University of Minho
School of Engineering

Vinicius Cunha Azevedo Carvalho

9S Drifter
Sensoring System for Surface Sea Streams and Scan of Sustainable Energy Streams

october 2025

University of Minho
School of Engineering

Vinicius Cunha Azevedo Carvalho

9S Drifter
Sensoring System for Surface Sea Streams and Scan of Sustainable Energy Streams

Master's Dissertation in
Industrial Electronics and Computers Engineering

Dissertation supervised by
Professor Luis Gonçalves
Co-Supervisor Name

october 2025

Copyright and Terms of Use for Third Party Work

License granted to users of this work:

Acknowledgements

Write your acknowledgements here. Do not forget to mention the projects and grants that you have benefited from while doing your research, if any. Ask your supervisor about the specific textual format to use. (Funding agencies are quite strict about this.)

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, Braga, october 2025

Vinicius Cunha Azevedo Carvalho

Abstract

The ocean is one of the ancient mysteries of mankind even before the written history. Humanity made the world ours over the water, from the Portuguese greatest discoveries, braving the raging ocean to the newest oil tanker demanding ever newer technology in order to tame the sea for safer and smoother sailing.

Nowadays, scientists believe only 20% to 26% of the ocean is discovered with the actual technology, which means that humanity knows as much about our great sky as our own seas. The 5S ocean drifter is equipment made to acquire data from superficial sea streams and expand the oceanographic knowledge about them.

Improved ocean knowledge supports development in various areas, enhancing safety, security, plannign of shorter trips over seas, less fuel consuption and most recently energy harvesting.

Keywords Drifter, Sea Streams, Renewable Energy, Embedded System

Resumo

Escrever aqui o resumo (pt)

Palavras-chave Boia, Correntes Maritmas, Energia Renovavel, Sistema Embebido.

Contents

List of Figures

List of Tables

Part I

Introductory material

Chapter 1

Introduction

Under the course unit of Integrative Project in Industrial Electronics and Computers, the students must apply for professors' projects in order to integrate under their respective laboratories and start to understand the pace demanded in the Master's final paper.

This project, given by Professor Luis Gonçalves under the CMEMS laboratory, has the main purpose of creating a drifter for data acquisition. As a multi-themed project, this report will explore multiple areas, as the **Printed Circuit Board** design for hardware and firmware manufacture, software design under the idea to optimize the execution, allowing for better performance. The main goal is to have the final product afloat at the end of the semester.

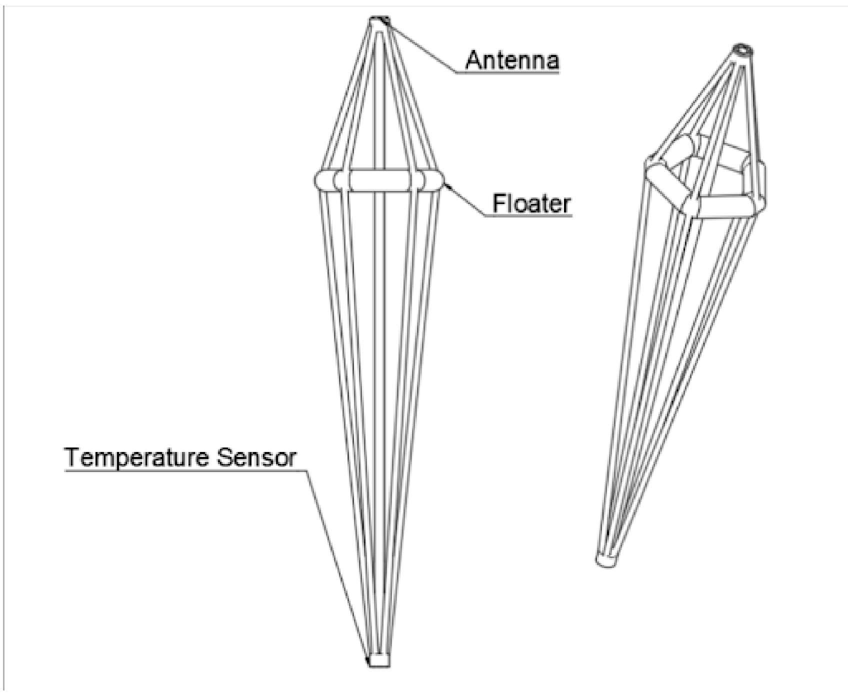

Figure 1: Draft Floater

1.0.1 Problem Statement

5S, an acronym for **Sensoring System for Surface Sea Streams**, is a low-cost, low-power solution to acquire said data with the focus to last autonomously for the longest time possible. The drifter has to attain its GPS coordinates in order to track its current and average velocity, alongside with the water temperature and accelerometer information to gather data about wave intensity. All this data will be stored locally and transmitted by a protocol, yet to be defined, using JSON format to be received by a preexisting database.

Oceanography

The current under study is the **Iberian Poleward Current (IPC)**, well documented by the Advanced Very High Resolution Radiometer (AVHRR) over the last two decades. It is a narrow (25–40 km) flow of water that follows the continental slope due to topography and/or water density, commonly referred to as a "slope-trapped tongue". Being strongest in the winter and weaker during summer, the current is significantly warmer and carries more salinity due to Mediterranean influence over the Eastern North Atlantic Central Water (ENACW).

Transport

It isn't uncommon to see transport accidents being reported, and even worse, for it to be a gigantic problem. Some of these accidents are caused by poor mapping of sea conditions — tankers spilling oil, fishing vessels capsizing — leading to financial problems and even loss of life. Even when there are no accidents, poor knowledge of tides results in higher energy consumption when routes are set against the currents.

A solution would be to create optimized shipping routes, minimizing accidents and improving energy efficiency while traversing the waves. Oil tankers could follow currents with lower fuel consumption. Fishing routes could become more efficient, as their target species may swim with the tides based on temperature and speed. This would ease the workload, making the activity less reactive and more predictable.

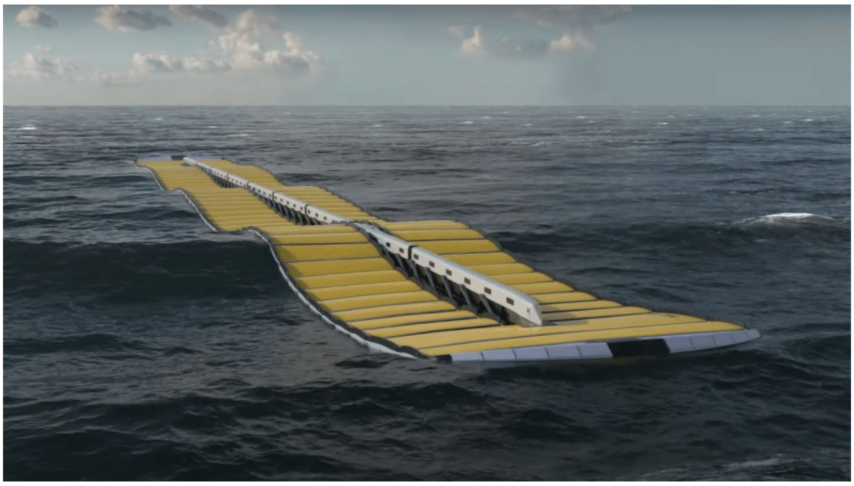

Figure 2: The Design of a Wave Energy Converter to Electricity

Ecology

The IPC has an important role in the ecosystem, transporting plankton, larvae, and nutrients along the coast — essential components for marine fauna and flora.

