ENGINEERING FOR impact

mit 6.900

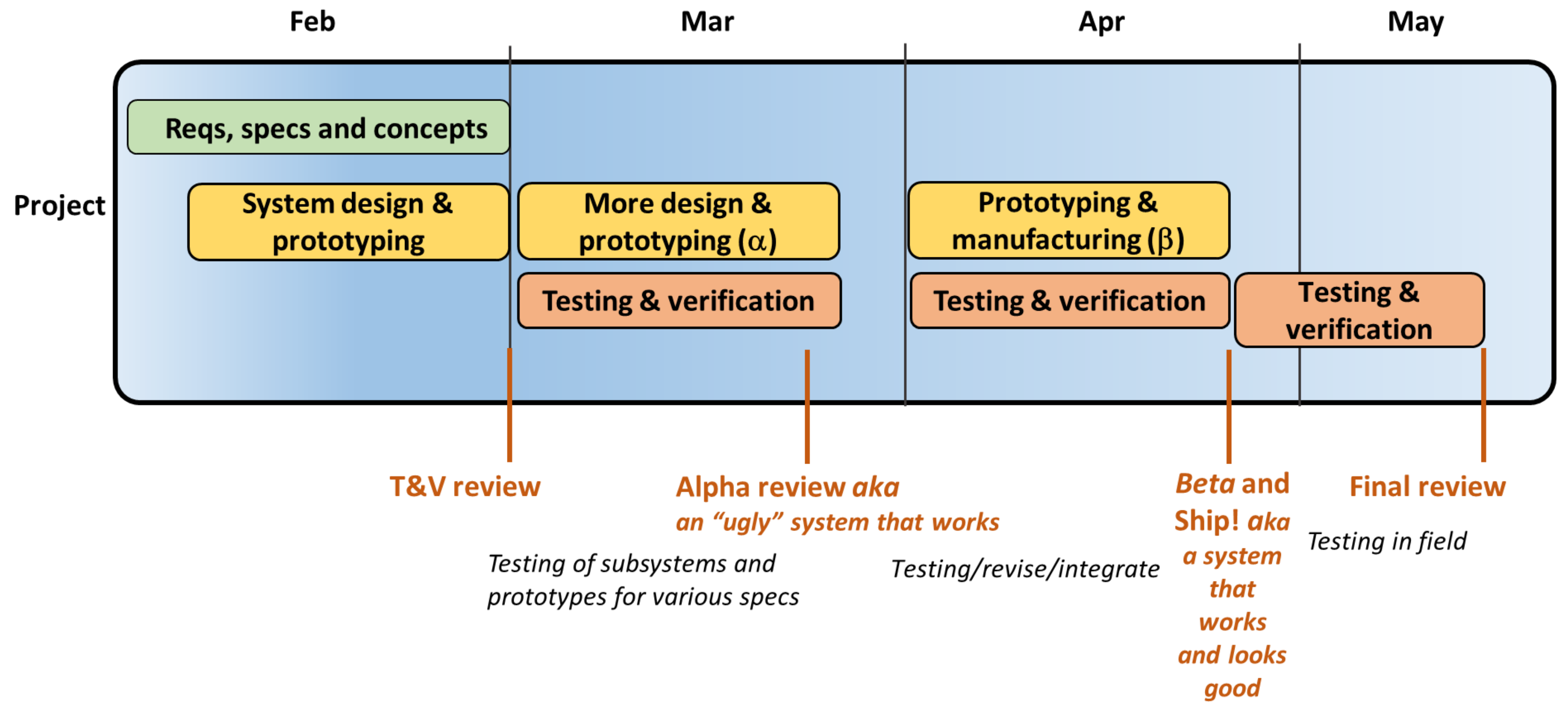
Lecture 3

February 13, 2024

Make something real.

This week

- EX02
  - Sensors PCB schematic design
  - Plotly
  - Users guide to me
  - Specs
- Lab 2
  - First team meeting!
  - Lots to do!


Feb
Mar
Apr
May
Project
Reqs, specs and concepts
System design & prototyping
More design & prototyping (α)
Testing & verification
Prototyping & manufacturing (β)
Testing & verification
Testing & verification
T&V review
Alpha review *aka an "ugly" system that works*
Testing of subsystems and prototypes for various specs
Testing/revise/integrate
Beta and Ship! *aka a system that works and looks good*
Final review
Testing in field

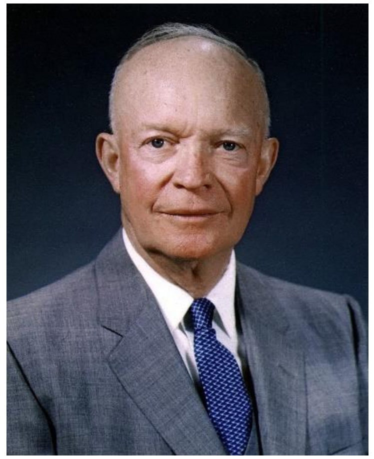
Teams

- Sent out this morning
- We may adjust over the next week or two
- Teams are not competitors – work together!

TODAY

- The HW/SW product development process *with a focus on engineering design*
- MAQS & Sentimet

"In preparing for battle I have always found that plans are useless, but planning is indispensable"
--Dwight D. Eisenhower



MAQS: MIT Air Quality System requirements

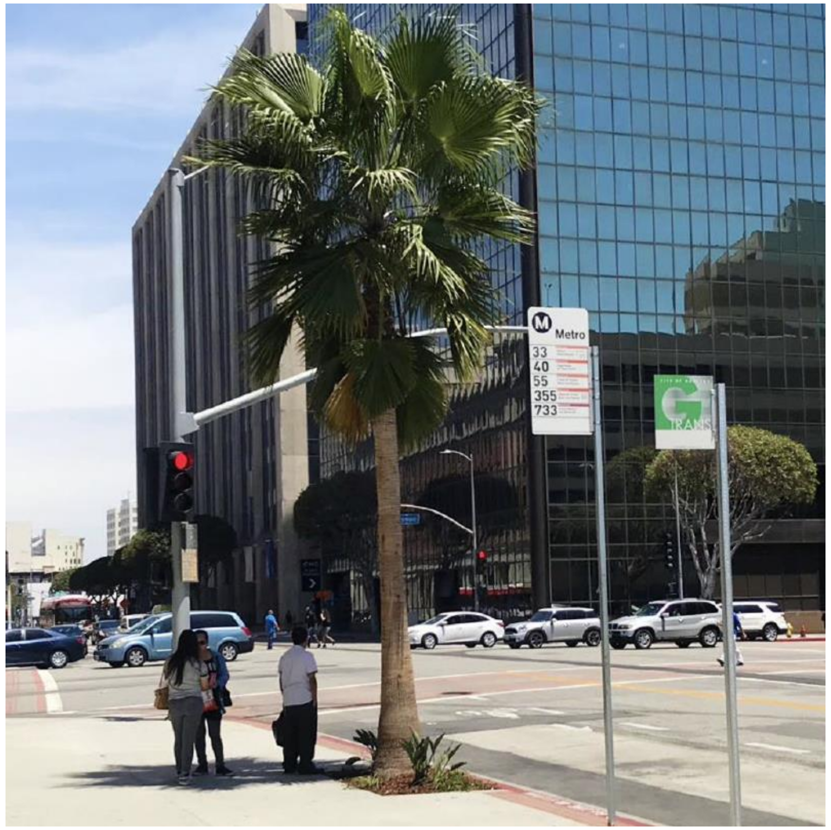
We developed requirements based on internal staff discussions and talking with EDS staff

1. It should accurately measure indoor air quality **
2. It should be portable ***
3. It should be possible to get the data off the device **
4. It should be a useful pedagogical exercise ***
5. It should maintain privacy *
6. It should be low cost *
7. It should be rugged and robust **
8. Multiple systems should be able to be used simultaneously ***
9. It should be easy to view the current and past data **
10. It should leverage MIT facilities **

Sentimet

- Two very similar needs

MIT Office of Sustainability



Miami-Dade County

Sentimet

MITOS

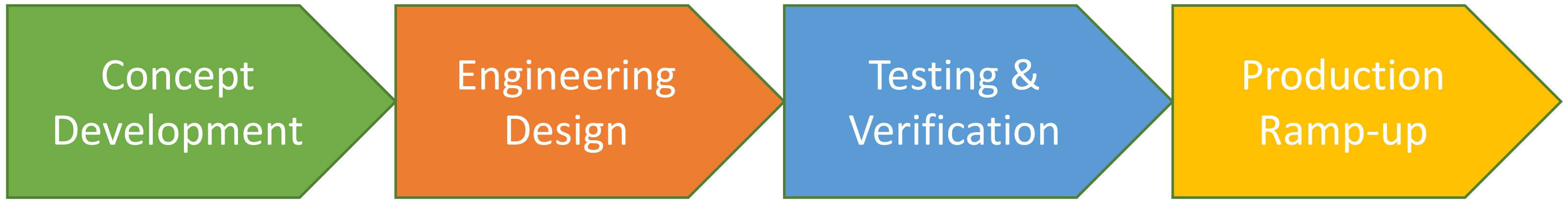
1. It should measure the local weather, at least temperature and humidity and ideally also sun exposure and ground surface temperature and air pressure, all with dynamics appropriate for the use case.***
2. It should be able to measure how many people are in the area passing through (e.g., foot traffic) and lingering.***
3. It should operate without being connected to line voltage. ***
4. It should be portable and able to be set up by an average person in a variety of outdoor environments on the MIT campus, including on a tripod or attached to poles of various dimensions.***
5. It should be able to be physically attached to a HOBO MX2302A data logger.*
6. It should report faults, such as battery failure, falling, vandalism, etc.**
7. It should be as inexpensive as possible. *
8. Data from a sensor node should be able to be tied to a location.***
9. It should maintain privacy. ***
10. It should operate independently without user intervention for 2+ weeks.***
11. It should be rugged and able to withstand a summertime Boston-area environment (heat, rain, wind and curious people). ***
12. Multiple systems should be able to be used simultaneously. ***
13. The system should present the information on a dashboard (with real-time data outputs to a dashboard if possible), and also allow downloading of raw data.***

Miami-Dade

1. It should measure the "heat experience" at each bus stop, at least temperature and humidity, but also could include air quality, all with dynamics appropriate to the use case.***
2. It should be able to measure how many people are waiting, and for how long.***
3. It should operate without being connected to line voltage. ***
4. It should be installable by a technician, and should be easy to set up.***
5. It should report faults, such as battery failure, falling, vandalism, etc.**
6. It should be as inexpensive as possible. *
7. Data from a sensor node should be able to be tied to a location.***
8. It should maintain privacy. ***
9. It should operate independently without user intervention for at least a month.***
10. It should be rugged and able to withstand shipping, setup, and operation in the Miami-Dade environment. ***
11. Multiple systems should be able to be used simultaneously. ***
12. The system should incorporate data from Swift.ly and present that information to the operator in a useful way. ***

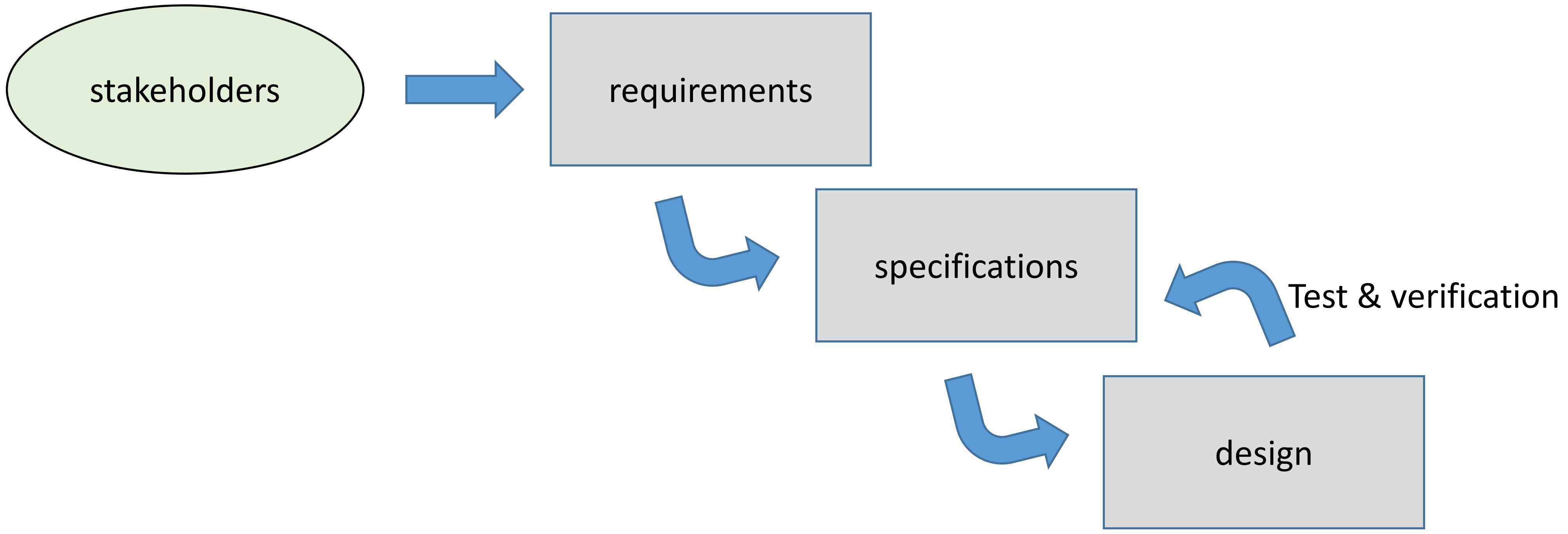
Mostly the same…could make one overall system, or two slightly different systems

Now what?



