



Bachelor thesis

Computer Science and Engineering 300 credits.



An implementation of an
autonomous IoT system for real-
time water quality monitoring
with 4G and satellite connection

CatFish - Embedded systems

Halmstad 2022-05-06

Johan Elfing, Daniel Persson

Abstract

This thesis is about implementing an Internet of Things system for measuring water quality in rivers and other aquatic environments with an autonomous water drone, where the data from various components are collected and sent wirelessly to the database in real-time. A Raspberry Pi is connected to the internet through a 4G modem and a wireless satellite communication connection called RockBlock for emergency calls and notifications. In addition, a sonar is also implemented to collect data for the unmanned surface vehicle's (USV) avoidance of collisions. Finally, batteries are connected to solar panels to auto-generate energy and provide the USV with its requested current and voltage.

The minimum parameters to measure water quality are four: potential hydrogen, dissolved oxygen, nitrates, and colored dissolved organic matter. As a result, the system in this thesis measures the four parameters mentioned above, plus turbidity and temperature, since the interconnected sensors can also measure those. In addition, optical sensors were chosen because of their exceptional accuracy and precision when measuring water quality. The environment, mainly the aquatic, will benefit from this project and change for the better with time.

Sammanfattning

Den här vetenskapliga artikeln handlar om att implementera ett Internet of Things system avsett för att mäta vattenkvaliteten i floder och andra vattenmiljöer med en autonom vattendrönare, där datan från diverse komponenter samlas in och skickas trådlöst till databasen i realtid. En Raspberry Pi är ansluten till internet via ett 4G-modem samt en trådlös satellitkommunikationsanslutning som kallas RockBlock för nödmeddelanden och notifikationer. Dessutom implementeras ett ekolod för insamling av data till den autonoma vattendrönaren för undvikande av kollisioner. Slutligen är batterierna anslutna till solcellspaneler för att automatiskt generera energi och förse den autonoma vattendrönaren med tillräcklig ström samt spänning.

Det minsta antalet parametrar för att mäta vattenkvalitet är fyra: potentiellt väte, löst syre, nitrat, och färgat löst organiskt material. Som ett resultat mäter systemet i denna avhandling de fyra parametrarna nämnda ovan, plus grumlighet och temperatur, eftersom de sammankopplade sensorerna också kan mäta dessa. Dessutom valdes optiska sensorer på grund av deras exceptionella noggrannhet och precision vid vattenkvalitetsmätning. Miljön, speciellt vattenmiljön, kommer att gynnas av detta projekt samt förändras till det bättre med tiden.

Acknowledgement

We want to express our deepest appreciation to our supervisor Björn Åstrand for his patience, motivation, enthusiasm, knowledge, and continuous support. His guidance helped us significantly during the writing of this thesis, and we could not have asked for a better mentor.

We would also like to extend our gratitude to everybody involved in CatFish. Namely Alfred Mattsson, Daniel Crnic, Jesper Sköld, Joakim Tell, Martin Rabaud, Michael Forshlé, Denis Markitanov, Arnold Kramer, and Joakim Wahlberg. Thank you for putting down time and effort to make CatFish possible.

Contents

1. Introduction	1
1.1. The startup for the CatFish project	1
1.2. Benefits of mobility versus stationary measurements of data	2
1.3. Descriptions of autonomous drone	2
1.4. Overview of CatFish functionally	3
1.5. Purpose	3
1.6. Problem statement	4
1.6.1. How to measure water quality with sensors?	4
1.6.2. The parameters	4
1.6.3. Sensors	4
1.6.4. Selection of components and design	5
1.7. Project goals	5
1.8. Requirement specifications	5
1.8.1. Structure and code	5
1.8.2. Power consumption and safety	5
1.8.3. Sensors and data	6
1.8.4. Communication and data	6
1.9. Delimitations	6
2. Background	7
2.1. Aquatic unmanned surface vehicles	7
2.2. Measurements of the water quality	8
2.3. Optical Sensors	8
2.3.1. Ratiometric and intensity-based measurements	8
2.4 Antenna receivers	9
2.5. Hardware and software	9
2.6. Power	10
2.7. Communication with 4G	10
2.8. Literature review for parameters	11
2.9. Conclusions from similar projects	12
3. Method	13
3.1. Measurement methods	13
3.1.1. Parameters for water quality	13

3.1.2. Sensors	14
3.1.3. How to measure the water quality with sensors?.....	14
3.2. Component selection	15
3.2.1. Microprocessor	15
3.2.2. Camera.....	15
3.2.3. Internet.....	15
3.2.5. Battery, solar panel and regulator	16
3.2.6. Sonar	17
3.2.7. RockBlock and GPS (Satellite communication).....	18
3.2.8. Subsystem	19
3.3. Analysis of results	19
3.3.1. Testing the sensors and water quality parameters	20
3.3.2. Test of TpH-D sensor	21
3.3.3. Test of TpH-D sensor for resolution.....	21
3.3.4. Test of TpH-D OS of accuracy	22
3.3.5. Test of TpH-D OS of precision.....	22
4. Results	23
4.1. System overview	23
4.2. Subsystem overview	24
4.2.1 Power consumption.....	24
4.2.2 Collecting data	24
4.3. Summary of components	25
4.4 Result of sampling with OS TpH-D	26
4.4.1 Calculations for resolution, accuracy and precision	26
4.4.2 Summarize from the test of TpH-D sensor	27
4.5. Result of data sent to the database with 4G communication.....	28
4.6. RockBlock/GPS emergency and coordinates message via satellites communication	29
4.7. Sonar/Lidar result of depth and	30
4.8. Integration test	31
5. Discussion	33
5.1. Results from the sensor test	33
5.2. Lidar	33
5.3. Limitations of the project's solution	33

5.4. Next version of CatFish.....	34
5.4.1. Improvement of communication.....	34
5.4.2. Wind power instead or in addition to solar energy	34
5.4.3. Angle the solar panels to maximize energy	34
5.5. Social requirements for technical product development	34
6. Conclusions	37
6.1. How to measure water quality with sensors?	37
6.2. What are the minimum number of parameters needed for measuring water quality?	37
6.3. What is the most common sensor type to use, and how well can it measure the water quality?	37
6.4. What are the best components to chose according to interconnection compatability, and effectiveness?.....	37
6.5. The system.....	38
7. References	39
8. Appendices.....	49
A System	49
B Tables.....	52
C Code fragment	54
D Tests.....	56
Testing the microprocessor with communication	56
Testing the GPS/RockBlock for satellite communication	57
Testing the sonar's measurement and confidence	58
Testing the lidar	59
Test of solar panels subsystem.....	60
Datasheet of TpH-D sensor.....	60
Integration test	61
E Autonomous drone	63
F Timeline	65

Acronyms

LEVA	Lokalt Engagemang för Vatten/Local Engagement for Water, Havs och Vattenmyndigheten
LBVA	Laholmsbuktens VA
IP	International protection
USB	Universal Serial Bus
AWS	Amazon Web Services
pH	Potential hydrogen
DO	Dissolved Oxygen
NO3	Nitrate
CDOM	Colored Dissolved Organic Matter
GPS	Global Positioning System
INS	Inertial Navigation System
RTK	Real-time kinematic
FPS	Frames Per Second
RTU	Remote Terminal Unit
V	Voltage
Ah	Ampere Hours
W	Watt
VDC	Volts of Direct Current
PDB	Power Distribution Board

I. Introduction

I.1. The startup for the CatFish project

There is a lack of measurement data to describe the water quality in different watercourses. According to (Swedish: Laholmsbuktens VA, LBVA), the measurements [1] taken are insufficient, namely using platforms placed in the watercourses; these lead to measuring water quality at the same static places every time. If desired to trace pollution, the tracing will be expensive and take a significant amount of time. Hence, Halmstad municipality had to take action, so they contacted Halmstad University, which responded by initiating the project called CatFish.

On behalf of the government, the Sea and the water authority initiated the project (Swedish: Lokalt Engagemang för Vatten, LEVA) [2], and as a result, more motivation was found to help the aquatic environment [3]. The project aims to reduce the overfertilization of lakes, watercourses, and the marine environment.

CatFish was introduced to LEVA and started to work together for a better sustainable future. The cooperation led to CatFish obtaining funding to build an autonomous prototype for water quality measurements.

CatFish is a project that aims to measure the quality of the water to facilitate companies' curiosity and unawareness of how much pollutants they release into the water and to comply with the environmental requirements and laws existing in Sweden. The project collaborates with Halmstad municipality and LBVA, which wants to continue building a sustainable future. Halmstad municipality and LBVA need to be satisfied with all those aspects and parts combined.

Many aspects affect water quality like agriculture, industries, vehicle traffic, and other activities that increase sediment and nutrient leakage. In [2], they show that, e.g., chemicals, such as phosphorus and nitrogen, leaked from fertilized fields end up in watercourses. It contributes to eutrophication. In order to remedy these problems, sharper analysis tools are required, and one such could be CatFish.

These are the recommended values from Halmstad Municipality. The pH value should be of an interval of 7.0 to 7.5. Dissolved Oxygen is around 6.5 to 8 mg/L. Nitrates 0.01 to 0.08 mg/L to be regarded to measure good water quality.

In Sweden, a mass media company called (Swedish: Sveriges Television Aktiebolag, SVT) wrote a coverage about CatFish named "*Se när drönaren skickas ut på vattnet – ska ta reda på hur Nissan mår*" [4].



Figure 1. Overview of the Catfish project. The three autonomous drones (The bird, cat and fish), collect data, and then the data is then sent and stored in a database that an application receives to display. Note, the bird is currently excluded from the catfish project. Adopted from [5]

1.2. Benefits of mobility versus stationary measurements of data

With the help of CatFish, the difficulty of gathering water quality data in specific areas will facilitate since autonomous mobility will lead to more accessibility and flexibility. One of CatFish's purposes is to be lowered in an optional watercourse containing pollution and track down the source or sources. With the help of the analyzed water data, the municipalities, county administrative boards, and municipal companies can better understand local watercourses and make better, more environmentally friendly decisions.

1.3. Descriptions of autonomous drone

The heads of Innovation Lab and CatFish set the project's specifications with input from Halmstad municipality and LBVA, see section 1.7.

The design of this unmanned surface vehicle (USV) sets limits on what kind of components fits into this prototype's area. The parameters for water quality calculations have been taken into account to decide on the sensors after these specifications. The project will have a website for presenting the graphical result of the parameters.

The project's goal is to create a functional system consisting of interconnected autonomous drones, which consists of two individual drones that communicate with each other. "Cat" is the leading drone of the system, an autonomous surface vehicle that transports on the water surface. Then there is the project's second drone called "Fish", an autonomous underwater drone that will be used to collect data in the depths of the water. In simpler words, the Cat is like a boat while the Fish is like a hoistable submarine. Together, these two drones create the "CatFish" project, see figure 1, which serves as an umbrella for several student projects at Halmstad University.

A slightly more profound explanation regarding one of the project's drones, the Cat, is a so-called self-propelled boat that will receive coordinates from the user via its interface. The Cat then moves to these specific coordinators by utilizing artificial intelligence. When the Cat arrives at the specified

location, the Cat will collect data from the sensors at the surface, and later on, the Cat's winch will hoist down the Fish to gather data from the depths. Avoiding a collision with various obstacles underneath and above the surface is done using sonar and lidar, respectively, which is more effective than a camera and similar components because a camera needs visibility to function even at minimum effectiveness.

1.4. Overview of CatFish functionally

A user can send one or multiple watercourse coordinates to the CatFish, which will then be inserted into a queue. The prototype will then initialize a task of the first coordinate from the queue, proceed toward the coordinate and initialize the water quality measurement stage once it arrives.

A small summary of a mission includes the following stages. Stages 2,3,6,7,8,9 are from the previous CatFish prototype [6]:

1. Sending coordinates to the Cat.
2. Move towards specific coordinates.
3. Arriving at the specific coordinates and stabilizing the motors.
4. Check the power consumption.
5. Collecting data from both the Cat and the Fish.
6. Lowering the Fish.
7. Collect new data from both Cat and Fish.
8. Hoisting the Fish.
9. Continuing to the following coordinates.
10. Sends the data to the AWS (Amazon Web Services) database on a remote server in real-time. The database is connected to an application and website that displays the analyzed water quality data.
11. Start over and repeat from stage 2 if more coordinates exist in the queue.

1.5. Purpose

This project focuses on the embedded system, with the primary goal being to measure water quality with a minimal amount of parameters. In addition, the sub-goals are to ensure that the project can supply the USV with enough electricity from batteries and solar panels. 4G and satellite connections are necessary for sending the essential components' data to the database and, in case of an emergency, sending an emergency message, respectively, to avoid an error or hardware fault.

Implementation of lidar and sonar is required to provide data and information to the Cat and Fish for object detection and avoidance.

A live broadcast of real-time data from the project's camera to the website lets visitors see what is happening in real-time, making the project more appealing; see figure 2 for an overview of CatFish's intended embedded system.

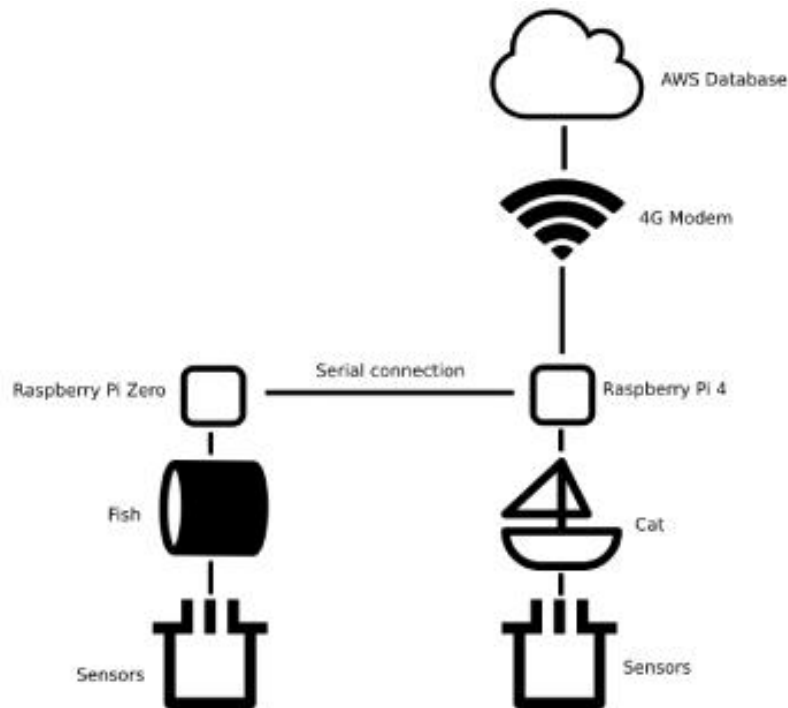


Figure 2. An overview of the Catfish's intended embedded system.

1.6. Problem statement

1.6.1. How to measure water quality with sensors?

The project's purpose is to measure water quality, and the problem is how to. Multiple parameters are received from water measurements worldwide, but not all are relevant for measuring water quality. Hence, there is potential to minimize the number of parameters. Although, the mandatory parameters from section 1.8.3. *How to measure water quality with sensors?*

1.6.2. The parameters

The main issue is figuring out which parameters contain the essential information about water quality, then deciding on a minimal amount of that collection. *What is the minimum number of parameters needed for measuring water quality?*

1.6.3. Sensors

We need to find the most appropriate sensors, even though some can measure one or multiple parameters. The sensors need to have an accuracy of $\pm 5\%$ and take measurements with $(\frac{1}{30})$ hertz, according to requirements in chapter 1.8.3. *What is the most common sensor type to use, and how well can it measure the water quality?*

1.6.4. Selection of components and design

Building a successful and efficient system design is very important. Everything from which components the system shall consist of to component placement, and in the end, program the hardware and establish a working communication between all the segments.

It takes quite the effort and time to choose the right components to determine the hardware. Mainly because of limitations and the need to analyze and familiarize with datasheets to, later on, be able to make the call on if the components are compatible altogether, waterproof, et cetera. Another aspect of why it is essential is that the components shall be placed on a boat, which will operate on water and not on solid ground; hence, the weight and positioning of the components need to be taken into consideration to prevent the boat from sinking. *What are the best components to choose according to interconnection compatability, and effectiveness?*

1.7. Project goals

Firstly, the primary goal of this project is to make the USV prototype able to measure water quality with proper parameters, and for that, choosing suitable sensors are needed. Secondly, the USV will also be autonomous. Therefore, gathering data for object detection is required, and sending emergency notifications if something with the project is inaccurate or in danger. In the end, all the data gathered will be sent to the database in real-time.

