Project Title:

Real-Time Health Monitoring System

Course:

Embedded Systems and IoT Lab

Team Members:

1. Sabit Ahmed Preanto [221-15-5059]
2. Faria Khan [221-15-5985]
3. Tapon Paul [221-15-5086]
4. Abid Khan [221-15-4894]
5. Md Didarul Islam [221-15-4679]

Abstract

The Real-Time Health Monitoring System integrates embedded systems with biomedical sensors to provide continuous monitoring of vital signs. Using an ESP8266 microcontroller with ECG and pulse oximeter sensors, this project measures and displays ECG waveform, heart rate, and SpO_2 levels on a web-based dashboard. Data is transmitted via USB serial communication and visualized in real-time using index.html, with built-in alert mechanisms for abnormal readings.

1. Introduction

Health monitoring plays a vital role in early detection of conditions such as heart disease and respiratory issues. Parameters like ECG, heart rate (BPM), and blood oxygen saturation (SpO_2) provide important indicators of a person's health. However, conventional monitoring systems are often expensive and limited to hospital use.

The Real-Time Health Monitoring System in this project offers a low-cost, portable alternative by combining biomedical sensors with web-based visualization. The AD8232 ECG sensor captures heart electrical activity, and the MAX30100 pulse oximeter measures heart rate and SpO_2. These sensors interface with an ESP8266 microcontroller, which streams data via USB to a browser-based dashboard.

Developed with HTML, CSS, JavaScript, and the Web Serial API, the dashboard displays live ECG waveforms, heart rate, and SpO_2 readings, with alerts for abnormal values. This system provides an accessible, user-friendly tool for continuous health monitoring outside traditional clinical environments.

2. Objectives

- To design and develop a USB-connected real-time health monitoring system.
- To measure ECG, Heart Rate, and SpO_2 continuously and visualize them through a web dashboard.
- To implement threshold-based alerts for abnormal readings.
- To ensure user-friendly UI for immediate interpretation of data.

3. System Architecture

3.1 Components Used

- Microcontroller: ESP8266 for sensor interfacing and USB communication.
- Sensors:
  - AD8232 ECG sensor for heart signal acquisition.
  - MAX30100 for heart rate (BPM) and SpO_2 measurement.
- Prototyping Tools: Breadboard, jumper wires, USB cable.
- Web Dashboard: Built using HTML, JavaScript, and index.html for visualization.

3.2 Block Diagram

The system architecture is designed around the integration of biomedical sensors with the ESP8266 microcontroller and a web-based dashboard for visualization. The AD8232 ECG sensor is responsible for capturing the electrical activity of the heart and sends the analog ECG signal to the ESP8266 through its analog input pin. At the same time, the MAX30100 pulse oximeter sensor measures the user's heart rate and blood oxygen saturation (SpO_2) levels. This sensor communicates with the ESP8266 using the I^2C protocol, where the SDA and SCL lines handle data exchange.

Once the ESP8266 collects the signals from both sensors, it processes and formats the readings into a structured, comma-separated data stream. This data is then transmitted in real-time to a computer via USB serial communication. The computer hosts a web dashboard (index.html), developed using HTML, CSS, JavaScript, and Chart.js, which receives and interprets the incoming data.

On the dashboard, the ECG signal is plotted as a continuous waveform, while the heart rate (in BPM) and SpO_2 percentage are displayed numerically. To enhance usability, the system includes threshold-based alerts: if the heart rate exceeds 120 BPM or drops below 50 BPM, or if SpO_2 falls below 90%, the dashboard immediately generates an on-screen warning. This ensures that abnormal health conditions are detected quickly and effectively.

Fig.3.1 illustrates the complete workflow: sensors acquire physiological signals → ESP8266 processes and transmits data → computer receives via USB → web dashboard visualizes readings and triggers alerts.

