Interview Presentation

Nathan Chung

2025-01-27

Who am I

- Mechanical design engineer by experience (but I like doing other things too)
- I like to raise dogs

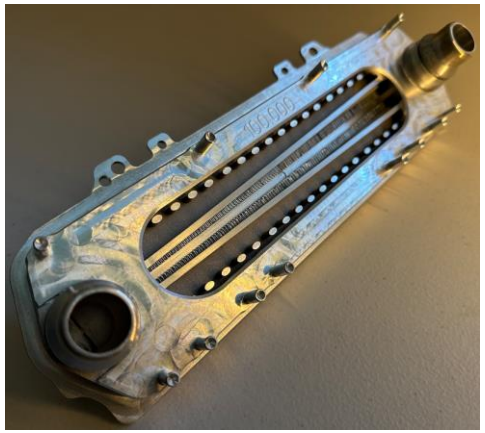
Rocky, 17

Cookie, 2

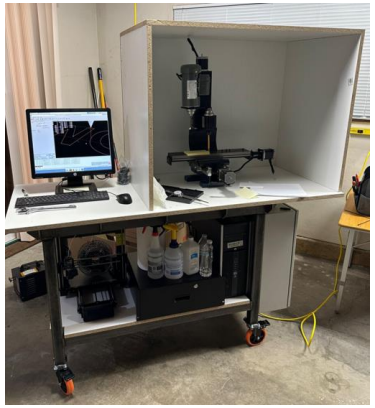
- I like to raise interns

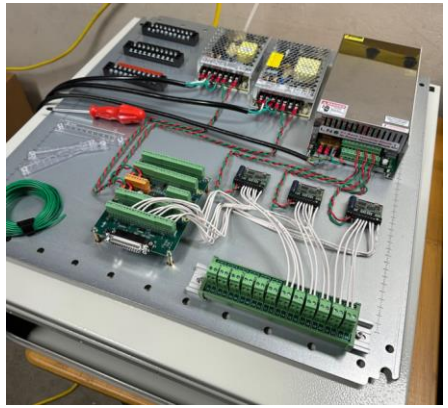
Notable Projects

Tesla Drive Inverter Heatsink, Patent Link



LinuxCNC

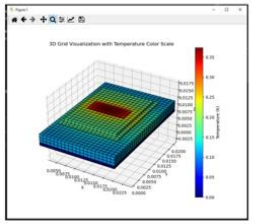


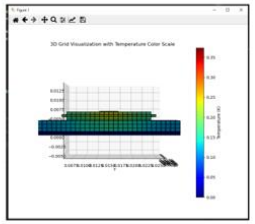


Thermal FEA Software, Version 1 Link, Version 2 Link

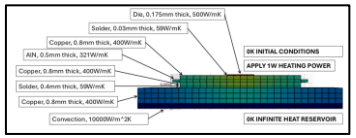
Sanity Check

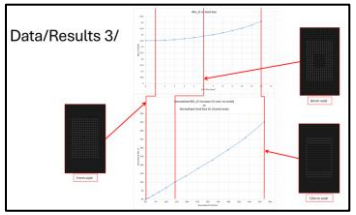
- Shown is a simulation run of 3 seconds.
- dT/dt is 0.0218 K/s @ 3 seconds (considering this converged)
- Power at die is 1W
- T_die_avg = 0.353 K
- T_copper_amb_b_avg = 0.205 K (results in roughly half of Rth_JC)
- This agrees quite well with empirical thermal impedance data, well enough for the purposes of this investigation, anyhow



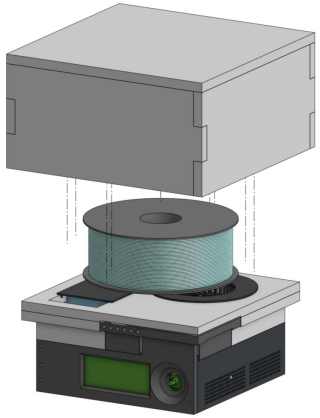


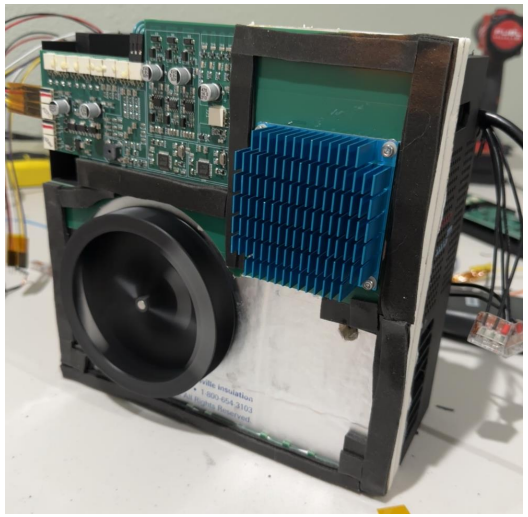
```
t:      2.997700        T:      0.353   dT/dt:  0.0218
t:      2.997800        T:      0.353   dT/dt:  0.0218
t:      2.997900        T:      0.353   dT/dt:  0.0218
t:      2.998000        T:      0.353   dT/dt:  0.0218
t:      2.998100        T:      0.353   dT/dt:  0.0218
t:      2.998200        T:      0.353   dT/dt:  0.0218
t:      2.998300        T:      0.353   dT/dt:  0.0218
t:      2.998400        T:      0.353   dT/dt:  0.0218
t:      2.998500        T:      0.353   dT/dt:  0.0218
t:      2.998600        T:      0.353   dT/dt:  0.0218
t:      2.998700        T:      0.353   dT/dt:  0.0218
t:      2.998800        T:      0.353   dT/dt:  0.0218
t:      2.998900        T:      0.353   dT/dt:  0.0218
t:      2.999000        T:      0.353   dT/dt:  0.0218
t:      2.999100        T:      0.353   dT/dt:  0.0218
t:      2.999200        T:      0.353   dT/dt:  0.0218
t:      2.999300        T:      0.353   dT/dt:  0.0218
t:      2.999400        T:      0.353   dT/dt:  0.0218
t:      2.999500        T:      0.353   dT/dt:  0.0218
t:      2.999600        T:      0.353   dT/dt:  0.0218
t:      2.999700        T:      0.353   dT/dt:  0.0218
t:      2.999800        T:      0.353   dT/dt:  0.0218
t:      2.999900        T:      0.353   dT/dt:  0.0218
t_max reached, terminating solver
          time  T_silicon  T_copper_amb_b
0       0.0000     0.000           0.000
1       0.0001     0.001           0.000
2       0.0002     0.002           0.000
3       0.0003     0.002           0.000
4       0.0004     0.003           0.000
...        ...       ...             ...
29995   2.9995     0.353           0.205
29996   2.9996     0.353           0.205
29997   2.9997     0.353           0.205
29998   2.9998     0.353           0.205
29999   2.9999     0.353           0.205

[30000 rows x 3 columns]
```



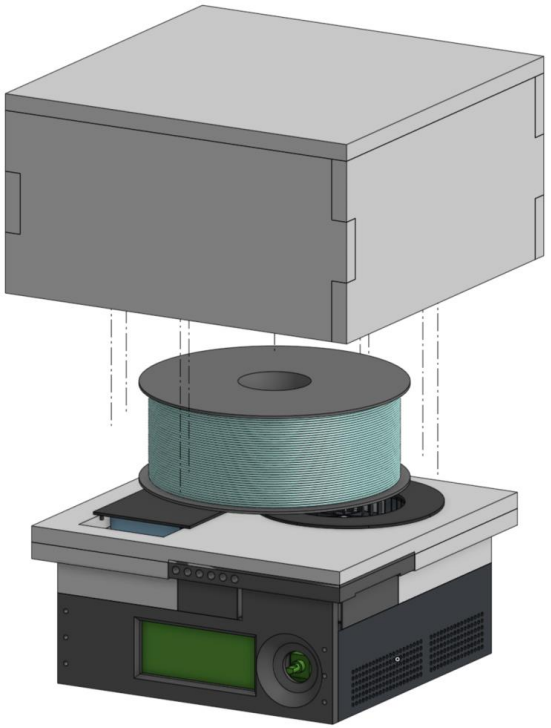


Filament Dryer



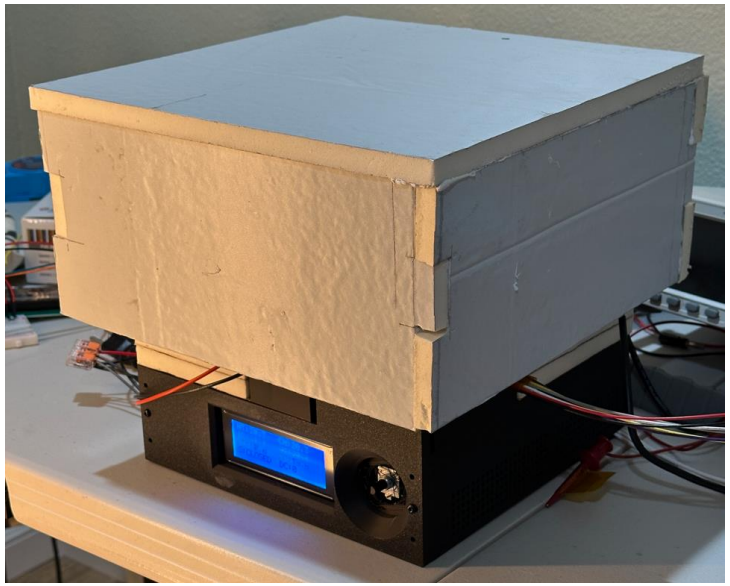


Project: 3D Printer Filament Dryer

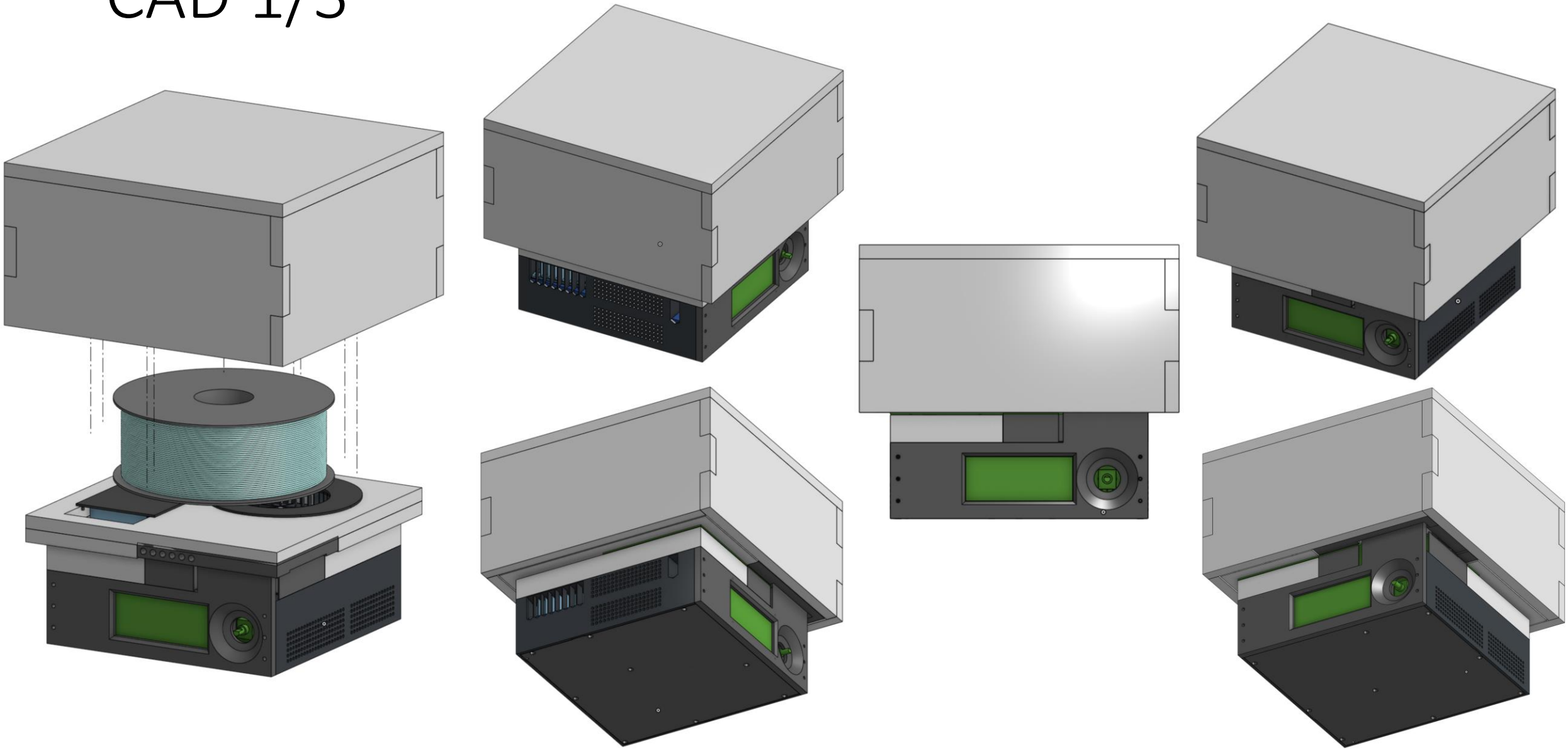


- What is this:
  - Plastic-drying oven
  - Intended for sale

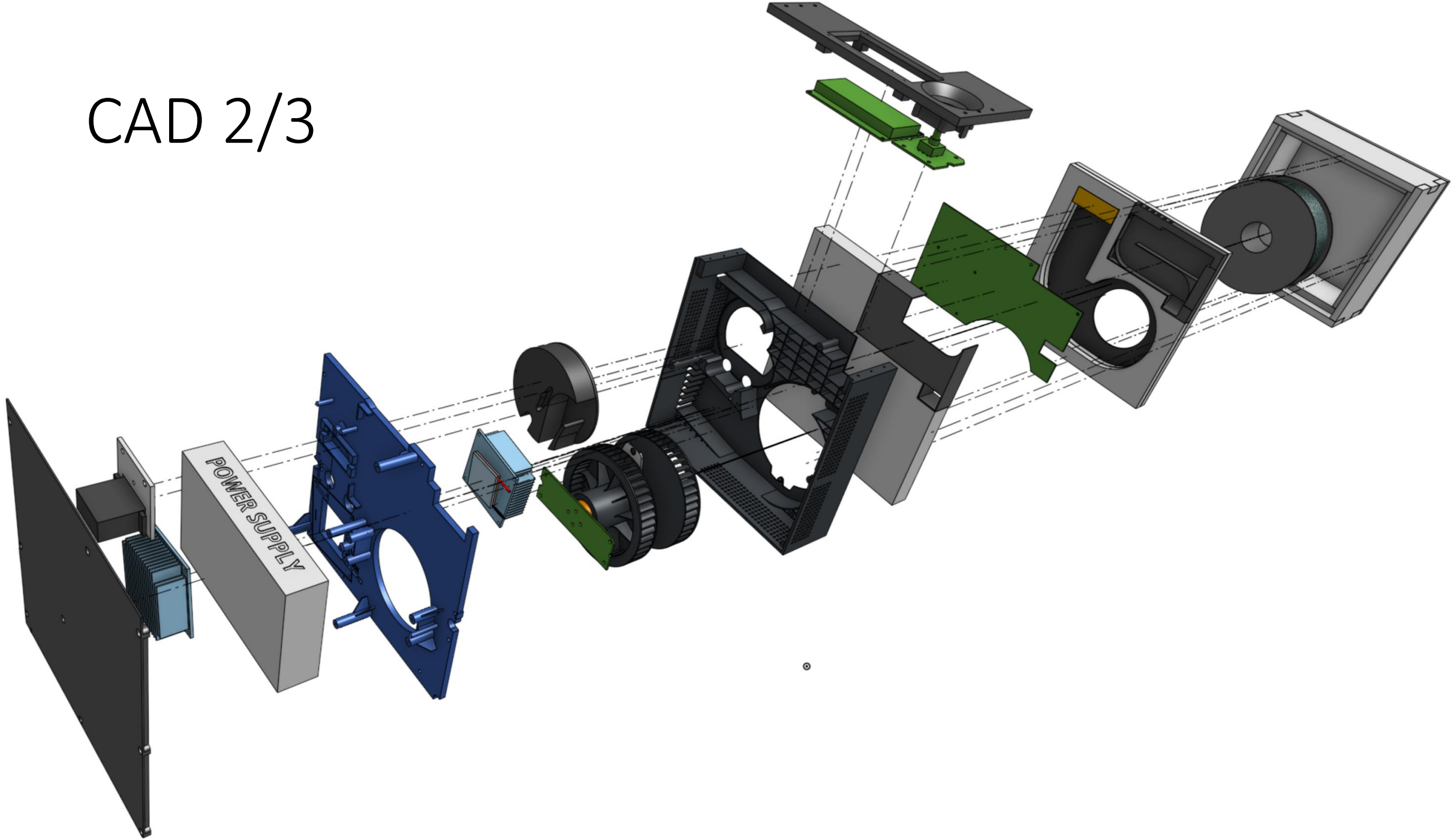
- Specs:
  - 100°C chamber
  - Input air pre-dryer
  - 150W peak power consumption
  - <$100 BOM cost at scale
  - Ships in box size: 12” x 12” x 6”



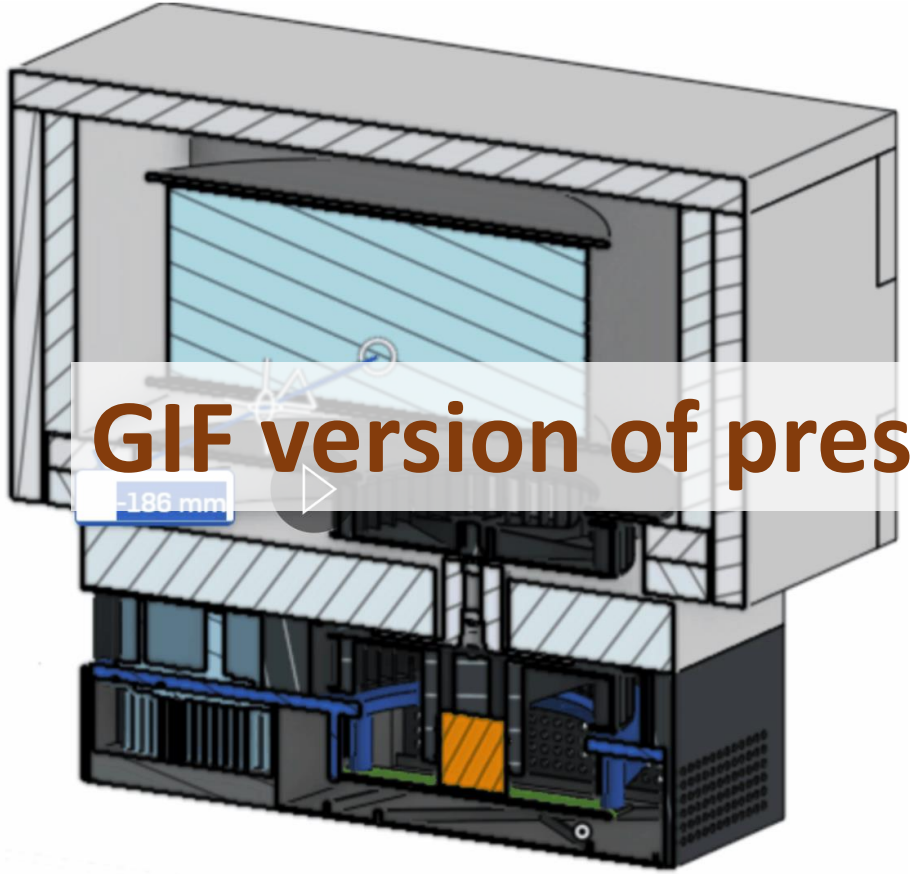
CAD 1/3



CAD 2/3



CAD 3/3

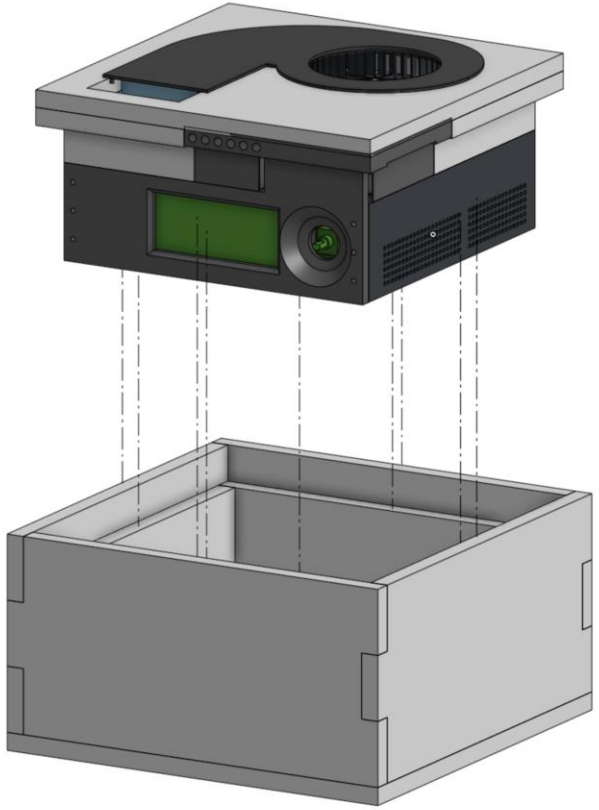


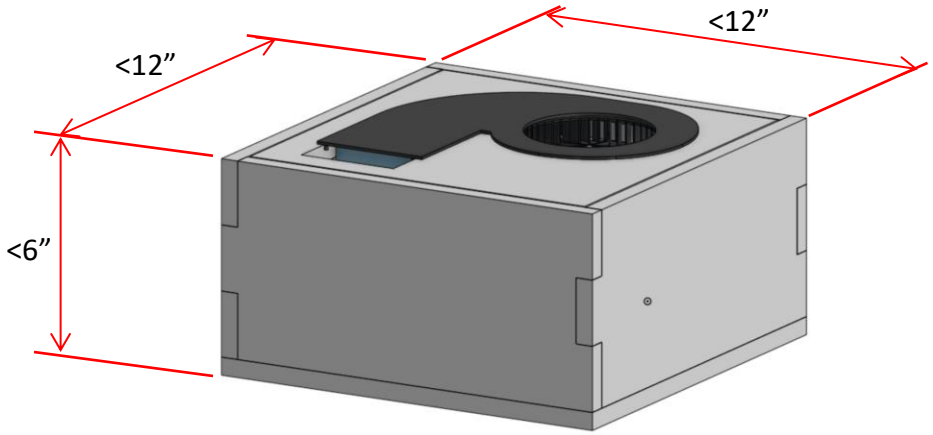
GIF version of presentation is available

Shipping

- Ships in a 12” x 12” x 6” box
- Additional ¼” allowance for protective padding all sides
- Shipping air is expensive. Amazon’s most competitive products are all packaged efficiently
  - 12”x12”x6”
    - $26.30 flat rate via USPS Flat Rate
    - $28.49 SF to NYC via UPS Ground
  - 12”x12”x**12”**
    - $41.71 SF to NYC via UPS Ground

