

TFE4850 - EiT - Student satellite

Groundstation network

Group 2

Eirik Skjeggestad Dale
Hanne Thorshaug Andresen
Marius Ekerholt
Børge Irgens
Leif-Einar H. Pettersen
Hallstein Skjølsvik

April 30, 2013

Abstract

Today the NTNU Test Satellite is using amateur radio and the transfer rate will therefore be very limited, of only 9600 bps. Combined with the limited access time to the satellite this means a very limited 600 kB of data can be downloaded per day. Since the antenna limits the transfer rate, a network of ground stations can be utilized to increase the transfer rate without having to changing the antenna.

The aim of this project was to consider some possible networks, examine whether such a network would be beneficial, and set up a ground station network at the student ground station at Gløshaugen.

Different ground station network are today available, but a network is dependent on other ground stations as participants to be beneficial. Carpcomm Space Network is an existing network which is already connected, and therefore seemingly the best choice.

To connect the ground station at Gløshaugen to the Carpcomm Space Network a Raspberry Pi were used to run the required software. During our work we discovered that it lacks support for many kinds of radios and the rotor control functionality is still experimental. In addition it cannot transmit to the satellite, and cannot be used to initiate downloads.

Results from our simulations show that even a small network with either Norwegian/Scandinavian universities will result in a significant gain in download capacity. If the discovered challenges can be overcome the improvements can be substantial. Another factor that must be taken into considerations when planning a network of ground stations is the position of the different nodes. The efficiency of a network of ground stations is reduced when the communication ranges overlap. Gløshaugen ground station should have the nodes geographically far apart to achieve a sufficient network. In this case the ground stations must be more than 1600 km apart to have maximum efficiency. A network of ground stations demands more power as the communication time increase. Calculations on the power supply were performed to verify that the satellite would have sufficient power to support this increase in demand.

However, further work must be carried out so that the Carpcomm Space Network can be fully exploited, but if this is done it will bring great benefits. Increased amount of data will lead to greater expertise in satellite research, which is useful for the society.

Acknowledgements

We would like to thank the NUTS project leader Roger Birkeland and the "NUTS studentsatellitt" EiT- group for many good ideas and advice. Many thanks also to Knut Magnus Kvamtrø for help with connecting to the ground station.

Contents

1	Introduction	3
1.1	The Project group	3
1.2	Network	4
1.3	NUTS	4
2	Theory	5
2.1	Communicating with the satellite	5
2.1.1	Other ground stations	7
2.2	Network technology for ground stations	8
2.2.1	Genso	8
2.2.2	Carpcomm Space Network	8
2.2.3	PYXIS	9
2.3	Raspberry pi	10
2.4	Solar Power	12
3	The project	15
3.1	Ground station networks	15
3.2	The project	16
3.3	Software	16
3.4	Further Work	18
3.5	Conclusion	19

List of Figures

2.1	LOS for 450 og 650	6
2.2	Ground station satellite geometry	6
2.3	ntnu footprint	7
2.4	world	7
2.5	Illustration of the GENSO network.	9
2.6	Raspberry pi highlevel	11
2.7	Schematics for the RS232-converter	11
3.1	Some possible ground networks	16

Chapter 1

Introduction

This chapter will contain a short introduction to our project. That means what we're doing and what people are working on the project. It will also be discussed some short background on why we're doing this project.

1.1 The Project group

The project group consisted of 6 people from 5 different institutes, so we have competency in a variety of fields. The group members can be seen in Table 1.1.

Name	Background
Marius Ekerholt	Computer technology
Eirik Skjeggestad Dale	Computer technology
Hanne Thorshaug Andresen	Energy and Environmental technology
Leif-Einar H. Pettersen	Electronics and Telecommunications
Børge Irgens	Theoretical Physics
Hallstein Skjølsvik	Electronics and digital design

Table 1.1: Group members

1.2 Network

The problem today is that the transfer rate for a satellite using amateur radio is low and the the number of passes are limited. We wanted to look more into this problem and find a solution. Since the transfer rate is limited by the antenna its hard to increase the transfer rate without changing the antenna itself. We therefore chose to dive more into how we could increase the number of downstream per pas. We looked into how we could utilize a network of ground stations. We needed to find an existing network that had other ground stations already connected. Carpcomm had seemingly a well established network and was one of the main reason for why we chose it.

