FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Smartboard for Surfing

Diogo Correia

WORKING VERSION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Sérgio Reis Cunha (Universidade do Porto)

External Supervisor: Carlos Resende (Fraunhofer Portugal AICOS)

Co-Supervisor: Eduardo Grifo (Fraunhofer Portugal AICOS)

July 31, 2020

Resumo

A *internet* das coisas veio democratizar o uso de dispositivos para a monitorização de desportos. A maioria destes dispositivos, têm como foco o mercado de *fitness*, criando assim a necessidade de desenvolver novas soluções para outros desportos.

Este trabalho foca-se no desenvolvimento de uma solução low-profile capaz de adquirir os dados necessários para alimentar um algoritmo de monitorização de *surf* desenvolvido na Fraunhofer Portugal. Este dispositivo deverá suportar longos períodos de tempo dentro de água e não deverá interferir com o desempenho do surfista.

Devido aos requisitos do projeto, algumas das tecnologias foram de utilização obrigatória: microcontrolador, giroscópio, magnetómetro, acelerómetro e módulo GNSS. No entanto, outras tecnologias necessitaram de ser selecionadas para solucionar certos requisitos, como: o sistema de comunicação sem fios, o método de armazenamento de dados e o sistema de carregamento do dispositivo.

Para selecionar um sistema de comunicação sem fios, tivemos que estimar a quantidade de informação gerada pelo sistema e achar uma tecnologia de comunicação capaz de almejar uma boa relação entre a taxa de envio e a potência consumida. Para selecionar um método de armazenamento, foi realizado um estudo onde comparámos os preços com os tamanhos para diversas soluções. Para selecionar um sistema de carregamento, dois estudos foram conduzidos com objetivo de validar se a utilização de painéis fotovoltaicos seria viável.

A solução apresentada neste trabalho foi desenvolvida utilizando a plataforma para desenvolvimento de soluções IoT desenvolvida pela Fraunhofer Portugal - Kallisto. A solução apresentada utiliza BLE para receber comandos provenientes do surfista e Wi-Fi para enviar os dados adquiridos. Para garantir que o sistema se mantinha à prova de água, este utiliza painéis solares cobertos com resina epoxídica para alimentar o sistema.

Um estudo adicional foi realizado para tentar perceber qual o efeito da utilização de resina epossidica sobre painéis fotovoltaicos, e assim validar a viabilidade desta solução.

Com o prototipo desenvolvido, tentámos validar todas as estimativas utilizadas como base para efetuar as nossas escolhas tecnológicas.

Todos os requisitos propostos na introdução do trabalho foram concluídos. No entanto, existem várias áreas do trabalho que requerem melhoramento, como, por exemplo, o baixo desempenho do nosso sistema durante a transmissão de dados.

Abstract

The Internet of Things has democratized the use of sports' monitoring devices. Most of these devices are targeted to the fitness market, creating a lack of solutions for other sports.

This work focuses on the development of a low-profile solution to acquire data for a surf-monitoring algorithm developed by Fraunhofer Portugal. This device shall support long periods inside water without interfering with the surfer's performance.

Due to the project's requirements, the use of some technologies was a must: microcontroller, gyroscope, magnetometer, accelerometer and GNSS module. Nevertheless, other technologies like: the communication system, storage method and charging method had to be selected.

For the selection of the wireless communication system, the amount of acquired data had to be estimated in order to select a technology capable of providing a good data-rate vs. power consumption relation. To select the storage system, a survey relating the amount of storage and the prices for each solution was conducted. To select the charging method, two studies were conducted to validate the use of photovoltaic cells.

The solution presented in this work was developed using a Fraunhofer's development platform for IoT devices - Kallisto. This solution uses BLE as a command interface between the surfer and the device and Wi-Fi to transfer the acquired data. To guarantee the system's waterproofness, the system is powered by photovoltaic panels coated with epoxy resin.

An additional study to understand the effect of epoxy resin over the photovoltaic panels was conducted, validating the viability of this option.

With the developed prototype, all estimations were validated through a set of individual laboratory tests.

All the proposed requirements during the introduction of the work were met. Nevertheless, there is space for improvement in the transmission performance.

Acknowledgements

First of all, I would like to express my sincere gratitude to Associação Fraunhofer Portugal Research for hosting me, it was a pleasure to collaborate with such a great team. And to Faculdade de Engenharia da Universidade do Porto, that for the last 5 year, provided me with great knowledge and a lot of amazing moments that I will never forget.

This dissertation wouldn't be possible without my supervisors Sérgio Reis Cunha (FEUP) and Carlos Resende (Fraunhofer Portugal) for all the great suggestions and knowledge that have provide me. Eduardo Grifo (Fraunhofer Portugal), my co-supervisor, and João Oliveira (Fraunhofer Portugal) that helped me a lot during this work, specially in introducing me to Kallisto SDK and helping me with code-debugging. Diana Gomes (Fraunhofer Portugal) and Ricardo Graça (Fraunhofer Portugal) that helped me to understand better the working principle on Fraunhofer's surf profiling algorithm. Ruben Moutinho (Fraunhofer Portugal) that helped me prepare the experiments on the PV cells. And to all the other amazing people at Fraunhofer AICOS that were always available to help.

A special thanks to Filipe Sousa (Fraunhofer Portugal) for believing in me and allowing me to collaborate with Fraunhofer AICOS in such amazing projects for the past 2 years.

I would also like to thank all my friends and colleagues at FEUP, that have made this last 5 years one of the best periods of my life. And to all residents and ex-residents at Residência Jayme Rios de Sousa with whom i shared my life these same years.

Would also like to thank my family, specially my mother and father, for giving me support in this journey even being far away from me.

At last, but not least I would like to thank my country that has permitted to me and other people, without enough financial support, to follow their dreams, study and have the opportunity of dreaming with a better future.

Diogo Faria Correia

"Here's to the crazy ones.
The misfits. The rebels. The troublemakers.
The round pegs in the square holes. The ones who see things differently.
They're not fond of rules. And they have no respect for the status quo.
You can quote them, disagree with them, glorify or vilify them.
About the only thing you can't do is ignore them.
Because they change things.
They push the human race forward.
And while some may see them as the crazy ones, we see genius.
Because the people who are crazy enough to think they can change the world,
are the ones who do."

Rob Siltanen

Contents

List of Figures

List of Tables

Abbreviations

3GPP	Third Generation Partnership Project
AP	Access Point
AHRS	Attitude and Heading Reference System
BLE	Bluetooth Low Energy
CoP	Centre of Pressure
DC	Direct Current
EEPROM	Electrically-Erasable Programmable Read-Only Memory
eMMC	Embedded Multimedia Card
eDRX	Extended Discontinuous Reception
FEUP	Faculdade de Engenharia da Universidade do Porto
FhP	Fraunhofer Portugal
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HAL	Hardware Abstraction Layer
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
IoT	Internet of Things
ISA	International Surfing Association
ISM	Industrial, Scientific and Medical
LSTM	Local Standard Time Meridian
LTE	Long-Term Evolution
MCU	Micro-controller Unit
MEMS	Microelectromechanical systems
MPPT	Maximum Power Point Tracking
NMEA	National Marine Electronics Association
NB-IoT	Narrow-band for IoT
OMTP	Open Mobile Terminal Platform
PGVIS	Photovoltaic Geographical Information System
PRN	Pseudo-random noise
PSM	Power Saving Mode
PPS	Precise Positioning Service
PV	Photovoltaic
SB4S	Smartboard for Surfing
SD	Secure Digital
SPI	Serial Peripheral Interface
SPS	Standard Positioning Service
STC	Standard Testing Conditions
TCP	Transmission Control Protocol
UCS	Universal Charging Connector
UE	User Equipment
USB	Universal Serial BUS
WPC	Wireless Power Consortium

Chapter 1

Introduction

This chapter introduces the reader to the sport's monitoring systems topic, identifying the lack of surf-related solutions and proposing a new one. This chapter also sets the requirements for this solution and describes the work to be developed in order to achieve it.

1.1 Context and Problem Description

Athletes are always searching for new training methods in order to improve their performance. In the past, techniques such as video-based time-motion analysis were used to track human locomotion and improve sports performance. Unfortunately, these types of systems were labour intensive, offered very questionable results and could not provide real-time information [2]. The advent of global position system (GPS) and inertial measurement units (IMUs), made possible the development of more efficient and accurate solutions for measuring athletes' and teams' performance. More recently, with the decrease in the price of such sensors, these solutions have become more accessible to the consumer market, resulting in a new cultural phenomenon, the quantified-self movement. Fitness-oriented devices represent the majority of this market [3]. As a result, there is a lack of other sports-oriented devices in this market. Despite this, several attempts to replicate this success with other sports have emerged more recently.

One of the sports that have started raising attention to its yet immature technology market is surf [4], which has seen an increase in popularity over the last decade. According to the International Surf Association (ISA), there are now over than 35 million surfers worldwide [5], and market associated with it expects to grow 12,24% between 2018 and 2022 [6]. A few solutions are already available on the market. However, most of them are position based only which, considering the nature of the activity, represent an issue on metric reliability [4].

Trying to solve this issue Fraunhofer Portugal (FhP) has developed a smartphone-based solution capable of detecting wave, paddle, sprint paddle, dive, lay, and sit events with 88.1% of accuracy [4]. FhP's algorithm uses a sensor fusion approach based on Madgwick's orientation filter [7] to overcome several individual sensor limitations. Despite solving the issues that are present on many of the market solutions, this solution is not ideal. The overall cost of a smartphone-based

solution associated with the risk that seawater represents to most electronic devices and the fact that an extra piece of equipment, not designed for the purpose, may in some instances be uncomfortable, affecting the practitioner's performance, are some issues in this type of solutions [8].

1.2 Aims and Objectives

Smartboard for Surfing (SB4S) aims to solve the issues associated with smartphone-based solutions by developing a less-intrusive and waterproofed hardware device compatible with FhP's existing algorithm. To achieve these goals, our solution must:

- not affect the surfer's performance
- be comfortable to use (small and light)
- work during an entire surf session
- have all the sensors to provide the required data to the algorithm
  - Positioning
  - Accelerometer
  - Gyroscope
  - Magnetometer
- have enough computing power to:
  - acquire the data
  - store the data
  - transmit the data

Another important goal to take into consideration is its price. For an immature market such as surf monitoring, an expensive solution would difficult its acceptance by the surf community and restrict the market to some professional surfers only.

This dissertation aims to build a design specification from these requirements and implement a prototype of the system. From the requirements mentioned at the beginning of this section, it is already possible to establish some of SB4S's specifications, e.g. the use of an accelerometer. To complete the specification, a survey must be conducted on communication technologies, storage and charging methods:

The communication technology should: Provide enough throughput to send the acquired information to an external device, from now on called monitor, capable of proceeding with the analysis of the acquired data. This technology should also have low-power consumption.

The storage method should: provide enough capacity, to store at least one session of data acquisition. It should also be reliable during mechanical stress and provide a good expected lifetime.

The charging method should: Be practical to use. It must also be able to provide the required amount of power to SB4S's operation and to charge its battery.

To accomplish these objectives, SB4S will be built on top of FhP's Kallisto IoT development platform. For more details, a description of this platform can be found in section 2.3.1.

1.3 Document Structure

This document is organised as follows:

- **Chapter 1:** Introduction of the problem and brief discussion about the requirements and specifications of the system.
- **Chapter 2:** Survey on related commercial and academic works. Identification of advantages and disadvantages of each work. Description of technologies in the specification. And technologies comparison.
- **Chapter 3:** Brief description of the algorithm used to analyse the acquired data. And description on the selection of technologies.
- **Chapter 4:** Description of the development of a functioning prototype. It describes the selection of the different modules, the dimensioning of a photo-voltaic charging system and the development of the firmware
- **Chapter 5:** Description on the methodology used to test the prototype and results from the tests.
- **Chapter 6:** Conclusions from the project and future work

Chapter 2

State of the Art

This chapter discusses the present state of surf monitoring technologies. In the first section, similar solutions (commercial and academic) get introduced and their advantages and disadvantages discussed. Other types of solutions (e.g. video) are addressed in the second section, together with relevant literature on the topic. And at last, we will survey different technologies (to be considered in the final specification) and briefly describe their working principles.

2.1 Related Work

2.1.1 Commercial

Rip Curl Search GPS [9] uses GPS to provide surf-related metrics: the number of waves, top speed, session's duration and waving time. As said in section 1.1, systems of this type tend to be imprecise and unreliable on its measurements when monitoring short, but high-intensity, periods of activity [10]. Even so, a second version of the product was released indicating a relative good acceptance by the surf community. Similar to it's first version, the second version used a standalone GPS to profile the surf session, resulting in the same issues of the previous version.

Glassy Pro tried to solve this issue by complementing the GPS' acquired data with inertial information obtained from the user's wrist. This approach has allowed a more accurate identification between waves and paddling. Just like Rip Curl Search GPS, this system is provided in a wristband format. This format, the most popular for wearables [3], grants familiarity with the users making the device more acceptable for the masses. Despite this, this approach isn't considered the most suitable for surf, preventing the acquisition of enough data for wave performance analysis [4].

Despite offering a more complete solution than the *Rip Curl Search GPS*, the *Glassy Pro* is no longer available on the market. The company announced a deal with *CASIO* and shifted its focus towards the development of a dedicated surf monitoring app for *CASIO* smartwatches.

Trace Up was a dedicated device designed to be placed on the top of the surfboard. It used GPS and an IMU to identify complex surf movement during a session. This device could provide a handful of useful features for surf monitoring, however, no performance studies were found to

validate its detections and measurements. The company no longer produces this solution and it has changed its focus solely to football monitoring solutions [4].

Slide-R [11] started as a *Kickstarter* project [12] only reaching 60% of its funding goal, even tough according to the last update on the *Kickstart*'s page the company has decided to continue producing and selling the Slide-R hardware for the price of €249,00 . According to its *Kickstart*'s, page Slide-R uses GPS and IMU to track the different activities during a surf session. It uses WiFi and BLE to communicate with a smartphone app and can be charged through either a photovoltaic panel or induction charging. This solution is, so far, the one that shares more similarities with SB4S, however, just like *Trace Up* there are no performance studies that validate its detections and measurements. Since the last update on *Kickstarter*, there is no additional information about the product. In the author's opinion, the lack of success in this solution dues to its high target price for a startup with a poorly known brand. This solution, when compared to Rip Curl's solution, is cheaper and can be considered technically superior. However, it's important to emphasise that Rip Curl is a very well-known brand in the surf market and that the chosen format, if needed, could be used as a simple watch.

Pukas SurfSense [13] was a project developed by *Pukas Surf* and *Tecnicalia* in 2011. The main objective of this project was to gather data in order to improve future surfboard designs [14]. It was controlled by a single board computer (SBC), an IGEPv2, and had various sensors like a pressure sensor, a flex sensor, a compass, GPS, and an accelerometer [15]. There is little information about the project available and like other solutions, there are no studies that could evaluate its performance.

In conclusion, the majority of attempts to develop a commercial solution have failed. To the author's knowledge, the only commercial attempt with relative success was the solution developed by RipCurl, a device far from being technically ideal to the purpose. The reason for this success may be related to the fact that RipCurl is already a well-known brand in the surf's market.

2.1.2 Academic

In 2014, a work called "Instrumentation of a Surfboard to Evaluate Surfing Performance" [16] developed a similar system to instrument a surfboard aiming to evaluate wave riding performance. This work focused on the possibility of determining the centre of pressure (CoP) and the pressure distribution over the surfboard from feet positioning. Their system was composed of twenty-four piezoresistive sensors and an IMU for data acquisition only. During their work, the IMU was not tested, and was not possible at the time to compare the data acquired from both, the piezoresistive and IMU sensors.

On an article published on the 2015 1st URSI Atlantic Radio Science Conference [17], a group of researchers from the University of Aveiro developed an electronic device with an accelerometer, a GPS, a temperature sensor and some capacitive paper sensors. The device was capable of reading the height and speed of the wave, the position of surfer's feet on the board, the distance travelled and the GPS coordinates of the surfer. All the acquired data was then used to represent the surfboard's motion with a 3D model of the surfboard.

2.2 Literature Review

Sport's performance can be measured by tracking external and internal loads [18]. External load measures the work performed, and the internal load describes the impact of this workload on the individual. There are a large number of methods that can be used to measure external load: strain gauges, speed sensors, time-motion analysis, GPS tracking and other video-based systems. Despite this variety, the majority of surf-related literature focuses on the performance analysis using GPS and time-motion analysis [18].

GPS systems are usually used in combination with accelerometers, gyroscopes, digital compasses, magnetometers and temperature sensors to provide accuracy, as their standalone use is only reliable to some extent. Without it, errors during high speed and rapid movements are expected [19]. Despite this, GPS alone is able to provide the speeds obtained in the paths, measure the total distance travelled, the total time of a session and the time spent on each one of the activities[19].

Video-based time-motion analysis, consists in the examination, frame-by-frame, of video footage of athletes during train or competition [20]. As already mentioned, in section 1.1, its use is labour intensive and cannot provide real-time support. In addition to this, when compared to solutions like SB4S, it requires a fixed monitor on the beach making its use not suitable for non-professional surfers.

Some of the methods, for measuring internal load, that have been used in sports include blood lactate, training impulse and heart rate. Due to surf literature being in its infancy [18], there are only studies on the use of heart rate to monitor surf performance. Heart rate measuring suffers from many variables (e.g. surfer's fitness level), its measures need to be interpreted with caution. In addition to this, the limited number of studies in the subject makes it difficult to establish definitive parameters for heart rate profiling in the surf, making its use not suitable, yet, to the consumer market.

2.3 Technologies

2.3.1 FhP's Kallisto Platform

Kallisto is a platform developed by Fraunhofer Portugal that aims to streamline all the process in developing a device for the Internet of Things. The Kallisto Platform is composed of two components: the Kallisto Board and the Kallisto SDK.

Kallisto Board is based on the Nordic Semiconductor's nRF52 Series SoCs and integrates with it a variety of sensors, radio communication and energy management modules and other peripherals.

Kallisto SDK sits on top of the Nordic Semiconductor's SDK, making the development of new IoT devices easier, trough a library of HALs (Hardware Abstraction Layers), Drivers and other libraries (power management, a watchdog, etc.).

