Sri Lanka Institute of Information Technology

IT1040 - Fundamentals of Computing

Year 1, Semester 1- 2024

Military Soldier Health Monitoring System

Progress Report

P15

IT Number	Name
IT24104076	Jayasinha W. M. S. S. N.
IT24104102	Amarakeerthi H. K. K. U.
IT24103832	Chamodika J. W. C.
IT24103910	Perera W. G. D.
IT24104301	Karannagoda K. V. R.
IT24103550	Weerasinghe A. E. K. P.

Table of Contents

Table of Figures

1.0 Introduction

The safety and well-being of soldiers during military operations is a critical concern, yet traditional systems often fall short in providing real-time monitoring of their health and location. Soldiers face significant risks, including injuries, infections, and environmental hazards, that can lead to severe consequences without timely medical intervention. To address these challenges, a Health Monitoring and Location Tracking System has been developed. This system utilizes a wearable device equipped with sensors to continuously track vital health parameters such as heart rate and body temperature, along with real-time location updates via GPS. Data is transmitted to the military control center using advanced microcontroller technology, enabling instant situational awareness.

By providing real-time insights into soldiers' conditions, this system enhances safety, improves tactical decision-making, and increases mission resilience. Commanders can optimize strategies based on the physical readiness and location of personnel, while medical teams can respond promptly to emergencies. This innovation represents a significant leap in military health and safety protocols, ensuring soldiers are better protected and supported, ultimately strengthening operational efficiency and mission outcomes.

2.0 Methodology

2.1 Design Phases

1. **Setup the Components**
    - **ESP-8266:** Central processor for data management and transmission.
    - **Pulse Sensor:** Monitors heart rate.
    - **Temperature Sensor:** Tracks body temperature.

- **GPS Module:** Provides real-time location tracking.
- **OLED Display:** Displays health and location data for soldiers.

2. **Designing the Wearable**

   Develop a durable, waterproof, and ergonomic wristband suitable for military use.

3. **Assembling Components**

   Securely attach all sensors and devices to the wristband.

4. **Wiring the System**

   Connect sensors, GPS, and OLED to the ESP-8266 with proper insulation and stability.

5. **Coding**

   Develop software for:

   - **Data Acquisition:** Collect sensor inputs.
   - **Data Processing:** Filter and analyze sensor data.
   - **Communication:** Transmit data via Wi-Fi/Bluetooth.
   - **Display Logic:** Show data on the OLED.

6. **Real-Time Location Tracking**

   Use the GPS module to continuously update the soldier's location.

7. **Health Monitoring**

   Measure pulse and body temperature regularly to detect abnormalities.

8. **Data Collection and Integration**

   Combine health and location data for comprehensive soldier monitoring.

9. **System Accuracy Verification**

   Compare sensor outputs with medical-grade devices to ensure reliability.

10. **Data Communication**

    Implement protocols to transmit real-time data to the control room for live monitoring and emergency response.

2.2 Hardware Requirements

1. **ESP-32 Microcontroller:**
   - Handles data processing and wireless communication with built-in Wi-Fi and Bluetooth.
2. **Pulse Sensor:**
   - Detects heart rate via blood flow analysis.
3. **LM35 Temperature Sensor:**
   - Measures body temperature accurately.
4. **GPS Module:**
   - Provides real-time position tracking using data transmission pins (Tx and Rx).

