Universidade do Minho
Escola de Engenharia

Carlos Miguel Sousa e Silva

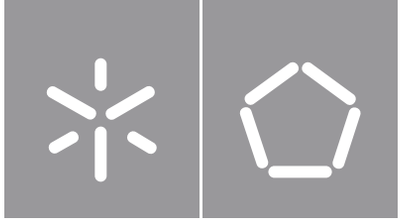
Monitoring of Marine Animais - Development and Testing of an Oximetry and Heartbeat Sensor

junho de 2022

Monitoring of Marine Animais - Development and Testing of an Oximetry and Heartbeat Sensor

Carlos Miguel Sousa e Silva

UMinho | 2022



Universidade do Minho
Escola de Engenharia

Carlos Miguel Sousa e Silva

Monitoring of Marine Animais - Development and Testing of an Oximetry and Heartbeat Sensor

Dissertação de Mestrado
Mestrado em Engenharia Eletrónica Industrial e Computadores
Instrumentação e Microssistemas Eletrónicos

Trabalho efetuado sob a orientação do
Professor Doutor Luís Gonçalves

junho de 2022

Este trabalho foi cofinanciado por fundos nacionais através da FCT - Fundação para a Ciência e Tecnologia no âmbito do Projeto de Investigação "ASTRIIS .: *Atlantic Sustainability Through Remote and Integrated In-situ Solutions*, ref.: POCI-01-0247-FEDER-046092,cofinanciado pelo Fundo Europeu de Desenvolvimento Regional –FEDER, através do SI&IDT Programas Mobilizadores, Programa Operacional de Competitividade e Internacionalização (POCI) -COMPETE 2020, do Portugal 2020,tendo como Organismo Intermédio de acompanhamento à realização do investimento, a Agência Nacional de Inovação –ANI

Acknowledgements

This dissertation marks the end of a journey. A journey through a difficult road, full of bumps, holes and crossroads that would not have been made, had I been alone. So, this won't feel like the end if I don't thank everyone that accompanied me and helped me to push through it.

To my parents, Maria Manuela and João Luís, for all the love and support, all the sacrifices made, all the patience you had dealing with me and mostly, all the guidance you gave to try and prepare me for what this world has to offer, I give the most special thank you. Without both I would not be who I am nor where I am.

To my brother Luís, who was also a guide and a friend throughout my whole life, thank you for all the advice and companionship.

A very special thanks to my grandparents, António Neves, and Rosa Cardoso, for always being present and for making sure that in my life I never lacked anything, even if it meant giving me everything they had.

A big thank you to my supervisor, Professor PhD Luís Gonçalves, for all the knowledge given, for the availability and for the support through this project. Without him it would have not been possible to accomplish this work.

To all my battle brothers and sisters, with special care for Merkel, Dimitri, Pipi, Xico, Rocky, Sofia, Monte, Ponto, Technão and Piggy, thank you for all the help, support, tears, and laughs. This trip was definitely easier with you guys by my side.

Also, a great thanks to Henrique, Martinho, Francisco, and Antero, for giving me the truest friendship anyone could ask for.

Lastly, to the person who fought all the battles by my side, who supported me through all the lows and shared all the highs. Who always helped me see the light when everything was dark. To Carolina, my wild rose, thank you.

Statement of Integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Resumo

A monitorização marinha tem se tornado bastante popular, com muitas instituições a usar "*tags*", dispositivos invasivos ou não-invasivos que são implantados em animais para monitorizar alguns parâmetros, usualmente relativos ao seu meio circundante, como a temperatura e a pressão da água, a sua localização, etc. Estes dispositivos são importantes, no entanto, são maioritariamente usados para monitorização de movimento e ainda existem muito poucos meios de monitorizar o que se está a passar dentro dos corpos durante algo como um mergulho profundo, durante caças ou para apenas verificar se o espécimen está saudável.

Esta dissertação representa uma tentativa de replicar a tecnologia que é mais comumente utilizada para a medição de saturação de oxigénio e ritmo cardíaco em humanos, adaptá-la e aplicá-la em animais marinhos. O método denomina-se Oximetria de Pulso.

Dois dispositivos desenvolvidos com a medição de oximetria de pulso em humanos em mente foram escolhidos, o MAX30100 e o MAX30110 da MAXIM Integrated. Foram analisados, estudados e testados *in vivo* num animal marinho (Raia Curva), o que deu um vislumbre de como se deve abordar o desenvolvimento de um oxímetro de pulso completo, pequeno, de baixo consumo, versátil para poder ser aplicado na maioria dos animais marinhos e capaz de ser integrado num sistema de monitorização.

Os testes realizados em ambiente controlado revelaram resultados promissores relativamente ao uso desta tecnologia em animais marinhos, sem haver a necessidade de grandes adaptações, mas também demonstrou que o verdadeiro desafio é aplicá-la de tal forma que os resultados sejam viáveis.

Palavras-Chave: Monitorização Marinha, Oximetria de pulso, não-invasivo, baixo consumo.

Abstract

Marine monitoring has become quite popular with lots of institutions using tags, invasive or non-invasive devices that are attached to animals to monitor certain parameters, usually of their surrounding mean, like water temperature, water pressure, their location, etc. These devices are important, however, they are mostly used for motion tracking and there are still very few means of monitoring what's going on inside of their bodies during somethings like deep dives, when they hunt or even if they are in a healthy condition.

This dissertation represents an attempt to replicate the technology that is most commonly used for oxygen saturation and heart rate measurement on humans, adapt it and apply it on marine animals. This method is called Pulse Oximetry.

Two devices developed for pulse oximetry measuring on humans were chosen, the MAX30100 and MAX30110, from MAXIM Integrated. They were analysed, studied, and tested in vivo on a marine animal (Undulate Ray), which gave an insight on how to approach the development of a small, non-invasive and low-power complete pulse oximeter, versatile so it can be applied in most marine animals and capable of being integrated in a monitoring system.

The tests carried out in a controlled environment showed promising results about the use of this technology on marine animals without any major adaptation, but also showed that applying it in such a way that the results are viable is the real challenge.

Key Words: Marine Monitoring, Pulse Oximetry, non-invasive, low-power.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two-Dimensional.
3D	Three-Dimensional.
AC	Alternate Current.
ADC	Analog to Digital Converter.
ALC	Ambient Light Cancellation.
BPM	Beats per Minute.
CO_2	Carbon Dioxide
DC	Direct Current.
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
Hb	Deoxyhemoglobin.
HbO_2	Oxygenated Hemoglobin.
HPF	High Pass Filter.
LED	Light Emitting Diode.
LPF	Low Pass Filter.
PCB	Printed Circuit Board.
PPG	Photoplethysmogram.
SaO_2	Blood Oxygen Saturation
SMD	Surface Mount Device.
SNR	Signal to Noise Ratio.
SpO_2	Peripheral blood oxygen saturation
SRAM	Static Random Access Memory

1. Introduction

In this chapter, the motivation, contextualization, the objectives of the project and the structure of the document are presented. The motivation consists of a brief introduction on why it is important to develop this sort of projects and the benefits it could bring for the future of ocean monitoring. The contextualization provides a small explanation of how the core technique that is used works, how it is used and how it is applied in the context of this dissertation. The next point is an enumeration and description of the objectives that were defined in order to achieve valid results. The last point of this chapter acts as a guide for the reader through the whole document.

1.1 Motivation

In a world that is being tormented by pollution and over-hunting, improving the ways of monitoring ecosystems and their inhabitants is crucial. Some monitoring/tracking devices are available for use in these applications, mainly implementing motion tracking. However, monitoring the physiological parameters of marine animals (e.g., oxygenation, heart rate) is still a very recent reality for ocean biologists. This kind of monitoring not only unveils the heath state of the animal, which allows to better understand the diseases that might plague it or could be a direct indicator of a problem in the ecosystem but can also be used to uncover something like the physiological phenomena that occur during the long deep dives of some cetaceans.

Having said this, having small, non-invasive devices that can track the vital signs of a marine animal, while causing minimal impact on its lifestyle has become a necessity since it is a less harmful way than conventional methods.

1.2 Contextualization

A typical human being can go on average, about three months without food and three days without water, but only about two or three minutes without oxygen. After that, the lack of oxygenation to the brain causes a faint - a protection mechanism of the body to force the person to start breathing again. It is safe to say that oxygen (O_2) is one of the most important elements required to sustain life. O_2 is acquired through the lungs in the process of respiration, where it bonds with red blood cells which become Oxygenated Hemoglobin (HbO_2) and flow through the circulatory system to provide O_2 to the body. Then, after

delivering, they bond with the CO_2 from the cells and become Deoxygenated Hemoglobin (Hb) and go to the lungs again, to the pulmonary alveoli, to replace the CO_2 with O_2, completing the cycle (Figure 1.1).

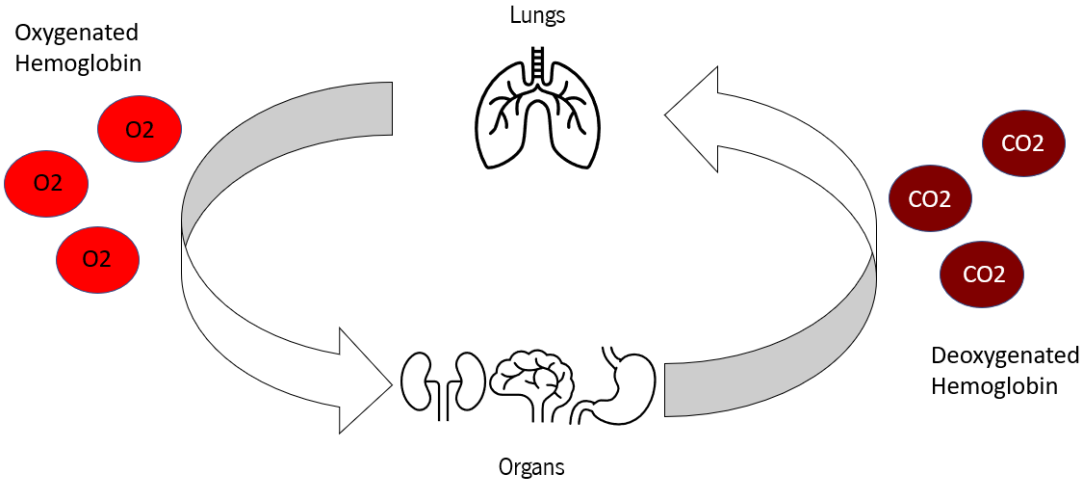


Figure 1.1 - Respiration Cycle

Blood oxygen saturation (SaO2) is the ratio of HbO_2 to total Hb, and usually in a healthy human specimen it should be between 95-100%. Even expert clinicians are only able to detect signs of low SaO_2 when the patients skin starts turning blue, at about 85%. When this happens, the doctor might only have close to three minutes to prevent the risk of brain damage, heart failure or death [1].Because of these risks, it is very important to keep track of SaO_2 as it is a very good indicator of a patient's health, human or non-human. The most common technique to estimate SaO_2, and the one that will be approached in this dissertation is Pulse Oximetry.

Pulse oximetry is a non-invasive test for estimating SaO_2 by interpreting the peripheral blood oxygen saturation (SpO_2) using two light sources. It is also able to detect heart pulses, so it can also be used to calculate the heart rate. Typically, an infrared light emitting diode (LED) and a red or green LED are used along with a photodiode to capture the light. Depending on the measurement site, either the transmissive or reflective method can be used. For the transmissive mode, the LEDs are placed on the opposite side of the photodiode. The reflective mode uses the LEDs and photodiode on the same side. Transmissive mode pulse oximeters are the ones used most due to their accuracy and stability since the light sources are directly transmitted to the photodiode. This operation, on humans, is usually carried out on a finger or on the ear lobe, because this technique requires that the place of measurement must be thin [2].
For this reason, the demand for reflective mode oximetry is increasing as it does not require a thin measurement site. It can be used in diverse measurement sites such as the feet, forehead, chest, and wrists, as long as its vascularized [3].

Pulse oximeters have been commercially available for some time and have seen a tremendous increase in popularity, becoming a standard in critical care monitoring devices in hospitals and operating rooms. The instrument transcutaneously estimates the oxygen saturation of arterial blood and provides vital information about a patient's cardiorespiratory functions.

There are already a lot of options of pulse oximeters for humans, and they work in most land mammals, but as for wild marine animals in general the story is not the same. There are not a lot of options, what exists is usually invasive and doesn't provide real time measurements. Having something like a regular pulse oximeter for humans, but specialized for marine animals, would provide a lot of insight into what happens inside their bodies during their normal daily activities, when they do deep dives or when they hunt, and would also allow to keep track of their health.

The focal point of this dissertation would then be to develop a prototype of a device that allows the application of this technique on most marine animals. It may seem odd to try and develop a device that works both on fish and on mammals, but, despite some differences, the blood vessels of fish are analogous to those of other vertebrates. However, the heart of most fish does not pump oxygenated blood like the heart of a mammal, but rather venous blood to the gills for respiration.

As it is important for the device to be versatile, the technique studied in this project is the reflective mode pulse oximetry. This only makes sense accounting the fact that it is a lot more forgiving regarding the choice of location to take measurements.

1.3 Objectives

The final goal of this dissertation is to develop a prototype of a device that can be applied along with a coupling mechanism to marine animals, to take heart rate and SpO_2 measurements using reflective mode pulse oximetry. The prototype will be paired with additional circuitry, to transmit the measurements via UART, so there will be no need for storing data. To aid in achieving this, a set of objectives was envisioned as seen below.

- **Comparison of commercially available sensors**

  This is the first objective of this dissertation, as it is directed towards the development of a fully operational prototype as it was said above and using commercially available devices would make it a lot easier than having to design the hardware from scratch. It was concluded by having two different Heart Rate and SpO_2 kits tested.

- **Use of commercially available components for measurement in marine animals**

  This objective was concluded using the Heart Rate Click board and an STM32 Microcontroller and programming it with an application software using C language. Using humans as test subjects this objective allowed to refine the filters used for digital filtering and the algorithms for heart rate and SpO_2 calculation.

- **Hardware development for oximetry and heartbeat sensor prototype**

  After the results from the first objective are deemed acceptable, the next stage can be set in motion, which involves the development of a PCB (Printed Circuit Board), including a mimicked version of the Heart Rate Click Board hardware, an integrated microcontroller, voltage regulators and connectors to allow a battery to be fitted.

- **Software development for prototype oximetry and heartbeat sensor**

  Aligned with the last objective, since the microcontroller of the prototype is slightly different from the one used for the early tests, there are some software changes that need to be addressed. Not only for this reason, but also because the prototype needs to be able to store/transmit the data acquired from the test animals, to be interpreted.

- **Sensor Miniaturization**

  Once the previous design is tried and tested, the PCB size must be reduced. This is made considering that the final device should be as small as it can be so that the impact on the animal lifestyle is minimal.

- **Development of a functional prototype to be tested in vivo**

  The in vivo tests consist of placing the prototype against the test animal's skin in a controlled environment, so the prototype should already be small and encapsulated in a waterproof case. The LED's and photodiode will also be placed in such a way that there is little to no distance between them and the animal's skin. The purpose of this objective is to take all this into consideration so that the device's measurement errors are minimized and to take conclusions on how to place it in using a coupling device.

- **Testing in a real environment**

  Lastly, the prototype will be paired with the battery and a coupling device, so that it is able to be tested on an animal in a real environment. This objective is the culmination of all the previous ones and will represent the conclusion of the final device.

1.4 Document Structure

This point acts as a guiding thread through the entire document, to allow the reader to understand better the thought process behind the structure of this dissertation.

Following this chapter's brief introduction, a literature review of the state of the art was made, where the main technique used throughout this project is explained, as well as the device used to apply it, the different ways of applying it, what is available commercially and some of the limitations intrinsic to this technology.

From there, the third chapter begins, and it is where the first implementation of a device using a pulse oximetry sensor was made, as well as the description of another sensor that was used for testing purposes. Chapter four refers to the implementation of a complete pulse oximeter with an embedded microcontroller unit and all the steps involved in this process. Chapter five is solely dedicated to all the tests made, in humans and in marine animals, to attempt in proving that this technology can indeed be applied in marine animals for monitoring purposes. Lastly, the final chapter presents the conclusions taken from the entire project and some of the future work that can be made to improve the project and to make it easier for someone who decides to pick up the work where it was left.

2. State of the Art

In this chapter, some aspects of the technology used are going to be discussed in-depth, starting with the method used to detect the heart rate and estimate the SpO_2. Then the device itself, and some of its iterations with different principles of functioning. Lastly, some of the considerations that are needed for projects involving wild marine animals.

2.1 Pulse Oximetry

Pulse oximetry is the non-invasive optical technique that is more commonly used to determine the heart rate and blood oxygen saturation of a specimen. This technology exists due to contributions coming from as far as the 1700s, with Johann Lambert which formulated a law stating that the absorbance of a material sample is directly proportional to its thickness. After that in the 1800s, August Beer added that the absorbance is proportional to the concentrations of the attenuating species in the material samples. The combination of these ideas became known as the Beer-Lambert Law and is considered the scientific cornerstone behind pulse oximetry. This law states that the intensity of light decreases exponentially while traversing through an absorptive medium. It is this diminishing in intensity that allows this technique to work, since this is what is seen in the photoplethysmogram (PPG) [4]. But a lot of other very important contributions were made before the technique became what it is today. It was only in 1972 that Takuo Aoyagi, a bioengineer, developed the first iteration of the method that is currently used [5][6].

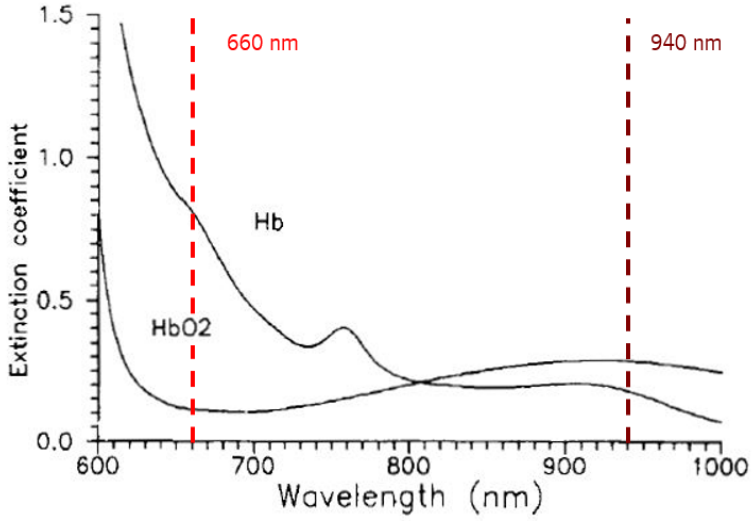


Figure 2.1 - Extinction Factor of Light Through Hemoglobin [9]

As it was said in the contextualization, the way this technique works is, a photodetector absorbs light energy, which can be used to monitor a patient's pulse rate and oximetry. It is a cheap optical process that can be used to perceive changes in the blood volume of the capillary beds of body tissues. The induced voltage of the photodetector changes along with the volume of blood flow in the vessels. This fluctuation from the heart pulses is used to calculate the heart rate [7]. Pulse oximetry estimates SaO_2 based on the light absorption properties of red blood cells. Hb changes its light absorption attributes when mixed with oxygen. Pulse oximetry exploits difference in light absorption between Hb and HbO_2. HbO_2 absorbs more red light (660 nm wavelength) and lesser infrared light (940 nm wavelength) than Hb as it can be seen in Figure 2.1 [3][8].

The PPG carries both an Alternate Current (AC) and a Direct Current (DC) component. The DC component is there because of reflective material that is always present such as skin, muscles, bones, and venous blood. The AC component will consist mostly of the light reflection of pulsatile arterial blood. This component is usually tainted due to motion artifacts since this test is very motion sensitive. It should be noted that only arteries carry an AC component, venous blood only contributes to the DC component of the signal. Figure 2.2 and Figure 2.3 tear down exactly what was just said.

