

Challenges for development of an ECG m-Health solution

Marjan Gusev¹, Aleksandar Stojmenski¹, and Ivan Chorbev¹

University Sts Cyril and Methodius, Faculty of computer science and engineering,
Rugjer Boshkovikj 16, PO BOX 393, 1000 Skopje, Macedonia
{marjan.gushev, aleksandar.stojmenski, ivan.chorbev}@finki.ukim.mk

Abstract. This paper presents the challenges to develop a system for early detection and alerting of the onset of a heart attack. The system consists of a wireless, easily wearable and mobile ECG biosensor, a cloud-based data centre, smartphone and web application. A significant part of the system is the 24h health monitoring and care provided by expert cardiac physicians. The system predicts potential heart attack and sends risk alerts to the medical experts for assessment. If a potential heart attack risk exists, an emergency ambulance is being called with the coordinates of the cardiac patient wearing the sensor. The timely reaction can prevent serious tissue damage or even death to the users of the system.

Our goal in this paper is to elaborate the challenges we met and solutions we have developed for development of an m-Health mobile application for detection of abnormalities in the ECG and alerting of a heart attack. The problems analyzed address Low Power Bluetooth connections, number conversion and transmission, decision making on what to be locally processed and what computations to be offloaded to cloud, software architecture, type of initial filtering for obtaining sufficient quality of the ECG signal, and visualization approach with relatively small processing requirements.

Keywords: electrocardiogram, sensors, cloud, Internet of things, medical application, heart monitoring

1. Introduction

Depolarization triggers each heartbeat as an electrical change of the sinoatrial node, located in the heart's right atrium. This signal is transmitted along the heart muscle and initiates a muscular contraction of the atria. Consequently, atrial chambers force blood into the ventricles and afterward to other parts of the body. The electrical changes can be recorded by an electro cardiogram (ECG) sensor over time.

If the captured ECG signals are not regular with determined shape and pattern then this indicates an irregularity, and can help the medical experts to establish a diagnosis of the heart malfunction.

Failures of the heart muscle do not appear suddenly and most of problems can be detected prior to serious health damages. For example, two hours before the heart attack takes place one can detect certain types of heart attacks [10]. Conventional ECG technology is based on scanning 12 leads by corresponding equipment in a special medical institution. Professional medical experts can establish a diagnosis by reading the results and analyzing the patterns in the patient's heart ECG.

New emerging technologies enable production of portable and wireless sensors and a medical institution will not be the only place where ECG tests are conducted [8]. These

new technology gadgets do not obstruct physical movements as they are patches on the human body. They allow remote online monitoring by transferring all scanned data to the nearby smart mobile device and then to the cloud.

mHealth gives new horizons for health using mobile technologies and a lot of initiatives have been taken to improve the healthcare worldwide [9]. Interventions categorized as medical and public health practice supported by mobile devices are called "mobile health" or "mHealth". The applications range from the use of mobile phones to improve point of service data collection, care delivery, and patient communication to the use of alternative wireless devices for real-time medication monitoring and adherence support [17].

In our earlier paper [15] we presented a proof-of-concept system for analysis of heart operation and heart attack detection using ECG sensor. In this paper, we elaborate the challenges we have met and solved in the development of the mHealth solution based on a mobile application that monitors the heart. The system uses wireless ECG sensors that are easy to wear on the human body, without any discomfort and can be worn even under a shower.

The organization of the paper is as follows. The background is presented in Section 2. Section 3 describes the system architecture. The main obstacles and challenges are discussed in Section 4 and related work in Section 5. Finally, the conclusions are given in Section 6.

2. Background work

Traditionally heart operation analysis is performed in the hospital or clinic where the patient is lying and sophisticated immobile medical equipment is being used for an ECG scan. This limits the ability for obtaining ECG data since the patient must be physically present in a medical institution.

Several improvements for this problem are provided. Some of them are data recorders such as Holter [10] or Cardionet [8]. These improvements use a lot of loose cables that are attached to the body in order to connect the device to multiple ECG electrodes.

In the research communities, IoT (Internet of Things) has been defined from various different perspectives and hence numerous definitions for IoT exist. The system that is presented in this paper is based on the IoT paradigm and presents a significant improvement of those problems since it uses only two electrodes for ECG scanning and utilizes cloud technology for real-time online analysis and diagnosis.

The main reason for the vagueness of the IoT term is the fact that it is composed of two distinct terms: Internet and things. Those two terms are distinct as the first one refers to the network-oriented vision of the IoT and the second one tends to focus on different objects in our surrounding that should operate together in a common framework [12]. Recent publications show that IoT is moving towards a cloud of things [5], which is why we decided to build a cloud-based system.

