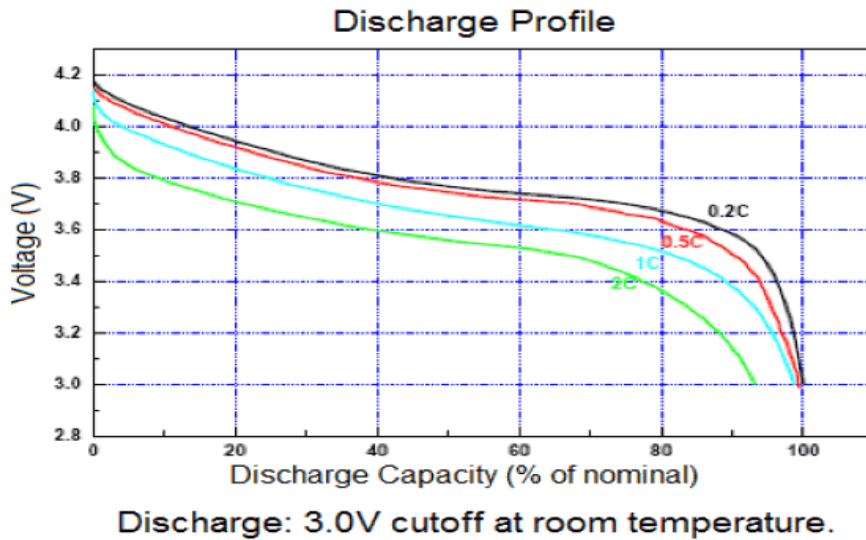
Seadme toiteahel saab alguse liitiumioonakust, mis on ühendatud läbi lüliti 3,3 V pingeregulaatorisse. Liitiumioonaku nominaalpinge on 3,7 V, täislaetult 4,2 V ning minimaalne akule turvaline pinge 3,0 V [20]. Aku mahtuvus on 1,0 Ah. Seadmes olevad komponendid on valitud nii, et need töötaksid 3,3 V nominaalpingega ning ei lõpetaks tööd ka väikese pingelangu korral. Pingeregulaatoriks on Microchip MCP1700, mille maksimaalne väljundvool on 250 mA ning tüüpiline pingelang ligikaudu 178 mV [21]. MCP1700 ostutus valituks just piisava väljundvoolu ja madala pingelangu tõttu.

Kuna seade peab olema töökindel, siis ei lasta pingeregulaatori väljundpingel langeda nominaalsest 3,3 V-st rohkem kui 100 mV. Väljundpinge on sisendpinge (aku pinge) ja regulaatori pingelangu vahe, kuid mitte kõrgem kui 3,3 V ± 0,4%. Pingejaguri abil mõõdetakse pidevalt pinget regulaatori sisendilt ning kui sisendpinge langeb allapoole lubatud piiri, siis lülitatakse seade tarkvaraliselt puhkerežiimi. Minimaalne lubatud sisendpinge arvutatakse järgmiselt:

$$V_{in} = V_{out} + V_{drop}$$

$$V_{in} = 3,2 \text{ V} + 0,178 \text{ V} \approx 3,4 \text{ V}$$

Sellise lahendusega ei kasutata liitiumakust kogu energiat, vaid joonise 2.3 põhjal veidi rohkem kui 90%. Kasutades 1,0 Ah akut, saame teoreetiliselt kasutatavaks mahtuvuseks vähemalt 0,9 Ah.



Joonis 2.3 Liitiumaku pinge ja mahtuvuse graafik tühjakslaadimisel [20].

Tabelis 2.2 on välja toodud põhikomponentide keskmised voolutarbimised olekus, kus GNSS moodul on asukoha tuvastanud ning tulemusi näidatakse displeil ja salvestatakse ka mälukaardile.

Tabel 2.2 Põhikomponentide keskmine teoreetiline voolutarve [1], [2], [20].

Komponent	Voolarve
ATmega328P mikrokontroller	10 mA
u-blox NEO-M8N moodul	34 mA
Heltec OLED displei	18 mA
microSD lugemine/kirjutamine	8 mA
RTC aku laadimine	1 mA
Kokku:	71 mA

Teades aku kasutatavat mahtuvust ja keskmist voolutarvet töörežiimis, saame arvutada teoreetilise aku kestvuse täislaetud aku korral. Üle 12-tunnine aku kestvus on tavapärase harrastaja kasutustingimusi arvesse võttes enamasti piisav.

$$t = E_{aku} / I_{tarbimine} = 0,9 \text{ Ah} / 0,071 \text{ A} \approx 12,67 \text{ tundi}$$

2.2 Kommunikatsioon

Spordikompuutri komponendid peavad seadme toimimiseks pidevalt omavahel suhtlema. Järgnevas alapeatükis on lühidalt kirjeldatud kasutatavaid suhtlusviise.

2.2.1 SPI

Spordikompuuter kasutab andmete microSD mälukaardile salvestamiseks SPI kommunikatsiooni. Antud juhul on olemas käskseade ATmega328P ning üks alamseade ehk microSD kaart. SPI on sünkroonne andmeprotokoll, milles on üks käskseade (master), mis suhtleb ühe või mitme alamseadmega (slave) [10]. SPI kommunikatsiooni jaoks on vajalik neli signaalijuhet:

- MISO (Master In Slave Out) – Suhtlus alamseadmelt käskseadmele
- MOSI (Master Out Slave In) – Suhtlus käskseadmelt alamseadme(te)le
- SCK (Serial Clock) – Taktpulsid, mis sünkroniseerivad andmevahetuse
- SS (Slave Select) – Alamseadme valik

MISO, MOSI ning SCK liinid otseselt ei vaja *pull-up* takisteid, kuid kuna microSD prototüüplaadil olid need olemas, siis said need ka skeemi lisatud. Võimalus on neid trükkplaadile mitte joota kui need üleliigseks osutuvad [10].

2.2.2 I²C

Mikrokontroller suhtleb displeiga üle I²C jadasiini, milleks on vaja kõigest kahte ühendust: andmesiin ja kellatakt [11]. Kuna displei on seadme põhitrükkplaadist eraldi, siis on mugavam kui nende vahel on võimalikult vähe ühendusjuhtmeid. Displeid ja mikrokontrollerit ühendab kokku neli juhet: andmesiin, kellatakt, maandus ja toide.

I²C andmesiin ning kellatakt on avatud kollektori disainiga, mis tähendab, et mõlema liini puhul on vajalik kasutada *pull-up* takisteid. *Pull-up* takisti on ühendatud liini ja positiivse toite vahele. Kui liinis ei toimu suhtlust, siis on liin tõmmatud kõrgeks [11].

I²C kommunikatsiooni teostatakse tarkvaraliselt Arduino Wire teegi abil [21].

2.2.3 USART

Spordikompuutris on kasutatud kahte USART kommunikatsiooni. Üheks on ATmega328P mikrokontrollerile sisseehitatud programmeeritav USART ning teine on tekitatud tarkvaraliselt.

Mikrokontrollerisse sisseehitatud USART jadaport on kasutusel mikrokontrolleri programmeerimiseks ehk suhtlus toimub mikrokontrolleri ja programmeeriva seadme vahel läbi FTDI FT232R kiibi, mis teisendab signaalid USB standardilt USART standardile ning vastupidi [22].

Teine USART ühendus on loodud mikrokontrolleri ja GNSS mooduli vahel. Kuna ATmega328P mikrokontroller omab vaid ühte USART jadaporti ning see on kasutusel mikrokontrolleri programmeerimiseks, siis oli vajalik tekitada see tarkvaraliselt. Arduino tarkvara SoftwareSerial teek võimaldab mikrokontrolleri digitaalklemme kasutada jadapordina [23]. Praegusel juhul tekitatakse tarkvaraline jadaport mikrokontrolleri jalgadele 11 ja 12.

2.3 Programmeerimine

Spordikompuutri mikrokontroller ja GNSS moodul vajavad enne kasutamist programmeerimist. Järgnev alapeatükk tutvustab lühidalt kasutatavaid programmeerimisviise.

