Adriann Liceralde
Dr. Brian McPherson
Carbon Science Research at the University of Utah
Wireless Sensor Network Project
Last Updated February 24, 2022

Assessment of the SCD30 CO_2 Sensor

I. Introduction

There are a variety of carbon dioxide (CO_2) sensors that differ in important factors such as cost, power consumption, accuracy, and precision. Generally, highly accurate and more precise instruments consume more power and are expensive. Consequently, cheaper sensors have lower quality in their data measurements. The importance of each factor depends on the application. For example, if the device must operate from battery power, then power consumption has significance in determining which sensor to choose.

The cost of CO_2 sensors spans anywhere from around $50 to thousands of dollars. Although expensive high-end analyzers are likely better, low-cost simple analyzers are still advantageous. If used in the proper application, a low-cost sensor's inexpensive and low power consumption can be comparable to a high-end instrument.

One model of a low-cost sensor is the SCD30 made by Sensirion with a price tag of under $100. The device draws small amounts of power and is only about a square inch in size. Although these aspects are attractive, the downside is the rated accuracy of ±(30 ppm + 3% of the measured value). In comparison, the GasHound LI-800, a high-end CO_2 analyzer, has a better accuracy of ±(3% of the measured value) but consumes over 50 times the amount of power than the SCD30. With the large gap in the rated accuracies, there is likely a difference in data quality. Therefore, experiments using the SCD30 and the LI-800 are necessary to assess a low-cost sensor's performance.

II. Equipment:

a. Low-cost sensor

The Sensirion SCD30 uses nondispersive infrared (NDIR) technology to measure CO_2, temperature, and humidity. Any microcontroller, such as an Arduino Uno, can control this device. The gas measurement range is 0 to 40,000 ppm with a sampling rate of 2.1 seconds. On average, the current draw is 19 mA at a voltage of 5 V. The repeatability is ± 10 ppm with an accuracy of ±(30 ppm + 3% of the measured value). The sensor operates passively, thereby not requiring any pumps to intake gas.

b. Sensor Multiplexer

Four SCD30 devices are hooked up to an Arduino Nano in the following experiments. Having four copies of the same sensor model allows more information to be gathered from the experiments. For example, if one device exhibits behavior that differs significantly from the others, that could indicate a poor sensor. The multiplexer circuit consists of the four SCD30 devices, a real-time clock, an I^2C multiplexer, a logic converter, and an Arduino Nano. Data is collected from all devices simultaneously. Below is the wiring diagram and the schematic for the multiplexer:

