

Michael Berg
Karl Grodås
Johan Vu

Development of an Independent and Mobile Measurement System for Geographical Mapping of 5G Coverage

In collaboration with SINTEF and Maritime Robotics

Bachelor's thesis in Electrical Engineering
Supervisor: Arne Midjo, Are Hellandsvik, Bård Myhre and Stig Petersen
May 2022

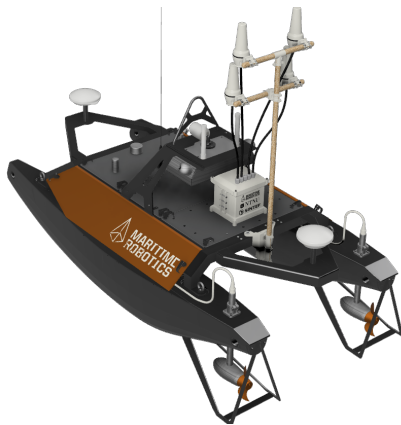


Norwegian University of
Science and Technology

Michael Berg
Karl Grodås
Johan Vu

Development of an Independent and Mobile Measurement System for Geographical Mapping of 5G Coverage

In collaboration with SINTEF and Maritime Robotics



Bachelor's thesis in Electrical Engineering
Supervisor: Arne Midjo, Are Hellandsvik, Bård Myhre and Stig Petersen
May 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems



Kunnskap for en bedre verden

Additional information

Thesis title:

ENG: *Development of an Independent and Mobile Measurement System for Geographical Mapping of 5G Coverage.*

NO: *Utvikling av et Uavhengig og Mobilt Målesystem for Geografisk Kartlegging av 5G-dekning.*

Group participants:

Michael Berg, m-r-berg@hotmail.com, +47 980 58 568

Karl Grodås, karlgrodaas@gmail.com, +47 974 88 552

Johan Vu, johan.2000@hotmail.no, +47 920 47 766

Date: 20/05/2022	Sub. code: IELET2900	Subject: Bachelor thesis Electronics and Sensor Systems	Document access: Open
Study program: Electrical Engineering		Num. pages/appendix: 53 / 21	Num. references: 33
Supervisors:			
Arne Midjo (NTNU), arne.midjo@ntnu.no, +47 734 12 662			
Are Hellandsvik (Sintef), are.hellandsvik@sintef.no, +47 997 12 637			
Stig Petersen (Sintef), stig.petersen@sintef.no, +47 930 03 112			
Bård Myhre (Sintef), bard.myhre@sintef.no, +47 917 39 801			

1 Abstract

In collaboration with SINTEF and Maritime Robotics, this thesis' ambition was to *develop an independent and mobile measurement system for geographical mapping of 5G coverage*. The concept required creating a system that could both be portable in addition to obtain coverage measurements autonomously and independently.

The project group began researching fundamental theories and technologies essential to better understanding the task at hand. It was decided that the key parameter to be measured was signal strength. A 5G modem was acquired, and the group began developing a system around it.

For the system to become mobile and independent, it needed a *power system* and a *computing device*. A battery was used to supply power to the system. The Raspberry Pi 4B was chosen as the computing device as it presented Ethernet capabilities for connection with the modem, and onboard memory for storing measurement data. An encapsulation was 3D printed to hold and protect all the components from harsh environments.

A python script was created to automate obtaining and saving data from the modem. Followingly, the GPS position at the measurement location was retrieved, thus acquiring the measurement system's position along with its measurement.

After the measurement data was obtained, it had to be visualized. This was done utilizing Google Earth. There were two methods used to visualize the data. One plot circles to indicate measurements taken on a trip along a route, and the other forms a grid to present the coverage of a larger area.

The resulting measurement system was complete and functioning system. Tests were conducted on land before being mounted on an unmanned surface vehicle, and measurements were obtained in Trondheim Fjord. The measurements were then visualized successfully.

Sammendrag

I samarbeid med SINTEF og Maritime Robotics ble det fastsatt at oppgavens formål var å utvikle et uavhengig og mobilt målesystem for geografisk kartlegging av 5G-dekning. Konseptets krav var å skape et system som både var mobilt i tillegg til å kunne hente måledata automatisk og uavhengig av teleoperatørene.

Prosjektgruppen begynte i startfasen med å undersøke essensielle teorier og teknologier for å opparbeide seg en dypere forståelse av oppgavens formål. Det ble besluttet at signalstyrken var den mest interessante parameteren. Gruppen begynte deretter å utvikle et system med utgangspunkt i et 5G-modem.

For at systemet skulle bli mobilt og uavhengig, var det en nødvendighet å implementere et strømforsyningssystem og en datamaskin. Det ble tatt i bruk et batteri for å forsyne systemet med strøm. En Raspberry Pi 4B ble også brukt som dataenheten da den tilbyr en rekke fordeler som Ethernet-tilkobling, samt innebygd minne for lagring av måledata. En egendesignet innkapsling ble 3D-printet for å beskytte systemet fra dens omgivelser.

Kode ble utviklet i Python for å automatisere prosessen av innhenting og lagring av data fra modemmet. I tillegg ble GPS-koordinatene av målesystemet samlet opp, slik at posisjonen kom i samsvar med målingene.

Etter at måledataene var innhentet, ble de visualisert ved bruk av Google Earth. Det ble brukt to metoder for å visualisere dataene. Den ene tegnet sirkler for å indikere målinger tatt på en rute, og den andre dannet et rutenett for å presentere dekningen av et større område.

Det resulterende målesystemet endte opp som et komplett og fungerende system. Det ble utført tester på land før det ble montert på et ubemannet kjøretøy og testet i Trondheimsfjorden. De oppsamlede målingene ble til slutt visualisert på en vellykket måte.

2 Preface

This report marks the end of the last semester of our 3-year Bachelor in Electrical Engineering program at NTNU Trondheim.

During the course of the project, the product went through several iterations. Rapid prototyping and testing led to constant revisions and improvement. In the beginning, the system was tested in a lab, but we quickly went mobile by using a bike for transportation. In the final stage, we completed our goal of mounting the system on one of Maritime Robotics *unmanned surface vehicles* (USV). This was a great experience and led to successful measurements and a coverage map of Telia's 5G cellular network in Trondheim Fjord.

We want to thank our advisors at SINTEF, Are Hellandsvik, Bård Myhre, and Stig Petersen for sharing their bottomless wisdom throughout the bachelor thesis. We would like to thank Helge Rustad for his expertise regarding waterproofing of the encapsulation, Guillaume Bour for guidance and 3D printing, and Knut Marius Røeberg for his help in the workshop and laboratories. Additionally, we are grateful for the Makerspace workshop, equipment, funding, office, and amenities provided to us by SINTEF.

Gratitude is directed towards our advisor, Arne Midjo at NTNU, allowing us to write this custom thesis and the guidance he has given us. We also want to thank Stein-Olav Lund at NTNU for help in the selection of coaxial cables.

We appreciate Maritime Robotics for their cooperation and allowing us to use their USV to obtain measurements in Trondheim Fjord. Thank you to Arild Hepsøe for coordinating meetings and giving us insight into useful technologies employed at Maritime Robotics. Thank you to Jon Petter for help with planning and mounting our system on the USV, and thank you to Eirik Moholt for support on the day of testing with regards to mounting our system and operating the USV. Maritime Robotics' cooperation in this project has been indispensable. In addition to their generosity in allowing the use of their USV, they have come with essential advice and guidance throughout the project.

Trondheim, 20/05/2022


Michael Berg


Karl Grodås


Johan Vu

Table of contents

1	Abstract	ii
2	Preface	iv
3	Introduction	1
3.1	Motivation	1
3.2	Concept	2
3.3	SINTEF	2
3.4	Maritime Robotics	2
3.5	Thesis Structure	2
4	Theory	4
4.1	Geographic Coordinate System	4
4.2	Electromagnetic Waves	4
4.2.1	Wireless Communications	5
4.2.2	Antenna	5
4.2.3	Radiation Pattern	6
4.2.4	Path Loss and Interference	6
4.2.5	Free-space Loss	7
4.2.6	Propagation over Sea	7
4.3	5G Frequency Bands	7
4.4	Measurement Parameters	8
4.4.1	Decibel	8
4.4.2	RSRP – Reference Signal Received Power	8
4.4.3	RSRQ – Reference Signal Received Quality	9
4.4.4	SINR - Signal to Interference and Noise Ratio	9
4.5	SNMP - Simple Network Management Protocol	10
4.6	KML - Keyhole Markup Language	10
4.7	Battery Technology	11
4.7.1	Low Voltage Disconnect	11
4.8	Fused Deposit Modeling - 3D Printing	11
4.8.1	Polylactic Acid - Filament	12
5	Hardware	13
5.1	System Architectures	13
5.2	Power System	13
5.2.1	DeWalt Battery	14
5.2.2	LVD - Low Voltage Disconnect	14
5.2.3	LM2596 Module	15
5.2.4	XT60 Connector	16
5.3	Computation	16
5.3.1	Raspberry Pi	16
5.3.2	Pepwave BR1 Pro 5G	17
5.4	Antenna	17

5.4.1	OMIN-404 Maritime Antennas	18
5.4.2	Coaxial Cables	18
5.4.3	Total Gain	18
5.4.4	GPS Antenna	19
5.4.5	Antenna Tower	19
5.5	Encapsulation	20
5.5.1	3D Printing and Limitation	20
5.5.2	Outer Shells	20
5.5.3	Threaded Inserts	21
5.5.4	Water Seal	21
5.5.5	Rail System	22
5.5.6	Packing Nipple	23
5.5.7	Expansion Slots	23
5.6	User Interface	23
6	Software	24
6.1	Base Shell	24
6.2	Raspberry Pi	24
6.3	Pepwave Software	24
6.3.1	SNMP Connection	25
6.3.2	GPS Forwarding	25
6.4	GUI - Blynk	25
6.5	Python	26
6.5.1	Important Libraries	26
6.5.2	Creating Data Files	26
6.6	KML - Data Visualization	27
6.6.1	Quantification and classification of Data	28
6.6.2	Trip Measurements	28
6.6.3	Base Stations	29
6.6.4	Heading	30
6.6.5	Grid Measurements	30
6.6.6	Grid Visualization - Processing	31
6.6.7	Grid Visualization - Presenting	31
6.6.8	Grid Commonality	32
7	Method	34
7.1	Risk Analysis	34
7.1.1	Lab Work	34
7.1.2	Field Work	34
7.1.3	Safety Procedure	35
7.1.4	Start and Stop a Measurement Session Procedure	35
7.2	Power System Test	36
7.2.1	LVD Test	36
7.2.2	System Power Consumption	36
7.3	Dry Runs	37
7.3.1	Run for an Extended Period	37

7.3.2	No Cellular Test	38
7.3.3	No Cellular and GPS Test	39
8	Results	40
8.1	On a Bike	40
8.2	On an USV	41
9	Discussion	44
9.1	Project Expectations	44
9.2	Gain Adjustment	44
9.3	Use Cases	45
9.4	Encapsulation	45
9.5	BMS Module	46
9.6	PCB Production	47
9.7	Visualization	48
10	Future work	50
10.1	SSH into Raspberry Pi	50
10.2	GUI Website and Database	50
10.3	Sensor Implementation	50
10.4	Multiple Carriers Simultaneously	51
10.5	Secondary Cell Providers	52
10.6	Delimitation of Improvements	52
11	Conclusion	53
12	References	54
13	Appendix	57

3 Introduction

This project, conducted in the spring of 2022, aimed to develop an independent and mobile measurement system for geographical mapping of 5G coverage on land and at sea for SINTEF in cooperation with Maritime Robotics. The system was supposed to be independent of the telecom operators and thus deliver a third-party mapping of 5G coverage. Mobility was an obvious necessity for the system used to take measurements over various areas. Generating coverage maps were an essential step of the project and would visualize the data collected by the system.

The project's scope was limited to the development and testing of the system. Therefore analysis of the collected data was kept to a minimum. Some hypotheses were established based on the findings, but these were not explored in depth.

The final test of the system was to mount it on one of Maritime Robotics' USVs and obtain measurements in Trondheim Fjord. A point of interest was the SINTEF OceanLab Observatory, referred to as the SINTEF Ocean buoy, located just north of Munkholmen in Trondheim Fjord. Utilizing the USV, the system would make a trip in the fjord and grid a small section around the buoy.

Using the data collected during the final test and the tests leading up to it, this project hopes to showcase the possibilities of the system and its many use cases.

3.1 Motivation

The project was motivated by the rapid expansion of the 5G cellular network. The configurability of 5G makes it adaptable for different use cases depending on the requirements of the consumer [1]. For many industries using 5G will provide higher speed, lower latency, and greater capacity than existing wireless solutions. An independent measurement system would give businesses the tools to make educated decisions based on recorded data.

There were many different parameters to consider when measuring a telecommunication link, but this project focused on the strength of the connection or signal strength. The goal was to generate coverage maps to show the reach of the 5G cellular connection. Thus enabling a better understanding of the real-life possibilities produced by this new technology.

3.2 Concept

The measurement system concept as a whole consisted of three parts. The first was the physical components, connections and encapsulation of the hardware. The second was software, responsible for obtaining and storing measurement data. The third and last part was the visualization of obtained measurement data.

These parts functioning together created a product that was easy to use and would quickly give accurate data on the 5G coverage.

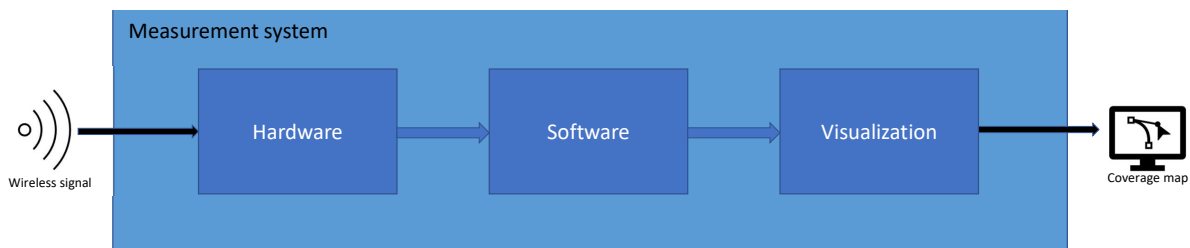


Figure 1: Measurement system concept

3.3 SINTEF

SINTEF is the largest independent research organization in Scandinavia and among the largest in Europe. Being a non-profitable organization, SINTEF carries out several thousand projects for customers large and small every year. Among other things, they aim to deliver innovation by developing knowledge and technologies that are brought into practical use [2]. SINTEF Digital and SINTEF Ocean are two subgroups within, and both play an essential role in this Bachelor thesis as partners and advisors.

3.4 Maritime Robotics

Maritime Robotics is a *leading provider of innovative unmanned solutions* and is based in Trondheim. One of their USVs, the Otter, is used in this project. It offers a highly modular and flexible platform for various operations [3]. Therefore, their role in this bachelor's thesis has been indispensable. The group would have had to search high and low for another partner with the same cooperative willingness and operational possibilities.

3.5 Thesis Structure

The following sections of this thesis give a brief recap of the necessary *theory* upon which the project is based. This is to enlighten the reader on the fundamental concepts,

which may not be general knowledge but is essential to understanding the measurement system's different parts.

Furthermore, the three consecutive sections *Hardware*, *Software*, and *Method* are all project-specific choices made during the process of this thesis to end up with the resulting measurement system. Hardware contains the different physical components or modules used and their relevant specifications, which were essential to take into account during the development. It starts by briefly explaining the system architecture before each component is presented in depth. Software introduces the different libraries and approaches used for the two central parts in this project; *obtaining measurements* and afterward with the *processing and mapping*. Following, the Method section includes different procedures and routines that were made to lower the risk of failure during testing and minimize uncertainties. This section also contains some designated experiments on the parts or the whole system to understand its behavior better.

Afterward, chapter *Results* includes a description of the two measurement tests *on a bike around Gløshaugen* and *on a USV in Trondheim Fjord* with Maritime Robotics, followed by a visualization of the results taken.

In the *Discussion* chapter, reflections are made on the measurement system's expectations, in addition to the different interconnections made by analyzing the results. *Future work* is also a part of the discussion but is separated into its section for a more clear thesis structure. This chapter contains the following approaches to implement based on a reflection of the failure points within the measurement system.

Lastly, *Conclusion* rephrases the thesis statement and summarizes the modules and approaches discussed in the different chapters above. It draws back to the introduction and answers whether the measurement system was a successful product.

4 Theory

To properly understand the work presented in this thesis, it is necessary to understand the underlying principles. Therefore, the most important concepts are shortly described here.

4.1 Geographic Coordinate System

The Geographic Coordinate System enables the measurement and communication of any position on earth using latitudinal and longitudinal coordinates. The latitude represents the north-south position, using degrees above or below the equator. Latitudinal coordinates ranges from 0° to $\pm 90^\circ$; 0° represents equator, 90° represents the north pole, and -90° represents the south pole. Longitude specifies the east-west position, again using degrees, from 0° to $\pm 180^\circ$. For longitude, it is the *prime meridian*, located near Greenwich in England, that is defined as 0° , travel in the eastward direction results in a positive increase and thus will approach $+180^\circ$. Conversely, westward travel will result in a negative prefix of -180° , $+180^\circ$ and -180° are equivalent. Positions on the globe are specified in the format (latitude, longitude), commonly referred to as (LAT, LON). Note that this is a spherical representation, and thus it is not synonymous with the Cartesian (x, y) representation. Latitude would, in that case, refer to "y", and longitude would refer to "x".

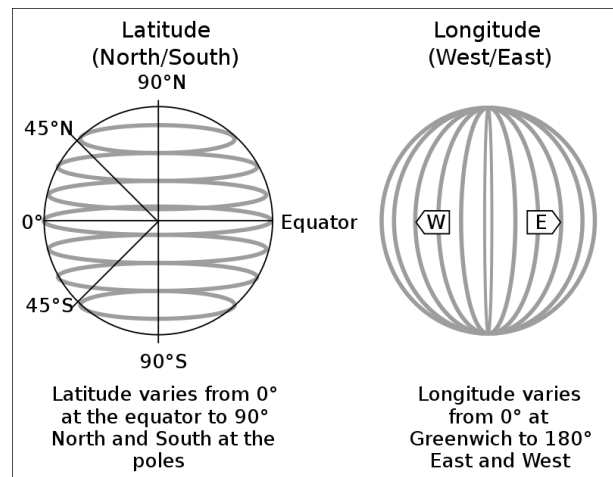


Figure 2: Spherical coordinate system [4]

One may also see the use of suffixes such as N, E, S, and W. By use of these suffixes, all coordinates can be expressed in positive numbers, $(-27.11^\circ, -109.35^\circ)$ is equivalent to $(27.11^\circ\text{S}, 109.35^\circ\text{E})$.

4.2 Electromagnetic Waves

Both electric and magnetic fields can be static, like those of static electricity and magnets. One can not exist without the other, and a changing electric field will result in a changing magnetic field and vice-versa. This cohesion between magnetic and electric fields form the basis for the designation *electromagnetic*. Waves are created when the fields vary. These electromagnetic waves differ from all other forms of waves in the

sense that they do not need a medium to propagate. This stands in contrast to sound waves that need a medium like air (one will not be able to hear sounds in a vacuum).

Electromagnetic waves travel at the speed of light and can do so with a wide range of frequencies with different energy levels. The wavelength is the distance between the slopes in the wave. The relationship between speed, frequency, and wavelength is given by the equation below [5].

$$\lambda = \frac{c}{f} \quad (1)$$

- λ : Wavelength
- c : Speed of light
- f : Frequency

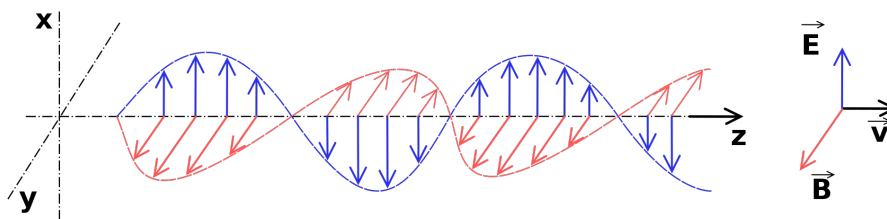


Figure 3: Electromagnetic wave, electric and magnetic field [6]

4.2.1 Wireless Communications

With electromagnetic waves, it is possible to communicate over great distances by manipulating a sinusoidal *carrier* signal. This is done in a process known as *modulation* where information is superimposed on the carrier by creating changes to either its phase, frequency, or amplitude. At the receiving end, the signal is *demodulated*, and information is extracted from the carrier.

Encoding is also necessary for complex forms of communication, e.g., in modern communication systems, speech must be encoded into bits and bytes, digital data, before it becomes modulated and transmitted [7].

4.2.2 Antenna

An antenna is essentially a transducer that converts electromagnetic waves in free space to electromagnetic waves in a cable or vice versa. The length and shape of the

antenna determine its characteristics and performance. For example, a longer antenna will be better suited to receive and transmit at lower frequencies where the wavelength is longer. Conversely, a shorter antenna would be better suited for signals with higher frequencies [8].

4.2.3 Radiation Pattern

A radiation pattern is a three-dimensional pattern that visualizes how well an antenna sends and receives electromagnetic waves in different directions. An ideal antenna would have an isotropic radiation pattern meaning it would radiate equally in all directions. The 3D diagram of an isotropic antenna would be a perfect sphere.

The radiation pattern of an antenna is helpful when seeking information about how much gain, or loss, one would have at different angles and frequencies. Referenced to an isotropic antenna, the unit of measurement is dBi [8].

The radiation pattern is commonly displayed as a 2D diagram for practical reasons. For an antenna seen from above, the gain would be displayed at angles from 0° to 360° .

4.2.4 Path Loss and Interference

Path loss refers to reducing power in an electromagnetic wave as it travels through space. In ideal conditions, there would be no path loss. However, in real-life applications of wireless communications, one must deal with objects that interfere and cause path loss. There are numerous possible causes of path loss, such as; refraction, diffraction, reflection, absorption, and shadowing effects [7].

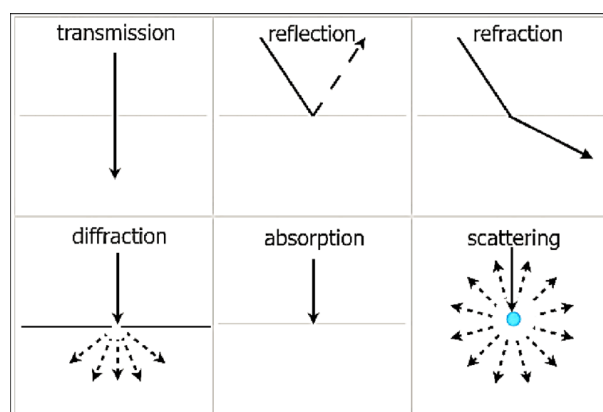


Figure 4: Shows the type of wave's transmission, reflection, refraction, diffraction, absorption and scattering [9]

4.2.5 Free-space Loss

When two transmitters have an unobstructed and clear line of sight, there will still be attenuation, called free-space loss. This is due to how a wave propagates and disperses, causing the energy density to decrease with increasing distance. A formula commonly used in telecommunications engineering is the *Friis transmission equation*. Under ideal conditions, the formula can calculate the power at a receiver at a given distance from a transmitter when the transmitting power is known [7].

The formula is given as follows:

$$P_r = G_t G_r P_t \left(\frac{\lambda}{4\pi r} \right)^2 \quad (2)$$

- P_r : Received power
- G_t : Transmitter gain
- G_r : Receiver gain
- P_t : Transmitted power
- λ : Wavelength
- r : Distance between transmitter and receiver

4.2.6 Propagation over Sea

Propagation of electromagnetic waves over saline and non-homogeneous surfaces such as the ocean differs significantly from the assumed ideal conditions that lay the basis for Friis' transmission formula. This is due to the reflection on the sea surface, as well as refraction through waves and potentially in the humid air as a result of evaporation. Additionally, the extra humidity can contribute to an increased attenuation due to absorption [10].

4.3 5G Frequency Bands

Norway has two leading telecom operators, Telia and Telenor. They have allocated 5G frequencies for both mobile communication and industrial use. These frequencies span from 3400 MHz to 3800 MHz and are also known as the n78 frequency bandwidth. With an already large distribution and coverage of the existing 4G network, most of the 5G Radio Networks in Norway are built upon the core of 4G, making it a *Non-Standalone* (NSA) architecture [11]. Hence, its advantages of being cheaper and more efficient to expand the network, in opposition to building a *Standalone* (SA) version of 5G.

4.4 Measurement Parameters

Based on the measurement parameters presented in both of the scientific articles from Poynting Tech, [12] and Cable-free [13] the chosen parameters to be measured and studied are *Reference Signal Received Power* (RSRP), *Reference Signal Received Quality* (RSRQ), and *Signal to Interference and Noise Ratio* (SINR). Furthermore, it is essential to establish *Received Strength Signal Indicator* (RSSI), serving as both RSRP and RSRQ. In short, RSSI is a presentation of the received power in a wireless system.

4.4.1 Decibel

Decibel (dB) is a relative unit commonly used to express quantities in the field of electrical engineering, and a logarithmic scale is used. When expressing power ratios, one uses a factor of 10 as the logarithmic base. Thus a halving of power will be expressed as -3dB. A factor of 20, also known as amplitude ratio, will be -6dB, and for both ratios, 0dB corresponds to unity. Another commonly used unit is dBm, being the same logarithmic relative unit. However, it is referred to as 1mW. Thus 1mW is 0dBm and 100mW is 20dBm [7].

As dB is a relative unit, calculating averages is not straightforward as linear units. Linearly averaging is also known as arithmetic averaging, and averaging for relative dB values is known as logarithmic averaging. Take the following example of two linear values, 10 and 20. Both are to be expressed in dB (power ratio). Thus we have 10dB and 13dB. The arithmetic average for both is then 15 and 11.5dB. However, the latter is incorrect as 11.5dB equals 14 and not 15. The correct answer would be 15 expressed in dB (power ratio) and thus 11.75dB. As dB is a logarithmic scale, this deviation will increase exponentially.

$$\text{Quantity in dB (power ratio)} = 10 \cdot \log\left(\frac{P}{1W}\right) \quad (3)$$

$$\text{Quantity in dBm (power ratio)} = 10 \cdot \log\left(\frac{P}{1mW}\right) \quad (4)$$

$$\text{Quantity in dB (amplitude ratio)} = 20 \cdot \log\left(\frac{A}{1}\right) \quad (5)$$

4.4.2 RSRP – Reference Signal Received Power

RSRP is derived from RSSI and is a power measurement commonly expressed in dBm used to quantify the relative received power to a reference signal. RSRP values greater

than -80dBm is considered *excellent*, and less than -100dBm is considered to be *cell edge* [13].

