TIPTEKNO'19
TIP TEKNOLOJİLERİ KONGRESİ
MEDICAL TECHNOLOGIES CONGRESS
3-5 Ekim 2019
Kuşadası / Aydın

Design of Real Time Cardiac Arrhythmia Detection Device

Yasemin YOL, Mehmet Akif OZDEMIR and Aydin AKAN
Electrophysiological Signals Laboratory, Department of Biomedical Engineering
Izmir Katip Celebi University
Izmir, Turkey
yaseminyol67@gmail.com, {makif.ozdemir, aydin.akan}@ikc.edu.tr

***Abstract*—In this study, Raspberry Pi 3B+ based Electrocardiogram (ECG) device has been designed for real-time detection of cardiac arrhythmia. ECG signals that were taken by using AD8232 heart rate sensor have been displayed with developed software using Python in real-time. By using R-peak detection algorithm, we determined beats per minute (bpm) and arrhythmia type that is related with bpm. These results have been screened into the user interface that has been created with Python. The arrhythmia detection success rate of this study is determined as 97.9%.**

***Keywords*—AD8232; Arrhythmia Detection; Electrocardiogram; Raspberry Pi 3B+.**

I. INTRODUCTION

ECG is one of the most widely used signal in healthcare system. The ECG signal is studied for diagnostics even at the very early stage of a disease and it is recorded at the surface of the body via electrodes attached in various configurations. This signal describes the electrical activity of the heart over time, and pictures the sequential depolarization and repolarization of the different muscles [1].

ECG signal consists of some main parts such as P-wave, QRS- complex and T-wave, sometimes U-wave is also present. Each of these parts is the result of the specific electric activity of the heart. Schematic representation of normal ECG waveform is shown in Fig. 1. The P-wave is the result of atrial depolarization. The QRS complex occurs due to ventricular depolarization and ventricular repolarization causes in formation of the T-wave. Each of these waves, when considering normal heartbeat, has own characteristics amplitude and time period which leads to detection of abnormality based on the ECG waveform [2].

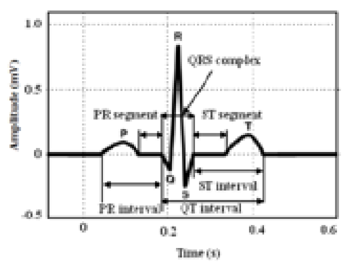

Fig. 1. Schematic representation of normal ECG waveform [3].

978-1-7281-2420-9/19/$31.00 ©2019 IEEE

The P wave depicts the depolarization of the right and left atria. This wave generally has a positive polarity for about 120 ms, while its spectral content is limited to 8-10 Hz, i.e., low frequencies. The QRS complex corresponds to the largest wave, since it represents the depolarization of the right and left ventricles. The duration of this complex is about 70-110 ms in a normal heartbeat. Finally, the T wave describes the ventricular repolarization. It has a relatively small amplitude and is usually observed approximately 300 ms after QRS complex [2].

An arrhythmia can be described as a heartbeat that is irregular or abnormal. The ECG can provide the most accurate cardiac arrhythmias information. Consequently, ECGs are widely used to monitor cardiac arrhythmias. A normal heart rate is 60–100 bpm. The heartbeat may be too fast (more than 100 bpm; tachycardia), too slow (less than 60 bpm bradycardia), too early (premature contraction), or too irregular (fibrillation). The arrhythmias are heart-rhythm problems and most of them are relatively harmless. However, some of them can be critical or even life threatening [4]. Therefore, the determination of arrhythmias is important.

Valiappan et al. [5] developed Raspberry Pi based portable real-time ECG device for arrhythmia detection by using Peak Detection Algorithm sensitive for detecting various types of arrhythmias. Raspberry Pi was the core device which received the ECG signal and transmitted the essential data along to the mobile application. The communication between Raspberry Pi and application has been set up by creating a Wi-Fi access point in Raspberry Pi. They have used AD8232 heart rate sensor and MCP3008 analog to digital converter.

Nicholar Clark et al. [6] developed wearable real-time ECG system for arrhythmia detection by using the Pan-Tompkins and the wavelet-based template-matching algorithms in order to identify premature ventricular contractions (PVCs). The low-cost TMS320C5515 and Raspberry Pi 3 have been chosen to be the hardware platform. When three or more consecutive PVCs have been detected, the device has alerted a patient's health care provider via wireless messaging.