- We iterate between developing concepts, setting specs, doing design, and developing a testing plan
  - Because Sentimet already has v1 from last year, we can start from that concept and keep/change it
- Not everything will necessarily be defined yet (and thus able to be concretely specified)
  - This may not be ok for some products (aerospace, medical) but often the case for consumer, etc. ➔ Iteration can be important
- HW and SW are specified differently
  - We'll see this

Requirements, specs, and so on



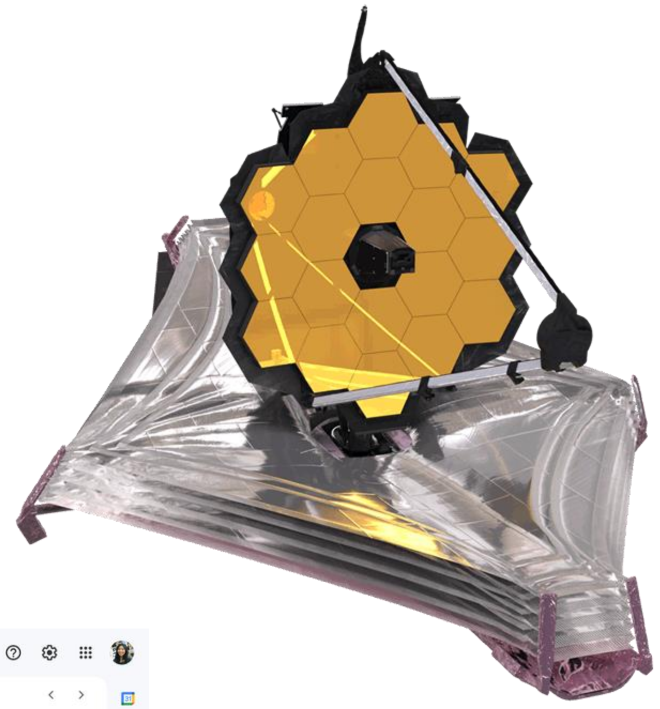
Done well, a design that passes all the tests will meet the specifications and thus the requirements, making the stakeholders happy

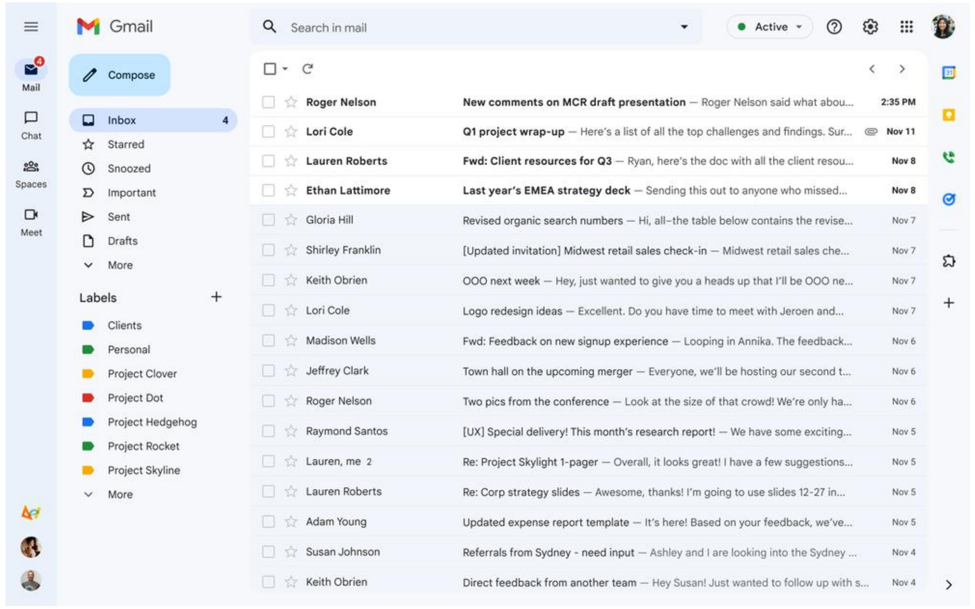
Specifications

- Translate requirements into a specification document
  - Covers both HW and SW aspects
  - The goal is to have constraints for our system→ *engineering is design under constraints*
  - If our specs are complete, and if we build something that meets spec, then it will meet our requirements
  - Some specifications will directly imply a specific approach
    - The need for interoperability with another product (that has WiFi) may immediately specify WiFi
    - The company may impose use of MySQL dB or STM32 MCU because that's what's used by the rest of the company
  - Sometimes specifications will be much more open-ended
    - An opportunity for creativity & innovation!

Specifications

- What aspects do we need to specify?
  - There is no all-encompassing approach
  - These products are all specified differently





Roger Nelson	New comments on MCR draft presentation — Roger Nelson said what abou...	2:35 PM
Lori Cole	Q1 project wrap-up — Here's a list of all the top challenges and findings. Sur...	Nov 11
Lauren Roberts	Fwd: Client resources for Q3 — Ryan, here's the doc with all the client resou...	Nov 8
Ethan Lattimore	Last year's EMEA strategy deck — Sending this out to anyone who missed...	Nov 8
Gloria Hill	Revised organic search numbers — Hi, all–the table below contains the revise...	Nov 7
Shirley Franklin	[Updated invitation] Midwest retail sales check-in — Midwest retail sales che...	Nov 7
Keith Obrien	OOO next week — Hey, just wanted to give you a heads up that I'll be OOO ne...	Nov 7
Lori Cole	Logo redesign ideas — Excellent. Do you have time to meet with Jeroen and...	Nov 7
Madison Wells	Fwd: Feedback on new signup experience — Looping in Annika. The feedback...	Nov 6
Jeffrey Clark	Town hall on the upcoming merger — Everyone, we'll be hosting our second t...	Nov 6
Roger Nelson	Two pics from the conference — Look at the size of that crowd! We're only ha...	Nov 6
Raymond Santos	[UX] Special delivery! This month's research report! — We have some exciting...	Nov 5
Lauren, me 2	Re: Project Skylight 1-pager — Overall, it looks great! I have a few suggestions...	Nov 5
Lauren Roberts	Re: Corp strategy slides — Awesome, thanks! I'm going to use slides 12-27 in...	Nov 5
Adam Young	Updated expense report template — It's here! Based on your feedback, we've...	Nov 5
Susan Johnson	Referrals from Sydney - need input — Ashley and I are looking into the Sydney ...	Nov 4
Keith Obrien	Direct feedback from another team — Hey Susan! Just wanted to follow up with s...	Nov 4

Common specifications for HW/SW products

Typically,

- Financial
  - BOM, COGS, etc.
  - Time to market
- Regulatory – safety, emissions
  - Anything with a radio, plugged into wall, etc.
  - For medical (and other regulated sectors) this can be quite extensive
- Industrial design
  - What does it look like, what materials are used, how does it interact with the user, etc.
- Environmental resistance
  - Is it used indoors? In salt water? In an auto engine? On Mars?
  - IP [Ingress Protection] rating
- Engineering
  - Sensing, actuation, compute, comms, firmware, software, etc.
- Security & Privacy
  - Typically, user data is being communicated…what data? how is it being secured? who has access?
  - There may be regulatory requirements here as well: HIPAA
- Packaging
  - How is sent to the customer, could be simple/elaborate
- Installation and servicing
  - How does one go from "in the box" to "in use"?
  - Will it be serviced in the field? Will the SW be updated? Can the HW be fixed? Warranty?

These are not disjoint:
Needing to be updated after install: is that installation or engineering?

Needing to work outdoors will impact the materials used in the industrial design

Don't worry. Just make sure it's somewhere in document

Ultimately, the specification document should encompass all requirements

MAQS: from requirements to specifications

1. It should accurately measure indoor air quality **

- What is air quality?
  - Based on before, this is PM2.5,PM10, NO2, SO2, CO, O3 [1h & 8h]
  - Will generally also need to measure temperature and relative humidity, since many other variables depend on those
- What is accurate?
  - Need to establish for each sensor – TBD for now!
- What does indoor imply?
  - Has implications for environmental resistance
  - And availability of power [there are wall plugs, USB ports] indoors

MAQS: from requirements to specifications

2. It should be **portable** ***

- What does this mean?
- Can work without being plugged in ➔ we really mean it should be small and light…watch vs. toaster
- For how long? 1 min? 1 h? 1 day? Other?
  - User is student: how long is student away from charger? 12h seems about right

These are both portable

MAQS: from requirements to specifications

3. It should be possible to **get the data off** the device **

- It needs to be “connected” somehow
- Probably not just looking at a display and writing data down…
- Wired or wireless? Left unspecified ➔ Let’s go with wireless ➔ this then has regulatory implications because of the radio
- Lots of wireless comms approaches – specific one is TBD for now

MAQS: from requirements to specifications

4. It should be a **useful pedagogical exercise** ***

- Should be possible for each student to learn some skills: PCB design/fabrication/assembly, 3DP, server set-up
- Want to connect to other classes
- Learn how to work as a team

Not sure how this will translate to specs…let's revisit

MAQS: from requirements to specifications

5. It should maintain **privacy** *

- No personally identifiable information should leave the device
  ➔ implications for processing on server
- Information should be secure, only available to authorized users

MAQS: from requirements to specifications

6. It should be **low cost** *

- What does that mean? $100 BOM [for electronics]

7. It should be **rugged and robust** **

- Able to be transported by holding and placing in backpack ➔ it will require an enclosure
- Able to survive 12” drop onto table

MAQS: from requirements to specifications

8. Multiple systems should be able to be used **simultaneously**

- This has implications for server set up ➔ one server for class with multiple accounts? One server/student? One server with one db w/ different permissions?

9. It should be **easy to view** the **current** and **past** data

- Needs some storage, some sort of display ➔ exactly what, TBD
- The choice will implications for firmware, software, server

10. It should **leverage MIT facilities**

- Must only require tools & equipment we can readily access

MAQS: from requirements to specifications

OK, let’s translate to specification document

What makes a good specification? No single approach for all of HW & SW

- It might be a well-defined metric and value (or range of values)
  - Example: BOM < = $100
  - Example: Measurement interval <= 10 min
- It could be qualitative
  - Example: HTTPS GET/POST for server comms
- It could directly imply a particular implementation
  - Example: Connectivity: WiFi 802.11a/b/g/n [2.4 GHz]
- Or you might not know what it should be yet
  - Example: Sensor accuracy: ???
- Or, you might not even know about that specification
  - Example: ???

