CS 437: Internet of Things
Final Project Report

Team

- Eric Schrock (ejs9)
- Devin Schmitz (devinms3)

Overview

We created a distributed network of CO2 sensors that report back via wifi to a central hub. The hub exposes a web interface on the LAN for accessing the data. This allows us to track variations in CO2 levels throughout a home and over time. This data could be used to understand and improve indoor air quality, leading to healthier lives.

Motivation

todo: "Talk about the importance of this problem. Pretend you are trying to convince someone to give you funding, or purchase what you developed. You may also want to give references/citations here."

todo: Rehash motivation section from the proposal. Maybe remove "low energy" as a design goal.

Technical Approach

Our overall system consists of a base station connected to a network of sensors over wifi. The base station is a Raspberry Pi 4B running a server. Sensors connect to the server and periodically report CO2 measurements. The base station then presents those measurements on a website available on the local area network.

Each sensor consists of a Raspberry Pi Pico WH connected to a CO2 sensor over I2C. Each sensor runs a client that is responsible for connecting to the server on the base station, reading CO2 measurements, and sending those measurements to the base station.

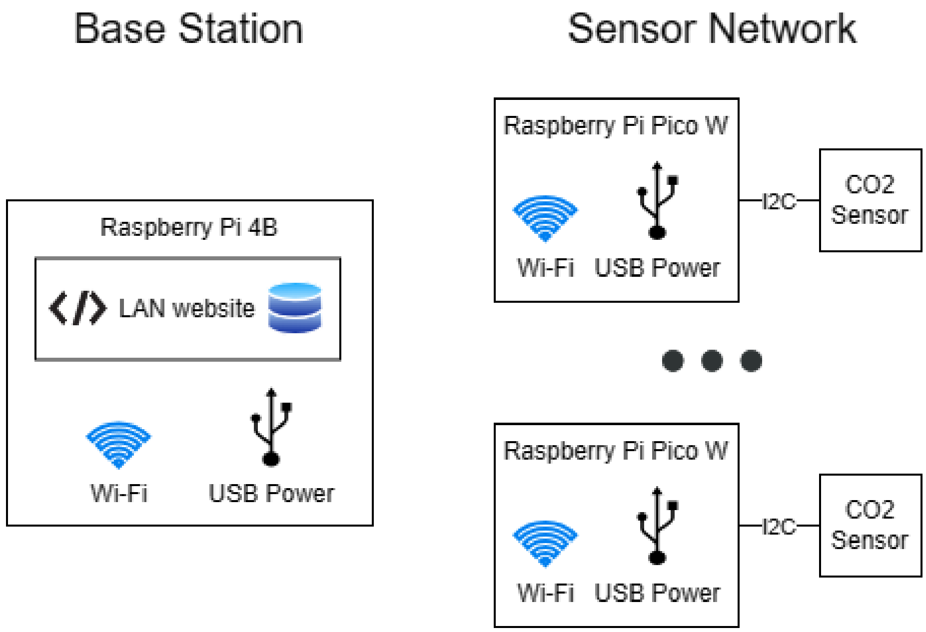

Figure 1: System Architecture

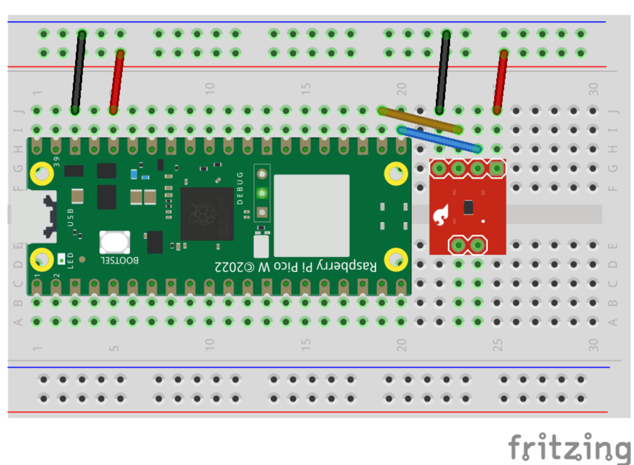

Figure 2: Sensor Layout

Note: The CO2 sensor shown in the Fritzing diagram above does not match the actual part and is only meant to give a general sense of the design.

