UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

PROGRAMMING AND TESTING OF A SMALL-SIZE ECG MONITOR FOR SMALL ANIMAL GATED IMAGING

Bachelor Thesis

Trabajo Fin de Grado

Department of Biomedical Engineering

Oral presentation date: 07/07/2015

Author: Irene María García López

Supervisor: Juan José Vaquero López

Leganés, June 2015

Author: **Irene María García López**
Supervisor: **Juan José Vaquero López**

Title: **Programming and testing of a small-size ECG monitor for small animal gated imaging**

THE TRIBUNAL

President: Luis Emilio García Castillo

Secretary: María Arrate Muñoz Barrutia

Vocal: Jorge Ripoll Lorenzo

The defense of this Bachelor Thesis took place on July 7th 2015 in *Leganés*, at the EPS (*Escuela Politécnica Superior*, School of Engineering) of the Universidad Carlos III de Madrid (UC3M), earning a final MARK of

ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to all the persons that have helped me along the way of culminating my Bachelor Thesis. Thank you for all your assistance, guidance and encouragement.

I feel especially thankful to my supervisor Juan José Vaquero for being a source of constant inspiration and offering me the opportunity of participating in this project. I am very grateful for his unceasing advice and his crucial role in guiding me.

I would also like to mention my great appreciation to Guillermo Vizcaíno for his patience in instructing me in the field of electronics and being at the same time an endless source of support and energy.

I would also like to highlight the important contributions of Rigoberto Chil whose master of electronic devices were vital for the development of this project.

I am very grateful to Sandra de Francisco and Yolanda Sierra for welcoming me at Hospital GU Gregorio Marañón and offering their untiring assistance always with a smile.

I want to give a special thanks to the people that shared with me this unique experience at the laboratory for their cheerfulness which was essential to keep working hard while having fun.

And last but not least, my most profound thankfulness goes to my family and friends who truly believed in me but did not get tired of being an enthusiastic source of motivation.

ABSTRACT

Pre-clinical cardiac imaging has become one of the most important non-invasive techniques widely used nowadays for studying the electrophysiology and hemodynamic behaviour of the heart using small animals' models.

Despite the great potential it offers for clinical applications in humans, the introduction of movement artefacts and blurring in the images due to the inherent blood flow and respiration among others, reduces significantly the quality of imaging leading to mistakes in results interpretation. A solution to this problem is the synchronization of the electrocardiogram with the imaging machine in a process called gated imaging to trigger image acquisition at specific moments of the cardiac cycle.

In this manner, the principal objective of the presented Bachelor Thesis was to design, develop, program and test a small size ECG monitor for small animal gated imaging. The main functions of this device are monitoring the cardiac activity at the same time a TTL pulse is activated in real-time for gating synchronization with the imaging machine.

A microcontroller, an analog-to-digital converter and a touch screen were connected together to form a small ECG device that is able to record bioelectrical signals of the heart, to process them and display the electrocardiogram on the screen and generate a TTL pulse for gated imaging. Communication in between the different modules was enabled by SPI protocol implementation.

The evaluation of the ECG prototype was performed first in the laboratory with an ECG simulator showing good performance to be later tried in small animals. On the whole the results were satisfactory except from some issues in TTL pulse activation that might need further improvements to enhance gated imaging implementation.

This project and the future improved versions of this ECG device, will have a big scientific and social impact. This small size ECG monitor will not only offer an easy to use portable monitoring device for pre-clinical research but also a powerful tool for gated imaging that will be essential for the development of revolutionary drugs to treat heart diseases.

Key words: heart, ECG, gated imaging, TTL pulse, SPI, pre-clinical research, microcontroller.

Contents

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADC	Analogue to Digital Converter
AP	Action Potential
A-V	Atrioventricular
BPM	Beats per minute
Ca2+	Calcium ion
CMRR	Common Mode Rejection Ratio
CPHA	Clock phase
CPOL	Clock polarity
CT	Computer tomography
ECG	Electrocardiogram
ED	End diastole
EEG	Electroencephalography
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMG	Electromyography
ES	End systole
eV	Electron Volts
K+	Potassium ion
MCU	Microcontroller Unit
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
Na+	Sodium ion
NSR	Normal Sinus Rythm
PET	Positron Emission Tomography
PGA	Programmable Gain Amplifier
PWM	Pulse width modulation
RX	Receiver
S-A	Sinoatrial
SCK	Serial Clock
SCL	Serial clock
SDA	Serial data

SPECT	Single Photon Emission Tomography
SPI	Serial Peripheral Interface
SPS	Samples per seconds
SS	Slave Select
TTL	Transistor-transistor logic
TWI	Two wire interface
TX	Transmitter
UART	Universal Asynchronous Receiver/Transmitter

1 Introduction

1.1 Motivation

In the last century, medical imaging has revolutionized the way medicine was conceived. For the first time, no invasive technique had to be applied to reveal the internal anatomical structures of the human body. Physiological abnormalities in organs and tissues could be also diagnosed in a non-invasive way.

The wide range of medical imaging modalities such as Computed Tomography (CT), Positron Emission Tomography (PET) or Magnetic Resonance Imaging (MRI) among others, have been shown to be extremely useful not only for clinical analysis in humans but also for pre-clinical in small animals [1]. In fact, imaging of rats used as models in cardiac studies has provided new insights in cardiovascular diseases in which concerns the physiology of the heart and its response to new treatments [2].

Despite the potential of cardiac imaging, an inherent problem that adds substantial complexity to this kind of studies remains: movement. Respiration, blood flow and heart motion cause artifacts during acquisitions limiting the quality of the images [3]. In order to counteract this challenging difficulty a synchronization of the cardiac cycle with the imaging machine is desirable [4] [5]. Hence, that is what gated imaging relies on. The different phases of the cardiac cycle either dictate the timing of image acquisitions in a prospective manner, or serve as a reference to reorganize data of the cardiac cycle during image reconstruction in a retrospective manner [6].

Along these lines, the main objective of this Bachelor Thesis is to develop a portable ECG device that not only monitors the cardiac activity of small animals but also provides a solution to motion blurring in imaging studies through synchronization with the imaging machine. This device is aimed to be used at the Hospital General Universitario Gregorio Marañón where there is a need of improving the quality of images acquired with the Argus CT/PET machine (Sedecal, Madrid) in cardiac studies. The necessity of visualising the heart at specific moments of the cycle to thoroughly extract important parameters of its contractile function represents an additional interest in the development of this project.

1.2 State of the art and previous work

There exists a wide variety of monitoring and gating instruments for small animals in the market. All of them share similar features such as monitoring ECG, respiration, oxygen saturation (SpO2), arterial pressure, temperature, TTL synchronism pulse etc. They can be connected to CT and PET scans in order to carry out gated imaging.

At Hospital General Universitario Gregorio Marañón a Vision Pet monitor (RGB, Madrid) is used in combination with the Argus PET/CT machine as shown in Figures 1 and 2. Although this device has shown to work accurately in the synchronization process the technicians and researchers have express their desire of having a smaller, portable unit that could be use in experiments where all the features of the RGB system are not needed.

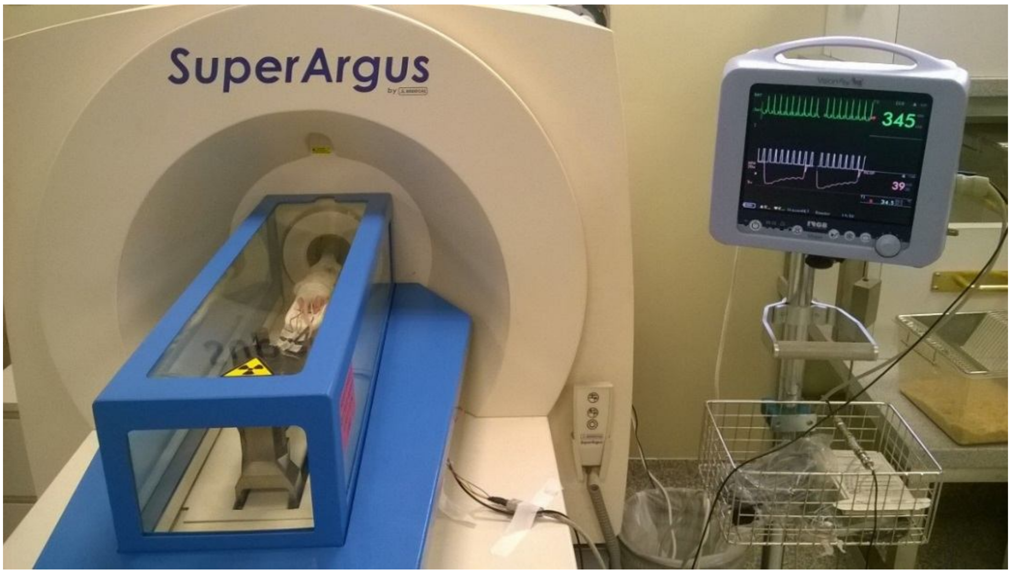


Figure 1. Gated imaging experiment with cardiac activity monitoring at "Laboratorio de Cirugía Experimental" of Hospital General Universitario Gregorio Marañón

Figure 2. Vision Pet monitor from RGB currently used at "Laboratorio de Cirugía Experimental".

The opportunity to develop a customized ECG system to be used with the imaging machine was evaluated on a previous TFG [7]. In that work, a proof of concept was developed, and the selection of components and architecture was defined as shown in Figure 3. The present project takes over the conclusions of this previous work.

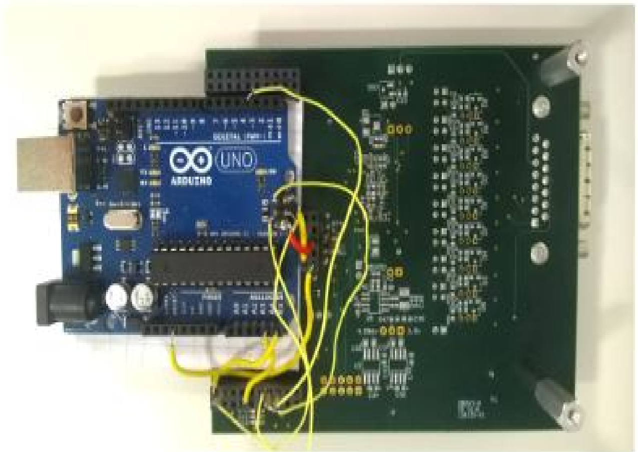
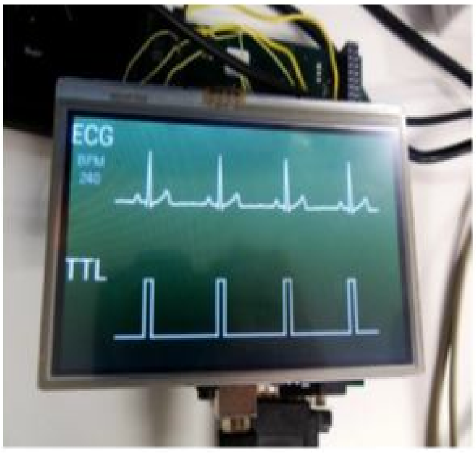



Figure 3. ECG device prototype of previous former thesis

1.3 Objectives

The principal objective of this project is to program and test a small ECG device for small animals in order to perform gated cardiac imaging by synchronization with the Argus CT/PET machine located in the "Laboratorio de Cirugía Experimental" at Hospital General Universitario Gregorio Marañón.

This general objective can be subdivided in the following specific objectives:

- Improving the architecture of the ECG prototype device from previous work in order to build a more compact system that will meet the requirements of portability and esthetical robustness.

- Refining and further developing the system software to ensure a good performance of the device in terms of cardiac activity monitoring and image acquisition. This will be achieved by modifying and enhancing the algorithms to calculate the heart rate and to activate the TTL pulse that will trigger the imaging process.

- Introduction of new features to enlarge the device's functionality. Body temperature of the small animal would be another interesting biomedical parameter to track for proper animal's physiological status follow-up.

- Testing the ECG device prototype with live animals during a pre-clinical experimental study to verify its correct operation.

On the whole, the aim is to transform this prototype into a final instrument ready to be used in practice.

1.4 Thesis structure

This thesis is structured in an intuitive and coherent way.

First in this part, the motivation, state of the art and objectives have clearly defined the scopes of this project.

Second some theoretical background on the fundamentals of the heart and ECG measurements are presented. At the same time the basic imaging modalities used in gated imaging will be also explained. Consequently cardiac gating imaging process would be easy to understand.

Third, the materials used in the development of the ECG device are technically analysed. The individual functions of each component will show how the different parts complement each other in order to achieve the ultimate tasks and operations of the system.

Further, the software implementation will be deeply explained to demonstrate the methods needed for calculating the heart rate, activating the TTL pulse or measuring temperature.

After that, the ECG prototype measurement results are evaluated in order to assess if the intended objectives have been accomplished.

In conclusion the current status of the system is examined to prospect any future improvements necessary for its completion and straight applicability.

Additionally, the scientific and social impact of this project is described and the estimated budget, in terms of cost, is included.

2 Theoretical background

In this section a review of the basic concepts on heart function, electrical cardiac activity recording and imaging modalities are presented. The acquired knowledge will set the fundamental multidisciplinary background necessary to further understand how gated imaging works.

2.1 Heart and electrocardiography fundamentals

In this section the function of the heart is precisely explained based on the anatomy and physiology of the human heart. Despite this project is focused on rodents, it has been shown that murine and human hearts share in general similar anatomical and electrophysiological properties that make possible the extrapolation [8].

2.1.1 The heart as a pump

The heart is a vital muscular organ present in humans and other small animals which primary function is to pump blood to the circulatory system of the body by contracting periodically into itself.

The heart is divided in four different chambers: the two upper ones are the atria and the two lower ones are the ventricles. In the left part of the heart, the left atrium and ventricle pump oxygenated blood coming from the lungs to the rest of the body; while at the same time in the right heart, the right atrium and ventricle pump deoxygenated blood coming from the organs to the lungs.

These events happen in a well-synchronized rhythmical manner defining the cardiac cycle as Figure 4 illustrates.

First, the heart is in a period of relaxation called diastole. Blood that had previously entered the heart and has accumulated in the atria, flows passively through the atrioventricular valves to fill the ventricles. At the end of diastole, both atria contract to propel additional blood just before the atrioventricular valves close. In a second stage, diastole is followed by a period of contraction called systole in which ventricles expel blood out of the heart [9].

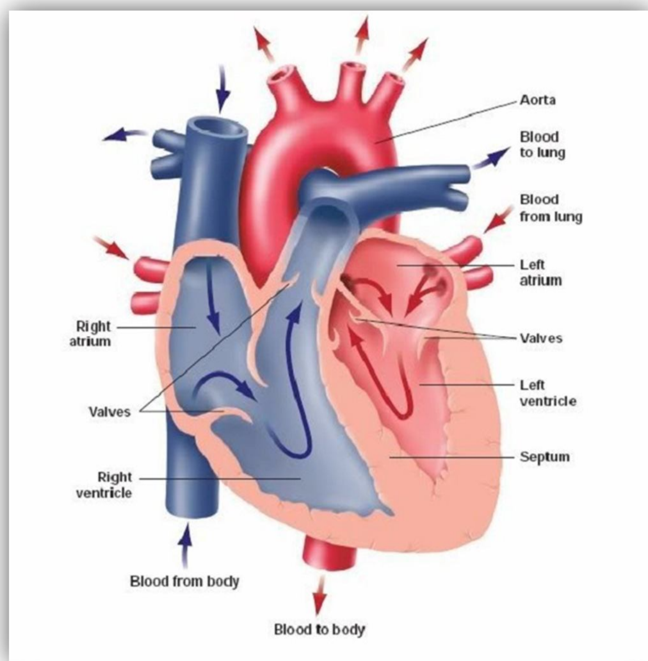


Figure 4. Structure of the heart and blood circulation through the different chambers and valves [9].

The contraction of atrial and ventricular muscle cells relies on the electrical stimulus of specialized excitatory and conductive muscle fibres. This cardiac muscle fibres are actually made up of individual contractile muscle cells interconnected one to another by a common permeable junction called *intercalated disk* that allows diffusion of ions. These fibres have the ability of triggering an automatic electrical discharge in the form of an action potential (AP), as well as to conduct the AP to neighbouring muscle cells of the heart through the intercalated disks as Figure 5a shows.

They have their membrane polarized which means that the concentration of electrical charges in the form of ions such as Na+, K+ or Ca2+, is different inside and outside the phospholipid bilayer. In normal conditions, the membrane of these muscle fibres is at a resting negative potential of -90 millivolts (mV).

When an action potential reaches a contractile cell, depolarisation of the cell membrane occurs. The sudden opening of fast sodium channels generates a rapid influx of Na+ inside the membrane causing the membrane potential to rise up to a peak value of +10 mV. During about 0.2 seconds the membrane potential starts repolarisation but remains positive exhibiting a plateau as shown in Figure 5b.

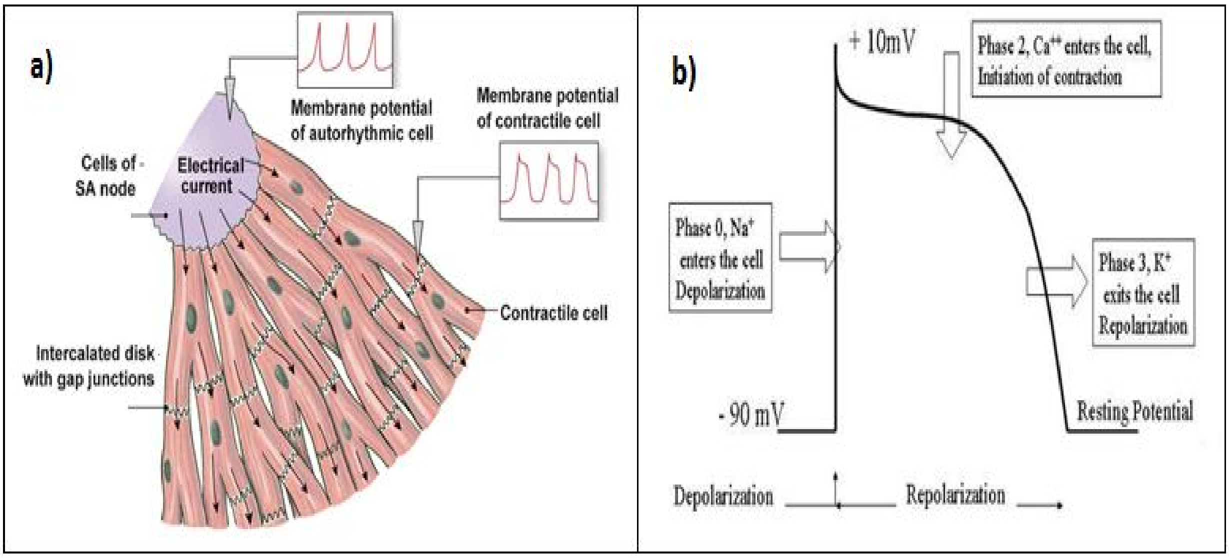


Figure 5. a) Action potential propagation through gap junctions of specialized fibres.
b) Action potential of cardiac muscle cells [10].

The plateau is due to the opening of slow calcium channels that allow Ca2+ go inside the cell and trigger the muscle fiber contraction. Once the calcium channels close, the permeability of potassium ion increases abruptly producing an out flux of K+ ions that reduces the membrane potential to its initial negative resting value [11].

The cardiac rhythmicity is ensured by a coordinated electrical activity that paces the succession of contractions during one heartbeat. Indeed the conduction of an action potential that propagates through the heart, activates the specialized excitatory fibres in a well-defined sequential order as shown in Figure 6.

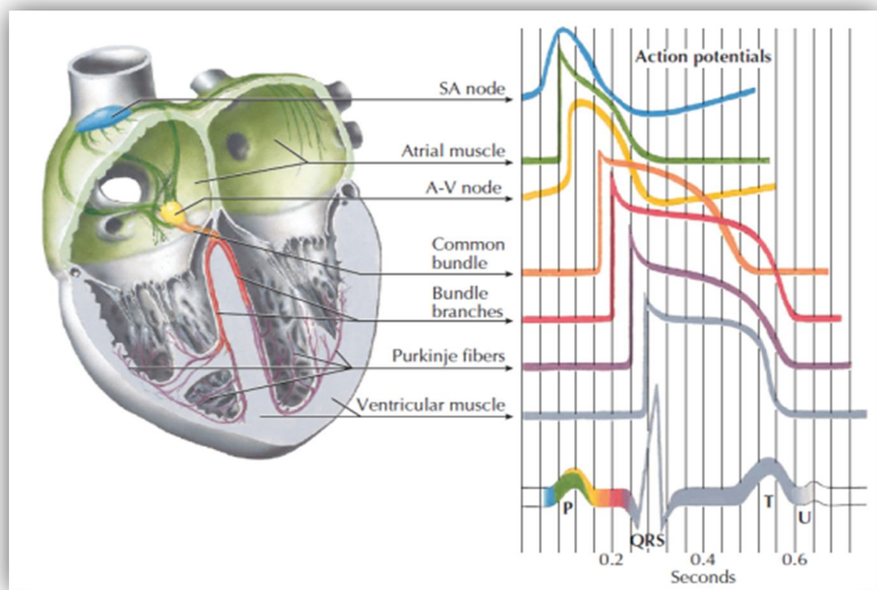


Figure 6. Excitatory and conductive system of the heart [12].

First, the cardiac cycle initiates when an action potential arises at the sinoatrial (S-A) node, also called the pacemaker of the heart, located in the right atrium as it is shown in Figure 5a. The internodal pathways in the atrial muscle conduct the impulses to the atrioventricular (A-V) node, which has the function of delaying the transmission of the AP. This time lapse allows the atria to empty their blood into the ventricles. After that, the action potential is transmitted by means of the A-V bundle into the right and left ventricular bundle branches. Finally it propagates through the Purkinje fibres which conduct the cardiac impulses to all ventricular muscle cells that contract.

The action potential changes its morphology throughout the conductive system of the heart and it does not propagate with a uniform velocity. Indeed the conduction velocity in the atrium is much slower than in the ventricles. This fact will have a direct effect in the shape of the recorded electrical activity of the heart which will be further explained in the next section.

2.1.2 Electrocardiogram (ECG)

The electrocardiogram (ECG) is the graphical representation of the electrical activity of the heart during the cardiac cycle. It is measured with an electrocardiograph by means of placing electrodes in the skin surface to record potential variations. The information extracted from this measurements gives clues for diagnose of the myocardium's conductive system function.

To understand how the action potentials activating the different muscle fibres of the heart can be sensed at the surface of the body, a simple electrocardiographic model has to be applied [13]. The heart is assumed to be an electrical generator. Thus its electrical activity can be modelled as a global current dipole located at the centre of the heart which generates an electrical field. The amplitude and orientation of this dipole vary along the cardiac cycle due to action potential propagation in different layers of the myocardium. Moreover the body is assumed to be a homogenous medium having uniform conductivity with the body tissues being purely resistive.

Consequently, the electrical potential at any point of the skin surface can be expressed as a combination of the dipole heart vector components. Then by placing two electrodes at two different locations, the potential difference along the stretch between them, called lead, can be measured. There are different combinations of leads to characterize the electrical cardiac activity: the Einthoven (I, II, III), Goldberg (aVR, aVL, aVF) and Wilson (V1 to V6) configurations [14]. All these lead configurations put together define the twelve standard leads.

For the sake of simplicity the Einthoven triangle configuration is the most frequently used in practice. Figure 7 shows the placement of the three electrodes in the extremity of the right arm (RA), left arm (LA) and left leg (LL), which form the three bipolar limb leads (I, II, III).

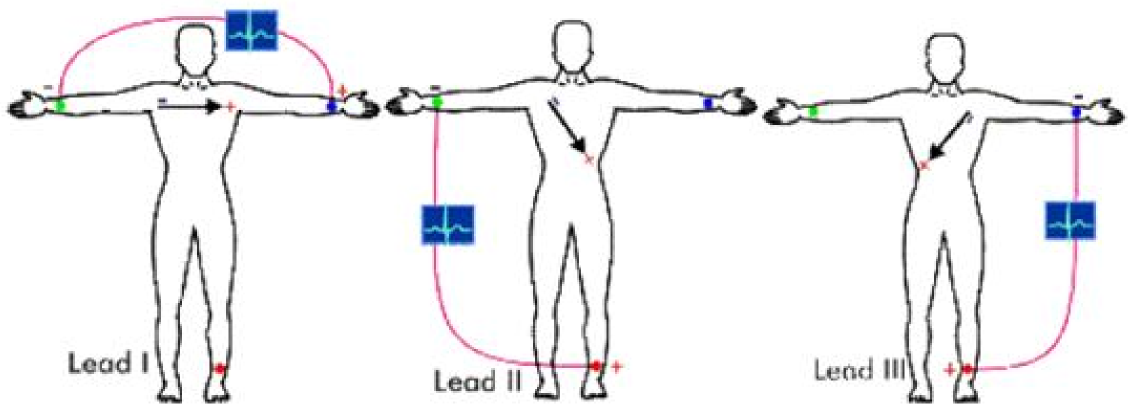


Figure 7. Einthoven standard bipolar limb leads [7].

The electrical information provided by leads I, II, III, as observed in Figure 8 is:

- Lead I: Voltage difference between left arm and right arm.
- Lead II: Voltage difference between left leg and right arm.
- Lead III: Voltage difference between left leg and left arm.

