Master's thesis

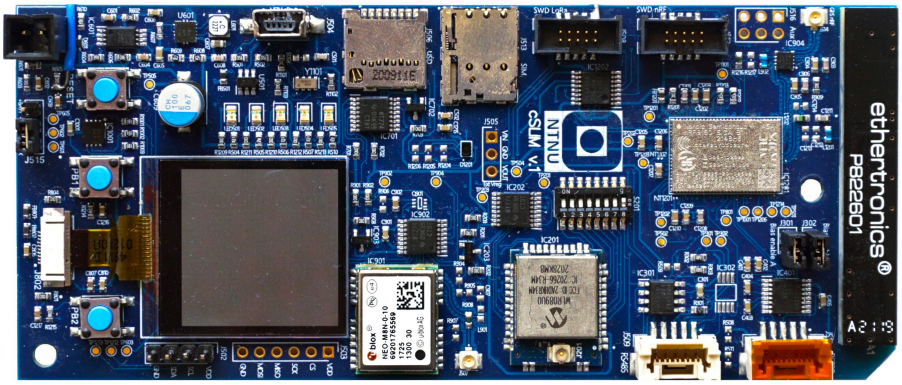
NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Vetle Berg Abrahamsen

Realization of an ultra-low-power embedded buoy controller for real-time acoustic telemetry monitoring

Internet of Fish

Master's thesis in Cybernetics and Robotics
Supervisor: Jo Arve Alfredsen
June 2022



NTNU
Norwegian University of
Science and Technology

Vetle Berg Abrahamsen

Realization of an ultra-low-power embedded buoy controller for real-time acoustic telemetry monitoring

Internet of Fish

Master's thesis in Cybernetics and Robotics
Supervisor: Jo Arve Alfredsen
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis aims at realizing a practical and robust embedded buoy controller (cSLIM-v1) by adopting the shield prototype for the Internet of Fish (IoF) project at NTNU for the purpose of wirelessly transmitting behavioral data of smolt during the migration or for fish farms in real-time to a backend visualization solution by utilizing acoustic telemetry receivers (UAR) and Low Power Wide Area Network (LPWAN), namely NB-IoT. Knowing when, how, and triggering factors for smolt migration is essential to understanding how environmental changes affect wild salmon. A growing population with increased awareness of sustainability and exploited resources and growth in the aquaculture industry forces new farming sites offshore, where inspection, accessibility, and operation are limited. The IoF project enables remote monitoring in near real-time and provides vital data as fish movement carries valuable insights into the welfare and response to the environment. However, battery life is a limiting factor for the practicality of the system, and implementation of LPWAN support and positional awareness using GNSS comes at the cost of current consumption and a reduced cycle of operation. To address this, cSLIM-v1 uses an ultra-low-power real-time clock (RTC) for timekeeping and a time-synchronization algorithm that estimates the actual RTC frequency, corrects drift and increases the period between GPS updates, keeping the GNSS module primarily disabled. Field tests conclude that the system reliably detects smolt and forwards all telemetry to the visualization solution. Results show that the time-synchronization algorithm estimates the RTC frequency in outdoor environments and meets the timing constraints while increasing GPS update intervals. Further research should focus on improving the power solution and software's robustness and investigate the performance of the nRF9160 SiP's embedded GNSS receiver to possibly replace the currently used GNSS module to reduce production costs.

Sammendrag

Denne oppgaven tar sikte på å realisere en praktisk og robust bøyekontroller (cSLIM-v1) ved å ta i bruk prototypen for Internet of Fish (IoF) prosjektet ved NTNU med formål om å trådløst overføre atferdsdata fra smolt under utvandring eller for oppdrettsanlegg i sanntid til en backend-visualiseringsløsning ved å bruke akustiske telemetrimottakere (UAR) og Low Power Wide Area Network (LPWAN), nemlig NB-IoT. Å få kunnskap om når, hvordan og utløsende faktorer for utvandring av smolt er avgjørende for å forstå hvordan miljøendringer påvirker laksebestanden. En voksende befolkning med økt bevissthet om bærekraft og utnyttede ressurser samt vekst i havbruksnæringen tvinger fram nye oppdrettslokaliteter offshore, hvor inspeksjon og tilgjengelighet er begrenset. IoF-prosjektet muliggjør fjernovervåking i nær sanntid og gir viktige data ettersom bevegelse av fisk gir verdifull innsikt i velferd og fiskens respons på miljøet. Batterilevetiden er imidlertid en begrensende faktor for nyttigheten av en slik løsning og implementering av LPWAN-støtte og posisjonsbevissthet ved bruk av GNSS kommer på bekostning av strømforbruk og redusert driftssyklus. For å løse dette bruker cSLIM-v1 en sanntidsklokke (RTC) med ultralav effekt til å drive den lokale klokken med en tidssynkroniseringsalgoritme som estimerer den sanne RTC-frekvensen, korrigerer drift og øker perioden mellom GPS-oppdateringer som fører til en stort sett deaktivert GNSS modul. Feltetester konkluderer med at systemet pålitelig detekterer smolt og videresender all telemetri til visualiseringsløsningen. Resultatene viser at tidssynkroniseringsalgoritmen estimerer RTC-frekvensen i utendørsmiljøer og møter tids-kravene samtidig som GPS-oppdateringsintervallene økes. Ytterligere forskning bør sette søkelys på å forbedre batteriløsningens robusthet og programvarens pålitelighet. I tillegg bør nRF9160 SiPs innebygde GNSS-mottaker utforskes for å muligens erstatte den nåværende brukte GNSS-modulen og dermed redusere produksjonskostnadene.

Preface

This master's thesis is submitted as a part of the requirements for the master's degree at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology. The work presented in this thesis has been carried out under the supervision of Prof. Jo Arve Alfredesen at the Department of Engineering Cybernetics at NTNU. It continues the work done by Eivind H. Jølsgard, Marius Rundhovde, and others before them on the Internet of Fish project.

This master's thesis is a continuation of a specialization project I conducted in the autumn of 2021. In accordance with habitual practice, the specialization project have not been published. This means that important background theory and methods from the project report will be restated in full throughout this report to provide the best reading experience. Below is a complete list of the material from the specialization project.

- Chapter 3 - Background (Specifically sections 3.1 to 3.3 with some modifications. Section 3.5.1, 3.5.2 and 3.5.3 have been included with minor modifications. That is also the case for section 3.6.1)
- Chapter 4 - System overview (Except section 4.3)
- Chapter 5 - Buoy controller development (Section 5.1 to 5.4, section 5.5.1 with some modification for the newly assembled units)
- Chapter 7 - Software development and optimization (Section 7.1 with modifications)

During the specialization project, Prof. Alfredsen provided me with the prototype version, Two nRF9160 development kits and the available components used for the previous assembly, a power debugger, and a new Digilent Analog Discovery 2 digital oscilloscope as recommended by E. Jølsgard. Through Jølsgards thesis, I was provided with links to Github repositories that included all design files such as schematics, PCB layout for the prototype, and the existing application software, both for the cSLIM-shield and for the LoRa module that runs its separate software.

Unfortunately, as I was conducting my first long-lasting outdoors test during this thesis, all equipment was stolen. The thief got away with a TBR700-RT, one nRF9160 development kit, and the only assembled cSLIM unit at the time. Thanks to my Supervisor, Prof. Alfredsen, the Department of Engineering Cybernetics at NTNU, we were quickly able to acquire a newer acoustic receiver, TBLive, from Thelma BioTel. Due to financial support, we could assemble three new units of cSLIM-v1. The cSLIM-v1 stand-alone embedded buoy controller has been realized and optimized for outdoor operations. New and existing drivers have been worked on to optimize power consumption and robustness, emphasizing the acoustic receiver serial interface. The new IoF message format (cSLIM GPS cycle) has been defined to allow developers to analyze system performance in a back-end visualization solution.

I specifically want to thank co-student Jon A. Kornberg, working in parallel with the SLIM module, for taking the lead on developing the shared visualization environment and standardizing the storage format in InfluxDB for a solution that allows storage of IoF data regardless of the buoy controller version used. Unforeseen issues required re-prioritization and would not have been possible without Jo. A Kornbergs good work with the visualization solution. I would like to point out that I have made a dashboard specific to cSLIM-v1 with the required InfluxDB database storage standards. Hence, this gave me the conscience to write a chapter dedicated to the visualization solution.

The entire system has been deployed for a field test in Stryn accompanied by Prof. Alfredsen, Ph.D. candidate Nikolai Lauvås (working on the Fish Otter project https://otter.itk.ntnu.no/doku.php), and Biologist Henning A. Urke. A special thanks to H. Urke for being available with his boat day and night and for allowing us to deploy our buoys to gather data from the 200 fish captured and tagged earlier that spring.

I want to thank my Supervisor, Jo Arve Alfredsen, for allowing me to work on this project and always being available for questions, discussions and the countless number of batteries I received (and broke) during power debugging. I would also like to thank the technical staff at the Department of Engineering Cybernetics for giving me access to the electronics workshop with soldering stations and electronic microscopes whenever needed and for the construction of the deployment rig prior to the field test. At last, thanks to Telia for sending me free SIM cards with NB-IoT coverage provided by their IoT solution.

Vetle Berg Abrahamsen

Trondheim, June 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

From 1990 to 2018, the Food and Agriculture of the United Nations (FAO) calculated a 527% rise in global aquaculture production in addition to a 122% rise in seafood consumption during the same period FAO (2020). Wild and farmed seafood is one of Norway's most significant export industries. In 2019 1.5 million tonnes of seafood worth 30.8 billion NOK were exported and is assumed to increase in the future Moe et al. (2020).

The annual Norwegian Aquaculture Analysis of 2021 identified five megatrends that will affect the global food industry; a growing world population, increasing rate of digitization, growing middle class coupled with urbanization, increasingly health-conscious consumers, and more focus on sustainability and exploited resources Moe et al. (2022). While these megatrends suggest an increase in demand for seafood marked with focus on healthy and high-quality fish, it also pressures the aquaculture industry to develop new technology and find solutions to biological challenges.

In 2019 the annual Norwegian Aquaculture Analysis emphasized that growth depends on environmental factors and that sustainability will be the enabler for an increased production EYGM-Limited (2019). The Norwegian fish farming industry faces biological and environmental challenges. Water temperature, especially in the North Atlantic, might increase by 0.6 to 2 degrees Celsius in the upper 100-meter depth compared to the 1986-2005 average. The Fisheries and Aquaculture technical paper 2018 write that this might lead to increased bacteria challenges, increased harmful algae, and mass movement of wild fish to more suitable environments. These challenges might increase the risk of severe economic loss and even lead to ecosystem collapse due to oxygen loss Moe et al. (2022). The environmental footprint from farming salmon along the coast of Norway has increased over the years. The main challenges threatening the marine habitat are sea lice, escapes and pollution of the seabed.

In search of more sustainable solutions, the Norwegian government supports new technology and solutions by allowing sea farmers to apply for development licenses to test alternative and better methods further offshore. Offshore farming has the advantage that ocean currents and deeper waters naturally remove feed residue and feces, limiting the footprint of the wildlife and seabed along the coast. It is also believed to help with sea lice and sickness and provide better living conditions in terms of colder and cleaner water. However, further offshore, the harsher environment might impose stress on the fish and can cause higher mortality rates, lower welfare, and reduced profit. In addition, moving farming locations further offshore reduces the site's availability and makes service routines more expensive. As fish inspection provides valuable insight into how fish feed and their general living condition, new technology must be developed to provide a substitute.

During this master's thesis, a circuit board that solves some of the issues regarding inspection has been realized and field-tested. The circuit board is called cSLIM-v1 and is part of the NTNU-owned project Internet of Fish (IoF) and supervised by Professor Jo Arve Alfredsen. The device can detect and capture fish telemetry and allows behavioral data to be sent through the Internet of Things (IoT) to base stations in near real-time, allowing farmers and scientists to monitor fish in offshore facilities with reduced costs remotely. The vital challenge for this

application is power consumption. A final product needs to be highly power efficient to increase deployment time and thereby reduce the cost of utilizing the cSLIM solution.

This thesis sets out to realize the first stand-alone embedded buoy controller cSLIM-v1. cSLIM-v1 uses a cellular modem that utilizes existing infrastructure for forwarding IoF messages and can easily be deployed without any extra gateways - in contrast to the LoRa LPWAN technology. The device is based on the previously built prototype, cSLIM-Shield, by Eivind H. Jølsgard. His work showed impressive results in decreasing power consumption by utilizing a time synchronization technique that allows precise local timekeeping without the need for frequent polling of GPS time data. This is achieved by implementing a low-power and precise Real-Time Clock (RTC) and correcting any unwanted drift at calculated intervals by estimating the actual RTC frequency, allowing the GNSS receiver to be powered off most of the time to reduce power consumption. This thesis has focused on acquiring experience with using NB-IoT in a realistic scenario and providing proof of concept by deploying a fully operational buoy in a fjord in Norway with a simplistic yet powerful visualization tool for monitoring buoy status and fish telemetry. To achieve this, the application had to be seamlessly integrated with the new cSLIM-v1 hardware, securing an optimal power solution and to quality-check and optimize each subsystem to prevent unexpected behavior.

1.2 Previous work

The Internet of Fish concept was first introduced by Hassan et al. (2019), who aimed at realizing a system for monitoring and tracking fish in 3D using Time Difference of Arrival (TDoA). The system consisted of buoys with submerged hydrophones for receiving acoustic signals from acoustic transmitters attached to the fish. Fish telemetry would then be forwarded to a presentation application using Low Power Wide Area Network (LPWAN) technology. Ph.D. candidate Hassan first realized the hardware solution named SLIM. The solution was designed to interface with an acoustic receiver by receiving detections and synchronizing the internal clock using GPS time. IoF messages were transmitted to LoRa gateways, placed along the shore using radio links responsible for forwarding messages to the user over the internet, such as WiFi and Ethernet.

To improve the power efficiency of the SLIM buoy controller, Rundhovde (2020) developed the software further. As the crystal oscillator embedded in the SLIM module caused a temperature-dependent drift, Rundhovde measured the different oscillator frequencies at different temperatures to reduce the local time drift, increasing the time between GPS updates. Based on Rundhoveds improvements, The overall power consumption was improved from 26mA to 10mA with tag detections and 7mA without tag detections.

An application layer for displaying real-time fish telemetry was developed during Kjelsvik's master's thesis. He also aimed at implementing the TDoA-based positioning service for monitoring fish in 3D Kjelsvik (2019). During this thesis, a new back-end server has been implemented. However, the work of Kjelsvik will be important when implementations of the TDoA become relevant for further progress of the application layer.

A new version of the SLIM module was developed by master student Jølsgard during his master's thesis, cellular SLIM or cSLIM - shield Jølsgård (2021), a prototype that investigated the possibility of adapting NB-IoT. This cellular LPWAN technology does not require any extra infrastructure for connecting to the network other than existing cell towers. The new buoy controller adapted the Nordic Semiconductor's nRF9160 System in Package (SiP) that implements an embedded cellular modem and a GNSS receiver. Additionally, the nRF9160 is RTOS-ready, enabling a real-time operating system to handle the scheduling of tasks, fault handling, and driver service using Zephyre RTOS. Jølsgard implemented a low-drift and temperature compensated real-time clock (RTC) and a frequency estimator using a low-pass filter for precise timekeeping. Based on indoor testing, with constant temperature, Jølsgard could achieve up to 10 hours between GPS wakeups with a local time drift of only 300us.

1.3 Contributions

As the cSLIM-shield provided promising results with the buoy controller, it was decided to realize a stand-alone buoy controller cSLIM-v1 based on the same design and software developed by E. Jølsgard. The stand-alone version was further developed during the author's specialization project. The main goal was to design and realize a hardware module and verify functionality by adapting the existing software. The hardware solution is essentially the same design as the prototype, with minor improvements that will be further described in the lists below, in addition to other main contributions relevant to the hardware solution. During the author's master thesis, the focus has been to integrate the existing software solution into the new cSLIM hardware platform and to develop the software to deploy a proof of concept for field tests.

Hardware contributions

- Adding additional hardware to the design - In order to integrate the nRF9160 SiP to the buoy controller, additional hardware, including the nRF9160, have been included in the design. Surrounding hardware consists of passive components, a cellular and GPS antenna with antenna matching circuits for improving signal stability, SIM card connector with protective filter on data lines and a reset circuit.
- Improved design of the new buoy controller, cSLIM-v1 - Design challenges due to lack of general input and output pins were solved using an external GPIO expander interfacing with the microcontroller using I2C.
- Power solution - A comprehensive investigation of the power solution on the cSLIM-v1 that unveiled a weakness in the hardware originally planned battery, requiring a new power solution.
- Designed for future - An optional header for an external voltage regulator have been implemented in the design for allowing support of the new TB Live acoustic receiver that operates on 5-9V.

Software contributions

- Data visualization - Created the visualization environment consisting of Node-RED, Influxdb and grafana for visualizing IoF messages.
- Defined new IoF message - Defined the GPS cycle IoF message for transferring previous and next wake-up time, drift and current estimated frequency.
- Power-limiting optimization - Sequential initialization of tasks, especially on the cellular modem and GNSS tasks. Additionally the Transmissions using the cellular network are postponed during GPS due to the total voltage causing a voltage drop in the battery.
- Fault tolerance - Reconnecting to the MQTT broker if disconnected.
- Thelma Biotel serial interface improvements - The time synchronization task for the acoustic transmitters have been improved and optimized for reducing chance of collisions and to improve the synchronization between other devices.
- Encrypted transmissions - TLS support have been implemented to the system by loading the certificate to the nRF9160 embedded modem, allowing data to be safely transferred over the internet of things.
- New LED driver - A driver have been developed for using the new GPIO expander for controlling the light emitting diodes.
- Physical ID dip-switch - The LoRa module parser software have been implemented with functionality for reading the 8-bit dip-switch for giving each device an unique ID.
- Bug fixes - IoF message formatting bugs have been fixed. A bug in the FRAM driver have been resolved.

1.4 Outline

The following outlines the entire organization of the thesis. Each chapter starts with a brief introduction to the specific topic and outline. Chapter 2 introduces the topic of LPWAN, covering the two IoT technologies supported by the cSLIM hardware platform. Chapter 3 will thoroughly give the reader the required background information on the hardware used in the project, communication protocols, Internet of Fish message frames, the software solution, and development tools. Chapter 4 sets out to give the reader an overview of the full concept. The chapter covers the hardware architecture, network layer, and application layer. Chapter 5 covers the design development during the specialization project and the assembly of three new units during this thesis. The chapter is included to provide the user with the best reading experience, as the development is considered an important part of the project. New findings regarding post-design modifications have been included in this chapter. The power solution has required extensive research during this thesis and is provided in Chapter 6. The power solution directly affects the systems' robustness, investigates important findings, and implements actions to overcome the identified issues. Chapter 7 Focuses on giving information about the cSLIM application, its drivers, and tasks in addition to modifications and improvements during the thesis. The new visualization solution are explained in Chapter 8. This chapter also briefly introduces how the IoF messages are decoded and injected into the database, explaining the InfluxDB storage standard and showing the resulting Grafana dashboards for monitoring fish telemetry. The field test has been covered in Chapter 9 with discussed results. Additional results from lab tests are included in Chapter 10. The whole thesis is discussed in Chapter 11 in addition to discussing the system requirements. Chapter 12 concludes the report followed by further work in Chapter 13. Appendix A.1 gives a guide to flashing software to both the nRF9160 and LoRa module have been included. This guide includes the toolchains needed to build the software and the hardware debuggers used.

Chapter 2

Survey of LPWAN technology

For the Internet of Fish project, specifically the cSLIM buoy controller, it is particularly interesting to investigate and compare the differences between types of Low Power Wide Area Network (LPWAN) technologies. Connectivity and communication consume a considerable amount of power and could significantly reduce battery life. Hence, the technology should be both effective concerning power consumption and data transfer and should handle the harsh offshore environment and still provide stability at a low cost. Therefore, a survey on IoT/LPWAN technologies has been conducted to gather necessary information from research previously done in the field of IoT/LPWAN. Relevant information regarding the usage of IoT/LPWAN in marine monitoring operations, where current usage of LPWAN technologies in acoustic telemetry systems will be of particular interest, will be included in this chapter. The survey will be used to create an informative basis for future decisions regarding selecting one LPWAN technology over another. The survey will also explore the potentials as well as their pitfalls in different applications for each technology in order to be able to create realistic and reasonable field tests to compare their strength and weaknesses specifically for offshore marine monitoring. Section 2.1 will provide the theory and specifications for each selected technology. Section 2.3 will provide existing research on the field.

2.1 LPWAN technologies

As of 2021, four LPWAN technologies accounted for 96% of the global installed base of LPWAN-enabled active devices: Narrowband (NB)-IoT, leading with 47%, followed by Long Range (LoRa) with 36% and with Long-term evolution for machines (LTE-M) and Sigfox currently holding the third and fourth place respectively Pasqua (2021).

2.1.1 NB-IoT

Narrowband-IoT is a Narrow Band IoT technology first released and specified in Release 13 of the 3rd Generation Partnership Project (3GPP) in 2016. NB-IoT operates on the licensed frequency bands in coexistence with GSM (Global System for mobile communications) and LTE (Long-term evolution). The protocol uses Quadrature Phase Shift Keying (QPSK) modulation, providing excellent noise immunity and low error probability. The maximum payload length is 1600 bytes with a maximum data rate of 200 kbps download, and 140 kbps upload Herrero (2022).

NB-IoT supports Power Saving Mode (PSM) and allows the user equipment (UE) to stay registered to the network despite being in a deep sleep state. While in the deep sleep state, the UE is unreachable by the network, hence introducing a downlink latency. PSM mode uses a timer called TAU (Tracking Area Update) or T3412 that determines the PSM cycle and should be optimized for each application. A large TAU (maximum 413 days) could be advantageous for UE that only requires upload transmissions. In addition to PMS, NB-IoT also supports extended Discontinuous Reception Mode (eDRX). When using eDRX, the device cycles through an On-Duration, monitoring the downlink channel and an eDRX period where the device stops monitoring and saves power. The

monitoring of the control channel for downlink or uplink grant is known as paging Sultania et al. (2018). During eDRX cycles, the device is considered idle, and when a timer T3324 expires without any uplink or downlink activity, the device switches to PSM for additional power savings. Hence, the Active Timer (AT) or T3324 is a vital tuning parameter as it determines the time the modem will spend listening for paging, allowing the UE to receive data before entering deep sleep Sultania et al. (2018).

The actual power savings are much dependent on the optimization of the PSM timers. A long TAU decreases power consumption but increases the downlink delay. AT can be optimized to reduce the time the modem listens on the network after transmission before entering the deep sleep mode. The AT timer can be set to zero for devices that do not expect any downlink transmissions, hence not consuming power for paging. Completely disabling the paging time essentially means that some quality of service will be lost due to only allowing downlink acknowledge messages to be received immediately after an uplink transmission. Sultania et al. (2018) states that an AT timer increased from 0 to 30 seconds increases the energy consumption threefold and sevenfold for an uplink transmission interval of 600 and 3600 seconds, respectively, and will be a valuable parameter for power optimization.

2.1.2 LoRa

LoRa (short for long range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology and was developed by Semtech (n.d.). Since the entire allocated bandwidth is used to broadcast a signal, the modulation technique used in LoRa makes it robust to channel noise. Additionally, the security of the LoRa system can be guaranteed as the transmissions present like noise due to a pseudo-random spread Zourmand et al. (2019). However, LoRa is the physical layer of the LoRaWAN - Long Range Wide Area Network and is categorized as an LPWAN technology. LoRaWAN is the media access control (MAC) layer protocol built on top of LoRa, the physical layer, that defines a network architecture that operates on the license-free sub-GHz band Zourmand et al. (2019).

Five essential parameters should be considered when deploying a LoRaWAN solution; carrier frequency (868MHz in Europe), transmit power, spreading factor(SF), code rate (CR), and bandwidth (BW)Zourmand et al. (2019). Bandwidth is the maximum frequency minus the minimum frequency of the ISM band it operates on. SF is the spreading factor of the chirp and indicates how fast the frequency changes within the minimum and maximum frequency (bandwidth) Jølsgård (2021). CR is the transmission's coding rate and gives the number of transmitted bits out of the total number of bits that carry information. The other bits are used for error correction.

Devices using LoRaWAN can be divided into three different classes LoRa-Alliance (n.d.b).

- **Class A - Lowest power, bi-directional end-device**: This is the default class. The end device always initializes communication. Uplink transmissions can be sent at any time and are followed by two downlink windows making the class bi-directional. This class has no requirements for periodic wakeups and can be put to low-power sleep for as long as designed in the application.
- **Class B - Bi-directional end-device with deterministic downlink latency** In addition to the downlink window found in class A after an uplink transmission, Class B uses periodic wakeups (programmable up to 128 seconds), allowing downlink "ping slots" at scheduled times. This increases the power consumption on the end device but decreases the downlink latency for applications where this is needed.
- **Class C - Lowest latency, bi-directional end-device** In addition to the two downlink windows after an uplink transmission as found in Class A, In Class C, the downlink latency is further reduced by always being able to receive data whilst the device is not transmitting, making the system half-duplex. This class is suited for devices with continuous power or for applications needing firmware over-the-air as the class can be temporarily switched between A and C.

2.2 Comparison of LoRaWAN and NB-IoT specifications

Previous studies regarding comparisons of different LPWAN technologies are many. This section and comparison are based on Mekki et al. (2019). The most relevant specification comparisons from Mekki et al. paper will be listed in Table 2.1

Table 2.1: Comparison of LPWAN specifications based on Mekki et al. (2019). Geolocation for NB-IoT was information found in Pelaez (2020)

Comparison of NB-IoT and LoRaWAN		
	LoRaWAN	**NB-IoT**
Modulation	CSS	QPSK
Standardization	LoRa-Alliance	3GPP
Frequency	Unlicensed ISM bands	Licensed
Bandwidth	250kHz and 125kHz	200kHz
Max data rate	50kbps	200kbps
Max payload length	243 bytes	1600 bytes
Range	5km(urban), 20km(rural)	1km(urban), 10km(rural)
Inteference immunity	Very high	Low
Geolocation	Yes (TDOA)	Yes (In 3GPP Rel 14)

Many factors must be considered when choosing the appropriate LPWAN for a solution. In his paper, Mekki et al. (2019) has done an excellent job explaining and comparing these factors for both NB-IoT, LoRaWan and Sigfox. Due to the limited uplink payload length of 12 bytes in addition to the limited 140 messages a day, Sigfox will not be considered in this survey. However, Kais Mekkis's discussion of NB-IoT and LoRaWAN will be summarised for the rest of this section.

Quality of Service QoS Is the measurement of network performance and is achieved by using a mechanism or technology to control traffic and reliability. LoRa, operated on the license-free spectrum with an asynchronous communication protocol, offers a lower QoS than NB-IoT due to NB-IoT being operated on the licensed spectrum with an LTE-based synchronous protocol that is optimal for QoS at the expense of cost. Therefore, for applications that require guaranteed QoS, NB-IoT is preferred over LoRaWAN.

Battery life & Latency Using LoRa and NB-IoT, the devices are in sleep mode most of the time, reducing energy consumption. However, due to the synchronous communication LTE protocol, QoS handling NB-IoT generally consumes more power than LoRa. In addition, NB-IoT utilizes the single-carrier frequency division multiple access (FDMA) for uplink transmissions and orthogonal FDMA (OFDMA) in the downlink that requires more peak current. Therefore, NB-IoT and LoRaWAN Class C are preferred for applications with low latency. Nevertheless, LoRaWAN class A is the best option for applications where latency is no issue.

Additional data regarding current consumption was found to support this factor. Although different chips offer different performance, these numbers will be considered a more general consumption. In Table 2.2 current consumption for receiving, transmitting, and peak currents will be listed for both NB-IoT and LoRaWAN and are based on the article written by Agustin Pelaez (2020).

Table 2.2: Power specification comparison between LoRa and NB-IoT Pelaez (2020).

Power specifications		
	LoRa	**NB-IoT**
Battery life	15+ years	10+ years
Peak current	32mA	120mA
Sleep current	1μA	5μA
TX current	24-44mA	74-220mA
RX current	12mA	46mA
Idle current	1.4mA	6mA

Scalability & Payload length Both LPWAN technologies support and tolerate an increasing number and density of connected devices. However, where immense scale deployment is required, NB-IoT allows up to 100K end devices per cell compared to 50K per cell for LoRa. NB-IoT also has the advantage of higher payload lengths up to 1600bytes compared to LoRa's maximum of 243 bytes. However, for smaller-scale solutions such as the IoF, neither of these factors becomes a limiting factor.

Network coverage & Cost LoRa has a more extensive range than NB-IoT but requires extra infrastructure such as LoRa gateways. Therefore, acquiring and setting up the additional infrastructure will have a higher cost than NB-IoT, that only requires the monthly network fees. However, having the possibility to set up the infrastructure wherever opens up possibilities for regions that do not benefit from LTE coverage, in addition to allowing private networks if desired.

2.3 LPWAN for marine monitoring

The exists a large number of papers that utilizes LoRaWAN specifically for marine monitoring. Lorenzo Parri et al. have published several papers on the topic. The first paper cited in this section was published in 2019 and discussed the realization of a LoRaWAN network infrastructure to be employed for monitoring purposes in marine environments in the specific context of an aquaculture industrial plant Parri et al. (2019). Lorenzo Parri et al. investigated different spreading factors and antenna heights. As a result, they achieved efficient data transmissions with a range of 8.33km, even using worst-case network settings, with two LoRaWAN gateways ashore, similar to the IoF concept. The paper concludes with an optimal spread factor (SF) of 7 since it ensured limited packet losses and satisfactory signal-to-noise ratio values in addition to the minimum power consumption compared with higher SF for the same packet length.

In 2020 Lorenzo Parri et al. published a new paper concerning LoRaWAN network infrastructure remote monitoring of offshore sea farms. The paper presents a complete real-time monitoring system to measure several parameters about the quality of water in the fish cages, as well as their maintenance status Parri et al. (2020).

In 2019 Song Yang et al. investigated the design and realization of a coastal acoustic tomography buoy based on the use of NB-IoT technology. The solution utilized an NB-IoT module (BC95) for reliable communication and a GPS module for precise timing Yang et al. (2019).

In 2020 Sofia Paiva released a paper on power consumption analysis of an NB-IoT end device for monitoring applications offshore. It showed promising results of using NB-IoT to forward data ashore. Sofia Paiva discovered possible problems with the NB-IoT modem concerning possible power solutions. It was found that the used power solution did not suffice the peak currents observed when establishing communication and resulted in a replacement to a battery capable of delivering on the current pulses Paiva et al. (2020). This paper is especially interesting as the same issue was discovered during this thesis resulting in the same conclusion of battery replacement.

Chapter 3

Internet of Fish: Background

3.1 Low Power Wide Area Network - LPWAN

A brief background infromation of Low-power wide area network (LPWAN), have been included to allow the reader to get a brief overview of the two LPWAN technologies used without having to read the LPWAN survey found in Chapter 2. LPWAN is a wireless network technology created for machine-to-machine and internet of things networks. LPWAN is a group of various network technologies serving the purpose of low-power wide area networking and enables greater power efficiency at a lower cost than traditional mobile networks Shea (2017).

3.1.1 Narrowband Internet of Things - NB-IoT

Narrowband Internet of Things (NB-IoT) is a wireless telecommunications technology standard developed by 3GPP, the same international standards body responsible for the LTE and 5G NR standards. This technology is designed to support IoT use-cases where low cost, strong coverage and low energy consumption are required Vos (2021). NB-IoT uses half-duplex communications, meaning that it is either transmitting or receiving data - not both simultaneously. Thanks to features such as Power Saving Mode (PSM) and Extended Discontinuous Reception (eDRX) in addition to NB-IoT's ability to optimize the amount of energy used for small data transmissions, NB-IoT reduces the power consumption when transmitting data by up to 75% compared to regular LTE Cat-1 modules Vos (2021). Due to NB-IoT being operated on the licensed spectrum, on existing LTE and GSM infrastructure, it offers up to 5-10 times better coverage than other cellular technologies Shea (2017) and allows service providers to quickly add cellular IoT connectivity to their services Vos (2021).

3.1.2 Long Range - LoRa

The unlicensed Long Range (LoRa), specified and backed by the LoRa Alliance, uses Chirp Spread Spectrum (CSS) technology to modulate transmissions and is robust against disturbances and provide long distance transmissions TheThingsNetwork (n.d.). LoRa can be operated on the licence free sub-gigahertz band. Doing so makes the technology less prone to interference and removing the need to pay frequency spectrum license fees to deploy a LoRaWAN network. LoRaWAN is the Media Access Control (MAC) layer protocol that is built on top of LoRa. LoRaWAN defines how devices utilize the LoRa hardware when transmitting and it also defines message formats TheThingsNetwork (n.d.). As LoRa is deployed on the license free band it does not allow continuous sending due to rules of the frequency bands de Ridder (n.d.). LoRaWAN is well suited for transmitting short and periodic payloads over long distances for as it offers ultra low power, long range and high capacity at a low cost.

3.2 Hardware

3.2.1 Acoustic telemetry system

Acoustic transmitters An acoustic transmitter or acoustic tag is a device that transmits data under water by emitting modulated sound waves. Tags are mainly used to transmit tag identification to a receiver and consist of a battery, modulator and transmitter. Some special tags may also include sensors such as depth, acceleration, inclination and temperature and can be fitted in capsules smaller than 10mm. Depending on power options, transmit intervals and size the acoustic transmitters are able to transmit data up to 1000 meters in radius for several months ThelmaBiotel (2021). Transmitters often implements random transmission time to avoid repeating collisions in applications with multiple transmitters.

Acoustic receivers Acoustic receivers are important devices in any fish tracking study and are responsible for detecting and decoding acoustic transmissions from fish tags. An acoustic receiver is a hydrophone that produces a small electrical signal when subject to changes in the underwater pressure, making it able to a produce and decode digital raw data from sound waves NOAA (2021). Receivers are designed to support multiple frequency channels in order to reduce tag collisions when using multiple transmitters ThelmaBiotel (2021).

3.2.2 LoRa gateway

Currently there exist several providers of LoRaWAN infrastructure, both public and community related. Still a private LoRaWAN network might be required where coverage is not supported. A LoRaWAN gateway implements a LoRa concentrator allowing the gateway to receive and transmit LoRa modulated RF signals from or to and LoRa end-node. The gateway do not process any of the data packages but checks if the integrity of the message is intact before forwarding the message to a LoRaWAN Network Server (LNS) through a network interface such as WiFi, Ethernet and GSM LoRa-Alliance (n.d.a).

3.2.3 Global Navigation Satellite System

Global Navigation Satellite System (GNSS) refers to a constellation of satellites surrounding the globe, providing global coverage of position and timing data to GNSS receivers. Four systems are available today, The American Global Positioning System (GPS) and Russian GLONASS, with operation started in the 1990s, were mainly used for military operations but have some services available for civil users. The Chinese BeiDou and the European Galileo, both fully operational closer to 2020, provides mainly services to civil users Kjerstad and Forssell (2021). Most receivers today are able to receive data from multiple GNSS systems, but receivers targeted at a specific GNSS system are also available.

3.2.4 nRF9160 System in Package and nRF9160 development kit

The nRF9160 System in Package (SiP) is a product developed by Nordic Semiconductor and is the first device in the nRF91 family to be released. Revision two of the SiP was released in xxxx, improving current consumption and increasing the modem performance. The SiP integrates a LTE-M/NB-IoT modem, supporting all power-saving features including eDRX and PSM for cellular IoT. nRF9160 SiP also integrates a GNSS receiver. The nRF9160 incorporates an Arm Cortex-M44 application processor with 1MB of flash and 256KB RAM, four peripheral drivers for SPI/UART/TWI, analog to digital converters and 32 general purpose input/outputs (GPIO), making the nRF9160 SiP capable of advanced application development in a single device solution, specifically designed for low power applications NordicSemiconductor (n.d.). Nordic Semiconductor provides development kits (DK) implemented with all the hardware components in order to test the nRF9160 SiP's capabilities and is commonly used during prototyping. The DK's embedded debugger also allows nRF9160DK to be used for programming/debugging of external nRF9160 SiPs in custom circuits.

3.3 Communication protocols and technology

3.3.1 RS232

RS232 is an asynchronous serial communication method where information is sent bit by bit. RS232 is a standard introduced in 1962 and have later been updated four times. The complete RS232 standard describes common voltage and signal levels, common pin-wiring configurations and a minimal amount of control information between the host and end-device in order to ensure compatibility between the host and peripheral system made by different manufacturers AnalogDevices (n.d.). The standard also specifies some functional characteristics such as slew-rate in order to reduce crosstalk between signals as slower rise and fall time reduces the chance of crosstalk AnalogDevices (n.d.). The interface consist of a transmit line (TX) and a receive line (RX) between two devices only. This allows for full-duplex communication, meaning that data can both be transmitted and received simultaneously without any chance of collision. However, RS232 is not very well protected against noise and should not be used where long distances is required. Maximum distance is 15 meters at 19.2kbps (kilo bit per second) and the maximum data rate is 1Mbps. The protocol differentiates between Mark (data 1) and Space (data 0) based on voltage levels on the signaling line AnalogDevices (n.d.).

3.3.2 RS485

RS485 is a serial communication method and a standard that defines the electrical characteristics of the driver and receiver during communication and listening operation. The RS485 protocol allows for longer cabling distance in noisy environments Kelly (n.d.). RS485 uses floated signals that are transmitted over a A(+) and B(-) signal line. The receiver compares the difference in voltage-level between both lines instead of using the absolute voltage with zero as a reference point. In addition, the wires should be twisted in order to spread the induced noise current from the magnetic field equally between both wires in the twisted pair. Non-twisted wires will have noise current flowing in the same direction (wire) at all times, making the interface less immune to noise Bies (2021).

Rs485 allows for multiple senders and receivers, up to 32 devices, hence only half duplex communication is guaranteed. The max distance according to the standard is 1200 meters, with 100k bits per second (bps) or 35Mbps max speed at 12 meter distance Bies (2021). In practice the bandwidth depends on the topology, in a common master-slave topology, where the slave device are only allowed to start data transmissions when told, almost 100 percent bandwidth can be used. But for applications where devices are able to start a data session on its own, collision can occur and it is advised to effectively use 37% of the bandwidth in addition to some higher level error detection implementations Bies (2021).

3.3.3 Serial peripheral interface - SPI

Serial peripheral interface (SPI) bus is a synchronous serial data bus. The bus consists of four lines, a serial clock (SCLK), two data lines - master in slave out (MISO) and master out slave in (MOSI) in addition to chip select signals. One benefit of the SPI protocol is that data can be transmitted without interruption in duplex mode, meaning that it can both receive and transmit data at the same time, allowing a continuous stream of data Campbell (n.d.b). This is due to the separate data lines, a slave and master relationship and the chip select signal, allowing only the selected device to use the bus. The data transmission is done bit by bit, synchronized with the serial clock driven by the master Campbell (n.d.b).

3.3.4 Inter-Intergrated circuit - I2C

Inter-intergrated circuit (I2C) is a communication protocol that uses two wires to transmit data between devices, serial data (SCA) and serial clock (SCL). The protocol is synchronous, meaning that the data is synchronized and sampled by the serial clock driven by the master Campbell (n.d.a). In contrast to SPI, the bus does not have a chip select signal. Instead a master sends an unique device address to the bus. The address is then compared within all separate devices and if a device has the given address it responds with an acknowledge bit on the bus line. With I2C, data is transferred using messages broken into separate frames of data. The first bit is the start condition followed by the slave address frame and read/write bit. The slave responds with an acknowledge bit before receiving the data frame(s). Transmission is stopped with a stop bit. The standard mode data rate is 100kbps, but faster data rates exist, the most common being fast mode, giving 400kbps Campbell (n.d.a).

3.3.5 Universal Asynchronous Receiver/Transmitter - UART

Universal Asynchronous Receiver/Transmitter (UART) is a hardware device inside integrated circuits (IC) or microcontrollers (MCU) that implements shift registers to convert parallel data from the controlling device into serial data and is converted back to parallel data by the receiving device. The communication protocol uses two wires, transmit TX and receive RX. The data flows from the TX pin from the transmitter device to the RX pin on the receiver Campbell (n.d.c). Since UART is asynchronous, it is not driven by a clock source, instead transmissions defines the start and stop of transmissions with start and stop bits. When receiving a start bit, the receiver starts to read the incoming serial data with the same baud rate as the transmitting device (10% baud rate tolerance) Campbell (n.d.c).

3.3.6 Message Queueing Telemetry Transport - MQTT

Message Queueing Telemetry Transport (MQTT) is a lightweight open-source communication protocol run on top of TCP/IP which provides security and reliability Catchpoint.com (n.d.). MQTT is specially designed to provide simple and reliable communication between multiple remote devices with low bandwidth and constrained resources, typically seen in IoT applications MQTT.org (n.d.). The protocol is bi-directional, allowing messages from device to cloud and from cloud to device. MQTT uses a publish/subscribe architecture, where a client publishes messages, e.g. sensor data, to a specific topic which can be accessed by an other client by subscribing to the same topic Catchpoint.com (n.d.).

3.3.7 Fish tag acoustic protocols

Acoustic tags are transmitters that emits data as sound waves under water. Due to water being close to incompressible, the sound waves propagates very well and can be transmitted for long distances, even with very low energy - making it possible to do scientific research on fish for several months on one small battery. No fluid is completely incompressible, hence signals will eventually be washed out or distorted by noise. This also means that the signal have a transmission delay that can be used to our advantage in order to calculate distance, speed of a moving object and even position in some applications Efteland (2016).

Multiple acoustic protocols are available today, some operates on different frequencies and carries different payload. for example some tags only transmitt tag identification (Tag ID) while other protocols enables senors data to be encoded into the payload such as pressure, temperature or other telemetry. Some protocols available today are "R256", "DS256", "R64K", "S64K", "HS256", "R04K" and "R01M". Newer open protocols have also been developed to seamlessly transmit and receive the most relevant data between different manufacturers. The message frame for tag detections will further describe the different protocols in Figure 3.4.

3.3.8 Thelma Biotel Serial Interface

Tag detections and status messages can optionally be sent from the acoustic receiver in real-time using RS485. As RS485 supports a duplex communication protocol, data can flow in both directions. However, the Thelma Biotel (TB) acoustic receiver system uses a topology where both the acoustic transmitter and the host can initialize a data session, in contrast to the typical master-slave topology, thus making it possible for data to collide, if not adequately handled. To overcome this problem, Thelma Biotel has defined a serial interface on top of RS485 with restrictions on when the acoustic receiver can transmit on the bus. Most importantly will the acoustic receiver be silent on the bus for one second, between 9.5 seconds and 10.5 seconds, to allow for undisturbed time synchronization commands and will not send anything on the bus until 10ms since the last received character. Note that this interface has been based on the TBlive acoustic receiver datasheet ThelmaBiotel (n.d.a) and might differ from previous acoustic receiver versions.

Table 3.1: Thelma Biotel acoustic receiver serial interface commands showing the expected response ThelmaBiotel (n.d.a).

Commands		
Command	**Functionality**	**Response**
?	Returns the serial number	SN=000745 ><>
(+)	Sets the clock to nearest 10 seconds	Ack01
(+)TTTTTTTTTC	Sets the clock to a specific unix timestamp (10 second precision)	Ack02

Table 3.1 shows the most essential commands used for the serial interface with Thelma Biotel's acoustic receivers. The last command allows setting the internal clock (Advanced synchronization), where TTTTTTTTT is a 9-digit timestamp using tens of seconds since Unix Epoch. C is Luhn's check digit calculated based on the 9-digit timestamp to validate the received time before the time is set.ThelmaBiotel (n.d.a) Theoretically, only one advanced sync will be needed when initializing the acoustic receiver. At the same time, the more simple synchronization command (+) can be used to keep the acoustic receiver in sync with the host's local time. In newer TB firmware, it will be possible to set the time using the complete 10-digit UNIX timestamp, giving the possibility to set the clock specific seconds. Additionally, more commands are listed in the Thelma Biotel Live datasheet ThelmaBiotel (n.d.a) but will not be listed in this thesis.

3.4 Internet of Fish message formats

Data transmitted from the end device over LPWAN is compressed by placing data in predefined message frames. Doing so makes the messages cost-effective since any unnecessary delimiters or strings are not compressed within the messages. On the receiving side, data is simply extracted using simple bit-shifting operations and placed in their respective variables. Messages have been divided into four types: bouy status, TBR status, tag detections, and GPS cycle messages. Every transmission contains both a header frame and a payload frame. The following subsections cover all the defined message formats.

3.4.1 IoF Header Frame

The header frame contains the serial number of the acoustic receiver and a reference timestamp. The reference timestamp allows sending multiple messages with a shared header. Additionally, a header flag is set to define the type of payload that follows. Header flag set to zero means that the payload contains either Tag detections or TBR status. The header flag set to 1 means that the received package contains bouy status, while the header flag set to 3 means that the payload is a GPS cycle message. Value of 2 is currently not used and is left for future use.

