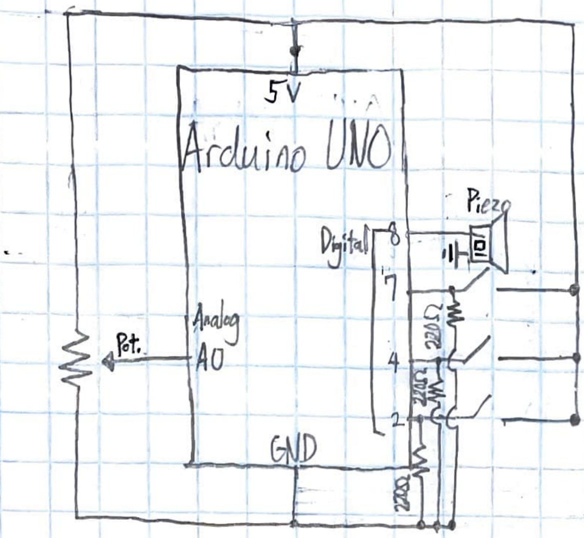
Electric Trumpet - 1st Design
11/25/24



Switches - Valves

Piezo - Sound output

Potentiometer - Simulate airspeed/pressure change

Pseudo Code / Software Design
11/27/24

```
void loop() {
    get partial input (potentiometer/air pressure sensor)
    get valve input ⟶ array of 0/1
                                unpressed/pressed
   get tone: partial - valve input
                       1: -2 Semitone
                       2: -1 Semitone
                       3: -3 Semitone
    tone output (piezo)
}
```

Note vs Frequency Relationship
11/27/24

Equal Temperament System

$$f = f_0 \left(2^{\frac{1}{12}}\right)^n$$

f: frequency
f_0: relative frequency
n: number of semitones from f_0 to f

Potentiometer & Piezo Upgrade
12/3/24

Potentiometer - knob to set partial

Pressure Sensor - pressure vs time (internal pressure oscillation matches pitch oscillation)

$\downarrow$ FFT

amplitude vs frequency

$\downarrow$

Output frequencies with corresponding volume (amplitude)

Piezo - single square wave

Speaker - emit overtones/harmonic series

Arduino → amplifier → speaker
(needed to achieve volume)

Valve Mechanism Design
12/15/24

Piston

Side

Top

Valve casing

Side

(Valve guide shown in dotted lines)

Top

Full Assembly (Design only, no dimensions)

Pressure Sensor Transfer Function
12/21/24

$$\text{Output (\% of } 2^{14} \text{ counts)} = \frac{80\%}{P_{max}-P_{min}} \times (Pressure_{applied} - P_{min}) + 10\%$$

$$\frac{O - O_{min}(10\% \text{ of } 2^{14})}{\underbrace{O_{max}(90\%) - O_{min}(10\%)}_{\text{Output range}}} = \frac{P_{applied} - P_{min}}{\underbrace{P_{max} - P_{min}}_{\text{Pressure range}}}$$

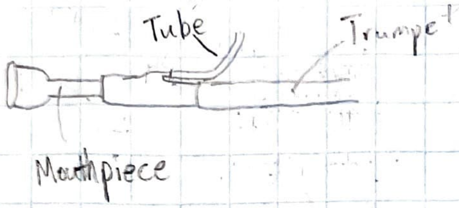
O : Raw output (14 bits)
O_{max} : Max output (90% of 2^{14})
O_{min} : Minimum output (10% of 2^{14})
P_{max} : 5psi
P_{min} : −5psi
$P_{applied}$: applied pressure

$$P_{applied} = \frac{O - O_{min}}{O_{max} - O_{min}} \times (P_{max} - P_{min}) + P_{min}$$

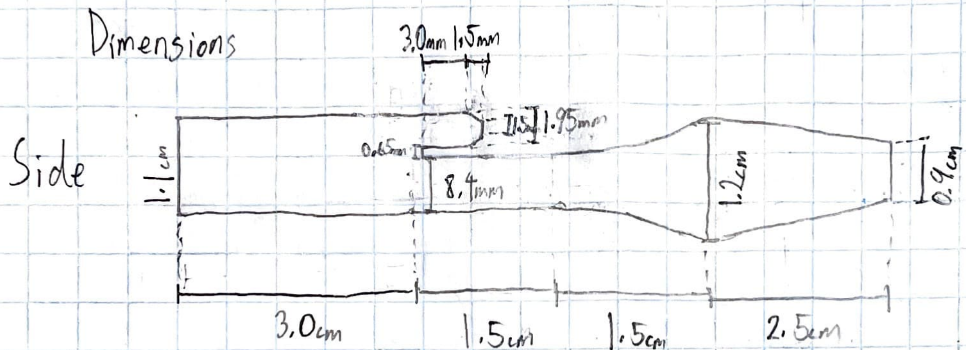
$$P_{applied} = \frac{O - 1638}{13107} \times 10 - 5$$