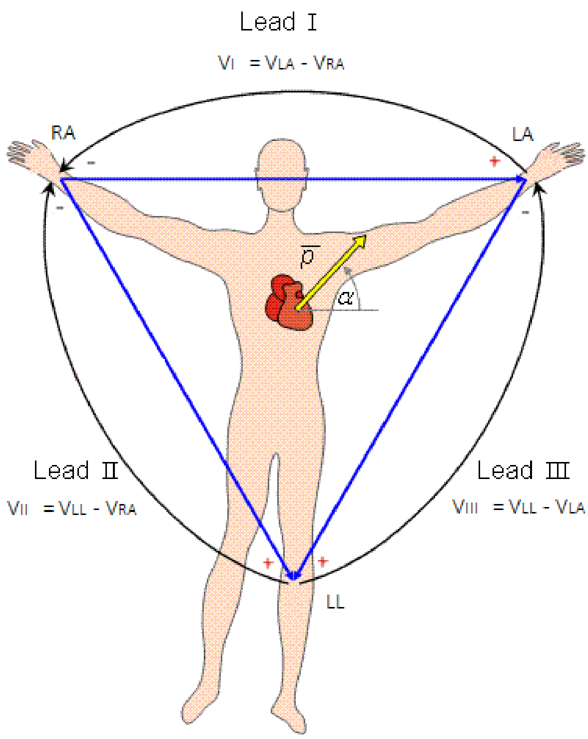


Figure 8. Einthoven triangle configuration. Adapted from [13].

Every lead records the electrical activity of the heart along a particular axis by projecting the cardiac vector along the directions of the leads as shown in in Figure 9.

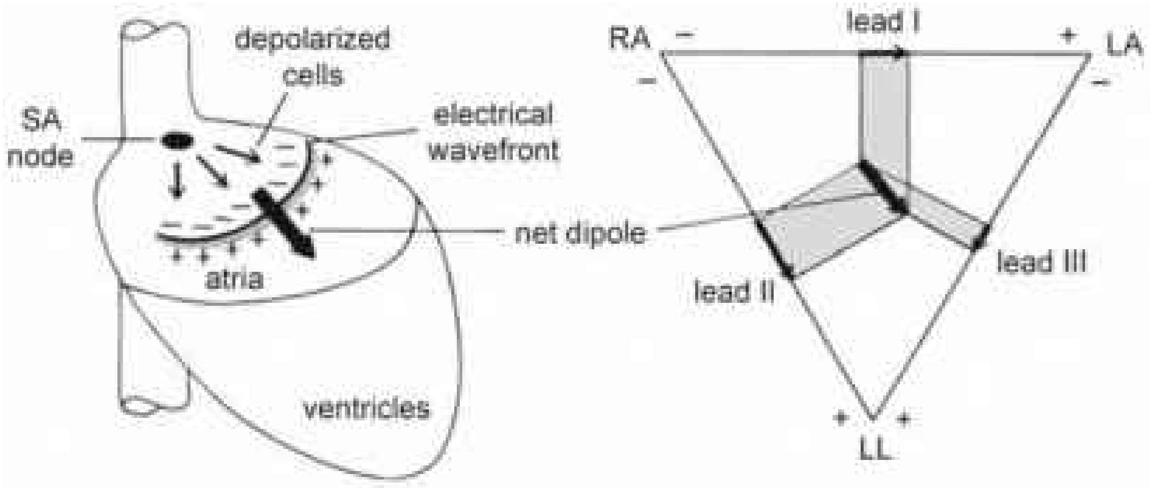


Figure 9. Net heart dipole and its projections in the Einthoven triangle [15].

The most useful lead for analysis of the ECG signal is lead II since it is the one that lies on the same direction as the cardiac axis. For example the first P deflection initiating the ECG signal is positive since the atrial depolarization net dipole follows the positive parallel direction of lead II axis. Focusing in this preferred lead it can be studied how the different waveforms P, Q, R, S and T arise and therefore their correspondence to specific events of the cardiac cycle.

The correlation between the parts of ECG and the phases of the cardiac cycle can be explained based on Figure 10.

- P wave: Corresponds to atrial depolarization and the spread of this excitation in the atria. Its amplitude is normally less than 300 μV, and its duration is less than 120 ms.

- QRS complex: Indicates ventricular depolarization and spread of excitation in the right and left ventricles. It has the largest amplitude of the ECG waveforms, sometimes reaching 2-3 mV. It lasts for about 70-110 ms.

- T wave: Represents ventricular repolarization. Its typical duration is 300 ms.

It is important to note that atria repolarization is not observable in the ECG signal due to its late occurrence that coincides with ventricles depolarization. Therefore, the high positive QRS complex masks its representation [16].

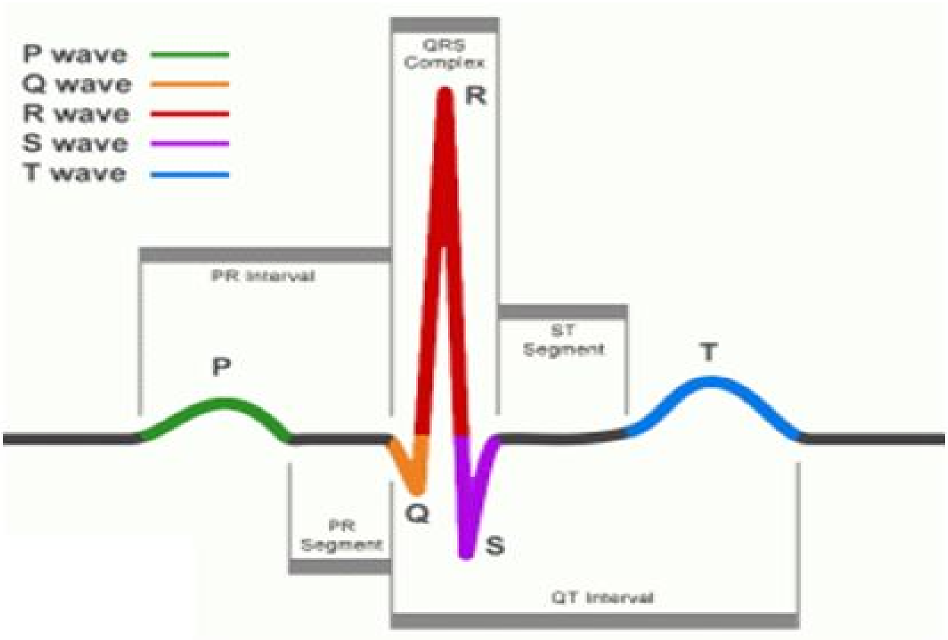


Figure 10. ECG of normal sinus rhythm. Adapted from [17].

2.1.3 Human and murine cardiac characteristics

There are some relevant differences between the murine and human hearts regarding the anatomical characteristics, pumping volumes and capacity, as well as in electrophysiology of the ECG [8]. Table 1 summarizes the more relevant data.

As it could be expected, the reduced size of rodents with respect to humans entails a reduction in the mass of heart which is in average 0.15g for rodents versus 300g in humans. However the ratio of the heart and body weights appear to be equal to 0.005 in both species which means that at the end everything is proportional. The same happens in hemodynamic: the cardiac output or stroke volume for example are significantly smaller in mice than in humans but are in accordance with the body size and weight.

Table 1. Comparison of mouse and human heart physiology [8].

	Human	Mouse
GENERAL		
Body mass (kg)	58–85	0.015–0.043
Lifespan (year)	70–80	2–2.5
Basal metabolic rate (kJ/d)	6279	15.6
Basal metabolic rate (O2 consumption L/(kg*h))	0.9	0.8–3
HEART		
Heart weight (g)	261–366	0.12–0.17
Heart weight/body weight ratio (kg/kg)	0.004–0.006	0.004–0.005
HEMODYNAMIC		
Stroke volume (mL)	50–100	0.015–0.05
Cardiac output (L/min)	4–8	0.005–0.03
Blood pressure (mean arterial pressure, mmHg)	88–100	73–125
Blood volume (L)	5–6	0.002–0.03

	Human	Mouse
CARDIAC EP		
Heart rate (beats/min)	56–101	500–724
PR interval (ms)	120–200	30–56
QRS duration (ms)	84–110	9–30
QT (ms)	385	29–109
QTc (ms)	398–430	30–124
Atrial ERP (ms)	172–245	23–71
Atrial CV (cm/s)	88	30–60
AV Wenckebach CL (ms)	329–453	66–133
Ventricular ERP (ms)	223–257	33–80
Ventricular CV (cm/s)	80	30–60

It is also worth pointing out some differences in electrophysiology. As it can be observed in Figure 11a, atrial action potential duration in rodents is much shorter than in humans. Nevertheless the amplitude of the depolarisation peak has almost the same amplitude and the overall shape does not show any other major dissimilarity. As a consequence the "zoomed in" morphology of the ECG signal in mouse is identical to the human ECG except from the fact that in mouse the frequency is much higher as shown in Figure 11b. Indeed mice heart rate is around 500-724 bpm, around 250-400 bpm [18] in rats, whereas in humans it is within the range of 56-100 bpm. Therefore the bigger the heart is, the greater cardiac output and the lower the heart rate.

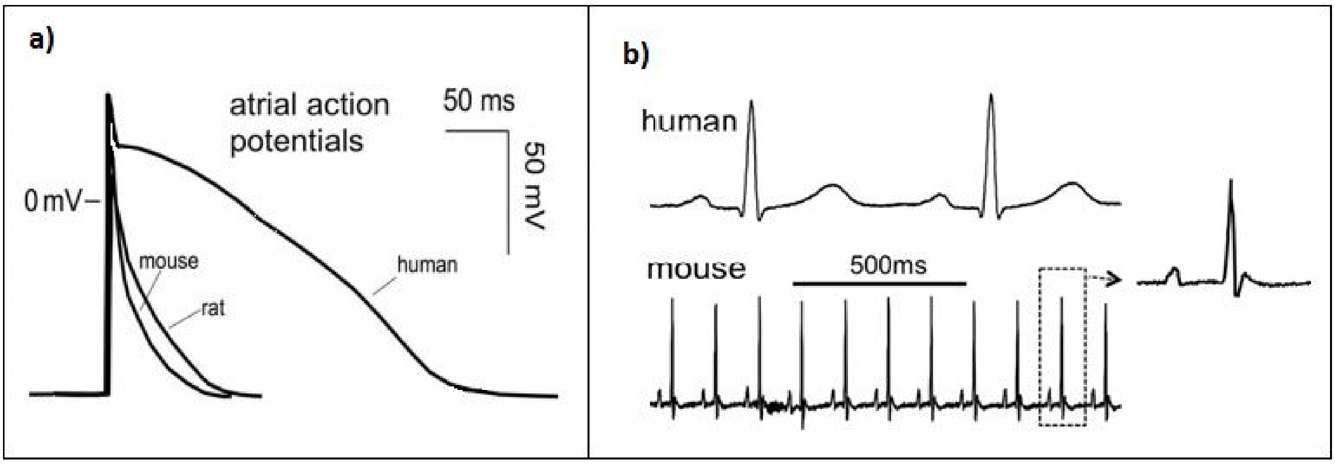


Figure 11. a) Comparison of atrial action potential morphology in human, mouse and rat. b) Comparison of murine and human ECG in lead I over a period of 1.5 seconds of time [8].

2.1.4 Bioelectrical ECG signal processing

Although it would be desirable to record directly from the electrodes a perfect ECG signal, this actually does not happen in reality. The morphology of the ECG gets usually distorted by the introduction of noise. The most common types of noise when recording the cardiac activity are:

- **Baseline Wander:** It is a low frequency noise below 0.5 Hz caused by body movement, respiration or perspiration (variations in electrode impedance). The result is that the ECG signal seems to be "riding" in a low frequency wave. The solution to avoid this type of distortion is the implementation of and analog or digital linear time-invariant high pass filter with 0.5 Hz cut-off frequency.

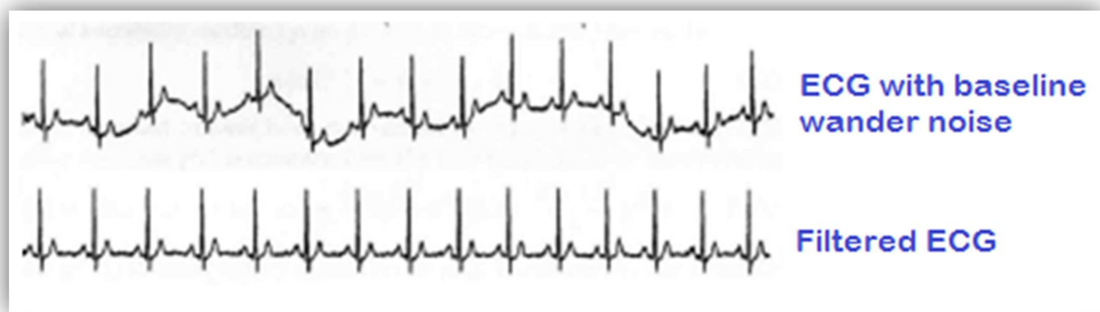


Figure 12. Baseline wander filtering.

- **Power line interference:** It is a sinusoidal noise of 50-60Hz introduced by electromagnetic fields of power lines. It can be removed by using a notch or band-stop filter centred at that specific frequency through analog or digital implementation.

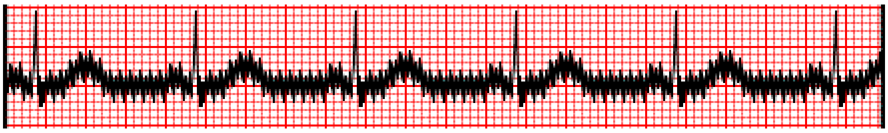


Figure 13. Power line interference noise [19].

- **Muscular noise:** It is a type of noise present in the broad-band of 20-1000Hz which is difficult to eliminate since it overlaps with the spectral components of the ECG waveforms. Usually adaptive filtering is applied to clean the signal or low pass filters with cut-off frequencies greater than 40Hz.

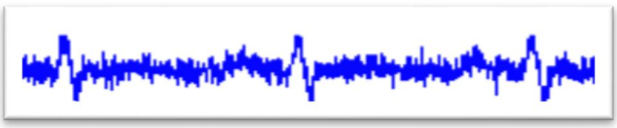


Figure 14. Muscular noise. Adapetd from [20].

2.2 Medical imaging modalities

In this section the more relevant imaging modalities will be explained without entering in deep details, just to provide the basic concepts necessary to understand how gated imaging works.

It is important to highlight that the aim of this bachelor thesis is not to focus on medical imaging but on instrument devices to be coupled to imaging machines in order to improve the final output and results. Consequently the study will focus on X-ray Computer Tomography (CT) and Positron Emission Tomography (PET) since these are the imaging modalities employed by the Argus scan at Hospital General Universitario Gregorio Marañón.

2.2.1 Computed tomography (CT)

Computed Tomography is a revolutionary image modality based on X-Ray imaging that is able to acquire high-quality cross-sectional images of internal structures of the body. In order to completely understand how CT imaging works a quick review of medical X-ray imaging is presented next.

In conventional X-ray radiography, the patient is irradiated with x-ray electromagnetic waves of energy between 5 to 150 keV produced in an X-ray tube. The projected X-rays go through the body of the patient and are finally captured by a photographic film or digital detector forming a two dimensional image based on X-ray absorption. Indeed X-rays interact with matter in three different ways: some simply penetrate it unaffected by transmission, others are deflected by scattering, and a third portion are blocked by absorption of the tissue. Image production relies on the fact that different parts of the anatomy absorb different amounts of X-rays depending on the particular density and composition of that anatomical structure. Therefore X-rays emerging from the body produce an image called shadowgram where each point has a different brightness in relation with X-ray intensity [21].

Computed axial tomography combines different projections of X-ray images taken from different angles to generate virtual slices of the scanned patient. The final images are sectional maps of the X-ray attenuation coefficients of different tissues along one ray. Complex reconstruction algorithms have to be applied to create a three-dimensional image of the inside of the patient from a large series of two-dimensional radiographic images taken around a single axis of rotation.

Different generations of CT scans have been developed since the early 1970's, but the most basic widely adopted design is the third generation one shown in Figure 15. It is composed of a rotating tube of X-rays emitting a fan beam. A ring array of scintillator detectors rotates at the same time around the patient, as the X-ray tube, until a complete turn of 360º is performed. After that the patient is moved in the perpendicular plane to continue obtaining projections that will form the next crosssection. Once the imaging process has finalized, all the cross-sections are put together to form a three-dimensional object of the internal structure of the body. To illustrate next figure presents the 3D abdominal internal reconstructed anatomy.

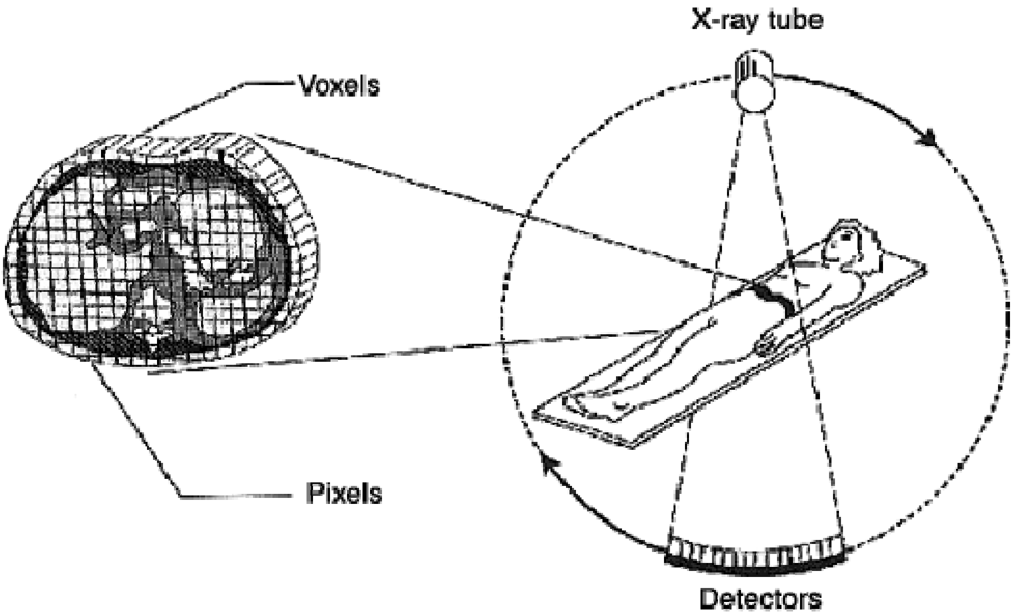


Figure 15. X-ray Computed Tomography acquisition [22].

2.2.2 Positron emission tomography (PET)

Positron emission tomography (PET) is a nuclear medicine imaging technique detecting gamma radiation emitted by a short-lived radioactive tracer isotope which is introduced into the body of the patient. Some examples of major positron emitting radionuclides are: carbon-11, nitrogen-13, oxygen-15 and fluorine-18.

The radionuclide can be either injected intravenously, rebreathed or ingested. But in any case it enters the circulatory blood system and distributes across the body. At the moment the target tissue is reached, the radioactive nuclei decays and emits a positron, the antiparticle of the electron with opposite charge. This free positron particle can annihilate with an electron producing a pair of annihilation gamma photons that travel in opposite directions at an angle of 180º with an energy of 511 keV each. When the ring of scintillator detectors surrounding the patient records the gamma rays at two opposite locations simultaneously, it transforms the photons in an electrical signal to construct the image.

Actually, the event of annihilation has occurred somewhere along the line connecting the two detectors. By acquiring views of the tracer distribution from a variety of angles, the three-dimensional tracer distribution within the body can be reconstructed by tomographic analysis procedures [23].

PET is considered a functional imaging technique since the recording of spatial and temporal concentrations of radioactive isotopes within the body allows to quantitatively describe functional processes, through acquisition of parameters such as blood flow, rates of metabolism, or receptor binding.

Figure 16 below is a graphical representation of the succession of events leading to PET image reconstruction.

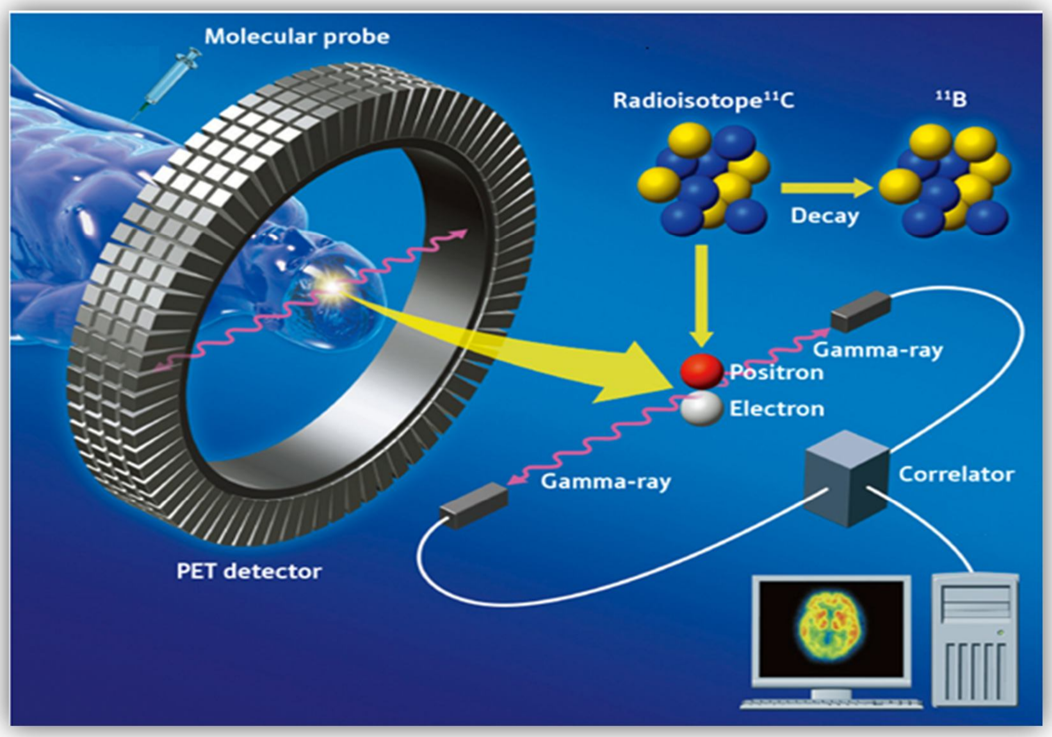


Figure 16. Molecular imaging by PET [24].

2.2.3 Single Photon Emission Computed Tomography (SPECT)

Single Photon Emission Tomography is a nuclear medicine imaging modality relying on the same working principle as PET but detecting single gamma photons with a gamma camera. Indeed, in contrast to PET imaging no annihilation between a positron and an electron occurs since the injected radionuclide directly generates gamma radiation.

The detection of a single photon instead of two coincident ones in opposite directions as in PET imaging, produces lower spatial resolution images. However, SPECT imaging is less expensive since it uses longer-lived isotopes which are produced more easily than PET short-lived ones. SPECT is also a tomographic modality which acquires 2D images at multiple angles in order to reconstruct the final 3D image showing the distribution and concentration of radiotracer inside the body. Registration of SPECT images with CT acquisitions of the same study provides useful functional localisation information as in myocardial perfusion imaging [25].

2.3 Electrocardiographic gated imaging

Respiration, body movement and blood flow are the principal causes of motion artefacts in medical imaging acquisitions. They are undesirable since they reduce image quality and might lead to mistaken diagnosis or inaccurate estimation of heart's hemodynamic.

In order to counteract motion blurring, electrocardiographic gated imaging synchronizes the ECG with the imaging modality. It can be implemented in two different ways: prospectively or retrospectively. In prospective gated imaging an electrocardiograph device is needed to trigger the start of the image acquisitions at the end of the diastole phase. In contrast, retrospective gated imaging makes use of continuous scanning in coordination with ECG recording in parallel for post-reconstruction of the images at the phases of interest. Since the primarily aim here is to eliminate motion artefacts from the beginning, prospective gating is preferable.

Figure 17 shows the two different approaches in CT acquisition [26]. In prospective gating, radiation is only turned on at mid-diastole (70% of R-R interval) and turned off during the remaining cardiac contraction. In retrospective gating the heart is constantly irradiated by the X-ray beam and images are reconstructed afterwards. Depending on the application either one or the other techniques are more or less suitable. In fact when irradiation can be avoided prospective gating is more interesting. But in the case of coronary angiography for example for which a follow up over the whole cardiac cycle is necessary, retrospective gating would be a better option.

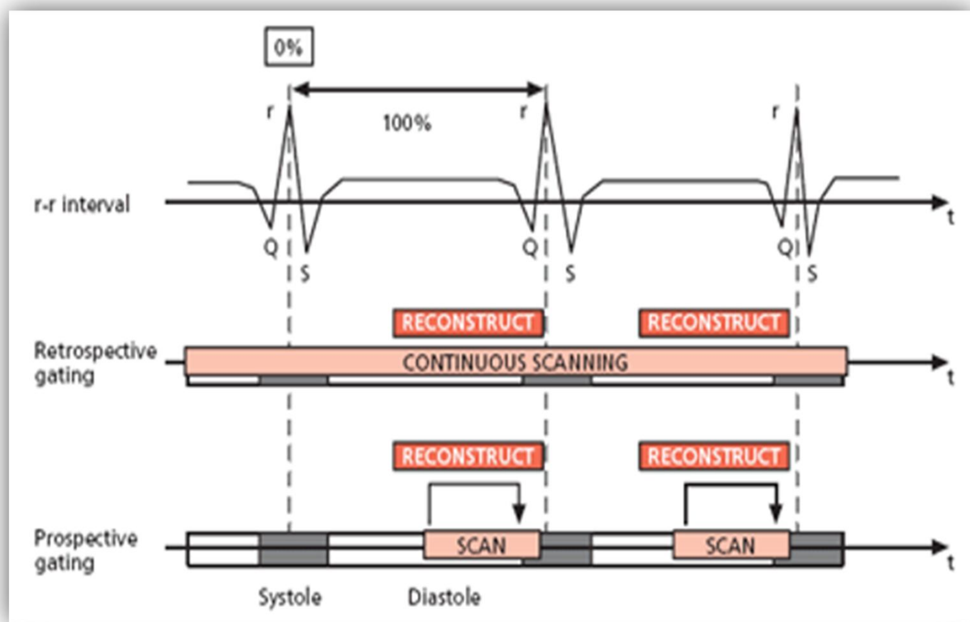


Figure 17. Prospective and retrospective gating CT imaging [26].