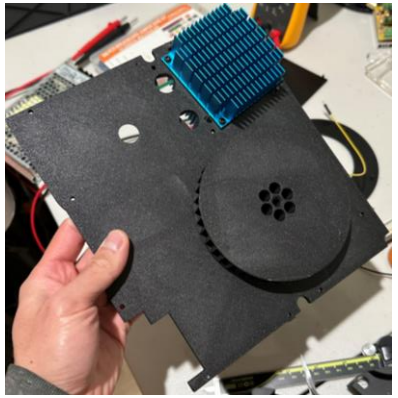
- Reminder, BOM cost target was $100

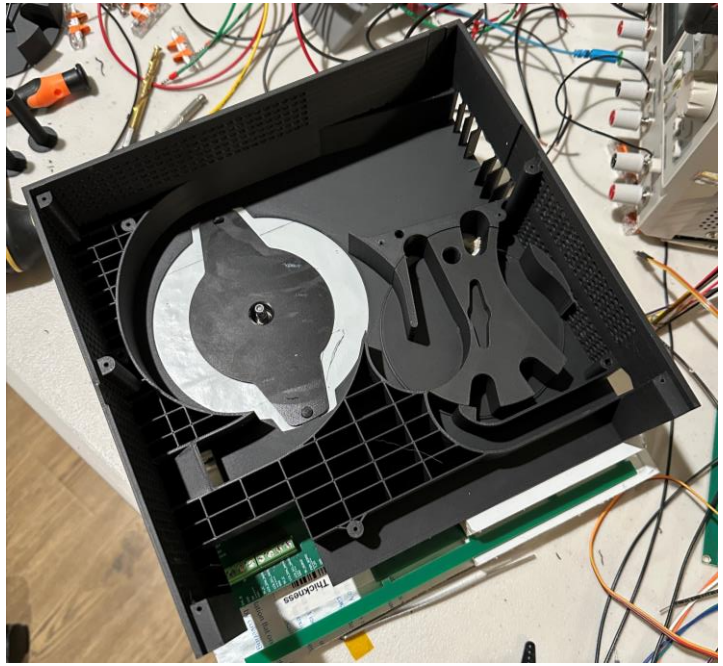


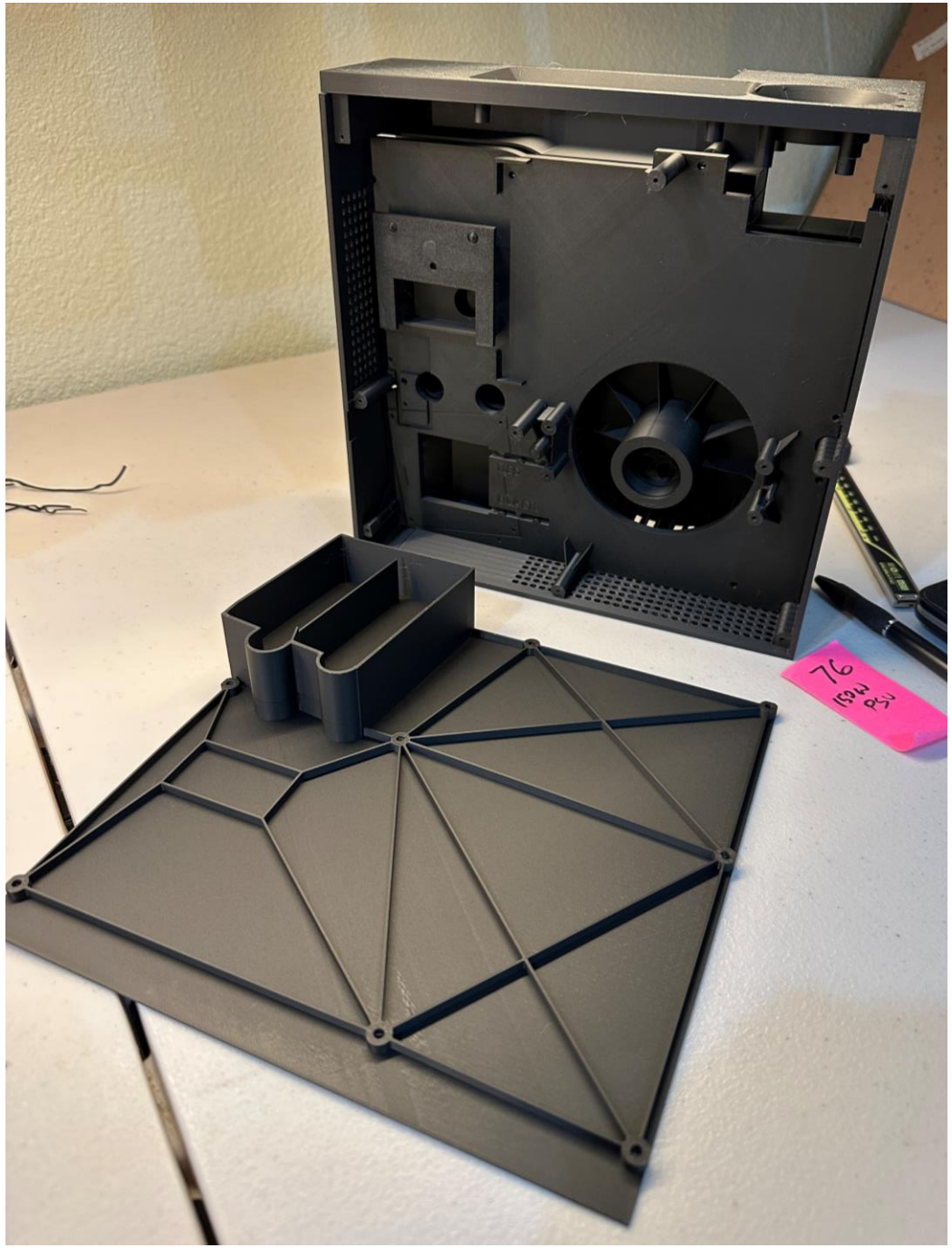


3D Printing

- PLA + PA6
- Bambu X1C + Dryer

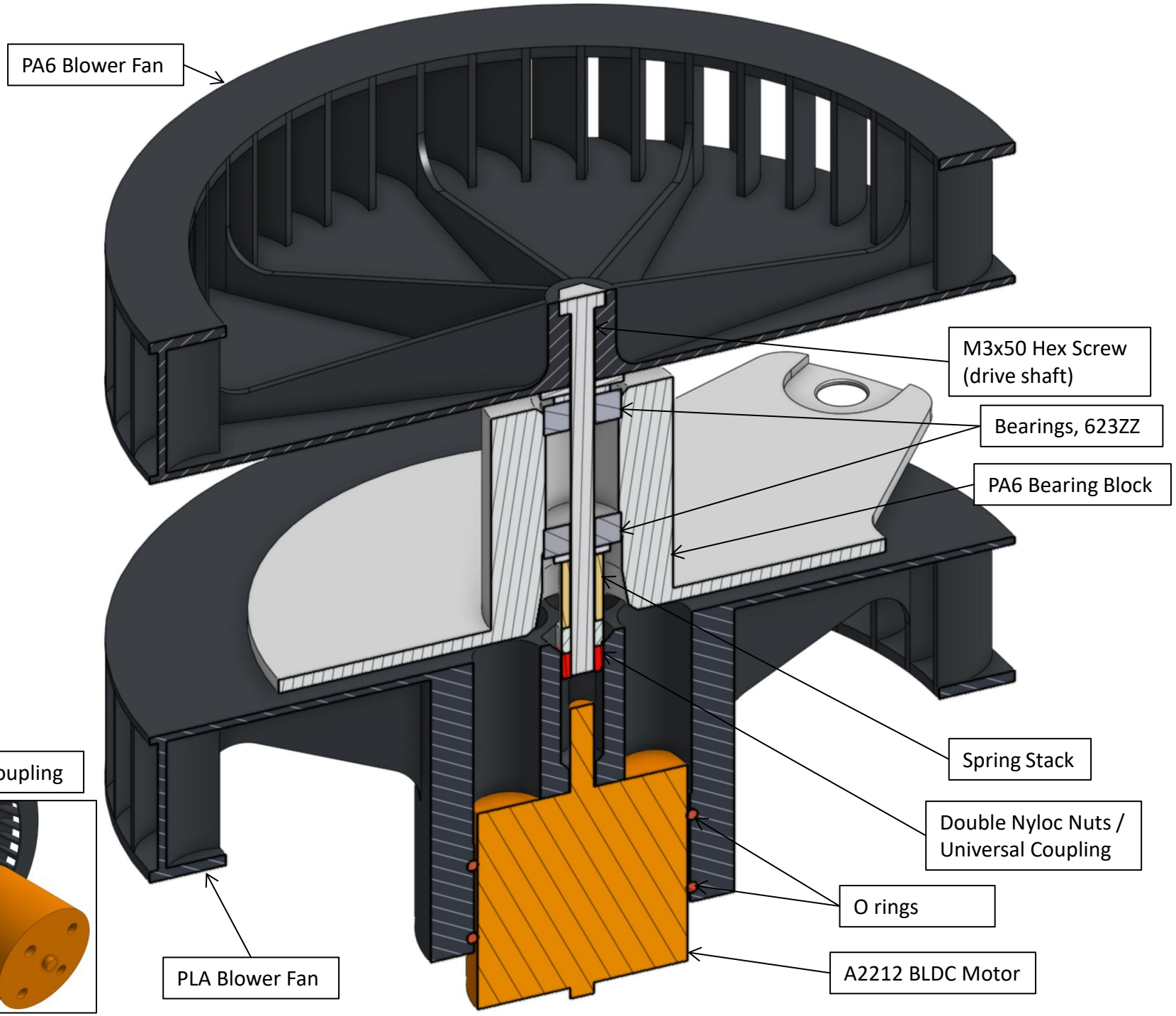




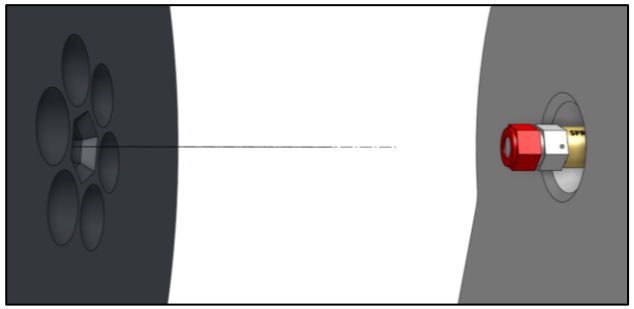


Blower 1/2

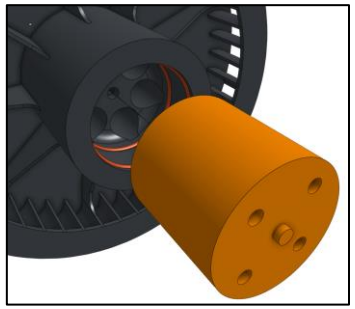
- 1 motor, 2 blowers
- 100°C Max, 2k RPM
- 3D Printed



Hex coupling between 2 blowers

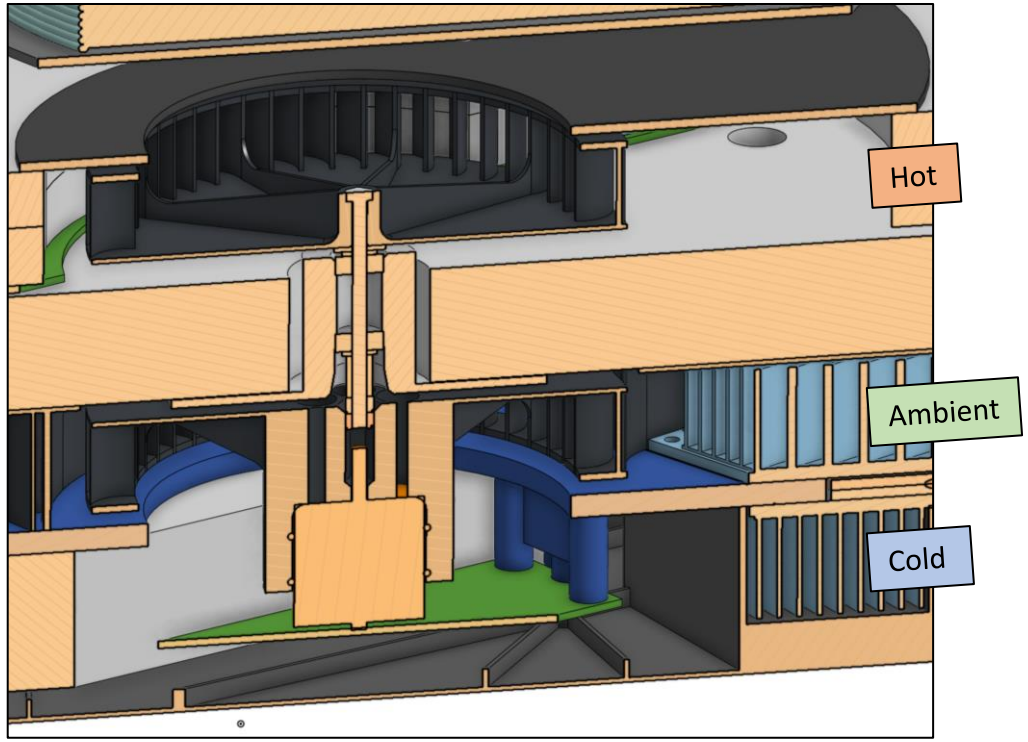


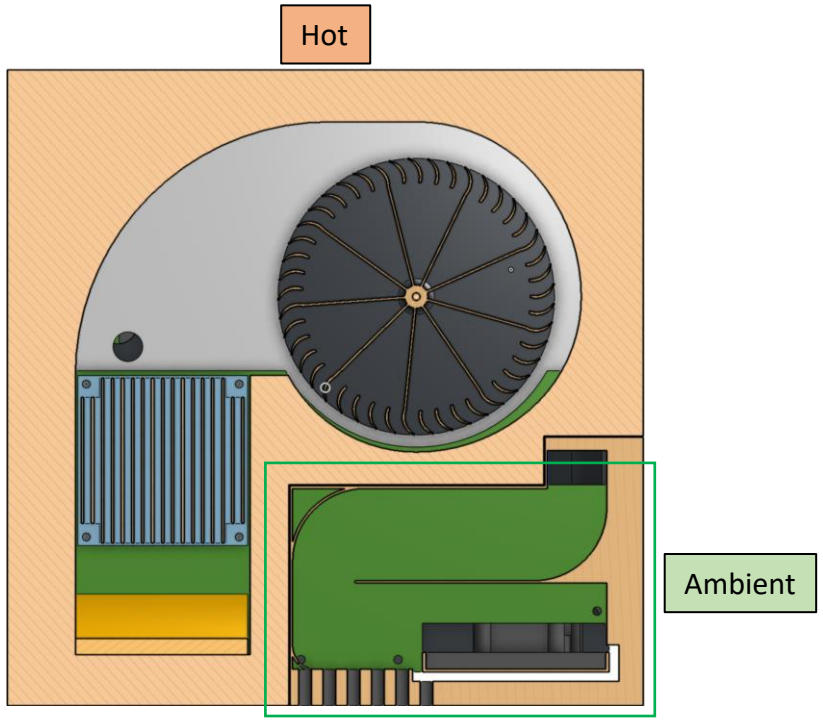
Motor oring coupling

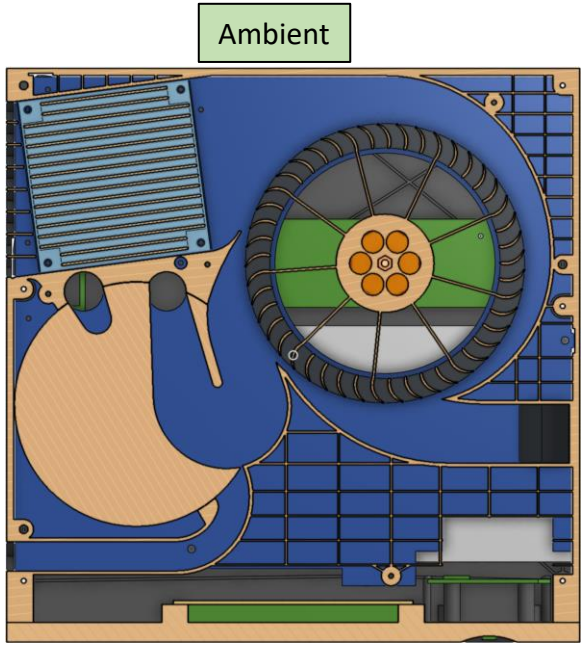


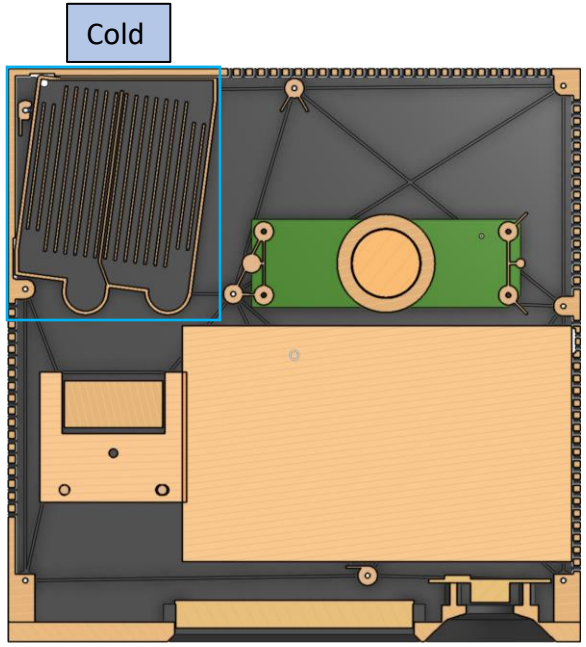
Blower 2/2

- Serves 3 domains
  - Hot (oven chamber)
  - Ambient (electronics cooling)
  - Cold (condenser)





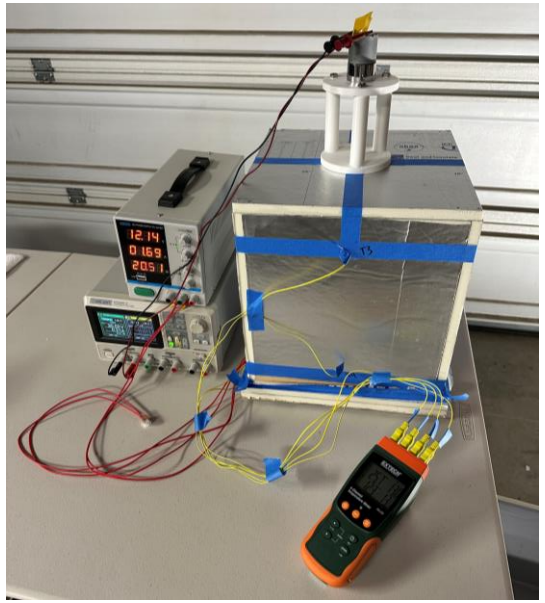




Insulation

- Polyisocyanurate foam board
  - Pros
    - Good insulator. Early chamber tests demonstrated 1.5~2.0 K/W Rth
    - Rigid/hard enough to be structural
    - Inexpensive
    - Accessible @ Lowes
    - Non-hazardous to process
  - Cons
    - Foilized, shorting hazard.
    - Without proper tooling, cut edges look very rough
    - Sheet stock has many imperfections (it's a home construction material after all)
    - Glued assemblies look amateurish. Hard to make look good.
    - Exposed foam edges have poor wear durability from everyday handling

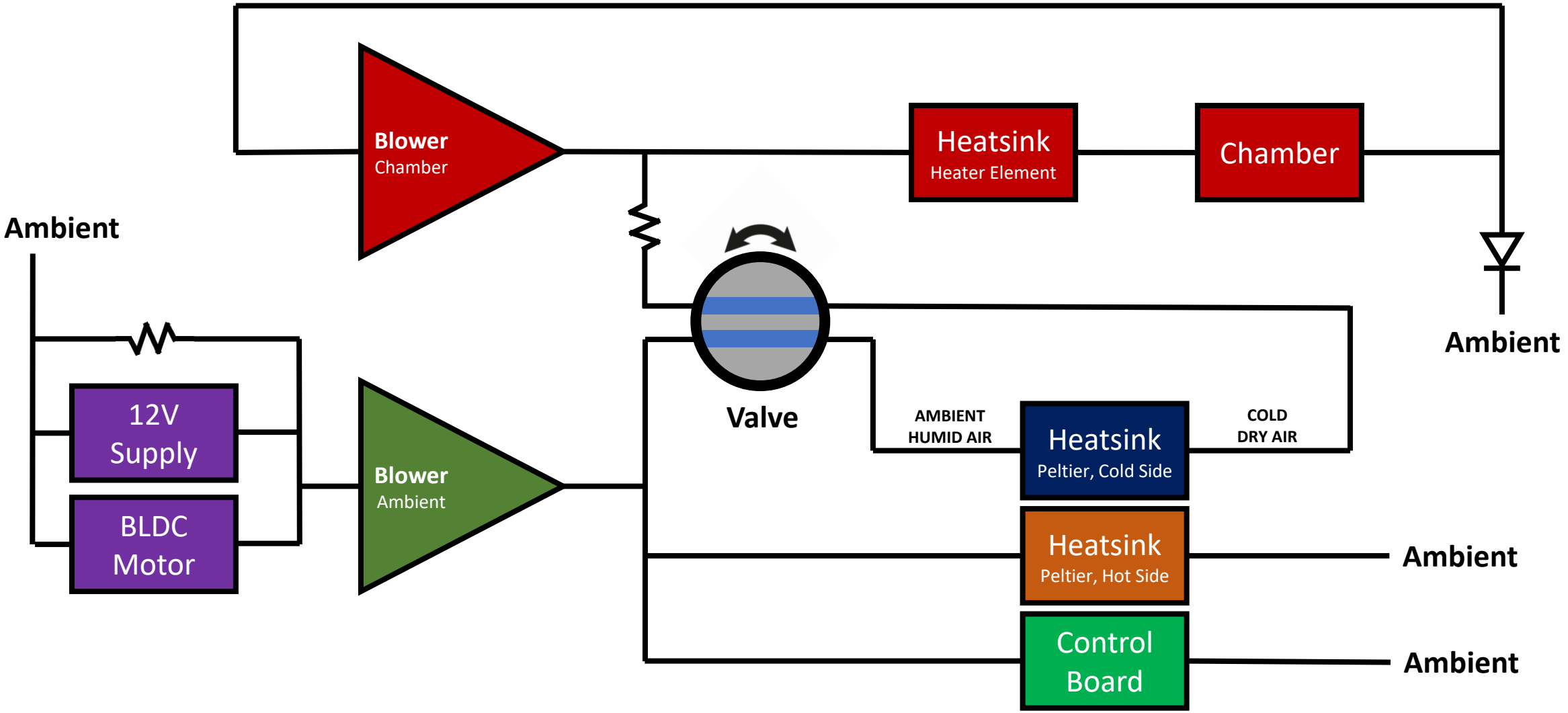
Early A-sample tests



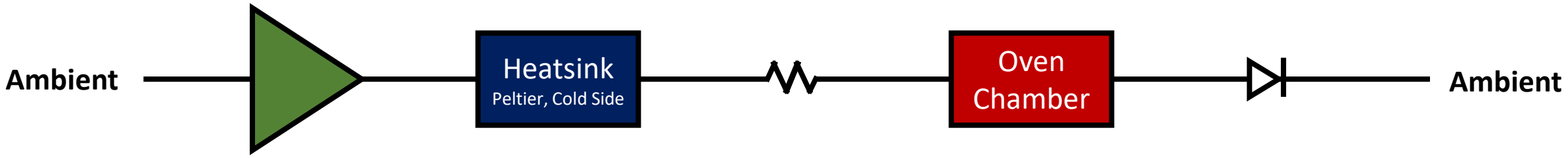
½" vs 1" thick chambers

Test Description	Rth_chamber-ambient [K/W]
1/2" walls, passive convection, unsealed	1.4
1" walls, passive convection, unsealed	1.9
1" walls, passive convection, sealed	2.0
1" walls, active convection, sealed	1.9

Airflow Schematic



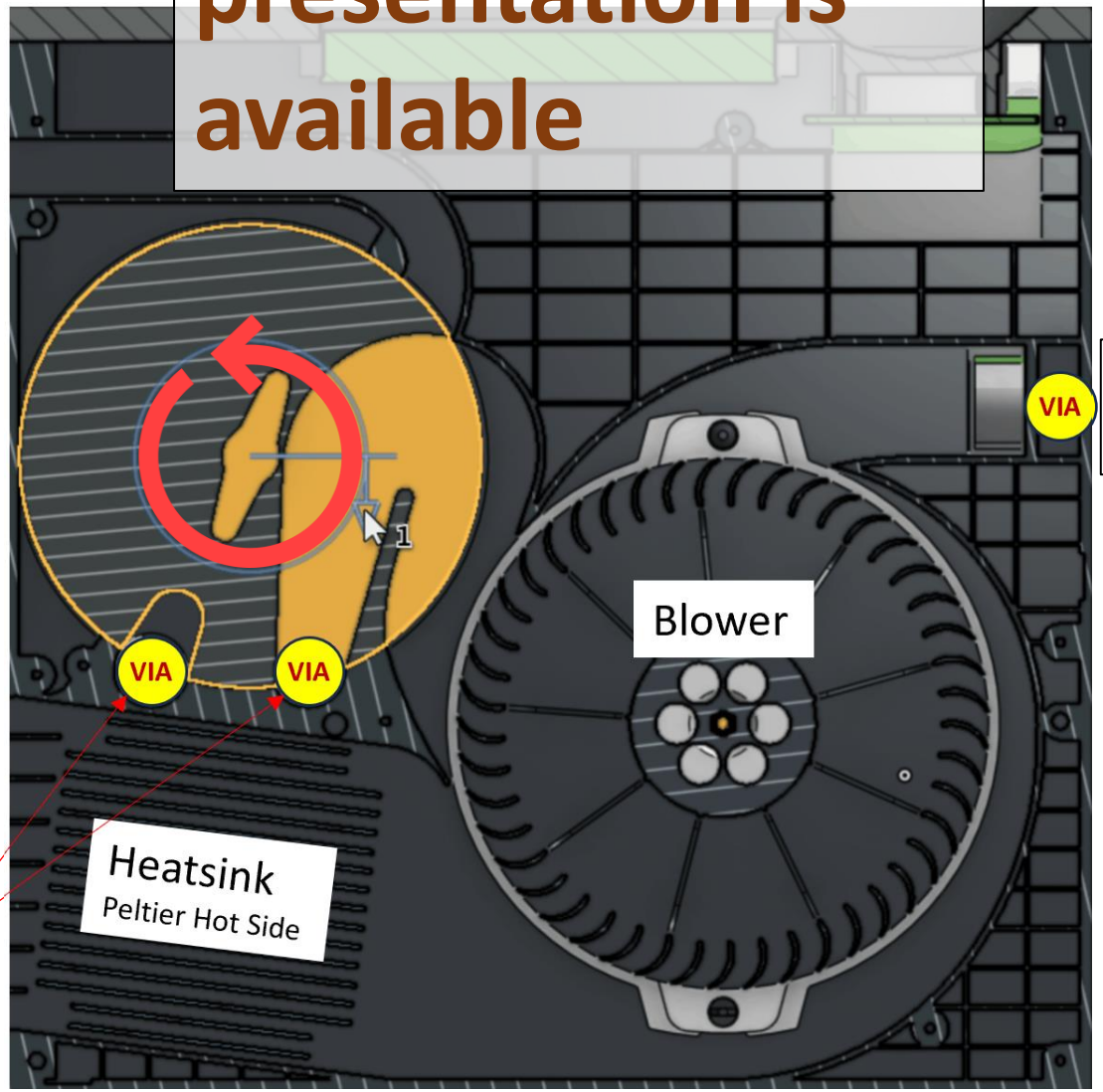
Airflow Schematic, Bare Minimum



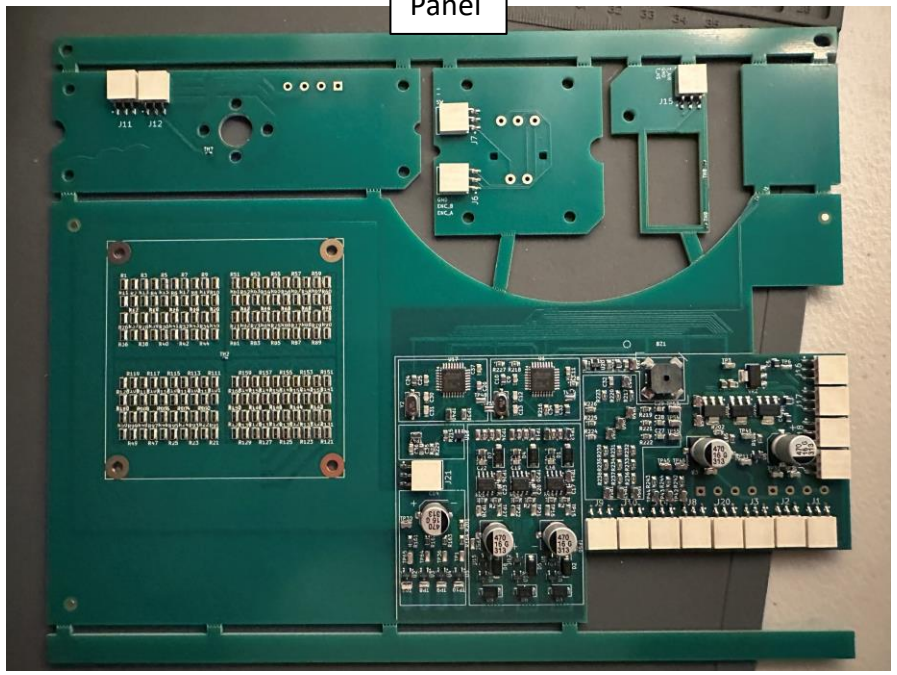
Valve

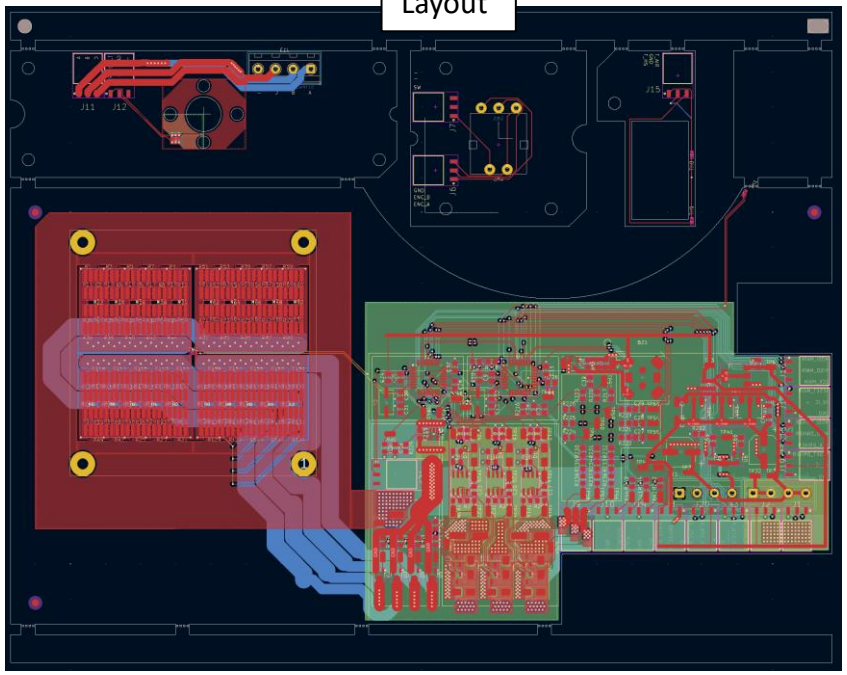
- This valve controls input airflow through the pre-dryer.
- Intention was to enable intermittent use of a lower-capability condenser.
  - We'd let the condenser "cold charge" without airflow, and then open the valve to send air through.
  - Predryer air output look like rising step response as the condenser heats up. We cut off this transient wherever the predryer output air is deemed too high.

