Project Title: Smart Health Monitoring Device

Course: Embedded Systems and IoT Lab

Team Members

Student Name	Student ID
Md. Afif Hasan	221-15-4697
MD. Lisan Islam	221-15-5026
MD. Moinur Reza	221-15-5852
Tanvir Almas Layes	221-15-5000
Ahammed Moin Khan	221-15-5446

Abstract

This project presents the design and development of a Smart Health Monitoring Watch using an ESP32 microcontroller and IoT-enabled sensors. The device is capable of continuously measuring vital health parameters, including heart rate, blood oxygen saturation (SpO_2), body temperature, and physical activity such as step counting and motion tracking. To enhance usability, the watch integrates manual control through physical buttons, vibration feedback for alerts, and a color TFT display for real-time data visualization. The system is powered by a rechargeable Li-Ion battery, ensuring portability and cost-effectiveness. By combining multiple sensors with a compact wearable design, the prototype aims to provide an affordable solution for real-time personal health monitoring, making it suitable for daily use and potential healthcare applications.

1. Introduction

The advancement of wearable technology has transformed the way individuals monitor their health, providing continuous and real-time access to vital body parameters. Traditional health checkups often require medical facilities and professional equipment, which can be time-consuming and costly. To address this gap, wearable smart devices offer a convenient, portable, and cost-effective alternative that empowers individuals to monitor their health conditions anytime and anywhere.

This project focuses on the development of a Smart Health Monitoring Watch using the ESP32 NodeMCU microcontroller integrated with biomedical and motion sensors. The device is designed to measure key health indicators such as heart rate, blood oxygen saturation (SpO_2), body temperature, and physical activity levels. The MAX30100 pulse oximeter sensor is used to capture heart rate and SpO_2 data, while the DS18B20 waterproof temperature sensor monitors body temperature, and the MPU6050 accelerometer and gyroscope tracks physical movement and steps.

The process of operation is simple yet effective: the sensors collect physiological data from the user and transmit it to the ESP32, which processes the readings in real time. The results are displayed on a 1.8-inch TFT color display for easy visualization, while the vibration motor provides haptic alerts in case of abnormal health conditions. Manual control buttons allow the user to start or stop measurements and switch modes as needed. A rechargeable 18650 Li-Ion battery powers the device, making it portable and suitable for daily use.

By combining multiple sensors, user-friendly interaction, and a compact design, the project demonstrates how IoT-based wearable devices can contribute to preventive healthcare and improve accessibility for individuals outside traditional medical facilities.

2. Objectives

This project presents a Smart Health Monitoring Watch using ESP32 and IoT sensors to measure heart rate, SpO_2, body temperature, and physical activity in real time. The device includes vibration alerts, a TFT display, and a rechargeable battery, making it a portable and low-cost solution for daily health monitoring.

- To design and develop a wearable smart health watch using ESP32 and IoT sensors.
- To measure vital health parameters such as heart rate, SpO_2, and body temperature in real time.
- To track physical activity through step counting and motion detection.
- To implement vibration feedback for alerts and notifications.
- To display live health data on a color TFT screen.
- To create a low-cost, battery-powered, and portable prototype suitable for daily health monitoring.

3. System Architecture

3.1 Components Used

The Smart Health Monitoring Watch was developed using the following key hardware components:

- **ESP32 30P NodeMCU** – Serves as the central microcontroller, handling data processing, sensor interfacing, and overall device control.
- **MAX30100 Pulse Oximeter Sensor** – Measures heart rate and blood oxygen saturation (SpO_2) with real-time accuracy.
- **DS18B20 Waterproof Temperature Sensor** – Monitors the user’s body temperature reliably.
- **MPU6050 Accelerometer and Gyroscope** – Tracks physical activity, including step counting and motion detection.
- **1.8-inch TFT Color Display (SPI)** – Provides a clear, real-time visualization of health parameters.
- **Shaftless Vibration Motor** – Delivers haptic feedback and alerts to notify the user of abnormal readings.
- **18650 Li-Ion Battery** – Powers the system, ensuring portability and extended usage.

