DCV Logic for Damper Control

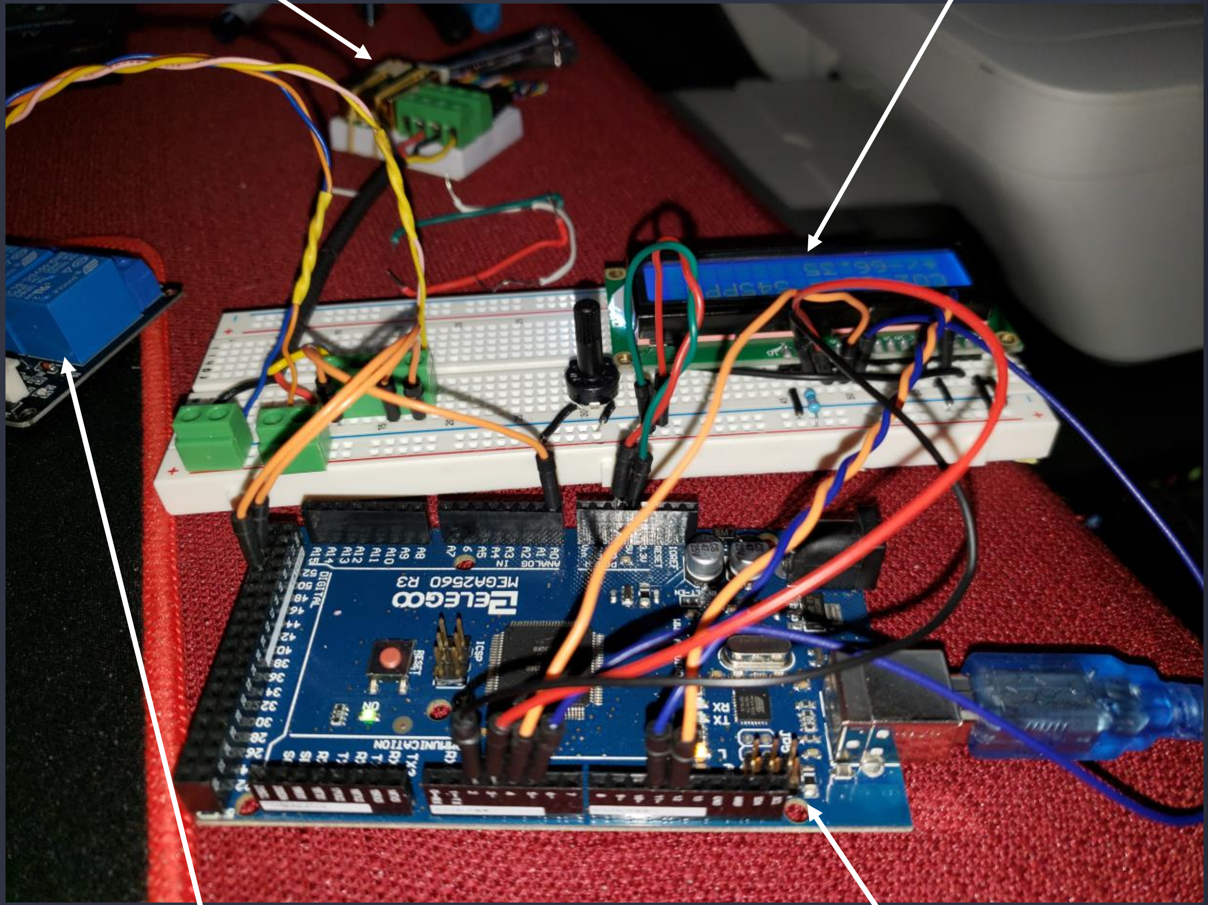
Team 1 –

Alex Craig, James Head, Caleb Hutchings, Ryley Tharp, Chris Whitton

Setup of Sensor and Damper

MHZ-19B CO2 Sensor

Breadboard and LCD

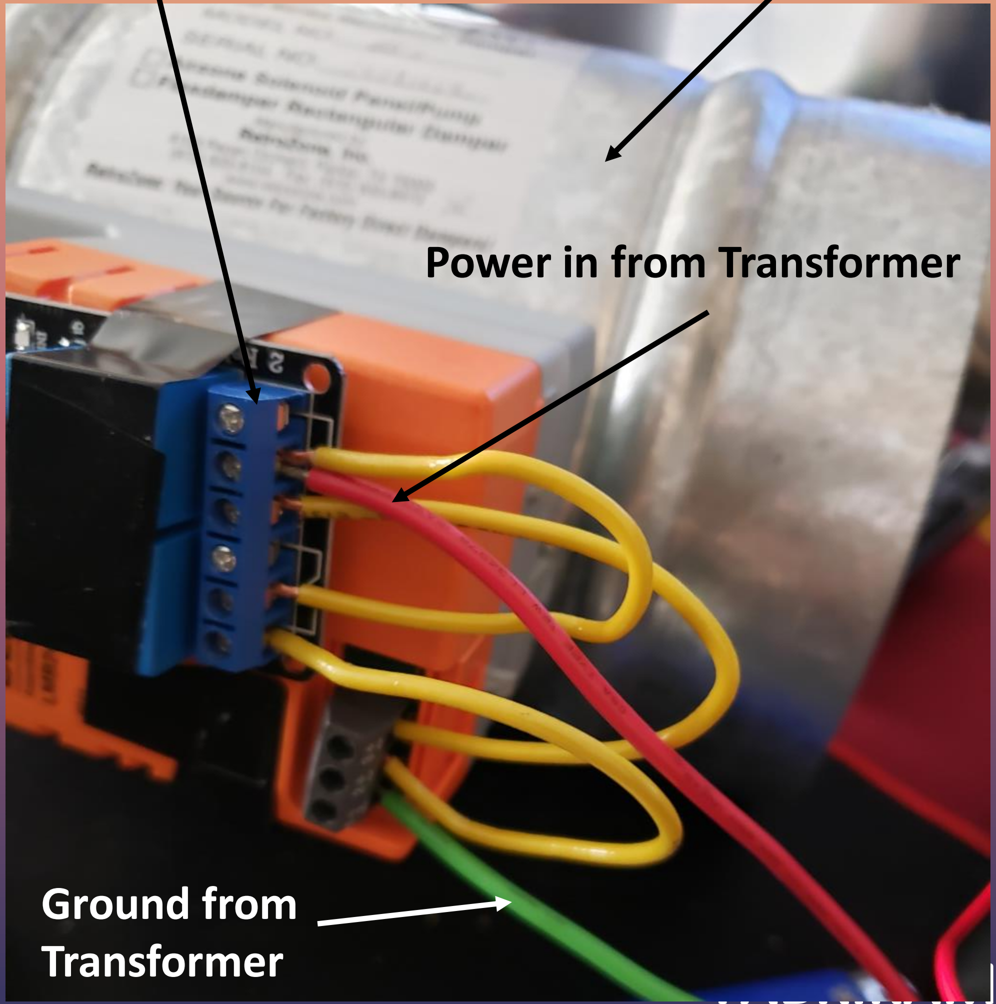


Dual Relay Switch

Elegoo MEGA 2560 Arduino Board

Dual Relay Switch

Motorized Damper



The Real World

What we learned through research

CO2 Dispersion

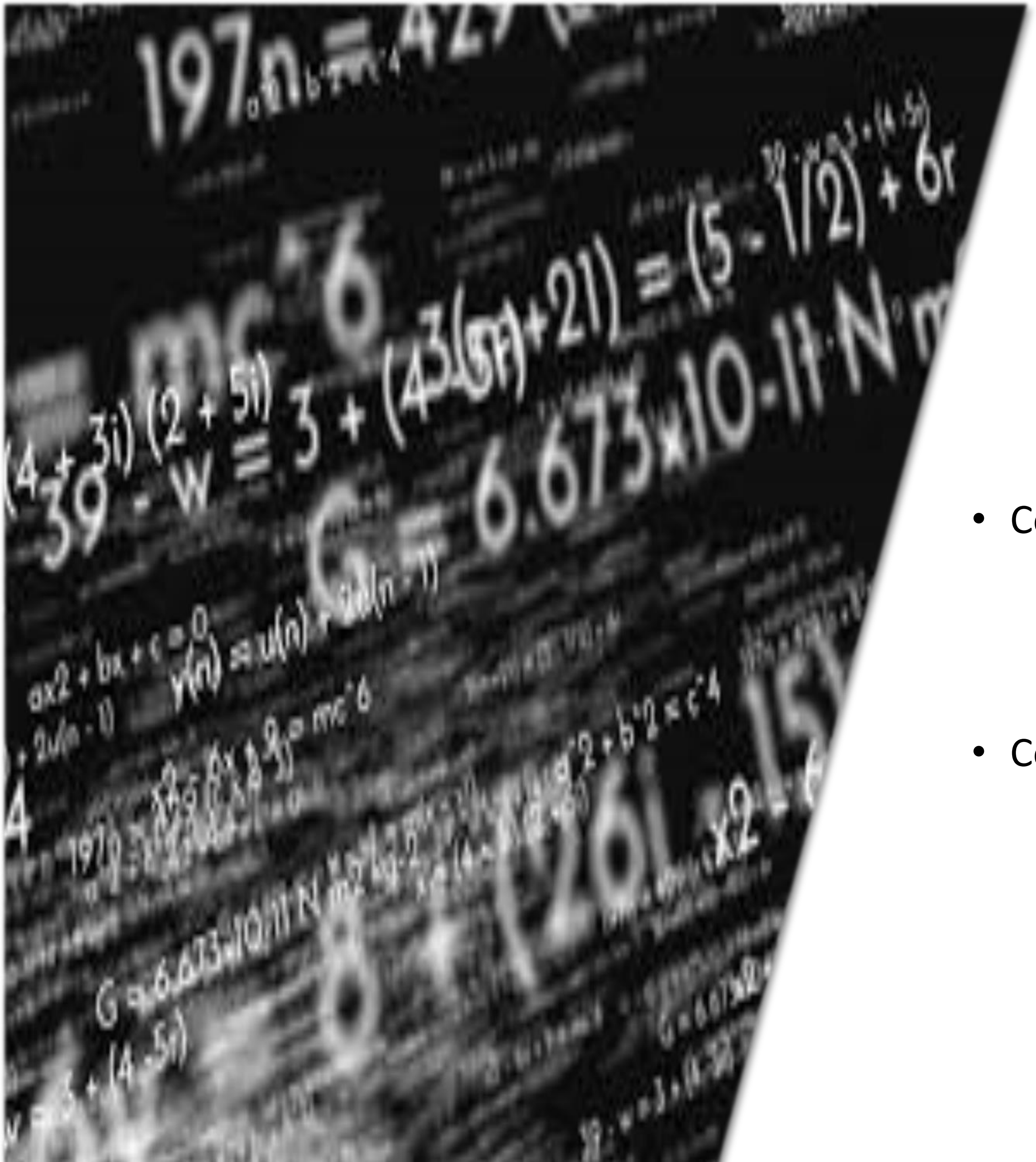
What to Consider:

VARIABLE	SYMBOL	UNIT
Number of People	N	-
Volume	V	m^3
Initial CO2	Ci	ppm
Measured CO2	Co	ppm
Critical CO2	C1	ppm
CO2 per Person	q	m^3/hr
Number of Exchanges per Hour	n	1/hr

$$t = \frac{\ln\left(\frac{\left(C_o - \frac{qN}{nV} - C_i\right)}{\left(C_1 - \frac{qN}{nV} - C_i\right)}\right)}{n}$$

Why we chose **not** to use that formula

- Requires several variables
- Only establishes time until critical level
- Does not consider flow rate
- Number of persons can change