The principle of prospective gating in functional molecular imaging is shown in Figure 18a. It starts when the ECG monitoring device detects an R wave and sends a pulse to the PET or SPECT scan to start the acquisition.

The aim is to acquire a succession of images perfectly located in time spanning the total cardiac cycle. For this purpose the R-R interval is divided into 8 frames of equal duration. The images at different phases of cardiac cycle are separately stored to easily differentiate the different phases when the process is repeated for different projection angles [27]. Then, all the images stored in a specific frame can be used to reconstruct the corresponding phase of the cardiac cycle with particular interest in the end systole (ES) and end-diastole (ED) phases as shown in Figure 18b. Adding the eight temporal frames of a cycle an endocardial volume curve can be obtained as shown in Figure 18a below.

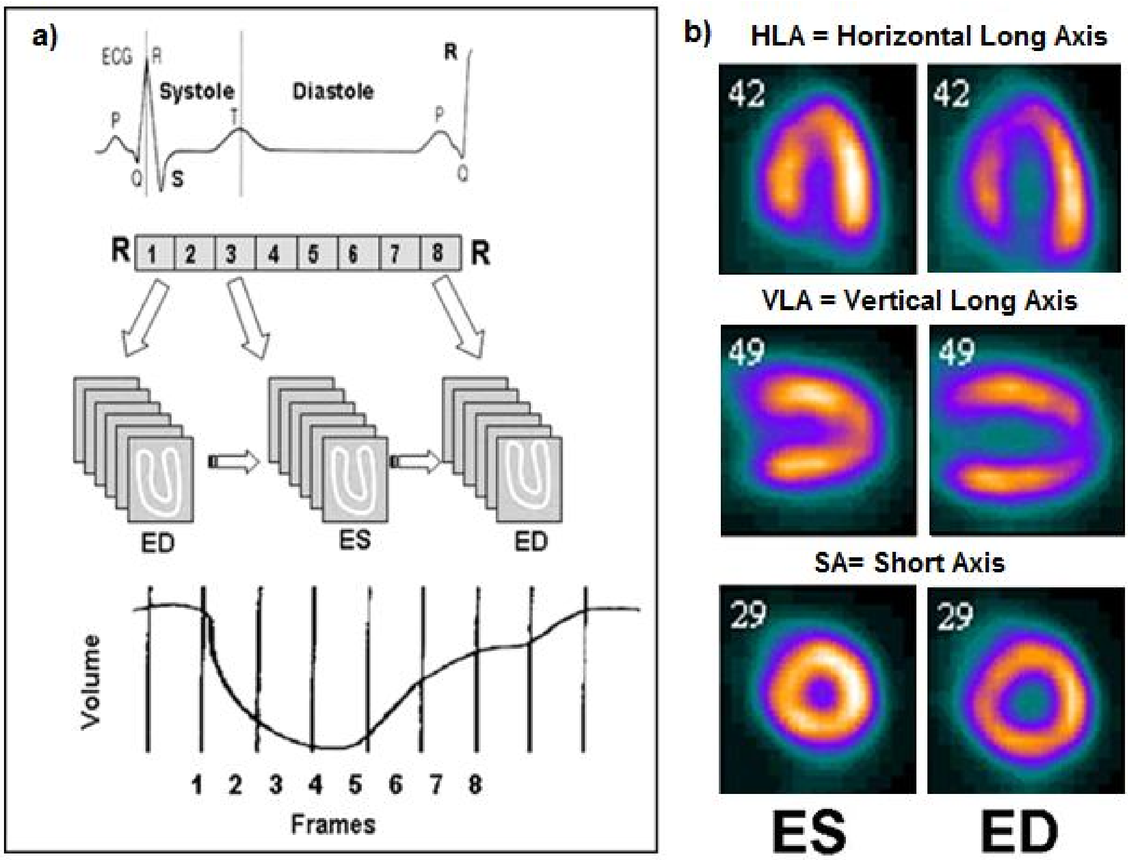


Figure 18. a) Principle of ECG-gated acquisition over one R–R interval on ECG [27].
b) PET images of end-systole and end-diastole at different acquisition axis.

Gated imaging becomes also a powerful tool to extract quantitative information about myocardial contractility and perfusion at different moments of the cardiac cycle. Further analysis of the cardiac images can give the cardiac output, stroke volume and ejection fraction easily.

3 Materials

This section presents the materials used in the development of the small ECG monitor for small animals. The different materials composing the ECG device prototype are technically described. In addition, complementary materials used during the programming and testing are explained.

3.1 ECG device architecture

The small-size ECG monitor consists of four main components: the microcontroller board Arduino MEGA2560, the analog-to-digital converter (ADC) ADS1298, the LCD screen with video controller Gameduino2 and the temperature sensor AD2210. All these parts are electronically connected through an adapter proto-board. The ensemble is embedded in a plastic box with input and output connectors that enable the device to be interfaced with the imaging machine and with a computer if necessary.

3.1.1 Arduino MEGA 2560

The Arduino Mega 2560 is an open source hardware microcontroller board based on the ATmega 2560 chip from ATMEL containing a processor core, memory, and programmable input/output peripherals in an integrated circuit. This 8 bits Atmel microcontroller that works under ultra-low power consumption of 1.8V and 500μA, relies on a RISC (reduced instruction set computer) architecture with a 16 MHz crystal oscillator clock [28]. Arduino Mega 2560 has a USB connector, a power jack, an ICSP header, and a reset button. It also counts with a total of 80 pins with different functions which can handle different types of input and output signals.

The board uses the ATmega16U2 programmed as a USB-to-serial converter that enables the upload of code to the Arduino Mega 2560 from the Arduino software environment through a virtual COM port [29].

The board can be powered either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V) by means of an external battery.

Figure 19 shows the different parts of the Arduino Mega 2560 board.

Taking as a reference the microcontroller ATMEL chip situated on the centre of the board, the 54 pins located on the top and to the left are digital input/output pins:

- Pins with numbers ranging from 2 to 13 can be used as PWM (pulse width modulation) outputs.
- Pin numbers 0, 1 and from 14 to 21 are used for serial communication. In fact eight of them are pairs of RX (receiver) and TX (transmitter) serial pins conforming four UARTs (universal asynchronous receiver/transmitter) that act as an intermediary between parallel and serial interfaces. The two remaining communication pins SDA (serial data) and SCL (serial clock) rely on the TWI (two wire interface) or I2C serial communication protocol, to respectively transmit data and clock pulses.
- Pin numbers ranging from 22 to 54 are basic digital pins. It is worth to point out pins 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) that support SPI (Serial Peripheral Interface) communication.

Below the Atmel chip, 5 power pins and 16 analog input pins can be found:

- Power pins: 5V and VIN pins output 5 volts from the regulator of the board. 3V3 pin supplies 3.3 volts. GND pin acts as ground.
- Analog pins A0 to A15 measure voltage from ground to 5 volts with 10 bits of resolution which corresponds to integer values between 0 and 1023. Consequently the resolution between readings is of: 5 volts / 1024 units or 4.9 mV per unit.

For further detail regarding the pin configurations on Arduino 2560 board refer to Appendix 1: Arduino Mega 2560 pin diagram and mapping table.

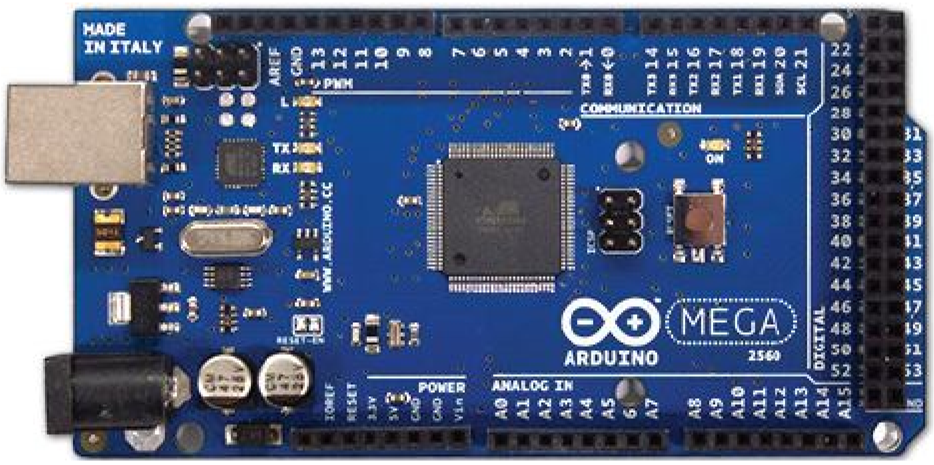


Figure 19. Arduino Mega 2560 board [29].

The main characteristics of Arduino Mega 2560 concerning voltage, current and memory specifications are presented in Table 2.

Table 2. Arduino Mega 2560 characteristics [29].

Arduino Mega 2560	
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Arduino software includes a serial monitor which gives the possibility to send data from and to the board. The ATmega2560 also supports TWI and SPI communication that will enable the cross-talk with the Gameduino2 and ADS1298.

3.1.2 ADS1298 ECG FE-PDK

The ADS1298 ECG Front End Performance Demonstration Kit developed by Texas instruments is an 8 channel continuous sampling of 24 bit resolution analog to digital converter (ADC) aimed for bio potential measurements. It counts with a built-in internal reference of either 2.4V or 4V, an on-board oscillator of 2.048MHz, as well as with eight low noise programmable gain amplifiers (PGAs) providing a gain of 1, 2, 3, 4, 6, 8, or 12. Each channel consumes a power of 0.75 mW which in combination with power down and standby modes yields a low power consumption. The sampling frequency can be programmable ranging from 250 SPS to 32 kSPS. The CMRR is -115 dB and the operating temperature range spans -40ºC to 85ºC [30].

The high precision simultaneous multichannel signal acquisitions make medical ECG recordings, along with EMG and EEG, one of the most suitable applications of this ADC. Furthermore, the ADS1298 presents specific useful features for ECG monitoring such as pace detection, electrode's lead off detection, integrated respiration impedance or Wilson Central Terminal.

The ADS1298 has a DB 15 connector for electrode signal inputs, as well as three blocks of pins J3, J4 and J5 for interfacing with a microcontroller through SPI communication as it can be seen in the top view of Figure 20. In the bottom view, the different levels of integration conforming the ADS1298 are shown.

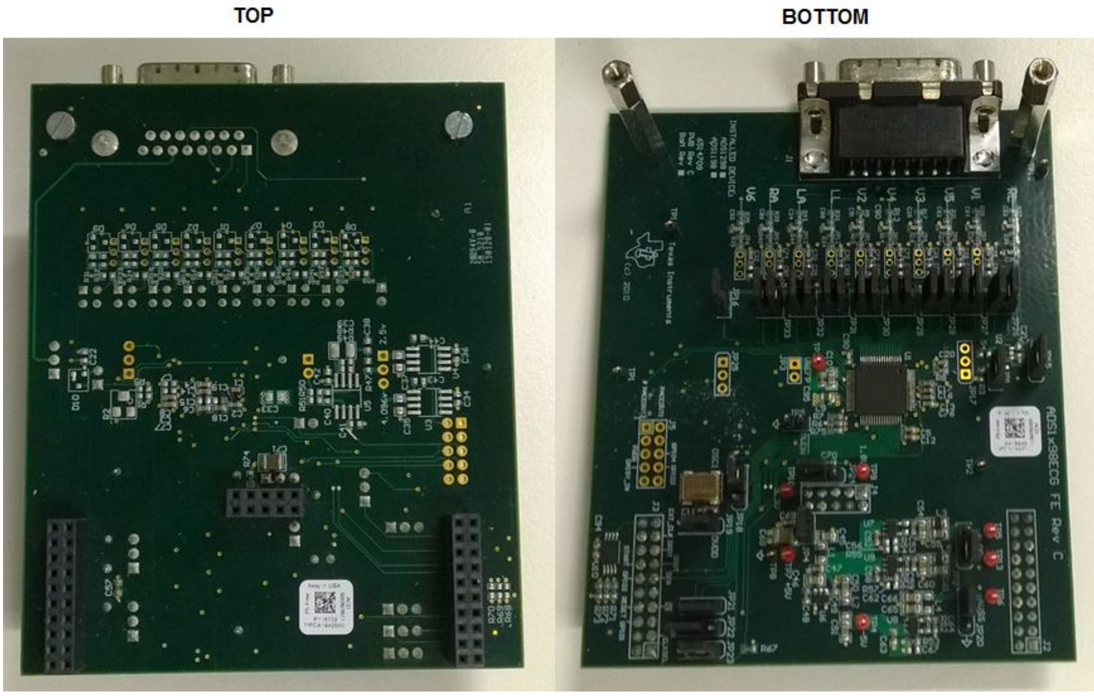


Figure 20. Top and bottom views of ADS1298 FE-PDK

The successive blocks transforming the analog to the digital signal can be easily studied in the circuit diagram of Figure 21.

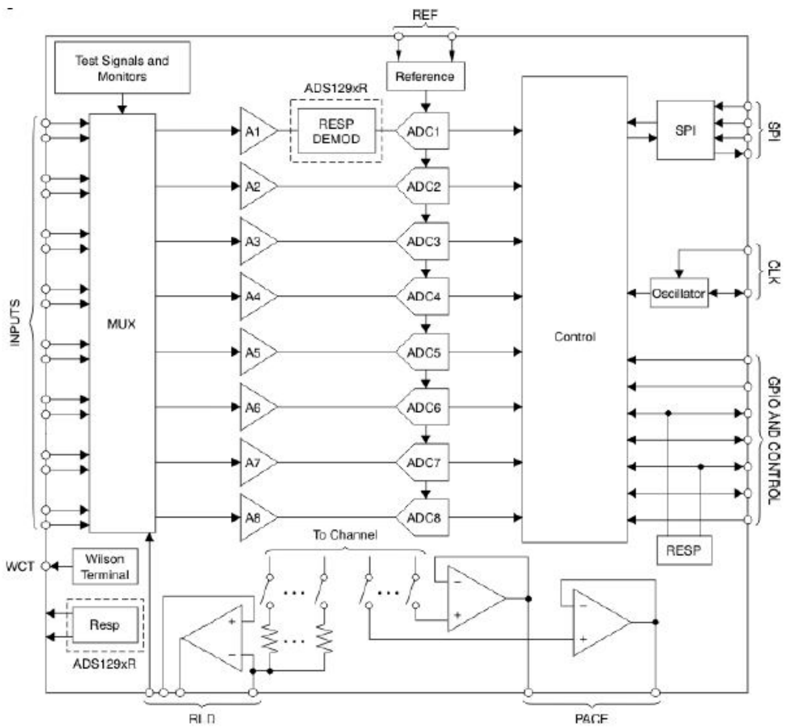


Figure 21. Circuit diagram of ADS1298 [30].

The ADS1298 counts with some basic filtering elements to reduce the unwanted introduction of noise in the bioelectrical signals of small amplitude. On one hand, it is important to highlight the presence of electromagnetic interference (EMI) analog filters at the inputs of every channel before the multiplexing stage. These passive RC filters of gain -3 dB at 3MHz eliminate the noise due to high electromagnetic frequencies.

On the other hand, ADS1298 implements also digital decimation filters in each channel after the modulator. These filters decimate the output signals received from the modulator by implementing third order low pass *sinc* filters that attenuate high frequency noise. The decimation ratios are modulated accordingly to the sampling frequency. Indeed the frequency response shows zeroes at sampling rate and its multiple frequencies giving infinite attenuation.

3.1.3 Gameduino 2

Gameduino 2 is an open source graphic and sound adapter shield for Arduino. As illustrated in Figure 22, it consists of a 4.3 inch touchscreen able to display 480x272 pixels in 24 bit colours. It also incorporates a powerful graphic, touch and sound controller chip: the FT800. In addition, Gameduino2 is composed of an accelerometer, a microSD slot and a jack headphone which adds the extra feature of sound output.

The FT800 chip can use either SPI or I2C connections to communicate with other microcontrollers. The SPI interface operates up to 30MHz in SPI mode 0. It uses a 12MHz crystal oscillator with a default system clock of 48MHz [31]. That's why in the bottom view of Figure 22, the SCK, MISO, MOSI, and GD SEL pins are present in addition to the basic power pins and accelerometer ones.

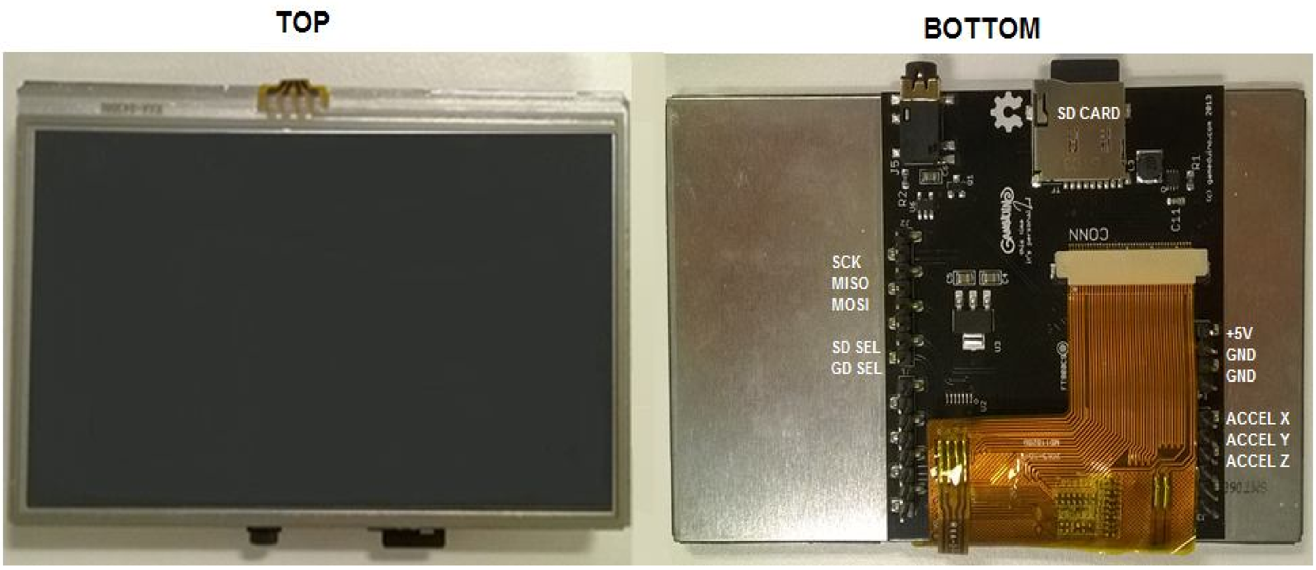


Figure 22. Gameduino 2 screen: top and bottom views

There are three main relevant features supported by the FT800 chip. First the 256 Kbytes of video RAM enables smooth circle, line and text drawing as well as JPEG display on the screen. Tags and track of touch movement can be recognized by the touch force sensing characteristic of the screen. Moreover, the system can play tones from video memory up to 48 KHz. As it can be seen in the block diagram of Figure 23 below, the three previous features are controlled by graphic, audio and touch engines in the memory and processing unit of the FT800 which in turn is in contact with the microcontroller unit (MCU) interface.

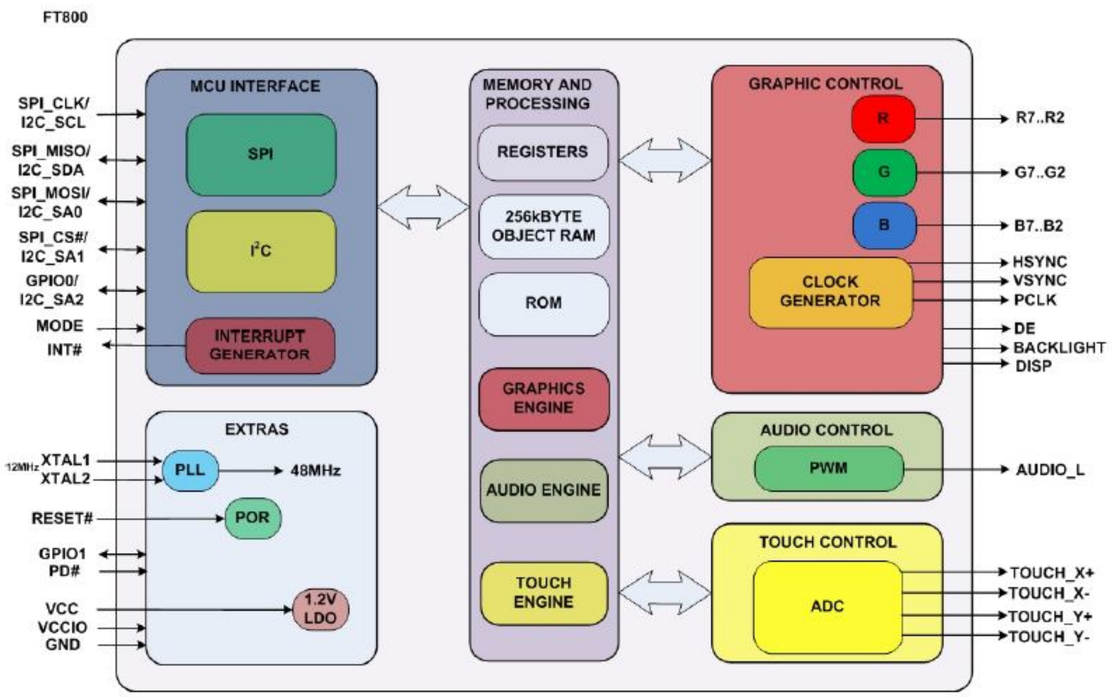


Figure 23. FT800 graphic, audio and touch microcontroller block diagram [31].

3.1.4 Temperature sensor AD22100

The AD22100 is a monolithic temperature sensor able to sense 200ºC of temperature span in the range of -50ºC to +150ºC. The sensitivity of this sensor is 22.5mV/ºC with an accuracy better than ±2% and a linearity better than ±1% [32]. It is ideal for instrumentation applications due to the incorporation of signal conditioning which greatly simplifies the system design by eliminating the need for trimming or buffering.

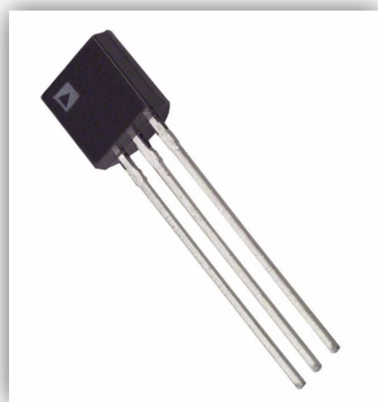
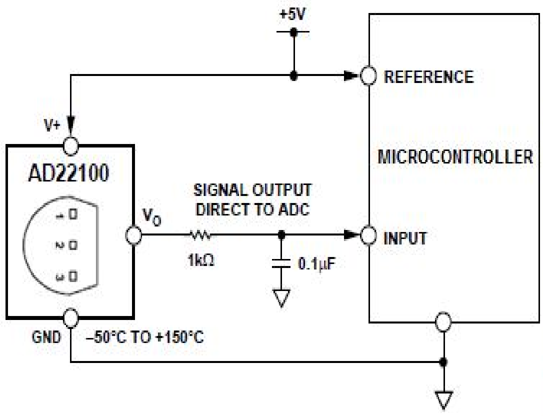


Figure 24. AD22100 [32].

Other relevant features making this sensor suitable to complement the small-size ECG monitor are the reverse voltage protection, minimal self-heating, low impedance output and single supply operation.

The AD22100 is a ratiometric sensor meaning that the output voltage will be proportional to the measured temperature times the supply voltage which in this case corresponds to the +5 Volts delivered by the Arduino microcontroller. In such a manner the output voltage swings from 0.25V at -50ºC to +4.75V at +150ºC.

In order to output the temperature variations detected by the AD22100 sensor into an ADC or microcontroller such Arduino, a 1 kΩ resistor and a 0.1 μF capacitor are recommended to be placed at the input of the microcontroller as illustrated in Figure 25 below.



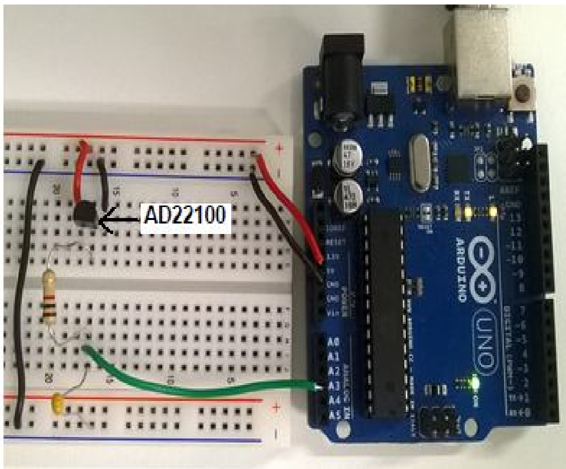


Figure 25. Implementation circuit for temperature sensor AD22100. Adapted from [32].

