

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Thomas Johann Seebecki elektroonikainstituut

IEE40LT

Andres Liba 134340IALB

**MEHITAMATA LENNUK RASBERRY PI 2
JA NAVIO BAASIL - SIDE JA
TELEMEETRIA**

Bakalaureusetöö

Juhendaja: Rauno Gordon
PhD
Vanemteadur

Tallinn 2016

Autorideklaratsioon

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

Autor: Andres Liba

06.06.2016

Annotatsioon

Töö eesmärgiks on mehitamata lennuki kokkupanek. Lennukiga peab olema stabiilne ja piiramatu sidustus ning kõik lennuki andmed peavad olema arvutist kättesaadavad. Lennukil on piisavalt andureid, et tagada stabiilne lend.

Töö on tehtud kahe inimesega. Alexander vastutab andurite eest, aga mina sidestuse eest.

Olulisemad probleemid on andurite sidustamine Raspberry Pi 2-ga, telemetria andmete saamine, käskude edastamine,

Tähtsamad tulemused on saada mehitamata lennuk, mida on võimalik juhtida piiramatult kauguselt, kas käsitsi või autopiloodi käskudega.

Järelduseks oleme harjutanud sidet organiseerida ja andurid siduda Raspberry Pi 2-ga.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 21 peatükki, 9 joonist, 7 tabelit.

Abstract

Unmanned plane using Raspberry Pi 2 and NAVIO - communications and telemetry

Main purpose of this work is assembling unmanned plane. Plane should have stable connection with ground-station and every sensors data should be accessible at PC. Sensors should provide opportunity to make stable flights.

Work is performed by two people. Alexander is responsible for sensors and Me for creating connection. Plane assembly is done together.

The most important problems are sensors integrating with Raspberry Pi 2, sensors calibration, telemetry data receiving, commands sending, flight planning. Work should be organised so that every person could manage with his part. Some components incompatible with main hardware.

Main results of this work is created unmanned plane that could controlled as manually either with autopilot commands. Plane got unlimited range of control. During flight plane could be observed as on map either through real-time camera. Furthermore, some unexpected problems were solved.

Significant conclusions – we are learnt to create some sensors and organize connection with plane, couple sensors with Raspberry Pi 2. Moreover some new network protocols were learnt. What is more everything should be backed up. We improved our teamwork ability.

The thesis is in Estonian and contains 30 pages of text, 21 chapters, 9 figures, 7 tables.

Lühendite ja mõistete sõnastik

3G	Kolmas põlvkond
4G	Neljas põlvkond
AHRS	Suuna ja taustsüsteem
APN	<i>Access point name</i> ehk interneti juurdepääsu punkt
BEC	<i>Battery eliminator circuit</i> ehk aku elimineeriv skeem
CPU	<i>Central processing unit</i> ehk keskprotsessor
DynDNS	<i>Dynamic domain name system</i> ehk dünaamiliste domeenide süsteem
ESC	<i>Electronic speed control</i> ehk elektriline kiiruskontroller
FPV	<i>First person view</i> ehk esimese isiku vaade
GCS	<i>Ground Control Station</i> ehk maapeal asuv juhtimiskeskus
GPIO	<i>General-purpose input/output</i> ehk üldotsatarbeline sisend/väljund
GPU	<i>Graphical Processing Unit</i> ehk graafikaprotsessor
GUI	<i>Graphical User interface</i> ehk graafiline kasutajaliides
HDMI	<i>High definition multimedia interface</i> ehk kõrglahutusega multimeedia kasutajaliides
I2C	<i>Inter-Integrated circuit</i> ehkkahejuhtmeliides
IP	<i>Internet protocol address</i> ehk internetiaadress
MAVlink	<i>Micro Aerial Vehicle Link</i> ehk mikro õhusõiduki ühendus
NAT	<i>Network Address Translation</i> ehk võrguaadresside tõlkimine
RC	<i>Radio control</i> ehk raadio kontroll
RCA	Audio- ja videotehnika pistikühendus
RTP	<i>Real-time Transport Protocol</i> ehk reaalaaja transport protokoll
RTCP	<i>Real-time Transport Control Protocol</i> ehk reaalaaja transpordi kontroll protokoll
SPI	<i>Serial Peripheral Interface</i> ehk jada välisseadme liidemispiistik
SSH	<i>Secure Shell</i> ehk turvakest
TCP	<i>Transmission control protocol</i> ehk edastusohje protokoll
TRRS	<i>Tip, Ring, Ring, Sleeve</i> pistikühendus

UDP

User Datagram Protocol ehk kasutajadatagrammi protokoll

VPN

Virtual Private Network ehk virtuaalne privaatvõrk

Sisukord

1 Sissejuhatus	10
2 Mehitamata lennuk	12
2.1 Mehitamata lennuki komponendid	12
2.2 Raspberry Pi	13
2.2.1 Raspberry Pi tarkvara ja riistvara	13
2.2.2 Ühenduse loomine	15
2.2.3 Autopiloodi tarkvara installeerimine ja käivitamine	16
2.3 3G USB Modem	17
2.3.1 3G võrgu eelistused ja puudused	17
2.3.2 Modemi sisse lülitamine	17
2.3.3 Wvdial	18
2.4 Andmeedastus.....	19
2.4.1 MAVlink.....	20
2.4.2 Probleem 3G võrguga ja serveri loomine	20
2.5 Videoside loomine ja kaamera valik	22
2.5.1 RTP ja RTCP protokollid	23
2.5.2 VPN	23
2.5.3 Gstreamer – video edastus ja saamine	23
2.6 Hilistumiseaeg	24
2.7 Automaatne käivitamine ja juhtkangi seadistamine	27
3 Kokkuvõte	28
Kasutatud kirjandus	29
Lisa 1 – Tabelite andmete saamine.....	31

Jooniste loetelu

Joonis 1. Navio kilp[1].	12
Joonis 2. Raspberry Pi 2B GPIO[1]	14
Joonis 3. Raspberry Pi B GPIO[1]	14
Joonis 4. Tekkitav viide rakendusega töö, roheline joon on Emlid Raspbian, punane joon tavaline Raspbian[1].....	15
Joonis 5. Wvdial.conf	19
Joonis 6. Ühendus Raspberry Pi-ga läbi serveri	21
Joonis 7. Video edastamiseks skript.....	24
Joonis 8. Kaameraga viide.....	31
Joonis 9. Ilma kaamerata viide	32

Tabelite loetelu

Tabel 1. Raspberry Pi parameetrite võrdlus.	13
Tabel 2. Kaamera võrdlus.....	22
Tabel 3. Juhtimis hilistumisaeg, kaamera on sisse lülitatud ja edastab videot.....	25
Tabel 4. Juhtimis hilistumisaeg kaamera on välja lülitatud ja ei edasta videot.....	25
Tabel 5. Interneti kaabliga video edastuse hilistumisaeg, CNE-CWC3 kaamera	26
Tabel 6. 3G võrgu kaudu video edastuse hilistumisaeg, suur ekraan, CNE-CWC3 kaamera.....	26
Tabel 7. 3G võrgu kaudu video edastuse hilistumisaeg, väike ekraan, CNE-CWC3 kaamera.....	27

1 Sissejuhatus

Mehitamata lennuk on õhusõiduk, mille pardal ei ole pilooti – see töötab kas iseseisavalt või kaugjuhtimisega[1] . Mehitamata lennukit võib kontrollida mitmel erineval viisil. Sellest viisist sõltub kui kaugemale saab lennata, kui kiiresti toimub andmeedastus, milliseid andmeid on üldse võimalik edastada. Tavaliselt kasutatakse juhtimiseks RC pulti, aga sellel meetodil on puudutused, näiteks kontrollimis kaugus on piiratud signaali levi kaugusega. Eksisteerivad ka teised sidustuse võimalused, näiteks mobiilsidevõrgu kaudu. Just selline sidustust on valitud, sest signaali kaugus ei ole piiratud.

