International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 04 | April – 2025 DOI: 10.55041/ISJEM02643
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

HEART RATE AND TEMPERATURE MONITORING

Dhanashree Kailas Dhapare
Zeal Polytechnic, Pune
Vaishnavi Dhananjay Shinde
Zeal Polytechnic, Pune
Reenal Kantilal Chauhan
Zeal Polytechnic, Pune
Prof.Pranjali.J.Kajale Lecture,
Zeal Polytechnic, Pune
Shrutika Hanmant Waghmare
Zeal Polytechnic, Pune

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 04 | April – 2025 DOI: 10.55041/ISJEM02643
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Abstract— The Heart Rate and Temperature Monitoring System Using Arduino Uno is an automated and economical solution for health monitoring. This implementation combines the Arduino Uno microcontroller and two fundamental sensors, a Pulse Sensor which detects the heart rate and DS18B20 temperature sensor to measure the body temperature. The specific sensor known as a Pulse Sensor indicates the heart rate in beats per minute (BPM) through light absorption which detects blood volume change. At the same time, the DS18B20 sensor measures the body temperature and converts the analog signal into a digital temperature resolute, which is shown in Celsius. The Arduino Uno microcontroller processes the data and displays it on a 16x2 Liquid Crystal Display (LCD) for monitoring purposes. This system has significant importance in the healthcare field, providing ease in monitoring vital signs in real time using low-cost and portable devices.

INTRODUCTION

A heart rate and temperature monitoring system using Arduino is a project that allows real-time tracking of a person's heart rate and body temperature. By integrating sensors like the Pulse Sensor and a temperature sensor,
this system can measure vital signs and display them on an LCD or serial monitor. The heart rate sensor detects
the pulse by measuring blood flow changes in the fingertip or earlobe, while the temperature sensor provides
accurate body temperature readings.
Arduino, a popular microcontroller platform, processes these sensor readings and displays the results, often with
an alarm system that activates when thresholds are exceeded. This system can be used in healthcare applications,
fitness monitoring, or for general wellness tracking. The easy-to-use Arduino IDE makes it simple to program
and customize the system, providing an excellent introduction to sensor integration and basic health monitoring technology.

METHODOLOGY

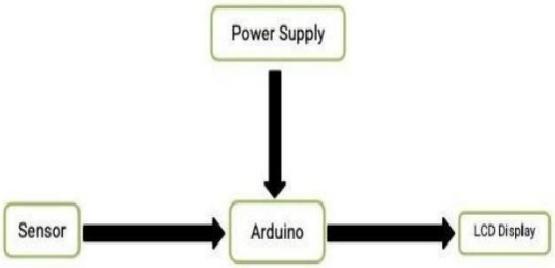

Fig. Block Diagram of Heartrate and Temperature Monitoring System

1. Hardware Setup:

   Heart Rate Sensor Integration: Connect the Pulse Sensor (or similar heart rate sensor) to Arduino Uno for heart rate monitoring. Typically, the signal wire from the pulse sensor is connected to an analog pin (e.g., A0).

   Temperature Sensor Integration: Connect the DS18B20 temperature sensor to the Arduino Uno using the OneWire protocol. The VCC pin is connected to the 5V pin, the GND pin to GND, and the Data pin to a digital pin (e.g., D2) on the Arduino.

   Display Interface: Interface a 16x2 I2C LCD to the Arduino Uno for displaying real-time feedback. The SDA and SCL pins of the LCD connect to A4 and A5 pins on the Arduino, respectively, for I2C communication.

   Power Supply Integration: Ensure the Arduino Uno and all connected components receive power via the 5V pin. Optionally, use an external DC power supply if required, but the standard 5V USB or battery supply should suffice for most applications.

2. Data Acquisition:

Heart Rate Data: Use the Pulse Sensor connected to an analog pin (e.g., A0) to acquire the heart rate data. The sensor detects pulse fluctuations due to blood flow, which the Arduino reads as an analog signal.

Temperature Data: Retrieve temperature data from the DS18B20 sensor using the OneWire library and DallasTemperature library. These libraries will facilitate communication with the sensor, enabling the Arduino to read the temperature in Celsius.

Data Protocols: Implement data reading protocols for both sensors. For the Pulse Sensor, filter out noise and false readings by using appropriate threshold values or signal processing techniques. For the DS18B20, request temperature readings at regular intervals (e.g., every 1 second) to ensure continuous data updates.