Another perspective on the importance of the IPC is in **wave energy converter placement**, a growing field of renewable energy. Good positioning improves efficiency and reduces costs of construction and maintenance.

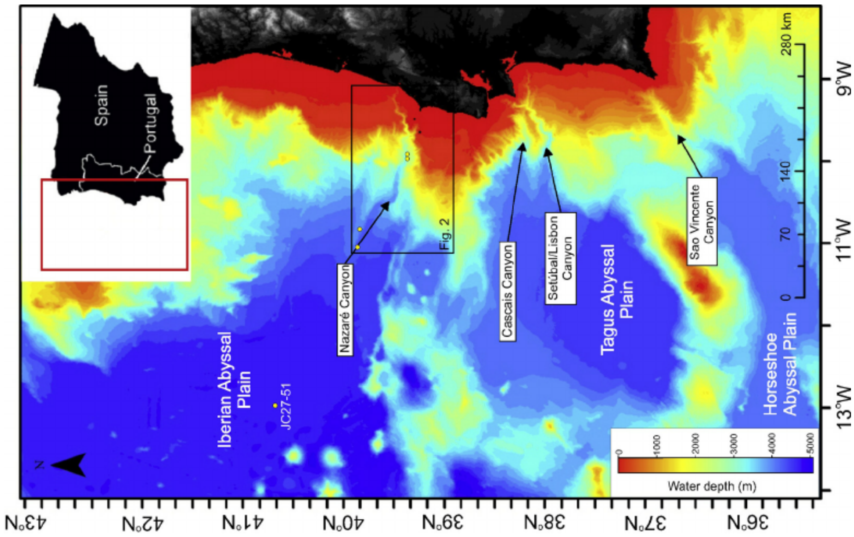

Figure 3: Depth Map of Portugal's Coast

Safety

As time passes, one of the important jobs of oceanography is the study of sediments of rocky structures at shore and their deposition. This data is useful to determine shore mineral composition, predict erosion areas, and evaluate coastal vulnerability and resilience.

The coast attracts human activity — sports, leisure, housing, and industry — so good information around it prevents harm and fosters safety.

1.0.2 Problem Statement Analysis

As a first step into solving this project, an initial construction of the demends is requested. Here will be presented, following the waterfall aprouch and UML standarts, the solutions to the individual problems presented by the project.

As stated, the floater has a series of data to acuare, format, and send in order to be considered concluded. The system will be divided in smaller and simpler packeges to solve each point then it will be assembled as a final product.

Equipment Objectives

The system, in order to acomplish said targets, must set the following topics

- Data acquisition
  - Power Source Level
    In order to alert a low batteries.
  - Wave intensity
    Measuring the force exercised by the water in the drifter
  - Position
    Track the movement of the water.
  - Temperature
    Measures the temperature directly.
- Wireless data transference
  As the floaters has no physical connections with the shore.
- Local data storage
  In case of lack of external communication.
- Autonomy
  The longer it survives, the more data it will gather.

- Resistant and buoyant shell

  The physical parts that support the system.

Chapter 2

Analysis

In order to accomplish said objectives listed on the problem Statement Analysis, it is first needed to enlist the embedded system components, this will help to choose a STM32 model as well the modules for this task without any over and under dimensions.

- Microcontroller

  The STM32 CPU that will control the Embedded System.

- GNSS

  A module to acquire the world position in latitude and longitude.

- Mobile Communication

  The module with the ability to communicate wirelessly with MongoDB

- Power Source

  The set of batteries the system will relay on for energy. It is stipulated that the system must have the autonomy of at least 30 days.

- SD Card Slot

  Local long term memory in case of field transmission.

- Sensors:

  - IMU

    System physical acceleration and angle data.

  - Temperature Sensor

    Water Temperature data.

  - Power Source Level Sensor

    Voltage reading data.

As for the outer shell there are a few things to have in attention.

- The Antenna to floater distance, once the water interferes with the antenna signal.
- The float size, that needs to support all the weight and float, respecting the first topic.
- The drifter ballast, that will be the drifter core, located between the floater and the temperature sensor tip. It should be heavy so the drifter points up, but it shouldn't exceed the second topic demand
- The Floater volume, as it has to balance the whole electronics and shell weight.

This creates a first view of the system as a block diagram. Indicating the physical connection between each component. This won't define the 5S architecture, but it will serve as an initial guide to build on.

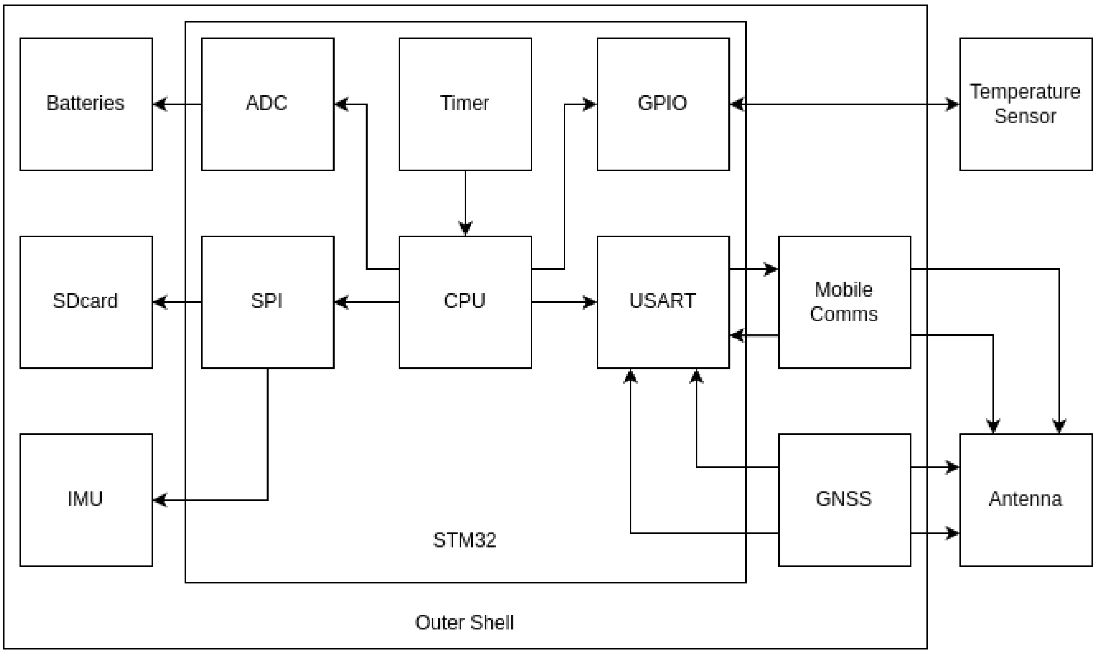

Figure 4: Block Diagram Initial Draft

The floater draft shows the initial concept of a drifter, as it has the above water antenna, a floater to counterbalance the buoy weight, and the temperature sensor. According to the blocked diagram.

