UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Grau en Enginyeria Física

Treball Final del Grau en Enginyeria Física

Portable low-power system for continuous monitorization of the cardiovascular system

Iván Mansergas Fernández

Thesis Director Jaime Oscar Casas Piedrafita
Instrumentation, Sensors and Interfaces Group
Universitat Politècnica de Catalunya

June 28, 2021

Iván Mansergas Fernández
Portable low-power system for continuous
monitorization of the cardiovascular system
June 28, 2021
Thesis Director: Jaime Oscar Casas Piedrafita
Universitat Politècnica de Catalunya

Grau en Enginyeria Física
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
C/Jordi Girona, 1
08034 and Barcelona

Acknowledgement

It all started with an email, where the subject was the possibility of doing a project related to electronics, in which I will focus my career. Prof. Oscar Casas kindly presented to me the line of work of his research group, and proposed to me this interesting project. From the beginning to the end I have never felt alone in this adventure, as he provided me with all the knowledge, tools and learning that I needed at every stage of the work done, being always available and interested in my advance day over day. This is why I would like to express my most sincere gratitude to all he has done for me.

I would like to thank as well the rest of the ISI group, for their availability and disposal for everything that I could have requested. In particular, to Francis López, from whom I learned very interesting things about the most technical electronic aspects of the project. Thanks to him, the devices designed have seen the day of light and with a professional quality. In addition, I would like to mention my laboratory colleagues, Marc and Marta, which apart from helping me in the biological aspects of measurements and acquisition software, they made my days more bearable, with conversations about life that made me grow up in mind, specially for the laboral life.

Last but not least, I am very thankful for my family and friends, which not only during the project, but throughout the degree, have been my stress canalization, as they have taken me out of the hard times to have great times that made me recover the happiness that sometimes the studies stole.

Abstract

In the last year, the appearance of a global pandemic due to the COVID-19 virus evidenced one problem: the medical system is not prepared to treat a lot of people at the same time, as all the clinical environments collapsed during practically the whole year. This is where performing ambulatory measurements plays an important role, because it would allow to carry out medical diagnosis without the need of assisting at the hospital; everyone could have a device at home which would do a first health analysis to see if it would be necessary to visit the doctor.

Nevertheless, recent medical technologies are very expensive and many people could not afford them. Thus, the aim of this project is to design and implement a portable low-cost and low-power consumption system capable of acquiring biomedical signals from which different physiological parameters will be extracted to give information about the health status of the patient. Although the project will be focused on the electronics with the hardware and software, a series of measurements will be carried out as well to show the feasibility of the data acquired.

Different measurements of the cardiovascular system have been carried out, obtaining the electrical signal: 12 ECG leads from the standard positions (10 electrodes), and a subset of leads with less electrodes and a not-standard position, to ease the measurements outside the hospital. For its importance, something that will be achieved as well is the possibility of synthesizing the 12 ECG leads from a subset of less electrodes incorporating also a not-standard placement of the electrode, so that it will not be necessary to place all the 10 electrodes on the body, making the measurement more comfortable. In addition, pulse wave signals have been measured in different positions. The signal acquisition is complemented with the development of a software capable of obtaining automatically the important parameters. Finally, relating the pulse wave signals with the electrical signals of the ECG, we will be able to monitor the circulation of blood and then have information about the health status of the cardiovascular system.

Contents

Introduction

1

1.1 History and motivation

Ever since ancient times, medical diagnosis have been of great importance for civilizations, as physicians were able to describe different disturbances and diseases, based on the observation of clinical symptoms [1]. The history tells how diagnosis has been improved through the development of new methodologies which allowed to gather valuable knowledge about symptoms and illnesses. This knowledge has been widely broadened over time, but it is never a closed book, due to the appearance of new diseases such as the recent COVID-19 of last year.

In the last century, it has been possible to obtain biological signals from the body which can give medical information about the health status of a subject. Nowadays, with the use of the latest technologies, this data can be easily acquired with great quality. These devices are able to capture physiological signals, which are then processed in order to obtain parameters and information about the state of the patient, and therefore prevent any complications.

The literature suggests two main types of biosignals for the cardiovascular system: electrical signals of the heart, and pressure signals of the blood circulation. The first one is typically performed through the electrocardiogram (ECG), used for measuring the electrical activity of the heart. It has been the most studied, as well as the electrode positioning around the body, and technology has allowed it to be the best non-invasive procedure to detect heart abnormalities. The second type of biosignal can be measured through different procedures such as photoplethysmography (PPG), impedance plethysmography (IPG), or tonometry, which all have similar waveforms.

The motivation of this project is to develop the electronics of a device capable of performing the acquisition of these cardiovascular signals, in order to characterize the heart and the circulation of blood. Nevertheless, we want our system to be flexible and measure any biological signal or parameter, some of them measured directly as biopotentials like ECG, electroencephalogram (EEG) or electromyogram (EMG), and other signals will be acquired with front-ends. For this, it will be necessary to design the hardware of the device, as well as the software that will control it, with the algorithms for the automatic extraction of the important parameters.

1.2 Current technologies and problem statement

As mentioned before, technology has allowed to greatly advance and improve the medical diagnosis methods, to the point that many commercial devices are in the market offering different functionalities for the biosignals acquisition. However, most of these devices could present some lacks, such as some minor distortions, the consumption and market prices, or even the incommodity of the electrodes placement.

Nowadays, for a conventional ECG measurement device, a set of 10 electrodes are needed to get the complete information of the heart status with a good quality. These electrodes are placed at the standard positions of the limbs (arms and legs) and the chest, so it is clear that it could be an inconvenience for the patient being connected to 10 electrodes at the same time, as well as it is not practical. In order to solve this problem, in the literature [2] [3] [4] it is found that some devices have been invented in order to obtain the required 12 ECG lead signals from a subset of leads, allowing this way to use less electrodes and to place them in comfortable positions of the body, even if they are not standard. Nevertheless, using this kind of techniques is that the correlation between the real and the synthesized one is not enough due to a loss of information, so these procedures cannot be used for diagnostic purposes, but it can be useful for monitoring.

The commercial devices do not usually design their hardware in order to provide a low-power consumption, they prefer to develop a functional device which can offer as many functionalities as possible in order to have a competitive product. In addition, most of these devices can be very expensive because of the costs of manufacture and branding; the prices in the market are around thousands of dollars.

Another problem of current technologies is their lack in functionalities, as some devices offer a very good quality in some aspects such as signal processing, but they are limited in other things like the number of channels to connect signals. The device in which this whole project is inspired is the one owned by the Instrumentation, Sensors and Interfaces research group: the BIOPAC Lab System [5]. This commercial device includes a user-friendly software that is capable of acquiring in real time different biosignals (muscular, cardiovascular and neurophysiological). However, the limitations presented by this device are the number of channels, the connection for these channels, and depending on the model the sampling resolution.

In conclusion, what is needed to develop is a low-cost and low-power consumption device which can compete in the market with the commercial ones, being it even improved to solve the limitations of them and provide many functionalities at the same time. Another important thing to take into account is the possibility of reaching not only a home device, but also to reach anywhere in the world.

1.3 Objectives and thesis approach

As it has been stated, the aim of this project is the development of a functional low-cost and low-power consumption portable device, capable of monitoring physiological parameters from a limb. The project will be carried out under three main objectives.

First of all, as a prelude, some research will be necessary to learn about the biological processes that generate the different types of signals, as well as the mechanisms of recent technologies for the data acquisition. In addition, a further study on the typical parameters of each signal will be useful in order to implement the different algorithms for our software.

After this, the first goal is to design the low-cost and low-power consumption hardware for multiparametric measurements. The starting point is the commercial BIOPAC system mentioned before, as it will be the inspiration for the development of a device that solves its limitations. The idea is to design the hardware of two different acquisition devices, one focused on ECG signals, and the other used for generic signals such as IPG, PPG or ECG as well. All of this will be controlled by a microcontroller that will receive the data with SPI communication protocol, and then transmit the information by USB to a computer. A final PCB board will be designed in order to gather all the different hardware components necessary for our purpose. Finally, the developed hardware must be able to perform good quality measurements in ambulatory scenarios.

The second objective is the development and implementation of the software. This comprises different aspects; first of all, the microcontroller program capable of controlling the whole system. Then, a program that receives the data and saves the information so that the data can be processed. Finally, the different data processing algorithms, plot representations and parameters must be obtained with another program as well; the main idea is to use MATLAB, and create an application program like the one provided for the BIOPAC system.

The data processing software is intended to contain a specific algorithm in which the 12 ECG leads are reconstructed from a subset of less leads, based on the work done in [6]. The reason for this is to solve one of the main problems stated before, in which the placement of 10 electrodes could be uncomfortable, so the goal is to use less electrodes and obtain the 12 leads from a subset through a series of transform matrices, as it will be explained in further sections.

The final goal is to prove the feasibility of the devices, by the performance of a set of measurements on different subjects. The information obtained from these measurements will be analysed in order to rate the quality of the acquisition systems, and then try to obtain the best quality with possible optimizations of both the hardware (mechanical connections, electrodes, front-ends. . .) and software (code and algorithm optimization).

Once all the objectives are accomplished, we will have a functional low-cost and low-power device comparable with a thousand of dollars commercial product, with the idea to create a platform that allows future extensions with newer communication systems. However, the most important part is that it will be smaller in dimensions, so that it is better for portability and even for thinking of the possibility of developing a “domotic product” which everyone could have in their homes, capable of monitoring in real time the biological status of the body without the necessity of assisting in a clinical environment. In these times, where the COVID-19 has overlapped the capacity of hospitals, this idea is very interesting.

1.4 Thesis Structure

Chapter 2

This first chapter presents the theoretical concepts that are important to take into account and that will be used all over the project. This concepts rely on the explanation of the cardiac cycle, and the introduction of the biological signals that will be measured with our devices: the electrocardiogram and other pulse wave signals. The waveform and the electrode placement will be the topics of study, as well as the physiological parameters that will be extracted.

Chapter 3

In this section will be explained the hardware development of our system. Firstly, the ECG acquisition device will be presented, from the prototype to its final version. Secondly, the generic device, whose procedure will be similar to the ECG device. Finally, the power bank that will be used to supply our system.

Chapter 4

This chapter will present the different software programs and algorithms implemented: the code to program the microcontroller, the data acquisition program in LABView, and the data processing and parameter extraction with algorithms in MATLAB.

Chapter 5

In this chapter different measurements will be carried out. First of all, a discussion on the power consumption and the material costs will be done, in order to compare with other commercial devices. Then, to prove that our systems work as expected, measurements on ECG and pulse wave signals will be performed, as well as the extraction of the important parameters, the reconstruction of 12 ECG leads from a not-standard subset, and the relation with the pulse wave signals for a complete cardiovascular system analysis.

Chapter 6

The last chapter will be dedicated to the conclusions of the project, as well as some future work and perspectives to improve our system.

Theoretical framework

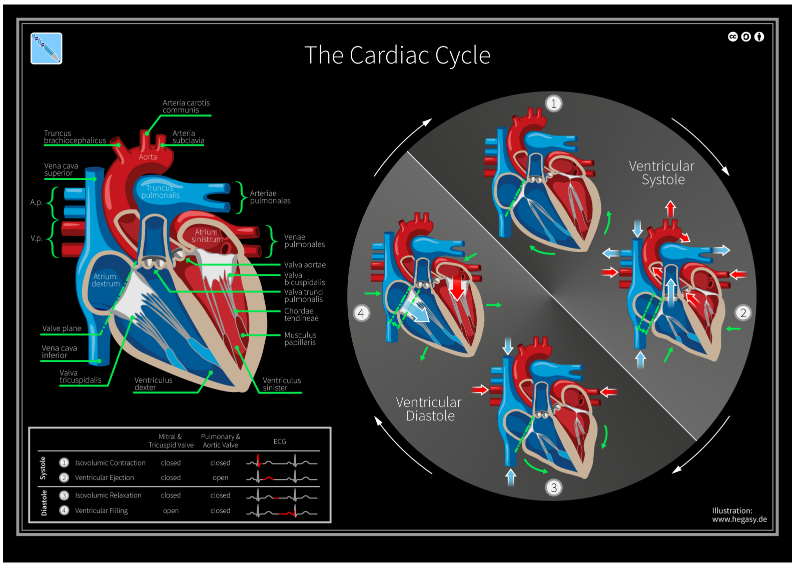
2

2.1 The heart as an electrical source

2.1.1 The heart and the heartbeat

It is well known that the heart is probably the most important organ in the human body, which has an essential role in the circulatory system pumping blood through the blood vessels. Therefore, oxygen and nutrients can be transported to the different parts of the body, as well as other metabolic waste.

One full cycle consists of two phases [7]. Firstly, the heart muscle relaxes (isovolumic relaxation) and then refills the ventricles with blood (ventricular filling) through the aorta; this phase is known as diastole. Secondly, it follows a period of robust contraction (isovolumic contraction) and pumping of blood (ventricular ejection); this last phase is called systole. This whole cycle constitutes one heartbeat and is repeated constantly, performing the necessary circulation of blood through the body.



			Mitral & Tricuspid Valve	Pulmonary & Aortic Valve	ECG
Systole	1	Isovolumic Contraction	closed	closed	
	2	Ventricular Ejection	closed	open	
Diastole	3	Isovolumic Relaxation	closed	closed	
	4	Ventricular Filling	open	closed	

Fig. 2.1.: Illustration of the cardiac cycle.

Nevertheless, what is interesting about the heart is its electrical properties, with which it is possible to obtain different signals that can give biological information about the subject. In fact, in order to pump the blood, an electrical current is needed at a certain rhythm; this is known as the cardiac rhythm, and is what we are used to see in hospitals with the heart rate monitors.

The most common signal is the electrocardiogram, which is directly related to these electrical properties. However, we can obtain other signals related to the pressure and blood volume, such as the plethysmography.

2.2 Electrocardiogram (ECG)

The electrocardiogram (abbreviated as ECG or EKG) represents an electrical tracing of the heart and is recorded non-invasively from the surface of the body [8]. It has become one of the main diagnostic modalities due to the fact that it has a substantial clinical impact on investigating the severity of cardiovascular diseases. Cardiovascular disease, as the number one cause of death, puts a great emphasis on health-care providers to develop skills and knowledge in interpreting ECGs to provide the best care promptly. This is why ECG is one of the most studied signals in the medical area.

The signals are obtained using electrodes placed on the skin. These electrodes detect the small electrical changes that are a consequence of cardiac muscle depolarization followed by repolarization during each cardiac cycle [9]. These electrodes are connected to an electrical machine that converts the electrical activity into a digital signal, so that it can be represented.

2.2.1 Leads

Any pair of electrodes can measure the electrical potential difference between the two corresponding locations of attachment [9]. Such a pair forms a lead. However, the leads can also be formed between a physical electrode and a virtual electrode, known as Wilson's Central Terminal (WCT), whose potential is defined as the average potential measured by three limb electrodes that are attached to the right arm, the left arm, and the left foot, respectively.

The conventional ECG machine consists of 10 electrodes that form 12 leads, which divide into two groups: limb leads and precordial leads [8]. Limb leads are further categorized as standard bipolar limb leads I, II and III, and augmented unipolar leads aVL, aVF, and aVR. The precordial leads include V1 to V6. The limb leads view the heart in a vertical plane, and the precordial leads record the electrical activity of the heart in the horizontal plane. The fundamental principles behind the recording of an ECG is an electromagnetic force, current or vector that has both magnitude and direction. When a depolarization current travels towards the electrode, it gets recorded as a positive deflection, and when it moves away from the electrode, it appears as a negative deflection.

The placement of the electrodes are specified in the following table.

Electrode name	Electrode position
RA	On the right arm, avoiding thick muscle
LA	In the same location as RA, but in the left arm
RL	On the right leg, lower end of inner aspect of calf muscle
LL	In the same location as RL, but in the left leg
V1	In the fourth intercostal space, just to the right of the sternum
V2	In the fourth intercostal space, just to the left of the sternum
V3	Between leads V2 and V3 location
V4	In the fifth intercostal space, in the mid-clavicular line
V5	Horizontally even with V4, in the left anterior axillary line
V6	Horizontally even with V4 and V5 in the mid-axillary line

Tab. 2.1.: Table explaining the placement of the electrodes

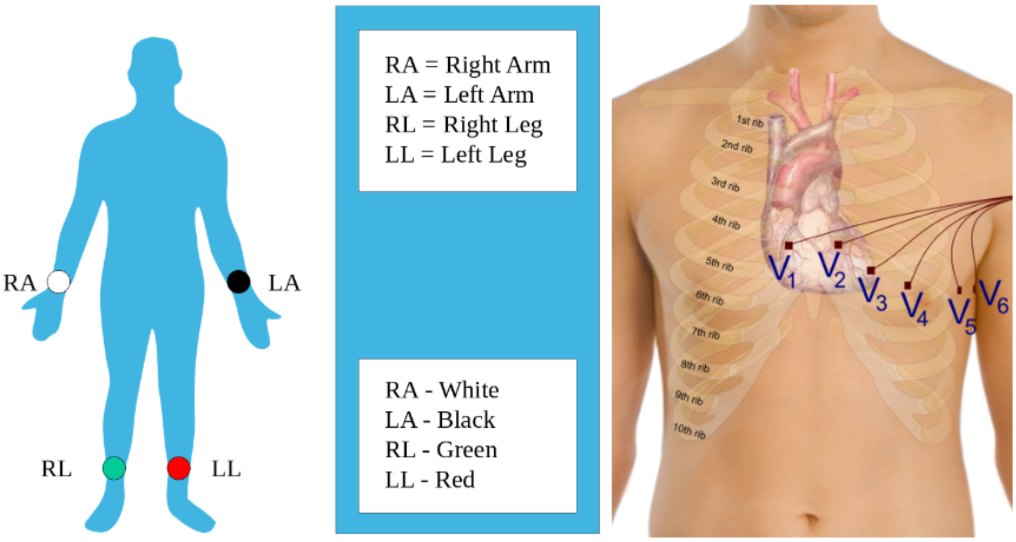


Fig. 2.2.: On the left, the position of the limb electrodes. On the right, the location of the chest electrodes.

Limb leads

The limb leads are known as Lead I, II and III, and are considered the main ECG signals to study. The electrodes correspond to the ones placed in RA, LA and LL, which form what is called the Einthoven's triangle.

- Lead I is the voltage between the (positive) left arm (LA) electrode and right arm (RA) electrode.
$$I = LA - RA \tag{2.1}$$

- Lead II is the voltage between the (positive) left leg (LL) electrode and the right arm (RA) electrode.
$$II = LL - RA \tag{2.2}$$

- Lead III is the voltage between the (positive) left leg (LL) electrode and the left arm (LA) electrode.
$$III = LL - LA \tag{2.3}$$

Augmented limb leads

These leads are derived from the previous ones, forming the aVR, aVL, and aVF signals.

- Lead augmented vector right (aVR) has the positive electrode on the right arm. The negative pole is a combination of the left arm electrode and the left leg electrode.
$$aVR = RA - \frac{1}{2}(LA + LL) \tag{2.4}$$

- Lead augmented vector left (aVL) has the positive electrode on the left arm. The negative pole is a combination of the right arm electrode and the left leg electrode.
$$aVL = LA - \frac{1}{2}(RA + LL) \tag{2.5}$$

- Lead augmented vector foot (aVF) has the positive electrode on the left leg. The negative pole is a combination of the right arm electrode and the left arm electrode.
$$aVF = LL - \frac{1}{2}(RA + LA) \tag{2.6}$$

Precordial leads

Finally, the precordial leads lie in the transverse (horizontal) plane, perpendicular to the other six leads [9]. The six precordial electrodes act as the positive poles for the six corresponding precordial leads: V1, V2, V3, V4, V5, and V6. Wilson's central terminal is used as the negative pole.

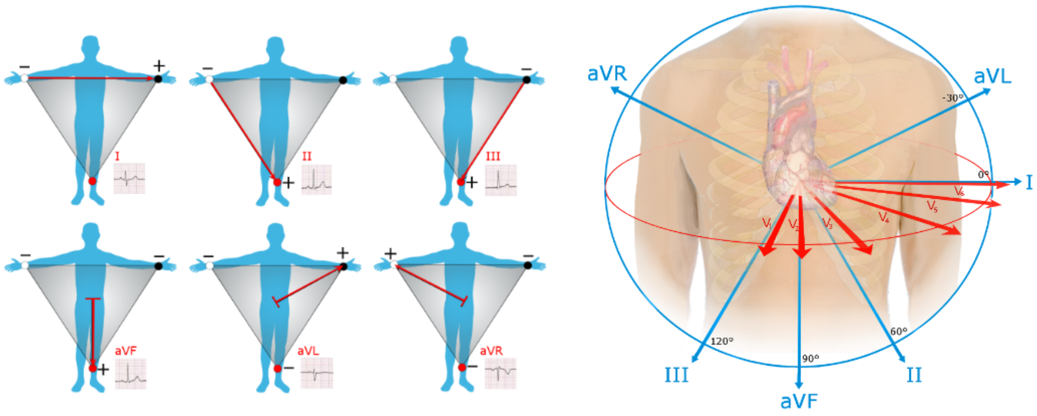
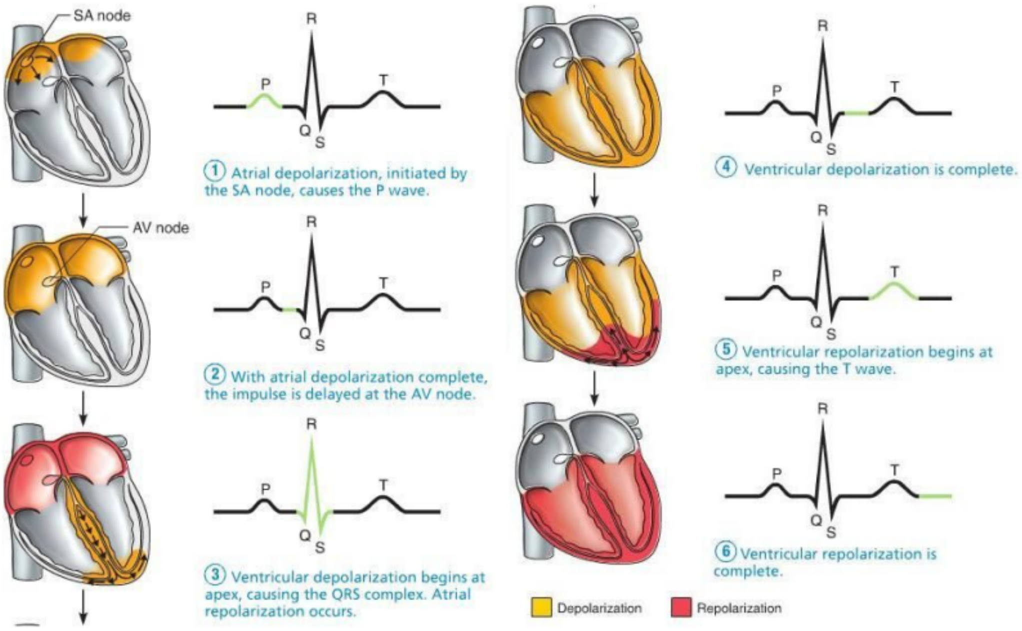


Fig. 2.3.: Visual representation of the limb and precordial leads.

2.2.2 The ECG wave

The wave considered in an electrocardiogram is related to the heartbeat cycle mentioned before. Each phase of the cycle produces an electric potential, and the combination of all parts form the ECG signal, as it is shown in the next figure. We can see that different parts appear in this signal, labelled with the letters P, Q, R, S and T. Each part corresponds to a different stage in the depolarization of the hearth.



SA node
AV node
P
Q
R
S
T
1 Atrial depolarization, initiated by the SA node, causes the P wave.
2 With atrial depolarization complete, the impulse is delayed at the AV node.
3 Ventricular depolarization begins at apex, causing the QRS complex. Atrial repolarization occurs.
4 Ventricular depolarization is complete.
5 Ventricular repolarization begins at apex, causing the T wave.
6 Ventricular repolarization is complete.
🟨 Depolarization
🟥 Repolarization

Fig. 2.4.: Representation of the different waves corresponding to the propagation of the electrical potential over different instants of time.