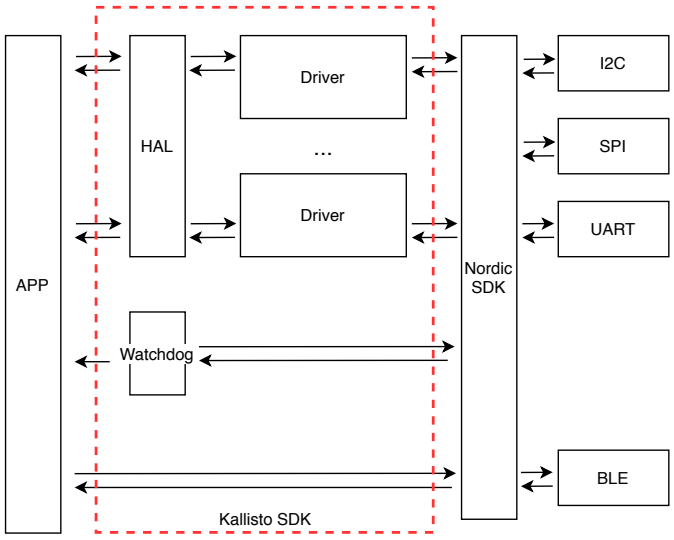


Figure 2.1: Kallisto SDK Internal Behaviour

2.3.2 Global Navigation Satellite System

Global Navigation Satellite System, or GNSS, is a term used to refer to any constellation of navigation satellites capable of providing global coverage. This term is frequently interchangeably used with GPS (Global Positioning System), the oldest and most popular system of the type. Besides GPS, there are other satellite systems available at the moment: Glonass, Galileo and Compass/BeiDou. All these systems share the same working principles, and can be used alongside each other. For ease of explanation, GPS will be used as an example of a GNSS system during this state of the art. This choice relies on the fact that GPS is the oldest and most popular of the systems, serving as a base for the newer ones.

2.3.2.1 Global Positioning System

The global positioning system (GPS), previously called NAVSTAR GPS[21], was developed by the U.S. military in order to provide positioning, navigation and timing services at any point of the globe during the cold-war.

The system was initially composed of a 24 satellites constellation and provides two levels of positioning service: Standard Positioning Service (SPS), for civil use, and Precise Positioning Service (PPS), for authorised use only [21].

In the beginning, SPS could only provide positioning with a horizontal accuracy equal to or better than 100 meters. This error, selective availability, was intentionally added to the service to prevent precision weapon guidance [22]. This feature was discontinued in 2000, and nowadays, it is possible to achieve accuracy with values in the order of the few meters [23].

GPS signals are transmitted in two primary carrier frequencies: L1 (1575.42 MHz) and L2 (1227.60 MHz). The majority of the commercial solutions can only receive using L1 band, however, there are also some dual-band receivers available on the market. The later ones tend to be

more expensive, but are more precise. Dual-band receivers can infer information about the atmosphere from the differences between the carrier signals. This information is then used do apply corrections over the measurements, in opposition to single frequency receivers that apply corrections using average atmosphere conditions instead. The accuracy of a single-frequency receiver reaches values in the interval of 1 to 10 meters, and dual-band receivers values under 1 meter.

2.3.2.2 Working Principle

In this chapter, a brief explanation of the GPS working principle will be shown. For more information on the subject, please watch the YouTube video by Dr. Medeiros [1].

Acquisition and Precision Codes

Each satellite in a constellation broadcasts its messages trough a direct-sequence spread spectrum communication system, in which messages are exclusive-ored with a pseudo-random noise (PRN) code, also known as ranging code. Replicas of these, well-known, codes are created in the receiver and then used to infer the distance between each satellite and the receiver by auto-correlating the expected code with the received one (Figure 2.2). Each code, from the family of codes called gold codes, is unique to its satellite and doesn't correlate well with the other codes. In fact, gold codes are well known to strongly auto-correlate only when exactly aligned with themselves.

There are two types of codes available on GPS: acquisition and the precision codes. The main difference between them is their chip-rate, rate at which each code is transmitted. Precision code has the highest chip-rate, making it more precise than the acquisition code. However, this code is encrypted and is only available to authorised entities.

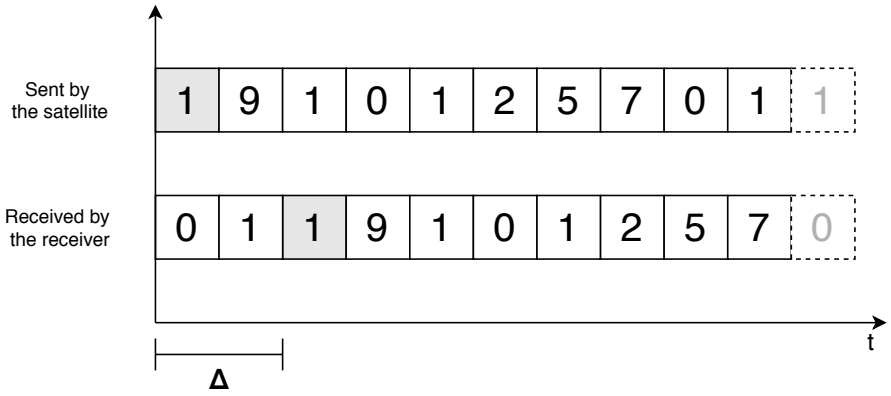


Figure 2.2: Example difference in phase on acquisition and precision codes

Navigation Messages

Knowing the distance between a set of satellites and the receiver isn't enough to calculate its position on the globe. The receiver must also know the time and position of each active satellite.

[1] GPS RTK Surveying Workshop by Dr. Medeiros at University of Central Florida [video], https://www.youtube.com/watch?v=vOJ3u7Zd_i0, (accessed at 20 April 2020)

This information is encoded into messages, called navigation messages, which are modulated onto both the acquisition and precision ranging codes. Navigation messages' information can be characterised into three types:

- GPS Date and Time and Satellite Status
- Ephemeris (satellite position)
- Almanac (info about the constellation itself)

Triangulation

The distance between the receiver and each satellite of a set with, at least, three satellites. As well as the position of each one of the satellites around the planet. Are needed to obtain the position of the receiver on the globe. The precision of this measurement increases with the number of satellites observed by the receiver.

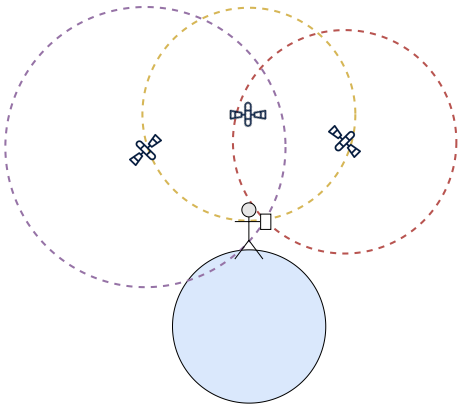


Figure 2.3: GPS Triangulation

2.3.2.3 GPS positioning modes

Positioning with GPS can be performed by either two ways: point or relative positioning.

Point positioning, or autonomous positioning, is the primary method of obtaining the receiver's positioning on the globe and was already described in the previous section.

Relative positioning, or differential positioning, requires at least two GPS receivers simultaneously tracking the same satellites. One called the rover, and which coordinates are unknown. And one or more receivers, with precisely known coordinates, who play the role of stationary bases. This positioning mode provides higher accuracy than that of autonomous mode. Depending on whether the measurements are inferred by the carrier-phase or PRN, they can achieve an accuracy level of a centimetre to a few meters.

There are a lot of different techniques that are used to achieve relative positioning measurements, some of them are real-time others require post-processing, some require the rover to be stationary during the collection of data others allow the rover to move. Some techniques use virtual stations instead of a physical one, but this type of techniques require the receiver to be connected with a network of any type.

2.3.3 Attitude and Heading Reference System

Attitude and heading reference system, or AHRS, are systems composed by an accelerometer, a magnetometer and a gyroscope. These sensors can be either solid-state or implemented as a micro-electro-mechanical system (MEMS). The use of an AHRS overcomes some problems inherent with the usage of a single IMU. AHRSs are able to provide measurements based on their relative orientation to the direction of gravity and earth's magnetic field in opposition to IMUs that can only measure their attitude relative to the direction of gravity [7].

Accelerometers are used to measure accelerations and can be used for sensing vibrations in systems or for orientation applications. There are two types of direct-current (DC) accelerometers: the capacitive and the piezoresistive. Accelerometers of the first type contain plates, some fixed, others attached to springs that move as acceleration forces act upon the sensor. The variation in these plates' position changes their capacitance relative to the others. Through this capacitance changes, it is possible to measure the acceleration forces applied to the sensor. Piezoresistive accelerometers base their measures from the output voltage of piezoresistive material, a crystal that outputs voltage under mechanical stress [24].

Gyroscopes are measuring devices that can measure angular velocity. Electronic gyroscopes use a suspended resonator that resonates along the x-axis, co-planar to the substrate, the Coriolis force produced by an angular rate on the y-axis makes the resonator to vibrate on the z-axis. This changes the capacity between the resonator and the substrate, which is used to calculate the angular velocity [25].

Magnetometers are used to measure the intensity of magnetic fields. They can be categorised in many types, accordingly to the physical effect in use, the more relevant types are: inductive, magnetic or magnetoresistive. Inductive magnetometers measure the field based on the current induced by them. Magnetic magnetometers estimate the magnetic field by measuring the Lorentz force of a magnet. Magnetometers that measure the magnetic field based on the resistivity of a conductor that changes with the presence of that field are called magnetoresistive [26].

2.3.4 Charging Technologies

In this section, some of the most popular charging approaches will be discussed. The chosen method will need to provide enough power to the system and at the same time guarantee its waterproofing design. To guarantee the first, the best solution would pass on using a wired system. However, an exposed interface, like a USB (Universal Serial BUS) port, could compromise the device's waterproofness. Even though a waterproof cover could be applied to the interface, it's reliability could be questioned when completely submerging the system. Other approaches like wireless charging and the use of photovoltaic panels could solve this problem, however, it is necessary to verify if they can ensure SB4S's power requirements.

2.3.4.1 USB charging

A few years ago, the majority of portable devices had proprietary charges. In 2009, together with 17 mobile operators and manufacturers, the GSM (Global System for Mobile Communication) Association announced to be committed in the implementation of a cross-industry standard for a universal charger for mobile phone. The Open Mobile Terminal Platform (OMTP) would be responsible for conducting this new specification, the universal charging connector (UCS). The OTMP adopted micro-USB, and more recently the USB type-C, as the universal interface for the UCS [27].

Only after the release of UCS, USB protocol would be updated to support power transference with the battery power specification. This update permitted to the standard the capability of providing charge up to 7.5W@5V, making USB a reliable charging method. Nowadays, with the even newer update, the Power Delivery Standard it's possible to supply laptops and displays up to 100W [28].

Even being the most reliable charging standard on this study, the use of a USB port as a charging interface would difficult the implementation of a waterproofing solution to our device.

2.3.4.2 Qi charging

The main advantage of wireless charging methods over cord-based methods is the possibility of waterproof product designs. They can use inductive or capacitive methods to transfer power between a transmitter (charger) and a receiver (device). Inductive chargers, like Qi, are more common than capacitive ones, due to this latter requiring relatively large coupling areas [29].

Qi is a wireless power transfer standard, developed by the Wireless Power Consortium (WPC), capable of transferring up to 30W of load power. Before the power transmission starts, the receiver and the transmitter agree with the quantity of power load. Every charger must be capable of providing at least 5W. Another specification of the standard is the outer diameter of the coils on the receiver (50mm) and the transmitter (40mm) [30]. Since its launch, in October 2010, the WPC has adopted the following key features for the standard:

- inductive wireless charging;
- vertical-flux approach;
- free or guided positioning (without any movable mechanical parts);
- localised charging;
- communications between loads and charging pad;
- item identification and compatibility checks.

2.3.4.3 Photovoltaic charging

Other solution that would enable to easily implement waterproofing, besides the use of the Qi standard, would be the use of photovoltaic (PV) cells.

PV cells can generate electrical current from the photons emitted by the sun. When absorbed by the semiconductor material, solar photons react with the electrons on the valence bands and generate electron-hole pairs. However, these pairs are meta-stable, since they tend to recombine. PV cells usually are made of two different semiconductor materials, a p-type and an n-type, to reduce this effect. This method is not ideal, as electron-hole pairs generated further to the p-n junction have fewer chances of being collected by their counterpart. This separation of charges is responsible for creating a potential difference between the two materials [31].

The major downside of this approach is the fact that energy harvest results heavily depends on the efficiency of the cells (η) and the amount of sunlight on the location of the harvesting [32].

The I-V curve of a solar cell varies with the amount of available sunlight. One way of optimising the production of energy by a solar system is controlling its maximum power point, by changing the load of the system. A maximum power point tracking (MPPT) unit is a device that tracks the variation of the I-V curve and tries to correct the solar panel's load. MPPT units can be designed using more than 19 different approaches [33]. These methods can be, so far, classified into two major groups: large-scale and small-scale PV power systems. The first type makes use of digital controllers, in opposition to the latter that is less accurate, but cheaper with advantages in PV applications below 50W [34].

2.3.5 Communication Systems

In order to monitor the athlete's performance, all data gathered by the system will need to be transferred to an external device (computer, smartphone or tablet) and be analysed by an entity (the coach, a judge or even the athlete itself). This analysis could be done in real-time or after the session. On the first use case, data would be sent in small messages resulting in lower throughput requirements. However, to support the transmission of data during a surf session some kind of wide range network, like cellular, would be required resulting in SB4S overall power consumption's increase. In contraposition with these requirements, the second use case would need to transmit an entire session resulting in the need of a system capable of providing a higher data throughput despite its range.

2.3.5.1 Long Term Evolution (LTE)

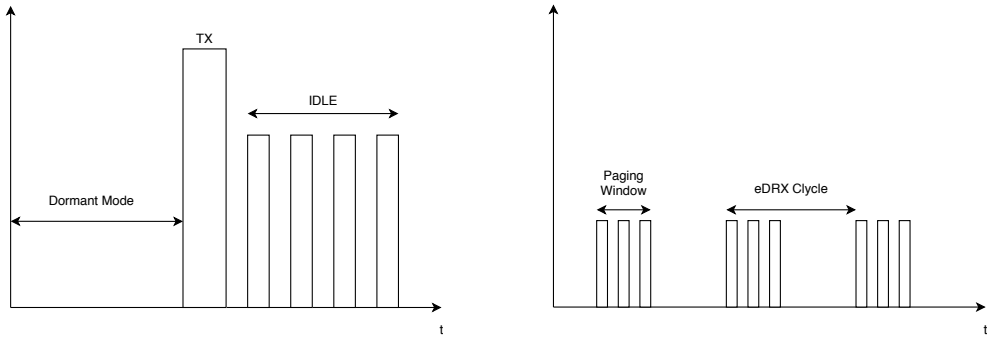
LTE was introduced by the Third Generation Partnership Project (3GPP) in the Release 8 to ensure the continuity of competitiveness of the 3G system, increasing its data rates and quality of service. Until the advent of NarrowBand-IoT, more information about it later, LTE and other regular cellular technologies were the optimal options for very long-range applications. LTE's power consumption is relatively high, making it not suitable to use in low-power portable devices

[35]. On the other hand, LTE can offer data rates in the range of 3 to 10Mbps, being optimal for applications that use high amounts of data (ex. video-vigilance systems).

2.3.5.2 Narrowband Internet of Things (NB-IoT)

To address the lack of IoT's wide-area network (WAN) solutions, 3GPP launched on its release 13, a new system called Narrowband IoT. On this release, 3GPP set 5 objectives for Machine-Type Communication (MTC): higher energy efficiency, enhanced indoor coverage, lower terminal complexity and cost and support for various latency features [36]. At its peak, NB-IoT can offer 26Kbps (downlink) and 66Kbps (uplink, multi-tone) data rates [37].

The specification of the NB-IoT was heavily inspired on LTE, reducing the development time of the specification and, as a consequence, of the time to develop a product that uses it [38]. The major differences, between the two technologies, are the inclusion of the Extended Discontinuous Reception (eDRX), in Release 13, and the Power Saving Mode (PSM), in Release 12. The first one allows user equipment (UE) devices to stay longer in sleeping mode between paging cycles (see Figure 2.4b). The latter one allows the UE to decide, based on some logic or timer, that it is time to transmit or not. After transmitting the UE remains in RX mode for a while so it can be reachable if needed (see Figure 2.4a).



(a) PSM (b) eDRX

Figure 2.4: Power saving mechanisms of NB-IoT

2.3.5.3 Bluetooth Low Energy (BLE)

BLE was developed as a complementary technology to classic Bluetooth and should not be considered an evolution or sub-type of the latter, even borrowing a lot of technology from its parent [39]. BLE was designed with low energy consumption and low-cost requirements in mind. BLE data rates range from 125 Kbps to 2 Mbps, contrasting to the supported data rates from 1 Mbps to 3 Mbps in classic Bluetooth [40].

Its power consumption its highly dependent on how frequently the radio works. Tei, R. et al in [1] made a power consumption estimation for BLE peripherals assuming the connection state. In a connection event, the central (master) sends periodically pulling requests to the connected

peripherals (slaves) and waits for a response. The time between those two events is defined as "Connection Interval" [39]. To lower its power consumption, a peripheral may be allowed to stay asleep and ignore the pulling requests from its master for a determined period of time, defined as "Slave Latency" [1]. It is important to emphasise that the power consumption depends also on the wireless circuit and Micro-controller Unit (MCU) utilised. The results from this study can be consulted on the Table 2.1.

Table 2.1: Tie, R. et al. Power Estimation Results [1]

Effective Connection Intervals (s)	Average Consumptive Current (μA)	Battery Life for a CR2032 (days)
0.0075	3312.1	3
0.1	248.9	37
0.2	124.7	74
0.5	50.2	183
1	25.3	362
2	12.9	711
5	5.4	1682
10	3.0	3092
16	2.0	4508

BLE uses frequencies on the 2.4GHz industrial, scientific and medical (ISM) band, that has very poor propagation characteristics especially in water, which may constitute an issue in the context of SB4S.

2.3.5.4 WiFi

IEEE 802.11, commonly known as WiFi, is a family of wireless communication protocols released in 1997 as an alternative to the high installation and maintenance costs inherent to wired LAN infrastructures [41]. Comparing with BLE and NB-IoT, WiFi is the best choice when considering data throughput as the unique criteria. The first protocol in the family, the now called Legacy 802.11 protocol, is able to provide two raw data rates of 1 and 2 Mbps [42]. And the most recent on the family, the IEEE 802.11ax, is able of providing raw data rates starting on 143Mbps up to 2.4Gbps [42]. The major drawback of using any of these protocols is their's high power consumption [35].

2.3.6 Storage Systems

In this section, some of the most popular data storage methods: EEPROM, Flash memory, SD Card and eMMC will be compared with each other.

2.3.6.1 EEPROM

Electrically-Erasable Programmable Read-Only Memory (EEPROM) is a type of non-volatile memory that can be written or erased in-circuit by applying special programming signals. "EEPROM reliability is limited by endurance, ie, the number of reprogramming cycles that can be accomplished and still comply with device specifications" [43]. EEPROM memory cells are composed of two transistors: storage and select, allowing good endurance performances with more than 10^5 P/E cycles. On the other hand, two transistors per cell require larger areas consequently making this type of memories more expensive [44]. For this reason, they are usually sold with storage sizes in the order of one megabyte. These type of memories are usually erasable and programmable per single byte.

