Arduino based Heart Rate Monitoring System

Prof Sridevi A, Pranav Raghavan C M, Shravani A Gowda, Srishanth K M, Vibha Shetty
Department of Electrical and Electronics Engineering, Dr.T. Thimmaiah Institute of Technology, KGF, India

Abstract: This project presents an Arduino-based heart rate monitor using a pulse sensor and Arduino Uno. It provides an affordable solution for real-time heart rate monitoring, suitable for fitness tracking and preliminary health diagnostics. The system processes blood flow data from a pulse sensor to calculate Beats Per Minute (BPM) and displays the results using Arduino IDE. This prototype is low-cost, accessible, and serves educational, research, and healthcare purposes.

Keywords: Arduino, Heart Rate Monitoring, Pulse Sensor, BPM, Real-Time Monitoring, Embedded Systems.

I. INTRODUCTION

The heart is a vital organ responsible for circulating blood and oxygen throughout the body. Heart rate is a critical parameter for assessing cardiovascular health. Traditional heart rate monitoring devices are often costly and not easily accessible. This project utilizes an Arduino Uno and pulse sensor to create a low-cost, real-time heart rate monitoring system. It aims to provide an effective alternative for use in educational, fitness, and medical applications.

II. LITERATURE REVIEW

Several research efforts have demonstrated the effectiveness and versatility of Arduino-based heart rate monitoring systems. These systems integrate affordable sensors with microcontrollers to capture and process biometric signals, making real-time health monitoring more accessible.

Sahana S. Khamitkar and Prof. Mohammed Rafi proposed an IoT-enabled heart rate detection system using a PPG sensor and Arduino, transmitting data via Bluetooth to an Android app for monitoring and storage. This setup offers a low-cost, mobile solution for patient monitoring [1]. Lantana Dioren Rumpa et al. explored mood detection using an ECG sensor with the Arduino Uno and e-Health Sensor Platform. Their system captured detailed bio signals including heart rate and visualized them using KST Plot, demonstrating the feasibility of emotion analysis from heartbeat patterns [2].

Bong Siaw Wee and D. Kong Shuet Fang developed a dual-purpose system for body temperature and pulse rate tracking. Their Arduino-based device displayed data on an LCD and proved effective for early disease detection, particularly in resource-limited environments [3]. Yesica Devis and team created a comprehensive patient monitoring system integrating heart rate, temperature, and infusion levels. The system used sound alerts for abnormalities, although it had limitations in storing historical data due to the I2C LCD used [4].

K. Shirisha et al. implemented a health monitoring system in ICU scenarios using IoT. Their design included pulse rate and other sensors with data uploaded to ThingSpeak for analysis and alert generation. This work emphasizes the use of Arduino in critical care [5]. T. Arun Prasath and co-authors introduced a heart attack detection system using IoT and multiple biosensors. By analyzing real-time data, their device could identify abnormal patterns and issue emergency alerts, highlighting its importance in preemptive healthcare [6].

Gurinder Singh et al. focused on combining ECG, pulse oximeter, temperature, and BPM sensors using

an Arduino and cloud platforms like Firebase for continuous data logging and analysis. Their work showcases the growing role of cloud integration in Arduino health projects [7]. Muhammad Hafiz Zulhilmi Zulkifli and his colleagues evaluated the efficiency of piezoelectric sensors to differentiate between stress and relaxed states based on heart rate. They suggested future improvements using machine learning and compact sensor designs for better accuracy [8].

III. METHODOLOGY

The development of the Arduino-based heart rate monitoring system involved both hardware and software components working together to provide real-time and accurate BPM (Beats Per Minute) readings.

A. System Architecture

The system comprises a pulse sensor, Arduino Uno microcontroller, jumper wires, and a serial communication interface. The pulse sensor detects the heartbeat signal via reflected light variations and transmits it to the Arduino Uno for processing. explain with dis only shortly.

B. Component Selection

1. Pulse Sensor: Detects heartbeat using photoplethysmography (PPG). It measures changes in light absorption through a fingertip or earlobe due to blood volume changes.

2. Arduino Uno: Serves as the primary microcontroller that reads the analog signal from the pulse sensor, processes it, and calculates the BPM.

C. Software Development

The Arduino IDE is used to write and upload the program. The PulseSensor Playground library simplifies the BPM calculation process. Key steps include:

- Initializing serial communication.
- Setting a threshold for heartbeat detection.

- Using the library functions to detect pulses and compute BPM.
- Displaying data in real time through the Serial Monitor.

D. System Workflow

1. The pulse sensor is connected to the Arduino's analog input.
2. The sensor outputs analog signals representing pulse waveforms.
3. The Arduino reads the signal; filters noise and detects peaks corresponding to heartbeats.
4. BPM is computed using the time difference between successive peaks.
5. Results are displayed in real time on a serial interface.

