

KiCad Crash Course

Jan Wolf

December 7, 2023

WüSpace e.V.



Table of Contents

1. Intro
2. What is a PCB?
3. PCB Design Tools
4. Schematics
5. PCB Layout
6. Further Topics

Intro

Who is this Nerd?



- jan.wolf@wuespace.de
- Aerospace Computer Science since 2018 (MSc TBD)
- No electrical engineering background, all learning by doing
- WüSpace member since 2019



Who is this Nerd?

Daedalus 2 Electronics





Who is this Nerd?

Daedalus 2 Electronics



BONK



Who is this Nerd?

Daedalus 2 Electronics



BONK



HONK



Motivation

Why am I doing this?

Motivation

Why am I doing this?

Because this guy kept bugging me about it





Goal of this Course

What we'll do:

- Transfer of knowledge! Teach people how to design their own PCBs
- Learning by doing! Everybody will go through the process of designing a PCB

What we won't do:

- Teach how electronics work in general



Goal of this Course

What we'll do:

- Transfer of knowledge! Teach people how to design their own PCBs
- Learning by doing! Everybody will go through the process of designing a PCB

What we won't do:

- Teach how electronics work in general
Read a book or smth idk



Based on HOPE

This course is ~~stolen from~~ loosely based on the Hands-On PCB Engineering (HOPE) course from Berkeley. Take a look at their program, it's great! ieee.berkeley.edu/hope/



University of California, Berkeley
IEEE Student Branch

Hands-On PCB Engineering

Lecture 11:

PCB Manufacturing

Attendance: <https://ieee.berkeley.edu/hopeforms/>

University of California, Berkeley
IEEE Student Branch

What is a PCB?



Contents

1. Intro

2. What is a PCB?

- PCB Examples

- PCB Construction

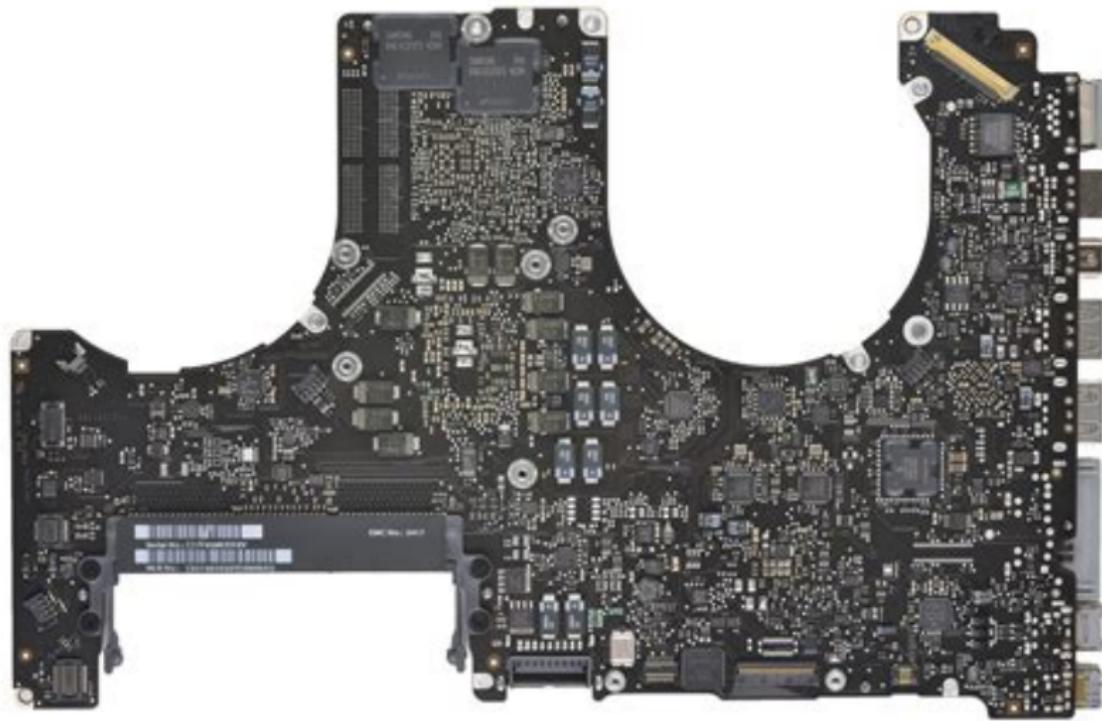
- PCB Features

3. PCB Design Tools

4. Schematics

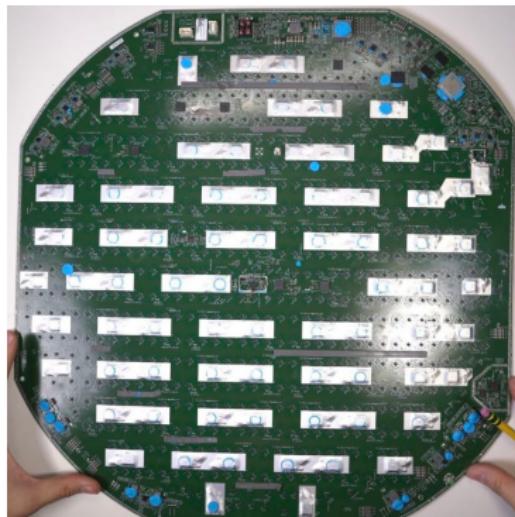
5. PCB Layout

Macbook Pro Logic Board

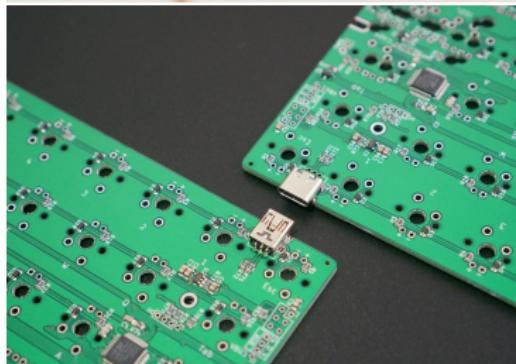
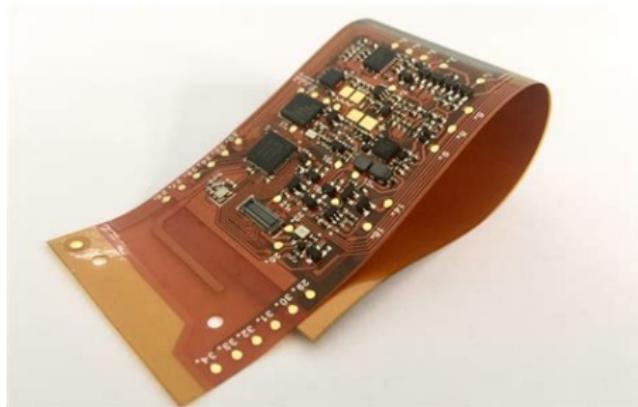




Starlink



And much more





Contents

1. Intro

2. What is a PCB?

- PCB Examples

- PCB Construction

- PCB Features

3. PCB Design Tools

4. Schematics

5. PCB Layout



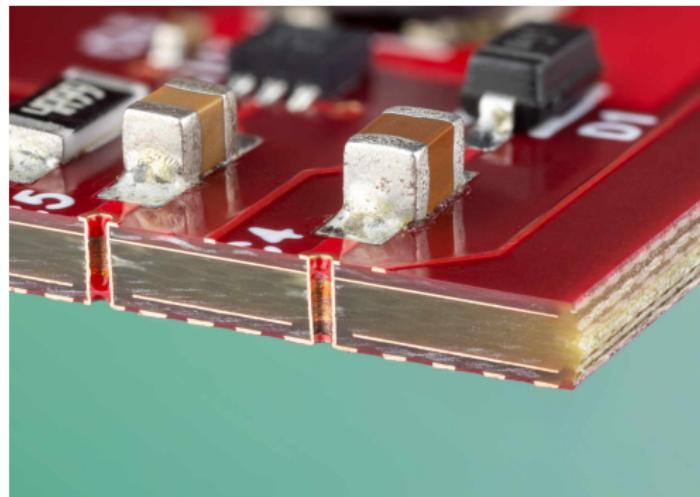
What is a PCB?

- PCBs (printed circuit board) are essential to electronic hardware design
- They provide form (mechanical) and function (electrical)
- Better, cheaper and more reliable than alternatives



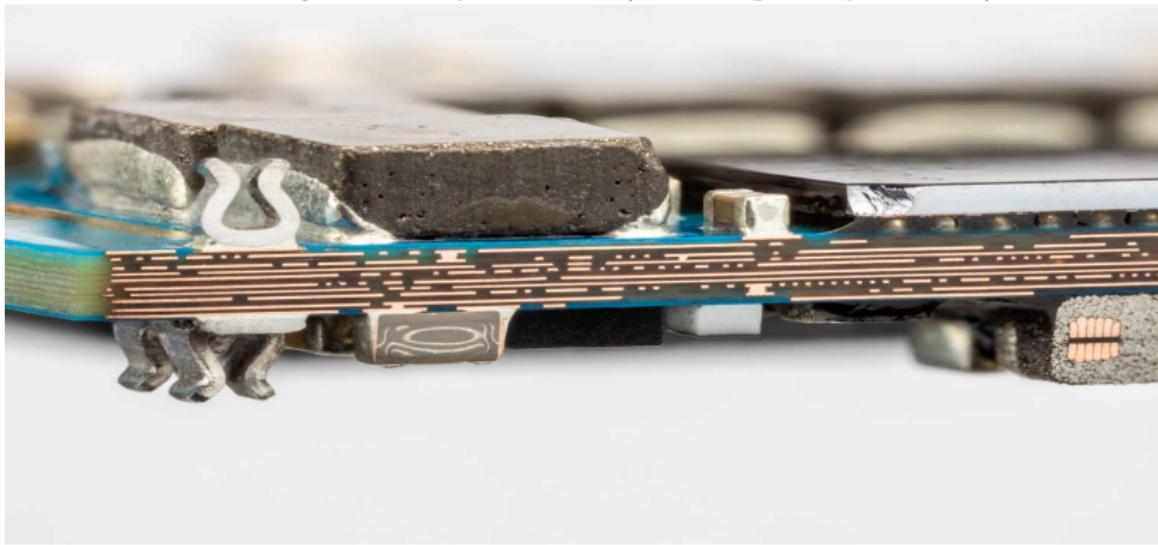
What is a PCB?

Sandwich of conductive layers!



What is a PCB?

More layers are possible (although expensive)



Find more brilliant pictures at www.opencircuitsbook.com/



Contents

1. Intro

2. What is a PCB?

- PCB Examples

- PCB Construction

- PCB Features

3. PCB Design Tools

4. Schematics

5. PCB Layout

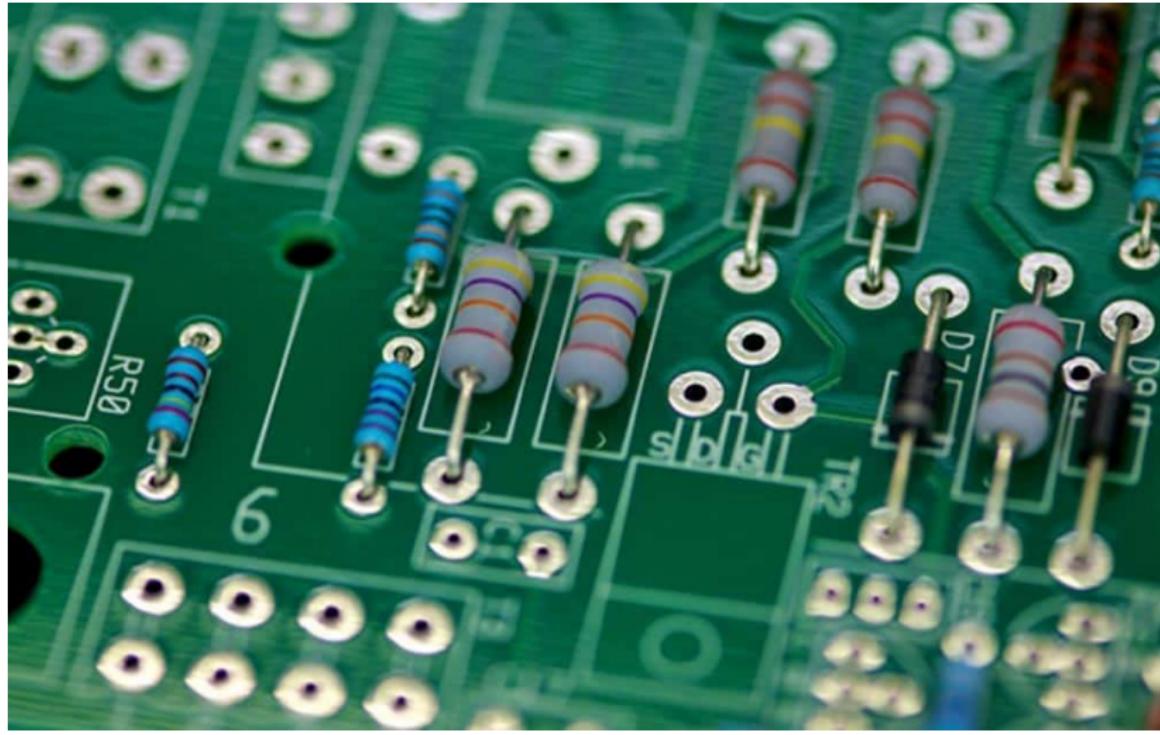


PCB Features

- **Tracks/Traces** are basically wires between components
- **Pads** make electrical connection to components, fat copper areas.
- **Vias** are holes with conductive plating to jump between layers

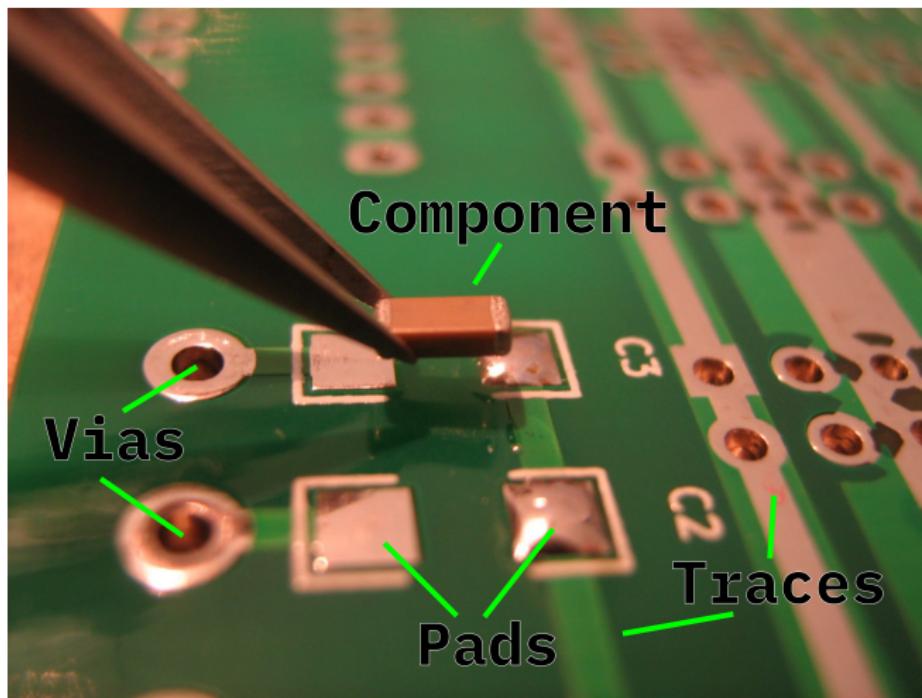
THT Components

Components can be THT (through hole technology)...



