

Viljandi Gümnaasium

**CanSat-i platvormi arendamine ja katsetamine  
koduste vahenditega**  
Uurimistöö

Autor Ergo Adams, III MF

Juhendaja Indrek Sünter, PhD

Viljandi 2021

# SISUKORD

SISSEJUHATUS	3
1. CANSAT JA SELLE ARENDAMINE	5
1.1 CanSat	5
1.2 CanSat-i alamosad	6
1.3 Trükkplaat	8
1.4 Ilmastikuõhupall	9
2. METOODIKA	10
2.1 Praktiline osa	10
2.1.1 CanSat-i nõuded	10
2.1.2 Riskid CanSat-i arendamisel	11
2.1.3 Riskid CanSat-i missioonil	12
2.1.4 Alamosade detailide valik	13
2.1.5 Struktuur ja disain	16
2.1.6 Plokkskeemi disain	18
2.1.7 Elektriskeemi disain	19
2.1.8 Trükkplaadi disain ja jootmine	20
2.1.9 Tarkvara arendamine	24
2.2 Mõõtmised	25
3. TULEMUSED	29
4. JÄRELDUSED	30
KOKKUVÕTE	31
SUMMARY	32
KASUTATUD ALLIKAD	33
LISAD	34
Lisa 1. CanSat-i aluskood	

## SISSEJUHATUS

Sputniku orbiidile jõudmisest saati (1957) on endiselt satelliitide orbiidile lennutamiseks kasutusel raketid. Samuti on kosmoseraketi start jätkuvalt kulukas ja riskantne ettevõtmine. Stardi eel jälgitakse pingsalt ilmastikku (tuul, äike, äärmuslikud temperatuurid), et vältida liigseid riske. Selleks, et määrata stardi eel ilmastikutingimused kõrgemal atmosfääri kihtides, kasutatakse vahel ilmastikuõhupalle.

Umbes kord aastas toimuvad erinevates riikides võistlused CanSat-idele ehk pisikestele satelliitidele, mida kasutatakse kosmosetehnoloogia õpetamiseks. Ettemääratud aja jooksul on vaja ehitada CanSat, mis saadetakse väikese raketi abil umbes 3 km kõrgusele. Meeskondadele on ette antud, mis andurid peavad CanSat-i pardal olema ja on antud ka veel teatud ala enda vabalt valitud missiooni jaoks. Meeskonnad võivad üritada veel ka täpsusmaandumist, kus CanSat-i langemisel püütakse kontrollida kukkumise suunda näiteks tiibadega.

Uurimistöö käigus uuritakse, kuidas toimub CanSat-i arendamine ja kas CanSat-i arendamine on võimalik tavainimestele kättesaadavate vahenditega. Tulenevalt eesmärgist püstitati järgmised uurimisküsimused:

- 1) Kuidas arendada CanSat-i?
- 2) Kas ilmastikuõhupalli põhifunktsionaalsus mahuks CanSat platvormile?
- 3) Kas atmosfääri kõrgkihtide ilmastikutingimusi määrava CanSat-i arendamine on võimalik tavainimestele kättesaadavate vahenditega?

Antud teema valiti, kuna autor tunneb kosmose ja kosmosetehnoloogia vastu suurt huvi. Aastatepikkune raketi õhkuütõusude ja tehnoloogia arengu jälgimine viiski sellise teema valikuni.

Töö esimeses osas selgitatakse CanSat-i arendustööga seonduvaid mõisteid, mida kasutatakse ka antud uurimistöös. Samuti selgitatakse CanSat-i tööpõhimõtteid. Töö teises osas kirjeldatakse uurimismeetodit. Kolmandas osas on välja toodud töö tulemused, millele järgneb analüüs.

# 1. CANSAT JA SELLE ARENDAMINE

## 1.1 CanSat

CanSat on sondrakettide last, mida kasutatakse kosmosetehnika õpetamiseks. CanSat-i tehnoloogia sarnaneb väikesatelliitidega. CanSat-i võistlustel peab satelliit mahtuma tavalisse plekkpurki diameetriga 66 mm ja kõrgusega 115 mm. Satelliidi mass peab jääma alla 350 grammi. CanSat-i purgist võivad väljaspool olla antennid juhul, kui satelliidi diameeter ei suurene enne õhkutõususeadmest eemaldumist. CanSat-id on hea viis kosmosetehnika õpetamiseks. Arendamise madal maksumus, vähene ajakulu ja disaini lihtsus võrreldes suuremate satelliitidega teevad CanSat-id õpilastele parimaks esimeseks sammuks kosmosetehnoloogias.<sup>1</sup>

Võistlustel liigitatakse CanSat-id kolme kategooriasse: telemeetria, maandumine ja vaba.

Telemeetria kategooriasse kuuluvate satelliitide eesmärgiks on missiooni jooksul koguda ja edastada andmeid maajaama, kus neid saab töödelda reaalsajas. Telemeetria satelliitidel ei pöörata tähelepanu kontrollitud maandumisele, vaid andmete kogumisele. Enamlevinud andmeliigid on õhurõhk, temperatuur, positsioon ja pildid.

Maandumise satelliitide eesmärgiks on kontrollitud maandumine võimalikult lähedale korraldajate poolt märgitud alale. Satelliit kogub GPS-andmeid ja teiste andurite näite, mida töödeldes arvutab tarkvara välja vajalikud manöövrid, mida tehes lennu missioon täidetakse.

---

<sup>1</sup> Walker, R, Galeone, P, Page, H, Castro, A, Emma, F 2010. ESA Hands-on Space Education Project Activities for University Students: Attracting and Training the Next Generation of Space Engineers.

<http://www.ieec.uned.es/Investigacion/Educon2010/SearchTool/EDUCON2010/papers/2010S10B01.pdf>

(16.01.2021)

Enamasti on maandumise satelliitidel kas tüüritav langevari või tiivad, millega on võimalik langemise suunda muuta.

Vabaklassi kuuluvad CanSat-id, mis ei liigitu eelmise kahe kategooria alla. Vabaklassi satelliidid katsetavad tavaliselt tehnoloogiaid või komponente, mida varem CanSat-idel kasutatud pole.

## 1.2 CanSat-i alamosad

CanSat-i põhilisteks alamosadeks on elektrisüsteem, raadioside, andmete edastamine ja salvestamine, missioonilast ja pardaarvuti. Iga alamosa juures on välja toodud erinevad valikuvõimalused ja nende plussid ja miinused.

Elektrisüsteem on CanSat-i üks tähtsamaid alamosasid, ilma selleta lakkab kogu CanSat-i töö. Üks enamlevinud akutüüp on LiPo (liitium-polümeer) aku. LiPo akud on massiühiku kohta suure mahutavusega ja aku nimipinged on CanSat-ile sobivad (3,3 V või 5 V). LiPo akud aga paisuvad vaakumis, võivad valesti kasutamisel lihtsasti süttida ja vajavad eraldi laadijat. LiIon (liitium-ioon) akud on võrreldes LiPo akudega väiksema mahutavusega massiühiku kohta. Ka LiIon akud on saadaval sobiliku nimipingega. Küll vajavad aga ka LiIon akud eraldi laadijat. Tavalised leelispatareid (AA, AAA) ei vaja eraldi laadijat ja on leitavad pea igas poes. Leelispatarei miinusteks on kehv mahutavus massiühiku kohta ja madal nimipinge (1,5 V).

Raadioside on vajalik andmete reaajas kätte saamiseks. Ilma raadiosideta on pea võimatu satelliiti peale missiooni üles leida. Raadioside võimaldab CanSat-iga kogu missiooni vältel ühendust hoida. Enamasti kasutatakse kolme põhilist raadiomodulatsiooni: AM, FM ja PM. AM ehk amplituudmodulatsioon on raadiosignaali, kus info edastatakse raadiosignaali amplituudi muutustena. AM signaali on aga signaali müratundlikkuse tõttu raske vastu võtta ja sellest aru saada<sup>2</sup>. FM ehk sagedusmodulatsiooni puhul edastatakse info signaali lainepikkuse muutustena. FM signaali edastamist segab müra vähem, küll aga on vaja laiemat

---

<sup>2</sup> Electronics Notes: Amplitude Modulation, AM

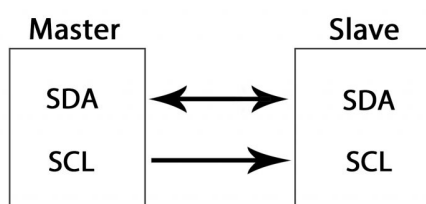
<https://www.electronics-notes.com/articles/radio/modulation/amplitude-modulation-am.php> (16.01.2021)

sageduste vahemikku. PM ehk faasimodulatsiooni puhul edastatakse info elektromagnetlaine faasi muutustega. PM signaali on tehniliselt keerulisem vastu võtta, küll aga segab müra signaali vähem kui amplituud- või sagedusmodulatsiooni korral. Kõige populaarsem raadiosaatja ilmastikuõhupallide valdkonnas on MTX2, mida on varem edukalt kasutatud mitmetes õhkutõusudes (näiteks TITAN 1 missioon<sup>3</sup>). MTX2 kasutab signaali edastamiseks sagedusmodulatsiooni.

Tähtsamaid andmeid (GPS koordinaadid) edastatakse raadio teel. Harvemini edastatakse näiteks temperatuuri ja rõhku, mis muutuvad aeglasemini. Kõik andmed salvestatakse microSD kaardile.

Missioonilastiks on erinevad andurid, mille abil kogutakse missiooni jooksul andmeid, mis edastatakse raadioside abil maajaama ja salvestatakse ka SD-kaardile. Suhtlus anduritega käib üle erinevate andmesiinide.