1.8. Requirement specifications

1.8.1. Structure and code

Create a well documented and working code for all the components

1.8.2. Power consumption and safety

Project requirements are to run for at least 2 hours per day. In addition, the project should be fully autonomous, leading to solar panels for re-charging the batteries with environmental power. They shall be installed to maximize energy, and finding the right angle for maximizing energy output was found in these articles [7] [8].

The system needs to be water- and fire-proof. Furthermore, the safety test of the BMS (Battery Management System) shall pursue the procedure outlined in [9] for the fire and explosion test. A more detailed article about BMS can be found in [10].

Since the CatFish will operate on the surface with dynamic temperatures, a requirement is that the IP (International protection) class will be chosen according to these conditions.

1.8.3. Sensors and data

According to Halmstad Municipality, the required parameters to measure water quality are pH(Potential hydrogen), DO(Dissolved Oxygen), CDOM(Colored Dissolved Organic Matter), NO₃(Nitrates) and log the data. In addition, these parameters will be the basis when choosing suitable sensors.

The system compatibility requires Modbus RTU, the physical interface used to transmit the electrical signal levels.

Sending the timestamp concurrently with the collected data to the AWS database wirelessly in real-time and JSON format [11].

Data shall be sampled with at least ± 5 % in measurement accuracy, have timestamps, and the intermediate hardware storage shall handle at least two hours of data.

1.8.4. Communication and data

The data shall be immediately stored until Cat receives a stable connection with a wide bandwidth before sending it to the cloud AWS database. To summarize, arrange a stable internet and satellite communication.

1.9. Delimitations

The primary funding is from LOVA at 2.2 million SEK, which sees CatFish as a three-year project. Besides LOVA, there are a few other smaller fundings, and when summarising everything, the final budget lands on a roughly 3.0 million SEK for CatFish during the life span of three years.

2. Background

Water quality detection is a critical requirement before all consumption from human activities, ranging from drinking purposes to industrial to agriculture usage. It is also one of the vital resources for all known forms of life on the earth to survive, which implies the necessity of freshwater for a sustainable ecosystem. Even though there is a clear argument for why freshwater is needed globally, and the amount of data gathering about water quality is growing as we enter a data-rich era [12], the analysis part of that data is impoverished. Hence, the gathered data becomes somewhat useless without being analyzed.

According to [12], the earth contains a fully 71% of water, and only 2.5% of this is fresh water. However, even if 2.5% is freshwater, not everything can be used by the lifeforms of the earth since the majority is buried deep in the ground or locked up in glaciers, ice caps, and permafrost, leaving less than 0.3% of the water usable which is from rivers, lakes, and the atmospheres.

2.1. Aquatic unmanned surface vehicles

Several research projects exist similar to CatFish, and many aquatic USVs measure water quality [13-21]. In addition, there are also flying drone projects measuring water quality [17] and boats with other purposes than monitoring the aquatic environment [22, 23]. The boats and drones have been and are undergoing development for numerous years with the primary mission to monitor pollution in the aquatic environment. As early as 1994, a group of engineers at the Massachusetts Institute of Technology demonstrated an aquatic project called RoboTuna, [15] a four-foot-long robotic fish. Since then, the Massachusetts Institute of Technology has developed many aquatic robots.

As for applications, several various applications can be applied depending on what kind of measurements the sensors can read. For instance, if oil is readable, just like the aquatic robot, Michigan State University started developing in 2010 [15] to monitor the impact and track the recent oil spill in the Gulf of Mexico. Then the robot can help trace the origin of the oil spill singlehandedly, making the water non-pollutant again. Alternatively, the application does not have to be monitoring water quality, like the Roboats [22] roaming around in the canal of Amsterdam with the primary mission to transport people to reduce heavy traffic on the bridge, et cetera.

An exciting project also worth mentioning is Vector [16]. The interesting part and the main benefit of the Vector is that the specifications and requirements for both the CatFish and the Vector are the most similar compared to any other projects found.

2.2. Measurements of the water quality

Water pollution has become one of the major issues over the last few years, especially for farmers nowadays [24, 25]. Since water is essential for humans, constant monitoring is necessary to develop an ideal sustainable aquatic environment. The selection of the physicochemical parameters to be monitored was based on a scientific literature review, see section 2.8. According to [26], the National Water Agency of Brazil, a primary water quality network, operates around 1340 water measurement stations nationwide. These reviews focus on analyzing pH, Dissolved Oxygen (DO), temperature, conductivity, flow, et cetera. With some acknowledgment, the standard parameters mentioned above can be refound when comparing these projects [16]. Although by analyzing the projects' parameters, it is noticeable that three out of the four projects measure turbidity, DO, and conductivity. These physicochemical parameters are the most important and can be used to detect particular water contaminations in water streams.

2.3. Optical Sensors

A working water quality sensor is typically characterized by three primary parameters: sensitivity, selectivity, and response time. Recent developments in optical sensors gained a high reputation in water quality detection due to increased sensitivity, selectivity, good response time, insensitivity to electromagnetic interference, and the possibility of real-time analysis [27].

Optical sensors use visible or ultraviolet light to interrogate sensors for analysis. Optical sensors are generally represented as a wavelength-selectable light source, the sensor material itself interacting with analytes, and a light detector, see figure 3. What the detector monitors vary by technique, can cover different regions of the electromagnetic spectrum, and allow measurement of multiple properties (e.g., the intensity of light, lifetime, polarization, et cetera.) [28]. To sum up, DO, pH, Turbidity, et cetera. can be measured using optical sensors because these parameters respond to light.

2.3.1. Ratiometric and intensity-based measurements

There are two types of measurements when using optical sensors, namely intensity-based and ratiometric measurements. Instruments that measure absolute intensity should be exceptionally well designed and calibrated. In contrast, ratiometric measurements are preferable to intensity-based measurements because they are insensitive to variations in light source intensity and changes in the light path. Furthermore, this approach negates the need for clean-ups and recalibrations of the sensor, as the sensor patches are already pre-calibrated [29, 30, 31].

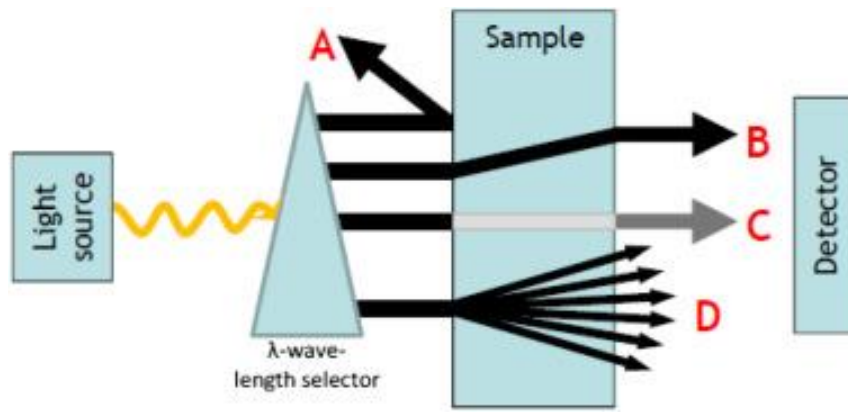


Figure 3. General arrangement of spectroscopic measurements: (A) light reflection; (B) light refraction; (C) light absorption; (D) fluorescent emission. Adopted from [28].

2.4 Antenna receivers

Having a GPS in an autonomous system is a must to determine the location through coordinates and then transport accordingly, as strengthened by table 8 in appendix B, where every project is autonomous and has a GPS. When comparing these projects [16], three different kinds of GPS were of interest, differential GPS (DGPS), RTK (Real-Time Kinematic), and Internal Navigation System (INS). The DGPS, RTK, and INS require a base station receiver set up at a known location and a rover that gets the corrections from the base station or a network of base stations. There is always a live communications link between the rover and the base station needed to receive these corrections. RTK is significantly more accurate than DGPS due to improved techniques and can be used for higher accuracy applications, like surveying. INS fuses Inertial Measurement Units (IMU) data with calibrated IMU data in the GPS solution and, as a result, object position, velocity, heading, and orientation data are collected. The update rate and the accuracy of INS are greater than the DGPS [32, 33].

2.5. Hardware and software

The Raspberry Pi is a popular microcomputer for many more significant projects. It has a very efficient and high data processing power than other microcontrollers such as an Arduino UNO [34] or an ESP32 [35]. In addition, the Jetson Nano Developer Kit [36] *"is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification and object detection"* [37], which matches well with a camera's usage. According to this face recognition project, they use a Jetson Nano Developer Kit since it is a powerful AI computer that delivers the estimated performance to run advanced AI workloads with a small size, low power, and low cost. It is also documented that their prototype with the Jetson Nano was processing 8.9 FPS compared to the last prototype consisting of a Raspberry Pi 4 [38] with a corresponding 2.6 FPS [39].

Which language the sensors and components will be programmed in will mainly decide what kind of microcontroller/computer will be used. Raspberry Pi 4 can program in Python, while Arduino, for example, can not

run Python directly. Some projects [16] use Raspberry Pi, while some use XILINX [40]. In the end, it comes down to preferences and simplicity.

2.6. Power

In order to make an autonomous boat with electric components onboard mobile, charging is essential for it to operate. Comparing similar autonomous boats' power supplies, as seen in table 1, a common thread is seen across various projects. Namely that every project [19-21, 23, 41], first and foremost, relies on having one or multiple stable and reliant batteries as the heart of its power supply. Moreover, out of those eight projects [19-21, 23, 41], even have around 12V batteries. The only project sticking out is Roboat II [23] which has 14.8V, which is not even that far away from the others. Another noticeable thread is that 50% of the projects take advantage of nature's energy in the form of wind and sun rays to convert it into power for continuously charging the batteries while operating, which leads to operating longer or for the same duration but in a more powerful way. Lead-acid and Li-PO are the most used batteries according to the comparison made in table 1; hence another comparison between those two batteries and LiFe-PO was created in appendix B, table 11, originating from table 12, which was made from the article [42].

Table 1. A compilation of what the power supply consists of from 8 various similar projects. Adopted from [19-21, 23, 41].

Reference	Aquatic boat	Battery amount	Battery voltage	Battery Ah	Battery type	Solar panel amount	Solar panel watts	Total Ah	Total solar watt	Wind turbine
[21]	AROO	1	12 V	4.2 Ah	Lead-Acid			4.2 Ah		
[21]	ARC	20	1.5V (12 V total)	2.5	AA			50 Ah		
[21]	Beagle-B	4	12 V	60 Ah	Lead-Acid	2	45W	240 Ah	90 W	
[21]	ASPIre	1	12 V	110 Ah	Gel Battery	1	50W	110 Ah	50 W	
[41]	Roboat	1	11.1 V		Li-PO					
[23]	Roboat II	1	14.8 V	22 Ah	Li-PO			22 Ah		
[19]	ASAROME	1	12 V	120 Ah		1	60W	120 Ah	60 W	50 W
[20]		1	12 V	20 Ah	LFP	2	40W	20 Ah	80 W	

2.7. Communication with 4G

A comparison chart was made when gathering information from the article [43], namely table 2, which will aid how an internet connection will be

established. The most common solution for internet access is having a 4G router or modem on the boat [13, 44] or any of the projects mentioned in [16]. 4G is more popular than 3G; even though it drains more power and battery, the advantages compensate for that and much more [45]. Some creative and interesting solutions were also found when looking at other projects that use communication through the internet. One example is using an android phone for internet connection through 4G [46]. Although this is a bit more expensive than just a 4G modem, it has multiple advantages. A standard phone already has information-rich components implemented that will save time, such as cameras, microphones, GPS, et cetera [47]. *Equation 1* can compare the time it takes to send the data packages to the database with different internet generations. Here, (t) is time in seconds, (FS) is the file size in megabytes, and (US) is upload speed in megabits [48].

$$t = \frac{FS}{\left(\frac{US}{8}\right)} \quad (1)$$

Table 2. A compilation of the frequency band, frequency spectrum, and maximum speed for the different network generations, namely 2G, 3G, and 4G. Adopted from [43]

Generation	Frequency Spectrum	Frequency Band	Maximum Speed	Maximum Speed (Generation upgrades)
2G	25 Mhz	900 Mhz	14.4 Kbps	150 Kbps (2.5G)
3G	15 Mhz – 20 Mhz	1800 Mhz – 2500 Mhz	2 Mps	42 Mps (3.5G & 3.75G)
4G	5 Mhz – 20 Mhz	2000 Mhz – 8000 Mhz	100 Mps	

2.8. Literature review for parameters

Finding suitable parameters for measurements is essential for water quality. The relation between the parameters and water quality concerns is well-established; The tables of parameters, see table 3, tables 8-10 in appendix B, and articles [16, 49-56] are utilized to create a summarization of the parameters used. However, these reviews mainly focus on various parameters needed to understand which parameters are necessary for measuring water quality. The connecting thread is pH, temperature, DO, and turbidity parameters.

Extensive scientific literature [56] mentions these essential parameters: physical, chemical, and biological parameters for measuring water quality.

The physical parameters of water quality are temperature and turbidity, and at a higher temperature of the water, the rate of chemical reactions commonly increases. In addition, turbidity can affect the taste and odor of drinking water.

Chemical parameters of water quality are pH, nitrogen, and DO.

Firstly, pH measures the amount of free hydrogen and hydroxyl ions in water since the pH value of water can be affected by chemicals. Namely, water with free hydroxyl ions is basic, and water with free hydrogen is acidic. Secondly, NO₃ supports the blossoming of algae and aquatic plants, which provides food and habitat for fishes and other aquatic living creatures. Thirdly, DO is essential for the survival of aquatic organisms and fishes. Nonetheless, the Halmstad municipality has an input aspect on the parameters that should be measured; they suggested that CDOM, pH, DO, and NO₃ are essential parameters.

Table 3. Table of parameters adopted from [18].

Parameter	Units	Quality Range
Temperature	Celsius	-
pH	pH	6.5-8.5
Electrical Conductivity	S/cm	500-1000
ORP	mV	650-800
Free Residual Chlorine	mg/L	0.2-2
Nitrates	mg/L	< 10
Dissolved Oxygen	mg/L	-
Turbidity	NTU	0-5

2.9. Conclusions from similar projects

When looking into similar aquatic USV projects, comparing the sensors and concluding that the Vector [16] used a multiparameter YSI Sonde [57], it combines multiple optical sensors into a large sensor that can measure all necessary parameters. Article [58] does not give the technical specification of the sensors, only mentioning the name "Atlas Scientific sensors," and most of them, according to their website, are optical.

The conclusion is that most successful projects that do not have a strict budget use optical sensors. According to the article mentioned above [58], they used an optical sensor that was quite expensive and reached a magnificent goal of measuring water quality. In contrast, there is [54], who did have a strict budget and could not afford optical sensors for all the required parameters to measure water quality. They tried a low-cost measure of water quality. As a result, NO₃ and DO were excluded. The disadvantage of optical sensors is that they are expensive, but the advantage is that they are accurate and do not need to be recalibrated every time they leave the water. This project has a budget that allows optical sensors, which helps us fulfill the required parameters to measure water quality. We can achieve a much better result with these sensors and save time if we do not need to recalibrate the sensors every time they are picked up from the water.

3. Method

This section provides an outline of the research methodology used to answer the multiple questions stated in section 1.6.

3.1. Measurement methods

3.1.1. Parameters for water quality

According to sections 2.2 and 2.8, information about both physical and chemical parameters can be measured for water quality. There is a huge list of parameters being utilized for this purpose. However, there is just a hand full of them that are essential for information on water quality. pH and DO are two physical parameters utilized in other similar projects. In addition, the CDOM was a demand from the Halmstad municipality, which stated it is crucial to measure. Some projects measure NO₃ since the OS [59] can measure that specific chemical, including the physical parameter turbidity.

In conclusion, the essential parameters for measuring water quality are pH, NO₃, DO, and CDOM based on the review from section 2.8. The temperature is not an essential parameter to qualify how the water quality is, even tho all of the articles from section 2.8 measure it. To summarize, the minimum number of parameters for measuring water quality is at least four; the parameters for this project are (pH-value, Temperature, DO, NO₃, Turbidity, and CDOM).

Temperature sensors operate by providing the reading via electrical signals; the sensors are combined with two metals that can generate an electrical voltage or even resistance when temperature changes by measuring the voltage across the diode terminals. When the voltages are increasing, so does the temperature [60].