3.0 System Diagram

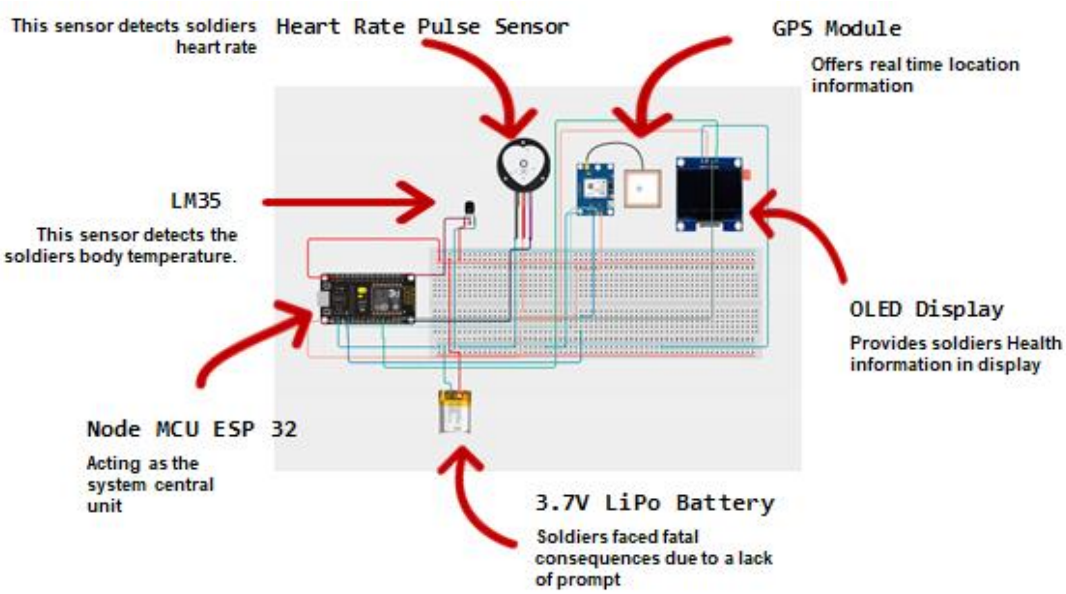

Figure 1: System Diagram

4.0 Current Progress

1. In the current progress, we successfully connected the pulse rate sensor to the ESP32 microcontroller and configured it to measure heartbeats per minute (BPM). Using the Pulse Sensor Playground library, we were able to initialize the sensor, read pulse data, and display the output on the serial monitor. The system accurately detected pulse rates and displayed consistent values such as 73 BPM, 76 BPM, and 80 BPM, indicating proper sensor functionality. This milestone demonstrates that the pulse sensor integration works effectively, providing real-time heart rate data as intended.

Figure 2: Pulse Rate Sensor Output

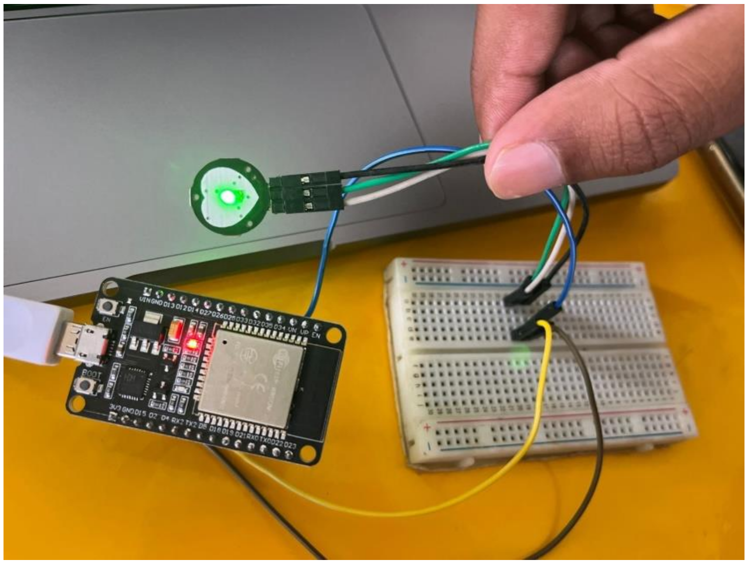

Figure 3: Pulse Rate Sensor connected to the ESP32

2. In addition to the pulse sensor, we successfully connected the body temperature sensor (LM35) to the ESP32 microcontroller to measure temperature values. Using analog input readings, we converted the sensor's voltage output into temperature readings in degrees Celsius and displayed the results on the serial monitor. The output showed values such as 17.49°C, 20.23°C, and 27.08°C, indicating the sensor's ability to capture temperature variations accurately. This confirms the proper functionality of the temperature sensor, marking another significant step in the project’s progress.