1.3 NUTS

NUTS is a student driven satellite project at NTNU. The goal of the project is to design and build a double Cubesat. The student satellite project is organized under the Department of Electronics and Telecommunications.

Chapter 2

Theory

Before we started our project, we did quite a lot of prestudy, covering which technology was available, to determine which fit our project best. This chapter will contain a summary of the different technologies, and also some theory behind the reasoning for having multiple ground stations listen to our satellite.

2.1 Communicating with the satellite

Since we don't know when the satellite will be launched we don't know the exact orbit. The project manager, Roger Birkeland, told us that we can assume the orbit will have a height above the Earth somewhere between 450km and 650km and inclination of 98 degrees.

The height above the earth dramatically changes the time the satellite is seen by the ground station, see Fig. 2.1. The signal will be weaker when the satellite is further away, it is therefore necessary to increase the minimum elevation angle ϵ , see Fig. 2.2, to maintain SNR. Previous work[7, 8] has calculated the minimum elevation angle, for a ground station here in Trondheim, for satellites with different altitudes and found that these effects cancel each other out. In the following we'll assume that the altitude is 500 km and the minimum elevation angle is 28 degrees.

The result of this is that the ground station can communicate with the satellite whenever the ground track is inside a rough circle centered on the ground station, see Fig. 2.3 for the estimated "range" of the ground station at Gløshaugen operating with these constraints. The efficiency of a network of ground stations is reduced when the ranges overlap, so to have a sufficient network the nodes must be geographically far apart. In this case

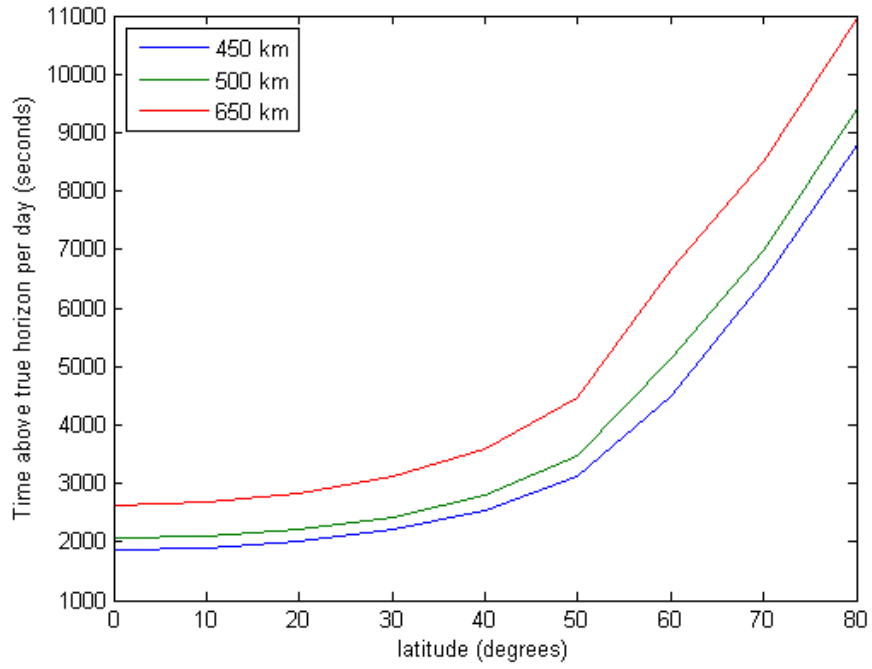


Figure 2.1: Time above the horizon for a satellite with altitude $h = 450km$, $h = 500km$ and $h = 650km$ as a function of ground station latitude