3.1.5 Enclosure box and complements

In order to build a robust ECG device all the previous components have to be assembled together and contained in a plastic box with any embedded complements necessary for power and input/output connectivity.

The enclosure box used is the RETEX model Series 103 P 33103005 of dimensions 190x115x72mm as shown in Figure 26. It is made of ABS plastic and aluminium cover of 1 mm thickness.

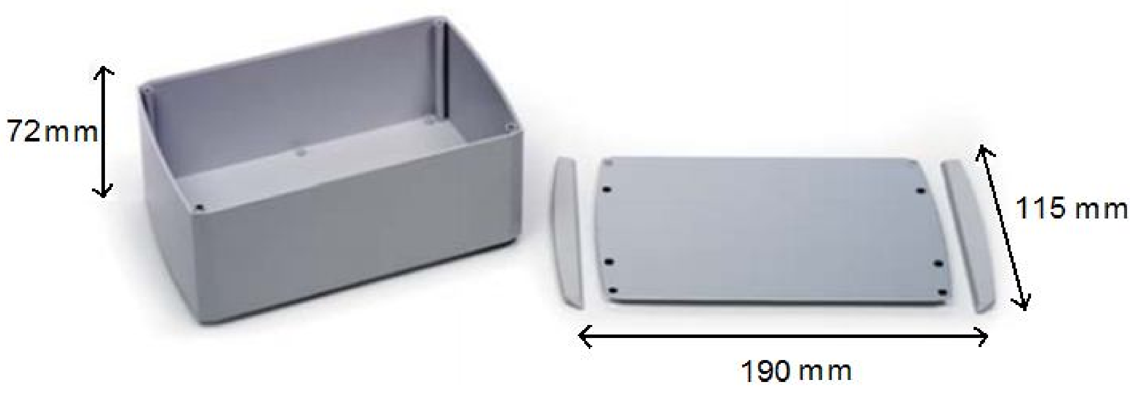


Figure 26. RETEX box enclosure dimensions [33].

On one hand, three lead electrodes of two different types were used for recording either the simulated or biological ECG input signals during the testing of the device. The first type used in the laboratory were the common spring clip electrodes typically used for human recording. However small animal experimental trials required needle electrodes, which were more appropriate for miniaturization.

On the other hand, a BNC female connector embedded at the surface of the enclosure and a BNC male coaxial cable were used to output the TTL pulse signal from the ECG device to trigger image acquisition.

A series combination of four 1.5Volts batteries was used as external power source to make the device portable. A complementary power switch was also needed to turn on and off the current supplied to the device.

Figure 27. Needle electrodes, BNC cable and 6V batteries.

Other extra materials used for the construction of the device were unipolar, twisted pair and coaxial cables, as well as pins, and a copper proto-board.

3.2 Testing and programming instruments

In order to program and test the small-size ECG device a variety of instruments and tools were used. On the one hand input ECG signals needed to be generated by a variety of electrocardiographic signal sources. Besides that, measurement and processing tools were also required for subsequent evaluation of performance.

3.2.1 ECG signal sources

As this project aims to build a medical instrumentation device for small animal monitoring, the input of the system is a bioelectrical signal. However, even if this entails dealing with living organisms to extract such biopotential, during the prototype development phase it was not possible to use small animals on a daily basis. A solution is the use of an ECG waveform simulator that enables the improvement of the device until translation into pre-clinical experimentation is feasible.

3.2.1.1 ProSim 8 ECG simulator

Pro Sim 8 vital signs simulator by Fluke Biomedical is an eight in one multifunction patient simulator. In fact it combines an ECG, a fetal, an arrhythmia, a respiration, a temperature, a cardiac output, cardiac catheterization and a non-invasive blood pressure (NIBP) simulator. It has also the functionality of a SpO2 tester.

As it can be observed in Figure 28, a normal sinus rhythm (NSR) waveform can be generated from a 12-lead independent outputs configuration referenced as: RA (Right arm), LL (Left Leg), LA (Left Arm), RL (Right Leg) and V1 to V6.

Different parameters such as waveform group, heart rate, axis or amplitude can be defined. It is worth to highlight that the range of heart rate is limited from 10 bpm to 360 bpm in 1 bpm steps, which is suitable for rat heart rate but is insufficient for mice cardiac frequency which is greater than 500 bpm. Furthermore, noise artefacts of different type can be introduced at each independent lead with different degrees of distortion 25%, 50% or 100%. The noise types available are: 50 Hz, 60 Hz, muscular, baseline wander and respiration.

Figure 28. ProSim 8 vital signs simulator

Additionally this simulator offers the possibility of inducing several heart failure events such as arrhythmia, tachycardia, bradycardia, ventricular fibrillation and asystole. It is an accurate source of ECG signal for device testing previous to animal monitoring.

3.2.1.2 Small Animals: rats and mices

Once the ECG device showed to work perfectly with the ECG simulator with the stability necessary to satisfy all the safety requirements, it was considered apt for testing in living rodents.

Figure 29 shows a rat and a mouse used in the Laboratory of Experimental Surgery at Hospital Gregorio Marañón for testing the device performance under technician specialists' supervision.

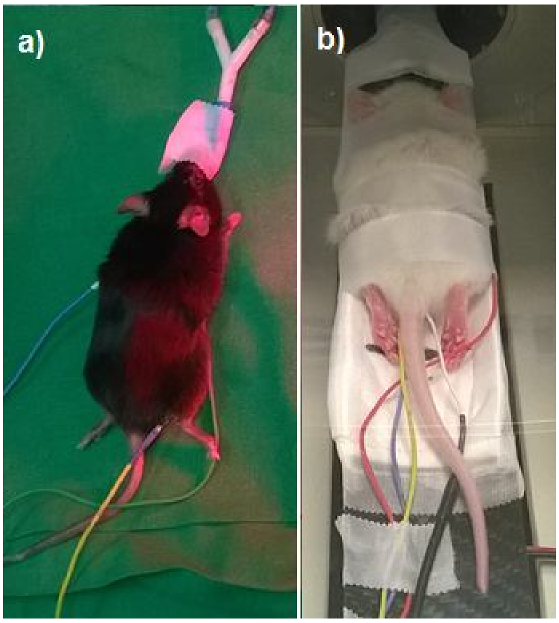


Figure 29. a) Mouse b) Rat

3.2.2 Measurement instruments and software tools

First of all, to run the device, programming of the Arduino Mega 2560 is needed. The Arduino open source software version 1.5.6r-2 was used to carry out this task. This integrated development environment (IDE) is written in Java and is based on *Processing* programming language that includes support for C and C++. The code is compiled and uploaded to the microcontroller Arduino board that executes it.

Even if the USB cable that connects the Arduino to the computer supplies power to the board to run, an external power supply source instrument was also used in some cases.

For the evaluation of the correct functioning of the device, some measurement instruments and software tools have been used to assess the validity of the results.

On one hand, to visualize the analog output TTL pulse signals from the BNC cable an oscilloscope was used. On the other hand, for analysis of digital data signals accessed through the serial monitor of the Arduino environment, the processing software Matlab, by MathWorks, was used.

4 Methods

Methods implemented in this project are thoroughly described in this section. The development of this ECG prototype can be subdivided in two tasks. First, the hardware implementation which mainly consist in the construction of the device itself. Second, software implementation which plays a crucial role in conferring the principal functionalities to the ECG monitor.

4.1 Hardware implementation

After an exhaustive technical description in section 3 of the main components: Arduino Mega 2560, ADS1298, Gameduino2 and temperature sensor; an assembling process is needed to make all the parts work together as a complete set . In order to define the suitable connections between the different components an understanding of the SPI protocol that allows communication between the modules is required.

4.1.1 SPI communication

4.1.1.1 SPI protocol description

The Serial Peripheral Interface (SPI) bus is a communication standard used for short data transmission in embedded systems of electronic devices.

In this protocol the different modules that communicate in a serial synchronous manner are classified in two categories: master or slave. The principal characteristic is that there can be many slaves but only one master which is in charge of controlling the communications between the different elements.

This protocol implements a four-wire serial bus including a clock line, a data input line, data output line and a line for slave selection [34]. The following four logic respective signals can be transmitted:

- SCLK: Serial clock. It is the synchronization pulse setting the pace of bit transmission. Each clock pulse sends or receives a bit.
- MOSI: Master Output Slave Input. Data output from master is the input for the slave.
- MISO: Master Input Slave Output. Data output from slave is the input for the master.
- SS: Slave select. Signal for selecting a single slave at a time to initiate communication. All the slaves that are not selected are unable neither to receive data through MOSI wire nor to send data through the MISO wire.

The block diagram in Figure 30 shows the implementation of the SPI protocol.

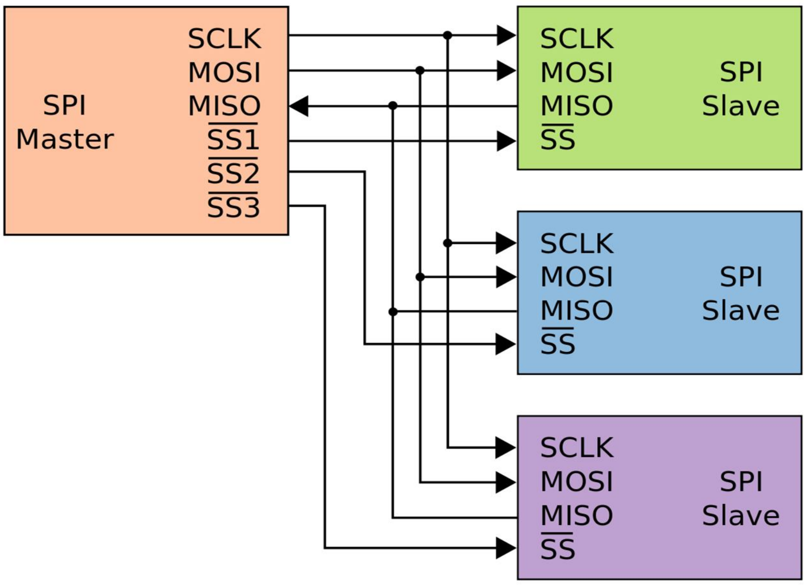


Figure 30. SPI communication protocol block diagram [35].

It is worth to point out that MISO, MOSI and SCLK lines are shared among all the slaves since they carry common data valid for them all. However, SS line connects independently the master with every slave for individual slave selection. As a consequence only one slave can be active at a time, in such a way it is not possible to establish communication with several slaves simultaneously.

Another important characteristic to focus at is the synchronous character of this protocol. Indeed when communication is started, clock frequency configuration is the first task performed by the master. Once the frequency is appropriate for data transmission, the master selects the slave with a logic level 0 or LOW on the SS line.

Then during clock cycles a duplex data exchange occurs as Figure 31 illustrates. The master sends a bit on the MOSI line and the slave reads it, while the slave sends a bit on the MISO line and the master reads it. Words of 8 bits are usually transmitted. Once the communication is over, the master deselects the slave.

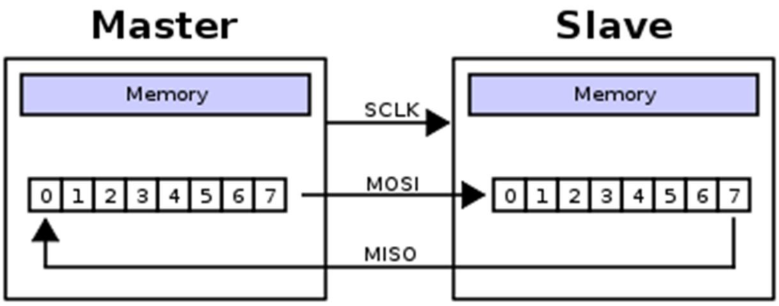


Figure 31. Data transmission in SPI communication [35].

The clock has different configurations that offer the possibility of setting four modes of data transmission by changing the clock polarity and clock phase. This options are called CPOL and CPHA respectively. The polarity indicates if the reference level is 0 (LOW) or 1 (HIGH). The phase indicates whether bits are sent in RISING or FALLING edge by sampling in the first (CPHA=0) or second clock cycle (CPHA=1). The four different configurations shown in the timing diagram of Figure 32 are:

- CPOL=0/CPHA=0 → Reference value is 0. Data is captured on clock's rising edge and data is propagated on a falling edge.
- CPOL=0/CPHA=1 → Reference value is 0. Data is captured on clock's falling edge and data is propagated on a rising edge.
- CPOL=1/CPHA=0 → Reference value is 1. Data is captured on clock's falling edge and data is propagated on a rising edge.
- CPOL=1/ CPHA=1→ Reference value is 1. Data is captured on clock's rising edge and data is propagated on a falling edge.

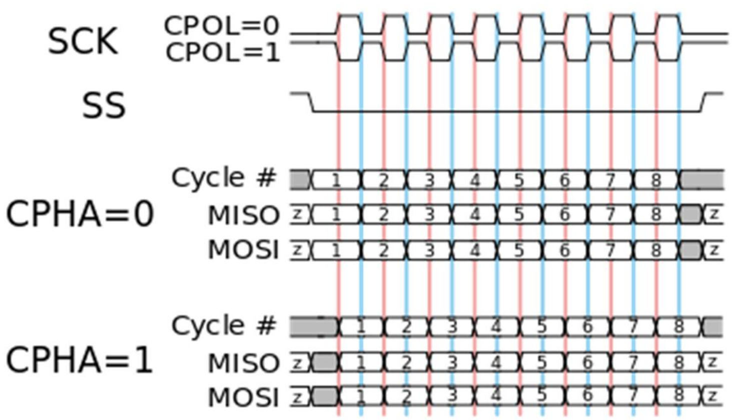


Figure 32. Timing diagram showing clock polarity and phase [35].

4.1.1.2 SPI implementation in our device

The small-size ECG device relies on SPI communication to transmit information between the master, which is the microcontroller Arduino Mega 2560, and the peripherals or slaves which are the ADS1298 and Gameduino2. It is important to note that the temperature sensor does not work as a slave since it does not need to implement the SPI protocol to establish communication with the Arduino microcontroller. It acts as a simple sensor which output value can be read at any moment no matter which slave is selected.

In this manner the functions carried out by each unit complement each other and can be executed in an ordered and synchronous way (as it will be exhaustively explained in the following software implementation section through an overview of the programming code). But first let's study the overall data transmission operation process through SPI protocol. It is worth to highlight that the clock polarity is the same for both slaves, taking as reference value 0.

The SPI protocol starts when the master, Arduino Mega 2560, selects the ADS1298 by activating the corresponding SS pin and sets the appropriate clock configurations at a frequency of 2MHz. The SPI mode implemented is mode 1 corresponding to CPOL=0 and CPHA=0. This means that data is captured on clock's rising edge and data is propagated on a falling edge. The analog-to-digital converter ADS1298 can then carry out its main functions which are data acquisition and conversion. First, analog data recorded by the 3 lead electrodes is inputted through the different channels of the DB 15 connector. Second, after multiplexing and basic filtering, signals in each channel are digitalized. The converted data is transferred through the MISO line to the Arduino microcontroller in words of bits.

The Arduino saves the data for every channel in separate vectors and process it. Communication with the ADS1298 finishes by disabling the SS pin. Temperature readings from AD22100 sensor can be inputted and TTL pulse activation turned on. All the processed data can be accessed through the serial monitor of the Arduino environment for further data analysis if needed. However, data visualization is offered to the user through the graphic display on the Gameduino2 screen.

Arduino sets the corresponding clock frequency configuration at a rate of 8 MHz to start transferring data to the Gameduino2 through the MOSI line. The SPI mode implemented here is mode 0 corresponding to CPOL=0 and CPHA=1. This means that data is captured on clock's falling edge and propagated on rising edge. Gameduino2 works as the interface with the user since its main function consist on data representation. It displays the ECG signal on the screen as well as other parameters calculated by the Arduino such as the heart rate, the TTL pulse or the temperature readings.

Once a complete data buffer has been transmitted, Arduino ends the communication with Gameduino2 by disabling the SS pin. The whole process starts again and is repeated from the beginning. The block diagram explaining the SPI protocol of the ECG device can be observed in Figure 33.

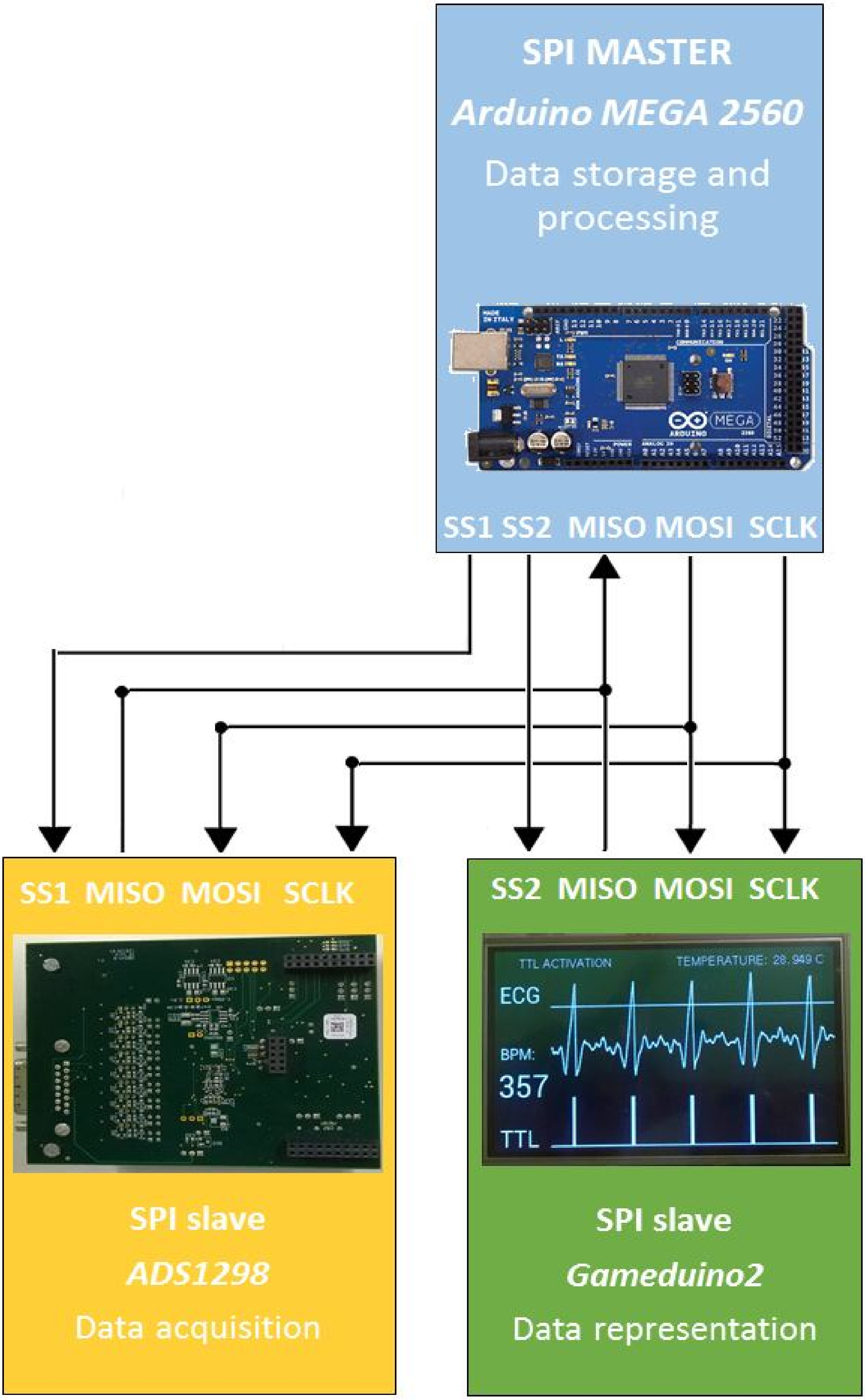


Figure 33. Block diagram representing the SPI protocol of ECG device

4.1.2 Connections and assembly

With the aim of establishing this SPI communication protocol and making possible all the modules to work together as one, some initial connections have to be made. In this part, the connections between the master Arduino and all the other three main components are detailed. In order to support all the connections and confer robustness to the prototype, an adapter proto-board was designed. The whole assembly contained in the plastic enclosure with all its complements is then presented as the final device.

4.1.2.1 ADS1298 – Arduino

Table 3. Pin mapping Arduino Mega 2560- ADS1298

Arduino Mega 2560	ADS1298
20	RESET
21	CS
3	DRDY
GND	GND
MOSI	DIN
SCLK	SCK
VCC	5V
MISO	DOUT
3.3V (POWER)	3.3V

4.1.2.2 Gameduino 2- Arduino

Table 4. Pin mapping Arduino Mega 2560-Gameduino2

Arduino Mega 2560	Gameduino2
52	SCK
50	MISO
51	MOSI
9	SD select
8	GD select
2	Interrupt
5V	5V
GND	GND
Vin	Vin
A0	Accelerometer X
A1	Accelerometer Y
A2	Accelerometer Z

4.1.2.3 Adapter proto-board

An adapter proto-board with copper pads was designed as it is detailed in Appendix 2, by soldering cables and pins that allow connecting the Gameduino2 and ADS1298 to the Arduino Mega 2560 microcontroller.

4.1.2.4 Temperature sensor - Arduino

For the sake of miniaturization and portability of the small-size ECG monitor, the implementation circuit has been integrated in the adapter proto-board using micro resistors and capacitors as shown in Figure 34. Moreover in order to adapt it as a rectal probe for its application of measuring body temperature of small animals, the AD22100 has been soldered at the end of an elongated twisted pair cable with copper core acting as ground as shown in Figure 35. To counteract any biocompatibility issues silicone has to cover the tip of the sensor in order to be introduced into the animal's rectum. A MOLEX adaptor serves as the union point with the Arduino circuit allowing an easy removal of the sensor.

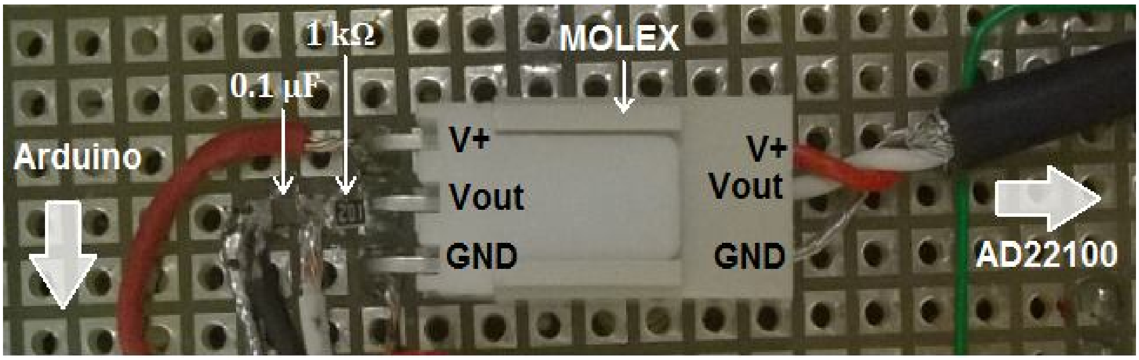


Figure 34. AD22100 sensor circuit integration.

The output of the sensor is connected to the analog pin A3 of the Arduino which will transform the measured voltage in a digital value that will be displayed on the screen of the Gameduino2. The other two power connections join pins 5V and GND of the microcontroller.

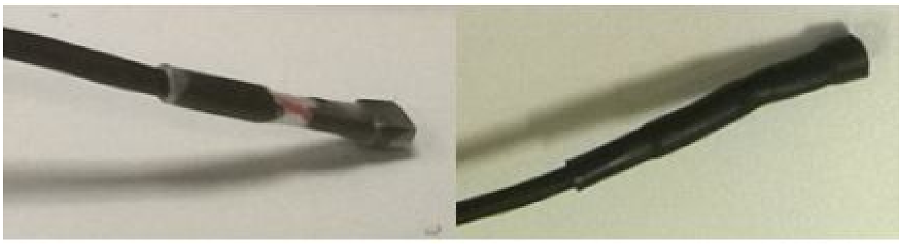


Figure 35. AD22100 sensor at the end of an elongated twisted pair cable as a rectal probe.

4.1.2.5 Final assembly

The final structure of the device presented in Figure 36 shows the "sandwich-like" configuration of the device where the Arduino and ADS1298 are fixed below and the Gameduino2 above the adapter proto-board by mean of male pins. The relevant pins are connected together following the pin mapping of Tables 3 and 4. Pins are linked by unipolar thin cables on the top of the proto-board by soldering.

