Index

1. Introduction and Goals
    1.1 Introduction
    1.2 Key restrictions
2. Existing speakers and specifications
    2.1 Metric definitions
3. Interpreting Data
4. Benchmark Targets
5. Non-Audio Targets + Rough feature ideas
6. Measuring specifications
    6.1 Purpose and Feasibility
7. Speaker Design
    7.1 Cabinet options
        7.1.1 The sealed box
        7.1.2 The Vented Enclosure
        7.1.3 The Passive Radiator
        7.1.4 Transmission Line
8. Enclosure shape and internal dampening
    8.1 Enclosure Shape Considerations
    8.2 Chosen shape
    8.3 Dampening
9. Downward-Firing Passive Radiator and Boundary Loading
    9.1 Rationale
    9.2 Planned experiment
10. Target size and volume
    10.1 External volume target
    10.2 Internal volume
11. Driver and Passive Radiator Selection
    11.1 Driver parameters (Thiele small)
    11.2 Driver types
    11.3 Crossover Considerations
12. Driver choice

12.1 Final decision
12.2 Dimensions for modelling and full driver specs
12.3 Driver revision

13. Modelling
13.1 Design A Process
13.2 Design A Final Dimensions
13.3 Design B Process
13.4 Design B Final Dimensions
13.5 Enclosure Material and 3D printing

14. Internal Electronics
14.1 Digital Sound Processing
14.2 Bluetooth Receiver
14.3 Amplifier Choice
14.4 Power Arrangement
14.5 Wiring plan

15. DSP Programming
15.1 Code Snippets
15.2 Building the final sketch

16. The build
16.1 Picking up the 3D Printed Enclosure
16.2 Prototype Assembly and Physical Fit
16.2.1 Electronics Assembly
16.2.2 Power Supply Revision and USB-C Issue
16.2.3 How I would have tested the fault
16.2.4 Power Solutions Moving Forward

17. What I Was Not Able to Complete
17.1 Shortcomings

18. Future Development Plan

19. Final Reflection and Conclusion

20. References

Introduction and Goals:

Introduction:

For this project I would like to design a compact portable loudspeaker that applies speaker design theory to achieve a high perceived audio quality relative to its size, while also exploring one or more novel design ideas that may improve typical loudspeaker attributes such as bass extension, usability, or even room interaction. The targets for this loudspeaker will be derived by three main areas: the performance of comparable commercially available speakers from leading brands, the physical constraints of size, materials, and budget, and published measurement methods used for loudspeaker benchmarking.

I am keeping these targets and my project vision somewhat flexible, as certain goals or design choices may change throughout the project in response to further research and restraints. This allows the project to stay grounded and realistic, giving room for adjustment if a particular idea proves less effective than expected, while keeping the overall aim of the design in focus.

Key restrictions

1. The speaker must remain relatively compact and portable.
2. It must be battery powered and rechargeable by USB-C.
3. It must support Bluetooth connectivity.
4. It must have sufficient internal volume to support internal components without becoming too large
5. It must be viable to 3D print the enclosure and straightforwardly assemble the internals.
6. There should be at least one novel idea
7. The controls must remain simple, robust, and intuitive for casual use.

Existing speakers and specifications:

Metric definitions:

- SPL_MAX_dB - The loudest the speaker can get at maximum volume (RTINGS, 2023a)
- DRC_MAX_dB - How the frequency response differs at max volume from a previously measured frequency response at a baseline value (76 dB) This measurement signifies dynamic range compression showing how much the audio is squeezed when reaching max volume. A lower value is arguably better for “true” sound, but a higher value may give a perceived louder sound at max volume subjectively. (RTINGS, 2023a)
- LF_Ext_Hz – The lowest frequency at which the frequency response reaches –6dB of the target response showing low-frequency extension. Lower is better. (RTINGS, 2023b)
- HF_Ext_kHz - The highest frequency at which the frequency response reaches –3DB of the target response showing high-frequency extension. Higher is better. (RTINGS, 2023b)
- Directivity Index (dB) - This measures the directivity of the speaker. In other words, how the sound changes at different angles around the perpendicular axis of the speaker. A higher value suggests a more focused “beam” of audio whereas a lower value suggests a wider spread and more uniform omnidirectional sound. A certain value isn't necessarily worse than another value as there are different use cases for different speakers. (RTINGS, 2025a)

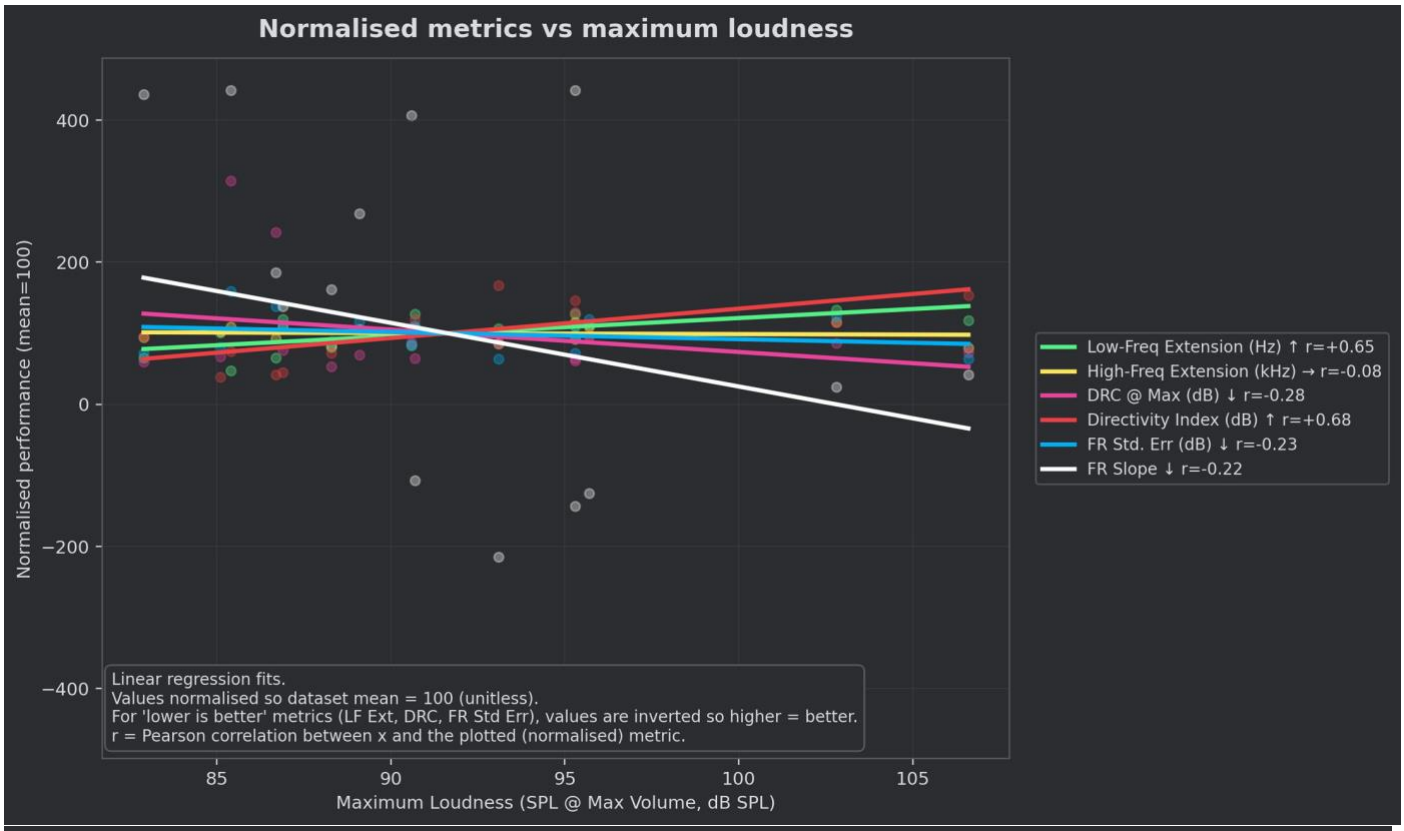
- Frequency response Slope - The derivative (gradient) of the regression line for the frequency response showing the tilt between low-frequency and high-frequency. A negative value suggests a greater bass presence in reference to the overall frequency response and a positive value suggests the opposite. The closer the value is to 0, the more probable it is that the frequency response is balanced. (RTINGS, 2023b)
- Frequency Response Std. Err. (dB) - This is the standard error in the frequency response i.e. a way to quantise deviation from the regression line. A lower value means the frequency response is more uniform whereas a higher value means a more sporadic frequency response. (RTINGS, 2023b)

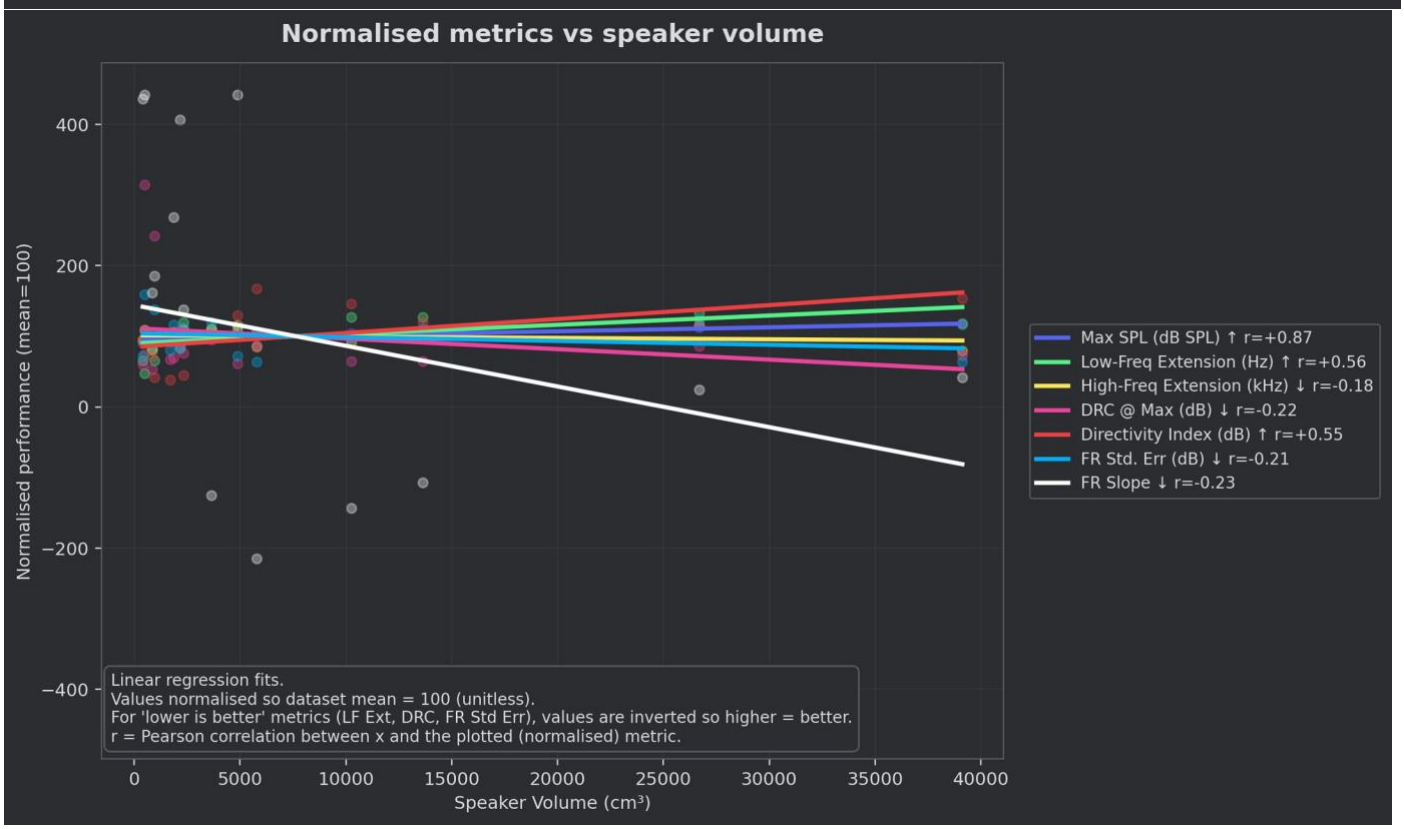
Below are some metrics from commercially available speakers that cover a range of sizes and budgets. All data was taken from RTINGS.com which is a third-party company that extensively tests a wide range of products. This data was collected for the purpose of gauging what average metrics commercially available speakers were presenting and to see the correlation between any metrics which may indicate certain trade-offs I may have to consider. There are only 15 speakers listed so any correlation should be taken with a grain of salt, but they generally follow expected patterns. For the scatter graphs all values were normalised around a mean value of 100 and regression lines were added for ease of viewing. (RTINGS, 2022; RTINGS, 2023a; RTINGS, 2023b; RTINGS, 2024a; RTINGS, 2024b; RTINGS, 2024c; RTINGS, 2024d; RTINGS, 2024e; RTINGS, 2024f; RTINGS, 2024g; RTINGS, 2025a; RTINGS, 2025b; RTINGS, 2025c; RTINGS, 2025d; RTINGS, 2025e; RTINGS, 2025f; RTINGS, 2026a; RTINGS, n.d.)