The pH can be measured by measuring the voltage produced between two electrodes immersed in the water. The electrodes, made of a special glass, are called the measurement electrode, and a small voltage proportional to the pH is generated between the electrodes. The pH of water can also be calculated by *equation 2*, where ($[H_3O^+]$) is the oxonium ion. The chosen sensor uses the buffer solution to calculate the specific pH, which is a solution based on the Henderson-Hasselbalch equation; see *equation 3*. Where (pKa) is the dissociation constant for the weak acid, ($[A^-]$) is the concentration of conjugate base, and ($[HA]$) is the concentration of the weak acid. [61, 62]

$$pH = -\log[H_3O^+]$$

(2)

$$pH = pKa + \log_{10} \frac{[A^-]}{[HA]}$$

A comparison between the temperature and its scale is needed to determine Dissolved Oxygen; see table 4. The result for Dissolved Oxygen can be either mg/L or ppm.

Table 4. A table adopted from [63]

Temperature [C°]	Solubility [mg/L]
0	14.6
10	11.3
20	9.2
30	7.7

3.1.2. Sensors

To monitor and collect the physicochemical parameters of water (e.g., pH, DO, CDOM, NO₃), a sensor is required. The sensors transform the parameter/parameters into digital information and send them to the microcomputer in a predefined time interval. Once the data of all sensors are collected, a protocol will be formatted and sent to the Raspberry Pi 4 board through the USB port.

The sensor YSI 6600 Sonde was of interest since the USV Beagle-B has used it for their project [16, 64]. Their sensor is a multiparameter, a singular sensor that can gather multiple parameters. Moreover, since the Fish of CatFish will be hoisted down to take the depth measurements, the YSI 6600 Sonde sensor was excluded. In addition, the sensors need to be separated to meet CatFish's requirements, so another choice of sensors could be the ones in the article [57]. However, according to the cooperation with Aquacom as an indirectly sub-seller to TriOS, their optical sensors could be bought at an affordable price within the project's budget [59, 65-67].

First of all, the parameters from section 1.8.3 are suitable for their sensor and measurement accuracy. As mentioned in this paper, *"TriOS' optical sensors have proven themselves as very reliable products of in-situ measurements of NO₃, NO₂, DOC, TSS, SAK254, turbidity as well as Chlorophyll, Phycocyanine, PAH, Colours and all electrochemical parameters over time."* [68]. Secondly, the recalibration time interval was between 4 weeks [66] and 12 months [59, 65, 67]. Recalibration is solved; they will send a technical guy to recalibrate the sensors. Thirdly, according to the datasheet of the sensors, the power supply is 12 to 24 VDC which is suitable for the project's system and power supply.

3.1.3. How to measure the water quality with sensors?

Finding a suitable method for this project to measure water quality with sensors is based on the knowledge done by searching for information about various methods and projects. In addition, the pathway is based on a scientific literature review; see 2.8. However, combining

sections 3.1.1 and 3.1.2 gives us information about the minimal number of parameters for the sensors and what kind of sensor.

3.2. Component selection

3.2.1. Microprocessor

Choosing a suitable microcomputer is crucial as the whole system depends on that one microcomputer managing all of the other interconnected components. Hence, the Raspberry Pi 4 was conveniently chosen because of its efficient and high data processing power compared to other microcontrollers such as an Arduino UNO or an ESP32, as read in section 2.5. When it came down to choosing a microprocessor for the camera [69], a conclusion was made to choose the Jetson Nano Developer Kit. This decision was made because of the backed arguments mentioned in section 2.5 about its effectiveness in image classification and object detection, which is the camera's job in this project. However, the nail in the coffin for this decision was the two prototypes made in a project for object detection. The one with a Raspberry Pi 4 lost significantly in FPS comparison against the corresponding prototype with a Jetson Nano Developer Kit instead, with a grand 3.42 times as much more FPS [39]. This kind of FPS increment increases the effectiveness of object detection significantly since the code can detect dangers a lot sooner, which results in the chances of a crash being lowered enormously.

3.2.2. Camera

A camera is needed to stream visuals of what is happening on the CatFish in real-time. Moreover, for it to be as effective as possible, a camera with small dimensions, a wide camera angle, and a high resolution is desired. The choice to go with the PiCam360 [69] was made since it can stream up to 4k HDR, an excellent resolution, with a wide-angle range of Horizontally 360° and Vertically 220°. The dimensions are also tiny, and the website itself recommends the Jetson Nano for interconnecting to this camera, which also strengthens this project's choice of a Jetson Nano Dev Kit.

3.2.3. Internet

For the data to be valuable and initiate the analyzing stage, it needs to be sent to the database via the internet. As seen in table 2 and section 2.7, speed is the main difference between the various generations of internet connections, and the power consumption also varies somewhat. However, the project's batteries [70] are mighty, and in comparison, the power consumption difference between the generations becomes non-existent. When comparing the parameters of the generations, 4G LTE wins the comparison by a mile since it has a higher and better range of frequency bands. This means that the project will be able to send more data within a shorter amount of time; hence 4G is the better option as the website will get

more parameter information faster which is great since the project is real-time based. Furthermore, 5G was disregarded in the decision since the development has not come far yet. Therefore, 4G was chosen because of its effectiveness, is relatively new, and is more attractive than older generations.

3.2.5. Battery, solar panel and regulator

Having enough power supply is critical, especially for a boat out in the marine environment without any fuel in case of running out of power. Therefore, a comparison to other similar autonomously driven aquatic USVs was made into a table, as seen in table 1 from section 2.6. Beagle-B [21] is the largest boat and the most power-consuming project of them all; hence, a comparison to Beagle-B was made to ensure this project was not running out of power. In addition, *equation 4* is used for calculating the watt strength of the solar panels [71]. In section 2.6, table 1, the Beagle-B uses four 12V batteries with a summarised 240 Ah. Therefore CatFish chose to go with two 12.8V batteries, with 96 Ah per battery. This sums up to 192 Ah, and comparing that to Beagle-B's, CatFish is barely falling short, which is understandable because of the project's size difference in power consumption where CatFish falls as the lesser one. Another choice is to go for LiFe-PO batteries instead of Li-Po and Lead-Acid batteries; since the LiFe-PO has excellent energy density, working temperature, and many cycles before reaching 80% SOH, as seen in table 5. In addition, power consumption for the components is summarized in table 5. As a result, our system consumes only about 4.566A out of the 192 usable Ah, which is more than enough to meet the requirement for being operatable for two hours.

$$\text{Solar Panel in KW} = \frac{\text{KWH Per month}}{(\text{Sunhours per month}) \times \text{Derate (0.77)}} \quad (4)$$

Table 5. Summary of the components' power consumption in ampere compared to the corresponding datasheet.

Component	Ampere measured	Ampere according to datasheet
Raspberry	0.94A	
Jetson Nano	1.02A	
NicoPLUS	0.57A	0.58A
MicroFlu	0.09A	0.09A
TpH-D	0.018A	0.016A
Dissolved Oxygen	0.084A	0.083A
Lidar	0.98A	0.95A
Sonar	0.1A	0.1A
RockBlock	0.49A	0.45A
PiCam360	0.24A	
GPS-RTK	0.034A	0.035A
In total:	4.566A	2.304A + Raspberry, Jetson, and PiCam360

3.2.6. Sonar

Since the boat will be autonomous, there must be a way of identifying objects nearby in order for the boat to avoid it or them. If this project were an automobile or anything similar on a hard surface, cameras would be enough for object detection. However, this is a boat; hence we can use cameras for the above water object detection part, but not for underwater. Something else is needed to detect underwater objects since a camera needs the vision to function as it should, which is uncommon, especially since this project will primarily measure water quality in places with dirty water. Therefore, using a camera was discarded for underwater object detection, and the choice stood between sonars or lidars, which do not need vision but use sound and light waves, respectively. Since sonars are effective even in low-visibility situations [72] and a quality lidar is too expensive, the CatFish was equipped with sonars [73] for underwater object detection instead of cameras and lidars.

Sonars have the default device configuration as a slave in a master-slave network [74], which means that the sonar will only transmit and send data when the master has requested it. The transmitting circuitry inside the sonar's transducer accepts a request from a computer to transmit a pulse, then creates something called a "pulse train," which sends out multiple pulses at a particular frequency depending on the sonar. As a result, the time elapsed clock is activated, and the pulses are emitted by the transmitter circuitry that gets shut down immediately after sending the signal. In contrast, the receiving circuitry for the same transducer gets activated. However, it gets activated after a slight delay. This is necessary since the foil layer in the actual sonar still has noticeable vibrations after emitting the pulse; it needs a brief period to self-dampen so that the residual vibrations do not get misinterpreted as a receiving echo pulse. Therefore, the delay needs to be long enough to self-dampen the vibrations until they are less than the reception of the echo pulse [75].

Later, when the high-frequency signal hits the target, the ultrasound is partially absorbed by the object and partially reflected towards the sonar, with a significantly attenuated pulse; which is attenuated by a factor of 106 or more due to atmospheric absorption, partial target reflection, and the inverse square law. When the attenuated reflected signal hits the receiving part of the transducer, it detects a change in capacitance and reports back to the computer through a digital signal that a pulse-echo has arrived. As a result, the elapsed time clock stops and gets logged as (Δt) ; it is also used concurrently with the velocity of sound (v_s) to calculate the travel distance back and forth to the object (d), as seen in *equation 5*. Furthermore, if we divide the right joint in *equation 5* by two, we get *equation 5.1*, which solely calculates the distance to the object (d_o) [75].

The velocity for sound underwater (v_{sw}) is given by *equation 6* acquired from [76], where (T) is degrees in Celsius and (v_{sw}) is the velocity for sound underwater in ($m \times s^{-1}$).

$$d = v_s \times \Delta t \quad (5)$$

$$d_o = \frac{(v_s \times \Delta t)}{2} \quad (5.1)$$

$$v_{sw} = 1404.3 + 4.7 \times T - 0.04 \times T^2 \quad (6)$$

As for algorithms, the sonar in this project runs a tracking algorithm that focuses on detecting the most significant object within the area of the sonar's view. The algorithm for determining the target considers the most powerful return strength and earlier measurements; in conclusion, it works like a low-pass filter and filtrates away all the signals with a weak return pulse. In addition to previously mentioned, the tracking algorithm also produces a confidence measurement that indicates the probability that the object has been correctly identified [74].

Sonars can be both passive and active. The difference between passive and active sonars is that active sonars emit and receive signals, while passive sonars solely listen and receive signals produced from external sources. Passive sonars have historically had the central area of use within military applications, such as listening and trying to detect enemies, especially submarines [77].

3.2.7. RockBlock and GPS (Satellite communication)

Since the CatFish will transport itself according to given coordinates through the website, it must know where exactly it is located. The integrated RTK receiver is a technique that uses carrier-based ranging and provides ranges and positions. RTK is used for higher accuracy applications, such as centimeter-level positioning, with up to 1 cm accuracy. The network for RTK is based on the use of several widely spaced permanent stations, and positioning data from the permanent stations are regularly communicated to a central processing station. On-demand from RTK user terminals, which transmit their approximate location to the central station, the central station then calculates and transmits correction information or the corrected position to the RTK user terminal. The benefit of this approach is an overall reduction in the number of RTK base stations required, and the data is transmitted with wireless communication [78].

Accuracy is a specification and aspect of a GPS. Hence a choice to go for a GPS with an implemented RTK board arose since the RTK board has an accuracy of 25mm [79].

Determining the great circle distance between two points, i.e., the shortest distance from point A to point B over the earth's surface, while ignoring and going through all existing obstacles is calculated by the Haversine formula, as seen in *equations 7-9*. Where $(\Delta\varphi)$ is the difference in latitude and $(\Delta\lambda)$ is the difference in longitude from both points in decimal form, in addition (φ_1, φ_2) is their individual latitude. Furthermore, (a) is half the length of the straight line between the points squared, (c) is the distance traveled on the surface in sphere radians, (d) is the distance between the points, and lastly, (R) , which is the radius of the earth in an optional unit which in turn decides the unit for the distance as well [80].

$$a = \sin\left(\frac{\Delta\varphi}{2}\right)^2 + \cos(\varphi_1) \times \cos(\varphi_2) \times \sin\left(\frac{\Delta\lambda}{2}\right)^2 \quad (7)$$

$$c = 2 \times a \times \tan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \quad (8)$$

$$d = R \times c \quad (9)$$

As for communicating in case of emergency, an emergency message should be sent to the web page or specific individuals even if CatFish loses internet connection. An easy solution was desired; therefore, the project went with the RockBlock MK2 [81], which uses the Iridium satellite communication. The concept is equal to the RTK network but can send data via E-mail or web services worldwide through satellites and is plug and play.

3.2.8. Subsystem

The advantages of having subsystems are that when an error or bug appears, the easiest and most effective way of problem searching is wanted for time-saving purposes. Therefore, dividing the system into multiple subsystems will help determine which part of the system does not affect the unwanted error by being able to test the various subsystems one after another and check them off. Also, if something starts a fire or components get burnt during the testing period, not every component will be in danger since everything is not interconnected as of yet.

3.3. Analysis of results

This section describes the experiments and tests that have been performed to reach the project's goals. In order to make a fully functioning system in this project, four significant tests will be performed. Firstly, a unit test to test if the actual component consumes power and works as intended. Secondly, a system test will be performed after a successful unit test. However, the unit and system test difference is to interconnect all the remaining components to the system. Thirdly, the integration test is performed with all of the CatFish

components to see how they interact and send the gathered data to the AWS database. The most important tests for this project are presented; the rest are in Appendix D. See figure 4 for a description of how the test will work.

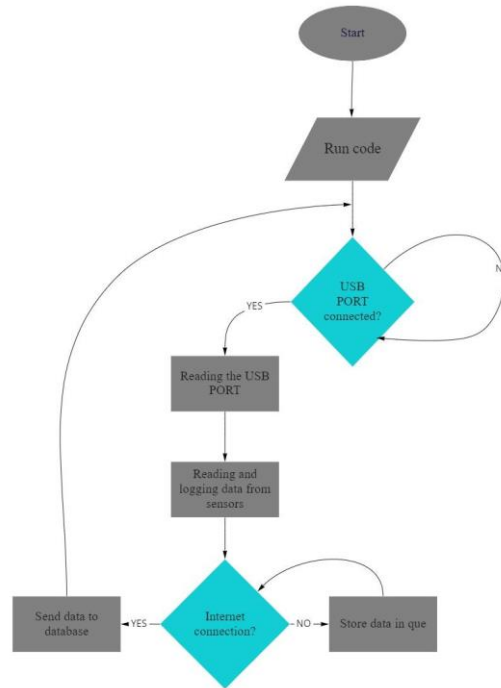


Figure 4. A UML diagram of how the unit and code test, system and integration test should be performed for the sensors with the microprocessor Raspberry Pi 4.

3.3.1. Testing the sensors and water quality parameters

To begin with, we need to perform a unit test on the optical sensor to see if the actual sensor is working when connected to the power supply. This provides the information that the microprocessor can see the component connected through a USB cable, and the Modbus RTU transmitter is in progress. After this test becomes successful, a code test is initialized to see that the required data can be logged and sent to the database. The microcomputer will be operational with the sensors running for at least two hours to verify that the desired time for them to be operational is achieved.

The expected results from the unit test are that the sensors do measure and log accurate values from the chosen parameters. Namely, pH, DO, Temperature, NO₃, CDOM, and turbidity. In addition, send these accurate values to the AWS database in JSON format with a $\pm 5\%$ accuracy.

In the end, a verification that that sensor behaves according to the datasheet will be made.



Figure 5. Setup for the test of the TpH-D sensor, in the aquarium with water from Nissan River. (Sensor A-MicroFlu Nano, B-Dissolved Oxygen, C-NICOplus, D-TpH-D).

3.3.2. Test of TpH-D sensor

The tests below are made with the TpH-D sensor [66] to measure the pH value in the test aquarium, which contains water from the Nissan river, for an as realistic result as possible. Since the water is identical and not many factors can change the pH in the aquarium, the value should be constant. See figure 5 for the test of the sensor.

According to the sensor's datasheet, seen in appendix D, figure 37, the sensor will be hoisted into the water and run for five minutes as a warmup time. Afterward, sampling initiates with a sampling time of $(\frac{1}{2})$ hertz; this is the minimum time interval for the sensor according to the datasheet. A mission for the project's requirements is a maximum of two hours of operation, so if the sensors are sampled with an interval of $(\frac{1}{30})$ hertz, approximately 30 seconds, a total of 240 samples are taken during that time. Nonetheless, 840 samples were measured for this test, more than three times the required samples.

