Real-Time Pervasive Monitoring System for Ambulatory Patients

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

Abstract— **Personalization is becoming an evermore present concept in healthcare. The amount of information it is possible to collect from a patient has been increasing in recent years and this allows for the development of better and more targeted clinical and therapeutic approaches. With this in mind, a novel data acquisition system is proposed, allowing to monitor patients inside and outside of hospital environment for long periods of time. It is completely generic in what sensors and variables, it collects, and makes possible for medical teams to monitor patients remotely and in real-time. The system was developed in collaboration with Hospital de Santa Marta - CHLN, Lisbon and there was tested on patients with cardiac pathologies.**

I. INTRODUCTION

In recent years, pervasive monitoring and personalized medicine, are two of the most common words when describing the future health-care. Physicians, engineers, patients and caregivers are evermore eager to have better and more efficient treatments and approaches, that are tailor made to best fit each situation. At the same time, the continuous collection of information about one's activities, symptoms and physiological parameters, is becoming common, as sensors are being included into our daily lives. This easy access to information about a patient's status, activities and even preferences, may constitute a very useful tool when diagnosing, treating and tracking the evolution of one's condition.

This work describes a novel system, designed to pervasively monitor patients for long periods of time in and outside of the hospital. It was based on a previously used system, to which new features and functionalities were added [1]. 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, creating the tools 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.

One of the main aspects of the implemented architecture is versatility. 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 50bpm for more than 5 minutes.

This 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.

Tests took place with patients suffering from cardiac diseases at Hospital de Santa Marta - CHLC (Lisbon), where 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 commonly used in their practice to diagnose and track patients [3], [4]. Variables to be collected determined the type of sensors needed.

II. SYSTEM ARCHITECTURE AND FUNCTIONALITIES

This system is based on two components: a smartphone, that is responsible for interacting with the remote sensors, connected via Bluetooth, and relaying information to a central server; and the server, that stores data and manages the interaction between the user interface, the database, and the smartphone. The system is completely versatile in what remote sensors are connected to the smartphone, as long as they are capable of sending their data as a stream of numerical values through Bluetooth. Figure 1 illustrates the connections between the various system components.

Data acquisition starts with the medical personnel configuring, through the web server, acquisition parameters. Parameters that can be configured are:

- The types of device currently available in the system (smartphone, smartwaches, etc...)
- What devices are to be given to each the patient;

*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

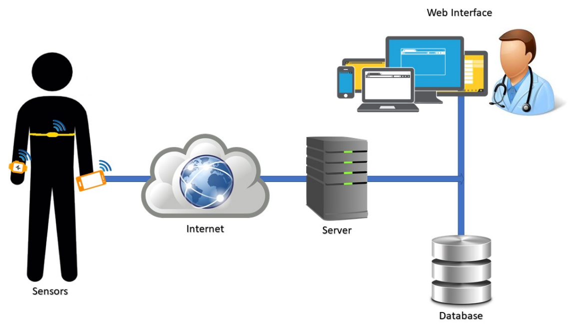


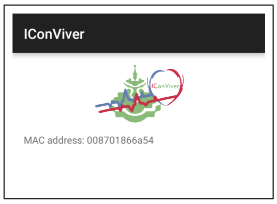
Fig. 1: System's architecture and connectivity of devices. Bluetooth connects the smartphone and remote sensors. The smartphone connects to the server through a mobile web connection.

- What variables are to be collected from each device (e.g. ECG), and with what frequency they should be sent to the central server;
- The variables that are expected to be received on the server (that may not be the ones collected, some processing can occur e.g. HR extracted from ECG) 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;

