IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15523

Smart Battlefield Helmet with Soldier Monitoring System

Dr. Pramod Sharma[1], Aryansh Gupta[2], Divya Gupta[3], Jayanth Bharadwaj[4], Sagar Mishra[5]

Dean, Department of ECE, R.B.S Engineering Technical Campus, Bichpuri, Agra [1]

B.Tech. Final year students, Department of ECE, R.B.S Engineering Technical Campus, Bichpuri, Agra [2,3,4,5]

Abstract: This paper presents a Smart Battlefield Helmet with Soldier Monitoring System. The aim of the project is to build a low-cost, embedded system that can continuously monitor a soldier's health and surroundings and transmit the data wirelessly to a base station. The helmet uses an Arduino Nano as the main microcontroller, paired with an ESP32 for communication. Sensors include the MAX30102 for heart rate and SpO_2 monitoring, DS18B20 for body temperature, an MQ gas sensor for detecting toxic or flammable gases, and an MPU6050 IMU for fall detection. A NEO-6M GPS module provides real-time location tracking, and a LoRa SX1278 radio module transmits all data to a base station dashboard several kilometres away. The circuit runs on a 5 V regulated supply using an LM7805 regulator powered from a 9 V DC adapter.

Keywords: Helmet, Soldier, Monitoring, Temperature, Tracking, Battlefield

I. INTRODUCTION

In modern warfare and border security operations, it is important for commanders to have real-time information about the health and location of soldiers in the field. However, existing systems such as voice radios and manual GPS reporting have significant limitations. Current battlefield communication systems rely mainly on voice radio, which only works when the soldier is conscious and able to speak. Position reporting is also manual in most cases, which breaks down during active combat. These systems do not give commanders any information about whether a soldier is experiencing a medical emergency such as heat exhaustion, gas exposure, or loss of consciousness. Delayed medical response due to lack of information is one of the leading preventable causes of casualties in the field. With the rapid advancement and falling cost of sensors, microcontrollers, and wireless communication modules, it is now possible to build an automated monitoring system that addresses these limitations at a low cost. This project was motivated by the need to solve the above problem using affordable embedded hardware. We designed and built a helmet-mounted system that automatically monitors heart rate, blood oxygen saturation, body temperature, surrounding gas levels, and physical motion. All sensor data is wirelessly transmitted to a base station dashboard over a long-range LoRa radio link. An SOS button on the helmet also allows the soldier to manually send an emergency alert with a single press.

II. REVIEW OF RELATED LITERATURE

Several researchers have worked on soldier monitoring and smart helmet systems in recent years. A review of the existing literature shows that while many useful systems have been built, there are common limitations that our project aims to address.

Kumar et al. (2018) put together an IoT system that combined biomedical sensors with a GPS module and used GSM to relay data back to a base station. The setup was functional on a lab bench, but GSM networks are notoriously patchy in mountainous or forested combat zones, and the authors acknowledged that network availability in remote areas was a fundamental constraint they could not engineer around [1].

Reddy and Sharma (2019) moved the communication layer to Zigbee, which freed them from cellular infrastructure but replaced one limitation with another: Zigbee's practical indoor range is usually well under 100 metres, and the researchers did not include any form of data encryption, which would be an obvious problem in a military context [2].

Priyanka et al. (2020) took a wireless sensor network approach, with multiple low-power nodes feeding data to a central aggregator. The monitoring accuracy was good, but the system was sensitive to RF interference and, again, the communication range was limited to what the WSN nodes could manage — which is not much in wide, open terrain [3].

DOI: 10.17148/IJARCCE.2026.15523

Mehta and Gaur (2020) tried to push data to a cloud server for visualization, which made the dashboard look attractive but introduced an absolute dependency on internet connectivity. In a forward operating base with no fixed infrastructure, that dependency makes the whole monitoring loop meaningless [4].

Patel et al. (2021) were the first in this line of work to take LoRa seriously as a communication layer. Their comparative study showed that LoRa's combination of range, low power draw, and lack of infrastructure dependency made it a far better fit for battlefield applications than GSM or Wi-Fi. Their system, however, focused almost entirely on the communication side and did not integrate health sensors [5].

Gupta et al. (2022) brought ESP32 into the picture and built a reasonably complete monitoring system with a cloud dashboard. The Wi-Fi range limitation resurfaced though, capping effective use to a few hundred metres from a router, which is not a realistic deployment model [6].