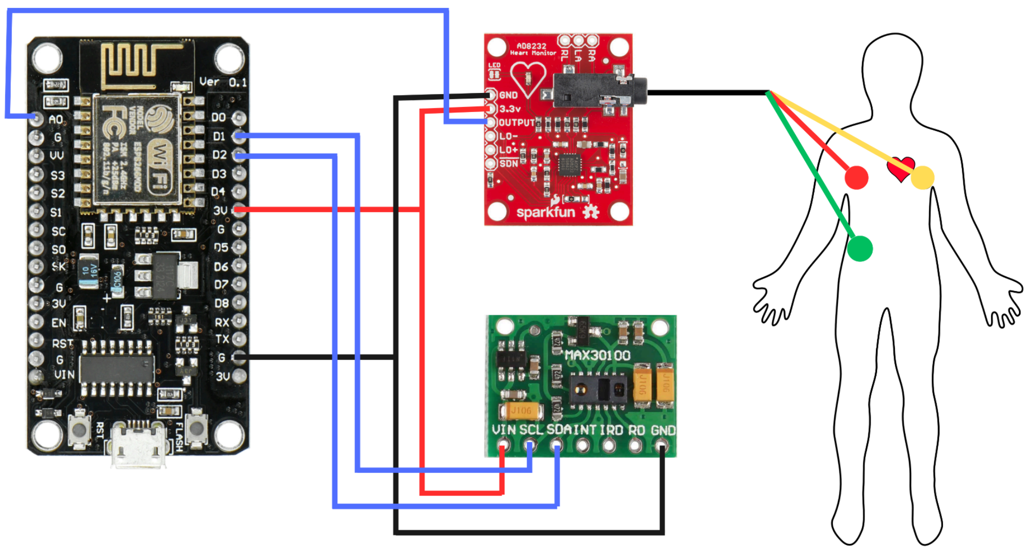

Figure 3.1: Overview of the Study's Methodological Workflow Block diagram.

4. Methodology

4.1 Hardware Design

The hardware design of the system is centered on the **ESP8266 microcontroller**, which acts as the main processing and communication unit. The **AD8232 ECG sensor** is connected to the analog input pin (A0) of the ESP8266 to capture the electrical activity of the heart. This sensor records small voltage changes from the body and converts them into signals that can be read by the microcontroller. For measuring blood oxygen saturation (SpO_2) and heart rate, the **MAX30100 pulse oximeter sensor** is used. It is interfaced with the ESP8266 through the **I²C communication protocol**, where the SDA pin is connected to **D2** and the SCL pin is connected to **D1** of the ESP8266. The entire system is powered through the **USB interface**, which not only supplies a stable 3.3V power source to the ESP8266 and sensors but also enables serial communication with the computer. This USB connection serves as the bridge for transmitting real-time health data to the web dashboard.

4.2 Software Design

The software design of the system is divided into three major parts: firmware programming, serial communication, and web-based visualization.

1. Firmware Programming (Arduino IDE): The ESP8266 was programmed using the Arduino IDE in C++ language. The firmware continuously reads raw data from the AD8232 sensor (ECG signals) and digital readings from the MAX30100 sensor (heart rate and SpO_2 values). These values are processed and formatted for transmission.

2. Serial Communication: To ensure efficient data transfer, the ESP8266 sends sensor readings to the computer via USB serial communication. The data is structured in a comma-separated format (HR, SpO_2, ECG) so that the web application can easily parse and identify individual parameters. This structured format minimizes errors and ensures synchronization between hardware and software.

3. Web Dashboard (Visualization and Alerts): On the computer side, a custom web dashboard is developed using HTML, CSS, and JavaScript. The visualization relies on the Chart.js library for plotting real-time ECG waveforms and for displaying numerical values of heart rate and SpO_2. In addition to visualization, the dashboard incorporates an alert mechanism, where threshold values are pre-defined. If abnormal readings are detected (e.g., HR > 120 BPM, HR < 50 BPM, or SpO_2 < 90%), on-screen alerts are triggered instantly, allowing users to take timely action.

Figure 4.1 shows the flowchart of the Real-Time Health Monitoring System. The system begins with powering on the device, which initializes all sensors and the ESP8266 microcontroller. Once the system is ready, the AD8232 ECG sensor and the MAX30100 heart rate and SpO_2 sensor are configured and checked for proper connection. The sensors then start acquiring real-time data, with the AD8232 capturing the ECG waveform and the MAX30100 measuring heart rate in BPM and blood oxygen levels. This raw data is processed and filtered to remove noise, ensuring accurate readings. The processed data is transmitted to a connected computer via USB, where it is displayed in real time. The software interface shows the ECG waveform, heart rate, and SpO_2 values, allowing continuous monitoring. Additionally, the system can include optional alerts if any abnormal values are detected. The process runs in a continuous loop, enabling ongoing, real-time health monitoring for the user.

