

Monitor and log environmental conditions using low-cost open-source hardware

Onan Agaba^{1*}, Elad Levintal², Devi Sarai Orozco Fuentes², and Thuc Thi Nguyen²

<https://github.com/OnanAgaba/monitor-and-log-environmental-conditions>

¹*French Associates Institute for Agriculture and Biotechnology of Drylands*, ²*Zuckerberg Institute for Water Research, Ben-Gurion University of the Negev, Israel*

1. Introduction

Environmental monitoring relies on long-term measurement of hydrological, meteorological or ecological variables to establish benchmarks, track trends, and inform policy (Chan et al., 2021). Such measurements are important for assessing changes caused by both human activities and natural phenomena such as climate warming. Additionally, environmental monitoring provides essential data for addressing public health concerns that directly affect human well-being and productivity. For example, air quality monitoring in urban areas helps pinpoint emission hotspots or their sources, making it easier to track pollution plumes and assess the influence of localized meteorological conditions (Horsburgh et al., 2019; Thompson, 2016). However, achieving high resolution and real-time environmental data remains challenging in resource-limited areas where deploying industry-grade sensors is impractical or cost prohibitive.

Common environmental monitoring approaches include remote sensing, predictive models, and high-precision industry sensors, among others. While remote sensing approaches such as satellites provide free data at a large scale, they usually lack sufficient spatial and temporal resolution (Mahan & Yeater, 2008). Predictive models, such as atmospheric dispersion models, can fill data gaps in monitoring, but are computationally intensive and limited in accuracy (Chan et al., 2021). On the other hand, industry-grade sensors provide in-situ measurements of variables such as humidity, temperature, carbon dioxide (CO₂), among others, but are expensive to install and maintain at scale. As an alternative, low-cost, do-it-yourself instrumentation is gaining popularity (Kumar et al., 2015), providing high-resolution environmental data at a lower cost (Horsburgh et al., 2019). This project aims to design and implement an open-source system to continuously monitor and log temperature (T), relative humidity (RH), and carbon dioxide (CO₂) levels in an indoor environment over three days. By using low-cost open-source hardware and sensors, the system provides a practical solution for environmental data acquisition inside a caravan.

2. Experimental Design

The environmental monitoring system consisted of four main components: a microcontroller, an SHTC3 sensor for temperature and humidity, an SCD-40 sensor for CO₂, temperature, and humidity, and a microSD card for data storage. The Arduino board served as the central microcontroller which coordinated all connected electrical components.

2.1 Wiring and Assembly

The SHTC3 sensor and microSD card breakout board were connected to the Arduino using jumper wires soldered to the appropriate pins (see Figure 2). The SCD-40 was connected through a STEMMA QT cable with male connectors. In total, four wires from each sensor and six wires from the microSD card breakout board were connected to their designated pins on the Arduino board. Once all components were wired, the system was powered through a USB cable from the computer, which ensured continuous operation. An operational sketch of the monitoring system setup is shown in Figure 1a.

2.2 Programming and Logging

The pseudocode (C++), developed based on the project's flow chart (see Figure 1b), was compiled and uploaded to the Arduino board using the Arduino IDE 2.3.4. The system was programmed to measure and record temperature, relative humidity, and CO₂ levels every 10 seconds. Each data entry was timestamped using the Arduino's internal millis() function, displayed on the serial monitor, and logged to the microSD card for three consecutive days.

