

# Study of a Low-cost Heart rate Remote monitoring system using the Arduino platform

Carlos Neves Faustina  
College of Computer Science  
and Electronics Engineering  
Hunan University  
Changsha, China  
Email: cneves@uta.cv

Kun Xie  
College of Computer Science  
and Electronics Engineering  
Hunan University  
Changsha, China  
Email: xiekun@hnu.edu.cn

**Abstract**—Currently, remote monitoring systems have evolved to respond to the peculiar needs in the healthcare industry, such as Cardiovascular diseases, which is an essential foundation in the modern concept of the smart city. We propose a system to monitor patient current health conditions as a healthcare system based on the widely spread available low-cost technologies, namely, GSM and Arduino platform. Statistics show that hypertensive heart disease and blood pressure are risk factors for the high death rate. To decrease it, preventive measures should be applied providing a health monitoring system to save the patient life at an acceptable time. The objective of this paper is to develop a prototype of a low-cost system, that is capable of performing remote monitoring of a patient's heart rate and body temperature in real-time and stores the data on the memory card using low-cost hardware such as Arduino, and also, using the services provided by the GSM network, to enable communication with a health specialists cell phone upon request or if there are anomalies in the values measured by the sensors. The study focuses on heartbeat rate using the Difference Operation Method (DOM), described by Yeh & Wang, and body temperature, thus, in case of emergency, a short call is done, and then an SMS is sent to the health specialist's mobile containing measured values. Using this algorithm and after some tests using data from the MIT BIH Database distribution, we manage to find the best sample frequency and eliminate the distortion of the signal.

**Index Terms**—Remote Monitoring (ECG); Arduino; GSM Technology; Algorithm Difference Operation Method; Sensor.

## I. INTRODUCTION

Since Cardiovascular diseases (CVD) are the leading causes of death worldwide according to the World Health Organization, it is estimated that 17.9 million people died from cardiovascular disease in 2016, accounting for 31% of all deaths worldwide and of these deaths, more than 75% of CVD deaths occur in low-income and middle-income countries [4], the paper is developed to be a proposal for a low-cost remote monitoring system, based on open source technology, so it is a must to use old technology such as GSM together with Arduino platform. Thus, with this alarming statistic in mind, it would be convenient if there is some simple and low-cost process which would send health parameters regarding the health condition of patients to their health's specialist who is staying away without any manual intervention.

Currently, there are several health monitoring systems [2], [3] and [7] - [9] in the market which are able to send

signal/alarms when a certain emergency occurs. The main problem with these existing systems is that they are quite costly. Despite having good hardware with good filters that we could get, we couldn't use them as an option for this present paper because that inevitably made these systems quite costly. In the recent works many novel architectures, algorithms [10]-[12] and software systems [13] are proposed to implement the health monitoring system. We study in this paper, our own architecture, algorithms for alerting the health specialist and we create our own management software. Finally, none of these systems takes in consideration to make a short alarm call to the health's specialist before sending the message, since, just sending the message, sometimes, cannot be well perceived in useful time.

We propose a prototype that consists of two modules: One module that is coupled to the patient's body for data acquisition, validation and storage. This module consists of a GSM / GPRS shield module, a memory card reader module, a real-time Clock (RTC) and the heart rate and temperature sensors. This module is controlled by the Arduino's micro-controller. One Management Module that is an application responsible for the management of patient information and the management of resources present in the system, such as, register users, authenticate users with their due access credentials, change and remove information from a patient, load data histories measured by the sensors from the memory card to the database, view the graphs with the historical data of a patient, perform backup/restore of the application database, install the default settings on patient equipment and edit system variables.

First is measured the heartbeat and body temperature of a patient and stores the data on a memory card, after the collection of the EKG (electrocardiogram) data, a filter is used to remove the noise of the signals. Then is applied the algorithm Difference Operation Method (DOM), described by Yeh & Wang [1], which detects and counts the complexes of QRS and returns it to the heart rate frequency. The collected data are validated according to medical specifications placed on the memory card. If the data do not match these specifications (threshold values), the system makes a short-duration phone call to the health specialist's phone and then sends a text

message (Short Message Service) with the measured values. The health specialist can also check these data remotely by sending a text message to the system, containing a code, requesting the current data. In the case of a malfunction, a LED (Light Emitting Diode) and buzzer produce an alert specific to the malfunction (failure of memory card to initialize and failure in the acquisition of data from the sensors). The prototype also uses three LED lights and a Buzzer for the signal the failures, a heart rate sensor AD8232, a temperature sensor Ds18B20, a SIM900 GSM Shield, an RTC Clock DS3231, a memory card reader, a push button and the Arduino Mega ADK.