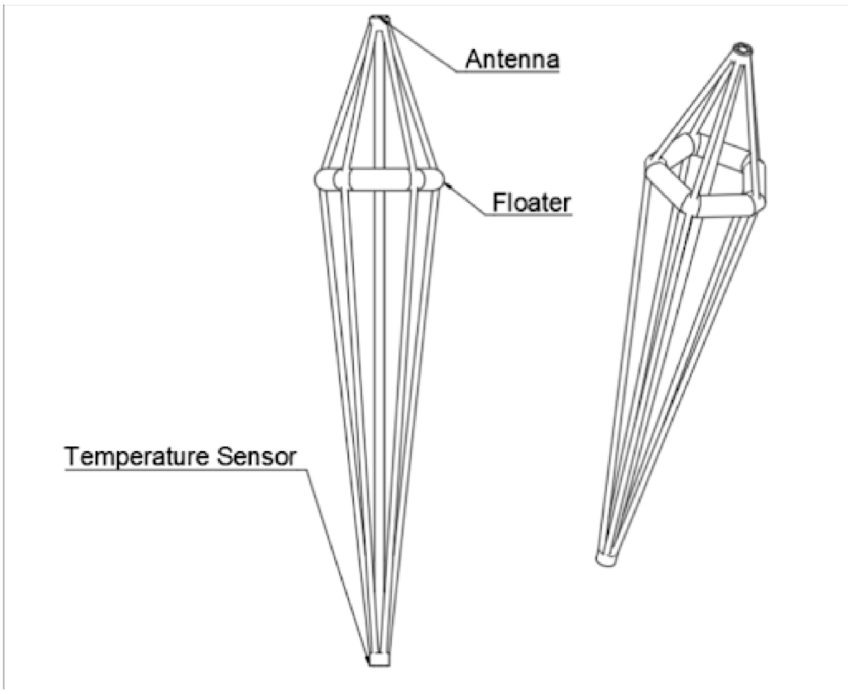

Figure 5: Floater Architecture Initial Draft

2.1 Requirements and Constraints

2.1.1 Requirements

- Search and selection of hardware components.
- Software design.
- **PCB** design.
- 5S outer shell 3D design.
- Actual product realization.
- Laboratory tests.

2.1.2 Constraints

- Limited Team
- The project must be presented for evaluation within deadline.
- The project has to be validated at the ocean.

- The pretended autonomy has to be of a month at minimum.

2.2 State of the art

Nowadays, there are a series of reasons in witch drifters are used. Usually, government departments, companies and universities with relations to oceanography, use drifter alike to research the local coast for the following reasons.

- Border Control
- Climate Modeling
- Traffic management
- Aquaculture management
- Public oceanographic research
- Marine spatial planning
- Defense and security

5S drifter aims for **Climate Modeling**, **Public oceanographic** or even **Traffic management** research as this project main objective. As the data for this project can be used, as stated in the Introduction, to model the environment in order to better the quality of said topics.

2.2.1 Copernicus Example

The Copernicus Marine Service, component of Copernicus Programme of the European Union, has developed the SVP-BRST, a drifter for temperature and depth measurement. Based on the SSP-B design, the updated version implements a HRSST in addition to the regular SST sensor, gathering the position with GNSS and transmitting the data using Iridium.

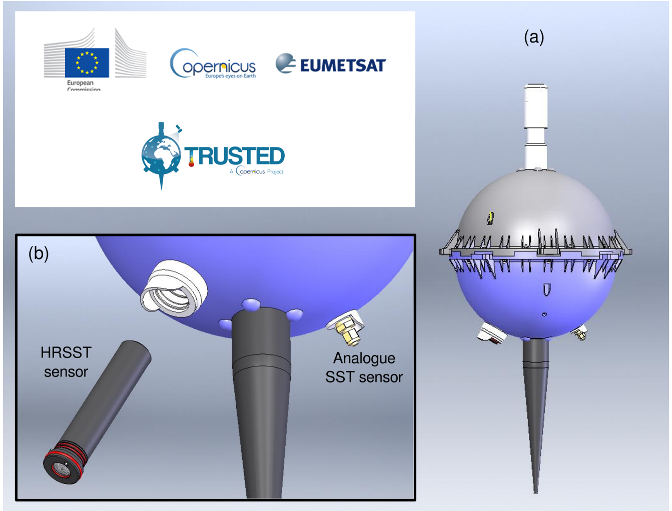

Figure 6: SVP-BRST Design

The image shows the SVP-BRST model. The (a) image shows the whole model with the antenna the buoyant part ad the weight to point the buoy up. As for the (b) image, shows the sensor's placement.

2.2.2 CMEMS Examples

The CMEMS lab at the University of Minho combines expertise in microsystems engineering with environmental research to develop innovative ocean monitoring technologies. One of its key areas involves the design and deployment of autonomous drifters to collect real-time data on sea surface conditions and currents.

SONDA

The project proposes a complementary system for atmospheric and oceanic monitoring using configurable probes launched by high-altitude balloons. These probes can measure environmental parameters from the stratosphere to the deep sea. After reaching the seafloor and collecting data (including acoustic imaging), the probe resurfaces and transmits the data via satellite. It then drifts until its material degrades. The system offers a low-cost, high-payload solution with limited positional accuracy. Matos et al. (2022)

NextSea

A new approach of monitoring coastal and estuarine relevant variables is presented. MEMS (Micro Electromechanical Systems), lab-on-chip and microelectronics are used to miniaturize and optimize main oceanic sampling. Beyond typical CTD (Conductivity, Temperature and Depth), other oceanic variables are also monitored. The system also samples type of phytoplankton and its concentration, pH, currents direction and intensity and turbidity. Matos et al. (2025)

2.3 System Architecture

Once studied what the system needs to follow, it is now needed to formulate a solution. Here it will be used UML diagrams to convey the solution in question, without specific implementation details, as this proposal serves to validate and organize the problem structure as a formal language.

Block Diagram

As previously stated, a block diagram is a formal representation of the system components and their connection. This one, shows how the peripherals communicate with the microcontroller STM32 as it considers connections with components outside the shell. Initially, as a first analysis the communication protocol isn't yet defined. However, it will be used to narrow it down.

The diagram shows the SMT32 minimum peripherals: ADC, TIMER, GPIO, I2C / SPI, USART. However, as it describes the hardware, it doesn't give a good notion on the system software.

Some points to be considered are:

- The temperature sensor communicate using the GPIO, as waterproof sensors, usually, use a driver with specific communication protocols.
- Mobile Comms is the protocol that will be used to send information to the database. Yet to be chosen.
- GNSS and Mobile Comms for now communicate to the same Antenna, However it probably will use different antennas as different protocols require different frequencies to be used.
- The communication to the SDcard, uses SPI with STM32 below 64 pins. As it doesn't include SDMMC supporting FatFS.
- The batteries power the whole system. Even if not shown.
- Depending on the implementation, the timer block will be the Sysclock, as the RTOS may use by default.

