Development of an Automated Platform for Functional Testing of Multiparametric Vital Signal Monitors

RAFAEL DE PAIVA TEIXEIRA
(Licenciado em Engenharia Eletrónica e Telecomunicações)

Dissertação para obtenção do grau de mestre em Engenharia Eletrónica e Telecomunicações, na Área de Eletrónica

Orientador(es):
PhD João Pedro Barrigana Ramos da Costa
PhD Luís Miguel Tavares Fernandes

Júri:
Presidente: PhD Paula Maria Garcia Louro
Vogais:
Paulo Jorge Passos Sério Lourenço

novembro 2024

Development of an Automated Platform for Functional Testing of Multiparametric Vital Signal Monitors

RAFAEL DE PAIVA TEIXEIRA
(Licenciado em Engenharia Eletrónica e Telecomunicações)

Dissertação para obtenção do grau de mestre em Engenharia Eletrónica e Telecomunicações, na Área de Eletrónica

Orientador(es):
PhD João Pedro Barrigana Ramos da Costa
PhD Luís Miguel Tavares Fernandes

Júri:
Presidente: PhD Paula Maria Garcia Louro
Vogais:
Paulo Jorge Passos Sério Lourenço

novembro 2024

Acknowledgements

I would like to express my deepest gratitude to my advisors, Professor João Costa and Professor Miguel Fernandes, for their essential scientific guidance and constant availability throughout the development of this project. Their expertise and support were crucial to my progress and to the realization of this work.

I am also profoundly grateful to my family, girlfriend, and friends for their continuous support throughout my entire academic journey. Their encouragement and understanding have been a constant source of strength and motivation.

My thanks also go to my collegues

I would also like to thank Engineer António Faria Gomes for his assistance in providing the monitor, the main component of this thesis, as well as a patient simulator and all the necessary equipment for interacting with these materials. I am grateful for his availability to clarify professional questions and share specific knowledge in the field.

This page was intentionally left blank.

Abstract

The system proposed in this document presents a solution to the development of an automated platform that performs functional tests for heart rate measurement in generic multiparametric monitors. The automation of this procedure is achieved based on two main blocks, the first is responsible for generating an electrocardiogram signal, passing it to the monitor that is being tested and, after this, the second block is responsible for interpreting the results presented in the monitor and compare it with the generated signal.

The generation of the electrocardiogram signal is composed by an hybrid solution that combines software and custom hardware. Initially, the signal is generated from an audio file previously stored in a database, then passes through a process of digital modulation in software. After this, the signal is sent to a custom circuit board where it is converted to its analog format, filtered and attenuated. This way, is possible to have a signal that simulates the human electrocardiogram.

After this, an analyser block uses techniques of image processing, mainly Optical Character Recognition, to identify the heart rate calculated by the monitor display. It communicates with a Graphical User Interface, where this value is compared with the intended and presented to the user.

The project is motivated by the critical need to guarantee the quality and safety of multiparameter monitors, which play a vital role in the continuous monitoring of patients in clinical environments. Implementing this platform is part of the effort to improve efficiency in quality control, but also contributes to the sustainability of the healthcare system by optimising hospital resources and minimising equipment downtime.

Key Words: Functional Testes, Multiparametric Monitor, ECG, Signal Generation, Computer Vision, OCR, GUI

This page was intentionally left blank.

Resumo

O sistema proposto neste documento apresenta uma solução para o desenvolvimento de uma plataforma que automatiza testes funcionais ao cálculo do batimento cardíaco de monitores multiparamétricos de sinais vitais. A automação deste procedimento é alcançada por dois blocos principais: o primeiro gera um sinal de eletrocardiograma, passando-o ao monitor sob teste, enquanto o segundo interpreta os resultados apresentados no monitor e compara-os com o sinal gerado.

A geração de um sinal de eletrocardiograma combina *software* e *hardware* especificamente desenvolvido para o efeito. Inicialmente, o sinal é gerado a partir de um ficheiro de áudio previamente guardado numa base de dados e passa por processos de modulação digital em *software*. Depois, o sinal é enviado para uma placa de circuito customizada, onde é convertido para o formato analógico, filtrado e atenuado, simulando um eletrocardiograma humano.

Em seguida, um bloco analisador usa técnicas de processamento de imagem, principalmente Reconhecimento Ótico de Caracteres, para identificar o ritmo cardíaco calculado pelo monitor. Este valor é comparado com o esperado e apresentado ao utilizador na interface gráfica.

Este projeto é motivado pela necessidade crítica de garantir a qualidade dos monitores multiparamétricos, que são essenciais para a monitorização contínua de pacientes. O desenvolvimento desta plataforma faz parte do esforço para melhorar a eficiência no controle de qualidade e sustentabilidade do sistema de saúde, otimizando recursos hospitalares e minimizando o tempo de inatividade dos equipamentos.

Palavras chave: Testes funcionais, Monitor multiparamétrico, ECG, Geração de sinais, visão computacional, OCR, GUI

This page was intentionally left blank.

Table of Contents

This page was intentionally left blank.

List of Figures

This page was intentionally left blank.

List of Acronyms

ECG Electrocardiogram

FFT Fast Fourier Transform

GPIO General-Purpose Input/Output

GUI Graphical User Interface

OCR Optical Character Recognition

OpAmp Operational Amplifier

OS Operating system

PCB Printed Circuit Board

ROI Region of Interest

RTC Real-Time Clock

Sd Card Secure Digital Card

SMD Surface-mount Device

SPI Serial Peripheral Interface

This page was intentionally left blank.

1 Introduction

1.1 Motivation

The motivation to the development of an automated platform that realizes functional tests to the electrocardiogram analysis of a generic multiparametric monitor comes from the critical importance that these devices perform in the modern health system. These monitors are widely used in hospitals, health care centers, ambulances and every institution that needs to constantly monitor a patient. These devices can provide several vital data, such as heart rate, blood pressure, oxygen saturation, temperature, and all the information that can be extracted from an electrocardiogram.

The proper functioning of these devices is essential to guarantee the accuracy of the diagnoses, and, consequently, the safety of the patient. The lack of preventive maintenance can result in measurement error compromising the effectiveness of the treatment. Particularly, the electrocardiogram signal is very sensitive to external interference and any technical error can result in poor diagnoses. This means that some regular functional tests are essential to ensure that it is working in ideal conditions.

The automation of the procedure offers a significant advantage in terms of efficiency and accuracy. The traditional methods are performed by a human, that means that it is very susceptible to errors and mistakes. This automation not only results in a more precise and fast verification, as can reduce the human margin of error.

The development of a platform that can operate independently to the manufacturer of the monitor can result in the reduction of costs to the hospital, because it can simplify procedures to the point that, in and ideal scenario, tests could be done by non technical staff as to rule out malfunction issues and provide a first assessment of the equipment.

In this project is presented the solution found to the analysis of the heart rate presented in the monitor's display, but the development of a platform that allows scalability can promote some future works that perform test to all the remaining signals presented in these monitors.

1.2 Goals

This work proposes an architecture and presents an implementation of a system to test the ECG detection of a Multiparametric Vital Signal Monitor. The system proposed is designed to be used by any user, so it has to be very user-friendly, intuitive and as automated as possible.

The proposed system is composed of four main subjects, a user interface, hardware capable of generating ECG signals, the monitor under test and a computer, that uses computer vision to identify the result and returns it to the user interface, where the user can check the results.

The user interface has to be friendly and very intuitive. The user has to be able to select a test from a set of tests and start the procedure. This interface has to have a stopping method, used to stop the generation and the test. At the end of the test, the user has to be able to see the final results and all the details about the test.

While signal generation can be achieved using a simple microcontroller, we have chosen to use a single-board computer, the Raspberry Pi version 4, as we also need to perform image processing tasks. The Raspberry Pi features a general-purpose, multi-core processor that is well-suited for this application.

About the computer vision analysis, it performs some pre-processing techniques to simplify the image, and then an Optical Character Recognition, or OCR, algorythm is performed to extract the relevant information.

Finally, the user can interact with the system using a Graphical User Interface, or GUI.

This solution, using only a single computer, allows to have a final product all in one and plug-and-play, where the user just needs to plug the electrodes, a mouse and a display to verify the results. This solution avoids all the typical human errors and the premature recognition of major errors in this critical devices.

The primary objective of this work is to develop hardware and software modules for the automated functional testing of the ECG component in multi-parametric patient monitors, minimizing human intervention. To achieve this, a system architecture is proposed that includes ECG signal generation and feedback analysis of the monitor's response. Key specifications include the ability to adapt the solution for various brands of monitors and to facilitate future expansion to additional biosignals, such as SpO_2 and blood pressure.

1.3 Document Structure

This document is structured in four main sections, the Fundamentals and State of the Art, the Proposed Architecture, the Results and Discussion, and the Conclusions and Future Work.

In the section of Fundamentals and State of the Art are presented some of the background that the reader needs to understand the rest of the document. In this section are described some of the historical context of the monitors and the patient simulators and a brief description of how the vital signals are generated and prop-

agated in a human body. Aafter this, some similar projects are described and compared, in terms of advantages and limitations.

In the Proposed Architecture section, are discussed all the technical decisions made. It begins by describing the project in a major scale, explaining the interaction between all of the blocks. After this, is presented a discussion about the signal generation process and all the custom hardware needed to preform this operation. Still in this section, is explained all the decisions taken during the development of the analyser module, all of the pre, and post, process techniques, OCR algorithm and the correlation between the value read and the associated signal. Finally, is presented the user interface developed.

After this, in the Results and Discussion section, are presented the global results. In the previous section are presented some results in micro scale, and in this section are presented some tests performed to evaluate the macro scale of the project.

Finally, in the section of Conclusions and Future Work, is presented a summary comparing the goals intended with the final results obtained. Here are proposed some improvements that can be performed in the future work.

This page was intentionally left blank.

2
Fundamentals & State of the Art

2.1 Multiparametrical Monitor

As the name suggests, a multiparametric vital signal monitor is a device used by doctors, physicians, and healthcare professionals, to constantly monitor various types of vital signals of a patient.

Since the XVII century, scientists have been trying to perfect the art of measuring vital signals. It started with Santorio Venice and Galileo with a system that was capable of measuring body temperature and pulse rate. After them, Ludwig Taube, in the XIX century, was able to add the respiratory rate to the list of vital signals. Only in the transition to the XX century, scientists could build the sphygmomanometer, a blood-pressure cuff capable of measuring the pulse rate. After this, Willem Eithoven developed the electrocardiogram and techniques to execute and analyse its results.[6]

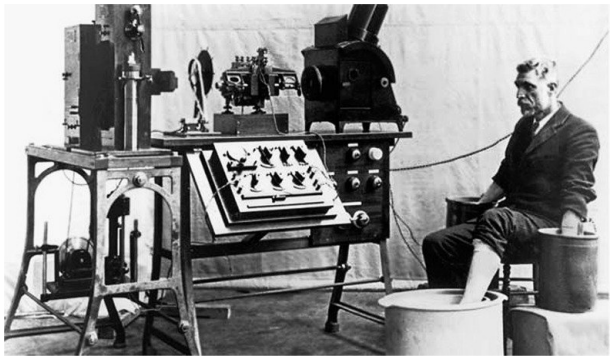

Figure 2.1: String galvanometer, developed by Willem Einthoven
Source: [2]

During the '40s decade, the first dual light source oxygen meter was invented, but only in 1995 was developed the fingertip technique, used until the modern days.[46]

Until now, all of these signals were captured independently and in an analog format. Only with the development of electronics and digital technology were healthcare professionals able to see vital signals on a high resolution display. With the development of the first microprocessors, in the 1980s, patient monitors could automatically analyze the signals and report in real-time arrhythmias. During this time,

the first battery-powered monitors were developed, making them more portable and the transport of patients much safer due to continuous monitoring.

Figure 2.2: Example of a modern multiparametric monitor
Source: [16]

Obviously, during the current century, with the development of new wireless communications, the development of the Internet, and microelectronics, these monitors have evolved exponentially. These devices can now report many more alerts and inform the healthcare professional via a cell phone or a smartwatch [39]. With all of these new data comes a new problem because, for the first time, medical professionals have too much data to process alone, so new techniques and tools are being developed using Artificial Intelligence, much faster pattern recognition, and big data techniques to help filter relevant information.[41]

Nowadays, modern monitors are capable of detecting a huge variety of diseases, and various kinds of arrhythmias, detecting the respiration rate based on the ECG signal, detecting the presence of an artificial pacemaker, analyzing fetal ECG, dosing drugs and anaesthesia. [4]

The features mentioned above are associated with direct analysis of signals. These features represent the physical well-being of the patient, but to help the patient the manufacturers of these devices had to develop some other features to help the healthcare professional. Some of these features are bigger displays and better resolutions, battery life, and portability. Some other features can help the hospital, not only the healthcare professional, such as internet connection (LAN port or Wifi), video output (VGA or HDMI), and history review stored over hundreds of hours [7]. Not all equipment currently has these features, but most modern models have many of them.

2.2 Biopotentials and Vital Signals

A vital signal is a measurable indicator of the physiological functions of the human body. When coupled with the healthcare professional's expertise, this signal may enable the assessment of the body's systems.

Vital signals can have various origins, but the bioelectric potential is a crucial concept for the study of vital signals. A bioelectric potential is a procedure generated by the body that creates an electrical polarization in excitable cells, such as nerve and muscle cells.

One function of the cell membrane is to allow the passage of some chemical elements into and out of the cell. In excitable cells at rest, the membrane naturally generates an electrical potential of approximately -70 mV. This electrical voltage occurs due to the high concentration of potassium ions (K+) inside the cell, while outside there is a high concentration of sodium ions (Na+) and chloride ions (Cl-). The following figure illustrates excitable cells at rest. [47][19]

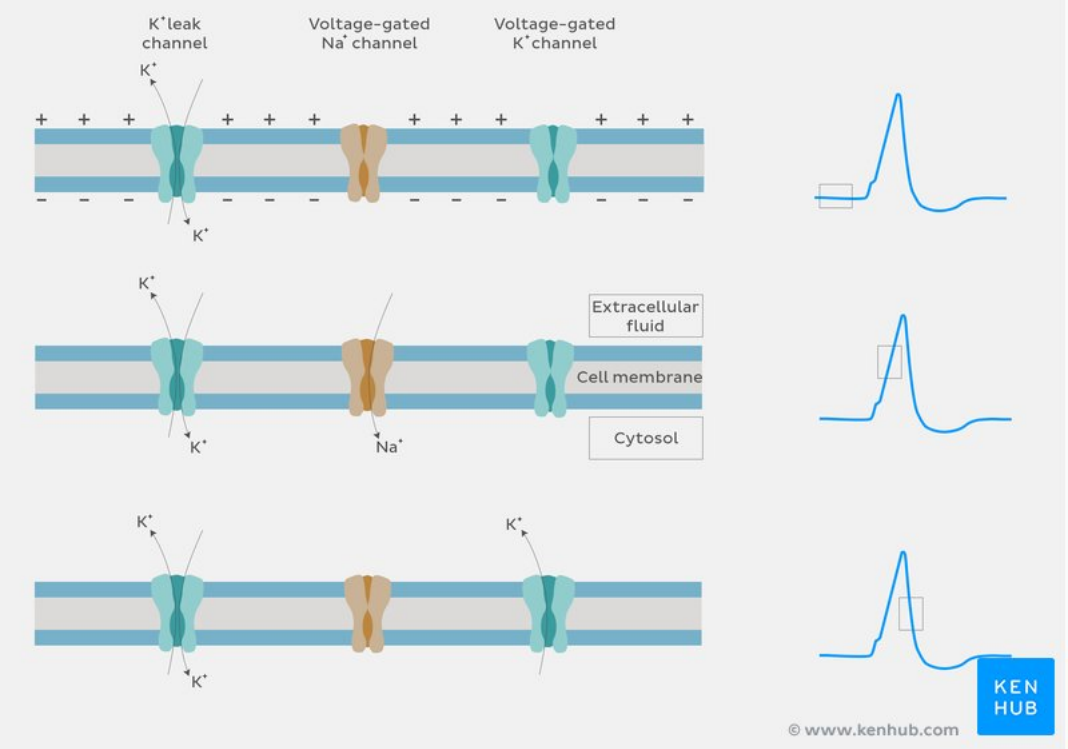

Figure 2.3: Ions exchange in action potential
Source:[43]

These cells are named for a characteristic they possess, because when appropriately stimulated, they can propagate action potentials. An action potential is an abrupt reversal of the cell's natural polarization and can be divided into 4 phases.