By inspection of one or multiple leads of the ECG it is possible to obtain a huge amount of information such as if the patient has suffered from a myocardial infarction or any kind of arrhythmias. Actually, the goal of the ECG interpretation is the ability to determine whether the ECG waves and intervals are normal or pathological. Electrical signal interpretation gives a good approximation of heart pathology [8]. All of the waves on an ECG tracing and the intervals between them have a predictable time duration, a range of acceptable amplitudes (voltages), and a typical morphology. Any deviation from the normal tracing is potentially pathological and therefore of clinical significance.

Some of the most important information to take into account from an ECG wave are the following points.

- P wave: it represents the depolarization of the atria, where the contraction of the atria is produced. Atrial depolarization spreads from the SA node towards the AV node (see Fig. 2.4), and from the right atrium to the left atrium.

- PR interval: it reflects the time the electrical impulse takes to travel from the sinus node through the AV node.

- QRS complex: it represents the rapid depolarization of the right and left ventricles. The ventricles have a large muscle mass compared to the atria, so the QRS complex usually has a much larger amplitude than the P wave. At this moment, systole is produced, where there is a ventricular polarization and contraction.

- ST interval: it represents the period when the ventricles are depolarized.

- T wave: it represents the repolarization of the ventricles. It is generally upright in all leads except aVR and lead V1. A relaxation of the ventricles occurs.

It is interesting to notice that there is no wave corresponding to atrial repolarization, this is due to the fact that it is masked by the QRS complex.

Another interesting parameter to take into account is the distance between two consecutive R peaks, the RR interval, which indicates the time between heartbeats. With this information, we can also obtain the tachogram, which compares the time intervals between heartbeats.

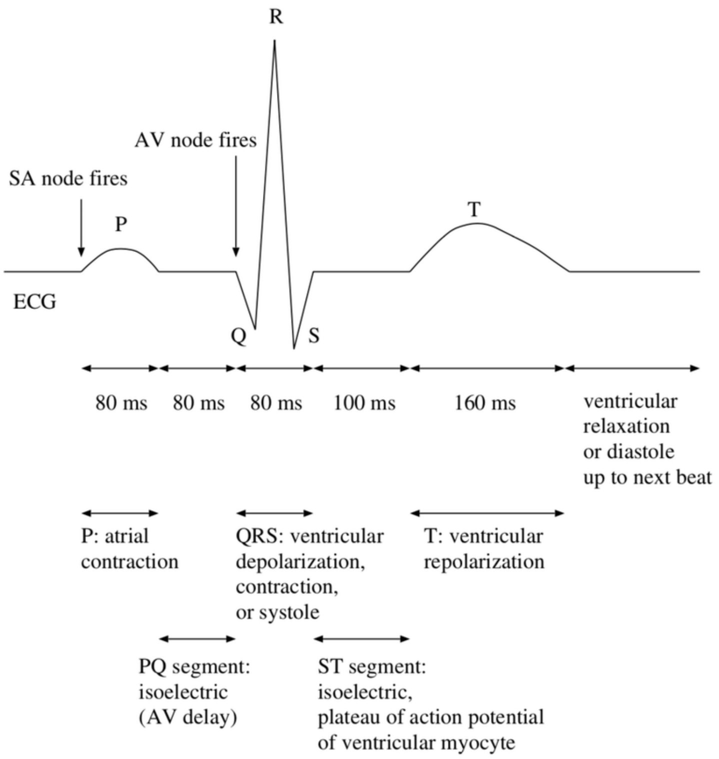

R
AV node fires
SA node fires
P
T
ECG
Q
S
80 ms
80 ms
80 ms
100 ms
160 ms
ventricular relaxation or diastole up to next beat
P: atrial contraction
QRS: ventricular depolarization, contraction, or systole
T: ventricular repolarization
PQ segment: isoelectric (AV delay)
ST segment: isoelectric, plateau of action potential of ventricular myocyte

Fig. 2.5.: Illustration of the ECG wave with the important peaks and intervals

One of the current main limitations in the measurement technique of this signal is that it takes a long time to carry out, because as it has been explained, a total of 10 electrodes (2.1) are needed to be located in some places that are not directly accessible, specially in technical placements of the electrodes (for instance, at the torax). This can also be a problem in clinical environments where time is gold in order to attend all the patients, and therefore the importance of optimizing the measurements with less and more comfortable electrodes, or even the possibility to make the measurements at home.

2.3 Pressure signals

There are many pulse wave techniques used to detect blood volume changes in the microvascular bed of tissue. As they all reach the same type of signals, this project will explain Photoplethysmography (PPG), which is a simple and low-cost optical technique. Unlike ECG that uses the electrical properties to obtain a signal, PPG is focused on pressure signals of the body. PPG is a non-invasive technology that uses a light source and a photodetector at the surface of the skin to measure the volumetric variations of blood circulation [10], often used for heart rate monitoring purposes.

The measurement of blood pressure is critical to the treatment and management of many medical conditions [11]. High blood pressure is associated with many chronic disease conditions, and is a major source of mortality and morbidity around the world. Green and red infrared light are often used to obtain PPG signals because of the difference in the wavelength; each light penetrates human tissue differently. Infrared light has the deepest penetration ability, and it can reflect the blood pulse from deep tissue. This is why the detection of blood pressure, atherosclerosis, blood sugar, and other physiological parameters uses the infrared light to obtain PPG signals.

Nevertheless, recent technologies are not very reliable in blood pressure measurements from a pulse waveform, so this project will not study the blood pressure, but the pulse waveform propagation through the body.

2.3.1 The pulse waveform

The PPG waveform comprises a pulsatile (AC) physiological waveform attributed to cardiac synchronous changes in the blood volume with each heartbeat, and is superimposed on a slowly varying (DC) baseline with various lower frequency components attributed to respiration, sympathetic nervous system activity and thermoregulation [12]. The AC component depicts changes in blood volume, which are caused by cardiac activity and depend on the systolic and diastolic phases [10]. The systolic phase (also known as rise time) starts with a valley and ends with the pulse wave systolic peak. The pulse wave end is marked by another valley at the end of the diastolic phase. Some features such as rise time, amplitude, and shape can predict vascular changes in the bloodstream. Additionally, PPG can be used to measure the variations between heartbeat time intervals (Peak-to-Peak or P-P Interval). The variation can be due to many factors such as the individual's age, heart conditions, and physical fitness. The PPG signal is divided into two unique phases: the rising edge of the pulse called anacrotic, which primarily describes the systole, and the falling edge of the pulse called the catacrotic, which represents the diastole. Additionally, a dicrotic notch is typically visible at the catacrotic phase.

The second derivative of this signal, known as acceleration photoplethysmogram (APG), is typically used because it is an indicator of the acceleration of the blood. There are also a number of critical points that can be extracted from the second derivative wave of a PPG signal. These critical points can be used to detect and diagnose cardiac abnormalities.

One last thing that can be done is that with the ECG and pulse waves signals, we can measure some important time intervals between waves that give information about

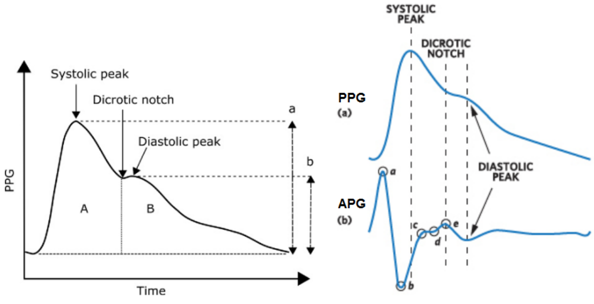


Fig. 2.6.: On the left, a typical PPG signal. On the right, the comparison between the PPG and APG key points.

the circulation of blood and the cardiovascular system. On each heartbeat, the blood ejected by the heart causes an increase of arterial pressure that propagates through the arterial tree from proximal to distal sites [13]. The velocity of this pulse is called pulse wave velocity (PWV), and it is related to the stiffness of the arterial walls, measured by dividing the estimated distance between these points by the difference of time arrival at two sites; this is the pulse transit time (PTT). As volume and flow are in phase on early waveform features, the arrival of the pressure pulse can be measured with the PPG, as it is the most widely used signal for that purpose due to its simplicity.

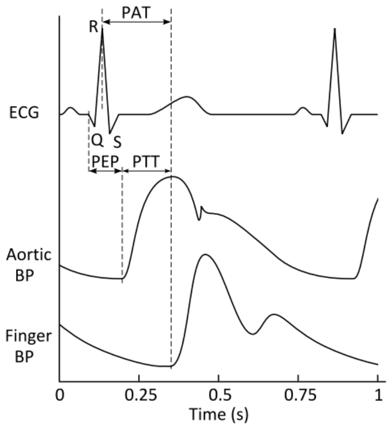


Fig. 2.7.: Representation of the important time intervals between ECG and PPG signals.

These three time intervals, the PTT, PAT, and the PEP, are widely used in research and clinical practice since several cardiovascular parameters of interest can be obtained from them. The PTT is a direct indicator of arterial stiffness, which is related to several factors such as ageing or hypertension and associated to many cardiovascular diseases including stroke and myocardial infarction. The PAT has proved to be useful for the detection of sleep disorders and autonomic nervous system activation. The PEP can be used to monitor left ventricular contractility, which provides valuable information about myocardial performance, and sympathetic activity under stable conditions of preload and afterload.

3 Hardware development

As it has been exposed in the introduction section, the aim of this project is to develop two acquisition systems: one specific for ECG signals and the other for general purpose. The idea is to design and implement low-cost and low-power devices which make a difference with respect to the commercial devices that cost thousands of dollars and have high consumption.

Both systems have been designed under the same main structure. Each one will have a specific integrated circuit which will acquire the signal from the body. This signal will be transferred to a microcontroller that will be programmed to receive and then send the data to the acquisition software, so that it can be processed.

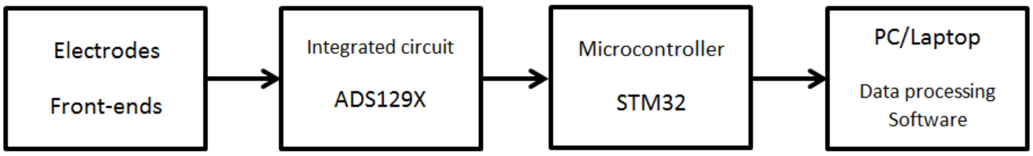


Fig. 3.1.: Block diagram of the acquisition system to be implemented.

First, prototypes of both systems will be done with commercial development kits, so that they can be studied and improved to the project's objective. Then, a PCB circuit board will be designed to replace the prototypes and get a final hardware.

3.1 ECG acquisition system

In order to have a final design, the start point is to work with the commercial development kit of the specific integrated circuit for ECG signals: the ADS1298 from Texas Instruments [14]. The first thing is to study the schematic and the jumpers setting from the user's guide provided by Texas Instruments [15]. This way, the different functionalities of the acquisition integrated circuit have been learned, which are the following.

- 8 channels with low-noise PGAs and high resolution 24-bit ADCs, with low-power consumption of 0.75mW/channel.

- Configurable data sampling rate from 250 SPS to 32 kSPS.

- Programmable gain of 1, 2, 3, 4, 6, 8 and 12.

- Supports systems meeting AAMI EC11, EC13, IEC60601-1, IEC60601-2-27, and IEC60601-2-51 Standards.

- Functionalities: Built-In Right Leg Drive Amplifier, Lead-Off Detection, Wilson Center Terminal, Pace Detection, Test Signals.

- SPI compatible Serial Interface.

The way of transmitting the ADS1298 data to a microcontroller is through SPI communication, so it is needed a microcontroller capable of this. It has been chosen an ultra-low-power and high performance microcontroller: the STM32L4R5 from the L4 family of STMicroelectronics [16] [17], which is precisely the low-power consumption family. For our project, it is a more than enough choice, because most of the microcontroller functionalities will be unused, so there exists the possibility to easily extend the functionalities of our platform. The microcontroller will be programmed in order to receive the signal data from ADS1298 through SPI communication, prepare the data to be sent including some headers for each channel so that they are separated, and transmit the data through UART communication to an Arduino circuit board.

An external circuit board which converts from UART to USB will be only used for the prototype, as a way of testing if serial USB communication is enough to transmit data to the computer software.

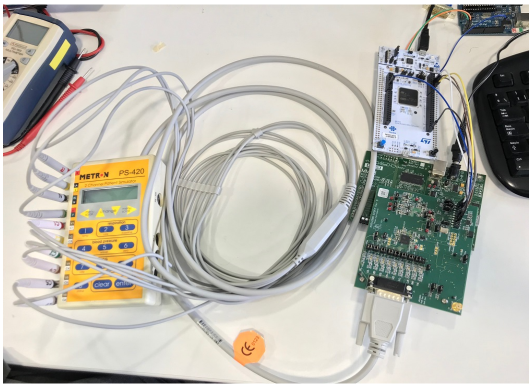


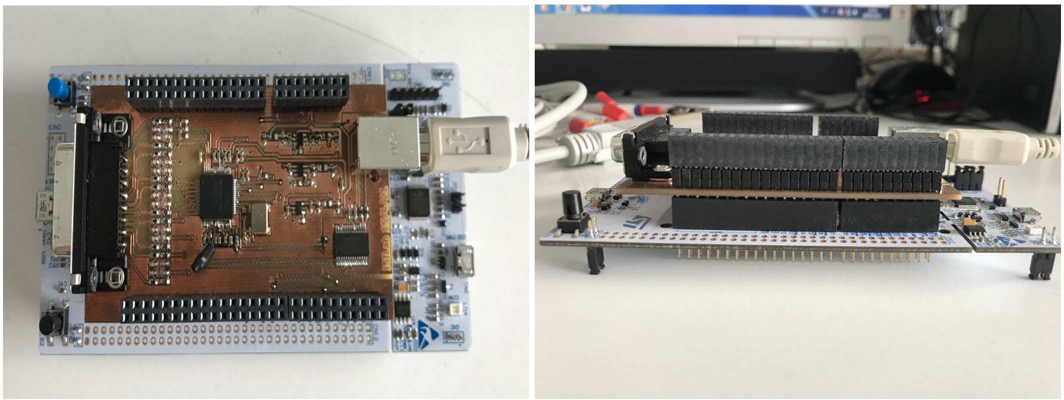
Fig. 3.2.: ECG acquisition system prototype.

Once we prove that the whole system works and data is received correctly, the next step is to design the final electronic circuit. It has been decided to build the hardware on the STM32 device, so that the designed PCB board can be placed over the microcontroller board as a shield. This way, the dimensions of the device will be those of the STM32 board, which are good enough. Therefore, the design of the PCB board must have the ADS1298 integrated circuit and the serial communication port.

The electronic circuit for the ADS1298 part must be adapted from the commercial development kit, as it will have a specific configuration and some parts of the original circuit will be removed. After a deep look at the schematics, a final design has been made, as we can see in A.1.1.

This schematic has been done with Eagle, a software of PCB designs. With the bill of materials of the circuit provided by the user's guide [15], the dimensions of the components used can be known in order to make a physical design of the PCB board (Fig. A.2). Once it is all designed, the next step is to print the PCB board on a copper plate; for this, the information of the design is passed to a machine that is capable of "printing" all the tracks and holes of the PCB board, using drills of different sizes.

Finally, the last step is to weld all the components in their respective positions, so that we end up with our final device.



(a) Upper view.

(b) Side view.

Fig. 3.3.: Pictures of the ECG acquisition device.

In this device, the data must be collected through a DB15 connector; this is because a specific cable is used, which has this type of connector, as well as the connection to the 10 electrodes necessary for a 12 ECG lead measurement.

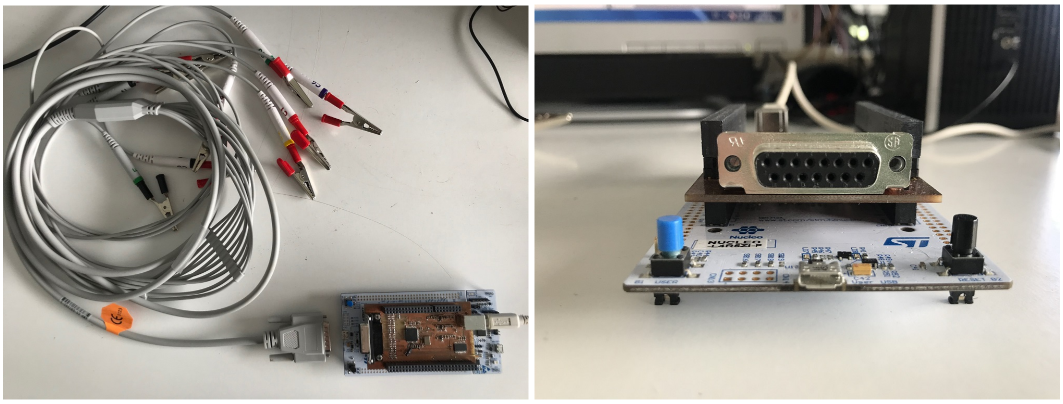


Fig. 3.4.: Illustration of the DB15 connector for the 10 ECG electrodes.

3.2 General acquisition system

The procedure of prototyping and ending up with a final design will be the same as explained for the ECG acquisition system. The difference between the previous one is that now we will use another integrated circuit: the ADS1299, also from Texas Instruments [18]. This integrated circuit is not specific of any particular signal, so it is ideal for generic use; with this one could measure ECG, PPG and other biological signals of the body. Taking a look into the user's guide of the development kit [19], the following characteristics are found.

- 8 channels with low-noise PGAs and high resolution 24-bit ADCs, with low-power consumption of 0.75mW/channel.
- Less input-referred noise than the ADS1298.
- Configurable data sampling rate from 250 SPS to 16 kSPS.
- Programmable gain of 1, 2, 4, 6, 8, 12 and 24. As it reaches a greater gain, an electroencephalogram signal can be acquired, which is a very small signal.
- Supports systems meeting AAMI EC11, EC13, IEC60601-1, IEC60601-2-27, and IEC60601-2-51 Standards.
- Functionalities: Built-In Bias Drive Amplifier, Lead-Off Detection, Test Signals.
- SPI compatible Serial Interface.

Another STM32L4R5 microcontroller will be used, as well as the same SPI and UART communication for data transmission. The difference between the two systems will be how the microcontroller is programmed in order to use the integrated circuit, as it will be explained in the next section. For the prototype, the external UART/USB conversion board will be still used so that we can prove that the system works correctly.

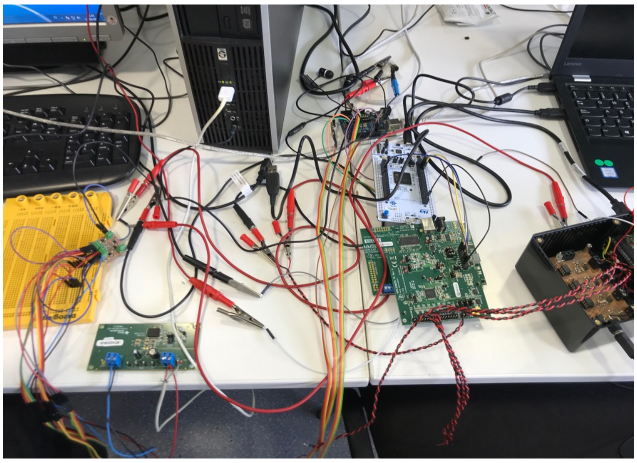
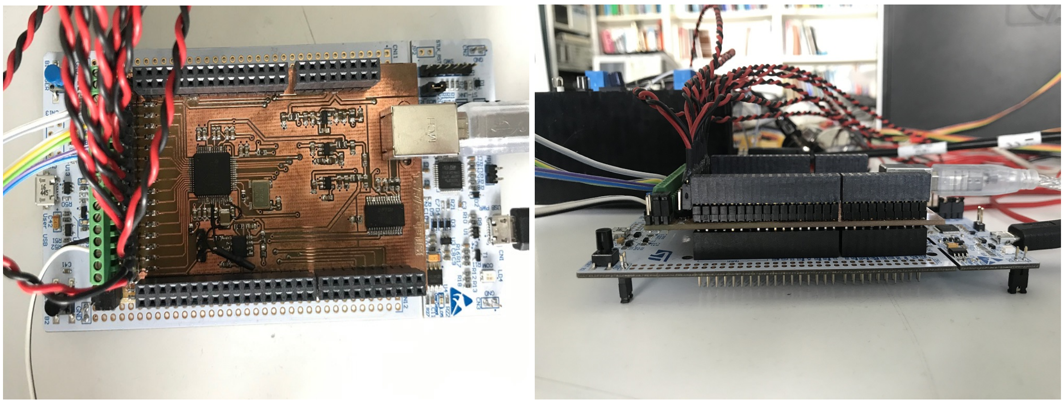


Fig. 3.5.: Generic acquisition system prototype.

As we proceeded in the ECG system, the next step is to design the final circuit by examining the original schematic of the development kit. Luckily, most of the circuit is practically the same as the previous system, so we only had to do some modifications of the final schematic of the ECG system, resulting in the schematic seen in A.1.2.

Like before, we get the bill of materials from the user's guide [19] so that we know the dimensions of the components and we make a physical design of the PCB board (Fig. A.3). Then, we print the design in a copper plate, and finally we weld all the components in their respective places. All in all, we end up having our final device.



(a) Upper view.

(b) Side view.

Fig. 3.6.: Pictures of the Generic acquisition device.

In this case, the data is collected from normal wires, so a specific connector is not necessary. This is why the device is generic, as any signal can be connected through these wires. Thus, a series of front-end boards can be used, in order to amplify the electrical signal so that they are more clearly seen. These front-ends have been developed by the research group for some of the biological signals that are typically acquired; then, we dispose of front-ends for the ECG and PPG signals in which this project will be focused.

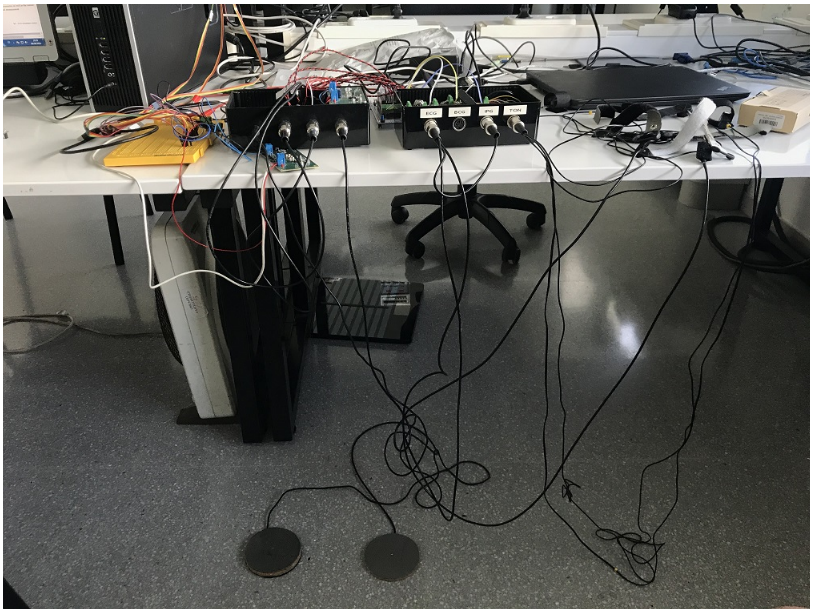
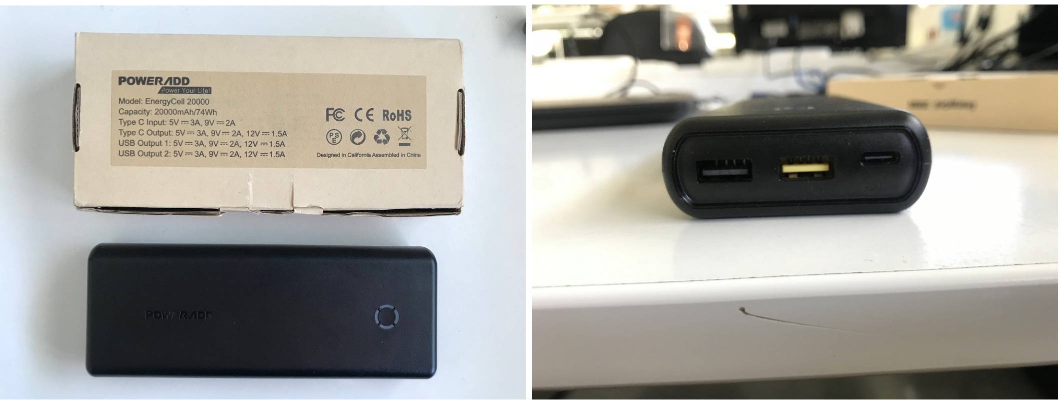


Fig. 3.7.: Different front-end devices developed by the research group, used to acquire different biosignals.

