Wireless Heart Rate Monitor

Ajay Shankar S, Vicky Daniel, K Vigneshwar, Vinnyeas K Naik
ajayshankarsriram@gmail.com, mkvigneshwar@gmail.com, vinnyas.k.naik@gmail.com
Department of Instrumentation and Control Engineering, National Institute of Technology, Tiruchirappalli, India

***Abstract*—We plan to make a portable wireless heart rate monitor by using an Arduino UNO as the brain due to its low power consumption, cost, and rich feature set. A PulseSensor from World Famous Electronics llc. Is selected to be the sensing element due to its low cost and small form factor. The data collected by the sensor will be collected and processed by the Arduino UNO and transmitted to a mobile application via Bluetooth. The Bluetooth transmission is made possible by the use of a Bluetooth transceiver module HC-05. The data will then be received and displayed using a custom app that has been developed in-house**

***Index Terms*—Heart rate Measurement, Photoplethysmography, Bluetooth, Wireless Monitoring**

I. Objective

The objective of the current project is to create a method to measure and monitor the heart rate of a subject to facilitate remote diagnosis and monitoring. There is a multitude of solutions that can achieve the objective. The disadvantage of these ready-made solutions are that they are either too bulky or too expensive to be viable for regular use by the common person.

We aim to reduce the size of the device to make it more portable. We also aim to make it more user-friendly and accessible to all the users by providing a mobile application that will interface with the device via Bluetooth and reduce the costs.

II. Heart Rate

A person's heartbeat is the sound of the valves in his/her heart contracting or expanding as they force the blood from one region to another. The number of times the heart beats per minute (BPM), is known as the heartbeat rate, also known as pulse rate. The pulse of the heart is felt in any artery that prevails adjacent to the epidermis.

The heart rate grows progressively slower as a person moves through childhood toward adolescence. The normal resting heart rate for adults over the age of 10 years, including older adults, is between 60 and 100 beats per minute (bpm) [1].Highly trained athletes have a resting heart rate below 60 bpm, sometimes reaching 40 bpm.

III. Measurement of Pulse Rate

Apart from indicating the soundness of the heart, heart rate helps assessing the cardiovascular system. Analysis of heart rate helps in diagnosis and detection of coronary diseases. Measuring heart rate has a long tradition in medicine as a non-invasive indicator of illness . An elevated resting heart rate has been associated with cardiovascular, and all-cause mortality independent of major cardiovascular risk factors, and with cancer mortality [2].

There are many devices available in the market to measure heart rate. The constructed device brings about some major changes compared to the available ones. Firstly, this is a cost-effective option to measure heart rate. Whereas the price of the devices already available in the market ranges from Rs. 3000 to Rs. 6000 [3], the device that has been constructed in this project cost roughly from Rs. 1000 to Rs. 1500. Secondly, the device provides an easy way to interface with personal computers and mobile phones. This opens up opportunities to perform further analysis of the recorded data. Thirdly, although there are a few analog devices to measure heart rate, the patient needs help from another person to do so. This device can be used without any need for assistance from others. Finally, the device is easy to use. The user only needs to place a fingertip on a sensor, as opposed to enclosing a fingertip inside a cap or wearing a cuff. The main component that has been used to detect heart rate in this project is an optical reflective sensor. An Arduino microcontroller has been used to count heartbeats and transmit it wirelessly to mobile phones and computers. Optionally a wired connection can be used to interface the device to a personal computer for further analysis.

IV. System Architecture

The functional block diagram, shown in Figure 1, depicts the fundamental design of the device.

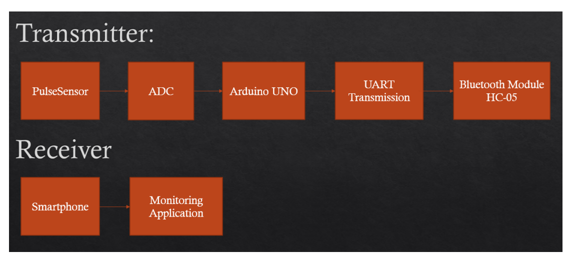


Fig. 1. Functional block diagram of the device

V. Hardware

The hardware that was used for this wireless heart rate monitor are listed below

A. Pulse Sensor

The working principle of the Pulse Sensor is photoplethysmography (PPG). PPG utilizes a high-precision light sensor to detect the volume of blood flow in order to observe heart rate. The heart rate can be recognized by measuring the changes in the intensity of the light passing through capillary blood vessels depending on the light absorption. The ideal waveform of a PPG signal is shown in Figure 2.

