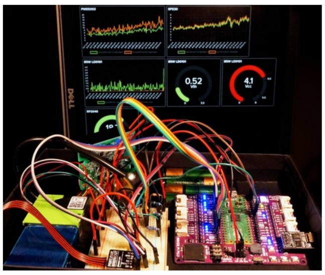
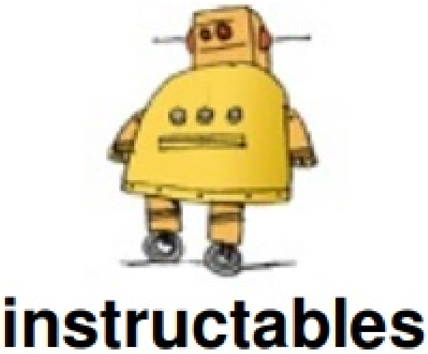
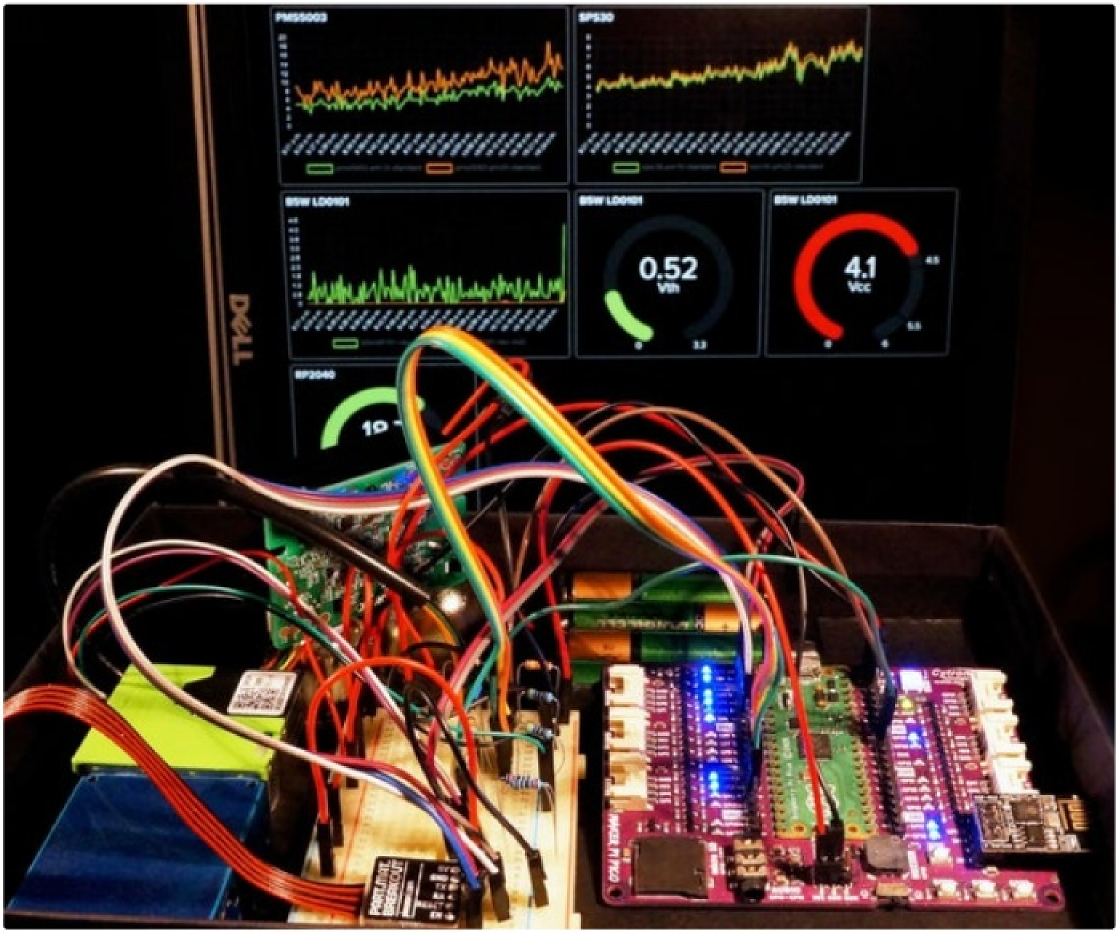


instructables ESP-01S Publishing Particulate Matter Sensor User Guide

Home » instructables » instructables ESP-01S Publishing Particulate Matter Sensor User Guide



ESP-01S Publishing Particulate Matter Sensor
User Guide



Contents

ESP-01S Publishing Particulate Matter Sensor

Publishing Particulate Matter Sensor Data to Adafruit IO With Maker Pi Pico and ESP-01S
by kevinjwalters

This article shows how to publish data from three low-cost particulate matter sensors to the Adafruit IO IoT service using the Cytron Maker Pi Pico running a CircuitPython program transmitting the sensors' outputs over Wi-Fi with an ESP-01S module running AT firmware.

The WHO identifies PM2.5 particulate matter as one of the greatest environmental risks to health with 99% of the world's population living in places where the WHO air quality guidelines levels were not met in 2019. It estimates 4.2 million premature deaths were caused by this in 2016.

The three particulate matter sensors shown in this article are:

- the Plantower PMS5003 using a serial connection;
- the Sensirion SPS30 using i2c;
- the Omron B5W LD0101 with pulse outputs.

These optical sensors are similar to the ones found in one type of domestic smoke alarm but they differ in their attempt to count particles of different sizes rather than just alarm at a threshold concentration.

The red laser-based PMS5003 is a commonly used hobbyist sensor and can be found in the PurpleAir PA-II air quality sensor. The SPS30 is a more recent sensor using the same principle and can be found in the Clarity Node-S air qualitysensor. The infrared LED-based B5W LD0101 sensor has a more primitive interface but is useful for its ability to detect particles larger than 2.5 microns – the other two sensors cannot reliably measure these.

Adafruit IO offers a free tier with a limited number of feeds and dashboards – these are sufficient for this project. The free tier data is retained for 30 days but the data can easily be downloaded.

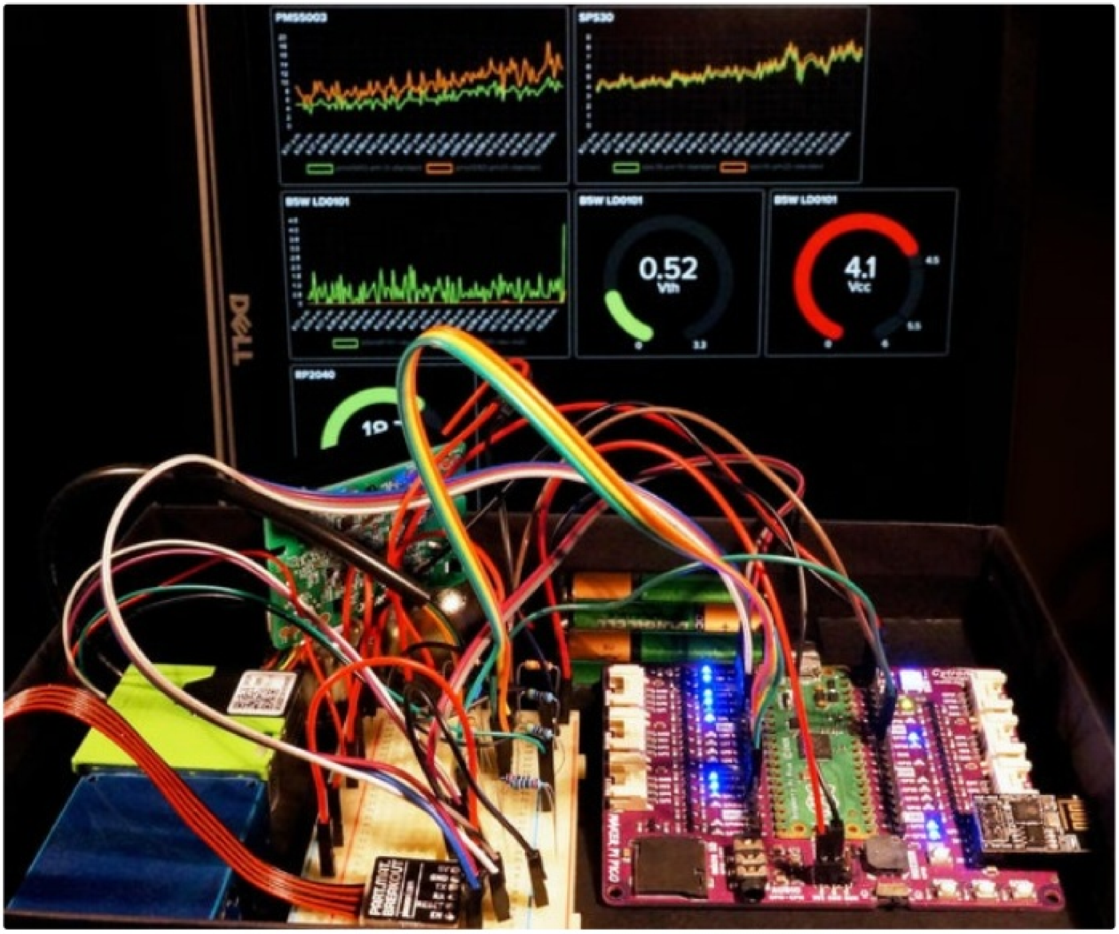
The Maker Pi Pico board in this article is a sample Cytron kindly sent to me to evaluate. The only difference to the production version is the addition of passive components to debounce the three buttons.

The ESP-01S module is likely to need an AT firmware upgrade. This is a relatively complex, fiddly process and may be time-consuming. Cytron sell the module with the appropriate AT firmware on it.

The Omron B5W LD0101 sensor is unfortunately being discontinued by the manufacturer with last orders in March 2022.

Supplies:

- Cytron Maker Pi Pico – Digi-key | PiHut
- ESP-01S – Cytron's board comes with appropriate ATfirmware.
- ESP-01 USB adapter/programmer with reset button – Cytron.
- Breadboard.
- Female to male jumper wires, maybe 20cm (8in) minimum length.
- Plantower PMS5003 with cable and breadboard adapter – Adafruit
- or Plantower PMS5003 + Pimoroni breadboard adapter – Pimoroni + Pimoroni
- Sensirion SPS30 – Digi-key
    - Sparkfun SPS30 JST-ZHR cable to 5 male pins – Digi-key
    - 2x 2.2k resistors.
- Omron B5W LD0101 – Mouser
    - Omron cable described as a harness (2JCIE-HARNESS-05) – Mouser
    - 5 pin male header (for adapting cable to breadboard).
    - solder – crocodile (alligator) clips could work as an alternative to soldering.
    - 2x 4.7k resistors.
    - 3x 10k resistors.
    - 0.1uF capacitor.
    - Battery power for Omron B5W LD0101:
        - 4AA battery holder for rechargeable NiMH batteries (better choice).
        - or 3AA batter holder for alkaline batteries.
- A USB power pack may be useful if you want to run outside away from a USB power source.



Step 1: USB Programmer for Updating Flash on the ESP-01S

The ESP-01S module is unlikely to come with appropriate AT firmware on it unless it is from Cytron. The easiest way to update it is to use a Windows desktop or laptop with a USB adapter which write-enables the flash and has a reset button.

Unfortunately a very common, no-brand adapter often described as something like an “ESP-01 Programmer Adapter UART” does not have buttons or switches to control these. The video above shows how this can be quickly retrofitted

with some improvised switches made from two male-to-female jumper wires cut in two and soldered onto the pins on the underside of the programmer board. An alternative approach to this using a breadboard can be seen in Hackaday:

ESPHome on ESP-01 Windows Workflow.

https://www.youtube.com/watch?v=wXXXgaePZX8

Step 2: Updating Firmware on ESP-01S Using Windows

A terminal program like PuTTY can be used with the ESP-01 Programmer to check the firmware version. The firmware makes the ESP8266 act a bit like a modem with commands inspired by the Hayes command set. The AT+GMR AT+GMR command shows the firmware version.

```
AT+GMR
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
```

Cytron have a guide describing how to apply the firmware update using the Espressif Flash Download Tool (Windows only) on GitHub: CytronTechnologies/esp-at-binaries. Cytron also provide a copy of the firmware binary, Cytron_ESP- 01S_AT_Firmware_V2.2.0.bin.

After a successful upgrade the new firmware will be reported as version 2.2.0.0

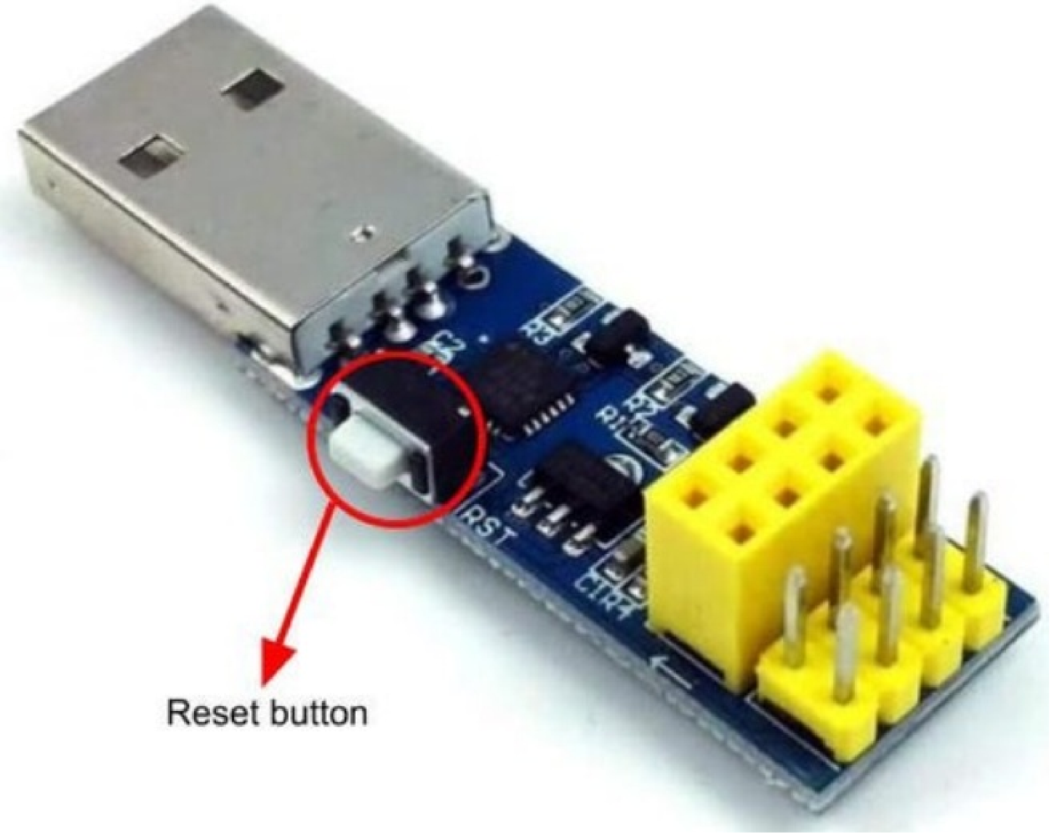
```
AT+GMR
AT version:2.2.0.0(b097cdf – ESP8266 – Jun 17 2021 12:57:45)
SDK version:v3.4-22-g967752e2
```

```
compile time(6800286):Aug 4 2021 17:20:05
Bin version:2.2.0(Cytron_ESP-01S)
```