3. Data Processing:

Heart Rate Calculation: Write a function to process the raw signal from the Pulse Sensor. This function will detect the pulse peaks, calculate the time intervals between them, and compute the heart rate in beats per minute (bpm).

Temperature Calculation: Use the DallasTemperature library to request temperature data from the DS18B20 and convert it to Celsius.

Signal Processing: Implement basic signal processing (e.g., noise filtering) for both sensors. The heart rate signal may require filtering to ensure smooth readings, and the temperature reading should be handled with appropriate delays for stability.

Vital Parameter Calculation: After processing, calculate the vital parameters (heart rate in bpm and body temperature in °C) and prepare them to be displayed on the LCD.

4. User Interface:

Display Design: Design a user-friendly interface on the 16x2 I2C LCD to display real-time heart rate and temperature data. The LCD will show:

Heart Rate (in beats per minute)
Temperature (in Celsius)

Ease of Use: Ensure that the interface is simple, clean, and readable. The font size and layout should allow users to quickly understand their vital signs.

Power Management:
Power Regulation: Power the Arduino Uno through its USB connection or 5V pin if using an external supply. Since the system uses low-power sensors and components (LCD and pulse sensor), this will be sufficient for typical use.

Power Efficiency: Since the Arduino Uno doesn't have advanced power-saving modes, the power management mainly involves efficient sensor polling intervals and display updates. Use a delay of 1 second between updates to save power.

Battery Management: If using a battery to power the Arduino Uno, ensure that the battery is adequately sized for continuous operation, considering the power consumption of the Pulse Sensor, DS18B20, and LCD.

5. Integration and Testing:

System Integration: Integrate the Pulse Sensor, DS18B20, and 16x2 I2C LCD into the system with the Arduino Uno. Ensure that all connections are secure and that each component is properly powered.

Thorough Testing: Test the system thoroughly: Ensure the heart rate sensor accurately detects pulse signals and provides stable readings.
Verify the temperature sensor returns accurate body temperature readings. Check that the LCD updates the readings correctly and clearly.

Debugging: If any issues arise (e.g., noise in the pulse sensor signal or inaccurate temperature readings), adjust the sampling rate, filtering techniques, or wiring to resolve these issues.

6. Deployment and Evaluation:
   Real-World Usage: Once testing is complete, deploy the health monitoring system for use. This could be for personal health monitoring, fitness tracking, or clinical applications (with further evaluation).

   User Feedback: Collect feedback from end users regarding system performance and ease of use. Users should feel comfortable interpreting the heart rate and temperature readings displayed on the LCD.

   Iterative Improvement: Based on feedback and testing results, make improvements to the system: Adjust the display interface if needed.
   Improve sensor calibration or modify the threshold values for alerts.
   If necessary, explore adding additional features like Bluetooth connectivity for wireless data transmission to a mobile app or computer.

BENFITS

1. Early Symptom Identification – Detection of heart rate or body temperature abnormalities can suggest infection, fever, or heart problems.
2. Disease Control – Assists in the management of chronic illnesses including hypertension, arrhythmia, and diabetes.
3. Fever Monitoring – Helps to detect infections in at risk and remote populations.
4. Stress & Fatigue Monitoring – Variability of heart rate measures (HRV) assists in measuring both stress and general health.
5. Fitness & Sports Achievement Optimization of Exercise – Athletes can monitor their heart rate to ensure they train within specific heart rate zones.
6. Avoiding Overexertion – Body temperature is monitored to prevent heat exhaustion or hypothermia.
7. Tracking Recovery – Evaluation of recovery post workouts or post-injuries is easier.
8. Wearable's & IoT Technology Smart Watchers – Incorporated into fitness trackers, smart watches, and other health-monitoring tools.
9. Fever Checking – Allows doctors to check the patient's vital statistics without being at the hospital.
10. Elderly Care – Smart house system technologies can be expanded for use with elderly patients.
11. Workplace & Advanced Industry Safety Employee Wellness – Assists workers in direct, physically extreme work environments (firefighters, factory workers) manage heat stress.transportation, and other priority vehicles.

