

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Svetlana Ušakova 164057IAPB

**TTÜ100/101 SATELLIITIDE
RAADIOSIGNAALIDE
MODULEERIMINE/DEMOMULEERIMINE**

Bakalaureusetöö

Juhendaja: Evelin Halling
MSc

Tallinn 2019

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: Svetlana Ušakova

21.05.2019

Annotatsioon

Bakalaureusetöös on kaks eesmärki. Esimene eesmärk on valmistada vastuvõtja tarkvaralise raadio põhjal, mis oleks võimeline demoduleerima vastuvõetud signaali ning seda dekodeerima. Teine eesmärk on realiseerida pakettide saatmine ja vastuvõtmine Dire Wolf tarkvara abil. Vastuvõetud andmed peab salvestama sobivas formaadis.

Töö koosneb kolmest osast. Esimeses osas antakse ülevaade demodulatsiooni viisist ning dekodeerimise etappidest. Samuti tutvustatakse kasutatavat kaadri struktuuri. Teises osas valmistatakse signaali vastuvõtmiseks ja töötlemiseks skeem Gnu Radio tarkvaras ning seletatakse detailsemalt iga skeemi komponenti ja selle parameetreid. Kolmandas osas tutvustatakse Dire Wolf tarkvara. Pakettide vastuvõtmiseks muudetakse konfiguratsiooni fail vastavalt vajadusele. Vastuvõetud sõnumite salvestamiseks kirjutatakse Python keeles programm. Pakettide saatmiseks täiendatakse olemasoleva tarkvara koodi, et pakettide struktuur oleks sobilik satelliitide raadio jaoks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 4 peatükki, 24 joonist, kolme tabelit.

Abstract

Modulation/Demodulation of Radio Signals from Satellites

TTÜ100/101

The thesis has two main goals. The first goal is to create a receiver based on software defined raadio, which is able to demodulate received radiosignal and then decode it. The second goal is to implement packet sending and receiving using Dire Wolf software. Received data must be stored in a suitable format.

The thesis consists of three parts. The first part provides an overview of the demodulation mode and the decoding steps. Also in this part the structure of the frame is introduced. In the second part the receiver using software defined radio is created and implemented in Gnu Radio software. All scheme blocks with parameters are described and explained. The third part introduces Dire Wolf software. For receiving signals the configuration file is changed as needed. To save the received messages a code written in Python language is created. For sending packets the existing software code is supplemented so that the packet structure is suitable for the satellite radio.

The thesis is in Estonian language and contains 30 pages of text, 4 chapters, 24 figures, 3 tables.

Lühendite ja mõistete sõnastik

FSK	<i>Frequency Shift Keying</i> , digitaalne sagedusmodulatsioon
VHF	<i>Very High Frequency</i> , ülikõrgsagedus
UHF	<i>Ultra High Frequency</i> , ultrakõrgsagedus
USB	<i>Universal Serial Bus</i> , universaalne järjestiksiin
SDR	<i>Software Defined Radio</i> , tarkvaraline raadio
AX.25	<i>Data Link Layer Protocol</i> , raadioside protokoll
ASCII	<i>American Standard Code for Information Interchange</i> , Ameerika Informatsioonivahetuse Standardkood
HEX	<i>Hexadecimal</i> , kuueteistkümnendsüsteem
FCS	<i>Frame Check Sequence</i> , kontrolljada
CRC	<i>Cyclic redundancy check</i> , kontrollsumma
PID	<i>Proportional Integraal Derivative Control</i> , Proportsionaal Integraal Diferentsiaalregulaator
NRZI	<i>Non Return to Zero Inverted</i>
FSK	<i>Frequency Shift Keying</i> , digitaalne sagedusmodulatsioon
VHF	<i>Very High Frequency</i> , ülikõrgsagedus
UHF	<i>Ultra High Frequency</i> , ultrakõrgsagedus
USB	<i>Universal Serial Bus</i> , universaalne järjestiksiin
SDR	<i>Software Defined Radio</i> , tarkvaraline raadio
AX.25	<i>Data Link Layer Protocol</i> , raadioside protokoll

Sisukord

1 Sissejuhatus	10
2 Signaali demoduleerimisviisi ülevaade	11
2.1 FSK modulatsioon.....	11
2.2 Skrambleerimine	12
2.3 NRZI kodeerimine ja dekodeerimine	14
2.4 Bititaitmine	15
2.5 AX.25 kaadri struktuur.....	15
2.5.1 Lipp.....	16
2.5.2 Aadress.....	16
2.5.3 Kontroll	17
2.5.4 PID.....	17
2.5.5 Informatsioon	17
2.5.6 FCS	17
3 Signaalide vastuvõtmine tarkvaralise raadio abil	18
3.1 Tarkvaraline raadio	18
3.2 Vastuvõtja DXPatrol	18
2.3 Si446x transiiver	19
2.3 Gnu Radio	20
3.3 Vastuvõtja disain GNU Radio abil.....	20
3.3.1 Signaali salvestamine.....	20
3.3.2 Signaali demoduleerimine	23
4 Pakettide saatmine ja vastuvõtmine Dire Wolf tarkvara abil	29
4.1 Kenwood raadio	29
4.2 Dire Wolf, TNC, KISS	29
4.3 Pakettide vastuvõtmine.....	30
4.3.1 Konfiguratsioon.....	30
4.3.2 Vastuvõetud sõnumite salvestamine.....	32
4.4 Pakettide saatmine.....	34
5 Tulemuste valideerimine	37

6 Kokkuvõte	38
Kasutatud kirjandus	40

Jooniste loetelu

Joonis 1. FSK signaali moodustamise printsiip.....	12
Joonis 2. Ühe biti edastuskiirus [3].....	12
Joonis 3. Skrambleerimine polünoomiga $1 + X^{12} + X^{17}$	13
Joonis 4. Deskrambleerimine polünoomiga $1 + X^{12} + X^{17}$	14
Joonis 5. NRZI kodeeringud.....	14
Joonis 6. Bititäitmise näide [9].....	15
Joonis 7. DXPatrol SDR vastuvõtja koos antenniga.....	19
Joonis 8. Si446x transiiver.....	19
Joonis 9. Signaali spekter SDR Console tarkvaras.....	21
Joonis 10. Saatja ja vastuvõtja seadistus.....	22
Joonis 11. Skeem signaali salvestamiseks.....	22
Joonis 12. Signaali spektrogramm.....	23
Joonis 13. Signaali spekter.....	23
Joonis 14. Signaali demoduleerimine.....	24
Joonis 15. Signaali spektrogramm.....	25
Joonis 16. Nihutatud signaali spektrogramm.....	26
Joonis 17. Quadrature demode ploki väljundsignaal.....	26
Joonis 18. Demoduleeritud sõnum.....	28
Joonis 19. Kenwood, antenni, audiokaardi ja arvuti seadistus.....	30
Joonis 20. Dire Wolfi avavaade.....	32
Joonis 21. Dekodeeritud sõnum.....	32
Joonis 22. Kissutil väljund.....	33
Joonis 23. Lisapäis ja lisaipud koos kaadriga.....	35
Joonis 24. Si644x sisendsignaal.....	35

Tabelite loetelu

Tabel 1. "UI" kaadri struktuur.	16
Tabel 2. Aadressi väli.....	16
Tabel 3. Si446x raadioga vastuvõetud sünkrosõna.....	35

1 Sissejuhatus

Satelliitide arendamisega tegelevad juba üsna paljud ülikoolid; ka Tallinna Tehnikaülikoolis on alates 2014. aastast satelliitide arendamisele suunatud tudengisatelliidi programm. Programmi eesmärk on ehitada võimekad satelliidid EV100/101, mida plaanitakse lennutada orbiidile juba 2019. aasta suvel.

Satelliiti on vaja opereerida ning jälgida Maalt, selleks otstarbeks peab maajaamas olema raadio. Satelliit hakkab saatma Maale mitmesugust informatsiooni oma alamsüsteemide kohta ning raadio peab suutma saadetud sõnumeid vastu võtta ning saada moduleeritud ja kodeeritud signaalist õige sõnumi. Samuti selleks, et öelda satelliidile, milline on järgmine missioon, peab raadio olema võimeline moduleerima ja kodeerima signaali nii, et saadetud sõnum oleks õige struktuuriga ning satelliit suudaks seda vastu võtta.

EV100/101 satelliitides kasutatakse FSK modulatsiooni viisi signaalide moduleerimiseks ning infoedastuskiirus on 9600 baud. Sõnumeid saadetakse sagedusel 435,45 MHz ehk raadio peab töötama UHF lainealas. Raadios kasutatakse NRZI kodeeringut ning isesünkroniseeruvat skramblerit. Satelliidi raadioside protokoll on AX.25 [1].

Lõputöö eesmärk on valmistada raadio, mis suudaks tarkvara abil satelliidilt saadetud signaale demoduleerida ning saadetavat infot moduleerida. Valminud tarkvara peab võimaldama vastuvõetud andmeid analüüsida ning salvestada.

Lõputöö koosneb teoreetilisest ja praktilisest osast. Teoreetilisest osast tutvustatakse raadios kasutatavat modulatsiooni ning kodeeringut. Lisaks kirjeldatakse AX.25 kaadri struktuuri. Praktiline osa on jagatud kaheks. Esimeses osas valmistatakse vastuvõtja tarkvaralise raadio baasil, teises osas realiseeritakse pakettide saatmist ja vastuvõtmist Dire Wolf tarkvara abil. Pakettide saatmiseks muudetakse Dire Wolf koodi osa nii, et välja saadetud pakettide ette lisatakse satelliidile pakettide vastuvõtmiseks vajalikud bitid.