Secondly, a helpful partner to this project that works in the chemical lab at Halmstad University as a professor measured the pH value before running the tests, so there was something to judge the measured accuracy. His measurement result was an 8.14 pH value with two decimals accuracy according to *equation 2*. These tests are constructed to calculate the resolution, accuracy, and the precision of the sensor and then compare it to the datasheet if it functions as it should.

3.3.3. Test of TpH-D sensor for resolution

After the samples are done, calculations are performed on the data to analyze and justify the actual value of resolution and find the noise for the TpH-D sensor. Firstly, we calculated the Bias of the pH value and the

difference between the measured pH and the professor's measured value. For this test, we summarized the Bias and calculated its mean. This will give us an indication of the resolution and where the noise is.

Here, (B) is the actual pH value that the professor measured, (N) is the total number of measurements taken, and (x_i) is every individual sensor measured value. According to *equation 10*, calculating the Bias is a method for estimations. Bias is used to find the error by subtracting each estimate (x_i) from the actual value (B) , summarizing all the errors, and dividing by the number of estimates (N) to achieve the Bias. The expected result for this test is to calculate Bias and resolution successfully and, with these values, get a better understanding of how many decimals should be sent to the database to save bytes.

$$Bias = \frac{1}{N} \left(\sum_{i=1}^N (x_i - B) \right) \quad (10)$$

$$Accuracy = \frac{1}{N} \left(\sum_{i=1}^N Bias_i \right) \quad (11)$$

3.3.4. Test of TpH-D OS of accuracy

As for the accuracy, we are reusing the calculations from section 3.3.3. *Equations 10* and *11* are used for calculating the sensor's accuracy.

The expected result from this test is to find the sensor's accuracy and compare it with the corresponding datasheet, seen in appendix D, figure 37.

3.3.5. Test of TpH-D OS of precision

Another essential calculation that should be taken into consideration is precision; namely, the standard deviation, variance, and mean, which are calculated with *Equations 12* and *13*. Where (μ) is the mean value of the measurements, (σ) is the standard deviation, (N) is the total number of measurements taken, and (x_i) is every individual sensor measured value.

The expected result is to compare the graphical views and calculated values with the sensor's datasheet.

$$\mu = \frac{1}{N} \left(\sum_{i=1}^N x_i \right) \quad (12)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (13)$$

4. Results

This section will present the results of all unit tests, the system test, and finally, the integration test. See figure 6, 7 for the finished prototype running an integration test.



Figure 6. Surface overview of the CatFish running integration test at Brottet in Halmstad with the finished prototype.



Figure 7. Underwater view of the Fish and winch collecting data while running integration test at Brottet in Halmstad with the finished prototype.

4.1. System overview

Figure 8 shows an overview of the final implemented interconnection with everything from hardware to software in this project. The entire project is also divided into several subsystems, which are also described and seen in appendix A, figures 21-25.

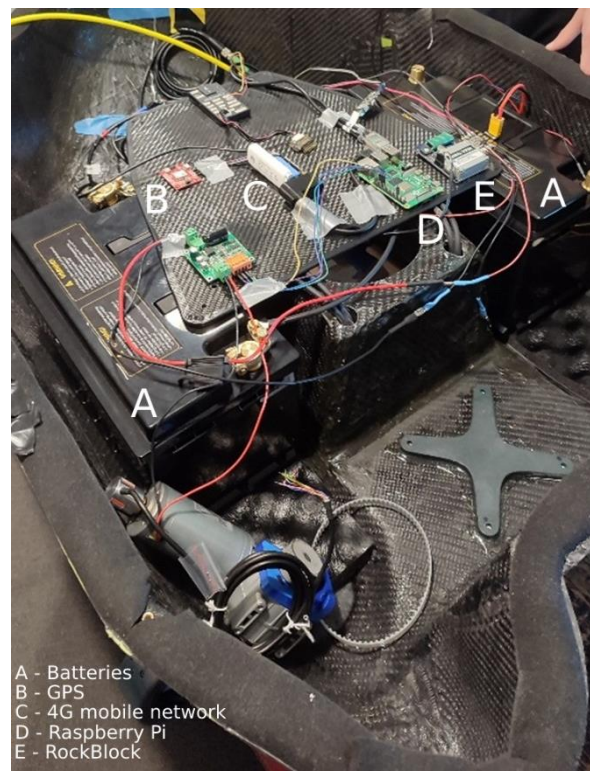


Figure 8. Overview of the system being assembly into the prototype. (A – batteries, B – GPS, C – 4G mobile network, D – Raspberry Pi and E – RockBlock).

4.2. Subsystem overview

As mentioned in the subsection above, the interconnected system was divided into several different subsystems to get a better overview of the system.

4.2.1 Power consumption

One subsystem was created to test the actual power consumption section of the project, consisting of the batteries, PDB (Power Distribution Board), solar panels, and their corresponding regulator, as seen in figure 21, appendix A. The battery is connected to the PDB, and it can distribute the battery's power in different doses, which in this project's case is +5V for the Raspberry Pi 4 and +12V for the sensors.

The total kWh can be calculated using *equation 4* and then divided from months to hours. That means it is possible to calculate the kWh for the prototype's two fully operational hours. As a result, the solar panel can recharge the batteries up to approximately 10 Wh per day, 0.83 Ah.

4.2.2 Collecting data

The interconnection between Raspberry Pi and the sensors consists of RS485 to USB converters. Furthermore, the sensors and the Raspberry Pi require power from the battery via the PDB. The RS485 cable's green and brown wires are interconnected with a resistor of 120ohm according to the datasheet since it is a non-multi-drop RS485 system. The code sends a master-slave request to the sensors, initializes readings, and gathers the wanted parameters to forward these data values to the AWS database, as seen in appendix A, figure 25. In addition, the code for the sensors can be found in appendix C figures 27, 28.

4.3. Summary of components

The criterium for Catfish is that it should be fully autonomous; therefore, this project must contain components detecting objects and obstacles in any way. As a result, this led to the decision of a camera, lidar, and sonars to visualize above and underneath surface surroundings. A power supply is needed for the CatFish to be operatable; hence batteries were chosen regarding the requirement to be fully operating for two hours, and solar panels plus a regulator to constantly charge the batteries if something happens and for the environmental aspects. A need for orientation and communication is also mandatory, consisting of a GPS for the location, RockBlock for the emergency messages, and a 4G mobile network for an internet connection. Finally, the Raspberry Pi and Jetson nano dev kit was chosen to run all these components because of the powerful data processing and the camera's high FPS compatibility, respectively. See table 6 for the components' summary, price, and reference.

Table 6. A table of the summary of the components that have been decided.

Type	Name	Name	Name	Name	Price	References
Microprocessor	Raspberry Pi 4	Jetson Nano			1049 sek [Raspberry Pi 4] 1165 sek [Jetson Nano]	[38] [36]
Sensors	NICO Plus	CDOM nanoFlu	TpH-D	DO	100 000 sek [NICO plus] 35 000 sek [COM nanoFlu] 7500 sek [TpH-D] 7500 sek [DO]	[59] [65] [66] [67]
Battery	LiFePO4				8490 sek x2 [LiFePO4]	[70]
Solar panels	SunWind Gotland				1310 sek [SunWind Gotland]	[82]
Regulator	Smart solar MPPT				1400 sek [Smart solar MPPT]	[83]
Sonar	Altimeter and Echosounder				3620 sek x2 [Altimeter and Echosounder]	[73]
Lidar	Lidar 360				1675 sek [Lidar 360]	[84]
Camera	PiCam 360				1500 sek [PiCam 360]	[69]
Universal Serial Bus hub	Hi-Speed USB Hub				700 sek x2 [Hi-Speed USB Hub]	[85]
Satellite communication /GPS	RockBlock	GPS-RTK Board			2700 sek [RockBlock] 2300 sek [GPS-RTK Board]	[81] [79]
Total price:					188 719 sek	

4.4 Result of sampling with OS TpH-D

Here the results according to the sampling test in section 3.3.1 are presented. The calculations and graphical view results will be compared to the sensor's datasheet, see appendix D, figure 37.

4.4.1 Calculations for resolution, accuracy and precision

The sampled data is measured from the test in section 3.3.2. Figure 9 shows the TpH-D's sampled data with a time interval of 28 minutes and 840 samples. In addition, figure 10 shows the data with mean and standard deviation, while figures 11 and 12 show histograms of the with standard deviation, which gives a more satisfactory overview of the sampled data.

By analyzing the graphical view in figure 10, it is detected that the sensor takes around 14 minutes to stabilize the values compared to the 5 minutes documented in the corresponding datasheet, more about this in section 5. However, when the settling time has passed, the sensor's accuracy and precision are exact compared to the professor's measured value.

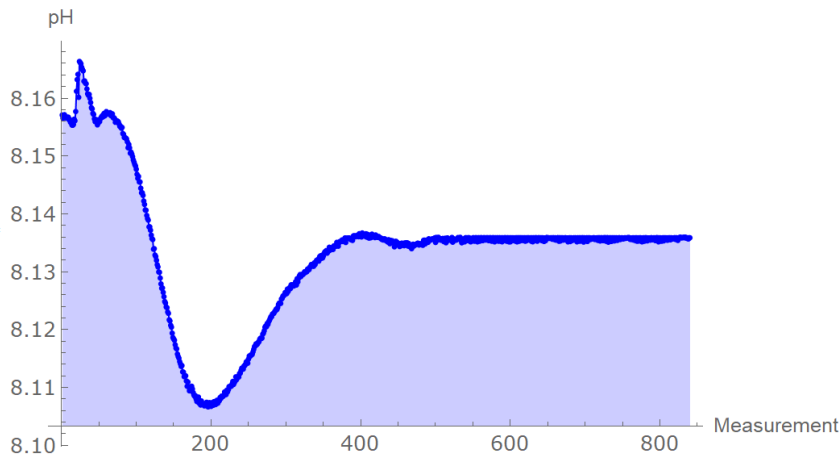


Figure 9. Graphical view of the sampled data from TpH-D sensor from the test with 840 measurements with a 5 minutes warmup time.

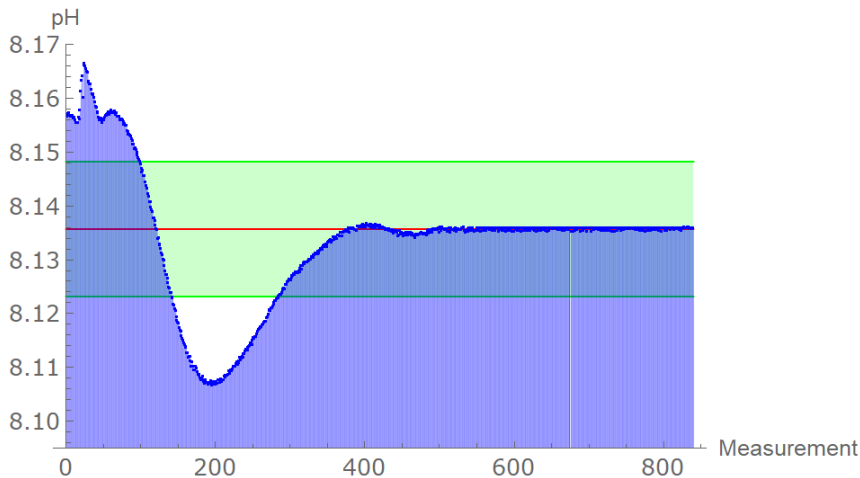


Figure 10. Graphical view of the sampled data from TpH-D sensor with mean and standard deviation from the test with 840 measurements with a 5 minutes warmup time. (Red line mean and green STD)

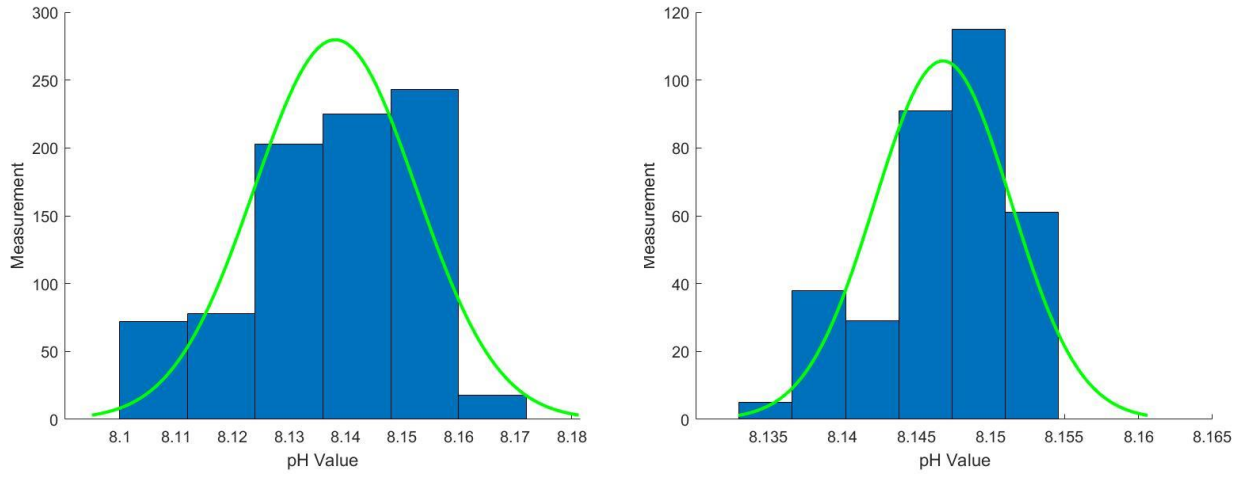


Figure 11. An overview of a histogram with 840 sampled data from the TpH-D sensor. The green is for the normal distribution fit of the pH values.

Figure 12. An overview of a histogram with 340 (from 500 to 840) sampled data from the TpH-D sensor. The green is for the normal distribution fit of the pH values.

4.4.2 Summarize from the test of TpH-D sensor

To get a more detailed overview of the comparison with our calculations and the sensor's datasheet and understand the behavior of the sensor, we divided it into two parts. Firstly with all measured samples, see figures 9 and 10. Secondly, from samples 500 to 840, see figures 13 and 14.

Moreover, according to *equation 10*, Bias was calculated to be able to use *equation 11* and get the accuracy of the sensor. In addition, *equations 12 and 13* are for calculating the variance and standard deviation, respectively.

For the first part, with all the sampled data, the pH mean value is 8.13369, pH variance 0.000149772, and pH standard deviation 0.0122381. The precision for our calculations is the variance compared to the datasheet's repeatability section. As a result, the difference between variance and the repeatability in pH is 0.0498502, de facto that our sampled data has greater precision than the datasheet. The calculations will help get an overview of the number of decimals significant for the water quality measurements. As for accuracy, taking the mean (μ) of the 840 data values and subtracting the actual value (B) gives an adequate overview of it according to the requirement specifications of $\pm 5\%$. The result from the test is ± 0.00630527 , in comparison to the datasheet, which has ± 0.06 .

Since the settling time of all the sampled data affects the resolution calculation, it was chosen to calculate the resolution for the data after the settling time was finished, namely between samples 500 and 840. For this second part, the pH mean value is 8.1357, pH variance 0.0000000641413, and pH standard deviation 0.000253261. From figure 13, it is graphically visible that the resolution is approximate 0.001 in pH value, but according to the datasheet, the actual value is 0.01. In summarization, according to the test and its corresponding

datasheet analysis, This project gets a higher precision and accuracy than the datasheet, but a lower resolution—more about this in section 6.

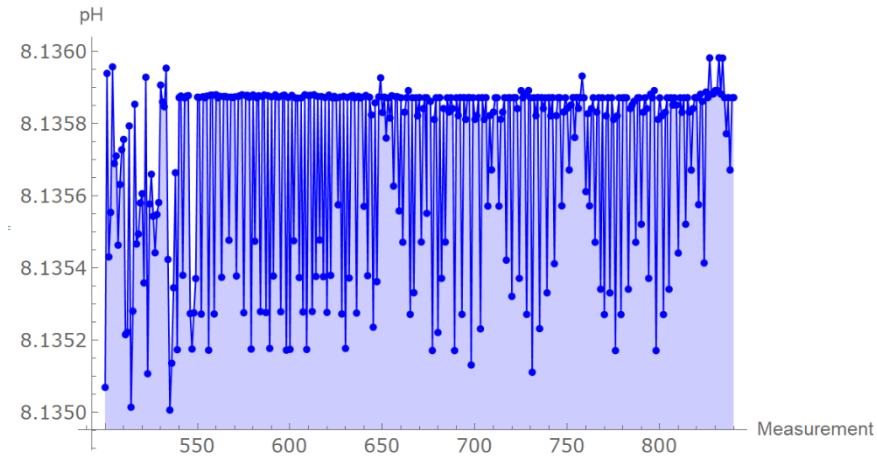


Figure 13. For accurate calculations and values, a graphical view of the sampled data (500 to 840 measurements) from the TpH-D sensor.