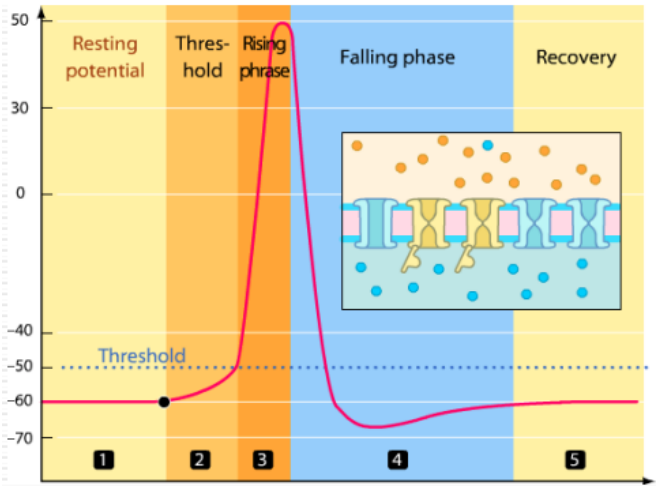

Figure 2.4: Action potential phases
Source: [1]

An action potential begins with the threshold phase, where some sodium channels open in the cell membrane, allowing these ions to enter. Thus, the cell's electrical polarity becomes increasingly positive, or less negative. If the channels close before the cell's electrical potential reaches a threshold, the cell's natural resting state returns the potential to its original state. However, if this does not happen, an action that cannot be prevented is triggered. These phenomena are called "all-or-none," meaning they either occur completely or roll back to the original state. [43]

If the cell's electrical potential exceeds its threshold, the rising phase begins. In this state, more sodium channels open, causing a sudden increase in the cell's electrical potential, as observed in the previous figure. Since the exterior of the cell is a good electrical conductor, this oscillation is propagated to adjacent cells, initiating an avalanche process.

Reaching its peak, many sodium channels close again, leaving only potassium channels open, responsible for returning to the resting state. This phase is called the falling phase. In this phase, it is normal for the cell's potential to undershoot, as this depolarization happens too fast, and practically all potassium channels are open.

Before returning to the resting state, the cell goes through a recovery phase. This phase mainly serves to prevent feedback effects from other cells.

This process has a duration of approximately 120 ms, meaning this is the minimum periodicity of actions that this kind of cells can operate.

ECG

Understanding how electrical signals are generated and propagated through the body allows us to measure and analyze some vital signals. The most important vital signal for this work is the Electrocardiogram, or ECG. As the name suggests, the electrocardiogram is used to visualize the electrical signal that passes through the heart.

The cardiac cycle is a periodical activity produced by the heart, that moves the blood around the body based on pressure changes [25]. The pacemaker cell is a

very specialized cell, responsible for generating an action potential that propagates through the heart, activating its muscles in the desired order. For an adult healthy heart, this cell generates this action potential, responsible for the heartbeat, at rates that vary between 60 to 100 beats per minute[23].

The electrical signal propagated through the heart can be measured with two electrodes, forming a dipole. Knowing this, during the transition between XIX and XX centuries, the physiologist Willem Einthoven developed the first experiments with ECG [24]. For this, he developed a system with three electrodes, plus one for reference, called The Einthoven Triangle.

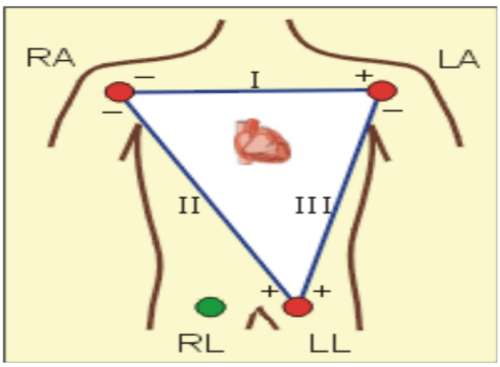

Figure 2.5: Einthoven Triangle and respective Leads
Source: [20]

As shown in the image above, the Einthoven triangle uses three electrodes positioned in each arm and the left leg and allows the measurement of the voltage across three axes of the heart during the cardiac cycle. These three axes allow the understanding of the propagation of the electrical signals across the plane shown in the figure, but during the XX century, more electrode disposals were developed allowing to visualize those signals across the three dimensions of the heart [27]. In this work will be studied the three leads proposed by Einthoven, because they are very simple to understand and easily scaled to more complex electrodes disposals.

Tracing the signal obtained in the electrodes, using the lead II of the Einthoven triangle, is obtained the following waveform.

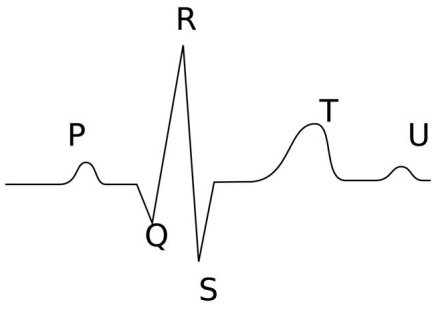

Figure 2.6: Lead II ECG waveform
Source: [18]

The above figure presents a full heartbeat, and happens during the main activity of the heart, increasing the blood pressure and causing blood to circulate throughout

the body. This waveform is notably composed of 6 peaks, nominated from P to U.

The heart starts its beat by depolarizing the atria, shown in the p wave, then causes an abrupt depolarization in the ventricles, shown by the QRS complex, followed by the repolarization of the ventricles, shown by the T wave. The U wave is not very well described and there is controversy about its origin. [45]

From those three waves, the most useful is the QRS complex, for its huge amplitude and high frequency. This abrupt variation is the main way used to automatically analyze these signals. From its amplitude, duration and periodicity is possible to detect a lot of diseases. [38]

Typically, the voltage measured across the electrodes, during the QRS complex, can reach up to 3 mV. This low voltage makes the signal very sensitive to external interference. There are some ways to preserve the signal, for example, the use of twisted pair to bring the signal into the monitor. Due to these low voltages one of the initial stages of ECG device is to amplify this signal.

2.3 Patient Simulators

As the name suggests, a patient simulator is a device that tries to simulate with high accuracy the patient's behavior. These devices are most often used for training students and healthcare professionals and to calibrate other tools, such as the multiparametric vital signal monitors. [21]

Although the term patient simulator is often used to refer to a device fully capable of simulating a patient, even with mechanical movement, in this work the concept of a patient simulator is a device that electronically simulates vital signals [22]. Mechanical simulator history is as old as human history, but electronic simulation started in the 1960 decade, with the "Sim One", a device made for anesthesiologists' training. At the same time, in parallel, other scientists were developing simulators for neurologists and cardiology.

Historically, patient simulators have been developed to help in the training, and education, of nursing and medical professionals, during the '90s decade. These devices were very limited, only used for training some social interaction between the nurse and the patient, or some simple mechanical problems associated. During the XX century, with the development of microelectronics, the monitors started evolving at an exponential rate. With this evolution rate, the manufacturers had to adapt and evolve the testing devices, starting a new era in patient simulation.[31]

Nowadays, modern patient simulators are capable of simulating all kinds of signals known, for example, standard ECG, fetal ECG, arrhythmic ECG, ECG influenced by respiration, temperature, and many others. These are all based in voltage variation, but the modern simulators can even vary the electrical current, used in the SpO2 analysis, or even pneumatic-based systems, for example, the non-invasive blood pressure analysis.[11]

2.4 Testing Protocols

Safety protocols in medical instrumentation are essential to ensure the safe operation of devices in clinical environments, protecting the patient and the healthcare professional.

The IEC 62353 [10] protocol is essential for protecting medical equipment against electrical overvoltages, such as transients and voltage spikes, which can occur due to lightning strikes or power grid faults. This protocol specifies test methods to ensure that the equipment supports these conditions without compromising its functionality or safety. Medical devices need to be robust enough to operate in hospital environments, where power fluctuations can be common, and IEC 62353 guarantees this resilience, helping to prevent damage that could cause interruptions to patient care.

The IEC 62353 protocol guarantees the proper functioning of medical devices in terms of electrical power, but there are other protocols that guarantee the proper functioning of the data. This is the case of IEC 60601-2-27 [9], for any electrocardiographic monitoring equipment, or the IEC 60601-2-25 [8], for the electrocardiograph. The devices used under extreme or uncontrolled environmental conditions outside the hospital environment, such as in ambulances and air transport, shall comply with these particular standards.

Of the three protocols presented, the most relevant to this project is IEC 60601-2-27. Within this protocol, test methods are defined for the monitors, including the injection of sine waves and the subsequent measurement of their characteristics, such as frequency and amplitude. The procedure also involves checking the monitor's alarms, as well as validating the signal in terms of frequency response. This ensures that the monitor reacts correctly to variations in the electrocardiographic signal, activating the alarms when necessary, and that the signal is processed accurately in relation to the different frequencies. These tests are crucial to ensure that the device maintains a consistent and reliable performance, meeting the safety and effectiveness standards required due to the criticality of the situation in which they are inserted.

EN 60601 is the European version of the international IEC 60601 series of standards. In Europe it is issued and published by CENELEC (European Committee for Electrical Standardization) as a European standard.

2.5 Related Projects

As this is a project that involves three areas, electronics, biomedical and software engineering, it was not possible to find any existing projects with exactly the same proposal as this one. However, several projects were studied, and some of them deserve special mention.

The following subsections present the main references used during the development of this project.

ECG Generator Projects

Two references related to the ECG signal generator are worth mentioning. Both of them uses a microcontroller, and some peripheral hardware to filter and attenuate the generated signal.

The most important, and the main reference to the generator, was the master thesis of D. Almeida with the title ”Non-intrusive ecg acquisition test-bed”[3]. In this project, the author generated the signal based on a .wav file previously stored in an SD Card, and accessed by a microcontroller. After this, this generator can perform various kinds of distortion and manipulation of the signal, for example, simulating interference between the electrode and the skin. In real-life situation, the bad contacts between these two surfaces are inevitable, and in this project the user can simulate extreme situations.

Other project that deserves a mention is a very similar project presented at the 2019 WITCON ECE.[12] This project is very similar to the previous one, with two big differences. First, this project does not seem to have the same level of versatility as the previous, as this does not allow to perform as many distortions as the last. Despite this, this project has a big advantage for commercial use, it is capable of simulating 12 electrodes, as is used professionally in hospitals.

Both mentioned projects have a common base, a microcontroller that has access to a previously stored signal. The microcontroller generates the intended signal in digital format and the peripheral hardware converts it to the analog domain, filters it, and attenuates it in order to emulate a real signal.

Signal acquisition and automated analysis

About this topic, the main reference considered during the development of the project, is a platform presented with the name ”Advanced ICU Patient Monitoring With Sensor Integration, IV Detection WITH Canny Edge Detection and ECG Monitoring With Live Feed”[37]

This project uses an external camera to analyse the monitor display and image processing techniques to perform the analysis. It has a user-friendly interface using a WEB page. This platform is capable of sending SMS when it finds an alert.

As can be seen in the following figure, this project has a major limitation.

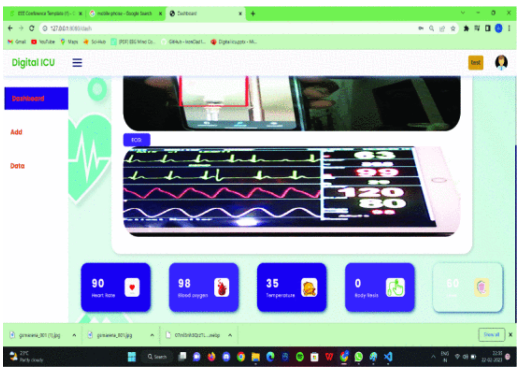

Figure 2.7: Related project interface
Source: [37]

As mentioned before, this project uses an external camera to acquire video livestream, and this causes many problems. As can be seen in the figure above, the livestream quality is very dependent of external factors, for example the external and internal illumination, the angle of the camera in relation to the monitor.

Other major problem found in this project, is the scalability of the project. In the article is not specified how the system works internally, but it seems that the system is a proof of concept, so it was developed to only work with a specific monitor, and in very controlled atmosphere.

Despite this, this project turned out to be a very useful reference, because it prooved that is possible the acquisition of all of these results.

This page was intentionally left blank.

3

Proposed Architecture

The goal of the work presented in this document is to develop a full system capable of realizing functional tests for the ECG analysis and heart rate calculations of a multiparametric monitor. The goal is also to develop a robust foundational platform that facilitates easy future integrations, enabling the analysis of additional parameters such as SpO2 and blood pressure. Another goal of this project is to develop a very easy-to-use platform, allowing non-technical staff to operate it. This chapter explains the process of developing this system.

The system architecture is divided into two primary components: the Simulator and the Analyser. The Simulator is a software module designed to generate digital signals based on data retrieved from a database. These signals are transmitted through the General-Purpose Input/Output (GPIO) pins, where an external hardware component converts them into analog form. This analog signal is then transmitted to the device under test, serving as the input for the evaluation process.

Following this, the Analyser module takes over, recording the signal output of the device under test. It then performs a comparison between the observed signal and the original signal generated by the Simulator. This comparison allows the system to assess the performance and correctness of the device under test, ensuring it operates within the expected parameters.

In addition to these core components, the system also includes a user interface module. This interface provides users with the ability to interact with the system, select specific tests to be performed, and review the results of these tests in an accessible and user-friendly manner. The user interface plays a crucial role in enabling efficient and effective use of the system, catering to both novice and experienced users.

The following diagram illustrates the described system.

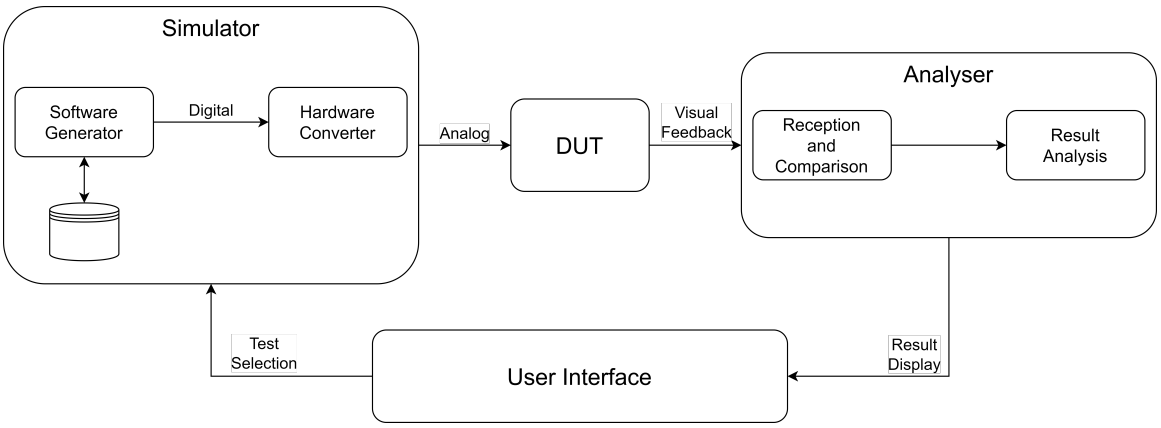

Figure 3.1: Full System Block Diagram

The figure above represents the proposed architecture. In this figure four main components and their interactions can be seen.

The main component of the project is the simulator block. Within this block, the Software Generator retrieves data from the database and creates a digital differential signal. This differential signal is passed to the Hardware Converter, which converts it into an analog format, filter it and attenuates it. After this, the final signal represents the three electrodes, so it can be passed directly to the Device Under Test, or DUT.

The output from the DUT is captured by the Analyser Module. This component is responsible for recording the signal and comparing it with the original signal generated by the Simulator. The Analyser's role is to evaluate the performance of the DUT and verify whether it meets the expected criteria.

Additionally, the system provides a Graphical User Interface, with the purpose of facilitating the use of the full system. Through this interface the user can select a specific test, from a list, and view the results. It connects to both the Simulator and the Analyser, enabling users to control the testing process and access data in a user-friendly manner.

The diagram above shows the functional relationship between the system's components, and the following sections aim to provide a more detailed description of each module.

The following sections are used to explain in more detailed form the blocks mentioned above.

3.1 Simulator Module

Of all the modules described previously, the most complex one is the Simulator. As the name suggests, the Simulator is responsible for simulating the human body, more specifically the ECG signal. Its main objective is to generate a controlled signal and it is divided into two main modules, the software module and the hardware module.

The software module is responsible for producing digital signals on data retrieved from a database. This digital signal is then passed to the Hardware Converter where it is converted to analog and attenuated.

A really important component in the development of the whole system was the precise and stable timings needed for the correct generation of the signals. To avoid distortion effects like jitter or aliasing, a microcontroller is typically used to deal with the generation, because of its rapid response to interrupts and easy integration with RTCs, and other precise timing peripherals. Due to the slow nature of the ECG signals and trying to maintain the final result as a single all-in-one solution, the microcontroller idea was put aside, and was preferred a single-board computer solution.

3.1.1 Software Generator

As mentioned before, the software generator is responsible for generating the desired ECG signal in digital form. To achieve this goal, this module can connect to a database previously populated with many signals, in a .wav format. After this, it has the objective to format it in a way that the next block, Hardware Converter, can interpret and manipulate them. This subsection is used to describe, justify and present some results about this module.

Firstly, is important to justify the previously introduced use of a single-board computer solution. There are many differences between microcontrollers and computers, but for this work, the most important were the operating system and file management system, for standard computers, and the easy access, control and interaction with peripherals (for microcontrollers).

.