The Equations

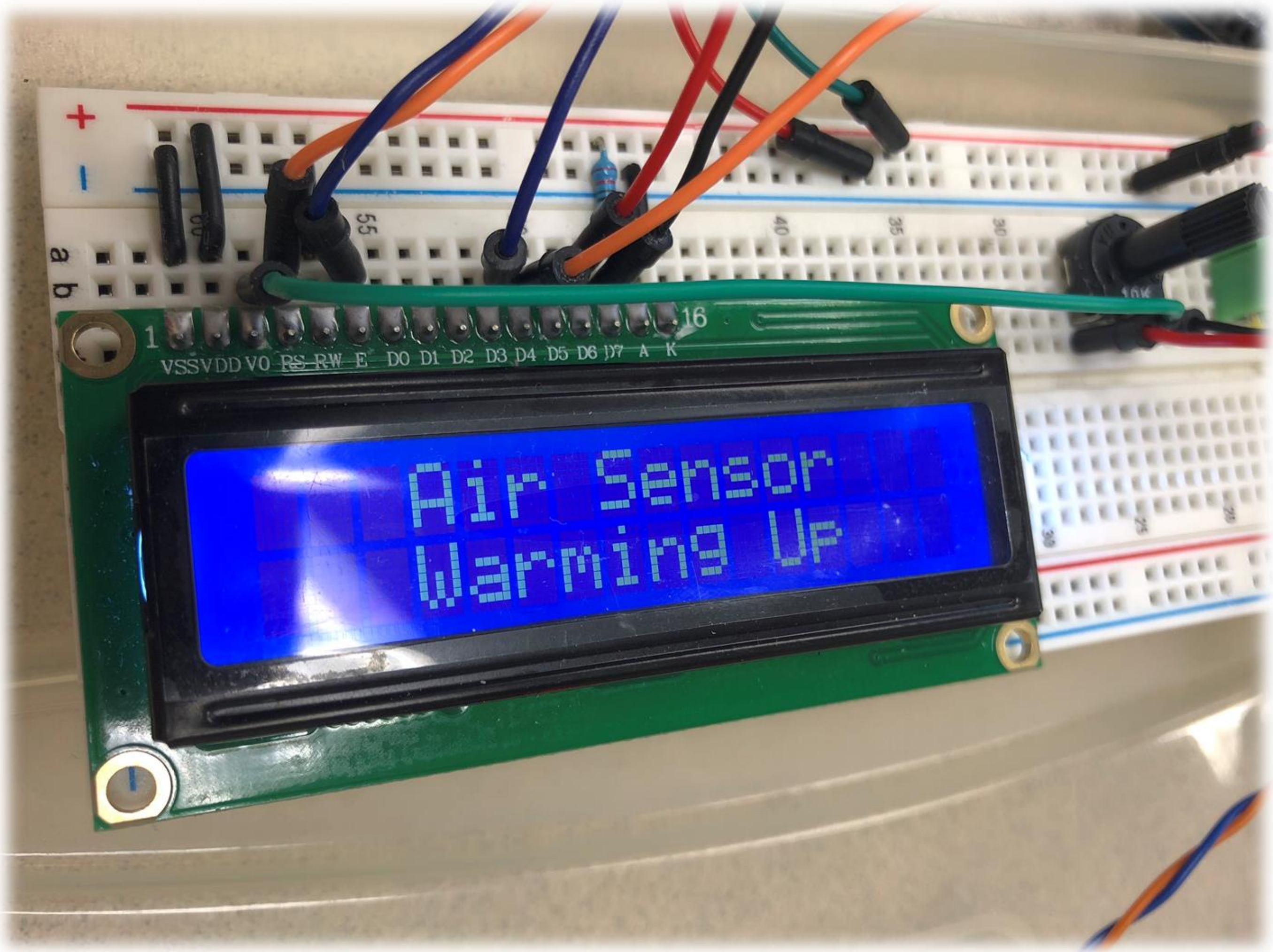
- Conversion from CO_2 measurement to degrees:
  - $Measured\ CO_2(ppm) * \frac{Maximum\ Degree\ of\ Openness\ (^o)}{Critical\ CO_2\ Change\ (ppm)}$
- Conversion from degrees to travel time:
  - $Degrees\ * \frac{Total\ Travel\ Time\ (sec)}{Maximum\ Degree\ of\ Oppenness\ (^o)}$