Mehitamata lennuk koosneb mitmest kokku ühendatud seadmetest. Korraliku autopiloodilt kontrollitava lennu kindlustamiseks on vaja, et lennukil oleks vähemalt GPS, kõrguse andur, kiiruseandur, kompass, güroskoop, kiirendusmõõtur ja mingi vastuvõtja. Kõik need seadmed ühendab kokku tarkvara ArduPilot.

ArduPilot suhtleb GCS-l viibiva tarkvaraga Mission Planner MAVlinki kaudu. Sõltumatu ühenduse tüübilt on võimalik GCS-lt edastada kalibreerimise parameetrid, autopiloodi käsud ja vastu võtta telemeetria andmed ja jooksva lennu logiraamatusse märkmeid teha.

Peaesmärk on tekitada lennuk, mis lendab autopiloodi käskudega.

Minu eesmärk oli korraldada side ja telemeetria edastust, Alexandri eesmärk oli sensorite paigaldus ja autopiloodi seadistamine.

Mehitamata lennuki kokku panemiseks on vaja lahendada järgmised probleemid:

- Riistvara valik, millele laaditakse pärast ArduPilot püsivara
- Kaamera, juhtkangi ja 3G modemi valik

Kõigepealt ma vaatan milliseid komponendid mul juba on, pärast vaatan ja valin mida on vaja tellida, tutvun olemasolevate komponentide võimalustega, seadistan riistvara, valin ühendus tüübi, lahendan ilmunud probleemid, tegelen videoid loomisega, paralleelselt lahendan sellega seotud probleemid, kontrollin tekkinud süsteemide töövõimekust, mõõdan viidet, seadistan riistvara nii, et viide oleks minimaalne, tekitan varu koopiad.

Selle projekti lähtetingimused :

- On olemas Raspberry Pi 1 ja Raspberry Pi 2
- NAVIO kilp Raspberry Pi-le
- 3G USB Modem Huawei E182E ja Huawei E173
- Kaamera Canyon CNE-CWC3 ja FPV kaamera
- Juhtkang Defender Game Racer Turbo
- Eelarve 300 EUR

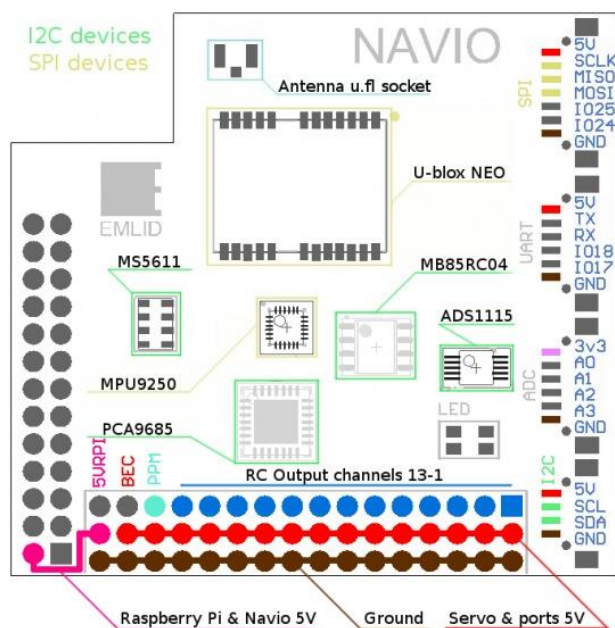
Minu alamülesanne on korraldada FPV vaatega juhtkangiga kaugjuhtimise süsteem.

2 Mehitamata lennuk

Mehitamata lennuki põhiidee koosneb sellest, et meie mikrokontroller loeb andmed erinevatest anduritest, töötleb need AHRS algoritmi järgi ja väljastab sobivad parameetrid lennukisüsteemidele.

2.1 Mehitamata lennuki komponendid

Põhikomponendid on GPS, kõrguse andur õhurõhumõõduri baasil, güroskoop, kiirendusmõõdur, kompass magnetomeetri baasil. Kõik need komponendid on paigaldatud NAVIO kilbil (Joonis 1). Kiiruseandur ühendatakse kas analoog pordile või I2C pordile[2].



Joonis 1. Navio kilp[2].

Kiirendusmõõdur, kompass ja güroskoop on ühendatud ühes seadmes MPU9250, mis suhtleb Raspberry Pi-ga SPI siini kaudu. Kõrguse andur on MS5611, see aga suhtleb Raspberry Pi-ga I2C siini kaudu. GPS eest vastutab U-blox NEO moodul, mis suhtleb Raspberry Pi-ga SPI siini kaudu[2].

2.2 Raspberry Pi

Raspberry Pi on ühest trükkplaadist koosnev arvuti. Seda võib kasutada erinevate seadmete loomiseks, sest tarkvaral on avatud lähtekood[3] .

2.2.1 Raspberry Pi tarkvara ja riistvara

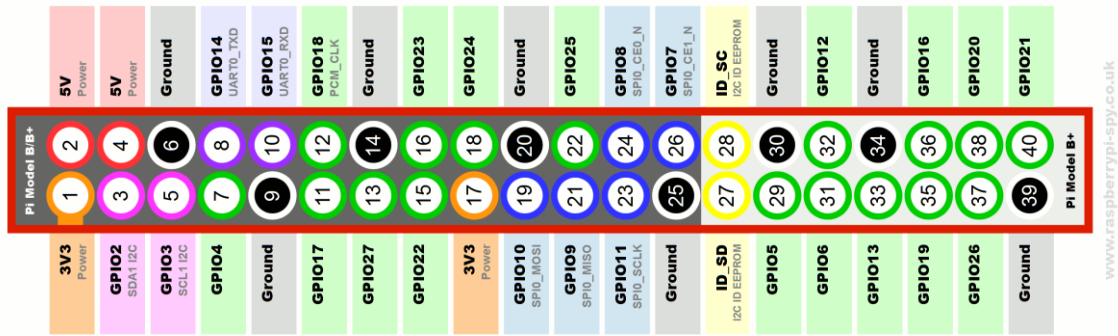
Meile on kättesaadavad kaks Raspberry Pi põlvkonna seadet. Esimene on Raspberry Pi B ja teine on Raspberry Pi 2B. Nende parameetrid on leitavad Tabel 1.

Tabel 1. Raspberry Pi parameetrite võrdlus.