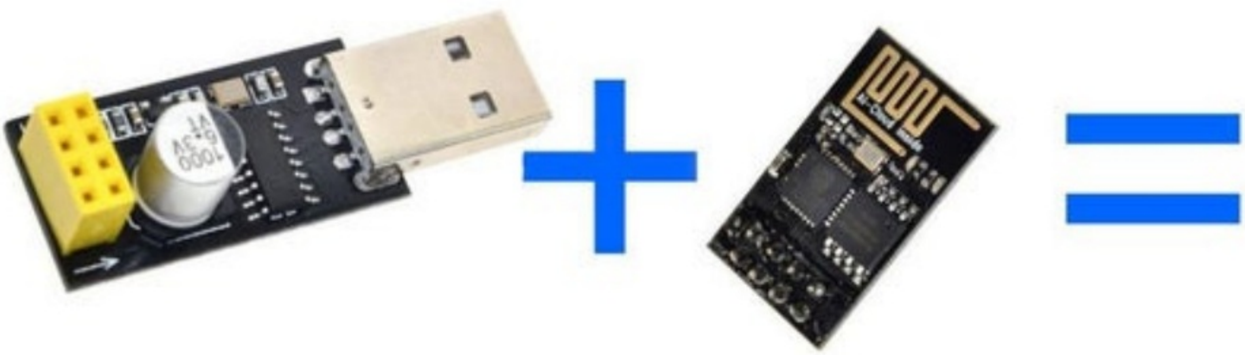
A command line program called esptool is available as an alternative for programming the ESP8266-based ESP-01S and could be used on Linux or macOS.

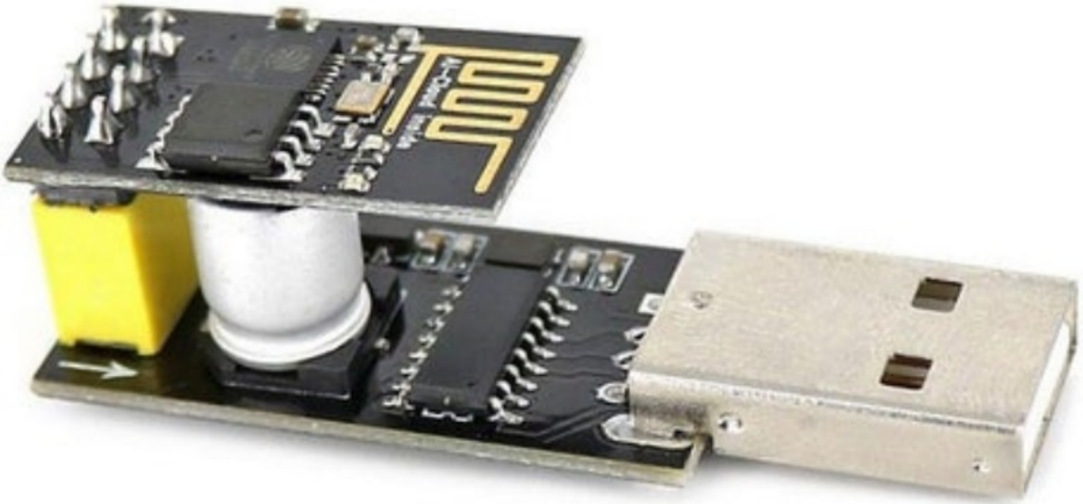
The firmware on the ESP-01S can be tested on the Maker Pi Pico using Cytron's simpletest.py. This sends an ICMP ping to a well-known service on the Internet every 10 seconds and shows the round-trip time (rtt) in milliseconds. This needs a secrets.py file with the Wi-Fi SSID (name) and password – this is described later in this article.

THE GOOD



THE BAD





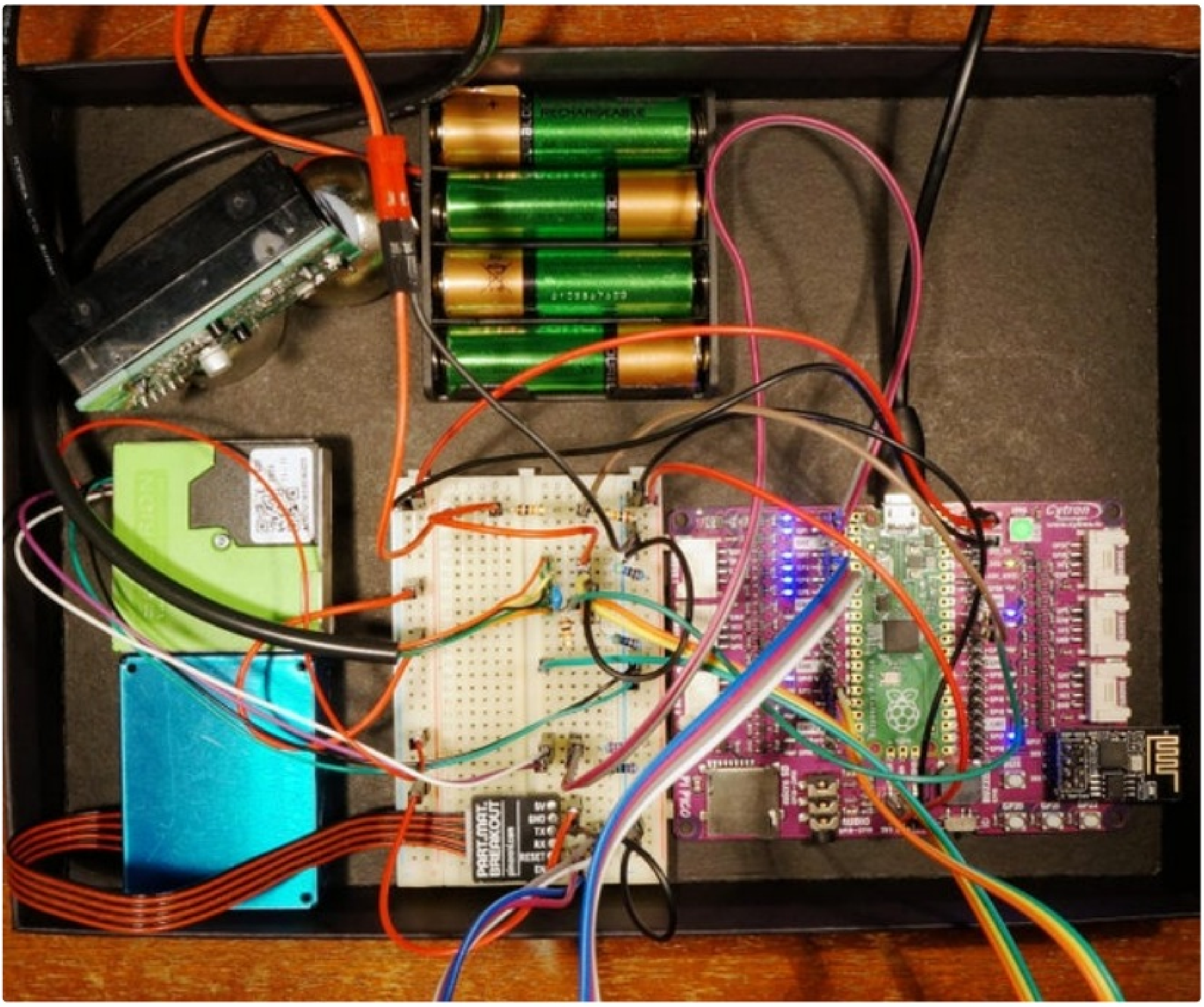
THE UGLY

Step 3: Connecting the Sensors

A half-size breadboard was used to connect the three sensors and to monitor the voltage from the four rechargeable NiMH batteries. A high resolution photo is included of the complete setup above and the next steps describe how each sensor can be connected.
The power rails on the breadboard are powered from the Pi Pico with

- VBUS (5V) and GND to the power rails on left side and
- 3V3 and GND to the right side.

The power rails are marked with a nearby red line for positive rail and blue for negative (or ground) rail. On a full-size (830 hole) breadboard these may have a top set of rails that are not connected to the bottom set of rails.
The batteries are only used to power the Omron B5W LD0101 which needs a steady voltage. The USB power from a computer is often noisy making it unsuitable.



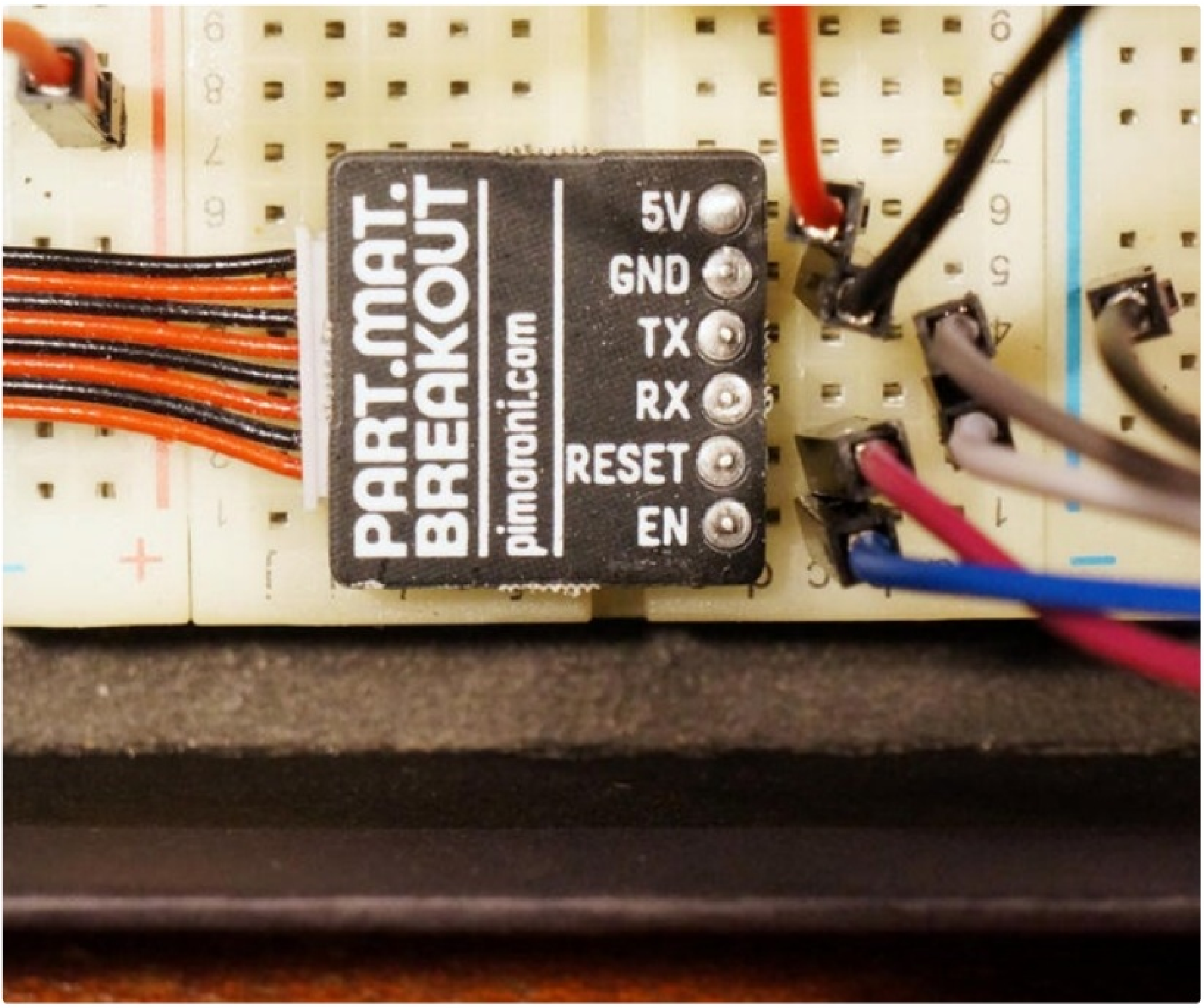
Step 4: Connecting the Plantower PMS5003

The Plantower PMS5003 requires 5V power but its serial “TTL style” interface is 3.3V safe. The connections from the
PMS5003 via breakout board to the Pi Pico are:

- VCC to 5V (red) via row 6 to 5V rail;
- GND to GND (black) via row 5 to GND;
- SET to EN (blue) via row 1 to GP2;

- RX to RX (white) via row 3 to GP5;
- TX to TX (grey) via row 4 to GP4;
- RESET to RESET (purple) via row 2 to GP3;
- NC (not connected);
- NC.

The datasheet includes a warning about the metal case.
Metal shell is connected to the GND so be careful not to let it shorted [sic] with the other parts of circuit except GND.
The component tends to ship with blue plastic fllm on the case to protect the surface from scratches but this should not be relied upon for electrical insulation.



Step 5: Connecting the Sensirion SPS30

The Sensirion SPS30 requires 5V power but its i2c interface is 3.3V safe. The only additional components are two 2.2k resistors to act as pull-ups for the i2c bus. The connections from the SPS30 to the Pi Pico are:

- VDD (red) to 5V5V rail;
- SDA (white) to GP0 (grey) via row 11 with 2.2k resistor to 3.3V rail;
- SCL (purple) to GP1 (purple) via row 10 with 2.2k resistor to 3.3V rail;
- SEL (green) to GND;
- GND (black) to GND.

The connector on the lead may require a firm push to insert it properly into the SPS30.