I. RESULTS

1. Heart Rate Measurement: The system uses a Pulse Sensor connected to the Arduino to measure the user's heart rate in beats per minute (BPM).
2. Body Temperature Measurement: The DS18B20 temperature sensor measures the user's body temperature in Celsius and displays it on the LCD screen.
3. LCD Display: A 16x2 LCD (I2C) display shows both the heart rate (BPM) and body temperature simultaneously, allowing for easy monitoring.
4. Data Validation: The system ensures that the heart rate falls within the range of 40-180 BPM and the temperature stays between 20°C-45°C. Invalid readings are either not shown or display a warning message.
5. Simple User Interface: The LCD screen provides an easy-to-read, user-friendly interface, displaying both heart rate and temperature for quick monitoring.
6. Continuous Monitoring: The system consistently monitors the user's heart rate and temperature without interruption, providing updated readings regularly.
7. Pulse Detection: The Pulse Sensor detects the user's pulse when a finger is placed on the sensor, and the heart rate is displayed based on the readings.

8. Temperature Detection: The DS18B20 sensor detects body temperature and displays the value on the LCD screen, providing an accurate temperature measurement.
9. Practical for Home Use: The system is ideal for personal health monitoring at home, helping users track their heart rate and body temperature conveniently.

II. FUTURE SCOPE

1. Wireless Monitoring (Bluetooth/Wi-Fi Integration) Bluetooth Module (HC-05/HC-06) or Wi-Fi (ESP8266/ESP32): Integrating wireless communication can allow you to send real-time heart rate and temperature data to a smartphone app or a web-based dashboard.

   Cloud Storage: You could store the data on the cloud (e.g., Google Firebase, AWS, or ThingSpeak) to track the user's health data over time and enable remote monitoring by healthcare professionals.

2. Mobile App for Health Tracking:

   Develop a mobile app (Android or iOS) that connects to your monitoring system via Bluetooth or Wi-Fi. The app can display the heart rate, temperature, and historical data, allowing the user to track their health over time.

   The app can also send notifications when readings fall outside normal ranges.

3. Multi-Sensor Integration:

   Blood Pressure Sensor: Add a blood pressure sensor to make your system a full-fledged health monitoring system. This could measure systolic and diastolic pressure.

   ECG/EKG Sensor: Integrate an ECG sensor to measure the electrical activity of the heart. This would allow for a more comprehensive cardiac health analysis.

   SpO2 (Blood Oxygen Level): Use a pulse oximeter sensor to measure oxygen saturation levels (SpO2), which can be critical for monitoring respiratory health.

4. Wearable Health Monitoring System:

   Portable/Compact Design: You can convert your monitoring system into a wearable device such as a wristband or chest patch, making it more convenient for continuous health tracking.

   Integrate a battery to make the system portable, allowing the user to wear it throughout the day and monitor their health continuously.

5. Voice Assistance Integration:

   Voice Control: Integrate the system with voice assistants such as Amazon Alexa or Google Assistant. This would allow users to ask about their current heart rate and temperature, and receive updates verbally.

V . CONCLUSION

This hand-crafted system is inexpensive and simple to operate as it monitors heart rates and temperature using pulse and DS18B20 temperature sensors. This system is tailored for people suffering chronic conditions as well as athletes to enable continuous monitoring of health. With Arduino’s vast libraries, the system is accessible for novice and expert users alike. Additonally, the system can easily be deployed in homes, hospitals, and fitness centers highlighting its portability. Overall, this system is simple yet effective enhancing health management and research applications.

VI. ACKNOWLEDGMENT

I would like to thank my respected guide
PROF. P.J. KAJALE and HOD. SONALI NAVALE for their support.

VII. REFERENCES

1. Smart Heart Rate Monitoring System Using Arduino-Uno (November 2023) by Md. Abdullah Al Hasan et al.
2. Analysis of Heart Rate and Body Temperature from the Wireless Monitoring System Using Arduino (August 2023) by Afandi Nur Aziz Thohari et al.
3. The System for Human Temperature Detector

and Pulse Analyze by Kartik Pisudde et al.

4. The System for Human Temperature Detector and Pulse Analyzer by Kartik Pisudde, Suraj Patil, and R. S. Sannakki.
5. An IoT-Based Health Monitoring System Using Arduino Uno by S. Priya, P. Priyanka, R. Santhosh Kumar.
6. https://www.researchgate.net/publication/317117507_Heartbeat_and_Temperature_Monitoring_System_for_Remote_Patients_using_Arduino