In this paper, we propose a new manner of advertising the health professionals in care of the patient, with two small algorithms, that can be seen in the algorithms 1 and 2. Our main contribution is when an anomaly is detected during the processing and calculation of the beating rate, an audible alarm is triggered and then SMS is sent to health professionals, after a short call that will help to improve the awareness, so the patient can be contacted in a short time. In comparison with those similar system studied for the development of this paper, none of them use the same procedure to advertise the health professional. The contributions of this paper can be summarized as follows:

- The system stores the measured values for later diagnostics;
- The system uses call services to alert in case of anomalies in the measured values, since just sending of a text message (SMS) is not always perceived by the health specialist;
- Measurement of the patients body temperature is also implemented;
- It does offer the physician the possibility to consult the current values measured by sending an SMS with a code to the patients equipment.

Since we are using cheap hardware with low processing power and memory, delays is introduced during the acquisition of the data, in order to obtain the desired sampling frequency that best suits the project.

This paper is organized into five chapters, where the main contents of each chapter are described below.

Chapter I give a brief description of the scope of the project, its general and specific objectives and also its organization. Chapter II describes the state of the art of some Systems that we can find nowadays in use. The procedures used for the development of the module are described in Chapter III, all the processes used for the implementation, assembly and operation of the electronic circuit are described also. In chapter IV an analysis is made of the experiences and results obtained. Finally, chapter V presents the conclusion and some future works or improvements that can be done in the proposed System.

## II. RELATED WORK

The "Holter" in [2], is a portable monitor System that records the heart's electrical activities over a period of 24 to 48

hours, so that it can detect anomalies that usually do not occur for a short period of time when the patient is being examined. The first *Holter* was developed in 1949 by Norman Jeff Holter, where the equipment weighed forty kilos and consists of the storage of the traces of electrocardiogram in a recorder and later reproduced. This device is widely used nowadays but it just can save the data in its internal memory for future analysis.

*Remote monitoring of the real-time heart rate System* in [3], proposed by Jasemian, Toft, & Arendt-Nielsen in 2004 that is divided into three sectors, that is, an ECG telemetry transmitter that will be coupled to the chest of the patient, which in turn is connected to a Sony Ericsson mobile phone via serial cable at a rate of 9600 baud. Subsequently, the data is automatically sent to the TMS server in the hospital through the GSM network. The data is collected at a frequency of 100Hz and sent in 34-byte packets every 80ms.

Mikhail St-Denis, designed *Life line project* in [8], that can monitor heart rate, blood sugar levels, humans body temperature, and by using a wireless communication technologies to synchronize and display these information into a smart mobile phone or a computer. Such device gather data from user equipment and display some related graphs in order to encourage users to remain aware of their health conditions by providing a week to week feedback.

*Heartbeat Monitoring and Alert System using GSM Technology* in [7], uses a pulse sensor to acquire and send signals to the control unit. In the control unit, the signal is processed and the heart rate is calculated. This beat rate value is then displayed on the LCD (Liquid Crystal Display). If an anomaly is detected during the processing and calculation of the beating rate, an audible alarm is triggered and an SMS is sent to health professionals. In comparison with the current project developed ( see the block diagram in figure 3), the system of remote monitoring of the patient's heart rate, the uses of an audible alert mechanism and the sending of SMS in the event of an anomaly are similar, but it is different in the following aspects:

- it does not store the measured values for later diagnostics;
- it does not use call services to alert the health professional;
- the patient's body temperature is not implemented;
- it does not offer the physician the possibility to consult the current values by sending an SMS to the patient devise module.

## III. ARCHITECTURE AND IMPLEMENTATIONS

This chapter will describe the procedures used to implement the system and the operations it performs. The procedures used to implement the module, the adjustments that have been made for the system to work in real-time, the algorithm that detects the QRS complexes from the patient's ECG, the physical connections of all the electronic devices used and their functionality when coupled to the Arduino board will be also explained here. The Arduino board is responsible for managing all iterations between them.

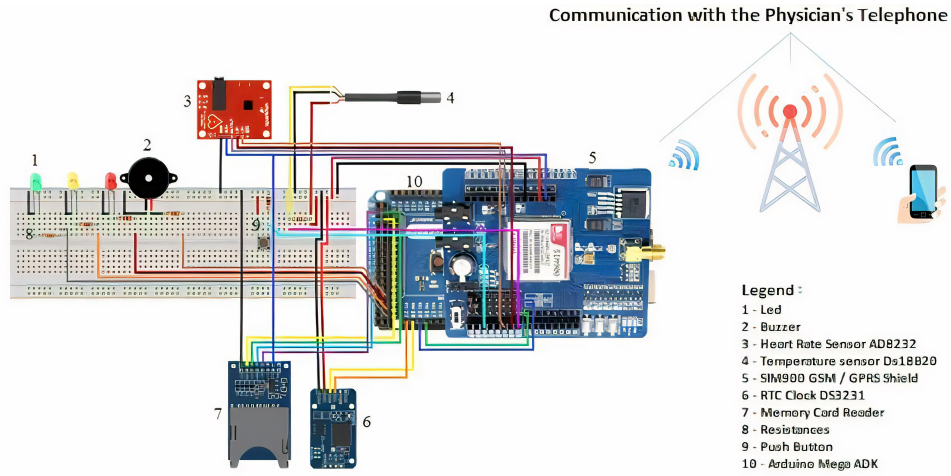


Fig. 1. Schematic Circuit of the body's Module.

