

# International Journal Research Publication Analysis

Page: 01-09

---

## ASTROPULSE-SMART SPACE HEALTH MONITOR

---

**\*Anagha Santosh**

---

India.

---

Article Received: 13 September 2026

\*Corresponding Author: Anagha Santosh

Article Revised: 03 October 2026

India.

Published on: 23 October 2026

DOI: <https://doi-doi.org/101555/ijrpa.7667>

---

### ABSTRACT

This paper describes the implementation of an Internet of Things (IoT)-based Smart Space Suit Health Monitoring prototype built using an ESP32 development board, a MAX30102 optical sensor, and a Node.js-based AstroPulse monitoring platform. The sensor acquires red and infrared photoplethysmographic signals for algorithmic estimation of pulse rate and peripheral oxygen saturation (SpO<sub>2</sub>). Measurements are processed by the ESP32 and transmitted through Wi-Fi using HTTP/JSON to a web application that stores data in SQLite and provides live visualisation, history and alert functionality. The implementation verified sensor communication, changes in optical readings with finger placement, wireless connectivity, successful HTTP ingestion and dashboard display. However, pulse and SpO<sub>2</sub> outputs remained intermittent and unstable, and further calibration and validation are required. A basic glove-mounted demonstrator was also assembled using Velcro straps and direct jumper connections.

**KEYWORDS:** ESP32, MAX30102, IoT, PPG, Heart Rate, SpO<sub>2</sub>, AstroPulse, Wearable Monitoring.

### I. INTRODUCTION

The Smart Space Suit Health Monitoring System investigates how compact embedded sensors can be integrated with wireless communication and a remotely accessible monitoring interface. The prototype focuses on heart-rate and SpO<sub>2</sub> estimation. It is a laboratory proof of concept motivated by applications in demanding environments; it has not been tested in space or validated for medical use.

## II. System Architecture

The architecture is divided into sensing, embedded processing, wireless communication and server-based monitoring. The MAX30102 supplies red/infrared measurements to the ESP32 via I2C. The ESP32 calculates candidate physiological values and, when configured validation checks pass, sends JSON data by HTTP POST. AstroPulse identifies the suit using a device key, records accepted readings in SQLite and presents results in the browser.

Data flow: MAX30102 → I2C → ESP32 → 2.4 GHz Wi-Fi → HTTP POST /api/ingest → AstroPulse Node.js server → SQLite → browser dashboard.

## III. Hardware Components

### A. ESP32 Development Board

The ESP32 is the central controller. It provides processing, I2C communication and built-in Wi-Fi. GPIO 21 (D21) serves as SDA, while GPIO 22 (D22) serves as SCL. The development board was programmed and powered by USB during the demonstrated tests.

### B. MAX30102 Optical Sensor

The MAX30102 uses red and infrared illumination together with a photodetector to acquire photoplethysmography (PPG) signals related to pulsatile changes in blood volume. A MAX30102 breakout with SDA, SCL, VIN and GND pins was connected to the ESP32. The breakout supply must match its actual board specification.

### C. I2C LCD Display

A 16×2 character LCD with an I2C backpack was present in the prototype hardware set for local readout. Shared SDA/SCL wiring is possible if the LCD and sensor operate at compatible logic voltages. The documented tests did not establish verified simultaneous LCD output. A 5 V LCD backpack may require appropriate level shifting to protect the ESP32.

### D. Breadboard, Jumpers and Power

A solderless breadboard and jumper wires were used for initial assembly and debugging. The USB connection supplied the ESP32 during testing. Rechargeable battery operation using a Li-Po cell, TP4056 module and appropriate regulation remains a proposed enhancement, not a verified operational subsystem.

## E. Components and Status

| Component / software              | Role                                | Status                       |
|-----------------------------------|-------------------------------------|------------------------------|
| ESP32 DevKit                      | Controller and Wi-Fi communication  | Implemented                  |
| MAX30102                          | Red/IR optical sensing              | Implemented                  |
| Breadboard + jumper wires         | Temporary circuit interconnections  | Implemented                  |
| USB cable                         | Programming and power               | Implemented                  |
| 16×2 I2C LCD                      | Local measurement display           | Present; output unverified   |
| Arduino IDE + SparkFun MAX3010x   | Firmware and sensor interface       | Implemented                  |
| Node.js + AstroPulse              | Data ingestion and monitoring       | Implemented                  |
| SQLite                            | Measurement storage                 | Implemented                  |
| Glove, Velcro wrist/finger straps | Wearable packaging / portable power | Glove demonstrator assembled |