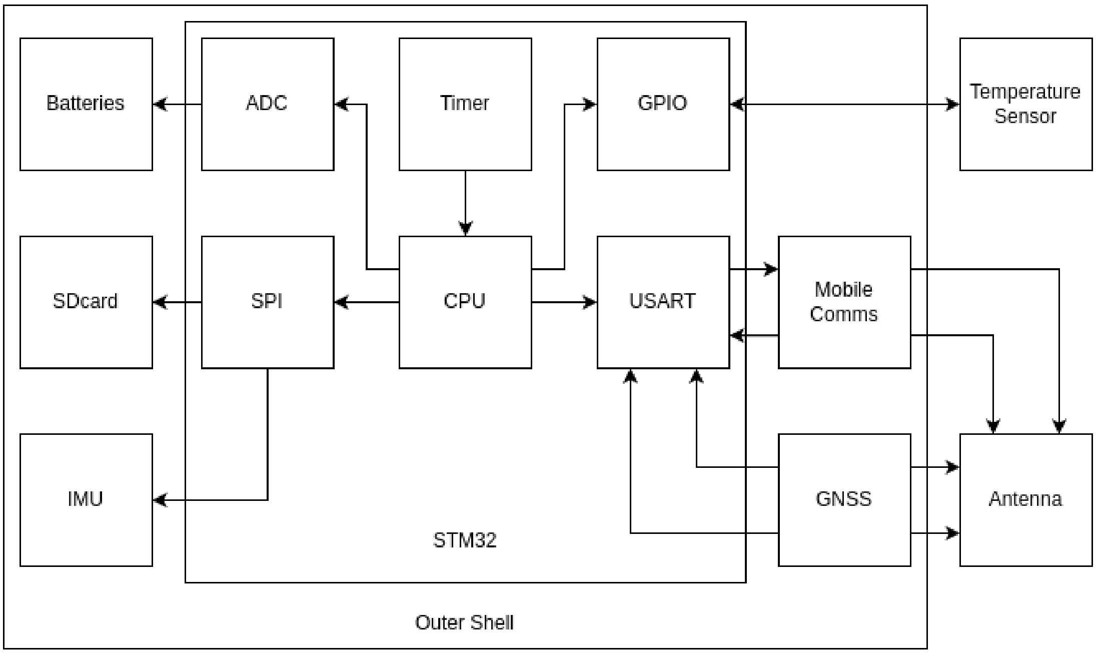

Figure 7: Block Diagram

Use Case

Now, for a software illustration, UML offers a user case, a diagram that exemplifies the system interactions and stimuli.

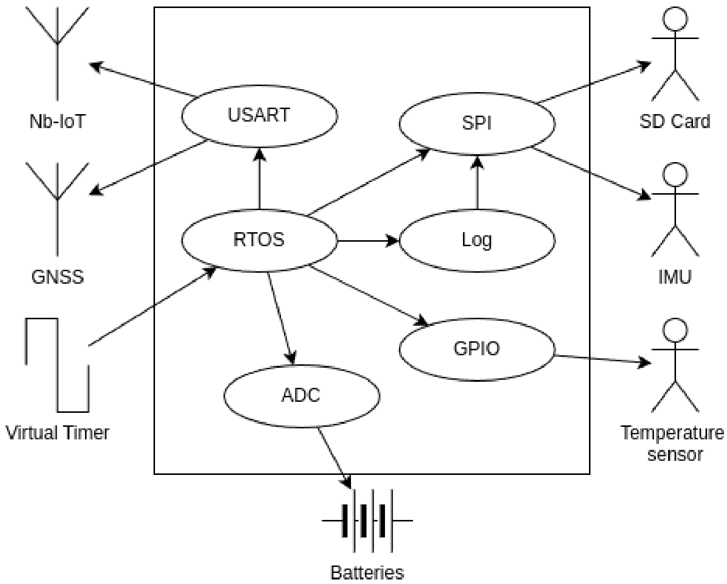

Figure 8: Use Case Diagram

Now it is introduced to one main component that will be explored further, an RTOS. As it is stimulated by the timer, it will actvate the peripherals in a organized form, allowing a fast and efficient use of energy. Other component is the Log, a virtual memory that will store temporary the information, only storing locally, once per cycle, reducing the SDcard entries.

Sequence Diagram

Once the software and hardware are planed, now a time related diagram will show the blocks interactions as time passes.

First task sequence, a generic "Data Acquisition Task" shows the general behavior of a task focused to manage the sensors.

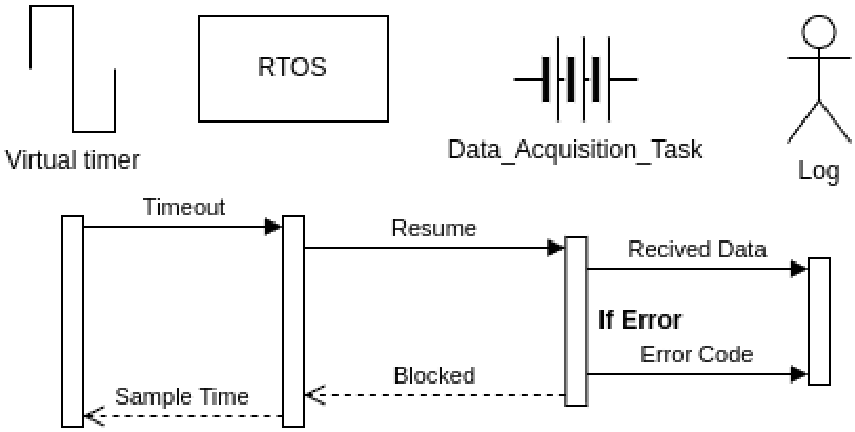

Figure 9: Sequence Diagram of Sensor Task

Next, the second task sequence, there will be 2 tasks with the behavior to get the GNSS information, store it on log, then call for the Mobile Comms task to send the stored memory to the Database. On the end, initiating the SDcard storage.

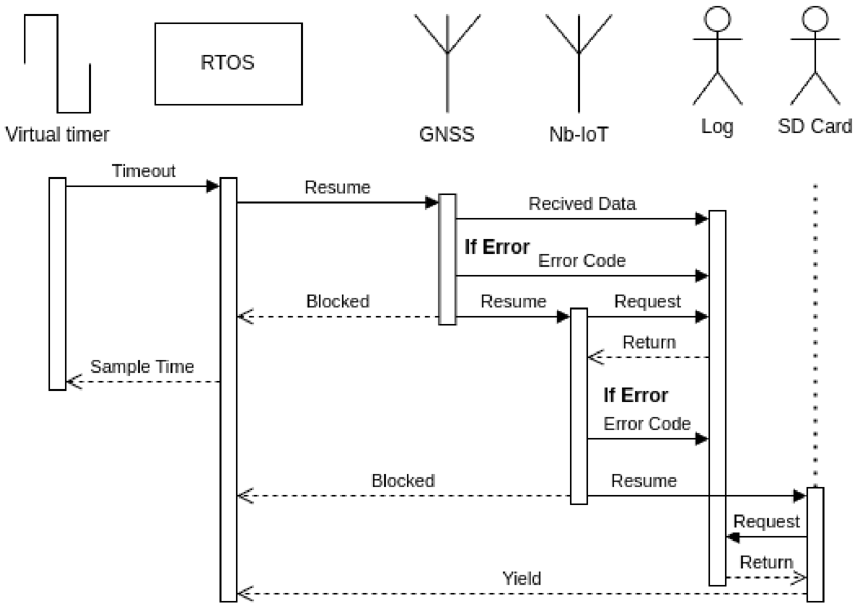

Figure 10: Sequence Diagram for Sending and Archive Task

Threads

Once this problem requires a list of tasks to be executed, using a OS will allow a better project organization and performance with little to no impact in power consumption, even helping to manage the low-power mode. However, the implementation will suffer, as the codification as verification grows, so this topic may become an additional information by the end of the project.

As the ST uC offers a variety of RTOS, the implementation will be accessible with good support due to the CMSIS v2 abstraction layer.

The division in Threads demands a separation in Priority levels, as the OS scheduler takes in consideration once both tasks are ready for execution.

