PicoAirSense

An Indoor Air Quality Monitor by Adrien Abbey

A Brief Introduction

The Inspiration and the Goal

Inspiration

- I live in an apartment of dubious air quality.
- My inner scientist: can I measure my discomfort in a quantifiable way?
- I can get ~~paid~~ graded for this?!

Connecting the Dots

I had something I wanted to know.

I now had the means to do it.

Baby’s First Steps

- The first step was research:
  - What hardware did I need?
  - Could I afford that hardware?
  - What languages does the hardware support?
  - Are there open-source libraries for the hardware?
  - How difficult would this be?
- Easy enough.

Motivation

I has it.

More Than Just a Grade

- Resume builder: look at this awesome thing I built!
- Open-source project with a public GitHub repository
  - Giving back to the community: I use open-source products *all the time*.
  - Getting my name out there: sometimes people get job offers without a single resume.
- Personal growth opportunity.
  - I really *enjoyed* working on this project.
  - I learned a *lot* of new skills too.
  - I can now share these skills with others!
    - The Piano Staircase project and participants will benefit immensely from this.

The Solution

I can build it!

The PicoAirSense: What is it?

An indoor air quality monitor

Measures temperature, humidity, air pressure, eCO_2 and TVOC levels

Hackable: add an e-ink display, use Wi-Fi to expose data, etc.

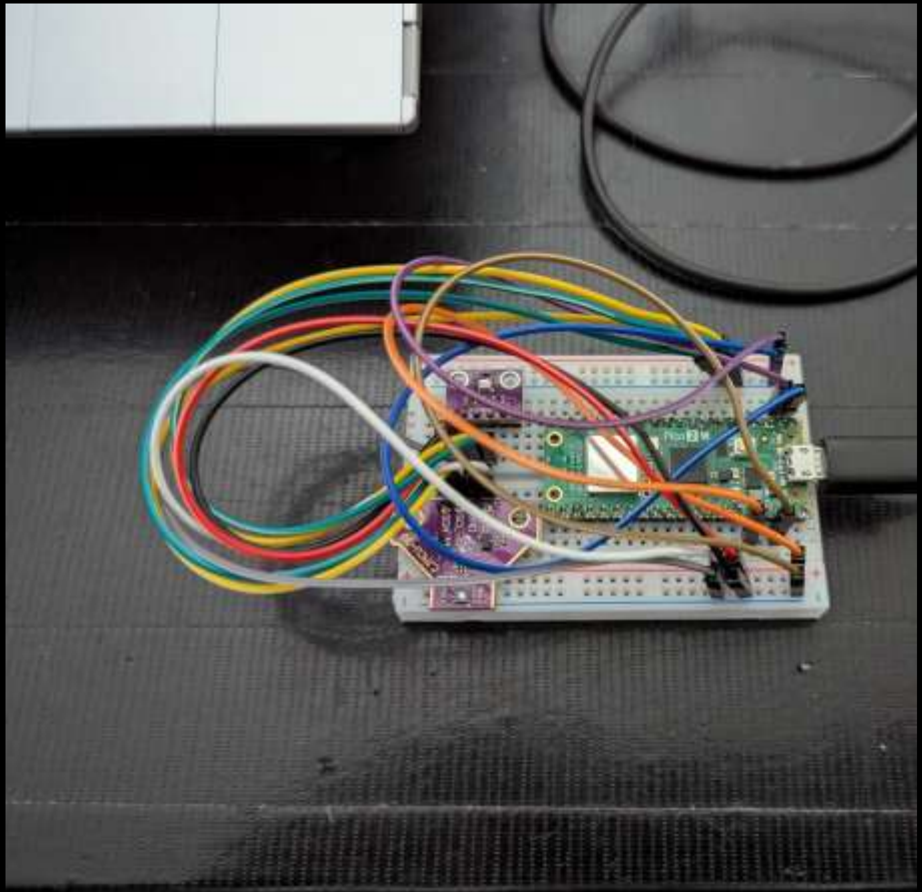
Open-source design: anyone can build and improve upon it