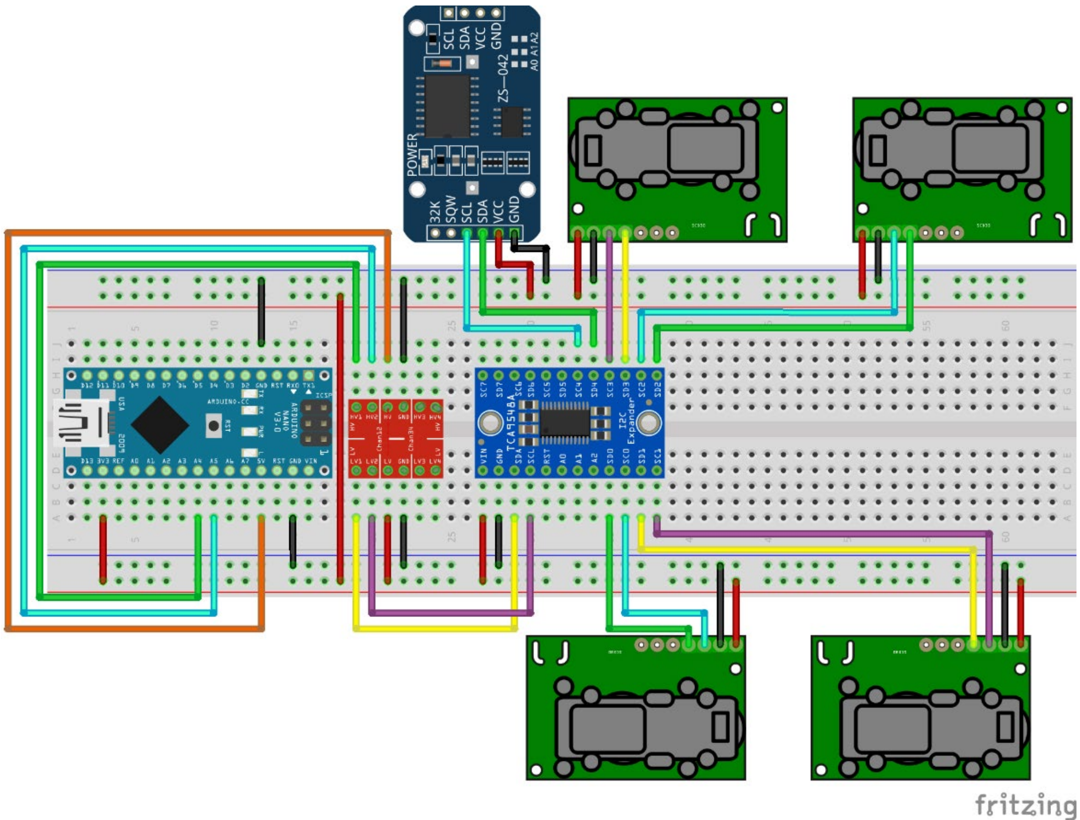

Figure 1. Breadboard diagram of the Sensor Multiplexer.

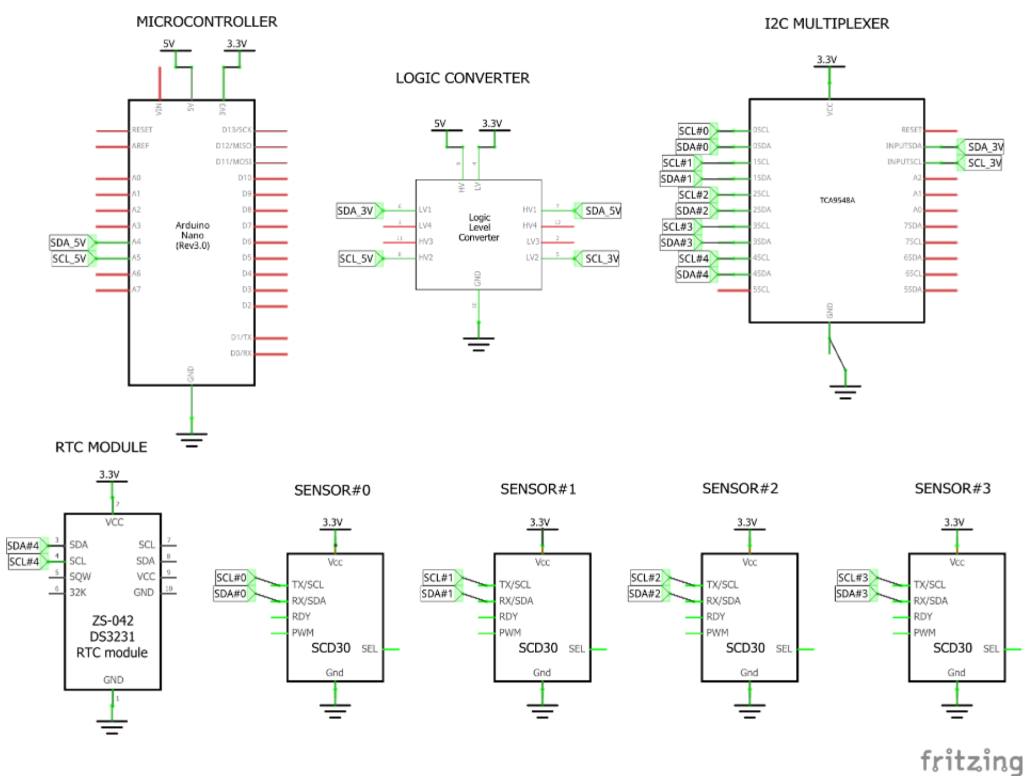

Figure 2. Circuit diagram of the Sensor Multiplexer.

c. High-end sensor

An expensive CO_2 gas analyzer is the GasHound LI-800, which uses NDIR technology. Typically, a computer is connected to the instrument to control it and extract data from it. However, an Arduino Uno is capable of retrieving measurements from the device. The LI-800's measurement range is limited to only 0 to 2,000 ppm, but the sampling rate can be as fast as 0.5 seconds. Because of its complexity and the requirement of a pump to intake gas, the current draw is generally 300 mA at a voltage between 12 V and 30 V.