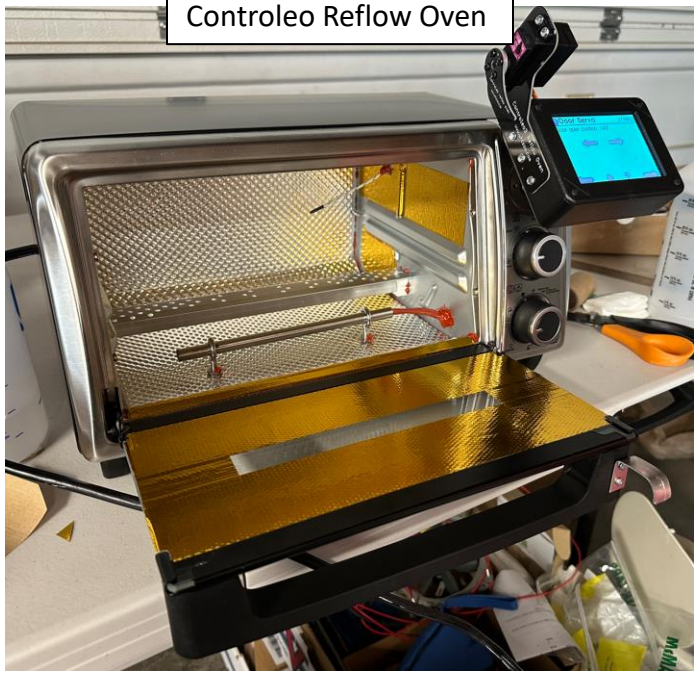
GIF version of presentation is available

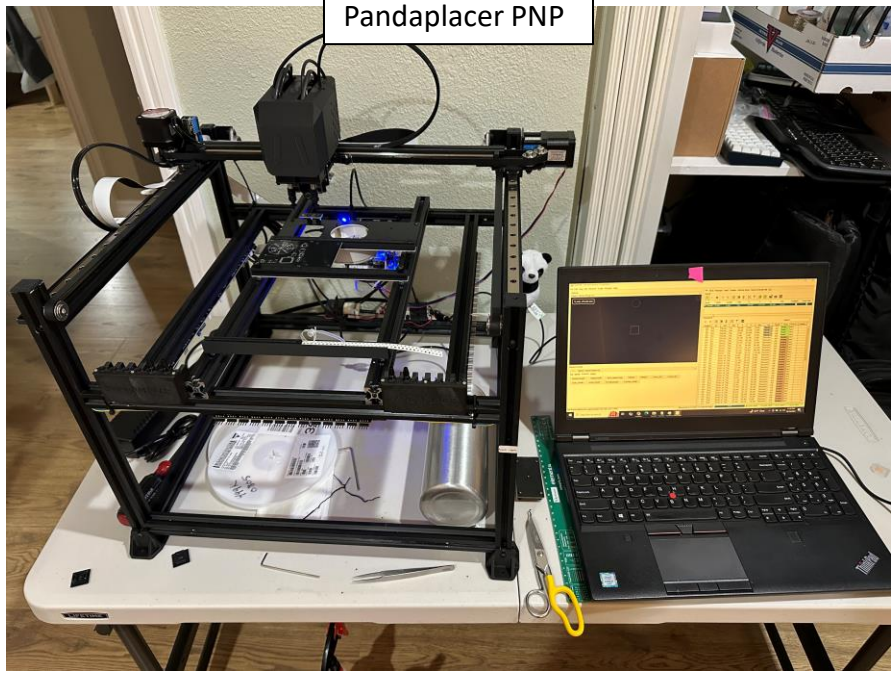


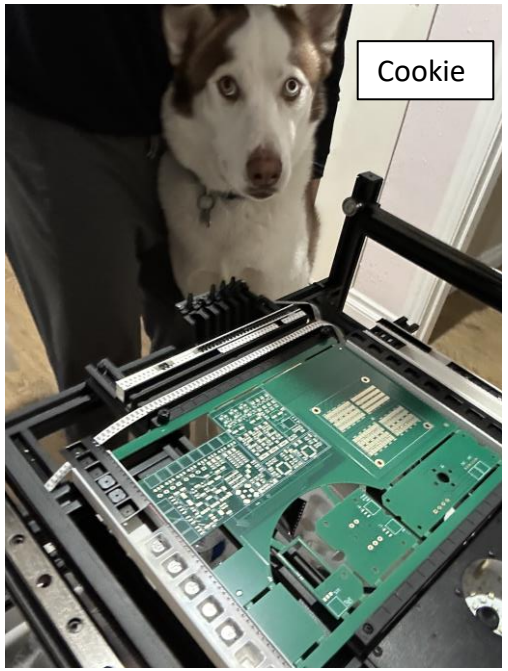
PCBA 1/2


Panel

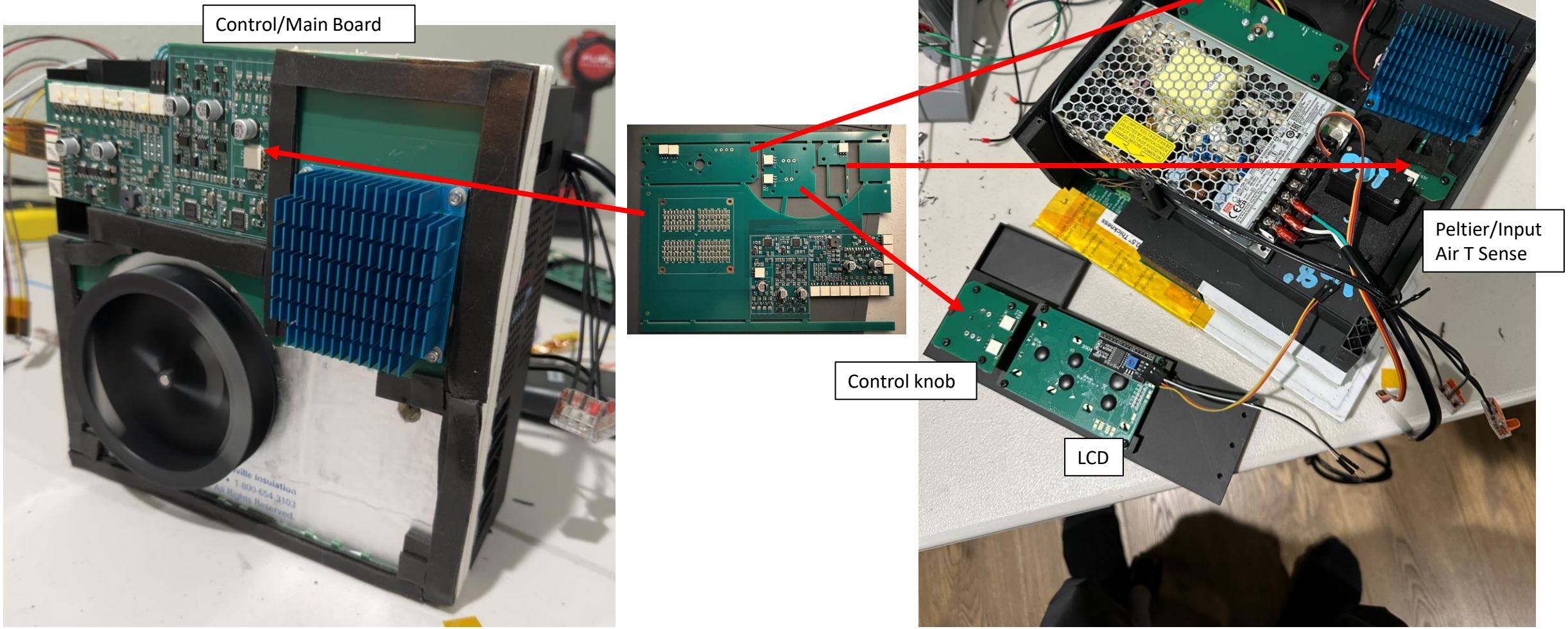

Layout


Controleo Reflow Oven


Pandaplacer PNP


Cookie

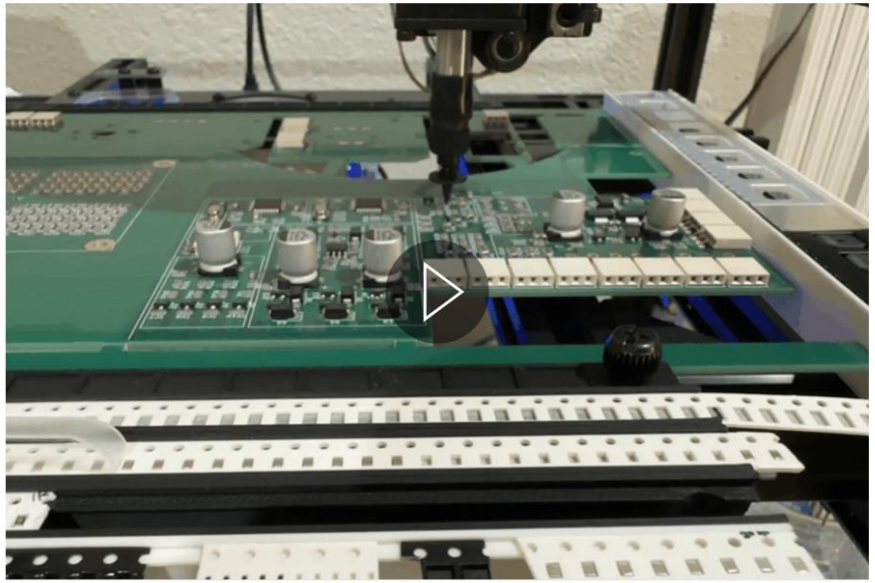
PCBA 2/2


Control/Main Board
Motor Bracket / Wire Adapter
Peltier/Input
Air T Sense
Control knob
LCD

Board Assembly Tools

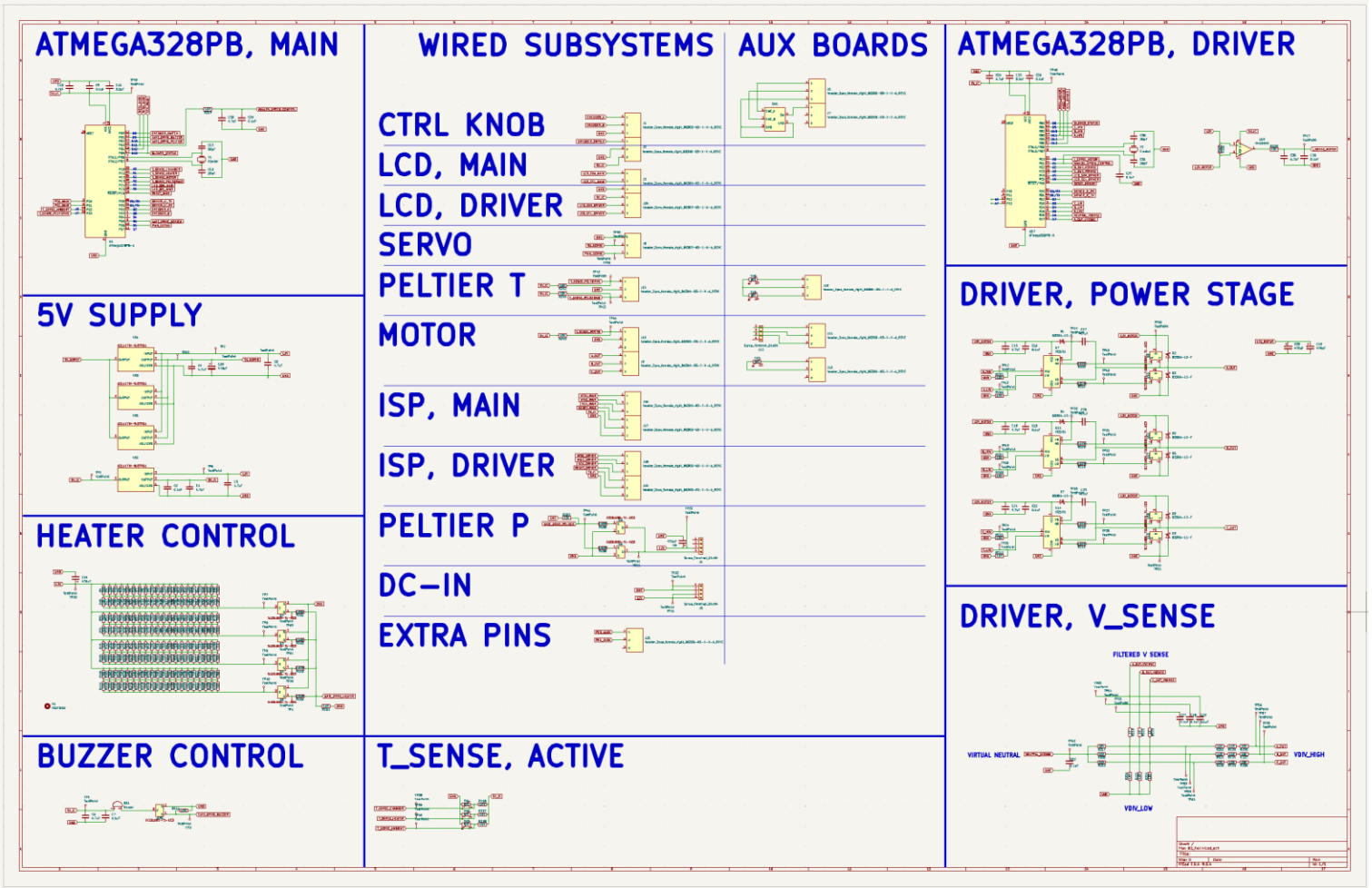
- Assembly houses are expensive
- Instead, get
  - $300 reflow oven kit
    - Controleo 3
    - Pb-free capable!!
    - Uses a $40 toaster oven as base
  - $800 PNP machine kit
    - Pandaplacer A1
    - 0402 capable!
    - OpenPNP
- Enabled PCBA designs that would be very difficult to hand-assemble.
  - 0402 passives + tight pitch footprints
  - Dense layouts
  - High component count

GIF version of presentation is available



Schematic

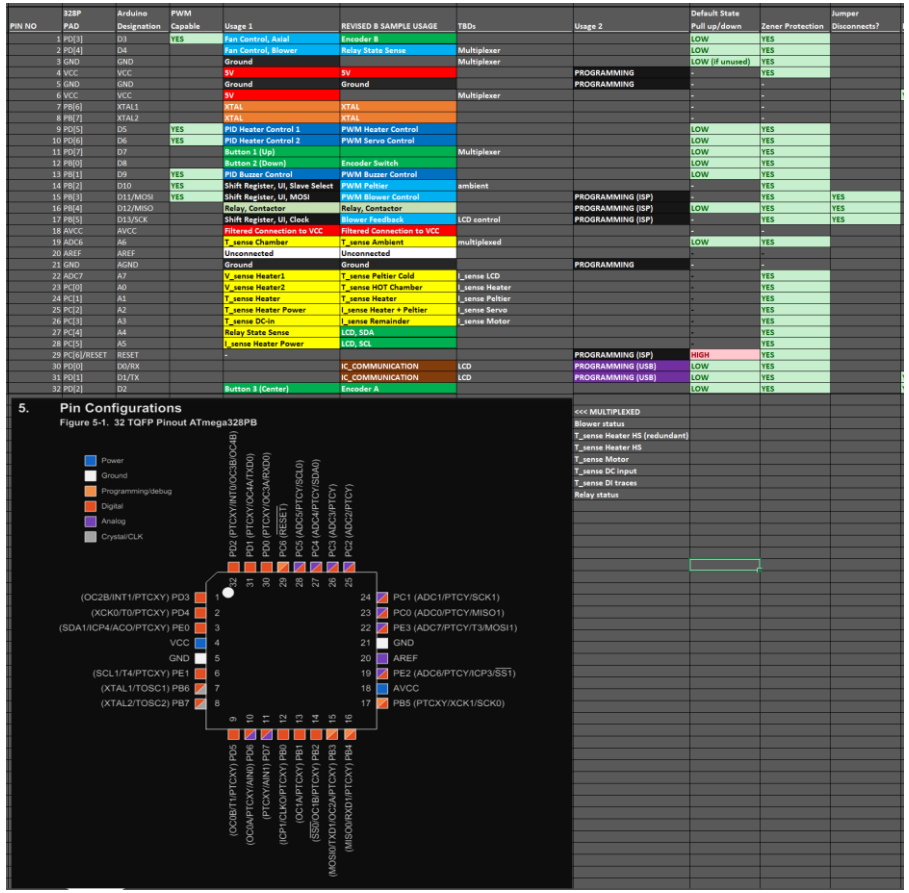
- 2x Atmega328
- 12V + 5V
- No 120V, for touch safety
- Notable subsystems
  - Inverter
  - Heater
  - Peltier
  - Passive-overtemperature protection



ATmega328PB

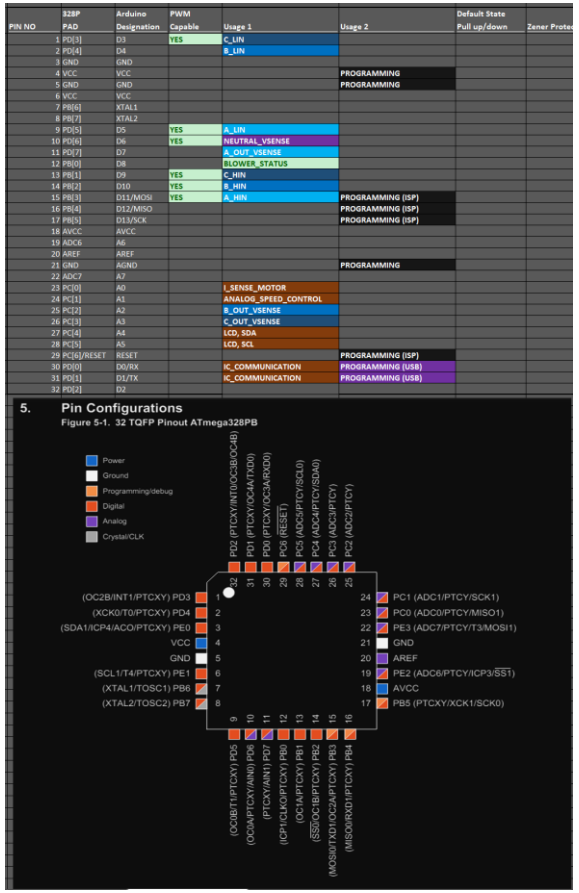
- Same chip used in Arduinos
- Pros
  - Cheap
  - Easy to program
  - Beginner-friendly
- Cons
  - Not enough pinouts
  - Bad ADC resolution
  - No wifi/BT capability
  - Actually bad for motor control

Main controller



PIN NO	328P PAD	Arduino Designation	PWM Capable	Usage 1	REVISED B SAMPLE USAGE	TBDs	Usage 2	Default State Pull up/down	Zener Protection	Jumper Disconnects?
1	PD[3]	D3	YES	Fan Control, Axial	Encoder B			LOW	YES	
2	PD[4]	D4		Fan Control, Blower	Relay State Sense	Multiplexer		LOW	YES	
3	GND	GND		Ground		Multiplexer		LOW (if unused)	YES	
4	VCC	VCC		5V	5V		PROGRAMMING	-	YES	
5	GND	GND		Ground	Ground		PROGRAMMING	-	-	
6	VCC	VCC		5V		Multiplexer		-	-	
7	PB[6]	XTAL1		XTAL	XTAL			-	-	
8	PB[7]	XTAL2		XTAL	XTAL			-	-	
9	PD[5]	D5	YES	PID Heater Control 1	PWM Heater Control			LOW	YES	
10	PD[6]	D6	YES	PID Heater Control 2	PWM Servo Control			LOW	YES	
11	PD[7]	D7		Button 1 (Up)		Multiplexer		LOW	YES	
12	PB[0]	D8		Button 2 (Down)	Encoder Switch			LOW	YES	
13	PB[1]	D9	YES	PID Buzzer Control	PWM Buzzer Control			LOW	YES	
14	PB[2]	D10	YES	Shift Register, UI, Slave Select	PWM Peltier	ambient		-	YES	
15	PB[3]	D11/MOSI	YES	Shift Register, UI, MOSI	PWM Blower Control		PROGRAMMING (ISP)	-	YES	YES
16	PB[4]	D12/MISO		Relay, Contactor	Relay, Contactor		PROGRAMMING (ISP)	LOW	YES	YES
17	PB[5]	D13/SCK		Shift Register, UI, Clock	Blower Feedback	LCD control	PROGRAMMING (ISP)	-	YES	YES
18	AVCC	AVCC		Filtered Connection to VCC	Filtered Connection to VCC			-	-	
19	ADC6	A6		T_sense Chamber	T_sense Ambient	multiplexed		LOW	YES	
20	AREF	AREF		Unconnected	Unconnected			-	-	
21	GND	AGND		Ground	Ground		PROGRAMMING	-	-	
22	ADC7	A7		V_sense Heater1	T_sense Peltier Cold	I_sense LCD		-	YES	
23	PC[0]	A0		V_sense Heater2	T_sense HOT Chamber	I_sense Heater		-	YES	
24	PC[1]	A1		T_sense Heater	T_sense Heater	I_sense Peltier		-	YES	
25	PC[2]	A2		T_sense Heater Power	I_sense Heater + Peltier	I_sense Servo		-	YES	
26	PC[3]	A3		T_sense DC-in	I_sense Remainder	I_sense Motor		-	YES	
27	PC[4]	A4		Relay State Sense	LCD, SDA			-	YES	
28	PC[5]	A5		I_sense Heater Power	LCD, SCL			-	YES	
29	PC[6]/RESET	RESET		-			PROGRAMMING (ISP)	HIGH	YES	
30	PD[0]	D0/RX			IC_COMMUNICATION	LCD	PROGRAMMING (USB)	LOW	YES	
31	PD[1]	D1/TX			IC_COMMUNICATION	LCD	PROGRAMMING (USB)	LOW	YES	
32	PD[2]	D2		Button 3 (Center)	Encoder A			LOW	YES	

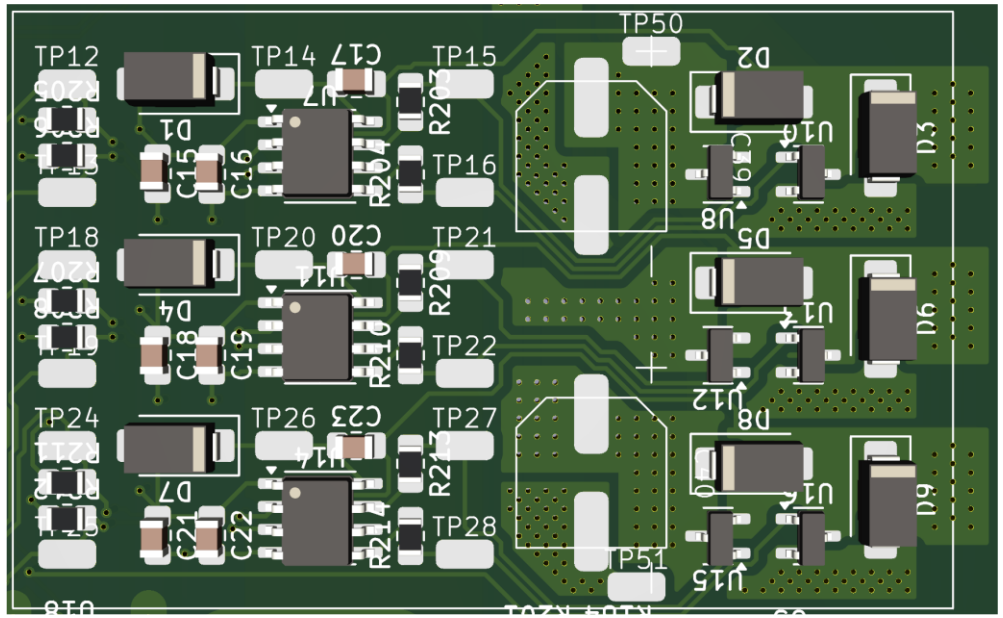
Motor controller

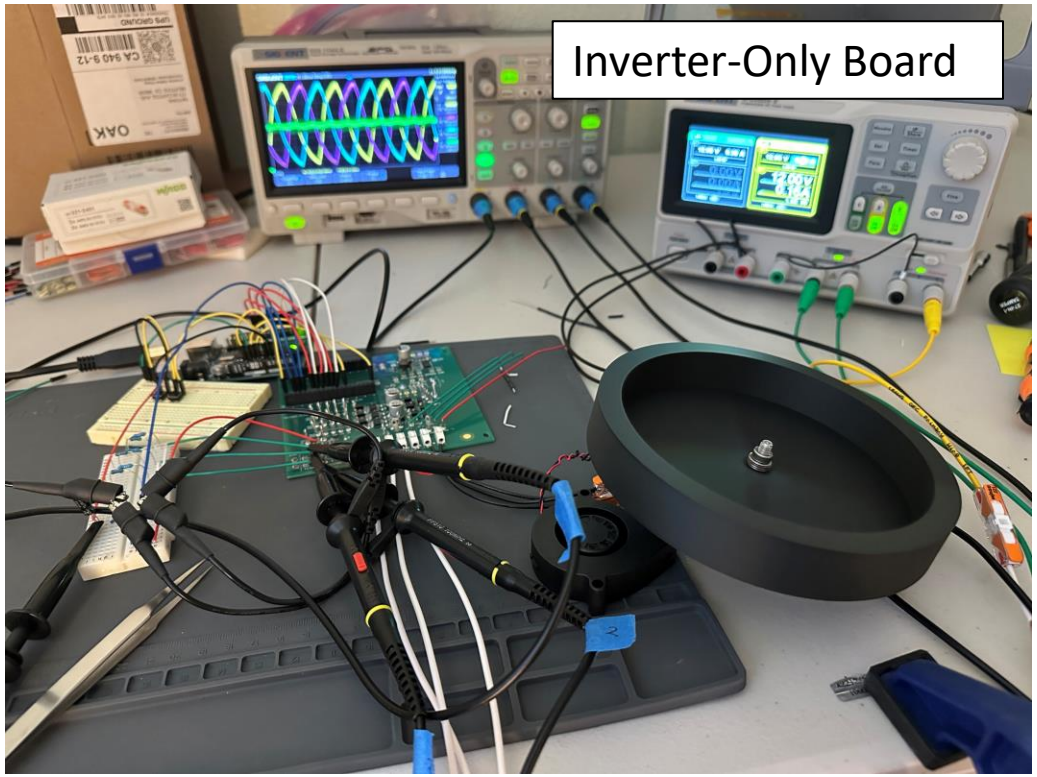


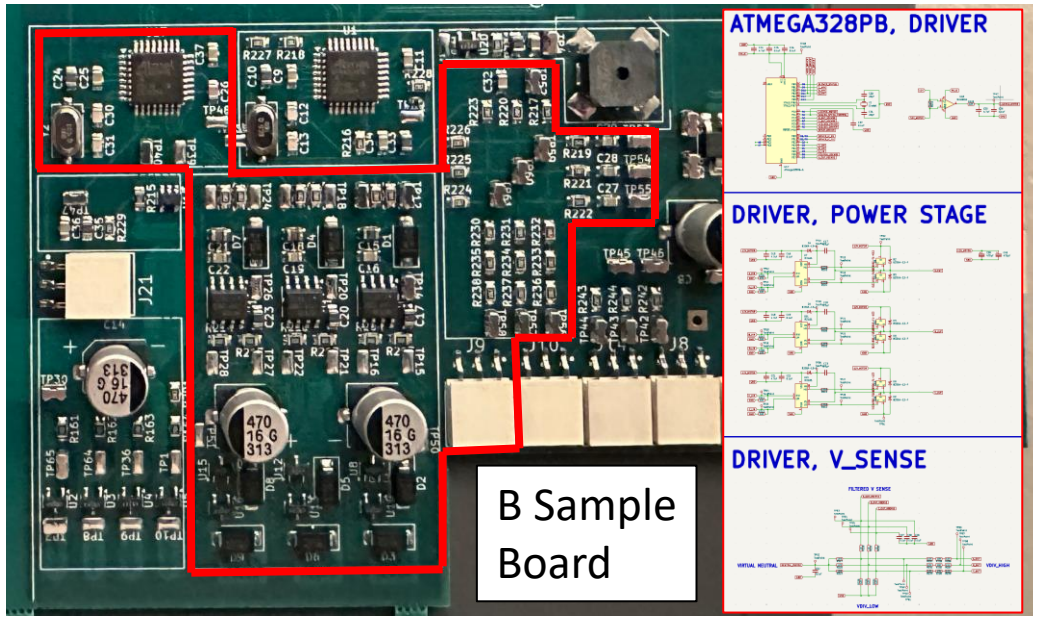
PIN NO	328P PAD	Arduino Designation	PWM Capable	Usage 1	Usage 2	Default State Pull up/down	Zener Protec
1	PD[3]	D3	YES	C_LIN			
2	PD[4]	D4		B_LIN			
3	GND	GND					
4	VCC	VCC			PROGRAMMING		
5	GND	GND			PROGRAMMING		
6	VCC	VCC					
7	PB[6]	XTAL1					
8	PB[7]	XTAL2					
9	PD[5]	D5	YES	A_LIN			
10	PD[6]	D6	YES	NEUTRAL_VSENSE			
11	PD[7]	D7		A_OUT_VSENSE			
12	PB[0]	D8		BLOWER_STATUS			
13	PB[1]	D9	YES	C_HIN			
14	PB[2]	D10	YES	B_HIN			
15	PB[3]	D11/MOSI	YES	A_HIN	PROGRAMMING (ISP)		
16	PB[4]	D12/MISO			PROGRAMMING (ISP)		
17	PB[5]	D13/SCK			PROGRAMMING (ISP)		
18	AVCC	AVCC					
19	ADC6	A6					
20	AREF	AREF					
21	GND	AGND			PROGRAMMING		
22	ADC7	A7					
23	PC[0]	A0		I_SENSE_MOTOR			
24	PC[1]	A1		ANALOG_SPEED_CONTROL			
25	PC[2]	A2		B_OUT_VSENSE			
26	PC[3]	A3		C_OUT_VSENSE			
27	PC[4]	A4		LCD, SDA			
28	PC[5]	A5		LCD, SCL			
29	PC[6]/RESET	RESET			PROGRAMMING (ISP)		
30	PD[0]	D0/RX		IC_COMMUNICATION	PROGRAMMING (USB)		
31	PD[1]	D1/TX		IC_COMMUNICATION	PROGRAMMING (USB)		
32	PD[2]	D2					