SMD Components

or SMD (surface mount devices)



PCB Design Tools



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

- Available ECAD Tools

- KiCad!

- PCB Design Flow

4. Schematics

5. PCB Layout



Terminology

EDA = ECAD

Electronic Design Automation

Electronic Computer Aided Design



Terminology

EDA = ECAD

Electronic Design Automation

Electronic Computer Aided Design

...how to pronounce KiCad?

Terminology

EDA = ECAD

Electronic Design Automation

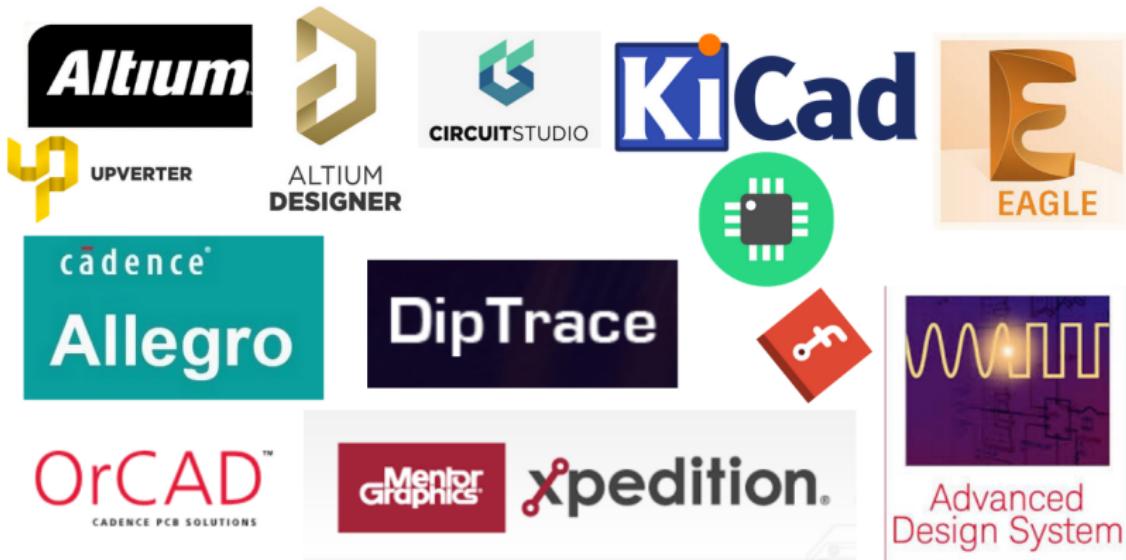
Electronic Computer Aided Design

...how to pronounce KiCad?





Other ECAD Programs are Available





Other ECAD Programs are Available



Some of these can simulate Maxwell's equation, design 20 layer PCBs and cost 40k€/year.



Other ECAD Programs are Available



Some of these can simulate Maxwell's equation, design 20 layer PCBs and cost 40k€/year.

Others are basically MS Paint in a trench coat.



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

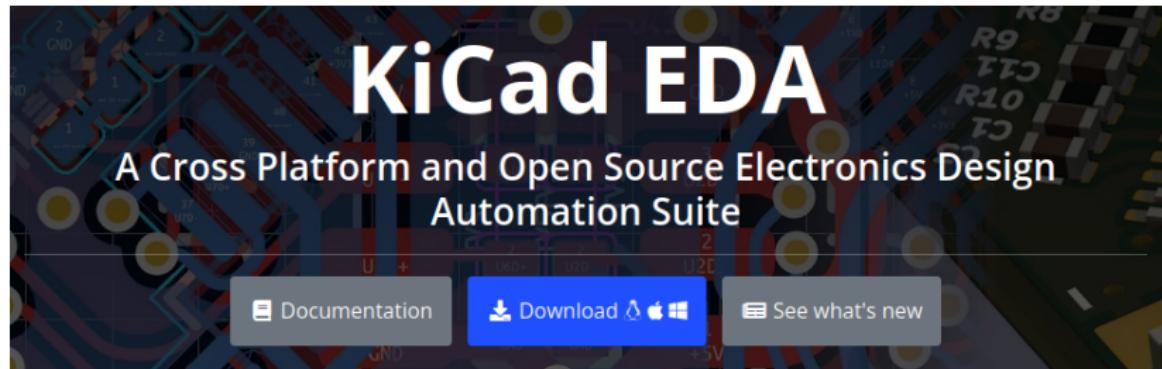
- Available ECAD Tools

- KiCad!

- PCB Design Flow

4. Schematics

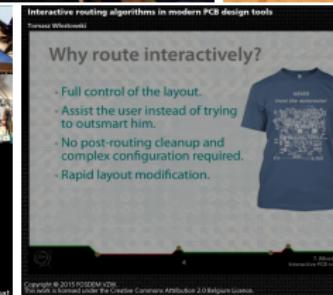
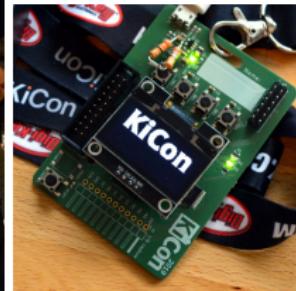
5. PCB Layout



- Free and Open Source (GPLv3)
- Windows, Mac, Linux, BSD compatible
- Released at ITU Grenoble 1992, receives funding by CERN, part of the Linux Foundation
- Easy file formats and open APIs: Lots of compatibility, 3rd party plugins and libraries



KiCad Community



Nerdy hackers and makers available on Hackaday/Social Media/Livestreams/Conferences...



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

- Available ECAD Tools

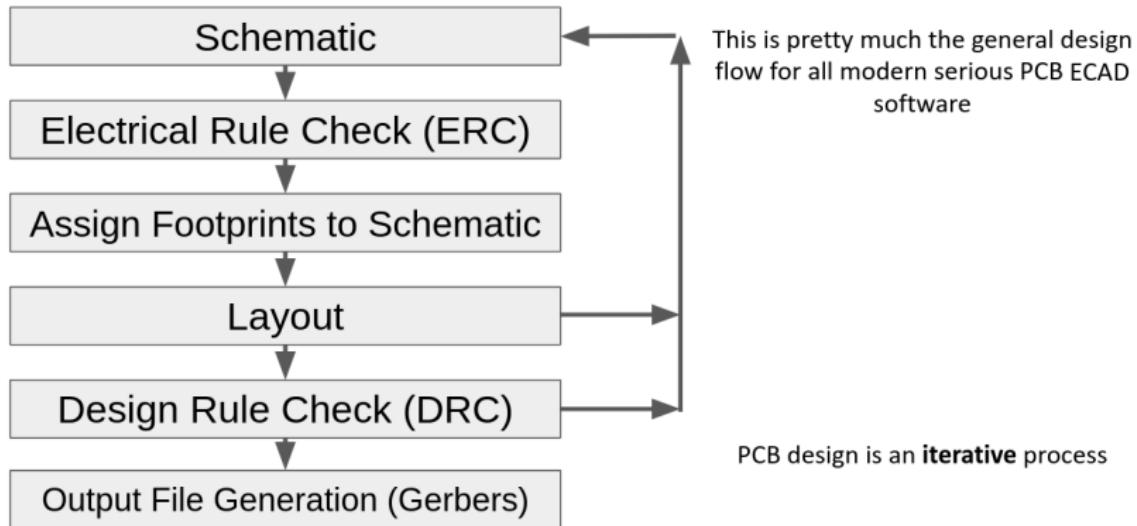
- KiCad!

- PCB Design Flow

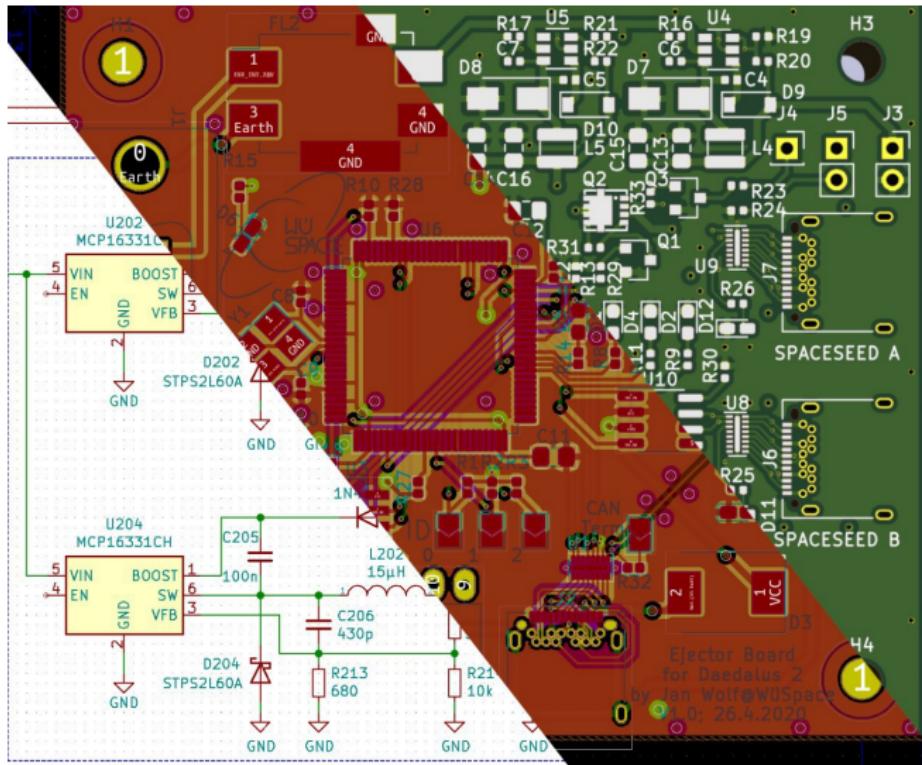
4. Schematics

5. PCB Layout

PCB Project Design Flow

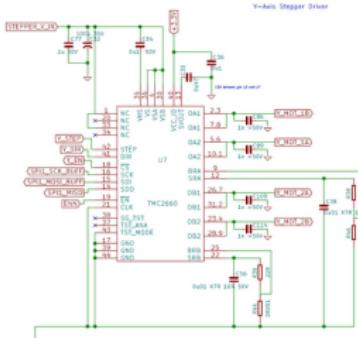


PCB Project Design Flow



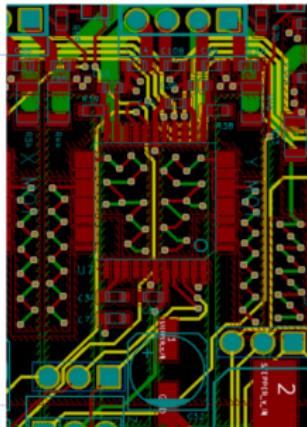
PCB Project Design Flow

1. Schematic



*not the full schematic

2. Layout



*not the full layout

3. Manufactured Board



Schematics



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

4. Schematics

- Schematic Flow

- Schematic Conventions & Etiquette

- Task 1: Designing a Light Sensor

- From Schematic to Physical Components

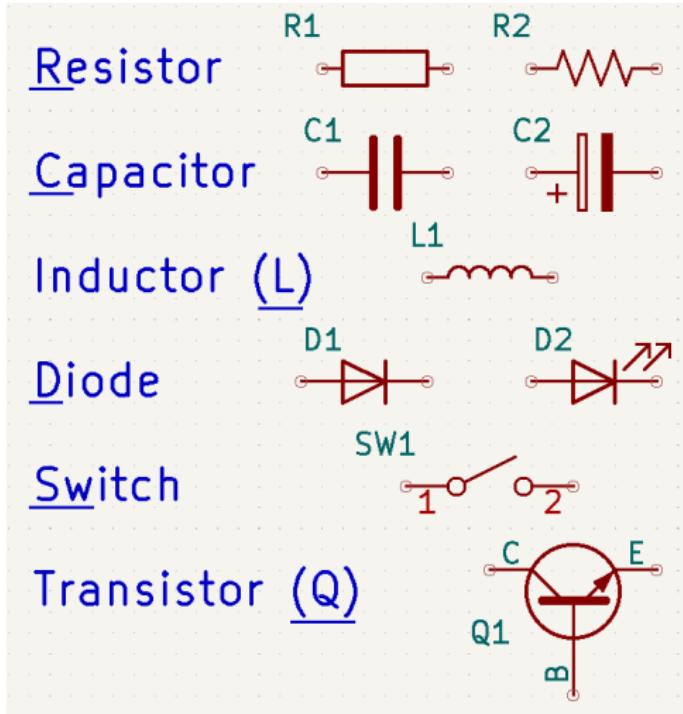


Schematic Flow

Draw schematic

- Describe the circuit in a way both humans and computers understand
- ERC (electrical rule check) can catch errors
- Assign footprints

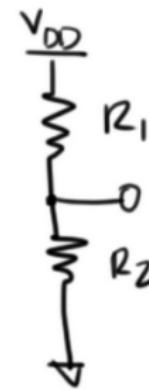
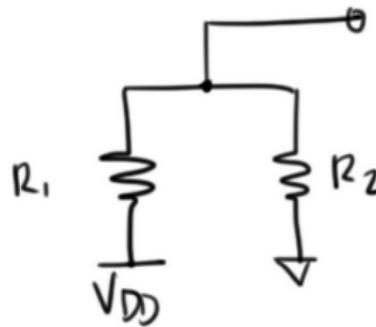
Component Symbols