2.3.1 Mikrokontroller

Spordikompuutri mikrokontroller kasutab Arduino avatud lähtekoodiga tarkvara. Selleks, et Arduino tarkvara saaks kasutada, on vaja mikrokontrollerisse installeerida alglaadur, mille ülesanne on mikrokontrolleri sisselülitamisel käivitada Arduino programmikood. Alglaadur installeeritakse ATmega328P mikrokontrollerisse kasutades MOSI, MISO, SCK ja RESET (vastavalt mikrokontrolleri jalad 15, 16, 17 ja 29) signaaliühendusi ning spordikompuutri toide peab olema sisse lülitatud. Samuti peab programmaatoril ja mikrokontrolleril olema ühine maandus. Alglaaduri installeerimiseks kasutatakse Arduino UNO prototüüpimise plaati ning Arduino IDE tarkvaraprogrammi [24], [25], [26].

Atmel ATmega328P mikrokontrolleri edasiseks programmeerimiseks kasutatakse eraldiseisvat programmeatorit, mis kasutab FTDI FT232R kiipi [21]. Arduino algaadur võimaldab programmeerimiseks kasutada mikrokontrolleri USART ühendust [24]. Programmeator ühendub mikrokontrolleri TX, RX, RESET (vastavalt mikrokontrolleri jalad 31, 30, 29) ja GND klemmidele ning programmeerimise ajal peab olema seadme toide sisse lülitatud. Programmikoodi mikrokontrollerisse kirjutamiseks kasutatakse Arduino IDE tarkvaraprogrammi [26].

2.3.2 GNSS moodul

u-blox GNSS mooduli tarkvara programmeerimine toimub läbi spordikompuutri mikrokontrolleri. Mikrokontrollerisse kirjutatakse programmikood, millega seadistatakse GNSS moodul. Mikrokontroller ja GNSS moodul suhtlevad omavahel läbi tarkvaraliselt seadistatud kommunikatsiooniliini, kuna ATmega328P protsessori ainus jadaport on kasutusel mikrokontrolleri programmeerimiseks.

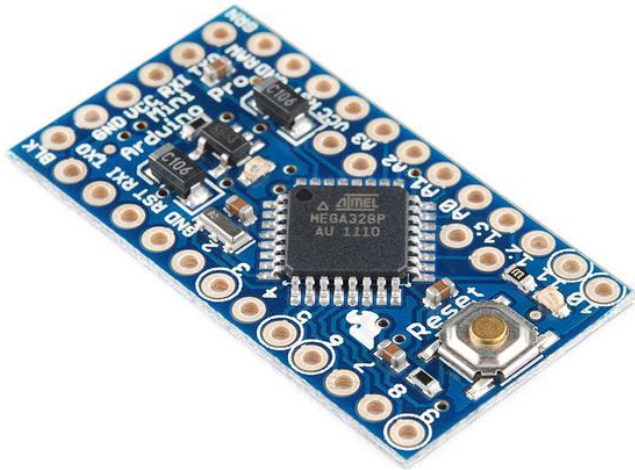
Teine võimalus on GNSS moodulit programmeerida Windows arvutiga kasutades programmi u-blox u-center [27]. Ühenduse loomiseks arvutiga saab kasutada mikrokontrolleri programmeerimiseks kasutatavat FTDI programmeatorit. Seadme trükkplaadile on välja toodud GNSS mooduli TX ja RX testklemmid, mille kaudu ühendus luua.

2.4 Prototüüpseade

Spordikompuutri testimiseks koostati prototüüpseade, mis koosneb mitmest eraldiseisvast moodulist: mikrokontrolleri moodul, GNSS prototüüp moodul, laadimismoodul, OLED displei moodul ja mälukaardi moodul. Moodulid on omavahel juhtmetega ühendatud. Järgnevas alapeatükis tutvustatakse lühidalt prototüüpseadme koostisosi ning testimist.

2.4.1 Prototüüpseadme moodulid

Mikrokontrolleri mooduliks on valitud Arduino Pro Mini plaat (joonis 2.4). See sisaldab ATmega328P mikrokontrollerit, keraamilist resonaatorit taktsageduse genereerimiseks, 35-te sisend/väljund jooteauku vajalikeks ühendusteks väliste komponentidega, kahte LED-i, taaslähtestuse nuppu ning LDO pingeregulaatorit MIC5205 ning vajalikke passiivseid komponente [28].



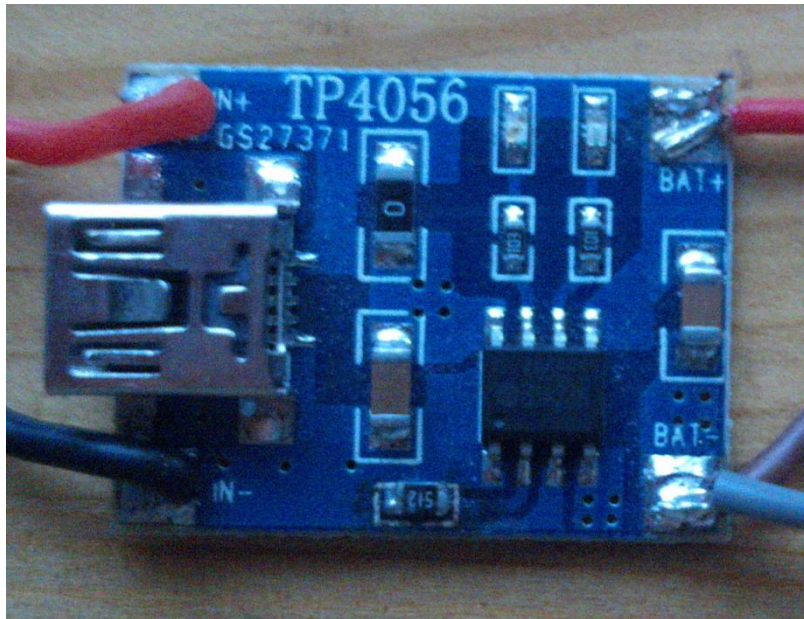
Joonis 2.4 Arduino Pro Mini prototüüplaat [28].

GNSS prototüüpmoodul (joonis 2.5) sisaldab GNSS moodulit u-blox NEO-M8N, keraamilist planaarrantenni, LED-i, kompassi, LDO pingeregulaatorit ning vajalikke passiivseid komponente.



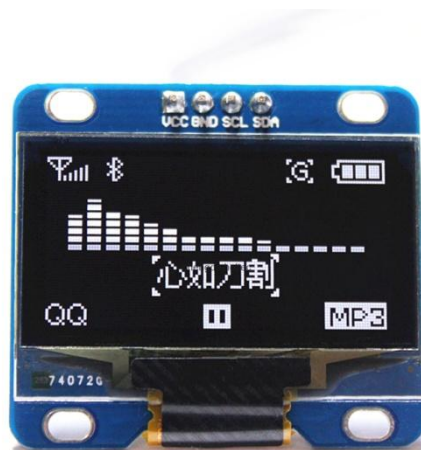
Joonis 2.5 Korpuse ja ühendusjuhtmetega GNSS moodul prototüüpmudeli jaoks.

Laadimismooduli (joonis 2.6) põhikomponendiks on Top Power TP4056 Li-ion laadimiskiip. Lisaks on moodulil USB mini-B ühenduspistik, mille kaudu toide jõuab kiibini. Samuti ei puudu plaadilt seadme tööks vajalikud passiivkomponendid, kaks jooteala Li-ioon akuga ühenduseks ning kaks LED-i, mis näitavad laadimiskiibi olekut. Punane LED on aktiivne, kui toimub aku laadimine ning roheline LED on aktiivne, kui aku laadimine on lõpetatud [5].



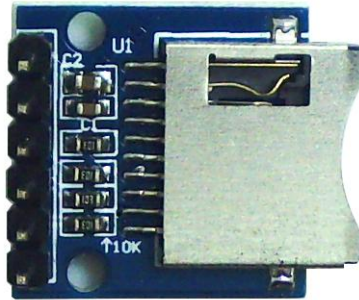
Joonis 2.6 Laadimismoodul TP4056 baasil.