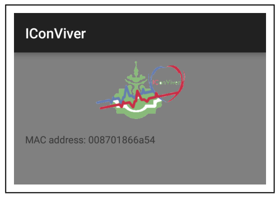
Each remote device is identified by a universally unique identifier (UUID), and each device is assigned as part of a type of devices e.g. smartphones. the number and identification of each family is completely configurable, but a single device of each type can be assigned to each patient simultaneously. After this initial configuration, when turned on, the smartphone queries the server for this information and starts connecting to the designated remote sensors, if any configured, and acquisition starts. Until the end of the study data is continuously sent to the server from which the physician can analyze it in real time. Data flow between the smartphone and the remote sensors is made using Bluetooth and the connection between the smartphone and the central server is accomplished using a TCP socket [5]. During and after the data acquisition, medical teams can monitor the patient's incoming data through the web interface. They can choose the time frame and the variables to be displayed in a static frame, or they can choose to see incoming data in real-time. Other type of information is also visible as events like connection to remote sensor lost, low battery detected or smartphone turned off.

A. *Smartphone*

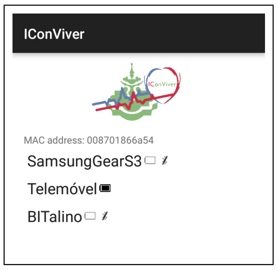
In this system, the smartphone acts as a central hub for collecting, storing, processing and relaying information between te various sensors and the central server, with all this being managed by a custom app. Incoming raw data from the sensors is stored into a local SD card, if necessary, processing occurs, applying the required algorithms, accordingly with the variables to be extracted, and finally, after processing,



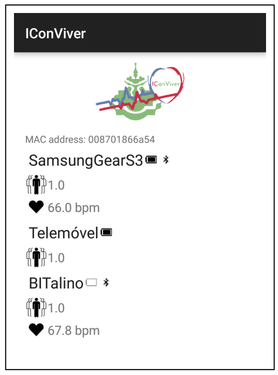
(a) Display at startup showing device's UUID



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



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



(d) 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.

Fig. 2: Android app screens in various states of acquisition.

data is sent to the central sever in the form of more informative clinical variables as per table I.

Raw data received through Bluetooth is buffered and every 10s it is compressed and stored in memory, and if necessary processed. As data samples are acquired at different rates from each device's sensors and a time stamp is associated with each sample to facilitate the time location of each one, and make easier temporal alignment of the overall signals.

The app was implemented using Android Studio® 3.0.1. All functions were designed to be as computationally efficient as possible to minimize loss of data and maximize battery life. The app was designed as a replacement for the android home screen, so, when the smartphone is turned on, the app will automatically initiate and shown on screen, to facilitate the use of the device by patients.

Data coming from each type of sensor is acquired, processed and stored by an independent thread to avoid clogging the processor and Bluetooth reading buffers. This aspect also eases the inclusion of new sensors and devices, as it allows to completely isolate the implementation of communication protocols for different devices.

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 all devices.

Optionally, using the web interface, medical teams 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 after it 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.

B. Server and Web Interface

The central server is responsible for serving the web interface and storing all the data and settings into an SQL database. The software is based on the .NET framewok and Windows SQL Database, and was designed to run in a windows machine. It is also responsible for awaiting and establishing socket connection with the smartphones.

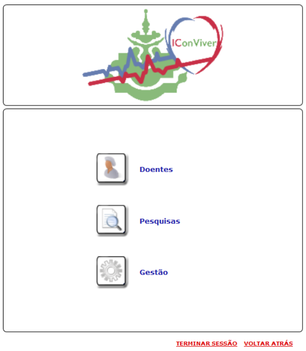


Fig. 3: Initial panel after login into the system's web interface. Here one can see patient information; search for specific data of a particular patient; configure all device and acquisition parameters.

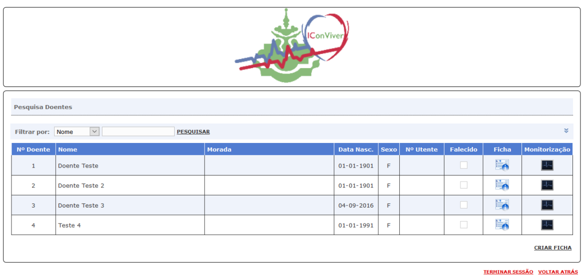


Fig. 4: Screen showing the patients enrolled in the system. From here one can access all the information and collected data from each patient.