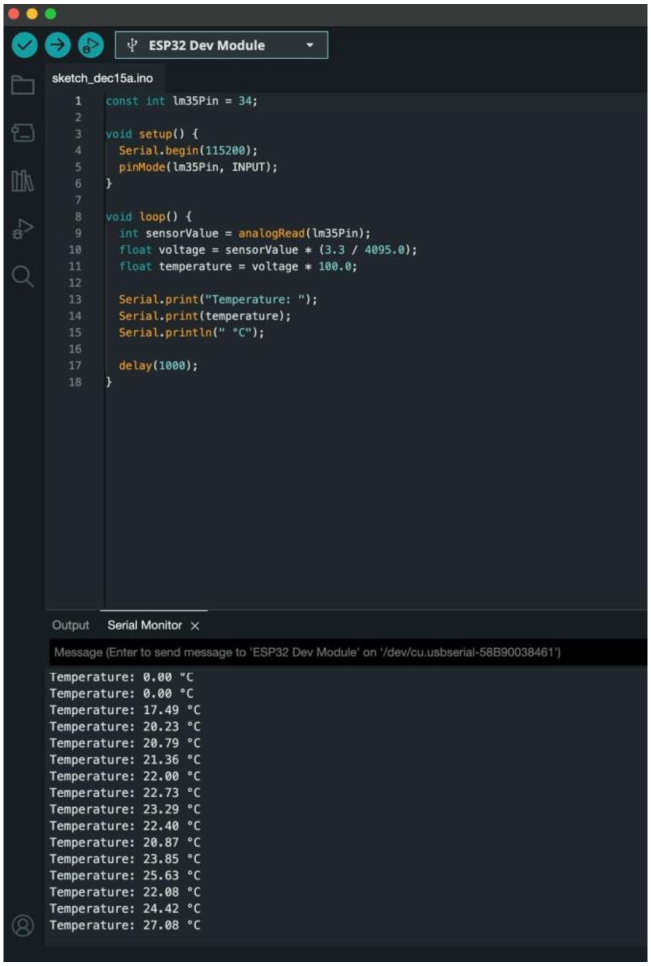

Figure 4: Body Temperature Sensor Output

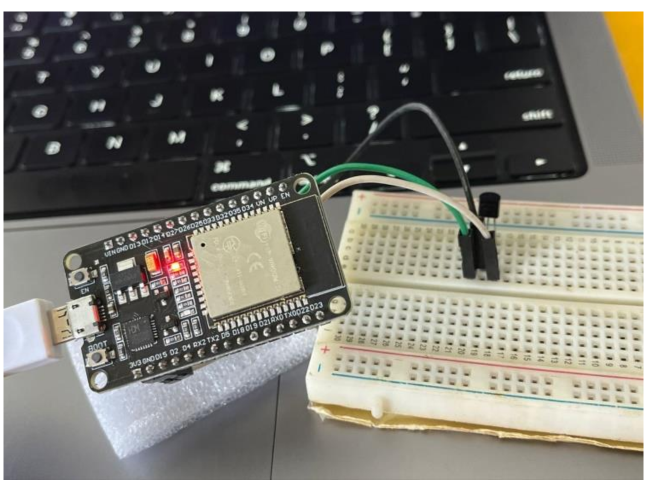

Figure 5: Body Temperature Sensor connected to the ESP32

5.0 Challenges Encountered

While connecting the sensors and obtaining outputs, we did not face any significant issues, as the components worked as expected. However, we are still in the process of progressing toward refining the product further. One of the key challenges is ensuring that the product meets industry standards in terms of performance, reliability, and design.

Additionally, we are focused on delivering a high-quality final output while keeping the improvements within an affordable budget. Balancing quality with cost-efficiency remains a critical task, as we aim to make the product accessible and practical for users without compromising its functionality.

6.0 Deviation from Initial Proposal

1. We replaced the ESP-8266 microcontroller with the ESP-32 microcontroller. This change was made to take advantage of the ESP-32's enhanced features, such as dual-core processing, higher performance, and additional GPIO pins, which provided better compatibility with our sensors and modules. The ESP-32 also supports both Wi-Fi and Bluetooth, offering greater flexibility for future upgrades and improved system efficiency.

2. We implemented a website using CSS, WordPress, Java, Moodle, and XAMPP, which includes comprehensive details about the product. Initially, we had planned to use a simpler platform, but as the project evolved, we realized the need for more robust tools to accommodate additional features and enhance functionality. These changes allowed us to create a more user-friendly and detailed interface.