Offsets	Byte	0								1							
Byte	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	TBR/TB Live serial number [0, 16383]														Header flag [0,3]	
2	16	Reference timestamp (UTC) [0, 4294967296]															
4	32																
6	48	Payload (Buoy status / TBR status / Tag detection / cSLIM GPS cycle)															

Figure 3.1: Internet of fish header frame. Offset byte and bit set to zero as header is always sent first. Figure: Vetle B. Abrahamsen/NTNU

3.4.2 Buoy Status Frame

Buoy status frame is used to monitor the buoy. It contains essential information such as battery voltage, air temperature, and position of the buoy, in addition to information about the GPS such as fix, PDOP, and the number of tracked satellites to give some information about the certainty of the GPS coordinates.

Offsets	Byte	0								1							
Byte	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	48	Battery voltage [0, 127]								Air temperature [0, 127]						Lon ...	
8	64	Longitude [0, 67108863]															
10	80	... Longitude								PDOP [0, 127]							Lat ...
12	96	Latitude [0, 33554431															
14	112	... Latitude								Fix [0, 7]				Number of tracked satellite [0, 31]			

Figure 3.2: Internet of Fish buoy status frame. The offset byte is set to 6 as this is the first byte of the payload when combined with the header. Figure: Vetle B. Abrahamsen/NTNU

3.4.3 TBR Status Frame

The acoustic receiver is by default configured to send status messages every tenth minute and provides information about the water temperature, frequency of operation, and average and peak noise during the ten minutes. TB status and tag detection frames share the same header flag (zero) in the header frame. To separate TB status messages from tag detections code type is fixed to 255 in the TB status frame. Seconds since reference timestamp can be used to share header with other messages that have header flag set to zero. This means multiple messages can be sent and correctly timestamped using only one header, reducing the energy needed to transmit.

Offsets	Byte	0								1							
Byte	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	48	Seconds since reference timestamp [0, 255]								Code type							
8	64	Temperature [0, 65535]															
10	80	Noise average [0, 255]								Noise peak [0, 255]							
12	96	Frequency (kHz) [63, 77]								Upper timing error [0, 255]							

Figure 3.3: Internet of Fish TBR status frame. The offset byte is set to 6 as this is the first byte of the payload when combined with the header. Code type is fixed to 255 and upper timing error is currently not used. Figure: Vetle B. Abrahamsen/NTNU

3.4.4 TB Tag Detection Frame

The detection data is compressed into the TB tag detection frame whenever a tag has been detected. Since detections can occur rapidly, the first byte gives the seconds since the reference timestamp and makes it possible to transmit multiple detections in one transmission using only one header. Code type encodes both transmitting frequency and the acoustic tag protocol used in the detected tag. Code type is essential when decoding and

parsing the message into variables, as different protocols compress the message differently. Note that the value of 255 is reserved for TB status messages and can not be used in tag detection frames. The encoding algorithm will not be explained as it is a property of Thelma BioTel.

Depending on the protocol used by the detected tag, data will be compressed differently, as shown in Figure 3.4. The different protocol transmits different information and can not be generalized. Hence, the total length of the tag detection frame and the offset for signal-to-noise ratio (SNR) and milliseconds depends on the protocol used. SNR and milliseconds are by far the most interesting data received from detection; it can give valuable insight into the fish movement, estimated speed, and positioning in some applications.

Offsets	Byte	0								1							
Byte	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	48	Seconds since reference timestamp [0, 255]								Code type (both protocol number and frequenzy) [0, 254]							
8	64	Tag ID (all protocols)								Tag ID (protocol: R04K, R64K, R01M, S64K, HS256, DS256) or Tag payload (protocol: S256) or Not used (protocol: R256)							
10	80	Tag ID (protocol: R01M) or Tag payload (protocol: S64K, HS256, DS256) or Not used (protocol: R256, R04K, R64K, S256)								SNR [0, 60]				Milliseconds ...			
12	96	Milliseconds [0, 999]															

Figure 3.4: Internet of Fish tag detection frame. The offset byte is set to 6 as this is the first byte of the payload when combined with the header. Note: For multiple transmitted detection frames, the offset will increase with respect to the current frame length. Due to different protocols this frame length will vary. Figure: Vetle B. Abrahamsen/NTNU

3.4.5 cSLIM GPS Cycle Frame

A new IoF message format, namely GPS cycle, has been defined during this thesis. The information transmitted with this message frame provides essential information for optimizing the timekeeping task of the buoy controller. The message frame contains the local time drift relative to GPS time when synchronized and previous GPS sleep duration, current sleep time, and the estimated frequency of the real-time clock. Currently, only cSLIM-v1 supports GPS cycle message frames and have exploited the available header flag 3 that was reserved for future IoF messages.

Offsets	Byte	0								1							
Byte	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	48	Drift in microseconds during previous GPS cycle [0, 4294967295]															
8	64																
10	80	Previous sleep duration (minutes) [0, 65535]															
12	96	Current sleep time (minutes) [0, 65535]															
14	112	Estimated frequency (nHz) [0, 4294967295]															
16	128																

Figure 3.5: Internet of Fish cSLIM GPS cycle frame. Message frame are not fully compressed as some variables can be reduced in length. The offset byte is set to 6 as this is the first byte of the payload when combined with the header. Figure: Vetle B. Abrahamsen/NTNU

3.5 Software

3.5.1 Real time operating system - Zephyr

Zephyr Real-Time Operating System (RTOS) is part of the Zephyr project, a full-featured true open-source operating system optimized for resource-constrained devices. It enables the user to focus on application development as the Zephyr RTOS provides communication and peripheral drivers specifically designed for each supported device, a kernel responsible for scheduling and timing, and a watchdog that can detect software crashes and restart the system Zephyrproject.org (2021). A significant advantage of using RTOS in an embedded system is that it allows tasks to run concurrently, passing data between concurrent tasks through buffers, scheduled by the kernel. Tasks might also be assigned different priorities, allowing more essential tasks to be prioritized while less critical tasks are put to sleep Skøien (2021). Furthermore, due to being run by the Linux Fundation, a collaborative effort, the supported devices are continuously growing Zephyrproject.org (2021).

3.5.2 nRF connect SDK

nRF Connect SDK is a tool-chain manager with a set of open-source projects, samples, and a full suite of drivers for the nRF52, nRF53 and nRF91 Series devices maintained by Nordic Semiconductor. The SDK also includes the Zephyr RTOS, specifically developed for each device NordicSemiconductor (2019). During the development of the cSLIM shield prototype and during this thesis, nRF Connect SDK version 1.5.1 have been used. Newer versions have been published, and the tool-chain manager currently offers version 1.9.1, with updated Zephyr libraries.

3.5.3 Message Queuing Telemetry Transport broker

The MQTT broker is the heart of the publish/subscribe architecture used in the MQTT protocol. The broker is essentially a network router responsible for managing publisher/subscriber clients, receiving all published messages, filtering and forwarding to subscribed clients Catchpoint.com (n.d.). The MQTT broker is running virtually on an NTNU-hosted server named *otter01.it.ntnu.no* protected by password and username authentication and can be accessed using Secure Shell (SSH) in a terminal for configuration or managing purposes.

3.5.4 Node-RED

Node-red is a flow-based development tool developed by IBM and is used for wiring together different hardware devices, and network services as a part of the Internet of Things. Applications are made by dragging and dropping different "nodes" also known as "black box" that has a specific purpose such as a dedicated node for subscribing to MQTT topics or a node for writing custom functions using JavaScript OpenJS-Foundation (n.d.). The Web browser-based editor can be accessed through *https://otter.itk.ntnu.no:1880* using login credentials. In Internet of Fish, Node-red is responsible for decoding all IoF messages, placeing data in their respective variables and inject the data to InfluxDB, a database that can be reached by the backend data visualizer.

3.6 Development tools

3.6.1 Altium Designer

Altium Designer is one of the first PCB design tool providers developed by the Australian software company Altium Limited and officially launched in 2005 Altium.com (n.d.b). Altium Designer is an Electronic Computer-Aided Design (ECAD) tool set out to provide an environment supporting every aspect of the electronics design process with an ALL-inclusive design environment Altium.com (n.d.b). Altium Designer provides a Schematic editor, PCB editor, advanced auto-routing tools, 3D visualization, easy-to-use library management, and generation of production files in one solution. In addition, Altium offers the user the possibility to install community add-ons, allowing additional features Altium.com (n.d.a).

3.6.2 InfluxDB

InfluxDB is an open-source data source, commonly known as a database, developed by InfluxData and is optimized for high-availability retrieval of data faster and storage of time series data. One database consists of measurements, tags, fields, and points. Measurement is the same as a table and is usually created for each project. Tags are indexed columns within a measurement. Fields are non-indexed columns within a measurement, while points are rows within a measurement Nair (2021). The database structure makes it easy to retrieve data when visualizing the time series data. By carefully placing variables in the measurement as either field or tag, one can sort data based on specific characteristics, such as displaying only temperature of a sensor with a specific ID or to show coordinates to where a specific fish was detected to give some examples. InfluxDB is hosted locally on a NTNU server named *otter01.it.ntnu.no*.

3.6.3 Grafana

Grafana is an open-source web browser data visualizer. It is used for displaying telemetry and metrics from various data sources, where InfluxDB is one of the supported databases. Multiple dashboards can be created and tailored for specific projects and data. Dashboards are built by adding panels such as graphs, tables, or even world maps for visualizing position data. As Grafana queries stored data from a time-series database, it is possible to go back in time when analyzing data or to export data to CSV format from a given time interval. The Grafana is hosted locally on an NTNU server named *otter01.it.ntnu.no* and can be reached through *https://otter.itk.ntnu.no:3000* using login credentials.

3.6.4 LiveView

LiveView can be used to configure the acoustic receiver manufactured by Thelma Biotel. The acoustic receiver connects to the computer through an RS485 to USB adapter or Bluetooth. The software can be downloaded from Thelma Biotel's website by requesting a download link by mail. The software makes it possible to configure status message interval, listening frequencies and more. As the newer acoustic receiver TBLive does not include status LEDs that indicate when a tag is turned on, the software can be used instead to see when the acoustic transmitter has been successfully turned off or on.

Chapter 4

System overview

Smolts are captured and tagged using acoustic transmitters and will be monitored during their mitigation out to the ocean along the fjords. This thesis sets out to develop and show the potential of a new and revised buoy controller able to detect fish at fixed positions along the fjord of Norway in order to monitor their journey out to the ocean. Data will help biologists and researchers to understand the behavior and travel habits and provide the desired fish telemetry with a simple-to-use web browser visualization tool.

The physical part of the system is the buoy illustrated in Figure 4.1. A floating device equipped with an acoustic receiver and a buoy controller was developed during this thesis - cSLIM-v1. The bouy controlle sends tag detections, environmental data, and status messages on the network layer using either NB-IoT with existing cellular infrastructure or LoRa gateway, depending on the LPWAN technology. On the network layer, the transmissions will be forwarded to the Internet of Fish application through an MQTT broker responsible for providing subscribers with new data published on a topic. The Application layer consists of Node-Red, InfluxDB, and Grafana and provides the user with a visualization of all fish telemetry and data. Node-RED is the first entity to receive the data and subscribes to the MQTT topic. In Node-RED, all IoF messages are decoded and stored in InfluxDB. Grafana is then used to poll all new data automatically from the database and displays it in a web browser. Figure 4.2 shows the entire architecture of the solution. The following sections will further describe the perception, network, and application layers.

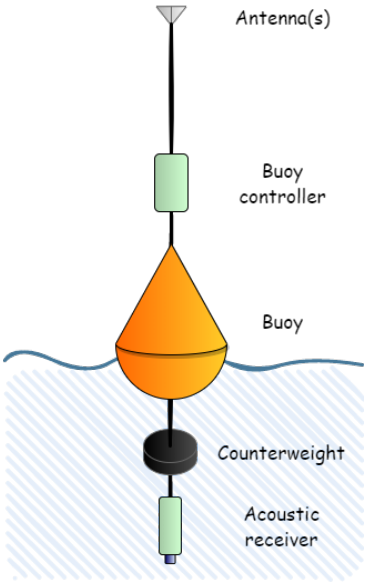


Figure 4.1: Illustration of deployed buoy. Figure: Vetle B. Abrahamsen/NTNU

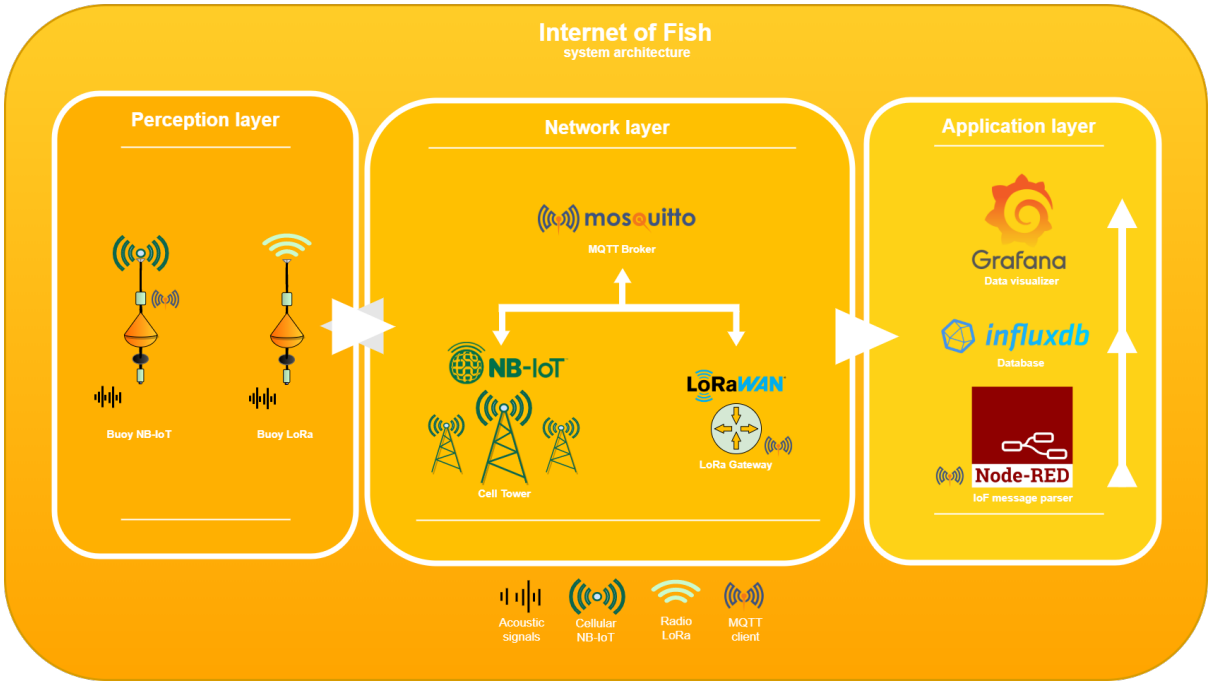


Figure 4.2: System overview: system architecture. Figure: Vetle B. Abrahamsen/NTNU

4.1 Perception layer

The perception layer is the physical part of the internet of fish project. The buoy consisting of a acoustic receiver, floating element, counterweight and a buoy controller will be deployed along the fjords in order to monitor and gather fish telemetry. Section 4.1.1 will cover the functionality and the hardware structure of the buoy controller while Section 4.1.2 covers the acoustic telemetry solution.

4.1.1 Buoy controller - cSLIM-v1

The standalone embedded buoy controller cSLIM-v1 was designed during the pre-project fall of 2021. The circuitry has been adopted from the prototype developed by E. Jølsgard in addition to some minor changes suggested by his master thesis. The cSLIM prototype was realized as a shield fitted to the nRF9160 development kit to test the nRF9160 SiP's functionality. Due to promising results in E. Jølsgard's thesis, the revised version cSLIM-v1 fully realizes the solution as a stand-alone circuit board. Hence, the new circuitry also integrates the nRF9160 SiP with required power components, reset circuitry, and both a GPS and LTE antenna to fully support the nRF9160 SiP's potential.

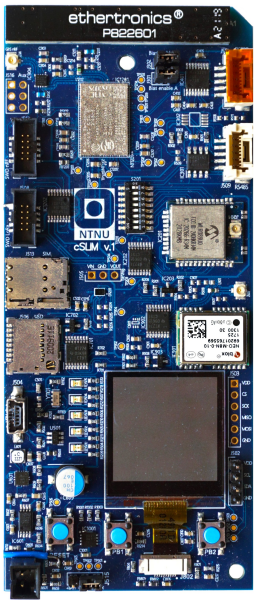


Figure 4.3: Buoy controller cSLIM-v1. Photo: Vetle B. Abrahamsen/NTNU

cSLIM-v1 utilizes an ultra-low-power, precise, and temperature-compensated real-time clock (RTC) to keep the buoy controller's local time as precise as possible. However, the RTC frequency is not ideal and will cause drift over time due to temperature changes and aging. A common solution is regularly synchronizing the local time with a precise time from a GPS. However, waking up the GNSS module consumes power and reduces the battery cycle drastically. Hence, E. Jølsgard introduced a solution where the RTC frequency is estimated and is used to correct the RTC frequency and calculate a new wake-up time. With a good frequency estimation, the GNSS module can be

kept disabled for hours, increasing the battery life while keeping the local time within its timing constraint of ± 500μs.

Keeping a precise local time is also essential for time-stamping tag detection. As the local time is used to set the acoustic receiver's internal clock, keeping the local time drift to a specified minimum ensures that other buoy controllers time-stamp the same detection with a known error margin. Thus, allowing more advanced estimations such as 3D location of the fish to be estimated with a known maximum error.

The cSLIM-v1 has been realized with LEDs for signaling and can be visible from a distance in addition to a display for showing important information to the user at site. The design also implements both support for uSD and FRAM such that data can be logged locally in case of network loss.

At last, cSLIM-v1 implements two LPWAN technologies, the cellular NB-IoT modem embedded in the nRF9160 SiP and a LoRa module for LoRaWAN. A deployed buoy is configured with one of the LPWAN options. The buoy controller interfaces with an acoustic receiver using RS485 and forwards all acoustic detections, status messages, and GPS cycle messages to the network layer.

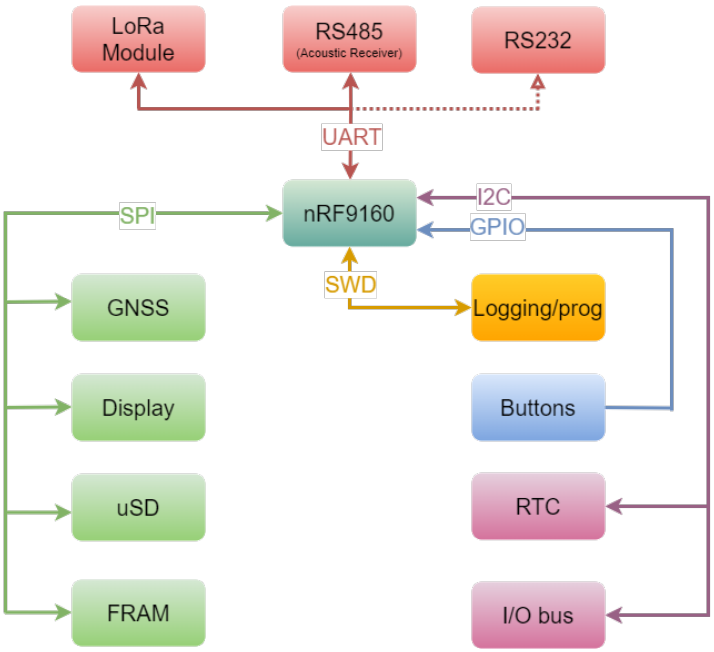
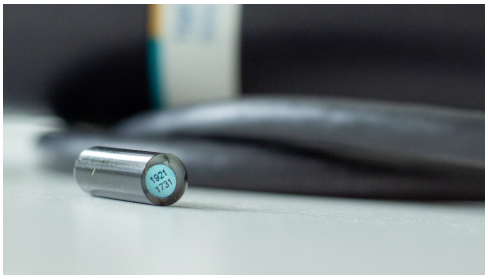


Figure 4.4: cSLIM-v1 Hardware modules and interface. Figure: Vetle B. Abrahamsen/NTNU

4.1.2 Acoustic telemetry solution

Thelma Biotel has been selected as the acoustic transmitter and receiver provider. The acoustic receiver previously used with the prototype was TBR 700 RT, shown in Figure 4.5a. With an internal battery, this receiver has a life span of up to 4 months. A newer version, TB Live, offers the same as TBR 700 RT but does not have an integrated battery. Hence, cSLIM-v1 was designed with optional voltage output powered by an external voltage regulator when needed. TB Live does not support internal local logging. Local logging is, however, supported on the buoy controller. Hence, TB Live is considered a good alternative for future buoy deployment and has also been used during this thesis due to stolen equipment. The acoustic tag is shown in Figure 4.5b and are a medium-size tag with a diameter of 9mm. Depending on the transmit-interval and configuration, the estimated battery life is between 7 to 27 months ThelmaBiotel (n.d.b).

(a) Acoustic receiver - TBR 700 RT



(b) 9mm acoustic transmitter (Tag)

Figure 4.5: Pictures illustrates the acoustic telemetry solution. Both the receiver and transmitter is from Thelma BioTel. TBR 700 RT includes battery, while TB Live (Not shown here) does not - giving a smaller form factor. Both models uses the same interface cable. Photo: Eivind Jølsgard/NTNU

4.2 Network layer

The network layer is responsible for carrying the IoF messages from the deployed buoy to the virtual Mosquitto MQTT broker running on a server at NTNU. Different infrastructure is needed depending on the LPWAN technology used with the deployed buoy. Section 4.2.2 describes the LoRaWAN solution and Section 4.2.3 describes the infrastructure needed in order to use NB-IoT. The MQTT broker is a shared solution for both NB-IoT and LoRaWAN and will be described in Section 4.2.1.

4.2.1 MQTT broker

A Mosquitto broker, running virtually on a virtual machine provided by NTNU, handles all traffic between end devices on the perception layer and the visualization solution on the application layer. In this thesis, the Mosquitto MQTT broker is responsible for reliably transmitting messages on the topic "p/806" and "s/806" respectively. cSLIM-v1 publishes all IoF messages to topic "p/806" and receives all messages such as acknowledged messages and ping-responses by subscribing to topic "s/806". Hence, the application layer can receive, parse and handle messages when subscribing to the topic "p/806". Furthermore, using the MQTT broker makes the IoT project scalable. New devices can be deployed to the network by subscribing to specific topics or providing a topic with new data as a publisher, given that all devices have unique IDs.

4.2.2 LoRaWAN infrastructure

As the LoRa module transmits IoF messages using a Radio link, extra infrastructure is needed in order to forward the messages to the IoF application layer. A LoRa gateway will be placed along the coast, receives IoF messages as radio signals, and puts the message in a TCP/UDP package to communicate with the back-end (The virtually runned MQTT broker) using both Ethernet and cellular networks such as 4G. A MultiTech Conduit LoRa Gateway shown in Figure 4.6 is used as a LoRa base station, creating the required infrastructure needed to transmit data through the internet.

Figure 4.6: MultiTech Conduit LoRa Gateway used as basestation. Photo: E. Jølsgard/NTNU

When using LoRa, the MQTT client is not initialized in the cSLIM application and is instead configured in the MultiTech Conduit. The Multitech is responsible for keeping the connection with the MQTT broker, publishing data to topics, or forwarding subscribed topics to the end device (cSLIM-v1) using radio-link. For communication and configuring the MQTT service, the Conduit runs Node-RED, a highly customizable development tool, making it possible to attach extra information to the IoF messages, such as signal strength to give an example.

4.2.3 NB-IoT infrastructure

NB-IoT's main advantage is that it does not require any additional infrastructure such as gateways since it operates on the existing cellular network. Hence, teleoperators are responsible for providing coverage. The MQTT broker needs to be initialized and configured in the cSLIM software for NB-IoT driven buoy controllers.

4.3 Application layer - Visualization

During this thesis, an application layer has been developed in cooperation with co-student J. Kornberg that focuses on the SLIM module. The application layer's main tasks are to parse incoming IoF messages in Node-RED, store all data in InfluxDB and visualize fish telemetry and other important data on Grafana. As the Internet of Fish project uses standardized message formats, it was natural to standardize the way data is stored in the database. This makes it possible to visualize all fish telemetry in one common dashboard without being specific to either cSLIM or SLIM modules.

4.3.1 Node-RED

Data is first received on the application level in Node-RED by configuring an MQTT client node listening to IoF messages published by the perception layer. As data is transmitted as compressed as possible, data is sorted and parsed to InfluxDB based on information in the message frames (see Section 3.4 for an in-depth description of all IoF message formats). The cSLIM Node-RED flow consists of seven nodes - one MQTT client node, one InfluxDB node for sending data to the database, and five custom function nodes for parsing the incoming IoF messages.

4.3.2 InfluxDB

Data injected and stored in InfluxDB should follow a standardized structure. The structure defines all stored data as either fields or tags. A measurement specific to cSLIM-v1 has been made for displaying GPS cycle data to the developer as local time-drift, GPS sleep period and estimated RTC frequency gives valuable insight for tuning and optimizing the sleep time of the GNSS module and local time precision. The InfluxDB standard for storing data can be found under Section 8.2.1.

4.3.3 Grafana

Grafana currently gives the user the possibility to monitor four different dashboards. Every dashboard is designed to give the user some specific type of information instead of displaying everything in one overwhelming dashboard.

Buoy data - Giving the user temperature data, both in air and water, a map with buoy coordinates, battery level in addition to some GPS stats such as GPS fix, PDOP and satellites in sight. The user can specify a specific TBR serial number in order to display the information of one specific buoy, multiple, or all. This dashboard are mainly covered by the buoydata measurement in InfluxDB in addition to water temperature from the tbrdata measurement.

TBR data - The dashboard shows the noise measured by the TBR and can be used to see if a tag detection with a low signal-to-noise ratio is due to large measured background noise or if the fish were swimming far away from the buoy. This dashboard is based on the tbrdata measurement in InfluxDB and can be sorted based on TBR serial number.

Tag detections - This is the main dashboard for IoF and provides the user with valuable information. Tag data, acoustic protocol information, timestamp and signal-to-noise ratio are displayed here. The tag detection dashboard can be sorted based on both tag ID and TBR serial number, meaning that the user will be able to monitor all detections of a specific ID or to monitor tag detections from a specific buoy.

cSLIM GPS cycle - Is meant to support the developer during the optimization and development of the cSLIM. The dashboard displays the current sleep time, previous sleep time, drift during the previous GPS sleep cycle, estimated RTC frequency based on the previous local time drift and sleep time. Three graphs show the time-series data of local time drift, sleep time and estimated frequency and are extremely useful when analyzing time synchronization performance. The dashboard can be sorted based TBR serial number.

Chapter 5

cSLIM-v1: Buoy controller development

the cSLIM-shield had previously been designed using KeyCad, a free electronics design tool. Due to experience in Altium Designer, it was decided to use this as development tool for the stand-alone version, introducing a lot of work converting existing schematics and component libraries from one environment to another. This section will start by highlighting the component selection in Section 5.1. In Section 5.2 development of schematics will be emphasized while PCB layout will be in focus in Section 5.3. Section 5.4 and 5.5 covers the manufacturing specifications and assembly of the additional units while Section 5.6 covers verification and post modifications of the cSLIM-v1.

5.1 Component selection

nRF9160 As the prototype showed promising results by utilizing the nRF9160 with the development kit, cSLIM-v1 integrates the nRF9160 SiP directly on the PCB. During the specialization project, the design was initially implementing the nRF9160 revision 1. However, during the assembly of new units, the nRF9160 SiP rev. 2 has been used as it offers optimized performance and current consumption. nRF9160 is a highly integrated low-power system-in-package that integrates both an LTE-M/NB-IoT modem and a GNSS receiver. Using power-saving modes, the power consumption can go as low as 2.7μA. The previous design did not utilize the inbuilt GNSS receiver due to lack of Pulse Per Second (PPS) support. However, during the design of cSLIM-v1, PPS support was discovered on the COEX1 pin on the nRF9160, given that the modem runs the firmware version 3 or later. Hence, 0Ω resistors have been included in the design to select between the existing PPS signal from the external GNSS module or the inbuilt GNSS receiver for future testing. SWD interface has been added to the design, enabling flashing of software on to the Arm Cortex M33 processor integrated in the nRF9160-SiP through an external SDW Arm-programmer.

The hardware integration guideline has been used carefully in order to provide a high quality integration of the nRF9160-SiP when deciding values for the passive components that provides power and noise reduction to the nRF9160-SiP. In addition, the nRF9160-DK has been used as a design reference when designing RF and GNSS antennas for the inbuilt modem and GNSS receiver. By designing cSLIM-v1's antenna layout similar to the nRF9160-DK design, the antenna matching network achieves approximately 50Ω impedance on the antenna line, which is required for optimal antenna integration.

GNSS LNA The new design of cSLIM-v1 aims to introduce support for the GNSS receiver embedded on the nRF9160. In nRF9160's antenna and RF integration manual, it was strongly recommended against designing the GNSS receiver without a Low-Noise Amplifier (LNA). The LNA is used for amplifying very low-power signals without degrading its signal-to-noise ratio. Not using LNA causes a sensitivity drawback of several dB Semiconductor (2019). SKY65943-11 is a low-power, low-noise amplifier front-end module with integrated pre-filter and post-filter. The module integrates output matching components inside the device and requires only one input matching inductor to be able to provide an internally matched impedance of 50Ω. The device has a low

current consumption of 2.9mA at 1.8 volts and can be enabled/disabled by a specific pin on the nRF9160. However, a newer version SKY55950-11, not available yet, are pin-compatible with the older version and reduces the current consumption to an impressive 1mA at 1.8 volts. A 9.1nH fixed inductor from Murata, LQW18AN9N1D00D, have been selected as the input matching inductor as suggested by the reference design and can be reused when the new SKY becomes available on the market.

SIM card The cSLIM-shield solution utilized the nRF9160-DK's onboard nRF9160 with surrounding hardware such as a SIM card. To be able to use the nRF9160's integrated LTE/NB-IoT modem, a SIM card connector had to be added to the design. The reset, clock and data line between the nRF9160 and SIM card connector is connected through an EMI filter for electrostatic discharge (ESD) protection. Due to the very small package size of the EMI filter with ten pins, the design includes bypass resistors of 0Ω that can be soldered in case soldering the filter is difficult.

I/O expander One of the issues with the prototype design was the lack of available GPIO pins on the nRF9160. Hence, LEDs and button signal was previously designed to be shared with other signals serving other purposes. The Low-power I/O expander PCAL6408A with eight inputs/outputs interfacing with the nRF9160 through the existing I2C bus was added to the design. This increased the amount of available GPIO and allowed buttons and LEDs to be driven independently from other signals. PCAL6408A has been used on the nRF9160DK and was therefore considered a safe choice.

The outputs from the I/O expander are used to control LEDs that are connected to VDD through a series resistor. The LED acts like a diode. Hence, when the LED is off, the voltage on the I/O expander output is around 1.2V lower than VDD, causing increased current consumption. The hardware integration guideline recommended maintaining the voltage at VDD with a pull-up resistor to minimize current consumption, which is critical for low-power applications. Pull-up resistors of 100kΩ have been included on each LED output from the I/O expander.

Passive components Passive SMD components such as resistors, capacitors and inductors have been chosen based on E. Jølsgard's prototype design, where identical resistance/capacitance values have been selected. Additional passive components surrounding the new hardware have been designed in compliance with their respective hardware integration guideline. Due to the limited space on cSLIM-v1 and extra hardware to be fitted on the PCB, all passive SMD components and footprints sizes have been reduced to 0603(Imperial)/1608(Metric), meaning 0.6 inches x 0.3 inches / 1.6mm x 0.8mm footprints. Doing so have been critical in order to fit all components on the PCB given the PCB size defined by the previous version SLIM.

Voltage regulator cSLIM-v1 requires a stable voltage of 3V to power the circuit. A switching voltage regulator have been chosen since these typically are more power-efficient than linear voltage regulators. The buck-boost switching regulator TPS63000 have been used as previous design, and can achieve up to 96% efficiency and provides current up to 1.2 A. The downside of using a switching regulator is the increased switching noise generated. Hence, this component should be placed away from noise-sensitive components.

Power multiplexer cSLIM-v1 should be able select and prioritize power from two different sources, battery and USB. Using the TPS2113PWR power multiplexer, the circuit selects power consumption from the USB when both power sources are connected simultaneously or select power from the battery when no USB is connected. The power multiplexer output is connected to input of the voltage regulator. Issues where no voltage was detected by the multiplexer, using battery as the only power source, have previously been detected. Hence, a 3-pin header (J515) have been added to the design in order to bypass the power multiplexer when deployed offshore.

Logic level converters LLCs, commonly used to connect devices with different voltage levels together, have been used in the design in order to ensure tri-state, effectively removing the input/output from the circuitry as the input/output gets a high impedance. LLCs have been placed between some peripherals and the SPI bus and ensure tri-state to components when powered down. Using LLC to power down parts of the device reduces power consumption.

LoRa Module WLR089u0, a low-power LoRa Module from Microchip have been reused from previous designs. It contains a SAM R34 family microcontroller based on Arm Cortex M0+ architecture. A separate SWD interface is included in the design as the LoRa module has to be flashed with separate software with the Atmel Ice programmer. The LoRa module has sleep currents down to 790nA, which makes it a good choice for low-power applications. An LLC have been added to the LoRa module to ensure tri-state to the pins connected to the nRF as recommended by E. Jølsgard as it was discovered that the module would draw power from the nRF9160 when powered down. On the cSLIM-shield, the LoRa module were designed with multiple communication protocols, SPI, UART and I2C, but only UART were used. Unused connections have been removed from cSLIM-v1 design.

GNSS Module The Ublox NEO-M8M was previously selected in order to provide precise timing by providing a pulse per second signal (PPS) synchronized with the GNSS clock. As described in the nRF9160 paragraph, the nRF9160 integrates a GNSS receiver but have previously not been utilized. However, PPS support was discovered when designing cSLIM-v1, making the integrated GNSS receiver a possible solution. Despite having all GNSS support needed integrated in the nRF9160, it was decided to keep the Ublox GNSS module, as a well-tested solution, in addition to adding support for the nRF9160 GNSS receiver for future testing. The Ublox GNSS receiver can be disabled and bus signals put to high impedance with the use of an LLC in order to power down the module completely.

RTC As crystal oscillators are not very precise, Jølsgard replaced the previous crystal oscillator with an ultra-low-power real-time clock with a temperature-compensated crystal. The RV-3032-C7 is supposed to drift an impressive 1.5 parts per million (PPM) between 0 and 50°C and 3PPM down to -40°C. The RTC uses only 160nA at 3V. By estimating the RTC frequency, the overall power consumption can be greatly reduced as the required period between GNSS receiver wakeups and synchronizations increases.

Storage The requirement of persistent storage have been met by the previous design and has been reused on cSLIM-v1. The design consists of FRAM chip CY15B108QN providing persistent storage while maintaining a low-power design with typically 300μA while active, 100μA standby and 3μA in sleep mode. Additionally, support for micro SD card holder was included in the design of cSLIM-shield in order to provide simple access to stored data that can be displayed on a computer. LLC have been added in order to turn the micro SD card off when not used to prevent unnecessary power consumption.

Display The display design from the cSLIM-shield have been adopted on cSLIM-v1. The LS013B7DH03 is a power-efficient display with SPI interface. The LLC from the previous design have been removed due to the display always being on and having an active-high chip select signal.

LEDs LEDs from the cSLIM-shield have been reused in the design of cSLIM-v1. These LEDs are low-power and emit light bright enough to be visible at a distance, as tested by E. Jølsgard and as required by the project requirements. LEDs will be powered by 3V. Red and yellow LEDs have a voltage drop of 1.8V, and blue and green have voltage drops of 2.65V. Hence resistors of 820Ω and 330Ω respectively have been in series with the LEDs and produce a forward current of 1.46mA and 1.06mA, respectively. Given a relatively high current consumption, these LEDs should not be powered on at all times and should rather be toggled when needed.

5.1.1 Bill of materials

A full list of components used in the design can be seen in Appendix A.2. The list includes component designator ID, the quantity of each component, value where relevant, description and the manufacturer part number.

5.2 Schematics

Existing schematics were downloaded from Jølsgard's GitHub. An add-on in the Altium Designer environment allowed design files from other ECAD tools to be imported. While this solution showed great results for PCB-layout and schematics, the add-on was not able to convert the existing component library, hence removing the link between the components' schematic symbol, footprint and 3D-model. This was solved by manually downloading all components and creating a new component library where the link between schematic models, footprints and 3D models was intact.

Figure 5.1 illustrates the different parts of a component in the component library. The schematic symbol is used in the schematics for connecting components together using their pins. Parameters can be added to each component in order to attach manufacturer part numbers or other information. The footprints represent areas of copper used for soldering, specifically designed and linked to each component. In addition, the footprint also includes guides for placing the component and a dot to indicate the orientation. The 3D-model is also linked to each component. Although it has no particular technical function, it gives valuable insights when structuring the component layout and gives a more intuitive perspective on the final PCB design.

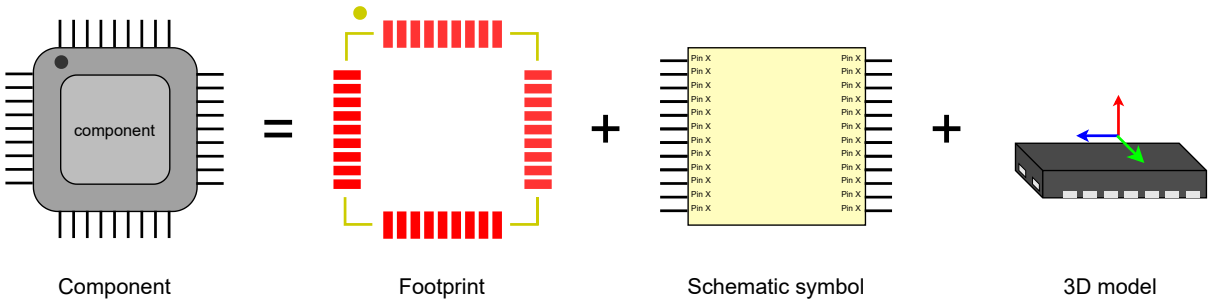


Figure 5.1: All components in the component library consists of the following parts. Parameters can also be added to each component, specifying manufacturer parts number and more. Figure: Vetle B. Abrahamsen/NTNU

Connections between two or more connection points in the schematics are made using nets and are gathered in a netlist that becomes very important when starting to develop the PCB layout. The netlist is a collection of footprint pads on the PCB that are electrically connected to one another, especially useful for the auto-router to know where to place tracks. By using the net identifier (net name) in the schematics, electrical connections can be made without having to physically draw the complete line. This is advantageous for such a complex design as the cSLIM-v1 by making the schematics cleaner and its wiring easier to digest.

In the provided schematics, Jølsgard added notes to address improvements or errors in the design. Resolving these issues and comments was the first priority before starting to add the extra schematics. The Schematics found in Appendix A.4 are divided into 12 separate sheets, where each sheet provides schematics for different hardware modules, subsystems or connectors. The top sheet lists the different sheets with its primary content. Additionally, the top sheet connects all sheets together using sheet symbols, shown as boxes representing sheets 2 to 12. Connections between sheet symbols are made using sheet entries, where nets from different sheets are specified as inputs and outputs. The top sheet also functions as an overview of the design, giving an intuitive view of the signal flow.

Since cSLIM-v1 integrates the nRF9160 SiP on board, the schematics now include one extra sheet (sheet 2) for the nRF9160 and surrounding hardware such as SIM connector, external I/O bus, and passive components. The GNSS LNA used for nRF's internal receiver have been placed on the GNSS sheet (10) in the schematics. The complete schematics can be seen in Appendix A.4.

5.2.1 Pin configuration

Table 5.1: Pin configuration, showing both signal name and physical pin on nRF9160.

Pin configuration			
Peripheral	**Interface**	**Name[pin] (input)**	**Name[pin] (output)**
nRF GNSS		nRF_GPS_LNA[GPS]	nRF_GPS_EN[COEX0], nRF_GPS_PPS[COEX1]
nRF LTE		nRF_SIM_IO[SIM_IO], LTE_ANT[ANT]	nRF_SIM_IO[SIM_IO], nRF_SIM_CLK[SIM_CLK], nRF_SIM_RST[SIM_RST]
nRF SWD	SWD	nRF_SWDIO[SWDIO], nRF_SWDCLK[SWDCLK]	nRF_SWDIO[SWDIO]
Buttons	GPIO	nRESET[NRESET], Button1[P0.03], Button2[P0.04]	
RS485 Tranciever	UART	RS485_RX[P0.00], RS485_RE[P0.14] RS485_DE[P0.15]	RS485_DI[P0.01], RS485_RE[P0.14] RS485_DE[P0.15], TBR_PPS[P0.16]
RS232 Tranciever	UART	nRF_RX[P0.23]	nRF_TX[P0.24], RS232_nShutdown[P0.11]
LoRa module	UART	UART_RX[P0.08], LoRa_EXTINT[P0.21]	UART_TX[P0.09], nRF_LoRa_nCS[P0.29], LoRa_nEnable[P0.05]
LEDs	I2C	I2C_SDA[P0.30]	I2C_SDA[P0.30], I2C_SCL[P0.31]
RTC	I2C	I2C_SDA[P0.30], nINT_RTC[P0.28]	I2C_SDA[P0.30], I2C_SCL[P0.31], RTC_Clockout[P0.26]
Display	SPI		nRF_DISPLAY_CS[P0.22], SPI_MOSI[P0.18]
GPS module	SPI	SPI_MISO[P0.17], NEO_Timepulse[P0.27]	SPI_MOSI[P0.18], SPI_CLK[P0.19], GPS_nRF_nCS[P0.10], GPS_nEnable[P0.12]
uSD	SPI	SPI_MISO[P0.17]	SPI_MOSI[P0.18], SPI_CLK[P0.19], nRF_uSD_nCS[P0.07], uSD_nEnable[P0.06]
FRAM	SPI	SPI_MISO[P0.17]	SPI_MOSI[P0.18], SPI_CLK[P0.19], nRF_FRAM_nCS[P0.25]

Table 7.3 shows the configuration of pins sorted by peripheral in the first column and with columns 3 and 4 containing the information of signal configuration (input or output respectively). Signal names are written in accordance with the schematics and their nRF9160-specific pin name. Pin names should not be confused with physical pin numbers on the nRF9160 body. Providing pin names instead of physical pin numbers have been done to aid the software development as pins are configured with their pin names and not physical numbers in the software. Some signals are bi-directional meaning that data flows in both directions, as SWDIO gives an example. Bi-directional signals can be seen in both the column for inputs and the column for outputs.

5.3 PCB layout

Before placing components, the dimensions and shape of the PCB and the board stack-up were defined with number the of layers. Material type and thickness will later be specified when placing a manufacturing order but can also be added to the board stack-up. The board stack-up, material, thickness and dimensions have been reused from the SLIM module to make a replacement as simple as possible. Given the increased number of components, the width had to be increased by 50mm in order to fit all on the top layer. Final board dimension is 130.302mm x 55.245mm. Figure 5.2 provides all the layer stack information, including the production files (Gerber) specific naming for each layer name.

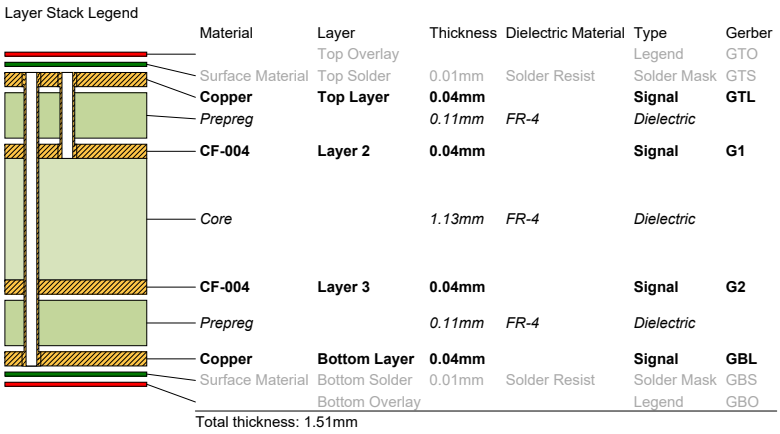


Figure 5.2: Layer stack cSLIM-v1. Total thickness of 1.51mm have due to manufacturer been adjusted to 1.6mm. Figure: Vetle B. Abrahamsen/Altium Designer

Four layers have been used, where the top layer (layer 1) is used for components and signals. Power routing have mostly been routed on the top layer as well. The PCB is designed with a pure and unbroken ground plane on layer 2, meaning that no signals will be allowed to be routed on that layer. A broken ground plane effectively increases the loop area, thus increasing the inductance Williams (1992). Hence, layer 2 is kept pure and unbroken, securing a fast path with low impedance for return current to ground. Layer 3 and bottom layer function as pure signal layers giving the design enough space to route all signals. When importing the circuit from the schematics to the PCB editor, the schematic models of components are imported as footprints, where the netlist holds information of all connections for every pad.