One last thing to take into account is the dimensions of our devices, which are 13.3 cm large, 7 cm wide and 3.5 cm height.

3.3 Power supply

The prototypes can be powered up either with the USB connection for the microcontroller, and with the electric grid through a typical transformer. For electrical security, it has been thought that a battery power supply will be better, as well as the possibility of it being portable. Two options are on the table: it can either be through the USB from the laptop (not connected to the electric grid), or with an external battery. We have seen that the measures with the power supply from the computer add some interference frequencies, whereas a power bank does not. Therefore, it has been decided to use a power bank that will supply the system, as it is a great choice for the acquisition of better quality signals.


POWERADD
Power Your Life!
Model: EnergyCell 20000
Capacity: 20000mAh/74Wh
Type C Input: 5V ⎓ 3A, 9V ⎓ 2A
Type C Output: 5V ⎓ 3A, 9V ⎓ 2A, 12V ⎓ 1.5A
USB Output 1: 5V ⎓ 3A, 9V ⎓ 2A, 12V ⎓ 1.5A
USB Output 2: 5V ⎓ 3A, 9V ⎓ 2A, 12V ⎓ 1.5A
FC CE RoHS
PSE
Designed in California Assembled in China
POWERADD

(a) Upper view.

(b) Side view.

Fig. 3.8.: Pictures of the power bank.

As we can see in 3.8a, it has a capacity of 20000 mAh, and 2 USB outputs of 5V, which is the voltage needed to supply the acquisition devices. In further section we will see that this power bank is more than enough, through a power consumption test of the devices.

The power supply is needed for two different parts. First of all, it is used to supply 5V to the electronic board designed, which creates and distributes the different voltages needed in the integrated circuit to be functional (5V and 3.3V as it can be seen in A.1). Secondly, the power bank also supplies a circuit board that generates ±5V in order to supply the different specific front-end and external devices provided by the research group so that different biosignal measurements can be carried out.

Software development

4

A total of three different software programs have been used for the acquisition systems: STM32CubeIDE in order to program the microcontroller, LabVIEW for receiving and saving the data, and MATLAB so that this data can be processed, filtered and represented, as well as for getting information and biological parameters. Details on how each of these programs have been used are explained below.

4.1 Programming the microcontroller

As the microcontroller is from STMicroelectronics, specifically from the STM32 family, the website provides a development platform dedicated to this kind of microcontrollers: STM32CubeIDE. This development environment is capable of different functionalities, such as peripheral configuration, code generation, code compilation, and debug features. With this, one can fully program the microcontroller so that it works as it has been designed. In the next paragraphs are exposed some of the features that have been programmed in the STM32R5L4.

The first thing is to select the peripheral configuration of the microcontroller. For this, the development platform is very user-friendly in a way that it is easy to use, due to it is very visual as we can see in Fig. 4.1. Here is a list of the implemented configuration.

- The clock configuration is set at 48 MHz either for the SPI and UART communications, so that there is enough clock speed for their requirements.

- In connectivity, the SPI is activated and configured with 8 bits data size, and three pins which correspond to the SCLK (Serial Clock), MISO (Master Input Slave Output) and MOSI (Master Output Slave Input) signals. In these systems, the master is the microcontroller, whereas the slaves will be the acquisition integrated circuits.

- The UART feature is also activated, with 8 bits data size as well, and with a baud rate (speed) of 921600 bits per second. Different baud rates have been tested until finding the most efficient. Two pins are configured, corresponding to the TX (transmision data) and RX (reception data) signals.

- Two GPIO pins are also configured. One of them will be used as an output which is active low, corresponding to the CS (Chip Select) signal of the SPI communication. The other one is configured as an active low EXTI (External Interrupt), which will be associated with the DRDY (Data Ready) signal of the SPI.

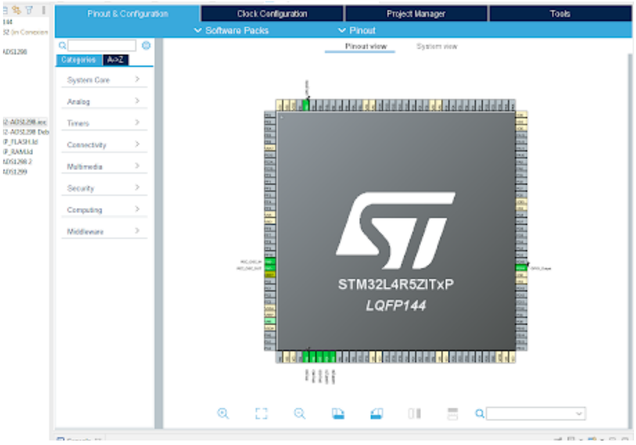


Fig. 4.1.: Interface of the STM32CubeIDE pin configuration.

Once we complete the peripheral configuration, the platform automatically creates a main code in which all the pins are configured as done in the previous interface. Some "User Code" spaces are also created, so that users can write their own code. This is where we will program what we want the microcontroller to do.

The code is based on an example found in GitHub [20], and adapted to the project's purposes. It calls a first function in which the integrated circuits are initialized, as they need a power-up sequencing. The ADS1298/99 can be configured through SPI and commands as it can be seen in the user's guide, where different registers can be read or written in order to configure the data acquisition. In the initialization, these commands are used so that the different registers are written as we want.

After the initialization, the code enters a loop in which it is constantly waiting for the receiving of a new configuration, which comes from the LabVIEW program through UART communication. How this new configuration is built will be explained in the LabVIEW section. When a particular configuration is received, the program starts to read the data from the integrated circuits, waiting for the interrupt (DRDY) signal to be active, which indicates that new data is ready to be read.

Once the data is read, the code transmits the data in a particular construction in order to separate the information of each one of the 8 channels. For this, before sending the channel data, a string of the type “DTX” is sent, where X indicates the channel number (X = 0,1,...7). Thus, the information sent for the first channel (channel 1) will be “DT0(data)”, so that the channel separation is accomplished. It is important to remark that the data is sent as characters. This will be then processed by the LabVIEW program.

The reading process is executed in a loop, until a new configuration is received. This is the way our microcontroller is programmed in order to receive data from the integrated circuit, and then send this data to the next step.

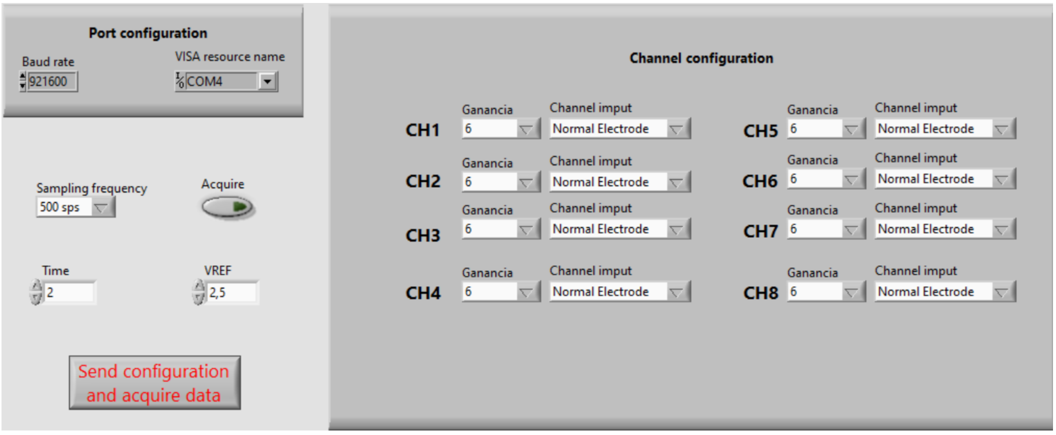
4.2 Data acquisition software

In the previous section, it has been explained how the microcontroller is programmed in order to receive and transmit the data. Now we want to be capable of reading these data in order to be saved in a file, and have a first representation of what we get. For this, we have chosen LabVIEW, a development environment which works with a visual programming language. This means that there is no code for the user to write, but a graphic user interface in which different components or virtual instruments can be placed.

It has two parts: the front panel, where the user can interact with the configuration of buttons and options, as well as seeing graphics; and the block diagram, where the features are programmed. It is done as a sequence of things that have to be executed to allow a correct data acquisition. The sequence goes as follows.

First, an initial configuration is sent through UART Serial Port communication to the microcontroller, so that it initiates correctly. Then, depending on the configuration of channels and sampling frequency chosen at the front panel, different configurations can be sent to the STM32 so that it can configure the ADS1298/99 as we want. If the “Acquire” button is activated in the front panel, the UART communication will be programmed to constantly read what the microcontroller is sending, indicating as well the total time that we want to receive data.

As the data received from the STM32 are all characters, what we do is to convert these characters into numbers, in order to make it easier to manipulate the data. This creates an array of hexadecimal numbers, which we recall that are separated by the corresponding tags for each channel. The importance of the “DTX” tags explained before is that now we can separate the data by searching for the corresponding numbers; for instance, if we want to find the data of channel 1, we know that we must find the hexadecimal numbers of the characters “D T 0”, which are “68 84 48”, and the four next numbers will be the data as we are using 32-bit channels (one hexadecimal number is 8 bits). After finding these 4 numbers of data, they are transformed into a decimal number, which at the same time is normalized with respect to the reference voltage, so that we get meaningful voltage information. This way all the information is separated in 8 different arrays, one for each channel, and this information is saved in an excel file so that it can be processed in MATLAB.



Port configuration

Baud rate: 921600

VISA resource name: COM4

Sampling frequency: 500 sps

Time: 2

VREF: 2,5

Channel configuration

	Ganancia	Channel imput
CH1	6	Normal Electrode
CH2	6	Normal Electrode
CH3	6	Normal Electrode
CH4	6	Normal Electrode
CH5	6	Normal Electrode
CH6	6	Normal Electrode
CH7	6	Normal Electrode
CH8	6	Normal Electrode

Fig. 4.2.: LabVIEW interface of the data acquisition program.

Before the next step, we represent in LabVIEW the different channels so that we take a first look into the data received, and evaluate if it has been a good enough measure or it has to be repeated because there has been an error. Once everything is correct, we can take the data into the final step.

4.3 Data processing, representation and parameters

In order to get the final representations and biological parameters from the data acquired, it is necessary to process the signals. This means that all the data must undergo some filtering processes, so that the signals can be clear and then one could extract the information of the biological parameters.

This section will be all implemented in MATLAB, with the final objective of creating a program with the App Designer tool. It is important to remark that most of the codes are inspired and adapted from a previous work on this topic [6].

4.3.1 Signal filtering

The first thing to do with the acquired data is to pre-process the information, due to the fact that the real world adds some undesired artifacts and noise which should be removed. As the ECG is a cyclic signal, it is governed by a set of frequencies that correspond to the different parts of the heart cycle. For instance, comparing the ECG frequencies with the common frequencies of artifacts and noise, it can be seen how these interferences affect directly to the signal.

Source	Frequency
Heart rate	0.67 - 5 Hz (40 - 300 bpm)
P-wave	0.67 - 5 Hz
QRS	10 - 50 Hz
T-wave	1 - 7 Hz
High frequency potentials	100 - 500 Hz

(a) Common ECG frequencies

Source	Frequency
Muscle	5 - 50 Hz
Respiratory	0.12 - 0.5 Hz (8 - 30 bpm)
External electrical	50 - 60 Hz
Other electrical	typically > 10 Hz

(b) Common noise and artifact frequencies

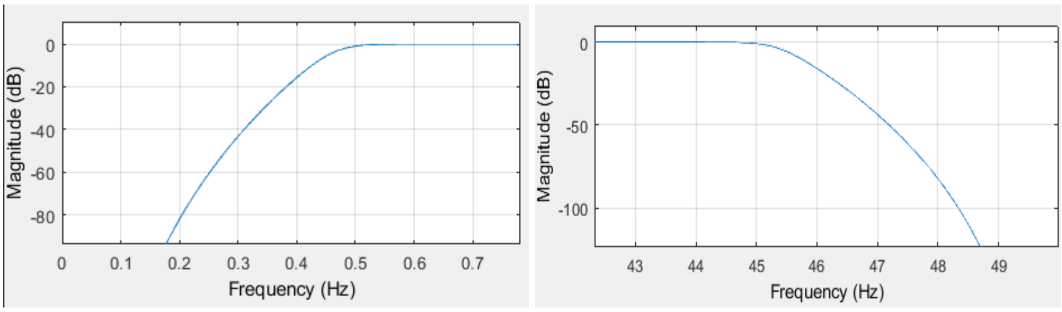
Tab. 4.1.: The most important frequencies of an ECG signal

In order to see the spectrum of a signal, MATLAB offers a Fast Fourier Transform function, which represents the different existing frequencies of the signal. With this, we can see if the undesired frequencies are present, so that we can apply filters for at least reducing these interferences, and try to obtain the cleanest signals possible. Doing some research, we find out that the typical filters used for biological signals are high and low pass filters, and notch filters [21].

High and low pass filters

On the one hand, low-pass filters on the ECG are used to remove HF muscle artifacts and external interference. They typically attenuate only the amplitude of higher frequency components. On the other hand, high-pass filters remove LF components such as motion artifact, respiratory variation, and baseline wander. The combination of both the high and low pass filters actually result in a bandpass filter, allowing only a certain frequency band to pass through.

In MATLAB, there are many ways to implement this kind of filters, and the one chosen is through a Butterworth filter, as it is done in [6]. This filter is characterized by offering a frequency response as flat as possible in the passband, as the filter order is increased. In our project, 5th order Butterworth filters have been used as high and low pass filters, with 0.5 Hz and 45 Hz cut-off frequencies, respectively. With this, we get rid off respiratory and other electrical noises in higher frequencies, but doing



(a) Low-pass filter.

(b) High-pass filter.

Fig. 4.3.: The combination of (a) and (b) result in the Band-pass filter.

again the spectral representation with the Fast Fourier Transform, we still see that the external electrical frequency at 50 Hz has not been entirely attenuated.

Notch filter

The purpose of this filter is to resolve the 50 Hz interference. It is basically a stop band filter which attenuates a specific frequency, removing then the external electrical frequency, or any other undesired interference that could be seen in the spectral representation. In particular, MATLAB offers a 2nd order Infinite Impulse Response (IIR) Notch filter, which is the one used in the project.

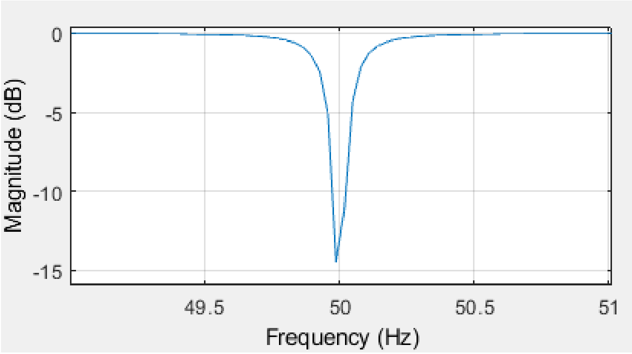


Fig. 4.4.: Notch filter at the 50 Hz frequency.

It is important to take into account that all filters introduce distortion in the resulting output signal. The distortion can be in amplitude or phase; basically, they distort different frequencies differently causing this distortion. This is why it is not always convenient to apply all the filters, as sometimes applying only the bandpass, the notch, or none, one can obtain clearer signals. The idea of the pre-processing is to analyze which case gives the best performance, so that accurate information can be then obtained from the signals.

4.3.2 Signal representation and automatic parameter extraction

After the filtering and synchronization processes, the signals can be finally represented in a MATLAB plot. Recall that the data acquired from the LabVIEW program is given as a value for every sample, and this value corresponds to the electrical potential in a magnitude of Volts. Therefore, if we represent directly these data, we would obtain a plot of the electrical signal but with respect to the samples. The idea is to get the plot with respect to the time, so what is done is the conversion of samples into time, by means of the sampling frequency. This is:

$$t = \frac{N}{F_s} \tag{4.1}$$

where N is the sample, F_s the sampling frequency, and t the time associated with the sample. Thus, we obtain the graphical representation of the electrical signal with respect to the time of the measure.

The next step is to get the typical biological parameters of the electrical signals, which can give to the user some information about the biological status of the body. For this, a series of algorithms will be used, as we can find in the literature of this topic.

Pan-Tompkins

The first algorithm to take into account is the Pan-Tompkins algorithm, which is commonly used to detect QRS complexes in ECG signals. The reason for finding this particular part is that it is the main peak of the signal, so it is easily detected, as well as it is commonly used for heart rate monitoring. [22] The Pan-Tompkins algorithm applies a series of filters to highlight the frequency content of this rapid heart depolarization and removes the background noise.

For this, high and low pass filters are needed, which have been already implemented, but a derivative filter is also required. This last filter accentuates QRS complexes relative to P and T waves, providing information of the QRS complex slope.

$$y(n) = \frac{1}{8}[2x(n) + x(n-1) - x(n-3) - 2x(n-4)] \tag{4.2}$$

Then, the algorithm squares the signal to amplify the QRS contribution. This is done to amplify the output of the derivative filter, emphasizing the higher frequencies and setting all values into positive ones. Finally, it applies adaptive thresholds to detect the peaks of the filtered signal; this is done with a moving-window integrator. It smooths the spiky signal coming from the squaring output, and it is applied to obtain waveform feature information. The window has the following equation:

$$y(n) = \frac{1}{N}[x(n-(N-1)) + x(n-(N-2)) + ... + x(n)] \tag{4.3}$$

where N is the number of samples in the width of the integration window, which should be the width of the QRS complex (about 150 ms).

This algorithm disposes of a decision rule in order to validate if the QRS complex found is valid, which is performed with an adaptive thresholding operation [6]. First of all, a set of samples are produced at the location of the maxima in order to localize the candidates to QRS complexes. Then, the algorithm uses two threshold values (signal and noise thresholds) to evaluate whether the maxima is or not a QRS complex. The algorithm also eliminates physically impossible events (such as two QRS complexes within less than 200 ms), and therefore removing false positives. Additionally, if an unreasonably long period has expired without an above threshold peak, the algorithm will assume a QRS has been missed and perform a search back. Finally, if a QRS candidate occurs after the 200 ms period but within 360 ms of the previous QRS, the algorithm determines whether it is the QRS complex of the next heartbeat, or a possible anomaly of the T wave.

P, Q, S and T peak detection

With the Pan-Tompkin algorithm, the QRS complexes are determined by finding the maxima of the QRS peaks, so in practice only the R peak indexes are found. The objective of the biological parameters is to evaluate the different intervals of time between all the signal peaks, so we should as well find the P, Q, S and T peaks. This code is based on [23].

The idea of these detections is to find the maxima or minima in strategic windows of the signal, starting from the Pan-Tompkin algorithm in which we can know the position of the R peaks. Doing some research about the typical position of each peak with respect to the R peak, the windows can be easily selected, and once we are in the specific region of each peak, we can find the maxima or minima; for the P and T peaks the maxima are found, whereas for the Q and S peaks the minima are seeked.

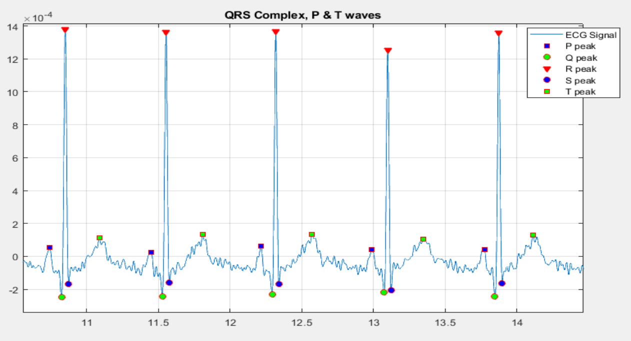


Fig. 4.5.: P, Q, R, S and T indexes found with the algorithm.

At the end, all the peak position indexes are found, which can be taken as samples or times (it can be converted from one to another by means of the sampling frequency).

The only thing left to do is the time intervals, which is easily done once we obtain all the indexes; these intervals are directly the difference of time between certain peaks. The typical intervals which are found for medical information are the ones presented in Fig. 4.6. The main interval could be the R-R interval, as it takes into account the time between heartbeats, and therefore with which one can know the heart rate in beats per second.

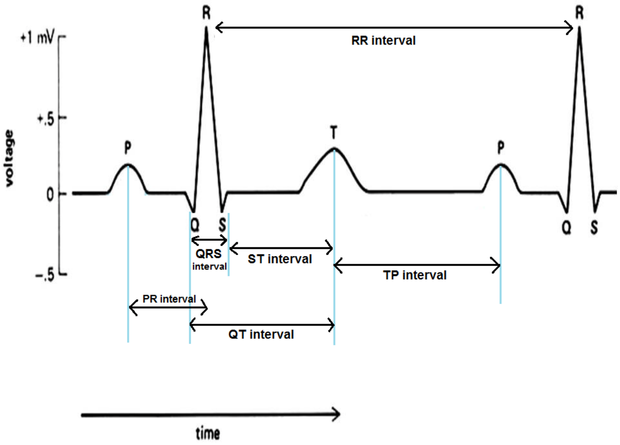


Fig. 4.6.: Representation of the important intervals of time in the ECG signal.

Pressure signal analysis

Apart from measuring ECG signals, the aim of the generic device is to be capable of measuring other biological signals. This project will focus on the PPG signal, whose waveform is different from the ECG, so another algorithm will be needed in order to find the key features of this signal. These key points will be the systolic peak, the dicrotic notch, the maximum slope and the valley where the systolic peak begins.

As for the ECG signal, the first step is to seek a relevant point for each heartbeat and then find the others with the help of windows. The first tests tried to find the systolic peak, so that it was similar to the Pan-Tompkin algorithm. However, it was found out that depending on the PPG signal, it was difficult to find some of these peaks because some of them did not reach the same amplitude as the others, and they were indetectable. Therefore, another strategy was thought: finding the maximum slope points, which are always just before the systolic peak. For this, the first derivative is needed, because finding the maxima of the derivative will directly provide the maximum slope points of the PPG signal, and then a reliable starting point to find the rest of the key points.

Once the maximum slope indexes are found, a series of windows are used to find the other key features. The systolic peak can be easily found by seeking for the maxima of the particular window just after the maximum slope point; the same for the valley where this peak starts, by finding the minima. The trickiest one is the dicrotic notch, which will use the first derivative as well. The literature [24] suggests that the dicrotic notch can be detected as follows: taking the first derivative, from the maxima point seeked for the maximum slope, a window is used in order to find the minima just after this maxima point, which will correspond to the descendent slope after the systolic peak. From this minima, another window is taken in order to find the following maxima; this new peak in the derivative corresponds to the dicrotic notch point.

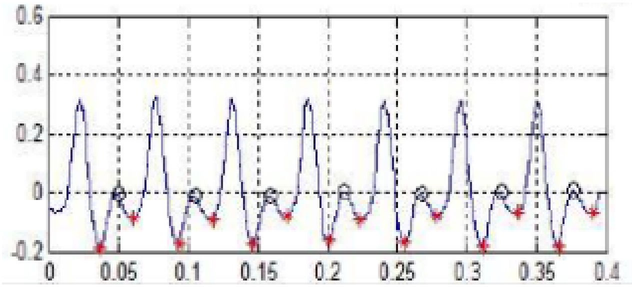


Fig. 4.7.: First derivative of the PPG signal. In black, the peak corresponding to the dicrotic notch.

With all the indexes found for the different key features, the last thing to do is to get the parameters. In this case, the time intervals are found as well, but for the PPG signal it is also important to get the amplitudes of the points, because the areas are directly related to the blood pressure and they give meaningful medical information about the patient.

The time and amplitude intervals considered are: from minimum to maximum slope, from minimum to maximum, from slope to maximum, and from maximum to dicrotic notch. The intervals between the same points but from two consecutive heartbeats are also considered.