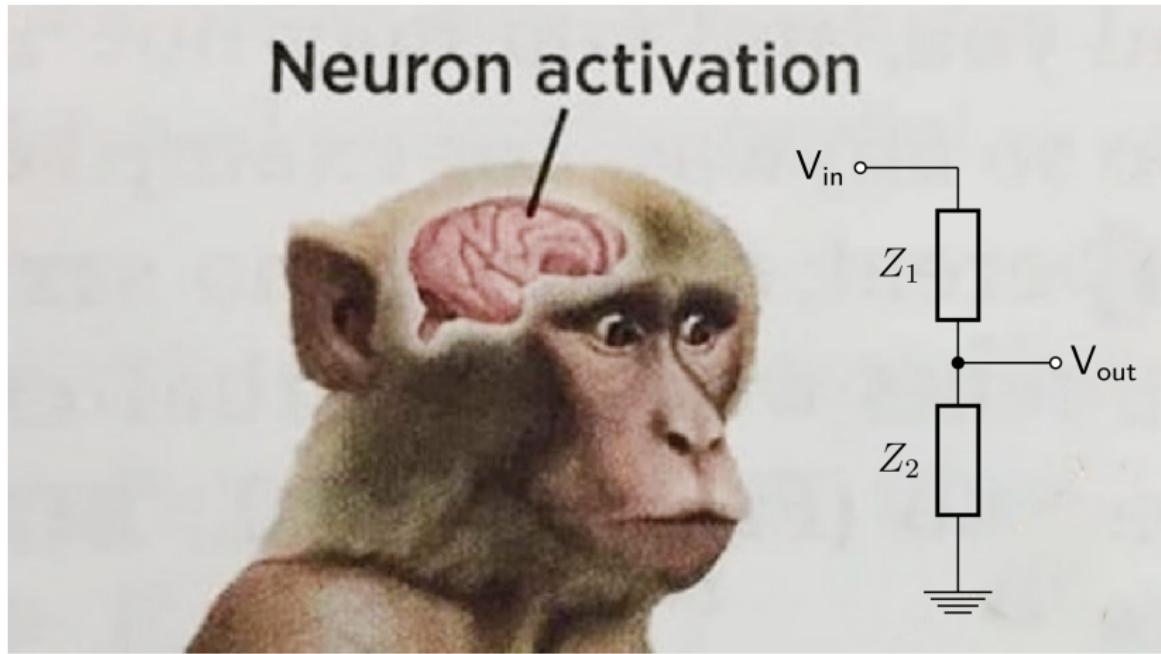
Circuits are built from a few basic components, which follow a symbol “alphabet”.

Do you know any more non-IC components?

Schematic Flow



What does this do? Which one do you prefer?



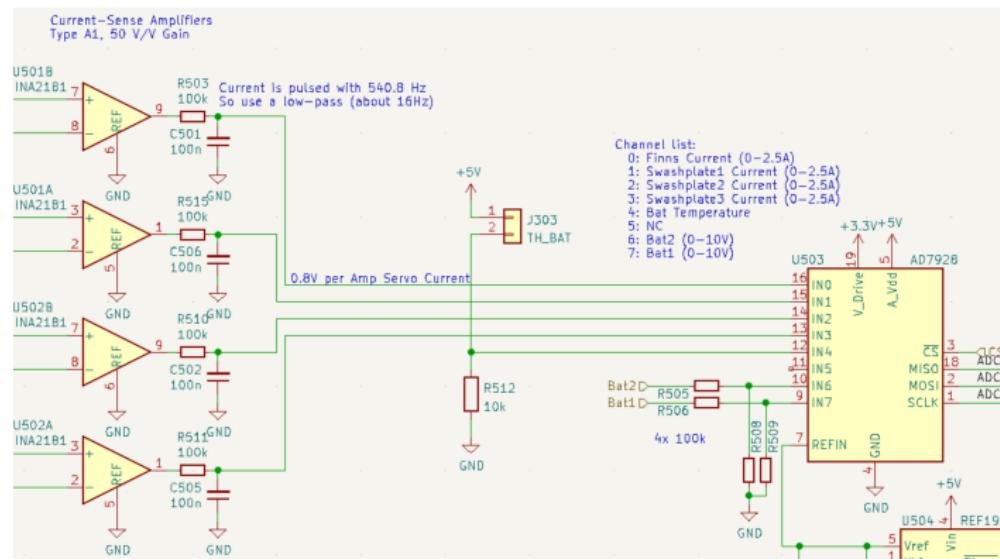
Following conventions makes understanding easier!

Schematic Flow

Try to follow this general pattern:

Power
to
Ground

Input to Output

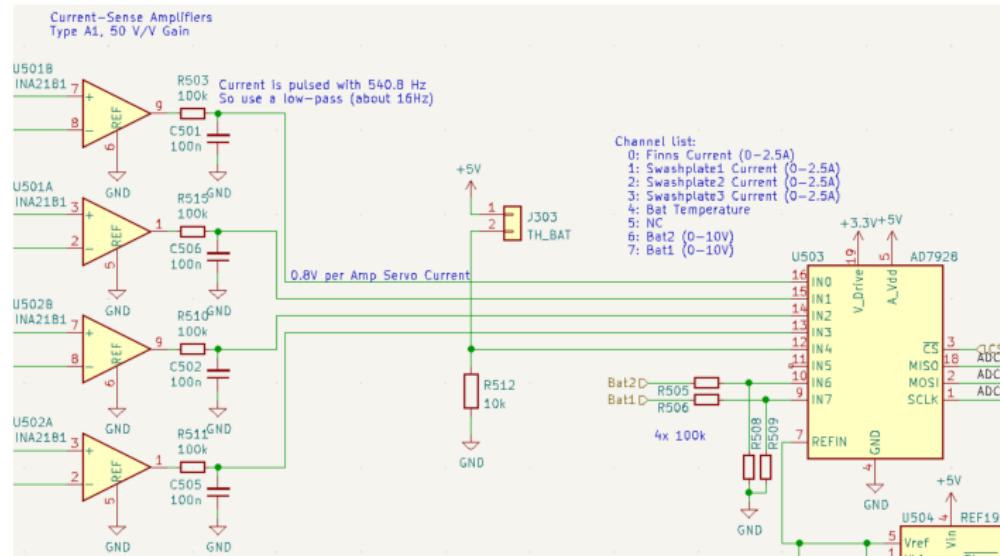


Schematic Flow

Try to follow this general pattern:

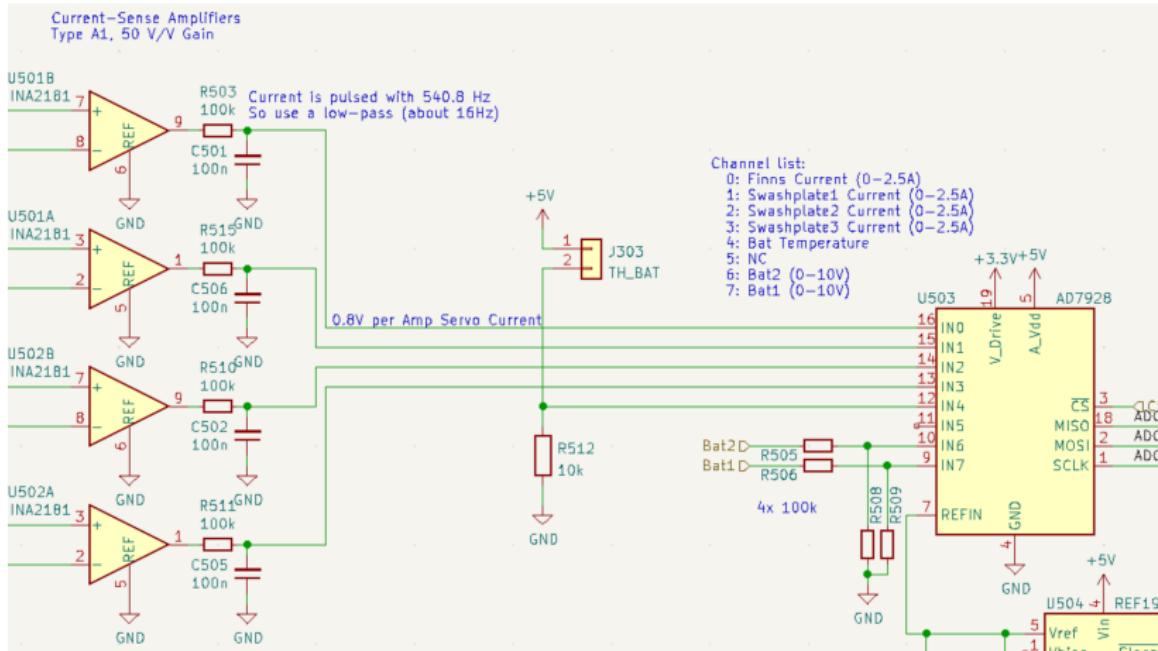
Input to Output

Power
to
Ground

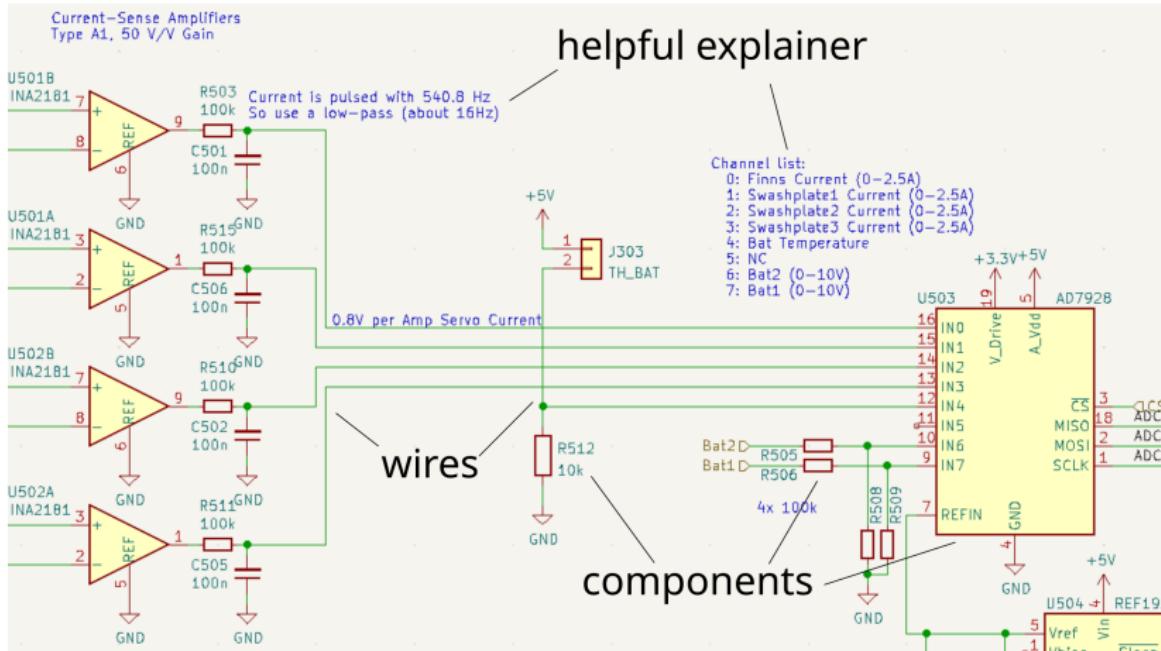


(not always possible)