The aim of this study is to design the ECG device capable of detecting basic arrhythmia in real-time such as, tachycardia and bradycardia in order to prevent sudden deaths.

II. MATERIALS AND METHODS

The device is based on Raspberry Pi 3B+. Raspberry Pi (RasPi) is a low cost, flexible, and programmable small computer board [7]. It has a 1 GB RAM which makes it ideal for faster calculations and real time implementation [5]. MCP3008 (analog to digital converter) has been used for communication between Raspberry Pi and AD8232. A simple serial interface compatible with the Serial Peripheral Interface (SPI) protocol is used to communicate with the devices [8]. AD8232 has been used for obtaining ECG signals via electrodes. The sensor can implement a two-pole high-pass filter for eliminating motion artifacts and the electrode half-cell potential [9]. Einthoven's triangle is the positioning of the electrodes for leads I, II, and III at the right arm (RA), left arm (LA), and left leg (LL) that creates the imaginary triangle [10]. Three-electrode system is very common system with telemetry monitors [11]. In this study, RA and LA are selected for active monitoring and RL is selected as ground electrode. The electrodes are connected to the tri-conductor electrode cable with the 3.5 mm jack output to connect to AD8232. Block diagram of the designed system is shown in Fig. 2.

Fluke Biomedical Prosim 8 Vital Signs Simulator is a full-featured, compact, portable simulator used to measure the performance of patient monitors. The simulator can simulate ECG functions, respiration, invasive and non-invasive blood pressure, temperature and cardiac output [12]. It has been used for generating abnormal or irregular ECG signal. The software design has been created with Python for plotting and digital processing of ECG signal in real-time. Python is easy-to-learn scripting language that can be run both in script or in interactive mode. It has become a common security tool platform as it is cross-platform, modular and comes with a large number of helper modules [13].

Fig. 2. Block diagram of the designed system.

A. Hardware Requirements

The appropriate General-Purpose Input/Output (GPIO) pins of RasPi were placed to the legs of MCP3008. Then, the legs of AD8232 were connected to MCP3008 and RasPi. The pin configuration is shown in Table I. Electronic circuit diagram of designed device is shown in Fig. 3. (a).

TABLE I. THE PIN CONFIGURATIONS OF DESIGNED DEVICE

RASPBERRY PI 3B+	MCP3008	AD8232
3.3 VDC POWER (Pin 1)	V_{DD} (Leg 16)	3.3 V (Leg 2)
3.3 VDC POWER (Pin 1)	V_{REF} (Leg 15)	-
GROUND (Pin 6)	AGND (Leg 14)	GND (Leg 1)
SCLK (SPI) - GPIO 14 (Pin 23)	CLK (Leg 13)	-
MISO (SPI) - GPIO 13 (Pin 21)	D_{OUT} (Leg 12)	-
MOSI (SPI) - GPIO 12 (Pin 19)	D_{IN} (Leg 11)	-
CE0 (SPI) - GPIO 10 (Pin 24)	CS/SHDN (Leg 10)	-
GROUND (Pin 6)	DGND (Leg 9)	-
-	CH0 (Leg 1)	Output (Leg 3)

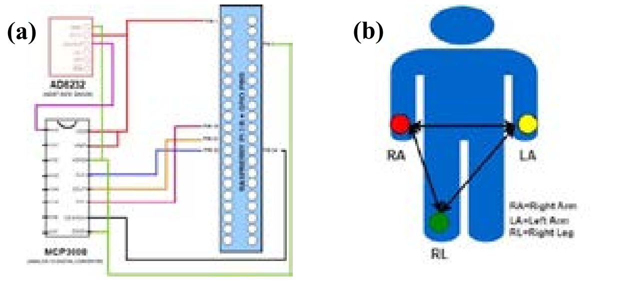

Fig. 3. (a) Electronic circuit diagram of designed device. (b) The positioning of three electrodes.