System works with three major components:

- **Web interface -** A website, built ASP.NET, is 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, in fact, has two different subcomponents that ensure the communication between remote devices and the database.

Fig. 5: Data visualization screen where one can choose the variables to see and their time frame.

  - One 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.
  - Other component is 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 to 5 show some examples of the implemented web user interface, from where medical teams can control and configure all stages of the patient monitoring.

III. SYSTEM TESTING

The system was used to monitor cardiac patients, and relevant variables chosen were HR and METs. To achieve this, alonside the smartphone, two other devices were used, a Samsung Gear S3 smartwatch [6] and a BITalino [7] based chest-band. Table I describes the variables collected by each device and what are the clinical indicators extracted from the data after processing. Figure 6 shows a patient wearing the system's sensors.

Due to the multiplicity of devices used, several independent software components were developed to interact with each device and with the various sensors.

A. Remote Devices and Sensors

*1) **Smartwatch**:* The smartwatch was chosen for being a very convenient and comfortable package, as it can replace the watch many use daily, and, at the same time, includes many useful sensors described in[8]. In particular, this smartwatch can monitor the patient's HR using data collected from

TABLE I: Variables acquired by each device and the corresponding information extracted. HR - Heart rate; CD - Chest Deflection; PA - Physical Activity (measured in METs); RR - Respiratory rate

Device	Variables	Sampling Rate (Hz)	Clinical Variable
Smartwatch	PPG	25	HR
	HR	25	HR
	ACC	100	PA
BITalino	ECG	1000	HR
	ACC	1000	PA
	CD	1000	RR
Smartphone	ACC	100	PA

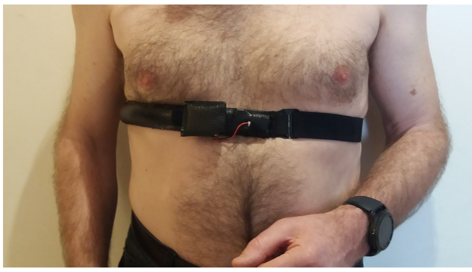


Fig. 6: Subject wearing chest band and smartwatch used for data acquisition.

a photoplethysmography (PPG) sensor [9], that is processed using a manufacturer's algorithm in order to determine HR. Besides PPG it can also collect tri-axial accelerometry data, that can then be processed to estimate the patient's physical activity.

A custom app was implemented to extract the information from the sensors, using a manufacturer provided API [8], and send it via Bluetooth to the smartphone. The sampling rate varies with the type of sensor as per table I. Data is sent in packets containing 10s of data from the PPG and accelerometry sensors and also the value of HR determined by the manufacturer's algorithm.

The developed app does not have a graphical user interface (GUI) as its only purpose is to collect and send data to the smartphone. The only interactions with the user occur via vibratory and sound notifications triggered by a loss of Bluetooth communication with the smartphone or a low battery level.

*2) **BITalino**:* BITalino was used to collect various signals as mentioned in table I in the form of a chest band that collects various signals and sends them to the smartphone via BT. The choice of this device is justified by the very good ECG signal quality acquired [10], [11], [7], and for being a very versatile and customizable sensor platform, which can collect several clinically relevant variables.

A chest band was built in order to accommodate the sensors described in table I with a comfortable and practical form-factor. This was a major concern as patients comfort was very important to ensure success of the system as it was intended to be used for long periods of time.

The base to the chest band was a standard commercial product band from Polar® used for sports tracker devices that includes conductive pads that rest in skin contact acting as electrodes.

B. Data Processing

*1) **Real Time Estimation Heart Rate from ECG**:* Once BITalino inside the chest band acquires ECG, and the target variable is HR, data processing must occur in order to determine HR from the collected ECG.

To accomplish this, an algorithm was implemented to identify R-peaks in the ECG signal, and from them calculate the HR. This algorithm had to be as simple and reliable as possible in order to limit the exhaustion of the smartphone's battery, which would harm system's usability, and also to maintain reasonable computing time, as a computationally heavy algorithm could clogg the processor and lead to delay in acquisition and loss of data. The proposed algorithm is based on the segmenter described in [12] although some modifications were implemented.