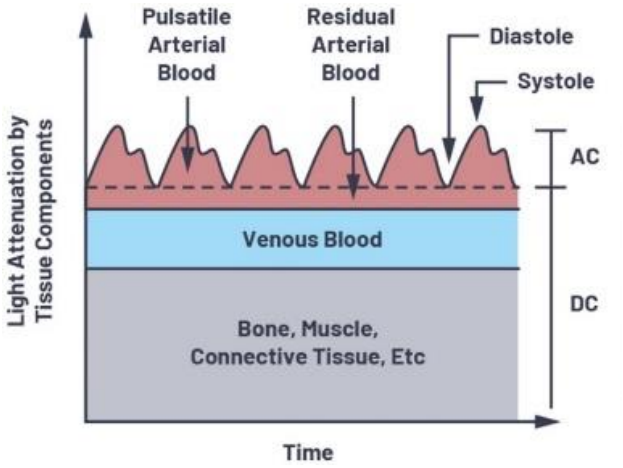


Figure 2.2 - Light intensity attenuation through tissue [4]

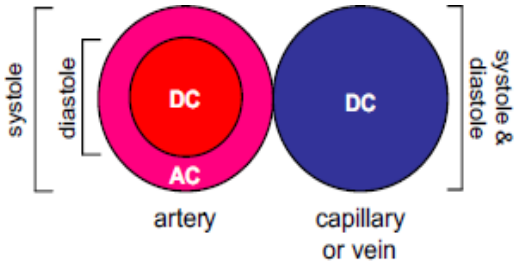


Figure 2.3 - Cross Section Comparison of Arteries and Capillary/Veins [10]

2.2 Pulse Oximeter

The most basic approximation of a pulse oximeter circuit is as seen below in Figure 2.4. This is by no means similar to the circuits that are inside the boards used throughout this dissertation, as they are more sophisticated, but it helps to get a better notion of how the system works.

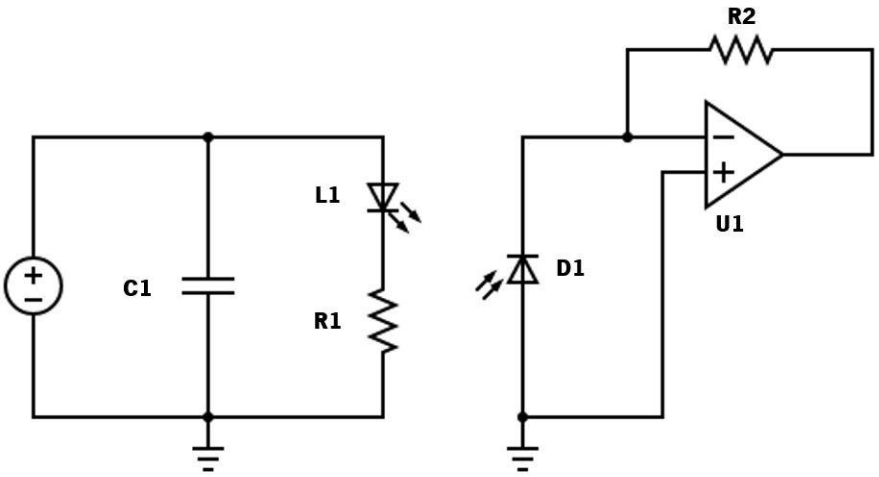


Figure 2.4 - Basic Pulse Oximeter Circuit

In a very crude way, the task of the circuit on the left is to power and drive the L1 LED, whose light is reflected in the tissues and received by the photodiode D1, which converts into a voltage that is amplified by U1. The output signal from U1 would be sent to a microcontroller to be processed and interpreted. Important to note that with this setup you can only measure heart rate, as to also estimate SpO_2, there must be a replication of this circuit with a second LED as both are needed to estimate SpO_2. The output signal from this type of circuit looks like the one on Figure 2.5. It is in this signal that it is seen the slow contraction followed by the quick dilation of the blood vessels, from the sudden increase in volume corresponding to the heartbeat of the specimen.

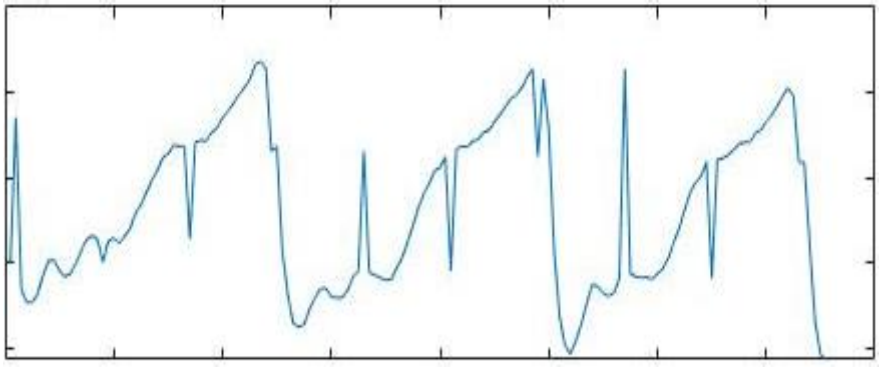


Figure 2.5 – Photoplethysmogram

Reflective vs Transmissive Pulse Oximetry

Picking up on what was said previously in the contextualization, there are two types of oximeters that make use of this technology. The reflective mode and transmissive mode pulse oximeters.

The general behaviour of them is the same, two LEDs of different wavelengths, one photodiode to detect light. They differ in the position of the LEDs relative to the photodiode and what exactly the photodiode detects. For the reflective mode, as the name states, the light pulses are directed toward the tissues, and the photodiode is placed near them, on the same side of the measuring site, and detects only what is reflected by them. For the transmissive mode, the measuring site is placed in between the LEDs and the photodiode, and it detects the amount of light that was able to pierce through the tissues. A simplistic image of these modes of operation can be seen in Figure 2.6.

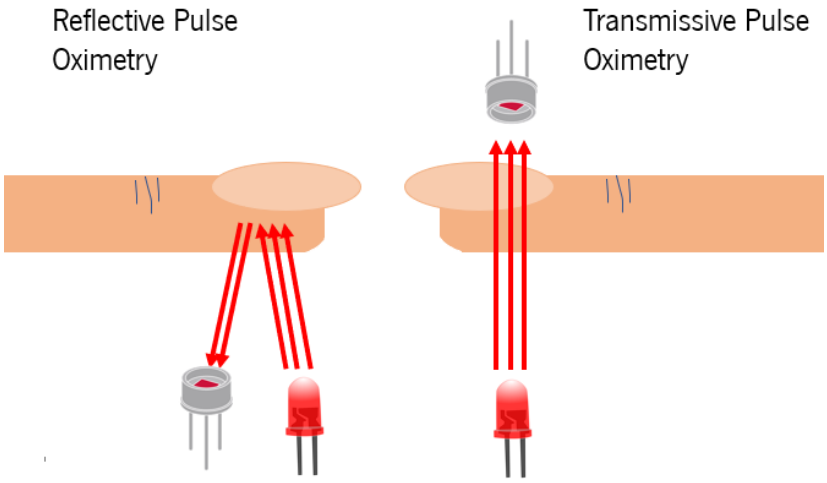


Figure 2.6 - Reflective Pulse Oximetry vs Transmissive Pulse Oximetry

Commercially Available Devices

Nowadays pulse oximeters have become very common, and there are a lot of devices that include them as one of their features. Not only for medical applications, but also in sports since it is a cheap option to keep track of heart rate and SpO_2 during effort. The most common uses for the reflective technology are present in devices like smart watches (Figure 2.7 on the left), where the LEDs pulse on the wrist, and give immediate readings. It is in these applications that are usually used green LEDs instead of red ones, because they have a lower tendency to induce errors due to motion artifacts, at the cost of lower penetrating power and sensitivity to darker skin colours. In most medical applications the transmissive technology is the current choice (Figure 2.7 on the right) because it isn't as limited in terms of results. It

has 10x more penetrating power and it is hardly affected by differences in skin colour and are usually used in a stationary position which makes them more reliable.

Figure 2.7 - Smart Watch with Reflective Pulse Oximeter (Left) and Transmissive Pulse Oximeter (Right)

Other Techniques for Pulse Oximetry/Oximetry Measuring

There is a technique known as CO-oximetry, which is also optical, and the principle of operation is pretty much the similar to regular pulse oximeters. The device used for this technique is the CO-oximeter (Figure 2.8), and it is more accurate because blood doesn't contain only Hb and HbO_2, it also contains abnormal or dysfunctional hemoglobin, the dyshemoglobins, like carboxyhemoglobin (COHb) and methemoglobin (MetHb). These abnormal types also absorb red and infrared light and therefore they affect SpO_2 measurements. For example, COHb is read by a usual pulse oximeter as HbO_2, because carboxyhemoglobin is scarlet red, this will wrongly elevate the SpO_2 levels. This device uses multiple wavelengths, so that it can detect them, and their quantitative contribution can then be weighed and removed to give more accurate readings [9].

Figure 2.8 - CO-oximeter

There is also a study about a method for SpO_2 estimation, that cannot be said to be a different technique than those that were previously mentioned, based on the technology, because it is still optical pulse oximetry, but interestingly, it diverges largely in the place of measurement. This method uses a fibre optic sensor, and the place of measurement is the bowel. This idea was tested due to the fact that reliable estimation of SpO_2 is dependant of the presence of good quality PPGs. Restricted arterial flow can result in small, unmeasurable PPGs which can be associated with Splanchnic ischemia (Restricted blood flow (and thus oxygen) to the bowel). But, while this is true for regular measurement sites, there are some abdominal organs, like the liver or the bowel, that in this situation, still show measurable PPGs. The device was developed and can be seen in Figure 2.9. The main difficulty associated with using this device is that it is hard to keep close to the surface of the organ under investigation [10].

Figure 2.9 - Reflectance fiberoptic splanchnic pulse oximetry sensor

2.3 Limitations

It is known that pulse oximeters have limitations, some have already been mentioned, but there are some more. In 1992, Severinghaus and Kelleher, provided a detailed list of these limitations, as seen below [11].

- Instrument incidence of failure
- Low signal-to-noise ratio (SNR)
- Light Deviation and poorly applied probes
- Vasoconstrictors
- Low perfusion limits
- Motion artifacts

- Abnormal Pulses
- Venous pulse interference
- Response times
- Error induced by ambient light
- Site selection for probe placement
- Skin pigments
- Dysfunctional hemoglobin
- Skin damages

Besides this list, there are some more limitations associated with conditions that can't be detected by pulse oximeters. Conditions like respiratory effort, hypercarbia (increase in carbon dioxide in the bloodstream) or disturbances in blood pH found in respiratory acidosis and alkalosis [12].

There are some more but weren't referred because of them being associated with pulse oximetry measuring in human patients, and that is not the focus point of this document. Since then, some of the limitations have already been mitigated with the evolution of the technology. Even so, the device is meant to be placed in wild animals, therefore, it is more than likely to encounter problems due to motion artifact derived errors, with the choice of place for the probe as to not interfere with the normal behaviour of the animal and also with skin damage, thick skin, etc. Knowing this, it is concluded that there will be things that need to be kept in mind, such as the choice of place for the probe, the pressure applied between the device and the skin of the animal, induced error from ambient light and the SNR should be optimized [3].

3. Analysis and Sensor Application

After researching the literature, the first step towards the development of a prototype was to choose a commercially available device to test, as it was said before. The chosen device was the MAX30100 Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health, which was included in the development kit Heart Rate Click from MIKROE. This device is very practical and relatively easy to use with the right microcontroller, and online support for it is not very hard to come by, even though it is only for usage on humans. But from testing on humans to testing on marine animals there were a lot of steps to be taken, such as the choice of microcontroller to be used, the software to interface between the sensor and the microcontroller, the waterproofing, etc and this is the chapter where they will be specified.

3.1 Requirements

Before addressing this chapter, some of the functional and non-functional requirements for the prototype to be developed are going to be mentioned.

Functional

- The developed device must be waterproof.
- Must last at least 72 hours while taking non-continuous measurements.
- Should consume as little power as possible, in the order of below 20 mA while measuring.
- Must be able to make the calculations to estimate heart rate and SpO_2.
- Must be able to communicate the data via UART.

Non-Functional

- It should be as cheap as possible.
- It must be as small as possible.
- It should cause the least amount of impact on the normal behaviour of any animal.

3.2 Heart Rate Click Board

This kit, also known as MIKROE-2000, uses a MAX30100 sensor which is a widely used Pulse Oximetry and Heart Rate sensor for students since it's easy to use with almost any microcontroller, like STM32s and Arduinos. The kit can be seen in Figure 3.1. It communicates with the microcontroller through I2C protocol, which is a reliable interface for this kind of application.

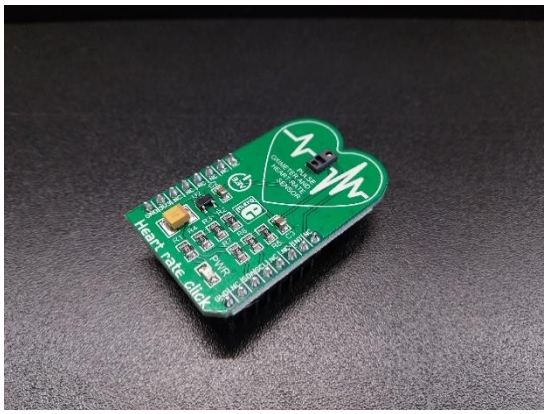


Figure 3.1 - Heart Rate Click Board

3.2.1 MAX30100

As it was said in the chapter introduction, the sensor featured in this board is the MAX30100. It is a complete pulse oximetry and heart rate sensor system designed with the demanding requirements of wearable devices in mind. The sensor has very helpful characteristics such as the Ambient Light Cancellation (ALC) module and discrete time filters ensure that the interference coming from ambient light or 50/60 Hz noise, is minimized. Another very important feature is its Standby mode which allows for very low power consumption. In Figure 3.2 it is shown the functional diagram of the sensor, taken from its datasheet [13].

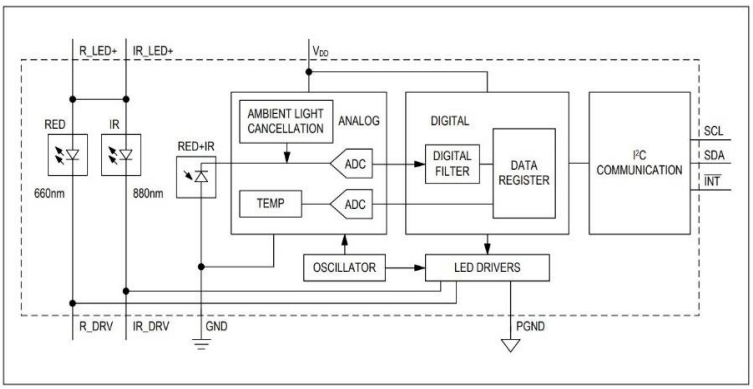


Figure 3.2 - MAX30100 Functional Diagram

The sensor communicates via serial I2C protocol, is configurable via software registers, and the output data is stored in a 16-deep FIFO in the device. The SpO_2 subsystem consists of three components, the ALC module, a 16-bit sigma delta analog to digital converter (ADC) and a discrete time filter. The sample rate of the ADC can be set from 50 Hz to 1 kHz. The time filter is responsible for filtering the 50/60 Hz interference and low-frequency residual ambient noise. The on-chip temperature sensor is included for (optionally) calibrating the temperature dependence of the SpO_2 subsystem. Lastly, the MAX30100 has individual LED drivers to drive the red and infrared LED pulses for SpO_2 and heart rate measurements. The LED current can be programmed from 0 mA to 50 mA and the pulse width can be set from 200 µs to 1.6 ms in order to enhance measurement accuracy and power consumption based on the application.

3.2.2 Board Schematic

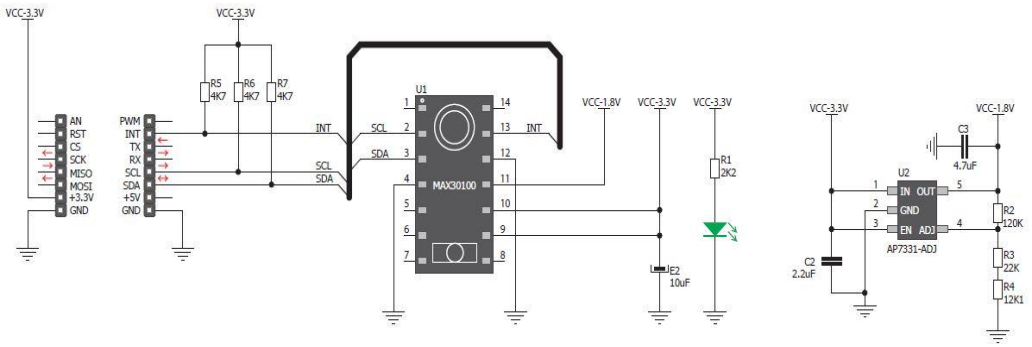


Figure 3.3 - Heart Rate Click Schematic

Since this sensor doesn't require a lot of external components to function, the complete board schematic, seen in Figure 3.3 is fairly simple. It includes three 4.7 kΩ pull-up resistors for the Interrupt (INT), Serial Data (SDA) and Serial Clock (SCL) lines, a 300 mA adjustable linear voltage regulator to drop the 3.3 V input voltages to 1.8 V, since the sensor uses 1.8 V to operate and 3.3 V to power the LEDs. The remaining components are to reduce the noise from the power supply, to adjust the output of the regulator to the desired output voltage and a resistor in series with an LED to show "Power On".

3.3 STM32F767ZI MCU

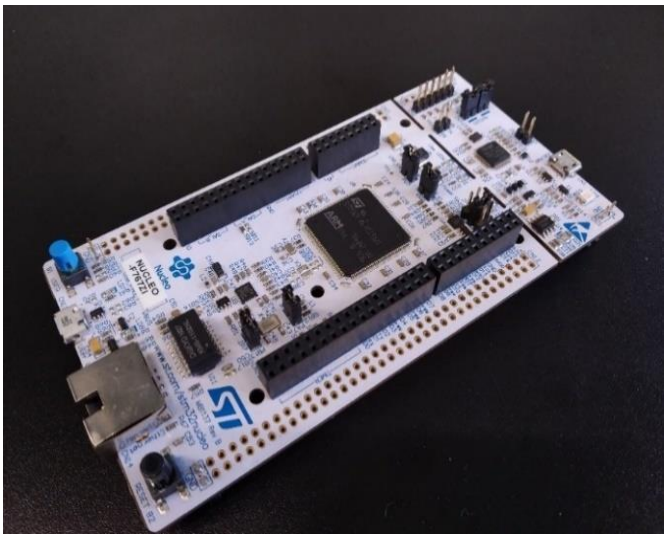


Figure 3.4 - STM32F767ZIT6 Nucleo 144

During this early stage of investigation, testing and development, the device chosen for the task of receiving and processing the sensor data was the STM32F767ZIT6 Nucleo 144 development board. This board contains the STM32F767ZI MCU, which is a 20 x 20 mm chip with a 216 MHz Arm® Cortex®-M7 core, 2Mbytes Flash and 512 Kbytes Static Random Access Memory (SRAM) [14]. The development board can be seen in Figure 3.4.

With these characteristics, it is capable of receiving and processing the data quickly even when the processing consists of applying complex filters and making calculations with the resulting signal.

This board is used mainly for developing purposes, but in a future PCB implementation, a smaller, low power version of this microcontroller will be used.

The connection diagram between sensor, microcontroller and computer can be seen in Figure 3.5 and in Figure 3.6 the real setup. As it was said before, the communication between sensor and microcontroller happens via I2C and between microcontroller and computer via UART. Also, a breadboard was used to assist in the connection of the board to the microcontroller during the earlier stages of development.