Three electrodes were placed on the patient body according to Einthoven's triangle. It is shown in Fig. 3. (b). The electrode cable was plugged to AD8232 via 3.5 mm jack output. Finally, Fluke Biomedical Prosim 8 Vital Signs Simulator has been used for 5 minutes of generating tachycardia (180 bpm), bradycardic (30 bpm) and normal sinus rhythm (70 bpm). The designed system with electrodes and simulator are shown in Figs. 4. (a) and 4. (b), respectively.

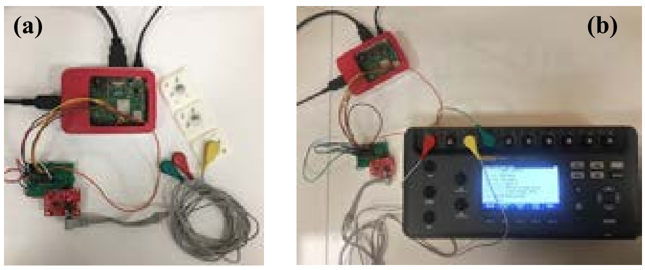

Fig. 4. (a) Main components of designed device. (b) Fluke Biomedical Prosim 8 Vital Signs Simulator.

B. Software Requirements

SPI channel has been initiated for taking ECG signal by using MCP3008 in real time. *Spidev* library has been used for creating a SPI channel. After that, ECG signals have been started to plot and updated in real time. The sample rate was about 170 Hz and this sample rate was enough to plot and process of ECG data which has 0-45 Hz. Block diagram of software requirement is shown in Fig. 5.

Fig. 5. Block diagram of software requirement.

After plotting the raw ECG signal, we decided to create a real time digital filter in order to remove noises from ECG signal. The power-line noises are about 50-60 Hz [14]. For the elimination of these noises, second order Butterworth 30 Hz low-pass filter has been occurred by using *SciPy* library and demonstrated in MATLAB in order to prove that noises can be removed from ECG signal. Finally, this filter has been applied to raw data in real time.

To calculate bpm, we had to determine R peaks at one minute because there is one R peak for one beat. R peaks determination algorithm has been applied to filtered ECG signal. The principle of algorithm has been based on finding maximum value by simple comparison of neighboring values. During finding R peaks, mean of R-R intervals has been calculated for one minute. Finally, we calculated bpm by using this equation [15];

*Beats per minute = 60 * Fs (sample rate) / R-R interval (mean)*

According to bpm result, the codes have been written for determination of arrhythmia type. If bpm was greater than 100, it was tachycardia, and if bpm was smaller than 60, it was bradycardia. It was normal sinus rhythm if bpm was between 60 and 100.

Finally, a user interface has been created for ECG plotting and showing bpm and arrhythmia result in order to make better visualization by using *matplotlib* library.

III. RESULTS AND DISCUSSION

Raw signal and the filtered signal data have been saved into file for 5 seconds and they demonstrated in MATLAB in order to prove that removing noises from ECG signal. The Butterworth second order 30-Hz low-pass filter has been created for filtering ECG signal from power-line noises. The power spectrums of raw and filtered data have been plotted in order to observe the distribution of frequency shown in Fig. 6. and Fig. 7 respectively. We have observed that the filter suppressed noises above 30 Hz. We have concluded that the designed filter was appropriate for removing noises from ECG signal.

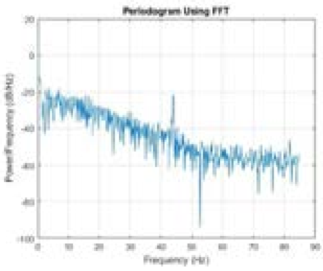

Fig. 6. The power spectrum of raw data in MATLAB.

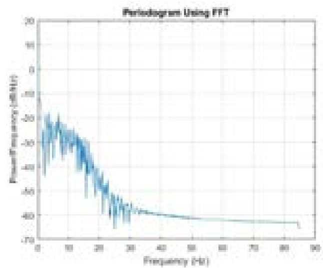

Fig. 7. The power spectrum of second degree 30 Hz Butterworth low-pass filtered data in MATLAB.