The main components, such as nRF9160, including the antenna, LoRa module, GNSS receiver, SIM connector and SD card were placed first. Passive components and required components afterward. Some components such as voltage regulators produce noise and were placed away from sensitive components such as the GNSS receivers, antennas and the nRF9160. The display has been placed at the bottom of the PCB with the buttons below, giving an intuitive button layout for future software/application development. Some test pads have been placed on the PCB to provide easy access measurements for signals that would have been hard/impossible to measure elsewhere.

Sections of the ground plane have been removed below the cellular antenna and below antenna pads on the nRF9160 and GNSS receiver as described in their respective hardware integration manuals to increase stability by reducing noise. The hardware integration manuals also recommended placing shielding vias around the antenna lines. Hence, via shielding, also known as via fencing, has been added in order to reduce cross-talk and electromagnetic interference on the antenna lines Altium.com (2020).

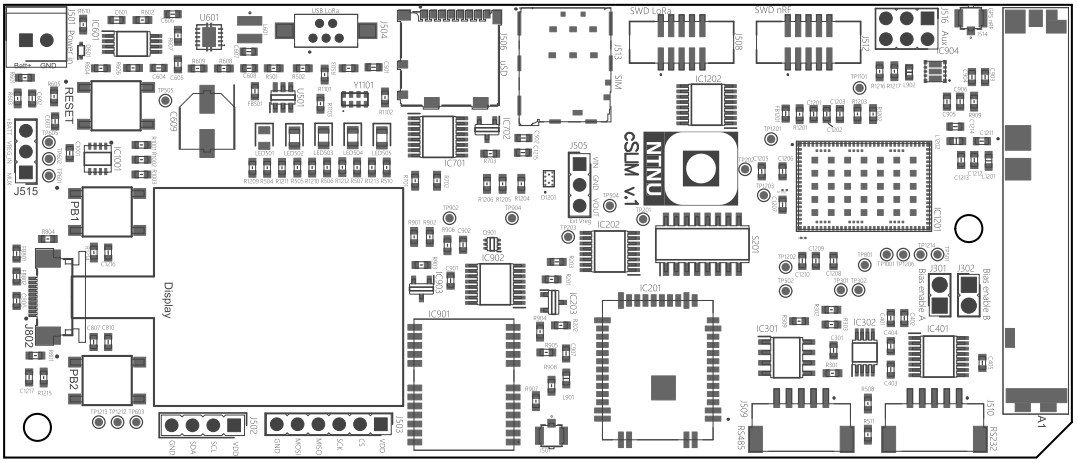


Figure 5.3: Board assembly including designators and placement guides. Figure: Vetle B. Abrahamsen/Altium Designer

5.3.1 Design rules

Using advanced path algorithms, the inbuilt autorouter in Altium Designer can be used to realize all the connections from the netlist automatically. However, the autorouter must be forced to follow a specific set of prioritized design rules. Design rules that have been specified are mostly hardware design guidelines provided by the component manufacturers i.e., spacing between differential pairs for USB signals, minimum, maximum and preferred track width. Additionally, specifications given by the PCB manufacturer such as minimum drilling size, via-size, minimum possible track width and spacing between tracks have been (and must be) specified in the design rules in order to guarantee a design that is physically manufacturable given the limitations of production machines.

Component-specific rules have been achieved by giving a component(s) a design rule with higher priority than a more general rule that applies to all components. The same method can be used to specify rules for certain nets and even specific pads on a component. For example, the via shielding for the internal and external GNSS and cellular antenna nets have been configured by specifically targeting these three nets in the design rule with a high priority. A PDF showing a complete list of all design rules have been generated and can be found in "DesignFiles\Altium project\cSLIM-V1" and is named "cSLIM_V1_rules.pdf".(For readers with access to the design files.) However, when opening the PCB project in Altium Designer, all used design rules will be included and enabled.

5.3.2 Routing

The auto-router managed to connect 99.87% of a total of 580 connections, leaving only one connection unrouted. Routed connections from the auto-router have been used as a starting point for fine-tuning and modifications to achieve an aesthetically satisfying final result. Modifications consist of connecting the remaining connection, placing vias on straight lines and trying to keep tracks as grouped as possible. Figure 5.4 shows each routed layer separately.

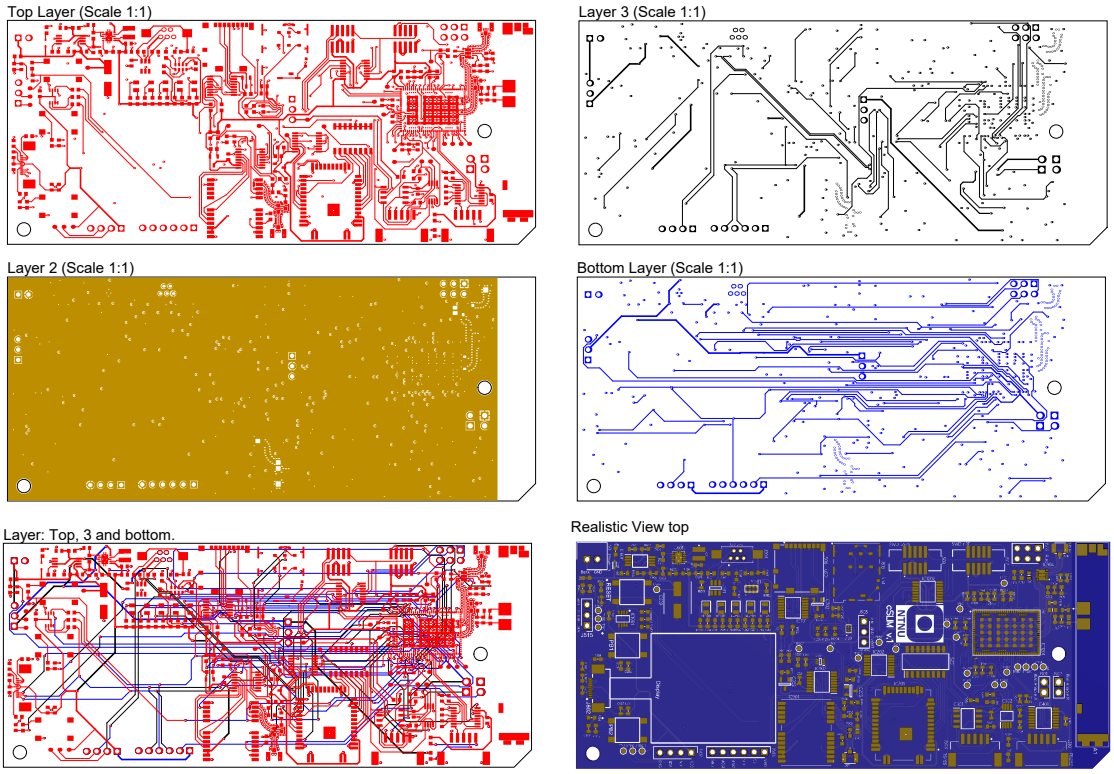


Figure 5.4: Routed connections on each layer and a realistic view of the top layer with solder pads visible. Figure: Vetle B. Abrahamsen/Altium Designer

5.4 PCB manufacturing

Two types of files are required for production, Gerber files and NC drill files. Gerber files are responsible for showing all tracks, pads, overlays and silkscreens, while the NC drill file contains coordinates and dimensions of every via or hole to be drilled. Production files were auto-generated by setting up an output job file and selecting both Gerber and NC-drill files. A manufacturer in China named PCBWay was chosen to produce the final PCB due to low prices, effective production and previous experience with the company. Production files were uploaded and specifications such as PCB dimensions, material type and surface finish were specified. The following table in Figure 5.5 shows the specifications selected when placing the order.

Parameter Information :

PCB Type :	Through hole board	BoardSpec :	IPC 6012 Class 2
Board type :	Single pieces	Panel Way :	
Different Design in Panel :	1	X-out Allowance in Panel :	
Size :	130.302 x 55.245 mm	Quantity :	10
Layer :	4 Layers	Material :	Tg150 FR-4:
Thickness :	1.6 mm	Min Track/Spacing :	3/3mil
Min Hole Size :	0.15mm ↑	Solder Mask :	Blue
Silkscreen :	White	Edge connector :	No
Surface Finish :	Immersion gold(ENIG) (2U")	"HASL" to "ENIG"	No
Via Process :		Finished Copper :	1 oz Cu (Inner Copper:1 oz)
Extra pcb product number :		Additional Options :	UL Marking:None,
PO No. :		Manufacturing :	
Final Inspection Report(free)	Default Inspection Report		

Figure 5.5: Screenshot of PCB order specifications in PCBway.com. Figure: Vetle B. Abrahamsen/PCBway.com

5.4.1 Considerations

ENIG were chosen as surface finish from the producer, it has a flat surface finish in contrast to thin-lead HASL that has the inherent issue of being non-planar or "bumpy" Lentz (2018). Bumpy pads can affect how the solder paste is distributed and might result in a gap between the pad and the stencil when applying solder paste. A paper by FCT Assembly have tested a wide variety of surface finishes in combination with types of solder paste.

The paper concludes that ENIG has the highest score of all tested surface finishes Lentz (2018) and was thus selected as the surface finish for cSLIM-v1. The same paper also gives the NC 63-37, a solder paste with a blend of 63% tin and 37% lead, the best overall score, which is why NC 63-37 solder paste was chosen. The ENIG and NC 63-37 combination scored 37 out of 45 points, 3 points below the combination with the highest score - OSP surface finish combined with NC 63-37. However, OSP was not recommended for reflow Lentz (2018), which is how the cSLIM-v1 will be soldered. The most critical component, nRF9160 SiP, did not have any recommendations regarding solder paste, but an employee from Nordic Semiconductor suggested on devZone.nordicsemi.no to use T5 when soldering the nRF9160 SiP, where T5 specifies the size of tin particles in the solder paste, where a lower number, means smaller tin particle size. T5 (T1-T8) is on the smaller size and a good fit for small SMD components.

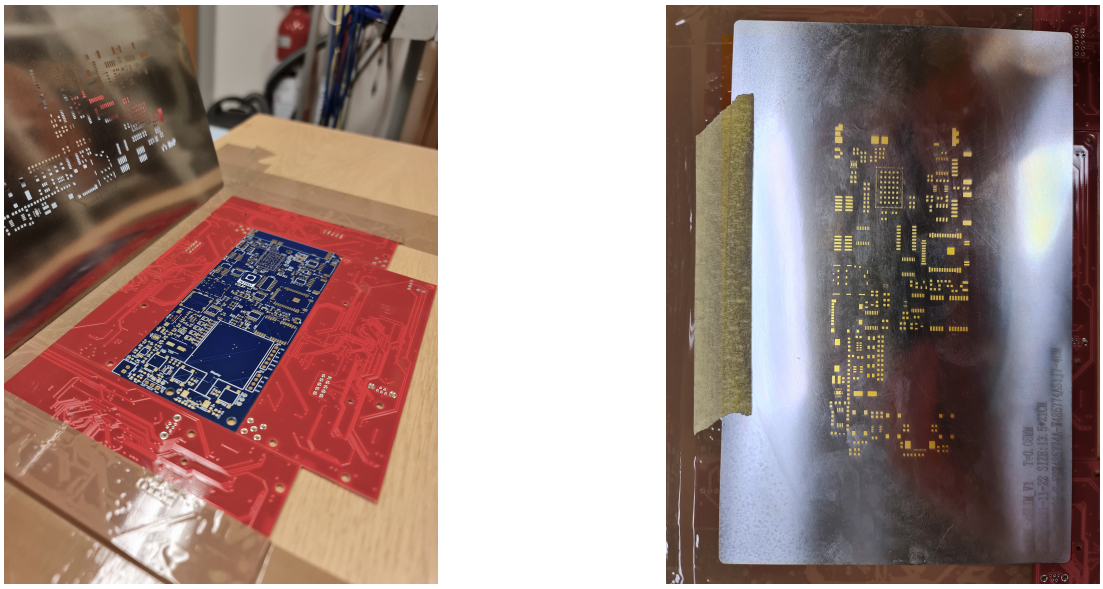
5.5 Assembly of multiple cSLIM-v1 for field tests

The assembly process is the most critical part of the project. A fault during soldering could, in worst case, result in a useless PCB. Having several components with tiny footprints significantly increased the risk of soldering complications. Hence, extreme care was taken to avoid any issues. Three fully cSLIM-v1 have been assembled during this thesis. One for field tests and additional cSLIM-v1's for further development.

5.5.1 Solder paste

For some components, specifically the nRF9160-SiP, where solder-balling, bridging and shorts are of high concern, it is crucial that the solder paste is applied evenly with the right amount of solder paste. During the project report, extensive tests were conducted in order to find the optimal stencil thickness and technique for applying solder paste. The project report concluded that a stencil thickness of 0.8mm gave the most suitable amount of

solder paste and that the solder paste should be applied in one drag using a flexible tool that covers the whole width of the PCB. This combination reduced the possibility of solder-balling and bridging greatly.

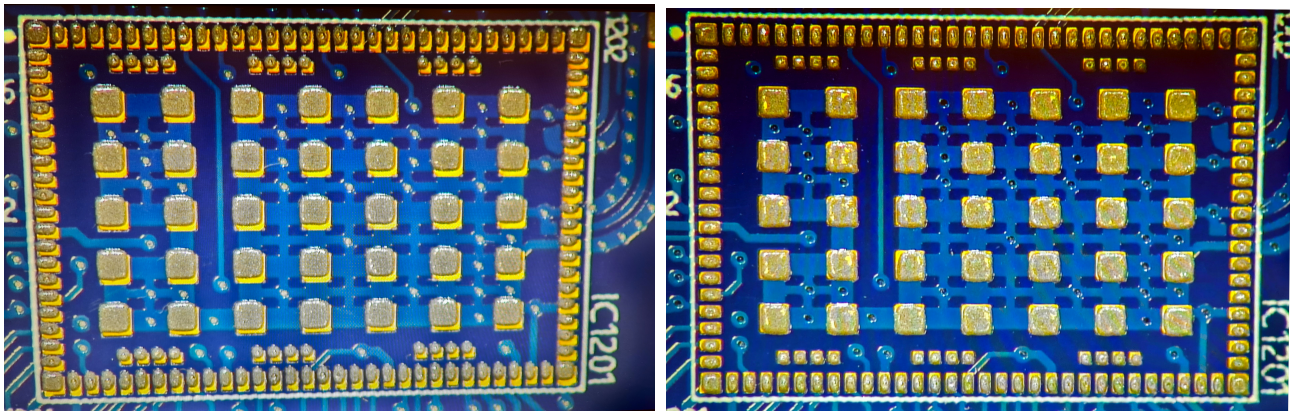


(a) Stencil frame with inserted PCB

(b) Aligned stencil: 0.8mm

Figure 5.6: Figure (a) shows the stencil frame with an inserted PCB ready for applying solder paste. Figure (b) illustrates how to PCB solder pads are visible through the stencil, indicating good alignment. Photo: Vetle B. Abrahamsen/NTNU

The stencil was taped to a frame as shown in Figure 5.6 after aligning the stencil holes with the solder pads on the PCB. A good tip is to use a tape without much flexibility since even the slightest movement causes stencil alignment error as shown in Figure 5.7a. The frame surrounding the PCB should be built by objects with the exact same height in order for the stencil to lay completely flat on the surface. A good stencil setup makes it possible to insert new PCB's without having to re-adjust the stencil alignment and will be more effective when assembling multiple circuit boards. Figure 5.7b shows an acceptable result that is ready for component placement,



(a) Not aligned stencil

(b) Aligned stencil

Figure 5.7: Pictures shows (a) applied solder paste with stencil slightly out of alignment - must be re-applied. Figure (b) shows a perfectly applied solder paste ready for component placement. Photo: Vetle B. Abrahamsen/NTNU

5.5.2 Soldering

Components with pads below their component body, passive components and parts that was too difficult to solder manually was soldered using a reflow oven. Through-hole components and parts that could be solder manually was later mounted and soldered with soldering iron. The process of component placement were intentionally slow in order to place the component as precise as possible on the PCB to prevent disturbing the solder paste too much. Components were placed using a digital microscope and tweezers, aided by the component outline that is printed on the PCB as shown in Figure 5.7.

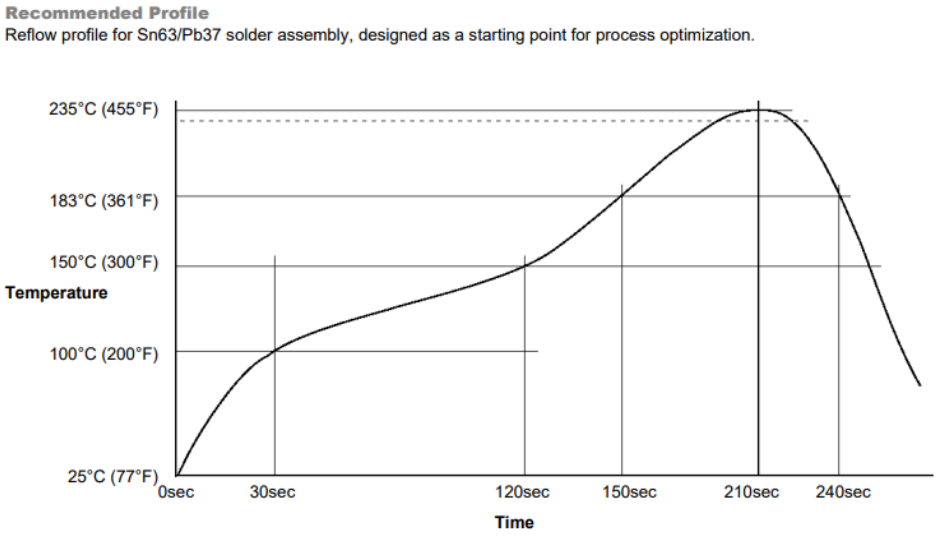


Figure 5.8: Recommended reflow profile for SMD291AXT5 by CHIPQUIK.

Reflow profile SMD291AXT5 from CHIPQUIK have been used as solder paste, hence their recommended reflow profile was configured on the reflow oven. Figure 5.8 shows the recommended profile for this specific solder paste in order to achieve optimal reflow. The reflow oven was configured, as specified in Figure 5.8, with a preheat temperature reaching 150°C over a period of 90 seconds, ramping up the heat over a period of 90 seconds to a maximum of 235°C. At this temperature the solder paste should achieve reflow before quickly decreasing the temperature. Extending the period from preheat to reflow by 5 seconds had to be done in order reach the specified reflow temperature of 235°C as the reflow oven could not reach the specified temperature at the specified period. Using the wrong profile or not reaching the reflow temperature could potentially prevent the solder paste from reaching reflow, leaving components unsoldered. Hence, the profile should be tested before soldering the actual PCB.

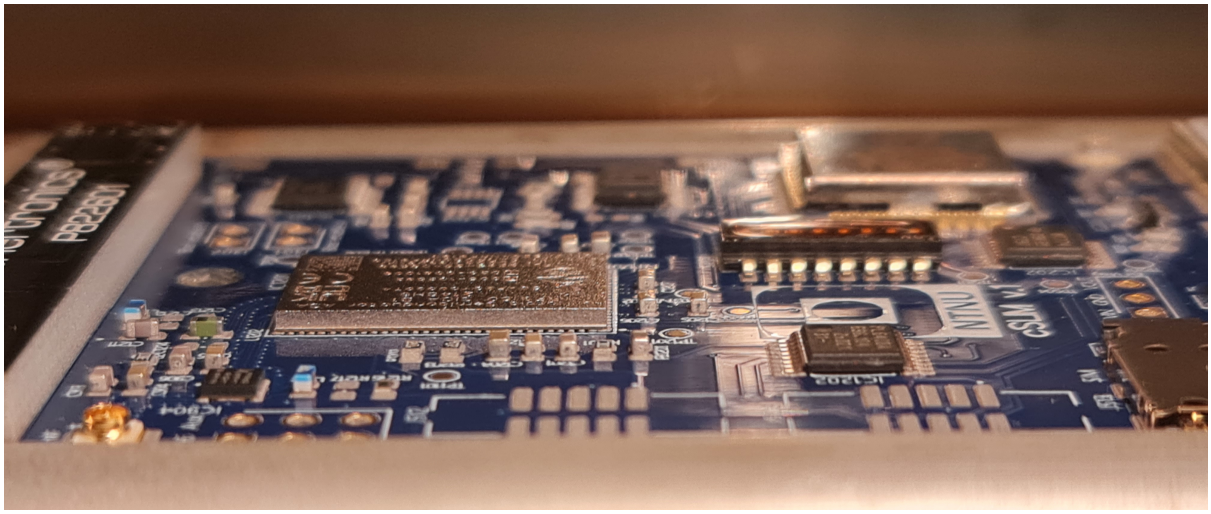


Figure 5.9: Partially assembled cSLIM-v1 in reflow oven. Photo: Vetle B. Abrahamsen/NTNU

5.6 Verification and post-design modifications

Inspection of the assembled PCB under a the electronic microscope showed no sign of solder-balling, bridging or bad connections for any of the assembled PCBs.

The circuits were tested for shorts from power sources to ground using the test pads on the PCB. Where possible, the trace resistance between the end and start points of tracks were measured using a multimeter to verify good connection. No faults detected. After applying power to the PCBs, with no software running, the voltage levels were measured to ensure functionality of the voltage regulator, power multiplexer and a safe voltage level for the components. 3V was measured implying a successful integration of the power multiplexer and voltage regulator. The reset circuit, being active high, was drawn low as the reset button was pressed, but after closer inspection with a oscilloscope a design error was detected and required modifications on the reset line, see Section 5.6.1.

5.6.1 Reset line modification

When measuring the voltage on the reset line with oscilloscope, the voltage levels did not match up with the required nRF9160 reset line specifications. The nRF9160 SiP uses an internal pull-up resistor to 2.2V. However, as shown in Figure 5.10 voltage levels of 2.7V were measured, additionally steps in voltage during transition were detected, implying a design fault. The root cause of the the reset fault was due to the shared reset line between components, specifically the LoRa module and GNSS. Both modules have a pull-up resistor on their reset pins to VDD, meaning 3V.

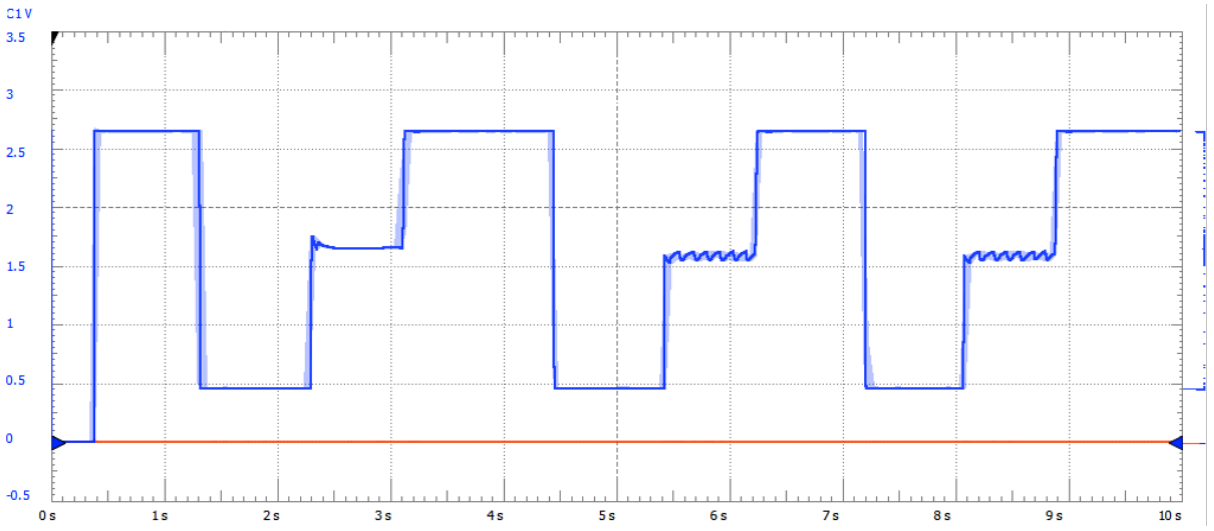


Figure 5.10: Voltage measurement on the reset line with error. Reset button pressed 3 times (falling edge). Figure: Vetle B. Abrahamsen/NTNU 2022

Both the GNSS module and LoRa module had to be removed from the reset line by cutting the reset-trace on the PCB, disconnecting them completely from the reset line. However, the reset line is not critical to these modules as a power-up cycle has the same function as a reset and will occur whenever the nRF9160 initializes after a reset due to the disabling/enabling of the transistors used for powering the modules. The IO expander - also connected to the reset line do not use pull-up on its reset-pin and do not require any modification. Due to the PCB layout and routing, when cutting the reset-line, the nRF9160 SiP will also be disconnected and requires a physical wire to bridge the connection from the trace between R606 and C608 to the reset-pin on the LoRa SWD header J508. Figure 5.12 shows the wire in the bottom left corner. See Appendix A.5 for full trace-cutting and bridging details. The design error have been modified in the PCB layout, and will be implemented in the new Gerber files. Figure 5.11 shows the transient response when pressing the reset button after the implemented modification - giving the expected 2.2V and no steps in voltage.

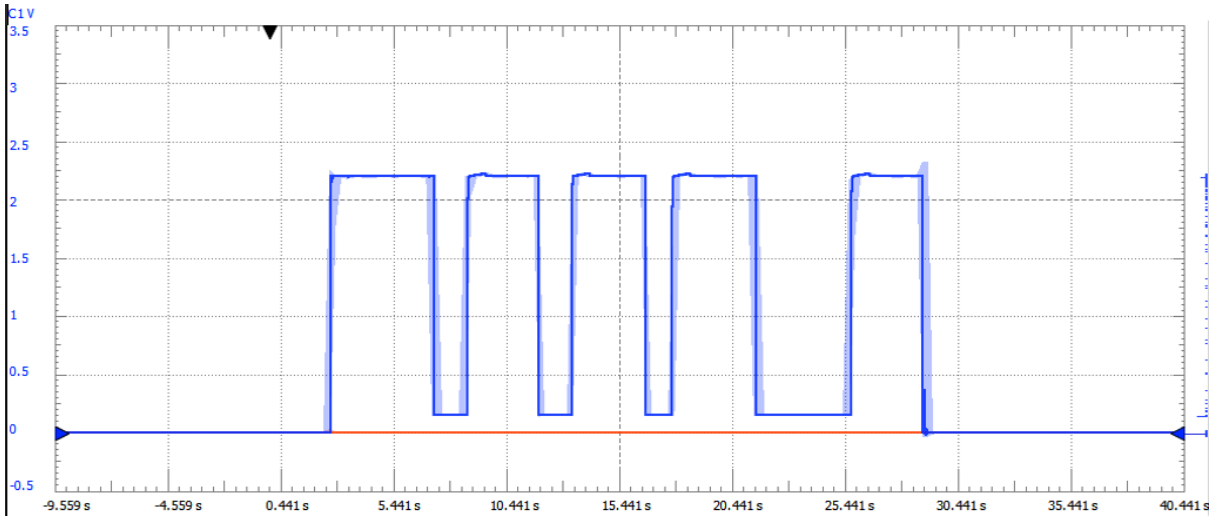


Figure 5.11: Voltage measurement on the reset line after modification. Reset button is pressed 5 times (falling edge). Figure: Vetle B. Abrahamsen/NTNU 2022

5.6.2 Verification using test software

As the verification of the assembly and modifications was successful, with no faults detected, the PCB was ready to be flashed with cSLIM software in order to further test and verify the functionality. The fully assembled cSLIM-v1 can be seen in Figure 5.12. Two out of three cSLIM-v1 assemblies were successful, with the last one not being able to get SPI mutex. All cSLIM-v1's were running the exact same software, giving reason to believe the fault is hardware-related, either soldering or damaged hardware. No further action have been taken in investigating the root cause as development of the software had to be prioritized for the upcoming field-test.

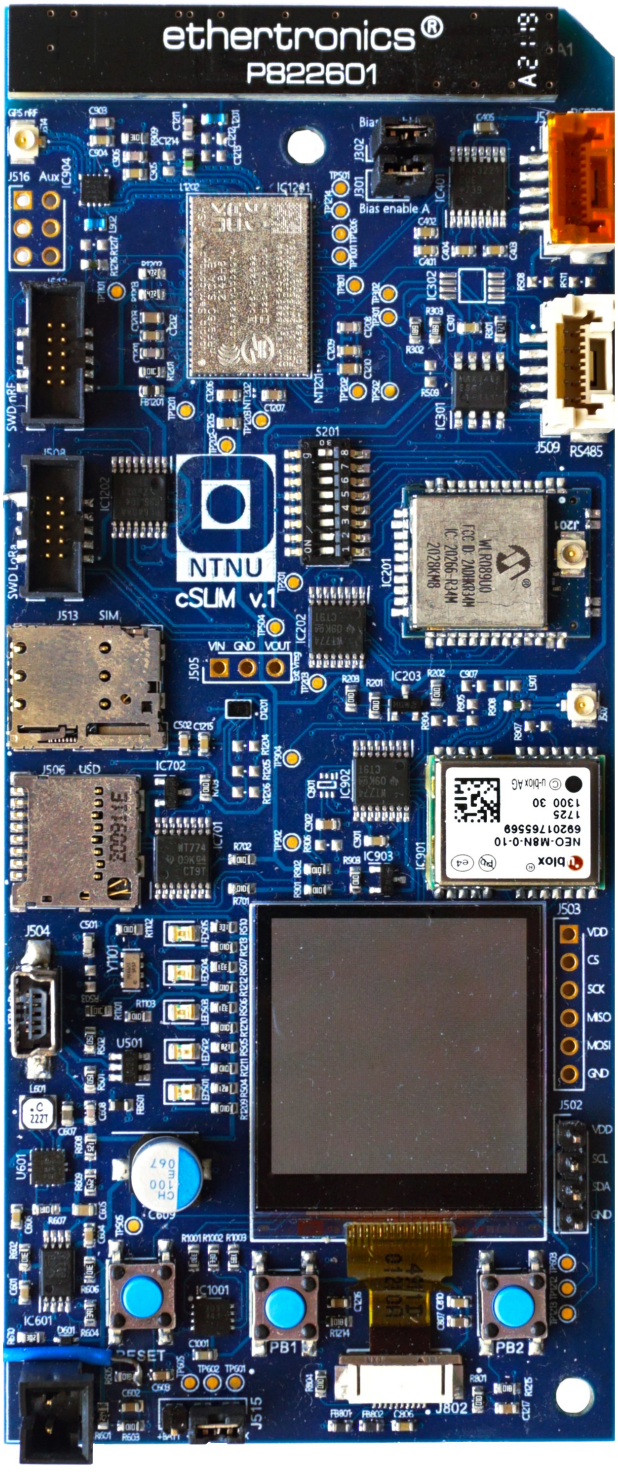


Figure 5.12: Fully assembled PCB. Photo: Vetle B. Abrahamsen/NTNU 2022

Chapter 6

cSLIM-v1: Power Solution

During this thesis a large amount of time have been spent on power debugging. The information gathered and observed are crucial for further development, hence a separate chapter have been introduced.

The battery originally planned to be used with the buoy controller was a Thionyl Chloride Lithium battery with a nominal voltage of 3.6V, nominal current of 10mA and a capacity of 35Ah seen in Figure 6.4. The battery specifies a maximum continuous discharge current of 450mA and handles pulse current up to 1A Batteries (n.d.).

6.1 Battery issue

When powering cSLIM-v1 using the battery, the system would repeatedly reset, with a constant period between resets. On closer inspection using oscilloscope the reset was caused by a brown out, meaning that the voltage level one a specific nRF9160 pin was measured below the brownout reset threshold of 1.6V. As seen in Figure 6.1 both the battery voltage and VDD experienced a voltage drop, with VDD being below the Brown-out threshold - causing the nRF9160 to perform a brown-out reset of the chip.

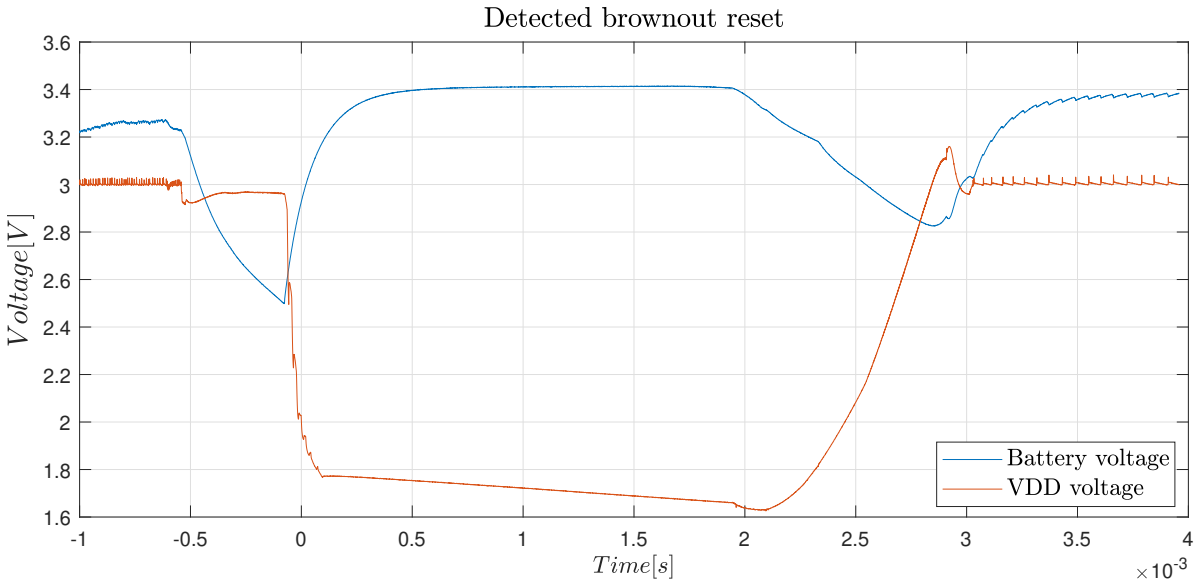


Figure 6.1: VDD and battery voltage during brown out reset. Figure: Vetle B. Abrahamsen/NTNU 2022

Analysing Figure 6.1 gives some information about what is happening. Around time 0s the battery voltage drops to 2.5V which is the lowest voltage the battery will have before cut-off as stated in the manufacturers technical

brochure Tadiran (n.d.). At this point the battery are not able to deliver on current demand and results in the VDD voltage drop seen in Figure 6.1.

The cSLIM-Shield prototype application software was used during this problem and tried to poll GNSS data and connect to the cellular network simultaneously. Both these operations requires a lot of power, above the 10mA nominal current specified in the battery and with demanding current spikes over a long period driving the battery voltage down.

6.2 Adding battery aiding circuit: Supercapacitor

Tadiran's technical brochure for their Lithium batteries suggests to use a supercapacitor in parallel with the battery to aid the battery during demanding current pulses/spikes Tadiran (n.d.). A "supercap" is a capacitor with high capacity and extremely low internal resistance, making it able to deliver high currents instantly when needed. Whenever the voltage level on the battery drops, the capacitor discharges its energy to the circuit and reduces the voltage drop. Figure 6.4 shows the supercapacitor used to aid the battery and Figure 6.2 shows the voltage measurement on cSLIM-v1 running its full application with and without the supercapacitor.

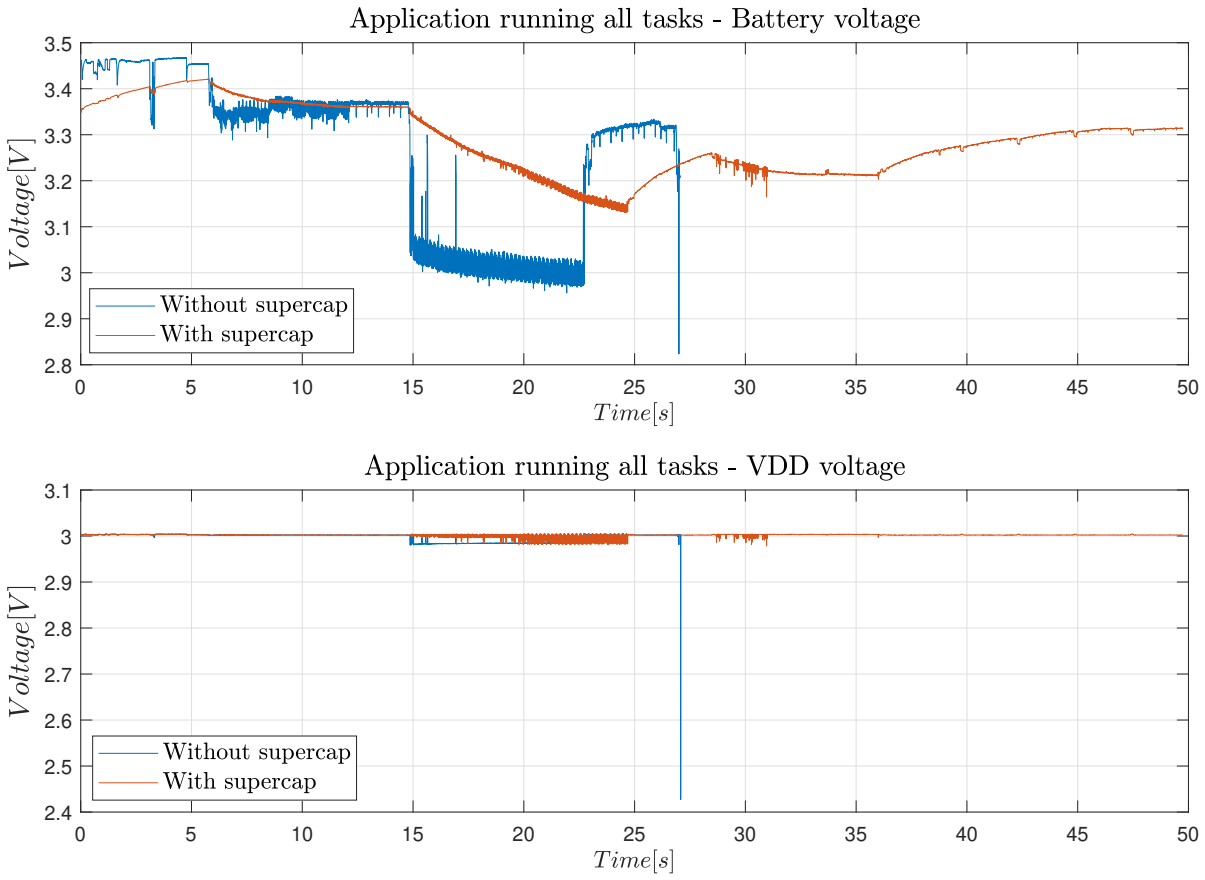


Figure 6.2: VDD and battery voltage, with and without super capacitor. Due to large time-window, the true VDD voltage without supercap at T=27 is not shown. However the voltage drop causing a brown out is clearly visible. Figure: Vetle B. Abrahamsen/NTNU 2022

Without the supercapacitor the battery struggles to retain a stable voltage and fails at T = 27s, resulting in an brown out reset. At time T = 15s the application starts the GNSS task while still trying to join the cellular network, causing extra high current demand and higher voltage drop. The voltage is very oscillating due to high current pulses. When the battery is aided by the supercap, the oscillations are reduced on the battery voltage due

to the supercap being able to provide the high current pulses. As the capacitor discharges the voltage continues to drop, but manages to both connect to the network and search for GNSS satellites simultaneously without resetting.

6.3 Sequential initialization for limiting current

Based on Figure 6.2 it is clear that the startup of the cSLIM-v1 have to be coordinated in order to limit the amount of current demand put on the battery. A sequential start-up of the cellular modem (MQTT task) and GNSS task have been implemented, meaning that the GNSS task will wait until the cSLIM-v1 connects to the network before waking the GNSS for initialization. Figure 6.3 show a comparison of the different power solutions.

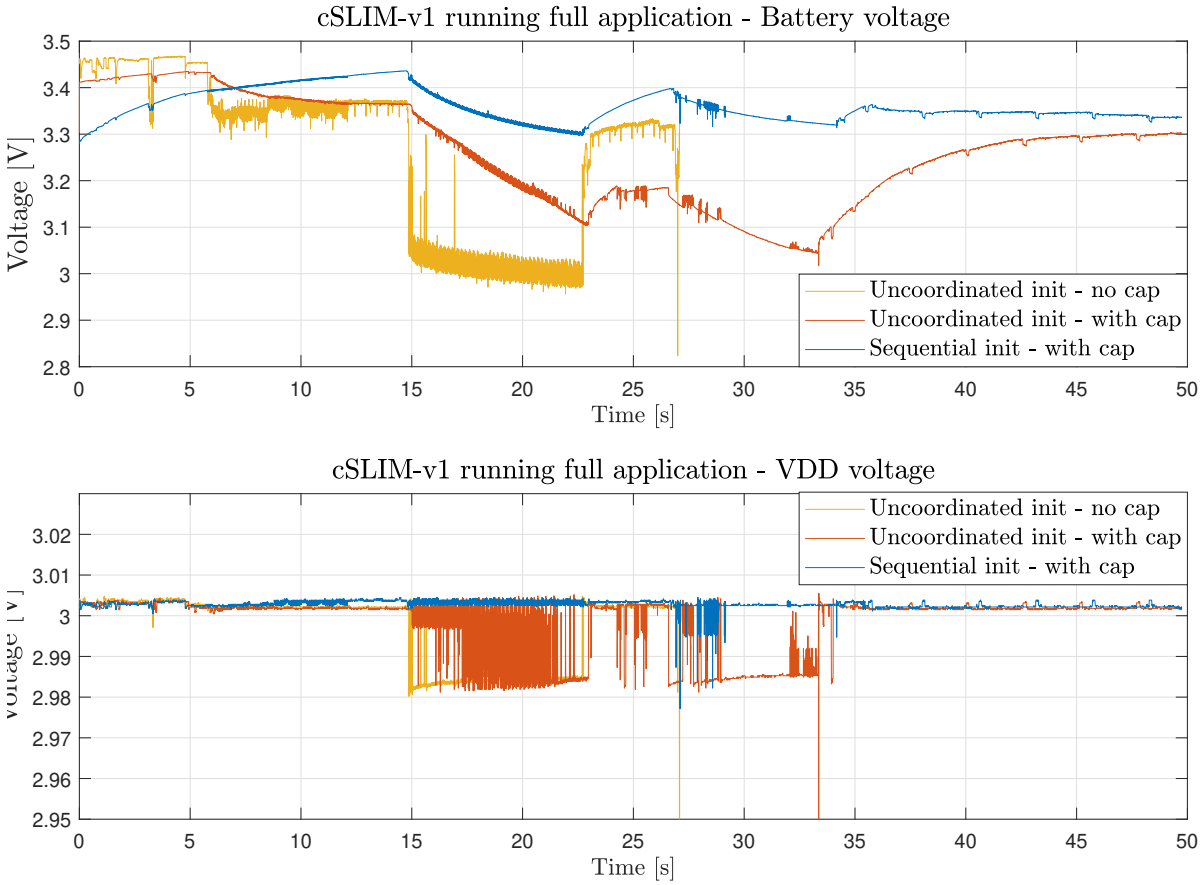


Figure 6.3: VDD and battery voltage comparison. Only battery, battery with supercap and no coordination and final solution with super cap and sequential initialization. Figure: Vetle B. Abrahamsen/NTNU 2022

Without any aid from super capacitor, the battery are not able to reliably power cSLIM-v1. Adding the supercapacitor improves the solution, but low voltage levels are still concerning and there can be still be seen a voltage drop in VDD when the current load is too demanding. The sequential startup combined with supercapacitor significantly reduces the stress on the battery and are successfully powering cSLIM-v1. However, tests were done with good network and GNSS reception. Longer periods with network connection problems might cause battery problems. The battery solution can be seen in figure 6.4.