2 Signaali demoduleerimisviisi ülevaade

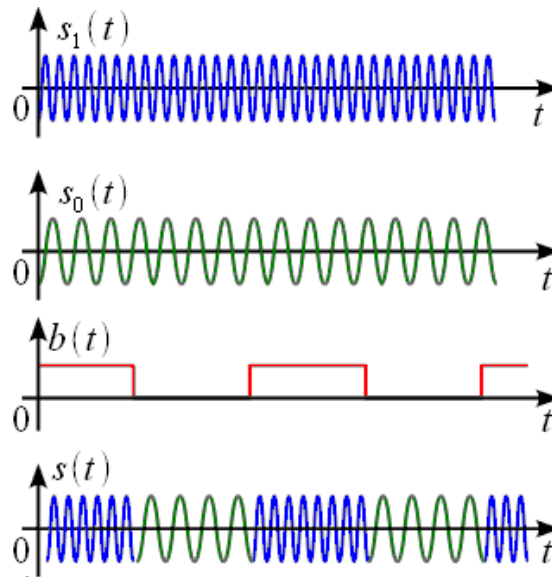
Sõnumeid saadetakse sagedusel 435,45 MHz ning side kiirus on 9600 baud. Satelliidi sideprotokolliks on AX 25. Raadios kasutatakse NRZI kodeeringut ning isesünkroniseeruvat skramblerit. Saatmiseks ja vastuvõetud andmete moduleerimiseks/demoduleerimiseks kasutatakse binaarset sagedusmodulatsiooni FSK [1].

Järgnevas peatükis tutvustatakse FSK modulatsiooni viisi ja sellega seotud mõisteid. Kirjeldatakse binaarandmete töötlemise etappe nagu dekodeerimine, deskrambleerimine ja bititaitmine ning seletatakse miks neid vaja on. Samuti seletatakse AX.25 kaadri struktuuri ning kirjeldatakse kaadri iga välja eraldi.

2.1 FSK modulatsioon

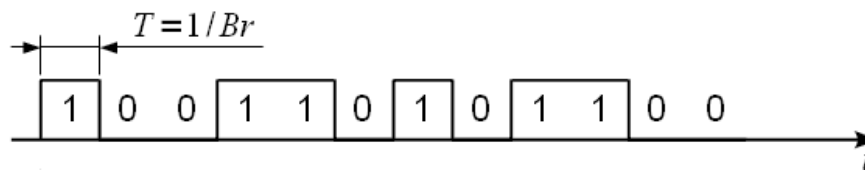
Modulatsioon tähendab signaali teisendamist või ülekandmist madalamate sageduste piirkonnast kõrgemate sageduste piirkonda selleks, et signaali oleks võimalik raadio teel edastada. Demodulatsioon on vastupidine protsess, mille käigus signaal kantakse üle kõrgematest sagedustest madalamatesse [2].

Üks lihtsamatest ja levinumatest digitaalse modulatsiooni liikidest on binaarne sagedusmodulatsioon konstantse amplituudiga. FSK modulatsiooni korral muutub kandesignaali sagedus sõltuvalt digitaalsignaali väärtusest. Digitaalsignaali $b(t)$ on ühtedest ja nullidest koosnev bitijada. Moduleeritud signaal moodustub kahe sinusoidi segmentidest $s_0(t)$ ja $s_1(t)$. Kui digitaalsignaali $b(t)$ hetkväärtus vastab loogilisele nullile, kasutatakse edastamiseks $s_0(t)$ signaali ning loogilise „1“ korral kasutatakse $s_1(t)$ signaali. $s_1(t)$ sagedus peab olema suurem kui $s_0(t)$.



Joonis 1. FSK signaali moodustamise printsiip.

Digitaalsignaali $b(t)$ edastatakse kiirusega Br baud ning bitijada ühe biti kiirus arvutatakse valemiga $T = 1/Br$ [3].



Joonis 2. Ühe biti edastuskiirus [3].

FSK signaali on võimalik edastada nii koherentselt kui ka mittekoherentselt, seega võib demoduleerimine toimuda koherentselt ning mittekoherentselt. Koherentse FSK korral on kõik vastuvõetava signaali parameetrid teada. Mittekoherentse FSK korral muutub faas tavaliselt juhuslikult ning vastuvõtul on faas teadmata.

Kõige sagedamini kasutatakse just mittekoherentset FSK modulatsiooni, kuna selle realiseerimine on lihtsam, kuigi vea tõenäosuse näitajad on koherentsel FSK-l paremad [2].

2.2 Skrambleerimine

Skrambleerimine on protsess, mille käigus digitaalsignaali muudetakse pseudojuhuslikuks digitaalsignaalliks. Telekommunikatsioonisüsteemides kasutatakse skrambleerimist andmete randomeerimiseks. See aitab parandada sünkroniseerimist, kuna välistab

olukorra, kui andmevoos tekivad pikad ühtedest või nullidest koosnevad jaded. Samuti kuna andmevoog on nüüd juhuslik, jaotub selle spektraalne energia kogu aeg ühtlaselt [4].

Skrambleerimine toimub skrambleri abil enne, kui signaal jõuab modulaatorisse. Skrambler on telekommunikatsiooniseade, mis rakendab loogilist liitmist mooduliga 2 lähtesignaali ja pseudojuhusliku digitaalsignaali vahel [5].

On olemas kahte tüüpi skramblereid: isesünkroniseeruvad (multiplikatiivsed) skramblerid ning skramblerid, mis nõuavad eraldi seadistust sünkroniseerimiseks [6]. TTÜ100/101 satelliidid kasutavad isesünkroniseeruvat skramblerit, mistõttu tuleb edaspidi juttu just sellest. Selleks, et oleks võimalik originaalsõnumit taastada, on vastuvõtjas vaja teostada deskrambleerimine. Pseudojuhuslik andmete jada läheb läbi deskrambleri, mis taastab esialgsed andmed. Satelliitide skrambleri polünoom on $1 + X^{12} + X^{17}$ [5].

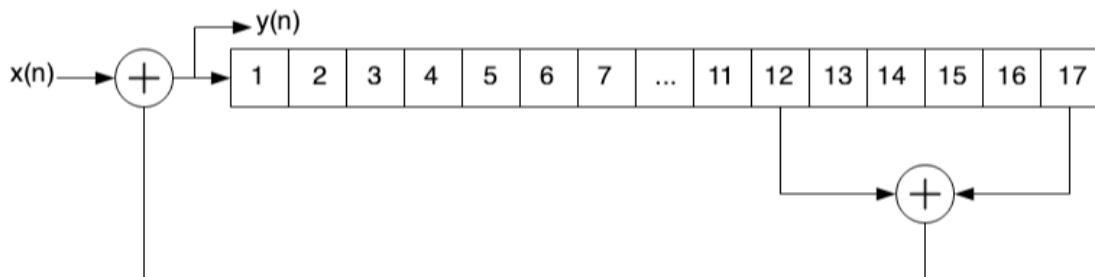
Skrambleerimisel on väljund esitatud valemiga

$$y(n) = x(n) \oplus (R_{12} \oplus R_{17})$$

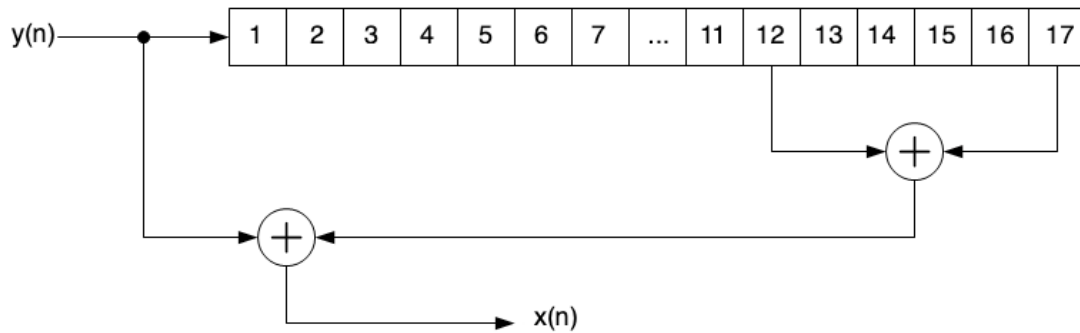
ning deskrambleerimise väljund on

$$x(n)' = y(n) \oplus x(n) \oplus (R_{12} \oplus R_{17}) \oplus (R_{12} \oplus R_{17}) = x(n) [6].$$

Joonisel 3 ja 4 on võimalik näha skrambleerimise ja deskrambleerimise protsesse, kus on kasutusel $1 + X^{12} + X^{17}$ polünoom.



Joonis 3. Skrambleerimine polünoomiga $1 + X^{12} + X^{17}$.



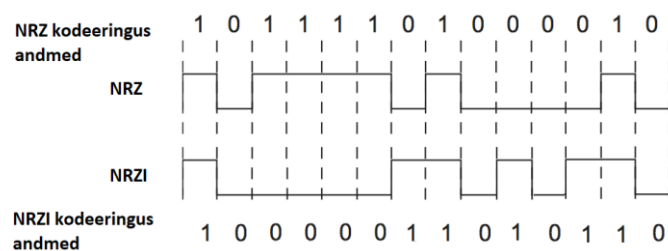
Joonis 4. Deskrambleerimine polünoomiga $1 + X^{12} + X^{17}$.

2.3 NRZI kodeerimine ja dekodeerimine

Enne skrambleerimist ja moduleerimist viiakse signaal NRZI formaati. Samuti pärast signaali demoduleerimist ning deskrambleerimist on tähtis viia signaal tagasi NRZ formaati ehk teostada NRZI dekodeerimine. Kodeerimise eesmärk on tagada edastamisel andmete terviklikkus ning kõrvaldada müra [7].

NRZL (*Non Return to Zero Level*) ja NRZI (*Non Return to Zero Invertive*) on lineaarse kodeerimise meetodid. NRZI kodeering tähendab, et signaalil on kaks olekut – positiivne ja negatiivne. Kui kodeerija võtab vastu loogilist „0“, muudab ta oma väärtust vastupidiseks, ning kui ta võtab vastu loogilist „1“, jääb väärtus samaks. Andmete edastamisel digitaalse sidekanali kaudu peavad vastuvõtja ja saatja olema sünkroniseeritud [7].