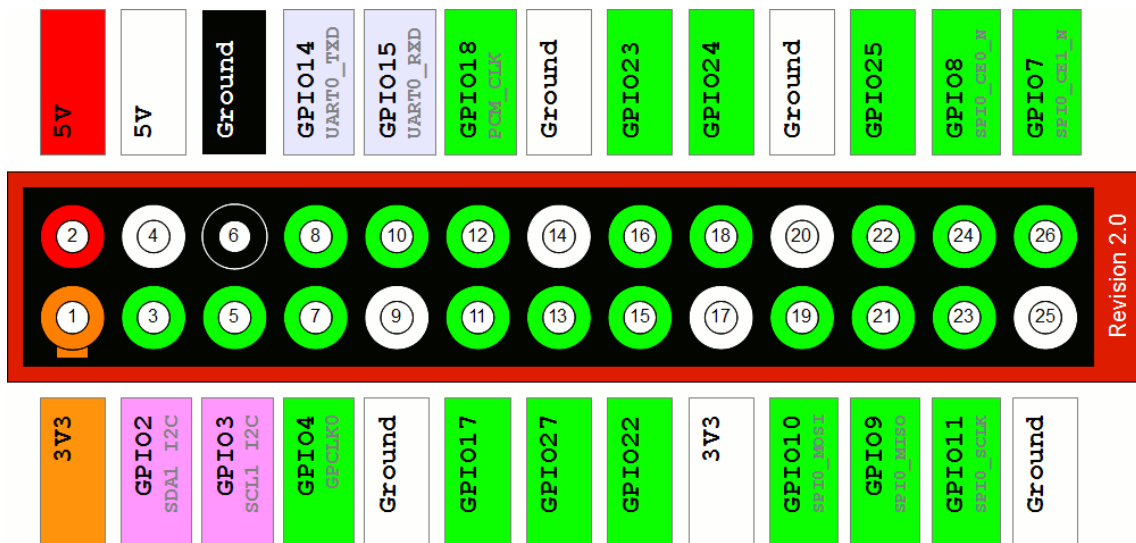
	Mudel B	Mudel 2B
Süsteemikiip	Broadcom BCM2835	Broadcom BCM2836
CPU	700 MHz ARM1176JZF-S	900 MHz ARM Cortex-A7
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 ja VC-1 (litsentsiga), 1080p30 h.264/MPEG-4 AVC kõrgprofili dekooder kooder	
Mälu (SDRAM)	512 MB	1 GB
USB 2.0 pesasid	2	4
Videoväljundid	HDMI, RCA komposiitvideo	HDMI, TRRS komposiitvideo
Heliväljundid	3,5 mm pistik, HDMI, I ² S	
Salvestusmaht	SD / MMC / SDIO kaardipesa	MicroSD
Integreeritud võrk	10/100 Mbit Ethernet	
Voolutarve	700 mA (3,5 W)	600 mA (3,0 W)
Vooluallikas	5V MicroUSB pistiku või GPIO pesa kaudu	
Kaal	45 g	45 g
Suurus	85,6 x 53,98 mm	

Komponentide võrdlusest selgub, et parim variant on Raspberry Pi 2B, aga GPIO siin on pikem ja tuleb kontrollida, kas see sobib kokku Navio kilbiga[3] .

Raspberry Pi 2B-l on 40 pin pikkune GPIO siin(Joonis 2)[4] , Raspberry Pi B on 26 pin pikkusega(Joonis 3)[4] . Võrdlusest saime aru, et Raspberry Pi 2B esimesed 26 pin on sama tähendusega nagu Raspberry Pi B -l, seetõttu on meie valik on Raspberry Pi 2B.



Joonis 2. Raspberry Pi 2B GPIO[4]

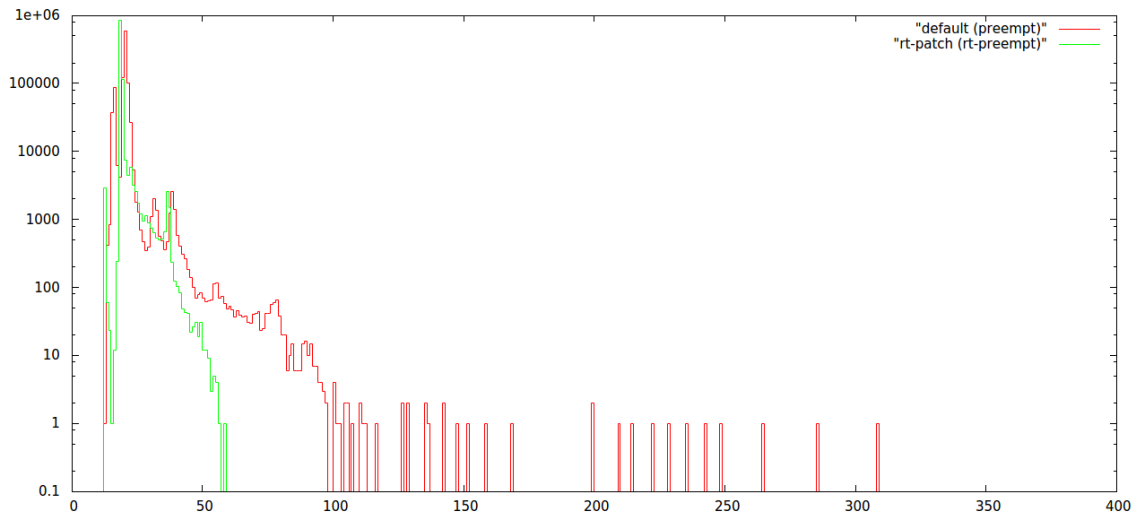


Joonis 3. Raspberry Pi B GPIO[4]

Tavaline Raspberry Pi tarkvara põhineb Linuxi tuumal töötaval operatsioonisüsteemil. Kõige levinum nendest on Raspbian[2].

Meie jaoks selline süsteem eriti ei sobi, seetõttu on meil vaja kasutada Emlid tehtud süsteemi. Me ei saa kasutada Raspbian, sest seal tekkiv viide on mitu korda suurem(Joonis 4) ja võib segada korrektset autopiloodi tööd.

Emlid operatsioonisüsteem on samuti tehtud Raspbian baasil, aga sellele on lisatud reaaja lapp ja tarkvara Navio kilbiga koostöök[5].



Joonis 4. Tekkitav viide rakendusega tööl, roheline joon on Emlid Raspbian, punane joon tavaline Raspbian[5] .

Operatsioonisüsteemi kasutuseks on vaja kujundus paigaldada MicroSD mälukaartile.

2.2.2 Ühenduse loomine

Pärast mälukaardi sisestamisest ja Raspberry Pi sisse lülitamist on meil kaks võimalust nägemaks, mis sees toimub:

- HDMI kaudu monitori ühendamine
- SSH protokolliga kaudu Etherneti vahendusel ühendamine

Me kasutasime teist ühendust rohkem, sest see võtab vähem aega ja tarvitse monitori otsida.

SSH on võrguprotokoll, mis võimaldab andmeid saata üle turvalise kanali. Tavaliselt kasutatakse seda võrguarvutisse sisselogimiseks ja võrgu sees tegutsemiseks. Pärast ühenduse loomist ilmub ekraanil käsurida. SSH kaudu võib ka faili edastada. SSH serveritega ühendumiseks kasutatakse TCP port 22[6] –[8] .

SSH ühendamiseks on Raspberry Pi-l SSH server. Kliendiks kasutasime Bitvise SSH tarkvarat.