I<sup>2</sup>C on andmesiin, millel on mitu võimalikku ülemat. Seda andmesiiini kasutatakse enamasti väikese kiirusega seadmete ühendamiseks. Andmesiin koosneb kahest ühendusest: andmesiin ja kellatakt. Ülem väljastab takti ja adresseerib alamaid, alamad aga võtavad takti vastu ja käituvad vastavalt aadressidele (vt joonis 1).<sup>4</sup>



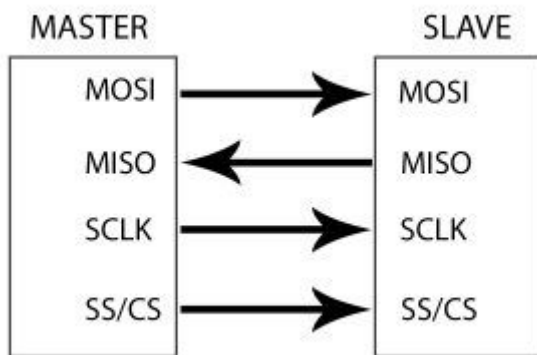
Joonis 1. I<sup>2</sup>C andmesiin. Allikas: Dcube Store Blog<sup>5</sup>

<sup>3</sup> Bombasaro, Emanuel 2015. TITAN 1. Design and Mission Documentation. [https://bombasaro.org/files/Titan1\\_Design\\_Mission\\_Doc.pdf](https://bombasaro.org/files/Titan1_Design_Mission_Doc.pdf) (16.01.2021)

<sup>4</sup> I2C Bus: I2C – What's That? <https://www.i2c-bus.org/> (16.01.2021)

<sup>5</sup> Dcube Store Blog: Difference between I2C and SPI (I2C VS SPI). <https://dcubestore.com/blog/difference-between-i2c-and-spi/> (16.01.2021)

SPI on järjestiksuhtluse standard, mida kasutatakse enamasti suhtluseks lühidistantsil. SPI eelis I<sup>2</sup>C ees on kordades suurem suhtluskiirus. Suhtlus käib nelja signaali kaudu: SS ehk alluvseadet valiv signaal, SCK ehk taktsignaali, MOSI ehk andmesignaali ülemalt alluvale ja MISO ehk andmesignaali alluvalt ülemale<sup>6</sup> (vt joonis 2).



Joonis 2. SPI andmesiid. Allikas: Dcube Store Blog<sup>7</sup>

Pardaarvuti ülesandeks on kontrollida kõikide andurite tööd ja saata andmeid üle raadioside maajaama. Pardaarvuti disaini ja töö käiku kirjeldatakse peatükis 2.1.9.

### 1.3 Trükkplaat

Trükkplaat ehk montaažiplaat võimaldab elektroonikakomponendid mehaaniliselt fikseerida ning elektriliselt omavahel ühendada. Trükkplaadi aluseks on klaaskiudleht, mille mõlemale poole on kantud õhuke vasekiht. Trükkplaadi rajad on freesitud arvuti poolt juhitud CNC pingiga. Töös kasutati ühekihilist kahepoolset trükkplaati. Täpsemalt trükkplaadi disainist peatükis 2.1.8.

---

<sup>6</sup> Embedded Staff 2002. Introduction to Serial Peripheral Interface.  
<https://www.embedded.com/introduction-to-serial-peripheral-interface/> (16.01.2021)

<sup>7</sup> Dcube Store Blog: Difference between I2C and SPI (I2C VS SPI).  
<https://dcubestore.com/blog/difference-between-i2c-and-spi/> (16.01.2021)



## 1.4 Ilmastikuõhupall

Ilmastikuõhupallid on õhupallid, mida kasutatakse mõõteseadeldiste saatmiseks atmosfääri kõrgematesse kihtidesse. Õhupall on valmistatud õhukesest hästi painduvast lateksmaterjalist. Õhupall täidetakse tavaliselt vesinikuga või heeliumiga. Ilmastikuõhupallid võivad jõuda umbes 30 km kõrgusele, mis hetkeks on õhupall paisunud rõhkude erinevuse tõttu nii palju, et ta lõhkeb.<sup>8</sup>

---

<sup>8</sup> Dabberdt, W, F, Shellhorn, R, Cole, H, Paukkunen, A, Hörhammer, J, Antikainen, V, 2003. Radiosondes. <http://www.radiopassioni.it/pdf/materialirsonde/Radiosondes.pdf> (16.01.2021)

## 2. METOODIKA

Uurimistöö protsess toimus kahes osas: praktiline osa ja andmete kogumine praktilises osas valminud CanSat-i abil. Praktilise osa käigus valmis CanSat ja andmete kogumise käigus tõestati CanSat-i töövõimet.

### 2.1 Praktiline osa

Praktiline osa koosneb järgmistest alamosadest: CanSat-i nõuete seadistamine, riskianalüüs, alamosade detailide valiku tegemine, CanSat-i elektroonikasüsteemi plokkskeemi tegemine, CanSat-i elektriskeemi disain, CanSat-i trükkplaadi disain, CanSat-i trükkplaadi jootmine, CanSat-i programmeerimine ja katsetamine.

#### 2.1.1 CanSat-i nõuded

CanSat-i nõuded on grupeeritud CanSat-i põhiosade järgi: missioon, struktuur ja elektroonika. Nõuete määramisel tugineti CanSat-ide 2020-2021. aasta juhendile<sup>9</sup>. Määratud said järgmised nõuded:

##### 1) Missioon

- a) CanSat peab kerkimise ajal tegema rohkem kui 1 pildi. Eelistatav oleks vähemalt 1 pilt ka kõrguselt ~10 km;
- b) Kaamera teeb pilte horisondist;
- c) Kaamera teeb värvilisi pilte;
- d) Kaamera teeb vähemalt VGA (640 x 480 pikslit<sup>2</sup>) lahutusega pilte;
- e) Mõõdame temperatuuri vähemalt iga 3 sekundi tagant. Piisavalt tihedalt temperatuuri mõõtes saame hiljem joonistada sujuvama graafiku;

---

<sup>9</sup> ESA - The European Space Agency: CanSat 2020-2021 Guidelines.  
[https://cansat.esa.int/wp-content/uploads/2020/09/CanSat\\_guidelines\\_2020\\_2021.pdf](https://cansat.esa.int/wp-content/uploads/2020/09/CanSat_guidelines_2020_2021.pdf) (16.01.2021)

- f) Mõõdame temperatuuri vähemalt  $\pm 1$  kraadi täpsusega (suurema täpsuse korral oleks anduri kalibreerimine keerulisem);
- g) Mõõdame rõhku vähemalt iga 3 sekundi tagant;
- h) CanSat peab elama üle maandumise, eriti oluline on tagada SD kaardi säilimine, millele on salvestatud andmed;
- i) CanSat peab olema pärast maandumist leitav;
- j) Salvestab andmed mälukaardile. Kõiki mõõdiseid ei jõua üle raadio edastada, ning osa raadio teel edastatavatest pakettidest läheb kaduma (sõltub muuhulgas CanSat-i orientatsioonist)

## 2) Struktuur

- a) Peab vastama CanSat-i nõuetele;
- b) Purgi kõrgus peab olema 115 mm ja diameeter 66 mm;
- c) CanSat-i kaal peab olema 300-350 g;
- d) Langevari peab vastu pidama 500 N jõule;
- e) Peab olema koht õhupalli kinnitamiseks;
- f) CanSat peab vastu pidama juhuslikku vibratsiooni võimsuse spektraaltihedusega 20 g

## 3) Elektroonika

- a) On vaja kasutada raadioamatöörade raadiosagedusi (võimaldab raadioamatööridel CanSat-i jälgida ning töö autorile CanSat-i koordinaate edastada);
- b) Andmed peavad olema raadioamatööridele lihtsasti tõlgendavad;
- c) CanSat peab töötama vähemalt 4 tundi üleslennutamise hetkest (peaks olema piisav langenud CanSat-i üle leidmiseks);
- d) Aku peab olema lihtsasti kättesaadav ja eemaldatav;
- e) Peab olema kergesti ligipääsetav pealüliti kogu elektroonika välja lülitamiseks;
- f) Peab olema GPS;
- g) Peab GPS koordinaate üle raadio edastama;
- h) CanSat võiks piiksuda, et oleks maandumise järel lihtsamini leitav

### 2.1.2 Riskid CanSat-i arendamisel

Ülevaade potentsiaalsetest riskidest, nende tõsidusest ja esinemise tõenäosusest CanSat-i arendamise käigus (vt tabel 1).

Tabel 1. Riskid CanSat-i arendamisel

Risk	Lahendus	Tõenäosus	Riski suurus
Arendamine võtab liiga kaua aega	Mõne anduri ära jätmine	väike	väga väike

<b>Risk</b>	<b>Lahendus</b>	<b>Tõenäosus</b>	<b>Riski suurus</b>
Arendamine läheb liiga kulukaks	Vaja otsida kas odavamad komponendid või mõni komponent üldse ära jätta	väike	väga väike
CanSat-i kaal on liiga suur/liiga väike	Kui kaal on liiga väike, siis peab panema purgile midagi raskuseks juurde. Kui aga kaal on liiga suur, siis peab mõne sensori ära võtma	väga väike	väga väike

### 2.1.3 Riskid CanSat-i missioonil

CanSat-i missioonil võimalike riskidega arvestamiseks sai tehtud tabel riskidest, lahendustest neile riskidele, riski esinemise tõenäosustest ja riski suurustest (vt tabel 2).

Tabel 2. Riskid CanSat-i missioonil

<b>Risk</b>	<b>Lahendus</b>	<b>Tõenäosus</b>	<b>Riski suurus</b>
Langevari ei avane	Oleneb sellest kuhu kukub, andmed on mälukaardil ja osasid edastatakse ka jooksvalt (GPS koordinaadid)	väike	keskmine
CanSat kukub lompi	Enne üleslennutamist on vaja vaadata ilmateadet ja tuule suunda, et ei lendaks kuskile kättesaamatusse kohta	väike	keskmine
Aku saab tühjaks	On vaja vaadata heeliumi õhupalli lendamise aeg, CanSat-i aku peab töötama kauem, et oleks aega CanSat-i juurde jõuda. Ülejäänud elektroonika saab välja lülitada, aga CanSat peaks ikkagi olema üles leitav	väike	keskmine
Raadiosaatja ei tööta	CanSat-i ilma selleta väga raske üles leida. Peab raadiosidet piisavalt testima	väike	suur
CanSat pöörleb nii kiiresti, et ei saa ühtegi pilti	Võib üritada kuidagi pöörlemist takistada (stabilisaatorid), aga see võtab jälle rohkem ruumi. Vaja on veel ka üle vaadata kaamera valgustundlikkus ja säriaeg, et ei tuleks liiga tumedad/heledad pildid	keskmine	keskmine

Risk	Lahendus	Tõenäosus	Riski suurus
Mälukaardi ühendus ei toimi	Saab saata olulisemaid andmeid üle raadioside, näiteks temperatuur ja rõhk.	väike	keskmine
Andurite ühendus ei toimi	CanSat-i korduv katsetamine	väike	keskmine
Tekib lühis	Purki saab katta näiteks lakiga, juhtmetele saab termokahanevad rüüsid ümber panna	väike	suur
Tarkvaras on viga ja ei toimu mõõtmisi	CanSat-i korduv katsetamine	väike	keskmine

### 2.1.4 Alamosade detailide valik

CanSat-i alamosade valimisel arvestati erinevate komponentide hindasid, kättesaadavust ja varasemaid näiteid kasutusest. CanSat-is on kasutusel temperatuuri- ja õhurõhuandur, GPS, kaamera, güroskoopiline andur, raadioside transiiver ja mikrokontrolleriga arendusmoodul.

