Biotelemetry System for Remote Monitoring of Cardiac Signals and Temperature Using Social Networks

Melissa Montalvo, Andrea García(✉), Julio Montesdeoca, and René Ávila

Universidad Politécnica Salesiana, Cuenca, Ecuador
{mmontalvoll,agarciacl}@est.ups.edu.ec,
{jmontesdeoca,ravila}@ups.edu.ec

Abstract. Patients suffering from heart problems such as arrhythmia, failure and prone to heart attacks, need regularly monitoring of their vital signs. It is described in this article an inexpensive, free software based, easy-to use prototype of a telemetry system, applied for monitoring biomedical signals using a wifi module, a photoplethysmograph and an infrared temperature sensor; the measurements obtained by this sensors are transmitted on real time to an Android application and to an online platform that can be accessed anytime. The prototype also includes an alarm system that sends a message to the phone application, and by using social networks such as Twitter, timely notifies the treating doctor and respective relatives whenever a critical level on the patient's body signals is detected.

Keywords: Biotelemetry · Heart rate streaming · Remote monitoring · Social Networks

1 Introduction

In recent years, monitoring of body signals and activities represents an important resource in multiple fields that are focused on the development of new technologies to improve human's health. Biotelemetry helps this purpose by acquiring biosignals, avoiding to disturb the patients in their regular activities, and transmitting them from a remote spot to a location where is interpreted by a specialist [1, 2].

The development of this system will constitute a reliable constant monitoring of cardiac and temperature signals, target to people who are predisposed to seizures, cardiac arrhythmia or hypertension. Previous prototypes of cardiac monitoring use wireless communication and mobile network for signal transmission [3], as well as PC communication and data transfer to Arduino devices [4]; the described prototype explores new uses for Social Networks as health alarms and urgent notifications that can be easily read by attending physicians and relatives of the patient; it also implements a free software and Internet of Things technology to transmit the biosignals to an online platform, where the data is available at all times and from any device connected to the web. This information, and the body signals graphics, can help to learn more about the symptoms of the patient, signals evolution and health status [5].

B. Mandler et al. (Eds.): IoT 360° 2015, Part II, LNICST 170, pp. 62–69, 2016.
DOI: 10.1007/978-3-319-47075-7_8

Three phases were established for the process of data acquisition; the first one consists on heart rate measuring to sense the patient's heart signals through a cardiac pulse sensor and determine critical levels on the collected data, the next phase involves sensing the patient's body temperature by the infrared temperature sensor Tmp006, finally takes place the transmission of the data via Wifi module, to further be sent to a cell phone with Android OS which will access to it by a develop APK. Additional to these procedures, the project also focuses on a code to provide alerts in the case that the patient presents critical heart rate or temperature levels.

2 Heart Rate and Temperature Acquisition

The Pulse Sensor essentially works as a photoplethysmograph, which is a well-known medical device used for non-invasive heart rate monitoring. Sometimes, photoplethysmography measure blood-oxygen levels (SpO_2), sometimes, as in this case, they don't [6]. The heart pulse, out signal of the photoplethysmograph, is a voltage analog fluctuation, and it has a predictable wave shape as shown in Fig. 1. The depiction of the pulse wave is called a photoplethysmogram, or PPG. The latest hardware version of the sensor used for this prototype, Pulse Sensor Amped, amplifies the raw signal of the previous Pulse Sensor, and normalizes the pulse wave around V/2 (midpoint in voltage) [7]. Pulse Sensor Amped responds to relative changes in light intensity, if the amount of incident light on the sensor remains constant, the signal value will remain at (or close to) 512 (midpoint of ADC range). In case of detecting more light, the signal goes up, and opposite with less light.

The temperature measures were taken using the TMP006 sensor which is characterized by the fact that uses a thermopile to measure the variable, it also captures the infrared energy being emitted for the surface of the object so it doesn't require any contact with it, making this kind of sensor desirable for not disturbing to the user.