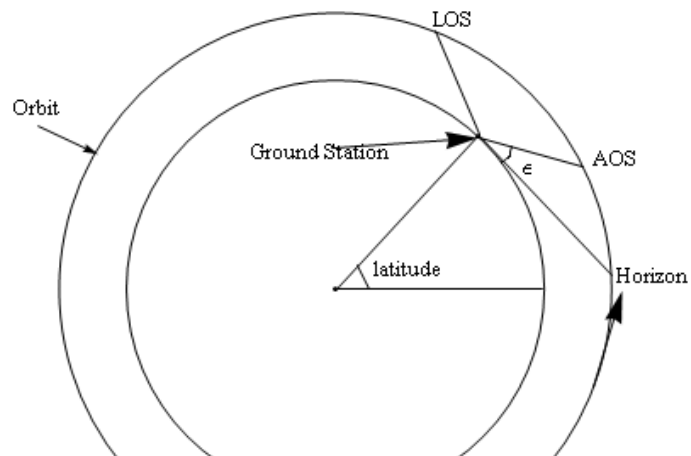


Figure 2.2: Illustration of geometry between a ground station and a satellite

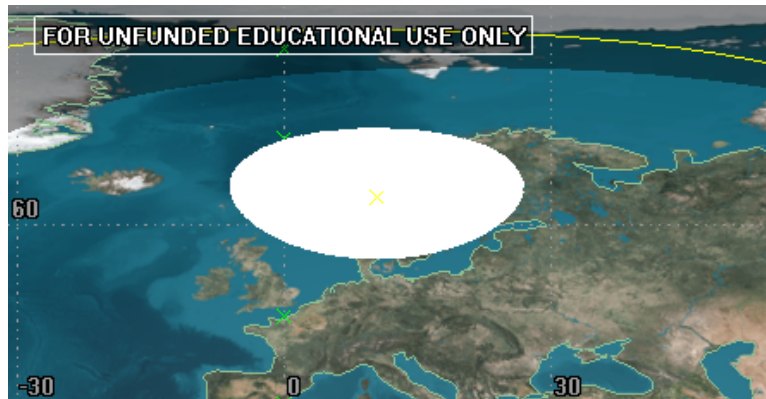


Figure 2.3: NTNU ground station range

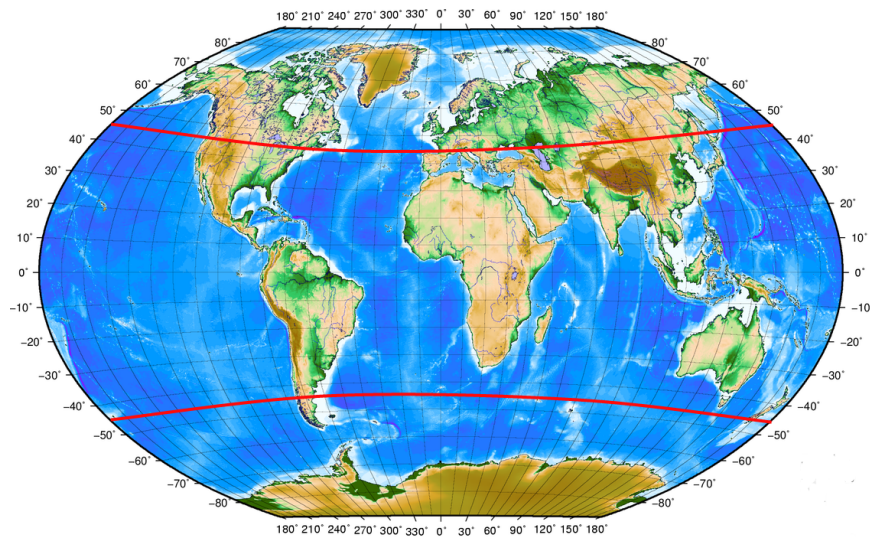


Figure 2.4: Map of the World

the ground stations must be more than 1600 km apart to have maximum efficiency.

The Gløshaugen ground station will on average be able to communicate with the satellite 520 seconds per day. With a bitrate of 9600 bps 600kB can be downloaded per day on average.

2.1.1 Other ground stations

Fig. 2.1 shows that the access time for a (near) polar satellite is almost latitude independent for ground stations at latitudes below 45 degrees. The access times of a ground station those low latitudes are about about 280

seconds/day, i.e. half the duration we get here in Trondheim. For higher latitudes the access time increases dramatically. And the average access time for a ground station in Longyearbyen (78N) is in fact as high as 1300 seconds/day.

2.2 Network technology for ground stations

When we decided to work on a network of ground stations, we first looked into four different ground station network technologies. We first hoped to work on a BlueBox, but this would require support from Aalborg that we couldn't get, as they were busy with a satellite launch of their own. Because of this, we decided to look into Carpcomm, that seemed more complete and doable than connecting to PYXIS with a BlueBox.