Setting a task priority it must take in vision the resources the task will use, the time it will take to execute said behavior and the actual importance in matching it time constraints. In order to manage this level of complexity, the RTOS offers a set of tools for tasks control that will be used for its synchronization and communication.

- High Priority Threads

  Tasks that will handle the outer communication as GNSS and the internet integration will take the higher priority once, as will be handled by a peripheral, its execution will be faster, only using the USART interface for AC transmission.

- Normal Priority Threads

  The only task here will be the one that has enough importance to be prioritized over the sensors but as the transmission begins it should release the processor for the outer communication.

- Low Priority Threads

  Tasks that only has to measure the sensors having no problem to be removed from the CPU execution once their execution is, in their majority, asynchronous.

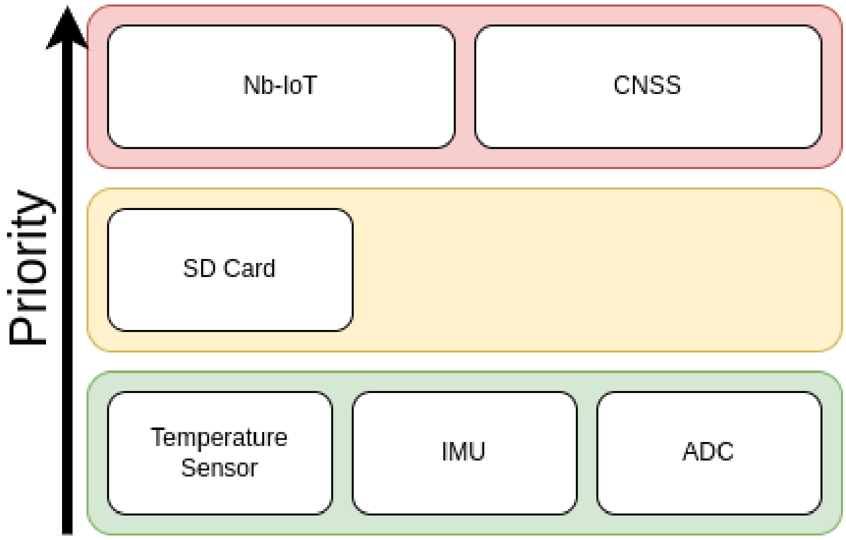

Figure 11: Thread Priority Stack

Other important perspective, is to emulate the task behavior with the priorities in action. Showing a repetion of X times a group of data acquisition tasks, then ending with the second task sequence. X yet to be defined as it first depends on the sample time.

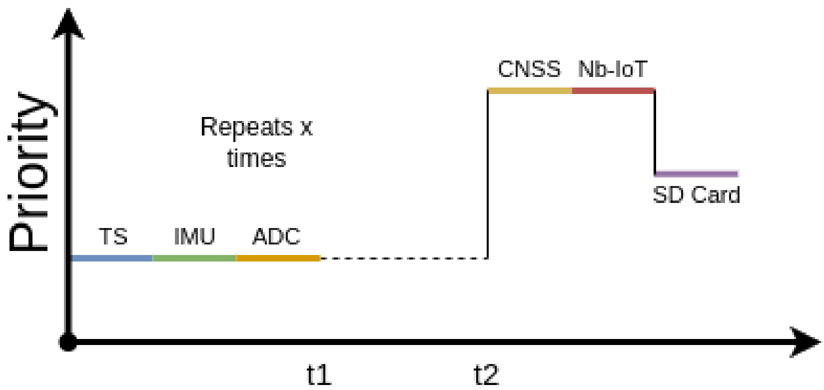

Figure 12: Thread Temporal Graph

Here up to t1 the system executes tasks, storing data, then entering low-power mode until the next sample. Then, at the end of the last repetition, at t2, stating the second task sequence. Then after the SDcard task behavior ends, and so does the cycle.

Memory Abstraction (Log)

The final diagrams, are a exception to UML diagrams, As it only builds a data structure and a simple interface to interact with it. Here the idea is to allow memory to be accessible while written by the system, optimizing the RTOS ITCs options.

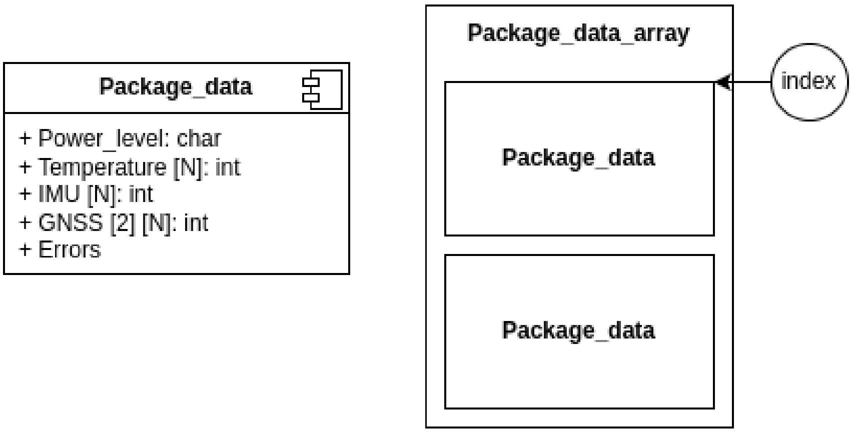

Figure 13: Package data structure

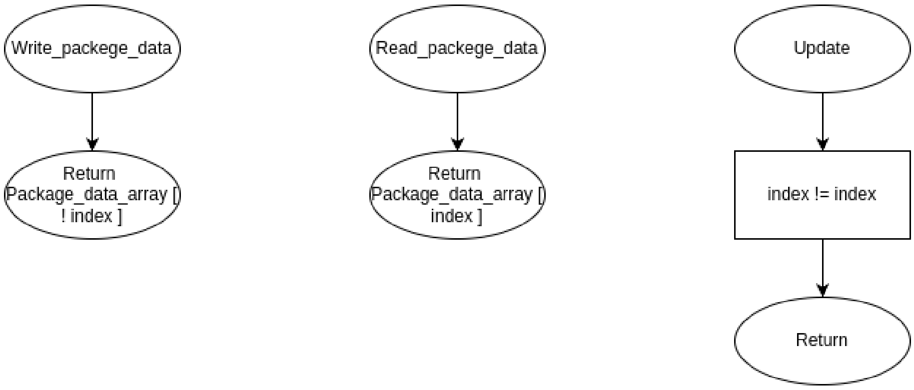

Figure 14: Memory Flowchart

2.4 Footnotes

This is a footnote example[1].

2.5 Acronyms and Glossary

Given a set of numbers, there are elementary methods to compute its **Greatest Common Divisor**, which is abbreviated **GCD**. This process is similar to that used for the **Least Common Multiple (LCM)**.

The **Latex** typesetting markup language is specially suitable for documents that include **mathematics**. **Formulas** are rendered properly an easily once one gets used to the commands.

[1] The quick brown fox jumps over the lazy dog.

2.6 Index

In this example, several keywords will be used which are important and deserve to appear in the Index.

Terms like generate and some will also show up. Terms in the index can also be nested .

Cf. the `dissertation.bib` file to see some index definitions like UMinho .

Chapter 3

Design

Now, once planed ”What” is the system, it is needed to plan the ”How” the system will be build in the implementation phase. Here is explained the components and the reason for it. As well for the instructions on the system construction and realization.