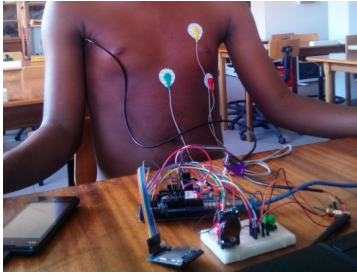


Fig. 2. Configuration of electrodes in the patient's body

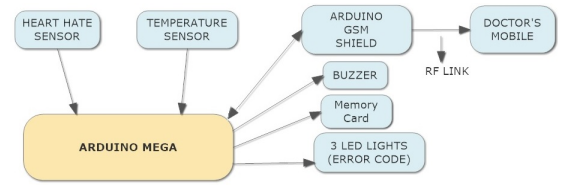


Fig. 3. System's Block Diagram

Mainly, the proposed system aims to cover an end-to-end low-cost health application that can be build up from two functional building blocks. However, the main function of the first building block is to gather all sensory data that are related to the monitored persons, whereas the second block functions are to store, process and present the resulted information of this stage to the doctors and nursery staff that are following the case of the monitored person.

#### A. Embedded Board (EB)

This subsection provides the hardware components details used to compose a smart board attached to the human body. Here also, shows the interconnection between the sensors with the GSM boards and the Arduino, and finally, the algorithms that make up the system alerts will be shown for a better understanding of the process.

As can be seen in the figure 1, serial communication between the Arduino board and the GSM module is performed using the Arduino's 18 (Tx) and 19 (Rx) ports and the 7 (Tx) and 8 (Rx) pins of the GSM module. The connections between the AD8232 heart rate sensor and the Arduino are performed as follows: the LO+ and LO- pins are connected to the Arduino's digital ports 5 and 6 to control the electrodes. If these pins are disconnected, it will be at the high level, and the low

level otherwise. The OUTPUT pin has been connected to the Arduino's analog A0 port for reading.

The three LEDs lights and the buzzer are connected to the digital ports 22, 24, 26 and 28 respectively. They are responsible for signalling failures that may occur in the system and 1k resistors are connected in series to reduce the voltage in the LEDs and buzzer. The signal pin of the DS18B20 temperature sensor is connected to the Arduino's digital port 7. A 4.7k pull-up resistances are used between VCC and the signal pin so that the sensor acquires state 1 or 0 when entering an indetermination range. The SDA and SCL pins of the RTC3231 have been connected to Arduino's digital ports 20 (SDA) and 21 (SCL). This module is responsible for controlling the time and date of the data collection of the sensors where they will be stored on the memory card. It is accompanied by its external power, which in case of power failure automatically triggers its battery to avoid losing records. The communication between the memory module and the Arduino board is performed using the MOSI, MISO, SCK and SS pins of the module and the ports 50 (MISO), 51 (MOSI), 52 (SCK) and 53 (SS) respectively. The memory card contains a text file (.txt) with the patient's personal information, a file with the medical specifications of the heart rate, body temperature and physician contact and also a file that will record all the data measured by the sensors, including the time (date and time) that they are acquired. The data in the text file are separated by a dot comma (;). Each minute is stored 28 bytes of information in its file in the following

format [temperature, beat, hour: min: ss; day-month-year].

The system starts all the components as follows: it verifies that the memory card is inserted correctly and initializes it, then initializes the GSM, then advances to the validation of the parameters inserted in the memory card. If the parameters are not within the predetermined specifications, the system uses the defaults setting placed in the Arduino EEPROM memory. If the number of the Doctor's cellphone inserted in the memory card is invalid (undergoes a change), the system uses the number recorded on the SIM card. If the memory card is not inserted or if it is not initialized correctly, the system does not initialize the GSM module, does not collect or analyze the data. The fault is signalled with each run cycle. The system waits for the patient to rectify the fault by inserting the memory card and restarting the system again. If the memory card initializes correctly, but an error occurs in accessing it, the system signals the failure and waits for the patient to restart the machine. In case of persistence of the error, the patient should turn off the system and contact the health' specialist, to make the necessary arrangements.

The ECG collection operation has the function of collecting the patient's ECG for a certain time and stored on the memory card. After the acquisition, a moving average filter (Smooth) is applied to the signal to remove noise from the signal. Then the (DOM) algorithm is applied to the signal to detect and count the number of patient's heartbeat during the time of collection. If an error occurs during this process, it is interrupted and the fault is signalled according to the code indicated. The electrodes are important elements for a good collection of the patient's ECG. They should be well affixed in places according to medical recommendations. For this, it is recommended to remove the chest hair and clean the electrode application sites. Inappropriate placement involves incorrect collection and analysis, so the patient should always make sure the electrodes are securely attached. The electrodes should be removed after the system is shut down. The arrangement of the electrodes used in this design should form a triangle with the heart in the interior. There are other provisions, but this is the one that best suits the project. Since the settings vary with the number of electrodes [1]. Figure 2 illustrates the configuration of the electrodes used in the design. For the ECG acquisition of the patient, a collection time of 15 seconds is used and the number of heartbeats per minute is calculated from the following formula:

$$NumbersOfBPM = 4 * NumbersHeartbeats_{15s} \quad (1)$$

The sampling frequency used for ECG acquisition is defined after several experiments, described below, to meet the proposed requirements (maximum allowable latency). Delays are introduced during the acquisition of the data in order to obtain the desired sampling frequency that best suits the project. Several delays have been tested in order to find the frequency range of sampling that contributes to the system being real-time and less oscillating. Applying the formula 2, we get the calculation of sampling frequency according to the number of samples and the time of acquisition:

$$Fa = \frac{N}{Ts} \quad (2)$$

where:

- $Fa$  - Frequency of sampling;
- $N$  - Number of samples read;
- $Ts$  - Total acquisition time.

The first experiments is done by performing the sensor readings on the Arduino and writing the values directly to the serial port. The MATLAB tool was used to collect serial port readings and write to a file. Through the readings performed and the total acquisition time, the sampling rate is calculated using the formula 2 above. From these tests, the sampling frequency of 192 Hz was defined as having the most stable values and adequate sample quantity for processing with Arduino. Delays are chosen based on several experiments performed during ECG collection. Initially, the objective was to maintain a sampling frequency equal to that found during the reading performed with the heart rate sensor and written to the serial port collected by MATLAB, but due to the instability in the sampling frequency, the use of delay is defined for reducing this instability. Table I below shows ECG acquisitions made during 15 seconds, with different delays.

Based on the tests performed, the delay of 5,195 ms is chosen, because it presented a more stable sampling frequency of ECG. Although the present delay shows a sampling frequency somewhat lower than that obtained in serial communication with MATLAB, it satisfies the requirements used to acquire and apply the algorithm to calculate the number of heartbeats obtained. The temperature collection operation has the function of obtaining the patient's body temperature and returns to the system at a temperature value in degrees Celsius. If the patient does not place the sensor in the location indicated by the physician or does not affix it properly, the sensor may return random temperature values. If the measured temperature or heart rate are very low than the default settings installed in the Arduino's EEPROM memory, the system emits a local alarm (on the patient equipment), indicating failure of the sensor data acquisition and the system considers that the electrodes or the temperature sensor are badly affixed.

The system analyzes the data and if it is out of medical parameters, checks if the GSM module is in standby, it initializes makes a short call and then sends an SMS with the values to the health specialist' phone. If an error occurs during the process, the system registers it, signals the occurrence and waits for the next iteration and tries again, in case of persistence of the error, the system will only signal with the LEDs corresponding to the fault (deactivating the beep) and disables the function that is returning the error, then places the GSM module in standby mode. If there is no error during the process after the system sends the SMS, it puts the GSM module on standby and waits approximately one minute to starts the data analysis process again. This process can be verified in the Algorithm 1. This operation allows the doctor to communicate with the patient's equipment by sending a text message (SMS) with its code. To check if there is unread SMS

in the box, the system launches the GSM module every five minutes. If it is on standby, the module is initialized.

If it finds an unread SMS, it starts the read process to obtain the sender's number and message.

TABLE I  
TESTS PERFORMED TO CHOOSE THE BEST DELAY WITH A MORE STABLE  
SAMPLING FREQUENCY

Test	Delays (ms)							
	5.18		5.19		5.195		5.2	
	N	F <sub>a</sub>	N	F <sub>a</sub>	N	F <sub>a</sub>	N	F <sub>a</sub>
100	2755	183.667	2783	185.533	2783	185.533	2783	185.533
101	2783	185.533	2783	185.533	2783	185.533	2783	185.533
102	2740	184.667	2769	184.600	2783	185.533	2783	185.533
103	2783	185.533	2783	185.533	2783	185.533	2755	183.667
104	2783	185.533	2783	185.533	2775	185.000	2778	185.200
105	2769	184.600	2768	184.533	2783	185.533	2783	185.533
106	2783	185.533	2754	183.600	2783	185.533	2783	185.533
107	2783	185.533	2783	185.533	2783	185.533	2774	184.933
108	2783	185.533	2783	185.533	2775	185.000	2772	184.800

---

**Algorithm 1:** Sending alert process

---

**Result:** Send message if no error during process  
temperature  $\leftarrow 0$ ;  
heartRate  $\leftarrow 0$ ;  
error  $\leftarrow false$ ;  
**while** error Not True **do**  
  **if** heartRate OR Temperature outside parameters **then**  
    make a call;  
    send message;  
    **if** there is an error during process **then**  
      flag the fault;  
      error  $\leftarrow True$ ;  
    **else**  
      wait about one minute;  
    **end**  
  **else**  
    wait about one minute;  
  **end**  
**end**