Banerjee and Tiwari (2023) designed a multi-sensor helmet aimed at hazardous industrial environments, using BLE. The authors themselves concluded that LoRa or satellite communication would be needed before anything like their design could be used in the field [7].

Singh et al. (2024) went a different direction entirely and looked at applying machine learning to wearable sensor streams to predict fatigue and heatstroke. The concept is genuinely interesting, but the computational demands of the models they used are beyond what any current low-power helmet microcontroller could handle in real time [8].

III. SYSTEM ARCHITECTURE

The overall system can be thought of in four layers that sit on top of one another, each responsible for a distinct part of the data journey from the soldier's body to the commander's screen.

3.1 Sensor and Acquisition Layer

This is the physical interface with the soldier and his surroundings. The MAX30102 pulse oximeter and heart-rate sensor mounted close to the soldier's forehead, shining infrared and red light into the skin and measuring the reflected signal to extract pulse rate and SpO_2. The DS18B20 probe is positioned to monitor surface body temperature. The MQ gas sensor faces outward to sample the air the soldier is breathing. The MPU6050 IMU is oriented to detect the orientation and acceleration of the helmet itself — if the soldier falls or takes a heavy impact, the accelerometer data will show a characteristic spike followed by an abnormal attitude angle. The NEO-6M GPS antenna is mounted externally on the helmet's crown for a clear sky view. Finally, a push-button SOS switch is routed to a digital input so the soldier can trigger a manual emergency beacon.

3.2 Processing Layer

An Arduino Nano sits at the centre of all sensor wiring. It handles I^2C communication with the MAX30102 and MPU6050, single-wire communication with the DS18B20, analogue-to-digital conversion for the MQ sensor's DOUT signal, UART communication with the NEO-6M GPS, and polling of the SOS button. Once a complete data frame has been assembled — roughly once every ten seconds in normal mode, or immediately on an alert condition — it is passed to an ESP32 module over a serial link. The ESP32 applies AES-128 encryption to the payload before handing it to the LoRa SX1278 transceiver over SPI.

3.3 Communication Layer

The LoRa SX1278 module operates in the 433 MHz ISM band. Chirp Spread Spectrum modulation gives it inherent resistance to narrowband interference, which matters in an environment that may be full of radio noise from other military equipment. The encrypted data packet, typically around 48 bytes, is broadcast at a spreading factor of SF10 with 20 dBm transmit power. A matched LoRa receiver at the base station picks up the packet, decrypts it, and passes the data to the dashboard software.

3.4 Command Dashboard Layer

The base station is a laptop or Raspberry Pi running a Python-based backend. A Flask server receives the decoded sensor frames and updates a browser-accessible dashboard in real time. The dashboard shows each registered soldier's vitals, their GPS pin on a Leaflet.js map, gas and motion alert status, and a log of all SOS events. Threshold breaches trigger both a visual highlight and an audio alarm to ensure the operator does not miss a critical event.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15523