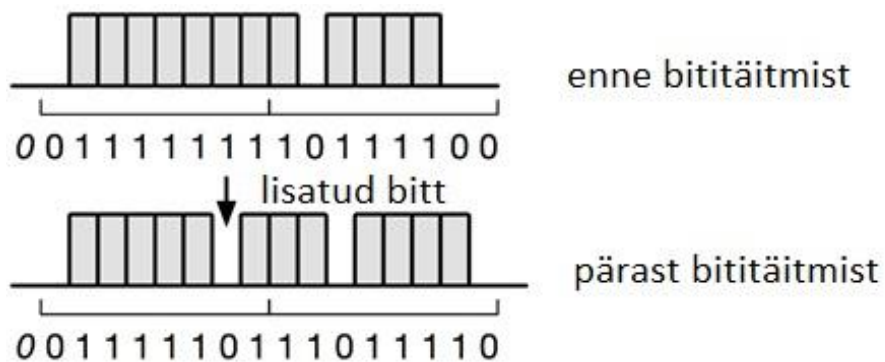
NRZL kodeeringu puhul on tegemist binaarse koodiga ning loogiline „0” on esitatud negatiivse pingega ja loogiline „1” positiivse pingega. Joonisel 5 on näha NRZ ja NRZI kodeeringut ning kuidas näeb välja bitivoog enne ja pärast NRZI kodeerimist.



Joonis 5. NRZI kodeeringud.

2.4 Bititaitmine

AX.25 kaader algab ja lõpeb lipuga. Kuna lipp määrab kaadri alguse ja lõpu, ei tohi sama bittide kombinatsioon esineda kuskil kaadri sees. Vältimaks sellist olukorda, kasutatakse bititaitmist (*bit stuffing*). Saatja jälgib, et kui kaadri sees tekib juhuslikult viis või rohkem „1“ järjest, siis pärast iga viiendat „1“ saadab saatja ühe lisabiti väärtusega „0“. Andmete saamisel peab vastuvõtja tegema pöördtegevust. Kui ta näeb, et tekib bitijada, mis koosneb viiest järjestikusest „1“, eemaldab ta „0“ biti kohe pärast viiendat „1“. Joonisel 6 on esitatud näide, kuidas näeb välja bitijada enne ja pärast bititaitmist [8].



Joonis 6. Bititaitmise näide [9].

2.5 AX.25 kaadri struktuur

AX.25 (**A**mateur **X**.25) on kanalikihi protokoll, mis põhineb HDLC (*High Level Data Link Control*) protokollil. Kanalikiht vastutab ühenduse loomise, andmete edastamise, vookontrolli ning vigade avastamise eest [10]. Saatmisel jaotatakse andmed väikesteks andmeplokkideks, mida nimetatakse kaadriteks. On olemas mitu AX.25 kaadri tüüpi, kuid TTÜ100/101 satelliitides kasutatakse ainult „UI“ (*Unnumbered Information*) struktuuriga kaadreid. Iga kaader koosneb mitmest väljast. „UI“ kaadri struktuuri on võimalik näha tabelis 1 [8].

Lipp	AX.25 kaadri päis (128 bitti)				Info	FCS	Lipp
	Vastuvõtja aadress	Saatja aadress	Kontroll	PID			
0x7E 8 bitti	56 bitti	56 bitti	0x03 8 bitti	0x0F 8 bitti	0 – 2048 bitti	16 bitti	0x7E 8 bitti

Tabel 1. "UI" kaadri struktuur.

Iga väli koosneb fikseeritud arvust oktetidest. Üks oktet on võrdne ühe baidiga, milles sisaldub 8 bitti binaarandmeid. Igal väljal on oma ülesanne.

FCS välja bitid edastatakse kõrgeim bitt enne, kõik ülejäänud väljade bitid edastatakse madalaim bitt enne [8].

2.5.1 Lipp

Lipu väli määrab kaadri alguse ja lõpu. Kahel kaadril võib olla üks lipp, esimesel kaadril määrab see lipp kaadri lõpu ning järgmisel kaadril määrab sama lipp kaadri alguse. Samuti võib kahe kaadri vahel olla mitu lippu. Lipp koosneb ühest oktetist ning on esitatud binaarkoodiga „01111110“ või 0x7E kuuekümnendsüsteemis. Kaadri lipp ei tohi esineda kaadri sees [8].

2.5.2 Aadress

Aadressi väli identifitseerib, kust konkreetne kaader saadeti ning kuhu see on saadetud. Aadressi väli koosneb 14 oktetist ehk saatja aadress on 7 oktetit pikk ning vastuvõtja aadress on samuti 7 oktetit pikk. Kõigepealt saadetakse need oktetid, mis sisaldavad sihtkoha aadressi (A1–A7). Samal ajal kui edastatakse kaadri ülejäänud osa, on vastuvõtjal aega kontrollida, kas see kaader on temale adresseeritud või mitte [8].

Aadressi väli	
Vastuvõtja aadress	Saatja aadress
A1 A2 A3 A4 A5 A6 A7	A8 A9 A10 A11 A12 A13 A14

Tabel 2. Aadressi väli.

2.5.3 Kontroll

Kontrollväli identifitseerib edastatava kaadri tüüpi. Kontrollvälja pikkus võib olla kas 1 või 2 oktetit, aga kuna TTÜ100/101 satelliitides on kasutusel „UI” kaadrid, on kontrollväli alati ühe baidi pikkune ning fikseeritud väärtusega, mis on alati võrdne 0x03 [8].

2.5.4 PID

PID väli identifitseerib, milline protokoll on kasutusel [8]. Kuna TTÜ100/101 satelliit ei kasuta ühtki protokollit, on PID välja väärtus samuti fikseeritud ning alati võrdne 0xF0.

2.5.5 Informatsioon

Informatsiooniväli edastab informatsiooni saatjast vastuvõtjani. Informatsioonivälja maksimaalne pikkus on 256 baiti ning baitide arv peab olema täisarv [8].

2.5.6 FCS

FCS (*Frame-Check Sequence*) kontrollib, kas edastamisel esines vigu või mitte. FCS on kahe baidi pikkune bitijada. FCS arvutatakse nii saatjas kui ka vastuvõtjas ning arvutatud väärtus peab langema kokku. Kui välja saadetud kaadri FCS ning vastu võetud kaadri FCS ei ole võrdsed, tähendab see, et vastu võetud kaader on vigane. FCS arvutamiseks kasutatakse CRC algoritmi. FCS polünoom on $X^{16} + X^{12} + X^5 + 1$ [11].

3 Signaalide vastuvõtmine tarkvaralise raadio abil

Lõputöö esimeses osas valmistati tarkvaralisel raadio põhinev raadiosignaalide vastuvõtja. Eesmärk oli kasutada SDR vastuvõtjat ning Gnu Radio tarkvara.

Saatmine toimub sagedusel 435,45 MHz, modulatsiooniks on FSK, edastuskiirus on 9600 baud, signaali ribalaius on 20 kHz.

3.1 Tarkvaraline raadio

Tarkvaraga määratletud raadio on raadio, milles mõned või kõik füüsilise kihi funktsioonid on defineeritud tarkvaraga. See tähendab, et signaali töötlemise peamisi etappe, nagu näiteks modulatsioon/demodulatsioon, vigade parandamine ja kodeerimine/dekodeerimine, on võimalik teostada tarkvara abil. Tarkvaraga määratletud raadio on täielikult konfigureeritav ning selle konfiguratsioone on võimalik ümber programmeerida tarkvaras [12].

3.2 Vastuvõtja DXPatrol

DXPatrol (vt joonis 7) on SDR vastuvõtja, millel on väga lai vastuvõtu sagedusala. DXPatrol võtab vastu raadiosignaale sagedusvahemikus 100 kHz kuni 2 GHz maksimaalse diskreetimissagedusega 3,2 MHz. DXPatrolit on võimalik kasutada ükskõik millise tarkvaraga, mis toetab RTL2832-põhist DVB-T kiibistikku, seega sobib Gnu Radio selleks väga hästi [13].

Selleks, et kasutada DXPatrol Windows operatsiooni süsteemiga, on soovitatav alla laadida sobiv draiver, näiteks selline, mida pakub „Zadig” installer.



Joonis 7. DXPatrol SDR vastuvõtja koos antenniga.

2.3 Si446x transiiver

Saatmiseks on kasutatud Si446x transiiverit, mis katab sagedusala vahemikus 142–1050 MHz. Si446x toetab (G)FSK, 4(G)FSK, (G)MSK ja OOK modulatsioone. Saatjal on automaatne sagedusjuhtimine ning võimendusjuhtimine [14].



Joonis 8. Si446x transiiver.

2.3 Gnu Radio

Gnu Radio on avaliku lähtekoodiga tarkvaraarenduse tööriistade komplekt, mis sisaldab laias valikus komponente signaali töötlemiseks, edastamiseks ja vastuvõtmiseks. See sobib kasutamiseks nii odavate ja kergesti kättesaadavate kui ka professionaalsete tööriistadega.

Iga komponent kujutab endast nn plokki, mis võimaldab teha mitte ainult põhilisi matemaatilisi operatsioone signaaliga, vaid ka keerulisemaid tegevusi nagu detsimeerimine, interpoleerimine või filtreerimine. Sobiva funktsionaalsusega ploki puudumisel võimaldab Gnu Radio ka oma plokkide lisada, mis peavad olema kirjutatud kas Python või C++ keeltes.