After MATLAB simulation, the procedure has been continued in Python as real-time. Before filtering, the raw data plotting is shown in Fig. 8. Subsequently, filtered data plotting is shown in Fig. 9. A filtered ECG signal has been obtained for detecting of R-peaks in order to calculate bpm and arrhythmia type. After determination of R-peaks, the procedure has been continued by marking the R peaks. R peaks marked data is shown in Fig. 10.

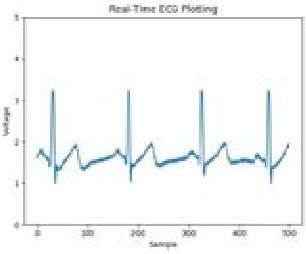

Fig. 8. The real-time plotting of raw data in Python.

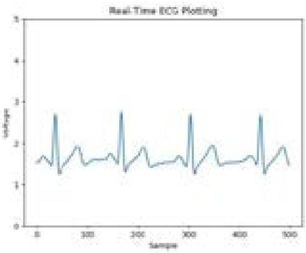

Fig. 9. The real-time plotting of filtered data in Python.

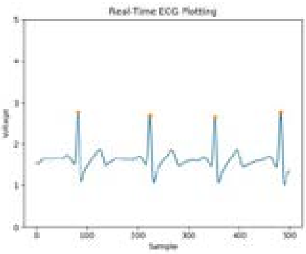

Fig. 10. The real-time plotting of both filtered and marked data in Python.

After software requirements for finding bpm and arrhythmia type, tachycardia (180 bpm), bradycardic (30 bpm), normal sinus rhythm (70 bpm) ECG signals have been generated and the results of these signals have been displayed for 5 minutes. They are shown in Fig. 11, Fig. 12 and Fig. 13 respectively.

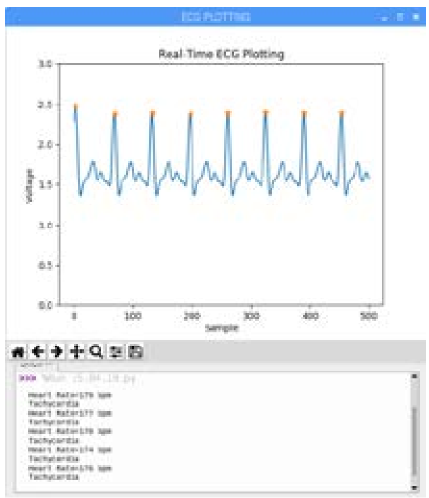

Fig. 11. The results of tachycardia ECG signal in 5 minutes.

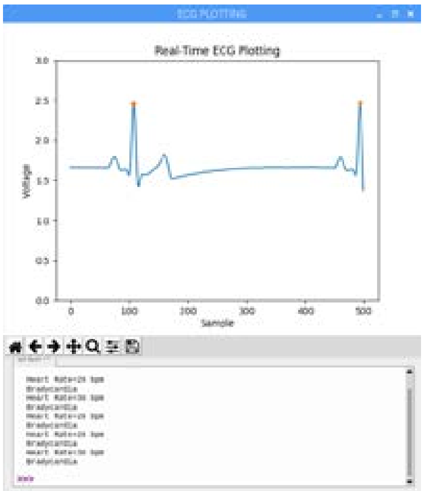

Fig. 12. The results of bradycardic ECG signal in 5 minutes.

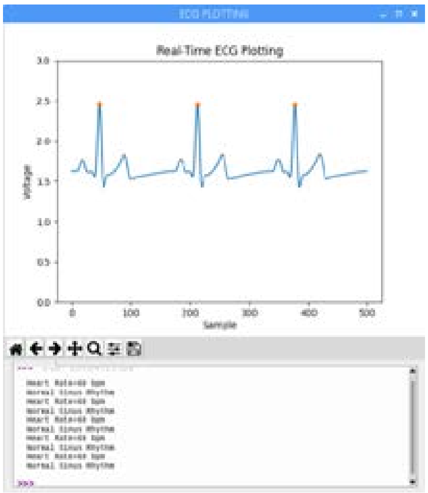

Fig. 13. The results of normal ECG signal in 5 minutes.

Table II shows the predicted and actual bpm results in the ECG signal predicted by developed software as compared to generated ECG signal from simulator. The accuracy of this study is determined as 97.9 %.

