TÉCNICO LISBOA

# Long Term Real-Time Pervasive Monitoring System for non-Hospitalized Patients

**André Ferreira Manso**

Thesis to obtain the Master of Science Degree in

## Biomedical Engineering

Supervisor(s): Prof. Ana Luisa Nobre Fred
Dr. Rui Cruz Ferreira

### Examination Committee

Chairperson: Prof. Full Name
Supervisor: Prof. Full Name 1 (or 2)
Member of the Committee: Prof. Full Name 3

**Month Year**

# Acknowledgments

O meu primeiro agradecimento será para a Prof. Ana Fred que me convidou a participar neste projeto e me apoiou em todas as fases, esclarecendo e solucionando todas as questões que foram surgindo.

Um obrigado também ao Rui César das Neves por ter ajudado a fazer deste projeto uma realidade, e ao Filipe Mariano por ter cedido o software do projeto Telemold e ajudado á sua compreensão, sem o qual a implementação teria sido muito mais morosa e custosa.

Gostaria também de agradecer a toda a equipa do Hospital de Santa Marta pelo apoio durante a realização do projeto, em particular ao Dr. Rui Ferreira por ser um impulsionador da inovação e digitalização da medicina em Portugal, por portugueses.

Um grande obrigado também ao Hugo Silva por ter contribuído com a montagem de uns bonitos Bitalinos, por ter sempre ideias espetaculares e ter sempre uma palavra de encorajamento a dar. Novamente agradeço a Prof. Ana Fred por ter também dado um toque de estilo ao que, sem ela, seriam umas faixas muito feias.

Não poderia também deixar de agradecer a todos os amigos que me apoiaram quando precisei de ajuda ou animo. Um agradecimento especial à minha família e à minha mais que tudo por toda a ajuda, apoio e carinho que me deram no decorrer destes meses.

Finalmente, quero deixar um agradecimento ao Miguel Martinho por ter aturado os meus queixumes e ter sido parceiro e alicerce em incontáveis brainstorms e aventuras.

## Resumo

Igual ao Abstract

**Palavras-chave:** palavra-chave1, palavra-chave2,...

## Abstract

In recent years, sensors and data collection devices have become an ever more ubiquitous presence. This has a major potential concerning medical procedures, opening the path for personalized medicine and allowing for a more targeted and efficient diagnose and therapeutic.

In this work a novel pervasive monitoring system is proposed. It is designed to allow constant and long-term acquisition of any desired variables and visualize the signals remotely in real time via a web interface. The system is implemented to be completely versatile concerning the sensors it can interact with.

System's functioning is based in two components, a smartphone app and a central server. The smartphone is responsible for connecting with all sensors via Bluetooth and processing, storing and relaying all data to the server. The later is responsible for long-term storing of the data, and for serving a web interface where acquisition parameters can be configured and data can be visualized through an interactive view.

The system was tested in collaboration with cardiology department of Hospital de Santa Marta – CHLC. Tests consisted in utilizing the system with two devices, a BITalino and a Samsung Gear S3 smartwatch, to monitor patient's heart rate and physical activity, being patient's comfort and system's scalability important factors in device choice.

The conducted tests allowed to get feedback from all intended users: patients, physicians and medical staff. They all reported the system as being easy to interact with and very useful in getting medically relevant data, with the easy inclusion of new sensors as a major advantage.

**Keywords:** Personalized Medicine, Pervasive Monitoring, Wearable sensors, Smart devices, Bio-signals

# Contents

# List of Tables

# List of Figures

# Acronyms

**ACC** Accelerometry

**bpm** Beats Per Minute

**BSN** Body Sensor Network

**QRS** ECG wave QRS complex

**BT** Bluetooth

**BVP** Blood Volume Pulse

**ECG** electrocardiogram

**FFT** Fast Fourier Transform

**GSM** Global System for Mobile Communications

**HR** Heart Rate

**HSM** Hospital de Santa Marta - CHLC

**INSTICC** Institute for Systems and Technologies of Information, Control and Communication

**LED** Light Emitting Diode

**METs** Metabolic Equivalent of Tasks

**OS** Operative System

**PDA** Personal Digital Assistant

**PPG** photoplethysmography

**RF** Rado Frequency

**RP** R-peaks, the point with largest amplitude in QRS complex

**SD** Secure Digital card

**SNR** Signal to Noise Ratio

**UI** User Interface

**UUID** Universally Unique Identifier

# 1. Introduction

This work describes a novel system, designed to pervasively monitor patients for long periods of time in and outside of the hospital. The system collects medically relevant data and display it in real time through a web interface. This allows medical teams to permanently monitor patients and remotely access real-time and past values of the desired variables. This allows to detect patterns associated with certain diseases and monitor their progression or even check medication effectiveness. The system is designed to require as least maintenance as possible and, apart from charging the device's batteries, it can operate for up to two months without intervention.

The proposed system is distinct from others used in medical practice (like Holter monitors for example) as it is designed to be used for long periods of time outside of hospital environment and it is completely versatile concerning the measured variables. In fact versatility is one of the main aspects of the implemented architecture. The system is prepared to deal with any sensor that provides Bluetooth connectivity with minimal implementation cost. This is accomplished using a smartphone as a mobile hub for data collection, centralizing the information from the patient's designated sensors. The smartphone stores the information incoming from the sensors and, if necessary, processes it to produce more informative measures. Information is then relayed to a central server where it is permanently stored and displayed when required through a web interface. Physicians can specify which sensors should be active with each patient and which are the relevant variables to be measured and displayed. Besides data visualization, it is also possible to configure alarms and receive a notification when a certain event occurs e.g. heart rate is below 50Beats Per Minute (bpm) for more than 5 minutes.

A previous project [1] served as base for the development of the currently proposed system. Although all system components were modified, updated or replaced, the software already implemented for this project served as a base for the currently proposed system.

The system was tested on patients with cardiac diseases from Hospital de Santa Marta - CHLC (HSM) in Lisbon and the main variables collected were Heart Rate (HR) and Metabolic Equivalent of Tasks (METs) [2]. These variables were indicated by the hospital's cardiology team as being common variables used in their practice to diagnose and track patients [3, 4]. Variables to be collected determined the type of sensors needed. In particular, heart rate can be used as a major clinical indicator for patients with heart diseases [5, 6, 7].

To match requirements or the test conditions two sensors were used, a photoplethysmography (PPG) sensor in a smartwatch allows to estimate HR and a tri-axial accelerometer in the smartphone allows to estimate METs. An additional device based on BITalino was used to collect both ECG and Accelerometry (ACC) from which HR and METs were also calculated. BITalino was chosen for being a very versatile and modular tool, with the ability of acquiring ECG with quality similar to a certified medical device [8, 9] which is a more informative signal that PPG.

## 1.1 Motivation

In recent years, pervasive monitoring and personalized medicine, are becoming two of the most common words when describing the future health-care. Medical teams and patients are becoming increasingly eager to have better and more efficient treatments and approaches, that are tailor made to best fit each patient's condition, maximizing health gains and life quality.

Continuous health monitoring is a very promising field of research [10] and to accomplish this, pervasive medical data collection with distributed wearable sensor networks can play an important role. Simultaneously, sensors and smart devices are becoming ubiquitous and is now possible to monitor one's activities and physiological parameters resorting to several types of devices with improving capabilities and decreasing costs. This allows for pervasive monitoring to be a tool in tomorrow's health-care, introducing continuous collection of information about one's symptoms, physiological parameters, activities and even preferences, that can make a difference when diagnosing, treating and tracking the evolution of one's condition.

The evolution of technology also allowed for the development of wearable sensors that can be integrated into the patient's life without causing to much discomfort. In fact wearable sensors, and smartwaches in particular, have been studied many times [11, 12] and even its applicability as a source of clinical information has been proposed [6, 13, 14].

Despite great evolution in technology, the leap into hospital environment has not taken place yet, at least not in a generalized manner. This has to do with the very hard requirements this context puts into devices and also with the lack of systems with simple interfaces, low maintenance and high quality sensors, capable of providing useful information. In this context the proposed system has a good chance of satisfying the needs for many medical applications, as it can interact with a great number of different sensor platforms, with a convenient interface for both medical teams and patients.

## 1.2 Objectives

The main objective of this work is to develop and test a monitoring system that can be used in hospital environment and also with patients in ambulatory treatment, with the particularity of being as generic as possible regarding the sensors and variables it can collect. This means that the same system, being able to interact with (almost) any sensors, could be used in an enormous variety of situations with minimal implementation cost.

Particularly, the expected result of this thesis is to have a functioning pervasive and long term monitoring system and also to conduct tests in hospital context for which it was developed.

Testing, ideally, would cover system functioning but also the reliability of the data coming from the sensors chosen for the test context, which will be the cardiology department of HSM. System functioning tests include the reliability of data acquisition, storing and processing, battery duration for each device and also more subjective aspects like ease of use by physicians, patients and medical teams.

In an ideal scenario, if the tests all have positive outcomes, the system should be implemented for regular use as a diagnose and follow up tool in the cardiology department of HSM.

## 1.3 Thesis Outline

This first chapter introduces the topic and establishes the main objectives of this work.

In chapter 2 an overview of the state-of-the-art is made, and bibliographic base for methods used are presented and justified.

The description of all the features implemented and the functioning of the entire system are presented in chapter 3.

Algorithms and data management methods used are described in chapter 4.

Testing procedures with their results are presented in chapter 5.

Finally, general comments on the results obtained and the future developments that can take place are presented in chapter 6.

Appendix A contains the user guides elaborated to serve as a reference for medical teams when syestem tests started.

In appendix B technical details are provided on the implemented communication protocol between the central server and the smartphone.

Appendix C refers to a submitted paper analyzing the accuracy of smartwatche's HR estimation.

**Anexar paper? (anexo C)**

# 2. Background

During the course of this work, several topics were studied in order to achieve a working and reliable system capable of producing useful data. The main part of the implementation was tied with the system architecture and signal processing to ensure no data was lost and informative measures were retrieved from the data.

As the test phase occurred in the cardiology department of HSM, a significant part of the work was dedicated to create support for the chosen sensors and processing the data into variables known no medical teams e.g. extracting METs [2] from accelerometry signal.

## 2.1 Remote Monitoring

Collecting data from patients using very diverse systems and architectures is an evermore common and feasible practice. In Varshney [15] some of the most common approaches for the design and architecture of this type of systems are presented. Technology now offers a variety of technologies that can be put to use, every one of them with their own set of advantages and setbacks, most of them related with cost, coverage and ease of implementation. From the various approaches, the most commonly used and versatile is utilizing Global System for Mobile Communications (GSM) connectivity between remote sensors and a central server, this is a popular choice as it provides coverage in almost every place and is easily accessible using ubiquitous ”smart” devices. Varshney [15] also mentions some of the general requirements of this type of systems and the main challenges still present when developing such a system. The currently proposed system covers most of the mentioned requirements, like data reliability and ”comprehensive health monitoring” and solves some of the main challenges pointed like scalability, versatility and the delivery of processed and useful information to the medical teams in an easy to use interface and format.

Patel et al. [16] presents a very complete overview of the wearable-based health monitoring systems, presenting several examples of systems and situations where this kind of setups can be useful. The main technological leaps that made the implementation of these apparatus are also presented, with miniaturization of computation and the cost-reduction of smart electronic devices playing a major part. Also Rawassizadeh et al. [11] talks about the main advantages and capabilities smart-wearables, with particular highlight to smartwaches. They have a seamless integration into one’s life, being comfortable and as they are in almost constant contact with the wearer they can pervasively collect data without any disturbance of the subject’s normal life.

There are, in the literature, some examples of systems that implement pervasive monitoring, for example the ones proposed by Atallah et al. [17] and Lo and Yang [18]. Most of these systems use wireless technology to perform data transmissions from a custom hardware device to a central storage. However, very few of this systems tackle practical implementation problems like scalability, integration of new sensors and even user interfaces for medical teams.

In Aziz et al. [19] and Lo et al. [20] a pervasive data acquisition system is utilized with the intent of monitoring post-operative surgical patient. This system is detailed by Lo and Yang [18]. The architecture implemented is quite similar to the one utilized in the currently proposed system. Custom hardware (nodes) was designed to acquire bio-signals and a Personal Digital Assistant (PDA) is utilized to relay information between the Body Sensor Network (BSN) and a central server. It is also mentioned the possible use of various methods for energy consumption optimization, like different Rado Frequency (RF) communication methods and energy scavenging.

Other studies propose systems that are quite comparable to the presented here. Atallah et al. [17] proposes a system with many similarities, it is versatile and includes many different sensors with the same architecture. The main point of distinction is the medical personnel oriented interface that makes the system more configurable as the medical teams can choose which sensors to use with each patient.

## 2.2 Privacy

When dealing with patients' data, privacy is always a must-have concern. In Farrell and Tschofenig [21] and Li et al. [22] remarks are made concerning what procedures and practices must be in place to protect data's integrity and security. The main aspects to consider are data transmission encryption, access control, data anonymity and permission control. This aspects were a major concern when developing the proposed system as described in section 3.4.

## 2.3 ECG

ECG is the recorded electrical activity of the heart muscle. This recording can be made with electrodes in a variety of standardized positions within the body as depicted in fig. 2.2. Each electrode location will reflect in different signal morphologies that may contain different information and even contain clues to different pathologies. The most commonly viewed morphology is depicted in fig. 2.1.

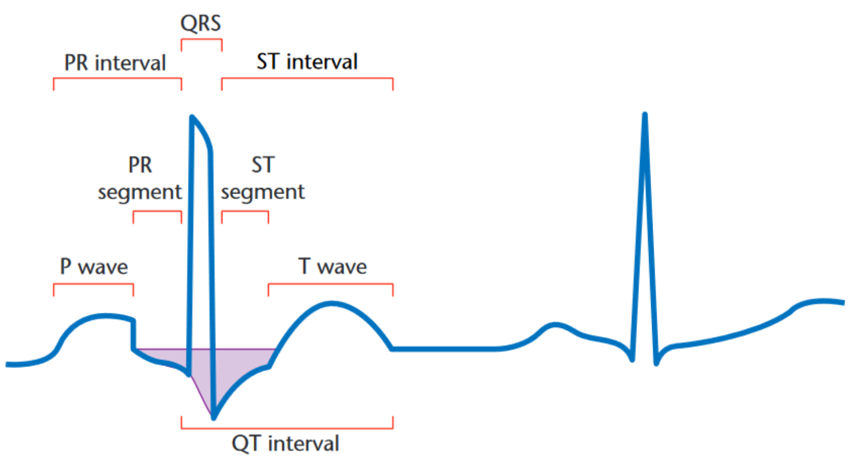



Figure 2.1: De Luna et al. [23] "ECG morphology recorded in a lead facing the left ventricular free wall showing the different waves and intervals. Shading, atrial repolarization wave."

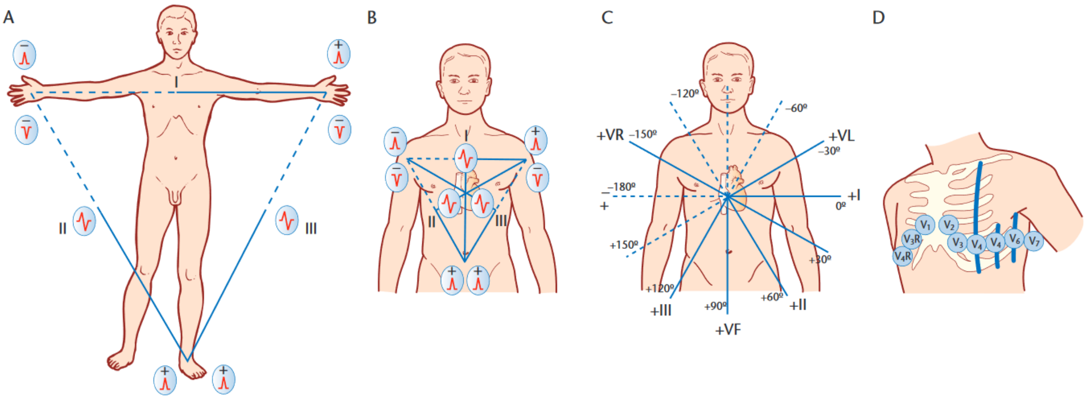



Figure 2.2: De Luna et al. [23] "(A) Einthoven's triangle. (B) Einthoven's triangle superimposed on a human thorax. Note the positive (continuous line) and negative (dotted line) part of each lead. (C) Bailey's hexaxial system. (D) Sites where positive poles of the six precordial leads are located."

The test conditions of the system required the ECG signal to be processed in order to estimate the HR. As described in chapter 4 this processing had to be made in two different situations, in real-time with limited computational resources and off-line without any computational restrictions and with maximum precision.

For the first situation the method proposed by Pan and Tompkins [24] served as a base to build the used algorithm. Although the publication having several years, the method it proposes fulfills all the requirements the test conditions required. Being based on amplitude thresholding it runs in O(N) [25] with very few calculations needed. In the literature it is possible to find similar algorithms to the one proposed, as in Lourenço et al. [26], Pan and Tompkins [27], Ruha et al. [28] with several more summarized in Elgendi et al. [29]. Although these algorithms may perform slightly better, they still require much more computations over the entire length of the signal window, making them less fitted for the intended application.

Lourenço et al. [26] summarizes the most popular algorithms to process the ECG wave QRS complex (QRS) and tries to determine which is the best one to process finger-based ECG, proposing a novel algorithm that is put against all others it mentions. The proposed algorithm is a real-time and very simple set of rules based on amplitude thresholding, similar to the one utilized in the present work.

In the algorithm proposed by Pan and Tompkins [27], several analog and digital filtering stages were implemented including bandpass and notch filters. The detection itself is made using and a convolution operation with a template QRS complex. Again, this approach presented a computational overhead t large to be feasible.

For the off-line HR calculation, where precision was a requisite, and there were no limitations on calculation time and resources, the algorithm proposed by Engelse and Zeelenberg [30] was the one chosen. This is a well established algorithm and although many more were proposed to do the same thing e.g. the one by Elgendi et al. [29], this one remains a good choice for processing ECG signal as it was one of best rated algorithms in the comparison made by Friesen et al. [31].

In the present work, the ECG sensor used was BITalino. It is a very customizable sensor platform with high quality ECG signal as demonstrated by Batista et al. [8] and Guerreiro et al. [9]. It is very convenient as it is a commercial product already including Bluetooth (BT) communication.

There are many possible variations in ECG acquisition setups regarding the number of electrodes (leads) and the type of these electrodes. According with Chi et al. [32], there are four main types of electrodes: dry, gelled, insulated and non-contact. This aspect influences the quality of the signal, resistance to noise introduction and motion artifacts, but most of all the choice of electrodes has major design implications. Although gelled electrodes ensure the best power transfer to the sensor, they tend to be glued to the skin with some kind of adhesive. Although this is suitable for short term acquisitions, it can become a source of discomfort for long-term acquisitions and is not suitable for wearable inclusion. Regarding the dry and insulated electrodes, they are usually not adherent to the skin, making them suitable for long-term acquisitions or where ease in placing and removing the electrodes is desired. Finally the non-contact electrodes provide the most versatile range of applications and can even be placed over cloth.

In Park et al. [33] an ECG sensor embedded in a t-shirt with insulated electrodes is presented, illustrating the easy integration of this sensors in the daily life of the person being monitored. This may present a major advantage regarding the adoption of systems based on this type of sensor.

Another approach is to embedd dry electrodes in a chest band. This is a popular method and was utilized by Braecklein et al. [34] in a very similar manner as in the present work. Dry electrodes are placed in a chest band with BT connectivity, sending data to a central monitor.

## 2.4 PPG

PPG acquisition is a way of estimating the Blood Volume Pulse (BVP), that can be used, among other, to determine the HR and, in certain conditions, the blood oxygen saturation as explained by Allen [35]. For oximetry, two Light Emitting Diode (LED) of different wavelength are necessary, whereas for HR determination a single LED is enough.

Traditionally PPG devices were placed in the fingers or ear lobes, however, in recent years, with the advent of wearables and smartwaches, there are many devices with wrist-worn PPG sensors and other locations as presented by Tamura et al. [36]. Although many locations for the sensors have been proposed, wrist-worn sensors are the ones that present less inconveniences and are less prone to disturb daily life activities, as it is desired for pervasive monitoring.

There are authors that have built and designed their own custom hardware with many variations to perform PPG measurement. As an example, Garbarino et al. [37] describes a wrist band designed to collect several bio-signals, including PPG used for HR determination. In addition, many commercially available products nowadays, include this type of sensor to appeal to customers who want to monitor themselves during daily life or even specifically during sports activities. This devices are evermore common and, some of them, have a good HR estimation capability as described by Wallen et al. [38].

Despite great potential, wearables have some drawbacks that greatly affect their usability. Smartwaches, in particular, position the PPG sensor in the distal portion of the posterior forearm. According with Lee et al. [39], this is a location where PPG signal is faint due to reduced concentration of blood flow. Another major problem is signal corruption by motion artifacts as these devices tend to be heavy enough to have their own dynamics i.e. they move independently from the forearm by inertia. In addition, it is not comfortable to have the device too tight to the skin, and excessive pressure reduces superficial blood flow, thus further reducing Signal to Noise Ratio (SNR). In fact smartwaches and wearable sensors have been studied many times, for example by Rawassizadeh et al. [11] or Bieber et al. [12] and even its applicability as a source of clinical information has been proposed by some authors like El-Amrawy and Nounou [13], Carpenter and Frontera [14] and Allen [6]. However, the accuracy of this type of devices has been questioned and error margins for HR estimates produced by smartwaches were proven considerable for some devices by authors like El-Amrawy and Nounou [13], Parak and Korhonen [40] or Dooley et al. [41].

As presented by Tamura et al. [36], the use of green LEDs in PPG sensors is an evermore popular tendency and is the case of the sensor utilized in the currently proposed system. However this wavelength is not the ideal for this task as it is absorbed by tissues in a greater proportion than other wavelengths, making it capable of measuring only superficial blood flow. Despite the grater absorption of green light, this wavelength is, in fact, one of the the most absorbed by oxyhaemoglobin and deoxyhaemoglobin when compared to the usual red and infrared wavelengths, so it is possible to obtain better SNR although at shorter depths.