A thermopile consists in a big amount of thermocouples aligned in parallel and wired in series, in a way that every microvolt-level signal proportional to the

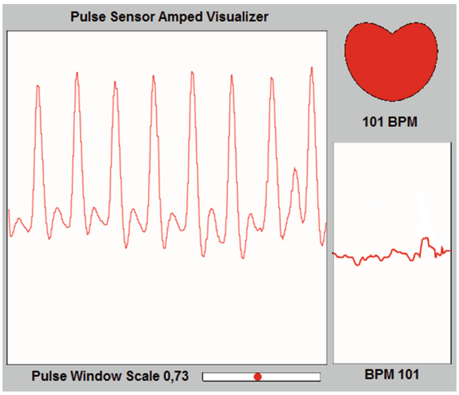

Fig. 1. Pulse sensor signal test using serial streaming and processing software.

temperature produced by each thermocouple will be part of the sum of these outputs [8]. The sensor has a better performance with non-reflective surface bodies which emissivity coefficient = 1, but due to the system is applied for body temperature the results of the sensor will be accurate given the emissivity coefficient of human skin has a value of 0.99 [9].

3 Methods and Materials

The primary code of the pulse sensor uses the Timer 2 of the ATmega328 to generate the interruption every 2 ms, this timer disables the PWM outputs number 3 and 11 of the Arduino Uno Board, which are used by the Wifi module, due to this inconvenient the Timer 1 of the sensor was used. Timer 1 disables PWM outputs 9 and 10, for this reason the Wifi module connection assigned to pin 10 was substituted by pin 6. Notice in the code and Fig. 2.

The best way to execute remote monitoring is through cloud web servers oriented to Internet of Things, based on the connection of physical devices to the web, allowing possible to access remote sensor data and remote controlling of physical world, this mash-up of captured data and the web, gives rise to new synergistic services that go beyond the services that can be provided by an isolated embedded system [10].

In order to plot the obtained data, it was proposed to work with different online platforms such as the case of Xively, ThingSpeak and Plotly Internet of Things Service; experimenting with each of them, various difficulties arose; for Xively, the access to their system requires to fill a form for a free developer account, therefore it was necessary to register as a company and request access to the servers.

In the case of ThingSpeak, a platform oriented to the Internet of Things streaming, its interface offers the creation of different channels and display charts. Using this tool the supposed real-time plotting was executed with a considerable delay producing a lack of determination and not trustable results, that leads to a disadvantage when

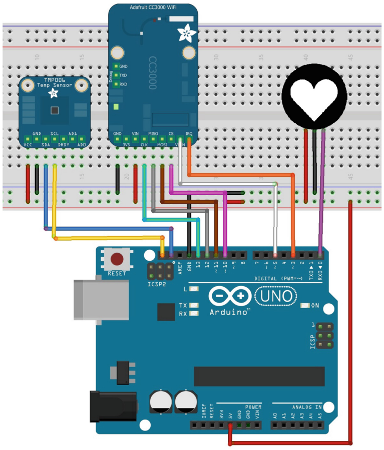

Fig. 2. Wifi module Adafruit CC3000, TMP006 sensor and pulse sensor wiring scheme.

improving human's health. Considering all this parameters, it was decided to work with Plotly platform, specialized on analytics and data visualization; this platform has a compatible library with the Wifi module Adafruit CC3000, which facilitated the process of streaming data and worked according to the objectives.

Although Plotly streaming was satisfactory, it was also explored the possibility of transmitting the data to our own Web Server. The data collected in the Arduino Board and its transmission to a server includes several steps; the main link of the data is performed by the statement aREST. The aREST framework was created to give RESTful interface to several embedded boards & platforms, in a nutshell, the library allows you to send commands to a given board running aREST, provoke an action (or just get some data), and send data back in a JSON container [11]. This requires running the code in a 1.5.7 Arduino IDE Version or higher.