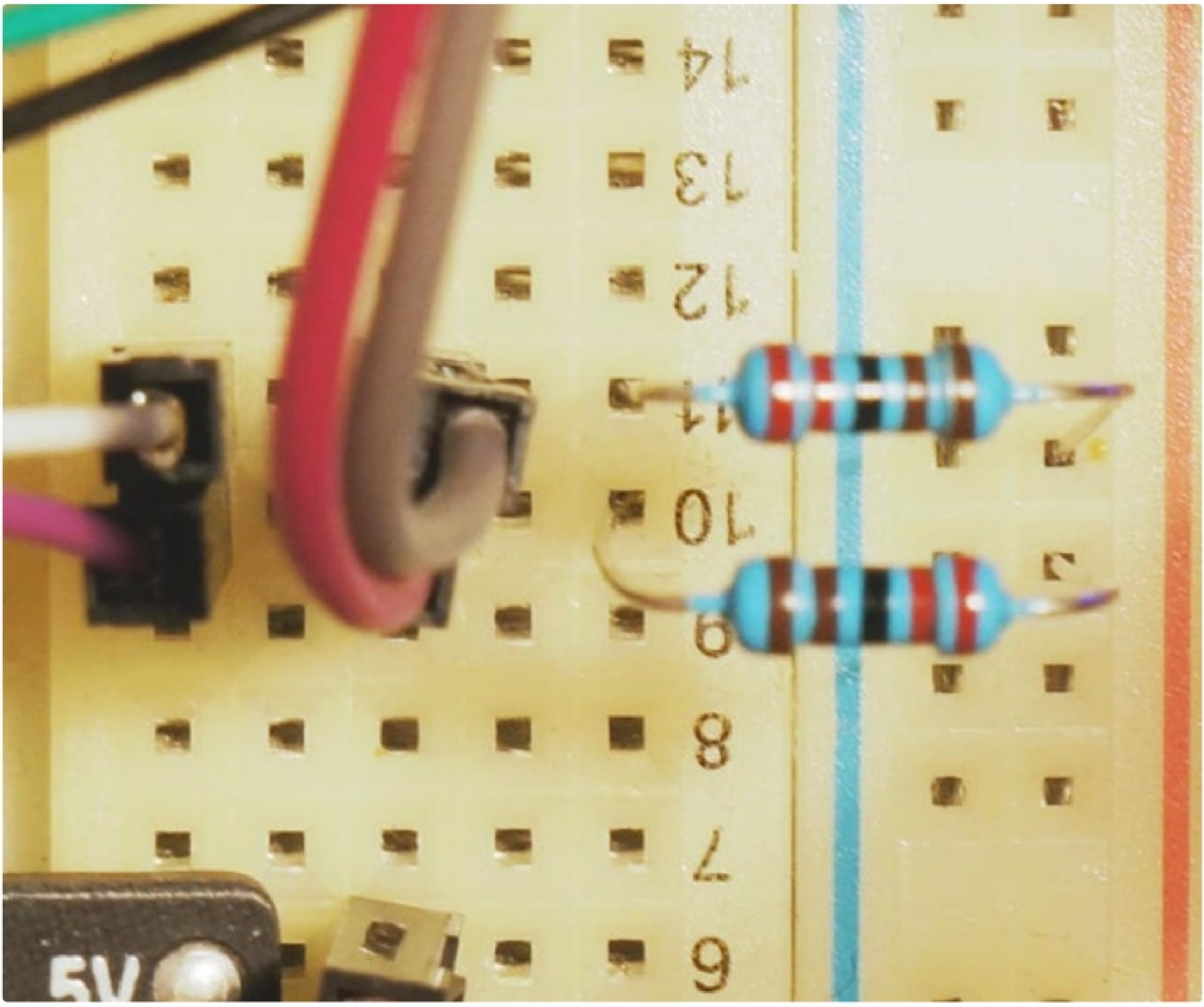
The SPS30 also supports a serial interface which Sensirion recommends in the datasheet.

Some considerations should be made about the use of the I2C interface. I2C was originally designed to connect two chips on a PCB. When the sensor is connected to the main PCB via a cable, particular attention must be paid to electromagnetic interference and crosstalk. Use as short as possible (< 10 cm) and/or well shielded connection cables.

We recommend using the UART interface instead, whenever possible: it is more robust against electromagnetic interference, especially with long connection cables.

There is also a warning about the metal parts of the case.

Note, that there is an internal electrical connection between GND pin (5) and metal shielding. Keep this metal shielding electrically oating in order to avoid any unintended currents through this internal connection. If this is not an option, proper external potential equalization between GND pin and any potential connected to the shielding is mandatory. Any current though the connection between GND and metal shielding may damage the product and poses a safety risk through overheating.



Step 6: Connecting the Omron B5W LD0101

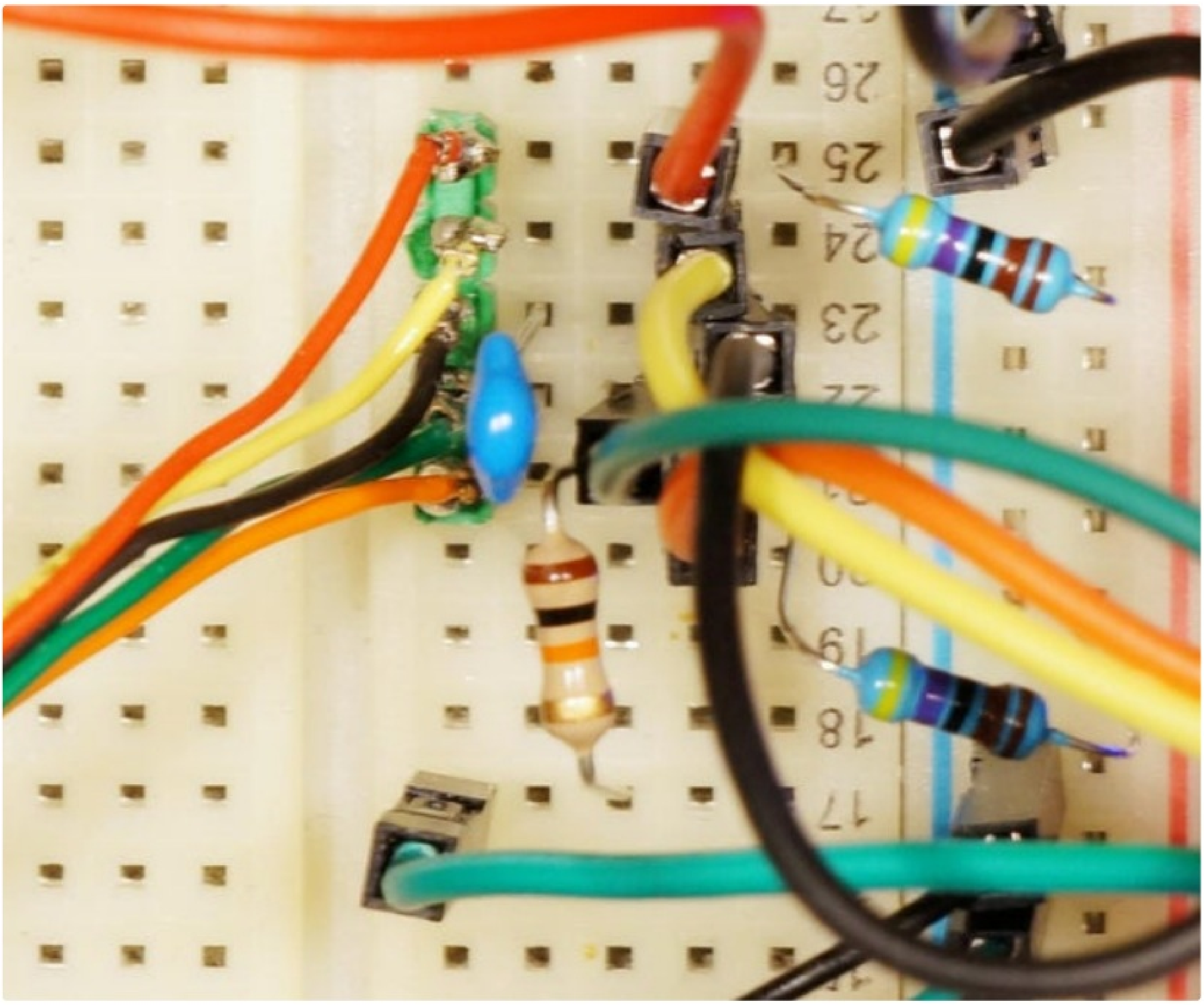
The Omron cable is not intended for use with a breadboard. One quick way to convert it to breaboard use is to cut off the socket, strip the wires and solder them to a five pin length of male header pins. Crocodile (alligator) clips could be used as an alternative approach to avoid soldering.

The Omron B5W LD0101 requires a 5V steady power supply. Its two outputs are also at a 5V level which is incompatible with the Pi Pico’s 3.3V inputs. The presence of resistors on the sensor board make it easy to drop this to a safe value by adding a 4.7k resistor to ground per output. The on-board resistors are documented in the datasheet which make this a reasonable approach.

The connections from the B5W LD0101 to the Pi Pico are:

- Vcc (red) to 5V (red) rail via row 25;
- OUT1 (yellow) to GP10GP10 (yellow) via row 24 with 4.7k resistor to GND;
- GND (black) to GND (black) via row 23;
- Vth (green) to GP26GP26 (green) via row 22 with 0.1uF capacitor to GND;
- OUT2 (orange) to GP11 (orange) via row 21 with 4.7k resistor to GND.

The **GP12** (green) from the Pi Pico connects to row 17 and a 10k resistor connects row 17 to row 22.
The datasheet describes the power supply requirement as:
Minimum 4.5V, typical 5.0V, maximum 5.5V, ripple voltage range 30mV or less is recommended. Ensure there is no noise below 300Hz. Con
rm the allowable ripple voltage value using an actual machine.
Three alkaline or four rechargeable (NiMH) batteries are the easiest way to provide steady, stable voltage of around 5V to the sensor. A USB power pack is likely to be a poor choice because the voltage is typically from a lithium battery using a buck-boost converter which makes it noisy.
The B5W LD0101 uses convection for its air flow and must be placed upright to work correctly. A change of supply voltage is likely to affect the temperature of the heater and the associated air ow. Ambient temperature must also have an effect.

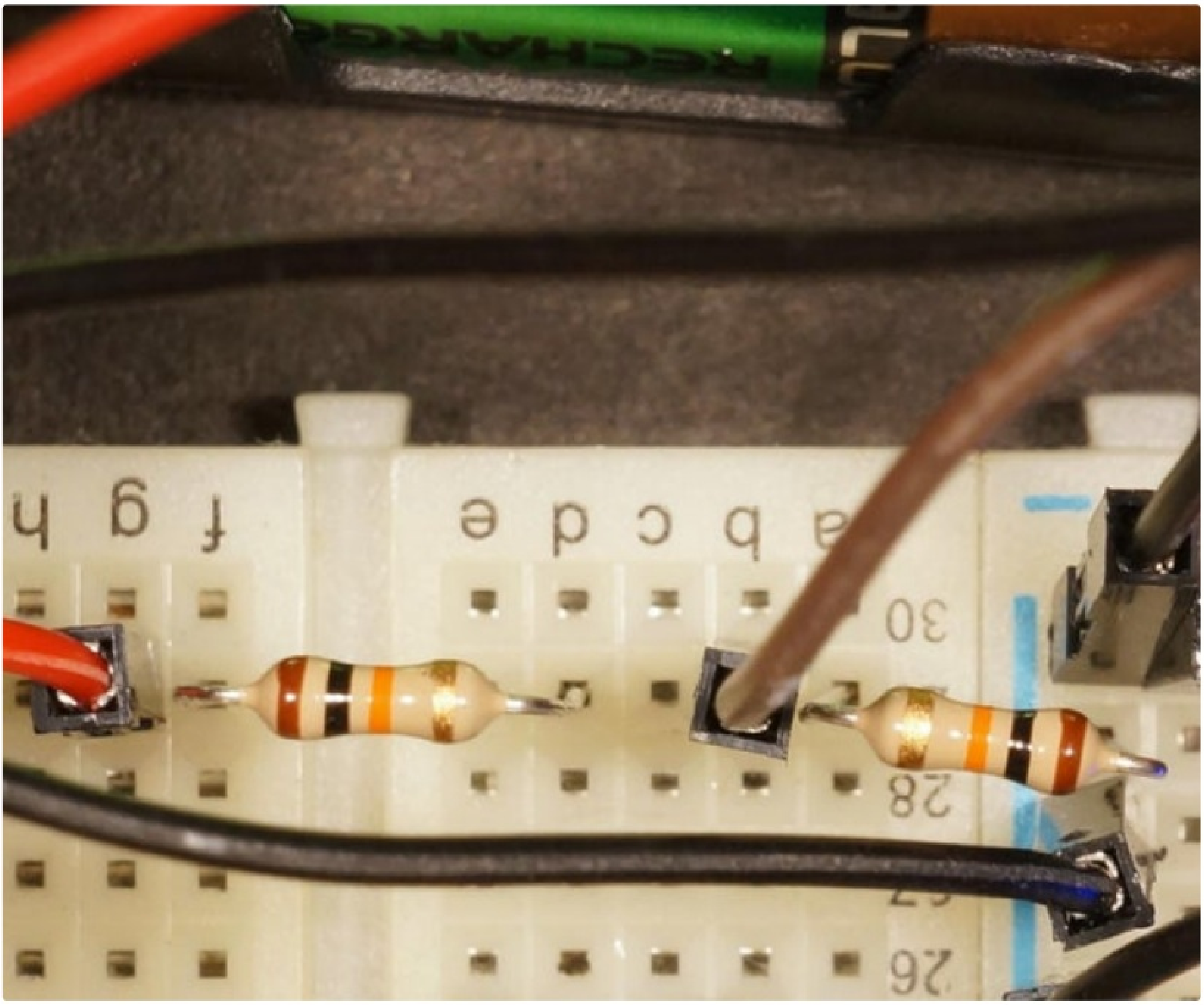


Step 7: Battery Monitoring With Potential Divider

The battery voltage exceeds the 3.3V level of the Pi Pico's RP2040 processor's inputs. A simple potential divider can reduce this voltage to be within that range. This allows the RP2040 to measure the battery level on an analogue capable (GP26 to GP28) input.
A pair of 10k resistors was used above to halve the voltage. It's common to see higher values used like 100k to minimise the wasted current. The connections are:

- B5W LD0101 Vcc (red) jumper wire to row 29 left side;
- 10k resistor on row 29 between left and right side on row 29;
- Brown jumper wire to Pi Pico GP27;
- 10k resistor from right side of row 29 to nearby GND rail.

