

MESH-BASED ENVIRONMENTAL MONITORING SYSTEM FOR RESIDENTIAL SPACES

A.C. GHEORGHE

Valahia University of Targoviste, Faculty of Electrical Engineering, Electronics, and Information Technology
E-mail: andrei.gheorghe@valahia.ro

Abstract. *This paper presents the design and implementation of a Wi-Fi mesh network system for distributed monitoring of environmental parameters (temperature, humidity, atmospheric pressure, and air quality/VOC) in a residential setting. The system comprises one ROOT node acting as gateway and web server, and three SLAVE nodes for data acquisition. Built on the ESP32 platform using the painlessMesh library, the system operates fully locally without Internet dependency. A web interface accessible via captive portal enables real-time visualization, configurable alert thresholds, topology mapping, and CSV data export. Hardware integration includes BME680 sensors, OLED displays, autonomous 18650 battery power, and 3D-printed enclosures. Test results confirm coherent multi-room data differentiation and stable mesh.*

Keywords: mesh network, IoT, ESP32, BME680, environmental monitoring, painlessMesh, indoor air quality.

1. INTRODUCTION

The digitalization of residential spaces has become a natural direction in the context of rapid IoT development and increased accessibility of wireless microcontrollers. Environmental parameters such as temperature, humidity, atmospheric pressure, and air quality directly influence comfort, health, and energy consumption. Excessive humidity can promote mold growth, while temperature fluctuations lead to inefficient use of heating or air-conditioning systems [1]. Most commercial monitoring solutions use a centralized star topology, which becomes limiting in multi-room dwellings with thick walls or areas of weak wireless coverage. A mesh topology, where nodes form a self-organizing network capable of retransmitting data through intermediate nodes, offers superior coverage and robustness [2], [3].

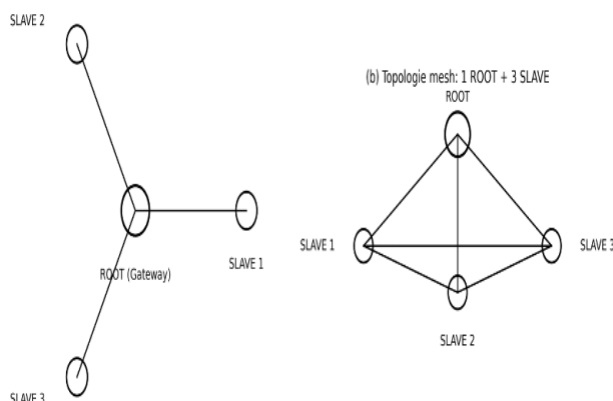


Figure 1. Network topologies: star and mesh with 1 ROOT + 3 SLAVE nodes.

This paper presents the HomeMesh system: a locally operating, autonomous, and easily extendable mesh monitoring solution. The system uses four ESP32 nodes (one ROOT, three SLAVE) with BME680 sensors, OLED displays, SD card logging, and a captive portal web interface

2. SYSTEM ARCHITECTURE AND DESIGN

2.1 Functional Requirements

The system must: (R1) monitor environmental parameters in multiple rooms; (R2) implement a distributed node architecture; (R3) provide stable mesh communication with automatic reconnection; (R4) offer a browser-accessible local web interface showing node status, historical graphs, topology map, and alert indicators; (R5) allow remote configuration of thresholds and reporting intervals; (R6) generate alerts when values exceed configurable thresholds; and (R7) support local CSV storage and file download [4].

2.2 Node Architecture and Communication

The ROOT node creates and maintains the local Wi-Fi network, aggregates data from SLAVE nodes, evaluates alert thresholds, logs data to SD card in CSV format, and serves a local web interface. The three SLAVE nodes acquire environmental data using BME680 sensors, display values on an OLED screen, and periodically transmit data and heartbeat messages through the mesh. The IEEE 802.11s standard and the painlessMesh library provide self-organizing, self-repairing behavior [5], [6].