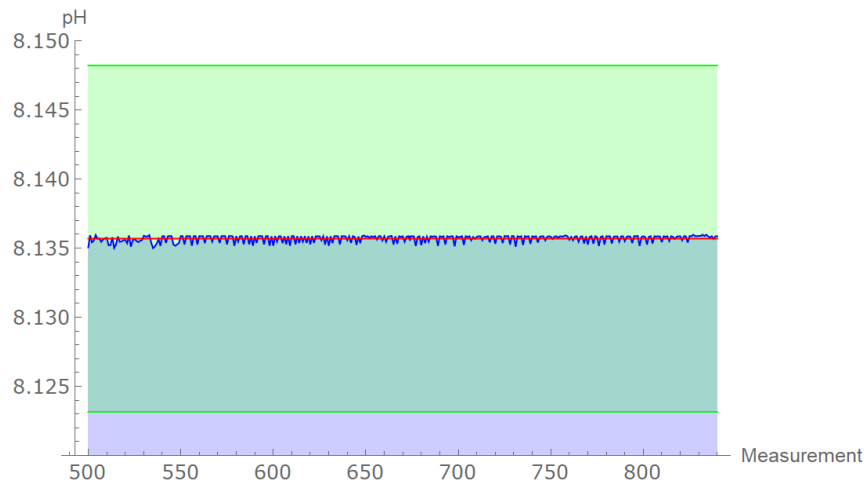


Figure 14. Graphical view of the sampled data from TpH-D sensor with mean and standard deviation. (blue sampled data, red mean and green standard deviation). Measurement numbers take from 500 to 840.

4.5. Result of data sent to the database with 4G communication

During this test, the components connected and operated are the sonar and GPS, and their data size does not ascend above one kilobyte. Neither does the whole interconnected system in total. According to *equation 1*, the data used when sending the data of the sonar and GPS should take approximately 0.0008 seconds, calculated with one kilobyte in file size. The data is sent to the AWS database in JSON format; see figure 15 for the result.

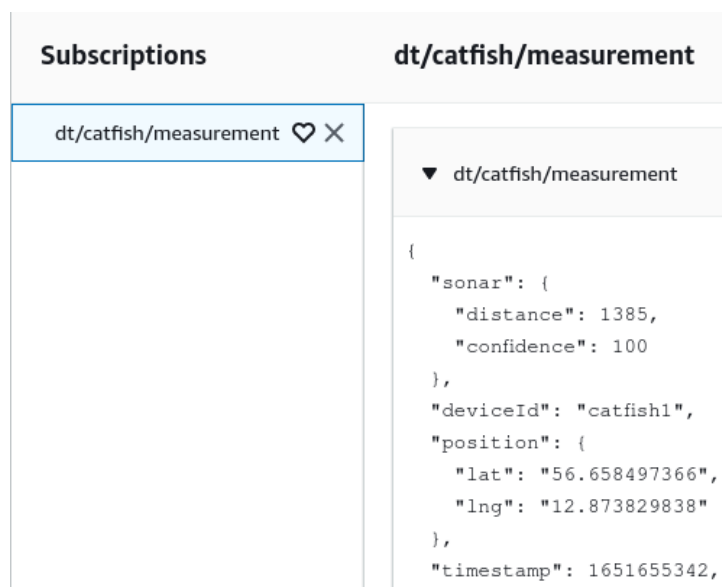


Figure 15. Result of sending data to the database AWS, from RockBlock, GPS and timestamp.

4.6. RockBlock/GPS emergency and coordinates message via satellites communication

As for the results of the GPS, when entering its fetched coordinates in google earth, the website showed the current and accurate position of the GPS component as predicted. Furthermore, *equations 7-9* were used to calculate the distance between the two points measured with the GPS, as seen in table 7. The distance from the calculations was *111,61* meters rounded, while Google earth's built-in function gives the distance *112,24* meters, as seen in figure 16. As a result, the GPS is accurate since the slight difference between the two distances can easily have been caused by disturbances or the error when setting out the two points manually in google earth. In addition, the GPS coordinates were successfully sent to the AWS database, as seen in figure 15. When it comes to the RockBlock results, it successfully sent an emergency message, as seen in appendix A, figure 26.

Table 7. Longitude and latitude for the two measured points.

Latitude	Longitude
56.663555	12.879074
56.663656	12.880906

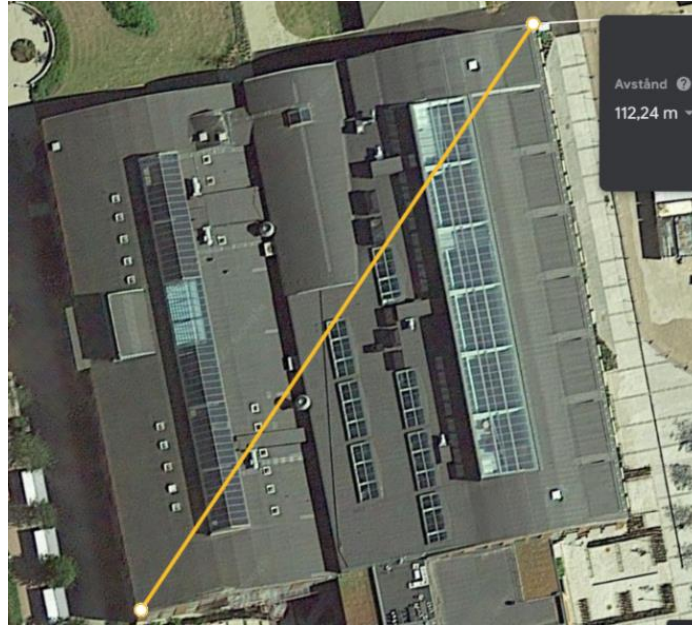


Figure 16. Picture of the test's measurement with Google earth's built-in function (The diagonal of the S-building on Halmstad University).

4.7. Sonar/Lidar result of depth and

The sonar result is presented in figure 15, where the sonar's measurement and the confidence data have been successfully sent to the database from the database's point of view. When it comes to the lidar, the test results of holding it in the air and letting it scan the room were terrible; a vision of the room's outline was barely visible. In conclusion, by looking at the test, the microprocessor and sonar have established communication, and the data is successfully gathered and sent to the AWS database, as it should. As for the sonar, we measured the temperature in the water to be approximately 20,5° Celsius with the TpH-D sensor. Therefore, calculating the velocity for sound underwater (v_{sw}) in the test aquarium using *equation 6*. As a result, *equation 14* is acquired.

$$v_{sw} = 1404,3 + 4,7 \times 20,5 - 0,04 \times 20,5^2 = 1483,84 \text{ m} \times \text{s}^{-1} \quad (14)$$

In addition, the distance was measured by hand and with the sonar, see figure 15, to acquire all parameters needed to use *equation 5.1*. Then break out and calculate the time elapsed for the emitted pulse to hit the object (Δt). By doing so, *equation 15* and *16* is generated and executed. As a result, the elapsed time for the emitted pulse to touch an object with the hand measured 1390 millimeters away is approximately 1,874 seconds. In addition, with the sonar's measured distance at 1385 millimeters, it was 1,867 seconds. To sum this up, the difference between the sonar's and ruler's measurements and the time to hit the object is five millimeters and seven milliseconds, respectively.

$$\Delta t_{ruler} = 2 \times \frac{d_o}{v_s} = 2 \times \frac{1390}{1483,84} \approx 1,874 \text{ seconds} \quad (15)$$

$$\Delta t_{sonar} = 2 \times \frac{d_o}{v_s} = 2 \times \frac{1385}{1483,84} \approx 1,867 \text{ seconds}$$

(16)

4.8. Integration test

As for the integration test, the prototype was carried out in a big swimming pool located in Halmstad. Since anything can happen during the first integration test, it should be more manageable and executed in a safe environment if something goes wrong.

The integration test result was successful, and by looking at figures 6 and 7, both the Cat and the Fish is seen in action. In figure 7, CatFish is in full operating mode with all the components inside the Cat and the two big sensors installed horizontally on the bottom side of the boat. In addition, figure 7 shows the Fish hoisted down with the TpH-D and DO sensors. Overall the sensors worked as they should, while the database gathered data from the sensors, sonar, and the GPS. Figures 8, 17, 18 shows the components before and after assembly into the prototype. See appendix D for more figures.

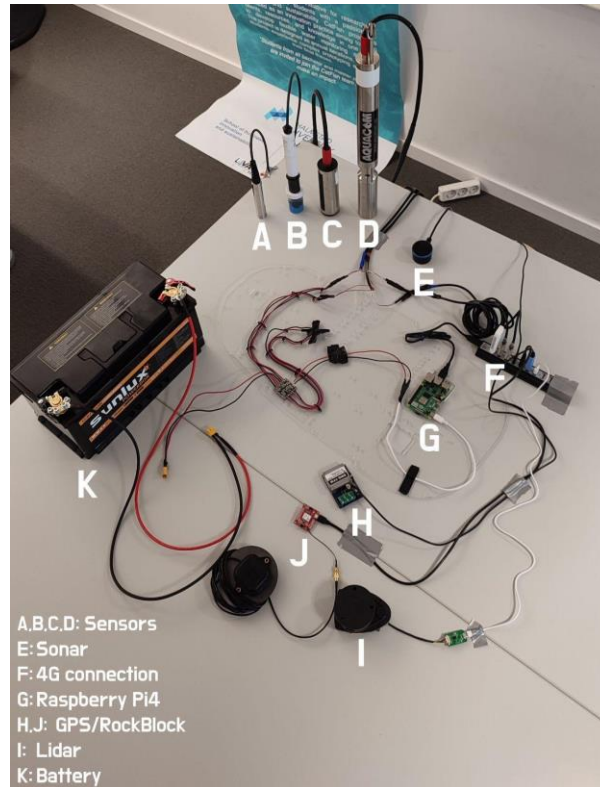


Figure 17. Overview of components. (A:Dissolved Oxygen, B:TpH-D, C:NanoFlu, D:NICOplus, E:Sonar, F:Mobile network, G: Raspberry Pi4, H:RockBlock, I:Lidar, J:GPS, K:Battery).

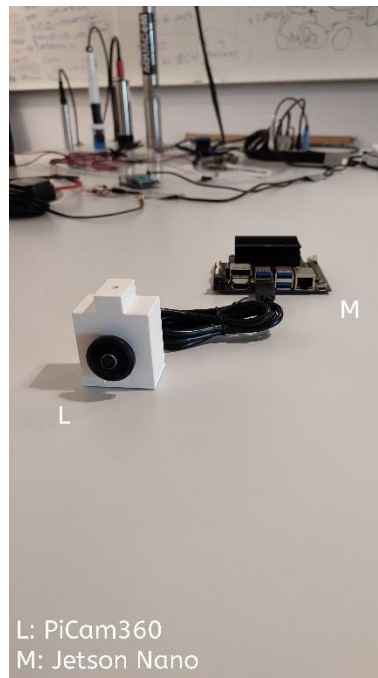


Figure 18. *Picture of Jetson Nano and PiCam360.*

5. Discussion

5.1. Results from the sensor test

According to the result from section 4.4, the settling time for the sensor was around 500 samples, which is more or less seven minutes in time. However, these seven minutes of settling time are taken after the five minutes warm-up time since that is the documented time in the sensor's datasheet.

Consequently, the actual warm-up time lands around twelve minutes instead of five, which must be considered. The solution is that we implement a code that compares the present measurement of pH with the previous one. If the difference between them is larger than the calculated standard deviation value, it does not send the data to the database. The code will run for the first seven minutes of an operation and then stop. On the other hand, there is a specific timeframe from the moment the coordinates are entered until the prototype measures the water quality, depending on how far the specified coordinates are from the current position. However, diagrams and calculations show that the accuracy is ± 0.00630527 pH, which is better than the ± 0.06 pH in the datasheet, see figures 9 and 10.

The minimal decimals must be at most three for the highest accurate measurements from the sensor test and corresponding calculations.

5.2. Lidar

When we began running our tests for the lidar, we could make it run as it should, getting the measurements and drawing up a 2D picture of the surroundings. However, it only worked every other time. Moreover, a decision was made to drop the lidar from this prototype and project. The information page on the lidar says that it works in an outdoor environment without sunlight. As a result of dropping the lidar, we did not continue with the remaining tests mentioned in appendix testing the lidar; therefore, the test of how the lidar reacts with sunlight and its laser reflections on the water surface remains unanswered.

The autonomous boat misses out on information about object detection on the surface without the lidar implementation; on the other hand, the 360-degree PiCam is still used for on-surface object detection.

5.3. Limitations of the project's solution

Firstly the camera and the lidar were not fully implemented by the project's deadline. Secondly, the design for the roof of the CatFish ended out completely flat with a 0° degree angle horizontally, which also limited us from being able to maximize energy from the solar panels.

5.4. Next version of CatFish

This section will discuss what components and technical improvements could be implemented for the next version of the CatFish prototype.

5.4.1. Improvement of communication

Instead of having a mobile internet connection through a USB dongle, a suggestion could be to use a 4G router to establish a more secure and stable connection than with a USB dongle, as this project did [13]. Alternatively, even use a mobile cellphone; these include an already programmed GPS, internet connection, and more sensors saving time, space, and money while giving the project the same amount of information.

5.4.2. Wind power instead or in addition to solar energy

By analyzing table 1, one project sticks out when using nature's movement to create power and electricity—namely, the ASAROME project [19], which is the only one having both solar panels and a wind turbine.

CatFish already have solar panels to charge the batteries; however, using a wind turbine would be of much help in addition to the solar panels. Especially since it is almost always windy when there are many clouds nearby preventing the sun from shining through, and in this way, the wind turbine and the solar panels would be complementary to each other.

5.4.3. Angle the solar panels to maximize energy

According to section 1.8.2, a requirement was to maximize the solar panels' energy. However, the project leader of CatFish decided to skip it; however, this is something that should be included in the next prototype. See figure 41 in Appendix E for an approximate solution to maximize the energy; this could lead to that CatFish being able to operate in the water for a longer time than the requirement specifications.

5.5. Social requirements for technical product development

This section will discuss ethical, economic, and environmental aspects. The autonomous drone, CatFish, is an excellent start to help the Halmstad municipality get an information-rich overview of the environmental and ecological situation. In the future, the plan is to start manufacturing the CatFish for other municipalities in Sweden with the same purpose.

Also, CatFish is driven by renewable energy, especially solar energy like articles [13, 15, 19, 53], which is a good aspect for future sustainability and the environment. However, ASAROME [19] also uses wind turbines; it is better to use two sources than one (wind and sun). On the contrary, the articles [6, 22, 16, 41] do not use renewable energy sources, and finally, the article [14] uses wave-powered energy.

An ethical aspect of the finished product is that it can improve the quality of life for many people and aquatic animals, not to mention the vegetation in the water. As the CatFish will be fully autonomous, we have to rely on machine learning for the aspect of safety. The Roboat USV [41] uses the same navigation system as CatFish. However, its purpose is transporting people across the canals, which is also a factor in improving the quality of human life instead of aquatic.

A comparison in an economic aspect is that we do not have a strict budget since this project got funding. The system for this project's roughly estimated cost landed at 230.000kr before the cooperation discounts and around 188.719kr after the discounts. This project would never be possible without the funding this project has. However, the project from this article [53] did have a strict budget and had to conclude parameters for measuring the water quality.

6. Conclusions

CatFish is an autonomous drone with an implementation of an IoT system for real-time water quality monitoring with 4G and satellite connections. In this thesis, the process of making the project's system successfully measure water quality for a sufficient amount of time is presented.

This section goes through the project's conclusions via the requirement specifications and answers the research questions.

6.1. How to measure water quality with sensors?

As a result, we were able to measure water quality with the help of optical sensors. Based on the literature review in chapter 2.8 and requirements from Halmstad municipality, we were able to find the appropriate parameters and gain the necessary knowledge about parameters for measuring water quality. To summarize, the parameters for measuring water quality are pH, DO, NO₃ and CDOM; the sensors used can be found in section 4.3, table 6.

6.2. What are the minimum number of parameters needed for measuring water quality?

Based on the literature review, see section 2.8, we justify that at least four parameters (pH, DO, NO₃ and CDOM) had to be measured. However, we added two more (Turbidity and temperature). The conclusion is that we could meet the requested parameters from Halmstad Municipality and fulfill the requirements specifications.