A good spec is verifiable…else how do you know if you meet the specs?

Don’t get hung up if you don’t know all of the specs at the beginning

The two most important points:

1. **Have a plan:** Work hard to plan ahead…and adjust the plan as needed

2. **Write stuff down:** Your team should have a single specifications document – a common understanding

MAQS specifications [1/2]

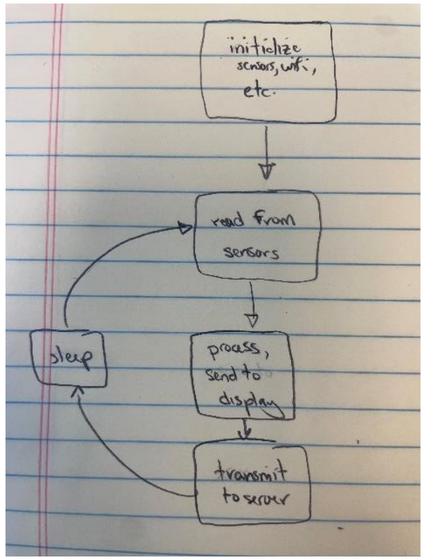
Ver. 1

- Financial
  - BOM <= $100 for electronics components, PCB
  - BOM: TBD for enclosure , mechanical parts
  - Time to market: ~8 weeks
- Regulatory
  - FCC certification for WiFi radio module
- Industrial design
  - Weight: < 300 g [~2 iphones]
  - Size: <10 x 10 x 10 cm [kinda small]
  - Survive 12” drop onto table
  - Enclosure materials: 3DP plastics available in EDS, laser-cut plastics available in EDS
- Environmental
  - Operating temperature: 0 to 70°C [commercial temp range]
  - Humidity: 10 to 95% RH
- Engineering
  - Sensors
    - Air quality: TBD
    - T: 0 to 70 °C
    - RH: 10 to 95% RH
    - Measurement interval: <=10 min
  - Compute
    - MCU: TBD
    - Firmware in C/C++
  - Comms
    - At least WiFi 802.11a/b/g 2.4 GHz
    - 5 GHz would be nice [801.22n]
    - WPA2-Enterprise w/ PEAP (MSCHAPv2) authentication and TLS encryption [this is what MIT Secure wants]
  - Energy management
    - LiPo battery
    - Lifetime between charging: >12 h
  - Server
    - Machine RPi 3 or 4, one for each student
    - SSH access for students, and staff
    - OS: Linux
    - Web server: NGINX
    - HTTPS GET/POST connections
    - DB: SQLite

MAQS specifications [2/2]

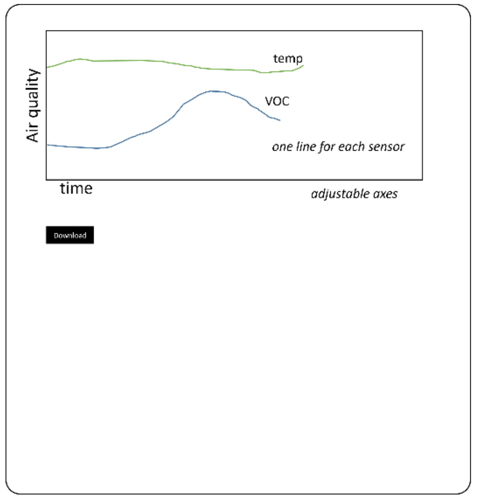
Ver. 1

- Firmware



- Still to specify
  - How to reset?
  - Data processing and what is transmitted
  - Sleep state, interval

- Software [on server]
  - Store data perpetually in SQLite table
    - Fields: Index number, Timestamp, RH, T, AQ measurements
    - No location information transmitted (or stored)
  - Web front-end
    - Framework: TBD



Web wireframe

Here we see that SW requirements often are specified differently [block diagram, wireframe, state machine, text] than HW

Does this cover all the requirements?

Test and verification

- Once you make it, does it work? **Does it meet spec?**
    - How will you debug if/when it doesn't work?
- For each spec, you want a way of testing it

- If you pass your tests
    - Then your design meets spec
        - And if you meet all your specs
            - Then you fulfill your requirements
                - And then success!

Test and verification

- Some tests are easy to write
- *Weight: < 300 g [~2 iphones]*
- *Test: weigh complete system*

Test and verification

- Some are more difficult
- *Energy Management*
  - *Lifetime between charging: >12 h*

- *Calculate energy budget and thus lifetime using part datasheets*

Test and verification

- Some are more difficult
- *Energy Management*
  - *Lifetime between charging: >12 h*

- *Measure energy consumption of components and use that to calculate lifetime*

Test and verification

- Some are more difficult
- *Energy Management*
  - *Lifetime between charging: >12 h*

- *Fully charge battery, run system using simplified FW, measure duration*

Test and verification

- Some are more difficult
- *Energy Management*
    - *Lifetime between charging: >12 h*

- *Fully charge battery, run system using simplified FW, measure consumption using energy meter, measure duration inside and outdoors*

What do we need for this test?

Test and verification

- Some are more difficult
- *Energy Management*
  - *Lifetime between charging: >12 h*

- *Fully charge battery, run system using simplified FW, measure consumption using energy meter, measure duration inside and outdoors*

What do we need for this test?

Some options

- *ESP32 + PM board + battery on breadboard*
- *ESP32 + PM board + battery on breadboard in housing*
- *ESP32 + [full sensor pack] + battery on breadboard in housing*
- *Etc.*

- Think about what subsystem/ testbed/prototype you need
- Who will develop it?
- How long will that take?
- What does its development depend on?

Test and verification

- Some are more difficult
- *Energy Management*
    - *Lifetime between charging: >12 h*

- *Fully charge battery, run system using simplified FW, measure consumption using energy meter, measure duration inside and outdoors*

What do we need for this test?

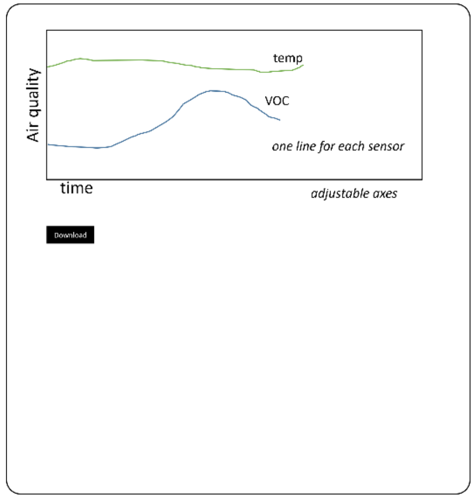
There is no correct answer here

The more sophisticated the test, the more complicated it will be to execute

Especially for March testing, you will not have the complete system ready so must make compromises

Test and verification

- Another simple one
- **Software [on server]**
  - Web front-end
    - Framework: TBD



- *Compare wireframe to implemented front-end, visually determine pass*

Test and verification

- More difficult
- **Software [on server]**
    - Store data perpetually in SQLite table
        - Fields: Index number, Timestamp, RH, T, AQ measurements
        - No location information transmitted (or stored)

Perpetuity is a long time…

Thinking about testing helps us understand the flaws in the spec

Test and verification

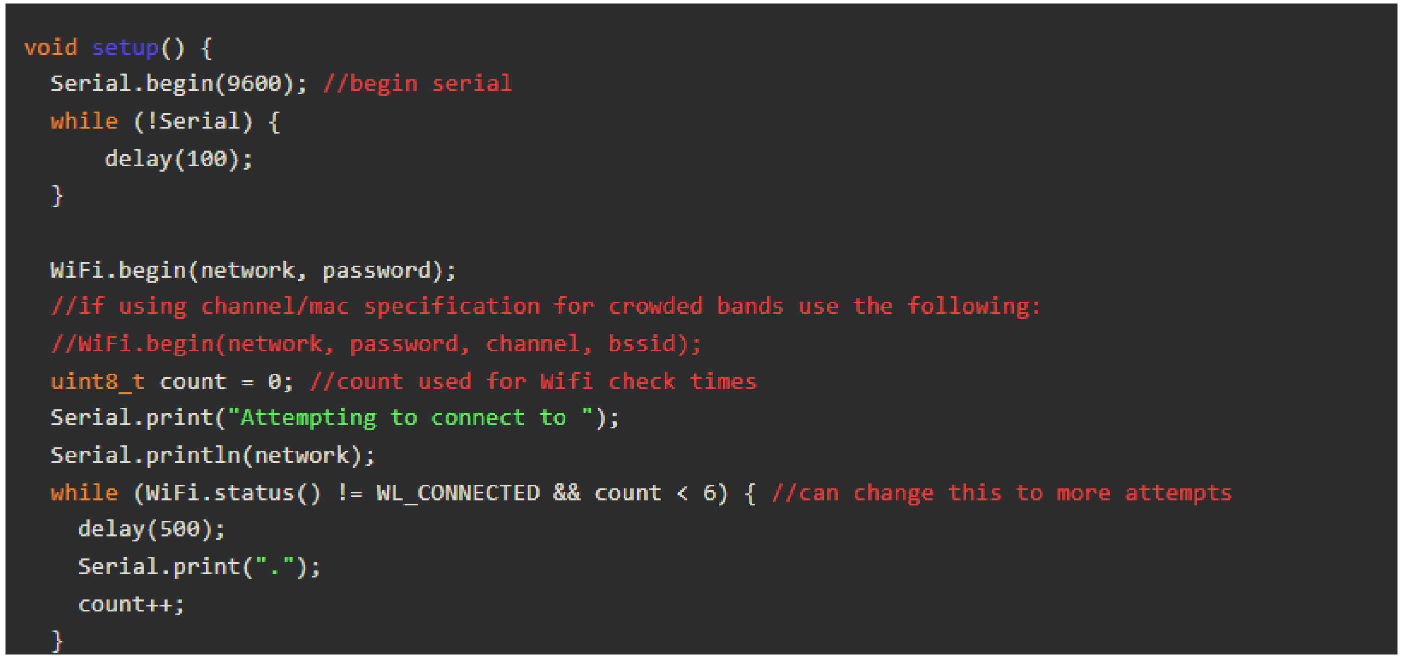
- More difficult
- **Software [on server]**
  - Store data in SQLite table, at least 1 y of data stored
    - Fields: Index number, Timestamp, RH, T, AQ measurements
    - No location information transmitted (or stored)

- Send 100 measurements, measure storage needed, extrapolate to 80% size of RPi SD card size

Is there a max SQLite db file size?
Does RPi need certain amount of free disk to operate robustly (w/o crashing)?

Test and verification: FW & SW

- Tests of each function (unit tests) and overall FW
- Ideally, not just when everything works, but consider common failures
  - WiFi down…
  - Reset
  - Sensors fail
  - And so on…



```
void setup() {
  Serial.begin(9600); //begin serial
  while (!Serial) {
      delay(100);
  }

  WiFi.begin(network, password);
  //if using channel/mac specification for crowded bands use the following:
  //WiFi.begin(network, password, channel, bssid);
  uint8_t count = 0; //count used for Wifi check times
  Serial.print("Attempting to connect to ");
  Serial.println(network);
  while (WiFi.status() != WL_CONNECTED && count < 6) { //can change this to more attempts
    delay(500);
    Serial.print(".");
    count++;
  }
```

What happens if WiFi goes down?