The interface is develop in Javascript on the platform Express Jade, installation and code execution took place in Node.js Software. Express Jade tool was used to automatically generate code in html5, the variables exported to our website page were the magnitude value of the beats and BPM, and transmission status, as shown in Fig. 3.

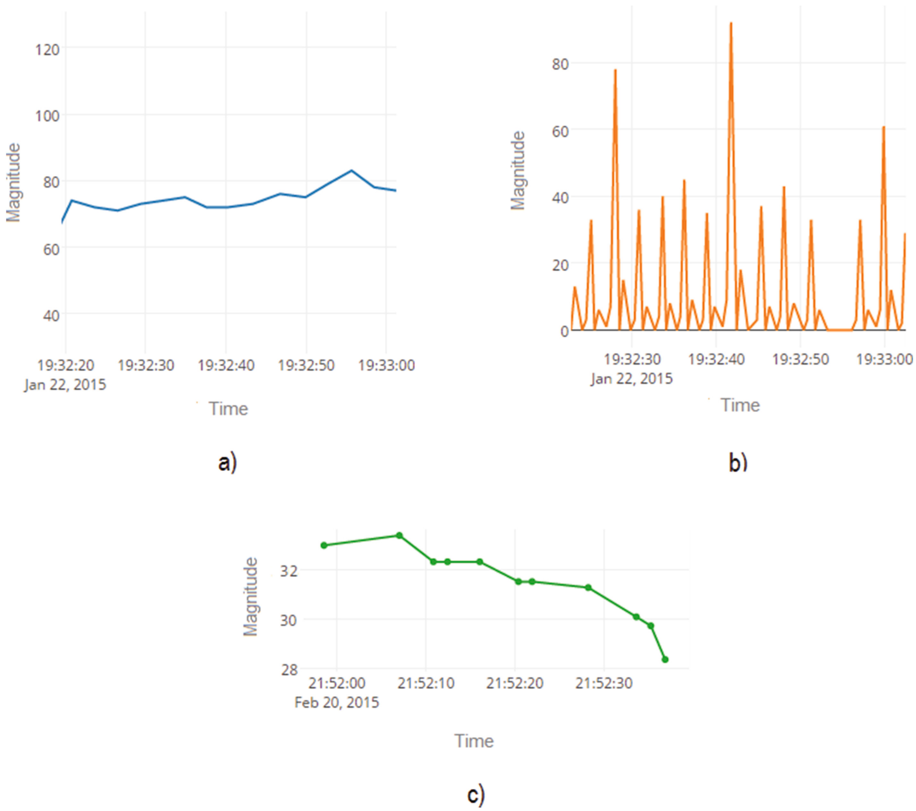

Fig. 3. BPM values (a), heart signal (b) and temperature values in (Celsius) (c) stream graph on Plotly web service.

The transmission and data displayed were achieved without inconvenience in the local computer IP address on port 3000. It has to be point out that in order to maintain the connection, the data refreshing should be done with a minimum delay of 250 ms, causing significant data loss in the heartbeat magnitude.

4 Connections and Wiring

Wifi Module. The order of the corresponding pins are shown as following VCC: 5v, GND: GND, VBEN: Digital 5, IRQ: Digital 3 [12].

Temperature Sensor. The TMP006 communicates over I2C, it is needed 4 wires to connect it to the Arduino. The order of the pins corresponds as shown as following VCC: 3.3v or 5v, GND: GND, SDA: SDA, SCL: SCL [13].

Pulse Sensor. The connection of the three different wires, colored as red, black and purple follows the following order, respectively: RED: 5v, BLACK: GND, PURPLE: Analog pin of preference, in this case Analog pin 0. The connection of the different devices can be appreciated in Fig. 2.