Ühenduse loomiseks on meil vaja teada Raspberry Pi IP lokaalvõrgu aadressi. Seda võib avaldada kasutades Nmap Zenmap GUI rakendust[9] .

Kõigepealt leiame meie arvuti IP aadressi lokaalvõrgus kasutades selleks Windows konsooli käsku:

```
ipconfig
```

Nüüd käivitame Nmap ja seal käsureal sisestame käsu:[9]

```
nmap -sn 192.168.37.*
```

kus 192.168.37 on meie arvuti IP-lt osa.

Kõik andmed SSH ühenduse loomiseks on nüüd teada.

2.2.3 Autopiloodi tarkvara installeerimine ja käivitamine

Pärast ühendust loomist läbi lokaalvõrgu avaneb käsurea aken.

Autopiloodi .deb pakendi laadimiseks kasutame käsku:[10]

```
wget 'https://files.emlid.com/apm/apm.deb' -O apm.deb
```

wget on rakendus, mille abil saab allalaadida faili lingi järgi, tema parameeter „-O failinimi“ tähendab, et fail salvestatakse samasse kataloogi kirjutatud nimega[11] .

Faili „.deb“ on Debian tarkvara pakendi laiendus ja binaarse pakendi nimi.[12]

Autopiloodi installeerimiseks kasutame käsku:

```
sudo dpkg -i apm.deb
```

kus „sudo“ käsk on õiguse andmine ja „dpkg“ on tarkvara „.deb“ pakendiga tööks ja „-i“ parameeter on installeerimine[13] .

Nüüd ArduPilot autopiloot on installeeritud. Käivitamiseks on vaja sisestada käsk:

```
sudo ArduPlane -A udp:192.168.1.2:14550
```


kus ArduPlane on meie kere valik, on olemas ka teised keretüübid aga neid me ei kasuta. Parameetri „-A“ asemel võib olla veel „-B, -C, -D“. „-A“ on primaarse telemeetri jaoks, „-B“ on vaja siis kui me kasutame välist GPS-i, „-C“ on sekundaarse telemeetria jaoks, „-D“ kui me kasutame kaht välist GPS-i. „UDP“ või „TCP“ parameeter vastutab ühenduse tüübi eest. UDP ühendusel me saadame telemeetria andmed GCS IP-le ja TCP ühendusel Raspberry Pi-l luuakse TCP server, kuhu võib GCS ühendada[10].

2.3 3G USB Modem

2.3.1 3G võrgu eelistused ja puudused

Tavaliselt toimub lennuki juhtimine läbi RC. Kuid sellel juhul on suur puudus, juhtimise kaugus on piiratud ja kaugusega kvaliteet kahaneb. Meie 3G võrgu valik põhineb sellel, et leviala on Eestis piisavalt suur ja mastid paigaldataks nii, et praktiliselt puuduvad nõrgasignaali alad. Kõik see võimaldab meil kontrollida lennukit piiramatult kauguselt. 3G võrgu läbilaskevõime on piisavalt suur selleks, et telemeetria edastada, aga kui me hakkame ka videot edastada siis viide kasvab. Parem variant on 4G võrk, aga seadmete hind on liiga kõrge[15].

2.3.2 Modemi sisse lülitamine

Meie lennuk kasutab ühenduse loomiseks 3G modemi Huawei E182E. Modemile on paigaldatud USB-mälupulk. Raspberry Pi 2 tarkvara ei oska töötada nendega koos ja selle jaoks on vaja töörežiimi muuta. Me muudame töörežiimi kasutades tarkvara USB_modeswitch. Iga kord kui me ühendame USB pordi, meie modem käivitab Windows draiveri, mis eristab meie modemi mälupulgast. Tavaliselt kui see oleks Windows OS installeeritakse modemi draiver ja ta automaatselt vahetab töörežiimi[16].

Seoses et me kasutame Raspbian baasil OS, kus tavaliselt ei kasuta GUI ja Windows draiverid ei sobi. Kuna draiverid ei sobi, vaja käsitsi või mingi rakendusega valida Raspbian draiverid, mis sobivad meie modemi jaoks. Meie USB modemi vaikimisi töörežiim on mälupulk. Seda on meil vaja iga kord muuta.

See on võimalik lahendada, kui kasutame USB_modeswitch, aga ta töötab ainult iga kord, kui me eemaldame ja taasühendame modemi. Meil on vaja imiteerida modemi

USB pordist eemaldamist ilma füüsiliselt seda eemaldamata. Kõigepealt on vaja teada saada, milline draiveri number on meie mälupulgal.[17]

```
dmesg
```

Saanud numbrit kasutame, et lahutada meie mälupulk loogiliselt ja pärast paigaldada tagasi[18] .

```
echo -n '1-1.4' > /sys/bus/usb/drivers/usb/unbind  
echo -n '1-1.4' > /sys/bus/usb/drivers/usb/bind
```

Nüüd töötab meie mälupulk 3G USB modemina.

2.3.3 Wvdial

Võrguga ühendamiseks on vaja tarkvara, mis saadab päringuid operaatorile ja seadistab saanud andmete järgi modemi. Üks nendest rakendustest on wvdial. Pärast selle installerimist, on meil vaja luua konfiguratsiooni fail ja paigaldada see /etc/ kataloogi. Fail sisaldab andmeid meie APN kohta, päringuid ühenduse loomiseks, baasandmed modemi kohta ja ühenduse usaldusväärseuse parameetrid. Meie konfiguratsiooni faili näide on toodud (Joonis 5)[19] .

```
Bitvise xterm - raspberry.bscp - pi@192.168.37.27:22 - pi@navio-rpi: ~
GNU nano 2.2.6 File: /etc/wvdial.conf
[Dialer Defaults]
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init3 = AT+CGDCONT=1,"IP", "internet.emt.ee"
Stupid Mode = 1
MessageEndPoint = "0x01"
Modem Type = Analog Modem
Baud = 460800
Modem = /dev/ttyUSB0
ISDN = 0
Phone = *99#
Password = { }
Username = { }
Auto_DNS = on
Auto Reconnect = on
[ Read 16 lines (Warning: No write permission) ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Joonis 5. Wvdial.conf

Wvdial on vaja käivitada taustal. Selle saavutamiseks on vaja sisestada järgmised käsud, mis annavad õigused failidele[20] .