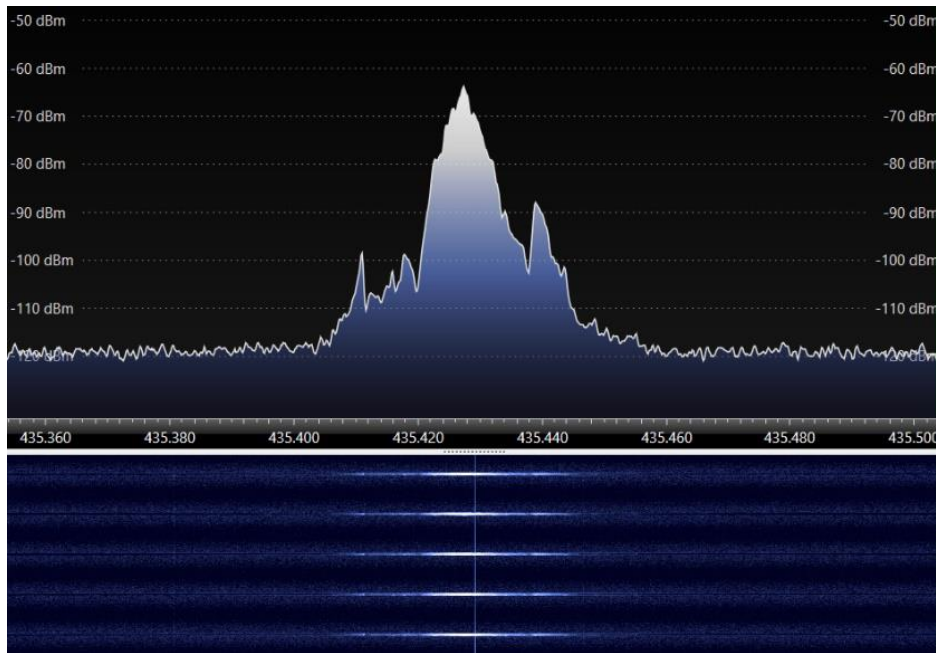
Lisaks paljudele programmiplokkidele sisaldab Gnu Radio ka graafilist liidest Gnu Radio Companion, mille abil on võimalik joonistada signaali töötlemise skeem ehk ühendada plokkid omavahel, mis konverteeritakse programmi käivitamisel Python koodiks. Selline andmetöötlusviis on mugav tänu sellele, et annab võimalust pöörata rohkem tähelepanu just signaali töötlemise printsiibile, mitte koodi kirjutamisele [15].

3.3 Vastuvõtja disain GNU Radio abil

Töö käigus koostati Gnu Radio tarkvaras kaks skeemi: üks skeem salvestab vastuvõetud signaal faili, teise skeemiga toimub signaali demoduleerimine.

3.3.1 Signaali salvestamine

Kõigepealt enne skeemi koostamist oli vaja leida, millisel sagedusel signaal asub. Selleks kasutati SDR Console tarkvara. Joonisel 9 on näha, et signaal asub sagedusel 435,430 MHz.

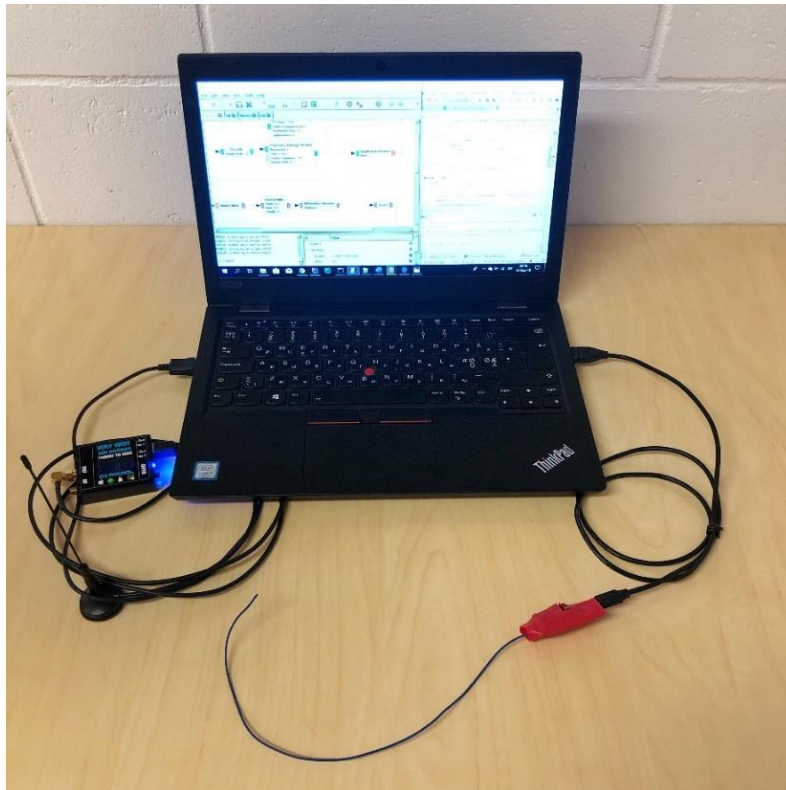


Joonis 9. Signaali spekter SDR Console tarkvaras.

Selleks, et vastuvõetud signaali oleks hiljem mugavam töödelda, salvestatakse see faili.

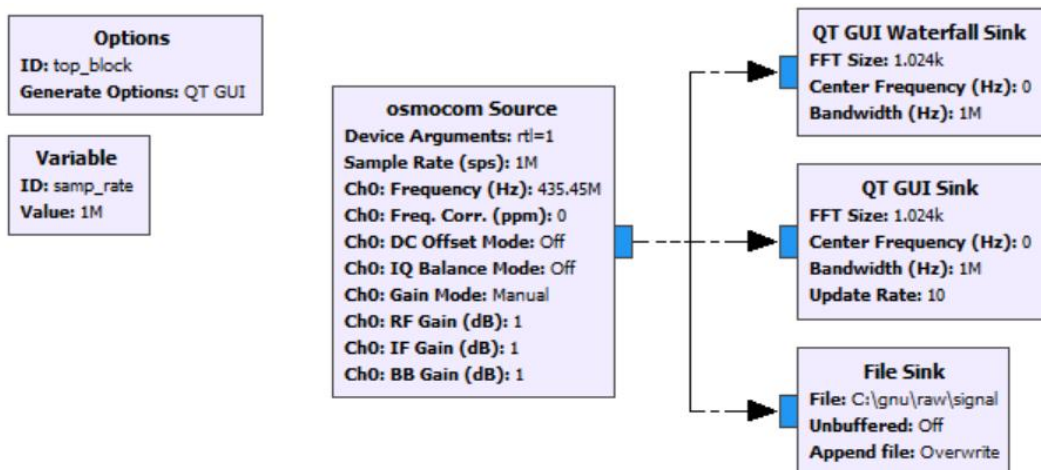
Joonisel 10 on võimalik näha saatja ja vastuvõtja seadistus. Kõigepealt on arvuti külge ühendatud transiiver (vt joonis, paremal) USB pordi kaudu. Transiiver pannakse saatma sõnumeid Java keeles kirjutatud programmiga, mille on koostanud teine inimene. On tähtis, et sisestatud kaadril oleksid kindlasti alguses ja lõpus lipud ning õige CRC enne lõpulippu, vastasel juhul transiiver signaali välja ei saada.

SDR vastuvõtja koos antenniga (pildil vasakul) on samuti ühendatud arvuti külge USB pordi kaudu. Arvutis on käima pandud Gnu Radio skeem, mis salvestab vastuvõetud signaali.



Joonis 10. Saatja ja vastuvõtja seadistus.

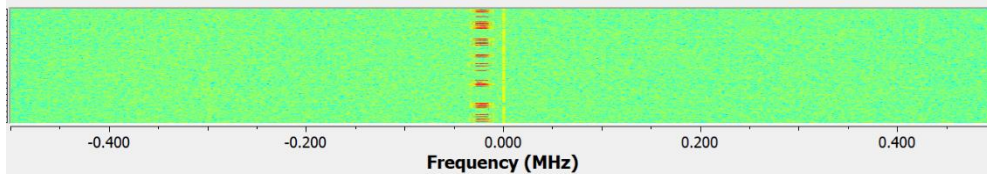
Joonisel 11 on kujutatud signaali salvestamiseks koostatud skeem.



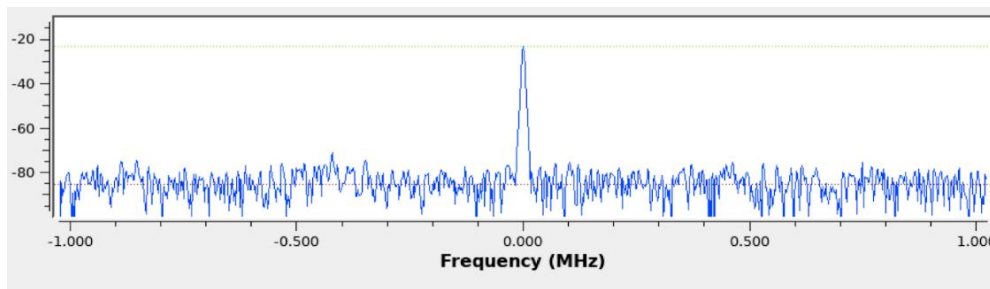
Joonis 11. Skeem signaali salvestamiseks

Skeemi plokid:

QT GUI Waterfall Sink ja *QT GUI Sink* on signaali visualiseerimise plokid. *QT GUI Waterfall Sink* kujutab signaali spektrogrammil ning *QT GUI Sink* näitab signaali spektrit.



Joonis 12. Signaali spektrogramm.