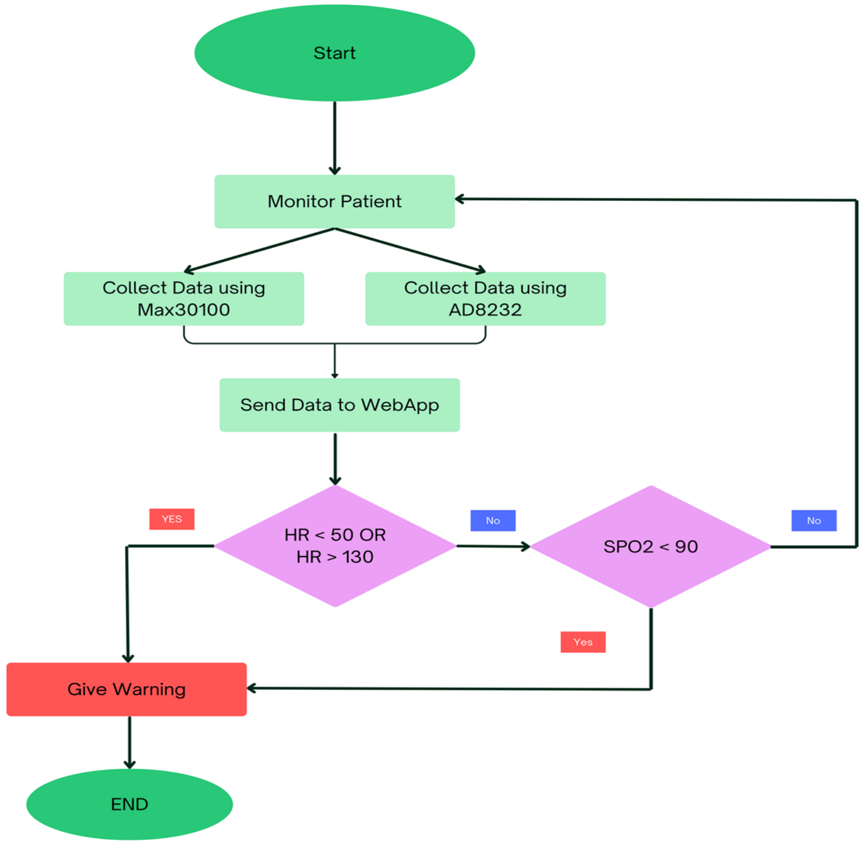

Figure 4.1: Flowchart of Real time health monitoring system

4.3 Communication Protocol

The communication between the ESP8266 and the computer is established through a custom-built web application. The ESP8266 transmits sensor readings in a structured format via USB serial, and the web application reads and interprets this incoming data stream. Unlike third-party platforms, this custom web app provides flexibility by enabling direct plotting of ECG waveforms and real-time display of heart rate and SpO_2 values without relying on external servers or message limits.

To ensure secure and reliable data transfer, the system leverages the Web Serial AP, which allows the browser to communicate directly with hardware devices through serial ports. However, the Web Serial API has certain limitations—it is only supported in secure contexts such as HTTPS connections or localhost environments, and primarily works on Chromium-based browsers like Google Chrome or Microsoft Edge. This design ensures both data security and compatibility while maintaining low latency for real-time health monitoring.

5. Implementation

5.1 Sensor Data Acquisition

The system utilizes two biomedical sensors to capture vital health parameters. The AD8232 ECG sensor is responsible for recording the electrical activity of the heart. It is connected to electrodes placed on the chest, which detect small voltage fluctuations generated during each heartbeat. These signals are then transmitted to the ESP8266 microcontroller for processing.

At the same time, the MAX30100 pulse oximeter sensor measures both heart rate (BPM) and blood oxygen saturation (SpO_2). The user places a fingertip on the sensor, where infrared and red LEDs pass light through the skin. Based on the absorption of light by oxygenated and deoxygenated hemoglobin, the sensor calculates the SpO_2 level while also deriving the pulse rate.

The outputs from both sensors are collected by the ESP8266, formatted, and then combined into a single data stream. This data is transmitted over USB serial communication to a connected computer, where it is processed and visualized in real time on the web dashboard.

