




JC

 Date @November 21, 2023

goals:

- ~~learn how to use kicad~~
- ~~watch cheru use kicad to build pcb and actually understand what's going on~~
- ~~be able to explain what the openbc1 pcb does~~
- get a list of parts

wack questions and thoughts and todos

- learn how to be able to read datasheet
- a collaborative pcb editor with history and other modern features would be useful

how to use kicad

build: traffic light pcb & sparkletilt pcb to understand how to design pcbs

(really long jam! should work on it more later)

<https://youtu.be/EPH23zhPg50?si=b0f7BcteMYxaCMWL>

<https://jams.hackclub.dev/batch/sparkletilt-pcb/part-1>

in addition, the rest of the workshop assumes you know basic electronics terms like:

1. resistor: a resistor limits the electric current that flows through a circuit.
2. capacitor: a capacitor stores charge, and thus energy. it's like a tiny battery that charges and discharges very quickly.
3. diode: a diode only allows current to flow in one direction. it also drops a little bit of voltage, turning that energy into heat.

part selection

1. select core components, for example:

- a. micorcontroller: the atmega series is the most robust and common 8-bit MCUs, being used in many Arduinos since the beginning. additionally, it requires very few external components, is a basic part on jlcpcb, and is available in qfp (quad flat package)
 - b. usb-serial interface: ch340n.
 - c. keep in mind pinout
- nets like VCC (voltage common connector) and GND serve as abstractions for actual connections in our schematic. if we connected every single chip to a central VCC and GND point in our schematic, it would be very messy. these symbols tell the PCB designer that we need to connect those pins while keeping the schematic clean.
 - decoupling capacitors: aka bypass capacitors. a decoupling capacitor is placed very close to the chip that needs or supplies power. when a chip suddenly demands power, it provides it while the battery and other components ramp up, essentially like a dam controls a river. when there are voltage spikes or noise, it absorbs the the, while letting normal DC power pass along. So, these have to be as physically close to their parent IC as possible in the PCB layout
 - how to come up with the value of these capacitors?
 - $C = 2 * CL - CS$

turn a kicad schematic into a pcb

- traces: a continuous stretch of copper that connects multiple parts
- layers: layers are parallel surfaces of stuff on your pcb. there are a bunch of diff. kinds of layers, like:
 - F.Cu/B.Cu: The layers of copper on your pcb. the conductive traces on your board should live on these layers. separate conductive traces can not cross each other on the same layer, you must use a via to connect two layers and not cross traces on the same side of the board.

- F.Silkscreen/B.Silkscreen: The printed (silkscreened) layers on your pcb. these are either white or black and must be monochrome (you can't have gradients). anything can go on these layers
 - F.Paste/B.Paste: Anything that will be soldered needs to have paste on it. You probably don't need to mess with this layer unless you know what you're doing, it should be automatically handled by your part footprints.
 - F.Mask/B.Mask: The soldermask is the colored coating that goes on the board. by default, it covers everything except pads and is what gets printed on. removing soldermask can be used to expose copper for cool effects.
 - objects > ratlines: these have no impact on the final production of your board, they are simply guides to tell you where you still need to draw traces.
 - edge.cuts: this is the actual outline of how your board will be cut.
 - user.comments: this layer has information that will not be printed on the board and is only visible in kicad.
 - reminders:
 - make sure you draw your traces. you shouldn't have any ratlines on your board when you send it for production.
 - remember separate conductive traces cannot touch each other.
 - footprints = exposed copper pads on the pcb for each component
-

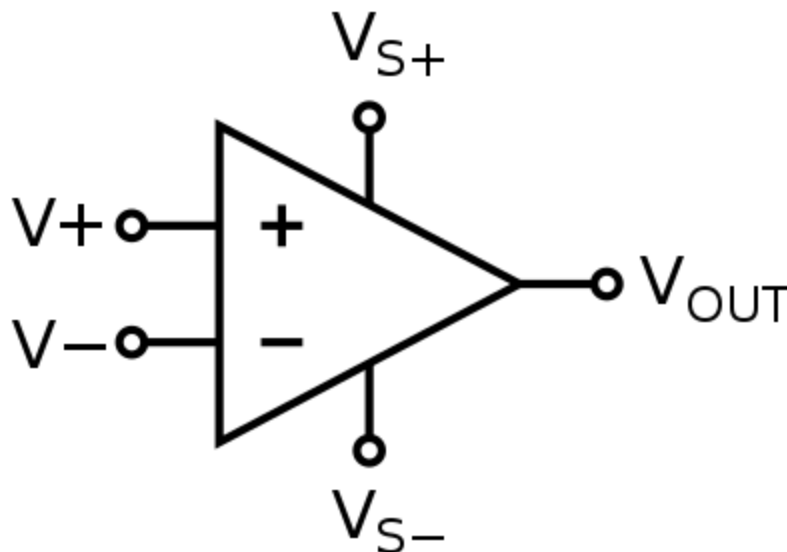
looking at other eegs

- hackEEG
- piEEG
- cyton
 - cheru tries to explain current pcb work to me
 - most are from snapmagic
- bill of materials from lcsc and digikey
- esp32-s2-mini-1 is the microcontroller we're using

- ads1299 (analog to digital [sensor?] 1299 from texas instruments converts analog to digital)
 - 8 channels to get signal from electrode
 - signals from the electrodes go through the filters (negative and positive) = 16 total
 - potentially using electrostatic discharge
-

understanding op amps

- an operational amplifier is a dc-coupled high-gain electronic voltage amplifier with a differential input and usually a single-ended output
- in this config an op amp produces an output potential that is typically 100,000 times larger than the potential difference between its input terminals
- op amps traces its origin and name to analog computers, where they were used to perform mathematical ops in linear, non-linear circuits
- symbol



- the amplifier's differential inputs consist of a non-inverting input (+) with voltage V_+ and an inverting input (-) with; ideally the op amp amplifies only the difference in voltage between the two, which is called the *differential input voltage*.

front end filtering

<https://openbci.com/forum/index.php?p=/discussion/154/front-end-filtering-esd-cmrr-rfi>

the resistor-capacitor low-pass filter components on the circuit augment the digital sinc filter inside the chip at frequencies above 1mhz.

electrode impedances in clinical practice can typically vary between a few kOhm through upwards of 50kOhm and this in combination with the filter capacitor will cause CMRR ratios

In Which I Attempt To Explain Everything

(Will be improved over the coming days.)