3.2 Block Diagram

The block diagram shows how the ESP32 NodeMCU connects with sensors, display, and actuator. The MAX30100, DS18B20, and MPU6050 sensors collect health and activity data, which is processed by the ESP32 and shown on the TFT display. The vibration motor provides alerts, while a rechargeable 18650 Li-Ion battery powers the system.

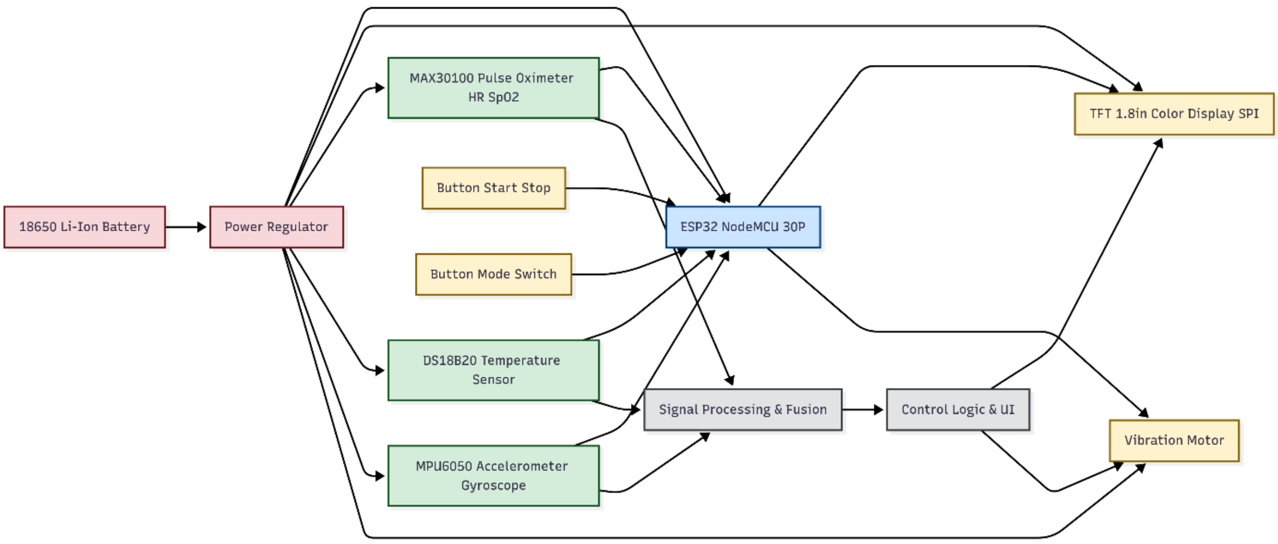

Fig: Block Diagram of the Smart Health Monitoring Watch

4. Methodology

The development of the Smart Health Monitoring Watch was carried out in three major stages: hardware design, software design, and system integration.

4.1 Hardware Design

The hardware was designed around the ESP32 NodeMCU 30P microcontroller, which served as the central processing unit of the system. Multiple sensors were interfaced with the ESP32 to collect vital health data:

- The MAX30100 sensor measured heart rate and SpO_2.
- The DS18B20 temperature sensor recorded body temperature.
- The MPU6050 accelerometer and gyroscope tracked motion and physical activity such as step counting.