Temperatuuri- ja rõhuanduri valikuid oli kolm, millest osutus valituks MPL115A2, mis mõõdab nii temperatuuri kui õhurõhku korraga. Nii ei ole vaja kahte erinevat andurit, mis tähendab kokkuhoidu ruumis ja erinevate liideste arvus (vt tabel 3 ja 4).

Tabel 3. Temperatuuriandurite valik

Nimi	Protokoll	Mõõtmed	Kaal	Hind
SparkFun TMP102	I <sup>2</sup> C	25 mm x 25 mm x 2,5 mm	1,3 g	5 €

Table 4. Rõhuandurite valikud

Nimi	Protokoll	Mõõtmed	Kaal	Hind
Adafruit BMP280	I <sup>2</sup> C või SPI	19.2 mm x 17.9 mm x 2.9 mm	1,3 g	10 €

Nimi	Protokoll	Mõõtmed	Kaal	Hind
MPL115A2	I <sup>2</sup> C	5 mm x 3 mm x 1,2 mm	0,61 g	8 €

Kaameraks sai valitud antud mikrokontrolleriga enimkasutatud OV7670 (vt tabel 5). Töö käigus sai katsetatud nii OV7670 ilma mälua ja koos mälua versiooni. Mälua versioon võimaldab teoorias salvestada pildi mälua ja lugeda seda endale sobiva kiirusega. Ilma mälua kaamerale on vaja lugeda pilt sensorilt säritamise ajal.

Tabel 5. Kaamerate valik

Nimi	Vaatenurk	Lahutusvõime	Voolu- tarve	Protokoll	Hind
OV9655	50°	1280 x 1024 pikslit <sup>2</sup>	90 mW	I <sup>2</sup> C	10 €
OV7670	48°	640 x 480 pikslit <sup>2</sup>	60 mW	I <sup>2</sup> C	9,5 €
MT9D11 1		1600 x 1200 pikslit <sup>2</sup>	75 - 125 mW	I <sup>2</sup> C	9,5 €
OV5642	48°	2592 x 1944 pikslit <sup>2</sup>		Serial Camera Control Bus, Arduino ja RPi jaoks vaja vahele eraldi shieldi	26 €

GPS sensori valimisel osutus valituks kõige soodsam GPS G28U7FTTL (vt tabel 6).

Tabel 6. GPS sensorite valik

Nimi	Voolutarve	Mõõtmed	Kaal	Hind
SainSmart NEO-6M		39 mm x 26 mm	23 g	30 €
G28U7FTTL	10mA	28 mm x 28 mm x 8.5 mm	30 g	12 €
Adafruit Ultimate Breakout	20mA	25.5mm x 35mm x 6.5mm	8,5 g	35 €

Güroskoopilise anduri valimine toimus taaskord hinna ja saadavuse põhjal, kuna kõigi andurite puhul oli dokumentatsioon kasutamiseks piisav. Valiti andur MPU-6050 (vt tabel 7).

Tabel 7. Güroskoopiliste andurite valik

Nimi	Protokoll	Voolutarve	Mõõtmed	Kaal	Hind
ADXL345	SPI või I <sup>2</sup> C	40 µA	3 mm x 5 mm x 1 mm		18 €
ADXL335		350 µA	4 mm × 4 mm × 1.45 mm		15 €
MPU-6050	I <sup>2</sup> C	3,9 mA	21,2 mm x 16,4 mm x 3,3 mm	2,1 g	7 €

Pardaarvuti valimine käis saadaval olevate liideste ning voolutarbe põhjal. Raspberry Pi pakub suuremat arvutusvõimsust, mistõttu kaasneb ka suurem voolutarve ja lühem akude kestvus. Käesoleva CanSat-i missiooni saab läbi viia ka vähemate ressurssidega. Valituks osutus Arduino Nano mikrokontroller, kuna tollel on piisav arv väljaviike kõikide andurite ühendamiseks. Arduino Nano on mõõtmetelt ka palju väiksem, kui Arduino Uno (vt tabel 8).

Tabel 8. Pardaarvutite valik

Nimi	Mõõtmed	Kaal	Toitepinge	Voolutarve	Väljaviigud	Hind
Arduino Uno	68,6 mm x 53,4 mm	25 g	5 V	50 mA	14 digital, 6 analog pin-i 1 SPI, 2 I <sup>2</sup> C	20 €
Raspberry Pi 2 Model B	85,6 mm x 53,98 mm	45 g	5 V	600 mA (3 W)	40 GPIO pin-i 1 SPI, 2 I <sup>2</sup> C	46 €
Raspberry Pi 3 Model B+	85 mm x 56 mm	45 g	5 V	459 mA (2,3 W) idle	40 GPIO pin-i, 1 SPI, 1 I <sup>2</sup> C	40 €

Nimi	Mõõtmed	Kaal	Toitepinge	Voolutarve	Väljaviigud	Hind
Arduino Nano	18 mm x 45 mm	7 g	5 V	19 mA	8 analog IN, 22 digital (6 PWM)	27 €

Komponendid said tellitud Ebayst ja Oomipoest. Paljud detailid olid saadaval ostukeskkonnas Ebay, mis võimaldas osa komponente soetada soodsamalt.

### 2.1.5 Struktuur ja disain

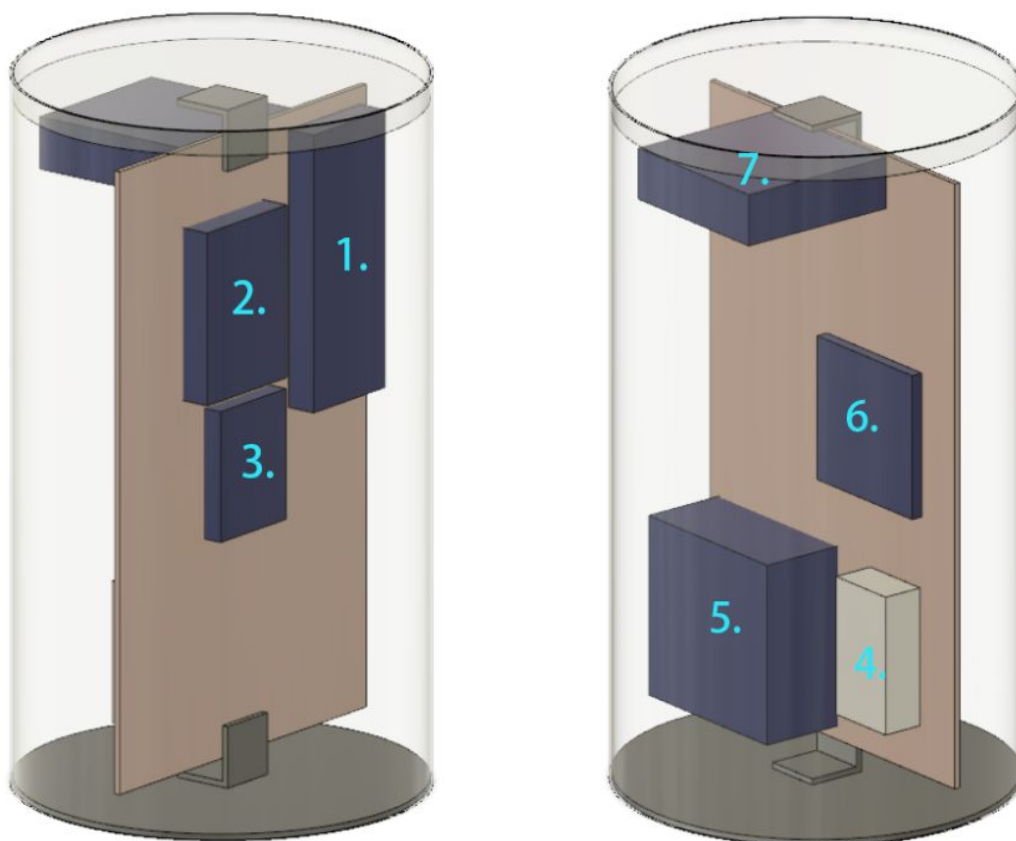
Valitud elektroonikakomponentide põhjal valmis 3D-mudel, millel on näha erinevate komponentide esialgselt planeeritud paiknemine ja gabariitmõõdud. Selline mudel (vt joonis 3) lihtsustab edaspidist trükkplaadi disaini, näidates, millised komponendid mahuvad üksteise kõrvale ja millised mitte. 3D-mudel valmis programmis Fusion 360<sup>10</sup>. Mudelil on numbritega tähistatud järgmised komponendid:

- 1) Arduino Nano
- 2) SD-kaardi moodul
- 3) GÜroskoopiline sensor
- 4) Raadioside moodul
- 5) Kaamera
- 6) Õhurõhu ja temperatuuri andur
- 7) GPS

Struktuur valmis plekkpurgist (vt joonis 4), pleksiklaasist kattest ja pisikestest alumiiniumist nurkadest. Pleksiklaasist ring kinnitub nurkadega trükkplaadi ja trükkplaat purgi külge. Pleksiklaasist katte külge on kinnitatud ka GPS (vt joonis 5). Plekkpurgi servades on kasutatud maalriteipi, et vältida vigastusi teravate äärte tõttu.