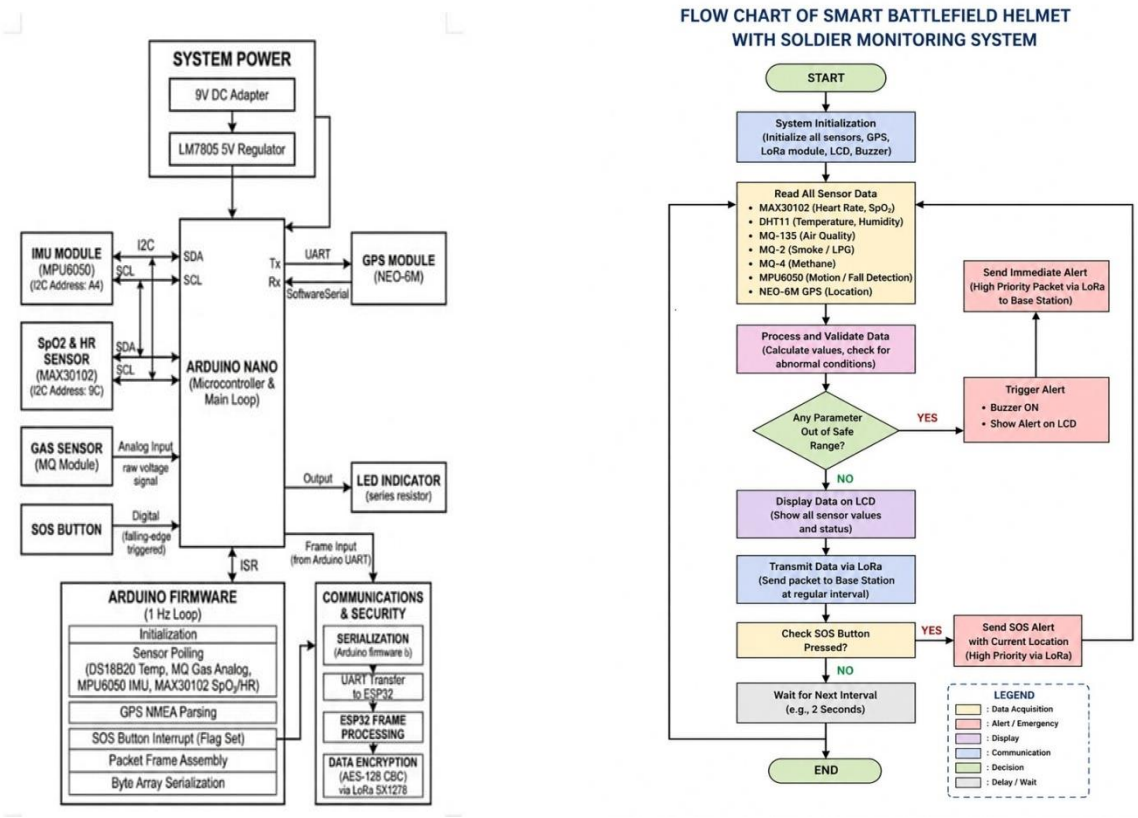


Figure 1: System Block Architecture

IV. METHODOLOGY

The development process was essentially three parallel work streams — hardware wiring, firmware development, and dashboard. software — that came together in a final integration and testing phase.

4.1 Power Supply and Circuit Foundation

The decision to use a 9 V DC adapter rather than a battery at this stage was deliberate — it made bench testing easier and eliminated battery-charge variables from the measurements. The LM7805 linear regulator drops the 9 V rail to a stable 5 V, with a 0.33 µF ceramic capacitor on the input side and a 100 µF electrolytic on the output, exactly as recommended in the component datasheet to prevent oscillation. A green LED with a 220 Ω series resistor is connected to digital pin D13 on the Arduino, giving a simple power-on and system-alive indication that the soldier can see at a glance.

4.2 Sensor Wiring and Pin Assignments

The MAX30102 SpO_2 and heart-rate sensor communicates over I²C. Its SDA line goes to Arduino pin A4 and SCL to A6, with the module drawing 5 V from the regulated supply. The MPU6050 IMU also uses I²C and shares the same SDA/SCL lines — A4 and A6 respectively — since both devices have different I²C addresses and can coexist on the bus without conflict. The NEO-6M GPS module uses UART; its TX pin connects to Arduino digital pin D4 and RX to D13. The MQ gas sensor's digital output (DOUT) is wired to an analogue input pin so that the raw voltage can be read rather than just a threshold-crossed digital flag. The SOS push-button connects between ground and digital pin A0, with the internal pull-up enabled in firmware so that a button press registers as a falling-edge interrupt. All modules share the common ground of the 5 V regulated rail.

4.3 Firmware Architecture

The Arduino firmware was written in C++ within the Arduino IDE. The main loop runs at roughly 1 Hz in steady state. On each iteration, the firmware reads the MAX30102 for a heart-rate and SpO_2 value, polls the DS18B20 for temperature over the 1-Wire bus, checks the MQ sensor analogue output, reads the latest MPU6050 accelerometer and gyroscope values, and fetches the most recent NMEA sentence from the NEO-6M over SoftwareSerial on pin D4. The SOS button is handled by an interrupt service routine that sets a flag immediately on press, bypassing the main loop