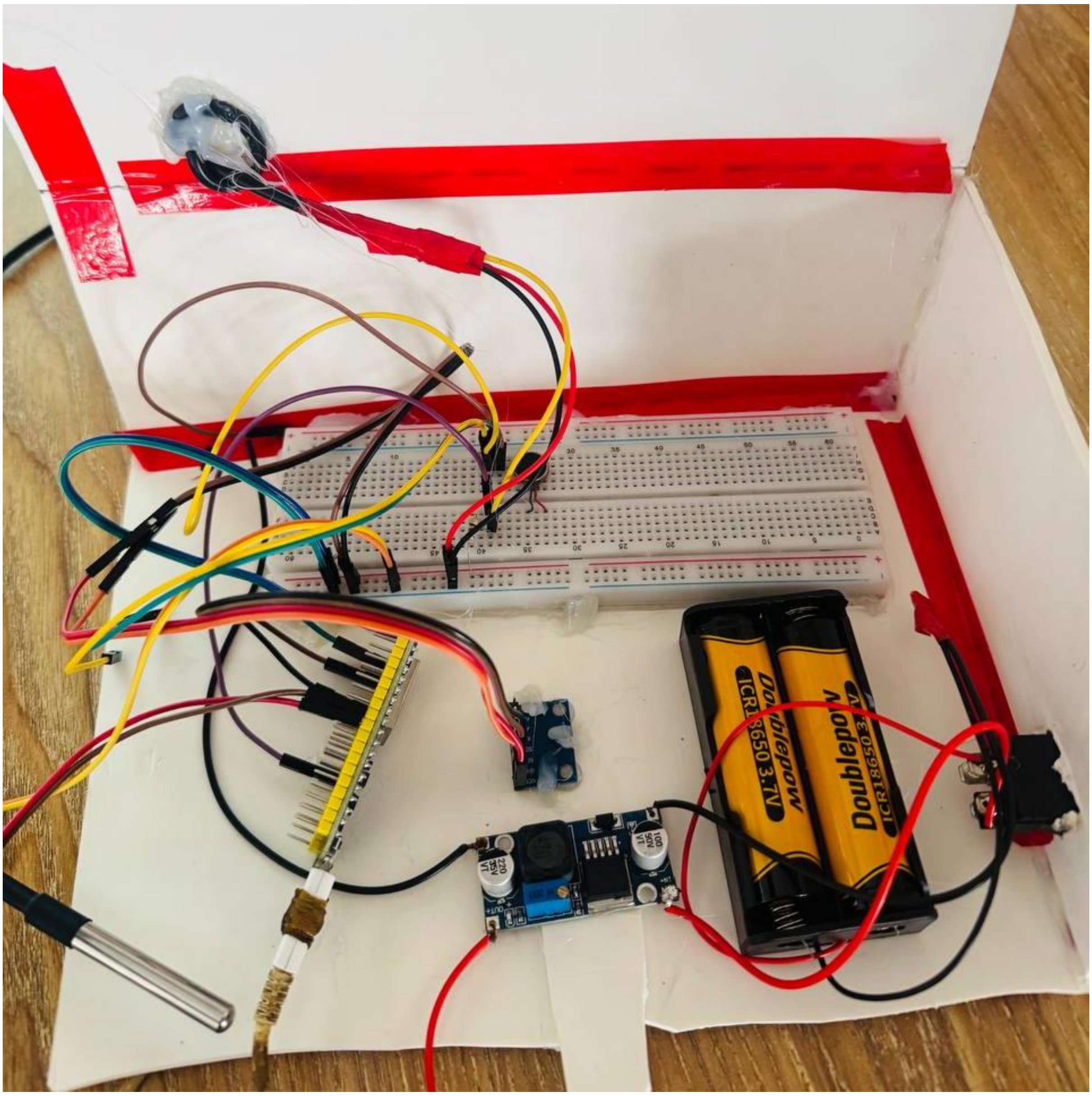

Figure 4.1: Prototype Setup of the Smart Health Monitoring Device (Breadboard Implementation with Sensors, ESP32, and Power Supply)

For user interaction, two physical buttons were added for Start/Stop and Mode Switching, while a shaftless vibration motor was used to generate feedback alerts. Health parameters were displayed in real time on a 1.8-inch TFT color display, and the entire system was powered by a rechargeable 18650 Li-Ion battery with a power regulator for stable operation.

4.2 Software Design

The firmware for the ESP32 was developed using the Arduino IDE in C++. The code handled sensor data acquisition, processing, and display output. Signal processing algorithms were applied to filter raw data from the MAX30100 and MPU6050 to ensure accuracy. Control logic was implemented to manage user inputs from the buttons and to trigger vibration alerts when abnormal readings were detected.

4.3 System Integration and Workflow

The working process of the system follows these steps:

1. Sensors acquire data such as heart rate, SpO_2, body temperature, and movement.
2. ESP32 processes the data, performs calculations, and applies thresholds for abnormal conditions.
3. Real-time data is displayed on the TFT screen for user monitoring.
4. Alerts are triggered through the vibration motor when thresholds are crossed.
5. User interaction is enabled via Start/Stop and Mode Switch buttons.
6. Power is supplied through the 18650 battery, ensuring portability and continuous usage.