Trumpet-Tube ~~a~~ Adapter 12/26/24

Adapter for pressure sensor tube to trumpet



Dimensions



eTrumpet Program V2 1/5/24

Valve Input

↓

Determine harmonic series / list of possible notes

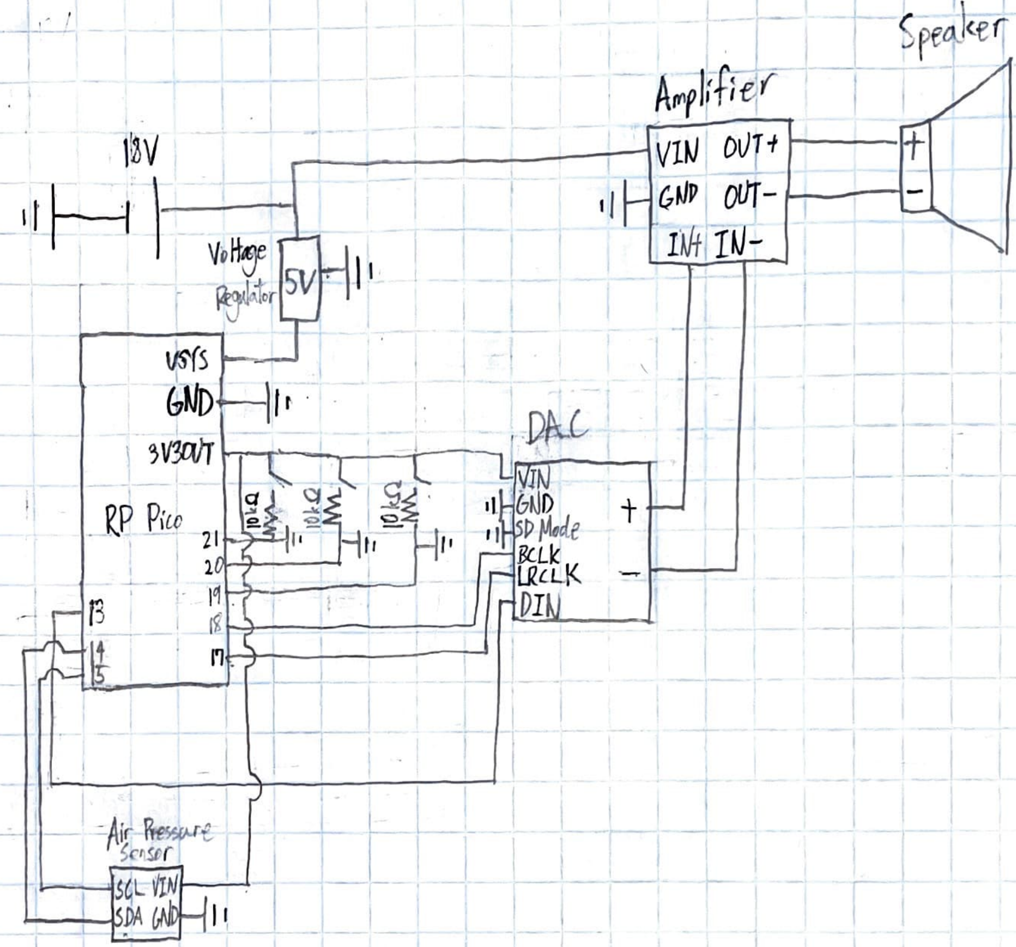
↓

Compare list w/ FFT output to determine pitch ← FFT output most prevalent frequency

