Correlating CO_2 Levels With Location to Determine Potential Covid-19 Hot-spots in Everyday Life.

Joey Incandela[1]
[1]*Boston University, Boston, MA 02130, USA*

(Dated: May 2, 2021)

Since the onset of Covid-19, individuals seeking to limit their risk of exposure have had to orient their lifestyle around generalized guidelines. An important guideline we have for limiting aerosol exposure, beyond say wearing masks, is to avoid enclosed spaces with poor ventilation or poor circulation. At first glance, it seems the only actionable behaviour we can derive from this advisory is to try to limit the amount of time we spend indoors below some arbitrary threshold. We lack an easy way to evaluate the relative risk of different establishements that are a part of our daily life. As is discussed in a preprint by Peng et al.[1], ambient indoor CO_2 levels correlate with density of viral particles, and so a measurement of CO_2 can be used as a proxy for air quality and exposure risk. Working off of this fact I want to design a portable device that records CO_2 level alongside GPS location as we go about our day. With the data collected during an outing (or multiple outings), we could create a heat-map of Covid-19 risk overlayed onto a literal map, (i.e. Google Maps). In this way, users should have access to quantitative measurements that allow them identify risky establishments and adjust their lifestyle to minimize risk.

I. DESIGN CONCEPT

The device is expected to have 5 main components;

- An independent power supply (9V battery), which allows the Uno to operate without a computer.
- A CO_2 sensor for measuring ambient air quality.
- A GPS Module for recording the device's coordinates.
- A formatted USB thumb-drive for recording all measurement data.
- An Arduino Uno to control the modules.

The device should have two fundamental states when powered; One active and one passive measurement state. The device should default to the 'passive' state, in which it will measure CO_2 at a comparatively low frequency. During this phase, the controller will compare passive measurements to an established CO_2 baseline, waiting for signal to surpass that threshold. Note that the baseline will have to be calibrated using an independent set of measurements. Furthermore, the GPS module will be powered off in the passive state, and no data will be recorded to the attached drive.

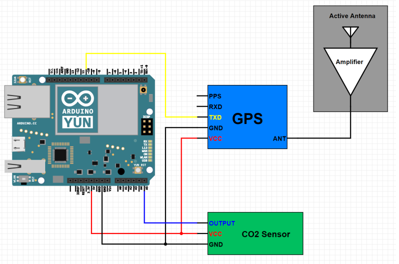

FIG. 1: Schematic Diagram

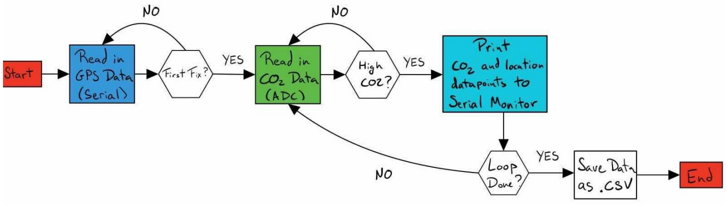

FIG. 2: Software Flow Chart

Once signal is measured to surpass the predetermined level, the device will shift to it's 'active' state. Here, the device will measure CO_2 and location simultaneously for a set time period, at the highest frequency possible. (The 'highest' frequency will be bottle-necked by whatever is the slowest module.). At the end of each active phase the data will be recorded to the drive, ideally in the form of a 4-5 column .csv file.

The columns recorded should be;

- 1 column with the data point timestamp.
- 1 column with the raw CO_2 measurement.
- 2-3 columns (Latitude Longitude or additional Altitude) recording the device's coordinates.

At the end of an outing, the Arduino and all modules will be powered off, and the thumb drive removed. The data can then be brought onto a computer, and ported into google maps for hot-spot analysis.

II. PARTS LIST

- Arduino Uno (Owned)
- 16GB Flash Drive (Owned)
- 9V Battery with Jack connector (Owned)

- GPS Module (Received)
- CO_2 Detector (Received)

III. GAME-PLAN

- Week 8: Acquire Parts
- Week 9: Individual Testing of CO2, and GPS module control. Test saving data arrays to formatted drive.
- Week 10: Attempt to measure and record data in concert. Take measurements in field.
- Week 11: Work on data analysis/ overlaying data onto google maps.
- Week 12: By now I would hope to be at the Debugging/optimization stage.

IV. USEFUL REFERENCES

- An introduction to GPS in the context of Arduino.

  https://create.arduino.cc/projecthub/ruchir1674/how-to-interface-gps-module-neo-6m-with-arduino-8f90ad

- A better alternative to TinyGPS Library for arduino:

  https://github.com/SlashDevin/NeoGPS
  https://github.com/SlashDevin/NeoGPS/blob/master/extras/doc/Data%20Model.md#usage

- A better alternative to Software Serial Library for arduino: https://github.com/SlashDevin/NeoSWSerial

- References and debug tools for prefered GPS module with U-Blox Neo-6M chipset: https://www.gie.com.my/download/um/wireless/gps/obsolete/um_gps.pdf http://cdn.sparkfun.com/datasheets/Sensors/GPS/760.pdf https://www.u-blox.com/en/product/u-center#tab-documentation-resources

[1] A. Z. Peng and J. L. Jimenez (2020), URL https://doi.org/10.1101/2020.09.09.20191676.