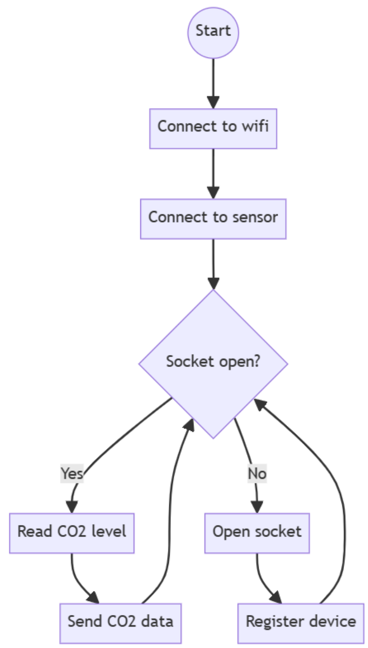

Figure 3: Sensor Flowchart

todo: Add a high-level diagram of the server code (details belong in the next section)

todo: Add a high-level diagram of the website code (details belong in the next section)

Implementation Details

todo: "This is where you give the details in your implementation. Talk about specific software packages you used, hardware modules, any algorithms or research papers you referred to, data structure and protocol choices, etc. You should provide at least an informal list of citations of all these external materials that went into your project."

Ideas

- Pictures of the actual sensors (Devin's and Eric's versions)
- Picture of the hub
- Links to websites of parts and technologies used
- Talk through how we chose the uC (Pico vs Arduino) and CO2 sensor

(CO2 vs eCO2) [1]

- Talk through how we chose a Raspberry Pi Pico dev language (Python vs C)
- Talk through how we chose the networking technology (wifi over BLE)
- Talk through how we chose the web technology (Flask?)

Results

todo: "So, how did things turn out? You can provide performance results, experiences you had interacting with it, etc. Also talk about what the takeaway is - why should we care about your results? And, it is ok for things to go wrong - what did not go right in your project, what was hard and what lessons did you learn?"

Ideas

- Talk about price (compared to CO2 sensor mentioned in the motivation section) (see the BOM in the repo README)
- Talk about productionization
  - Refactor into wall wart with a custom PCB and protective case
  - Solution for configuring IP address of server in each sensor
  - Drive down price with economies of scale and resource usage optimization (e.g. could use a cheaper Pi with less RAM for the hub)
- Talk about changing our design from BLE to wifi
  - Since sensors are wall powered instead of battery powered, power usage was not as high a priority as we initially thought in our proposal
  - The wifi connection was easier to implement in code and we suspect it will be more user friendly as well
- Talk about proposal timeline vs reality
  - Initial design took longer than expect? (lots of options to choose between)
  - HW bring up went more smoothly than expected and no 2nd prototype was needed
  - Etc.

CO2 Data Analysis (Eric)

One question we had was whether our sensors would be accurate enough to provide useful data. They very much were! For example, below are CO2 readings from my home from December 4th, 2023. The data does not start until 11 AM, but you can see the bedroom CO2 levels dropping from their nighttime highs. You can also see the CO2 levels in the kitchen jump between 12 PM and 1 PM when my wife and I ate lunch together.

todo (Eric): Update with full day of data. Add any notes on data after 9:30 PM.