4.4.3 RSRQ – Reference Signal Received Quality

RSRQ is a measurement to quantify the signal quality considering RSSI as well as the number of used *resource blocks*, resource blocks being the smallest unit of resources allocated to a user. In LTE, it has a frequency width of 180kHz and one allocated *slot of time*. An RSRQ of 0 dB is the best possible scenario. However, an RSRQ greater than -10 dB is considered *excellent*. RSRQ provides additional information when RSRP is insufficient in providing reliable data. [14] RSRQ can be calculated as:

$$RSRQ = \frac{N \cdot RSRP}{RSSI} \quad (6)$$

- *RSRQ* : Reference signal received quality
- *N* : Number of resource blocks
- *RSRP* : Reference signal received power
- *RSSI* : Received signal strength indicator

4.4.4 SINR - Signal to Interference and Noise Ratio

SINR measures the signal quality based on the ratio between the signal strength and the present noise, including interference. Therefore, it is desirable to have as high of a SINR as possible. Ideally, it would be infinite, meaning there is no noise and interference. To increase the SINR, one must transmit with more power or seek means of minimizing the noise and interference.

SINR is measured in dB, where the intelligent signal is referenced to the noise. A SINR of 0 dB means that the signal strength equals the noise and interference. Greater SINR values indicate that the signal is more substantial than the noise and interference. Values below zero are also possible and equate to a situation with more noise and interference than intelligent signal present.

Greater SINR allows for more significant modulation and thus greater data rates. A good way to think of SINR is trying to listen to a person speaking in a loud environment, like a construction zone. The person speaking would likely have to speak louder or repeat themselves to ensure that the information is reliably transmitted [15].

RF Experience	RSRP (dBm) Reference Signal Received Power	RSRQ (dB) Reference Signal Received Quality	SINR (dB) Signal to Interference & Noise Ratio
Excellent	≥ -80	≥ -10	≥ 20
Good	-80 to -90	-10 to -15	13 to 20
Mid Cell	-90 to -100	-15 to -20	0 to 13
Cell Edge	≤ -100	≤ -20	≤ 0

Figure 5: Poynting quantization of measurement values [12]

4.5 SNMP - Simple Network Management Protocol

SNMP is a standard protocol used in internet applications to monitor information concerning devices on IP networks as well as modify device behavior. SNMP may be used to obtain information about a device's status, e.g., wireless parameters, device information, and more. Furthermore, with SNMP, one can change certain settings over Ethernet as well as WiFi. Each individual device parameter has a unique Object Identifier (OID) in numbers, identical to how IP addresses are written. The OIDs are well organized into tree-shaped directories, or so-called Management Information Bases (MIB). They are unique to every manufacturer, even though the market has a general MIB structure. By using the common SNMP functions as *GetRequest* or *SetRequest* one may change or retrieve a desired OIDs information. Essentially SNMP enables configuration, organization, and information to be managed exclusively by computers and programs [16].

4.6 KML - Keyhole Markup Language

Keyhole Markup Language (KML) is based on the *Extensible Markup Language*, also known as XML, notation and is used for visualization and geographic annotation in 2D maps and 3D Earth browsers. Google has been responsible for KML since 2014, and it is used as the international standard for the *Open Geospatial Consortium*. Within KML, one can geographically place images onto a map and control their size, color, and transparency. This is known as a *ground overlay*. Furthermore, the implementation of a *screen overlay* is possible and thus can be used as a legend [17].

4.7 Battery Technology

The most common battery technology employed in rechargeable consumer-grade batteries, the lithium-ion (Li-ion) battery, can be a safety hazard.

Li-ion cells may pose a fire hazard as they contain flammable electrolytes. In addition, internal short circuits can occur if the battery cells were to charge or discharge too quickly, as well as if the battery cells were to be charged or discharged to unsafe voltage levels[18].

4.7.1 Low Voltage Disconnect

A *Low Voltage Disconnect* (LVD) or commonly also referred to as a *Low Voltage Cutoff Circuit* serves the purpose of disconnecting the battery when the voltage drops below a certain threshold. This will, in turn, protect the battery against discharging it to a voltage lower than the safe range. To ensure optimal battery health and user safety, it is desirable to implement an LVD in any electrical, battery-driven product or appliance.

Most LVD circuits require some current to operate and would eventually drain the battery even after cutoff. However, the current can be small enough to increase the life of a battery to several years if left unattended. In practice, this is considered a safe solution [19].

4.8 Fused Deposit Modeling - 3D Printing

Fused Deposit Modeling (FDM) is a standard 3D printing technology. It uses a string of *filament* fed through a *hot end*. At the hot end is a heated nozzle to deposit melted material onto the model. The material quickly cools down and solidifies, building the model. The heated nozzle is mechanically moved to add more material to the model continuously.

To avoid the hot end colliding with an already built structure, the model is cut into horizontal layers in a process called *slicing*. The first layer is printed on a solid surface called a *build plate*, and each layer is printed in its entirety before the next one. This ensured that the hot end always is above the previously printed structure.

The printing method in layers introduces a constraint on the model's geometry. All layers of the model must be printed on top of the already existing layers. The exception is the first layer which is printed on the build plate. However, the method does allow some degree of an *overhang* where a new layer exceeds

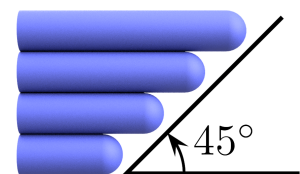


Figure 6: 45° overhang

the previous one. Generally, the overhang can not be greater than a 45° angle.

The support structure can be printed if a model demands an overhang greater than 45° . Support is extra material added to the layers underneath the overhang. This can later be removed from the model after printing. However, the process of printing support structure uses more filament than without, and the removal process can often be challenging.

4.8.1 Polylactic Acid - Filament

Polylactic Acid (PLA) is a soft plastic material widely used as filament for 3D printing. It has the benefit of being simple to use and relatively cheap. In addition, it is a bioplastic that can be made from renewable sources.

The downside to PLA comes from its poor physical properties. It has a glass transition temperature of $60-65^\circ$ and a melting temperature of $130-180^\circ$. It is also quite brittle [20].

5 Hardware

Considerable effort was devoted to selecting hardware and the system architecture layout. The motivation behind the different choices and the restrictions that dictated much of the design is presented here.

5.1 System Architectures

The measurement system consisted of several components that all needed to work together to achieve the goal of obtaining accurate measurements of the 5G cellular coverage. The system was divided into five subsystems. The five subsystems were *Power System*, *Computation*, *Antenna*, *User Interface*, and *Encapsulation*. A complete diagram showing all the subsystems with their respective components is shown in figure 7.

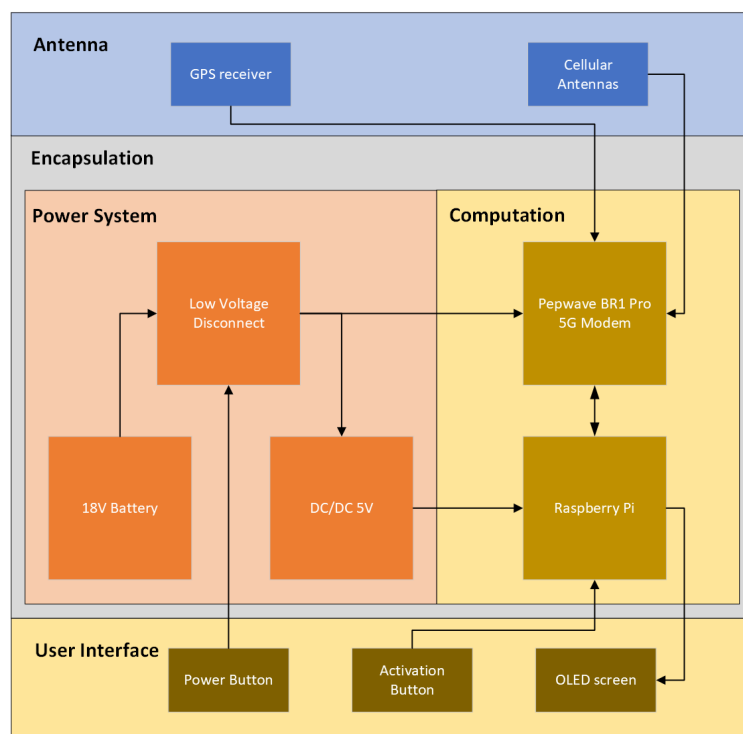


Figure 7: Complete System Architecture

5.2 Power System

To realize the goal of mobility, the system needed some form of battery. The different components of the system required different voltage levels, and the system had to be

able to deliver enough current without having an impact on the voltage stability. Lastly, the Power System needed to be relatively small and lightweight to make transportation of the system practical.

As previously stated in section 4.7, using a battery introduced numerous safety issues. Batteries need to be protected against short circuits, low voltage discharges, and overheating. It would also be advantageous to use a rechargeable battery, where the recharging should be done safely to ensure optimal battery health.

5.2.1 DeWalt Battery

The battery used for the measurement system in this project was the *DeWalt 18V 5Ah XR LI-Ion battery*. As the name suggested, the battery delivered 18V and had a capacity of 5Ah. The DeWalt batteries were designed for use with power tools and could handle substantial current draw while maintaining a stable output voltage. It was rechargeable and came with its own charger. The batteries also had a capacity indicator for feedback on battery level. DeWalt batteries could be bought from many resellers, and replacements could easily be acquired if needed. Last but not least, DeWalt is a trusted brand with a good reputation [21].

The power pins on the DeWalt battery were well-hidden and not exposed. This ensured that there was no way for an accidental short circuit. A pair of spade connectors were used to access the power pins. These could be inserted into the power slots to access the conductive terminals. The spade connectors were fastened to a 3D-printed bracket and glued in place. The bracket was designed by user KlimukVI on grabCAD and matched the battery slot on the DeWalt power tools [23]. This resulted in a safe and easy battery socket where the battery could be removed and "hot-swapped" without fear of accidental electrocution.

5.2.2 LVD - Low Voltage Disconnect

At first, it was assumed that the DeWalt battery had a built-in BMS module. However, it was later discovered that this was not the case and that prolonged usage without charging could put the battery at critically low voltage levels, as stated in section 4.7. Therefore the group decided it was necessary to design and implement an LVD. Due to the fact that this discovery was made late in the project, emphasis was placed on using components that were available to the group in Trondheim. Furthermore PCB production took place in Trondheim at *Elektronikk og prototypelaboratoriet* at NTNU.

The LVD was designed around a NE555 timer, and a non-latching relay was used for connecting and disconnecting the other subsystems from the Power System based on the battery voltage.

The 555 timer had two internal comparators connected to pin 6 *Threshold* and pin 2 *Trigger*. The voltage on these pins, referred to as the supply voltage of the NE555, was set to the upper and lower bounds for the output pin of the 555 timer, high or low. The use of two voltage bounds prevented oscillation of the output when the battery voltage lied on the threshold. Oscillations would occur when the battery voltage increased as a result of disconnecting the load. This would cause the LVD to reconnect the load, thus causing the voltage to drop again. Hence, it will oscillate between connecting and disconnecting the load.

When the battery was at a safe voltage, the output pin of the 555 timer was low. This output pin was connected to the *negative* relay side. When this output signal was low (0V), the output terminal of the 555 timer functioned as a current sink, and there was a voltage drop across the relay, causing it to switch and thus connect the system to the battery.

The LVD switching of the relay happened when the voltage fell below the lower threshold value, causing the output pin of the 555 timer to go high. When the output pin of the 555 timer was high, there was no voltage drop across the relay, and it switched back to its default state, causing the system to be disconnected from the battery. A switch was attached to the LVD power source, allowing for the implementation of an external power switch. See appendix 1 for schematic, PCB and simulation.

5.2.3 LM2596 Module

The central part of the Computation subsystem was a Raspberry Pi 4B. The operating voltage was 5V and could not be powered by the 18V power supply directly [26]. To get the correct voltage, an LM2596 module was used. The LM2596 was a *Step-Down Voltage Regulator* IC capable of delivering 3A at a switching frequency of 150Hz [24]. A potentiometer on the module could be adjusted to set the desired output voltage of 5V.

The choice of the module was dictated by availability. The LM2596 module could be purchased from the student organization *Omega Verksted* without having to wait for delivery. Omega Verksted did not provide any documentation on the module. Several resellers on Amazon and AliExpress sold the same module, but these also had poor documentation. To ensure the module would work, it was used during the testing of the system. When the power delivery to the Raspberry Pi was deemed to be reliable, the decision to use this module in the final measurement system was made.

5.2.4 XT60 Connector

As the system was intended to be modular, it was necessary for the components to be able to connect and disconnect to the power source without the need for soldering. To achieve this, XT60 connectors were implemented. The XT60 connector was a socket-style connector that could handle up to 60A of current [25]. No part of the system would ever demand 60A of current, but the connector was chosen because its size makes them easy to handle.

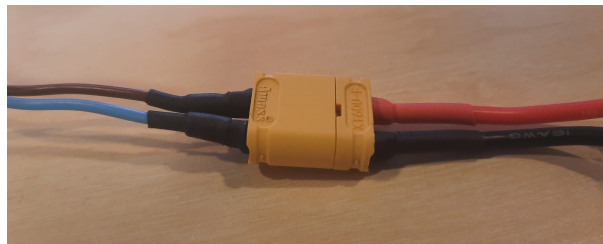


Figure 8: XT60 Connector

New components could be fitted with XT60 connectors and connected to the Power System without needing tools.

5.3 Computation

The Computation subsystem handled the system's operation, obtained and stored measurements, and established a connection with the 5G cellular network. As the main computer, a Raspberry Pi 4B was used. In addition, a Pepwave BR1 Pro 5G cellular modem was used for cellular connection.

5.3.1 Raspberry Pi

The Raspberry Pi was a small, single-board computer [26]. It was programmed with a script that automatically started the test procedure once connected to power. There was also a button and screen for the user interface and a remote *graphical user interface* (GUI). This way, the Raspberry Pi could work both autonomously or through human interaction.

For communication with the modem, an Ethernet cable was used. This was connected to the Raspberry Pi's Ethernet port. This also gave internet access to the Raspberry Pi, which it got from the cellular modem [26].

5.3.2 Pepwave BR1 Pro 5G

To enable measurements of the 5G cellular network, the Pepwave BR1 Pro 5G from Peplink was used. The Pepwave was a 5G modem with a built-in Wi-Fi router and GPS antenna. It was marketed for use in vehicles akin to buses, recreational vehicles, and boats.

Because the Pepwave was designed for use in vehicles, it had a slim profile and was lightweight. In addition, it had a wide range of acceptable operating temperatures and humidity levels. This made it perfect for light weight, mobile system for use in many different locations [22].

The modem had an operating voltage of 10V to 30V and could be used with a wide variety of power supplies. The nominal power consumption of the modem was 8W, and the maximum was 19W [22].

The Pepwave had four antennas for cellular connection. These were connected with SMA connectors. Additionally, the modem had two sim card slots which enable switching between two different telecom operators without needing physical access to the modem [22]. However, it must be noted that the Pepwave could not utilize both sim cards simultaneously, and the procedure of changing operators took approximately 60 seconds. Nevertheless, the possibility of a dual sim card was still an added benefit and is perfect for use in Norway, where two significant operators dominate the market.

For local connectivity, there were two Gigabit Ethernet LAN ports and one additional WAN port [22]. The LAN port was utilized using an Ethernet cable for high-speed communication with a Raspberry Pi 4B.

Another significant advantage of the Pepwave was its included GPS module. This was an essential requirement for geographical mapping. The package came with its GPS antenna that used an SMA connector [22].

5.4 Antenna

The antenna subsystem included the hardware used for transmitting and receiving. Therefore, the subsystem needed a 5G cellular connection and GPS connection. In addition, it needed a suitable transmission line to transfer the received signal to the Computation subsystem.

For a wireless cellular connection, four OMIN-404 Maritime Antennas were used. In addition, the Pepwave came with a GPS antenna that was used for GPS reception. For the transmission line, four RG223 coaxial cables were used.

5.4.1 OMIN-404 Maritime Antennas

One of the goals of the project was to perform measurements at sea. The antennas used for the system needed to handle the harsh environment in Trondheim Fjord. The safest choice was to use maritime antennas designed for use in boats that could handle contact with a large amount of seawater.

To achieve this, the OMIN-404 Maritime Antennas from Poynting were used. The OMIN-404 was compatible with frequencies in the range of 3.5GHz to 3.8GHz, which is the frequency typically used for 5G [27]. The gain at these frequencies was given at approximately 1.5dBi in the antennas' datasheet [28]. The antenna had an omnidirectional radiation pattern with a near-perfect circular radiation at lower frequencies. The pattern took on more of a torus shape at a higher frequency. At around 3.6GHz, the radiation pattern became an asymmetric torus with a 3dB difference between the front and backside. Figure 9 shows a 2D view of the radiation pattern at higher frequencies. The antennas used n-type connectors [28].

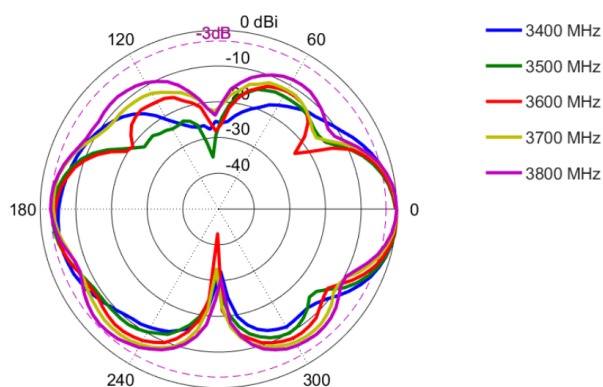


Figure 9: Radiation pattern for OMIN-404 [28]

Four RG223 coaxial cables that were terminated on one side with an SMA connector and on the other with an n-type connector were purchased. The cable length was 1m, and the loss of the cable was 1.05dB at 3.0GHz.

5.4.3 Total Gain

The total gain of the cellular receptor to the system was calculated by taking the sum of the antenna's gain and the cable's loss.

$$G = 1.5dBi - 1.05dB = 0.45dB \quad (7)$$

- G : Total Gain

Because the gain was reasonably low, it was neglected in the 5G measurements. The reason behind this choice is presented in section 9.2.

5.4.4 GPS Antenna

The quality of the GPS signal was not of significant consideration to the project. Therefore the antenna that Pepwave was used. It was designed for use with the 1575MHz [22]. The GPS antenna was reliable and accurate enough for this project.

5.4.5 Antenna Tower

The antennas had to be mounted in a specific way to get excellent reception with no immediate blockage. To achieve this, a simple *antenna tower* was developed. The goal of the antenna tower was to get the antennas as high as possible and optimize the spacing between them.

The tower's main structure was built using a broom handle, vertically mounted and functioning as a mast. Another broom handle was cut into two shorter pieces and mounted horizontally at the top of the mast. These functioned as mounting poles for the antennas. On each side of each mounting pole, an antenna was mounted using a 3D-printed bracket. The antennas were mounted with a given spacing, providing horizontal separation. The mounting poles were also connected to the mast with a given spacing, providing vertical separation.

To calculate the optimal distance between the antenna, it was assumed that the ideal placement would be a multiple of half a wavelength. In Norway, the telecom operator is committed to delivering 5G on the 3.6GHz-band [27]. Therefore, this was used as the base for the calculation. Using formula 1 from theory section 4.2.

$$\lambda = \frac{c}{3.6 * 10^9 Hz} \approx 8.3cm \quad (8)$$

- λ : Wavelength
- c : Speed of light

Half a wavelength equaled 4.15cm. However, this was a bit shorter than what was practical. Therefore, a fourth multiple was used, resulting in a distance of 16.6cm between the antennas.

Because the antenna had an asymmetric, torus-shaped radiation pattern, the four antennas were mounted in four separate directions with 45° angle between them. That way, the sum of the radiation patterns would be near isotropic.

5.5 Encapsulation

The goal of the project placed a lot of emphasis on encapsulation. Firstly, it needed to be mobile and therefore have a small form factor. Secondly, it needed to protect the system from the harsh weather. Thirdly, since the project's final goal was to mount the system on a USV to do measurements in Trondheim Fjord, the encapsulation had to be waterproofed to protect the internal electronics. Finally, it was also an internal goal to make the design modular to accommodate different operations in the future. The design went through numerous iterations, but the final version is presented here.

5.5.1 3D Printing and Limitation

It was decided to use 3D printing as the manufacturing method early on. FDM 3D printers were available for use, and the final product could be produced within a short amount of time. However, there were some downsides to 3D printing technology. One of the most significant restrictions was that the available 3D printers all had a build platform of 220x220mm. This put a maximum limit on the size of the "footprint" of the encapsulation. In addition, an overhang greater than a 45° angle needed support structure. To reduce the printing time, it was beneficial to avoid this. Lastly, 3D printing did not guarantee a watertight seal, and the final product needed to be inspected for leakage. Design choices were made to negate the limitations mentioned above.

All design files were created in CAD using Autodesk's Fusion 360 and printed using PLA plastic.

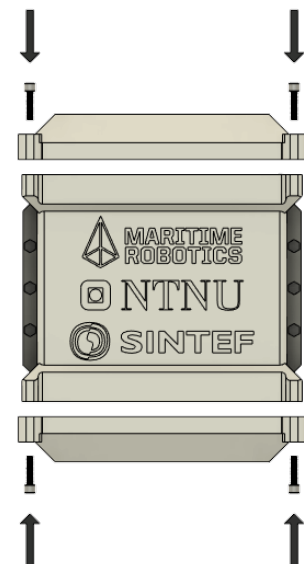


Figure 10: Encapsulation assembly concept

5.5.2 Outer Shells

The other walls of the body were split into three parts. A *main body* considered the large majority of the mass and a top and bottom functioning as *lids*. The main body was designed to fit all the components of the system. It was designed as a hollow cube without a top and bottom, which allowed it to be printed without support. The top and bottom parts were printed separately and needed minimal support structure. After manufacturing, the three parts could be combined to create a sealed box containing all the components.

Printing the bottom lid with the main body would have been possible as it would not have needed any support. The benefit of splitting the encapsulation into three parts

was that it would be possible to access the components from both sides of the box. This would increase user-friendliness and make repair and maintenance easier. A visualization of the encapsulation assembly is shown in figure 10. The size of the components dictated the size of the encapsulation. The two largest components in the system were the Pepwave modem and the DeWalt battery. To take advantage of the Pepwaves slim profile, it was turned on its side and placed next to the battery. The combined width of the Pepwave and the DeWalt battery determined the total width of the encapsulation, while the height and depth were given by the length and width of the Pepwave. All other components were placed above the battery filling the empty space created by the large footprint of the Pepwave.

5.5.3 Threaded Inserts

Threaded inserts connected the top and bottom lid to the main body and closed the box. Threaded inserts are metal inserts that are threaded on the inside. These can be embedded in the model to give it threaded holes for screwing machine screws into.

To securely fasten the top and bottom lids, RS Pro M5 Brass inserts were used [29]. The insert was heat-inserted into pre-modeled holes in the main body. The benefit of using threaded inserts was that the screws could be removed for disassembly and reassembly without causing wear and tear to the soft PLA plastic [20].

5.5.4 Water Seal



Figure 11: Cross section of the water seal

To ensure that the encapsulation could handle being exposed to the environment, especially being mounted on a USV, a *water seal* was developed. The water seal consisted of a sunken canal along the top of and bottom of the main body of the encapsulation. The canal had a depth of 4mm and a width of 3mm. The bottom of the canal was filled with silicon sealant, intended for use in bathrooms, and left to dry. The lids were modeled with an extruded canal matching the main body's one. However, this had a depth of only 3mm, leaving a 1mm space between the sunken and extruded canal. The now dry silicon sealant occupied this space.

When the lids were connected to the main body, the extruded canal would be inserted into the sunken one, compressing the silicone sealant. The sealant would be pressed against the walls creating a watertight seal. The machine screws could be tightened to ensure enough pressure was applied to the seal. Because the sealant was left to dry completely before assem-

bling, the different parts would not fuse together. Figure 11 shows a cross-section of the water seal.

One possible point of failure was the screw holes. These could not be sealed with silicon as they would tear through it after extensive use. To negate this problem, the water seal was modeled on the inside of the screw. That way, water could leak through the screws without compromising the water seal.

5.5.5 Rail System

To mount the components inside the encapsulation, a simple *rail system* was developed. First, each part was mounted on a *mounting plate* using PCB mounts. The mounting plates could then slide onto rails modeled on the inside wall of the encapsulation. The rails were locked at the bottom. The top lid would serve as a locking mechanism closing the rails from the top once screwed in place. Figure 12 demonstrates the Rail System.

This design added several benefits to the system. For example, the placement of the different components could be changed without redesigning the encapsulation, or components could be swapped out completely if needed. In addition, one extra unused rail was modeled into the encapsulation for future expansions.

Any component small enough to fit inside the encapsulation could be mounted to a mounting plate and used in the system. In later iterations, it would be possible to fill the entire inside wall with rails and only use the ones suitable for the given project, but this was not done to save print time and material.

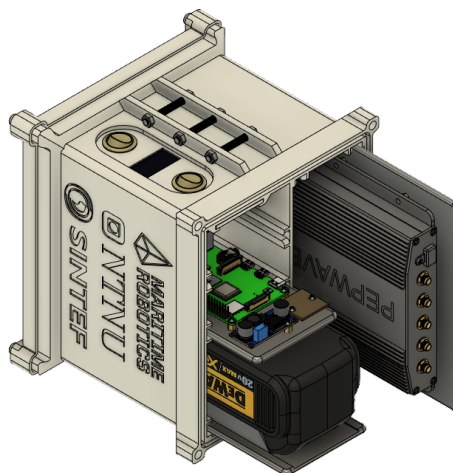


Figure 12: Demonstration of the Rail System

5.5.6 Packing Nipple

The cables from the antenna to the Pepwave modem needed to pass through the encapsulation wall. This would mean bypassing the water seal and creating a potential point of failure. To avoid this, the holes for the antenna cables were fitted with packing nipples. The ones used had an IP rating of 68 [32]. This would ensure that no water could enter the system through the antenna cables.

5.5.7 Expansion Slots

The last feature added to the design was an expansion slot in one of the walls. This was an open slot that could be inserted with a plate. The plate could be changed out and replaced as needed.

The idea was that the plate could be modified to fit different use cases depending on the use cases of the system.

To ensure that the expansion slot would not leak, it was fitted with the same concept as the water seal: A canal filled with silicone sealant and a reverse canal on the inserted plate.

5.6 User Interface

It was decided that the system needed some form of user interface even though it was designed to be operated remotely. Most notably was the need for feedback for the user to inform if the system was receiving power and what state it was in. It would also be practical to turn it on and off without having to use a remote connection.