3. System architecture

This system consists of two parts: a mobile application and a web application in the cloud.

The mobile application aims to:

- communicate with the ECG sensor worn on a patient's body,
- monitor the patient's ECG scans,
- alert on abnormal heart function.

The cloud web application is used to:

- manage the information for all patients,
- analyze the received signals,
- alert on abnormal heart function,
- monitor the ECGs.

An overview of the system architecture and main functions is presented in Fig. 1.

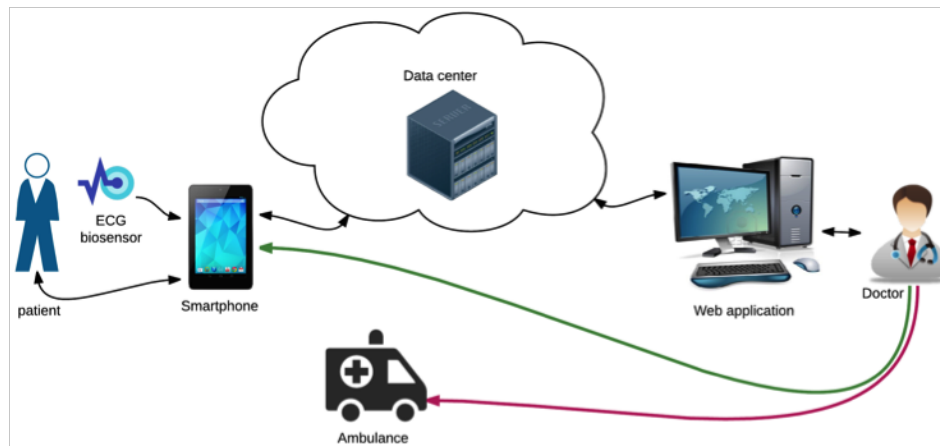


Fig. 1. Main functions and workflow of the ECGAlert system

The mobile application communicates with the ECG sensor via Bluetooth (eventually Ultrasound or other communication technology). It will also communicate the accelerator, GPS sensor or other data stored on the mobile phone. Further on, it communicates the cloud server via WiFi or 3G/4G mobile operator network. As an alternative, the mobile phone can communicate the local computer (laptop or desktop computer) connected by an appropriate cable.

Since this is a data-centric product it will need somewhere to store the data. Both the mobile application and cloud web application will communicate with the database, however in slightly different ways. All of the database communication will go over the Internet.

Data is stored in the mobile phone in a limited capacity dependent on the available phone's memory. Data files are also stored in the cloud server and can be subject for various analyses.

The patient can monitor and access the data on the phone received directly from the sensor. The caregiver is another user role that can monitor the patient's ECG and receive alerts for abnormal heart functions. This role can access the cloud and retrieve data files

representing the patient's ECGs. The doctor can also use a web application to retrieve patient's ECGs.

Whenever detected, the alerts are sent to the patient and the caregiver. When the doctor receives an alert, he/she is able to make a proper diagnosis and specify an EHR (Electronic Health Record), which can be overviewed by the patient or the caregiver.

The workflow starts with the cardiac patient that attaches the ECG sensor to the chest. The sensor sends information to a smartphone using low energy Bluetooth 4.0 technology and specific data transfer protocols. Afterwards, authorized users can login into the web application for status reports and history of the patients. Authorized users consist of medical personnel (this personnel can give a diagnosis based on an ECG and monitor alerts for their validity) and caregivers (these users could only view the ECG history and diagnosis given from the medical personnel for a particular patient), as presented in Fig. 1.

3.1. Sensors

The *Savvy* ECG biosensor is used in our proof of concept. It is fixed on two standard ECG electrodes that are waterproof and can be easily attached/detached to the body of the users. The sensor is small and lightweight with dimensions of 7x2cm and weight of only 3 grams. Fig. 2 compares the *Savvy* ECG biosensor with a two-euro coin.