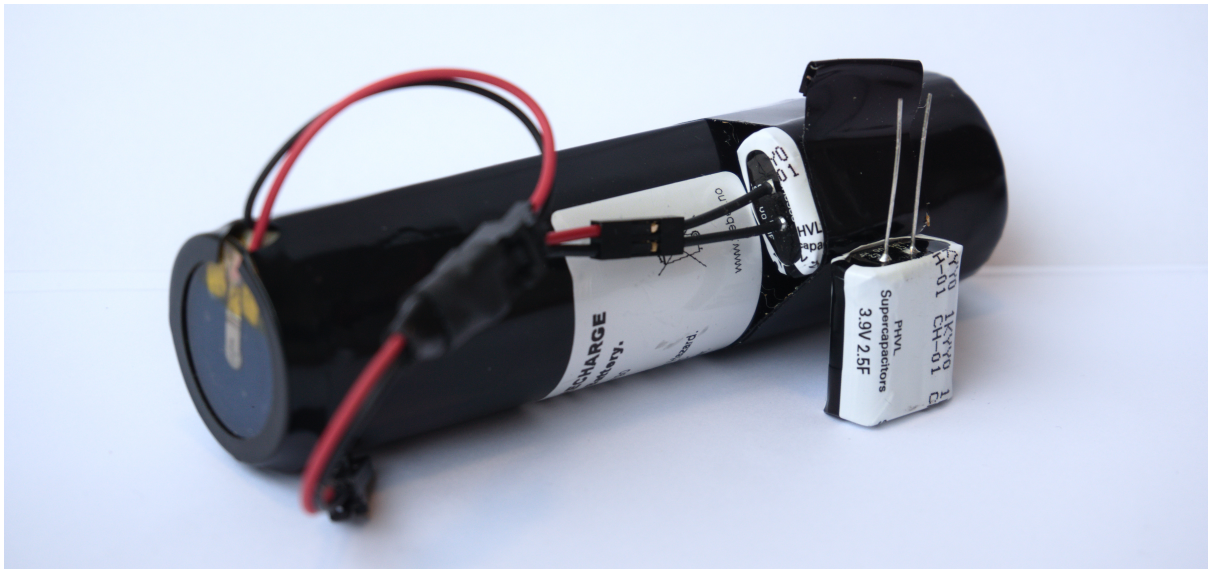


Figure 6.4: Tadrian SL-2790, 3.6V Thiomyl Chloride Lithium battery solution with 3.9V, 2.5F supercapacitor. Photo: Vetle B. Abrahamsen/NTNU 2022

Chapter 7

cSLIM-v1: Software development and optimization

Software was developed by Eivind Jølsgard during his work on the prototype cSLIM-Shield. E. Jølsgard adapted the previous SLIM software with modifications in order to seamlessly implement Zephyr support, a real-time operating system responsible for scheduling tasks, detecting fatal errors and recovering the system.

The main goals of this thesis have been to realize the stand-alone cSLIM-v1 and be able to test the buoy controller in a real-life scenario during the migration of fish in Stryn. In order to deploy a test buoy, it is essential that all communication protocols and functionality work as expected, that messages are reliably transmitted and that the buoy controller is able to detect and handle unexpected behavior not handled by the Zephyr RTOS. This chapter will focus on providing an overview of the software solution, the development and changes in the cSLIM software.

7.1 cSLIM-v1 application

The existing software uses Zephyr RTOS, consisting of application service, OS service, kernel and hardware-specific software that controls the LTE modem, Cortex-M33 processor, ram, flash and peripherals in the nRF9160-SiP. On top of Zephyr, E. Jølsgard provided application software divided into drivers, devices and tasks. Although Zephyr includes drivers for all peripherals available on the nRF9160-SiP, additional drivers were developed by E. Jølsgard in order to adapt existing software from the SLIM version. Using Zephyr enables the tasks to be scheduled based on priority by the real-time operating system.

7.1.1 nRF Connect SDK

Using the nRF Connect SDK, both Zephyr RTOS and nRF-drivers are built with the solution by including it in the source code. The cSLIM software uses nRF Connect SDK v1.5.1 and have also been used to implement software on the cSLIM-v1. Upgrading to the nRF Connect SDK v1.9.1 would be preferable but will require a lot of re-writing as some functions used in the cSLIM application might not be supported or have been replaced in the newer version. As this thesis sets out to realize the prototype, it was decided to keep the current SDK version in order to not introduce software issues prior to having proved functionality of the current software.

7.1.2 Folder: Boards

The boards folder contains the configuration of the targeted board, in this case, cSLIM-v1. The board configurations include the device tree file, responsible for configuring the pin-out from the nRF9160 SiP, and Kconfig file

giving configuration options without the need for changing any source code. Multiple boards can be included but only one is chosen when building the solution. The board files included in this folder are based on the nRF9160-DK as used in the prototype and contain both non-secure (used for cSLIM application) and secure configuration files for nRF9160-SiP.

7.1.2.1 Device tree

Since cSLIM-v1 have a slightly different pin-out than the prototype, the pin-out configuration have been modified in the cSLIM_common.dts file located under the boards/arm/cSLIM folder. The device tree have been modified to match the new design by changing the pin parameters. An example showing the device tree configuration of the LoRa module pins for enabling a transistor to power the device and a signal for enabling LLC output is shown in listing 7.1.

```
wlr089u0_gpio {
  compatible = "gpio-leds";
    wlr089u0nen: wlr089u0_nen {
      gpios = <&gpio0 5 0>;
      label = "WLR89U0 nSHDN";
    };
    wlr089u0llcnen: wlr089u0llc_nen {
      gpios = <&gpio0 29 0>;
      label = "WLR89U0 LLCnEN";
    };
};
```

Listing 7.1: Zephyr device tree pin configuration for controlling power and LLC to the LoRa module. Source: cSLIM\boards\arm\cSLIM\cSLIM_common.dts

7.1.3 Folder: Drivers

The cSLIM-v1 primarily uses the exact same hardware as tested on the cSLIM-shield. The only change is an added I/O expander that uses I2C to communicate with nRF9160. The I/O expander controls all LEDs on the PCB, and a LED driver have been written from scratch. As the thesis sets out to realize the prototype as a stand-alone buoy controller, drivers are supposed to be adopted as is. However, some drivers required editing and have been modified accordingly to ensure functionality. Drivers, especially the RS485, have been modified with extra fault tolerance in order to prevent loss of tag detections.

Table 7.1: cSLIM drivers. Source: cSLIM_v1\src\drivers

Drivers	
Driver	**Functionality**
cSLIM button	Currently not implemented.
cSLIM LED	Responsible for initializing the GPIO expander and drive LEDs.
cSLIM i2c	Initializes and manages the I2C bus and avoiding collision by managing an I2C mutex. contains functions for reading and writing data from/to slaves. used by RTC and the I/O expander
cSLIM SPI	Initializes and manages the SPI bus and avoiding collision by managing the chip select signals with an SPI mutex. Contains functions for reading and writing data. Used for GNSS, Display, uSD and FRAM.
cSLIM RS232	Initializes the RS232 zephyr device and manages the RS232 interrupt routine. Responsible for receiving and transmitting data. Currently not used as no UART peripheral are available.
cSLIM RS485	Initializes the RS485 zephyr device using uart1. Manages the UART1 interrupt routine that parses the incoming TBR data or transmits commands to the acoustic receiver. Used by the acoustic receiver (tbr device).

7.1.4 Folder: Devices

Higher level device drivers have been developed. The folder contains drivers specific to one device such as the acoustic receiver for supporting their specific serial interface, or a specific driver that supports the ublox GNSS protocol to give some examples. Table 7.2 lists the different higher level device drivers used in the cSLIM application.

Table 7.2: cSLIM devices. Source: cSLIM_v1\src\devices

Device drivers	
Device	**Functionality**
cSLIM analog	Contains drivers for initializing the analog to digital converter (adc) and reading battery level.
cSLIM fram	Initializes the fram, functions for writing data to address spaces or reading data from specified data addresses in addition to functions for enabling or disabling the fram chip.
cSLIM sd	Initializes the uSD driver. Currently not used.
Display	Initializes the display and contains functions for drawing pixels and toggeling the display at a given interval.
tbr	Contains functions for Thelma BioTel's serial interface including commands, calculation of Luhn's error check digit and receiving acknowledge messages after a sent command.
ublox gps	Contains initialization of ublox, functions for managing the device state and parsing incoming ublox messages.
pcal6408	Contains initialization of the external I/O expander and functions for controlling the internal registers for controlling LEDs.
rv-3032-c7	Contains initialization of the RTC and all functionality for managing its operation and setting time or frequency.

7.1.5 Folder: Messages

Messages contain two scrips, cSLIM_messages or tbr_messages. cSLIM messages are responsible for placing the received data in the correct header or payload frames. cSLIM messages manage the buoy status message frame and GPS cycle frame. During this thesis, both scripts have been thoroughly tested and have been fixed and verified using Node-RED to display incoming IoF messages.

tbr_message contains, first of all, a function that sets the transmitter protocol used by the acoustic tag as an identifier integer between 0 to 7. The detection frequency is then combined with the transmitter protocol and forms the complete "code type" data that is transmitted as one value to the application layer. The algorithm for combining these values is owned by Thelma Biotel and will not be explained any further in this thesis. This script is also responsible for generating the tbr header frames as well as the tbr log and tbr detection frames. As different transmitter protocols contain different data, the tbr tag detection message will be formatted uniquely based on their acoustic protocol using a switch case.

7.1.6 Folder: MQTT, time

cSLIM_MQTT can be found in the "mqtt" subfolder and initializes the MQTT client as configured in the configuration file prj.conf. Here the MQTT can be initialized with a client that connects to the MQTT broker using TLS for secure data transmissions. Functions for sending AT commands to the nRF9160 modem are implemented with some predefined commands, including the AT command for loading the TLS certificate to the nRF modem, as Nordic's AT interface through nRF SDK didn't work with this custom circuit. All cSLIM-v1 have been loaded with the certificate and it is not necessary re-load the CA certificate unless the certificate is changed.

Within the time subfolder, the local_time script can be found and implements functions for receiving the local time, setting the local time or to drive the local time whenever a new clock pulse arrives from the RTC. A time_conversion script is located within this sub-folder and contains functions for converting between different time formats.

7.1.7 Folder: Tasks

The tasks folder contains all application tasks and are spawned as independent threads with specified priority and are scheduled by the Zephyr RTOS. Each task is responsible for either specific hardware, communication or monitoring. Tasks can communicate with each other by passing messages through buffers. Table 7.3 shows all tasks and their respective functionality.

Table 7.3: cSLIM tasks. Source: cSLIM_v1\src\tasks

Tasks	
Task	**Functionality**
cSLIM_status_task	Gathering information such as battery level, air-temperature and timestamp and forwards it to MQTT and LoRa tasks depending on the LPWAN technology used. When new navigation data is available or when a pre-defined timer runs out a status message will be sent.
MQTT_task	Enables the nRF9160 modem when using NB-IoT. The tasks connects to the cellular network and establishes connection with the MQTT broker and publishes messages to the MQTT topic. The MQTT task is also responsible for reconnecting to the network in case of disconnection.
LoRa_task	Configures the LoRa module, establishes connection with the LoRaWAN gateway and forwards messages to the LoRaWAN gateway.
GPS_task	Configures the Ublox GNSS module and managing waking up/sleep, polling navigation data or UTC time data. This task passes messages to the time task and cSLIM status task.
Time_task	Synchronizes the local time on nRF9160 with UTC time and computes an estimate of the RTC frequency in order to determine the next wake up of the GNSS module. The task will calculate time drift and detect if the drift is above the limit. The time task also forwards GPS cycle messages to the MQTT task.
TBR_sync_task	Synchronizes the acoustic receiver time and store the synchronization offset.
TBR_detect_task	Receives messages from the acoustic receiver, decodes the received message, adds synchronization offset and passes message to the LoRa and MQTT tasks.
Display_task	Initializes the display and ugui library. The tasks updates the display with messages passed to the display buffer on request. Continuously toggles the display.

7.2 Modifications

Most software have intentionally been left as is due to hardware realization. However, some modifications to optimize existing functionality and resolve bugs have been implemented. The following subsections will cover the most essential modifications.

7.2.1 Frequency estimation filter

The time task estimates the RTC frequency in order to reduce the drift and be able to extend the GPS sleep cycle, thus increasing the battery life. The frequency estimator given in Equation 7.1 is a low-pass filter implemented by E. Jølsgard in his thesis. The low-pass filter tuning parameter α determines the weight of the previously estimated frequency, while $(1 - \alpha)$ determines the weight of the most recently calculated frequency.

$$f_{estim[k+1]} = \alpha * f_{estim[k]} + (1 - \alpha) * f_{estim[k]} * f_{SinceLastUpdate} \tag{7.1}$$

$$f_{SinceLastUpdate} = \frac{elapsed\ seconds * \mu s\_per\_sec}{elapsed\ seconds * \mu s\_per\_sec - \mu s\ local\ at\ GPS\ correction} \tag{7.2}$$

The frequency since last update, seen in Equation 7.2 calculates an unitless frequency scaling factor. The frequency scaling factor multiplied by the previously estimated frequency constitutes the average frequency during the last GPS cycle. This means that if no drift is measured, the frequency since the last update will be equal to 1 and results in no correction of the estimated frequency. It is worth noting that the scaling factor is calculated based on the total elapsed microseconds and the total drift during this cycle. This means that the scaling factor is the average frequency since the last update (with 1Hz as base), indirectly implementing a low-pass filtered measurement of frequency estimation.

The original filter used α = 0.7, meaning that the estimated frequency was based 70% of the previous estimation, giving relative powerful low-pass filtering of the estimation. This will result in reasonable estimations of the frequency when the temperature is close to constant. But given that the buoy controller will be deployed offshore, where temperature fluctuations are expected, α have been tuned to allow for more responsive and aggressive estimation by relying more on the most recent frequency calculation. α is now set to a value of 0.35.

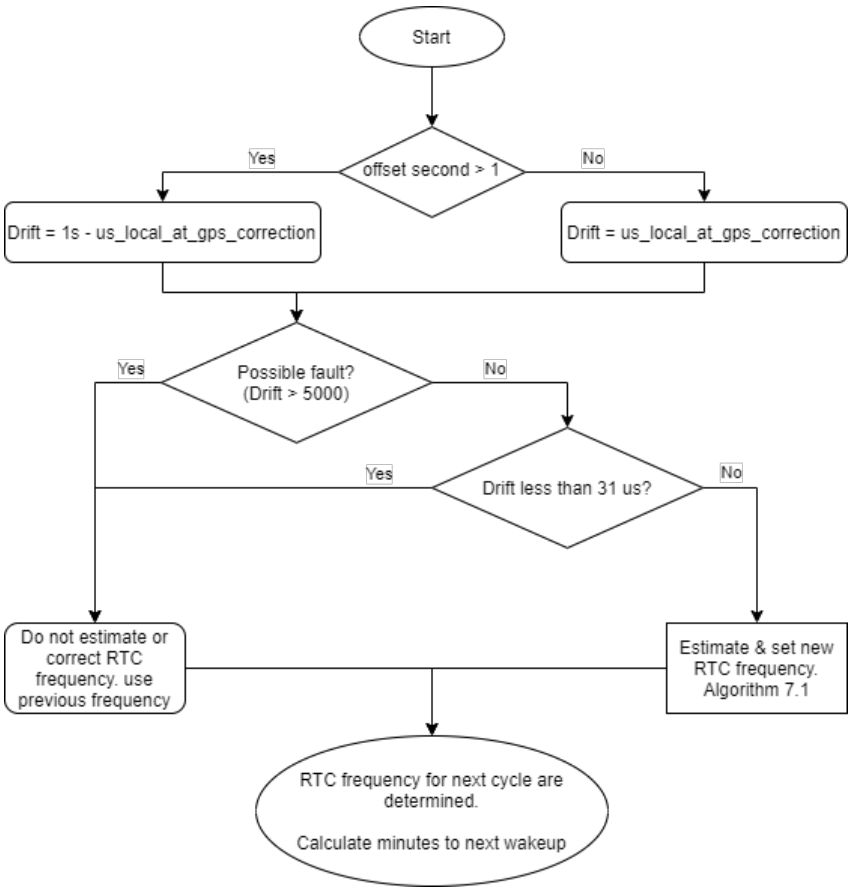


Figure 7.1: Flowchart for the frequency estimation in the Time task. Figure: Vetle B. Abrahamsen/NTNU 2022

7.2.2 Minutes to next GPS wake-up

No matter how precise the frequency is estimated, the RTC will eventually cause a drift due to temperature changes. Especially when the sleep period exceeds 1 hour. To give an example, assume that a GPS cycle and frequency estimation was completed before sun-rise with a time-drift of 10 microseconds over a period of 1 hour. By utilizing the sleep time calculation given in Equation 7.3, the new sleep time would be increased to 330 minutes or 5.5 hours. When the sun rises, temperature changes would quickly affect the RTC frequency giving large time drifts and tag detections without valid timestamps for up to 5.5 hours. To prevent this problem, the GPS sleep cycle will be interrupted whenever the temperature change exceeds a predefined threshold and estimate the RTC frequency and a new GPS sleep time. In Equation 7.3 M is the minutes to the next GPS update, β is the filter tuning parameter set to 0.5, and c is a constant that can be used to manipulate how aggressive the next calculated update should be with respect to the measured drift and previous sleep duration. c have been set to 100.

$$M[k+1] = \beta M[k] + (1-\beta)\left(c\left(\frac{M[k]}{\mu s\ local\ at\ GPS\ correction}\right)\right) \tag{7.3}$$

If the local time drift is above 500 microseconds, the application will now reset the minutes to the next GPS wake-up to one of three different predefined values depending on how long the previous sleep time was. e.g., a sleep time between 20 to 60 minutes and drift above 500us will result in 10 minutes to the next wake-up. Everything above 60 minutes results in 20 minutes to the next wake-up. This is done to prevent the 500us constraint from being violated repeatedly and to reduce the interval for RTC frequency corrections.

Drift up to 20ms have been observed, even for short sleep cycles without any significant RTC frequency corrections in advance. Hence, it is assumed that the drift is caused by the application itself and will not be treated as an actual local time drift. Whenever a considerably large drift is detected, the time task will not correct the frequency and minutes to the next GPS wake-up is reset to 2 minutes.

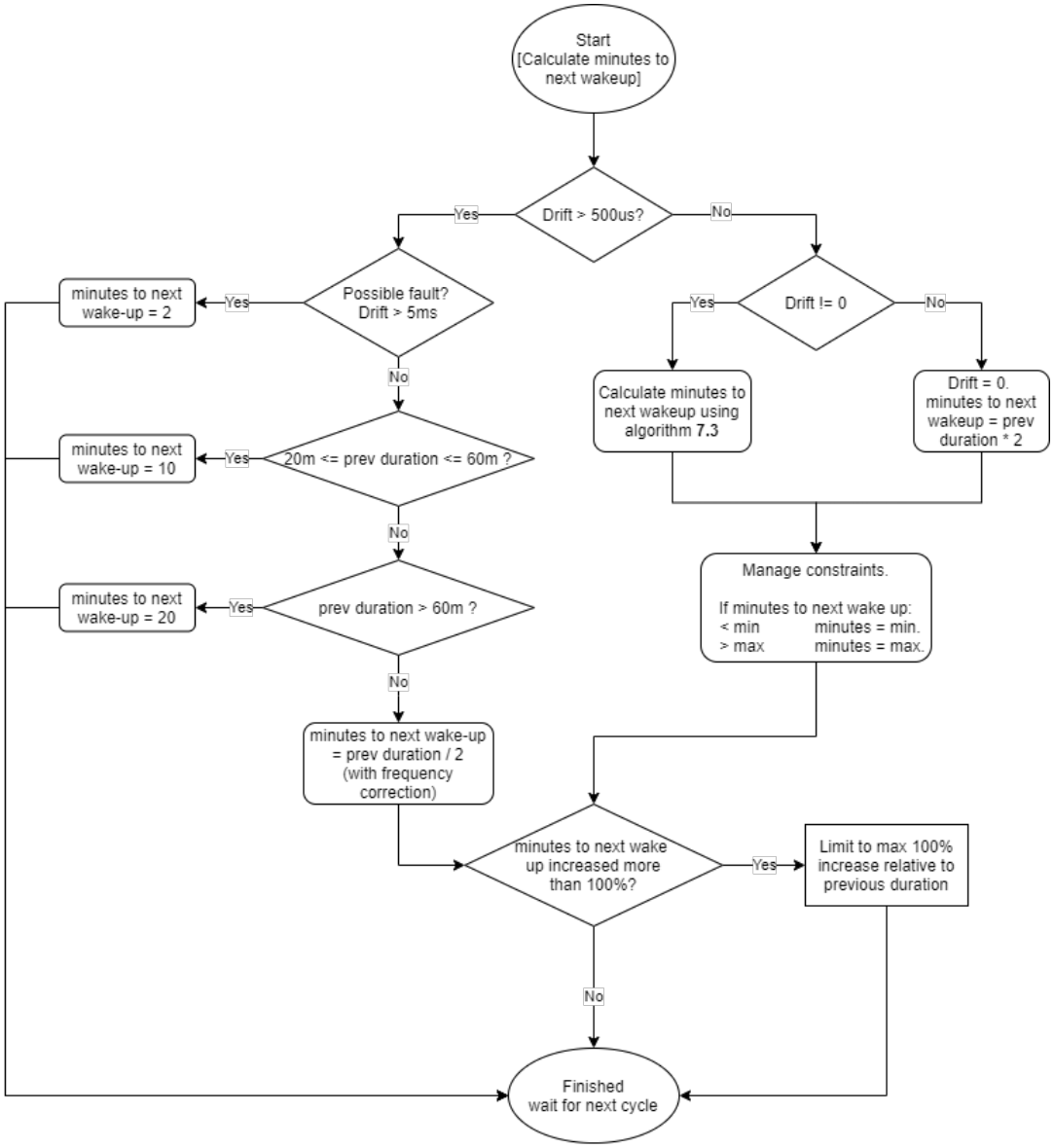


Figure 7.2: Flowchart for calculating minutes to next wake-up. With new logic for reducing chance of constraint violations. Figure: Vetle B. Abrahamsen/NTNU 2022

7.2.3 Limiting total current

The application have been updated with restrictions for accessing the modem for transmitting IoF messages while the GPS is polling data. As previously described and illustrated in Chapter 6, implementing a sequential initialization of tasks for limiting the total current demand showed to improve the battery voltage level significantly. Hence, the same concept have been introduced to the MQTT task. Whenever the GPS task is active and polling data, any transmissions of IoF messages are put on hold until the GPS have been put to sleep. If the GPS fails to poll data after several attempts, the GPS task is put to sleep, allowing IoF messages to be sent before starting the GPS task again. This will reduce the total amount of current drawn from the battery simultaneously and reduce the chance of critical voltage drops.

7.2.4 Acoustic receiver synchronization and detection

A lot of time during this thesis have been spent on debugging and optimizing the acoustic synchronization and detection tasks. The TBR serial interface is created for bi-directional communication where both the buoy controller and the acoustic transmitter are able to start bus transmissions, allowing collisions if not adequately handled. Hence, Thelma BioTel defined a period of time (listening mode) when the acoustic transmitter is silent on the bus, specifically between 9.5 to 10.5 seconds, allowing uninterrupted time synchronizations. However, during development, it was discovered that TBR 700 RT and TBLive were not working identically, and the TBR 700 solution would not detect any TBR status messages while the interface was working as intended for the TBLive solution after some optimization. It is assumed that the TBR 700 does not support the 1-second listening mode and is supported by a master thesis utilizing the TBR700 RT from 2016 stating a 1ms listening mode Efteland (2016).

During investigation, the TBR status messages were detected and injected into the first-in-first-out buffer (FIFO), but with a TBR synchronization command being scheduled every tenth second, it would cause problems every tenth minute when the TBR transmits its status message. This resulted in an issue where the tbr driver would clear the buffer in search of the received acknowledge message. To overcome this issue, the FIFO is not cleared if there exist any TBR status/detection messages in the buffer. Instead, the "ack01" will be removed from the buffer when extracting the TBR status message in the RS485 driver instead.

7.2.5 Acoustic receiver clock initialization

The tbr_synchronize task have been rewritten during this thesis. After a reboot, the cSLIM-v1 local time will be 0 until time data is polled from the GPS task. Synchronizing the acoustic receiver before a valid time would set its timestamp back to January 1970. Any tag detections during this period would not be visible on Grafana as the detection would be timestamped too far back in time.

The modification postpones any synchronization of the TBR until a valid time have been polled from the GPS task. This allows the TBR to rely on the most recent time synchronization before the reboot occurred until valid time are polled from the GPS task. It is worth noting that detections during this period, usually below 1 minute, will not be guaranteed to be within the limit of 1ms precision. However, the availability of the detection in Grafana is assumed to be more worth to the user rather than the precision for now.

Chapter 8

Back-end server for data visualization

This chapter intents to explain how data is processed in the back-end server visualization environment. Section 8.1 shows the cSLIM Node-RED flow that is responsible for decoding IoF messages and injecting data into the database. In Section 8.2, the standard that defines the naming convention, fields and tags are explained, while Section 8.3 shows figures of the different dashboards that present data to the user.

8.1 Node-RED

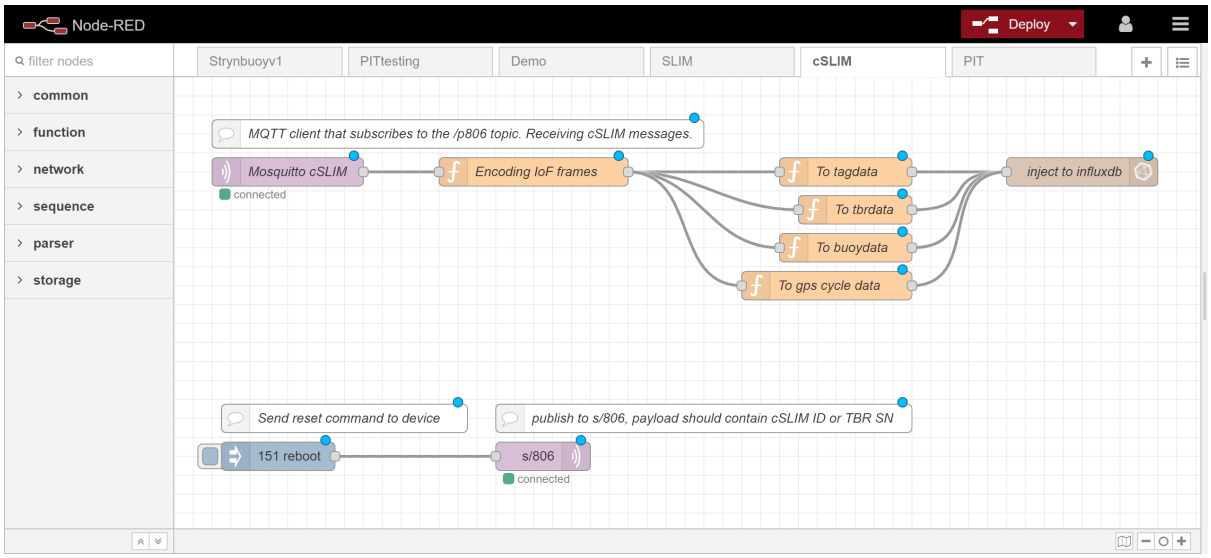


Figure 8.1: cSLIM-v1 specific Node-RED flow. Screen capture from the online editor. Figure: Vetle B. Abrahamsen/Node-RED

Figure 8.1 shows the complete flow of data in Node-RED. The Mosquitto cSLIM node is the MQTT client that subscribes to IoF messages. On received messages, the first custom function node is responsible for determining the IoF message type using header information and decoding the compressed data corresponding to the IoF message formats. The four parallel custom function nodes structure the data in a specific format before injecting the new data into its respective influxDB measurement. Data should be injected to InfluxDB using the standard given in Section 8.2.

In Section 8.1.1, Listing 8.1 provides an example of how the compressed IoF messages are decoded and stored in separate variables for further parsing. Section 8.1.2 provides Listing 8.2 that shows how the data are handled and injected into Influxdb. Note that the script have been modified to only show the processing of cSLIM GPS

cycle messages. As some of the algorithms for decoding data are owned by Thelma Biotel, the scripts will be available for users with access to the project zip file.

8.1.1 Decoding IoF messages: GPS cycle example

```
const msg               = msg.payload;
var msg_length          = msg.length;
var header_index_start  = 0;
var of  = header_index_start; // Offset in byte, start at first byte.

// Decoding header - Gives timestamp, TBR serial number and
// header flag for determining type of IoF message to be parsed.
var tbr_sn              = (msg[of]<< 6) | (msg[of+1] >>> 2);
var header_flag         = (msg[of+1] & 0x03);
var ref_ts              = (msg[of+2]<<24)|(msg[of+3]<<16)|(msg[of+4]<<8)|msg[of+5];

// Decoding payload - Based on header flag.
var payload_index_start = 6;
of = payload_index_start;
var frame   = {};
var frames  = [];

switch (header_flag) {
    case 0:             // Tag detections or TBR sensor data frames.
        .....
    case 1:             // Buoy controller status frame
        .....
    case 2:             // Reserved for future message formats
        .....
    case 3:             // cSLIM GPS cycle
        while(of < msg_length){
            value1 = (msg[of]<<24)|(msg[of+1]<<16)|(msg[of+2]<<8)|msg[of+3];
            value2 = (msg[of+4]<<8) | msg[of+5];
            value3 = (msg[of+6]<<8) | msg[of+7];
            value4 = (msg[of+8]<<24)|(msg[of+9]<<16)|(msg[of+10]<<8)|msg[of+11]);
            frames.push({drift_usec:value1,last_gps_cycle_m:value2,next_gps_cycle_m:value3,
    frequency_estim:value4});
            offset += 12;
        }
        frame.gps_cycle_packet = frames;
        break;
    }
const header    = {tbr_sn:tbr_sn, h_flag:header_flag, ref_ts:ref_ts};
frame.header    = header;
msg.payload     = frame;
return msg;
```

Listing 8.1: Custom function block: Decoding IoF message - only showing GPS cycle messages. Variable names have been modified to reduce size

8.1.2 Injecting IoF messages to Influxdb: GPS cycle example

```
var data_points = [];
const gps_cycle_msgs = msg.payload.gps_cycle_packet;
const ref_ts         = msg.payload.header.ref_ts;
const tbr_sn         = msg.payload.header.tbr_sn;
const h_flag         = msg.payload.header.h_flag;

if(h_flag == 3){
    for (let i = 0; i < gps_cycle_msgs.length; i++){
        time            = ref_ts;
        drift           = gps_cycle_msgs[i].drift_usec;
        last_interval   = gps_cycle_msgs[i].last_gps_cycle_m;
        next_interval   = gps_cycle_msgs[i].next_gps_cycle_m;
        wakeup_time     = time + next_interval*60;              // Convert to seconds
        estim_freq      = gps_cycle_msgs[i].frequency_estim;

        let data_point = [
            { // Influxdb Fields
            time            :   time,
            drift           :   drift,
            last_interval   :   last_interval,
            next_interval   :   next_interval,
            wakeup_time     :   wakeup_time,
            estim_freq      :   estim_freq
            },
            { // Influxdb Tags
            tbr_sn      :   tbr_sn
            }
        ];
        data_points.push(data_point);
    }
}
msg.payload     = data_points;
msg.measurement = "gpscycle_cslim"; // Influxdb measurement definition.

if (msg.payload.length > 0){
    return msg;
}
```

Listing 8.2: Custom function script: To GPS cycle data

8.2 InfluxDB

After setting up InfluxDB on the virtual server running on NTNU, no additional work is required. Adding new databases and managing or displaying existing data can be done through SSH using user credentials. However, this will only be necessary on rare occasions. New measurements, fields and tags within a database are simply generated once new fields, tags or measurements are sent to the database the first time. The database used in Internet if Fish project is called "Buoys" and manages four different measurements, one for each IoF message type. InfluxDB supports several databases allowing unrelated projects to be separated from each other.

8.2.1 IoF InfluxDB storage standard

A standard format for storing data in the "Buoys" database have been suggested by J. Kornberg. The format allows the database and visualization tool to work seamlessly between the two current IoF buoy controllers SLIM and cSLIM in addition to future versions, as long as the data are injected to InfluxDB using the same format as defined below. Tag data(8.2.1.1), TBR Sensor data(8.2.1.2) and Buoy data(8.2.1.3) are used by both cSLIM and SLIM. While cSLIM GPS cycle(8.2.1.4) have been designed by the author for use with cSLIM specifically.

8.2.1.1 Tag data

Table 8.1: InfluxDB standard for fields and tags in the "tagdata" measurement in "Buoys" database.

Measurement: "tagdata", Database: "Buoys"		
	Name	Description
Fields	"time"	Tag detection timestamp - seconds
	"ms"	Tag detection timestamp - milliseconds
	"tag_data"	Data sent from tag if any
	"snr"	Signal-to-noise ratio of tag detection
Tags	"tbr_sn"	Serial number of the TBR which detected
	"tag_prot"	Transmit protocol used by tag
	"tag_freq"	Transmit frequency used by tag
	"tag_id"	ID of detected tag

8.2.1.2 TBR Sensor data:

Table 8.2: InfluxDB standard for fields and tags in the "tbrdata" measurement in "Buoys" database.

Measurement: "tbrdata", Database: "Buoys"		
	Name	Description
Fields	"time"	TBR sensor reading time in seconds
	"temperature"	Temperature data from TBR
	"noise_avg"	Average noise in noise logging frequency
	"noise_peak"	Peak noise in noise logging frequency
Tags	"tbr_sn"	Serial number of the TBR
	"noise_freq"	Noise logging frequency

8.2.1.3 Buoy data:

Table 8.3: InfluxDB standard for fields and tags in the "buoydata" measurement in "Buoys" database.

Measurement: "buoydata", Database: "Buoys"		
	Name	Description
Fields	"time"	Approximate buoydata time in seconds
	"latitude"	Latitude in degrees
	"longitude"	Longitude in degrees
	"pdop"	Position dilution of precision
	"fix"	Fix type
	"sat"	Satellites used in nav solution
	"bat_status"	Battery status in voltage
Tags	"tbr_sn"	Serial number of the TBR

8.2.1.4 cSLIM GPS cycle:

Table 8.4: InfluxDB standard for fields and tags in the "gpscycle" measurement in "Buoys" database.

Measurement: "gpscycle_cslim", Database: "Buoys"		
	Name	Description
Fields	"time"	Time in seconds
	"drift"	Previous GPS cycle local time drift
	"last_interval"	Previous GPS cycle duration in minutes
	"next_interval"	Current GPS cycle duration in minutes
	"wakeup_time"	next GPS wake-up time. UTC
	"estim_freq"	Current estimated frequency
Tags	"tbr_sn"	Serial number of the TBR

8.3 Grafana

In Grafana, the displaying part of the visualization environment, queries can be made to access authoritative information from the database - InfluxDB. Figure 8.2 shows how the "correction in estimated RTC frequency" graph in the cSLIM GPS cycle dashboard is configured to receive estimated frequency data from the gpscycle_cslim measurement with the respective tbr serial number. As Grafana integrates the InfluxDB in their solution, all tags, fields, and measurements are automatically listed when configuring and makes Grafana easy to use. The following subsections will provide figures of the four different dashboards available to the users.

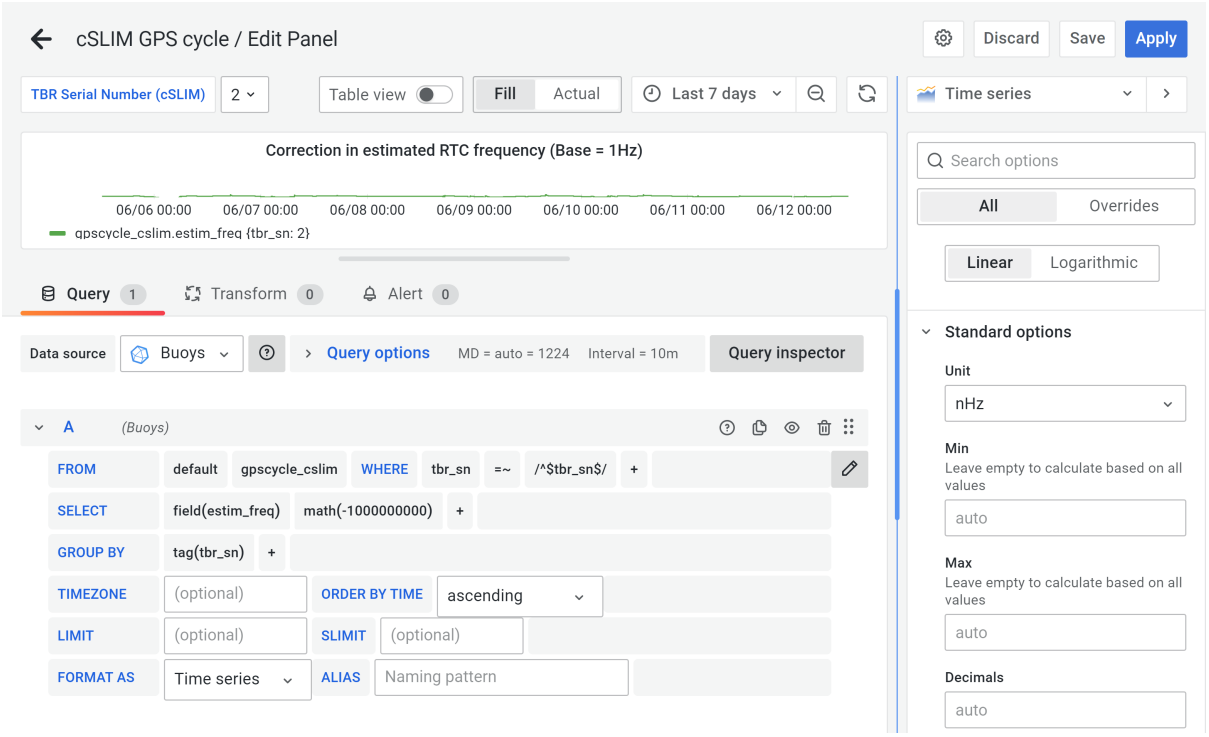


Figure 8.2: Polling data from InfluxDB in Grafana panel - edit mode in GPS cycle dashboard, estimated frequency graph. Figure: Vetle B. Abrahamsen/Grafana

8.3.1 Dashboard - Buoy data

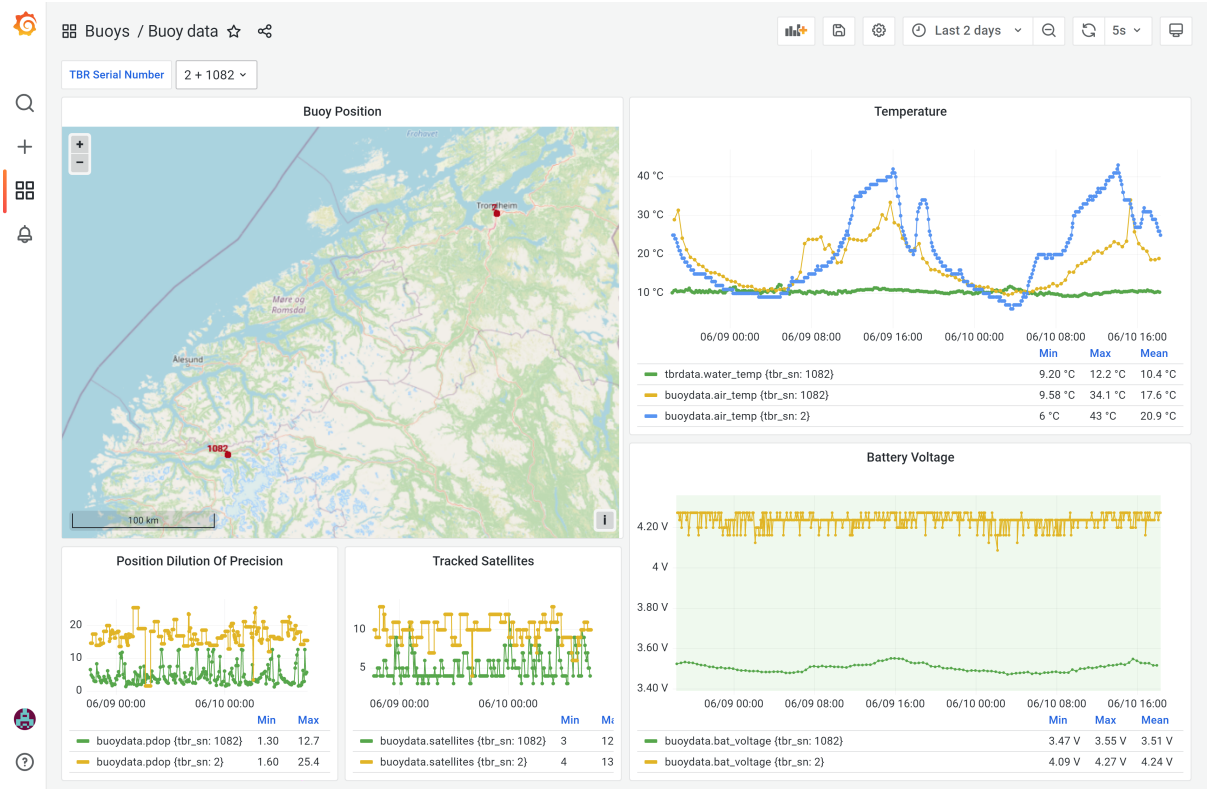


Figure 8.3: Grafana dashboard: Buoy status, both for cSLIM-v1 (ID 2) and SLIM (ID 1082) in same dashboard. Figure: Vetle B. Abrahamsen/Grafana

8.3.2 Dashboard - TBR data

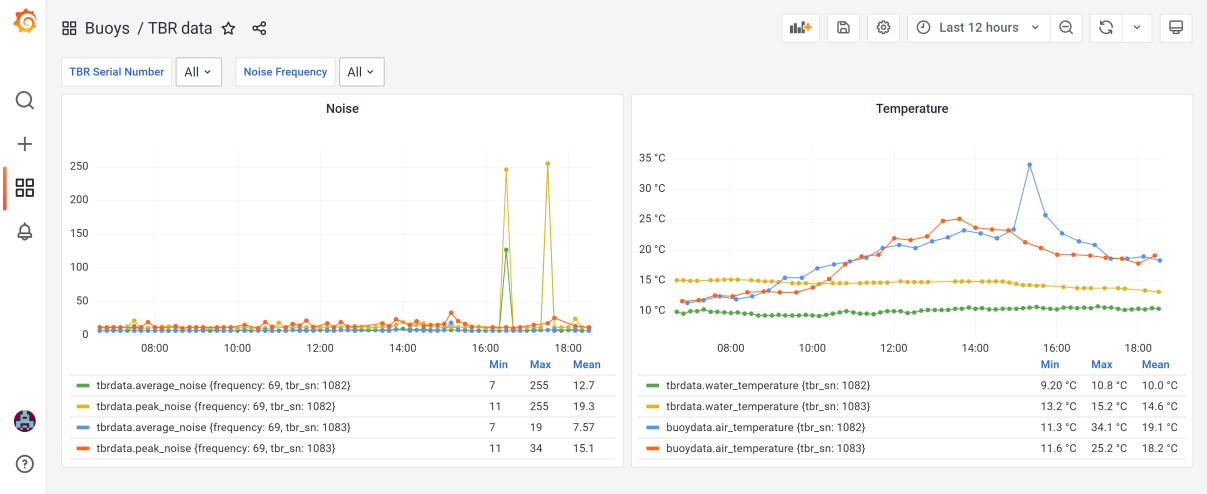


Figure 8.4: Grafana dashboard: TBR status. Figure: Vetle B. Abrahamsen/Grafana

8.3.3 Dashboard - Tag detections

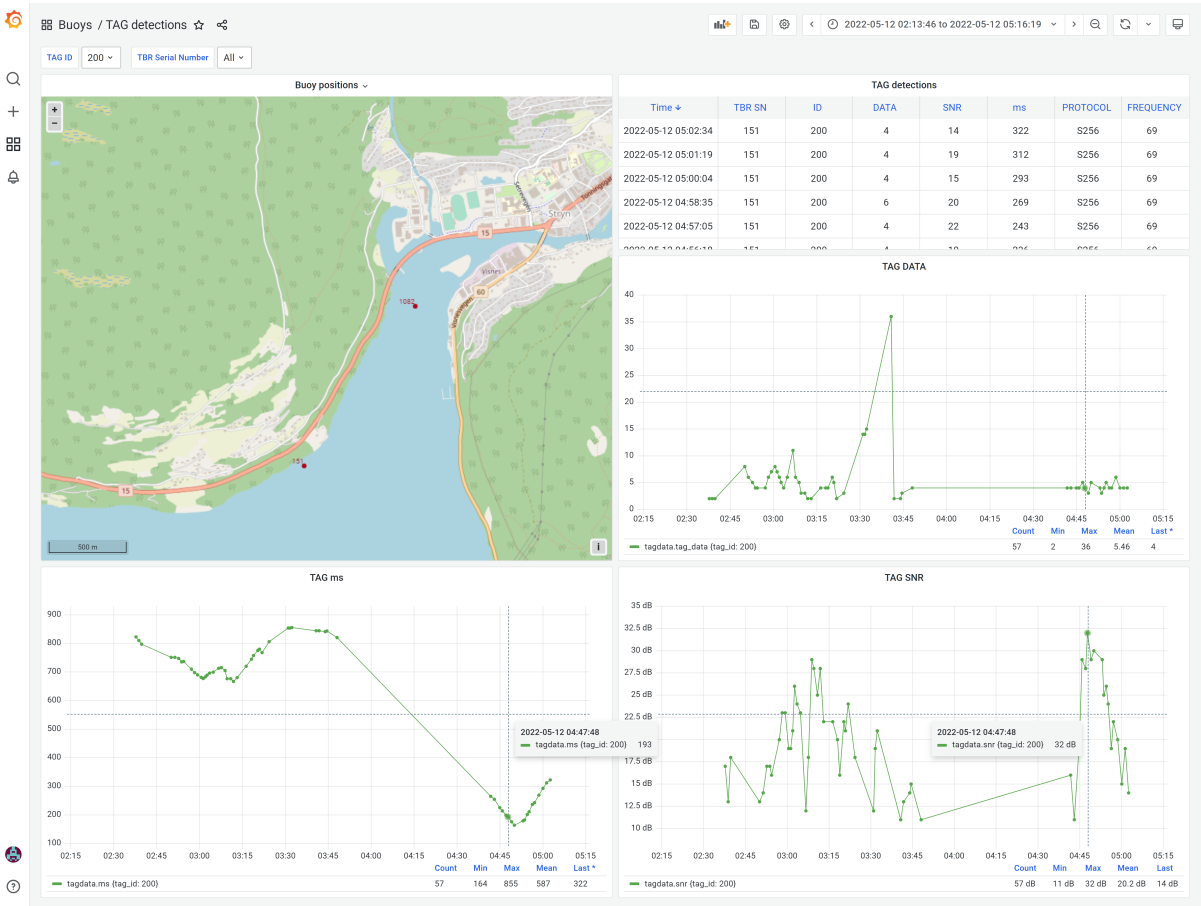


Figure 8.5: Grafana dashboard: Tag detections. Figure shows that tag ID were detected both by TBR serial number 1082, then by 151. Figure: Vetle B. Abrahamsen/Grafana

8.3.4 Dashboard - GPS sleep cycle

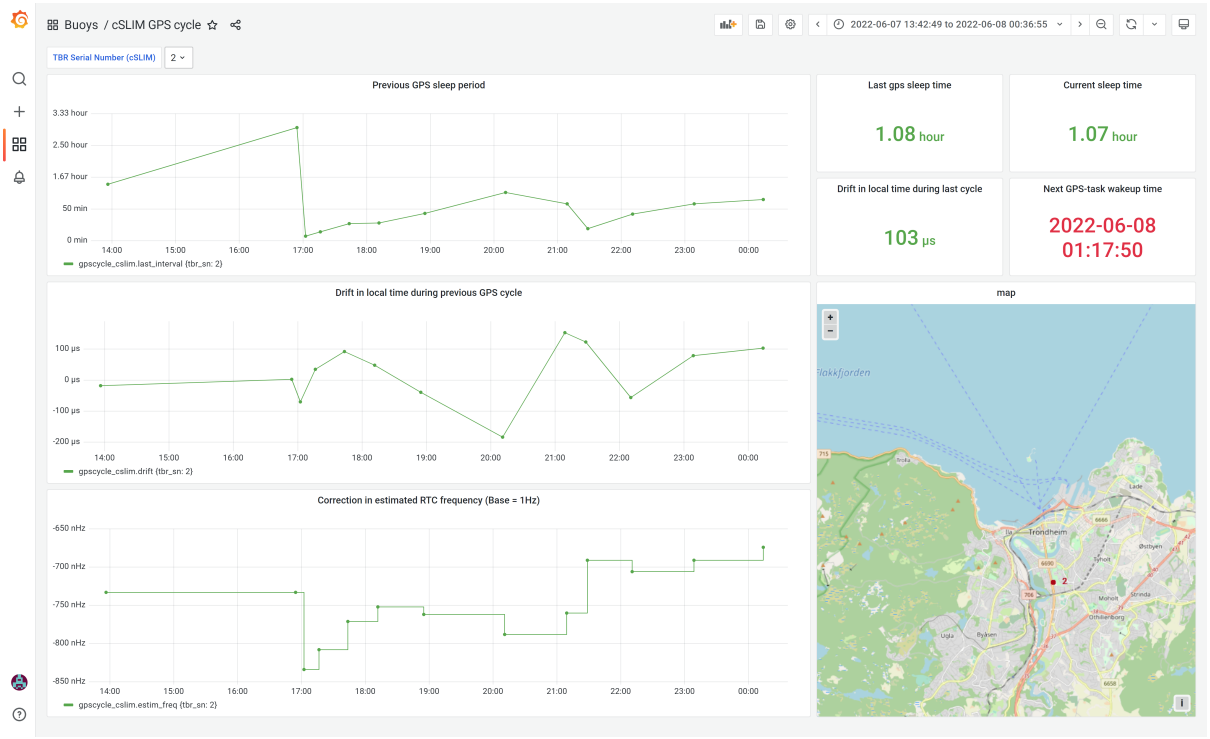


Figure 8.6: Grafana dashboard: cSLIM GPS cycle. Figure: Vetle B. Abrahamsen/Grafana

Chapter 9

Field test in Stryn, Norway

Smolt migrates to the sea from their natal rivers as a part of their natural life cycle. Usually, the mitigation occurs during the spring and early summer when the smolts are 1-6 years old and 15-25cm long Bjerck (2021). Prior to the migration, Biologist Henning andrè Urke captured, tagged and released 200 fish that were likely to migrate to the ocean during the following spring.