Website link: https://nimna-jayasinha.github.io/demo/

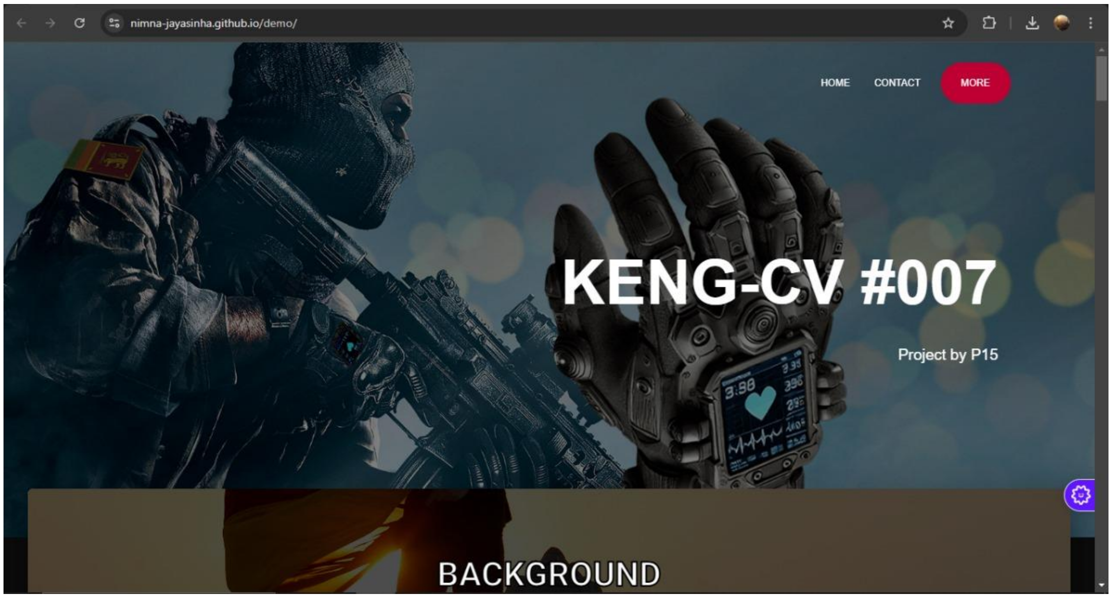

Figure 6: Screenshot of the Website

3. Additionally, we plan to integrate a solar power system to minimize battery usage, addressing the challenge of carrying and replacing batteries frequently. By incorporating solar power, we aim to make the product more convenient, sustainable, and efficient for users in the long run.

Figure 7: Solar Power System

7.0 Remaining Tasks

1. In the remaining tasks, we plan to integrate a GPS module into the system, which will provide real-time location tracking. This feature will enhance the overall functionality by enabling users to access precise and up-to-date location information. The integration process involves hardware installation, coding, and testing to ensure seamless performance with the existing system. This addition is crucial for improving the product's reliability and user experience.

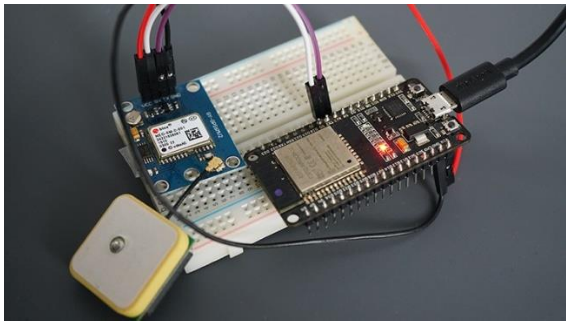

Figure 8: GPS Module

2. We will also be adding a 7V LiPo battery to the system to ensure efficient and reliable power supply. This battery was chosen for its lightweight design, high energy density, and compatibility with the existing components. Integrating the 7V LiPo battery will not only improve the system's portability but also extend its operational time, making it more user-friendly. The installation and testing of the battery will be a critical part of our remaining tasks to ensure optimal performance.

Figure 9: 7V LiPo Battery

3. We plan to integrate a proximity sensor into the system. This sensor will enable the detection of nearby objects or obstacles, enhancing the product's functionality for applications requiring spatial awareness. The integration process will involve connecting the sensor to the ESP32, configuring the necessary code, and testing its accuracy and responsiveness. This addition will further improve the system's versatility and user experience.