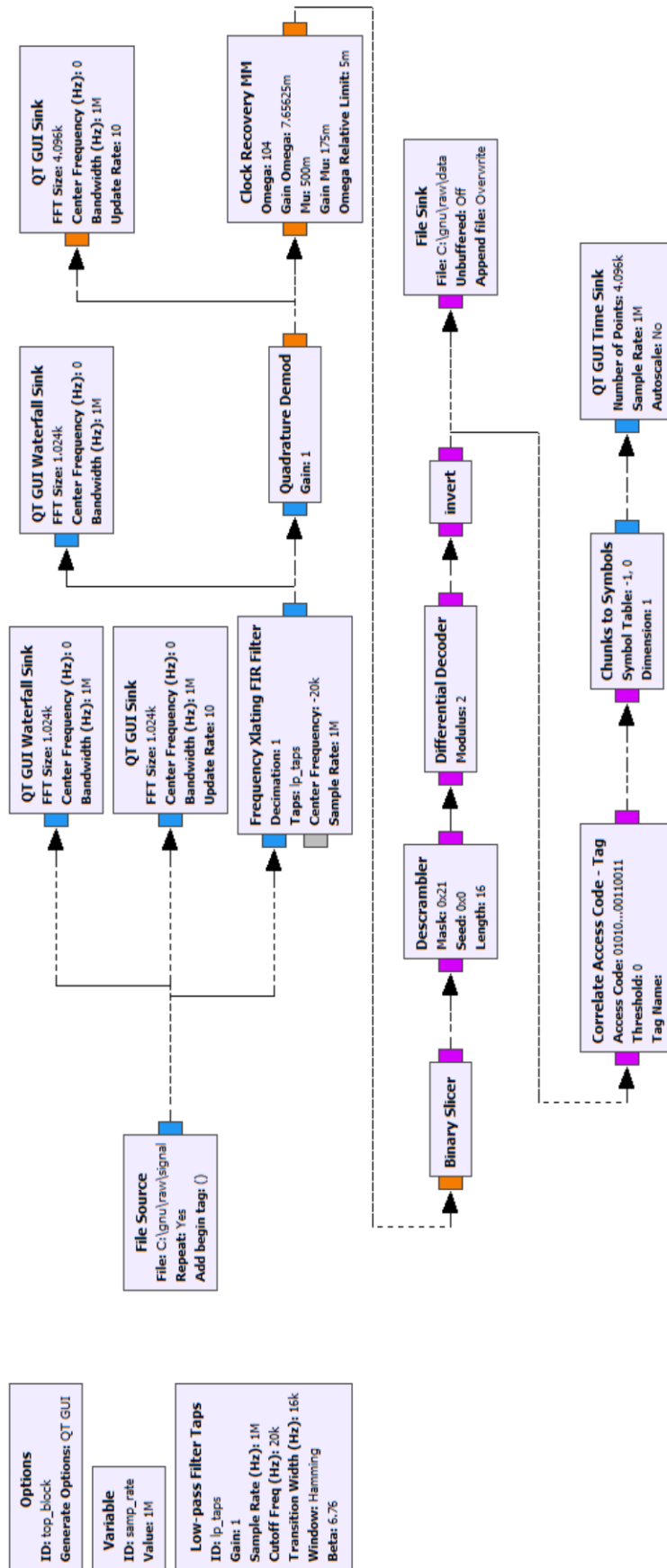
Joonis 13. Signaali spekter.

Osmocom Source plokk sobib peaaegu kõikide vastuvõtjate juhtimiseks. Selleks, et oleks võimalik kasutada RTL2832 kiibistikku signaali allikana, tuleb kõigepealt installeerida rtl-sdr teeki. Kui teek on õigesti installeeritud, peab *Osmocom Source* plokk ilmuma Gnu Radio menüüs 'Source' kategooria plokkide hulka [16].

Kuna on teada, et signaal saadetakse sagedusel 435,45 MHz, on ploki parameetrik *Ch0:Frequency* valitud sama sagedus. Diskreetimissagedus on 1 MHz. Võimendused on võrdsed 1-ga, kuna saatja ja vastuvõtja asuvad väga lähedal ning signaal on tugev. Kui tegu oleks reaalse satelliidiga, siis tulenevalt satelliidi ja maajaama suurest vahest oleks signaali vaja võimendada.

3.3.2 Signaali demoduleerimine

Joonisel 14 on võimalik näha Gnu Radio abil koostatud voodiagramm FSK signaali demoduleerimiseks. Vastuvõetud signaal kõigepealt filtreeritakse, seejärel demoduleeritakse. Moduleeritud signaal deskrambleeritakse ning viiakse tagasi NRZ kodeeringusse. Demoduleeritud ja dekodeeritud signaal salvestatakse. Plokke ja nende parameetreid on pikemalt kirjeldatud allpool.



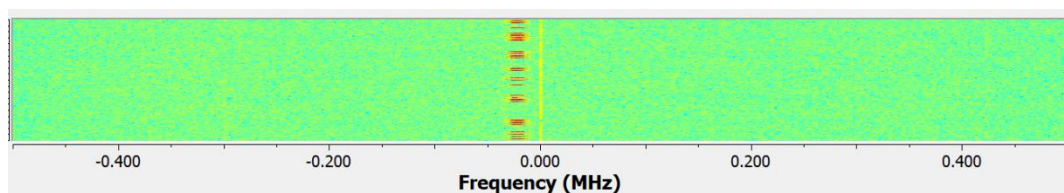
Joonis 14. Signaali demoduleerimine.

Plokid:

File Source on plokk, mis loeb failist binaarformaadis olevat signaali. Plokist väljuv signaal on kompleksne.

File Sink on plokk, mis salvestab signaali faili binaarformaadis.

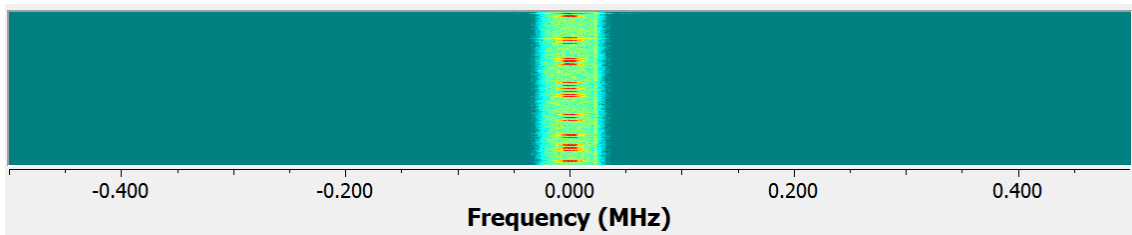
Frequency Xlating FIR Filter on plokk, mis teeb korraga kahte asja: määrab kesksageduse ehk nihutab signaali sageduse 0 Hz juurde ning peale seda filtreerib madalpääsfiltri abil kõik kõrgemad sagedused välja [17]. Kuna vastuvõetava signaali ribalaius on umbes 20 kHz, valiti madalpääsfiltri lõikesageduseks 20 kHz. Liiga suure lõikesagedusega pääseb lisaks signaalile läbi ka palju müra ja võimalikke häireid. Kuna selles plokis detsimeerimine on võrdne 1-ga, peab diskreetimissagedus jääma samaks nagu sisendsignaali diskreetimissagedus. Teistsuguse detsimeerimisväärtuse korral peaks diskreetimissagedus olema võrdne sisendsignaali diskreetimissagedusega, mis on jagatud detsimeerimise väärtusega. Joonisel 15 on näha vastuvõetud signaali spektrogramm.



Joonis 15. Signaali spektrogramm.

Spektrogrammil on näha, et signaal on nihutatud nullist vasakule. Kuna töö eelmises peatükis leiti, et vaatamata sellele, et saatja saadab signaal sagedusel 435,45 MHz, asub see reaalselt sagedusel 435,43 MHz. Toetudes sellele on võimalik välja arvutada, kui palju tuleb signaali nihutada ehk *Center Frequency* väärtus on võrdne $435,43 - 435,45 = -20$ kHz.

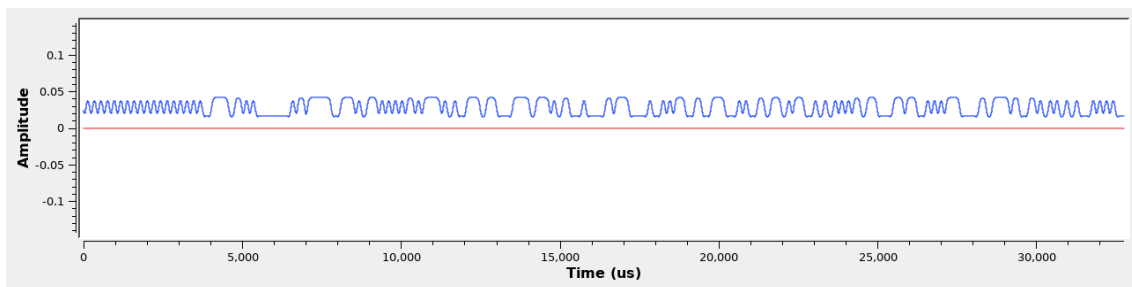
Joonisel 16 on näha, kuidas näeb välja spektrogramm pärast signaali nihutamist ja filtreerimist.



Joonis 16. Nihutatud signaali spektrogramm.

Pärast filtreerimist läheb signaal demodulaatorisse, mis koosneb kolmest ploki: *Quadrature Demod*, *Clock Recovery MM* ja *Binary Slicer*.

Quadrature Demod arvutab faasi erinevust kahe järjestikuse diskreedi vahel ja korrutab selle kasutaja määratava konstandiga ehk *Gain* parameetri väärtusega. Demodulaatori väljundsignaal kujutab endast faaside erinevuste voogu. Kui aktiivne oli suurem sagedus, peab väärtus olema nullist suurem, kui aktiivne oli madalam sagedus, peab väärtus olema nullist väiksem [18].



Joonis 17. Quadrature demode ploki väljundsignaal.

Clock Recovery MM on plokk, mille ülesanne on taastada kodeeritud sümbolid ning leida sümboli algus- ja lõppkoht. Selles ploki on 5 parameetrit. *Omega* näitab, mitu diskreedi kuulub ühe sümboli peale ehk teisisõnu tähendab see parameeter detsimeerimistegurit. *Mu* on saatja ja vastuvõtja vahel leitud faasinihe. *Omega Relative Limit* piiritleb *Omega* maksimaalset ja minimaalset võimalikku väärtust. *Gain Mu* ja *Gain Omega* reguleerivad tööaja jooksul sagedust ja faasi [19].

Kõikidele parameetritele peale *Omega* on jäetud vaikimisi pandud väärtused. *Omega* on arvatud valemiga:

$$\omega = \frac{\text{diskreetimissagedus}}{\text{edastuskiirus}} = \frac{1000000}{9600} = 104 \text{ (sps)}.$$

Binary Slicer on plokk, mis markeerib kõik nullist suuremad sisendsignaali väärtused kui „1” ning kõik sisendsignaali väiksemad väärtused kui „0”. Tegelikult tähistab ta neid kui <0x01> ja <0x00> [20].