Inverter 1/2

- Schematic is based on some work by Electronoobs (a youtuber)
- Adds 2 unique parts to BOM, diodes and gate driver ICs
- Sensorless, no position sensing.
- Commutation is timed by detecting zero-crossing of the phase voltages relative to virtual neutral node.
- Requires a “dumb” start up sequence to build up rotor inertia, before activating the sensorless commutation loop.
- Stopped-rotor detection is… work in progress

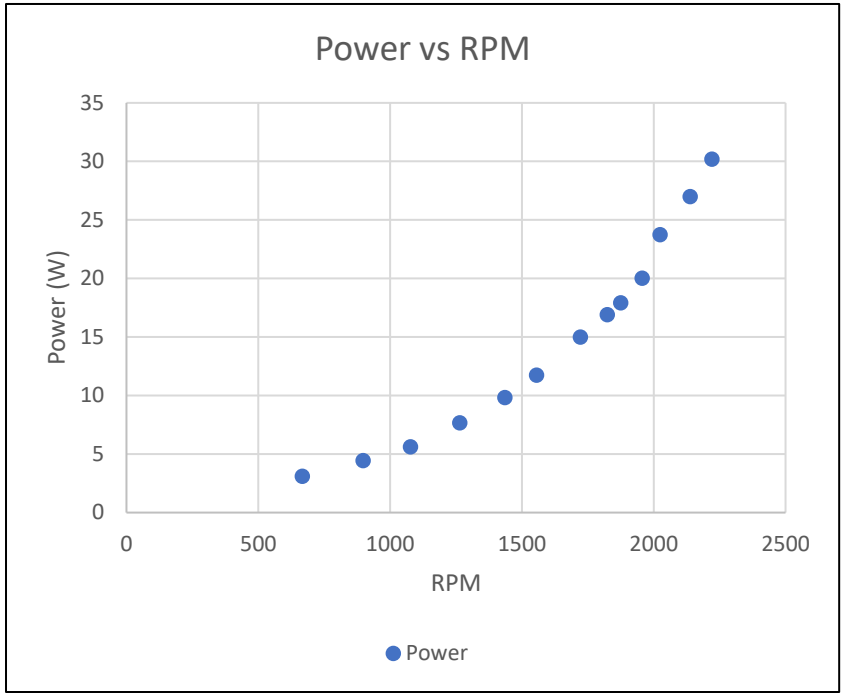






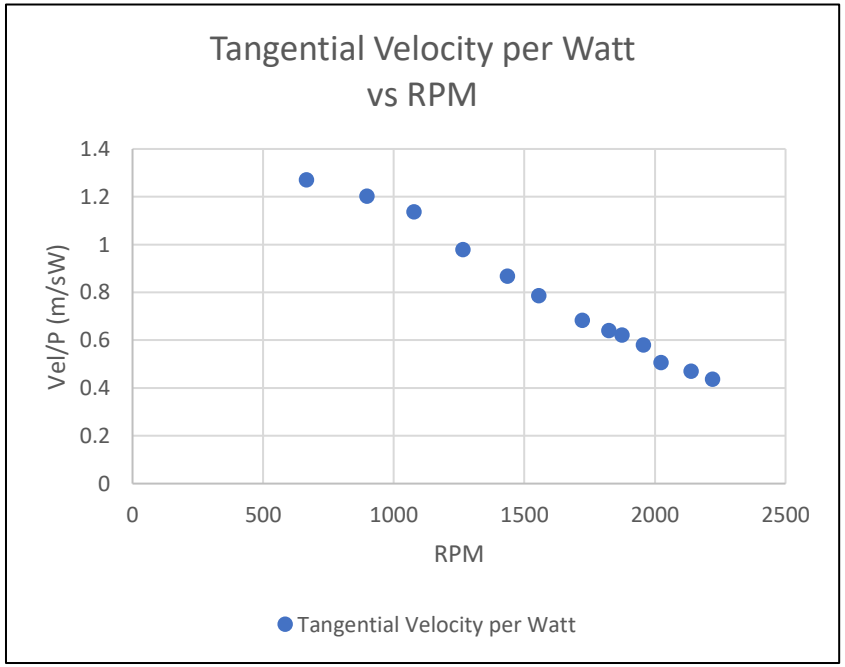
Inverter 2/2

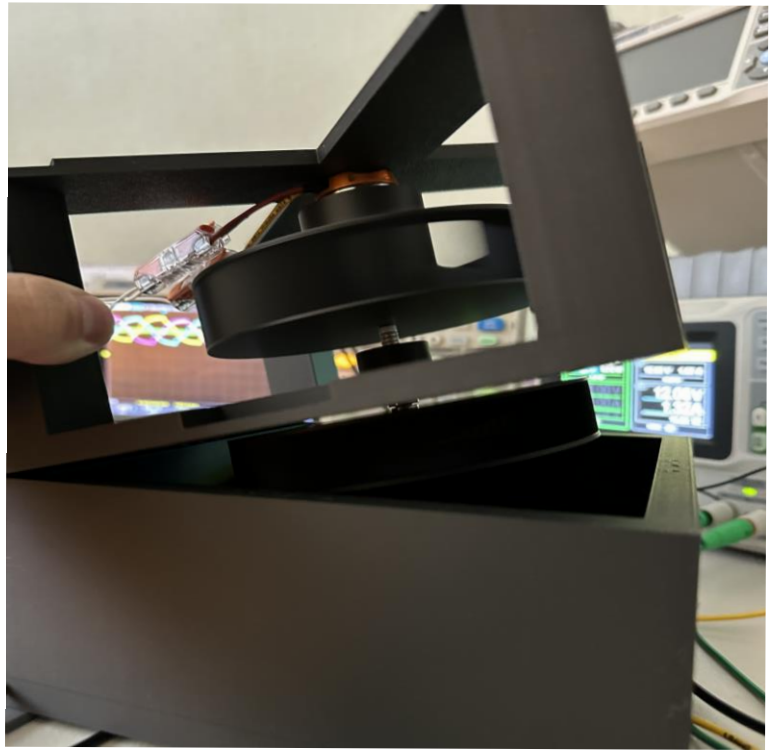
- The good:
  - can do 2k RPM with 2 blowers
  - Plenty of air velocity to asymptote the Rth of the selected heatsink.
- The bad:
  - Commutates loudly (square wave).
  - FETs would burn out whenever control IC experienced voltage fluctuations
  - Difficult to detect stuck-rotor (inverter would still commutate even if the rotor was halted), electrical signature for such an event was not clear.

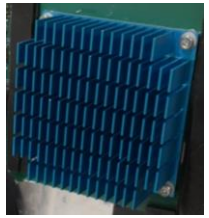


Blower System RPM-Power Sweep

RPM	Tangential Velocity	Power
rpm	m/s	*W*
667	4.0	3.1
898	5.3	4.4
1078	6.4	5.6
1265	7.5	7.7
1436	8.5	9.8
1556	9.3	11.8
1722	10.2	15.0
1824	10.9	16.9
1875	11.2	17.9
1957	11.6	20.0
2024	12.0	23.8
2139	12.7	27.0
2221	13.2	30.2





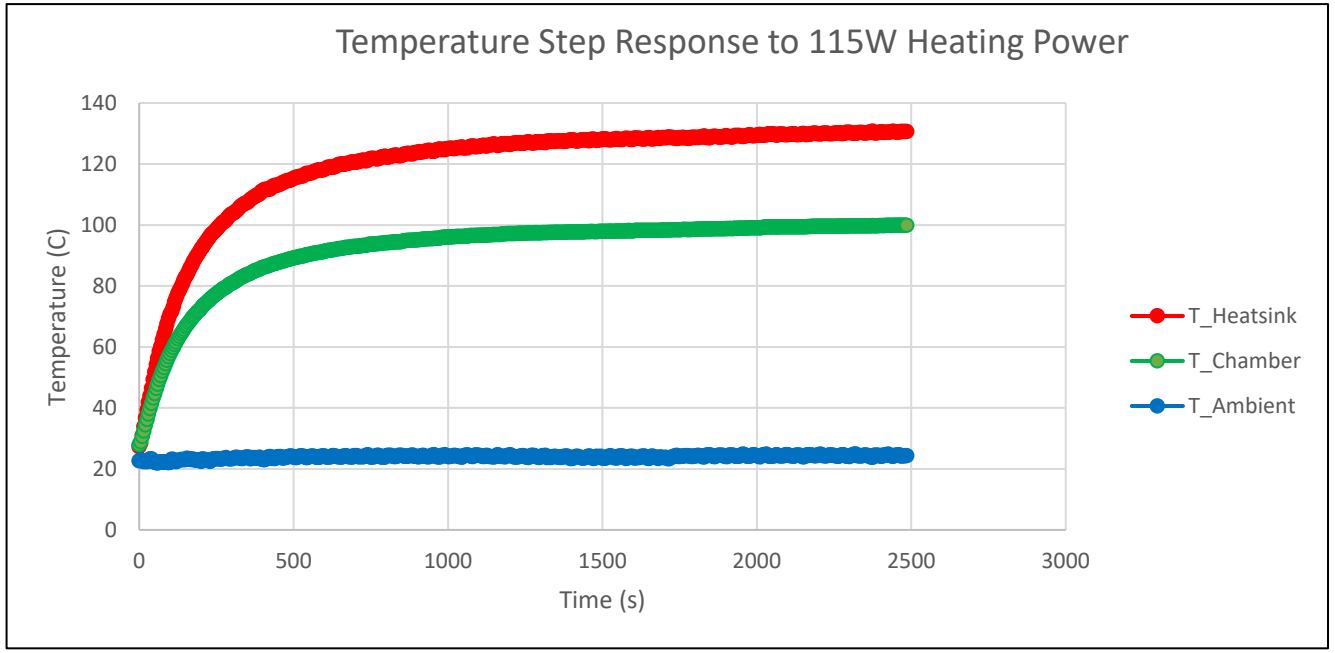


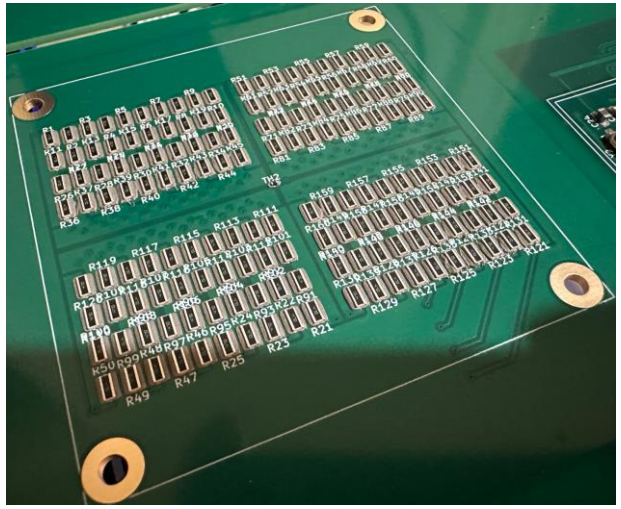
Heatsink Specs

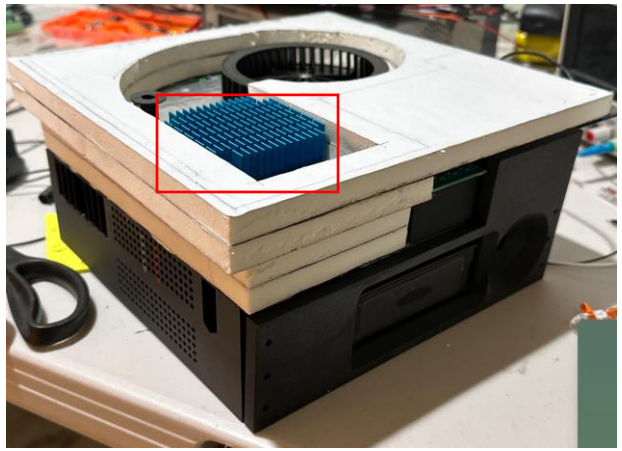
AIR VELOCITY - LFM (m/s)		100 (0.5)	200 (1.0)	300 (1.5)	400 (2.0)	500 (2.5)	600 (3.0)	700 (3.5)
Thermal Resistance °C/W	Unducted Flow	2.20	1.10	0.90	0.70	0.70	0.60	0.60
	Ducted Flow	1.10	0.80	0.70	0.60	0.50	0.50	0.50

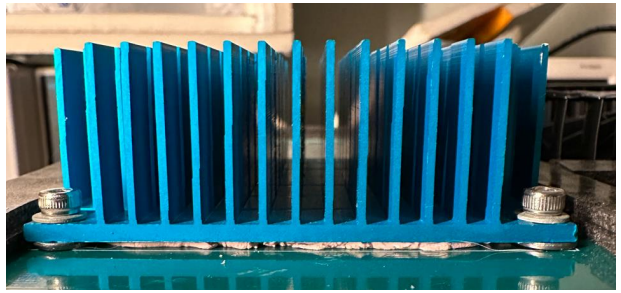
Heater

- Construction
  - Forged, cross-cut Aluminum heatsink
  - Parker Therm-a-gap Gel30
  - 160x wide-terminal chip resistors, arranged 2S-80P
    - Wider terminal chips are more tolerant to TIM voids
- Performance
  - With no thermal load, chamber time constant = 195s
  - With 0.67 K/W enclosure, it takes 115W to reach 100C from ~20C ambient



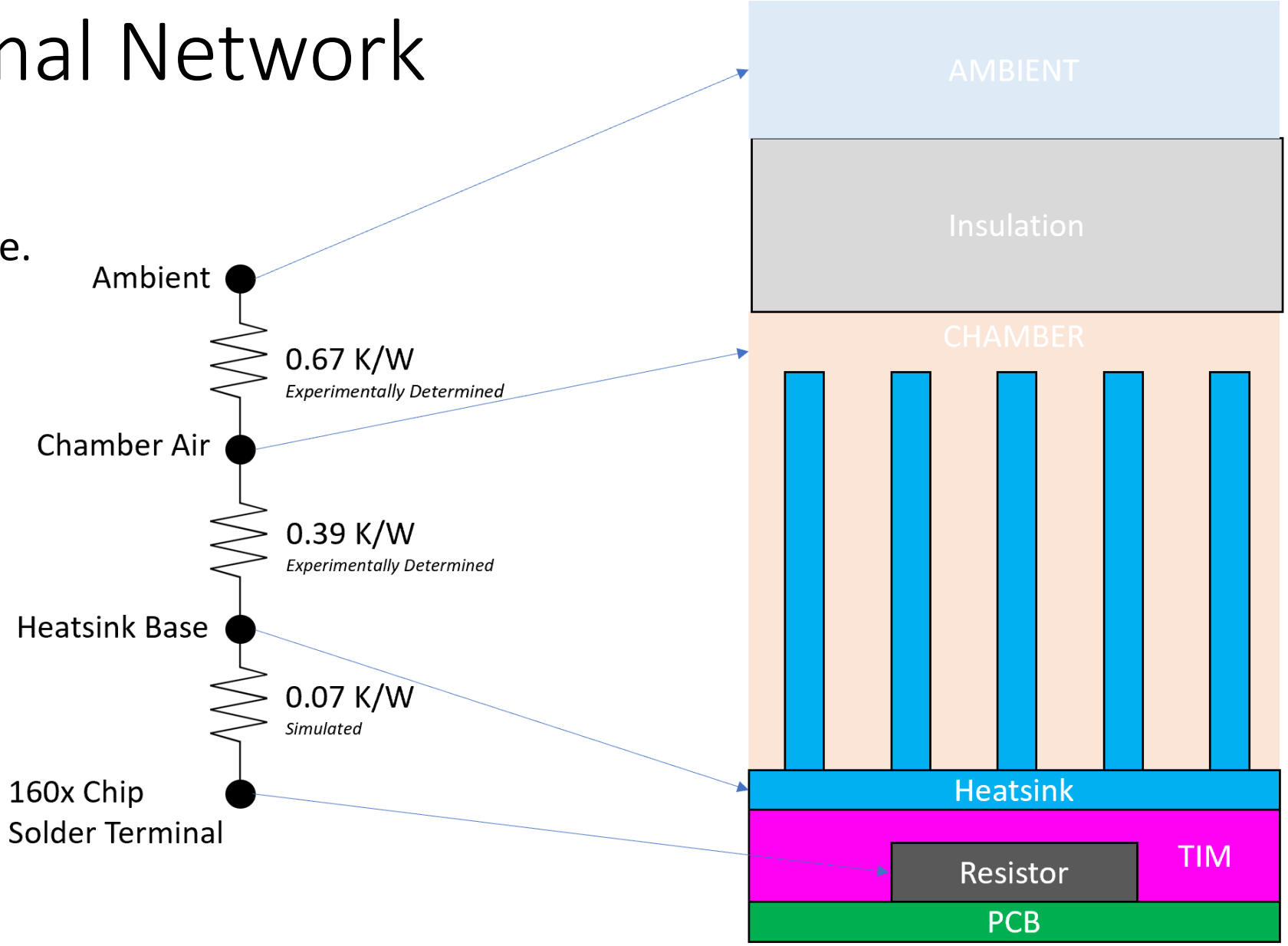






Heater Thermal Network

- Heater chip resistor reliability is questionable.
- Simulated thermals to estimate chip temperature
- @100C chamber temp, chips are expected to reach **TBD**



Thermal simulation

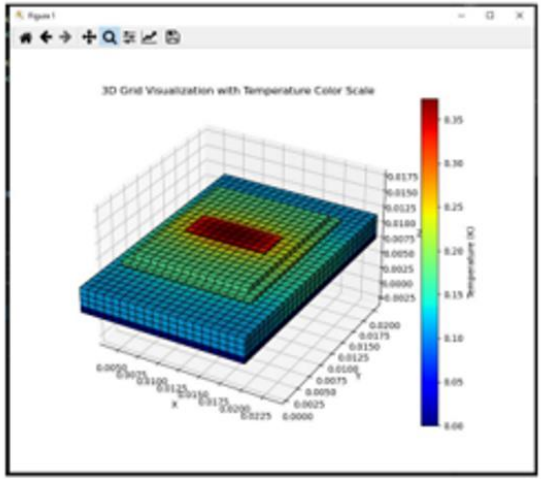
- I write my own thermal simulation scripts. Why? Because…
  - It's quick to set up
  - It costs no money
  - Niche simulation features are easy program in

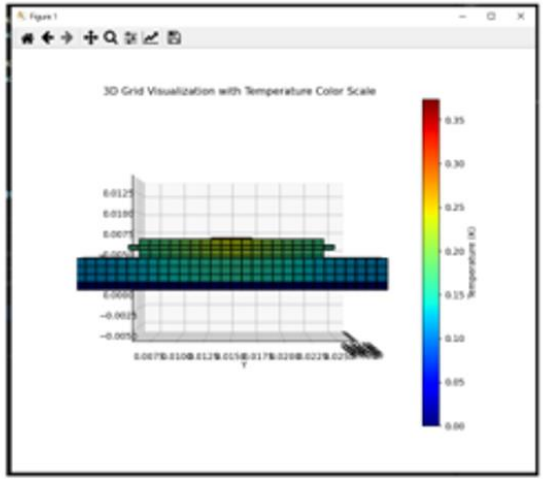
- Github
  - github.com/nvchung599/ThermalStackFEA
    - C++
    - Runs *fast*
    - 3D modelling limited to XY symmetrical block stacks only
    - Good for steady-state solutions
  - github.com/nvchung599/ThermalStackFEA2
    - Python
    - Runs a bit slower
    - Full 3D-modelling freedom, with color heatmap 3D visualizer
    - Great for design-space sweeps
    - Great for transient problems
      - Features P(t) and T(t) control
      - T_avg, T_max, and P observers

TSFEA1 Example Outputs

```
 t = 0 seconds          T_avg = 40.000 C
 t = 1.000 seconds      T_avg = 107.736 C      <- @ one time constant
 t = 6.700 seconds      T_avg = 137.064 C      <- @ steady state

Converged on the following solution after 21 minutes and 32 seconds

                                        Matl        T_avg      T_var      Q_gen      Vol

   Block 0     --------------------     Water       40 C       0 C        0 W        113 mm^3
   Block 1     --------------------     Aluminum    47 C       8 C        0 W        675 mm^3
   Block 2        --------------        TIM Pad     64 C       5 C        0 W        50 mm^3
   Block 3        --------------        Copper      78 C       10 C       0 W        200 mm^3
   Block 4            ------            Silicon     87 C       4 C        100 W      25 mm^3
   Block 5        --------------        Copper      78 C       10 C       0 W        200 mm^3
   Block 6        --------------        TIM Pad     64 C       5 C        0 W        50 mm^3
   Block 7     --------------------     Aluminum    47 C       8 C        0 W        675 mm^3
   Block 8     --------------------     Water       40 C       0 C        0 W        113 mm^3

Thermal impedance, heat source to infinite heatsink = 0.474 K/W
```

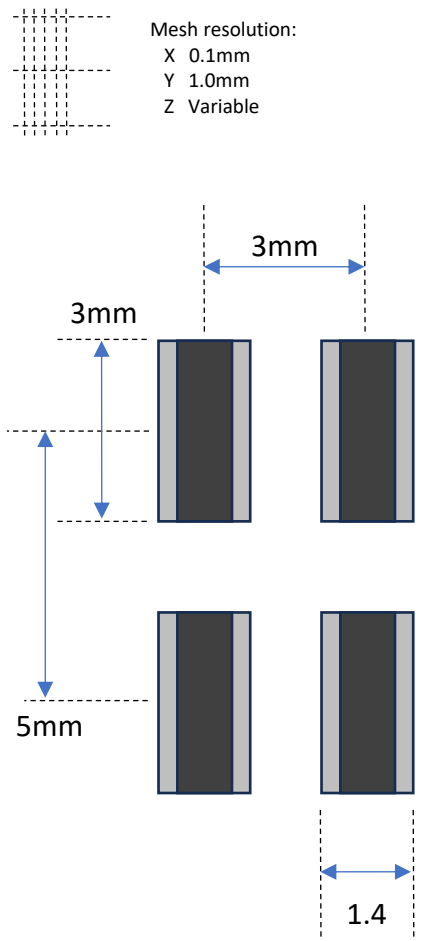
TSFEA2 Example Outputs



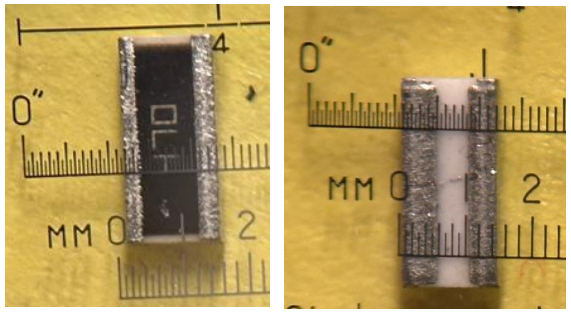


Simulation Parameters

Chip Layout



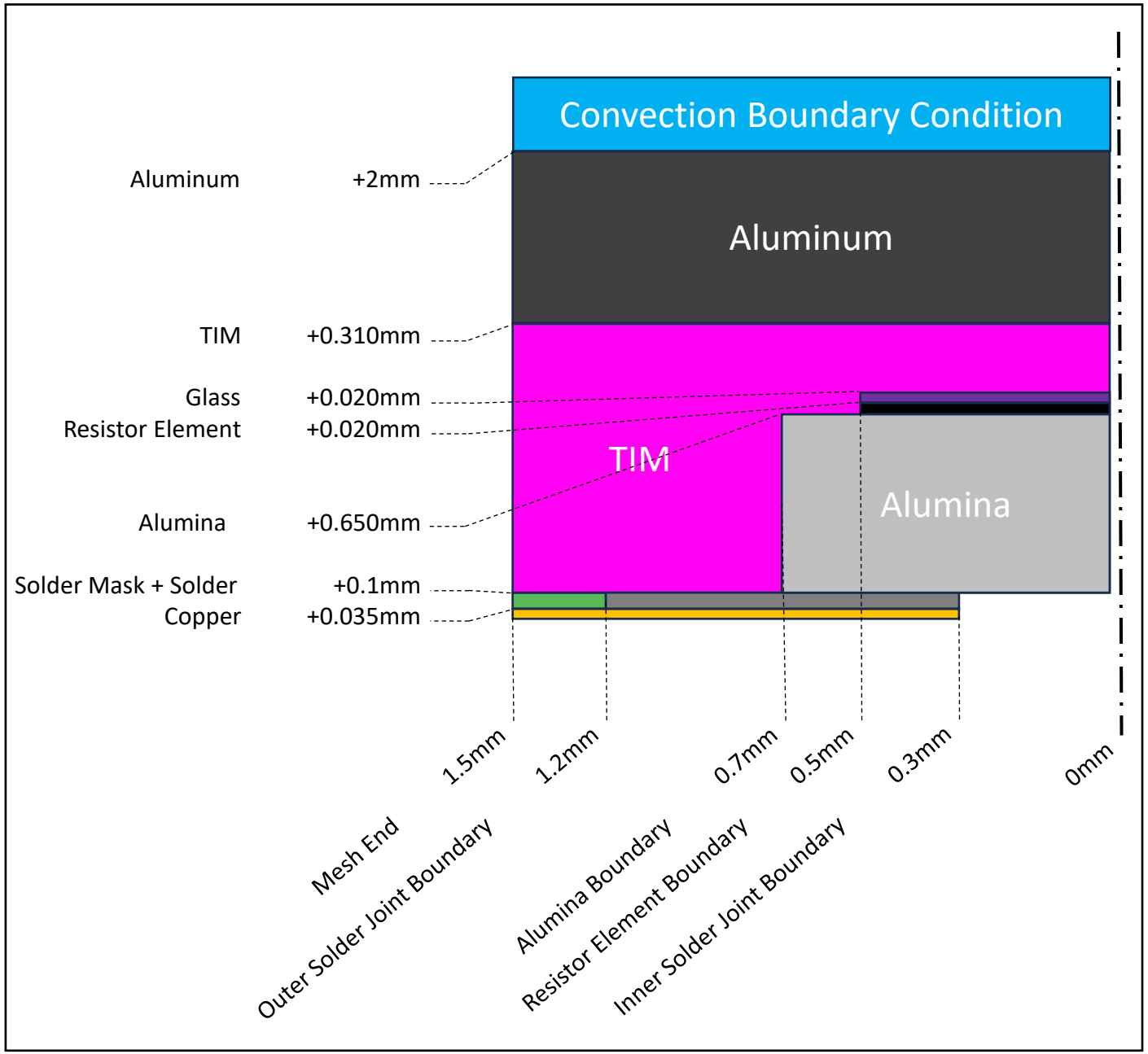
Chip Dimensions



Material Data

Name	Thermal Conductivity (W/mK)	Volumetric Thermal Capacity (J/m^3K)
Aluminum	200	2.43 E6
TIM	3.5	2.70 E6
Glass	1	1.87 E6
Resistor Element	3	2.24 E6
Alumina	30	3.60 E6
Solder	50	1.71 E6
Solder Mask	0.35	1.69 E6
Copper	400	3.08 E6