Test and verification & tradeoffs

- More difficult
- **Software [on server]**
  - Store data in SQLite table, at least 1 y of data stored
    - Fields: Index number, Timestamp, RH, T, AQ measurements
    - No location information transmitted (or stored)

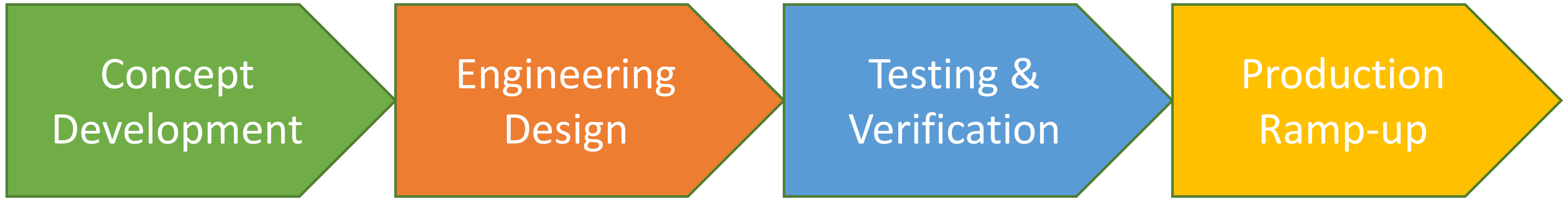
Server

Machine RPi 3 or 4, one for each student
Storage: 16 Gb SD card
SSH access for students, and staff
OS: Linux
Web server: NGINX
HTTPS GET/POST connections
DB: SQLite

← Needed to add a spec!

*Here we see where the **team** must work together to balance tradeoffs between disk size, cost, and data frequency*

Iterating

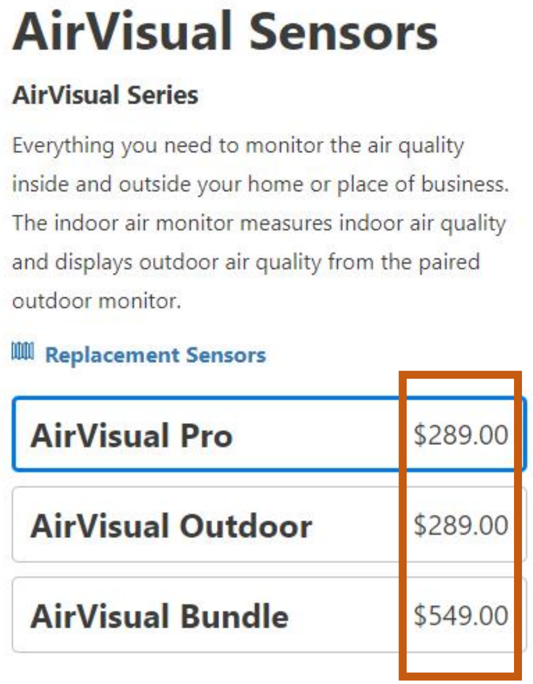


We iterate between:

- Developing and refining concepts: form and function
  - This will involve system design and partitioning
- Research: what's out there and available, what do our competitors do?
- Update specifications document as needed (incl. tests!) ← remember this is a *working document*
- Even before our first testing phase, you can research, model, prototype & test
  - Identify high-risk questions that threaten overall system
  - De-risk
- Once you have a system design & partition that is suitably stable – start detailed design & development

MAQS: market research

- This is an active space
  - For-profit, non-profit, DIY



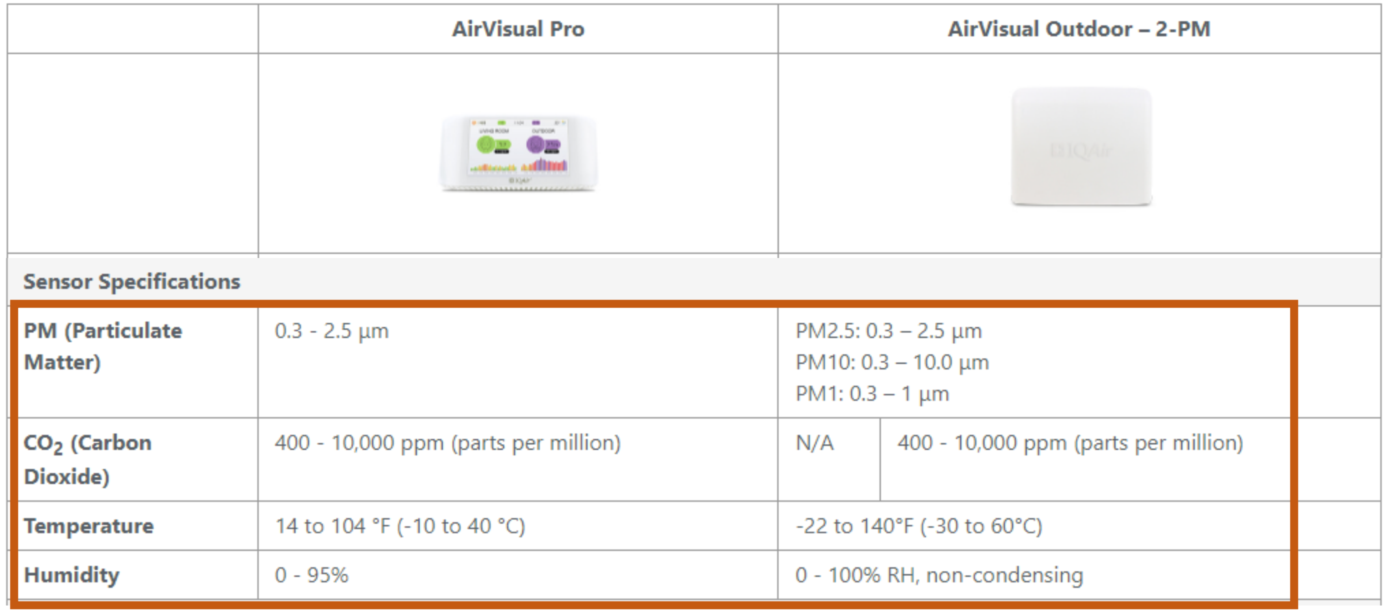
AirVisual Sensors

AirVisual Series

Everything you need to monitor the air quality inside and outside your home or place of business. The indoor air monitor measures indoor air quality and displays outdoor air quality from the paired outdoor monitor.

Replacement Sensors

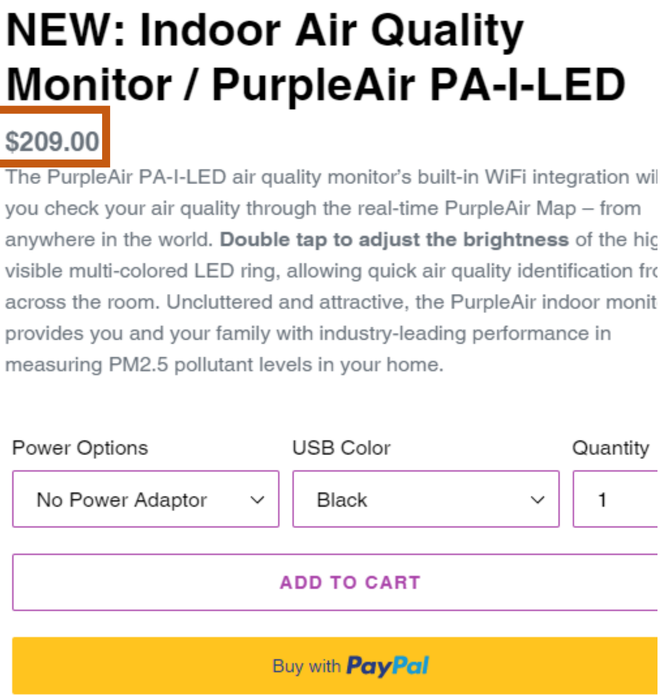
AirVisual Pro	$289.00
AirVisual Outdoor	$289.00
AirVisual Bundle	$549.00



	AirVisual Pro	AirVisual Outdoor – 2-PM	
Sensor Specifications			
PM (Particulate Matter)	0.3 - 2.5 µm	PM2.5: 0.3 – 2.5 µm PM10: 0.3 – 10.0 µm PM1: 0.3 – 1 µm	
CO_2 (Carbon Dioxide)	400 - 10,000 ppm (parts per million)	N/A	400 - 10,000 ppm (parts per million)
Temperature	14 to 104 °F (-10 to 40 °C)	-22 to 140°F (-30 to 60°C)	
Humidity	0 - 95%	0 - 100% RH, non-condensing	

www.iqair.com/us/air-quality-monitors

MAQS: market research



NEW: Indoor Air Quality Monitor / PurpleAir PA-I-LED

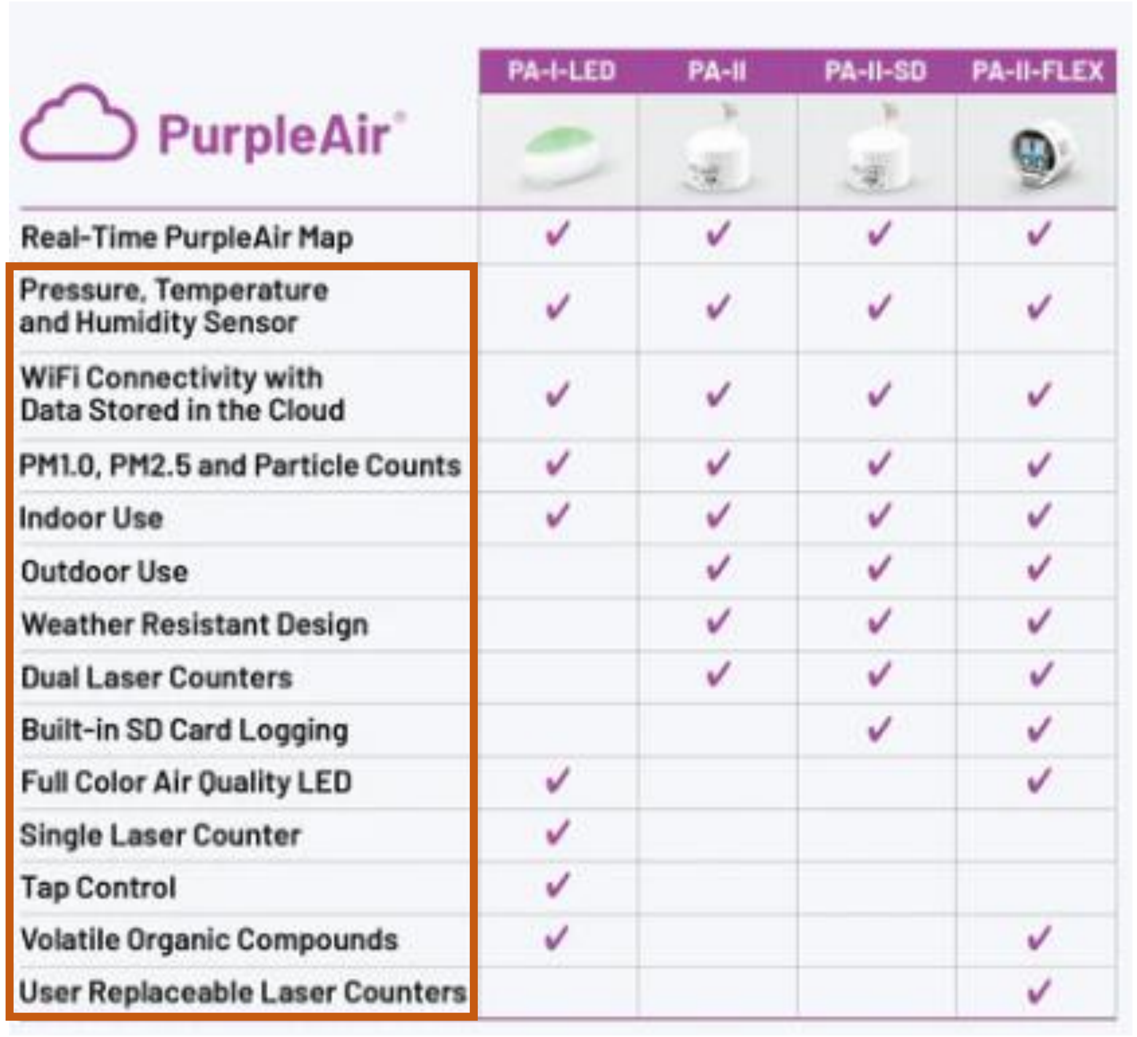
$209.00

The PurpleAir PA-I-LED air quality monitor's built-in WiFi integration wil you check your air quality through the real-time PurpleAir Map – from anywhere in the world. **Double tap to adjust the brightness** of the hig visible multi-colored LED ring, allowing quick air quality identification fro across the room. Uncluttered and attractive, the PurpleAir indoor monit provides you and your family with industry-leading performance in measuring PM2.5 pollutant levels in your home.

