**SenseHAT Condition Monitor — Progress & Commissioning Pack (Lite+)**

Plain-language summary with quick-run steps and current status.

Owner: Caoilte Donohoe · Generated: 2025‑10‑16

# Project Overview

Raspberry Pi + Sense HAT edge agent publishing metrics (Temp, Humidity, Pressure, 3‑axis accel) to MQTT and exposing a /health endpoint. Node‑RED dashboard provides live gauges and a temperature trend chart.

# 1) Executive Summary

* Built a compact sensor unit that reads room conditions and device health every ~2 seconds.
* Shows live values on a simple web dashboard and a health page that confirms the unit is OK.
* Shares data to a message hub (MQTT) so other tools can reuse it now or later.

# Current Status (v0.2.0)

* Agent runs as systemd service: pi-sense-agent
* MQTT topic: spBv1.0/sensehat/DDATA/pi-edge (QoS1, retained)
* Health endpoint: http://<pi-ip>:8080/health
* Node‑RED dashboard: http://<pi-ip>:1880/ui
* Modbus‑TCP server (optional): port 5020

# Run / Verify (quick)

mosquitto\_sub -t 'spBv1.0/sensehat/DDATA/pi-edge' -v

curl http://<pi-ip>:8080/health

Open Node‑RED /ui and view Temp gauge + chart

# 2) System Architecture

* Sensors → Pi agent → MQTT → Dashboard/other systems.
* Simple health web page for quick checks.
* Optional: Modbus‑TCP view for building systems (same readings).
* Placeholder for diagram: docs/images/SenseHAT\_Architecture.png

# 3) Implementation Details

## 3.1 Edge Agent

* Starts automatically and restarts if it stops.
* Collects readings (temperature, humidity, pressure, 3‑axis accel) and basic device health.

## 3.2 MQTT (Sparkplug‑B)

* Standard topic naming so tools can subscribe easily.
* QoS1 and retained messages for reliable, last‑value display on dashboards.

## 3.3 HTTP & Dashboard

* Health page shows status “OK” and basic uptime/time.
* Node‑RED /ui shows live gauges and a simple temperature trend chart.

## 3.5 Optional Modbus‑TCP Mapping

* Provides a small register plan so building systems can poll the same readings (port 5020).
* Final tags/registers to be agreed with facilities/BMS team.

# Register Map (Modbus‑TCP)

|  |  |  |  |
| --- | --- | --- | --- |
| Register | Signal | Units | Notes |
| 30001 | env.temp\_c | 0.1 °C | Temperature ×10 |
| 30002 | env.humidity\_pct | 0.1 % | Humidity ×10 |
| 30003 | env.pressure\_hpa | 0.1 hPa | Pressure ×10 |
| 30010 | sys.cpu\_temp\_c | 0.1 °C | CPU temp ×10 |
| 30020 | sys.uptime\_s\_low | s | Low word |
| 30021 | sys.uptime\_s\_high | s | High word |

# 4) Commissioning Runbook (Delivered & Verified)

* Device prepared, named, and connected to the network.
* Software installed; set to start automatically on power‑up.
* Dashboard updates every ~2 seconds; health page shows OK.
* Messages visible on the MQTT hub for other tools to use.

# 5) Configuration

* Set the MQTT server address, login, and how often to send readings.
* Use friendly names; no specialist knowledge required.

# 6) Operations & Troubleshooting

* If the dashboard stops, power‑cycle or restart the unit.
* If values look odd, ensure good ventilation and correct date/time.
* If the dashboard is slow, check Wi‑Fi or use Ethernet.

# Notes / Quirks

* Sense HAT temperature can read slightly high from CPU heat; simple compensation used.
* Pressure may briefly return 0 on first reads; agent guards against bogus values.

# 7) Security Hardening (baseline)

* Protect MQTT with login and encryption (TLS) for production.
* Password‑protect the dashboard admin area.
* Keep network access to what’s needed (web + dashboard).

# 8) Bill of Materials (current)

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Part | Qty | Notes |
| SBC | Raspberry Pi 3B/4B | 1 | Small computer |
| Sensor | Raspberry Pi Sense HAT | 1 | Temperature, humidity, etc. |
| Storage | microSD 32 GB | 1 | Stores software and data |
| PSU | 5 V / 3 A | 1 | Stable power supply |
| Case | Vented case | 1 | Keeps things tidy and cool |

# 9) Repository & Files

* Source code folder (agent).
* Dashboard flow file (Node‑RED).
* Docs/images folder (screenshots, this document).
* Read‑me with quick start steps.

# 10) Risks & Open Items

* Calibrate sensors if needed (temperature drift with heat).
* Ensure correct device time for aligned timestamps.
* Agree final Modbus‑TCP register list if used.

# 11) Next (as needed)

* Store‑and‑forward buffer (SQLite) for offline periods.
* Service unit for Modbus‑TCP server.
* TLS + credentials for MQTT in production.

# Placeholders to Fill

* docs/images/01\_hardware\_hero.jpg (cover)
* docs/images/SenseHAT\_Architecture.png (architecture)
* docs/images/02\_nodered\_dashboard.png (dashboard screenshot)