2.3.6.2 Flash memory

Flash memory is other type of non-volatile memories usually found in SSD, SD Cards and eMMCs. It's working principle derives from EEPROMs, differing on the number of used transistors. Flash-memories are composed of a single transistor and, therefore, are cheaper than EEPROM memories. When compared to the latter, it's major disadvantage stays in its lower endurance, being only capable of achieving less than 10^4 P/E cycles [44].

2.3.6.3 SD Card

Secure Digital (SD) is a proprietary non-volatile memory card format developed to be used in portable devices. They can resist water, bending, torque, electrostatic discharge and up to 10,000 different insertions [45]. SD cards utilise flash memory cells to store data, making them a very cheap solution. SD cards can be sold within a broad price range, that may vary with the type of NAND flash used, controller robustness, minimum operating temperature, etc. [46] They are sold with storage sizes in the order of one gigabyte and can deliver speeds up to 624 MBps depending upon which host device they are being used [45].

2.3.6.4 eMMC

Embedded MultiMediaCard (eMMC) is an embedded, non-volatile and low power memory standard based on MultiMediaCard and delivered on a BGA format. After the dissolution of the MultiMediaCard Association, all the assets were transferred to JEDEC Solid State Technology Association who is now responsible for the development of the standard [47].

Just like SD cards, they also use flash memory cells to store data and can be more reliable, when comparing to the first, due to the fact of being soldered directly into the PCB. They are also smaller and more power-efficient than SD cards. However, they are more expensive than SD Cards, which derives from the fact that eMMC is subject to NAND flash market situation [46]. They are sold with storage sizes in the order of one gigabyte and can deliver speeds up to 400 MBps [48].

Chapter 3

Technological Selection

In this chapter, the methodologies used to develop SB4S's specification and select the modules to use in the final design of the system are described. This process took into consideration, the requirements in section 1.2, the use of FhP's Kallisto Platform and the FhP's Surf Profiling Algorithm presented in [10] [4].

Section 3.1 addresses the estimation of the amount of produced data, by the system. This study served as a basis for the selection of the technologies in the communication system and the storage method.

In sections 3.2 and 3.3, we will discuss the advantages and disadvantages of each one of the technologies and describe the methodology used to select the most suitable one.

3.1 Surf Profiling Algorithm

The algorithm developed by FhP [10][4] uses the smartphone's accelerometer, gyroscope, magnetometer, and GNSS data to detect the surfer's movements. Then this data is grouped into 1-second length windows with 50% overlap, with 100 samples from each sensor in the AHRS module and two from the GNSS module, resulting in the following data output requirements:

Table 3.1: Algorithm's Data Output Requirements

	Accelerometer	Gyroscope	Magnetometer	GNSS
Output Data Rate (ODR)	100 Hz	100 Hz	100 Hz	1 Hz

The quantity of data produced, and then transmitted, was estimated based on the data sizes in table 3.2. The values on the table were obtained from the respective datasheets, for the AHRS sensors, and from the NMEA 0183 standard RMC messages [49], for the GNSS module. NMEA 0183 is a standard, created by National Marine Electronics Association (of U.S.A), that defines the structure of the messages to be sent/received by nautical instruments. For historical reasons, this nautical standard has become the default for interfacing with GNSS modules.

Table 3.2: Data size estimation

		Data Size	Data Type
Accelerometer (Kallisto)	x axis	16 bits	uint16_t
	y axis	16 bits	uint16_t
	z axis	16 bits	uint16_t
Gyroscope (Kallisto)	x axis	16 bits	uint16_t
	y axis	16 bits	uint16_t
	z axis	16 bits	uint16_t
Magnetometer (Kallisto)	x axis	13 bits	uint16_t
	y axis	13 bits	uint16_t
	z axis	15 bits	uint16_t
GNSS (NMEA standard)	timestamp	8 bytes	uint64_t
	longitude	7 bytes	char + uint16_t + float
	latitude	7 bytes	char + uint16_t + float
	speed	4 bytes	float
	bearing	4 bytes	float

The duration of a surf session depends on a lot of factors: the athlete's physical condition, type of session (professional or amateur), weather conditions and others. In some cases, surf sessions may last to over 4-5 hours [50].

Using these values as a reference, SB4S would produce 1830 bytes each second (eq. 3.1). That, after a session of 5 hours, would result in around 33Mb (eq. 3.2) of raw data.

$$(100 \text{ samples} * 3 \text{ sensors} * 3 \text{ axis} * 2 \text{ bytes}) + 8 \text{ bytes} + 7 \text{ bytes} + 7 \text{ bytes} + 4 \text{ bytes} + 4 \text{ bytes} = 1830 \text{ bytes} \tag{3.1}$$

$$1822 * 60 \text{ seconds} * 60 \text{ minutes} * 5 \text{ hours} = 32940000 \text{ bytes} \approx 33 \text{ Megabytes} \tag{3.2}$$

3.2 Transmission of Acquired Data

Before comparing technologies is necessary to stipulate our objectives for the communication system and which methodology to use during the survey itself. How can we deliver the acquired data to the monitor? To begin with, we can assume two different data transfer approaches: real-time and post-surf-session.

The first one would require the monitor to be statically close to the surfboard or to have a stable connection with an external database. However, the option of having a static monitor isn't worthwhile. There is no study, to the author's knowledge, on the distance between the surfer and the coastline, making it difficult to estimate this distance. In addition to this, a fixed monitor on the beach would require a second person to surveil the device what, in the author's opinion, could be considered an obstacle to the device's adoption. In the other hand, the use of an external database would add costs to SB4S's operation such as server's maintenance costs and network carrier costs which could also be considered an obstacle.

In the second approach, the surfer would have to synchronize both devices manually after the surf-session. This requires the selected communication system to be able to transfer the acquired data in a feasible amount of time (10 minutes[1]).

To select one of the technologies purposed on the state-of-the-art, two criteria were established: the theoretical transmission duration of the acquired data for a 5-hour session (t_{TX}) (Eq. 3.3); And the energy consumption during its transfer (E_{TX}) for the module from the survey with the least power consumption (Eq. 3.4). E_{TX} is obtained from the multiplication of the transmission time (t_{TX}), the module's voltage (V_{TX}) and current (I_{TX}) during transmission.

$$t_{TX} = \frac{\text{5-hour raw data size in Megabytes}}{\text{theoretical data rate in Mbps}/8 \text{ bits}} = \frac{33}{\text{theoretical data rate in Mbps}/8} \quad (s) \tag{3.3}$$

$$E_{TX} = t_{TX} \cdot V_{TX} \cdot I_{TX} \quad (Wh) \tag{3.4}$$

This study didn't consider LTE as a valid option due to being easily substituted by NB-IoT, a more adequate solution for a low-power device such as SB4S.

3.2.1 Bluetooth Low Energy

BLE data rates can range from 125Kbps to 2Mbps [40]. Even if, in the best conditions possible, BLE would take 3 minutes to transfer a 5-hour surf-session in a post-surf-session mode. This value could increase up to 10 times more.

Other option would be its use in a real-time approach. In this scenario, BLE would require an additional monitor closed to the beach. This wouldn't be ideal as its range may vary "(...) from more than a kilometre down to less than a meter" [51], depending on factors such as its surroundings, radio performance and antennas.

From all the surveyed technologies, BLE has the least current consumption of them all. The transfer of a 5-hour surf-session would consume at its peak around 10.8mA [52], resulting in a power consumption of 23.4 mWh.

3.2.2 WiFi

As already mentioned in section 2.3.5, IEEE 802.11 is a family of wireless protocols, each one with different maximum theoretical data rates and signal ranges. Through the module's survey, see Appendix C.1, is possible to infer that the majority of the modules available on the consumer market support IEEE 802.11 b, n and g protocols. Unlike BLE, the author didn't found a minimum theoretical data rate for any of these protocols, for this reason, the author decided to use the least maximum theoretical data rate as the worst-case scenario. This approach is far from being ideal, however, is considered by the author suitable for the purpose of comparing technologies.

[1] Value defined by the author

Using 1Mbps as the worst-case scenario for transferring, a 5-hour surf-session to the monitor, and the equation in 3.3, is possible to estimate that this transference would last for about 4 minutes and 24 seconds.

What about its current consumption? As already mentioned in section 2.3.5, this is the major downside of using WiFi to transfer data in low-power devices. Using ESP32-WROOM-32 as a reference, this modules uses an average of 140 mA for transmitting data (current consumption will vary with the power of the WiFi antenna), would use 35.28 mWh for transferring the 5-hour surf session. The use of WiFi to transmit the acquired data only is feasible, as there are batteries with 100 times more capacity than required. However, SB4S requires the selected communication technology to be used as an interface between the surfer and the device, and therefore to be always on.

3.2.3 NB-IoT

NarrowBand IoT data rates can reach up to 26Kbps (downlink) and 66Kbps (uplink), at its peak. However, even in this condition, the technology would not be able of providing enough throughput for a post-session approach. Using the equation in 3.3 is possible to estimate that NB-IoT would last about 67 minutes, to transfer of a 5-hour surf-session.

In the other hand, NB-IoT could be ideal to use with the real-time approach, as the need for an external monitor would not be required. Unfortunately, even using the most power-efficient module on the survey (Nordic's nRF9160), NB-IoT would consume a high amount of current (540 mWh for one hour of transmission [53]).

In opposition to the other studied technologies, NB-IoT would require the use of an external server, for both real-time as post-surf approaches, requiring additional operation costs.

3.2.4 Conclusions

When taking in consideration the theoretical maximum data rate for each technology, BLE could be considered an adequate option for SB4S's communication system due to two factors: being already implemented on Kallisto Platform and having a good power consumption. At the same time, BLE is the only technology that the author could find a minimum theoretical data rate which reveals that, at the worst-case scenario, BLE could last up to 36 minutes to transfer an entire 5-hour surf-session.

From the three technologies in the study, NB-IoT has shown to be the least suitable solution for our system. Its use would require a large battery, affecting the overall system's weight, and additional costs (carrier and server maintenance) to the use of the device.

Wi-Fi has shown to be the fastest solution of the three, being capable of providing an adequate data rate for transferring all the acquired data. However, just like NB-IoT the use of this solution would require a large battery affecting the surfer's performance.

For this work the author purposes a mixed-solution, using Wi-Fi and BLE, trying to combine the advantages of both solutions. In this fourth solution, the device would use the Wi-Fi system

only to transfer the acquired data and the BLE as an interface between the user and the device. In this fourth option Wi-Fi would only consume power during the transference, which for the small duration of it would not highly affect the overall power consumption of the system.

3.3 Choosing the Storage Method

As estimated in section 3.1, for a 5-hour surf-session, SB4S would produce about 33 Megabytes of raw data. To select the best storage method for SB4S, we will take into consideration two factors: how many sessions can we store into a module and its price. As a reference for the price and data sizes of storage modules (SD, eMMC and EEPROM) available on the market, the author has decided to use the information present on FhP's default electronics supplier, Mouser [2].

The EEPROM module with the largest capacity had only 1MB of storage. To store an entire 5-hour surf-session, 33 of these modules would be needed. This module had the cost, at 23/02/2020, of € 2.65 each unit making a 33 modules array cost € 87.45. For the amount of delivered storage, when compared to other solutions, the use of EEPROM is considered, by the author, excessively expensive.

eMMC storage capacity starts at 512MB and SD Card at 128MB. Both technologies can reach up to 256GB. Both technologies' starting capacity can store more than one 5-hour surf-sessions (about 3 sessions for the SDCard and 15 sessions for the eMMC module).

When comparing the price of both solutions, there is no clear winner. The use of an SD card is more affordable than the use of an eMMC module up to 1GB. From this point, both technology prices start to converge.

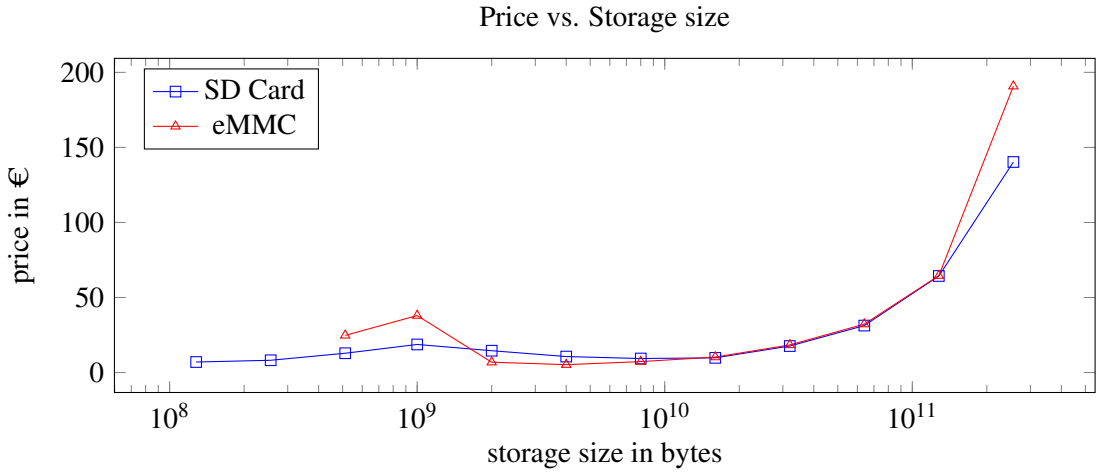


Figure 3.1: Storage Technologies Price Comparison

The fact that SD cards can't be soldered into a PCB, makes this technology vulnerable to mechanical effects inherent to the sport's practice. However, eMMC requires the implementation of a specific interface to communicate with it, in opposition to SD Card that uses SPI. Another

[2]Mouser Electronics, Inc [website], https://pt.mouser.com/, (accessed 23 February 2020)

point to be considered is that SD Cards are more affordable than eMMC to capacities up to 1GB, what to the author's opinion is more than sufficient for the project. For these two last reasons, the author has decided to use an SD Card as the SB4S's storage method.

3.4 Charging Method

The selected charging method for SB4S's specification requires to provide waterproofness and enough power to the system. The use of a solution with exposed metal, like USB charging, increases the probability of having a short-circuit. In addition to this, the exposure of metal to seawater results in corrosion. The use of a Qi charger to remove the existence of an interface between the seawater and the electrical system could be an interesting choice if this technology. But the use of Qi charger can be expensive and requires the use of a proper charger, perfectly aligned with charging interface on the device, beyond requiring the user to charging routines. At last, we have the use of PV panels as a charging method. This method guarantees the device's waterproofness and doesn't require any additional concern to the user. The major downside of its use would be the amount of produced power versus its collecting area during the period with low solar irradiance, which will not be a problem for our solution, as a surfboard's area is more than the required (see the study on section 4.3).

Chapter 4

Development

In this chapter, we will discuss the process of defining SB4S's specification itself. In section 4.1, we'll start with a description of the overall system, hardware and firmware. Then, in section 4.2, we proceed with a survey on some modules available on the market to select the most suitable one. In section 4.3, we estimate the overall system's power consumption, from the selected modules, and, with it, its charging method and battery. On section 4.4, we'll describe the development of the firmware using the Kallisto SDK.

4.1 Overall System Overview

SB4S is a surf monitoring solution embedded on a surfboard that uses GPS, Gyroscope, Accelerometer and Magnetometer to collect relevant data on the activity. This solution, based on FhP's Kallisto Platform, uses BLE to interface with the user and WiFi to transfer the acquired data to an external monitor. This external monitor will then feed FhP's surf profiling algorithm with the acquired data and show all the statistics to the user.

These modules are interconnected by a microcontroller, responsible for controlling their behaviour. SB4S is powered up by a photovoltaic charging system, that ensures the waterproofness

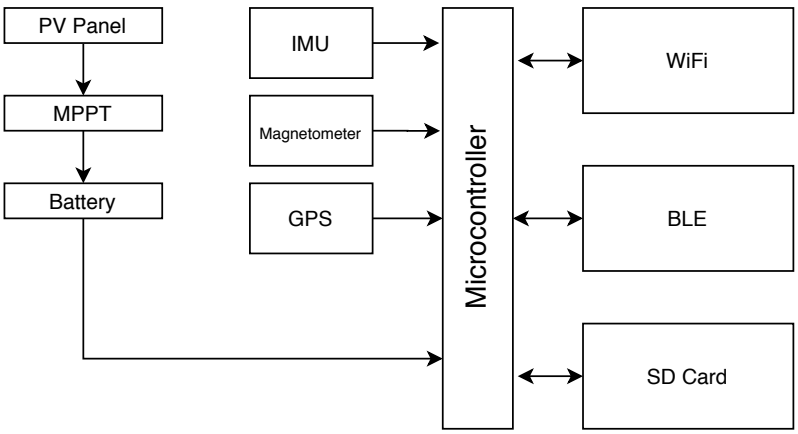


Figure 4.1: Block Diagram

of the system. The system is composed of a set of PV cells, an MPPT and a battery (Figure 4.1). The first is responsible for generating electrical current from solar radiation. The MPPT is responsible for imposing an optimal load into the cells and maximise the generated power and the batteries for storing the generated power and provide it to the system.

The standard use case of SB4S, see Figure 4.2, the user starts by asking an external monitor (e.g. smartphone) to connect with SB4S's BLE module. Then, the BLE module notifies the microcontroller of the event, enabling the system to receive commands from the external monitor: start a session, stop a session or sync data. These commands are responsible for controlling SB4S's internal state machine, composed of three states: IDLE, SURF and TX (Figure 4.3).

After connecting the SB4S with a smartphone (external monitor), the user sends a request to start a new surf session. When this message is received, by the BLE module, the microcontroller turns on all the sensors and requests them to start measuring data at the required data rates. Every time a new measurement is acquired, the sensors notify the microcontroller of the event and provide the acquired measurement, which is then stored into the SD Card. This procedure will be repeated until the end of the surf session.

At the end of the session, the user requests to the SB4S to stop acquiring new data by sending a 'stop session' command through BLE. When this command is received, the microcontroller requests to all the sensor to stop acquiring data and then requests them to turn off.

The user then sends a request for syncing the data between SB4S and the external monitor. The external monitor sends a 'sync data' command to SB4S's BLE module, which notifies the microcontroller that turns on the WiFi module and requests to it, the creating of an access point (AP). Then, the WiFi module requests to the microcontroller to notify the external monitor, via BLE, of the AP creation together with its SSID and password. Upon this notification, the external monitor connects itself to the AP and opens a TCP (Transmission Control Protocol) socket at a specific port. When the connection between the two (external monitor and SB4S's WiFi module) is established, the WiFi module notifies the microcontroller that reads the data from the SD card and requests the WiFi module to start a TPC connection and send this data through it. When all data is sent, the WiFi module notifies the microcontroller, that requests to the module the TCP connection and AP closure, deletes the sent data from the SD Card and turns of the module.