Heltec OLED displeimoodul (joonis 2.7) koosneb OLED digitaalspleist, displeikontrollerist SH1106, vajalikest passiivkomponentidest ning neljast jootealast vajalike ühendusjuhtmete jootmiseks [3], [29]. OLED displeimoodul on samal kujul kasutusel nii prototüüpmodelis kui ka spordikompuutri lõppversioonis, v.a. *pull-up* takistid R2 ja R3 (lisa 2) [28], sest need on paigutatud põhiplaadile.



Joonis 2.7 Heltec OLED displei[3].

MicroSD moodul (joonis 2.8) sisaldab microSD kaardi pesa, *pull-up* takisteid ning lahtisidestuskondensaatoreid. Moodulil on kuus ühendusjuhet: neli SPI kommunikatsiooniliini, toide ja maandus [30].



Joonis 2.8 LC Technology microSD mälukaardi ühendamise moodul [30].

2.4.2 Prototüüpseadme testimine

Prototüüpseadet (joonis 2.9) on testitud treeningutel ja võistlustel aastaringsetl. Kogu kasutatav elektroonika on paigutatud 85x50x20 mm plastkorpusesse, mille külge on monteeritud kaks lülitit (toitelüliti ja režiimilüliti) ning displei. Seadme OLED displeile kuvatakse liikumiskiirus, kellaeg, altituud ja töörežiim ning displei on selgelt loetav, olenemata välistest valgustingimustest. Mälukaardile salvestatakse seadme kehtivad koordinaadid, kellaeg ning altituud sagedusega 1 Hz, mis on määratud tarkvara lähtekoodis (lisa 3).

Prototüüpseadme liikumiskiiruse mõõtetulemuste täpsust võrreldi keelreleed ja magnetit kasutava kalibreeritud rattaspidomeetri mõõtetulemustega ning erinevused on tühised. Sellest järeldub, et seade on spordikompuutrina kasutamiseks piisavalt täpne. Märkatavad erinevused tekkisid absoluutse kõrguse ehk altituudi mõõtmistel. Prototüüpseadme altituudi mõõtetulemusi võrreldi internetipõhise rakenduse FreeMapTools kõrguskaardi andmetega [31]. Mõõtetulemuste erinevus on kohati kuni 10 meetrit. Selline mõõteviga esineb prototüüpseadmél sageli linnas või metsas - kohtades, kus on kehvem GNSS signaal.

Li-ion aku laadimise testimiseks ühendati toiteplokk laadimismooduli USB pistikuga. Mooduli punane LED hakkas valgust kiirgama ning mõne aja möödudes see kustus, mis näitab, et aku laadimine on lõppenud. Voltmeetriga aku pinget kontrollides saadi tulemuseks 4,196 V, mis on vägagi ligilähedane aku maksimaalsele pingele [20].



Joonis 2.9 Spordikompuutri prototüüpmodel.

3 Elektriskeem ja trükkplaat

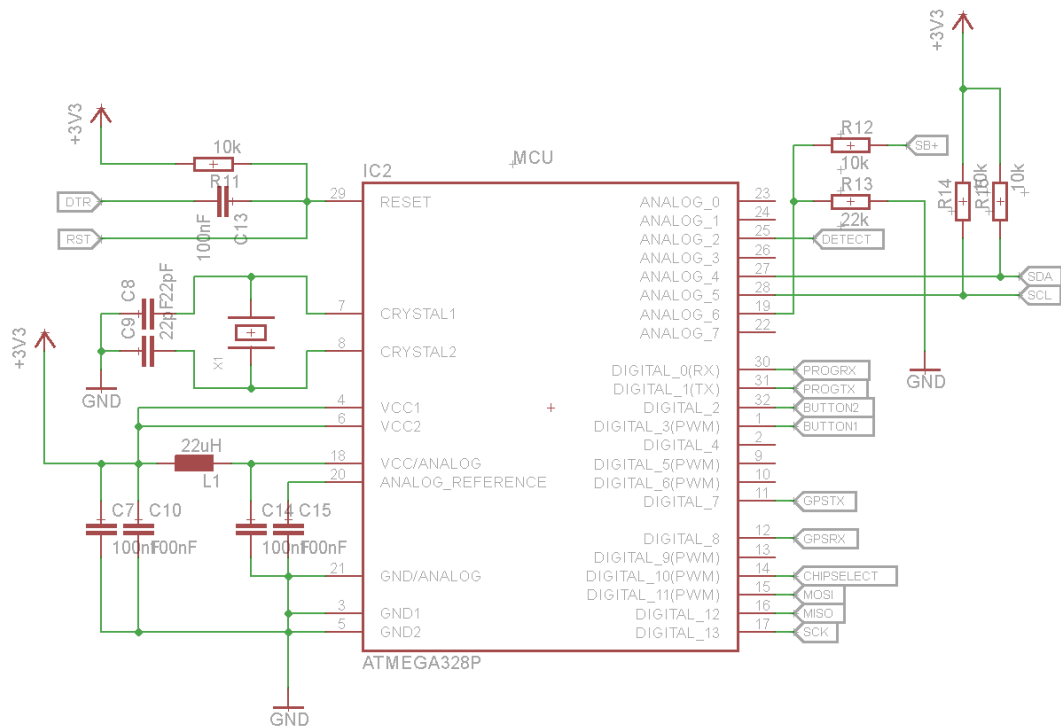
Järgnev peatükk annab ülevaate spordikompuutri elektriskeemist, selles kasutatavate aktiivsete ja passiivsete komponentide vajalikkusest ja trükkplaadi disainist ning selle nõuetest.

3.1 Elektriskeemi koostamine

Elektriskeemi koostamisel on lähtunud eelkõige komponentide andmelehtedest ehk tootjatepoolsetest soovitustest konkreetsete komponendide integreerimisel skeemi.

3.1.1 Mikrokontroller

Mikrokontrolleri skeemiosas (joonis 3.1) näeme protsessori taktsagedust genereerivat kvartsi X1, pingejagurit (R12, R13) aku pinge mõõtmiseks, *pull-up* takisteid I²C ning RESET liinidel (R14, R15, R11), madalpääsfiltrit (L1, C14) VCC/ANALOG liinil ning lahtisidestuskondensaatoreid (C7, C10, C15).



Joonis 3.1 Mikrokontrolleri skeem.

Mikrokontrolleri kvartsi kondensaatorite C8 ja C9 väärtused on valitud kvartsi ja mikrokontrolleri tootjapoolsete soovitude alusel [1], [32].

Madalpääsfilter analoogtoite sisendil eraldab selle ahela ülejäänud toiteahelatest ning vähendab ahelasse jõudvat kõrgsageduslikku müra. Selle tulemusena on ADC töö stabiilsem. Seade kasutab ADC sisendit aku pinget mõõtmiseks pingejaguriga.