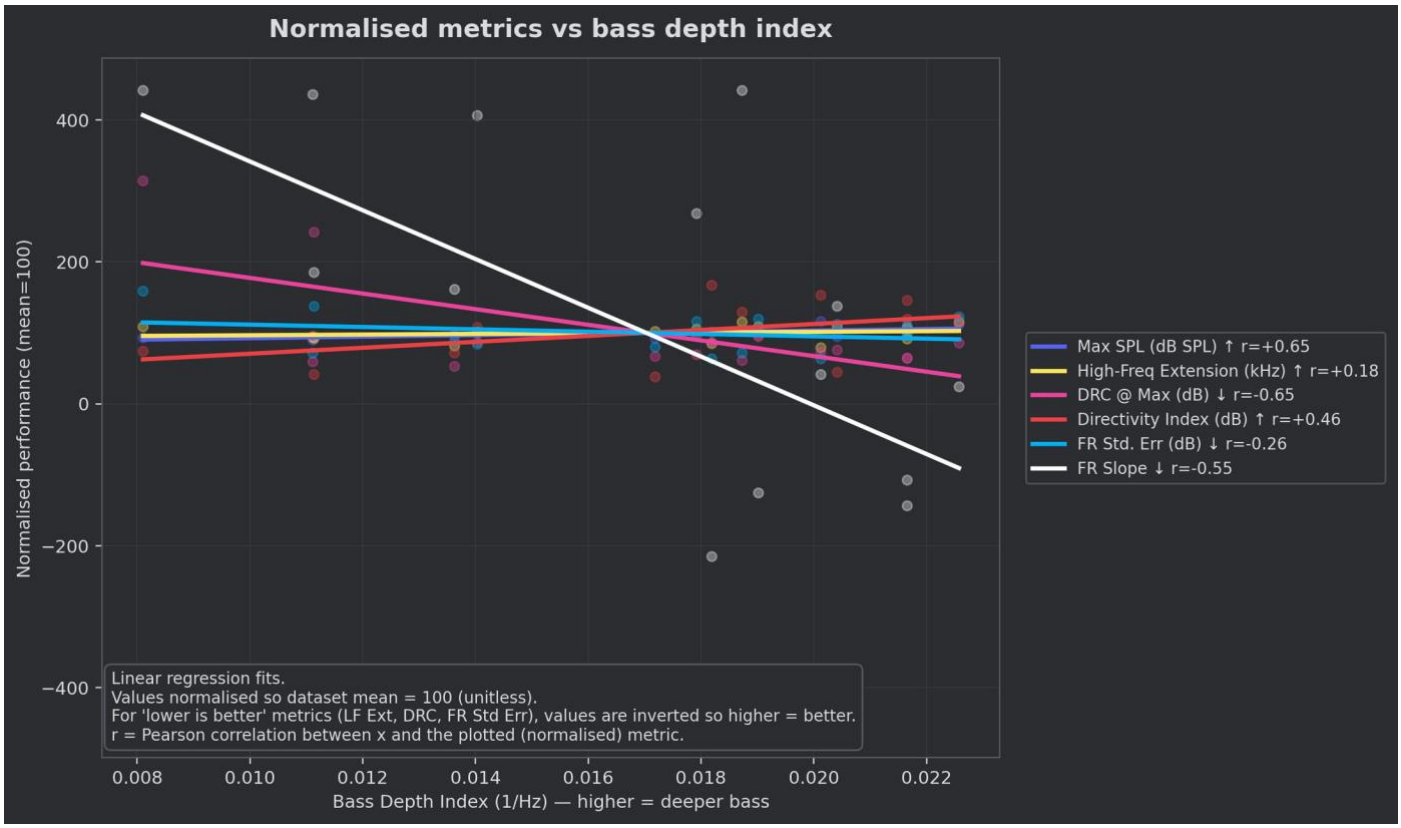
Model	Volume _cm3	Volume _L	Weight_kg	SPL_Max_dB	DRC_Max_dB	LF_Ext_Hz	HF_Ext_kHz	Directivity_Index	Stereo	Battery_Life_h	Charge_Time_h	FR_Slope	FR_Std Err_dB
Tribit StormBox Micro 2	400	0.4	0.3	82.9	4.07	90.0	16.3	4.16	No (mono)	8.1	2.5	0.73	4.37
Anker Soundcore Select 4 Go	484	0.484	0.3	85.4	0.77	123.4	18.8	3.21	No (mono)	10.4	3.9	0.74	1.99
JBL Flip 7	834	0.834	0.6	88.3	4.56	73.4	14.1	3.13	No (mono)	5.6	2.8	0.27	3.58
UE Wonderboom 4	944	0.944	0.4	86.7	1.0	89.8	16.0	1.79	Yes	13.1	2.5	0.31	2.29
UE Megaboom 4	1703	1.703	0.9	85.1	3.63	58.2	17.7	1.67	Yes	20.0	2.3	-0.76	3.96
JBL Charge 6	1848	1.848	1.0	89.1	3.5	55.8	18.2	4.17	No (mono)	13.7	2.5	0.45	2.72
LG XBOOM Bounce	2167	2.167	1.4	90.6	2.79	71.3	17.5	4.67	Yes	25.6	3.4	0.68	3.74
Marshall Middleton II	2315	2.315	1.8	86.9	3.19	49.0	18.8	1.95	Yes	10.8	2.7	0.23	2.82
Bose SoundLink Max	3644	3.644	2.2	95.7	2.55	52.6	18.8	4.23	Yes	15.5	3.7	-0.21	2.65
JBL Xtreme 4	4876	4.876	2.1	95.3	3.95	53.4	19.9	5.59	Yes	27.2	3.1	0.74	4.41
Anker Soundcore Motion	5775	5.775	1.5	93.1	2.78	55.0	14.7	7.24	Yes	22.0	6.1	-0.36	4.95

Model	Volume _cm3	Volume _L	Weight_kg	SPL_Max_dB	DRC_Max_dB	LF_Ext_Hz	HF_Ext kHz	Directivity Index	Stereo	Battery_Life_h	Charge_Time_h	FR_S lope	FR_Std Err_dB
EarFun UBOOM X	10260	10.26	4.1	95.3	3.77	46.2	15.8	6.31	Yes	39.3	3.5	-0.24	3.25
Harman/ Kardon Onyx Studio 8	13640	13.64	3.6	90.7	3.77	46.2	18.5	5.17	Yes	6.7	3.3	-0.18	2.85
JBL Authentic 500	26700	26.7	8.0	102.8	2.81	44.3	19.9	4.99	Yes			0.04	2.58
SOUNDB OKS Go	39124	39.124	9.7	106.6	3.34	49.7	13.7	6.64	No (mono)	35.7	2.1	0.07	4.99

Interpreting Data:






Normalised metrics vs bass depth index
Normalised performance (mean=100)
400
200
0
−200
−400
0.008
0.010
0.012
0.014
0.016
0.018
0.020
0.022
Bass Depth Index (1/Hz) — higher = deeper bass
Linear regression fits.
Values normalised so dataset mean = 100 (unitless).
For 'lower is better' metrics (LF Ext, DRC, FR Std Err), values are inverted so higher = better.
r = Pearson correlation between x and the plotted (normalised) metric.
Max SPL (dB SPL) ↑ r=+0.65
High-Freq Extension (kHz) ↑ r=+0.18
DRC @ Max (dB) ↓ r=-0.65
Directivity Index (dB) ↑ r=+0.46
FR Std. Err (dB) ↓ r=-0.26
FR Slope ↓ r=-0.55

There are a few correlations here that can be spoken about, an intuitive one being the positive correlation between speaker size and max SPL with a strong positive correlation value of +0.87 where +1 indicates a perfect positive correlation. It is hard to see as the scale of which we measure sound is logarithmic so adifference of 10dB, while roughly 2 times as loud, looks small in a graph with linear value ranges. Arguably the most important takeaway from these values is that bass extension gets better with size fast. The smaller the speaker is, the less bass presence. Since this is the main challenge, this will be my primary focus. These two metrics seem to be the primary trade-offs as we make a speaker smaller.
A notable speaker is the Marshall Middleton II with a volume of 2315c m 3 and a bass extension of 49Hz. It has two 3-inch woofers, two 0.6-inch tweeters and 2 passive radiators. The bass will be from a combination of the driver and DSP choices plus those radiators. I will investigate this in more depth further into designing and planning.

Benchmark Targets:

Given the data I collected, below is a table of values I would be happy with from my speaker:

Metric	Target
External Volume (gross)	≤ 2400 cm^3
Weight	≤ 3.0 kg
Max SPL	≥ 80 dB SPL
DRC @ Max	≤ 4.0 dB
Low-Frequency Extension	≤ 60 Hz
Directivity Index	≤ 4.0 dB (lenient)
Stereo	Mono (optional virtual stereo)
Battery Life	≥ 8 hours
Charge Time	≤ 3.5 hours
FR Slope	No fixed target (if LF extension met)
FR Std. Err	No fixed target (if LF extension met)
Idle Noise / Hiss	No audible hiss @ 0.5 m (quiet room)
Distortion	THD ≤ 10% (100 Hz–10 kHz) @ 76 dB/1 m
Artefacts	No rub/buzz/chuff up to Max SPL target

Non-Audio Targets + Rough feature ideas:

I want a sleek, robust feel to the speaker. The buttons should be secure and resistant to light taps. There will be a charging port that is USB-C. This speaker will be Bluetooth but also have an Aux input. The base should be sturdy and grip whatever surface it is on. The tangible user interface should be eloquent and intuitive using labels or more preferably button shape. Room tone detection and EQ adjustment calculate by a startup sound when you turn the speaker on. Manual EQ adjustment post room detection EQ (rough low-end, high-end boosts/ reductions) Potential add-ons to the speaker to increase the low-frequency extension. Real time pitch shifting via buttons (middle button to reset)

Measuring specifications:

Purpose and Feasibility:

Benchmark measurements are a way of assessing the capabilities of a design by measuring its performance against existing designs that use the same testing standards. There are many published standards for measuring a speaker's capability. As I used RTINGS.com in my research into the correlation between benchmark values, it would be helpful to follow the same procedure so I can directly compare my speaker to other speakers. Measurements taken outside of this are still useful within this project because I will still be able to define and contextualise this data. Any other measurements taken will also follow a published standard to ensure accuracy and to give my values more credibility. (RTINGS, 2025a; Eargle, 2003, pp. 325-340)

Speaker Design:

Cabinet options:

The sealed box:

Lower wavelengths are long so if we had a speaker with an open back, the waves from the rear of the driver could wrap around the front and side panels and somewhat cancel out the waves from the front of the driver leading to phase cancelation. Sealing the box solves this by essentially trapping the rear waves. In turn this creates pressure within the box meaning the driver must work harder to overcome the pressure changes. However, this introduces a level of control so the driver doesn't overshoot and can always predictably return to its idle state (Dickason, 2006, pp. 29–31).

Pros:

- Closed box is the simplest enclosure type due to there being comparably fewer sensitive factors to consider and the ease of designing and building (Dickason, 2006, p. 29).
- Predictable response controlled by enclosure size (Fc-system resonant frequency) and the level of damping (Qtc- Quality factor, system damping) (Dickason, 2006, pp. 29–30)
- No port chuff or port resonance which are components in other cabinet options
- The driver is less likely to unload (excessive excursion) below Fc

Cons:

- Less bass output than other designs for the same enclosure size
- The pneumatic pseudo spring forces the driver to work harder at lower frequencies due to Hooke's law F=Kx where F is force, K is stiffness (increased due to extra "spring"), and x is displacement (higher at lower frequencies) (Dickason, 2006, p. 29)

The Vented Enclosure:

The vented enclosure is when we take a sealed box and add a duct (port) that is tuned to a specific frequency. The air inside this port acts like a moving mass that resonates the most around the tuned frequency. This boosts a band around that frequency by acting like a second driver controlled by the pressure changes within the box. Dickason describes it as analogous to a 24 dB/oct high-pass system (Dickason, 2006, p. 61).

Pros:

- Higher LF output/efficiency around tuning than sealed for similar box size
- Lower cone excursion near port frequency meaning higher power handling (Dickason, 2006, p. 61)

Cons:

- High excursion below tuning, Dickason warns vented designs can have high excursion at frequencies
- substantially below resonance, making them sensitive to subsonic content unless filtered. This is
- counter-intuitive because you would think high excursion means louder bass, but below the
- resonance frequency there is low internal pressure so the excursion can cause damage to the driver.
- The air can freely flow out of the port so there is more likely to be phase issues also.
- More parameters to consider than a sealed box making it harder to build physically. Essentially there
- are more chances for things to go wrong (Dickason, 2006, p. 61).

The Passive Radiator:

A passive radiator design is like the vented enclosure but instead of having a lump of air that is acting as a mass, we do have a physical mass in the form of a passive diaphragm. This is not supplied with any power and is solely there to react to the air pressure within the enclosure. At lower frequencies the passive radiator shares acoustic load with the driver reducing over-excursion (Dickason, 2006, p. 85).

Pros:

- Avoids vent artefacts: Dickason lists elimination of vent coloration (resonant sounds), wind noise,
- and internal HF leakage through the vent.
- More practical for small enclosures where the required vent length would exceed internal
- dimensions.
- Easier to fit components inside compared to a long duct (Dickason, 2006, p. 85).

Cons:

- Less transient stability, physical mass has inertia, takes time to start moving.
- Adds to the speaker cost
- PR excursion limit can become the new bass bottleneck.
- Steeper cutoff at resonant frequency (Dickason, 2006, p. 85).

Transmission Line:

While this is worth mentioning, I think that it is beyond my current capabilities due to the space the line will occupy and the added complexity in design especially as I'm planning to 3D print the enclosure. It works by using a long (quarter of the target frequency) internal path that gives the rear radiation enough length so it can be in phase with the front radiation. These require significant damping due to the line being a strong resonator of higher frequencies. (Dickason, 2006, p. 95)

For my project, a passive radiator design is most suitable. It eliminates vent artefacts like turbulence and air noise and reduces the complexity of the design by reducing very sensitive parameters, while being able to deliver great results. My speaker is not designed to be used in a setting where audio needs to be completely accurate, so the downsides are much less applicable than the downsides of say a ported design. What I mean by this is that his speaker is designed for music that is already mixed and mastered to a certain standard.

Enclosure shape and internal dampening:

Enclosure Shape Considerations:

The next decision that affects both the sound and the practicality of building it is the physical shape of the cabinet and how I treat the inside of it. The shape is not just for aesthetics. The cabinet shape is part of the acoustic system, so the surfaces and curves and corners affect what happens outside the box and what happens inside the box. Dickason discusses enclosure shape as a measurable factor in the final response because edge geometry and cabinet form change the way the speaker radiates and how smooth or rough the response ends up being (Dickason, 2006, p. 105)

My upper limit for the enclosure is 2400c m 3 but this doesn't consider the internal component volume which I will have to subtract. I will also have to subtract wall thickness from the external volume to get my enclosure internal volume which is what will be used to calculate the resonant frequency and in turn to tune the passive radiator.