Dimensions used in simulation mesh

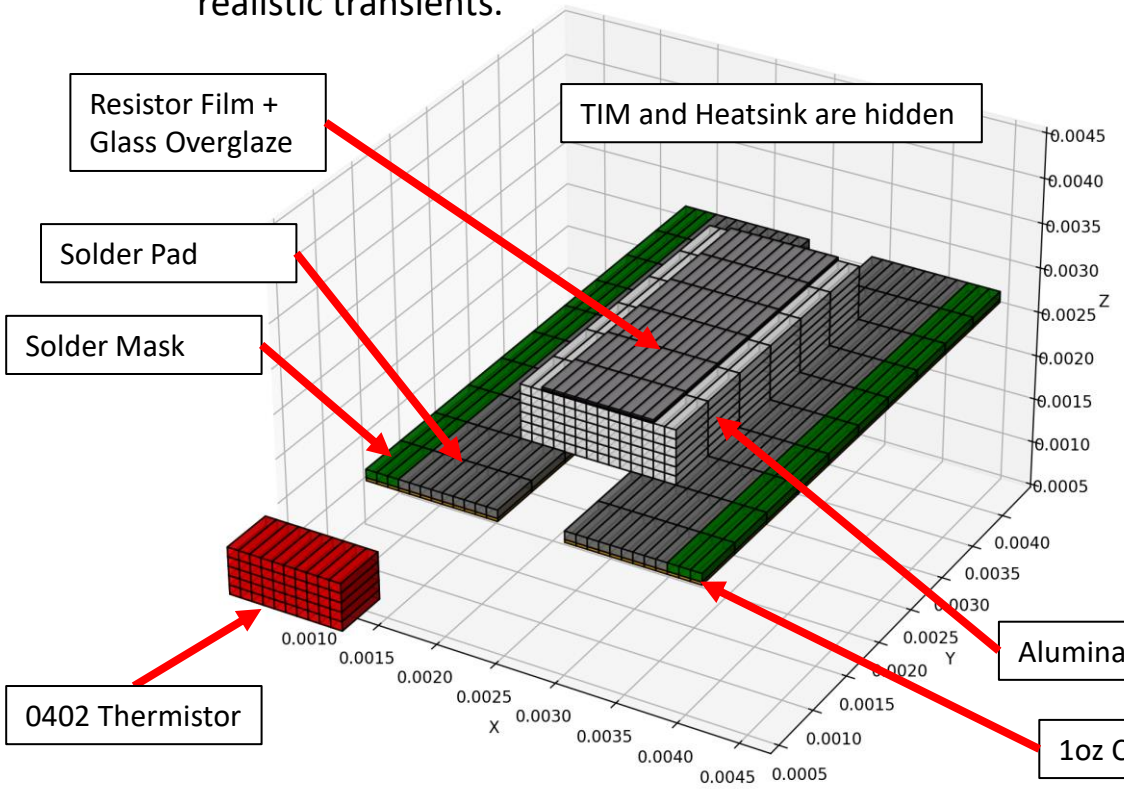


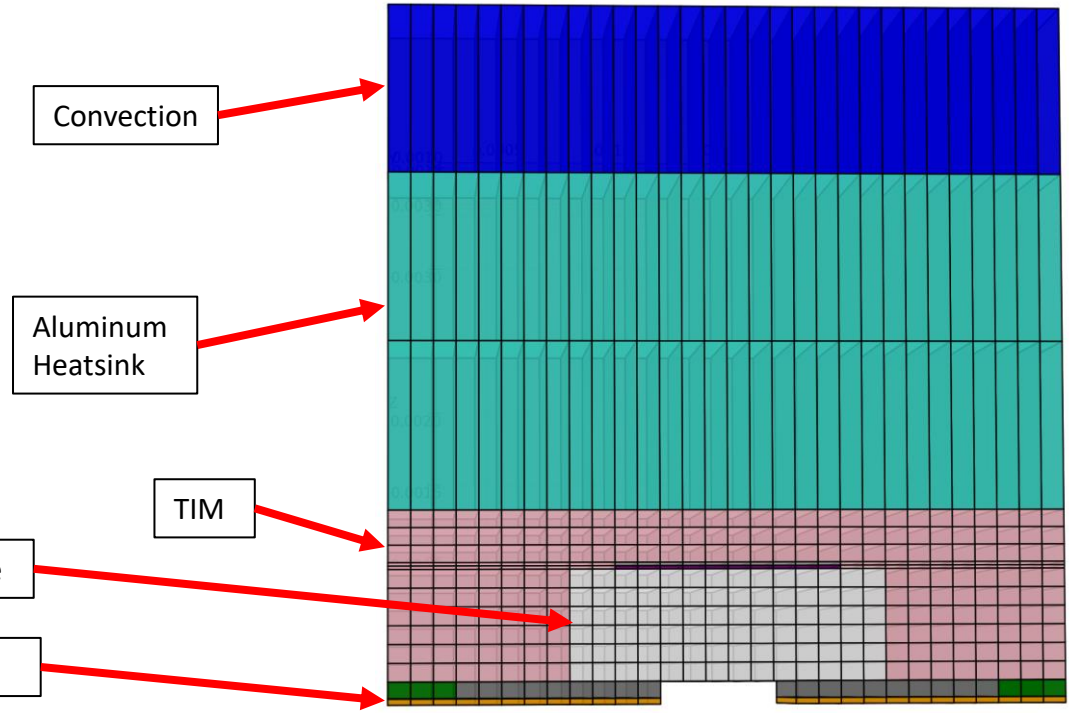
Simulation Setup

- We model an approximation of the chip using primitive geometry-creation functions
- A temperature sensor mass is included, so we can study the thermal offsets and transient lag that we expect in our real-world 0402 thermistor
- Notice that element thickness are variable (20 microns to 1mm). This allows us to run the simulation fast, while still producing realistic transients.

Geometry-shaping with primitive CAD functions

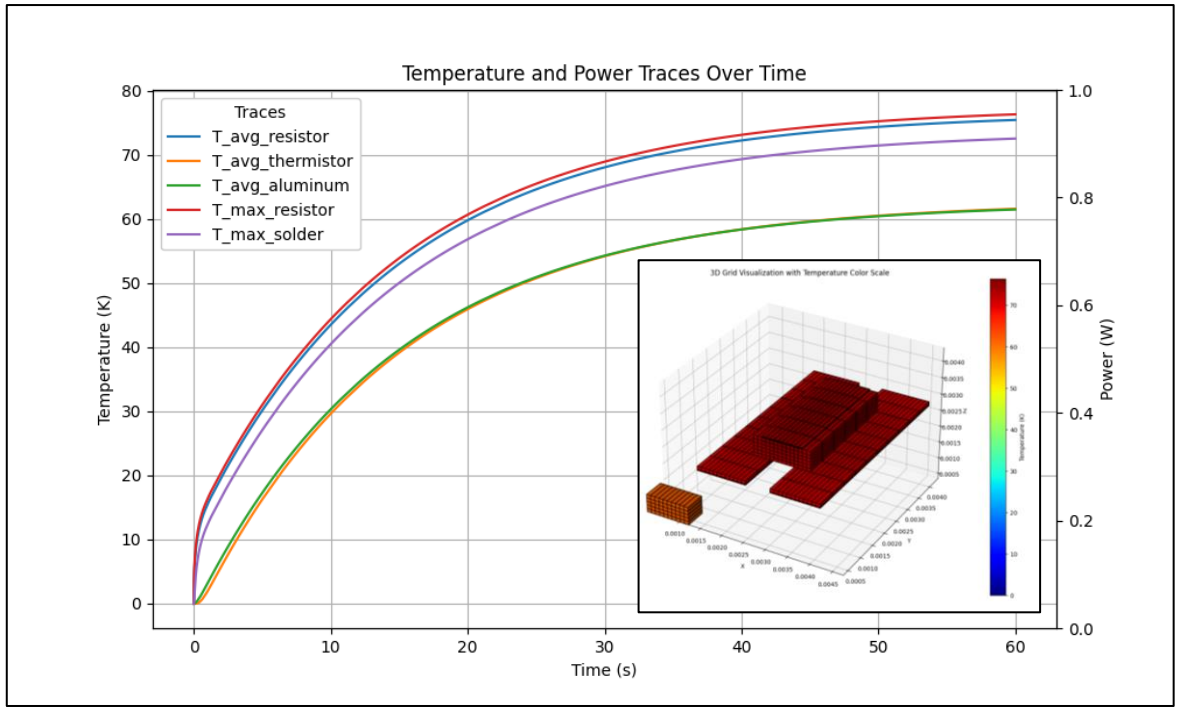
```
my_model = ThermalModel()
# copper
my_model.create_volume(x_vol=0.003,y_vol=0.010,z_vol=0.000035,x_el=0.0001,y_el=0.0005,z_el=0.000035,properties=null,T=0,P=0)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=30,y_ext=10,z_ext=1,properties=copper,T=0,P=0)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=6,y_ext=10,z_ext=1,properties=null,T=0,P=0)

# solder
my_model.extend_volume(axis="z",polarity=1,size_extrusion=0.0001,thickness_element=0.0001,properties=null)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=30,y_ext=10,z_ext=1,properties=mask,T=0,P=0)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=24,y_ext=10,z_ext=1,properties=solder,T=0,P=0)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=6,y_ext=10,z_ext=1,properties=null,T=0,P=0)

# Alumina
my_model.extend_volume(axis="z",polarity=1,size_extrusion=0.00066,thickness_element=0.00011,properties=tim)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=14,y_ext=6,z_ext=6,properties=alumina,T=0,P=0)
my_model.modify_volume(x_coord=1,y_coord=1,z_coord=2,x_ext=10,y_ext=1,z_ext=5,properties=thermistor,T=0,P=0)

# resistor
my_model.extend_volume(axis="z",polarity=1,size_extrusion=0.00002,thickness_element=0.00002,properties=tim)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=10,y_ext=6,z_ext=1,properties=resistor,T=0,P=1)

# glass
my_model.extend_volume(axis="z",polarity=1,size_extrusion=0.00002,thickness_element=0.00002,properties=tim)
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=10,y_ext=6,z_ext=1,properties=glass,T=0,P=0)

# tim, remainder
my_model.extend_volume(axis="z",polarity=1,size_extrusion=0.000309,thickness_element=0.000103,properties=tim)

# void
my_model.modify_volume_centered_slice(axis="z",polarity=-1,index_start_extrusion=-1,x_ext=10,y_ext=6,z_ext=3,properties=null,T=0,P=0)

# aluminum
my_model.extend_volume(axis="z",polarity=1,size_extrusion=0.002,thickness_element=0.001,properties=aluminum)

# convection
my_model.extend_volume(axis="z",polarity=1,size_extrusion=0.001,thickness_element=0.001,properties=reservoir)
```





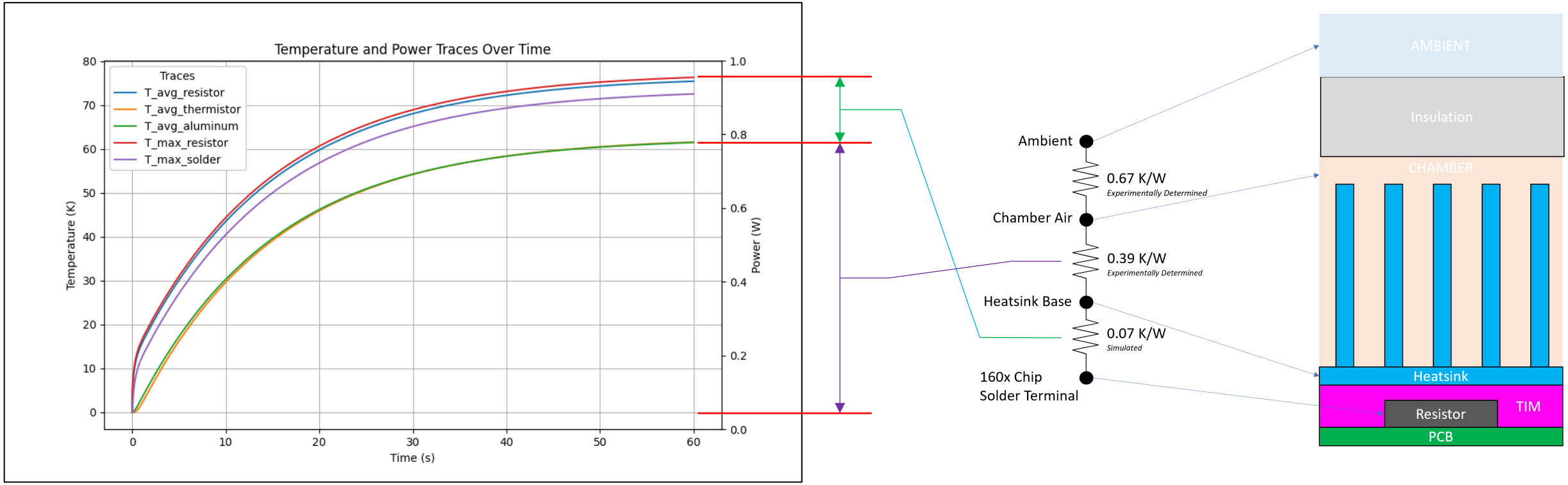
Simulation Results, Steady State 1/2

- Potential proxy measurements for product reliability (depends on which drives failures first… we don't know yet)
  - Rth calculated from Solder T_max
  - Rth calculated from Resistor Film T_max
  - Rth calculated from Resistor Film T_avg
- The alumina substrate of this chip (KOA Speer WK73R2BTTD75R0F), is relatively thick. The high power capability of this chip is owed to the alumina substrate thickness.
- For our application, the majority of heat flux goes through the footprint of the resistor film, directly upwards, where the TIM bondline is thinnest.
- The thicker aluminum substrate does two things for us, thermally:
  - Alumina thickness is a fine tuning knob for TIM bondline thickness, since we shim the heatsink at discrete offsets using common washers. As thickness of the alumina substrate increases, TIM bondline decreases, and Rth of the pure Z conduction path proportionally decreases.
  - Alumina serves as a heat spreader/absorber, smoothing out hot spots if we encounter air voids directly above the resistor film.
- Thermistor and heatsink are very well coupled. Temperatures align almost exactly with virtually no lag.



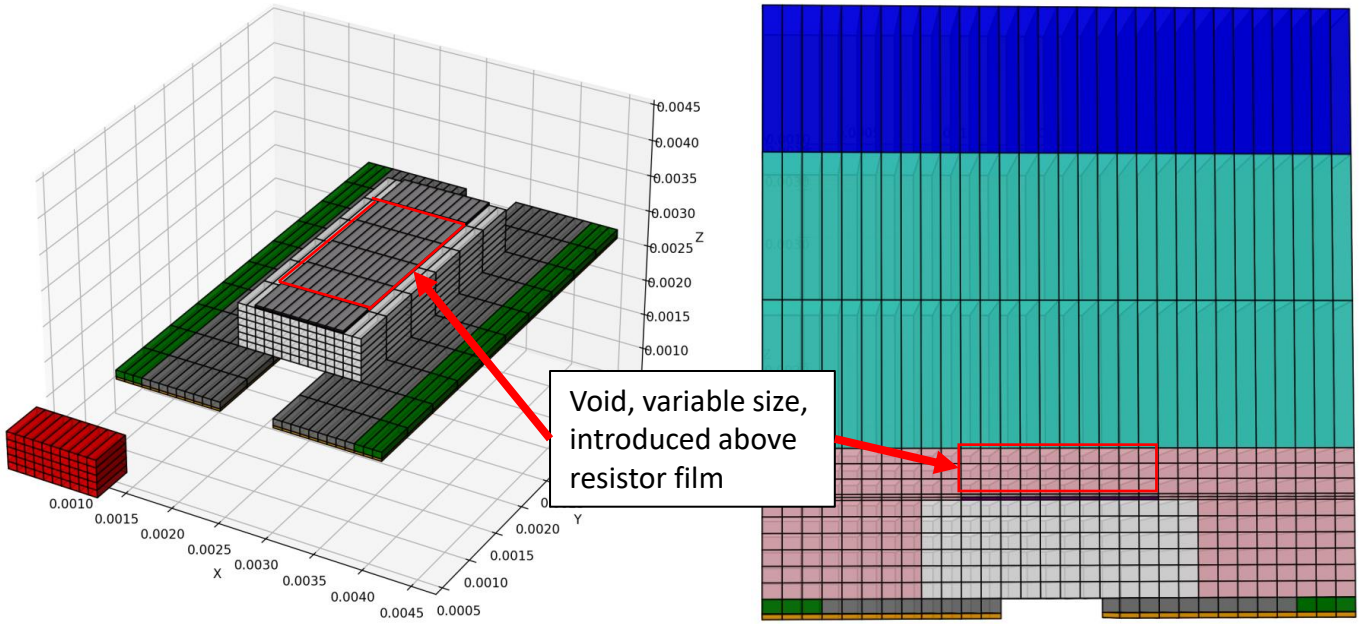
Item	Value	Unit
Power	1	Watts
dT_max_solder	72.6	Kelvin
dT_max_resistor_film	76.3	Kelvin
dT Scaling Factor, Thermistor/Solder	0.85	-
t_convergence	>60	Seconds
Time constant	12	Seconds
Rth_solder-ambient, 1x device	72.6	K/W
Rth_solder-ambient, 160x device	0.45	K/W
Rth_solder-heatsinkbase, 1x device	11.1	K/W
Rth_solder-heatsinkbase, 160x device	0.07	K/W

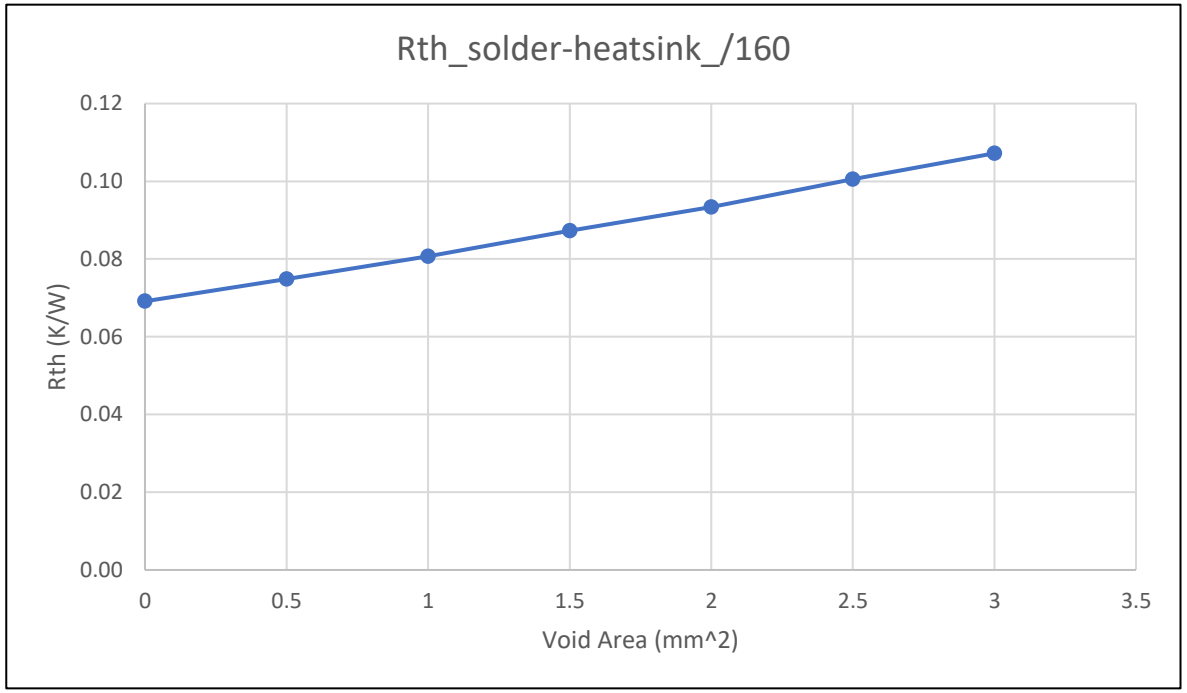
Simulation Results, Steady State 2/2



Simulation Results, TIM Voids, 1/2

- Realistically, we must accept that some distribution of void size and prevalence will be encountered in a mass-produced product. The design must tolerate some percentile of void severity.
- We de-risk voids by simulating air pockets of variable size, feeling out Rth sensitivity.
- A void the size of the resistor element will
  - Increase Rth_solder-heatsink by 55%
  - Increase Rth_solder-ambient by 8%
- As it turns out, given our arrangement of TIM, solder pads, heatsink, and 160x resistors sharing the power requirement... We're pretty safe from reasonably sized voids.



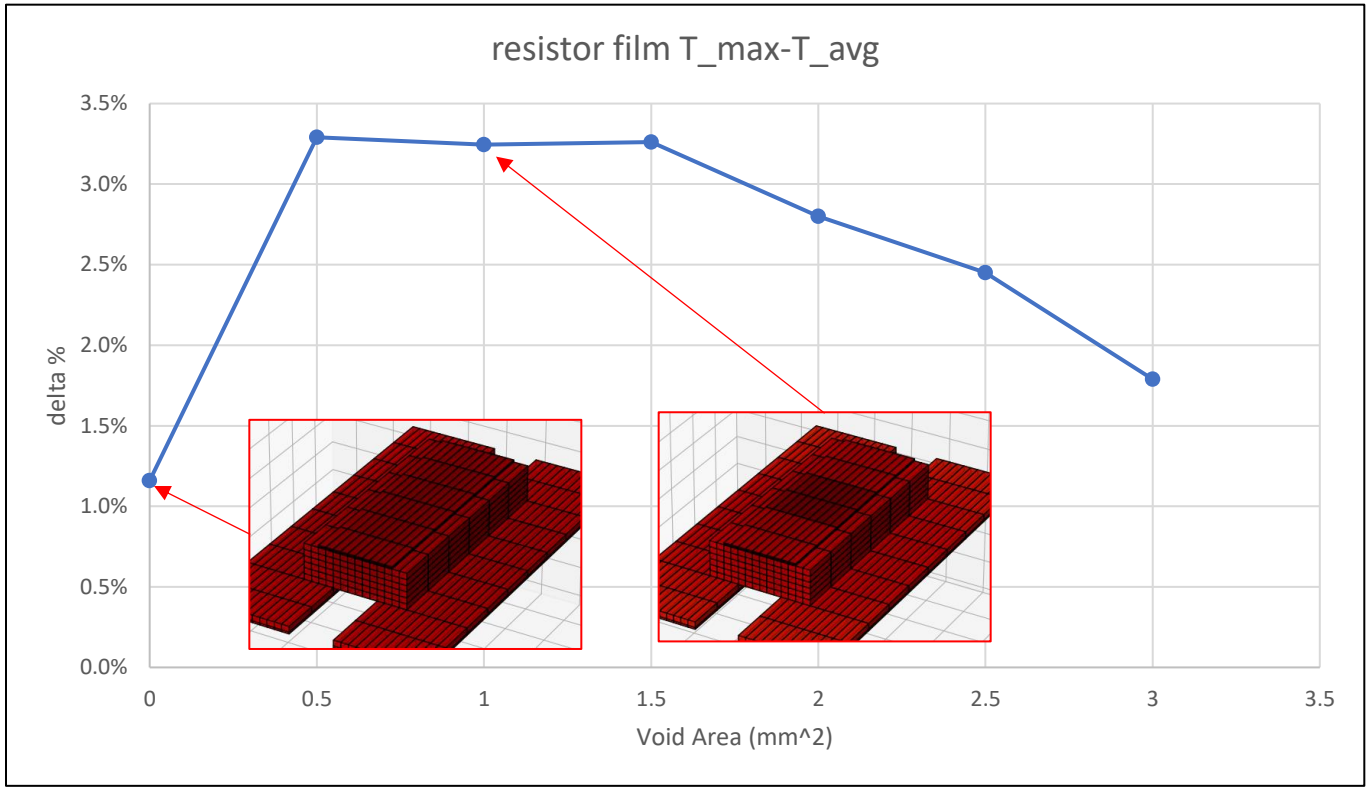


Simulation Results, TIM Voids, 2/2

- What about localized hot spots on the resistor film? Will it burn up?

- No.

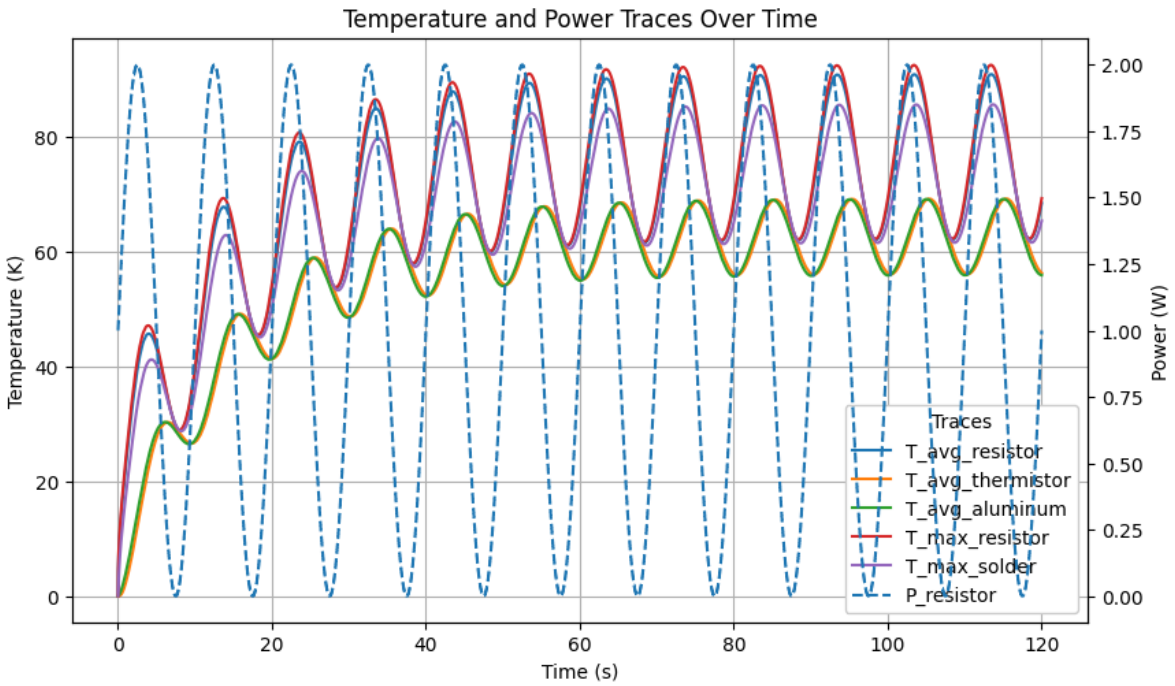
- The alumina substrate is an excellent heat-absorber and temperature smoother.