---

**Algorithm 2:** Receiving and sending SMS process

---

**Result:** Send SMS if no error during process  
sms  $\leftarrow false$ ;  
error  $\leftarrow false$ ;  
**while** error Not True **do**  
  **if** have received SMS **then**  
    analyze the number and sender's code;  
    **if** number OR code invalid **then**  
      delete SMS;  
    **else**  
      recover current values measured;  
      send SMS with the values retrieved;  
      delete SMS;  
    **end**  
    **if** there is an error during process **then**  
      flag the fault;  
      error  $\leftarrow True$ ;  
    **else**  
      wait about five minutes;  
    **end**  
  **else**  
    wait about five minutes;  
  **end**  
**end**

---

If the sender's number and the code obtained in the message are valid, the system collects the current temperature and heart rate values and send them to the health specialist' phone and

then eliminates the SMS. If the number or message with the code is invalid the system deletes the SMS and then places the GSM module in standby mode. If an error occurs during the process, the system registers it, signaling the occurrence and waits for the next iteration to try again. In case of persistence of the error, the system will only signal with the corresponding LEDs the fail and deactivate the function that is returning the error and then places the GSM module in standby mode. This process also can be verified for easy understanding in the Algorithm 2.

The prototype system features a switch to turn the module ON and OFF. When the system is OFF, only the RTC module is in operation to keep the time and date updated. The RTC is used in this paper to control the date and time of the storage of the values measured by the sensors on the memory card. This additional information helps in the detailed analysis in the management application. When it is turned ON, the system initializes the other modules and then deletes all temporary files created on the memory card and then activates all the features that were disabled due to errors. The system still has a reset button that is used to control the features disabled by the system due to errors. This button is associated with a system interrupt, that is, whenever it is pressed it triggers a short interruption that will activate the deactivated features and return to the point where the system stopped.

### B. Heart Spikes Detector Algorithm

This subsection will discuss in more details the algorithm that detects the QRS complexes.

Currently, there are several algorithms developed for the detection of QRS complexes in the ECG, this is due to the fact that there is not yet an algorithm that detects this complex with 100% effectiveness. The algorithm implemented for the detection of QRS complexes is based on the Difference Operation Method (DOM), described by Yeh & Wang [1], but with some changes to its better adaptation in this paper. The MATLAB tool is used to implement all the procedures used for the development, testing and validation of the algorithm. After successfully completing these procedures, it is implemented on the Arduino platform with some changes for its better operation. The DOM method basically consists of these distinct phases:

- Application of filters to remove ECG noise;
- Calculation of the first derivative of the signal;
- R spikes detection;
- Detection of the valleys Q and S from the R spikes found.

Filters applied by the method described above:

- Application of a filter for rejection of power line interference;
- Application of a morphological filter to remove the electrocardiogram;
- Adaptive filter application to remove interference from motion architeccts;
- Application of the high-pass filter with the cutoff frequency of 0.5 Hz to remove the baseline oscillations.

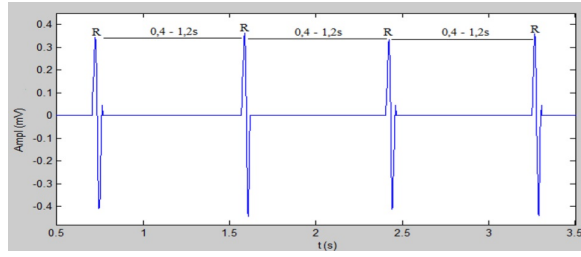


Fig. 4. Interval between two spikes R-R

The first derivative is calculated by applying this formula:

$$(Xd[n] = X[n] - X[n-1]) \quad (3)$$

where  $X[n]$  is the input signal at time  $n$  and the  $Xd[n]$  is the output signal. The resultant signal obtained is divide in two parts, one relative to the positive part of the signal ( $Xp$ ) and the other part corresponding to the negative part ( $Xn$ ). According to Yeh & Wang, [1], calculation of Thresholds  $T1$  and  $T2$ , indicate the mean values of all positive and negative waveform amplitudes.

$$T = 2average(Xp) - T2 = 2average(Xn) \quad (4)$$

The resulting signal is divided into windows of size 140, wherein each window the maximum spike is determined. According to the medical definition, the duration of each R-R interval (i.e., each Beat) is about [400ms to 1.2s]. Several ECGs are collected with the heart rate sensor used in the design, where a comparative table of the tests is constructed with the algorithm proposed by Yeh & Wang, and the modified algorithm. All ECGs are requested for 15 seconds and with a sampling frequency (192MHz). Table II shows the ECGs collected and the spikes detected by the implemented algorithms.

$$\hat{x}_{df} = \begin{cases} 0, T_2 < x_{df} < T_1 \\ x_{df}, x_{df} \leq T_2 \vee x_{df} > T_1 \end{cases} \quad (5)$$

For the validation of the extreme points in each sample window, the first *R-end* is selected and the medical recommendations, are applied to validate the other extremes to analyze if the distance between the newly selected point and the previous point satisfies the requirements, if not meet, the spike is eliminated.