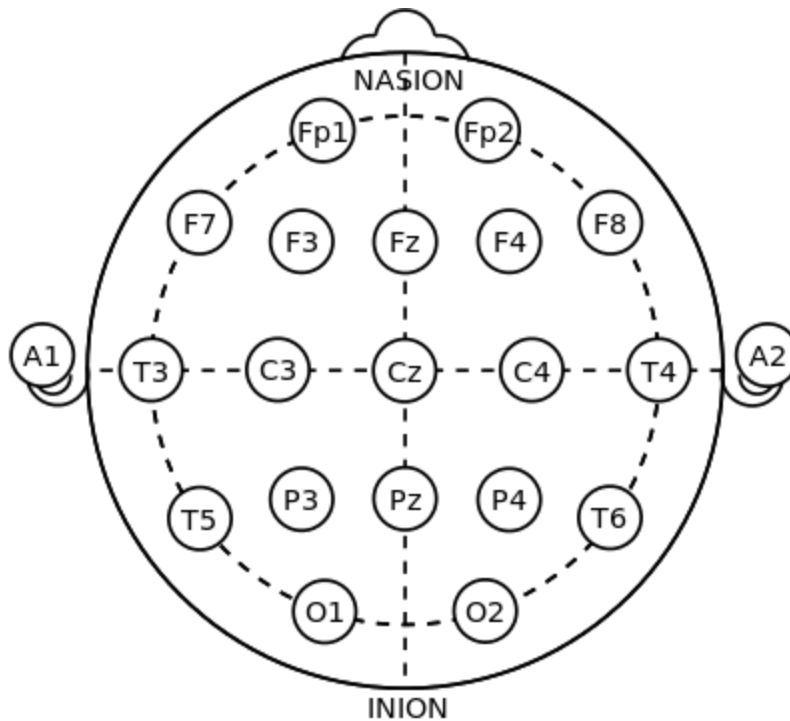
What exactly are we planning to do? During week one, we'll be designing a board similar to OpenBCI's Cyton board among other initiatives out there, like hackEEG and piEEG, that will allow us to read brain waves using EEG. The significance of this board is that it will be significantly more financially viable compared to the rest of the market. We'll order this board through JLCPCB (with parts sourced from LCSC and DigiKey) and assemble it during the second week, which will be more focused on software: getting readings to work; designing a programmable API wrapper around the board; and finally using the board with electrodes we already own to play a game, e.g. Flappy Bird.

What is EEG? EEG is short for electroencephalography, and it's a method that can be used to record electrical activity in the brain from a bird-eye's perspective. A better analogy, perhaps, is to compare it to "the aggregate flow of activity on the Internet, where you can't read any individual email but can see macro traffic patterns" (Sapien Labs).

There are various applications of this, but the one that stands out to me the most is using EEG to measure and improve focus. From ACX: "In 2022, a team in Cambridge found that experimental subjects learned faster stimuli... when presented at their brain's unique alpha rhythm. The scientists monitored their brain waves to figure out exactly what each subject's alpha rhythm was (usually a pattern of flashes about a dozen times

per second), then presented a flashing pattern that hit the trough of each alpha wave, then asked subjects to solve tough visual recognition problems. They found the alpha entrainment helped them learn faster... Consumer grade EEG headbands could potentially be used to replicate this result... but someone should figure out whether this can be used for the sorts of things normal people want to learn.”

Back to the technicalities of EEG. We start with non-invasive electrodes that are placed along the scalp. Placement diagrams are typically done using the international 10-20 system, where a measurement is taken across the top of the head from the nasion (point above the nose between the eyes) to the inion (the crest point of the back of the skull). It looks like this:



Each of the circles represents a potential electrode placement and has a letter to identify it by - for example, FP = pre-frontal; F = frontal; T = temporal; P = parietal; O = occipital; and A = the prominent bone usually found just behind the outer ear. These all match up to various parts of the brain that are responsible for sending different signals.

The signals from these electrodes typically go through electrical components known as op-amps to filter out potential interference from other factors, such as moving your head (this is why most EEG boards will also come with an accelerometer to cancel out said

factor). Each electrode is considered to either feed into a positive or negative channel.
How do op-amps work?

After filter(s) are applied, the electrical signal, which is currently in analog form, needs to be converted to a digital signal. Most commercial EEG boards use some variant of the ADCs (analog-digital converter) that Texas Instruments manufacture; in our case, we're looking at the [ADC1299](#), which has support for 8 channels.

It then goes through a microcontroller that transfers the data to a computer - through WiFi or direct transfer, or some variant of both. It's important to prevent interference from this as well, so having an external antenna or having the antenna be directly on the board is important. In our case, something like the [ESP32S2MINI1U](#) will suffice.

And boom! You have an EEG board. (Easier said than done.) From there it's more software-based.