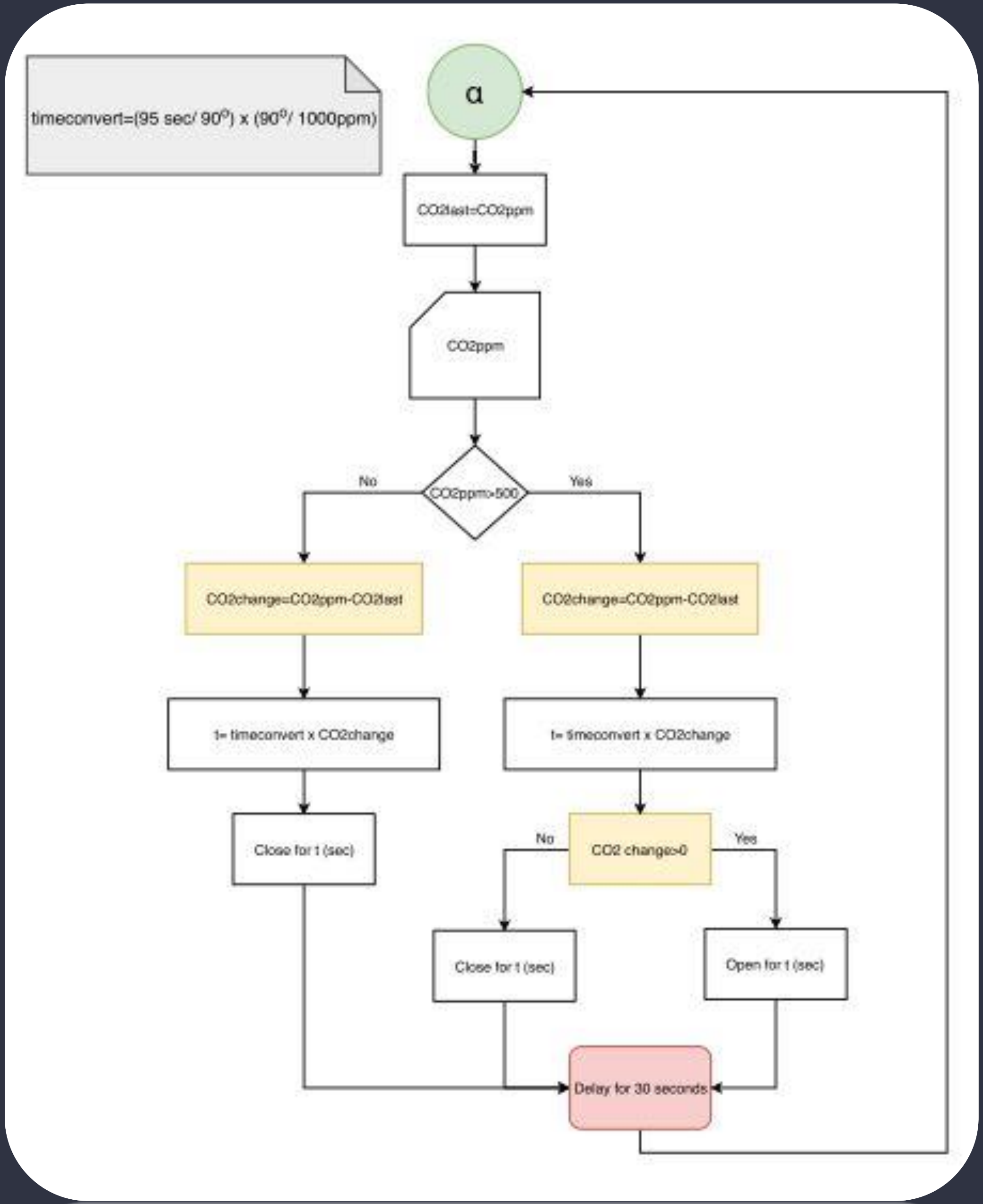
What We Chose and Why

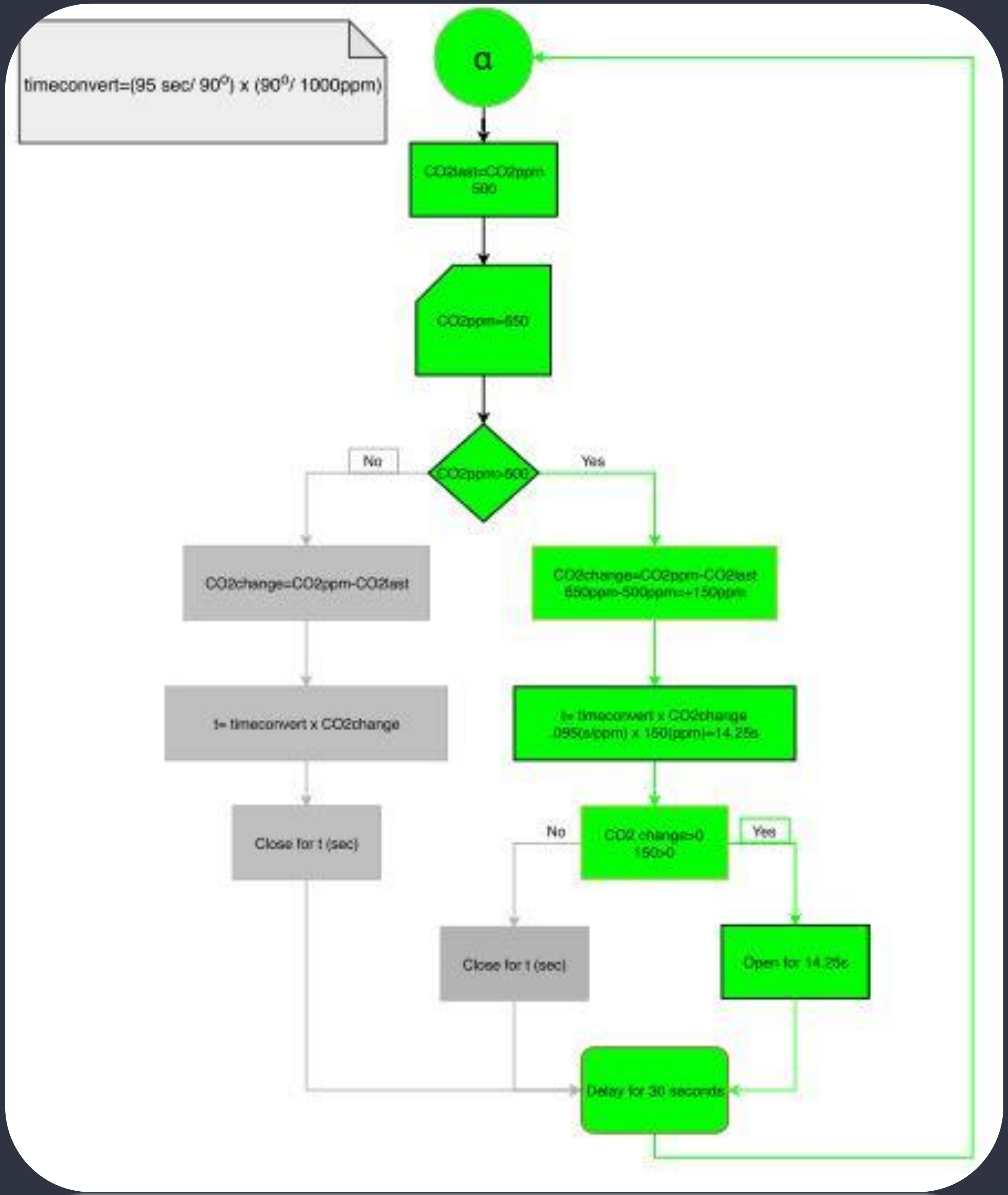
- Simplistic: Only requiring one variable
  - The sensor provides all of the information
- Arduino is limited in storage of previous measurements
  - Can only remember one previous value
- Team is limited in coding knowledge
  - Unfamiliar with Arduino coding language and only limited training in regards to general coding
- Manages damper position and allows for variable airflow
- Provides real-time feedback from sensor to damper
- Can track damper position relative to the x-axis