Some adjustments are made in the application of the algorithm described by the method Yeh & Wang, and in the sequence, several tests are performed with data requested from the MIT-BIH Database Distribution, 2016, where it comes from with the ECG collection time of 10 seconds and the sampling frequency 360Hz. Several ECGs are collected with the heart rate sensor used in the design, where a comparative table of the tests is constructed with the algorithm proposed by Yeh & Wang, and the modified algorithm. All ECGs are requested for 15 seconds and with a sampling frequency (192MHz). Table II shows the ECGs collected and the spikes detected by the implemented algorithms.

TABLE II  
COMPARISON BETWEEN THE IMPLEMENTED ALGORITHMS IN RELATION TO THE EFFICIENCY IN THE DETECTION OF THE QRS COMPLEXES

ECG	Spikes correctly marked	Spikes detected by the algorithm without Smooth			Spikes detected by the algorithm with Smooth		
		Total	Insertion	Extinction	Total	Insertion	Extinction
Test 1	18	15	0	3	17	0	1
Test 2	18	18	0	0	18	0	0
Test 3	18	19	1	0	18	0	0
Test 4	18	19	1	0	18	0	0
Test 5	19	19	0	0	19	0	0
Test 6	20	20	0	0	20	0	0
Test 7	20	19	0	1	19	0	1
Test 8	20	19	1	2	20	1	1
Test 9	19	18	1	2	18	0	1

The spikes detected manually, obtained by manual spikes on ECG. The spikes detected by the algorithm without Smooth, obtained by following the phases described earlier Yeh & Wang, with the exception of applying some filters. The spikes detected by the algorithm with Smooth (see table II), is obtained by applying a mobile medium filter on the signal before following the phases described by the Yeh & Wang [1]. Mobile Medium filter (Smooth with span five) is applied to get the signal average. This filter is obtained by calculating the average of a set of values (defined by the length of the span), always adding a new value to the set and dropping the oldest, as will be described below.

$$y[n] = \begin{cases} x[0], n=0 \\ \frac{x[1]+x[2]+x[3]}{3}, n=1 \\ \frac{x[n-2]+x[n-1]+x[n]+x[n+1]+x[n+2]}{5}, 1 < N < N-1 \\ \frac{x[n-2]+x[n-1]}{3}, n=N-1 \\ x[N], n=N \end{cases} \quad (6)$$

where:

- $x$ -input signal;
- $y$ -output vector.

For the validation of the algorithms presented in table II, is calculated the average and median values for each of the data collected. Below will be described the calculations performed for test 1 and the others will be presented the results in table III.

1) *Spikes detected manually*: Applying the first-derived formula to obtain distances between the detected spikes:

$$d[n] = x[n] - x[n-1] \quad (7)$$

Distance Between Spikes = [169 169 165 173 173 169 169 165 161 150 146 151 157 158 169 173 158]

$$\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i \quad (8)$$

Applying the above equation 8 gets: Average = 163.2353. Before applying the median the distances between the spikes are arranged in ascending order.



$$\tilde{x} = \begin{cases} x \left[ \frac{n+1}{2} \right], & \text{if } n \text{ is odd} \\ \frac{\left[ \frac{n}{2} \right] + x \left[ \frac{n}{2} + 1 \right]}{2}, & \text{if } n \text{ is even} \end{cases} \quad (9)$$

Applying the above equation 9 gets: Median = 165.

2) *Spikes detected by the algorithm without Smooth* : Using the same equation as applied in the equation 7 we obtain: Distance between spikes = [169 169 165 515 163 171 161 150 146 151 157 158 169 173]. With these values the calculated average is 186 and the median is 164.

3) *Spikes detected by the algorithm with Smooth* : Applying the same formulas that are used in equation 8 we got: Distance between spikes = [169 169 165 173 173 169 169 165 161 150 146 151 157 158 169 173 158]

TABLE III  
RESULTS OF OTHER TESTS CONDUCTED

ECG	Duration					
	Manually		No Smooth		With Smooth	
	Average	Median	Average	Median	Average	Median
Test 2	153.1667	154.0	162.1765	154.0	162.1765	154.0
Test 3	164.0000	165.0	154.8889	165.0	164.0000	165.0
Test 4	162.7059	165.0	153.6667	159.5	162.7059	165.0
Test 5	155.5000	154.0	155.3889	153.0	155.5556	154.0
Test 6	144.0526	142.0	144.0526	142.0	144.0526	142.0
Test 7	147.2105	146.0	147.2778	147.5	147.2778	147.5
Test 8	143.5263	142.0	151.1667	139.5	143.4211	140.0
Test 9	153.5566	155.5	153.3529	158.0	153.3529	152.0

According to the Table II, validation of the algorithm is chosen based on the results of the spike detector algorithm with Smooth, because its result approaches more of the manually detected spikes.

#### IV. EXPERIMENTS AND RESULTS

This chapter describes all the experiments and the results achieved during the execution of this paper.