Descrambler plokis toimub andmete deskrambleerimine. Selles plokis on 3 parameetrit: polünoommask (*mask*), algväärtus (*seed*) ja nihkeregistri pikkus (*length*). Polünoommaski saab arvutada skrambleerimise polünoomi põhjal ning välja arvatud väärtus on võrdne 0x21. Kuna satelliidis kasutatakse isesünkroniseeruvat skramblerit, võib algväärtus olla ükskõik milline ning antud juhul on selle väärtus 0x00. Nihkeregistri pikkus on võrdne 16 [21].

Differential Decoder ja *Invert* on plokid, mis koos teostavad NRZI dekodeeringut.

Correlate Access Code – Tag ploki *Access Code* parameetrina määratakse bitijada, mida tuleb leida ning tähistada lipuga. Ehk kui *Access Code* on võrdne „01111110”, mis tähistab AX.25 kaadri algust ja lõppu, siis iga kord, kui bitivoos ilmneb selline bitijada, tähistatakse see lipuga. Siinses töös otsiti kaadri algust mitte lipu järgi, vaid saadetud andmete esimeste baitide järgi.

Vastuvõtja testimiseks saadeti sõnum, mis on esitatud kuueteistkümnendsüsteemis:

7E455331572F53604553315A57006103F00493000074657265531F7E

Vastuvõetud sõnum, mis on esitatud binaarkoodis, peab vastuvõtul olema selline:

01111110 10100010 11001010 10001100 11101010 11110100 11001010 00000110
10100010 11001010 10001100 01011010 11101010 00000000 10000110 11000000
00001111 00100000 11001001 00000000 00000000 00101110 10100110 01001110
10100110 01010011 00011111 01111110

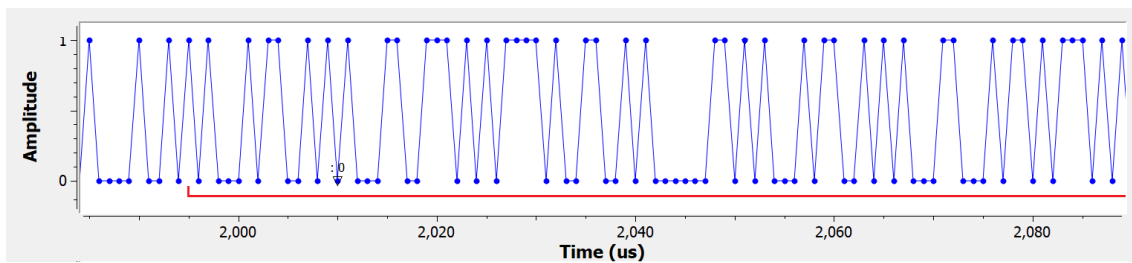
Nagu juba eespool kirjutatud, AX.25 kaadri kõik väljad peale FCS välja edastatakse madalaim bitt enne.

Valminud vastuvõtjaga demoduleeritud sõnum:

00011111 10100010 11001010 10001100 11101010 11110100 11001010 00000110
10100010 11001010 10001100 01011010 11101010 00000000 10000110 11000000
00001111 00100000 11001001 00000000 00000000 00101110 10100110 01001110
10100110 00010011 00010111 01111110

Kollasega on esile tõstetud baidid, mida vastuvõtjal ei õnnestunud hetkel õigesti demoduleerida. Üheks põhjuseks võib olla see, et kasutatud skrambleris puudub algväärtustamine.

Joonis 18 kujutab vastuvõetud ja demoduleeritud kaadri algust.



Joonis 18. Demoduleeritud sõnum.

4 Pakettide saatmine ja vastuvõtmine Dire Wolf tarkvara abil

Lõputöö teise osana realiseeriti pakettide saatmine ja vastuvõtmine Dire Wolf tarkvara abil. Selles osas kasutati SDR vastuvõtja asemel Si446x raadiot. Signaale saadakse Kenwood raadioga. Vastuvõetud paketid salvestati faili.

Kuna signaalide demodulatsioon on väga tihedalt seotud ka teiste satelliidi alamsüsteemidega, realiseeriti see osa tööst meeskonnas koos Madis Kaalu ja Ivo Mürseppaga.

4.1 Kenwood raadio

Kenwood TM-D710G on VHF/UHF amatöörraadio, millega on võimalik nii saata kui ka vastu võtta raadiosignaale. Kenwood raadiol on sisseehitatud TNC, mis ühildub AX.25 protokolliga. Ühendamine arvutiga võimaldab ka pakettandmesidet kiirusel 1200/9600 baud [22].

4.2 Dire Wolf, TNC, KISS

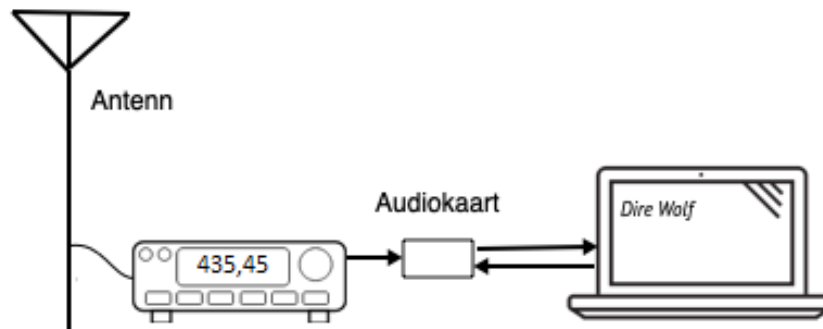
Mõni aeg tagasi pidi arvuti ja raadio vahel olema spetsiaalne seade – TNC modem. Kui eesmärgiks on andmete edastamine, siis TNC paneb kokku kaadri arvutist saadud andmetest, kontrollib vead (arvutab CRC) ning moduleerib signaali vastava sagedusega, et seda oleks võimalik edasi saata arvuti külge ühendatud raadio kaudu. Kui eesmärgiks on andmete vastuvõtmine, siis TNC pöörab edastamise protsessi ümber, ning vastuvõetud signaal demoduleeritakse ja saadud bitivoog saadetakse arvutisse [23].

Kuna tänapäeva arvutid on palju võimekamad, on võimalik asendada TNC funktsioone arvutis töötava tarkvara abil.

KISS TNC võimaldab luua kommunikatsiooni arvuti ja TNC vahel KISS protokollil abil. KISS (*keep it simple, stupid*) on protokoll, mille eesmärk on anda arvutile täielik kontroll TNC funktsioonide üle võimalikult madalal tasemel ehk viia mõned TNC funktsioonid arvutisse. KISS protokoll võimaldab eraldada kaadrid ning saata käsk TNC-sse. [24].

Dire Wolf on avatud koodiga tarkvara, mis asendab füüsilist TNC-d, kasutades selleks KISS protokolliga TCP socketi peal [23].

Joonisel 19 on näha, kuidas on omavahel ühendatud raadio, antenn ja arvuti. Nagu näha, on raadio külge ühendatud antenn, raadio on omakorda ühendatud audiokaardiga ning audiokaart on ühendatud arvuti külge USB pordi kaudu. Arvutis on käima pandud Dire Wolf.



Joonis 19. Kenwood, antenni, audiokaardi ja arvuti seadistus.

4.3 Pakettide vastuvõtmine

Esimese sammuna realiseeriti pakettide vastuvõtmine. Selleks tehti muudatused konfiguratsioonifailis nii, et Dire Wolf softimodem oleks võimeline kasutama modemit sobiva kiirusega ning valiks sobivad heliseaded. Vastuvõetud sõnumeid töödeldi ja salvestati faili kindlas formaadis.

4.3.1 Konfiguratsioon

Selleks, et maajaamas oleks võimalik Kenwood TM-D710G koos Dire Wolf-iga kasutada, peab raadio olema ühendatud arvuti külge mitme juhtme abil:

1. Antenni pööramiseks
2. Raadio sageduse tuunimiseks
3. Raadio saatja aktiveerimiseks (*Push To Talk*)
4. USB audiokaardi jaoks

Kui kõik juhtmed on paika pandud, tuleb konfiguratsioonifail konfigureerida nii, et Dire Wolf kasutaks modemit õige kiirusega ning oleks võimeline raadioga saatma ja vastu võtma sõnumeid.

Kõigepealt tuleb valida, millist heliseadet hakkab Dire Wolf kasutama. Kõik heliseaded on nähtavad peaknas, kui Dire Wolf käima pannakse. Kuna kasutatakse mitte arvutisse sisseehitatud, vaid välist audiokaarti, mis on ühendatud arvutiga juhtme abil, on konfiguratsioonifailis valitud *ADEVICE USB* ehk Dire Wolf valib esimese heliseadme nendest, mis on ühendatud USB pordi kaudu.

Seejärel valiti modemi kiirus, milleks on 9600 baud. On tähtis meeles pidada, et kuna kiirus on suur, ei ole võimalik kasutada mikrofoni ja kõlareid [23].

Saatja aktiveerimiseks on mitu võimalust. Siinses töös kasutatakse selleks PTT (*Push To Talk*). Kuna PTT on ühendatud arvuti külge serial pordi (USB-jadapordi) kaudu, on kõigepealt vaja leida õige pordi nimi. Kui on valitud vale pordi nimi või PTT konfigureerimine jäetakse üldse tegemata, ilmub programmi käivitamisel peaknas veateade.

Dire Wolf toimib nagu virtuaalne TNC ning seepärast peab ta suhtlema kliendirakendustega KISS protokollil abil, mis peab olema TCP socketi peal. Pordi numbriks võib jääda vaikimisi pandud pordi number ehk 8001.

Dire Wolf võtab vastu ainult neid sõnumeid, mis on saabunud õige FCS-ga. Kuna mõned bitid võivad kaotsi minna, pakub Dire Wolf võimalust proovida rekonstrueerida neid sõnumeid, mis on saabunud vale FCS-ga. Vastavalt valitud konfiguratsioonile võib Dire Wolf üks kord inverteerida üht või mitut bitti ning arvutada FCS uuesti. Vaikimisi on konfiguratsiooni failis valitud FCS ülearvutamine ühe biti inverteerimisel. Siinses töös on FCS ülearvutamine välja lülitatud [23].