Power Options: No Power Adaptor
USB Color: Black
Quantity: 1

ADD TO CART

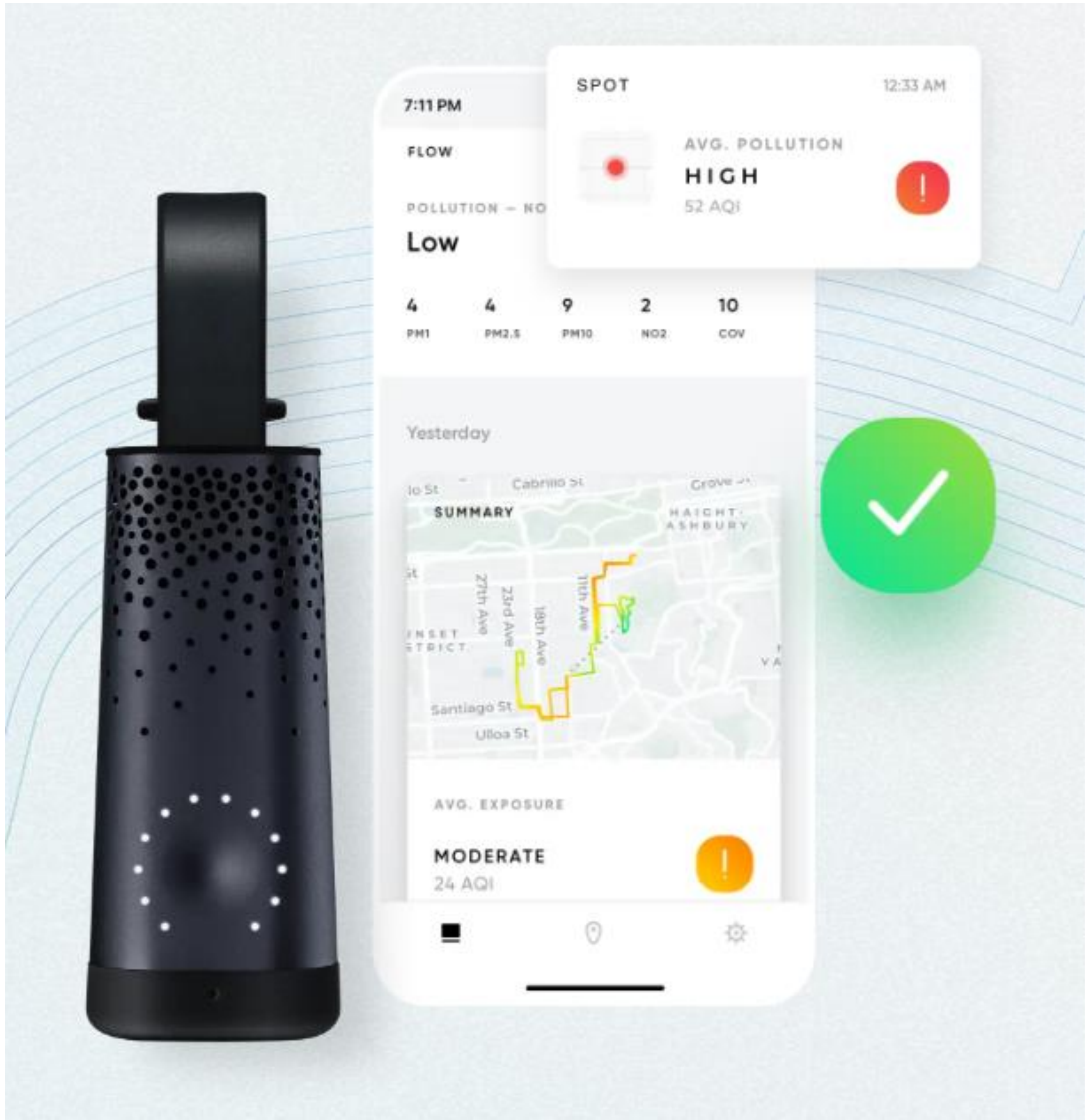
Buy with PayPal

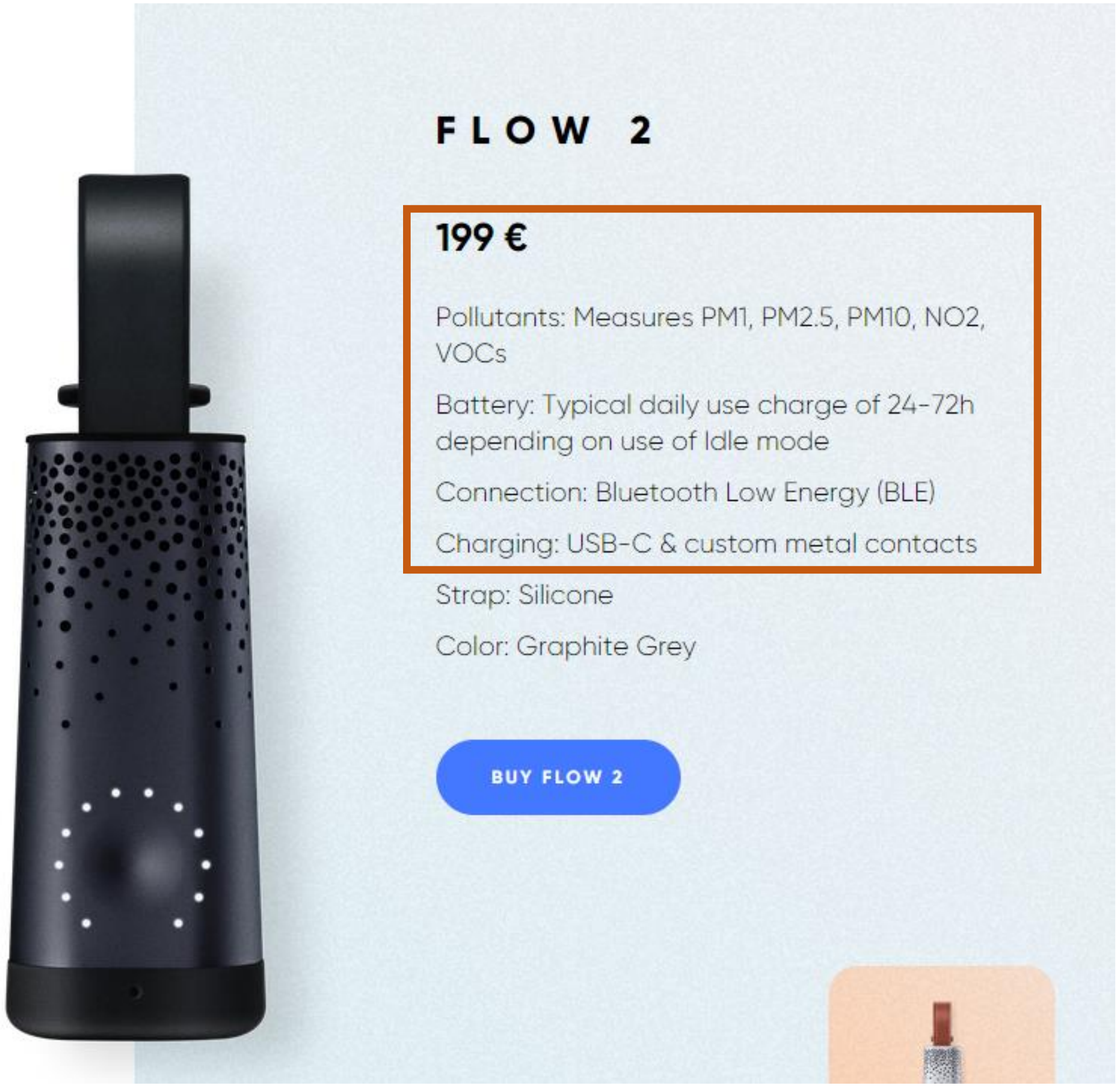


PurpleAir	PA-I-LED	PA-II	PA-II-SD	PA-II-FLEX
Real-Time PurpleAir Map	✓	✓	✓	✓
Pressure, Temperature and Humidity Sensor	✓	✓	✓	✓
WiFi Connectivity with Data Stored in the Cloud	✓	✓	✓	✓
PM1.0, PM2.5 and Particle Counts	✓	✓	✓	✓
Indoor Use	✓	✓	✓	✓
Outdoor Use		✓	✓	✓
Weather Resistant Design		✓	✓	✓
Dual Laser Counters		✓	✓	✓
Built-in SD Card Logging			✓	✓
Full Color Air Quality LED	✓			✓
Single Laser Counter	✓			
Tap Control	✓			
Volatile Organic Compounds	✓			✓
User Replaceable Laser Counters				✓

www2.purpleair.com

MAQS: market research





FLOW 2

199 €

Pollutants: Measures PM1, PM2.5, PM10, NO2, VOCs

Battery: Typical daily use charge of 24-72h depending on use of Idle mode

Connection: Bluetooth Low Energy (BLE)

Charging: USB-C & custom metal contacts

Strap: Silicone

Color: Graphite Grey

plumelabs.com/en/flow/

MAQS: market research

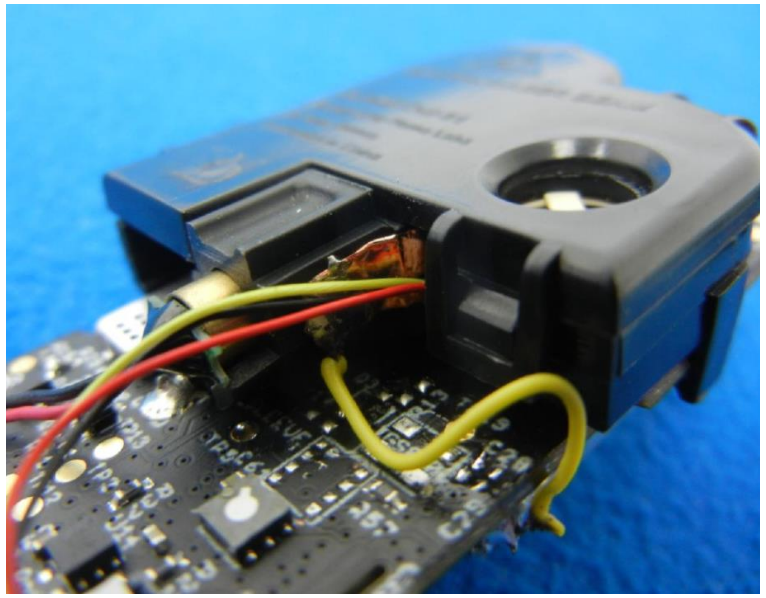
- Many products out there measure PM10, PM2.5, VOC, NOx (and T, RH)
  - But CO2? Pressure?
- AQI: Comprised of **$PM_{2.5}$, PM_{10}, NO_2,** SO_2, CO, O_3 [1h & 8h]

What about these?

