Personal Air Quality Monitor

This project monitors the temperature, humidity, and air quality of 1.0 μm and 2.5 μm particulates every 5 minutes, placing the results in a log file for later inspection. A Pimoroni scrollphathd displays the current 2.5 μm data.

With wildfires cropping up all over the world due to climate change, a personal air quality monitor is very useful to know whether you should refrain from exercising or wear a mask just to go outside. This sensor won't quite agree with the official ones that are mounted 5 meters above the ground, but the numbers aren't that far off and numbers over 100 are pretty unhealthy.

Parts list

- Pi Zero W or Pi Zero WH
- >=8 GB μSD card
- Suitable board or plastic piece to mount the Pi
- DHT22 Temperature Sensor (Adafruit product 385)
- PMSA003I air quality monitor (Adafruit product 4632)
- Pimoroni scrollhat HD (Adafruit product 3473)
- Stacking header (Adafruit 1979)
- GPIO header for the ZeroW
- a 2x6 pin header to mount the sensors on (cut down a standard 2x20 header)
- optionally but useful, a I2C Stemma cable (Adafruit product 4209)
- 5V 2A USB power supply

Software Setup

Download and install the latest version of RaspiOS Lite on your micro SD card using any of the excellent instructions in the Raspberry Pi Forum. Initialize Raspian with raspi-config to the proper international options (time zone, keyboard, wireless country code, etc.), *change the default password*, set the host name in advanced options to something like Air_monitor. Make sure that you enable SSH in the Interface section of raspi-config, and enable I2C, then reboot.

At this point you can ssh into the pi from another computer or work with a local keyboard and monitor. Do the usual update and upgrade to the operating system:

```
$ sudo apt-get update && sudo apt-get upgrade –y
```

Next install the pip3 and libgpiod applications:

```
$ sudo apt-get install python3-pip python3-libgpiod
```

Now load the needed AdaFruit libraries for Python access.

```
$ sudo pip3 install adafruit-circuitpython-pm25 python3-scrollphathd
```

Next load wiringPi to access the DHT22

```
$ sudo apt-get install wiringpi
```

The Programs

Download the air_monitor.py, dht22.c, and logupdate.sh files from github: https://github.com/wpballa/air-monitor. The python code uses Adafruit's Circuit Python codes to get the air quality numbers. Unfortunately, the current Circuit Python DHT22 control code is unreliable, so a direct c version must be used. The python and shell programs are designed to be executed by crontab, so make things executable:

```
$ chmod +x air_monitor.py
$ chmod +x logupdate.sh
```

Now compile the dht22 code (the first comment line has this if you want to cut and paste).

```
$ gcc -Wall dht22.c -o dht22 -L/usr/lib/ -l wiringPi
```

We will run the codes every 5 minutes with cron, so type:

```
$crontab -e
```

and add the following lines at the end of the file

```
*/5 * * * * /home/pi/air_quality.py
1-56/5 * * * * /home/pi/dht22
0 0 * * * /home/pi/logupdate.sh
```

The last line takes the air_quality.log file and renames it with the date it was generated and starts a fresh log file at midnight. Save with control-O and exit with control-X. This crontab file will execute air_quality.py every 5 minutes of every day and hour, writing the information to the air_quality.log file. The dht22 file runs one minute later and appends information to the file. The log file is structured to be opened with LibreOffice Calc as a comma-separated data file (CSV), where you can plot it at your leisure. To split the columns of particulate data for plotting, add empty columns after the two particulate columns, then select each column and use the Data menu Text to Columns feature in LibreOffice Calc. You can also ssh into the monitor and view the latest data with

```
$ tail -n 24 air_quality.log
```

which will show the last 24 entries covering 2 hours.