Pingejaguri takistid on valitud optimaalse väärtusega, et voolutarve poleks liialt suur. Mida väiksemad on takistite väärtused, seda suurem on tarbimine, aga seejuures on mõõtetulemus täpsem. Teades aku maksimaalset pinget ning pingejaguri takistite väärtuseid, saame arvutada maksimaalse pingejagurit läbiva voolu:

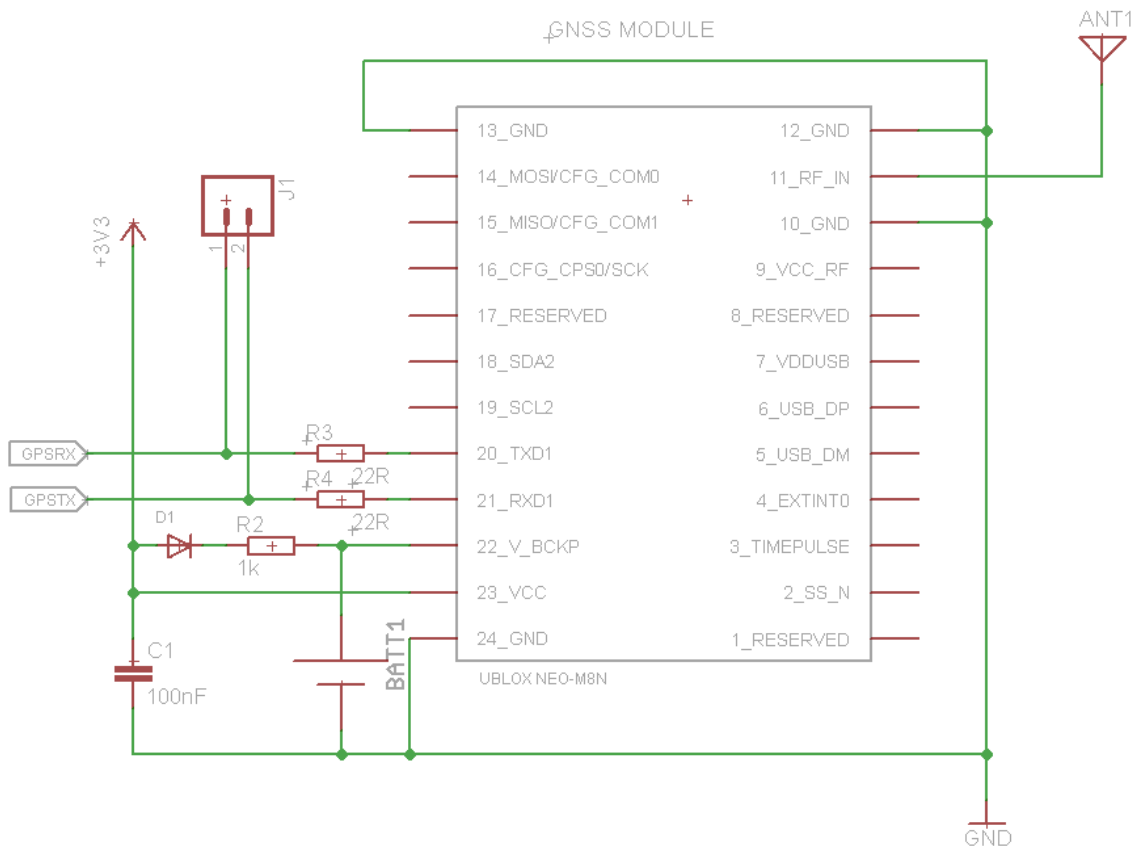
$$I_{max} = \frac{V_{max}}{R_{12} + R_{13}} = \frac{4,2V}{10k\Omega + 22k\Omega} = 131,25 \mu A.$$

3.1.2 GNSS moodul

GNSS mooduli skeem koosneb NEO-M8N moodulist, GNSS antennist, varutoitest ja selle laadimisskeemist ning takistitest kommunikatsiooniliinidel.

GNSS moodulil on soovituslik kasutada varutoidet, et võimaldada kiirema starti funktsioon. Kui spordikompuutri toide on välja lülitatud, siis GNSS moodul kasutab varutoidet, et hoida mälus reaajakella ja akutoitel olevat RAM-i, millel on mälus GNSS orbiidiandmed. Varutoiteks on Seiko MS621FE 3 V liitiumaku mahtuvusega 5,5 mAh [32]. GNSS mooduli andmelehe põhjal on tarbimine varutoitest 15 μ A, mis 100% laetuse korral annab aku teoreetiliseks kestvuseks ligikaudu 366 tundi ehk ümardatult 15 päeva [2].

Varutoite laadimisskeem koosneb diodist ja takistist. Joonisel 3.2 diod D1 [32] on pingelangu tekitamiseks, et saada laadimiseks optimaalne pinge. Takisti R2 on vajalik laadimisvoolu piiramiseks.



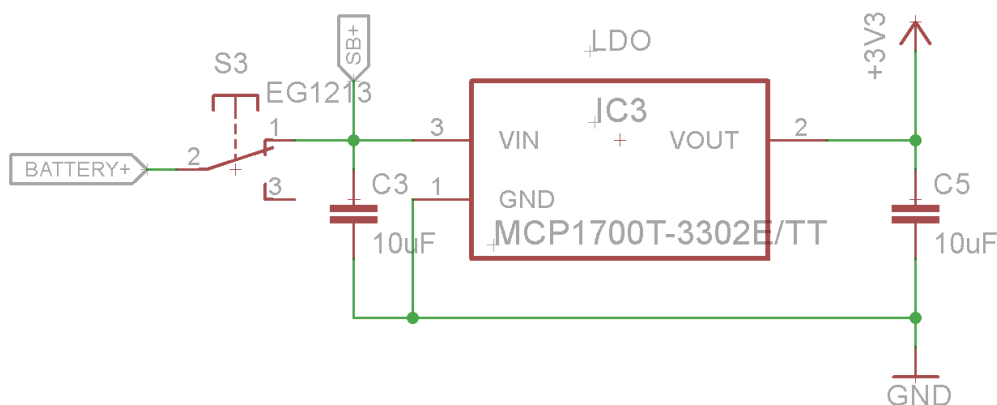
Joonis 3.2 GNSS mooduli skeem.

Takistid R3 ja R4 kommunikatsiooniliinidel on elektromagnetilise interferentsi moodulisse jõudmise takistamiseks. Ükskõik milline sisend/väljund signaalijuhe, mis on pikem kui 3 mm, võib hakata toimima antennina ning olla vastuvõtlik ümbritsevate signaalide suhtes. Sellepärast on soovituslik kasutada tundlike komponentide

signaalijuhtmel takistit, ferriiti või induktorit, mis takistab EMI jõudmist juhtme kaudu komponendini [34].

3.1.3 Pingeregulaator

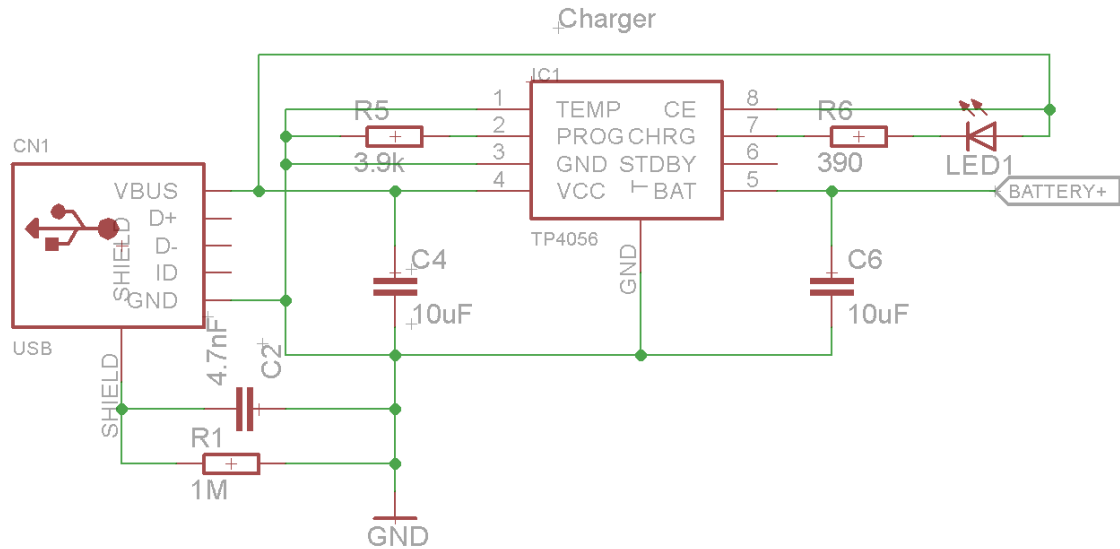
Seadme pingeregulaator (joonis 3.3) Microchip MCP1700 vajab stabiilseks tööks nii sisendisse kui väljundisse minimaalselt 1,0 μF keraamilist kondensaatorit. Valitud said 10 μF kondensaatorid pingestabiiliseerimiseks (C3, C5). S3 on seadme toitelüliti.



Joonis 3.3 Pingeregulaatori skeem.