2.2.1 Genso

GENSO is an abbreviation for Global Educational Network for Satellite Operations. As the name suggest GENSO is a network where satellite operators across the world can utilize each others ground stations. GENSO provides a stable network by the use of several Authentication Servers (AUS), that will synchronize with each other. This will make the network resistant to single point failures. As user of GENSO a software applications is available. This is the Ground Station Server (GSS). A GSS application allows Mission Control Clients (MCC) in the GENSO network to connect to the ground station and download data from space crafts. If the space craft is able, and local laws permit, it is also possible to upload data[6]. Figure 2.5 illustrates this set up.

At first glance GENSO seems like a well organized ground station network which corresponds to the network we want to join. However it is difficult to find anything about the development of the project since September 2010. In addition the GENSO home page "GENSO.org" is not operational. Due to the difficulties with finding information of the current state of GENSO, we chose to look for other alternatives.

2.2.2 Carpcomm Space Network

Carpcomm is a private company that delivers a plug and play ground station [3] that costs \$700. The software for the ground station is open source and is provided precompiled for x86 and arm processors. CarpsD, the software from Carpcomm connects to an server where it can upload data from

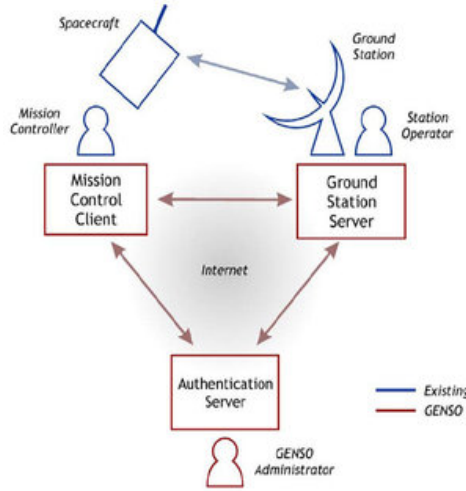


Figure 2.5: Illustration of the GENSO network.

satellites. From a web interface at this server the groundstation can also be controlled remotely, it can also be set up to control the groundstation automatically. The advantage of using this solution is that the network is actually functioning, though there are few other operational ground stations.

2.2.3 PYXIS

The BlueBox is part of a distributed ground station network called PYXIS, developed primarily for the AAUSAT3 by Aalborg University (2013 [1]). The PYXIS goal is to offer a robust and effective ground station network for satellite developers, and one of the key factors is that everyone is free to setup a ground station using the open source BlueBox hardware.

The PYXIS concept includes a backend server, BlueBox hardware and a Ground Station Server (GSS). The backend server runs an individual instance for each satellite utilizing the BlueBox, and is operated by the persons responsible for the ground station.

The BlueBox itself is hardware to receive and transmit signals from the satellites.

Control of the BlueBox and ground station mechanics is handled by the GSS, and both the BlueBox and the GSS is operated by the responsible for the Ground station.

Both the backend server and the GSS is already in place at each ground station, and to join the PYXIS network we would only have to make a

BlueBox, and test that it works.

2.3 Raspberry pi

Raspberry Pi is a small computer, with everything gathered in one board. In our project we will be using the B model, revision 2, which have a 700MHz ARM CPU, 512MB of RAM and a SC-card reader, in addition to the leads to connect to different devices, for the full overview, see figure 2.6. The recommended operating system is Raspbian, a linux distribution based on Debian.

The Raspberry pi was originally intended to help teach programming, but it can also perform many of the standard computer tasks, and it can be connected to a monitor or tv using an HDMI lead. In our project we hope to be able to use a Raspberry pi to run the software required to control the ground stations. The software provided by the Carpcomm project has Raspbian as one of its supported platforms, so we hope this will work well.

To control the movement of the antennas, a serial port is needed. Raspberry Pi has an serial port included in its gpio (general purpose input/output) connector. This serial port uses ttl-standard for its voltage levels, this is 0/3.3V while RS232 which is the standard used in computers uses (3V-15V)/-(3V-15V). Because of this an converter is needed. We chose to make an custom circuit board using the MAX3232 RS232 line driver. The circuit board is designed to be mounted on the gpio connector, because of small space in the case for the Raspberry Pi, the output is connected with cable to the external connector.