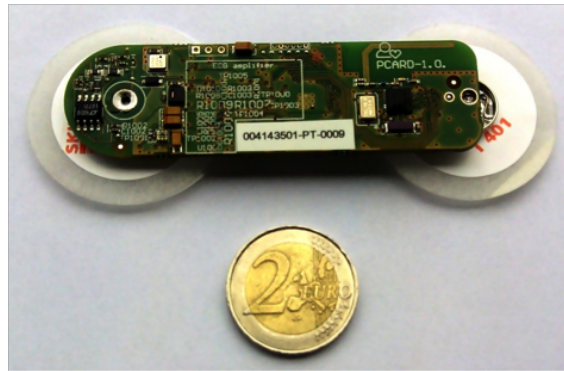


Fig. 2. Savvy Ecg BioSensor prototype

The main characteristics of the *Savvy* sensor are ECG sensing with a sampling frequencies of 126 Hz as predefined frequency and 500 Hz for specific diagnosis purposes. The AD conversion used is 10 bit and Bluetooth is used to communicate with the smartphone mobile device in range of at least 2-5 m.

3.2. Web application

The cloud-based web application helps users to detect the onset of a heart attack, using sophisticated, intelligent algorithms that analyze ECG data. The ECG data sent to the cloud servers is analyzed and stored on the server, where registered doctors have permission to

check it. If a potential risk of a heart attack is detected, the patient, doctor, and caregivers are alerted. Data is received, analyzed and processed prior to storing. The main database keeps information about all users and realizes authentication, authorization and alerting.

The main function of the web application is to enable heart condition monitoring by analyzing ECG data. This function is authorized to doctors, patients, and their caregivers. They can monitor and access the cloud and retrieve data files representing the patient's ECGs. If the doctor receives an alert for an abnormal heart operation, he/she is able to make a proper diagnosis and specify an EHR (Electronic Health Record), which can be overviewed by the patient or the caregiver on the web application.

A sample patient ECG scan on the web application interface is shown in Fig. 3, when a doctor is the authenticated user, according to [13]. The diagnosis of the current scan is provided on the current display of the web application, under the actual ECG visualization. The doctor can also view all the patients in this application, view the history of ECG scans and diagnosis for a specific patient using the electronic health record.

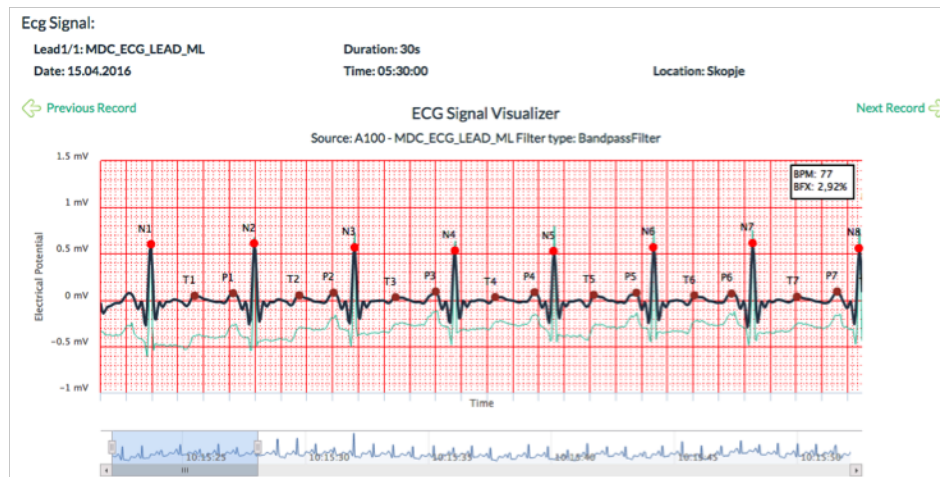


Fig. 3. Web application interface

3.3. Mobile application

The mobile application receives streaming data from the sensor, which is captured and saved as files 30 sec. Those files are stored locally on the device for history view and sent to the server for analysis. The Bluetooth connection to the sensor provides 126 samples per second, and each sample is 10 bits. A sample file of 5130 bytes is packed for a 30 sec ECG recording. If abnormal function detected, an alert file is saved for the detected problem. A listing of alert files is always updated on the mobile phone. Each alert file should be accompanied by an EHR (Electronic Health Record). A doctor forms an EHR after examination of the alert ECG file. Additionally, the doctor sets diagnosis and eventually modifies the therapy or recommends a more thorough examination.

In the case of a persistent Internet connection, each ECG file is sent to the cloud. However, the patient user can set an Internet saver mode. In the Internet saver mode, the system sends only files on a given period, for example, 1 file per 1, 5, 10 or 60 minutes. Alert files are always sent! If there is no Internet connection, then files are sent when an Internet connection is possible. In this case, the alert files are sent first, and then one file per hour is sent from the normal heart function files. Files can be kept in the mobile phone's memory up to 2 months, depending on available memory capacity. In the case of keeping the files for last 60 days, the required phone memory is 1.6GB.

