## Alles build info

See [Alles BOM](https://docs.google.com/spreadsheets/d/1Vnr_DOK79qasKqgbS-QxVXTVbeHfjhG20f4hVg7nSzU/edit#gid=0)

[There’s an EagleCAD project here](https://github.com/bwhitman/alles/tree/master/pcbs). This is a PCB that has a USB-A plug on one end for battery, not what I’d like for a final design. I have not confirmed this design to be working, but I’ve verified the connections in it.

### Battery / Charging / Power

The 98357A needs 5V. (It turns on at 3.3 but then the amplifier is lousy at loud volumes.) The ESP32 needs 3.3V.

In the current Alles implementation, I use a [ESP32 dev board](https://www.amazon.com/gp/product/B07Q576VWZ/) that has a voltage regulator, powered over USB. I use the USB power pin to power the 98357A. I use [a cheap USB battery](https://www.amazon.com/gp/product/B00MWU1GGI) that luckily does not do low current shutoff. This works fine but has reliability issues: the micro USB jack on the dev board is of variable quality, as is the USB A female slot on the battery. And micro-USB cables are also often faulty.

I would like to get to a more permanently wired battery. If shipping batteries with the unit is hard, perhaps a JST connector on the PCB for a lipo battery and a charging / boost IC for the ESP32 and 98357A?

If we take an existing speaker and replace the main board, it might be possible to re-pack them into their original packaging and ship them that way.

A second option for a replaceable battery is to use an 18650 cylindrical battery, and a battery holder on the board that they can fit into.

### Case

Best would be an existing bluetooth mono speaker case that sounds “OK” -- since there’s hundreds of these the individual acoustics are not too important. A 4 ohm speaker should be best. [Something like this](https://www.amazon.ca/Bluetooth-Farsaw-Portable-Wireless-Handsfree/dp/B085R9MP4H/ref=sxin_9_sxwds-deals-bau?cv_ct_cx=bluetooth+speaker&dchild=1&keywords=bluetooth+speaker&pd_rd_i=B085R9MP4H&pd_rd_r=ccf78902-a2a2-4078-8a9e-1151a8640b7c&pd_rd_w=gh119&pd_rd_wg=fbWPj&pf_rd_p=39bac3f0-4d33-4681-8bc9-3a08fe95537e&pf_rd_r=MZD084G1STT4FZ52A07P&psc=1&qid=1602344283&sr=1-1-30cfe86c-7394-4cfa-835a-285db8139249).

The case should have some sort of hanging or mounting hole, like a strap hook or we can place sticker velcro on them after.

We’d want to make sure the PCB antenna on the ESP32 is not too attenuated from whatever case we end up using.

Some alternative case designs (click for links):  
[](https://www.bol.com/nl/p/bestdeal-bluetooth-speaker-model-360-silver/9200000080357843/)[](https://www.bol.com/nl/p/rapoo-bluetooth-speaker-a300-yl/9200000035782648)[](https://www.bol.com/nl/p/esperanza-bluetooth-speaker-ritmo-rood/9200000063710880/)

The original one is readily available on Aliexpress (‘A70 bluetooth speaker’):

<https://www.aliexpress.com/item/33060503268.html>

<https://www.aliexpress.com/i/4000048777199.html>

The plastic/fake wood one is probably best for avoiding antenna attenuation.

### Programming

For now, I program my ESP32 flash over USB in the dev board. I don’t expect end users to have to do this too regularly. We will have to flash each one manually at the start. So a programming port of some sort has to be broken out.

You can program the ESP32 directly over serial using a 6 pin header laid out like the [Sparkfun FTDI basic dongle](https://www.sparkfun.com/products/9873). The programming software does something with DTR and RTS to get the MCU in a bootloader. So if a 6 pin male programming header was on the PCB we’re all set.

While it would be more convenient for end users, I assume getting USB programming going would require a USB serial IC and more complexity / cost. Maybe we can price it out to see. Having just one USB C receptacle on the back for charge, programming, power would be the best case.

It’s $1-2 for the USB-to-Serial converter chip + reset transistors. Since there should be a USB port for charging anyway, it might make sense to put the footprints for those parts on the board, as well as the FTDI header, so you could choose to populate one or the other.

### MIDI

Not a priority, but one use case of Alles is where a single speaker receives MIDI input (on GPIO 19, via a optoisolator) and it acts as a bridge for the rest of the wireless mesh, forwarding on messages over UDP. This allows controlling the mesh from any sequencer or hardware synth. This is not needed for my immediate use case, but it may make sense to also put a 3 pin male header on the PCB for GPIO 19 (midi in), VCC (5V or 3.3V) and GND.

Sure, that’s reasonable.