To build this, the expansion slot was used. Following, a mounting hole for an OLED SH1106 screen was added, and a thin, transparent plastic screen was glued in place [31]. The screen was sealed with silicone sealant. For user input, mounting holes for two buttons were fitted to the expansion slot. One of the buttons was to be connected to the LVD and would function as a power switch. The other button was connected to the Raspberry Pi and would switch the system from active to non-active. The buttons used had an IP rating of 65 and were selected based on availability.

Once the expansion slot was finalized, the top lid could be inserted into the main body and locked in place.

6 Software

The software used in this project was built specifically to meet the assignment's requirements. The different parts are presented below.

6.1 Base Shell

For the PC and the Raspberry Pi to communicate, mainly two methods were used for accessing and controlling, dependent on the task. During the development, *Virtual Network Computing* (VNC) was used because of the ease of maneuverability and management of files. The whole screen of the Raspberry Pi was mirrored to the connected PC, essentially making the Raspberry Pi a remote virtual machine. This made file structuring and debugging along the way more practical, whereas a *Secure Shell* (SSH) connection was only used when a measurement session was to start. This will be described in more detail in the Method section 7.1.3 regarding the measurement session start. Furthermore, to sync the files with one another, a versatile file copying tool named *rsync* was used. All the aforementioned functions were accessed through the base shell on either the Raspberry Pi or the PC.

6.2 Raspberry Pi

A Raspberry Pi 4B was used as the computing device of the measurement system while running Raspbian OS. Furthermore, the desired python script was automatically launched after the Raspberry Pi was powered and an internet connection was established. Finally, the python script entered a state event *idle* to wait for further instructions, either through hardware or a GUI input from the user.

6.3 Pepwave Software

All information regarding the parameters of interest was available and displayed on the Pepwave admin panel. However, this was not a practical solution for an autonomous measurement system.

The tools used for managing the retrieval of measurement parameters are described in the following sections.

6.3.1 SNMP Connection

To retrieve the relevant information and parameters from the Pepwave modem, an SNMP analysis of the respective MIB was needed to acknowledge the OIDs and perceive the data values. The manufacturer provided Pepwave's MIB in their database ¹, and the navigation could be done using the *MIB Browser* Software. Furthermore, in the python script, by gathering all the individual OIDs and making a class, instances of multiple objects could be made for easy sorting and further scalability. In other words, all the different information was stored into an inherited object that the user could make. Another benefit of doing so was that the user could easily access it in an informative way by calling the object and its parameter during the test period.

6.3.2 GPS Forwarding

On their own, the data points are worthless without any knowledge about where it was sampled. However, with the Pepwave's included GPS module, the position could be fetched in parallel with the other relevant information from the modem. This was done by port forwarding the GPS coordinates to the desired IP address with the TCP protocol within the admin panel itself. On the other end, the Raspberry Pi listened to port 8080, which was the TCP port, and received NMEA-GPS data in the format of GPRMC, which later was processed in the script to obtain the latitudinal and longitudinal components.

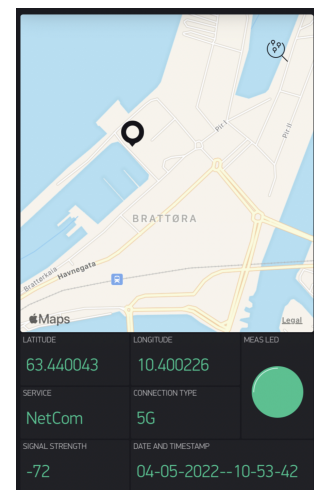


Figure 13: GUI in Blynk

6.4 GUI - Blynk

During the test period and data collection, a GUI was needed for the user to receive live feedback from the system or, in other words, to know if the system was running and responding at any given time. By incorporating *Blynk*, an easy-to-use IoT platform, the problem could be solved with many additional benefits. By connecting to the Blynk server through the mobile application or within the Raspberry Pi module itself, desired data packets could be sent and accessed through the internet. Furthermore, Blynk provided the possibility to quickly build interfaces for monitoring and controlling multiple systems with various actuators and indicators. In addition, different users could access the Blynk control panel simultaneously, thus making it possible to monitor the measurement activity live from various locations.

¹The manufacturer provided an extraordinary MIB because the "UTRAN Cell ID" OID was found to be missing during the thesis. Thanks to the quick act from Peplink Support, it then became possible to acknowledge the base tower to the respective data sample.

Figure 13 shows the custom-designed GUI for the project with multiple indicators and an interactive map. The different widgets were communicated through the python script, where data and instructions were sent if an internet connection was present. The measurement indicator (MEAS LED) showed if a measurement session was on-going, where different modem parameters such as signal strength and connection type were displayed.

Another helpful feature with the Blynk GUI was that if the app happened to disconnect, it either meant that the python script stopped working or that the measurement system was out of reach for internet coverage. With this being a prototype, by knowing the position of the USV from external sources, e.g., Maritime Robotics' sensors, it was possible to tell if an unfortunate event had occurred.

6.5 Python

The script created to manage the connection and store the measurements was written in Python. This is described in closer detail below.

6.5.1 Important Libraries

External Python libraries that were central to the script's functionality were *easysnmp* among other libraries, which made it possible to establish an SNMP connection between the cellular modem and the Raspberry Pi. Alongside *schedule* and *datetime*, the script could be executed repeatedly with a desired time interval, thus regulating the spacing between each measurement taken during a trip. Subsequently, the *os* library provided a remote way of using the Raspbian operating system, which was convenient for creating folders and files locally on the Raspberry Pi SD card.

To simplify and automate the process of creating KML files, *simpleKML* was used in Python. The library, *simpleKML*, was chosen due to its extensive documentation that can be found at simplekml.readthedocs.io.

6.5.2 Creating Data Files

Following the sampling of every data point during a measurement session, it had to be stored in an organized way for processing and mapping. This was done using the above-mentioned *os* library in Python, where text files in Comma-Separated Values (CSV) format were created in different directories, depending on different parameters.

Each session was firstly categorized by which telecom operator was currently in use, followed by a date and timestamp of when the session started. Furthermore, each data

point was differentiated into two possible outcomes where the GPS was either present or not. Because the GPS position of a given data sample is critical to generating a map, it was less practical to gather the data sets with and without GPS in one file. Lastly, the CSV files were created depending on the connection type, e.g. no cellular, 4G or 5G, where relevant parameters were appended to the data set in the following order: *timestamp, latitude, longitude, RSRP, RSRQ, SINR* and *CellID*.

```
./measurements/NetCom/29-04-2022-14-10-23/GPS/5G.csv
```

The directory above is an example of how the files would be organized and structured, being automatically created by the script. By reading the directory, it was possible to recognize that the main folder *measurements* contained all the collected data values. Furthermore, the measurement session started on the 29th of April 2022 at 14:10:23, while the modem was connected to the NetCom (Telia) network. In the CSV file containing different parameters, there was also a GPS position to be found whilst being connected to 5G.

6.6 KML - Data Visualization

It was decided that the data obtained from the measurements would be represented on a 3D Earth map (Google Earth Pro). For this, KML was used.

To visualize the data, two methods were employed. The first method was referred to as *trip visualization*, and was the simplest of the two. In short, it visualized the measurements obtained when moving from point A to point B. A point on the map was assigned a color representing its measurement value in a trip visualization. Furthermore, the direction of the heading was indicated and visualized. The base station responsible for the measurement was indicated by a line between the location of the measurement and the location of the base station.

The second method, *grid visualization*, served the purpose of visualizing the data obtained when *gridding* an area, essentially moving back and forth over an area and obtaining multiple measurements within the vicinity of the exact location of previously obtained measurements. A grid cornerstone was used to ensure consistently aligned grids when visualizing and comparing different sets of data. This grid cornerstone was geographically located at Maritime Robotics' headquarters at Brattørkaia in Trondheim. All grid sections were the same size across all measurements. Thus mapping of a larger area would only result in more grid sections, not larger grid sections. To visualize the data, the grid section was assigned a color based on the logarithmic average of all measurements taken within the grid section's coordinates.

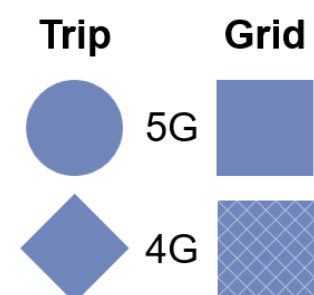


Figure 14: Trip and grid differentiation

Differentiation of 4G and 5G measurements in trip visualization was as simple as using circles for 5G, and for 4G, a *rhombus* or *diamond* shape was used. In grid visualization, 4G was a square with diagonal, straight, parallel, and transparent lines, whereas 5G was a homogeneously colored uniform square. See figure 14 for example on 4G and 5G differentiation, and figure 15 and 16 for the two different visualization techniques.

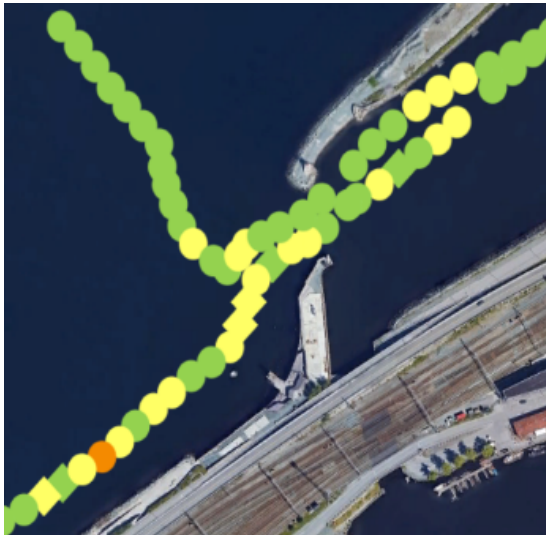


Figure 15: Trip visualization



Figure 16: Grid visualization

6.6.1 Quantification and classification of Data

To visualize measurements and to have the ability to compare them, it was necessary to quantify and classify them. Poynting Technologies provides a *cheat sheet* with threshold values as well as color for RSRP, RSRQ, and SINR. The cheat sheet that would be used is presented and referred to in the theory section of this report [12].

6.6.2 Trip Measurements

In this subsection, a more detailed description of how trip measurements are visualized will be presented. As stated previously, it is the simplest form of measurement visualization used in this project and is used to represent data obtained from a trip.

Firstly the GPS coordinates were extracted from the CSV file. The measurement data was then the *third variable* of the plot, shape, and color of a PNG was used instead of using the z-axis, or altitude, as a third dimension. Based on the type of measurement and the value of said measurement, an algorithm selected the shape and color of the PNG used to visualize the measurement at its respective GPS coordinate. Once all the measurements had been visualized, a descriptive legend was then created as a PNG to be added as a screen overlay in the KML file that was to be created. Given a

measurement that states *no coverage*, a simple PNG with a cross was used in place of a colored PNG.

6.6.3 Base Stations

One of the predominant factors regarding the strength and quality of the cellular connection is the position of the base station relative to the point of measurement. In addition to free-space path loss, other factors such as blockage and shadowing can impact which station the link is established with.

As each measurement had a corresponding base station that also had a specific GPS location, it was then possible to show what base station was responsible for what measurement. The website cellmapper.net was of great use. Where the location and eNB IDs of base stations for the major telecom operators, Telia and Telenor, as well as ICE, can be found. This database was used to map the base stations in a separate CSV file along with the GPS coordinates. A line could then be created between the location of the base station and the location of the measurement.

To create this line, first, the CellID was obtained from the data set, then the CellID was converted to an eNB ID. This conversion was as simple as dividing by 256 and discarding any floating-point. Now that the eNB ID to the specific measurement was obtained, an algorithm checked if there was a base station with the same eNB ID. If there was, then the GPS coordinates of the said base station were acquired. A line was then created from the location of the base station to the location of the measurement using the obtained GPS coordinates for each.

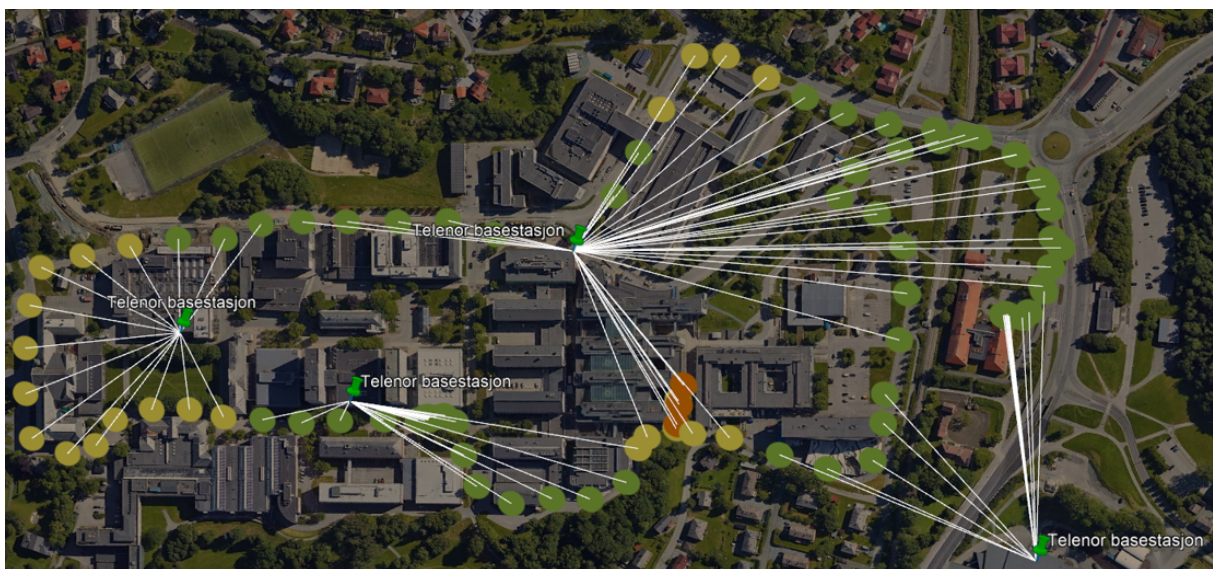


Figure 17: Lines based on nearest cell tower, not real data

6.6.4 Heading

The direction of travel was relevant information when an examination of the visualized data would be performed with regards to cell tower switching and coverage. Therefore, it was desirable to visualize the direction of travel.

As the relevant data sets were split into four separate files, one each for LTE, LTE-A, 5G, and *No cellular*, they had to be combined and sorted before any heading indicators could be created. Each data set had a timestamp. This timestamp would determine the sorting of the combined data set.

First, the four data sets were combined in no particular order. Following, the combined data set was sorted by time, which was initially in the string format "dd-mm-yyyy-hh-mm-ss". This conversion took place by converting the aforementioned string format to an integer format in the form of *hhmmss*. Now that a sorted data set was obtained, each consecutive GPS coordinate represented the location of the following known position. Knowing the current and next positions, one could find the angle between them using the inverse tangent function. The distance traveled in meters along the latitudinal and longitudinal represent both catheters of a triangle. Here it was essential to operate in a unit such as meters due to the fact that latitude stretches from 0° to 90° , whereas longitude stretches from 0° to 180° , meaning they were not "1 to 1" by nature.

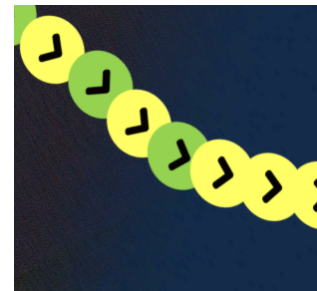


Figure 18: Heading visualization

When the angle between the current and the next point was found, one could move on to visualizing this angle. Similar to how the data was visualized with circular PNGs, the heading was visualized with an arrow-like PNG, and a built-in KML attribute controlled its rotation.

6.6.5 Grid Measurements

When obtaining 5G data within a larger set area, the aforementioned method for visualization was sub-optimal, as multiple measurements might have been taken at the same location both at the same time and at different times. As a result, large amounts of overlap could occur. Therefore the implementation of another method was necessary.

The method to be used for all future visualizations of such a measurement method was the previously presented *grid visualization*. The idea was to take the average of all measurements taken within a set geographical area, a grid section, and then alter the visual characteristics of said grid section based on the measurement type and logarithmic average value.

6.6.6 Grid Visualization - Processing

To calculate the average value of all measurements taken within a set geographical location, an algorithm was developed. This algorithm essentially created a 2D array to contain the data, where the rows and columns represented the respective x-axis and y-axis. Furthermore, movement along the x-axis represented longitudinal movement, whereas movement along the y-axis conversely represented movement in the latitudinal direction. This 2D-array was created based on the minimum and maximum latitudinal and longitudinal values, where they defined the boundaries for the grid measurements that were to be visualized. Once the boundaries were set, a *grid size* was used to partition the boundary into multiple sections. The dimensions of the grid were the same as the dimensions of the 2D-array, where the minimum coordinate values represented the lower left-most index of the matrix, conversely it was the higher right-most index for the maximum coordinates. Now the algorithm iterates through the 2D-array and checks the measurement data to see if there was any corresponding coordinate values, if there was, the measurement value was gathered and the amount of measurements in said grid section was counted, when the whole measurement matrix has been iterated through, the averages were calculated, and a matrix containing the processed data was created.

6.6.7 Grid Visualization - Presenting

The processed data was now ready to be visualized in a grid pattern, this was a procedure that ended up having multiple revisions, the revisions will be discussed in the following paragraphs.

The first attempt at creating a grid visualization involved the use of the native built-in function in KML that is *polygons*. With polygons in KML one can create geometric shapes in 2D- and 3D-space, style them with colors and transparency, and most importantly control their geographic placement based on latitudinal, longitudinal and altitude values. Again, the data matrix was iterated through and this time the indexes control the coordinates. Starting from index (0, 0), representing the minimum coordinate values, it started the creation of the polygons, the grid size was a constant linear unit and moving one gridsize corresponded to moving one index. Thus for every index in the matrix that was to be visualized one needed only to move one gridsize in coordinates, on the map. Square 2D-polygons were drawn out and given a color and transparency based on the value found corresponding to the indexes in the matrix.

Upon review by the group, and supervisors, there was a unanimous opinion that the white lines apparent when using the KML polygon grid visualization method was deemed undesirable. Therefore this was not the method that was settled on, and another method was developed.

The second method used, bearing much resemblance to the circular method, was to

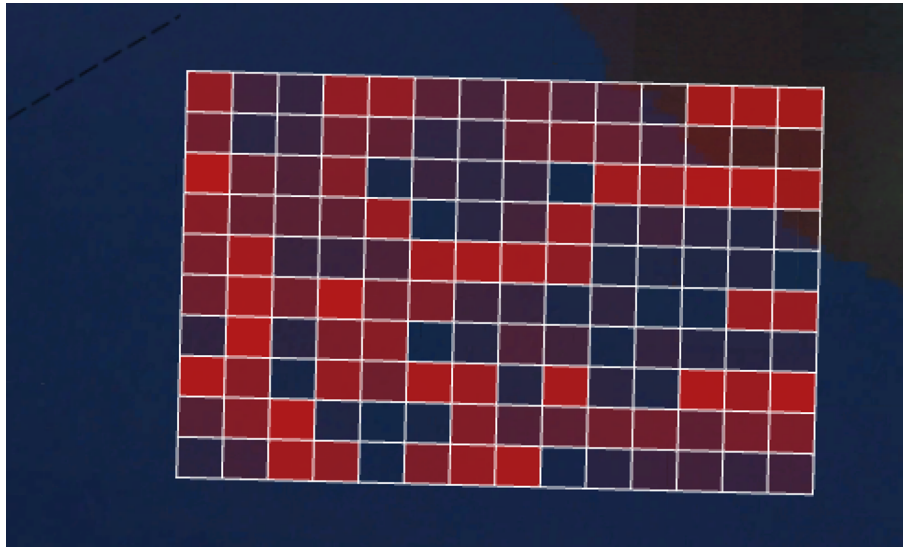


Figure 19: KML polygon grid visualization

geographically place a set of colored PNGs onto the map, similarly to trip visualization. However this time with square PNGs, each square PNG represented its own grid section, and its color represents the logarithmic average of all measurements within the coordinates of said grid section. The iteration and coordinate algorithm was the same as used in the polygon method, but instead of assigning the polygon borders, the image borders are assigned. For representing the data values, a colored image was selected instead of assigning a color to a polygon. The algorithm that selected the colored image was the same as described for trip visualization. Representation of areas where no data was available would be done by placing a dark and transparent PNG in said grid section. All in all, it was very similar to the polygon method previously described, and it shared similarities with the trip visualization method.

6.6.8 Grid Commonality

Ensuring a common grid was necessary for making future qualitative comparisons across different measurements possible, the process in doing so will be described here.

Firstly a common *grid* was made. This grid had Maritime Robotics office at Brattørkaia 11, 7030 Trondheim, or in coordinates: (63.44027859540668, 10.399713079298035), as the *cornerstone*. All future grid patterns for this project would be based on that set location, the purpose of this was to create a consistent grid that could be used for all measurements and prevent miss-alignment of data visualization from different data sets. The minimum and maximum values of the above-described grid visualization had to be equal to Maritime Robotics' HQ coordinates, or the sum of the HQ location and a multiple of the grid size. The shortcoming of this method was that any data that fell on

the outside of the minimum and maximum coordinates of the common grid would be lost. However, this was the solution chosen to go with, as it was deemed unproblematic that minor data was discarded.

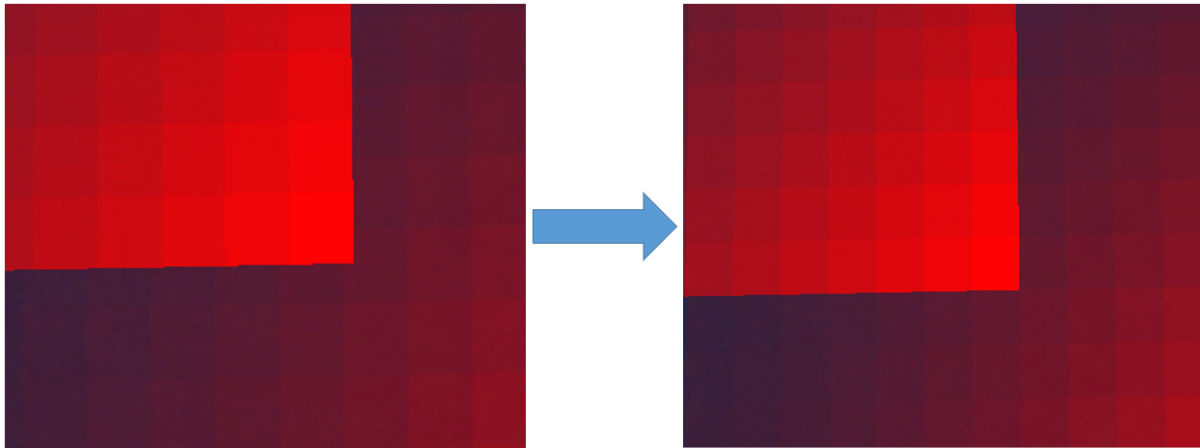


Figure 20: Visualization of grid commonality

7 Method

The following chapter describes the method used for testing the system. It describes the procedure and safety measures employed to reduce the risk of failure or accidents.

7.1 Risk Analysis

The system was intended for use in close contact with water using a battery with a relatively high voltage. This introduced a considerable risk to the project participants and the equipment. In addition, part of the project was conducted during the local Covid-19 restrictions. Therefore, to reduce the risk of accidents that would either harm the well-being of the participants or render the equipment unusable, three risk analyses were conducted at three different stages of the project.

The risk analyses were completed using standardized forms. The forms would identify unwanted events and set a graded value on the chance of the event happening, and the magnitude of the consequence should the event happen. If an event was calculated to have too great of a risk, actions were put into place to either reduce the chance of it happening or decrease the consequence if it were to happen.

The work was divided into two parts: *Lab Work* and *Field Work*. The two first risk analyses contained the risk related to the Lab Work, while the last risk analysis contained the risk related to the fieldwork. See attachment 18, 19 and 20 for the complete risk analysis forms.

7.1.1 Lab Work

Lab Work included all the work done building the system and testing it before it was ready for in-field measurements. The *risk analysis form* for the Lab Work was downloaded from the Norwegian *Arbeidstilsynet* web pages. The risk analysis was performed twice and early on in the project. After the second analysis, it was concluded that all possible unwanted events during Lab Work were taken into account. Therefore, the subsequent Risk Analysis would occur right before the Field Work.

7.1.2 Field Work

Field Work included all the work done after the system was completed and ready for in-field measurements. It was mostly done outdoors. Field Work included two types of tests. The first one, called *Bike Test* was done by placing the measurement system

in a backpack and using a bicycle to take a trip around Gløshaugen. The second test was conducted in the fjord using one of Maritime Robotics' USVs.

These tests introduced the potential for significant risk for both the participants and the equipment. Because the risk was much higher during Field Work than Lab Work, the risk analysis was completed in cooperation with SINTEF and with the guidance of the project's supervisors.

7.1.3 Safety Procedure

In some cases, it was concluded that there were no sufficient actions that could be done to reduce the risk permanently. Therefore, a procedure was established to further lower the risk of accidents. In addition the procedure was developed to ensure accurate measurements and repeatability.

The procedure was to be followed before and after every test. The main focus was to ensure battery safety and that the software was running before fully assembling the encapsulation.

7.1.4 Start and Stop a Measurement Session Procedure

1. Check if the battery is fully charged before proceeding to the next step
2. Connect the battery to the power slot in the encapsulation
3. Go to the admin panel and check if cellular and GPS are both initialized
4. Establish an SSH connection between PC and Raspberry Pi
5. Start the desired python script
6. Check if the OLED screen has the state "idle"
7. Login to the Blynk app and check if a connection is established
8. Switch the button to start a *measurement session*
9. Obtain measurement data
10. Switch the button back to stop the *measurement session*
11. Sync the data files from Raspberry Pi to PC with the rsync tool
12. Disassemble encapsulation and disconnect the battery

7.2 Power System Test

To gain a better understanding for the Power System, it was necessary to test it. There were multiple sources of motivation for testing the system. Firstly, by knowing the power consumption of the system, the group could estimate how long the system could operate on a given battery with its known capacity. Secondly, it was necessary to test if the different subsystems functioned as intended before implementing them into the system. The tests performed will be described in the following sections.

7.2.1 LVD Test

The LVD test was performed on a breadboard with an equivalent circuit due to PCB production errors and miscommunication with *Elektronikk og prototypelaboratoriet* at NTNU. See discussion section 9.6.

Before implementation, the LVD was simulated using *LTSpice* and tested to ensure correct operational functionality.