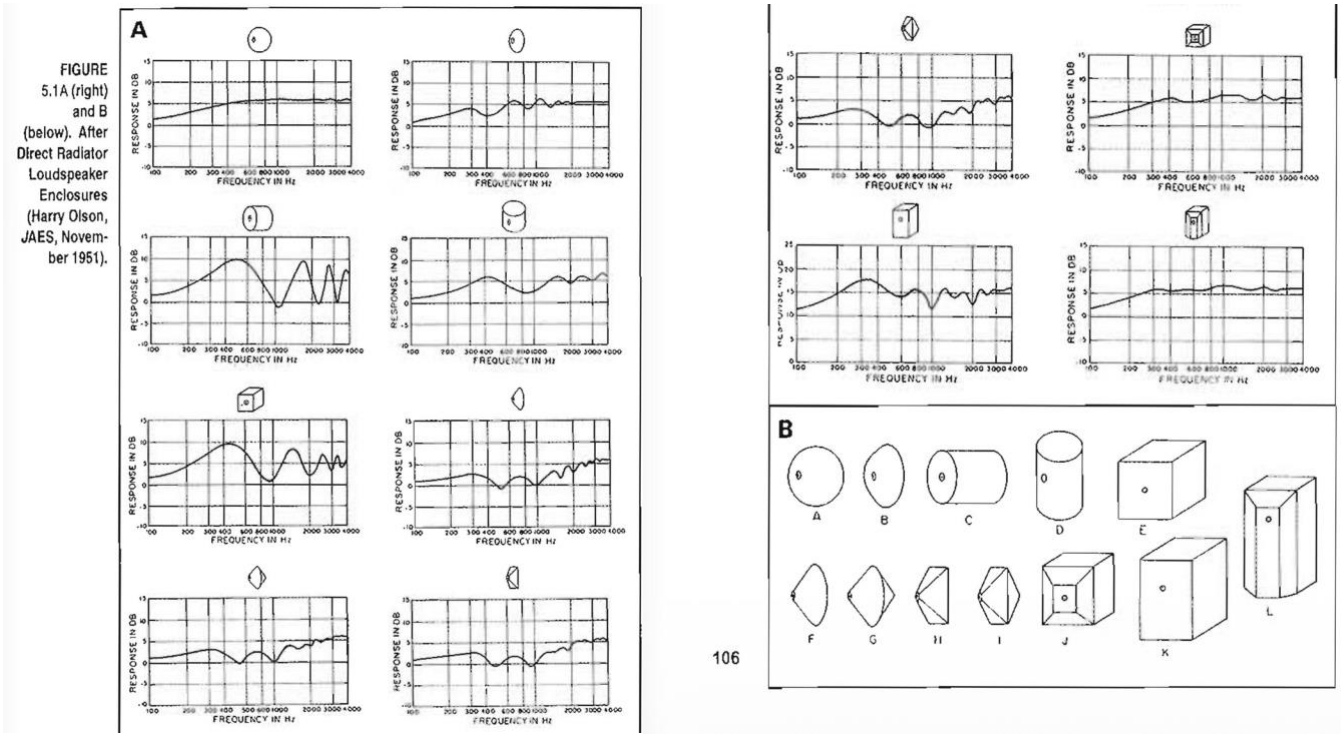


FIGURE 5.1A (right) and B (below). After Direct Radiator Loudspeaker Enclosures (Harry Olson, JAES, November 1951).

106

Dickason references research by Olson where different enclosure shapes are compared using the same driver and measurement approach, and the results show a clear ranking. The sphere comes out best, then the egg shaped and cylindrical forms are close behind, beveled edges reduce the problem on rectangular boxes, but he notes there needs to be significant beveling, and simple cubes and basic rectangular forms show larger variations. It’s important to note that exact values may vary depending on enclosure size, so the overall trend is what carries the significance. (Dickason, 2006, pp. 105-107).

A spherical enclosure is impractical for me, driver mounting becomes problematic as well as creating a stable base. Rounded surfaces tend to be better because inside an enclosure, sound waves can bounce around, and with predictable flat surfaces and especially parallel walls, those reflections line up in a repeatable way. That repeatability is what lets standing waves build up at certain frequencies, so the box starts acting like a little resonant chamber instead of just a pneumatic spring. The problem is that the energy does not just stay inside, it can leak back out through the cone and through the enclosure surface via transmission. It can also interact with the passive radiator, which can add a boxy sound to the midrange. A curved or oval internal shape is less likely to have one dominant repeating distance between surfaces due to constant angle changes, so the reflections are less likely to reinforce the same frequencies as strongly. It does not fully remove resonances, but it makes them easier to control.

Chosen shape:

Based on the shape evidence and the impracticality of a sphere, I'm thinking an attractive shape to look more into is a half ellipsoid or an egg shape with the bottom cut off. It has the curved walls great for minimizing problematic reflections, and it has a good volume to surface area ratio meaning it can be physically smaller while having a suitable internal volume for a good bass response and all internal components. The flat base from it being a half ellipsoid gives it a sturdy base. While I could put the speaker on small feet to hold up a full ellipsoid, I want to investigate the proximity effect that come with bottom mounting the passive radiator, in which case the speaker will still have to be supported by feet. This trapped air increases the acoustic mass the radiator is working against, lowering its resonant frequency. There is an issue with the sound being dependent on the surface the speaker is on, to tackle this I could mount a false bottom on the speaker so I can control the exact distance from the passive radiator. While a curved surface does not fully remove resonances, it makes them easier to control, and with dampening we can get close to removing them entirely.

Dampening:

Dampening works by absorbing energy inside the enclosure reducing reflections. The standard materials used for damping are fibreglass-based insulation, polyester fibre fill, long fibre wool type stuffing, felt type linings, and open cell acoustic foam lining. I can easily work out the internal volume decrease from the lining, but this is trickier for the looser damping material which will be kept away from the internal component with a net. The easiest way is to treat it as negligible for now because it creates a looser pneumatic spring which is a characteristic of a larger enclosure. These fine tunings will have to be done with trial and error when taking frequency response measurements.
(Dickason, 2006, pp. 111-115)

Downward-Firing Passive Radiator and Boundary Loading:

Rationale:

I’m planning on using a bottom mounted passive radiator for this project for a few reasons, due to the shape I have chosen the bottom gives me a nice flat surface to mount the radiator, also while printing this leaves a big non-awkward hole where I can put the internals inside easily, and finally I would like to explore the effect that boundary loading has on a portable speaker. The idea is that the sound waves firing downwards onto a surface reflect and reinforce the low-frequency waves that are already travelling to the user's ear. This relies heavily on the surface it is placed on so I'm proposing a false bottom on the speaker where I can a) control the gap distance as to not put too much stress on the passive radiator, and b) control the surface that will be reflecting the waves. This will be a secondary component that I will attach after the first speaker is complete just for ease of assembly and ease of refining the distance if needed be.
The usual idea of a transmission line is that the waves firing backwards travel through a tube and then reinforce the forward firing waves, and the distance of the tube is exactly ¼ of the specific frequency you are trying to boost and this boost acts over a bell curve around that frequency. Lower wavelengths are very long so my idea isn't trying to mimic reinforcement in the transmission line way unless I used very large legs, but then most of the low end sound pressure would have most likely been lost to diffraction and diffusion before it could bounce back up to reinforce in a meaningful way. The idea is that the gap won't be big enough for most low frequencies to become out of phase fully giving partial boosts across a broad range of low end. Since the passive radiator won't react well to high frequencies anyway, it won't cause issues for the high end by the time the distance between the false bottom and the radiator lines up exactly to cause phase issues. For example, if the gap is 2cm then this could cause issues for a frequency band around 4287 HZ (Not exact and maybe even a few cm’s out but let's say it's a 4cm travel distance from radiator to driver,

and then to be out of phase that distance would have to be half of the problem wavelength so a wavelength of 8cm) which the passive radiator wouldn't be reacting to anyway.

Planned experiment:

Assembling the main speaker body first allows me to run experiments on the best distance between the passive radiator and the false bottom without committing to a final design. To do this I will play a frequency sweep through the speaker and roughly record the EQ profile at planned varying heights (1cm, 2cm, 3cm etc) from a solid surface, ideally one like the material the false bottom will be made from. Once the distance has been refined, If the timeframe left allows it I can simply model and print out the second component and attach it onto the speaker.

Target size and volume:

External volume target:

Since playing around with modelling, I've decided a more suitable maximum external volume is around 3000 cm^3. This keeps the speaker within a size range that still feels genuinely portable but still on the larger size for better bass response. Looking at the benchmark data I collected, this seems to be around the point where some speakers start to gain a more convincing bass response without becoming too large to comfortably carry.

Internal volume:

This is what matters for the acoustics. With all the internals + wall thickness, it’s reasonable to say the internal volume will be around 1.6 litres. When this is modelled in cad I will have a better general idea but for now any tuning decisions should be based off the expected internal volume rather than the external.

Driver and Passive Radiator Selection:

Driver parameters (Thiele small):

I feel it's important to note that most of these parameters are derived from a fundamental set of parameters but are useful in deciding which driver suits my project. It’s a way of analysing how changing the fundamental property balance affects a range of characteristics in sometimes unintuitive ways. A practical example of this is if you increase the motor strength to get higher damping, you might think this would increase the amount of air the cone will be able to move but it ends up fighting the drivers low end natural response reducing bass output.

Fs-

This is the driver’s free air resonance frequency. In simple terms, it gives an idea of how low the driver naturally wants to go. For this project, a lower Fs is preferred, because the main challenge in a small speaker is getting a good bass output. (Hill, 2019, pp. 45-50)

Qts-

This is the overall damping of the driver; it strongly affects how the driver will behave in a specific enclosure. Lower damping tends to be better inside of ported, medium suits sealed and high damping suits open baffle. (Hill, 2019, pp. 45-50)

Vas-

This is the specific volume of air that aligns with the driver's suspension, in general terms you can work out the ideal vas based on internal volume where a smaller volume calls for a lower V as.
(Hill, 2019, pp. 45-50)
Sd-

This is the effective cone area; it tells us how much surface area is pushing air. A larger Sd would help with bass output, but the drawback is needing a bigger enclosure to mount it on which is challenging for my portable speaker. (Hill, 2019, pp. 45-50)
Xmax-
This is essentially how far the cone can move back and forth before the audio quality declines, this is different from the mechanical max (how far it can move back and forth before breaking). For a relatively small speaker, it is unlikely I would even reach the Xmax limit on the speaker before I hit the thermal limit due to the tighter air spring inside the enclosure. This can be seen as a positive because it means the driver is protected from destroying itself, but it also means it needs to work harder to produce low frequencies.
(Hill, 2019, pp. 45-50)
Vd-
This stands for volume displacement and is thought of as the cone area x excursion. It is arguably the most important parameter for me as I'm focusing on bass extension which is an air movement issue. For example, to produce a tone one octave lower, the driver must be able to move four times as much air. (Hill, 2019, pp. 45-50)
Sensitivity-
This tells me how loud the driver is for a given input, standardly measured by dB at 2.83 V at 1 metre. Higher sensitivity seems attractive but can come with trade-offs. To achieve a higher sensitivity, manufacturers will often use lighter cones and stiffer suspension leading to a poorer bass response. (Eargle, 2003, pp. 357-360)
Le-
A higher inductance essentially means more wires around the voice coil which makes it harder for electricity to “switch directions” meaning higher frequencies become harder to reproduce. A lower inductance is desirable for good high end so is commonly found in tweeters. (Eargle, 2003, pp. 21-44)
Re and nominal impedance-
These matter for amplifier loading and how I wire multiple drivers together. For this project, 8 ohm drivers are attractive to squeeze some more power out of the amp.
The driver(s) I will be looking for are, relatively small units with preferably low Fs, manageable vas, high excursion and enough top end extensions that I am not forced into adding more drivers to compensate. I need to look for drivers that are suited to smaller enclosure rather than big boxes as my speaker will is portable. I also need to think about physical fit. With an external volume of roughly 2300 cm3 and an egg like enclosure with a flat bottom, I'm limited for size when it comes to mounting. In cad with a roughly sized egg shape and some circles I saw that 2, 3.5inch drivers were probably the maximum size I could sensibly go for if I want to have a dual stereo configuration.