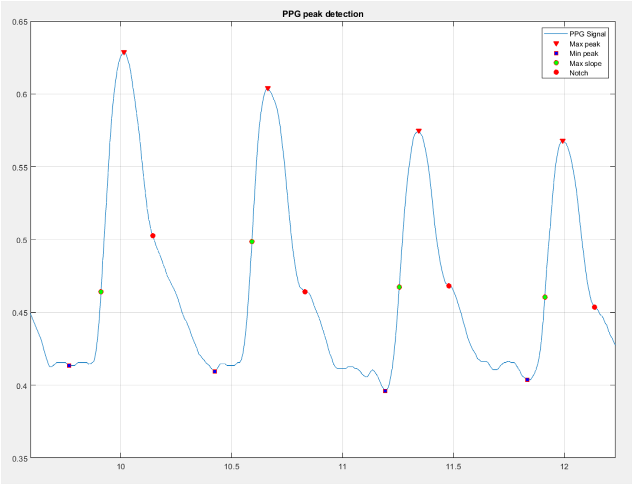
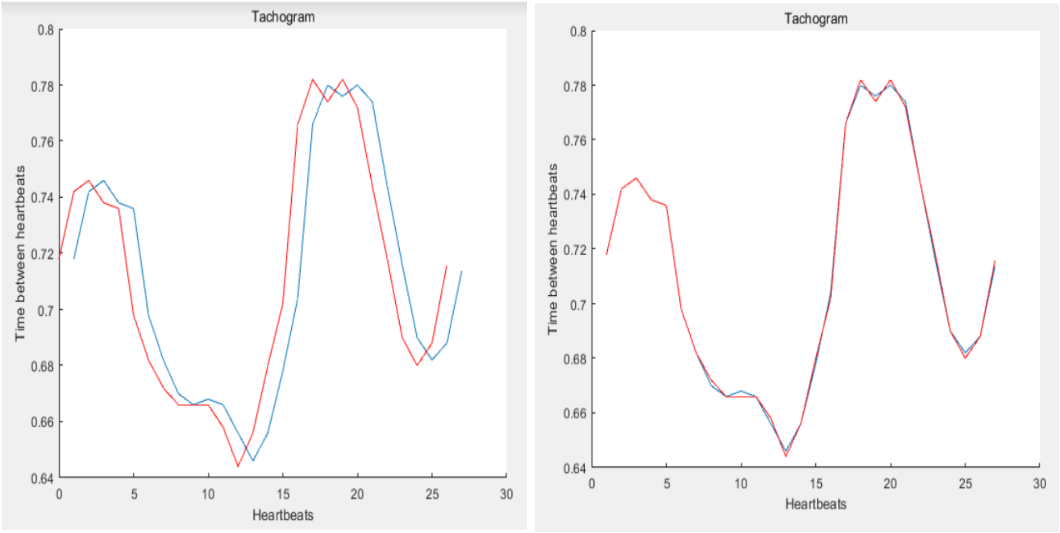


Fig. 4.8.: Representation of the important points in the PPG signal.

4.3.3 Synchrony between devices

For the ECG processing, before getting the parameter information, there is a previous step to be done. One of the approaches of the project is to be able to obtain the 12 ECG leads from a subset of some of these leads, by means of transform matrices as it will be explained in further sections. For this, we need to acquire the data from both devices, and this could add some time delay between the signals, as they could be acquired at different instants of time.



(a) Tachogram with one heartbeat delay.

(b) Synchronized tachogram.

Fig. 4.9.: Representation of the tachogram synchrony.

In order to solve this problem, a synchronization process is carried out, which uses the tachogram of the signals of both devices. The tachogram is built by finding the heartbeat peaks (R peak) with the Pan-Tompkin algorithm explained before, as it is the clearest peak of any biosignal, so that the difference between two consecutive heartbeats can be computed. This tachogram signal is the same for the acquired data from both devices, so the way of solving the delay problem is to represent both tachogram signals and see how much delay there is between them. Once we know the delay, represented by a number of heartbeats, we update the original signals so that each of them start from the same heartbeat. Therefore, we will have the ECG signals synchronized and ready to be compared.

4.3.4 Transform matrices

One of the main and most interesting objectives of this project is to be capable of obtaining the 12 conventional ECG leads from a standard, or even from a not-standard subset. Then, it is necessary to pass from a smaller dimension of data to a greater one; in algebra, this is the same as thinking of certain matrices that contain the required information to reach this dimension increment. Thus, there must exist transformation matrices which relate the source leads (the acquired subset) with the target leads (the 12 ECG reconstructed leads).

This project will use three different methods to obtain these matrices, based on [6]. This is one of the advantages of working with the same research group, as the codes are inherited and it is not necessary to start from nothing, but only adapt the codes to our purposes.

Types of matrices

First of all, it is interesting to know about the different types of matrices that can be obtained. Most of them are fixed and do not adapt to changes over time, but there are also adaptive matrices. However, the latter ones do not manifest any improvement with respect to the non-adaptive matrices, but in practice they are better to deal with measures that change in the heart rate or movement.

The first type of matrix is the universal transform based on statistical data or models, which can be applied to any patient regardless of their condition. It is logical to think that these matrices are not very reliable due to the variance that can produce a generalization across patients, as there will be differences between them such as gender, age, or diseases.

Secondly, there are population-specific transforms, which are more effective than the universal ones. The idea is to separate the patients into different groups that will have similar characteristics, depending on the age, gender or possible diseases. Nevertheless, other important properties of the heart position or orientation can not be taken into account, which are directly related to the signals received, so there are still some uncertainties about the reliability of these matrices.

The last type is patient-specific transforms, which are the ones with the best performance as one could expect. This is the same as considering that every person is different and with unique characteristics, and therefore each one must have their own matrix. The project attaches to this last type of transforms, so that each of the three methods obtain the specific matrices for each patient.

Methodology

Three different methods will be used in order to see which of them give the best results. They all take 3 beats to synthesize the matrices needed later to reconstruct the signal. The methods are the following.

- Method 1

  This method will use the Singular Value Decomposition (SVD) in order to obtain the transform matrices. The SVD is useful to decompose large n x m matrices into three matrices, so the first step is to decompose the matrix containing both the source leads and the target leads measured. In the same way, the matrix containing only the source leads is also decomposed, so that we end up with:

  $$A_{n\times m} = \hat{U}_{n\times m}\hat{\Sigma}_{m\times m}\hat{V}^{T}_{m\times m} \tag{4.4}$$

  $$A'_{n\times p} = \hat{U}'_{n\times p}\hat{\Sigma}'_{p\times p}\hat{V}'^{T}_{p\times p} \tag{4.5}$$

  where n is the number of samples, m the number of total leads and p the number of source leads (the subset). U and V matrices, corresponding to the entire set of leads, must be kept as they carry the information of the 12 leads, specifically information about the shape and the leads themselves.

  In order to pass the subset of leads into a higher dimension, a matrix T^1 is defined to relate the information of $\hat{\Sigma}$ with $\hat{\Sigma}'$:

  $$\hat{\Sigma}_{m\times m} = T^{1}_{m\times p}\hat{\Sigma}'_{p\times m} \tag{4.6}$$

Notice that $\hat{\Sigma}'$ is presented with a different number of columns so that it fits to the dimensions of $\hat{\Sigma}$. With this relation, we can obtain T^1 as follows:

$$T^1_{m\times p} = \hat{\Sigma}_{m\times m}\hat{\Sigma}'^{-1}_{p\times m} \tag{4.7}$$

The last step is to reconstruct the 12 leads using the U, V, and T^1 matrices, as well as the $\hat{\Sigma}'$ matrix to transmit the data from the source leads. It goes like this:

$$A_{n\times m} = \hat{U}_{n\times m}(T^1_{m\times p}\hat{\Sigma}'_{p\times m})\hat{V}^T_{m\times m} \tag{4.8}$$

- <u>Method 2</u>

This second method directly relates both dimensions of the source and target leads, so that the target leads are a linear combination of the source leads. The direct transform matrix T2 makes the following relation.

$$A_{n\times m} = A'_{n\times p}T^2_{p\times m} \tag{4.9}$$

This equation is solved by applying a Least Squares Method (LMS) that uses SVD as well. Notice that for this method it is not necessary to reconstruct the leads with 3 beat periods, but it is applied in a way that the correlations at intervals of 3 beats can be compared to other methods.

- <u>Method 3</u>

The last method is a combination of the previous two methods; the source and target leads matrices are still being decomposed by SVD as in method 1, but now the diagonal matrices $\hat{\Sigma}$ and $\hat{\Sigma}'$ are related by a linear combination of their singular values.

$$\sigma = \sigma'\mathbf{t}_{p\times m} \tag{4.10}$$

The solution to this equation is solved by LMS. Finally, in the building phase the leads are reconstructed as:

$$A_{n\times m} = \hat{U}_{n\times m} \begin{pmatrix} \sigma' t_1 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & & 0 \\ \vdots & \ddots & \sigma' t_i & \ddots & \vdots \\ 0 & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \sigma' t_m \end{pmatrix} \hat{V}^T_{m\times m} \qquad (4.11)$$

Algorithm

Once all the methods are taken into account, the steps from the beginning to the end are the following.

First of all, we get all the necessary data: the target 12 leads from the ECG device, and the subset of signals from the generic device. In this project, a subset of 4 source leads are considered, containing Leads I, II and III, and a not-standard lead (such as the electrical signal measured in the neck or at the waist). With this data, a matrix A is constructed with both the source and target leads, which is then filtered in order to work with clean information.

Then, this data goes through a Training Phase process, which first uses the Pan-Tompkin algorithm to localize the beat position so that the 3 beat period can be applied. Then, it is proceeded by finding the different transform matrices of all three methods, as well as the U and V matrices which contain the leads information.

After this, the index of the last beat of the signal is found, so that the number of 3 beat intervals can be known. This will be important so that all the information can be stored in vectors with a length corresponding to the number of intervals, which are initialized in a preallocation phase.

Finally, the algorithm ends with a Building Phase process, in which the 12 lead data is reconstructed from the matrices found in the training phase, and the subset of source leads. Three different sets of 12 leads are obtained, corresponding to the three different methods; the idea is to compare them through the correlation between the original 12 lead data, and the 12 leads reconstructed.

4.3.5 Final program

All the data processing, representation and parameter processes are synthesized in a final software program, with the objective of it being user friendly and simple to use; the user does not have to know how the codes work, only the functionalities that this program will provide. MATLAB offers the possibility of creating an application program with the App Designer tool [25]. In this tool, there is the possibility to implement our codes with the interaction of the user with the interface, similar to LabVIEW. The design of the program can be personalized as well, in order to offer the desired functionalities through menus, buttons, checkboxes... The program designed will have the following functionalities.

Main menu

In this window, the user can load the data acquired from the LabVIEW program as an excel file, for both the ECG and Generic devices. Other configuration settings can be changed, such as the sampling frequency, the filters to be used (bandpass and notch), and the buttons which start the ECG and other signal representation and parameters. There is also the option to change a threshold value in case that the code has an error getting the parameters; this value should be changed from 0.1 to 0.5. See Fig. A.4.

ECG menu

The first tab (Fig. A.5) of this menu allows the user to see the synchrony between the loaded data, and try to find the delay time manually. For this, the tachograms are represented together in a plot so that it is easy to see if they present delay or not.

Then, there are three tabs corresponding to the visualization of the limb leads, the chest leads, and the generic signals acquired (Fig. A.6). All the plots of the program have three different tools: a zoom button, with which the user can amplify a certain zone of the graphic; a pan button, which permits to drag the plot and move; and a reset button, which restarts all the plots.

Finally, there are two tabs in which the parameters of the 12 leads and the generic signals are obtained. These parameters are represented in tables, and only the mean and standard deviation of the intervals for each lead, in milliseconds (Fig. A.7). However, there is the option to save not only the mean and standard deviation data, but also all the parameter vectors used to get these values, so that the user can then use all the data for deeper analysis.

Pressure Signals menu

In this section there is a first tab in which different pressure signals are represented, and then the tabs in which the parameters are shown (Fig. A.8). In this case, we get the mean and standard deviation of the times, and also the amplitudes, as it has been explained in section 2.3. Finally, there is a tab in which the important time intervals explained before are calculated and represented in a table for each heartbeat (Fig. A.9).

ECG reconstruction menu

The last functionality of the program is to elaborate the ECG reconstruction as it has been exposed in the transform matrices section. The first tab contains the configuration options for the training and building phases (Fig. A.10). For the training phase the user can click on the Train button to start seeing the best method to use, and then save the transform matrices, whereas for the building phase the user can load existing transform matrices and then reconstruct directly from these matrices.

At the same time, a training plot is incorporated in order to compare the reconstructed signal with the original one, as well as the correlations of all three methods. This way, it is easy to choose which method gives the best performance. Once the method is chosen, the reconstructed leads are represented in the next two tabs of the menu.

In conclusion, we end up with a final complete program which synthesizes all the algorithms, codes and processes executed for the different functionalities carried out in this project.

5 Measurements, results and discussion

The last thing left to do in this project is to prove that our acquisition systems are functional and that good quality results are obtained as well. First, a power consumption test will be carried out in order to see if the low-power specifications are reached, and the costs of production will be discussed. Then, a series of measurements will be done to different subjects, and for different biosignals, to show that the devices work as expected.

First of all, different test measurements have been done in order to see that effectively our hardware work as expected, acquiring good quality signals. For this, a patient simulator and a waveform generator of the laboratory have been used, and both acquisition devices receive correctly the signals for each one of the channels.

5.1 Power consumption test

In order to calculate the consumption of our devices, a multimeter will be used for measuring the current of the power supply. Recall that the devices are connected via USB to a power bank that provides a voltage of 5 V, so the power can be known with the value of the current.

$$P = V \times I \tag{5.1}$$

For each device, two cases will be considered: the consumption of only the power supply without data transmission, and the consumption with data transmission, in order to see how much this last case affects the total consumption.

ECG device

Mode	Current	Power	Without LED
No measurements nor data transmission, only powered up	88 mA	**0.44 W**	77 mA
With data transmission at 500 SPS	103 mA	**0.515 W**	92 mA

Tab. 5.1.: Power consumption of the ECG device

The first important thing to remark is that the data transmission increments the total consumption as expected, specifically around 15 mA (75 mW). Moreover, in the measure it has been seen that there is a blinking LED in the microcontroller board that made the current change its value constantly, which apported a consumption of 11 mA (55 mW) when it was turned on.

Generic device

Mode	Current	Power	Without LED
No measurements nor data transmission, only powered up	102 mA	**0.51 W**	91 mA
With data transmission at 500 SPS	110 mA	**0.55 W**	99 mA

Tab. 5.2.: Power consumption of the ECG device

For this device a similar approach is measured. The data transmission consumption in this case is of 8 mA (40 mW), but the total power consumption is a bit higher than the other case. The blinking LED affects as well in the measures, the same way as in the ECG device.

In conclusion, the total power consumption for the devices are from 0.44 W to 0.55 W. In the BIOPAC system, as specified in the datasheet [5], the power supply requirements are 12 V at 1 A current; then, the power consumption is of 12 W. Therefore, the commercial system consumes much more than our devices, so we can conclude that they result in low-power consumption systems.

Just to make some numbers, the power bank used to supply our systems provides a capacity of 20000 mAh. Thus, if we divide by the maximum current measured (transmision data in the generic device, 110 mA), we obtain that the battery will last for a total of 181 hours, whereas the BIOPAC system would last only 20 hours. It is important to remark that these numbers are calculated considering that continuous measures are performed, so the duration of the battery can be even larger if there are no measures.

5.2 Material costs

As it has been mentioned in the introduction, one of the objectives of this project was to develop a low-cost system that could compete with the expensive commercial devices in the market. The BIOPAC system owned by the research group was purchased for around 7000€; this price cannot be affordable for people with a small budget, this is why the interest in producing a low-cost device.

Both acquiring systems are divided into two hardware parts as explained. The first one is the microcontroller board, which has a market price of 16.50€. The second part is the PCB board designed, which has an estimated production cost of around 300€ (approximately 200€ the board itself and 100€ the components). This price includes the copper board and printing, the electronic components used, and the manufacturing costs by the technician. In addition, compared to the dimensions of the BIOPAC system, our devices are much smaller making it easier to take from one place to another.

Nevertheless, it is important to remark that this price is based on a production of a single device manufactured at the university; if we think of a massive production and manufactured in a cheap industry, the price is greatly reduced to ridiculous values, and for the use that it will have it is highly amortized.

In conclusion, a very low-cost competent portable system has been produced, with most of the BIOPAC system functionalities and even with better specifications in some of the limitations presented in the commercial product.

5.3 ECG measurements

We will start by performing ECG measurements, as they are the most interesting signals to study and process information. The idea is to get the 12 ECG leads from the ECG system, and acquire at the same time a subset of leads with the Generic device, so that we could then perform the 12 lead reconstruction. The measurements proceed as follows.

5.3.1 12 ECG leads acquisition

First of all, both acquisition systems are connected to the power bank in order to supply the power needed to work. We also connect the systems via USB to the laptops that have the LabVIEW programs so that the data can be received and saved in an excel file. For the ECG acquisition device, it is necessary to locate the 10 electrodes as it has been explained in the ECG section (Tab. 2.1). These 10 electrodes will be connected directly to the ECG system with the particular connector. For the Generic acquisition device we will use a series of front-ends provided by the research group, which are prepared to amplify the electrical signal. One of these front-ends will be the not-standard lead, where the electrodes can be placed anywhere in the body; we will see the cases for the neck and the waist. Another three front-ends will be used to measure the 3 first standard leads: Lead I, II and III; these 3 leads will be measured with a system developed by the research group that use two metallic balls for the hands and two metallic plates for the feet, that will work as electrodes but being them more comfortable to use.

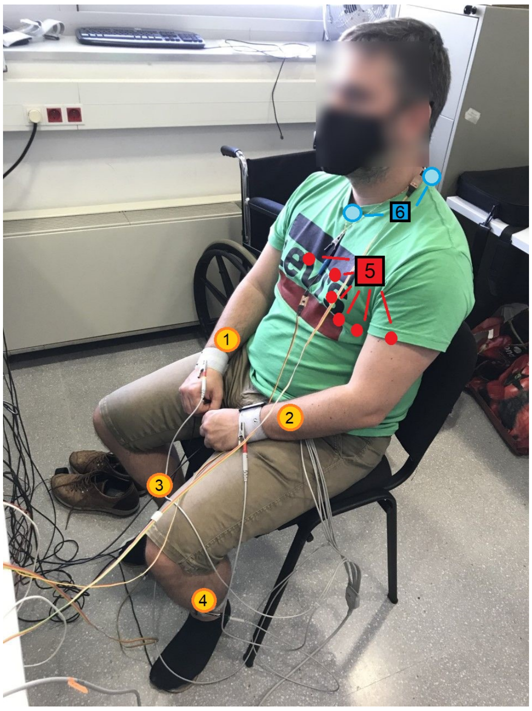


Fig. 5.1.: Measurements on a patient. Electrode position: 1. RA; 2. LA; 3. RL; 4. LL; 5. Chest electrodes; 6. Not-standard position.

Once the measurements have been performed and the data has been saved, we can now go to the processing program in which we will represent the signals and extract some parameter information. We load the excel files in the program for the 12 ECG leads and the subset, and applying the filtering processes we end up with the results seen in annex A.3.1.

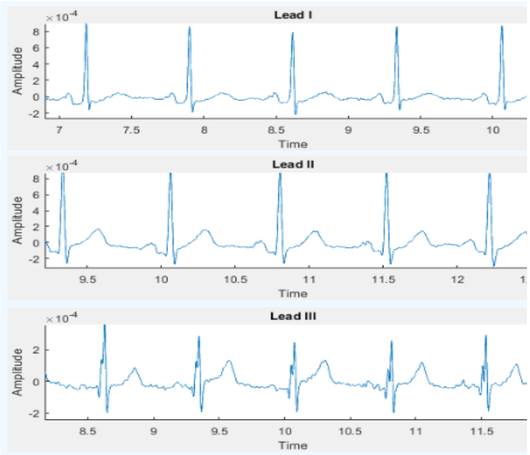
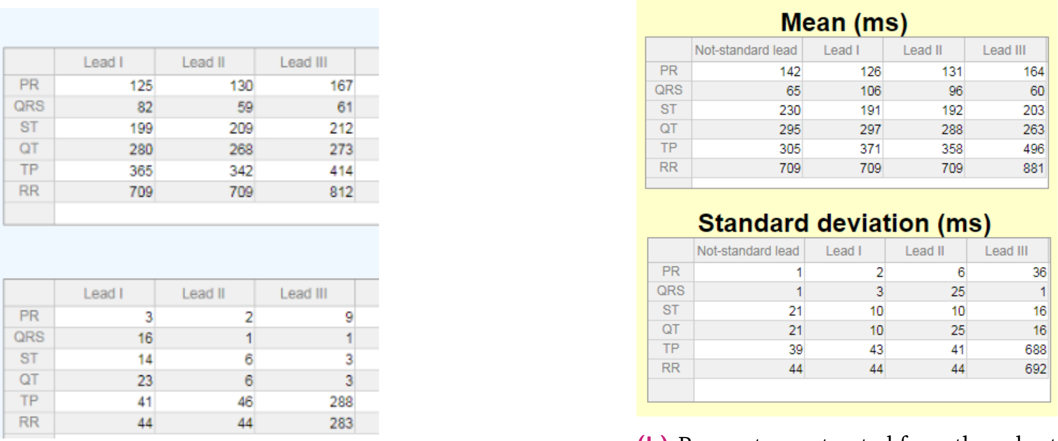


Fig. 5.2.: Some of the 12 ECG leads acquired with the ECG device.

As it can be seen, good enough quality signals are obtained. It is true that for some of them a bit of noise or interferences are observed, but this can be due to possible mechanical imperfections such as bad contacts with the electrodes or movement of the subjects. It is also important to remark that the signals can differ from different patients, as their conditions are not the same; the age, physical condition, or heart position can infer in these results, as for some subjects the signals are better and clearer than for others.

The parameters of these signals have been obtained as well, following the criteria seen in Fig. 4.6. The parameters are represented in tables, where we obtain the mean and standard deviation of all the intervals.



	Lead I	Lead II	Lead III
PR	125	130	167
QRS	82	59	61
ST	199	209	212
QT	280	268	273
TP	365	342	414
RR	709	709	812

	Lead I	Lead II	Lead III
PR	3	2	9
QRS	16	1	1
ST	14	6	3
QT	23	6	3
TP	41	46	288
RR	44	44	283

(a) Some of the parameters extracted from ECG leads.

Mean (ms)

	Not-standard lead	Lead I	Lead II	Lead III
PR	142	126	131	164
QRS	65	106	96	60
ST	230	191	192	203
QT	295	297	288	263
TP	305	371	358	496
RR	709	709	709	881

Standard deviation (ms)

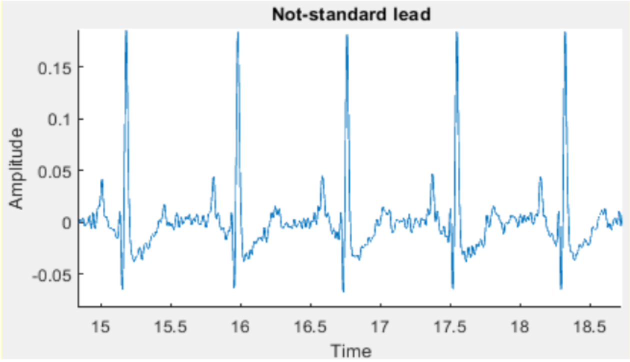
	Not-standard lead	Lead I	Lead II	Lead III
PR	1	2	6	36
QRS	1	3	25	1
ST	21	10	10	16
QT	21	10	25	16
TP	39	43	41	688
RR	44	44	44	692

(b) Parameters extracted from the subset of leads.

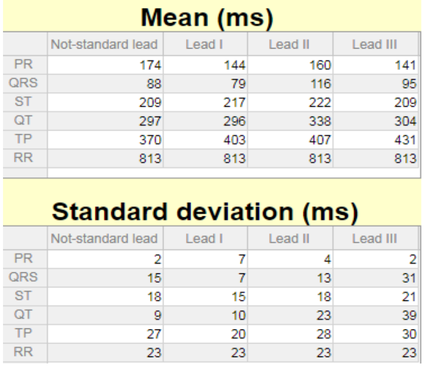
Fig. 5.3.: Parameters of the ECG measurements.