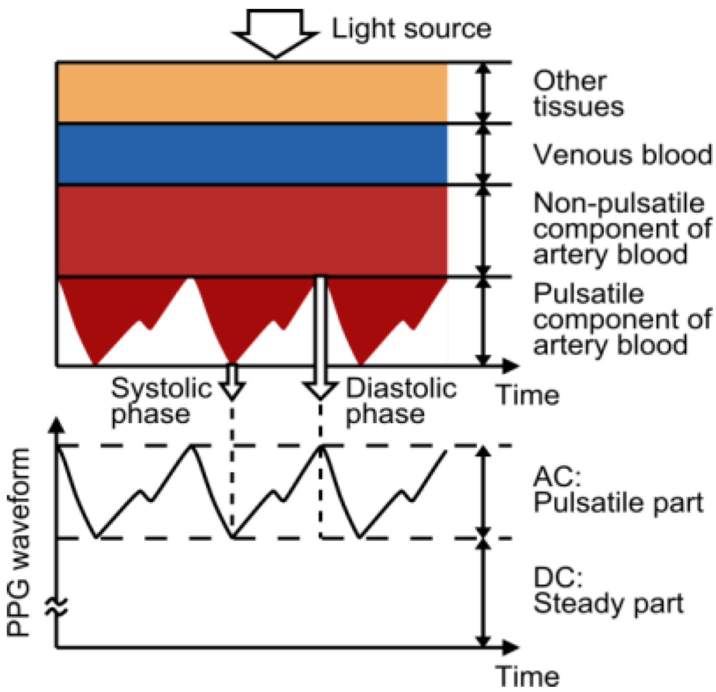



Figure 2.3: Tamura et al. [36] "Variation in light attenuation by tissue."

As depicted in fig. 2.4, there are two main types of PPG sensor, detecting transmitted and reflected light. Each of these types of sensors has their own set of positive and negative aspects. Transmission sensors have placing limitations, as not every part of the body is suitable for light to be transmitted in a practical way. This limitation makes reflection sensors the appropriate ones for wearable devices. However, reflection sensors are more sensitive to noise, as motion artifacts, mainly, can ruin the signal.

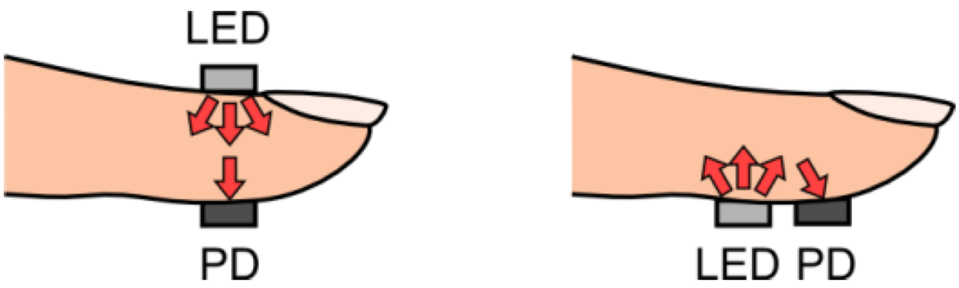



Figure 2.4: Tamura et al. [36] " Light-emitting diode (LED) and photodetector (PD) placement for transmission- and reflectance-mode photoplethysmography (PPG)."

To deal with this difficulty several algorithmic approaches have been developed.

As motion artifacts are one of the most prominent noise sources, most algorithm are designed to address it specifically, and not random noise like in other applications.

The simplest approach to PPG signal processing is the detection of signal features like the systolic peak, onset or others. This type of approaches were compared by Blazek and Lee [42] and evaluated on the performance in detecting onset, systolic peak or dicrotic notch of the PPG signal. Although this is, theoretically, enough to determine the HR from the PPG signal, in practice, this methods are very prone to error when noise is present in the signal. As this family of methods highly depends on the morphology of the signal, with the introduction of noise and motion artifacts the detection error increases very fast.

One of the most common approaches to motion artifact removal is adaptive filtering, as described by Tamura et al. [36]. There are many techniques based on this principle, that most frequently sets on the assumption of interference between the measured PPG signal and accelerations, usually measured with an accelerometer positioned in the same device as the PPG sensor. This interference can be modeled in many different ways, but most authors describe it as being a linear combination, many times simply additive i.e. the captured PPG signal is a linear combination of the true PPG signal and a linear transformation of acceleration. This was the model used by Wood and Asada [43] to decorrelate ACC signal from the acquired PPG signal. In addition, the authors included one step of representing the signal with Laguerre coefficients before applying the adaptive filtering, this allows for a reduction of dimensionality, thus, decreasing the computational burden of the algorithm, while increasing its performance.

A different approach used by Zhang [44] and Zhang et al. [45] is to look to the spectrum of the signal and reconstruct it using carefully chosen Fast Fourier Transform (FFT) segments. This works well even when there is in-band ACC noise i.e. when the spectral power of the motion artifacts is non-neglectable in the same regions as the relevant bandwidth PPG signal components.

## 2.5 Physical activity

Physical activity estimation can be useful in a variety of scenarios, from sports performance tracking, to patient recovery monitoring. Although a method to take this measurement may, intuitively, not be obvious, there are a few different ways of doing so. The main methods used are questionnaires, GPS monitoring and the extraction of indicators from ACC signals. All of them have downsides, and may not be suitable to cover all situations. Intuitively it is easy to understand that self-reported physical activity level is very prone to error. On the other hand GPS quantification is only suitable for activities implying great dislocations, as the GPS is not suitable for exercises in a gym environment for example. Finally accelerometry has a considerable disadvantage as the location of the sensors on the body conditions the type of activities it can successfully record, and most type having a full body monitoring is not practical nor possible.

ACC sensors are used to measure the acceleration they are subjected to, and when incorporated into a wearable device they can be used to detect movements and even estimate the physical activity level of the wearer as it was made by Troiano et al. [46]. In this study, the purpose was to quantify the physical activity for different age groups. For this, subjects were asked to wear the measuring devices for 7 days and quantification was made using METs. This is a popular way to quantify physical activity and it can be calculated from ACC data as defined by Crouter et al. [2].

Another approach to the physical activity analysis is trying to identify the activity being performed, in spite of trying to quantify its intensity. Many authors proposed methods for doing this type of activity recognition. Chen et al. [47] presented a review of current methods for accomplishing this task. Methods are usually divided into similar stages with the collection of data, that is preprocessed, and then fed into some type of classifier producing the identification. Alongside algorithms to process ACC data, also video capture is a popular approach, although, only suitable for studying subjects in controlled environments, as opposed to daily life monitoring.

## 2.6 Data compression

Regarding data compression for storing and relaying, some requirements were established, mainly related with performance and complexity of algorithms. Concerning the computational burden, the algorithms used had to be as simple and efficient as possible to reduce resource expenditure whereas for the algorithms themselves, they had to be lossless and it should be possible to compress segments of the signal, instead of having an algorithm compress the whole data.

There are many types of data compression algorithms, and the majority exploits some statistical property to find a smaller representation of the data. Although some algorithms are built to deal with a particular format of data, as the ones in Mamaghanian et al. [48] and **?** ] for example. However to maintain the versatility of the proposed system, the chosen algorithms to perform this task must be completely general on the signals to be compressed. For this reason, all data is seen as text strings, and like so, when compressing them as text, no data is lost and statistical properties become clearer when compared with numerical ones i.e. to represent numbers as strings only 12 symbols are needed (0-9, "-" and ".") eliminating the need to cover the range of each signal's value. At the same time, memory is limited and data size is relatively small, further limiting algorithm choice.

In Bell et al. [49] many algorithms are presented to perform data compression. However when taking into account the required utmost simplicity, Huffman static codding and time differencing were the chosen alternative.

# 3. System Description

## 3.1 Overview

The implemented data acquisition system was designed to collect medically relevant data and display it in real time through a web interface. This allows medical teams to permanently monitor patients and access patient's information remotely, including real time data and also the history of past values, allowing to detect patterns associated with certain diseases and monitor the progression or even check medication effectiveness. Being completely mobile the system allows for a constant monitoring of hospitalized and non-hospitalized patients for periods that can go up to two months without the need for any maintenance, except for charging devices' battery obviously. The system in prepared to deal with any numerical variables that can be collected with any sensor that has a BT interface. This makes for a versatile completely in what sensors it can interact with and, thus what variables it can acquire, which make it suitable for almost all situations where pervasive long term monitoring can be useful.

Incoming data from sensors is collected into a smartphone that sends it, via web, to a central server that stores the data and serves the web interface. Physicians can specify which sensors should be active with each patient and which are the relevant variables to be acquired and displayed.

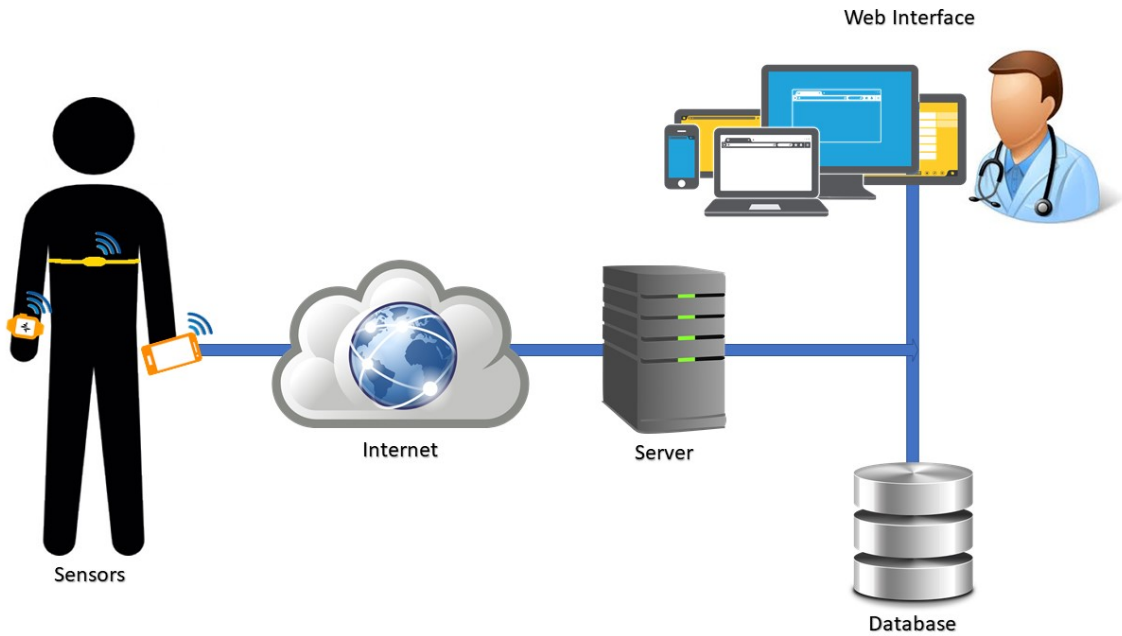



Figure 3.1: System's architecture. Bluetooth connectivity between wearable devices. Smartphone connects to server through mobile web connection.

The system works by having a physician configure all the acquisition parameters relevant for a certain patient and after this step no further intervention is needed, apart from equipping the patient and turning devices on. Parameters that can be configured are:

- The type of devices currently available in the system (smartphone, smartwaches, etc...)

- What devices are to be given to the patient;
- What variables are to be collected from each device, and with what frequency they should be sent to the central server;
- The variables that are expected to be received by the server (that may not be the ones collected, some processing can occur) and how they should be visualized (color, range, etc...)
- Several alarms can be set defining what variable should be in what range for how long to trigger the alarm e.g. heart rate is below 50bpm for more than 5 minutes;

This information is saved into the database. After the smartphone is turned on, information is retrieved from the server and acquisition starts. Until the end of the study data is continuously sent to the server where the physician can analyze it in real time (which is actually a 30s delay that is irrelevant for clinical practice.)

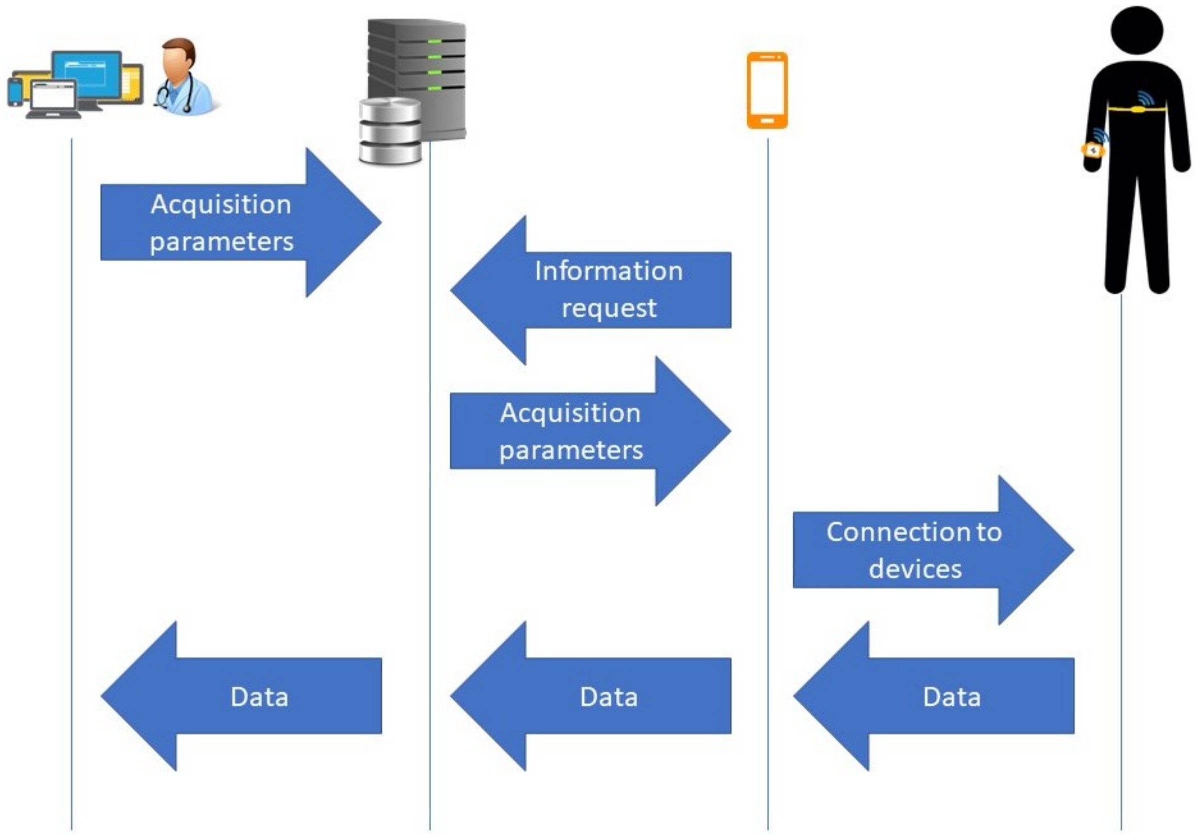



Figure 3.2: System's data flow overview. Bluetooth connectivity between wearable devices. Smartphone connects to server through mobile web connection.

Figure 3.1 illustrates the general system architecture, where the local sensor network connects to a smartphone that relays information through web to a central server that manages the database and the web interface accessible to medical teams. Figure 3.2 depicts the timeline of communications between devices. Initially the user specifies the parameters in the interface that are stored in the database. When devices are turned on the smartphone pings the server which answers with the user configurations so it can start the communication with the configured sensors. After this the flow becomes mostly unidirectional with data coming from the sensors being relayed by the smartphone (eventually with some processing happening before) to the server then to saved into the database and displayed in the user interface when required.

## 3.2 Devices

### 3.2.1 Smartphone

The smartphone is used to centralize data from the various sensors and as a local storage and processing unit. It receives all the raw data collected from the connected sensors, stores it into an Secure Digital card (SD) memory card, applies the necessary algorithms to extract information from the raw signals and finally sends this computed quantities to a central server so they can be visualized through a web interface.

All the sensors' data is transmitted through a Bluetooth interface and is stored as an array so it can be used to make the necessary calculations. Every 10s the data is saved into files stored in an SD memory card. Data to be saved consists of time-series composed by consecutive data samples from the sensors. This samples are acquired at different rates from each sensor and a timestamp is associated to each sample in order to facilitate the time location of each one and make easier to align the signals from different sensors. The app is prepared to deal with any number of sensors per connected device and it stores data in a separate file for every remote device, containing data from all the device's sensors.

#### Android App

The android app was implemented using Android Studio 3.0.1. All functions were designed to be as computationally efficient as possible to minimize time delay and maximize battery life. The app was designed as a replacement of the android home screen, so when the smartphone is turned on, the app will automatically initiate and show on screen, ensuring the patient does not forget to initiate it. Another focal point of the implemented app is the thread-oriented logic that eases future development and new sensors integration. This means that support and communication for each sensor runs in an independent thread, easing the process of introducing support for new remote sensors.

At startup, in the main thread, the app checks for Internet and Bluetooth connectivity, requesting its activation if necessary. A background service is initiated and it will be responsible for all the non-display actions. This was necessary not to crash the display as background tasks can be time consuming and will need to run even if the display is deactivated or if the screen is locked. The background service then starts its task by establishing a socket [50] to the server @learnib.lx.it.pt and sending the smartphone's Universally Unique Identifier (UUID) which is the Wi-Fi mac address. Initially the Bluettoth adapter address was to be used as UUID for the smartphone, but recent versions of Android Operative System (OS) do not allow to programmatically get it so, alternatively, Wi-Fi mac address was used.

Code used to determine smartphone's mac address:

```java
import android.annotation.TargetApi;
import java.net.NetworkInterface;
import android.bluetooth.BluetoothAdapter;
import java.util.List;


@TargetApi(9)
public static String getMacAddr() {
  BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
  if(bluetoothAdapter == null){
    return null;
  }
  String addr = bluetoothAdapter.getAddress();
  try {
    if (addr.equals("02:00:00:00:00:00")) {
      List<NetworkInterface> all = Collections.list(NetworkInterface.getNetworkInterfaces
          ());
      for (NetworkInterface nif : all) {
        if (!nif.getName().equalsIgnoreCase("wlan0")) continue;
        byte[] macBytes = nif.getHardwareAddress();
        if (macBytes == null) {
          return "";
        }
        StringBuilder res1 = new StringBuilder();
        for (byte b : macBytes) {
          res1.append(String.format("%02x", b) + ":");
        }
        if (res1.length() > 0) {
          res1.deleteCharAt(res1.length() - 1);
        }
        Log.d(TAG, "MAC ADRSS: " + res1.toString());
        return res1.toString();
      }
    }
  } catch (Exception ex) { }
  return addr;
}
```

Connection to the server is made using java's native socket implementation:

```java
import java.net.Socket;

private Socket socket;
private OutputStream out;
private InputStream in;
private final String host = "learnbig.lx.it.pt";
private final int port = ********;

```

```
public void connect() {
  try {
    socket = null;
    out = null;
    in = null;
    socket = new Socket();
    socket.connect(new InetSocketAddress(this.host,this.port), 10000);
    out = socket.getOutputStream();
    in = socket.getInputStream();
  }
  catch (Exception e) {}
  catch (Throwable t) {}
}
```

After the socket is connected, the UUID is sent to the server that replies with a message containing the Bluetooth mac addresses of all the remote devices and sensors that the smartphone should connect to. It also sends the information about the frequency at which data should be sent to the server and, if necessary, other device-specific parameters. In case of connection failure the background service sends this information using a previously registered ResultReceiver causing the User Interface (UI) activity to change the background color and send a notification with sound to the user while simultaneously retrying to connect.

As this information is received and parsed, the necessary devices are initialized. A different class is implemented to interact with each type of device, as different communication protocols are required. Each of this classes extends the native Thread class in order to have complete independence between devices. This aspect enhances the versatility of the system, easing the implementation of support for new sensors. This is also needed to avoid clogging the processor and the Bluetooth reading buffers, as events occur asynchronously between devices and timing is relevant to ensure no data is lost.

Device's classes all work in similar ways. They establish a Bluetooth connection and handle the incoming data. As sensor values reach the smartphone, they are stored into buffers holding 10s of data. When buffers are full, the 10s window of data is compressed and saved into a file and the data is processed in order to extract medically relevant indicators like the HR for example which are then displayed in the UI thread. Saving data into the files is accomplished using BufferedWriter and StringBuilder classes for optimized performance [51]. BufferedWriter class uses a non blocking write that allows the device thread to continue his strenuous work without much delay. StringBuilder class is used in order to compose the strings that ought to be saved without the need to allocate new memory blocks with each append as it would be necessary if dealing with Strings. Data is saved into an SD card or to the smartphone memory if the early is not present. All file writing processes are synchronized to protect against a case where two threads are reading or writing to the same file at the same time.

Figure 3.3 illustrates the general architecture of the implemented app, showing the two services with their threads. Background service, as refered before, has a main thread that establishes connection to the server, than a separate thread to interact with each device and finally a thread to save and process data.