Purpleair teardown

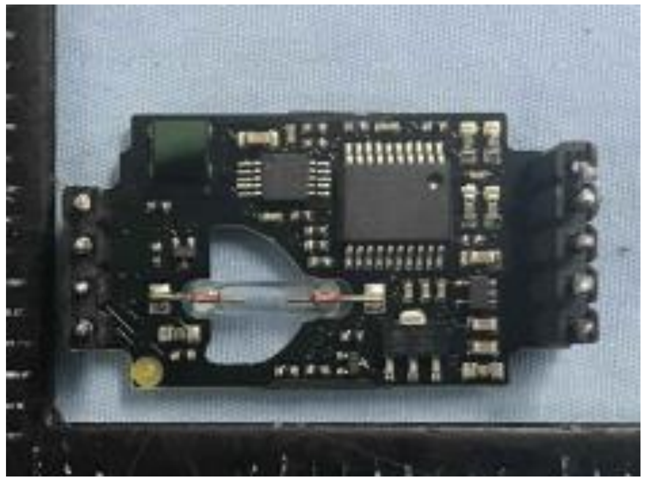
drive.google.com/file/d/11Y-x0m3KHEIeb5fRSV8UtnKr9MZzPcdv/view

Flume FCC cert teardown



fccid.io/2APMO-FLOW/Internal-Photos/Internal-Photos-3952125

IQAir FCC cert teardown



fccid.io/2AMBQ-N1/Internal-Photos/Internal-Photos-01-3640386

MAQS: market research

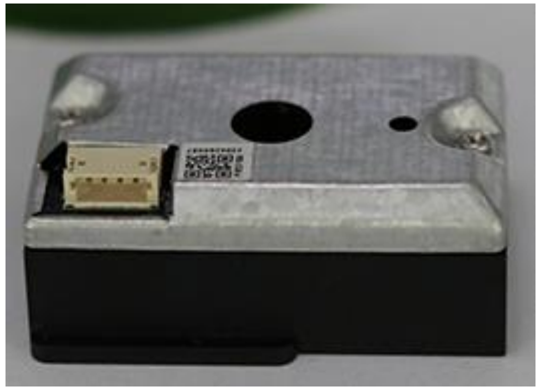
- What sensors are commercially available?
- COTS: commercial off-the-shelf

Particulate: PM2.5, PM10

Plantower PMS series [1003, 3003, etc.]
$10-20/ea

Honeywell HPM series
$70-80/ea

Sensirion SPS30
$30-50/ea



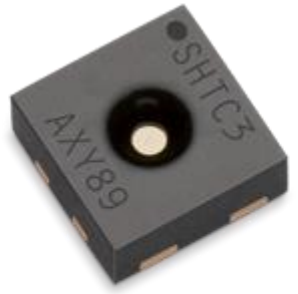
Plantower PIRS10A
Price: $4

MAQS: market research

- T/RH are readily available
  - Lots of specs, sizes, costs, etc.
- VOC/NOx also readily available
  - But difficult to relate to absolute levels
- Others are less common and/or expensive
  - O_3: \$20-50/ea
  - CO: most are \$20-50+/ea
  - SO2: \$20+/ea

Temperature, humidity
Typically bundled together…why?

TE connectivity TSYS02S
\$3-4/ea @10
~7 parts in their product line



Sensirion SHTC3
\$2-3/ea @ 10
~25 parts in their product line

Formaldehyde/VOC/NOx

Plantower DS-HCHO-20
~\$25/ea

Sensirion SGP41
\$5-8/ea @ 10

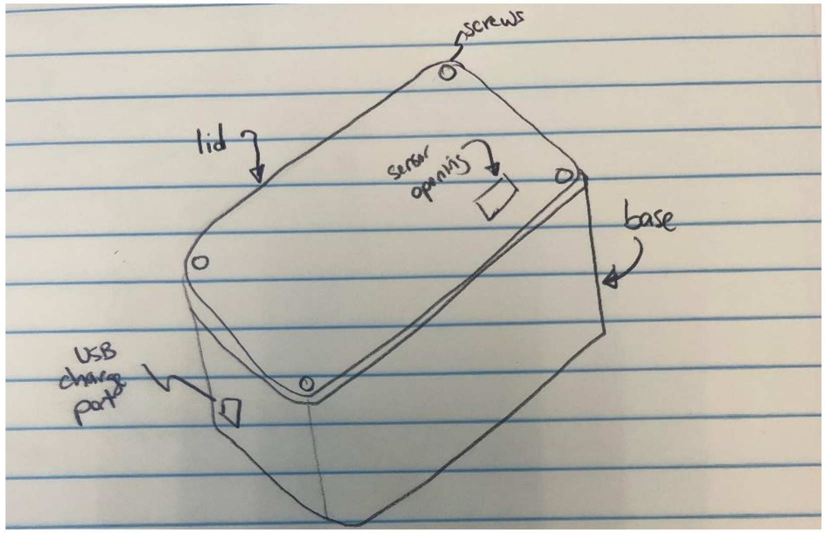
Bosch BME680
~\$10/ea @10

Outcome: let's focus on RH/T (useful, cheap) & PM (useful, moderate cost)

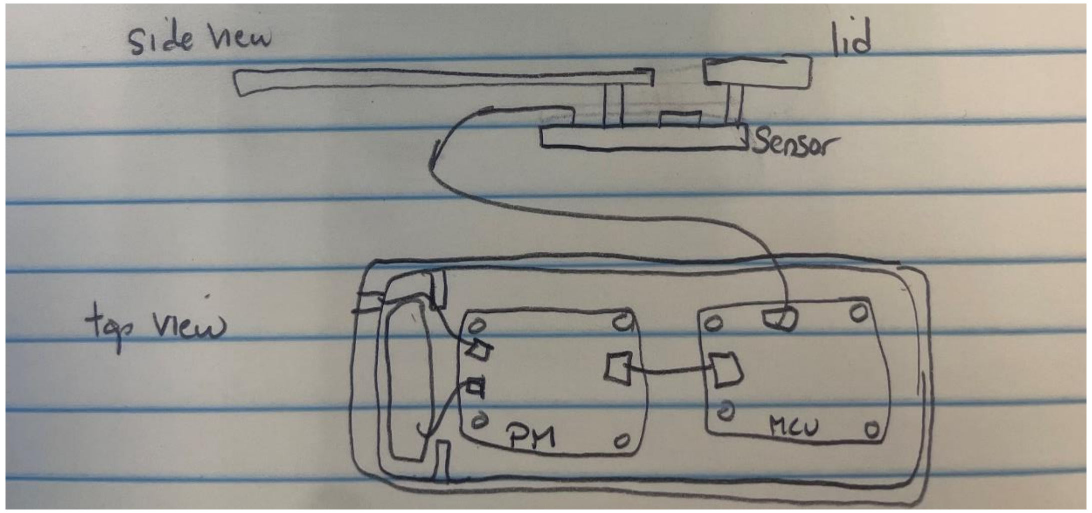
MAQS: concepts & systems

- Next, let’s sketch some concepts & systems
- We need to consider
  - Industrial design: what it “looks-like”, how it interacts with user
  - Engineering: how it functions
- We can “sketch”
  - On paper with pen or pencil
  - On computer in ppt, figma, solidworks, fusion360, etc.

MAQS: industrial design

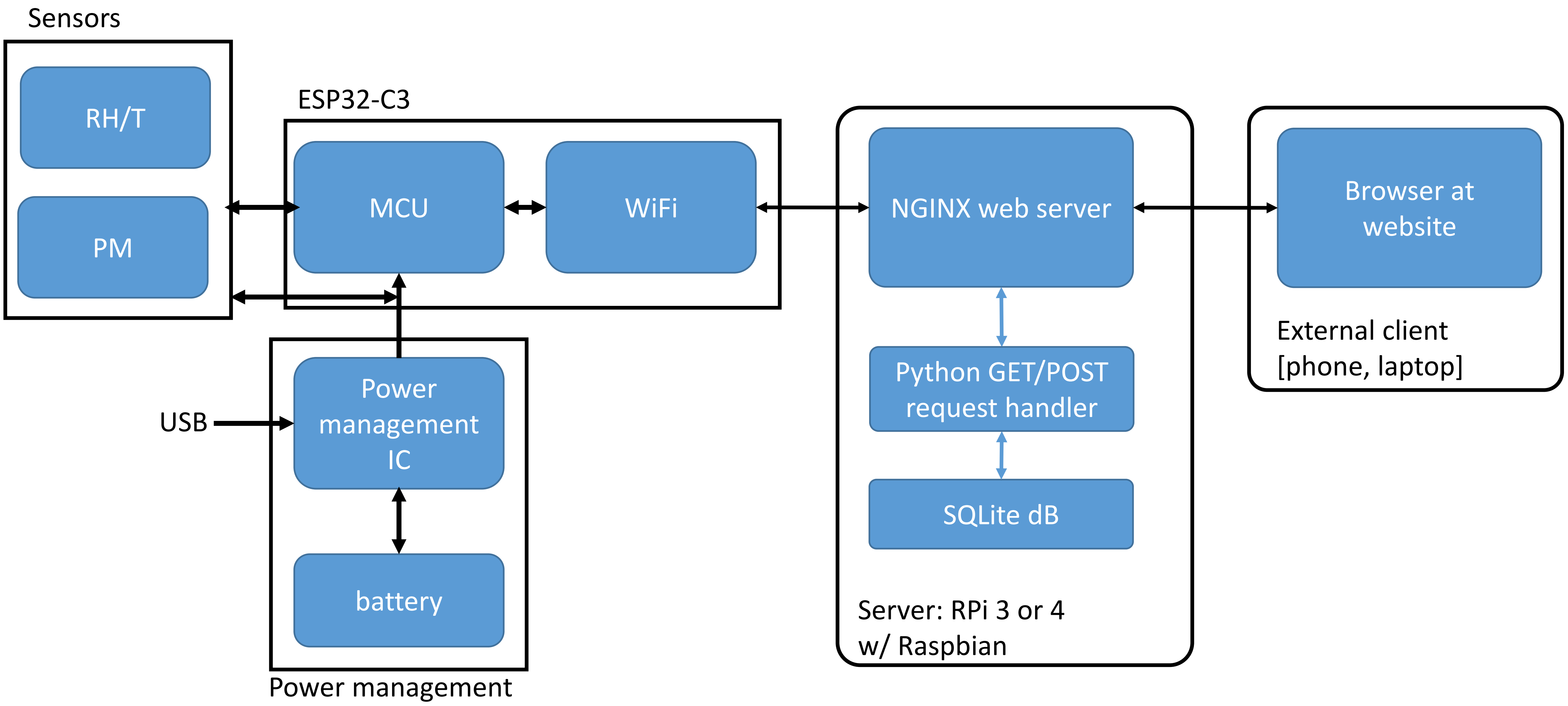


External view



Internal view

MAQS: system diagram

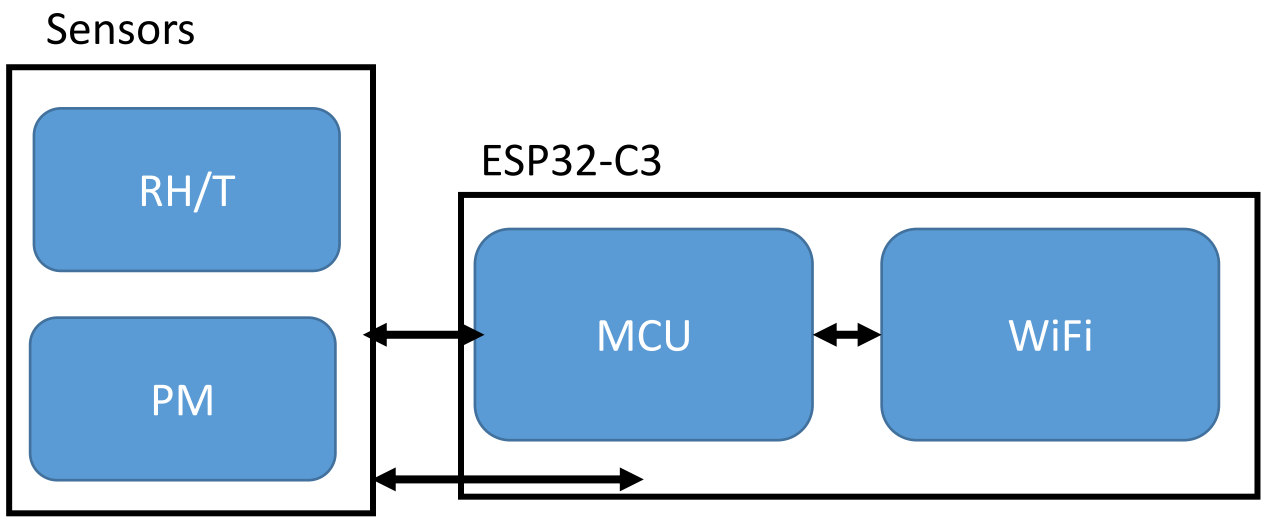


MAQS: system design & partitioning

- Our system block diagram starts to imply a system partition
  - Functional partitioning: allocating functions to different parts of the system
  - Physical partitioning: What parts go where, how do they physically connect to each other
- Partitioning can be applied recursively
  - Big blocks into smaller subblocks
- How far to go?
  - As far as needed to make it clear what to design, and so a person/team can start to design
- We partition to manage complexity
  - Subsystems can be designed independently as long as interface is well-defined
  - Allows abstracting away details of other subsystems