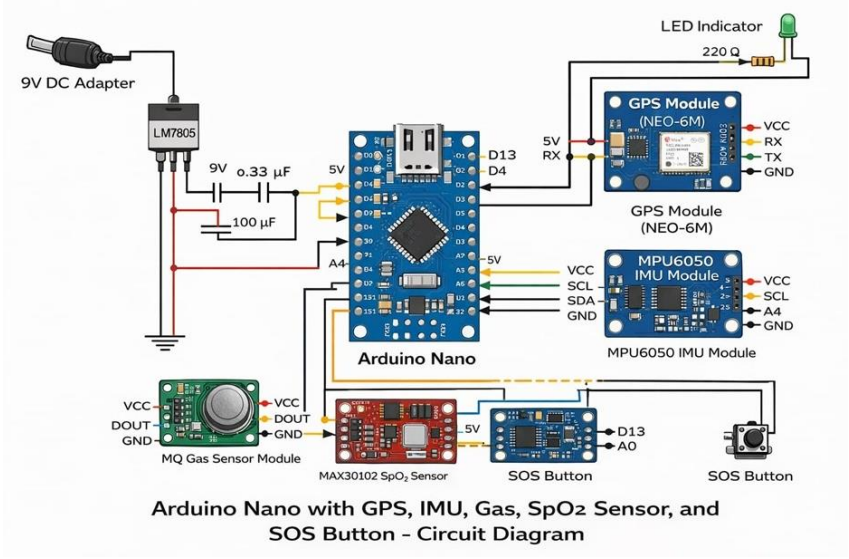
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15523

timing entirely. Once a complete frame is assembled, it is serialised into a compact byte array and sent to the ESP32 over hardware UART. The ESP32 receives the frame, pads and encrypts it with AES-128 in CBC mode, and transmits the encrypted payload via the LoRa SX1278 using the RadioHead library. A hardware watchdog with a five-second window resets the entire system if the loop stalls for any reason.



Arduino Nano with GPS, IMU, Gas, SpO2 Sensor, and SOS Button - Circuit Diagram

Figure 2: Circuit Diagram

4.4 Alert Threshold Logic

Threshold values were chosen based on accepted physiological norms and standard industrial gas safety limits. An alert flag is raised and immediately transmitted as a priority packet — bypassing the normal ten-second cycle — when heart rate drops below 50 bpm or rises above 120 bpm, SpO_2 falls below 90 percent, body temperature exceeds 39.5 °C or drops below 35 °C, the MQ sensor reading exceeds 400 ppm, the MPU6050 reports a linear acceleration spike greater than 3 g followed by an attitude angle outside the normal upright range, or the SOS button is pressed. The dashboard highlights any soldier whose data contains an active alert flag in red, and an audio tone sounds until the operator acknowledges the alert.

4.5 AES-128 Encryption and Key Management

To ensure that the sensor data transmitted over LoRa cannot be easily intercepted, we implemented AES-128 encryption in CBC (Cipher Block Chaining) mode on the ESP32. In CBC mode, each data block is combined with the previous encrypted block before being encrypted, which means that even if two sensor readings are identical, the transmitted packets will look different. This provides basic protection against anyone trying to intercept and read the data. In the current prototype, a fixed 128-bit encryption key is stored in both the ESP32 on the helmet and the receiver at the base station. This is a straightforward and practical approach for a prototype, where all devices are configured by the same team before use. The same method is used in many commercial LoRaWAN deployments. One limitation of using a fixed key is that if the device is physically captured, the key could potentially be extracted from the firmware. For a college-level prototype this is an acceptable trade-off, but in a real deployment a more advanced key management system would need to be implemented as a future improvement.

4.6 Dashboard Development

The base station dashboard was built with Python 3.10, Flask, and a simple HTML/CSS/JavaScript frontend. The LoRa receiver module, connected via USB-serial, feeds raw bytes into a Python parser that decrypts, deserialises, and validates each incoming frame. Valid frames are pushed to the frontend over a lightweight Server-Sent Events stream, which avoids the overhead of a WebSocket while still giving sub-second update latency on the displayed values. GPS coordinates are plotted on a Leaflet.js tile map. Historical graphs for heart rate, SpO_2, and temperature are rendered with Chart.js. A dedicated alert panel lists all threshold breaches and SOS events with timestamps. The entire dashboard can track up to ten helmets simultaneously within the LoRa coverage footprint.