Void X	Void Y	Void Area	resistor film T_avg	thermistor T_avg	Aluminum T_avg	resistor film T_max	Solder T_max	Rth_SLD-HS / 160	Rth_SLD-AMB / 160	resistor film T_max-T_avg
mm	*mm*	*mm^2*	*K*	*K*	*K*	*K*	*K*	*K/W*	*K/W*	*K*
0	0	0	75.5	61.6	61.5	76.3	72.6	0.07	0.45	1.2%
1	0.5	0.5	76.5	61.6	61.5	79.0	73.5	0.07	0.46	3.3%
1	1	1	77.6	61.7	61.5	80.2	74.4	0.08	0.47	3.2%
1	1.5	1.5	78.9	61.7	61.5	81.5	75.5	0.09	0.47	3.3%
1	2	2	80.2	61.7	61.5	82.5	76.5	0.09	0.48	2.8%
1	2.5	2.5	81.7	61.8	61.5	83.7	77.6	0.10	0.49	2.5%
1	3	3	83.3	61.8	61.5	84.8	78.7	0.11	0.49	1.8%

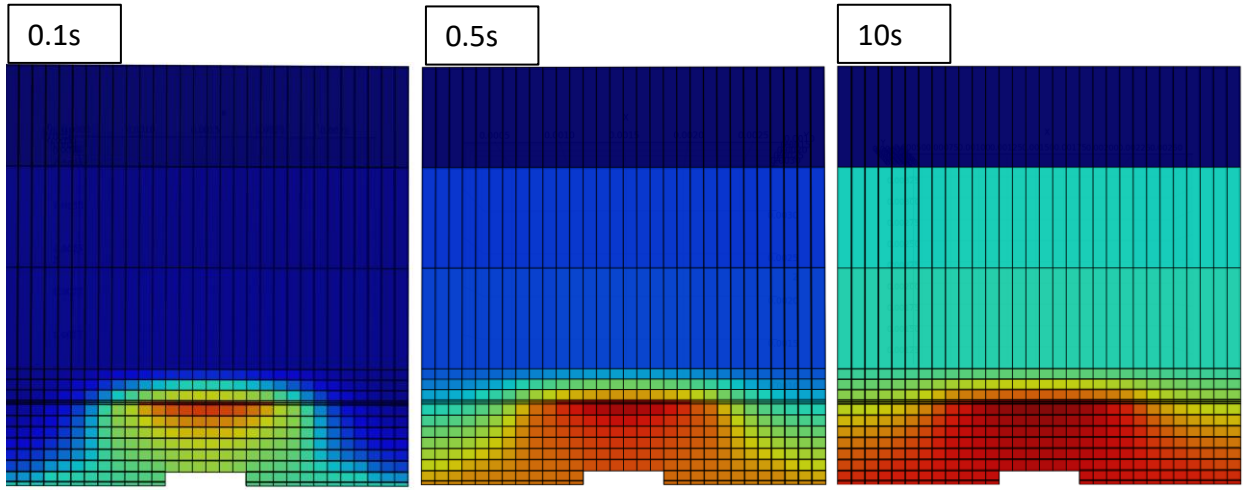
Simulation Results, Transients

- A sinusoidal power input helps us visualize the “lag” between the thermocouple and resistor chip.
- This lag seems to be acceptably low (1.5 seconds phase shift). So long as we are able to detect impending overtemperature (perhaps due to blower fan failure) before a critical failure temperature is reached.
- If we so choose, lag and offset can effectively be reduced by implementing a temperature observer algorithm that runs on the IC
  - $\int (T_{thermistor}, T_{ambient}, P_{heat}) = T_{resistor}$



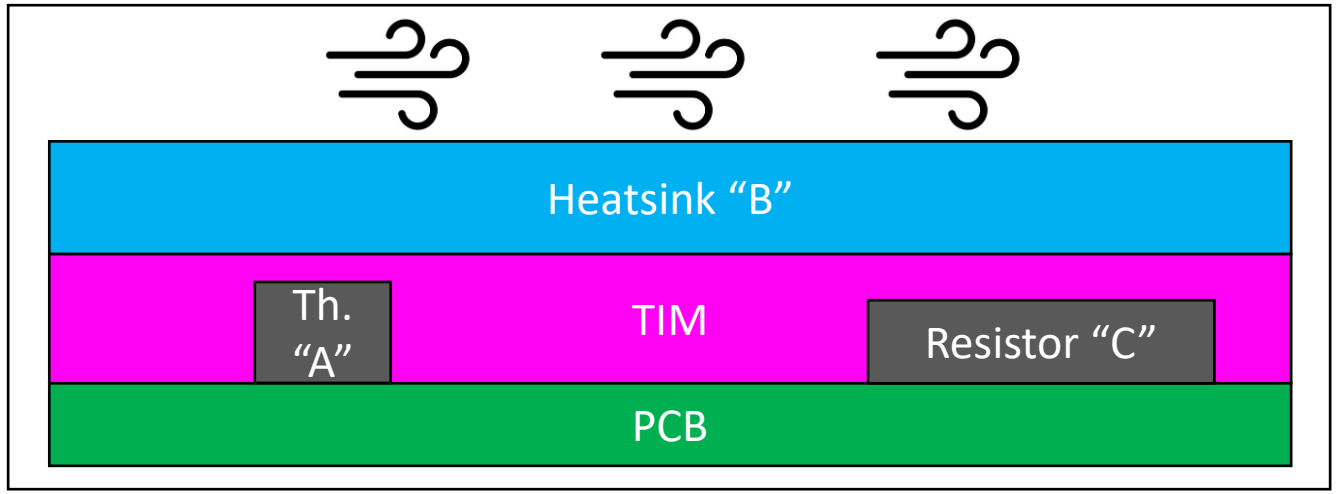
Visualizing the heat-up of a resistor chip. Notice how homogenous heatsink temperature is.

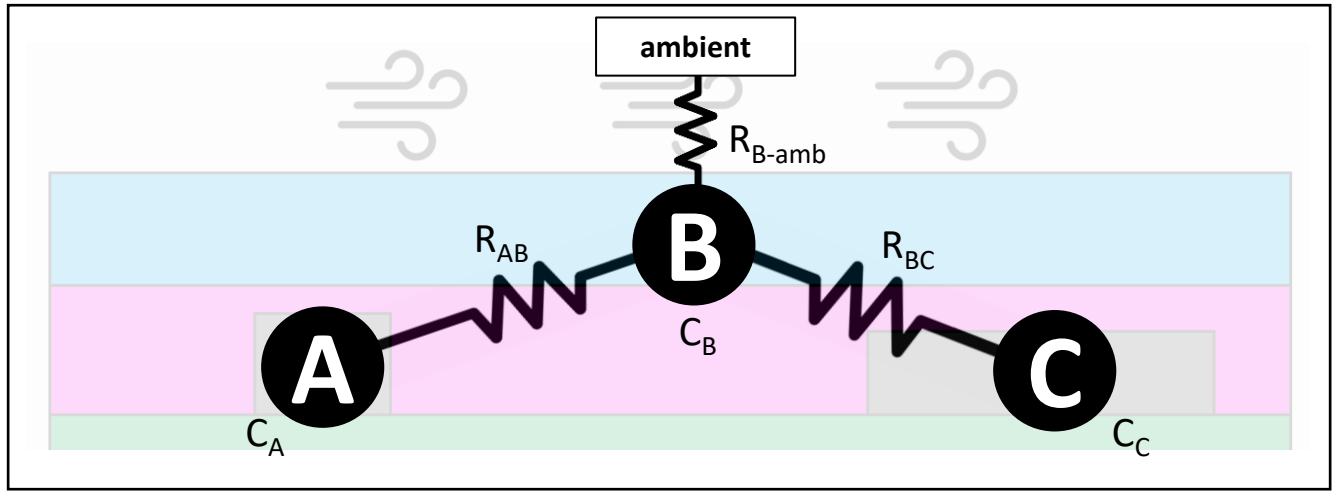
At steady state, the most aggressive temperature gradients are: within the TIM and between heatsink-ambient

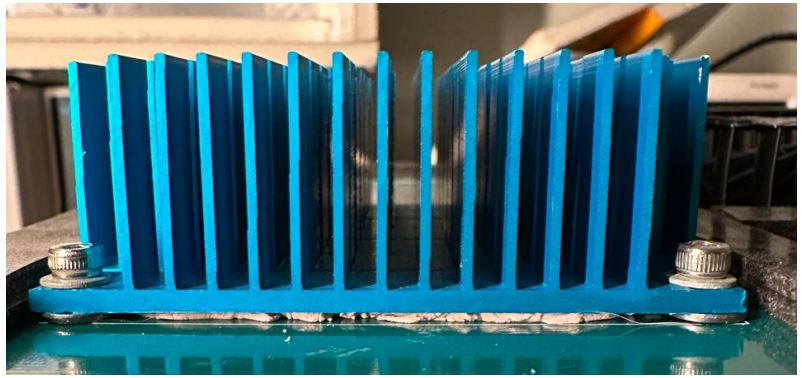


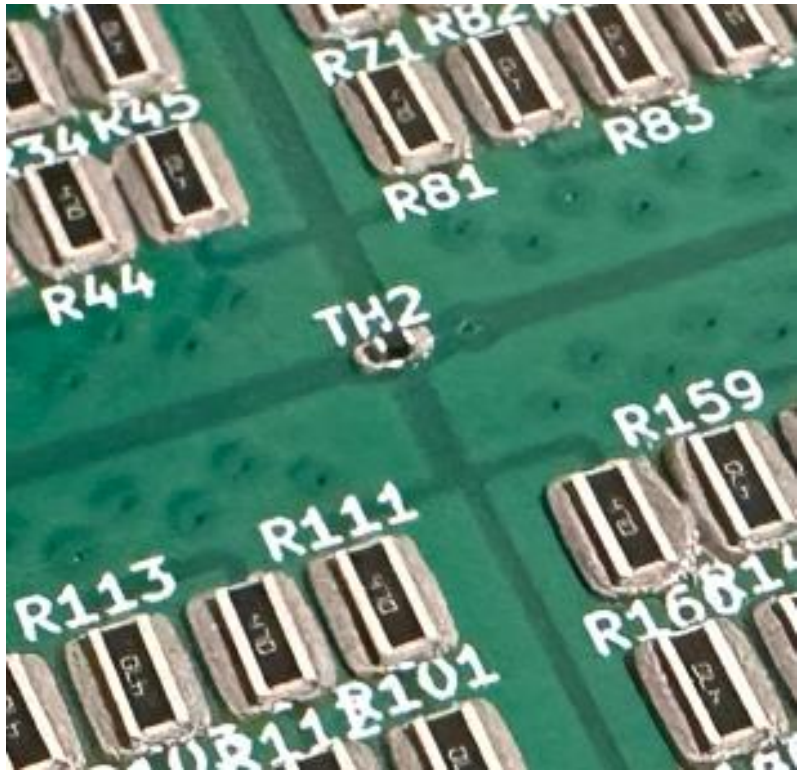
Temperature Observer Implementation 1/10
Simplified, Lumped Parameter Thermal Model

- TIM capacitance is a large contributor to lag, but it is not explicitly represented here. Instead we lump it into the things we actually care about and measured in simulation: thermistor and resistor
- The thermal path directly between thermistor and resistor (call it R_{AC}) is negligible, as suggested by our simulations, so we exclude it from this model.



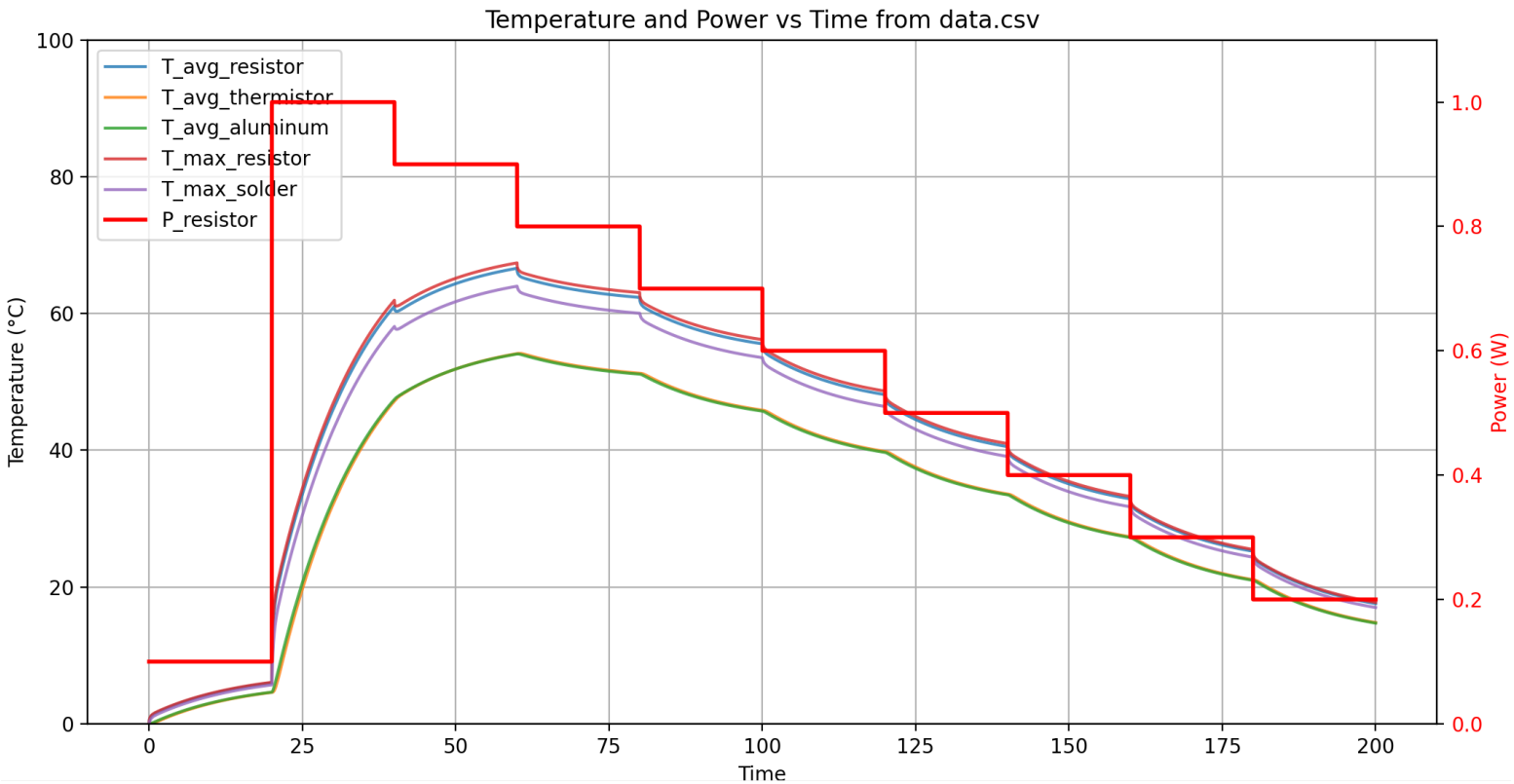






Temperature Observer Implementation 2/10 Training Data

- We prepare a simulated time history of system temperatures and variable power, that adequately demonstrates system dynamics.
- We will use this to “train” or fit our lumped parameters (R’s and C’s) for the simplified model



Temperature Observer Implementation 3/10

- Here’s the chatbot prompt that one-shots this task

I have 3 thermal masses, A B and C
When describing equations and code, you will use the naming convention A B and C

In reality these 3 masses represent the components of my thermal system
A is thermistor, which we measure known T_A from
B is heatsink
C is the power resistor, which we power with known P_C

Each mass has capacitance

There are the following thermal paths, each with resistance
A-B
B-C
B-ambient (ambient has variable temperature in deployment, but our training data only exhibits constant 0, I want to plan for variable temperature though)

I want to implement a temperature observer algorithm using a lumped parameter model to estimate realtime the temperature at the power resistor chip, so I can eliminate the lag and offset associated with only using the raw thermistor data.

I have the following data streams to work with in my circuit and IC: T_A, P_C, as well as T_ambient which is not included in the df

To fit/tune the model, I have a simulated time history in a df/csv(in discretized time steps, filename is data.csv) of the following columns (exact strings):

time
T_avg_resistor (T_C)
T_avg_thermistor (T_A)
T_avg_aluminum (T_B)
T_max_resistor
T_max_solder
P_resistor (P_C)

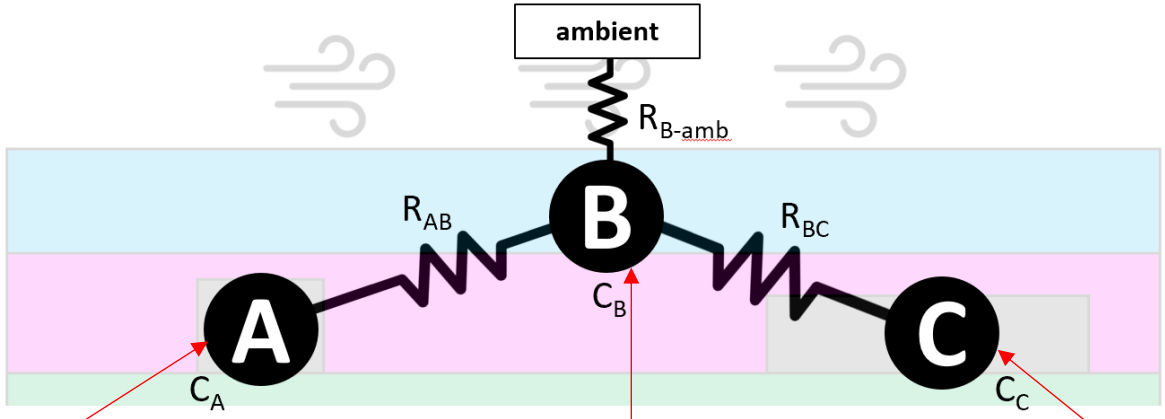
T_ambient is a constant 0 and is not in the df
Add this extra column before plotting and fitting

I eventually want to implement the observer here such that it produces an additional column for observed/predicted resistor temperature
This should be plotted along with the other traces so I can observe that it is doing its job right

Break down the implementation for me, theory first (with differential equations, etc), and then code implementation. Be thorough

Temperature Observer Implementation 4/10

- These ODEs describe our system dynamics



Intuitively:
"thermal mass A responds with a temperature rate change, when subjected to a thermal potential between B and A, with thermal path of resistance R_{AB}"

$$C_A \frac{dT_A}{dt} = \frac{T_B - T_A}{R_{AB}}$$

$$C_C \frac{dT_C}{dt} = \frac{T_B - T_C}{R_{BC}} + P_C$$

$$C_B \frac{dT_B}{dt} = \frac{T_A - T_B}{R_{AB}} + \frac{T_C - T_B}{R_{BC}} + \frac{T_{amb} - T_B}{R_{B_{amb}}}$$

Temperature Observer Implementation 5/10

- These ODEs can be discretized, and their parameters neatly factored out.

$$C_A \frac{dT_A}{dt} = \frac{T_B - T_A}{R_{AB}}$$
$$C_B \frac{dT_B}{dt} = \frac{T_A - T_B}{R_{AB}} + \frac{T_C - T_B}{R_{BC}} + \frac{T_{amb} - T_B}{R_{B_{amb}}}$$
$$C_C \frac{dT_C}{dt} = \frac{T_B - T_C}{R_{BC}} + P_C$$

Define $\theta_1 = \frac{1}{C_A R_{AB}}$, so:

$$\frac{T_A[k+1] - T_A[k]}{\Delta t} = \theta_1 (T_B[k] - T_A[k])$$

Define:

- $\theta_2 = \frac{1}{C_B R_{AB}}$
- $\theta_3 = \frac{1}{C_B R_{BC}}$
- $\theta_4 = \frac{1}{C_B R_{B_{amb}}}$

So:

$$\frac{T_B[k+1] - T_B[k]}{\Delta t} = \theta_2 (T_A[k] - T_B[k]) + \theta_3 (T_C[k] - T_B[k]) - \theta_4 T_B[k]$$

Define:

- $\theta_5 = \frac{1}{C_C R_{BC}}$
- $\theta_6 = \frac{1}{C_C}$

So:

$$\frac{T_C[k+1] - T_C[k]}{\Delta t} = \theta_5 (T_B[k] - T_C[k]) + \theta_6 P_C[k]$$

Temperature Observer Implementation 6/10

- We are now ready to apply linear regression to fit our parameters
- Below is our simplest case, with ODE #1 and θ_1

Outputs
This is known from our simulation data

Inputs
This is also known

$$\frac{T_A[k+1] - T_A[k]}{\Delta t} = \theta_1 (T_B[k] - T_A[k])$$

$$Y = \theta_1 \cdot X$$

Now it looks like a classic linreg problem

Fit theta to our known X Y data using least-squares

Temperature Observer Implementation 7/10
State Space Model

Lumped Parameters, which we have fitted with linreg

Parameter	Expression in R and C
θ_1	$\frac{1}{C_A R_{AB}}$
θ_2	$\frac{1}{C_B R_{AB}}$
θ_3	$\frac{1}{C_B R_{BC}}$
θ_4	$\frac{1}{C_B R_{B_{amb}}}$
θ_5	$\frac{1}{C_C R_{BC}}$
θ_6	$\frac{1}{C_C}$

These are the parameters, fitted to our training data

C_A = -0.0657
C_B = 0.2712
C_C = 0.0183
R_AB = -5.5354
R_BC = 13.9510
R_B_amb = 63.8913

Lumped Parameters

$$A = \begin{bmatrix} -\theta_1 & \theta_1 & 0 \\ \theta_2 & -(\theta_2+\theta_3+\theta_4) & \theta_3 \\ 0 & \theta_5 & -\theta_5 \end{bmatrix}$$

More Lumped Parameters

$$B = \begin{bmatrix} 0 & 0 \\ 0 & \theta_4 \\ \theta_6 & 0 \end{bmatrix}$$

Measured State Selector

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\dot{x} = Ax + Bu, \quad y = Cx$$

$$\dot{X} = \begin{bmatrix} \dot{T}_A \\ \dot{T}_B \\ \dot{T}_C \end{bmatrix}$$

States, Rates or Responses

$$X = \begin{bmatrix} T_A \\ T_B \\ T_C \end{bmatrix}$$

States

$$U = \begin{bmatrix} P_C \\ T_{amb} \end{bmatrix}$$

Inputs

$$y = T_A$$

Measurements

Temperature Observer Implementation 8/10 State Observer

- This is a Luenberger Observer
- This is what we actually run on the IC to "predict" $T_{resistor}$ in realtime
- Reminder:
  - We want to predict $T_{resistor}$
  - … With partial known information $T_{thermistor}$, $T_{ambient}$, P_{heat}

X-hat implies state estimations, NOT real measurements. These are virtually simulated on the IC in real time.

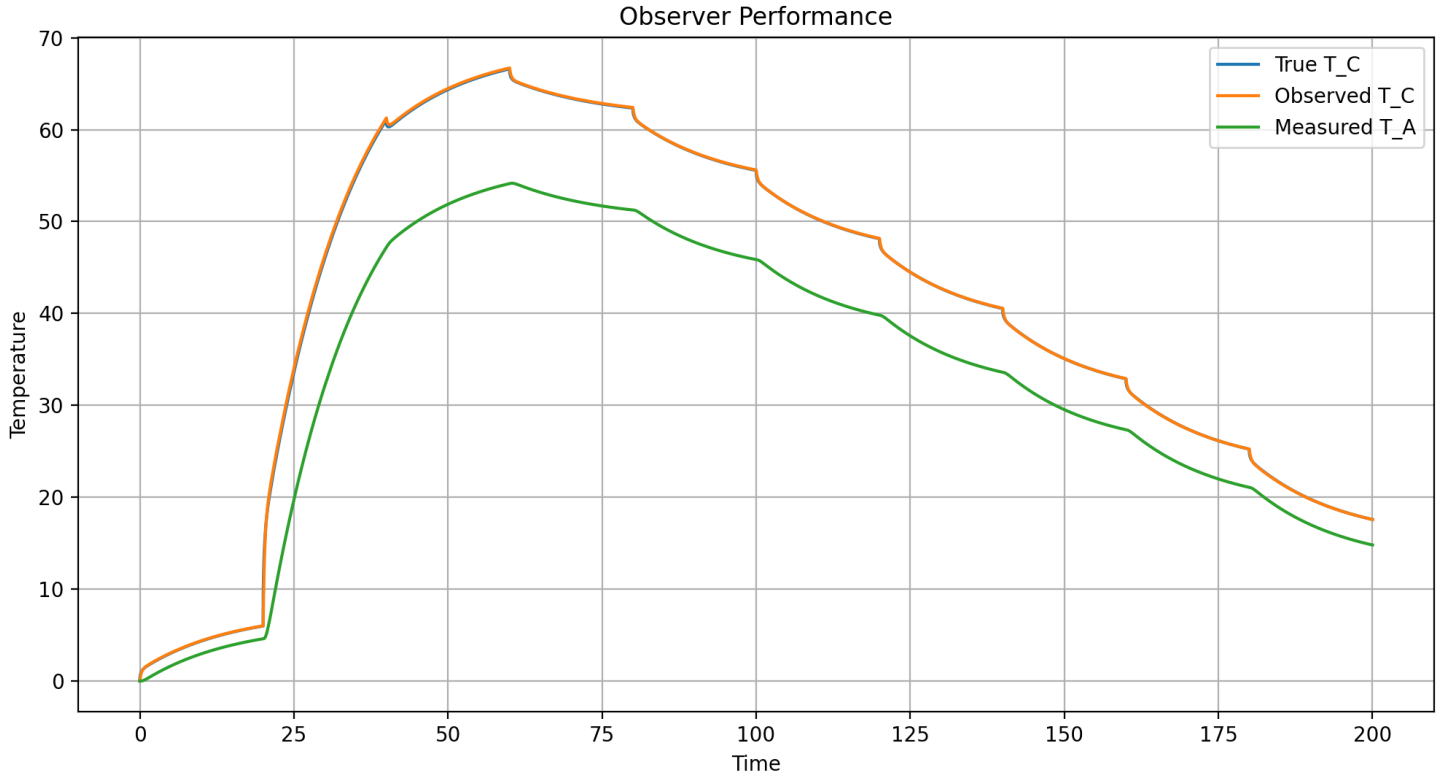
Luenberger gain, which, when informed by the error term, applies a corrective nudge to the state model

$$\dot{\hat{x}} = A\hat{x} + Bu + L\underbrace{(y - C\hat{x})}$$

error term, difference between measured and estimated $T_{thermistor}$

Temperature Observer Implementation 9/10 Observer Performance on Simulated Data

- We now simulate the observer, on top of our training data
- Below we plot $T_{resistor}$ (pre-existing data) and $T_{resistor\text{-}predicted}$ (what our observer predicts)
- Prediction performance seems spot-on, with very little tuning to the L weights

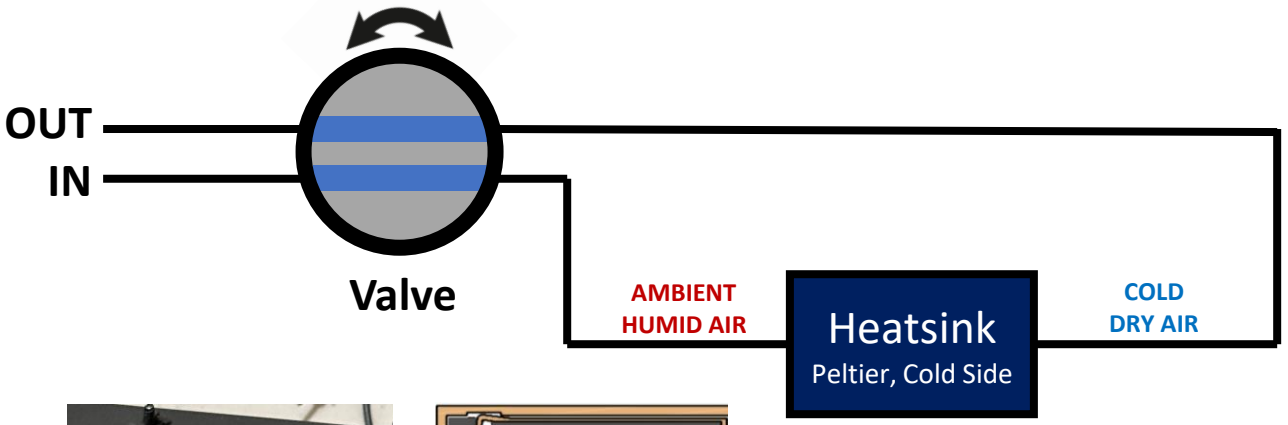


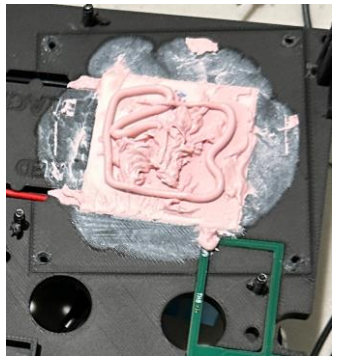
Temperature Observer Implementation 10/10 Thoughts

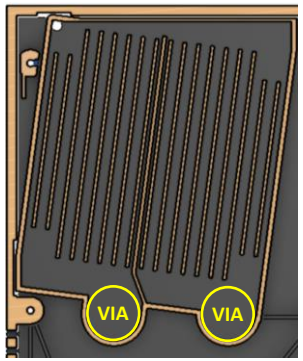
- The state-space model assumes constant R and C parameters. In practice, R can change.
  - Variable temperature means variable fluid viscosity and density. This affects convection coefficient.
  - Thermal conductivity of materials varies with temperature, though not by much in our application.
  - Convection coefficient is also affected by varying flow resistance, which comes from variable fluid temperatures and items placed in the flow path
  - The fan can get jammed or stop functioning
- The last case ^^^ is really scary since a Luenberger observer doesn't directly tell us if R_convection abruptly changes. In the code we may expose the error terms, which may be able to catch such a failure mode in progress.
- A fun alternative to a Luenberger observer is a neural net
  - A neural net can approximate any "traditional" mathematical structure like a PID controller or state observer. **And you don't need to know the math to get it to work**, which is fascinating.
  - All you need is a training data set that adequately demonstrates system dynamics. Diverse training data is easy to acquire if you already have a simulation set up.