When generating any signal, the sampling rate is one of the most important characteristics that have to be very accurate and stable. Slight inaccuracies in this rate result in many distortions of the final result. The aliasing distortion occurs when the signal is generated using a rating frequency slower than the Nyquist frequency. To correct this distortion is simple, it just needs to accelerate the sampling process.[5]

Other very common problem encountered in analog signal generation is the jitter distortion. The jitter effect occurs when the sampling rate is not constant, that means that some regions of the region are well sampled, while others are slowed, and others are accelerated. This effect can occur for various reasons, such as too much computer operations between samples, slowing it down, or bad timing management. Instead of the internal processor's clock, is usually used an external piezoelectric element that resonates in a multiple of the desired frequency, and using the access to the GPIO pins that microcontrollers provide, makes the sampling rate con-

sistency really easy because the use of interrupts turns this operations really fast and stable.[13]

With this argument, the obvious solution would be the microcontroller, but the previous explanation is only relevant for fast signals. As mentioned in the section about Fundamentals and State of the Art, the ECG signals are composed of low frequencies, a maximum of approximately 100 Hz. That means that the Nyquist frequency is 200 Hz, or between samples, the processor has 5 ms to process the next sample. Nowadays, most of the processors, in microcontrollers and computers, are operating at frequencies many orders of magnitude above these values, which means that any of the options is good. For modern frequencies, even using the internal timing is enough to generate accurate signals, as will be shown further in this document.

Another important observation about the signals used in this project is that they were obtained by other studies, mainly biomedical, which means that the maximum sampling rate possible is already defined, the recording sampling rate. After some investigation, most signals are not sampled at frequencies above 16 kHz. These frequencies still can be easily achieved using a microcontroller, but that also means that the generator would have a constant oversampling process. Assuming an average frequency of 12 kHz, that means that the generator are generating 120 samples (12000 divided by 100) for each period, or 60 more samples (12000 divided by 200), per period, than what was needed by the Nyquist criterion. That means that this oversampling can be reduced for, for example, 1 kHz, maintaining the oversampling, 5 more samples per wavelength than needed by the Nyquist criterion, and maintaining frequencies that any processor can operate.

Considering that the project has the objective of developing a device that would be used by any professional in medical care, would be perfect if it was an all-in-one solution or a product like a ”plug-and-play”, That means the product has to be as simple as the typical user is familiar with. Nowadays, these professionals are used to standard computers and is much more simple to interact with it if it has an intuitive display. This is a really good reason to use the standard computer solution, but the argument that really stated this option was the was the image processing requirements (used in the Analyser block, explained later). The necessity of an all-in-one solution, with easy interaction, such as an operating system with visual interaction, and image processing capability, makes the use of a computer a good solution for the standard public.

Instead of a standard computer, the solution adopted was the single-board computer, in this case a Raspberry Pi 4. This can operate as a standard computer, using various types of Operating Systems, such as Real-Time OS, or standard Linux, but the main advantage that it provides is the price, much cheaper than a standard computer (approximately 10 times), and the other feature is the easy access to peripherals, such as GPIO, or communicating modules. One more argument to support the use of this solution is the support, there are many of libraries built for microcontrollers, but none of these are compared to the Raspberry and Linux communities. There are many discussions about all the problems encountered during the development of this project, which resulted in a really fast development. The last argument is

the option to use an OS with a file system, this allows the interaction between the generator and the database much simpler. If the microcontroller solution was adopted, an entire library would be needed to interact with a peripheral SD Card, but in this case, the OS provides really simple-to-use functions in any language.

Another available option would be to use a Real-Time Operating System, also known as RTOS. An RTOS is an OS built to manage hardware and software in order to ensure that specific tasks are executed within very specified times. These types of OS are commonly used in embedded systems, like this one. This option was not pursued because, considering all the other implementation tasks defined, the additional effort required would be too great. Therefore, alternative solutions were explored. During the development and testing phase the graphical interface helped a lot in the debugging process of the Analyser Mode. These are the main reasons why it was not used, but certainly, if it was a commercial product, the RTOS would probably be a optimal option.

Now that the architecture is defined is possible to start developing the intended system. This software needs to receive a command from a user interface that will start the process, after this it interprets the command, reads from a database the desired signal and starts outputting through the SPI buses samples at a constant rate. This software has to be able to stop this interaction at any time. Another step that is important to notice is that the samples can not be sent raw, they have to be formatted in such a way that the following module, the Hardware Converter, can receive and interpret them successfully. The following figure is used to help understanding the flow that will be described in the following paragraphs.

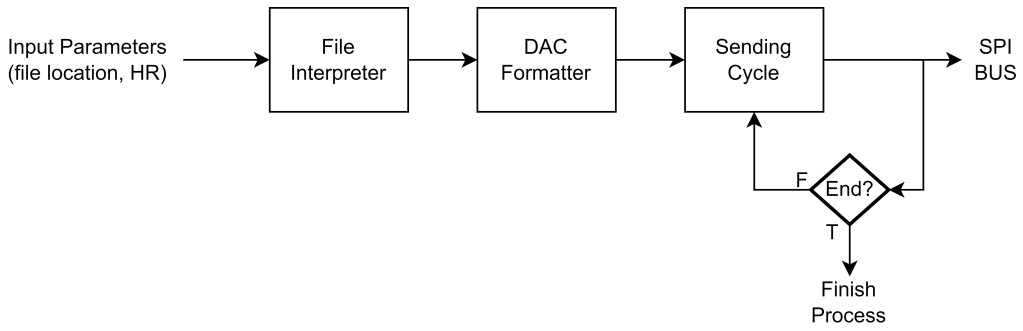

Figure 3.2: Software Flow Diagram

Starting with the communication between the user interface and this module, and given that an operating system is being used, the process can be initiated simply by passing input parameters directly via the command line. This parameter can be an integer, for example, that represents the heart rate desired to generate. To stop the Generator, the respective process can be terminated. To avoid any damage through an unintended interruption, in Linux, the program can intersect some signals, for example SIGINT, or SIGKILL, this allows for a graceful shutdown of this application. To indicate to the user interface that the process has terminated, it can just return from its main function, and the return can be interpreted.

There are some better ways to implement the communication between modules, such as pipes, or sockets. Using these OS features, the communication can be

much more complex and would be a much more robust solution, but such a complex communication system was out of the scope of this work.

About the access to a database, in this case, the file management of the OS is enough to grant easy access to any file. The files are stored in a proper folder, that can be managed in the user interface. The generator program has access to this folder, and using the input parameter it can choose which file it has to read.

After this, the library Simple DirectMedia Layer was used to interpret the audio file easily. This library loads all the bytes to a dynamic array and returns all the important information about its data, such as the full length of the array, size of each sample, signedness and endianness. Knowing all of this information about the data, a new array can be created to store all the samples well arranged. A final formatation has to be done, the chosen DACs, presented in the next subsection, need to have some special bits at the beginning of each frame. This format can be seen in the following figure.

W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
$\overline{A}$/B	BUF	$\overline{GA}$	$\overline{SHDN}$	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
bit 15															bit 0

Figure 3.3: DAC payload frame

As can be seen in the figure above, each frame is composed of 16 bits, where 12 are used for data, and the 4 first bits are used to indicate the port being used, if the output is buffered, if the DAC has to add a gain of two times and a shutdown bit.

After sending a sample, the generator has to wait for the respective sampling time. As explained before, the architecture used is a single-board computer, that means that this timing has to be controlled very carefully and precisely. The Pigpio library is a Raspberry library that allows the control of the GPIO pins and most of the hardware peripherals that can interact with it. This library can be used in Python and C language, but the Python interface is too slow for the timings required for this project. Knowing that, the entire generator was developed in C language, allowing for much faster and less distorted signals.

The Raspberry board has implemented internal peripherals for SPI communications, timers and Real-Time Clock, or RTC. As the name suggests, the Serial Peripheral Interface, or SPI, modules are used to communicate with the SPI buses, and the used library API allows for a simple integration providing very simple, and intuitive, functions such as spiWrite(), used for writing an array in the specified SPI bus. This module is really useful for communicating with the Hardware Converter, because it provides a layer of abstraction to the programmer, where he does not need to know how it works, for example, the use of DMA, is all handled inside of the library.

The timer and RTC modules seem really useful for the project, but they ended up not being used. Both are peripherals used to handle time, usually, RTC is used for timing big intervals, for example, an interrupt in 2 days, while the timers usually are used for timing little intervals, for example, an interrupt in 200 microseconds. Knowing this, the sampling rate cannot be controlled by the RTC module, but the timers can be configured to produce an interrupt and use it to send the next sample

via SPI bus. That would be possible using a microcontroller because the handling of the interrupt is really fast, but using an OS it is really difficult. Using an OS, when an interrupt occurs, the processor has to give permissions to the kernel to interpret it, after this, the kernel has to stop (or not, depending on the nature of the OS) the current process after this has to give permission to the generator process, and only after this, the program will run the communications of the next sample. This results in a lot of randomness in the intervals between calls of the interrupt handling function, producing the jitter effect commented before. The Raspberry Pi 4 has a processor composed of 4 cores, by dedicating one core to the generator this issue would be minimized because the OS wouldn't need to stop other processes and the core would be ready to start communications for the next sample. The following figures are used to understand how fast can this timer be used.

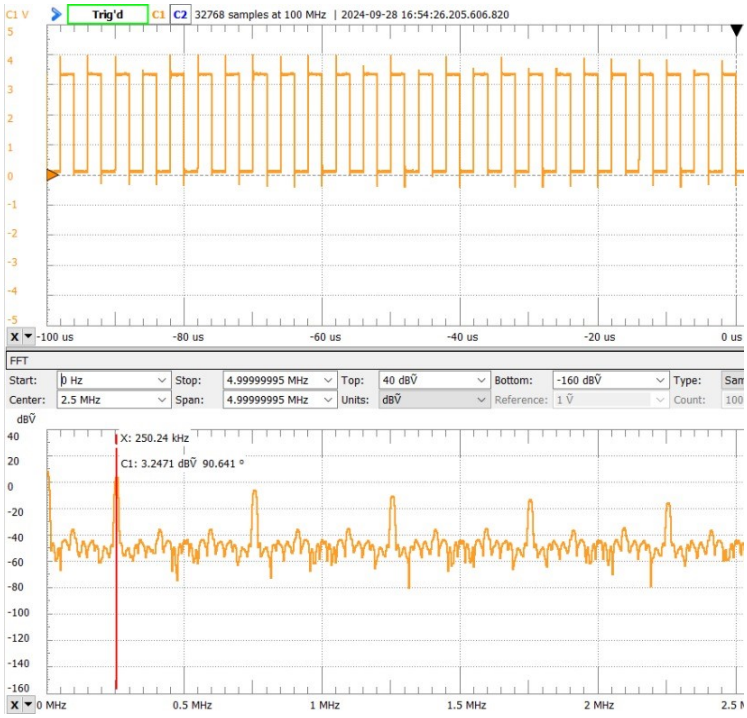

Figure 3.4: Sampling rate tests, only GPIO toggle, Real-Time (top subfigure) and FFT (bottom subfigure)

Note that to obtain these results was used the digital scope of the Analog Discovery 3, of Digilent. This tool can operate as an oscilloscope, seen in the upper half of the image, and perform the Fast Fourier Transform, or FFT, in real-time, seen in the bottom half of the figure. This oscilloscope presents a bandwidth of 9 MHz, that is considered enough to the desired tests.[14]

The figure above has been obtained with an infinite cycle that performs a transition of GPIO at each iteration. After this toggle, the cycle waits for 1 μs before the next iteration. This means that a full period of signal must be 2 μs, or 500 kHz. From the FFT, shown in the bottom region of the figure, it is possible to verify that the signal has a frequency of 250 kHz. The conclusion taken from this information is that this system can not handle such fast frequencies.

The results presented above were enough to prove that a delay of 1 μ was im-

possible to reach, but to maintain the consistency of the tests, the following figure presents the results obtained during a similar procedure, but writing 2 Bytes to the SPI controller. This was important, because, as shown in the figure 3.3, each SPI frame has the size of 2 Bytes.

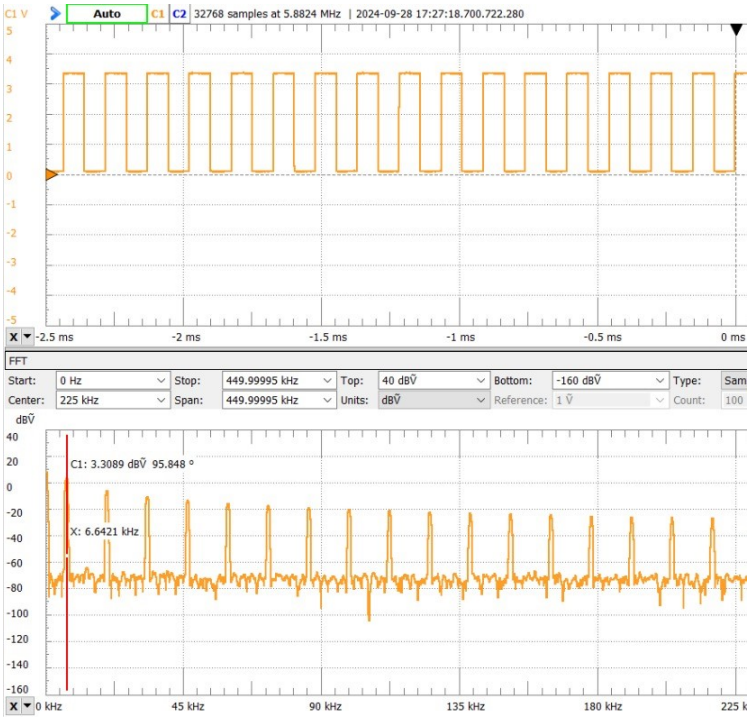

Figure 3.5: Sampling rate tests, SPI with 2 Bytes, Real-Time (top subfigure) and FFT (bottom subfigure)

The figure presented above was obtained using the same cycle described for the previous cycle, with a delay of 1 μs between iterations, but sending 2 Bytes for one of the SPI modules. Acknowledging that the toggle of a GPIO is much faster than the communications with the SPI module, than its interval is insignificant, and the frequency presented in the figure above represents the fastest approach possible for a sampling rate. The 2 bytes sent represent the data sent to the Hardware Converter, described in the figure 3.2, relative to a single electrode.

Note that the figure above presents a maximum sampling frequency of 6 kHz, but in reality, this frequency must be doubled, because each SPI communication only toggles once the GPIO, and a full period is composed of two toggle actions.

In reality, the SPI communication must send 4 bytes, corresponding to two electrodes, 2 Bytes for each. The following figure presents the results obtained.

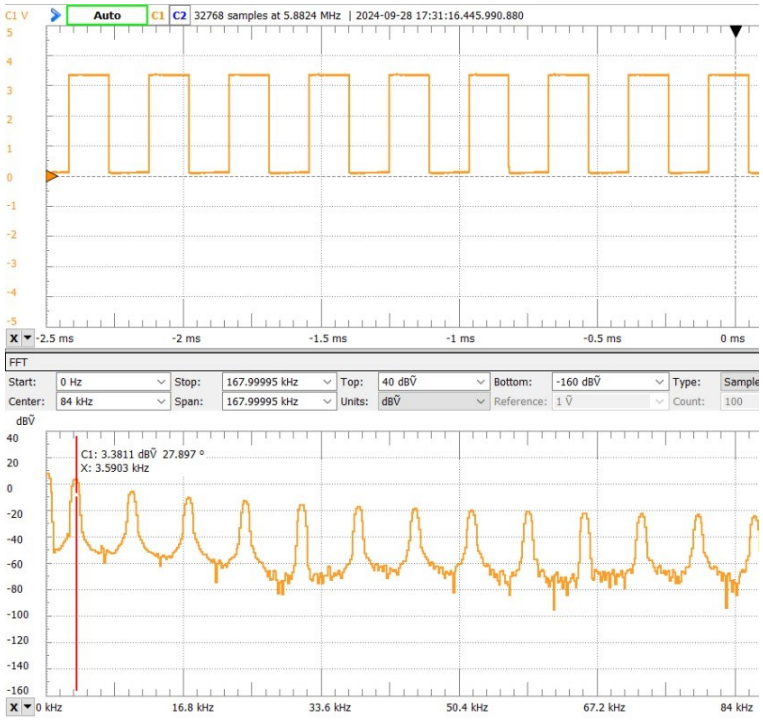

Figure 3.6: Sampling rate tests, SPI with 4 Bytes, Real-Time (top subfigure) and FFT (bottom subfigure)

Based on the figure above, it is possible to conclude that the definitive maximum sampling frequency of this generator must not be superior to 7 kHz. Remembering that the figure presents the frequency of 3.5 kHz, but it must be doubled.

Comparing this result to the previous, sending only 2 bytes, noticing that it is approximately halved, it is possible to verify that most of the time is spent during the SPI communication itself, not during the communication with the SPI module. This means that by increasing the baud rate of communication these results could be better, but that was not necessary, as will be explained in the following text.

All the results presented above were obtained using an infinite cycle with a specific delay, and the results show that this delay is not very accurate, allowing for jitter distortion. Using another function of the pigpio library, the gpioTick(), is possible to develop an infinite loop that are permanently verifying if the sampling time has passed, and if it is true, then starts immediately the communications for the next sample. This technique is called busy wait.

Usually, the busy wait is not a good option, because of energy consumption and the time blocking of the OS, but in the case of this project, neither of these are serious preoccupations. With this in mind, in the end, the system developed uses a busy wait cycle, using the gpioTick() to be more accurate in the timings, instead of the delaying functions provided.

Until now, the digital signal was only intended for the arms electrodes, but, actually, the right-leg electrode. In the Eithoven Triangle2.5, this electrode is used as a reference for all the leads in the study. This means, that this must present a neutral and stable voltage. Knowing this, the best value to be used is the middle-

point between the arms electrodes. The calculation of this value can be executed before the generation cycle begins. It can be obtained by calculating the average of all the signals.