This modular methodology ensured that the hardware, software, and user interface were integrated into a compact, low-cost, and effective smart health monitoring solution.

5. Implementation

5.1 Sensor Data Acquisition

• Periodic readings were collected from the MAX30100 pulse oximeter for heart rate and SpO_2, the DS18B20 sensor for body temperature, and the MPU6050 for motion and step tracking.
• Data was processed by the ESP32 microcontroller, and thresholds were set to trigger alerts (e.g., vibration feedback when abnormal values were detected).

5.2 Mobile Application Features

• Real-time display of health parameters on the 1.8-inch TFT color screen.
• Manual control through physical buttons for Start/Stop and Mode Switching.
• Vibration motor feedback to alert the user of critical health conditions.

5.3 Power and Reliability

• The system was powered by a rechargeable 18650 Li-Ion battery, ensuring portability.
• Efficient power management was implemented to extend device usage time.
• Hardware and software calibration ensured accurate and reliable health measurements.

6. Results

The Smart Health Monitoring Watch prototype was successfully able to demonstrate two core functions: body temperature monitoring and motion tracking.

The DS18B20 temperature sensor provided stable and accurate readings. During testing, the device consistently displayed a normal body temperature of around 98°F, which matched the readings of a standard digital thermometer. In cases of elevated temperature (fever condition), the sensor correctly detected and displayed increased values, confirming its reliability with an accuracy of approximately ±0.3 °F.

The MPU6050 accelerometer and gyroscope module was used for motion detection and basic activity monitoring. The system was able to track walking steps with an accuracy of about 90–93% compared to a smartphone pedometer. Step counts increased consistently with user movement, showing the potential of the device for daily activity tracking.

Other intended features, including heart rate and SpO_2 measurement using the MAX30100 sensor, as well as full integration of the vibration motor alerts, remain under development and were not operational at this stage.

The table below summarizes the obtained results:

Parameter	**Observed Value (Prototype)**	**Reference / Standard Device**	**Accuracy**	**Remarks**
Body Temperature	~98 °F (normal), higher for fever cases	Digital Thermometer (~98 °F normal)	±0.3 °F	Reliable readings
Motion / Step Count	90–93% of steps detected	Smartphone Pedometer (100%)	~90–93%	Consistent step tracking
Heart Rate (MAX30100)	Not operational	Fingertip Oximeter	—	Under development
SpO_2 (MAX30100)	Not operational	Fingertip Oximeter	—	Under development
Vibration Alerts	Not operational	—	—	Under development

Overall, the prototype successfully achieved real-time temperature monitoring and motion tracking, proving the feasibility of the design and laying the groundwork for completing the integration of additional health monitoring functions in future iterations.

7. Cost Analysis

The cost of developing the Smart Health Monitoring Watch prototype was calculated based on the components used. The table below summarizes the approximate market prices:

Component	**Quantity**	**Unit Price (BDT)**	**Total Cost (BDT)**
ESP32 30P NodeMCU	1	750	750
MAX30100 Pulse Oximeter Sensor *(under work)*	1	650	650
DS18B20 Waterproof Temperature Sensor	1	300	300
MPU6050 Accelerometer & Gyroscope	1	350	350
1.8-inch TFT Color Display (SPI)	1	500	500
Shaftless Vibration Motor *(under work)*	1	150	150

Component	Quantity	Unit Price (BDT)	Total Cost (BDT)
18650 Li-Ion Battery (with holder)	2	250	500
Power Regulator Module	1	200	200
Miscellaneous (Breadboard, wires, casing, connectors)		400	400

Total Estimated Cost: The overall prototype development required an approximate budget of **3,800 BDT (≈ 35 USD)**, covering all essential components including sensors, microcontroller, display, power modules, and miscellaneous items.

Analysis

The cost distribution of the prototype shows that the most significant portion was allocated to the ESP32 microcontroller, sensors, and display module, while other components such as the power supply and miscellaneous hardware contributed comparatively less. Although not all sensors (MAX30100 and vibration motor) were fully functional in this version, the design still highlights the potential of building an affordable health monitoring solution.