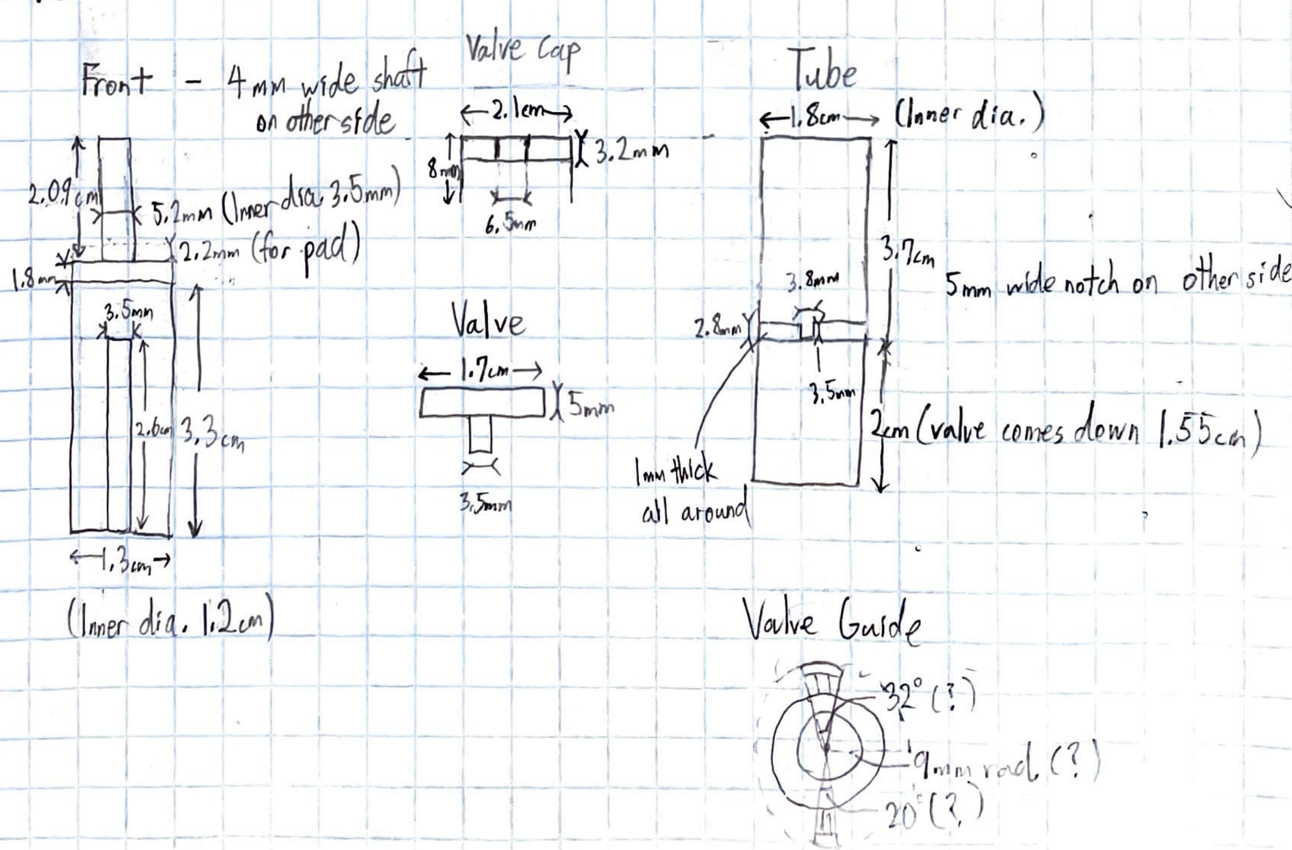
↓

Output

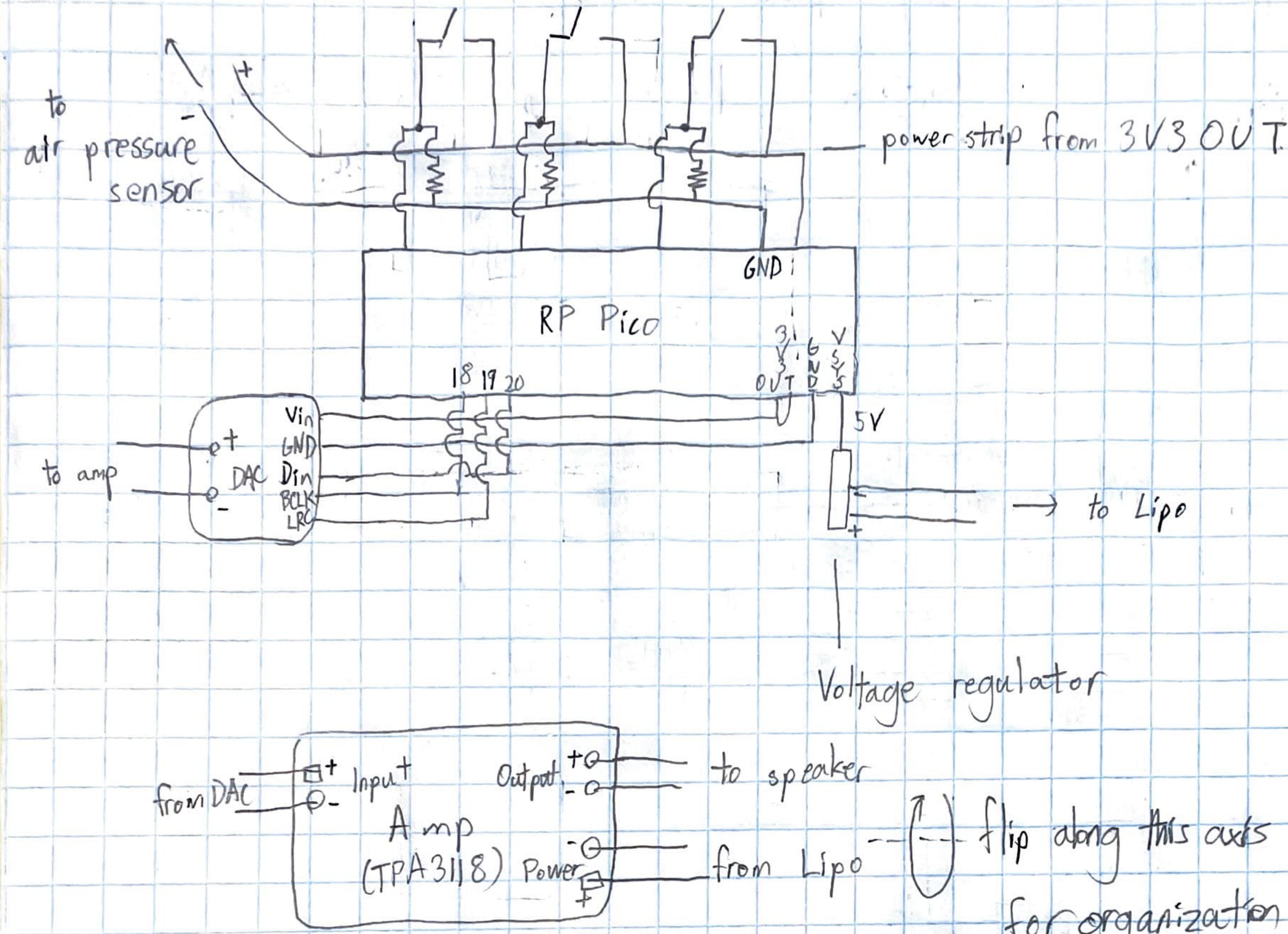
eTrumpet Schematic w/ Amp & Speaker 2/15/25



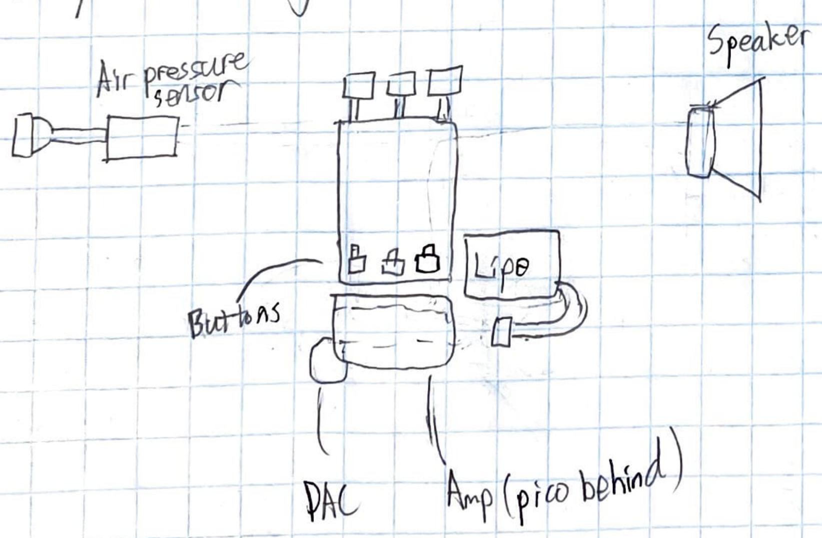
Valve Schematic 2/20/25



Component Placement 3/11/25

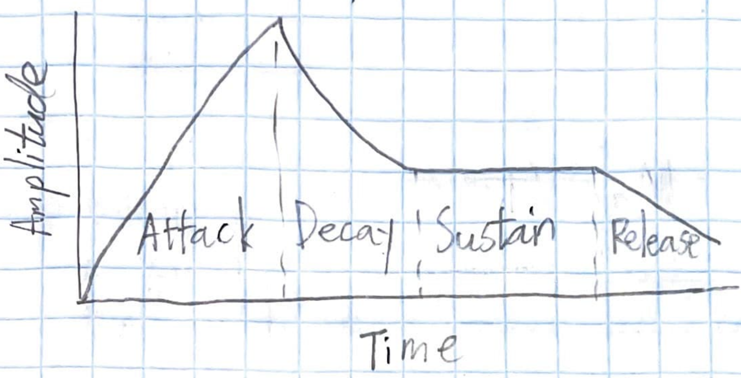


Layout (wiring removed for ease of viewing)



Tonguing Detection 4/20/25

- ADSR Envelope



Detecting Attack
- sudden spike in pressure

Detecting Release
- sudden drop in pressure

Threshold
- won't work for different volumes
- unreliable (only relies on 1 point)

☆ - Z score of pressure
independent of volume

Standard deviation
- Find the SD of SD...?