```
chmod +s /usr/bin/wvdial
```

```
chmod +x /usr/sbin/pppd
```

```
chmod +s /usr/sbin/pppd
```

```
chmod 0755 /etc/ppp/peers/wvdial
```

```
chmod +s /etc/ppp/peers/wvdial
```

2.4 Andmeedastus

Telemeetria edastus toimub läbi UPD protokoll. Me saadame andmed GCS IP-le, kus juba andmeid töödeldakse.

Telemeetria andmed pakitakse MAVlinki.

2.4.1 MAVlink

MAVlink on ühenduse protokoll. Läbi selle ühendub meie GCS lennukiga ja lennuk GCS.

MAVlinki sõnum on baitide voog, mis on kodeeritud GCS ja saadetud APM-le. Neid edastatakse läbi TCP või UDP protokoll, ning kas USB või telemeetria seadme kaudu. Kodeerimine siin tähendab seda, et me panime paketid konkreetsele andmete struktuurile ja lisasime veel paar veaparanduse baiti.

Iga MAVlingi pakett on 17 baiti pikkusega ja koosneb: 6 baiti päis, 9 baiti kasulikku koormuse ja 2 kontrollsumma baiti.

Päis koosneb: sõnumi päis, sõnumi pikkus, jada number, süsteemi ID, komponent ID ja sõnumi ID.

Oluline tarkvara ülesanne on kontrollida kas saadetud ja vastuvõetud sõnumid on samad ja ei ole rikunud, kui midagi on valesti, siis sõnum kõrvaldatakse. Andmeedastuse töökindluseks on vaja kiirust piirata, sest suurem kiirus tähendab, rohkem vigu. Just sellepärast toimub andmevahetus 57600 bps kiirusel.

Igal päise osal on oma ülesanne. Süsteemi ID on allika ID kust tulevad andmed, APM tarkvara kontrollib iga kord, kas sõnum on tema jaoks. Komponenti ID on allsüsteemi ID, mis asub peasüsteemi sees. Sõnumi ID kirjeldab, kust tuleb sõnum. Kasulik koormus ongi see informatsioon, mida meil oli vaja.

MAVlinki lahti seletamine – kõigepealt meil on käepide sõnum mida saadame iga sekundi jooksul, selleks et aru saada kas oleme sünkroonitud. Sealt loetakse süsteemi ID ja komponentide ID. Igal süsteemil, mis kasutab MAVlinki on selliseid andmeid. Nüüd saab meie tarkvara välja võtta meie paketi ja suunab seda vaadates milliseid andmeid ta sisaldab sobiva kohale. Paketi sisu peab oleme sarnane, saadetud paketi sisule[21] .

2.4.2 Probleem 3G võrguga ja serveri loomine

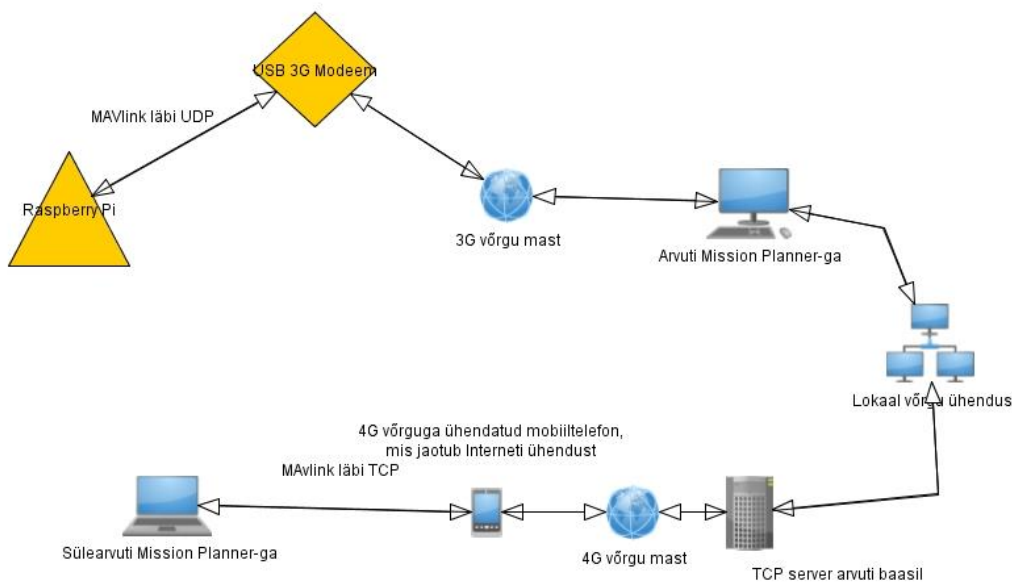
Meil oli vaja ühendada Raspberry Pi 2 GCS-ga. Selleks me kasutame 3G võrku, aga ühenduse loomise ajal selgus, et meil on võimatu otseselt andmeid edastada.

Esimene lahendus, mida me proovisime oli DynDns. Põhimõtte oli selline, et me sidustame meie Raspberry Pi 2 IP aadressi mingi domeeninimega. Installeerime DynDns kliendi, mis automaatselt peaks tunnistama IP ja vahetama IP, mis oli enne sidustanud DynDns-ga. Rakendus töötas korrektselt ja tegi oma tööd hästi, kuid siiski ei olnud võimalik ühendust luua[22].

Probleem oli selles, et operaatoril on paigaldatud NAT ja Firewall. Me ei ühenda otseselt, sest IP mille me oleme saanud kehtib ainult operaatori võrgus. Public IP on mingi seade IP mastil.

Lähemalt uurides leidsime teenuse Static IP, aga see oli kättesaadav vaid ärikliendile ja hinna tõttu selle kasutamine oleks ebaratsionaalne, sest meil oli vaja vähemalt kaht sellist. Üks Raspberry Pi-le ja teine meie GCS jaoks.

Mission Planneri uurimise ajal leidsime, et ta võib peegeldada MAVlink-i ehk edastab kõik andmed, mida võtab vastu TCP server sama IP aadressiga, mis oli GCS-l. See kergendas lahenduse otsimist. Nüüd oli meil vaja leida staatilise IP-ga võrk ja ühendada see meie arvutiga koos GCS installeeritud tarkvaraga. Sellise leidsime Jõhvis. Pärast kõige installerimist ja ühendamist nägi andmeliiklus välja nii(Joonis 6):



Joonis 6. Ühendus Raspberry Pi-ga läbi serveri

Meie Raspberry saadab andmed läbi operaatori võrguseadme GCS-le, mis asub Jõhvis. Andmeedastus toimub UDP protokolliga kaudu. Edasi me seame üles TCP serveri, kuhu peegeldakse kõik MAVlink-i andmed. Nüüd kuna me teame GCS IP ühendame selle

TCP protokoll kaudu ja võtame vastu kõik andmed meie GCS-le. Tagasi ühendus toimub samamoodi kuid vastupidises järjekorras.

2.5 Videoside loomine ja kaamera valik

Meil tekkis vajadus videoside loomiseks, sest 3G modemi kasutusel lennukite kaugus võib olla eriti suur ja maast ei ole see nähtav.

Valida oli meil kolm kaamerat erinevate ühenduse viisidega. Esimene nendest Raspberry Pi Camera Module, mis ühendatakse otseselt Raspberry Pi-l asuva Camera portile. Me ei saanud seda kaamerat kasutada, sest ta tekitab palju raadiosageduste müra, mis segab korrektse GPS-si tööd, nagu ütleb Emlid[23].

Teine kaamera oli lennukiga kaasa tulnud FPV kaamera komplekt. Videoedastus toimub sellel kaameral kasutades 5-6GHz raadiosagedusel töötavat saatjat ja vastuvõtjat. Eelisteks on see, et sellel kaameral puudub nähtav viide, aga sellega kõik eelised lõppevad. Puuduseks on piiratud kaugus, kus kaamera töötab, sellel saatjal on see 1.5 km[24]. Seda kaamerat ei saa ühendada arvutiga ja selle toitmiseks on vaja lisa akut. Viimasels video kvaliteet on palju madalam kui meie kolmanda kaamera korral(Tabel 2).

Kolmas kaamera on USB CNE-CWC3. Selle kaamera eelisteks on see on kvaliteet on palju parem, ei ole vaja lisa toidet (USB toide jätkub), lihtne paigaldada ja rohkem parameetreid on võimalik seadistada. Puuduseks on vaid Raspberry Pi võimsus, mis ei suuda täies mahus selle kaamera andmeid töödelda[25].

Tabel 2. Kaamera võrdlus

	FPV kaamera	CNE-CWC3 kaamera	Raspberry Pi HD Camera Board V2
Sensor	1/3 Sony CCD	1/4 CMOS	1/4 Sony CMOS
Resolutsioon	480TVL	1920x1080 30fps	1080p
Valgustundlikus	0,01 lux	-	-

2.5.1 RTP ja RTCP protokollid

RTP ja RTCP on transpordi taseme protokollid. Nad vastutavad real-ajas video edastamise eest.

Videoedastus toimub nii: video edastatakse allikast video kodeerimiseseadmele, kus seda kodeeritakse, edasi märgitakse kasuliku koormus, pärast lisatakse RTP päis ja nüüd edastatakse UDP kaudu kasutaja arvutini, paralleelselt edastatakse ka RTCP protokollide andmed, mida on vaja et sünkroniseerida ja video kaotuste eest jälgida.

Video töötlemine toimub peaaegu samamoodi, kõigepealt eemaldatakse RTP päis siis dekodeeritakse video ja kuvatakse ekraanile või edastatakse edasi.[26] .

2.5.2 VPN

Praegu edastatakse video serverile, mis asub Jõhvis, kuid meil on vaja seda näha meie arvutis. Selle jaoks me loome VPN ühenduse.

VPN on privaatne ja turvaline arvutivõrk, mille loomiseks kasutatakse avalikku telekommunikatsiooni infrastruktuuri[27] .

Tavaliselt kasutatakse VPN võrku turvalisuse tõttu. VPN võrgus toimub andmeedastus läbi Interneti kodeeritud viisil[28] .

Meie jaoks pole turvalisus esimesel kohal, aga me kasutame teist VPN tunnusoont. VPN ühenduse loomisel me ühendame kaks arvutit ühe lokaalvõrguga, kuna meie arvuti ei ole kättesaadav väljaspool Interneti ja me ei saa edastada videot, siis kui on loodud lokaalvõrk ilmub selline võimalus. Me võime edastada videot lokaal IP aadressile.

2.5.3 Gstreamer – video edastus ja saamine

Gstreamer on multimeedia andmete töötlemise tugiraamistik. Gstreamer mitmest rakendusest, mida võib kasutada video ja audio salvestamiseks, edastamiseks, töötlemiseks[29] .

Me kasutame seda, et videot määratud parameetriga kinni püüda ja edastada see lõppkasutajani. Video edastamiseks me kasutame skripti mis käivitatakse Raspberry Pi-l(Joonis 7).