The Design

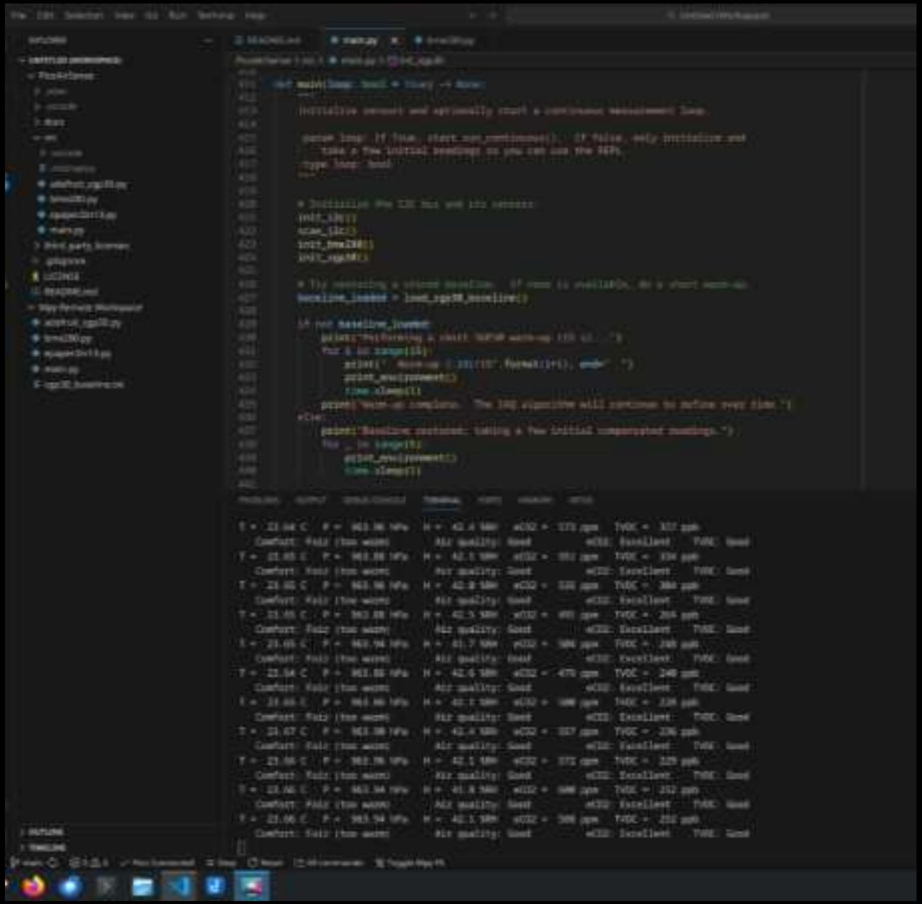
Bringing Hardware and Software together

The Hardware

- Raspberry Pi Pico 2 W: the low-cost ‘brains’ of the system
- BME280 environmental sensor module: temperature, humidity and air pressure
- SGP30 air quality sensor module: eCO2 and TVOC
- Breadboard and cables



The Software



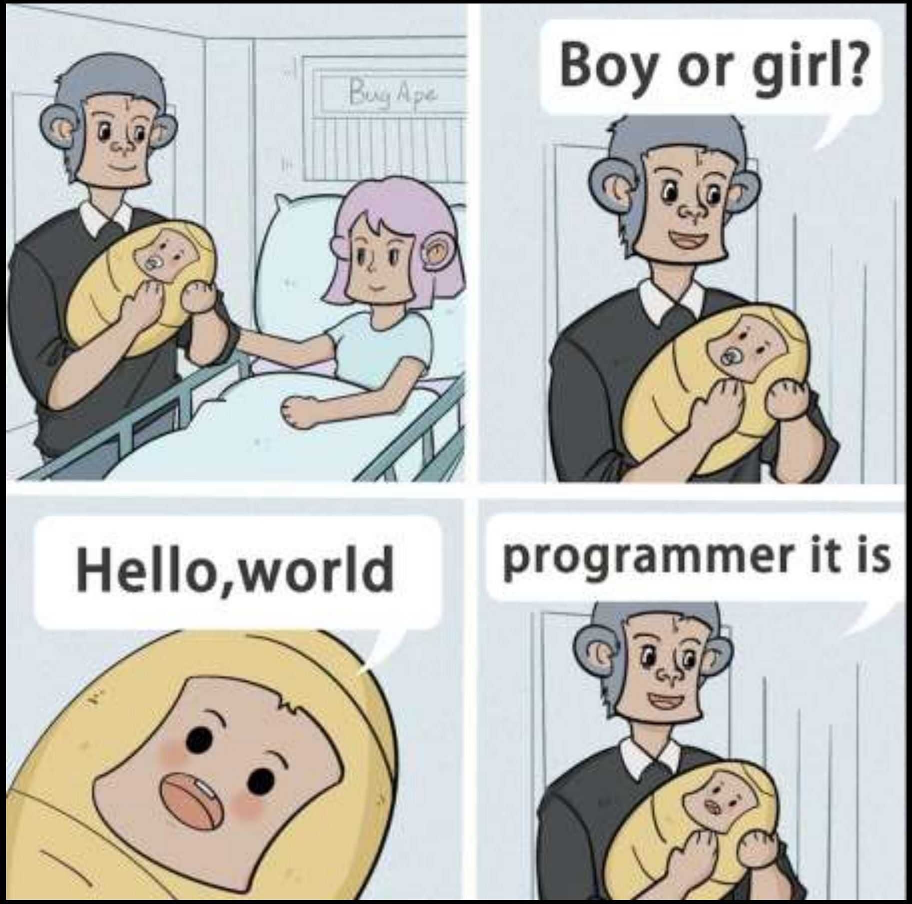
- MicroPython for the Raspberry Pi Pico 2 W
- VSCode with the official Raspberry Pi Pico extension
  - Both an IDE for coding and an interface to read sensor values.
- Open-source SGP30 library
- Public GitHub Repository
- MIT License