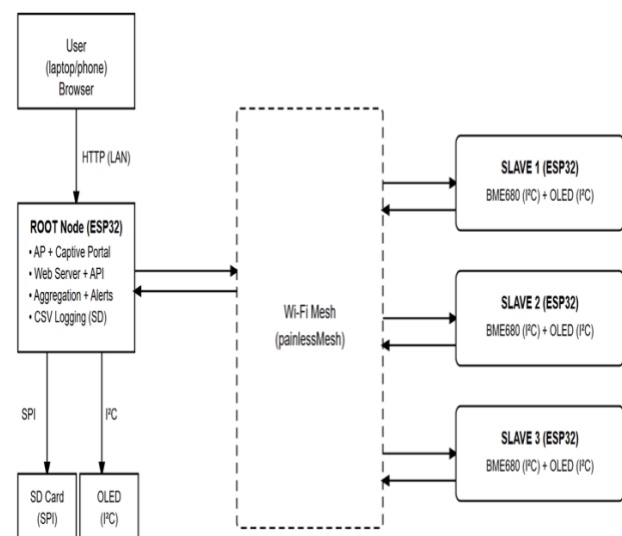


Figure 2. System architecture: 1 ROOT + 3 SLAVE nodes communicating via Wi-Fi mesh.

2.3 Message Protocol (JSON)

Inter-node communication uses JSON messages differentiated by a "type" field: (1) data messages from SLAVE nodes containing T/H/P/gas measurements; (2) heartbeat (hb) messages containing RSSI, uptime, heap, and neighbor list; and (3) configuration (cfg) messages from ROOT containing thresholds, reporting interval, OLED state, and node label. Alert generation is centralized at ROOT: temperature alert if $T > t_{High}$, humidity alert if $H > h_{High}$, gas/VOC alert if resistance $< gasLow$ [7], [8].

3. HARDWARE IMPLEMENTATION

3.1 Key Components

All nodes are based on the ESP32-DevKitC (2.4 GHz Wi-Fi, Arduino ecosystem support). The Bosch BME680 sensor provides temperature, humidity, pressure, and gas/VOC resistance on I2C alongside an SSD1306 OLED display. Autonomous power comes from an 18650 Li-Ion cell (2984 mAh) with TP4056 USB-C charger and a 5V boost converter. All four nodes share the same power architecture [8], [9], [10].

3.2 Node Construction and Enclosures

SLAVE nodes are assembled on perfboards with dedicated zones for power distribution and I2C signal routing. Custom 3D-printed PETG enclosures (0.20mm layer height) include compartments for ESP32 antenna, BME680 sensor isolated from heat sources, TP4056 charger with external USB-C access, power switch, and 18650 cell. The lid features an OLED cutout for local readout. The ROOT node adds a microSD module on SPI for CSV logging, with a corresponding compartment in its enclosure [11].

4. SOFTWARE IMPLEMENTATION

4.1 SLAVE Firmware

SLAVE firmware uses non-blocking millis()-based timers. BME680 initialization uses an explicit heater profile (320°C, 150ms) for repeatable VOC measurements. Configuration parameters (thresholds, interval, OLED state, label) are stored in NVS via the Preferences API and restored on boot. The bmeOk flag is always transmitted, allowing ROOT to distinguish a valid node with sensor error from an absent node [8], [12].

4.2 ROOT Firmware

ROOT maintains a static array nodes[MAX_NODES] with measurements, timestamps, thresholds, diagnostics, and a circular history buffer (HIST_MAX=300 samples). HTTP endpoints include: /api/data (current state + alerts), /api/history (chunked circular buffer), /api/topology (mesh edge reconstruction from neighbor reports), and /api/cfg (POST for configuration). Configuration changes are saved to NVS and propagated as unicast mesh messages. A DNS server implements captive portal redirection for immediate browser access [13], [14]

5. TEST RESULTS AND ANALYSIS

The system was tested with three SLAVE nodes in different rooms (living room, office, bedroom) and ROOT as gateway. Data was logged to CSV over one session and resampled at 3-minute intervals.

5.1 RSSI and Connectivity

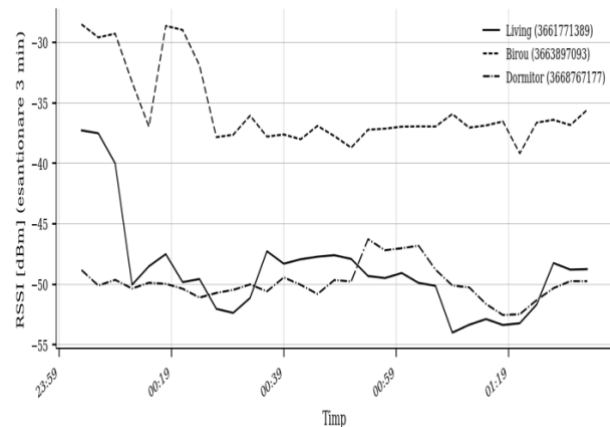


Fig. 3. RSSI levels for the three SLAVE nodes during the test session.