3.1.4 Aku laadimine

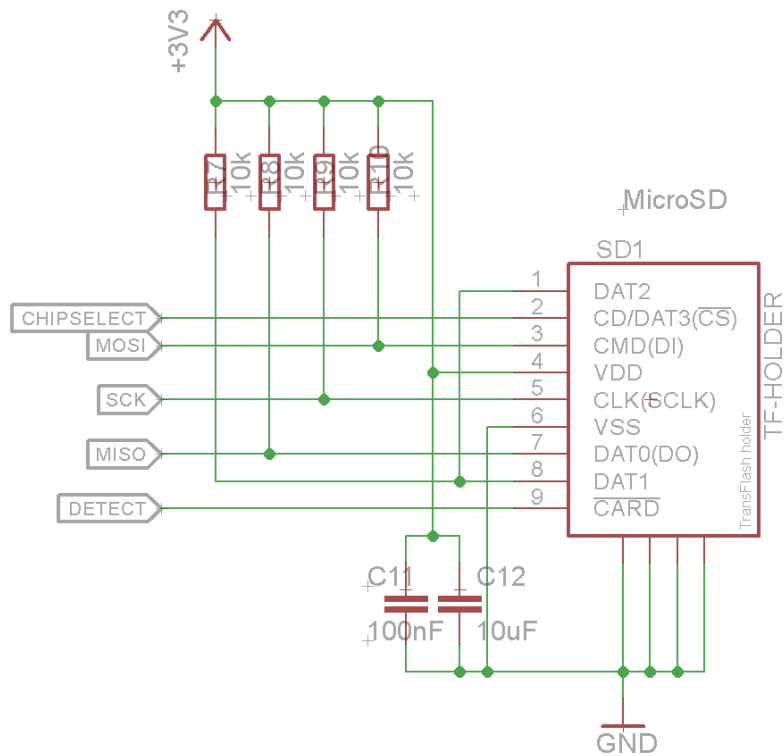
Aku laadimisskeem on joonisel 3.4. Aku laadimiseks tuleb seade ühendada 5 V toiteallikaga kasutades seadme USB mini-B tüüpi pistikut. Selleks, et pistikusse ühendatav kaabel ei hakkaks toimima antennina, on pistiku korpuse ja seadme maandusala vahel RC-filter (R1 ja C2). Aku maksimaalne laadimisvool määratakse takistiga R5, mille väärtus 3,9 $\text{k}\Omega$ on valitud komponendi andmelehe põhjal 300 mA laadimisvoolu seadmiseks [5]. Selleks, et aku olekut teada laadimise ajal, on lisatud laadimiskiibile valgusdiod LED1 koos voolu piirava takistiga R6. Valgusdiod kiirgab punast valgust, kui akut laetakse, ning kustub, kui aku on täis laetud. Kogu laadimisskeem on aktiivses olekus vaid juhul, kui seade on USB pistiku kaudu ühendatud välise toitega.



Joonis 3.4 Aku laadimiseks vajalik skeem.

3.1.5 Mälukaardi ühendamine

Mälukaardi ühendamiseks vajalik skeem on joonisel 3.5. R7-R10 on liinide *pull-up* takistid ning C11 ja C12 on toiteahela lahtisidestuskondensaatorid [30]. Komponent SD1 on microSD mälukaardi pesa, kuhu sisestatakse mälukaart.



Joonis 3.5 Mälukaardi pesa skeem.

3.1.6 Pistikud ja lülitid

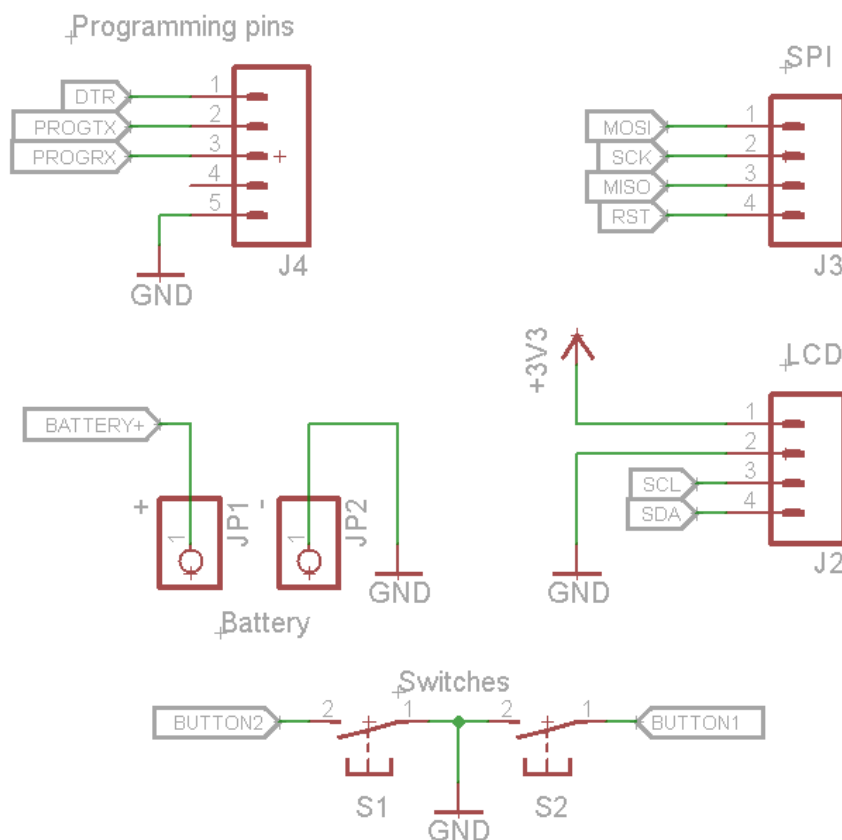
Mikrokontrolleri programmeerimiseks on skeemis välja toodud programmeerimispistik J4 (joonis 3.6), mis koosneb USART-i TX ja RX signaalidest, DTR signaalist (mikrokontrolleerile taaskäivituse tegemiseks) ning maandusest.

Pistik J3 (joonis 3.6) koosneb mikrokontrolleri SPI liidesest, mida kasutatakse mikrokontrolleerile algladuri installeerimiseks. Samuti on vajalik taaskäivituse signaal.

Klemmid JP1 ja JP2 (joonis 3.6) on toitejuhtmete ehk akujuhtmete ühendamiseks trükkplaadiga.

Seadme displei ühendatakse pistikusse J2 (joonis 3.6), mis koosneb neljast juhtmest: toide, maandus ning kaks I²C siini juhet.

S1 (lülit 1) ja S2 (lülit 2) joonisel 3.6 on seadme kaks lülitit, millega kasutaja saab spordikompuutri režiimi muuta (lisa 1). Trükkplaadi ruumi säästmiseks kasutatakse siinkohal mikrokontrolleerile sisseehitatud *pull-up* takisteid (lisa 3).



Joonis 3.6 Pistikute ja lülitite skeem.