↓

Works, but slow

- derivative of SD → volume dependent

Volume Mapping 4/21/25

Min Amplitude – FFT Amplitude – Max Amplitude

↓

Min Volume – Volume – Max Volume

Finding min/max amplitude

1. Play trumpet as loud/soft as possible
2. Record amplitude reading

Finding min/max volume

- Adjust volume until sound distorts / is inaudible

Min amplitude : 1
Max amplitude : 20 (mouthpiece only)
30 (full horn)

Volume Scaling
Min volume : 0.001
Max volume : 0.017

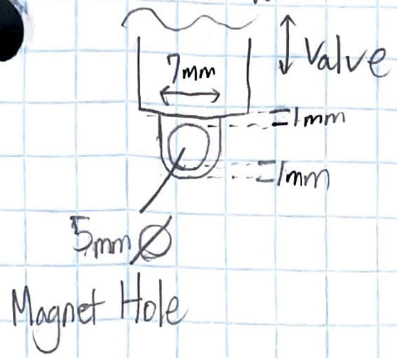
Mapping Function

$$Volume = Vol_{min} + \frac{Amplitude - Amp_{min}}{Amp_{max} - Amp_{min}} (Vol_{max} - Vol_{min})$$

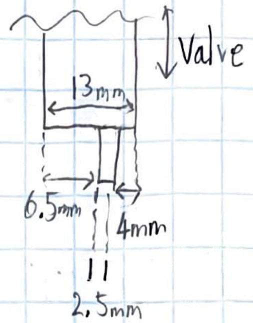
☆ Combine volume mapping with ADSR

Hall Effect Valve 5/17/25

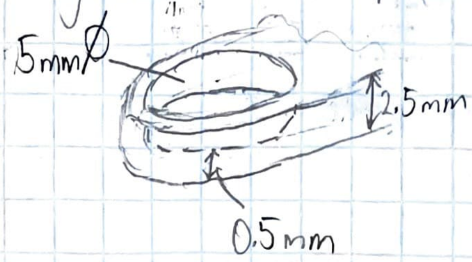
Front View



Side View

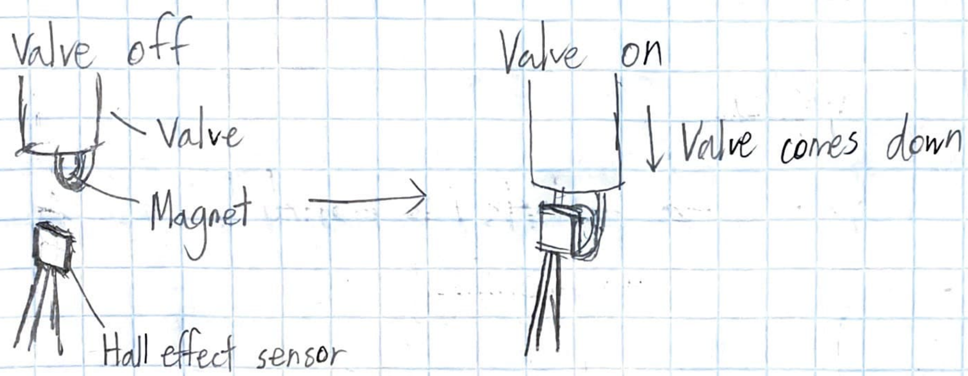


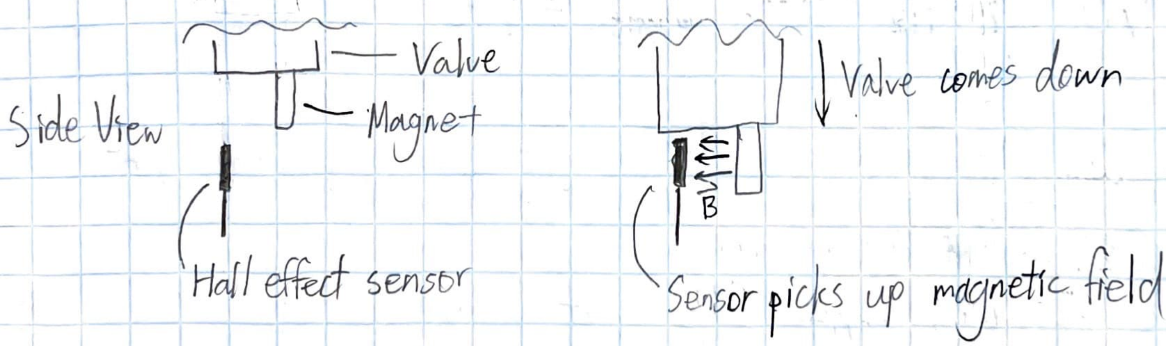
Magnet Hole Detailed View



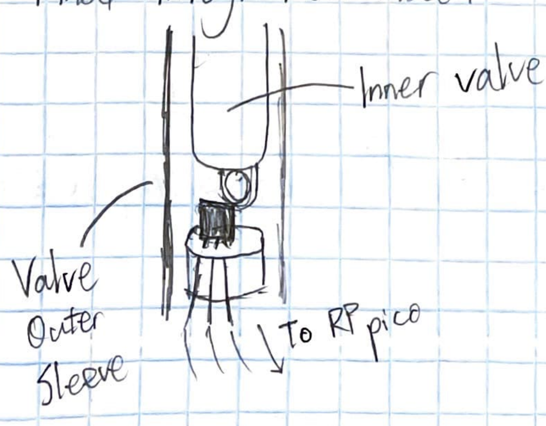
* Extend outer valve sleeve by 1 cm to cover hall effect sensor