The GasHound was calibrated using gas cylinders with known concentrations before the experiments. Therefore, the measurements given by this high-end gas analyzer are assumed to be the correct readings, thereby granting the LI-800 100% accuracy. The purpose of this is so that the low-cost SCD30 sensors have a baseline to compare their measurements with.

d. CO2 Gas Cylinder

Assessing the performance of both types of sensors requires a stable supply of carbon. Therefore, the following experiments use a cylinder containing 1500 ppm of CO_2 gas. The cylinder has a pressure regulator controlled by a valve that releases gas through connected tubing. It is reasonably assumed that the gas released from the cylinder is precisely 1500 ppm at all times.

III. Experiment 1: Sudden Spikes

The purpose of this first experiment is to observe how well the SCD30 responds to a sudden change in concentration. The measurements from the GasHound LI-800 acts as a baseline to compare the cheap sensors with. Below is the experimental procedure.

a. Procedure

1. Hook up the SCD30 sensors into the Multiplex circuit with an Arduino powered by a local computer.
2. Hook up the LI-800 sensor to a second Arduino connected to the same computer. Power the LI-800 to a wall outlet.
3. Place all five sensors in the same open container.
4. Connect a tube from the container to a gas cylinder containing 1500 ppm of CO_2. Leave the cylinder valve closed.
5. Boot up all sensors and ensure that the computer receives all the data correctly.
6. Allow the sensors to reach a steady-state value.
7. Open the cylinder valve for about 10 seconds to introduce a gas spike into the system. Then close the valve.
8. Repeat steps 6 and 7 as desired.

b. Results

The experiment spanned a total of 40 minutes, with six gas injections. The data collected by the Arduinos were analyzed using Python and the Matplotlib library. Figure 3 showcases the concentration versus time plot for Experiment 1; Figure 4 is a close-up of two gas injections to visualize the graphs better.

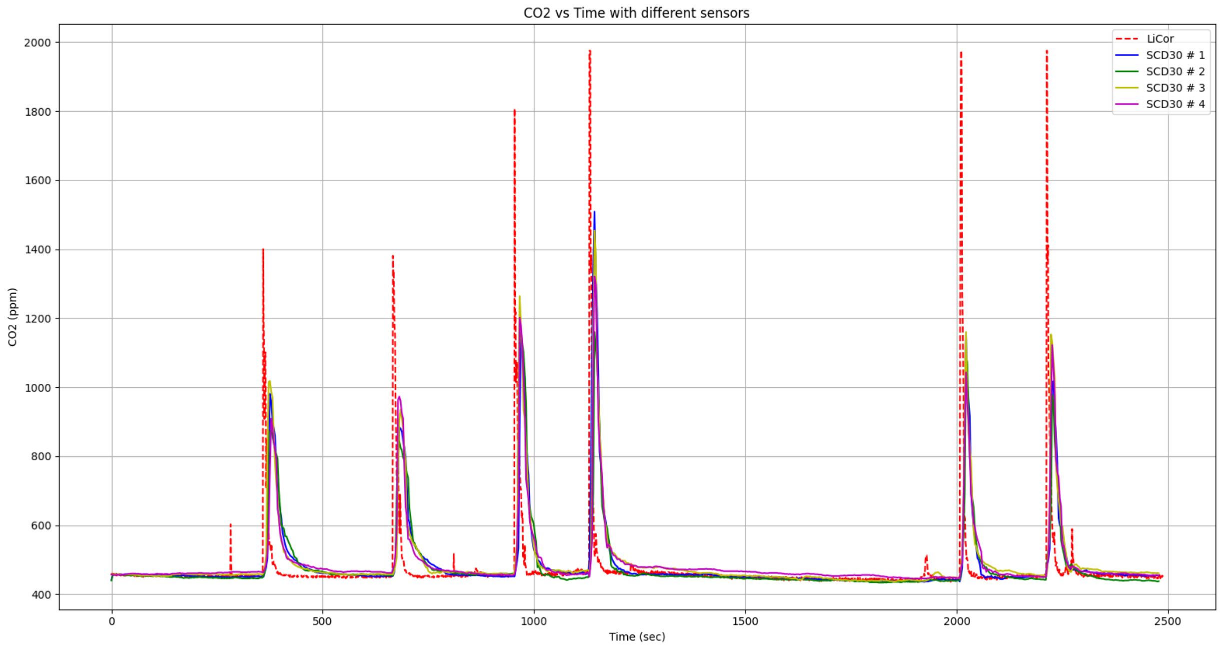

Figure 3. Concentration versus Time plot of Experiment 1.