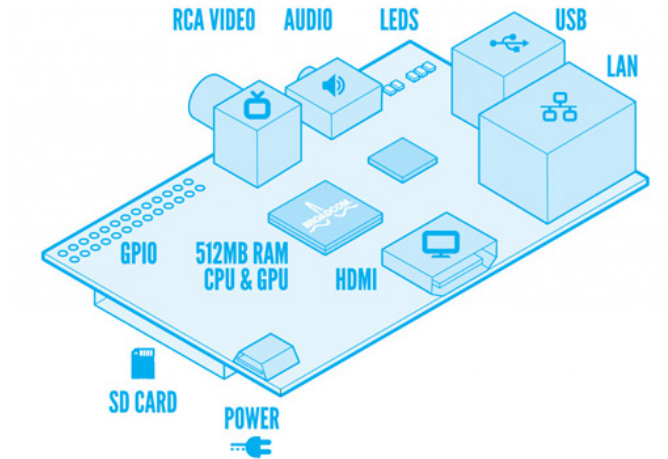


Figure 2.6: A highlevel schemantic of the Raspberry pi, model B rev 2

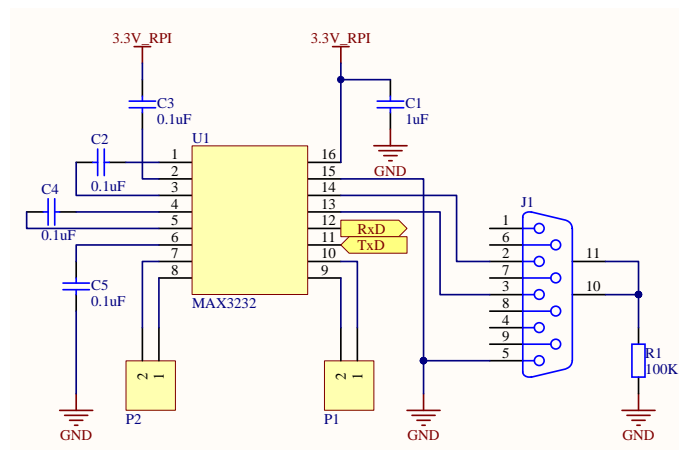


Figure 2.7: Schematics for the RS232-converter

2.4 Solar Power

Communication with the satellite through the groundstation network demands power from the satellite. Without any electrical power a satellite will not be able to support its payload or radio communication. NUTS double CubeSat uses sunlight as an energy source through solar panels.

Solar panels base their operation on the ability to convert sunlight into electricity. By using semiconductors the photovoltaic effect can be exploited. The conversion process where the suns radiation is converted into an electrical current is achieved by creating mobile charged particles in the semiconductor. They are in turn separated by the device structure and produce the electrical current.(kilde)

Batteries are used to store power during eclipse and support the payload. When the satellite is in eclipse the earth blocks the solar radiation and the battery must supply the power.

Identifying the satellites communication necessary to establish if a groundstation network is profitable. Forthcoming calculations are based on an estimate done by De Bruyne [9].

When estimating the period of the satellite the Earth and satellite orbit is assumed to be spheres. Kepler's third law for circular orbits is used:

$$\left(\frac{2\pi}{T}\right)^2 = \frac{GM}{R^3} \quad (2.1)$$

Where T is the period of the satellite, G is the gravitational constant¹, M is the mass of the Earth² and R is the distance between the centers of mass of the Earth and the satellite.

$$T = 2\pi \left(\frac{R^3}{GM}\right)^{\frac{1}{2}} \quad (2.2)$$

where $R = R_{\oplus} + h$ where R_{\oplus} is the radius of the earth³ and h is the altitude of the satellite. Since it is uncertain what altitude the satellite will settle in after it is launched, two heights are used in the calculations and it's assumed that the satellite wil settle somewhere within interval $h_1 = 450$ km and $h_2 = 650$ km. Which gives

¹ $G = 6.6742 \cdot 10^{-11} \text{ km}^3/\text{s}^2$

² $M = 5.9736 \cdot 10^{24} \text{ kg}$

³ $R_{\oplus} = 6371 \text{ km}$

$$T(h_1) = 5610 \text{ s} = 93.5 \text{ min} \quad (2.3)$$

$$T(h_2) = 5850 \text{ s} = 97.5 \text{ min} \quad (2.4)$$

when inserted into Equation 2.2.

