

BOMP Case Study: ECJ

Since we are limited by the distance between hubs and sensors, the idea is to now switch to a case study where we simply measure IAQ throughout ECJ. However, some digging shows that we can perhaps use RPi as beacons to ping the AirThings devices *without* Hubs to circumvent our problem.

Sensor Setup and Deployment

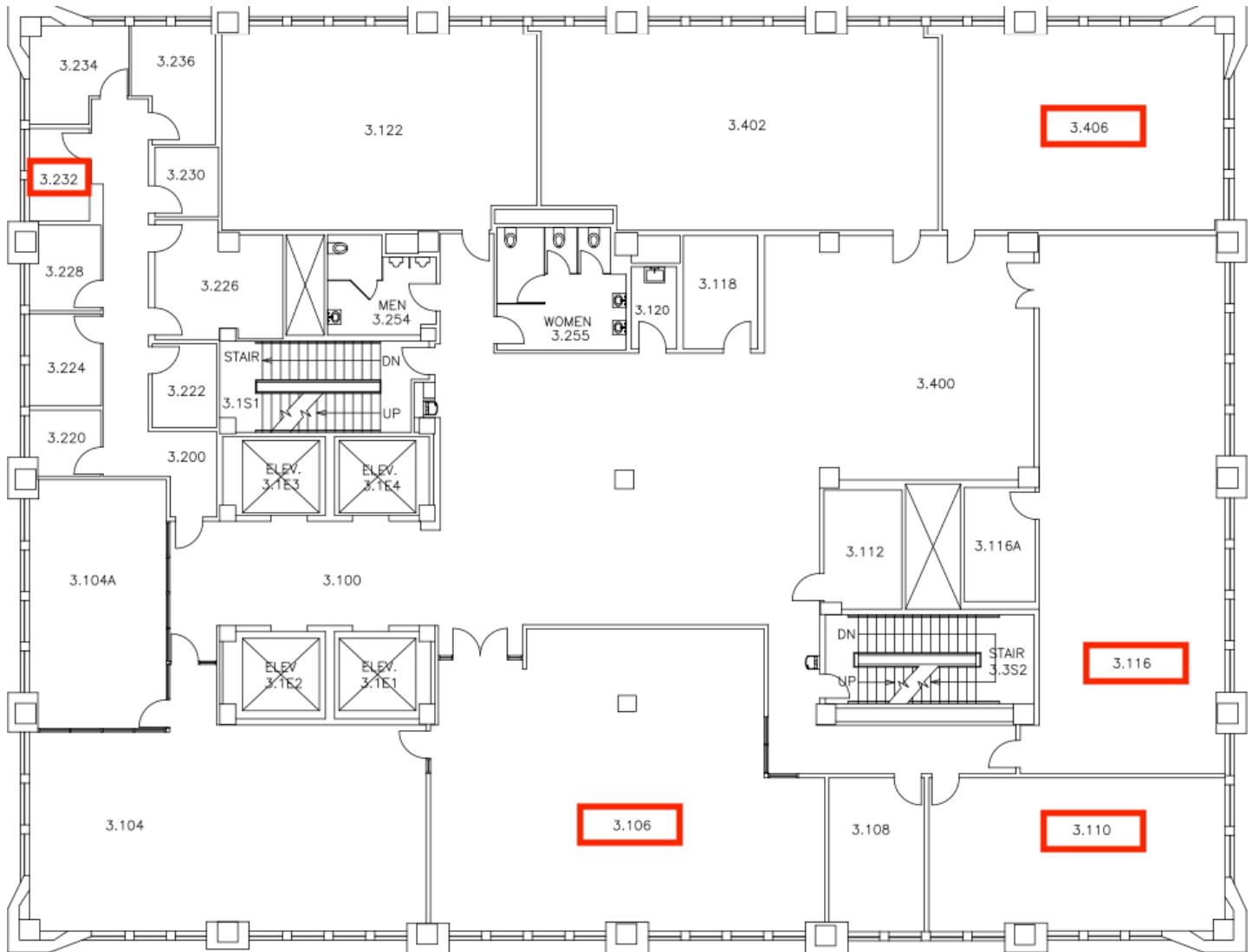
We can only realistically monitor two floors in ECJ and, even then, maybe only a portion of the floor plan. We still want to target as many different locations that we possibly can. I think the 3rd and 5th floors are our best options. We would need to:

- get the devices registered
- set up devices in the locations below
- monitor device connection and power level
- provide maintenance whenever necessary
- change sensor locations if we cannot guarantee location

Third Floor

We will target more of the classrooms on this floor. For `ID`, fill in with the appropriate name i.e. `airthings` or `purpleair`.