From the system's point-of-view, the SB4S's microcontroller has an internal state machine, Figure 4.3, that controls all the modules in the system. This state machine is composed by three different states: IDLE, SURF and TX.

On IDLE state, all the modules besides the microcontroller, BLE and PV charging system are disabled. The IDLE state is the default state of the system. And SB4S will stay in this state until a 'start session' or 'sync data' message is received (Figure 4.3).

On SURF state, the MCU starts all the sensors (GPS, IMU and Magnetometer) of the system. For every new measurement, a handler gets called, and its values get stored into the SD Card. The MCU's internal state machine only returns to the IDLE state, when a "stop session" message is received (Figure 4.3).

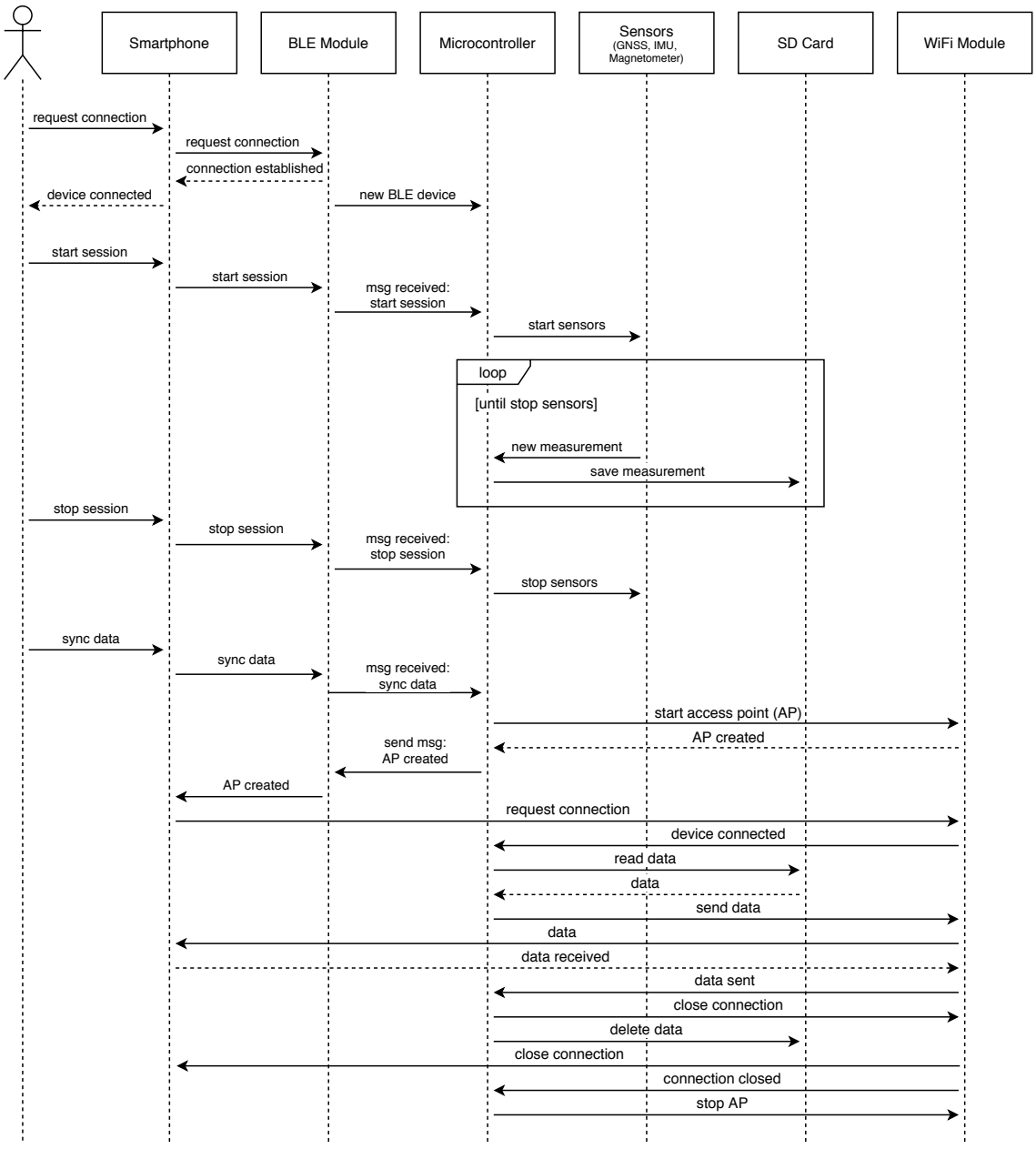


Figure 4.2: Sequence Diagram: Standard Use Case

On TX state, only the WiFi module, BLE module, the microcontroller and the PV charging system are enabled. On this state, SB4S creates an access point (AP) and waits for an external monitor to connects with it. When this connection is established, the SB4S looks for a TCP connection at a specific port and sends through it all the acquired data. When all the data is sent, SB4S returns itself to IDLE mode (Figure 4.3).

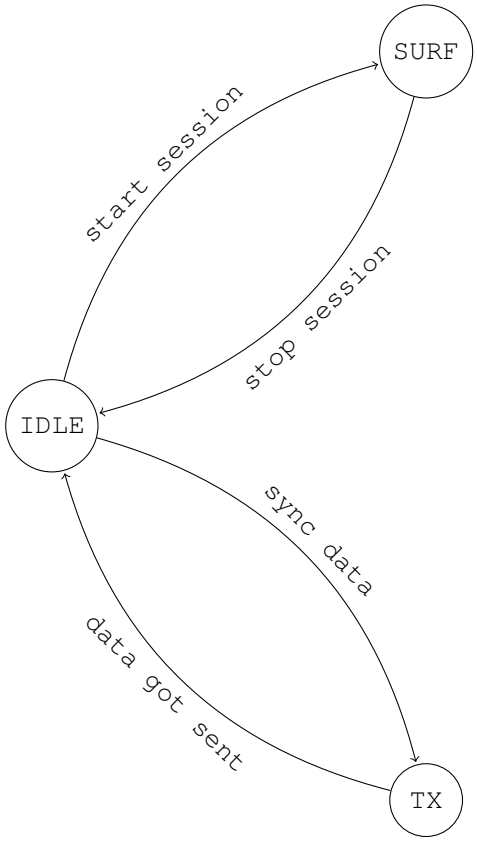


Figure 4.3: MCU Internal State Machine

4.2 Modules' Selection

A set of surveys, available on Appendix C, were conducted on different modules (GPS, WiFi, NB-IoT, PV Panels and Batteries). These surveys served as the basis for the selection of the modules used in the final system. Due to page size constraints, only the most relevant characteristics of each type of module were considered in the document. All the information on it was obtained from the datasheets of each module.

The selection of modules is constraint by the number of GPIO and interface ports on the Kallisto Board: UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit) and GPIO (General Purpose Input/Output) ports. The UART protocol is the more restrictive of the three, allowing only one device for each port (point-to-point link). In addition to this, it is the slowest of the three interfaces. For this reason, the use of other interfaces, when available, is recommended.

In opposition to UART, SPI allows multiple slaves to connect to a single master. However, for each new slave an additional, chip-select, wire is required limiting the number of slaves to the number of available GPIO ports on the board. SPI is the fastest of the three communication protocols available, and its use is recommended for high-speed applications (e.g. SD Card).

I2C supports up to 128 devices (master and slaves) to be connected between each other by the same bus line (two wires: SDA and SCL). This interface is recommended for all the low-speed modules, as it doesn't require additional ports beside the two bus wires.

Another important module's characteristic to take in consideration is its power consumption at standby.

From the three most power-efficient GNSS modules on the survey: *STMicroelectronics* TESEO LIV3F, *Maestro Wireless* A2200 and *Linx Technologies* GPS-F4. Only the first two provided other interfaces than UART. Comparing both solutions, TESEO LIV3F was the most power-efficient of the two, was compatible with other GNSS satellite systems than GPS and provided more interfaces (SPI and TWI) than the A2200. On the other hand, A2200 can provide a wider range of update rates than TESEO LIV3F, which cannot be considered an advantage for the use in SB4S.

To select a WiFi module, the author used the powers consumed during transmission as a reference. The two most power-efficient modules found were developed by the same Manufacturer (*Espressif Systems*), the ESP8266EX and the ESP32. As both solutions had the same data-rates, the selection criteria was the price of the module. The chosen module was ESP8266EX.

Table 4.1: Selected GNSS and WiFi modules

Module	Model	Communication Interface
WiFi	Espressif ESP8266EX	UART
GNSS	STMicroelectronics TESEO LIV3F	I2C

4.3 Power System Development

4.3.1 Consumption Estimation

Now that we have selected all the components that will be part of SB4S, we are able to estimate the consumption of the system. This consumption will not be constant during all the system's operation, as different components will be in need at each mode. The author has identified three different operation modes: IDLE, SURF and TX (Table 4.2).

Table 4.2: SB4S's operation modes

	BLE	AHRS	GPS	WiFI
IDLE	X			
CONTROL	X			
SURF	X	X	X	
TX	X			X

IDLE mode uses Kallisto alone, without enabling the internal AHRS. In this state, Kallisto only manages the BLE connections and controls it's internal state machine.

After identifying the different operation modes in SB4S, it is now possible to estimate the power consumption of SB4S. All values used in this estimation, to the exception of Kallisto, can be found on each modules datasheet ([54] and [55]). Information about Kallisto can be found, on its power profile report [56].

Table 4.3: SB4S's instant power estimation for each operation mode

Module	Model	Voltage (V)	Current (mA)			Power (mW)		
			IDLE	SURF	TX	IDLE	SURF	TX
Kallisto	Light	4	1,120	2,060	1,120	4,480	8,240	4,480
GPS	TESEO-LIV3F	3.3	0,006	28,000	0,006	0,020	92,400	0,0198
WiFi	ESP-WROOM-02	3.3	0,002	0,002	140,000	0,007	0,007	462,000
Total			1,128	30,062	141,126	4,506	100,647	466,500

4.3.2 Photo-voltaic Panel Dimensioning

Surfboards are usually stored indoors, to prevent their exposure to the elements. As a consequence, SB4S's battery will probably be empty at the beginning of the majority of the sessions. For this reason, a PV panel must be capable of providing enough instant power to maintain SB4S basic operations (SURF and IDLE modes) and charge its battery sufficiently to transfer the acquired data during TX mode. In mathematical terms, the instant power provided by the PV panel must always be greater than the maximum instant power consumption of the device, excluding TX mode. Using the estimations in Table 5.3.3, a PV panel must be capable of supplying at least 0,1W of instant power to the system. Considering that available solar panel solutions on the market are capable of delivering watt units of instant power, in standard test conditions (STC), we will assume that a solar panel can supply enough energy to charge and operate SB4S.

Considering that available solar panel solutions on the market are capable of delivering watt units of instant power, in standard test conditions (STC), we can assume that a solar panel will supply enough energy to charge and operate SB4S. Despite this, the instant power @STC cannot be used as a reference, as it assumes an irradiance of $1000W/m^2$, air mass of 1.5 and a temperature of 25ºC [57].

On this section, we will estimate the amount of advertised power (@STC) required to guarantee the use of SB4S on the majority (98%) of the cases and select a suitable PV panel for our system. This value will be estimated using the irradiance in Portuguese beaches, throughout the year.

4.3.2.1 The Portuguese case

PV cells' instant output power always depends on the quantity of incident solar irradiance. This last varies with PV cells' geographical position, date and time. With equation 4.1, it's possible to estimate the amount of solar irradiance for a determined location (Latitude and Longitude),

number of days since the beginning of the year (d) and the time at Greenwich's meridian, or Local Standard Time Meridian (LSTM), by a sequence of calculations (Eq. 4.1 to Eq. 4.10) [58].

$$I_D = 1,353 \times 0,7^{AM^{0,678}} \quad (W/m^2) \tag{4.1}$$

$$AM = cos(Zenith) \tag{4.2}$$

$$Zenith = 90 - Elevation \tag{4.3}$$

$$\text{Elevation } = \arcsin(\sin(\text{declination}) \cdot \sin(\text{latitude}) + \cos(declination) \cdot \cos(\text{latitude}) \cos(\text{HRA})) \tag{4.4}$$

$$Declination = -23,45 \cdot \times \cos(\frac{360}{365} \cdot (d - 81)) \tag{4.5}$$

$$HRA = 15 \cdot (LST - 12) \tag{4.6}$$

$$LST = LT + \frac{TC}{60} \tag{4.7}$$

$$TC = 4 \cdot (Longitude - LSTM) + EoT \tag{4.8}$$

$$EoT = 0,87 \cdot \sin(2B) - 7,53 \cdot \cos(B) - 1,5 \cdot \sin(B) \tag{4.9}$$

$$B = \frac{360}{365} \cdot (d - 81) \tag{4.10}$$

The main flaw of this approach is that it doesn't take in consideration the weather forecast, as this one cannot be obtained, by absolute values such as Longitude, Latitude, Day and Time. For this reason, the author has chosen to use a more practical approach by using a data set provided by the European Photovoltaic Geographical Information System (PVGIS)[59].

From a small selection of articles [60][61], on the web, the author selected 18 reference Portuguese beaches in surf practice for the analysis of their hourly's solar irradiance through the year from the PVGIS data set. The samples in this dataset were continuously measured between 2005 and 2014. In this analysis, a Python's script, found in Appendix A, was used to calculated the frequency of the amount of solar's irradiance , Figure 4.4 (A), and the single-event and cumulative probabilities of their occurrence, Figure 4.4 (B).

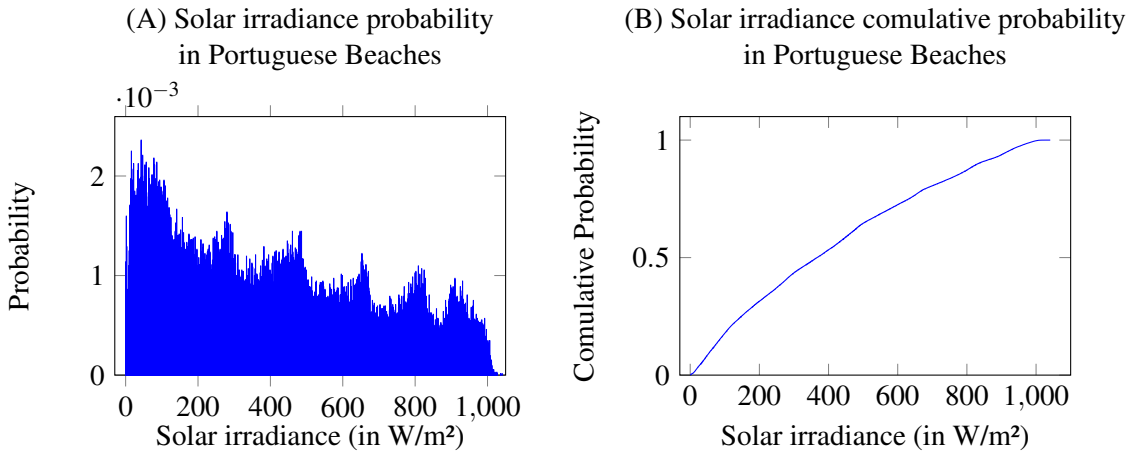


Figure 4.4: Results on solar irradiance study in Portuguese Beaches

4.3.2.2 PV's colletion system area

Surfboards come in many shapes and sizes, difficulting the estimation of an area for SB4S's PV collection system. A survey was conducted on surfboard's sizes, using a small sample from two major sufboard brands (Quiksilver[62] and Lost[63]). From this survey, see Appendix C.3, the smallest surfboard found, the Lost Cobra Killer) measured 18" (46 cm) at its widest point and 5'2" (157 cm) of height.

However, the surfboard's physical dimensions aren't the only constraint to the PV system's usable area. During arm paddling, that represents a significant part of total surf time [50], surfers lay down on the surfboard, covering most of its available area. The remaining section, usually on surfboard's nose, will depend on the surfer's height. Taking this into consideration, a maximum of 30cm was arbitrarily defined to SB4S's PV collection system's height.

4.3.2.3 PV cells survey

Solar cell manufacturers specify the amount of instant power provided by the cell in standard test conditions (irradiance of $1000W/m^2$, Air Mass of 1.5 and 25ºC of Temperature), P_{STC} [57]. However, it is not recommended to directly compare this value with the required amount of power, as these conditions are rare in real-life, as shown by Figure 4.4.

The real amount of instant power produced by a PV panel can be obtained by multiplying its efficiency, its area and the solar irradiance at a given time. The efficiency of the PV panel, if not provided by the manufacturer, can be obtained from dividing the output power with the input power, both at STC. See equation 4.11.

$$P_{PV_{out}} = \eta \cdot E_e \cdot A = \frac{P_{out}}{P_{in}} \cdot E_e \cdot A = \frac{P_{STC}}{E_{e_{STC}} \cdot A_{STC}} \cdot E_e \cdot A = \frac{P_{STC}}{1000} \cdot E_e \cdot A \quad (W) \tag{4.11}$$

To select an appropriate PV cell for SB4S, this must provide a real output power of at least the maximum instantaneous power consumed by the system during IDLE and SURF states plus a safety factor, defined by the author, of 50%. The exceeded power will be used to charge a battery capable of providing enough power for the TX state.

$$P_{PV_{out}} > P_{consumed} + 0,5 \cdot P_{consumed} \Leftrightarrow P_{PV_{out}} > 0,151 \quad (W) \tag{4.12}$$

For each entry of the Solar Cell's survey, Appendix C.4, the number of required panels to supply the minimum amount of instant power calculated on equation 4.11 was estimated using equation 4.13. If the area occupied by this solution exceeds the available one, $120cm^2$, the entry would be opted out (Figure 4.5). This area assumes a margin of 2.55 cm around the panels.

$$n = \lceil \frac{0,151}{\eta \cdot E_e \cdot A_{Cell}} \rceil \tag{4.13}$$

To select a solar cell from the survey, Annex C.4, the author used the methodology in Figure 4.5 for different values of solar irradiance, with 99%, 95% and 90% probability of occurrence (see Table 4.4). The results show that 1 out of 13 modules (the *SeeedStudio* 313070003) would work during sunhours with 99% confidence, 10 out of 13 with 95% and 12 with 90%.