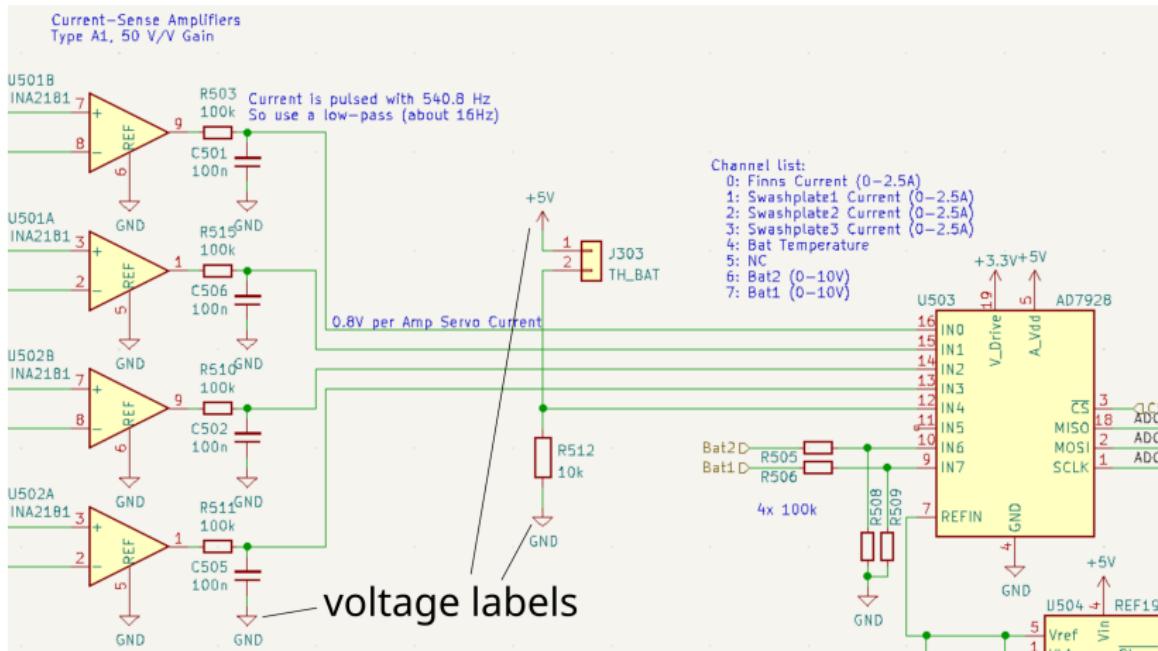
Schematic Flow



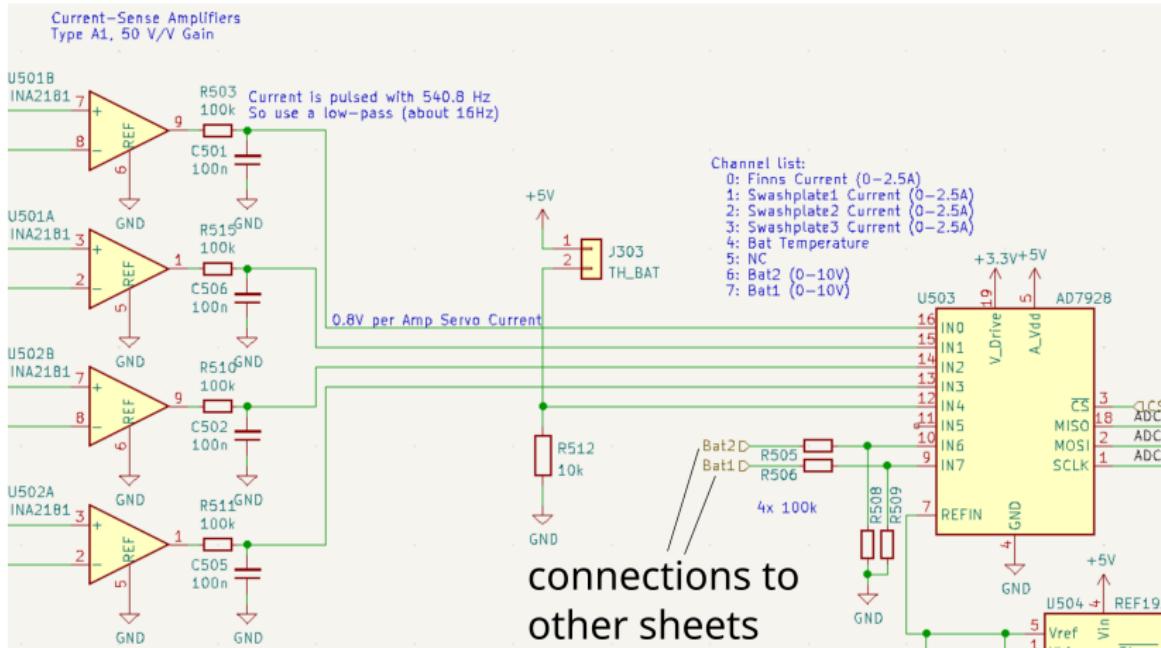
Schematic Flow



Schematic Flow



Schematic Flow





Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

4. Schematics

- Schematic Flow

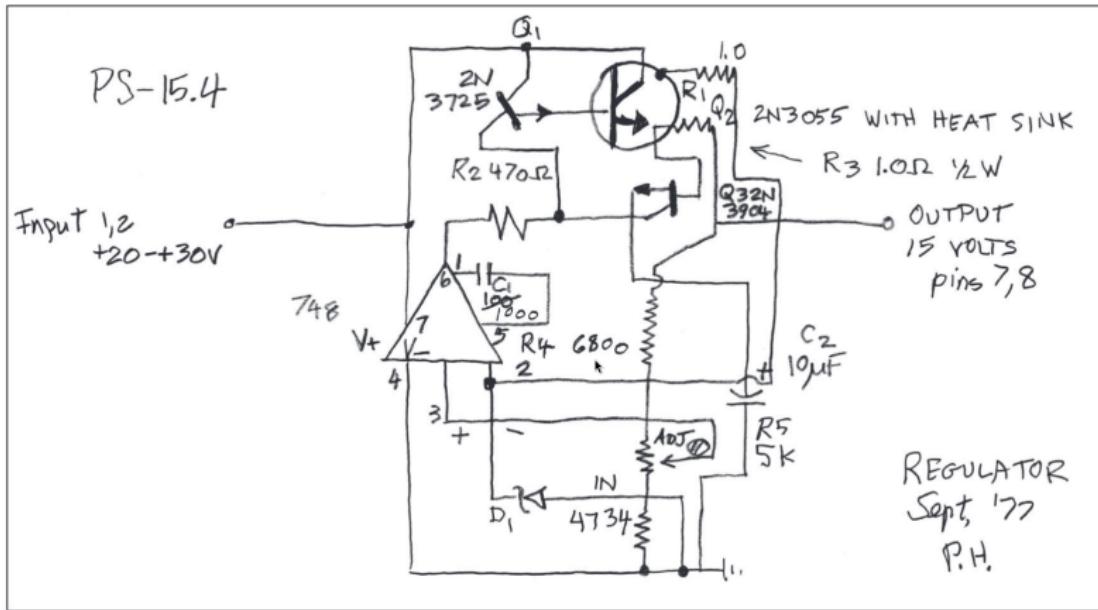
- Schematic Conventions & Etiquette

- Task 1: Designing a Light Sensor

- From Schematic to Physical Components

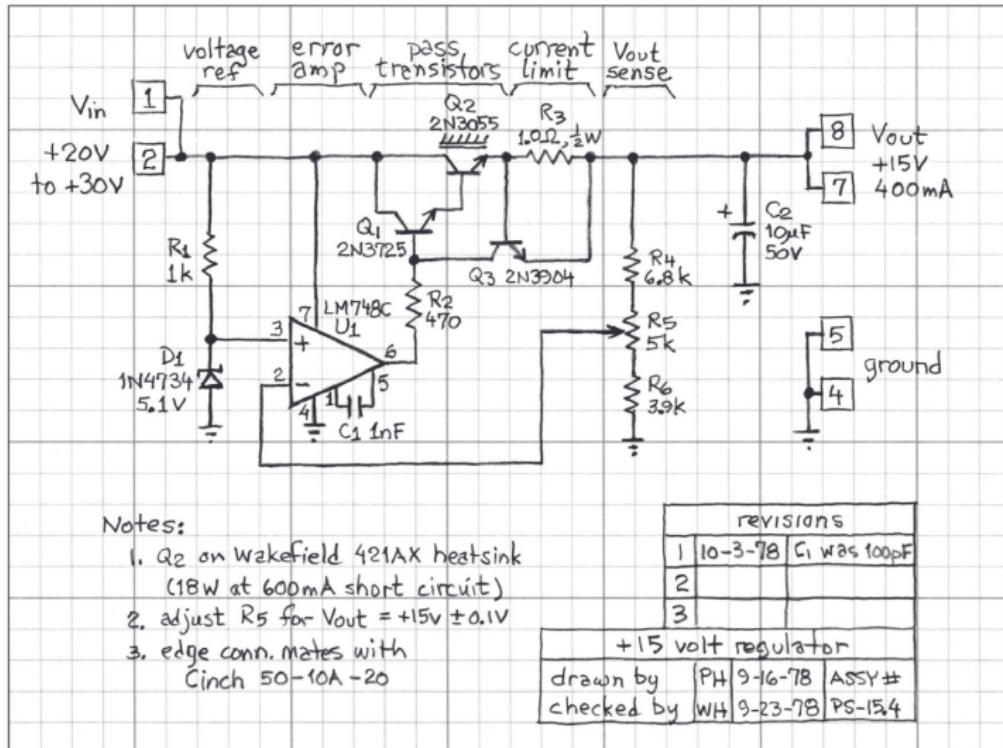
Schematic Etiquette

Don't!



Schematic Etiquette

Do!





Schematic Etiquette

- Drawing nice schematics comes with years of experience, but it's worth it.
- Debugging a neatly drawn circuit is 100x more fun than debugging a confusing rat's nest.



Schematic Etiquette

- Drawing nice schematics comes with years of experience, but it's worth it.
- Debugging a neatly drawn circuit is 100x more fun than debugging a confusing rat's nest.
- It's the same as debugging the horrible spaghetti code you wrote 2 years ago.

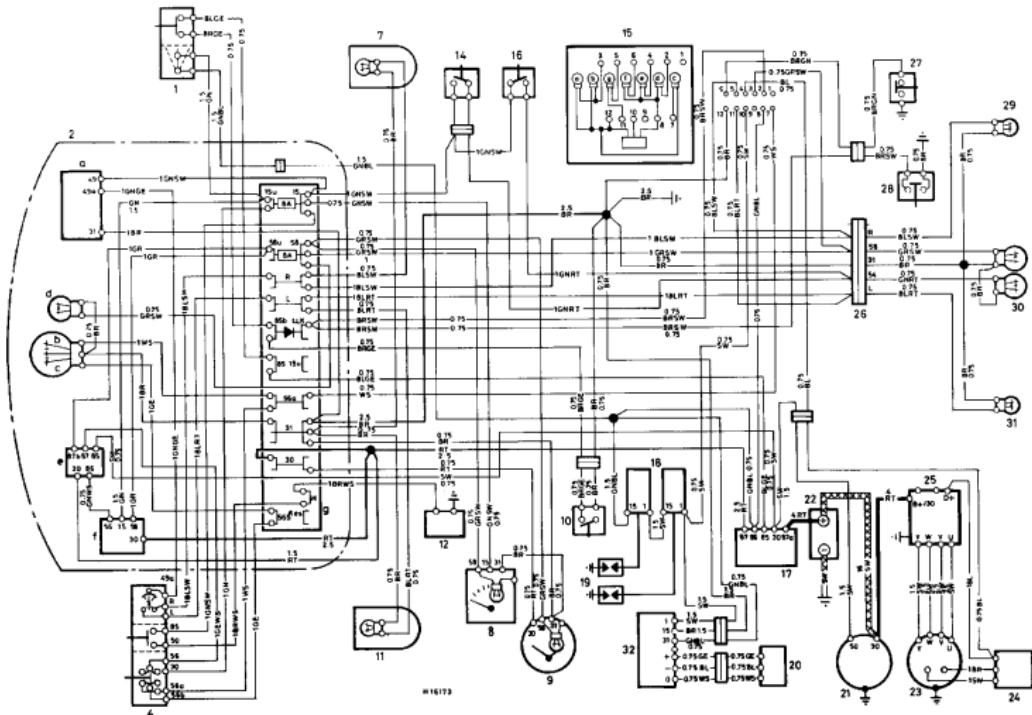


Schematic Etiquette

- Drawing nice schematics comes with years of experience, but it's worth it.
- Debugging a neatly drawn circuit is 100x more fun than debugging a confusing rat's nest.
- It's the same as debugging the horrible spaghetti code you wrote 2 years ago.
- But I'm still messy sometimes, especially if it's a quick&dirty first prototype ;)

Schematic Etiquette

For the love of God, DON'T!



Wiring diagram – 1981 on R100 and R100CS models

See page 266 for component key

Colour key	GR Grey
BL Blue	RD Red
BR Brown	SW Black
GE Yellow	VI Violet
GN Green	WS White



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

4. Schematics

- Schematic Flow

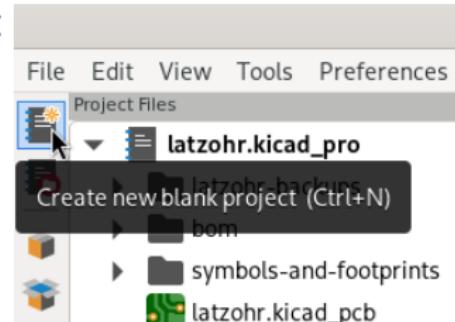
- Schematic Conventions & Etiquette

- Task 1: Designing a Light Sensor**

- From Schematic to Physical Components

Let's begin!

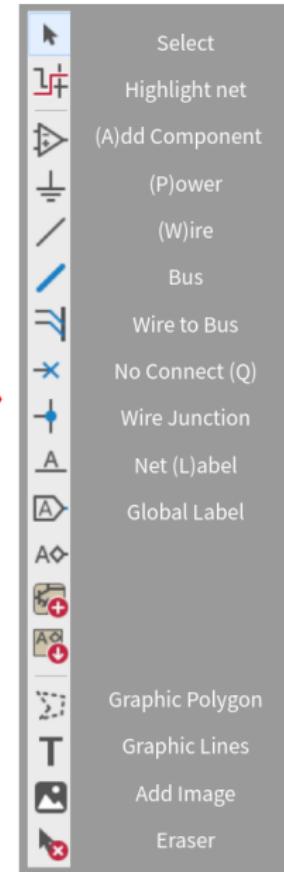
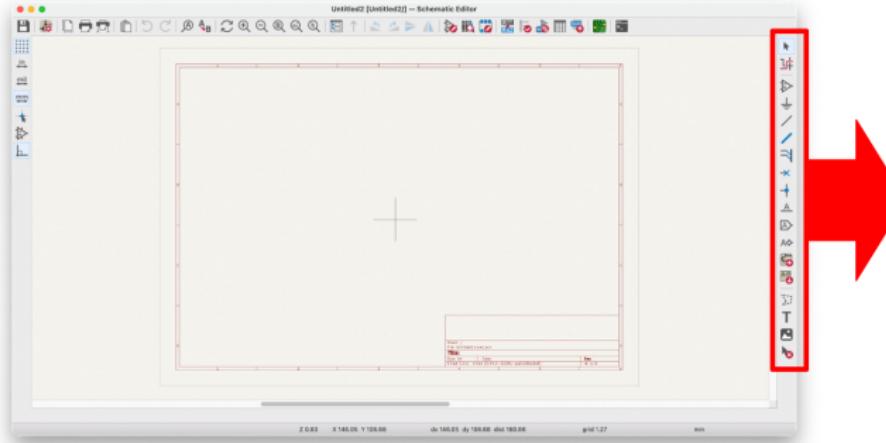
- Open KiCad by clicking on this icon: 
- Create a new project:



- Open the Schematic Editor!