To test the LVD, it was connected to an adjustable bench power supply. The power supply had a variable output voltage, and it could be adjusted to trigger the LVD. In addition, the power supply had an indicator for total current that was used to measure the total current draw of the LVD.

Results:

Testing concluded that the LVD functioned as intended, when the voltage from the power supply went below 16V, the OUT pin of the NE555 timer went high, causing the relay to switch. Subsequently, when the voltage was increased to 18V, the OUT pin went low, causing the relay to switch back 3.

The LVD had a total current draw from the power supply of $30 \pm 5 \text{mA}$, and at 18V that resulted in a power consumption of 0.5W, when the relay was active. As for the case in which the LVD was powered, but the relay was inactive, essentially in the state of which it was to disconnect the system from the battery, the current draw was found to be less than 5mA.

7.2.2 System Power Consumption

The total power consumption of the system could be estimated by calculating the sum of the power consumption of both the Pepwave and the Raspberry Pi. The documentation for the Pepwave stated a nominal power consumption of 8W [22], and the Raspberry Pi had a maximum power consumption of 5W [26]. The system's power

consumption was thus estimated to be 15W, including overhang, based on rough estimates for each of the components. However, this introduced a significant degree of uncertainty. Therefore, it was necessary to test the system before conducting the field test.

To test the total power consumption of the system, it was connected to the DeWalt battery. The voltage was measured to be 20V, and the system was left to run for approximately 10 minutes. The system needed to be given enough time on the test bench to replicate an extended test run in the field.

Sources of Error: The DeWalt battery did not deliver a stable 18V output but had an output voltage dependent on its capacity. At full capacity the output voltage was about 20V, and the LVD disconnected the battery at 16V. In addition, the system was inside a building, without in-field-equivalent cellular and GPS connection. It was possible that an active connection would demand more from the system than a poor one. Alternatively, the opposite could be true.

Results: The power system had a measured current draw of 500 ± 50 [mA]. At the measured battery voltage of 20V, that equates to a power consumption of 10 ± 1 [W]. Thus the system could be expected to operate for a maximum of nine hours given the DeWalt battery capacity of 90Wh [21].

7.3 Dry Runs

For the same reason as having procedures, dry runs, also known as practice runs, were performed to the measurement system to better understand the behavior during different hypothetical scenarios of the in-field tests. The purpose of the dry runs were to recreate and mitigate different points of failure. By doing so, the risk of the system failing in the in Trondheim Fjord, where it is most critical, was reduced. The following dry run was done to improve the overall system performance.

7.3.1 Run for an Extended Period

The predefined route of the Maritime Robotics' USV was set to drive along the pier of Trondheim and afterward out in the fjord in proximity to SINTEF Ocean's bouy. The estimated duration of the measurement session could therefore be calculated to last between one and two hours. For that reason, the script needed to run continuously for the whole duration. It was also worth knowing if the theoretical calculation of the DeWalt battery was in accordance with the practical usage of the measurement system over an extended duration of time.

Results: The system managed to obtain measurements uninterrupted while running

for 2 hours, and the battery capacity indicator showed approximately two-thirds remaining charge. Therefore, it was possible to assume that the battery would last at least six hours in-field, given the same power consumption as the dry run. Another observation was that the system emitted a decent amount of heat from the encapsulation, mainly from the Raspberry Pi's CPU, but this was deemed to be unproblematic.

7.3.2 No Cellular Test

One of the main concerns with the system was if it would still be able to record data measurements at the very moment that the Pepwave lost cellular connection with the telecom operator. This scenario could happen quite frequently, due to the fact that the USV would drive away from the coastline beyond any cellular coverage. It was as essential to know where there was no connectivity and blind spots, as it was to know where coverage was present. In that event, this scenario could be tested by turning off the cellular connection in the admin panel.

Results: After cellular was deactivated, representing no connection between the modem and the base towers, the system kept running locally, although it lost connection to the Blynk server. An SNMP analysis gave the parameter RSRP a return value of -9999 when no signal was detected, which allowed the system to differentiate between having a signal and being offline.

On the other hand, Blynk failed at first to reconnect after the cellular connection was re-established due to the library being a dated version. This critical problem would have made it difficult to know if the system was still running and obtaining measurements, or if it had failed. Because of the dry run, the problem was discovered and fixed.

Another observation was that the Blynk app used considerable amounts of time to reconnect after the Pepwave's cellular connection was re-established. Essentially this meant that the end-user would get a significant delay before getting an update of the USVs position and parameters. This delay was tested to be around one minute but would not influence the process of obtaining measurements, because the data would be stored locally on the Raspberry Pi.

Knowing how the Pepwave GPS module would behave under different circumstances would also contribute to a better understanding of the system's behavior. For this reason, by testing the sensitivity of the GPS, for instance, inside a building and by physically covering the GPS antenna, one could simulate the situation of the system being inside a tunnel or shadowed by an object while the signal is still present.

Results: For proper functionality, the GPS antenna needed a clear line of sight to the sky to be able to port forward the GPS location. Otherwise, it would lose connection rather quickly, but it would regain the connection roughly within five seconds. The system also separated the data points depending on having GPS position and passed

the dry run. Tilting of the GPS antenna was found to have no significant impact on the GPS functionality and accuracy.

7.3.3 No Cellular and GPS Test

The last scenario that the system must be able to handle was the case where both the cellular connection and the GPS module were to fail.

Results: The system managed to handle the combination of having no cellular and no GPS connection without causing any bugs and errors in the code, and the automated creation of files and directories continued to function correctly.

8 Results

In this chapter, the results from the different field tests on land and in the fjord are presented.

8.1 On a Bike

Measurement sessions were first conducted on land before the system was utilized in the fjord. This was done to ensure that everything worked correctly and to reduce the risk of failure when obtaining measurements. The *bike trip* was set to start at SINTEF's offices at Strindvegen 4, circle around Gløshaugen, and end up at the starting point, lasting for approximately 10 minutes.

Furthermore, two measurement sessions were conducted. For the first one, Telia was used. For the second one, Telenor was used. That way, one would get two separate data sets and could compare the service of the two operators.

The goal of the test was first to check if the system functioned as intended. Following, the system needed to record all the chosen parameters and link them to the correct GPS position for the test to be successful. Finally, the data obtained would be used to generate multiple and different maps, visualizing the results.

The measurement system was placed in a backpack. The antenna tower was set to protrude from the backpack and secured with straps. For transportation, an electric bicycle was used. One of the group's participants equipped the backpack before cycling the predefined route. Figure 21 shows the participant right before one of the tests. The antennas were placed higher above the participant's head during the testing to ensure better reception.

Results: Based on observations from the coverage maps, it was found that Telia generally tended to have a higher signal strength than Telenor. This could be done by having a side-by-side view of the two coverage maps. Figure 22 and figure 23 illustrate this. An important observation was that Telia had a lower threshold with regards to switching back and forth between 5G and 4G, which frequently happened throughout the trip. In other words, Telia chose to maintain a continuously good signal strength, as opposed to maintaining the 5G connection with worse coverage. In contrast, Telenor sustained a 5G connection almost throughout the whole trip, however allowing for a substantially lower signal strength, being as low as under -100dB in some places. The

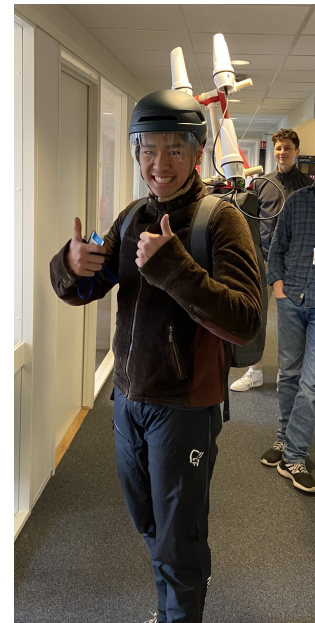


Figure 21: Measurement system ready for testing



Figure 22: Telia RSRP



Figure 23: Telenor RSRP

coloring of the data samples of both the maps was also unison in groups, meaning that the measurements were realistic due to the placement of different the base towers and the shadowing of buildings.

8.2 On an USV

The test in the fjord started at Maritime Robotics' headquarters at Brattøra in Trondheim. Using a pre-programmed route, the USV would autonomously sail in and around a section of Trondheim Fjord. The measurement system was mounted on the USV together with the antenna tower. Once the USV left the starting point, the measurement system would be inaccessible until the test was completed and the USV returned to harbor. Therefore, everything needed to work as intended before starting the trip.

Due to time constraints, there was only time for one trip. This made it so that it was only possible to obtain measurements using one of the telecom operators. Telia was chosen for the following measurement session.

The trip was set to go south-west from Brattøra down along the pier to Ila. Then, the USV would turn northward and venture out toward the SINTEF Ocean's buoy right behind Munkholmen. Lastly, the USV would drive in a grid pattern around the buoy before returning to Brattøra. The trip would take approximately two hours, traveling for a total length of 10km.

The route was selected to satisfy three goals.

- Pass as many base stations as possible, providing data on cell tower switching
- Travel far away from the base stations, testing the reach of the 5G coverage
- Record the cellular connection at the SINTEF Ocean's buoy

Results:



Figure 24: RSRP, 4G & 5G

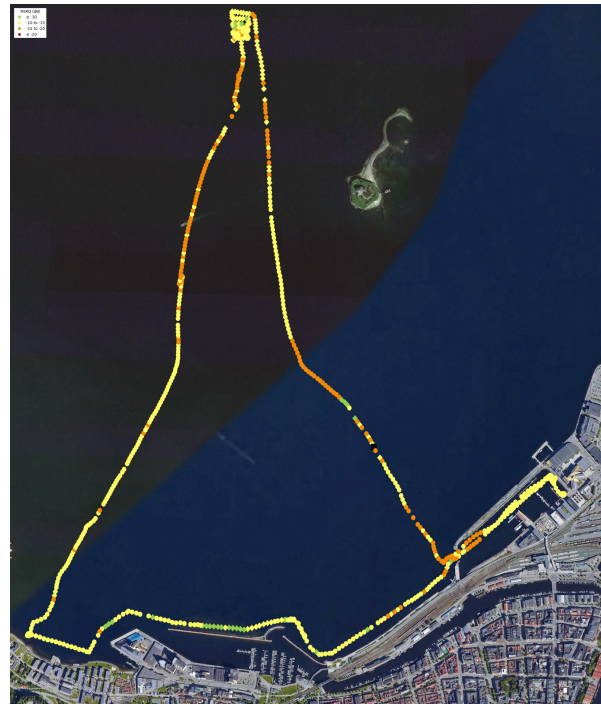


Figure 25: RSRQ, 4G & 5G

Throughout the entire trip in the fjord with Maritime Robotics, Telia maintained a stable RSRP value, alternating mainly between green ($>-80\text{dBm}$) and yellow (-80dBm to -90dBm), for the most part with a 5G connection. However, in some places further out in the fjord, the Telia made the decision to switch to 4G with the purpose of having a better connection and signal strength. This again reinforces the statement of Telia being more susceptible to switching connection types, as observed and mentioned previously. An observation was that even with good RSRP, the connection could be lost, particularly at the furthestmost points, while driving the grid-formatted route. See figure 24 and 25 for visualization of the collected measurement data.

Looking at the SINR parameter and coverage map of the same trip, it was noticeable that the SINR gradually worsened the further out in the fjord the USV went. On the other hand, the map showed a remarkably greater SINR in the areas around Brattøra and Ila, being distinctly green ($\geq 20\text{dB}$). The places with low SINR also correlated with having a lower RSRP. One example is the red areas with a low SINR on the way to SINTEF Ocean's buoy, where the RSRP was significantly worse. See figure 26 and 27.



Figure 26: SINR, 4G & 5G

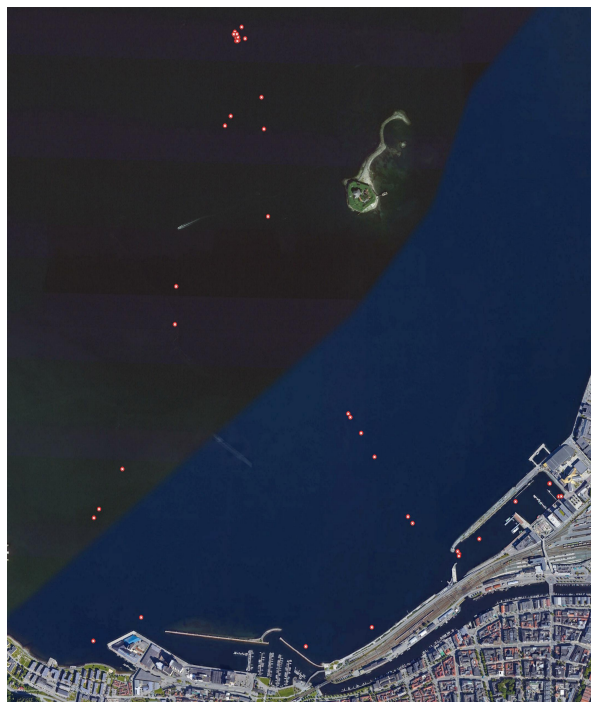


Figure 27: No cellular

It is apparent that in areas where the SINR is low, the connection is often lost.

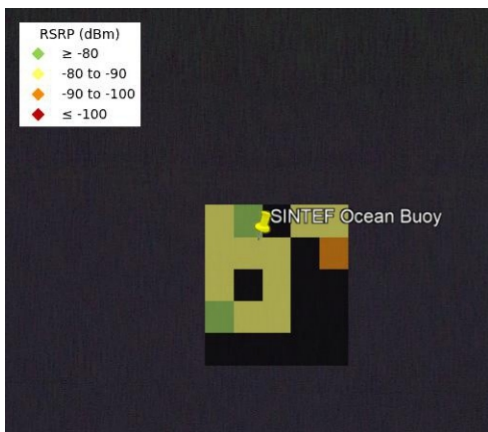


Figure 28: 5G RSRP

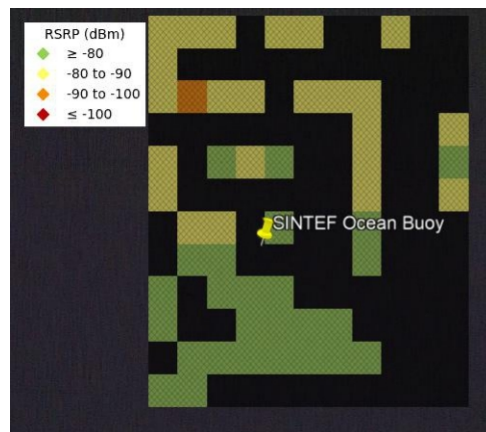


Figure 29: 4G RSRP

Looking at the grid visualizations for 5G and 4G, the first thing that became apparent was that the 4G grid was larger than the 5G grid. A larger area of 4G coverage was measured than that of 5G coverage. Lastly if one directed attention towards the coverage map showing points with no cellular connection, one would notice a large amount of points at the farthest extent from the trip. This is near SINTEF Ocean's buoy. See figure 28 and 29.

9 Discussion

This section contains discussions regarding the final result of the project. In addition to the results from the various measurement trips, the system's quality in itself is also discussed.

9.1 Project Expectations

Generally speaking, the result of the project met all the desired expectations. The final product satisfied the requirements established in the thesis' title and, in some cases, exceeded what would be the minimum satisfactory solution.

Early on, a *Minimum Viable Product* (MVP) was established together with the groups' supervisor. This was done to set a reasonable scope of the assignment and would ensure that the group would have been able to deliver *something*, even if the tests had failed. This MVP only required the system to be able to obtain measurements and store them. It did not demand that the system was mobile, that there was any fieldwork conducted, or that the obtained data had to be visualized. The final product contained all of these requirements and more.

The relative success of the work allowed for creative solutions to be developed. The remote access and use of Blynk to monitor the system gave the group confidence when the system was deployed in the field. Additionally, the encapsulation made the product completely mobile, and exposure to the environment was of minimal concern. Finally, the use of KML in Google Earth for visualization enabled the visualization of the obtained measurements. Thus the whole process of creating a measurement system, obtaining measurements, and finally visualizing the recorded data was complete.

Although the product has room for expansion, it still bears indications and resemblance to a final product. Given the short time frame and allocated resources, it would not have been reasonable to expect further product development beyond what has already been achieved.

9.2 Gain Adjustment

The gain of the system was known, and therefore it would be possible to adjust the RSRP, RSRQ, and SINR. However the group decided to neglect the gain and used the values as recorded. This was done due to multiple factors. Firstly, the group lacked knowledge regarding the calculation of the measurement parameters, performed internally in the Pepwaves firmware. Secondly, relevant parameters in calculating the output values remained unknown. Thirdly, the gain of the noise and interference also

remained unknown. Lastly, the gain was considered to be of a reasonably low value, resulting in minuscule increases in signal strength.

9.3 Use Cases

With the results of the system being realistic and practical, it is now possible to discuss the correlations between different parameters and make speculative conclusions. This is not within the scope of this thesis, but some of the key observations made based on studying the coverage maps were:

1. Telenor and Telia switched to 4G at different thresholds, Telia being more susceptible to alternation
2. The coverage maps taken on land showed a remarkable correlation between signal strength and the position of the connected base tower
3. The measurement session was taken in Trondheim Fjord with Maritime Robotics and had a relatively stable RSRP throughout the trip. However, the SINR suffered greatly as the USV ventured further away from the coastline. This could have been the reason as to why the connection was lost when in proximity to SINTEF Ocean's buoy

Real-life 5G measurements for coverage maps can be helpful in a wide array of use cases. An example would be Maritime Robotics' interest in discovering the capabilities of 5G in areas they currently use other forms for communication, such as 4G or even their proprietary wireless WAN. The use of their proprietary wireless network serves as a testament to the fact that the current 4G infrastructure is inadequate with respect to Maritime Robotics' use cases in Trondheim Fjord.

MIDT-NORGE 110-CENTRAL has also expressed interest with regards to obtaining 5G measurements with the purpose of examining the possibilities of 5G in the vertical space. Infrastructure for 5G is currently aimed at the ground-based consumer. However, the *110-central* would like to know if this is a technology they can use with their drones. Not only for steering and low latency applications but also for uploading a high-quality video stream. The potential for use in the vertical domain stretches beyond the needs of the *110-central*, it could be said that it is of great interest for anyone operating in the air within lower altitudes in urbanized environments.

9.4 Encapsulation

The encapsulation performed as expected, giving a practical form factor to the system and providing protection from the environment. However, regardless of the successful

testing period, there is still room for improvement.

The manufacturing of the encapsulation using 3D printing was a practical solution, but it had several drawbacks. First, the PLA plastic used was a very brittle and soft material. Therefore, it was not reasonable to believe the encapsulation would withstand a prolonged testing period spanning several weeks. In addition, there was by no means a guarantee that the walls of the 3D printed model did not have minuscule cracks that could leak water into the interior wall. Given more resources, a better manufacturing process could be developed. However, this was not possible in the short time frame for the project.

One effect of the poor structural integrity of the 3D printed material, was the need for extra support when mounting the system on the USV. Because the plastic could not be trusted to withstand significant amounts of pressure, the encapsulation was fastened using a perforated metal strip. Unfortunately, this made the internals of the encapsulation inaccessible after installation on the USV. The antenna tower also needed additional support, using a rope to ensure the mounting bracket would not snap. This lengthy process of mounting the system on the USV could have been avoided if the material used had been more substantial and more rigid.

The overall concept of encapsulation was proven to perform well. In addition, the rail design added modularity and allowed for a changed focus in future work. For example, the system could be fitted with internal sensors, any components could be upgraded, or complete subsystems could be redesigned without making changes to the encapsulation.

On the other hand, the rail system had no particular components to assist the slide of the mounting plate. It was only plastic rubbing against plastic. This was an obvious flaw, and the plastic rails would be worn out after extensive use. This would have to be improved to a more permanent solution in later iterations.

9.5 BMS Module

The battery had no BMS module but was still equipped with safety features. The interior circuit of the DeWalt charger was unknown, but there was no doubt that it delivered a safe way of charging the battery. In addition, the LVD module added extra safety by disconnecting the battery from its load before the battery reached critically low voltage levels. The fuse also protected the system from short circuits and would reduce the risk of damaging the components or the user. Nevertheless, there were still other safety features that could have been added.

In addition, it would be practical to build a capacity measurement system that could be connected to the Raspberry Pi. This could be used to program battery-level feedback, either over the wireless connection or the OLED screen. Real-time feedback on the

remaining battery capacity of the system could help inform the user to make a decision in the field when taking measurements. However, this would have required some testing to estimate the voltage to capacity curve because this information was not stated in the specifications.

One of the most considerable restrictions in designing a BMS module was size constraints determined by the available space. Given the production method used, with NTNU Elektronikk og prototypelaboratoriet, it would not have been possible to manufacture a PCB small enough to house all the features mentioned above and still fit on the mounting plate and the battery. A solution would have been to produce the PCB via a professional PCB manufacturer, but this was outside of the project's time frame.

It must also be noted that the LVD did not get produced in time for any of the testing, and once it was produced, it was unusable. So during the test, the participants relied on the procedure to ensure battery safety 7.1.4. The LVD design was simulated in LTSpice, assembled on a breadboard, and subsequently tested. The LVD was found to function as intended, and therefore it is considered part of the system. The equivalent breadboard circuit used for testing is shown in figure 30.

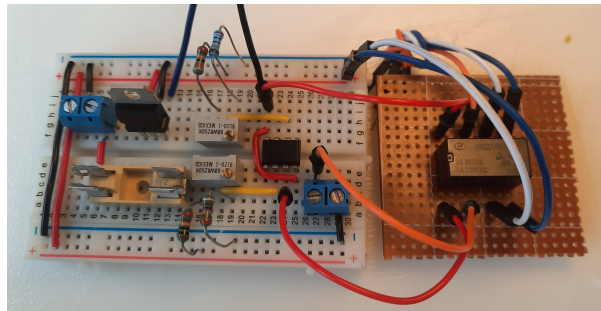


Figure 30: LVD - Breadboard Circuit. The wires going out of the images was connected to the switch. The fuse was not included

9.6 PCB Production

As stated above and in section 7.2.1, the LVD was tested using an equivalent breadboard circuit due to production errors and communication mishaps. This section will discuss the reasons for the faulty PCB.

There was a lack of connections between the two layers of the PCB, essentially creating two separate boards in an electrical sense. As a result, all the components had to be soldered on both sides to connect the two layers. This was not possible for the relay, potentiometers, and connection terminals. Furthermore, a lack of solder masks made soldering difficult with regards to avoiding undesired solder-induced short circuits. Traces would also unfasten from the substrate beneath after attempting to correct errors and remove short circuits.

The first and most straightforward solution would be to order a PCB from a manufacturer like JLCPCB. The second solution would be to redesign the PCB to a one-sided PCB, meaning all electrical connections and soldering would occur on one side. In the case of impossible routing due to a lack of a second layer, a zero-ohm resistor or a wire short could be used to establish an electrical connection over the trace that otherwise would block it. The third solution would be to specify vias for each non-mounting hole drilled in the PCB. The last solution is what should have been done at first, but this was not done due to inexperience regarding PCB manufacturing at NTNU Elektronikk og prototypelaboratoriet. The group was too quick to assume that it would be similar to ordering from a traditional PCB manufacturer.

Had there been more time, the issues related to the production of the PCB would have been solved, and assuming the PCB design had no significant flaws, an LVD would have been finalized and implemented.

9.7 Visualization

In this project, the foundation for visualization was using KML in Google Earth Pro. This was found to be a suitable solution for this thesis, mapping ground- and ocean-based measurements in the trip and grid form. However, there were some possible issues related to the method of visualization used, these potential issues will be discussed here, and a possible solution will be presented.

All mathematical operations were done in degrees using a *flat-earth model*, essentially operating in a Cartesian plane using latitude and longitude. This is due to the fact that any planned and performed visualization had taken place in a geographically confined area somewhere in and around Trondheim. The issue had not presented itself in work related to this project. Nevertheless, it would become apparent when attempting to visualize measurement data in other parts of the world and compare them. The grid sections in grid visualization, and circles for trip visualization, would become deformed and misshaped in areas closer to the equator. One can think of how the Mercator projection inflates the size of objects further away from the equator [33]. This is also illustrated in figure 31. Thus, comparing a grid visualization of an area of 100m by 100m in Trondheim would result in fewer grid sections than mapping an area of the same size in Singapore. The grid sections in Trondheim would be larger than those in Singapore. Trondheim is located at approximately 63° and Singapore at 1° .

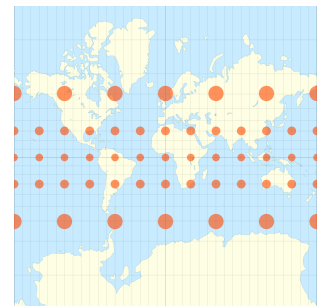


Figure 31: Mercator projection with indicators of deformation [33]

The true size does an excellent job of interactively visualizing this, and a practical visualization showing what happens once one attempts to apply the same algorithm for visualization in Singapore can be found in appendix 9.

The solution would be to convert from degrees to a unit of proper length, such as the meter. It would be essential to consider where the conversion is done, as it would not be the same for all locations. Then, all calculations should be done using meters before converting back to degrees for KML visualization.

10 Future work

Even though the system is functional and ready for use, there is still room for improvements and additions. Therefore, the logical next steps are described here.

10.1 SSH into Raspberry Pi

An unpleasant experience during testing in the fjord, was to run the script manually. Meaning, an SSH connection to Raspberry Pi was established locally through the modem as a router to connect to the PC. The Raspberry Pi was, in other words, remotely controlled through the WiFi connection, essentially a limited area near the router. This meant that when a certain distance was exceeded, the connection was broken. This happened after a bug crashed the code for an unknown reason while the USV and measurement system were lowered into the water. Therefore, the overall system would be improved by implementing a port forwarded SSH through the cellular network to access the Raspberry Pi.

10.2 GUI Website and Database

The next step would be to create a GUI website with full-duplex communication through cellular. This solution would have been more robust and configurable than Blynk. It would also be possible to include multiple actuators and additional indicators from both modem parameters, and potentially sensors to make the system bear resemblances to that of a more finished and polished product. Another feature would be to store the sampled data live into a database during a measurement session instead of eventually retrieving all the data on land. This would relinquish the risk associated with losing valuable data if the measurement system was critically damaged or destroyed.

Additionally, the data processing and coverage mapping could be automatically processed during a measurement session or afterward instead of being done manually.

10.3 Sensor Implementation

During testing on the ocean with Maritime Robotics' USV, there was uncertainty regarding the status of the measurement system that was kilometers away and out at sea. The group posed questions akin to:

- How much heat development is there in the encapsulation?