The power lines, 3.3 V and GND were connected to the respective 3.3 V and GND from the STM board. The INT, SCL and SDA lines were connected respectively to the PB0, PB6 and PB9 pins.

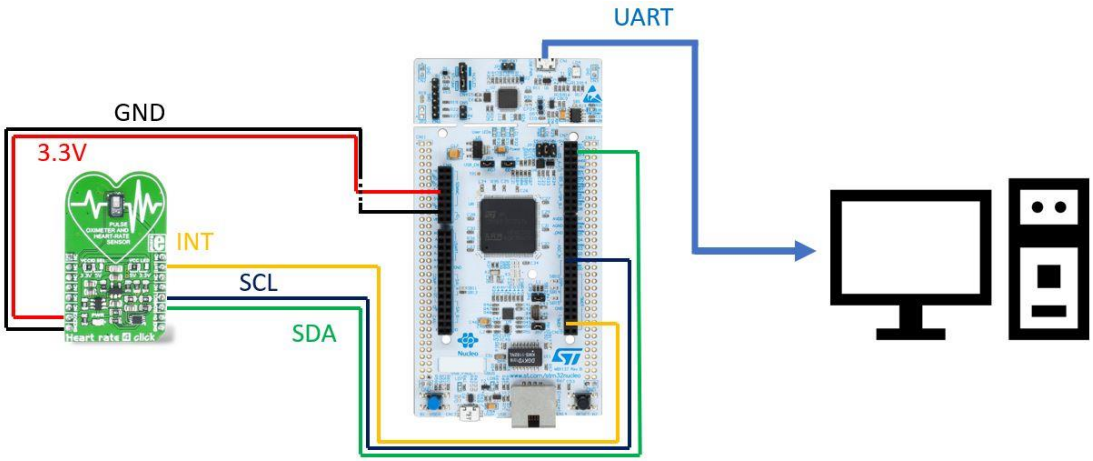


Figure 3.5 - Connection Diagram

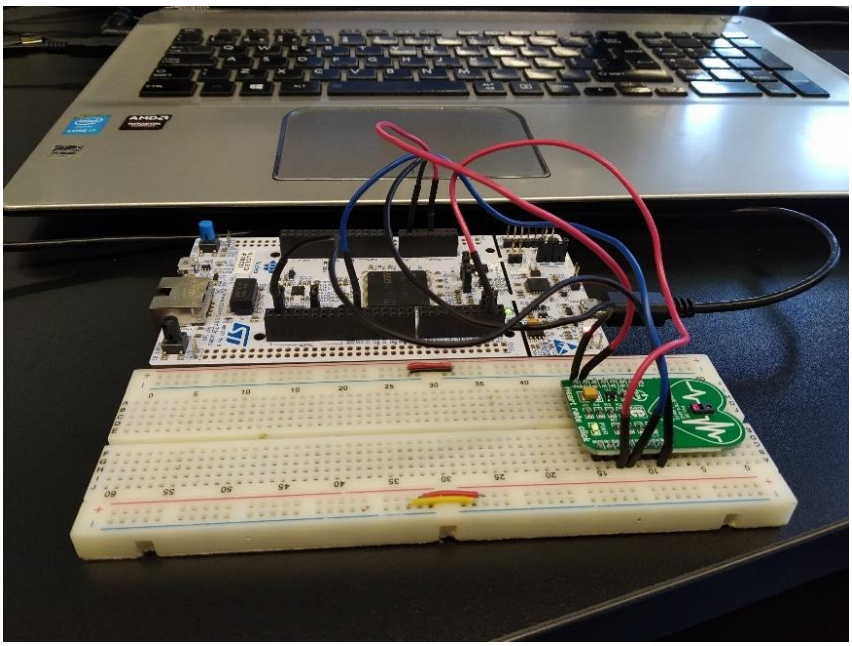


Figure 3.6 - Sensor, microcontroller, and computer setup for testing purposes

3.4 Software

Software-wise, the approach is pretty straightforward. The computer language chosen for this project was C/Keil C and the first step was to establish the connection sensor-MCU and MCU-computer so that the data can be sent and received for processing and visualization. The unprocessed samples can be analysed and used to make the heart rate and SpO_2 calculations manually using the computer, but they are very noisy, thus making it impossible for the MCU to use them without first processing them. This brings us to the second step, where the raw samples go through digital filters to remove noise and the DC component of the signal. The last step is the development of functions to compute the heart rate and SpO_2 effectively.

Important to note that to make the task of optimizing the filters easier, all the digital filtering and calculations were first made using MATLAB since this is a very user friendly tool when it comes to digital processing. But, to complete this, an Open Source library for the MAX30100 sensor in C was used, as it was still needed to transmit the data to the computer via UART [15]. There was no need for software development in this stage as the library already had all the functions required. Once the results were deemed satisfying, the next stage could take place, which was the implementation of similar filters and making those same calculations using the microcontroller.

Lastly, for the sake of consistency some parameters that were selected for the sensor will always be the same through the entirety of the project. These parameters are the sensors sample rate which will be kept at 50 Hz, for power saving, and the LED pulse width will be 1600 µs, that corresponds to an 8% duty cycle.

3.4.1 Filtering

MATLAB was the software chosen for the first implementation of the digital filtering, because it has all the tools required to design digital filters easily. It has the Filter Designer which is a very useful and intuitive tool when it comes to designing and analysing filters. It allows the user to set all the parameters to match the specifications required for the project. Having said this, it makes sense to use this tool to design and optimize the filters, before heading towards their implementation in C language.

To better understand what needs to be made, the first step is to look at the raw data.

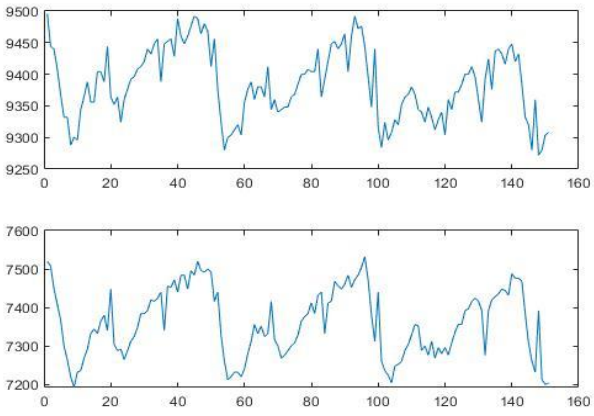


Figure 3.7 - Raw Red Samples (Top) and Raw Infrared Samples (Bottom)

Figure 3.7 is taken from the MATLAB plotting tool where the signal was captured with the 50 Hz sample frequency, which means that each sample has a 20 ms period. At first glance it can be seen that there is a lot of high frequency noise that must be mitigated. Because of that, a Low Pass Filter (LPF) was

implemented. The cut off frequency of the LPF had to be higher than 1 Hz, so as to not lose the heartbeat oscillation (1 Hz corresponds to 60 beats per minute (BPM)), and lower than 5 Hz, to filter most of the noise and keep the fundamental frequency of heartrate. After some trial and error, between the response time and the noise cancellation, the conclusion was that the most efficient frequency was 3 Hz, instead of 5 Hz.

Besides that, there was still the need to remove the DC component to make the process of acquiring the heart rate easier and for the SpO_2 calculations. The DC component is the 0 Hz frequency element of a signal, so, to remove it, it is also applied a High Pass Filter with a cut off frequency very close to 0. The closer it is to 0, the more effective the filter is but it also becomes very slow to stabilize, the further away from 0, it's the opposite, it is quite fast to stabilize but also less effective. After comparing results, a 0.3 Hz cut off frequency was chosen. Lastly, the signal was inverted because that makes it easier for the software to acknowledge the heartbeat peaks.

The magnitude response of the implemented filters can be seen in Figure 3.8.

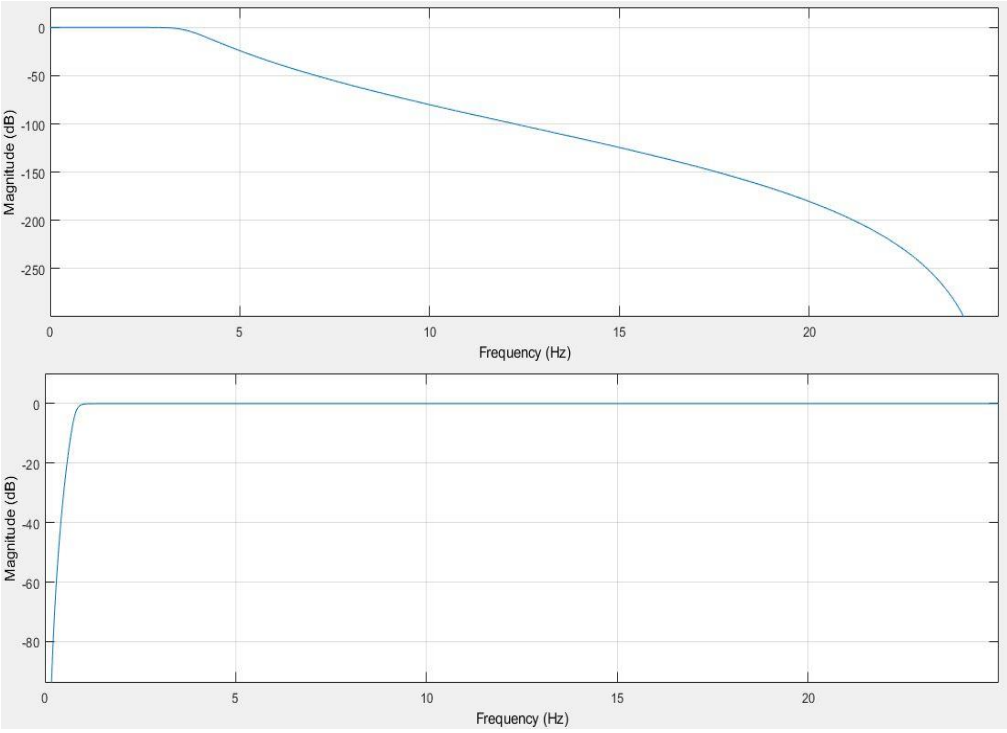


Figure 3.8 - Top: Magnitude response of the Low Pass Filter; Bottom: Magnitude response of the High Pass Filter

The results from applying these filters are shown in Figure 3.9 (Red and infrared samples not taken at the same time).The results are as expected, after applying the LPF, most of the noise is cancelled, and after going through the HPF, the DC component is removed, leaving a clean signal with distinct peaks that can now be used to calculate the heart rate and SpO_2.

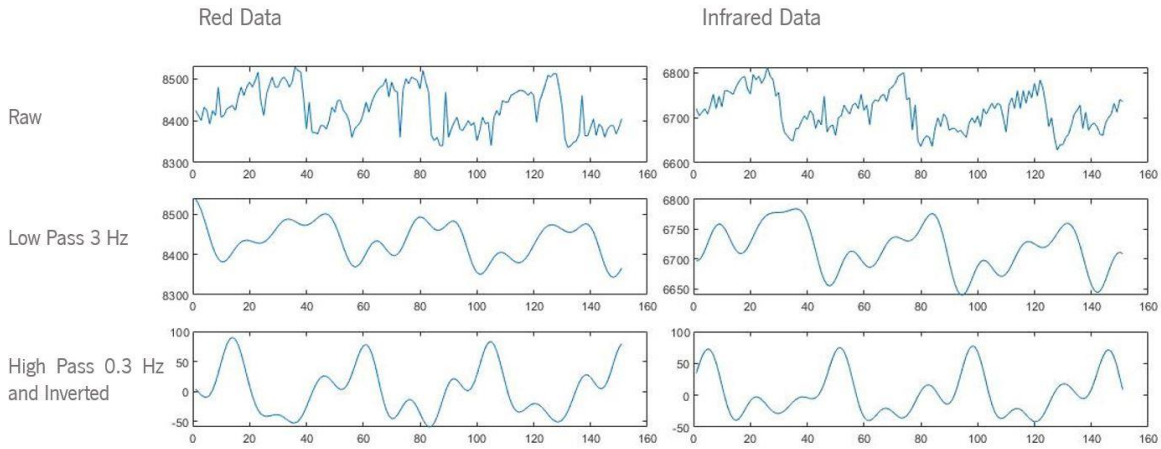


Figure 3.9 - Filter Results in MATLAB

3.4.2 C/Keil C Implementation

Having attained the desirable results on MATLAB, the implementation in C can be tackled. Aside from some functions in the library that were used, it was still required to code the most important ones, since the library didn't feature any signal processing, heart rate or SpO_2 calculator functions.

Research was made, because, despite finding support to some extent, most of what was found was not reliable enough. So, in the end, what mostly happened was an adaptation, and mix of different approaches, until all implementations were considered to be good enough.

3.4.2.1 Algorithm Flow Charts

Flow charts of the three main components of the code were made to help in the understanding of the thought process behind the developed software. These correspond to the main code, the interrupt routine and the data processing.

Main code

The main code begins with the configurations of the peripherals of the microcontroller, followed by the configurations of the MAX30100 sensor. The microcontroller configurations were made using STM32CUBEmx software, and the sensor configurations were made with the MAX30100 C library. After that, the UART is activated and waits for a character (char) to be received. After the char has been received, the program waits for the external interrupt from the sensor, as it is the indication that the FIFO of the sensor is full, and the data can be read. From there, we enter the Interrupt Routine where the data is collected and then there is a flag that signals that all the data has been collected and is ready to be processed, so if the value of the flag is '1', the program enters the stage of data processing (Figure 3.10).

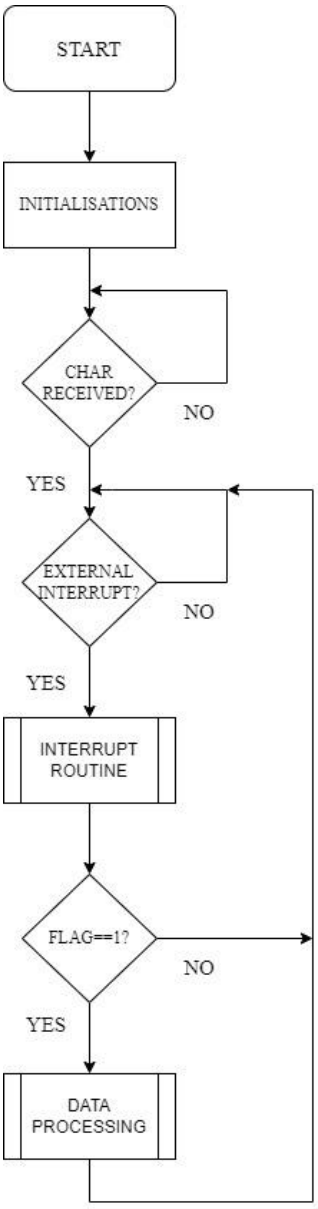


Figure 3.10 - Main Code Flow Chart

Interrupt Routine

Has said above, in this routine the data is collected, 16 samples at a time. But in terms of processing, 16 is a very low number of samples, so, while the value of the flag is '0', the program keeps entering a function that gathers a predetermined number of samples. This number of samples is defined by the user based on the desired application. Once the function finished collecting, the value of the flag is set to '1' so that the program knows that it can enter the final stage, the data processing (Figure 3.11).

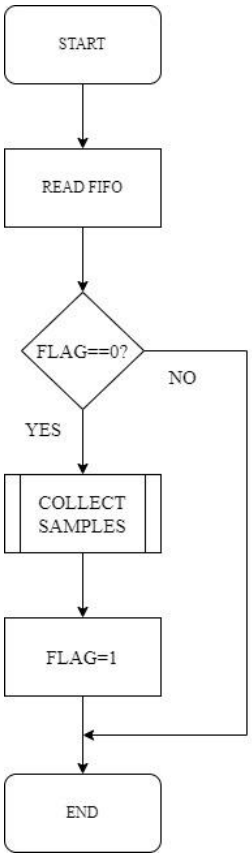


Figure 3.11 - Interrupt Routine Flow Chart

Data Processing

Following the flow, the first thing that happens in this component is a test of the flag value. If it's '0', not enough samples have been saved, so the program has to wait. If it's '1', the data is ready for the processing. The processing consists of applying the low and high pass filters to both the red and infrared signals. After that, the low pass filtered infrared data is used in the "FindPeaks" function, whose duty is to find the location of the peaks in the signal and save them in an array. This array is then passed to another routine, that uses the locations found to calculate the period of the signal and use that to calculate the heart rate. After that, the filter data is used again, now to estimate the SpO_2. In the end, the flag value is set at '0' so the program knows that it can start collecting samples again (Figure 3.12).

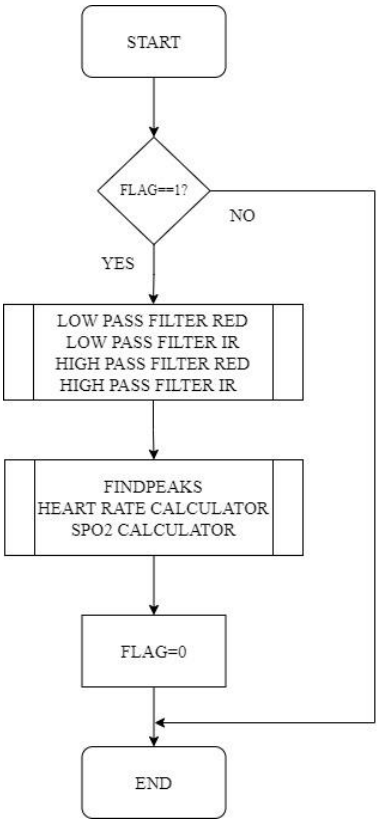


Figure 3.12 - Data Processing Flow Chart

3.4.2.2 Signal Processing

Using what was learned the MATLAB phase of the project, the execution of the signal processing in C began with the development of a 3 Hz LPF and a 0.3 Hz HPF. After searching, and applying different types of approaches on filters, all with less favourable results, a website was recommended that had filter implementations that matched the needs for the project and that could, with some adaptation, function in this application [16]. The filters chosen for the application were a Butterworth LPF and a Butterworth HPF.

The implementation was not in conformity with the methodology that was being used in the project so the first thing that was done was to save processing power, and so, the calculation of coefficients was made into a function that runs before the "while(1)" loop, so that after the boot, they only need to be calculated once. Then the filtering was split into two different functions, one that receives every sample, and feeds them to the other, which filters them one by one and returns the filtered sample.

The process begins when the program initiates, with calculating the constants used in the filters based on the desired sample frequency, order and cut off frequency. Four filters are used, but because the same LPF and HPF are used for the red and infrared signals, the coefficients are duplicated. So, it only makes sense to use the LPF coefficients in both LPFs and the same for the HPFs. Then, once the

"CollectSamples" routine is completed, the raw red and infrared samples are fed as an input to the LPFs. The output of this is fed to the HPFs. These actions can be visualized in Figure 3.13.

Figure 3.13 - Signal Processing Block Diagram

The magnitude response of the implemented filters can be seen in Figure 3.14 and the resulting data from applying them can be analysed in Figure 3.15. Taking into consideration Figure 3.7 representing the raw data, as can be seen, the LPFs do their job of removing most of the noise and leaving two fairly clean signals. The HPFs receive these signals, remove the DC component, and invert them. Overall, the results are as expected, and the outputs can now be used in the next step.