E. Testing and Validation

To ensure the accuracy of the system, the BPM values obtained were compared with manual pulse counts across different age groups and physical conditions. The results confirmed consistency and acceptable precision for non-clinical applications. This structured methodology ensures the system is robust, repeatable, and adaptable for further enhancements like wireless transmission or mobile display.

IV. IMPLEMENTATION

The implementation phase focuses on both hardware setup and software integration to realize a functional Arduino-based heart rate monitoring system.

A. Circuit Design

The pulse sensor is connected to the Arduino Uno using jumper wires. The sensor's signal pin is attached to analog pin A0, with power and ground connected to the 5V and GND pins on the Arduino respectively. This simple configuration allows for effective data acquisition from the user's fingertip or earlobe.

B. Hardware Components

1. Arduino Uno – Acts as the core processing unit, executing the program logic.

2. Pulse Sensor – Senses the heart rate by detecting changes in blood flow using photoplethysmography.

C. Software Implementation

The Arduino IDE is used to write and upload code. The program utilizes the PulseSensor Playground library, which simplifies signal acquisition and BPM calculation. The code initializes serial communication, sets thresholds, and prints BPM data to the serial monitor.

D. Program Algorithm

1. Initialize pulse sensor and serial monitor.

2. Read analog data from the sensor.

3. Detect heartbeats using threshold detection.

4. Calculate the time between beats to determine BPM.

5. Display the BPM value in the serial monitor.

E. Output Interface

The real-time BPM output is displayed through the Arduino Serial Monitor. Optional feedback such as LED blinking with each heartbeat can be enabled for visual indication. This implementation effectively demonstrates how embedded systems can be leveraged for health monitoring applications, combining sensor input and microcontroller processing with minimal hardware.

V. RESULTS AND DISCUSSION

The system was tested on users from different age groups to evaluate the accuracy and stability of BPM readings.

A. Experimental Setup

The setup includes the pulse sensor connected to the Arduino Uno and interfaced with a laptop via USB. The user places a fingertip on the pulse sensor, and BPM is displayed in real time on the Arduino Serial Monitor.
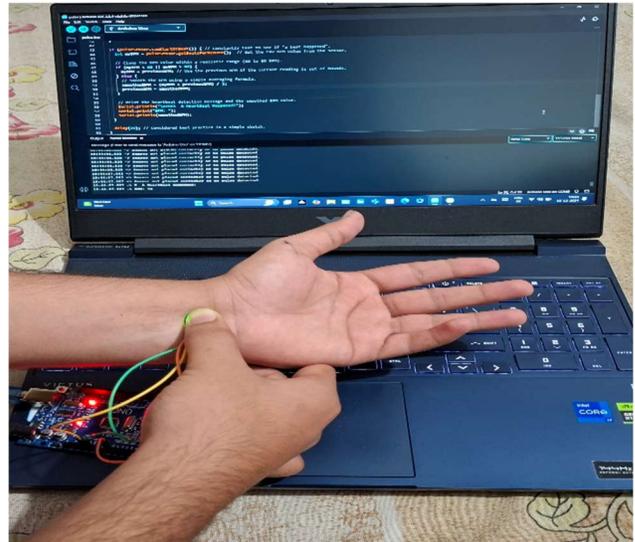

Fig 1: Reading obtain in Trail.

B. Observations

Fig 2: Output obtain in Heart rate Monitoring.

Output is typically used for real-time heart rate monitoring, where the Arduino processes sensor input to calculate and display heart rate for different ages.

C. Output Interpretation

Each heartbeat is detected and logged with a timestamp. The system calculates BPM by measuring time intervals between consecutive heartbeats. Results are updated every few seconds, ensuring real-time monitoring.

Table 1: Average output reading

Age group	Average BPM
12-15	72.5
15-20	73.7
30	69.5
40	71.4
50	69
60	63.6

D. Graphical Output

Graphical analysis of BPM values across age groups confirmed expected physiological trends—lower heart rates in older individuals. These results validate the effectiveness of the system in monitoring heart activity.
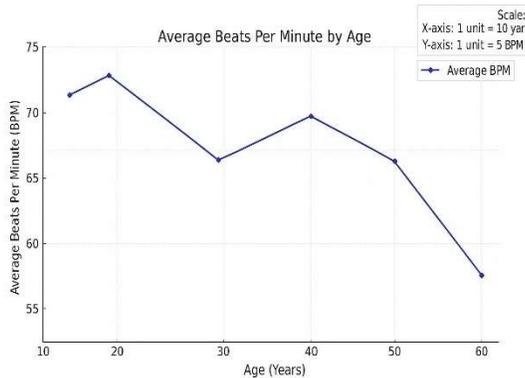
Fig 3: Output Graph

VI. DISCUSSION AND FINDING

The Arduino-based heart rate monitoring system demonstrated effective performance in detecting and displaying real-time BPM across users of various age groups. The results showed good consistency with manual pulse readings, typically within a margin of ±3–5 BPM. The system effectively filtered noise and detected pulse peaks, validating the functionality of the PulseSensor Playground library. During testing, minor inaccuracies occurred due to finger movement, improper sensor contacts, or variations in skin tone, which are common challenges in optical pulse detection. Nevertheless, the system provided reliable and stable output under normal conditions. These findings highlight the suitability of the setup for non-clinical environments such as fitness tracking, academic projects, and initial health screenings. The successful integration of low-cost components with open-source software confirms the potential of Arduino-based systems for accessible and portable biomedical monitoring solutions.