During this calculation, other parameters can be obtained from the signal, such as the total amplitude. This ended up being used only for short calibrations during the development but it is a tool much more useful than this. It can be used to modulate the amplitude of the ECG signal, allowing for new tests, such as testing what is the lowest signal that the monitor can evaluate, or how this amplitude influences in the quality of the monitor's analysis.

3.1.2 Hardware Converter

As mentioned at the beginning of this section, the Hardware Converter is responsible for converting the digital frames that Software Generator produces into an analog form, simulating a human body.

The hardware diagram is presented in the following figure.

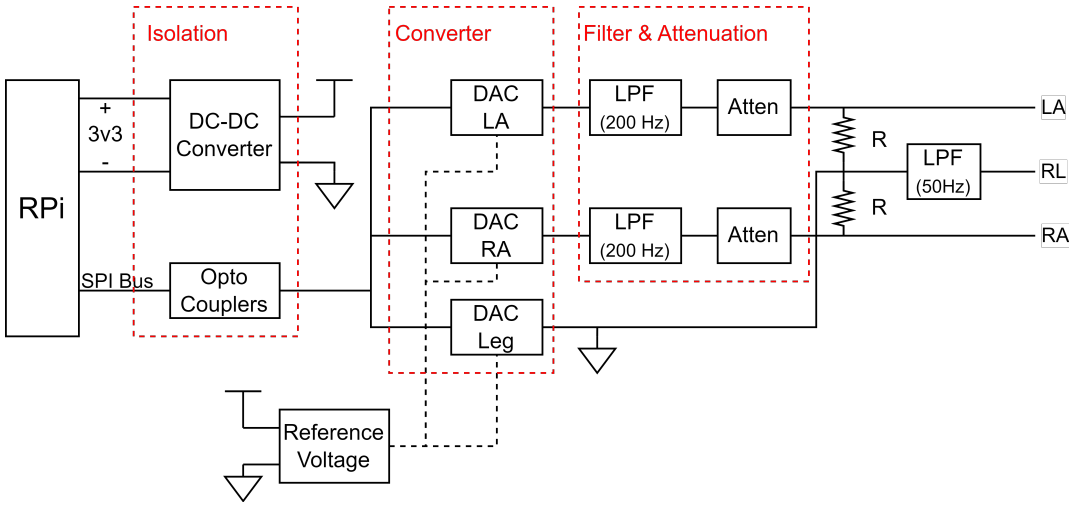

Figure 3.7: Hardware Block Diagram

As can be seen in the figure above, the hardware developed is composed of three main regions, first region is the isolation, followed by the conversion, and ending with a filter and filter.

Digital Format

To explain the necessity of the isolation region, it is important to mention that some monitors manipulate the right leg electrode to induce variations in the remaining electrodes. As can be seen further in this chapter, the right leg electrode voltage is used as an intermediate voltage, allowing to produce the differential signals in the arms electrodes. In a human body, when the monitor varies the right leg voltage, the entire body varies in a same way. It means that the entire hardware has to be sensible to this variations. In contrast, the raspberry GPIOs must not be influentiated by this variations. That means that both circuit can not be in a direct contact.

As can be seen in the figure above, in figure 3.7, both circuits are only connected by supply voltage, and the SPI bus. Knowing this, the solution developed to isolate both circuits uses a DC-DC converter for the supply voltages and fast opto-couplers to the communication bus. Note that the selection of each component must consider its application, for the DC-DC converter, it must be able to handle the desired voltages, and the opto-coupler it must be able to function faster than the baud rate used in the SPI bus. Considering that, the DC-DC Converter used was the *Hi Link B0305S* [15] and for the opto couplers were used the *Vishay 6N137*[44], that allows up to 1 MBd, or 1 million symbols per second.

The conversion region is composed by Digital-Analog Converters, or DACs. As the name suggests, the DAC is responsible for converting a digital signal to an analog format. In this project the DAC used, the *Microchip MCP 4922*[29], uses an SPI interface to interpret the digital signal, and control the internal peripherals, as the DAC, output buffer or gain.

The integrated circuit chosen is a dual channel, it means that it provides two DACs. In this case, the circuit has to generate two arm electrodes and ensure the right leg voltage is kept as a middle voltage on the isolated region. For this, was used two of the reffered integrated circuit, where one is responsible for both arms DACs, in figure 3.7, and the other is used to ensure that a stable voltage is applied, and can be programmed.

A DAC has an important pin called reference. This pin is used to set the maximum voltage that the DAC can provide at its output and this voltage needs to be as stable and consistent as possible. Recognizing this, a solution using a voltage regulator was used. Firstly, the circuit was developed using, as a reference, the supply voltage, but that resulted in problems in fluctuations of the reference voltage, due to the variation in the right leg electrode produced by the monitor. To avoid inserting a reference voltage greater than the supply, was used a 3.3 V voltage regulator, using a supply voltage of 5 V[42]. Due to a problem with voltages near the positive supply voltage, in the filter and attenuator, that is explained later in this document, this drop of 5 to 3.3 V was really helpful and avoided it.

To show evidences that both regions, isolation and conversion, functions as it was intended, the following figures present the generation of a sinusoidal signal of 100 Hz, sampled at 12 kHz.

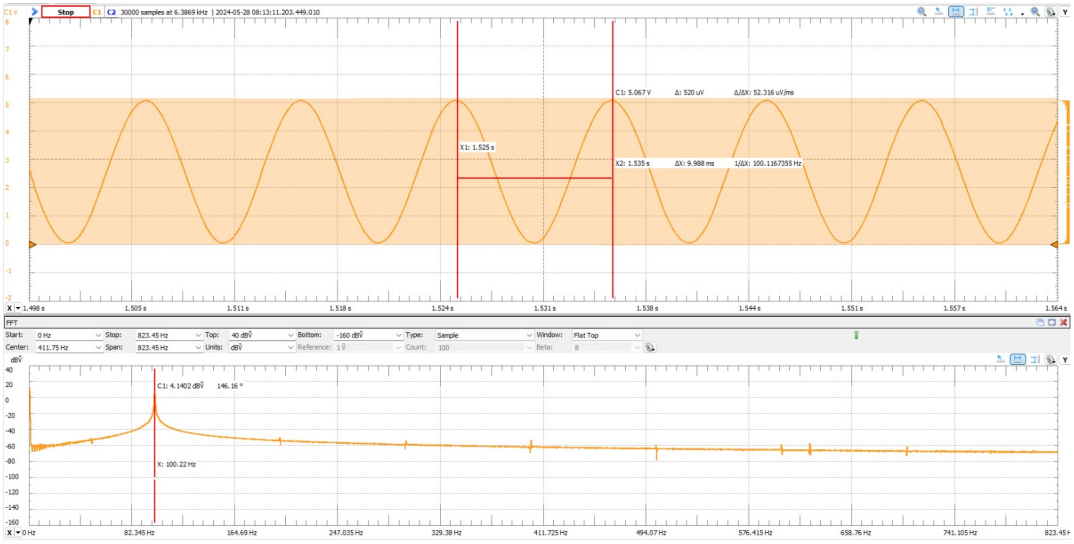

Figure 3.8: DAC tests, FFT with central frequency, Real-Time (top subfigure) and FFT (bottom subfigure)

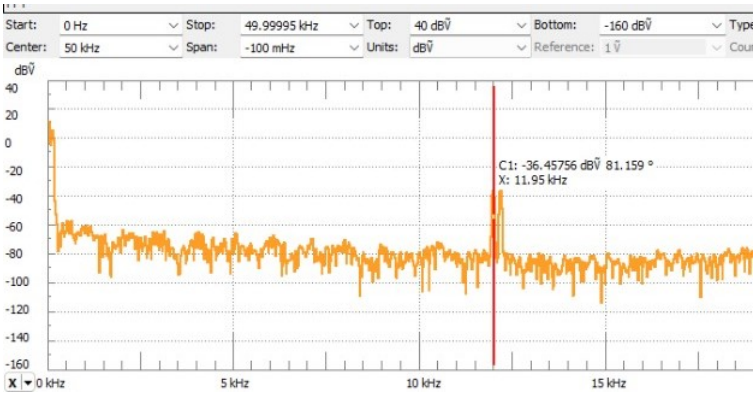

Figure 3.9: DAC tests, FFT with sampling frequency

As can be seen in the figures above, the frequency of 100 Hz was achieved with an error of 0.22 Hz, equivalent to 0.22%. In the figure 3.9 is clearly seen a peak near the 12 kHz frequency. This peak is the main reason that made necessary the existence of a low pass filter.

Before the filter and attenuator development, is important to mention that in the figure above is not clear, but at the 12 kHz, does not exist only one peak, exist two peaks centered at 12 kHz and spaced exactly 100 Hz. That would need more tests to understand completely this effect. This did not result in any problems due to the oversampling, meaning the 12 kHz is one hundred and twenty times bigger than the maximum frequency present in the generated signal.

Analog Format

In the end of the subsection above is shown how the digital signal can be converted to its analog format, but it still not perfect. Two main problems exist with the actual signal, the first is the significant presence of the sampling frequency and the second is the voltage amplitude.

About the sampling frequency, the audio files used are recorded with sampling frequencies between 8 kHz and 16 kHz, but this is not the final sampling frequency, because the software can not handle this speeds, so it reduces this frequency to 1 kHz.

For the low-pass filter, knowing that the highest relevant frequency present in an ECG signal is 100 Hz and the sampling rate is at 1 kHz, than the cutoff frequency does not need to be very precise. So, due to that, a cutoff frequency of 200 Hz was chosen. For the type of filter, there are various types to choose from, for example Bessel, or Chebyshev, each one produces different effects on the signal, but knowing that this frequency does not have very restrictive limits, the chosen type was a Butterworth.

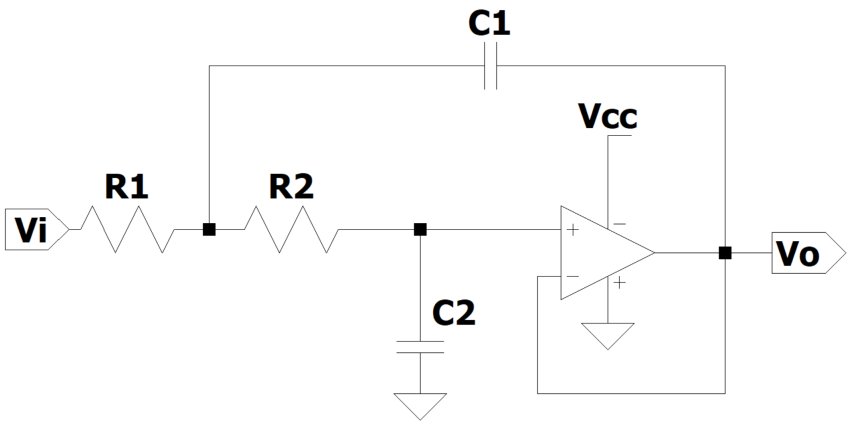

Figure 3.10: Proposed circuit for filter[17]

The equation that describes the circuit above is the following

$$H(s) = \frac{\frac{1}{R_1 R_2 C_1 C_2}}{s^2 + s\left(\frac{1}{R_1 C_1} + \frac{1}{R_2 C_1}\right) + \frac{1}{R_1 R_2 C_1 C_2}} \tag{3.1}$$

Solving this equation for a cutoff frequency of 200 Hz, is possible to verify that the components must follow the values 11 kΩ for both resistors and 100 nF and 200 nF for the C_1 and C_2 capacitors. The resistors used follows the E12 series, so the real value used is 12 kΩ.[17]

The following figures shows the result obtained for different frequencies.

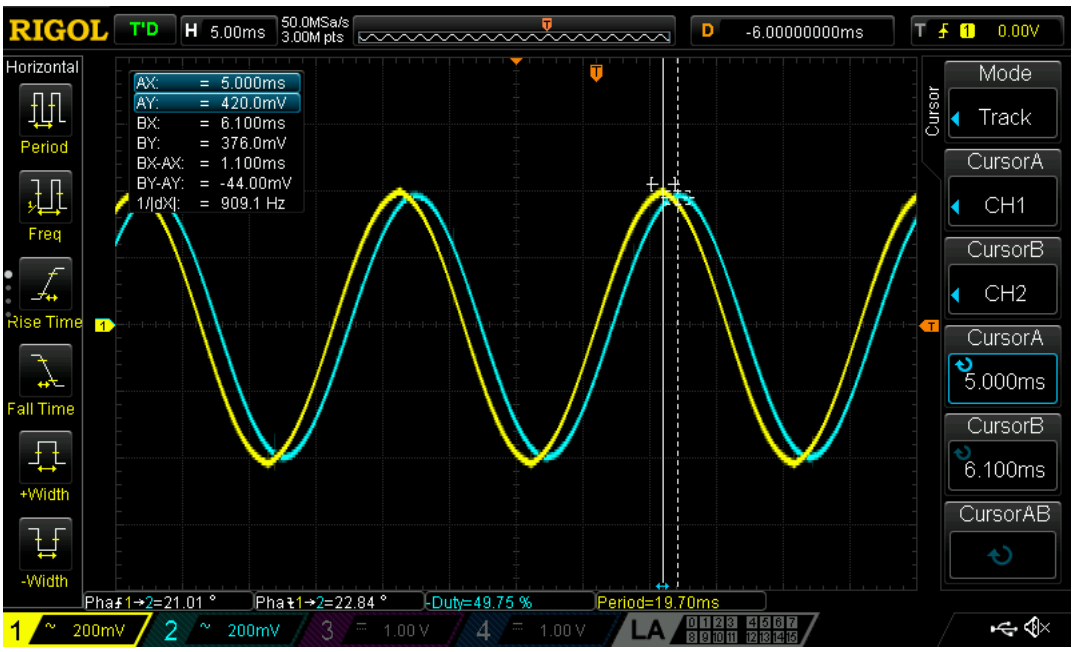

Figure 3.11: Butterworth Filter tests, input frequency 50 Hz, yellow for input and blue for output

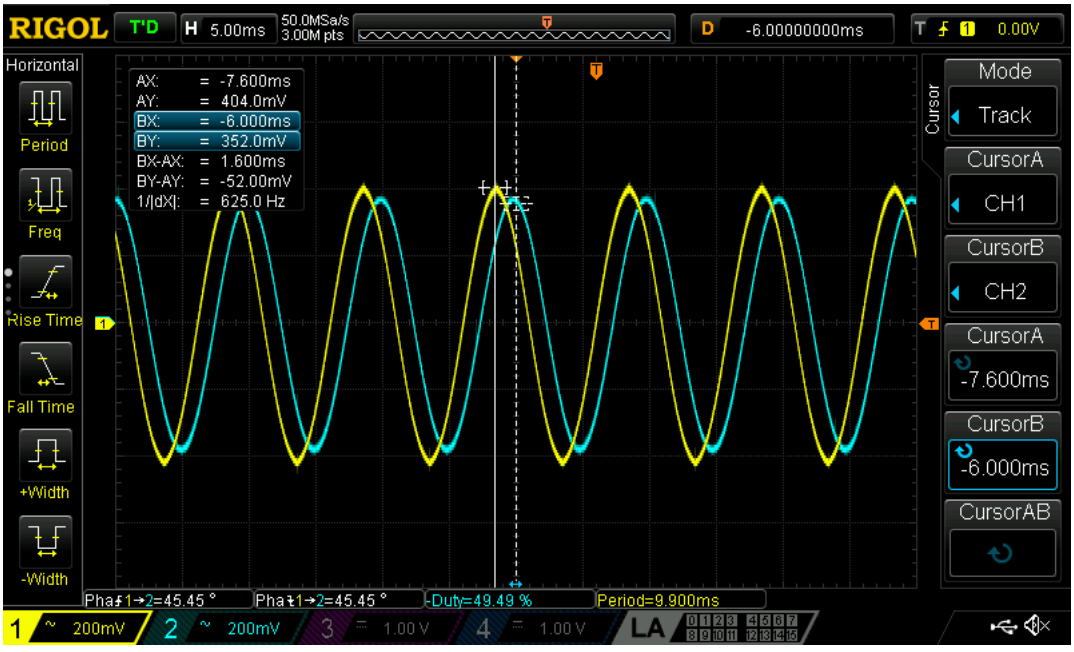

Figure 3.12: Butterworth Filter tests, input frequency 100 Hz, yellow for input and blue for output