3.2 Trükkplaadi disain

Spordikompuutri trükkplaadi disain sõltub suuresti seadme välisdisainist. Põhilised autoripoolsed nõuded välisdisainile on järgmised:

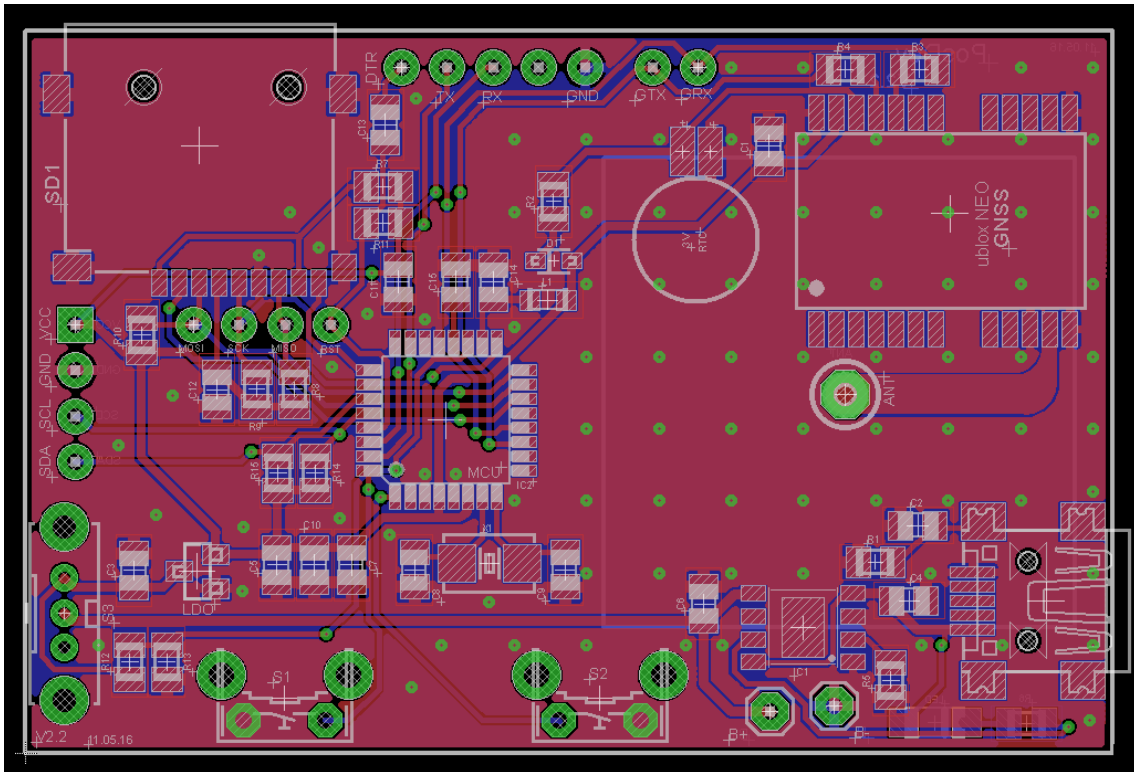
- Displei ja GNSS antenn asuvad seadme pealmisel küljel
- Toitelüliti asub seadme esiküljel ning juhtimislülid parempoolsel küljel.
- Spordikompuutri mõõtmed on ligikaudu 65x45x25 mm

Trükkplaadi disaini esimeseks sammuks on plaadi mõõtmete määramine. Kuna trükkplaati ümbritseb seadme korpus paksusega umbes 2 mm ning kompuutri üldmõõtmed on teada, siis trükkplaadi suuruseks on 60x40 mm. Esimesed plaadile paigutatud komponendid on GNSS antenn, lülid ning displeiga ühendamise klemmid.

Antenn ning displei peavad asetsema plaadi ühel poolel ehk seadme pealmisel poolel, et displei oleks kasutajale nähtav ning antenn oleks GNSS signaalidele vastuvõtlikum [35]. Joonis 3.7 on seadme suhtes alt vaadatuna. Aktiivsed komponendid ja signaalijuhtmed on vajalik paigutada GNSS antennist võimalikult kaugele, sest need võivad segada GNSS signaale. Trükkplaadi väikese pindala tõttu oli vajalik mõned komponendid siiski ka antenni lähedusse paigutada.

Antenni alla plaadi vastaspoolele on paigutatud GNSS moodul, RTC aku ning laadimisskeem, sest teoreetiliselt on seadme kasutamisel nende komponentide mõju GNSS signaalile väike [35]. Antennijuhe peab omama antenniga ligilähedast impedantsi ehk 50Ω [4]. Antennijuhtme laiuse kalkuleerimiseks kasutati EEWeb impedantsi kalkulaatorit [36]. Vastavalt u-blox juhistele on kogu antennialune pind maandusala [35].

Järgmisena paigutati toitelüliti S3 plaadi esiosasse (joonisel 3.7 vasakul) ning juhtimislülid S1 ja S2 paremale küljele (joonisel 3.7 all), mis jättis mälukaardi skeemile ruumi vaid plaadi vasemale küljele. Mikrokontrolleri ja LDO skeem paigutati plaadi kesk- ja esiosasse.



Joonis 3.7 Spordikompuutri trükkplaadi disain programmis CadSoft Eagle.

4 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk luua globaalsel satelliitnavigatsioonisüsteemil baseeruv spordikompuuter õnnestus. Tulemus ei ole kindlasti spordikompuutri lõplik variant, sest mõõtetulemuste parandamiseks ja seadme töö täiustamiseks on võimalik ja vajalik täiendada seadme tarkvara (lisa 3) ning ka riistvara.

Prototüüpmodeli põhjal võib öelda, et spordikompuutri komponentide, kommunikatsiooni- ja programmeerimisviiside valik on tehtud õigesti. Komponentid vastavad seatud nõuetele, nende omavaheline suhtlus toimib ning programmeerimisel probleeme ei tekkinud.

Mõõtetulemuste test prototüüpmodelil osutus edukaks: OLED displei näitab vajalikke ja piisavalt korrektseid andmeid, mälukaartile salvestatud failid on korrektselt kirjutatud.