It is observed that all the parameters in A.3.1 make sense between them, as for a same interval they are all of the same order of magnitude, and we even see small deviations of the order of the resolution error of our devices (around 2 ms). There are some punctual signals that have a worse performance, but as mentioned before, not all the signals can be perfect for a particular subject.

In order to see the differences between not-standard locations of the electrodes, we performed two different measurements for a patient: one with the not-standard electrode in the nape, where the Wilson's Central Terminal (WCT) is located, and the other one in the waist. The goal of these measurements is to find a comfortable electrode position for the patient, as the idea is to then reconstruct the 12 ECG leads from the subset without the need of using all the 10 uncomfortable electrodes. The results are the following.



(a) Not-standard plot.



Mean (ms)

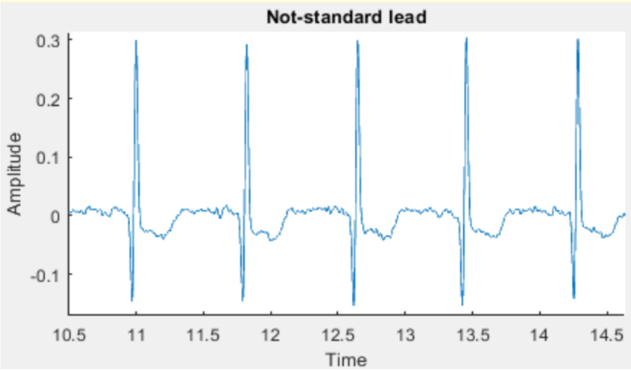
	Not-standard lead	Lead I	Lead II	Lead III
PR	174	144	160	141
QRS	88	79	116	95
ST	209	217	222	209
QT	297	296	338	304
TP	370	403	407	431
RR	813	813	813	813

Standard deviation (ms)

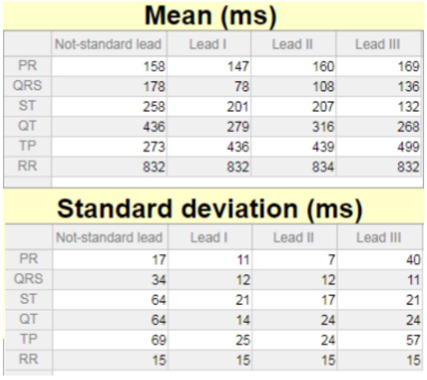
	Not-standard lead	Lead I	Lead II	Lead III
PR	2	7	4	2
QRS	15	7	13	31
ST	18	15	18	21
QT	9	10	23	39
TP	27	20	28	30
RR	23	23	23	23

(b) Mean and standard deviation.

Fig. 5.4.: Results of the not-standard electrode placed at the nape.



(a) Not-standard plot.



Mean (ms)

	Not-standard lead	Lead I	Lead II	Lead III
PR	158	147	160	169
QRS	178	78	108	136
ST	258	201	207	132
QT	436	279	316	268
TP	273	436	439	499
RR	832	832	834	832

Standard deviation (ms)

	Not-standard lead	Lead I	Lead II	Lead III
PR	17	11	7	40
QRS	34	12	12	11
ST	64	21	17	21
QT	64	14	24	24
TP	69	25	24	57
RR	15	15	15	15

(b) Mean and standard deviation.

Fig. 5.5.: Results of the not-standard electrode placed at the waist.

Some differences are spotted; the signal of the waist measurement seems to be clearer than the nape signal, as this last one presents some noise. In contrast, the P and T waves are better seen in the nape signal, this is why we observe greater deviations in the waist signal parameters. Nevertheless, both measurements give similar parameters, with deviations that are typical for these signals, so we can conclude that the placement of the not-standard electrode is insignificant for the nape and the waist, allowing to decide wherever is more comfortable.

5.3.2 Reconstruction of the 12 ECG leads

The last thing left to do is the reconstruction of the 12 ECG leads from the subset of leads acquired. As explained in the transform matrices section, three different methods will be used in order to see which one gives the best performance for our data. The three are carried out at the same time in the code, so a total of 5 matrices are obtained: the U and V matrices which are equal for the three methods, and T1, T2 and T3, corresponding to the matrices of methods 1, 2 and 3, respectively.

Using the application program, first the training phase is done in order to see which method is the best, by looking at the correlations between the 12 ECG original signal and the 12 reconstructed signals, for each one of the methods. To see rapidly the best performance, the program provides the mean of these correlations, and we get the following results for different measurements on subjects.

Subjects	Not-standard placement	Method 1	Method 2	Method 3
Patient 1	nape	75.65%	66.01%	91.88%
Patient 2	nape	58.92%	62.13%	92.26%
Patient 2	waist	86.18%	90.45%	92.26%
Patient 3	nape	27.94%	30.41%	90.05%

Tab. 5.3.: Performance of the different methods for the measurements.

As we can see, for the three measurements where the not-standard electrode was placed at the nape, method 1 and 2 do not give good results, but method 3 does. However, for the measurement where the not-standard electrode was placed at the waist, all three methods give good enough results, being method 3 better. All in all, we can conclude that method 3 gives the best performance in all cases, but for measurements where the not-standard electrode is placed at the waist the three methods give good results as well.

After performing the training phase, the matrices can be saved for every patient, as they will be specific for each one. This will allow that for future measurements it will not be necessary to do the annoying 10 electrodes placement for the 12 ECG leads, as these matrices can reconstruct the 12 leads with a subset of leads measured with a comfortable placement of electrodes. We can try this with the building phase, as we can load the matrices of the subject, as well as the subset of leads measured, and reconstruct the 12 ECG leads (see annex A.3.3). This was done with Patient 1, who did a first test measurement obtaining the 12 ECG leads and the subset, and the following week performed a second measurement. With the new subset measured and the matrices obtained in the first measurement, we were able to reconstruct the 12 ECG leads and we saw that the differences in the parameters were minimum.

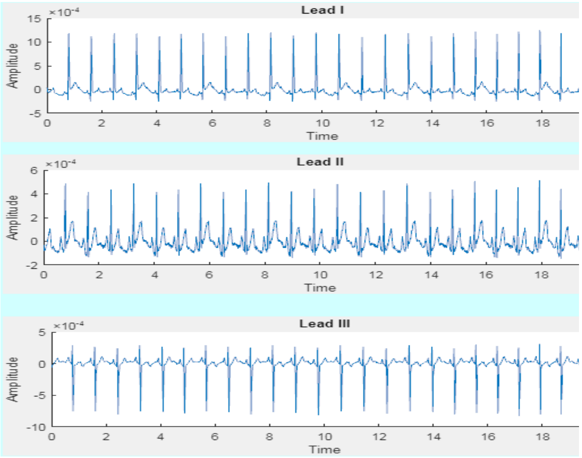


Fig. 5.6.: Some of the 12 reconstructed ECG leads.

Finally, we have performed the reconstruction for both not-standard placements (nape and waist), and we have seen small variations between the parameters obtained, being them practically the same. This leads to the possibility of being able to choose where to place the not-standard electrode, which could be in the most convenient position for the patient.

5.4 Pressure signal measurements

In order to show that the generic device is capable of acquiring any kind of signal, some pressure signal measurements will be carried out with the devices provided by the research group. In the theoretical framework section the PPG has been explained, but we will perform IPG and tonometer measurements. Nevertheless, the theory and algorithms explained for the PPG signals are valid as well for IPG and tonometer, as they are the same type of signal.

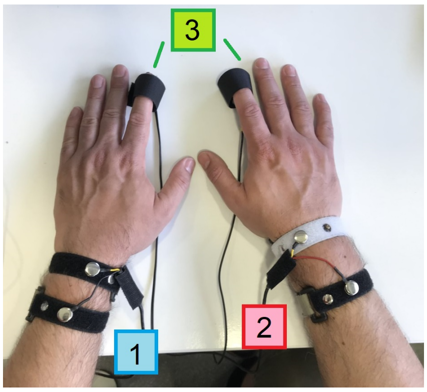


Fig. 5.7.: Measurements on a patient. Signals acquired: 1. ECG; 2. IPG; 3. Tonometer 1 and 2.

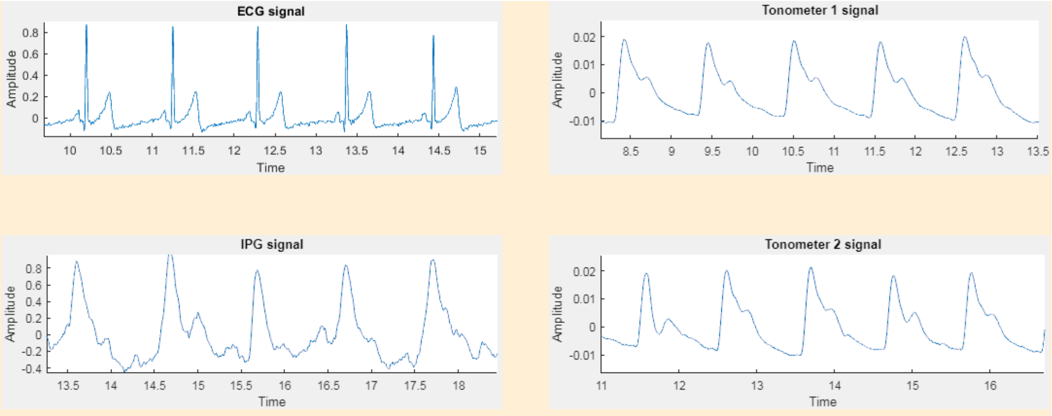
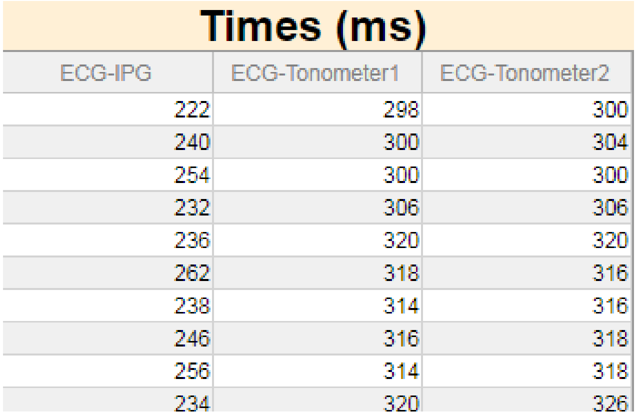


Fig. 5.8.: Pressure signals and ECG.

The signals acquired (Fig. 5.8) and the parameters obtained (Fig. 5.9) can be completely seen in annex A.3.2. The device provided by the research group does the measurement of the ECG, IPG (which corresponds to the aortic blood pulse) and two tonometer (which correspond to the finger blood pulse) signals at the same time, so the application program is prepared for this particular device. This way, as it has been exposed in section 2.3, we can perform the acquisition of different important time intervals between the ECG and pressure signals, in particular between the peaks of the ECG and IPG, and between the peaks of the ECG and the tonometer signals.



Times (ms)

ECG-IPG	ECG-Tonometer1	ECG-Tonometer2
222	298	300
240	300	304
254	300	300
232	306	306
236	320	320
262	318	316
238	314	316
246	316	318
256	314	318
234	320	326

Fig. 5.9.: Table with the time intervals of study.

In figure 5.9 are represented the time intervals for each peak of the signals, with the following means: 244 ms for the ECG-IPG interval, 316 ms and 318 ms for the ECG-Tonometers intervals. As we can see, the time from which the blood exits the heart (ECG) and the instant of maximum aortic blood pressure (IPG) is greater than the interval of time from the aortic zone to the finger (tonometer). This means that the blood goes faster trough the arm, as it should be in a healthy person.

6 Conclusions

The design and implementation of a functional low-cost and low-power portable system has been carried out successfully. The system finally consists of two electronic devices, one capable of acquiring the complete set of 12 ECG leads, and the other with the possibility of getting any existing biosignal. In addition, apart from monitoring signals, the software implemented computes the important parameters for each measurement, so that a medical analysis can be done for the patient.

Different measurements and results have been performed during the project, where we have seen that for the two different placements of the not-standard electrode (nape and waist) there are not many differences with the parameters obtained, as they are in the range of typical values for these intervals. This means that the not-standard electrode could be placed wherever the patient feels more comfortable, something that we were seeking in this project. Moreover, the reconstruction matrices worked as expected and we are able to synthesize the 12 ECG leads from a subset and the patient-specific matrices, with correlations between the original and reconstructed leads of more than 90%. In fact, these matrices need to be obtained just once, as they reach the same results for subsets measured in different times. Thus, it is not necessary to place all the electrodes to perform future measurements, allowing more comfortable and time-saving measurements.

Finally, we were able to obtain the parameters related to pressure signals, and then compute the different important time intervals between ECG and pulse wave signals. These parameters give information about the cardiovascular system and circulation of blood, so this is the last step to reach a complete medical diagnosis. Therefore, with our devices and software programs we have been able to perform a complete analysis of the cardiovascular system.

It is important to remark the efficiency in the low-power consumption test with respect to the BIOPAC commercial system, with an autonomy of around 180 hours for continuous measurements and a power bank of 20000 mA. In addition, the low costs of the devices means a greater accessibility to this kind of system, because the devices designed can be affordable for anyone. Therefore, it is not crazy to think of having this system at homes, where any subject could monitor their health status constantly, and in the case something is wrong they could send the data to a doctor in order to carry out a medical diagnosis. This would be a great advantage for clinical environments, which would be less congested of people and presenciality would not be necessary for minor cases or measurements.

In conclusion, all of our main objectives have been reached successfully. The hardware accomplishes the specifications that we wanted in terms of consumption and good quality acquisition. The software captures, saves and processes the data so that the signals are effectively represented and the parameters are automatically extracted in order to give information about the health status of the subject. Finally, the algorithms that were proposed can be carried out in a final application program that comprises all the functionalities of our system; thus, our devices are fully competitive in the market.

6.1 Future work and perspectives

Although our system is already functional in all the topics described above, there are a series of aspects that can be improved or extended, in order to have a better product. Firstly, the communication used to transmit the data could not be the fastest one, so there is the possibility to improve the communication system with other functionalities provided by the microcontroller. This is possible because the shield designed for our hardware connects all the pins of the STM32 to the PCB board designed, with the thought of using in the future some of the unused pins that can configure the other communication systems. With this we could gain some data transmission speed, and think of the possibility of representing the signals in real time.

Secondly, another thing that can be done in the future is to compact the LabVIEW and MATLAB programs in just one software program, so that it would be more comfortable to acquire the data and then process the information. In addition, it would be convenient to optimize some of the codes and algorithms so that they could be useful even for bad quality signals where it is difficult to detect the important peaks or points.

Finally, as an aesthetic improvement, we could compact all our hardware systems in one box with the corresponding inputs and outputs visible. Thus, we would have a final physical product that can be presented in any interested company.

Bibliography

[1] Darlene Berger. „A brief history of medical diagnosis and the birth of the clinical laboratory Part 1-Ancient times through the 19 th century“. In: (1999) (cit. on p. 1).

[2] Philips Medical Systems. *PAD 12-Lead ECG Monitoring with EASI ™ Lead System*. 2007 (cit. on p. 2).

[3] Sibasankar Padhy and S Dandapat. „Synthesis of 12-Lead ECG from a Reduced Lead Set using Singular Value Decomposition“. In: *IEEE Power, Communication and Information Technology Conference (PCITC) 322.10* (2015) (cit. on p. 2).

[4] Julie A. Scherer and John M. Nicklas. „Synthesis of the 12 lead electrocardiogram from a 3 lead semi-orthogonal subset using patient-specific linear transformation arrays“. In: 1988 (cit. on p. 2).

[5] Biopac Systems. „BIOPAC Product Sheet“. Pat. 2021 (cit. on pp. 2, 40).

[6] David Cano. „Synthesis of the standard 12 leads from atypical leads“. Universitat Politècnica de Catalunya, June 2018 (cit. on pp. 4, 25, 26, 29, 33).

[7] Wikipedia. *Cardiac Cycle*. 2021 (cit. on p. 6).

[8] Yasar Sattar and Lovely Chhabra. *Medical Reference - Electrocardiogram*. Mar. 2021 (cit. on pp. 7, 11).

[9] Wikipedia. *Electrocardiography*. June 2021 (cit. on pp. 7, 9).

[10] Mohammad Ghamari, Denisse Castaneda, Aibhlin Esparza, Cinna Soltanpur, and Homer Nazeran. „A review on wearable photoplethysmography sensors and their potential future applications in health care“. In: *International Journal of Biosensors Bioelectronics* 4 (4 2018) (cit. on pp. 12, 13).

[11] Mohamed Elgendi, Richard Fletcher, Yongbo Liang, et al. „The use of photoplethysmography for assessing hypertension“. In: *npj Digital Medicine* 2 (1 2019) (cit. on p. 13).

[12] Allen John. „Photoplethysmography and its application in clinical physiological measurement“. In: *Physiological Measurement* 28 (3 2007) (cit. on p. 13).

[13] Joan Gómez Clapers and Ramon Casanella Alonso. „Assessment of trends in the cardiovascular system from time interval measurements using physiological signals obtained at the limbs“. Universitat Politècnica de Catalunya, July 2015 (cit. on p. 14).

[14] Texas Instruments. „ADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements“. Pat. 2010 (cit. on p. 15).

[15]Texas Instruments. „User's Guide ADS1298ECG-FE/ADS1198ECG-FE“. Pat. 2010 (cit. on pp. 15, 17).

[16]STMicroelectronics. „STM32L4R5xx Ultra-low-power Arm® Cortex®-M4 32-bit MCU+FPU, 150DMIPS, up to 2MB Flash, 640KB SRAM, LCD-TFT MIPI DSI, ext. SMPS“. Pat. 2017 (cit. on p. 16).

[17]STMicroelectronics. „Nucleo-144 board User Manual“. Pat. 2015 (cit. on p. 16).

[18]Texas Instruments. *ADS1299-x Low-Noise, 4-, 6-, 8-Channel, 24-Bit, Analog-to-Digital Converter for EEG and Biopotential Measurements*. 2012 (cit. on p. 18).

[19]Texas Instruments. *User's Guide EEG Front-End Performance Demonstration Kit*. 2012 (cit. on pp. 18, 19).

[20]Serj1032. *STM32 ADS1298 codes*. Apr. 2017 (cit. on p. 23).

[21]Christopher Watford. *Understanding ECG Filtering*. Mar. 2014 (cit. on p. 26).

[22]Wikipedia. *Pan–Tompkins algorithm*. 2021 (cit. on p. 28).

[23]Asadur Rahman. *MATLAB based Functions for ECG Signal Preprocessing*. Oct. 2018 (cit. on p. 29).

[24]Sanjeev Nara, Manvinder Kaur, and Kundan Lal Verma. „Novel Notch Detection Algorithm for Detection of Dicrotic Notch in PPG Signals“. In: *International Journal of Computer Applications* 86 (17 2014) (cit. on p. 31).

[25]Guillermo Cid Espinosa. *Programación de interfaz gráfica en App Designer para el control vectorial de motores de imanes permanentes*. Universidad Politécnica de Madrid, July 2018 (cit. on p. 37).

List of Figures

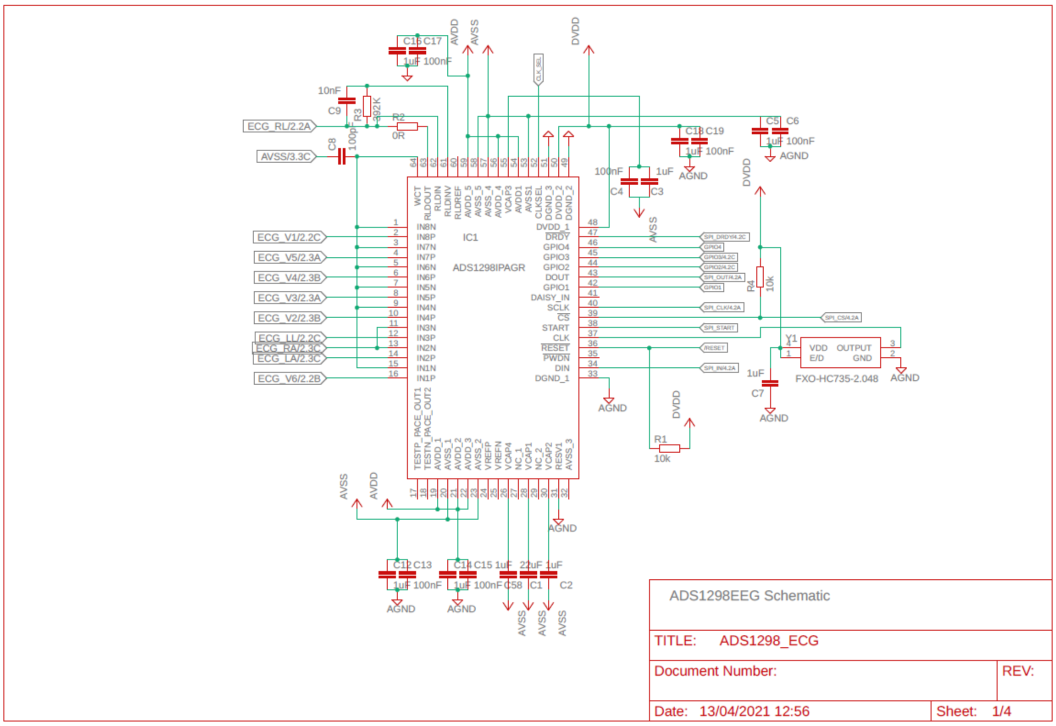
Acronyms

ADC	Analog to Digital Converter.
ECG	Electrocardiogram.
EEG	Electroencephalogram.
EMG	Electromyogram.
HF	High Frequency.
IPG	Impedance Pletysmogram.
LF	Low Frequency.
LMS	Least Mean Square.
PCB	Printed Circuit Board.
PGA	Programmable Gain Amplifier.
PPG	Photopletysmogram.
SPI	Serial Peripheral Interface.
SPS	Samples Per Second.
SVD	Singular Value Decomposition.
UART	Universal Asynchronous Receiver Transmitter.