The rig (full buoy) was constructed by Terje Haugen at NTNU and can be seen in Figure 9.1. Only one buoy utilizing the cSLIM-v1 was deployed. However, Jon A. Kornberg had previously deployed his buoy further up the river controlled by SLIM. The idea was that detections made by Kornberg's buoy should eventually be detected by the cSLIM buoy deployed further out towards the fjord, making it possible to monitor the fish as they started their migration out to the sea in real-time.

This chapter explains the testing and results from the first actual deployment of a buoy using both the cellular cSLIM-v1 and the TB Live acoustic receiver. The buoys' performance and analysis of received data will be in focus.

Figure 9.1: Deployed buoy during field test. Photo: Vetle B. Abrahamsen/Stryn 2022

9.1 Setup

The deployed buoy used the cellular NB-IoT as LPWAN technology with the ublox GNSS module for polling time and position data. The acoustic telemetry solution consisted of the new TB Live powered by Pololu's external 9V step-up voltage regulator. The power solution is the same as described in section 6 but was later replaced by a new battery solution. In addition, a signaling light was strapped to the top of the rig to prevent boats from colliding during the night.

9.2 Tests

9.2.1 cSLIM-v1: Power solution

Goal: The power solution should be able to keep the buoy controller operational without restarts due to voltage drops. The included supercapacitor should be able to aid the power solution during larger current spikes.

Method: Monitor the cSLIM GPS cycle in order to determine if the buoy controller rebooted during operation. An estimated frequency of 0 Hz means that the time synchronization task was restarted and implies a system reboot of the cSLIM-v1.

Results: During the deployment using the power solution described in Chapter 6, the buoy controller repeatedly rebooted. Although reboots can occur when the network connection is lost over a longer period, the voltage level seen in Figure 9.2 in addition to the repeated reboots of the system seen in the estimated frequency plot gives reason to believe that the reboots were caused by the battery not being able to provide the power needed while using the NB-IoT modem. Hence, The supercapacitor solution is not enough to aid the original battery. A new battery package with 3x1,5V alkaline batteries in series (Energizer max plus type D), providing 4,5V were made and used for the rest of the field test (With no supercapacitor). After replacing the battery solution with a battery pack with a higher nominal voltage and without limits on the maximum current loads, cSLIM-v1 could operate without repeatedly rebooting.

Note: All plots in this section, except Figure 9.2, use data captured while using the new and improvised alkaline battery pack.

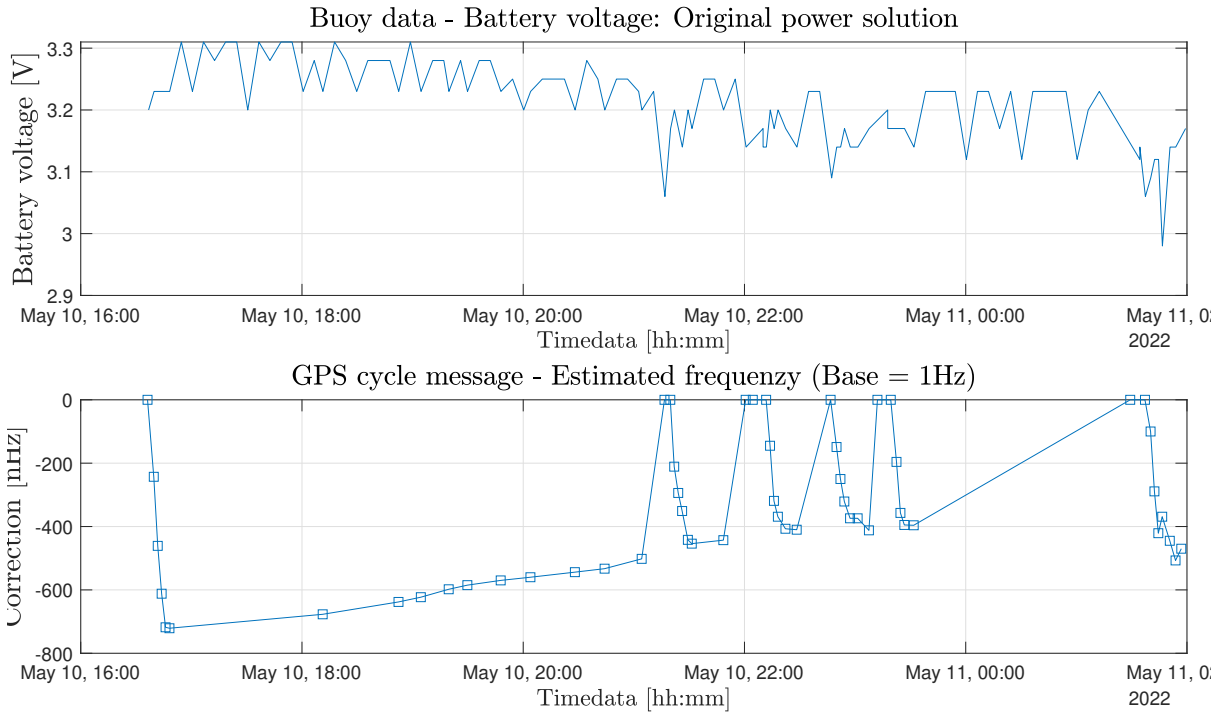


Figure 9.2: Field test data: Showing the voltage levels from the buoy status messages and the estimated frequency during the same time period as a frequency correction of 0 indicates a reboot have occurred. A total of five reboots were detected and is caused by voltage drops on VDD during polling GPS data and transmitting messages. Figure: Vetle B. Abrahamsen/Grafana

9.2.2 TBR Serial Interface: Detecting tbr messages and tags

Goal: Reliable communication with the acoustic receiver. All messages from the acoustic receiver should be received and forwarded to the the IoF application layer.

Method: Verify that TBR status messages are received on Grafana every 10 minutes without any loss. Given that a tag ID have been detected, a list of all tag IDs will specify the acoustic transmission interval for that specific tag id. Hence, while the fish is close to the acoustic receiver, a detection should be visible on Grafana within that given interval. Deviations imply a lost detection. However, detections may be lost due to noisy environments and the TBR status message showing average and peak noise should be taken into consideration.

Result: During testing, while the buoy controller was operational, no TBR status messages were lost as a TBR status message have been received every tenth minute, as see in Figure 9.3. A fish with tag ID 200 was detected and shown in Figure 9.4. Between 04:50 and 04:53, no acoustic tags were available for 180 seconds. This specific tag should have 30-90 seconds transmitting interval, implying that at least one detection was lost. However, the cause of the lost package can be related to noise or other environmental factors. Hence, the TBR serial interface needs more tests in order to be fully verified. The buoy controller was not online until minutes before the first detection. As more detections have been captured while the fish was swimming away from the buoy than towards, it is assumed that the fish was already within the detection area of the acoustic receiver when the buoy controller was connected to the network. Hence, the first couple of detections as the fish entered the detection area might have been missed.

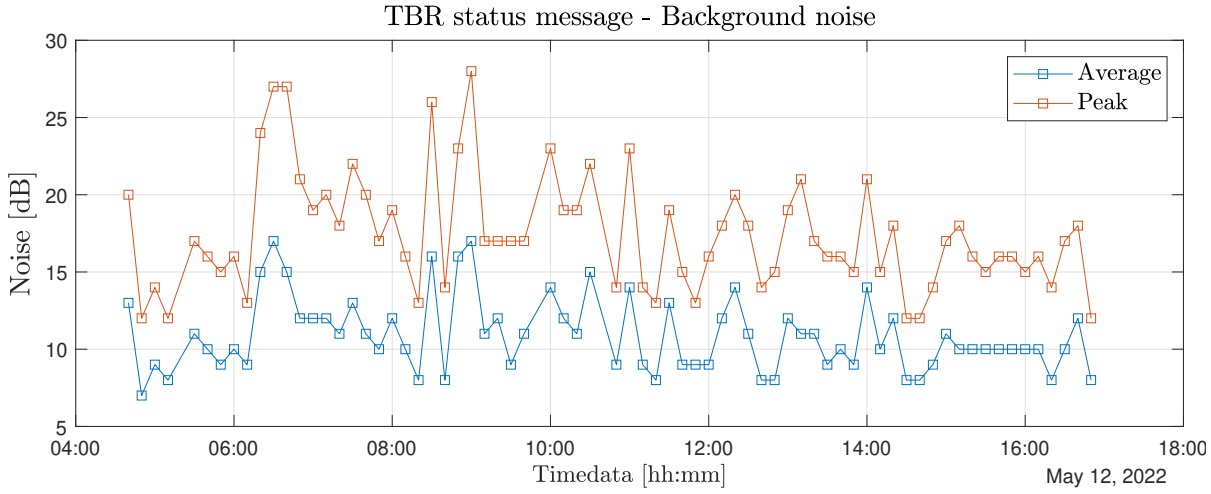


Figure 9.3: TBR status message data: Background noise. This message have an fixed interval of 10 minutes. No data lost. Figure: Vetle B. Abrahamsen/NTNU 2022

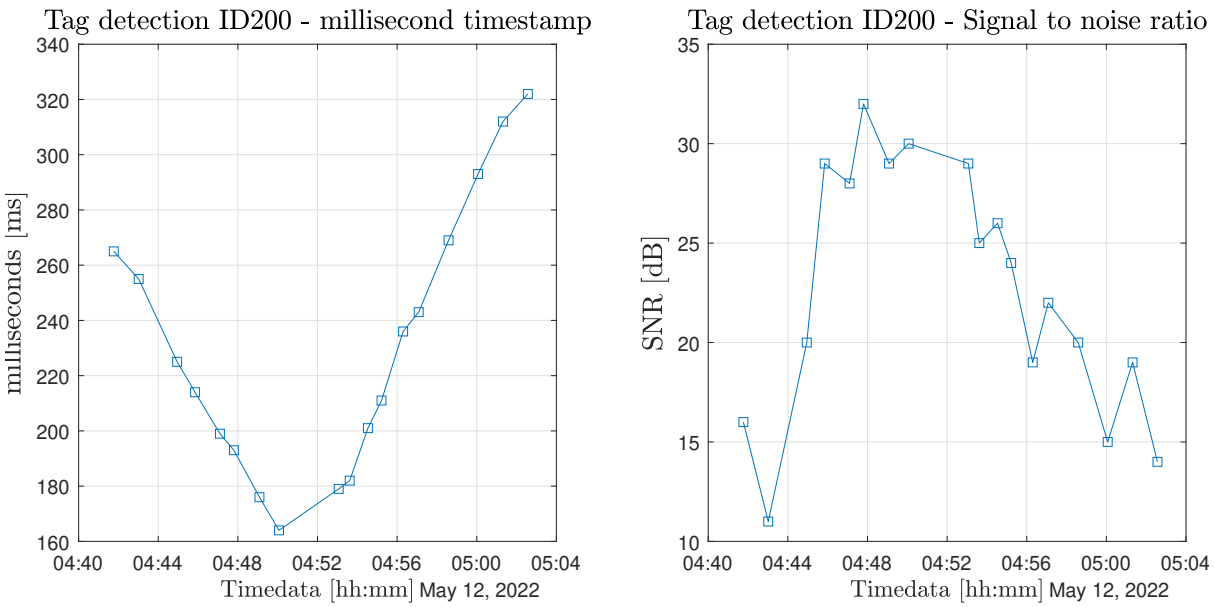


Figure 9.4: Detection of tag ID 200, salmon, 18.1cm, 42g. Plot shows the millisecond part of the timestamp and signal to noise ratio and can give information about the behaviour when analysed. Figure: Vetle B. Abrahamsen/NTNU 2022

Note: Comparing the millisecond part of the timestamp and the signal-to-noise ratio in Figure 9.4 gives a foundation for extracting behavioral information. As tag ID 200 was first detected further up the river by the SLIM buoy, more information regarding its behavior can be read. A complete analysis of the ID200 detections will be covered in Section 9.3.

9.2.3 Time synchronization

Goal: Be able to keep the local time-synchronized with GPS time by estimating the RTC frequency. The synchronization error should not be greater than ± 500μs. Doing so should make it possible to increase the GPS wake-up period and keep buoys synchronized with other deployed buoys.

Method: GPS cycle data will be sent to the cSLIM GPS cycle dashboard in Grafana and will be used to analyze the time synchronization method. Any drift more significant than ± 500μs will not be considered successful.

Results: Figure 9.5 shows the GPS cycle data during the longest up-time of the deployed cSLIM-v1. The period between GPS updates increases as the RTC frequency converges to the actual frequency and drifts towards 0. When the drift is significant, the GPS update period is reduced. At time = 14:00, the GPS slept for 90 minutes and resulted in a drift of -385us. The most prolonged GPS sleep period was 2 hours, giving an acceptable local time drift of 244us. The sleep time plot shows how the GPS sleep interval oscillates and indicates that the time synchronization is not fully optimized as expected with parameters tuned for indoor use.

From 18:00 to 19:00, the GPS slept for 50 minutes, resulting in a local time drift of 1s (not shown in the plot since this large value would cause a scaling that would make the other data points unreadable). See Grafana during this time period for the entire plot). The GPS sleep cycle prior to the 1s drift was 20 minutes, with a time drift of -70us. The estimated frequency based on the previous sleep cycle caused minor corrections and should not have resulted in a local time drift of 1 second. Hence, it is assumed that this large drift was caused during the time synchronization task itself. With that assumption, detections during this GPS sleep period might still be within the timing constraint but have to be regarded as non-valid due to an unknown actual drift.

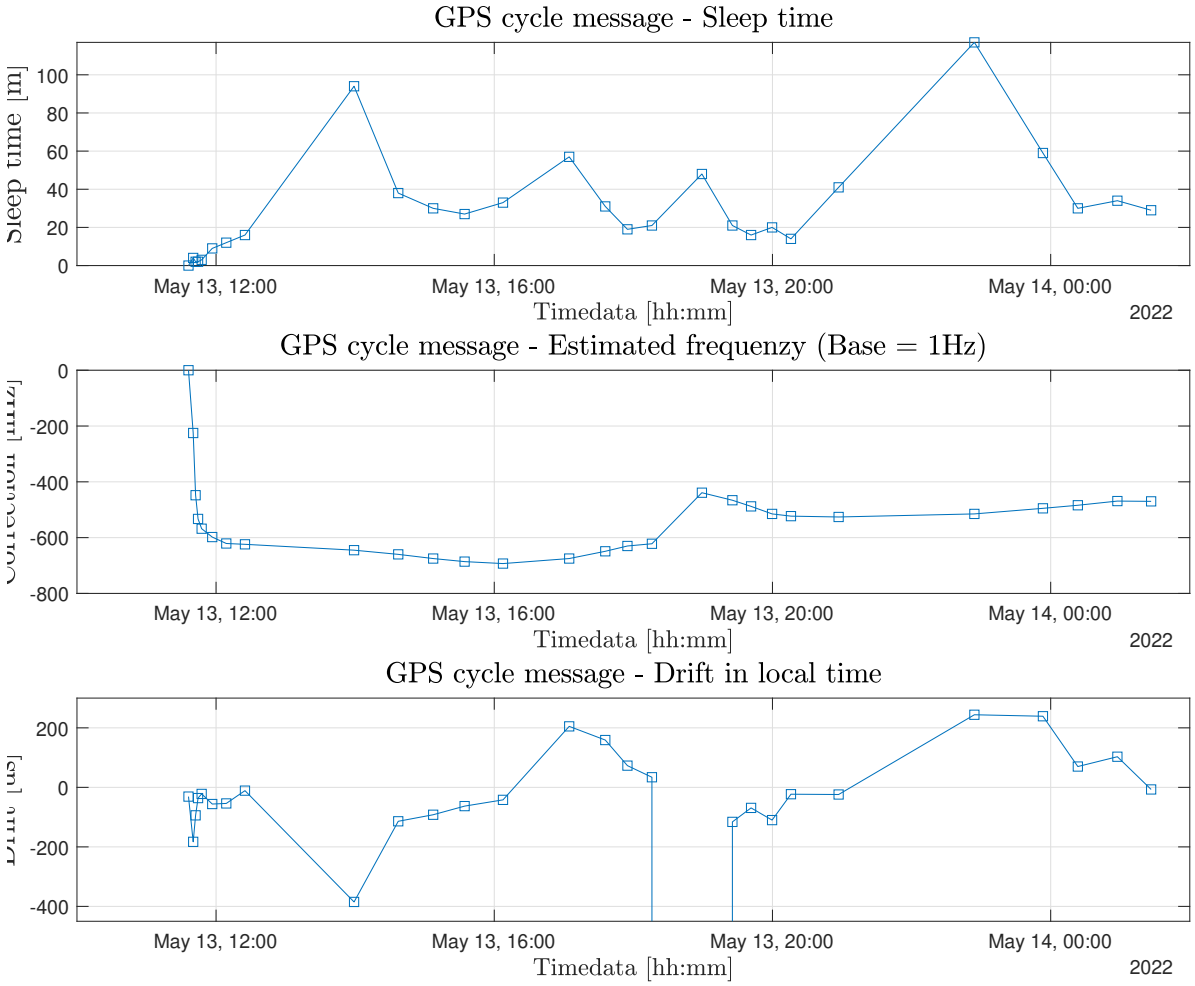


Figure 9.5: GPS cycle data downloaded from Grafana showing the system performance in frequency estimation and timekeeping. Figure: Vetle B. Abrahamsen/Grafana

Note: This section only covers the result from the field test itself. More in depth analysis of the time synchronization and optimization will be covered in Chapter 10 Section 10.2.

9.3 Detection analysis

All tagged fish have been logged with information such as weight and length in addition to the location of capture. Given the logged information, tag ID 200 is a salmon captured in the lower part of the Stryn river with a length of 18.1cm and weighted 42g. Tag ID 200 were detected both by the cSLIM buoy and the SLIM buoy, deployed about 1km further up the river, as the fish started its migration to the sea.

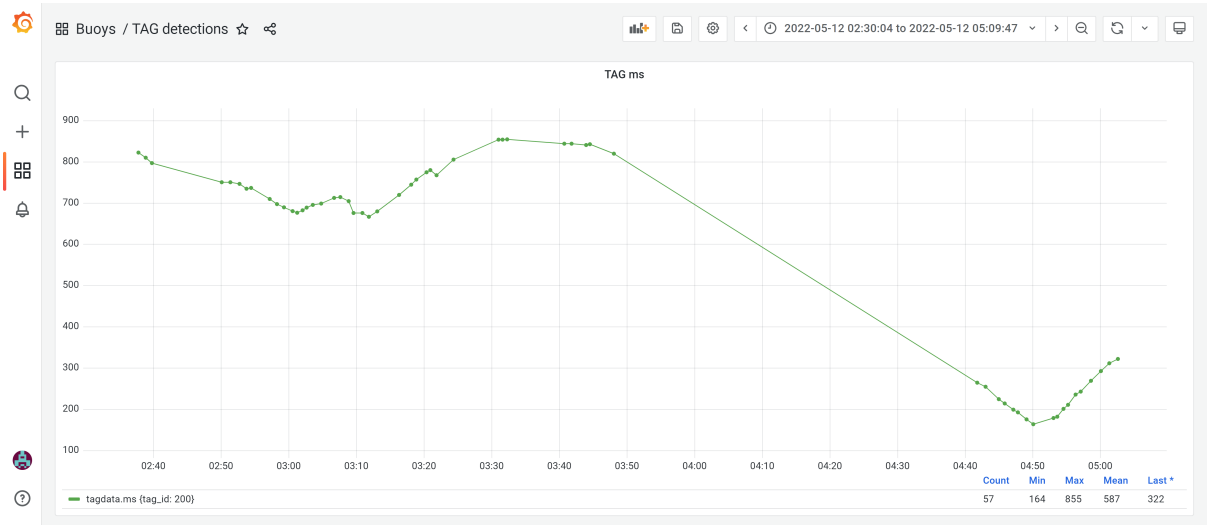


Figure 9.6: Illustrates how a specific tag can be analysed in Grafana. Figure 9.7 will be used for the analysis with the data exported from Grafana. Figure: Vetle B. Abrahamsen/Grafana

Figure 9.6 Shows how the detection history of ID200 would look for the user in the Grafana tag detection dashboard. The graph have been sorted by TAG ID, meaning that all detections will be visualized regardless of the acoustic receiver serial number, making it easy to analyze and monitor the behavior of specific fish IDs. In this section, the behavior of ID200 will be analyzed. Figure 9.7 will be used for the rest of the analysis.

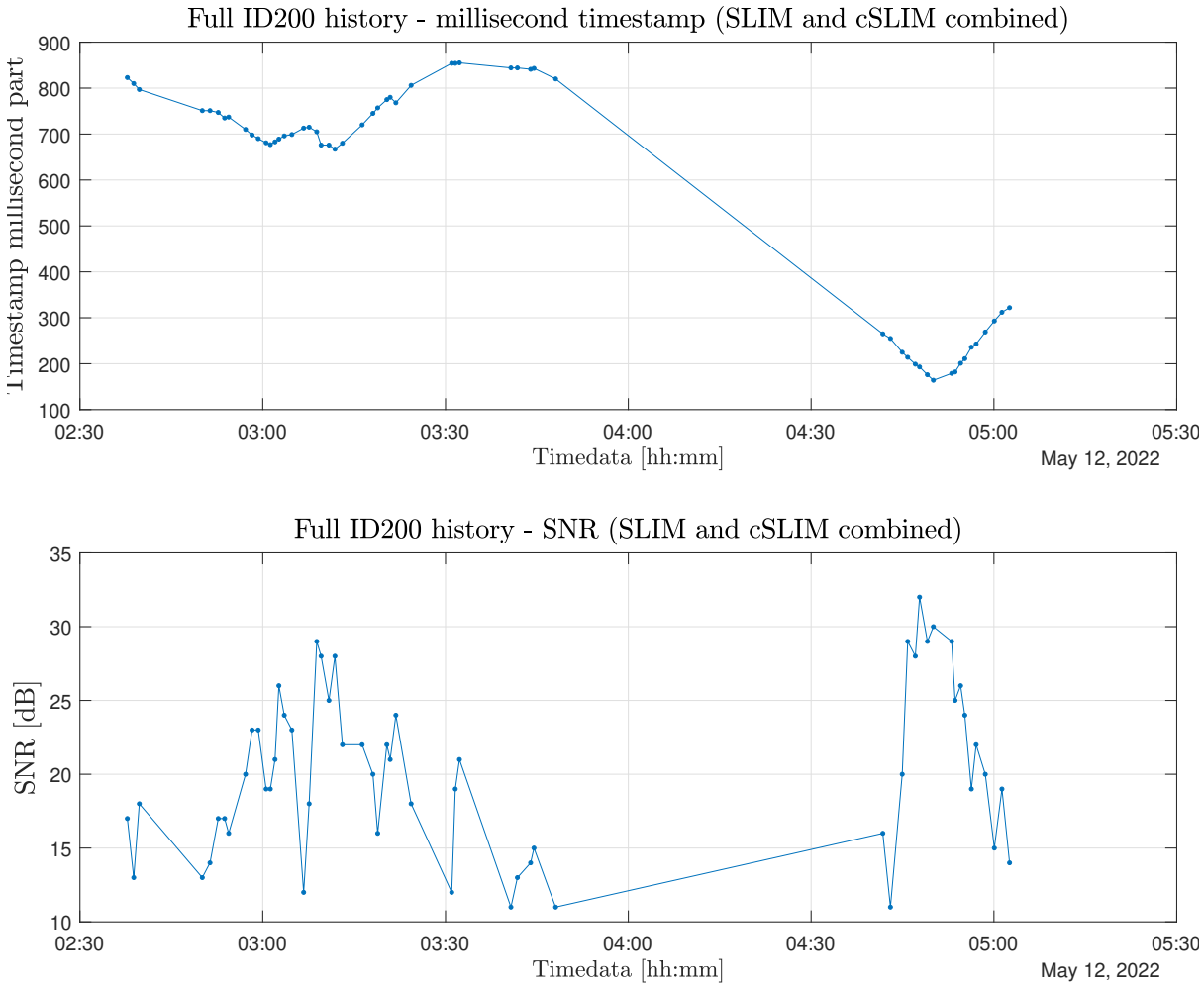


Figure 9.7: Tag detections from two different boys. ID200 were first detected by the SLIM buoy before the cSLIM buoy detected the fish further down the river. Figure: Vetle B. Abrahamsen/Grafana

The first detection was captured 12th. of May, 02:37 by the SLIM buoy furthest up the river. The millisecond part of the timestamp, as it either increases or decreases, indicates whether the fish is swimming away from or towards the boy. This is due to the acoustic tag constantly transmitting signals at whole seconds. Drift in the tags internal clock causes the temperature-based drift, and movement causes an increase/decrease in the millisecond part of the timestamp, which is significant compared to the constant drift due to its internal clock. Indications of if the fish is swimming towards/away from the buoy can also be seen by the signal-to-noise ratio (SNR) as the signal intensity decreases as the distance from the receiver increases.

Looking at Figure 9.6 around 03:00 ID200's millisecond timestamp changes from decreasing to increasing, implying that the distance to the SLIM buoy reached a minimum before swimming away from the buoy. Minutes later, the smolt stops and swims towards the buoy again, meaning that ID200 had not yet fully started its migration out to the sea. In fact, the factors that trigger the smolt to start its journey is a mystery that biologists have been trying to understand for decades. However, water temperature above 10°C and increased water flow are often assumed to be the key triggering factors for smolt migration Jutila et al. (2005).

03:11, ID200's millisecond timestamp changes from decreasing to increasing, implying that the smolt started the migration towards the sea. This observation is supported by the SNR value being at its maximum, meaning that the signal intensity was at its peak in this period. The last detection by the SLIM buoy was captured at 03:48 but was again detected by the cSLIM-v1 buoy one hour later at 04:41.

As the buoy was not operational until minutes before the first detection, it is assumed that the first detections might have been missed. However, the shape of the millisecond timestamp part has a distinct "V-Shape", with a steeper slope, implying that the smolt had caught up to speed and was swimming at a steady paste past the buoy. With the SNR being relatively high, it is believed that the fish was swimming close to the buoy. The fish reached its closest distance to the cSLIM-v1 buoy at 04:50, supported by SNR data being at its peak.

By measuring the distance between the SLIM and cSLIM buoys, approximately 1.18km, and calculating the time difference between the two points where ID200 were closest to both the SLIM and cSLIM buoy, an average speed of 0.2m/s has been approximated. However, the average speed is assumed to be underestimated as the salmon had not fully caught up to speed when analyzing the data detected by the SLIM buoy. Hence an average of 0.25m/s might be more correct. With a length of 18.1cm, this converts to approximately 0.9 to 1.25 body length/second (BL/s).

The distance from Stryn and all the way out to the sea is approximately 100km when following the fjord's shoreline. With the calculated speed of 0.2m/s or 0.25m/s, the salmon with ID200 should reach the sea after 5 days and 20 hours or 4 days and 15 hours. Other stations with acoustic receivers have been placed along the fjord, but in order to analyze the entire history of ID200, the data have to be physically extracted by the acoustic receivers as the stations do not support real-time transmissions. This emphasizes the practicality of the real-time buoy controllers.

Chapter 10

Additional testing and overall results

The following sections provide more in-depth tests of the system as the field test did not provide enough data. After the deployment of the buoy in Stryn, a cSLIM-v1 have been continuously operating in Trondheim with the same power solution that was improvised in Stryn and with the exact same software. This has been done to exclude potential sources of error related to the offline field-deployed buoy and gather more test data.

10.1 Labtest - Full system

Goal: To test the full IoF/cSLIM-v1 solution synchronization performance and to forward all detections from the acoustic receiver and other telemetry to the backend visualization solution.

Method: The improvised alkaline battery solution will power the cSLIM-v1 in order to resemble the deployed buoy. The system interfaces with a TBR700-RT acoustic receiver submerged in water together with the acoustic transmitter (tag). The cSLIM-v1 was placed outside for good cellular coverage and GNSS reception in addition to seeing the effect of temperature changes. At the same time, the acoustic receiver and tag are kept inside to simulate the stable temperature in a realistic submerged deployment environment. The software solution have been configured to use NB-IoT, GNSS- and time task for local time synchronization, TBR detect- and synchronization task. All IoF messages are enabled and will be displayed on Grafana as data is received. The buoy data dashboard in Grafana should show a water temperature every 10th. minute to verify that all TBR status messages are detected. The test tag will transmit all standard telemetry and a number that increments from 0 to 255. A skipped number means that the acoustic signal was not detected. Looking at the signal-to-noise ratio and TBR background noise during that period gives additional information as to whether the undetected acoustic signal was lost due to the receiver or the buoy controller. As the tag have an internal crystal oscillator and is submerged in constant temperature, the millisecond part of the timestamp is expected to have a constant drift.

Results The test shows that the system is successfully able to maintain a synchronized local time within the time constraint using the tuned filter and GPS sleep time calculation. Furthermore, as seen in Figure 10.1, the time task decreases the wakeup frequency while keeping the timing constraint. As a result, the current consumption will be reduced.

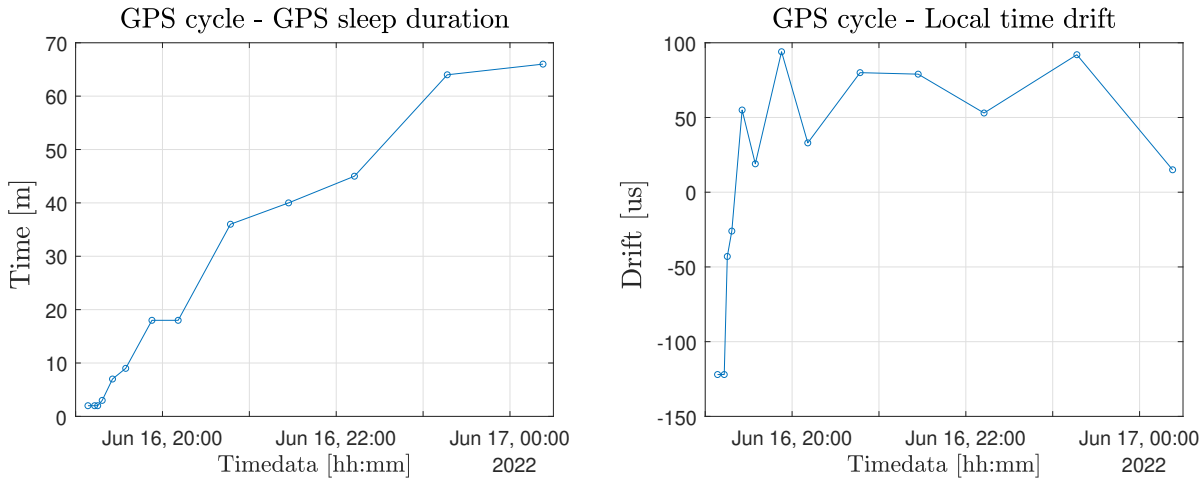


Figure 10.1: Lab test data: Full system test - GPS sleep duration and drift. Figure: Vetle B. Abrahamsen/NTNU 2022

During testing, the system did not detect 37 out of 312 acoustic signals. This can be seen as gaps in the incremented tag payload in Figure 10.3. However, looking at the signal-to-noise ratio graph, a correlation between lost detections and the reduced signal-to-noise ratio suggests that signals have been polluted by noise. This hypothesis is supported by the very high average and peak background noise measurement during the respective period. Hence, it is assumed that the acoustic receiver did not receive the signals due to noise and that the acoustic receiver interface is working. Additionally, the buoy controller was able to detect all TBR status messages (e.g. background noise messages) without a single loss. This was also the case during the field test using TBLive and leads to the belief that the buoy controller is able to detect all data sent from the acoustic receiver.

As the tag was submerged in water with close to a constant temperature, the drift seen in Figure 10.2 is as expected, a constant and stable drift. This allows the fish behavior to be monitored as an increase or decrease in the speed component that acts towards/away from the receiver will result in a significant change in the millisecond part relative to the drift caused by the crystal oscillator. A 9ms spike was observed 20:00 and could have been caused by a false calculation of the synchronization offset that is added to the timestamp millisecond part.

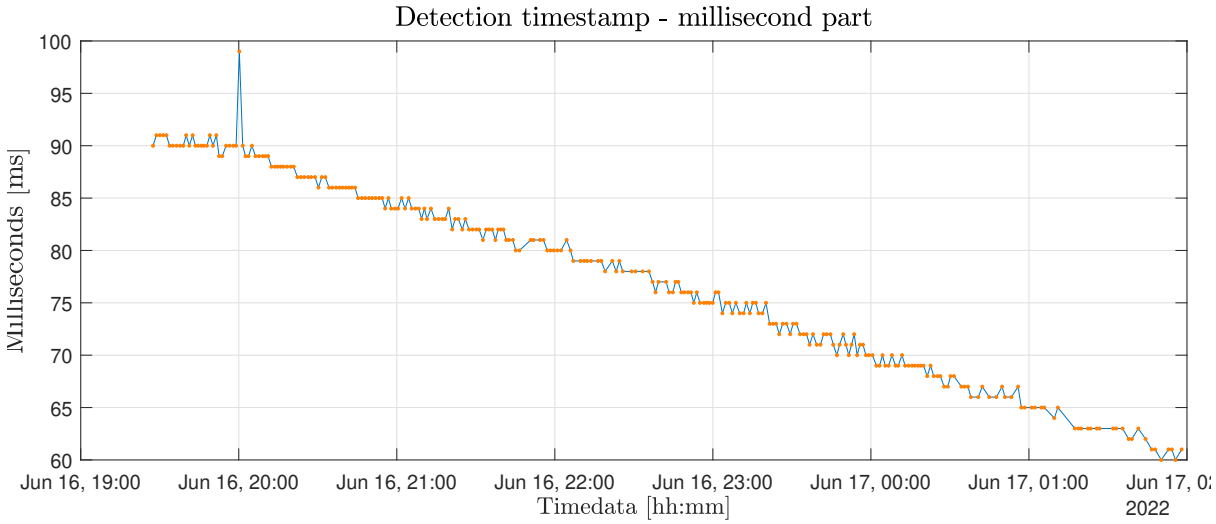


Figure 10.2: Lab test data: Full system test - millisecond part of timestamp. showing an expected constant drift. Figure: Vetle B. Abrahamsen/NTNU 2022

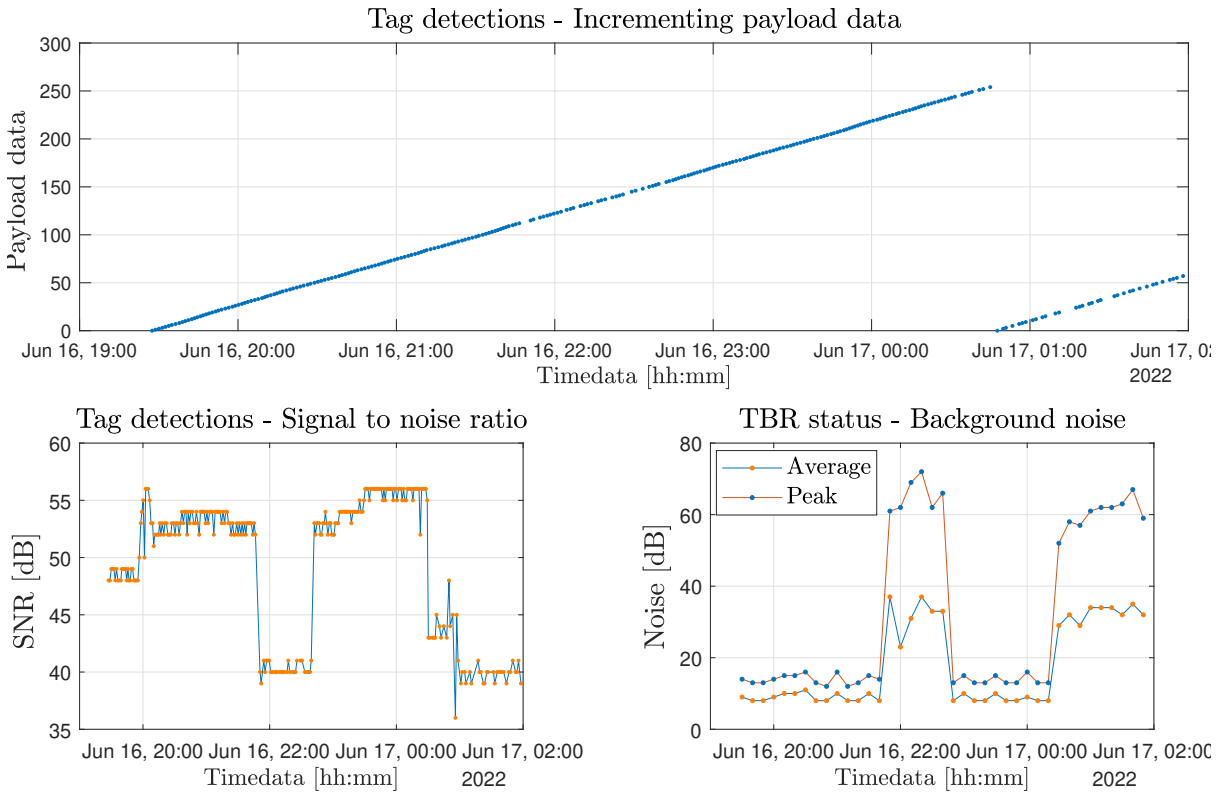


Figure 10.3: Shows all captured tag detections, 37/312 detections were lost due to TBR background noise. Figure: Vetle B. Abrahamsen/NTNU 2022

10.2 Time synchronization

10.2.1 Local time

Measured sleep time and drift in local time have been captured during a period of 10 days. Due to an unknown bug, the cSLIM-v1 reboots every night when the local time is 00:00. Whenever the system restarts, the estimated frequency resets to 1Hz before continuously correcting RTC frequency, during this period the sleep time starts at 2 minutes and increases once the estimated frequency improves the drift in local time. Therefore, the reboot with a 24-hour period contributes to focusing a lot of the captured data points below 10 minutes of sleep time.