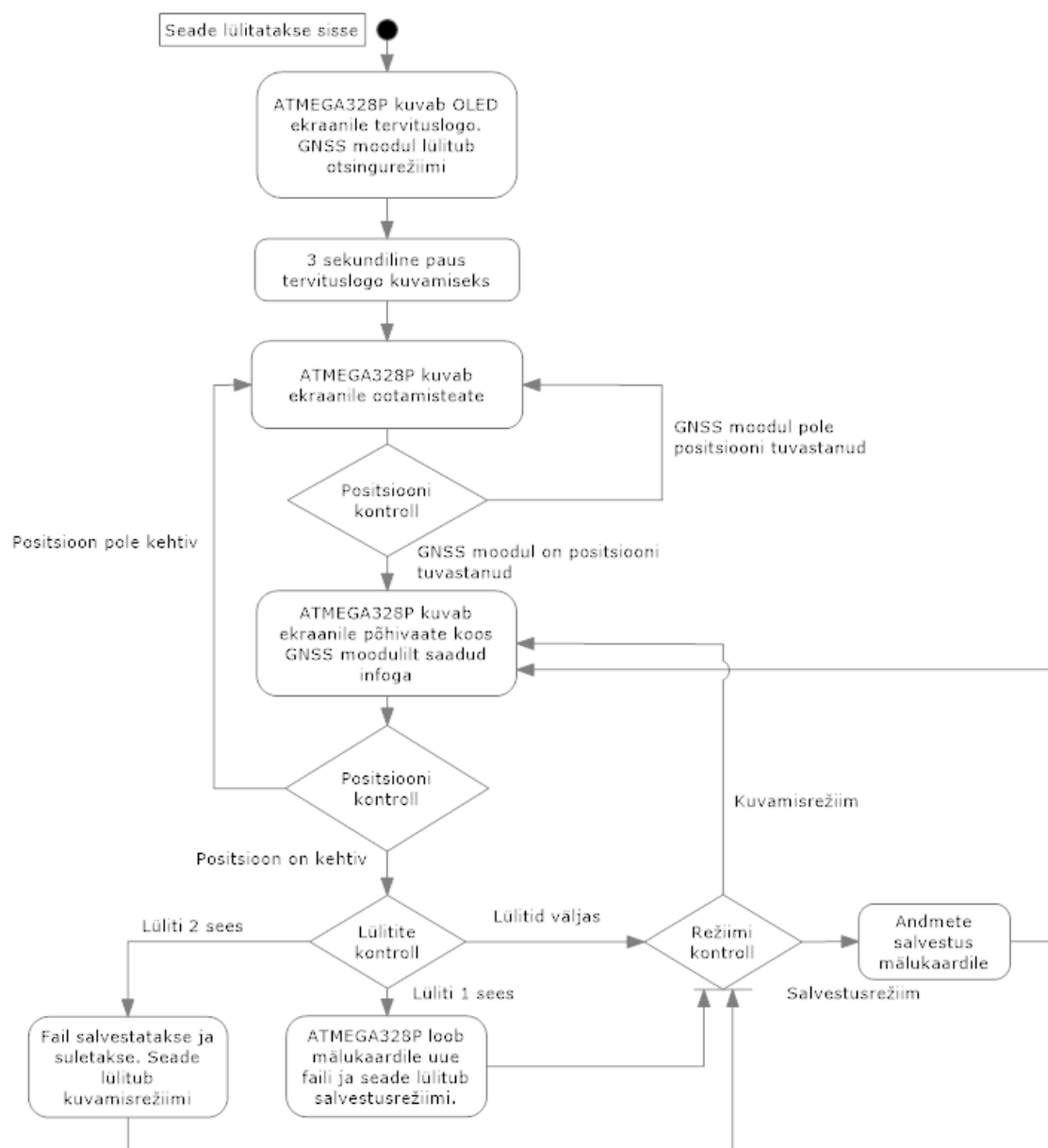
Kasutatud kirjandus

1. Atmel Corporation. (2015). **ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH.**
2. **u-blox NEO-M8 Data Sheet** [www] https://www.u-blox.com/sites/default/files/NEO-M8_DataSheet_%28UBX-13003366%29.pdf (11.03.2016)
3. **Heltec 1.3" OLED displei** [www] http://heltec.diytrade.com/sdp/2044581/4/pdf-6785993/18966329-0/1_3inch_IIC_OLED_module_white_color_for_Arduino_a.html (17.05.2016)
4. **Cirocomm antennid** [www] <http://www.cirocomm.com/category-Pin-Type-2-3-1.html> (11.04.2016)
5. **NanJing Top Power TP4056 Li-ion Battery Charger** [www] <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf> (09.05.2016)
6. **Microchip MCP1700 LDO pingeregulaator** [www] <http://ww1.microchip.com/downloads/en/DeviceDoc/20001826C.pdf> (15.05.2016)
7. **Arduino FAQ** [www] <https://www.arduino.cc/en/Main/FAQ> (17.05.2016)
8. **RISC mikroprotsessor** [www] <http://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/riscisc/> (23.05.2016)
9. **USART introduction** [www] <https://www.arduino.cc/en/Reference/Serial> (23.05.2016)
10. **Arduino SPI library introduction** [www] <https://www.arduino.cc/en/reference/SPI> (23.05.2016)
11. **I²C tutvustus** [www] <https://learn.sparkfun.com/tutorials/i2c> (23.05.2016)
12. **GPS satelliitnavigatsiooni süsteem** [www] <https://et.wikipedia.org/wiki/GPS> (23.05.2016)

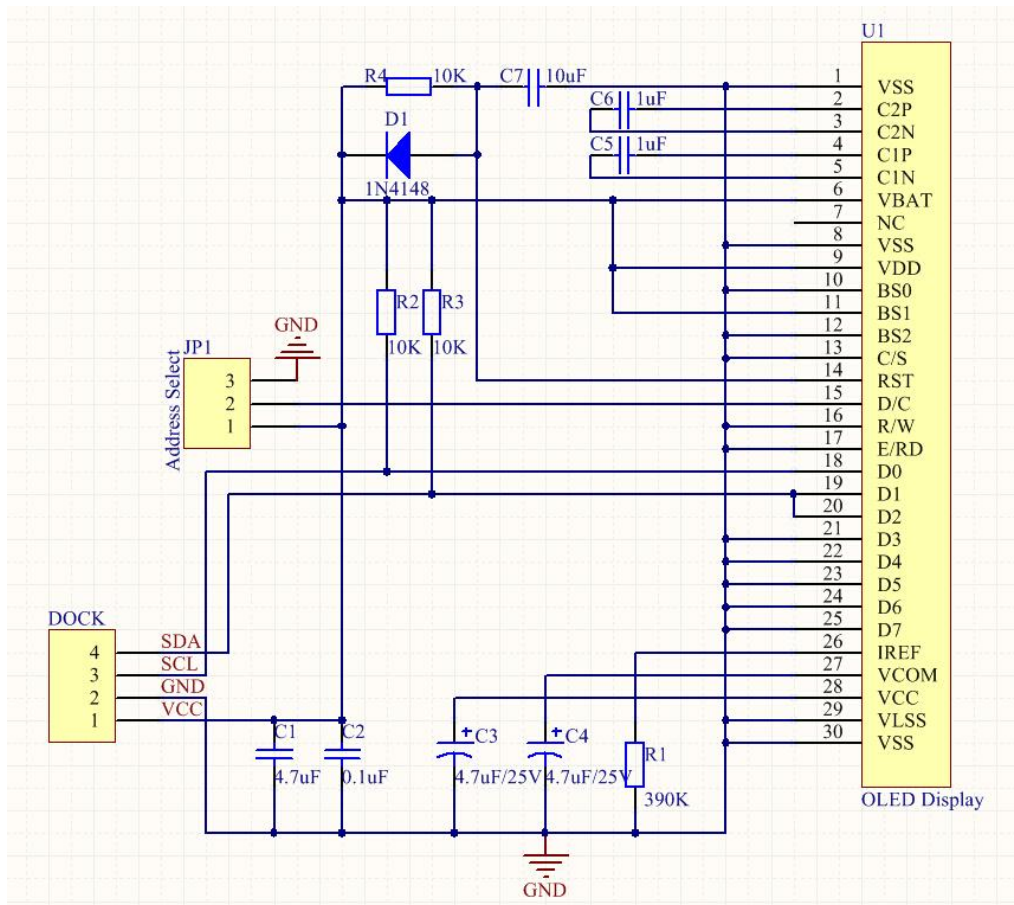
- 13. Galileo positsioneerimissüsteem [www]**
https://et.wikipedia.org/wiki/Galileo_%28positsioneerimiss%C3%BCsteem%29
 (23.05.2016)
- 14. GLONASS satelliitnavigatsiooni süsteem [www]**
<https://et.wikipedia.org/wiki/GLONASS> (23.05.2016)
- 15. u-blox GNSS moodulite valiku ülevaade [www]** https://www.u-blox.com/sites/default/files/products/documents/GNSS-module-selector_Overview_%28UBX-14000426%29.pdf (17.05.2016)
- 16. u-blox online shop, NEO series [www]** <https://shop-emea.u-blox.com/en/eur/3~130~EMEA/Position-Time/High-Precision-GNSS/NEO-series> (23.05.2016)
- 17. u-blox, GPS antennid [www]** https://www.u-blox.com/sites/default/files/products/documents/GPS-Antenna_AppNote_%28GPS-X-08014%29.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2FGPS_Antennas_ApplicationNote%28GPS-X-08014%29.pdf (17.05.2016)
- 18. OLED introduction [www]** <http://www.oled-info.com/introduction>
 (17.05.2016)
- 19. U8glib graphics library for SH1106 [www]**
<http://platformio.org/lib/show/7/U8glib> (23.05.2016)
- 20. Adafruit Li-ion and Li-poly battery voltages [www]**
<https://learn.adafruit.com/li-ion-and-lipoly-batteries/voltages> (23.05.2016)
- 21. Arduino Wire library [www]** <https://www.arduino.cc/en/Reference/Wire>
 (20.05.2016)
- 22. FTDI FT232R [www]** <http://www.ftdichip.com/Products/ICs/FT232R.htm>
 (23.05.2016)
- 23. Arduino SoftwareSerial [www]**
<https://www.arduino.cc/en/Reference/SoftwareSerial> (23.05.2016)
- 24. Arduino bootloader [www]**
<https://www.arduino.cc/en/Hacking/Bootloader?from=Tutorial.Bootloader>
 (20.05.2016)
- 25. Arduino UNO [www]** <https://www.arduino.cc/en/Main/ArduinoBoardUno>
 (23.05.2016)