6.3. What is the most common sensor type to use, and how well can it measure the water quality?

We started by comparing similar projects to determine what kind of sensors they used and if they mentioned measurement accuracy—a comparison of the projects with a budget and those that did not were also performed. Based on the conclusions from similar projects, see 2.9, the result we came across was that the optical sensors are the most common and have the highest accuracy and precision. Furthermore, this result is strengthened by the test in 4.4, executed to calculate the accuracy, precision, and resolution of the TpH-D sensor. The conclusion was that the optical sensors were the most expensive but most used to measure water quality because of their exceptional accuracy and precision.

6.4. What are the best components to chose according to interconnection compatability, and effectiveness?

Finding effective and suitable components were done by comparing the requirement specifications from section 1.8 with similar projects and

analyzing which ones are being used, how often, and why it was chosen for that specific project. In addition, when deciding whether or not a particular component's ability to be interconnected with the other components in this system, comparing pins, wires, et cetera in datasheets gave the necessary information and answered that question.

Tests for the different subsystems were created, as seen in Appendix D, and the result is presented in sections 4.4, 4.5, 4.6, and 4.7.

As a result of the question, a working system was constructed with all the interconnected components. In conclusion, it was effective; however, improvements can be made, mentioned, and discussed in section 5.4.

6.5. The system

The system, see figure 6 that we have built together consists of four different sensors that can measure the following parameters: (pH, temperature, DO, NO₃, CDOM, and Turbidity). The system's components have been individually unit tested, and experiments in the aquarium have evaluated the TpH-D sensor's precision, accuracy, and resolution. Afterward, it was compared to the datasheet. The results concluded was accuracy is ± 0.006 , resolution 0.001 , and precision 0.0498502 . To summarize, the test result gave a better output than documented in the datasheet.

We successfully transferred data with a frequency of $(\frac{1}{30})$ hertz to the AWS database wirelessly in real-time, with a timestamp in JSON format. The system can handle minor communication errors by logging the data and entering it in a queue. As a result, the data is dequeuing to the AWS database when the communication is back.

The whole system consumes about $4.566A$, which means that it can run the whole system for at least two hours according to the requirement specifications. We consume approximately 4.75% of the whole energy resources available each hour.

The conclusion drawn from this project is that an improvement has been made compared to last year's prototype [6] to measure water quality with more parameters and precision. It has the full potential of becoming fully autonomous without remote control and with renewable energy in the near future.

7. References

- [1] Vattenmyndigheterna. Åtgärdsprogram för Västerhavets vattendistrikt 2021–2027;. [Accessed: 8 February 2022]. Available from: <https://www.vattenmyndigheterna.se>
- [2] Hansson A, Svensson Floer L. LEVA mellersta Halland – utveckling av åtgärdssamordning för jordbruket i Falkenbergs, Halmstads och Varbergs kommuner [Internet]. LEVA – Lokalt engagemang för vatten; 2021. Available from: <http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-45259>
- [3] Prutzer M. Lokal samverkan i vattenförvaltningen med vattenråden i fokus : Utvärdering av projektet Water Co-Governance i Sverige [Internet]. Göteborg: Havs- och vattenmyndigheten; 2020. (Havs- och vattenmyndighetens rapport). Available from: <http://urn.kb.se/resolve?urn=urn:nbn:se:havochvatten:diva-396>
- [4] Åkergren K. ”Se när drönaren skickas ut på vattnet – ska ta reda på hur Nissan mår.”. SVT [online]. 3 June 2022. [Accessed: 5 June 2022].
- [5] Blanco H, Esmailzadeh A, Forshlé M. CATFISH. Available from: <https://wiki.hh.se/caistr/images/1/14/CatFish.pdf>. [Accessed: 6 February 2022].
- [6] Nihl E, De Bruijkere E. An implementation of a real-time water quality monitoring IoT system with 4G connectivity : Catfish - Data and Sensors; 2021. Halmstad University. Available from: <http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-45009>
- [7] Nfaoui M, El-Hami K. Extracting the maximum energy from solar panels. Energy Reports. 2018;4:536-45.
- [8] Bodur M, Ermis M. Maximum power point tracking for low power photovoltaic solar panels. In: Proceedings of MELECON '94. Mediterranean Electrotechnical Conference; 1994. p. 758-61 vol.2.

- [9] Gabbar HA, Othman AM, Abdussami MR. Review of Battery Management Systems (BMS) Development and Industrial Standards. *Technologies*. 2021;9(2).
- [10] Rahimi-Eichi H, Ojha U, Baronti F, Chow MY. Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles. *IEEE Industrial Electronics Magazine*. 2013;7(2):4-16.
- [11] Uotila T. Designing scalable microservices: Case: AWS with Python. 2019.
- [12] Chen Y, Han D. Water quality monitoring in smart city: A pilot project. *Automation in Construction*. 2018;89:307-16.
- [13] Alves JC, Cruz NA. FASt - an autonomous sailing platform for oceanographic missions. In: *OCEANS 2008*; 2008. p. 1-7.
- [14] Sakagami N, Terao Y. Development of a measurement and autonomous control system for wave-powered boats. In: *2012 Oceans - Yeosu*; 2012. p. 1-6.
- [15] Bogue R. Robots for monitoring the environment. *Industrial Robot: An International Journal*. 2011.
- [16] Melo M, Mota F, Albuquerque V, Alexandria A. Development of a Robotic Airboat for Online Water Quality Monitoring in Lakes. *Robotics*. 2019;8(1).
- [17] Castendyk D, Hill B, Filiatreault P, Straight B, Alangari A, Cote P, et al. Experiences with autonomous sampling of pit lakes in North America using drone aircraft and drone boats. *Mine Water–Risk to Opportunity* Tshwane University of Technology, Pretoria, South Africa. 2018:1036-42.
- [18] Degel C, Peter L, Schmieger M, Hewener H, Bost W, Ehrhardt M, et al. HydroCrawler – An innovative autonomous surface vehicle for high precision underwater measurements. In: *OCEANS 2019 - Marseille*; 2019. p. 1-7.

- [19] Plumet F, Pêtrès C, Romero-Ramirez MA, Gas B, Ieng SH. Toward an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*. 2015;40(2):397-407.
- [20] Dunbabin M. In: Wettergreen DS, Barfoot TD, editors. *Autonomous Greenhouse Gas Sampling Using Multiple Robotic Boats*. Cham: Springer International Publishing; 2016. p. 17-30.
- [21] Silva MF, Friebe A, Malheiro B, Guedes P, Ferreira P, Waller M. Rigid wing sailboats: A state of the art survey. *Ocean Engineering*. 2019;187:106150.
- [22] Wang W, Gheneti B, Mateos LA, Duarte F, Ratti C, Rus D. Roboat: An Autonomous Surface Vehicle for Urban Waterways. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2019. p. 6340-7.
- [23] Wang W, Shan T, Leoni P, Fernández-Gutiérrez D, Meyers D, Ratti C, et al. Roboat II: A Novel Autonomous Surface Vessel for Urban Environments. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2020. p. 1740-7.
- [24] Evans AE, Mateo-Sagasta J, Qadir M, Boelee E, Ippolito A. Agricultural water pollution: key knowledge gaps and research needs. *Current Opinion in Environmental Sustainability*. 2019;36:20-7. *Environmental Change Assessment*.
- [25] Sung WT, Fadillah FN. Water Quality Monitoring Using Physio Chemical Sensors. In: 2020 International Symposium on Computer, Consumer and Control (IS3C); 2020. p. 440-3.
- [26] PNQA. (n.d.). *Portal da qualidade das águas [online]*. NWA. [Retrieved: May 11 2022]. Available from: <http://portalpnqa.ana.gov.br/rede-nacional-introducao.aspx>

- [27] Bhardwaj J, Gupta KK, Gupta R. A review of emerging trends on water quality measurement sensors. In: 2015 International Conference on Technologies for Sustainable Development (ICTSD); 2015. p. 1-6.
- [28] Regan F. Sensors — Overview. In: Worsfold P, Poole C, Townshend A, Miró M, editors. *Encyclopedia of Analytical Science (Third Edition)*. third edition ed. Oxford: Academic Press; 2019. p. 172-8.
- [29] Wang Xd, Wolfbeis OS. Optical methods for sensing and imaging oxygen: materials, spectroscopies and applications. *Chem Soc Rev*. 2014;43:3666-761.
- [30] Wencel D, MacCraith BD, McDonagh C. High performance optical ratiometric sol–gel-based pH sensor. *Sensors and Actuators B: Chemical*. 2009;139(1):208-13. EUROPT(R)ODE IX Proceedings of the 9th European Conference on Optical Chemical Sensors and Biosensors.
- [31] Markle DR, Wessling RA, Kolehmainen DJ. Optical systems and methods for ratiometric measurement of blood glucose concentration. Google Patents; 2014. US Patent 8,838,195.
- [32] Morales Y, Tsubouchi T. DGPS, RTK-GPS and StarFire DGPS Performance Under Tree Shading Environments. In: 2007 IEEE International Conference on Integration Technology; 2007. p. 519-24.
- [33] Leung KT, Whidborne JF, Purdy D, Dunoyer A. A review of ground vehicle dynamic state estimations utilising GPS/INS. *Vehicle System Dynamics*. 2011;49(1-2):29-58.
- [34] Arduino. Arduino Uno Rev3. Arduino Official Store [online]. [Retrieved: February 5 2022]. Available from: <http://store.arduino.cc/products/arduino-uno-rev3>
- [35] ESPRESSIF. ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems [online]. [Retrieved: February 5 2022]. Available from: <https://www.espressif.com/en/products/socs/esp32>

[36] Nvidia. Jetson Nano Developer Kit [online]. [Retrieved: March 26 2022]. Available from: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>

[37] Nvidia. Jetson Nano. Jetson Nano [online]. [Retrieved: May 11 2022]. Available from: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>

[38] Raspberry Pi datasheet, Raspberry Pi 4 Model B [online], 21/06/2019. Available from: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>

[39] Salih TA, Gh MB. A novel Face Recognition System based on Jetson Nano developer kit. IOP Conference Series: Materials Science and Engineering. 2020 nov;928(3):032051.

[40] Sauzé C, Neal M. Design considerations for sailing robots performing long term autonomous oceanography. In: Proceedings of the International Robotic Sailing Conference; 2008. p. 21-9.

[41] Wang W, Gheneti B, Mateos LA, Duarte F, Ratti C, Rus D. Roboat: An Autonomous Surface Vehicle for Urban Waterways. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2019. p. 6340-7.

[42] Lebkowski A. Temperature, overcharge and short-circuit studies of batteries used in electric vehicles. Przegląd Elektrotechniczny. 2017;1(5):69-75.

[43] Patel S, Shah V, Kansara M. Comparative Study of 2G, 3G and 4G. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2018;3(3):1962-4.

[44] Burke PJ. 4G Antipode: Remote Control of a Ground Vehicle From Around the World. IEEE Journal on Miniaturization for Air and Space Systems. 2020;1(3):150-3.

- [45] Caimi FM. Antenna design challenges for 4G. IEEE Wireless Communications. 2011;18(6):4-5.
- [46] Ramadan DN, Hadiyoso S, Sakti AR. Internet of Things: Roboboat for Water Area Monitoring using 4G network and Google Firebase. In: 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIIT); 2019. p. 234-7.
- [47] Ma Z. Experimental evaluation of mobile phone sensors. IET Conference Proceedings. 2013 January;49-9(1).
- [48] Upload Time Calculator - Calculator Academy [Internet]. Calculator Academy. 2021 [cited 24 May 2022]. Available from: <https://calculator.academy/upload-time-calculator/>
- [49] Bates H, Pierce M, Benter A. Real-Time Environmental Monitoring for Aquaculture Using a LoRaWAN-Based IoT Sensor Network. Sensors. 2021;21(23).
- [50] Chen K, Chen H, Zhou C, Huang Y, Qi X, Shen R, et al. Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data. Water Research. 2020;171:115454.
- [51] Dodds WK, Smith VH. Nitrogen, phosphorus, and eutrophication in streams. Inland Waters. 2016;6(2):155-64. Available from: <https://www.tandfonline.com/doi/abs/10.5268/IW-6.2.909>.
- [52] Summers JK. Water Quality: Science, Assessments and Policy. BoD Books on Demand; 2020.
- [53] Dunbabin M, Grinham A, Udy J. An autonomous surface vehicle for water quality monitoring. In: Australasian conference on robotics and automation (ACRA). Citeseer; 2009. p. 2-4.

- [54] Lambrou TP, Panayiotou CG, Anastasiou CC. A low-cost system for real time monitoring and assessment of potable water quality at consumer sites. In: SENSORS, 2012 IEEE; 2012. p. 1-4.
- [55] Cloete NA, Malekian R, Nair L. Design of Smart Sensors for Real-Time Water Quality Monitoring. IEEE Access. 2016;4:3975-90.
- [56] Omer NH. Water Quality Parameters. In: Summers K, editor. Water Quality. Rijeka: IntechOpen; 2019.
- [57] YSI. Multiparameter Sondes for Water Quality Monitoring YSI Sonde. [Retrieved: January 31 2022]. Available from: <https://www.ysi.com/products/multiparameter-sondes>
- [58] Castellini A, Bloisi D, Blum J, Masillo F, Farinelli A. Multivariate sensor signals collected by aquatic drones involved in water monitoring: A complete dataset. Data in Brief. 2020;30:105436.
- [59] TriOS. NICO plus. TriOS Mess- Und Datentechnik. [Retrieved: January 31 2022]. Available from: <https://www.trios.de/en/nico-plus.html>
- [60] How Do Temperature Sensors Work? | Atlas Scientific [Internet]. Atlas Scientific. 2021 [cited 13 April 2022]. Available from: <https://atlas-scientific.com/blog/how-do-temperature-sensors-work/>
- [61] Pasquesi A. How to Calculate PH of Buffer Solutions [Internet]. Sciencing. 2022 [cited 26 May 2022]. Available from: <https://sciencing.com/calculate-ph-buffer-solutions-5976293.html>
- [62] pH Measurement [Internet]. allaboutcircuits. 2022 [cited 28 May 2022]. Available from: <https://www.allaboutcircuits.com/textbook/direct-current/chpt-9/ph-measurement/>
- [63] Percent Saturation of Dissolved Oxygen. 1st ed. 2022. Available from: <https://dep.wv.gov/WWE/getinvolved/sos/Documents/SOSKit/DOSaturation.pdf>

[64] Renke. Water treatment sensors [online]. [Retrieved: January 30 2022]. Available from: <https://www.renkeer.com/water-treatment-sensors/>

[65] TriOS. NanoFlu. TriOS Mess- Und Datentechnik [online]. [Retrieved: January 31 2022]. Available from: <https://www.trios.de/en/nanoflu.html>

[66] TriOS. TpH-D. TriOS Mess- Und Datentechnik [online]. [Retrieved: January 31 2022]. Available from: <https://www.trios.de/en/tph-d.html>

[67] TriOS. DO. TriOS Mess- Und Datentechnik [online]. [Retrieved: January 31 2022]. Available from <https://www.trios.de/en/do.html>

[68] Duester L, Blohm W, Arndt J, Floury P, Degel C, Meyer A, et al. The Chemical Monitoring Station of the Future - Abstracts; 2021.

[69] PICAM360. PICAM360 [online]. [Retrieved: March 16 2022]. Available from: <https://store.picam360.com>

[70] Sunlux. LiFePO4 batteri 12.8V/96Ah - Sunlux® Powerbox [online]. [Retrieved: February 14 2022]. Available from <https://sunlux.se>

[71] Osman N. Calculating the Kilowatt Hours Your Solar Panels Produce (Solar Panel Output) - Understand Solar [Internet]. Understand Solar. 2015 [cited 1 June 2022]. Available from: <https://understandsolar.com/calculating-kilowatt-hours-solar-panels-produce/>

[72] Neves G, Ruiz M, Fontinele J, Oliveira L. Rotated object detection with forward-looking sonar in underwater applications. Expert Systems with Applications. 2020;140:112870.