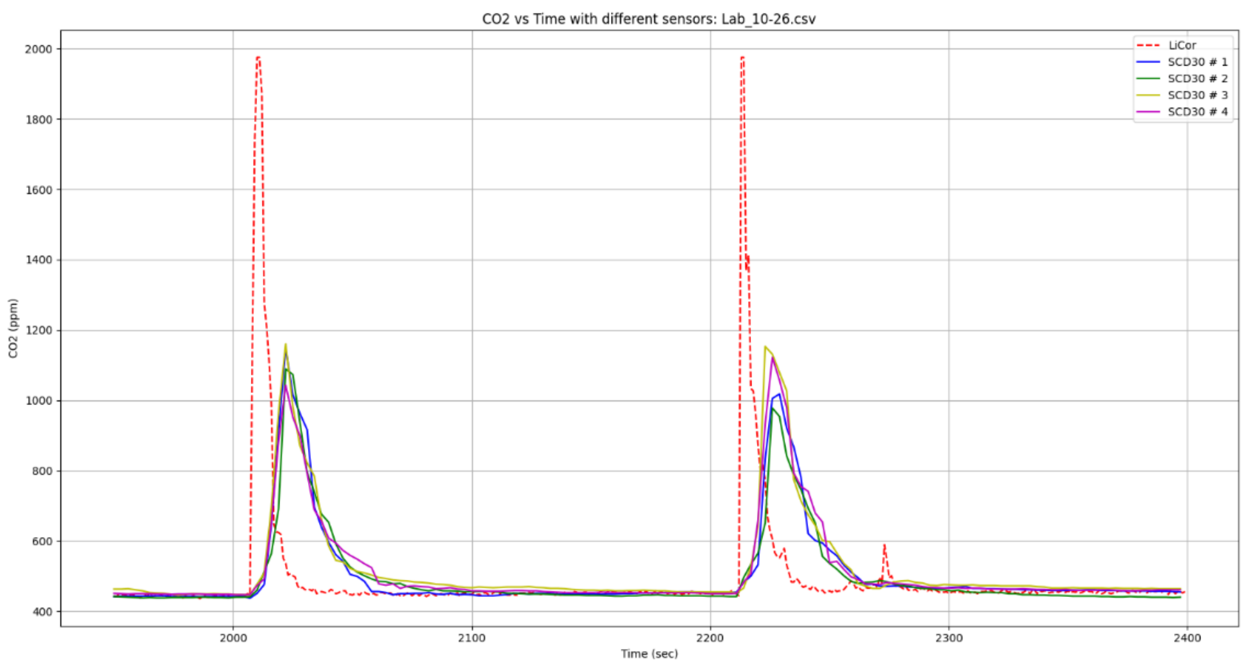

Figure 4. Zoomed-in version of the Concentration versus Time plot of Experiment 1.

c. Analysis

The results from this experiment reveal helpful information about the SCD30s. Below are four key observations:

1. All four SCD30 sensors exhibit similar behavior to one another. When the gas spikes were introduced into the system, the readings of the low-cost sensors spiked almost simultaneously. The values of the peak readings are roughly similar but still vary due

to the rated accuracy of ±(30 ppm + 3% of the measured value). Their readings then return to equilibrium at about the same rate. This observation shows that the response speed among the SCD30s does not vary from copy to copy.

2. The SCD30 model has a noticeable response delay. When each gas spike occurred, the GasHound responded much faster. On the other hand, the SCD30s took a few seconds to respond. Part of this may be attributed to the difference in frequencies. The GasHound can take measurements every second, while SCD30 is limited to taking measurements every 3 seconds. Therefore, analyzers with lower frequencies are also limited in response speed.

3. The GasHound's readings return to equilibrium much faster than the SCD30s'. The GasHound has a sharper and narrower peak, reflecting that the gas injection lasted only a few seconds. It is interesting that according to the expensive analyzer, the 1500 ppm gas was only in the system for a short amount of time. In contrast, the four SCD30s report that the gas injection remained in the system for twice as long, as shown by how its peaks are wider than the GasHound's.

4. There was a slight gas spike that only the GasHound could detect at about time = 2280 seconds. Still working under the assumption that the GasHound has 100% accuracy, this means that the SCD30s were unable to detect a slight change in concentration. This most likely occurred because the expensive sensor has a higher sensitivity due to its better technology.