3.1 Analysis Review

Taking in account the limited personal and time, some of what is specified in Analysis has its scope reduced as to accomplish according to the requirements and constraints. Using as inspiration the Copernicus SVP-BRST and the CMEMS labs previous experience, the 5S can/ and will recall for used technology for easier design process.

The project then will be divided in three main parts; The Software, Hardware and the outer shell. Then, lately, it will be shown the pieces of software used to accomplish this project.

3.2 Software

As for software there will be two possible approaches for the problem, as the use of a RTOS as stated on the Analysis, may require more time than it is worth. However, the use of a RTOS, allow for a better control off timings and the low-power mode. Either way, as the implementation of both alternatives may differ, the overall design will remain the same.

This section will first decide the protocol available, abstractions, then the possibility of using a RTOS

3.2.1 Communication protocol

The selection of a network protocol is a complicated process as it interests with several factors that imply directly on the data to be acquired. To list the ones with more influence.

- Distance
- Bit-Rate
- Signal Availability (SIM card and protocol local integration)
- Price per package
- Integrated GNSS Hardware
- low-power

There are several protocols to choose from, so first it is needed to categorize the system needs and filter the list.

The first filter will be the distance. The IPC extends for kilometers along the shore, however signal along the shore can be mostly guaranteed as the service providers are mostly equally distributed. The real distinct characteristic is the reach offshore going inside the ocean. This category of protocol is a WWAN (Wireless Wide Area Network)

Some available protocols are; 2G(GS), 3G(UMTS), 4G(**LTE**), 5G, NB-IoT, Lora, Iridium

As the distance increases, the effective bit-rate tends to lower, due to protocols overhead and errors, so is espected a lower than usual bit-rate.

The protocol availability will depend on the service provider. This one is another set of varieties itself, however it can be divided as local and global service providers.

Services like SIMBASE, a global provider, are available for the lab to use, however it will offer coverage for a specific set of protocols in Portugal (coast). This one uses the local network interface, so the coverage of each protocol depends on the local provider range of signal.

Portugal	**2G**	**3G**	**4G**	**5G**	**LTE-M**	**NB-IoT**
Meo	V	V	V	–	–	–
Nos	V	V	–	–	–	–
Vodafone	V	–	V	V	V	–

Table 1: Protocol per provider table

Doing a local search, the Vodafone service provider, differs from the others, as also offers NB-IoT protocol if accorded directly.

By these factors, it is reducible to LTE and NB-IoT protocols. The price is inclined to the LTE protocol, as is already available on lab, but the NB-IoT is more centered around the low-power. There are several boards that include both protocols with GNSS integrated so it also ain't a defining criterion.

As low-power is a main concern on this project, the NB-Iot is the chosen protocol, however the 4G is a strong alternative, in case of a substitution.

3.2.2 Abstraction

As the implementation require multiple communication protocols between components, the development will require a layer of abstraction named wrappers. Protocols like OneWire, SPI and AT commands can and should be a higher level function, even with HAL assistance.

AT commands can be simplified to simpler functions without the necessity to write down the command. As HAL handles the **UART** transmission by taking an array, it is possible to use a sprintf to format the command using function parameters. Then it can be received the module answer and writing it on an array given when the function is called.

As for the SPI, here the intention is to simplify the multiple registers needed to configure the system components like the IMU. A simple Get Function that sends a pre-defined signal, activating the right CS and making the communication easier.

The OneWire, as not covered by the HAL, can be the hardest one, as it would be needed to be written from scratch. However, there is already libraries specific for the module in use, simplifying the process to program on this phase.

3.3 Hardware

This section elaborate on the system physical components, their connections and the reasons they were chosen. It is important to remember that the main objective, even if it is not specified on the topic, is the low-power functionalities. So the ability to consume less power is a favoring point. https://chatgpt.com/c/687081d7-f034-8000-bc62-035392361453

3.3.1 Autonomy

Autonomy is the main point to look for as the system is designed, as the higher autonomy, the longer 5S can stay adrift without maintenance.

As for the autonomy there are two main factors to consider, the batteries and the board consumption

To do it, so it is first need to set a consumption goal and the autonomy is good enough if the power needed is lower or equal then the goal.

Although the renewable energy is out of the scope of this project, the components suggested for future improvements will be shown at the end as a possible future feature.

As a restriction, the minimum amount of time in ocean is 1 month(720 hours), this will help us define the consumption goal.

Board Consumption

Remembering the Analysis list of components, there are several components that consume power in the system. And the sum of them will be used to choose the battery capacity.

The main power sinks in the systems will be the Microcontroller and the Communication Shield. As a head-start, choosing these two components will help. There are several options for both, however it must be selected according to the settled rules. The microcontroller has to be an STM32, containing the necessary peripheries, and the module have NB-IoT (And LTE just in case), both with low-power functionalities.

According to the microcontroller rules, there are still several boards with the minimum resources to accomplish the goal, even if it is selected the L0 models, designed for low-power. As this first prototype, it was selected a STM32H7, as it is available, that has an elevated power consumption, however it allows for a better system modulation. As the HAL is used, the transference for a future STML0 board model, with the necessary resources is mostly effortless.

Feature	**STM32H7 μA/MHz**	**STM32L0 μA/MHz**
Run Current @ 3.3V	$\sim$600	$\sim$87
Sleep/Stop Mode	$\sim$2.5	$\sim$0.5
Standby Mode	$\sim$0.25	$\sim$0.2
Supply Voltage	1.62V to 3.6V	1.65V to 3.6V

Table 2: Typical power consumption values for STM32H7 vs STM32L0.

Initial, the prototype will consume, on average as it will mostly stay on sleep mode , 2.5 μA/MHz, and lately 0.5μA/MHz.

As for he GNSS/NB-Iot or LTE module, there is several modules that fit the design rules. The list of module available are SIM 7000,7020,7080,7600, Quectel BG77, Quectel BG95a and EVKITST87M01-1.

In conclusion, the Module to choose is the SIM7000E, as it has the better average consume and has both protocols at disposal.

Module	Sleep Mode (µA)	Idle Mode (mA)	Peak Current (mA)
SIM7020	20	N/A	2,000
SIM7080G	600	10	N/A
SIM7000E	1,000	11	167
SIM7600	2,800	18	896
Quectel BG77	530	1	559.980
Quectel BG95-M3	3,840	24	357
EVKITST87M01-1	N/A	N/A	N/A

Table 3: Current Consumption of Modules

Module	2G	3G	4G LTE	NB-IoT	GPS
SIM7020	No	No	No	Yes	No
SIM7080G	No	No	No	Yes	No
SIM7000E	Yes	No	Yes	Yes	Yes
SIM7600	Yes	Yes	CAT4	No	Yes
Quectel BG77	No	No	No	Yes	Yes
Quectel BG95-M3	Yes	No	LTE-M/NB-IoT	Yes	Yes
EVKITST87M01-1	No	No	No	Yes	No

Table 4: Supported Protocols by Module

The following items were chosen by the laboratory availability.

IMU 9DOF GY-85 - ITG3205 + ADXL345 + HMC5883L

The **GY-85** is a compact Inertial Measurement Unit (IMU) that integrates three key motion sensors, offering 9 degrees of freedom (9DOF). It combines the following components:

- **ITG-3205**: a 3-axis gyroscope for measuring angular velocity,
- **ADXL345**: a 3-axis accelerometer for detecting linear acceleration,
- **HMC5883L**: a 3-axis magnetometer for sensing magnetic fields and orientation.
- **Amperage consume**: 23uA while turned on.