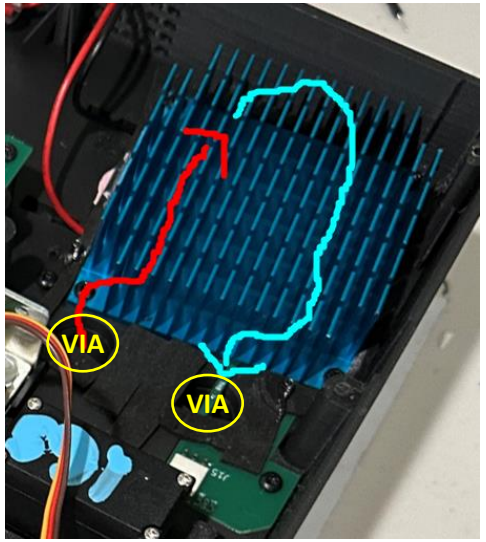
Peltier 1/5

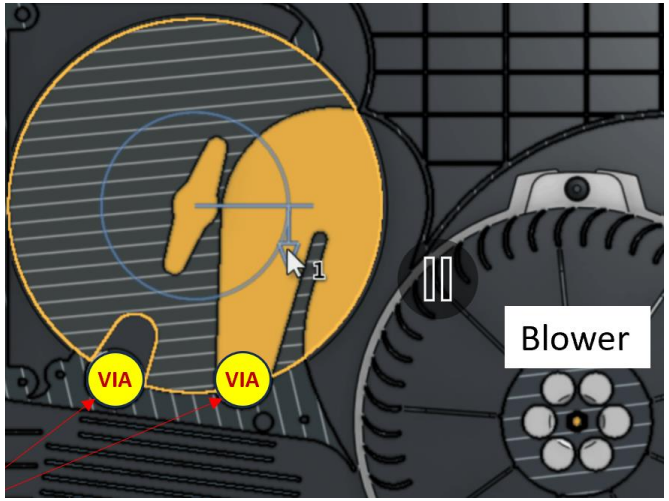
- We use a thermoelectric condenser to draw moisture out of the input air stream.
- A valve allows us to “cold charge” the condenser, letting us control input air temperature (and thus, input air moisture content)
- The dryer should **perform the same** in these two environments
  - **100ºF 100%RH**, *humid jungle*
  - **0ºF 10%RH**, *polar region*





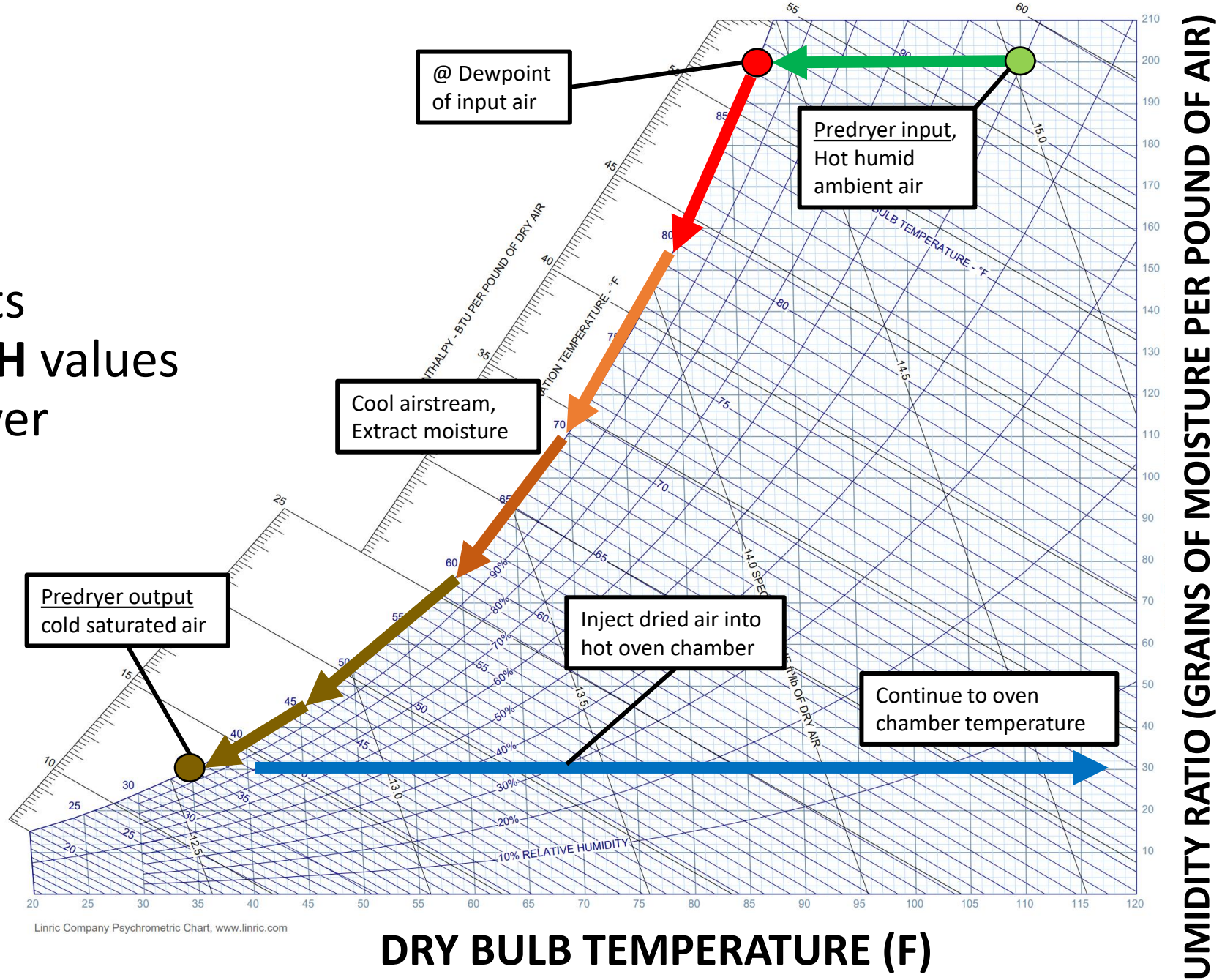






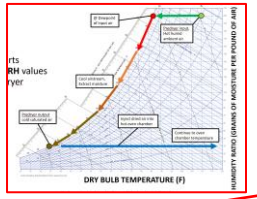
Peltier 2/5

- Psychometric charts illustrate **T** and **%RH** values through the predryer



https://www.engineeringtoolbox.com/docs/documents/816/psychrometric_chart_29inHg.pdf

Peltier 3/5



- Condenser calculator, STEADY STATE
- This row is the path taken on the previous slide

T in	T out	dT	RH In	Saturation Vapor Pressure, In	Partial Pressure of Water Vapor, In	Absolute Humidity, In	Saturation Vapor Pressure @ T Out	Partial Pressure of Water Vapor, Out	Absolute Humidity, Out	RH out @ Equiv T_out	RH out @ Equiv T_in		Air Vol Flowrate	Air Velocity out of 9mm^2 orifice	Air density, In	Air mass flow rate	Condensation rate	Time to collect 1g water		Latent Heat of Condensation, Water	Latent Heat of Fusion, Water	cp, Water	cp, Ice	cp, Air	Latent Heat, Vapor-to-Liquid	Fusion Heat, Liquid-to-Solid	Sensible Heat, Air	Sensible Heat, Liquid Water	Sensible Heat, Solid Water		Total Power
C	C	K	%RH	Pa	Pa	g/m^3	Pa	Pa	g/m^3	%RH	%RH		m^3/s	m/s	g/m^3	g/s	g/s	min		J/g	J/g	J/gK	J/gK	J/gK	W	W	W	W	W		W
43.3	1.67	41.7	50%	8789.1	4394.6	30.1	689.1	689.1	5.4	100%	8%		0.0001	11.1	1115	0.112	0.002465	6.8		2257	334	4.18	2.1	1.005	5.56	0.00	4.67	-	-		10.23
25	20	5	50%	3167.7	1583.8	11.5	2338.2	1583.8	11.7	68%	50%		0.0001	11.1	1184	0.118	-2E-05	-848.9		2257	334	4.18	2.1	1.005	-0.04	0.00	0.59	-	-		0.55
37	-5	42	100%	6274.6	6274.6	43.8	421.2	421.2	3.4	100%	7%		0.0001	11.1	1138	0.114	0.004043	4.1		2257	334	4.18	2.1	1.005	9.13	1.35	4.80	-	-		15.28

For some airflow rate (tunable with chokes)…

… Of an airstream with these spec targets …

… Our peltier absorbs this much power …

… And our arrangement of peltier(s), TIM material, and heatsinks, must ensure this dT capability on the airstream

Peltier 4/5

$T_{ambient}$ is the anchor from which we eventually determine $T_{cold\text{-}air}$ (through the thermal network)

$Rth_conv = (T_{heatsink\text{-}base} - T_{ambient}) / (P_{electrical} + Q)$

Q is sometimes published, but if not, it is non-trivial to find experimentally

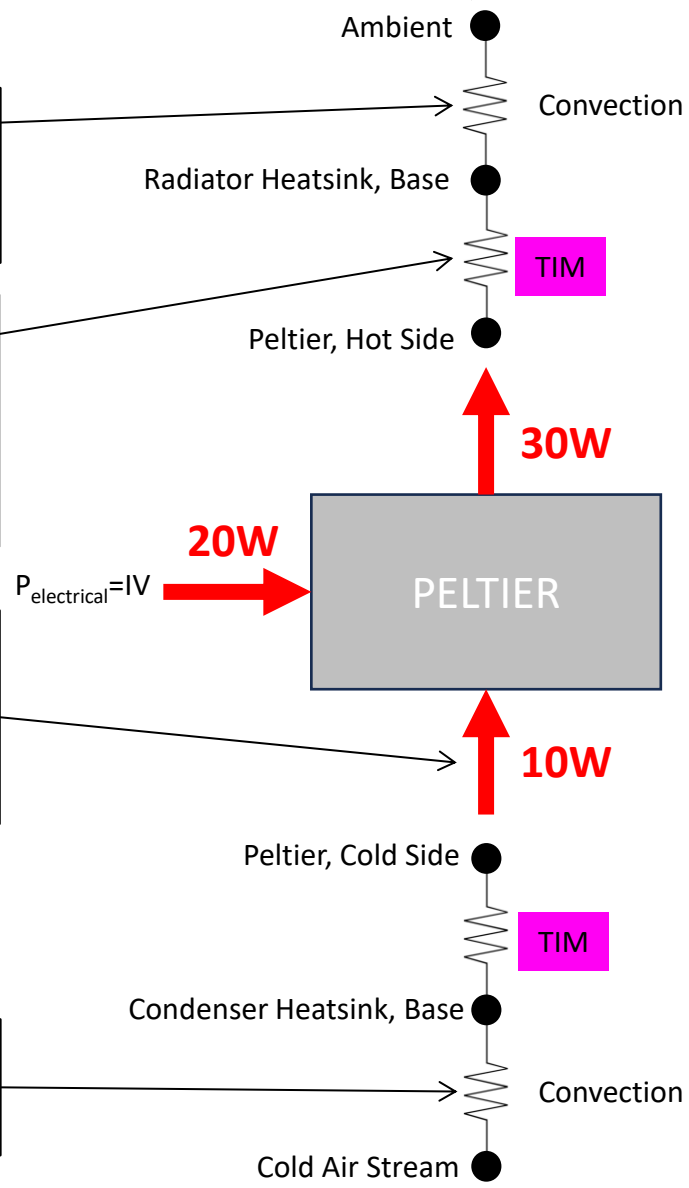
$Rth_TIM = t_{bondline} / (k * A_{cross\text{-}section})$

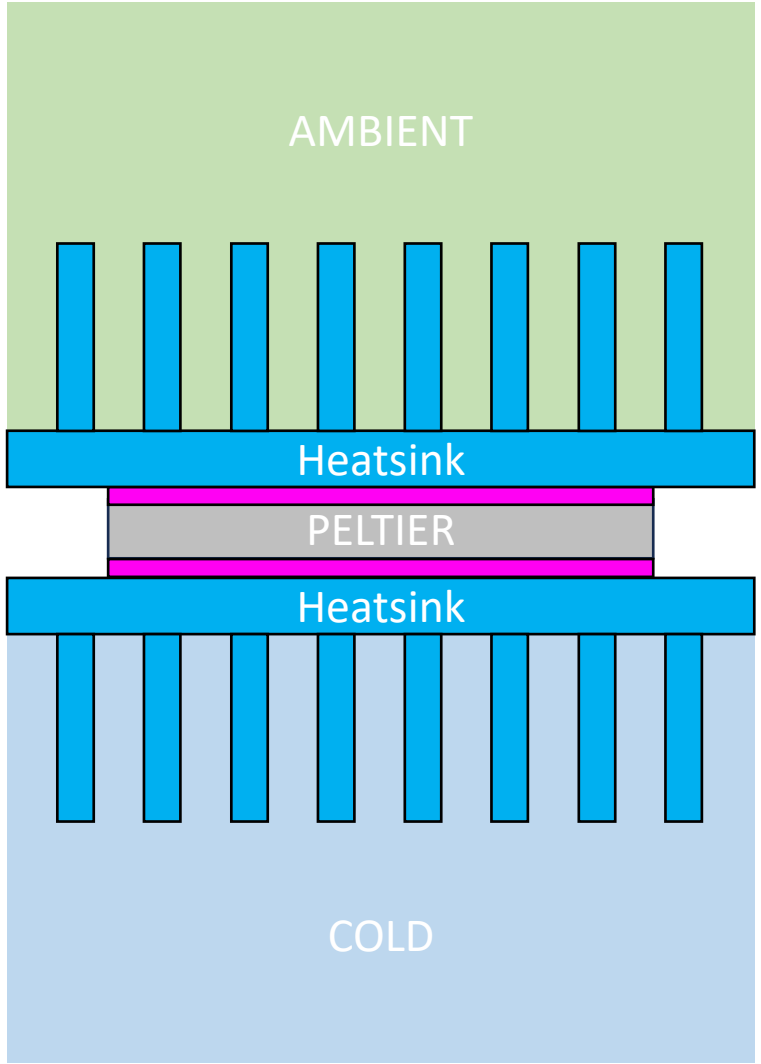
TIM Rth can be easily derived from known variables. However, bondline may not be known, especially at very thin bondlines. This can be isolated and measured with special instrumentation.

Peltier datasheets won't always publish Q_{cold}-dT curves, which are critical for sizing the peltier stack components

It's possible to DIY your own thermal power meters, which can be used to isolate and measure Q_{hot} and Q_{cold}.

Rth_conv increases with air flow rate … BUT
We want air flow rate to be low enough that we enhance the dT of the air stream, so that we condense effectively





Peltier 5/

- 1st iteration of peltier design **problems**:
  - Didn't do any peltier sizing calculations prior.
  - Evidence that air bleed is the dominating chamber Rth.
    - Need better seals.
      - Gaskets on lid interface.
      - Checkvalve exhaust on chamber.
      - Gaskets on valve.
    - The amount of airbleed required to negate peltier condenser benefits is ***TINY*** (as shown in previous calcs). So sealing has to be very good.
  - Heatsink has low thermal mass. This makes it less useful for a transient-use implementation.
  - dT capability only 20C with current TIM and hot-side heatsink selection.

Peltiers 6/

- <u>Thoughts on allowing condenser icing…</u>

- The good
  - **We can condense better.**
    - Accepting subzero condenser temperatures implies larger dT relative to ambient. This scales convective heat transfer proportionally.
    - It's very hard to bring air down to near 0°C… if your heatsink is only near 0°C

- The bad
  - Ice eventually clogs airpaths, preventing continuous use.
  - Condenser has to be warmed intermittently to melt the ice. This has implications:
    - Every freeze-thaw cycle will thermal cycle the peltier to some degree, stressing it. Long-term reliability would be a concern.
    - Requires some form of Icing detection or ice formation modeling.
    - No continuous positive air pressure in chamber. Leakage-into-chamber risk.

Peltiers 7/

- <u>Thoughts on air leakage…</u>

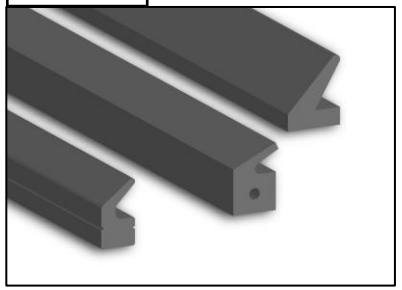
- Air leakage in 1st iteration was terrible. Achieved 0.67 K/W chamber-ambient Rth. Prior data suggests that 1.5 K/W is possible with better sealing!

- Design was intended to be self-sealing by buoyancy effect, since lid lip is lower than all chamber components. In theory, should not need very good sealing (expensive gaskets and compression force). I think I was wrong.

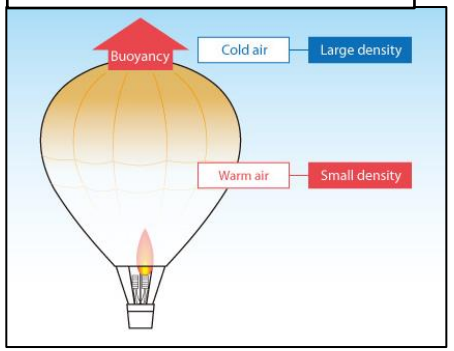
- Leakage-in
  - Is not tolerable, since this introduces moisture
  - Generally occurs where high-velocity air paths in-chamber are adjacent to sealing interfaces (Venturi effect)

- Leakage-out
  - Is required to exhaust moisture extracted from the workpiece.
  - But the magnitude of this leakage scales heater and peltier sizing.
  - This can be actively adjusted via valving, or left constant.

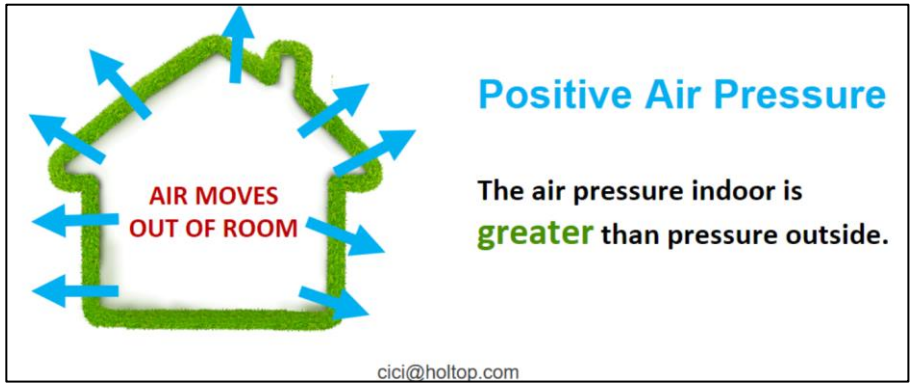
Gaskets



Current chamber design is like a hot air balloon



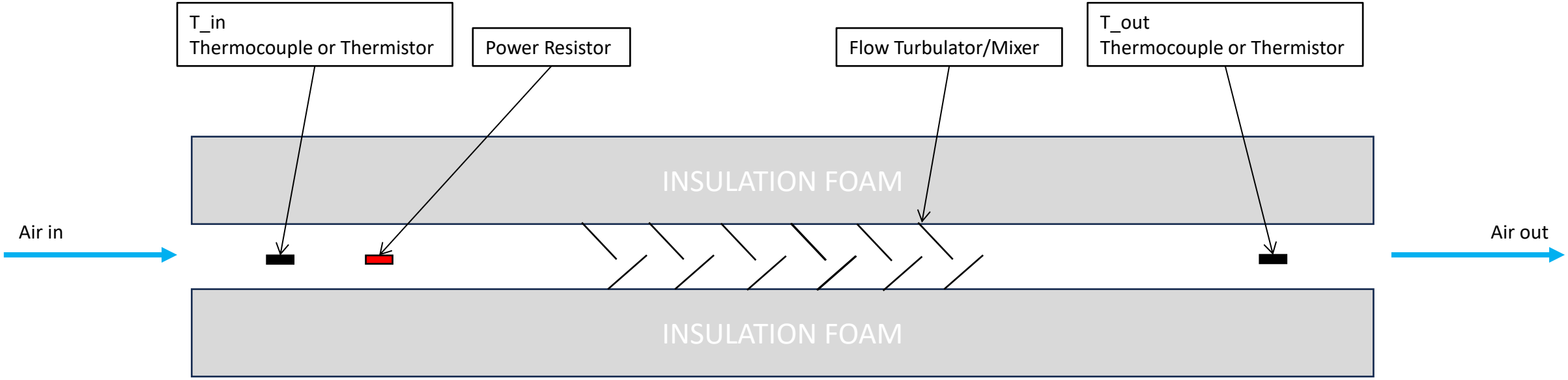
Peltiers 8/



- <u>Thoughts on positive air pressure in chamber</u>
- Positive air pressure, applied through the predryer air output, offers a way to eliminate Leakage-in entirely.
- Some flowrate is required to maintain positive air pressure given some degree of sealing (w/ or w/o gasket).
- The less sealing present, the higher this flowrate must be. So in this manner, gasketing/sealing establishes peltier/condenser sizing.
- Ideally, we should have great sealing, with minimal flowrate to maintain positive air pressure

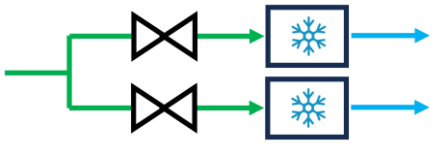
Peltiers 9/

- Solid-state air mass flow sensor for very low flowrates, a concept.
- mdot = P / (dT * cp)
- For test purposes, place in-line with condenser
- dP of flowmeter should be much lower than condenser, to avoid excessive shift after removing flowmeter from production intent assembly



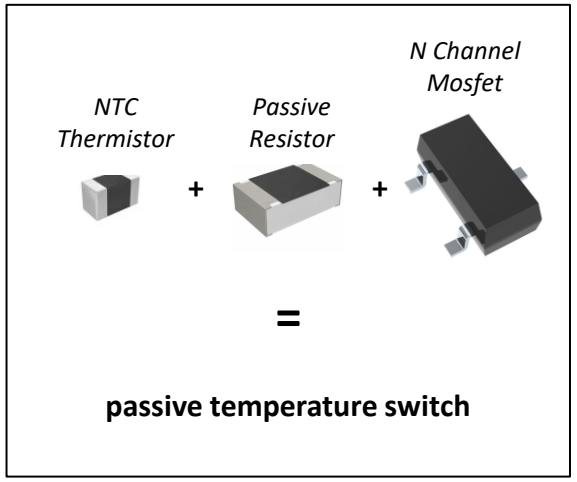
Peltiers 10/

- Simplified concept generation
- Missing
  - Attached documentation substantiating each cell, especially cost for given capability
  - Drying capability is a spectrum. Cost is discretized based on available parts.
  - The gaskets column may be a moot point, since we may just want super good gaskets regardless of concept, given our sensitivity to leakage-in

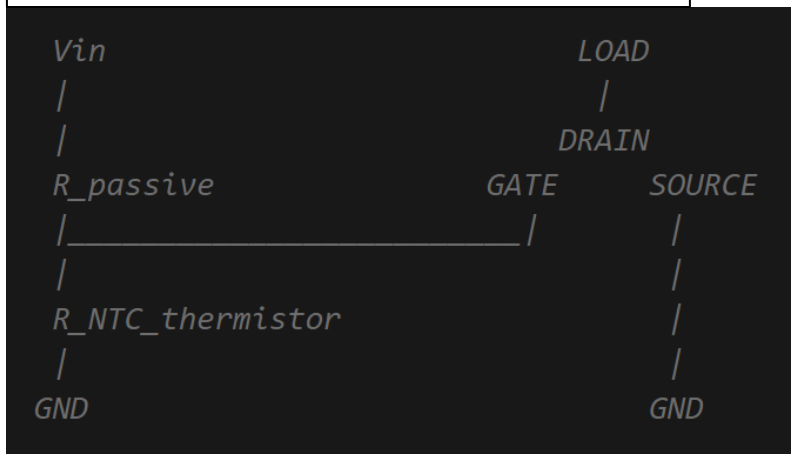
		Continuous use possible	Requires Active Valving	Requires Performant Gaskets	Positive air pressure maintained with dry air	Drying capability	COST
Single Condenser, Icing Allowed		NO	YES	YES	YES	GOOD	$$
Double Condenser, Icing Allowed		YES	YES	NO	NO	GOOD	$$$
Single Condenser, NO Icing Allowed		YES	NO	NO	YES	BAD	$

Passive Overtemperature Protection 1/3

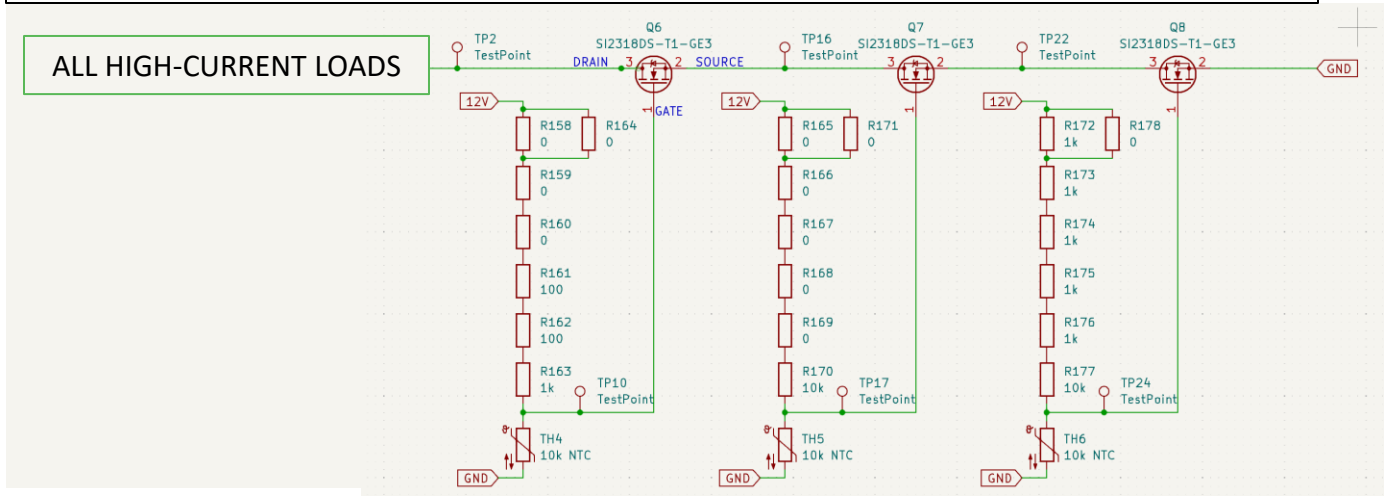
- This is a passive (no IC logic) temperature switch
  - Basically a thermistor voltage divider driving a MOSFET.
  - When temperature reaches a predetermined threshold, voltage divider output should cross Vgs(th), turning the MOSFET off
  - Resistor-tunable
  - Daisy chainable
  - Requires no new parts (already have MOSFETs, Thermistors, and passives)
- Why do we need this? Because active safety doesn't always work.
  - Firmware bugs
  - Failed ICs
  - Shorted MOSFETs
- Why not use off-the-shelf thermostat ICs?
  - Expensive
  - Insufficient temperature ranges
  - Too tall to sneak into a TIM layer



Schematic for single passive temperature switch



Example application, daisy-chained passive protection circuits, one for each temperature monitor
It's like a breaker box



Passive Overtemperature Protection 2/3

- A spreadsheet calculator can be used to manually tune our temperature switches
- …But this is tedious.
- …And this ignores electrical tolerances.
- Let’s use a numerical solver instead

					we measure voltage (Vout) across this NTC resistor				
					Back-calculate the resistance of this based on assigned V_out and known R1 passive				
		Resistor in series with thermistor			R2 in photo				
		R1 in photo			v		Both of these are provided on NTC thermistor datasheet (B @ Temp)		
		v					v	v	
		assign	*assign*	*assign*	*calc*		*assign*	*assign*	*calc*
Name	**URL**	**R_passive**	**Vout**	**Vin**	**R**		**B_2585**	**R_0**	**T**
		Ohms	*Volts*	*Volts*	*Ohms*		*Kelvin*	*Ohms*	*Celcius*
Vishay 10k	https://ww	1200	0	12	0		3940	10000	#NUM!
"	"	1200	0.5	12	52.17391		3940	10000	221.9
"	"	1200	1	12	109.0909		3940	10000	179.9
		1200	1.5	12	171.4286		3940	10000	157.5
		1200	2	12	240		3940	10000	142.2
		1200	2.5	12	315.7895		3940	10000	130.6
		1200	3	12	400		3940	10000	121.0
		1200	3.5	12	494.1176		3940	10000	112.8
		1200	4	12	600		3940	10000	105.6
		1200	4.5	12	720		3940	10000	99.1
		1200	5	12	857.1429		3940	10000	93.1
		1200	5.5	12	1015.385		3940	10000	87.4
		1200	6	12	1200		3940	10000	82.0
"	"	1200	6.5	12	1418.182		3940	10000	76.7
"	"	1200	7	12	1680		3940	10000	71.5
"	"	1200	7.5	12	2000		3940	10000	66.3
"	"	1200	8	12	2400		3940	10000	61.1
"	"	1200	8.5	12	2914.286		3940	10000	55.7
"	"	1200	9	12	3600		3940	10000	50.0
"	"	1200	9.5	12	4560		3940	10000	43.8
"	"	1200	10	12	6000		3940	10000	37.0
"	"	1200	10.5	12	8400		3940	10000	29.0
"	"	1200	11	12	13200		3940	10000	18.9
"	"	1200	11.5	12	27600		3940	10000	3.7
"	"	1200	11.9	12	142800		3940	10000	-24.9

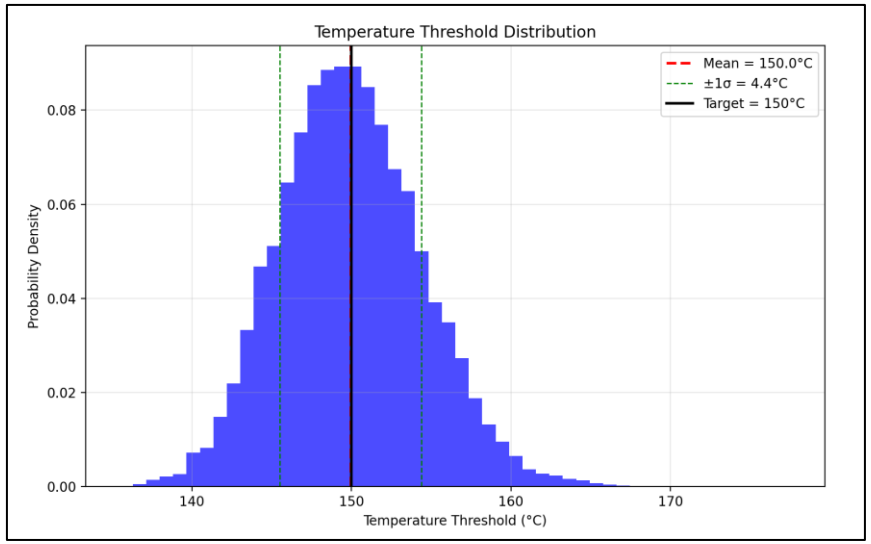
Transition zone for threshold Vgs.