TABLE II. PREDICTED AND ACTUAL BPM RESULTS IN GENERATED ECG SIGNALS

BPM RESULTS	Tachycardia (180 bpm)	Bradycardic (30 bpm)	Normal Sinus Rhythm (70 bpm)
At minute 1	179	29	69
At minute 2	177	30	69
At minute 3	178	28	69
At minute 4	174	28	69
At minute 5	176	30	69
Success Rate (%)	**98.2**	**96.9**	**98.6**
Average Success Rate = 97.9 %			

IV. CONCLUSION

In this study, the real-time cardiac arrhythmia detection device is designed by using Raspberry Pi 3B+. It determines bpm and arrhythmia related to bpm like tachycardia, bradycardia with aiming of developed software that is created by using Python programming language. Real-time determination of arrhythmia type provides advantages for both physician and patient. The device will enable the detection of arrhythmia types that may endanger the patient's life, and will accelerate the appropriate treatment method. The device can be portable with small LCD screen and power bank.

In the future, we aim to detect other type of arrhythmia such as atrial fibrillation, ventricular fibrillation etc. in real time. An alarm system can be designed for identifying different type of arrhythmia in order to inform the hospital staff at hospitals. A mobile application can be developed for monitoring of ECG signals.

REFERENCES

[1] F. Agrafioti, "ECG in Biometric Recognition: Time Dependency and Application Challenges," Ph.D. dissertation, University of Toronto, 2011.

[2] K. V. Kurangkar, A. B. Nandgaonkar, and S. L. Nalbalwar, "ECG Analysis and Abnormality Detection," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2018, pp. 1761–1764.

[3] J. P. Sahoo, "Analysis of ECG signal for Detection of Cardiac Arrhythmias," M.S. Thesis, National Institute of Technology, 2011.

[4] R. C.-H. Chang, H.-L. Chen, C.-H. Lin, and K.-H. Lin, "Design of a Low-Complexity Real-Time Arrhythmia Detection System," *J. Signal Process. Syst.*, vol. 90, no. 1, pp. 145–156, Jan. 2018.

[5] C. A. Valliappan, A. Balaji, S. R. Thandayam, P. Dhingra, and V. Baths, "A Portable Real Time ECG Device for Arrhythmia Detection Using Raspberry Pi," in *Wireless Mobile Communication and Healthcare*, vol. 192, P. Perego, G. Andreoni, and G. Rizzo, Eds. Cham: Springer International Publishing, 2017, pp. 177–184.

[6] N. Clark, E. Sandor, C. Walden, I. S. Ahn, and Y. Lu, "A wearable ECG monitoring system for real-time arrhythmia detection," in *IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, Windsor, ON, Canada, 2018, pp. 787–790.

[7] P. Patil and K. Bhole, "Real time ECG on internet using Raspberry Pi," in *International Conference on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India, 2018, pp. 267–270.

[8] MCP3008 Datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf, 08.07.2019.

[9] AD8232 Datasheet: https://www.analog.com/media/en/technical-documentation/datasheets/ AD8232.pdf, 08.07.2019

[10] R. Ortega, M. Mazzini, K. Xue, and D. Espaillat, "Electrocardiographic Monitoring in Adults," *N. Engl. J. Med.*, vol. 372, no. 8, p. e11, Feb. 2015.

[11] J. Francis, "ECG monitoring leads and special leads," *Indian Pacing Electrophysiol. J.*, vol. 16, no. 3, pp. 92–95, May 2016.

[12] Fluke Biomedical ProSim ™ 8 Vital Signs Simulator Documentation: https://www.flukebiomedical.com/sites/default/files/resources/prosim8_umeng0300.pdf, 08.07.2019.

[13] M. Rocha and P. G. Ferreira, "An Introduction to the Python Language," in *Bioinformatics Algorithms*, Elsevier, 2018, pp. 5–58.

[14] Ö. Yakut, S. Solak, and E. D. Bolat, "EKG İşaretindeki Gürültülerin Temizlenmesi için IIR Tabanlı Sayısal Filtre Tasarımı," *J. Polytech.*, Jan. 2018.

[15] R. Mehra, "Global public health problem of sudden cardiac death," *J. Electrocardiol.*, vol. 40, no. 6, pp. S118–S122, Nov. 2007.