RSSI values ranged between -30 and -55 dBm across the three nodes, with stable time series confirming reliable periodic data delivery. The mesh maintained connectivity throughout the session, validating the self-organizing behavior of painlessMesh in a real indoor environment.

5.2 Temperature and Humidity

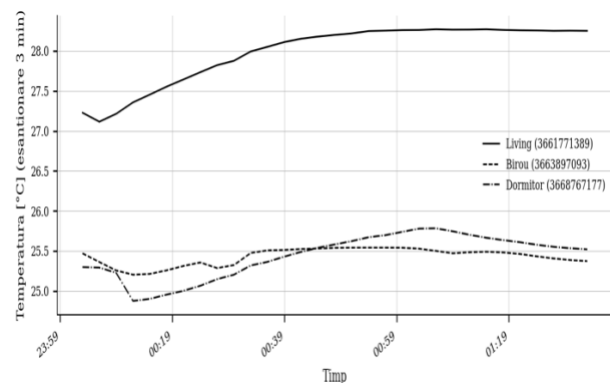


Fig. 4. Temperature measured by SLAVE nodes (3-minute resampling).

Temperature curves show clear inter-room differentiation (25.0-28.2°C), with slow realistic variations typical of indoor environments. Humidity series display room-specific separation consistent with different ventilation and occupancy. Atmospheric pressure from all three nodes showed close agreement, confirming measurement consistency. Gas/VOC resistance curves (115,000-160,000 ohm) showed room-specific profiles coherent with local VOC sources.

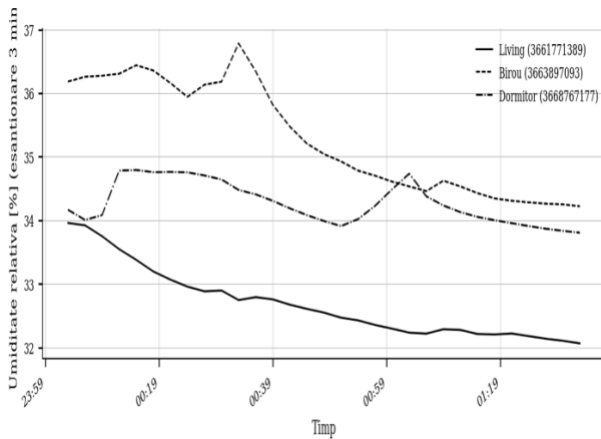


Fig. 5. Relative humidity measured by SLAVE nodes (3-minute resampling).

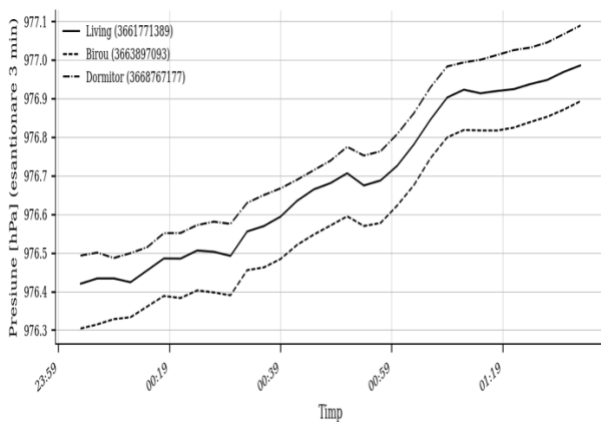


Fig. 5. Air pressure measured by SLAVE nodes (3-minute resampling).