GP28 on the Maker Pi Pico can be used as an analogue input but since it's also connected to the RGB pixel that may have afltiny effect on the value and may even illuminate or change if the input looks like the WS2812 protocol!



Step 8: Installing CircuitPython and Sensor Data Publishing Program

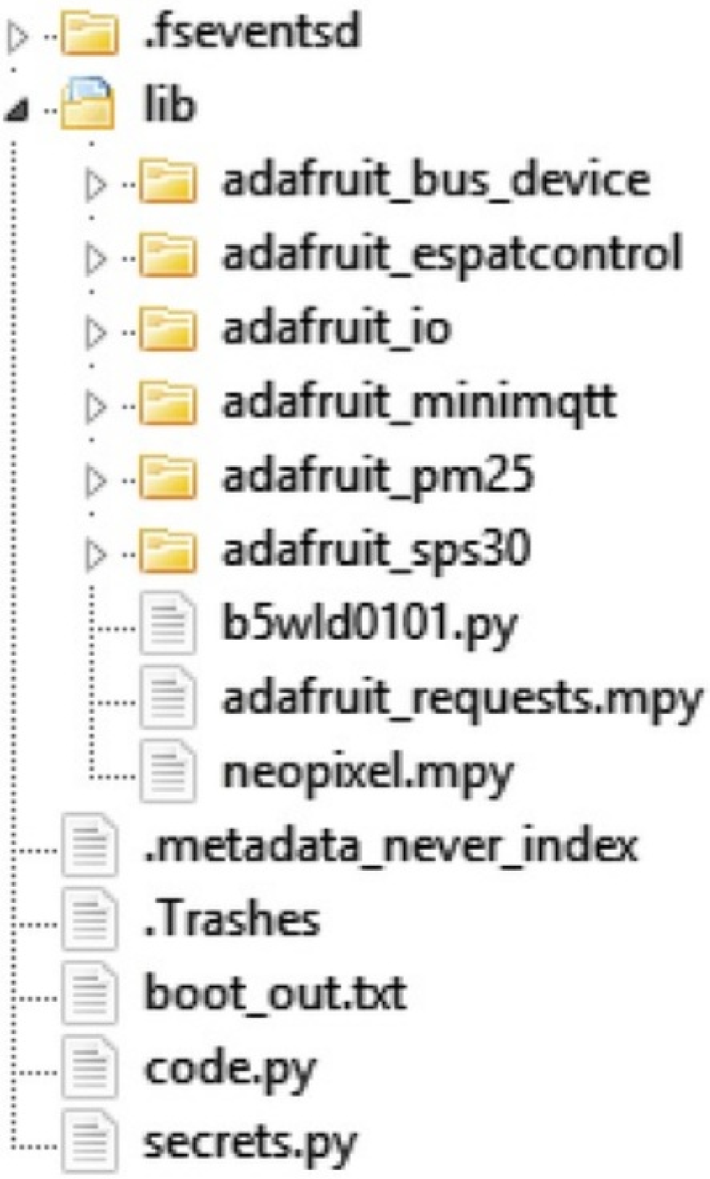
If you are not familiar with CircuitPython then it's worth reading the Welcome to CircuitPython guide first.

1. Install the following seven libraries from the version 7.x bundle from **https://circuitpython.org/libraries** into the lib directory on the CIRCUITPY drive:
    1. adafruit_bus_device
    2. adafruit_minimqtt
    3. adafruit_io
    4. adafruit_espatcontrol
    5. adafruit_pm25
    6. adafruit_requests.mpy
    7. neopixel.mpy

2. Download these two extra libraries to the lib directory by clicking Save link as… on the files inside the directory or on the file:
   1. adafruit_sps30 from **https://github.com/kevinjwalters/Adafruit_CircuitPython_SPS30**
   2. b5wld0101.py from **https://github.com/kevinjwalters/CircuitPython_B5WLD0101**
3. Create the secrets.py file (see example below) and fill in the values.
4. Download the program to CIRCUITPY by clicking Save link as… on pmsensors_adafruitio.py
5. Rename or delete any existing code.py file on CIRCUITPY then rename the pmsensors_adafruitio.py to code.py This file is run when the CircuitPython interpreter starts or reloads.

```
# This file is where you keep secret settings, passwords, and tokens!
# If you put them in the code you risk committing that info or sharing it
secrets = {
“ssid” : “INSERT-WIFI-NAME-HERE”,
“password” : “INSERT-WIFI-PASSWORD-HERE”,
“aio_username” : “INSERT-ADAFRUIT-IO-USERNAME-HERE”,
“aio_key” : “INSERT-ADAFRUIT-IO-APPLICATION-KEY-HERE”
# http://worldtimeapi.org/timezones
“timezone” : “America/New_York”,
}
```

The versions used for this project were:
CircuitPython 7.0.0
CircuitPython library bundle adafruit-circuitpython-bundle-7.x-mpy-20211029.zip- earlier verisons from September/October must not be used as the adafruit_espatcontrol library was buggy and half works in a confusing manner.



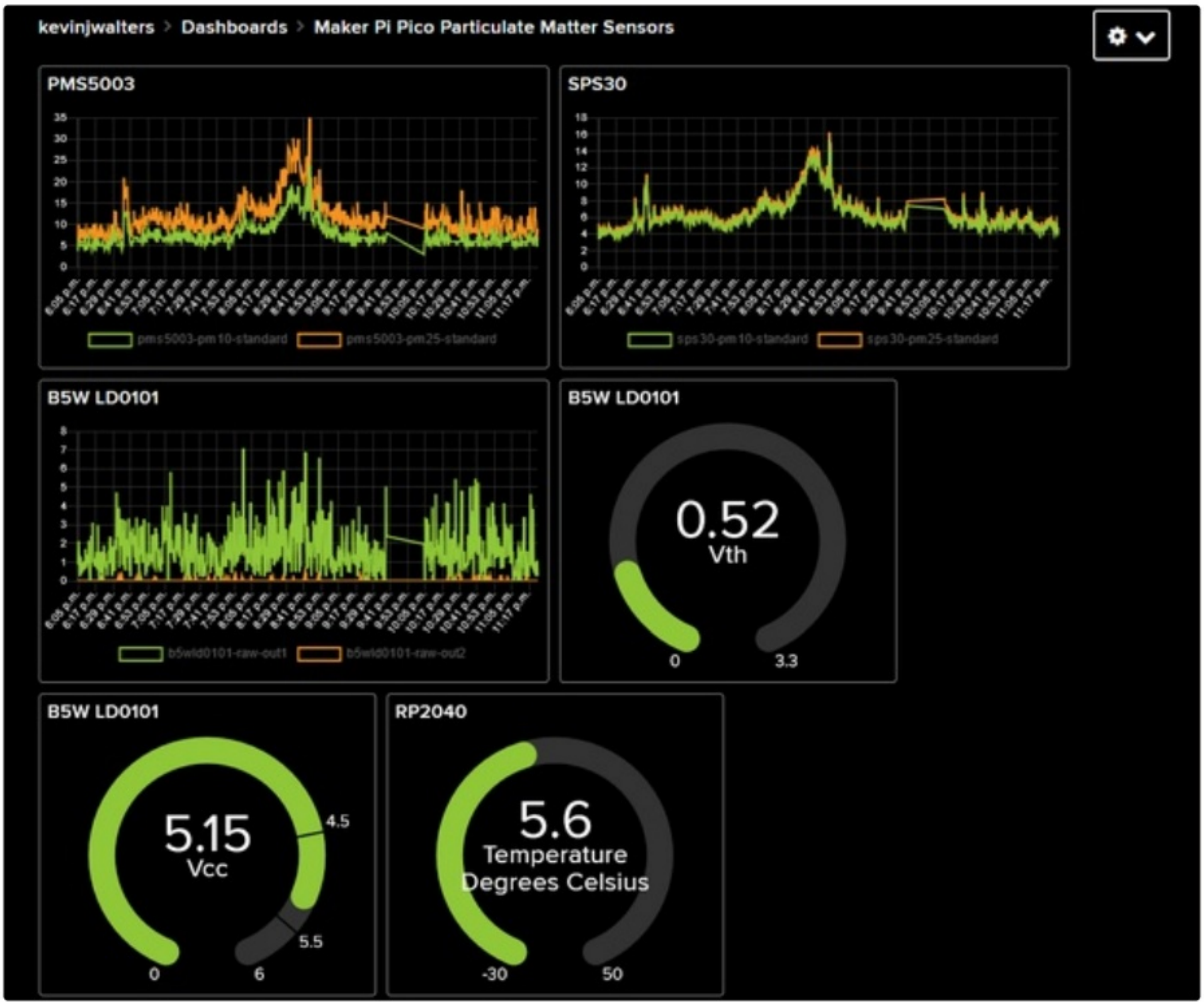
```
.fseventsd
lib
    adafruit_bus_device
    adafruit_espatcontrol
    adafruit_io
    adafruit_minimqtt
    adafruit_pm25
    adafruit_sps30
    b5wld0101.py
    adafruit_requests.mpy
    neopixel.mpy
.metadata_never_index
.Trashes
boot_out.txt
code.py
secrets.py
```

Step 9: Adafruit IO Setup

Adafruit have many guides on their Adafruit IO service, the most relevant ones are:
Welcome to Adafruit IO
Adafruit IO Basics: Feeds
Adafruit IO Basics: Dashboards
Once you are familiar with feeds and dashboards, follow these steps.

1. Create an Adafruit account if you do not already have one.
2. Make a new group called mpp-pm under Feeds
3. Make nine feeds in this new group by clicking on + New Feed button, the names are:
    1. b5wld0101-raw-out1
    2. b5wld0101-raw-out2
    3. b5wld0101-vcc
    4. b5wld0101-vth
    5. cpu-temperature
    6. pms5003-pm10-standard
    7. pms5003-pm25-standard

    8. sps30-pm10-standard
    9. sps30-pm25-standard
4. Make a dashboard for these values, suggested blocks are:
    1. Three Line Chart blocks, one for each sensor with two lines per chart.
    2. Three Gauge blocks for the two voltages and temperature.



Step 10: Verifying the Data Publishing

The Monitor page under Pro file is useful to verify data is arriving in real-time by looking at the Live Data file section. The program turns the RGB pixel blue for 2-3 seconds when it sends the data to Adafruit IO and then returns to green.
The temperature from the RP2040 appears to vary widely between different CPUs and is unlikely to match ambient temperature.
If this does not work then here are a few things to check.

- If the RGB pixel stays for or if data is not received by Adafruit IO then check the USB serial console for output/errors. The numerical output for Mu on the serial console will show if the sensors are working with new lines being printed every 2-3 seconds – see below for example output.
- The Live Errors section on the Monitor page it worth checking if data is being sent but not showing up.
- The debug variable in the program can be set from 0 to 5 to control the volume of debugging information. Higher levels disable the tuple printing for Mu.
- The simpletest.py program is a useful way to prove the Wi-Fi connection is made and connectivity to the Internet works for ICMP traffc.

- Ensure you are using a recent version of the adafruit_espatcontrol library.
- The Maker Pi Pico’s blue LEDs on each GPIO are very useful for getting an instant visual overview of the GPIO state. All of the connected GPIO will be on with the exception of:
  - GP26 will be off because the smoothed voltage (around 500mV) is too low;
  - GP12 will be dim because it’s a ~ 15% duty cycle PWM signal;
  - GP5 will be on but will flicker as data is sent from the PMS5003;
  - GP10 will be offbut will flicker as small particles are detected by the B5W LD0101;
  - GP11 will be off but will flicker very occasionally unless you are in an exceptionally smoky place.

The output intended for the plotter in Mu will look something like this in a room:

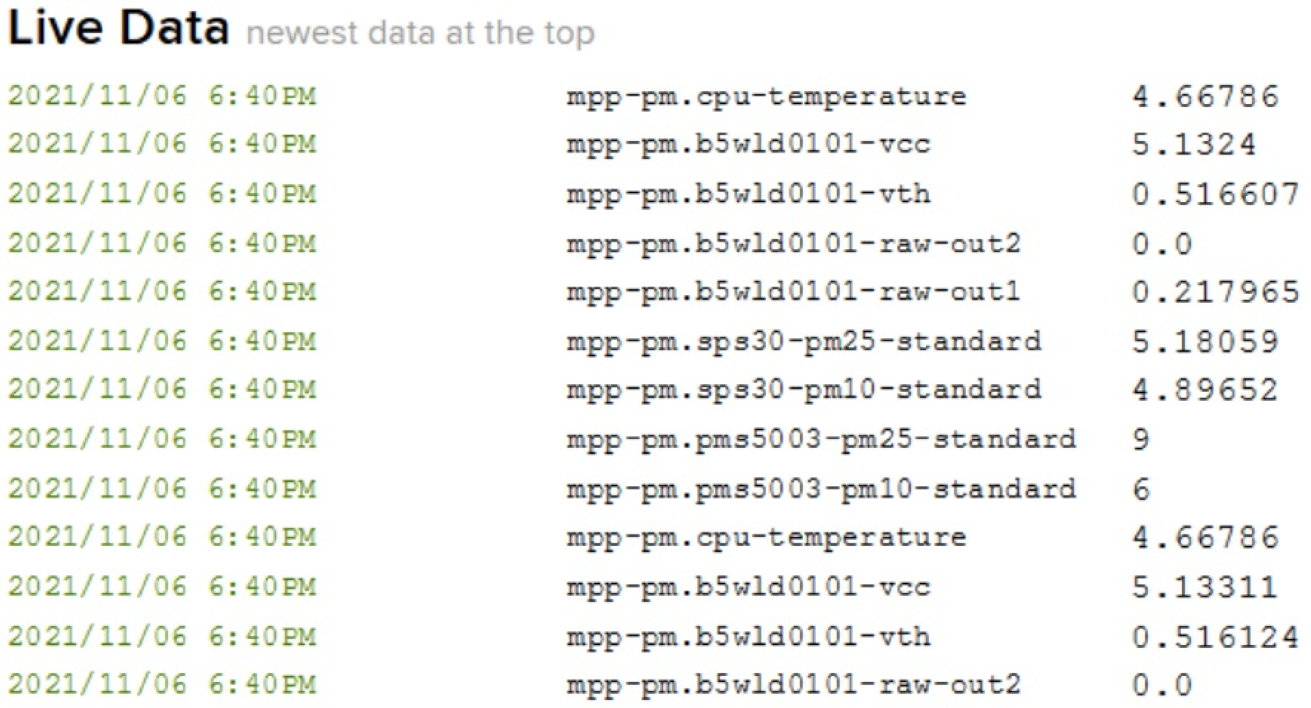
```
(5,8,4.59262,4.87098,3.85349,0.0)
(6,8,4.94409,5.24264,1.86861,0.0)
(6,9,5.1649,5.47553,1.74829,0.0)
(5,9,5.26246,5.57675,3.05601,0.0)
(6,9,5.29442,5.60881,0.940312,0.0)
(6,11,5.37061,5.68804,1.0508,0.0)
```