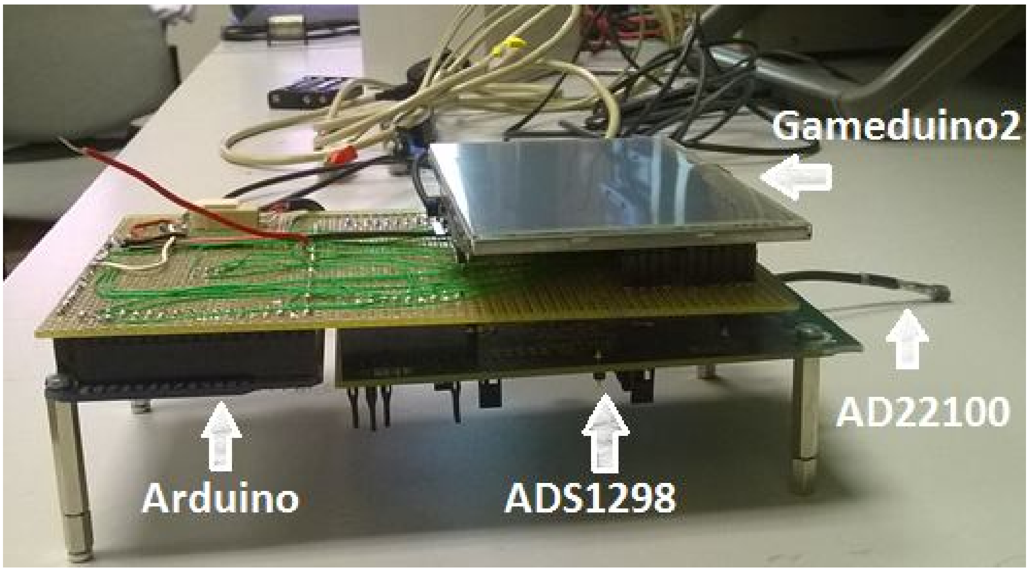


Figure 36. Lateral view of final structure of prototype.

The proto-board also supports the molex connector of the temperature sensor AD22100 which is accordingly connected to the Arduino Mega 2560 pins. Similarly, the coaxial cable carrying the TTL pulse is soldered on one side to pin 4 (and ground) of the Arduino and at the other end to the BNC external connector embedded at the surface of the plastic enclosure. The battery of 6V is put inside the plastic box and connected to both the Arduino and power switch as it can be observed in Figure 37.

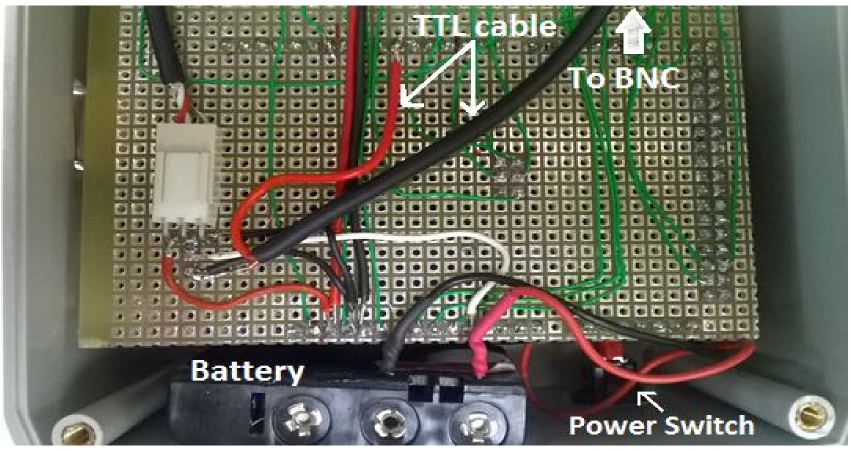


Figure 37. Bottom part of adapter proto-board showing TTL cable, temperature sensor and battery connections.

The assembly is fitted inside the plastic enclosure as shown in Figure 38a to give the final small-size ECG monitor shown in Figure 38b when the aluminium cover is screwed on. As it can be observed the temperature rectal probe exits the front of the plastic enclosure just under the BNC connector that outputs the TTL pulse.

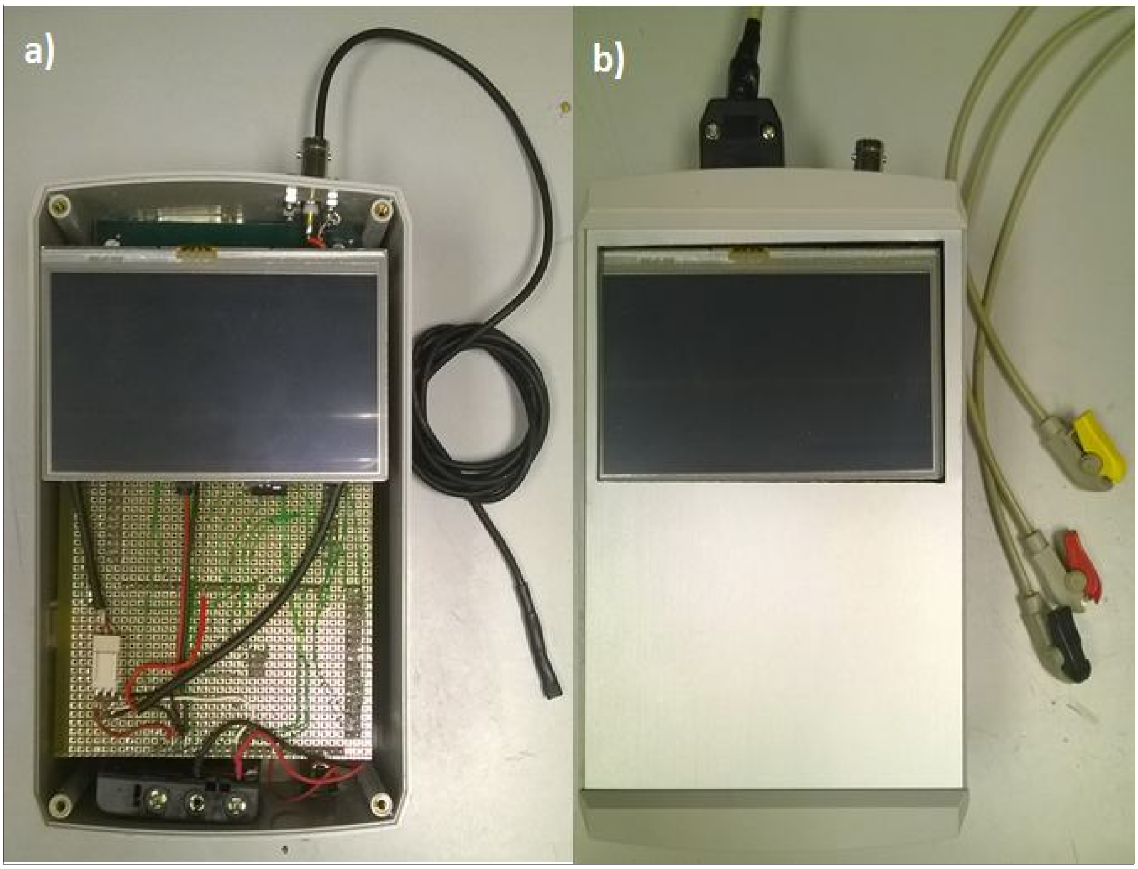


Figure 38. Top view of final assembly enclosed in the plastic box. a) Without aluminium cover. b) With cover and electrodes connected to the DB15 input

4.2 Software implementation

Once the hardware implementation is done, the device is ready to be programmed. The code is written in the Arduino software environment and uploaded to the board that will execute the necessary commands for transferring data from the ADS1298 to the Gameduino2. This section will go in detail into the programming code to not only present how the SPI communication is applied digitally but to focus on some important processing functions employed corresponding to the detection of R waves for heart rate calculation, TTL pulse activation and temperature readings.

Although comprehensive explanations are given in the next subsections, as well as intuitive block diagrams that translate the programming language into concepts for an easier understanding, the programming code executing the device is enclosed in Appendix 3 section for further study.

4.2.1 Overview of programming code

The programming code can be subdivided in three main parts as programming in Arduino usually dictates: variables initialization, setup and main loop.

4.2.1.1 Initialization of variables

All the variables that are going to be used have to be declared at the beginning. Not only the data type has to be specified but also the length or dimensions in case of vectors or matrices. In this part the numbers of the different pins are assigned, the number of samples composing each buffer of stored data is defined, as well as the counters are initialized.

The libraries needed for the code execution have to be included. This is the case of SPI, EEPROM (Electrically Erasable Programmable Read-Only Memory) and Gameduino2 libraries. The latter is of significant relevance since it will provide all the commands that govern the FT800 controller of Gameduino2 and therefore controls simultaneously the displays on the screen.

4.2.1.2 Setup

After variable initialization, some more information about the mode and initial state of the pins is required to be inputted for the Arduino to work properly. In other words, pins have to be specified as INPUT or OUTPUT ones as well as they have to be enabled or disabled before the main loop gets started. To illustrate it is of crucial importance to de-select the slave select (SS) pins of both slaves ADS1298 and Gameduino2, to avoid interference with Arduino initialization.

During the setup, the beginning of communications with Gameduino2 and through the serial port has to be trigger. A small delay must be introduced previously to give Gameduino2 enough time to boot.

4.2.1.3 Principal loop

The main loop is the core of the programming code. As its name indicates, this portion of code is repeated over and over until the reset button is pressed or the power supply is switched off

It is subdivided in different functions which carry out the principal tasks of data reading by the ADS1298; data management, storage and processing by the Arduino; and final data representation by the Gameduino2. The output of the TTL pulse is also activated by the Arduino but after four loops once the thresholds are calibrated.

First, the ADS1298 is selected as slave and SPI communication begins. It is important to note that once the first time this cross-talk between the ADC and the microcontroller is enabled, the ADS1298 is reset to avoid any booting problems. Similarly, the ADS1298 registers are also configured only once. After that, the device is ready to function in *high resolution mode* at a data rate of *500 SPS* (samples per second). The gain of the channels of interest which are channels 1, 2 and 3 is set to the maximum value of *12* for maximal amplification of the low amplitude bio signal, whereas the rest of the channels are power down to diminish power consumption and sampling time. The ADC starts acquiring data in a continuous mode.

Second, the Arduino microcontroller generates an interruption to read one sample of each channel transferred through SPI communication by the ADS1298 when data conversion is ready. Since the sampling frequency was defined to be 500Hz, each interruption transferring a single sample lasts 2ms. The successive samples read by the Arduino are stored in the SRAM memory in two separate vectors, one for data of channel 2 (right arm) and the other for data of channel 3 (left arm). This reading interruption is repeated as many times as total number of samples defined in the initialization, so that the channels data vectors are filled.

In a third step, the Arduino microcontroller processes the stored data. But first, after each interruption, the values read from both channels are subtracted to generate a third vector containing the data that corresponds to the ECG signal of lead I from Einthoven's triangle. Each data sample of the resulting signal is evaluated to decide whether to activate the TTL pulse, as explained in section 4.2.3. This data buffer is then analyzed by Arduino to extract cardiac features such as R wave detection and heart rate as described in section 4.2.2.

Finally, Arduino selects Gameduino2 for data representation. Gameduino2 library allows to display the ECG data points joined together as a curve as well as the TTL pulse that shows the detection of R waves just below. It is worth to highlight that data scaling is performed by Arduino in order to constrain the ECG graph in a specific area of 400x36 pixels. Additionally, the heart rate value is presented in BPM. During Gameduino2 slave selection, Arduino retrieves the output voltage value of the sensor AD22100 and transform it to a temperature value as it would be explained in section 4.2.4. This offers the option of directly displaying the temperature value on the screen of the Gameduino2.

Once the communication with Gameduino2 ends, the loop starts again and the whole process is repeated for a real-time continuous monitoring and image acquisition triggering. Each time the loop re-starts the data vectors and counters are reset to zero, so previous data buffers are erased and new data can be stored. The duration of a single loop is proportional to the number of samples multiplied by single sample conversion duration. To illustrate for rat monitoring, the number of samples chosen 400, entails a loop duration of roughly 800ms.

A schematic summarizing the different steps of the main loop code is shown in the following Figure 39.

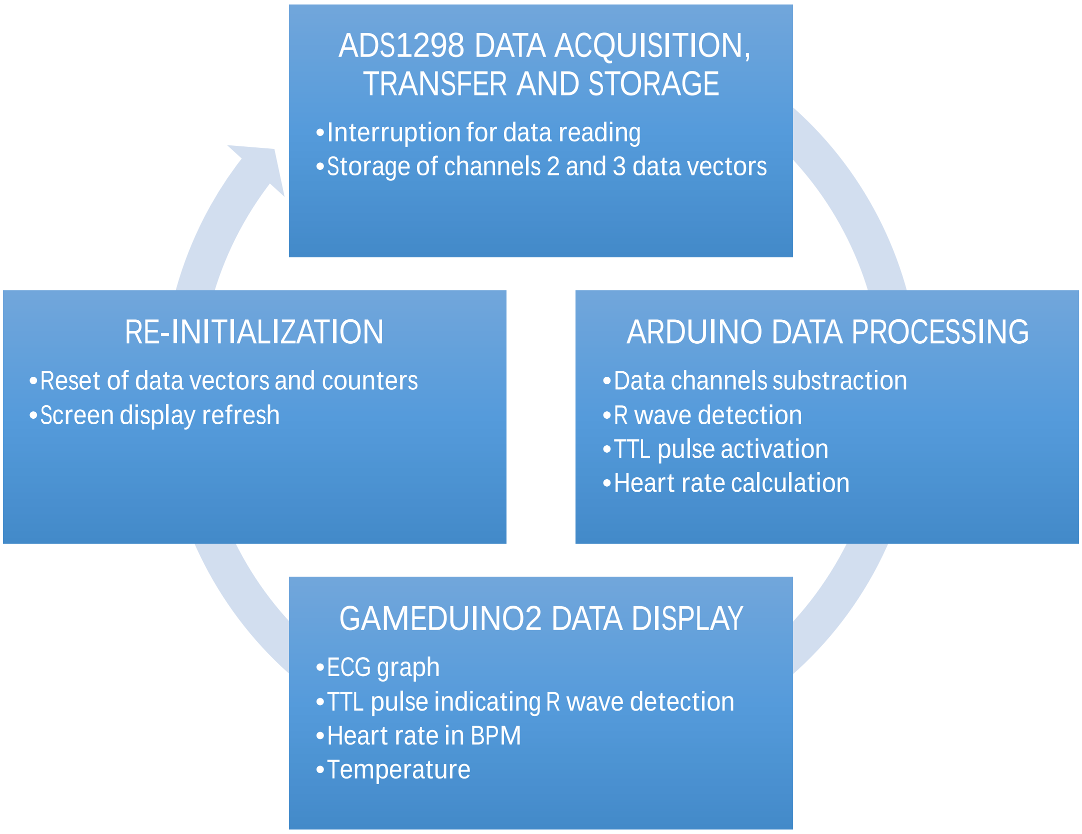


Figure 39. Diagram of the main loop of the device program

4.2.2 Heart rate calculation

For the purpose of showing to the user the cardiac frequency of the small animal that is being monitored, the Arduino Mega2560 microcontroller has to process the ECG signal data.

First thing to do is to detect when QRS complexes occur. A varying threshold is established each time a data buffer is ready for representation according to the oldest and most famous "Real-Time QRS Detection Algorithm" [36]. Minimum and maximum values of data are extracted for that period in order to delimit the dynamic range. Then, the threshold takes the value corresponding to the 70% percent of the dynamic range which will ensure that only R wave data points can exceed.

Once the threshold is established, the signal is evaluated sample by sample to estimate whether it is above or below the threshold and therefore whether an R wave is respectively occurring or not. Each time the threshold is surpassed in every new cardiac cycle, the position is saved in an auxiliary vector. When all the samples in the buffer have been analysed the successive RR interval lengths are estimated by subtracting every two successive positions of R wave detection. The average of the RR intervals is calculated and multiplied by the sampling time of 2ms to get the period of one sample.

Finally the pulse in BPM is given by the following equation:

$$Heart\ Rate\ (BPM) = \frac{60}{RR\ interval\ in\ secs} = \frac{60}{period}$$

4.2.3 TTL pulse activation

The TTL pulse is generated in real time by the Arduino microcontroller when an R wave is detected signalling the start of an R-R interval. As a remainder of section 2.3 electrocardiographic gated imaging, this pulse will mark the starting point of image acquisition along one cardiac cycle divided in eight frames.

The way this is accomplished in the code is by evaluating whether each sample obtained just after the interruption and subtraction of channels 2 and 3, is above or below the threshold value calculated in the previous loop.

If the ECG signal is greater than the threshold, the TTL pin of the microcontroller is set HIGH and remains in that state until a sample does not exceed the threshold. Thus a 5V pulse of roughly 10ms goes through the BNC cable and reaches the imaging machine that decides at which moment of the cardiac cycle the image is acquired.

4.2.4 Temperature reading

In order to display on the screen of Gameduino2 the temperature sensed by the AD22100 at the end of the rectal probe, the following transfer function relating the output voltage VOUT and the unknown temperature TA has to be applied:

$$VOUT = \frac{V+}{5\,V} \times (1.375\,V + 22.5\,mV/°C \times TA)$$

V+ : corresponds to the power supplied by the microcontroller which is 5V.
22.5mV/°C : corresponds to the sensitivity of the sensor.
1.375 V : corresponds to the voltage for 0ºC.

Arduino measures the voltage in analog pin 3 taking as reference from ground to 5 volts with 10 bits of resolution which corresponds to integer values between 0 and 1023. Consequently the measurement in units has to be multiplied by $\frac{5V}{1024\ units}$ or equivalently by 4.9 mV/unit.

Therefore by simple rearrangement of the equation above, the temperature can be presented in the screen using the relation:

$$TA = \frac{\left(measurement \times \frac{5V}{1024} - 1.375\,V\right)}{22.5\,mV/°C}$$

5 Results of ECG device evaluation

After complete hardware and software implementation, the small-size ECG monitor was ready for evaluation. For that, it was primarily tested at the laboratory with an ECG waveform simulator. Further trials were performed with small animals at Hospital General Universitario Gregorio Marañón.

5.1 Tests with ECG simulator

As a first approach to evaluate the small-size ECG prototype, the ProSim8 waveform simulator by Fluke was used. The performance of every functionality of the device was assessed and the results are presented in the following subsections.

5.1.1 ECG monitor performance

Initially, data acquired by the ADS1298 and read by the Arduino was plotted in Matlab as a reference for later verification of correct ECG signal representation on the screen of the Gameduino2.

In Figure 40, four cardiac cycles, of duration 0.2 seconds in average each, are presented in two plots in a time window of 0.8 seconds, which is the same one chosen for the Gameduino2 display. The signal shown in both plots is the same but the one on the left allows localization of individual samples while the one on the right just shows the ECG curve with all the points joined together as it will be observed in Gameduino2.

It is interesting to include the graph on the left in order to assess whether the sampling rate of 500 SPS is appropriate for the simulated heart rate of 360 BPM. Apparently, it seems to be suitable for the overall signal although it is true that a higher sampling rate would also be desirable to increase the resolution of the QRS complex. However this idea was discarded as the number of samples for visualization of the same time window exceeded the Arduino's SRAM capability.

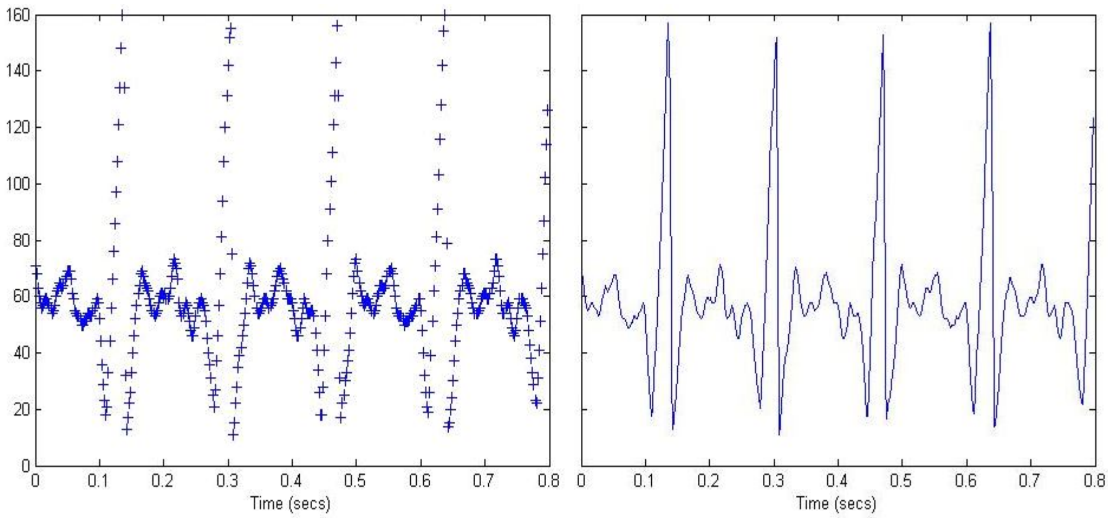


Figure 40. Graphical representation of a 360 BPM ECG signal in Matlab.

The ECG monitor was then tested in an ideal case of noise absence with the complete setup as observed in Figure 41. As expected, in the zoomed image of the screen it can be seen how the ECG graph representation on the screen is a perfect ECG signal. The R waves are correctly detected with the TTL pulse and the heart rate value of 357bpm is almost equal to the simulated value of 360 bpm.

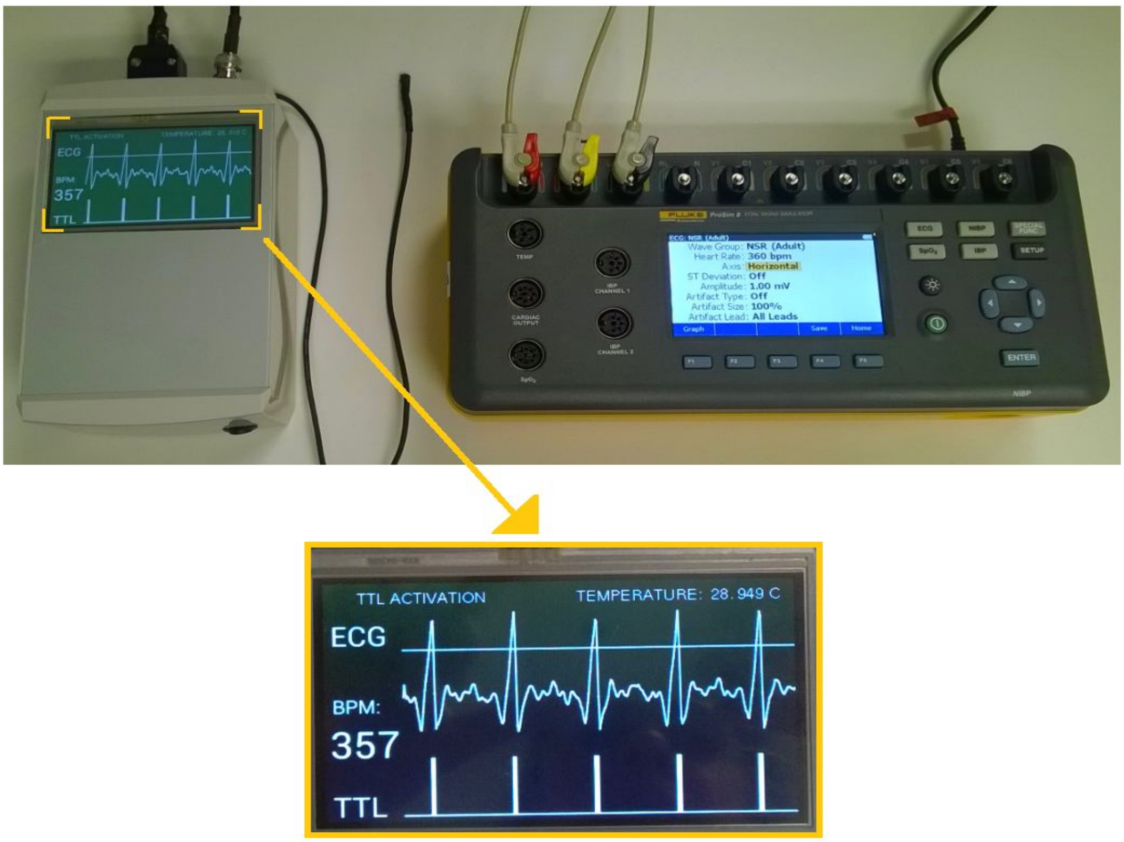


Figure 41. ECG monitor evaluation performance for ideal signal

5.1.2 TTL activation

TTL pulse correct activation was assessed by connecting the ECG device to the oscilloscope due to the absence of imaging machine in the laboratory. TTL periodic pulses of 5 volts were visualized on the oscilloscope screen as shown in Figure 42. The distance between them perfectly corresponds to the RR-interval duration of roughly 200ms as measured in Matlab graphs of previous section.

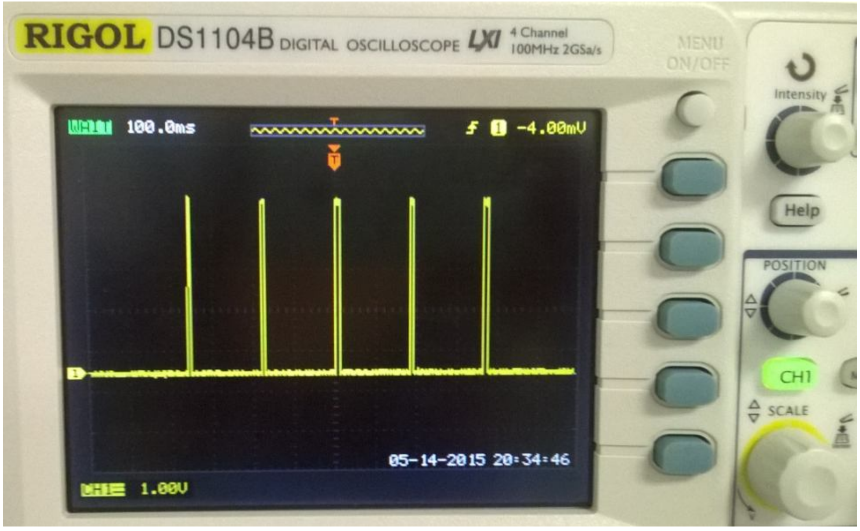


Figure 42. TTL pulse activation visualization on oscilloscope

Although the TTL pulse is activated precisely most of the time in a continuous manner, some errors are stochastically introduced as observed in Figure 43. This might be a major problem for image reconstruction if it occurs repeatedly as it is further described in next sections.