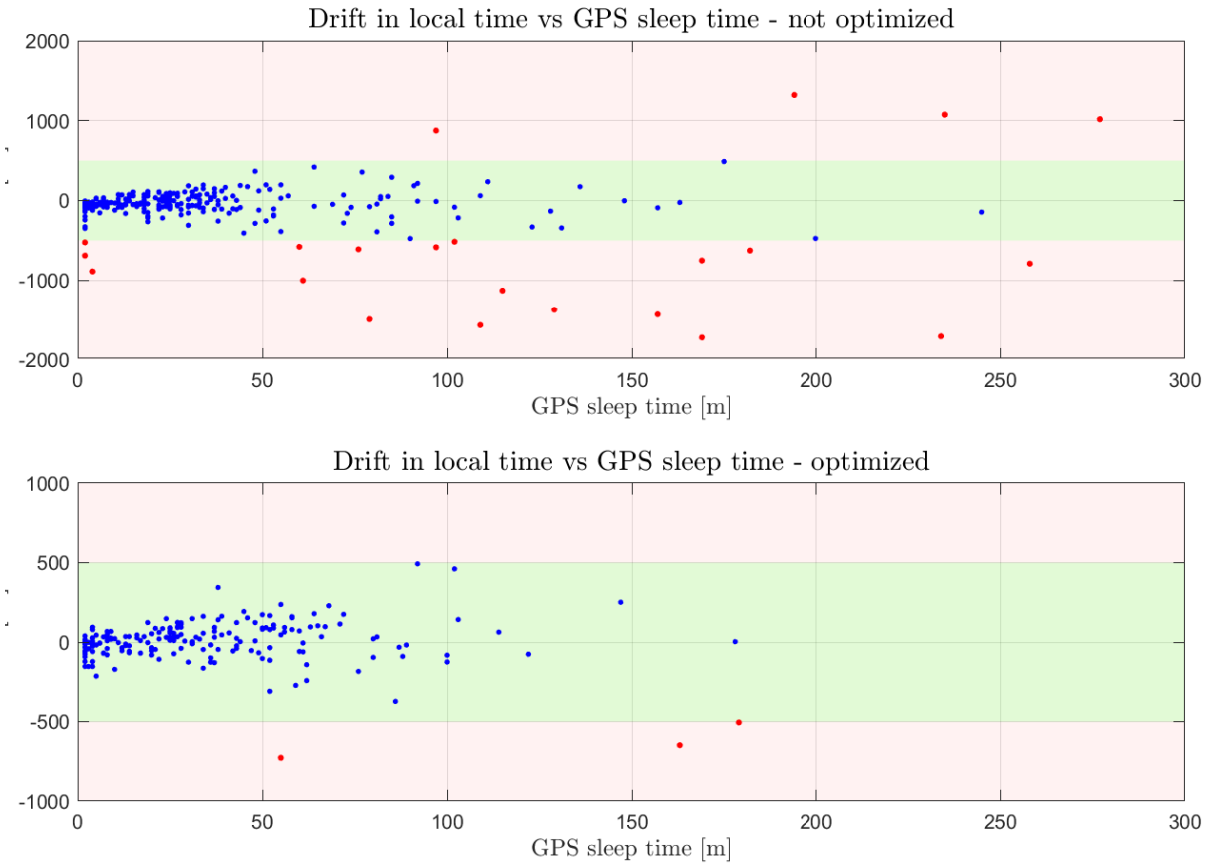


Figure 10.4: Drift in local time for different GPS sleep cycles. Data have been captured during 10 days of testing using the original frequency estimation and sleep time calculation in the top subplot. For the lower subplot, data have been captured during a period of 7 days with optimized frequency estimation and sleep cycle calculations. Outliers above 2ms drift have been excluded from the top subplot while the lower did not have any outliers (except some points with 1s drift). Figure: Vetle B. Abrahamsen/NTNU 2022

As seen in figure 10.4 a significant amount of time drift violations have been observed in the top subplot without tuned filter parameters. The frequency estimator used for the original solution implemented a low-pass filter not suited to the temperature changes in outside environments. The new tuned filter is designed to be more responsive and estimate the frequency closer to the newly measured average frequency during the previous GPS cycle. Additional logic for manipulating the minutes to the next wake-up have resulted in a more stable solution. Only three violations were detected during the week of testing.

Limiting the minutes to the next GPS wake-up have improved the time drift and, by that, securing synchronization of tag detections. By comparing the two subplots in Figure 10.4 it is clear that there is a trade-off between GPS sleep time and local time drift.

10.2.2 Frequency estimation

The tuned frequency estimator are now more responsive to temperature changes. Figure 10.5 illustrates the correlation between RTC temperature and RTC frequency. Increase in temperature results in a decrease in RTC frequency corrections.

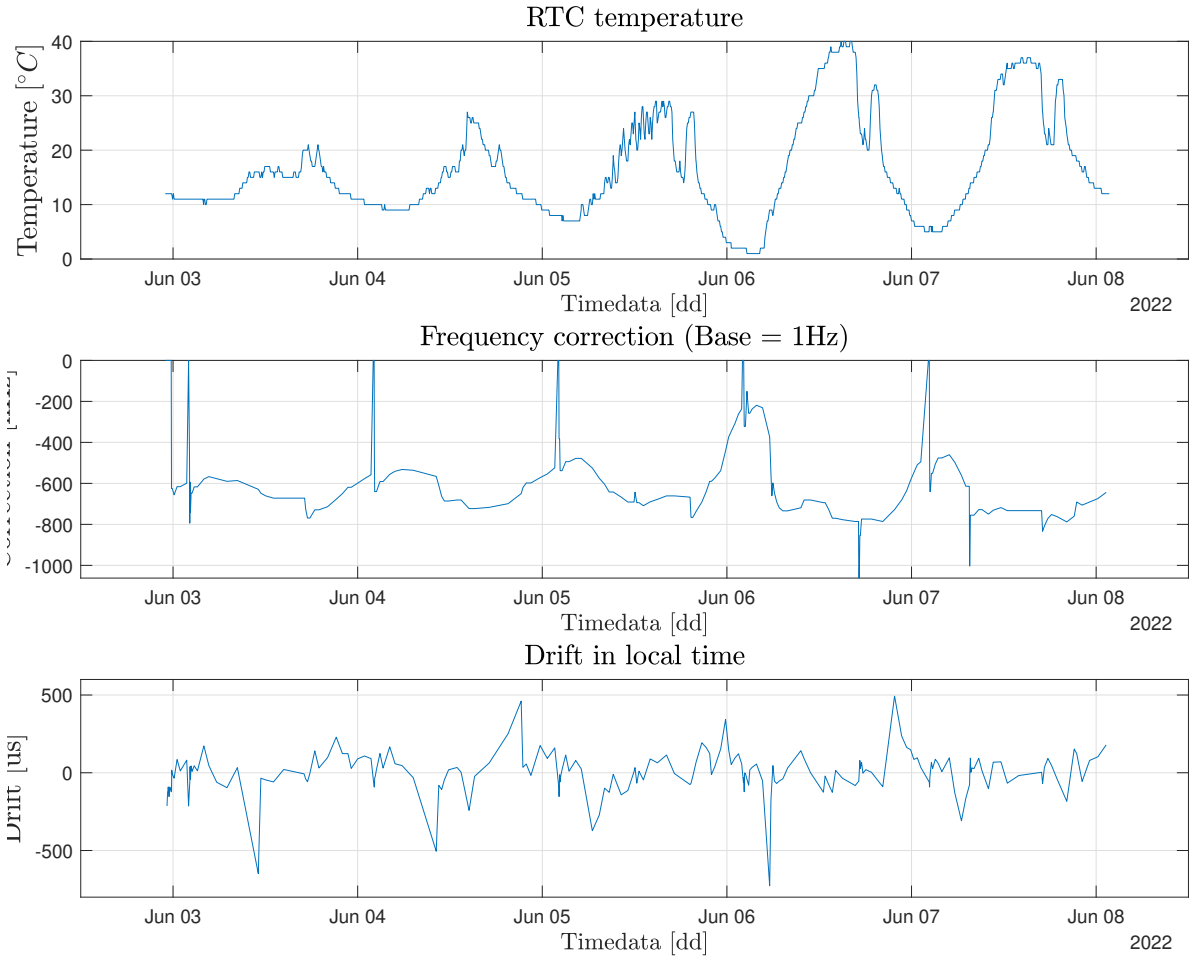


Figure 10.5: Shows the time-series plot of the frequency estimation. RTC temperature have been included and shows the effect temperature change have on the RTC frequency. Figure: Vetle B. Abrahamsen/NTNU 2022

Using Matlabs polyfit function the coefficients to a second-order polynomial, shown in Figure 10.6, have been calculated based on the data points for RTC temperature and RTC frequency correction. The polynomial function used to approximate the coefficients can be seen in Equation 10.1, and the coefficients are given in Table 10.1. The polynomial have been calculated in order to illustrate that a function describing the frequency with respect to temperature is possible and could be used as a starting point for frequency estimation to give an example. However, even the curve will change over time due to aging and could benefit from gathering more and more data to update a dynamic model of the frequency.

$$f_{correction}(T) = p1T^2 + p2T + p3 \tag{10.1}$$

p1	0.5735
p2	-38.2160
p3	-137.7611

Table 10.1: Polynomial coefficients from curve fitting.

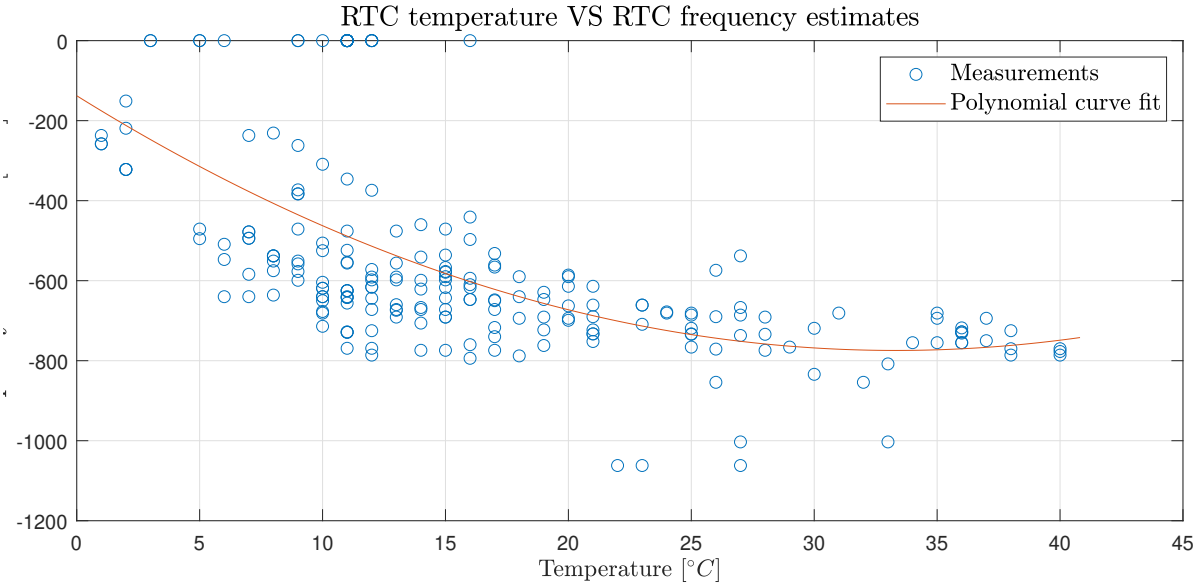


Figure 10.6: Frequency correction as a function of RTC temperature. The data points have been curve fitted. Figure: Vetle B. Abrahamsen/NTNU 2022

10.3 Energy consumption

For a bouy deployed in a remote fjord or offshore, it is crucial that the buoys current consumption is kept to a minimum. This will allow buoys to be operational for a more extended period of time without the need for personnel to replace batteries. Effectively this will reduce the cost of operating the solution.

Current is categorized as either static or dynamic current consumption. Static current is the constant current drawn from passive components. Dynamic current consumption varies with the application and contributes to the main current consumption in the system. The nRF9160's embedded modem for transmitting IoF messages and the GNSS module used to poll navigation data are the two top contributors to the dynamic current and average current consumption. Hence, optimizing the GNSS sleep cycle have been important for reducing power consumption.

The current have been measured over a period of 5 hours. Microchip's Atmel Power Debugger have been used during the current measurements, due to a bug in microchip's software, the measurements could not be written to file for logging as it would freeze the software. Hence screenshots of the visualization tool were the only option.

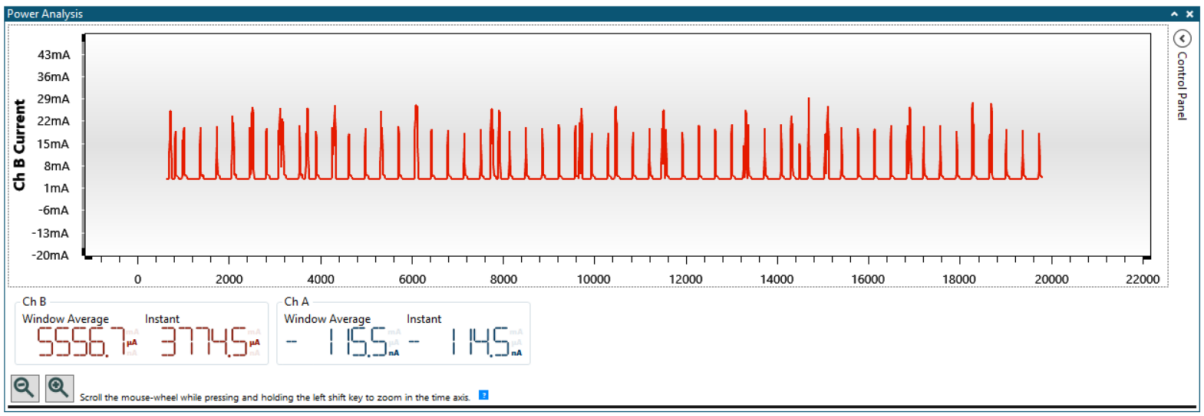


Figure 10.7: Average current measurement during full system test with TBR 700 RT communication and tag detections using a 4.2 volt alkine battery pack. Figure: Vetle B. Abrahamsen/NTNU 2022

Figure 10.7 shows the full measurement, but discards the application's initialization as it is assumed to be negligible in the long run. Due to good GPS cycle results and an NB-IoT transmitting period of 6 minutes, an average current consumption of 5.56mA at 4.2V was achieved. Converting this to power gives a power consumption of 23mW. Using the original battery solution at 3.6V would give an average current consumption of 6.4mA. The original battery solution had a capacity of 35Ah. Assuming an average current consumption of 6.4mA the 35Ah would operate the system for 224 days or approximately 7 months. However, the power solution originally planned is currently not suited for the cSLIM-v1.

In short, the MQTT task waits for new GPS data before transmitting data using NB-IoT. If no new GPS data is polled during a period of 6 minutes, the application will send IoF messages before resetting the timer. Figure 10.8 shows this flow, at 400 seconds the 6-minute timer passed and triggered an NB-IoT transmission, resetting the timer. At 450 seconds, the GPS task woke up to update the GPS data and synchronize the local time, again triggering an NB-IoT message to be sent. A strange wake up of the GPS was discovered at 700 seconds. Using Grafana, it could be verified that the next wakeup was not scheduled until the 1000-second mark in Figure 10.8. Hence it is assumed that the Ublox GNSS module unnecessarily wakes up due to some bug and results in an increased average current consumption. Even lower current consumption than 5.56mA at 4.2V will be possible if this issue is solved.

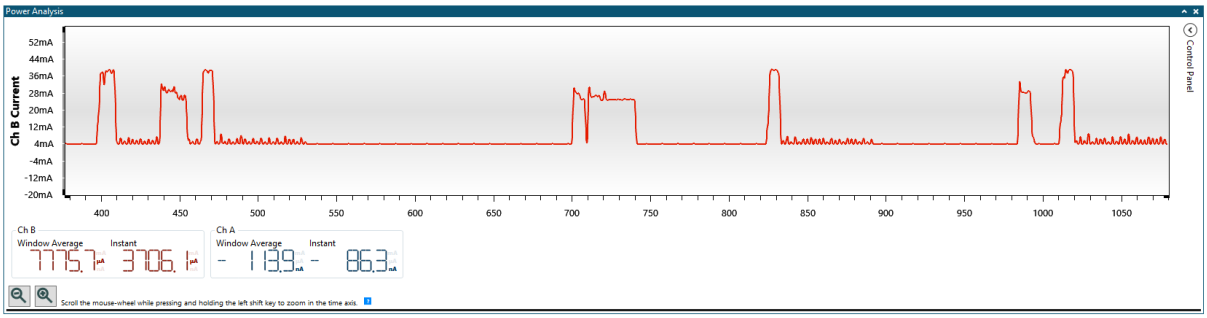


Figure 10.8: Application analysis based on current measurement. Figure shows strange wake-up of GNSS module at 700s. Figure: Vetle B. Abrahamsen/NTNU 2022

As a result of upgrading to the newest nRF9160 SiP revision and implementing the external GPIO expander that resolved the previous hardware layout where some LEDs had to be powered on all the time, the current consumption during idle operation of the cSLIM application have been reduced. A 3.7mA average at 4.2V can be seen in Figure 10.9. This converts to 15mW or 4.3mA at 3.6 volts for comparison to other buoy controllers powered from the original power solution. Small increases in current every tenth second can be seen and is the result of the rs485 bias resistors that start to draw power as the bus is operated for clock synchronization. These currents will be assumed to be negligible. However, for areas with a lot of detection activity, RS485 communication will affect the current consumption slightly.

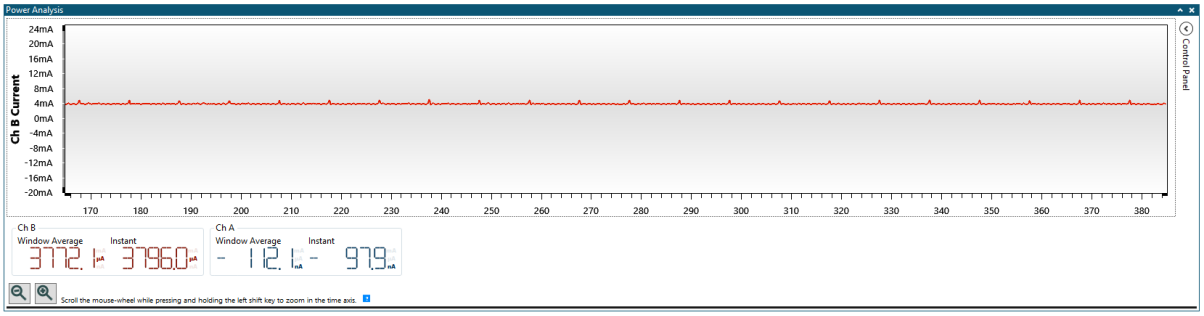


Figure 10.9: Average current measurement during cSLIM application idle and basic TBR synchronization. Figure: Vetle B. Abrahamsen/NTNU 2022

Chapter 11

Discussion

11.1 cSLIM-v1 hardware platform

Design - The cSLIM-v1 is a compact circuit board with a large number of components making the routing a complex task. When designing the circuit, not much thought was put into the pin-out from the nRF9160 relative to the component receiving the respective signal, as the author relied too much on the prototype schematics that utilized the nRF9160 development kit to interface with the nRF9160 SiP. This leads to a large amount of vias due to the need for crossing traces and is a potential source of error from the manufacturer's point of view. A future design revision should consider optimizing the pin-out of the nRF9160 relative to components to reduce the number of vias needed and reduce track lengths.

Hardware selection As the goal have been to realize the prototype as a stand-alone circuit board, components have mainly been reused from the previous design, except for some new components. Component selection in the prototype was also based on the previous SLIM module, hence newer and more power-efficient components might be available today, such as LEDs, LoRa modules and GNSS receivers.

Due to Covid, electronics manufacturers struggle to produce enough components to meet the demand, leading to a global component shortage. Components vital to the cSLIM design are, unfortunately, affected by this shortage, making it hard to produce new units. It is believed that the component shortage will continue to limit the available components. Finding equivalent components to replace unavailable hardware should at least be considered for future revisions if this continues to be a problem for further development. A document listing the number of available components ordered during this thesis and by previous students has been created. It can be found in the CAD file folder for students who continue on the development.

Assembly Three new buoy controllers were assembled during this thesis. Assembling PCBs with this degree of complexity and amount of components is time-consuming and requires precision and extreme care. One of the assembled buoy controllers did not work as expected. As all three buoy controllers were tested with the same software, it is assumed to be an error caused by the assembly. It is suggested to have all components assembled by a manufacturer to save time and reduce the risk of assembly issues as manufacturers often use machines to place components.

During this thesis, a design error was detected on the reset line as described in Section 5.6.1. For future assembles, new PCB's with the updated schematics should be ordered, if not, the modifications shown in Appendix **??** have to be implemented for a reliable reset line. The modification was able to fix the reset line with no functional issues and should cause no problems for further development.

11.1.1 Verification of hardware

Verification of the hardware solution have been thoroughly tested and discussed in the specialization project report. However, due to some equipment being out of reach; the acoustic receiver and a LoRa gateway, these are not covered in the specialization project report and will be discussed here..

LoRa Module The LoRa module have not been used during this Thesis. It was decided to focus on the realization of the full system using NB-IoT to provide a proof of concept, as NB-IoT was the central new technology that differed from previous buoy controllers. A LoRa gateway have not been available during this thesis, but the LoRa module have been tested up to the point of connecting to a gateway, showing no signs of faults. However, a full verification must be completed to verify LoRa transmissions.

nRF9160 embedded GNSS receiver Due to unexpected issues with the power solution and problems with the acoustic receiver interface, the verification of the GPS and Time tasks using the embedded GNSS receiver have not been prioritized. Instead, it was decided to focus on improving the acoustic receiver interface before implementing new experimental functionality to the solution. However, the GNSS receiver and LNA filter have been tested using a test driver that verifies that the nRF9160 modem is able to poll data from satellites. If future tests show no sign of reduced timing precision and current consumption is acceptable, removing the existing GNSS module solution would reduce the production cost by approximately 300 NOK and reduce the complexity of the design. It is also worth repeating here that a new revision of the SKY GNSS LNA will become available on the market, reducing the current consumption of its predecessor by 65%.

Power solution The nRF9160 has vast potential as it embeds both a cellular modem and a GNSS receiver in addition to supporting Zephyr RTOS. However, issues regarding the cellular modem where bursts of current spikes over longer periods result in voltage drops on the initially planned power solution, drastically reducing the system's robustness. With both the GPS and modem operating simultaneously, current loads became too much for the battery and resulted in brownout and reboots of the buoy controller. Software patches were developed to limit the total current, and a supercapacitor was added to aid the battery through the most extreme current spikes. However, the battery has been concluded as unsuitable for the current solution. With non-optimal environmental conditions for cellular communication during the field test, rebooting started to occur. Hence, a temporary battery solution 3x1.5V alkaline batteries in series was made and the rebooting problem caused by brownout was removed. The alkaline battery pack was able to provide the current required by the buoy controller without the need of a supercapacitor. If possible, reducing the transmission output power might reduce the current pulses. However, it will also reduce the performance in low reception areas meaning that the total robustness is not improved. During the LPWAN survey, a paper discussing the same issue was found Paiva et al. (2020) where high current pulses caused the battery solution to fail. In addition to the NB-IoT information gathered in the survey, the conclusion of the power issue is very likely to be correct. Thus, investigating an optional power solution with better pulse current capabilities is suggested.

Ublox GNSS receiver unwanted wake up In his master thesis, Rundhovde Rundhovde (2020) describes a problem with the Ublox NEO-M8N where activity on the MOSI line would cause the GNSS module to wake up, causing an increase current consumption. In an attempt to overcome this, E. Jølsgåd added a logic level converter to the GPS design to only forward the MOSI line to the module when the chip select signal is enabled. During current measurement tests, random wake-ups of the GPS module were still observed, suggesting that some issues still cause the GNSS module to power unexpectedly. However, no undefined signals that could cause undesired cross-talk routed close to the GNSS module have been identified in the hardware platform. As suggested by E.Jølsgard Jølsgård (2021) the Ublox NEO-M9N, compatible with the current design, should be considered when it becomes available on the marked as issues with the module itself might cause this issue.

11.2 Acoustic receiver solution

Unfortunately, an acoustic receiver was stolen during the first overnight test in addition to the only assembled cSLIM-v1 unit. Thelma Biotel's TBLive was instead acquired as the cSLIM-v1 have been designed to support this model. By connecting an external voltage regulator to the available 3-pin connector on cSLIM-v1, TBLive can be powered from the buoy controller's power source as TBLive does not include an internal battery. TBLive also supports a pulse per second (PPS) signal to synchronize the internal clock. Doing so removes the need for sending synchronization commands every 10th. second, hence reducing the chance of collisions on the serial interface. PPS support have been designed on the buoy controller but have not been implemented in the software and currently uses commands to synchronize the local clock. The biggest functional difference between TBR700 and TBLive is that TBLive does not support local logging. However, the cSLIM-v1 supports FRAM and a micro SD card for onboard storage and can be used for local logging. However, it is worth noting that the system must be fault-tolerant for the buoy controller's local logging to be robust, as no detections or messages would be stored during downtime.

Every 10 minute, both receiving a TBR status message and synchronizing the TBR internal clock coincide. For the TBLive, this is no problem as Thelma Biotel have designed its serial interface to allow for undisturbed clock synchronization and postpones potential tag detections or status messages to be sent after the reserved time window. In listening mode, the acoustic receiver is silent on the bus for 1 second. However, it is believed that the TBR 700 does not implement the 1 second listening mode in its serial interface, resulting in problems for the rs485 driver to detect the message. No detailed data sheet for the TBR-700-RT acoustic receiver's serial interface have been found, but in his master thesis from 2016 regarding real-time fish tracking with the TBR 700, Efteland Efteland (2016) explained that the listening mode window was only 1ms. Hence the tbr and rs485 driver had to be modified to avoid loss of tbr status messages.

In all, the TBLive is a good candidate for replacing the TBR700 and could reduce the driver complexity for detecting acoustic messages due to the missing 1 second listening mode in TBR700. An increase in current consumption will be expected as the buoy controllers' power source will power TBLive. TBLive datasheet ThelmaBiotel (n.d.a) states a current consumption of 9mA at 5V when processing data. However, the receiver have been optimized for low-power applications and only wakes up momentarily to process the data, giving an efficient solution.

11.3 Time synchronization

The RTC used, RV-3032-C7, states a drift of ±2.5 parts per million (ppm), which translates to 0.216 seconds per day. With a 500us timing constraint, this limit can be violated in only 3.33 minutes. This will also change depending on the ambient temperature. Using frequency estimation, Jølsgard Jølsgård (2021) was able to sleep for 10 hours with only 300us drift when testing indoors, where the temperature is assumed to be constant. This gives a drift of 0.0083 ppm. When testing cSLIM-v1 outdoors in direct sunlight, temperatures up to 42 degrees Celsius have been measured. However, even shadows or clouds blocking the direct heat from the sun cause the temperature to converge towards the ambient air temperature. Hence large temperature changes are expected, changing the RTC frequency and timekeeping precision.

The low-pass filter in the frequency estimation reduces the risk of one single measurement throwing the frequency estimation off course. However, given that the "frequency since last wake up" is in fact an average measurement based on the previous sleep cycle, some low-pass filtering naturally enters the estimation. Hence, the parameters have been tuned to be more aggressive by relying more on the current "frequency since last wake up" as the previous filter would be more suited for indoor environments, as Jølsgard also mentioned in his thesis Jølsgård (2021).

The algorithm for calculating the sleep time of the GNSS module have been constrained only to allow an increase/decrease in sleep time by a factor of two to avoid increasing from a sleep time of 20 minutes to 10 hours in one measurement. On average, the GNSS sleeps for one hour with an average below 200us, giving 0.056 ppm drift. Sleep time of 3 hours with a drift of only 3us has been achieved, but given temperature change, the actual drift might have been -200us at 1.5 hours, then up 203us the next 1.5 hours, giving an average of 3us drift. This is why temperature is checked every 6th minute, and if the increase/decrease is above a predefined limit, then the sleep cycle will be interrupted. The temperature threshold should be reduced if a longer sleep duration is desired.

The use of RTC frequency estimation to increase the time between waking the GNSS module has proven to be extremely valuable. It reduces the average current drastically and allows the buoy to extend its deployment time.

11.4 cSLIM GPS cycle message format

Defining a new IoF message to forward GPS sleep cycle status to the cSLIM GPS cycle dashboard have been highly valuable for optimizing the time synchronization of cSLIM buoy controllers. It gives the possibility to compare and analyze different tuning parameters. As the dashboard displays drift in local time during the last cycle, tag detections can be validated with respect to the timing constraint of 500μs as specified in the system requirement cSLIM-FR.5b - Backend timing error awareness.

The cSLIM GPS cycle message format was defined to monitor the performance during field tests and the size of the different compressed data is not fully optimized. For example, the estimated frequency in nano Hertz is sent as 32bit. However, in Grafana, a math operation subtracts the estimated frequency with 1Hz in order to display only the frequency correction, giving 1Hz as base. Hence 16 bits will, under normal circumstances, be enough to display up to ±32768 nHz frequency corrections, a reduction of 16 bits that saves energy when transmitting the messages. Some margins should be added to allow larger frequency corrections to be displayed to detect possible outliers. The drift in us is also sent as 32-bit, making it possible to display drifts up to ±2147 seconds or 35 minutes, which is not very likely. The GPS cycle message format should be optimized in future development.

11.5 Field and lab test

Testing the buoy controller in realistic environments have provided valuable information on using cellular LPWAN technology. The use of NB-IoT to transmit acoustic telemetry and buy status information was successful and the cSLIM-v1 as a buoy controller showed promising results in capturing data. However, the field test discovered weaknesses in the power solution. The implemented supercapacitor seemed to help when testing in the lab, but when deployed, the system started to experience trouble with powering the solution due to high current pulses. Hence, the field test have been extremely important both to prove the concept and detect the system's weaknesses.

11.5.1 Discovered issues

During the field test, it was observed that the buoy controller was rebooting every midnight at 00:00 local time. In Jølsgard's thesis, a similar bug was described where the system would reboot with a period up to 22 hours. These two bugs are most likely the same, but due to Grafana, the bug have bee more precisely timestamped. Furthermore, the bug do not depend on when the system was booted the first time, suggesting that the root cause could be the jump from 23:59 to 00:00. Keep in mind that this is only a suggestion and could be used as a starting point for further investigation. The issue can also be related to outdated drivers in the nRF Connect SDK V1.5.1.

A more critical problem was that the buoy controller could not successfully initialize and froze at 00:02. In an attempt to get more information about the cause of the issue, a second buoy controller was placed on a rooftop with the same power solution and software as the field test unit but without the acoustic receiver. After 15 days of continuous operation, with the midnight-bug present, the buoy controller experienced the same bug that froze the unit at 00:02 after a midnight-bug reboot. However, after physically restarting the device, the test unit was operational without the bug re-appearing for another 10 days before the device was stopped for other tests.

11.5.2 Current measurements

Unfortunately, the current measurements have been minimal and only cover average consumption when running the complete cSLIM application. This was due to the cSLIM hardware platform failing to run applications where different components were disabled to test consumption from different modules. Even when flashing

older software that has been adequately verified, the buoy controller would not correctly function, making it impossible to acquire meaningful current measurements.

11.6 Goal and method

This thesis aims to realize the stand-alone embedded buoy controller based on the prototype, with particular emphasis on developing and optimizing the acoustic receiver interface, time synchronization, NB-IoT, and LoRa links. In addition, assembling a suitable amount of cSLIM-v1 was desired to conduct a realistic field test to explore the solution's function and performance and make a backend server for storage and visualization of buoy data.

The hardware realization have been based on the previous prototype and was designed during the specialization project. Since Jølsgard also based his prototype design on the previous SLIM module, new hardware might be available today, but it was desired to keep the design with the tested and supported hardware. New drivers have been made during this thesis in order to adapt the new functionality, such as the external GPIO bus and the ability to shut off the LoRa module completely. Having the previous prototype software have been essential in order to get to where the cSLIM-v1 is today. The software package is complex and includes many drivers and tasks that interact with each other. Getting familiar with the software have been more time-consuming than first expected.

Unforeseen issues with the power solution caused a drawback in the progress of optimizing the functionality. Powering the buoy controller with a battery is vital for the concept concerning robustness and was thus prioritized over the main tasks of the thesis. After finding a possible power solution fix, the available time before the planned field test forced the author to focus only on one of the LPWAN technologies - NB-IoT as this was not tested in the field previously. Other focus areas were optimizing and testing the acoustic receiver interface, bug fixing, and making the backend server for data storage and visualization. Jon Andreas Kornberg has all the credit for making the dashboards and Influxdb standard that is common for both cSLIM-v1 and SLIM. However, a dashboard and InfluxDB standard unique to the cSLIM-v1 for monitoring the GPS cycle status was made by the author. Without Kornberg taking the lead on Grafana and InfluxDB, the field test would in worst case be canceled, causing drawbacks for the development of cSLIM-v1.

As a result of re-prioritizing, no further development or tests of the LoRa links have been done during this thesis. Making it hard to compare the two LPWAN technologies. Re-writing the GNSS task in order to utilize the nRF9160 embedded GNSS receiver have not been prioritized as initially planned but have instead been verified by a simple test script for polling satellite data.

11.7 System requirements

As cSLIM-v1 have been based on the cSLIM-Shield prototype, all functional requirements have been inherited by adopting the shield hardware design. This section will cover some of the requirements that have been further optimized or require future work to fulfill. The complete list of system requirements can be found in Appendix A.3.

UAR-Functional Requirements During the thesis **UARUAR-FR.1a - Underwater acoustic detections** have been further optimized for robustness and high reliability for detecting signals. Extensive testing shows that detections are captured and forwarded to the backend application (Grafana). **UAR-FR.1b - Time synchronization** Have been fulfilled by optimizing the frequency estimation tuning parameters for outdoor environments where the temperature is not constant. However, synchronization have not been verified between two or more buoy controllers and should be prioritized for future development. **UAR-FR.1d - cSLIM communication** communication between cSLIM and the underwater acoustic receiver is achieved with half-duplex rs485 that supports bi-directional communication for time synchronization and data transmissions.

cSLIM requirements Requirements regarding **cSLIM-FR.4 - Position awareness** have mostly been fulfilled. Metadata such as position dilution of precision (pdop) is not correctly decoded in the compressed status message

and have not been prioritized during this thesis. cSLIM is able to acquire its position using the Ublox GNSS module. Requirement **cSLIM-FR.5 - Timing accuracy** have been improved, more specifically, the maximum timing error with almost no violations of the timing criteria of $\pm 500\mu$s. **cSLIM-FR.5b Backend timing error awareness** have been implemented by defining the new cSLIM GPS cycle IoF message format and is forwarded to Grafana after each GPS update. The requirement of storing data offline for persistent storage in requirement **cSLIM-FR.6** have not been fulfilled. However, cSLIM offers offline storage both with FRAM or uSD and the FRAM driver have been bug-fixed and verified. By using Zephyr RTOS to handle fault detection and handling and the modified drivers to detect and handle unwanted behavior. e.g. re-connecting to the MQTT broker in case of network loss or avoiding collisions on the acoustic receiver interface have greatly improved the system's fault tolerance. However, the requirement **cSLIM-FR.9 - Fault tolerance** should also be prioritized in further development as bugs have been observed without being able to find the root cause and is not considered fulfilled. The non-functional requirement **cSLIM-NRF.1 - Battery life** is also improved. The stand-alone cSLIM-v1 improves power consumption but lacks extensive current measurements. The battery solution originally planned have also proven to be unsuited for cSLIM-v1. Hence this requirement will be considered not fulfilled.

cSLIM-v1 - stand-alone requirements All peripherals and connectors from the cSLIM shield prototype have been implemented on the stand-alone module cSLIM-v1. New connectors and hardware solutions have been added to the design, such as GPS antennas, and cellular antenna in addition to antenna matching networks that fulfills the **cSLIM_SA-FR.3** and **cSLIM_SA-FR.4** requirement. **cSLIM_SA-FR.5** have during this thesis been fulfilled as the design and bill of material have been updated with the nRF9160 SiP revision 2 that supports the newest modem firmware to allow PPS support for the embedded GNSS receiver.

Chapter 12

Conclusions

During the LPWAN survey, no papers except the previous work on the Internet of Fish project was found explicitly covering the use of LPWAN to transmit acoustic telemetry data in real-time. This does not imply that research and paper do not exist on the topic but suggests that NTNU and the Internet of Fish project are amongst the lead, if not in the lead, of developing a solution for real-time fish monitoring using acoustic telemetry and LPWAN buoys. However, similar concepts have been published where offshore stations gather sensory data and transmit it to ashore base stations using LPWAN technology. Most papers found covering the specific use case of marine monitoring have been implemented using LoRaWAN. The comparisons suggest that LoRaWAN is the most power-efficient solution, but with lower quality of service. The costs of employing an LPWAN solution are also suggested to be lower for LoRaWAN, however, the costs for the extra infrastructure needed, e.g., multiple LoRa gateways, should be taken into consideration. NB-IoT, as a result of using a synchronous LTE communication protocol, naturally uses more current. Moreover, the extra current pulses due to the FDMA/OFDMA access modes during uplink and downlink transmissions require special attention to the power solution's capability to handle current pulses. This issue was discovered to be a problem for the initially planned power source. However, the NB-IoT's practicality is hard to come close, using existing infrastructure supporting payloads up to 1600 bytes with higher data rates, quality of service, and suitable sleep modes make NB-IoT a good candidate for marine monitoring, given offshore coverage.

The cSLIM-v1 embedded bouy controller have been realized. Newly assembled units have been made and software have been further developed and optimized. Results show that the system using NB-IoT is a particularly convenient and practical solution for deployment in remote fjords. The biggest achievement is that the system have been tested in the field, giving a proof of concept as the deployed buoy was able to detect fish, track and visualize fish telemetry in real-time with a backend server for visualization that has proven to be remarkably useful. However, work remains before the solution could be considered a minimum viable product for early "customers".

Using the precise, ultra-low-power, and temperature compensated RTC outdoors have given good results after tuning the frequency estimation filter. The results show that the buoy controller can keep the local time synchronized within the time constraint while increasing the time between the need for GPS update, which greatly reduces the average current consumption. Modifications were done to the component selection and circuit design, especially the newer nRF9160 SiP and GPIO expander for controlling LEDs separately have further reduced the current consumption.

The acoustic receiver serial interface, both for the TBLive and TBR700, shows robustness and reliability as all signals from the acoustic receiver are detected by the buoy controller.

The backend environment for visualizing data in real-time have been a big step in the right direction towards monitoring fish in real-time for both cSLIM and SLIM buoy controllers. The dashboards visualizing fish telemetry are simple and provide the user with all available data. During this thesis, the GPS cycle dashboard have proven to be especially useful for analyzing and optimizing the time synchronization performance for cSLIM-v1 buoy controllers.

Chapter 13

Further work

Future work should be based on both this list found in this chapter and suggestions given in the master thesis written by Jølsgård (2021). Suggestions found in Jølsgard's thesis list some ueful approaches for simplifying the time task and improving NB-IoT and LoRa module software.s

13.1 Power solution

A new battery solution will be required for a robust operation of cSLIM using NB-IoT. As the NB-IoT solution causes high pulse currents in the nRF9160 SiP modem, a new battery should be able to handle these without causing system reboots. The 3x1.5V alkaline battery pack have proved to work as a temporary solution for testing purposes.

13.2 LoRa verification

A LoRa-driven application should be tested for verification, with a generally lower power consumption than NB-IoT and the implemented RTC frequency estimation that significantly improved the GPS sleep time is assumed to give good average measurements of the cSLIM hardware platform.

13.3 RS232 Verification

No peripheral driver is available for controlling the rs232 interface. However, peripheral sharing is supported by nRF9160 and can be investigated if RS232 becomes vital for the solution, e.g., for logging purposes.

13.4 Persistent storage

In contrast to TBR 700 RT, the TBLive acoustic receiver does not support local logging of detections. Implementing software to enable local logging on the buoy controller using either FRAM or uSD have to be implemented for a future revision. A possible solution would also implement a check on reboots to see if any detections was not transmitted before power-down. As the back-end solution logs all received tag detections, managing "expired" or already transmitted detections should be considered to prevent storage problems for new and detections.

13.5 Evaluation of the nRF9160 embedded GNSS receiver

The nRF9160 embedded GNSS module has been tested and successfully polls satellite navigation and time data. The next step toward comparing the two GNSS solutions will be to adapt the nRF9160 GNSS receiver to the GPS task, making it possible to compare efficiency.

Additionally, as the newest nRF9160 revision fully supports the latest modem firmware where 1PPS support are enabled, testing PPS signals generated by nRF's embedded GNSS receiver to synchronize the acoustic detections should be investigated.

13.6 Optimizing cSLIM GPS cycle message format

Re-defining the cSLIM GPS cycle message format to save transmission energy by optimizing the message compression will be advantageous.

13.7 Optimizing time task

Drifts of up to 1 second occur often. The root cause of this issue has not been found and can be the result of a faulty RTC or it can be solved by optimizing the time synchronization task. Jølsgård (2021) provides some implementation details of the task and some tips for further work.

13.8 Acoustic receiver synchronization between multiple buoys

Neither the acoustic receiver's synchronization between multiple buoys during this thesis nor previous work has been verified. This should be prioritized.

13.9 Bug fixing

Midnight bug First discovered by Jølsgård (2021) in his thesis, the midnight bug reboots the system every night at 00:00 (device local time). This causes extra current consumption as the RTC frequency estimation is restarted. The root cause have not been found during this thesis despite attempts. Issues with the currently implemented version of nRF Connect SDK V1.5.1 might cause this issue, as imported libraries might include bugs. The libraries are constantly improved with newer SDK versions, which might resolve this issue.

System freeze A more difficult bug have caused the system to freeze after reboots. This bug was first discovered during field test on one unit. After 15 days of another unit's continuous testing, the bug also appeared. Logging and multiple restarts of the system successfully forced the bug to re-appear with error messages "time until next wake-up" being negative. Hence a starting point to find the root cause should be to investigate the time task. As the negative "time until wakeup" kept decrement, the RTC clock-out is working as expected. Flashing older software to the cSLIM was attempted, but the bug kept blocking the application. Hence, hardware is also a likely cause of the problem.

Ublox metadata The GPS metadata, especially pdop does not provide meaningful data. Ublox datasheet states that the pdop value are transmitted on the SPI bus as a scaled value. hence re-scaling the pdop value back could solve the issue but have not been prioritized.

13.10 nRF Connect SDK update

The current application runs on the nRF Connect SDK version 1.5.1 and is behind the current version 1.9.1. Rewriting the cSLIM software to utilize the newest SDK will be a lot of work. However, it would be advantageous to keep up with the latest SDK versions at some point as the newest SDK verisons uses updated and optimized Zephyr RTOS kernel drivers, peripheral drivers, functions, tool-chains and libraries. This could potentially fix bugs that are not caused by the application itself. This should not be a requirement or be prioritized but should be considered for future revisions.

13.11 Extended current consumption measurements

Due to a faulty cSLIM unit at the end of the semester, the current measurements have been minimal as I could only run the full system measurement before having problems running the application at all. Hence further current measurements with a particular focus on comparing LoRa and NB-IoT should be prioritized.

Appendix A

My Appendix

A.1 Getting Started

A.1.1 nRF9160 SiP

Toolchain

1. Download and install the latest **nRF Connect** from https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-desktop/download#infotabs.
2. Open nRF Connect and open **Toolchain Manager**.
3. Install **nRF Connect SDK V1.5.1**.

Flashing cLSIM software A debugger is required for flashing software through SWD. The nRF9160 DK can be as a debugger for cSLIM-v1 by connecting a **SWD-cable** between P19 on nRF9160 DK and J512 on cSLIM-v1. Make sure that both the cSLIM-v1 and debugger are disconnected from their power source before connecting the SWD cable. (PS: Do not use JTAG cable, these are very similar but has the wrong pinout.)

1. Extract the cSLIM_v1 folder from the provided Zip file directly to C://cSLIM_v1
2. In the Toolchain manager window, open the drop-down menu next to nRF Connect SDK v1.5.1 and select **Open command prompt**
3. Inside the command prompt, navigate to the cSLIM_v1 path. This is done to resolve links to the nRF Connect SDK when utilizing its toolchain.
4. Connect the debugger to cSLIM-v1 J512 SWD header, power on both the device and debugger.
5. Compile the software with **west build -b cSLIMns -p**. -p is used to compile the full solution the first time or when changes have been made to any files within *cSLIM_v1\boards\arm\cSLIM*. -p can be omitted if changes only apply to the cSLIM application files within the src folder.
6. flash the software using **west flash**

A.1.2 LoRa module

Toolchain

1. Download and install the latest Microchip Studio from https://www.microchip.com/en-us/tools-resources/develop/microchip-studio

Flashing to LoRa module For flashing software to the LoRa module an Atmel ICE debugger will be required. Connect the Atmel ICE SAM output to the SWD header J508 on cSLIM-v1. verify that the red and green lights on the Atmel ICE is showing when power is connected.

1. Extract the cSLIM_v1_LoRa_parser folder from the provided Zip.
2. Open code in Microchip Studio
3. Make sure that the LoRa module have been enabled by the cSLIM software. e.g. flash a cSLIM application to nRF9160 where all tasks have been disabled except the LoRa task.
4. Compile and flash the program through Microchip Studio.

A.1.3 RTT Viewer

Once flashed, the RTT Viewer can be used to visualize messages in real-time from the application software if logging is enabled in the prj.conf file. This requires the debugger to be connected to J512 SWD header. Each time cSLIM-v1 is restarted, including when flashed, the RTT viewer has to be reconnected. Figure A.1 shows the configuration in order to established a connection over RTT using an external SEGGER. From the drop-down menu from file, press "connect", choose USB, specify nRF9160as target device, select auto detect control block and choose target interface as SWD using 4000kHz.

Note that when logging is enabled in prj.conf, it causes extra current consumption and should only be used when testing. Be sure to disconnect both the cSLIM-v1 and debugger power-source before connecting and disconnecting the SWD cable.