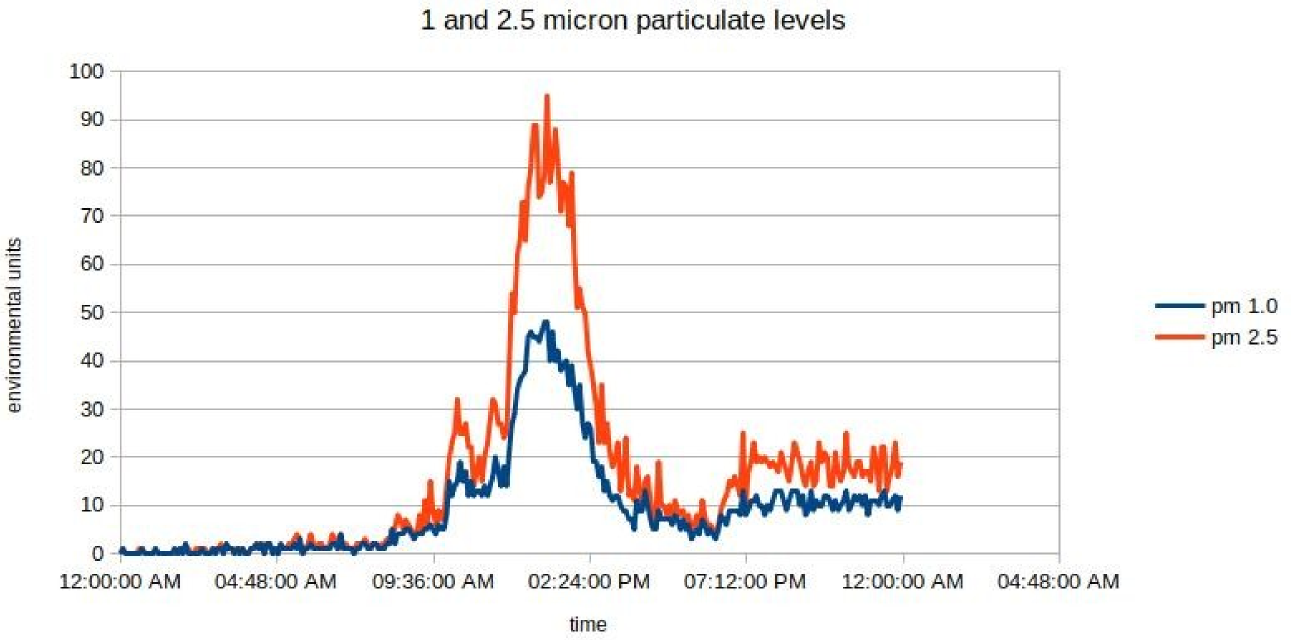

Figure 1: A typical day with smoke rolling through the valley.

Hardware Setup

Solder a GPIO header into the 40-pin holes on the ZeroW. Next solder a stacking header onto the scrollphathd so that the pins extend on the display side. For flexibility, I recommend using a 2x6 header to connect all the sensors, and then plug this onto the scrollphathd once it is mounted on the ZeroW.

The DHT22 temperature sensor has 4 pins, but only 3 of them are active. With the openings on the sensor facing you, pin 3 (numbered from left) has no utility and you can cut it off or bend it out of the way. Next, we will connect to physical pins 1, 7 and 9 of the GPIO so bend pin 1, 2 and 4 of the DHT22 to align with these pins. Next solder the 10k resistor between pins 1 and 7 of the 2x6 header. Then solder the DHT22 to the three active header pins.

Next you need to connect the PMSA003I to the header. There is a header provided with the PMSA but you can wire directly using the holes in the circuit board or use the optional cable that plugs into the PMSA003I and attach the ends to the Raspberry Pi. The power connections are to the +5V (physical pin 2) and Ground (physical pin 6) To connect the I2S use wire to attach the SCA to physical pin 3 and SCL to physical pin 5. It gets a bit crowded so take your time soldering.

Connect the 2x6 header to the ZeroW, and power up the Raspberry Pi, which will also power up the PMSA300I and see that both power lights are on. Then you can test things either directly or via SSH by entering:

```
$./air_quality.py
```

and see if you get both 1.0 μm and 2.5 μm particulate information printed out. Similarly,

```
$./dht22
```

should show the temperature and humidity levels.