```

Bitwise xterm - raspberry.bscp - pi@192.168.37.27:22 - pi@navio-rpi: ~
GNU nano 2.2.6 File: gstreamer.sh Modified
#!/bin/bash 320x240 360x288 640x480 704x576
WIDTH=320
HEIGHT=240
ip=213.35.152.34
v4l2-ctl --set-fmt-video-width=$WIDTH,height=$HEIGHT,pixelformat=1
v4l2-ctl -i 0
gst-launch-1.0 -v \
rtplib name=rtplib v4l2src device=/dev/video0 \
! video/x-raw,width=$WIDTH,height=$HEIGHT,framerate=33/1 \
! omxh264enc target-bitrate=1000000 control-rate=1 \
! "video/x-h264,profile=high" \
! h264parse \
! queue max-size-bytes=1000000 \
! rtph264pay pt=96 config-interval=1 \
! rtplib.send_rtp_sink_0 rtplib.send_rtp_src_0 \
! udpsink port=1234 host=$ip ts-offset=0 name=vrtpsink \
#rtplib.send_rtcp_src_0 \
#! udpsink port=1235 host=$ip sync=false async=false name=vrtcpsink udpsrc port=5000 name=vrtpsrc \
#! rtplib.recv_rtcp_sink_0
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify     ^W Where Is    ^V Next Page    ^U UnCut Text  ^T To Spell

```

Joonis 7. Video edastamiseks skript

Video saamiseks serveril ja ümbersuunamiseks kasutame järgmise käsku[30] :