J-Link RTT viewer can be downloaded from:
https://www.segger.com/products/debug-probes/j-link/tools/rtt-viewer/

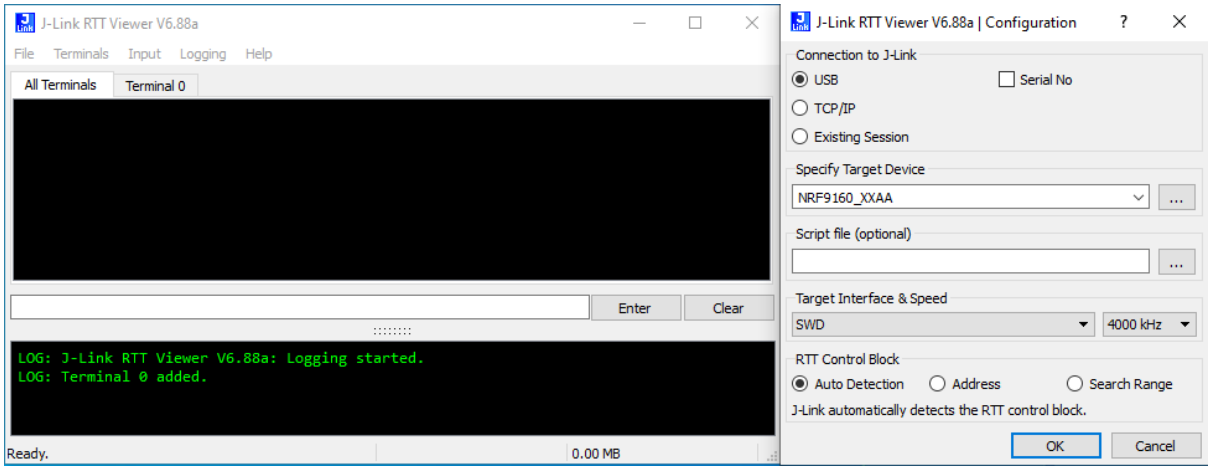


Figure A.1: The J-Link RTT viewer software showing how to connect to a device in order to read logging information in real-time. Figure: Vetle Berg Abrahamsen/NTNU 2022

A.2 cSLIM-v1 Bill of materials

Designator	Value	Quantity	Manufacturer_Part_Number	Description
A1		1	P822601	CDMA, DCS, EDGE, GPRS, GSM, HSDPA, LTE, PCS, UMTS, WCDMA Flat Bar RF Antenna
C301, C401, C402, C403, C404, C405, C601, C604, C606, C806, C810, C901, C904, C905, C907, C1001, C1202, C1205, C1209, C1215	100nF	20	CL10B104KB8NNNL	Capacitor
C501	12pF	1	0603N120J500CT	Capacitor
C502	10nF	1	CL10B103KB8NNNC	Capacitor
C602, C603, C902, C1216, C1217	1uF	5	CL10A105KO8NNNC	Capacitor
C605	10uF	1	CL10X106MP8NRNC	Capacitor
C607	22uF	1	CL10A226KQ8NRNE	Capacitor
C608	1.8pF	1	CL10C1R8BB8NNNC	Capacitor
C609		1	PCH1E101MCL4GS	Alum.Organic Polymer Capacitors 100UF 25V ESR=27
C807	1uF/50V	1	CL10A105KB8NNNC	Capacitor
C903, C906	100pF	2	CL10C101JB8NNNC	Capacitor
C1201, C1208	15pF	2	CL10C150JB81PNC	Capacitor
C1203, C1210	47uF	2	GRM188D70E476ME01D	Capacitor
C1206, C1207	4.7uF	2	CL10A475KQ8NNNC	Capacitor
C1211	3.5pF	1	CL10C3R3CB81PNC	Capacitor
C1212, C1213, C1214	N.C.	3		Capacitor
D601		1	1SS390SMFHT2R	RF Diode PIN - Single 35V 100 mA EMD2
D1201		1	CM1402-03CP	FILTER RC(PI) ESD SMD
FB501	500 mA, 600 Ohms	1	BLM18EG601SZ1D	Ferrite bead
FB801, FB802	60 OHM	2	742792602	Ferrite bead, [NoValue]
FB1201	120ohm 0.7A	1	BLM18AG121SH1D	Ferrite bead - 120ohm 0.7A
IC201		1	WLR089U0-I/RM	LONG RANGE LORA MODULE (863-928
IC202, IC701, IC902		3	SN74AVC4T774PWR	Logic Level Converter (LLC)
IC203, IC702, IC903		3	NX2301P,215	P-Channel 20V 2A (Ta) 400mW (Ta), 2.8W (Tc) Surface Mount TO-236AB
IC301		1	MAX3486ESA+	1/1 Transceiver Half RS422, RS485 8-SOIC
IC302		1	MAX3471EUA+	1/1 Transceiver Half RS485 8-uMAX-EP|8-uSOP-EP
IC401		1	MAX3221CUE+	1/1 Transceiver Full RS232 16-TSSOP
IC601		1	TPS2113PWR	OR Controller Source Selector Switch N-Channel 2:1
IC901		1	NEO-M8N-0-10	NEO-M9N-00B should be used if available
IC904		1	SKY65943-11	GPS LNA MODULE
IC1001		1	CY15B108QN-20LPXC	IC FRAM 8MBIT SPI 20MHZ 8GQFN
IC1201		1	NRF9160-SICA-B1A-R7	IC RF TxRx + MCU Cellular GPS 700MHz ~ 2.2GHz 161-TFLGA Module
IC1202		1	PCAL6408APWJ	I/O Expander 8 I2C, SMBus 400 kHz 16-TSSOP
J301, J302		2	61300211121	CONN HEADER VERT 2POS 2.54MM
J501		1	70543-0001	BAT CONN HEADER VERT 2POS 2.54MM
J502		1	PH1-04-UA	CONN HEADER VERT 4POS 2.54MM
J503		1	61300611121	CONN HEADER VERT 6POS 2.54MM
J504		1	10119313-302TLF	USB - mini B USB 2.0 Receptacle Connector 5 Position Through Hole
J505, J515		2	61300311121	CONN HEADER VERT 3POS 2.54MM
J506		1	104031-0811	CONN MICRO SD CARD PUSH-PULL R/A
J507, J514		2	CONUFL001-SMD-T	CONN UMC JACK STR SMD
J508, J512		2	SHF-105-01-L-D-SM-LC-K-TR	CONN HEADER SMD 10POS 1.27MM
J509, J510		2	BM05B-ZESS-TBT (LF)(SN)	RS232/RS485_CONN HEADER SMD 5POS 1.5MM
J513		1	SF72S006VBAR2500	NANO SIM CARD PUSH-PUSH
J516		1	TSW-103-08-S-D	Auxillary 2x3 conn
J802		1	52746-1071	LCD connector CONN FFC BOTTOM 10POS 0.50MM R/A
L601		1	LPS3015-222MRB	Inductor
L901	27nH	1	LQG18HH27NJ00D	Inductor
L902	9n1	1	LQW18AN9N1D00D	Inductor
L1201	22nH	1	LQW18AN22NJ00D	Inductor
L1202	1.5nH	1	LQG18HN1N5S00D	Inductor
LED501		1	APT2012LSECK/J3-PRV	LED RED CLEAR 2SMD
LED502		1	APT2012LSYCK/J3-PRV	LED YELLOW CLEAR 2SMD
LED503		1	APT2012LZGCK	LED GREEN CLEAR 2SMD
LED504, LED505		2	APT2012LVBC/D	LED BLUE CLEAR 2SMD
PB1201, PB1202, S601		3	TLS A 070J LFS	SWITCH TACTILE SPST-NO 0.05A 12V
Q901		1	SI1040X-T1-GE3	IC PWR SWITCH P-CHAN 1:1 SC89-6
R201, R202, R203, R603, R701, R702, R703, R801, R804, R901, R902, R903, R906, R1102, R1103, R1209, R1210, R1211, R1212, R1213	100k	20	RC0603FR-07100KL	Resistor
R301	120R	1	RT0603DRD07120RL	Resistor
R302, R303	680R	2	RC0603JR-07680RL	Resistor
R501, R502	15R	2	RC0603JR-0715RL	Resistor
R503, R508, R509, R511, R904, R907, R908, R1204, R1205, R1206, R1216, R1217	0R	12	YJP1608-R001	Resistor
R504, R505	820R	2	RC0603FR-07820RL	Resistor
R506, R507, R510	330R	3	RC0603FR-07330RL	Resistor
R601, R609	240k	2	RC0603FR-07240KL	Resistor
R602, R606, R909	1M	3	RC0603FR-131ML	Resistor
R604	390R	1	RC0603JR-07390RL	Resistor
R605, R1214, R1215	1k	3	RC0603FR-071KL	Resistor
R607	100R	1	RC0603FR-07100RL	Resistor
R608	1M2	1	RC0603FR-071M2L	Resistor
R610	3k	1	RC0603FR-073KL	Resistor
R905	10R	1	RC0603FR-1310RL	Resistor
R1001, R1002	33R	2	RC0603FR-0733RL	Resistor
R1003	50R	1	RC0603FR-0749R9L	Resistor
R1101, R1201	10k	2	RC0603FR-0710KL	Resistor
R1202, R1203	4.7k	2	RC0603FR-074K7L	Resistor
S201		1	97C08ST	8-Pin address switch
U501		1	USBLC6-2SC6	17V Clamp 5A (8/20µs) Ipp Tvs Diode SMD
U601		1	TPS63000DRCR	Buck-Boost Switching Regulator IC Positive Adjustable 1.2V 1 Output 1.6A (Switch) 10-VFDFN Exposed Pad
Y1101		1	RV-3032-C7-TA-QC	Real Time Clock Serial I2C 16Bytes 8-Pin SON T/R
Components not specified in the generated BOM				
Display		1	LS013B7DH03	LCD TFT 1.28" 128X128 FPC
Jumper		3	SPC02SXIN-RC	CONN JUMPER SHORTING .100" GOLD
GNSS antenna		1 (2)	FXP611.07.0092C	RF ANT 1.6GHZ FLAT PATCH IPEX. (Internal or external)
cSLIM-v1 PCB + stencil		1	W485774ASF10	Prod.num can be used to repeat order: stock 9 PCB
Required accessories				
SWD cable		1	1528-2009-ND	10-PIN 2X5 SOCKET-SOCKET 1.27MM
USB cable		1	Not specified	USB 2.0 to USB MINI for power.
SEGGER programmer		1	Not specified	SWD prog and RTT view. SEGGER EDU or nRF9160 DK.

A.3 system requirements

All system requirements for the cSLIM-v1 (referred to cSLIM-SA in the system requirements) have been adopted from the cSLIM-shield (referred to as cSLIM in the system requirements) and Jølsgårds thesis. Since cSLIM-v1 is the stand alone realization of the previous prototype, all requirements from cSLIM-shield also applies to the cSLIM-v1, in addition to some extra cSLIM-v1 specific requirements.

A.3.1 Underwater Acoustic Receiver(UAR) requirements

Functional requirements

UAR-FR.1a - Underwater acoustic detections: Detect acoustic signals from acoustic tags with high reliability and range, process the signals and transmit to the cSLIM module.

UAR-FR.1b - Time synchronization: To compute precise tag-locations, the acoustic receivers time must be synchronized with other cSLIM (and preferably also SLIM) modules).

UAR-FR.1c - Underwater sensor data: Detect, process, and transmit sensor data, e.g. water temperature.

UAR-FR.1d - cSLIM Communication: The UAR must have a duplex communication with the cSLIM module for time synchronization and data transmission.

A.3.2 Cellular SLIM (cSLIM) module requirements

Functional requirements

cSLIM-FR.1 - UAR communication: See UAR-FR.1d.

cSLIM-FR.2 - Cellular IoT communication

cSLIM-FR.2a - Power: The c-IoT communication must be power efficient.

cSLIM-FR.2b - Range: Have a long range to support use at a distance from base stations.

cSLIM-FR.2c - Data rate: Use a data rate suitable to operation conditions and data volume.

cSLIM-FR.2d - ITU and local radio regulations: Use radio frequency bands in accordance with regulations.

cSLIM-FR.2e - Heartbeat: Send status messages to the back-end at regular intervals.

cSLIM-FR.3 - LoRa communication

cSLIM-FR.3a - Power: The LoRa communication must be power efficient.

cSLIM-FR.3b - Range: Have a long range to support use at a distance from base stations or other buoys.

cSLIM-FR.3c - Data rate: Use a data rate suitable to operation conditions and data volume.

cSLIM-FR.3d - ITU and local radio regulations: Use radio frequency bands in accordance with regulations.

cSLIM-FR.4 - Position awareness

cSLIM-FR.4a - position acquiring: The cSLIM module must be able to acquire its position based on existing satellite networks.

cSLIM-FR.4b - position transmission: Position and metadata (signal losses, dilution of precision, and number of satellites) must be transmitted to the back-end.1cm

cSLIM-FR.4c - operability on position loss: System must work with the position acquiring turned off and without having an accurate position. This will reduce the usefulness in precise timing operations.

cSLIM-FR.5 - Timing accuracy:

cSLIM-FR.5a - maximum timing error: The timing error should be less than 500us for use cases involving transmitter localization where precise time synchronization is required, for example when using multiple cSLIM modules in a receiver grid and/or mobile receiver platforms (USVs). For stand alone operation the timing error should be less than 500ms.

cSLIM-FR.5b - Back-end timing error awareness: The backend should be made aware of timing errors in the cSLIM modules.

cSLIM-FR.6 - Offine storage

cSLIM-FR.6a - Persistent storage Save logged data offine to persistent storage, e.g a uSD card or F-RAM chip.

cSLIM-FR.7 - Usability

cSLIM-FR.7a: The system should convey its state without the need for debug and serial communication equipment.

cSLIM-FR.7b: Show system status during booting, operation and failure in a way that can be observed at a distance.

cSLIM-FR.7c: Use display to convey relevant information about state and working conditions (radio, network and peripheral connections, status and parameters)

cSLIM-FR.8 - Self-optimization during operation

cSLIM-FR.8a - Choosing optimal parameters: The system should choose optimal parameters without user-interaction to minimize the power usage while maintaining stable radio communication and required timing accuracy.

cSLIM-FR.9 - Fault tolerance

cSLIM-FR.9a - fault detection: The module must be able to detect faults in the different parts of the system and, if possible, forward them to the user.

cSLIM-FR.9b - fault recovery: The module must be able to recover from software errors, for instance by resetting the cSLIM module.

cSLIM-FR.9c - local data logging: The module must be able to log data locally, that is not corrupted on system reset or power loss. See cSLIM-FR.5.

cSLIM-FR.10 - Other hardware requirements

cSLIM-FR.10a - Antennas: The module must have antennas for c-IoT applications LoRa and GNSS, either internal or external.

cSLIM-FR.10b - Connectors: The module should have connectors for external GNSS, LoRa and c-IoT antennas as well as connectors for the acoustic receiver, programming, debugging etc. The module should also be equipped with extra connectors for the ease of future adaptions.

cSLIM-FR.10c - Programming: The module should be programmable via SWD or another suitable programming interface.

cSLIm-FR.10d - Debugging: The module should have a suitable debugging interface.

Non-functional requirements

cSLIM-NFR.1 - Battery life: The module should have a long battery life for unattended installation in remote fjords and waters. Ideally up to seven months, as the acoustic receivers.

cSLIM-NRF.2 - Code quality: Consistent variable and function naming, descriptive comments.

cSLIM-NRF.3 - Code maintainability: Well defined code structure, modulation and function scopes.

A.3.3 nRF9160-DK cSLIM-shield requirements

functional requirements

cSLIM SHIELD-FR.1: The shield must be designed such that it, In association with a nRF9160-DK and cSLIM application source code, satisfies the cSLIM functional requirements.

cSLIM SHIELD-FR.2: The shield must fit on top of the nRF9160-DK.

cSLIM SHIELD-FR.3: The shield must contain peripherals and connectors needed to realise the cSLIM module requirements.

cSLIM SHIELD-FR.3a - The shield must have the following peripherals on board:

- uSD card reader.
- Offline storage.
- Status LEDs (Status green, status red, status gps, status cellular).
- Configuration switches.
- Low power display
- Battery measurement circuit.
- RS485 interface module.
- RS232 interface module.
- Powerdownable LoRa module, i.e. RN2483 or WLR089U0.

cSLIM SHIELD-FR.3b - The shield must have peripheral connectors to:

- Accoustic receiver.
- SPI interface for debugging/extra connectors.
- I2C interface for debugging/extra connectors.
- UART interface for debugging/extra connectors.
- Battery connector.
- RS485 interface debugging.
- RS232 connector.
- LoRa antenna connector.
- USB connector.

A.3.4 cSLIM-v1, stand alone (cSLIM-SA) requirements.

Functional requirements

cSLIM_SA-FR.1: The standalone cSLIM module (cSLIM-SA) must satisfy the cSLIM and cSLIM SHIELD functional requirements.

cSLIM_SA-FR.2: The module should consist of a single PCB with all components and connectors necessary to fulfill the functional requirements in cSLIM_SA-FR.1.

cSLIM_SA-FR.3: The module must have the following peripherals on board (in addition to the cSLIM-SHIELD):

- Voltage regulator.
- Reset circuit and button.
- Antenna matching networks for GNSS, LoRa and cellular antenna.
- Power multiplexer to be powered over USB when available.

cSLIM_SA-FR.4: The module must have the following connectors (in addition to the cSLIM SHIELD):

- External and internal GNSS antenna.
- External cellular antenna (NB-IoT/LTE-M).
- External LoRa antenna.
- Power supply (battery).
- Programming/debugging.

cSLIM_SA-FR.5: The stand alone module must be designed to support the GNSS receiver embedded on the nRF9160 SiP. This includes PPS signal and separate GNSS antenna circuit.

Non-functional requirements

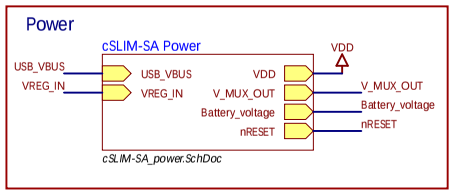
cSLIM SA-NFR.1: The cSLIM module should have a form similar to the current SLIM module to ease replacement in a bouy.

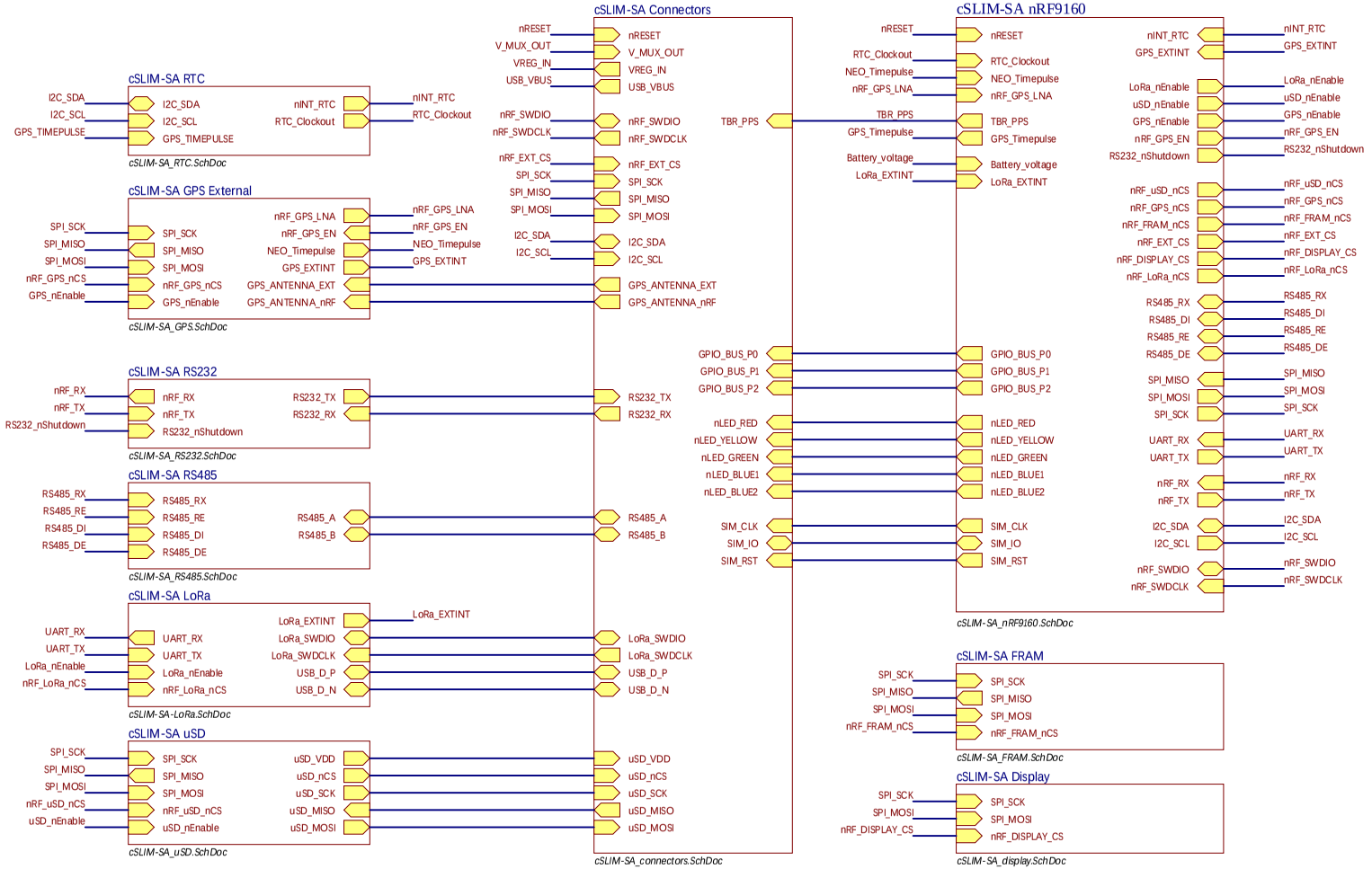
A.4 cSLIM-v1 schematics

cSLIM-v1 (Stand Alone buoy-controller)

Project 2021 NTNU

Vetle Berg Abrahamsen





This symbol is used on components that is normally not mounted on the PCB. However, the components will be found in the bill of meterials. Hence, schematics should be used together with the bill of material when assembling the pcb.

The No ERC object is a design directive. This directive is placed on a node in the circuit to suppress harmless warnings and/or error violation conditions that are detected when the schematic project is compiled.

Based on framework for the cSLIM-shield by Eivind Holen Jølsgard

See Eyvinds Github for schematics and PCB on this link.

https://github.com/eivinhj/cSLIM_nRF9160-DK_shield

NTNU

cSLIM v1

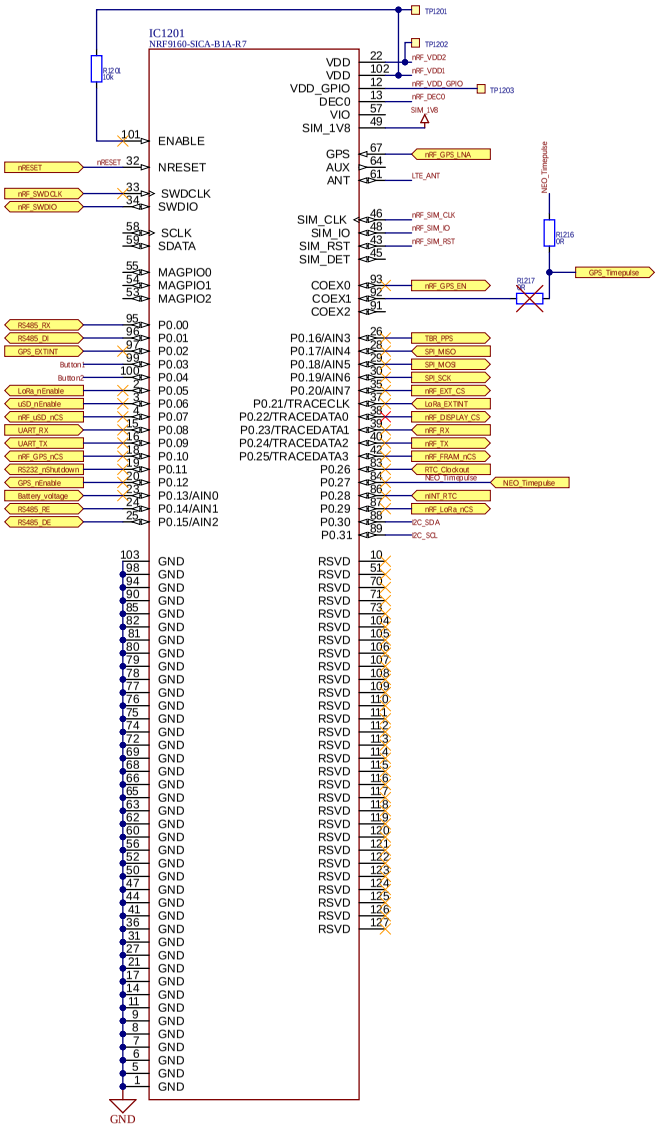
Title: cSLIM v1 - Overview

Size	Number	Revision
A4		v1

Date: 20.06.2022 Sheet 1 of 12

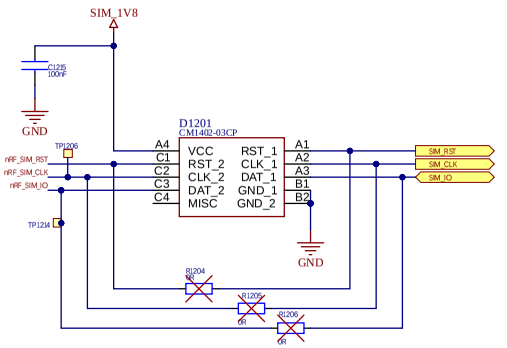
File: cSLIM-SA.SchDoc Drawn by: Vetle Berg Abrahamsen

nRF9160 SiP



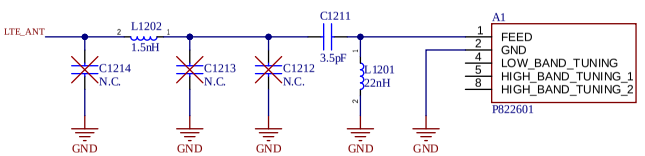
NOTE:
pin 45 (SIM_DET) not used and needs to ble left floating.
External pull-up resitor on nRESET not allowed
When internal GPS is used, COEX1 provides 1PPS, only use either R1216 or R1217

EMI-protection for SIM

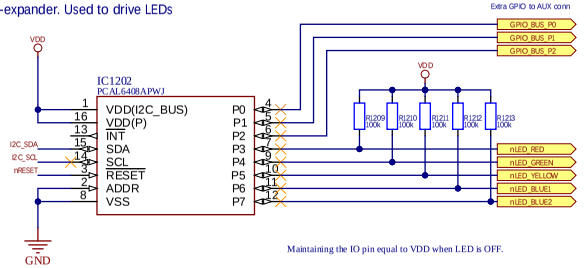


Added buypass in case it dont work.

LTE Antenna matching

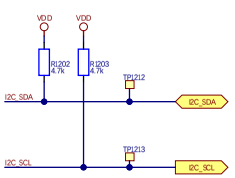


8-bit I2C-bus IO-expander. Used to drive LEDs



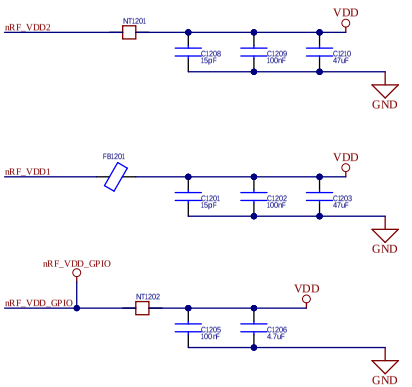
Maintaining the IO pin equal to VDD when LED is OFF.

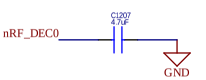
I2C pullup resistors



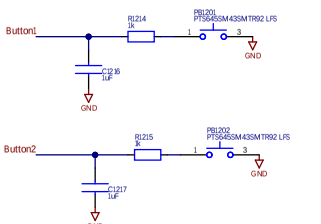
Power to nRF

Using net-tie to connect two different nets without warnings.





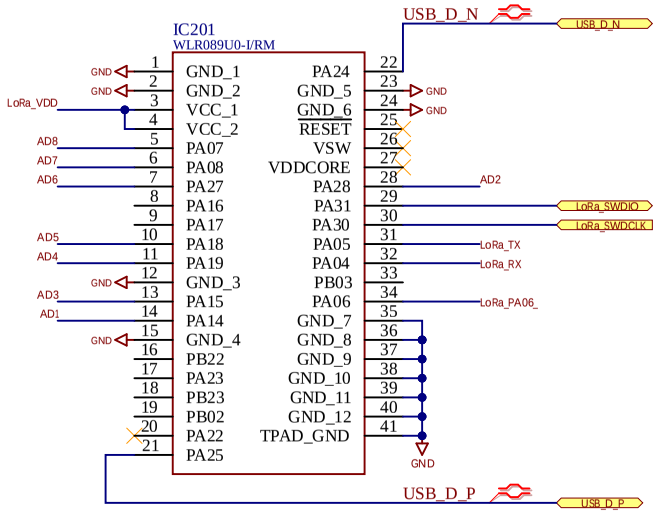
Pushbuttons



NTNU cSLIM v1	Title: cSLIM v1 - nRF9160		
	Size: A3	Number:	Revision: v1
Date: 23.06.2022		Sheet 2 of 12	
File: cSLIM-SA_nRF9160.SchDoc		Drawn by: Vetle Berg Abrahamsen	

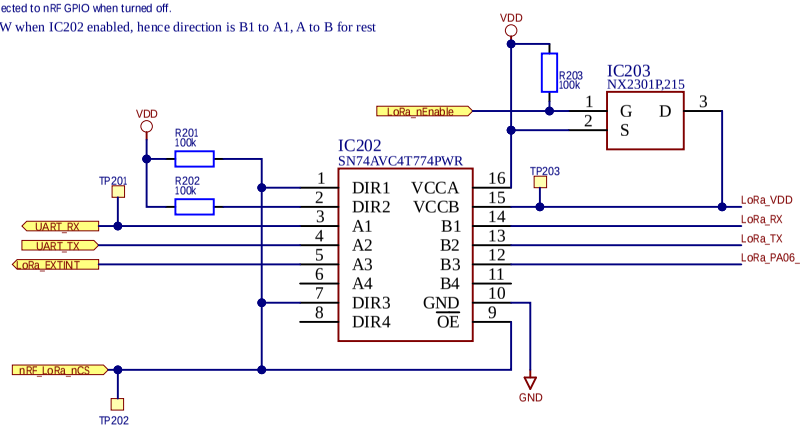
note: LoRa_VDD is left floating when mosfet (IC203) is off. include pulldown on LoRa_VDD

LoRa Module



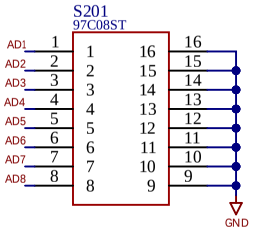
LLC

Logic level converter with controlled
output enable used to ensure tri-state
on pins connected to nRF GPIO when turned off.
DIR1 = LOW when IC202 enabled, hence direction is B1 to A1, A to B for rest



Adress Switch

Functionality must be added to the LoRa software
to make use of Adress switches.



REMOVED interface for LoRa:

I2C_SDA - PA16
I2C_SCL - PA17

LoRa MOSI - PB22
LoRa MISO - PB02
LoRa SCK - PB23
LoRa nCS - PA23

PB03 output removed (used for led)

MARK: LoRa module pins will draw power from nRF GPIO if it is not powered on.
This lead to problems with I2C, SPI and UART comunication with other devices.
Remove unneeded connections and/or insert LLC to ensure tri-state-inputs if
LoRa module should be powered down completely.

NTNU
cSLIM v1

Title: cSLIM v1 - LoRa Module

Size	Number	Revision
A4		v1

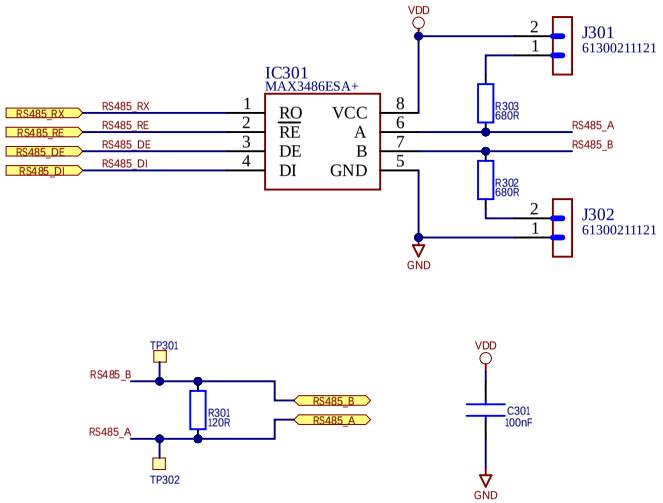
Date: 20.06.2022 | Sheet 3 of 12

File: cSLIM-SA-LoRa.SchDoc | Drawn by: Vetle Berg Abrahamsen

RS485 controller

MAX3486

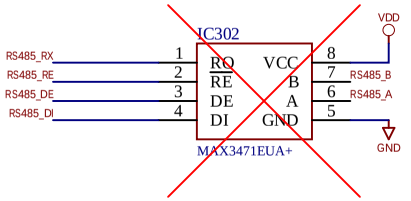
Only mount either MAX3486 or MAX3471,
dependent on baudrate of TBR.
Default baudrate of TBR-700 is 115.200kbps.
MAX3471 is more power efficient, however
it has a maximum baud of 64kbps.



RS485 controller

MAX3471

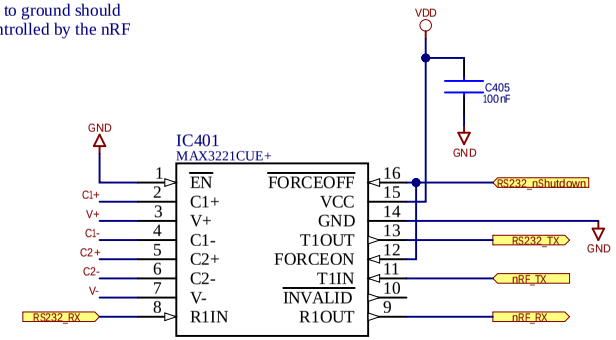
Only mount either MAX3486 or MAX3471,
dependent on baudrate of TBR.
Default baudrate of TBR-700 is 115.200kbps.
MAX3471 is more power efficient, however
it has a maximum baud of 64kbps.

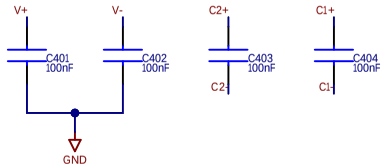


NTNU cSLIM v1	Title: cSLIM v1 - RS485		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022		Sheet 4 of 12	
File: cSLIM-SA_RS485.SchDoc		Drawn by: Vetle Berg Abrahamsen	

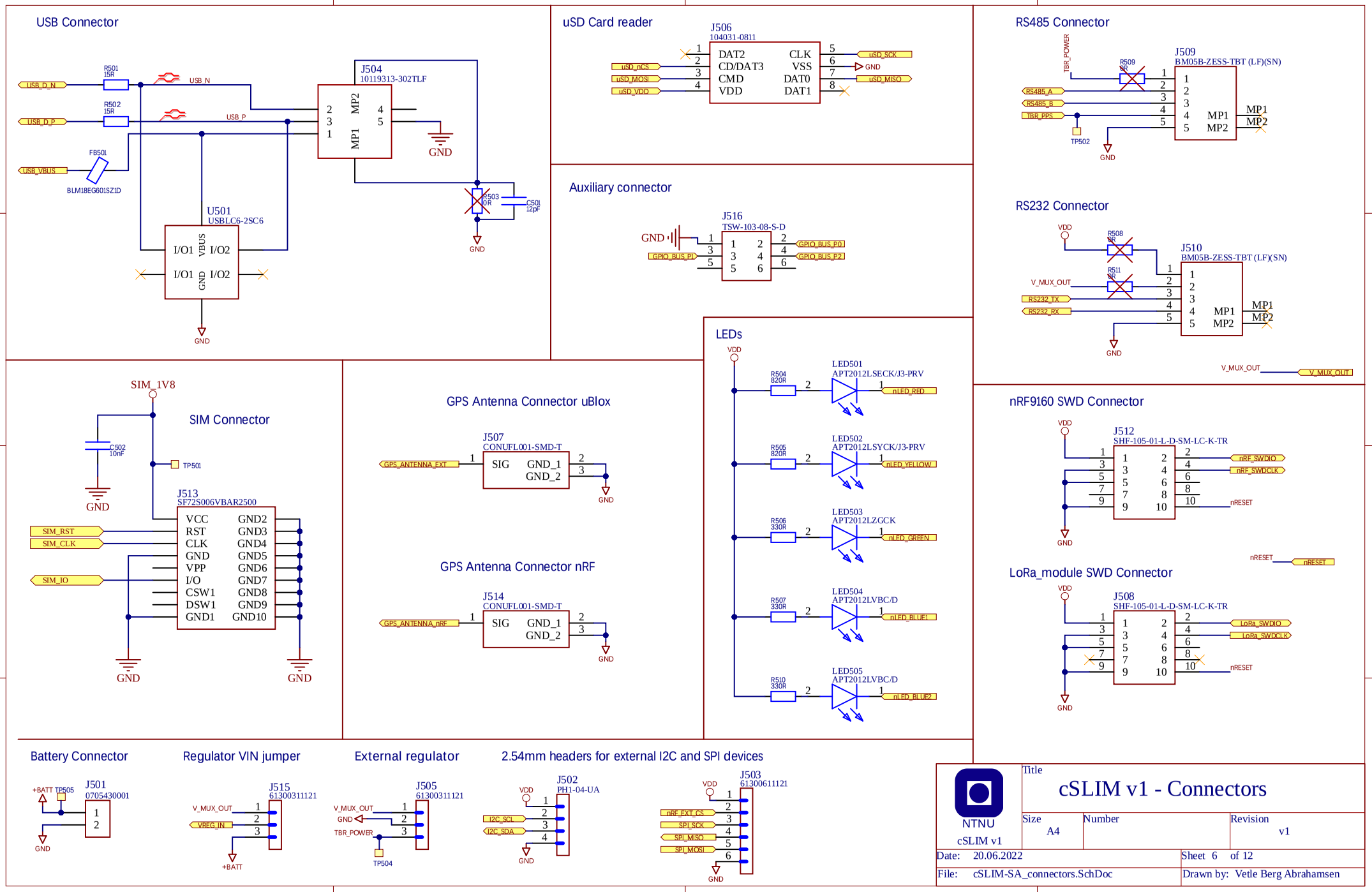
RS232 controller

Note: The nEN pin connected to ground should be considered to be controlled by the nRF for future designs



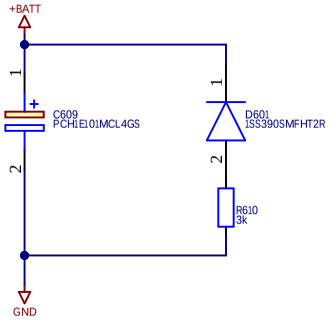


NTNU cSLIM v1	Title: cSLIM v1 - RS232		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022		Sheet 5 of 12	
File: cSLIM-SA_RS232.SchDoc		Drawn by: Vetle Berg Abrahamsen	

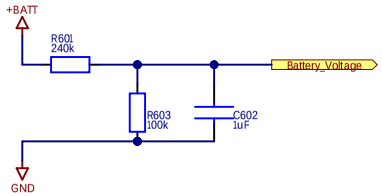

USB Connector
R501 15R
R502 15R
USB_D_N
USB_D_P
USB_N
USB_P
J504
10119313-302TLF
FB501
USB_VBUS
BLM18EG601SZ1D
U501
USBLC6-2SC6
I/O1 VBUS I/O2
I/O1 GND I/O2
R503 0R
C501 12pF
GND
uSD Card reader
J506
104031-0811
DAT2 CD/DAT3 CMD VDD CLK VSS DAT0 DAT1
uSD_nCS
uSD_MOSI
uSD_VDD
uSD_SCK
uSD_MISO
Auxiliary connector
J516
TSW-103-08-S-D
GPIO_BUS_P0
GPIO_BUS_P1
GPIO_BUS_P2
RS485 Connector
J509
BM05B-ZESS-TBT (LF)(SN)
TBR_POWER
R509 NR
RS485_A
RS485_B
TBR_PPS
TP502
MP1
MP2
RS232 Connector
J510
BM05B-ZESS-TBT (LF)(SN)
VDD
R508 NR
R511 NR
V_MUX_OUT
RS232_TX
RS232_RX
LEDs
R504 820R
R505 820R
R506 330R
R507 330R
R510 330R
LED501 APT2012LSECK/J3-PRV
LED502 APT2012LSYCK/J3-PRV
LED503 APT2012LZGCK
LED504 APT2012LVBC/D
LED505 APT2012LVBC/D
nLED_RED
nLED_YELLOW
nLED_GREEN
nLED_BLUE1
nLED_BLUE2
SIM Connector
SIM_1V8
C502 10nF
TP501
J513
SF72S006VBAR2500
VCC RST CLK GND VPP I/O CSW1 DSW1 GND1
GND2 GND3 GND4 GND5 GND6 GND7 GND8 GND9 GND10
SIM_RST
SIM_CLK
SIM_IO
GPS Antenna Connector uBlox
J507
CONUFL001-SMD-T
GPS_ANTENNA_EXT
SIG GND_1 GND_2
GPS Antenna Connector nRF
J514
CONUFL001-SMD-T
GPS_ANTENNA_nRF
nRF9160 SWD Connector
J512
SHF-105-01-L-D-SM-LC-K-TR
nRF_SWDIO
nRF_SWDCLK
nRESET
LoRa_module SWD Connector
J508
SHF-105-01-L-D-SM-LC-K-TR
LoRa_SWDIO
LoRa_SWDCLK
Battery Connector
J501
0705430001
+BATT
TP505
Regulator VIN jumper
J515
61300311121
V_MUX_OUT
VREG_IN
External regulator
J505
61300311121
TP504
2.54mm headers for external I2C and SPI devices
J502
PH1-04-UA
I2C_SCL
I2C_SDA
J503
61300611121
nRF_EXT_CS
SPI_SCK
SPI_MISO
SPI_MOSI
NTNU
cSLIM v1
Title
cSLIM v1 - Connectors
Size A4
Number
Revision v1
Date: 20.06.2022
Sheet 6 of 12
File: cSLIM-SA_connectors.SchDoc
Drawn by: Vetle Berg Abrahamsen

Polarization protection

Capacitor for battery supply
and diode to protect circuitry
of wrong polarization

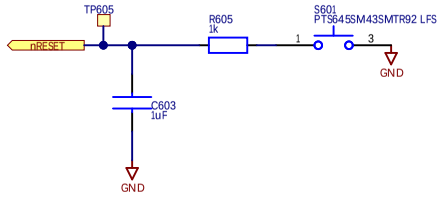


Battery measurement



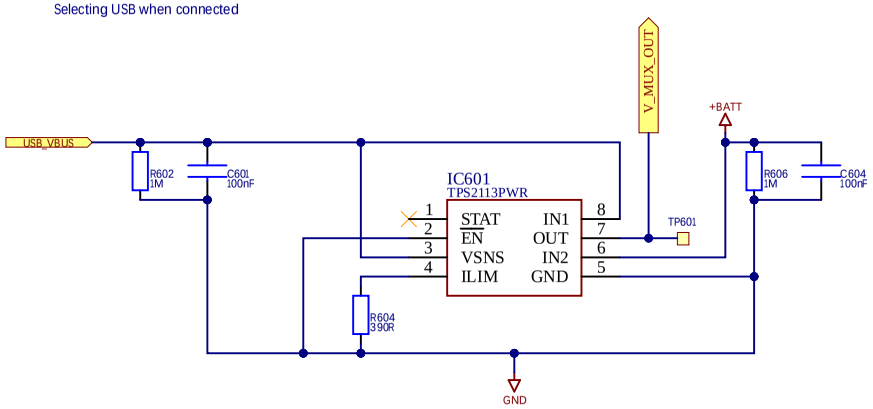
nRESET signal

RESET button
The nRESET pin of the nRF9160 has a 13kOhm internal pull up



Power multiplexer

Selecting USB when connected



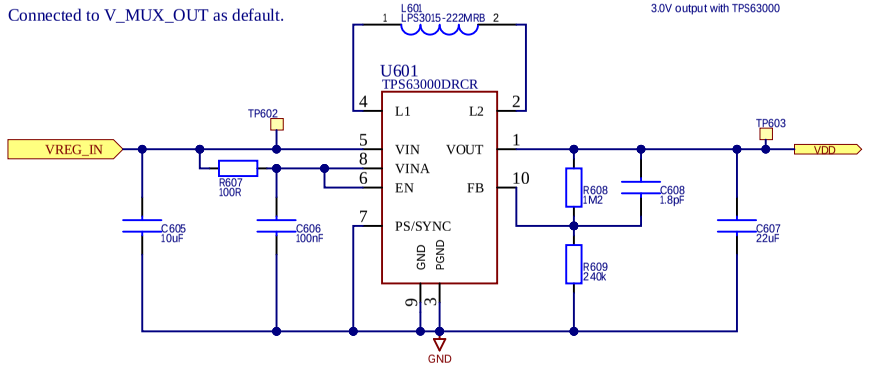
3.3V Buck-Boost regulator

VREG_IN selects either V_MUX_OUT or +BAT as VIN on Buck-boost regulator
Physical jumper J515. This is due to battery related problems
where the battery do not get selected by voltage multiplexer,
even when USB is not connected.