- Is there a lot of movement due to waves?
- Are the antennas still mounted correctly?
- Is the antenna tower still standing?
- Has water leaked into the system?
- What is the battery status?

The only answer to these questions was a measurement and GPS update every 10 seconds, excluding the status of the USV that was continuously available in the *Control Center* at Maritime Robotics. With a background in a desire to minimize uncertainties regarding the measurement system's status when obtaining data in the field, it is deemed beneficial to incorporate sensors in a future revision or upgrade.

An example of a sensor package to obtain the desired measurements and thus sensors to fulfill the task of obtaining the desired measurement is as follows: For temperature measurement, the relevant sensors could be *thermistors* or *thermocouples*. It would then be possible to use the *analog to digital converter* (ADC) on the Raspberry Pi. As for battery voltage, a simple, high resistance voltage divider could be used to lower the voltage so that the ADC on the Raspberry Pi can measure it. Regarding the antenna tower and antenna orientation, *strain gauges* could be mounted on the tower mast to check for breakage or flexion. *Gyroscopes* could be placed on the antenna mount to measure orientation. If the system was to be dropped on the ground or experience a lot of movement and vibrations, an *accelerometer* could be used to detect this. If water leaks into the system, a *moisture detector* placed at the bottom of the measurement system could be implemented. Thus a sensor package could include the following; thermistor and/or thermocouple, strain gauges gyroscope, accelerometer, and a moisture detector.

10.4 Multiple Carriers Simultaneously

Another implementation that would drastically improve the system's quality, would be to take measurements of multiple cellular providers simultaneously. One significant benefit is that it would only be necessary to do a measurement session once, making the values more qualitative because of the values being measured at the same time. In addition, it would minimize the uncertainties of not maintaining the same route when repeating the process. This could be due to imprecision, weather unpredictability, and other factors. A possible solution would be to buy a second modem and run them at the same time. On the other hand, it comes with the considerable drawback of being expensive. Other solutions could be explored.

10.5 Secondary Cell Providers

The two major telecom operators in Norway are Telia and Telenor. The coverage maps were created using both of these providers. However, there exists secondary telecom providers, and they usually *brag* about having "Full Telenor coverage." Therefore, it will be of value to examine if this statement holds true, not only for the end consumer, but also for the secondary telecom providers.

10.6 Delimitation of Improvements

Based on all previously suggested future improvements to the measurement system, it is necessary to state the importance of delimiting what should and should not be incorporated and included. If too many modules are incorporated, the measurement system will become bloated and will no longer retain its mobility and ease of transportation. Furthermore, increasing the system's complexity will result in an increased chance of failure under operation, essentially reducing the measurement system's reliability. For a measurement system that is to be mounted on a USV and venture out on the sea, it would be of critical importance that the measurement system stays operative for the whole duration. In the case of an error, the measurement system would potentially have to return to shore for human interaction to bring it back to operation. Therefore it is necessary to state the importance of separating what *can* be implemented and what *should* be implemented.

11 Conclusion

The goal of the project was to fulfill the requirements posed by the title of this thesis, *Development of an independent and mobile measurement system for geographical mapping of 5G coverage*. This has been achieved with a high degree of success. The measurement system was made with an overall vision to provide a complete product, being both mobile and independent of the telecom operators.

The final product exceeded the expectations of the collaborators, SINTEF and Maritime Robotics. The custom-designed housing with a complete system, made it possible to automatically sample and store different parameters of the 5G coverage. Furthermore, the coverage maps generated from the raw data using python scrips and Google Earth made for an almost instant visualization of the measured data.

Several test runs demonstrated the functionality of the final product. The coverage maps created from the data collected from the trips around Gløshaugen and Trondheim Fjord made it possible to gain a better understanding of the cellular coverage. This knowledge can be used to assert the capabilities, and with that, the possibilities of 5G in these areas.

The modular nature of the system, both in its hardware configuration and visualization options, made it a potential tool for a wide range of different applications. It has become a quick and easy way of acquiring standalone measurements for mapping the 5G coverage in a geographical location.

12 References

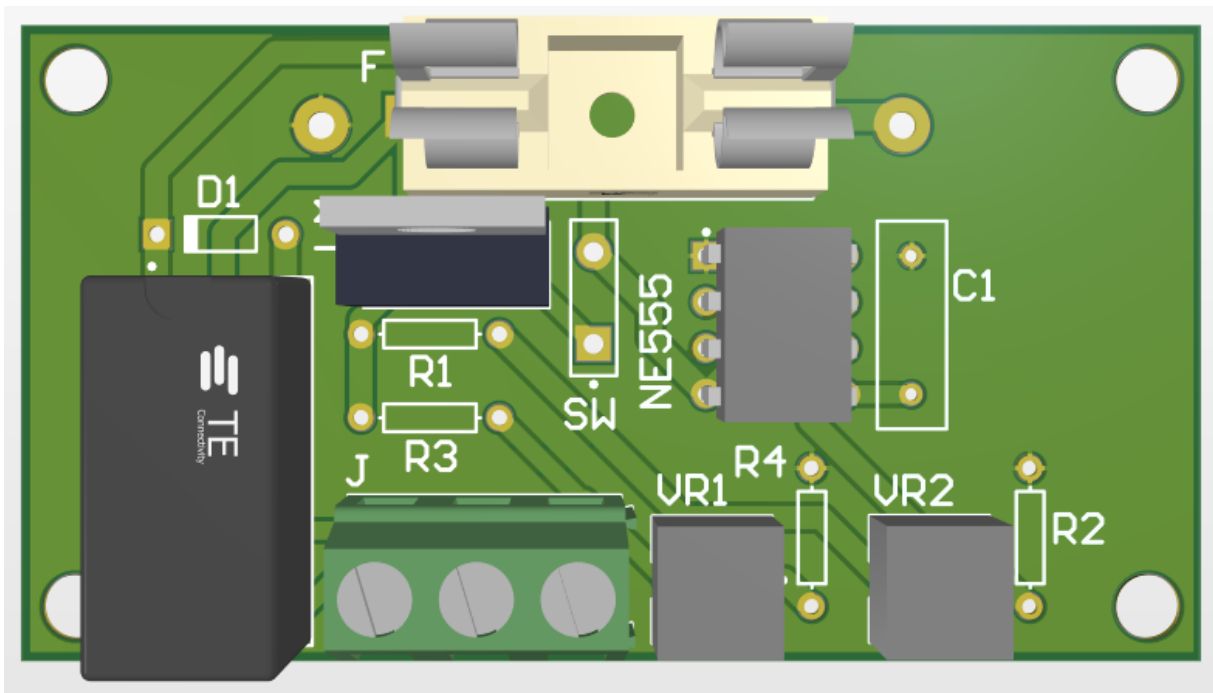
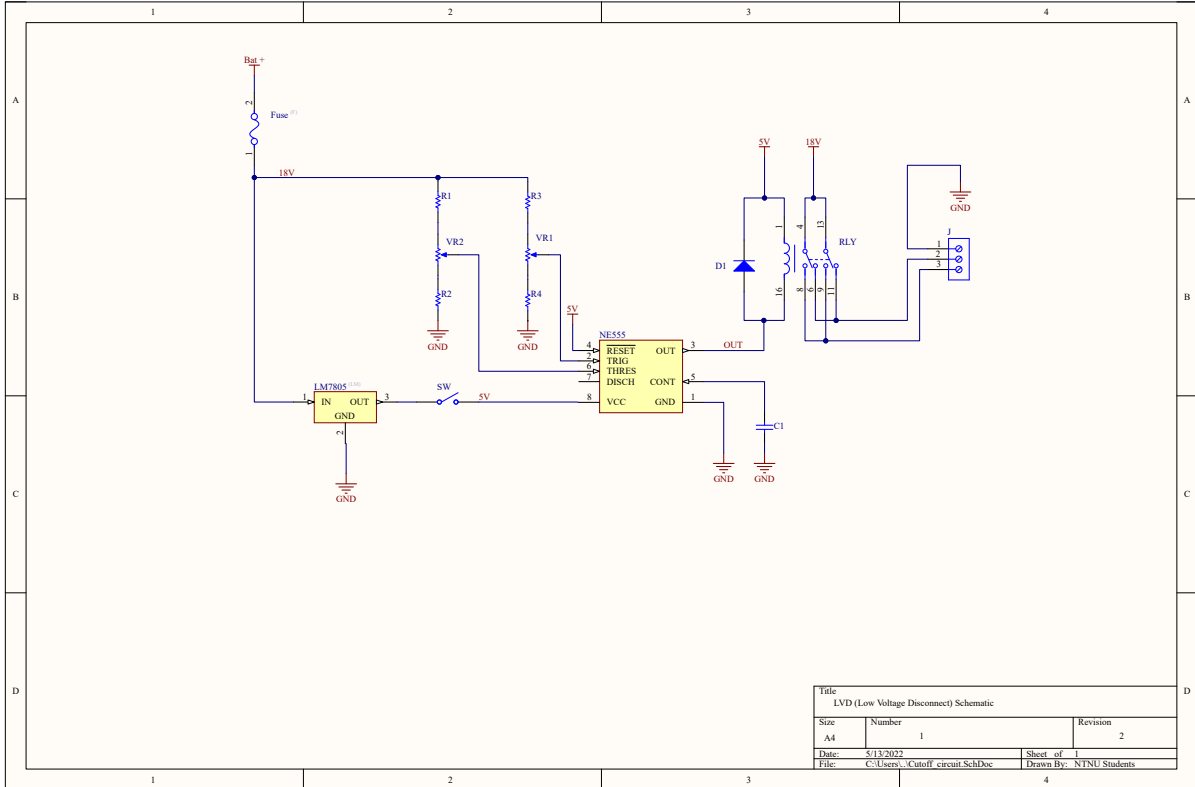
- [1] 5G ACIA (February, 2019). *5G for Connected Industries and Automation* [White Paper]. 5G Alliance for Connected Industries and Automation Retrieved 01 February 2022, from https://5g-acia.org/wp-content/uploads/5G-ACIA_WP_5G-for-Connected-Industries-and-Automation-Second-Edition_SinglePages.pdf
- [2] SINTEF AS. (2022). *Technology for a better society*, Retrieved 14 May 2022 from: <https://www.sintef.no/en/>
- [3] "Otter" (May, 2022). *Otter USV*. Maritime Robotics Product page. Retrieved 25 April 2022, from <https://www.maritimerobotics.com/otter>
- [4] Geographic coordinate system - Wikipedia. En.wikipedia.org. (2022). Retrieved 11 May 2022, from https://en.wikipedia.org/wiki/Geographic_coordinate_system
- [5] National Aeronautics and Space Administration, Science Mission Directorate. (2010). *Anatomy of an Electromagnetic Wave*. Retrieved 06 May 2022, from NASA Science website: http://science.nasa.gov/ems/02_anatomy
- [6] Electromagnetic radiation - Wikipedia. En.wikipedia.org. (2022). Retrieved 11 May 2022, from https://en.wikipedia.org/wiki/Electromagnetic_radiation
- [7] Frenzel, L. E. (2016). *Principles of Electronic Communication Systems* (4th ed.). McGraw-Hill Education.
- [8] Bevelacqua, P. (2022). The Antenna Theory Website. Antenna-theory.com. Retrieved 13 May 2022, from <https://www.antenna-theory.com>.
- [9] Obaid, Fadhil & Al-Rahim, Ali. (2020). *Imaging of 2D Seismic Data Using Time Migration of Ajeel Oilfield, Central of Iraq*. 10.13140/RG.2.2.21505.28003. (Retrieved on 05/05/2022).
- [10] Manz Nilsson. (2021). *Radio-wave propagation modelling over rough sea surfaces and inhomogeneous atmosphere* [Master's thesis]. Karlstad University (Retrieved on 21/04/2022).
- [11] National Instruments (2018). *3GPP Release 15 Overview 3rd Generation Partnership Project (3GPP) members meet regularly to collaborate and create cellular communications standards*, Retrieved 10 May 2022, from <https://spectrum.ieee.org/3gpp-release-15-overview>
- [12] "Signal strength measure" (May 28, 2021). *Signal strength measure RSRP, RSRQ and SINR Reference for LTE Cheat Sheet* [Scientific article]. Poynting (Retrieved from 21/04/2022). <https://poynting.tech/articles/signal-strength-measure>

- [13] "LTE RSSI, RSRP and RSRQ Measurement" (2022). *Signal strength measure RSRP, RSRQ and SINR Reference for LTE Cheat Sheet* [Scientific article]. Cablefree LTE RSSI, RSRP and RSRQ Measurement - CableFree. CableFree. (2022). Retrieved 25 April 2022, from <https://www.cablefree.net/wirelesstechnology/4glte/rsrp-rsrq-measurement-lte/>.
- [14] LTE Physical Layer Overview (2021). *RF and Microwave Navigation*. Keysight Technologies, Inc. Retrieved 06 May 2022, from <https://rfmw.em.keysight.com/wireless>
- [15] "Signal to Interference and Noise Ratio" (May 28, 2021). *What is SINR?* [Scientific article]. ISCO International <https://iscointl.com/sinr-optimization/>
- [16] W. Stallings, "SNMPv3: A security enhancement for SNMP," in *IEEE Communications Surveys*, vol. 1, no. 1, pp. 2-17, First Quarter 1998, doi: 10.1109/COMST.1998.5340405.
- [17] Keyhole Markup Language - Wikipedia. En.wikipedia.org. (2022). Retrieved 2 May 2022, from https://en.wikipedia.org/wiki/Keyhole_Markup_Language.
- [18] Lithium-ion battery - Wikipedia. (2022). Retrieved 16 May 2022, from https://en.wikipedia.org/wiki/Lithium-ion_battery
- [19] Zhao, H. (2022). Low Voltage Cutoff Circuit: The Working Principle and How to Make One. Circuit Board Fabrication and PCB Assembly Turnkey Services - WellPCB. Retrieved 19 May 2022, from <https://www.wellpcb.com/low-voltage-cutoff-circuit.html>.
- [20] "Polylactic Acid" (May, 2022). Polylactic Acid's Wikipedia page. Wikipedia.com. Retrieved 13 May 2022, from https://en.wikipedia.org/wiki/Polylactic_acid
- [21] "DeWalt Battery" (May, 2022). DeWalt Battery 18V 5Ah XR Li-ion. Official product page. Retrieved 02 May 2022, from <https://www.dewalt.no/attachments/productdetails/catno/DCB184/>.
- [22] "Peplink" (May, 2022). Pepwave BR1 Pro 5G datasheet. Peplink Product page. Retrieved 25 April 2022, from <https://www.peplink.com/?s=pepwave+br1+pro+5g>
- [23] "GrabCad Download page" (May, 2022). Users *KlimukVI* "Adaper for Battry" file. Grabcad.com. Retrieved 06 May 2022, from <https://grabcad.com/library/adapter-for-battery-1>
- [24] "LM2596 IC documentation" (May, 2022). LM2596 Datasheet. Texas instrument documentation. Retrieved 6 May 2022, from <https://www.ti.com/lit/ds/symlink/lm2596.pdf>
- [25] "RS Comptens XT60 Product Page" (May, 2022), XT60 Specifications. RS Online, Norway. Retrieved 12 May 2022, from <https://no.rs-online.com/web/p/compact-power-connectors/1805365>

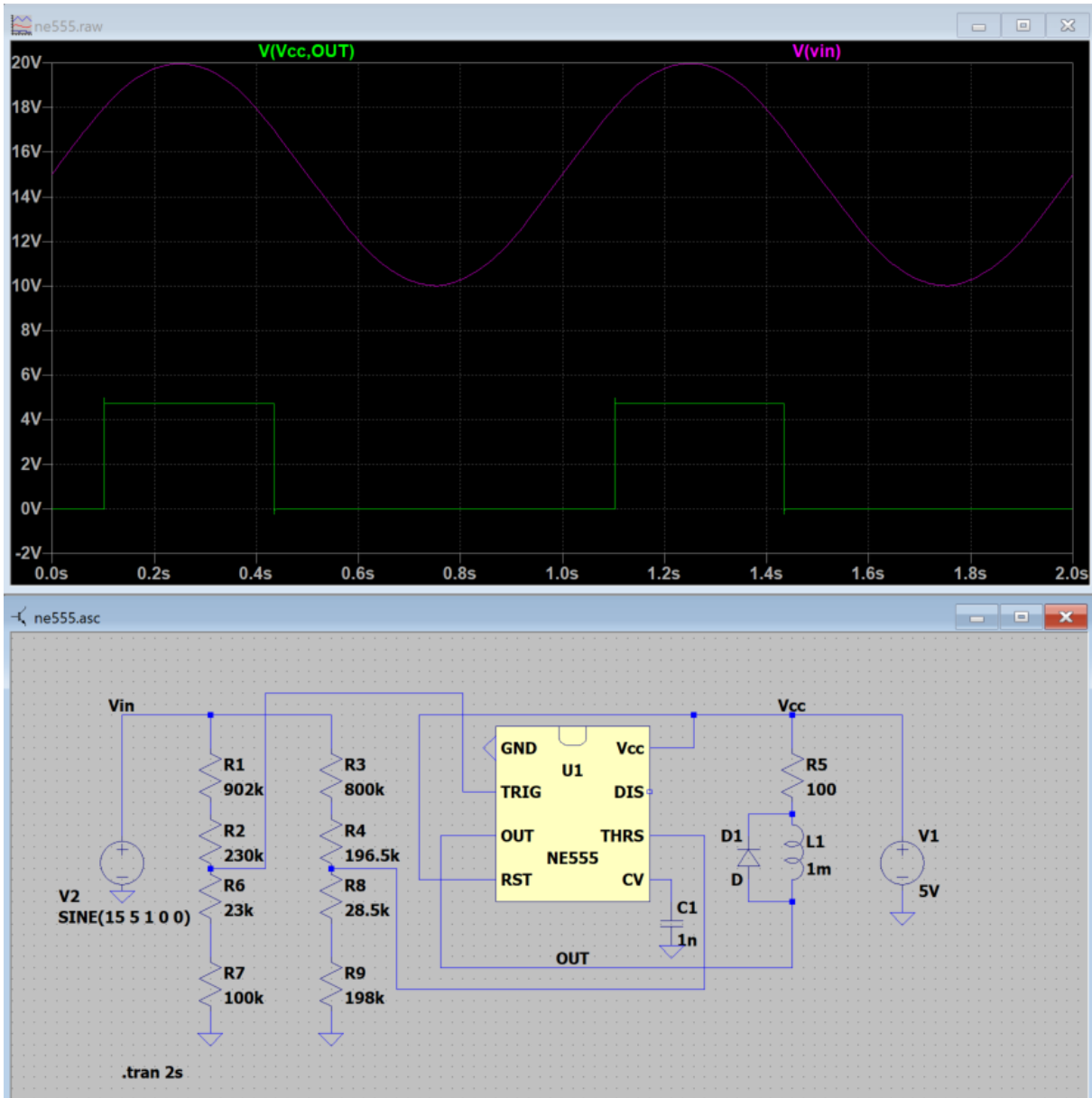
- [26] "Raspberry Documentation" (May, 2022). Raspberry Pi 4B datasheet. Raspberry Pi documentation site. Retrieved 21 April 2022, from <https://datasheets.raspberrypi.com/>
- [27] "Nkom 5G frequency allocation" (Sep, 2021). Norwegian n78 frequency band allocation. Norwegian Communications Authority. Retrieved 6 May 2022, from <https://www.nkom.no/aktuelt/fire-vinnere-i-en-spennende-5g-auksjon>
- [28] "OMIN-404 Product Page" (May, 2022). OMIN-404 Specifications. Poynting. Retrieved 6 May 2022, from <https://poynting.tech/antennas/omni-404/>
- [29] "RS Comptens M5 Brass Threaded Insert Product Page" (May, 2022), Brass Threaded Inserts Specifications. RS Online, Norway. Retrieved 12 May 2022, from <https://no.rs-online.com/web/p/threaded-inserts/2040618>
- [30] JLCPCB. (2022). *PCB Prototype and PCB Fabrication*. Retrieved 14 May 2022, from <https://jlcpcb.com/>
- [31] SINO WEALTH (2013). *SH1106*. Retrieved 14. May 2022, from https://www.velleman.eu/downloads/29/infosheets/sh1106_datasheet.pdf
- [32] "Packing Nipple" (May, 2022). Pakcing Nipple product page. Elektroimportoren.no. Retrieved 13 May 2022, from <https://www.elektroimportoren.no/nippel>
- [33] Mercator projection - Wikipedia. (2022). Retrieved 15 May 2022, from https://en.wikipedia.org/wiki/Mercator_projection

13 Appendix

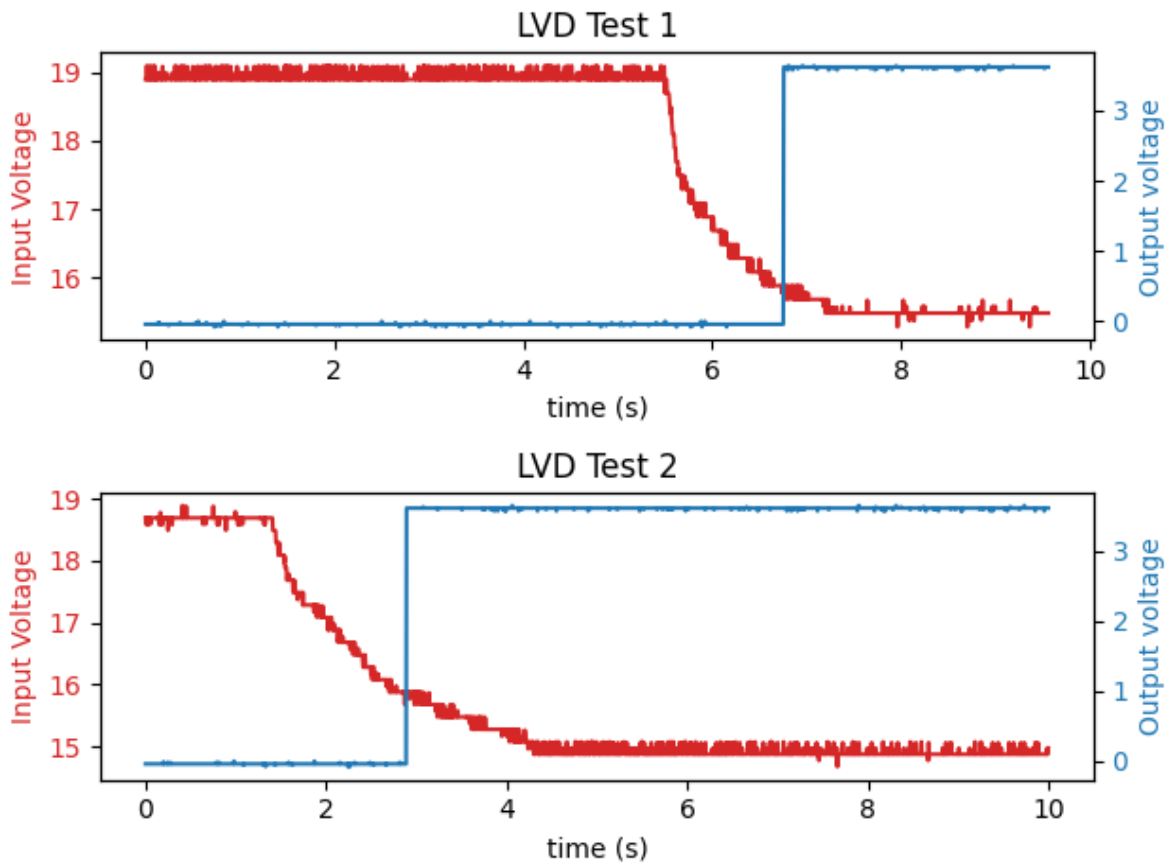
1. LVD schematic and PCB render



2. LTSpice LVD simulation



3. LVD test



4. Ocean based RSRP measurement, 4G and 5G



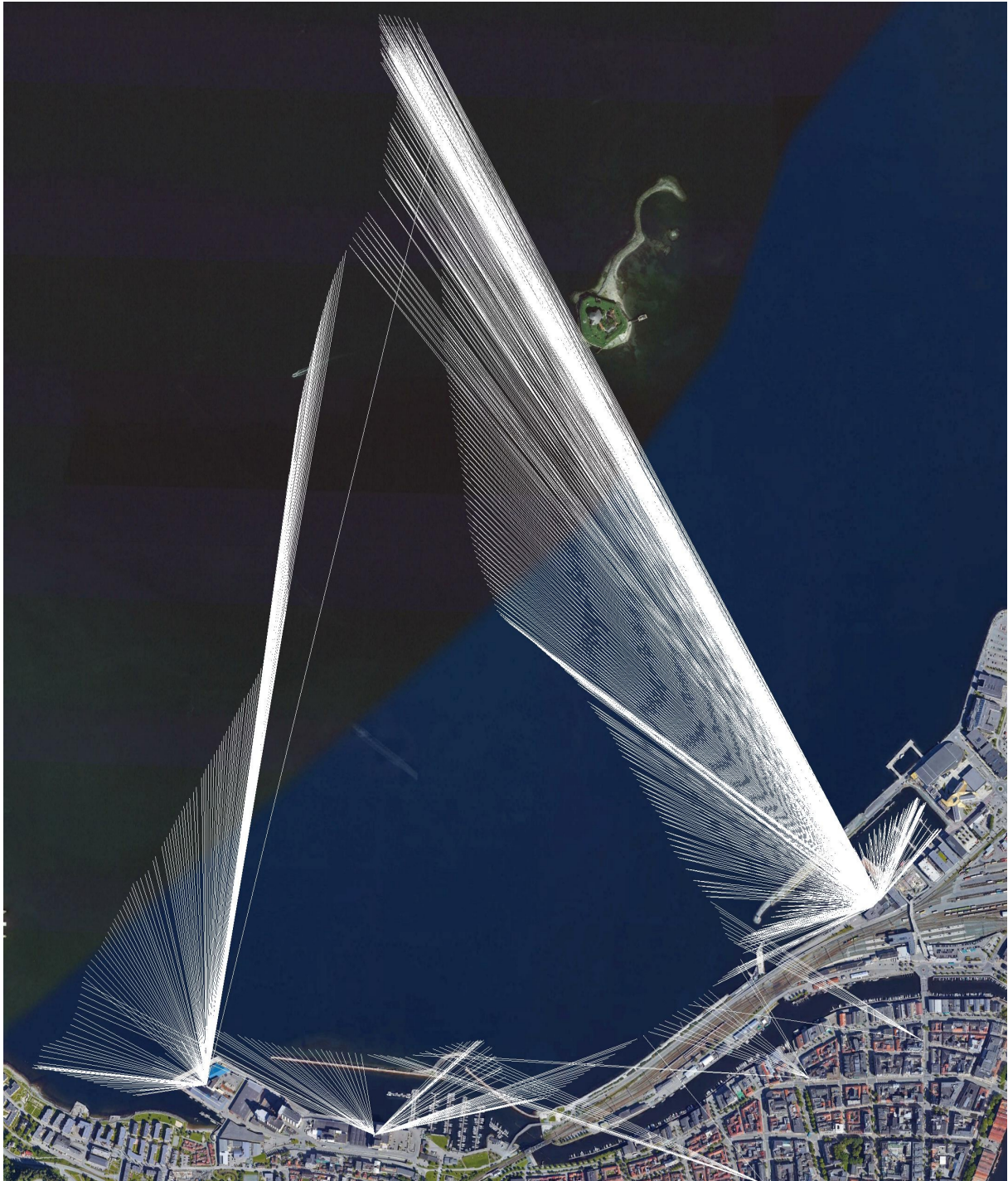
5. Ocean based RSRQ measurement, 4G and 5G