The implemented algorithm is as follows.

The ECG signal is processed in non-overlaping 10s widows, and the first stage is to filter it using a 5-40Hz 5th order bandpass Butterworth filter [13] and then an adaptive amplitude threshold is applied to locate R-Peaks. If a local maximum has an amplitude higher than this threshold, it is considered an R-Peak, otherwise it is classified a noise peak.

Threshold is initialized using the value with 98% of the cumulative probability function, i.e. the histogram, of the ECG amplitude over the first 10s window. This value is calculated as an initial approximation of the peaks amplitude on that window as defined by eq. (1).

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.

Initialization:

$$\begin{cases} THR = \arg_x \left(\frac{F_{ECG}(x)}{\lim\limits_{y \to +\infty} (F_{ECG}(y))} = 0.98 \right) \\ SP = THR \\ NP = 0.5 * THR \end{cases} \tag{1}$$

Iteration step:

$$\begin{cases} SP = 0.125 * P + 0.875 * SP & \text{if } P > THR \\ NP = 0.125 * P + 0.875 * NP & \text{if } P < THR \\ THR = SP - 0.25 * (SP - NP) \end{cases} \tag{2}$$

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

If no peaks are found in a 2s segment, which would correspond to a HR below 30bpm, the search is backtracked and threshold's value is lowered to 90% of current value. In an analogous way, if two peaks are found in less than

100ms, which is physiologically impossible [14] 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 local maxima classified as R-peaks in the 10s window is used to produce an estimate of instantaneous HR according to eq. (3). To improve robustness, instead of returning instantaneous HR, a single value is returned for the 10s window, meaning a 10Hz HR estimation is produced. To do this, the median of the various HR (calculated from each pair of R-Peaks) is taken, improving performance based on the assumption of HR being relatively constant inside the 10s window.

If the final estimate falls outside the 30-220bpm range, the calculation is rejected and a special error value of -3bpm is returned.

$$HR_i = \frac{60 * f_s}{t_{R_{i+1}} - t_{R_i}} \tag{3}$$

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

*2) **Data Compression**:* Raw data from all the sensors is stored into the smartphone's SD card as text files. There is a separate file created for each connected remote device and each sample of data is saved with the timestamp taken at the acquisition moment. Data vectors to be saved have the format [Timespamp, S1, S2, S3] where S1, S2 and S3 are samples from 3 sensors acquired at the same time.

Given the limited memory, battery capacity and computational power of smartphones, the compression algorithm should be simple (and hence of fast computation), and lossless (in order to allow future reliable data processing)

taking these requirements in mind the following data compression algorithm was designed and implemented: Data is saved into the files in 10s batches and the first value of each batch is stored as a raw value to make the recovery of the original data possible, and thus ensuring this method is lossless. The following samples are the time derivative of each sensors values.

To recover a single sample from the file, one must integrate the saved values up to that sample according with eq. (5).

$$X_i^{coded} = \begin{cases} X_i^{original} & i = 0 \\ X_i^{original} - X_{i-1}^{original} & i \neq 0 \end{cases} \tag{4}$$

$$X_i^{decoded} = \sum_{j=0}^{i} X_j^{coded} \tag{5}$$

$X_i^{coded} \rightarrow$ 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

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
...
```

Corresponding to the raw:

```
1508325806082;8;268;5418
1508325806092;8;338;6592
1508325806102;-2;401;772
...
```

When real time raw data is being sent to the central server, further compression is needed to reduce mobile data plafond requirements and also to limit battery expenditure, as the constant relaying of information requires much energy. However algorithm's efficiency is still a major priority as battery life is also very much affected by the algorithms' complexity. With this in mind, a simple Huffman codding [15] was implemented using a static code for all subjects, as statistical properties of data are approximately invariant with subject. This was concluded by 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 codding is performed after time-difference operation to encode each digit of the values obtained from the first step.

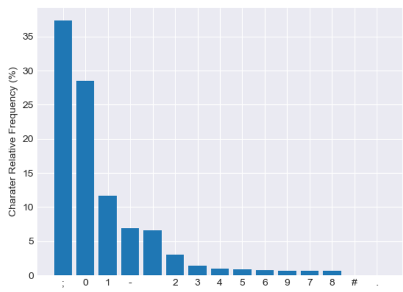


Fig. 7: Character relative frequency obtained from all the data collected from 4 different subjects for at least 30min each.

Figure 7 shows the very clear differences between characters' relative probability, which favours the use of this kind of codding. With this character count the codes from table II were determined and each character was switched by their corresponding code when being written to files.

Table III shows the compression ratios, as defined in eq. (6). These ratios were estimated from data collected from several patients, and they clearly show the efficacy of the very simple techniques utilized. Taking into consideration the very light computional burden these methos imply, they allowed to compress data into a format almost 8 times smaller, making possible for the system to be used for longer periods without memory shortage, even storing all the raw data from all sensors.

TABLE II: Codes determined for each character based on counts from fig. 7. [15]

Charachter	Code	Charachter	Code
';'	'0'	'5'	'1110010'
'0'	'10'	'6'	'11101011'
'1'	'1111'	'7'	'11101000'
'-'	'1101'	'8'	'111010101'
'\n'	'1100'	'9'	'11101001'
'2'	'111011'	'#'	'1110101001'
'3'	'111000'	'.'	'1110101000'
'4'	'1110011'		

$$CompressionRatio = \frac{UncompressedSize}{CompressedSize} \tag{6}$$

TABLE III: Memory usage when data is stored as a raw sensor value, as time-differences and using Huffman coding. Compression ratios are calculated according to eq. (6)

Device	Memory Usage (MB/day)		
	Raw	**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**
Compression Ratio	—	**4.43**	**7.55**

*3) **Physical Activity METS estimation**:* METS estimated using rules presented in [2] and refined in [16].

Counts estimation from accelerometry data was made according with 7.

$$Counts_{10s} = \sum_{i=1}^{10*f_s} |ACC_{i-1} - ACC_i| \tag{7}$$

ACC_i → ith accelerometry sample
f_s → Accelerometry sampling frequency

This metric allows physicians to estimate physical activity of the patient.

IV. Conclusion

The system was put into practice at Hospital de Santa Marta (Lisbon), with sensor technology and overall interfaces receiving a very positive feedback from both medical staff and patients.

Versatility was the most prized feature as the easy inclusion of different sensors, make it suitable for potentially all fields of medicine and in some sense, also make it timeless, as the same system can be used as sensor technology evolves.

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.

V. Acknowledgments

We thank Filipe Mariano for providing a very useful software basis and also some precious help on understanding it to ease the implementation of the proposed system.

References

[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*, vol. 20, no. 7, pp. 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*, vol. 100, no. 4, pp. 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*, vol. 42, no. 5, pp. 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*, vol. 132, no. 4, pp. 726–732, 1996.

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

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

[7] 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.

[8] 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

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

[10] A. L. A. L. N. F. R. C. d. N. R. F. H. Silva, C. C. Carreiras, "Off-the person electrocardiography: Performance assessment and clinical correlation," *Health and Technology*, vol. 4, no. 4, pp. 309–318, 2015.

[11] 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.

[12] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, March 1985.

[13] I. W. Selesnick and C. S. Burrus, "Generalized digital butterworth filter design," *IEEE Transactions on signal processing*, vol. 46, no. 6, pp. 1688–1694, 1998.

[14] H. Tanaka, K. D. Monahan, and D. R. Seals, "Age-predicted maximal heart rate revisited," *Journal of the American College of Cardiology*, vol. 37, no. 1, pp. 153–156, 2001.

[15] M. Sharma, "Compression using huffman coding," *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, no. 5, pp. 133–141, 2010.

[16] 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*, vol. 42, no. 5, p. 1029, 2010.