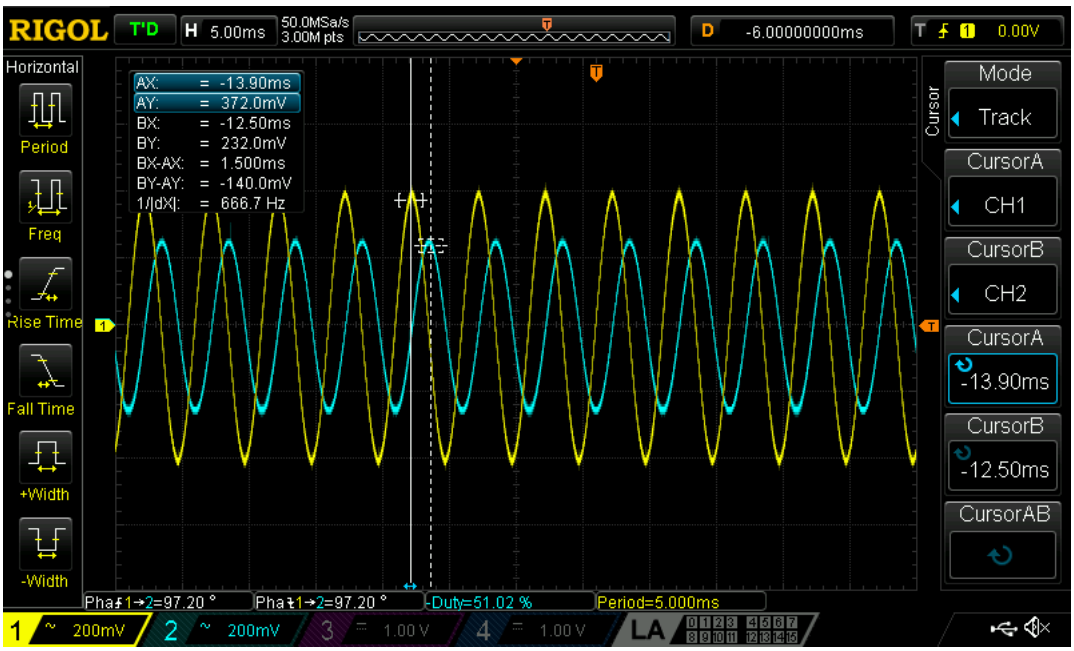

Figure 3.13: Butterworth Filter tests, input frequency 200 Hz, yellow for input and blue for output

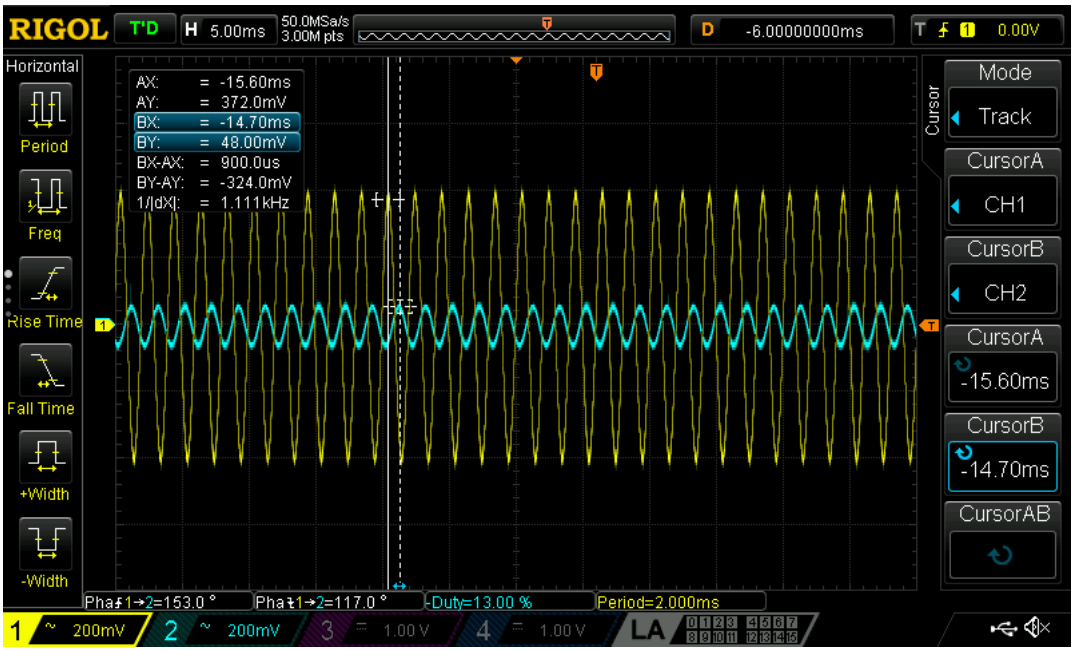

Figure 3.14: Butterworth Filter tests, input frequency 500 Hz, yellow for input and blue for output

Note that for all the figures presented above, the yellow signal represents the input, in yellow, and the output, in blue, of the filter.

As can be seen in the figures above, for the frequencies present in the typical ECG, 50 and 100 Hz, the filter does not affect significantly the signal, and for the high frequencies, 500 Hz, the signal is attenuated significantly. For the cutoff frequency, at 200 Hz, is seen some attenuation, approximately 0.6 times. The value expected was -3dB, or 0.71, but this imprecision is not relevant.

This way, the high frequency noise produced during the conversion, from digital to analog, has been attenuated. Now, the signal is very similar to a real ECG, except for its amplitude, and to solve this issue, an attenuator is used.

As can be noticed, since the output of the DAC, in figure 3.9, the signal can have a maximum amplitude of peak-to-peak of 5 V. As mentioned in the theoretical chapter, the typical voltages of an ECG signal are around units of millivolts, that means that this is the range of amplitudes that the monitor is dimensioned and that if such high voltages enters the monitor will not be analysed, and, in worst cases, can cause physical damage to the monitor.

The most simple attenuator possible is a voltage divider circuit, presented in the following figure.

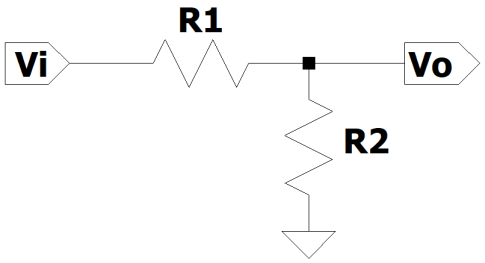

Figure 3.15: Voltage Divider circuit

The output voltage of this circuit follows the next equation.

$$\frac{Vo}{Vi} = \frac{R_2}{R_2 + R_1} \tag{3.2}$$

Although this is one of the most simple circuits with electronics, with only two resistors, it is, nevertheless, suitable to perform the attenuation on an analogue signal. This circuit only presents one disadvantage, the load effect, which means, its attenuation is highly dependent on the input impedance of the next stage. Knowing that the following block is the monitor, and probably the first stage of it is a differential amplifier, it means that is a large input resistance, but just to be sure that there is no load effect to affect the signal, the proposed circuit to the attenuator is the following.

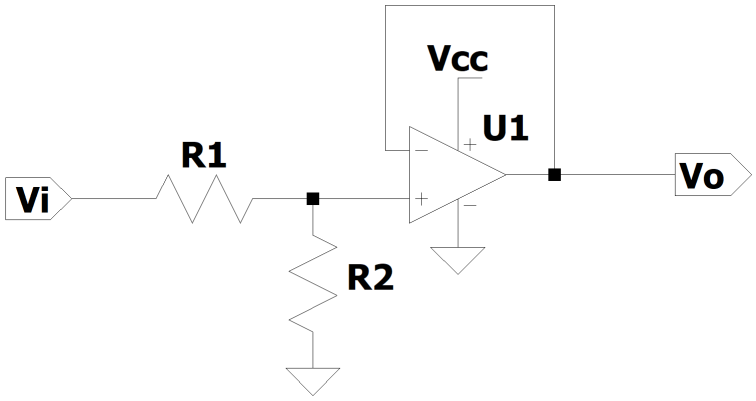

Figure 3.16: Proposed attenuator circuit

In the circuit above, a buffer was implemented after the voltage divider. The buffer circuit is characterized by its high input impedance and low output voltage.

As mentioned before, this attenuator is dimensioned to attenuate 3000 times the input voltage, passing it from units of volts to units of millivolts. Such low voltages are almost impossible to see in a standard laboratory, without amplification, because it is in the same range as the noise level. Due to this, was impossible to get any figure to show the well behaviour of this circuit. But in the Results chapter are presented the final results, demonstrating the effective performance of this module.

3.1.3 PCB Design

When working with low voltage analog signals, the noise level becomes a significant problem because these signals are very sensible to external interference, resulting in distortions. To minimize this problem, a dedicated Printed Circuit Board, or PCB, has been developed. A good PCB design have a direct impact in the quality of the signal.

Considering that for the Software Module was used a Raspberry Pi 4 and that the goal of the project is to have an all-in-one product, then dimensions of the PCB became very clear, it must be dimensioned to fit as a Raspberry hat.

The first step to a PCB design is the design of the schematic. A schematic is a graphic representation of the circuit, it shows how the various components are interconected, but it is not the physical layout. In the schematic some simple components, as resistores or capacitor, are represented using the standard representation, but not all. In this case, the OpAmp used are embedded in an integrated circuit, that means that, in the schematic, the OpAmp symbol does not appear, only the integrated circuit, with all of its pins.

The following figure represents the schematic developed.

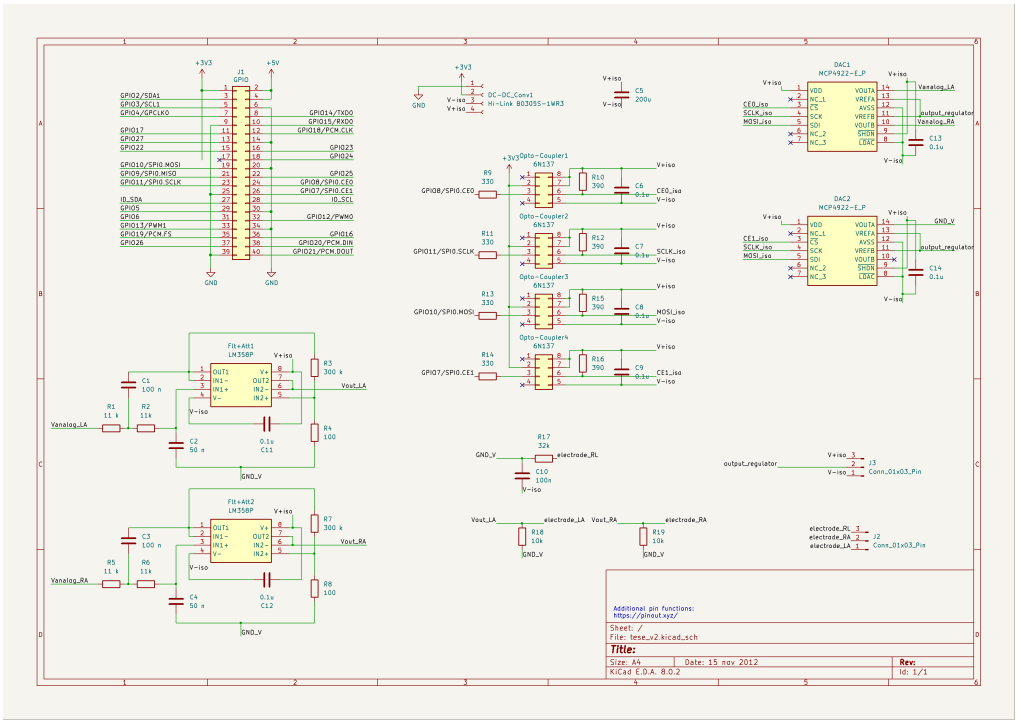

Figure 3.17: PCB Schematic draft

Note that for a much clear view of the figure above, it is present in the Appendix section at the end of the document.

Acknowledging the pinout of all components, it is possible to verify that the circuit described in the figure 3.7 and the one above are identical.

After the schematic design, the next step is the PCB layout, that is where are defined where the components are physically located in the final board.

The footprint of a device is its representation in a PCB. It includes the size of the component, dimensions of mounting holes and copper pads, where the components are mounted and soldered. During the PCB layout phase, each component has a specific footprint.

The final layout of the PCB can be seen in the following figure.

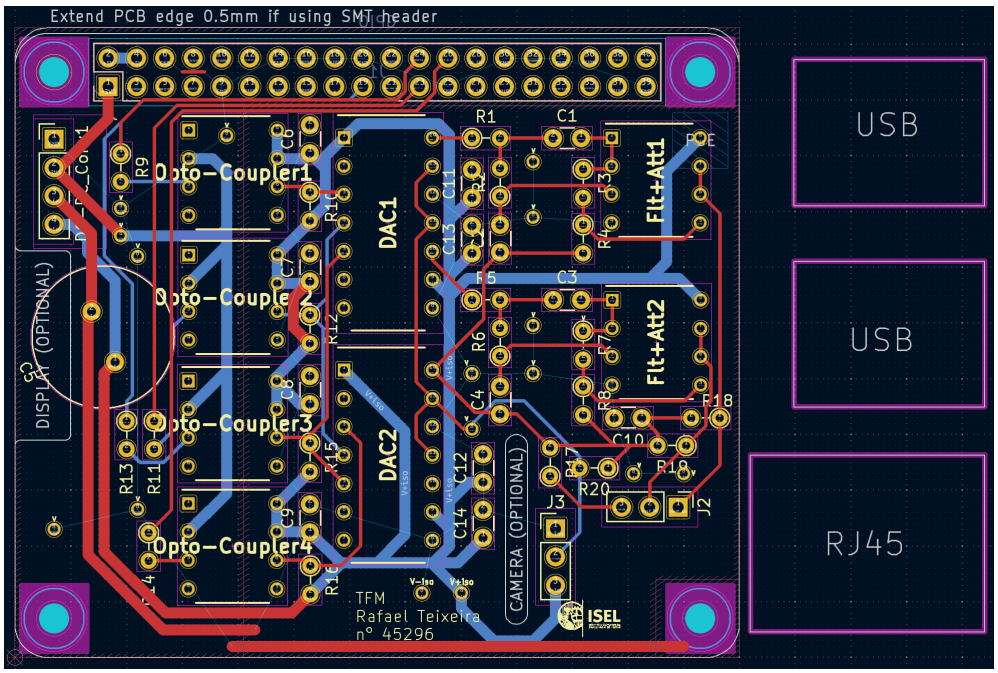

Figure 3.18: PCB Layout draft

Note that for a much clear view of the figure above, it is present in the Appendix section at the end of the document. Another important note is that some traces seem incomplete due to the filling zones that are not represented in the figure, but explained further in this section.

In the figure can be seen all the components used in the final result. In the upper section are the Dupont pins used to communicate with the Raspberry Pi, and in the corners are 4 holes designed to attach this board to the Raspberry Pi and grant much more physical stability, instead of using only the Dupont pins.

The blue and red lines are the PCB traces. in this case, this PCB have only two layers, front and back, so each color is used for each layer. Some traces are clearly thicker than others, those are power lines. They are thicker, because they have much less impedance, and often they have a significant amount of current passing through them. The thinner lines are used for communication and analog signals.

In this case, all footprints used are *through hole*, that means that it needs a hole passing through all the existing layers. Other very used technology is called Surface-mount Device, or SMD, where the copper pads are much smaller, it does not need a hole through all layers and the component is, typically, much smaller, about 5 times. Comparing both technologies, a very clear conclusion is that the SMD is better in terms of area, due to the size of the components, and because it allows much more space to pass traces beneath the big components.

In this project, *through hole* technology was used due to the ease it provides in the process of error identification and correction, as will be shown more ahead. Another advantage of this technology is the soldering process, the through-hole technology turns this procedure much easier, when compared to the remaining options. Another reason for using this technology was the ease of printing, allowing it to be printed at the university campus.

As mentioned before, the filling zones are not represented in the figure 3.18. A filling region is an area that is fully covered with copper. These regions are usually used to spread a very useful voltage, usually ground, through the whole layer, and facilitating its access, using vias. In this PCB were used two regions, one covering the Raspberry pins and the left side of opto couplers, and other covering the right side of the opto couplers. The opto couplers and the DC-DC converter were used to isolate the grounds, from raspbery and voltage from the right leg electrode, and using the filling zones any part of the circuit has an easy access to the respective reference. The separation between both regions can be seen in the vertical red line passing through the opto couplers.

Now that all PCB is explained, the following figures present the board after printed.

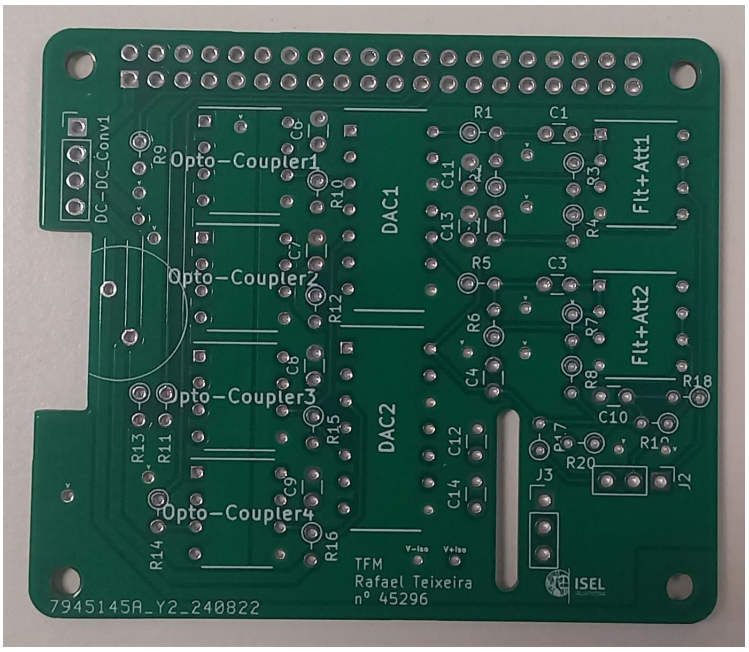

Figure 3.19: Final PCB without components

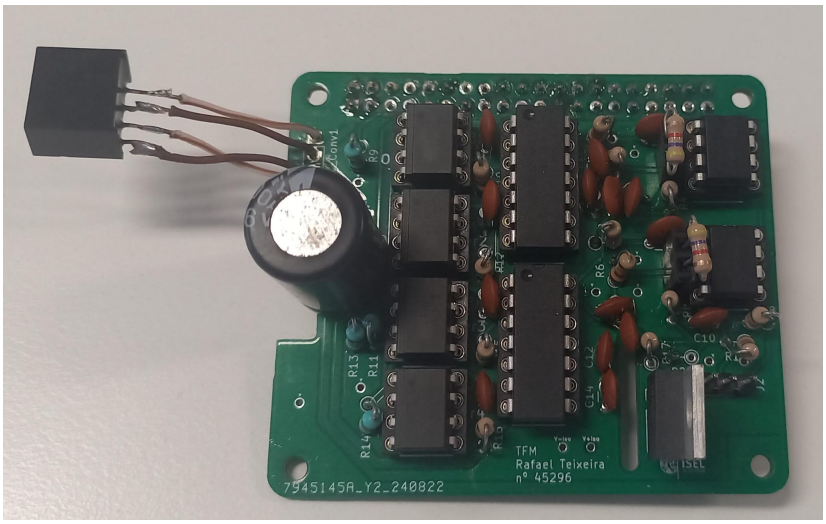

Figure 3.20: Final PCB with components

Is important to mention that the PCB shown in the figures were not printed from the figures 3.17 and 3.18. the printed version was a prototype version, and some errors were encountered and corrected for the explanation in this document.