Driver types:

Full range drivers:
The main advantage of a full range driver is simplicity. A two driver full range layout keeps the front of the speaker clean, leaves more internal volume for the passive radiator system and electronics, and avoids the need for an added complexity being a crossover. The downside is that there is always a trade off between bass and treble. A larger full range driver can help low end, but as the cone gets bigger it tends to become
Separate woofer and tweeter

A small woofer with a tweeter is completely doable, but I fear it adds a complexity that I will not have the time to address. It could produce better overall results if done properly, especially in the upper frequencies and off axis behaviour as sources online have stated that full range drivers are very directional. The main issue for this project is deciding whether it is worth the extra space and complexity. (Eargle, 2003, pp. 112-113)

Coaxial drivers

A coaxial driver is attractive because it combines high and low-frequency elements into one unit, so in theory it gives a compact layout and a more point source like radiation pattern. Looking online I see that there are coaxial drivers that are 3.5 inch. However, these tend to be a bit more expensive and while they save space compared to the woofer and tweeter, they will still need some crossover work unless I can find some that perform well and match my size and have a built-in crossover which tend to only be built into drivers designed for cars. (Eargle, 2003, p. 166; Hill, 2019, pp. 305-307)

Crossover Considerations:

If I use two full range drivers and one passive radiator, then there is no crossover between separate high and low-frequency drivers. In that case, the DSP is mainly being used for EQ, protection, and limiting rather than frequency band splitting. This is one of the main reasons the full range approach suits the project. It keeps the electrical side much simpler and means I can focus more on enclosure design and other aspects. It would also allow me time to play around with the DSP to see if I can introduce new or novel features. (Hill, 2019, pp. 305-307)

A 2-way design is possible in principle. That could be done either with a physical crossover or with a digital crossover. A digital crossover is possible on Teensy, because PJRC's Audio Adaptor supports 16-bit, 44.1 kHz stereo audio, and PJRC also states that "more advanced multi-channel operation can be achieved using dual audio adaptors." However, each woofer and tweeter would need its own signal path and amplifier channel. So, although the DSP route is feasible, it would add quite a lot of complexity. (Hill, 2019, pp. 305-307; Eargle, 2003, pp. 226-227; PJRC, n.d.a; PJRC, n.d.c)

Pre-built passive crossover boards exist, and they do make a 2-way speaker easier to wire than a custom crossover. But this would be a compromise because I would have to find drivers that are compatible with these generic crossovers. Also, in a small enclosure such as mine and the fact that it is an awkward shape, I don't have much surface for the drivers to mount onto. This also adds more components which I do not think is a sensible commitment for the time I have. For these reasons, it is safe to say it is possible but not feasible for my project. (Hill, 2019, pp. 305-307)

Coaxial drivers initially seemed like a favourable option, because they combine the woofer and tweeter into one unit and sometimes include simple crossover hardware. However, many of the small 3.5-inch coaxial drivers I found are mainly intended for car audio use or free air use rather than for a portable enclosure. That makes them less predictable for this project. Although they seem appealing, the full range drivers look like my current best option. (Eargle, 2003, p. 166; Hill, 2019, pp. 305-307)

Driver choice:

What I have found is two Dayton Audio ND91-8 full-range drivers with one Dayton Audio DSA135-PR passive radiator. The ND91-8 is a strong choice as it has a Vas of 1.53 litres with a high Xmax of 5.1 mm and a cone surface area of 30.4cm2 which means a Vd of 15.5 cm although official specs from Dayton Audio gives a Vd of 12.2cm3. This is great for a compact enclosure where excursion matters as much as cone size. Its published response of 65 Hz to 17 kHz means it can cover a broad range without needing a separate tweeter, which keeps the project simpler. (Dayton Audio, 2026) The DSA135-PR makes sense practically for the passive radiator. Its published figures include an Sd of 75.4 cm^2, Xmax of 8 mm, and Vd of 60.3 cm^3. Dayton states that passive radiators should generally provide at least double the displacement

of the active drivers. With two ND91-8 drivers giving a combined Vd of 24.4 cm³, the DSA135-PR is well above the threshold. The DSA135-PR also allows me to add screw on weights to the back. This is a powerful addition as it means I can easily tune the radiator based on the final internal volume. (Dayton Audio, n.d.a)

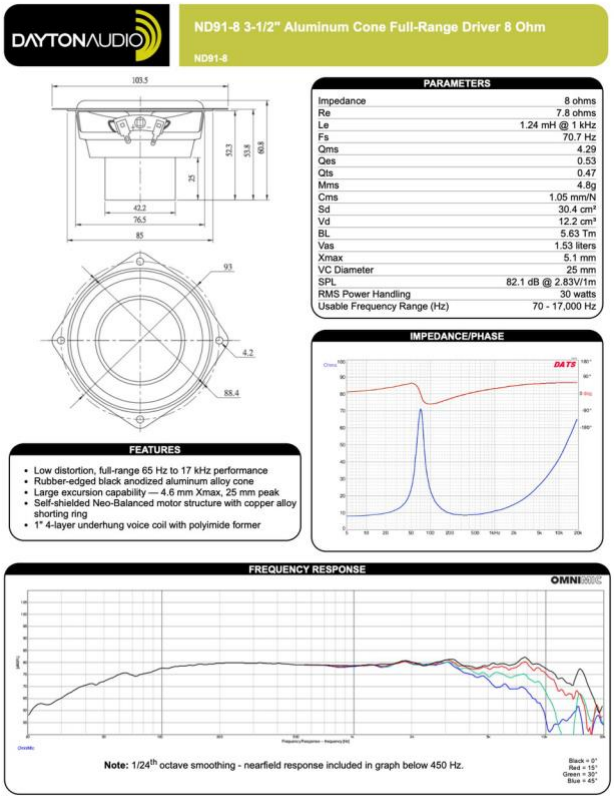
I also looked at 4-inch full-range alternatives, as I thought a bigger cone might help with the bass extension. I was weighing whether I should increase the driver size and sacrifice on portability. However, the results were not great. The Dayton DMA105-8 has much more cone area than the ND91-8, but because its Xmax is only 2.5 mm, its Vd only rises to 13.5 cm³, which is not a substantial increase for the extra space it takes up. The Dayton RS100-8 is similar, it has a Vd of 12.4 cm³ which is barely above the ND91-8 and its Fs of 92.2 Hz is worse for bass extension. Because of that, I do not think these 4-inch full-range drivers are better choices for this enclosure.

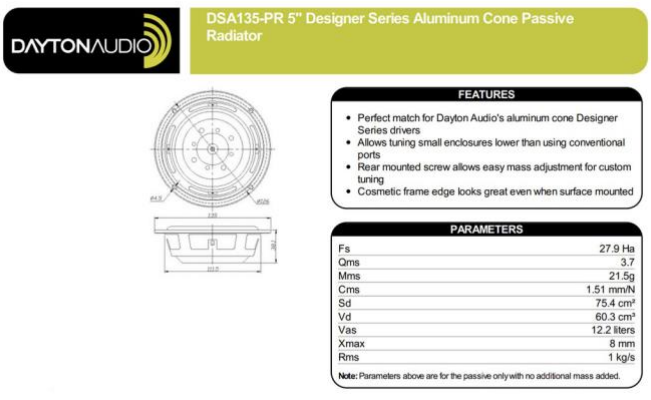
I also found a 4-inch mid bass driver also. The Dayton ND105-8 has 51.5 cm² Sd, 20.6 cm³ Vd, 4.0 mm Xmax, and Fs of 66.6 Hz, which makes it a much stronger than the ND91-8. It would have noticeably stronger low-end output. The downside is that it is not a full range driver, so it would require a tweeter and crossover, which as discussed earlier is not the route I am planning (Dayton Audio, n.d.b)

I also considered small coaxial drivers with included crossover hardware, but most of the realistic 3.5-inch coaxial options I found were mainly intended for car systems rather than for a small passive radiator enclosure. They would reduce crossover work, but they would give more uncertainty and anecdotal experience with using coaxial drivers intended for cars has been mostly negative.

Final decision:

After comparing some options, I think it is fair to say that larger drivers are not always better for bass. For my compact enclosure think the most sensible final choice is two Dayton Audio ND91-8 full-range drivers with one Dayton Audio DSA135-PR passive radiator mounted on the bottom.





Dimensions for modelling and full driver specs:

DSA135-PR:
Dimensions: Overall diameter: 134.87 mm, Cutout diameter: 111.76 mm, Depth: 38.1 mm.
ND91-8 full-range drivers:
Overall outside diameter: 103.5 mm, overall depth: 56.4 mm, baffle cutout diameter: 85 mm, mounting holes: 4

Driver revision:

Unfortunately, the drivers shipping from outside of the UK likely wouldn't arrive in the timeframe I have. Through Rob Phillips, I have access to smaller set of drivers but as there are only two the design shifts from a 2 driver 1 radiator, to a single driver and a radiator. The project has now shifted from trying to build a fully-fledged final design the first try, to building a proof-of-concept prototype that allows me to better understand the modelling and building process and serve as a justification for the intended final design.
The prototype will now use a pair of BMR 12 W 8 ohm NXT Audio 2.5 inch drivers made available through Rob Phillips with one being repurposed into a passive radiator. The public information I could find for this exact unit, AUBA05809-0002, only gives basic mechanical and electrical information rather than a proper set of Thiele small-signal parameters. The figures that are available are: (ICCFL, n.d.)

- frame size of roughly 64-65 mm,
- overall thickness of 25 mm,
- nominal impedance of 8 ohm,
- rated power of 12 W,
- installation opening of roughly 57 mm,
- M3 mounting geometry.

What I could not find in any public source were the parameters that matter for consideration, such as Fs, Qts, Vas, Sd, Mms and Xmax. As the project has transformed into designing a prototype, these specifications matter less but will make expected values and radiator tuning incredibly difficult.

Modelling

I started modelling the main design in onshape before I was fully aware that I would be using different drivers, but I still wanted to model the initial design anyway to show what it would have looked like and this is likely going to be the next iteration once the prototype is complete. In simple terms this was the design process for both the initial design (let's call it Design A) and the prototype design (Design B) After my third meeting with Rob we took a trip to the 3D printing workshop and spoke with a very kind woman called Hailey who told me all about the process of getting a model 3D printed and informed me of the different materials they currently have available. Due to the machines being in demand she advised me to finish the 3D models as soon as possible and to send them over to her through the official form via a STL file.

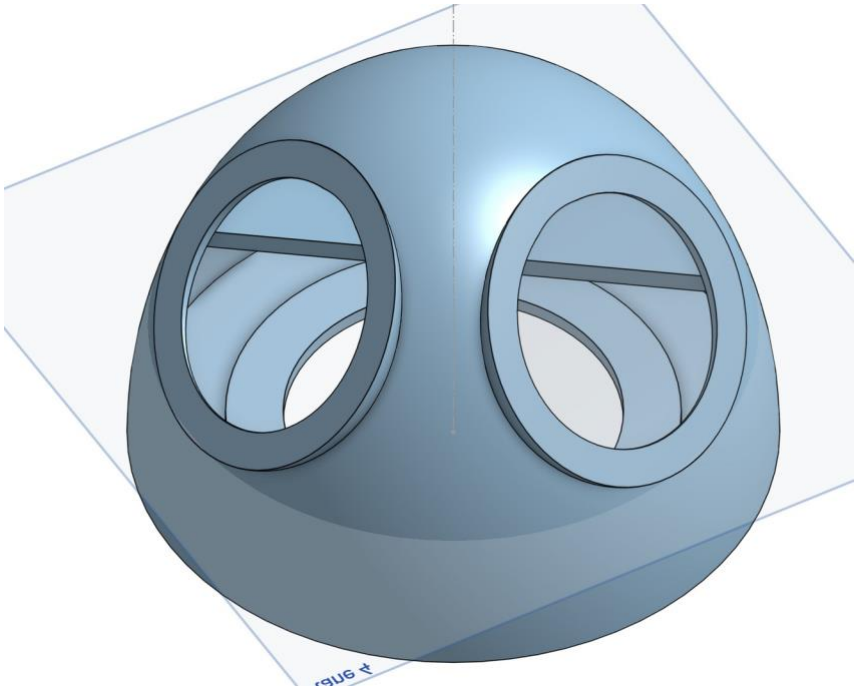
Design A Process:

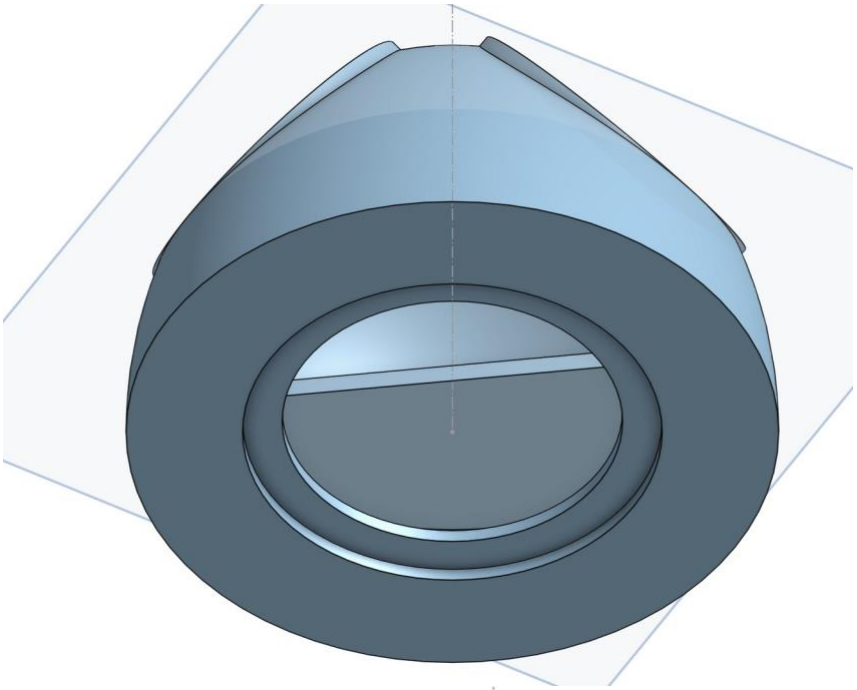
1. Started with a sphere and z scaled it to form an oblate spheroid shape roughly like a rugby ball.