5 Data Streaming

Plotly streaming library for CC3000 Wifi module was used to send the acquired data to Plotly Servers. The streaming library uses an Application Programming Interface (API) combined with a REST API to stream data arrays to a plot into your personal account by linking tokens to data objects.

The code use for the prototype must initialize the user name, api key and tokens instances, subsequently the connection to the Plotly servers is initiated to confirm the user data and start the stream heading to http://stream.plot.ly address, thus, the transmitted data matches with the data object that corresponds to each one of the desired fields to stream and plot.

The main commands used in the prototype code are graph.init(), that enables the connection to Plotly Serves, and graph.openStream() that sets the way to stream the data to the selected plot of the account. On the streaming plotly library several aspects can be set depending on the users requirements including the maximum points displayed on each plot, number of axis, private or public visibility and the possibility of overwrite on every new stream. In Fig. 3 it is shown the plotted data of transmitted signals of BPM, heart signal and temperature data, respectively.

5.1 Alert Management

Alert messages in case of critical heart levels, i.e., values over the 120 beats per minute or under the 60 beats per minute, were managed using a notifications managing server Pushingbox.

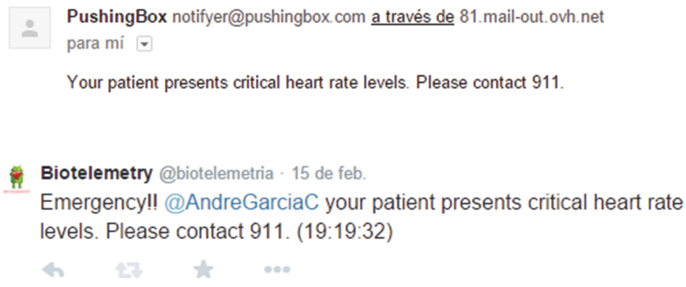

Fig. 4. Email and Twitter emergency message.

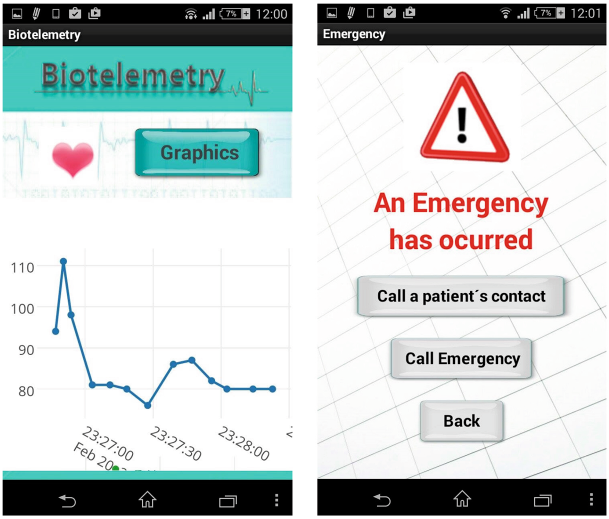

Fig. 5. Front screen (BPM streaming) and emergency screen of the APK.

Pushingbox it's an online service that allows to generate notifications through API calls. It is based on the creation of scenarios that are launched every time a Device ID calling is produced. Each scenario may have services as email send, twitter messages, MAC OS, iOS or Android OS notifications.

Each time a critical level is detected, the data stream is stopped and opens a connection to the Pushingbox servers through the Device ID linked to a scenario. The scenario starts a series of actions that include email send to the doctor's mail address, see Fig. 4, and a Twitter message, as shown in Fig. 5, indicating that the patient presents and emergency and that his/her assistance is needed.

6 Android APK

In order to create an Android OS application to display the stream data and manage alert messages in case of critical levels the MIT App Inventor 2 was used. App Inventor is based in a block diagram programming that allows to manage different cellphone aspects such as notifications, web access, calls and messaging.

The Biotelemetry App comprises a frontal page, indicated in Fig. 5, that displays in real time the graphic evolution concerning to the beat per minute values of the patient. On a secondary page, the BPM data can be visualized more detailed along with the pulse beat signal stream of the patient.