## IV. Hardware Connections

The ESP32 and MAX30102 were linked through separate electrically common breadboard rows. The A–E holes on each numbered row form a connected strip in the typical breadboard layout used. Table II specifies the planned signal arrangement; placement is illustrative rather than a universal breadboard pinout. The assembled laboratory prototype is documented in Fig. 1.

| ESP32 pin     | Breadboard route | MAX30102 pin | Function                               |
|---------------|------------------|--------------|----------------------------------------|
| D21 / GPIO 21 | A10 → B10        | SDA          | I2C data                               |
| D22 / GPIO 22 | A15 → B15        | SCL          | I2C clock                              |
| 3V3           | A20 → B20        | VIN          | Supply if breakout is 3.3 V compatible |
| GND           | A25 → B25        | GND          | Common ground                          |

The I2C LCD would use GPIO 21 (SDA), GPIO 22 (SCL), a common ground and a voltage-compatible supply. Where its backpack pulls signals to 5 V, an I2C level shifter is necessary. Only the verified MAX30102 sensing arrangement should be presented as completed hardware integration.

## V. Embedded Software and Signal Processing

### A. Initialization and Sampling

The firmware was developed in Arduino IDE using Wire.h, WiFi.h, HTTPClient.h and the SparkFun MAX3010x library. The ESP32 configures Wire.begin(21, 22), initializes the

sensor, sets the red/IR LED operating parameters and reads digital samples into red and infrared buffers. The later experimental firmware used 100-sample windows with 25 new measurements per update.

### **B. Heart-Rate Estimation**

Heart rate is derived from rhythmic changes in PPG. Conceptually,  $BPM = 60/T$ , where  $T$  is the time between successive beats in seconds. The project uses the bundled pulse-oximetry processing routine to estimate heart rate from samples; results are not guaranteed valid solely because they fall within a plausible numerical range.

### **C. SpO<sub>2</sub> Estimation**

SpO<sub>2</sub> estimation depends on the contrasting optical absorption of oxygenated and deoxygenated haemoglobin. The conceptual ratio of ratios is  $R = (AC\_red/DC\_red) / (AC\_IR/DC\_IR)$ . A calibrated algorithm maps optical characteristics to estimated oxygen saturation. The implementation used the MAX3010x-associated `spo2_algorithm` processing function; neither a simple IR amplitude nor an HTTP success response establishes physiological accuracy.

### **D. Data Validity**

The firmware checks the algorithm-reported validity flags and numerical ranges before transmitting a record. In several tests the sensor returned changing red/IR measurements while BPM and SpO<sub>2</sub> were invalid. This distinction is essential: optical response demonstrates sensor operation, not reliable pulse oximetry.

## **VI. Communication and AstroPulse Application**

### **A. Wi-Fi and HTTP**

The ESP32 joins a compatible 2.4 GHz wireless network and sends JSON objects containing heartRate and spo2 to the AstroPulse `/api/ingest` endpoint. A device-specific key is supplied in the X-Device-Key request header. The server address during one local test used a laptop-local IP and TCP port 3000; this address varies by network and is not a permanent deployment URL. Secrets are intentionally omitted from this document.

### **B. Backend and Database**

The AstroPulse application uses Node.js and SQLite to receive and retain measurements. The project design supports registered suits, account sessions, data history, device-key rotation and configured alerts. The backend expects valid numeric parameters and identifies each suit separately.

### C. Web Dashboard

The browser interface presents incoming BPM and SpO<sub>2</sub> values, status information and historical statistics. During testing the dashboard displayed values that had been accepted by the server. The status could subsequently switch to “Suit offline” when new accepted readings were not received within the configured stale period. This does not necessarily indicate loss of electrical power or Wi-Fi, because the firmware rejects invalid sensor estimates. Example dashboard measurements and offline status are presented in Fig. 2, while the alert and unavailable gravity panel are shown in Fig. 3.