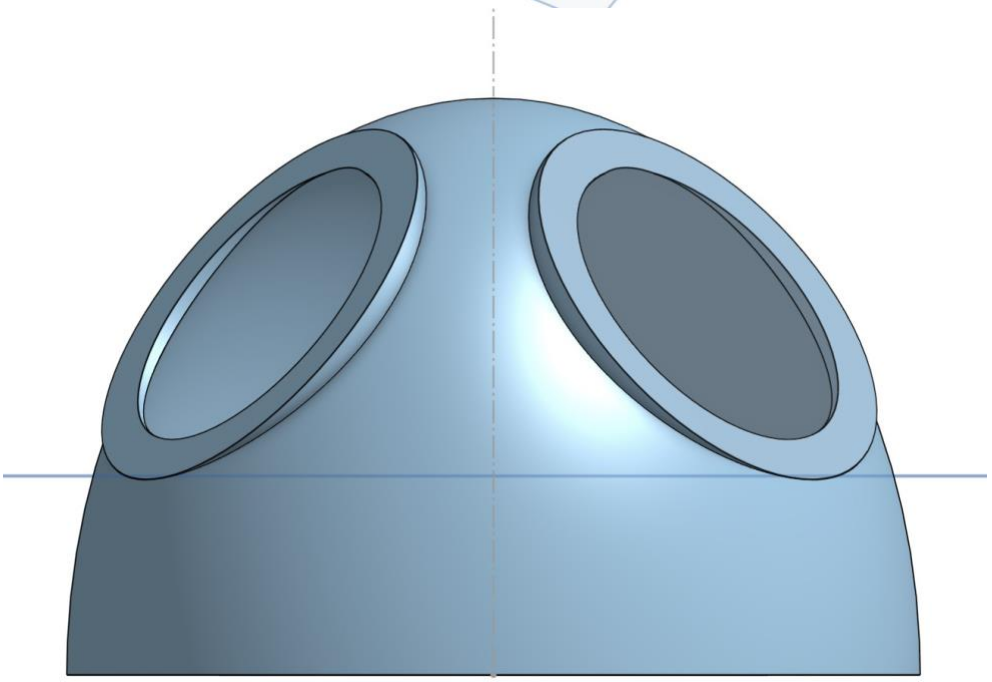
2. Repeat with a smaller sphere and subtract from the inside to hollow the shape out leaving a 5mm wall thickness
3. Cut that shape in half leaving me with a flat base.
4. Placed a tangent plane against the ellipsoid at a height that gave a moderate upward angle (ideally to listeners ears)
5. Used the driver measurements to draw circles and used the extrusion tool to bore and create mounds for driver mounting. The extrusion tool was also used to make the mount sit flush to the tangent plane instead of extruding outwards too much.
6. Copied that same plane a rotated it around the centre line 112.5 degrees and then did the same process for the second driver mount
7. Created a plane at the base of the shape and used the driver measurements to create the bottom mount, however this mount was indented so the PR would have hopefully sat flush against the base. 8. A plane was created in the middle of the shape high enough to clear the passive radiators and then a shelf was added by manually drawing a rectangle and extruding it to the inside face of the shape

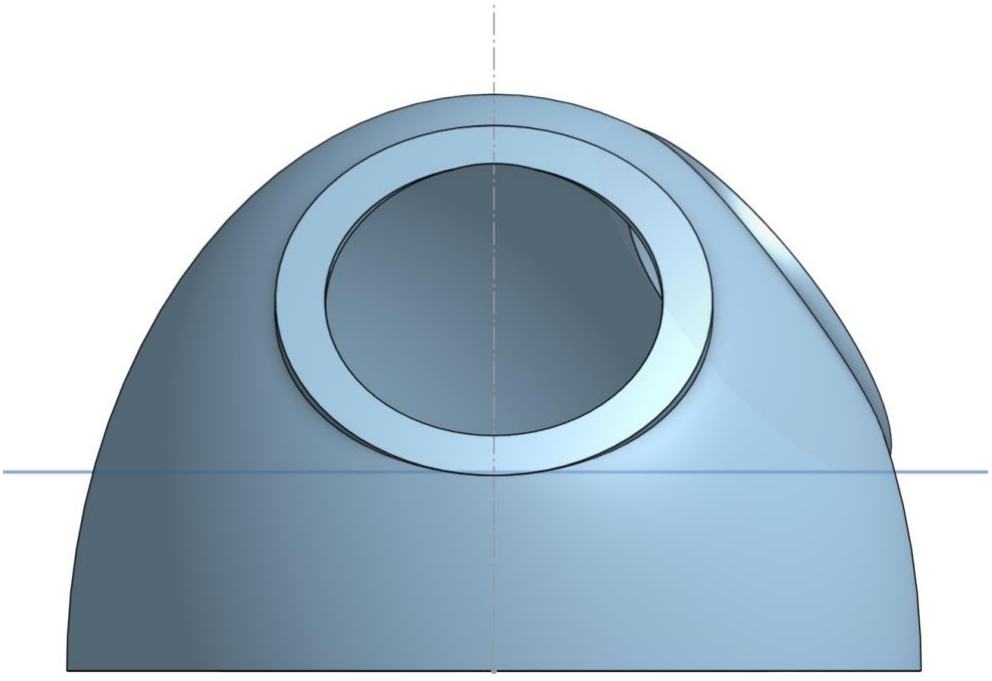
Design A Final Dimensions:

- Egg height: 144.92 mm
- Egg radius: 107.35 mm
- Wall thickness: 5 mm
- Internal volume: 2903 cm^3
- External volume: 3497.8 cm^3
- Final overall body size:189.89 mm × 169.63 mm × 95.80 mm







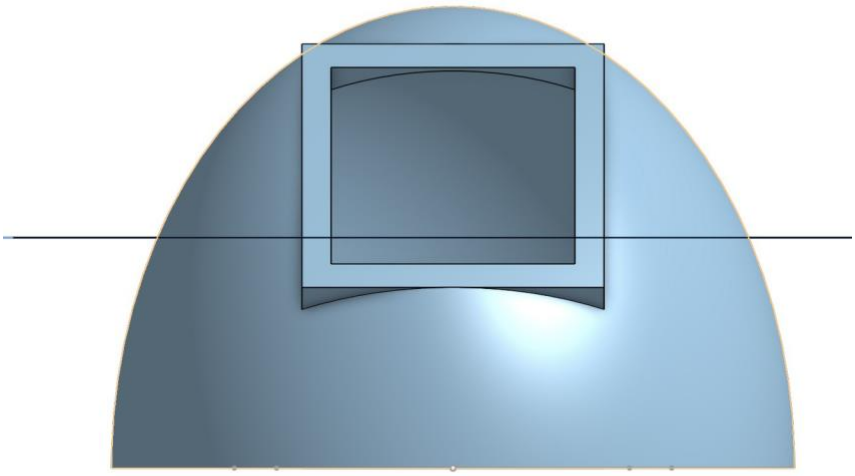


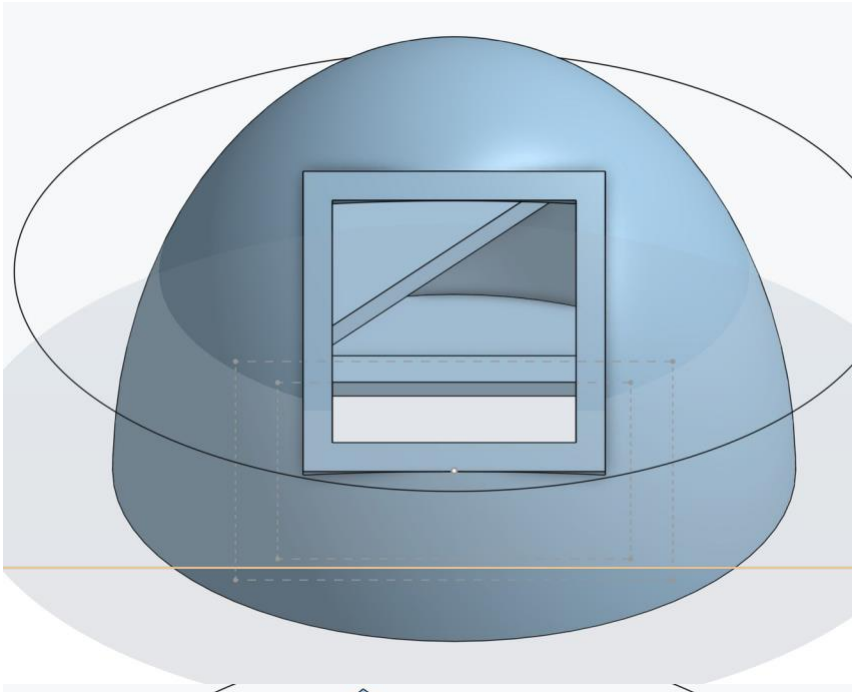
Design B Process:

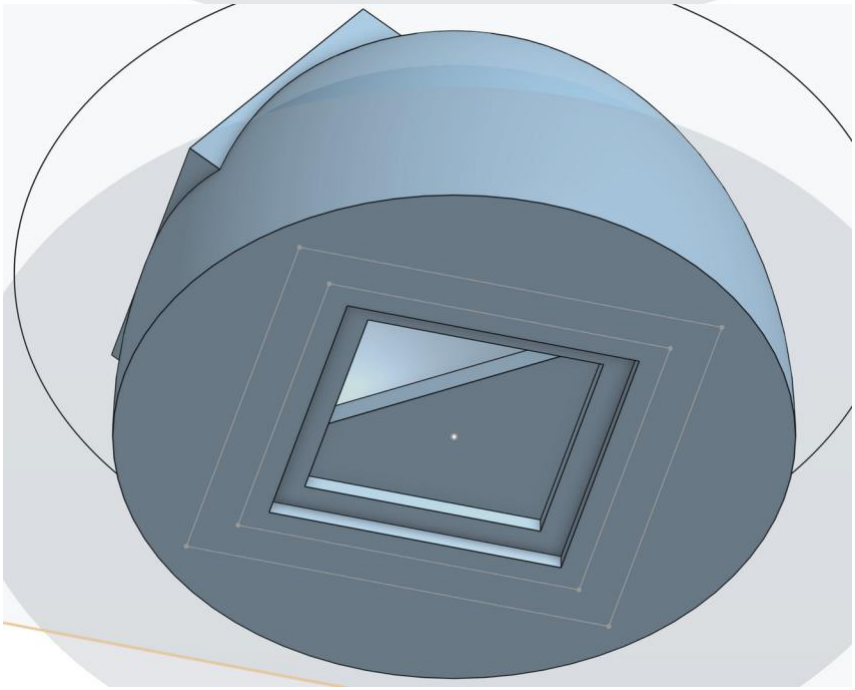
1) Starting with a copied version of design A I deleted one of the tangent driver mounts
2) I worked out the scaling factor between the diameter of the A drivers and the Length of the B drivers so that I could make direct changes without re-doing a lot of work at a smaller size, and then I could just scale it down
3) I turned the circles into squares with the scaled-up length and did the same extruding tricks I used on the circles
4) Leaving the shelf as is, I scaled the whole design down so that now the length of the squares I made fit the driver specs

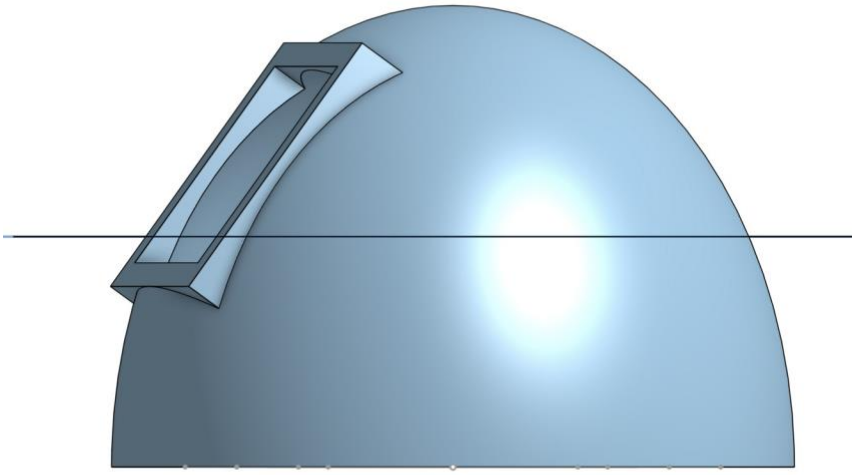
Design B Final Dimensions:

- Uniform scale factor: 0.6909 (69.09%)
- Effective outer radius: 74.17 mm
- Effective outer height: 100.13 mm
- Overall body size: 129.65 mm × 90.73 mm × 100.13 mm
- Wall thickness: 3.45 mm
- Internal volume: 957.4 cm^3
- External volume: 1153.6 cm^3









Enclosure Material and 3D printing:

While I was in the process of 3D modelling my designs, I visited the university 3D printing workshop to discuss the best print route. The processes available to me were FDM 0.4 mm, FDM 0.2 mm, SLA, and SLS. As the part has a practical use rather than being a decorative object, I had to consider more than just its visual appeal. I had to consider structural integrity, acoustic properties, time to print, and cost.

FDM is the most straightforward process for prototype work. Protolabs Network describes FDM as a process where molten thermoplastic is laid down layer by layer. Its main advantages are low cost, quick turnaround, and suitability for functional prototype parts, while its weaknesses are visible layer lines, lower precision than SLA, and more obvious post-processing requirements. For a prototype, the post-processing, visual appeal, and somewhat precision become moot considerations given the upside of a fast, cheap turnaround. FDM 0.4 mm and FDM 0.2 mm are not different technologies, simply different levels of print fineness. The 0.4 mm route is the rougher option. It is better suited to quickly producing a cabinet shell that can be test-fitted, drilled, sanded, and revised.

(Protolabs Network, n.d.; Formlabs, n.d.)

Acoustically, FDM can be suitable for a prototype enclosure because thermoplastics such as PLA or PETG can create a stiff enough shell if the wall thickness is sufficient. A stiffer enclosure is useful because unwanted panel flex can absorb energy from the driver and create some unwanted frequency boosts. FDM prints can have small gaps between layers, which can reduce airtightness. This would reduce air pressure inside the enclosure, making tuning the passive radiator challenging. An FDM enclosure may need thicker walls and possibly sealing on the inside if there are any gaps post-printing. The acoustic difference between 0.2 mm and 0.4 mm FDM is likely to be much smaller than the difference caused by wall thickness, stiffness, internal volume, sealing quality, and driver mounting, as well as other design factors. The 0.2 mm choice would also take longer to print, making it undesirable for fast turnarounds. I see it mainly as an aesthetic option, and since this speaker is a prototype, small visible layer lines are not a genuine problem at all.

(Hill, 2019, pp. 28-33)

Both Protolabs Network and Formlabs describe SLA as the option that gives higher precision and a much smoother surface finish than FDM. The downside is that SLA parts generally need post-curing and more careful handling as resin systems are more brittle than the thermoplastics used in FDM. For a portable speaker enclosure, where the speaker might get knocked about, put in a bag, or handled frequently, the brittleness is a disadvantage and neither good for prototyping nor a finished product. A brittle shell may also lead to cracks while using screws. (Protolabs Network, n.d.; Formlabs, n.d.).

(Protolabs Network, n.d.; Formlabs, n.d.)