Table 4.4: Incident solar irradiance associated with popular confidence intervals

Confidence Interval (%)	Incident solar irradiance (W/m^2)
99	10
95	32
90	56

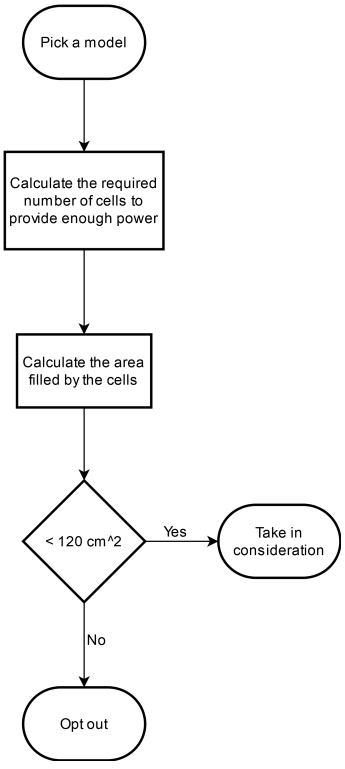


Figure 4.5: Methodology used to select a PV cell

Due to stock shortage on the *SeeedStudio* 313070003, the selected module had to be changed. A new arbitrary confidence interval was set (98%), resulting in more three modules to take in consideration: the *SeeedStudio* 313070002, the *Xunzel* MS6V900 and the *SeeedStudio* 313070001. From this new sample, only two were available to purchase: the *Xunzel* MS6V900 and the *SeeedStudio* 313070001. The selection criteria to choose between these two modules was their size, as smaller cells can be used to follow the surfboard curvature, resulting in the choice of *SeeedStudio* 313070001 ($\eta = 13.59\%$).

As already mentioned on section 2.3.4, MPPTs can be used to maximise the output power of a PV solution. The selection of an MPPT module must take in consideration two factors of the PV panels specification: its short-circuit current and its open-circuit voltage. These two values must

be lower than the accepted input values of current and voltage of the MPPT module respectively. This makes difficult the task of finding a suitable solution for an PV system, specially when the size and cost of solutions tend to increase with the accepted input values. The selected module for SB4S was the Linear Technology LT3652.

4.3.3 Battery Dimensioning

The required amount of battery charge to support SB4S's standard use case can be calculated by multiplying the instant amount of pulled current for each mode and their expected duration (eq. 4.14). The required amount of battery charge to support SB4S's standard use case can be calculated by multiplying the instant amount of pulled current for each mode and their expected duration (eq. 4.14). A margin of 18 hours for IDLE mode was defined to provide the user enough time between the surf's session and the synchronization of data.

$$\text{Battery Capacity } = I_{surf} \cdot t_{surf} + I_{surf} \cdot \frac{t_{surf} \cdot R_{output}}{R_{TX}} + I_{IDLE} \cdot t_{IDLE} \text{ (mAh)} \tag{4.14}$$

Table 4.5: Battery's charge capacity estimation

	hours	minutes	mAh
IDLE (guess)	18	0	20,3
SURF	5	0	150,3
TX (worst-case condition)	0	4	9,4
Total	23	5	180

A better approach would be to try to maximize the amount of collected power in a battery. Using the same data sample from the analysis on solar irradiance at Portuguese beaches, section 4.3.2.1, the amount of required instant power and the equation 4.11 it's possible to generate a graph relating the probability of not fully charge a battery with its capacity (Figure 4.6). For a better understanding of the methodology in use, the developed Python's script can be found in Appendix B.

Having a battery unable from fully charging would be a waste of battery's capacity. In the other hand, having a battery with a high probability of being fully charged could be considered a waste of the PV panel's output power. Despite this, the PV panel's output power is more unstable than the battery's capacity, making this last option a better approach for solving our problem. In addition to this, a battery with, for example, a capacity around 4Wh (80% of full charge) would be able to support six complete standard use cases.

A survey, see Appendix C.5, was conducted to select the battery's selection. In addition to their capacities, other characteristics such as weight and price were taken into consideration. This resulted in the choice of the *CELLAVIA* L573450 with a capacity of 3,6Wh.

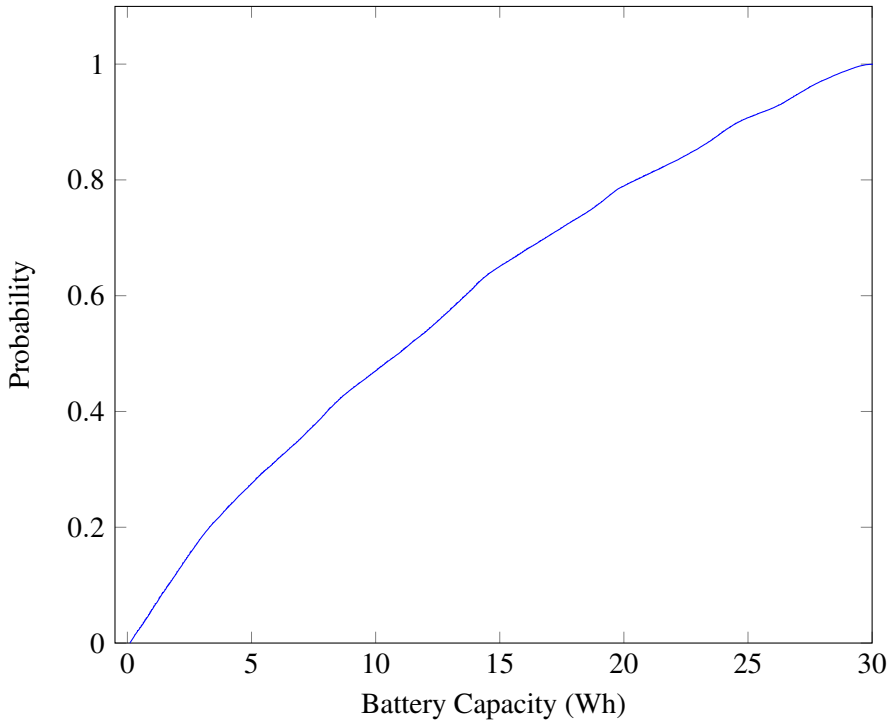


Figure 4.6: Probability of not full charging a battery vs battery capacity

4.4 Firmware Development

In this section we will address the development of SB4S's firmware using Kallisto Platform. To facilitate the development of new IoT solutions, Kallisto Platform provides a SDK with a respectable amount of HALs and Drivers to communicate with peripherals. Due to the lack of GNSS nor WiFi support by this platform, the author developed drivers for the modules selected on section 4.2 and their respective HALs.

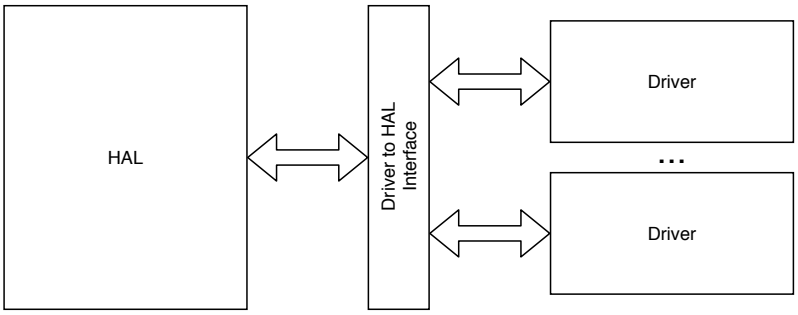


Figure 4.7: HAL to Driver Interface

HAL, or Hardware Abstraction Layer, is a piece of software that enables the use of the same main firmware code for different electronic modules (Figure 4.7). It usually offers access to the commons functionalities of a Hardware category, e.g. check the temperature in a thermometer.

A Driver-to-HAL Interface is responsible for mapping the functions on the HAL with their respective functions on the different drivers (Figure 4.7). The drivers are responsible to interface the software and the hardware (Figure 4.7). This piece of software operates/controls the connected peripheral, acting as a translator between it and the applications.

The interface between these three software pieces is done by using a handler, not represented on the picture.

For the rest of this section, we will address the development of the WiFi and GPS drivers and HALs, as well as the development of the SB4S's main firmware.

4.4.1 TESEO LIV3F driver and GNSS HAL

4.4.1.1 Driver

The interface between TESEO Liv3F and the Kallisto board can be either I2C or UART. As mentioned on 4.2, the use of I2C is prefered over UART. In order to provide compatibility with the NEMA 0183 standard, the I2C interface emulates a UART interface on the register 0xFF.

In opposition to UART, that is a full-duplex protocol, I2C's implementation of the NMEA standard requires a push and pull strategy in order to avoid collisions. To do so, a timer is set during the initialization of the driver. This timer is responsible for pulling the NMEA sentences out of the GNSS module. These sentences follow the NEMA standard and don't require post-processing over them. After a new NMEA sentence is received, the timer is reset, and the NMEA sentence is sent to the GNSS HAL.

The functionalities of the driver are exposed through 3 functions (see Figure 4.8): an init functions, responsible for initializing the timer and setting up the function to be called after a new sentence is receives; a start and a stop functions to control the timer.

```
/* Initialises the timer and sets the handler to be called
after receiving a NMEA sentence */
void skp_drv_liv3f_init(
    nrf_drv_twi_t *twi_master,
    skp_drv_liv3f_evt_handler_t evt_handler
    );

/* Starts the timer */
void skp_drv_liv3f_start();

/* Stops the timer */
void skp_drv_liv3f_stop();
```

Figure 4.8: LIV3F driver functions

4.4.1.2 HAL

The GNSS HAL extends the functionality of the different GNSS drivers. In other words, it is responsible for calling the functions of the driver (init, start and stop) 4.9. In addition to this, the HAL's init function is responsible for setting its handler as the handler to be called from the driver, as well as setting a function to be called on the main's app.

When the HAL's handler function is called by the driver, the HAL checks the integrity of the message and parses its content. The content will then be sent to the main's app through its handler. The content of the *NMEA* sentence is exposed to the main function as a data structure, *skp_hal_gnss_evt_t*, enabling the developer to use the HAL without knowledge on the *NMEA* standard 4.9.

```
typedef struct skp_hal_gnss_data_rmc_s{
    skp_hal_gps_timestamp Timestamp;
    bool Status;
    skp_hal_gps_coordinates Latitude;
    skp_hal_gps_coordinates Longitude;
    float Speed;
    float Trackgood;
    float MagVar;
    char MagVarDir;
    skp_hal_gps_positioning_system_mode Mode;
} skp_hal_gnss_data_rmc_t;

typedef union{
    skp_hal_gnss_data_rmc_t rmc;
} skp_hal_gnss_evt_data_t;

typedef struct{
    skp_hal_gnss_evt_type_t type;
    skp_hal_gnss_nmea_talker_t talker;
    skp_hal_gnss_evt_data_t data;
} skp_hal_gnss_evt_t;

/* Exposes the driver's init function and sets the handler
to be called when new data is created */
void skp_hal_gps_init(
    skp_hal_gps_drv_t *gps_driver,
    skp_hal_gps_evt_handler_t evt_handler
);
```

```
/* Exposes the driver's start function */
void skp_hal_gps_start(skp_hal_gps_drv_t *gps_driver);

/* Exposes the driver's stop function */
void skp_hal_gps_stop(skp_hal_gps_drv_t *gps_driver);

/* Handler called by the driver. It is responsible for parsing
the NMEA sentence and call the handler on the main app. */
void skp_hal_gps_drv_evt_handler(char *str);
```

Figure 4.9: GNSS HAL functions

4.4.2 ESP8266EX driver and WiFi HAL

4.4.2.1 Driver

The interface between *ESP8266EX* and the Kallisto board can only be done through UART (with or without flow control). The later requires two additional signals Request to Clear (RTC) and Clear To Send (CTS), over the traditional UART interface.

During the initialisation of the driver, the Nordic's SDK UART interface is set up with two FIFO buffers, each with 2048 positions, and with the handler, to be called on the arrival of new strings to the RX buffer.

To configure or send data to the *ESP8266EX*, the master (Kallisto Board) requires to send the requests encoded as AT commands. These commands have the following structure:

AT+COMMAND=PARAMETERS

A list with the accepted commands and its valid parameters is available on *Espressif's GitHub* page [64].

For each message sent to *ESP8266EX*, the master receives a response with a copy of the command and its status: OK, if successfully executed, or ERROR, if not.

When sending data to *ESP8266EX*, the data to be sent must only be transmitted after the request's response arrives the master. Then the master is notified of the reception by a "DATA OK" message.

However, not all messages received by the master are responses to some command. Another type of messages, called notifications, can be received by the master. This messages start with a plus sign and are normally received when an event occurs on the *ESP8266EX* (e.g. new device get connected to the module).

Every time a new message is received by the Kallisto Board, through the UART interface, the message is parsed. This process is controlled by an internal finite state machine (See Figure 4.10). Messages are received, during the IDLE state. Where its type: command, notification or DATA OK is identified.

When a command is received, the state machine changes its state to AT CMD, where it parses its parameters and adds them to an event data structure, *skp_drv_esp8266_evt_t*. Then, on WAIT state, the state machine waits for the status message, and depending on its value, selects a new state. In both states, a flag with the status of the command is added to the structure, the HAL's handler is called and the state machine returns to IDLE mode.

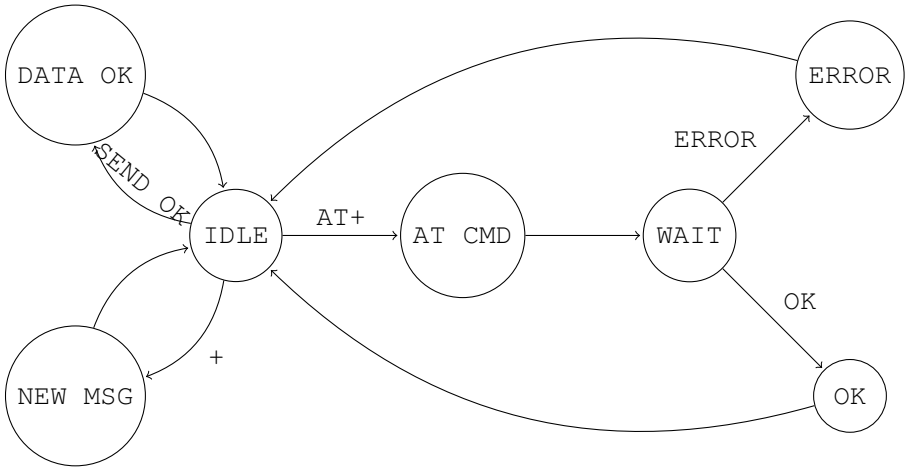


Figure 4.10: Finite State Machine: Driver's RX parser

When a notification or DATA OK message is received, an event is generated and sent to the HAL at their respective state.

The functionalities of the driver are exposed through 5 functions (see Figure 4.11): an init function, responsible for setting the handler to be called upon an event occurs; a function that starts a new access point given its SSID and password; a function that initiates a new connection, either TCP or UDP, a function that closes the connection and a function that is used to send data through a determined connection.

```
/* Type of event */
typedef enum {
    ESP8266_NULL,
    ESP8266_CWMODE,
    ESP8266_CWSAP,
    ESP8266_CIPSTART,
    ESP8266_CIPCLOSE,
    ESP8266_CIPSEND,
    ESP8266_STA_CONNECTED,
    ESP8266_STA_DISCONNECTED,
    ESP8266_DIST_STA_IP
} skp_drv_esp8266_evt_type_t;
```

```
/* Status of the event */
typedef enum {
    OK,
    ERROR
} skp_drv_esp8266_evt_status_t;

/* Parameters of the event */
typedef union{
    bool error;
    skp_drv_esp8266_data_cwmode_t cwmode_params;
    skp_drv_esp8266_data_cwsap_t cwsap_params;
    skp_drv_esp8266_data_cipstart_t cipstart_params;
    skp_drv_esp8266_data_cipclose_t cipclose_params;
    skp_drv_esp8266_data_sta_connected_t sta_connected_params;
    skp_drv_esp8266_data_dist_sta_ip_t dist_sta_ip_params;
} skp_drv_esp8266_evt_params_t;


/* Event structure */
typedef struct skp_drv_esp8266_evt_s {
        skp_drv_esp8266_evt_type_t type;
    skp_drv_esp8266_evt_status_t status;
    skp_drv_esp8266_evt_params_t params;
} skp_drv_esp8266_evt_t;


/* Initialises the driver */
uint32_t skp_drv_esp8266_wifi_init(
    skp_hal_wifi_drv_evt_handler_t p_evt_handler
    );


/* Starts an access point */
void skp_drv_esp8266_wifi_startAP(
    char *ssid,
    char *password,
    bool hide_ssid
    );
```

```
/* Opens socket */
void skp_drv_esp8266_wifi_startConnection(
    uint8_t id,
    char * type,
    char * ip,
    uint16_t port
    );

/* Closes socket */
void skp_drv_esp8266_wifi_stopConnection(uint8_t id);

void skp_drv_esp8266_wifi_sendData(uint8_t id, char * data);
```

Figure 4.11: ESP8266EX driver

4.4.2.2 HAL

Similar to the GNSS HAL, the WiFi HAL extends the functionalities from its drivers: start access point, start and close connections, and send data. In addition to this, the WiFi HAL is responsible for managing the connections and connected devices. Which are identified on the driver by their id. This abstraction layer enables the developer to mention devices by their IP and MAC addresses, and connections by their IP and port. Both devices and connections are stored into two linked lists of types: *skp_hal_wifi_devices_list_t* and *skp_hal_wifi_connection_t*. Devices/connections are added to or removed from the lists upon notification of the driver.

```
/* Type of event */
typedef enum{
    WIFI_UNDEFINED,
    WIFI_AP_CREATED_OK,
    WIFI_AP_CREATED_ERROR,
    WIFI_CONN_OPENED_OK,
    WIFI_CONN_OPENED_ERROR,
    WIFI_CONN_CLOSED_OK,
    WIFI_CONN_CLOSED_ERROR,
    WIFI_POWER_ON,
    WIFI_POWER_OFF,
    WIFI_NEW_DEVICE,
    WIFI_NEW_DEVICE_IP,
    WIFI_DEVICE_OUT,
    WIFI_DATA_SENT
} skp_hal_wifi_evt_type_t;
```

```
/* Data in the event */
typedef union{
    uint8_t connectionID;
    skp_hal_wifi_data_connection_t connection;
    skp_hal_wifi_data_dev_t device;
} skp_hal_wifi_evt_data_t;

/* Event's structure */
typedef struct{
    skp_hal_wifi_evt_type_t type;
    skp_hal_wifi_evt_data_t data;
} skp_hal_wifi_evt_t;


/* Connection's structure */
typedef struct skp_hal_wifi_connection_s{
    uint8_t id;
    char ip[16];
    uint16_t port;
    int8_t next;
}skp_hal_wifi_connection_t;


/* Device's structure */
typedef struct skp_hal_wifi_device_s{
    char mac[18];
    char ip[16];
    int8_t next;
}skp_hal_wifi_device_t;


/* Initialises the WiFi HAL */
uint32_t skp_hal_wifi_init(
    skp_hal_wifi_drv_t * wifi_driver,
    skp_hal_wifi_devices_list_t * devs,
    skp_hal_wifi_connections_list_t * connections,
    skp_hal_wifi_evt_handler_t evt_handler
    );
```

```
/* Starts an access point */
uint32_t skp_hal_wifi_startAP(skp_hal_wifi_drv_t * wifi_driver,
    char * ssid,
    char * pwd,
    bool hide_ssid
    );


/* Opens a TCP connection */
uint32_t skp_hal_wifi_openTCP(skp_hal_wifi_drv_t * wifi_driver,
    char * ip,
    uint16_t port
    );


/* Sends data through a TCP connection */
uint32_t skp_hal_wifi_sendTCP(skp_hal_wifi_drv_t * wifi_driver,
    char * ip,
    uint16_t port,
    char * data
    );


/* Closes the TCP connection */
uint32_t skp_hal_wifi_closeConnection(
    skp_hal_wifi_drv_t * wifi_driver,
    char * ip,
    uint16_t port
    );
```

Figure 4.12: WiFi HAL

4.4.3 SB4S Firmware

The SB4S firmware is divided in two pieces of software: the Nordic's SoftDevice, responsible for handling the BLE stack and radio events, and the SB4S's state machine, responsible for handling all the system's logic (turn on the sensors, store data, send data, etc.).