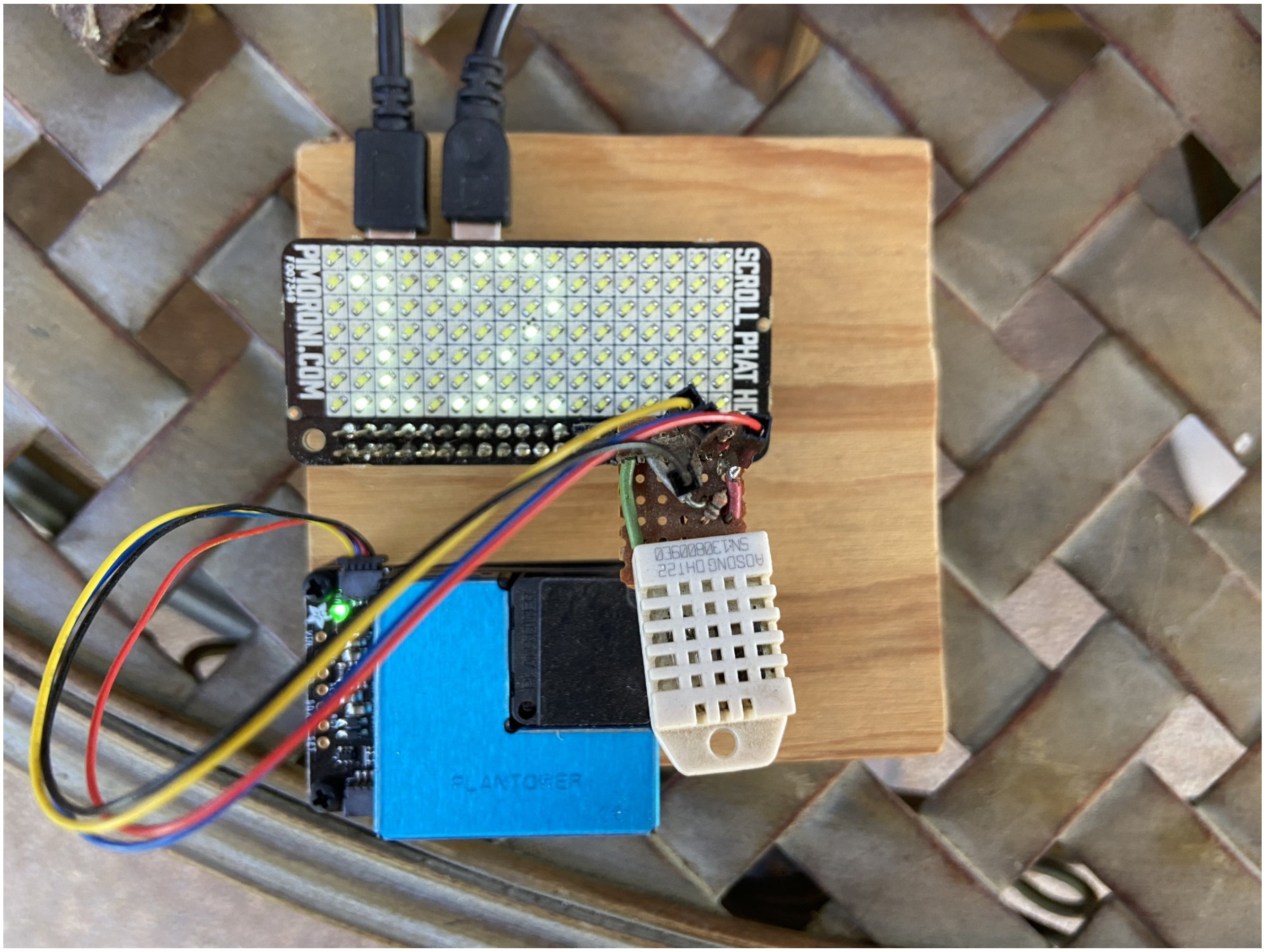

Figure 2: The air monitor is the blue box on the bottom, with the ZeroW hidden by the ScrollphatHD. Note the sensor connections to the header for the GPIO. The DHT sensor connects to physical pins 1, 7, and 9 while the PMS300I connects to physical pins 3, 5, 2, and 6.