Heating power shall be throttled before we reach **top temperature**.

The **low temperature** in this transition zone should still be above normal operating temperatures, such that we don't interfere with the heater circuit.

Passive Overtemperature Protection 3/3

- Binary search to tune passive resistors!
- Monte Carlo to characterize T(th) distribution!
  - Get resistor and thermistor tolerances from datasheets.
  - V_gs(th) has pretty massive tolerance on a typical data sheet, which is scary. But we can characterize this ourselves by measuring shipments of devices. It’s typically a lot tighter :D
- This is how we build confidence in a DIY circuit to protect customers at scale.



Results printout

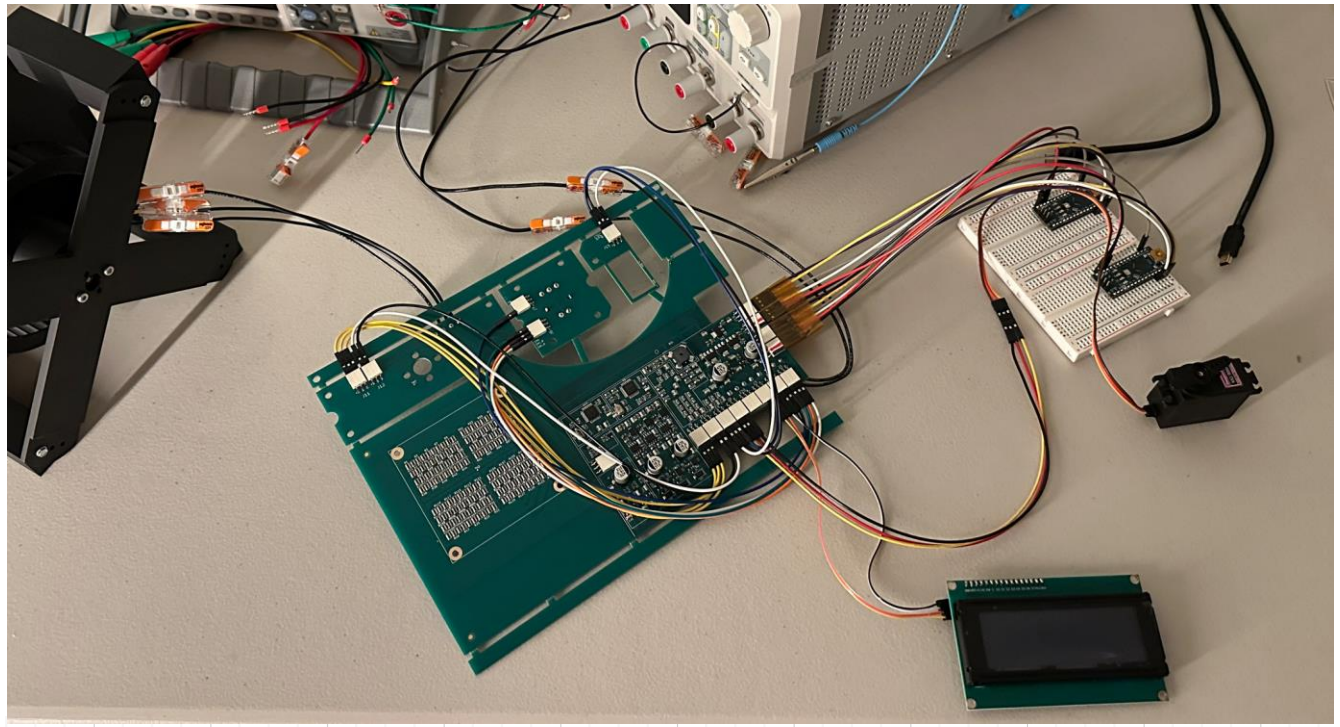
```
iteration,R_passive,mean,stdev
1,10050.00,80.31,3.06
2,5075.00,106.87,3.52
3,2587.50,137.29,4.14
4,1343.75,171.97,4.84
5,1965.62,151.16,4.37
6,2276.56,143.59,4.28
7,2121.09,147.24,4.31
8,2043.36,149.13,4.43
9,2004.49,150.22,4.45
10,2023.93,149.62,4.37
11,2014.21,149.97,4.37
solved!

passive R =     2014.21 ohms
mean T_th =     149.97 C
stdev T_th =    4.37 C
```

Monte carlo circuit simulation setup in Python

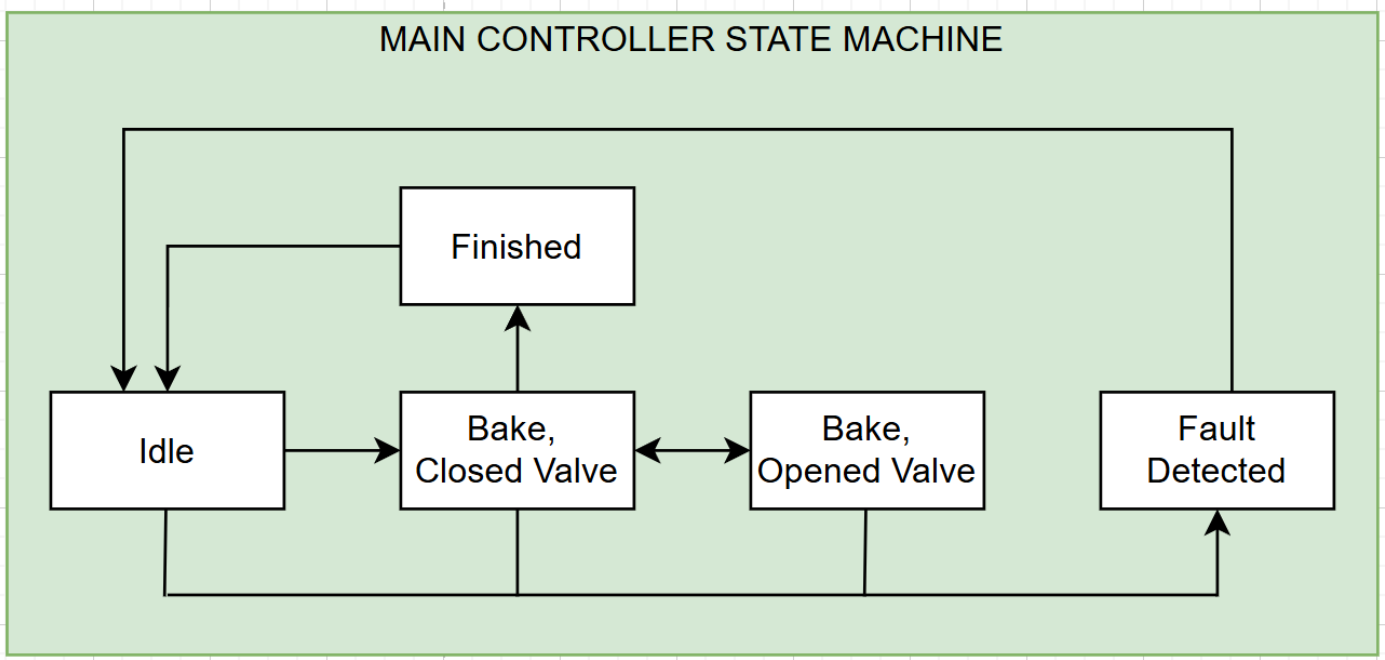
```python
import numpy as np
from matplotlib import pyplot as plt

# ================================ 1. ELECTRICAL CIRCUIT PARAMETERS ================================

mean_thermistor_B = 3455
stdev_thermistor_B = (mean_thermistor_B*0.01)/3

mean_thermistor_R = 10000
stdev_thermistor_R = (mean_thermistor_R*0.01)/3

mean_mosfet_Vgs = 0.7
stdev_mosfet_Vgs = 0.05

passive_R_lower = 100      # lower bound for binary search
passive_R_upper = 20000    # upper bound for binary search
#assume exact value, not a distribution, for passive R. This keeps algorithm simple.

mean_V_in = 5  # voltage divider high-side voltage
stdev_V_in = (mean_V_in*0.01)/3


# ================================ 2. SOLVER ================================


T_th_target = 150           # <<< USER INPUT  temperature switch will turn off after exceeding this temperature

convergence_threshold = 1   # binary search completes when solution is < X from target
iteration_count = 1         # binary search iteration
population = 10000          # monte carlo population size per iteration
mean_T_th = -100            # fake off-target value for initialization purposes
passive_R = 0               # fake off-target value for initialization purposes

print("iteration,R_passive,mean,stdev")

while abs(T_th_target-mean_T_th) > convergence_threshold:

    T_th_arr = []

    for i in range (population):

        # average passive resistor for binary search
        passive_R = (passive_R_lower + passive_R_upper)/2

        # vary electrical parameters
        thermistor_B = np.random.normal(mean_thermistor_B, stdev_thermistor_B)
        thermistor_R = np.random.normal(mean_thermistor_R, stdev_thermistor_R)
        mosfet_Vgs = np.random.normal(mean_mosfet_Vgs, stdev_mosfet_Vgs)
        V_in = np.random.normal(mean_V_in, stdev_V_in)

        # compute T_th
        T_th = 1/((1/298.15)+(1/thermistor_B)*np.log(mosfet_Vgs*passive_R/(thermistor_R*(V_in-mosfet_Vgs)))) - 273.15
        T_th_arr.append(T_th)

    # describe the distribution
    mean_T_th = np.mean(T_th_arr)
    std_T_th = np.std(T_th_arr)

    print(f"{iteration_count},{passive_R:.2f},{mean_T_th:.2f},{std_T_th:.2f}")

    # narrow the binary search
    if mean_T_th > T_th_target:
        passive_R_lower = passive_R
    elif mean_T_th < T_th_target:
        passive_R_upper = passive_R

    iteration_count += 1

print("solved!")
print("")
print(f"passive R = \t{passive_R:.2f} ohms")
print(f"mean T_th = \t{mean_T_th:.2f} C")
print(f"stdev T_th = \t{std_T_th:.2f} C")
```

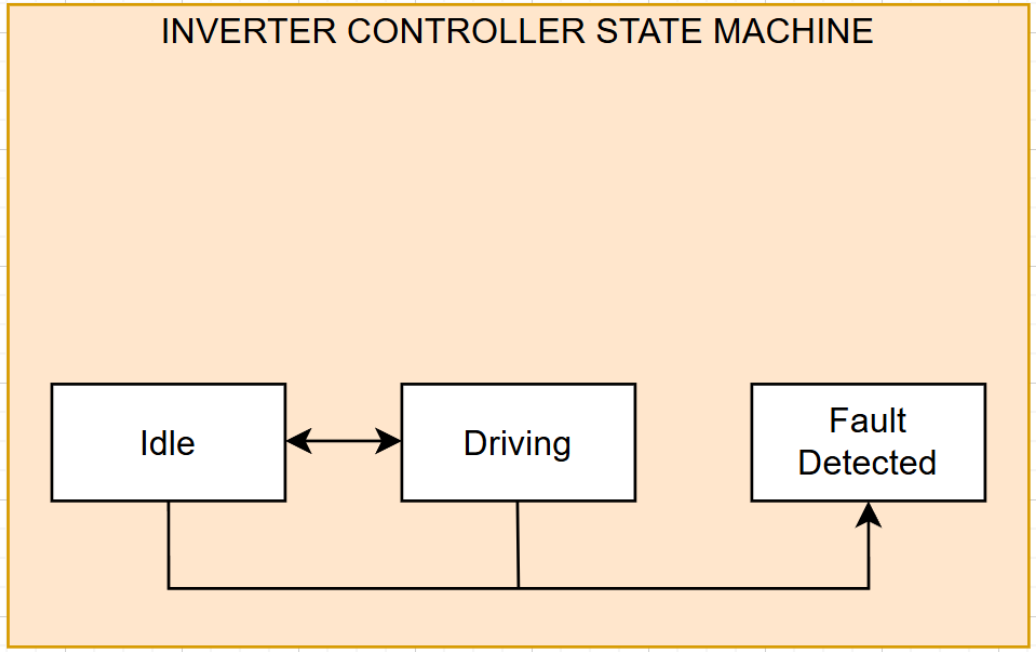
Firmware

- 2x ATmega328pb
- Arduino C



- CurrentSensor.cpp
- CurrentSensor.h
- CustomServo.cpp
- CustomServo.h
- EncoderSwitch.cpp
- EncoderSwitch.h
- FD_B_MAIN.ino
- Heater.cpp
- Heater.h
- LCD.cpp
- LCD.h
- Thermistor.cpp
- Thermistor.h
- UI.cpp
- UI.h

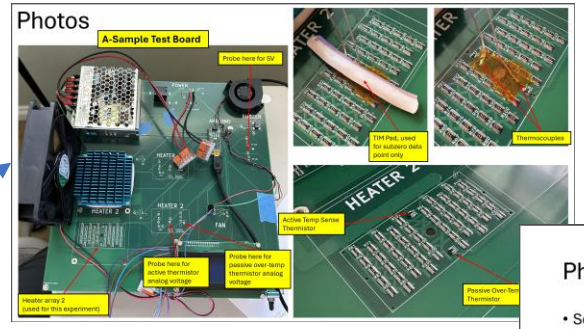


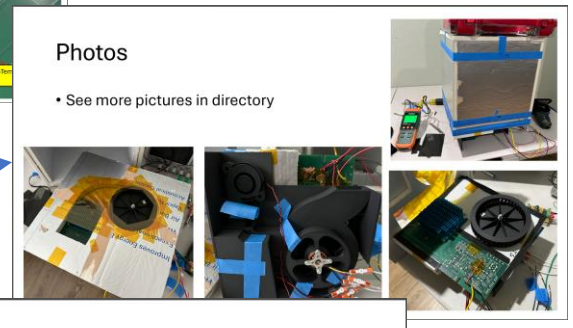


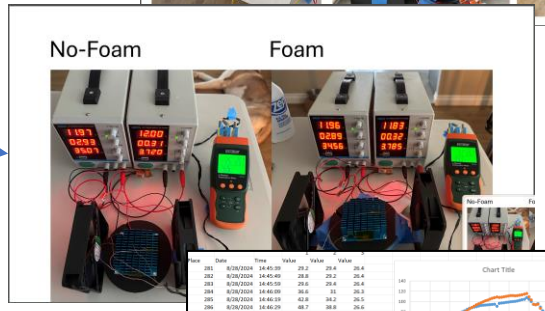
Documentation

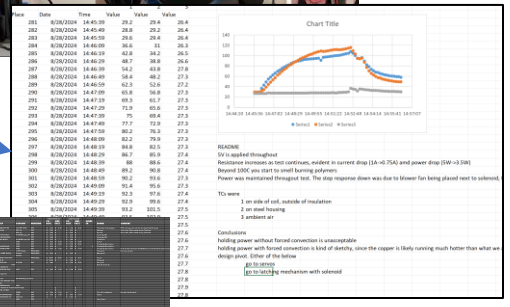
- Follows Tesla-style documentation practices, all in onedrive
- Even alone, this is very useful for maintaining mental continuity over months of work
- For engineering teams, this makes or breaks the mission

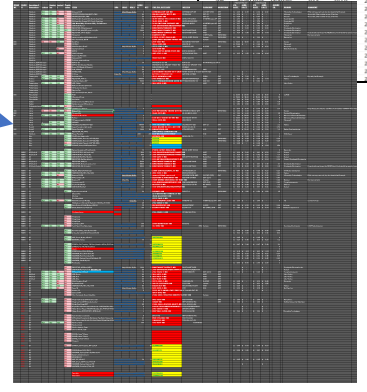
- 2024-01-13 - NC - Insulated Box Experiment
- 2024-01-19 - NC - Blower Heater Experiment
- 2024-06-12 - NC - Thermistor Thermocouple Agreement, and Passive Overtemperature
- 2024-06-13 - NC - Current Sensing, Noise Mitigation
- 2024-07-08 - NC - A2 Build and Test
- 2024-07-10 - NC - Peltier Module, First Experiment
- 2024-08-21 - NC - BLDC Motor Controller Development
- 2024-08-28 - NC - Solenoid Thermals
- Datasheets
- Drawings
- Literature
- Manufacturing Files
- 2023-12-31 - NC - BOM, Filament Dryer
- 2023-12-31 - NC - Illustrations
- 2024-01-08 - NC - General Notes
- 2024-01-24 - NC - Introduction, High Temperature Filament Dryer
- 2024-06-10 - NC - Spend Timeline
- 2024-06-14 - NC - Atmega328P Pin Utilization
- 2024-06-14 - NC - DFMEA, Filament Dryer
- 2024-06-21 - NC - Thermistor Trace Design Guideline
- 2024-07-28 - NC - Capex Spend
- 2024-08-18 - NC - BLDC Motor Control

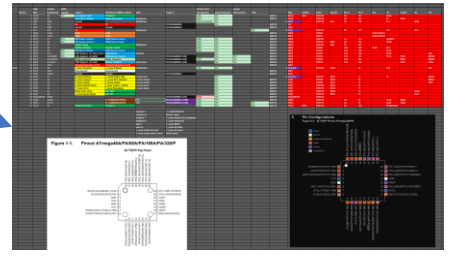






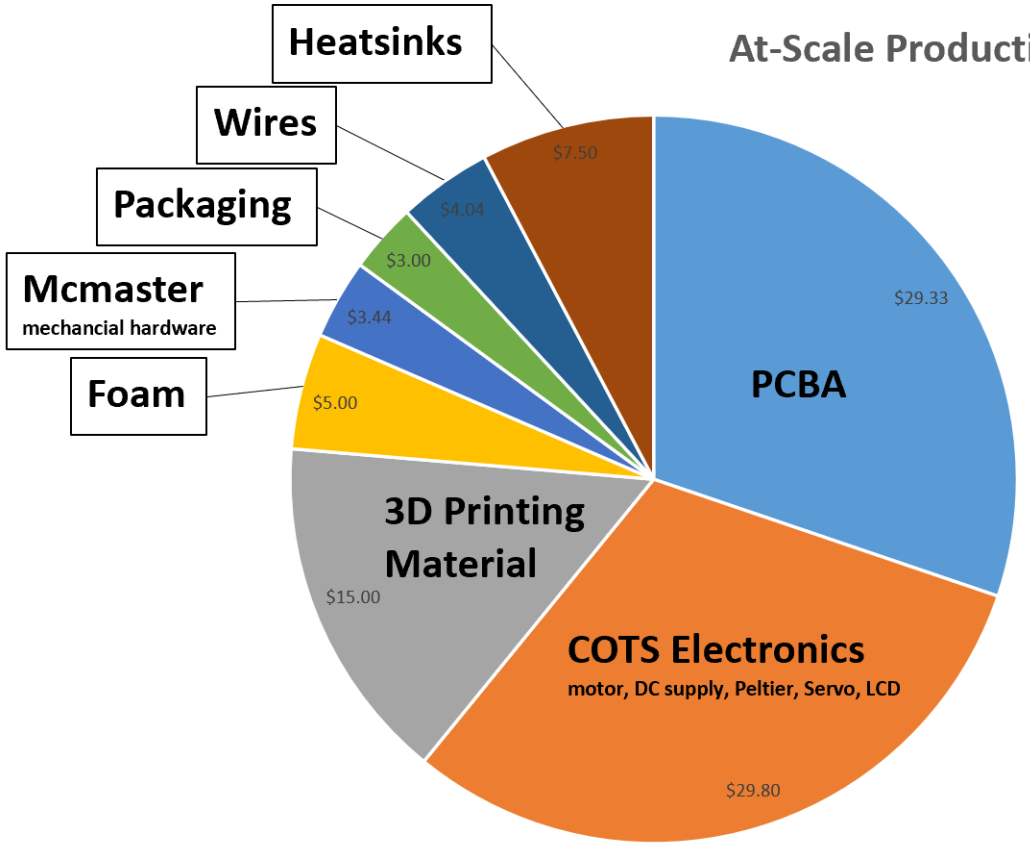






BOM

- Suppliers: Digikey, McMaster, JLCPCB, Lowes, Bambulabs, Amazon
- Production-intent PCBA has 22 unique components
- 6 parts need direct factory sourcing to access lower pricing. Everything else can be bought via online portals.



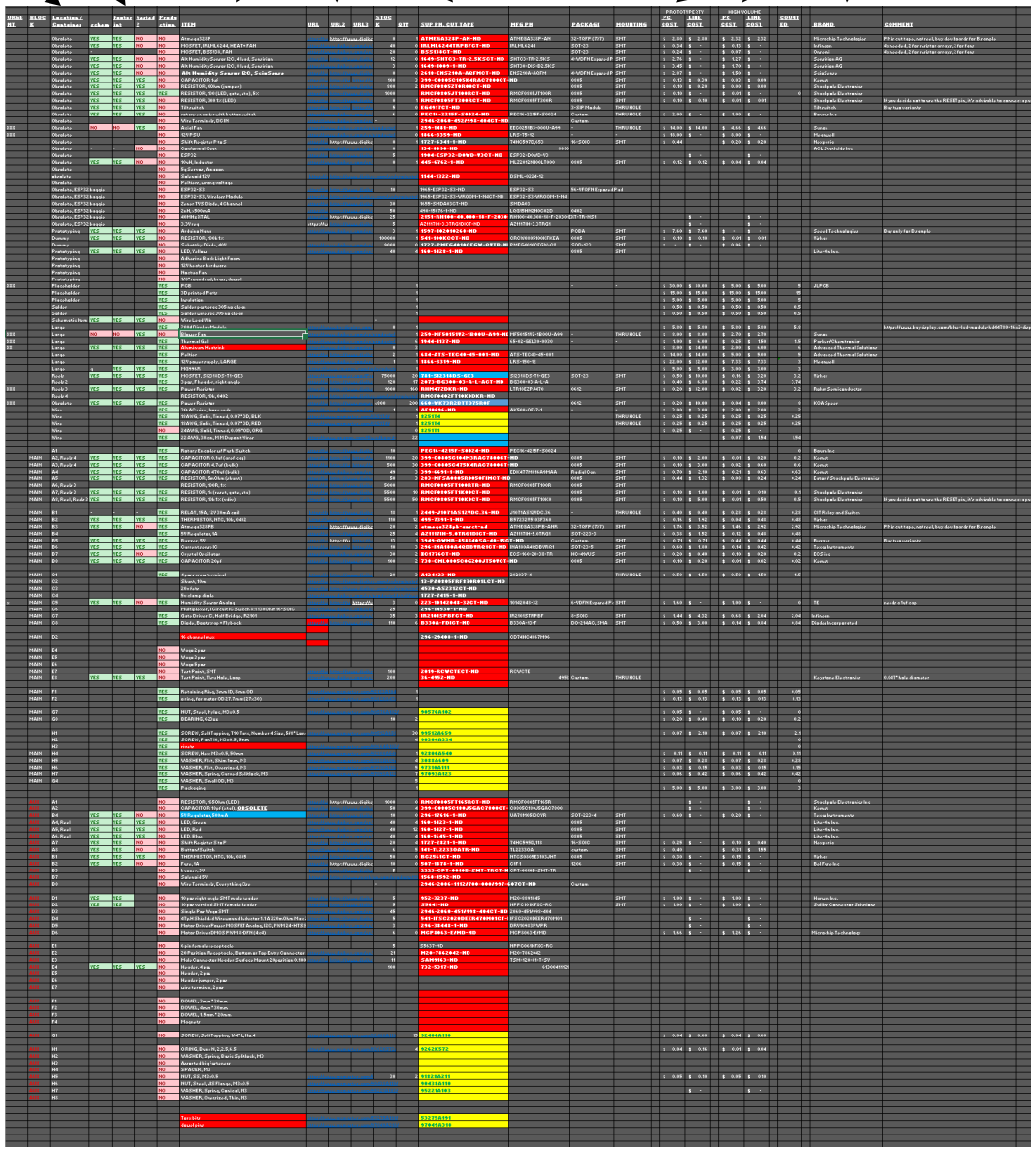
At-Scale Production Cost: $97.11

Inventory Location
Ex. Organizer Box 1 – Cell C7

Status Trackers
Design, Obsolete, etc

Name, URLs, PNs

Low Vol $, High Vol $, Supplier



Lab Space and Inventory

- Lab infrastructure enables complex projects
- Every BOM item has a documented location
- I appreciate lab/manufacturing managers

Lessons Learned

- Fully demonstrate the physics before investing a lot into mechanical integration CAD.
- PCBA assembly and mechanical assembly are extremely time consuming. A single technician would have been very well leveraged.