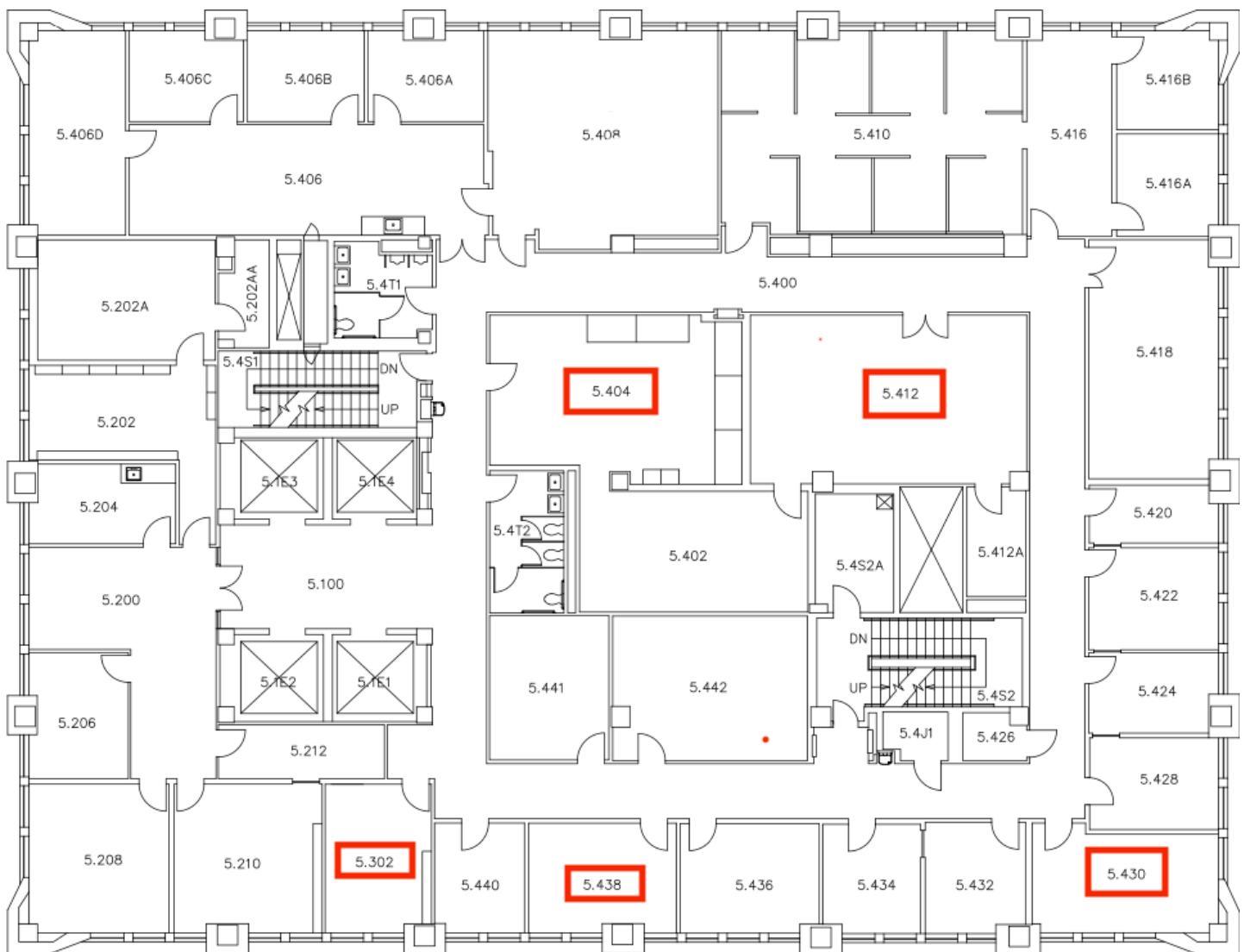
Room Number	Room Description	Contact	Confirmed	ID
3.106	Computer Room	Gregory Brooks (gregorybrooksmail@gmail.com)	✓	_ecj_3106
3.110	Small Classroom	Jeff Sarich (Jeff.Sarich@austin.utexas.edu)	✓	_ecj_3110
3.116	Large Classroom	Jeff Sarich (Jeff.Sarich@austin.utexas.edu)	✓	_ecj_3116
3.406	Medium Classroom	Jeff Sarich (Jeff.Sarich@austin.utexas.edu)	✓	_ecj_3406
3.232	Office Spaces	Danny Quiroz (quiroz@mail.utexas.edu)	✓	_ecj_3232



Fifth Floor

Office spaces only on this floor. For **ID**, fill in with the appropriate name i.e. **airthings** or **purpleair**.