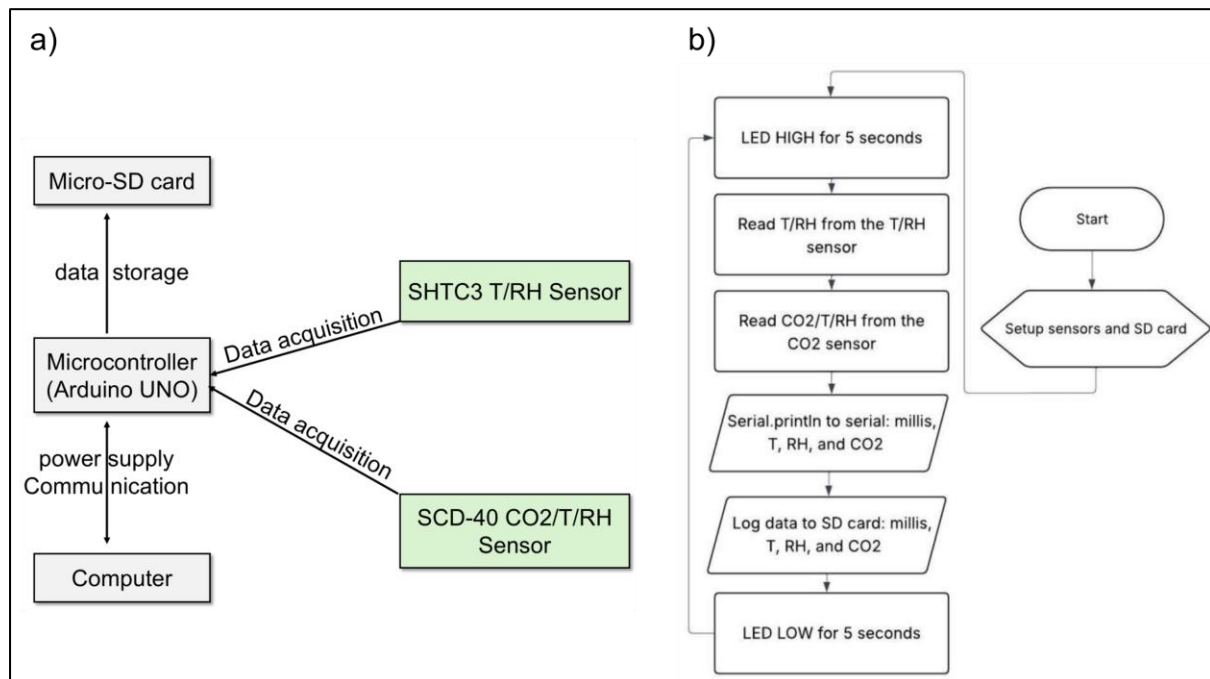


Figure 1: Illustrations of general experimental design; a) operational sketch of the entire system, b) the flow chart followed to develop a code

2.3 Bill of Materials

Table 1: Summary of hardware components used in this study

Component	Part Type	Description	Cost	Reference/Link
Arduino UNO	DFROBOT DFRduino UNO V3.0 [r3]	microcontroller board was used	\$12.90	Arduino Uno
MicroSD Module	SparkFun - microSD Transflash	SD card interface module	\$5.95	SD Breakout
SD Card	SanDisk Ultra HC1	Storage for logged data	\$15	SD Card
SHTC3 T/RH	Adafruit SHTC3 STEMMA QT	Temperature and humidity sensor (I ² C)	\$6.95	SHTC3
SCD-40 CO ₂ /T/RH	Adafruit SCD4x	CO ₂ , temperature and humidity sensor (I ² C)	\$44.95	SCD40
Cables		Jump wires and a USB cable for power and communication	\$6.98	Wires

2.4 System Overview

The system uses an Arduino Uno with SHTC3 and SCD-40 sensors to continuously monitor temperature, relative humidity, and CO₂ levels in a room and log on a microSD card.