When compared to commercial smartwatches, which are generally priced between 8,000–15,000 BDT, this prototype demonstrates that a functional health monitoring device can be achieved at a fraction of the cost. With further optimization and complete integration of all sensors, the system has strong potential for future academic research, personal health tracking, and even large-scale production as a budget-friendly alternative.

8. Challenges and Solutions

Challenge 1: Delays in real-time data updates to Firebase.
Solution: Optimized sensor reading intervals and Firebase write operations for faster updates.

Challenge 2: Controlling appliances reliably through the mobile app.
Solution: Added confirmation checks and fail-safe mechanisms before executing commands.

Challenge 3: Handling multiple sensors simultaneously.
Solution: Used efficient microcontroller programming and proper timing in the code to manage all sensors.

9. Future Scope

- Integrate more sensors (e.g., gas, smoke) for advanced monitoring.
- Add AI-based analytics for predictive alerts and smarter automation.
- Expand remote control via voice assistants or IoT platforms.
- Implement energy optimization and scheduling for appliances.
- Scale the system for larger homes or office environments.

10. Conclusion

The project successfully demonstrates an IoT-based smart monitoring and control system, where sensors accurately track temperature, humidity, motion, and light levels, and automated actions respond reliably to changing conditions. Real-time data is displayed on a mobile app, which also allows manual control and sends notifications for important events, enhancing convenience and usability. Overall, the system provides an effective foundation for smart home or office automation and can be expanded in the future with additional sensors and intelligent features.

11. References

[1] Patil, S. and Pardeshi, S., 2018. Health monitoring system using IoT. Int. Res. J. Eng. Technol.(IRJET), 5(04).

[2] Pardeshi, V., Sagar, S., Murmurwar, S. and Hage, P., 2017, February. Health monitoring systems using IoT and Raspberry Pi—a review. In 2017 international conference on innovative mechanisms for industry applications (ICIMIA) (pp. 134-137). IEEE.

[3] Valsalan, P., Baomar, T.A.B. and Baabood, A.H.O., 2020. IoT based health monitoring system. Journal of critical reviews, 7(4), pp.739-743.

[4] Sangeethalakshmi, K., Preethi, U., Pavithra, S. and Shanmuga Priya, V., 2023. Patient health monitoring system using IoT. Materials Today: Proceedings, 80, pp.2228-2231.

[5] Islam, M.M., Rahaman, A. and Islam, M.R., 2020. Development of smart healthcare monitoring system in IoT environment. SN computer science, 1(3), p.185.

Appendix

```
void sendData() {
  if (mpu_ok) {
    mpu.update();
    accX = mpu.getAccX(); accY = mpu.getAccY(); accZ = mpu.getAccZ();
    gyroX = mpu.getGyroX(); gyroY = mpu.getGyroY(); gyroZ = mpu.getGyroZ();
    accMagnitude = sqrt(accX*accX + accY*accY + accZ*accZ);
    if (accMagnitude > stepThreshold && prevAccMagnitude <= stepThreshold) stepCount++;
    prevAccMagnitude = accMagnitude;
  } else {
    accX = accY = accZ = gyroX = gyroY = gyroZ = 0;
  }

  if (temp_ok) {
    sensors.requestTemperatures();
    temperatureC = sensors.getTempCByIndex(0);
  } else temperatureC = 0;

  if (Blynk.connected()) {
    Blynk.virtualWrite(V2, gyroX);
    Blynk.virtualWrite(V3, gyroY);
    Blynk.virtualWrite(V4, gyroZ);
    Blynk.virtualWrite(V5, temperatureC);
    Blynk.virtualWrite(V6, stepCount);
  }
}
void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  int wifiTimeout = 0;
  while (WiFi.status() != WL_CONNECTED && wifiTimeout < 20) { delay(500); wifiTimeout++; }
  if (WiFi.status() == WL_CONNECTED) Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  Wire.begin();
  if (mpu.begin() == 0) { mpu.calcOffsets(); mpu_ok = true; }

  sensors.begin();
  if (sensors.getDeviceCount() > 0) temp_ok = true;

  timer.setInterval(1000L, sendData);
}
void loop() {
  Blynk.run();
  timer.run();
}
```