IV. Experiment 2: Continuous Spikes

The purpose of the following experiment is to assess the low-cost sensor's ability to detect and hold a change in concentration for long durations. For example, if a sudden change in concentration caused a new steady-state value to arise, then the sensor should show a steady-state value, as well.

a. Procedure

1. Hook up the SCD30 sensors into the Multiplex circuit with an Arduino powered by a local computer.
2. Hook up the LI-800 sensor to a second Arduino connected to the same computer. Power the LI-800 to a wall outlet.
3. Place all five sensors in the same closed container.
4. Connect a tube from the container to a gas cylinder containing 1500 ppm. Leave the cylinder valve closed.
5. Boot up all sensors and ensure that the computer receives all the data.
6. Allow the sensors to reach a steady-state value.
7. Open the cylinder valve and leave it open for about 15 minutes to continuously introduce a gas spike into the system.
8. Allow the system to reach a new steady-state value.
9. Close the cylinder valve and let the system return to its initial steady-state concentration.
10. Repeat steps 6-9 as desired.

b. Results

Two trials were performed for this experiment, each lasting about 25 minutes. There were no differences in setup or procedure between the two trials. Below are Figure 5 and Figure 6 that shows the concentration versus time plots for Experiment 2.

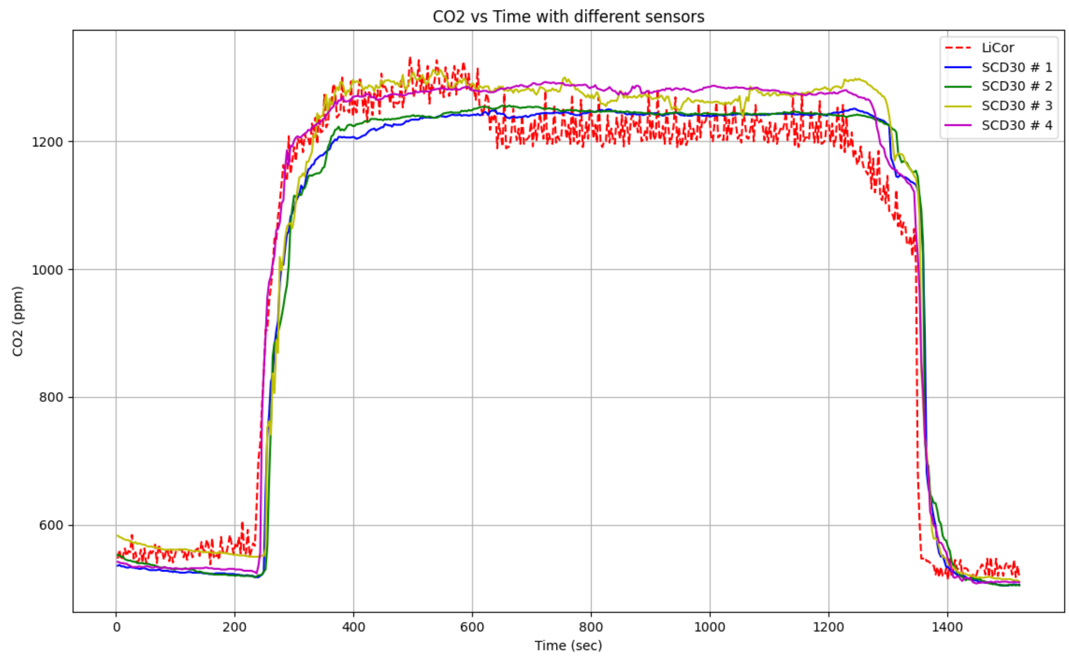

Figure 5. Concentration versus Time plot of Experiment 2 – Trial 1.