Some Assembly Required

Not *that* Assembly. MicroPython. Kinda like Python, but more micro.

What Needed Coding

- No open-source MicroPython BME280 library
  - No problem! Sounds like a fantastic learning opportunity.
  - Official BME280 documentation is robust.
- Main application
  - Initialize the devices
  - Pull sensor data
  - Display sensor data
  - Don’t catch fire



The Results

Wait, how many lines of code did I just write?

The BME280 Library

- This ended up being far easier than expected, but still so very difficult too.
- SO. MUCH. RESEARCH.
  - Registers for days. I dream in hexadecimal now. 🐏
  - ChatGPT is my new best friend. Maybe my only real friend… ❤️
  - Did you know that bitwise operators can be useful? I do now.
  - Can you convert decimal to hexadecimal to binary, do bitwise operations on the values, then convert the new value back, all in your head? I can't.
  - Some crazy person had to sit down and create a formula to compensate raw sensor values into proper values based on calibration values, all so that someone with far less aptitude could make use of those formulas later.
- The code got written. Nobody wants to see it. They just want it to work. 🛠️
  - It works. That's all that matters. 🥳

The main() application

- This should be simple, right? Wrong! ❌
  - Nearly 500 lines of code. Almost as long as the BME280 library!
- Initializes the I2C bus and both sensors.
- The SGP30 has a baseline for long-term environmental calibration.
  - This means saving the new baseline every so often and loading it on power-up.
- Numbers are hard. Humans prefer being told how they should feel, so I do that for them. 📰
- The sensors need to be told to measure their sensors. Every time. All the time.
- The main entry point does all the above, so you don't have to. 👍

Did it work?

Unfortunately, yes, yes it did work. And quite well, too.

```
T =  24.01 C   P =  974.02 hPa   H =  40.1 %RH   eCO2 =  707 ppm   TVOC =  195 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Good         TVOC: Good
T =  24.02 C   P =  973.99 hPa   H =  40.4 %RH   eCO2 =  794 ppm   TVOC =  222 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Good         TVOC: Good
T =  24.01 C   P =  974.05 hPa   H =  40.0 %RH   eCO2 =  744 ppm   TVOC =  218 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Good         TVOC: Good
T =  24.01 C   P =  974.05 hPa   H =  39.9 %RH   eCO2 =  654 ppm   TVOC =  221 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Excellent    TVOC: Good
T =  24.02 C   P =  974.02 hPa   H =  40.2 %RH   eCO2 =  650 ppm   TVOC =  234 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Excellent    TVOC: Good
T =  24.02 C   P =  973.99 hPa   H =  40.5 %RH   eCO2 =  609 ppm   TVOC =  219 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Excellent    TVOC: Good
T =  24.02 C   P =  974.07 hPa   H =  39.2 %RH   eCO2 =  524 ppm   TVOC =  224 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Excellent    TVOC: Good
T =  24.02 C   P =  974.05 hPa   H =  40.2 %RH   eCO2 =  455 ppm   TVOC =  205 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Excellent    TVOC: Good
T =  24.01 C   P =  974.02 hPa   H =  40.0 %RH   eCO2 =  447 ppm   TVOC =  209 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Excellent    TVOC: Good
T =  24.01 C   P =  974.02 hPa   H =  39.8 %RH   eCO2 =  410 ppm   TVOC =  189 ppb
   Comfort: Fair (too warm)      Air quality: Good          eCO2: Excellent    TVOC: Good
```

Maybe it works a little too well…

- I created a machine that makes numbers go up any time I'm close to it.
- Yes, I shower regularly. The numbers go up *especially* after I shower.
- Washed my hands recently? Numbers go up the moment I reach for it.
- Cat sitting on it? Numbers go *way* up.
- At least it lacks the ability to destroy us all. For now.