Or a room with cleaner air:

```
(0,1,1.00923,1.06722,0.0,0.0)
(1,2,0.968609,1.02427,0.726928,0.0)
(1,2,0.965873,1.02137,1.17203,0.0)
(0,1,0.943569,0.997789,1.47817,0.0)
(0,1,0.929474,0.982884,0.0,0.0)
(0,1,0.939308,0.993282,0.0,0.0)
```

The six values per line in order are:

1. PMS5003 PM1.0 and PM2.5 (integer values);
2. SPS30 PM1.0 and PM2.5;
3. B5W LD0101 raw OUT1 and OUT2 counts.



Live Data newest data at the top

2021/11/06 6:40PM	mpp-pm.cpu-temperature	4.66786
2021/11/06 6:40PM	mpp-pm.b5wld0101-vcc	5.1324
2021/11/06 6:40PM	mpp-pm.b5wld0101-vth	0.516607
2021/11/06 6:40PM	mpp-pm.b5wld0101-raw-out2	0.0
2021/11/06 6:40PM	mpp-pm.b5wld0101-raw-out1	0.217965
2021/11/06 6:40PM	mpp-pm.sps30-pm25-standard	5.18059
2021/11/06 6:40PM	mpp-pm.sps30-pm10-standard	4.89652
2021/11/06 6:40PM	mpp-pm.pms5003-pm25-standard	9
2021/11/06 6:40PM	mpp-pm.pms5003-pm10-standard	6
2021/11/06 6:40PM	mpp-pm.cpu-temperature	4.66786
2021/11/06 6:40PM	mpp-pm.b5wld0101-vcc	5.13311
2021/11/06 6:40PM	mpp-pm.b5wld0101-vth	0.516124
2021/11/06 6:40PM	mpp-pm.b5wld0101-raw-out2	0.0

Step 11: Testing the Sensors Inside With Mu and Adafruit IO

The video above shows the sensors reacting to a match being struck to light the incense stick. The PM2.5 peak values from the PMS5003 and SPS30 are 51 and 21.5605, respectively. The B5W LD0101 has uncovered optics

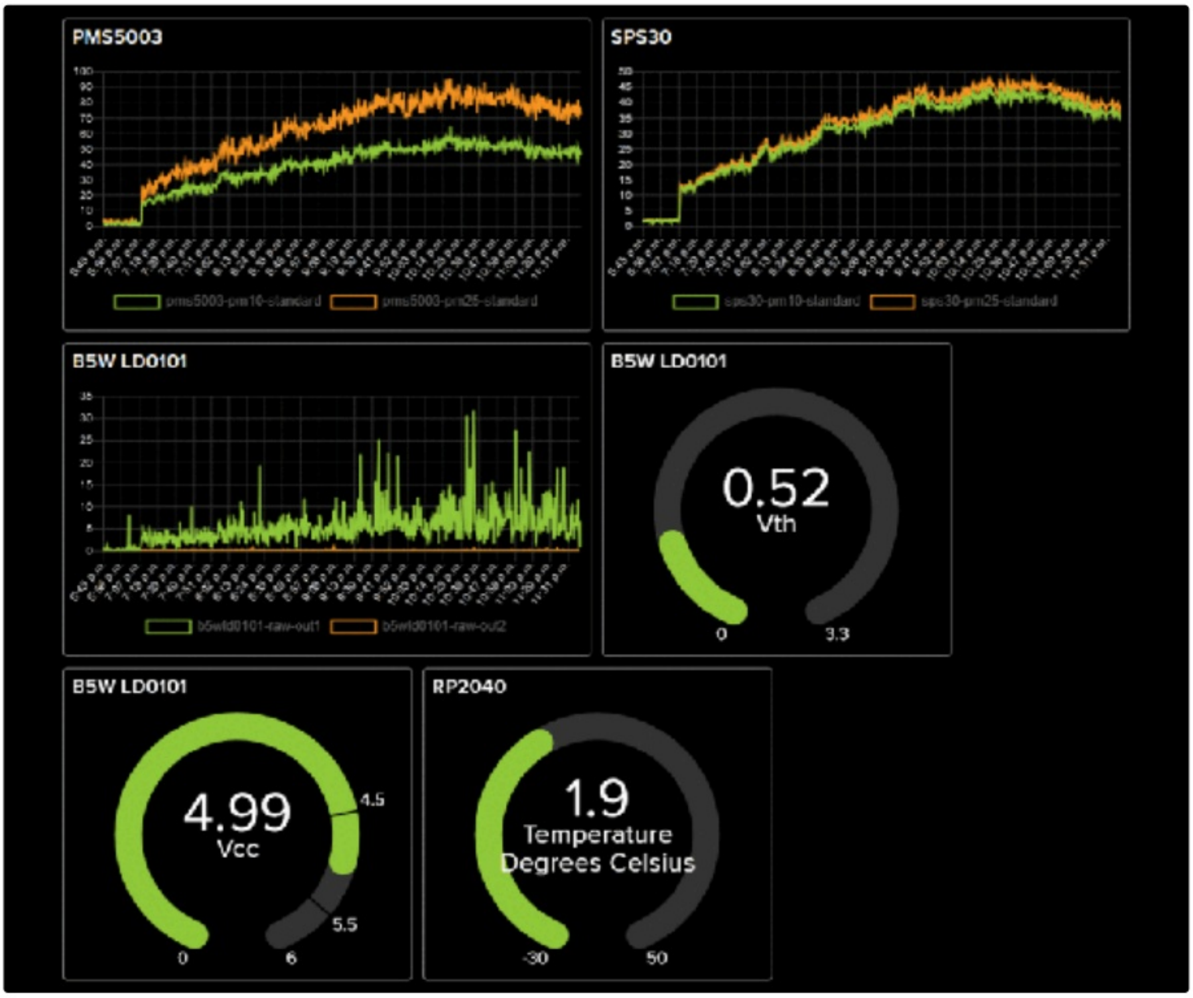
and is flunfortunately affected by the tungsten halogen lighting used for this video. There is an elevated level of particles in the air from a previous test run.
Remember to disconnect the battery pack when not in use otherwise the B5W LD0101's heater will drain the batteries.
https://www.youtube.com/watch?v=Ig5e6KOiMnA

Step 12: Particulate Matter Outside on Guy Fawkes Night

Guy Fawkes Night is associated with bonfires and fireworks which can contribute to an increase in air pollution for an evening or two. The charts above show the three sensors being placed outside just after 7pm on Friday 5th November 2021. There were no fireworks in the immediate vicinity but they could be heard in the distance. Note: the fly scale varies between the three charts.
The feed data stored in Adafruit IO shows the sensors detecting the air already had a slightly raised level of PM2.5 based on the SPS30 numbers:
2021/11/05 7:08:24PM 13.0941
2021/11/05 7:07:56PM 13.5417
2021/11/05 7:07:28PM 3.28779
2021/11/05 7:06:40PM 1.85779
The peak was around 46ug per cubic metre just before 11pm:
2021/11/05 10:55:49PM 46.1837
2021/11/05 10:55:21PM 45.8853
2021/11/05 10:54:53PM 46.0842
2021/11/05 10:54:26PM 44.8476
There are short spikes elsewhere in the data when the sensors were outside. These could be due to wafts from:

- exhaust from gas central heating,
- people smoking nearby and/or
- smells/fumes from cooking.

Check the weather before putting exposed electronics outside!



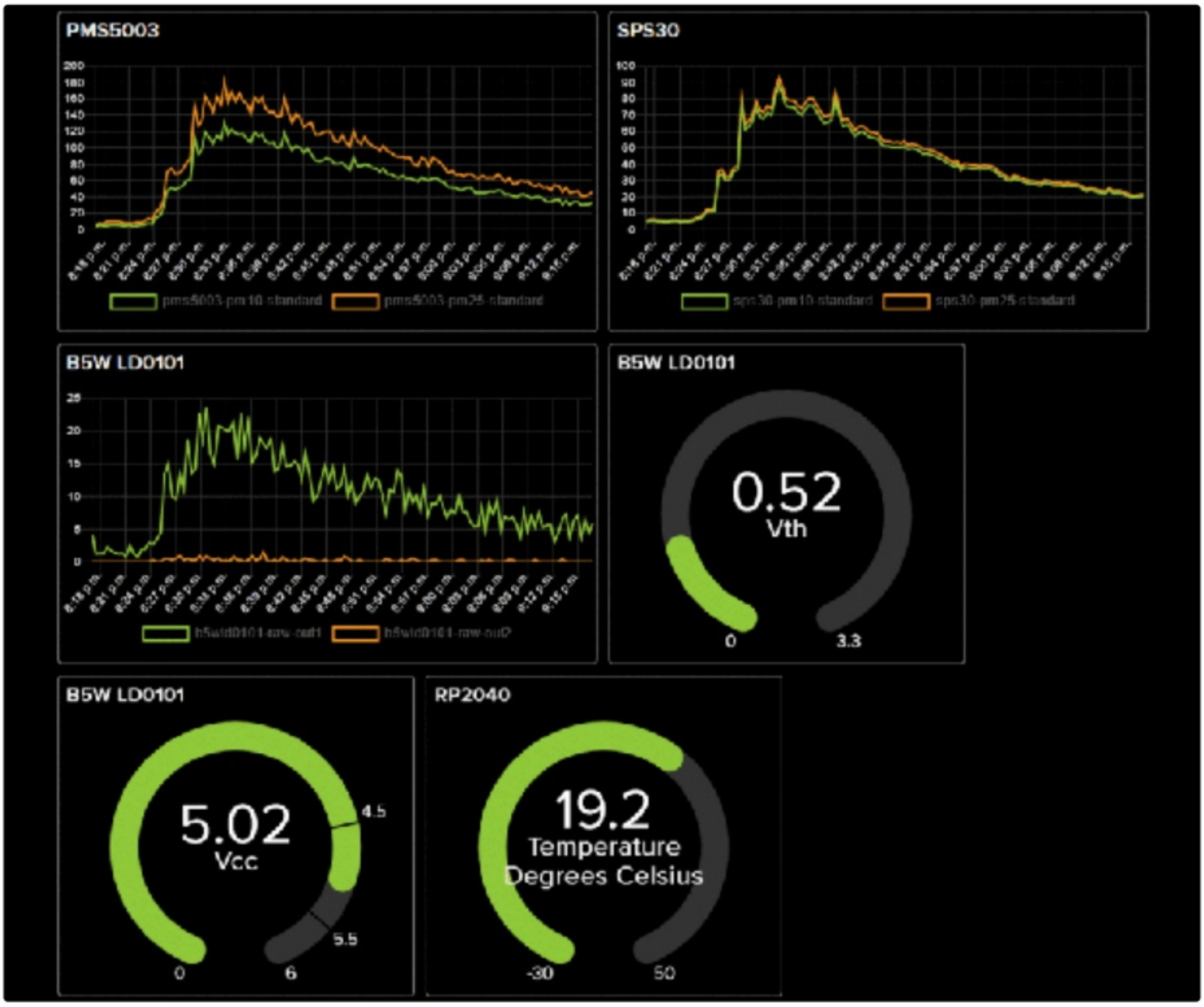
Step 13: Particulate Matter Inside With Cooking

The charts above show how the sensors react to bacon and mushrooms being fried in a nearby kitchen with mediocre extraction. The sensors were about 5m (16ft) from the hob. **Note:** the y scale varies between the three charts.

The feed data stored in Adafruit IO shows the sensors with a brief peak PM2.5 level of around 93ug per cubic metre based on the SPS30 numbers:

```
2021/11/07 8:33:52PM 79.6601
2021/11/07 8:33:24PM 87.386
2021/11/07 8:32:58PM 93.3676
2021/11/07 8:32:31PM 86.294
```

The pollutants will be very different to the ones from reworks. This is an interesting example of the varied sources of particulate matter in the air we breathe.



Step 14: Public Particulate Matter Sensors

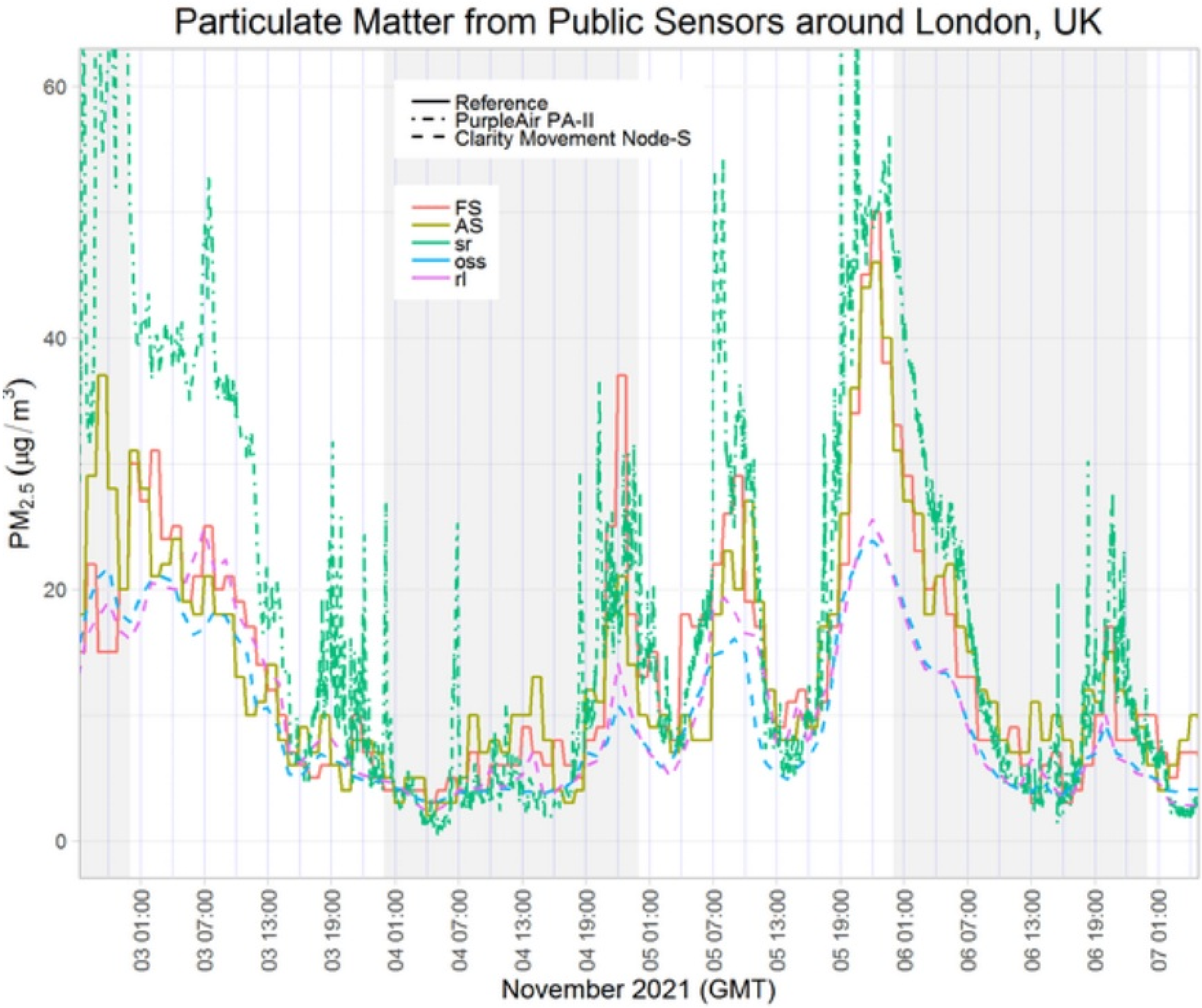
The data graphed above is from nearby public sensors.