Concept Idea

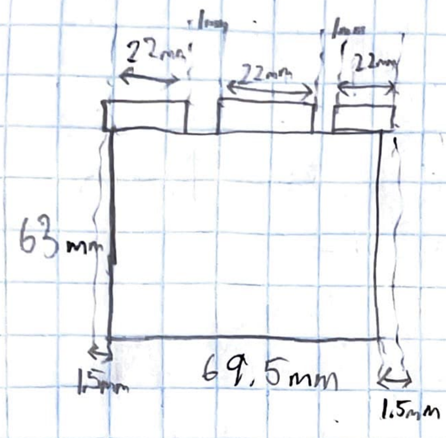




Final Integration Model



Valve Module Dimensions 5/25/25



FFT Filtering 6/16/25

Previously:

- Filtered with frequency (ignore $f < 100$ Hz)
- Inconsistent (changing FFT samples/frequency also changes frequency of the noise)

ex,

256 samples 4500Hz → 128 samples 4500 Hz

$f_{noise} = 70 \sim 85$ Hz $\longrightarrow$ $f_{noise} = 145 \sim 170$ Hz

- Previously (256 samples) had no problem, lowest note on trumpet is about 150 Hz
- Now ($\underbrace{\text{128 samples}}_{\text{make FFT faster}}$), can't filter out noise as $f_{noise} \approx 150$ Hz

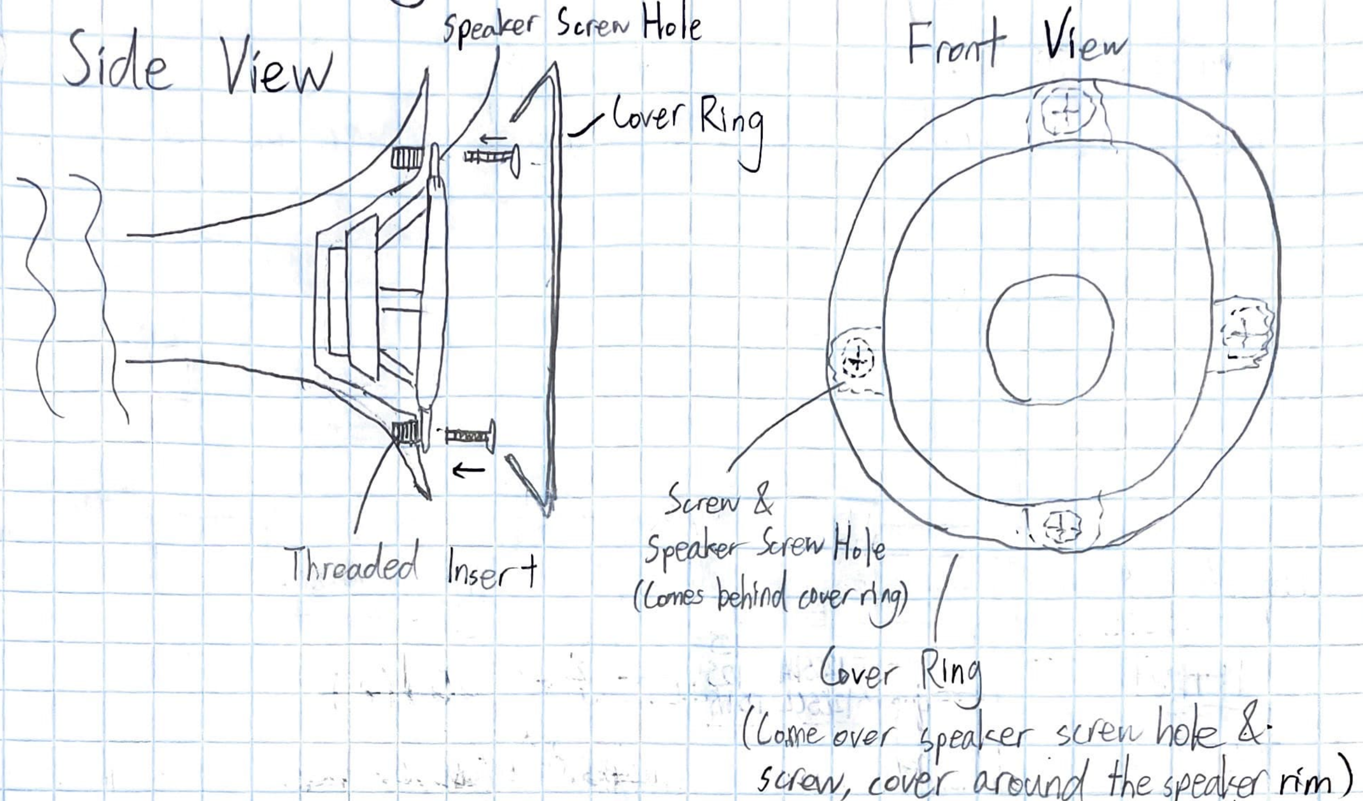
Instead:

Filter with amplitude

- Throw out readings with amplitude < 0.42 (for 64 samples)