Future Potential

Full open-source design

The Wish List

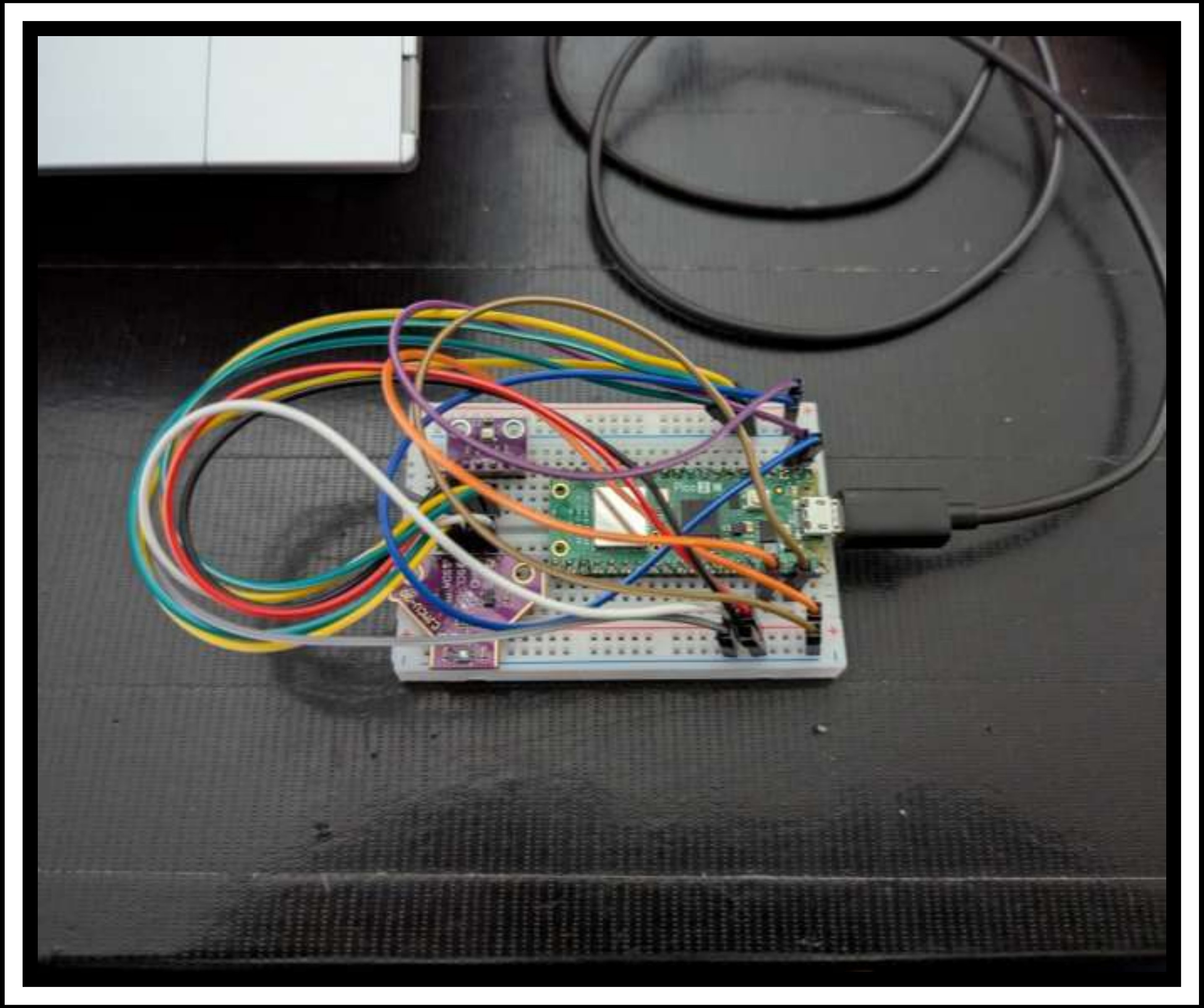
- Attached e-ink display
  - I originally planned to implement this feature, but the display's ribbon cable was ripped. DOA. RIP.
- Exposing data over Wi-Fi
  - The Pico has built-in Wi-Fi, so it's entirely possible to use that to start collecting data on an external service.
  - This could enable historical data collection, monitoring, and potentially even alert notifications.
- Open-source 3D-printed enclosure
  - Currently all the cables and circuit boards are exposed.
  - This could provide not only protection, but a professional appearance too.

Conclusion

Just because a machine can smell doesn’t mean it wants to.

Final Product

- Indoor air quality monitor
- Fully open-source design
- Uses affordable, off-the-shelf parts
- Full-featured BME280 MicroPython library written from scratch
- Complete main application that displays sensor data with metrics
- Success!



Thank you!

- Adrien Abbey

https://github.com/adrienabbey/PicoAirSense/