Fig. 4 presents two screens of the mobile application. The first screen shows the current ECG measurement along with the information for the connected sensor and the connected cloud network for ECG analysis. The second screen shows the screen for examination of a particular ECG in the phone history. The ECG scan contains the average values of beats per minute (BPM) and heart rate variation (HRV), the diagnosis and anamnesis for that scan, the prescribed therapy if it exists and some additional notes that can be added from the medical personnel or the user.



Fig. 4. Mobile ECG application

The history part of the mobile application enables to view recorded ECG files with alert info and corresponding note. Filters for daily, monthly and yearly view are provided and the alert files are first on the list. The mobile application sends a compressed ECG

file to the file server (streaming unit) in order to keep the data transfer as low as possible. Each ECG file represents samples within a 30-second interval.

4. Identified obstacles and solutions

This section describes the challenges met while developing the mobile application. We elaborate Bluetooth connection problems, data conversion and transmission, offload or local processing dilemma, initial filtering and visualization issues in more details.

4.1. Bluetooth connection problems

The wearable ECG sensors need to save energy as much as possible to enable longer functioning and, therefore, most of the producers use low power communication technologies to communicate with the nearby smart mobile device, including Bluetooth or ultrasound.

Since the Savvy sensor is using Bluetooth Low Energy (BLE) version 4 protocol for transmission, connection using the standard Bluetooth libraries on Android and iOS were not applicable. The sensor is using a Generic Attribute Profile (GATT) which defines the way that two Bluetooth Low Energy devices transfer data back and forth using the services and characteristics concepts [1]. Services are collections of characteristics and relationships to other services that encapsulate the behavior of part of a device [18]. Services, characteristics, and related data are stored in a simple lookup table using 16-bit IDs for each entry in the table.

The GATT Server is a peripheral device represented by a sensor in our case. It holds the data stored in the lookup table of the Attribute Protocol (ATT). The GATT Client in our scenario is the smartphone and used to send requests to the GATT server (sensor). In our scenario, the GATT client (smartphone) is the master device that initiates the transactions, and the GATT master (sensor) sends responses back. The interaction between the sensor and the smartphone, described as GATT client and server is shown in Fig. 5.

In the beginning, when the GATT server establishes a connection, it suggests a certain "Connection Interval" to the GATT client (smartphone) as a central device. Afterward on each connection interval, the smartphone tries to reconnect. The GATT server responds if new data is available.

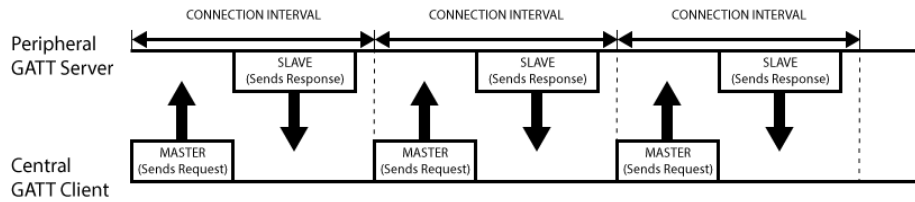


Fig. 5. The GATT client and server interactions

We had to design a new GATT server and GATT client architecture matching the custom peripheral codes from the sensor, following the Bluetooth LE standards.

On the iOS application, the CoreBluetooth framework was used with custom codes for each service and characteristic. On the Android side, the BLE API was used. The necessary permissions were appropriately set in the manifest and plist files respectively.

After the connection was established, we faced issues with the connectivity and discovery of the Bluetooth Savvy ECG sensor. There are certain limitations introduced to increase the stability, uninterrupted work, and privacy. The sensor is only able to connect with one smartphone at a time since the GATT connections are exclusive. One BLE peripheral device can only be connected to one central device, that can be sensor, smartphone, or similar device.

Interestingly, once a peripheral device connects to a certain central device, according to GATT it stops advertising itself to the outside world, and therefore, other devices will no longer be able to discover it. The device is not discoverable to other devices while the connection is active and it needs to be broken to be visible again.

Also, when an active connection of the sensor goes out of the Bluetooth range, the sensor does not automatically reconnect and is not discoverable for other devices for a certain period of time. Furthermore, we have programmed mechanisms for releasing the session when getting out of range and reconnection on the smartphone side.

Another issue emerges when multiple ECG sensors are in the Bluetooth discovery field. We tried to connect to one device in a room in which two other ECG devices were active and they all had the same identification name. The iOS system application for Bluetooth crashed each time a certain device was selected from the list. This problem can be resolved if the Bluetooth sensors identification names are accompanied by a UUID (Universally Unique Identifier).