The interaction of the management application with the database was carried out successfully. From the application it was possible to connect to the database, insert, change and remove information in the tables from the interfaces created, load data from the memory card and store it in a table, view the graphs with the data histories measured by the sensors and Backup/Restore the database from the application interfaces.

The serial communication application - Arduino was successfully performed, having been able to write, query and change the default settings of the heart rate and temperature in the EEPROM memory of the Arduino micro-controller from an interface of the application and returned a confirmation code that the data have been inserted into memory successfully. All of the features described above are made available after the authentication, which ensured the security of the data regarding unauthorized access.

The download of the SD card data was successful, after downloading the data in the database, the *txt* file with the data measured by the sensors was deleted from the card. It also simulated a situation where for some reason the patient's personal information contained in the memory card does not match the data of the selected patient, the application does not perform data download.

The tests with the GSM/GPRS shield module was successful, having been able to carry out all the operations proposed in the project requirement. A call was successfully done for 20 seconds, then sent an SMS with the data measured by the sensors when anomalies occur or when the doctor requests a message with his personal code "dados". At the end of each of these operations, the module is placed in standby mode to reduce power consumption.

The data acquired by the temperature sensor are compared with the pharmacy digital thermometer and the results obtained were approximate.

ECG collection was also successfully performed after several tests performed with different collection times (15s, 30s and 60s), simulating the various situations of daily life and observing the behaviour of the electrodes and also the data that were collected in activities. The electrodes and their distribution in the body contribute a lot to a good ECG collection.

The QRS complex detector algorithm from the ECG was developed and implemented successfully using the MATLAB tool and manually validated, counting the QRS complexes on the ECG and compared with the QRS numbers detected by the algorithm. After the implementation and validation of the algorithm in MATLAB, the same algorithm is implemented in the Arduino, after several failed attempts due to memory problems, but through several improvements made in the algorithm code and change of the application method, these problems are overcome successfully.

The algorithm implemented in Arduino has an average duration of 33 seconds to filter the noise, apply the algorithm and return the number of heartbeats obtained from the collected ECG. The results obtained by the Arduino are compared with the results of the MATLAB tool (testing with the same ECG) and the results are the same as expected, however, the Arduino delay time is increased, due to its low processing capacity and sequential execution.

The system is simulated, with the prototype coupled to the body of a person, where we have obtained the positive results during a request by sending SMS, simulating the retrieval of the data remotely from the system. The system made a call to the health specialist's phone and then sent an SMS, because the data measured by the sensors were outside the parameters installed on the memory card of the patient's equipment. Several tests are also performed with a temperature sensor and beat values sensor and the system. A mobile number (simulating the doctor) is used to receive an SMS sent with the data. Due to the lack of heart rate measuring equipment, it was not possible to perform a test on the results obtained with the results of specialized metered equipment. Is also simulated the various situations in which system failures occur and all are flagged with successes.

The estimation of the autonomy of the system is made based on current consumption. Below will be described the procedures used to calculate the estimate of battery life in hours is equal to the capacity of the battery in mAh (milliampere-hour) divided by the consumption of the equipment in A

(Ampere). Using a rechargeable 12V Ion-Lithium battery with a capacity of 9800 mAh, several tests are carried out, measuring the current consumed by the module, using a digital multimeter connected to the micro-controller input for two minutes. Current variations from system start-up to the end of the cycle (initialization of the modules, collection, analysis and validation of the data and sending alerts to the mobile phone of the doctor). The average consumption obtained during the whole process is 84.3087 (mA). So the Average battery life will be:

$$\frac{9800(mAh)}{84,308(mA)} = 116.239(h)$$

This value is more or less five days of autonomy. We can say, that is not a good autonomy, so it is a must that in future papers it will be improved.

The use of the Arduino Mega ADK suppressed all the needs with respect to the module, several digital and analog ports are left free, that means, other sensors could be coupled and other functionalities can be also added to the module. As for flash memory (RAM), 41,262 bytes (16%) of 253,952 bytes are used and 2,994 bytes (36%) of 8,192 bytes are used for dynamic memory (SRAM).

## V. CONCLUSION AND FUTURE WORKS

With this study, it is concluded that the prototype presented, is able to perform remote monitoring of heart rate and patient body temperature in real-time and store the data on the memory card. The prototype worked as expected and also demonstrated that the Arduino platform, along with the technologies provided by the GSM network, presents a huge potential in the development of projects related to monitoring and remote control at a very low cost. It can be said thus, that the paper fulfilled all objectives initially proposed.

The application of the algorithm for the detection of QRS complexes based on Difference Operation Method (DOM), is proven to be useful, despite small changes made to it due to low memory and limited Arduino processing power.

Finally, it should be noted that the development of this paper is extremely important since it allows to a better understanding about advantages of monitoring the electrical activities of the heart and data acquisition remotely using low-cost equipment like Arduino.