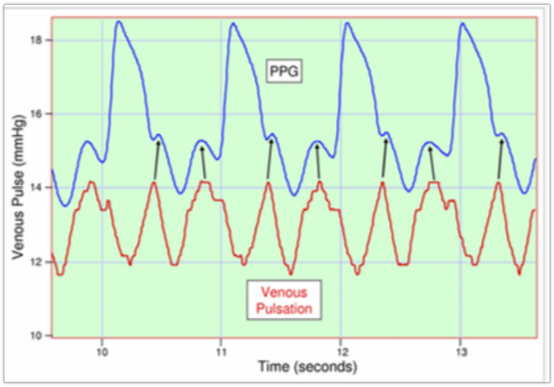
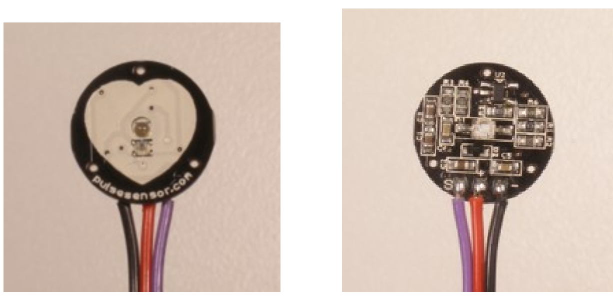


Fig. 2. PPG Signal

The Pulse Sensor (Figure 3) is a non-invasive sensor, meaning no insertions into the body. The Pulse sensor consists of a transmitter light generated by a green light emitting diode (LED), a reflective sensor based on a photo detector, an optical filter, and a power supply board.



(a) Front side

(b) Back side

Fig. 3. Pulse sensor module

An optical filter pulse is used to minimize the effects of ambient light, such as red and infrared rays. This enables high-quality pulse signals to be acquired, even indoors and outdoors. The low brightness of the LED used makes it possible to achieve a low-power HR monitoring system without the need for a boost circuit. In addition, this contributes to longer operating times in a wearable device with a limited battery capacity.

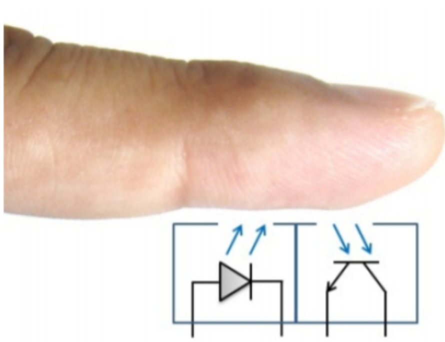


Fig. 4. Reflective Photoplethysmography

B. Microcontroller

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. The ATmega328P is programmed using C++. Some off-the-shelf libraries can be adopted with some modifications to improve measurement performance and support the system's successful use. In this work, microcontroller programming performs three main activities: Monitoring, processing, and transmitting to the Bluetooth module. First, the microcontroller processes the data received via the analog input pin, A0. Then, the microcontroller calculates the bpm based on the raw sensor data. Finally, the microcontroller will transmit the bpm data via the serial port (Tx) at a rate of 9600 kbps, and then the Bluetooth module will transmit the data to the mobile application via the builtin antenna using the standard Bluetooth communication protocol.

C. Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is a fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with a complete 2.4GHz radio transceiver and baseband. It uses the CSR Bluecore 04-External single-chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

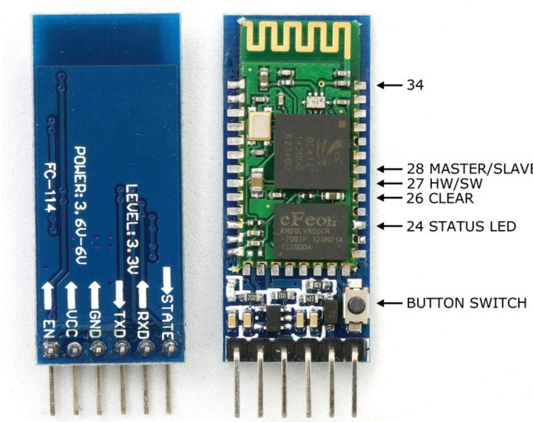


Fig. 5. Bluetooth Module HC-05

The data is sent to the module by using the SoftwareSerial library using the UART protocol at the baud rate of 9600.

VI. Android Application

The accompanying Android application was developed using Android studio as explained in section VI-A. The classes' Bluetooth adapter and Bluetooth socket were used to identify the connected Bluetooth device and extract the data that was being sent by the HC-05 module. Refer to the section VI-B and VI-C for more information.

A. Android Studio

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.[4]

To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and Github integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.[5]

Android Studio uses an Instant Push feature to push code and resource changes to a running application. A code editor assists the developer with writing code and offering code completion, refraction, and analysis. Applications built in Android Studio are then compiled into the APK format for submission to the Google Play Store.