### VII. Integration and Experimental Observations

| Test                     | Observed outcome                                                                     | Interpretation                                    |
|--------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------|
| ESP32 firmware upload    | Upload verification succeeded                                                        | Controller programmable                           |
| I2C communication        | MAX30102 returned red/IR readings                                                    | Digital sensing interface operational             |
| Finger placement         | IR changed from roughly 2,450 uncovered to approximately 200,000 covered in one test | Strong optical response; accuracy not established |
| Wi-Fi                    | ESP32 connected and obtained a local IP                                              | Wireless link established                         |
| HTTP data ingestion      | HTTP 201 observed                                                                    | At least one reading accepted by server           |
| AstroPulse interface     | BPM/SpO <sub>2</sub> values displayed                                                | End-to-end data path demonstrated                 |
| Physiological estimation | Frequent calculating/invalid and unstable results                                    | Further signal processing and validation required |
| Online status            | Suit occasionally marked offline                                                     | Recent valid ingestion not sustained              |

The results support a successful proof-of-concept connection from sensor measurements through the ESP32 to the web application. The optical readings exhibited substantial variation, including occasional very high IR values; sensor contact, ambient light, acquisition settings and algorithm calibration need further investigation. No claim of medical-grade SpO<sub>2</sub> or continuous monitoring performance is made.

### VIII. Glove-Based Wearable Prototype Assembly

A glove-mounted demonstrator was assembled using a black fabric glove, a Velcro finger strap, and a wrist band. Four jumper wires connect the MAX30102 finger sensor to the ESP32. The sensor is positioned at the fingertip and the wires are routed along the glove to the wrist, where the electronics are secured for a stationary demonstration. Figures 4 and 5 show the assembled prototype in front of the AstroPulse dashboard. This is an experimental

wearable arrangement powered by USB, not a fully enclosed or independently powered device. Connector protection, mechanical strain relief and validation under motion remain necessary.

## **IX. Limitations and Future Work**

Limitations include sensitivity to fingertip placement, movement, illumination, optical saturation, incomplete validation of estimated BPM/SpO<sub>2</sub> and loss of live dashboard updates whenever readings are rejected. Further work includes reference-device comparison, acquisition-quality metrics, robust pulse processing, power budgeting, safe enclosure design, durable glove integration, tested LCD output and a separate device heartbeat so communication status can be distinguished from measurement validity. Gravity monitoring with an MPU6050 is described in the AstroPulse project architecture but was not implemented in the documented sensor build.

## **X. CONCLUSION**

The implementation demonstrates an ESP32–MAX30102 sensor interface with Wi-Fi transmission to the AstroPulse Node.js monitoring platform. Sensor optical response, successful network operation, HTTP ingestion and browser-based display were observed. The work establishes the feasibility of the IoT software/hardware pathway while identifying sensor stability and physical wearability as ongoing engineering tasks. Claims of health-measurement accuracy or suitability for actual space operations require additional validation.

## **XI. Photographic Evidence of Prototype Implementation**

Figures 1–3 document the initial breadboard prototype and monitoring dashboard; Figures 4–5 show the subsequently assembled glove-mounted demonstration. Displayed physiological values are prototype estimates and do not establish medical accuracy.

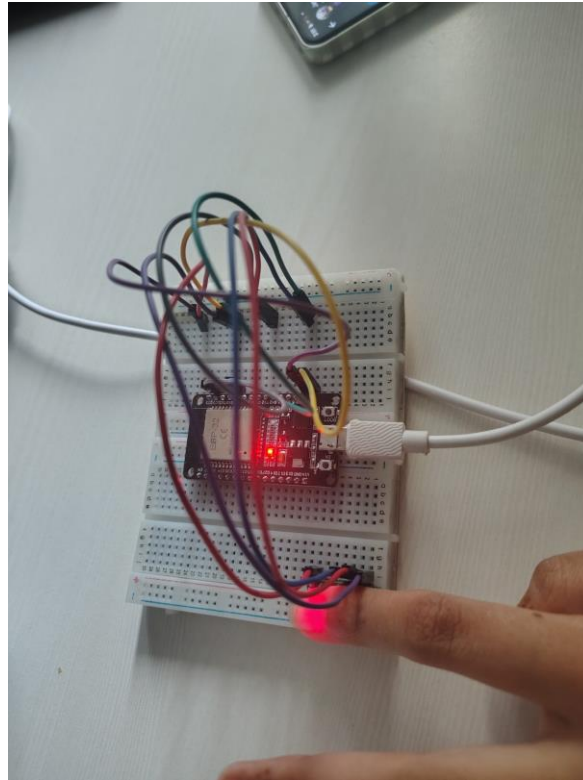


**Fig. 1.** ESP32 and MAX30102 breadboard prototype powered through USB, showing finger placement over the optical sensor.