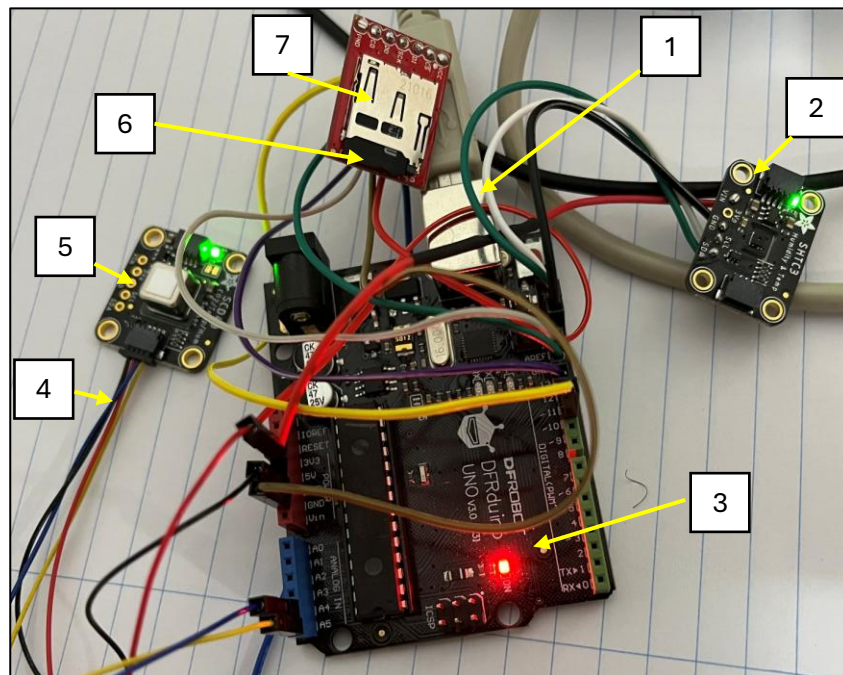


Figure 2: System components; 1) power into the board through the USB cable, 2) SHTC3 sensor, 3) Arduino UNO, 4) jump wires, 5) SCD-40, 6) microSD card, and 7) microSD card breakout module.

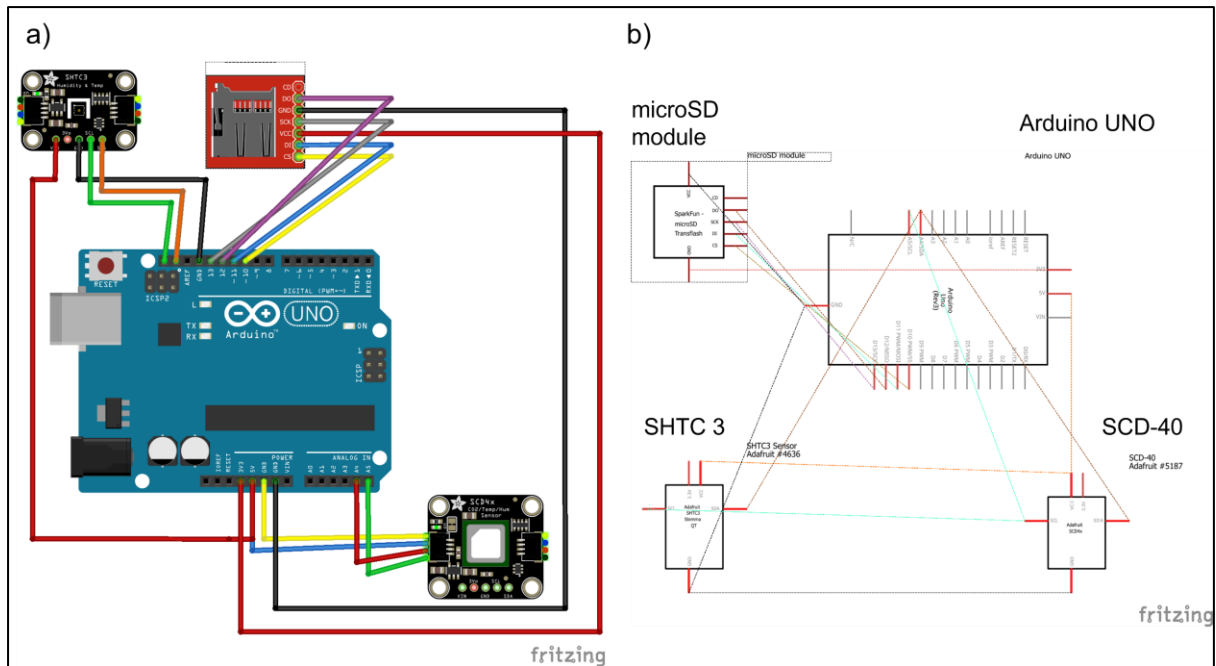


Figure 3: illustration of system connection layouts a) connection of SD card module and other sensors to the Arduino UNO board, b) card layout of connections