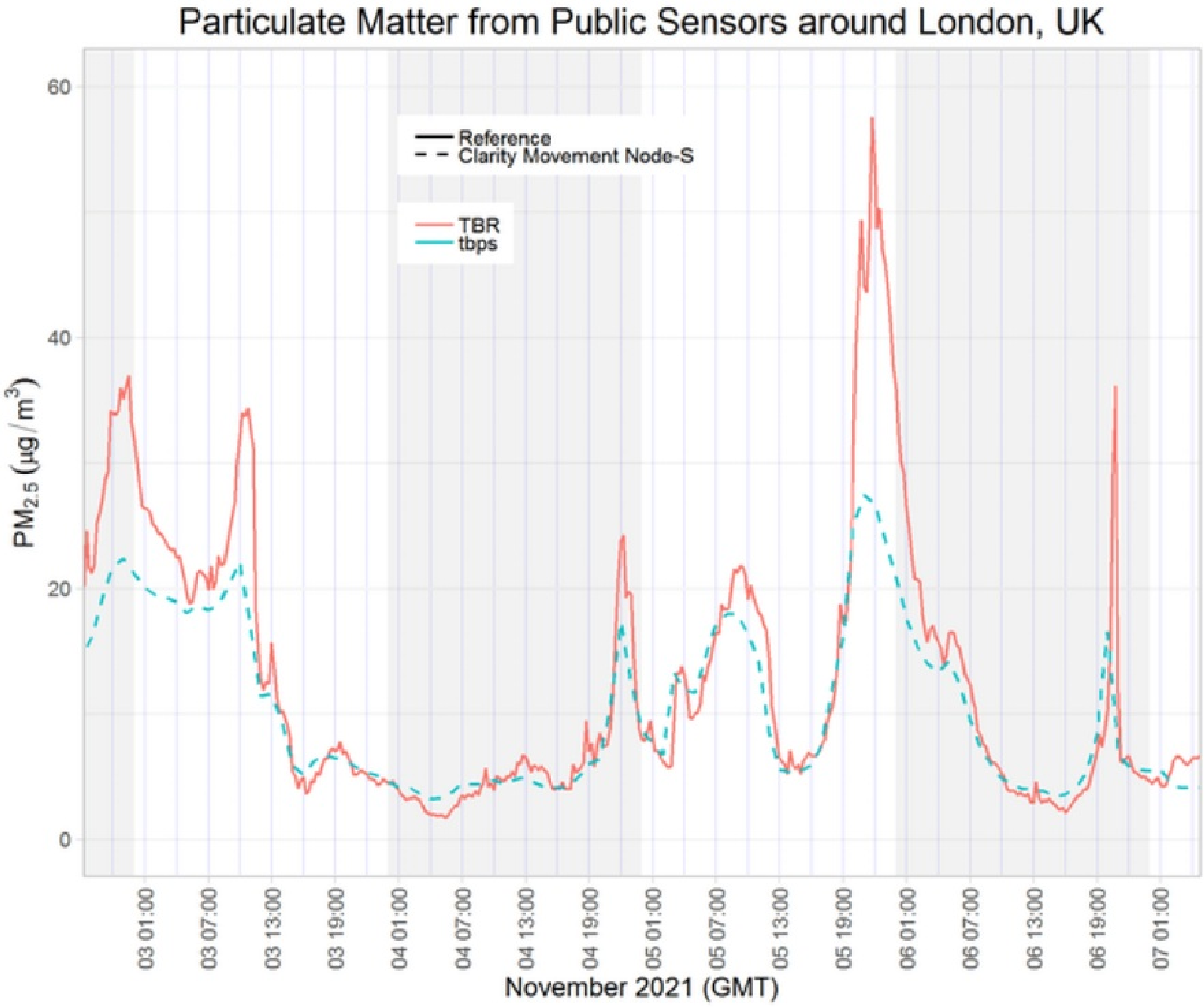
- Breathe London
    - Clarity Movement Node-S
        - tbps
        - oss
        - rl
- OpenAQ
    - PurpleAir PA-II
        - sr
- London Air Quality Network
    - Reference-quality (Met One BAM 1020 and others)
        - FS
        - AS
        - TBR

The tbps and TBR sensors are almost co-located and are graphed together to show the correlation between the SPS30- based device and the reference one nearby. The SPS30 appears to under-read significantly on the evenings of the 5th and 6th of November when it’s reasonable to assume the evening increase is due to fireworks. This could be due to the difference in the mass of the particles as the sensors used for this article can only detect volume and need to guess the density of the particles to produce values in micrograms per cubic metre.

The PMS5003 in the PurpleAir PA-II appears to over-read significantly for any elevated PM2.5 levels based on this short period. This could match the results shown on the previous pages or there could be other factors nearby causing this.

The SPS30 and PMS5003 produce data for particles larger than 2.5 microns but the following pages show why this should be treated with caution.





Step 15: Comparison of Sensors – Particle Size

The graphs above are from the Laboratory evaluation of particle-size selectivity of optical low-cost particulate matter sensors by the Finnish Meteorological Institute. Three sensors of each kind were tested with different particle sizes shown on the logarithmic x axis. The coloured lines indicate the calculated values of specific particle size bands based on the sensor outputs, the banding shows the distribution. The three SPS30 values above 1 micron overlap heavily making them very diffcult to distinguish.

The common metrics for particulates are PM2.5 and PM10. While the number in the name refers to the maximum size of the particle the units are in micrograms per cubic metre. The inexpensive sensors can only measure particle diameter (volume) and have to make some guesses about density to calculate the likely PM2.5 and PM10 values.

The PMS5003 uses a constant density value, Sensirion describe their density approach for the SPS30 as:

Most low-cost PM sensors on the market assume a constant mass density in calibration and calculate the mass concentration by multiplying the detected particle count by this mass density. This assumption only works if the sensor measures a single particle type (for instance, tobacco smoke), but in reality we nd many different particle types with many different optical properties in everyday life, from ‘heavy’ house dust to ‘light’ combustion particles. Sensirion’s proprietary algorithms use an advanced approach that allows a proper estimation of the massconcentration, regardless of the particle type measured. In addition, such an approach enables a correct estimation of the size bins.

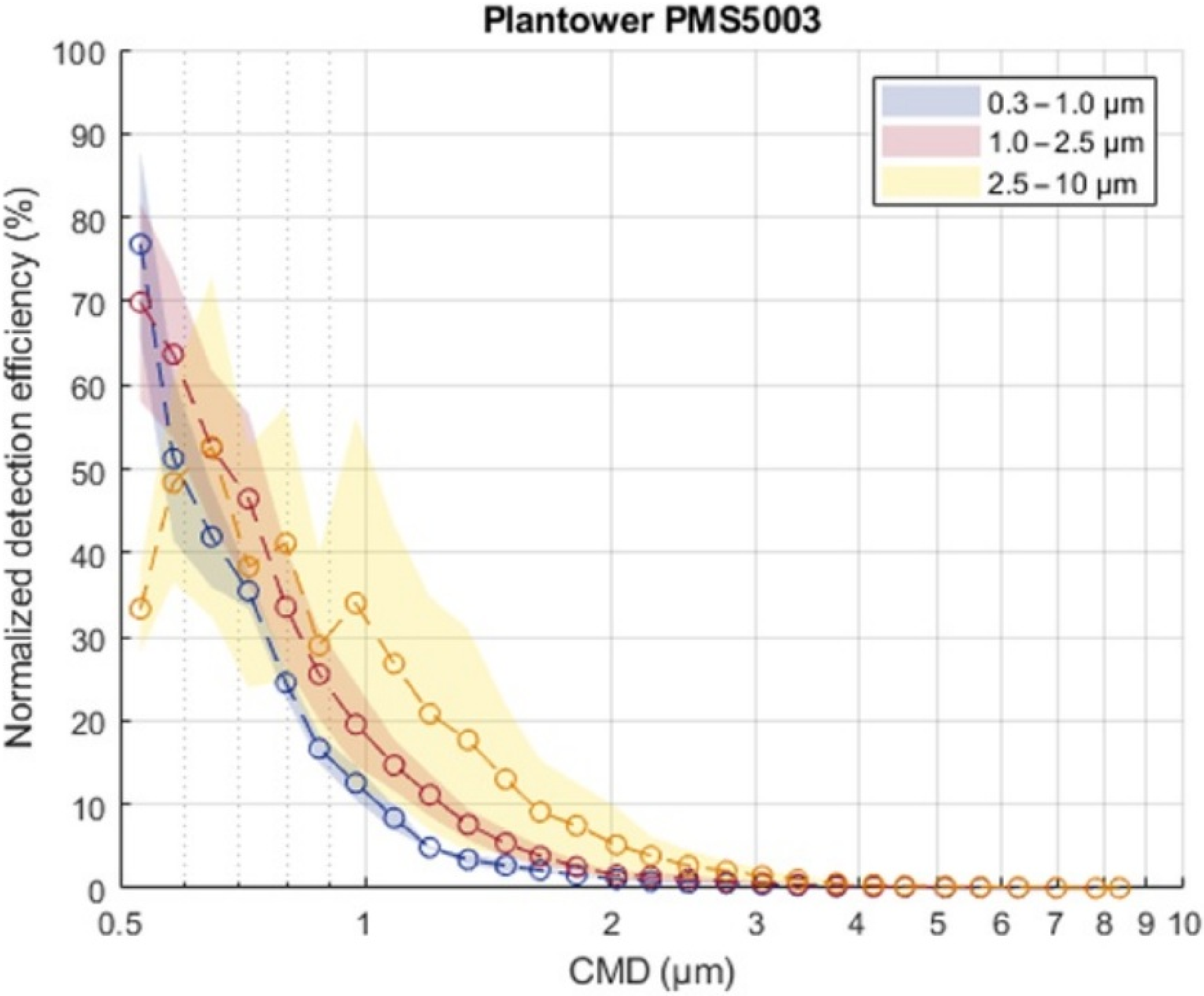
The PM metrics encompass all of the particles below the size parameter, i.e.

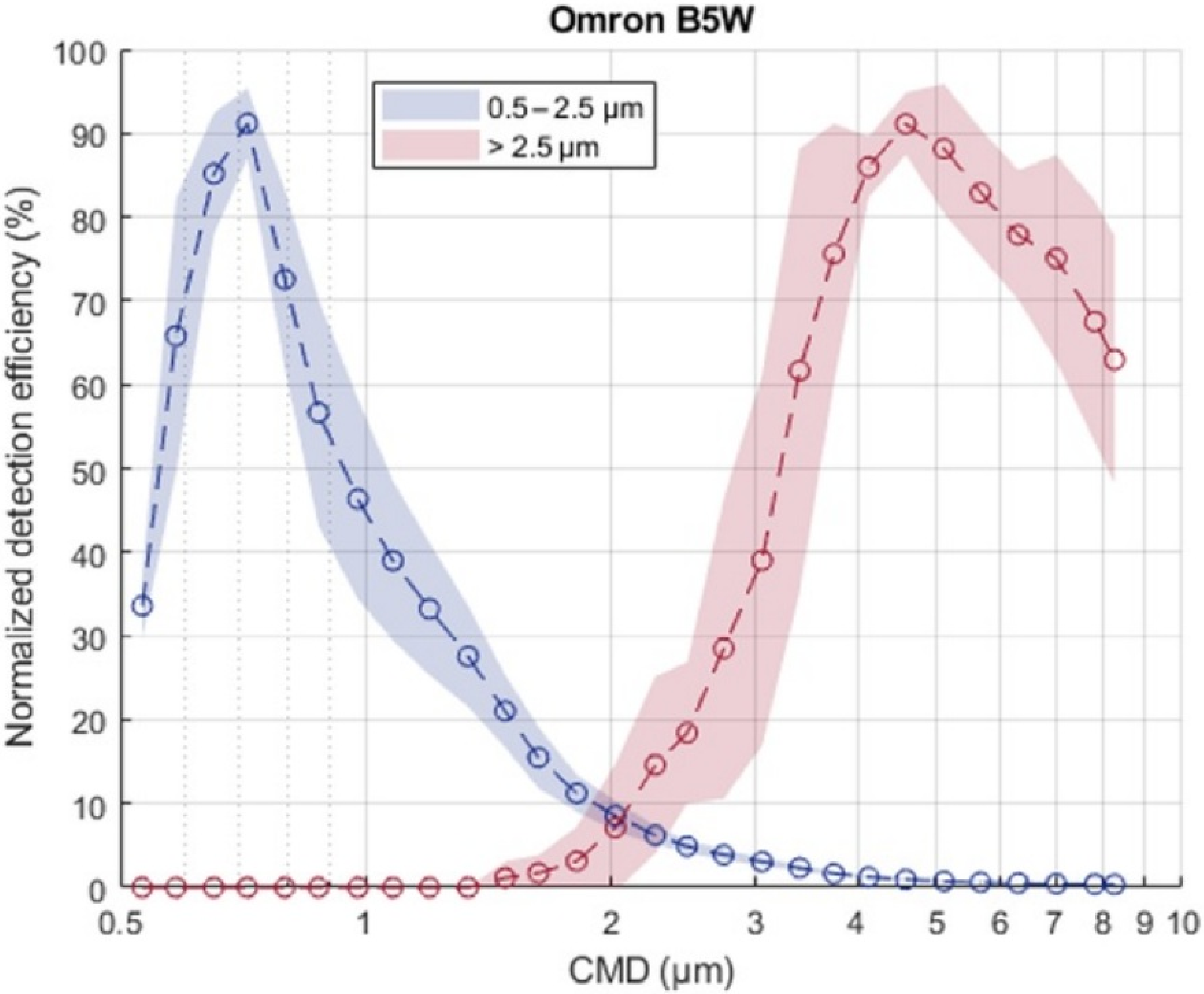
PM1 + mass of all the particles between 1.0 and 2.5 microns = PM2.5,