<sup>10</sup> Autodesk: Fusion 360. <https://www.autodesk.com/campaigns/education/student-design> (16.01.2021)

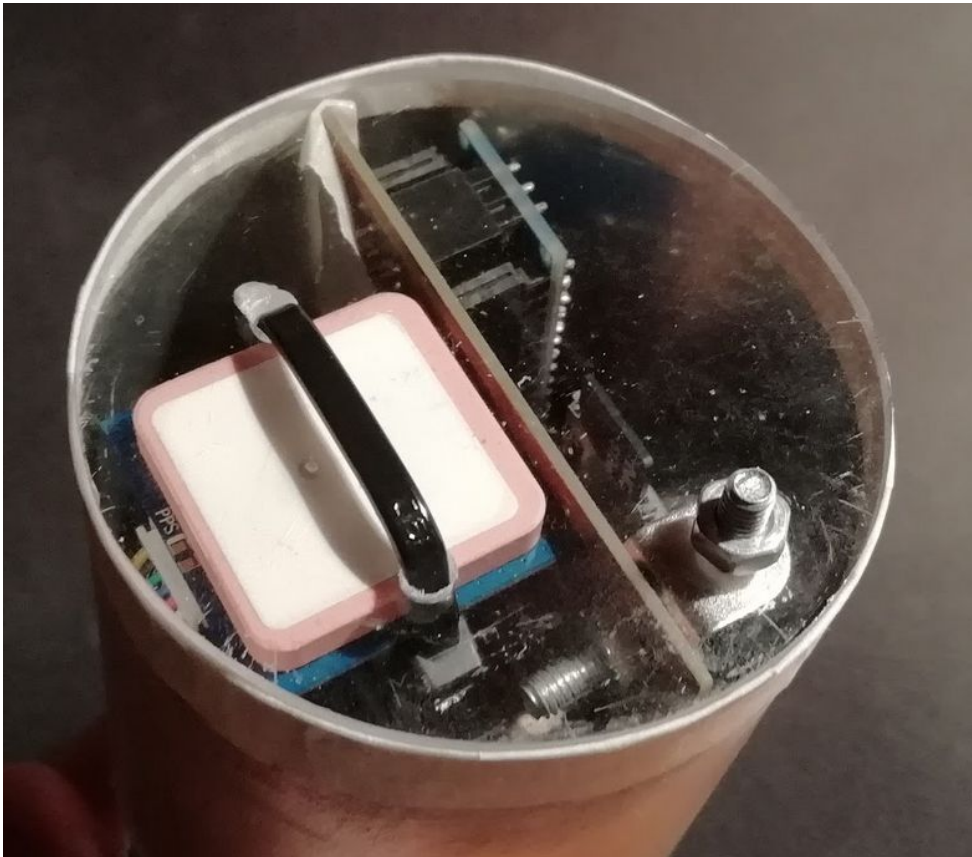




Joonis 3. Vasakul struktuuri 3D-mudeli tagumine pool, paremal eesmine pool



Joonis 4. Näha on antenn ja kaamera. Plekkpurgi teravad servad on kaetud maalriteibiga

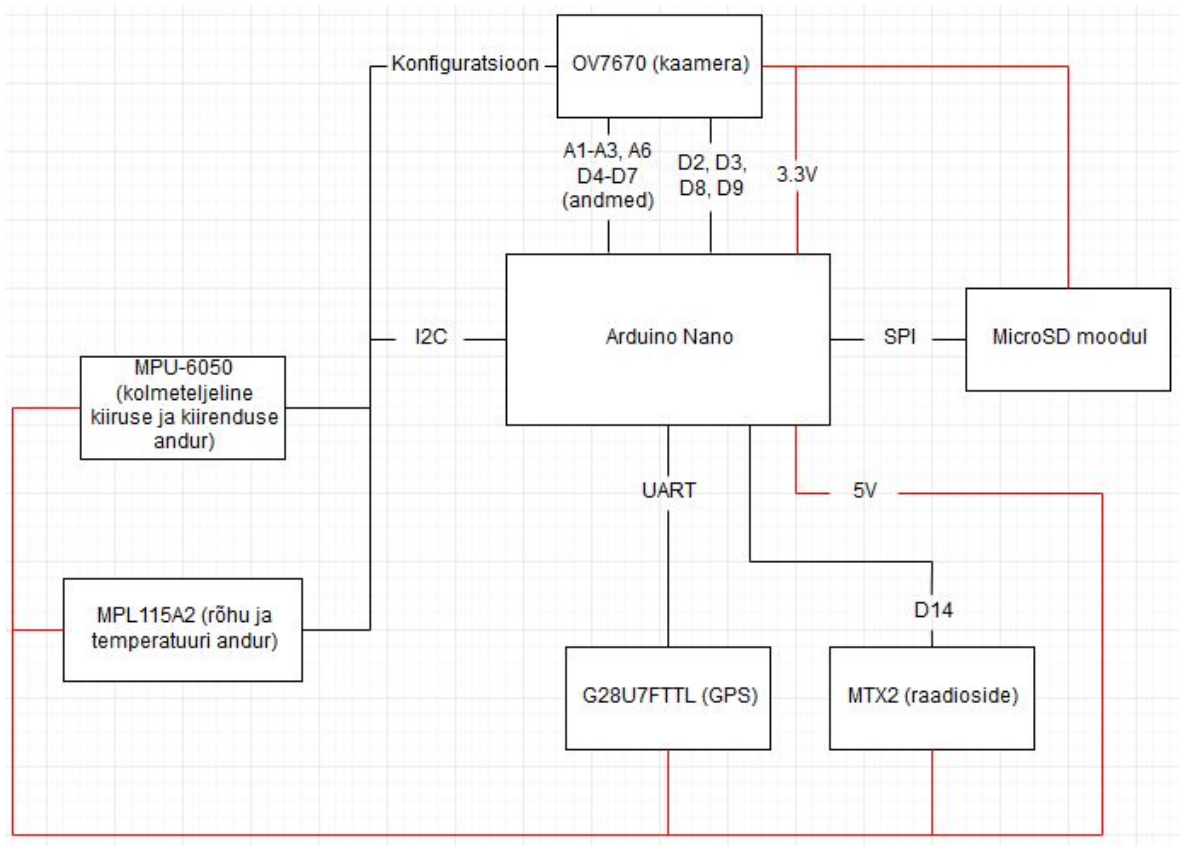


Joonis 5. Pleksiklaasist kate koos GPS sensoriga

### 2.1.6 Plokkskeemi disain

Plokkskeemi kasutatakse elektrisüsteemi lihtsustatud kujutamiseks. Plokkskeem näitab, mis komponentidest CanSat koosneb, kuidas need komponendid elektriliselt toidetud on ning missugust andmesiini kasutatakse nendega suhtlemiseks.

CanSat-i plokkskeemi disainimisel tuli arvestada valitud komponentide andmesiinide ja mikrokontrolleri väljaviikude saadavusega (vt joonis 6).



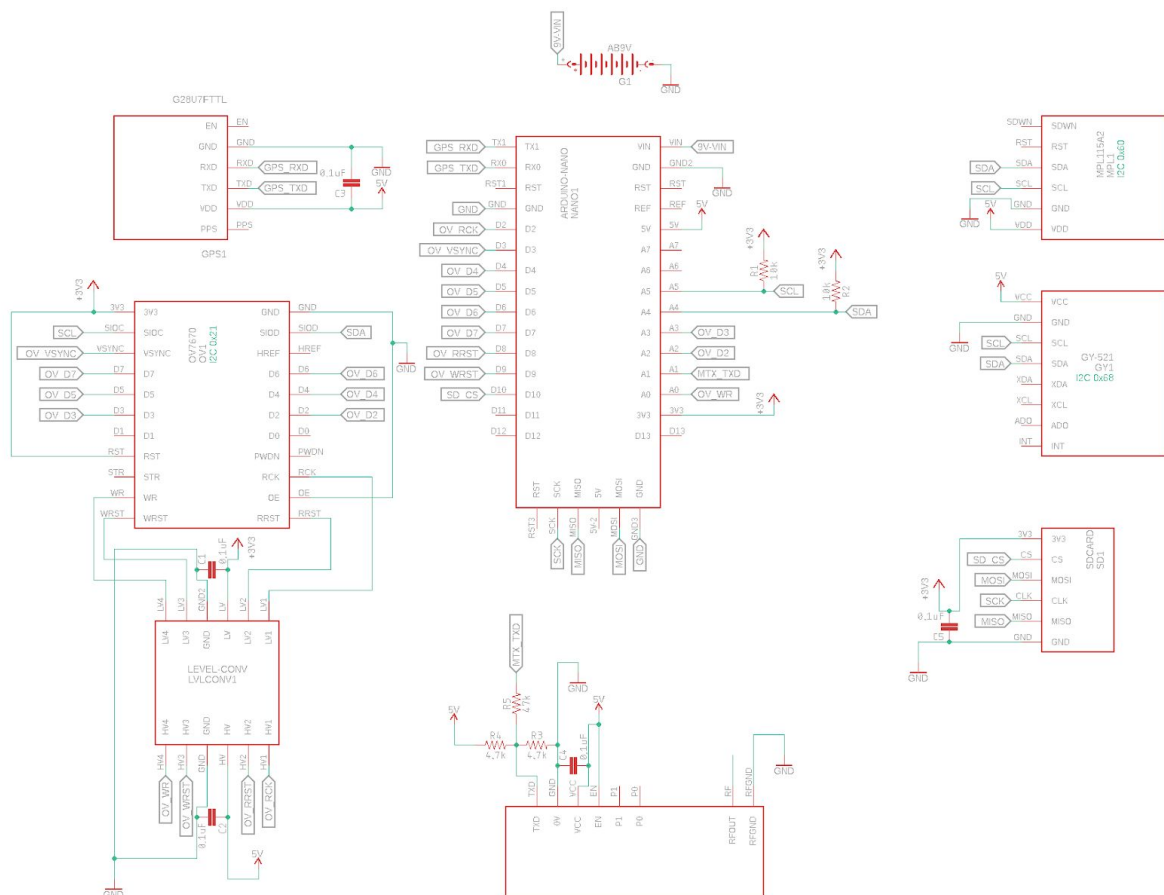
Joonis 6. CanSat-i elektrisüsteemi plokskeem

### 2.1.7 Elektriskeemi disain

Elektriskeem näitab võrreldes plokskeemiga täpsemalt, kuidas elektroonikakomponendid omavahel ühendatud on. Elektriskeemil on näidatud ka kõige väiksemad detailid.

Elektriskeemi disainimisel tuleb arvestada kõikide elektroonikakomponentidega (vt joonis 7). Elektriskeemi disainimiseks kasutati elektriskeemide ja trükkplaatide disainimise tarkvara nimega EAGLE.<sup>11</sup>

<sup>11</sup> Autodesk: EAGLE. <https://www.autodesk.com/products/eagle/overview> (16.01.2021)