Additional Figures

A.1 Schematics and PCBs

A.1.1 ECG device



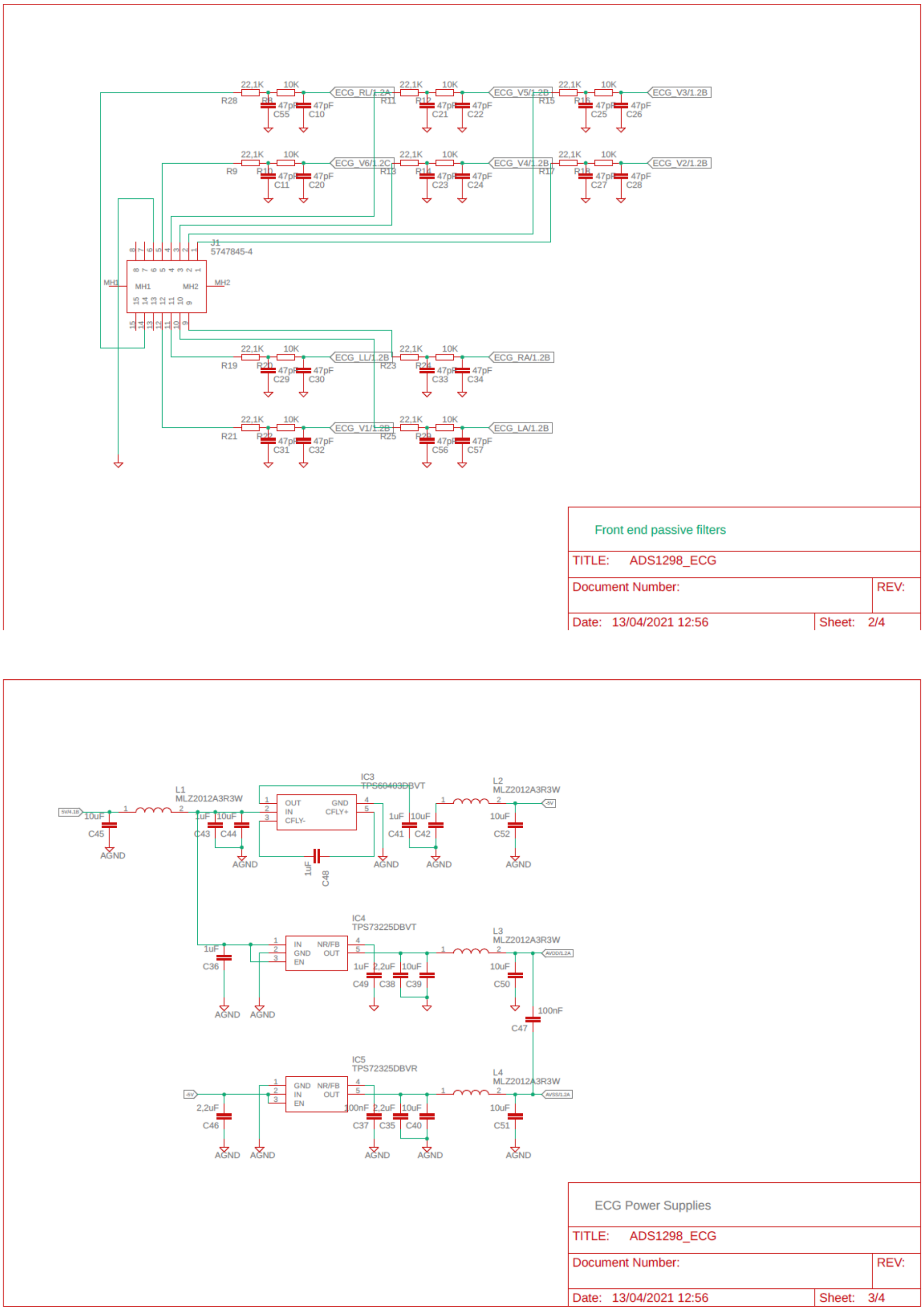
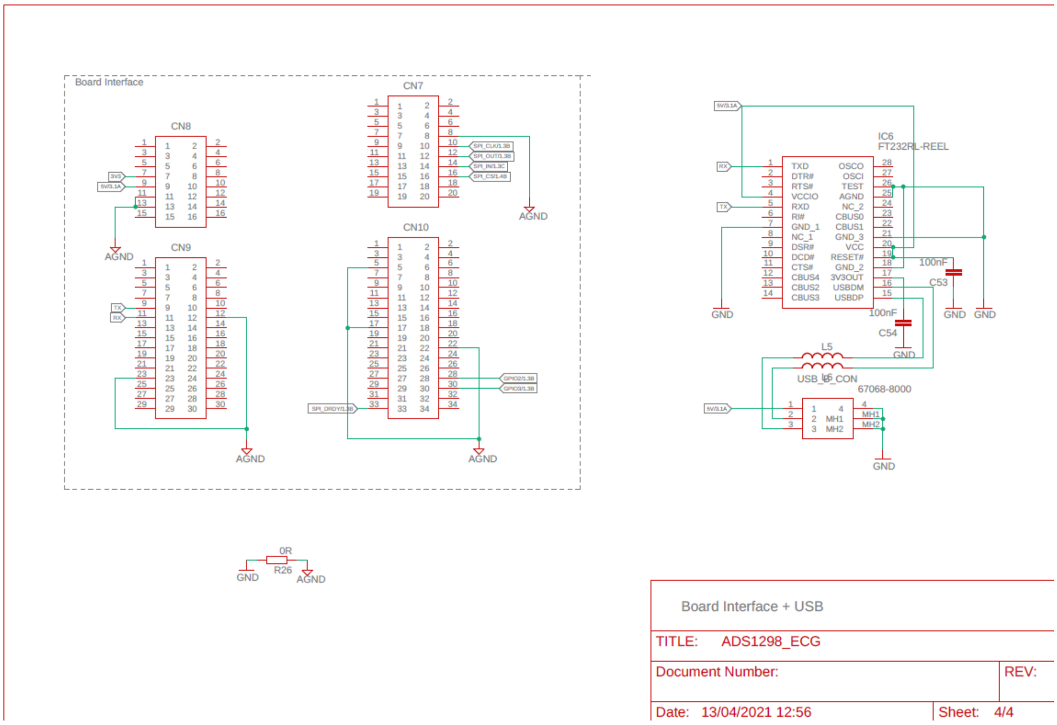


Fig. A.1.: Power supplies.



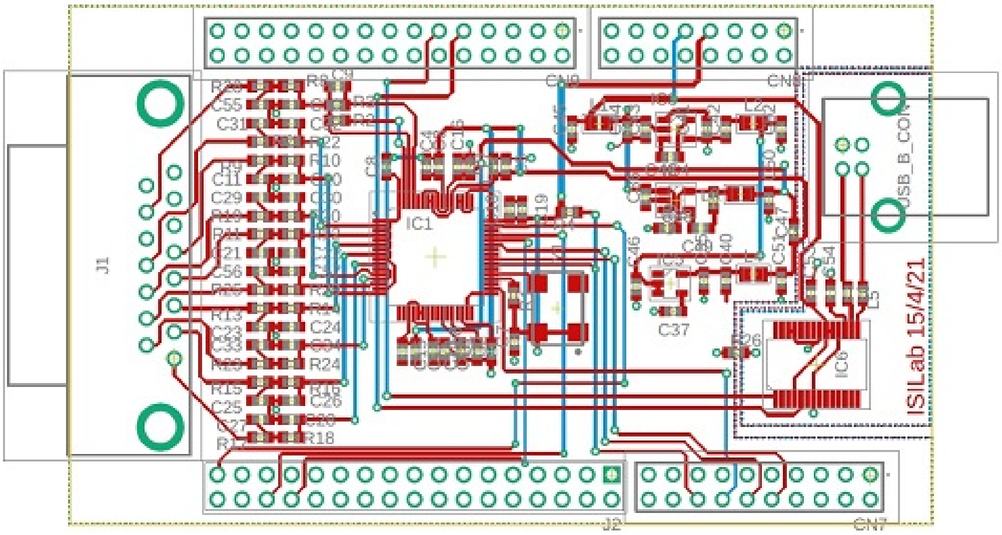
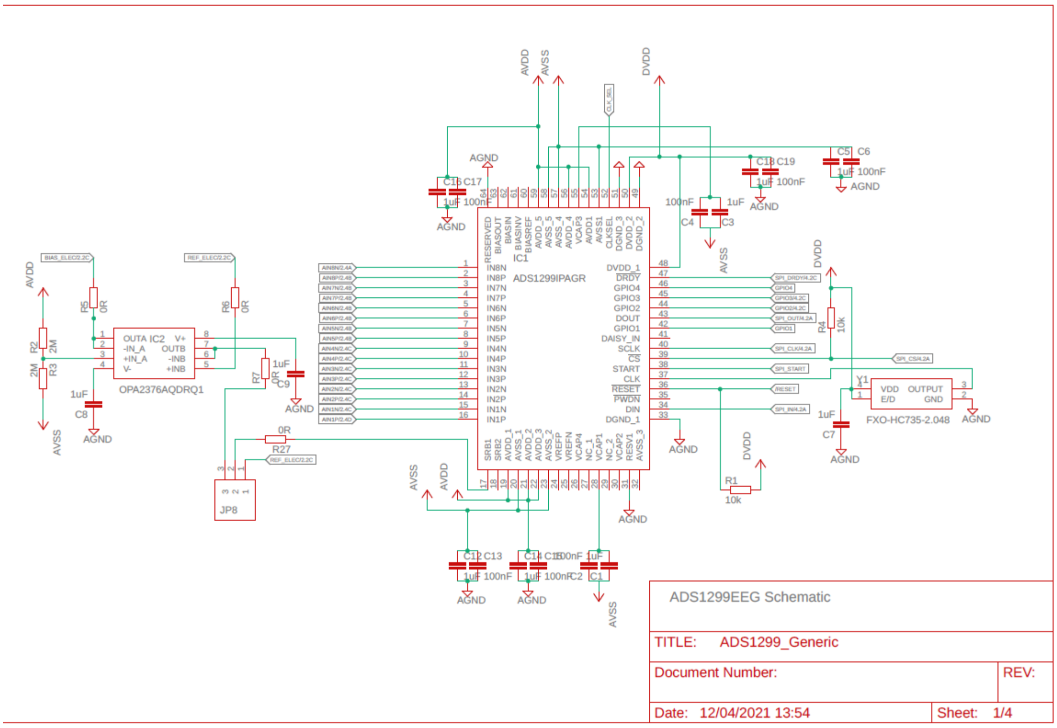
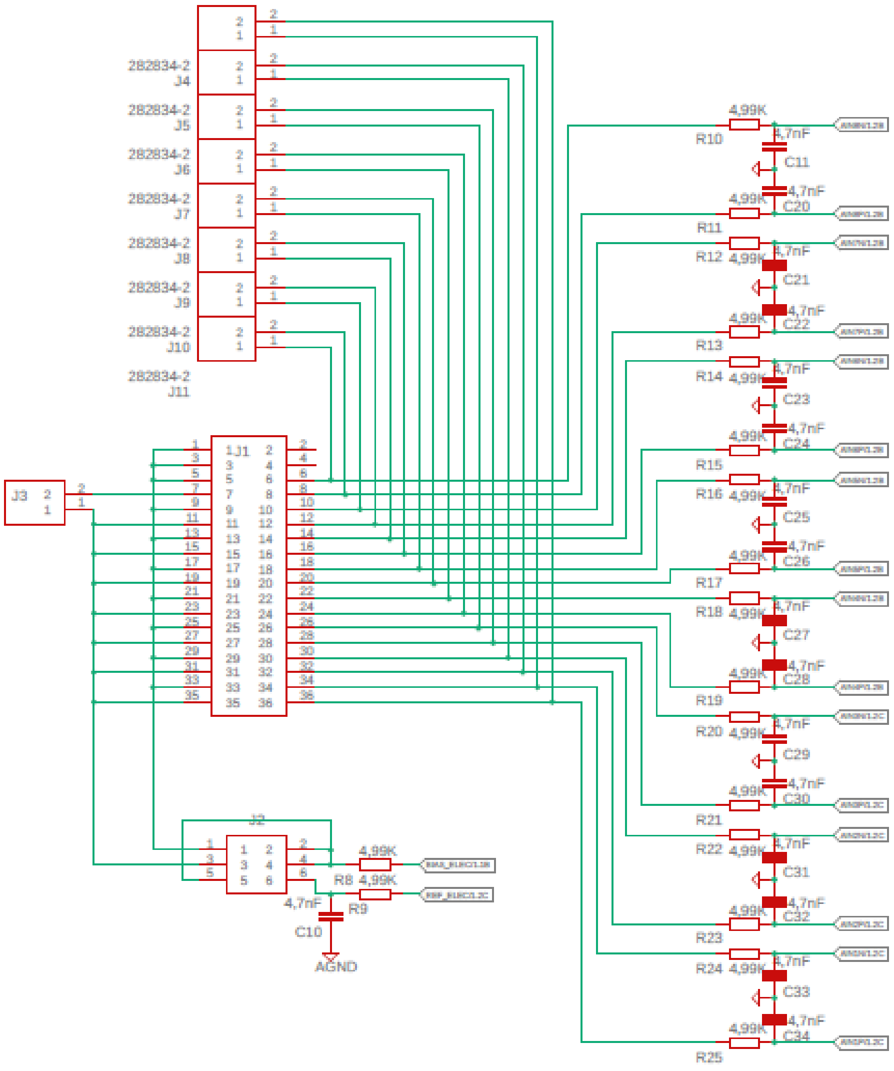


Fig. A.2.: ECG device PCB board.

A.1.2 Generic device





Jumper Schematic

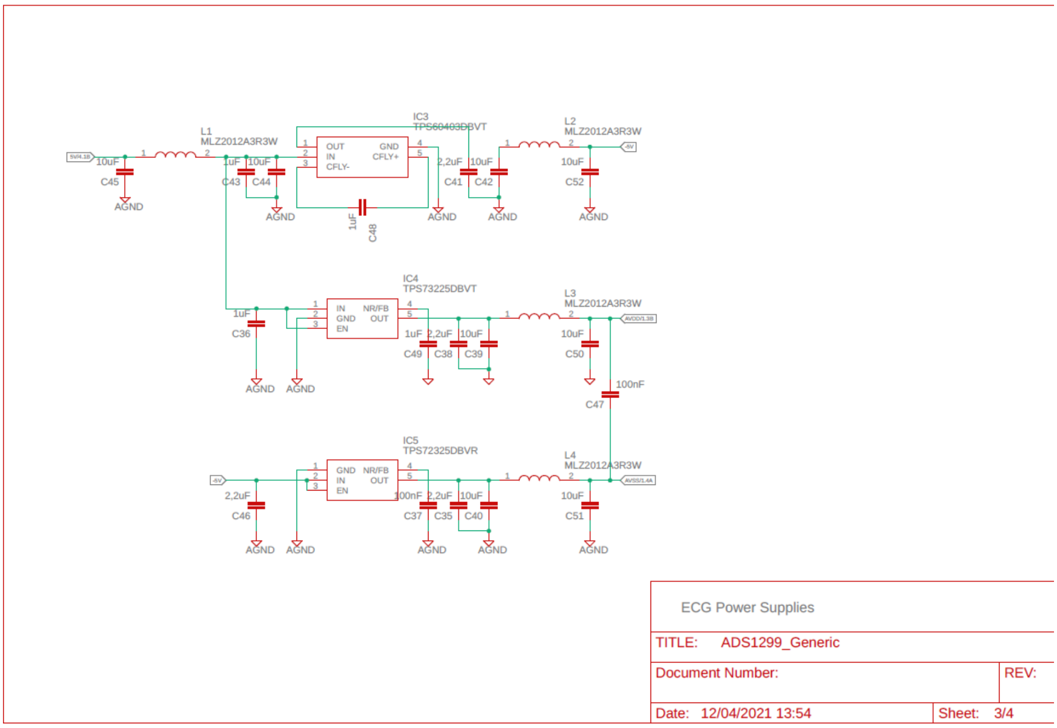
TITLE: ADS1299_Generic

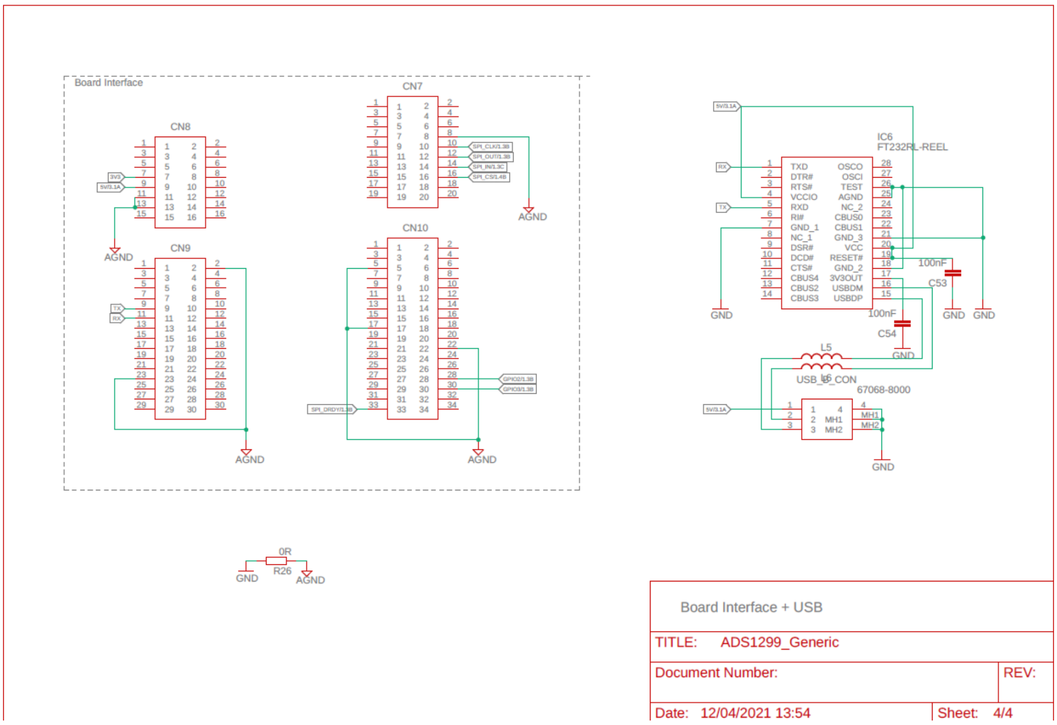
Document Number:

REV:

Date: 12/04/2021 13:54

Sheet: 2/4


IC3
TPS60403DBVT
L1
MLZ2012A3R3W
L2
MLZ2012A3R3W
OUT
IN
CFLY-
GND
CFLY+
10uF
C45
1uF
C43
10uF
C44
2,2uF
C41
10uF
C42
10uF
C52
1uF
C48
AGND
IC4
TPS73225DBVT
IN
GND
EN
NR/FB
OUT
L3
MLZ2012A3R3W
1uF
C36
1uF
C49
2,2uF
C38
10uF
C39
10uF
C50
100nF
C47
IC5
TPS72325DBVR
GND
IN
EN
NR/FB
OUT
L4
MLZ2012A3R3W
2,2uF
C46
100nF
C37
2,2uF
C35
10uF
C40
10uF
C51
ECG Power Supplies
TITLE: ADS1299_Generic
Document Number:
REV:
Date: 12/04/2021 13:54
Sheet: 3/4


Board Interface
CN8
CN9
CN7
CN10
AGND
SPI_CLK/1.3B
SPI_OUT/1.3B
SPI_IN/1.3C
SPI_CS/1.4B
GPIO2/1.3B
GPIO3/1.3B
SPI_DRDY/1.3B
TX
RX
3V3
5V/3.1A
IC6
FT232RL-REEL
TXD
DTR#
RTS#
VCCIO
RXD
RI#
GND_1
NC_1
DSR#
DCD#
CTS#
CBUS4
CBUS2
CBUS3
OSCO
OSCI
TEST
AGND
NC_2
CBUS0
CBUS1
GND_3
VCC
RESET#
GND_2
3V3OUT
USBDM
USBDP
100nF
C53
100nF
C54
GND
L5
USB_CON
67068-8000
MH1
MH2
0R
R26
GND
AGND
Board Interface + USB
TITLE: ADS1299_Generic
Document Number:
REV:
Date: 12/04/2021 13:54
Sheet: 4/4

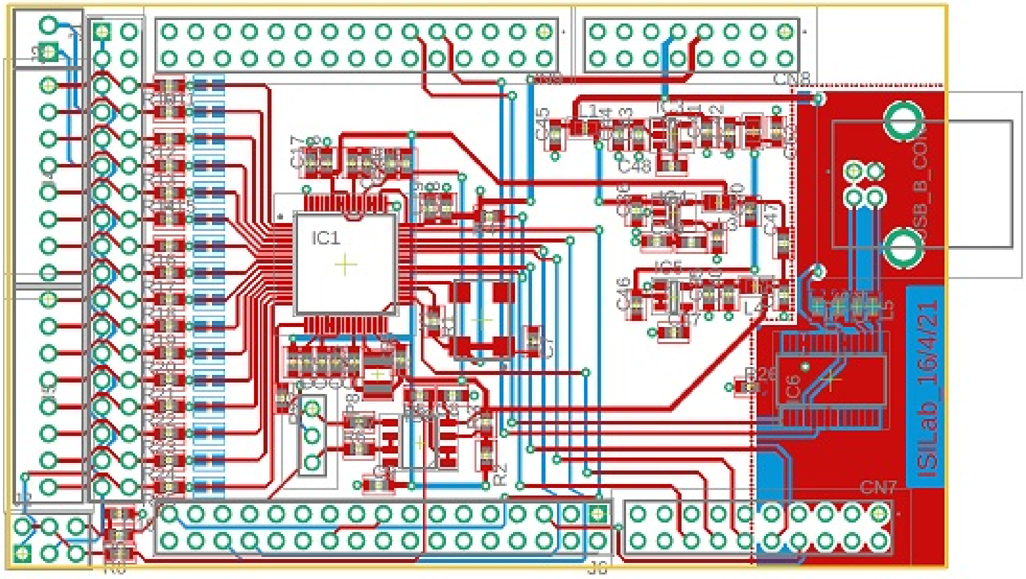
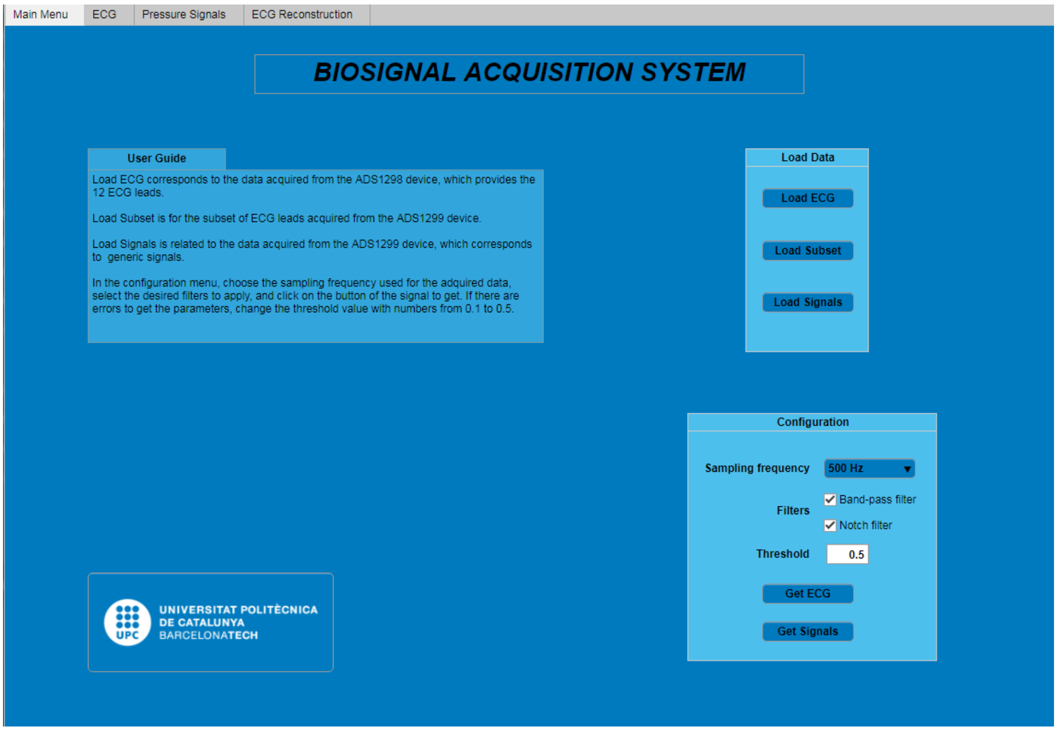


Fig. A.3.: PCB board of the generic system.

A.2 Final program

Menu



BIOSIGNAL ACQUISITION SYSTEM

User Guide

Load ECG corresponds to the data acquired from the ADS1298 device, which provides the 12 ECG leads.

Load Subset is for the subset of ECG leads acquired from the ADS1299 device.

Load Signals is related to the data acquired from the ADS1299 device, which corresponds to generic signals.

In the configuration menu, choose the sampling frequency used for the adquired data, select the desired filters to apply, and click on the button of the signal to get. If there are errors to get the parameters, change the threshold value with numbers from 0.1 to 0.5.

Load Data

Configuration

Sampling frequency: 500 Hz

Filters: ☑ Band-pass filter ☑ Notch filter

Threshold: 0.5

Fig. A.4.: Main menu interface.