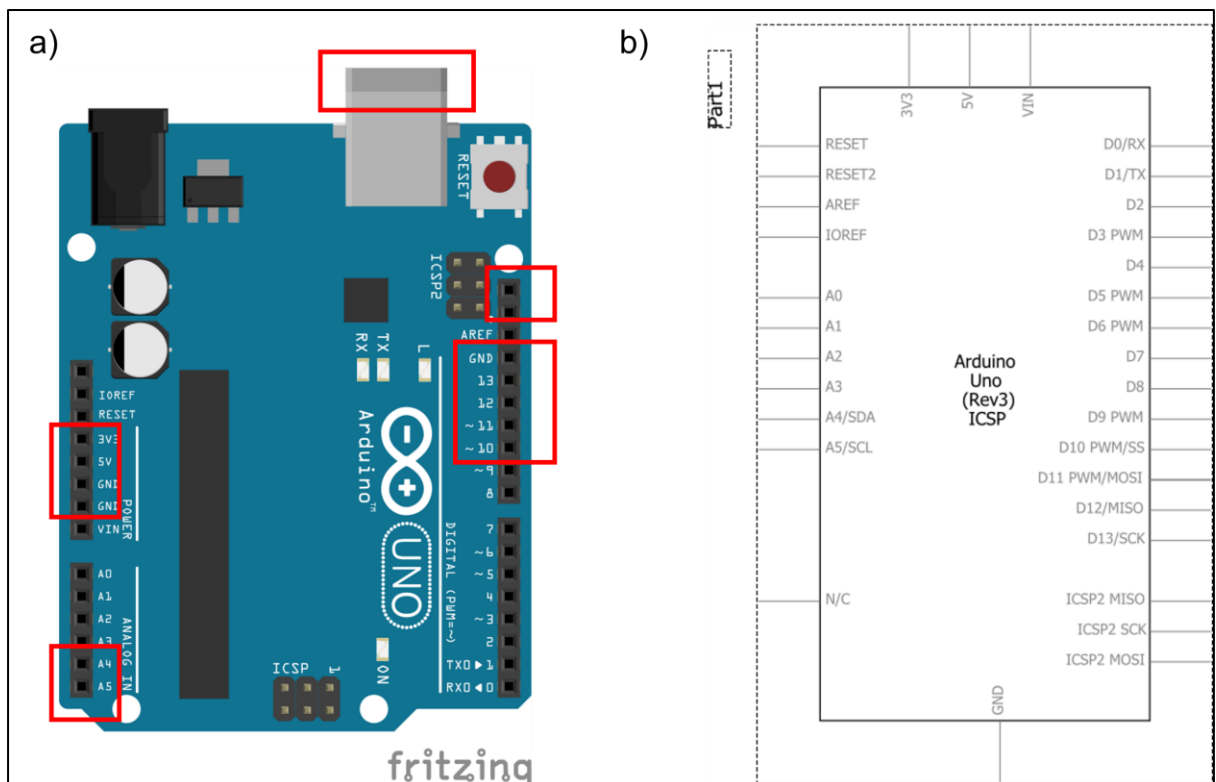


Figure 4: The anatomy of an Arduino board and the ports used to make complete connection

Table 2: Port numbers used and their purposes

Port	Function
A4, A5	SDA and SCL pins for I ² C communication with peripheral devices
10,11,12,13	CS, MOSI, MISO, and CLK data pins for Serial Peripheral Interface communication with peripheral devices such as a microSD card or real-time clock
3v3, 5v, and GND	3v3 and 5v is regulated power for peripheral devices and GND provides the grounding for the whole circuit

3. Results

Measurements began on May 14, 2025, at 22:00 pm for three consecutive days. CO₂ levels are seen to increase especially during periods of occupancy. Temperature measurements from both sensors showed the same trend whereas relative humidity varied differently for both sensors. SHTC3 showed a similar trend as SCD40 but had much higher values of relative humidity.