```

gst-launch-1.0 -v rtplib name=rtplib latency=200 udpsrc caps=application/x-
rtplib,media=video,clock-rate=90000,encoding-name=H264 port=1234 !
rtplib.recv_rtp_sink_0 rtplib. ! rtph264depay ! avdec_h264 ! queue ! x264enc
tune=zerolatency byte-stream=true bitrate=300 ! rtph264pay ! udpsink
host=169.254.227.191 port=1235 ts=offset=0

```

2.6 Hilistumiseaeg

Saadud lennukil peame kontrollima, kui kiiresti ta reageerib käskudele juhtkangist. Hilistumiseaja mõõtmiseks me kasutame arvutis asuvat mikrofoni(täpsemalt vaata Lisa 1) ja „Iphone 6“ kaamerat, mis võimaldab pildistada videot 60 kaadrit sekundis. Mõõtetulemuste täpsustamiseks võtame arvesse ka heli levikiiruse. Reaalse aja saamiseks on järgmine valem:

$$t_{\text{heli levimise aeg}} = S/V$$

$$t_{\text{reaalne}} = t_{\text{mõõdetud}} - t_{\text{heli levimise aeg}}$$

$V=343$ m/s

$S=1,6$ m

$t_{levimise\ aeg}=0,0047$ s

Tabel 3. Juhtimis hilistumisaeg, kaamera on sisse lülitatud ja edastab videot

Proovi number	Mõõdetud, s	Reaalne, s
1	0,197	0,1923
2	0,279	0,2743
3	0,418	0,4133
4	0,337	0,4086
5	0,313	0,3083
6	0,418	0,4133
7	0,348	0,3433
8	0,290	0,2853
9	0,209	0,2043
10	0,302	0,2973
Keskmine Hilistumisaeg, s		0,31403

Tabel 4. Juhtimis hilistumisaeg kaamera on välja lülitatud ja ei edasta videot

Proovi number	Mõõdetud, s	Reaalne, s
1	0,290	0,2853
2	0,255	0,2503
3	0,163	0,1583
4	0,255	0,2503
5	0,290	0,2853
6	0,232	0,2273
7	0,174	0,1693
8	0,232	0,2273
9	0,232	0,2273
10	0,290	0,2853
Keskmine hilistumisaeg, s		0,2366

On hästi nähtav, et ilma kaamerata on hilistumiseaeg 100 ms vähem. Nüüd mõõdame hilistumiseaega video edastamisel.

Tabel 5. Interneti kaabliga video edastuse hilistumiseaeg, CNE-CWC3 kaamera

Proovi number	Hilistumiseaeg, s
1	0,43
2	0,44
3	0,5
4	0,34
5	0,5
Keskmine hilistumiseaeg, s	0,442

Keskmine hilinemine on 442 ms, mis on tegelikult väike ja praktiliselt mitte märgatav ja peamine põhjus on video töötlemine mitte edastamine. Meie rakenduses on see päris hea, kui me oleksime juhtinud otseselt läbi RC puldi siis oleks see hilistumine kriitiline.

Tabel 6. 3G võrgu kaudu video edastuse hilistumiseaeg, suur ekraan, CNE-CWC3 kaamera

Proovi number	Hilistumiseaeg, s
1	0,63
2	1,34
3	1,26
4	0,97
5	0,47
Keskmine hilistumiseaeg, s	0,936

Suure ekraani kasutusel on vaja rohkem ressursi video töötlemiseks.

Tabel 7 . 3G võrgu kaudu video edastuse hilistumiseaeg, väike ekraan, CNE-CWC3 kaamera

Proovi number	Hilistumiseaeg, s
1	0,87
2	0,64
3	0,84
4	1,24
5	0,43
Keskmine hilistumiseaeg, s	0,804

Nüüd kui võrdleme läbi 3G võrgu, hiline mine kasvab praktilisel kaks korda ja on nähtav sõltuvus ekraani suurusest ehk arvuti võimsus omab samuti mõju. See toimub nii, sest videot on vaja dekodeerida.

2.7 Automaatne käivitamine ja juhtkangi seadistamine

Mugavaks kasutamiseks on vaja, et autopiloot ja kõik teised rakendused käivitataks automaatselt Raspberry Pi sisse lülitamisega. Realiseeritakse see skripti abil.

Automaatseks käivitamiseks lisame kõik meie käsud ühele skriptile ja lisame faili asukoha failile /etc/rc.local. Nüüd käivitub kõik automaatselt. USB Modemi lahutamise ja tagasi paigaldamisega on vaja ooteajale lisada vähemalt 10 sekundit, sest Raspberry Pi ei saa modemi draiverit momentaalset tagasi paigaldada.

Juhtkangi kasutamine vajab kalibreerimist, kui ühendame selle Mission Planneri-ga seadistame kõik vajalikud nupud[32] . Juhtkangi kasutamine toimub läbi 3G võrgu. Lennu jaoks, et oleks võimalik „relvastada“ ehk teha nii, et lennuk reageeriks mootori pöörämiskiiruse muutusele on vaja maha võtta kriipstähis radiokontrolleri tõrkekindlast parameetrist ja seadistada seda, et ta reageeriks ühenduse kaotusele GCS-ga

3 Kokkuvõte

Töö põhieesmärk oli mehitamata lennuki kokkupanek. Eesmärk on saavutanud ja me oleme saanud täielikult töötava lennuki.

Ühendus toimub läbi 3G modemi ja serveri kaudu asukohaga Jõhvis. Kaamera on ühendatud USB kaudu ja videot edastatakse 3G võrgu kaudu. Lennuk on juhitav kas juhtkangiga või lihtsalt autopiloodi käskudega.