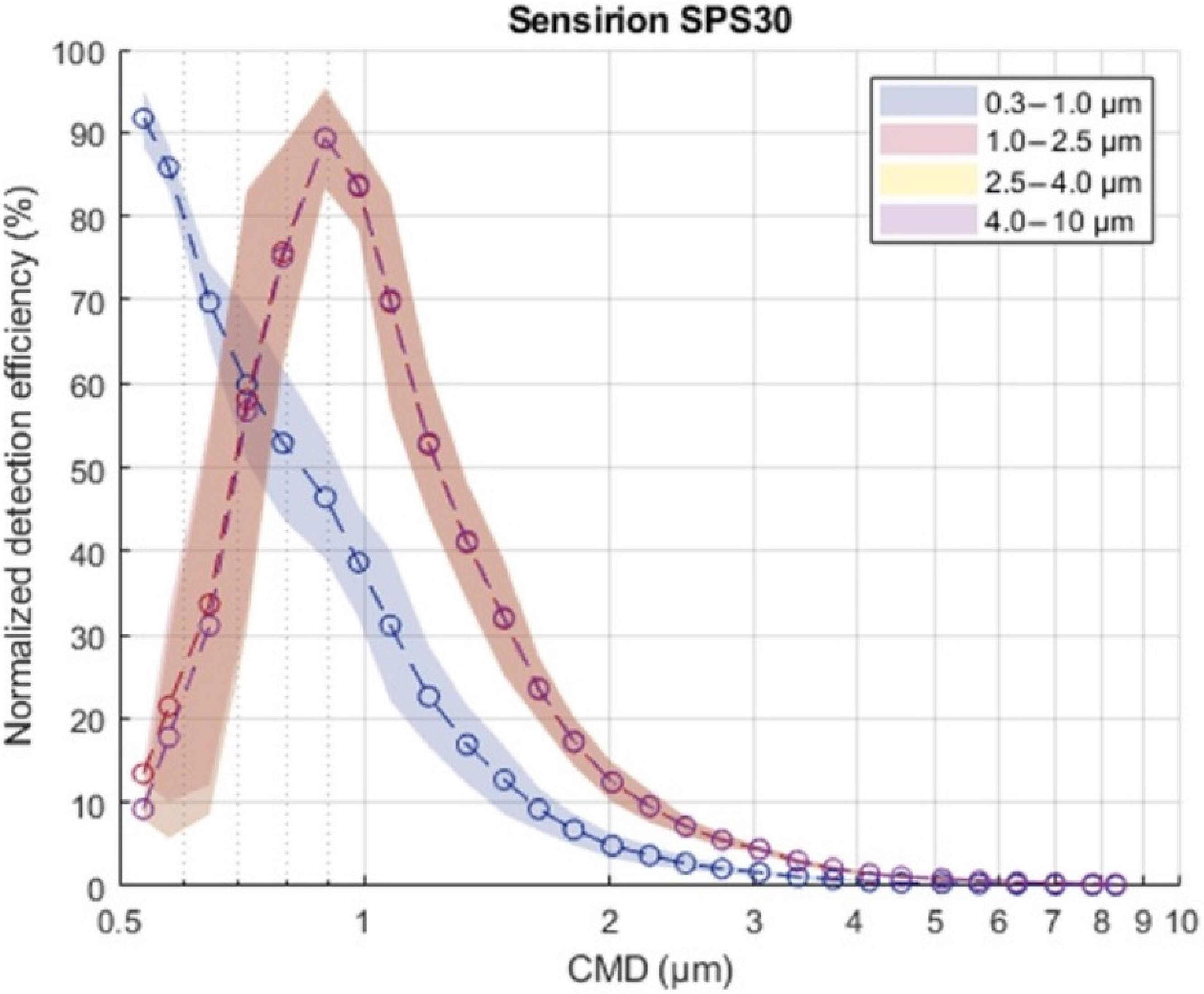
PM2.5 + mass of all the particles between 2.5 and 10 microns = PM10.

The PMS5003 and SPS30 are unable to detect the particles in this laboratory test above 2-3 microns. It’s possible they may detect other types of particles above this size.

The B5W LD0101 looks credible from this laboratory test for measuring PM10.


Plantower PMS5003
Normalized detection efficiency (%)
CMD (µm)
0.3 – 1.0 µm
1.0 – 2.5 µm
2.5 – 10 µm
0
10
20
30
40
50
60
70
80
90
100
0.5
1
2
3
4
5
6
7
8
9
10


Omron B5W
0.5 – 2.5 μm
> 2.5 μm
Normalized detection efficiency (%)
100
90
80
70
60
50
40
30
20
10
0
0.5
1
2
3
4
5
6
7
8
9
10
CMD (μm)

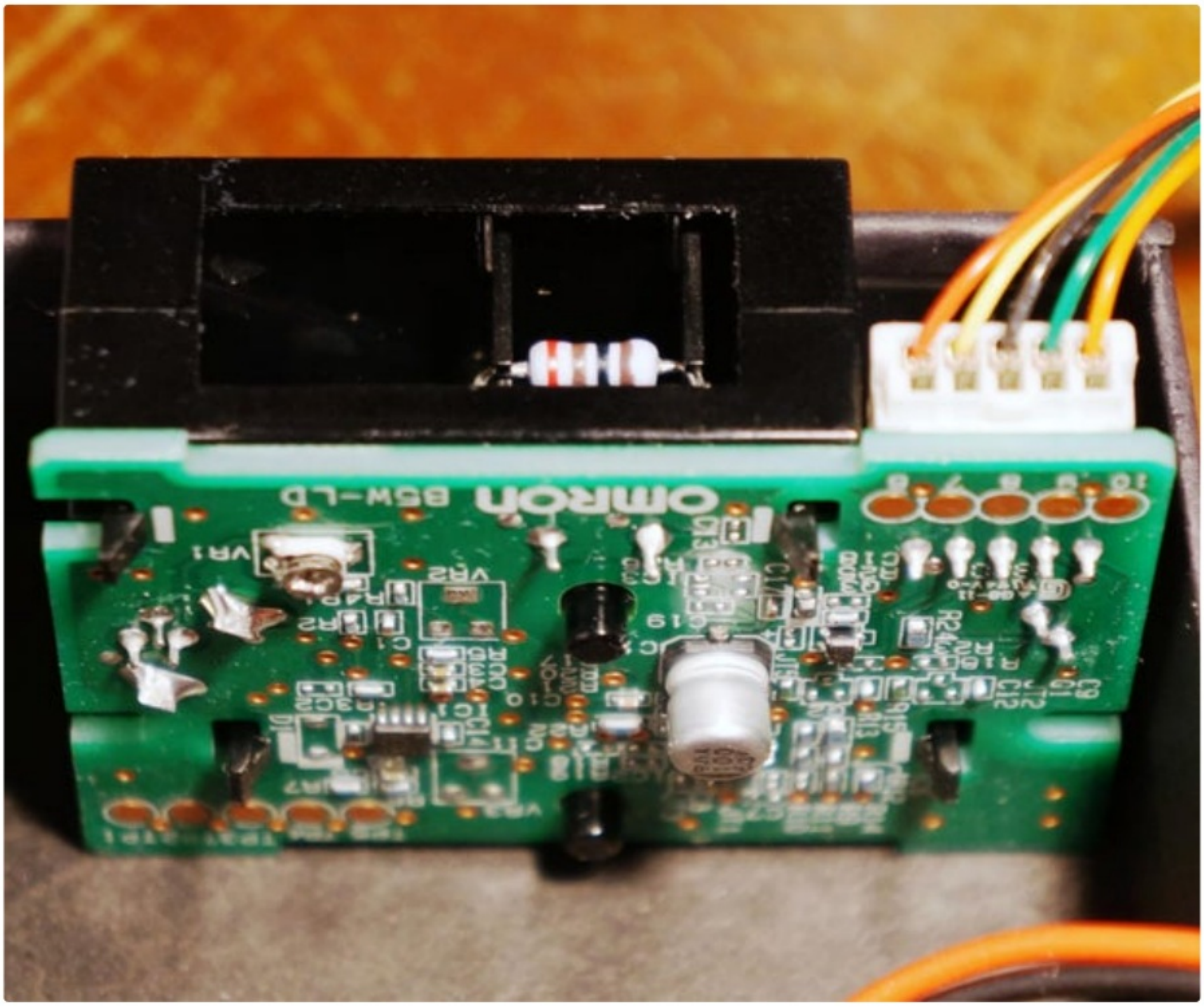


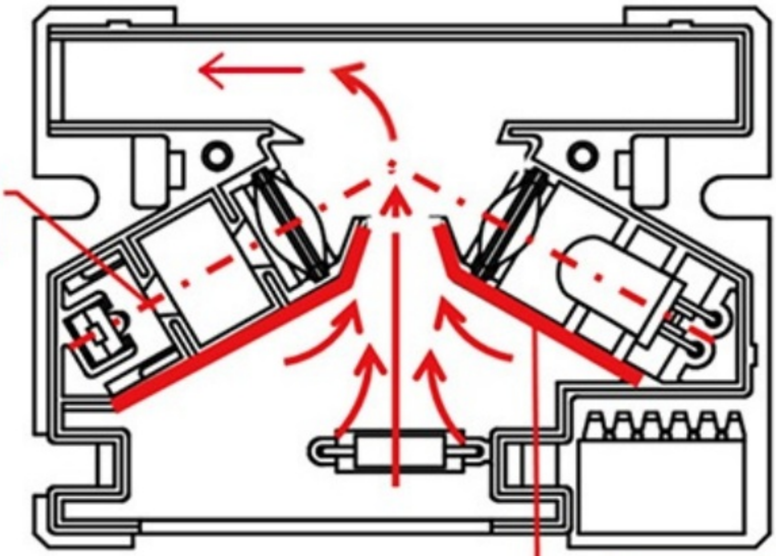
Step 16: Comparison of Sensors – Design

The Omron heater (a 100 ohm +/- 2% resistor!) can be seen if the sensor is turned upside-down. The design is discussed in detail in Omron: Development of air quality sensor for air puriffer . The use of convection seems crude but it can be a higher reliability solution compared to a mechanical component like a fan which has a finite lifetime and a lifetime that may be reduced by operating in a dusty environment. The SPS30 fan appears to be designed to be easily replaceable without opening the case. Other Plantower models have the same design feature.
All three of the sensors will be prone to the effects of high relative humidity which unfortunately erroneously increases the PM values.
The certiffed, reference-quality sensors (UK's DEFRA list) which monitor particulate matter do not use an optical approach for measurement. The Met One BAM 1020 works by

1. separating and discarding the particles larger than the size limit from the air sample,
2. heating the air to control/reduce the relative humidity,
3. depositing the particles on a new section of a continuous fibrous tape and
4. then measuring the attenuation of a beta radiation source by the accumulated particles on the tape to calculate a good estimate of the total mass of the particles.

Another common technique is the Tapered Element Oscillating Microbalance (TEOM) which deposits particles on a replaceable filter on the free end of a tapered tube which is fixed at the other end. The accurate measurement of the oscillation frequency of the naturally-resonant tube allows the additional tiny mass of the particles to be calculated from the miniscule variation in frequency. This approach is suitable for creating higher rate PM values.