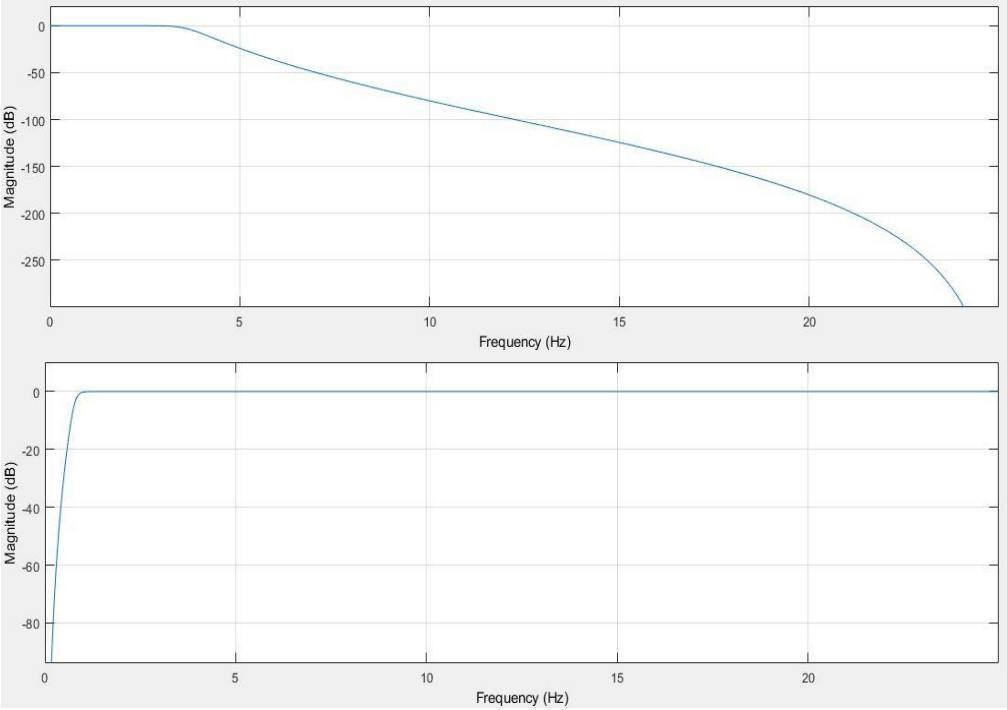


Figure 3.14 - Top: Magnitude response of the Low Pass Filter Implemented in C language; Bottom: Magnitude response of the High Pass Filter Implemented in C language

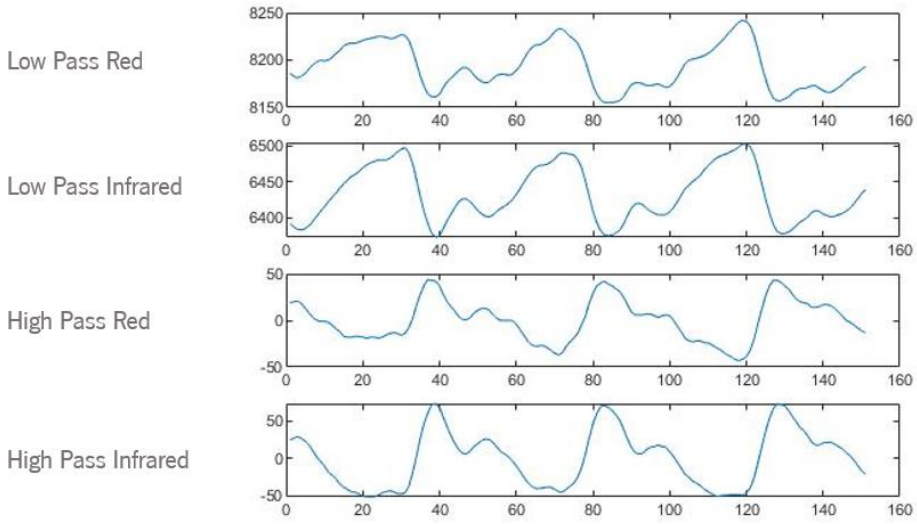


Figure 3.15 - Digital Filtering Results

3.4.2.3 Heart Rate Calculator

The heart rate calculation is divided in two main parts. The first part being the "FindPeaks" routine, to detect the peaks present in the signal and the second the heart rate calculator. The approach for finding peaks in the signal was changed from finding the actual peaks from the PPG, to finding the peaks where the highest gradient is found. This made this task a lot easier, and the approach will be explained below.

For the "FindPeaks" routine, the implementation began with a simple question: "What is a peak?" To which the answer is rather simple, "The peak of a signal is the point that has a higher value than both the previous and the next point". Putting it this way, it seems basic, but in a signal that usually has some residual noise and undesirable peaks, this statement is recurrent. An example of this is seen in Figure 3.16. In green are the peaks that should be considered for the heart rate calculation, and in red are examples of the ones that shouldn't.

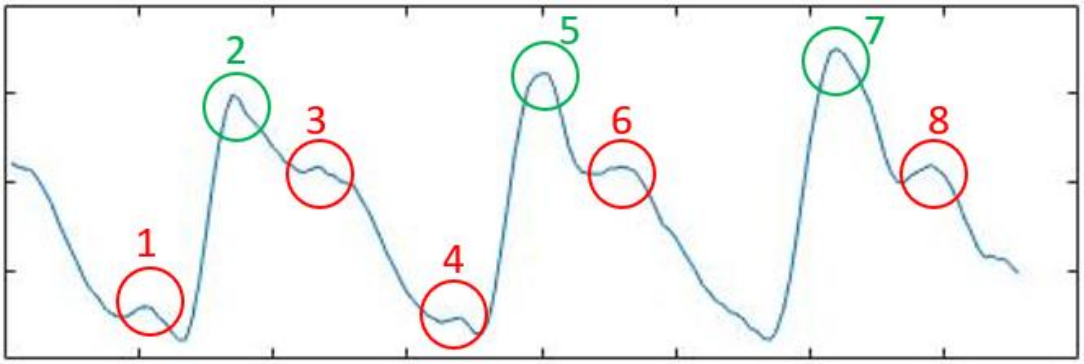


Figure 3.16 - Green: Desirable Peaks; Red: Undesirable Peaks

Looking at the figure above, it is noticeable that some conditions can be implemented to make the MCU acknowledge the right peaks. The first condition implemented was, any peak that is below the average

absolute value of the signal is discarded, this solves peaks like 1 and 4. But this was not enough, as most of the times peaks similar to 3,6 and 8 were selected. But, if one notices that before and after every green peak is a steep gradient ramp, that should allow to develop the conditions necessary to find the right peaks most of the time. Since the gradient of a curve can be simplified into the difference in value between two points, that was the approach taken. An array of samples was collected and used to consistently verify the conditions. Only when the function returned the correct indexes of the peaks would these be deemed acceptable.

To better explain what happens, let's consider an array of N samples, "array []", a "for" loop that iterates the entire array searching for peaks and "i" is the current sample being tested. It was found in Excel that if the signal was reduced to the difference between the sample [i+1] and the sample [i-2], the peaks would become a lot more prominent, as it becomes an analysis of gradient. This can be seen in Figure 3.17 where the filtered samples and the gradient analysis are displayed in blue and orange respectively.

Figure 3.17 - Gradient Analysis

During testing, a lot of tweaking had to be made, as most of times, the function would acknowledge the same peaks more than once. In the end, three conditions were developed, and they worked fine enough.

- The absolute value from (array[i+1]-array[i-2]) must be higher than 1.4*average of the signal. This condition removes low value and small peaks.
- The absolute value from (array[i+2]-array[i-1]) must be higher than the absolute value from (array[i+1]-array[i-2]). This one made sure that the peaks being selected were the ones that had the highest peak prominence.
- The final one was, if this is not the first peak to be collected, the value of the index must be at least 10 units higher than the last one. This was implemented as extra protection for the instances where the program acknowledges the same peak twice.

This function outputs an array of structures containing the value and indexes of the peaks found, which is passed as a parameter to the heart rate calculator function.

The operation of this function is a bit simpler, as the tricky part is the selection of the right peaks. All it does is compute the heart rate formula using the indexes of the peaks, makes an average calculation of the time between them, in case the array contains more than two peaks. The formula used is the equation (3.1).

$$Bpm = \frac{60}{TPeak_2 - TPeak_1} \tag{3.1}$$

3.4.2.4 SpO_2 Calculator

As it was said before, the process to estimate SpO_2 begins with the expression (3.2).

$$SpO_2 = a^2R + bR + c \tag{3.2}$$

With R being given by the expression (3.3):

$$R = \frac{AC_{red}/DC_{red}}{AC_{ired}/DC_{ired}} \tag{3.3}$$

So, the first step is to calculate the ratio of absorbances R. For that, the AC and DC values have to be computed from the respective signals. The AC value is the Root Mean Square (RMS) value of the signal, and the DC value is the mean value of the signal. The equations used are as seen in (3.4)(3.5).

$$AC_{red/ired} = RMS_{red/ired} = \sqrt{\frac{1}{n}\sum_{i} x_i^2}$$

(3.4)

$$DC_{red/ired} = Mean_{red/ired} = \frac{1}{n}\sum_{i} x_i$$

(3.5)

With these values calculated, the function applies the formula and R is obtained. After that all that's left to be done would be to apply the SpO_2 formula but for that to be possible, first, the calibration coefficients a, b and c must be discovered. To reach reliable values, red and infrared samples were collected whilst taking heart rate and SpO_2 measurements using a commercially available device. The samples were then used to calculate the R value, which led to a quadratic expression with a, b and c being the unknowns. There is already some information on the internet about what values to use, and for this sensor, most websites advised the same values. Instead of engaging in a trial and error approach, those values were tested, and the results matched the readings from the device used for calibration.

In the end, with a=0, b=-25 and c=110, to estimate SpO_2 the equation (3.6) was used:

$$SpO_2 = 110 - 25 * R$$

(3.6)

3.5 Device Adapting for Tests in Controlled Environment

During the course of device development an opportunity emerged, to test the sensor in a marine animal. Before the testing could be done, it was obvious that some adapting had to be done, as placing the sensor without any kind of protection on an animal would probably damage it. So, it was decided that the sensor would be encased in resin. For this, a small open box was designed using the program Fusion 360 and 3D printed to act as a mould for the resin. As for measurements, the box is 43x58x17 mm. The design can be seen in Figure 3.18.

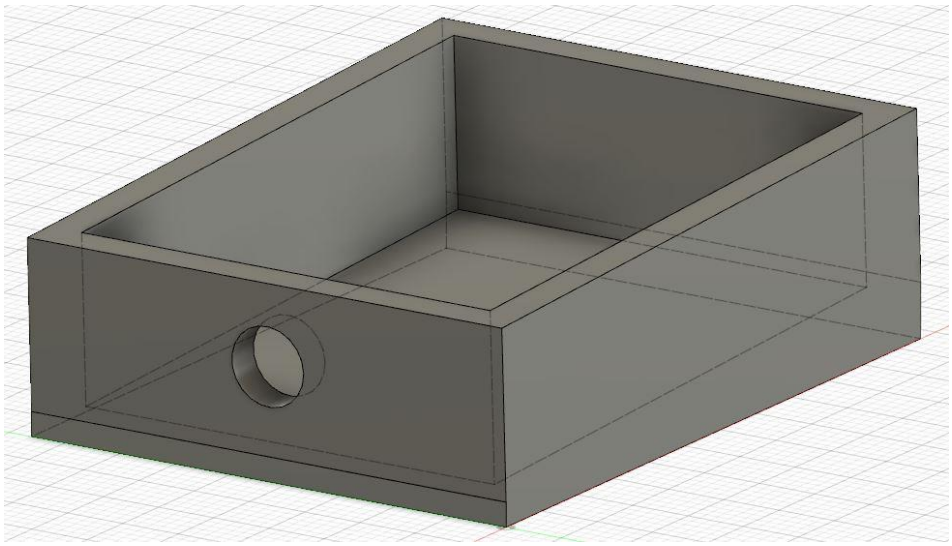


Figure 3.18 - 3D Casing Design

Besides that, the sensor was welded to a protoboard, along with longer wires so that the MCU could be placed further away from the wet animal. This device was then placed inside the box and the box was covered with aluminium tape because the resin is a thin liquid and could penetrate the mould and generate leaks. In Figure 3.19 can be seen the end result, after pouring the resin. Once it hardens, the resin turns from completely transparent, to opaque white and because of that, the optical sensor could not be covered.

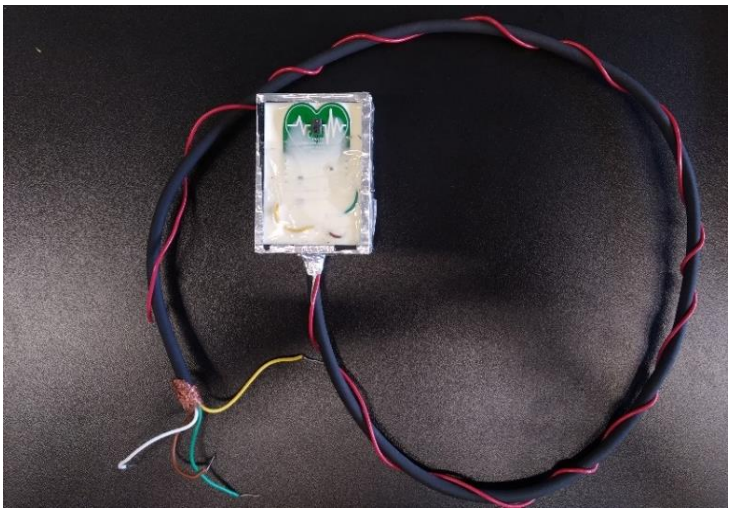


Figure 3.19 - Prototype for in vivo testing

3.6 MAX30110ACCEVKIT

To obtain a wider range of results, the MAX30110ACCEVKIT, from Maxim Integrated was also tested. This kit has three boards, MAX30110_UC_ EVKIT is the main data acquisition board, with the MCU while MAX30110_ SFH7050_EVKIT and MAX30110_OSB_EVKIT are the sensor daughter boards in which the

MAX30110 devices are placed in different optical configurations. With different optical configurations meaning that one board contains the MAX30110, two green LEDs and a TEMD5010X01 photoreceptor while the other contains the MAX30110 and the SFH7050 which is a chip with three emitters (red, green and infrared) and one receiver. These PCBs can be seen in Figure 3.20,Figure 3.21 and Figure 3.22 respectively.

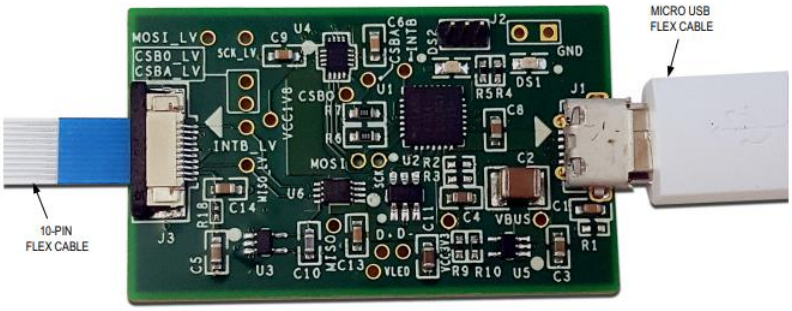


Figure 3.20 - MAX30110_UC_ EVKIT

Figure 3.21 - MAX30110_OSB_EVKIT

Figure 3.22 - MAX30110_SFH7050_EVKIT

When compared with using the MAX30100, using the MAX30110 is different because this device is only a front end analog device, it has no sensors, and needs to be coupled with one to work. But this also means that it is more specialized for the task, as it allows for higher resolution samples from its 19 bit ADC to support the lowest perfusion situations and higher sample rates of up to 3200 samples per

second, it has a built in front and back-end ALC which is better than the one found in the MAX30100. Basically, it allows to extract the maximum potential from the sensor that is paired with it and therefore is an excellent alternative to test in a situation where it is still not known if this technology works. For sensor configuration, display, and data storage, Maxim Integrated provides the MAX30110 Evaluation Kit software at no added cost. This software allows the automatic creation of a Logfile, with all the data that was measured with timestamps. The interface is seen in Figure 3.23

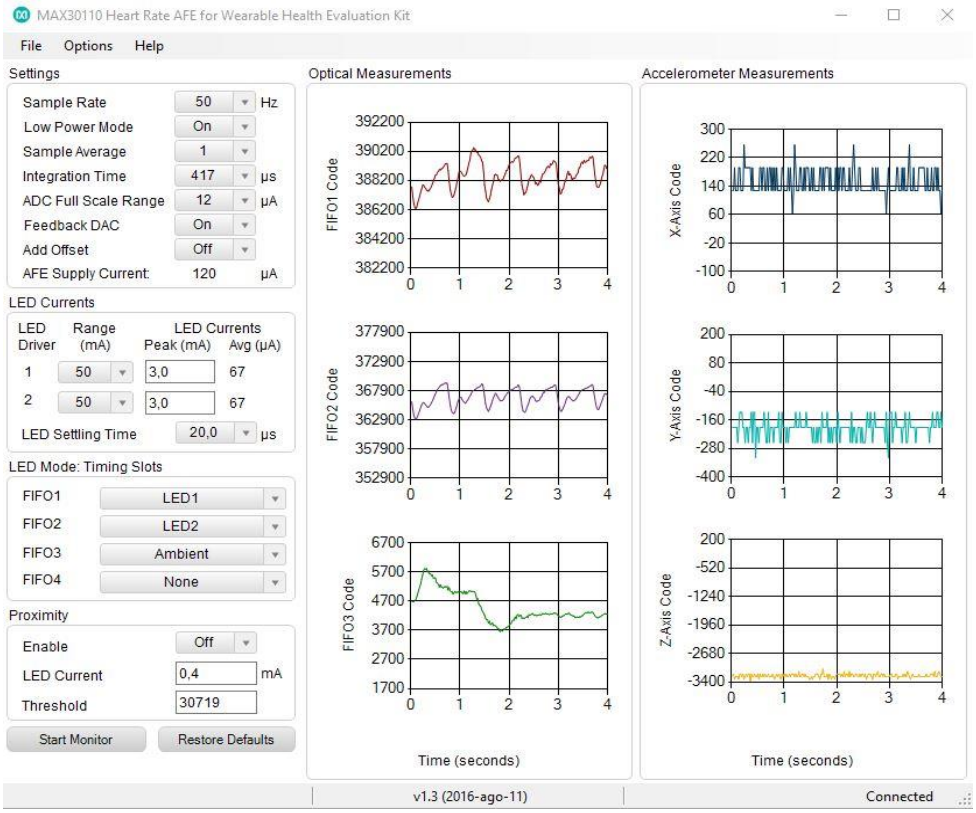


Figure 3.23 - MAX30110 Evaluation Kit Software Interface

3.7 Chapter Summary

In this chapter, the Heart Rate Click board was used along with the STM32F767ZI to learn more about pulse oximetry sensors and understand better how they are used, and how the information provided by them can be processed in order to achieve reliable SpO_2 and heart rate readings. This task was concluded, as the software developed worked as intended. From there, the device adaptation for in vivo testing also was completed successfully. Lastly the MAX30110 was also acquired in order to broaden the range and maximizing the chances of getting good results while testing in vivo, since this device allows for a more specialized approach than the MAX30100.

4. Implementation Results

After confirming that the technology aimed and developed for humans could work for marine animals without a lot of adaptations, the next stage was set in motion, which was the development of a miniaturized version of a device, with an embedded microcontroller. Basically, an all-in-one low-power prototype capable of being attached to an animal and without any peripherals, that is capable of collecting and processing the samples, make the SpO_2 and heart rate calculations and transmitting the results via serial port.

4.1 Low Power Implementation

The first step of this topic is to choose all the components required for the correct functioning of the prototype. A battery is going to be used to power the device, and it outputs 4 V when fully charged and drops to 3.1 V when discharged, so, a voltage regulator is necessary to keep the power supply from the battery constant. This powers the MCU, but the sensor requires a different voltage to work properly, 1.8 V being the typical value, so there's a need for another voltage regulator, in this case, one that drops the power from 3.1 V to 1.8 V. After this, the MCU needs a clock so, a low speed external crystal was chosen. From there, the remaining components are passive and because of that, were chosen for the most part based on their price.