Speaker Mounting 6/26/25

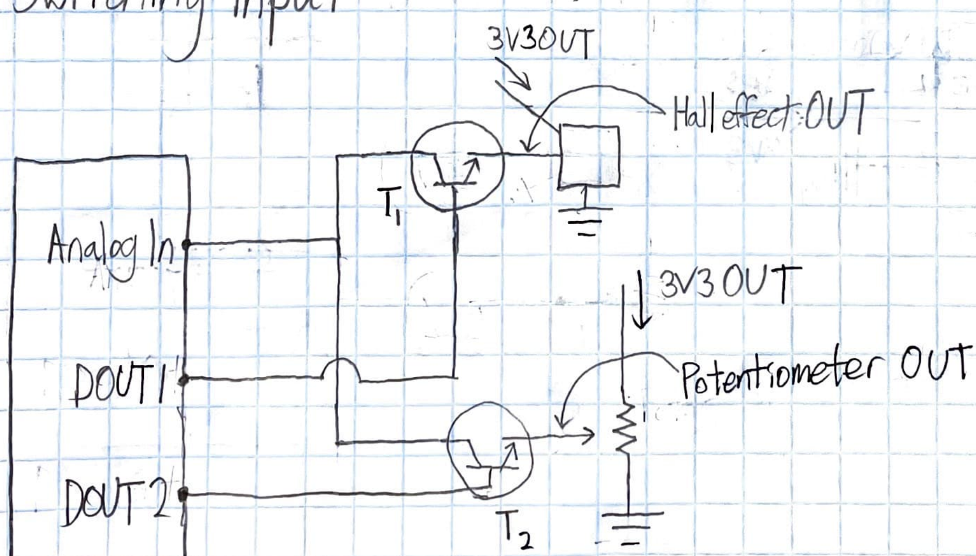
- Side View / Front View



- ADC - 2 Sensors, 1 GPIO 7/5/25

Problem - 4 analog inputs, only 3 ADC GPIO on RP Pico 2

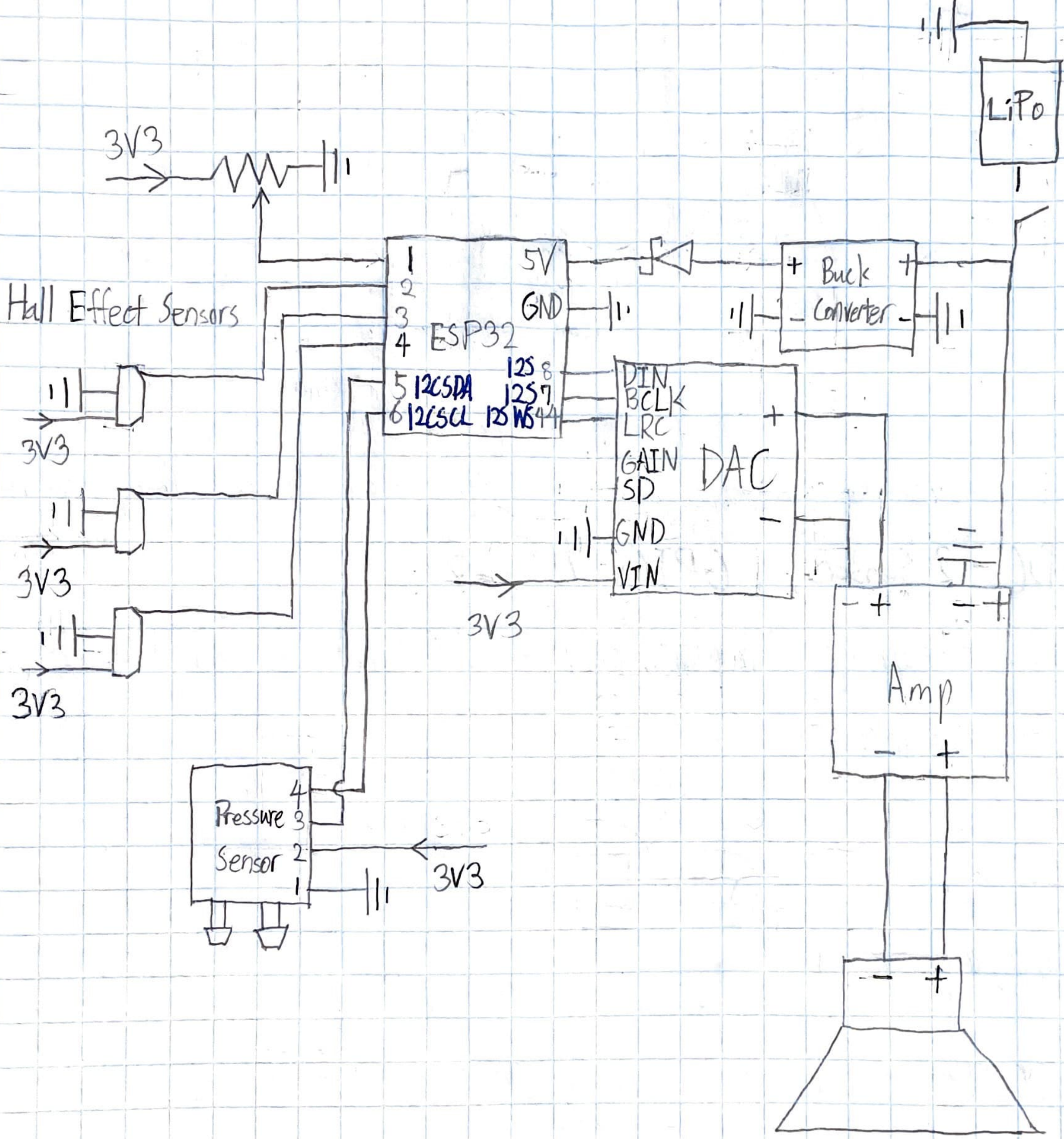
Switching input



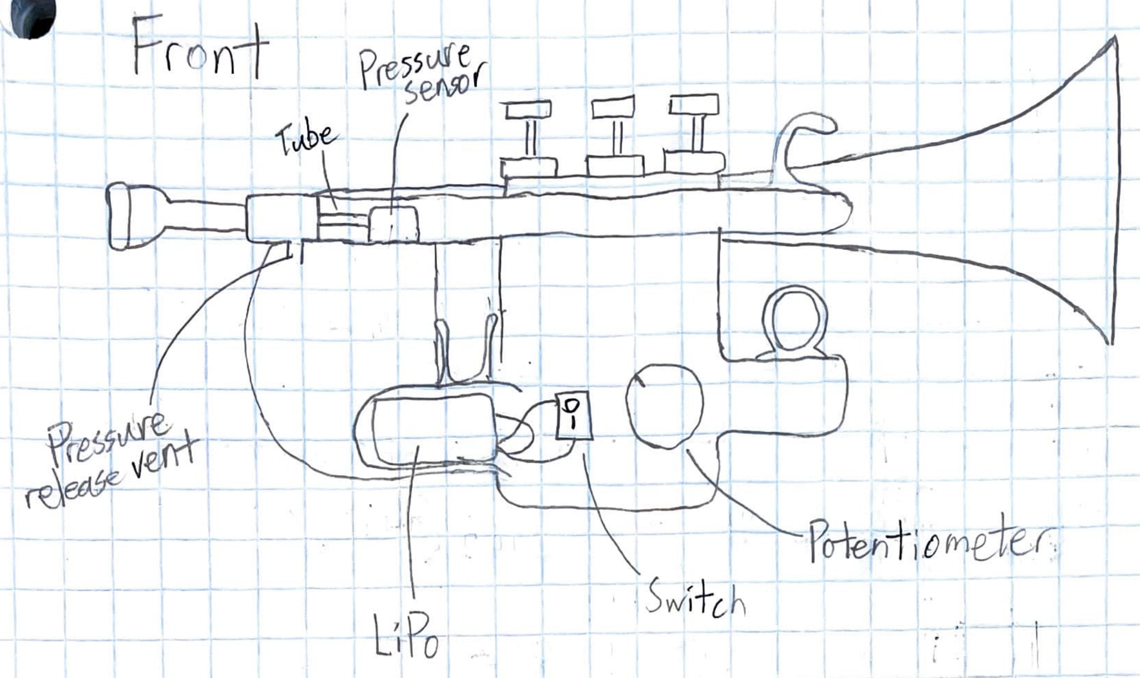
- *DOUT1 & DOUT2 alternate on and off, switching T_1 and T_2 on/off. Analog In receives inputs alternating between hall effect sensor & potentiometer.