[73] Blue Robotics. Ping Sonar Altimeter and Echosounder [online]. [Retrieved: February 17 2022]. Available from: <https://bluerobotics.com/store/sensors-sonars-cameras/sonar/ping-sonar-r2-rp/>

- [74] Ping Echosounder Sonar User Manual [Internet]. Blue Robotics. 2019 [cited 29 May 2022]. Available from: <https://bluerobotics.com/learn/ping-sonar-technical-guide/>
- [75] MacIsaac D, Hämäläinen A. Physics and technical characteristics of ultrasonic sonar systems. *The Physics Teacher*. 2002;40(1):39-46.
- [76] Lubbers J, Graaff R. A simple and accurate formula for the sound velocity in water. *Ultrasound in Medicine Biology*. 1998;24(7):1065-8.
- [77] Maranda BH. In: Havelock D, Kuwano S, Vorländer M, editors. *Passive Sonar*. New York, NY: Springer New York; 2008. p. 1757-81.
- [78] Real-Time Kinematic (RTK) [Internet]. Hexagon. 2015 [cited 1 June 2022]. Available from: <https://novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/real-time-kinematic-rtk>
- [79] SparkFun Electronics. SparkFun GPS-RTK Board - NEO-M8P-2 (Qwiic). GPS-15005 - SparkFun Electronics [online]. [Retrieved: March 11 2022]. Available from: <https://www.sparkfun.com/products/15005>
- [80] Winarno E, Hadikurniawati W, Rosso RN. Location based service for presence system using haversine method. In: 2017 International Conference on Innovative and Creative Information Technology (ICITech); 2017. p. 1-4.
- [81] SparkFun Electronics. RockBLOCK Mk2 - Iridium SatComm Module. WRL-13745 - SparkFun Electronics [online]. [Retrieved: March 11 2022]. Available from: <https://www.sparkfun.com/products/13745>
- [82] Elkatalogen. Sunwind Solpanel Gotland Long 60W 12V [online]. [Retrieved: April 2 2022]. Available from: <https://www.elkatalogen.se/produkter/sunwind-solpanel-gotland-long-60w-12v.html>

[83] Bauhaus. Solcellsregulator Victron Energy Smartsolar MPPT 75/10 10A [online]. [Retrieved: April 2 2022]. Available from: <https://www.bauhaus.se>

[84] HiTechChain. RPLiDAR A1M8 360 Degree Laser Scanner Kit - 12M Range [online]. [Retrieved: February 17 2022] Available from: <https://hitechchain.se/iot/rplidar-a1m8-360-degree-laser-scanner-kit-6m-range>

[85] Dustin Home [online]. Manhattan Hi-Speed USB Desktop Hub USB Hub. [Retrieved: March 29 2022]. Available from: <https://www.dustinhome.se>

[86] Sunchallenger [online] SunChallenger - The Autonomous Solar-Powered Vessel. [Retrieved: May 5 2022]. Available from: <https://sunchallenger.com/>

8. Appendices

A System

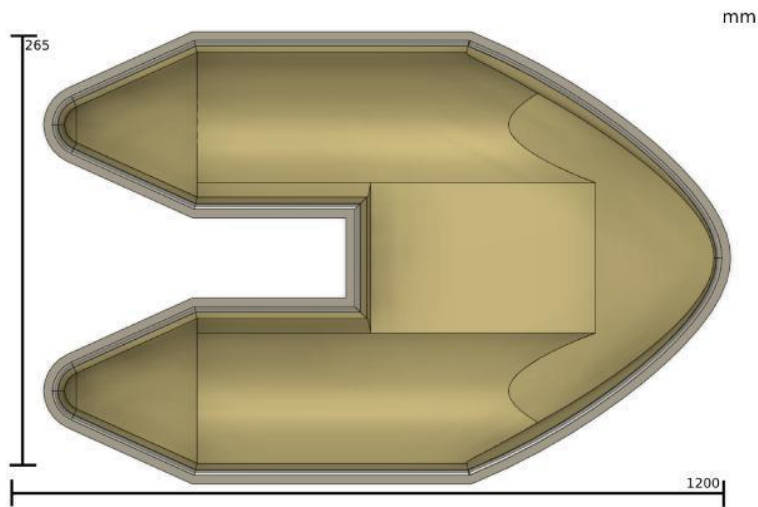


Figure 19. The visualization of the prototype, by the shape and measurements.

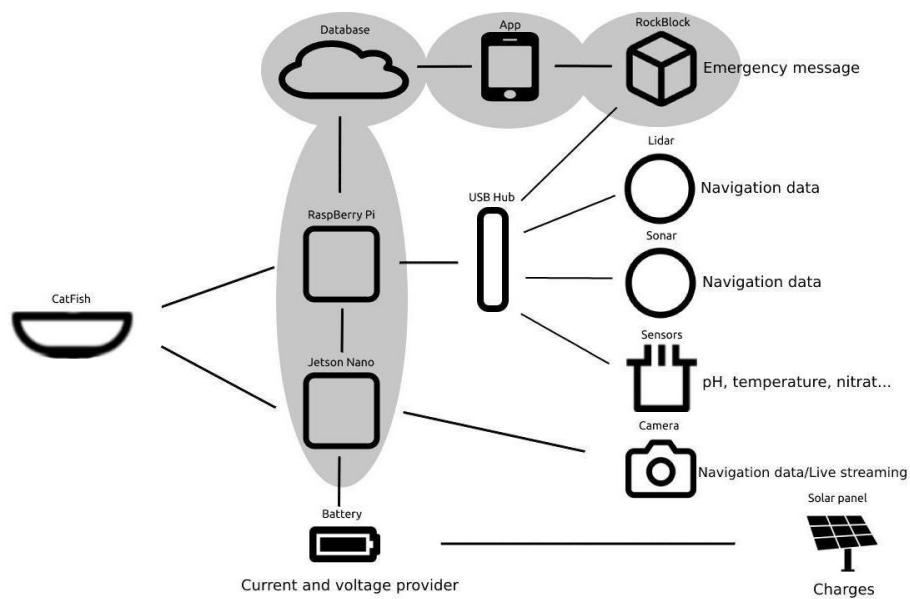


Figure 20. System overview of the whole system for embedded system group.

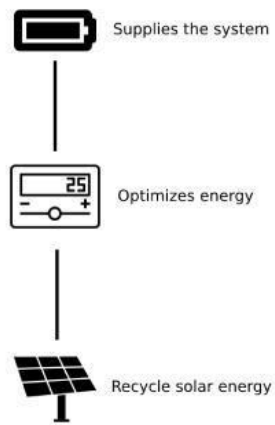


Figure 21. Subsystem of battery, regulator and solar panels.

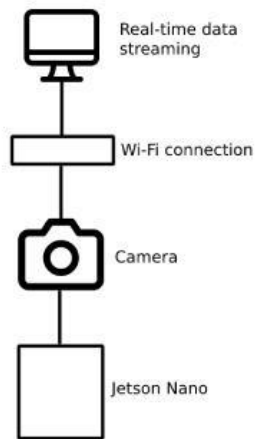


Figure 22. Subsystem overview of Jetson nano, camera, modem and live broadcast.

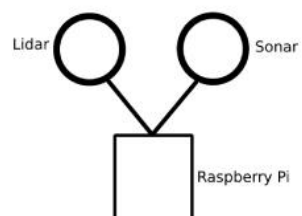


Figure 23. A subsystem overview of the navigation, namely lidar, sonar and Raspberry.

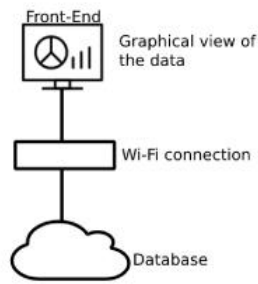


Figure 24. A subsystem overview of the data to graphical view.

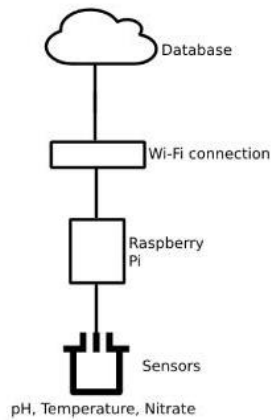


Figure 25. A subsystem overview of the microprocessor to the sensors.

Received At (UTC)	04/Mar/2022 15:01:31
Device	RockBLOCK 206876
Direction	↗ MO (Transfer OK)
Message Size	15 bytes (1 credit)

0000: 54 65 73 74 69 6e 67 20 2f 2f 6a 6f 68 61 6e |Testing //johan

Copy

Plain Text	Testing //johan
Status	✓
Location	
Approx Lat/Lng:	056° 39.731N 012° 53.808E 4KM ?

Delivery Status

Address	Last Attempt / Delivered At (UTC)	Status
alireza.esmaeilzadeh@hh.se	04/Mar/2022 15:01:36	✓
johanelfing@gmail.com	04/Mar/2022 15:01:36	✓
martin.rabaud@epitech.eu	04/Mar/2022 15:01:36	✓

Delivery Log

Attempted At	Address	Duration	Result
04/Mar/2022 15:01:36	martin.rabaud@epitech.eu	3 ms	Delivery OK
04/Mar/2022 15:01:36	johanelfing@gmail.com	10 ms	Delivery OK

Figure 26 An overview of the AWS database where the sonar and GPS data were sent in real-time. The format of the data in JSON

B Tables

Table 8. Similar autonomy boats within the specifications of our project. Adopted from [17].

Autonomous Boat	Beagle-B	ALANIS	ARC	Unmanned Airboat	Hydrus	USVe	Vector
Electric propulsion	X	X		X	X	X	X
Wind propulsion			X				
WiFi communication		X		X	X		X
Radio communication	X	X				X	
Camera				X		X	
pH measurement				X	X	X	X
OD measurement				X		X	X
CE measurement				X		X	
Turbidity measurement				X	X	X	
Temperature measurement				X	X	X	X
Depth measurement				X		X	
Chlorophyll NO ₂ concentration measurement				X			
GPS	X	X	X	X	X	X	X
IMU	X	X	X	X	X	X	X
Solar panel	X						
Batteries	X	X	X	X	X	X	X
Fuel		X		X			
Wind Sensor	X	X	X				

Table 9. Table of parameters adopted from [49].

Sensor	Level
TDS	< 300 – 600 mg/liter
Turbidity	0 – 2.4 NTU
pH	6.5 – 8.5
Conductivity	0 – 2500 μ S/cm
Temperature	20 – 25°C

Table 10. Table of parameters adopted from [50].

Parameter	Units	Quality Range	Meas. Cost	
1	Turbidity	NTU	0–5	Medium
2	Free Residual Chlorine	mg/L	0.2–2	High
3	ORP	mV	650–800	Low
4	Nitrates	mg/L	<10	High
5	Temperature	°C	–	Low
6	pH	PH	6.5–8.5	Low
7	Electrical Conductivity	μ S. μ S/cm	500–1000	Low
8	Dissolved Oxygen	mg/L	–	Medium

Table 11. A comparison for the Energy density, number of cycles before hitting 80%, discharge current, nominal voltage, and the working temperature for nine different battery types. Adopted from [42].

Type of battery	Energy density [Wh/kg]	Charge/Discharge current[C]	Nominal Voltage [V]	No of cycles [SOH 80%]	Working temp [C°]
Lead-Acid	35 ÷ 50	0.1 / 2	2.1	600	-20° to 40°
Ni-Cd	50 ÷ 80	1 / 15	1.2	500	-20° to 50°
Ni-MH	50 ÷ 100	1 / 5	1.2	800	-20° to 50°
Na-NiCl ₂	90 ÷ 110	1 / 2	2.6	1,500	245° to 350°
LiFePO ₄	90 ÷ 120	5 / 30	3.2	3,000	-20° to 60°
Li-PO	130 ÷ 220	2 / 25	3.7	500	-20° to 60°
Li-ION	160 ÷ 200	5 / 30	3.6	1,000	-20° to 50°
LTO	70 ÷ 80	5 / 20	2.4	20,000	-25° to 55°

Table 12. A summary of the comparison for the Energy density, number of cycles before hitting 80%, and the working temperature for the two most used battery types in table 11, plus LiFe-PO 4. Adopted from Table 11.

Type of battery	Energy density [Wh/kg]	No. Of Cycles [80% SOH (state-of-health)]	Working Temp. [C°]
Lead-Acid	0.7	600	-20° to 40°
LiFePO ₄	0.75	3,500	-20° to 60°
Li-PO	0.59	500	-20° to 60°

C Code fragment

```

while 1:
    try:

        status = rb.satellite_transfer()
        #Not sure if needed this
        if myPing.initialize() is False:
            print('Failed to initialize Ping!')
            exit(1)

        #time.sleep(3)

        doInstrument.write_register(1, 31)
        doName = doInstrument.read_string(3344, 16)

        ph = phInstrument.read_float(PH_REGISTER)
        temperature = phInstrument.read_float(TEMP_REGISTER)
        doTemperature = doInstrument.read_float(DO_TEMP_REGISTER)
        domql = doInstrument.read_float(DO_MQL_REGISTER)
        doppm = doInstrument.read_float(DO_PPM_REGISTER)

        ct = datetime.datetime.now()
        date = ct.isoformat()
        jsonDateTime = json.dumps(date)

        # Only for printing the results
        PHParameters = '{"Current time for Tph-D Sensor": '+jsonDateTime+ '\n'
        + ', "Temperature Degree": ' + \
        str(temperature)+'', "PH-value": '+str(ph)+'}'
        DOParameters = '{"Current time for DO Sensor": '+jsonDateTime+ ', "Dissolved Oxygen temperature Degree": ' + \
        str(doTemperature)+'', "Dissolved Oxygen MQL": ' + \
        str(domql)+'', "Dissolved Oxygen PPM": '+str(doppm)+'}'

        json_object = json.loads(PHParameters)
        json_obj2 = json.loads(DOParameters)
        #json_obj3 = json.loads(distancePara)

        dataToSend = {}

        # The array for Cat to send to database
        CatSensorsDataToSend = {

            "trios nico plus sensor": {
                "nitrate": 133.7,
                "turbidity": 133.7
            },
            "trios edon sensor": {
                "CDOM [µg/L]": 133.7,
                "chlorophyll [µg/L]": 133.7,
                "phycocyanin [µg/L]": 133.7,
                "rhodamine [µg/L]": 133.7,
                "fluorescein [µg/L]": 133.7
            }
        }

        # The array for Cat to send to database
        FishSensorsDataToSend = {

            "trios tph-d sensor": {
                "temperature [Celsius]": temperature,
                "ph-value [pH]": ph
            },
            "trios dissolved oxygen sensor": {
                "temperature [Celsius]": doTemperature,
                "Dissolved oxygen [mg/L]": domql,
                "Dissolved oxygen [PPM]": doppm
            }
        }

        #nested array for sending it to the database
        dataToSend['sonar'] = myPing.get_distance_simple()
        dataToSend['timestamp'] = int(time.time() * 1000)
        dataToSend['cat'] = CatSensorsDataToSend
        dataToSend['fish'] = FishSensorsDataToSend

        messagetoDump = json.dumps(dataToSend)

        mqttClient.publish("measurements", messagetoDump, 1)

        #printing the results of json format
        print(json_object)
        print(json_obj2)
        # print(json_obj3)

        #Only for print the measurements of sonar
        getDistance()

        time.sleep(5)
    except:
        print("Could not read, retrying")
        #serialcmd = input("serial command: ")
        #serLoRa.write(serialcmd.encode())
        #serLoRa.read_all
        retry = 0
        while status[0] > 8:
            time.sleep(10)
            status = rb.satellite_transfer()
            print(retry, status)
            retry += 1
        continue

```

Figure 27. Python code for the sensors.

```

# The array for Cat to send to database
FishSensorsDataToSend = {

    "trios tph-d sensor": {
        "temperature [Celsius] ": temperature,
        "ph-value [pH]": ph
    },
    "trios dissolved oxygen sensor": {
        "temperature [Celsius]": doTemperature,
        "Dissolved oxygen [mg/L]": domql,
        "Dissolved oxygen [PPM]": doppm
    }
}

#nested array for sending it to the database
dataToSend['sonar'] = myPing.get_distance_simple()
dataToSend['timestamp'] = int(time.time() * 1000)
dataToSend['cat'] = CatSensorsDataToSend
dataToSend['fish'] = FishSensorsDataToSend

messagetoDump = json.dumps(dataToSend)

mqttClient.publish("measurements", messagetoDump, 1)

#printing the results of json format
print(json_object)
print(json_obj2)
# print(json_obj3)

#Only for print the measurements of sonar
getDistance()

time.sleep(5)
except:
    print("Could not read, retrying")
    #serialcmd = input("serial command: ")
    #serLoRa.write(serialcmd.encode())
    #serLoRa.read_all
    retry = 0
    while status[0] > 8:
        time.sleep(10)
        status = rb.satellite_transfer()
        print(retry, status)
        retry += 1
    continue

```

Figure 28. Python code for the sensors.

Figure 29. Python code for the lidar.