SLS is probably the strongest option overall for the final design. Formlabs describes SLS as producing strong nylon parts with excellent mechanical properties. It also doesn't need dedicated support structures because the unsintered powder supports the part during printing. That is great for a tight enclosure with a shelf or cutouts that visually should be smooth. Acoustically, SLS nylon would be a strong candidate because it would produce a robust and impact-resistant shell. Enclosure stiffness and secure mounting reduce the chance of unwanted vibration, panel resonance, and rattling. SLS parts can have a slightly porous surface texture, so an external finish to seal any micro-holes would be the best practice, such as an epoxy resin seal. Overall, SLS would be the best option.

(Formlabs, n.d.)

For cost and turnaround, the best prototype option is FDM 0.4 mm. All I need for the prototype is proof that the driver and passive radiator fit and that the shape is feasible. It doesn't yet need to be the polished final shell. For the prototype, it is important that it is sealed properly and there are no loose internals.

Internal Electronics:

The goal was to have a speaker that had functional button controls, Bluetooth compatibility, wirelessly rechargeable and potentially some interesting creative DSP coding such as an auto eq or pitch up/down. As the project has shifted to a prototype I'm just going to be focusing on getting sound from a phone to play through the speaker with basic processing, but the infrastructure will be there for when I want to advance. The signal flow will be Phone -> Bluetooth chip -> audio Sheild/Teensy -> amplifier -> Drivers. I will also need to find a suitable rechargeable battery to power the hardware. I have the shelf to mount electronics but if I find I am tight for space, as the components would be quite light, I can use double sided tape and/or mounting putty to secure them to the inner walls.

Digital Sound Processing:

Since my first meeting with Rob, I have known I was going to use a Teensy with an audio shield. The two I bought were the Teensy 4.0 and the audio adapter board for Teensy 4.0 Rev D. These two boards are a good starting point, but they are not a complete portable speaker on their own. The audio shield provides line in and out, headphone output and audio processing but does not directly power a driver fully. The Teensy 4.0 is a strong choice because it provides more than enough processing power that I would need for my initial design. Also, PJRC notes that the board uses cortex-m7 dsp instructions and that code runs immediately on powerup which means that I won't need a complicated startup routine to start passing audio.

The audio shield is the part that makes the Teensy practically useful for audio rather than just as a general microcontroller. PJRC states that the Rev D audio board for Teensy 4 provides 16-bit, 44.1 kHz audio, with stereo line level input, stereo line level output, and stereo headphone output. PJRC also states that the Teensy Audio Library allows the input and output to be used at the same time alongside audio processing objects, which is exactly what I need for a Bluetooth speaker where the incoming signal must be received, optionally processed and then sent back out in real time. For the prototype, the most crucial factor is that the shield gives me a line input and line output path, but it still does not directly drive a loudspeaker. It is an audio interface and processing stage, so I will need an amplifier to boost the signal to be able to move the drivers. (PJRC, n.d.c)

For the physical connection between the two boards, PJRC states that the Rev D board is specifically for Teensy 4.0 and 4.1, and that the two boards can be joined using two 14x1 pin strips. PJRC also publishes the required signal map between the boards. The key point from that map is that the audio chip on the shield communicates with the Teensy over 7 important signals. SDA and SCL for control over I2C, then MCLK, BCLK, LRCLK, DIN and DOUT for I2S digital audio. I'm not too sure exactly what these terms specifically mean, but I do know vaguely that they are different paths for different purposes such as a clock to sync up signal flow, a left and right signal controller, raw audio in and out and a main data connection used for control commands. (PJRC, n.d.c)

A critical point to note is that PJRC states the Teensy's digital pins are not 5V tolerant and should not be driven above 3.3 V. This is another reason to use an audio shield. In simple terms the audio shield handles the big signals while the Teensy sits back and controls and modifies those signals without dealing with messy fluctuating voltages. (PJRC, n.d.d)

The first stage of this design does not need to be ambitious as the project has shifted into a prototype proof of concept. The sensible starting point is simply to configure the audio shield, select line input, and pass that signal through to the line output with minimal intervention. Based on the driver's frequency response however it would be worth using a high pass filter so that unnecessary data is not being sent to drivers that cannot physically keep up. The main point of choosing the Teensy platform now is to establishes the processing infrastructure needed for future revisions.

Bluetooth Receiver:

For Bluetooth input I am using a cheap MH-M18 family Bluetooth audio receiver module. These boards are good for this project because they are cheap, very small, and designed specifically for receiving Bluetooth audio and outputting it as analogue stereo. The published module documentation states that this family supports Bluetooth 4.2, automatic reconnection, and stereo two-channel output, and that once powered the phone searches for the default Bluetooth name associated with the specific board version, such as MH-M18. For the first prototype that is enough. For a final product, I would investigate Bluetooth boards where the discoverable name is changeable. However, at this stage I am prioritising reliable connection.

This board outputs an analogue left and right signal which can be fed straight into the audio shield's line input. On the MH-M18, the relevant pins are VCC, GND, L and R. On the MH-M28, the relevant pins are 5V, GND, L, R and AGND, and the datasheet also notes an alternative VBAT input if it is being run directly from a single-cell lithium supply. In the simplified prototype I am powering the Bluetooth board from a 5V bus setup, and then taking L, R and AGND/GND into the audio shield line input. The Bluetooth module handles wireless reception and decoding, and the Teensy/audio shield pair handle everything after that.

Amplifier Choice:

For the power amplifier, I am using the Adafruit PAM8302 Mono 2.5W Class-D amplifier. This is a good choice for the prototype because it is physically small, efficient, and intended for battery powered projects. It is also an amplifier that is commonly bought alongside the Teensy and audio shield leading me to believe that this is a standard tried and tested setup. Adafruit states that it can run from 2.0 V to 5.5 V DC, is suitable for 4–8 Ω speakers, and can deliver up to 2.5 W into 4 Ω, or around 1.5 W into 8 Ω when supplied from 5.5 V. Since my prototype driver is an 8 Ω unit, this is a slightly underpowered solution but that is acceptable for this stage because the main priority is proving the signal path works rather than maximising SPL.

The PAM8302 is also useful because its input arrangement makes it easy to integrate with the audio shield. Adafruit describes the amplifier input as differential, using A+ and A-, but also notes that when the source is not differential the A- input can simply be tied to ground. That works for me as the audio shield has a simple line out so I can wire the a- input to ground. This does make the audio signal unbalance but for my prototype this is completely acceptable as there is unlikely to be any noise interference from a 5-volt system. The main source of noise if any would be from the power supply so I need to ensure to get a clean connection. (Adafruit, n.d.b)

Power Arrangement:

Initially, I intended to use a more integrated internal battery system based around a single-cell 3.7 V LiPo, a JST-PH lead, and a dedicated charging/boost board so that the speaker could be fully self-contained and rechargeable through a built-in USB-C port. I acquired the battery but after research showed I ideally needed a boost/charger board, it became a struggle to source one that would be delivered in time. Because of this I figured I could utilise a phone power bank. I found the smallest one I could find and ill use a USB-C male to 2 pin bare wire connector.

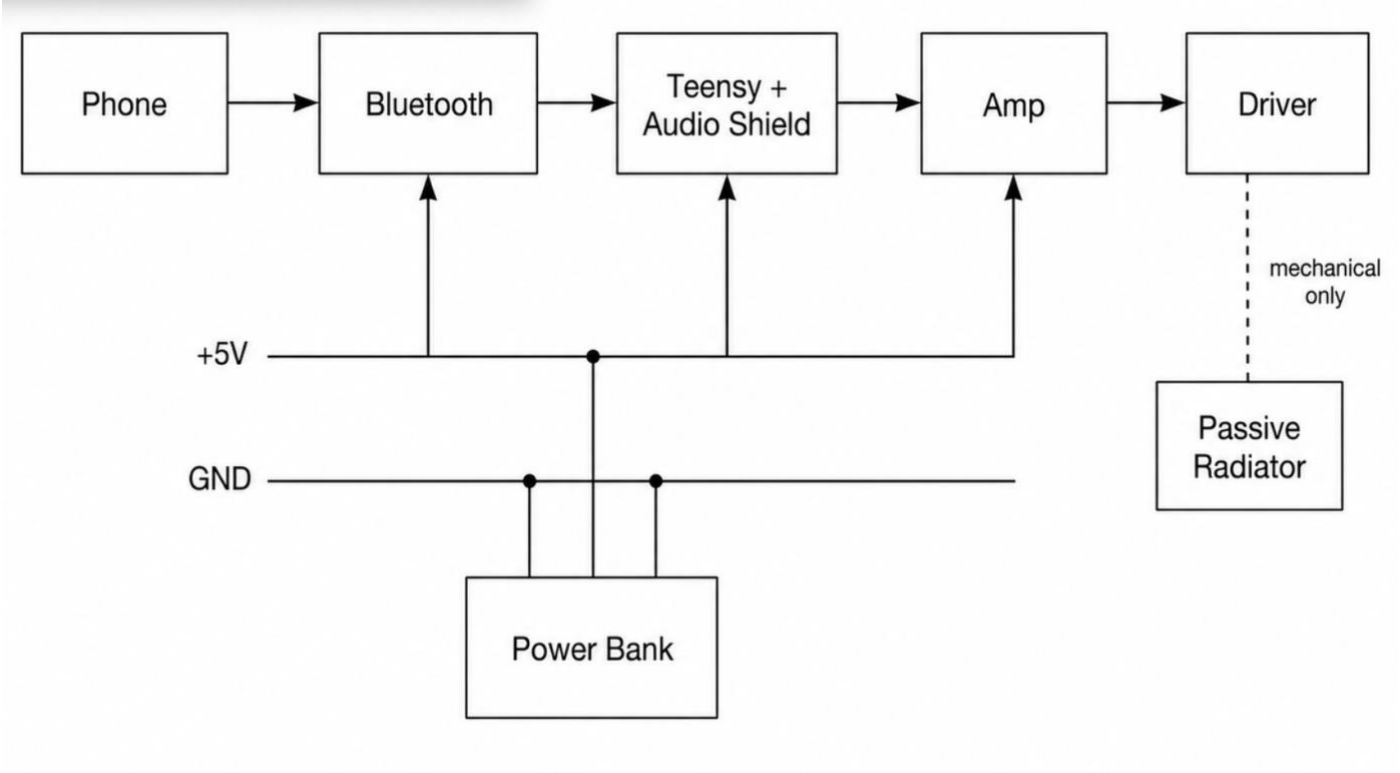
This is a clunky but sufficient compromise. PJRC states that Teensy can operate from power applied to its Vcc / +5V pin and Ground and gives a recommended operating range of 4.5 V to 5.5 V for full-speed operation. A regulated 5 V USB-C power source sits in the correct range for the Teensy. PJRC also warns that if USB power is used together with external power, it can damage the Teensy. This simply means that I should avoid powering the Teensy via my laptop while coding while it is also connected to the power bank. (PJRC, n.d.d)

The main benefit of the power-bank approach is that it is a solution that doesn't stall my project for a week.

It is rechargeable, already regulated to 5 V, and removes the need to solve battery charging and voltage boosting inside the enclosure before the first working prototype exists. It is not as elegant as the intended design but for a proof-of-concept build it is sufficient. Once the audio chain is tested and working, the power arrangement can be revised into a more integrated battery design.

Wiring plan:

5 V USB power bank connected to a USB-C bare-wire lead.
The positive 5 V and ground from this lead are split so that the same supply powers the Teensy 4.0, the Bluetooth receiver, and the PAM8302 amplifier in parallel. NOT SERIES
The Teensy is powered through 5V and GND, the Bluetooth receiver through its 5 V/VCC and GND pins, and the amplifier through VIN and GND.
The amplifier power and speaker output connections are current hungry, so these are the connections where I can double up wire if necessary as I only have 26 AWG wire. Double the wire will split the current. The Bluetooth receiver is the wireless audio source.
Its left output, right output, and audio ground are wired to the line input of the Audio Adapter Board Rev D.
The Audio Adapter Board is stacked directly onto the Teensy
The processed output is then taken from the left line output and line ground of the audio shield and sent into the A+ and A- inputs of the PAM8302 mono amplifier.
Only the left channel is used because this prototype is mono.
The amplifier drives the loudspeaker directly through its SPK+ and SPK- outputs.
These two speaker wires go only to the driver terminals and are not connected to ground, because the PAM8302 uses a bridge-tied output stage meaning it essentially creates its own ground reference so connecting them to another ground would likely short the circuit.



DSP Programming:

Programming the Teensy is relatively straightforward for the prototype as the code only needs to do 4 jobs at this stage: Receive audio from the Bluetooth board, sum it to mono, apply a high pass filter and then send it to an output channel. The reason I'm planning on applying a high pass filter is because the driver that I'm using, although I don't have reputable sources of Thiele Small parameters, it's reasonable to assume given the physical limits of its size, that the bass output is unlikely to be under 100Hz. Sending this low-frequency data to the driver would cause over excursion in most cases as it would not be able to keep up. As I do not have buttons or a crossover system for this prototype I do not need to worry about on/off, pairing, audio manipulation logic as well as the extra code needed to facilitate the crossover.