ECG menu

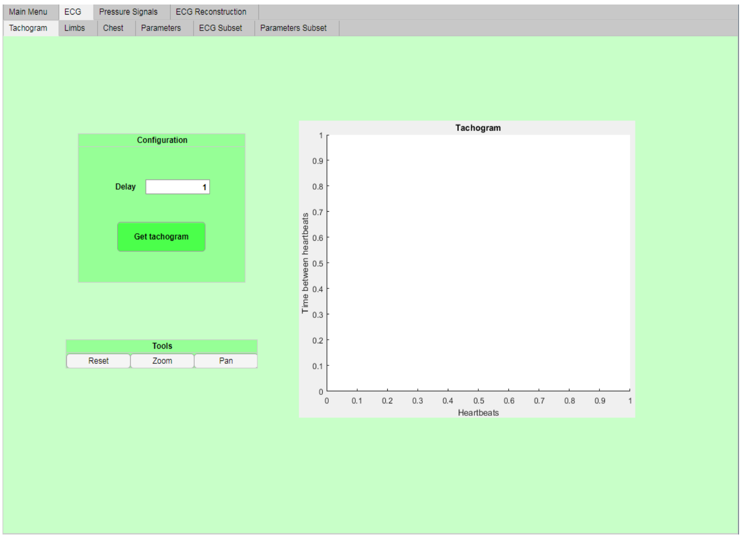
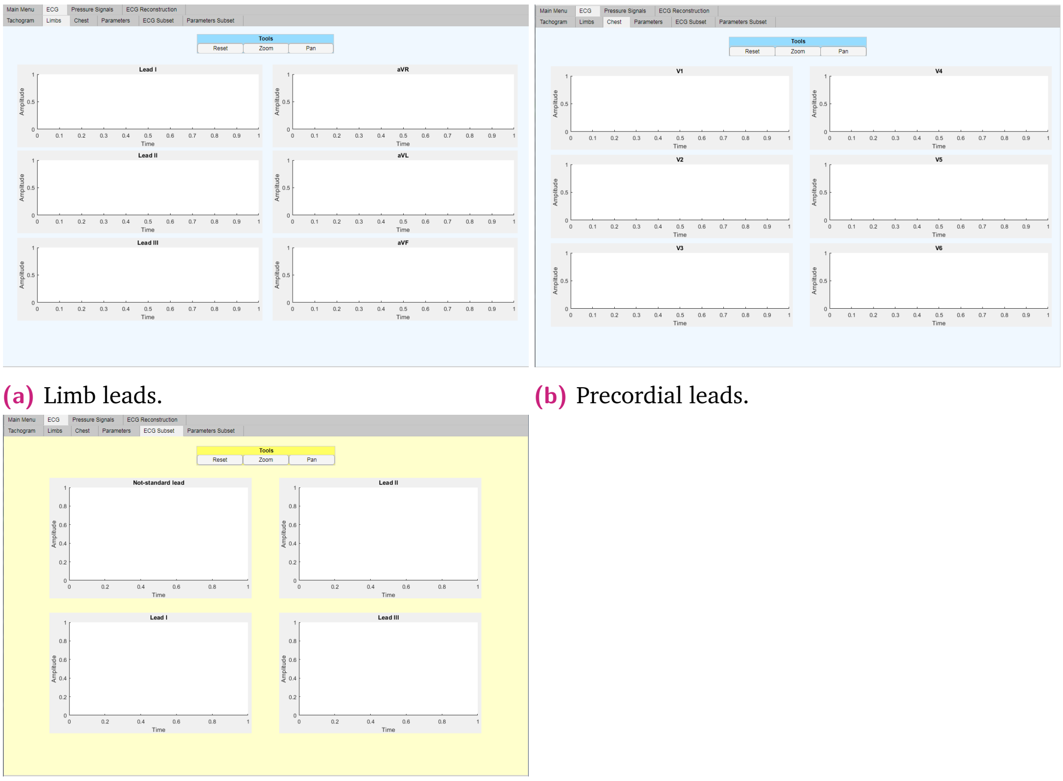


Fig. A.5.: Tachogram tab.

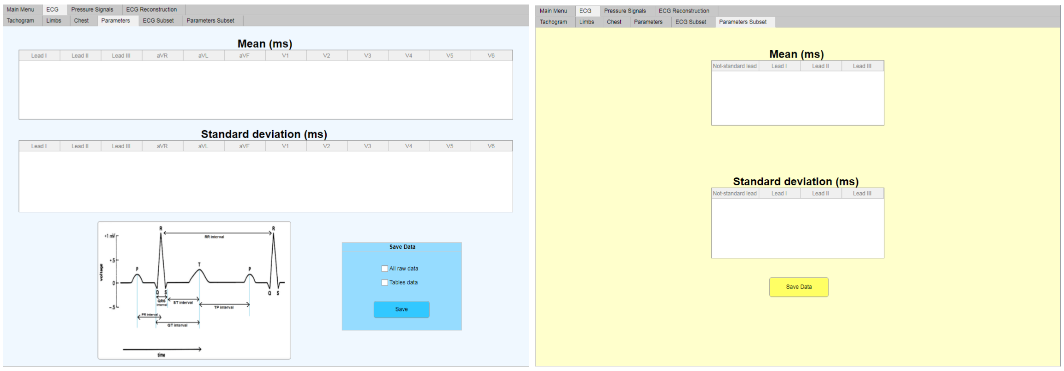


(a) Limb leads.

(b) Precordial leads.

(c) Generic subset of leads.

Fig. A.6.: ECG plots.

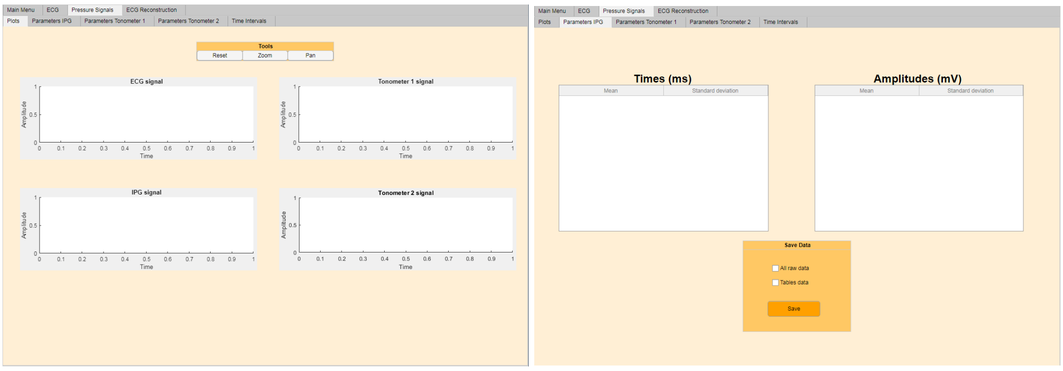


(a) Parameters 12 ECG leads.

(b) Parameters subset.

Fig. A.7.: Parameter tabs.

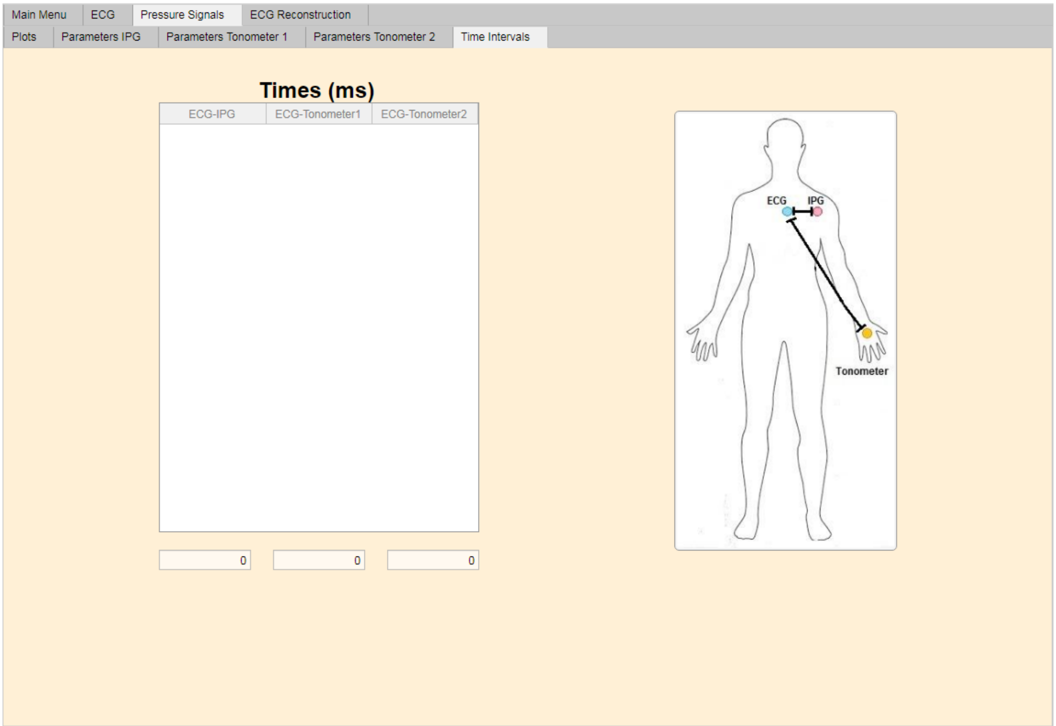
Pressure signals menu



(a) Plots.

(b) Parameters (the same for Tonometer 1 and 2).

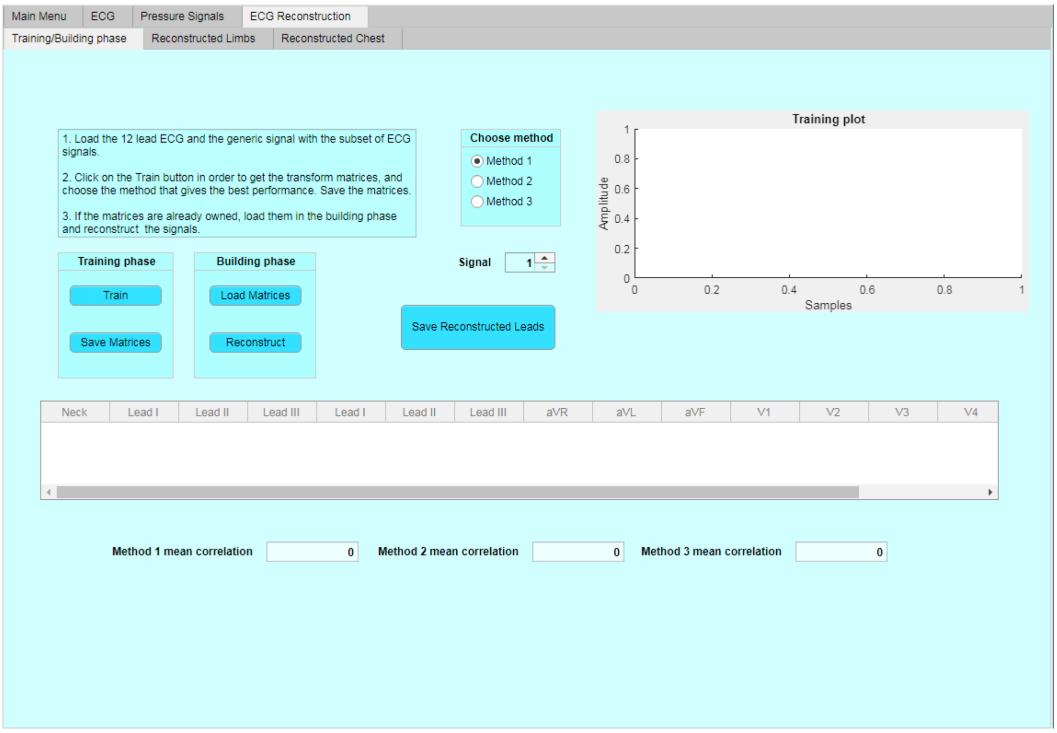
Fig. A.8.: Pressure signal representation and parameters.



ECG-IPG	ECG-Tonometer1	ECG-Tonometer2
0	0	0

Fig. A.9.: Time intervals tab.

ECG Reconstruction menu



1. Load the 12 lead ECG and the generic signal with the subset of ECG signals.

2. Click on the Train button in order to get the transform matrices, and choose the method that gives the best performance. Save the matrices.

3. If the matrices are already owned, load them in the building phase and reconstruct the signals.

Choose method

● Method 1
○ Method 2
○ Method 3

Signal 1

Neck	Lead I	Lead II	Lead III	Lead I	Lead II	Lead III	aVR	aVL	aVF	V1	V2	V3	V4

Method 1 mean correlation 0 Method 2 mean correlation 0 Method 3 mean correlation 0

Fig. A.10.: ECG Reconstruction tab.

A.3 Measurements

A.3.1 ECG measurements

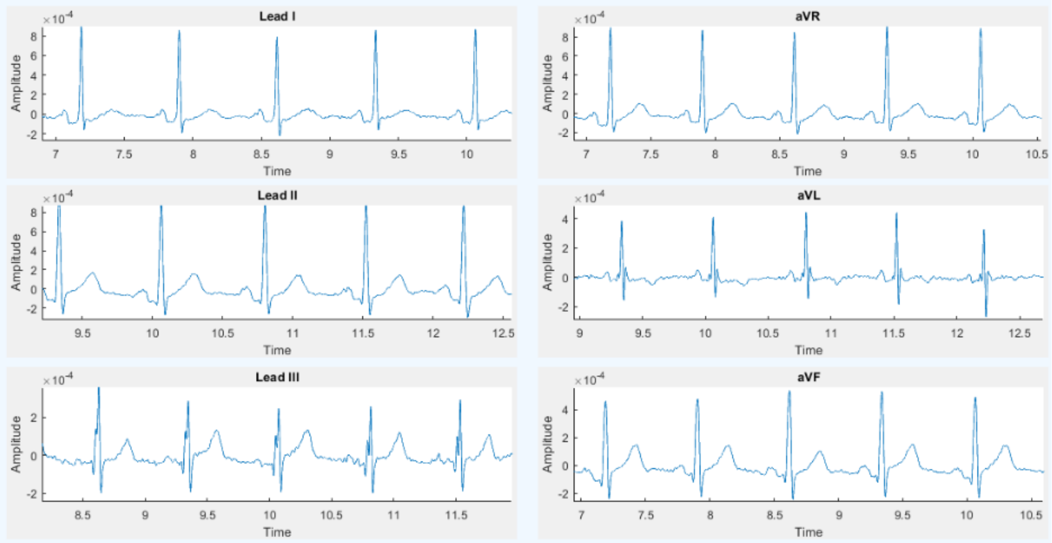


Fig. A.11.: ECG Limb Leads.

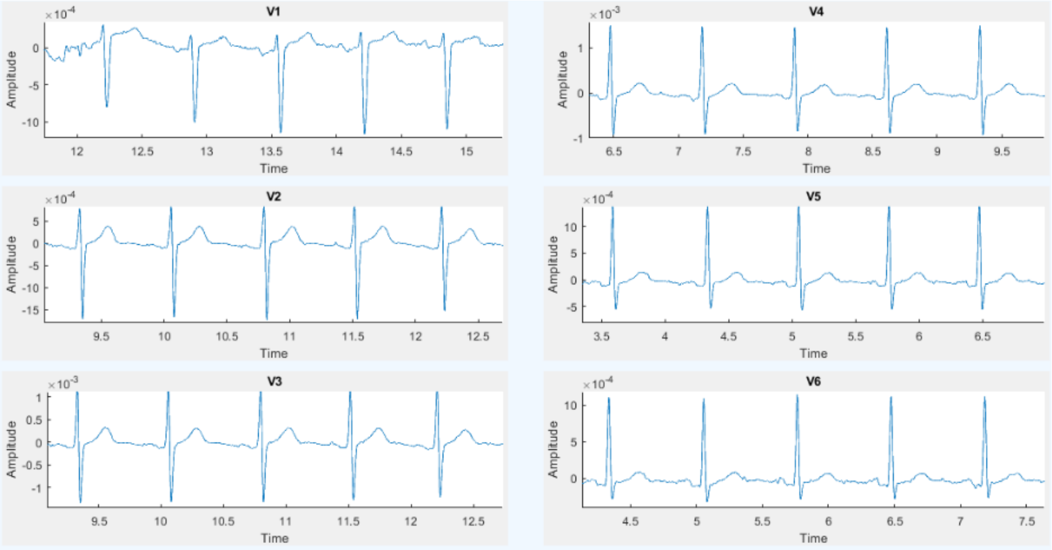


Fig. A.12.: ECG Precordial Leads.

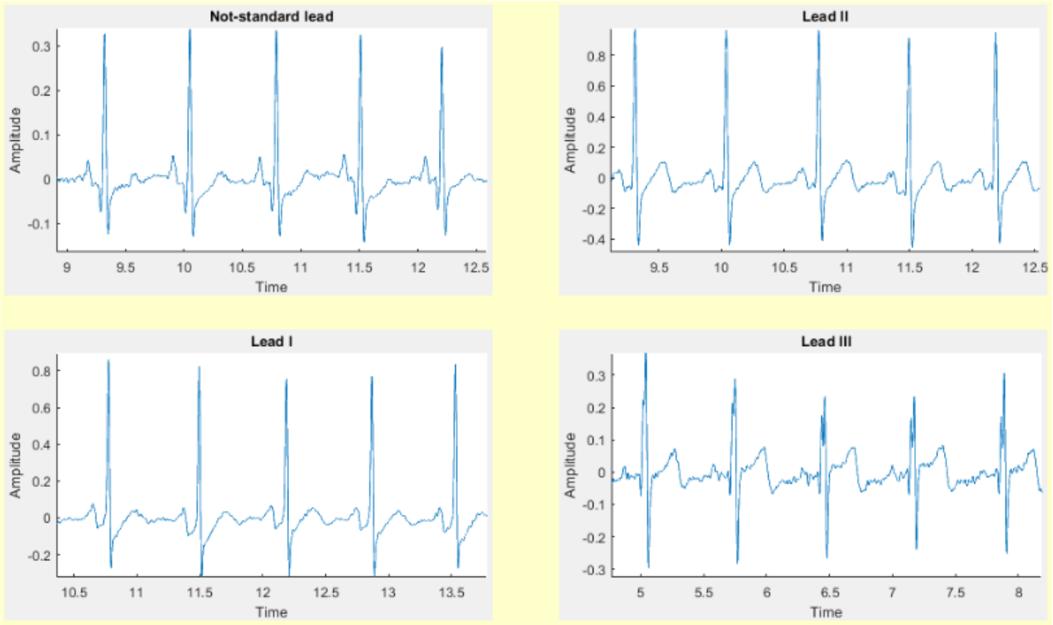
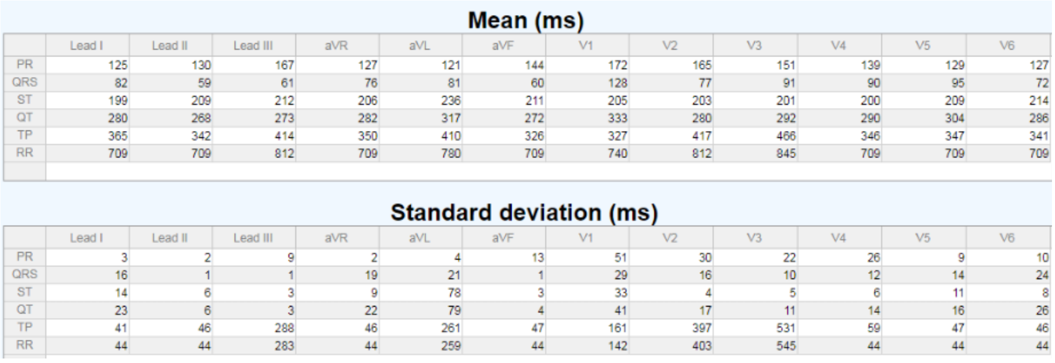


Fig. A.13.: ECG Subset with not-standard lead.



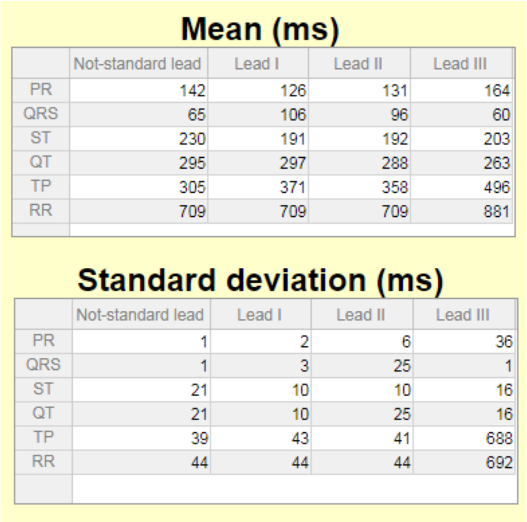
Mean (ms)

	Lead I	Lead II	Lead III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6
PR	125	130	167	127	121	144	172	165	151	139	129	127
QRS	82	59	61	76	81	60	128	77	91	90	95	72
ST	199	209	212	206	236	211	205	203	201	200	209	214
QT	280	268	273	282	317	272	333	280	292	290	304	286
TP	365	342	414	350	410	326	327	417	466	346	347	341
RR	709	709	812	709	780	709	740	812	845	709	709	709

Standard deviation (ms)

	Lead I	Lead II	Lead III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6
PR	3	2	9	2	4	13	51	30	22	26	9	10
QRS	16	1	1	19	21	1	29	16	10	12	14	24
ST	14	6	3	9	78	3	33	4	5	6	11	8
QT	23	6	3	22	79	4	41	17	11	14	16	26
TP	41	46	288	46	261	47	161	397	531	59	47	46
RR	44	44	283	44	259	44	142	403	545	44	44	44

Fig. A.14.: 12 ECG Lead parameters.



Mean (ms)

	Not-standard lead	Lead I	Lead II	Lead III
PR	142	126	131	164
QRS	65	106	96	60
ST	230	191	192	203
QT	295	297	288	263
TP	305	371	358	496
RR	709	709	709	881

Standard deviation (ms)

	Not-standard lead	Lead I	Lead II	Lead III
PR	1	2	6	36
QRS	1	3	25	1
ST	21	10	10	16
QT	21	10	25	16
TP	39	43	41	688
RR	44	44	44	692

Fig. A.15.: ECG Subset paramaters.

A.3.2 Pressure signal measurements

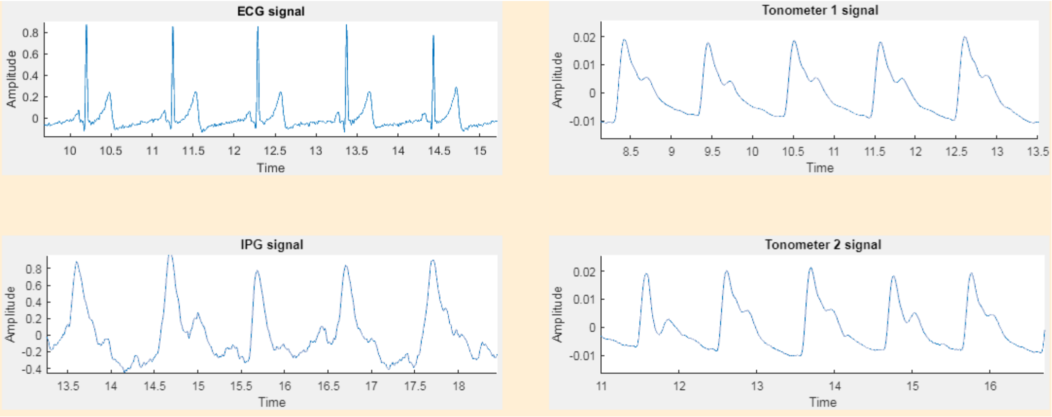
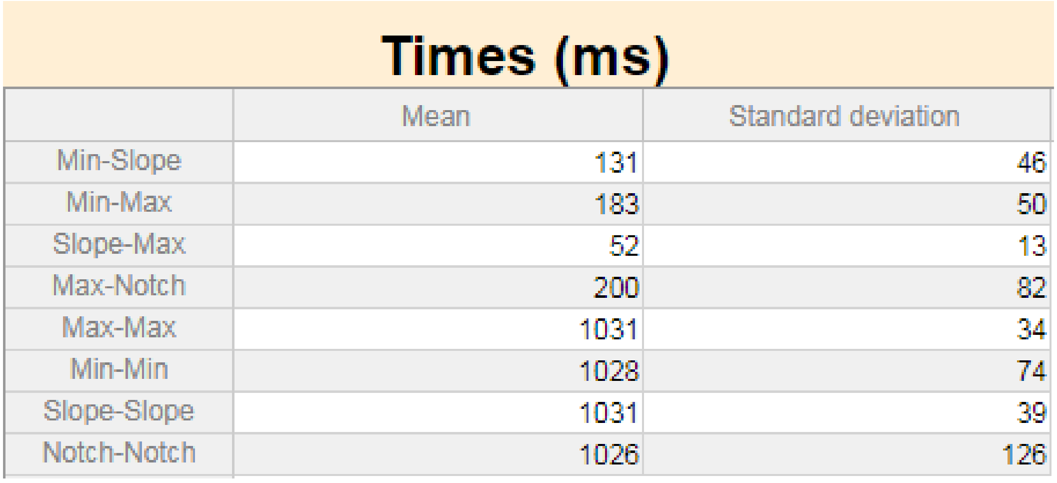


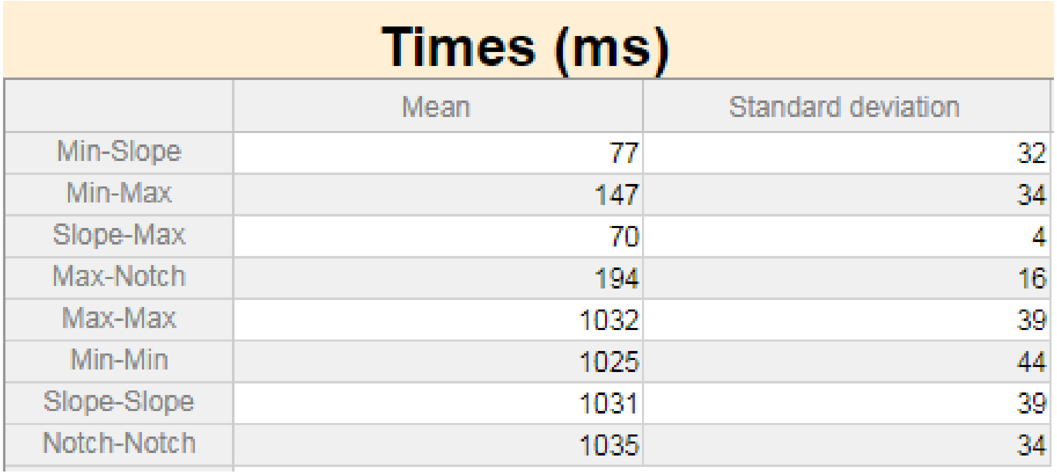
Fig. A.16.: Pressure signals and ECG.