Atmospheric pressure series recorded by all three SLAVE nodes show a high degree of coherence, with values evolving along a common trend throughout the monitoring session. The close agreement between nodes placed in different rooms of the same dwelling confirms the absence of systematic inter-node measurement errors and validates the consistency of the BME680 sensor chain across the deployed hardware. Minor deviations between nodes are within the expected sensor tolerance and do not affect the practical utility of the measurements

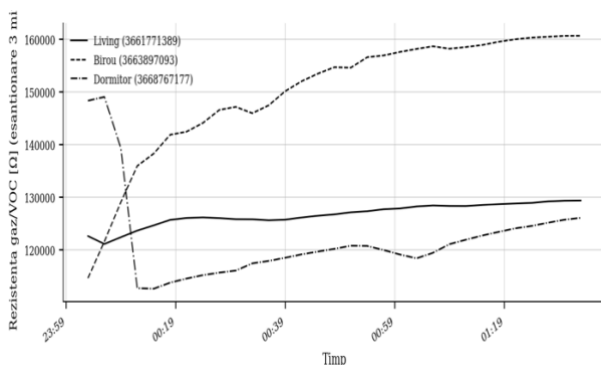


Fig. 5. VOC measured by SLAVE nodes (3-minute resampling).

The gas/VOC channel of the BME680 sensor measures the resistance of a metal-oxide semiconductor (MOS) element, which decreases in the presence of reducing gases and volatile organic compounds (VOCs). Rather than providing a calibrated concentration value, the sensor serves as a relative air quality indicator: a sustained decrease in resistance signals increased VOC loading in the local environment. The three nodes exhibit distinct resistance profiles throughout the session, consistent with different local VOC sources such as furniture off-gassing, occupant activity, and ventilation patterns in each room. The observed upward trend in resistance across all nodes suggests progressive air quality improvement over the monitored period, likely due to natural ventilation. This behavior confirms that the BME680 gas channel, while not a substitute for a dedicated CO₂ or VOC sensor, provides meaningful qualitative information for residential air quality monitoring.

5.3 Alert Evaluation

Alert thresholds configured via the web interface (tHigh=29°C, hHigh=70%, gasLow=40,000 ohm) were correctly evaluated per sample at ROOT. No false alerts were triggered during the session; all indicators showed "OK," consistent with the measured ranges staying within thresholds. The bmeOk validation mechanism prevented propagation of invalid readings during sensor initialization.

6. CONCLUSIONS

This paper presented HomeMesh, a complete mesh-based residential environmental monitoring system. Key contributions include: a fully local operation model with captive portal access; a JSON-based message protocol with three distinct message types; persistent NVS configuration surviving reboots; a circular history buffer with chunked HTTP delivery; topology reconstruction from neighbor reports; and custom 3D-printed PETG enclosures with sensor thermal isolation.

Test results confirm correct operation of the full acquisition-transmission-aggregation-visualization chain, with natural inter-room differentiation for temperature, humidity, and VOC. Future directions include: HMAC message signing; NTP time synchronization; hysteresis-based alert filtering; MQTT/Home Assistant integration; additional sensors (CO₂, PM2.5/PM10); and energy optimization for extended battery autonomy.

7. REFERENCES

- [1] Atzori L., Iera A., Morabito G., "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, 2010.
- [2] Akyildiz I.F., Wang X., "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445-487, 2005.
- [3] Tanenbaum A.S., Wetherall D.J., *Computer Networks*, 5th ed., Pearson, 2011.
- [4] ITU-T, Recommendation Y.2060: Overview of the Internet of Things, International Telecommunication Union, 2012.

- [5] IEEE, IEEE Std 802.11s: Amendment — Mesh Networking (WLAN Mesh), IEEE Standards Association, 2011.
- [6] gmag11, painlessMesh: ESP8266/ESP32 mesh networking library, open-source project documentation.
- [7] Bray T. (Ed.), RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format, IETF, 2017.
- [8] Bosch Sensortec, BME680: Integrated Environmental Sensor — Datasheet and Application Notes, Bosch Sensortec.
- [9] Espressif Systems, ESP32-DevKitC User Guide and ESP32 Series Datasheet, Espressif Systems.
- [10] Panasonic, 18650 Li-Ion Cell Datasheet (NCR family), Panasonic.
- [11] Ahn S. et al., "Design guidelines for FDM 3D-printed enclosures in embedded systems," Journal of Manufacturing Processes, 2019.
- [12] Blanchon B., ArduinoJson: Efficient JSON serialization for embedded systems, official library documentation.
- [13] Espressif Systems, ESP-IDF Programming Guide: Non-Volatile Storage (NVS), Espressif Systems.
- [14] Espressif Systems, ESP-IDF Programming Guide: Wi-Fi SoftAP and DNS Server, Espressif Systems.