As it is important for the prototype to work for as much time as possible, power consumption is a priority. When choosing hardware for this solution, three metrics were followed to improve battery life: active power consumption, standby power consumption and quiescent current (I_Q). In the end, only three components are actively consuming power, the MCU and the voltage regulators. For that, the reasoning behind their choices is discussed below.

MCU

The MCU used in the first stages of this project, the STM32F767ZI has an Arm® 32-bit Cortex®-M7 CPU, so it is very capable, but in terms of size and power consumption is by far not the best. So, an alternative had to be found and special attention was given to devices featuring an Arm® Cortex®-M0+, which is usually used for low-power and ultra-low-power applications.

Taking this into consideration, the MCU chosen for the application is a STMicroelectronics device, from their ultra-low-power line, the L051K8T6TR. This is a 7 mm x 7 mm chip with a 32 MHz Cortex®-M0+,

64 kB Flash, 8 kB SRAM, 2 kB EEPROM and 27 GPIOs [17]. Thanks to the small size and low power consumption (Table 4.1), this MCU is good choice for the proposed application.

Table 4.1 - STM32L051K8 Specifications

MCU	Core	Power Supply	Consumption	Size	Flash	Max. CPU Frequency
STM32 L051K8	Arm® Cortex® -M0+	1.65 V – 3.6 V	88 µA/MHz @ run 0.27 µA @ stop 0.4 µA @ standby	7 x 7 (mm2)	32 kB	32 MHz

Sensor

The choice of pulse oximetry and heart rate sensor was kept the same because the software developed was already developed specifically for the MAX30100 and changing it at this point would add difficulty and unnecessarily delay the development of the project. And since it has good power consumption, consuming an average of 600 µA, it is the best choice (Table 4.2) [13].

Table 4.2 - MAX30100 Specifications

Sensor	Power Supply	Consumption	Size	Interface	Low Power Modes
MAX30100	1.7 V – 2.0 V (Sensor) 3.1 V – 5 V (LEDs)	600 µA	5.6mm x 2.8mm	I^2C	Shutdown

Voltage Regulators

As it was said above, the main concern when choosing the voltage regulators was the quiescent current. This current can be defined as the amount of current used by an IC when in a quiescent state, which roughly translates to any period of time in which the IC is in either no load or non-switching condition, but still enabled. So, it is quite important in mobile applications where maximizing battery life is essential. Based on this, to regulate the voltage to 3.1 V it was chosen the TPS7A0531PDBVR which is a Low-Dropout regulator (LDO) with a very low, 1 µA I_Q [18]. The 1.8 V required to power the sensor were achieved using the ADP166AUJZ-1.8-R7 which is also a LDO with an 890 nA I_Q with 1 µA load [19].

Table 4.3 - Voltage Regulators Specifications

Voltage Regulator	Input Voltage	Output Voltage	Output Current	Quiescent Current
TPS7A0531PDBVR	1.4 V – 5.5 V	3.1 V	0 – 200 mA	540 nA w/ 0 µA Load 890 nA w/ 1 µA Load
ADP166AUJZ-1.8-R7	2.2 V – 5.5 V	1.8 V	0 – 150 mA	1 µA

The use of switching voltage regulators was also considered. However, due to small voltage drops expected in the application (4 V to 3.1 V in the primary regulator and 3.1 V to 1.8 V in the secondary regulator), the efficiency of switching regulators becomes similar to linear regulators, and therefore, the linear option overcomes the other in design simplicity.

Power Supply

The requirement of this project for battery life is that it must last at least 72 hours. With this in mind, to choose the battery, the total power consumption of the device has to be considered as it is what defines the capacity that the battery needs to have in order to last the amount of time defined.

Table 4.4 - PCB Component Power Consumption

Component	Active	Sleep
MAX30100	3.6 mA	730 µA
Microcontroller	8 mA	21 µA
Voltage Regulators	65 µA	1 µA

The theoretical total current consumption value can be estimated from the sum of the consumption from each individual active component as seen in Table 4.4.

This adds up to a little bit less than 12 mA active consumption and less than 1 mA sleep consumption. As this is only theoretical, it will be considered that the device consumes 15 mA to provide some margin of error.

$$72\ h * 15\ mA = 1080\ mAh$$

For the device to continuously take measurements for 72 hours straight, the battery must have a capacity of at least 1080 mAh. As 72 hours is the minimum, a larger battery will be chosen, for the device to have a larger battery life, in this case, the LG Lithium Ion INR18650 F1L 3350 mAh.

$$\frac{3350\ mAh}{15\ mA} = 223{,}33\ h$$

With this device and this consumption, the device can last about 223 hours on one charge which translates to roughly 9 days.

4.2 PCB Design

With all the active components chosen, the process of making the PCB to affix them in designated locations can begin. This process consists of two steps, with the first being the drawing of the circuit schematic and the second being the implementation of the PCB. These steps were made using the software Altium Designer.

4.2.1 Circuit Schematic

To better understand what was done, the approach will be to tear down the complete design in subcircuits and they will be explained individually.

Having said this, four main subcircuits can be identified. The MCU subcircuit, the sensor subcircuit and the two voltage regulators subcircuits. There are also some other components that are necessary such as resistors and decoupling/bypass capacitors.

Decoupling/Bypass Capacitors

The first component that will be mentioned will be one that is seen through both the MCU and sensor circuits, the decoupling or bypass capacitor. This tool is used because active devices are connected to their respective power supplies using traces that have finite resistance and inductance. There are situations in which the current drawn by them changes, causing the voltage drop between the power supply and the device to change accordingly, due to these impedances. If this happens with several

devices, there's a chance that the operation of one of them causes a voltage change large enough to affect the others. The decoupling capacitor offers a route for the transient currents, rather than letting them flow through the common impedance [20].

Having said this, five capacitors, one 1nF, two 10 nF and two 100nF were included in the schematic with the intention of being placed, one near the MCU, and the other four, two near both VDD inputs and two near the VDDA inputs (Figure 4.1).

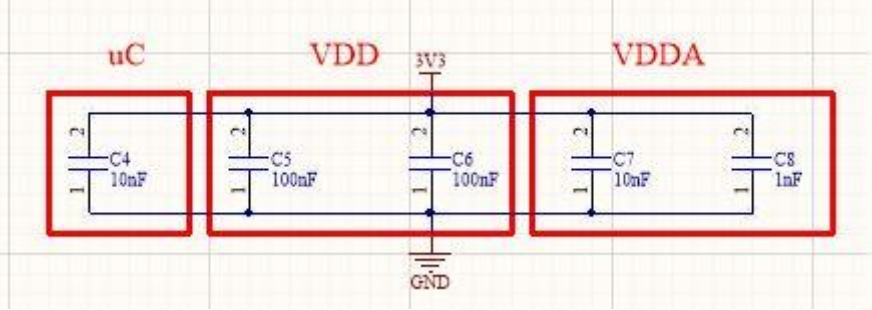


Figure 4.1 - Decoupling Capacitors

The remaining ones that were added will be mentioned below, in the text concerning the sensor subcircuit.

MCU Subcircuit

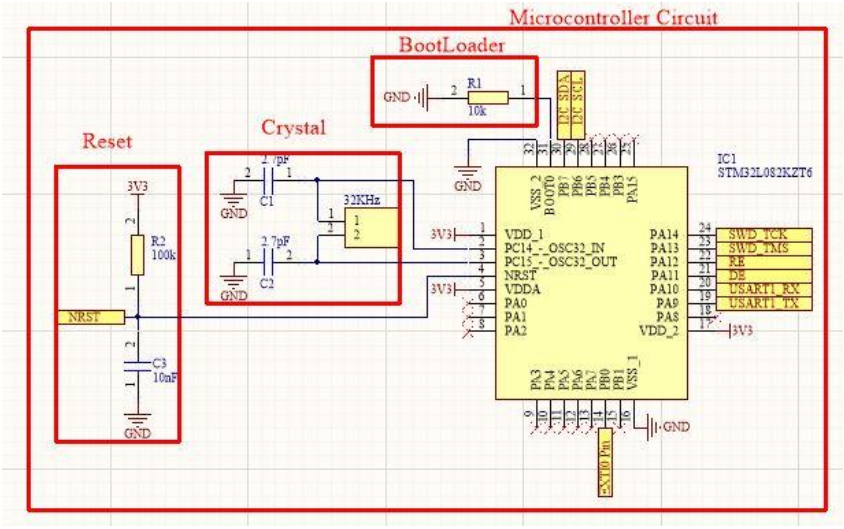


Figure 4.2 - MCU Subcircuit

The MCU Subcircuit (Figure 4.2) will essentially contain all the components used for it, and its desired peripherals, to operate correctly, in particular the Reset, the Bootloader and the clock input.

For the Reset are used a 100 kΩ pull-up resistor connected to the 3.3 V and a 10 nF decoupling capacitor connected to the ground. For the Bootloader a 10 kΩ resistor is used connected to ground. As a clock input, a 32.768 kHz crystal is used with both terminals coupled to capacitors connected to the ground.

Sensor Subcircuit

The MAX30100 sensor requires three pull up resistors for the I²C communication lines and for the Interrupt line, the value chosen was 47 kΩ. Besides that, a 10 µF and a 100 nF capacitors are used in the same way as the ones for the MCU, as decoupling capacitors, placed near the power inputs of both the LEDs and the sensor. All this is represented in Figure 4.3.

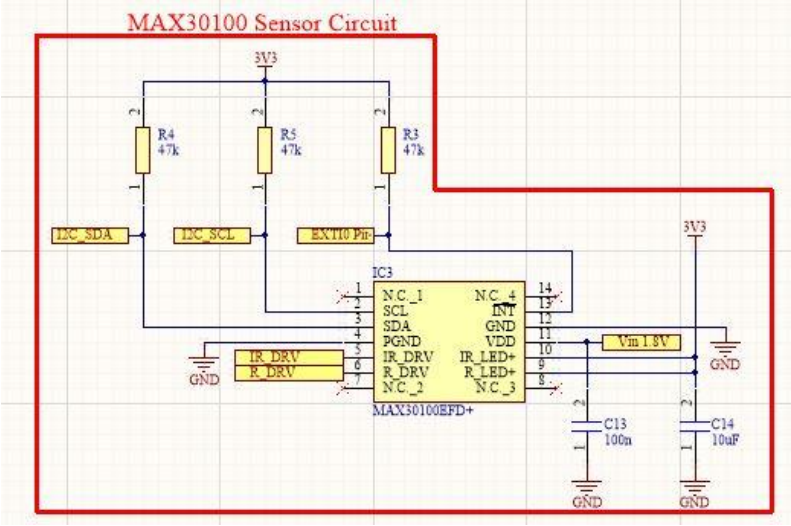


Figure 4.3 - Sensor Subcircuit

Voltage Regulators Subcircuits

Both voltage regulator circuits included additional components as advised in their respective datasheets, to ensure the correct functioning of the devices. For the TPS7A0531PDBVR (Figure 4.4), it is recommended to use an input capacitor to minimize transient currents drawn from the power supply and an output capacitor to greatly improve the load transient response. In the case of the ADP166AUJZ-1.8-R7 (Figure 4.5), the input capacitor ensures stability for the device and the output capacitor is included for the same reason as the other regulator, to improve the transient response from the load.

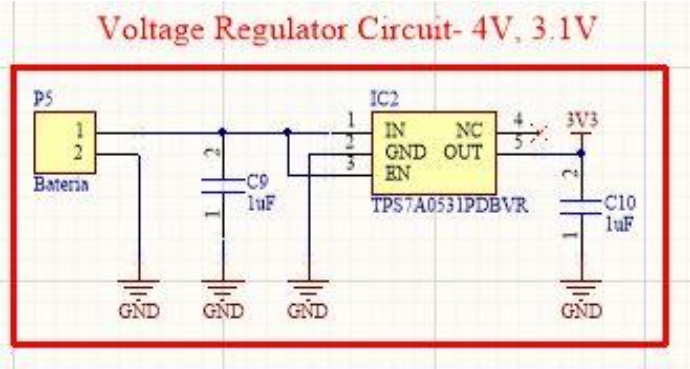


Figure 4.4 - 3.1 V Voltage Regulator Subcircuit

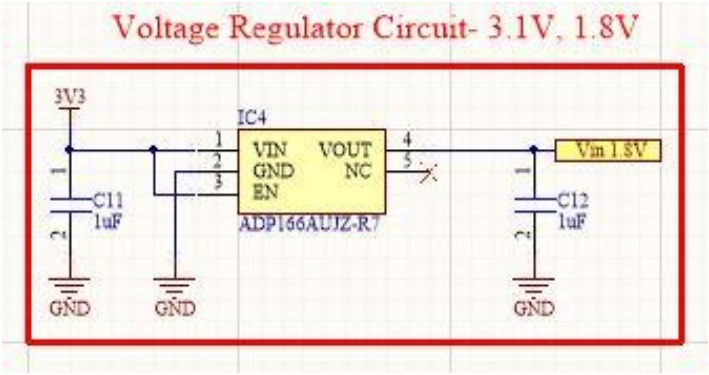


Figure 4.5 - 1.8 V Voltage Regulator Subcircuit

Ports

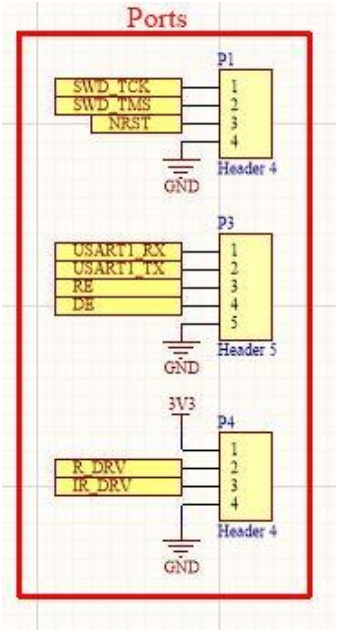


Figure 4.6 - Included Ports

In addition to all the components, ports were included in the design for the input of power, the programming of the MCU, the UART communications and, if necessary, exterior drivers for the sensor's

LEDs (Figure 4.6). They are very important, especially in first or early implementations of PCBs as they provide easy access not only to the points required for outward connections, but also for finding errors in the PCB.

Now that all the subcircuits were mentioned and explained, they can be seen all together in Figure 4.7 which shows the complete schematic. From there, the implementation of the PCB can begin as will be discussed in the next subchapter.

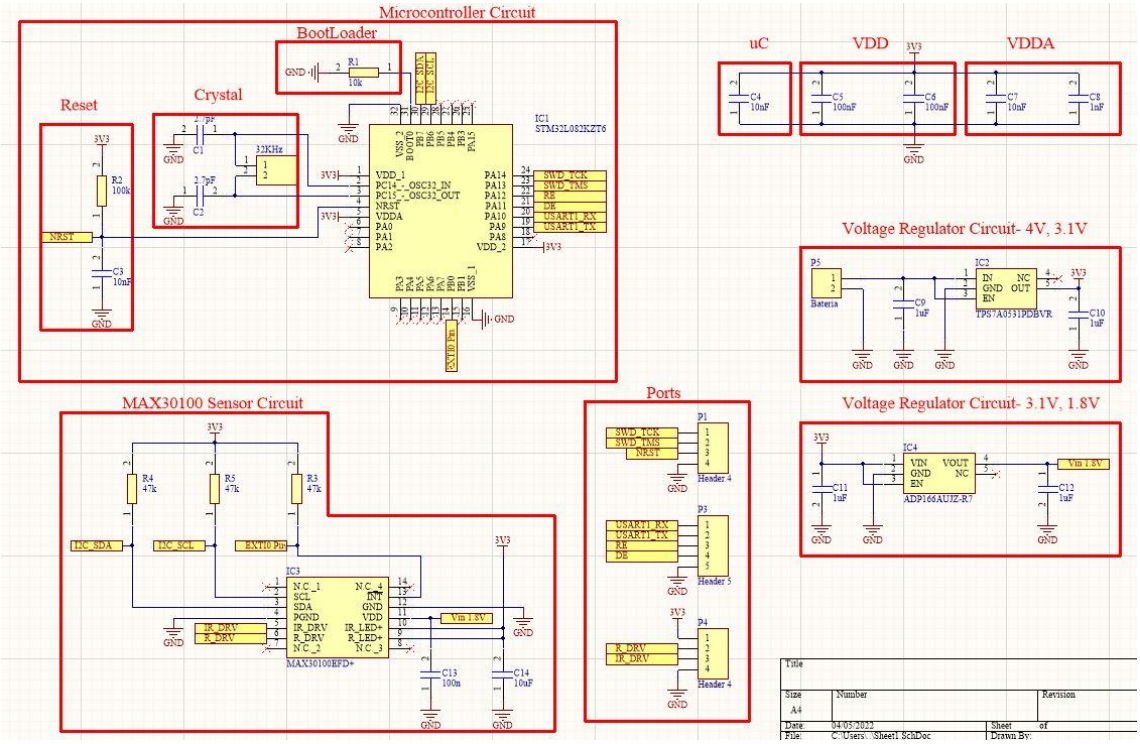


Figure 4.7 - Complete Circuit Schematic

4.2.2 PCB Implementation

For the design of the PCB, there were two main points to be kept in mind, as to increase the chances of developing a successful prototype:

- It should be as small as possible
- It will have to be waterproof

Knowing this, the components were arranged in a way so that the overall size would be as small as possible, while also bearing in mind that they will have to be hand welded. This is an important point, although not crucial, because surface mount devices (SMD) are already hard to weld, even while having some room to work with, so everything that does not need to be very close to other components, was given a little space. To help with the task of waterproofing, the main decision taken was that all the

components would be placed on one side of the PCB, with the exception of the sensor that would be placed on the opposite side. This allows for easy utilization of resin again, since even if some components are taller than the sensor, they are on the other side, and can be completely encased, posing no problems or risks whatsoever.

With all this in mind, the design was addressed almost in a similar fashion as the schematic, with the board being "divided" into subcircuits. These were then placed in strategic positions, in a balance between being near crucial points for that subcircuit like power inputs or the microcontroller and being placed in such a way so that the traces are as short as possible, because long traces can become antennas that generate unwanted noise. This is especially important in low power solutions.

Figure 4.8 represents in 2D the final layout of the components as they stand on the PCB. The red lines represent the top layer traces and the blue ones, the bottom layer traces. It may not be perceivable at first, but the components are grouped in sub-circuits throughout the layout (please note Figure 4.9).

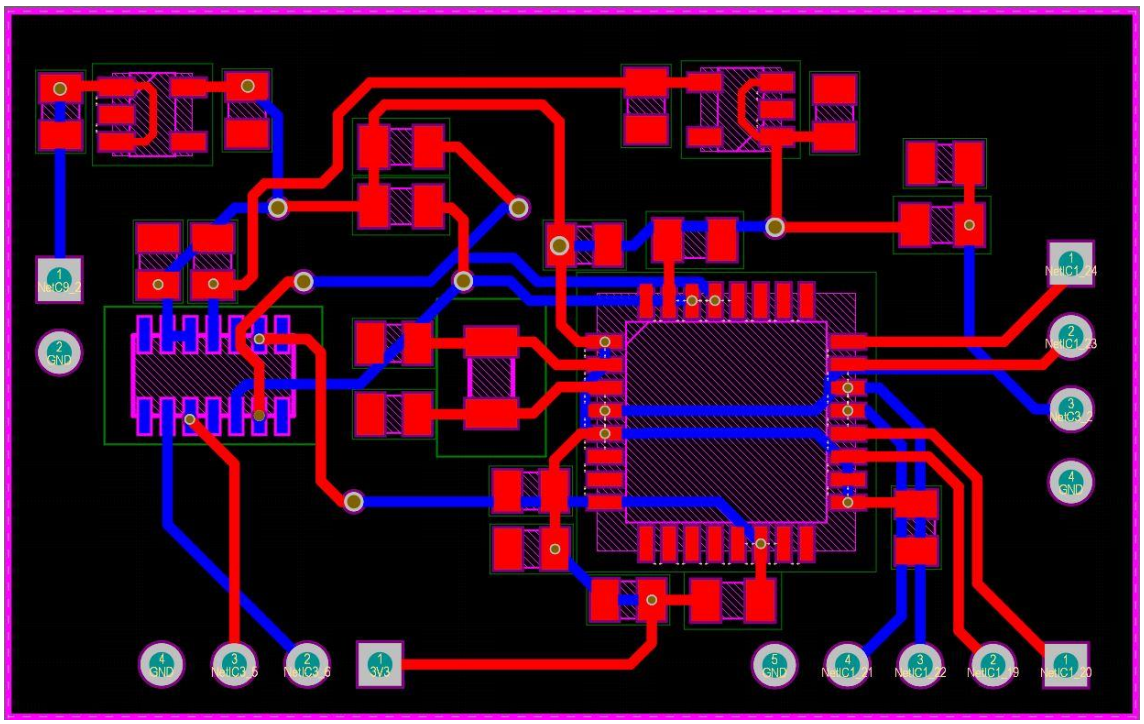


Figure 4.8 - 2D PCB Layout

- The green background areas numbered as “1” contain both voltage regulators circuits, and their associated components.
- The yellow background areas numbered as “2” contain the sensor circuit. Some components like the pull-up resistors don’t have to be placed near the sensor, so they were placed where it suited best.
- The blue background area numbered as “3” contains the MCU circuit with the crystal and all the remaining components.