6. Ocean based SINR measurement, 4G and 5G



7. Ocean based assigned base station location indication



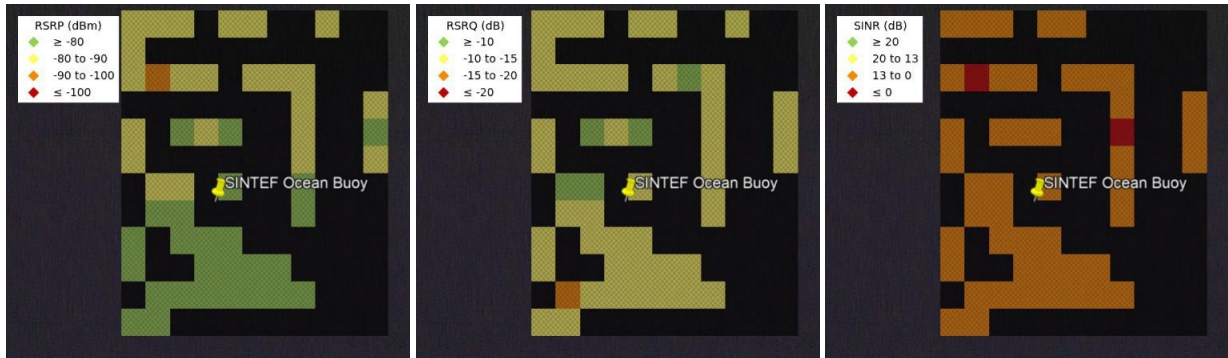
8. Ocean based no cellular visualization



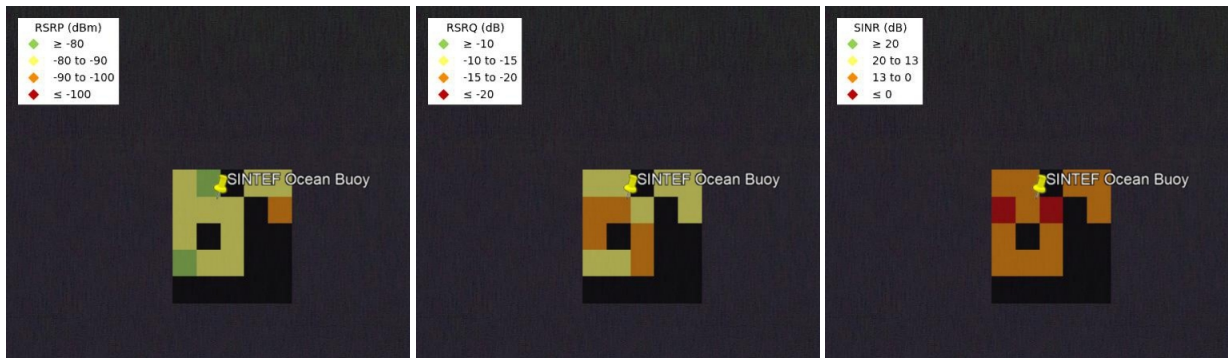
9. Comparison of grid visualization in Trondheim (1) and Singapore (2)



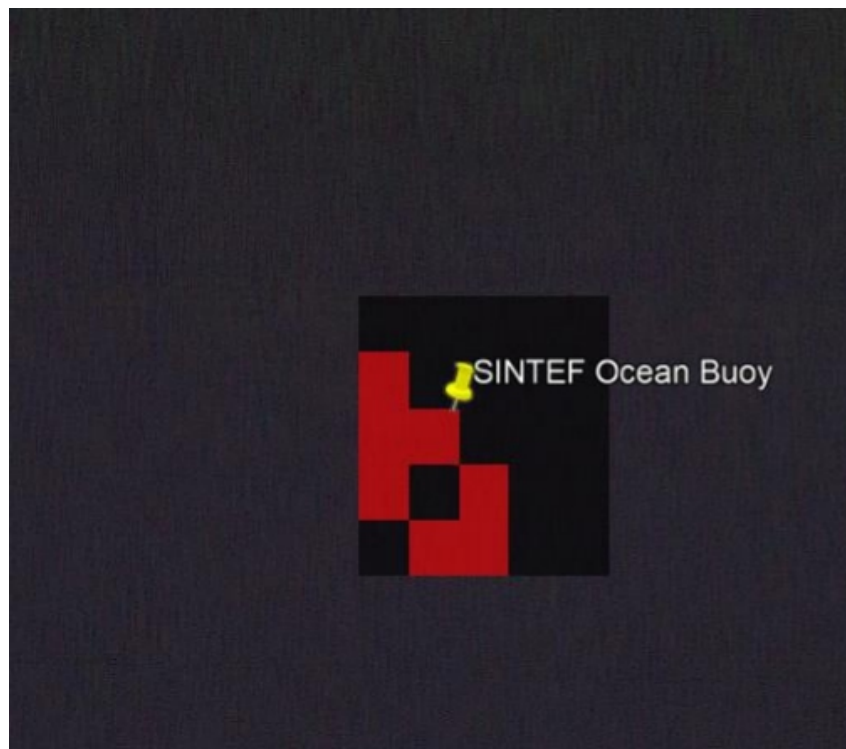
10. 4G Grid visualization near SINTEF OCEAN buoy



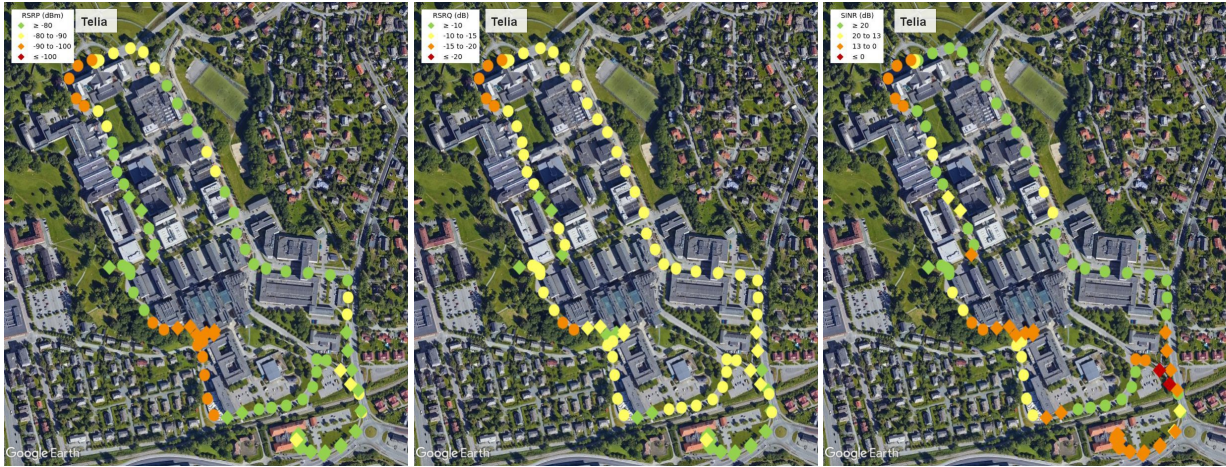
11. 5G Grid visualization near SINTEF OCEAN buoy



12. "No cellular" Grid visualization near SINTEF OCEAN buoy



13. Trip visualization Telia (RSRP, RSRQ and SINR), Gløshaugen



14. Trip visualization Telia, heading and lines



15. Trip visualization Telia vs Telenor (RSRP), Gløshaugen



16. Code to start a measurement run

```

1 #####
2 # Libraries
3 #####
4 from easysnmp import Session
5 from tcp_latency import measure_latency
6 import numpy as np
7 from pynmeagps import NMEAReader
8 import socket, math, time, random
9 import csv
10 import os
11 from datetime import datetime
12 import BlynkLib
13 import schedule
14 import geopy.distance
15 import pprint
16 import RPi.GPIO as GPIO
17
18 from luma.core.interface.serial import i2c
19 from luma.core.render import canvas
20 from luma.oled.device import ssd1306, ssd1325, ssd1331, sh1106
21
22 #####
23 # Class
24 #####
25
26 class routerInfo():
27     def __init__(self, routerMIB):
28         # Device info
29         self.routerMIB = routerMIB
30         self.service =
31         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.4.1.1.1") [0].value
32         self.date = routerMIB.walk("iso.3.6.1.4.1.27662.200.1.1.1.2.1") [0].value
33
34         # Base station info
35         self.cellID = routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1")
36         self.cellInfo =
37         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.2.1.1.2") [0].value
38         self.eNBID =
39         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.13") [0].value
40         self.cellUtranID =
41         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.16") [0].value
42
43         # Parameters
44         self.signalwanID =
45         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.2") [0].value
46         self.signalRSSI =
47         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.3") [0].value
48         self.signalSNR =
49         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.4") [0].value
50         self.signalSINR =
51         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.5") [0].value
52         self.signalECIO =
53         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.6") [0].value

```



```

45     self.signalRSRP =
46         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.7")[0].value
47     self.signalRSRQ =
48         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.8")[0].value
49
50     # Connection info
51     self.connectionType =
52         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.9")[0].value #
53         ↪ 5G, 4G or 3G
54     self.bandType =
55         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.12.1.1.1.10")[0].value #
56         ↪ Freq band (5G band n78 (3500 MHz))
57
58     # Misc
59     self.systemTime =
60         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.1.1.2.1")[0].value
61     self.systemUpTime =
62         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.1.1.2.2")[0].value
63     self.cpuLoad =
64         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.1.1.3.1")[0].value
65     self.totalMemory =
66         ↪ routerMIB.walk("iso.3.6.1.4.1.27662.200.1.1.1.3.2")[0].value
67
68     #####
69     # Functions
70     #####
71
72     def establishConnection(routerIP, securityUsername, authPass, privPass):
73         connection = Session(hostname=routerIP, version=3,
74             ↪ security_level="auth_with_privacy",
75             security_username=securityUsername, auth_protocol="SHA",
76             auth_password=authPass, privacy_protocol="AES",
77             privacy_password=privPass)
78
79         return connection
80
81     def decimalToDM(nemaIn):
82         theDegrees = nemaIn / 100 # move decimal
83         ↪ point for manipulation
84         degsAndMins = math.modf(theDegrees) # create tuple of
85         ↪ degrees and mins
86         theDegrees = int(degsAndMins[1]) # Grab the
87         ↪ Degrees as an integer
88         theMins = degsAndMins[0] * 100 # restore
89         ↪ original decimal point
90         theMins = round(theMins, 6) # grab the mins
91         convertedMins = theMins / 60 # convert decimal
92         ↪ minutes to mins
93         convertedResult = str(round(theDegrees + convertedMins, 6)) # Add the degrees
94         ↪ and mins together
95
96         return convertedResult
97
98     def getGPSPosition():
99         # Set up a client socket to connect to the GPS enabled device

```

```
83     client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
84     client_socket.settimeout(1)
85     client_socket.connect((routerIP, gpsPort))
86     # ping = np.average(measure_latency(host='google.com', runs=1, timeout=1))
87
88     # Whilst we've received a response from the socket...
89     data = client_socket.recv(1024)
90
91     # make sure we actually have some data (and not just a response).
92     if len(data) > 0:
93         # decode the data from byte to string to make working with it easier
94         thedata = data.decode('ascii')
95         # split the data into a list of lines
96         lines = thedata.splitlines(1)
97         # iterate over each line
98         for line in lines:
99             # Data sent is comma delimited so lets split it
100            gpsstring = line.split(',')
101            # if the first column contains $GPRMC and
102            # if it has enough elements we have hit paydirt
103            if gpsstring[0] == '$GPRMC' and len(gpsstring) > 6:
104                # check that there is returned GPS data
105                if gpsstring[3]:
106                    # Get Lat and Long String by converting decimals and adding
107                    # ↪ compass pos
108                    theLat = decimalToDM(float(gpsstring[3])) # + (gpsstring[4])
109                    theLong = decimalToDM(float(gpsstring[5])) # +
110                    # ↪ (gpsstring[6])
111                    #print("Lat: " + theLat)
112                    #print("Long: " + theLong)
113
114                    return [theLat, theLong]
115
116     client_socket.close()
117
118 def appendCSV(direction, service, gpsStatus, current_val):
119     path = './Bachelor/measurements/' + service + '/' + folderInfo + '/' +
120     ↪ gpsStatus + '/'
121     filename = direction + '.csv'
122
123     if os.path.exists(path) == False:
124         os.makedirs(path)
125         print("making folder")
126
127     header = ['timestamp', 'longitude', 'latitude', 'RSRP', 'RSRQ', 'SINR',
128     ↪ 'CellUtranID']
129     print("test1")
130     with open(path + filename, 'a+', encoding='UTF8') as f:
131         writer = csv.writer(f)
132         writer.writerow([folderInfo])
133         writer.writerow(header)
134         writer.writerow(current_val)
135
136     else:
137         print("test2")
138         with open(path + filename, 'a', encoding='UTF8') as f:
139             writer = csv.writer(f)
```

```

134         writer.writerow(current_val)
135
136 #####
137 # Constants and variables
138 #####
139 # Pepwave IP and port with GPS module
140 routerIP = "192.168.50.1"
141 gpsPort = 60660
142
143 # Booleans
144 runMeasurements = 0
145 refreshTime = 5
146 resetTime = 0
147 folderInfo = None
148 client_socket = None
149
150 # Button initialiser
151 GPIO.setwarnings(False)
152 GPIO.setmode(GPIO.BOARD)
153 GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_UP)
154
155 # Oled display
156 serial = i2c(port=1, address=0x3C)
157 device = sh1106(serial, rotate=2)
158
159 # Establish connection with SNMP
160 pepwaveMIB = establishConnection("192.168.50.1", "EASYSNMP", "AUTHPASS",
    ↪ "PRIVPASS")
161
162 # Establish blynk connection
163 BLYNK_AUTH = 'JFQ4mBNncVtuDt4dVuHCkGkyr_PtzGHJ'
164 blynk = None
165
166 # Try conencting to blynk
167 try:
168     blynk = BlynkLib.Blynk(BLYNK_AUTH, server="blynk-cloud.com", port=8080)
169 except:
170     print("no internet")
171
172 # Print to OLED screen
173 with canvas(device) as draw:
174     draw.rectangle(device.bounding_box, outline="white", fill="black")
175     draw.text((10, 10), "IDLE", fill="white")
176
177 print("idle state. Waiting to start new meas")
178
179
180 #####
181 # Code
182 #####
183 def main():
184     # Update the info
185     client = routerInfo(pepwaveMIB)
186
187     try: # Attempt to retrieve real GPS position from the Pepwave module

```

```

188     test = getGPSPosition()
189     current_lat = test[0]
190     current_long = test[1]
191
192     # If connectiotype and signal strength is found
193     if len(client.connectionType) != 0 and client.signalRSRP != '-9999':
194         appendCSV(client.connectionType, client.service, 'GPS',
195                 → [datetime.now().strftime("%d-%m-%Y--%H-%M-%S"), current_lat,
196                 → current_long,
197
198                                     client.signalRSRP,
199                                     → client.signalRSRQ,
200                                     → client.signalSINR,
201                                     → client.cellUtranID])
202
203         print("signal + gps")
204     else:
205         appendCSV('NoCellular', client.service, 'GPS',
206                 → [datetime.now().strftime("%d-%m-%Y--%H-%M-%S"), current_lat,
207                 → current_long,
208
209                                     client.signalRSRP,
210                                     → client.signalRSRQ,
211                                     → client.signalSINR,
212                                     → client.cellUtranID])
213
214         print("!signal + gps")
215     except:
216         # If no GPS signal, create fake coordinates for testing
217         current_lat = round(random.uniform(63.359685, 63.461767), 5)
218         current_long = round(random.uniform(10.37548, 10.559925), 5)
219
220         # If connectiotype and signal strength is found
221         if len(client.connectionType) != 0 and client.signalRSRP != '-9999':
222             appendCSV(client.connectionType, client.service, 'noGPS',
223                     → [datetime.now().strftime("%d-%m-%Y--%H-%M-%S"), 0, 0,
224
225                                     client.signalRSRP,
226                                     → client.signalRSRQ,
227                                     → client.signalSINR,
228                                     → client.cellUtranID])
229
230             print("signal + !gps")
231         else:
232             appendCSV('NoCellular', client.service, 'noGPS',
233                     → [datetime.now().strftime("%d-%m-%Y--%H-%M-%S"), 0, 0,
234
235                                     client.signalRSRP,
236                                     → client.signalRSRQ,
237                                     → client.signalSINR,
238                                     → client.cellUtranID])
239
240             print("!signal + !gps")
241
242     # Print parameters
243     print(datetime.now().strftime("%d-%m-%Y--%H-%M-%S"))
244     with canvas(device) as draw:
245         # Running
246         draw.rectangle(device.bounding_box, outline="white", fill="black")
247         draw.text((10, 10), "RUNNING", fill="white")
248
249         # Service (Telia or Telenor)
250         draw.text((10, 20), "Service:", fill="white")

```

```

225         draw.text((70, 20), client.service, fill="white")
226
227         # Connection type (4G, 5G)
228         draw.text((10, 30), "Type:", fill="white")
229         draw.text((70, 30), client.connectionType, fill="white")
230
231         # Signal strength
232         draw.text((10, 40), "RSRP:", fill="white")
233         draw.text((70, 40), client.signalRSRP, fill="white")
234
235         # Timestamp
236         draw.text((5, 50), datetime.now().strftime("%d-%m-%Y--%H-%M-%S"),
237             ↪ fill="white")
238
239     # Update Blynk GUI
240     try:
241         blynk.virtual_write(0, 1, current_lat, current_long, 'Position') #
242         ↪ Update GPS-map
243         blynk.virtual_write(1, current_lat) #
244         ↪ Update latitude
245         blynk.virtual_write(2, current_long) #
246         ↪ Update longitude
247         blynk.virtual_write(3, client.service) #
248         ↪ Update longitude
249         blynk.virtual_write(4, client.connectionType) #
250         ↪ Update connection type
251         blynk.virtual_write(5, client.signalRSRP) #
252         ↪ Update signal RSRP
253         blynk.virtual_write(7, datetime.now().strftime("%d-%m-%Y--%H-%M-%S")) #
254         ↪ Update timestamp
255     except:
256         print("BLYNK ERROR. Broken pipe when virtual write to blynk")
257
258     # If physical or virtual button is pressed to stop a measurement session
259     if runMeasurements == 0 and resetTime == 1:
260         try: # If ssh connecton can be established, sync files, Removed for now
261             ↪ due to access error
262             #os.system("rsync -a ./Bachelor/measurements
263             ↪ johanvu@192.168.50.10:./Documents/GitHub/RaspberryPi-5G/RaspberryPi/")
264             #print("sync to PC")
265             print("idle state. Waiting to start new meas")
266
267             # Print to OLED screen
268             with canvas(device) as draw:
269                 draw.rectangle(device.bounding_box, outline="white",
270                     ↪ fill="black")
271                 draw.text((10, 10), "IDLE", fill="white")
272         except:
273             print("failed to sync")
274         return schedule.CancelJob
275
276     while True:
277         # Run measurement session with interval x seconds
278         schedule.run_pending()
279
280

```

```
269     try: # Blynk error when no internet connection is present
270         blynk = BlynkLib.Blynk(BLYNK_AUTH, server="blynk-cloud.com", port=8080)
271         blynk.run()
272
273         # Switch actuator removed for now
274         # @blynk.ON('V6')
275         #def V6_ButtonRead(value):
276         #     global runMeasurements
277         #     runMeasurements = int(value[0])
278         #     print('Current switch value: {}'.format(runMeasurements))
279
280         # Refresh time actuator
281         # @blynk.ON('V5')
282         #def V6_SliderRead(value):
283         #     global refreshTime
284         #     refreshTime = int(value[0])
285         #     print('Current slider value: {}'.format(refreshTime))
286
287     # Known errors
288     except IOError as e:
289         if e.errno == errno.EPIPE:
290             print("ERROR BLYNK. Broken pipe")
291             pass
292     except IndexError as ie:
293         print("ERROR BLYNK. Index error. Fault in library")
294     except:
295         print("ERROR BLYNK")
296
297     # If physical button is pressed
298     if GPIO.input(12) == GPIO.HIGH:
299         time.sleep(0.5) # Easy way to avoid debounce
300         runMeasurements = 1
301     else:
302         time.sleep(0.5) # Easy way to avoid debounce
303         runMeasurements = 0
304
305     # Update button in blynk if any real changed
306     #try:
307     #     blynk.virtual_write(6, int(runMeasurements))
308     #except:
309     #     print("BLYNK ERROR. Update from script to button blynk")
310
311     # Start measurement session, this will run once in the loop
312     if runMeasurements == 1:
313         if resetTime == 1:
314             folderInfo = datetime.now().strftime("%d-%m-%Y--%H-%M-%S")
315             main()
316             schedule.every(refreshTime).seconds.do(main)
317             resetTime = 0
318             print(folderInfo)
319             try:
320                 blynk.virtual_write(6, 255)
321             except:
322                 print("blynk error. virtual write turn on led")
323     else:
```

```
324         resetTime = 1
325         try:
326             blynk.virtual_write(6, 0)
327         except:
328             print("blynk error. virtual write turn off led")
329
330 # Refresh time in code
331 time.sleep(refreshTime)
```


17. Code to make a coverage map

```

1 #####
2 # Libraries
3 #####
4 import numpy as np
5 import simplekml
6
7 #####
8 # Dataset import, variables and constants
9 #####
10 # Import datasets
11 BaseStations = np.loadtxt("data/TeliaNetComBS.txt", delimiter=',', skiprows=2,
12                          unpack=True)
13
14 CSV_5G = np.loadtxt("data/GPS_ocean/5G.csv", delimiter=',', skiprows=2,
15                   unpack=True, usecols=range(1, 7))
16 CSV_5G_TIME = np.loadtxt("data/GPS_ocean/5G.csv", delimiter=',', skiprows=2,
17                          usecols=[0], dtype=str)
18
19 CSV_LTE = np.loadtxt("data/GPS_ocean/LTE.csv", delimiter=',', skiprows=2,
20                    unpack=True, usecols=range(1, 7))
21 CSV_LTE_TIME = np.loadtxt("data/GPS_ocean/LTE.csv", delimiter=',', skiprows=2,
22                          usecols=[0], dtype=str)
23
24 CSV_LTE_A = np.loadtxt("data/GPS_ocean/LTE-A.csv", delimiter=',', skiprows=2,
25                      unpack=True, usecols=range(1, 7))
26 CSV_LTE_A_TIME = np.loadtxt("data/GPS_ocean/LTE-A.csv", delimiter=',',
27                             skiprows=2, usecols=[0], dtype=str)
28
29 CSV_NOCELL = np.loadtxt("data/GPS_ocean/noCellular.csv", delimiter=',',
30                       skiprows=2, unpack=True, usecols=range(1, 7))
31 CSV_NOCELL_TIME = np.loadtxt("data/GPS_ocean/noCellular.csv", delimiter=',',
32                              skiprows=2, usecols=[0], dtype=str)
33
34 # Variables
35 hhmss_5G = []
36 hhmss_LTE = []
37 hhmss_LTE_A = []
38 hhmss_NOCELL = []
39
40 # Thresholds
41 THRESHOLDS_RSRP = np.array([-80, -90, -100])
42 THRESHOLDS_SINR = np.array([20, 13, 0])
43 THRESHOLDS_RSRQ = np.array([-10, -15, -20])
44
45 # Size of images to be placed, and size of grid
46 DLAT = 0.0005 # latitude is -90 to 90
47 DLON = 0.001 # longitude is -180 to 180
48
49 # Maritime Robotics HQ Brattøra
50 SNAP_LAT = 63.44027859540668
51 SNAP_LON = 10.399713079298035
52
53

```

```

54 #####
55 # Functions
56 #####
57
58 def snap_coordinates_from_array(lat_arr: np.ndarray, lon_arr: np.ndarray,
59                               lat_snap: float, lon_snap: float,
60                               lat_step: float, lon_step: float):
61     """
62     This function is the same as snap_coordinates, however it differs by taking
63     on arrays instead of max and min values of said arrays.
64     :param lat_arr: Latitudinal array
65     :param lon_arr: Longitudinal array
66     :param lat_snap: Desired "snap" coordinates, latitudinal
67     :param lon_snap: Desired "snap" coordinates, longitudinal
68     :param lat_step: "Step" length of grid, in latitudinal terms/direction.
69     :param lon_step: "Step" length of grid, in longitudinal terms/direction.
70     :return: A 2D array containing the minimum and maximum snap coordinates
71             in the format ([[lat_min,lon_min],
72                             [lat_max, lon_max]])
73     """
74     lat_arr_max = np.max(lat_arr)
75     lat_arr_min = np.min(lat_arr)
76     lon_arr_max = np.max(lon_arr)
77     lon_arr_min = np.min(lon_arr)
78     calc_min_num_snap_grids_lon = round((lon_arr_min - lon_snap) / lon_step)
79     calc_min_num_snap_grids_lat = round((lat_arr_min - lat_snap) / lat_step)
80     calc_max_num_snap_grids_lon = round((lon_arr_max - lon_snap) / lon_step)
81     calc_max_num_snap_grids_lat = round((lat_arr_max - lat_snap) / lat_step)
82
83     # Calculate new coordinates for grid outline
84     snap_x_lon_min = calc_min_num_snap_grids_lon * lon_step + lon_snap
85     snap_y_lat_min = calc_min_num_snap_grids_lat * lat_step + lat_snap
86     snap_x_lon_max = calc_max_num_snap_grids_lon * lon_step + lon_snap
87     snap_y_lat_max = calc_max_num_snap_grids_lat * lat_step + lat_snap
88
89     print("Minimum coordinates", "(" , lat_arr_min, lon_arr_min, ")")
90     print("Minimum snapcoords", "(" , snap_y_lat_min, snap_x_lon_min, ")")
91     print("Difference: ", lat_arr_min - snap_y_lat_min,
92           lon_arr_min - snap_x_lon_min)
93     print("-----")
94     print("Maximum coordinates", "(" , lat_arr_max, lon_arr_max, ")")
95     print("Maximum coordinates", "(" , snap_y_lat_max, snap_x_lon_max, ")")
96     print("Difference: ", lat_arr_max - snap_y_lat_max,
97           lon_arr_max - snap_x_lon_max)
98     return np.array(
99         [[snap_y_lat_min, snap_x_lon_min], [snap_y_lat_max, snap_x_lon_max]])
100
101
102 def calc_z_snapgrid(data_array: np.ndarray, snap_array: np.ndarray,
103                    step_array: np.ndarray):
104     """
105     This function serves the purpose of calculating the average
106     value of all measurements taken within a "grid-section".
107     :param data_array: Data array, [lat,lon,sig]
108     :param snap_array: Snap coordinates in array form