From De Bruyns equations [9] the longest possible time in eclipse can be calculated. The worst case average power is approximately $P_{avg} = 5.42W$ from by De Bryens calculations. This power is calculated when the satellite has its longest time in eclipse.

$$t_{ecl,max} = 2R_{sat} \left(\frac{R_{sat}}{MR} \right)^{\frac{1}{2}} \quad (2.5)$$

$$P_{avg,orbit} = P_{avg} \times \frac{(t - t_{ecl,max})}{t} \quad (2.6)$$

Putting in $h_1 = 450 \text{ km}$ and $h_2 = 650 \text{ km}$ gives:

$$P_{avg,h_1} = 3.34 \text{ W} \quad (2.7)$$

$$t_{ecl,h_2} = 2118.9 \text{ s} = 35.3 \text{ min} \quad (2.8)$$

$$P_{avg,h_2} = 3.46 \text{ W} \quad (2.9)$$

These are simplified calculations where the temperature changes of the solar cells is not taken into account. As the satellite moves through orbit the temperature will effect the solar cells, but including this effect requires more extensive calculations. The average power calculated here is based on worst-case estimate from De Bruyns [9], the real average power can be calculated with the exact orbit parameters.

The battery used in the NUTS Cubesat will consist of lithium-ferrite-phosphate cells ($LiFePO_4$)[10]. These cells have a typical voltage of 3.3 V. NUTS will have $4 \times 1.1 \text{ Ah}$ cells, where two are in series and two series are in parallel. This means a total of $C = 2.2 \text{ Ah}$ at 6.6 V[9]. This is used to calculate the worst-case Depth Of Discharge (DOD). The DOD represents the percentage of the discharged battery capacity expressed as a percentage of maximum capacity. It indicates the state of charge where $DOD = 100\%$ is a empty battery and $DOD = 0\%$ is a full battery.

The battery capacity used during eclipse:

$$C_{ecl} = \frac{P_{avg,orbit}}{V_{tot}} \times t_{ecl,max} \quad (2.10)$$

$$DOD_{max} = \frac{C_{ecl}}{C_{tot}} \quad (2.11)$$

Calculated with the different heights gives:

$$\text{equation 2.10 } C_{ecl_1} = \frac{3.34 \text{ W}}{6.6 \text{ V}} \times \left(\frac{35.9 \text{ min}}{60 \text{ minhour}} \right) = 0.3 \text{ Ah}$$

$$\text{equation 2.11 } DOD_1 = 0.136 = 13.6\%$$

$$\text{equation 2.10 } C_{ecl_2} = \frac{3.46 \text{ W}}{6.6 \text{ V}} \times \left(\frac{35.3 \text{ min}}{60 \text{ minhour}} \right) = 0.31 \text{ Ah}$$

$$\text{equation 2.11 } DOD_2 = 0.141 = 14.1\%$$

The battery capacity will decrease with the number of charge-discharge cycles. Discharge of at least 80% is referred to as deep discharge. When the critical DOD is reached it will be in risk of battery failure. From our calculations the DOD is not in the critical region.

To establish the communication time a simplified power balance is used. We assume that the radio uses $P_{radio} = 2.5 \text{ W}$ and the payload uses $P_{payload} = 5 \text{ W}$ for ninety seconds. The standby power is unknown, but a microcontroller in standby running of 18.5mA and 3.3V uses $P_{standby} = 0.0612 \text{ W}$

These numbers come from various NUTS/NTNU publications are only rough estimates of what can be expected.

$$P_{avg,orbit} \cdot t - P_{payload} \times 90s - P_{standby} \times t = P_{radio} \times t_{communication} \quad (2.12)$$

Equation 2.12 is a simplified model for computing the amount of time the satellite can broadcast. Inserting h_1 and h_2 gives:

$$t_{communication} = 7172.12 \text{ s} = 110.5 \text{ min} \quad (2.13)$$

$$t_{communication} = 7778.77 \text{ s} = 129.6 \text{ min} \quad (2.14)$$

The calculations on power- and orbit estimates results in long communication time. There are several simplifications done during these calculations. The power drawn from the satellite are rough estimates and the available communication time will likely differ from these results. Since $t_{communication} > T$ we think that there will be enough power for communication even without these simplifications.

Chapter 3

The project

3.1 Ground station networks

There are a lot of unknowns in this area. We don't know which network will be used, who will be participating or what kind of radio equipment participants will have.