As it is a prototyping board, the integrated circuits are not soldered directly in the board, sockets were used to help on the debug procedure. Another important tool for debugging was opting not to solder the resistors responsible for the attenuator directly. Although they were theoretically calculated in the previous section, inaccuracies in the components during the physical implementation can affect the overall behavior of the circuit. To address this, a female Dupont socket was used, allowing for easy adjustment and fine-tuning of the attenuator.

3.2 Analyser Module

This section describes how a program can receive an RGB image and extract a numerical value that represents the measured heart rate. The proposed structure is described in the following diagram.

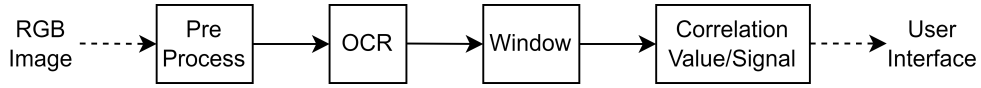

Figure 3.21: Analyser block architecture

As seen in the diagram above, the edge computer uses 4 main stages, a pre-processing of the image, then an OCR algorithm, and, finally, an algorithm to associate the string with the signal's name with the corresponding value.

All of the following sections will describe how the proposed system was achieved. Similar to the previous block, but for different reasons, the efficiency of the system is one of the biggest concerns. In the previous block, the timings were very important because it could produce many distortions in the resulting signal, but in this case, the analyser needs to be very efficient because it is composed of various heavy computational operations.

Although the full operation has to be quick, this module was developed in the Python language. This decision is typically very slow when compared to a compiled language, but the main libraries used in this module are written in C, or C++, so the Python interpreter does not present a big concern in terms of efficiency, since the heavy operations are performed in compiled languages.

Pre-Processing

As will be explained later, the OCR is a very heavy computational operation, so for this, typically, are applied a pre-process techniques to the input image. These pre-process techniques are used to minimize the size of the image, or to shade some noise present in the image. These alterations make the results of the OCR algorithm much more reliable.

First of all, is important to describe what is an RGB image. An RGB image is a 2D array of pixels, where each pixel has three bytes representing the 3 primary colors of light, Red, Green and Blue. Being the primary colours means that all other colours, in the visible spectrum, can be formed by a combination of these three.

As mentioned before, each pixel occupies three bytes of memory, one byte per colour. If all the colors have the same value, then the generated colour is grey, and its tone depends on the quantity used. If this value is minimal, then the resultant grey is the darkest possible, or black, and if it is the highest possible, then the resultant colour is the brightest grey or white. Since a grey pixel has a double redundancy, an image fully covered with grey pixels can be easily compressed three times. This procedure is called grayscale conversion and is commonly used in image processing, allowing the computational operations to be performed much faster.

If the image under tests is not fully covered with gray pixels, then it has to be converted to grey, before the compression. This conversion is described by following expression.[32]

$$G = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{3.3}$$

After these processes, the image is much lighter, compared to the initial, and has a good contrast between the characters and the background, but highlights another problem. The original image has noise distributed through all the layers, but after this, all layers are converted into a single one, amplifying the noise. To remove the noise is applied a blur in the image. The blur can remove the high frequency spatial noise, smoothing out most of the abrupt variations caused by the noise.

Exists various types of blurs, but all of them are based on a simple principle, applying a kernel convolution. A kernel is typically a small matrix, for example, 3x3, or 5x5, that passes over the input image and, in each pixel, applies a transformation. Typically this transformation is only a dot product between both matrices.

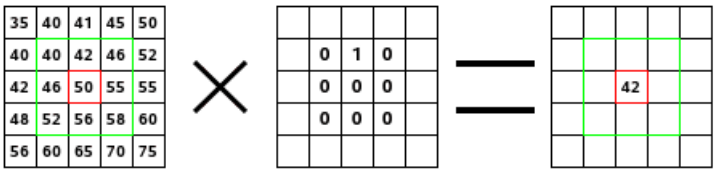

Figure 3.22: Convolution Algorithm applied to a single element [34]

The most simple blur is the median blur, where all the pixels of the kernel are composed of the same value, resulting in a median of them all. Variations on the distribution of values on the kernel produce different results, for example, if the distribution follows a Gaussian curve, centered in the center pixel, then this kernel applies a Gaussian Blur. This variations are very important for other types of image processing, but in this project was used a simple Median Blur.

The matrix convolution is a very heavy computational operation because, for an image NxM and kernel AxB, this operation performs a total of AxBxNxM multiplications. This is one of the main reasons to use kernels of small size, as mentioned before, but extending the kernel dimensions allows for various types of smoothing effect. This effects were not necessary in this project, so the kernel chosen was the smallest one.

The matrix convolution, if it is not using padding, performs compression on the image, but that is very negligible, and that is not its main purpose so that is not typically referred to.

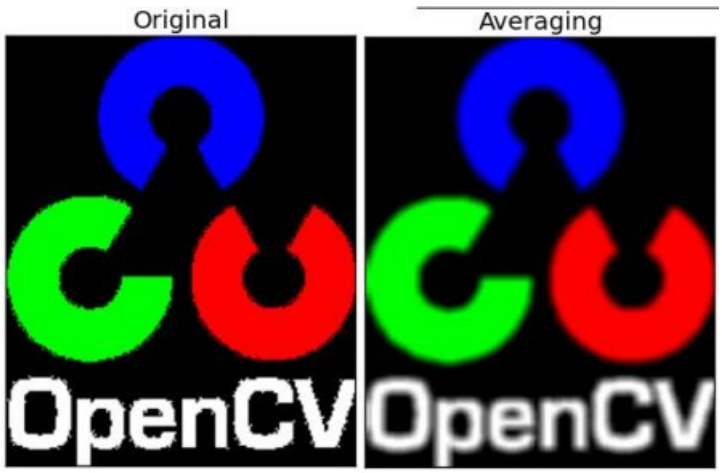

Figure 3.23: Blur applied as a noise removal [33]

Note that, the figure above is presented as RGB, but it is only used for demonstration purposes. In the project, the bluring technique was used in a grayscale image.

After the grayscale conversion each pixel of the image is composed by one byte, but that can be even more compacted. The binarization technique, or thresholding, consists in a transformation of a value between 0 and 255 to a new boolean value of 0 or 1. For this, a threshold level is defined to calculate the new pixel value, for example, if the pixel is higher than 128 then it converts to 1, then it converts to 0. Theoretically, this conversion compacts the image 255 times, but in high-level languages, for example Python, this compression may not be correctly applied, due to the lack of low-level control. Although this conversion does not produce a huge difference in size of the image, this is really helpful in increasing the edges and contours.

Although the threshold technique is very effective, there are cases in which it will not fit the problem. In cases where the background colour is very similar to the object that needs to be detected, a static threshold can solve the problem, but that value may not be in the middle and has to be configured manually by the user, or using some other algorithms like the Otsu's algorithm. [36]

The histogram of an image is a graph where all the occurrences of a colour are counted. Displaying this on a graph makes it possible to visualize some peaks, and distinguish the background colour from the object's colour. Otsu's algorithm uses these peaks to find the best threshold for each image. Applying this algorithm to different parts of the image can be useful in cases where the brightness levels are not uniform along the image.[26]

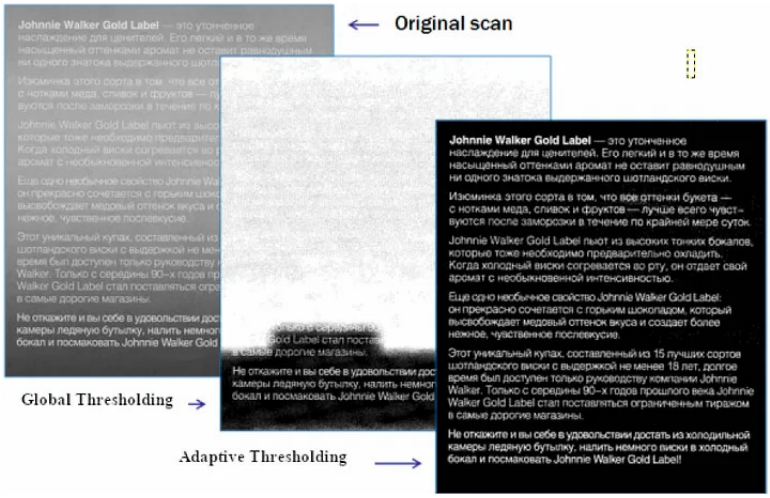

Figure 3.24: Global and Adaptive Threshold comparison

Source: [35]

As can be seen in the above figure, choosing a good binarization process for the image is very important to perform a good OCR. This decision must be adapted for each project, depending on the input image expected. In the context of this project, the

The following figures present the various steps applied during the pre processing.

Figure 3.25: Frame before pre processing

Figure 3.26: Crop example

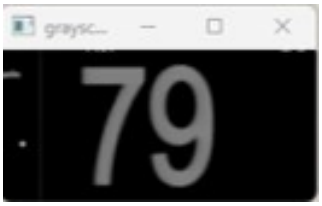

Figure 3.27: Grayscale example

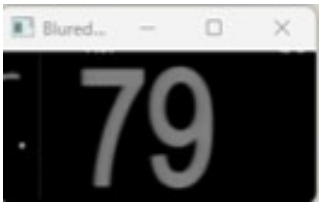

Figure 3.28: Blur example

Figure 3.29: Binarization example

The figures presented above how the image complexity can be reduced, step by step. Obviously, it is perceptible the reduction of complexity between the original frame and the cropped version, or between the colored version to the grayscale, but comparing the grayscale with the blur does not exist much of a visual difference. In this case, the display has low resolution and good contrast, and the image is being taken directly through a video port, but if it was being taken from an external camera this would be a very important procedure. From an external camera, the noise present in the original frame is much more significant.

Comparing the original with the binarized figure, the compression is evident, compression in total size, that reduces significantly the convolution opperations, and the size of each pixel, that has been transformed from a colored pixel that occupies 3 bytes to a single bit, theoretically. As explained previously, this level of memory management is almost impossible to grant in Python, but it still maintain a significant level of compression.

OCR

Optical Character Recognition, or OCR, is a system developed to read an image, recognize if any characters are displayed and translate them into machine-encoded format.

OCR is a very useful system and is being used in various areas, with the objective of digitizing all the data in the world. With a digitized form, the data turns out to be much more easy to store, preserve and search.[28]

Nowadays, there are various types of sectors using OCR techniques to effectively transcribe some handwritten, or computer-written, documents to easier formats such as pdf. Some of these examples are the organizations that have to deal with huge amounts of physical documents, such as commercial or financial, and are very useful in the educational sector for transcribing historical documents and making much easier the research procedure. In the medical sector, this technique is very useful for granting more precision and accessibility to patient data, but can be used to read the information in various physical devices and making it easier for their integration, as is entended in this project.[30]

One of the most famous OCR engines is Google's Tesseract-OCR. This engine supports various kinds of input image formats, such as PNG, JPEG and TIFF [40] [48]. Other aspects that influenced this decision was some previous personal experiences with this engine and the easy access to documentation and helping material, because of its active community.

Window

As the OCR algorithms evolve, they are becoming much more effective and accurate, but it is not nearly the desirable. The modern OCR algorithms still present inaccuracies in readings, missing some characters, or making mistakes, confusing them.

Applications of OCR in real-time have to avoid this type of error. Fortunately, this problem can be easily solved with a technique called windowing. This technique is based in a temporary buffer that is filled with samples at each frame processed, and, when it is full, it calculates an average and store it, finally the buffer can be flushed and ready for the following frames.

In this case, the output rate of the window is reduced proportionally to the size of the buffer. In some applications, the output rate is very restrictive, so, to avoid this reduction, a sliding window must be implemented. A sliding window is similar to the previous technique, but instead of flushing the whole buffer, each sample is shifted, deleting the oldest and remaining one empty slot. The empty slot can be occupied by the next frame, and the cycle repeats.

The following figures present the results obtained after the OCR algorithm and window.

```
window processed with values: [82, 82, 83, 82]
Average of filtered samples: 82

window processed with values: [82, 82, 82, 82, 81]
Average of filtered samples: 82

window processed with values: [82, 82, 83, 83, 85]
Average of filtered samples: 82.5

window processed with values: [86, 86, 89, 90, 90]
Average of filtered samples: 88.0

window processed with values: [91, 91, 91, 90, 90]
Average of filtered samples: 91

window processed with values: [90, 90, 88, 88, 87]
Average of filtered samples: 89.0

window processed with values: [86, 86, 86, 86, 86]
Average of filtered samples: 86

window processed with values: [88, 88, 90, 90, 90]
Average of filtered samples: 90
```

Figure 3.30: Windowing demonstration

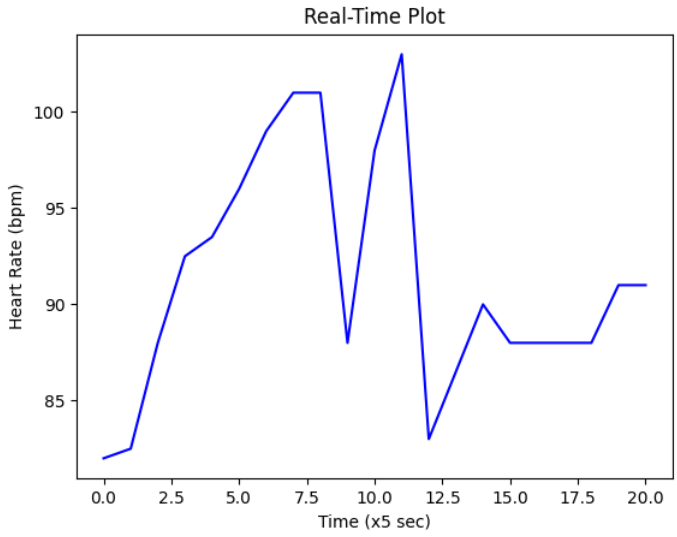

Figure 3.31: Real-Time HR plotting demonstration

Unfortunately, the figures above were taken with some portuguese phrases.

About the figure 3.30, the windowing technique can be seen. For the second group of 5 seconds, are read twice the value 75 and three times the value 71, so it stored the most appeared value and stored it.

After this, the figure 3.31 presents an heart rate plot in real time. This plot proves that the value was compressed within the range between 71 and 88 bpm.

After obtaining the value associated to the last 5 seconds, the analyser can send this value to the user interface, where it is compared with was intended.

3.3 Graphical User Interface

This section describes how can the proposed system interact with the user and integrate both of the previously explained modules, the Generator and the Analyser.

The Graphical User Interface, or GUI, aims to provide the user with a simple and intuitive way of configuring and controlling the process of generating and analysing the functional test. Using the TKinter library, for Python language, the GUI was divided into three main pages, each with specific functionalities that facilitate interaction with the system.

Opening the application, the GUI presents the home page. On this page, the user sees some informations, as the title and the author of the application, and can interact with buttons, one saying ”Config.” and the other saying ”Start Essay”. The following figure presents the home page.

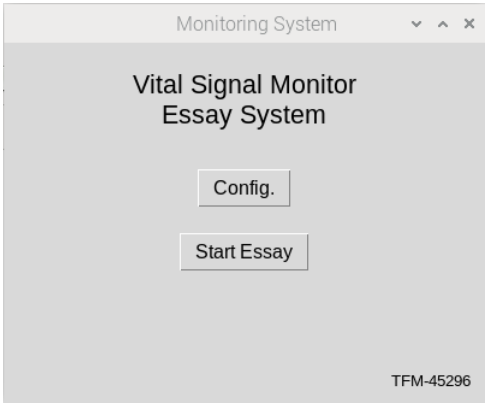

Figure 3.32: GUI demonstration, Home page

Starting with the configuration page, the user has access to choose which test he wants to perform, from a list of pre-saved tests, and save his decision. Here the user has access to another button called ”Draw ROI”, meaning draw Region of Interest. The following figures present both pages mentioned above.