VII. CONCLUSION

This project presents a comprehensive design and implementation of an Arduino-based heart rate monitoring system that offers a cost-effective, portable, and real-time solution for tracking heart rate. Through the integration of a pulse sensor with the Arduino Uno microcontroller and the use of the PulseSensor Playground library, the system was able to accurately detect heartbeat signals, calculate Beats Per Minute (BPM), and display results in real-time using the Arduino IDE's Serial Monitor. Experimental results confirmed that the BPM values recorded were consistent with standard physiological expectations and manual pulse measurements, validating the system's functionality. The simplicity of the hardware setup, combined with the accessibility of Arduino programming, makes this system particularly valuable for educational purposes, student projects, and basic personal health monitoring in non-clinical environments.

The system also highlights the potential of embedded systems in biomedical applications, especially in scenarios where affordability and ease of use are essential. While the system performs well under stable conditions, limitations such as motion artifacts, sensor placement sensitivity, and ambient light interference were observed. These findings open avenues for future enhancements, including implementing wireless data transmission via Bluetooth or Wi-Fi modules, developing a mobile application for remote monitoring, and integrating more sophisticated algorithms or filtering techniques to improve accuracy and user experience. In conclusion, the project successfully demonstrates

how open-source platforms like Arduino can be harnessed to develop scalable and customizable health monitoring tools that serve as both educational models and functional prototypes for broader biomedical applications.

REFERENCES

[1] Sahana S Khamitkar, Prof. Mohammed Rafi, "IOT BASED SYSTEM FOR HEART RATE MONITORING" Vol 9 IJERT_IoT_based_System_for_Heart_Rate_Mo[1].pdf Issue 07, July-2020

[2] Lantana Dioren Rumpa, Sallolo Suluh, Irene Hendrika Ramopoly, Wilson Jefriyanto, "Development of ECG sensor using Arduino uno and e-health sensor platform: mood detection from heartbeat", Jornal Physics: Conference Series 1528 012043 Dioren_Rumpa_2020_J._Phys.__Conf._Ser._1528_012043[1].pdf

[3] Bong Siaw Wee, D Kong Shuet Fang, "Design and Implementation of an "Arduino-Based Body Temperature and Pulse Rate Monitoring System", International Applied Business and Engineering Conference 2021 August 25, 2021, Riau, Indonesia, IABEC_ENG_0002[2].pdf

[4] Yesica Devis, Yuda Irawan, Junadhi, Fransiskus Zoromi, Herianto, Mohd Rinaldi Amartha, "Monitoring System of Heart Rate, Temperature and Infusion in Patients Based on Microcontroller (Arduino Uno)" Journal of Physics: Conference Series, ICIASGA 2021, Devis_2021_J._Phys.__Conf._Ser._1845_012069[1].pdf

[5] K. Shirisha, Kammari. Bharath, Nimmala. Mounika , O.Venkatalaxmi , Nakka. Ashwini, "Arduino based ICU patient health monitoring system over IoT", International Journal for Advanced Research in Science &Technology (IJARST), Volume 12, Issue 12, Dec 2022, ISSN 2457 – 0362. 931871676787793[1].pdf

[6] T Arun Prasath, M Mohamed Arif, S Srinivasan, A Muthumanoj kumar, M Sushmitha, Sakthivel Sankaran, "IoT Based Heart Attack Detection and Heart Rate Monitoring System", The publisher of the project is IEEE. IoT Based Heart attack Detection and Heart Rate Monitoring System | IEEE Conference Publication | IEEE Xplore

[7] Gurinder Singh, Naina Chaudhary, Danish Ather, Rajneesh Kler, Manik Arora, "Arduino based ICU patient health monitoring system over IoT", published by IEEE in the year 2023. Revolutionizing Remote Healthcare: Proposing An IoT & Arduino-Based Integrated Approach for Real-Time Health Monitoring | IEEE Conference Publication | IEEE Xplore 5th sem, Dept of EEE, Dr.T.T.I.T, KGF 36 Arduino Based Heart rate monitoring system 2024-25.

[8] Muhammad hafiz zulhilmi zulkifli , farahiyah jasni , aimi shazwani ghazali , nor hidayati diyana nordin, shahrul nalim sidek1, syamimi mardiah shaharum, fadhlan hafizhelmi kamaru zaman, "Exploring the efficacy of piezoelectric based sensory systems for heart rate monitoring in differentiating stress vs relax conditions", IIUM Engineering Journal, Vol. 25, No. 2, 2024. 3017-Article Text-20313-20016-10 20240704[1].pdf