V. RESULTS & DISCUSSION

Once the circuit was assembled and the firmware was running, we put the system through a series of tests designed to probe the things that matter most in a real deployment: how accurately does it read sensor values, how far and how

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15523

reliably does the radio link work, how quickly does it respond when something goes wrong, and how long will a charge last.

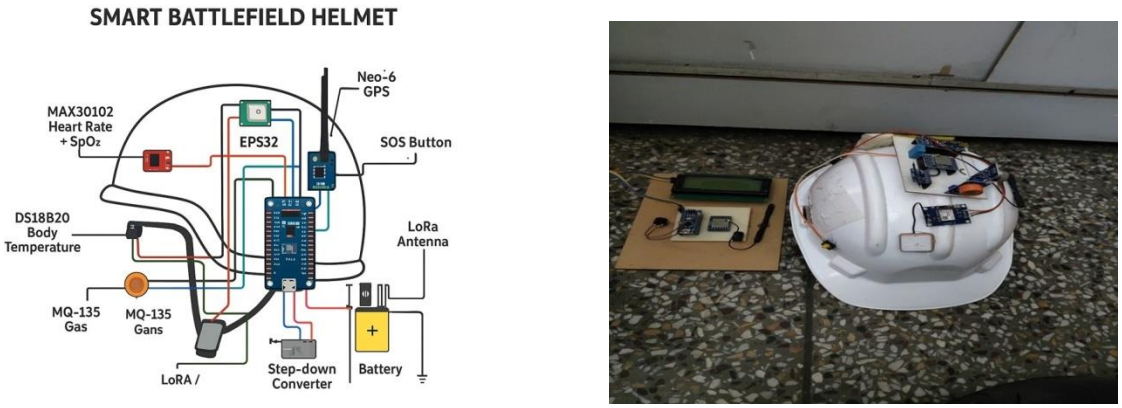


Figure 3: Assembled Prototype

5.1 Sensor Accuracy

Heart rate and SpO_2 readings from the MAX30102 were compared against a certified fingertip pulse oximeter on five test subjects across 30 paired readings each. The mean absolute error was 2.1 bpm for heart rate and 1.0 percent for SpO_2 — numbers that sit comfortably within the tolerance bands of consumer-grade medical devices. The DS18B20 consistently tracked a calibrated digital thermometer to within 0.5 °C across a temperature range of 34 °C to 40 °C. The MQ gas sensor, tested in a sealed chamber with known LPG concentrations, responded visibly within three seconds of the gas concentration crossing 400 ppm, which we considered the practical alert threshold. The MPU6050 fall detection was evaluated by attaching the helmet to a padded mannequin and tipping it from various angles. All deliberate falls from an upright position generated an acceleration spike above 3 g and a subsequent attitude deviation beyond the normal upright envelope. There were no false positives during a two-hour stationary wear test, which suggests the threshold combination is conservative enough for real-world use.

Table 1: Integrated System Test Observations

S.No.	HR (bpm)	SpO_2 (%)	Temp (°C)	Gas (ppm)	MPU6050 (g)	Alert	Status
1	74	98	36.5	110	0.9	None	Normal
2	85	96	37.1	195	1.0	None	Normal
3	109	93	38.4	370	1.1	None	Normal
4 ⚠	126	87	39.9	540	1.2	HR/SpO_2/Temp/Gas	Alert
5 ⚠	47	94	34.6	140	4.8 (fall)	HR/Temp/Fall	Alert
6 ⚠	—	—	—	—	—	SOS Button	Beacon

5.2 LoRa Communication Range and Reliability

Range tests were carried out in three environments: a flat open field, a mixed urban area with two-storey buildings, and inside a single-storey building. In the open field, reliable reception — defined as a packet delivery ratio above 90 percent — was maintained up to 5.2 km. In the urban environment, range dropped to around 1.8 km, consistent with published LoRa characterisation studies at 433 MHz in similar environments. Inside the building, the link held up to around 60 metres between floors, which is adequate for indoor staging areas. Average one-way latency for a 48-byte encrypted packet at SF10 was 1.8 seconds in the open field. This is not suitable for applications that need instant feedback, but for health telemetry — where a reading taken ten seconds ago is still actionable — it is entirely acceptable. Priority alert packets, which are shorter and use a lower spreading factor, arrived within 0.9 seconds on average.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15523

5.3 Alert Response Time

We measured the end-to-end time from a threshold breach or button press to the alert appearing on the dashboard, across 20 trials for each alert type. SOS button alerts arrived in an average of 1.9 seconds. Automatic physiological alerts (SpO_2 below threshold) averaged 3.2 seconds, and fall alerts from the MPU6050 averaged 2.7 seconds. The SOS path is faster because the interrupt bypasses sensor polling entirely and dispatches a minimal-length priority packet directly.