Some different possible networks are simulated using the same constraints as at Gløshaugen. The results are summarised in [Table 3.1](#).

These results show that even cooperating with other norwegian/scandinavian universities will result in significant gains in download capacity, even though there will be a certain overlap in coverage. Cooperating with other universities further afield will also increase download capacity, but if they are located at lower latitudes the decreased number of passes observed will decrease their contribution.

Other locations	Time	Improvement
Aalborg	790s	50%
Longyearbyen	1900s	260%
Narvik	900s	73%
Stanford	800s	54%

Table 3.1: Results of simulations

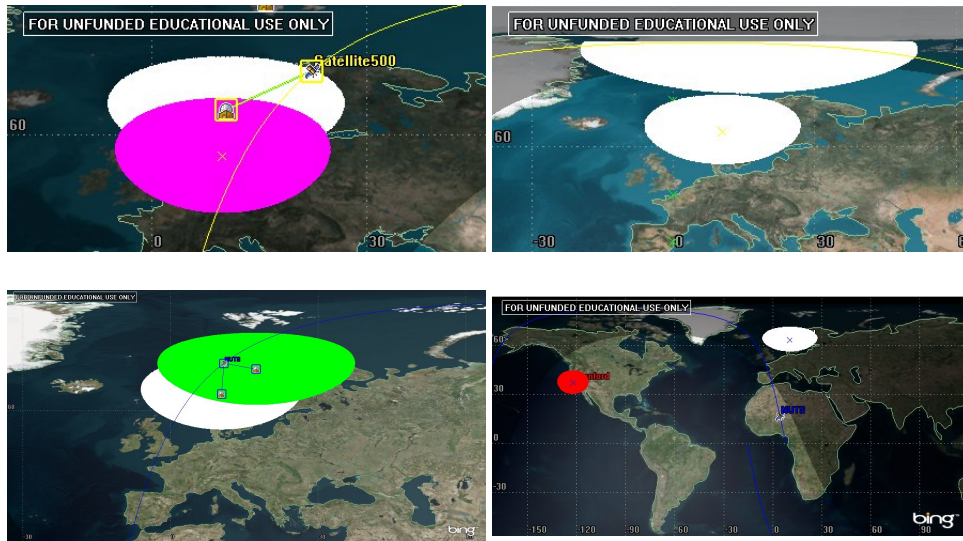


Figure 3.1: Some possible ground networks

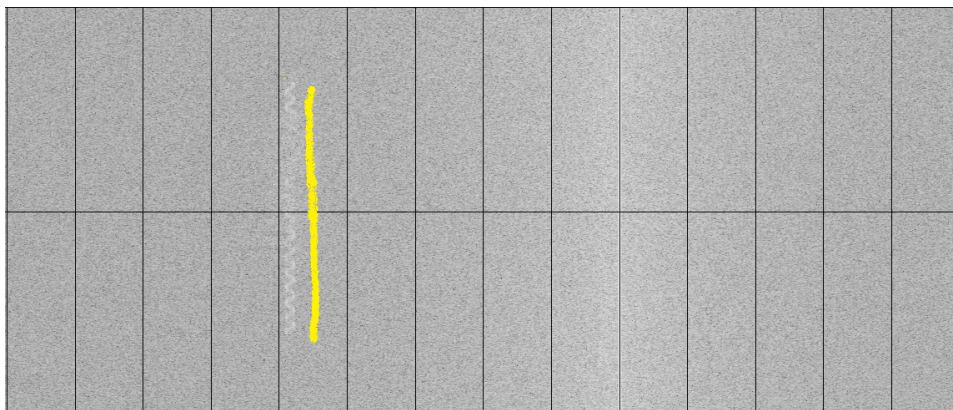
3.2 The project

The problem today is that the transfer antenna on the NUTS satellite is limited. We started looking for a network that connects ground stations to Internet. From there we wanted a ground station that was portable and easy to use (plug and play). We therefore decided to try out Raspberry Pi.

3.3 Software

The first thing we had to do was to install the Carppcomm software. We installed the CarpsD ground station control software. This is an open-source program primarily developed by the team behind Carppcomm. The purpose of this software is to make it possible to connect to the Carppcomm server. It also makes it possible to control the ground station if you have access to a web-browser. This is possible because the software runs a background process on the Raspberry Pi. The software runs continuously in the background making it possible to connect to at any time and start receiving from satellites.