As future works, some themes are suggested, like adding more sensors in the module, application of Artificial Intelligence, specially deep learning, to analyze the data and improve the algorithm of sending alarm to the health's specialist and also present the trends of the patient's health in a period of time, add an electronic circuit that indicates the level of the battery that is feeding the module, so when the battery is low the system signals the patient.

## VI. ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 62025201, 61972144, and Grant 61976087.

## REFERENCES

- [1] Yeh, Y.-C., and W.-J. Wang. QRS Complexes Detection for ECG Signal: The Difference. 91st ed., 17 Apr. 2008, computer methods and programs in biomedicine.
- [2] GRUPI, C. J., F. S. BRITO, and A. H. UCHIDA. Eletrocardiograma De Longa Duração: O Sistema Holter Parte Ie. June 1999, Brazilian Journal of Cardiac Arrhythmias.
- [3] Kakria, P., N. K. Tripathi, and P. Kitipawang. A Real-Time Health Monitoring System for Remote Cardiac Patients Using Smartphone and Wearable Sensors. 12 Nov. 2015, <https://www.hindawi.com/journals/ijta/2015/373474/>.
- [4] World Health Organization. "Cardiovascular diseases (CVDs)". Retrieved 22, January, 2021. <https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-cvds>.
- [5] Misodor.(2011).ELETROCARDIOGRAMA. 23. Abril. 2017, <http://www.misodor.com>.
- [6] Analog Devices. (2016, Agosto 30). AD8232: Single-Lead, Heart-Rate Monitor Front End Data. 29. Setemb. 2016, <http://www.analog.com/media/en/technical-documentation/data-sheets/AD8232.pdf>.
- [7] P. Rajashri , 2Prof. H.Balaji On Line Real Time Health Monitoring of ICU Patients using ARM 7 1 Electronics Department, DattaMeghe College Of Engg ,Airoli , Navi Mumbai , Maharashtra 400708, India 2 Department of Electronics, TernaEngg College, Nerul, NaviMumbai , Maharashtra , 400708 , India.<https://www.arduino.cc/en/Main/ArduinoBoard>.
- [8] N. Indumathy and K. K. Patil Medical Alert System for Remote Health Monitoring Using Sensors and Cloud Computing International Journal of Research in Engineering and Technology, vol. 3, no. 04, pp. 884-888, 2014.
- [9] M. Jagadishwara Rao and M. Kameswara Rao, An RTOS based Architecture for Patient Monitoring System with Sensor Networks, Indian Journal of Science and Technology, vol. 9(17), May 2016.
- [10] Christina Orphanidou, Timothy Bonnici, Peter Charlton, David Clifton, David Vallance, and Lionel Tarassenko, Signal-Quality Indices for the Electrocardiogram and Photoplethysmogram: Derivation and Applications to Wireless Monitoring, IEEE Journal of Biomedical and Health Informatics, Vol. 19, pp. 832-838, May 2015.
- [11] Jonathan Herzig, Amitai Bickel, Arie Eitan, and Nathan Intrator, Monitoring Cardiac Stress Using Features Extracted From S1 Heart Sounds, IEEE Transactions On Biomedical Engineering, vol. 62(4), pp. 1169-1178, April 2015.
- [12] Sebastian Thelen, Michael Czaplik, Philipp Meisen, Daniel Schilberg and Sabina Jeschke, Using off-the-Shelf Medical Devices for Biomedical Signal Monitoring in a Telemedicine System for Emergency Medical Services, IEEE Journal Of Biomedical And Health Informatic, vol. 19, pp. 117-123, January 2015.
- [13] Aureliano, A. (2017, Abril 4). Microcontroladores. Retrieved Junho 15, 2017, from fiozera: <https://fiozera.com.br/microcontroladores-914a59cbf7de>.
- [14] Bertoncini Simes, A. L., & Figueiredo De Martino, M. M. (2007, Abril 13). Variabilidade de circadiana da temperatura oral, timpnica e axilar em adultos hospitalizados. Retrieved Julho 3, 2016, from <http://www.scielo.br/pdf/reeusp/v41n3/20.pdf>.
- [15] Coutino Melco, T. (2006). Retrieved Novembro 04, 2016, from Estudo de Eletrocardiograma sob uma Abordagem Matemática: <http://www.teses.usp.br/teses/disponiveis/3/3152/tde-14122006113034/publico/TITOCOUTINHOMELCO>.
- [16] FilipeFloP. (2013). Mdulo de Carto SD Card. Retrieved from <http://www.filipefloP.com/pd-6b847-modulo-cartao-sd-card.html>.
- [17] G. R. Faulhaber, Design of service systems with priority reservation, in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 308.
- [18] Dr. Ana Luiza Lima, D.A. (2016, Abril 18). Retrived from <https://www.tuasaude.com/frequencia-cardiaca>.
- [19] Lin, E. (2008, Maro 4). Designing a Serial Peripheral Interface. Retrieved from <http://www.spansion.com/Support/Application>.
- [20] Arduino Mega ADK. (2017). Retrieved Fevereiro 23, 2017, from [www.arduino.cc](http://www.arduino.cc).