The remaining squares and circles that are placed near the outward edge of the board are all the ports that were included and placed there since the edge makes the access to them easier.

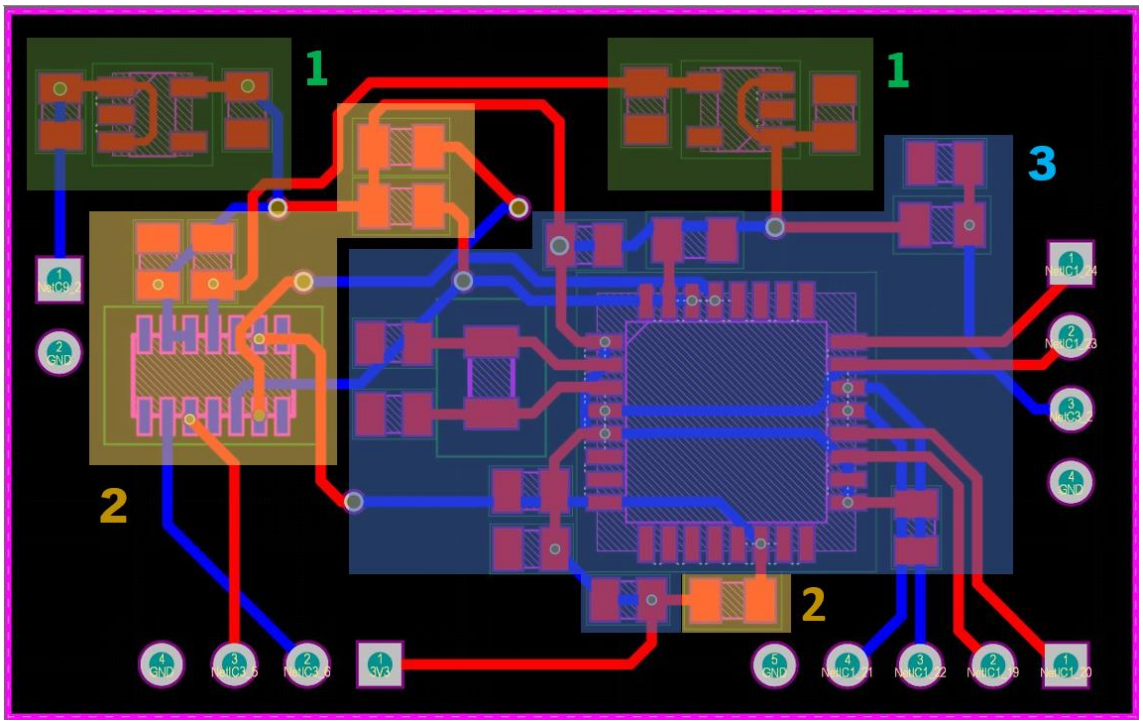


Figure 4.9 - PCB Sub-circuits

All that’s left is to send the PCB documents for printing and then weld all the components in their rightful places. The resulting top and bottom layers should look something like Figure 4.10.

The real PCB after printing is seen in Figure 4.11. The final size is 39 mm * 24 mm, so it is smaller than the entire Heart Rate Click Board, while also having an embedded microcontroller. This means that dimension wise, the miniaturization was a success.

It is ready for testing all the continuities and making sure that everything is in order and ready for welding. If everything checks out, the clean PCB is concluded to be in working order and ready for the next stage, the welding.

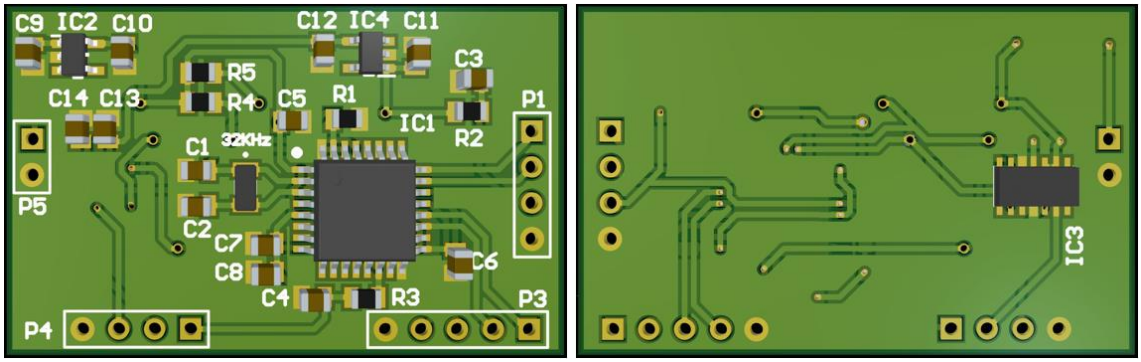


Figure 4.10 - 3D Rendering of Topside view (left) and Bottom side view (right) of the PCB

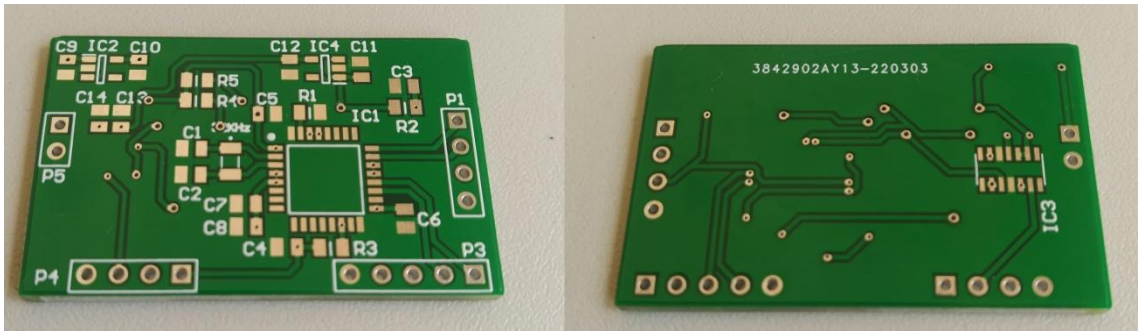


Figure 4.11 - Resulting PCB

4.3 Software

The software used for this MCU is the same as the one used for the earlier prototype with the STM32F7xxx MCU, since they are both STM32 and the code developed isn't so complex that the STM32L0xxx processor and memory can't handle it. So, the only things that were changed, were the intrinsic libraries referring to this specific MCU.

In the beginning of this project, it was suggested to use an SD card to store the data, but because of the huge shortage of components that has been happening since a few years ago, all MCUs with integrated SD card modules were either sold out or were extremely expensive and hard to come by, so, the decision taken was that the data would be sent via serial port to a "master" device which would have the job of receiving and storing the data.

4.4 Chapter Summary

The main goal in this chapter was to design and develop a functional PCB to accommodate all the circuitry necessary to complete the low power application. Starting with the choice of components with reasonably good low power characteristics, like the capability of entering a low power sleep mode for the MCU and the sensor, and the quiescent current consumption for the voltage regulators. From there the schematic

was made without any major difficulties which led the way for the actual PCB design. In the end, the goal of this chapter was completed, with the biggest difficulty being having to choose components in a time where the supply for them isn't able to follow up with the demand.

5. Tests and Results

This chapter is dedicated to all the tests that were made in an attempt to prove the concept that was idealized in the beginning of the project and their corresponding results. Not only the hardware, but also the software developed was tested. The devices tested were the ones mentioned, studied and developed through the course of this dissertation. Some of the tests were validated using a commercially available, transmissive technology pulse oximetry device.

5.1 Heart Rate Click Board

For this device, the tests were divided into two categories. Tests made in humans and tests made in other in vivo animals. For humans, it began with testing the different options for LED power to see which one provided the best results, then testing the software developed to make sure it was working properly and thus could be applied on marine animals and, testing the board with different settings to see how that affects the power consumption. For the in vivo testing, the sensor was used on marine animals to verify if the technology could be used following the same approach used for humans.

All tests performed in this chapter using this board, were conducted using a supply voltage of 3.3 V.

5.1.1 Humans

LED Power

The MAX30100 library used presents 16 different values for the LED current. These values are 0, 4.4, 7.6, 11, 14.2, 17.4, 20.8, 24, 27.1, 30.6, 33.8, 37, 40.2, 43.6, 46.8 and 50, all in mA.

The difference between using one or another value in not only in the power consumption of the sensor, depending on the application, using different values can result in very different PPGs.

It is unnecessary to showcase the results from all values, as they are very close between each other. As of that, the values, 4.4 mA, 14.2 mA, 20.8 mA, 27.1 mA, 33.8 mA, 43.6 mA and 50 mA were the choices of red and infrared led currents to be tested.

- 4.4 mA

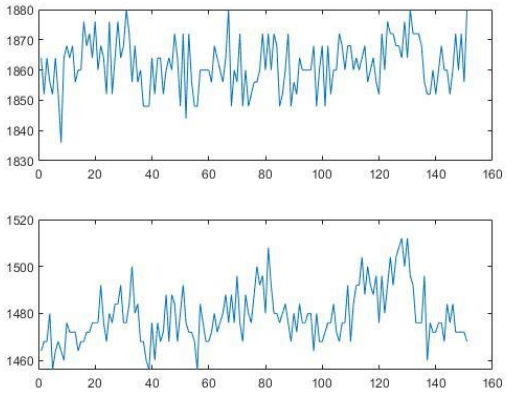


Figure 5.1 – Output results from red (top) and infrared (bottom) illumination using 4.4 mA led current

- 14.2 mA

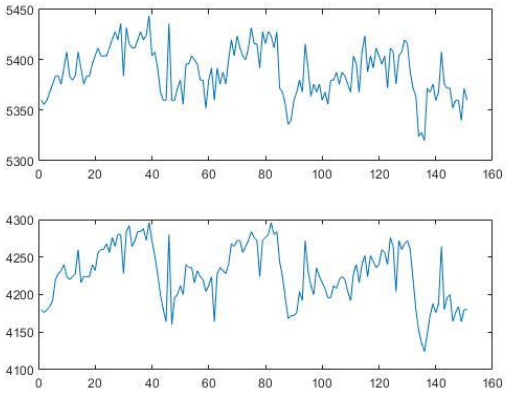


Figure 5.2 – Output results from red (top) and infrared (bottom) illumination using 14.2 mA led current

- 20.8 mA

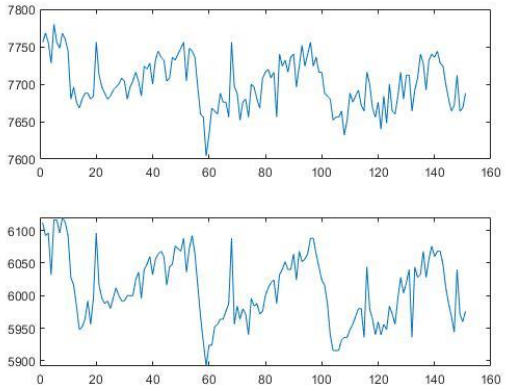


Figure 5.3 – Output results from red (top) and infrared (bottom) illumination using 20.8 mA led current

- 27.1 mA

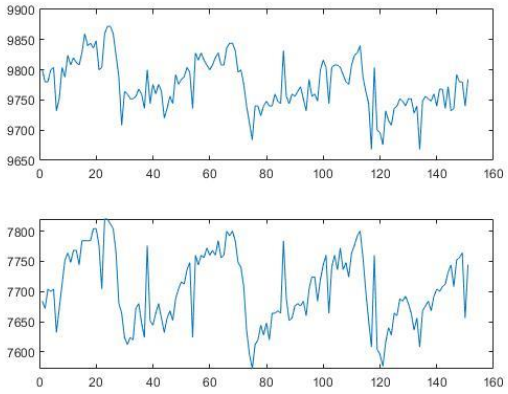


Figure 5.4 – Output results from red (top) and infrared (bottom) illumination using 27.1 mA led current

- 33.8 mA

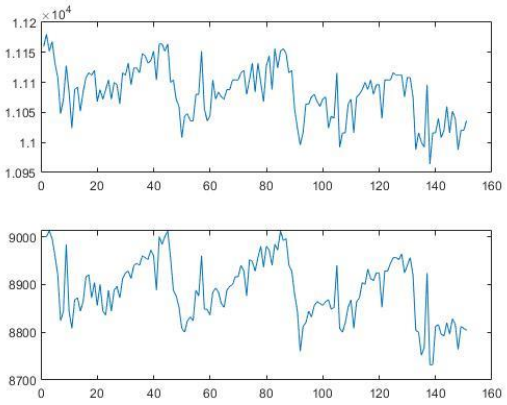


Figure 5.5 – Output results from red (top) and infrared (bottom) illumination using 33.8 mA led current

- 43.6 mA

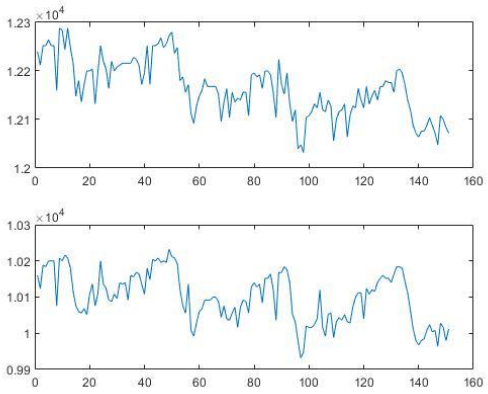


Figure 5.6 – Output results from red (top) and infrared (bottom) illumination using 43.6 mA led current

- 50 mA

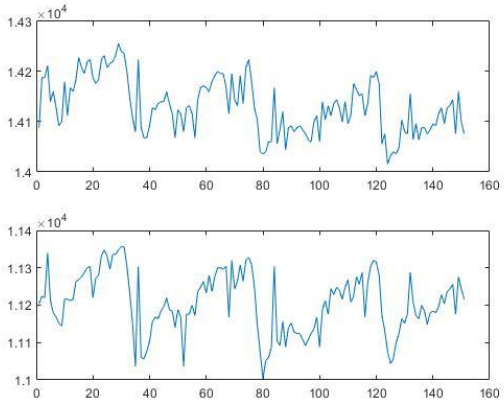


Figure 5.7 – Output results from red (top) and infrared (bottom) illumination using 50 mA led current

With 4.4 mA, the results are quite bad, without even being perceivable that this is a graph of a PPG (Figure 5.1). With 14.2 mA, the graph starts to look like a regular PPG, but still with a lot of noise (Figure 5.2). Between 20.8, 27.1 and 33.8 mA, the results are very similar, and the best presented so far, as both the red and infrared signals look like regular PPGs (Figure 5.3, Figure 5.4 and Figure 5.5). In both 43.6 and 50.0 mA settings, the infrared signal is cleaner than in the previous settings of LED current and the only noticeable difference is that with 50 mA, the red signal is slightly clearer (Figure 5.6 and Figure 5.7).

All of these signals are quite noisy, with some being better than others, but they shouldn´t be like this. During the progress of this dissertation, two Heart Rate Click boards were used, the first one being the one that was encased in resin for the in vivo testing, and the raw results from that board were very different than the raw results from the one used for this test. It can't be understood why or how the results are different, since the setup is exactly the same, but since the first one broke after the in vivo testing and this one was the only one remaining, more tests couldn't be performed.

Power Consumption Test

As for this test, it was evaluated how the sample rate and the LED current affect the overall power consumption of the device. To perform it the device was set as usually, but with a digital multimeter connected in series between the microcontroller and the board to measure the current consumed (Figure 5.8) and then the parameters mentioned were changed to find the best balance between performance and power consumption. The values chosen were the same as the previous test, for the same motive.

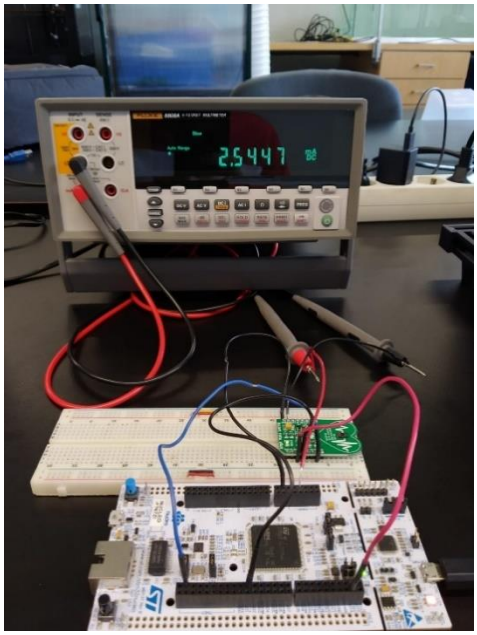


Figure 5.8 - Current and Sample Rate Test Setup

- 50 Hz

Table 5.1 - Board Consumption at 50 Hz Sample Rate

LED peak current (mA)	Board consumption (mA)
4.4	2.54
14.2	3.19
20.8	3.52
27.1	3.80
33.8	4.04
46.2	4.48
50	4.58

- 100 Hz

Table 5.2 - Board Consumption at 100 Hz Sample Rate

LED peak current (mA)	Board consumption (mA)
4.4	3.08
14.2	4.56
20.8	5.31
27.1	5.94
33.8	6.51
46.2	7.52
50	7.74

From these two tests, two main conclusions can be taken.

- Using a sample rate of 100 Hz over 50 Hz isn't justified, because the difference of performance is nearly negligible but the current consumed almost doubles.
- Since the LPF developed can nullify most of the high frequency noise present in the signal, nearly every option of LED current is viable so the choice can be the one with the best power consumed/performance ratio.

With this in mind, the values that were set are:

Sample Rate: 50 Hz

LED current: 20.8 mA

Software Test

The test used to validate the software was a comparison of measurements between the MAX30100 sensor with the software developed and the device used for calibration. The setup was to place one finger on the sensor while another finger from the same hand had the medical device while a helper took notes of the SpO_2 and heart rate values from the sensor and the medical device. This setup can be seen in Figure 5.9.

At first it seemed like something was wrong because the readings were not matching, but after some time, it was concluded that the medical device was giving slightly delayed results compared to the sensor and that the results from the developed software were in fact accurate.