4.2. Number conversion and transmission

After the Bluetooth connection problems were resolved, data was gathered from the sensor for further analysis and transferred to the cloud server. The server side was implemented in ASP.NET, while the first prototype of the application was running on iOS. The sensor transfers raw data 19 bytes to the mobile application. To each record, we added 8 more bytes for the date, time and location.

A problem appeared while transferring the raw byte data from the mobile application to the cloud server. Custom Media formatter was made on the ASP.NET side, in order to enable "application/octet-stream" MIME type transfer using the HTTP protocol. However, after receiving the input stream, in the attempt to convert the data into an array of integers, we have noticed that several bytes were missing in both Little Endian and Big Endian formats.

This problem was resolved with the usage of Base64String for data transfer. Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation. The data was transferred as encoded String from the iOS application and afterwards decoded on the server. Base64 encoding converts three octets into four encoded characters. Specifically, given an input of n bytes, the output will be $4\lceil\frac{n}{3}\rceil$ bytes long, including padding characters. This can be calculated with the equation (1).

$$\begin{cases} 3n \text{ bytes} & \xrightarrow{\text{base64}} & 4n \text{ characters} \\ n \text{ bytes} & \xrightarrow{\text{base64}} & \lceil\frac{4n}{3}\rceil \text{ characters} \end{cases} \quad (1)$$

This method increases the packages and transferred data approximately 33% more than the raw byte data transfer.

Although this increases the time and energy requirements of the smart mobile device, it was necessary to establish a proper communication without data loss.

4.3. Offload or local processing dilemma

In the phase of mobile application design, we wanted to find what is better for a modern mobile smart phone, to transfer data or to realize data processing. For this purpose, we have set a real experiment with an ECG sensor, which sends samples in a 30 sec interval. The experiments aim at analysing the energy consumption (battery life) and performance of the mobile device [14].

The conducted experiments show the influence of data transfer and data processing, or how many operations per data item require the same energy consumption as the transfer of a data element over a WiFi or 3G network. We made several experiments that used different algorithms for local processing. Each of the algorithms performs several operations per sample respectively and finds an average of 100 sequential elements. Then it sends the results to the server using WiFi or 4G Internet connection.

The input test data for the experiments consists of two data chunk types, DC1 that represents a signal with a sampling frequency of 500 Hz and precision of 16 bits, and DC2 stands for a signal representation using a sampling frequency of 126 Hz with 10-bit number data precision. The same algorithm that does not perform any operation per sample and just transfers all data to the server is represented by DC0.

Sample results are shown in Fig. 6. Only two smartphone devices are presented in the figure for clarity of presentation, the iPhone 6 (labeled with i6) and Samsung Galaxy Note 3 (labeled with S3). The *x*-axis presents the time measured in minutes of performed activities, and *y*-axis the measured energy consumption.

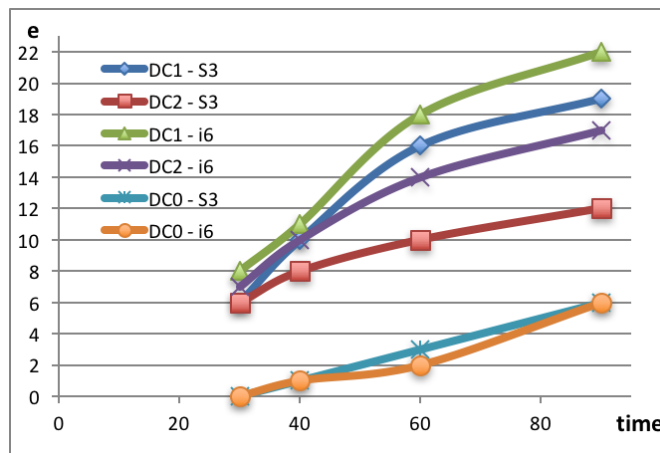


Fig. 6. Average energy consumption for processing one operation per sample

The experimental results showed that the algorithm which does not perform any operation on the input data stream and just sends data to the server consumes two or three times energy, and thus saves the smartphone battery. We have concluded [14] that even with increasing capabilities of the new smartphones, still, it is worthwhile to offload computations instead of using the mobile phone as a small processing unit.

4.4. Software architecture

As a consequence of the previous conclusions, we had to develop only those essential software modules to run on the smart mobile phone in order to save its battery as much as possible. The architecture of these modules is presented in Fig. 7.