26. **Arduino IDE software** [www] <https://www.arduino.cc/en/Main/Software>
(23.05.2016)
27. **u-blox u-center** [www] <https://www.u-blox.com/en/product/u-center-windows>
(23.05.2016)
28. **Arduino Pro Mini** [www] <https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v> (16.05.2016)
29. **Heltec OLED skeem** [www] <https://bengoncalves.net/2015/10/01/oled-display-and-arduino-with-power-save-mode/> (18.05.2016)
30. **LC Technology Mini SD** [www] <http://www.lctech-inc.com/plus/view.php?aid=12> (23.05.2016)
31. **FreeMapTools kaardirakendus** [www]
<https://www.freemaptools.com/elevation-finder.htm> (29.05.2016)
32. **NEWXTAL Quartz Series** [www]
<http://www.newxtal.com/files/data/200810159377274.pdf> (23.05.2016)
33. **Seiko MS621FE aku andmeleht** [www]
<http://www.stm32circle.com/resources/Datasheets/MS621FE.pdf> (23.05.2016)
34. **Diodes Incorporated SD103AWS Diode** [www]
http://www.diodes.com/_files/datasheets/ds30101.pdf (09.05.2016)
35. u-blox. (2015). **u-blox NEO-M8, u-blox M8 concurrent GNSS module, Hardware Integration Manual**
36. **EEWeb Microstrip Impedance** [www]
<https://www.eeweb.com/toolbox/microstrip-impedance> (23.05.2016)

Lisa 1 – Spordikompuutri olekumudel



Lisa 2 – Heltec OLED displei elektriskeem



Heltec OLED skeem [www] <https://bengoncalves.net/2015/10/01/oled-display-and-arduino-with-power-save-mode/> (18.05.2016)

Lisa 3 – Prototüüpseadme tarkvara lähtekood

```
#include <TinyGPS++.h>           //NMEA liidese teek
#include <SoftwareSerial.h>      //Tarkvaralise UARTi teek
#include <SD.h>                  //Malukaardi teek
#include <Wire.h>                //I2C teek
#include <U8glib.h>              //displei graafika teek

U8GLIB_SH1106_128X64 u8g(U8G_I2C_OPT_NONE); //displei draiveri valik
File logFile;
TinyGPSPlus gnss;
SoftwareSerial ss(8, 7);        //Tarkvaralise UARTi klemmide maaramine
unsigned char val = 6;          //algreziimi seadmine
unsigned long t = 0;
unsigned long y = 0;
const long interval = 1000;     //salvestusintervalli maaramine
char filename [12];             //faili nime muutuja defineerimine

void setup()
{
  LCDStart();                   //tervitusteksti kuvamine displeile
  pinMode(2,INPUT_PULLUP);      //S1 nupu pull-up resistori aktiveerimine
  pinMode(3,INPUT_PULLUP);      //S2 nupu pull-up resistori aktiveerimine
  pinMode(A2, INPUT_PULLUP);    //kaardi DETECT pini pull-up aktiveerimine
  pinMode(10,OUTPUT);
  SD.begin(10);                 //malukaardi SPI juhtme maaramine
  ss.begin(9600);               //GNSS mooduliga suhtluse alustamine
  gpsDelay();                   //GNSS mooduli positsioonikontroll
}
void loop()
{
  getData();                    //GNSS moodulilt andmete vastuvott
  mode();                        //reziimikontroll
  fileWrite();                   //andmete salvestamine malukaardile
  LCDWrite();                    //andmete kuvamine displeile
}
void getData()
{
  while (ss.available() > 0)
  {
    gps.encode(ss.read());
  }
}

void fileWrite()
{ switch(val)
  { case 1: gpxStart();          //faili alustamine
    val = 3;
    break;
```

```

        case 2: gpxEnd();          //faili lopetamine
                val = 0;
                break;
        case 3: dataWrite();      //andmete salvestus malukaardile
                break;
    }}

void mode()
{
    const long interval2 = 1200; //luliti allhoidmise aeg
    if(digitalRead(2) != HIGH)
    {
        unsigned long y = millis();
        while(digitalRead(2) != HIGH)
        {
            unsigned long currentMillis = millis();
            if(currentMillis - y >= interval2)
            {
                switch(val)
                { case 0: val = 1; break;
                  case 3: val = 2; break;
                }
            }
            delay(1);
        }
    }
}

void gpxStart() //faili alustamine
{
    fileName();
    File logFile = SD.open(filename, FILE_WRITE); //faili avamine
    logFile.println(F("<gpx version=\"1.1\">"));
    logFile.println(F("<trk>"));
    logFile.println(F("<trkseg>"));
    logFile.close(); //faili sulgedes toimub kirjutamine puhvrist
    delay(10);
}

void dataWrite() //faili andmete kirjutamine
{
    unsigned long currentMillis = millis();
    if(currentMillis - t >= interval) {
        t = currentMillis;
        if(gps.location.isValid() && gps.location.isUpdated())
        {
            File logFile = SD.open(filename, FILE_WRITE);
            logFile.print(F("<trkpt lat=\""));
            logFile.print(gps.location.lat(),6);
            logFile.print(F("\\" lon=\""));
            logFile.print(gps.location.lng(),6);
            logFile.println(F("\\">"));
            logFile.print(F("<ele>"));
            logFile.print(gps.altitude.meters());
        }
    }
}

```

```

        logFile.println(F("</ele>"));
        logFile.print(F("<time>"));
        logFile.print(gps.date.year());
        logFile.print(F("-"));
        if(gps.date.month() < 10) logFile.print(F("0"));
        logFile.print(gps.date.month());
        logFile.print(F("-"));
        if(gps.date.day() < 10) logFile.print(F("0"));
        logFile.print(gps.date.day());
        logFile.print(F("T"));
        if (gps.time.hour() < 10) logFile.print(F("0"));
        logFile.print(gps.time.hour());
        logFile.print(F(":"));
        if (gps.time.minute() < 10) logFile.print(F("0"));
        logFile.print(gps.time.minute());
        logFile.print(F(":"));
        if (gps.time.second() < 10) logFile.print(F("0"));
        logFile.print(gps.time.second());
        logFile.print(F("Z"));
        logFile.println(F("</time>"));
        logFile.println(F("</trkpt>"));
        logFile.close();
        delay(10);
    }
}
}
void gpxEnd() //faili lopetamine
{
    File logFile = SD.open(filename, FILE_WRITE);
    logFile.println(F("</trkseg>"));
    logFile.println(F("</trk>"));
    logFile.println(F("</gpx>"));
    logFile.close();
    delay(10);
}
void fileName() //failinime valik ja kontroll
{
    int n = 0;
    snprintf(filename, sizeof(filename), "log%02d.gpx", n);
    while(SD.exists(filename)) {
        n++;
        snprintf(filename, sizeof(filename), "log%02d.gpx", n);
    }
}
void gpsDelay()
{
    while(!gps.altitude.isValid())
    {
        getData();
        mode();
        delay(100);
    }
}

```



```

    }
    val = 0;
}
void LCDWrite()
{
    u8g.firstPage();
    do {
        draw();
    } while( u8g.nextPage() );
}
void draw()
{
    u8g.setFont(u8g_font_freedoomr25n);    //kirjastiili maaramine
    u8g.setPrintPos(22,46); //kuvamise positsiooni maaramine
    if(gps.speed.kmph() >= 1.0) {u8g.print(gps.speed.kmph());}
    if(gps.speed.kmph() < 1.0) { u8g.print(F("0.00")); }
    u8g.setFont(u8g_font_fub11n);
    u8g.setPrintPos(1,64);
    u8g.print(gps.altitude.meters());
    u8g.setPrintPos(86,11);
    if(gps.time.hour()+2 < 10) { u8g.print(F("0")); }
    //u8g.print(gps.time.hour()+2);           //talveaeg
    u8g.print(gps.time.hour()+3);           //suveaeg
    u8g.print(F(":"));
    if(gps.time.minute() < 10) { u8g.print(F("0")); }
    u8g.print(gps.time.minute());
    u8g.setPrintPos(2,11);
    if(val==3)
    {
        u8g.print(F("S"));
    }
    else
    {
        u8g.print(F("-"));
    }
}
void LCDStart()
{
    u8g.firstPage();
    do
    {
        u8g.setFont(u8g_font_freedoomr25n);
        u8g.setPrintPos(22,46);
        u8g.print(F("0.00"));
    } while( u8g.nextPage() );
    delay(10);
}

```