Nordic's SoftDevice requires to operate with the minimum latency possible, for this reason the use of Nordic's scheduler is almost obligatory. This allow us to preempt time consuming tasks, such as processing the sensor's data, in favour of highly important tasks such as system interrupts.

The state machine, described in Figure 4.3, is responsible for enabling/disabling all the sensors, storing the data acquired from the sensors and sending it through Wi-Fi to the external device accordingly with the present state (IDLE, SURF or TX).

When a BLE message is received, its interrupt handler is responsible for switching the state of the SB4S's state machine and require its execution. According to the current state, the state machine requires to the scheduler the execution of the respective tasks associated with the state.

During SURF state, when a new measurement is received from a sensor, its interrupt handler is responsible for putting the new value into a 32 position buffer, for each sensor. When full, the same interrupt handler requires to the scheduler the execution of the processing task. This relives the scheduler from having to execute 301 new tasks for each second, one per each new measurement. After being processed, the measurements are stored into the SD card in different files, one for each sensor (ACC.txt; GYRO.txt; MAG.txt; and GPS.txt).

As mention in section 4.1, when in TX state, the SB4's firmware creates an AP and waits for a device to connect to it. When a device connects to the AP, the interrupt handler responsible for this event schedules the process of read and transfer of the measurements' files. After which, all files are erased from the SD card.

When the state machine enters the IDLE state, all the sensors and the WiFi module are shutdown.

Chapter 5

System Validation

In this chapter, a series of system validation tests are described: In section 5.1, on the effect of epoxy resin; and in section 5.2, on the system's power consumption and transmission rate. In section 5.3, the results from the tests in 5.1 and 5.2 are presented and discussed. Also, in this section, the author evaluates the system as a global.

5.1 Effect of Epoxy Resin over PV Cells

SB4S is meant to be embedded into a surfboard, that usually is coated with epoxy resin to protect their polyurethane or polystyrene foam core from the elements. This results in the PV panels being covered with this type of resin.

To the author's knowledge, there are no studies available on the effects of the epoxy resin over PV cells. A small experiment was conducted to understand if the presence of epoxy resin over the PV cells would affect their behaviour.

In this experiment, a sample of 3 *Seeedstudio 313070001* PV panels, were coated with different quantities of epoxy resin: 24ml, 48ml and 72ml (Figure 5.1), and were characterized, using a Source Measuring Unit (SMU), the *Keithley SourceMeter 2460*. From the I-V curve, we can infer the real values for V_{mp} (Voltage at Maximum Power), I_{mp} (Current at Maximum Power), V_{oc} (Voltage at Open Circuit) and I_{sc} (Current at Short Circuit). With this information, we can compare the performances between all the test units.

5.2 Surf-session Simulation

On chapters 3 and 4, we have estimated the SB4S's transfer duration and overall device's current consumption. From these values, we selected technologies and modules for our specification. Now that a prototype based on these specifications exists, we can test and validate, along with the results from the tests on 5.1, if the dimensioning of the solar cells and battery needs to be optimized.

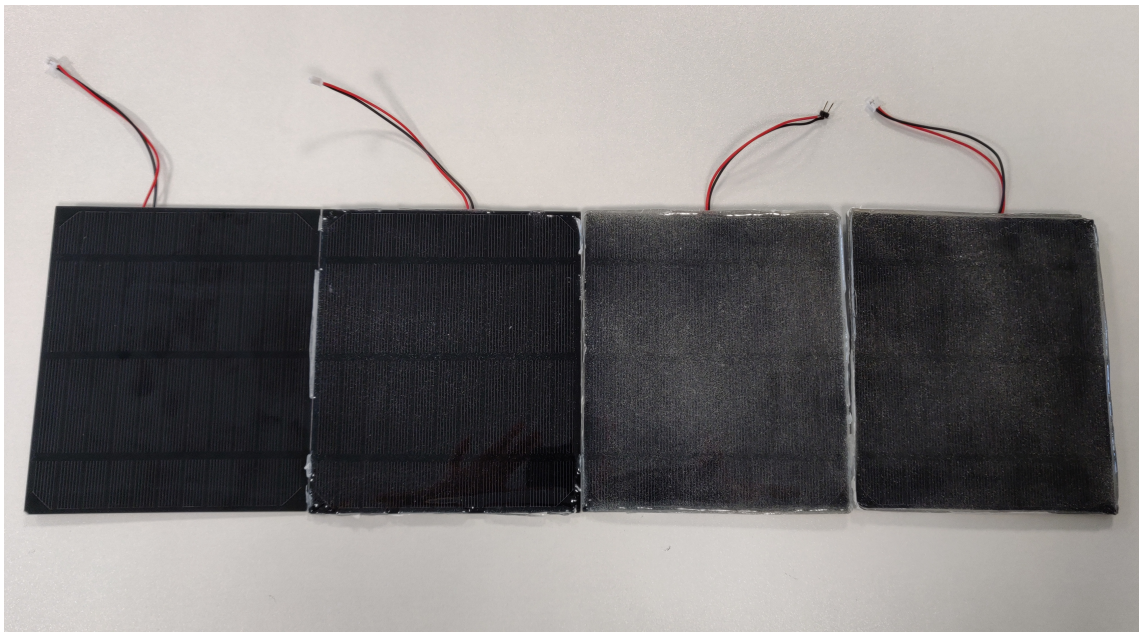


Figure 5.1: Sample of PV cells each coated with 0ml, 24ml, 48ml and 72ml of epoxy resin, respectively

The development of a smartphone app as the external monitor for SB4S is outside the scope of this work. For this reason, in the tests, a smartphone with Nordic's nRF Connect [1] was used to send the control messages to SB4S by BLE. And a computer with GNU/Linux was used to receive the data through Wi-Fi.

An SMU (*Keithley SourceMeter 2460*) was used to source the SB4S and measure its power consumption during all the experiment. To measure the amount of data and the duration of the transference, we used some GNU tools such as *diff*[2], *time*[3] and *netcat*[4] tools.

Unfortunately, due to COVID-19 pandemic, it wasn't possible to test the system using Kallisto boards (still in the manufacturing phase) and the MPPT (delivering delays). To substitute the use of a Kallisto board in the tests, the author used a set of development kits from the different components on the Kallisto: the Nordic Semiconductor's nRF52820 DK, the Bosch BMM150 Shuttle Board and the Bosch BMI160 Shuttle Board. To validate the system, the "Kallisto breadboard" was connected with the other components of the SB4S system (see Figure 5.2).

The GNSS antenna was set outside the building for data and consumption accuracy, as GNSS signals suffer attenuation from the building's structure, and therefore the receiver tends to spend more power to acquire it.

[1] smartphone app that allows for scanning BLE devices and communicating with them
[2] identifies the differences between a set of files
[3] measures the running time of a process
[4] network tool, that enables to easily create TCP/UDP sockets

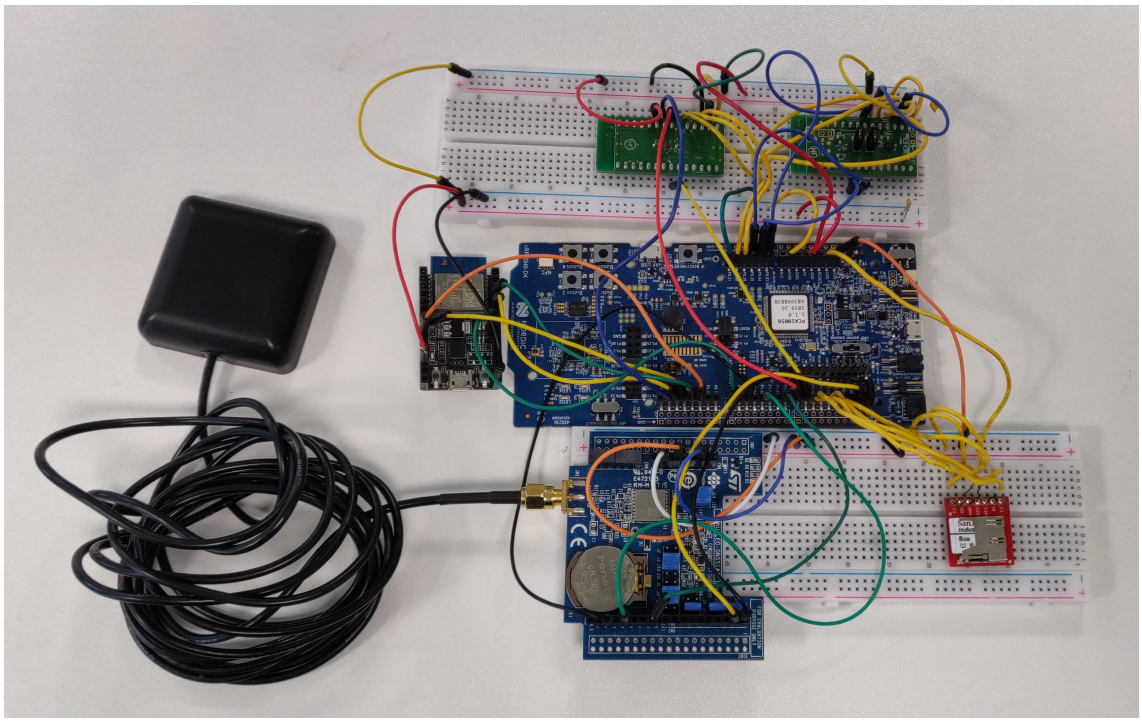


Figure 5.2: SB4S's Breadboard Prototype

5.2.1 Power Consumption

Using the *Keysight 34410A* bench multi-meter, the author measured the SB4S's current consumption for each one of the different states of the system: IDLE, TX and SURF.

The nRF52's devkit allows measuring its current consumption through a specific port on the board (Figure 5.3). To enable this mode, it was necessary to cut a small pad on the PCB (Figure 5.4). From this point on, the measuring port had to be short circuited, with the jumper or an ammeter, to enable the boards operation.

5.2.2 Transmission

To measure the transfer duration over Wi-Fi, the behaviour of the SB4S app was simulated using Nordic nRF Connect and a computer with GNU/Linux. After acquiring one minute of data, the SB4S was set into TX mode using nRF Connect. With the access point created, *netcat* was used to open a TCP socket on the computer and then connected it to the provided access point. The received data from this socket was pipelined into a file. At the end, the connection was closed. *time* tool was used to measure the run-time of the *netcat*, this value will be considered the duration of the all transfer. To verify the integrity of the data, the acquired measurements were extracted from the SD Card and using *diff* tool compared the files received trough Wi-Fi with the ones extracted.

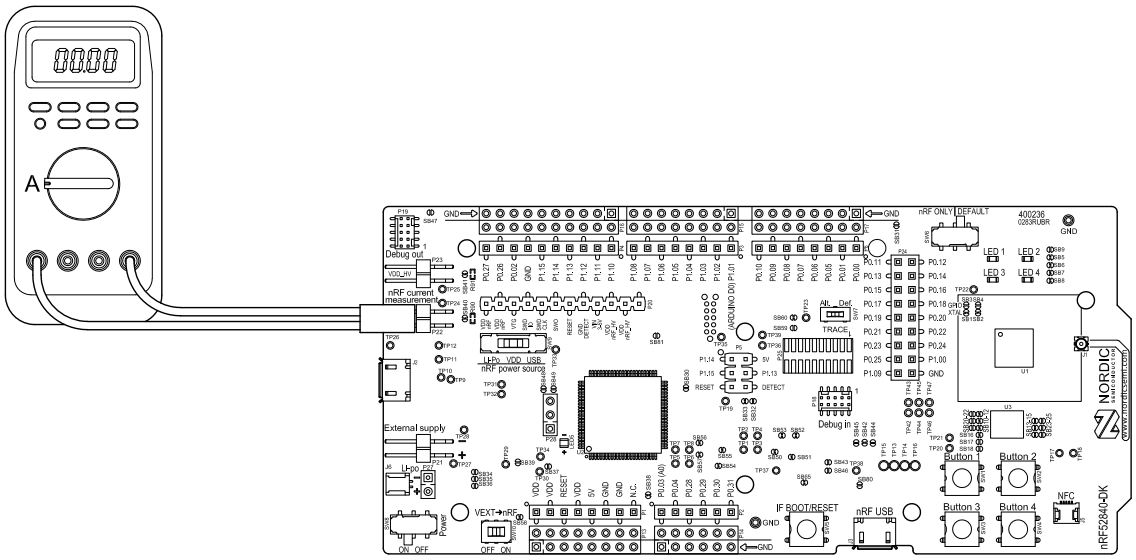


Figure 5.3: DK current measurement with an ampere-meter, from [65]

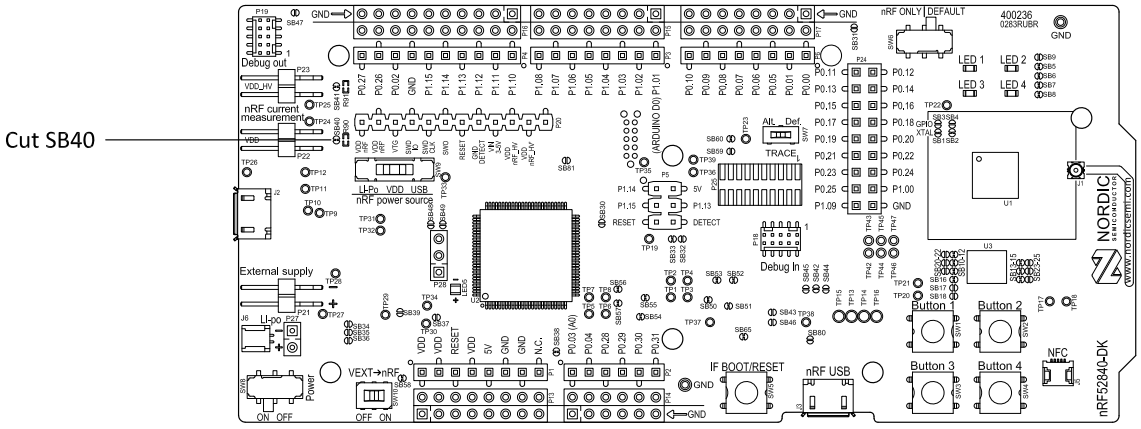


Figure 5.4: Preparing the DK for current measurements, from [66]

5.3 Discussion on results

5.3.1 Effect of epoxy resin over PV Cells

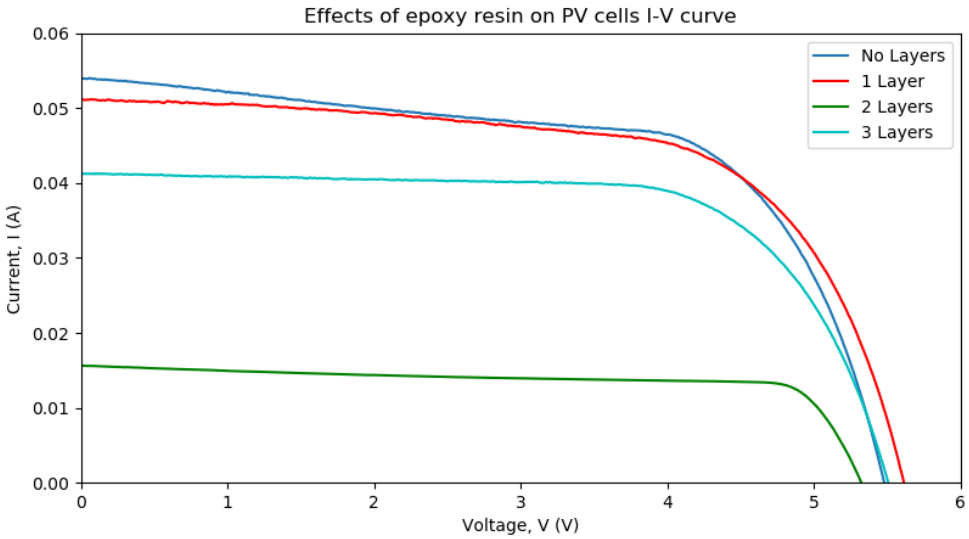


Figure 5.5: Experimental results on PV cells

Figure 5.1 demonstrates the results of the experiment described in 5.1. As expected, the sample without epoxy shown the best result. As a reference, the author assumed the efficiency (in standard testing conditions - STC) for this sample. However, the tests weren't conducted with these conditions and the resulting efficiency values serve only the purpose of comparison between the samples on the test. This result is followed by the samples with 1 and 3 coats of epoxy resin, with efficiencies of 13.32% and 11.23% respectively. At last, we have the sample with two layers. This result was not expected and deviates substantially from the other samples. This may result from: uneven application of epoxy, process variation [67] or variations in the experiment conditions. This deviation could be attenuated with the use of a bigger sample size.

Table 5.1: Effect of Epoxy over the Maximum Power Point

No. of Layers	P_{max} (W)	V_{mp} (V)	I_{mp} (A)	V_{oc} (V)	I_{sc} (A)	$\eta(\%)$
0	0.189	4.214	0.045	5.47	0.054	13.59
1	0.186	4.314	0.043	5.62	0.051	13.32
2	0.061	4.754	0.013	5.32	0.016	4.35
3	0.156	4.224	0.037	5.52	0.041	11.23

From this results, we can conclude that the efficiency of the PV panel decreases with the thickness of the epoxy's coat. For smaller quantities of epoxy no additional calculation is needed, as we can observe from the Table 5.1, the difference between having zero or one layers of epoxy is (almost) imperceptible.

In addition to this, by analysing the obtained values for P_{max}, it is possible to infer that the use of one solar panel (Seeedstudio 313070001) would be enough to provide sufficient power for SB4S operation ($P_{max} > 0.151$) in the conditions of the test.