[1] https://electronics360.globalspec.com/article/17986/what-are-eco2-sensors

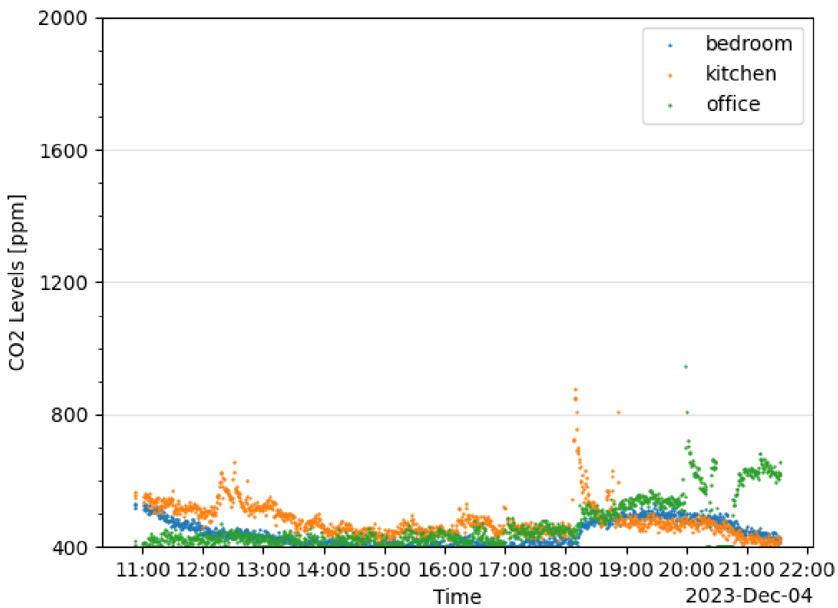

Figure 4: CO2 Data from Eric's Home (12/4/23)

Overall, the CO2 levels stayed fairly low unless we were both in the same room. During the afternoon, you can see that the bedroom CO2 levels were the lowest, followed by the office, and then the kitchen. I spent most of the afternoon sitting at my desk in the office and my wife spent large chunks of her afternoon moving around the kitchen or the adjoining living room. We breath out more CO2 when we are active, so it makes sense that the kitchen CO2 levels were higher than the office. Additionally, she was doing things like vacuuming and cleaning that likely stirred up TVOCs into the air, influencing the eCO2 calculation in the sensor.

At around 6:15 PM, the kitchen CO2 levels spiked when we vented the pressure cooker and then dissipated to the rest of the house, leaving every room at a higher CO2 level. From 6:30 - 7 PM, the kitchen CO2 levels again spiked as we ate supper together. After supper, my wife left for time with friends and I returned to my office to work on this report. As a result, the kitchen CO2 levels dropped below those of the office.

At about 8 PM, I ran a paper shredder right next to the office CO2 sensor. The spike in the reported CO2 is probably another example of TVOCs driving the CO2 estimate. Soon after that, I opened the office window briefly at 8:15 PM and then for 15 minutes at 8:30 PM. This caused the office CO2 readings to peg to 400 ppm, which is the minimum value the sensor can report. This lines up well with the estimated atmospheric CO2 baseline of 410 ppm.

Overall, the sensor readings lined up well with our activities, giving me confidence that the data is accurate enough to drive decision making (e.g. when to open a window). Additionally, I was pleasantly surprised to see that, on average, the CO2 levels in my home stayed well below the recommended max of 800 ppm.

CO2 Data Analysis (Devin)

todo (Devin): Talk about what you can see in your data. Perhaps also compare your data to mine.

Ideas

- sleep cycles
- wake times
- movement between rooms
- opened windows and doors
- average levels
- etc.

Demo Videos

We recommend watching these videos on a large screen and/or setting your video player to HD/1080p.

todo: Record demo videos (we both need to contribute)

Ideas

- Eric?: Demo sensor HW and bring up (talk through HW design and demo hello.py)
- Eric? (prettier sensors): Demo sensor installation and connection to the server (talk through sensor LED transitions and show server and data logs)
- Devin?: Demo website with live and historical data. Show some of the interesting trends observed in the data.

Project Repository

https://github.com/EricSchrock/co2-monitor

What We Learned (move to results section?)

Eric

todo: Add a paragraph on things learned. Compare to the things you expected to learn in the project proposal.

Ideas

- Using Python for Embedded applications
- Refresher after years without soldering
- Networking via the Python sockets lib
- Web development using Flask
- Using the Raspberry Pi Pico WH
- Importance of CO2 levels to health
- Configuring Linux startup scripts

Devin

todo: Add a paragraph on things learned. Compare to the things you expected to learn in the project proposal.

Conclusion

In conclusion, we accomplished the high level goal of our project, to prototype a distributed, networked, and affordable CO2 monitoring solution. Along the way, we tried new technologies, built new skills, and gained a deeper understanding of the quality of the air in our homes and of how it impacts our health.