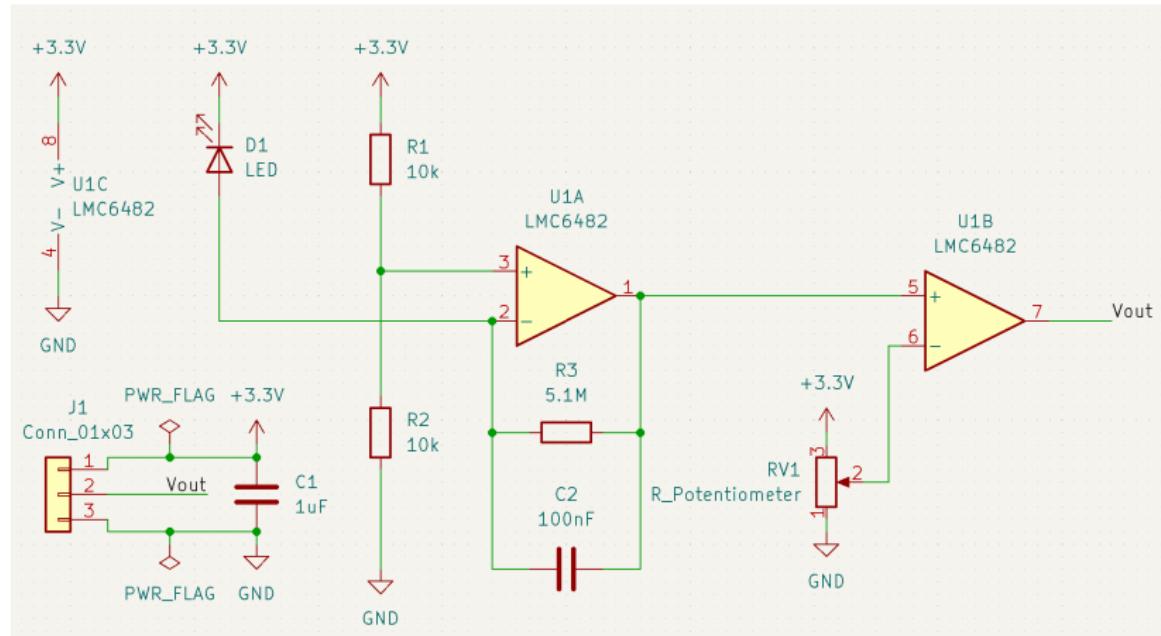


Schematic Commands



Our Goal

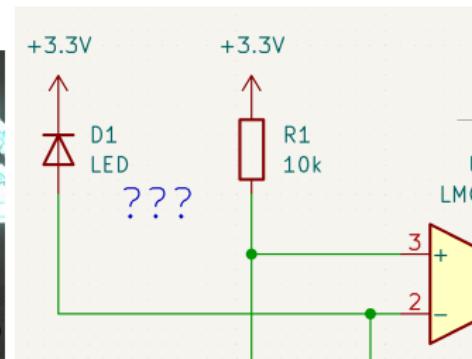
Our goal is to replicate this schematic:



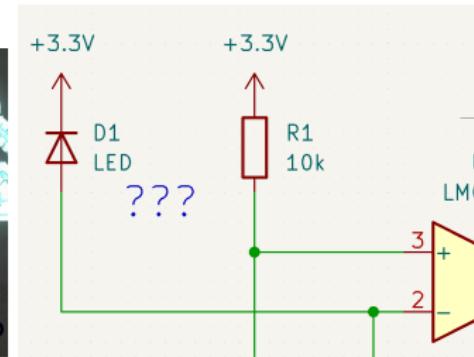
It's a light detector!



Wait a Moment



Wait a Moment



Isn't this LED the wrong way around?

Yeah, but we're cheating a bit. A LED can also work as a light sensor. It generates a negative voltage when light shines on it!



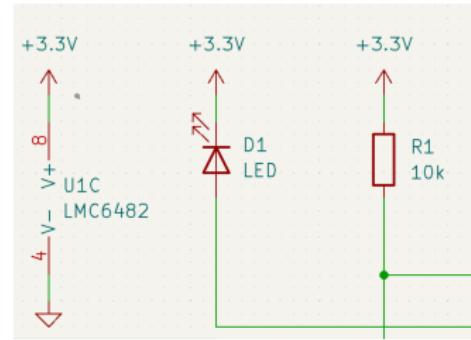
First Component

We will first place the diode D1

- Press the shortcut **a** to add a component and enter **led** into the search bar
- Press **r** to rotate
- If happy, click anywhere to place the component

Use the force shortcuts, Luke!

- Obi-Wan, probably



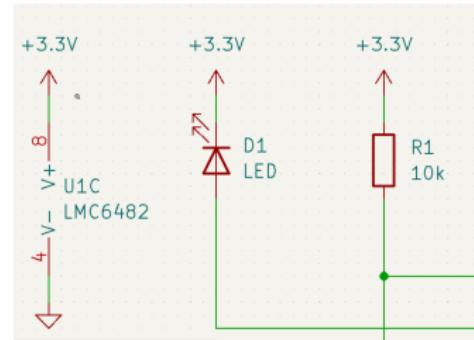
First Component

Now repeat those steps for the resistor **R1**.

- Again, add with **a** and type **r** for resistor.
- Unhappy with the placement? Press **m** to **move** a component

Use the force shortcuts, Luke!

- Obi-Wan, probably





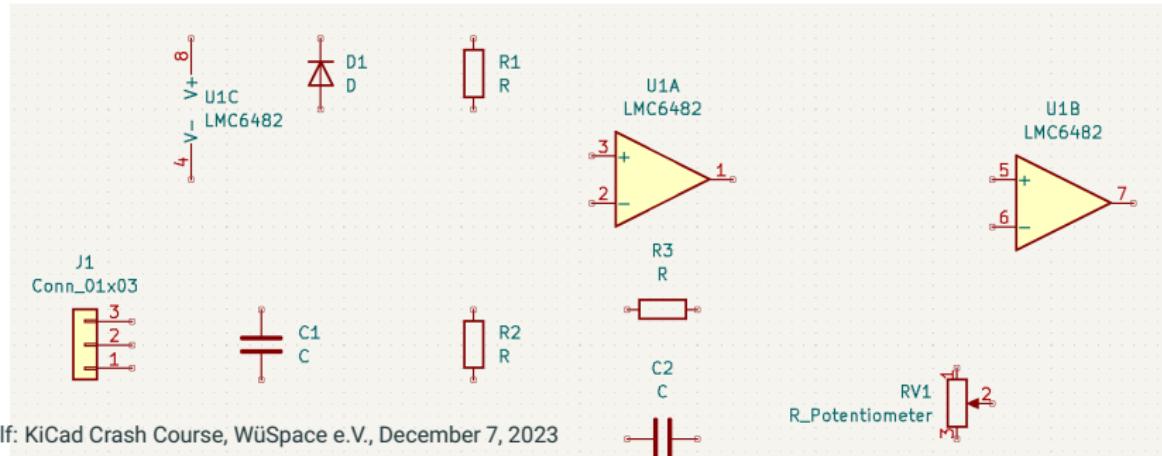
Multi-Symbol Components

- Now we add the operational amplifier (OpAmp).
- Add with a and start typing *LMC6482*
- KiCad will allow you to place multiple components! Why could that be?
- Hint: Look up the LMC6482's datasheet

Placing all Components

In total we need:

- 2 capacitors (C or C_Small)
- 3 resistors (R or R_Small)
- a LED part symbol (LED)
- a potentiometer part symbol (R_Potentiometer)
- a 1x3 connector part symbol (CONN_01x03) - should be listed as generic





Adding Wires

- The values for the resistors and capacitors will be blank (R, C), ignore this for now



Adding Wires

- The values for the resistors and capacitors will be blank (R, C), ignore this for now
- Wire up your components! Press `w` for **wire** to activate the wire tool and press `Esc` to go back to the selection tool.
- Repeat until the schematic is fully captured.

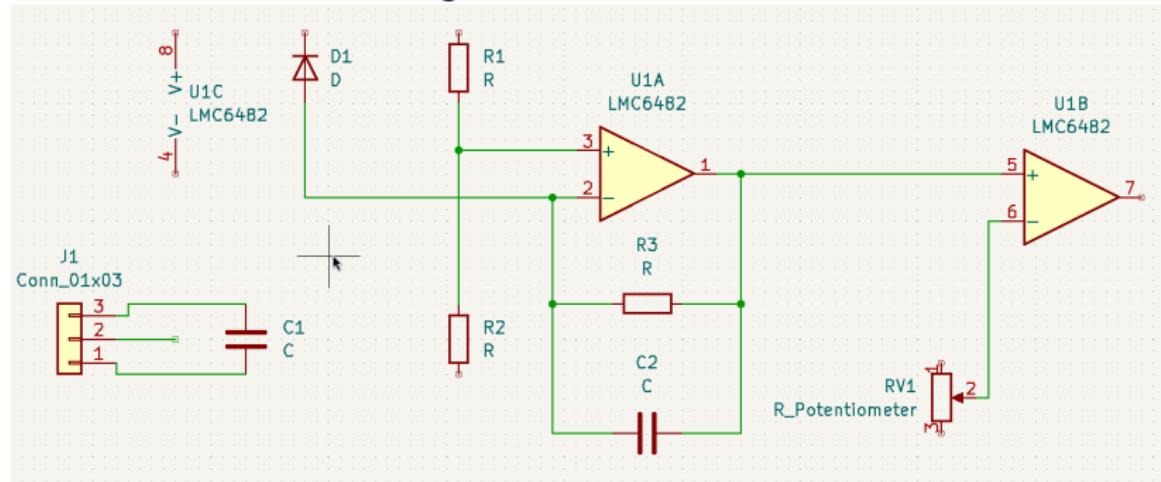


Adding Wires

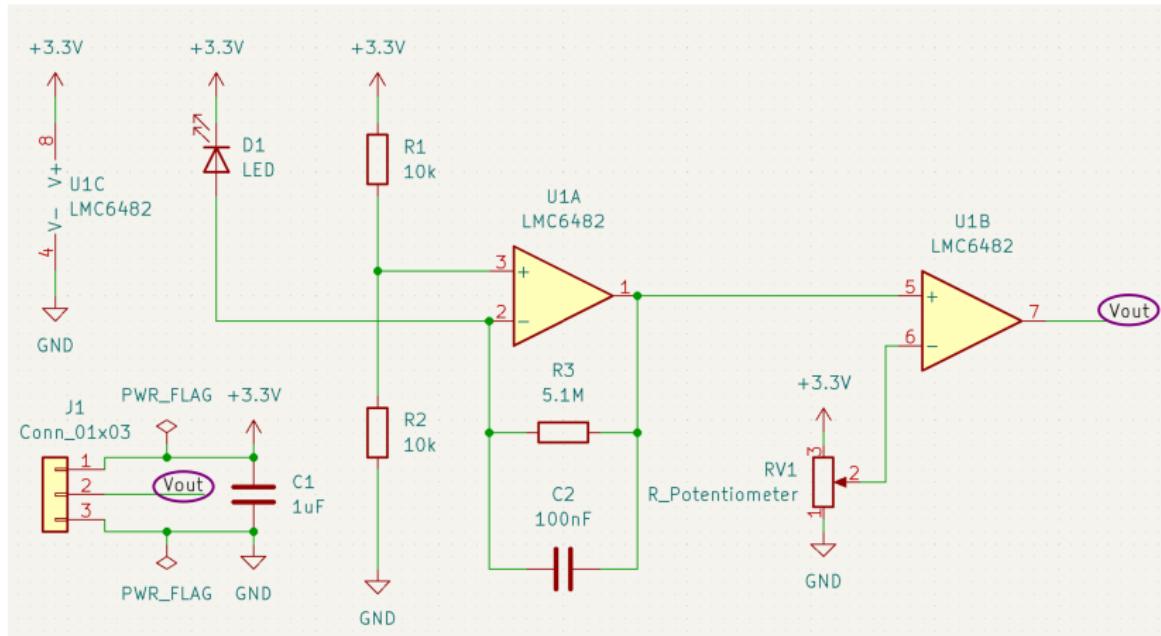
- The values for the resistors and capacitors will be blank (R, C), ignore this for now
- Wire up your components! Press **w** for **wire** to activate the wire tool and press **Esc** to go back to the selection tool.
- Repeat until the schematic is fully captured.
- Drag placed wires by selecting or hovering over them and pressing **g** for **grab**.
- Delete segments by selecting or hovering over and pressing **Backspace** or **Del**, or right-click the wire for more options.
- To create a wire that does not connect to a component on one end (floating wire), double-click where you want the wire to end.

Placing Wires

It should look something like this:



Adding Labels



Adding Labels

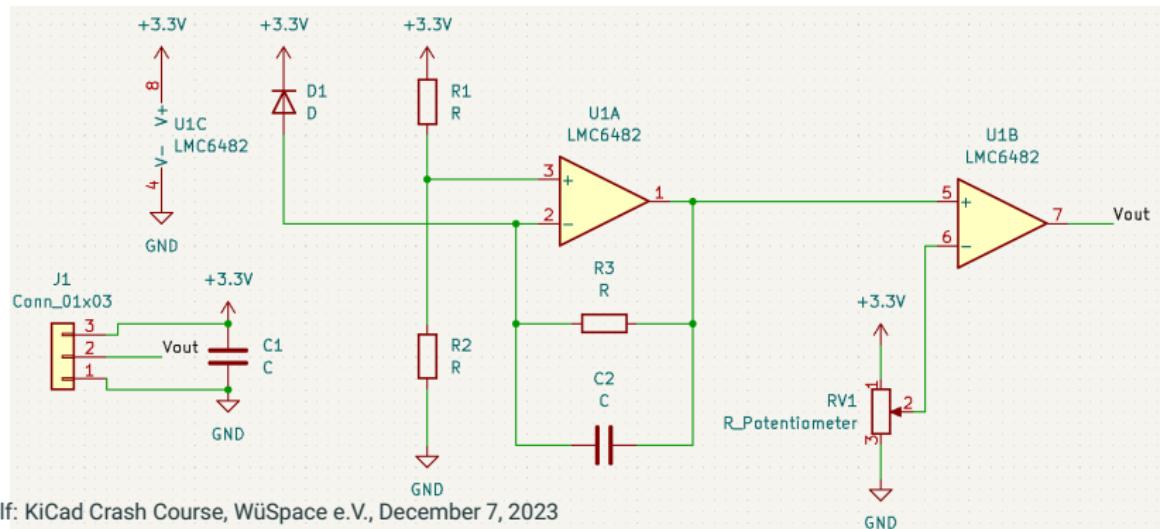
To add labels (the 'Vout' label shown above), press *L* and type in the name of your label. Labels connect two or more nodes together without actually drawing the wire on screen. They are basically magic wire tunnels linked by name.



Adding Power Ports

Now add power symbols to your schematic!

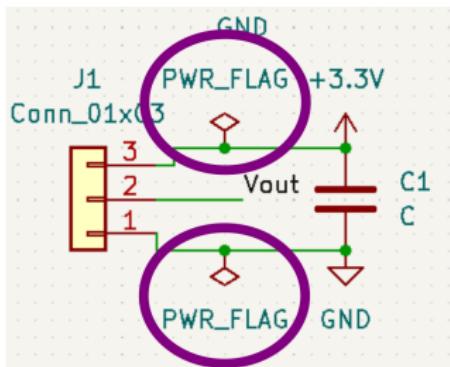
- We need GND and +3.3V
- It may be easier to duplicate components, instead of adding them separately. Try out 'Ctrl+C' and 'Ctrl+V'! ('Cmd+C' and 'Cmd+V' for Mac)



Adding Power Flags

Next, add power flags to the schematic. Again, we can duplicate the first flag.