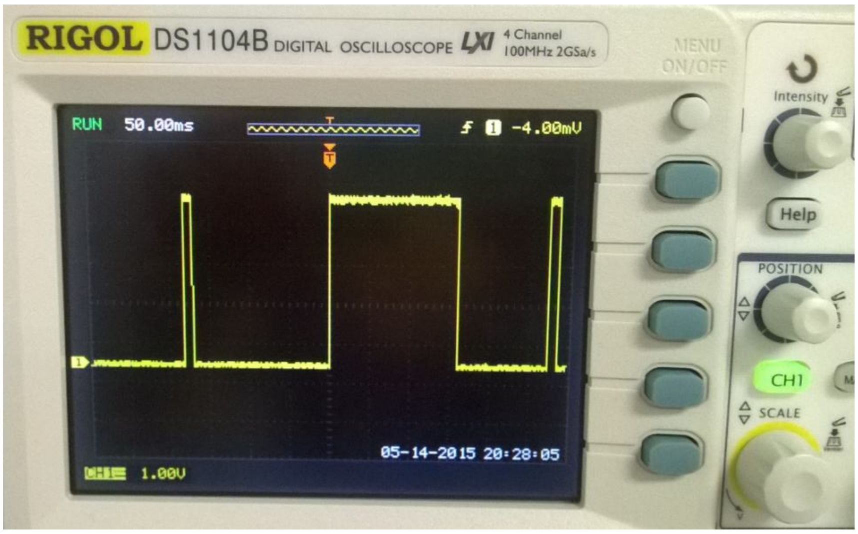


Figure 43. Random TTL pulse activation failure.

5.1.3 Temperature measurements

During testing of the device in the laboratory the temperature value displayed on the screen of Gameduino2 corresponded to the ambient temperature, which was more or less stable with slight variations across the days depending on the heat conditions. In order to assess if the temperature sensor was effectively sensitive to changes in temperature a performance test of sudden heating and cooling was carried out to evaluate if the AD22100 responded accordingly.

For that, the end of the rectal probe was hold in a close hand to increment the temperature during 20 seconds and then was released for cooling at ambient temperature. The temperature fluctuations were followed up on the screen display but also printed in the serial monitor of the Arduino environment for later data analysis in Matlab as shown in Figure 44 below.

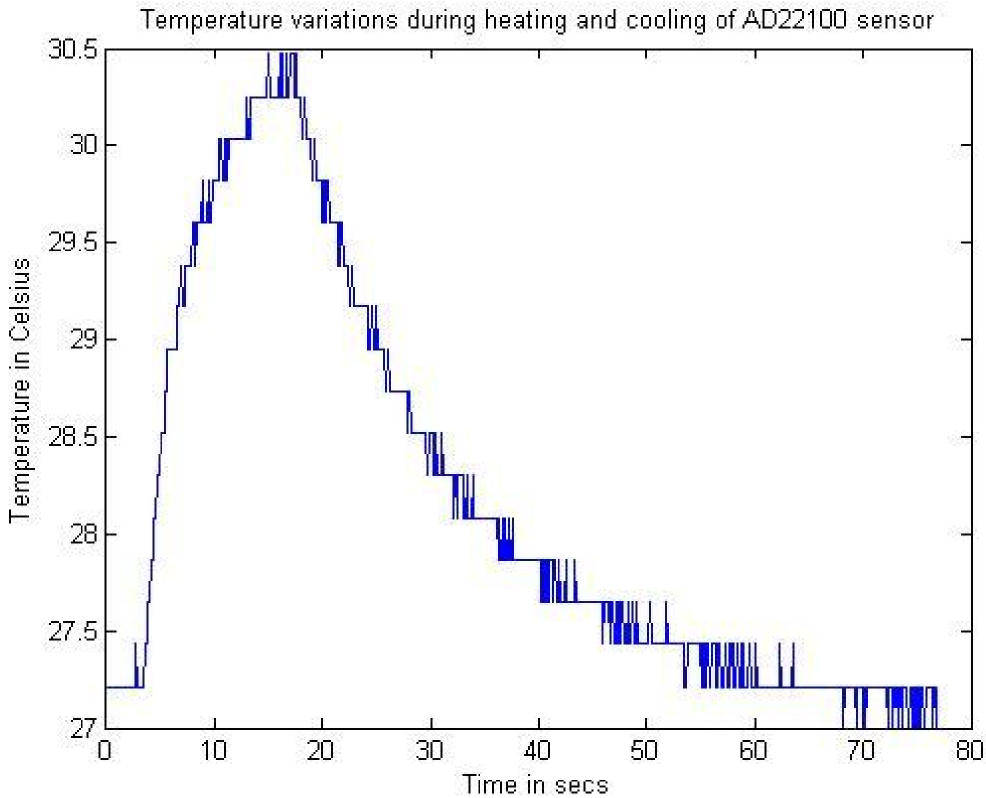


Figure 44. Graph of temperature variations during sudden temperature variation

The graph, where temperature is plotted versus time, shows a fast and almost linear rise of temperature during heating from 27.25ºC to the peak of 30.5ºC at a rate of 0.1625 ºC/s. In the cooling phase an exponential slow decay of temperature can be observed. It takes 40 seconds, the double of the heating time, to fall back to the initial temperature. The ladder shape decay indicates that the temperature is stabilized by steps. Nevertheless these values can be considered acceptable for our purpose. The temperature sensor AD22100 shows good performance.

5.1.4 Noise performance

As explained in previous sections, the ADS1298 contains basic filters which deal just with very high frequency noise caused by electromagnetic interferences.

It is suspected that these filters might not be sufficient, but it is unknown how critical this would be for the performance of the small-size ECG monitor. With the aim of elucidating whether this could be a possible problem, a noise test was performed making use of the different noise options of the Prosim8 waveform simulator by Fluke.

5.1.4.1 Power line interference: 60Hz noise

The 60 Hz noise introduced with an artefact size of 100% with the ProSim8 simulator greatly distorts the ECG signal as shown in Figure 45. However, even if the ECG signal cannot be recognized anymore, the heart rate does not seem to be very affected since this type of noise does not disturb significantly the R wave which is accurately detected.

Figure 45. Noise performance: Power line interference (60Hz) at 100%

However, as shown in Figure 46, the power line interference noise can become a serious problem when the batteries are used and do not supply enough voltage for the whole system to work properly. In fact, the ADS1298 probably fails in the data acquisition process. The introduction of inaccurate data points can be crucial for the calculation of the heart rate which significantly deviates from the expected value of 360 BPM due to an inaccurate R wave detection. Image acquisition would not be properly triggered which is unacceptable. The implementation of a notch filter would be needed to solve this problem.

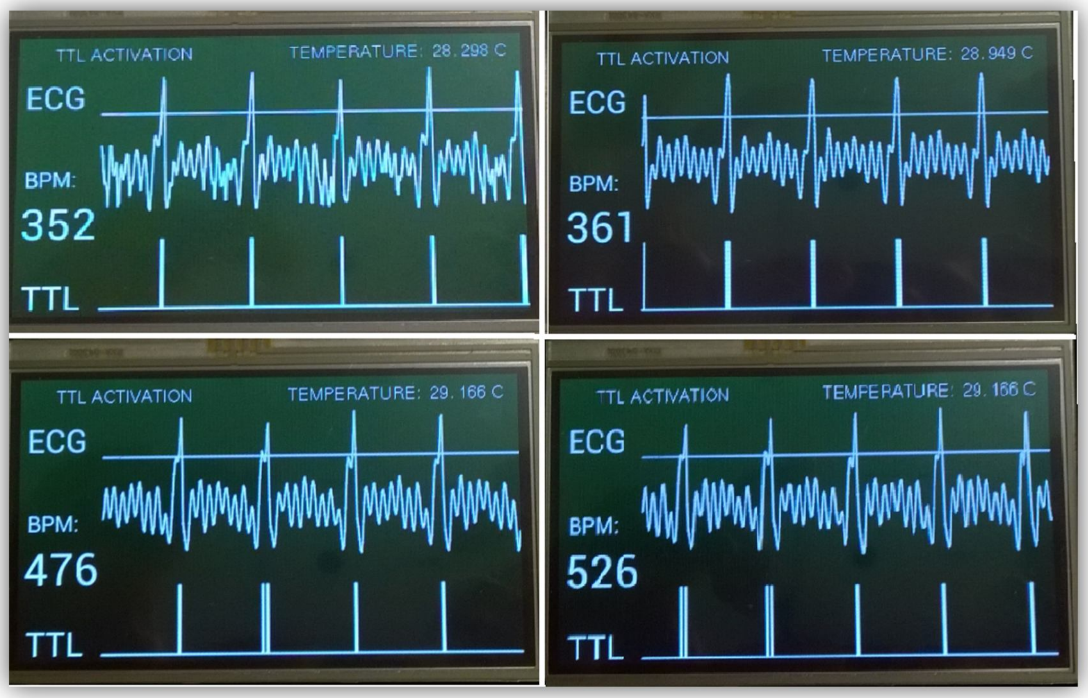


Figure 46. Power line interference (60Hz) at 100% comparison with new batteries (top images) and used batteries (bottom images).

5.1.4.2 Baseline wander

When baseline wander noise is introduced at 100%, the ECG device works perfectly similar to an ideal signal as it can be observed in Figure 47. The noise does not affect the detection of QRS complexes as illustrated by the perfect buffer of TTL pulses, neither the heart rate value.

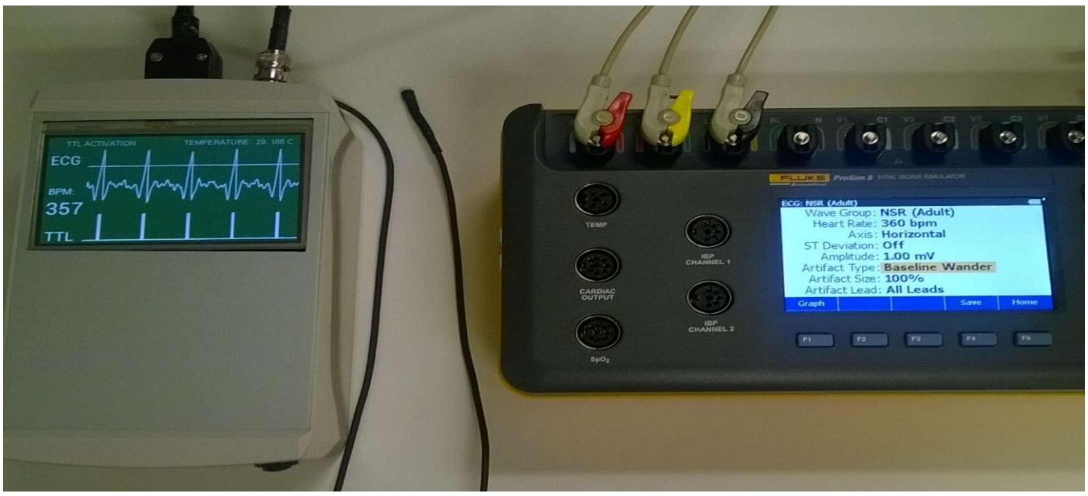


Figure 47. Noise performance: Baseline wander at 100%

5.1.4.3 Respiration

The respiration noise at 100%, as the baseline wander, does not seem to disturb the overall functioning of the device. The R waves are properly spotted despite the little winding of the ECG signal as illustrated in Figure 48. In a similar manner, the heart rate is correct. Therefore, respiration would not be a big issue.

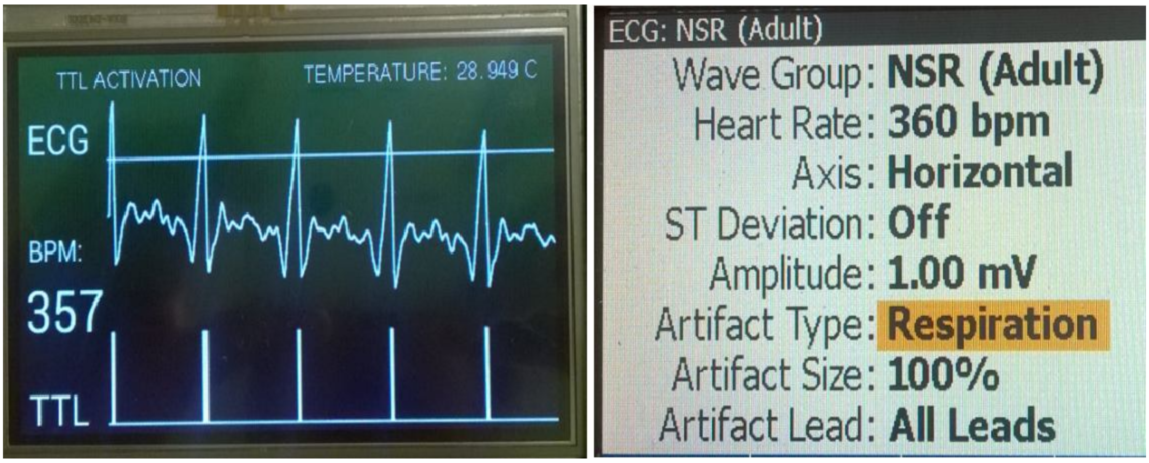


Figure 48. Noise performance: Respiration noise at 100%

5.1.4.4 Muscular noise

The muscular noise with an artefact size of 100% slightly distorts the morphology of the ECG signal displayed on the screen of Figure 49. However the rest of functionalities performed by the ECG monitor which represented in the screen are as accurate as if it was an ideal signal. This noise would not cause any major problems.

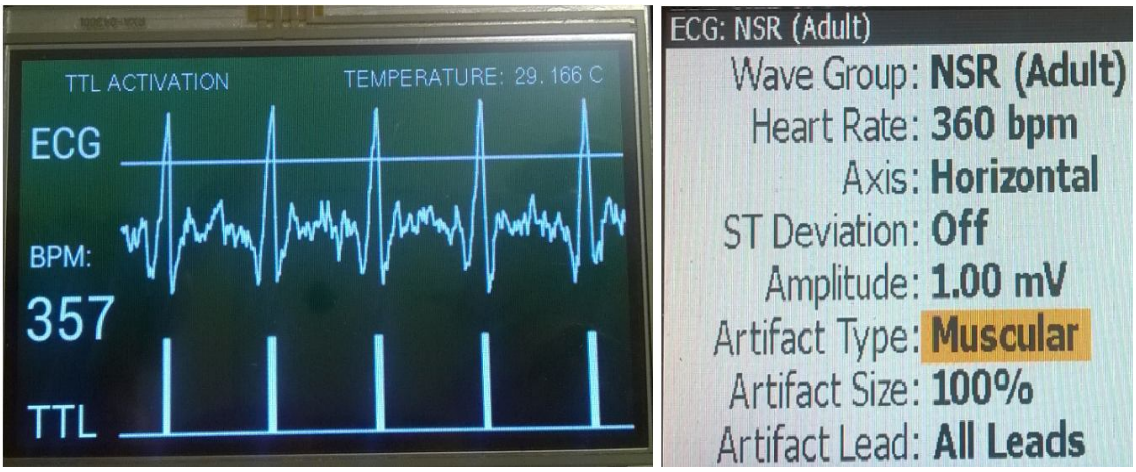


Figure 49. Noise performance: Muscular noise at 100%

5.2 Tests with small animals

The results of the ECG monitor evaluation with the simulator at the laboratory were very satisfactory. The small-size ECG device showed to work as expected with a high degree of accuracy even in the presence of a great quantity of different types of noise. Consequently, the device was considered to be ready for testing in small animals in order to validate its applicability for pre-clinical gated imaging.

The testing of the ECG monitor with rodents was held in the "Laboratorio de Cirugía Experimental" at Hospital General Universitario Gregorio Marañón. All experimental procedures that involve animal manipulation were performed in compliance with the European Communities Council Directive 2010/63/EU and approved by the Institutional Animal Care and Use Committee of the Hospital, supervised by the Comunidad de Madrid according to the Annex X of the RD 53/2013.

A male rat weighting 410 grams was used to carry out the experiments. Laboratory technicians assisted in the manipulation of the small animal especially in the tasks of sedation and placement of the needle electrodes. The supervisor Juan José Vaquero supervised the whole experiment as well.

The assessment of the ECG device for vital signs monitoring was performed. As it can be seen in Figure 50, the needle electrodes and the rectal temperature sensor were introduced in the animal to track in real time the electrocardiogram and body temperature respectively on the device on the right.

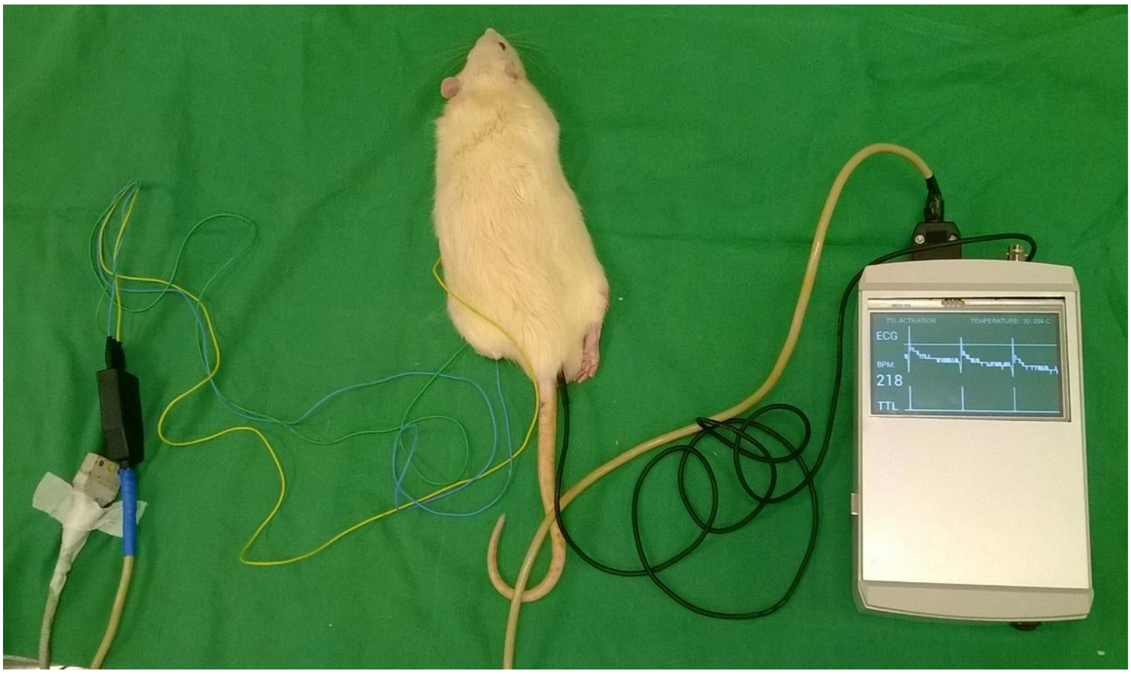


Figure 50. ECG monitor evaluation in male rat

The zoomed in image of the screen, shows in Figure 51, a heart rate of 218 bpm, a temperature of 30.0ºC and the electrocardiogram of the rat at the same moment. It is important to highlight that the body temperature measured is a very low value due to the fact that the rectal probe was too big to be properly introduced inside the rat.

The electrocardiogram morphology does not seem as perfect as the one obtained with the simulator at the laboratory since this is a real case, but the QRS complex can be identified perfectly which allows a correct recognition of the R wave as shown by the TTL pulse. Additionally the T wave can be identified as well just after the QRS complex. This visualization of the ECG can be considered sufficient for a simple monitoring of the small animal. However it was suspected that this less accurate representation was due to the presence of unfiltered noise or a bad positioning of the electrodes.

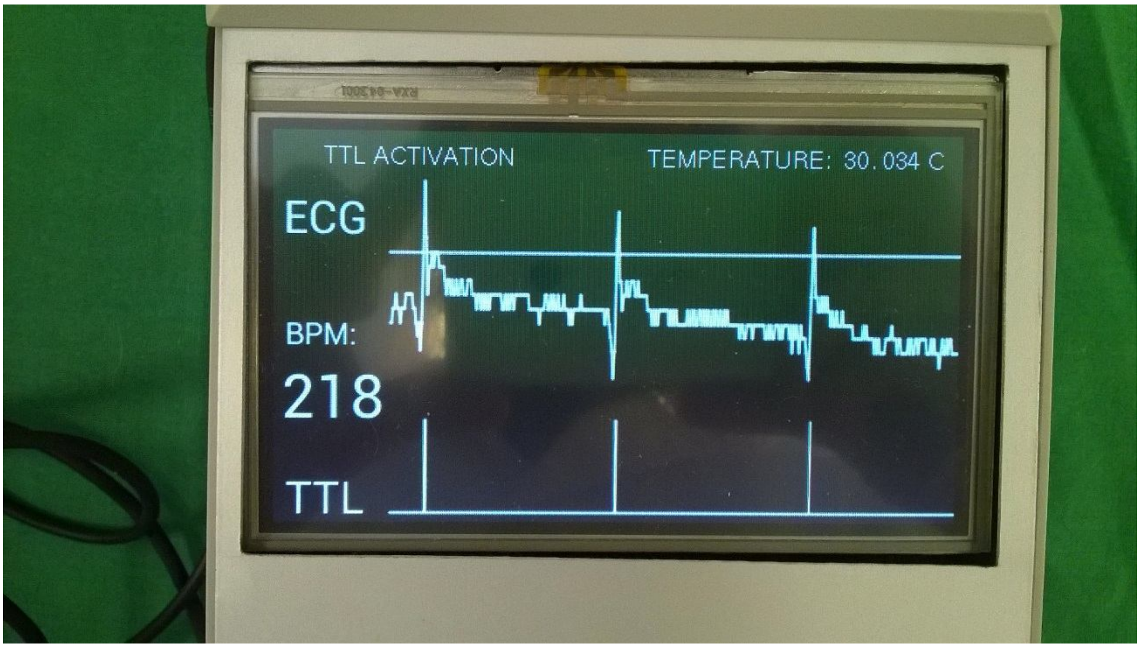


Figure 51. Zoom in image of the device's screen during rat monitoring

In order to verify that the heart rate of 218 bpm was the right value, the Vision PET monitor was alternatively connected to the rat as it can be observed in Figure 52. The heart rate given by this monitor is 219 bpm which is almost equal to the one calculated by our device. Therefore, it can be inferred that the algorithm implemented to extract the cardiac frequency works precisely.

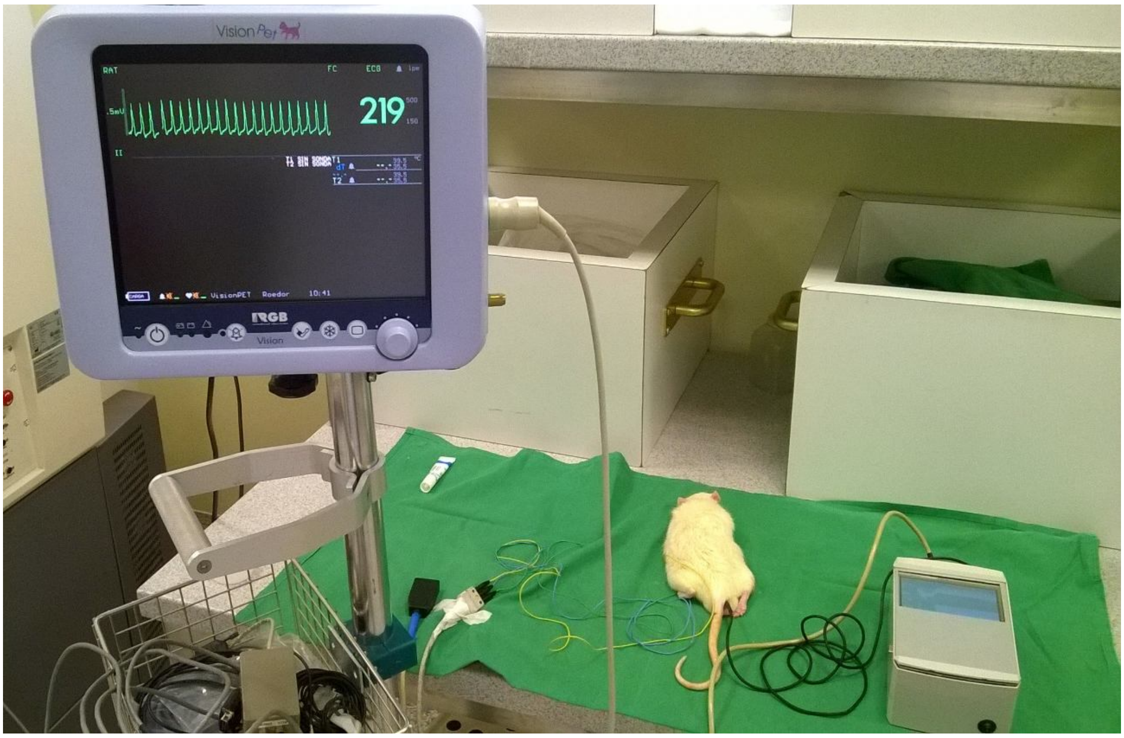


Figure 52. Monitoring with Vision Pet device as a reference for ECG device evaluation

In order to ameliorate the ECG representation on screen the electrodes were replaced again meticoulously. The results shown in Figure 53, demonstrate that a bad insertion of the needles was the cause of te introduction of some distoritons in ECG morphology. However, a considerable amount of noise of high frequency is still present and should be removed by filter implementation.