Joonis 7. CanSat-i trükkplaadi elektriskeem

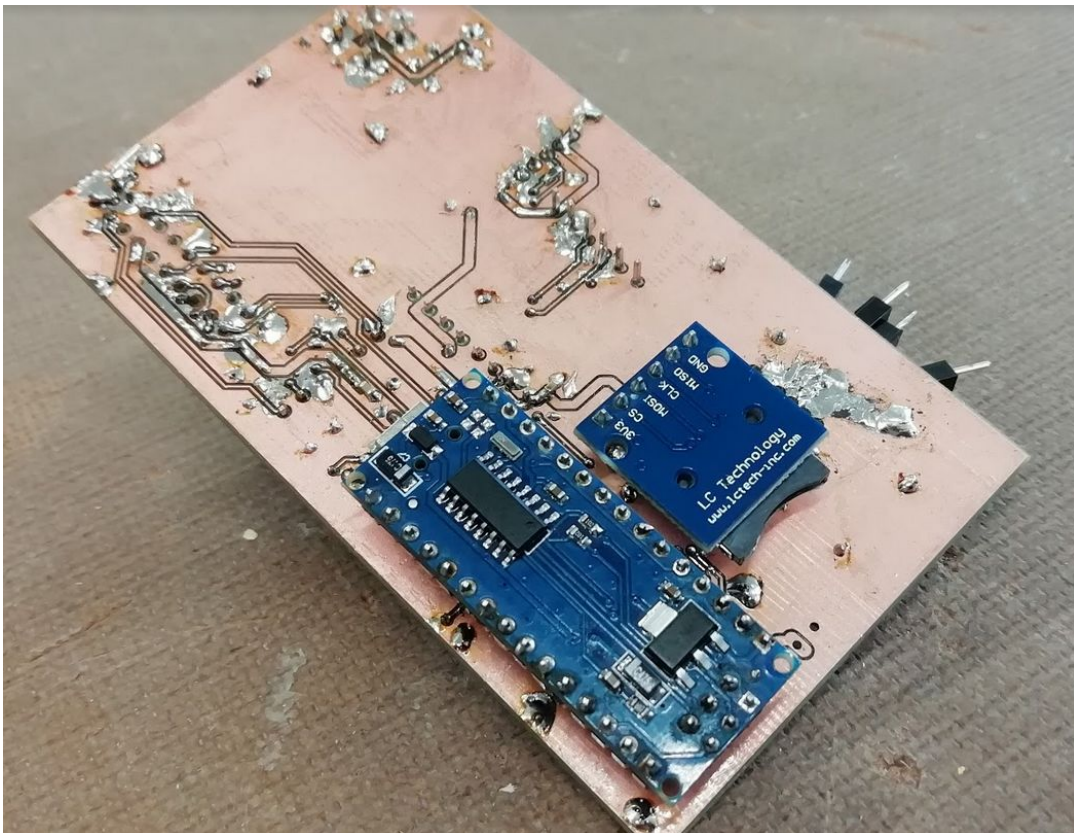
## 2.1.8 Trükkplaadi disain ja jootmine

Trükkplaadil paiknevad pisemad elektroonikakomponendid (takistid ja kondensaatorid) on pindpaigaldatud komponendid. Pindpaigaldatud komponendid kinnituvad plaadi ühele poolele ega vaja läbiviikuseid.

Trükkplaadi disainimise juures tuleb arvestada komponentide suurusel ja väljaviikude paiknemisega. Radade laius peab olema vastav raja liigile: signaalirajad kitsamad, toiterajad laiemad. Radade omavahelised vahed peavad olema piisavad, et jootmisel ei tekiks probleeme lühistega. Rajad ei tohi olla trükkplaadi äärele liiga lähedal, vastasel juhul võib ilmuda probleeme radade freesimisega. Trükkplaadi üheks tähtsamaks osaks on *ground plane* ehk

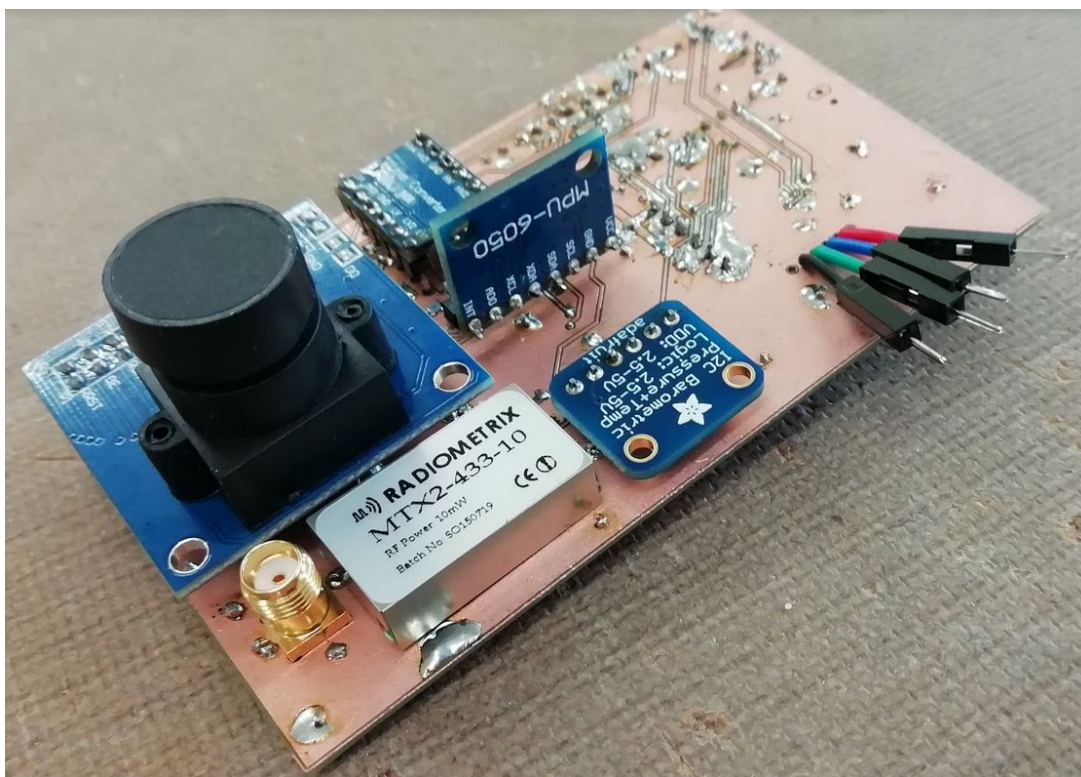
maapind, seega tuli leida optimaalne arv läbiviikused, et tagada võimalikult väike takistus maaga ühendatud komponentide ja maapinna vahel.

Trükkplaadi disainimise käigus valmis kaks trükkplaati. Esimene valminud trükkplaat (vt joonis 8 ja 9) oli liiga kitsaste radadega, et jootmist primitiivsemate tööriistatega mugavalt sooritada. Trükkplaadi katsetamise käigus tekkinud lühise tõttu põles nii trükkplaadil kui ka mikrokontrolleril läbi paar takistit ja diood. Hävinud diood mikrokontrolleri plaadil oli USB-toite diood, mis ei lase mikrokontrolleri välist toidet arvuti poole. Kuna CanSat-il ei ole võimalust mõlemat toiteallikat korraga kasutada, asendati diood mikrokontrolleril juhtmega (vt joonis 10). Lühise tekkimise põhjuseks olid liiga kitsad rajad, mis raskendasid jootmist. Trükkplaadi teine disain oli seega laiemate radadega ja komponendid said paigutatud üksteisest kaugemale (vt joonis 11 ja 12).