Power flags serve to tell the Electrical Rules Checker (ERC) in subsequent steps that the pin or wire is connected to power even if a power source isn't specified.





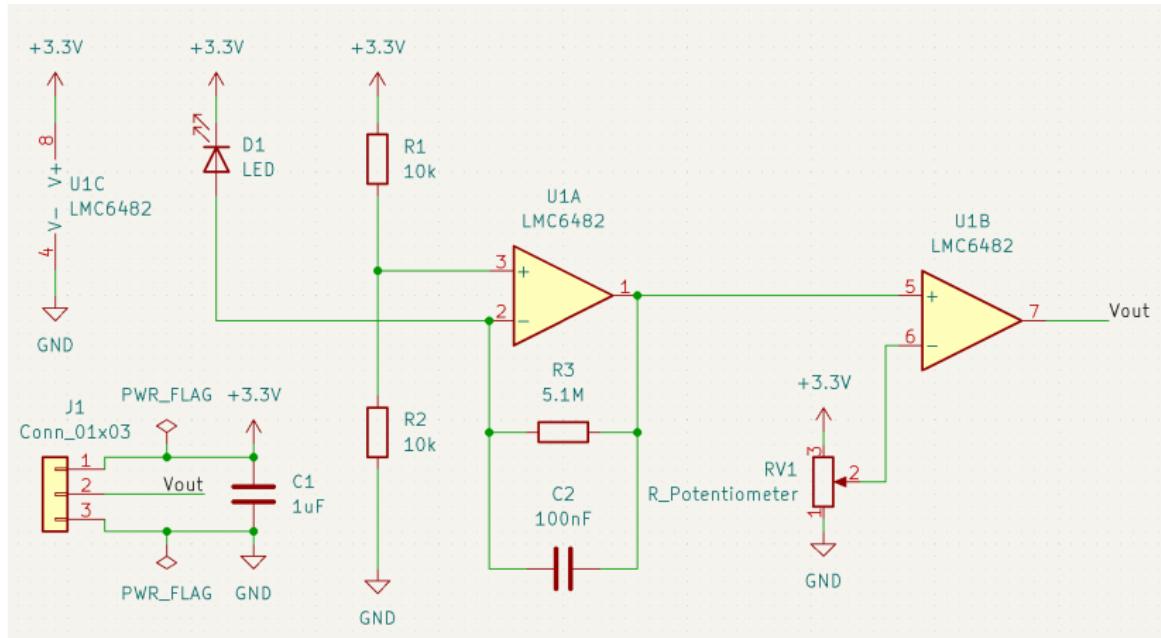
Assign Value

Assign values to components. The easiest way to do this is to select or hover over the component and press V. You can also double-click the component to open its properties or right-click and open 'Properties'.

Type the appropriate value. Omit units for resistors but include units for capacitors and inductors (F for farads, H for Henries, etc.). For example, a $1\text{k}\Omega$ resistor value would simply be 1k and a 1nF capacitor value would be 1nF.

Our Goal

Any problems?





Hell Yeah!

WHAT ARE THEY DOING?



THEY ARE BEGINNING TO BELIEVE



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

4. Schematics

- Schematic Flow

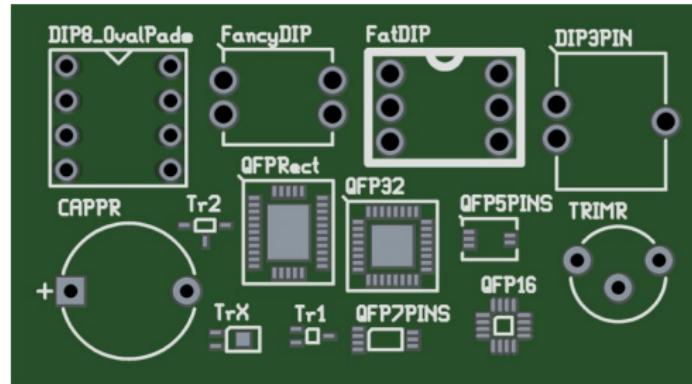
- Schematic Conventions & Etiquette

- Task 1: Designing a Light Sensor

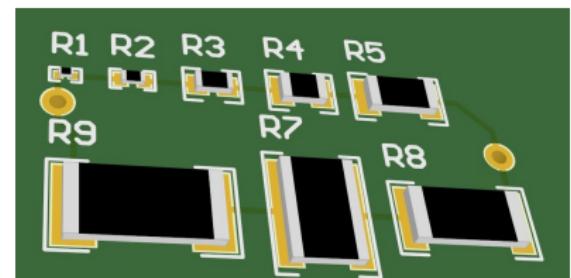
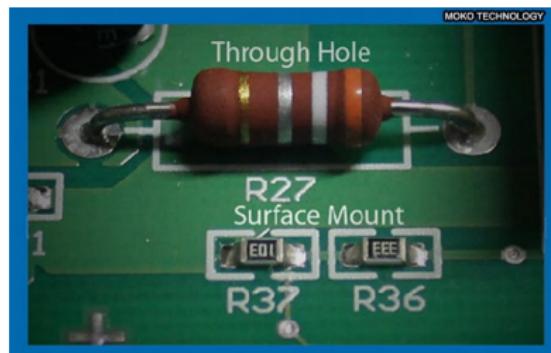
- From Schematic to Physical Components

Assigning Footprints

We have described our circuit in a way the computer and humans (hopefully) understand. Congratulations! Now we need to tell KiCad how these abstract components look in the real world. Or at least, how they will look on the PCB. This representation is called a footprint.



What footprint to choose?



SMD Footprint Codes

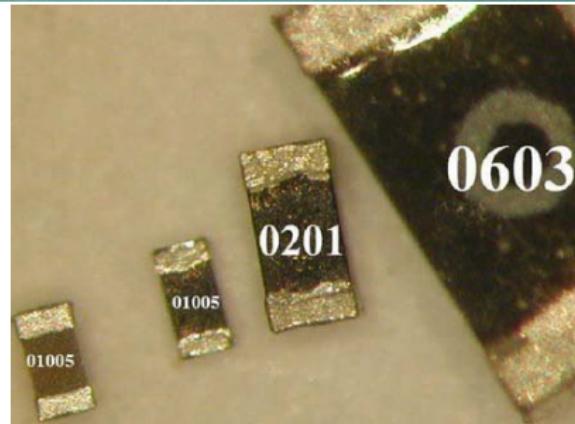
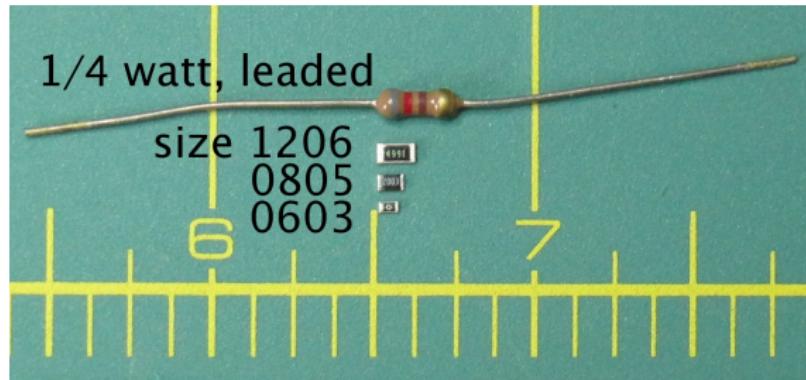
EIA footprint codes:

Imperial				Size
Resistor Case Code	Approx. Length (in)	Approx. Width (in)	Power (W)	
01005	0.016	0.008	0.031	-
0201	0.02	0.01	1 / 20 (0.05)	-
0402	0.04	0.02	1 / 16 (0.062)	-
0603	0.06	0.03	1 / 10 (0.10)	-
0805	0.08	0.05	1 / 8 (0.125)	■
1206	0.125	0.06	1 / 4 (0.25)	■■
1210	0.125	0.10	1 / 2 (0.5)	■■■
1812	0.18	0.125	3 / 4 (0.75)	■■■■
2010	0.20	0.10	3 / 4 (0.75)	■■■■
2512	0.25	0.125	1	■■■■■



Don't order 0402 components for your first project. You won't be happy.
1206 or 0805 are beginner-friendly, but you still need a good pair of tweezers.

Size Matters



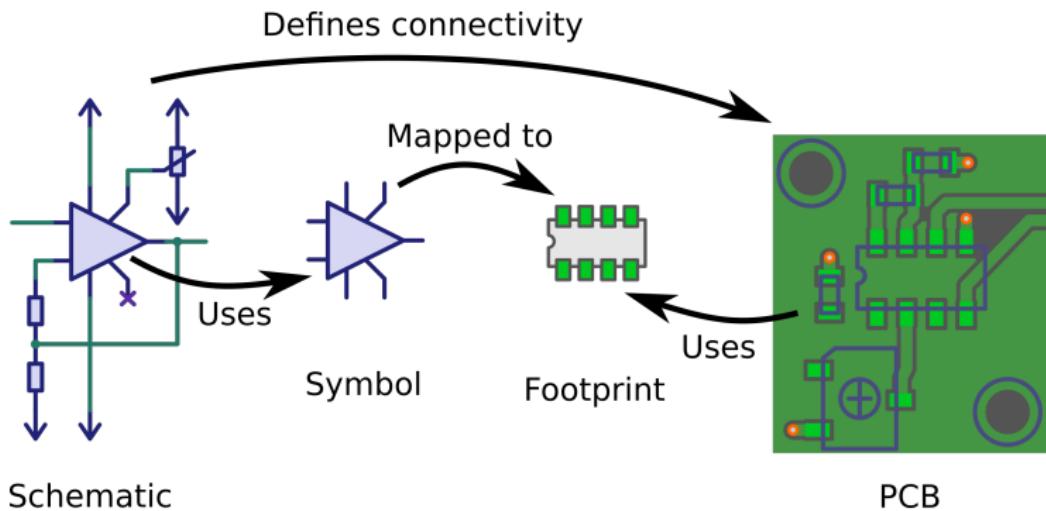


Imperial vs Metric footprints

Imperial				Size	Metric			
Resistor Case Code	Approx. Length (in)	Approx. Width (in)	Power (W)		Resistor Case Code	Approx. Length (mm)	Approx. Width (mm)	Power (W)
01005	0.016	0.008	0.031	-	0402	0.4	0.2	0.031
0201	0.02	0.01	1 / 20 (0.05)	-	0603	0.6	0.3	1 / 20 (0.05)
0402	0.04	0.02	1 / 16 (0.062)	-	1005	1.0	0.5	1 / 16 (0.062)
0603	0.06	0.03	1 / 10 (0.10)	-	1608	1.6	0.8	1 / 10 (0.10)
0805	0.08	0.05	1 / 8 (0.125)	■	2012	2.0	1.25	1 / 8 (0.125)
1206	0.125	0.06	1 / 4 (0.25)	■	3216	3.2	1.6	1 / 4 (0.25)
1210	0.125	0.10	1 / 2 (0.5)	■	3225	3.2	2.5	1 / 2 (0.5)
1812	0.18	0.125	3 / 4 (0.75)	■	4532	4.5	3.2	3 / 4 (0.75)
2010	0.20	0.10	3 / 4 (0.75)	■	5025	5.0	2.5	3 / 4 (0.75)
2512	0.25	0.125	1	■	6332	6.3	3.2	1

I know it hurts, but we use **imperial** footprint codes to avoid mixups! Or even better, name both:
Resistor_SMD:R_0603_1608Metric

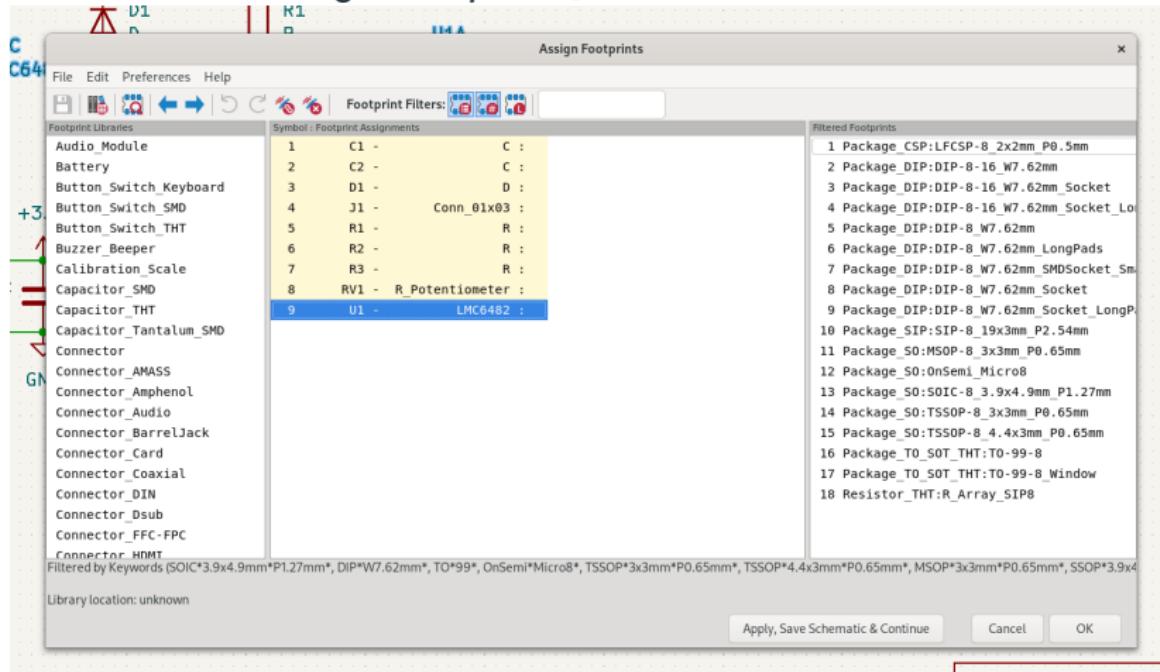
Assigning Footprints



Assigning Footprints

We will now assign footprints to all components.

Go to *Tools > Assign Footprints*, it should look like this:





Using the Assignment Tool

Activate the first two footprint filters. This will filter out footprints:

- That are in the same *category* as your symbol (e.g. show only resistor footprints)
- That have the same *pin count* as your symbol (e.g. show only footprints with 2 pins)



Using the Assignment Tool

Activate the first two footprint filters. This will filter out footprints:

- That are in the same *category* as your symbol (e.g. show only resistor footprints)
- That have the same *pin count* as your symbol (e.g. show only footprints with 2 pins)

These are some suggested footprints, but feel free to choose your own!