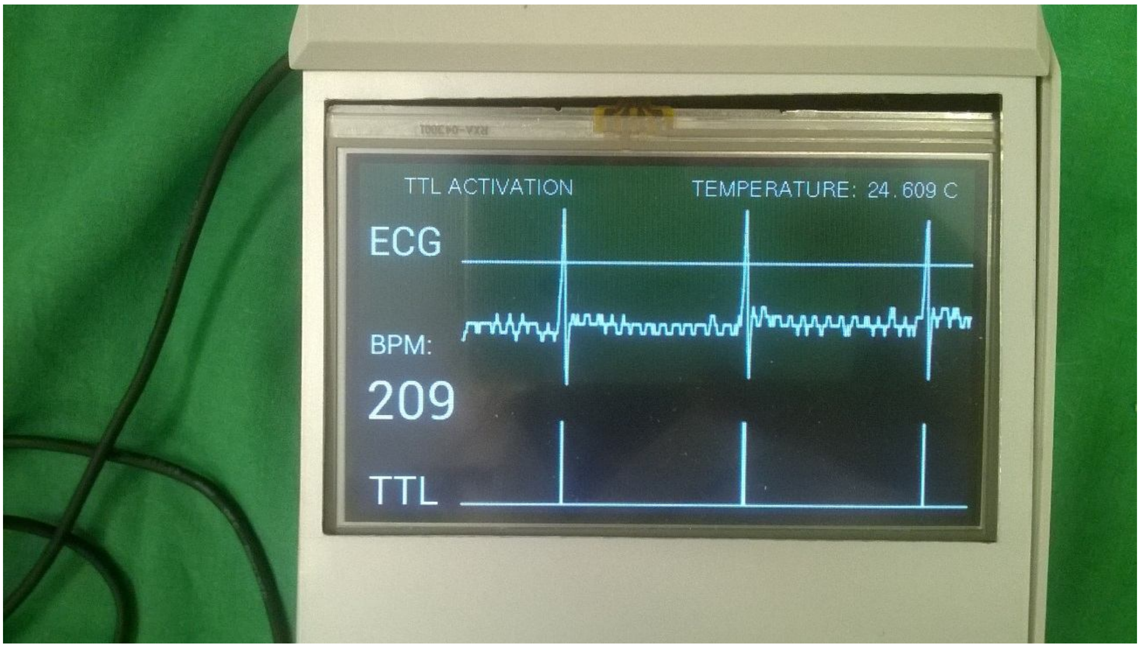
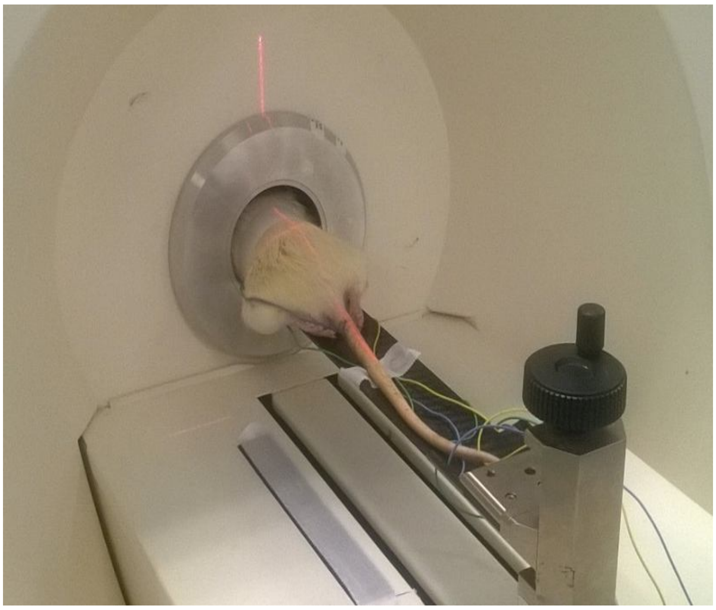


Figure 53. ECG representation results after proper placement of electrodes.

After testing the vital signs' monitoring functionalities of the device, the positive results and accurate TTL pulse display on screen lead to a gated imaging performance evaluation experiment.

This time, the rat was placed on the stretcher of the Argus PET/CT for PET image gated acquisitions, but instead of injecting a radionuclide, a ^{22}Na source was placed underneath the chest of the rodent. The gated imaging experiment was executed with the Vision Pet monitor to have a reference as shown in Figure 54. As it can be observed on the screen, the TTL pulse is activated each time an R wave is detected triggering the beginning of the 8 frames acquisitions spanning a cardiac cycle or R-R interval.

Figure 54. Gated imaging with Vision Pet monitor



During the imaging process the animal was introduced inside the scanner and its exact position was localized with a laser as shown in Figure 55.

This step is significantly relevant to posteriorly repeat the gated image acquisition experiment in the same conditions for comparison.

Figure 55. Gated imaging with laser positioning

In fact the same gated imaging experiment was repeated but using the ECG device prototype instead of the Vision Pet monitor. A BNC cable connected the ECG monitor to the PET/CT Super Argus machine for TTL synchronization.

A comparison of the results of the gating are presented in Figure 56 for the Vision Pet and the ECG device.

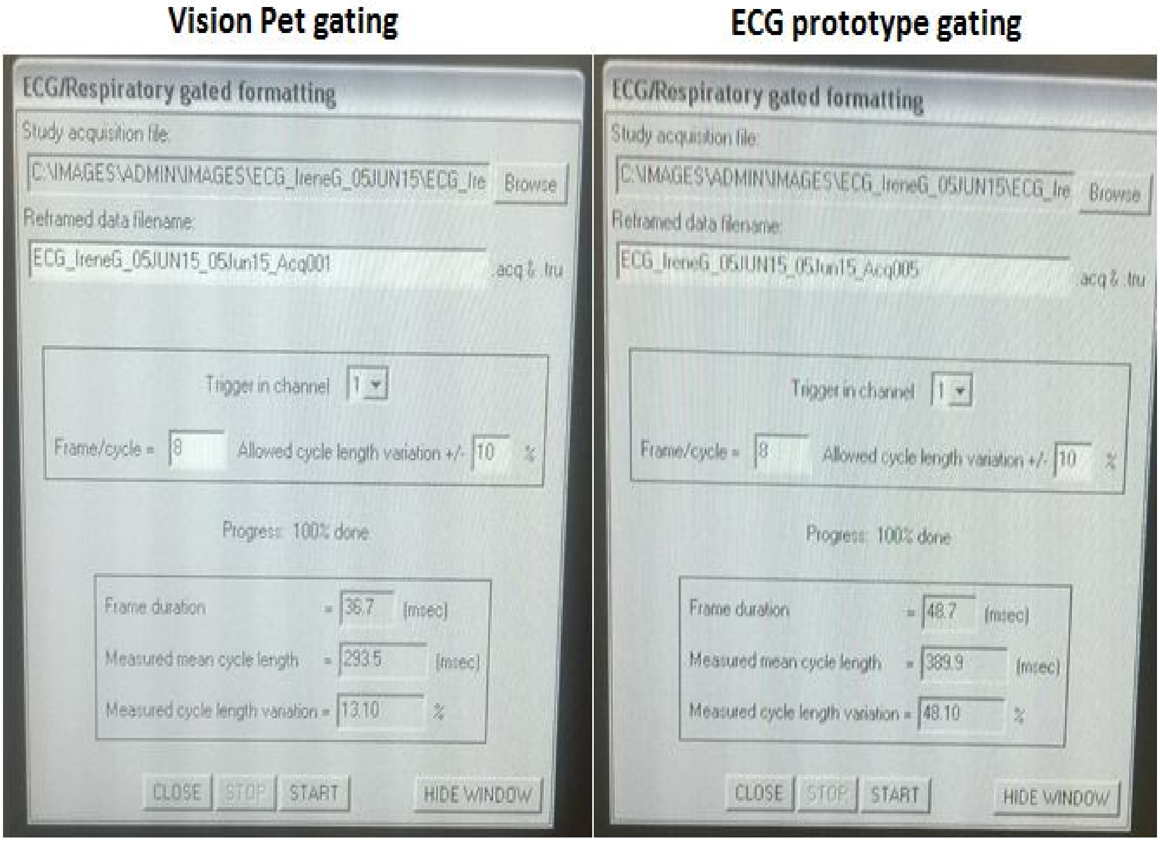


Figure 56. Gating comparison: Vision Pet vs ECG prototype

For the Vision Pet the Argus measured a mean cardiac cycle of 293.5ms which corresponds to 204 bpm dividing the R-R interval in eight frames of 36.7ms of duration. For the ECG prototype the Argus measured a mean cardiac cycle of 389.9ms which corresponds to 154bpm dividing the R-R interval in eight frames of 48.7ms of duration.

It's important to highlight that the gating process lasted 5 minutes each so the heart rate could vary slightly from one experiment to the other. So in that aspect they are completely independent one from the other, but the overall performance can still be compared.

In fact, the most relevant parameter here giving clues about the gating accuracy is the measured cycle length variation. For the Vision Pet it is in the order of 13% whereas in the case of our prototype it is critically higher, in the order of 48%.

This means that the TTL pulse is quite homogenous for the Vision Pet but very discontinuous in the case of our device. With a tolerance of 10%, half of the cardiac cycles will not be considered when using our device which is critical for obtaining high quality images. This problem is directly related to the spontaneous loss of the TTL pulse as noted when evaluating the TTL activation with the oscilloscope.

Additionally, short TTL pulses like the one produced by the prototype (less than 10 ms) could cause pulse losses on the Argus synchronization system since it was verified that the Vision Pet produces 40ms TTL pulses. This issue is under current investigation.

6 Conclusions

The principal objective of the presented Bachelor Thesis was to design, develop, program and test a small size ECG monitor device for small animal gated imaging.

In order to accomplish this, the main characteristics of the ECG device were specified to be: ECG bioelectrical signal acquisition, signal processing and representation in a screen to interface with the user. Some extra features performed by the prototype and to be displayed as well were sensing body temperature, heart rate calculation and detection of QRS complexes for generation of a TTL pulse for imaging synchronization.

All these functionalities were implemented in a sole system thanks to the combination of four main components: a microcontroller board Arduino MEGA2560, an analog to digital converter (ADC) ADS1298, a touch screen with video controller Gameduino2 and a temperature sensor AD2210. All these parts were electronically connected by means of an adapter proto-board.

A synchronous serial communication was established based on the SPI protocol in which the master (Arduino) selected alternatively one of the slaves (ADS1298 or Gameduino2) for data transmission. The ensemble was fitted in a plastic box with input and output connectors that enabled the device to interface with the imaging machine for image acquisition and with a computer for programming. The device could be powered by batteries, a power adapter or an USB connector.

The main code running the device was programed in the Arduino software environment and uploaded to the board for execution of a well-ordered list of successive commands. First, the ECG data acquired by ADS1298 is digitalized and transferred to the Arduino microcontroller for storage and processing. Arduino analyses the ECG signal to extract the heart rate and detect R waves for TTL activation. All this in combination with the temperature sensed by the AD22100 is displayed on the Gameduino2 screen in real time and the cycle is reinitialized again.

The final prototype's performance was evaluated in the laboratory with an ECG simulator. The successful results lead to further testing in small animals at Hospital General Universitario Gregorio Marañón where it was aimed to be used.

The ECG device showed to work properly in monitoring of rats, with some tolerable inaccuracies in gating image acquisition that are reflected in the next section.

6.1 Current device status

The small size ECG monitor developed in this project has undergone a vast improvement in comparison to the basic proof of concept prototype that was inherited from the previous bachelor project. At the moment the ECG device is a powerful system that complies the user requirements.

The construction of the adapter proto-board and the enclosure of the assembly in a box containing external batteries, make up a robust portable instrument ready to be implemented in a hospital and research environment.

The enhancement of the programming code allows the device to perform accurately in representing the ECG signal on screen, reading the temperature and calculating the heart rate when tested in rats. Additionally the TTL pulse is correctly activated with R wave detection even in the presence of important amounts of noise such as the baseline wander, respiration and muscular.

Yet it is true the results have been very positive in general, the device presents some limitations that prevent an excellent desired performance.

The introduction of noise such as muscular and power line interference, distorts the morphology of the electrocardiogram making difficult the identification of the P and T waves. An improper placement of the electrodes might also be the cause of this distortion. However, this is not a major problem since this device is not aimed for analysing the ECG features that usually indicate heart failure events. Nevertheless, representation of a clean signal would be desirable.

TTL pulse is activated in real time accurately most of the time triggering image acquisition but it is being lost from time to time, and this could have some consequences in the process of gating. The source of this problem is supposed to be a communication glitch that occurs when the ADS1298 is not ready to send the data for the Arduino to process it; this implies that communication protocol should be reviewed. Also, the duration of the TTL synchronism pulse might require some adjustments depending on the device where it is going to be connected. This issue has to be refined in the future to allow an initial programmable setup by the user according to its needs.

Finally, some electrical problems were encountered during testing, causing the screen to freeze at the boot up. We have identified the proto-board as the source of that error, so a PCB should be designed to replace it.

6.2 Future improvements

Some ideas can be suggested to overcome the current limitations of the ECG device, as well some extra improvements to improve the device.

Firstly, the problems related to noise interference could be solved by implementation of analog or digital filters. Especially a notch filter for removing the power line interference and an additional low pass filter to eliminate even more the muscular noise would be a good solution to clean the ECG signal. Digital filters would be more appropriate since there is not physical space for analog implementation. Although digital filtering of the ECG signal can be successfully accomplished in Matlab or Labview, filter implementation in the Arduino microcontroller was not as straightforward due to computational delays and memory limitations. As a consequence, some more investigation in this matter would be needed in the future to include digital ECG filters in the Arduino code in a more intuitive way.

Secondly, a more powerful microcontroller such as the TM4C1294 LaunchPad from Texas Instruments could be a good solution to avoid communication problems between the ADS1298 and the microcontroller causing inaccuracies in TTL activation. In that manner SPI communication could be greatly enhanced, as well as the screen refresh rate could also improve. Also it will provide more digital signal processing that could be used to implement more sophisticated filters.

A printed circuit board substituting the adapter proto-board should solve the electrical problems that have arisen. This would eliminate the potential issues in soldering connections and would increase significantly the stability of the system.

As extra features, some developments in the programming code can improve the system as well. The time window of the ECG signal on screen could be adapted for different measurement modes in different manners. The heart rate calculated could automatically identify whether it is a rat or mouse and adapt the number of samples displayed. This option was tested in the current device but the mandatory initialization of the variables at the beginning of the code made difficult sudden modifications of the size in the main loop since it compromised the timing. Another alternative could be to take advantage of the touch screen to create an interface menu in which the user could select on the screen different operation parameters.

6.3 Scientific and social impact

The satisfactory results of this project could have interesting outcomes for science and society.

The incorporation of a small size ECG device like this in the "Laboratorio de Cirugía Experimental" of HGUGM entails direct advantages for pre-clinical research. First, the miniaturization and portability of the device will give the possibility of cardiac monitoring small animals anywhere. The constraint of a heavy device with reduced mobility and difficult to be manipulated will be eliminated. This will boost the practicability of a medical instrument like this, which could be used in any environment.

Second, it is an easy to use device that requires minimum training. By simply switching it on and placing the electrodes, the monitoring of different vital parameters will be displayed. The TTL synchronism activation is also automatic, which facilitates the tasks of gated imaging set up. Furthermore, the ECG prototype offers the possibility to be reprogrammed through the external USB connector if any other extra features need to be implemented.

The good performance of the gating will enable to increase the quality of the cardiac images acquired during pre-clinical experiments by reducing the blurring of the images due to movement. This will be an important advance in small-animal cardiac studies that investigate how the hemodynamic properties become affected as a consequence of electrophysiological heart failures or diseases. An improvement of the resolution will push the barriers of knowledge in which concerns discovering new treatments.

Consequently, an amelioration of pre-clinical cardiac imaging will not only benefit science but society as well for a reasonable price. The estimated material cost of producing a single unit of this ECG device is roughly 300€ as detailed in the next section. Even if the final price would increase up to 500€ when adding some other factors, still it would be much affordable than the more basic monitors available in the market, and if the adequate business case in prepared, it could be an interesting candidate for a commercial product in mass production.

7 Budget

In this section a breakdown of the costs is detailed to have an idea of the total budget of this project.

7.1.1 Material costs

Table 5. Material costs breakdown

Description	Costs (€)
Arduino Mega 2560	36
ADS1298	155
Gameduino2	70
RETEX enclosure 103 P 33103005	14
Others (batteries, electrodes, connectors, cables...)	20
TOTAL Material costs	**295**

7.1.2 Human resources

Table 6. Human labour costs breakdown

Category	Human's months	Cost of human/month (€)	Costs (€)
Technical engineer	3	1500	4500
Laboratory technician	0.5	1500	750
Senior engineer supervisor	4	3000	12000
Biomedical Engineer	8	2000	8000
TOTAL Human labor costs			**25250**

7.1.3 Indirect costs

The indirect costs correspond to a 20% of the material and human costs, which corresponds to 3909 €.

7.1.4 General costs and industrial benefit

The general costs and industrial benefit correspond respectively to 16% and 6% of the material costs. Then the general costs are estimated to be 47.2€ and the industrial benefit 17.7€.

7.1.5 Summary of the costs

Table 7. Summary of the costs

Description	Costs (€)
Total of material Costs	295.00
Total of Human labor costs	19,250.00
Indirect costs	3,909.00
Total without IVA	23,454.00
IVA 21%	4,925.34
TOTAL	**28,379.34**

8 Bibliography

[1] Aimen K. Farraj, Mehdi S. Hazari and Wayne E. Cascio, "Review: The Utility of the Small Rodent Electrocardiogram in Toxicology," *Toxicological sciences,* vol. 121, no. 1, pp. 11-30, 2011.

[2] Richard D. Patten and Monica R. Hall-Porter, "Small Animal Models of Heart Failure. Development of Novel Therapies, Past and Present.," *Circulation: Heart Failure,* vol. 2, pp. 138-144, 2009.

[3] Chao Wu, Pieter E.B. Vaissier, Brendan Vastenhouw, Johan R. de Jong, Riemer H.J.A. Slart, Freek J. Beekman, "Influence of Respiratory Gating, Image Filtering, and Animal Positioning on High-Resolution Electrocardiography-Gated Murine Cardiac Single-Photon Emission Computed Tomography," *Molecular Imaging,* vol. 14, pp. 1535-3508, January-March 2015.

[4] Benoit Desjardins and Ella A. Kazerooni, "Review: ECG-Gated Cardiac CT," *American Journal of Roentgenology,* vol. 182, no. 4, pp. 993-1010, 2004.

[5] Yongfeng Yang, Stephen Rendig, Stefan Siegel, Danny F Newport and Simon R Cherry, "Cardiac PET imaging in mice with simultaneous cardiac and respiratory gating," *Physics in Medicine and Biology ,* vol. 50, no. 13, 2005.

[6] Z. Sun, "Multislice CT angiography in cardiac imaging: prospective ECG-gating or retrospective ECG-gating?," *Biomed Imaging Interv Journal,* vol. 6, no. 1, p. e4, 2010.

[7] B. V. Regulez, "Dispositivo de monitorización de la señal cardíaca de pequeño animal de laboratorio para la adquisición de imágenes médicas con sincronismo cardíaco.," 2014.

[8] Sven Kaese and Sander Verheule, "Cardiac electrophysiology in mice: a matter of size," *Frontiers in Physiology: Cardiac Electrophysiology,* vol. 3, no. 345, pp. 1-19, 2012.

[9] Arthur C. Guyton and John E. Hall, Textbook of Medical Physiology, Saunder Elsevier, 2011.

[10] M. Homoud, “http://ocw.tufts.edu/Content/50/lecturenotes/634463/634544,” Tufts University, 2007. [Online].

[11] R. Northrop, “Chapter 1: Sources and Properties of Biomedical Signals p.1-16,” in *Analysis and Application of Analog Electronic Circuits to Biomedical Instrumentation*, CRC Press, 2012.

[12] "Netter images," Elsevier, [Online]. Available: http://www.netterimages.com/ecg-and-the-cardiac-conduction-system-labeled-hansen-ca-1e-cardiology-frank-h-netter-37916.html.

[13] Vicente Zarzoso, Ronald Phlypo, Olivier Meste, Pierre Comon, “Chapter 3 – Signal Extraction in Multisensor Biomedical Recordings,” in *Advances in Biomedical Engineering*, Elsevier B.V., 2009, pp. 95-143.

[14] Johannes Pätzold and Heinz Kresse, “Electrocardiography,” in *Handbook of Electromedicine*, Berlin, Siemens Aktiengesellschaft, 1985, pp. 1-28.

[15] [Online]. Available: http://perpetuum-lab.com.hr/wiki/plab_wiki/interna-medicina/ekg-r151/.

[16] Leif Sörnmo and Pablo Laguna, “Chapter 6- The electrocardiogram: a brief background,” in *Bioelectrical Signal Processing in Cardiac and Neurological Applications*, Elsevier, 2005, p. 411–452.

[17] "Small Animal Cardiology," [Online]. Available: http://research.vet.upenn.edu/smallanimalcardiology/ECGTutorial/tabid/4930/Default.aspx.

[18] Detweiler D.K. and Erickson H.H, Regulation of the Heart, in Dukes' Physiology of Domestic Animals, 12th ed., Reece W.O., Ed. Copyright 2004 by Cornell University.

[19] "Mauvila," [Online]. Available: http://www.mauvila.com/ECG/ecg_artifact.htm.

[20] "MEDTEQ," [Online]. Available: http://www.medteq.info/med/ECGFilters.

[21] J. D. Bronzino, “X-ray,” in *Biomedical Engineering Handbook*, CRC Press, 2000, pp. 61-1 to 61-36.

[22] "Physics Central," [Online]. Available: http://www.physicscentral.com/explore/action/scans.cfm.

[23] R. Salzer, “Chapter 7: Tracer imaging,” in *Biomedical Imaging: Principles and Applications*, John Wiley & Sons, 2012.

[24] [Online]. Available: http://www.tmland.it/news.aspx?IDNews=7644.

[25] Z.H. Cho, Joie P. Jones, Manbir Singh, Foundations of medical imaging, John Wiley & Sons, 1993.

[26] Edward D Nicol and Simon Pg Padley, "Non-invasive Cardiac Imaging: Current and Emerging Roles for Multi-detector Row Computed Tomography.," *British Journal of Cardiology,* vol. 14, no. 3, pp. 143-150, 2007.

[27] Asit K. Paul and Hani A. Nabi, "Gated Myocardial Perfusion SPECT: Basic Principles, Technical Aspects, and Clinical Applications," *Journal of Nucluear Medicne Technology,* vol. 32, no. 4, pp. 179-187, 2004.

[28] Atmel, "8-bit Atmel Microcontroller with 16/32/64KB In-System Program," *ATmega 2560 datasheet,* 2014.

[29] Arduino, “Arduino Mega 2560,” [Online]. Available: http://www.arduino.cc/en/Main/ArduinoBoardMega2560.

[30] TexasInstruments, "Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements," *ADS1298 datasheet,* 2014.

[31] F. T. D. I. FTDI, "FT800 Embedded Video Engine," *FT800 datasheet ,* 2013.

[32] AnalogDevices, "Voltage Output Temperature Sensor With Signal Conditioning," *AD22100 datasheet.*

[33] RETEX, “Retex series 103P 33103005 catalogue,” [Online]. Available: http://www.retex.es/Downloads/20141007-Enclosures-catalogue-low.pdf.

[34] Motorola, "SPI block guide V03.06," *SPI communication document number S12SPIV3/D,* 2000.

[35] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus.

[36] Jiapu pan and Willis J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEE Transactions on Biomedical Engineering,* Vols. BME-32, no. 3, pp. 230-235, 1985.

[37] RIGOL, "DG4000 Series Waveform Generators," *DG4062 datasheet,* 2011.