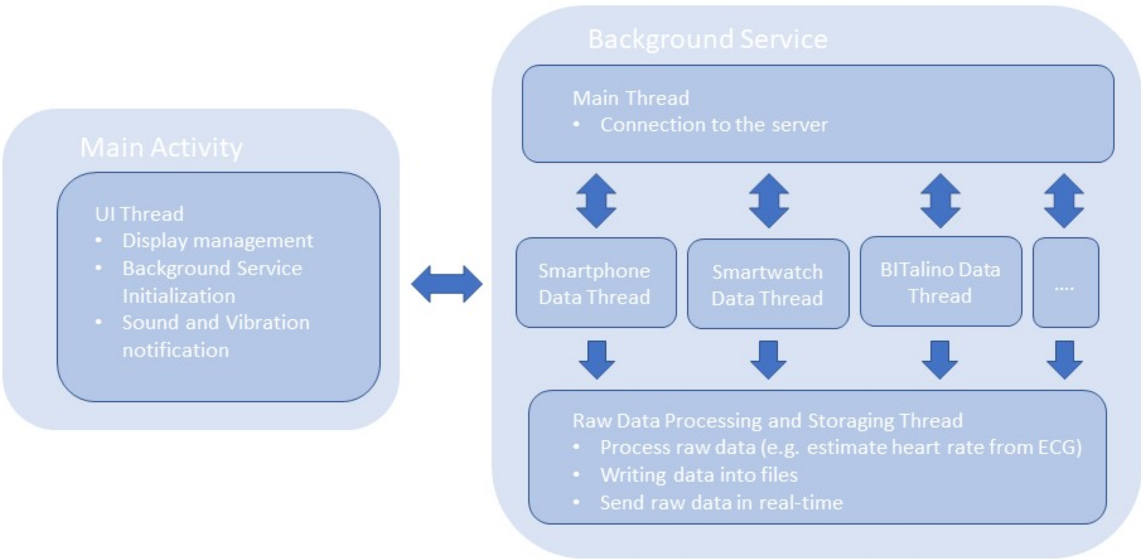



Figure 3.3: Android App threads call-stack

Data is saved to files in a specific format detailed in section 4.3 and the code utilized to save it most efficiently is as follows:

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.*;

private BufferedWriter bw = null;
private File file;

public open_file(Context ctx, String device, String person_ID) {

  context = ctx;
  this.person_ID = person_ID;
  File[] paths = context.getExternalFilesDirs(null);
  File path;
  int n = 0;

  if (paths.length == 2) {
    path = paths[1];
  } else {
    path = paths[0];
  }
```

```java

  file = new File(path, device + "_" + person_ID + ".txt");

  try {
    bw = new BufferedWriter(new FileWriter(file, true));
  } catch (IOException e) {
    e.printStackTrace();
    Log.d(TAG, "Erro erro ao abrir o writer");
    try {
      file.createNewFile();
      bw = new BufferedWriter(new FileWriter(file, true));
    } catch (Exception ex) {
      Log.d(TAG, "Erro erro ao abrir o writer depois de criar o ficheiro");
      e.printStackTrace();
    }

  }

  Log.d(TAG, "Iniciou com sucesso");
}

public void insert_data_list(int[] RAWmeasurementsArray, Long[]  RAWtimestampsArray) {

  StringBuilder str = new StringBuilder();
  int buf = 0;
  Long buf_time = 0L;
  String separator = ";";

  str.append("#");
  str.append(separator);

  for (int i = 0; i < RAWmeasurementsArray.length; i++) {
    str.append(String.valueOf((Long) RAWtimestampsArray[i] - buf_time));
    buf_time = (Long) RAWtimestampsArray[i];
    str.append(separator);
    str.append(String.valueOf(RAWmeasurementsArray[i] - buf));
    buf = RAWmeasurementsArray[i];
  }
  str.append("\n");

  synchronized (bw) {

    try {
      bw.write(str.toString());
    } catch (IOException e) {
      e.printStackTrace();
      Log.d(TAG, "Erro ao escrever ficheiro");
      try {
        bw[this.index] = new BufferedWriter(new FileWriter(files[this.index], true));
```

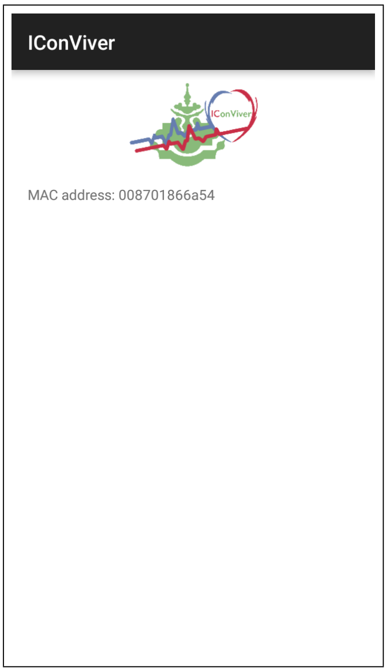
```
      } catch (IOException e1) {
        e1.printStackTrace();
        Log.d(TAG, "Erro ao abrir ficheiro");
      }
    }
  }

  str = new StringBuilder();

  try {
    bw.flush();
  } catch (IOException e) {
    e.printStackTrace();
    try {
      bw = new BufferedWriter(new FileWriter(file, true));
    } catch (IOException e1) {
      e1.printStackTrace();
    }
  }
}
```

Optionally, using the web interface, user can request raw data in real time from a specific or all devices to be sent. This process is accomplished by a separate thread that receives each 10s block of data that is written to files, further compresses and sends it through a socket to the server where data is stored. Further compression is needed as bandwidth and internet plafond are more restrictive than memory space. Data compression and sending occurs in a separate thread to ensure there is no delay in data acquisition.

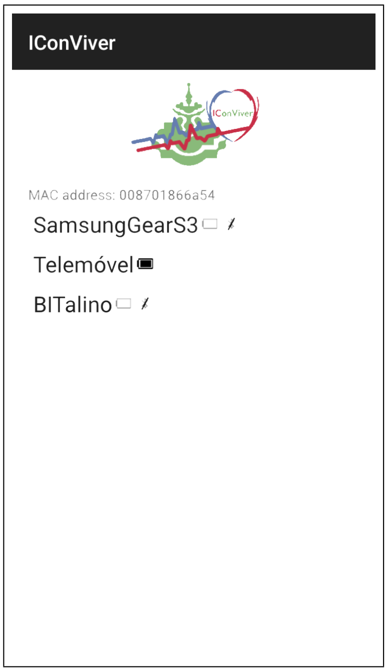
At the same time data is saved into files by each device's thread, the result of processing that 10s window of data is sent to the UI thread to be displayed on screen. These threads also request battery status from the devices. This information is sent to the UI thread where a symbol indicates battery status of each device.

To summarize, raw data coming from the sensors is stored into files and optionally sent to the server. That same data is then processed to extract more relevant and informative measures, that are again sent to the server and shown in the smartphone screen alongside with battery status and Bluetooth connection status. Figure 3.4 shows the main screens of the app with the main stages of initialization (figs. 3.4a to 3.4c) and when connection to the server cannot be made (fig. 3.4d).

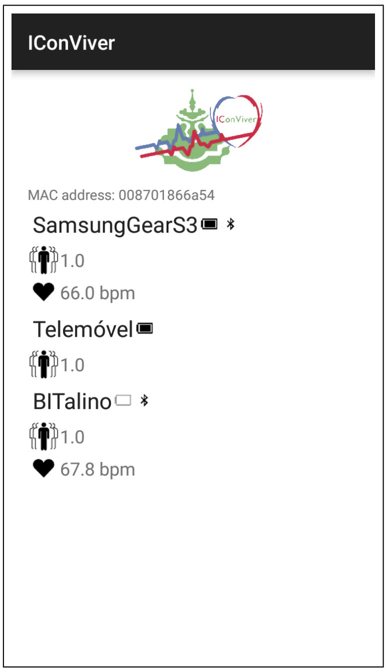
In case any Bluetooth connection to any remote device is lost, the corresponding device's thread will try to reconnect it indefinitely and will trigger a notification with sound and an icon will appear in the app screen informing the user connection is lost. This is useful in case the user is not wearing one of the devices and forgets it. Also a warning message will be sent to the server and a mark will appear in data visualization, informing the medical teams of the momen the device was disconected. All threads and procedures will proceed normally when the lost connection is reestablished. In case the server connection is lost, background service's main thread will also try to reconnect indefinitely, leaving




(a) Display at startup showing device's UUID




(b) Display after information from the server is received. Bluetooth symbols indicate there is still no connection with peripheral devices.




(c) Display after connection has been established and data from devices is received. Below each device's name there is the value resulting from processing the last 10s of data collected.


(d) Screen when it was not possible to establish server connection. A sound notification also occurs in this event.

Figure 3.4: Android App screen in different situations.

all other threads running with no interference. When reconnected, all other threads are stopped and the call-stack is restarted as if the app was just initialized. This occurs in this way to ensure that data acquisition continues even if the patient enters an area with no mobile data communication and, ant the same time, when the communication is reestablished, acquisition restarts to preclude any change in acquisition parameters. When the app cannot communicae with the server, or server is lost, a notification is emmited and the background color of the home screen changes.

### 3.2.2 Server and Web Interface

**Pôr as figuras num anexo em vez de aqui no meio do texto?**

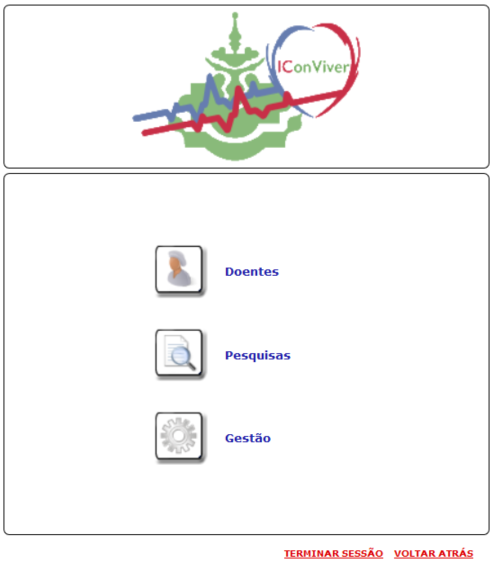



Figure 3.5: Initial panel after login into the system's web interface. Here one can see patient information; search for specific data of a particular patient; configure all device and acquisition parameters.

Software for the server was adapted from software already implemented for a different project [1] by Institute for Systems and Technologies of Information, Control and Communication (INSTICC). All parameters and information to be passed to the remote device can be configured using the web interface and any numerical data coming from it can be shown in an interactive panel as well as warnings and events.

All aspects of the UI were designed to provide a simple and versatile interface that can be used to interact with very different types of variables coming from a multitude of sensors. The main goal was to have a single interface that could be used in various contexts inside and out of hospital applications, and capable of being easily used with many variables and sensors.

The server itself is a computer with windows operative system and an internet connection that acts as a central control unit for the whole system. It is responsible for serving the user web interface, relaying information between the user and the devices and finally store all the information into a database. The interface is based in the .NET framewok with a Windows SQL Database.

System works with three major components:


Figure 3.6: Management screen, where all the systems and device's parameters can be defined.

- **Web interface -** This is an ASP.NET website served using Windows Internet Information Services where user can configure all patient, device and acquisition parameters that are then stored into the database

- **Socket -** This component is in fact the two different subcomponents that ensure the communication between remote devices and the database.

  – A .NET script is constantly waiting for socket connections and sends acquisition parameters to remote devices. It also receives and stores data sent from the devices containing patient processed data like HR and METs values.

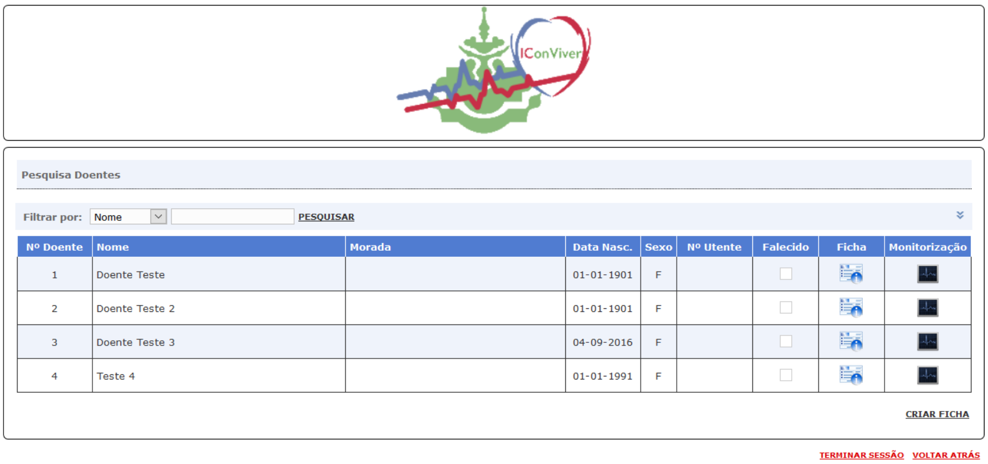


Figure 3.7: Screen showing the patients enrolled in the system. From here one can access all the information and collected data from each patient.

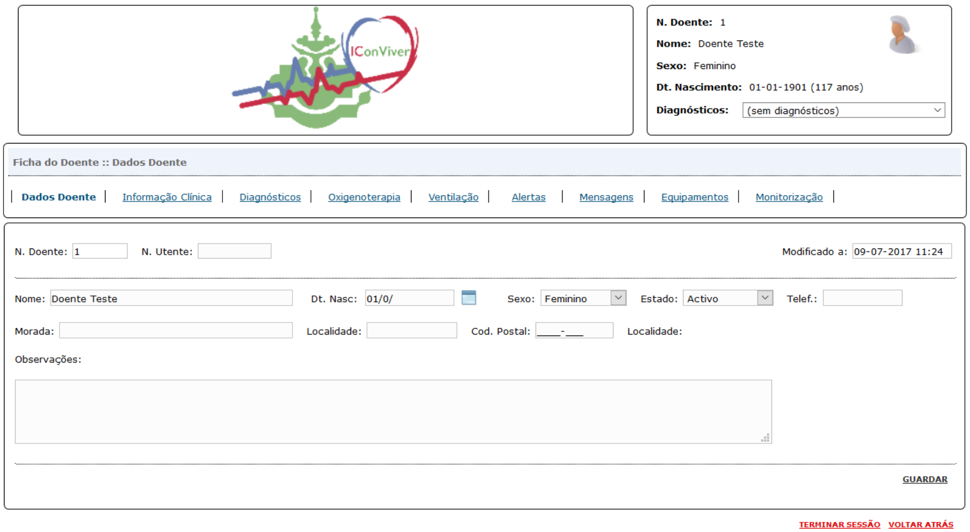


Figure 3.8: Data visualization screen where one can choose the variables to see and their time frame.

– A Python script also constantly waiting for socket connections in a different port. This script is responsible for receiving and storing realtime raw data that may be sent by remote device (only active if raw data is requested).

- **Database -** This component is responsible for receiving and storing all data that comes from all other components.

Two different socket processes were implemented to prevent clogging which could lead to data being lost. Also different languages were used for implementation convenience.

Figures 3.5, 3.6 and 3.8 to 3.10 and **??** show some examples of the implemented web user interface, from where medical teams can control and configure all stages of the patient monitoring. From this interface medical teams can configure and access all information in the system ensuring the system is versatile to deal with many use cases.

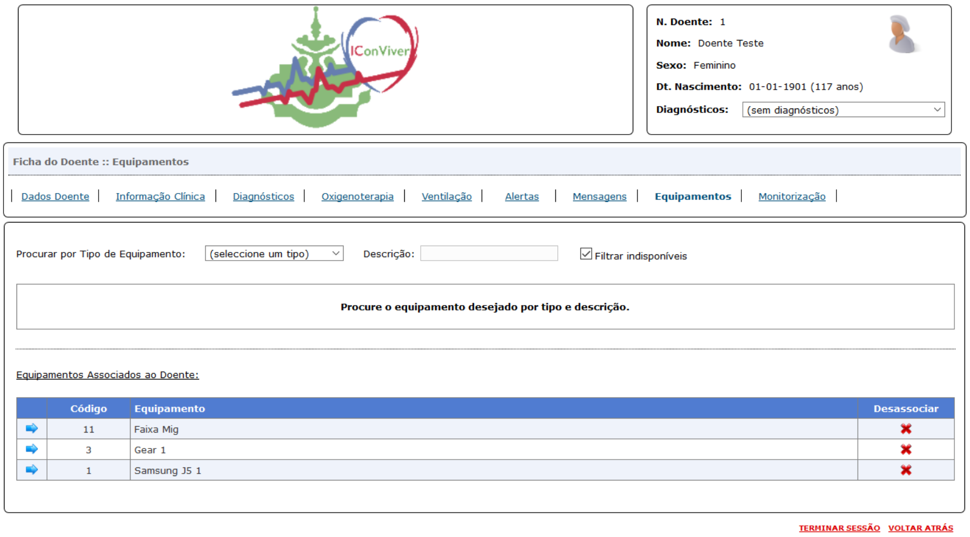



Figure 3.9: Data visualization screen where one can choose the variables to see and their time frame.


Figure 3.10: Data visualization screen where one can choose the variables to see and their time frame.

### Initial Server Configuration

In order to setup the system to start acquisition, some parameters need to be configured in the management (Gestão) screen showed in fig. 3.6 that are accessed from screen in fig. 3.5 after successful login.

Parameters to be configured on system setup:

1. Devices connected to the system are categorized into types like smartphones or chest bands or others. The names of these types of devices are completely configurable, but to prevent mistakes

only one device of each type can be assigned to a patient simultaneously. Each type of devices is identified by a name and single character code e.g. an S for smartphones.

2. For each type of device it is required to define which parameters they need, these parameters must include a key parameter that includes the UUID of the device labeled with the identifier ”I”. Other parameters can be configured but they vary based on the implemented features and support from the Android app, however they tend to be related with what sensors are active in each remote sensors and with what frequency they acquire and transmit data. These parameters are sent to the smartphone when it is turned on and are interpreted or relayed to the remote devices depending on the implemented interactions.

3. It is also required to define what variables are to be displayed in the data visualization screen from fig. 3.10. To do this user must define the name of the variable, its range and the color of the corresponding line in the plot panel. Furthermore user can define intervals of values to see later the percentage of data that lies inside those intervals. This statistics can be seen in left panel of fig. 3.10.

4. Optionally medical teams can configure rules that trigger an notification e.g HR is below 50bpm for more than 2minutes. To configure this rules, one should select a variable, a reference value, a relation between them ($<$, $>$or =) and a duration.

**Acquisition Start**

In order to start data acquisition it is necessary to add patient information in screen showed in fig. 3.8 and add the devices that should be active with that patient in screen of fig. 3.9. After this step, the smartphone and all other devices can be turned on and acquisition starts with data being displayed in panel of fig. 3.10. As data is received from the remote devices, it is displayed and user can chose to see data from a specific time-interval or to see data as it arrives in real time. From the web interface it is also possible to send text messages to the smartphone given to the patient.

All this configuration steps are necessary to ensure compatibility with any sensor medical teams consider adequate, making the system able to deal with any peculiarities each sensor may have.

Steps necessary for rotine aquisition start are presented in appendix A.

## 3.3 Communication Protrocol

Appendix B Details this topic.

When the smartphone is turned on, the application starts its execution and sends an initial message to the server containing a timestamp and its own UUID. The server then responds with n identic messages, where n is the number of remote/wearable sensors. These messages contain the code number of the patient, a timestamp, the UUID of the remote sensor, the type of the remote sensor (smartwatch, smartphone, etc...) and pairs of (name, value) for all the custom parameters that user defined in the configuration. The first and last character of these handshake messages is the special mark ”\”.

After this initial handshake, the smartphone periodically sends data to the server in a similar fashion, the message is composed of a timestamp, the patient code number and pairs of (name, value) for each of the variables that are being sent. For this messages, the first and last character is #.

Other messages are exchanged including alerts and events, each type having a similar structure, timestamp, patient code, and pairs of values and identifications containing the UUID of the relevant device and the respective error or event code. Each type of message has a special character that is repeated in the begining and end of the message.

Regarding the communication between the remote sensors and the smartphone, this is a very variable protocol as it depends on the peculiarities of each sensor utilized, and for this reason no standard protocol is defined.

## 3.4 Privacy

Regarding data transmissions between devices, privacy was also a concern. Having smartphnes connected through mobile data connectivity the chances of intercepting data are smaller when comparing with the hypothesis of having devices connected through wifi. Also Bluetooth connectivity is encrypted, so data coming from the wearable sensors is also secured.

Data is stored in the database in a completely anonymous form i.e. the patient information is stored separately, each patient is assigned a number, and is identified by that number in all other processes. Furthermore, when the smartphone connects to the server and sends data, it is always sent being labeled by only the patient's number.

Using a login system, it is possible to define for every user of the system, if the patient's informations are visible or not, further increasing security, not allowing everyone to access the user interface.

# 4. Data Processing

## 4.1 Off-line Estimation of Heart Rate from ECG Data

In order to obtain a ground truth reference for HR calculated, a chest band collecting ECG was used. This ECG is then processes off-line using an established segmenter [30] that detects the RP and allows to estimate patient's HR. The chest band ensures a good signal quality even when patient is moving as it is elastic and tight to the chest, minimizing the introduction of moment artifacts. Another aspect that improves the signal's quality is the material of the electrodes, this rubber-like material has a certain adhesion to the skin that prevents slippage. This chest band was chosen as it is appropriate for sports tracking devices that monitor athletes' HR.

The HR estimation is accomplished by directly aplying the algorithm described in [30] to the raw ECG collected. Following the application of the mentioned algorithm, the RP position is known and instantaneous HR can be calculated according to eq. (4.1). After this step, to eliminate errors of any wrongly detected R-peak, and to allow comparison with estimates made by other devices, data in sliced into 10s windows and the returned HR value is the median of the instantaneous HR values inside that 10s window. This procedure produces an HR estimate that is calculated with a 10s window which is the sampling frequency used for the other methods of HR estimation used in the system's testing setup.

$$HR(bpm) = \frac{60}{RR_{interval}} = \frac{F_s * 60}{R_{i+1} - R_i} \tag{4.1}$$

Where:

$RR_{\text{interval}}$ → interval between consecutive RP in seconds

$R_{\text{i}}$ → sample number of the ith RP

$F_{\text{s}}$ → ECG sensor's sampling frequency

## 4.2 Real Time Estimation of Heart Rate from ECG Data

Since ECG was being collected, and it is, theoretically, a reliable and very informative signal, a method was designed to extract HR estimates in real time from it.

The proposed algorithm is a real-time ECG segmenter that is simple enough to be implemented in the smartphone without draining too much battery which would harm system's usability. It is based in the segmenter described in [24] and some modifications were implemented, being the final result similar to the algorithm described in [29].

As this algorithm is intended to run in real-time, it must be computationally very light and, thus, the methods used must be relatively simple. With this in mind the following algorithm was designed, running in O(N) [25] with each signal sample being used in computation approximately twice, meaning that the computational burden is reduced to the bare minimum, as other methods require many times this number of computations.

The ECG signal is analyzed in 10s widows and the first step is filtering using a 5-40Hz, 5th order, Butterworth bandpass filter [52] and then an adaptive amplitude thresholding technique is applied to locate RP.

To initialize threshold's value, an histogram i.e. cumulative probability plot, of the signal's amplitude over the first window is taken. The threshold is then initialized to a value corresponding to 98% of this histogram. This value is calculated as an initial approximation of the peaks amplitude on that window as is exemplified in fig. 4.1 and formally defined by eq. (4.2).

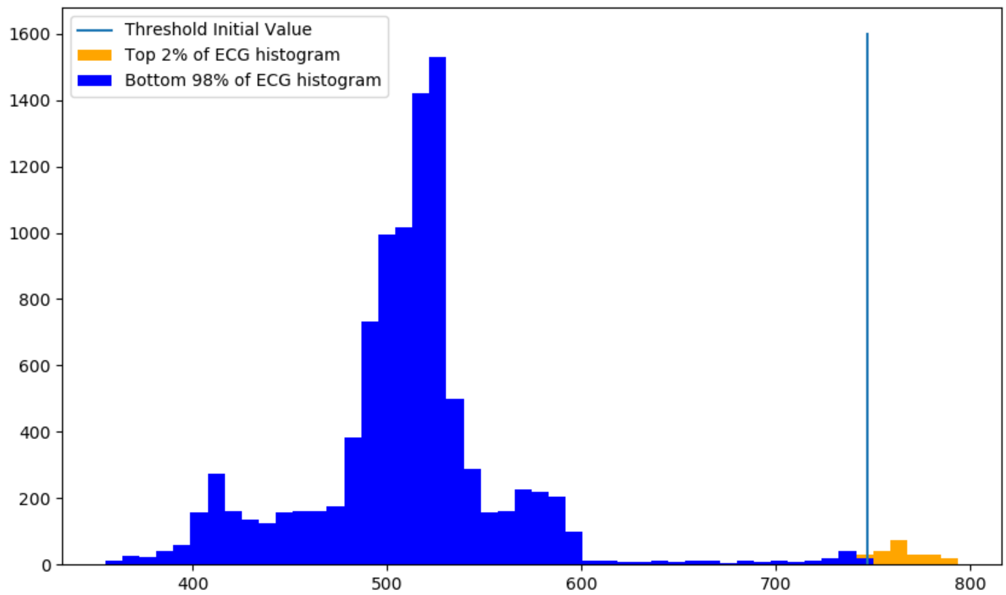



Figure 4.1: Example of ECG amplitude histogram from a 10s window, showing the bottom 98% of data points (blue), the top 2% (orange) and the threshold's initialization value (blue vertical line), in this case 743.

After this, the first local maximum that exceeds the threshold is considered to be an R-Peak. Threshold's value is then dynamically updated. To do this, a running estimate of the peaks amplitude is kept and a running estimate of other peak's amplitude, designated noise peaks, is kept.

Each local maximum is classified as an R-Peak or not if it is either above or below the threshold and is used to update the running estimates according with eq. (4.3)

Initialization:

$$\begin{cases} THR = \arg_x \left( \frac{F_{ECG}(x)}{\lim\limits_{y \to +\infty} (F_{ECG}(y))} = 0.98 \right) \\ SP = THR \\ NP = 0.5 * THR \end{cases} \tag{4.2}$$

Iteration step:

$$
\begin{cases}
SP = 0.125 * P + 0.875 * SP & \text{if } P > THR \\
NP = 0.125 * P + 0.875 * NP & \text{if } P < THR \\
THR = SP - 0.25 * (SP - NP)
\end{cases}
\tag{4.3}
$$

$THR \rightarrow$ decision boundary between R-peaks and noise
$P \rightarrow$ amplitude of the local maximum to be classified
$SP \rightarrow$ running estimate of the peak's amplitude
$NP \rightarrow$ running estimate of noise peaks
$F_{\text{ECG}} \rightarrow$ ECG signal cumulative distribution function

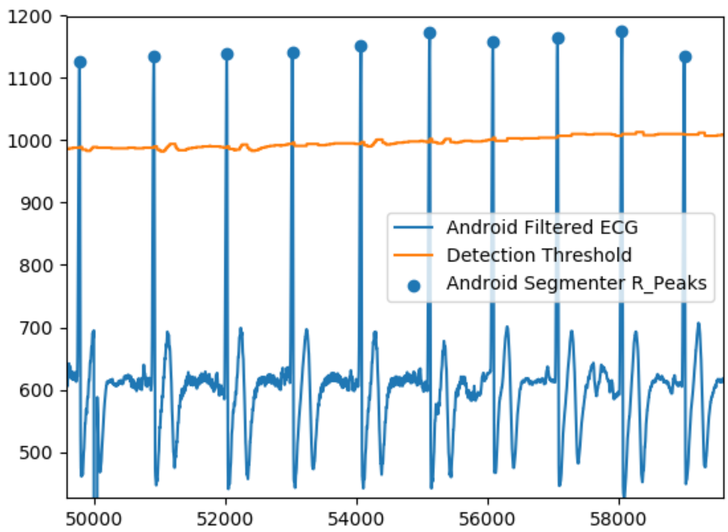



Figure 4.2: Filtered ECG signal (blue line) with the identified RP (blue dots) and the decision threshold (orange line)

If no peaks are found in a 2s window, which would correspond to a HR below 30bpm, the search is backtracked to the start of the 10s signal window being analyzed and threshold's value is lowered to 90% its current value. In an analogous way, if two peaks are found in less than 100ms, which is physiologically impossible [53] the threshold is increased in 10%. These strategies allow to keep up with sudden changes in the ECG amplitude, reducing false positives and missed beats.

Finally each pair of consecutive peaks found in the 10s window is used to produce an estimate of instantaneous HR according to eq. (4.1). To avoid the inclusion of outliers in the final result, due to errors in the R-Peak location, the median of the instantaneous HR values (calculated from each pair of RP) is taken, improving performance. The success of this step lies on the assumption of HR being relatively constant inside the 10 window, allowing to assign a single HR value to the whole 10s period. If this estimate falls outside the 30-220bpm range, the calculation is rejected and a special value of -3bpm is returned. This same value is returned by Gear BVP segmenter in analogous situation.

To test the algorithm, subjects used the system while saving HR calculated by Gear and the proposed segmenter. Calculated HR was then compared with HR determined by state-of-the-art algorithm from the saved raw ECG data. Testing is detailed in section 5.5.

## 4.3 Data Compression

As the memory available in the smartphone is relatively small, some data compression is required to allow long periods of continuous data acquisition. On the other hand, elaborate compression algorithms are not advised once they tend to be computationally heavy, most are not fitted for real-time use and the memory limitations were not big enough to justify the time investment on the implementation of complicated algorithms. Another requirement was to have a lossless compression, this because signal processing to be posteriorly done would be harmed by noise introduction or information loss.

Like so the method chosen to reduce memory usage was to save the derivative of the data, calculated according to eq. (4.4), instead of saving raw values. Data is saved into the files every 10s and the first value of each batch is stored as a raw value to make possible, and faster, the recovery of the original data and thus, ensuring this method is lossless. As an example, to store the vector [516, 515, 515, 515, 514, 514, 514, 514, 513, 514], instead of saving the string ”516;515;515;515;514;514;514;514;513;514” the saved string would be ”516;-1;0;0;-1;0;0;0;-1;1”, reducing the string from 39 to 22 characters, which is a reduction of almost 50% that would be even greater in a larger vector. The first entry of the saved string is the original value and the following ones are the differences of consecutive values.

This method lies on the assumption that sampling frequency is high and so sensor values do not vary much between consecutive samples making the raw values greater than the differences between consecutive samples. Another thing that reduces memory usage is the fact that all values are stored as integers and thus eliminating the need for an extra character to mark the decimal digits. To do this, floating point values are multiplied by 1000, to conserve 3 decimal digits, and are then cast to integers.

To recover a single sample from the file, one must integrate the saved values up to that sample according with eq. (4.5).

$$X_i^{coded} = \begin{cases} X_i^{original} & i = 0 \\ X_i^{original} - X_{i-1}^{original} & i \neq 0 \end{cases} \tag{4.4}$$

$$X_i^{decoded} = \sum_{j=0}^{i} X_j^{coded} \tag{4.5}$$

$X_i^{coded}$ $\rightarrow$ ith sample to be written to file
$X_i^{original}$ $\rightarrow$ ith sensor’s sample
$X_i^{decoded}$ $\rightarrow$ ith sensor’s sample after decompression

As a way of easing the time location of the data, each sample is saved with the timestamp at which it was collected. This timestamp is also subjected to the time-differencing operation. To ease the parsing of the saved files, a special character ”#” marks the the first sample of each 10s window. This line has to be clearly identified as it marks the start of a new window ant the saved values are the true sensor values that should be used to retrieve the following samples, until the next ”#” character is found.

An example of data from a 3-axial accelerometer saved in the described format would be:

```
#;1508325806082;8;268;5418
10;0;70;1774
10;-10;63;1129
10;303;47;699
10;-302;27;353
...
```

The same data stored in the raw data format:

```
1508325806082;8;268;5418
1508325806092;8;338;6592
1508325806102;-2;401;7721
1508325806112;301;448;8420
1508325806122;-1;475;8773
...
```

When realtime raw data is being sent through web, further compression is needed. However efficiency is still a major priority as battery life is very much affected by algorithms’ complexity. With this in mind a simple Huffman codding [54] was implemented using a static code for all subjects as statistical properties of data are mostly invariant with subject (although they are not invariant with the sensors used, so for different sensors a new code must be generated to ensure efficiency). This was concluded when determining Huffman code for different subjects in different activities and having very similar codes for all (only least common characters had some minor changes). This analysis made for the test case of the system described in chapter 5. This codding is performed after time-difference operation to encode each digit of the values obtained from the first step.

The encoding of digits and not the values themselves had to do with code length as the number of characters to be encoded is rather small, 10 digits plus 3 support characters (”-”, ”;” and ”\n”), whereas the number of values in the range of the sensors is of several hundreds, so the length of the code to represent each value would be much larger then even the 8bit per digit representation. This is accomplished looking at values to be saved as if they were text rather than numerical values.

With the character count showed in fig. 4.3 the following code was determined:

```
';' -> '0'
'0' -> '10'
'1' -> '1111'
'-' -> '1101'
'\n' -> '1100'
```

```
'2' -> '111011'
'3' -> '111000'
'4' -> '1110011'
'5' -> '1110010'
'6' -> '11101011'
'7' -> '11101000'
'8' -> '111010101'
'9' -> '11101001'
'#' -> '1110101001'
'.' -> '1110101000'
```

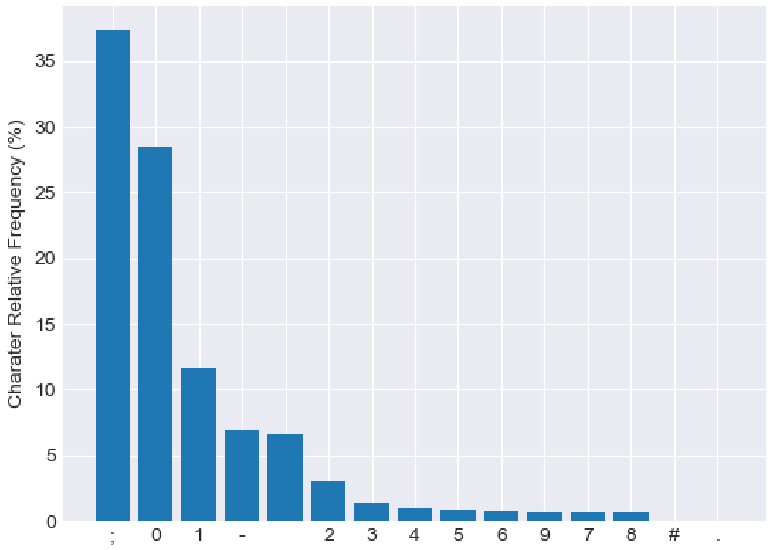



Figure 4.3: Character relative frequency in all the data collected from 4 different subjects.

Using this technique, average character length went from 8bit to just 2.6bit, which corresponds to a compression ratio of 3 times smaller files.

To evaluate algorithm's performance, memory usage was measured by having the system acquire continuously for 48h and then estimating the amount of data stored per day. Table 4.1 depicts the results of this experiment and it is easy to see that even with this extremely simple technique, the size of the data to be saved has been greatly reduced.

$$CompressionRatio = \frac{UncompressedSize}{CompressedSize} \tag{4.6}$$

Taking into consideration the algorithm's simplicity the compression rate obtained was very good. In fact when looking at proposed algorithms with a comparable objective [55], the method used has a performance just slightly worse, having a much lighter computational burden.

The smartphone is also responsible to send the relevant values to a central server where the data

Table 4.1: Memory usage when data is stored as a raw sensor value and as differences. Compression ratios calculated according to eq. (4.6)

| Device | Memory Usage (MB/day) | | |
|---|---|---|---|
| | **Raw Values** | **Time difference** | **Huffman** |
| Smartwatch | 345.2 | 145.4 | 71.3 |
| BITalino | 3037.3 | 574.3 | 355.7 |
| Smartphone | 253.3 | 99.6 | 54.6 |
| **TOTAL** | **3635.8** | **818.9** | **481.6** |
| **Total Compression Ratio** | — | **4.43** | **7.55** |

can be visualized and permanently stored. Communication with the server is made using GPRS/3G technology and accomplished by a socket that allows the transmission of information. Due to restrictions in the amount of data that can be sent between the server and the smartphone, only processed and more informative values are sent and not the raw data, e.g. the ECG is measured, but the HR is sent as it is easier to interpret and has a much smaller sampling rate, reducing enormously the volume of data to be sent.

## 4.4 Heart Rate Estimation from Photoplethysmography Data

Due to poor quality PPG signal acquired, several algorithms were used in an attempt to get better HR estimates.

The first family of algorithms used tried to do event detection in time domain, in particular, peak and onset detection [42], with different filterings of the raw signal. These algorithms presented a very high error rate due to very low SNR of the cardiac frequency components in the acquired signal.

As movement artifacts were one of the main sources of noise, the next family of algorithms tried included diferent types of adaptive filtering with and without Laguerre expansion [43, 56] in an attempt to minimize the correlation between PPG and the simultaneous accelerometry data. Although slightly better that just filtering, adaptive filters still weren't capable of isolating the relevant components well enough to get a good HR estimate.

The last family of methods tried was based on sparse signal reconstruction [44, 45]. These recent methods are said to be more adequate to heavily corrupted signals and perform better than the previously mentioned, although, the results obtained were still not better than the ones provided by the smartwatch's manufacturer algorithm.

A total of 8 algorithms were used to process the PPG and accelerometry data coming from S3 to produce estimates of HR. However due to highly corrupted signal, none of these methods was able to surpass the performance of the manufacturer algorithms which must be a highly complex and tailor made algorithm (that still does not perform ideally).

## 4.5 Physical Activity METS estimation

METs were estimated using rules presented in [2] and refined in [57].

Counts estimation from accelerometry data was made according with 4.7.

$$Counts_{10s} = \sum_{i=1}^{10*f_s} |ACC_{i-1} - ACC_i| \tag{4.7}$$

$ACC_i$ → Accelerometry sample number i

$f_s$ → Accelerometry sampling frequency

This value allows to estimate what is the effort level of the individual, helping doctors to perceive the health status of the patient, for example crossing this information with the simultaneous HR, allowing them to have a notion of the effort the patient is capable of enduring. Another utility of this measure is to determine the physical activity profile of the patient, and track its changes along time, allowing to determine if a certain medication is taking effect for example.

# 5. System Evaluation

The system was used and tested on patients with cardiac diseases from Hospital de Santa Marta - CHLC (Lisbon) and the main variables collected were heart rate (HR) and Metabolic Equivalent of Tasks (METs) [2]. These variables were indicated by the hospital's cardiology team as being common variables used in their practice to diagnose and track patients [3, 4]. Variables to be collected determined the type of sensors needed.

Devices utilized to test this system were chosen as they provide cost effective platforms to support the desired functions. Cost and quality were taken into account and established market products were preferred as they offer greater consistency in quality over time and over numbers i.e. it is easy to ensure the good working of the system even if more devices are added after some time and they can be added in larger numbers to use with more patients.

After all hardware and software was finished the system was tested in lab and hospital conditions. The main objectives of the tests were to assess system's usability, concerning comfort and ease of use by patient and doctor. Another major objective of testing the system was to verify the accuracy and reliability of the sensors utilized and the robustness of all the software, data transmissions and storage. All the software components were tested individually but still real life situations brought to light some minor bugs that were corrected once detected. Another major concern to be tested was loss of data during transmissions.

The devices used in the present work are directed to patients diagnosed or with suspects of cardiac insufficiency, hence the most relevant variables to collect are related with heart rate (HR) and accelerometry as a way of estimating the level of physical activity during the patients' daily life. To collect this data 3 devices were used, each with different sensors as detailed in table 5.1. Complete system architecture for the testing phase is depicted in fig. 5.1.

Table 5.1: Variables acquired by each device and the corresponding information extracted. CD - Chest Deflection (respiratory frequency); PA - Physical Activity (measured in METs); ACC - Accelerometry

| Device | Variables | Sampling Rate (Hz) | Extracted information |
|---|---|---|---|
| Smartwatch | PPG | 25 | HR |
| | HR | 25 | HR |
| | ACC | 100 | PA |
| BITalino | ECG | 1000 | HR |
| | ACC | 1000 | PA |
| | CD | 1000 | RR |
| Smartphone | ACC | 100 | PA |

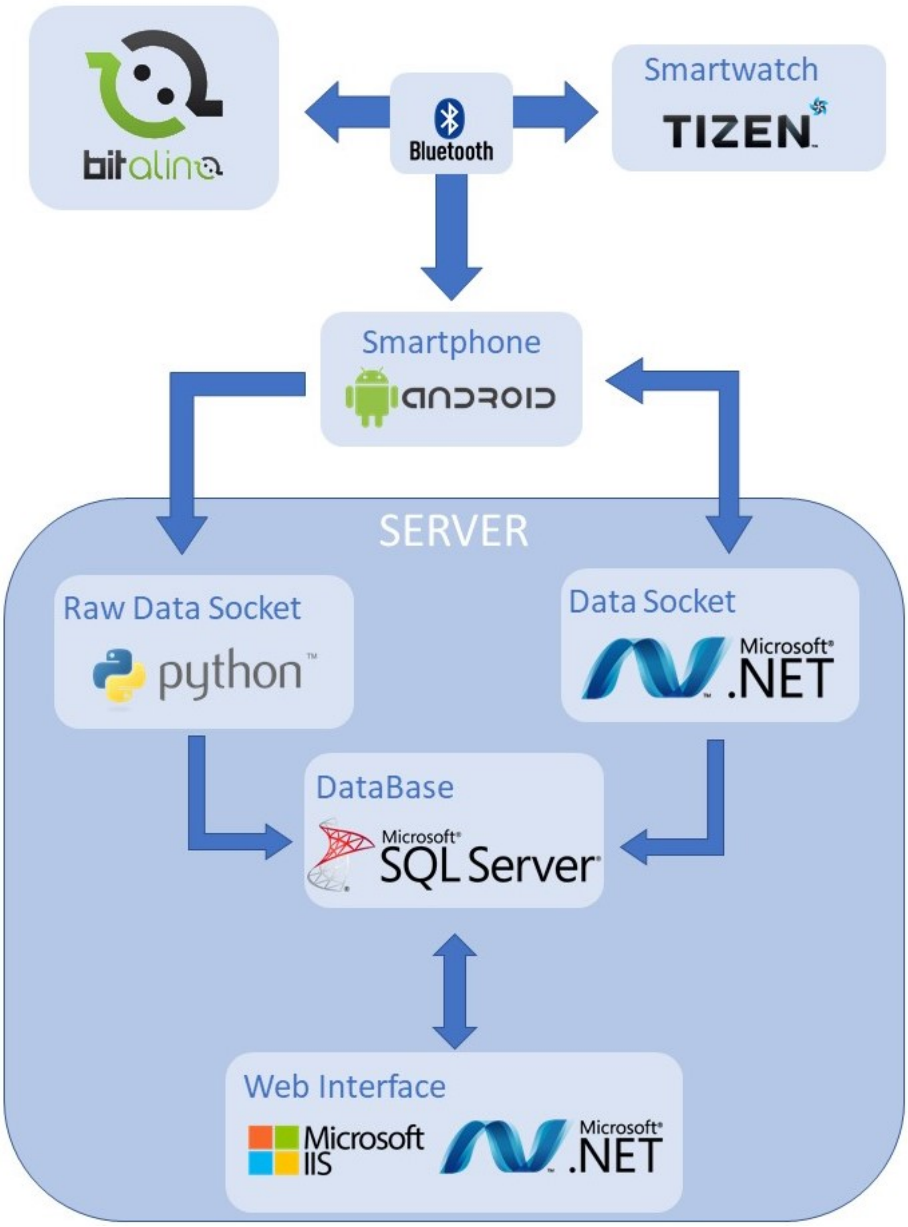



Figure 5.1: Platforms and programing languages used to build and run the system. Choice of platforms were based on device

## 5.1 Wearable Sensors

### 5.1.1 Smartwatch

The smartwatch used was the Samsung Gear S3 and it's function is to collect data and send it to the smartphone, acting like a sensor. This device was chosen as a way to reduce the patient's discomfort, as it can be a less invasive sensor, replacing the watch many use daily. Another reason for the choice of this device was its novelty, as it was a new model at the time devices were chosen for this work, and it was expected to perform better than every previous smartwatch. Finally, the choice of this highly capable and versatile smart device was made with future developments in mind as it allowed for more functionalities to be implemented, like its use as a data hub (replacing the smartphone) and even allowing for an interface so the patient could interact with the system using only the smartwatch. It has several sensors, including luminosity, atmospheric pressure, PPG, ACC and others decribed in [58]. This device also has another ”sensor” that produces values of instantaneous HR calculated from PPG data.

### Tizen App

A native application software was developed to collect data from the desired sensors and send it, through BT, to the smartphone. This smartwach's operative system (OS), Tizen, supports applications written in C language using many functions provided specifically for this OS.

When app is first initialized, a Bluetooth server is mounted awaiting for incoming connection from the smartphone. To prevent app being interrupted when screen is off, it is needed to lock CPU in on state and also, initialize sensors API with the appropriate option.

Code responsible for initializing the app Bluetooth server and sensors with the correct formulaion to prevent unintentional deactivation:

```c
#include <bluetooth.h>
#include <app_control.h>
#include <glib.h>
#include <stdlib.h>

static bool app_create(void *data){
  int server_socket_fd = -1;
  bt_adapter_state_e bt_ad_state = BT_ADAPTER_DISABLED;

  device_power_request_lock( POWER_LOCK_CPU, 0);

  bt_initialize();
  bt_adapter_get_state(&bt_ad_state);

  if (bt_ad_state != BT_ADAPTER_DISABLED) {
  } else {
    bt_socket_create_rfcomm(BT_MGR_UUID, &server_socket_fd);
    bt_socket_set_connection_state_changed_cb(_socket_conn_state_changed_cb);
    bt_socket_listen_and_accept_rfcomm(server_socket_fd, MAX_NUM_PENDING);
  }

  data_initialize(data, update_sensor_values);

  if(feedback_initialize() == FEEDBACK_ERROR_NONE){
    feedback_play_type(FEEDBACK_TYPE_VIBRATION ,FEEDBACK_PATTERN_LOWBATT);
    feedback_play_type(FEEDBACK_TYPE_VIBRATION ,FEEDBACK_PATTERN_BT_CONNECTED);
    feedback_play_type(FEEDBACK_TYPE_VIBRATION ,FEEDBACK_PATTERN_BT_DISCONNECTED);
    feedback_play(FEEDBACK_PATTERN_LOWBATT);
    feedback_play(FEEDBACK_PATTERN_BT_CONNECTED);
    feedback_play(FEEDBACK_PATTERN_BT_DISCONNECTED);
  }

```