Together, these sensors provide comprehensive motion and orientation data, making the GY-85 suitable for applications in robotics, drones, wearable devices, and general motion tracking systems. The module communicates using the I2C protocol, enabling easy integration with microcontrollers and embedded systems.

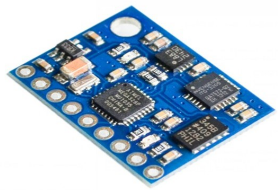

Figure 15: 9DOF GY-85 - ITG3205 + ADXL345 + HMC5883L

Taking an average consume of 23uA and speculating a 10% usage over the whole cycle.

$23uA * 10\% = 2.3uA/Cicle$

SIM7000E Arduino NB-IoT/LTE/GPRS/GPS Expansion Shield

The **SIM7000E Expansion Shield** is a versatile communication module designed for use with Arduino boards. It integrates the **SIM7000E** cellular module, enabling support for multiple communication technologies, including:

- **NB-IoT (Narrowband IoT)** for low-power, wide-area communication,
- **LTE Cat-M1** for efficient, low-latency data transmission,
- **GPRS/EDGE** as a fallback for 2G networks,
- **GPS** for accurate positioning and navigation.
- **Amperage consume** of 1 mA as it requests for data transference.

The shield is ideal for IoT applications requiring reliable connectivity and geolocation, such as asset tracking, environmental monitoring, smart agriculture, and remote sensing. It interfaces easily with Arduino via **UART**, making it accessible for both prototyping and deployment in embedded systems.

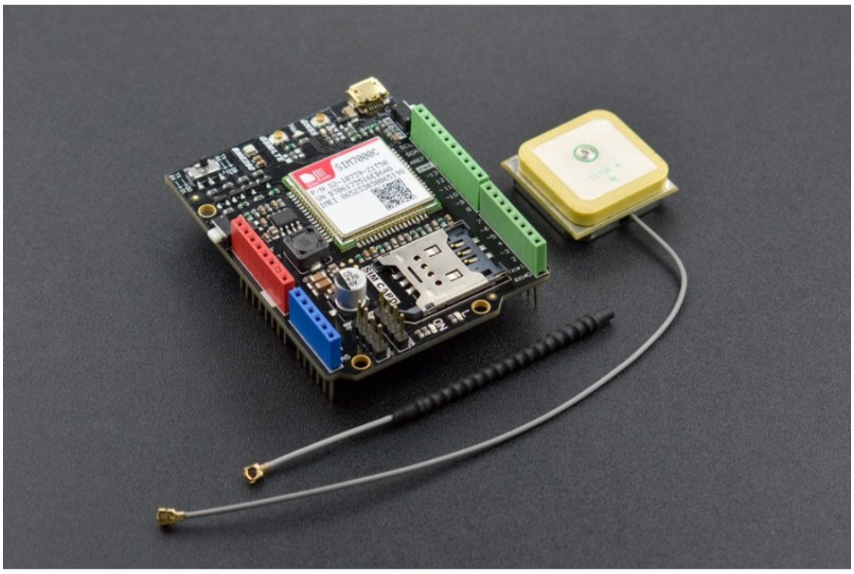

Figure 16: SIM7000E

Using both LTE and GNSS and a usual assuming that would take around 1 minute to transmit, with 10 minutes cycle ,gives the systems a 10% active mode per cycle as it.

In Idle mode consumes 11mA, as in LTE and GNSS pulls for 167mA for both.

$11\ mA \times 90\% + 167\ mA \times 10\% = 9.9\ mA + 16.7\ mA = 26.6\ mA/Cicle$

Waterproof Temperature Sensor (DS18B20)

The **DS18B20** is a digital temperature sensor known for its accuracy, reliability, and ease of use. Encapsulated in a waterproof stainless steel probe, this version of the sensor is ideal for use in wet or outdoor environments.

Key features include:

- **Temperature range**: -55°C to +125°C,
- **Accuracy**: ±0.5°C in the -10°C to +85°C range,
- **Digital output** via the **1-Wire** protocol, allowing multiple sensors to share a single data line,
- **Waterproof design** for robust operation in harsh environments.

- **Amperage consume** of 1mA per conversion.

The DS18B20 is commonly used in applications such as HVAC systems, weather stations, industrial temperature monitoring, and smart farming. Its simple interface and compatibility with microcontrollers like Arduino and STM32 make it a popular choice for both hobbyist and professional projects.

Figure 17: DS18B20

Using the same formula as before, now with a 0.17% as the module takes around 1 second per cycle, consuming 1mA

$1mA * 0.17\% = 0.0017mA = 1.7\mu A/Cicle$

Micro SDHC Card INF01016

The size of the SD card, will depend on the amount of data to be stored.

$StorageSize = \frac{PackegeSize(bytes)*fracTimeOnline(H)}{Sampleperiod(H)}$

Speculating that 5S will produce around 200 bytes each 10 minutes over 4 months it is possible to reach the value of

$StorageSize = \frac{200*4*30*24}{\frac{10}{60}} = 3.4Gbytes$

Figure 18: Micro SD card

At last, using the same values as the temperature sensor, the micro SD card will consume proximally the same.

$1mA * 0.17\% = 0.0017mA = 1.7\mu A/Cicle$

Micro SDHC Card Reader Module and Battery Holder

As the last components, there's will be used some support modules and battery support to hold the electronics in place.

Figure 19: Battery Holder

Possible Solar Energy

As an additional topic, here will be briefly explored the option of renewable energy, for future exploration.

The module AEM10941 and the solar panel SM111K06L are good recommendations.Is ad visible to follow the datasheet as the module requires specific configuration and circuits to work properly.

Batteries

Using the formula $BatteryCapacity = Amperage * Time$, it can be determinate the Amperage consumption limit. As Batteries capacity has a discrete value available on market, it is possible to make a table as graph the results for better visualization.

Using the 3,7V batteries available by the laboratory, the following table can be constructed.

Cell Capacity (mAh)	**Max Amperage (A)**	**Price (▯)**
800	1.11	4.95
2000	2.77	3.40
2200	3.05	3.90
2500	3.47	4.40
2600	3.61	4.60
3200	4.44	5.40
3350	4.65	9.40

Table 5: Battery Capacities, Prices, and Current Capabilities

However, 1 month is a short amount of time when the IPC cycle take almost one third of the year. Then, using more batteries, 5S will accomplish to acquire date of a full cycle. As the equation for the Amperage consumption is linear, multiplying the amount of time by four implies four times the capacity meaning the system will support. Using the following table the amount of current needed per cycle is estimated.

Component	**Average Consumption**
STM32H7	27.25 μA
SIM7000E	26.6 mA
DS18B20	1.7 μA
IMU (GY-85)	2.3 μA
MicroSD Card	1.7 μA
Total Estimated	**26.63 mA**

Table 6: Average Power Consumption Per Component Per 10 Minutes Cicle

With a total of 26.63mA per cycle (mostly from the GNSS and LTE module), a system would need a 76Ah capacity making 22 from the most expensive batteries. That is unreal and unpractical. Here would perfectly fit a renewable source of energy such as solar or even harvest the wave undulation, however this is out of the project scope. Making the project unable to fulfill the four months period all by himself, needing a change of batteries i between. So in order to make the least trips along these 4 months as possible. In order to do it, was stipulated that 4 batteries was a good number, considering that it won't add as much weight. Then ,in order to get the drifter autonomy, the value is divided by the system consumption.