Töö tulemus on minu teadmised erinevatest anduritest, ühendus meetodid ja autopiloodi süsteemi laiendus. Mina olen saanud koostöö kogemusi. Täiendasin oma oskusi lennuki ehitamisest ja suudan nüüd täielikult töötava autopiloodiga lennuki ehitada.

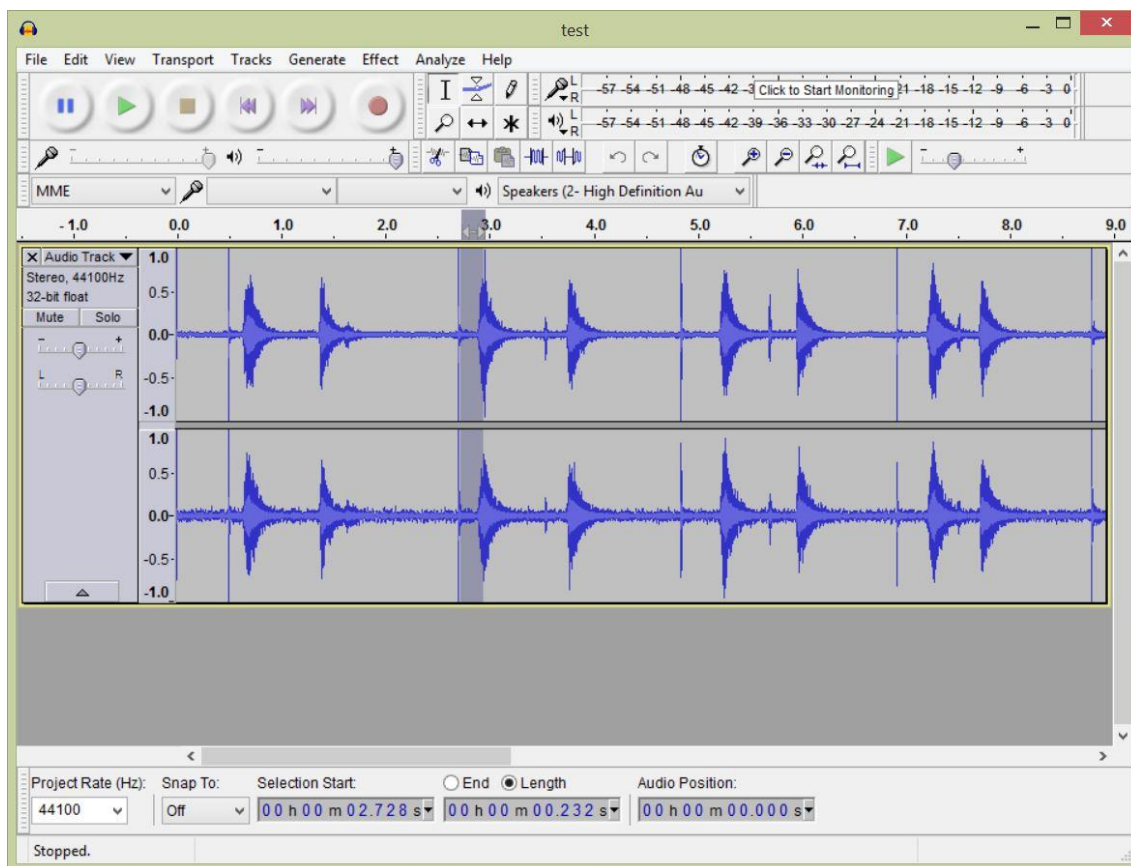
Sain rakendada oma oskusi omandatud õppimise ajal, nagu näiteks mikrokontrolleri programmeerimine, teadmised erinevatest ühenditest, jootmine ja skeemi mõistmine.

Kasutatud kirjandus

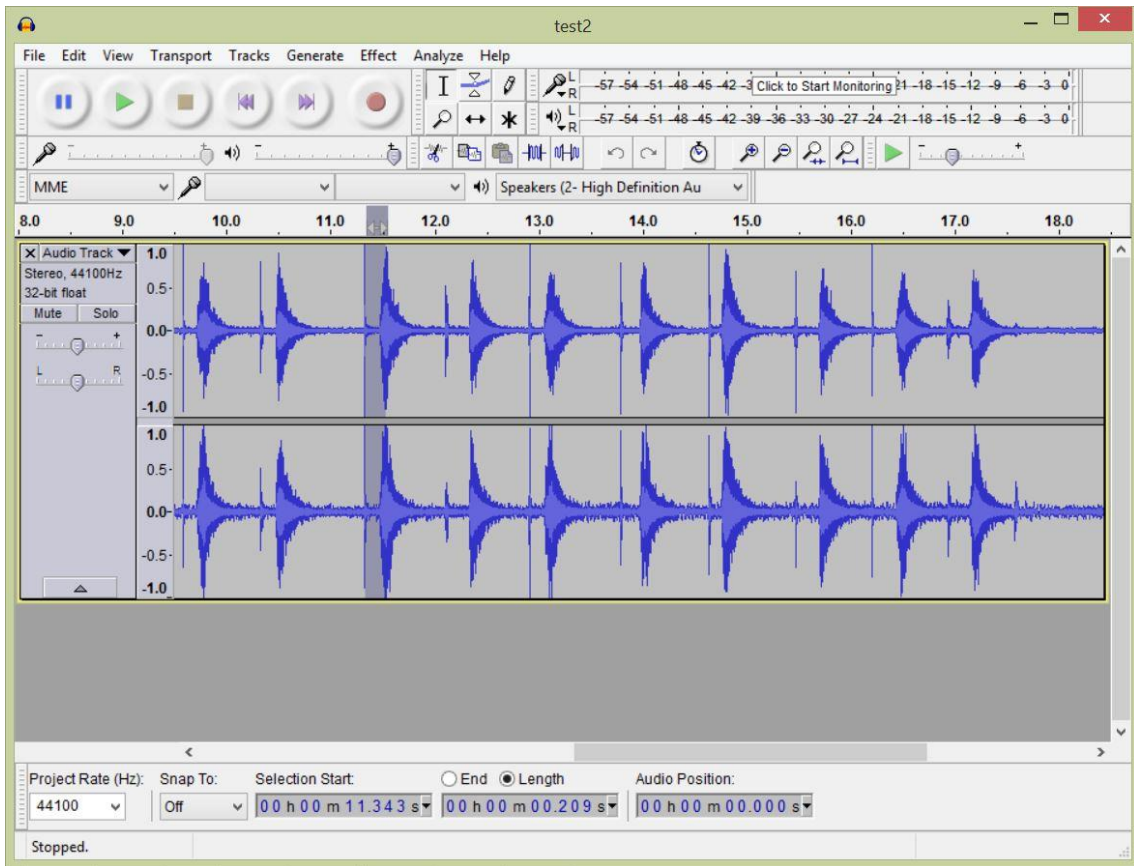
- [1] Dalamagkidis, K., Valavanis, K. P., Piegl, L. A. On Integrating Unmanned Aircraft Systems into the National Aerspace System. 2nd Edition. Springer. 2012. lk 1–2. (30.05.2016)
- [2] Navio hardware setup [WWW]
<https://docs.emlid.com/navio/Navio-APM/hardware-setup-navio/> (30.05.2016)
- [3] Raspberry Pi [WWW]
https://et.wikipedia.org/wiki/Raspberry_Pi (30.05.2016)
- [4] Raspberry Pi GPIO [WWW]
<http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/#prettyPhoto> (30.05.2016)
- [5] Raspbian ja Emlid Raspbian [WWW]
<https://docs.emlid.com/navio/Downloads/Real-time-Linux-RPi2> (30.05.2016)
- [6] SSH protokoll [WWW]
<https://tools.ietf.org/html/rfc4252> (30.05.2016)
- [7] SSH protokoll [WWW]
<http://www.serverwatch.com/news/print.php/3551081> (30.05.2016)
- [8] SSH port [WWW]
<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml> (30.05.2016)
- [9] Raspberry Pi Setup [WWW]
<https://docs.emlid.com/navio/Navio-APM/configuring-raspberry-pi/> (30.05.2016)
- [10] Ardupilot setup and run [WWW]
<https://docs.emlid.com/navio/Navio-APM/installation-and-running/> (30.05.2016)
- [11] Wget [WWW]
<https://en.wikipedia.org/wiki/Wget> (30.05.2016)
- [12] Deb file format [WWW]
[https://en.wikipedia.org/wiki/Deb_\(file_format\)](https://en.wikipedia.org/wiki/Deb_(file_format)) (30.05.2016)
- [13] Dpkg käsk [WWW]
<https://en.wikipedia.org/wiki/Dpkg> (30.05.2016)
- [14] Sudo käsk [WWW]
<https://en.wikipedia.org/wiki/Sudo> (30.05.2016)
- [15] 3G ja 4G leviala ja kiirus [WWW]
<https://www.telia.ee/era/internet/nutiseadmeh/leviala> (30.05.2016)
- [16] USB modeswitch [WWW]
http://www.draisberghof.de/usb_modeswitch/ (30.05.2016)
- [17] Dmesg käsk [WWW]
<http://www.linfo.org/dmesg.html> (30.05.2016)

- [18] Usb eemaldamise imiteerimine [WWW]
<https://www.linux.com/blog/disable-enable-usb-terminal> (30.05.2016)
- [19] Wvdial [WWW]
<https://github.com/wlach/wvdial> (30.05.2016)
- [20] Autorun Wvdial [WWW]
<http://mandrivausers.org/index.php?/topic/3300-creating-a-launcher-of-wvdial-to-run-in-the-background/> (30.05.2016)
- [21] MAVlink [WWW]
http://dev.ardupilot.com/wp-content/uploads/sites/6/2015/05/MAVLINK_FOR_DUMMIESPart1_v.1.1.pdf
(30.05.2016)
- [22] DynDNS [WWW]
<http://www.noip.com> (30.05.2016)
- [23] Video stream [WWW]
<https://docs.emlid.com/navio/Navio-APM/video-streaming/> (30.05.2016)
- [24] FPV kaamera [WWW]
http://www.hobbyking.com/hobbyking/store/__65997__Skyzone_Plug_N_Play_%3Cspan%20style= (30.05.2016)
- [25] Canyon kaamera [WWW]
<http://canyon.ru/product/cne-cwc3-2/> (30.05.2016)
- [26] RTP ja RTCP protokollid [WWW]
<https://www.netlab.tkk.fi/opetus/s38130/k99/presentations/4.pdf> (30.05.2016)
- [27] VPN [WWW]
https://et.wikipedia.org/wiki/Virtuaalne_privaatv%C3%B5rk (30.05.2016)
- [28] VPN põhimõtte [WWW]
<http://www.dcc.ttu.ee/ReinP/Arvutiv%C3%B5rgud%20Chapter8.pdf> (30.05.2016)
- [29] Gstreamer [WWW]
<https://gstreamer.freedesktop.org/download/> (30.05.2016)
- [30] Gstreamer video edastamine [WWW]
<https://cgit.freedesktop.org/gstreamer/gst-plugins-good/tree/tests/examples/rtp/server-v4l2-H264-alsasrc-PCMA.sh> (30.05.2016)
- [31] Gstreamer video saamine [WWW]
<https://cgit.freedesktop.org/gstreamer/gst-plugins-good/tree/tests/examples/rtp/client-H264-PCMA.sh> (30.05.2016)
- [32] Joystick setup [WWW]
<http://ardupilot.org/copter/docs/flying-with-a-joystickgamepad-instead-of-rc-controller.html> (30.05.2016)

Lisa 1 – Tabelite andmete saamine



Joonis 8. Kaamera viide



Joonis 9. Ilma kaamerata viide