Code Snippets:

The Arduino bare-minimum structure uses setup() and loop(). setup() runs once when the board starts or resets. loop() normally repeats forever after setup() has finished. PJRC have lots of code examples that are relevant to me. The best starting point is the stereo pass through section which uses the Audio Shield line input, I2S input and output, patch cords, AudioMemory(), and the SGTL5000 control object (audio codec).
(Arduino, 2024a; Arduino, 2024b; Arduino, 2024c)

```
AudioInputI2Si2s1;
AudioOutputI2Si2s2;
AudioConnectionpatchCord1(i2s1, 0, i2s2, 0);
AudioConnectionpatchCord2(i2s1, 1, i2s2, 1);
AudioControlSGTL5000sgtl5000_1;
constintmyInput = AUDIO_INPUT_LINEIN; voidsetup() {
AudioMemory(12);
sgtl5000_1.enable();
sgtl5000_1.inputSelect(myInput);
sgtl5000_1.volume(0.5);
}
```

this receives stereo audio from the Audio Shield line input and sends it straight back out as stereo. Naturally, the left input goes to the left output, and the right input goes to the right output. It also shows the setup calls needed for the Audio Shield. For my mono speaker however, this will be tweaked to sum the signals and only send them to one output channel. The reason the signal needs to be mono is that individual drivers do not have left and right inputs, they have positive and negative terminals that represent one channel. As mentioned before, the driver is bridge tied so grounding one terminal would result in a short. The A- input on the amp should be grounded as the Teensy only sends out 1 signal + ground, the amp will do the polarity switching for me, and then the A+ and A- outputs should go to their corresponding terminal (although for a single mono driver the terminal polarity is essentially irrelevant)
PJRC shows the general way patch cords connect audio objects, and the way AudioMemory() allocates audio blocks. (PJRC, n.d.b)

```
AudioConnectionmyConnection(source, sourcePort, destination, destinationPort);
AudioMemory(numberBlocks);
```

AudioConnection routes audio object outputs to audio object inputs. AudioMemory() reserves blocks of memory so that the audio library can move sound through the objects reliably. For the data flow i want to achieve, my code would look like this: (PJRC, n.d.b)

```
AudioConnectionpatchCord1(audioInput, 0, monoMix, 0);
AudioConnectionpatchCord2(audioInput, 1, monoMix, 1);
AudioConnectionpatchCord3(monoMix, 0, driverHighPass, 0);
AudioConnectionpatchCord4(driverHighPass, 0, audioOutput, 0);
```

PJRC's guidance says that if more than one audio signal needs to feed into the same place, the signals should not be joined directly. Instead, they should be combined using a mixer object. If several outputs are connected to the same input, only the last one may work properly. (PJRC, n.d.b)

```
// Guidance pattern from PJRC documentation:
// To feed multiple outputs to a single input, use the Mixer object.
classAudioMixer4 : publicAudioStream {
public:
AudioMixer4(void) : AudioStream(4, inputQueueArray) { } virtual void update(void);
void gain(unsigned int channel, float gain) { ... }
};
```

This would not be a copy-paste into my project as this is showing the code that defines AudioMixer4, but the important part is understanding how the mixer object works and how to implement it as follows.

```
AudioMixer4monoMix;
AudioConnectionpatchCord1(audioInput, 0, monoMix, 0);
AudioConnectionpatchCord2(audioInput, 1, monoMix, 1); monoMix.gain(0, 0.5);
monoMix.gain(1, 0.5);
```

Here I've set up the AudioMixer4 and it's called monoMix, the left and right inputs are fed into the mixers channels 1 and 2 and then using the gain function I'm halving the gain from left and right as they are summing which will essentially add back up to 1 again.
The Teensy Audio Library biquad filter source provides setHighpass() (Stoffregen, n.d.b)

```
void setHighpass(uint32_tstage, float frequency, float q = 0.7071) { ...
setCoefficients(stage, coef);
}
```

In Hill's modelling, a Butterworth response corresponds to a total system quality factor (Q) of 0.707. This value is widely considered the ideal "reputable" target for a flat response in closed-box speaker designs before the roll-off begins. I'll start the cutoff frequency at 130hz as this can always be tweaked. (Hill, 2019, pp. 45-50)

Building the final sketch:

```
#include <Audio.h> // Loads the Teensy Audio Library, which provides the audio objects such as input, mixer,

#include <Wire.h> // Loads the I2C communication library, needed so the Teensy can control the SGTL5000 audio shield chip.
```

```
// Selects the physical input on the Teensy Audio Shield.

// AUDIO_INPUT_LINEIN means the project is using the line input pins, not the microphone input.

const int Bluetooth= AUDIO_INPUT_LINEIN;

// Creates an I2S audio input object.

// This receives digital audio from the Teensy Audio Shield. // Channel 0 = left input, channel 1 = right input.

AudioInputI2SaudioInput;

// Creates a 4-channel mixer object.

// In this project, only two mixer inputs are used: left and right. // The mixer combines them into one mono signal.

AudioMixer4monoMix;

// Creates a biquad filter object.

// This will be used as a high-pass filter to reduce low frequency content going to the driver.

AudioFilterBiquaddriverHighPass;

// Creates an I2S audio output object.

// This sends processed digital audio back to the Teensy Audio Shield output.

AudioOutputI2SaudioOutput;

// Creates the control object for the SGTL5000 chip on the Teensy Audio Shield. // This is needed to enable the shield, select the input and set output level.

AudioControlSGTL5000audioShield;

// Routes the left input channel from the audio shield into mixer input 0.
AudioConnectionpatchCord1(audioInput, 0, monoMix, 0);

// Routes the right input channel from the audio shield into mixer input 1.
AudioConnectionpatchCord2(audioInput, 1, monoMix, 1);

// Routes the mono mixed signal into the high-pass filter input. AudioConnectionpatchCord3(monoMix, 0, driverHighPass, 0);

// Routes the filtered mono signal to output channel 0 of the audio shield. // Output channel 0 is usually the left output.

AudioConnectionpatchCord4(driverHighPass, 0, audioOutput, 0);

// Sets the gain for each mixer input.

// Left is 0.5 and right is 0.5, so the stereo signal is averaged into mono.
```

```
// This helps avoid clipping.

const float MIX_GAIN = 0.5f;

// Sets the high-pass filter cutoff frequency to 130 Hz.

// Frequencies below this point are reduced to help protect the driver from excessive bass movement.

const float HPF_FREQ_HZ = 130.0f;

// Sets the Q value of the high-pass filter.

// 0.7071 is the standard Butterworth Q for a second-order filter.

// Butterworth response is commonly used because it gives a smooth filter shape without a resonant peak at the cutoff frequency.

const float HPF_Q = 0.7071f;

void setup() {

// Reserves audio memory blocks for the Teensy Audio Library.

// 16 blocks is an arbitrary choice based on the number of processes I'm putting the audio through
AudioMemory(16);

// Turns on the SGTL5000 (codec) audio shield.

// Without this, the audio shield wouldnt pass audio correctly.

audioShield.enable();

// Selects the line input as the audio source.

// This matches the earlier Bluetooth variable which is set to AUDIO_INPUT_LINEIN.

audioShield.inputSelect(Bluetooth);

// Sets the audio shield output volume to 50%.

audioShield.volume(0.5);

// Sets mixer input 0, the left channel, to half gain.

monoMix.gain(0, MIX_GAIN);

// Sets mixer input 1, the right channel, to half gain.

monoMix.gain(1, MIX_GAIN);

// Configures the biquad filter as a high-pass filter.

// Filter stage 0 is used.
```

```
// The cutoff frequency is 130 Hz.

// The Q value is 0.7071, giving a Butterworth-style response.

driverHighPass.setHighpass(0, HPF_FREQ_HZ, HPF_Q); }

void loop() {

// The Teensy Audio Library processes the audio continuously in the background.

}
```

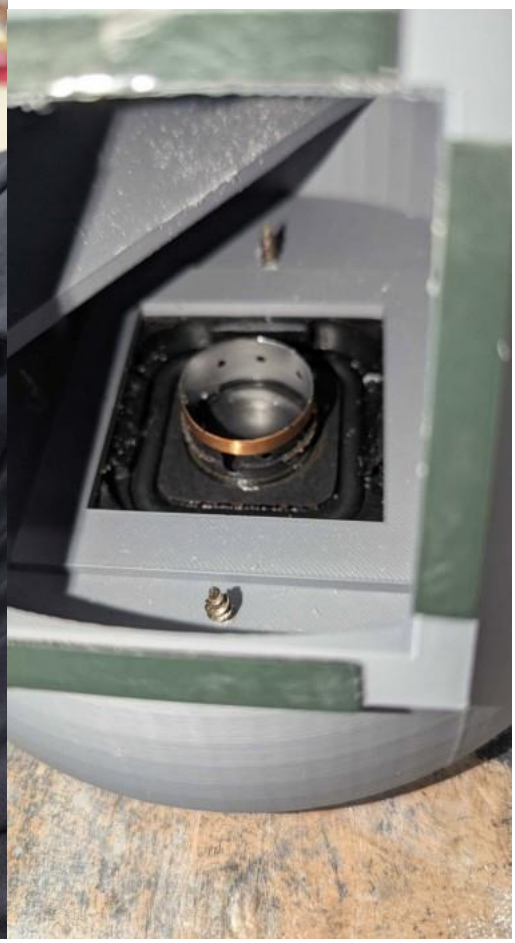
I was able to upload this sketch onto the Teensy with no compile errors.

The build:

Picking up the 3D Printed Enclosure:

After sending the STL file to the university 3D printing workshop, I collected the printed prototype enclosure a few days later. It ended up costing me 4 pounds. This was a very successful stage of the project because the print came out almost exact to the CAD design. There were still normal FDM limitations visible, such as layer lines and slightly rougher edges around some of the cut-outs, but for a prototype these were not problems. The main functional features were there and, in my opinion, looks great. It also confirmed that the 0.4 mm FDM route was the best choice for this stage. A higher detail print would have perhaps looked better but cost more and taken more time which at my current stage is of the essence.





Prototype Assembly and Physical Fit:

Once I had the printed enclosure, the next stage was to test the physical fit of the driver, passive radiator, electronics, and power plan. Because the project had shifted from the originally intended two-driver Dayton Audio design into a proof-of-concept prototype, the physical assembly became more experimental. The aim was no longer to produce a fully optimised final consumer ready loudspeaker, but to show that the design route could be built, show understanding of the process, and reflect on future iterations given the prototypes outcome.

To make the passive radiator, I used one of the available BMR-style drivers and removed the magnet and voice coil assembly. This left the diaphragm, surround, and a copper(?) tube acting as an unpowered membrane. It is important to note that this is not the same as using a properly specified passive radiator with a known moving mass, excursion and any other relevant parameters. However, it did let me create a working equivalent for a prototype, which was useful because the project was now mainly proving the enclosure layout and bottom firing radiator concept rather than a fully tuned and tested acoustically sound speaker.

The driver and improvised passive radiator were planned to be mounted using double-sided foam tape. This gave a good physical fit and acted like a simple gasket. However, the seal would still need more work if this was a final build. A passive radiator design relies on pressure changes inside the enclosure, so small leaks around the driver frame, radiator frame, or cable entry points could reduce the radiator movement and worsen the radiators response. In a later revision I would use a cleaner gasket material like a thin bead of removable sealant around the mount.

I was also not able to print the intended feet or false bottom in time. Instead, because this is a prototype, I used four screws as adjustable feet. This is not an ideal final solution because the screws are visually rough and do not give a wide base. As a testing method it is useful though because the screw depth can be adjusted to change the gap between the passive radiator and the surface below it. In the original plan this gap was meant to be tested at different heights, so the screws support this idea.

Electronics Assembly

The electronic assembly was the part of the project where I felt the most out of my depth. The signal path was planned as phone to Bluetooth receiver, Bluetooth receiver to Teensy Audio Shield line input, Teensy DSP processing, Audio Shield line output to PAM8302 amplifier, and then amplifier output to the driver. The DSP sketch uploaded successfully earlier in the project, so this wasn't a point of failure. The hard part was the physical power and signal wiring. The prototype wiring used a shared 5 V and ground arrangement. The intention was that the power input would split in parallel to the Teensy, Bluetooth receiver, and amplifier. The Bluetooth left and right audio outputs would feed the Audio Shield line input, and the Audio Shield output would feed the mono amplifier. Im confident that my planned architecture was solid, but my mistake was not testing each power stage before soldering and insulating the full circuit. I should have tested the USB-C input first, then the Teensy alone, then the Bluetooth board, then the amplifier, and only then joined the full system together. I also underestimated how much of electronics is not just knowing which wire goes where but how much understanding why what goes where helps streamline the process. For example, understanding that ground is a shared reference point rather than a separate channel like audio left and right was something I had to learn during the build. The same is true of the PAM8302 amplifier output. The speaker terminals should not be treated like one signal wire and one normal ground wire because the board uses a bridge-tied output stage. These details are basic to someone experienced in electronics, but they were new to me and were an unexpected major time sync. I would be lying if i claimed to fully understand it even now, but electricity is famously confusing, though I may just be speaking from an inexperienced perspective.

Power Supply Revision and USB-C Issue:

The original intention was to use an internal rechargeable battery. I had considered a small 3.7 V lithium battery, but this would have needed a suitable charging and boost circuit so that the speaker could receive charge safely and still provide a stable supply voltage to the electronics. At the point I had learned this through some research, my time was running out and i could not find a component that would have been guaranteed to arrive on time. I then considered using the smallest power bank I could find so i bought one, but physically it was still too large to sit comfortably and be fully supported inside the enclosure. Because of this I changed the power design into a plug-in prototype. The speaker would be powered externally through a USB-C socket rather than being fully battery powered in this version. For this I bought a set of Sourcing Map Type-C female chassis 2-pin wires rated at 3 A and 5 V. The listing described them as charging-only extension cables and stated that they do not support data transmission or voltage conversion. That seemed fine because I only needed a 5 V power input for my chosen components rather than USB data. I cut off the PH2 terminal to use the red and black wires directly. This was a necessity as i need to split the cable to power 3 separate components as i was wiring in parallel, not series.

Midway through soldering, I decided to test whether the Teensy would turn on from the USB-C to wire input. Admittedly, I should have done this as soon as the power setup was first prepared. Unfortunately, the Teensy did not power on. As I did not have a multimeter available on hand, I was essentially working blind and could not prove whether the problem was the USB-C input or my soldering.

After looking online, I found examples of people having a similar issue with USB-C to wire adapters and breakout boards. The issue is that a proper USB-C source may not output 5 V unless the device side identifies itself correctly on the CC pins. I do not fully understand this, but I can acknowledge that it may have been the issue. Adafruit explains that its downstream USBC breakout uses two 5.1 k resistors on the CC pins to indicate to the upstream port that it should provide 5 V, and Digi-Key describes the same basic approach for simple 5 V USB-C sink devices (Adafruit, n.d.a; Digi-Key TechForum, 2020). This means a 5V looking power input does not guarantee that a Mac USB-C port or USB-C power bank will actually provide 5 V through to the intended destination. Not foreseeing this issue while researching was an oversight on my part.
The fault could also still have been my soldering or through cutting the PH2 connector. The truth is that I cannot prove the exact fault without a meter. What I can say is that the fault appeared while validating powering, so this issue is almost certainly here rather than an issue with the Teensy or another component.