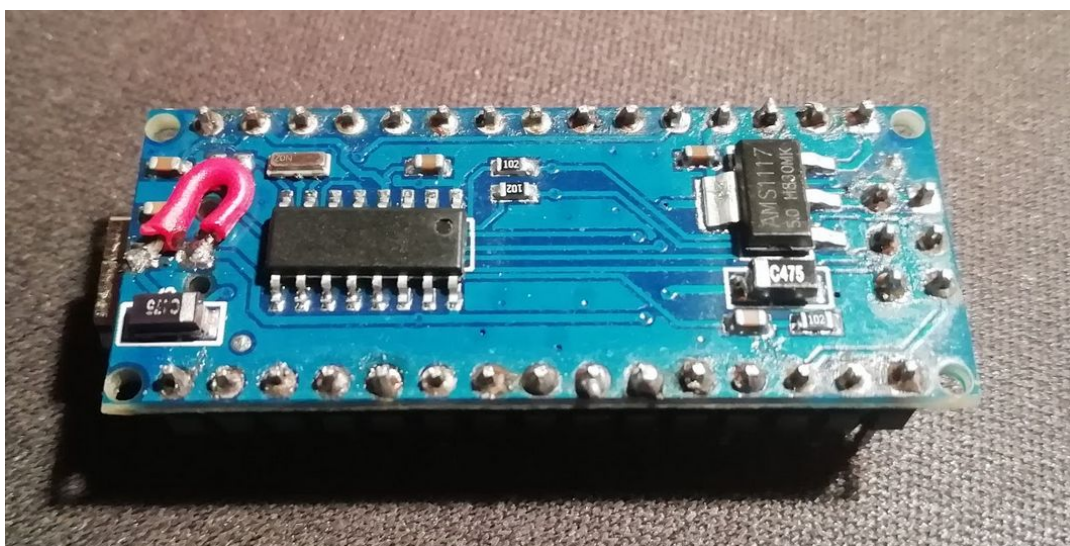


Joonis 8. Esimese valminud trükkplaadi tagumine pool

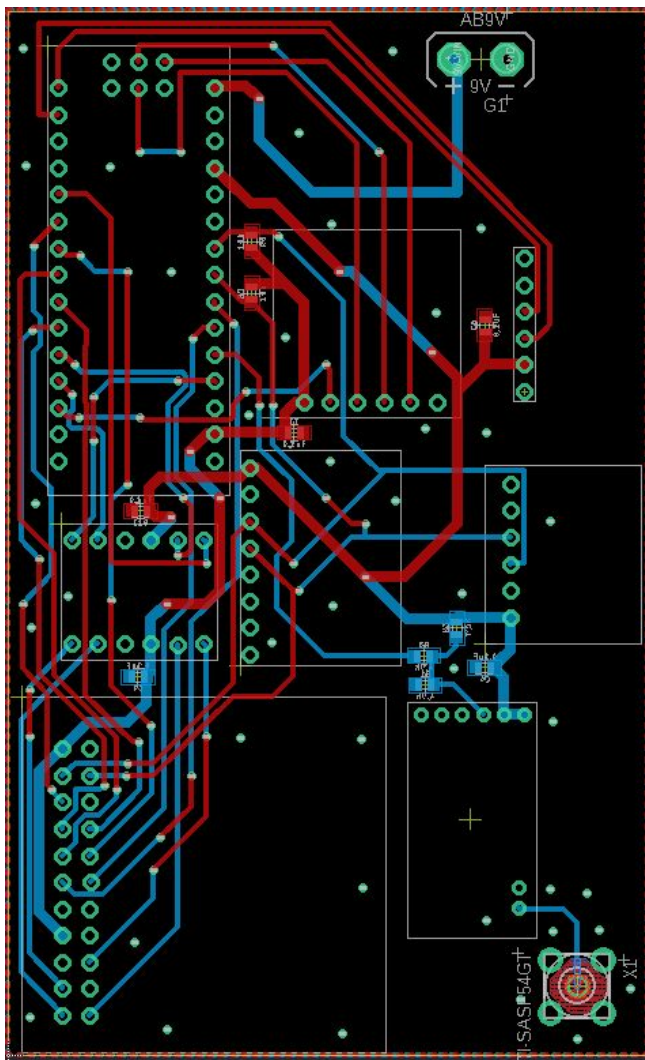




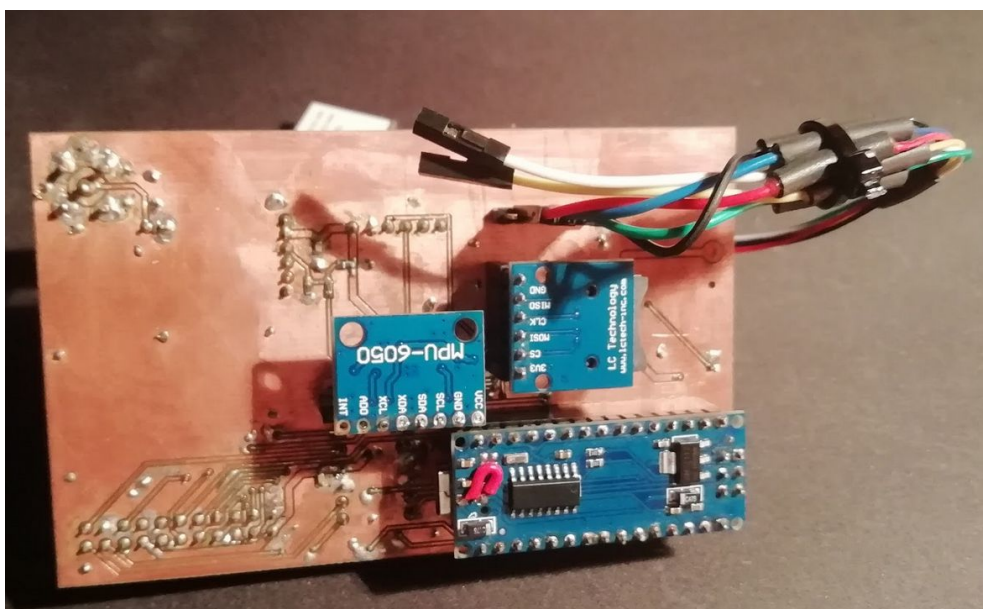
Joonis 9. Esimese valminud trükkplaadi eesmine pool, pildil näha kaamera, raadioside moodul, güroskoopiline sensor, õhurõhu ja temperatuuri andur ja ühendused GPS-ile



Joonis 10. Arduino Nano diodi asendamine juhtmega



Joonis 11. CanSat-i trükkplaadi disain



Joonis 12. Teine valminud trükkplaat koos komponentidega

### 2.1.9 Tarkvara arendamine

CanSat-i tarkvara peab juhtima kõikide satelliidi alamosade tööd. Tarkvarast on töö jooksul valminud kokku 22 erinevat versiooni. Esimesed neist koosnesid ainult üksikute sensorite näitude lugemistest, hilisemad juba mitme sensori lugemisest ja andmete salvestamisest ning edastamisest üle raadioside. Kõige rohkem erinevaid versioone on kaamera tarkvarast. Kuna töö jooksul on proovitud kahte erinevat kaamerat, siis on ka mõlema jaoks olnud eraldi tarkvara. Kuna elektroonikakomponentide valimise ajal ei olnud populaarseid kaameraid, mida oleks väljavalitud mikrokontrolleriga varem kasutatud, siis on olnud raske leida häid näiteid kaamera juhtimiseks. Seetõttu ei ole kaamera nõutud resolutsioonil töövõimeline, kuna puuduv dokumentatsioon ei võimalda kaamerat korrektselt konfigureerida.

Erinevate tarkvaraversioonide arendamise käigus on ilmnenud ka probleem valitud mikrokontrolleri mälu suurusega. Tarkvaras defineeritud muutujate ja programmi varustamiseks on vaja mälu. Valitud mikrokontrolleril on seda aga liiga vähe (30720 baiti programmimälu ja 2048 baiti dünaamilist mälu). Mälu puuduse tõttu on esile tulnud kordi, kus pardaarvuti programm ei hakka tööle, vaid mikrokontroller jääb kinni pidevasse restartide tsükklisse. Probleemi lahenduseks on olnud programmimälu kasutamine erinevate sõnede ja konstantide salvestamiseks, sõnede töötlemine säästlikumal viisil ja mitmete funktsioonide optimeerimine. Optimeeritud on andmete salvestamine SD kaardile ja raadioside funktsioon. Optimeerimisel on piiratud ujukoma arvude ja 32-bitiste ning 16-bitiste muutujate kasutust. Lisaks on välditud ja lihtsustatud erinevaid operatsioone sõnedega. Valminud tarkvara kompileerimisel hoiatatakse ikka vähese mälu probleemist, kuid tarkvara töötab.

Tarkvara on üles ehitatud järgmiselt: satelliidi käivitamisel alustatakse suhtlust anduritega, oodatakse teatud aeg (testimisel 1 minut), võttes samal ajal vastu GPS andmeid, loetakse andmed teistelt sensoritelt, salvestatakse saadud andmed SD-kaardile ja saadetakse andmed üle raadioside maajaama. Peale andmete edastamist alustatakse uuesti GPS andmete lugemist ja ootamist uueks tsükliks (vt Lisa 1).



## 2.2 Mõõtmised

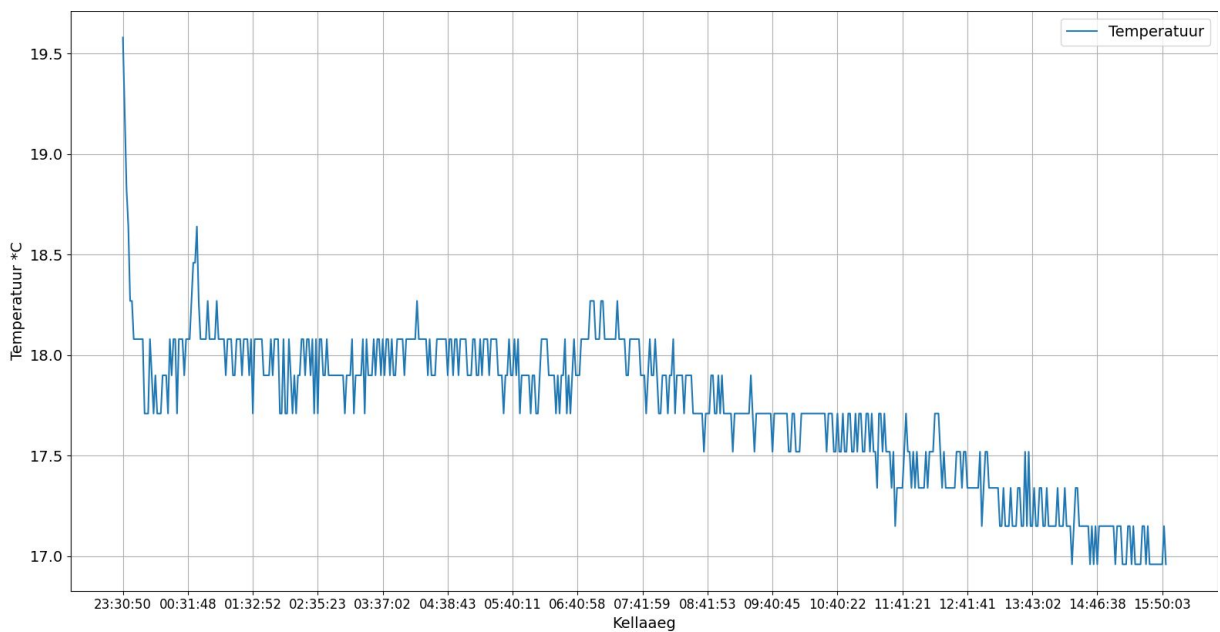
Tõestamaks, et CanSat on võimeline pikemate perioodide jooksul andmeid mõõtma, toimus 16 tunnine mõõteprotsess, mille käigus jälgis satelliit temperatuuri (vt joonis 13) ja õhurõhku (vt joonis 14). Vertikaalsed jooned graafikul tähistavad tunde. Järsk temperatuuri langus on tingitud satelliidi viimisest ühest toast teise. Töös oli ka GPS, mille kaudu sai CanSat täpse kellaaja ja kuupäeva. Tavalise *delay* ehk ootamise asemel on tarkvaras defineeritud funktsioon, kus soovitud perioodi vältel (ootamise aeg) hoitakse ühendust GPS satelliitidega ja võetakse vastu andmeid, mida need saadavad. Teostades mõõtmisi on olemas kõige värskemad GPS-andmed. Graafikutelt on eemaldatud osad kellaajad loetavuse huvides. Uurimistöö valmimise jooksul on satelliit kokku teostanud üle 30 tunni mõõtmisi. Mõõtmised on teostatud tubastes tingimustes, seega on graafikutel näha toatemperatuuri muutuseid. On märgata mõõtmistulemuste minimaalset kõikumist, mis võib olla tingitud mõõtetulemuste ümardamisest, kuid siiski on saavutatud nõutud mõõtmistäpsus  $\pm 1\text{ }^{\circ}\text{C}$ .

Satelliit saadab mõõtmiste ajal andmeid ka üle raadioside, et oleks võimalik lihtsalt kontrollida satelliidi tööd (vt joonis 15 ja 16). Raadioside signaali vastu võtmiseks kasutati programme SDR#<sup>12</sup> ja Fldigi<sup>13</sup> koos Lifeview LV5TDeluxe tuuneriga.

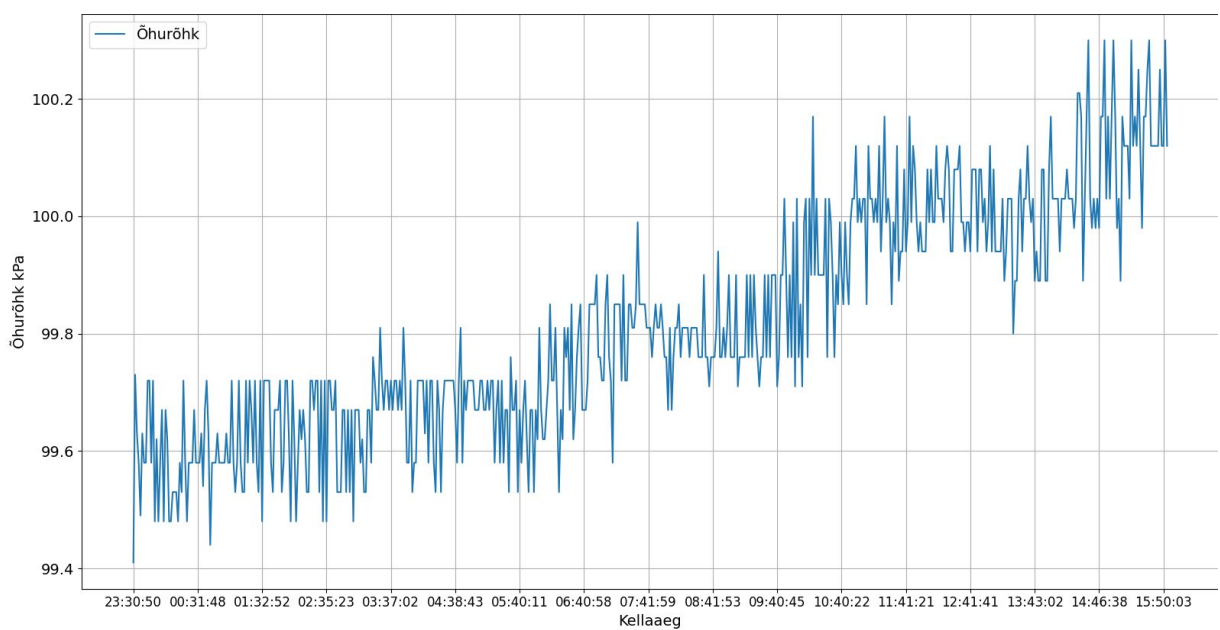
---

<sup>12</sup> Airspy: SDR#. <https://airspy.com/download/> (16.01.2021)