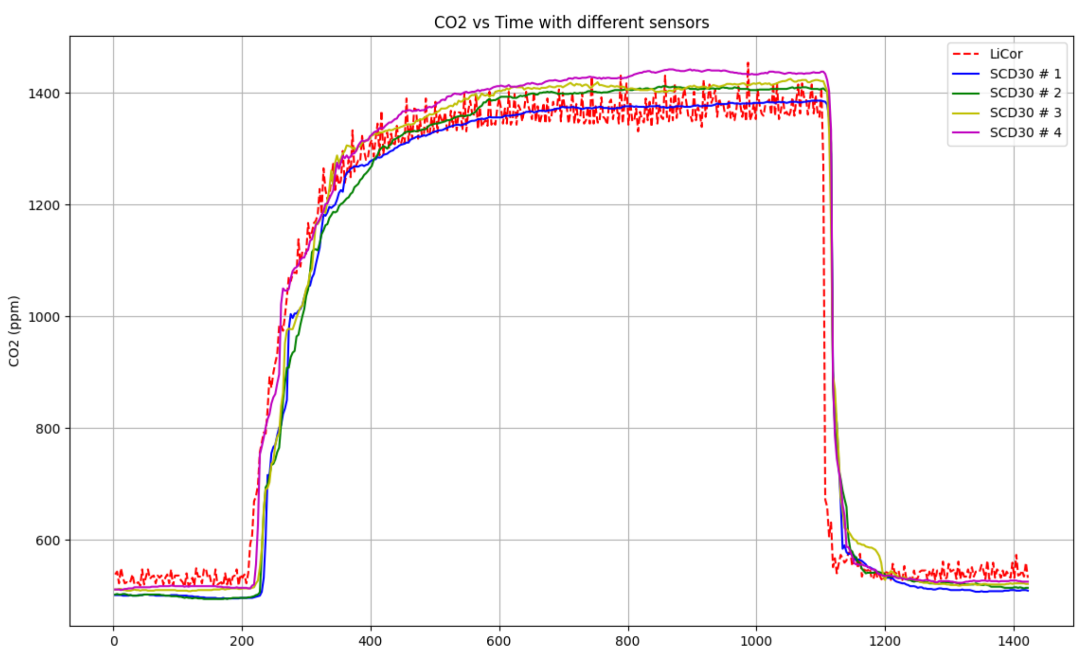

Figure 6. Concentration versus Time plot of Experiment 2 – Trial 2.

c. Analysis

The results from the second experiment reveal additional information that helps determine the reliability of the SCD30. The following observations are seen from these results:

1. Like in Experiment 1, the four low-cost sensors exhibit similar behavior to each other. This result corroborates the assertion that the response speed among the SCD30s does not vary significantly from copy to copy.

2. In addition to the low-cost sensors behaving similarly, their measurements also reflect the same trend as the GasHound. Unlike in Experiment 1, the graphs of the four low-cost sensors in Experiment 2 align well with the GasHound's graph. The likely reason for this occurrence is that the gas injection was continuous rather than a pulse. The gas pulse caused the GasHound to have sharp peaks in the first experiment, contrasting with the SCD30s' broad peaks. Conversely, a continuous injection in the second experiment allowed both types of sensors to record similar behavior.

3. The GasHound responds faster to concentration changes, whether an increase or decrease in concentration. The faster response can be seen clearly at t = 1200 seconds in Figure 5 and t = 1100 seconds in Figure 6. Again, these results corroborate Experiment 1, where the GasHound has faster response times.

4. Interestingly, at time t = 600 seconds in Figure 5, the GasHound reading drops, whereas the low-cost sensors remain at a steady state. The fact that only the GasHound detected this slight drop in concentration highlights the higher sensitivity of the GasHound. This occurrence also provides evidence that low-cost sensors may not be powerful enough to detect small changes in concentration.

V. Conclusion

Low-cost CO_2 sensors are limited in their capabilities due to their small size and simpler technology. The experimental results show that the Sensirion SCD30 pales in comparison to the expensive high-end GasHound LI-800. The latter responds faster to changes in concentration and seems to reflect a more accurate representation of the actual gas concentrations. In addition, the GasHound was capable of detecting slight gas spikes that were undetectable to the SCD30s. Despite the GasHound having higher data quality, the low-cost SCD30s performed reasonably well given their cheap design. The measurements of all four SCD30 sensors were similar to that of the LI-800 sensor, especially during moments of gas injections. The SCD30 quickly detected significant changes in concentration, and the measurements were accurate to the LI-800 if the concentration was held steady at a higher value than the pre-injection value. Therefore, the Sensirion SCD30, despite having low accuracy and precision, performs adequately well for applications where readings significant changes in concentration are desired.

Important Links

- Sensirion SCD30 Product Information: https://sensirion.com/products/catalog/SCD30/
- SCD30 Guide and Code: https://github.com/RiceAllDay22/CSR_Arduino_Collection/blob/main/Individual_Modules/SCD30/Guide_SCD30.pdf
- GasHound LI-800 Manual: https://www.GasHound.com/documents/3a029cfc7eyih2ryt8x1
- GasHound LI-800 Guide and Code: https://github.com/RiceAllDay22/CSR_Arduino_Collection/tree/main/Individual_Modules/GasHound_LI-800
- SCD30 Multiplexer Guide and Code: https://github.com/RiceAllDay22/CSR_Arduino_Collection/tree/main/Miscellaneous_Modules/MultiSCD30

Contact

For any questions or assistance, email Adriann Liceralde at adriann8399@gmail.com