```
  return true;

}

bool data_initialize(void *data, Update_Sensor_Values_Cb sensor_update_cb)
{
  int i=0;

  //sensor_type_e sensors_list[] = {SENSOR_ACCELEROMETER, SENSOR_LIGHT, SENSOR_PRESSURE,
      SENSOR_HRM, SENSOR_HRM_LED_GREEN};

  sensor_type_e sensors_list[] = {SENSOR_ACCELEROMETER, SENSOR_HRM, SENSOR_HRM_LED_GREEN
      };

  for (i = 0; i < SENSOR_COUNT; ++i) {
    sensor_get_default_sensor(sensors_list[i], &sensors[i].handle);
    sensor_create_listener(sensors[i].handle, &sensors[i].listener);
    sensor_listener_set_option(sensors[i].listener, SENSOR_OPTION_ALWAYS_ON);
    sensor_listener_set_event_cb(sensors[i].listener, LISTENER_TIMEOUT, sensor_update_cb,
        i);
    sensor_listener_start(sensors[i].listener);
  }
  return true;
}

void update_sensor_values(int count, float *values, sensor_type_e type, appdata_s *ad)
{

// Manipulate sensor's values

}
```

When a Bluetoth connection is established, a particular message 'go' must be sent to initialize data collection. Other messages are available to stop the acquisition 'quit' or to request battery information 'batt'.

Code used to get battery status and percentage:

```

device_battery_level_e battery_status = DEVICE_BATTERY_LEVEL_EMPTY;

int battery_percent = 0;

device_battery_get_percent(&battery_percent)