<sup>13</sup> W1HKJ: Fldigi. <http://www.w1hkj.com/> (16.01.2021)



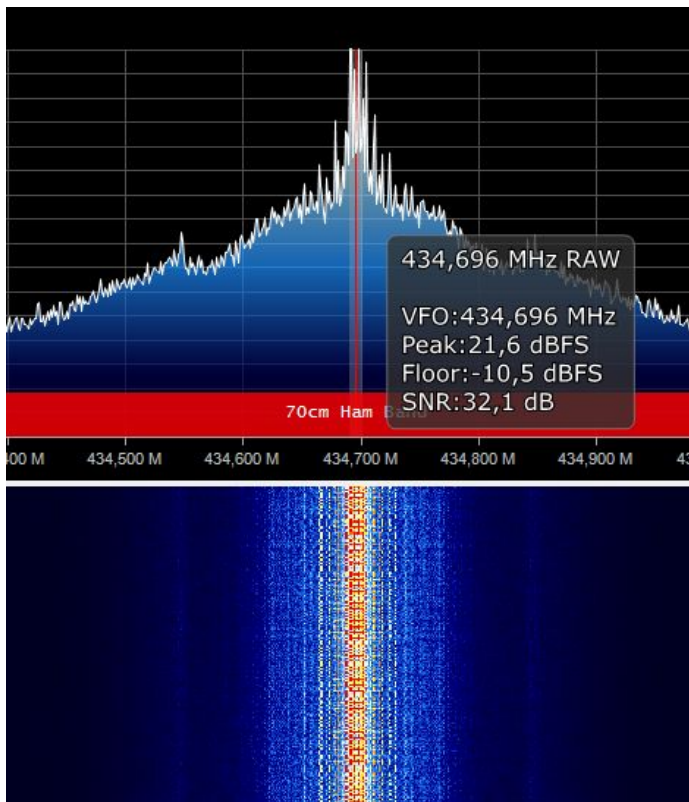
Joonis 13. Temperatuur kontrollmõõtmistel



Joonis 14. Õhurõhk kontrollmõõtmistel

ES1	15.1.2021	20:53:18	N	58.485	E	25.419	7.6m	1.2kmph	19.58	*C	100.9kPa	*4ECF
ES1	15.1.2021	20:53:41	N	58.485	E	25.419	69.7m	0.7kmph	19.58	*C	100.8kPa	*B63C
ES1	15.1.2021	20:54:4	N	58.485	E	25.419	69.7m	0.4kmph	19.77	*C	100.8kPa	*10E3

Joonis 15. Andmete edastamine üle raadioside



Joonis 16. Raadioside signaal

Katsetustest ilma lisamäluta kaameraga on säilinud ka mõned pildid, millel on näha kaamera pilte erinevate sensori lugemise kiiruste ehk säriaegadega (vt joonis 17 ja 18)



Joonis 17. Pilt kaamera testimisest



Joonis 18. Vigase ajastusega pildi lugemine

### 3. TULEMUSED

CanSat-i arendusprotsessi käigus tuli ette mitmeid probleeme sensorite ja teiste elektroonikakomponentidega. Probleemide lahendamine on nõudnud dokumentatsioonide lugemist ja lihtsate, kuid oluliste lahenduste leidmist. Juba arendustöö esimestes etappides tuli põhjalikult süveneda andurite spetsifikatsioonidesse. Esialgu sai ostetud õhurõhuandur, mis ei olnud võimeline mõõtma õhurõhku 30 km kõrgusel. CanSat-i kõrguse määramisel tuleb seega tugineda ka GPS näitudele, mis ei pruugi olla nii täpsed, kui õhurõhu järgi määratud tulemused. Ostetud sai ka ilma mäluta kaamera. Arduino madal taktsagedus ei võimaldanud pilti nõutud lahtusel piisavalt kiiresti lugeda. Mäluga kaamera puhul aga ei olnud saadaval piisavalt dokumentatsiooni, et kaamerat korrektselt konfigureerida.

Trükkplaadi katsetamise käigus põles läbi üks USB toite kaitsediod. Kuna CanSat töötab välise toiteallikaga mille vale polaarsusega ühendamise risk on madal, sai hävinenud diod asendatud juhtmega.

CanSat-i arendamine ja katsetamine on täiendanud autori teadmisi elektriskeemide ja trükkplaatide disainist. Kätt sai proovitud ka imepisikeste takistite ja kondensaatorite jootmises. Just need kogemused võivad tulevikus kasuks tulla.

CanSat-i testimised on näidanud, et satelliit on töövõimeline. Küll aga tuleks lennuplaanide korral satelliidi struktuuri kallal veel tööd teha. Täitmata on kaamera nõuded, mis on suureks osaks missioonil. Täiustada tuleks ka satelliidi ümbrise ilmastikukindlust, lisada tuleks ka kinnitus ilmastikuõhupallile ning langevarjule.

Olenemata probleemidest ja kõrvalepõigetest on satelliidi arendus olnud edukas. Valminud on trükkplaat, millele on joodetud kõik elektroonikakomponendid, struktuur ja tarkvara, kõik kasutades tavainimesele kättesaadavaid vahendeid.

## 4. JÄRELDUSED

Nagu mõõtmised on tõestanud, on võimalik koduste vahenditega arendada CanSat, mis teostab ilmastikuõhupalli põhifunktsioone, milleks on temperatuuri, õhurõhu, pöörlemiskiiruse ümber telgede ja GPS koordinaatide mõõtmine, jäädvustamine ning edastamine üle raadioside. Satelliidi arenduse võib kokku võtta järgmiste etappidena:

- 1) Eesmärkide püstitamine ja nende täitmiseks nõuete seadmine
- 2) Elektroonika ja struktuuri planeerimine, detailide valik ja soetamine
- 3) Elektrisüsteemi ja struktuuri disainimine ja valmistamine
- 4) Tarkvara arendamine
- 5) Valminud platvormi katsetamine

Tuleb meeles pidada, et selline arendustöö on väga aeganõudev ja keeruline tegevus, mille käiku on raske täpselt ette näha. Tuleb arvestada, et töö käigus tuleb ette mitmeid probleeme, millele lahenduste leidmine ei ole alati lihtne.

Töö käigus valminud CanSat ei vasta praeguseks kõigile püstitatud nõuetele, kuid aja puuduse tõttu on tulnud tähelepanu pöörata tähtsamatele satelliidi aspektidele (tarkvara töökindlus). Probleemid (kaamera) on seega jäänud tahaplaanile. Täitmata nõudeid (näiteks juhuslike vibratsioonide vastupidavus) saaks testida Tartu Observatooriumi laborites.

Kindlasti on palju võimalusi valminud platvormi ja struktuuri edasiseks täiendamiseks. CanSat-i tarkvaras on mitmeid võimalusi edasiseks optimeerimiseks juhul, kui oleks vajadus näiteks satelliidile lisada mõni andur. CanSat-i ümbrise saaks asendada näiteks 3D prinditud ümbrisega, mis oleks visuaalselt meeldivam, kuid kalduks kõrvale CanSat-i kui plekkpurgi sisse mahtuva satelliidi ideest. Erinevate edasiarenduste puhul tuleks taaskord läbi viia mitmeid teste, et veenduda satelliidi platvormi töökindluses.

## KOKKUVÕTE

Uurimistöö käigus valmis töötav CanSat ja analüüsi, kas ilmastikuõhupalli elektroonika põhifunktsionaalsus mahub standardsesse CanSat-i vormi. Töö käigus selgus, et püstitatud eesmärgid on teostatavad üsnagi lihtsate ja kodus käepärast olevate vahenditega. Uurimistöö vältel sai analüüsitud ka CanSat-i standardist ning ilmastikuõhupallide põhifunktsionaalsusest tulenevaid nõudeid ja riske, sai võrreldud elektroonikakomponente, neist sobivad välja valitud ning satelliidi disain tehtud. Satelliidi elektroonika sai kokku joodetud, sellele tarkvara kirjutatud ning seda ka korduvalt testitud.

Autor sai täpsemalt teada, kuidas toimub pisikese satelliidi arendustöö, kuidas tööd planeerida ning ellu viia. Selleks, et seda kõike teostada, oli vaja nii loogilist mõtlemist, põhjalikku uurimist, kui ka loovust ja järjepidevust.

Töö käigus paranes autori ettekujutus tööst kosmosetehnoloogia valdkonnas, saadud on ka kogemusi elektroonika ja struktuuri disainis.

Uurimistöö autor tänab töö juhendajat ja kõiki abilisi.

## **SUMMARY**

In the course of this research a working CanSat was built. It was also analyzed whether the basic functionality of a weather balloon can fit in the standard form of a CanSat. Reaching the goals was found to be easy and doable with tools found at home. As a part of this, research risks and requirements derived from the CanSat and weather balloon standards were analyzed, different electronic components were compared and chosen and a design for the satellite was made. All the electronics were soldered, a software was written and tested multiple times.

The research author got to know better how the development of a small satellite is done, how to plan the work and how to put it into practice. Logical thinking, in depth research but also creativity and consistency were needed to get all of that done.

During the research author's idea of work in the fields of space technology, electrical and structural design abilities improved.

The author thanks his instructor and everyone, who helped.