5.3.2 Data transfer duration and integrity

SB4S took around 3,9 seconds to transfer the 1830 bytes correspondent to 1 second of acquired data. At this rate, 3.8Kbps average, we would took about 2 hours and 25 minutes to send all the data measured on a 5 hour session. This value exceeds the limit established during the technology selection stage, and is far from being acceptable. In the author's opinion, this may result from the preemption overhead on the other scheduled tasks. It could be solved by: reducing the number of scheduled tasks on the tasks queue and/or optimising the existing tasks.

5.3.3 Power consumption

During SB4S's initialisation, the device consumed about 6mA. This state wasn't considered during the power's estimation study. This will not be a problem, as this state only occurs when the device is booted. During IDLE, SB4S consumed 1.8mA instead of the estimated 1.12mA this may occur due to the estimation using Kallisto as the reference instead of using the Nordic's devkit. On SURF mode, SB4S has a better performance than the estimated. Instead of the 30mA estimated on the study (see Table), SB4S only consumes 4.5mA, when acquiring GPS signal, and 2.6mA after the procedure. The same occurred upon the TX state, in which SB4S consumed only 7mA when compared with the estimated value of 141mA.

The measured currents, during TX and SURF mode, are close to the consumption of Kallisto. It should be noted that the power consumed by "Kallisto Breadboard" may differ when compared to the Kallisto Board itself. These results reveal that the GNSS and Wi-Fi modules current consumptions never reach the peak values described in their datasheets, during the conducted tests, as these technologies' power consumption varies a lot with the environmental conditions.

5.3.4 Global Validation

One of the expected outcomes from this work, was to develop a low-cost solution to facilitate SB4S's market entry. The estimated BOM's (Bill of materials) cost rounds €81.84, for batch production, and €69,89, for mass production (see Table 5.2). This value was estimated using the same distributors used during the prototype's development: Mouser[5] and TME[6]. Due to not having designed a final PCB for SB4S, operation and minor building costs (e.g. PCB, wires, resistors, capacitors,...) weren't take in consideration. The overall cost of BOM doesn't take in consideration the final price of the Kallisto board, using instead a estimation based on the price of the separated modules.

[5]https://pt.mouser.com/
[6]https://www.tme.eu/pt/

Table 5.2: Production costs

Module	Price in € (Retail)	Price in € (Wholesale)	Min Quantity (Wholesale)
MCU	5,45	3,07	1000
IMU	3,99	1,8	1000
Magnetometer	1,49	0,648	1000
GPS	12,79	8,1	1000
WiFi	1,44	1,44	5000
PV Cells	43,04	43,04	1
MPPT	6,83	3,39	250
Battery	4,82	3,38	100
SD Card	8,82	8,41	25
Total	88,67	73,278	

The major cost on SB4S's production is attributed to its four solar panels. This stems from the fact that the PV panels used on our study were off-the-shelf solutions, as buying solar cells in bulk wasn't a good approach for a dissertation. However, in the context of mass producing this solution, would be the best approach for cost reduction.

Chapter 6

Conclusion and Future Work

Higher prices, low fidelity results and lack of strategy for creating a new market, were some of the issues found in the existing surf monitoring solutions which this work aimed to solve. The proposed solution was the development of a low-cost and low-profile device capable of acquiring data to feed an already existing algorithm, developed by Fraunhofer Portugal.

Some of the specifications of the system: use of Fraunhofer's Kallisto Platform, IMU, Magnetometer and GPS were already defined. And others had to be defined, such as wireless communication technology, storage method and charging method.

Smartboard for Surfing - SB4S uses a mixed solution of Wi-Fi and BLE to transfer data and interface with the user respectively. This choice resulted from the necessity of having a solution capable of providing enough throughput without having to consume a large amount of power outside the transfer period. The BLE module was already included, on Kallisto board, and the selected Wi-Fi module was the Espressif ESP8266EX due to its relatively low average current consumption (80mA) and a theoretical maximum data rate output of 72.2Mbps.

SB4S uses an SD Card as a storage method due to its overall low price (when compared with EEPROM and eMMC for the same data sizes). And due to the use of an industry-standard (SPI) interface in opposition to a proprietary one (in eMMC).

For the power supply system, SB4S uses a set of PV panels (SeeedStudio 313070001) connected to an MPPT (Linear Technology LT3652) and a battery (Cellavia L57340). This system was prefered to USB and Qi charging standards due to providing a waterproof solution without adding constraints to the device's usability. The dimensioning of this system required a study over the solar irradiance on Portuguese beaches. From this study, we concluded that to guarantee power to the system with 98% accuracy it would be needed a set of panels (constraint by the available area on the surfboard) capable of outputting at least 150mW with a solar irradiance of 15 W/m^2.

The firmware's development used the SDK provided by the Kallisto Platform to access its resources through a set of HALs and drivers. Wi-Fi and GNSS functionalities were inexistent in the SDK and had to be developed during the dissertation.

The system was validated by two different experiments: a study over the effect of epoxy resin on PV panels. And a laboratory simulation of the standard use case of SB4S.

The first experiment served the purpose of validating the use of a PV panel embedded into a surfboard coated with epoxy resin. From this experiment, we concluded that PV cells suffer from a residual performance decrease when exposed to a layer of epoxy.

Regarding the simulation on the standard device's use, was possible to understand that the performance of the system didn't meet the estimated results and that its overall power consumption was lower than the expected. Despite the low performance of the transmission, all the requirements were met. Nevertheless, there is space for improvement in this area, when removing the nordic's scheduler from the tests was possible to identify an improvement in the overall performance of the system. In the author's opinion, the optimization of the scheduled tasks and the decrease in the number of tasks would improve the overall system's performance.

In a future work would be good to validate the study on the effect of epoxy over PV cells on a larger scale, to minimize the presence of outliers in the conducted experiment. And to optimize the device's firmware in order to obtain a better performance, especially during its data transfer. With the achievement of these two goals and the production of a final PCB, SB4S could have its last validation by a focus group, of surfers, and be tested on the field where new metrics could be analysed: PV cells' generated power during a surf-session; sensor's positioning on the surfboard; GNSS module fix-time and others.

Appendix A

Solar Irradiance Probability Script

```
import requests as req
import sys
from collections import Counter

base_url = 'https://re.jrc.ec.europa.eu/api/tmy'

coords = [
        (41.676, -8.827), # Viana do Castelo - Praia do Cabedelo
        (41.432, -8.783), # Povoa de Varzim - Praia da Agucadoura
        (41.193, -8.706), # Matosinhos - Praia de Leca
        (41.176, -8.690), # Matosinhos - Praia de Matosinhos
        (41.008, -8.646), # Espinho - Praia da Baia
        (40.138, -8.862), # Figueira da Foz - Praia do Cabedelo
        (39.614, -9.083), # Nazare - Praia do Norte
        (39.339, -9.360), # Peniche - Praia do Medao Grande
        (38.988, -9.418), # Ericeira - Praia de Ribeira d'Ilhas
        (38.681, -9.333), # Cascais - Praia de Carcavelos
        (38.556, -9.188), # Almada - Praia da Costa da Caparica
        (37.921, -8.803), # Sines - Praia de Sao Torpes
        (37.522, -8.787), # Odemira - Praia da Zambujeira do Mar
        (37.293, -8.864), # Aljezur - Praia da Arrifana
        (37.198, -8.905), # Carrapateira - Praia da Bordeira
        (37.047, -8.872), # Vila do Bispo - Praia do Zavial
       (37.819, -25.543)  # Acores - Areal de Santa Barbara
        ]
```

```
def getData(lat, lon):
    payload = {
        'lat':lat,
        'lon':lon,
        'outputformat':"json"
    }

    res = req.get(base_url, params=payload)
    if(res.status_code == 200):
        return  res.json()['outputs']['tmy_hourly']
    else:
        print("Wrong coordinates")

def main():
    gh = []
    p = {}
    freqs = []
    ps = []
    cum = 0.0
    cums = []
    for coord in coords:
        print(coord)
        data = getData(coord[0], coord[1])
        if (data is not None):
            for sample in data:
                if(float(sample['G(h)'])>0):
                    gh.append(float(sample['G(h)']))
    c = Counter(gh)
    c = dict(sorted(c.items()))
    for freq in c:
        p[float(freq)] = float(c[freq])/len(gh)
        freqs.append(freq)
        ps.append(p[freq])
        cum += float(p[freq])
        cums.append(cum)
        print(str(freq)+"\t"+str(p[freq])+"\t"+str(cum))

main()
```

Appendix B

Full Charge Probability Script

```
import requests as req
from matplotlib import pyplot as plt
import sys
from collections import Counter

# Required instant power
p_sb4s = 0.15

# PV panel output power @ STC
p_stc = 3

# PV panel efficiency
eta = 13.59

# Area of one PV panel
area = 0.022

# Number of PV panels
n = 4

# Duration of a surf's session
t_surf = 5

# Battery charging factor
c = 0.5
```

```
coords = [
        (41.676, -8.827), # Viana do Castelo - Praia do Cabedelo
        (41.432, -8.783), # Povoa de Varzim - Praia da Agucadoura
        (41.193, -8.706), # Matosinhos - Praia de Leca
        (41.176, -8.690), # Matosinhos - Praia de Matosinhos
        (41.008, -8.646), # Espinho - Praia da Baia
        (40.138, -8.862), # Figueira da Foz - Praia do Cabedelo
        (39.614, -9.083), # Nazare - Praia do Norte
        (39.339, -9.360), # Peniche - Praia do Medao Grande
        (38.988, -9.418), # Ericeira - Praia de Ribeira d'Ilhas
        (38.681, -9.333), # Cascais - Praia de Carcavelos
        (38.556, -9.188), # Almada - Praia da Costa da Caparica
        (37.921, -8.803), # Sines - Praia de Sao Torpes
        (37.522, -8.787), # Odemira - Praia da Zambujeira do Mar
        (37.293, -8.864), # Aljezur - Praia da Arrifana
        (37.198, -8.905), # Carrapateira - Praia da Bordeira
        (37.047, -8.872), # Vila do Bispo - Praia do Zavial
       (37.819, -25.543)  # Acores - Areal de Santa Barbara
        ]

base_url = 'https://re.jrc.ec.europa.eu/api/tmy'

def getData(lat, lon):
    payload = {
        'lat':lat,
        'lon':lon,
        'outputformat':"json"
    }

    res = req.get(base_url, params=payload)
    if(res.status_code == 200):
        return  res.json()['outputs']['tmy_hourly']
    else:
        print("Wrong coordinates")

def main():
    samples = []
    p = {}
    cum = 0.0
    for coord in coords:
```

```
            print(coord)
            data = getData(coord[0], coord[1])
            if (data is not None):
                for sample in data:
                    if(float(sample['G(h)'])>15):
                        # Real Output Power Estimation
                        p_prod = float(sample['G(h)'])*(eta/100)*area*n
                        # Remaining Power
                        p_rem = (p_prod - p_sb4s) * t_surf * c
                        '''
                        Due to the charging factor there will be some
                        negative values. These samples need to be ignored.
                        '''
                        if(p_rem > 0):
                            samples.append(p_rem)
        count = Counter(samples)
        count = dict(sorted(count.items()))
        for p_rem in count:
            p[float(p_rem)] = float(count[p_rem])/len(samples)
            print(str(p_rem)+"\t"+str(cum))
            cum += float(p[p_rem])

main()
```

Appendix C

Surveys

Table C.1: WiFi Module Survey

Manufacturer	Model	Built-in Antenna	Supported Protocols	Processor	Freq. (GHz)	Data Rate (Mbps)	TWI	SPI	UART	Current (mA)		Voltage (V)	
										RX	TX	Min	Max
Microchip Technology	ATWINC1500	YES	b/g/n	Cortus APS3	2.4	72.2		X		61	266	3	3.6
Silicon Labs	WGM160P	OPT.	b/g/n	ARM Cortex-M4	2.4	72.2	X	X	X	240	248	3	3.6
Silicon Labs	WF111	YES	b/g/n	No	2.4	72.2		X		240	240	2.7	3.6
Espressif Systems	ESP32	YES	b/e/g/i/n	Telesilica Xtensa LX6	2.4	150	X	X	X	80	80	3	3.6
Murata Electronics	LBEE5HY1MW	NO	a/b/g/n/ac	N/A	2.4/5	433			X	130	420	3.2	4.2
Murata Electronics	LBEE5ZZ1MD	YES	b/g/n	ARM Cortex-M4	2.4	11/54/56	X	X	X	45	300	3	3.6
Texas Instruments	CC3235SF	NO	a/b/g/n	ARM Cortex-M4	2.4/5	72	X	X	X	67	318	2.1	3.6
Murata Electronics	LBEE5PA1LD	NO	b/g/n	ARM Cortex-M4F	2.4	11/56/65	X	X	X	100	410	3.2	3.6
Sierra Wireless	BX3105	YES	b/g/n	N/A	2.4	100	X	X	X	N/A	N/A	N/A	N/A
Espressif Systems	ESP8266EX	YES	b/g/n	Tensilica L106	2.4	72.2	X	X	X	80	80	2.5	3.6

Table C.2: GPS Module Survey

Manufacturer	Model	Built-in Antenna	GNSS Systems	Frequency Bands	Update Rate (Hz)	TWI	SPI	UART	Supply Current (mA)			Voltage (V)	
									Standby	Tracking	Acquisition	Min	Max
Linx Technologies	GPS-R4	No	GPS	GPS: L1 C/A	1 to 5	N/A	N/A	N/A	0.43	33	56	3	3.6
ST Microelectronics	Teseo-LIV3F	No	GPS Galileo Glonass BeiDou QZSS	GPS: L1 C/A BeiDou: B1	1	X	X		0.017	22.72	N/A	2.1	4.3
Maestro Wireless	A2200	No	GPS	N/A	1 to 5	X			0.025	41	52	3	3.6
Linx Technologies	GPS-F4	No	GPS	GPS: L1 C/A	1 to 5			X	0.020	33	56	1.71	1.89
Linx Technologies	GPS-FM	No	GPS	GPS: L1	1 to 10			X	0.15	12	14	3	4.3
u-blox	MAX-M8	No	GPS Galileo Glonass BeiDou	GPS: L1 C/A GLONASS: L1 BeiDou: B1 Galileo: E1	0.25 to 10	X		X	5.4	23	26	1.65	3.6
Sierra Wireless	XA1110	Yes	GPS + Glonass GPS+Glonass +Galileo	GPS: L1 GLONASS: L1 QZSS: L1 SBAS: L1 GALILEO: E1	10		X	X	1.6	9	24	3	4.3
Sierra Wireless	XM1210	No	GPS+Glonass GPS+BeiDou GPS+Galileo	GPS: L1 GLONASS: L1 QZSS: L1 SBAS: L1 GALILEO: E1	1	X		X	<2	<35	N/A	N/A	N/A

Table C.3: Surfboard's sizes survey

Brand	Model	Lenght	Width	Thikness
Lost	Hydra	4'10	19"	2.12"
Lost	Party Crasher	5'8"	19"	2.50"
Lost	Sword-Fish	5'4"	18.63"	2.25"
Lost	Rocket Redux	5'2"	18.50"	2.25"
Lost	Puddle Jumper HP Round	5'0"	19.25"	2.32"
Lost	Puddle Jumper HP	5'0"	19.25"	2.32"
Lost	Rad Ripper	5'2"	18.00"	2.18"
Lost	Evil Twin	5'2"	18.50"	2.20"
Lost	MR California Twin	5'2"	19"	2.25"
Lost	Cobra Killer	5'2"	18"	2.20"
Lost	Lost Maysym	5'0"	18.5"	2.18"
Lost	Smooth Operator	6'6"	20.50"	2.65"
Lost	Crowd Killer Round	6'6"	20.50"	2.56"
Lost	Crowd Killer	6'2"	20"	2.6"
Lost	Uber Driver XL	5'4"	18.75"	2.30"
Lost	Uber Driver	5'4"	18.25"	2.25"
Lost	El Patron	6'0"	20"	2.75"
Lost	El Patron Round	6'0"	19.64"	2.66"
Lost	Sub Driver	5'4"	18.25"	2.22"
Lost	Sabo Taj	5'4"	18"	2.25"
Lost	Retro Ripper	5'6"	18.38"	2.32"
Lost	Quiver Killer	5'2"	18.13"	2.13"
Lost	Round Nose Fish Retro	5'0"	19.50"	2.25"
Lost	Round Nose Fish REDUX	5'2"	18.50"	2.13"
Lost	Puddle Jumper Round Pin	5'0"	19"	2.25"
Lost	Freak Flag Bean Bag Lib Tech	5'2"	20.50"	2.35"
Lost	Retro Gun	6'2"	18.72"	2.38"
Lost	Driver 2.0	5'6"	18.13"	2.19"
Quiksilver	ST Super Grom	4'8"	18.15"	2.13"
Quiksilver	ST Comp Fish	5'10"	21"	2.8"
Quiksilver	ST Comp Egg	6'2"	20"	2.63"
Quiksilver	ST Comp Mini Malibu	7'	20.8"	2.75"
Quiksilver	ST Comp Longboard	9'	22.3"	2.87"
Quiksilver	Grace Purple Haze	6'8"	20"	5.63"
Quiksilver	Grace Big Boy	6'6"	20.5"	2.63"
Quiksilver	Grace Single Fin	5'10"	19"	2.38"
Quiksilver	Grace Minibu	5'5"	19.5"	2.38"
Quiksilver	Grace Demibu	6'6"	20.5"	2.63"
Quiksilver	Bradley Thing	5'5"	18.5"	2.13"
Quiksilver	Bradley CB1	5'9"	18"	2.13"
Quiksilver	Bradley Devil	5'10"	18.5"	2.25"
Quiksilver	Bradley Viper 4	5'6"	18.5"	2.25"
Quiksilver	Bradley Pie	5'6"	19"	2.25"
Quiksilver	Phipps MK9	5'10"	18.5"	2.25"
Quiksilver	Phipps MK11	5'8"	18.75"	2.32"
Quiksilver	Phipps Hartza	7'6"	20"	2.75"
Quiksilver	Phipps One Bad Egg	6'8"	20.25"	2.63"
Quiksilver	Phipps Snowshoe	6'0"	20.25"	2.38"
Quiksilver	Phipps Bobby Dazzler	5'6"	20.5"	2.5"
Quiksilver	Phipps Shrimpboat	5'4"	18.75"	2.32"
Quiksilver	Grace Little Boy	5'8"	19.75"	2.38"