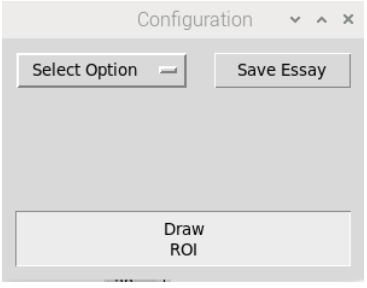

Figure 3.33: GUI demonstration, Configurations page

Figure 3.34: GUI demonstration, Draw ROI page

Clicking on the draw ROI button a new page opens, where it can be seen a real-time footage of the monitor and the user can click with the mouse on two points, drawing a rectangle that will be used by the analyser block for cropping the image, as seen in the figure 3.34. After the selection, user needs to press the key 'A' on his keyboard, to accept the changes.

About the starting essay button, this feature is not entirely finished yet, but it starts the Generator and the Analyser. It does not compare the value generated with the value read. The following figure shows both blocks functioning.

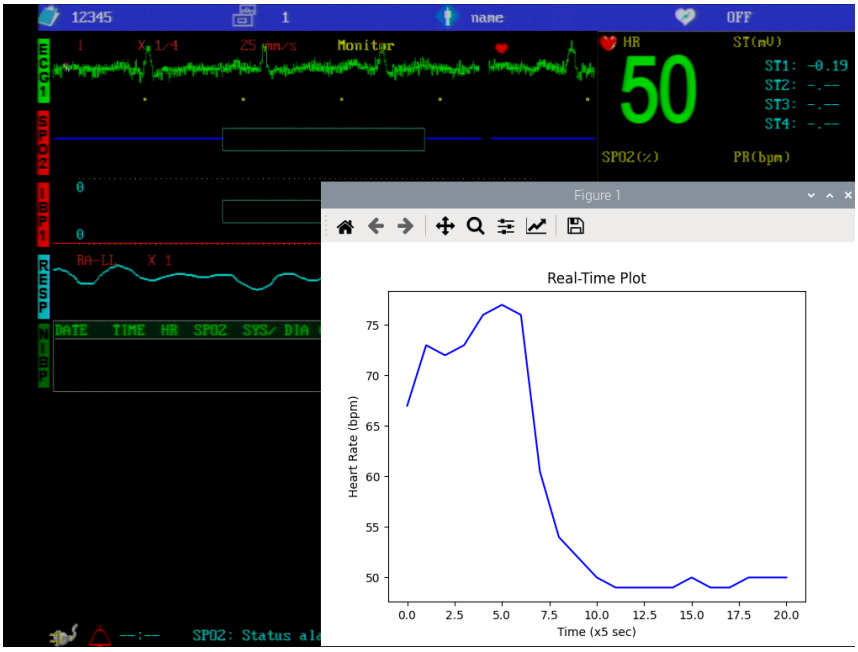

Figure 3.35: Real-Time plotting demonstration

In the image above can be seen, in real-time, both modules working at the same time. The Generator can be seen in the green analog signal present in the monitor, and the Analyser can be seen in the real-time plot.

4

Results and Discussion

This section is dedicated to present results of the system, in macro scale. During the previous sections, all of the different modules have been tested independently, but this section is intended to ensure the full operation of all modules as an embedded system. Some limitations of each block have been presented in the correspondent development section, so it the specifications of each module will not be tested in this section.

To achieve the results intended, four different tests are proposed, and each test must validate as few variables as possible. The first test aims to validate the hardware block's functionality and ensure that the system has access to a video stream displaying the content shown on the monitor. The second test's goal is to validate the implementation of the OCR. The third test intention is to validate the file manager access and the simplest modulation of the standard ECG signal. After all of this, the different components are well-functioning, so the fourth test pretends to modulate the ECG signal, allowing it to have custom HR, and check its stability over time.

Note that the GUI is not ready to generate sine waves, so most of these tests don't use this functionality, but the fifth, and final, test purposed intends to test this feature.

First Test: Hardware and Video Stream

Starting with the most basic thing possible with a generator and a monitor, the first proposed test is to validate subjectively that a sine wave can be generated and presented in the monitor display. The following figures present the results obtained.

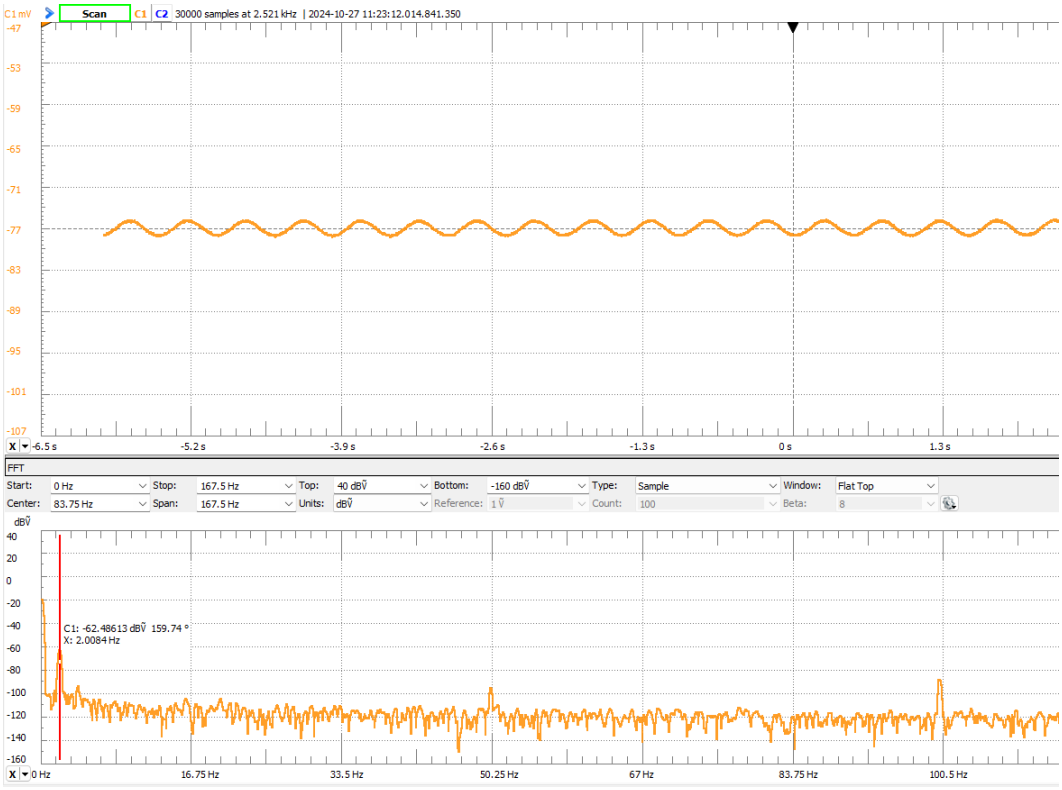

Figure 4.1: Sinusoidal signal in oscilloscope, f = 2 Hz, Real-Time (top subfigure) and FFT (bottom subfigure)

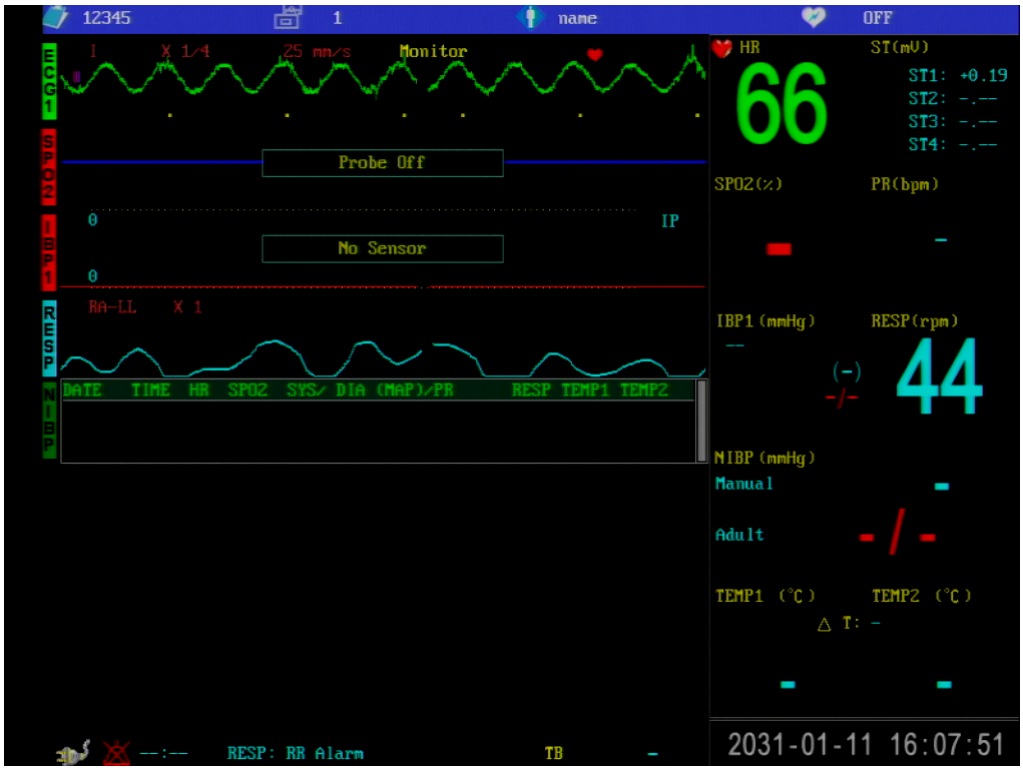

Figure 4.2: Sinusoidal signal in monitor, f = 2 Hz

In the figures above is possible to see the signal in the oscilloscope, with the correspondent FFT, and the monitor video stream.

About the sine wave, it shows a relative error of less than 1% in frequency. This signal presents a big influence of the electrical power supply 50 Hz, but this is intended, as have been explained before. In the monitor, the signal presented is more noisy than the one presented in the oscilloscope because of the high capacity of the electrodes cable, that does not exist in the oscilloscope cables. The electrode cables have a metallic shield used to avoid external influence, but it also increases its capacity, allowing the generation of high frequency noise in the output OpAmp.

Despite the little noise presented, the result are considered good. These type of monitors are designed to be used in extreme conditions where the noise can present much higher amplitudes than the usual.

Second Test: Pre-Processing techniques and OCR algorithm

For the second test, as can be seen in the first test, the monitor presents an HR estimation for the sine wave.

Obviously, this value does not have any real meaning, it is just the output of the monitor's ECG detection algorithm. Even without meaning, this value is being generated and can be measured.

Knowing this, this second test presents a real-time plot of this noisy values presented by the monitor during the generation of a sine wave.

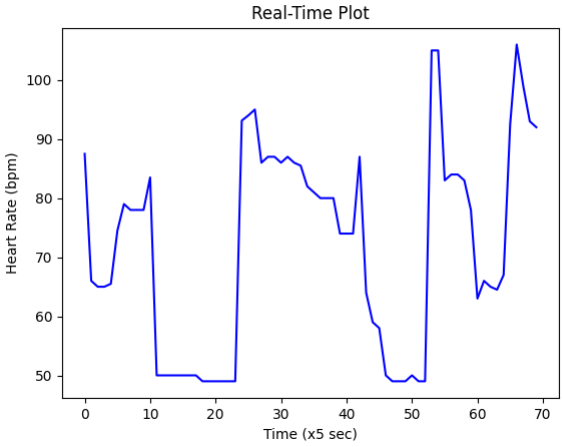

Figure 4.3: Real-Time plotting of the HR measured during a sine wave

The figure above presents the noisy signal expected from the sine wave analysis. Even knowing that this values do not have any physical meaning, is to important to note that the HR measured is always within the range of the expected values to a human HR.

The results to this test are hard to present in this document, but during the acquisition of the signal above, the monitor was present the same values that were registered.

Third Test: Generating the standard ECG signal

Knowing that the generator, the video stream and the OCR blocks are all implemented successfully, than the ECG signal can be generated.

This test pretends to generate an ECG signal provided from an audio file. This aims to test the access to the OS file manager system and the signal generation accuracy. In the following figures will be presented the signal in an audio software, where can be seen the intended signal, and then is presented the results obtained in the monitor.

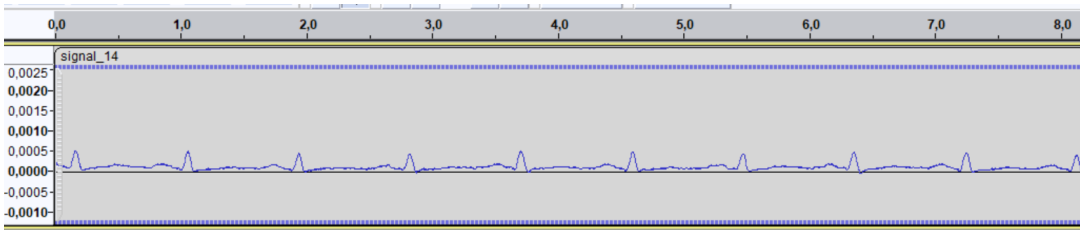

Figure 4.4: Example of the Audio file saved in file system

Unfortunately, it was not possible to fit the entire audio in the same figure. In the figure above are only presented the first 8 seconds, but the actual signal has 20 seconds. In this figure can be clearly seen 10 peaks, representing the QRS complex. Ten peaks during eight seconds is equivalent to approximately 75 BPM. This is not accurate, because, for the full duration of 20 seconds are present 22 peaks, equivalent to 66 peaks in a minute. Even knowing that this is the theoretical value, as will be seen further, actually this signal is analysed as a 67 BPM, because of a faster transition between the last peak of an iteration, and the first peak of the next iteration. The following figures present the results obtained.

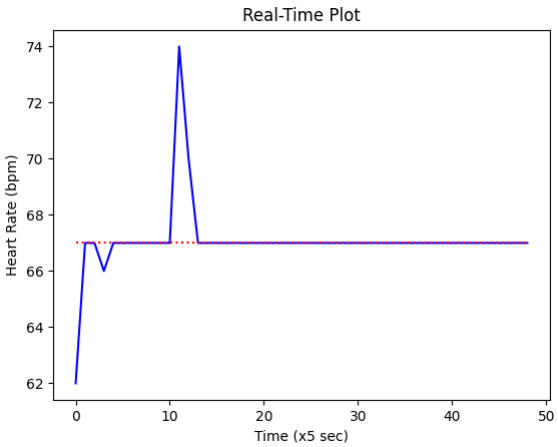

Figure 4.5: Real-Time plot for the standard ECG signal, short duration

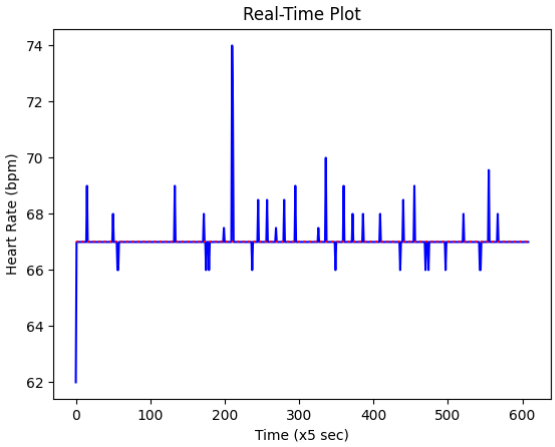

Figure 4.6: Real-Time plot for the standard ECG signal, long duration

In the figures above is possible to see a red dotted line that represents the intended HR value. This means that this values were all obtained using the GUI, specifying the desired HR.

Based on the figures presented above, in both results, a large percentage of the time, the signal is very stable in the 67 BPM, as intended. In the short duration result is possible to a little variation to the 66 BPM, equivalent to an error of 1.5%. This result is considered good, knowing that the monitor is rounding this value to the units value, than this result can much more acceptable.

Obviously, in the figures above are presented some irregular and short peaks out of the acceptable range, ranging from 62 to 74 BPM. This effect lacks a clear explanation regarding its origin, but it is likely due to some irregularities in the monitor's algorithm. Analysing the signal in function of time, it does not represent any irregularities.

Fourth Section: Generating modulated ECG signals

Considering that the generation is well-functioning, then the next parameter that can be tested is the modulation. As explained before, the GUI has an option to insert a number, and it starts generating an ECG signal with an HR equivalent to the inserted value.

As explained in the Vital Signals section, the human heart rate varies from 60 to 100 BPM. Knowing this, some tests were performed trying to reach the limits of the monitor. The results are presented in the following figures.