1	C1 -	1u : Capacitor_SMD:C_0805_2012Metric
2	C2 -	100n : Capacitor_SMD:C_0805_2012Metric
3	D1 -	LED : LED_THT:LED_D5.0mm
4	J1 -	Conn_01x03 : Connector_PinHeader_2.54mm:PinHeader_1x03_P2.54mm_Vertical
5	R1 -	10k : Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P5.08mm_Horizontal
6	R2 -	10k : Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P5.08mm_Horizontal
7	R3 -	5.1M : Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P5.08mm_Horizontal
8	RV1 -	R_POT : Potentiometer_THT:Potentiometer_Vishay_T7-YA_Single_Vertical
9	U1 -	LMC6482 : Package_DIP:DIP-8_W7.62mm

PCB Layout



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

4. Schematics

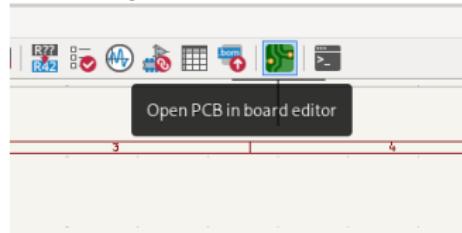
5. PCB Layout

- The PCB Editor

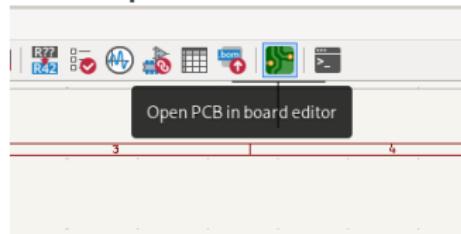
- Designing the Board

- Prepare for Manufacturing

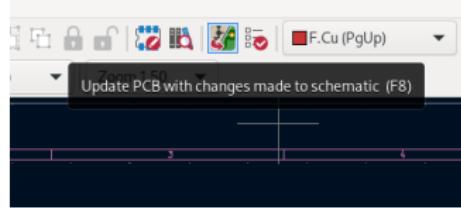
Now open the PCB editor!



Now open the PCB editor!



We want to import the changes from the schematic into our PCB editor:





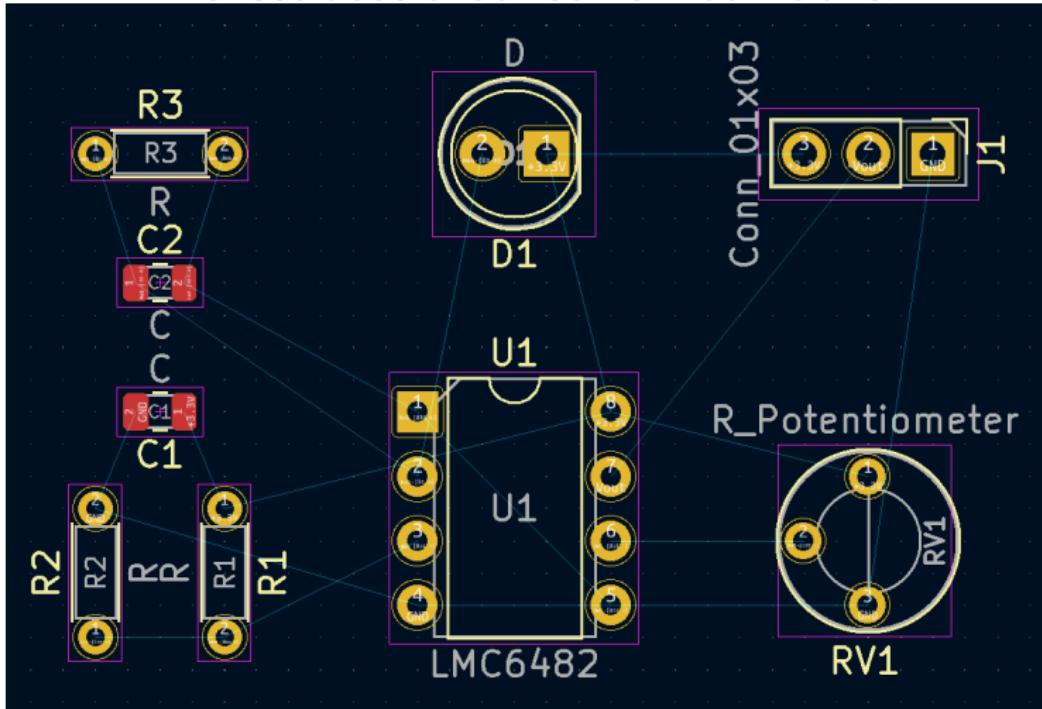
Component Distribution

Just like in the schematic we can move components with m and rotate them with r . Try to position everything in a manner that makes sense and avoid too many crossing blue lines.

*Pro-Tip: The OpAmp is the most complex component.
Put it in the middle.*

Component Distribution

The result could look somewhat like this:





Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

4. Schematics

5. PCB Layout

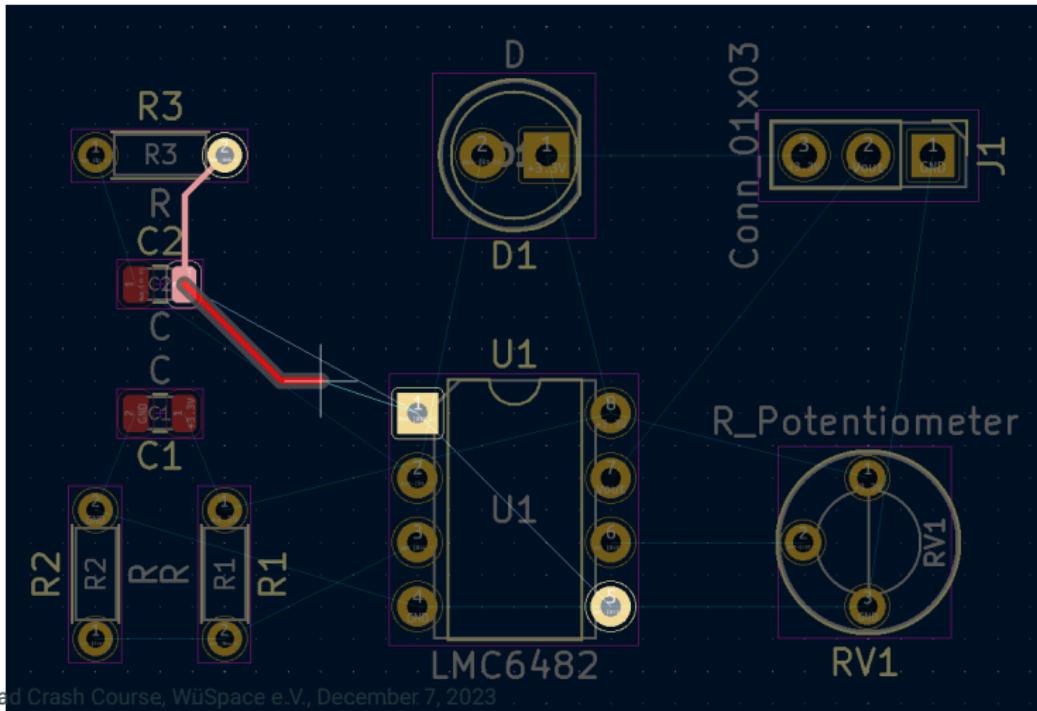
- The PCB Editor

- Designing the Board

- Prepare for Manufacturing

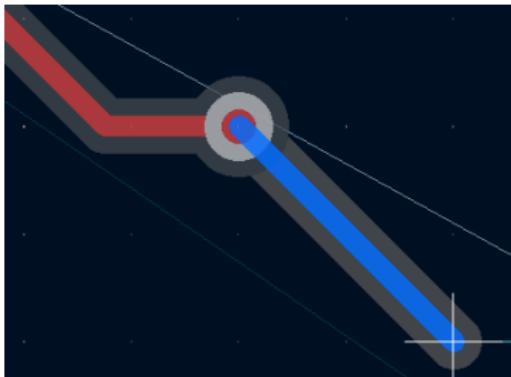
Routing

Now starts the routing. Press x over any pad to start drawing a trace. The corresponding pads should be highlighted:



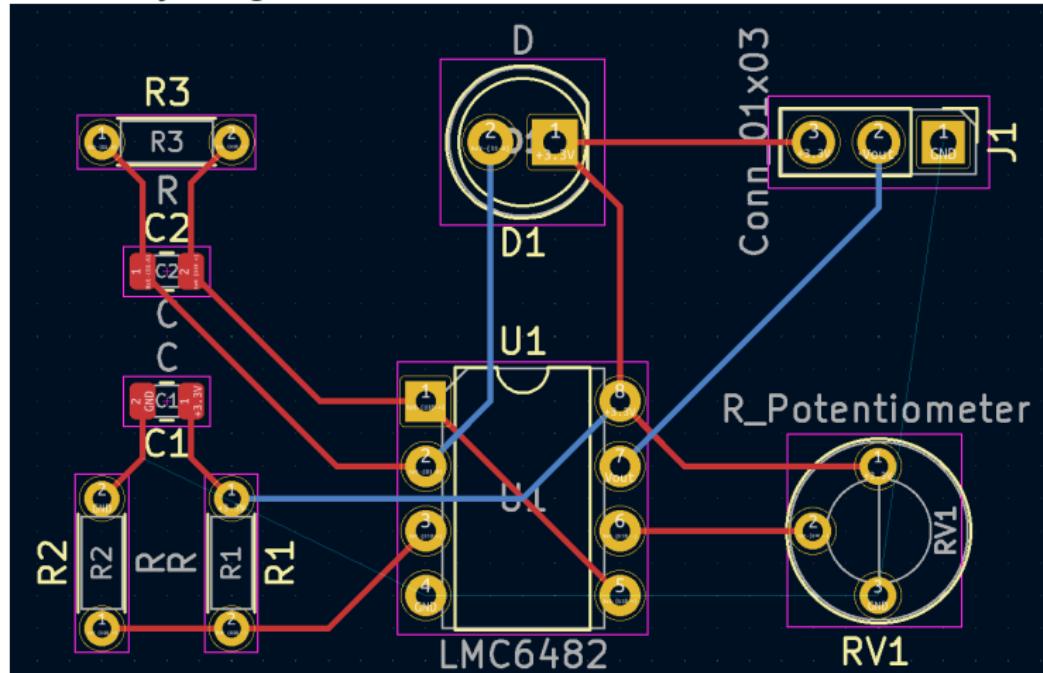
Routing

- You can delete tracks with *Del* (*Entf*)
- Don't worry about the GND connections. We'll do them last!
- You may realize that one layer is not enough. You can switch between layers with *Page Up* and *Page Down* (*Bild ↑/Bild ↓*)
- You can also jump between layers with *v*. While routing, *v* places a via!



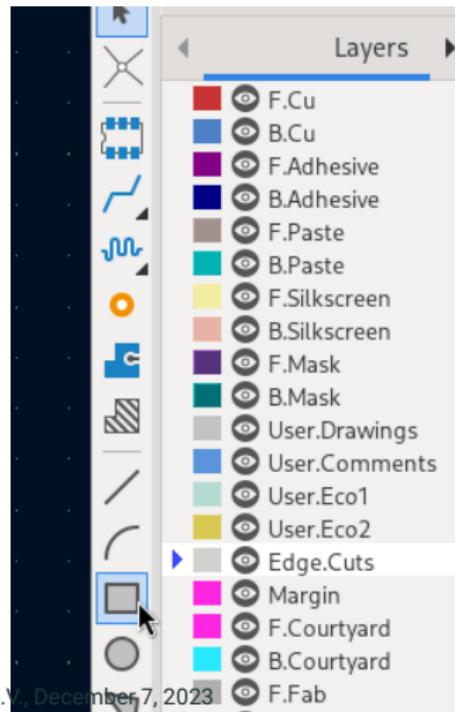
Routing

Did everything work out?



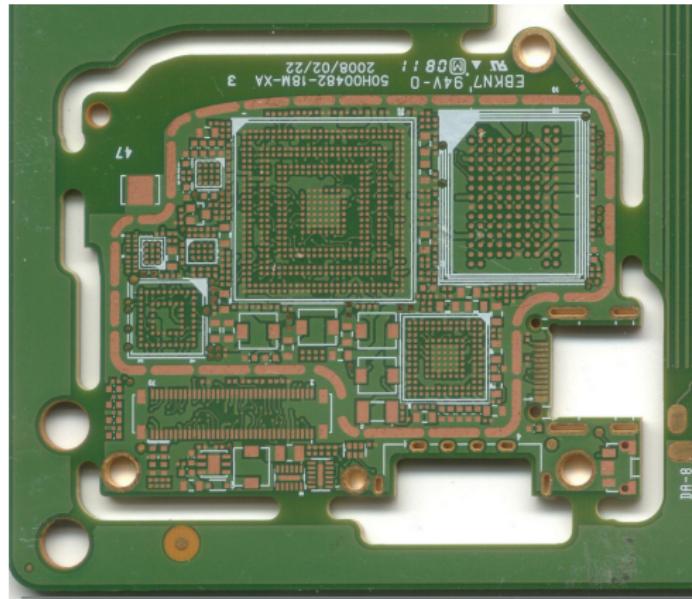
Board Outline

We need to define the board outline. For this, switch to the layer Edge.Cuts and use the rectangle or line tool:



Board Outline

Outlines can be really complicated, but a rectangle is enough for now.