The next thing we had to do was to create an entry for our station plotting the latitude, longitude and elevation. Since the Raspberry didn't have a connector for the box that controls the antenna we had to make a cable so we could get the two components to communicate. (image) We had some



troubles with this cable because it wasn't robust enough. We thought about making a new one, but we didn't have the time for it. The last thing we had to do before we could start testing was to install the software for the antenna controller. We installed the Hamlib rotator library since it was supported by our rotator.

Now everything was set to start testing the station. We started of with connecting a small TV-antenna to see if we got a signal. On the station page at the Carpcomm website we tuned it to 433.3 MHz. The reason for this frequency is because its used by the police in Trondheim and there are likely to be a lot of transmissions there. We managed to receive data and publish it to the Carpcomm network.

A short explanation of the recorded transmission. The x-axis is the frequency and the y-axis it time. Black and gray is noise and the white left of the yellow line is signals.

Next stop was to try with a full scale antenna. We succeed in receiving data with the Carpcomm software, but we could not get the automated rotor control to work. So this remains unsolved. Another problem is that the Carpcomm software only supports receiving data, not sending. So if we must command the satellite to start transmitting, some other solution must be devised.

3.4 Further Work

As it is now Carpcomm can not be combined with our own usage of the antennas and radio. It is fairly simple to implement code to upload data to Carpcomm, then we will be able to control the radio and antennas as we wish and still upload data to the network when where not using the radio ourselves. This will also let us use the big radio rig instead of the small radios that Carpcomm supports.

If the Carpcomm software shall be used, the rotor control should be looked in to. It is probably necessary to make an modification to carpsd or to the driver for the rotor interface.

The existing rotor interface have some quirks that makes the movement very choppy which can be bad for the antennas and rotors. Because of this it is desirable with a new rotor interface. The Raspberry Pi have the necessary inputs and outputs to act as an interface, it only needs some voltage converters and some software to control it.

3.5 Conclusion

The Carpcomm software are easy to install and set up, as long as the radio are supported. The supported radios are all small and inexpensive, which makes them suitable for radio amateurs who want to help with receiving satellite transmissions. These smaller radios can only use one antenna, and since only one radio can be used Carpcomm can only receive either UHF or VHF signals. Unfortunately this makes it less suitable for us, since we want to use our bigger radio which is not supported.

Since we could not get the software to control the antenna rotors we can only receive data from satellites in a small area of the sky, which makes the station less useful.

Carpcomm is only suitable for data that can be transmitted continuously. Since the Carpcomm network can't be used to transmit to the satellites it is impossible to tell the satellite that it should send its data.

Carpcomm are suitable for amateurs who want an inexpensive way to help the micro satellite community. For professional usage it lacks important features and hardware compatibility.

Bibliography

- [1] *AAUSAT3*. February 13 2013. Available at: [<http://www.space.aau.dk/aausat3/>](http://www.space.aau.dk/aausat3/)
- [2] *Carpcomm Space Network*. Available at: [<http://carpcomm.com/howitworks.html>](http://carpcomm.com/howitworks.html)
- [3] *Carpcomm Ground Station 1*. Available at: [<http://carpcomm.com/gs1.html>](http://carpcomm.com/gs1.html)
- [4] *Raspberry pi*. Available at: [<http://www.raspberrypi.org/>](http://www.raspberrypi.org/)
- [5] Amundsen, Morten et al., Ekstern kommunikasjon, 2011
- [6] *The educational part of ESA's home page*. Available at: [<http://www.esa.int/Education/Global_Educational_Network_for_Satellite_Operations/>](http://www.esa.int/Education/Global_Educational_Network_for_Satellite_Operations/)
- [7] Marholm, Sigvald, Master Thesis: Antenna Systems for NUTS, July 2012
- [8] Bakken, Marianne, Master Thesis: Signal Processing for Communicating Gravity Wave Images from the NTNU Test Satellite, July 2012
- [9] De Bruyn, Dewald, Master Thesis: Power Distribution and Conditioning for a Small Student Satellite, June 2011
- [10] Birkeland, Roger, Gutteberg, Odd, IAA-CU-13-09-09 Overview of The NUTS Cubesat Project, February 2013
- [11] Holberg, Fredrik Sola, Master Thesis: Electrical Power System of the NTNU Test Satellite, June 2012