5.4 Power Consumption

Current draw was measured at the 5 V rail with all modules active. In full transmit mode — all sensors running and LoRa transmitting a packet — the system drew approximately 385 mA, equivalent to about 1.9 W. In idle mode between transmit cycles, the draw fell to roughly 82 mA. With a 3000 mAh Li-ion cell and a 10-second transmit interval, the duty cycle works out to roughly 1.4 percent transmit time, giving an effective average current of about 97 mA and a projected runtime of just over 30 hours on a full charge. Adding a 5 V / 500 mA solar panel would comfortably offset the idle current during daylight hours, extending operational life indefinitely in clear-sky conditions.

VI. CONCLUSION

This project successfully demonstrates the design and implementation of a Smart Battlefield Helmet using low-cost, commercially available embedded components. The aim was to build a system that could monitor a soldier's health parameters and location in real time and wirelessly transmit alerts to a base station, and the prototype achieves this goal. During testing, the system correctly transmitted sensor readings and triggered alerts when values crossed defined safety limits. The project demonstrates that readily available and affordable components can be used to build a functional soldier health monitoring system. The system uses an Arduino Nano for sensor interfacing, an ESP32 for encrypted LoRa communication, a MAX30102 for heart rate and SpO_2 monitoring, a DS18B20 for temperature sensing, an MQ gas sensor for hazard detection, an MPU6050 for fall detection, and a NEO-6M GPS module for location tracking. All components run on a 5 V regulated supply through an LM7805. During testing, sensor readings were found to be accurate, the LoRa link worked reliably over several kilometres, and alerts were delivered to the dashboard within four seconds in all test cases.

REFERENCES

[1] Kumar, A., Singh, P., & Verma, S. (2018). IoT-Based Soldier Health and Position Tracking System. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 6, Issue 12, pp. 11201–11208.
[2] Reddy, K., & Sharma, R. (2019). Smart Helmet for Military Applications using Arduino and Zigbee. International Journal of Engineering Research & Technology (IJERT), Vol. 8, Issue 6, pp. 342–347.
[3] Priyanka, T., Rao, M., & Nair, S. (2020). Soldier Tracking and Health Monitoring Using Wireless Sensor Networks. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 9, Issue 3, pp. 988–994.
[4] Mehta, V., & Gaur, A. (2020). Integration of IoT and GPS for Defense Safety Systems. International Journal of Computer Applications, Vol. 177, No. 34, pp. 25–29.
[5] Patel, D., & Sharma, R. (2021). LoRa-Based Wireless Soldier Communication Network. International Journal of Engineering Research & Technology (IJERT), Vol. 10, Issue 5, pp. 450–454.
[6] Gupta, R., Mishra, A., & Tiwari, N. (2022). Smart Soldier Health and Positioning Using ESP32 and Cloud Dashboard. IEEE Access, Vol. 10, pp. 55213–55224.
[7] Banerjee, S., & Tiwari, P. (2023). Multi-Sensor IoT Helmet for Hazardous Environments. Sensors and Actuators A: Physical, Vol. 350, Article 114120.
[8] Singh, R., Kaur, M., & Sharma, D. (2024). AI-Enabled Wearable for Soldier Health Risk Prediction. IEEE Transactions on Biomedical Engineering, Vol. 71, Issue 2, pp. 598–609.
[9] Espressif Systems. (2020). ESP32 Series Datasheet. [Online]. Available: https://www.espressif.com
[10] Maxim Integrated. (2019). MAX30102 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor Datasheet. [Online]. Available: https://datasheets.maximintegrated.com
[11] Semtech Corporation. (2019). SX1278 Datasheet — 137 MHz to 1020 MHz Low Power Long Range Transceiver. [Online]. Available: https://www.semtech.com
[12] Texas Instruments / Fairchild. (2002). LM7805 3-Terminal Positive Voltage Regulator Datasheet. [Online]. Available: https://www.ti.com