MAQS: system design & partitioning

- A good partitioning will have *parts* that
  - Make internal sense –are coherent in terms of the functionality
    - Sensor suite partitioned from ESP32
  - Minimize coupling between parts
    - Minimize interfaces
    - Interfaces often translate to connectors, wires, cables, tubes, APIs, function calls, methods, etc.
    - Strong coupling can suggest that parts belong together rather than separate
  - In a company, partitions may be organized by team for each subsystem
    - Sensors/electronics, power, firmware, mechanical, industrial, SWE, backend, frontend, etc.
- There is no optimal partition…



MAQS: system partitioning & tradeoffs

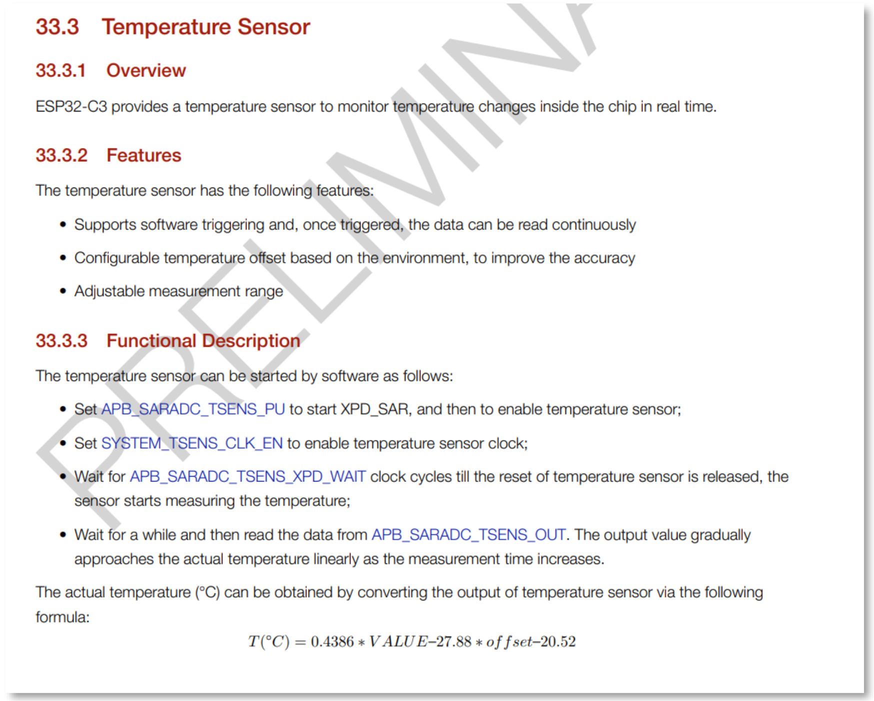
- How do we evaluate/compare designs?
- Trade-off analysis
    - Translate a design back into specs: Performance, cost, size, power, etc.
    - Tradeoff implies that there is no single optimum – it’s up to you as the designer (or your team) to choose!
- Identify & focus on high-risk and addressable unknowns
- De-risk
    - Research
    - Model
    - Prototype
- Once you have confidence in a subsystem partition…do detailed design

	Sensirion SPS30	Plantower PMS7003
Price	\$30-50/ea	\$10-20/ea
Size	41 x 41 x 12 mm³	48×37×12 mm3
Power	5V@80 mA	5V@<=100mA
Measures	PM1, PM2.5, PM4, PM10	PM1, PM2.5

And so on. You might care about lifetime, accuracy/precision, low-power modes, connection interface, etc.

MAQS: sensor subsystem design

- Focus on sensing
  - Several approaches to temperature sensing (more on this later in term)
  - ESP32 (and many MCU) have on-board temp sensor



33.3 Temperature Sensor

33.3.1 Overview

ESP32-C3 provides a temperature sensor to monitor temperature changes inside the chip in real time.

33.3.2 Features

The temperature sensor has the following features:

- Supports software triggering and, once triggered, the data can be read continuously
- Configurable temperature offset based on the environment, to improve the accuracy
- Adjustable measurement range

33.3.3 Functional Description

The temperature sensor can be started by software as follows:

- Set APB_SARADC_TSENS_PU to start XPD_SAR, and then to enable temperature sensor;
- Set SYSTEM_TSENS_CLK_EN to enable temperature sensor clock;
- Wait for APB_SARADC_TSENS_XPD_WAIT clock cycles till the reset of temperature sensor is released, the sensor starts measuring the temperature;
- Wait for a while and then read the data from APB_SARADC_TSENS_OUT. The output value gradually approaches the actual temperature linearly as the measurement time increases.

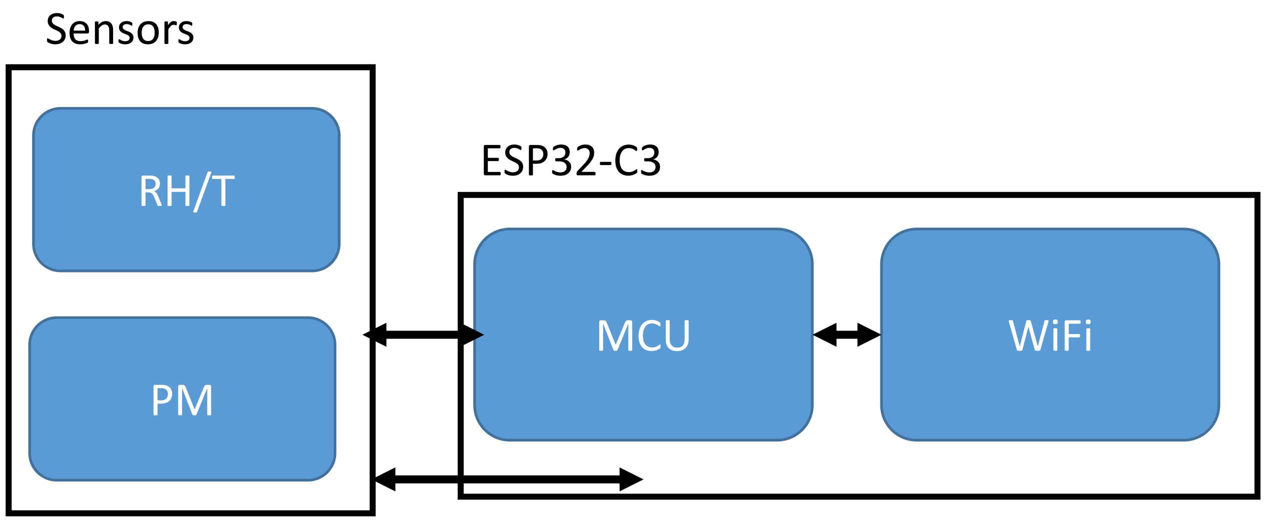
The actual temperature (°C) can be obtained by converting the output of temperature sensor via the following formula:

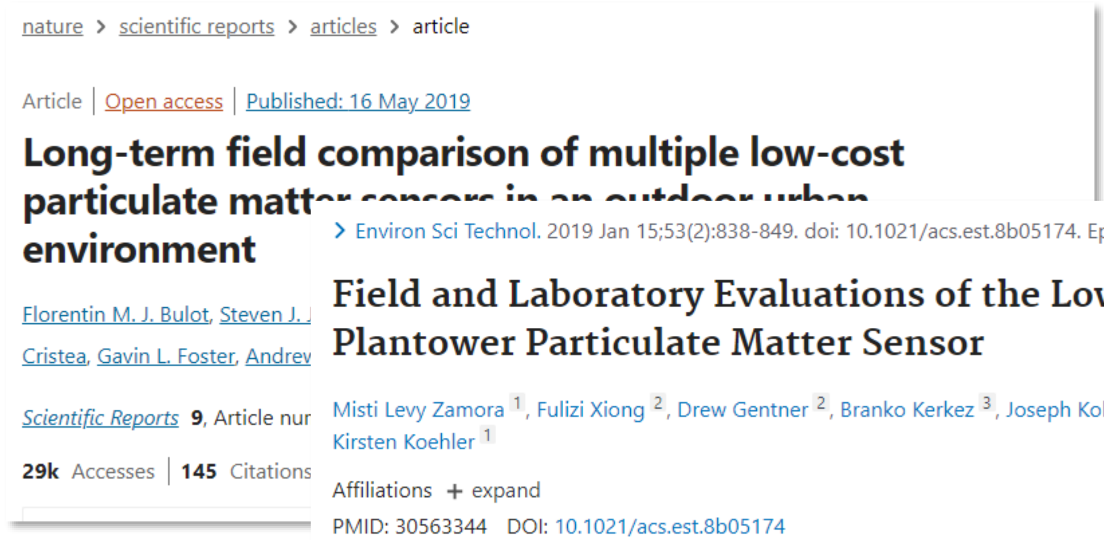
$$T(°C) = 0.4386 * VALUE - 27.88 * offset - 20.52$$

  - But we don't necessarily want *that* temp
  - And we also want RH
  - So choose separate RH/T sensor

MAQS: sensor subsystem design

- Sensors
  - Connect to MCU, but partitioned separately
  - Because ~0 MCUs have integrated sensors
    - T is exception…more in a few weeks
    - But anyway we want RH also
  - Some sensors do have integrated MCUs
    - Such as for incorporating processing, AI, etc.
    - Reduce part count on board
    - But typically not full-feature MCU
- RH/T
  - Together, or partition?
  - Almost all RH sensors also include T, so no benefit to separate T
  - ~all RH/T sensors have digital outputs
- PM
  - Relatively costly but important and reasonably accurate





nature > scientific reports > articles > article

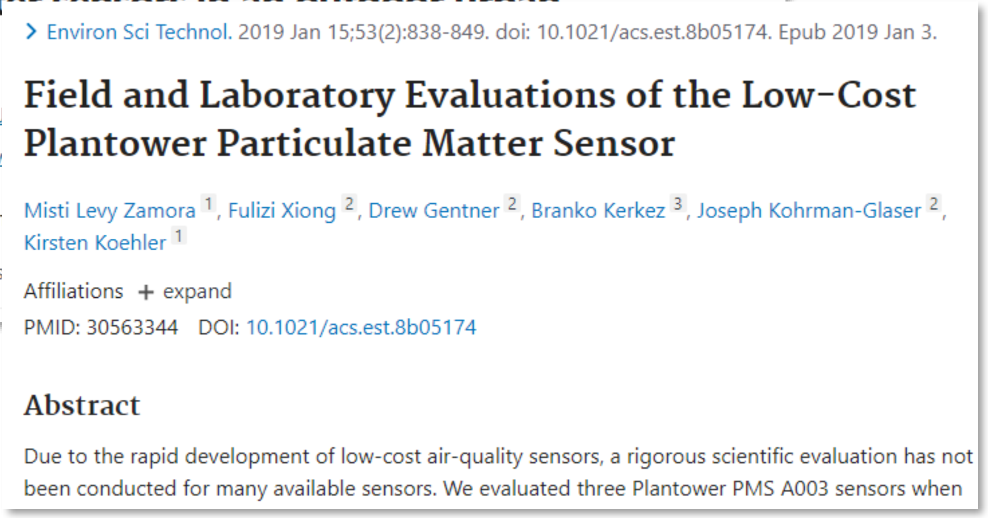
Article | Open access | Published: 16 May 2019

Long-term field comparison of multiple low-cost particulate matt… sensors in an outdoor urban environment

Florentin M. J. Bulot, Steven J. … Cristea, Gavin L. Foster, Andrew…

Scientific Reports **9**, Article nu…

29k Accesses | **145** Citations



> Environ Sci Technol. 2019 Jan 15;53(2):838-849. doi: 10.1021/acs.est.8b05174. Epub 2019 Jan 3.