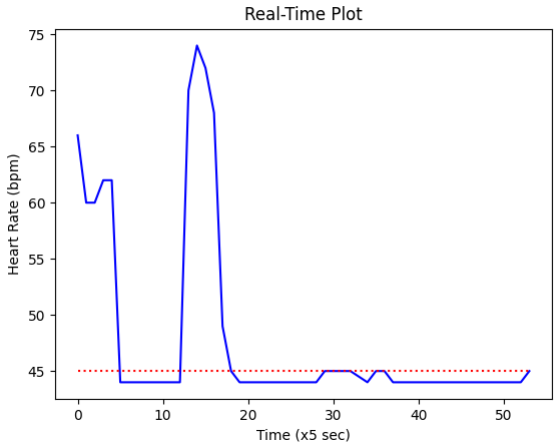

Figure 4.7: Real-Time plot for the modulated ECG signal, HR = 45

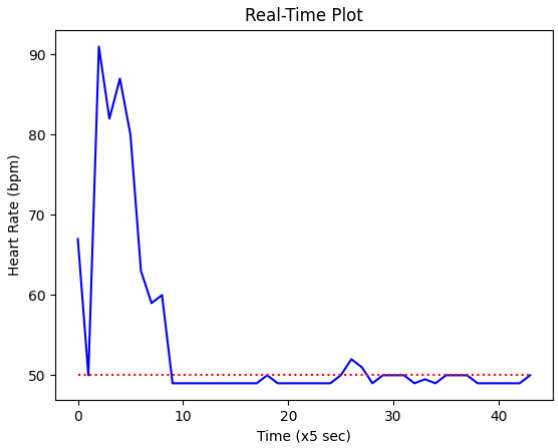

Figure 4.8: Real-Time plot for the modulated ECG signal, HR = 50

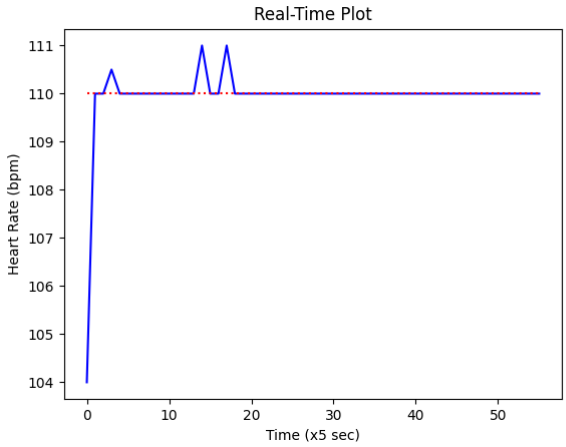

Figure 4.9: Real-Time plot for the modulated ECG signal, HR = 110

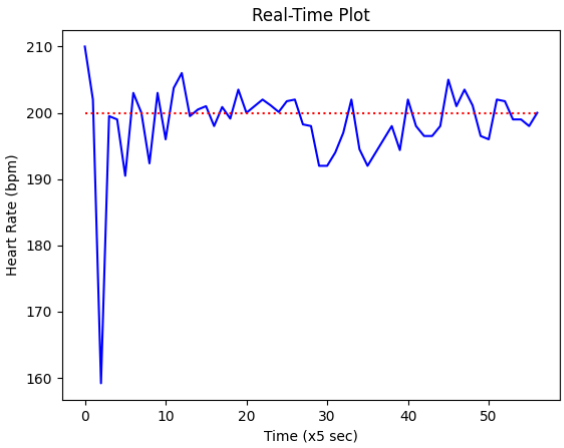

Figure 4.10: Real-Time plot for the modulated ECG signal, HR = 200

Except for the last figure, all the remaining tests presented very good results, with oscillations of approximately 1 BPM, representing approximately 1%, or 2% of error. About the 200 BPM figure, assuming that the greatest deviation is 10 BPM, than the error is 5%.

Values bellow 45 BPM were not recorded because for these values the results shown up to be very unstable and hard to collect. For values above the 100 BPM, the monitor keeps presenting stable and satisfactory because exists a lot of emergencies that results in the increase of the HR, such as heart attack.

This page was intentionally left blank.

5
Conclusions and Future Work

This work presents an innovative platform to automate the functional testes performed to the heart rate calculations of a vital signal multiparametric monitor. The proposed approach integrates techniques of signal processing, digital to analog conversion, and optical character recognition. This project has the intention of optimizing the precision and efficiency during the control of quality of these devices. The automated system developed offers a solution that can substitute the traditional, and manual, methods reducing the human error factor and accelerating the process of validation.

The potential of this platform is huge. Its full implementation promises to make a significant contribution to quality assurance in medical equipment, as well as paving the way for future improvements. This includes integration with many other measured vital data that can be monitored, such as oxygen saturation or blood pressure. This way, it could be an essential tool to any hospital helping premature maintenance.

Although the main blocks have been implemented successfully, from the generation of an ECG signal to the analysis of the results presented in the monitor display, some aspects can be improved, such as software quality and robustness, or integration between the GUI and the peripheral blocksf.

For the future, there are several details that can, and should, be improved. Starting with more versatility in the ECG generation, the software modulation implemented is very limited, only HR modulation, and it can be used to test the monitor in extreme conditions. This can be done using a more complete database of ECG signals, maybe with known pathologies, or even generating the ECG signal through some mathematical parametrization.

Other improvement suggested is the automated recognition of the regions of interest in the graphical display. This version has a configuration screen, where the user indicates the region where the signals are present, but most of the commercial monitors have similarities that could allow to the fully automated recognition of these zones.

This solution uses a computer to do the full operation, and the user needs to plug a mouse, keyboard and external display, to operate it. A significant improvement in portability of the solution would be the implementation of the GUI in a WEB server. In terms of digital security, this seems a very unsafe option, but the computer

can connect to an isolated network in the hospital, or even create its own isolated network, acting as an Access Point. This way the healthcare professional that is operating this device, would just need to plug the electrodes and interact with the GUI in his smartphone. This would be great to achieve the initial purpose of the project, being user-friendly, and allowing its operation by non technical staff.

Most of the improvements mentioned above would result in much more computational process, so the first improvement should be the use implementation of the signal generator in an embedded solution, with a microcontroller. The single board computer was used to preserve the simplicity, but it has several problems resultant of the use of a standard OS. Obviously, to achieve this, maintaining the portability, the custom PCB would need to receive several improvements, but the most important would be to use SMD technology, instead of the Through Hole, resulting in much more space available to the new improvements. Using a microcontroller, would improve significantly the energy consumption of the system, improving even more the portability of this solution.

One of the intial goals was to have a proof-of-concept device. To achieve this, the product designed were developed to help hospitals, and other health facilities, but allowing external tools, as an API, it could be very useful to the manufacturers too.

Concluding this reflection, the final state of development presented is not a commercial device, but nevertheless is very valuable in proving that this platform may be very useful. A plataform that integrates an ECG generator electronic board and image processing analysis was developed to test the Heart Rate functionality of Multiparametric Monitors. The plataform is easily adapted to different multiparametric monitors and most importantly contains a feedback loop for automated tests. This is an important first step in demonstrating that operations formerly performed by highly trained professionals, can be done by this system, preventing natural human errors and inaccuracies.

Bibliography

[1] Dr. Kurtuluş Erinç Akdoğan. *The Origin of Biopotentials*. URL: https://mece493.cankaya.edu.tr/uploads/files/TheOriginOfBiopotentials%281%29.pdf. (accessed: 11.2023).

[2] Majd AlGhatrif and Joseph Lindsay. "A brief review: history to understand fundamentals of electrocardiography". In: *Journal of Community Hospital Internal Medicine Perspectives* 2.1 (2012). PMID: 23882360, p. 14383. DOI: 10.3402/jchimp.v2i1.14383. eprint: https://doi.org/10.3402/jchimp.v2i1.14383. URL: https://doi.org/10.3402/jchimp.v2i1.14383. (accessed: 8.2024).

[3] Daniel Almeida. "Non-Intrusive ECG Acquisition Test-bed". MA thesis. Instituto Superior de Engenharia de Lisboa, 2018.

[4] Evismar Andrade et al. "State of the Art and Future Trends in the Usability of Patient Monitors". In: *Human Systems Engineering and Design*. Ed. by Tareq Ahram, Waldemar Karwowski, and Redha Taiar. Cham: Springer International Publishing, 2019, pp. 338–344. ISBN: 978-3-030-02053-8. (accessed: 8.2024).

[5] H. Austerlitz. *Data Acquisition Techniques Using PCs*. Elsevier Science, 2002. ISBN: 9780080530253. URL: https://books.google.pt/books?id=gK4DxMaqmYYC.

[6] G.E. Burch and N.P. DePasquale. *A History of Electrocardiography*. Norman cardiology series. Norman Pub., 1990. ISBN: 9780930405212. URL: https://books.google.pt/books?id=_w9v_ZkzTZoC.

[7] Dawei medical Co. *Product Comparison Chart for Patient Monitors*. URL: https://daweihealth.com/pt-pt/pages/product-comparison-chart-for-patient-monitors. (accessed: 3.2024).

[8] International Electrotechinal Comission. *IEC 60601-2-25*. URL: https://webstore.iec.ch/en/publication/2636. (accessed: 9.2024).

[9] International Electrotechinal Comission. *IEC 60601-2-27*. URL: https://webstore.iec.ch/en/publication/2638. (accessed: 9.2024).

[10] International Electrotechinal Comission. *IEC 62353*. URL: https://webstore.iec.ch/en/publication/6913. (accessed: 9.2024).

[11] Fluke Corporation. *Patient Simulators*. URL: https://www.flukebiomedical.com/products/biomedical-test-equipment/patient-simulators. (accessed: 3.2024).

[12] Abhranila Das, Chirasree Roy Chaudhuri, and Indranil Das. "Advanced Portable ECG Simulator: Product Development Validation". In: *2019 Women Institute of Technology Conference on Electrical and Computer Engineering (WITCON ECE)*. 2019, pp. 187–191. DOI: 10.1109/WITCONECE48374.2019.9092906.

[13] Analog Devices. "An Introduction to Jitter in Communications Systems". In: (2024). URL: https://www.analog.com/en/resources/technical-articles/an-introduction-to-jitter-in-communications-systems.html. (accessed: 4.2024).

[14] Digilent. *Analog Discovery 3*. https://digilent.com/reference/test-and-measurement/analog-discovery-3/specifications?srsltid=AfmBOoqGmVoZGWNTiJOopWfdI8jn5sXkH9P$_TXaDJnjn8E7$ (accessed: 3.2024).

[15] Hi-Link. *HLK-B$_S - 1WR3series$*. Datasheet. 2024. URL: http://www.kosmodrom.com.ua/pdf/B_S-1WR3.pdf. (accessed: 6.2024).

[16] Inc. Infinium Medical. *Patient Monitors*. URL: https://infiniummedical.com/product-category/patient-monitors/?gclid=CjwKCAjwx_eiBhBGEiwA15gLN3LzCii2UVhjLoI8KPjBwEclass=\%22elementor-button. (accessed: 3.2024).

[17] Texas Instruments. *Operational Amplifier Gain Stability, Part 3: Why You Should Care About Capacitance at the Inverting Input*. Tech. rep. SBOA226. Accessed: 2024-09-29. Texas Instruments, 2015. URL: https://www.ti.com/lit/an/sboa226/sboa226.pdf.

[18] Chankyu Joung et al. "Deep learning based ECG segmentation for delineation of diverse arrhythmias". In: *PLOS ONE* 19 (June 2024). DOI: 10.1371/journal.pone.0303178.

[19] Michael H. Grider; Rishita Jessu; Rian Kabir. *Physiology, Action Potential*. URL: https://www.ncbi.nlm.nih.gov/books/NBK538143/. (accessed: 12.2023).

[20] Pratik Kanani and Dr. Mamta Padole. "Recognizing Real Time ECG Anomalies Using Arduino, AD8232 and Java: Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018, Revised Selected Papers, Part I". In: Apr. 2018, pp. 54–64. ISBN: 978-981-13-1809-2. DOI: 10.1007/978-981-13-1810-8_6.

[21] R. Kyle and W.B. Murray. *Clinical Simulation*. Elsevier Science, 2010. ISBN: 9780080556970. URL: https://books.google.pt/books?id=gonEANOwR-gC.

[22] A.I. Levine et al. *The Comprehensive Textbook of Healthcare Simulation*. Springer New York, 2013. ISBN: 9781461459934. URL: https://books.google.pt/books?id=WWRDAAAAQBAJ.

[23] Vincent Luppino. *How to Use Heart Rate Variability Data in Your Training*. URL: https://www.hss.edu/article_heart-rate-variability.asp. (accessed: 11.2023).

[24] National MagLab. *Willem Eithoven*. URL: https://nationalmaglab.org/magnet-academy/history-of-electricity-magnetism/pioneers/willem-einthoven/. (accessed: 11.2023).

[25] Joshua D. Pollock; Amgad N. Makaryus. *Physiology, Cardiac Cycle*. URL: https://www.ncbi.nlm.nih.gov/books/NBK459327/. (accessed: 11.2023).

[26] Kamil Maliński and Krzysztof Okarma. "Analysis of Image Preprocessing and Binarization Methods for OCR-Based Detection and Classification of Electronic Integrated Circuit Labeling". In: *Electronics* 12.11 (2023). ISSN: 2079-9292. DOI: 10.3390/electronics12112449. URL: https://www.mdpi.com/2079-9292/12/11/2449.

[27] Steve Meek and Francis Morris. "ABC of clinical electrocardiography.Introduction. I-Leads, rate, rhythm, and cardiac axis". en. In: *BMJ* 324.7334 (Feb. 2002), pp. 415–418.

[28] Jamshed Memon et al. "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)". In: *IEEE Access* 8 (2020), pp. 142642–142668. DOI: 10.1109/ACCESS.2020.3012542.

[29] Microchip. *8/10/12-Bit Dual Voltage Output Digital-to-Analog Converter with SPI Interface*. Datasheet. 2024. URL: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/22250A.pdf. (accessed: 3.2024).

[30] Rishabh Mittal and Anchal Garg. "Text extraction using OCR: A Systematic Review". In: *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2020, pp. 357–362. DOI: 10.1109/ICIRCA48905.2020.9183326.

[31] W. Nehring and F. Lashley. *High-Fidelity Patient Simulation in Nursing Education*. Jones & Bartlett Learning, 2010. ISBN: 9780763756512. URL: https://books.google.pt/books?id=EJpZU0saqo0C.

[32] OpenCV. *Color Conversions*. 2017. URL: https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html. (accessed: 8.2024).

[33] OpenCV. *Smoothing Images*. URL: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html. (accessed: 3.2024).

[34] GIMP GNU Image Manipulation Program. *Convolution Matrix*. URL: https://docs.gimp.org/2.8/en/plug-in-convmatrix.html. (accessed: 3.2024).

[35] Mageshwaran R. *Survey on Image Preprocessing Techniques to Improve OCR Accuracy*. URL: https://medium.com/technovators/survey-on-image-preprocessing-techniques-to-improve-ocr-accuracy-616ddb931b76. (accessed: 3.2024).

[36] V. Rajinikanth, N.S.M. Raja, and N. Dey. *A Beginner's Guide to Multilevel Image Thresholding*. Intelligent Signal Processing and Data Analysis. CRC Press, 2020. ISBN: 9781000228335. URL: https://books.google.pt/books?id=NJ4FEAAAQBAJ.

[37] Kothapally Aditya Reddy et al. "Advanced ICU Patient Monitoring With Sensor Integration, IV Detection WITH Canny Edge Detection and ECG Monitoring With Live Feed". In: *2023 International Conference on Recent Advances in Electrical, Electronics Digital Healthcare Technologies (REEDCON)*. 2023, pp. 335–339. DOI: 10.1109/REEDCON57544.2023.10151176.

[38] R. Rhoades and D.R. Bell. *Medical Physiology: Principles for Clinical Medicine*. Lippincott Williams & Wilkins, 2009. ISBN: 9780781768528. URL: https://books.google.pt/books?id=tBeAeYS-vRUC.

[39] Vanessa Sofia da Silva Mota. "Planos de Manutenção Preventiva de Monitores Multiparamétricos". MA thesis. Instituto Superior de Engenharia de Lisboa, 2018. URL: https://repositorio.ipl.pt/bitstream/10400.21/15977/1/Dissertacao.pdf. (accessed: 1.2024).

[40] Ray Smith. *An Overview of the Tesseract OCR Engine*. URL: https://storage.googleapis.com/pub-tools-public-publication-data/pdf/33418.pdf. (accessed: 3.2024).

[41] Guijuan Song et al. "Research on Medical Service System Based on Big Data Technology". In: *2019 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*. 2019, pp. 302–304. DOI: 10.1109/ICITBS.2019.00079.

[42] Texas Instruments. *LM1084 5-A Low Dropout Positive Regulator*. 2023. (accessed: 5.2024).

[43] Jana Vasković. *Action Potential*. URL: https://www.kenhub.com/en/library/anatomy/action-potential. (accessed: 12.2023).

[44] Vishay. *High Speed Optocoupler, Single and Dual, 10 MBd*. Datasheet. 2024. URL: https://www.alldatasheet.com/datasheet-pdf/view/967009/VISHAY/6N137.html. (accessed: 6.2024).

[45] Sami Viskin, David Zelster, and Charles Antzelevitch. "When u say "U Waves," what do u mean?" en. In: *Pacing Clin. Electrophysiol.* 27.2 (Feb. 2004), pp. 145–147.

[46] J.G. Webster. *Design of Pulse Oximeters*. Series in Medical Physics and Biomedical Engineering. CRC Press, 1997. ISBN: 9781420050790. URL: https://books.google.pt/books?id=eQh1DQtvowUC.

[47] J.G. Webster and A.J. Nimunkar. *Medical Instrumentation: Application and Design*. Wiley, 2020. ISBN: 9781119457336. URL: https://books.google.pt/books?id=1ovgDwAAQBAJ.

[48] Stefan Weil. *Tesseract OCR*. URL: https://github.com/tesseract-ocr/tesseract/tree/main. (accessed: 3.2024).