AN EFFICIENT PULSE DETECTION TECHNIQUE USING ANDROID APPLICATION

Arun Prasad R[1], Sharan Karthik A[2], Sathya Narayanan R[3], Ayyappan B[4]
[1, 2, 3] Department of Information Technology, Jeppiaar Engineering College, Chennai, Tamil Nadu, India
[4]Assistant Professor, Department of Information Technology, Jeppiaar Engineering College, Chennai, Tamil Nadu, India
[3]sathya.narayanan4@gmail.com

Abstract— People around the world have the problem of measuring their pulse rates (bpm). Initially people went to clinic and hospitals to measure them and later they moved to measuring aids and instruments. Pulse detection involves measuring the human heart rate. Pulse rate is counted by putting slight pressure on any artery in the body where pulsations can be felt. In order to ease this, android applications in mobile phones are developed to measure the heart rates. This application focuses on measuring the pulse using a pulse sensor which is interfaced with an Arduino board which makes the connection with the android application. Then the pulse sensor measures the pulse rate from the finger in bpm units and sends the inference to the application. Then the application displays the information and also tracks the nearby hospitals in that particular locality.

Keywords: **bpm, pulse rate, pulse sensor, GPS etc.**

I. INTRODUCTION

Heart rate or pulse rate is the rate at which heart beat per unit time, is commonly referred as beats per minute (bpm).A normal heart rate for adult ranges between 60-100bpm. A lower heart rate at rest indicates a better functionality of heart. For example a well-trained athlete has heart rate in the range of 40bpm. The measurement of heart rate is used my physicians to diagnoses the body condition of a person. The heart rate can be measured commonly by auscultation at palpation sites.The common palpation sites are carotid artery, brachial artery, radial artery, ulnar artery, femoral artery, popliteal artery, posterior tibial artery and dorsalis pedis.

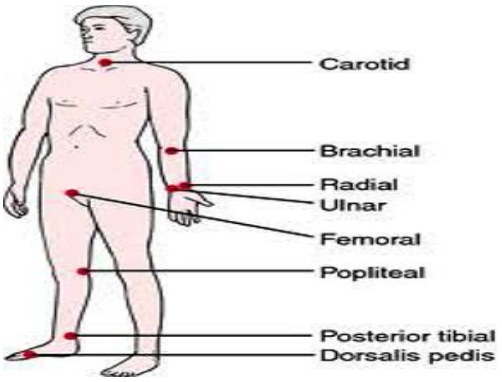


Fig. 1. Palpation Sites

People around the world go to clinic and hospitals to measure their heart rates.

With the advancement in technology this doesn't prove to be worthy. Researchers came with devices and systems for measuring them. These prior implementations are single-function devices and systems. They just measure and predict the pulse rates.

However in the recent past, application developers came with pulse detection applications [1] in mobile phones. The major drawback was that it provides just an approximation of the pulse rates and their plots weren't correct enough.

In this paper we define a non-invasive system which provides a constant plot of the pulse rates. Our application just measures the pulse rate by using a sensor which sends the pulse information to the mobile phone's android application via a bluetooth technology.

II. RELATED WORK

Much work has been done in just plotting the approximation of the pulse rates. The prior works made by Joel Murphy [2] and Yury Gitman [3] involved the same set up which consisted of the same pulse sensor and the Arduino board. They used the exactly the same algorithm as that of ours. The only difference between the prior works [4] and ours is the application interface and the additional features. The algorithm for the measurement of pulse involves:

Algorithm- Pulse Detection

```
1. Initialization- Initial Assignment
    1.a Power ()
2. Working- pulse sensor
    2.a Amarino ()
    2.b Bluetooth ()
    2.c UserInt ()
```

Fig. 2. Pulse Detection Algorithm

a) **Power()** module involves the powering up of the Arduino board and the pulse sensor attached to it. This is acknowledged by the glowing of LED in the Arduino board. This sets up the execution of the *Amarino ()* [5] module.
b) **Amarino()** module initiates the power sensor and gathers the data, i.e., the pulse rates from the sensor. This module in turn kick starts the *Bluetooth ()* [6] module.
c) **Bluetooth()** module just creates an interface between the mobile application and the Arduino board via a bluetooth device.
d) **UserInt()** [7] module is a java module which is responsible for the application user interface in which the pulse rates will be displayed.