Table C.4: PV Panel Survey

Manufacturer	Model	W	L	Power @ STC	Efficiency	Power Output	Required Panels	Area?	Redundancy?	Flexible	Needs Disassembling
		m	m	W	%	W					
Panasonic	AM5902	0,15	0,038	0,3	5,26	0,0045	34,000	FALSE	FALSE	Yes	No
SeedStudio	5W CIGS	0,35	0,17	5	8,40	0,075	3,000	FALSE	FALSE	Yes	No
PowerFilm	PT7.2-75	0,073	0,253	0,86	4,66	0,0129	12,000	FALSE	FALSE	Yes	No
PowerFilm	PT15-300	0,325	0,27	3	3,42	0,045	4,000	FALSE	FALSE	Yes	No
PowerFilm	PT7.2-150F	0,175	0,27	1,73	3,66	0,02595	6,000	FALSE	FALSE	Yes	No
SeedStudio	313070002	0,081	0,137	1,5	13,52	0,0225	7,000	TRUE	FALSE	No	No
CELLEVIA	CL-SM10P	0,354	0,251	10	11,25	0,15	2,000	FALSE	FALSE	N/A	Yes
Xunzel	SolarPower-HD 5W	0,58	0,31	20	11,12	0,3	1,000	FALSE	FALSE	N/A	No
Xunzel	SolarPower 10W	0,35	0,265	10	10,78	0,15	2,000	FALSE	FALSE	N/A	Yes
Xunzel	MS6V900	0,18	0,31	5,4	9,68	0,081	2,000	TRUE	FALSE	N/A	No
Decatlhon	Forclaz Trek100	0,35	0,29	10	9,85	0,15	2,000	FALSE	FALSE	N/A	Yes
SeedStudio	313070001	0,138	0,16	3	13,59	0,045	4,000	TRUE	FALSE	No	No
SeedStudio	313070003	0,18	0,08	2	13,89	0,03	6,000	TRUE	FALSE	No	No

Table C.5: Batteries Survey

Manufacturer	Model	Voltage (V)	Capacity (mAh)	Capacity (Wh)	Weight (g)
MIKROE	112	3,7	2000	7,4	38
VARTA	56427 201 018	3,7	1400	5,18	23
BAK	LP-503759-1S-3	3,7	1350	4,995	25
VARTA	56457 201 012	3,7	1000	3,7	20
BAK	LP-523450P-IS-3	3,7	950	3,515	20
PKCELL	667100	3,7	3900	14,43	77
PKCELL	456473	3,7	2000	7,4	40
PKCELL	53562	3,7	1000	3,7	20,5
Renata	ICP606168PRT	3,7	2800	10,36	70
Renata	ICP543759PMT	3,7	1320	4,884	26
Renata	ICP622540PMT	3,7	600	2,22	11
CELLEVIA	L453350	3,7	800	2,96	18,88
CELLEVIA	L603048	3,7	850	3,145	20
CELLEVIA	L573450	3,7	980	3,626	22,8
CELLEVIA	L503759	3,7	1350	4,995	25,6

Table C.6: NB-IoT Modules Survey

Manufacturer	Model	Region	Approvals	Fallback	2G	LTE	Peak Data Rate		Location Services	Operating voltage (V)			Current (mA)
					Frequency Bands	Frequency Bands	Downlink	Uplink	Satellite Systems	min	typ	max	
u-box	SARA-R510M8	Global	FCC, ISED, GCF, PTCRB, KCC, ICASA, NCC, Giteki, RCM	N/A	-	B1, B2, B3, B4, B5, B8, B12, B13, B18, B19, B20, B25, B26, B28, B66, B71, B85	127 Kbps	159 Kbps	GPS GLONASS BeiDou Galileo	3	N/A	4,5	N/A
Nordic Semiconductor	nRF9160	Global	GCF, PTCRB, FCC, CE, ISED, ACMA, RCM, NCC, IMDA, MIC, MSIP, IFT, ICASA, N/A	No	-	B1, B2, B3, B4, B5, B8, B12, B13, B14, B17, B20, B25, B26, B28, B66	375 Kbps	300 Kbps	GPS	-0,3	N/A	5,5	150 mA
Sierra Wireless	HL7802	Global	FCC, GCF, IC, JRD, JPA, PTCRB	No	-	B1, B2, B3, B4, B5, B8, B9, B10, B12, B13, B14, B17, B18, B19, B20, B25, B26, B27, B28, B66	300 Kbps	375 Kbps	GPS GLONASS	3,2	3,7	4,35	500 mA
Telit	ME310G1-WW	Global	FCC, RED, GCF, PTCRB	GPRS	B2, B3, B5, B8	B1, B2, B3, B4, B5, B8, B12, B13, B14, B17, B18, B19, B20, B25, B26, B27, B28, B66, B85	365 Kbps	1 Mbps	GPS GLONASS BeiDou Galileo	3,8			N/A
Telit	ME310G1-W1	Global	FCC, RED, GCF, PTCRB	No	-	B1, B2, B3, B4, B5, B8, B12, B13, B14, B17, B18, B19, B20, B25, B26, B27, B28, B66, B85	365 Kbps	1 Mbps	GPS GLONASS BeiDou Galileo	3,8			N/A
Quectel	BG95-M3	Global	GCF, CE, FCC, PTCRB, RCM, IC, JATE, KS, IFETEL, IMDA, NCC, CCC	GPRS	B2, B3, B5, B8	B1,B2,B3,B4,B5,B8,B12, B13,B14,B17,B18,B19,B20, B25,B26,B27,B28,B31,B66,B71	375 Kbps	1,2 Mbps	Optional: GPS GLONASS BeiDou Galileo	2,4	3,8	4,8	TBD
Quectel	BG600L-M3	Global	N/A	GPRS	B2, B3, B5, B8	B1, B2, B3, B4, B5, B8, B12, B13, B14, B18, B19, B20, B25, B26, B27, B28, B66, B85	N/A	N/A	-	N/A	N/A	N/A	TBD
Quectel	BG77	Global	GCF; CE; FCC; RCM; IC; KC; IFETEL; IMDA; NCC; CCC	No	-	B1, B2, B3, B4, B5, B8, B12, B13, B14, B17, B18, B19, B20, B25, B26, B27, B28, B66, B71	375 Kbps	1,2 Mbps	Optional: GPS GLONASS BeiDou Galileo	2,4	3,8	4,8	TBD
Quectel	BG96	Global	GCF; CE; FCC; PTCRB; IC; Anatel; IFETEL, CCC, KC, NCC, JATE, TELEC, RCM, NBTC, IMDA, ICASA	GPRS	B2, B3, B5, B8	B1, B2, B3, B4, B5, B8, B12, B13, B18, B19, B20, B26, B28, B39	375 Kbps	375 Kbps	Optional: GPS GLONASS BeiDou Galileo	3,3	3,8	4,3	N/A
SIMCom	SIM7070E	Global	CE; RCM	GPRSS	B2, B3, B5, B8	B1, B2, B3, B4, B5, B8, B12, B13, B14, B18, B19, B20, B25, B26, B27, B28, B31, B66, B72, B85	589 Kbps	1 Mbps	GPS GLONASS BeiDou Galileo	3,2	N/A	4,2	N/A

References

[1] R. Tei, H. Yamazawa, and T. Shimizu. Ble power consumption estimation and its applications to smart manufacturing. In *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 148–153, 2015.

[2] Carla Dellaserra, Yong Gao, and Lynda Ransdell. Use of integrated technology in team sports: A review of opportunities, challenges, and future directions for athletes. *Journal of strength and conditioning research / National Strength Conditioning Association*, 28, November 2013.

[3] Gobinath Aroganam, Nadarajah Manivannan, and David Harrison. Review on wearable technology sensors used in consumer sport applications, 2019.

[4] Diana Gomes, Dinis Moreira, João Costa, Ricardo Graça, and João Madureira. Surf session events' profiling using smartphones' embedded sensors. *Sensors*, 19(14):3138, jul 2019. doi:10.3390/s19143138.

[5] International Surfing Association. Olympic surfing. https://www.isasurf.org/olympicsurfing/. [Online; accessed 25-November-2019].

[6] Prnewswire. The global surfboard market is forecasted to grow at a cagr of 12.24during the period 2018-2022. https://www.prnewswire.com/news-releases/the-global-surfboard-market-is-forecasted-to-grow-at-a-cagr-of-1224-during-the-period-2018-2022-300577382.html.

[7] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7, 2011.

[8] J. Madureira, R. Lagido, and I. Sousa. Comparison of number of waves surfed and duration using global positioning system and inertial sensors. *International Journal of Biomedical and Biological Engineering*, 9(5):444–448, 2015.

[9] Rip Curl. Rip curl search gps. https://searchgps.ripcurl.com.

[10] Diana Gomes, Dinis Moreira, and João Madureira. Detection and characterization of surfing events with smartphones' embedded sensors. *2018 12th International Conference on Sensing Technology (ICST)*, pages 132–138, 2018.

[11] SLIDE-R. Slide-r website. http://www.slide-r.com/.

[12] Loïc Griffié. Slide-r: The world's first premium connected surfboard. https://www.kickstarter.com/projects/647736013/slide-r-the-worlds-first-premium-connected-surfboa. [Online; accessed 25-November-2019].

[13] Pukas Surf. Pukas / surfsense by tecnalia. https://pukassurf.com/surfboards/surfsense/. [Online; accessed 27-January-2020].

[14] Aaron Saenz. The future of surfing – sensors in the board record everything. https://singularityhub.com/2011/03/03/check-out-the-future-of-surfing-sensors-in-the-board-record-everything-video/. [Online; accessed 27-January-2020].

[15] Olivia Solon. Surfing on a circuitboard: analytics help surfers hone technique. https://web.archive.org/web/20150915173647/https://www.wired.co.uk/news/archive/2011-03/04/surfsens-future-of-surfing. [Online; accessed 27-January-2020].

[16] Daniel Bona, Maria Marques, and Miguel Correia. *Instrumentation of a surfboard to evaluate surfing performance*. February 2014.

[17] A. Costa, M. Jordão, H. Machado, H. Miranda, and N. Borges Carvalho. Surf board electronic device. In *2015 1st URSI Atlantic Radio Science Conference (URSI AT-RASC)*, pages 1–1, 2015.

[18] Oliver Farley, Chris Abbiss, and Jeremy Sheppard. Performance analysis of surfing: A review. *Journal of strength and conditioning research / National Strength Conditioning Association*, 31, April 2016.

[19] Raoni Caselli, Marcelo Ferreira, and Berenice Gonçalves. Analysis of tools to measure the user experience during the sports practice of recreational surfing. volume 496, pages 71–83. January 2017.

[20] Beatriz Minghelli. Time-motion analysis in surf: benefits. *International Journal of Sport, Exercise and Health Research*, 2:97–99, January 2018.

[21] Navigation Center, U.S. Department of Homeland Security. NAVSTAR GPS USER EQUIPMENT INTRODUCTION. PUBLIC RELEASE VERSION. https://www.navcen.uscg.gov/pubs/gps/gpsuser/gpsuser.pdf, 1996.

[22] Richard B. Langley. Dilution of precision. *GPS world*, 10(5):52–59, 1999.

[23] National Coordination Office for Space-Based Positioning, Navigation, and Timing. Selective availability. https://www.gps.gov/systems/gps/modernization/sa/.

[24] Sparkfun. Accelerometer basics. https://learn.sparkfun.com/tutorials/accelerometer-basics. [Online; accessed 20-January-2020].

[25] K. Tanaka, Y. Mochida, S. Sugimoto, K. Moriya, T. Hasegawa, K. Atsuchi, and K. Ohwada. A micromachined vibrating gyroscope. In *Sensors and Actuators A: Physical*, pages 278–281. IEEE, 1995.

[26] Zheng You. Chapter 9 - magnetometer technology. In *Space Microsystems and Micro/-nano Satellites*, pages 341–360. Butterworth-Heinemann, 2018. URL: http://www.sciencedirect.com/science/article/pii/B9780128126721000096.

[27] GSM Association. Mobile industry unites to drive universal charging solution for mobile phones. http://web.archive.org/web/20090217192039/http://www.gsmworld.com/newsroom/press-releases/2009/2548.htm, 2009. [Online; accessed 20-January-2020].

[28] Andrija Stojkovic. Usb charging - embedded devices. https://www.toradex.com/blog/usb-charging. [Online; accessed 20-January-2020].

[29] S. Y. Hui. Planar wireless charging technology for portable electronic products and qi. *Proceedings of the IEEE*, 101(6):1290–1301, 2013.

[30] Wireless Power Consortium. Introduction to the power class 0 specification. https://www.wirelesspowerconsortium.com/knowledge-base/specifications/download-the-qi-specifications.html.

[31] Christiana Honsberg; Stuart Bowden;. Light generated current. https://www.pveducation.org/pvcdrom/solar-cell-operation/light-generated-current. [Online; accessed 31-May-2020].

[32] Dgs Altener Eu Ist and U. Europeia. Energia fotovoltaica-manual sobre tecnologias, projecto e instalação. *Janeiro de*, page 368, 2004.

[33] Christian Schuss and Timo Rahkonen. Solar energy harvesting strategies for portable devices such as mobile phones. pages 132–139. IEEE, 2013.

[34] Davide Brunelli, Clemens Moser, Lothar Thiele, and Luca Benini. Design of a solar-harvesting circuit for batteryless embedded systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(11):2519–2528, 2009.

[35] Thomas Søderholm. 6 wireless technologies for wearables. https://blog.nordicsemi.com/getconnected/wireless-technologies-for-wearables. [Online; accessed 14-May-2020].

[36] M. Chen, Y. Miao, Y. Hao, and K. Hwang. Narrow band internet of things. *IEEE Access*, 5:20557–20577, 2017.

[37] Ronan Le Bras. What is the difference in data throughput between lte-m/nb-iot and 3g or 4g? https://www.gsma.com/iot/resources/what-is-the-difference-in-data-throughput-between-lte-m-nb-iot-and-3g-or-4g/. [Online; accessed 14-May-2020].

[38] Y.-P. Eric Wang, Xingqin Lin, Ansuman Adhikary, Asbjorn Grovlen, Yutao Sui, Yufei Blankenship, Johan Bergman, and Hazhir S. Razaghi. A primer on 3gpp narrowband internet of things. *IEEE communications magazine*, 55(3):117–123, 2017.

[39] R. Heydon. *Bluetooth Low Energy: The Developer's Handbook*. Pearson Always Learning. Prentice Hall, 2012. URL: https://books.google.pt/books?id=Ag2tXwAACAAJ.

[40] Bluetooth SIG. Learn about bluetooth - radio versions. https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/radio-versions/.

[41] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai. Ieee 802.11 wireless local area networks. *IEEE Communications Magazine*, 35(9):116–126, 1997.

[42] Intel Corporation. Different wi-fi protocols and data rates. https://www.intel.com/content/www/us/en/support/articles/000005725/network-and-i-o/wireless-networking.html.

[43] T. Haifley. Endurance of eeproms with on-chip error correction. *IEEE Transactions on Reliability*, R-36(2):222–223, 1987.

[44] P. Pavan, R. Bez, P. Olivo, and E. Zanoni. Flash memory cells-an overview. *Proceedings of the IEEE*, 85(8):1248–1271, 1997.

[45] Crystal Chang. What to know before storing data on a sd memory card. https://www.sdcard.org/press/thoughtleadership/170710What_tot_Knowt_Beforet_Storingt_Datat_ont_at_SDt_Memoryt_Card.html. [Online; accessed 21-May-2020].

[46] Panasonic. emmc vs. sd cards: A head-to-head comparison. https://na.industrial.panasonic.com/emmc-vs-sd-cards.

[47] MultiMediaCard Association. Mmca: Home. https://web.archive.org/web/20090504081108/http://www.mmca.org/.

[48] JEDEC SOLID STATE TECHNOLOGY ASSOCIATION. Embedded multi-media card (e•mmc) electrical standard (5.0). techreport, JEDEC SOLID STATE TECHNOLOGY ASSOCIATION, September 2013.

[49] Dale DePriest. gpsinformation.org | nmea data. URL: https://www.gpsinformation.org/dale/nmea.htm#nmea.

[50] Alberto Mendez-Villanueva and David John Bishop. Physiological aspects of surfboard riding performance. *Sports medicine (Auckland, N.Z.)*, 35:55–70, 02 2005. doi:10.2165/00007256-200535010-00005.

[51] Bluetooth SIG. Understanding bluetooth range. . [Online; accessed 20-June-2020].

[52] Nordic Semiconductor. *nRF52840 - Product Specificiation*.

[53] Nordic Semiconductor. *nRF9160 - Product Specificiation*.

[54] Espressif Systems. *ESP8266EX Datasheet*. Espressif Systems, 2018. Version 5.8.

[55] STMicroelectronics. Teseo-liv3f datasheet. https://www.st.com/resource/en/datasheet/teseo-liv3f.pdf.

[56] NANIUM. Iotip current consumption analysis. Technical report, 2016.

[57] Christiana Honsberg; Stuart Bowden;. Measurement of solar cell efficiency. https://www.pveducation.org/pvcdrom/characterisation/measurement-of-solar-cell-efficiency.

[58] Christiana Honsberg; Stuart Bowden;. Calculation of solar insolation. https://www.pveducation.org/pvcdrom/properties-of-sunlight/calculation-of-solar-insolation.

[59] European Commision. Photovoltaic geographical information system (pvgis). https://ec.europa.eu/jrc/en/pvgis.

[60] Surfer Today. The best surf spots in portugal. URL: https://www.surfertoday.com/surfing/the-best-surf-spots-in-portugal.

[61] Marilia Poiares. 10 best surfing hotspots in portugal. URL: https://www.idealista.pt/en/news/lifestyle-portugal/2019/05/21/332-10-best-surfing-hotspots-portugal.

[62] Quiksilver. Quiksilver. URL: https://www.quiksilver.com/.

[63] Lost Surfboards. Lost surfboards by mayhem. URL: https://lostsurfboards.net/.

[64] Espressif Systems. Esp at commands set. https://github.com/espressif/esp-at/blob/master/docs/en/get-started/ESP_AT_Commands_Set.md.

[65] Nordic Semiconductor. Nordic semiconductors infocenter: Current measurement with an ampere-meter. https://infocenter.nordicsemi.com/topic/ug_nrf52840_dk/UG/nrf52840_DK/images/pca10056_current_measurement_with_ampere_meter.svg.

[66] Nordic Semiconductor. Nordic semiconductors infocenter: Preparing the dk for current measurements. https://infocenter.nordicsemi.com/topic/ug_nrf52840_dk/UG/nrf52840_DK/images/pca10056_prepare_board_for_current_measurements.svg.

[67] M. J. Hossain, G. Gregory, E. J. Schneller, A. M. Gabor, A. L. Blum, Z. Yang, D. Sulas, S. Johnston, and K. O. Davis. A comprehensive methodology to evaluate losses and process variations in silicon solar cell manufacturing. *IEEE Journal of Photovoltaics*, 9(5):1350–1359, 2019.