device_battery_get_level_status (&battery_status)
```

To interact with the sensors, a manufacturer provided API was used. It is built in a logic of callbacks, i.e. an event is generated when the sensor value changes. Sampling rate is defined to be the default for each sensor that, according with the manufacturer guide, can range between 100Hz and 1 HZ [58] meaning that events occur at different rates for different sensors.

To have a uniform sampling protocol, a buffer is kept containing the current value of each sensor and whenever a sensor event occurs, the corresponding values in this buffer are updated. In parallel, in another thread, there is a timer running at 100Hz that takes a snapshot of values in the buffer and writes them into a different array accompanied by a timestamp of the moment the snapshot was taken. Data from all sensors is held and when the array has 10 seconds of data, it is sent to the smartphone. This transmission strategy was implemented as a way of reducing the overhead of sending samples in real time, at 100Hz, which had a larger battery consumption and also induced the loss of samples, reported by the BT API provided by the manufacturer, returning an error for every message lost. The buffer technique was used as a very easy and simple way of upsampling data from sensors with a sampling frequency lower than 100Hz using a sample and hold method.

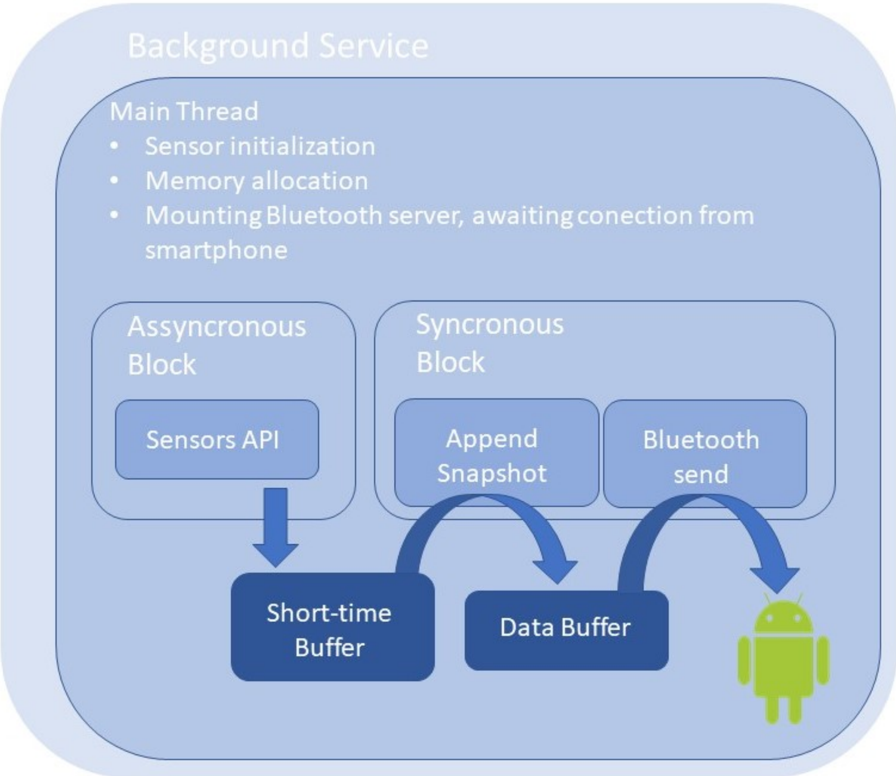



Figure 5.2: Smartwatch App threads call-stack

Figure 5.2 shows the implemented app architecture having two main components, an asynchronous and synchronous block. The first runs when a sensor reports a new sample and its callback ir triggered. This function updates the buffer value that corresponds to the sensrs that triggered its callback. The other block that runs periodically and appends the current buffer values to another, larger, array, that stores 10s of data. This block is also responsible for sending the data when the 10s of data is collected, clearing the storage array.

Figure 5.3: Smartwtch Samsung Gear S3 Frontier

### 5.1.2 BITalino

BITalino[9] was used in the form of a chest band that collects various signals and sends them to the smartphone via BT. In particular it was chosen because it provides a high quality ECG signal [8, 9] that complemets the information collected by the smartwatch. BITalino was used, also, because it was a familiar tool that fitted the requirements to provide a ground truth to the other sensors used in the system. Initially the use of this device was meant to be as a validation tool and would not be used once the system was fully operational and the number of monitored patients increased. But the relevance of the information provided by the ECG signal it collected justified its presence in the context of cardiac patients monitoring.

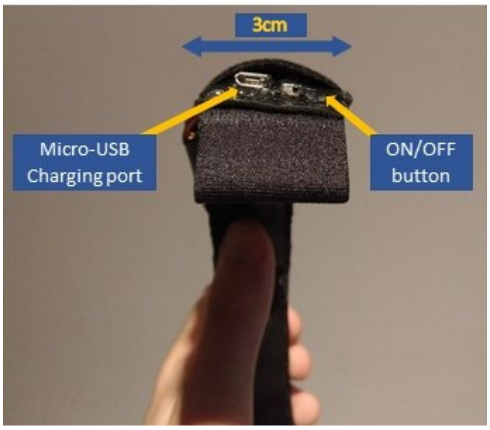
A chest band was built in order to accommodate the sensors enumerated in table 5.1 with a comfortable and practical form-factor. This was a major concern as patients comfort was very important to ensure success of the system, once it was intended to be used for long periods of time. Because of this, also battery size was an issue. Stock BITalino battery has 700mAh of capacity, but to eliminate any chance of this not being enough, a 1300mAh was used (table 5.2).

Apart from the electronic components, the chest band was built using a standard commercial chest band from Polar®used for sports tracker devices. This was a very good base as this band had several positive aspects:

- It had sown into the fabric two conductive rubber pads from which ECG could very easily be collected.
- Elastic and adjustable bands with easy to use locks, meaning it was very easy to put on and take off while ensuring a good fit to most patients.

To ensure protection of the electronics components, a tough fabric cover was sewn into the elastic band with holes for the on/off switch and charging port.

(a) Polar®band that served as base for the BITalino components.




(b) Charging port and on/off switch.

Figure 5.4: Chest band.

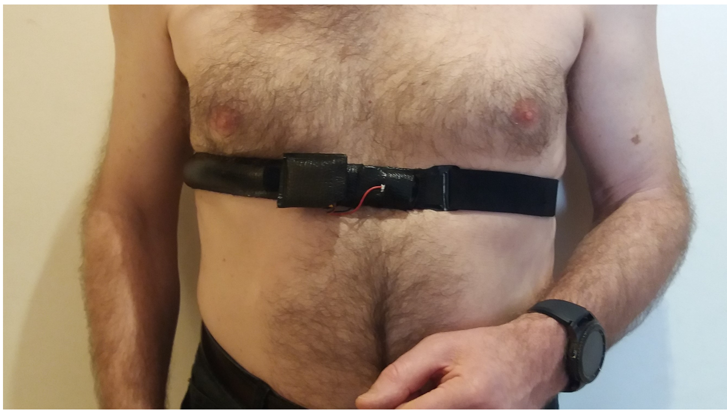


Figure 5.5: Patient wearing the selected sensors.

## 5.2 Comfort and Usability

As mentioned before, comfort and ease of use for patient and physician were two major concerns when designing and choosing all hardware and software features.

For the patient the importance of these aspects comes from the fact that system is meant to be used continuously for long periods of time and if the system becomes a hassle it is harder to get patient's compliance and get useful data. It was also very important that all interfaces and charging procedures were as simple as possible. This was important because great part of the patients to be monitored are elderly and in some cases even illiterate so simplicity was a major concern. For this, on screen battery information and sound notifications were utilized to notify the patient to charge the devices when needed (fig. 3.4).

To ensure practicality and ease of use for physicians and medical staff, requisites were to have a simple web interface (appendix A) in what concerns patient enrollment into the platform, and data visualization.

All of these aspects were assessed along the design process and were put to test in hospital context.

During the design process students were asked to wear the system and report on aspect to be improved relating with comfort and ease of use.

As an initial hospital test, a medical team from cardiology service was given all the equipment and, using the guides provided in appendix A, fitted the system into a 72 year old male patient. Following this the patient wore the system for 24 hours and reported the experience.

From this experience the comments that emerged were mostly related with charging procedures. Web, Android and Tizen interfaces were considered very convenient and simple to use. Battery life was one of the main aspects referred as to be improved by the medical professionals. This led to the introduction of a power bank for the smartphone and a bigger battery for the chest band to ensure battery life expectations were met. Despite efforts, concerning software optimization and increased battery sizes, table 5.2 still depicts slightly shorter-than-optimal battery life-times.

Another negative aspect reported was the difficulty with charging the chest band. As a safety measure, BITalino does not charge the battery while acquiring data, this was designed to guarantee isolation of the patient from the electricity mains. Although this feature creates an additional step of turning the chest band on and off when charging, the main problem reported by medical teams had to do with the toggle button. This is a very small component, as shown in section 5.1.2, which was considered difficult to use due to its size. Although it was a very valid comment, to change it, a major redesign of the chest band was required since all alternative buttons had a significantly larger size and for this reason, this aspect was left unchanged for the course of this work.

Overall the system had very positive feedback by all participants, patients and medical teams. On the medical personnel side, the system worked well as a data acquisition platform with minimum configuration required, and interfaces were considered very convenient, presenting very useful information. On the patients side there was an unexpected effect, besides considering the system comfortable to wear with minimum disturbance of their daily-life, patients reported to feel more accompanied and better taken care knowing that they are being monitored 24hr a day. This was not expected initially and may contribute to even better outcomes, as feeling more taken care of may improve patients' health status acting similarly to the placebo effect.

Table 5.2: Battery lifetime of acquisition devices

| Device | Conditions | Duration |
|---|---|---|
| Smartphone | Connected only to Smartwatch | 13h 30min |
| | Connected to BITalino and Smartwatch | 11h 30min |
| | Connected to BITalino, Smartwatch and power bank | **16h 30min** |
| BITalino | With 1300mAh battery | 48h |
| Smartwatch | —– | 12h 30min |

## 5.3 Communications

In order to determine if data transmission was occurring without losses nor delays, each message had a mark to allow its localization in time. For communications between the smartphone and BITalino, a sequence number is included and allows to determine if there are missing messages by checking the increment of this number. For messages between the smartphone and Gear S3 a timestamp was sent with each message and data loses can be detected by checking the time delay between consecutive messages.

To assess the situations that led to data loss, this sequence numbers and timestamps were checked while the system was acquiring data. Distance is one of the factors that may contribute for packets being lost in transmission, although, during the normal use of the system, distances are rather short so it is not expected that this interferes with message transmission. In fact the only situation where data loss systematically occurred was when the smartphone was requesting a Bluetooth connection i.e. if the smartphone is already receiving data from device A and is trying to connect with device B, some packets coming from A are lost. Otherwise the system performed in a very robust way concerning data transmissions.

## 5.4 Heart Rate Estimation from PPG data

In order to evaluate the accuracy of HR estimates given by Samsung Gear S3, the later was used to record HR in different situations with several subjects with and without cardiac pathologies. Simultaneously a BITalino [9] and a medical-grade certified device were used to determine the HR from ECG data and provide a ground truth reference for Gear's values.

### 5.4.1 Methodology

#### Experimental Protocol

To validate heart rate values calculated by Gear S3 three experiments were conducted:

- E1 - Consisted on acquiring data for short periods of time with subjects performing a specific activity:
  - Resting state, moving as least as possible (2min)
  - Walking at a regular pace (5min)
  - One subject was asked to pedal on a stationary bike at moderate pace (5min)
  - One subject was asked to run at moderate pace (2min)

  Some of these acquisitions were repeated to verify reproducibility and infer on system robustness.
- E2 - On this experiment, data was recorded for periods between 1 and 2 hours while subjects performed their usual activities during their daily lifes.
- E3 - The experiment consisted on collecting data from 3 hospitalized patients with various cardiac diseases during a stress test, taking place at hospital. These tests consisted of patients walking on a treadmill, increasingly fast until they are not able to continue, followed by a rest period. Patients ECG and energy consumption are monitored while executing the task. The procedure was performed according to Bruce protocol [59].

### Test Equipment

Multiple devices were used to evaluate subjects' HR:

- Samsung Gear S3 Frontier - HR is calculated by the smartwatch from PPG data using manufacturer's algorithm, as detailed in section 4.4.
- A BITalino based chest band, collecting one derivation ECG at 1000Hz, with HR estimation being made according with section 4.1
- For stress tests HR information is also collected by a Mortara XScribe 3.10.10 by Mortara Instrument [60] that collects 12 lead ECG, producing a very reliable HR estimate as it is a certified medical device.

### Accuracy Measurement

The accuracy of the HR values produced by the smartwatch, and also the ones estimated from the ECG signal, was summarized as statistics of the absolute error (AE) e.g. mean absolute error (MAE) as defined in eq. (5.2).

$$AE_i = \left|HR_i^{true} - HR_i^{est}\right| \tag{5.1}$$

$$MAE = \frac{1}{N}\sum_{i=0}^{N}\left|HR_i^{true} - HR_i^{est}\right| \tag{5.2}$$

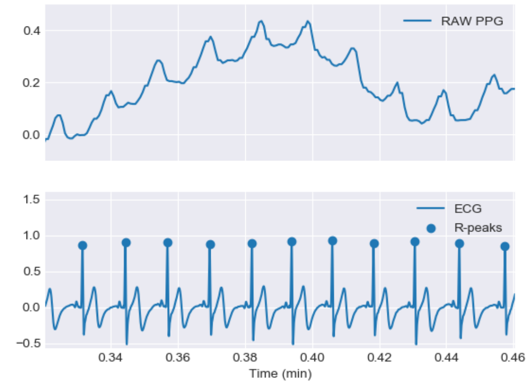
$N$ $\rightarrow$ nr. of HR values estimated for a subject
$HR_i^{true}$ $\rightarrow$ Ground-truth value of HR for ith 10s signal window
$HR_i^{est}$ $\rightarrow$ Estimated value of HR for ith 10s signal window

This measurement was chosen as it provides an easy and fast interpreting quantity to be known and used by physicians when analyzing patient data. Furthermore, unlike usual metrics like percent error, absolute error is completely independent from HR range which is a desirable feature as, ideally, the precision should be high for all HR ranges, and there is no reason for errors at higher HR values being less important.

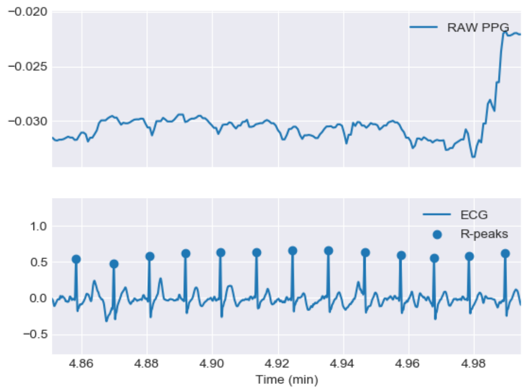
## 5.4.2 Experimental Setup and Results

Data was collected from two separate groups performing different activities. Stress tests were undertaken by 3 patients with various cardiac conditions and ages 42, 47 and 50 years old. The other experiments enumerated in section 5.4.1 were performed by 3 healthy volunteers all 23 year olds. In following figures and tables Si denotes subject i and Tj denotes jth trial.

After data collection from subjects, the first observation of the PPG signal revealed the vast influence of motion artifacts (MA). This is depicted in fig. 5.6. When the subject is completely immobile it is easy to find, even by inspection, a correlation between ECG and PPG signal peaks (fig. 5.6a) corresponding to systolic peaks. The same is not true when the subject is moving, as signal corruption greatly affects PPG signal and correlation is no longer obvious and maybe not present at all (fig. 5.6b).




(a) Raw signal while completely immobile. HR calculated from PPG (first curve): 75bpm; HR calculated from ECG (second curve): 79bpm.




(b) Raw signal while walking. HR calculated from PPG (first curve): 52bpm; HR calculated from ECG (second curve): 81bpm

Figure 5.6: Segments of raw signal captured by the sensors in different conditions with the synchronous identified R-peaks.

Observing fig. 5.9, it is visible that Gear S3 is not capable of keeping up with fast variations in HR and fig. 5.11 clearly illustrates the difficulty in determining HR accurately during movement, with a very large estimation error being present, until subject enters the resting phase of the stress test.

### Short Acquisitions (E1)

When analyzing the absolute error of HR determined from the smartwatch's PPG data collected while subjects were performing specific activities, it is very obvious that motion highly corrupts sensor data and thus, greatly damages accuracy. In fig. 5.7 is very clear a tendency to error increase as subjects go from resting to walking, cycling or running. Another thing that can be noted in fig. 5.7 and table 5.3 is the relatively large difference in the error values between subjects and trials. This may indicate low robustness of the device's measurements, as they are affected by sensor positioning, tightness and also

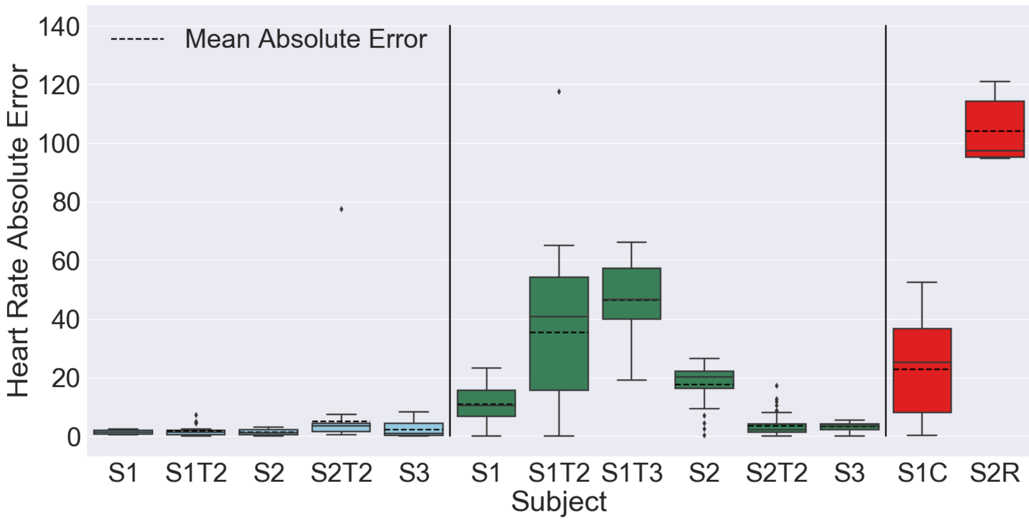



Figure 5.7: Absolute Error of HR while performing different activities: (left) resting; (middle) walking; right cycling and running.

by how the subjects move, as some individuals present accentuated arm movement which can further disrupt accuracy.

Table 5.3: Mean Absolute Error (MAE) of each experiment and activity averages.

| Activity | Subject | MAE | Average |
|---|---|---|---|
| **Resting** | S1 | 1.4 | 2.4 |
| | S1T2 | 1.8 | |
| | S2 | 1.3 | |
| | S2T2 | 5.1 | |
| | S3 | 2.2 | |
| **Walking** | S1 | 10.8 | 21.2 |
| | S1T2 | 35.3 | |
| | S1T3 | 46.4 | |
| | S2 | 17.7 | |
| | S2T2 | 3.4 | |
| | S3 | 3.3 | |
| **Cycling** | S1 | 22.8 | 63.4 |
| **Running** | S2 | 104.1 | |

#### Daily life (E2)

Table 5.4 and fig. 5.8 demonstrate clearly that during daily-life activities like walking or any activity that implies arm movement, Gear S3 has a poor performance as a sensor platform and, in certain occasions, it is not even possible to detect any type of signal trends. In a medical context, this could lead to gross errors when trying to diagnose or follow a patient, or it would require further tests which renders the pervasive monitoring less efficient or even counterproductive.

#### Stress Test (E3)

As expected, during a stress test, where the subject is moving with some intensity, HR determination error is very large and as can be seen in fig. 5.11, error is specially large during the exercise phase

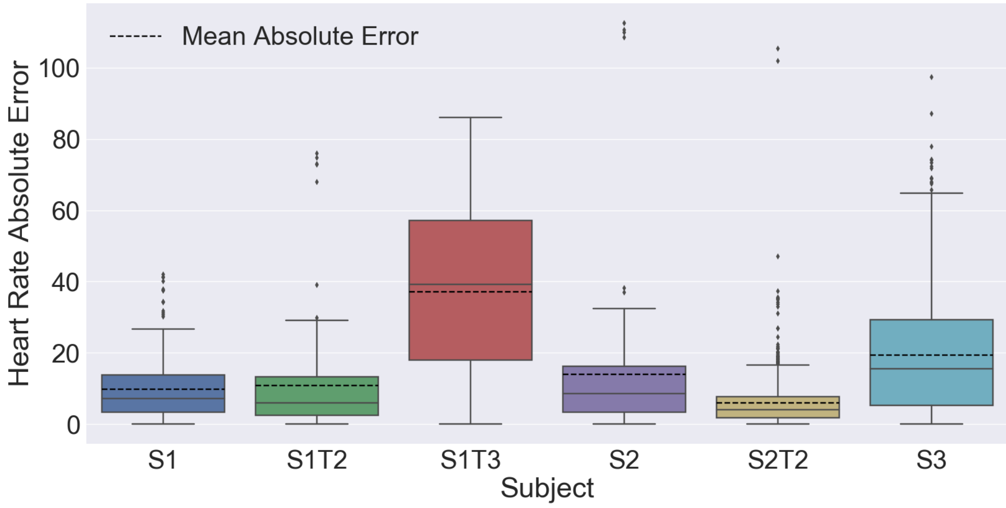



Figure 5.8: Absolute Error of HR while subjects perform their usual life activities (office working, walking, eating, etc...).

Table 5.4: Mean Absolute Error (MAE) of each subject during daily life activities.

| Activity | Subject | MAE | Average |
|---|---|---|---|
| **Daily-life activities** | S1 | 9.7 | 16.2 |
| | S1T2 | 10.8 | |
| | S1T3 | 37.1 | |
| | S2 | 13.9 | |
| | S2T2 | 6 | |
| | S3 | 19.3 | |

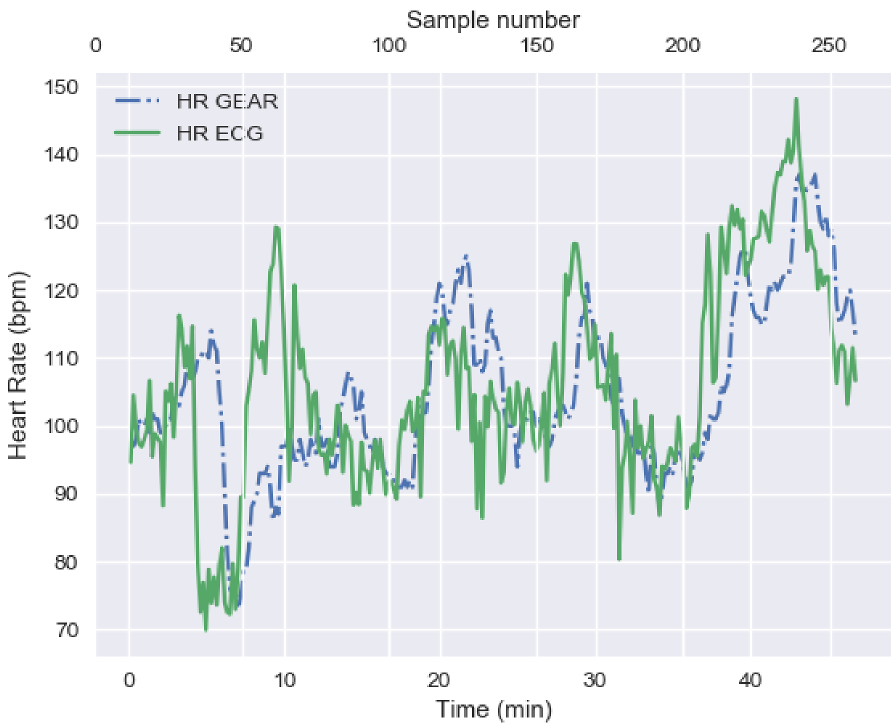



Figure 5.9: Example of HR curves obtained during daily life activities.

of the stress test, supporting hypothesis of motion artifacts corrupting the signal. Error during these tests reached >100bpm wich renders this sensor platform completely unfit for this kind of contextwhere precision is of utmost importance to avoid wrong diagnostics and treatments.

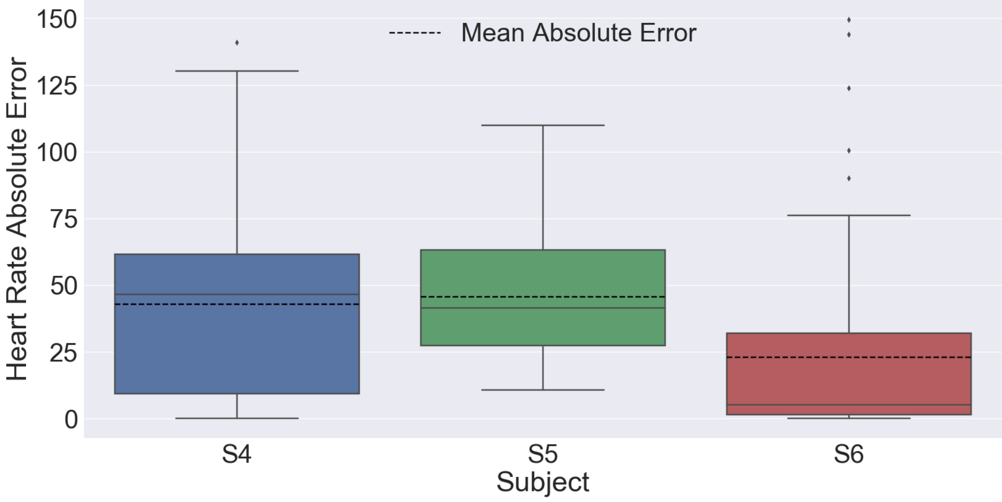



Figure 5.10: Absolute Error of HR during stress test.

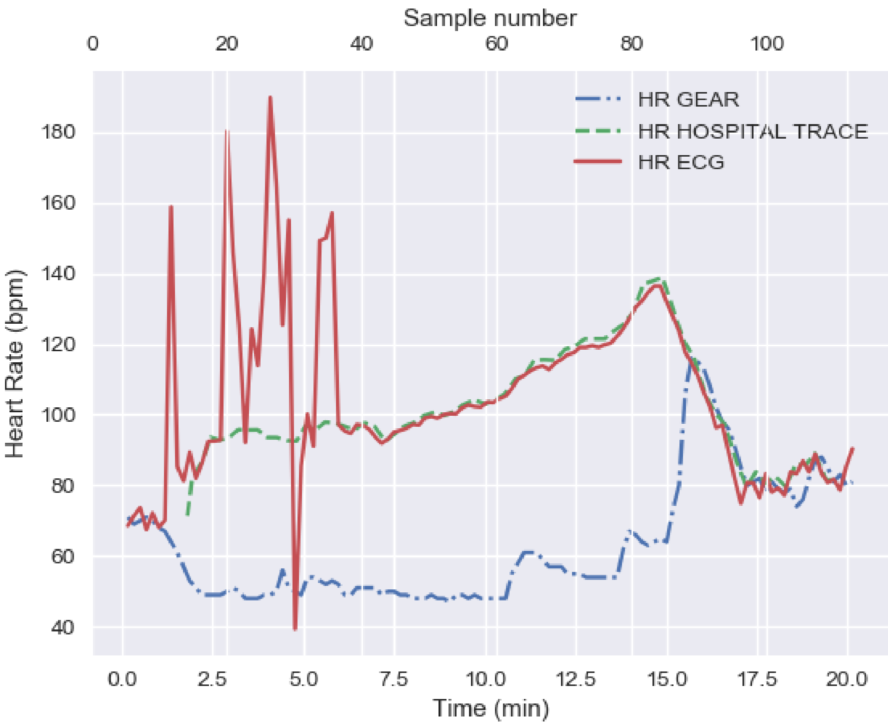



Figure 5.11: Example of HR curve obtained during stress test.

### 5.4.3 Discussion

When observing the results obtained, the first conclusion that can be made is that the PPG signal collected with Samsung Gear S3 is very easily corrupted by motion artifacts. This corruption is present either during exercise or daily-life activities and even with very light movements, some corruption can be noticed. This means that very easily, the identification of systolic peaks in the signal cannot be made reliably, having a negative impact in the HR determination accuracy thus rendering this device unfit for pervasive monitoring in a medical context.

The main reasons for this low accuracy can be related with the sensor itself and also with the device as a whole. The sensor has only one green LED and one photodiode, while other sensors have two of each, with LEDs of different wavelengths. This apparent redundancy has a great impact as it allows for a different signal processing approach that can greatly improve the performance, while having only one wavelength means less information being collected, reducing the signal processing options.

Concerning the weight, it must be taken into account that this is a very complex device that can perform endless tasks with native support for 3rd party apps. All these functionalities and capabilities make the device heavier and lager, thus, making it harder to keep comfortably and tightly secure to the wrist without it having its own dynamics. This is a problem as the relative movement between the sensor and the subject is the sole cause for movement artifacts in the signal. In addition the positioning of the sensor in an area with a relatively low SNR greatly increases the impact of motion artifacts. Also this device has the sensor at the same level of the watch back plate, whereas other devices place the sensor in an elevation of the back plate, which makes the sensor to be slightly pressed against the skin increasing SNR, and thus, accuracy.

Due to poor quality PPG signal acquired, several algorithms were used in an attempt to get better HR estimates. Algorithms used included adaptive filtering with and without Laguerre expansion [43, 56], signal separation by sparse signal reconstruction [44, 45] and onset detection [42]. A total of 8 algorithms were used to process the PPG and accelerometry data coming from S3 to produce estimates of HR. However, the results obtained using all this algorithms performed equally bad, or even worse, than Gear's algorithm and, for this reason, they were not considered. This clearly indicates that elevated error in Gear's HR estimations is probably related with a low quality signal and not with a poorly performing algorithm.

## 5.5 Heart Rate Estimation from ECG data

As a way of validating the algorithm described in section 4.2 running on the smartphone, that is being used to determine the HR from the ECG signal, a comparative analysis was made between the HR estimates based on the proposed algorithm and a standard one described in section 4.1.

The main difference between algorithms is that one is a very simplified RP detector running in the smartphone in real-time while the other runs off-line with no complexity restrictions.

### 5.5.1 Methodology and Experimental Setup

During E2, described in section 5.4.1, ECG was collected by the chest band and was used to test the algorithm proposed in section 4.2. At the same time, the raw signal was recorded and processed off-line by the standard algorithm. This allowed to isolate signal quality from the tests, as the same signal, divided in the same 10s windows was being processed by both algorithms and HR estimates were compared.

### 5.5.2 Results

In fig. 5.12 the HR curves calculated by both algorithm can be seen. The calculations, for this particular subject, present an almost zero error, as the curves appear to be overlapping for almost all samples, and with minor deviations in a few points.

Looking at fig. 5.13 it can be noticed that observations made from fig. 5.12 are in agreement with error values for most subjects.

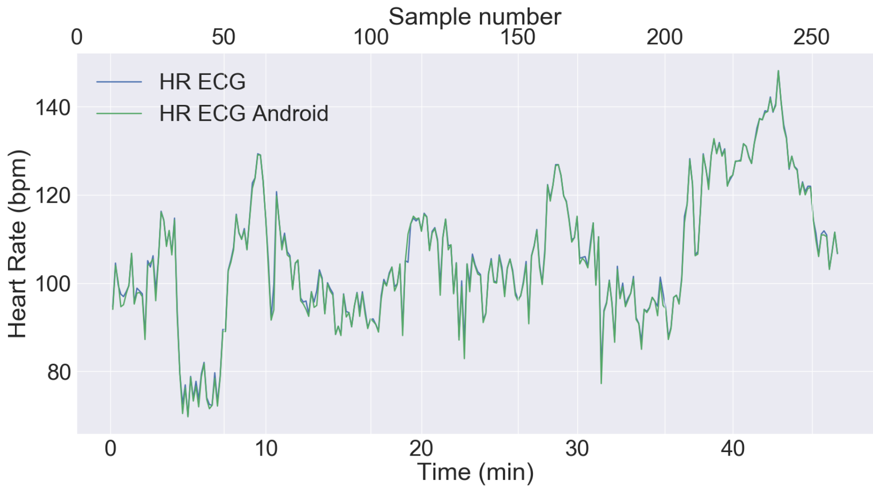



Figure 5.12: Error of android segmenter.

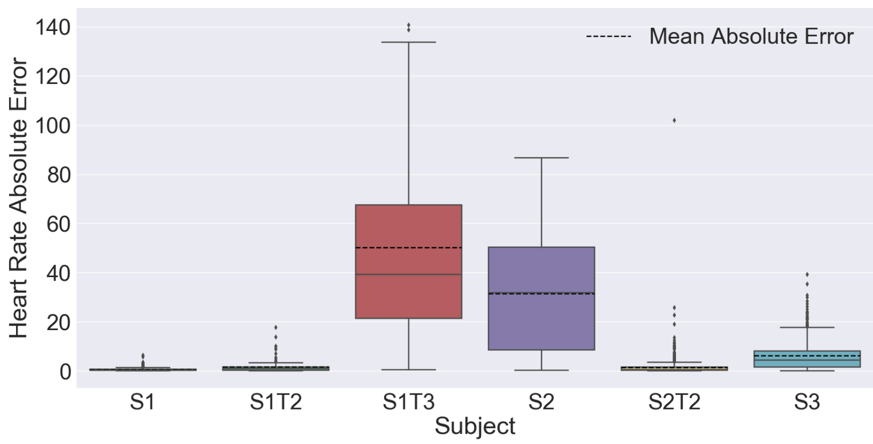



Figure 5.13: Error of android segmenter.

### 5.5.3 Discussion

Analyzing fig. 5.12, it can be noticed that HR estimation made using the proposed algorithm and the reference algoritm [30] are rather small. This demonstrates that, even being computationally lightweight, it perform well during daily-life tasks. Although BITalino produces a good quality ECG signal, when the wearer is moving, some artifacts are introduced, and even in this situation, HR estimates are reliable. This is confirmed by fig. 5.13, where very small errors predominate with exception for two subjects S1T3 and S2, where Si denotes subject i and Tj denotes jth trial. In this particular pair of acquisitions, the increase of error is probably due to the introduction of motion artifacts in the ECG signal, as subjects reported to be exercising.

Besides motion artifacts, the other major weakness of this algorithm is chest band placement. This occurs due to the nature of the algorithm, as it is based in amplitude, it is expected of the R-peak to have an amplitude much larger than other signal features, which is only true for certain ECG electrodes placement. In case of a patient misplacing the chest band, it may be that signal morphology is altered leading to great errors in the HR estimation. However, with careful teaching of each patient on how to operate the system, this problem can be reduced (although not completely eliminated). To solve this problem a completely different algorithm would be needed, being, probably, much more elaborate and resource consuming.

Overall, the algorithm performed very well, speacially when taking its simplicity into account. Robustness may be questionable when exercising, but for daily-life activities, it is very reliable, and improving the reliability would, probably, require a great increase in the computational burden of the algorithm, harming the smartphone's battery life and the general system's usability.

# 6. Conclusions

The present work was divided into four main stages, 1) the implementation of the data collection system and all its required components, 2) designing, choosing and integrating wearable sensors to allow hospital testing, 3) assessing the system concerning robustness of data collection, device battery life and software performance and 4) a user directed test, with sensor data quality, comfort for patients, ease of use and integration in hospital environment being the main points to be assessed.

The first stage of software implementation was centered into developing all the user interfaces and software components, with versatility and user-friendliness being the key-words.

The second stage was somewhat more challenging, as the choice, design and construction of the wearable sensors to be used in later tests had to cover a large range of requirements. On one side, it should collect relevant medical information for the test conditions, which were to have the system monitor cardiac patients. On the other hand, aspects like battery life, comfort and robustness to wear and tear should also be taken into consideration. The final aspect that proved decisive was the future developments of the system, which motivated the integration of a smart-device into the sensors array. This was intended to allow for other functionalities to be implemented in this smart device.