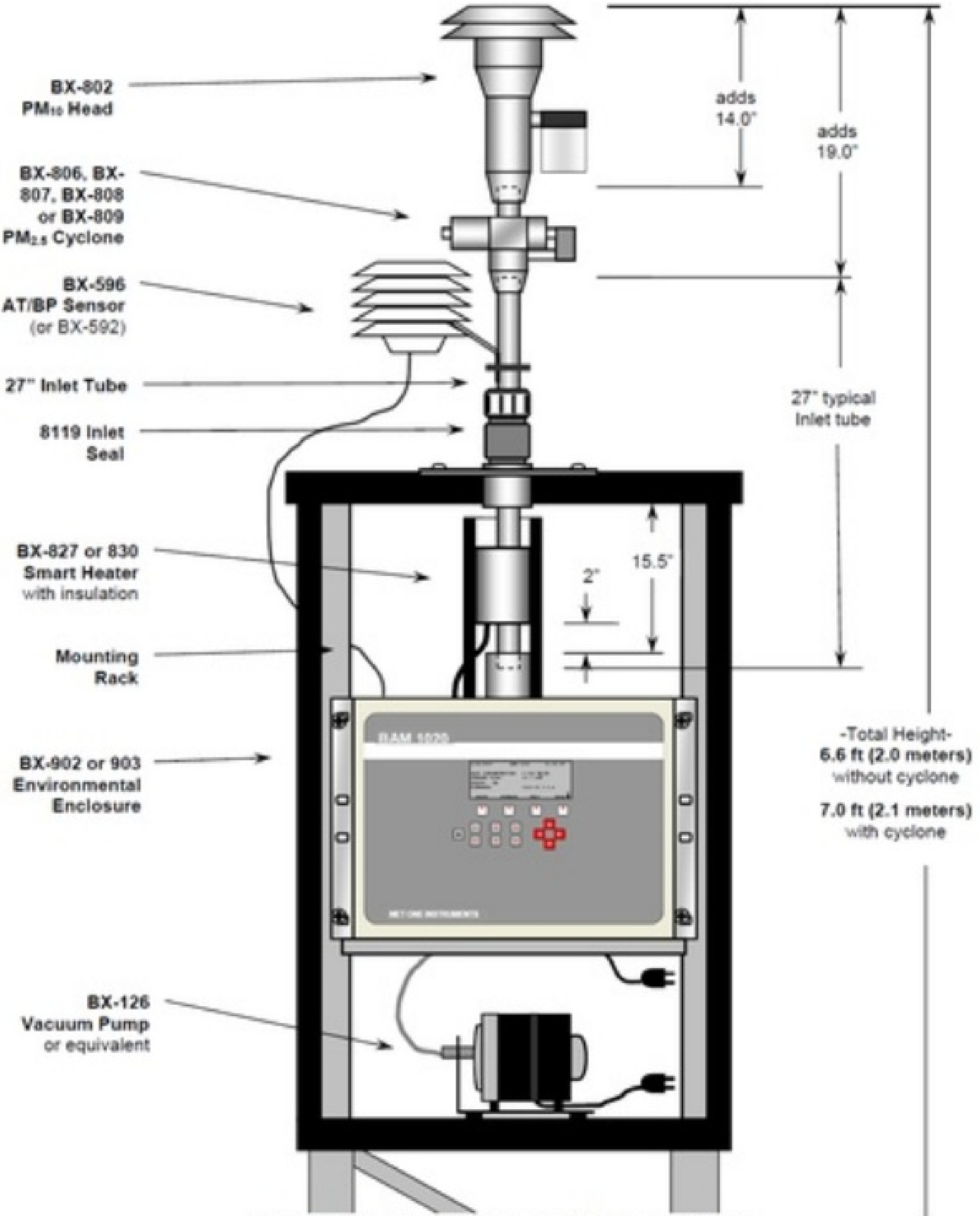




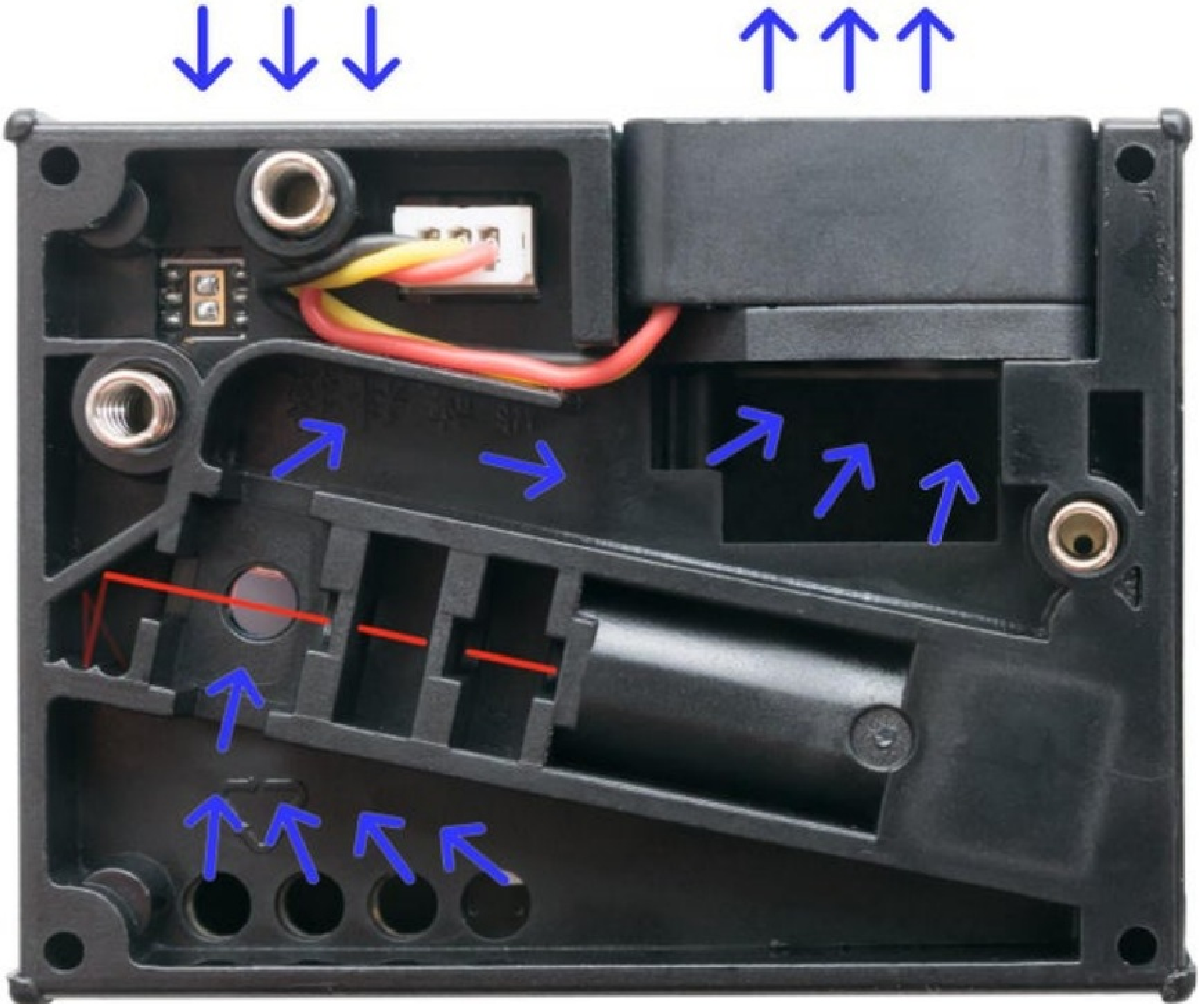
Light axis

The funnel-shaped structure was formed using the optical shape of projected and received light

Met One
Instruments Inc.



Typical BAM 1020 Installation in a BX-902 mini enclosure



Step 17: Going Further

Once you’ve setup your sensors and are publishing data to Adafruit IO, here are some other ideas to explore:

- Test each room in your home over time noting the activity and ventilation. Test your home when you’re cooking. Test a barbeque.
- Use the three buttons on the Maker Pi Pico. These are connected to GP20, GP21 and GP22 which were intentionally left unused to allow for button use.
- If you live near a public air quality monitoring station compare your data with it.
- Add a display for attended use showing sensor values. The SSD1306 is small, a ordable and easy to add/use in CircuitPython. See Instructables: Soil Moisture Sensing
- With the Maker Pi Pico for an example of its use.
- Investigate the MQTT library to see if all of the sensor data can be sent in one batch. This should be more effcient.
- Integrate in some way with the standalone IKEA Vindriktning Air Quality Sensor.
    - Soren Beye’s MQTT connectivity for the Ikea VINDRIKTNING shows how to add an ESP8266 to the sensor and identifies the particulate matter (dust) sensor as “Cubic PM1006-like”.
    - An advanced project would be to replace the main PCB with an ESP32-S2 based board with extra digital environmental sensors to create a Wi-Fi-enabled, CircuitPython-based device.
    - This device is discussed on Home Assistant Forum: IKEA Vindriktning Air Quality Sensor.
    - LaskaKit produce an ESP32-based replacement PCB for the sensor to allow it to be easily used with

ESPHome.

- Study the effects of varying the supply voltage within the permitted ranges for the sensors. This might change the fan speed or the temperature of the heater affecting the results.
- Build a weather and wildlife proof enclosure with careful design for air inlet, outlet and air flow past sensors. An umbrella taped to a railing was used to protect the open, exposed electronics for thecollection of data over the weekend for this article.

Related Projects:

- Costas Vav: Portable Air Quality Sensor
- Pimoroni: An outdoor air quality station with Enviro+ and Luftdaten
- Instructables: Using the Pimoroni Enviro+ FeatherWing With the Adafruit Feather NRF52840 Express – the
- Enviro+ FeatherWing includes a connector for the PMS5003. The SPS30 can be used with i2c pins and there are just about enough pins to use the B5W LD0101 too.
- The nRF52840 does not support Wi-Fi so this cannot be used on its own to publish data over the Internet.
- Adafruit Learn: Air Quality Sensor 3D Printed Enclosure . – uses the Adafruit Feather M4 with ESP32-based Airlift FeatherWing and PMS5003.
- Adafruit Learn: Quickstart IoT – Raspberry Pi Pico RP2040 with WiFi – uses an ESP32-based Adafruit AirLift breakout board.
- GitHub: CytronTechnologies/MAKER-PI-PICO Example Code/CircuitPython/IoT – example code for Adafruit IO, Blynk and Thinkspeak.
- Cytron: Air Monitoring Using Mobile Phone – uses an ESP8266-based Arduino shield to send data from a
- Honeywell HPM32322550 particulate matter sensor to Blynk, no (smart)phone required.

Intermediate sensors, more expensive but with better ability to detect larger particle sizes:

- Piera Systems IPS-7100
- Alphasense OPC-N3 and OPC-R2

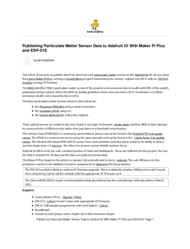
Further Reading:

- Sensors
  - Finnish Meteorological Institute: Laboratory evaluation of particle-size selectivity of optical low-cost particulate matter sensors (May 2020)
  - Gough Lui: Review, Teardown: Plantower PMS5003 Laser Particulate Monitor Sensor includes a comparison with Sensirion SPS30.
  - Karl Koerner: How to Open and Clean a PMS 5003 Air Sensor
  - Met One Instruments, Inc., BAM-1020 EPA TSA Training Video (YouTube) – shows what’s inside and how it works.
  - CITRIS Research Exchange: Sean Wihera (Clarity Movement) talk (YouTube) – talk including details on the Node-S sensor which uses the Sensirion SPS30.
- Legislation and Organisations involved with air quality
  - The Air Quality Standards Regulations 2010 (UK)
  - World Health Organisation (WHO) Air Pollution Guidelines

    - British Lung Foundation – Air Quality (PM2.5 and NO2)
- Research
    - Imperial College London: The Indoor-Outdoor Air-Pollution Continuum (YouTube)
    - Primary school children gathering air quality data using backpacks in London in 2019:
        - Dyson: Tracking pollution on the school run. Breathe London (YouTube)
        - King's College London: Environmental Research Group: The Breathe London Wearables Study
    - Atmosphere Journal: Indoor Air Pollution from Residential Stoves: Examining the Flooding of Particulate Matter into Homes during Real-World Use
- News and Blogs
    - The Economist: Midnight sky – Poland's coal-fired home heating creates widespread pollution (January 2021)
    - US NPR: Sheltering Inside May Not Protect You From The Dangers Of Wildfire Smoke ?
    - Reuters: The party's over: Diwali leaves Delhi wheezing in dangerously unhealthy air
    - Pimoroni Blog: The Most Polluted Night of the Year (in UK)
    - Clarity Movement: Wild FIre Smoke, Public Health, and Environmental Justice: Better
    - Decision Making with Air Monitoring (YouTube) – presentation and discussion on air quality of the western US particularly around 2020 wild FIre smoke.
    - Guardian: Dirty air affects 97% of UK homes, data shows
- Particulate Monitoring and data warehousing
    - Netherlands Rijksinstituut voor Volksgezondheid en Milieu (National Institute for Public Health and the Environment): Vuurwerkexperiment (Fireworks Experiment) 2018-2019
    - Google: Street by street: How we're mapping air quality in Europe – street view cars collect particulate matter and pollutant gas data.London Air Quality Network
    - Breathe London – a network to supplement the London Air Quality Network with "affordable, easy-to-install and maintain air quality sensors to anyone", currently using the Clarity Movement Node-S.
    - US Embassy in Beijing particulate matter monitoring (Twitter)
    - World Air Quality Index – collects data from many diFFerent sources with map views and historical data.
    - Sensor.Community (formerly known as Luftdaten) – "making the world a better place through community driven, open environmental data".
- Software Libraries
    - Software Bugs in a Particulate Matter Sensor Library – the adafruit_pm25 suFFers from at least one of the issues described necessitating exception handling around read() for serial (UART).
- Courses
    - HarvardX: Particulate matter air pollution (YouTube) – five minute video from the short course EdX: Energy Within Environmental Constraints

Safety critical detection and alarms are best left to commercial appliances from reputable suppliers.
https://www.youtube.com/watch?v=A5R8osNXGyo
Publishing Particulate Matter Sensor Data to Adafruit IO With Maker Pi Pico and ESP-01S:

Documents / Resources

	**instructables ESP-01S Publishing Particulate Matter Sensor** [pdf] User Guide ESP-01S Publishing Particulate Matter Sensor, ESP-01S, Publishing Particulate Matter Sensor, Particulate Matter Sensor, Matter Sensor

References

- ESP-01
- Full list of Timezones, via World Time API: Simple JSON/plain-text API to obtain the current time in, and related data about, a timezone.
- Yours for the making - Instructables
- Kevinjwalters's Profile - Instructables
- Publishing Particulate Matter Sensor Data to Adafruit IO With Maker Pi Pico and ESP-01S : 17 Steps (with Pictures) - Instructables
- Beijing Air Pollution: Real-time Air Quality Index
- Street by street: How we're mapping air quality in Europe
- The Most Polluted Night of the Year
- Blynk: a low-code IoT software platform for businesses and developers
- CircuitPython
- Libraries
- adafruit_pm25 — Adafruit PM25 Library 1.0 documentation
- IKEA Vindriktning Air Quality Sensor - #3 by GSzabados - Share your Projects! - Home Assistant Community
- Beta particle - Wikipedia
- Convection - Wikipedia
- Guy Fawkes Night - Wikipedia
- Hayes AT command set - Wikipedia
- Humidity - Wikipedia
- Internet Control Message Protocol - Wikipedia
- Micrometre - Wikipedia
- Modem - Wikipedia
- Particulates - Wikipedia
- Round-trip delay - Wikipedia
- Smoke detector - Wikipedia

- W Switch - Wikipedia
- W Voltage divider - Wikipedia
- ESPHome — ESPHome
- GitHub - adafruit/Adafruit_CircuitPython_PM25: CircuitPython library for PM2.5 sensors
- GitHub - CytronTechnologies/esp-at-binaries: Custom compiled ESP-AT firmware binaries.
- esp-at-binaries/Cytron_ESP-01S_AT_Firmware_V2.2.0.bin at main · CytronTechnologies/esp-at-binaries · GitHub
- MAKER-PI-PICO/simpletest.py at main · CytronTechnologies/MAKER-PI-PICO · GitHub
- MAKER-PI-PICO/Example Code/CircuitPython/IoT at main · CytronTechnologies/MAKER-PI-PICO · GitHub
- GitHub - espressif/esptool: Espressif SoC serial bootloader utility
- GitHub - Hypfer/esp8266-vindriktning-particle-sensor: Wifi MQTT Data Logging via an esp8266 for the Ikea VINDRIKTNING PM2.5 air quality sensor
- GitHub - kevinjwalters/Adafruit_CircuitPython_SPS30: CircuitPython library for Sensirion SPS30 sensor