Hardware report/Build instructions

Student Name: Dhairya Pal **Student Number: N01576099**

Table of Contents

1.0 Introduction

This project utilizes a Raspberry Pi development board as the central controller within a modular sensor integration system. It forms the backbone of the broader Computer Systems Project – Phytoplankton Air Systems, which aims to innovate air purification technologies for urban environments.

The Phytoplankton Air Systems prototype employs four distinct sensors to continuously monitor the condition of a phytoplankton solution – an algae-based medium chosen for its potential to produce oxygen and improve air quality. These sensors feed data into the Raspberry Pi, allowing for dynamic tracking, analysis, and presentation of environmental parameters crucial to sustaining the phytoplankton's health.

2.0 Added functionality

The project architecture centers around a Raspberry Pi, which serves as the main processing unit and temporary data storage. Core hardware components include – Raspberry Pi Sense HAT (to provide and intuitive, real-time visualization of sensor readings. Changes in carbon dixoxide concentrations are dynamically displayed, enhancing usability and interpretability.),

Custom-designed PCB (engineered and soldered to interface directly with the ENS160 sensor. This PCB is strategically positioned – sandwiched between the Raspberry Pi and *the Sense HAT – and electrically connected through a standard 40-pin header. This configuration supports modularity, maintains a compact form factor, and ensures reliable data flow between all components, and*

SparkFun ENS160 Sensor breakout board (more about this in section 2.1).

Together, these elements enable real-time sensing, local-visualization, and synchronized data handling within a streamlined hardware stack.

2.1 Sensor/Effector purchase

To enable accurate air quality monitoring within the embedded system, several key components were sources and tested for compatibility:

- ***SparkFun ENS160 (Breakout Board)**: Chosen for its high sensitivity in detecting air pollutants, particularly CO2 concentration. Initial sourcing attempts were made via DigiKey – the primary supplier for this project's components – but due to stock unavailability, the sensor was ultimately purchased from Amazon. The Raspberry Pi and Sense HAT were provided by Humber as part of the CENG153 course in the Computer Engineering Technology program.*
- ***Qwiic Connector System:** Implemented to streamline the wiring process between the ENS160 sensor and the custom PCB. The female Qwiic connector and Grove-compatible cable were procured from DigiKey. These components enable plug-and-play functionality post-soldering, significantly reducing setup complexity and minimizing the potential for wiring errors.*
- ***40-pin header:** This header facilitates a reliable physical and electrical interface between the Raspberry Pi and Sense HAT, ensuring seamless communication across all modules.*

2.2 PCB design and soldering

A custom PCB was designed to serve as an intermediary layer between the Pi and external components – namely the sensor and Sense HAT. The design began with a comprehensive schematic in KiCad, an open-source PCB design software. This schematic incorporated all essential components, including two resistors (of resistance 220 ohms and 2.2k ohms), a capacitor, an LED, a 3-pin female connector for the sensor, holes for a 40-pin GPIO header, and additional connectors as required. Once finalized, the schematic was translated into a PCB layout where precise component placement and electrical routing were determined.

Signal paths were optimized to minimize noise interference and ensure stable power delivery to the ENS160 module. Following the PCB fabrication, assembly was carried out using lab-grade soldering equipment. Components such as the connectors, LED, resistors, and capacitor were carefully soldered onto the board to maintain electrical integrity and mechanical stability.

2.3 Case design and assembly

A custom enclosure was created designed to house Raspberry Pi, PCB, and sensor hardware. Using the open-source vector design tool InkScape, the case was tailored to the precise dimensions of all three components, including cutouts for the Raspberry Pi's ports and ventilation zones. The enclosure safeguards internal hardware while maintaining full port accessibility and supporting adequate airflow for reliable environmental testing. The final design also facilitates modularity, allowing easy disassembly and reconfiguration for future upgrades.

2.4 Firmware development and use

All essential modules and Python libraries required for sensor communication were installed, including I2C development tools, the Firebase Admin SDK, and custom scripts to interface with the Sense HAT and SparkFun ENS160 sensor. Initial testing was performed using commands such as i2cdetec -y 1 to verify successful device recognition on the I2C bus before progressing to more complex scripting.

The system leverages the I2C protocol for real-time data acquisition from the ENS160 sensor via the Qwiic interface. A continuous Python loop was implemented to log environmental data in real time. This data is visualized on the Sense HAT display while eing concurrently pushed to a Firebase Realtime Database, ensuring both local and remote monitoring.

Development followed a modular testing approach:

- *Module imports were verified to prevent runtime errors.*
- *Sensor communication was validated by checking live data readings; for example, CO_2 levels responded appropriately to changes in ambient conditions, confirmed by manually inducing airflow near the sensor.*
- *The final implementation employed multithreading to handle three parallel tasks: data acquisition, display updates on the Sense HAT, and structured data upload to*

Firebase fields – each operating independently to maintain performance and responsiveness.

3.0 Testing and Results

Thorough validation was carried out to ensure all system components operate reliably and in tandem. The following testing phases were implemented:

Hardware Recognition and Initialization

Initial detection of the SparkFun ENS160 sensor was verified using the i2cdetect -y 1 command, confirming successful I2C communication.

Module imports for the sensor, Sense HAT, and Firebase connectivity were individually tested to isolate integration issues before full deployment.

Sensor Responsiveness Testing

Functional testing involved reading baseline air quality data and observing changes in CO2 levels by manually influencing ambient conditions (e.g., exhaling near the sensor).

Sensor readings fluctuated consistently in response to atmospheric changes, validating real-time responsiveness and calibration.

Multithreaded Performance Checks

Three-thread architecture was tested:

1) *Continuously acquired data from ENS160*
2) *Updated Sense HAT display to reflect environmental data*
3) *Pushed sensor data asynchronously to the Firebase Realtime database.*

Threads operated concurrently without significant lag or data collisions, showcasing effective resource utilization and thread isolation.

Data Integrity and Cloud Sync

Live data logging to Firebase was tested over extended periods to assess consistency and reliability.

Sensor values were accurately mapped to designated device fields in the Realtime Database, and remained in sync with visual output on the Sense HAT.

Observational Insight

Minor latency was observed during startup due to sensor initialization overhead – mitigated by adjusting loop delay and connection timeout parameters.

The custom PCB successfully facilitated seamless electrical communication and acted as a durable bridge between hardware layers.

4.0 References

Footnotes

1. Technology Report Guidelines. OACETT, Revised September 2022. Available at: https://www.oacett.org/getmedia/5ad707d7-f472-4b24-a7fe-f34e270b0c41/2022_TR_Guidelines_-_Updated_Version_-_Sept_2022.pdf ↩
2. SparkFun Electronics. ENS160 Air Quality Sensor Hookup Guide. Available at: SparkFun ENS160 Guide
3. Raspberry Pi Foundation. Sense HAT Documentation. Available at: Sense HAT Official Docs
4. Firebase Admin SDK Documentation. Google Developers. Available at: Firebase Admin Python SDK
5. Python Software Foundation. Python 3.11 Documentation. Available at: Python Official Docs
6. Digi-Key Electronics. Product Listings and Datasheets. Available at: Digi-Key Homepage
7. IEEE. Guide to Writing Technical Reports. Available at: IEEE Technical Writing Resources