The application also contemplates the management of emergency notifications, this is possible due to the Twitter Developer App linked to the Biotelemetry App. When an emergency advise is published on a Twitter wall, the Biotelemetry app enables its own emergency management, this process includes the execution of a “talking message” and the display of emergency options, see Fig. 5, such as Call a Patient’s relative, Contact the Emergency number 911 or ignore the message in case of false alarm.

7 Results and Discussion

This prototype has centered its efforts on developing an Internet of Things based technology and focused on the area of biotelemetry systems, it has been proposed a new type of transmission of biosignals as a first step to generate an online data base for patient’s behavior and evolution. In order to test the system, extensive trials were conducted on three subjects in various conditions, their vital signs transmission was successful in each case and was notorious the effectiveness of the alarm system when simulated critical values in both heart rate variables and temperature.

Similar previous studies do not use free software on their systems, this characteristic has hindered the expansion of these researches in any field and restring the visualization features of the data from any platform. The proposed system may improve its performance and effectiveness in the future by using wearable sensors that will provide continuous sampling and transmission throughout the day as well as more accuracy in the results and graphics.

8 Conclusions

A Biotelemetry system consists on a necessary tool for any patient on treatment or under critical conditions whose biosignals must be monitored continuously. This prototype has been implemented for both patients and physician, constituting a system where the data is acquired by sensing the patient’s signals and subsequently monitoring by a doctor anytime through access to any device connected to the web and the Biotelemetry Android APK.

The objectives of developing a free software based system and easy to use prototype were fully achieved. Recent developments on Internet of Things technologies allow to use on a optimal way the existing web platforms, open to free data stream and storage. This prototype approaches to the extended use of Social Networks in our media and repurposes them in order to extend their original functions into new fields of health information and real time alerts, nevertheless, it constitutes a new study sector that can take advantage of recent studies and developers work to improve the versatility of biosignal transmissions, increasing the overall health of the patient.

The implemented code in this prototype can be downloaded from the following link: https://onedrive.live.com/redir?resid=61BE7C7AD6CBB9C4%21156.

References

1. Güler, N.F., Übeyli, E.D.: Theory and applications of biotelemetry. J. Med. Syst. **26**, 159–178 (2002)
2. Karagozoglu, B.: Ambulatory monitoring, pp. 244–249 (2013)
3. Paradiso, R.: Wearable health care system for vital signs monitoring, pp. 283–286 (2003)
4. Hanady, A., Abdulkareem, A.: Wireless sensor network for medical applications. Iraqi J. Electr. Electron. Eng. **11**(1), 49–59 (2015)
5. Poblet, J.M.: Introducción a la Bioingeniería. Marcombo, Barcelona (1988)
6. Kirk, H.S.: Photoplethysmography: beyond the calculation of arterial oxygen saturation and heart rate. Anesth. Analg. **105**, S31–S36 (2007)
7. Pulse Sensor Datasheet: Technical Code Walk-through. PulseSensor (2012). http://pulsesensor.com/
8. Van Herwaarden, A.W., Van Duyn, D.C., Van Oudheusden, B.W., Sarro, P.M.: Integrated thermopile sensors. Sens. Actuators A: Phys. **22**, 621–630 (1990)
9. Optotherm Inc.: Emissivity in the Infrared. Optotherm Thermal Imaging (2014)
10. Kopetz, H.: Internet of things. In: Real-Time Systems: Design Principles for Distributed Embedded Applications, pp. 307–323. Springer Science & Business Media, Berlin (2011)
11. Jakl, M.: Representational State Transfer. Citeseer (2005)
12. Lesniak, R.: Adafruit CC3000 WiFi. Adafruit Industries, New York (2014)
13. Bill, E.: TMP006 Infrared Sensor Breakout. Adafruit Industries, New York (2013)