5.2 Web Dashboard Features

The web dashboard serves as the primary user interface for visualizing and monitoring health data in real time. It was developed using HTML, CSS, JavaScript, and the Chart.js library, which allows continuous plotting of the ECG waveform as it is received from the ESP8266. This provides users with a clear, real-time graphical view of heart activity, similar to clinical monitoring systems.

In addition to the ECG signal, the dashboard displays numerical values of heart rate (BPM) and blood oxygen saturation (SpO_2). These readings are updated dynamically as new data is streamed, ensuring minimal latency between sensor measurement and visualization.

To enhance safety, the dashboard is equipped with a threshold-based alert system. If the heart rate exceeds 120 BPM (indicating tachycardia) or drops below 50 BPM (indicating bradycardia), the system immediately generates a high/low heart rate warning. Similarly, if the SpO_2 level falls below 90%, a low oxygen saturation alert is triggered. These warnings are displayed prominently on the dashboard in red font, ensuring that users can quickly recognize abnormal conditions and take appropriate action.

5.3 Alert Mechanism

- Warning messages displayed in **red font** when thresholds are crossed.
- Users are notified instantly of abnormal health conditions.

6. Results

The Real-Time Health Monitoring System successfully demonstrated its capability to capture and display vital health parameters. The ECG signals were plotted continuously in real time, providing an accurate visual representation of the heart's electrical activity. Simultaneously, heart rate (BPM) and SpO_2 values were displayed with minimal latency, ensuring that the dashboard reflected the most current physiological measurements.

The threshold-based alert system operated reliably, triggering on-screen warnings whenever abnormal conditions were detected, such as high or low heart rate and low oxygen saturation. Additionally, the USB-based serial streaming proved stable and accurate, enabling consistent data transfer from the ESP8266 to the web dashboard without noticeable delays or interruptions. Overall, the system provided a responsive and reliable platform for continuous health monitoring outside clinical settings.

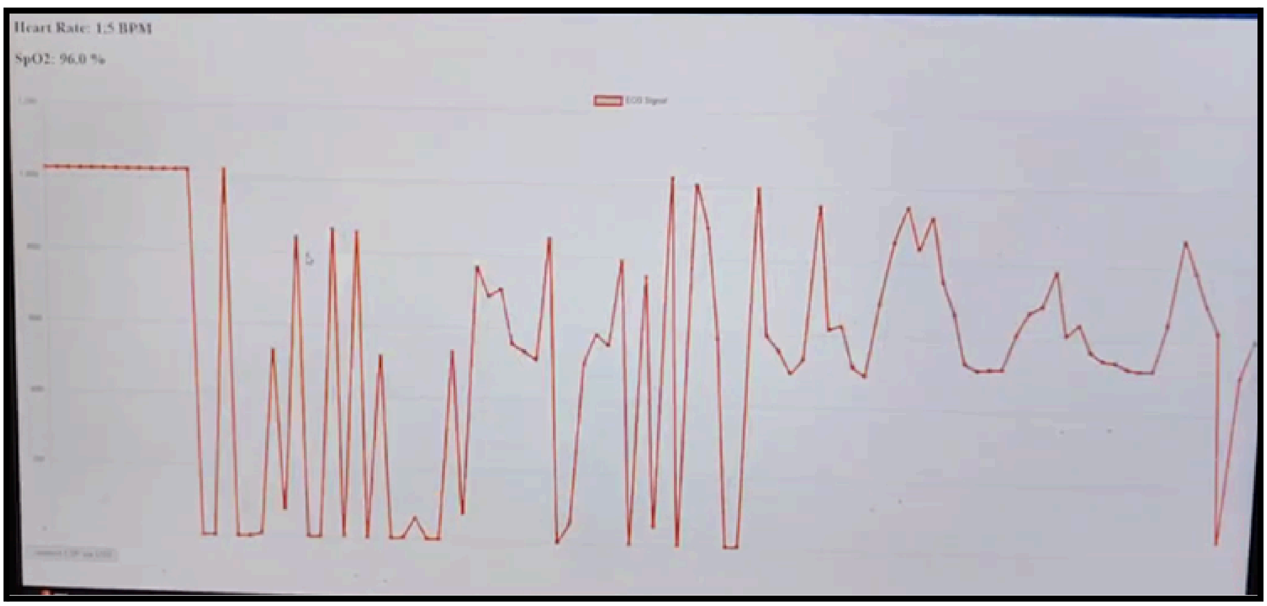

Figure 6.1: User Dashboard Result of Real time health monitoring system (user1)