Lõplik konfiguratsiooni fail koosneb järgmistest seadistustest:

ADEVICE USB

ACHANNELS 1

CHANNEL 0

MYCALL ES1ZW

MODEM 9600

```
PTT COM9 RTS
KISSPORT 8001
FIX_BITS 0
```

Joonisel 20 on peaakna pilt.

```
Available audio output devices for transmit (*=selected):
 * 0: Speakers (2- USB Audio Device) (channel 0)
   1: Speaker/HP (Realtek High Defini
Channel 0: 9600 baud, K9NG/G3RUH, +, 44100 sample rate x 2.
The ratio of audio samples per sec (44100) to data rate in baud (9600) is 4.6
This is on the low side for best performance. Can you use a higher sample rate?
Ready to accept AGW client application 0 on port 8000 ...
Ready to accept KISS TCP client application 0 on port 8001 ...

Attached to KISS TCP client application 0 ...

Ready to accept KISS TCP client application 1 on port 8001 ...
```

Joonis 20. Dire Wolfi avavaade.

4.3.2 Vastuvõetud sõnumite salvestamine

Vaikimisi Dire Wolf vastuvõetud sõnumeid ei salvesta, vaid need kuvatakse programmi peaaknas juba demoduleerituna ja dekodeerituna ASCII-formaadis. Allpool oleval joonisel on võimalik näha, kuidas näeb välja vastuvõetud sõnum. Nagu on näha, dekodeeris Dire Wolf ise sõnumi ASCII formaati. Sümbolid, mida ei ole võimalik dekodeerida, esitatakse *hex*-formaadis.

```
ES1WS audio level = 20(+37/-34)   |||||___
[0.2] ES1WS>ES1ZW:<0x1f><0x00><0x00>HELLO
```

Joonis 21. Dekodeeritud sõnum.

Eesmärgiks on salvestada vastuvõetud AX.25 struktuuriga kaader *hex*-formaadis ilma lipu ja FCS-ta ehk kaader peab koosnema aadressist, PID väljast, kontrollväljast ja informatsioonist.

Dire Wolf pakub küll mitmesuguseid võimalusi andmete salvestamiseks, kuid nendest ei sobi ükski, sest andmed salvestatakse valel kujul.

Otsustati kasutada käsureaprogrammi „kissutil”, mis on arvutisse automaatselt installeeritud koos Dire Wolfiga. „Kissutil“ on lihtne kasutajaliides KISS TNC ja kasutaja vahel, mis proovib ühenduda TCP KISS TNC juurde [23].

Joonisel 22 on näha, millisel kujul „kissutil” kuvab andmeid ekraanile. Mitme saadud sõnumi korral paneb „kissutil” need lihtsalt üksteise alla. Kaks esimest baiti ja üks viimane bait on KISS protokollilipud ning kõik, mis on nende vahel, ongi vajalik osa kaadrist, mida tuleb salvestada.

```
C:\Users\usako\direwolf-1.5-win>kissutil -o rec2 -v
From KISS TNC:
 000: c0 00 8a a6 62 b4 ae 40 60 8a a6 62 ae a6 40 61   ...b..@`..b..@a
 010: 03 f0 e0 23 00 00 48 45 4c 4c 4f c0               ...#..HELLO.
[0] ES1WS>ES1ZW:α#<0x00><0x00>HELLO
Save received frame to rec2\20190415-150217-400
From KISS TNC:
 000: c0 00 8a a6 62 b4 ae 40 60 8a a6 62 ae a6 40 61   ...b..@`..b..@a
 010: 03 f0 e0 24 00 00 48 45 4c 4c 4f c0               ...$.HELLO.
[0] ES1WS>ES1ZW:α$<0x00><0x00>HELLO
```

Joonis 22. Kissutil väljund.

Python keeles kirjutati kood, mis käivitab „kissutil“-i, loeb käsurea aknas ilmuvaid ridu, töötleb neid ning salvestab. Samuti on võimalus saata edasi TCP pordi kaudu õiges formaadis kokku pandud kaader.

Valminud kaader näeb välja selline:

8aa662b4ae40608aa662aea6406103f0e023000048454c4c4f

Oranž – vastuvõtja aadress;

Kollane – saatja aadress;

Roosa – kontrollväli;

Sinine – PID väli;

Roheline – informatsioon.

4.4 Pakettide saatmine

Teises peatükis juba kirjeldati AX.25 kaadri struktuuri. Maajaama vastuvõtja jaoks on selline struktuur täiesti sobilik, kuid satelliidi vastuvõtja vajab lisapäist ja lisalippe AX.25 kaadri ees selleks, et sõnumit kätte saada. Sellepärast tuleb Dire Wolfi tarkvara koodi täiendada nii, et saadetavate AX.25 kaadrite ette pandaks Si446x raadiokivile vajalikud bitid.

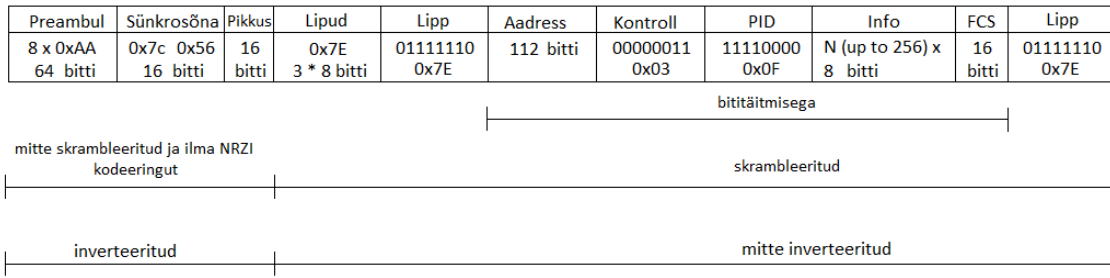
Lisapäis peab koosnema 12 baidist. Kõigepealt tuleb saata preambul, mis koosneb 8 baidist ning iga bait on esitatud kuueteistkümnendsüsteemis väärtusega 0xAA. Preambulit on vaja teatud andmete alguse tähistamiseks ja vastuvõtja sünkroniseerimiseks andmetega. Pärast preambulit saadetakse kaks baiti väärtustega 0x7c ja 0x56, see on sünkrosõna. Järgmisena saadetakse välja arvutatud kaadri pikkus, mis peab samuti koosnema kahest baidist ning peab olema esitatud nii, et vanemad bitid saadetakse varem. Näiteks kui kaadri pikkus on 258 baiti, siis pikkuse baidid on võrdsed 0x01 ja 0x02 ning 0x01 saadetakse esimesena. Kõik need lisatud baidid, nagu ka kaader, tuleb lasta läbi skrambleri. Seda tehakse selleks, et skrambler saaks õige seisu. Vaatamata sellele, et kõik baidid tuleb skramblerisse lasta, saadetakse lisabaidid vastuvõtjale mittedskrambleerituna ning kaader saadetakse skrambleerituna.

Peale lisapäist saadetakse 3 lippu väärtusega 0x7E, mida on vaja selleks, et sünkroniseerida deskramblerit.

Testimise käigus selgus, et lisatud Si446x raadio lisapäise saatmisel tuleb lisaks skrambleri keelamisele ka bittide NRZI kodeerimine ära jätta. Sellega tekkis uus probleem, sest NRZI kodeeringus bitid on polaarsusest sõltumatud, aga ilma kodeeringuta bitid peavad olema õige polaarsusega. Samuti selgus katsetamisel, et Si446x raadio saab saadetud bitid kätte inversioonis – põhjuseks on raadiosaatjas signaali invertteerimine. Selle parandamiseks tuli lisapäis saatmisel veelkord invertteerida. Lisaks on sünkrosõna ja pikkuse baidid saadetud pööratuna.

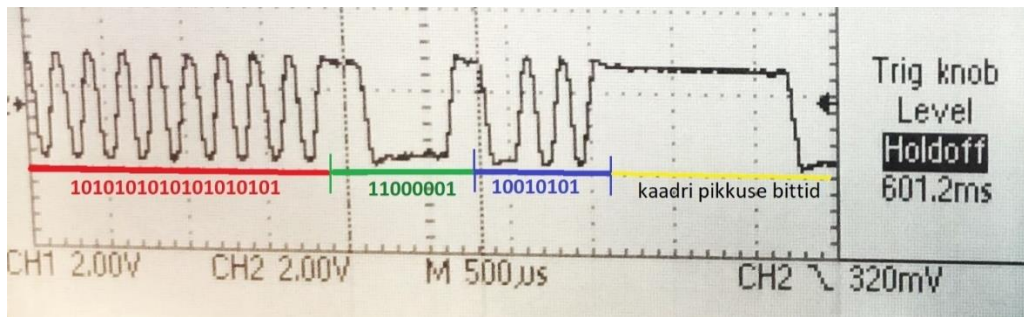
Nagu on juba teises peatükis öeldud, kasutatakse AX.25 kaadrites bititaitmist (*bit stuffing*), et välistada olukorra tekkimist, kui kaadri sees esineb juhuslikult bait, mis on võrdne kaadri lippudega ehk bait, mille väärtus on 0x7E. Bititaitmist ei pea rakendama lisapäise jaoks. Samuti kuna bititaitmise tõttu võivad kaadri sees tekkida lisabitid, tuleb kaadri pikkuse välja arvutamisel arvestada juurde tekkinud lisabittidega.

Joonisel 24 on esitatud AX.25 kaadri struktuur koos lisapäisega ja lisalipudega.



Joonis 23. Lisapäis ja lisalipud koos kaadriga.

Joonisel 24 on näha bitijada, mille võttis vastu Si644x raadio. Kõigepealt on näha preambuli lõpp, seejärel on kaks sünkrosõna baiti ning edasi on kaadri pikkus, mis on ka esitatud kahe baidina.



Joonis 24. Si644x sisendsignaali näide.