Room Number	Room Description	Contact	Confirmed	ID
5.430	larger, single office	Dr. Atila Novoselac (atila@mail.utexas.edu)	✓	_ecj_5430
5.438	normal, single office	Dr. Kasey Faust (faustk@utexas.edu)	✓	_ecj_5438
5.302	normal, single office	Dr. Fernanda Leite (fernanda.leite@austin.utexas.edu)	✓	_ecj_5302
5.404	shared office (CEPM)	InBae Chung	✓	_ecj_5404
5.412	shared office (CEPM)	InBae Chung	✓	_ecj_5412



Sensor Interface and API

As our sensors get put into place, we need to make sure that we can pull data from them easily.

AirThings

API

The API is a bit complicated and uses a language I am not familiar with. Since none of us have experience (as far as I know) with setting this up, we might want to go with a second option.

RPi as Beacon

From poking around, it looks like there is a way that we can use RPi (which we have plenty of) to get data via BT from the devices and store it on the RPi. If these RPi are on campus, then we can easily access them through the utexas-iot network. From what I have read, we don't even need to use the API to do this meaning we spent money for no reason 👍.

There have been some projects developed on Github:

- [AirThings Wave Plus Reader](#): This script was developed by AirThings but uses Python2 still...
- [Updated AirThings Wave Plus Reader](#): Luckily, some kind soul has updated the script to Python3 for us and updated the output format (which we could also update/change.)

My thinking is that we test out this script on a few RPi that we each setup to see if it works. *If so, then we just side-stepped our biggest issue which was the connection distance to the hub.* The setup looks easy enough - we will just need:

- 3 RPi (we have both 3B+ and Zero W)
- 3 clean SD cards
- Monitors and keyboards for an initial connection

Bonus: we could setup some of the beacons to read from the AirThings monitors once this works so not only would we have Purple Air and AirThings measuring in spaces, but beacons as well.

We will want to document this process and write it on our Wiki page. As of now, I think the general steps involve:

1. Formatting an SD card with the latest Raspbian Lite
2. Connect RPi to utexas-iot
 1. Will have to register to network with Zoltan's account
 2. Update RPi config file
 3. Add device to Tailscale so we can monitor easily
3. Clone the repository linked above
4. Test script

The author suggest running the script via crontab which I guess works, but we could also borrow from the beacons to read data continuously when powered on and append to a daily csv file at regular intervals. We can leave that for a later day while we are running our case study.

Purple Air

In the past, we have developed a way to pull data from Purple Air's website. We should be able to use the same functionality, but just update the sensors that we are pulling from.

We will have to adapt the code from our [other Github page](#). My suggestion would be to clone the repository onto your computer and update the files as necessary.

- Look at the `main` method for the `purpleair_data_retriever.py`
- Looks like you will need to update the `thingspeak_keys.json` file with the sensors we are using. Currently they have meta data from the outdoor monitoring sensors.
- Then you should be able to use the `main.ipynb` notebook to download some data. I would ensure that you have the correct packages installed which are specified on the projects `readme`.

Next Steps

Immediate

Hopefully we can complete the following by the next time we meet.

- Confirm we can setup devices in the marked locations
 - Hagen to email out to various people listed above
- Device retrieval
 - We need to grab some of the devices that have been left around ECJ
- Rolling setup
 - When you are available, set up the remaining devices
 - Get the "cages" ready and order more if necessary - I think we may need one more
- Place devices in specified locations
 - Based on responses and availability of some of the people, determine if one or all of us can setup the devices
 - Schedule time to setup the devices that don't require a specific time
- Test AirThings devices connection with hub
 - Monitor the connection remotely

Secondary (but the one I am most excited about)

This should be what keeps us busy while we are gathering data from ECJ. We have easy access to some of the devices so we can always go grab them if necessary.

- Create RPi beacons to read and save AirThings data
 - Need to schedule a time for this as well since the 1-hour meeting time is likely too short to get everything worked out