Connected to V_MUX_OUT as default.

R608 and R609 is not necessary when using TPS63001 as the output voltage is fixed to 3.3V. If another voltage is required use TPS63000 with appropriate resistor values

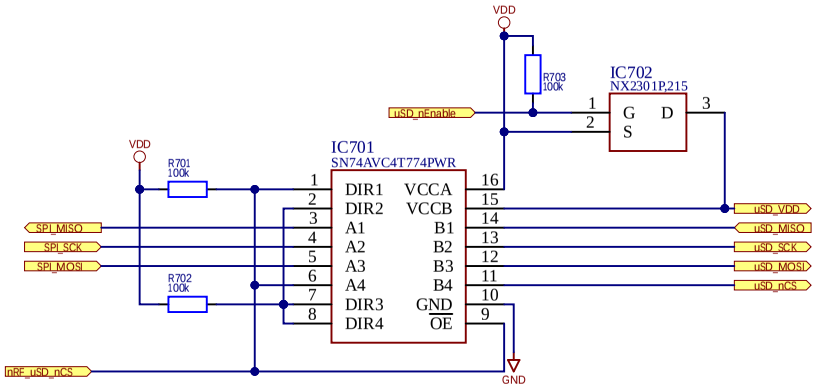
Resistor values computed for 3.0V output with TPS63000



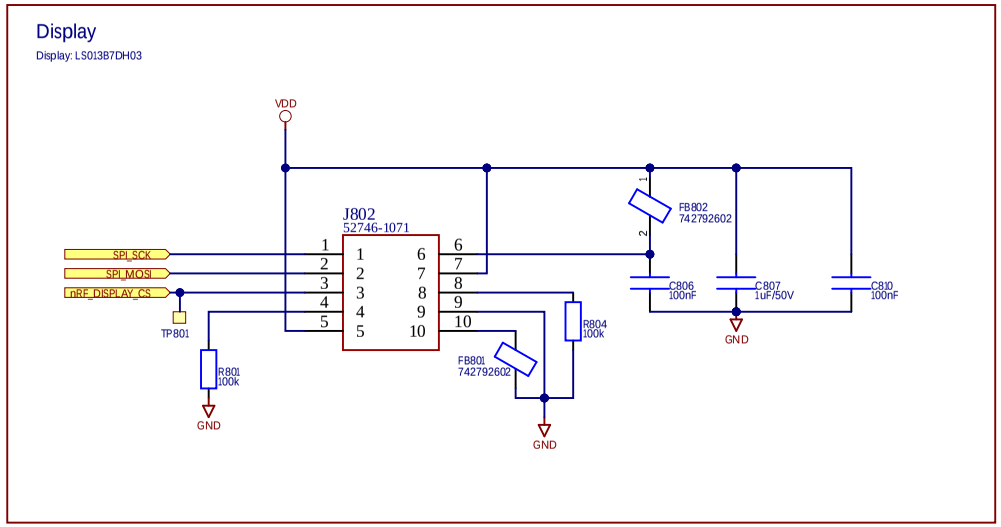
NTNU cSLIM v1	Title: cSLIM v1 - Power		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022		Sheet 7 of 12	
File: cSLIM-SA_power.SchDoc		Drawn by: Vetle Berg Abrahamsen	

uSD LLC

Logic level converter with controlled output enable used to ensure tri-state on MOSI pins when uSD is turned off



NTNU cSLIM v1	Title: cSLIM v1 - uSD		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022		Sheet 8 of 12	
File: cSLIM-SA_uSD.SchDoc		Drawn by: Vetle Berg Abrahamsen	

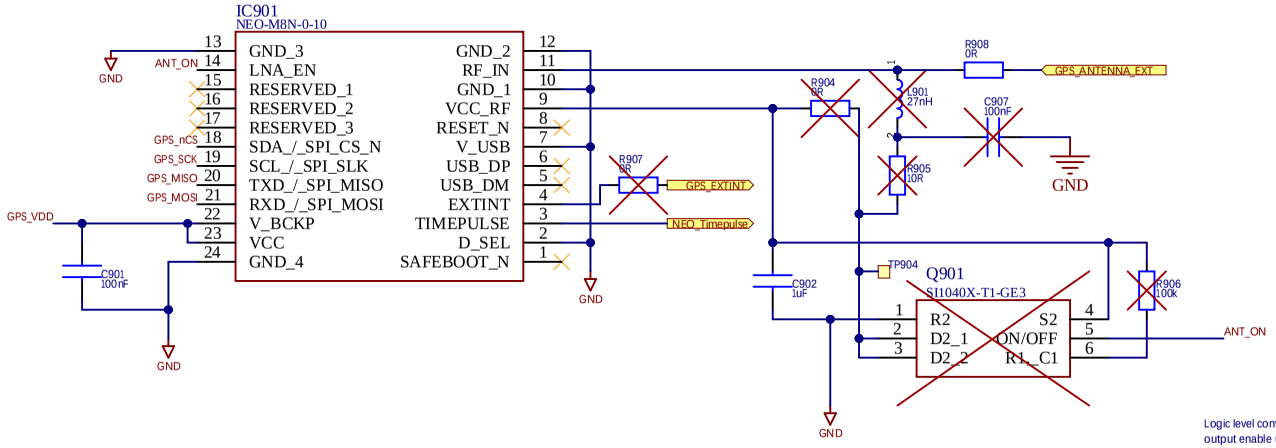


NTNU cSLIM v1	Title: cSLIM v1 - Display		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022		Sheet 9 of 12	
File: cSLIM-SA_display.SchDoc		Drawn by: Vetle Berg Abrahamsen	

Ublox GNSS receiver and LLC

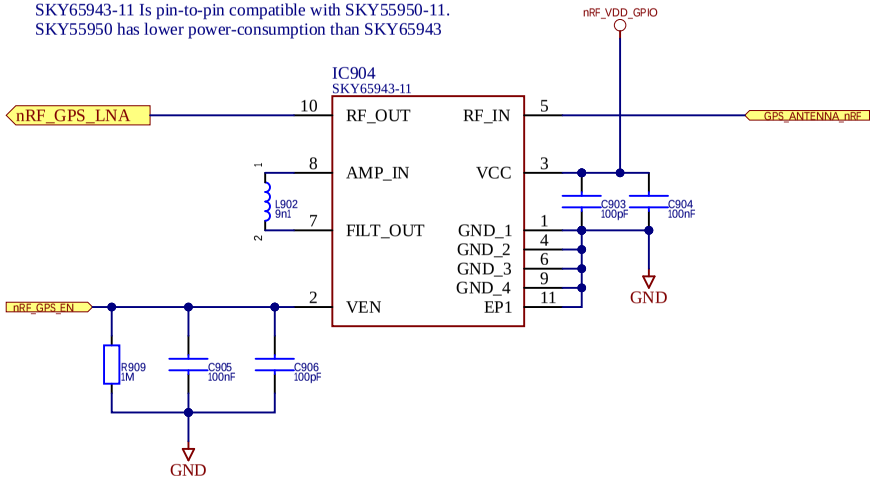
Q901, L901, R905, and C902 is only needed with active antenna (Taoglass FXP.611 is passive).R904 can be used instead of Q901, C902 and R906 for antenna always on.

Only mounting one of the two GNSS receivers is required. Both can be mounted.

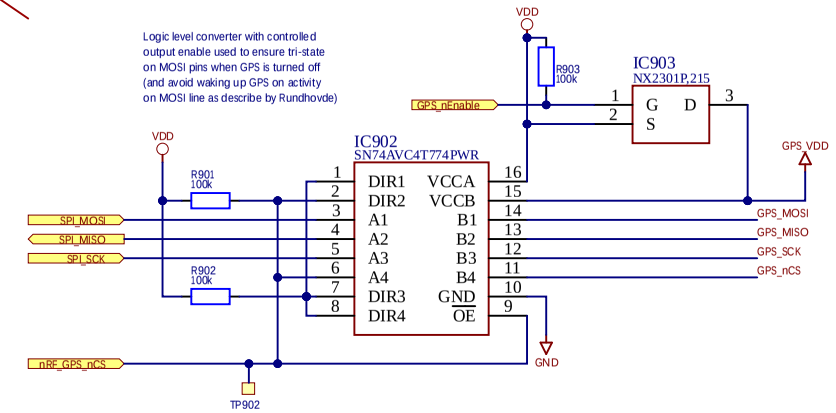


External LNA for internal GNSS receiver

SKY65943-11 Is pin-to-pin compatible with SKY55950-11.
SKY55950 has lower power-consumption than SKY65943

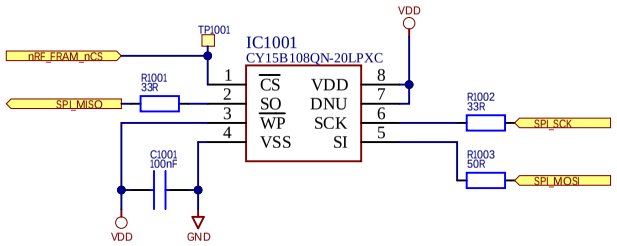


Logic level converter with controlled output enable used to ensure tri-state on MOSI pins when GPS is turned off (and avoid waking up GPS on activity on MOSI line as describe by Rundhovde)



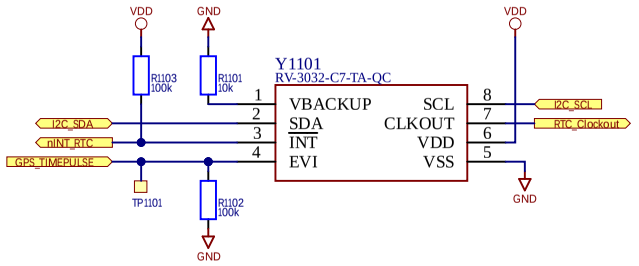
NTNU cSLIM v1	Title: cSLIM v1 - GNSS		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022		Sheet 10 of 12	
File: cSLIM-SA_GPS.SchDoc		Drawn by: Vetle Berg Abrahamsen	

FRAM



NTNU cSLIM v1	Title: cSLIM v1 - FRAM		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022			Sheet 11 of 12
File: cSLIM-SA_FRAM.SchDoc			Drawn by: Vetle Berg Abrahamsen

Real Time Clock



NTNU cSLIM v1	Title: cSLIM v1 - RTC		
	Size: A4	Number:	Revision: v1
Date: 20.06.2022			Sheet 12 of 12
File: cSLIM-SA_RTC.SchDoc			Drawn by: Vetle Berg Abrahamsen

A.5 nRESET modification

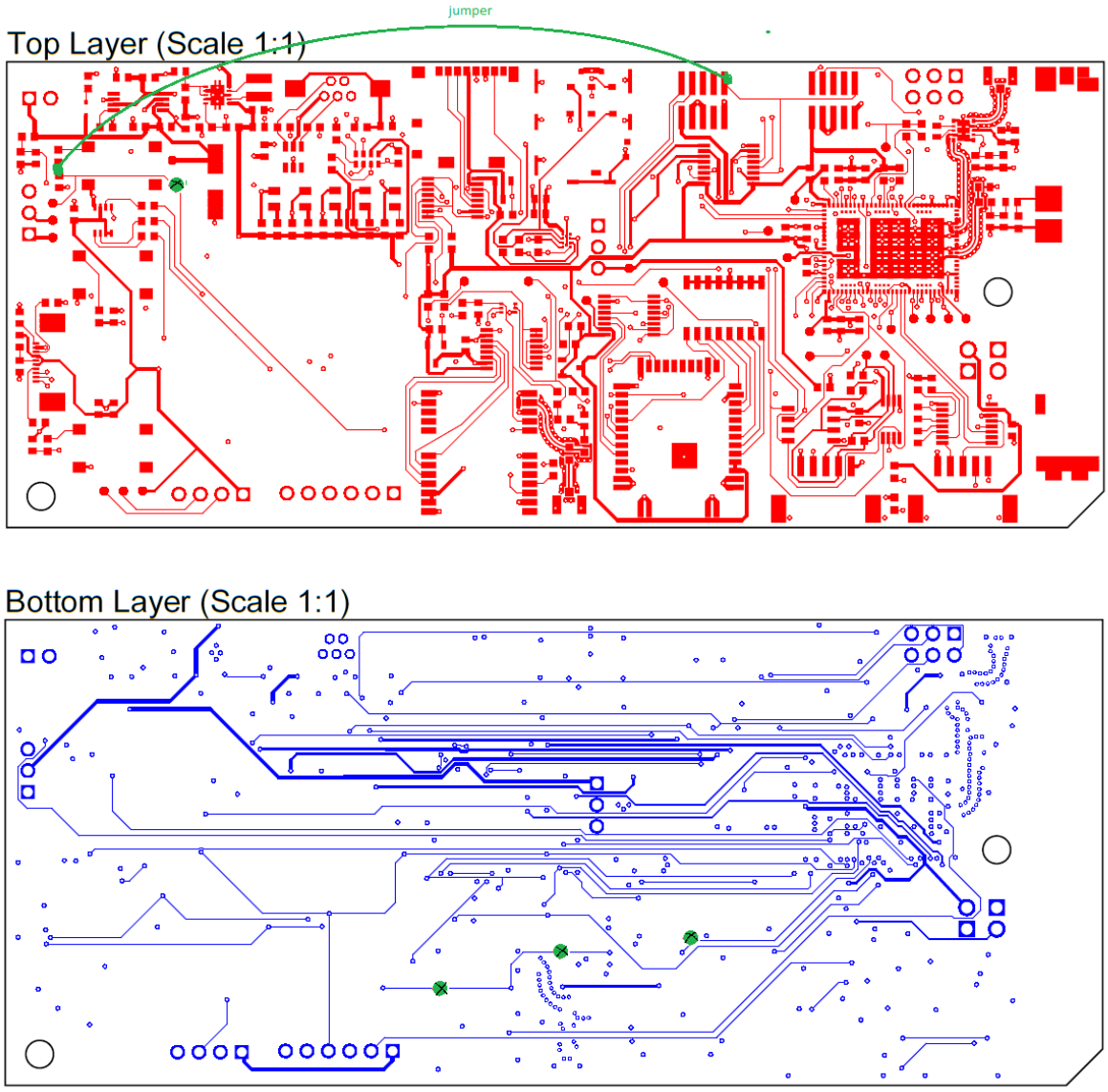
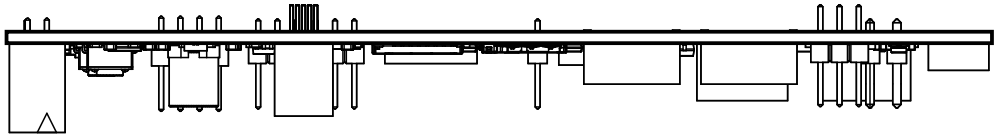


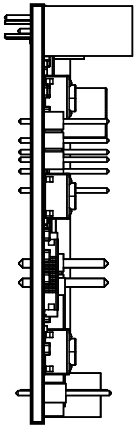
Figure A.2: Illustration of paths to cut and the jumper to bridge the reset line to nRF9160

A.6 Draftsman files

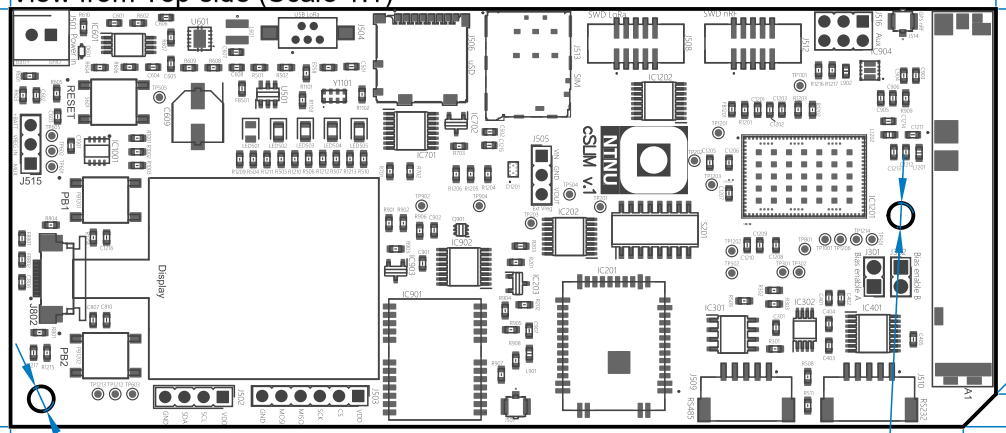
View from Back side (Scale 1:1)



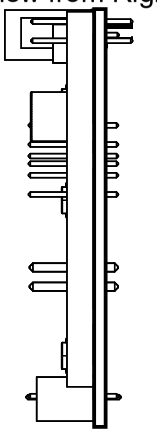
View from Left side (Scale 1:1)



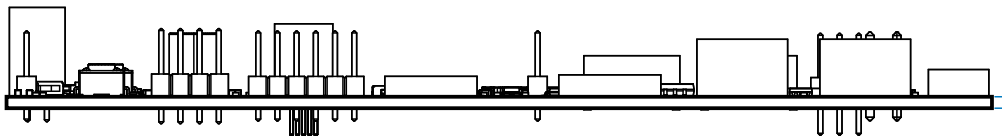
View from Top side (Scale 1:1)



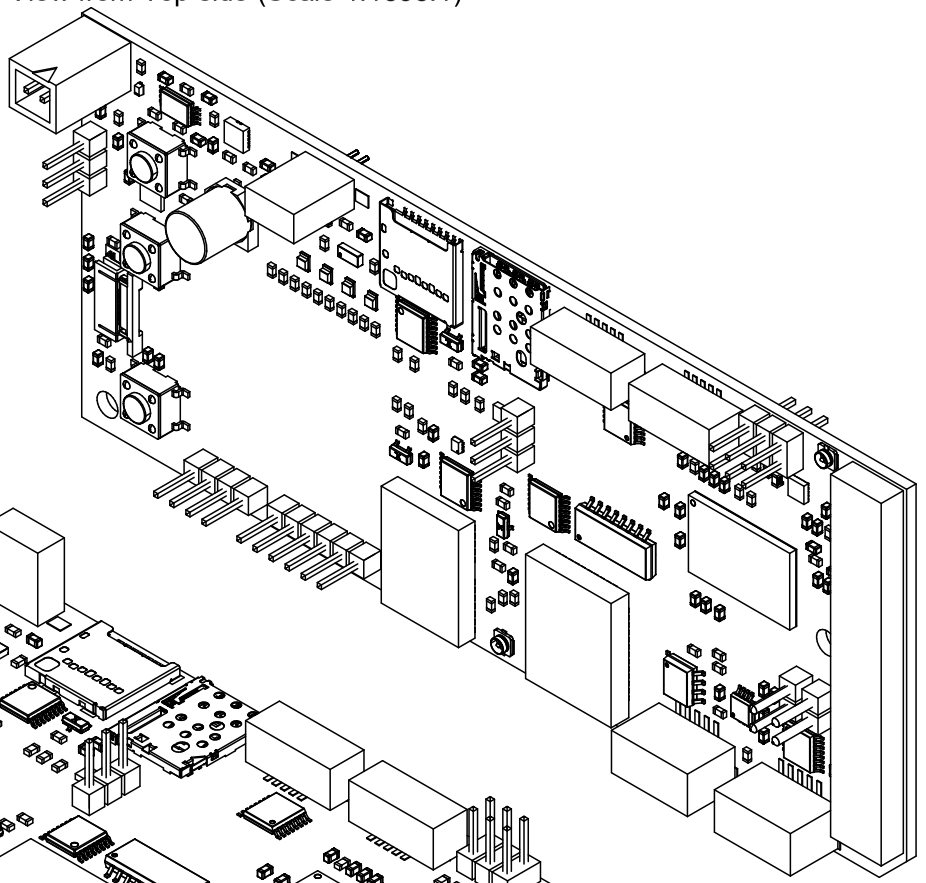
View from Right side (Scale 1:1)



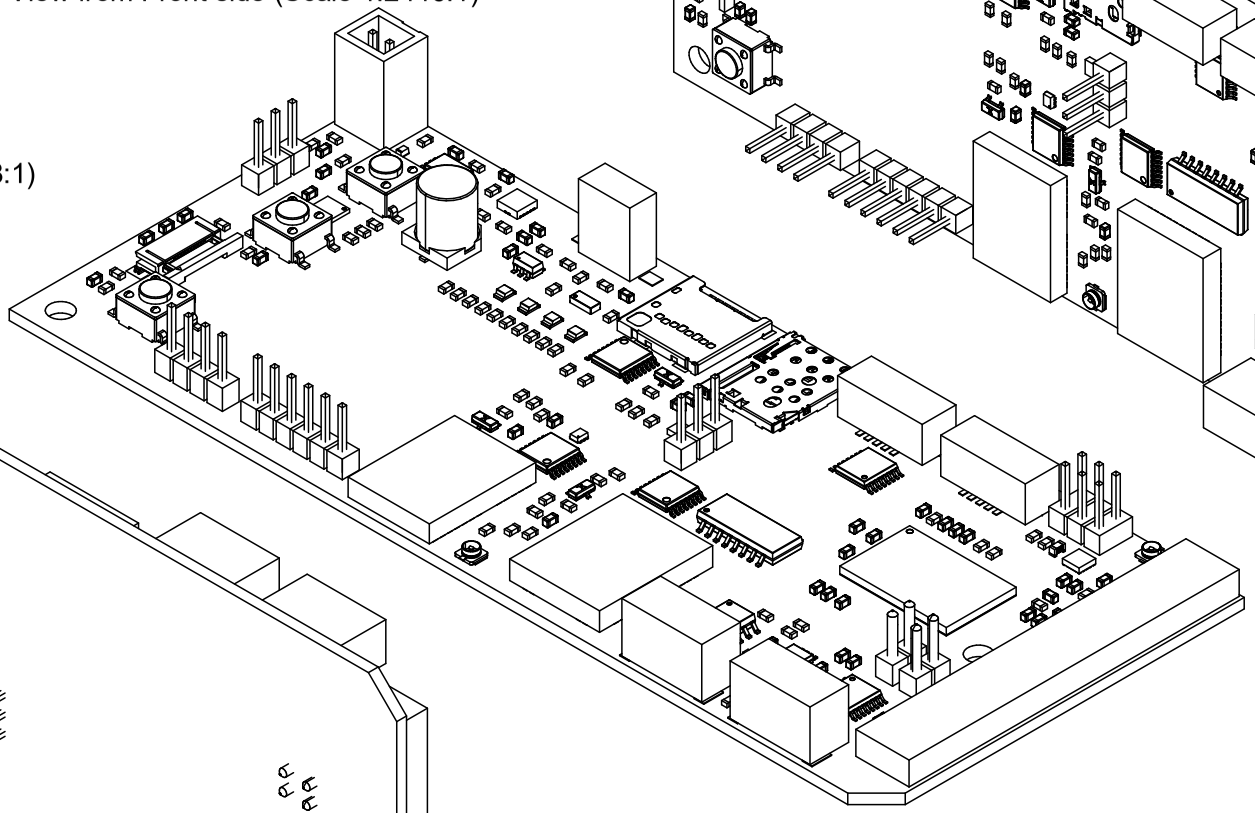
View from Front side (Scale 1:1)



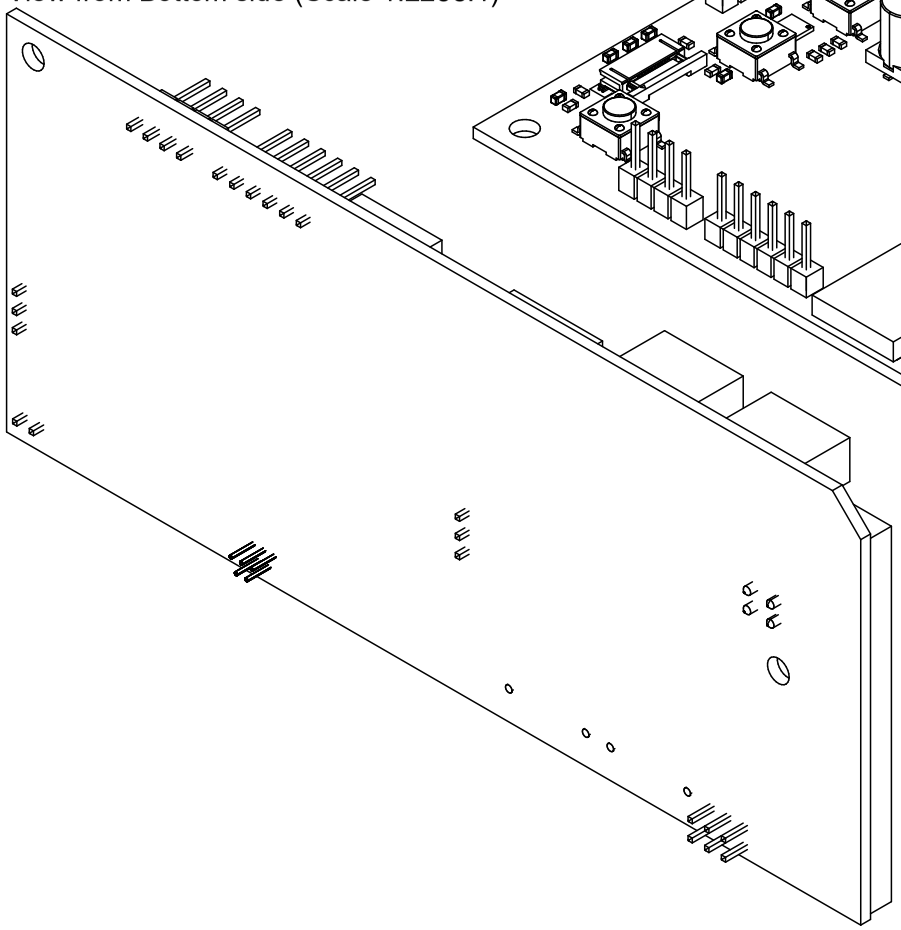
View from Top side (Scale 1.1838:1)



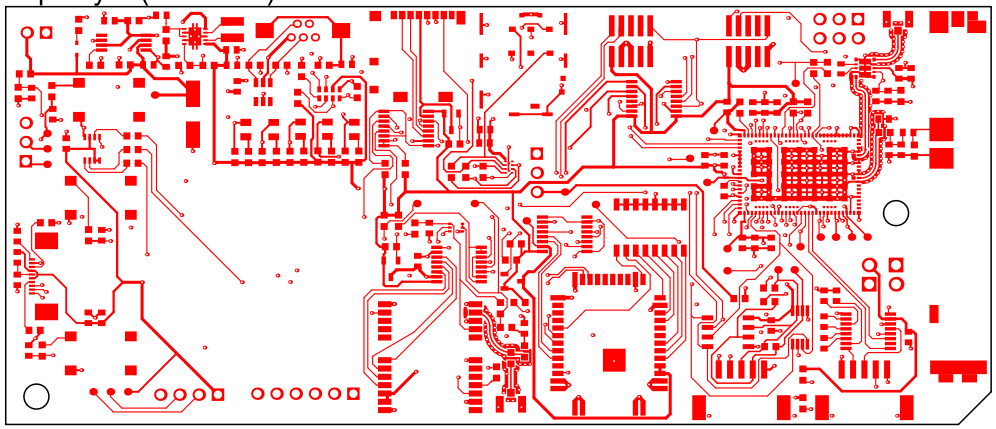
View from Front side (Scale 1.2446:1)



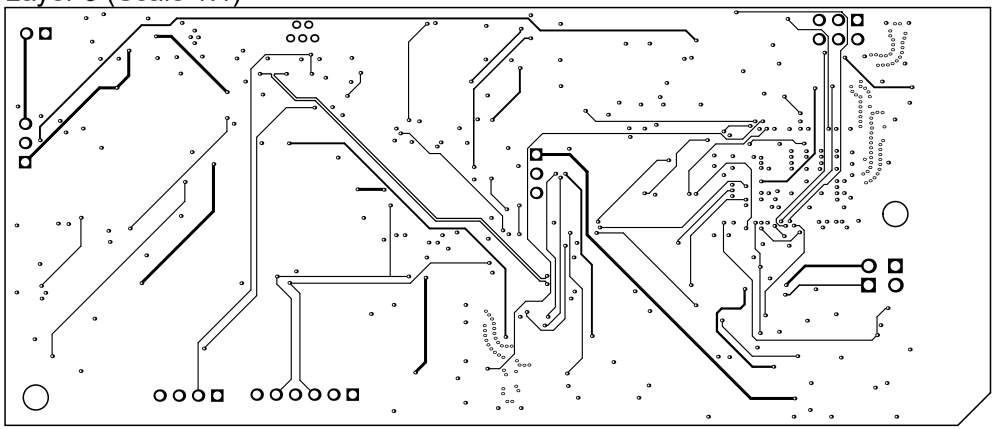
View from Bottom side (Scale 1.2258:1)



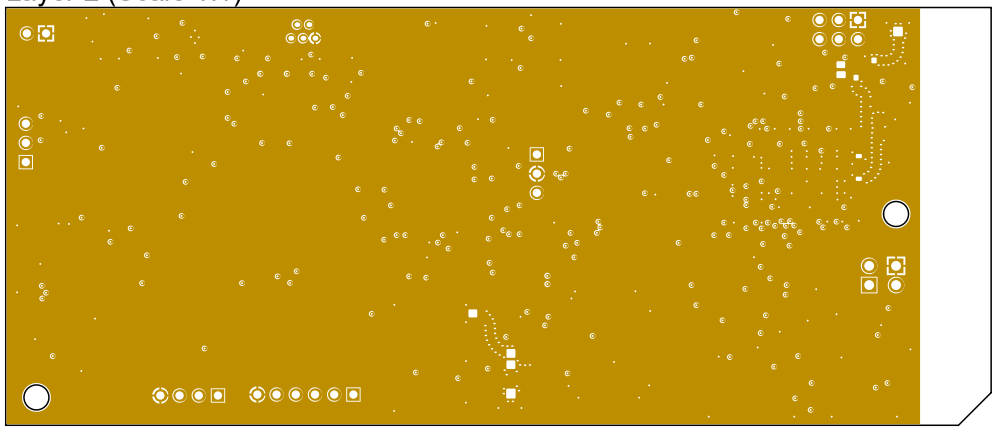
Top Layer (Scale 1:1)



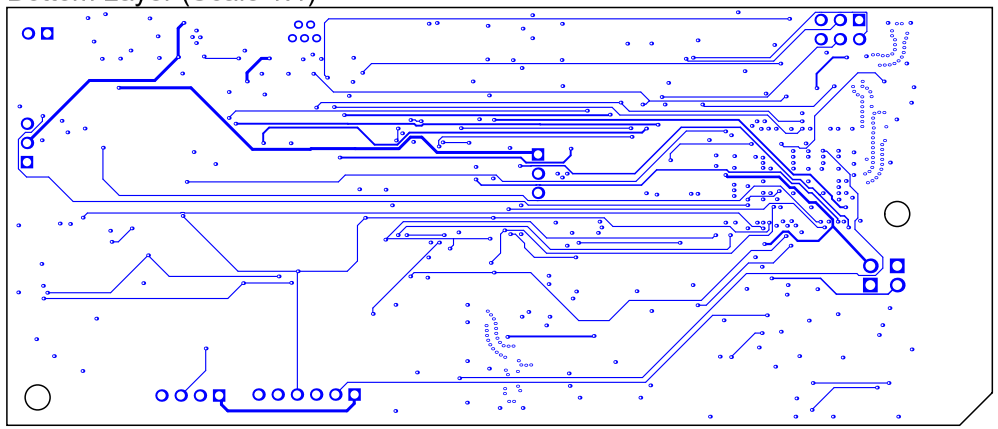
Layer 3 (Scale 1:1)



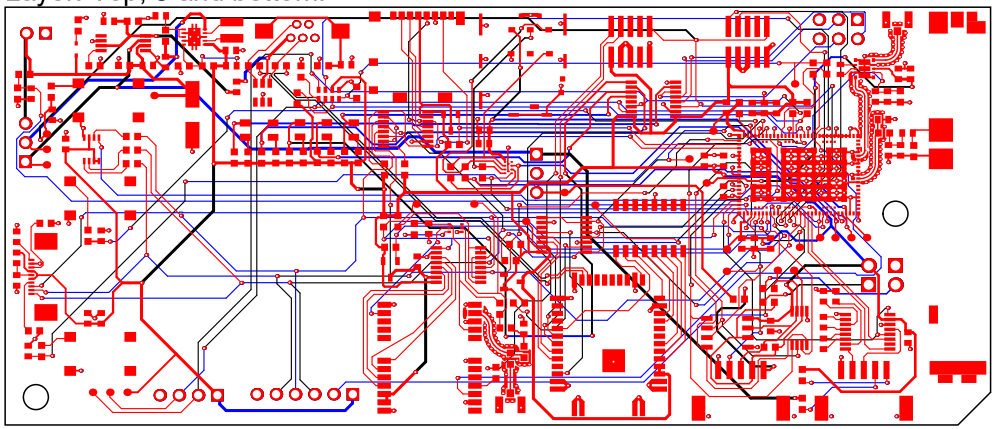
Layer 2 (Scale 1:1)



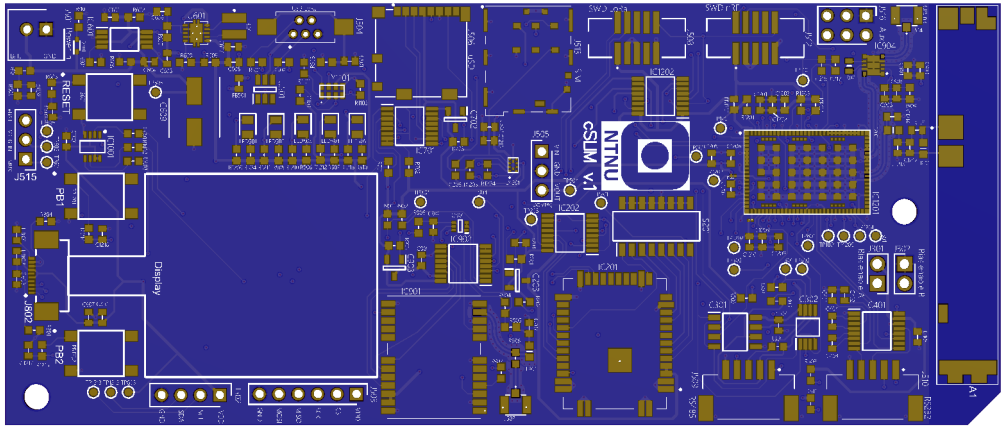
Bottom Layer (Scale 1:1)



Layer: Top, 3 and bottom.



Realistic View top



References

Altium.com (2020). Adding via stitching via shielding to a pcb in altium designer, https://www.altium.com/documentation/altium-designer/via-stitching-and-via-shielding-ad. Accessed: 2022-01-19.

Altium.com (n.d.a). All-inclusive design environment, https://www.altium.com/altium-designer/features. Accessed: 2021-11-25.

Altium.com (n.d.b). Easy, powerful, modern, https://www.altium.com/altium-designer/. Accessed: 2021-11-25.

AnalogDevices (n.d.). Fundamentals of rs-232 serial communications, https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/83.html. Accessed: 2022-05-19.

Batteries, T. (n.d.). Ltc batteries, https://tadiranbatteries.de/wp-content/uploads/2021/05/SL-2790.pdf. Accessed: 2022-06-19.

Bies, L. (2021). Rs485 serial information, https://www.lammertbies.nl/comm/info/rs-485. Accessed: 2022-05-19.

Bjerck, H. B. (2021). Synchrony and multimodality in the timing of atlantic salmon smolt migration in two norwegian fjords, https://www.nature.com/articles/s41598-021-85941-9. Accessed: 2022-06-15.

Campbell, S. (n.d.a). Basics of the i2c communication protocol, https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol. Accessed: 2022-01-18.

Campbell, S. (n.d.b). Basics of the spi communication protocol, https://www.circuitbasics.com/basics-of-the-spi-communication-protocol. Accessed: 2022-01-18.

Campbell, S. (n.d.c). Basics of uart communication, https://www.circuitbasics.com/basics-uart-communication/. Accessed: 2022-01-18.

Catchpoint.com (n.d.). The complete mqtt broker selection guide, https://www.catchpoint.com/network-admin-guide/mqtt-broker. Accessed: 2021-11-25.

de Ridder, L. (n.d.). Is lora the game-changer for iot?, https://internetofbusiness.com/lora-game-changer-iot/. Accessed: 2021-11-24.

Efteland, J. I. (2016). Underwater acoustic positioning system for real-time fish tracking, https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2417993. Accessed: 2022-06-17.

EYGM-Limited (2019). The norwegian aquaculture analysis 2019, https://assets.ey.com/content/dam/ey-sites/ey-com/no_no/topics/fiskeri-og-sj%C3%B8mat/norwegian-aquaculture-analysis_2019.pdf. Accessed: 2021-11-23.

FAO (2020). The state of world fisheries and aquaculture 2020. sustainability in action. rome, https://doi.org/10.4060/ca9229en. Accessed: 2021-11-23.

Hassan, W., Føre, M., Urke, H. A., Kristensen, T., Ulvund, J. B. and Alfredsen, J. A. (2019). System for real-time positioning and monitoring of fish in commercial marine farms based on acoustic telemetry and internet of fish (iof), https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2611967. Accessed: 2021-10-15.

Herrero, R. (2022). *LPWANLPWANTechnologies*, Springer International Publishing, Cham, pp. 193–212.
URL: *https://doi.org/10.1007/978-3-030-70080-5_8*

Jutila, E., Jokikokko, E. and Julkunen, M. (2005). The smolt run and postsmolt survival of atlantic salmon, salmo salar l., in relation to early summer water temperatures in the northern baltic sea, *Ecology of Freshwater Fish* **14**: 69 – 78.

Jølsgård, E. H. (2021). Lpwan connectivity and embedded solutions for smart ocean monitoring buoys, https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2828788. Accessed: 2021-09-10.

Kelly, J. (n.d.). Rs-485 serial interface explained, https://www.cuidevices.com/blog/rs-485-serial-interface-explained. Accessed: 2022-01-18.

Kjelsvik, P. A. (2019). Internet of fish: Real-time monitoring of fish through lpwan and internet technologies, https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2625681. Accessed: 2022-01-17.

Kjerstad, N. and Forssell, B. (2021). Gnss, https://snl.no/GNSS. Accessed: 2021-11-24.

Lentz, T. (2018). How does surface finish affect solder paste performance?, https://fctsolder.com/wp-content/uploads/2018/10/2018-SMTAI-How-Does-Surface-Finish-Affect-Solder-Paste-Performance.pdf. Accessed: 2021-10-12.

LoRa-Alliance (n.d.a). What are lora and lorawan?, https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/. Accessed: 2021-11-24.

LoRa-Alliance (n.d.b). What is lorawan specification, https://lora-alliance.org/about-lorawan/. Accessed: 2022-06-21.

Mekki, K., Bajic, E., Chaxel, F. and Meyer, F. (2019). A comparative study of lpwan technologies for large-scale iot deployment, *ICT Express* **5**(1): 1–7.
URL: *https://www.sciencedirect.com/science/article/pii/S2405959517302953*

Moe, E., Skage, M. and Helsengreen, M. B. (2020). The norwegian aquaculture analysis 2020, https://www.ey.com/en_no/strategy-transactions/key-megatrends-exposed-in-aquaculture-and-fishing-industry. Accessed: 2021-11-23.

Moe, E., Skage, M. and Helsengreen, M. B. (2022). The norwegian aquaculture analysis 2021, https://www.ey.com/en_no/strategy-transactions/ey-report-reveals-the-latest-aquaculture-and-fishing-industry-trends. Accessed: 2022-06-09.

MQTT.org (n.d.). Mqtt: The standard for iot messaging, https://mqtt.org. Accessed: 2021-11-25.

Nair, A. (2021). Introduction to influxdb: A time-series database, https://wearecommunity.io/communities/india-java-user-group/articles/891. Accessed: 2022-05-22.

NOAA (2021). What is a hydrophone, https://oceanservice.noaa.gov/facts/hydrophone.html. Accessed: 2021-11-24.

NordicSemiconductor (2019). Welcome to the nrf connect sdk!, https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/index.html. Accessed: 2022-01-22.

NordicSemiconductor (n.d.). nrf9160, https://www.nordicsemi.com/Products/nRF9160. Accessed: 2021-11-24.

OpenJS-Foundation (n.d.). About, https://nodered.org/about. Accessed: 2022-05-22.

Paiva, S., Branco, S. and Cabral, J. (2020). Design and power consumption analysis of a nb-iot end device for monitoring applications, *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2175–2182.
URL: *https://ieeexplore.ieee.org/document/9254374*

Parri, L., Parrino, S., Peruzzi, G. and Pozzebon, A. (2019). Low power wide area networks (lpwan) at sea: Performance analysis of offshore data transmission by means of lorawan connectivity for marine monitoring applications, *Sensors* **19**(14).
URL: *https://www.mdpi.com/1424-8220/19/14/3239*

Parri, L., Parrino, S., Peruzzi, G. and Pozzebon, A. (2020). A lorawan network infrastructure for the remote monitoring of offshore sea farms, *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6.

Pasqua, E. (2021). 5 things to know about the lpwan market in 2021, https://iot-analytics.com/5-things-to-know-lpwan-market/. Accessed: 2022-04-20.

Pelaez, A. (2020). Lorawan vs nb-iot: A comparison between iot trend-setters, https://ubidots.com/blog/lorawan-vs-nb-iot/. Accessed: 2022-06-22.

Rundhovde, M. (2020). Optimization of lora surface buoy for underwater acoustic receiver, https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2780999. Accessed: 2022-01-17.

Semiconductor, N. (2019). nrf9160 antenna and rf interface guidelines, https://infocenter.nordicsemi.com/pdf/nwp_033.pdf. Accessed: 2022-01-22.

Semtech (n.d.). What is lora?, https://www.semtech.com/lora/what-is-lora. Accessed: 2022-06-21.

Shea, S. (2017). Lpwan (low-power wide area network), https://internetofthingsagenda.techtarget.com/definition/LPWAN-low-power-wide-area-network. Accessed: 2021-11-24.

Skøien, K. R. (2021). Rtos: Real-time operating systems for embedded developers, https://blog.nordicsemi.com/getconnected/what-is-rtos-real-time-operating-systems-for-embedded-developers. Accessed: 2021-11-24.

Sultania, A. K., Zand, P., Blondia, C. and Famaey, J. (2018). Energy modeling and evaluation of nb-iot with psm and edrx, *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7.
URL: *https://ieeexplore.ieee.org/document/8644074*

Tadiran (n.d.). Tadiran lithium batteries, https://www.tme.eu/Document/66a4af5ebe2c06371f1b7b9951ee318d/TADIRAN%20LTC-Batteries.pdf. Accessed: 2022-05-27.

ThelmaBiotel (2021). Products, https://www.thelmabiotel.com. Accessed: 2021-11-24.

ThelmaBiotel (n.d.a). Thelma biotel live datasheet, https://www.thelmabiotel.com/wp-content/uploads/tb-live-datasheet-1.pdf. Accessed: 2022-05-22.

ThelmaBiotel (n.d.b). Transmitters, https://www.thelmabiotel.com/transmitters/9mm/. Accessed: 2022-05-24.

TheThingsNetwork (n.d.). What are lora and lorawan?, https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan. Accessed: 2021-11-24.

Vos, G. (2021). What is narrowband iot (nb-iot)?, https://www.sierrawireless.com/iot-blog/what-is-nb-iot. Accessed: 2021-11-24.

Williams, T. (1992). Chapter 5 - circuits, layout and grounding, *in* T. Williams (ed.), *EMC for Product Designers*, Newnes, pp. 121–169.
URL: *https://www.sciencedirect.com/topics/engineering/ground-plane*

Yang, S., Khan, S., Chuanxi, X., Yifeng, Z. and Shengchun, P. (2019). Design and realization of a buoy for ocean acoustic tomography in coastal sea based on nb-iot technology, *OCEANS 2019 - Marseille*, pp. 1–4.
URL: *https://ieeexplore.ieee.org/abstract/document/8867230*

Zephyrproject.org (2021). The zephyr project, https://zephyrproject.org/. Accessed: 2021-11-24.

Zourmand, A., Hing, A. L. K., Hung, C. W. and AbdulRehman, M. (2019). Internet of things (iot) using lora technology, *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 324–330.
URL: *https://ieeexplore.ieee.org/document/8825008*

Vetle Berg Abrahamsen

Realization of an ultra-low power embedded buoy controller

NTNU
Norwegian University of
Science and Technology