The software was first announced at Google I/O in May 2013, and the first stable build was released in December 2014. Android Studio is available for Mac, Windows, and Linux desktop platforms. It replaced Eclipse Android Development Tools (ADT) as the primary IDE for Android application development. Android Studio and the Software Development Kit can be downloaded directly from Google.[6]

B. Bluetooth Adapter

Represents the local device Bluetooth adapter. The BluetoothAdapter lets you perform fundamental Bluetooth tasks, such as initiate device discovery, query a list of bonded (paired) devices, instantiate a BluetoothDevice using a known MAC address, and create a BluetoothServerSocket to listen for connection requests from other devices, and start a scan for Bluetooth LE devices.

To get a BluetoothAdapter representing the local Bluetooth adapter, call the BluetoothManagergetAdapter function on BluetoothManager. On JELLY_BEAN_MR1 and below you will need to use the static getDefaultAdapter() method instead.

Fundamentally, this is your starting point for all Bluetooth actions. Once you have the local adapter, you can get a set of BluetoothDevice objects representing all paired devices with getBondedDevices(); start device discovery with startDiscovery(); or create a BluetoothServerSocket to listen for incoming RFComm connection requests with listenUsingRfcommWithServiceRecord(java.lang.String, java.util.UUID); listen for incoming L2CAP Connection-oriented Channels (CoC) connection requests with listenUsingL2capChannel(); or start a scan for Bluetooth LE devices with startLeScan(android.bluetooth.BluetoothAdapter.LeScanCallback).

This class is thread safe.[7]

C. Bluetooth Socket

A connected or connecting Bluetooth socket.

The interface for Bluetooth Sockets is similar to that of TCP sockets: Socket and ServerSocket. On the server side, use a BluetoothServerSocket to create a listening server socket. When a connection is accepted by the BluetoothServerSocket, it will return a new BluetoothSocket to manage the connection. On the client side, use a single BluetoothSocket to both initiate an outgoing connection and to manage the connection.

The most common type of Bluetooth socket is RFCOMM, which is the type supported by the Android APIs. RFCOMM is a connection-oriented, streaming transport over Bluetooth. It is also known as the Serial Port Profile (SPP).

To create a BluetoothSocket for connecting to a known device, use BluetoothDevicecreateRfcommSocketToServiceRecord. Then call connect() to attempt a connection to the remote device. This call will block until a connection is established or the connection fails.

To create a BluetoothSocket as a server (or ”host”), see the BluetoothServerSocket documentation.

Once the socket is connected, whether initiated as a client or accepted as a server, open the IO streams by calling getInputStream() and getOutputStream() in order to retrieve InputStream and OutputStream objects, respectively, which are automatically connected to the socket.

BluetoothSocket is thread safe. In particular, close() will always immediately abort ongoing operations and close the socket. [8] [9]

VII. Results

The data from the sensor was plotted using the serial plotter present in the Arduino IDE. In the plot, the raw data (green), BPM value(blue), and the R-R timing (red) of each beat is shown. Presently, only the BPM value is only transmitted to the application to reduce the amount of information the user has to analyze thereby making improving the ease of use. The rest of the data can be observed by connecting the device to a computer and using the serial plotter in the Arduino IDE.

All tests were conducted with the sensor placed on the tip of the left index finger of the subject.