```

```

109     :param step_array: Coordinal step length in array form
110     :return: A 2D array that contains average signal values
111     """
112
113     lat_arr = data_array[0, :]
114     lon_arr = data_array[1, :]
115     sig_arr = data_array[2, :]
116     min_lat, min_lon = snap_array[0, :]
117     max_lat, max_lon = snap_array[1, :]
118     print("min", min_lat, min_lon)
119     print("max", max_lat, max_lon)
120     step_lat = step_array[0]
121     step_lon = step_array[1]
122     arr_len = min(len(lat_arr), len(lon_arr),
123                  len(sig_arr)) # Get minimum of all just incase
124
125     num_x_grids = round((max_lon - min_lon) / step_lon) # With "round"
126                  # we will discard data if it is < x.5
127     num_y_grids = round((max_lat - min_lat) / step_lat)
128     print(num_x_grids, num_y_grids)
129     z = np.empty((num_x_grids, num_y_grids),
130                 float) # Create empty 2D array x*y to hold float values
131
132     # Iterate through entire grid
133     for x_lon in range(0, num_x_grids): # X axis is LON
134         globals()['counter'] = 0
135         globals()['sig_sum'] = 0
136         for y_lat in range(0, num_y_grids): # Y axis is LAT
137             counter = 0
138             sig_sum = 0
139             # Find coordinates that belong to grid boundaries and calculate
140             # average signal value
141             for j in range(0, arr_len): # Iterate through data and sort
142                 if (lon_arr[j] > min_lon + x_lon * step_lon) and (
143                     lon_arr[j] < min_lon + (1 + x_lon) * step_lon):
144                     # If LON coord belongs within grid
145                     if (lat_arr[j] > min_lat + y_lat * step_lat and lat_arr[
146                         j] < min_lat + (1 + y_lat) * step_lat):
147                         sig_sum += 10**(sig_arr[
148                             j]/10) # Log to linear
149                         counter += 1 # Increment counter
150             if counter == 0: # Incase no measurements taken in grid
151                 z[x_lon, y_lat] = -1 # Set value to -1
152             else:
153                 z[x_lon, y_lat] = 10*np.log10(sig_sum / counter) # 2D array
154                 # containing
155                 # logarithmic averages
156
157     return z
158
159 def kml_lines(data_arr: np.ndarray, bs_arr: np.ndarray, kml_filename: str):
160     """
161     This function serves the purpose of creating lines between a measurement
162     and its base station.
163     :param data_arr: Array containing measurement location and UTRAN Cell ID

```

```

163         (lat, lon, ... UTRAN Cell ID)
164     :param bs_arr: Array containing base stations, (lat, lon, eNB ID)
165     :param kml_filename: Desired filename, no .kml ending
166     :return: None
167     """
168     kml = simplekml.Kml()
169     for i in range(len(data_arr[0, :])):
170         for j in range(len(bs_arr[0, :])):
171             if int(bs_arr[0, j] == int(data_arr[5, i] / 256)):
172                 linestring = kml.newlinestring(name="NoName")
173                 linestring.coords = [(data_arr[1, i], data_arr[0, i]),
174                                     (bs_arr[2, j], bs_arr[1, j])]
175     print("KML file created with name: " + kml_filename + ".kml")
176     kml.save(kml_filename + ".kml")
177
178
179 def kml_arrow(gps_arr: np.ndarray, lon_step: int or float,
180             lat_step: int or float,
181             img_path_fname: str, kml_filename: str):
182     """
183     This function serves the purpose of creating arrows that indicate the direction
184     of heading for a set of measurements
185     :param gps_arr: Array containing (lat, lon)
186     :param lon_step: Size of step in longitudinal direction
187     :param lat_step: Size of step in latitudinal direction
188     :param img_path_fname: Path and name of image to be placed
189     :param kml_filename: Name of kml_filename to be created
190     :return: None
191     """
192     lon_arr = gps_arr[1, :]
193     lat_arr = gps_arr[0, :]
194     kml = simplekml.Kml()
195     for i in range(len(gps_arr[1, :]) - 1):
196         dlon = 50 * lon_arr[i + 1] - lon_arr[i] * 50 # Convert to "proper" unit
197         dlat = 100 * lat_arr[i + 1] - lat_arr[i] * 100
198         deg = np.arctan2(dlat, dlon) * 180 / np.pi + 180
199         name = str(deg)
200         globals()[name] = kml.newgroundoverlay(name=name)
201         globals()[name].icon.href = img_path_fname
202         # Define borders and rotation on map for PNG
203         globals()[name].latlonbox.north = lat_arr[i] + lat_step / 2
204         globals()[name].latlonbox.south = lat_arr[i] - lat_step / 2
205         globals()[name].latlonbox.east = lon_arr[i] - lon_step / 2
206         globals()[name].latlonbox.west = lon_arr[i] + lon_step / 2
207         globals()[name].latlonbox.rotation = deg
208     kml.save(kml_filename + ".kml")
209     print("KML file created with name: " + kml_filename + ".kml")
210
211
212 def kml_screenoverlay(kml_class, name: str, overlay_path_and_fname: str):
213     """
214     This function serves the purpose of creating a screenoverlay in KML
215     :param kml_class: simpleKML usage
216     :param name: Name of screenoverlay in KML
217     :param overlay_path_and_fname: Image to be overlaid

```

```

218     :return: None
219     """
220     screenoverlay = kml_class.newscreenoverlay(name=name)
221     screenoverlay.icon.href = overlay_path_and_fname
222     screenoverlay.overlayxy = simplekml.OverlayXY(x=0, y=1,
223         xunits=simplekml.Units.fraction,
224         yunits=simplekml.Units.fraction)
225     screenoverlay.screenxy = simplekml.ScreenXY(x=15, y=15,
226         xunits=simplekml.Units.pixels,
227         yunits=simplekml.Units.insetpixels)
228     screenoverlay.size.x = -1
229     screenoverlay.size.y = -1
230     screenoverlay.size.xunits = simplekml.Units.fraction
231     screenoverlay.size.yunits = simplekml.Units.fraction
232
233
234 def create_grid_snap_png_kml(data2d: np.ndarray, step_lat: float,
235                             step_lon: float, snap_arr: np.ndarray,
236                             thresholds: np.ndarray,
237                             img_path_and_filename: str,
238                             legend_path_and_filename: str, kml_filename: str):
239     """
240     This function serves the purpose of placing PNGs at different geographical
241     locations into Google Earth, their color is set
242     and constant and represents a value.
243     :param snap_arr: Array calculated using former functions.
244     Defines minimum and maximum lat/lon coordinates
245     :param data2d: Dataset, in the form of a 2d-array
246     with num_x_grids*num_y_grids
247     :param step_lat: Step size in latitudinal direction,
248     this will be size of grid in y-direction
249     :param step_lon: Step size in longitudinal direction,
250     this will be size of grid in x-direction
251     :param thresholds: Threshold values to adhere to when assigning colored PNGs
252     :param img_path_and_filename: Image path and filename of desired image to
253     plot in kml, e.g. "Path/fname.png"
254     :param legend_path_and_filename: Image path and filename of desired
255     legend to overlay in kml, e.g. "Path/fname.png"
256     :param kml_filename: Desired filename of kml file, e.g. "fname.kml"
257     :return: None
258     """
259     kml = simplekml.Kml()
260     min_lat = snap_arr[0, 0]
261     min_lon = snap_arr[0, 1]
262     num_x_grids = len(data2d[:, 0])
263     num_y_grids = len(data2d[0, :])
264     # Iterate through grids and assign data to location.
265     for x in range(0, num_x_grids):
266         for y in range(0, num_y_grids):
267             value = data2d[x, y]
268             kml_name = kml_filename + str([x, y])
269             locals()[kml_name] = kml.newgroundoverlay(
270                 name=kml_filename + str([x, y]), altitudemode='absolute',
271                 altitude=1)
272             locals()[kml_name].icon.href = select_colored_image(value,

```

```

273         thresholds, color_img_path_no_png_end=
274             img_path_and_filename[0:-4])
275     locals()[kml_name].description = "Measurement: " + str(
276         value) # + param
277     locals()[kml_name].color = "99ffffff"
278     locals()[kml_name].gxlatlonquad.coords = [
279         (min_lon + x * step_lon, min_lat + y * step_lat),
280         (min_lon + step_lon * (x + 1), min_lat + y * step_lat),
281         (min_lon + (1 + x) * step_lon, min_lat + (1 + y) * step_lat),
282         (min_lon + x * step_lon, min_lat + (1 + y) * step_lat)]
283
284     kml_screenoverlay(kml, "ScreenOverlayLegend", legend_path_and_filename)
285     kml.save(kml_filename + ".kml")
286     print("KML file created with name: " + kml_filename + ".kml")
287     return None
288
289
290 def create_path_no_cellular_png_kml(lon_step: float, lat_step: float,
291     gps_arr: np.ndarray,
292     img_path_and_fname: str, kml_filename: str):
293     """
294     This function serves the purpose of creating a path visualization with KML
295
296     :param lon_step: Step in longitudinal direction, determines size of png
297     :param lat_step: Step in latitudinal direction, determines size of png
298     :param gps_arr: Array containing coordinates, [lat,lon]
299     :param img_path_and_fname: Path and filename of images to be placed on map
300     :param kml_filename: Output kml file name
301     :return: None
302     """
303     lat_arr = gps_arr[0, :]
304     lon_arr = gps_arr[1, :]
305     num_meas = len(lat_arr)
306
307     kml = simplekml.Kml()
308     for i in range(num_meas):
309         name = str([lat_arr[i], lon_arr[i]])
310         globals()[name] = kml.newgroundoverlay(name=name) # +output_fname)
311         globals()[name].icon.href = img_path_and_fname
312         globals()[name].gxlatlonquad.coords = [
313             (lon_arr[i] - lon_step / 2, lat_arr[i] - lat_step / 2),
314             (lon_arr[i] + lon_step / 2, lat_arr[i] - lat_step / 2),
315             (lon_arr[i] + lon_step / 2, lat_arr[i] + lat_step / 2),
316             (lon_arr[i] - lon_step / 2, lat_arr[i] + lat_step / 2)]
317
318     kml.save(kml_filename + ".kml")
319     print("KML file created with name: " + kml_filename + ".kml")
320     return None
321
322
323 def select_colored_image(value: int or float, thresholds: np.ndarray,
324     color_img_path_no_png_end: str):
325     if value == -1: # No measurement
326         return "png/black_square.png"
327     elif value == -9999: # No cellular

```

```

328     return "png/red.png"
329 elif value >= thresholds[0]:
330     return color_img_path_no_png_end + "0.png"
331 elif thresholds[0] >= value >= thresholds[1]:
332     return color_img_path_no_png_end + "1.png"
333 elif thresholds[1] >= value >= thresholds[2]:
334     return color_img_path_no_png_end + "2.png"
335 elif value <= thresholds[2]:
336     return color_img_path_no_png_end + "3.png"
337 else:
338     return_str = "Could not assign value"
339     print(return_str)
340     return return_str
341
342
343 def create_path_png_kml(lon_step: float, lat_step: float, gps_arr: np.ndarray,
344                        val_arr: np.ndarray,
345                        thresholds: np.ndarray,
346                        img_path_and_name: str, legend_path_and_filename: str,
347                        kml_filename: str):
348     """
349     This function serves the purpose of creating a trip visualization.
350     :param lon_step: Step in longitudinal direction, determines size of png
351     :param lat_step: Step in latitudinal direction, determines size of png
352     :param gps_arr: Array containing coordinates, [lat,lon]
353     :param val_arr: Array containing values to be visualized
354     :param thresholds: Thresholds values for color assignment
355     :param img_path_and_name: Path and filename of images to be placed on map
356     :param legend_path_and_filename: Legend path and filename to be overlay
357     :param kml_filename: Output kml file name
358     :return: None
359     """
360     lat_arr = gps_arr[0, :]
361     lon_arr = gps_arr[1, :]
362     num_meas = len(val_arr)
363
364     kml = simplekml.Kml()
365     for i in range(num_meas):
366         name = str([lat_arr[i], lon_arr[i]])
367         globals()[name] = kml.newgroundoverlay(name=name)
368         globals()[name].icon.href = select_colored_image(value=val_arr[i],
369                 thresholds=thresholds,
370                 color_img_path_no_png_end=img_path_and_name)
371         globals()[name].description = "Measurement: " + str(
372             val_arr[i]) # + param
373         globals()[name].gxlatlonquad.coords = [
374             (lon_arr[i] - lon_step / 2, lat_arr[i] - lat_step / 2),
375             (lon_arr[i] + lon_step / 2, lat_arr[i] - lat_step / 2),
376             (lon_arr[i] + lon_step / 2, lat_arr[i] + lat_step / 2),
377             (lon_arr[i] - lon_step / 2, lat_arr[i] + lat_step / 2)]
378
379     kml.screenoverlay(kml, "ScreenOverlayLegend", legend_path_and_filename)
380     kml.save(kml_filename + ".kml")
381     print("KML file created with name: " + kml_filename + ".kml")
382     return None

```



```

383
384
385 #####
386 # Arrange dataset for arrow
387 #####
388 # Get time as hhmss format
389 for listindex in CSV_5G_TIME:
390     hhmss_5G.append(
391         float(listindex[-8:-6] + listindex[-5:-3] + listindex[-2:]))
392 for listindex in CSV_LTE_TIME:
393     hhmss_LTE.append(
394         float(listindex[-8:-6] + listindex[-5:-3] + listindex[-2:]))
395 for listindex in CSV_LTE_A_TIME:
396     hhmss_LTE_A.append(
397         float(listindex[-8:-6] + listindex[-5:-3] + listindex[-2:]))
398 for listindex in CSV_NOCELL_TIME:
399     hhmss_NOCELL.append(
400         float(listindex[-8:-6] + listindex[-5:-3] + listindex[-2:]))
401
402 # Convert to numpy array
403 TIME_ARR_5G = np.asarray(hhmss_5G)
404 TIME_ARR_LTE = np.asarray(hhmss_LTE)
405 TIME_ARR_LTE_A = np.asarray(hhmss_LTE_A)
406 TIME_ARR_NOCELL = np.asarray(hhmss_NOCELL)
407
408 # Add hhmss to data set arrays
409 CSV_5G = np.row_stack((CSV_5G, TIME_ARR_5G[None, :]))
410 CSV_LTE = np.row_stack((CSV_LTE, TIME_ARR_LTE[None, :]))
411 CSV_LTE_A = np.row_stack((CSV_LTE_A, TIME_ARR_LTE_A[None, :]))
412 CSV_NOCELL = np.row_stack((CSV_NOCELL, TIME_ARR_NOCELL[None, :]))
413
414 # Combine all arrays
415 ARR_COMBINED = np.column_stack((CSV_5G, CSV_LTE_A, CSV_LTE, CSV_NOCELL))
416
417 # Sort by hhmss in increasing order
418 SORTED_ARR_BY_TIME = ARR_COMBINED[:, ARR_COMBINED[6].argsort()]
419
420 #####
421 # Create KML files
422 #####
423 # Lines
424 kml_lines(data_arr=CSV_5G, bs_arr=BaseStations,
425           kml_filename="5G_MaritimeAntenna_ocean" + "_lines")
426 kml_lines(data_arr=CSV_LTE, bs_arr=BaseStations,
427           kml_filename="LTE_MaritimeAntenna_ocean" + "_lines")
428 kml_lines(data_arr=CSV_LTE_A, bs_arr=BaseStations,
429           kml_filename="LTE_A_MaritimeAntenna_ocean" + "_lines")
430 kml_lines(data_arr=CSV_NOCELL, bs_arr=BaseStations,
431           kml_filename="NOCELL_MaritimeAntenna_ocean" + "_lines")
432
433 # Heading
434 kml_arrow(SORTED_ARR_BY_TIME, DLON / 4, DLON / 4, "png/arrow2.png",
435          "MR_png_arrow")
436
437 # Trip visualization

```

```

438 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
439                     val_arr=CSV_5G[3 - 1, :], gps_arr=CSV_5G,
440                     thresholds=THRESHOLDS_RSRP,
441                     img_path_and_name="png/poyntings",
442                     legend_path_and_filename="png/RSRP_circle_LEGEND.png",
443                     kml_filename="telia_5G_RSRP_MaritimeAntenna_ocean")
444 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
445                     val_arr=CSV_LTE[3 - 1, :], gps_arr=CSV_LTE,
446                     thresholds=THRESHOLDS_RSRP,
447                     img_path_and_name="png/poyntings_square",
448                     legend_path_and_filename="png/RSRP_diamond_LEGEND.png",
449                     kml_filename="telia_LTE_RSRP_MaritimeAntenna_ocean")
450 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
451                     val_arr=CSV_LTE_A[3 - 1, :], gps_arr=CSV_LTE_A,
452                     thresholds=THRESHOLDS_RSRP,
453                     img_path_and_name="png/poyntings_square",
454                     legend_path_and_filename="png/RSRP_diamond_LEGEND.png",
455                     kml_filename="telia_LTE_A_RSRP_MaritimeAntenna_ocean")
456
457 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
458                     val_arr=CSV_5G[4 - 1, :], gps_arr=CSV_5G,
459                     thresholds=THRESHOLDS_RSRQ,
460                     img_path_and_name="png/poyntings",
461                     legend_path_and_filename="png/RSRQ_circle_LEGEND.png",
462                     kml_filename="telia_5G_RSRQ_MaritimeAntenna_ocean")
463 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
464                     val_arr=CSV_LTE[4 - 1, :], gps_arr=CSV_LTE,
465                     thresholds=THRESHOLDS_RSRQ,
466                     img_path_and_name="png/poyntings_square",
467                     legend_path_and_filename="png/RSRQ_diamond_LEGEND.png",
468                     kml_filename="telia_LTE_RSRQ_MaritimeAntenna_ocean")
469 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
470                     val_arr=CSV_LTE_A[4 - 1, :], gps_arr=CSV_LTE_A,
471                     thresholds=THRESHOLDS_RSRQ,
472                     img_path_and_name="png/poyntings_square",
473                     legend_path_and_filename="png/RSRQ_diamond_LEGEND.png",
474                     kml_filename="telia_LTE_A_RSRQ_MaritimeAntenna_ocean")
475
476 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
477                     val_arr=CSV_5G[5 - 1, :], gps_arr=CSV_5G,
478                     thresholds=THRESHOLDS_SINR,
479                     img_path_and_name="png/poyntings",
480                     legend_path_and_filename="png/SINR_circle_LEGEND.png",
481                     kml_filename="telia_5G_SINR_MaritimeAntenna_ocean")
482 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
483                     val_arr=CSV_LTE[5 - 1, :], gps_arr=CSV_LTE,
484                     thresholds=THRESHOLDS_SINR,
485                     img_path_and_name="png/poyntings_square",
486                     legend_path_and_filename="png/SINR_diamond_LEGEND.png",
487                     kml_filename="telia_LTE_SINR_MaritimeAntenna_ocean")
488 create_path_png_kml(lon_step=DLON / 2, lat_step=DLAT / 2,
489                     val_arr=CSV_LTE_A[5 - 1, :], gps_arr=CSV_LTE_A,
490                     thresholds=THRESHOLDS_SINR,
491                     img_path_and_name="png/poyntings_square",
492                     legend_path_and_filename="png/SINR_diamond_LEGEND.png",

```

```

493         kml_filename="telia_LTE_A_SINR_MaritimeAntenna_ocean")
494
495 create_path_no_cellular_png_kml(gps_arr=CSV_NOCELL, lon_step=DLO / 2.6,
496                               lat_step=DLAT / 2.6,
497                               img_path_and_fname="png/cross_red_circle.png",
498                               kml_filename="telianocell_ocean_trip")
499
500 # Grid visualization
501 SNAP_ARR = snap_coordinates_from_array(lat_arr=CSV_5G[0, :],
502                                       lon_arr=CSV_5G[1, :], lat_snap=SNAP_LAT,
503                                       lon_snap=SNAP_LON,
504                                       lat_step=DLAT, lon_step=DLO)
505 DATA2D = calc_z_snapgrid(data_array=CSV_5G, snap_array=SNAP_ARR,
506                           step_array=np.array([DLAT, DLO]))
507 create_grid_snap_png_kml(data2d=DATA2D, snap_arr=SNAP_ARR, step_lat=DLAT,
508                          step_lon=DLO, thresholds=THRESHOLDS_RSRP,
509                          img_path_and_filename="png/square_colored.png",
510                          legend_path_and_filename="png/RSRP_diamond_LEGEND.png",
511                          kml_filename="telia_5G_RSRP_MaritimeAntenna_ocean_grid")
512
513 SNAP_ARR = snap_coordinates_from_array(lat_arr=CSV_LTE[0, :],
514                                       lon_arr=CSV_LTE[1, :], lat_snap=SNAP_LAT,
515                                       lon_snap=SNAP_LON,
516                                       lat_step=DLAT, lon_step=DLO)
517 DATA2D = calc_z_snapgrid(data_array=CSV_LTE, snap_array=SNAP_ARR,
518                           step_array=np.array([DLAT, DLO]))
519 create_grid_snap_png_kml(data2d=DATA2D, snap_arr=SNAP_ARR, step_lat=DLAT,
520                          step_lon=DLO, thresholds=THRESHOLDS_RSRP,
521                          img_path_and_filename="png/v2_square_colored_LTE.png",
522                          legend_path_and_filename="png/RSRP_diamond_LEGEND.png",
523
524                          ↪ kml_filename="telia_LTE_RSRP_MaritimeAntenna_ocean_grid")
525
526 SNAP_ARR = snap_coordinates_from_array(lat_arr=CSV_LTE_A[0, :],
527                                       lon_arr=CSV_LTE_A[1, :],
528                                       lat_snap=SNAP_LAT, lon_snap=SNAP_LON,
529                                       lat_step=DLAT, lon_step=DLO)
530 DATA2D = calc_z_snapgrid(data_array=CSV_LTE_A, snap_array=SNAP_ARR,
531                           step_array=np.array([DLAT, DLO]))
532 create_grid_snap_png_kml(data2d=DATA2D, snap_arr=SNAP_ARR, step_lat=DLAT,
533                          step_lon=DLO, thresholds=THRESHOLDS_RSRP,
534                          img_path_and_filename="png/v2_square_colored_LTE.png",
535                          legend_path_and_filename="png/RSRP_diamond_LEGEND.png",
536
537                          ↪ kml_filename="telia_LTE_A_RSRP_MaritimeAntenna_ocean_grid")
538
539 SNAP_ARR = snap_coordinates_from_array(lat_arr=CSV_LTE_A[0, :],
540                                       lon_arr=CSV_LTE_A[1, :],
541                                       lat_snap=SNAP_LAT, lon_snap=SNAP_LON,
542                                       lat_step=DLAT, lon_step=DLO)
543 DATA2D = calc_z_snapgrid(data_array=CSV_LTE_A, snap_array=SNAP_ARR,
544                           step_array=np.array([DLAT, DLO]))
545 create_grid_snap_png_kml(data2d=DATA2D, snap_arr=SNAP_ARR, step_lat=DLAT,
546                          step_lon=DLO, thresholds=THRESHOLDS_RSRP,
547                          img_path_and_filename="png/v2_square_colored_LTE.png",

```

```
546         legend_path_and_filename="png/RSRP_diamond_LEGEND.png",
547
548         ↪ kml_filename="telia_LTE_A_RSRP_MaritimeAntenna_ocean_grid")
549 SNAP_ARR = snap_coordinates_from_array(lat_arr=CSV_NOCELL[0, :],
550                                       lon_arr=CSV_NOCELL[1, :],
551                                       lat_snap=SNAP_LAT, lon_snap=SNAP_LON,
552                                       lat_step=DLAT, lon_step=DLON)
553 DATA2D = calc_z_snapgrid(data_array=CSV_NOCELL[:, :], snap_array=SNAP_ARR,
554                           step_array=np.array([DLAT, DLON]))
555 create_grid_snap_png_kml(data2d=DATA2D, snap_arr=SNAP_ARR, step_lat=DLAT,
556                          step_lon=DLON, thresholds=THRESHOLDS_SINR,
557                          img_path_and_filename="png/v2_square_colored_LTE.png",
558                          legend_path_and_filename="None",
559                          kml_filename="telia_NOCELL_Maritime_antenna_ocean_grid")
```

18. Risk analysis - 04.02.2022

KARTLEGGING OG RISIKOVURDERING

Virksomhet/avdeling e.l.:
Bacheloroppgave om kartlegging av 5G nett
Ansvarlig leder:
Karl Gran Grodås

Skjema 1 av 3.
1: Kartlegging og risikovurdering
2: Risikodiagram
3: Handlingsplan

Bruk dette skjemaet til å dokumentere farer og problemer som er kartlagt. Vurder hvor ofte farene eller problemene inntreffer og konsekvens dersom det skjer. Sett også opp hvem som er ansvarlig for vurderingen og dato for når den ble gjort.