How I would have tested the fault:
The way to diagnose this would be to use a multimeter before powering the full circuit. First, I would set the meter to DC voltage and measure between the red and black wires from the USB-C socket while it is connected to the intended power source. A correct result would be close to +5 V. If the reading is 0 V, the USB-C source is not providing power through that lead. If the reading is around -5 V, the polarity is reversed. If it reads roughly +5 V, the power input itself is probably working, and the fault is further along the wiring.
A potential solution I found in an online thread created by Reddit user u/thatdecade was to use a USB-A to USB-C cable as USB-A power sources are often less dependent on USB-C CC negotiation, whereas USBC to USBC power sources normally expect the connected device to identify itself correctly through the CC pins which a cheap type C female chassis may not be doing. If the prototype powered up from USB-A to USB-C but not from USBC to USB-C, this would strongly suggest that the chassis socket lacked the required USB-C sink resistor arrangement and would give some hope for my soldering.

Power Solutions Moving Forward:
The fix would be to replace the simple USB-C chassis lead with a proper USB-C 5 V sink or trigger board that includes the correct CC resistor arrangement for 5 V output. A second option would be to use a USB-A 5 V power lead for the prototype only if it worked in testing. This would not match the original USB-C design goal, but it would reduce the CC resistor problem and allow the audio to be tested quickly. The third and best option would be to return to the original battery idea, but with a properly selected LiPo charge and boost board, this solves the issue of cables or connectors having missing requirements and it gives me the wireless portability I wanted in the first place.

What I Was Not Able to Complete:
The main thing I could not complete was a full working powered audio test inside or outside the enclosure. The code uploaded successfully and the physical enclosure printed well, but the power issue halted me

before I could properly test Bluetooth audio through the Teensy, DSP, amplifier and driver. Because of this I was also not able to produce frequency response measurements, SPL measurements or passive radiator gap comparisons. This was disappointing because the project had reached the integration stage and was so close to completion, but it also makes the failure point very clear. The unresolved section is the final power and electronics integration.

Shortcomings:

- The final speaker did not reach a working audio playback state before submission.
- I could not verify the USB-C input voltage because I did not have a multimeter available at the time.
- The passive radiator was improvised from a modified driver, so it cannot be treated as a properly tuned passive radiator.
- The driver and passive radiator seal would need more work before meaningful acoustic measurements.
- The printed false bottom/feet were not completed, so adjustable screws were used instead.
- The BMR driver lacked public Thiele-Small data, which made modelling the enclosure and passive radiator behavior much less accurate.
- Revisions had to be made along the way that compromised my initial project scope such as a redesign of the 3D model

Future Development Plan

I do plan to continue the project after submission; the first step will be to rebuild the power path rather than redesigning the whole speaker. The enclosure and signal flow idea are still usable, but the power input must become a tested module rather than an assumption. I would switch back to using an internal battery as the delivery time for a charging/booster board would no longer be an issue and this would simplify my powering issue.

Once the speaker produces sound, the next stage would be acoustic testing. RTINGS.com use more controlled speaker tests, including frequency response, low-frequency extension, maximum loudness, dynamic range compression, and directivity. I would not be able to fully replicate their test setup, but their method gives a useful structure. A simplified version would involve measuring the speaker at around one metre, playing pink noise or a frequency sweep, and checking how evenly it reproduces sound from low to high frequencies. This would not have to be perfect as it would be a rough way for me to guage if any drastic changes need to be made. (RTINGS, 2025a; RTINGS, 2023a; RTINGS, 2023b)

I would also compare the speaker at a normal listening level and at its loudest clean level. This would show whether the bass disappears, whether the driver distorts, or whether the enclosure starts to buzz when pushed harder. Then, I would test the speaker from a few angles, such as directly in front, 30°, 60°, and 90°, to see how much the sound changes off axis. This would be relevant because the enclosure is curved and the driver is angled and full range drivers tend to be quite directional. Lastly, I would then test the radiator to surface gap using the adjustable screw feet on a consistent surface. Even if the measurement setup is not perfect, repeating the same measurement at different heights would still show whether the bottom gap is changing the low-frequency response.

For a more robust future version that could use my initial design, I would also consider whether a more integrated amplifier board would have been a better route. For example, Dayton Audio KAB/KABD style boards combine Bluetooth, amplification, and in some cases DSP into one board with harnesses and battery/DC power options. This would still allow me to choose and design around my own drivers but would reduce the number of separate modules and soldered interconnects or in other words, reduce the number of human error failure opportunities. (Dayton Audio, n.d.c; Dayton Audio, n.d.d; SoundImports, n.d.)

Final Reflection and Conclusion:

Although the prototype did not reach the working stage I wanted it to, I do not think the project was a waste. In fact, I found this experience to be extremely beneficial in expanding my knowledge in so many different areas. A portable Bluetooth speaker sounds like a small product, but it combines acoustic design, driver selection, enclosure modelling, 3D printing, DSP programming, soldering, power management, mechanical building, sealing, and calibrated measuring. I underestimated that scope at the beginning.

The biggest weakness in my process was time management. I spent a lot of time learning specific areas as they became problems which bogged me down mentally and left me little time to iterate and test after failures. I should have tested the USB-C power input before soldering the whole system. I should have had a multimeter from the first day of wiring. I also should have made a simpler first electronics goal, such as getting the Teensy and amplifier playing sound on the desk before trying to solder everything.

However, I have learned a large amount. I now understand much more about speaker acoustics, why bass extension is so difficult in small enclosures, why different speaker build types are used, why sealing matters, why a driver is not just chosen by diameter, how to use and source code for a Teensy and all the components needed for a basic speaker setup. I also learned practical CAD (onshape) and the 3D printing workflow, including how to design around mounting, internal access, and sizing limitations. On the electronics side, I learned soldering basics, what shared ground actually means, why an amplifier output is not the same as a normal ground referenced signal, and why USB-C power is not just two wires. Some of these are real life transferrable practical areas of knowledge that I am grateful to have picked up.

I think it is only fair to mention that the support I had during the project was significant. Hailey in 3D printing helped me understand the 3D print process, Malcolm in the workshop helped with soldering advice, and Rob was great as an FMP tutor because he gave direct technical feedback when an idea was too ambitious or when something needed grounding in practical reality. That helped me keep turning the project into something buildable rather than just falling back to a theoretical document.

In hindsight, I could have chosen a more pre-built electronics platform and focused more heavily on the enclosure, acoustics and I might have been able to get the speaker working. A modular Bluetooth amplifier board with DSP integrated would have reduced the wiring burden and let me spend more time on measurements and passive radiator experiments. On the other hand, attempting the more open Teensy-based route taught me more, because I had to understand the signal path and not just plug in a finished board so there is an upside to the workflow i chose.

My final conclusion is that the project did not fail in a simple way. The enclosure was designed and printed successfully, the DSP code was written and uploaded, and the physical prototype reached assembly. The point of failure was the final power integration, most likely around the USB-C supply stage and/or my unverified soldering. This is unfortunate, but this is the process of prototyping, it shows exactly what needs to change in the next revision. I am still going to keep working on it because the idea still interests me and it would be a shame to get this far and to call it quits, I especially find the bottom passive radiator and adjustable boundary loading concept fascinating. The next version will be simpler, tested with a mutlimeter in stages, and built around a verified power system before any final assembly.

References

Adafruit (n.d.a) USB Type C Breakout Board - Downstream Connection. Adafruit. Available at: https://www.adafruit.com/product/4090

Adafruit (n.d.b) Adafruit Mono 2.5W Class D Audio Amplifier - PAM8302. Adafruit. Available at: https://www.adafruit.com/product/2130

Arduino (2024a) Bare Minimum code needed. Arduino Documentation. Available at: https://docs.arduino.cc/built-in-examples/basics/BareMinimum/

Arduino (2024b) setup(). Arduino Documentation. Available at: https://docs.arduino.cc/language-reference/en/structure/sketch/setup/

Arduino (2024c) loop(). Arduino Documentation. Available at: https://docs.arduino.cc/language-reference/en/structure/sketch/loop/

Dayton Audio (2026) ND91-8 Specification Sheet. Dayton Audio. Available at: https://www.daytonaudio.com/images/resources/290-226--dayton-audio-nd91-8-specification-sheet.pdf

Dayton Audio (n.d.a) DSA135-PR 5" Designer Series Aluminum Cone Passive Radiator. Dayton Audio. Available at: https://www.daytonaudio.com/product/1589/dsa135-pr-5-designer-series-aluminum-cone-passive-radiator

Dayton Audio (n.d.b) ND105-8 4" Aluminum Cone Midbass Driver 8 Ohm. Dayton Audio. Available at: https://www.daytonaudio.com/product/56/nd105-8-4-aluminum-cone-midbass-driver-8-ohm

Dayton Audio (n.d.c) KAB-100Mv2 1 x 100W Class D Audio Amplifier Board with aptX HD Bluetooth 5.0. Dayton Audio. Available at: https://www.daytonaudio.com/product/1874/kab-100mv2-1-x-100w-class-d-audio-amplifier-board-with-aptx-hd-bluetooth-5-0

Dayton Audio (n.d.d) KABD-4100 4 x 100W All-in-one Amplifier Board with DSP and Bluetooth 5.0 aptX HD. Dayton Audio. Available at: https://www.daytonaudio.com/product/1870/kabd-4100-4-x-100w-all-in-one-amplifier-board-with-dsp-and-bluetooth-5-0-aptx-hd

Dickason, V. (2006) The Loudspeaker Design Cookbook. 7th edn. Peterborough, NH: Audio Amateur Press.

Digi-Key TechForum (2020) Simple way to use USB Type-C to get 5V at up to 3A (15W). Digi-Key TechForum. Available at: https://forum.digikey.com/t/simple-way-to-use-usb-type-c-to-get-5v-at-up-to-3a-15w/7016

Eargle, J. (2003) Loudspeaker Handbook. 2nd edn. Boston: Kluwer Academic Publishers/Springer.

Formlabs (n.d.) FDM vs. SLA vs. SLS: 3D Printing Technology Comparison. Formlabs. Available at: https://formlabs.com/blog/fdm-vs-sla-vs-sls-how-to-choose-the-right-3d-printing-technology/

Hill, G. (2019) Loudspeaker Modelling and Design: A Practical Introduction. New York: Routledge.

ICCFL (n.d.) BMR 12W 8Ω NXT Audio speaker 2.5" AUBA05809-0002. ICCFL. Available at: https://www.iccfl.com/product_info.php?products_id=41257

PJRC (n.d.a) Teensy Audio Library. PJRC. Available at: https://www.pjrc.com/teensy/td_libs_Audio.html

PJRC (n.d.b) Audio Connections and Memory. PJRC. Available at: https://www.pjrc.com/teensy/td_libs_AudioConnection.html

PJRC (n.d.c) Audio Adaptor Boards for Teensy. PJRC. Available at: https://www.pjrc.com/store/teensy3_audio.html

PJRC (n.d.d) Teensy 4.0 Development Board. PJRC. Available at: https://www.pjrc.com/store/teensy40.html

Protolabs Network (n.d.) FDM 3D Printing Through Protolabs Network. Protolabs Network. Available at: https://www.protolabs.com/en-gb/services/3d-printing/
RTINGS (2022) SOUNDBOKS Go Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/soundboks/go
RTINGS (2023a) Our Speaker Sound Tests: Dynamics. RTINGS.com. Available at: https://www.rtings.com/speaker/tests/sound/dynamics
RTINGS (2023b) Our Speaker Sound Tests: Frequency Response Accuracy. RTINGS.com. Available at: https://www.rtings.com/speaker/tests/sound/frequency-response-accuracy
RTINGS (2024a) Bose SoundLink Max Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/bose/soundlink-max
RTINGS (2024b) JBL Xtreme 4 Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/jbl/xtreme-4
RTINGS (2024c) Anker Soundcore Motion Boom Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/anker/soundcore-motion-boom
RTINGS (2024d) Harman/Kardon Onyx Studio 8 Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/harman-kardon/onyx-studio-8
RTINGS (2024e) Ultimate Ears WONDERBOOM 4 Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/ultimate-ears/wonderboom-4
RTINGS (2024f) Ultimate Ears MEGABOOM 4 Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/ultimate-ears/megaboom-4
RTINGS (2024g) Anker Soundcore Select 4 Go Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/anker/soundcore-select-4-go
RTINGS (2025a) How We Test Speakers: Transforming Data into Comparable Reviews. RTINGS.com. Available at: https://www.rtings.com/speaker/learn/how-we-test
RTINGS (2025b) Tribit Stormbox Micro 2 Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/tribit/stormbox-micro-2
RTINGS (2025c) JBL Flip 7 Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/jbl/flip-7
RTINGS (2025d) JBL Charge 6 Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/jbl/charge-6
RTINGS (2025e) EarFun UBOOM X Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/earfun/uboom-x-speaker
RTINGS (2025f) Marshall Middleton II Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/marshall/middleton-ii
RTINGS (2026a) LG xboom Bounce Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/lg/xboom-bounce-bluetooth
RTINGS (n.d.) JBL Authentics 500 Speaker Review. RTINGS.com. Available at: https://www.rtings.com/speaker/reviews/jbl/authentics-500
SoundImports (n.d.) Dayton Audio KABD-4100 4 x 100W All-in-one Amplifier Board. SoundImports. Available at: https://www.soundimports.eu/en/dayton-audio-kabd-4100.html
Stoffregen, P. (n.d.a) PassThroughStereo.ino, Teensy Audio Library. GitHub. Available at: https://github.com/PaulStoffregen/Audio/tree/master/examples/HardwareTesting/PassThroughStereo
Stoffregen, P. (n.d.b) filter_biquad.h, Teensy Audio Library. GitHub. Available at: https://github.com/PaulStoffregen/Audio/blob/master/filter_biquad.h