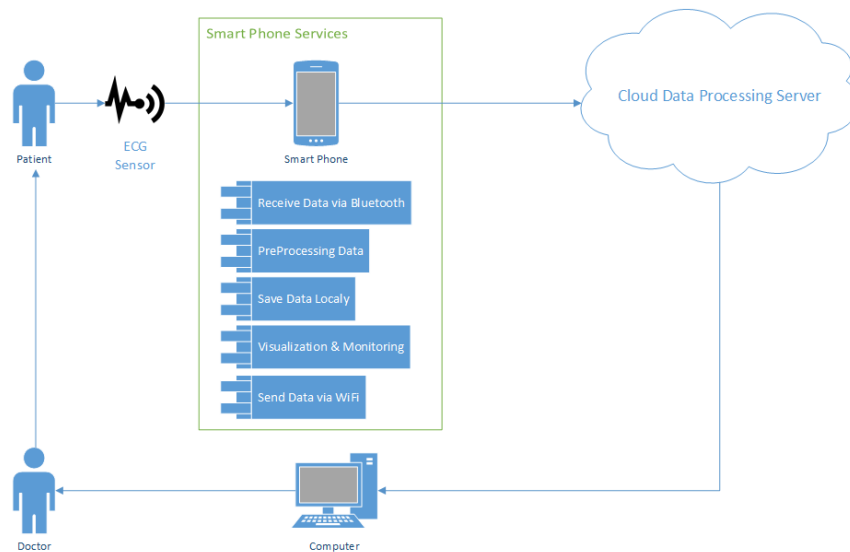


Fig. 7. Architecture of software modules for mobile application

There are software modules for

- receiving data via Bluetooth,
- preprocessing data for ECG signal analysis and prediction,
- visualisation and monitoring,
- storage and encoding,
- sending data to the server.

The module for receiving data is decoding the raw byte array received by the sensor via Bluetooth LE. The signal analysis and prediction module retrieves the data from the previous module and preprocesses the ECG signal for feature extraction using a modification of the QRS Complex algorithm. It calculates the heart beats and detects if there is an abnormal heart operation by analyzing the pattern scans. In the case of detected abnormality, it sends the code of abnormality and predicts the possible diagnosis. If there is an

onset of a heart attack, this module sends an alert to the patient and the cloud application with the appropriate data.

This is the essential algorithm that needs to be very efficient to save the smartphone's battery as much as possible. We have realized a program by a streaming algorithm, without using arrays and additional storage requirements. The visualisation and monitoring module present the data on the smartphone screen with a detected diagnosis for the current ECG signal by presenting the result from the preprocessing module. The output (sending) data module is used for data transfer of the current ECG signal data samples between the mobile device and the cloud application in regular 30 seconds intervals. If an alert is detected it is sent immediately.

4.5. Initial filtering

The ECG signals scanned by the ECG biosensor are influenced by a lot of noise from the surroundings and also from the human body. A typical example can be a 50Hz (or 60Hz) noise generated by a nearby charger or electrical device connected to the voltage power outlet. Human body movements generate additional noise, reflected mostly by human breathing, or physical movements.

Let's start with an analysis of ECG signal features. A normal heart beat is found in the range between 50 and 150 beats per minute, which corresponds to the frequencies of 0.8 and 2.5 Hz. To be able to extract the features one has to use higher frequencies, and the analysis shows that most of the power spectrum can go up to 30 Hz.

A low-pass filter is usually applied to weaken all signals with frequencies above the cutoff frequency, while passing all lower frequencies. Actually, it can eliminate all signal components produced by the electric power voltage produced by the environmental devices.

A high-pass filter is used to weaken the signal components lower than the cutoff frequency, while passing the higher frequencies. This is also known as baseline wander in the ECG signal, since the physical movements and respiratory rate are within the range of 12-20 breaths per minute which corresponds to 0.2 up to 0.35 Hz.

The effects of the respiratory rate are presented in Fig. 8 [11]. The ECG signal raises or lowers the baseline of the ECG signal, and the 50Hz noise produced by a local electricity source is presented as a high frequency signal. A combination of the low-pass filter with a frequency of 0.5Hz and a high-pass filter with a frequency 35Hz was used as a bandpass filter for the ECG signal analysis.

Several experiments were conducted to determine which programming solution will be the most energy efficient and saves the energy the most [11]. Finite Impulse Filter (FIR) and Infinite Impulse Filter (IIR) are computationally efficient, but they lack a precision when compared to Discrete Wavelet Transformation (DWT) or similar techniques. However, we are motivated to develop an efficient algorithm with small memory and processing requirements.