Following the implementation and development stage, the first tests started. The main focus of this stage was to ensure there were no data loses, bugs, and ensuring the system was easy to use and provided relevant information. Conclusions from all these tests led to some iterations, revealing some software aspects that had to be improved. The final result was a lossless data acquisition system with a very convenient interface that could interact with a great variety of sensors with minimum implementation overhead. These tests also revealed some less positive aspects concerning battery life and data quality. Regarding battery life, it was concluded that it was rather short for what was expected, this led to some modifications in the devices used, with a bigger battery and a power bank being introduced for chest band sensor and smartphone, respectively. Also some software optimizations were implemented in order to reduce the computational burden. Regarding sensors' reliability, issues were detected when looking at the PPG data collected by the smartwatch. The sensor was very prone to noise and motion artifacts, rendering the data unfit for a rigorous medical context. This is one of the main points to address, that can lead to the replacement of the chosen device, or the elaboration of better signal processing algorithms.

Finally, the system was tested in a real life scenario, being used to monitor several patients. The main aspects in analysis were the ease of use by patients and medical teams. Battery life and charging schedule, together with the previously detected lack of reliability from one of the sensors, were the less positive aspects pointed out by medical teams. Although, both this issues could be easily solved by replacing the low performing sensors and designing a device charging schedule, making these aspects minor drawbacks for the system as a whole.

Overall, very positive feedback arose from this last stage. Interfaces and devices were considered easy to use and comfortable by patients and medical teams, providing very useful information and contributing to an increased ability to better diagnose and treat patients. Versatility was one of the most prized aspects, as it allows the system to be used with a vast horizon of medical conditions, and contexts, making timeless, in a sense, as the same system can be used as sensor technology evolves. All medical personnel involved in this testing phase considered this to be a valuable addition to their practice.

An unexpected outcome of this work had to do with patients reaction to the system. Besides considering the system comfortable to wear with minimum disturbance of their daily-life, patients reported to feel more accompanied and better taken care knowing that they are being monitored 24hr a day. This was not expected initially and may contribute to even better outcomes, as feeling more taken care of may improve patients' health status acting similarly to the placebo effect.

## 6.1 Achievements

The main objectives of the present work were the development, testing and implementation of a pervasive monitoring system for non-hospitalized patients. The system's features should include support for various sensors, ensuring versatility, and an easy to use web interface that allows for remote real time data visualization to be integrated as a diagnose and follow up tool for medical teams. Additionally, it was intended that the final system would be tested and integrated into a final user scenario in HSM.

After all the work and tests, the outcome of this work is as previously intended: a fully functional pervasive monitoring system that can be used with a great variety of sensors, making it suitable for almost any long term monitoring requirement, with real time data visualization for medical teams witha convenient interface. Alongside with visualization, there is also the possibility to configure alarms when specific conditions occur. The system had very positive feedback from medical teams, that considered it as a very useful diagnosis and follow-up tool, and also patients who considered the system to be comfortable to use and felt they were being given better care as the doctors could observe them continuously, and not sporadically, as happens in traditional hospital health care.

## 6.2 Future Work

As discussed before, the single most important aspect still to address is the battery life of the android smartphone. Despite being the major point that could lead to the misfit of this device to a hospital environment, this may not be easy to solve, as BT connection to the remote sensors, and the signal processing algorithms require a lot of computations making it hard to optimize the software into extending battery life.

Adding to this, there is the specific warable sensors chosen to the cardiology department testing phase. One of the selected wearale sensors, the smartwach, revealed not to be reliable enough to be integrated into this type of monitoring system. This would require this specific sensors to be improved, replaced, or alternatively, that a super-efficient algorthm was used to process the acquired data.

Being the system completely versatile in what sensors it uses, possible improvements on the system are always possible with the implementation of support for newer and better sensors, which could even include environmental sensors. The system was designed to deal with wearable sensor platform constantly collecting data on the patient's physiological parameters. Although other types of sensors could be considered, and integrated into the system, like cameras to allow for motion detection, activity and environmental monitoring, or even devices periodically to monitor patients weight or arterial pressure. Although these types of sensors do not comply with pervasive monitoring, they could still be integrated and provide useful information to medical teams.

A possible addition for this system could be an interface for family and caregivers, allowing them to also be informed about the health state of the patient and even receive notifications when a relevant event takes place e.g. when a patient falls. Which leads to another possible improvement, which would be the detection of events that could be detected through the accelerometer. These events could be simply fall detection or could evolve to full activity recognition. This still presents many challenges elated with computational burden and battery exhaustion, and also with sensors placement, as activity recognition is still a search topic and thus a good and efficient way of accomplishing it may be a challenge to implement in practice.

A final thought on a possible improvement, would be to incorporate sporadically inquiries to the patient as a mean to retrieve the self reported health state or other relevant information.

# Bibliography


[1] I. Faria, C. Gaspar, M. Zamith, I. Matias, R. César das Neves, F. Rodrigues, and C. Bárbara. Telemold project: Oximetry and exercise telemonitoring to improve long-term oxygen therapy. *Telemedicine and e-Health*, 20(7):626–632, 2014.

[2] S. E. Crouter, K. G. Clowers, and D. R. Bassett Jr. A novel method for using accelerometer data to predict energy expenditure. *Journal of applied physiology*, 100(4):1324–1331, 2006.

[3] A. Elhendy, D. W. Mahoney, B. K. Khandheria, K. Burger, and P. A. Pellikka. Prognostic significance of impairment of heart rate response to exercise: impact of left ventricular function and myocardial ischemia. *Journal of the American College of Cardiology*, 42(5):823–830, 2003.

[4] R. V. Milani, C. J. Lavie, and M. M. Cassidy. Effects of cardiac rehabilitation and exercise training programs on depression in patients after major coronary events. *American heart journal*, 132(4): 726–732, 1996.

[5] K. H. Shelley. Photoplethysmography: beyond the calculation of arterial oxygen saturation and heart rate. *Anesthesia & Analgesia*, 105(6):S31–S36, 2007.

[6] J. Allen. Photoplethysmography and its application in clinical physiological measurement. *Physiological measurement*, 28(3):R1, 2007.

[7] H. B. Abramowitz, L. A. Queral, W. R. Flinn, P. F. Nora, L. K. Peterson, J. J. Bergan, and J. S. Yao. The use of photoplethysmography in the assessment of venous insufficiency: a comparison to venous pressure measurements. *Surgery*, 86(3):434–441, 1979.

[8] D. Batista, H. Silva, and A. Fred. Experimental characterization and analysis of the bitalino platforms against a reference device. In *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*, pages 2418–2421. IEEE, 2017.

[9] J. Guerreiro, R. Martins, H. Silva, A. Lourenço, and A. L. Fred. Bitalino-a multimodal platform for physiological computing. In *ICINCO (1)*, pages 500–506, 2013.

[10] C. S. Bloss, N. E. Wineinger, M. Peters, D. L. Boeldt, L. Ariniello, J. Y. Kim, J. Sheard, R. Komatireddy, P. Barrett, and E. J. Topol. A prospective randomized trial examining health care utilization in individuals using multiple smartphone-enabled biosensors. *PeerJ*, 4:e1554, 2016.

[11] R. Rawassizadeh, B. A. Price, and M. Petre. Wearables: Has the age of smartwatches finally arrived? *Communications of the ACM*, 58(1):45–47, 2015.

[12] G. Bieber, M. Haescher, and M. Vahl. Sensor requirements for activity recognition on smart watches. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, page 67. ACM, 2013.

[13] F. El-Amrawy and M. I. Nounou. Are currently available wearable devices for activity tracking and heart rate monitoring accurate, precise, and medically beneficial? *Healthcare informatics research*, 21(4):315–320, 2015.

[14] A. Carpenter and A. Frontera. Smart-watches: a potential challenger to the implantable loop recorder? *Europace*, 18(6):791–793, 2016.

[15] U. Varshney. Pervasive healthcare and wireless health monitoring. *Mobile Networks and Applications*, 12(2-3):113–127, 2007.

[16] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of neuroengineering and rehabilitation*, 9(1):21, 2012.

[17] L. Atallah, B. Lo, G.-Z. Yang, and F. Siegemund. Wirelessly accessible sensor populations (wasp) for elderly care monitoring. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pages 2–7. IEEE, 2008.

[18] B. Lo and G.-Z. Yang. Key technical challenges and current implementations of body sensor networks. In *Proc. 2nd International Workshop on Body Sensor Networks (BSN 2005)*, 2005.

[19] O. Aziz, B. Lo, R. King, A. Darzi, and G.-Z. Yang. Pervasive body sensor network: an approach to monitoring the post-operative surgical patient. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, pages 4–pp. IEEE, 2006.

[20] B. Lo, L. Atallah, O. Aziz, M. El ElHew, A. Darzi, and G.-Z. Yang. Real-time pervasive monitoring for postoperative care. In *4th international workshop on wearable and implantable body sensor networks (BSN 2007)*, pages 122–127. Springer, 2007.

[21] S. Farrell and H. Tschofenig. Pervasive monitoring is an attack. 2014.

[22] M. Li, W. Lou, and K. Ren. Data security and privacy in wireless body area networks. *IEEE Wireless communications*, 17(1), 2010.

[23] A. B. De Luna, V. N. Batchvarov, and M. Malik. The morphology of the electrocardiogram. *The ESC Textbook of Cardiovascular Medicine Blackwell Publishing*, pages 1–35, 2006.

[24] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, March 1985. ISSN 0018-9294. doi: 10.1109/TBME.1985.325532.

[25] P. Danziger. Big o notation. *Source internet: http://www. scs. ryerson. ca/˜ mth110/Handouts/PD/bigO. pdf, Retrieve: April*, 2010.

[26] A. Lourenço, H. Silva, P. Leite, R. Lourenço, and A. L. Fred. Real time electrocardiogram segmentation for finger based ecg biometrics. In *Biosignals*, pages 49–54, 2012.

[27] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236, 1985.

[28] A. Ruha, S. Sallinen, and S. Nissila. A real-time microprocessor qrs detector system with a 1-ms timing accuracy for the measurement of ambulatory hrv. *IEEE Transactions on Biomedical Engineering*, 44(3):159–167, 1997.

[29] M. Elgendi, B. Eskofier, S. Dokos, and D. Abbott. Revisiting qrs detection methodologies for portable, wearable, battery-operated, and wireless ecg systems. *PloS one*, 9(1):e84018, 2014.

[30] W. A. H. Engelse and C. Zeelenberg. A single scan algorithm for qrs-detection and feature extraction. *Computers in cardiology*, 6(1979):37–42, 1979.

[31] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle. A comparison of the noise sensitivity of nine qrs detection algorithms. *IEEE Transactions on biomedical engineering*, 37(1):85–98, 1990.

[32] Y. M. Chi, T.-P. Jung, and G. Cauwenberghs. Dry-contact and noncontact biopotential electrodes: Methodological review. *IEEE reviews in biomedical engineering*, 3:106–119, 2010.

[33] C. Park, P. H. Chou, Y. Bai, R. Matthews, and A. Hibbs. An ultra-wearable, wireless, low power ecg monitoring system. In *Biomedical Circuits and Systems Conference, 2006. BioCAS 2006. IEEE*, pages 241–244. IEEE, 2006.

[34] M. Braecklein, I. Tchoudovski, C. Moor, K. Egorouchkina, L. Pang, and A. Bolz. Wireless telecardiological monitoring system for the homecare area. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 3793–3795. IEEE, 2006.

[35] J. Allen. Photoplethysmography and its application in clinical physiological measurement. *Physiological measurement*, 28(3):R1, 2007.

[36] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida. Wearable photoplethysmographic sensors-past and present. *Electronics*, 3(2):282–302, 2014.

[37] M. Garbarino, M. Lai, D. Bender, R. W. Picard, and S. Tognetti. Empatica e3 - a wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. In *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*, pages 39–42. IEEE, 2014.

[38] M. P. Wallen, S. R. Gomersall, S. E. Keating, U. Wisløff, and J. S. Coombes. Accuracy of heart rate watches: implications for weight management. *PLoS One*, 11(5):e0154420, 2016.

[39] S. Lee, H. Shin, and C. Hahm. Effective ppg sensor placement for reflected red and green light, and infrared wristband-type photoplethysmography. In *Advanced Communication Technology (ICACT), 2016 18th International Conference on*, pages 556–558. IEEE, 2016.

[40] J. Parak and I. Korhonen. Evaluation of wearable consumer heart rate monitors based on photopletysmography. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 3670–3673. IEEE, 2014.

[41] E. E. Dooley et al. *Measuring the validity of self-monitoring heart rate and activity tracking wearables*. PhD thesis, 2016.

[42] R. Blazek and C. Lee. Multi-resolution linear model comparison for detection of dicrotic notch and peak in blood volume pulse signals. *Anal Biomed Signals Images*, 20:378–386, 2010.

[43] L. B. Wood and H. H. Asada. Active motion artifact reduction for wearable sensors using laguerre expansion and signal separation. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 3571–3574. IEEE, 2006.

[44] Z. Zhang. Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction. *IEEE transactions on biomedical engineering*, 62(8):1902–1910, 2015.

[45] Z. Zhang, Z. Pi, and B. Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, 2015.

[46] R. P. Troiano, D. Berrigan, K. W. Dodd, L. C. Masse, T. Tilert, M. McDowell, et al. Physical activity in the united states measured by accelerometer. *Medicine and science in sports and exercise*, 40 (1):181, 2008.

[47] C. Chen, R. Jafari, and N. Kehtarnavaz. A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, 76(3):4405–4425, 2017.

[48] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. *IEEE Transactions on Biomedical Engineering*, 58(9):2456–2466, 2011.

[49] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text compression*, volume 348. Prentice Hall Englewood Cliffs, 1990.

[50] G. R. Wright and W. R. Stevens. *TcP/IP Illustrated*, volume 2. Addison-Wesley Professional, 1995.

[51] Google. Android API. URL `https://developer.android.com/reference/android/support/classes.html`.

[52] I. W. Selesnick and C. S. Burrus. Generalized digital butterworth filter design. *IEEE Transactions on signal processing*, 46(6):1688–1694, 1998.

[53] H. Tanaka, K. D. Monahan, and D. R. Seals. Age-predicted maximal heart rate revisited. *Journal of the American College of Cardiology*, 37(1):153–156, 2001.

[54] M. Sharma. Compression using huffman coding. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):133–141, 2010.

[55] H. Mamaghanian, K. Abushab, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. 58:2456 – 2466, 10 2011.

[56] P. T. Gibbs, L. B. Wood, and H. H. Asada. Active motion artifact cancellation for wearable health monitoring sensors using collocated mems accelerometers. In *Proc. SPIE*, volume 5765, pages 811–819, 2005.

[57] S. E. Crouter, E. Kuffel, J. D. Haas, E. A. Frongillo, and D. R. Bassett Jr. A refined 2-regression model for the actigraph accelerometer. *Medicine and science in sports and exercise*, 42(5):1029, 2010.

[58] T. Project. Tizen Sensor API, 2012. URL `https://developer.tizen.org/development/api-references/native-application?redirect=/dev-guide/2.3.1/org.tizen.native.mobile.apireference/group__CAPI__SYSTEM__SENSOR__MODULE.html`.

[59] M. W. Luong, M. Ignaszewski, and C. Taylor. Stress testing: A contribution from dr robert a. bruce, father of exercise cardiology. *British Columbia Medical Journal*, 58(2):70–76, 2016.

[60] M. Instruments. Mortara XScribe. URL `https://www.mortara.com/products/healthcare/hospitals/cardiac-stress-testing/xscribe/`.

# A. User Guides

In case an appendix if deemed necessary, the document cannot exceed a total of 100 pages...

## A.1 Complete User Guide

# Guia do Utilizador Hospitalar

Sistema de Telemonitorização de Insuficiência Cardíaca

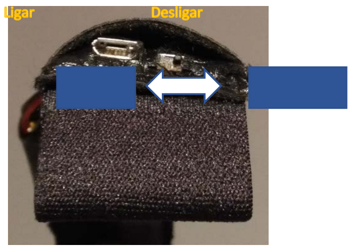
Introdução a dizer variáveis e equipamentos usados

## Configuração Inicial

1. Ligar os dispositivos
    a. Telemóvel – Premir prolongadamente o botão assinalado

    b. Relógio – Premir prolongadamente o botão assinalado

    c. Faixa – colocar o interruptor na posição de Ligar




2. Clicar no icon  do relógio
3. Criar a ficha do doente na plataforma
    a. Iniciar Sessão

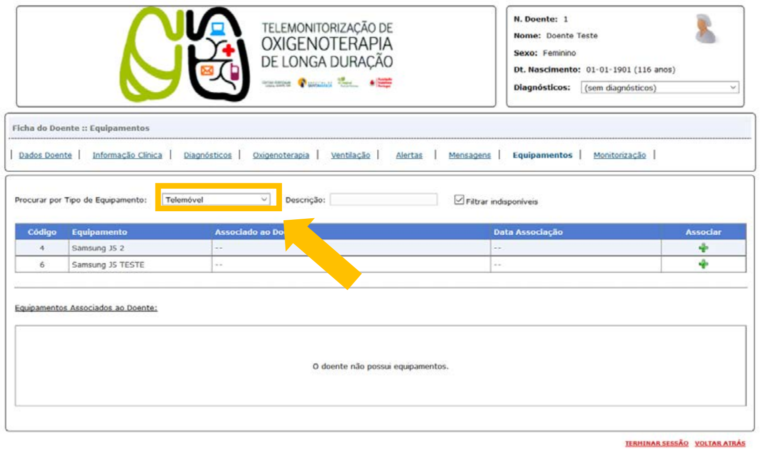

    b. Clicar em "Doentes"


    c. Clicar em "Criar Ficha"

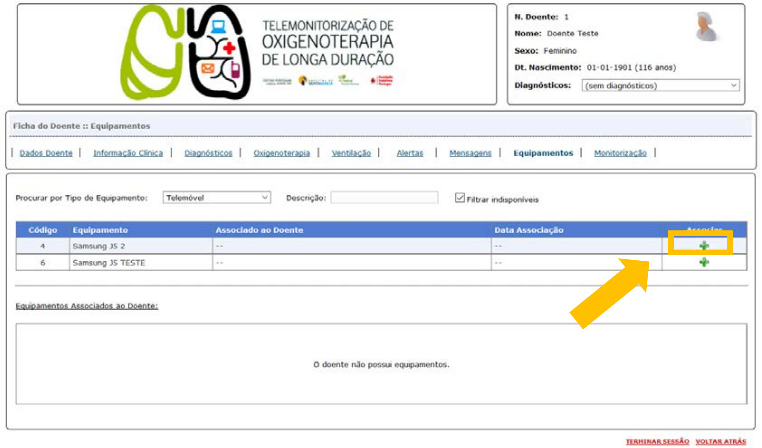

    d. Introduzir os dados do paciente


4. Adicionar os dispositivos no separador "Equipamentos" **(POR ESTA ORDEM)**
    i. GearS3
    ii. BITalino Plux
    iii. Telemóvel

    b. Escolher o tipo de dispositivo a adicionar




    c. Adicionar o dispositivo




5. Colocar os dispositivos no doente
    a. Telemóvel dentro da bolsa que fica a cintura, o cinto que suporta a bolsa deve estar justo de forma a que não oscile
    b. Relógio deve estar justo à pele do paciente (na posição habitual de um relógio na mão **NÃO DOMINANTE**)
    c. Faixa deve ter a etiqueta com o numero centrada no externo do doente, com o numero na posição correta para quem olha para o doente

## Carregamento

- De 12 em 12 horas é necessário carregar o relógio e o power bank que acompanha o telemóvel, não é necessário desligar o relógio durante o carregamento
- Os dispositivos carregam durante aproximadamente 2h
- Após o carregamento é necessário voltar a colocar o relógio, e caso este se tenha desligado deve-se voltar a ligar o relógio sendo necessário voltar a clicar no icon 
- Após o carregamento do power bank, este deve ser ligado ao telemóvel
- Quando o power bank estiver ligado ao telemóvel e o LED do power bank se apagar deve voltar a carregar-se o power bank (isto ocorre cerca de 1h depois de se ligar o power bank ao telemóvel)

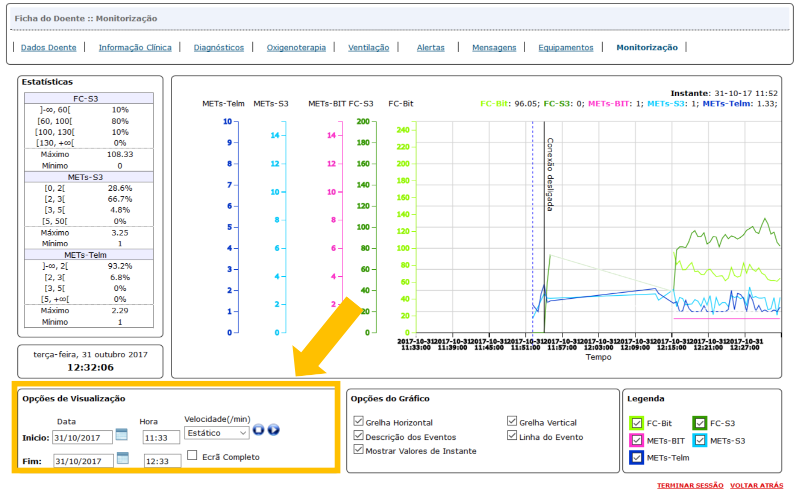
## Visualização de dados

1. Iniciar Sessão


2. Clicar em “Doentes"


3. Clicar em “Monitorização”


4. Definir o intervalo de tempo desejado




5. Depois de definido o intervalo de tempo premir

## A.2 Quick User Guide

# Guia Rápido do Utilizador

Sistema de Telemonitorização de Insuficiência Cardíaca

## Configuração Inicial

1. Ligar os dispositivos, pela ordem indicada:
   a. Telemóvel. Deve garantir que o programa de monitorização fica ativo.
   b. Relógio
   c. Faixa BITalino Plux
2. No relógio, clicar no botão de ligar e surge o menu de aplicações, onde é necessário clicar no ícone da aplicação
3. Aceder à plataforma em http://learnbig.lx.it.pt/, introduzir o nome de "Utilizador" e "Password" para aceder ao sistema.
   a. Utilizar o botão "Doentes" para aceder à zona dos doentes
   b. Utilizar link "CRIAR FICHA" para criar uma ficha do doente na plataforma.
   c. Preencher os dados referentes a:
      i. "N. Utente" (será o número de processo hospitalar)
      ii. "Nome"
      iii. "Data de Nascimento"
      iv. "Sexo"
      v. "N. Doente". Neste ultimo campo surge num tom de cinzento o número de doentes já configurados, pelo que o número a introduzir deve ser o que surge somado de um.
   d. Após a introdução destes dados deve ser utilizado o link "GUARDAR" para gravar os dados do doente.
4. Após a criação da ficha do doente é necessário indicar os dispositivos que serão utilizados pelo doente na telemonitorização. Para tal utiliza-se o separador "Equipamentos". Deve-se escolher o "Tipo de Equipamento", na *DropDownList*, e usar o sinal + para indicar o equipamento associado ao doente. Os equipamentos devem ser adicionados **pela ordem indicada:**
   a. GearS3
   b. BITalino Plux
   c. Telemóvel
5. Colocar os dispositivos no doente
   a. Telemóvel dentro da bolsa, que deve ser colocado à cintura, próximo do "centro de gravidade". É conveniente garantir que o mesmo fica ajustado ao doente e que telemóvel não se move de forma independente.
   b. Relógio deve estar justo à pele do paciente (na posição habitual de um relógio normal, na mão não dominante), para se maximizar a leitura da frequência cardíaca e se garantir que não se move de forma independente
   c. Faixa deve ter a etiqueta com o número centrada no externo do doente, com o número na posição correta para quem olha para o doente

## Carregamento

- De 12 em 12 horas é necessário carregar o relógio e o power bank que acompanha o telemóvel, não é necessário desligar o relógio durante o carregamento
- Os dispositivos carregam durante aproximadamente 2h
- Após o carregamento é necessário voltar a colocar o relógio, e caso este se tenha desligado deve-se voltar a ligar o relógio sendo necessário voltar a clicar no icon
- Após o carregamento do power bank, este deve ser ligado ao telemóvel
- Quando o power bank estiver ligado ao telemóvel e o LED do power bank se apagar deve voltar a carregar-se o power bank (isto ocorre cerca de 1h depois de se ligar o power bank ao telemóvel)

## Finalizar a Recolha

1. Desligar todos os dispositivos
2. Aceder à plataforma em http://learnbig.lx.it.pt/, introduzir o nome de "Utilizador" e "Password" para aceder ao sistema.
   a. Utilizar o botão "Doentes" para aceder à zona dos doentes
   b. Selecionar o doente que está a finalizar a monitorização
   c. Aceder ao separador "Equipamentos" e clicar em desassociar cada um dos equipamentos no icon
   d. Terminar sessão na plataforma.

# B. Communication Protocol

# TelemOLD – Protocolo de Comunicação

## 1. Comunicação inicial: cliente-servidor

***Estrutura:***

_ _ _ _

1 2 3 4

1. **\ (char)** - inicio da mensagem
2. **(long)** – hora da comunicação
3. **(6 hex)** – mac address do dispositivo que pretende conectar com o servidor
4. **\ (char)** – fim da mensagem

***Exemplo:***

\ 1417162942424 0019B9FBE258 \

- Tipo de Mensagem "\ ... \": inicio de comunicação cliente-servidor
- Mensagem enviada no dia 28-11-2014 às 8h20
- Mac Address "00:19:B9:FB:E2:58"

## 2. Comunicação inicial: servidor-cliente

***Notas:***

Cada dispositivo configurado para o cliente irá originar uma mensagem diferente.

***Estrutura:***

_ _ _ _ _ _ _ _

1 2 3 4 5 6 7 8

1. **/ (char)** - inicio da mensagem
2. **(long)** – hora da comunicação
3. **(int)** – código do doente que se está a ligar
4. **(int)** – período de comunicação entre cliente e servidor (em segundos)
5. **(char)** – equipamento
6. **(6 hex)** – mac address do equipamento
7. **(char float)[]** – pares de parâmetro e valor referentes à configuração do equipamento
8. **/ (char)** – fim da mensagem

***Exemplo:***

/ 1417162942424 457 30 N 0088144D4CFB h30 o30 /

- Tipo de Mensagem "/ ... /": inicio de comunicação servidor-cliente