Allpool on näidatud, kuidas on esitatud baidid sünkrosõna näidel.

hex-formaadis	0x7c	0x56
binaarkoodis	01111110	01010110
inverteeritud	10000011	10101001
Inverteeritud ja pööratud	11000001	10010101

Tabel 3. Si446x raadioga vastuvõetud sünkrosõna.

Satelliidi raadio võttis edukalt vastu Kenwoodiga saadetud signaali, millest võib järeldada, et välja saadetud lisapäis ja kaader on õige struktuuriga.

5 Tulemuste valideerimine

Esimeses praktilises osas oli loodud Gnu Radios kaks skeemi: üks vastuvõetud signaali salvestamiseks ja teine demoduleerimiseks. Selleks, et veenduda, et skeem töötab korrektselt, saadeti kõigepealt välja kindel sõnum Si446x saatja abil. SDR vastuvõtjaga püüti see signaal ning salvestati faili binaarformaadis Gnu Radios signaali salvestamiseks koostatud skeemi abil. Seejärel prooviti teise skeemi abil signaali demoduleerida. Kuna vastuvõetud sõnum peab olema esitatud madalaim bitt enne, siis bitijadast otsiti väljasaadetud sõnum pööratud kujul. Selleks kasutatakse *Correlate Access Code – Tag* plokki, mis otsib bitijadast parameetriga antud bittide kombinatsiooni.

Teises praktilises osas realiseeriti kõigepealt pakettide vastuvõtmine ning salvestamine. Vastuvõetud sõnum peab kattuma välja saadetud sõnumiga. Seda oli võimalik näha peaaknas. Kui raadio võtab vastu paketti Dire Wolf-i abil, siis „kissutil” konsoolis kuvatakse saadud sõnum *hex*-formaadis. Sama sõnum, aga ilma KISS protokollilippudeta peab olema salvestatud ka faili.

Pakettide saatmiseks Dire Wolf tarkvara abil täiendati Dire Wolf-i koodi. Testimiseks oli kasutatud Kenwood raadiot ja satelliidi raadiot Si446x. Kenwoodiga saadeti välja sõnum ning Si446x võttis selle vastu. Kõigepealt tehti kindlaks, et Kenwood saadab välja signaali ning selle tugevus on piisav. Seejärel prooviti muuta lisapäist ning vaadati, kas vastuvõtjal õnnestub seda vastu võtta. Katsete käigus selgus, et lisapäis peab olema ilma NRZI kodeeringuta ning inverteeritud kujul.

6 Kokkuvõte

Bakalaureusetöö esimene eesmärk oli valmistada vastuvõtja, mis suudaks võtta vastu satelliidiga saadetud signaale ning neid demoduleerida. Signaalide vastu võtmiseks ja töötlemiseks kasutati Gnu Radio tarkvara, kus autor koostas skeemid signaali salvestamiseks ning salvestatud signaali demoduleerimiseks.

Teine eesmärk oli realiseerida pakettide saatmine ja vastuvõtmine Dire Wolf tarkvara abil. Kuna satelliidi raadio puhul on vaja, et vastuvõetavate pakettide ees oleksid lisabitid, täiendati Dire Wolf tarkvara koodi nii, et saadetavate pakettide ette pannakse radiokivile vajalik lisapäis. Koodi täiendamine realiseeriti koos Madis Kaaluga, kes aitas ja suunas autorit. Vastuvõetud pakettide salvestamiseks kirjutati meeskonnatöös autoriga programm Python keeles, mis käivitab „kissutil” konsooli ning salvestab sõnumid vajalikul kujul.

Bakalaureusetöö esimene osa on teoreetiline ning seletab moduleerimise, kodeerimise ning kaadri struktuuri olemust. Teine osa kirjeldab valminud vastuvõtjat, mille koostamiseks kasutati tarkvaralist raadiot ja Gnu Radio raamistikku. Töö kolmandas osas realiseeritakse sõnumite saatmist ja vastuvõtmist Kenwood raadio ja Dire Wolfi tarkvara abil.

Kokkuvõtteks koostati autori osalusel kaks skeemi Gnu Radios, kirjutati Python kood pakettide salvestamiseks ning täiendati Dire Wolfi kood vastavalt vajadusele.

Vaatamata sellele, et töö alguses püstitatud eesmärgid õnnestus saavutada, on veel palju arenguruumi. On vaja edasi töötada tarkvaralisel raadiol põhineva vastuvõtjaga ning täiendada Gnu Radio skeemi nii, et vastuvõtja oleks võimeline eristama paketi algust lipu järgi.

Gnu Radio skeemid signaali salvestamiseks ja demoduleerimiseks ning Python kood pakettide salvestamiseks on kättesaadavad githubis aadressil: <https://github.com/svusak/thesis>.

Täiendatud Dire Wolf tarkvara kood asub githubis aadressil:
<https://github.com/svusak/direwolf>. Täiendati viit klassi: hdlc_send.c, xmit.c, direwolf.c,
gen_tone.c, gen_packets.c.

Kasutatud kirjandus

- [1] «TTÜ100 Satelliidi tutvustus,» [Võrgumaterjal]. Available: <https://ttu.ee/projektid/mektory-est/satelliidiprogramm-4/satelliidiprogramm/>. [Kasutatud: 14.05.2019].
- [2] A. Meister, Modulatsioon, Tallinn, 1999.
- [3] S. Bahurin, «Частотно-манипулированные сигналы FSK (frequency shift key) и FSK сигналы с непрерывной фазой CPFSK (continuous phase FSK),» [Võrgumaterjal]. Available: http://ru.dsplib.org/content/signal_fsk/signal_fsk.html. [Kasutatud: 15.03.2019].
- [4] S. H.Mnoney, An Introduction to Digital Signal Processing: A Focus on Implementation, Denmark: River Publishers ApS, 2008.
- [5] J. Miller, «9600 Baud Packet Radio Modem Design,» [Võrgumaterjal]. Available: <https://www.amsat.org/amsat/articles/g3ruh/109.html>. [Kasutatud: 16.03.2019].
- [6] «Скремблер,» [Võrgumaterjal]. Available: <https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80>. [Kasutatud: 16.03.2019].
- [7] D. Anderson, Universal Serial Bus System Architecture, Addison-Wesley, 2001.
- [8] W. A. Beech, D. E. Nielsen и J. Taylor, «AX.25 Link Access Protocol for Amateur Packet Radio,» [Võrgumaterjal]. Available: <http://www.ax25.net/AX25.2.2-Jul%2098-2.pdf>. [Kasutatud: 20.03.2019].
- [9] «Digital Packet Radio – APRS,» [Võrgumaterjal]. Available: <https://web.stanford.edu/class/ee179/labs/Lab7.html>.
- [10] I. Müürsepp, «Kanalikiht,» [Võrgumaterjal]. Available: <https://www.lr.ttu.ee/side/2016sygis/Loeng06.pdf>. [Kasutatud: 20.03.2019].
- [11] «HIGH LEVEL DATA LINK CONTROL PROCEDURES,» 1986. [Võrgumaterjal]. Available: <https://law.resource.org/pub/in/bis/S04/is.11418.1.1986.pdf>.
- [12] J. H. Reed, Software Radio: A Modern Approach to Radio Engineering, 2002.
- [13] «DXpatrol-MK3-SDR,» [Võrgumaterjal]. Available: www.dxpathrol.pt/images/Docs/DXpatrol-MK3-SDR.pdf. [Kasutatud: 10. 03. 2019].
- [14] «HIGH-PERFORMANCE, LOW-CURRENT TRANSCEIVER,» [Võrgumaterjal]. Available: <https://www.silabs.com/documents/public/data-sheets/Si4463-61-60-C.pdf>. [Kasutatud: 20.04.2019].
- [15] «What is GNU Radio?,» [Võrgumaterjal]. Available: https://wiki.gnuradio.org/index.php/What_is_GNU_Radio%3F. [Kasutatud: 10.03.2019].
- [16] «gr-osmosdr,» [Võrgumaterjal]. Available: <https://osmocom.org/projects/gr-osmosdr/wiki>. [Kasutatud: 25.04.2019].

- [17] «GNU Radio Companion Filters,» [Võrgumaterjal]. Available: https://www.ece.uvic.ca/~ece350/grc_doc/ar01s12s08.html. [Kasutatud: 25.04.2019].
- [18] «GNU Radio Manual and C++ API Reference,» [Võrgumaterjal]. Available: https://www.gnuradio.org/doc/doxygen/classgr_1_1analog_1_1quadrature__demod__cf.html. [Kasutatud: 25.04.2019].
- [19] T. Šolc, «NOTES ON M&M CLOCK RECOVERY,» [Võrgumaterjal]. Available: https://www.tablix.org/~avian/blog/archives/2015/03/notes_on_m_m_clock_recovery/. [Kasutatud: 25.04.2019].
- [20] «Binary Slicer,» [Võrgumaterjal]. Available: https://www.gnuradio.org/doc/doxygen/classgr_1_1digital_1_1binary__slicer__fb.html. [Kasutatud: 25.04.2019].
- [21] D. Estévez, «Scramblers and their implementation in GNUradio,» [Võrgumaterjal]. Available: <https://destevez.net/2016/05/scramblers-and-their-implementation-in-gnuradio/>. [Kasutatud: 25.04.2019].
- [22] «TM-D710A/E/GA/GE Multi-communicator VHF/UHF FM Dual Bander,» [Võrgumaterjal]. Available: https://www.kenwood.com/i/products/info/amateur/pdf/TM-D710A_E_GA_GE_IDM.pdf. [Kasutatud: 20.04.2019].
- [23] «Documentation for Dire Wolf,» [Võrgumaterjal]. Available: <https://github.com/wb2osz/direwolf/blob/dev/doc/README.md>. [Kasutatud: 15.04.2019].
- [24] M. Chepponis и P. Karn, «The KISS TNC: A simple Host-to-TNC communications protocol,» [Võrgumaterjal]. Available: <http://www.ax25.net/kiss.aspx>. [Kasutatud: 15.04.2019].