The working of this application involves the powering up of the Arduino board which in turn will power up the pulse sensor interfaced with it. The powering up of the board initiates the Amarino () module which will fine tune the settings required for gathering the information from the pulse sensor. This Amarino () module in turn is responsible for initiating the Bluetooth () module. Then the application is switched on, provided that the bluetooth of the mobile phone is switched on during the entire process. Then the forefinger of either hand is placed on the pulse sensor for hardly 4-5 secs. The pulse sensor after reading the pulse sends them to the board using Amarino (), which in turn will send them to the mobile phone using the Bluetooth () module via a bluetooth device interfaced with the board. The pulse rates are then displayed on the mobile application developed using UserInt ().

III. SYSTEM MODEL

The system model comprises of an Arduino board [7] with the sensors (Pulse sensor and Accelerometer sensor [8]) interfaced to it. The other components involve a step down transformer, Bluetooth serial module [9], IC for reading the data from the board via an USB output. The step down transformer is responsible for stepping down the input voltage to 7-12 on which the board operates. The Bluetooth serial module is capable of communicating with all the Bluetooth-equipped devices, such as laptops, tablets, and PC’s or smart phones. The schematic representation of the system model follows:

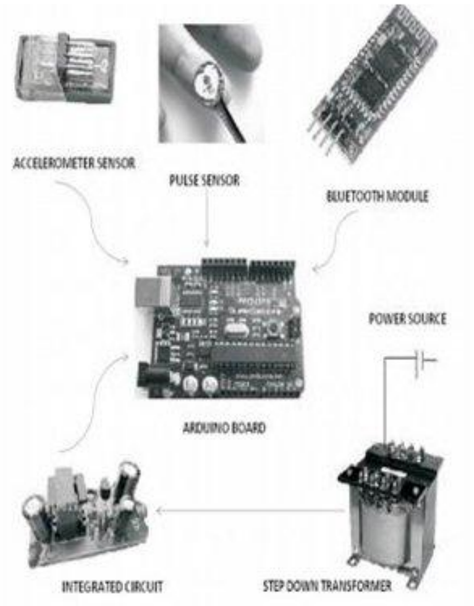


Fig. 3. Pulse Detection System Architecture

IV. SIMULATION RESULTS

Our application is simulated using the Google emulator provided by the Eclipse IDE. The simulation we do is a plot of the human pulse rates using a metre output. The prior implementation output is a digital one. The user interface which we introduce is a combination of both analog and digital output. The application also tracks the nearby hospitals of that particular locality and displays them in the map provided by the Google maps.

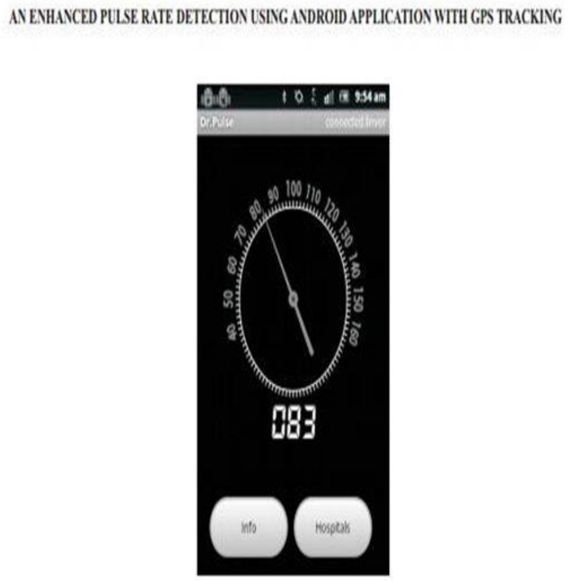

AN ENHANCED PULSE RATE DETECTION USING ANDROID APPLICATION WITH GPS TRACKING
9:54 am
Dr.Pulse
connected
40
50
60
70
80
90
100
110
120
130
140
150
160
083
Info
Hospitals

Fig. 4. Pulse Detection Android Application (Current Implementation)