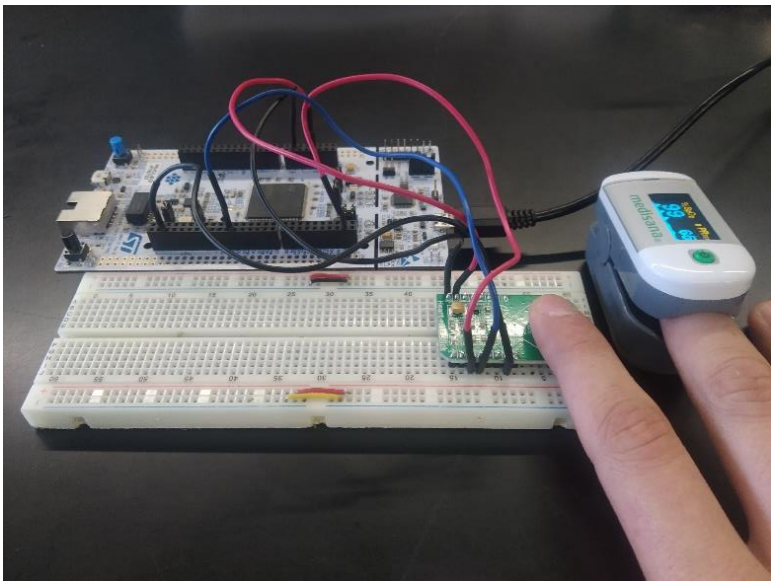


Figure 5.9 - Human Tests Setup

5.1.2 In Vivo

The marine animal that was used for testing was a ray, specifically the undulate ray (*Raja undulata*). The animal was captured by fishermen and brought to shore alive, inside a container filled with sea water. The test subject is seen in Figure 5.10.

Figure 5.10 - Test Subject

Since the filters were calibrated for usage on humans, they could not work for the animal in question. So, in this first attempt, all that was done was the collecting of raw samples for later analysis.

For testing, the animal was brought out of the container, the sensor was placed in different places on the body of the animal, for roughly the same amount of time, to see which location would give the best results.

Everything had to be done very quickly since samples taken from a stressed animal, can be very different from the same animal in a relaxed state. Figure 5.11 is a picture taken in one of the moments of testing.

Figure 5.11 - Testing the Resin Encased MAX30100 Prototype

The results from performing this test can be seen in Figure 5.12 and Figure 5.13. These correspond to the red and infrared LED results respectively. They had to be taken separately due to a software problem that occurred at the time. In both graphs, the DC component was removed and a 3 sample moving average filter was applied after to reduce most of the high frequency noise and leave the signal more readable. In both figures, in blue are the raw samples and in orange, the filtered samples.

Looking at these results it can't be said that there is a heartbeat among all the noise, some repetitive peaks are noticeable. This could be promising but at the moment, all that can be done is speculate about what is present in this samples and hopefully the results from the other sensor being tested will show better results.

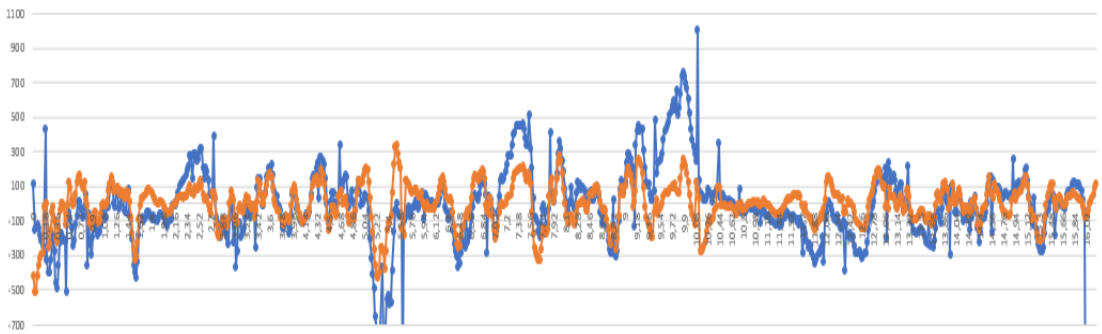


Figure 5.12 - MAX30100 Red led result in a marine animal

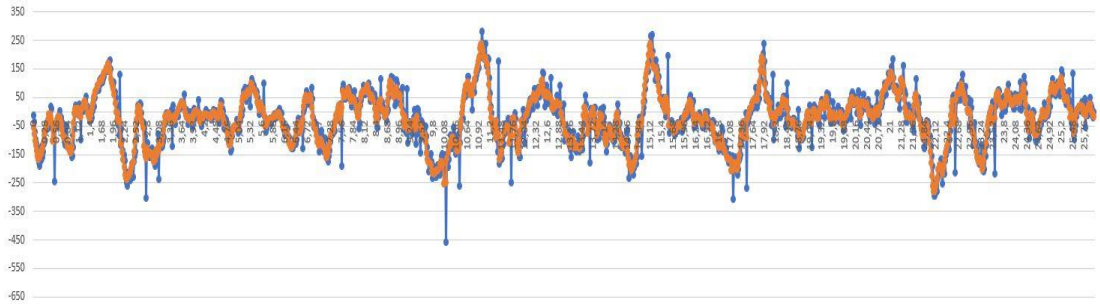


Figure 5.13 - MAX30100 Infrared led result in a marine animal

5.2 MAX30110ACCEVKIT

The tests performed with this kit followed the same metrics as the ones done with the Heart Rate Click Board, to make sure the comparison was as reliable as possible with the exception that power consumption wasn't tested. Since this kit doesn't allow for the usage of self-developed software, all that was done was the collecting of raw samples, to see if the MAX30110 could be a useful addition to the project. Also, the interface used to operate the kit also allowed the testing of different parameters while receiving live results, which was very useful for optimizing their values.

5.2.1 Humans

As was said, human testing of the MAX30110 consisted of changing the main parameters, sample rate and LED power, using the software provided and comparing the results as to optimize them and maximize the probability of achieving a good outcome when it came to in vivo testing.

In all tests a 30 sample medium filter was applied to remove the DC component from the signals and in blue are the red samples and the infrared in orange.

Sample Rate

- 50 Hz

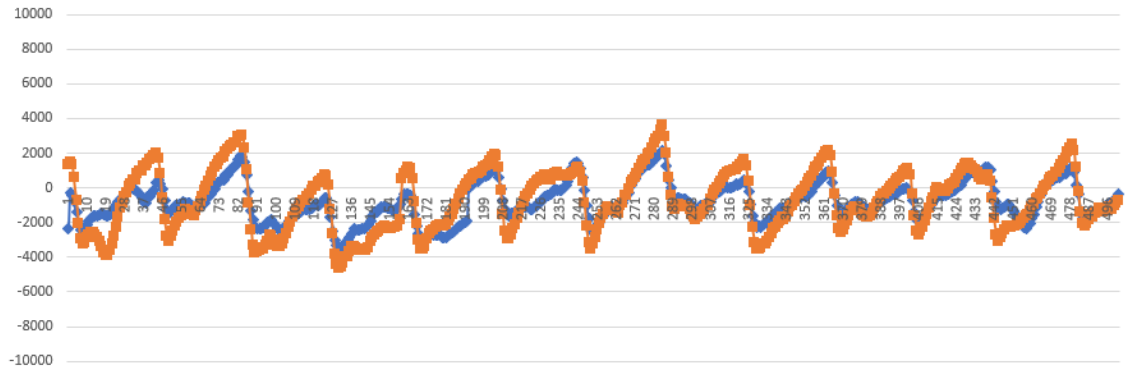


Figure 5.14 - MAX30110 50 Hz Sample Rate Result

- 100 Hz

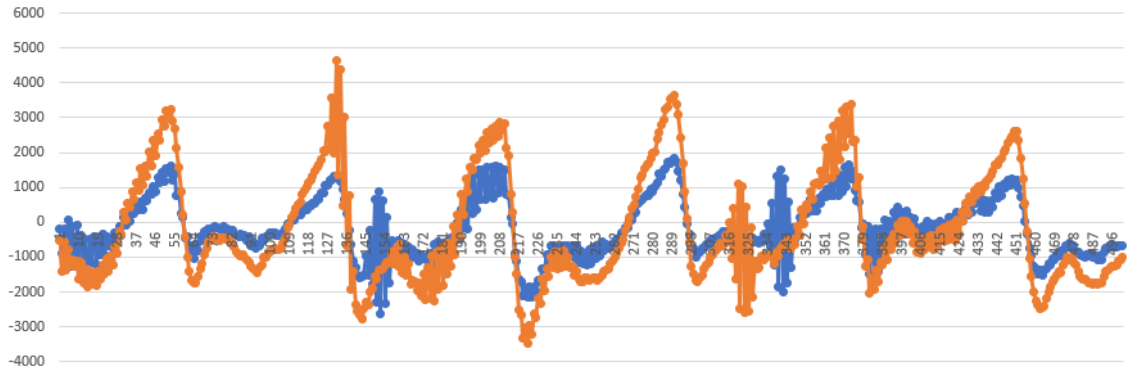


Figure 5.15 - MAX30110 100 Hz Sample Rate Result

After comparing results from 500 samples (Figure 5.14 and Figure 5.15), it is observed that having a higher sample rate isn't needed since the results are very much alike. Therefore, it was decided to use the 50 Hz sample rate.

LED Power

- 1 mA

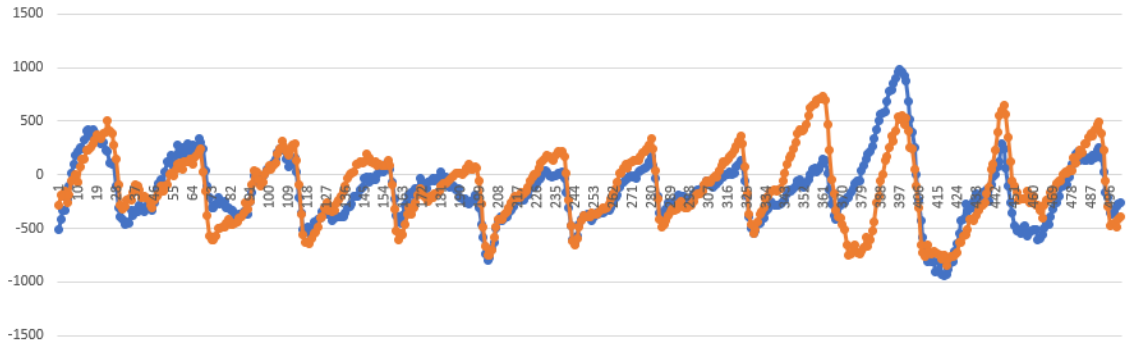


Figure 5.16 - MAX30110 1 mA LED Power Result

- 3 mA

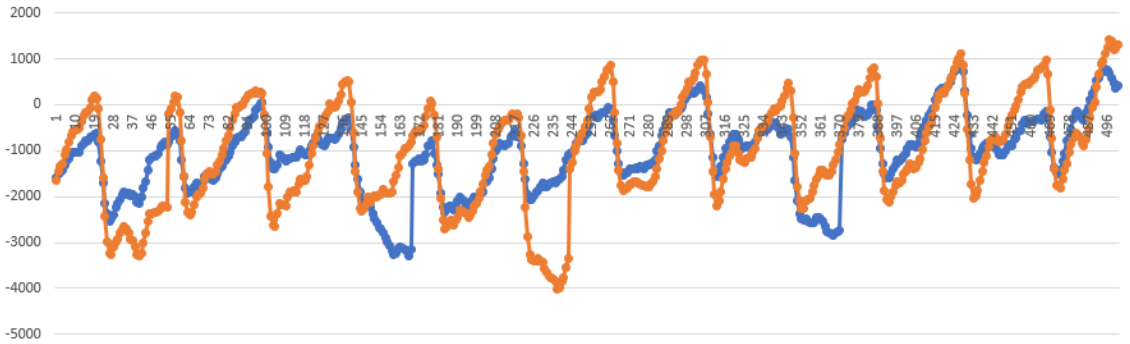


Figure 5.17 - MAX30110 3 mA LED Power Result

- 5 mA

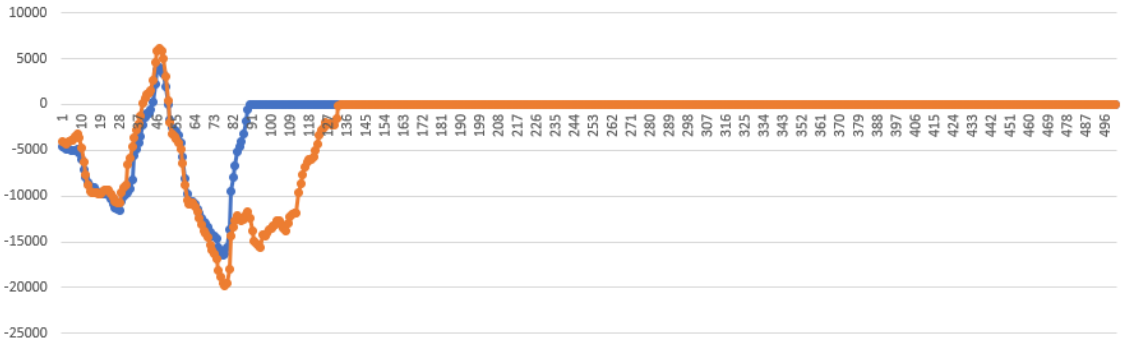


Figure 5.18 - MAX30110 5 mA LED Power Result

The 1 mA and 3 mA (Figure 5.16 and Figure 5.17) results are very much similar and as expected, but when the current goes up to 5 mA (Figure 5.18) something goes wrong. In this case what's happening is the amount of light being reflected is too much for the ADC range and it saturates. Since both 1 mA and

3 mA results where acceptable, choosing one over the other is indifferent, but as the animal's veins could be found deeper than human veins, it was opted for the 3 mA to give the light more penetrating power.

5.2.2 In Vivo

As it was said, the test performed in vivo with this sensor was virtually the same as the one performed for the MAX30100, therefore meaning that the raw samples were collected by placing the sensor in various places of the animal's body, focusing on the belly (Figure 5.19). Since this sensor allowed for live visualization of the data being collected, the test wasn't made "blindly", without knowing if the data was by any means viable. This meant that it was possible to focus on points of the body where the PPG looked the most like a human PPG, or where the signal was most rhythmic, resembling a heart rate.

Figure 5.19 – Testing the MAX30110 Sensor

The results from this test can be seen in Figure 5.20. In blue are the red samples and in orange the infrared. This time both samples were taken concurrently. A lot more samples were taken, but only the most promising results will be shown.

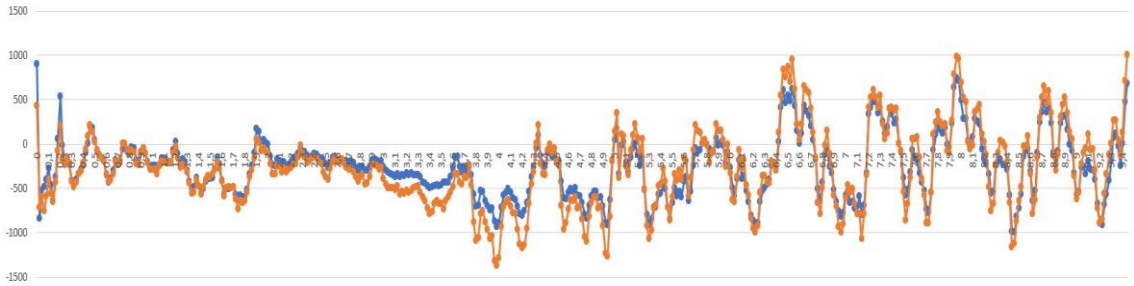


Figure 5.20 - MAX30110 In Vivo Results

With this device, there is still a lot of noise but towards half of the samples, there starts to appear something rhythmic possible resembling a heart rate. Moreover, the red and infrared signals are similar, as expected from the literature, if the SpO_2 is constant.

Two consecutive peaks were chosen, and after converting the x axis from "sample number" to "time", it was even possible to make a heart rate calculation, using the formula mentioned in chapter 3.

$$Bpm = \frac{60}{TPico_2 - TPico_1} \Leftrightarrow$$

$$\Leftrightarrow Bpm = \frac{60}{5{,}02 - 4{,}34} \Leftrightarrow$$

$$\Leftrightarrow Bpm = 88$$

5.3 Chapter Summary

The results obtained from the tests performed and discussed in this chapter are representative of three things. First, the optimization of every available parameter to achieve the best performance to consumption ratio in the MAX30100 sensor, to be applied in the low power application and for the in vivo testing. Next the testing of different parameters to attain the best possible results while testing the MAX30110 in vivo.

From the low power application to the in vivo testing, the results were considered good, while also acknowledging that they can be better since this technology, without any major adaptations isn't as reliable for wild animals as it is for humans.

6. Conclusions and Future Work

This chapter will contain an analysis of the objectives completed and of the conclusions reached during the progress of this dissertation, as well as some of the future work that can be done to improve the prototype from the findings made. The purpose of this being, if someone decides to pick up the work where the author left it, there will already be sort of a guideline of where to begin and what to improve/change.

6.1 Conclusions

The aim of this dissertation was to test if pulse oximetry techniques developed with human care in mind could function in monitoring devices for marine animals and then designing a device capable of applying them but specialized in marine animals.

Through the progress of the project, two devices were tested, focusing on the MAX30100 as it was included in a board easier to adapt and to develop software for. The software was developed with versatility in mind so that when it came to testing, the device could be tuned to fit the requirements for each different individual that it could be tested on.

The opportunity to test the sensors in an actual marine animal, brought light to some important considerations that weren't taken into account at the beginning of the project. A lot of marine animals, like rays, produce a thick, slimy coat around their bodies to protect them. This slime, being transparent, doesn't affect the light from the sensors, but highly affect their stability while placed against the animal's skin. This means that any non-invasive means of attaching the device to the animal will most likely be associated with poor results due to motion artifacts, so either different approaches on the technique or new ways of fixating must be considered.

This opportunity also showed that despite being very similar, the MAX30110, with its higher sensitivity, provided better results and would probably be a better option for the development of a new prototype.

Unfortunately, the objective of designing a complete low power pulse oximeter with an embedded microcontroller couldn't be completed, but with a little more effort it can be as every aspect of the design is already made, and the problems it underwent aren't associated with it.

6.2 Future Work

Subcutaneous LEDs

After realizing that that this technology, applied in wild animals as is, will be subject to a lot of problems with the fixation of the sensor with the right amount of pressure and without it moving around inducing errors, and not only that, there is also the problem of different animals having different thicknesses of skin and fat which will make it difficult for the light to reach the veins. With this in mind it was thought of a slightly invasive method, where small LEDs are mounted on needles and placed subcutaneously. This would theoretically solve the problems mentioned and would even help with the fixation. Including this in the design would add a lot of versatility to the device, but it is still not known if pulse oximetry works subcutaneously.

Some LEDs for this purpose where suggested, like the APG0603SURC-TT (Figure 6.1), which is a 0.65 x 0.35 x 0.2 mm SMD Chip LED or the APTD1608SECK/J3-PF a 1.6 x 0.8 mm SMD Chip LED. Both are small enough to fit in a needle. Using this LEDs will also mean that there will be the need to develop a circuit to drive them and for the photodiode.

Figure 6.1 - APG0603SURC-TT SMD LED

PCB Size and Power Consumption Optimization

The PCB made was made with the dimensions in mind, but there is still a lot of room for improvements. Especially if considered that with the circuitry required for driving the subcutaneous LEDs and photodiode, the size will increase.

As for power consumption, if it can be reduced is always better, and there are ways, using sleep modes for the microcontroller, choosing low power LEDs, voltage regulators with smaller quiescent current values, etc.

This was only the first version, and there are a lot of ways to improve it.

Monitoring System

Including this project on a monitoring system would maximize its potential if the monitoring system supported wireless data transmission. This means that the data and results could be received without the need of collecting the sensor and that, paired with optimized power consumption and a good battery equals less impact and stress on the animal's everyday life. Not only that, if the monitoring system has other sensors, it would be easier to understand what the animal was doing, in every reading, e.g., with a pressure sensor, the readings could be associated with dives; with a temperature sensor, it can help understanding if the temperature of the water has any effect on the physiological phenomena happening inside their bodies; with a dissolved oxygen sensor, how low and high water oxygen levels affect the SpO_2; etc.