Figure 30. Python code for the RockBlock.

Figure 31. Python code for the GPSRTK.

D Tests

Testing the microprocessor with communication

The power consumption is essential to overview both microprocessors; a unit test is needed for each of them. We need to check that the microprocessors can find the attached components for communication. We also need to check that the bandwidth for the 4G communication is strong enough to send the data to the AWS database. That's why a code test is needed to perform these tests.

The reason why performing these tests is to see the data on the AWS database, get an overview of the power consumption, and communicate between the microprocessors and the components.

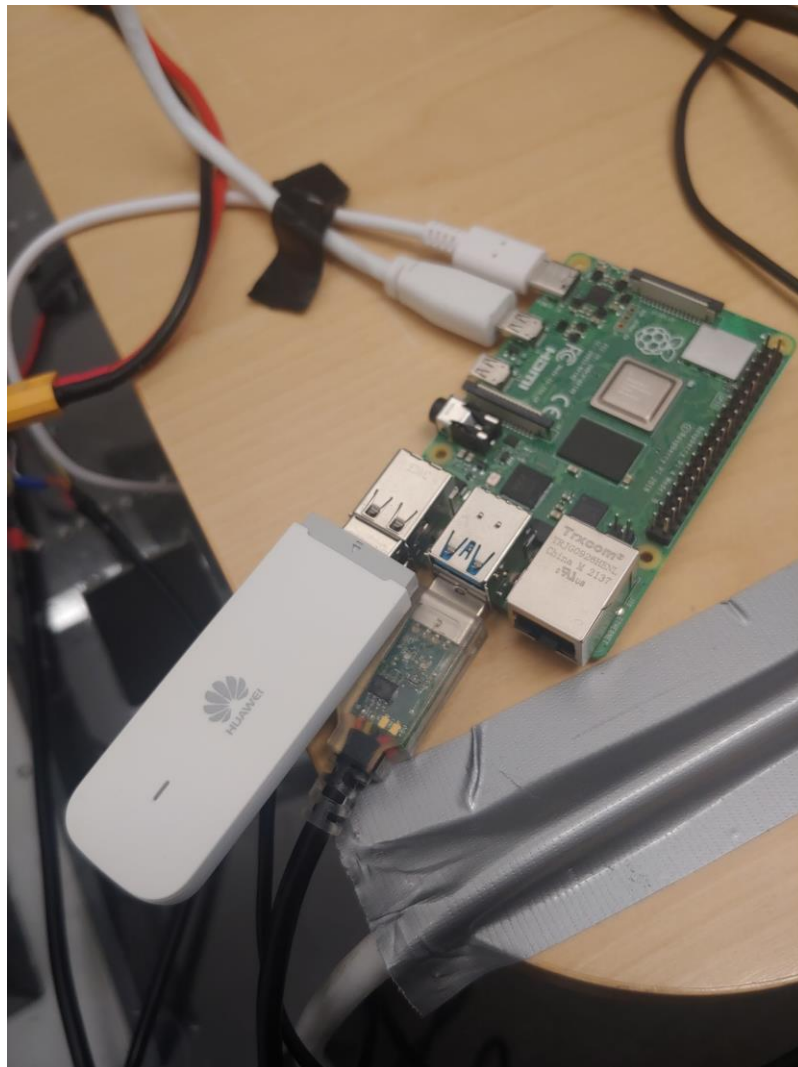


Figure 32. Test of 4G connection the mobile network connected to the microprocessor.

Testing the GPS/RockBlock for satellite communication

These unit tests are performed to understand the behavior and see if the components work as intended. The tests will be performed outdoors with a free sight of the sky since that is the most similar to the prototype's operational environment.

The output data from the GPS will be tested in two ways: first, by entering the coordinates in google earth and seeing if the website accurately shows the GPS's current position. Secondly, coordinates from the GPS will be taken at two points, and then use the *equation*

s 7-9 in order to calculate the distance between them. In addition, the result from the equations will be compared to the distance calculated by google earth's built-in function to measure distances between two optional points. As for the RockBlock, it will be tested by being hardcoded to send an emergency message.

The expected results from the GPS and RockBlock are that they are accurate



Figure 33. Test of RockBlock connected to computer for testing the satellite connection.

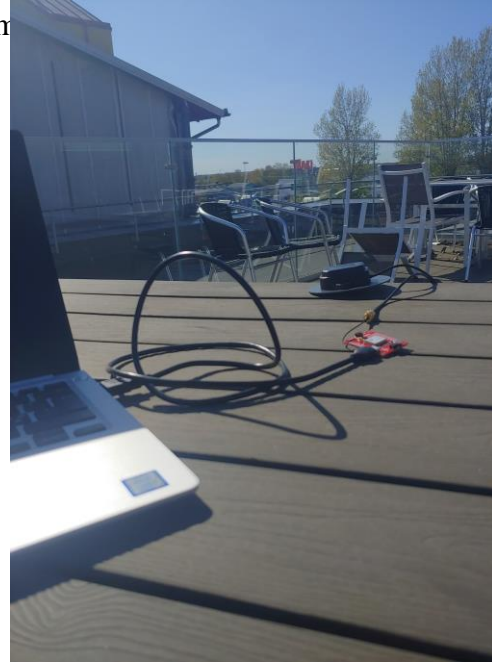


Figure 34. Test of GPS connected to computer for testing the satellite connection.

Testing the sonar's measurement and confidence

A unit test will be done to see that the sonar is working. Firstly, send the measurements and confidence data to the microprocessor and then to the AWS database. This test will be performed in the test aquarium to ensure that it gathers the data and displays it. Then the measured value from the sonar will be compared to a hand-measured distance of the same object. In addition, the travel time for the emitted pulse will be calculated and compared with the help of these two parameters and *equations 15 and 16*.

Expectations for this result are to understand how the sonar measures and detects the nearest obstacle or depth if there are no obstacles in its path; the data will also be sent successfully to the database and have reasonable values for both measurements and confidence.

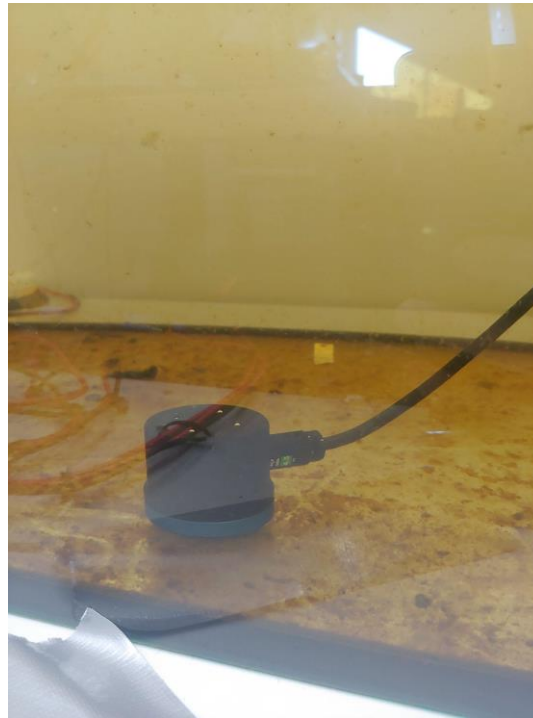


Figure 35. Test of the sonar in the aquarium for confidence and depth.

Testing the lidar

According to the datasheet of the lidar, it is suitable for outdoor usage. However, it does not say how it works when scanning obstacles on the water. The test needed for this lidar is how it behaves when scanning the distance between itself and the water's surface—a unit test and then a code test to accomplish this test. This is to live stream the graphical view and collect the distance to the nearest obstacle along the water's surface.

Results, to see if the laser beam from lidar is bouncing on the water's surface or if it actually can measure the distance to the nearest obstacles. If so is the case, wants to send the data of XYZ coordinates to the AWS database and live stream the graphs view on our website for users.



Figure 36. Test of the lidar connected to computer for able to plot the lidars output graphically.

Test of solar panels subsystem

As for the testing of the solar panels, it is problematic to get the Raspberry Pi to read the battery's percentage since it does not have Bluetooth. However, daily usable mobile phones have Bluetooth, so the first step is to connect the battery with a phone to establish communication and be able to read the actual battery percentage. The second step is to manipulate the number of sun rays accessing the solar panels; as a result, this opens up the possibility to see if the solar panels increase the battery percentage overtime when it is supposed to or if it remains static. When testing the solar panels' effectiveness, four main occasions need to be tested: complete darkness, cloudy weather, and finally, when the sun is shining. As a result of these three test occasions, It gets visible how it performs with none, some, and many sun rays being accessed, which are the three critical occurrences on the spectrum.

Datasheet of TpH-D sensor

Resolution		0.01 pH
Measurement accuracy		± 0,06 pH
Repeatability	pH7	± 0.05 pH
Warm-up time		< 5 min

Figure 37. According to the datasheet of TpH-D sensor, the datasheet that will be used for comparison with calculations.

Integration test

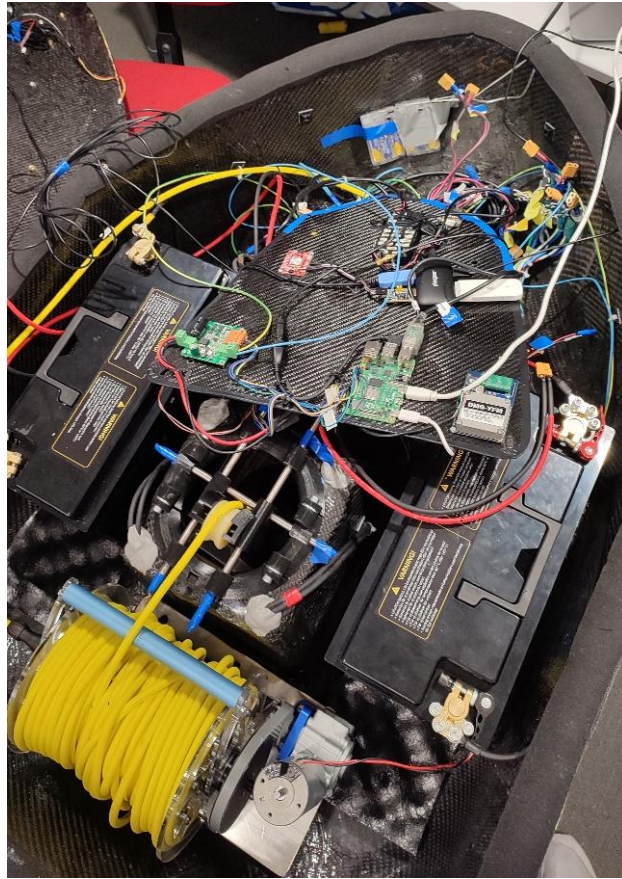


Figure 38. According to the datasheet of TpH-D sensor, the datasheet that will be used for comparison with calculations.

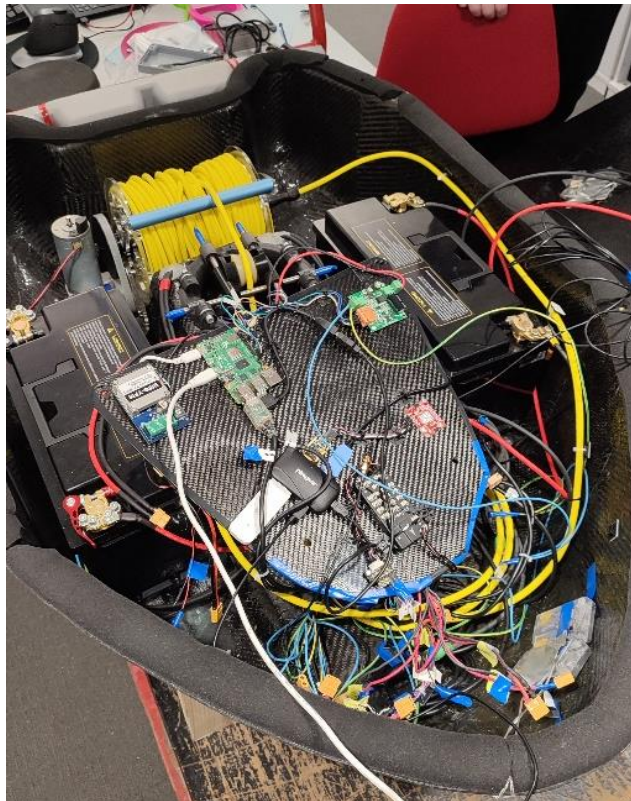


Figure 39. According to the datasheet of TpH-D sensor, the datasheet that will be used for comparison with calculations.



Figure 40. According to the datasheet of TpH-D sensor, the datasheet that will be used for comparison with calculations.

E Autonomous drone



Figure 41. Sunchallenger #2 is a autonomous drone boat that operates in the ocean with the goal to measure the water, and air quality with sensors. .Adopted from [86].

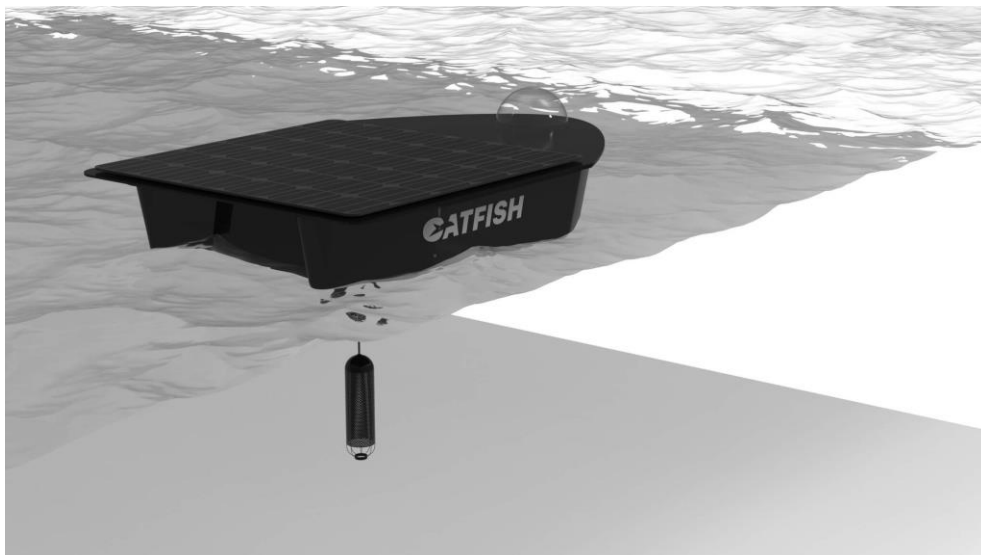


Figure 42. CatFish design prototype with solar panels made in CAD Fusion 360.



Figure 43. The almost finished prototype, without the implemented system, only solar panel and regulator.



Figure 44. *The almost finished prototype, without the implemented system, only solar panel and regulator.*

F Timeline

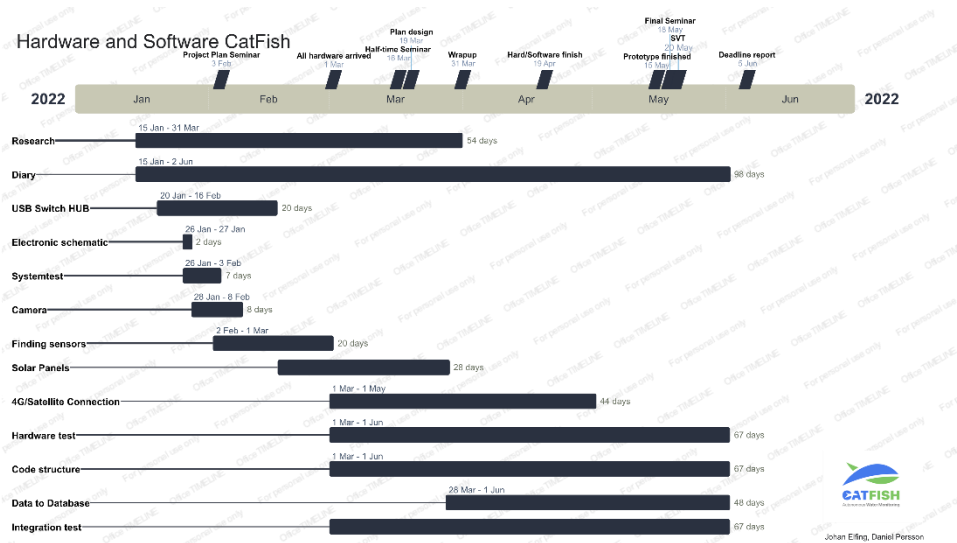


Figure 45. The projects timeline with milestones.