9 Appendices

9.1 Appendix 1: Arduino Mega 2560 pin diagram and mapping table

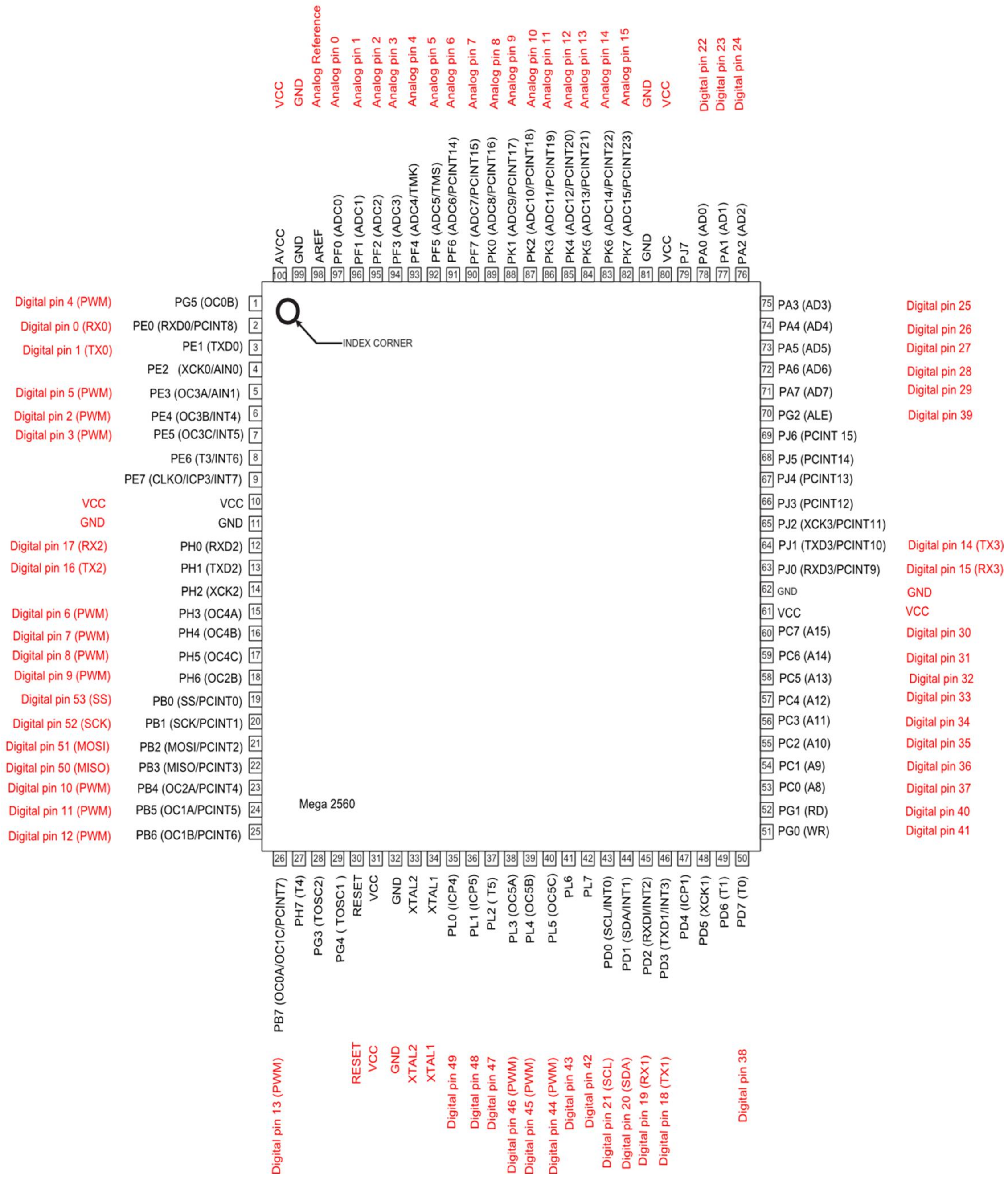


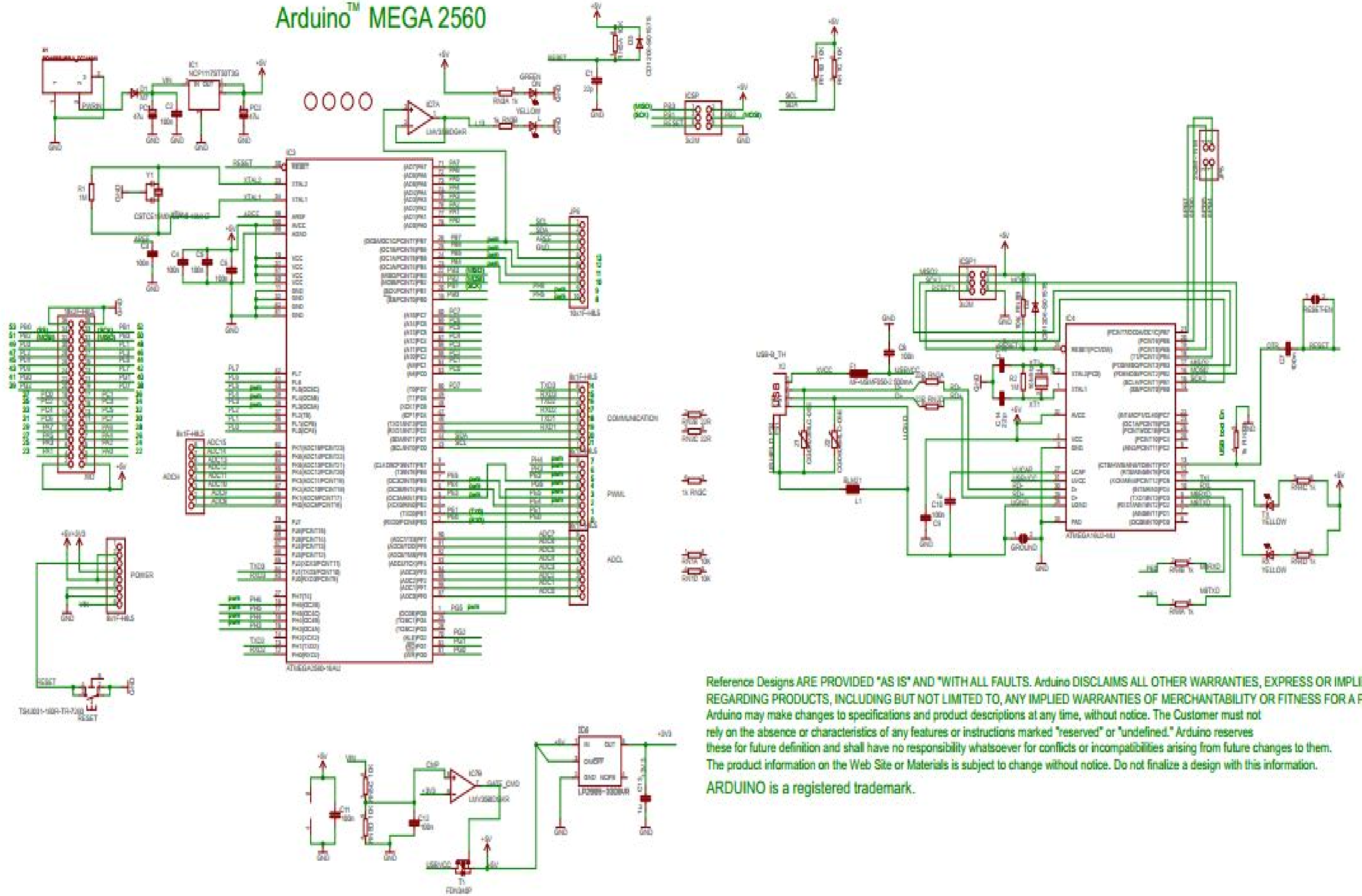
Figure 57. Arduino Mega 2560 pin diagram

Table 8. Arduino Mega 2560 PIN mapping table

Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	

50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

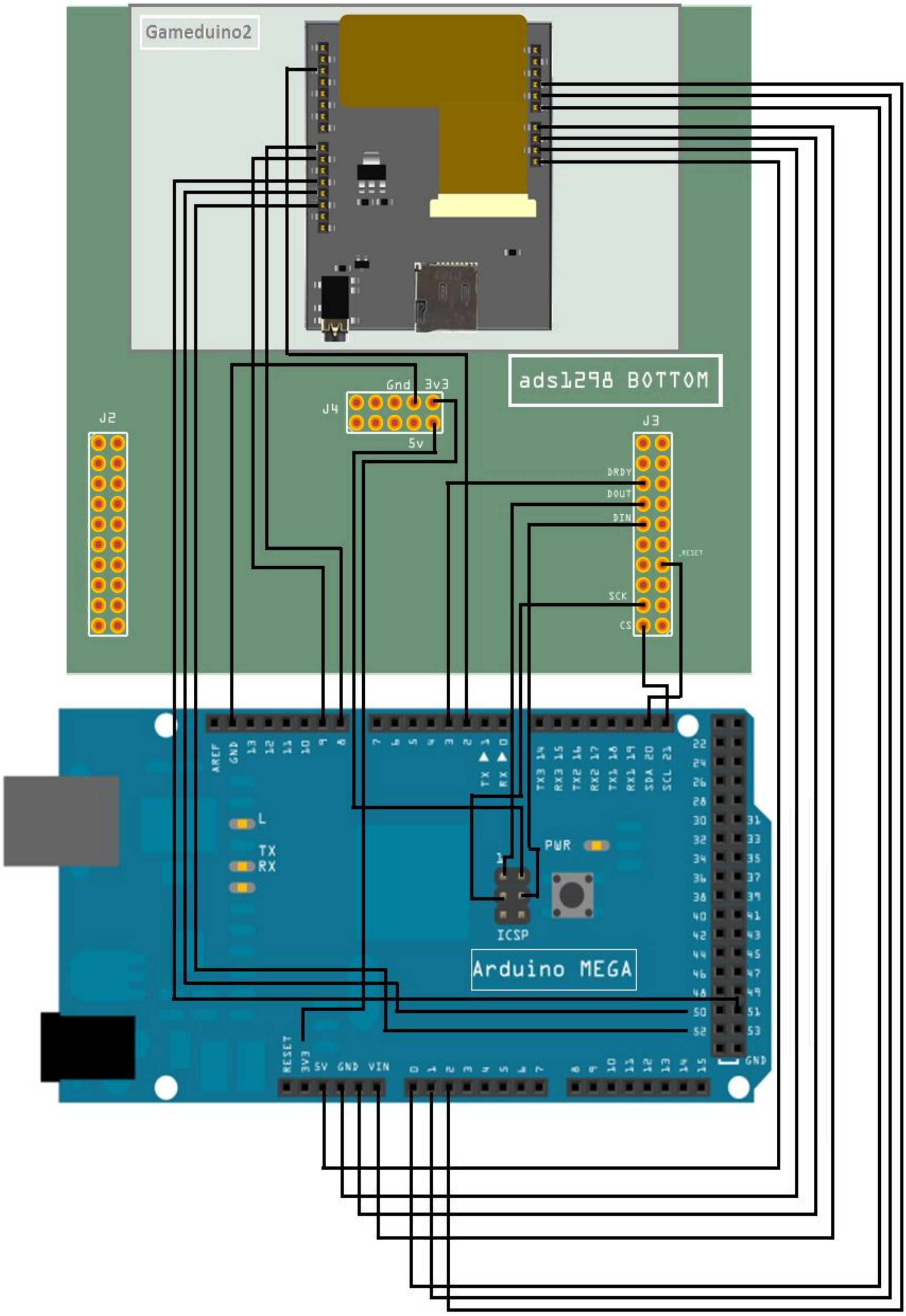
Arduino™ MEGA 2560



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE
Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

9.2 Circuit schematic illustrating connections of adapter proto-board.



9.3 Appendix 2: Arduino programming code

```
// Include libraries: SPI, EEPROM and GD2
#include <SPI.h>
#include <EEPROM.h>
#include <GD2.h>

//Pin configuration and datasize
//ADS1298
const int DRDY_pin = 3;  //ADS Data Ready pin (interrupt pin)
const int SS_pin =21; //ADS slave select //SCL TWI pin
const int ADSReset_pin = 20; //SDA TWI pin

//Gameduino
const int MOSI_pin = 51; //MEGA
const int MISO_pin = 50; //MEGA
const int SCK_pin = 52; //MEGA
const int CS_GD_pin = 8; //MEGA //GD slave select (53)

const int TTL_pin = 4; //pin for TTL output
const int ADSInterrupt = 1; //Interrupt int.1 for ADS interruptions (corresponding to
pin3)
const int dataSize = 400;
const int Temp_pin= A3;

float temperature;
int measure; //byte

//2. and 3. channels' matrixes and number-of-data counters
 int nData2=0;
```

```
int nData3=0;
int nData=0;

volatile int data2[dataSize]; //Channel 2 data matrix
volatile int data3[dataSize]; //Channel 3 data matrix

volatile int data[dataSize]; // data containing: Subtraction channel 3- channel 2

volatile int dataAux[3]; //auxiliar matrix for reading data from ADS of length 3
int countA=0; //auxiliar counter used in ADS configuration
byte auxCont=0; //auxiliar counter used in TTL activation
int auxTTL[10]; //dataSize
int RR_interval[dataSize];

int ContPulso;
int Pulso ;
volatile byte QRS;

volatile int threshold;


void setup() {

  //Pin settings

  pinMode(CS_GD_pin, OUTPUT);
  digitalWrite(CS_GD_pin, HIGH); //Disable GD during setup



  pinMode(SS_pin, OUTPUT);
  digitalWrite(SS_pin, HIGH);  //Disable ADS during setup
  pinMode(TTL_pin, OUTPUT); //Disable ADS during setup
```

```
  digitalWrite(TTL_pin, LOW); //Disable ADS during setup

  pinMode(MOSI_pin, OUTPUT);
  pinMode(MISO_pin, INPUT);
  pinMode(SCK_pin, OUTPUT);
  pinMode(ADSReset_pin,OUTPUT);
  pinMode(DRDY_pin, INPUT);

  pinMode(Temp_pin,INPUT);

  delay(400); // wait 400ms for Gameduino to boot

  GD.begin(0); //0 value to avoid SD card initialization

  delay(400);

  Serial.begin(115200); //Baud rate for serial sending of data
}

void loop() {

  //During the first 4 loops: Calibration of thresholds and dynamic range
  //TTL pulse is inactive
  if (auxCont <4){
   auxCont++;
   }

   plot();

   QRS=0;
   ADS_ready();  //ADS config. and emptying matrix
```

```
    Serial. print("\n");
    Serial.print("dataa");

  for (int i=0; i<dataSize; i++) {
    attachInterrupt(ADSInterrupt, read_subfunction, FALLING);
    delay(2); //0.8 secs --> 400samples/500(samples/sec)
    detachInterrupt(ADSInterrupt); //deactivate reading interrupt+

    data[i] = ((data3[i]) - (data2[i]));

    //TTL actvation
    if (auxCont >=4){
    //after calibration of thresholds and stabilization of dynamic range
       if (data[i] > threshold){
           digitalWrite(TTL_pin, HIGH); //ACTIVATION TTL PIN  //change to DIGITAL
WRITE
           /*if (QRS==0){ //is the previous point was under threshold
          QRS=1; //an Rpeak is detected
          }   */

        }

        else {
           digitalWrite(TTL_pin, LOW); //DEACTIVATION TTL PIN
           //QRS==0;
        }
      }



  }

   Serial.print("\n");
```

```
  Serial.print("AFTER INTERRUPT");
  Serial.print("\n");
  for (int i = 0; i<dataSize; i++) {
   data[i] = ((data3[i]) - (data2[i]));
   Serial.print(data[i]);
   Serial.print("\t");
   }
   Serial.print("\n");
   Serial.print("\n ");

  Serial.print("\n ");
  digitalWrite(SS_pin, HIGH); //disable ADS as slave

}


void plot() {

 //Set SPI parameters for GD BEFORE selecting GD as slave
  SPI.setDataMode(SPI_MODE0);
  SPI.setBitOrder(MSBFIRST);
  SPI.setClockDivider(SPI_CLOCK_DIV2); //16MHz/2= 8MHz

 GD.resume();
 digitalWrite(CS_GD_pin, LOW); //Select GD as slave

//Print screen
 GD.ClearColorRGB(0x103000); //sets the color used for screen
//clear. Here the color is 0x103000, a dark green.
 GD.Clear();//clears the screen to the dark green color.

 GD.cmd_text(320, 17, 22, OPT_CENTER, " TEMPERATURE:");
```

```
 GD.cmd_text(35, 60, 30, OPT_CENTER, "ECG");
 //In this case the text is being drawn in the center of the screen, at (x; y) coordinates (25,
20).
 GD.cmd_text(35, 255, 30, OPT_CENTER, "TTL");
 GD.cmd_text(32, 140, 28, OPT_CENTER, "BPM:");

 if (auxCont>=4){
  GD.cmd_text(100, 17, 22, OPT_CENTER, " TTL ACTIVATION ");
  }
 else{GD.cmd_text(120, 17, 22, OPT_CENTER, "CALIBRATION IN PROCCESS");}



//Subtraction of channels 2 and 3 signals to get a unique signal:
  Serial.print("\n");
  /* for (int i = 0; i<dataSize; i++) {
    data[i] = ((data3[i]) - (data2[i]));
    Serial.print(data[i]);
    Serial.print("\t");
    }
    Serial.print("\n");*/



//Algorithm for calculating parameters for data scaling (matrix´s maximum and
minimum)
  int auxMin=data[1];
  int auxMax=data[1];
  int m;
  int M;
   for (int i = 0; i<dataSize; i++) {
    m=data[i];
    if (m<auxMin){
     auxMin=m;
    }
```

```
    M=data[i];
    if (M>auxMax){
     auxMax=M;
    }
  }
  Serial.print("auxMin");
  Serial.print(auxMin);
  Serial.print("\n");
  Serial.print("auxMax");
  Serial.print(auxMax);
  Serial.print("\n");

  threshold= 0.7*(auxMax-auxMin)+auxMin;
  Serial.print("threshold");
  Serial.print(threshold);
  Serial.print("\n");
  int screen_threshold=map(threshold,auxMin, auxMax,170,34);
   Serial.print("screenthreshold");
  Serial.print(screen_threshold);
  Serial.print("\n");
   //Pintar el threshold en pantalla
   GD.Begin(LINES);
   GD.Vertex2ii((80),screen_threshold);  GD.Vertex2ii((480),screen_threshold);


//Info for scaling
  int x1=0; //data in old scale
  int x2=0; //data in new scale


 // PLOT ECG
GD.Begin(LINE_STRIP);
```

```
 for (int i = 0; i<dataSize; i++){
//Scale and plot data
  x1=data[i];
  x2=map(x1,auxMin, auxMax,170,34);//map(value, fromLow, fromHigh, toLow, toHigh)

 //Plot data. Start in horizontal axis pixel 80
     GD.Vertex2ii((i+80),x2);
    //draws the points at coordinates of x from 80:480 (400 points =dataSize) and y from
34:170

  }

  //Pulse variables
 int ContPulso=0;
 int Pulso=0;
 byte QRS=0;
 int auxperiod=0;
 float period=0;

 //PLOT TTL
   for (int i = 0; i<dataSize; i++) {
    x1=data[i];

    //R-peak detection and TTL activation
    if (x1>threshold){ //threshold for Rpeak
     //analogWrite(TTL_pin, 255);
     if (QRS==0){ //is the previous point was under threshold
      QRS=1; //an Rpeak is detected
      auxTTL[ContPulso]=i; //when a QRS is detected the position is saved to calculate R-
R distances later
      Serial.print(auxTTL[ContPulso]);
      Serial.print(" ");
    }
```

```
  if (i==(dataSize+1)){
   //To count the last pulse & set TTL_pin LOW at the end of reading
   ContPulso++;
   //analogWrite(TTL_pin, 0);
   }
 GD.Begin(LINES);
 GD.Vertex2ii(i+80, 265);
 GD.Vertex2ii(i+80, 200);     }

 else {
  //analogWrite(TTL_pin, 0);
  if (QRS==1){ //when previous point was above the threshold

   ContPulso++; //a pulse has to be counted
   QRS=0;
   }
 GD.Begin(LINES);
 GD.Vertex2ii((i+80),265);
 GD.Vertex2ii((i+80),265);
 }
}

if(ContPulso>1){
  Serial.print("\n ");
  Serial.print("RRinterval");
   for (int j=0; j<ContPulso-1;j++) {
   //Succesive RR intervals in seconds.

  RR_interval[j]= (auxTTL[j+1] - auxTTL[j] +1); //Each sample takes 1/500SPS =0.002
secs.
```

```
     Serial.print(RR_interval[j]);
     Serial.print(" ");

      auxperiod=auxperiod+RR_interval[j]; //sum of RR_intervals for mean calculation
      //GD.cmd_text(125+ j*25, 255, 22, OPT_CENTER, RR_interval[j]);
      }

    // period = ((auxTTL[ContPulso]-auxTTL[0]+1)/ContPulso)*0.002;
      period=auxperiod/(ContPulso-1);//mean RRinterval corresponding to the period.
      Serial.print("\n ");
      Serial.print("period");
      Serial.print("\n ");
      Serial.print(period);
      Pulso= (1/period)*60*500; //500=1/0.002
    //Pulso=ContPulso*75; //75=60/0.8, for taking pulses counted during 0.8 secs to
pulses in one minute
      Serial.print("\n ");
      Serial.print("pulso");
      Serial.print(Pulso);
   }

   else {
     if (auxTTL[0] <= (dataSize/2)){
      period= dataSize-auxTTL[0];
     }
     else{
    period=auxTTL[0]; }

      Pulso= (1/period)*60*500; //500=1/0.002

   }
```

```
//Plot pulse
 GD.cmd_number(40, 160, 31, OPT_CENTERX, Pulso);

// TEMPERATURE
 measure= analogRead(Temp_pin);
 temperature=(measure*(5.0/1024) - 1.375) / 0.0225;//.0049 volts (4.9 mV) per unit.
  //temperature=(measure* 0.217226044) - 61.1111111; //faster

 plotfloat(415,7,temperature);

 GD.cmd_text(460, 17, 22, OPT_CENTER, "C");



   GD.swap();  // For "refreshing" display
  //tells the graphics system that the drawing is finished and it can be made visible on the
screen.

//Commands for ending communication with GD
 GD.__end();
 digitalWrite(CS_GD_pin, HIGH);
}



void ADS_ready() {  //ADS setup. First time complete setup, then only empty matrixes

//Set SPI parameters
   SPI.setDataMode(SPI_MODE1);
   SPI.setBitOrder(MSBFIRST);
   SPI.setClockDivider(SPI_CLOCK_DIV8); //16MHz/8= 2MHz
   digitalWrite(SS_pin, LOW);
   SPI.begin(); //Initialize SPI for ADS
```

```
//complete setup (once only)
  if (countA==0){
  //ADS reset, enable and required delays
   delay(1);
   digitalWrite(ADSReset_pin, LOW); //Analog pin A4 is set LOW
   delay(2);
   digitalWrite(ADSReset_pin,HIGH); //Analog pin A4 is set HIGH
   delay(10);
   countA++;
   ADS1298_init_conf();
  }

//initialize matrixes (always).clear stored data.
 Packet_init();
}



void read_subfunction(){ //function entered when ADS indicates data ready interruption


  //Read channels (all, mandatory due to ADS specifications) and save only ch2 and ch3 in
respective matrixes, if nData<dataSize.
  for (int i = 0; i<9; i++){ //i<8


   for (int j = 0; j<3; j++){
    dataAux[j] = SPI.transfer(0x00);
   }


  if (nData2<dataSize){
   if (i==2){


    data2[nData2++] = int (word(dataAux[0],dataAux[1]));
```

```
    }
   }
   if (nData3<dataSize){
    if (i==3){
     data3[nData3++] = int (word(dataAux[0],dataAux[1]));
     }

    }
   }

   }

void  ADS1298_init_conf() {  //ADS1298 configuration. Opcode and multi-byte commands.

  //WAKEUP ADS1298 opcode
   SPI.transfer(0x02);

  //SDATAC opcode (stop RDATAC mode)
   SPI.transfer(0x11);
   delay(2);

  // Set device in HR (High resolution) Mode and DR (Data Rate) = Fmode/1024 500 SPS
  //WREG opcode:
   SPI.transfer(0x41);// 4=Write Registers, 1=Starting at register with address 1
   SPI.transfer(0x02);// Write 3registers-1=2
  //Config1 0x86.
   SPI.transfer(0x86); //0x86=10000110 --> HR Mode and DR (Data Rate) = Fmode/1024
500 SPS
  //Config2 0x00
   SPI.transfer(0x00); //0x00= Reset Value --> Everything by default
  //Config3 0xC1
   SPI.transfer(0xC1);//0xC1=11000001--> Enable internal reference buffer &RLD not
connected
```

```
//Set all channels to Input Shorts and Power Down
//Except for Ch2 and Ch3 to gain 12
SPI.transfer(0x45);//0x45= Write(4) Starting at register with address 5 (CH1)
SPI.transfer(0x07);//Write 8registers-1=7
//CH1 address 05
SPI.transfer(0x81);//0x81=10000001 --> 1=PD (Power Down), 000=Gain default, 0,
001=Input short
// CH2 address 06
SPI.transfer(0x60);//0x60=01100000 --> 0=Not Power Down, 110=Gain of 12, default
//CH3 address 07
SPI.transfer(0x60);//0x60=01100000 --> 0=Not Power Down, 110=Gain of 12, default
//CH4 address 08
SPI.transfer(0x81);//0x81=10000001 --> 1=PD (Power Down), 000=Gain default, 0,
001=Input short
//CH5 address 09
SPI.transfer(0x81);//0x81=10000001 --> 1=PD (Power Down), 000=Gain default, 0,
001=Input short
//CH6 address 0A
SPI.transfer(0x81);//0x81=10000001 --> 1=PD (Power Down), 000=Gain default, 0,
001=Input short
//CH7 address 0B
SPI.transfer(0x81);//0x81=10000001 --> 1=PD (Power Down), 000=Gain default, 0,
001=Input short
//CH8 address 0C
SPI.transfer(0x81);//0x81=10000001 --> 1=PD (Power Down), 000=Gain default, 0,
001=Input short

//START Opcode. Start/sincronize conversion
 SPI.transfer(0x08);
 delay(1);

//Opcode RDATAC. Read Data Continous from now on.
 SPI.transfer(0x10);
```

```
}

//Initialize matrixes data2 and data3 and data counters
void Packet_init(){
  for (int i = 0; i<dataSize; i++){
    data2[i] = 0x00;
    data3[i] = 0x00;
    data[i] = 0x00;
    auxTTL[i]=0x00;
    RR_interval[i]=0x00;

  }
  nData2=0;
  nData3=0;
  nData=0;
}

static void plotfloat(int x, int y, float f)
{
 byte font = 22;
 GD.cmd_number(x - 2, y, font, OPT_RIGHTX | OPT_SIGNED, int(f));
 GD.cmd_text(  x,    y, font, 0, ".");
 GD.cmd_number(x + 8, y, font, 3, int(1000 * abs(f)));
}
```