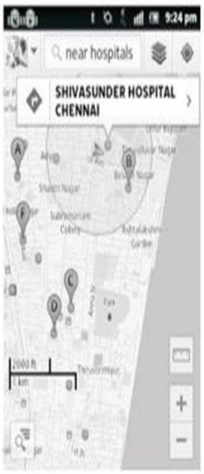
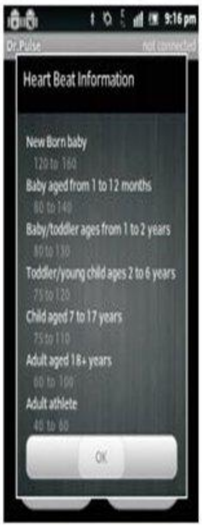


Fig. 5. GPS tracking or nearby hospitals

Our implementation of the android application also involves the voice based output (Compatible only on certain devices). Once the pulse rate is displayed via the analog and digital means, the voice based output will be activated. The application also has a general pop up for displaying the actual pulse rates for human beings based on age factor.

The prior works were just based on a digital display of the pulse rates. The implementations mainly consisted of a single activity which just pops ups when the application is opened. The prior implementation also lacks in user interface attraction.

AN ENHANCED PULSE RATE DETECTION USING ANDROID APPLICATION WITH GPS TRACKING



Heart Beat Information

New Born baby
120 to 160
Baby aged from 1 to 12 months
80 to 140
Baby/toddler ages from 1 to 2 years
80 to 130
Toddler/young child ages 2 to 6 years
75 to 120
Child aged 7 to 17 years
75 to 110
Adult aged 18+ years
60 to 100
Adult athlete
40 to 60

OK

Fig. 6. General pulse rate information (Current implementation)

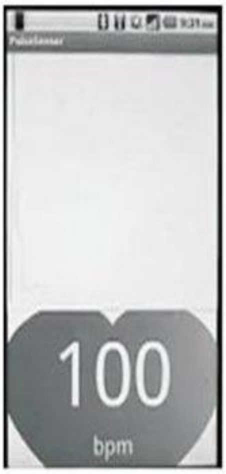


Fig. 7. Pulse Detection Application (Prior implementation)

V. CONCLUSION

In recent years, many android applications have emerged and had their impact in the people’s minds. Our work in this paper will deal with the human pulse rate detection which overcomes the drawbacks of the existing applications by providing a constant plot of the pulse rates.

Using the IDE emulator the application’s layout and design will be displayed.

VI. EXTENSIONS

This application can be further enhanced by adding additional informations like the blood pressures, namely, systolic and diastolic pressures [8]. Blood pressure (BP) or arterial blood pressure is nothing but the force or pressure exerted by blood against the walls of the blood vessels during the heartbeat. Measuring the BP involves measuring two things first systolic blood pressure (SBP), second diastolic blood pressure (DBP). The SBP occurs when heart beats and pushes the blood around the body. The DPB occurs when the heart get relax between beats.

The pulse sensor used in the application can be used to design theatre safety glasses [9] with the help of Arduino board. The glasses would have the pulse sensor on an earring clipped to one of the lobes. When the sensor detects exaltation in pulse servos on both sides of the glasses will close the curtains covering both eyes. The Arduino board and two demo buttons are housed in a small plastic box that fits in your pocket.The same sensor can also be used to build a pulse-sensing headband [12] that flashes a heartshaped LED display to the beating of your heart [13]. This provides a visualization of the original heartbeat. The pulses are measured exactly in the same way as that of the theatre safety glasses, i.e., through ear clipped pulse sensor. The Arduino board which is being involved in this application can be further enhanced by using a shrinkified Arduino board.

REFERENCES:

[1] Pulse Sensor, http://pulsesensor.myshopify.com/blog.
[2] Joel Murphy, (BFA Mass Art, Boston; MFA UC San Diego), http://pulsesensor.myshopify.com/pages/about-us.
[3] Yury Gitman, (BS Semiotics, Ga Tech, Atlanta; ITP NYU, New York), http://pulsesensor.myshopify.com/pages
[4] PulseSensor, http://www.youtube.com/watch?feature=player_embedded&v=5sS74MEzr2o
[5] Amarino, http://www.amarino-toolkit.net.
[6] AUBTM-22, Bluetooth V2.0 Class 1 Module Product datasheet.
[7] Eclipse, http://www.eclipse.org/.
[8]BloodPressures, http://www.diffen.com/difference/Systolic_vs_Diastolic_Blood_Pressure.
[9]Theatre safety glasses, http://www.youtube.com/watch?feature=player_embedded&v=H8nE_sYLmOk