4.6. Visualisation issues

Since the application is available for multiple platforms, the visualisation takes part in real time on the Android and iOS applications and afterwards on the Web application. In

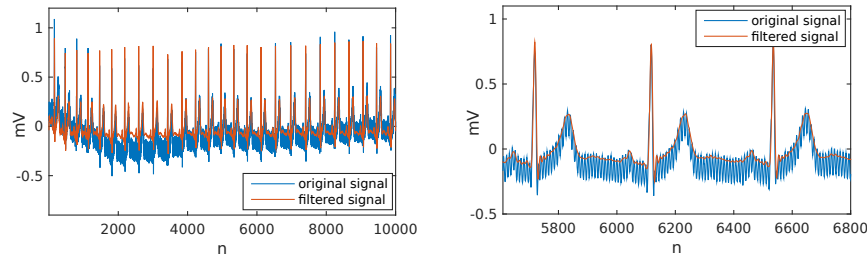


Fig. 8. Results by applying a bandpass filter to an ECG signal

order to keep consistency in the visualisation between all applications instead of native, we decided to use JavaScript visualisation.

The visualisation on the mobile applications is inside an embedded Web view, which uses the JavaScript library Highcharts. The visualisation was unsuccessful in the first attempts because the sensor was sending updates to the library 9 times each second and in the Highcharts library each new point that is added to the series is followed by a redraw method. To solve this problem we implemented a custom redrawing method which is called two times each second.

Another issue emerged when the visualisation was stuck after more than 10 minutes of ECG signal presenting. This was because the Highcharts library did not delete the points of the series after they were removed from the display and placed them in memory. This issue was resolved by a custom code that deletes every point immediately after it vanishes from the display.

Other minor problems appeared related to the changing height of the signal's plot which was dynamically set according to the minimum and maximum height of the R peaks which is individual for each person's ECG.

The final version of the visualisation shows not only the scanned ECG signal but also a rectangular grid for ECG signal and heart rhythm analysis with which the doctors are already familiar. On the other hand, the visualisation is made for users that are not experts at reading ECG, but can notice some main parameters, such as a repeatable signal. They could also read if the condition of the heartbeat is regular or some deviation has been detected.

5. Related work

A medical cloud is a cloud that offers healthcare-oriented services to the customers and mainly is working with EHRs. Tasic et al. [16] present a design of a cloud. For the purpose of this project, we have adapted this idea for the use of ECG signals.

Recently, few researchers work towards a wearable wireless, mobile, and remote ECG biosensor that is going to monitor heart rate and display. All of them differ to our work in few crucial aspects.

For instance, the sensor Shimmer3 ECG [7] provides a continuous recording of vital signs, but does not analyze the data and has no mechanisms for early warning.

Similarly, NeuroSky [3] provides a recording of vital signs but is more intended for conducting stress tests in hospitals.

QardioCore [2] is a multi-sensor platform that besides ECG, measures body temperature, the level of activity, and other vital signs, and thus presents a complete picture of the patient's health. Nonetheless, the data is only presented, and not processed at all.

Philips health applications [4] also introduced a sensor in the form of a replaceable patch where the battery lasts for up to 14 days and sends data to clinical software, where medical professionals set a diagnosis and therapy. While this is a good starting point, no warning in case of emergencies is offered.

One such comprehensive study [6] examined 120 different ECG biosensors from several aspects. The authors conclude that this area is a hot research topic and that innovative, applicative solutions may seize a unique market chance.

Worldwide, there is no published work that uses a 24h remote medical care based on remote sensing and early alerting. Usually, they function by telephone or by physical presence. The end-user benefit is the reduced mortality rate due to early alerting of potential heart attack and prolonged patient life. The medical experts have the possibility to react faster and be more successful in the treatment of the patients. Also, potentially large data knowledge databases will be generated from the system operation. This is a kind of research heaven for doctors in their process of setting a diagnosis or correlations between various parameters and user groups using similar scenarios.

6. Conclusion

This paper elaborates the challenges we faced in the development of a mobile application used in m-Health solution scanning and monitoring the ECG signals.

To deal with the constraints introduced by the realization of the Bluetooth LE GATT protocol we created special modules that control the GATT client (smartphone) and a new architecture for our purposes. Also, we built procedures to deal with the situation when the sensor is out of the Bluetooth range in order to reconnect and establish a proper connection.