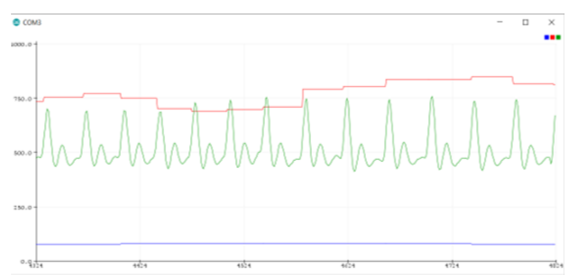


Fig. 6. Output with a calm subject in a seated position

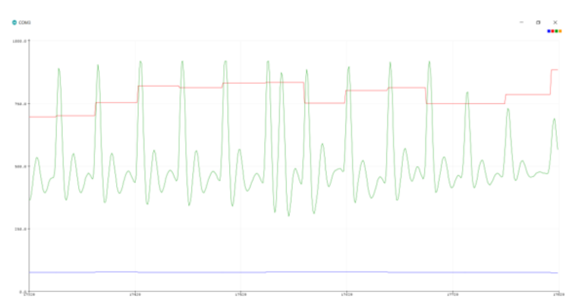


Fig. 7. Output with a mildly excited subject in a seated position

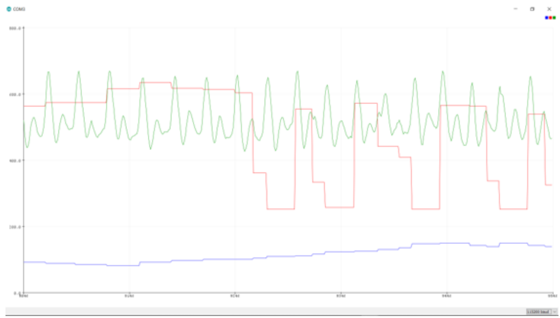


Fig. 8. Output with a calm subject in a seated position with improper pressure

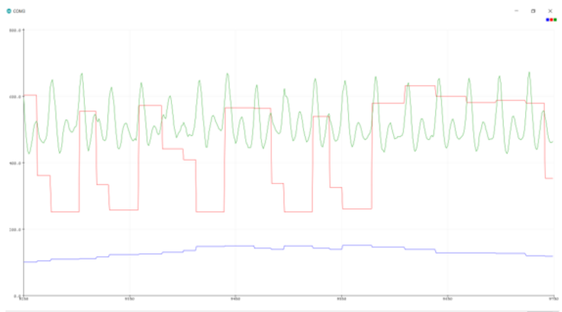


Fig. 9. Output with a calm subject in motion

VIII. Conclusion

The device produces results that are in the acceptable range based on current testing. Compared to other available options, this is a cost-effective and compact solution to monitor heart rate wirelessly. However, with a few changes, the device can be made a bit more reliable. Currently while measuring the heart rate the body part held against the sensor must be kept very still. Even a slight movement results in a bit of variation. This can be avoided by using a different optical reflective sensor. A sensor-based on transmittance PPG could also be used for comparison, as they are less susceptible to motion artifacts. The surface of the sensor also has exposed contacts that lead to an increased rate of failure of the sensor modules. This can be rectified by coating the modules in an optically neutral resin/epoxy coating to make them weather resistant.

Studies have shown that PPG can also be used to measure peripheral capillary oxygen saturation (SpO2) [10], blood pressure [11], and respiratory rate. If future studies ensure acceptable results for blood pressure, oxygen saturation and respiratory rate using PPG; attempts can be taken to devise a single module having all four features. As the basic mechanism is the same, chances are the circuitry will be similar, resulting in a compact device. This will, in turn, result in a cost-effective way to create a device capable of measuring heart rate, blood pressure, oxygen saturation, and respiratory rate.

IX. Future Development

Further testing and validation of the apparatus is required to determine the accuracy of the device before implementing it as a product. Design of an ergonomic enclosure for it to be placed obstructively on the subject. An accelerometer can also be added to aid in the detection of fall for the monitoring of patients [12]. A customized sensor module that is resistant to water can be developed to improve the life of the sensor.

More analytical and diagnostic features such as data storage for long term analysis, plotters, etc., can be added to the application to improve the functionality and ease of diagnosis. Further study has to be conducted to find a better place on the body to position the sensor that will reduce the motion artifacts.

References

[1] Heart.org. *All About Heart Rate (Pulse)*. URL: https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse#.Wg1mcBO0OCU (visited on 07/31/2015).

[2] G.B.M. Mensink and H. Hoffmeister. "The relationship between resting heart rate and all-cause, cardiovascular and cancer mortality". In: *European Heart Journal* 18 (1977).

[3] Amazon. *Amazon.com Exercise Fitness: Heart Rate Monitors*. URL: http://goo.gl/K5m2QM (visited on 07/31/2015).

[4] Wikipedia. *Android Studio*. URL: https://en.wikipedia.org/wiki/Android_Studio.

[5] Margaret Rouse. *What is Android Studio?* URL: https://searchmobilecomputing.techtarget.com/definition/Android-Studio.

[6] Google. *Download Android Studio*. URL: https://developer.android.com/studio.

[7] Google. *BluetoothAdapter*. URL: https://developer.android.com/reference/android/bluetooth/BluetoothAdapter.

[8] Google. *BluetoothSocket*. URL: https://developer.android.com/reference/android/bluetooth/BluetoothSocket.

[9] TutorialsPoint. *Android-Bluetooth*. URL: https://www.tutorialspoint.com/android/android_bluetooth.htm.

[10] G. Ates and K. Polat. "Measuring of oxygen saturation using pulse oximeter based on fuzzy logic". In: (2012).

[11] M.S. Ahsan M. Asif-Ul-Hoque and B. Mohajan. "Measurement of Blood Pressure Using Photoplethysmography". In: *13th International Conference on Modelling and Simulation* (2011).

[12] Saif Saad Fakhrulddin and Sadik Kamel Gharghan. "An Autonomous Wireless Health Monitoring System Based on Heartbeat and Accelerometer Sensors". In: *Journal of Sensors and Actuator Networks* (2019).