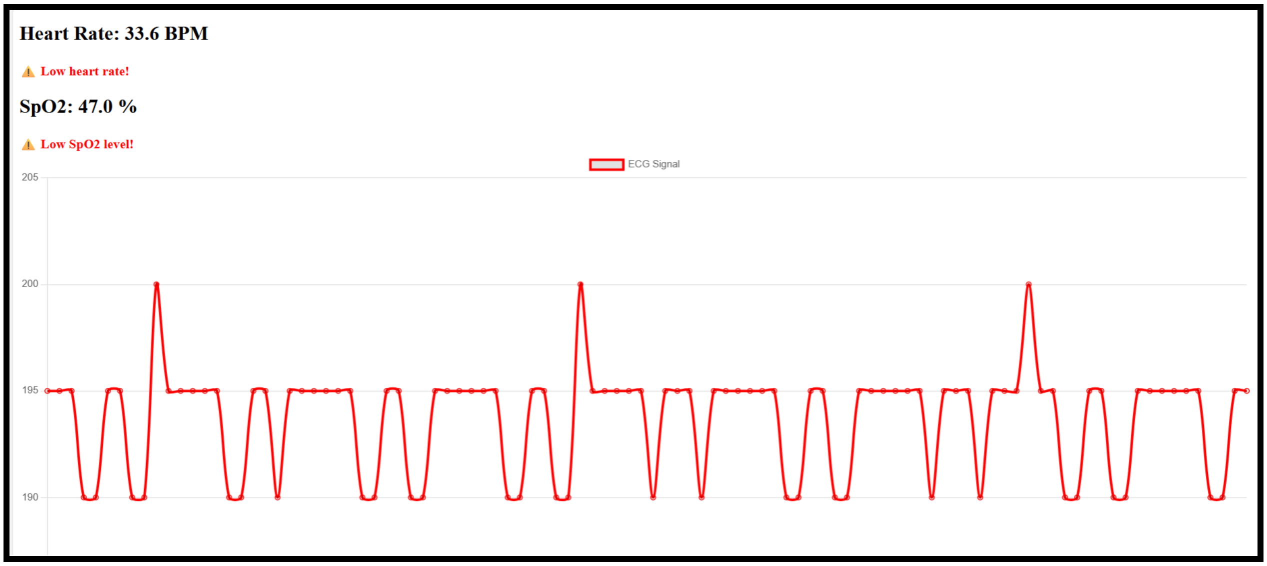

Figure 6.1: User Dashboard Result of Real time health monitoring system (user2)

Figures 6.1 and 6.2 display the real-time output of the web dashboard for two different users, showing the ECG waveform, heart rate (BPM), and blood oxygen saturation (SpO_2). The ECG signals are plotted continuously using index.html which is a custom Webapp, with peaks and troughs reflecting the electrical activity of the heart, demonstrating stable and accurate signal acquisition. Heart rate values update dynamically in real time as each new reading is received from the MAX30100 sensor, while SpO_2 levels are displayed alongside, highlighting any abnormal readings with red alerts if oxygen saturation drops below 90%. Together, these figures illustrate that the system can simultaneously monitor multiple vital signs, provide immediate visual feedback, and reliably trigger alerts for abnormal conditions, confirming the effectiveness of the Real-Time Health Monitoring System.

Compare with Honor Band 5

To better understand the advantages of our Real-Time Health Monitoring System, it is useful to compare it with a popular commercial wearable device, the **Honor Band 5**. This comparison highlights differences in functionality, data access, and monitoring accuracy between a dedicated embedded system and a consumer fitness tracker. While the Honor Band 5 provides convenient daily activity tracking, it lacks detailed ECG monitoring and customizable real-time alerts. In contrast, our system offers precise measurement of ECG, heart rate, and SpO_2, along with a fully customizable web dashboard for real-time visualization and threshold-based alerts.