The problem of data transmission to the data server and loss of data was solved using the base64string in the ASP.NET code. This introduced a 33% more data transfer but ensured no data loss.

Our measurements showed even increased data transfer is more efficient than processing the signals locally and sending only average values to the server. The overall solution as presented in the software architecture needs very efficient streaming algorithms.

The problem of noise elimination introduced by external environment sources and internal human sources are efficiently eliminated by use of a band-pass filter with cutoff frequencies between 0.5 and 35 Hz. The visualization and ECG monitoring were realized by adaptation of the Highcharts JavaScript library.

The prototype of the mobile application showed a performance close to the extended processing realized by sophisticated algorithms. We have realized that this is sufficient for the intended use of the smartphone as a data collector, preprocessing, transmitter and monitoring device.

References

1. Introduction to bluetooth low energy, a basic overview of key concepts for ble
2. Qardiocore: Ecg sensor (2016), <https://www.getqardio.com/qardiocore-wearable-ecg-ekg-monitor-iphone/>
3. Ecg biosensor (2016) (2016), <http://neurosky.com/>
4. Ackermans, P.A., Solosko, T.A., Spencer, E.C., Gehman, S.E., Nammi, K., Engel, J., Russell, J.K.A.: user-friendly integrated monitor-adhesive patch for long-term ambulatory electrocardiogram monitoring. *Journal of electrocardiology* 45(2), 148–153 (2012)
5. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. *Computer networks* 54(15), 2787–2805 (2010)
6. Baig, M.M., Gholamhosseini, H., Connolly, M.J.A.: comprehensive survey of wearable and wireless ecg monitoring systems for older adults. *Medical & biological engineering & computing* 51(5), 485–495 (2013)
7. Burns, A., Greene, B.R., McGrath, M.J., O’Shea, T.J., Kuris, B., Ayer, S.M., Stroiescu, F., Cionca, V.: Shimmer?—a wireless sensor platform for noninvasive biomedical research. *IEEE Sensors Journal* 10(9), 1527–1534 (2010)
8. Karlstädt, A., Fliegner, D., Kararigas, G., Ruderisch, H.S., Regitz-Zagrosek, V., Holzhütter, H.G.C.: a human metabolic network suited for the study of cardiomyocyte metabolism. *BMC systems biology* 6(1), 114 (2012)
9. Kay, M., Santos, J., Takane, M.m.: New horizons for health through mobile technologies. *World Health Organization* 64(7), 66–71 (2011)
10. Laguna, P., Thakor, N., Caminal, P., Jane, R., Yoon, H.R., de Luna, B., A., M., V., G., New, J.: algorithm for qt interval analysis in 24-hour holter ecg: performance and applications. *Medical and Biological Engineering and Computing* 28(1), 67–73 (1990)
11. Milchevski, A., Gusev, M.: Performance evaluation of FIR and IIR filtering of ECG signals. In: *ICT Innovations 2016, Advances in Intelligent Systems and Computing, AISC, Springer series* (2016), in press
12. Parwekar, P.: From internet of things towards cloud of things. In: *Proceedings of the IEEE 2nd International Conference on Computer and Communication Technology (ICCCCT)*. pp. 329–333 (2011)
13. Ristovski, A., Guseva, A., Gusev, M., Ristov, S.: Visualization in the ECG QRS detection algorithms. In: *MIPRO*. pp. 218–223. *Proceedings of 39th Int. Convention on ICT, IEEE Conference proceedings, Opatija, Croatia* (2016)
14. Stojmenski, A., Gusev, M., Chorbev, I.: Mobile data processing or computational offload. *Tech. Rep 21* (2016)
15. Stojmenski, A., Gusev, M., Chorbev, I.A.: mobile application for ECG detection and feature extraction. In: Pece, M. (ed.) *Proceedings of the Int.*, pp. 299–306. *Conf. on Applied Internet and Information Technologies, AIIT2016* (2016)
16. Tasic, J., Gusev, M., Ristov, S.A.m.c.: In: *Mipro*. In: *proceedings, I.C., Opatija, C. (eds.) Proceedings of 39th Int. Convention on ICT*. pp. 425–430 (2016)
17. Tomlinson, M., Rotheram-Borus, M.J., Swartz, L., Tsai, A.C.S.u.m.: where is the evidence? *PLoS Med* 10, 2 (2013)
18. Yu, B., Xu, L., Li, Y.: Bluetooth low energy (ble) based mobile electrocardiogram monitoring system. In: *Proceedings of the IEEE 2012 International Conference on Information and Automation (ICIA)*. pp. 763–767 (2012)