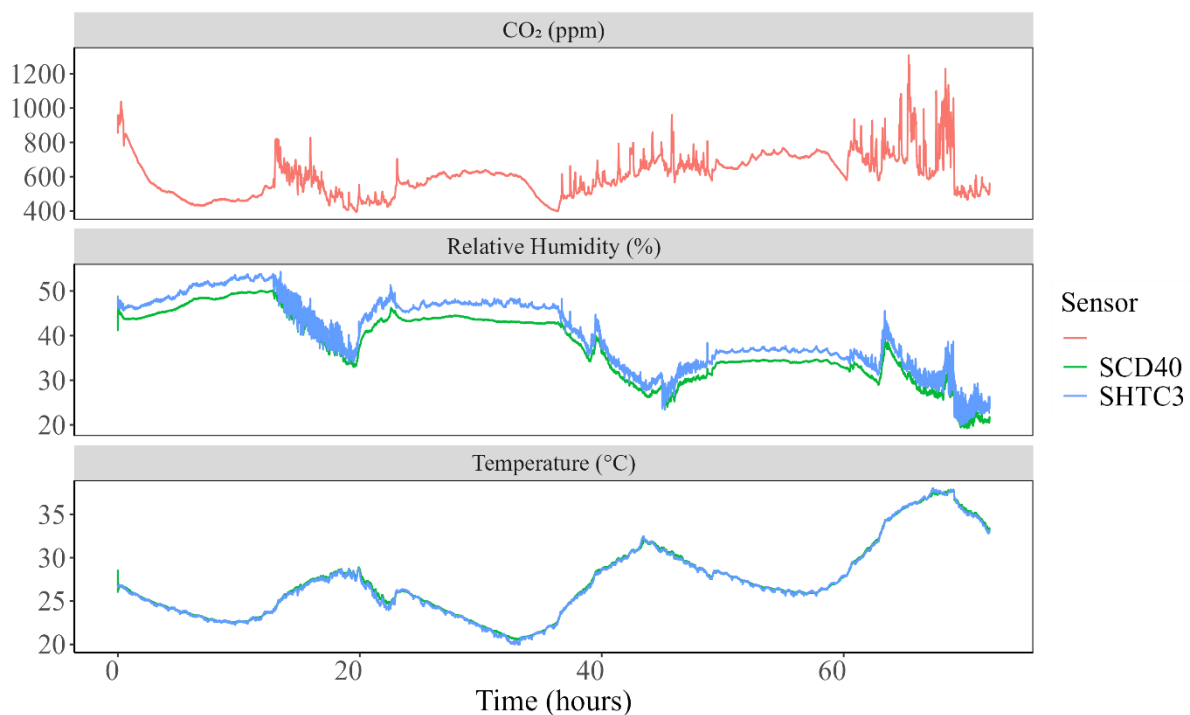


Figure 5: Results from the two sensors over the course of three days

4. Conclusion

The project demonstrates the feasibility and effectiveness of using open-source hardware for environmental monitoring in an indoor setting. The use of Arduino and widely available sensors such as SHTC3 and SCD-40 enables affordable and scalable solutions for both educational and practical applications. The absence of an RTC and enclosure simplifies the

design but limits the ability to correlate measurements with real-world time and may expose components to environmental stress. Future improvements could include integrating a real-time clock, adding wireless data transmission, and implementing sensor calibration for enhanced accuracy.

References

1. Chan, K., Schillereff, D. N., Baas, A. C., Chadwick, M. A., Main, B., Mulligan, M., O'Shea, F. T., Pearce, R., Smith, T. E., & Van Soesbergen, A. (2021). Low-cost electronic sensors for environmental research: Pitfalls and opportunities. *Progress in Physical Geography: Earth and Environment*, 45(3), 305-338.
2. Horsburgh, J. S., Caraballo, J., Ramírez, M., Aufdenkampe, A. K., Arscott, D. B., & Damiano, S. G. (2019). Low-cost, open-source, and low-power: But what to do with the data? *frontiers in Earth Science*, 7, 67.
3. Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., Bell, M., Norford, L., & Britter, R. (2015). The rise of low-cost sensing for managing air pollution in cities. *Environment international*, 75, 199-205.
4. Mahan, J. R., & Yeater, K. M. (2008). Agricultural applications of a low-cost infrared thermometer. *Computers and Electronics in Agriculture*, 64(2), 262-267.
5. Thompson, J. E. (2016). Crowd-sourced air quality studies: A review of the literature & portable sensors. *Trends in Environmental Analytical Chemistry*, 11, 23-34.