Software Optimization

Using digital filters to process the data into a clean signal with easily detectable peaks is not the only method that exists to calculate heart rate. Research as shown that using Fast Fourier Transforms (FFT) can actually be faster and consume less CPU processing power in that task.

A FFT is an algorithm that computes the discrete fourier transform (DFT) of a sequence. A fourier analysis converts a signal into individual spectral components, providing frequency information about the signal. It essentially converts the signal domain, usually time or space, into the frequency domain. Since the heart beats rhythmically, it only makes sense that the FFT can be used to compute the frequency at which it beats, therefore calculating the heart rate [21][22].

It was also shown that this approach is less prone to error induced from motion artifacts and taking into account that the main objective of this project is to apply the technology in a wild animal that is almost constantly moving, this may prove to be a very powerful tool in accomplishing that [23].

References

[1] R. Abedalmoniam and S. Fadul, "Oxygen Level Measurement Techniques: Pulse Oximetry," *J. Eng. Comput. Sci.*, vol. 16, no. 2, pp. 1–5, 2017, [Online]. Available: http://journal.sustech.edu/index.php/JECS/article/view/186/53%0Ahttp://journal.sustech.edu/index.php/JECS/article/view/186.

[2] S. Worapruekjaru and K. Dougjitjaroen, "Improved Efficiency of Fingertip Reflective Pulse Oximetry with 2 positions Integrated Optical Biosensor," *ACM Int. Conf. Proceeding Ser.*, pp. 1–5, 2020, doi: 10.1145/3406601.3406637.

[3] H. Lee, H. Ko, and J. Lee, "Reflectance pulse oximetry: Practical issues and limitations," *ICT Express*, vol. 2, no. 4, pp. 195–198, 2016, doi: 10.1016/j.icte.2016.10.004.

[4] R. Finnerty, "How to Design a Better Pulse Oximeter."

[5] T. Sušec, "Historical Review," *Acta Clin. Croat.*, vol. 60, no. 2, p. 332, 2021, doi: 10.5005/jp/books/12492_2.

[6] A. Van Meter *et al.*, "Beat to Beat: A Measured Look at the History of Pulse Oximetry," *J. Anesth. Hist.*, vol. 3, no. 1, pp. 24–26, 2017, doi: 10.1016/j.janh.2016.12.003.

[7] C. T. Yen and C. H. Liao, "Blood pressure and heart rate measurements using photoplethysmography with modified Ircn," *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 1973–1986, 2022, doi: 10.32604/cmc.2022.022679.

[8] J. E. Sinex, "Pulse oximetry: Principles and limitations," *Am. J. Emerg. Med.*, vol. 17, no. 1, pp. 59–66, 1999, doi: 10.1016/S0735-6757(99)90019-0.

[9] P. E. Parsons and J. P. Wiener-Kronish, *Critical Care Secrets*, 4th ed. 2007.

[10] M. Hickey, N. Samuels, N. Randive, R. M. Langford, and P. A. Kyriacou, "An In Vivo Investigation of Photoplethysmographic Signals and Preliminary Pulse Oximetry Estimation from the Bowel Using a New Fiberoptic Sensor," vol. 112, no. 5, pp. 1104–1109, 2011, doi: 10.1213/ANE.0b013e31820f8df3.

[11] P. King, "Design Of Pulse Oximeters," *IEEE Eng. Med. Biol. Mag.*, vol. 17, no. 3, pp. 117–117, 2005, doi: 10.1109/memb.1998.677180.

[12] J. L. Mildenhall, "The theory and application of pulse oximetry," *J. Paramed. Pract.*, vol. 1, no. 2, pp. 52–58, 2008, doi: 10.12968/jpar.2008.1.2.42016.

[13] Maxim Integrated, "Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health," *Lect. Notes Energy*, vol. 38, pp. 1–29, 2014, [Online]. Available: www.maximintegrated.com.

[14] S. Microelectronics, "UM1974 User manual STM32 Nucleo-144 boards," no. December, 2017, [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/user_manual/group0/26/49/90/2e/33/0d/4a/da/DM00244518/files/DM00244518.pdf/jcr:content/translations/en.DM00244518.pdf.

[15] "MAX30100_for_STM32_HAL." https://github.com/eepj/MAX30100_for_STM32_HAL/blob/master/max30100_for_stm32_h

al.c (accessed Jun. 21, 2021).

[16] "Digital Signal Processing." https://exstrom.com/journal/sigproc/dsigproc.html?fbclid=IwAR34_cT5t0iNYak-M_NhyXzDuml1IlWAvPuOZhdzd3r2z6XsjKjrCS7ZRYw (accessed May 31, 2022).

[17] M. C. U. Arm, K. B. Flash, K. B. Sram, and K. B. Eeprom, "STM32L051x6 STM32L051x8," no. June, pp. 1–28, 2014.

[18] "1.8 V, 150 mA , LDO Regulator, ADP165/ADP166, Rev. A, Analog Devices [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adp165_166.pdf." pp. 1–24, 2014.

[19] "3.1 V, 200-mA , Low-Dropout Regulator, TPS7A05, Rev. D, Texas Instruments. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps7a05.pdf?ts=1656337230669&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS7A05%253FHQS%253Dti-null-null-verifima." 2019.

[20] B. Archambeault and D. White, "Printed Circuit Board decoupling capacitor performance for optimum EMC design," no. January, pp. 1–40, 1999.

[21] R. Aisuwarya, Hendrick, and Meitiza, "Analysis of Cardiac Frequency on Photoplethysmograph (PPG) Synthesis for Detecting Heart Rate Using Fast Fourier Transform (FFT)," *ICECOS 2019 - 3rd Int. Conf. Electr. Eng. Comput. Sci. Proceeding*, pp. 391–395, 2019, doi: 10.1109/ICECOS47637.2019.8984512.

[22] N. H. Mohd Sani, W. Mansor, K. Y. Lee, N. Ahmad Zainudin, and S. A. Mahrim, "Determination of heart rate from photoplethysmogram using Fast Fourier Transform," *2015 Int. Conf. BioSignal Anal. Process. Syst. ICBAPS 2015*, no. May, pp. 168–170, 2015, doi: 10.1109/ICBAPS.2015.7292239.

[23] B. B. Jayadevappa, M. S. Holi, and J. B. α Mallikarjun S Holi σ, "An Estimation Technique using FFT for Heart Rate Derived from PPG Signal," *Glob. J. Res. Eng.*, vol. 15, no. 7, 2015, [Online]. Available: https://engineeringresearch.org/index.php/GJRE/article/viewFile/1364/1295.

Appendix A: Code Developed

```
//-----------------------------------------------------------------------------------------------------------------------
/* Variables Used */
#define samplesize 128
uint8_t sampleRate=50;
uint16_t output_LPF_Red[samplesize];
uint16_t output_LPF_Ir[samplesize];
int16_t output_HPF_Red[samplesize];
int16_t output_HPF_Ir[samplesize];
uint16_t output_Mean_Red[samplesize];
uint16_t output_Mean_Ir[samplesize];
uint16_t all_samples_red[samplesize];
uint16_t all_samples_ir[samplesize];
uint8_t flag=0;
struct Peaks peak_array[10];
float heart_rate=0;
uint8_t SPO2=0;
float R=0;
/* Filter Variables */
uint8_t n_LP = 2;
uint8_t n_HP = 2;
float s_LP = 50;
float s_HP = 50;
float f_LP = 3;
float f_HP = 0.3;
float r_LP;
float r_HP;
float a_LP;
float a_HP;
float a2_LP;
float a2_HP;
uint16_t x_LP_R;
int16_t x_HP_R;
uint16_t x_LP_Ir;
int16_t x_HP_Ir;
float A_LP[32];
float A_HP[32];
float d1_LP[32];
float d2_LP[32];
float d1_HP[32];
float d2_HP[32];
float w0_red[2];
float w1_red[2];
float w2_red[2];
float w0_ir[2];
float w1_ir[2];
float w2_ir[2];
float w0_red_HP[2];
float w1_red_HP[2];
float w2_red_HP[2];
float w0_ir_HP[2];
float w1_ir_HP[2];
float w2_ir_HP[2];
```

A. 1 - Variables Used

```
//-----------------------------------------------------------------------------------------------------------------------
/* Data Processing Function */
void data_processing(void){
  if(flag==1){
      Low_Pass_Filter_Red(samplesize, all_samples_red);
      Low_Pass_Filter_Ir(samplesize, all_samples_ir);
      High_Pass_Filter_Red(samplesize, output_LPF_Red);
      High_Pass_Filter_Ir(samplesize, output_LPF_Ir);
      findpeaks(samplesize, output_HPF_Ir);
      heart_rate_calculator(peak_array);
      SPO2_calculator(samplesize, output_LPF_Red, output_LPF_Ir, output_HPF_Red, output_HPF_Ir);
      MAX30100_Plot_HR_SPO2ToUART(_max30100_uuart, heart_rate, SPO2);
      flag=0;
  }
}
//-----------------------------------------------------------------------------------------------------------------------
/* Filter Coeficients Calculation */
void Filters_Prep(void){
  n_LP = n_LP/2;
  n_HP = n_HP/2;
  a_LP = tan(3.1416*f_LP/s_LP);
  a2_LP = a_LP*a_LP;
  a_HP = tan(3.1416*f_HP/s_HP);
  a2_HP = a_HP*a_HP;
  for(uint8_t i=0; i<n_LP; i++){
    r_LP = sin(3.1416*(2.0*i+1.0)/(4.0*n_LP));
    s_LP = a2_LP + 2.0*a_LP*r_LP + 1.0;
    A_LP[i] = a2_LP/s_LP;
    d1_LP[i] = 2.0*(1-a2_LP)/s_LP;
    d2_LP[i] = -(a2_LP - 2.0*a_LP*r_LP + 1.0)/s_LP;
  }
  for(uint8_t i=0; i<n_HP; i++){
    r_HP = sin(3.1416*(2.0*i+1.0)/(4.0*n_HP));
    s_HP = a2_HP + 2.0*a_HP*r_HP + 1.0;
    A_HP[i] = 1.0/s_HP;
    d1_HP[i] = 2.0*(1-a2_HP)/s_HP;
    d2_HP[i] = -(a2_HP - 2.0*a_HP*r_HP + 1.0)/s_HP;
  }
}
```

A. 2 - Data Processing Function and Filter Coefficients Calculator Function

```
//-------------------------------------------------------------------------------------------------------------------------------
/* Red Signal Filtering */
uint16_t Low_Pass_FilterR(uint16_t sample){
   x_LP_R = sample;
   for(uint8_t i=0; i<n_LP; i++){
      w0_red[i] = d1_LP[i]*w1_red[i] + d2_LP[i]*w2_red[i] + x_LP_R;
      x_LP_R = A_LP[i]*(w0_red[i] + 2.0*w1_red[i] + w2_red[i]);
      w2_red[i] = w1_red[i];
      w1_red[i] = w0_red[i];
   }
   return x_LP_R;
}

void Low_Pass_Filter_Red(uint8_t sampleSize, uint16_t *samples){
   for(uint8_t k=0; k<sampleSize; k++){
     output_LPF_Red[k] = Low_Pass_FilterR(samples[k]);
   }
}

int16_t High_Pass_FilterR(uint16_t sample){
   x_HP_R = sample;
   for(uint8_t i=0; i<n_HP; i++){
      w0_red_HP[i] = d1_HP[i]*w1_red_HP[i] + d2_HP[i]*w2_red_HP[i] + x_HP_R;
      x_HP_R = A_HP[i]*(w0_red_HP[i] - 2.0*w1_red_HP[i] + w2_red_HP[i]);
      w2_red_HP[i] = w1_red_HP[i];
      w1_red_HP[i] = w0_red_HP[i];
   }
   return x_HP_R;
}

void High_Pass_Filter_Red(uint8_t sampleSize, uint16_t *samples){
   for(uint8_t k=0; k<sampleSize; k++){
     output_HPF_Red[k] = -1 * High_Pass_FilterR(samples[k]);
   }
}
```

A. 3 - Red Signal Filters

```
//-------------------------------------------------------------------------------------------------------------------------------
/* Infrared Signal Filtering */

uint16_t Low_Pass_FilterIr(uint16_t sample){
   x_LP_Ir = sample;
   for(uint8_t i=0; i<n_LP; i++){
      w0_ir[i] = d1_LP[i]*w1_ir[i] + d2_LP[i]*w2_ir[i] + x_LP_Ir;
      x_LP_Ir = A_LP[i]*(w0_ir[i] + 2.0*w1_ir[i] + w2_ir[i]);
      w2_ir[i] = w1_ir[i];
      w1_ir[i] = w0_ir[i];
   }
   return x_LP_Ir;
}

void Low_Pass_Filter_Ir(uint8_t sampleSize, uint16_t *samples){
   for(uint8_t k=0; k<sampleSize; k++){
     output_LPF_Ir[k] = Low_Pass_FilterIr(samples[k]);
   }
}

int16_t High_Pass_FilterIr(uint16_t sample){
   x_HP_Ir = sample;
   for(uint8_t i=0; i<n_HP; i++){
      w0_ir_HP[i] = d1_HP[i]*w1_ir_HP[i] + d2_HP[i]*w2_ir_HP[i] + x_HP_Ir;
      x_HP_Ir = A_HP[i]*(w0_ir_HP[i] - 2.0*w1_ir_HP[i] + w2_ir_HP[i]);
      w2_ir_HP[i] = w1_ir_HP[i];
      w1_ir_HP[i] = w0_ir_HP[i];
   }
   return x_HP_Ir;
}

void High_Pass_Filter_Ir(uint8_t sampleSize, uint16_t *samples){
   for(uint8_t k=0; k<sampleSize; k++){
     output_HPF_Ir[k] = -1 * High_Pass_FilterIr(samples[k]);
   }
}

//-------------------------------------------------------------------------------------------------------------------------------
/* Sample Collecting */
void MAX30100_CollectSamples(uint16_t *samplesRed, uint16_t *samplesIr, uint8_t sampleSize){
   static int count=0;
   for(uint8_t k=count*sampleSize; k<count*sampleSize+sampleSize; k++){
         all_samples_red[k]=samplesRed[k-count*sampleSize];
         all_samples_ir[k]=samplesIr[k-count*sampleSize];
   }
   count++;
   if(count==8){
   count=0;
   flag = 1;
   }
}
```

A. 4 - Infrared Signal Filters and Sample Collect Function

```
//-------------------------------------------------------------------------------------------------------------------------------------------
/* Findpeaks Function */
void findpeaks(uint8_t sampleSize, int16_t *samples){
  struct Peaks peak;
  uint8_t k_findpeaks=0;
  int16_t sum=0;
  float avg=0;
  for(uint8_t k=0; k<sampleSize; k++){
    sum += abs(samples[k]);
    peak_array[k].value=0, peak_array[k].index=0, peak_array[k].idx=k;
  }
  avg = (float)sum/(float)sampleSize;
  for(uint8_t i=2; i<sampleSize-1; i++){
    if(abs(samples[i+1]-samples[i-2])>1.4*avg && abs(samples[i+2]-samples[i-1])<abs(samples[i+1]-samples[i-2])){
        peak.value=samples[i];
        peak.index=i;
      if(peak_array[k_findpeaks].idx==0){
        peak_array[k_findpeaks].value=peak.value;
        peak_array[k_findpeaks].index=peak.index;
        k_findpeaks++;
      }
      else if((peak.index-peak_array[k_findpeaks-1].index)>10){
        peak_array[k_findpeaks].value=peak.value;
        peak_array[k_findpeaks].index=peak.index;
        k_findpeaks++;
      }
    }
  }
}
```

A. 5 - Findpeaks Function

```
//-------------------------------------------------------------------------------------------------------------------------------------------
/* Heart Rate Calculator Function */
void heart_rate_calculator(struct Peaks *array){
  uint8_t i=0;
  uint8_t time_between=0;
  uint8_t time_sum=0;
  while(array[i+1].index!=0){
    time_between=array[i+1].index-array[i].index;
    time_sum+=time_between;
    i++;
  }
  time_sum=(float)time_sum/(float)i;
  heart_rate=((float)60/(time_sum*((float)1/(float)sampleRate)));
}
//-------------------------------------------------------------------------------------------------------------------------------------------
/* SpO2 Calcultor Function */
void SPO2_calculator(uint8_t sampleSize, uint16_t *samplesRed, uint16_t *samplesIr, int16_t *highsamplesRed, int16_t *highsamplesIr){
  float sqrsumR=0;
  float sqrsumIr=0;
  float sumR=0;
  float sumIr=0;
  float RMS_Red=0;
  float RMS_Ir=0;
  float red_mean=0;       //valor DC vermelho
  float ir_mean=0;        //valor DC infravermelho
  R=0;
  SPO2=0;
  for(uint8_t i=0;i<sampleSize;i++){
    sumR+=samplesRed[i];
    sumIr+=samplesIr[i];
    sqrsumR+=highsamplesRed[i]*highsamplesRed[i];
    sqrsumIr+=highsamplesIr[i]*highsamplesIr[i];

  }
  RMS_Red=sqrt(sqrsumR/(float)sampleSize);
  RMS_Ir=sqrt(sqrsumIr/(float)sampleSize);
  red_mean=sumR/(float)sampleSize;
  ir_mean=sumIr/(float)sampleSize;
  R=(RMS_Red/red_mean)/(RMS_Ir/ir_mean);
  SPO2=110-25*R;
  if(SPO2>100)SPO2=100;
  sqrsumR=0;
  sqrsumIr=0;
  sumR=0;
  sumIr=0;
  RMS_Red=0;
  RMS_Ir=0;
  red_mean=0;
  ir_mean=0;
}
```

A. 6 - Heart Rate and SpO2 Calculator Functions

```
//-----------------------------------------------------------------------------------------------------------------------------
/* Interrupt Handler */
void MAX30100_InterruptHandler(void){
  uint8_t itReg = MAX30100_ReadReg(MAX30100_INTERRUPT);
  if((itReg >> MAX30100_A_FULL) & 0x01){
    MAX30100_ReadFIFO();
    if(_max30100_mode == MAX30100_HRONLY_MODE)
      MAX30100_PlotIrToUART(_max30100_uuart, _max30100_ir_sample, 16);
    else if(_max30100_mode == MAX30100_SPO2_MODE)
    if(flag==0){
      MAX30100_CollectSamples(_max30100_red_sample, _max30100_ir_sample, 16);
    }
    MAX30100_SetMode(_max30100_mode);
  }else if((itReg >> MAX30100_TMP_RDY) & 0x01){
    _max30100_temp = MAX30100_ReadTemperature();
    MAX30100_EnableInterrupt(1, 0, 0, 0);
  }else if((itReg >> MAX30100_HR_RDY) & 0x01){

  }else if((itReg >> MAX30100_SPO2_RDY) & 0x01){

  }
}
```

A. 7 - Interrupt Handler Function

```
//-----------------------------------------------------------------------------------------------------------------------------
/* Main */
int main(void)
{
  /* MCU Configuration--------------------------------------------------------*/
  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */
  Filters_Prep();
  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART3_UART_Init();
  MX_I2C1_Init();

  /* MAX30100 Configurations */
  MAX30100_Init(&hi2c1, &huart3);
  MAX30100_SetSpO2SampleRate(MAX30100_SPO2SR_50);
  MAX30100_SetLEDPulseWidth(MAX30100_LEDPW_DEFAULT);
  MAX30100_SetLEDCurrent(MAX30100_LEDCURRENT_20_8, MAX30100_LEDCURRENT_20_8);
  MAX30100_SetMode(MAX30100_SPO2_MODE);

  /* Activate receive interrupt for one char */
  HAL_UART_Receive_IT(&huart3, &Rx_Buffer[Rx_index], 1);
  printf("\nPress enter to begin measure....\n");

  /* USER CODE END 2 */
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    data_processing();
  }
}
```

A. 8 - Main Loop Function