Times (ms)

	Mean	Standard deviation
Min-Slope	131	46
Min-Max	183	50
Slope-Max	52	13
Max-Notch	200	82
Max-Max	1031	34
Min-Min	1028	74
Slope-Slope	1031	39
Notch-Notch	1026	126

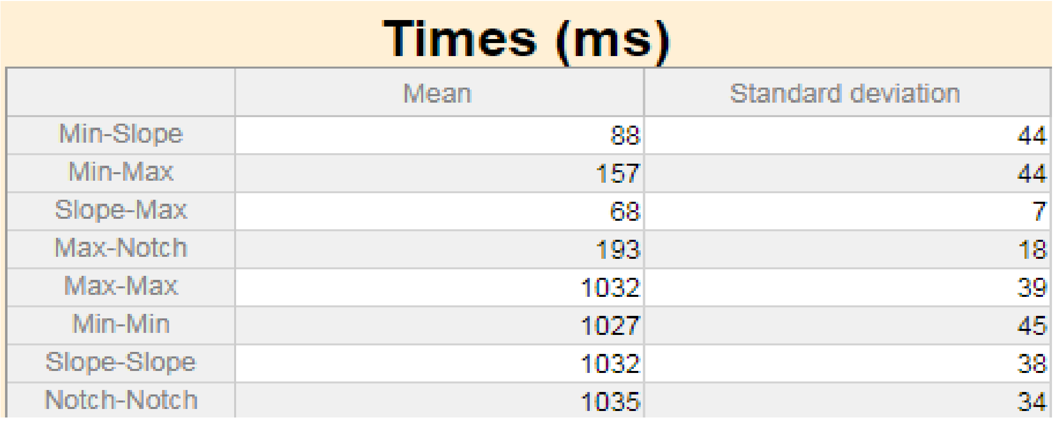
Fig. A.17.: Parameters IPG.



Times (ms)

	Mean	Standard deviation
Min-Slope	77	32
Min-Max	147	34
Slope-Max	70	4
Max-Notch	194	16
Max-Max	1032	39
Min-Min	1025	44
Slope-Slope	1031	39
Notch-Notch	1035	34

Fig. A.18.: Parameters Tonometer 1.



Times (ms)

	Mean	Standard deviation
Min-Slope	88	44
Min-Max	157	44
Slope-Max	68	7
Max-Notch	193	18
Max-Max	1032	39
Min-Min	1027	45
Slope-Slope	1032	38
Notch-Notch	1035	34

Fig. A.19.: Parameters Tonometer 2.

A.3.3 ECG reconstruction

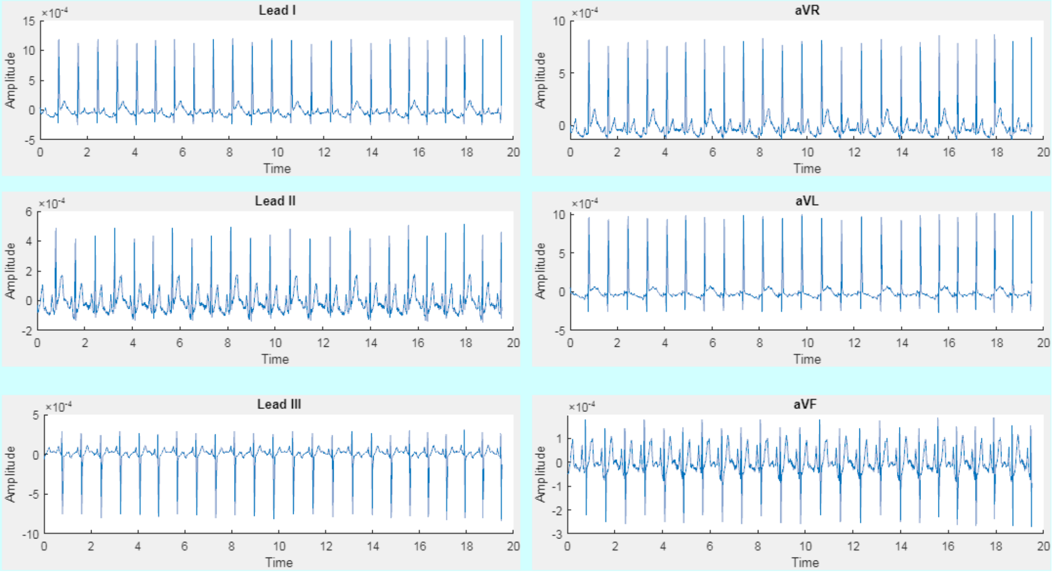


Fig. A.20.: Reconstructed limb leads.

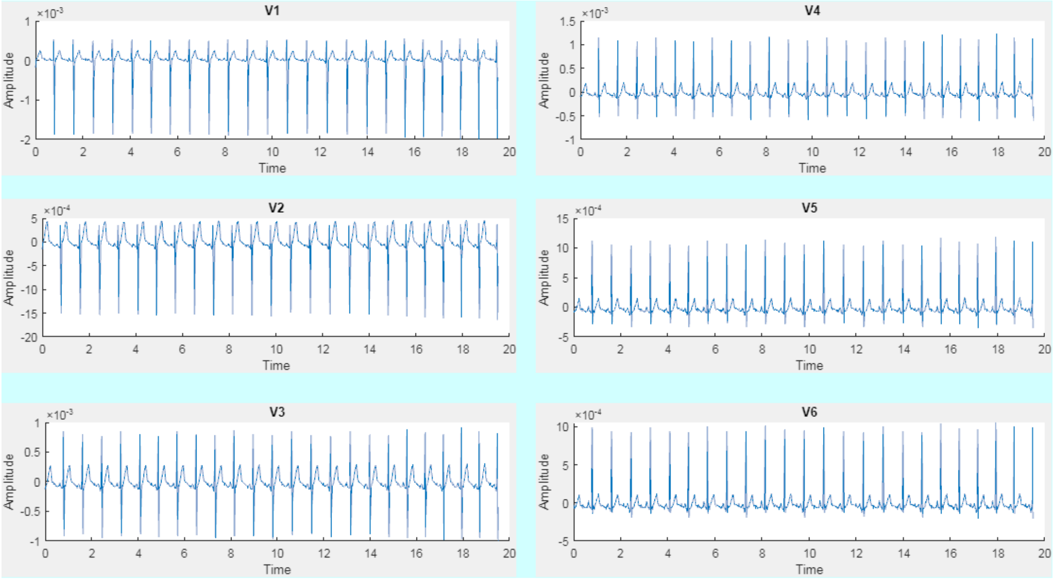


Fig. A.21.: Reconstructed precordial leads.

B Codes

B.1 Microcontroller codes

B.1.1 STM32 main code

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

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

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_SPI1_Init();
  MX_USART3_UART_Init();

  ADS_Init(reset);

  int length = sizeof(reset)/sizeof(reset[0]);
  for(int i = 0; i<length; i++){
          conf[i] = reset[i];
  }
HAL_UART_Receive_DMA(&huart3, conf, 10);

  //HAL_UART_Receive_IT(&huart3, conf, 10);


  /* Infinite loop */
  while (1)
  {
          int equal;
  int emembers = 0;
  int len = sizeof(conf)/sizeof(conf[0]);
  for(int i=0; i<len; i++){
          if(conf[i] == confref[i]){
                  emembers++;
          }
  }
  if (emembers == len){
          equal = 1;
  }
  else{
          equal = 0;
  }

  if(equal == 1){
  }
  else{
          ADS_Init(conf);
          for(int i = 0; i<length; i++){
                  confref[i] = conf[i];
          }
  }

  HAL_Delay(50);

if(conf[9] == 0x00){
        ADS_Init(reset);
}

if(conf[9] == 0x01){
        ADS_Conf(conf);
 while(conf[9] == 0x01){
  if(intDRDY){
                          //startAcq = true;
                          intDRDY = false;
                          //ADS_updateChannelData();
                          ADS_RDATA();
                          ADS_sendUARTData();
                          ADS_START();
                          //HAL_Delay(1000);
                  }
```

```
                    }
            }
     }
}
```

B.1.2 ADS initialization

```
void ADS_Init(uint8_t conf[]){

        ADS_RESET();
        HAL_Delay(500);

        ADS_SDATAC();
        HAL_Delay(500);

        ADS_STOP();
        HAL_Delay(500);

        ADS_getDeviceID();
        HAL_Delay(1000);

        //Work settings
        ADS_WREG(CONFIG1,conf[0]);
        HAL_Delay(100);
        ADS_WREG(CONFIG2,0x10);
        HAL_Delay(100);
        ADS_WREG(CONFIG3,0xDC);
        HAL_Delay(100);

        ADS_WREG(LOFF,0x03);
        HAL_Delay(10);

        ADS_WREG(CH1SET,conf[1]);
        HAL_Delay(10);
        ADS_WREG(CH2SET,conf[2]);
        HAL_Delay(10);
        ADS_WREG(CH3SET,conf[3]);
        HAL_Delay(10);
        ADS_WREG(CH4SET,conf[4]);
        HAL_Delay(10);
        ADS_WREG(CH5SET,conf[5]);
```

```
HAL_Delay(10);
ADS_WREG(CH6SET,conf[6]);
HAL_Delay(10);
ADS_WREG(CH7SET,conf[7]);
HAL_Delay(10);
ADS_WREG(CH8SET,conf[8]);
HAL_Delay(10);

ADS_WREG(BIAS_SENSP,0x00);
HAL_Delay(10);
ADS_WREG(BIAS_SENSN,0x00);
HAL_Delay(10);

ADS_WREG(LOFF_SENSP,0xFF);
HAL_Delay(10);
ADS_WREG(LOFF_SENSN,0x02);
HAL_Delay(10);

ADS_WREG(LOFF_FLIP,0x00);
HAL_Delay(10);


ADS_WREG(LOFF_STATP,0xFE);
HAL_Delay(10);
ADS_WREG(LOFF_STATN,0x06);
HAL_Delay(10);

ADS_WREG(GPIO,0x00);
HAL_Delay(10);

ADS_WREG(MISC1,0x00);
HAL_Delay(10);
ADS_WREG(MISC2,0xF0);
HAL_Delay(10);

ADS_WREG(CONFIG4,0x22);
HAL_Delay(10);

ADS_WREG(0x18,0x0A);
HAL_Delay(10);

ADS_WREG(0x19,0xE3);
```

```
        HAL_Delay(10);


        ADS_RREGS(0,17);
        HAL_Delay(1000);

        ADS_START();
        HAL_Delay(100);
}
```

B.2 Matlab codes

B.2.1 ECG Pan Tompkins algorithm

```
function [signal,tachogram,Indexes,Frr]=PT_ECG(InECG,Fs)
x=detrend(InECG);
M=length(x); intrange=round(5*Fs/20);
der=zeros(1,M); Pulse=zeros(1,M);
timevec=linspace(0,M/Fs,M);
hb=5;       %Number of heartbeats to average
ldelay=.13; %aprox Delay of the negtive slope with respect to the r wave
udelay=.05; %aprox Delay of the positive slope with respect to the r wave
confi=.3;   %Maximum variation from one hearbeat with respect to the others
Frr=8;      %Sampling frequency of the tachogram

Wn = 45/Fs;
NN = 3; % order of 3 less processing
[a,b] = butter(NN,Wn,'low'); %bandpass filtering
y = filtfilt(a,b,x);
y = y/ max(abs(y));
Wn = .5/Fs;
NN = 3; % order of 3 less processing
[a,b] = butter(NN,Wn,'high'); %bandpass filtering
fECG = filtfilt(a,b,y);
fECG = fECG/ max(abs(fECG));

for i=5:M
der(i)=(2*fECG(i)+fECG(i-1)-fECG(i-3)-2*fECG(i-4))/8;%derivative operator
end
```

```
derr=detrend(der).*detrend(der);%squaring operator

for n=intrange:M
Pulse(n)=sum(derr(n-intrange+1:n))/intrange;%Integrator
end

%setting the treshold for detecting the rising and falling slopes
treshold=mean(ecg);
QRSStart=[]; QRSEnd=[];

for i = 1:length(ecg)-1
    %detect if the signal crosses the slope and save its index
    if Pulse(i) <= treshold && Pulse(i+1) > treshold
        QRSStart = [QRSStart i];
    end
    if Pulse(i) <= treshold && Pulse(i+1) > treshold
        QRSEnd = [QRSEnd i];
    end
end

Indexes=zeros(1,length(QRSStart));
for i=1:length(QRSStart)
    %search for the maximum point arround the tresholde crossing points
   [~,ind]=max(fECG(QRSStart(i)-round(ldelay*Fs):QRSStart(i)+round(udelay*Fs)));
    Indexes(i)= ind+QRSStart(i)-round(ldelay*Fs)-1;
end

%define the interval vector
rr=[Indexes(2)-Indexes(1)]; truetime=[rr(1)];
i=2;
Indexes=Indexes';
while i<length(Indexes)
    if i>=hb+1
        bb=hb;
    else
        bb=i-1;
    end
    %define the tresholds
    avgbeat=sum(rr(end-bb+1:end))/bb;
    uppbound =(1+confi)*avgbeat;
    lowbound =(1-confi)*avgbeat;
    %check the RR intervals
    [rr,Indexes,i]=tach_check(uppbound,lowbound,rr,Indexes,fECG,Fs,i);
end
%interpolate the tachogram for evenly spaced data
MM=round(Frr*truetime(end)/(2*Fs))*2;
truetime2=linspace(truetime(1),truetime(end),MM);
tacho = interp1(truetime,rr,truetime2, 'pchip');
%store the signal and the tachogram
signal=[timevec',fECG];
tachogram=[truetime2'/Fs,tacho'/Fs];
return
```

B.2.2 Filtering

```
function [fSig2]=Filter_BPF(InSig,Fs,Fh,Fl,order)

M=length(InSig);
tv=linspace(0,M/Fs,M);

x=InSig;
% x=detrend(x);

%%Interference filtering

 Wn = (Fl*2)/Fs;
 NN = order; % order of 3 less processing
 [a,b] = butter(NN,Wn,'low'); %lowpass filtering
 y = filtfilt(a,b,x);
 fSig = y;


 Wn = (Fh*2)/Fs;
 NN = order; % order of 3 less processing
 [a,b] = butter(NN,Wn,'high'); %highpass filtering
 fSig2 = filtfilt(a,b,fSig);

 fSig2=[tv',fSig2];
```

B.2.3 PQRST detection

```
function out = P_QRS_Tdetect(y, Fs, peak_threshold)

Q_th = round((20/360)*Fs);
Q_th_1st = round((60/360)*Fs);
Q_th_2nd = round((5/360)*Fs);
P_th_1st = round((40/360)*Fs);
P_th_2nd = round((1/360)*Fs);
S_th_1st = round((5/360)*Fs);
S_th_2nd = round((60/360)*Fs);
T_th_1st = round((40/360)*Fs);
T_th_2nd = round((120/360)*Fs);
% ------------------QRS detection by FS1 and FS2 Algorithm ---------%
y0 = []; y1 = []; y2 = []; y3 = [];
y0       = diff(y);
for i = 2:length(y)-2
    y1(i) = (y0(i-1) + 2*y0(i) + y0(i+1))/4;
end
y1(1) = 0;
y2 = diff(y0);
for i = 1:length(y)-2
    y3(i) = y1(i) + y2(i);
end

th = peak_threshold*max(y3);
QRS = [];
for i = 2:length(y3)-6
    if (y3(i) > th)  &&  y3(i) > y3(i+1) && y3(i) > y3(i-1)
        QRS = [QRS i];
    end
end
QRS_Actual = [];
for i = 2:length(QRS)-1
    if QRS(i+1) -QRS(i) > (Fs*0.6)
       QRS_Actual = [ QRS_Actual QRS(i) ];
    end
end
% -------------------------------------------------------------%
%  ---------- R peak detection in original signal ------------------%
R_L = [];
```

```
for i = 2:length(QRS_Actual)-1
    a = QRS_Actual(i)-Q_th: QRS_Actual(i)+S_th_2nd;
    m = max(y(a));
    b = find(y(a)== m);
    if numel(b) ~= 0
        b = b(1);
        b = a(b);
        R_L = [R_L b];
    end
end
% -------------------------------------------------------------------%
% ----------- Q peak detection signal -------------------------------%
Q_L = [];
for i = 1:length(R_L)
    a = R_L(i)-P_th_1st: R_L(i)-P_th_2nd;
    m = min(y(a));
    b = find(y(a)== m);
    if numel(b) ~= 0
        b = b(1);
        b = a(b);
        Q_L = [Q_L b];
    end
end
% ------------------------------------------------------------------ %
% -------------- S peak detection -----------------------------------%
S_L = [];
for i = 1:length(R_L)
    x = R_L(i)+S_th_1st:R_L(i)+S_th_2nd;
    m = min(y(x));
    p = find(y(x) == m);
    if numel(p) ~= 0
        p = p(1);
        p = x(p);
        S_L = [S_L p];
    end
end
% -----------------------------------------------------------------%
% ------------- P Peak detection ----------------------------------------%
P_L = [];
for i = 1:length(Q_L) % He cambiado de coger a partir del 2do indice porque sino daban ind
    a = Q_L(i)-Q_th_1st: Q_L(i)-Q_th_2nd;
    if a(1)<0
```

```
        b = 2;
        P_L = [P_L b];
    else
    m = max(y(a));
    b = find(y(a)==m);
    b = b(1);
    b = a(b);
    P_L = [P_L b];
    end
end
% ----------------------------------------------------------------------- %
% ----------------- T peak detection ------------------------------------%
T_L = [];
for i = 1:length(S_L)
    a = S_L(i)+T_th_1st: S_L(i)+T_th_2nd;
    m = max(y(a));
    b = find(y(a)==m);
    b = b(1);
    b = a(b);
    T_L = [T_L b];
end
%-----------------------------------------------------------------------%
% ------------------------------------------------------------------%

T = 1/Fs; L = length(y); t = (0:L-1)*T;

% Suponemos que este codigo pierde una muestra
P_Lt = (P_L-1)*T; Q_Lt = (Q_L-1)*T; R_Lt = (R_L-1)*T; S_Lt = (S_L-1)*T; T_Lt = (T_L

out = [P_Lt; Q_Lt; R_Lt; S_Lt; T_Lt];

% figure
% plot(t,y);
% hold on;
% grid on;
% plot(P_Lt, y(P_L), 'rs', 'MarkerFaceColor', 'b');
% plot(Q_Lt, y(Q_L), 'ro', 'MarkerFaceColor', 'g');
% plot(R_Lt, y(R_L), 'rv', 'MarkerFaceColor', 'r');
% plot(S_Lt, y(S_L), 'ro', 'MarkerFaceColor', 'b');
% plot(T_Lt, y(T_L), 'rs', 'MarkerFaceColor', 'g');
% % plot(S_L, y(S_L), 'rs', 'MarkerFaceColor', 'b');
% title('QRS Complex, P & T waves');
% legend('ECG Signal','P peak','Q peak','R peak','S peak','T peak');
% %axis([1000 29500 -1 1.0]);
% %disp(out)
end
```

B.2.4 Pulse wave signal detection

```
function [times, amps] = pulsedetect(data, Fs)

T = 1/Fs; L = length(data);
t = (1:L)*T;

der = diff(data); der2 = diff(der);

window = round(0.5*Fs);
window2 = round(0.2*Fs);

% Max slope
[pks,locsslope] = findpeaks(der,'MinPeakDistance',450);

tiempo3 = locsslope*T;
amp3 = data(locsslope);

% Max peak detection
locsmax = [];
for i = 2:length(locsslope)-1
    a = locsslope(i):locsslope(i)+window2;
    m = max(data(a));
    b = find(data(a)==m);
    if numel(b) ~= 0
        b = b(1);
        b = a(b);
        locsmax = [locsmax b];
    end
end
tiempo = locsmax*T;
amp = data(locsmax);

% Min peak detection
locsmin = [];
for i = 2:length(locsslope)-1
```

```
    a = locsslope(i)-window2:locsslope(i);
    m = min(data(a));
    b = find(data(a)==m);
    if numel(b) ~= 0
        b = b(1);
        b = a(b);
        locsmin = [locsmin b];
    end
end
tiempo2 = locsmin*T;
amp2 = data(locsmin);

% Punto inflexion (notch)
locsnotch = [];
for i = 2:length(locsslope)-1
    a1 = locsslope(i):locsslope(i)+window;
    a1 = a1(2:round(0.9*length(a1)));
    m = min(der(a1));
    c = find(der(a1)==m);
    c = a1(c);
    a2 = c:locsslope(i)+window;
    a2 = a2(1:round(0.4*length(a2)));
%     a2 = a2(1:round(0.3*length(a2)));
    %[pksa,locsa] = findpeaks(derPPG(a2));
    m2 = max(der(a2));
    %b = locsa(1);
    b = find(der(a2)==m2);
    if numel(b) ~= 0
        b = b(1);
        b = a2(b);
        locsnotch = [locsnotch b];
    end
end
tiempo4 = locsnotch*T;
amp4 = data(locsnotch);

lmin = min([length(tiempo),length(tiempo2),length(tiempo3),length(tiempo4)]);

tiempo1 = tiempo(2:lmin);tiempo2 = tiempo2(2:lmin);tiempo3 = tiempo3(3:lmin+1);tiem
times = [tiempo1' tiempo2' tiempo3 tiempo4'];

amp1 = amp(2:lmin);amp2 = amp2(2:lmin);amp3 = amp3(3:lmin+1);amp4 = amp4(2:lmin);
```

```
amps = [amp1 amp2 amp3 amp4];

locsmax = locsmax(2:lmin);locsmin = locsmin(2:lmin);locsslope = locsslope(3:lmin+1);locsno

figure()
plot(t,data); hold on; grid on;
plot(tiempo1, data(locsmax), 'rv', 'MarkerFaceColor', 'r');
plot(tiempo2, data(locsmin), 'rs', 'MarkerFaceColor', 'b');
plot(tiempo3, data(locsslope), 'ro', 'MarkerFaceColor', 'g');
plot(tiempo4, data(locsnotch), 'ro', 'MarkerFaceColor', 'r');
title('PPG peak detection');
legend('PPG Signal','Max peak','Min peak','Max slope', 'Notch');


end
```