Nr.	Hva kan gå galt?	Beskriv konsekvensen hvis det skjer	Hvor ofte skjer det	Konsekvens	Kommentar	Vurdert av/dato
1	Strømgjennomgang fra 5V / 3.3V signalspenning.	Selv små strømmer kan være skadelig, og i verste fall dødelig, dersom de passerer gjennom kroppen.	Svært sjelden	Ubetydelig	5V /3.3V signalspenning er ikke kraftig nok til å penetrer huden. Risikoen er derfor neglesjerbar.	Karl Gran Grodås / 04.02.2022
2	Strømgjennomgang av 12V/24V batteri.	Strømgjennomgang fra 12/24V batteri kan foresake store skader, og i verstefall være dødelig, dersom det passerer gjennom kroppen.	Sjelden	Alvorlig	For å unngå kortslutninger må forhåndsregler tas når man bruker 12/24V batteri. Eksempler kan være å isolere alle kontaktpunkter, og å koble fra batteriet når det ikke er i bruk.	Karl Gran Grodås / 04.02.2022
3	Maritime Robotics velger å ikke samarbeide med oss.	Dersom Maritime Robotics ikke ønsker å samarbeide med oss får vi ikke testet systemet vårt i havet. Dette har ingen helsefarlige konsekvenser.	Sjelden	Mindre alvorlig	God og jevnlig kommunikasjon er viktig for å opprettholde samarbeid.	Karl Gran Grodås / 04.02.2022
4	Opptopping av nasjonale / lokale koronatiltak.	Dersom myndighetene avgjør at man må isoleres for å begrense spredning av koronasmitte må deltagerne jobbe hjemmefra. Dette vil påvirke arbeidsmiljøet og gjøre testing og samarbeid mer utfordrende.	Sjelden	Mindre alvorlig	Etter snart to år med korona er deltagerne godt vandt med å jobbe rundt koronaristreksjoner.	Karl Gran Grodås / 04.02.2022
5	En eller flere av gruppe medlemmene blir alvorlig syke over en lengre periode.	Dersom en av gruppe medlemmene blir syke vil gruppen miste nyttig kompetanse, og arbeidsmengden for de resterende vil bli større.	Svært sjelden	Mindre alvorlig	Alle gruppe medlemmene er i god helse, og er vaksinert mot COVID 19.	Karl Gran Grodås / 04.02.2022

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 1 av 2

Nr.	Hva kan gå galt?	Beskriv konsekvensen hvis det skjer	Hvor ofte skjer det	Konsekvens	Kommentar	Vurdert av/dato
6	Problemer med leveranse av nødvendig utstyr.	Dersom man ikke får tak i nødvendig utstyr vil det være nesten umulig å gjennomføre den praktiske delen av oppgaven.	Svært sjelden	Svært alvorlig	Det er ingen grunn til å tro at det skal være umulig å få tak i riktig utstyr.	Karl Gran Grodås / 04.02.2022
7	Ulykke under feltarbeid med MR	Ved bryggen til MR kan det være tungtransport eller andre fartøy som kan utgjøre en trussel for gruppe medlemmene.	Sjelden	Alvorlig	Gruppen har liten kunnskap til forholdene på kaien. Dette må oppklares med MR på forhånd.	Karl Gran Grodås / 18.02.2022
			Klikk for å velge	Klikk for å velge		

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 2 av 2

HANDLINGSPLAN

Virksomhet/avdeling e.l.:
Bacheloroppgave om kartlegging av 5G nett

Ansvarlig leder:
Karl Gran Grodås

Skjema 3 av 3.
1: Kartlegging og risikovurdering
2: Risikodiagram
3: **Handlingsplan**

Dokumenter tiltak for å redusere risikoen. Bruk samme nummerering som i skjema for kartlegging og risikovurdering og risikodiagrammet. Farer eller problemer som inntreffer ofte/svært ofte med en alvorlig/svært alvorlig konsekvens må prioriteres først.

Nr.	Kort beskrivelse av faren/problemet	Prioritering	Tiltak for å redusere risikoen	Ansvarlig(e)	Tidsfrist
1	Strømgjennomgang fra 5V / 3.3V signalspenning.	3	Ingen tiltak nødvendig	Karl Gran Grodås	-
2	Strømgjennomgang av 12V/24V batteri.	1	Sette opp rutiner for sikker bruk av batteriet.	Karl Gran Grodås	04.03.2022
3	Maritime Robotics velger å ikke samarbeide med oss.	2	Undersøke med oppdragsgiver om de har vært i kontakt med Maritime Robotics.	Johan Vu	08.02.2022
4	Opptrapping av nasjonle / lokale koronatiltak.	3	Som medborgere er deltagerne pliktet til å følge anbefalinger fra myndigheter når det gjelder begrensning av smitte. Utover det er det ingen tiltak gruppen kan gjøre.	Alle	04.03.2022
5	En eller flere av gruppelemmene blir alvorlig syke over en lengre periode.	3	Alle gruppelemmene er i god helse og det er ingen grunn til å anta at noe alvorlig skal skje. Som medborgere er deltagerne pliktet til å følge anbefalinger fra myndigheter når det gjelder begrensning av koronasmitte. Ellers ønsker ikke gruppen å innføre frihetsberøvende tiltak for å ivareta gruppelemmehelse.	Alle	04.03.2022
6	Problemer med leveranse nødvendig av utstyr.	2	Selv om det ikke finnes noen grunn til å tro at det skal være problemer med å tak i nødvendig utstyr ønsker gruppen å følge opp prosessen med innkjøp og transport for å forsikre seg om at alt går etter planen.	Michael Berg	04.03.2022
7	Ulykke under feltarbeid med MR	1	Det vil bli holdt møter med MR på forhånd for å forsikre at gruppelemmene har kjennskap til sikkerhetsrutinene rundt feltarbeid på kaien.	Johan Vu	04.03.2022

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 1 av 2

Nr.	Kort beskrivelse av faren/problemet	Prioritering	Tiltak for å redusere risikoen	Ansvarlig(e)	Tidsfrist

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 2 av 2

RISIKODIAGRAM

Virksomhet/avdeling e.l.:
Bacheloroppgave om kartlegging av 5G nett

Ansvarlig leder:
Karl Gran Grodås

Skjema 2 av 3.
1: Kartlegging og risikovurdering
2: Risikodiagram
3: Handlingsplan

Plasser farer og problemer i skjemaet basert på vurderingen av hvor ofte de inntreffer og hvor alvorlige de er. Bruk samme nummerering som i skjema for kartlegging og risikovurdering.

		RISIKODIAGRAM			
		Ubetydelig	Mindre alvorlig	Alvorlig	Svært alvorlig
Samsynlighet	Svært ofte				
	Ofte				
	Sjelden		3,4	2,7	
	Svært sjelden	1	5		6
		Ubetydelig	Mindre alvorlig	Alvorlig	Svært alvorlig

Malen er utarbeidet av Arbeidstilsynet – september 2017.

19. Risk analysis - 24.03.2022

KARTLEGGING OG RISIKOVURDERING

Virksomhet/avdeling e.l.:
Bacheloroppgave om kartlegging av 5G nett

Ansvarlig leder:
Karl Gran Grodås

Skjema 1 av 3.
1: Kartlegging og risikovurdering
2: Risikodiagram
3: Handlingsplan

Bruk dette skjemaet til å dokumentere farer og problemer som er kartlagt. Vurder hvor ofte farene eller problemene inntreffer og konsekvens dersom det skjer. Sett også opp hvem som er ansvarlig for vurderingen og dato for når den ble gjort.

Nr.	Hva kan gå galt?	Beskriv konsekvensen hvis det skjer	Hvor ofte skjer det	Konsekvens	Kommentar	Vurdert av/dato
1	Strømgjennomgang fra 5V / 3.3V signalspenning.	Selv små strømmer kan være skadelig, og i verste fall dødelig, dersom de passerer gjennom kroppen.	Svært sjelden	Ubetydelig	5V /3.3V signalspenning er ikke kraftig nok til å penetrere huden. Risikoen er derfor neglesjerbar.	Karl Gran Grodås / 04.02.2022
2	Strømgjennomgang av 12V/24V batteri.	Strømgjennomgang fra 12/24V batteri kan foresake store skader, og i verstefall være dødelig, dersom det passerer gjennom kroppen.	Sjelden	Alvorlig	For å unngå kortslutninger må forhåndsregler tas når man bruker 12/24V batteri. Eksempler kan være å isolere alle kontaktpunkter, og å koble fra batteriet når det ikke er i bruk.	Karl Gran Grodås / 04.02.2022
3	Maritime Robotics velger å ikke samarbeide med oss.	Dersom Maritime Robotics ikke ønsker å samarbeide med oss får vi ikke testet systemet vårt i havet. Dette har ingen helsefarlige konsekvenser.	Sjelden	Mindre alvorlig	God og jevnlig kommunikasjon er viktig for å opprettholde samarbeid.	Karl Gran Grodås / 04.02.2022
4	Opptrapping av nasjonale / lokale koronatilak.	Dersom myndighetene avgjør at man må isoleres for å begrense spredning av koronasmitte må deltagerne jobbe hjemmefra. Dette vil påvirke arbeidsmiljøet og gjøre testing og samarbeid mer utfordrende.	Sjelden	Mindre alvorlig	Etter snart to år med korona er deltagerne godt vandt med å jobbe rundt koronaristreksjoner.	Karl Gran Grodås / 04.02.2022
5	En eller flere av gruppelemmene blir alvorlig syke over en lengre periode.	Dersom en av gruppelemmene blir syke vil gruppen miste nyttig kompetanse, og arbeidsmengden for de resterende vil bli større.	Svært sjelden	Mindre alvorlig	Alle gruppelemmene er i god helse, og er vaksinert mot COVID 19.	Karl Gran Grodås / 04.02.2022

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 1 av 2

Nr.	Hva kan gå galt?	Beskriv konsekvensen hvis det skjer	Hvor ofte skjer det	Konsekvens	Kommentar	Vurdert av/dato
6	Problemer med leveranse av nødvendig utstyr.	Dersom man ikke får tak i nødvendig utstyr vil det være nesten umulig å gjennomføre den praktiske delen av oppgaven.	Svært sjelden	Svært alvorlig	Det er ingen grunn til å tro at det skal være umulig å få tak i riktig utstyr.	Karl Gran Grodås / 04.02.2022
7	Ulykke under feltarbeid med MR	Ved bryggen til MR kan det være tungtransport eller andre fartøyer som kan utgjøre en trussel for gruppe medlemmene.	Sjelden	Alvorlig	Gruppen har liten kunnskap til forholdene på kaien. Dette må oppklares med MR på forhånd.	Karl Gran Grodås / 18.02.2022
			Klikk for å velge	Klikk for å velge		

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 2 av 2

HANDLINGSPLAN

Virksomhet/avdeling e.l.:
Bacheloroppgave om kartlegging av 5G nett
Ansvarlig leder:
Karl Gran Grodås

Skjema 3 av 3.
1: Kartlegging og risikovurdering
2: Risikodiagram
3: **Handlingsplan**

Dokumenter tiltak for å redusere risikoen. Bruk samme nummerering som i skjema for kartlegging og risikovurdering og risikodiagrammet. Farer eller problemer som inntreffer ofte/svært ofte med en alvorlig/svært alvorlig konsekvens må prioriteres først.

Nr.	Kort beskrivelse av faren/problemet	Prioritering	Tiltak for å redusere risikoen	Ansvarlig(e)	Tidsfrist
1	Strømgjennomgang fra 5V / 3.3V signalspenning.	3	Ingen tiltak nødvendig	Karl Gran Grodås	-
2	Strømgjennomgang av 12V/24V batteri.	1	Sette opp rutiner for sikker bruk av batteriet.	Karl Gran Grodås	04.03.2022
3	Maritime Robotics velger å ikke samarbeide med oss.	2	Undersøke med oppdragsgiver om de har vært i kontakt med Maritime Robotics.	Johan Vu	08.02.2022
4	Opptrapping av nasjonle / lokale koronatiltak.	3	Som medborgere er deltagerne pliktet til å følge anbefalinger fra myndigheter når det gjelder begrensning av smitte. Utover det er det ingen tiltak gruppen kan gjøre.	Alle	04.03.2022
5	En eller flere av gruppe medlemmene blir alvorlig syke over en lengre periode.	3	Alle gruppe medlemmene er i god helse og det er ingen grunn til å anta at noe alvorlig skal skje. Som medborgere er deltagerne pliktet til å følge anbefalinger fra myndigheter når det gjelder begrensning av koronasmitte. Ellers ønsker ikke gruppen å innføre frihetsberøvende tiltak for å ivareta gruppe medlemmene helse.	Alle	04.03.2022
6	Problemer med leveranse nødvendig av utstyr.	2	Selv om det ikke finnes noen grunn til å tro at det skal være problemer med å tak i nødvendig utstyr ønsker gruppen å følge opp prosessen med innkjøp og transport for å forsikre seg om at alt går etter planen.	Michael Berg	04.03.2022
7	Ulykke under feltarbeid med MR	1	Det vil bli holdt møter med MR på forhånd for å forsikre at gruppe medlemmene har kjennskap til sikkerhetsrutinene rundt feltarbeid på kaien.	Johan Vu	04.03.2022

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 1 av 2

Nr.	Kort beskrivelse av faren/problemet	Prioritering	Tiltak for å redusere risikoen	Ansvarlig(e)	Tidsfrist

Malen er utarbeidet av Arbeidstilsynet – september 2017.

Side 2 av 2

RISIKODIAGRAM

Virksomhet/avdeling e.l.:
Bacheloroppgave om kartlegging av 5G nett
Ansvarlig leder:
Karl Gran Grodås

Skjema 2 av 3.
1: Kartlegging og risikovurdering
2: **Risikodiagram**
3: Handlingsplan

Plasser farer og problemer i skjemaet basert på vurderingen av hvor ofte de inntreffer og hvor alvorlige de er. Bruk samme nummerering som i skjema for kartlegging og risikovurdering.

		RISIKODIAGRAM			
Samshynghet	Svært ofte				
	Ofte				
	Sjelden		3,4	2,7	
	Svært sjelden	1	5		6
		Ubetydelig	Mindre alvorlig	Alvorlig	Svært alvorlig
		Konsekvens			

Malen er utarbeidet av Arbeidstilsynet – september 2017.

20. Risk analysis - 24.03.2022

RISIKOVURDERING AV HMS-FORHOLD (aktivitet, reise, lab/verksted, felt, utstyr, etc.)

[Obligatoriske* felt kopieres over til skjema]

Organisatorisk enhet*
[Institutt/avdeling/faglag]

SINTEF Digital

Lokasjon*
[Hvor er utstyret plassert? Hvor skal aktiviteten foregå?]

Gløshaugen og Brattøra

Utstyr / Aktivitet*
[navn på utstyrsaktivitet]

Gjennomføring av tester på Gløshaugen (på sykkel) og Brattøra (USV) (Maritime Robotics)

Linjeleder*
[nærmeste ansvarlige linjeleder]

Bengt Holter

Prosjektnummer-navn
[skal fylles ut ved prosjekt]

Bacheloroppgave

Prosjektleder
[skal fylles ut ved prosjekt]

Are Hellandsvik

Dato*
[dato for gjennomføring av risikovurderingen]

29. april og 2. mai 2022

Vurdert av*
[navn på alle deltakere i risikovurderingen]

Are Hellandsvik, Bård Myhre
Michael Rizan Berg, Johan Toan Vu, Karl Gran Grodås

Legg inn bilder / skisser / lenker

Beskrivelse av utstyret / aktiviteten
[legg inn mer utfyllende beskrivelse av utstyret / aktiviteten som har betydning for vurdering av risiko. Eventuelt legg ved filer/lenke til filer]

Det skal gjennomføres dekningsmålinger av 5G-dekning med egenprodusert utstyr, både på sykkel (Gløshaugen) og på ubemannet båt (Trondheimsfjorden). Førstnevnte gjøres av studentene, mens sistnevnte håndteres av Maritime Robotics i samarbeid med studentene.

Vedlegg
[legg inn lenker til andre relevante dokument]

[This Photo by Unknown Author is licensed under CC BY](#)



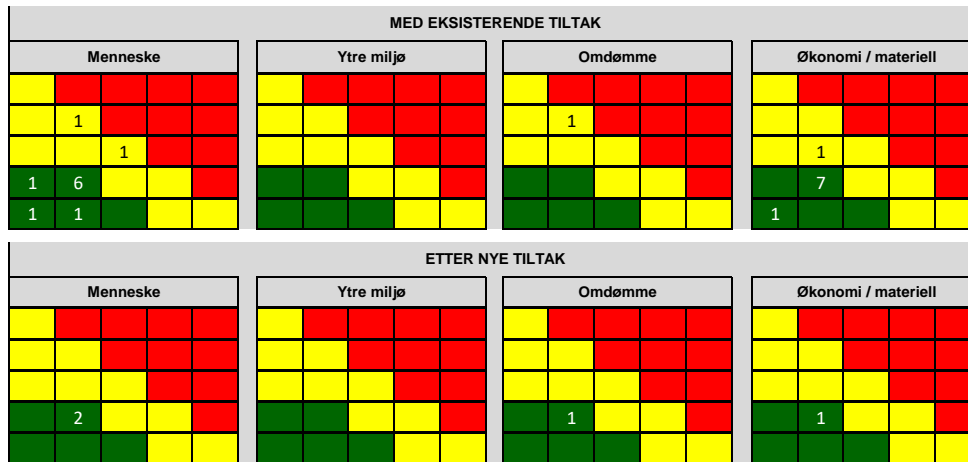
Organisatorisk enhet SINTEF Digital
Lokasjon Gløshaugen og Brattøra
Utstyr / Aktivitet Gjennomføring av tester på Gløshaugen (på sykkel) og Brattøra (USV) (Maritime Robotics)

Dato 29. APRIL OG 2. MAI 2022
Vurdert av ARE HELLANDSVIK, BÅRD MYHRE
Linjeleder BENGT HOLTER

Delaktivitet	Mulig uønsket hendelse	Eksisterende barrierer	Risikoene med eksisterende tiltak				Nye barrierer / risikoreducerende tiltak (handlingsplan)	Risikoene med nye tiltak				Ansvarlig	Dato	Status
			Personell	Utstyr	Materialer	Prosedyrer		Personell	Utstyr	Materialer	Prosedyrer			
Se fanen "Ledcord" for tips														
Trenger vi skjema for SJA?														
Trenger vi skjema for feltarbeid?														
Har MR i praksis HMS-ansvaret?														
Test på Gløshaugen 29. april 2022														
Forflytning fra kontor til sykkel	Ødelegge antennemast	Prosedyre: To stykker når man tar på/av søkk					B2							
	Fall i gang/trapp	Passer på hvor man går, modulen er i søkk					B2							
Gjennomføring av målinger på sykkel	Johan faller på sykkel og skader seg	Ingen	D2			Kjøpe og bruke sykkelhjelm	B2			Johan	29/04/2022			
	Johan faller på sykkel og utstyr blir ødelagt	Utstyret er i en boks, i en søkk.	B2											
	Kjøre på folk	Kjører i lav hastighet, i sykkelfelt	B1											
	Sykkeluhell pga. fokus på måleutstyr	Ingen	C3			Tiltak: Stå stille på sikkert sted når man ser på måleutstyr	B2			Johan	29/04/2022			
Værproblematikk - ikke aktuell 29. april.	Batteri begynner å trenne pga. utlading	Braker kun fulladet Dewalt-batteri som er laget for kommersielle verktøy.	B2											
	Forflytning fra sykkel til kontor	Se "Forflytning fra kontor til sykkel"												
Test på Brattøra 2. mai 2022														
Transport fra kontor til Brattøra	Kontor til bil	Mister utstyr, ødelegger utstyr				En bærer plate, en bærer mast, bruke heis	B2							
	Mister utstyr, skader seg på utstyr	En bærer plate, en bærer mast, bruke heis	B2											
I bil	Utstyr ødelegges i transport	Utstyr legges i bagesjerm				En bærer plate, en bærer mast, alt skjer på bakkeplan	B2							
	Mister utstyr, ødelegger utstyr	En bærer plate, en bærer mast, alt skjer på bakkeplan	B2											
Mister utstyr, skader seg på utstyr	Mister utstyr, skader seg på utstyr	En bærer plate, en bærer mast, alt skjer på bakkeplan	B2											
	Skader personell under montering/tilpassning	Manuelle tilpassninger gjøres under veiledningstilsyn av MR siden vi er i deres verksted; klareres av Are i forkant	B2											
Skader utstyr under montering/tilpassning	Skader utstyr under montering/tilpassning	Manuelle tilpassninger gjøres under veiledningstilsyn av MR siden vi er i deres verksted; klareres av Are i forkant					B2							
Gjennomføring av måling på USV	Folk faller i vannet under drift.	MR opererer alt som har med vann å gjøre, vi er tilskuere.	B2											
	Folk faller i vannet under felttesting	Felttesting gjøres på land /verksted	A1											
Vann på utstyr - utstyr ødelegges, og kan ikke benyttes	Vann på utstyr - utstyr ødelegges, og kan ikke benyttes	Har kapellet inn ting, og gjør målinger på land /forkant					B2							
	Brann i batteri													

Detaktivitet	Mulig uønsket hendelse	Eksisterende barrierer	Risikoen med eksisterende tiltak				Nye barrierer / risikoreducerende tiltak (handlingsplan)	Risikoen med nye tiltak				Ansvarlig	Dato	Status	
			Personell	Ytre miljø	Omdømme	Økonomi / materiell		Personell	Ytre miljø	Omdømme	Økonomi / materiell				
	USV begynner å brenne pga. utlading i batteri. Får ikke ønsket resultat til bacheloroppgaven	Braker kun fulladet Dewalt-batteri som er laget for kommersielle verktøy. Vi har planer om sykkelbæst i forkant					D2	Are informerer MR om denne risikoen, for ordens skyld.					Are	29/04/2022	
							A1						Johan	29/04/2022	
Demontering av måleutstyr på USV	Skader personell og utstyr under montering/tilpassning	Forsiktig demontering	A2				C2	Ta med håndklær og multimeter, for tørking av vann og sjekk mot jord				B2	Karl	29/04/2022	
Transport fra Brattøra til kontor	Se "Transport fra kontor til Brattøra"														

Utstyr/aktivitet Dato Gjennomføring av tester på Gløshaugen (på sykkel) og Brattøra (U 29. april og 2. mai 2022



21. Poster in A3 format (next page)

Development of an Independent and Mobile Measurement System for Geographical Mapping of 5G Coverage



Michael Rizan Berg
Karl Gran Grodås
Johan Toan Vu

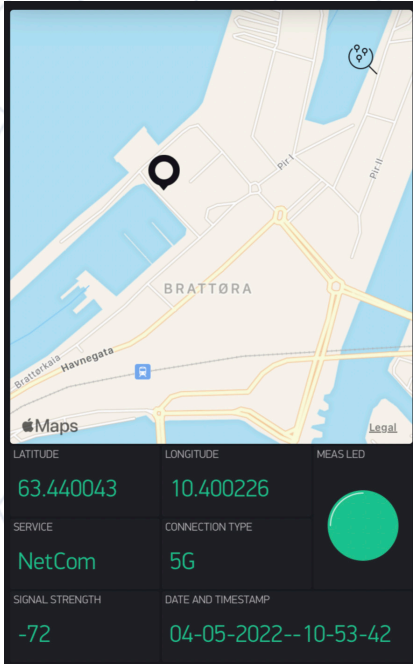
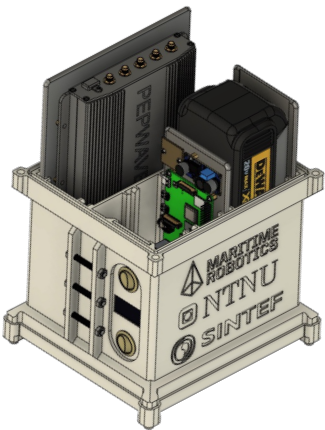
Bachelor in Electrical Engineering
Electronics and Sensor systems
NTNU Trondheim



Motivation

The project was motivated by the rapid expansion of the 5G cellular network. The configurability of 5G makes it adaptable for different use cases depending on the requirements of the consumer.

For many industries, using 5G will provide higher speed, lower latency and greater capacity than existing wireless solutions. An independent measurement system would give businesses the tools to make educated decisions based on recorded data. Maritime Robotics was incentivized to cooperate in this project due to their dependence on high-speed wireless for their control systems and live video streams. The final goal for this project was to develop an independent and mobile measurement system for geographical mapping of 5G coverage on land and at sea.



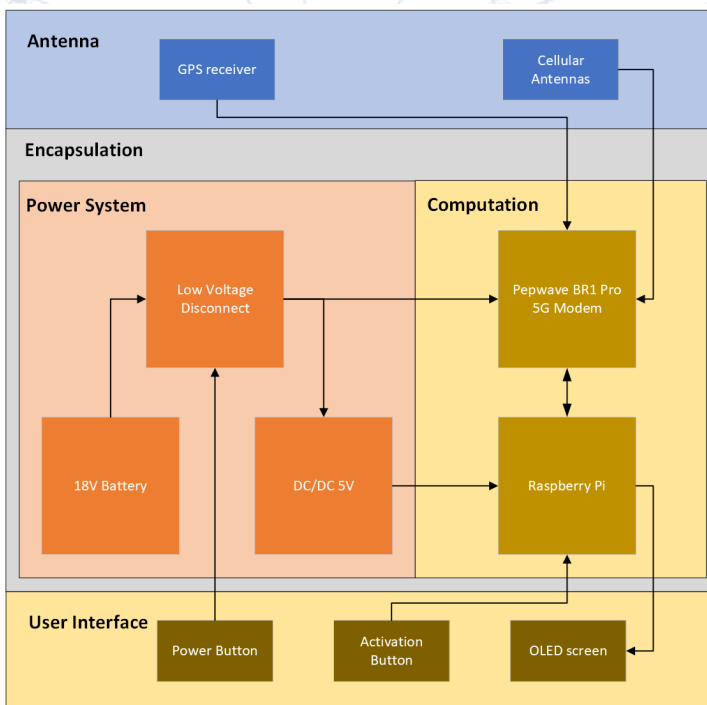
Method and Concept

The measurement systems concept as a whole consisted of three parts. The first part was the physical components, connections, and encapsulation that made up the hardware. The second was software, responsible for obtaining and storing measurement data. The third and last part was the visualization of the obtained measurement data. At first, there was conducted a series of tests in and around Gløshaugen. When these tests were successful the project moved on to the final test. The final test was to mount the system it on one of Maritime Robotics' USVs and obtain measurements in Trondheim Fjord. Using the data collected during the final test and the tests leading up to it, this project hopes to showcase the possibilities of the system and its many use cases.



Hardware

The measurement system consisted of several components that all needed to work together to achieve the goal of obtaining accurate measurements of the 5G cellular coverage. The system was divided into five subsystems. The five subsystems were *Power System*, *Computation*, *Antenna*, *User Interface*, and *Encapsulation*. The system was design to be completely modular to facilitate expansions in the future



Software

To obtain and save data from the modem, a python script was created to automate the process. The script enabled the SNMP protocol for communication with the modem and the measurements data was stored locally on the Raspberry Pi. Followingly, the GPS position at the measurement location was retrieved, thus acquiring the measurement systems position along with its measurement.



Coverage maps

After the measurement data was obtained, it had to be visualized. This was done in Google Earth Pro, with the use of *keyhole markup language* (KML). The coverage map to the left visualizes the measured *Reference Signal Received Power* (RSRP) at Gløshaugen. The coverage map to the right visualizes the measured *Signal to Interference and Noise Ratio* (SINR) in Trondheim Fjord.