## KASUTATUD ALLIKAD

Airspy: SDR#. <https://airspy.com/download/> (16.01.2021)

Autodesk: EAGLE. <https://www.autodesk.com/products/eagle/overview> (16.01.2021)

Autodesk: Fusion 360. <https://www.autodesk.com/campaigns/education/student-design> (16.01.2021)

Bombasaro, Emanuel 2015. TITAN 1. Design and Mission Documentation.  
[https://bombasaro.org/files/Titan1\\_Design\\_Mission\\_Doc.pdf](https://bombasaro.org/files/Titan1_Design_Mission_Doc.pdf) (16.01.2021)

Dabberdt, W, F, Shellhorn, R, Cole, H, Paukkunen, A, Hörhammer, J, Antikainen, V, 2003.  
Radiosondes. <http://www.radiopassioni.it/pdf/materialirsonde/Radiosondes.pdf> (16.01.2021)

Dcube Store Blog: Difference between I2C and SPI (I2C VS SPI).  
<https://dcubestore.com/blog/difference-between-i2c-and-spi/> (16.01.2021)

Electronics Notes: Amplitude Modulation, AM.  
<https://www.electronics-notes.com/articles/radio/modulation/amplitude-modulation-am.php> (16.01.2021)

Embedded Staff 2002. Introduction to Serial Peripheral Interface.  
<https://www.embedded.com/introduction-to-serial-peripheral-interface/> (16.01.2021)

ESA - The European Space Agency: CanSat 2020-2021 Guidelines.  
[https://cansat.esa.int/wp-content/uploads/2020/09/CanSat\\_guidelines\\_2020\\_2021.pdf](https://cansat.esa.int/wp-content/uploads/2020/09/CanSat_guidelines_2020_2021.pdf) (16.01.2021)

I2C Bus: I2C – What’s That? <https://www.i2c-bus.org/> (16.01.2021)

Walker, R, Galeone, P, Page, H, Castro, A, Emma, F 2010. ESA Hands-on Space Education Project Activities for University Students: Attracting and Training the Next Generation of Space Engineers.  
<http://www.ieec.uned.es/Investigacion/Educon2010/SearchTool/EDUCON2010/papers/2010S10B01.pdf> (16.01.2021)

W1HKJ: Fldigi. <http://www.w1hkj.com/> (16.01.2021)

# LISAD

## Lisa 1. CanSat-i aluskood

```
//
// ES1 CanSat-i tarkvara
// Ergo Adams
// 2020
//

#define RADIOPIN A1

#include <string.h>
#include <util/crc16.h>
#include <Adafruit_MPL115A2.h>
#include <TinyGPS++.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <SdFat.h>

// Temperatuuri ja õhurõhu sensor ja muutujad temperatuuri ja rõhu jaoks
Adafruit_MPL115A2 mpl115a2;
float temperature = 0;
float pressure = 0;

// GPS
TinyGPSPlus gps;
SoftwareSerial ss(0, 1);
float gpslatitude, gpslongitude;
float gpsaltitude;
float gpsspeed;
int gpssday, gpsmonth, gpsyear;
int gpshour, gpsminute, gpssecond;

// SD
const int PROGMEM CS_Pin = 10;
SdFat sd;
```

```
SdFile logFile;
```

```
// Gyro
```

```
const int PROGMEM MPU=0x68;
```

```
int16_t AcX,AcY,AcZ,GyX,GyY,GyZ;
```

```
double pitch,roll;
```

```
// Raadioside, millegipärast kood ei toimi, kui muutujad tõsta funktsiooni endasse. Pole aimugi, miks nii on...
```

```
const char *radiomsg = "ES1 %d.%d.%d %d:%d:%d N %s E %s %sm %skmph %s*C %skPa";
```

```
char chrTemperature[15];
```

```
char chrPressure[15];
```

```
char chrLatitude[10];
```

```
char chrLongitude[10];
```

```
char chrAltitude[10];
```

```
char chrSpeed[10];
```

```
char chrPitch[10];
```

```
char chrRoll[10];
```

```
char charBuffer[80];
```

```
void rtty_txstring (char * string)
```

```
{
```

```
    char c;
```

```
    c = *string++;
```

```
    while ( c != '\0')
```

```
    {
```

```
        rtty_txbyte (c);
```

```
        c = *string++;
```

```
    }
```

```
}
```

```
void rtty_txbyte (char c)
```

```
{
```

```
    int i;
```

```
    rtty_txbit (0); // Start bitt
```

```
    for (i=0;i<7;i++) // 7/8 vastavalt sellele, kas kasutatakse ASCII-7 / ASCII-8
```

```
    {
```

```
        if (c & 1) rtty_txbit(1);
```

```

        else rtty_txbit(0);
        c = c >> 1;
    }

    rtty_txbit (1); // Stopp bitt
    rtty_txbit (1); // Stopp bitt
}

void rtty_txbit (int bit)
{
    if (bit)
    {
        digitalWrite(RADIOPIN, HIGH);
    }
    else
    {
        digitalWrite(RADIOPIN, LOW);
    }

    //delayMicroseconds(3370); // 300 baud rate
    delayMicroseconds(10000); // 50 baud rate
    delayMicroseconds(10150);
}

uint16_t gps_CRC16_checksum (char *string) {
    size_t i;
    uint16_t crc;
    uint8_t c;

    crc = 0xFFFF;

    // Kontrollsumma arvutamine, esimesed kaks $s ei loe
    for (i = 2; i < strlen(string); i++) {
        c = string[i];
        crc = _crc_xmodem_update (crc, c);
    }

    return crc;
}

void sendRTTY(char *datastring) {

```

```

unsigned int CHECKSUM = gps_CRC16_checksum(datastring); // Kontrollsumma
char checksum_str[6];
sprintf(checksum_str, "%04X\n", CHECKSUM);
strcat(datastring,checksum_str);

rtty_txstring(datastring); //24852, 1558
}

// Kuidas RTTY funktsioonid üldse töötavad? Pisikene kokkuvõte (loodetavasti)
// 1. CRC16 checksum, liidetakse datastringi lõppu -> kuidas checksum tehakse?, kas saab
efektiivsemalt datastringi juurde panna
// 2. kuni pole jõudnud null täheni (andmete lõpp), võtame tähe, saadame start biti, iga biti
tähes (alustades lsb-st) ja 2 stopp bitti
// saatmine on vastavalt bitile, kui on 1 siis radiopin high, muidu low

static void smartDelay(unsigned long ms) {
    ss.begin(9600);
    unsigned long start = millis();
    do {
        while (ss.available())
            gps.encode(ss.read());
    } while (millis() - start < ms);

    ss.end();
}

void getAngle(int Vx,int Vy,int Vz) {
    double x = Vx;
    double y = Vy;
    double z = Vz;

    pitch = atan(x/sqrt((y*y) + (z*z)));
    roll = atan(y/sqrt((x*x) + (z*z)));
    //convert radians into degrees
    pitch = pitch * (180.0/3.14);
    roll = roll * (180.0/3.14);
}

void readGyro() {
    // Loeme kiirendussensorit
    AcX = Wire.read()<<8|Wire.read();
    AcY = Wire.read()<<8|Wire.read();
}

```

```

AcZ = Wire.read() << 8 | Wire.read();

// Loeme güroskoop
GyX = Wire.read() << 8 | Wire.read();
GyY = Wire.read() << 8 | Wire.read();
GyZ = Wire.read() << 8 | Wire.read();
getAngle(AcX, AcY, AcZ);
}

void readPT() {
  mpl115a2.getPT(&pressure, &temperature);
}

void readGPS() {
  gpslatitude = gps.location.lat();
  gpslongitude = gps.location.lng();
  gpsspeed = gps.speed.kmph();
  gpsaltitude = gps.altitude.meters();
  TinyGPSDate gpsdate = gps.date;
  TinyGPSTime gpstime = gps.time;
  gpsday = gpsdate.day();
  gpsmonth = gpsdate.month();
  gpsyear = gpsdate.year();
  gpshour = gpstime.hour();
  gpsminute = gpstime.minute();
  gpssecond = gpstime.second();

  if (gpshour + 2 >= 24) {
    gpshour = (gpshour + 2) % 24;
    gpsday = gpsday + 1;
  } else {
    gpshour = (gpshour + 2);
  }
}

void writeData() {
  // Avame logifaili, kui ei õnnestu avada, siis error
  if (!LogFile.open("log.txt", O_RDWR | O_CREAT | O_AT_END)) {
    sd.errorHalt("opening log.txt for write failed");
  }

  // Kirjutame andmed kaardile

```

```

// See on väga kole viis, kuidas andmeid kirjutada, aga kuidagi on vaja mälu säästa :)
logFile.print(gpsday);
logFile.print(F("."));
logFile.print(gpsmonth);
logFile.print(F("."));
logFile.print(gpsyear);
logFile.print(F(" "));

logFile.print(gpshour);
logFile.print(F(":"));
logFile.print(gpsminute);
logFile.print(F(":"));
logFile.print(gpssecond);
logFile.print(F(" "));

logFile.print(F("N "));
logFile.print(gpslatitude);
logFile.print(F(" E "));
logFile.print(gpslongitude);
logFile.print(F(" "));

logFile.print(gpsaltitude);
logFile.print(F("m "));
logFile.print(gpsspeed);
logFile.print(F("kmph "));

logFile.print(temperature);
logFile.print(F("*C "));
logFile.print(pressure);
logFile.print(F("kPa "));

logFile.print(pitch);
logFile.print(F("* pitch "));
logFile.print(roll);
logFile.println(F("* roll "));

// Paneme logifaili kinni
logFile.close();
}

void sendData() {

```

```

dtostrf(temperature, 5, 2, chrTemperature);
dtostrf(pressure, 5, 1, chrPressure);
dtostrf(gpslatitude, 7, 3, chrLatitude);
dtostrf(gpslongitude, 7, 3, chrLongitude);
dtostrf(gpsaltitude, 5, 1, chrAltitude);
dtostrf(gpsspeed, 5, 1, chrSpeed);

    snprintf(charBuffer, 80, radiomsg, gpsday, gpsmonth, gpsyear, gpshour, gpsminute,
gpssecond, chrLatitude, chrLongitude, chrAltitude, chrSpeed, chrTemperature, chrPressure);
    sendRTTY(charBuffer);

}

void setup() {
    // Siin oli bootup protsess, mis kontrollis, kas sensor läks tööle, aga see võtab liiga palju
    mälu, et alles jätta.
    // Tuleb lihtsalt loota, et hakkab õigesti tööle

    pinMode(RADIOPIN,OUTPUT);
    Serial.begin(9600);

    // SD kaart
    if (!sd.begin(CS_Pin, SPI_HALF_SPEED)) sd.initErrorHalt();

    // MPL115A2 (temp ja õhurõhk)
    mpl115a2.begin();
    delay(1000);

    // GPS init, ärme siin veel alusta, lülitame ainult andmete lugemise ajaks sisse, muidu segab
    teisi funktsioone
    //ss.begin(9600);

    // Gyro init
    Wire.begin();
    Wire.beginTransmission(MPU);
    Wire.write(0x6B);
    Wire.write(0);
    Wire.endTransmission(true);

    Serial.println("ES1 started!");
}

```



```
void loop() {  
  smartDelay(60000);  
  readGyro();  
  readPT();  
  readGPS();  
  writeData();  
  sendData();  
}
```