The Logic

- Initial Setup
  - The Sensor requires time to initialize and stabilize.
- Damper Returns Home
  - The damper closes in order to establish a “home” position of 0^{o}.




timeconvert=(95 sec/ 90^0) x (90^0/ 1000ppm)
α
CO2last=CO2ppm
CO2ppm
CO2ppm>500
No
Yes
CO2change=CO2ppm-CO2last
CO2change=CO2ppm-CO2last
t= timeconvert x CO2change
t= timeconvert x CO2change
Close for t (sec)
CO2 change>0
No
Yes
Close for t (sec)
Open for t (sec)
Delay for 30 seconds


timeconvert=(95 sec/ 90^0) x (90^0/ 1000ppm)
α
CO2last=CO2ppm
500
CO2ppm=650
CO2ppm>500
No
Yes
CO2change=CO2ppm-CO2last
CO2change=CO2ppm-CO2last
650ppm-500ppm=+150ppm
t= timeconvert x CO2change
t= timeconvert x CO2change
.095(s/ppm) x 150(ppm)=14.25s
Close for t (sec)
CO2 change>0
150>0
No
Yes
Close for t (sec)
Open for 14.25s
Delay for 30 seconds

RESULTS

- Shown are the first and last 3 readings
- Reading 15 was taken after the sensor was exhaled upon
- This illustrates the variable nature of the logic in relation to CO_2 levels and damper open position

	1	2	3	13	14	15
Sensor Reading (ppm)	624	632	628	644	646	**783**
Uncertainty (+/-)	68.72	68.96	68.84	69.32	69.38	**73.49**
Change in CO2 (ppm)	0	8	-4	2	2	**137**
Movement Time (sec)	0.00	0.76	-0.38	0.19	0.19	**13.02**
Damper Position (degrees)	11.16	11.88	11.52	12.96	13.14	**25.47**
Direction	CLOSE	OPEN	CLOSE	OPEN	OPEN	**OPEN**

CONCLUSIONS

What we learned:

- Demand Control Ventilation has many factors to consider
- To monitor CO_2 levels, a sensor is very beneficial
- Using this sensor, a stable concentration of CO_2 can be maintained
- The logic process takes careful consideration
  - Streamlining this process comes with time and experience
- Proficiency in logical thinking and coding of Arduino and Excel systems was achieved

The Experiment:

- The CO_2 sensor works
- The position of the damper can be controlled
- The position of the damper allows a variable flow rate into the system
- In theory, the damper could effectively control the concentration of CO_2 in a real-world environment

Questions?