Total Capacity (mAh)	**Total Price (□)**	**Autonomy (h)**	**Autonomy (days)**	**Trips per Month**
3200	19.80	120.16	5.01	6
8000	13.60	300.41	12.52	3
8800	15.60	330.45	13.77	3
10000	17.60	375.55	15.65	2
10400	18.40	390.57	16.27	2
12800	21.60	480.70	20.03	2
13400	37.60	503.10	20.96	2

Table 7: Battery Capacity, Price, Estimated Autonomy, and Required Battery Swaps per Month

So in order to make the least trips per month and pay less, four of the 2500mAh batteries are the ideal.

Figure 20: LI-ION 18650 3,7V 2000MAH

3.3.2 System Pinout

Here will be displayed the systems' pinout. This table is used for further software and hardware development.

Name	Pinout	uC Pin
Temperature Sensor	VDD	3.3V
	GND	GND
	Signal	PA0 (ADC1_INP16)
IMU	VDD	3.3V
	GND	GND
	SDA/SCL	PB9 / PB8 (I2C1)
UART AT	TX/RX	PA9 / PA10 (USART1)
UART DEBUG	TX/RX	PC10 / PC11 (USART3)
ADC Input	Signal	PA1 (ADC1_INP17)
SD Card	VDD	3.3V
	GND	GND
	CMD	PD2 (SDMMC1_CMD)
	CLK	PC12 (SDMMC1_CK)
	D0	PC8 (SDMMC1_D0)
	D1	PC9 (SDMMC1_D1)
	D2	PC10 (SDMMC1_D2)
	D3	PC11 (SDMMC1_D3)

Table 8: Peripheral Pinout Mapping to Microcontroller (STM32H755ZI-Q)

3.4 Shell

Taking inspiration the SONDA drifter, the drifter's shell should have a way to conduct the wiring to the antennas and the temperature sensor and a basket to hold the electronics. Other point is to coat the electronics in plastic to waterproof the system.

The Shell has the following demands, as stated in analysis.

- 2.5 dB Antenna should be at least 15 cm form water.
- The buoy should lift at least 500g ballast to hold the weight of the electronics and the pillars.
- The basket should have 10cm in diameter by 4cm in hight.
- It should be impact resistant to resist the waves and possible boat impacts.

Using the Archmedes principle, to hold 500g of electronics on a 10cm in diameter by 4cm in hight it would sink in water. So in order to float neutral of even float more secure, it would be necessary to project a buoy with the density according to the formula

$$\rho_{\text{buoy}} = \frac{1000 \cdot (0.000314 + V_{\text{buoy}}) - 0.5}{V_{\text{buoy}}} \tag{3.1}$$

Which, considering a 20cm per 4cm buoy it would make at least $852\,\text{kg/m}^3$ or lower.

3.5 Tools and COTS

3.5.1 Tools

Inkscape

A free and open-source vector graphics editor used for creating and editing SVG-based diagrams, schematics, and illustrations. Useful for generating custom graphics, **PCB** artwork, and documentation visuals.

draw.io

A web-based diagramming tool for creating flowcharts, system architectures, block diagrams, and wiring schematics. Supports real-time collaboration and integrates with cloud services like Google Drive and GitHub.

STM32 CUBEmx

A configuration and code generation tool for STM32 microcontrollers. Simplifies peripheral setup, clock tree configuration, and generates initialization C code for use with IDEs like STM32CubeIDE or Keil.

Fusion360

A professional-grade 3D CAD, CAM, and CAE tool from Autodesk. Used for mechanical design, simulation, and 3D modeling, making it ideal for designing enclosures, mounts, and mechanical parts.

LaTeX

A high-quality typesetting system used for technical and scientific documentation. Ideal for creating well-formatted reports, theses, datasheets, and documentation with complex mathematical content

Minicom

Minicom is a text-based serial communication program for Unix-like systems. It allows you to connect to serial devices (like routers, microcontrollers, or modems) via a serial port (e.g., /dev/ttyACM0). It's commonly used for debugging, configuration, and monitoring of hardware over **UART**.

3.5.2 COTS

All COTS are specified on the Board Consumption chapter.

Chapter 4

The problem and its challenges

The problem and its challenges.

4.1 Images

Example of inserting an image as displayed text,

— wrapped into the text, bla-bla

— or as a floating body.

Figure 21: Caption

Part II

Core of the Dissertation

Chapter 5

Contribution

Main result(s) and their scientific evidence

5.1 Introduction

5.2 Summary

Chapter 6

Applications

Application of main result (examples and case studies)

6.1 Introduction

6.2 Summary

Chapter 7

Conclusions and future work

7.1 Conclusion

The 5S Drifter project almost successfully demonstrated the feasibility of developing a low-power, low-cost, and autonomous oceanographic drifter for surface sea stream monitoring. This integrative effort combined knowledge in embedded systems, wireless communication, sensor integration, and mechanical design to deliver prototype capable of acquiring GPS position, sea surface temperature and IMU data.

The prototype, works in isolation, but it still needs polishing due to the problems in development. The physical design of the drifter shell, inspired by successful CMEMS projects such as SONDA and NextSea, will continue to be improved and tested in sea.

Overall, the project represents a solid step toward scalable, autonomous marine data collection and demonstrates how embedded electronics can be applied effectively in oceanographic research. With future iterations and optimizations, the 5S Drifter could become a viable tool for environmental monitoring, coastal safety planning, and marine traffic management.

7.2 Limitations and Future improvements

Despite the functionality, the prototype has limitations. Most notably its dependence on battery replacement and inability to cover long deployment periods without manual intervention. These issues point to clear paths for future improvements, such as integrating solar charging and switching to more energy-efficient microcontrollers like the STM32L0 family. A potential full migration to an RTOS-based architecture would enhance multitasking and improve energy management.

7.3 Final GIT

Here is the full project on github.

7.4 Special Greetings

At last, it's important to add the support from the CMEMS labs personal as well of the professor Tiago Matos and professor Carlos Faria for the support with hardware selection and shell mechanical advise respectively.

7.5 Prospect for future work

Chapter 8

Planned Schedule

8.1 Activities

Task	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
Background and **SOA**	●	●	●							
PDR preparation		●	●	●						
Contribution				●	●	●	●	●	●	
Writing up							●	●	●	●

Table 9: Activities Plan

Bibliography

T. Matos, C. L. Faria, M. S. M. Martins, et al. Development of an automated sensor for in-situ continuous monitoring of sediment deposition and erosion: the sonda optical instrument. *Science of the Total Environment*, 808:152164, 2022.

T. Matos, J. L. Rocha, M. Martins, and L. M. Goncalves. Enhancing sea wave monitoring through integrated pressure sensors in smart marine cables, 2025. Submitted to [Journal/Conference], April 2025.

Index

Part III
Appendices

Appendix A
Support work

Auxiliary results which are not main-stream.

Appendix B
Details of results

Details of results whose length would compromise readability of main text.

Appendix C
Listings

Should this be the case.

Appendix D
Tooling

(Should this be the case)

Anyone using LaTeX should consider having a look at TUG, the TeX Users Group.

Place here information about funding, FCT project, etc. in which the work is framed. Leave empty otherwise.