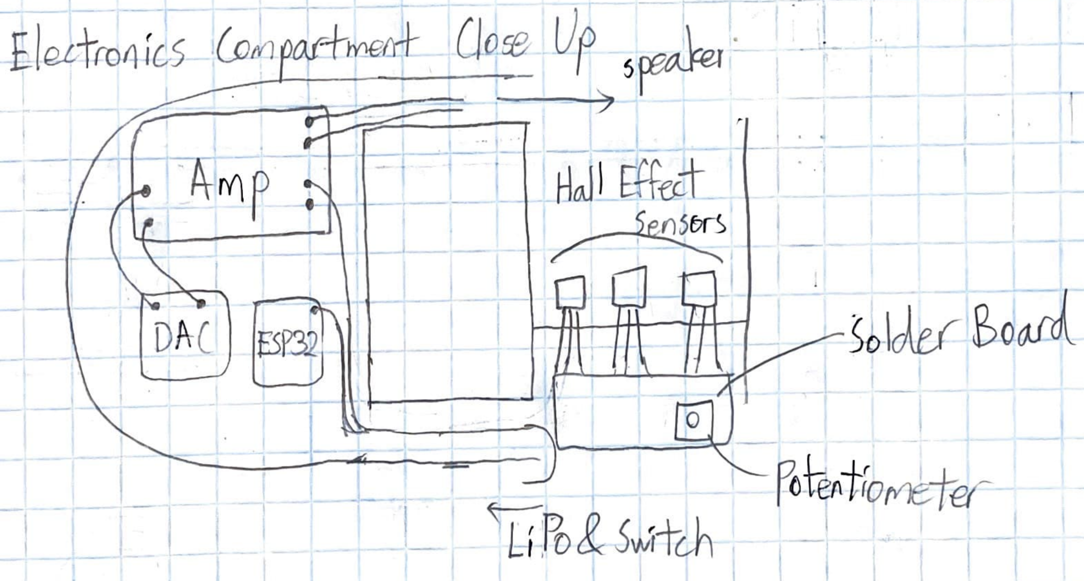
Final Circuit Design 7/11/25

- Switched to ESP32
- Faster, more analog GPIO's → no need for multiplexing



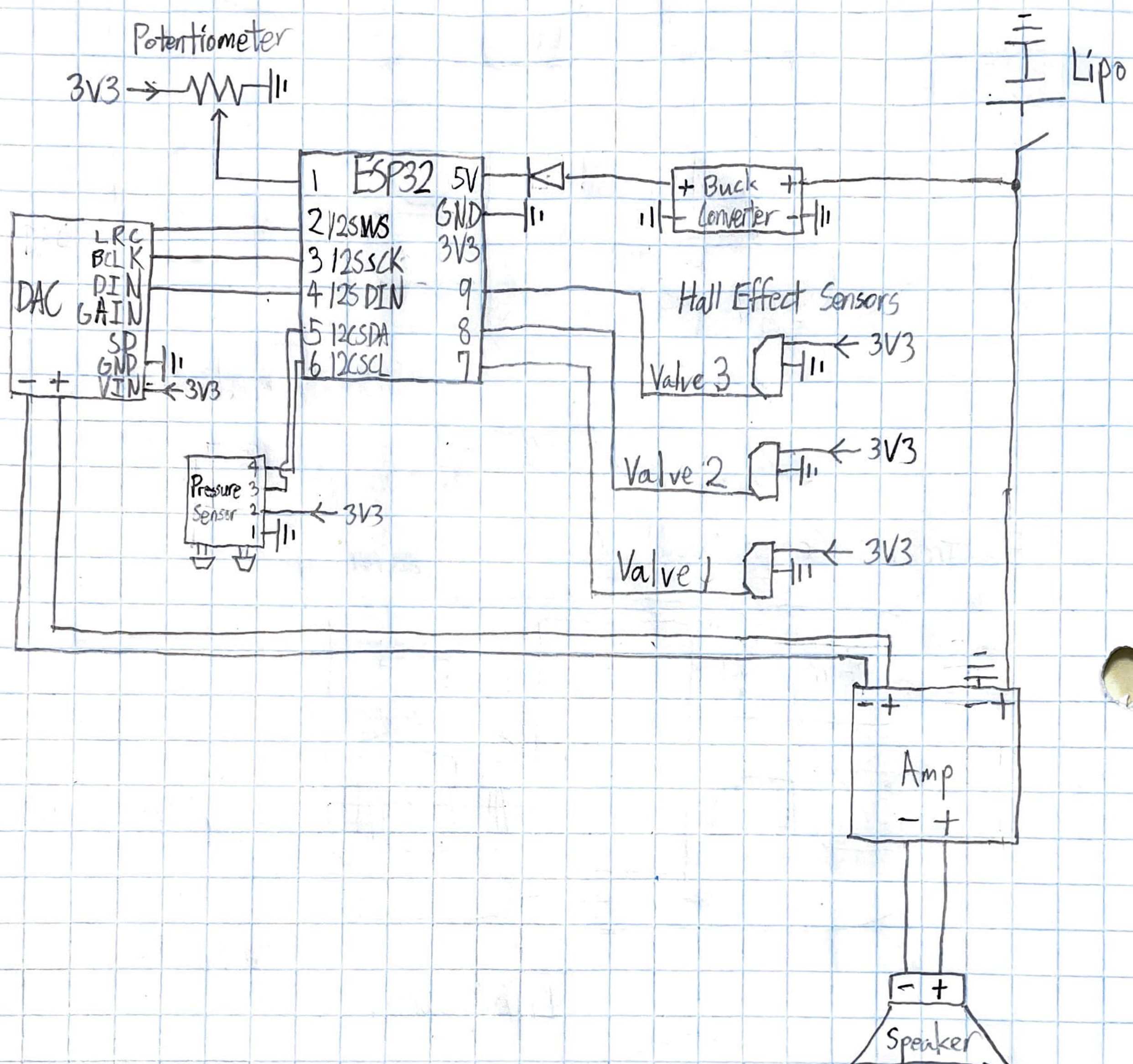
Chasis Idea 7/11/25



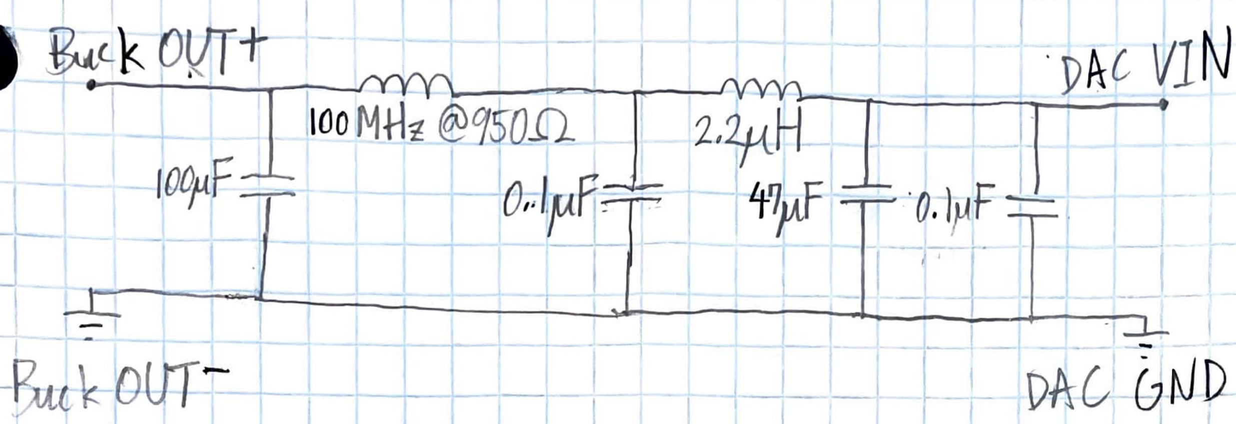


Circuit Design V2 9/7/25

* Updated GPIO for better wire management



Buck Converter Filter 11/9/25



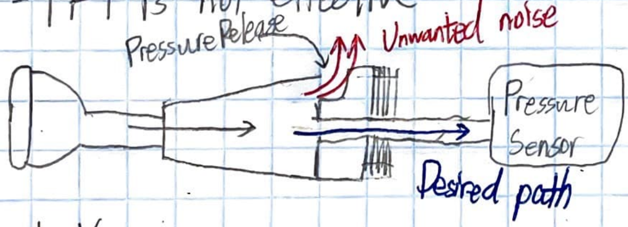
LC Frequency Cutoff

$$f_c = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{2.2\mu H \cdot 47\mu F}} \approx 24\,kHz$$

Ferrite Bead - provides high frequency impedance, blocks EMI

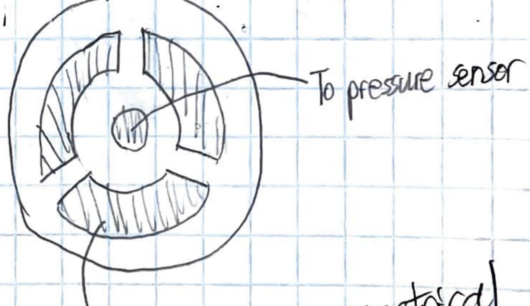
Mouthpiece Receiver Redesign 11/18/25

- Currently, airflow is disrupted
- FFT is not effective



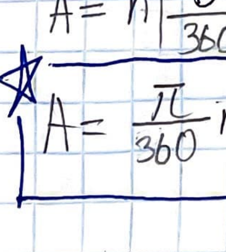
Side View

New
Top (Enlarged) *Dashed lines represent holes



Pressure Release Vents - symmetrical

Pressure Vent Area Derivation



n - number of vents (in this case, $n = 4$)

θ - in degrees ($n\theta < 360$)

$$A = n\left[\frac{\theta}{360}\pi r_1^2 - \frac{\theta}{360}\pi r_2^2\right]$$

$$\boxed{A = \frac{\pi}{360} n\theta \left(r_1^2 - r_2^2\right)}$$

Isometric