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- 30 – período de comunicação entre cliente e servidor (30 segundos)
- N - Nonin
- Mac Address "00:88:14:4D:4C:FB"
- h30 – HR com período de amostragem de 30seg
- o30 – SPO2 com período de amostragem de 30seg

## 3. Comunicação periódica: cliente-servidor

***Estrutura:***

_ _ _ _ _

1 2 3 4 5

1. **# (char)** - inicio da mensagem
2. **(long)** – data de registo dos parâmetros de actividade
3. **(int)** – código do doente
4. **(char[2] float)[]** – pares de parâmetro e valor referentes aos indicadores de actividade que os equipamentos registam. A primeira letra representa o dispositivo, e a segunda o parâmetro. Se a segunda letra estiver em letra minúscula serve de indicação ao servidor que o valor é pouco fiável. Por outro lado, caso seja uma letra maiúscula o valor é fiável.
5. **# (char)** – fim da mensagem

***Exemplo:***

# 1417162942424 457 Nh75 No98 C0 M1 #

- Tipo de Mensagem "# ... #": comunicação periódica cliente-servidor
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- Nh75 – HR com valor de 75 (pouco fiável)
- No30 – SPO2 com valor de 98 (pouco fiável)
- C0 – 0 counts (fiável)
- M1 – 1 mets (fiável)

## 4. Logs: cliente-servidor

***Estrutura:***

_ _ _ _ _ _

1 2 3 4 5 6

1. **? (char)** - inicio da mensagem
2. **(long)** – hora do log
3. **(int)** – código do doente
4. **(int)** – id do evento
5. **(char)** – equipamento
6. **? (char)** – fim da mensagem

***Exemplo:***

? 1417162942424 457 N 25 ?

- Tipo de Mensagem "? ... ?": logs cliente-servidor
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- N - Nonin
- 25 – id do evento

## 5. Alertas: servidor-cliente

***Estrutura:***

_ _ _ _ _ _

1 2 3 4 5 6

1. **! (char)** - inicio da mensagem
2. **(long)** – hora do alerta
3. **(int)** – código do doente
4. **(int)** – número de caracteres da mensagem de alerta
5. **(char[])** – mensagem de alerta
6. **! (char)** – fim da mensagem

***Exemplo:***

! 1417162942424 457 20 Mensagem do Servidor !

- Tipo de Mensagem "! ... !": alerta servidor-cliente
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- Mensagem do Servidor – mensagem de alerta

## 6. Feedback da Comunicação

Para cada tipo de mensagem enviada poderá existir um feedback do receptor, como por exemplo uma mensagem de acknowledge. Essa mensagem será composta por o caracter inicial e final de cada tipo de mensagem enunciada anteriormente, a hora da mensagem original, e um valor inteiro que representa a resposta.

***Estrutura:***

_ _ _ _

1 2 3 4

1. **(char)** - inicio da mensagem
2. **(long)** – hora da mensagem original
3. **(int)** – código do feedback
4. **(char)** – fim da mensagem

***Exemplo:***

# 1417162942424 0 #

- Acknowledge para a comunicação periódica
- Mensagem enviada no dia 28-11-2014 às 8h20
- 0 – Sucesso
- > 0 – Erros / Eventos inesperados

# C. Submitted paper on the Smartwatch HR estimation

# Smart-Wearables and Heart-Rate Assessment Accuracy

André F. Manso[1], Ana L. N. Fred[2], Rui C. Neves[3] and Rui C. Ferreira[4]

***Abstract*— Wearables and smart devices are making their way into people's life bringing constant connectivity. The inclusion of bio-sensors into these devices allows for a pervasive and constant monitoring of habits and also physical condition and physiological variables. In particular, seamless heart rate measurements can be of paramount importance in cardiac conditions, paving the way to personalized health care scenarios. However, reliability and accuracy of these sensor platforms remains an issue that is many times not properly addresses by manufactures. To shed some insight into this issue a smart device was placed against state-of-the-art equipment to assess its accuracy in heart rate measurements.**

## I. Introduction

In recent years, wearables have been taking their place in biological signal acquisition. Sensors are becoming smaller, lighter, and cheaper as computing power is evermore affordable and small-sized. In particular, the inclusion of biosensors in smart devices has seen a spike, with purposes like biometrics or sports performance tracking. In this context, Photoplethysmography (PPG) is a major tool, being one of the less invasive techniques in heart rate (HR) estimation which makes it a very good candidate for inclusion into wearable devices [1]. Most Smartwatches nowadays include one of these sensors to allow user heart rate monitoring.

Continuous health monitoring is a very promising field of medical development [2] and this type of devices can play an important role in pervasive medical data collection. This allows physicians to monitor their patients in real time and collect data over long periods, paving the way to personalized health care scenarios. In particular, heart rate can be used as a major clinical indicator for patients with heart diseases [3], [4], [5].

Despite all the advantages and promises [6], wearables have some drawbacks that greatly affect the quality of their HR estimates. Smartwaches, in particular, position the PPG sensor in the distal portion of the posterior forearm. This is a location where PPG signal is present but faint due to reduced concentration of blood flow [7]. Another major problem is signal corruption by motion artifacts. This conundrum is present because these devices tend to be heavy enough to have their own dynamics i.e. they move independently from the forearm by inertia. In addition, it is not comfortable to have the device too tight to the skin, and excessive pressure reduces superficial blood flow, thus further reducing signal to noise ratio (SNR). In fact smartwaches and wearable sensors have been studied many times [6], [8] and even its applicability as a source of clinical information has been proposed [9], [10], [4]. However, accuracy of this type of devices has been questioned [9], [11], [12] and error margins for HR estimates produced by smartwaches were proven considerable.

When developing a platform for remote patient monitoring, for example, it is very relevant to have information about performance and technical aspects of sensors and algorithms included into commercial systems that manufacturers are most commonly not whiling to share. This makes it harder to choose the sensor platform, fitness tracker, smartwatch, etc... that fits better with project requirements, as reliable information on how a specific device behaves may be very important for platform selection.

Being Samsung one of the great players in the smartwaches and wearables field, a very recent smartwatch sensor platform, Samsung Gear S3 [13], was tested against an ECG-based ground truth device to provide a characterization of the error in HR determination.

Results are presented in the form of absolute error (AE) as it is an easy to read quantity providing a more informative measure, making easier for physicians to have a perception of measurements and the confidence interval provided by any novel system based on these devices.

## II. Methodology

In order to evaluate the accuracy of HR estimates given by Samsung Gear S3, the later was used to record HR in different situations with several subjects with and without cardiac pathologies. Simultaneously a BITalino [14] and a medical-grade certified device were used to determine the HR from ECG data and provide a ground truth reference for Gear's values.

### A. Data Acquisition

*1) Samsung Gear S3 Frontier:* A custom app was implemented with sensor interaction being made through the manufacturer provided API [15]. Data was sent to an android smartphone via Bluetooth for storage. The API allows to retrieve raw PPG data as well as HR calculated by the smartwatch from PPG data using manufacturer's algorithm. The device produces HR estimations at 25Hz, PPG is collected at 25Hz and tri-axial accelerometry is collected at 100Hz.

*2) BITalino:* BITalino is a customizable sensor platform that was used to collect one derivation of ECG at 1000Hz. This device was proven to be a reliable sensor with performances similar to medically certified devices [16], [14]. Based on BITalino, a chest band was designed and built using an elastic and adjustable band that sits under the breasts. This device in this location allows for a low-noise ECG acquisition even when the subject is moving or exercising.

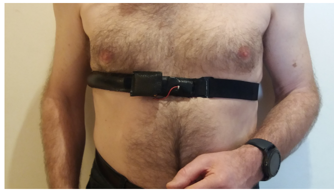


Fig. 1: Subject wearing chest band and smartwatch used for data acquisition.

*3) Hospital Equipment:* The device used is a Mortara XScribe 3.10.10 by Mortara Instrument [17] that collects 12 lead ECG, producing a very reliable HR estimate as it is a certified medical device.

### B. Experimental Protocol

To validate heart rate values calculated by Gear S3 three experiments were conducted:

- E1 - Consisted on acquiring data for short periods of time with subjects performing a specific activity:
  - Resting state, moving as least as possible (2min)
  - Walking at a regular pace (5min)
  - one subject was asked to pedal on a stationary bike at moderate pace (5min)
  - one subject was asked to run at moderate pace (2min)

  Some of these acquisitions were repeated to verify reproducibility and infer on system robustness.
- E2 - On this experiment, data was recorded for periods between 1 and 2 hours while subjects performed their usual activities during their daily lifes.
- E3 - The experiment consisted on collecting data from 3 hospitalized patients with various cardiac diseases during a stress test, taking place at hospital. These tests consisted of patients walking on a treadmill, increasingly fast until they are not able to continue, followed by a rest period. Patients ECG and energy consumption is monitored while executing the task. The procedure was performed according to Bruce protocol [18].

### C. Data Processing

After the experiments, data was processed in order to produce HR estimates from the acquired signals (PPG, ECG). To evaluate estimates quality, performance was measured having as ground truth HR values calculated from the ECG. For stress tests HR ground truth is provided by the hospital's equipment, while for the other experiments the ground truth is calculated from data collected by the BITalino chest band.

ECG data processing consisted on the following. First, the R peaks were detected using method proposed in [19], from which the instantaneous Heart Rate was computed using eq. (1). Since the smartwatch does not provide instantaneous HR estimates, but rather provides HR values in a 25Hz frequency, the ECG-based HR for consecutive, non-overlaping, 10s windows was determined as the median of the instantaneous HR values for each window. The HR provided by Gear S3 was also determined as the median of consecutive, non-overlaping, 10s windows from the 25Hz samples it returns.

$$HR_i = \frac{60 * f_s}{t_{R_{i+1}} - t_{R_i}} \tag{1}$$

$HR_i \rightarrow$ ith instantaneous HR
$f_s \rightarrow$ Sensor's sampling frequency
$t_{R_i} \rightarrow$ sample number of the ith R-peak

### D. Accuracy Measurement

The accuracy of the HR values produced by the smartwatch, and also the ones estimated from the ECG signal, was summarized as statistics of the absolute error (AE) e.g. mean absolute error (MAE) as defined in eq. (2).

$$MAE = \frac{1}{N}\sum_{i=0}^{N}\left|HR_i^{true} - HR_i^{est}\right| \tag{2}$$

$N \rightarrow$ nr. of HR values estimated for a subject
$HR_i^{true} \rightarrow$ Ground-truth value of instantaneous HR
$HR_i^{est} \rightarrow$ Estimated value of instantaneous HR

## III. Experimental Setup and Results

Data was collected from two separate groups performing different activities. Stress tests were undertaken by 3 patients with various cardiac conditions and ages 42, 47 and 50 years old. The other experiments enumerated in section II-B were performed by 3 volunteers all 23 year olds. In following figures and tables Si denotes subject i and Tj denotes jth trial.

After data collection from subjects, the first observation of the PPG signal revealed the vast influence of motion artifacts (MA). This is depicted in fig. 2. When the subject is completely immobile it is easy to find, by inspection, a correlation between ECG and PPG signal peaks (fig. 2a) corresponding to systolic peaks. The same is not true when the subject is moving, as signal corruption greatly affects PPG signal and correlation is no longer obvious and maybe not present at all (fig. 2b).

Observing fig. 5, it is visible that Gear S3 is not capable of keeping up with fast variations in HR and fig. 7 clearly illustrates the difficulty in determining HR accurately during movement, with a very large estimation error being present, until subject enters the resting phase of the stress test.

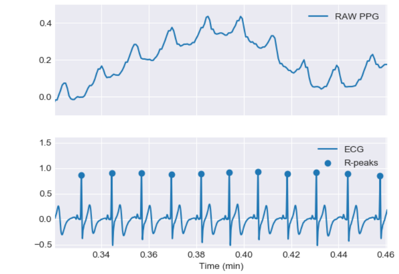
*This work was supported by the Portuguese Foundation for Science and Technology under grant PTDC/EEI-SII/7092/2014

[1]André F. Manso is with IST - Instituto Superior Técnico Universidade de Lisboa, Biomedical Engineering, Av. Rovisco Pais, 1, Lisboa, Portugal e-mail: andre.manso@ist.utl.pt

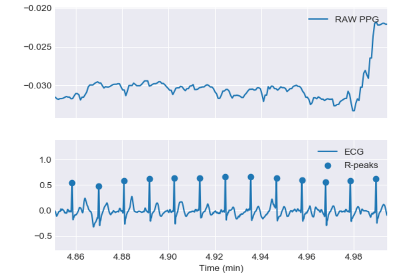
[2]Ana L. N. Fred is with Instituto de Telecomunicações, IST - Instituto Superior Técnico Universidade de Lisboa, Av. Rovisco Pais, 1, Torre Norte - Piso 10 1049-001 Lisboa, Portugal e-mail: afred@lx.it.pt

[3]Rui C. Neves is with CAST - Cons. e Apl. em Sistemas e Tecnologias, Lda., R. General Silva Freire 157 H, 1800-210 Lisbon, Portugal e-mail: rcneves@cast.pt

[4]Rui C. Ferreira is with the Cardiology Department of Hospital Santa Marta – CHLC, R. de Santa Marta 50, 1169-1024 Lisbon, Portugal e-mail: cruzferreira@netcabo.pt




(a) Raw signal while completely immobile. HR calculated from PPG (first curve): 75bpm; HR calculated from ECG (second curve): 79bpm.




(b) Raw signal while walking. HR calculated from PPG (first curve): 52bpm; HR calculated from ECG (second curve): 81bpm

Fig. 2: Segments of raw signal captured by the sensors in different conditions with the synchronous identified R-peaks.

### A. Short Acquisitions (E1)

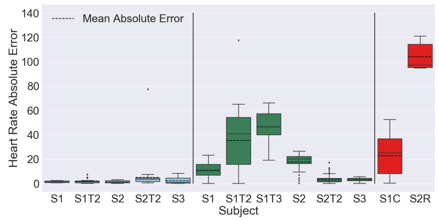



Fig. 3: Absolute Error of HR while performing different activities: (left) resting; (middle) walking; right cycling and running.

When analyzing the absolute error of HR determined from data collected while subjects were performing specific activities it is very obvious that motion highly corrupts sensor data and thus greatly damages accuracy. In fig. 3 is very clear a tendency to error increase as subjects go from resting to walking, cycling or running. Another thing that can be noted in fig. 3 and table I is the relatively large difference in the error values between subjects and trials. This may indicate low robustness of the system, as it is affected by sensor positioning and tightness and also by how the subjects move, as some individuals present more arm movement which can further disrupt accuracy.

TABLE I: Mean Absolute Error (MAE) of each experiment and activity averages.

| Activity | Subject | MAE | Average |
|---|---|---|---|
| **Resting** | S1 | 1.4 | 2.4 |
| | S1T2 | 1.8 | |
| | S2 | 1.3 | |
| | S2T2 | 5.1 | |
| | S3 | 2.2 | |
| **Walking** | S1 | 10.8 | 21.2 |
| | S1T2 | 35.3 | |
| | S1T3 | 46.4 | |
| | S2 | 17.7 | |
| | S2T2 | 3.4 | |
| | S3 | 3.3 | |
| **Cycling** | S1 | 22.8 | 63.4 |
| **Running** | S2 | 104.1 | |

### B. Daily life (E2)

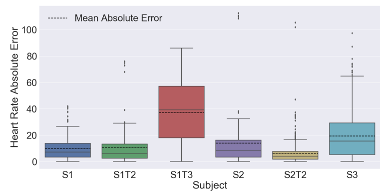



Fig. 4: Absolute Error of HR while subjects perform their usual life activities (office working, walking, eating, etc...).

TABLE II: Mean Absolute Error (MAE) of each subject during daily life activities.

| Activity | Subject | MAE | Average |
|---|---|---|---|
| **Daily-life activities** | S1 | 9.7 | 16.2 |
| | S1T2 | 10.8 | |
| | S1T3 | 37.1 | |
| | S2 | 13.9 | |
| | S2T2 | 6 | |
| | S3 | 19.3 | |

Table II and fig. 4 demonstrate clearly that during daily-life activities like walking or any activity that implies arm movement, Gear S3 has a poor performance as a sensor platform and in certain occasions it is not even possible to detect any type of signal trends. In a medical context, this could lead to gross errors when trying to diagnose or follow a patient, or it would require further tests which renders the pervasive monitoring less efficient or even counterproductive.

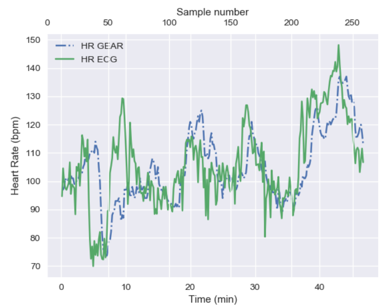



Fig. 5: Example of HR curves obtained during daily life activities.

### C. Stress Test (E3)

As expected, results during a stress test, where the subject is moving with some intensity, HR determination error is very large and as can be seen in fig. 7, error is specially large during exercise part of the stress test, supporting hypothesis of motion artifacts corrupting the signal. Error during these tests reached $>$100bpm wich renders this sensor platform completely unfit for this kind of context.

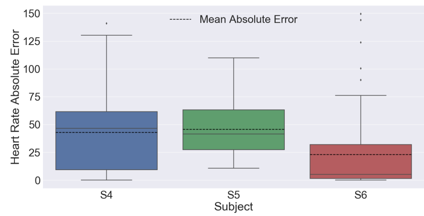



Fig. 6: Absolute Error of HR during stress test.

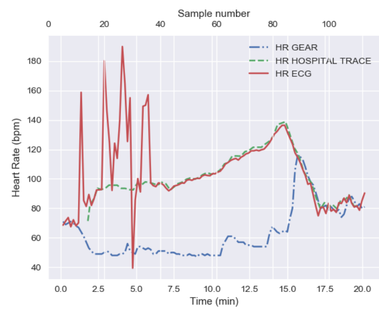



Fig. 7: Example of HR curve obtained during stress test.

## IV. Discussion

Analyzing the results obtained, the first and main aspect to point out is that the PPG signal collected with Samsung Gear S3 is very easily corrupted by motion artifacts. Even with mild movement, the signal is not reliable and systolic peaks can no longer be identified. This renders the device unfit for continuous monitoring, since HR estimation from PPG data has very low accuracy.

Probable causes for this poor performance are the fact that only one source of light and one detector are present, the weight of the device and the development cost.

Samsung's Gear S3 sensor has a single LED with a single wavelength. Some other PPG sensors use two LEDs and some two wavelengths which eases the signal processing and improves performance. The existence of a single source of light, with only one wave length, implies that less information is gathered and the signal processing options are somehow limited.

Concerning the weight, it must be taken into account that this is a very complex device that can perform endless tasks with native support for 3rd party apps. All these functionalities and capabilities make the device heavier and lager, thus, making it harder to keep comfortably and tightly secure to the wrist without it having its own dynamics. This is a problem as the relative movement between the sensor and the subject is the sole cause for movement artifacts in the signal. In addition the positioning of the sensor in an area with a relatively low SNR greatly increases the impact of motion artifacts. Also this device has the sensor at the same level of the watch back plate, whereas other devices place the sensor in an elevation of the back plate, which makes the sensor to be slightly pressed against the skin and increases SNR, and thus, accuracy.

Due to poor quality PPG signal acquired, several algorithms were used in an attempt to get better HR estimates. Algorithms used included adaptive filtering with and without Laguerre expansion [20], [21], signal separation by sparse signal reconstruction [22], [23] and onset detection [24]. A total of 8 algorithms were used to process the PPG and accelerometry data coming from S3 to produce estimates of HR. However, the results obtained using all this algorithms performed equally bad, or even worse, than Gear's algorithm and for this reason, they were not mentioned previously. This clearly indicates that elevated error in Gear's HR estimations is probably related with a low quality signal and not with a poorly performing algorithm.

## V. Conclusions

In this work the accuracy of HR estimates produced by a wearable smartwach was studied using by Samsung Gear S3. Analyzing the results obtained, it is fair to conclude that this system, and this type of commercial wearables, are not yet fit for long term monitoring to be used by physicians in either diagnostics nor patient follow up. This type of application would require a precision larger than what is possible to achieve with current devices and algorithms.

As researchers and manufacturers continue to improve device and algorithms, it is expectable that in a near future patients and physicians benefit from more accurate devices to provide better health-care.

## References


[1] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiological measurement*, vol. 28, no. 3, p. R1, 2007.

[2] C. S. Bloss, N. E. Wineinger, M. Peters, D. L. Boeldt, L. Ariniello, J. Y. Kim, J. Sheard, R. Komatireddy, P. Barrett, and E. J. Topol, "A prospective randomized trial examining health care utilization in individuals using multiple smartphone-enabled biosensors," *PeerJ*, vol. 4, p. e1554, 2016.

[3] K. H. Shelley, "Photoplethysmography: beyond the calculation of arterial oxygen saturation and heart rate," *Anesthesia & Analgesia*, vol. 105, no. 6, pp. S31–S36, 2007.

[4] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiological measurement*, vol. 28, no. 3, p. R1, 2007.

[5] H. B. Abramowitz, L. A. Queral, W. R. Flinn, P. F. Nora, L. K. Peterson, J. J. Bergan, and J. S. Yao, "The use of photoplethysmography in the assessment of venous insufficiency: a comparison to venous pressure measurements," *Surgery*, vol. 86, no. 3, pp. 434–441, 1979.

[6] R. Rawassizadeh, B. A. Price, and M. Petre, "Wearables: Has the age of smartwatches finally arrived?" *Communications of the ACM*, vol. 58, no. 1, pp. 45–47, 2015.

[7] S. Lee, H. Shin, and C. Hahm, "Effective ppg sensor placement for reflected red and green light, and infrared wristband-type photoplethysmography," in *Advanced Communication Technology (ICACT), 2016 18th International Conference on*. IEEE, 2016, pp. 556–558.

[8] G. Bieber, M. Haescher, and M. Vahl, "Sensor requirements for activity recognition on smart watches," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2013, p. 67.

[9] F. El-Amrawy and M. I. Nounou, "Are currently available wearable devices for activity tracking and heart rate monitoring accurate, precise, and medically beneficial?" *Healthcare informatics research*, vol. 21, no. 4, pp. 315–320, 2015.

[10] A. Carpenter and A. Frontera, "Smart-watches: a potential challenger to the implantable loop recorder?" *Europace*, vol. 18, no. 6, pp. 791–793, 2016.

[11] J. Parak and I. Korhonen, "Evaluation of wearable consumer heart rate monitors based on photopletysmography," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, 2014, pp. 3670–3673.

[12] E. E. Dooley *et al.*, "Measuring the validity of self-monitoring heart rate and activity tracking wearables," Ph.D. dissertation, 2016.

[13] Samsung, "Samsung Gear S3." [Online]. Available: http://www.samsung.com/pt/wearables/gear-s3/

[14] J. Guerreiro, R. Martins, H. Silva, A. Lourenço, and A. L. Fred, "Bitalino-a multimodal platform for physiological computing." in *ICINCO (1)*, 2013, pp. 500–506.

[15] T. Project, "Tizen Sensor API," 2012. [Online]. Available: https://developer.tizen.org/development/api-references/native-application?redirect=/dev-guide/2.3.1/org.tizen.native.mobile.apireference/group__CAPI__SYSTEM__SENSOR__MODULE.html

[16] D. Batista, H. Silva, and A. Fred, "Experimental characterization and analysis of the bitalino platforms against a reference device," in *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*. IEEE, 2017, pp. 2418–2421.

[17] M. Instruments, "Mortara XScribe." [Online]. Available: https://www.mortara.com/products/healthcare/hospitals/cardiac-stress-testing/xscribe/

[18] M. W. Luong, M. Ignaszewski, and C. Taylor, "Stress testing: A contribution from dr robert a. bruce, father of exercise cardiology," *British Columbia Medical Journal*, vol. 58, no. 2, pp. 70–76, 2016.

[19] A. Lourenço, H. Silva, P. Leite, R. Lourenço, and A. L. Fred, "Real time electrocardiogram segmentation for finger based ecg biometrics." in *Biosignals*, 2012, pp. 49–54.

[20] P. T. Gibbs, L. B. Wood, and H. H. Asada, "Active motion artifact cancellation for wearable health monitoring sensors using collocated mems accelerometers," in *Proc. SPIE*, vol. 5765, 2005, pp. 811–819.

[21] L. B. Wood and H. H. Asada, "Active motion artifact reduction for wearable sensors using laguerre expansion and signal separation," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. IEEE, 2006, pp. 3571–3574.

[22] Z. Zhang, "Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction," *IEEE transactions on biomedical engineering*, vol. 62, no. 8, pp. 1902–1910, 2015.

[23] Z. Zhang, Z. Pi, and B. Liu, "Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 522–531, 2015.

[24] R. Blazek and C. Lee, "Multi-resolution linear model comparison for detection of dicrotic notch and peak in blood volume pulse signals," *Anal Biomed Signals Images*, vol. 20, pp. 378–386, 2010.