Field and Laboratory Evaluations of the Low-Cost Plantower Particulate Matter Sensor

Misti Levy Zamora [1], Fulizi Xiong [2], Drew Gentner [2], Branko Kerkez [3], Joseph Kohrman-Glaser [2], Kirsten Koehler [1]

Affiliations + expand

PMID: 30563344 DOI: 10.1021/acs.est.8b05174

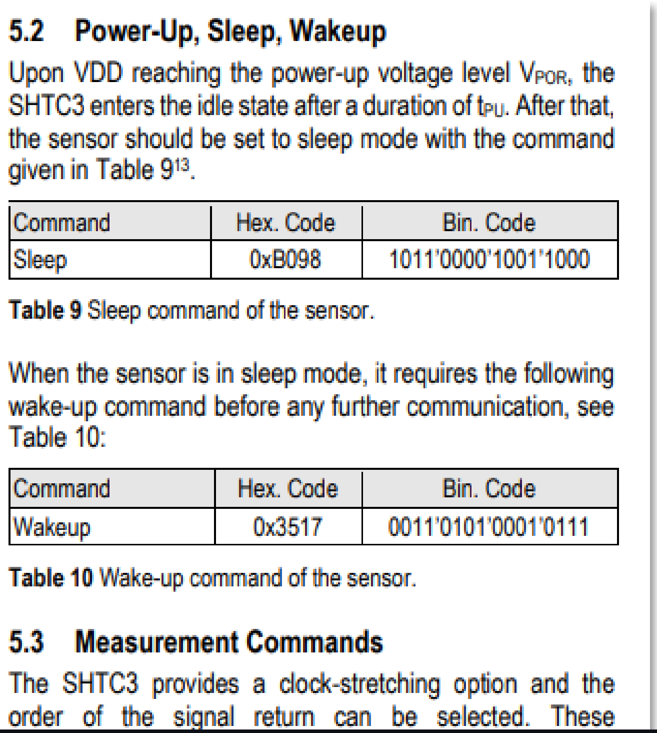
Abstract

Due to the rapid development of low-cost air-quality sensors, a rigorous scientific evaluation has not been conducted for many available sensors. We evaluated three Plantower PMS A003 sensors when

MAQS: sensor subsystem design

- What is the interface between sensors and MCU?
- Physical interface
  - Chip-level comms is often via I2C, SPI, UART
  - 2+ traces on PCB, 2+ pins on MCU
    - More MCU pins ➔ bigger MCU (sometimes), more expensive
- Functional interface
  - A digital communications protocol: I2C, SPI most common
    - MCU should have the needed communications peripheral (else you have to bit-bang your own)
  - An API/library
    - A set of commands from sensor manufacturer OR a library that encapsulates those commands
    - You can always write your own as well

The datasheet is your friend



5.2 Power-Up, Sleep, Wakeup

Upon VDD reaching the power-up voltage level V_{POR}, the SHTC3 enters the idle state after a duration of t_{PU}. After that, the sensor should be set to sleep mode with the command given in Table 9[13].

Command	Hex. Code	Bin. Code
Sleep	0xB098	1011'0000'1001'1000

Table 9 Sleep command of the sensor.

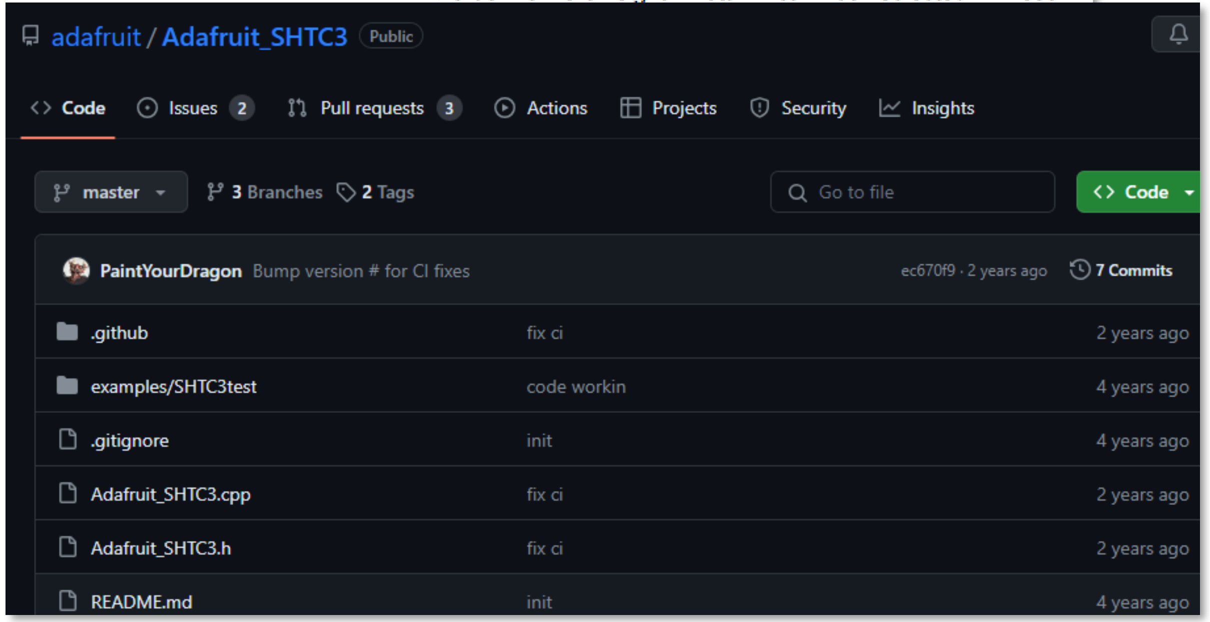
When the sensor is in sleep mode, it requires the following wake-up command before any further communication, see Table 10:

Command	Hex. Code	Bin. Code
Wakeup	0x3517	0011'0101'0001'0111

Table 10 Wake-up command of the sensor.

5.3 Measurement Commands

The SHTC3 provides a clock-stretching option and the order of the signal return can be selected. These



adafruit / Adafruit_SHTC3 Public

Code | Issues 2 | Pull requests 3 | Actions | Projects | Security | Insights

master | 3 Branches | 2 Tags | Go to file | Code

PaintYourDragon Bump version # for CI fixes ec670f9 · 2 years ago 7 Commits

.github	fix ci	2 years ago
examples/SHTC3test	code workin	4 years ago
.gitignore	init	4 years ago
Adafruit_SHTC3.cpp	fix ci	2 years ago
Adafruit_SHTC3.h	fix ci	2 years ago
README.md	init	4 years ago

MAQS: sensor subsystem design

- What is the interface between sensors and MCU?
- PMS7003 sensor
  - Comes in two different interfaces: I2C and UART
  - I2C
    - No extra pins needed on MCU b/c already using I2C for RH/T sensor
    - ~$30-40/ea
  - UART
    - Need extra UART on MCU – luckily ESP32 has 2 (actually 3) UARTs
      - One UART typically used for USB programming
    - Two extra pins/traces
    - ~$10/ea

Hi-risk question:

Will two UARTs work simultaneously on ESP32C3?

➔ purchase PMS7003 (UART) and try it out!

…works

MAQS: power subsystem design

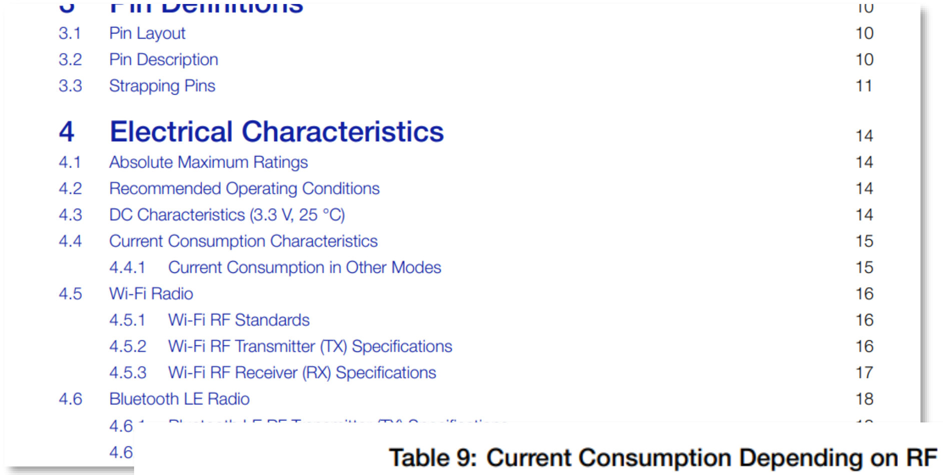
- Modeling can sometimes save time and money
- You know a lot from all your training…use it!
- Specifications
  - Energy management
    - LiPo battery
    - Lifetime between charging: >12 h

Medium-risk question:

Is this going to be easy or hard?

MAQS: Power management – modeling

- Power
  - We'll do a lot more in a few weeks…
- Can we estimate the lifetime?
- What's the biggest energy consumer – usually MCU or comms
  - In our case, WiFi
  - Let's check ESP32-C3 datasheet
- What about battery?
  - ~infinite number of choices
  - Most common rechargeable choice these days is LiPo
  - Let's look at 18650 b/c it is used in 6.08
    - 3.7V nominal for single cell
    - Typical capacities 2200 mAh for Adafruit one



3 Pin Definitions 10
3.1 Pin Layout 10
3.2 Pin Description 10
3.3 Strapping Pins 11

4 Electrical Characteristics 14
4.1 Absolute Maximum Ratings 14
4.2 Recommended Operating Conditions 14
4.3 DC Characteristics (3.3 V, 25 °C) 14
4.4 Current Consumption Characteristics 15
4.4.1 Current Consumption in Other Modes 15
4.5 Wi-Fi Radio 16
4.5.1 Wi-Fi RF Standards 16
4.5.2 Wi-Fi RF Transmitter (TX) Specifications 16
4.5.3 Wi-Fi RF Receiver (RX) Specifications 17
4.6 Bluetooth LE Radio 18
4.6.1
4.6

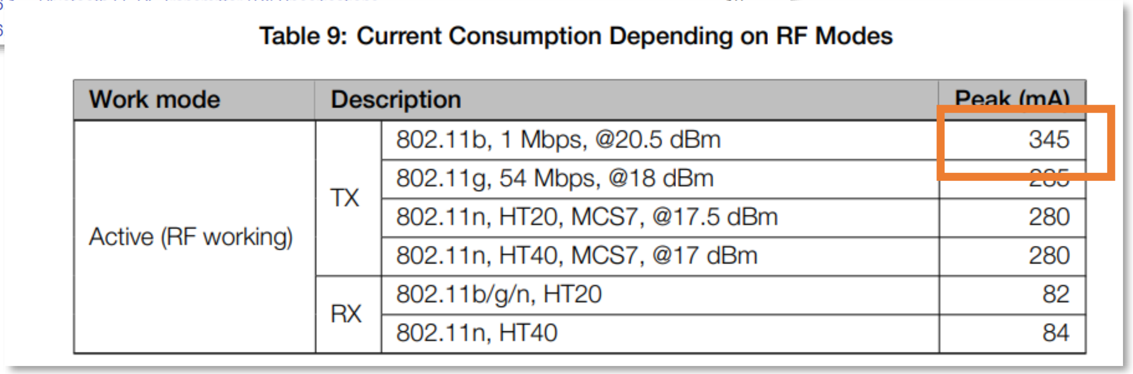


Table 9: Current Consumption Depending on RF Modes

Work mode	Description		Peak (mA)
Active (RF working)	TX	802.11b, 1 Mbps, @20.5 dBm	345
		802.11g, 54 Mbps, @18 dBm	285
		802.11n, HT20, MCS7, @17.5 dBm	280
		802.11n, HT40, MCS7, @17 dBm	280
	RX	802.11b/g/n, HT20	82
		802.11n, HT40	84

Assume 400 mA @ 3.3V consumption
Assume 2200 mAh capacity @ 3.3V [assume no energy savings for 3.7V to 3.3V conversion]

~5.5h if transmitting WiFi continuously…which we aren't going to be doing ➔ should be ok!