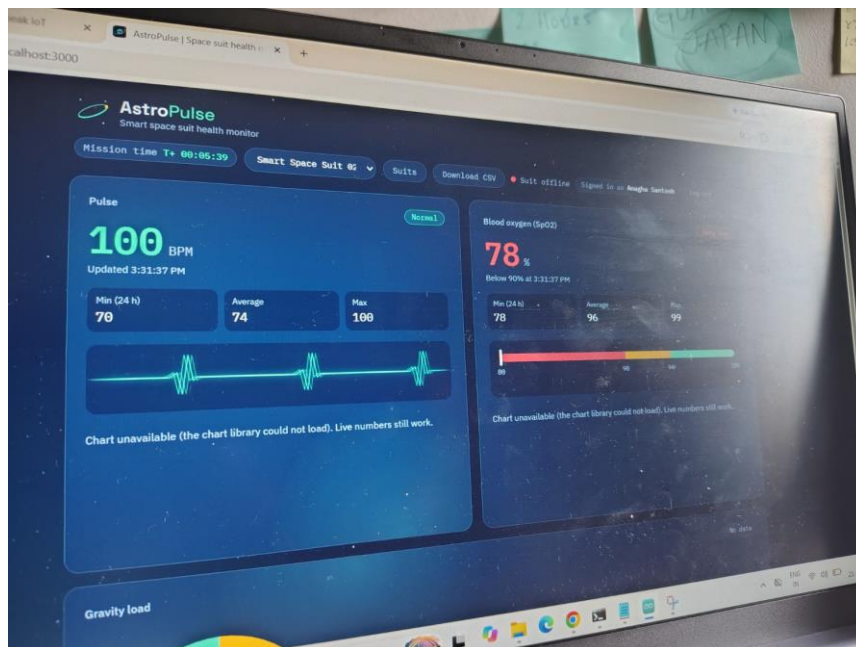


**Fig. 2.** AstroPulse dashboard displaying recorded heart-rate and SpO<sub>2</sub> estimates, statistical summaries, and device offline status.

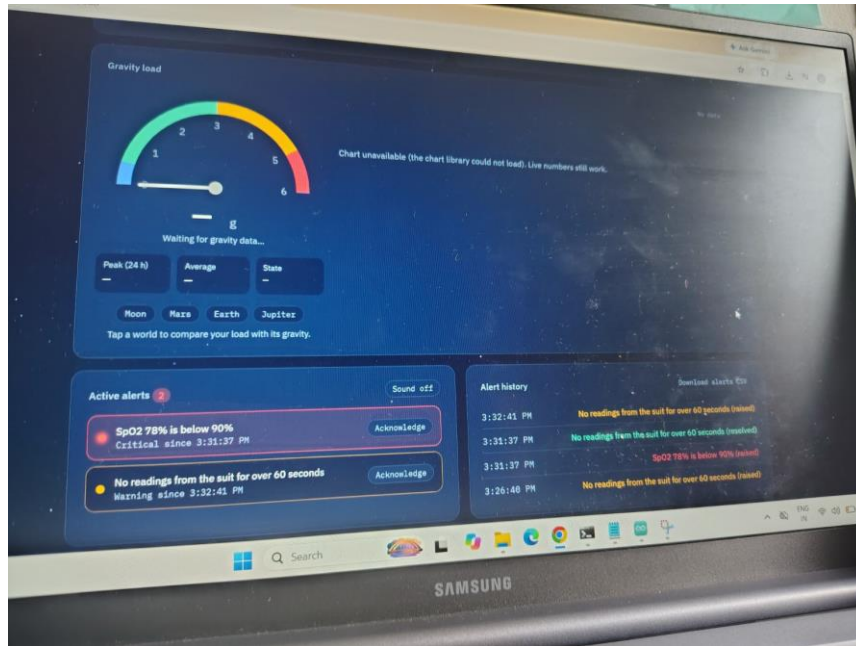


*Fig. 3. AstroPulse gravity panel and alert history showing low-SpO<sub>2</sub> and missing-reading alerts; gravity measurement is not available in this setup.*



*Fig. 4. Assembled glove-mounted AstroPulse prototype with fingertip MAX30102 sensor, routed jumper wires and Velcro wrist support.*



**Fig. 5. Alternate view of the wearable glove prototype shown in front of the AstroPulse monitoring dashboard.**

#### **Author Notes Before Submission**

Replace this note with your institution's author details; add a measured validation dataset and independently checked IEEE references. The current document is an implementation draft, not a complete peer-reviewed research article. Never include network passwords or device keys in a publication.