Feature	Real-Time Health Monitoring System	Honor Band 5
ECG Monitoring	Yes, uses AD8232 sensor with chest electrodes and real-time waveform plotting	No, only heart rate measurement from wrist [6]
Heart Rate (BPM)	Yes, measured via MAX30100 and displayed in real time	Yes, measured via optical wrist sensor [6]
SpO_2 Monitoring	Yes, measured via MAX30100 and displayed in real time	Yes, measured via optical wrist sensor [6]
Data Visualization	Custom web dashboard using HTML, JavaScript, and index.html	Proprietary mobile app with limited visualization [6].
Alerts	Threshold-based alerts for abnormal HR and SpO_2 displayed on the dashboard	Some alerts available, but limited customization [7].
Data Access	Full access to raw sensor data for analysis	Limited access; raw ECG data not available [7].
Power Source	USB-powered (stable and continuous)	Battery-powered (portable but limited by battery life) [7].
Portability	Less portable, requires USB connection and external sensors	Highly portable, wrist-worn device [7].
Customization	Fully customizable dashboard and data processing	Limited to vendor-provided app features [7].

Use Case	Suitable for research, personal health tracking, and clinical-like monitoring	Primarily for daily fitness tracking and general health monitoring [6].

While both systems measure heart rate and SpO_2, the Real-Time Health Monitoring System provides additional features like ECG waveform visualization, custom alerts, and access to raw data. This makes the results more detailed, accurate, and clinically useful compared to the Honor Band 5.

7. Challenges and Solutions

Challenge	Solution
ESP8266 Blynk server had a 30k free message limit and lacked waveform graph support	Designed a custom web application to visualize ECG waveforms and display results without relying on Blynk limitations.
MAX30100 sensor malfunctioned when connected to ESP8266	Tried 4.7kΩ pull-up resistors, but issue persisted; finally replaced with a new sensor, which worked reliably.
ECG signal noise due to movement	Recommended stable posture and use of filtering techniques.
Limited Web Serial API browser support	Implemented testing on Chrome/Chromium-based browsers only.
Power fluctuations with sensors	Powered both sensors consistently with 3.3V from ESP8266.

8. Future Scope

The Real-Time Health Monitoring System can be further enhanced with Bluetooth and Wi-Fi connectivity, allowing wireless data transmission to smartphones and other devices. Integration with cloud platforms enables remote monitoring, so health data can be accessed from anywhere.

Additionally, the development of a mobile application increases portability, making it convenient for users to track their health on the go. The system can also incorporate additional sensors, such as temperature and blood pressure sensors, to provide more comprehensive health monitoring. Furthermore, AI-based anomaly detection can be implemented to identify unusual patterns in the data, enabling predictive healthcare and timely interventions.

9. Conclusion

This project successfully developed a low-cost, real-time health monitoring system using embedded systems and IoT principles. By integrating the AD8232 ECG sensor and the MAX30100 heart rate and SpO_2 sensor with the ESP8266 microcontroller, the system is able to capture and process accurate physiological data in real time. The measurements are visualized on a user-friendly web dashboard, providing immediate insight into the user's heart activity and oxygen levels. The system demonstrates reliable performance in monitoring key health parameters, making it suitable for personal health tracking. With the addition of wireless connectivity via Bluetooth and Wi-Fi, as well as potential integration with mobile applications and cloud platforms, the system can support remote and continuous healthcare monitoring. Furthermore, the incorporation of additional sensors and AI-based anomaly detection could enhance its predictive capabilities, enabling early detection of health issues. Overall, this project illustrates the practical application of Embedded systems and IoT in healthcare, offering an accessible and scalable solution for real-time health monitoring and telemedicine.

10. References

[1] Arduino, "Arduino IDE," Available: https://www.arduino.cc/ [Accessed: Aug. 8, 2025].

[2] Espressif Systems, "ESP8266 Technical Reference," Available: https://www.espressif.com/ [Accessed: Aug. 8, 2025].

[3] Analog Devices, "AD8232 Heart Rate Monitor Front End Datasheet," 2016.

[4] Maxim Integrated, "MAX30100 Pulse Oximeter and Heart Rate Sensor Datasheet," 2015.

[5] MDN Web Docs, "Web Serial API," Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Serial_API [Accessed: Aug. 8, 2025].

[6] honor, "HONOR Band 5 – Practical, Functional, and Healthful | HONOR Official Site (Global)," *Honor.com*, Mar. 04, 2020. https://www.honor.com/global/blog/honor-band-5-practical-functional-and-healthful/?utm_source=chatgpt.com.

[7] G. James, “Article headline,” *Accompanying Technology*, Mar. 02, 2022. https://www.accompanyingtechnology.com/review-of-the-honor-band-5-an-excellent-health-monitor-and-fitness-tracker?utm_source=chatgpt.com