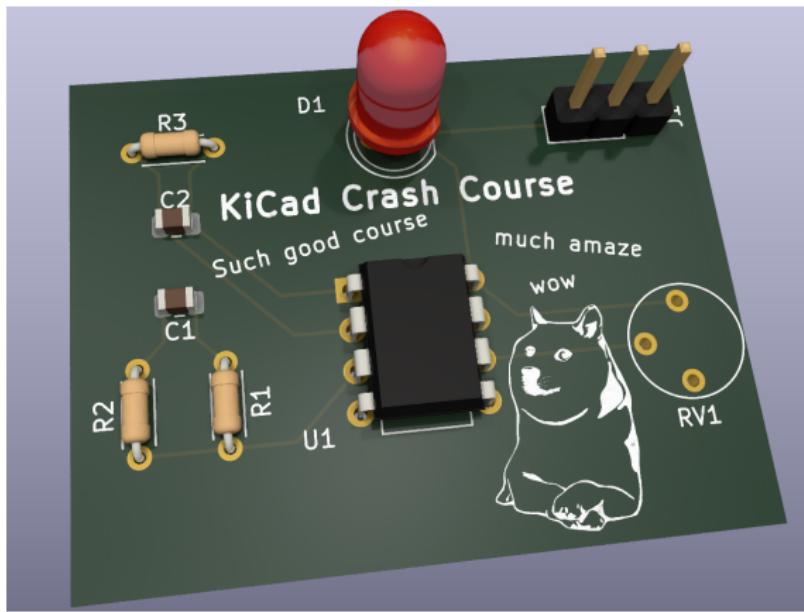
Silkscreen Comments

By switching to the layer F.Silkscreen and using the text tool we can add helpful comments/memes to our PCB.



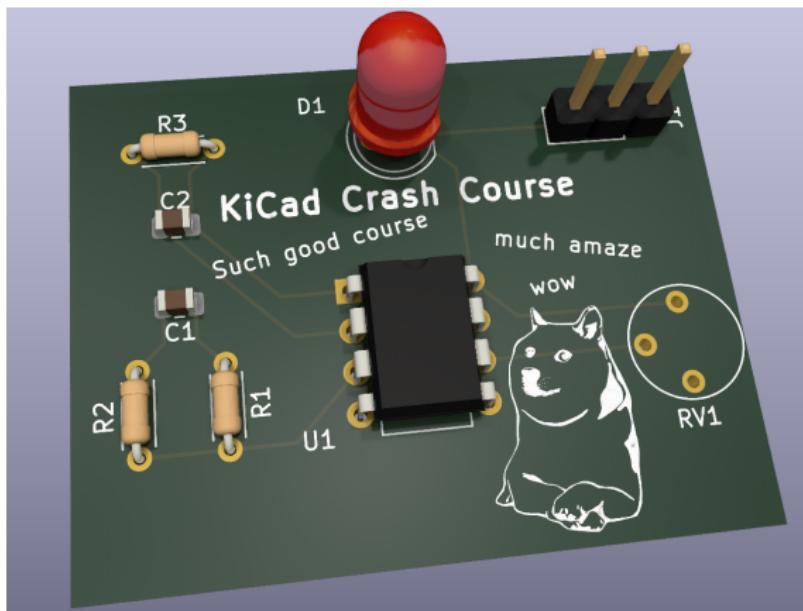
Preview Time

By pressing *Alt + 3* we can view a 3D render of our board!



Preview Time

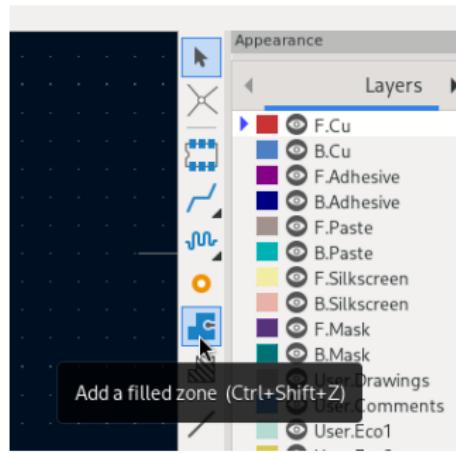
By pressing *Alt + 3* we can view a 3D render of our board!



I'm missing the model of RV1, the potentiometer :(

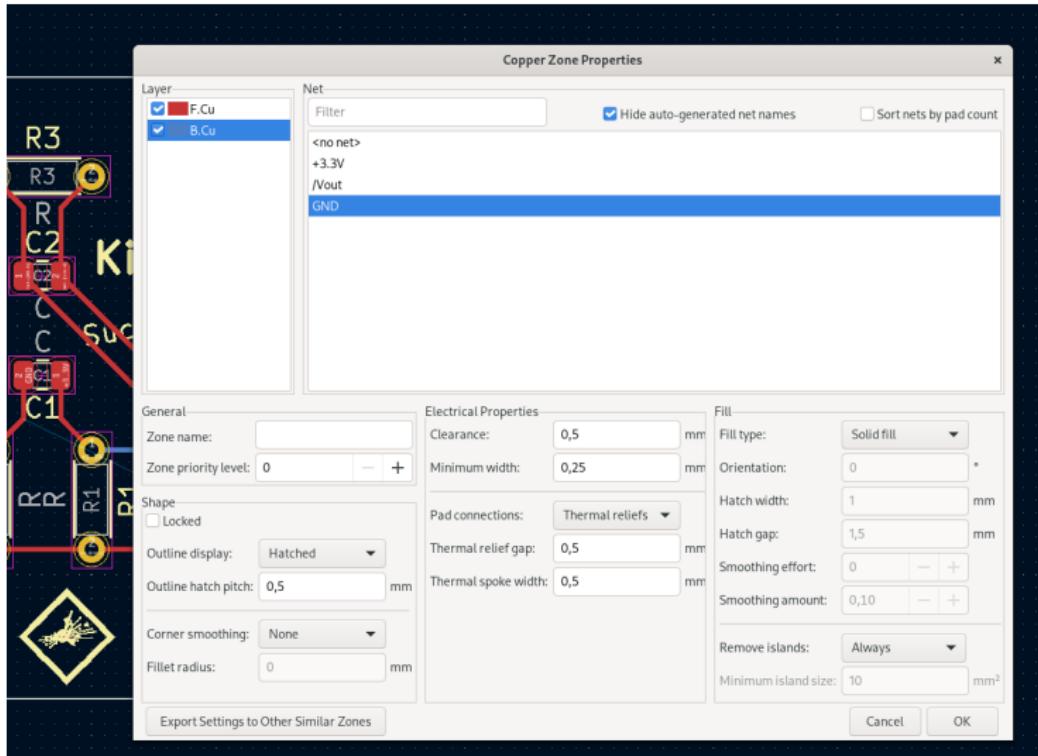
Ground Pour

Next we want to fill the remaining space on our board with a copper pour. Press this button to add a filled zone:



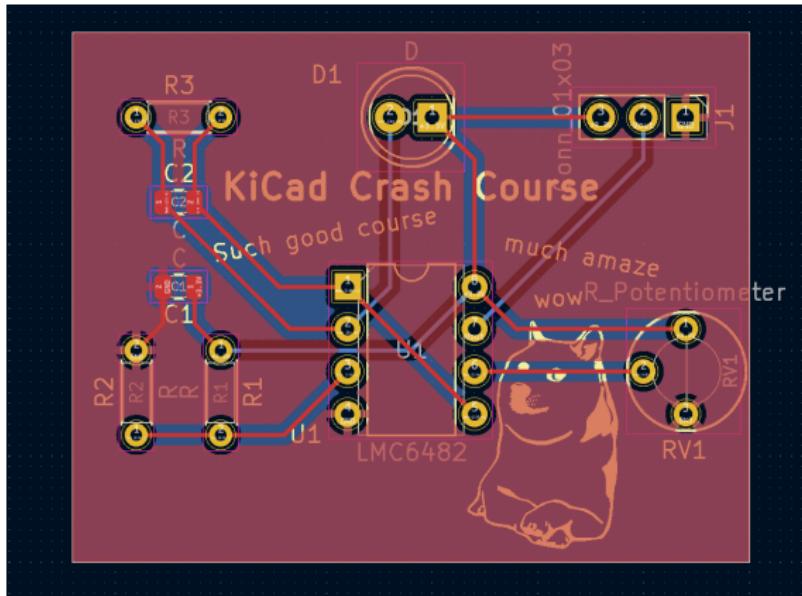
- Select both front and back layer
- Select the GND net

Ground Pour



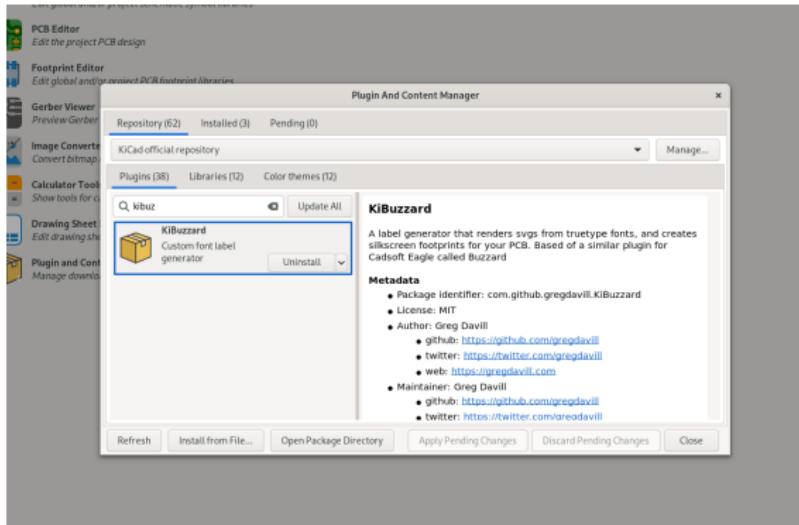
Ground Pour

- Draw a rectangle along the edge cut and double click to end.
- Press *b* to fill the copper pour.



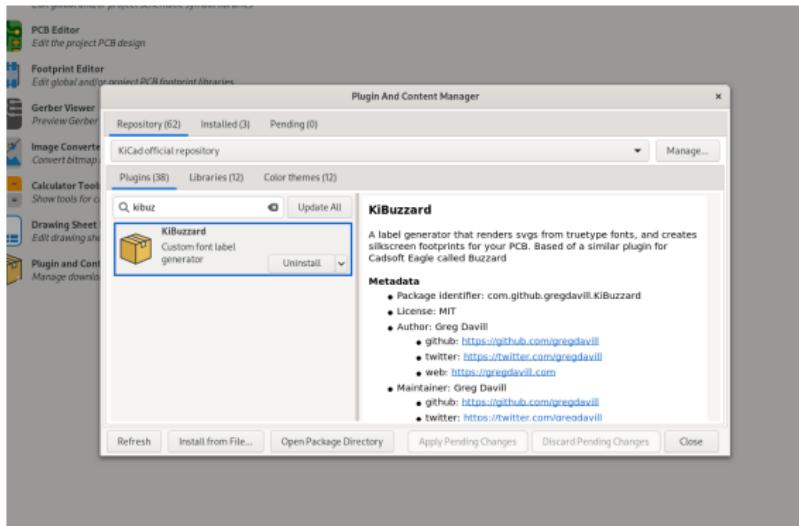
Label your Pins!

We can use an amazing plugin to generate beautiful pin labels:

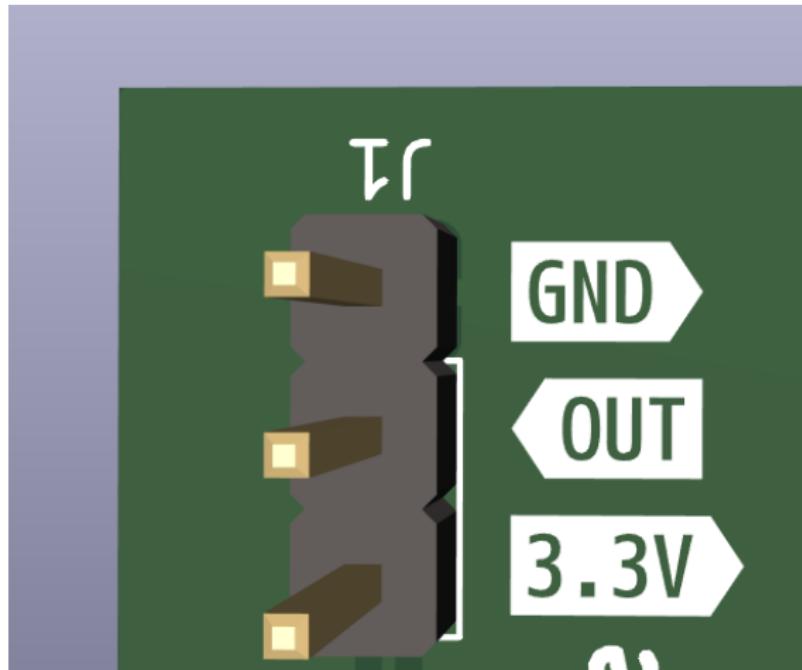


Label your Pins!

We can use an amazing plugin to generate beautiful pin labels:



Label your Pins!





Ground Pour

A ground pour has multiple functions:

- Almost everything needs a ground connection, so a complete pour is useful
- A hot component can dissipate heat with such a big copper area
- Every flowing current also has a return current. With a ground plane, the return path can be next to the trace which reduces electromagnetic emissions



Contents

1. Intro

2. What is a PCB?

3. PCB Design Tools

4. Schematics

5. PCB Layout

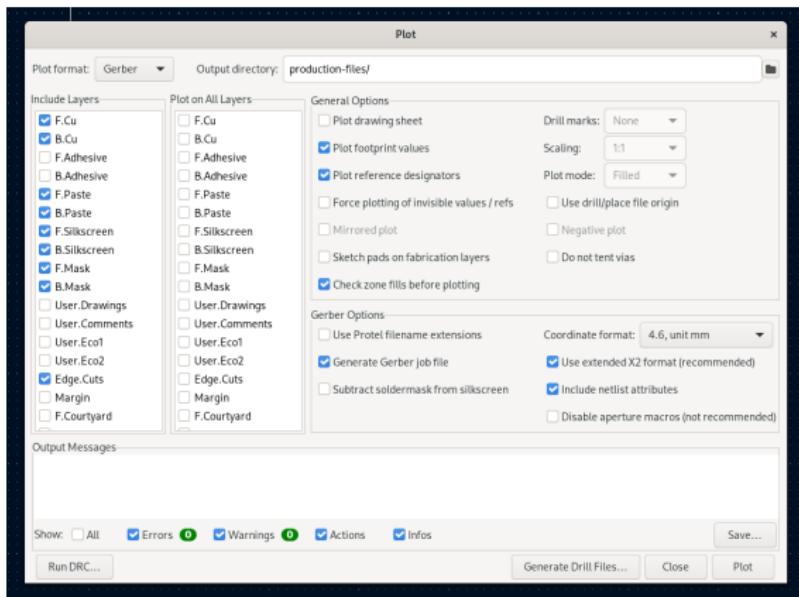
- The PCB Editor

- Designing the Board

- Prepare for Manufacturing

Gerber Export

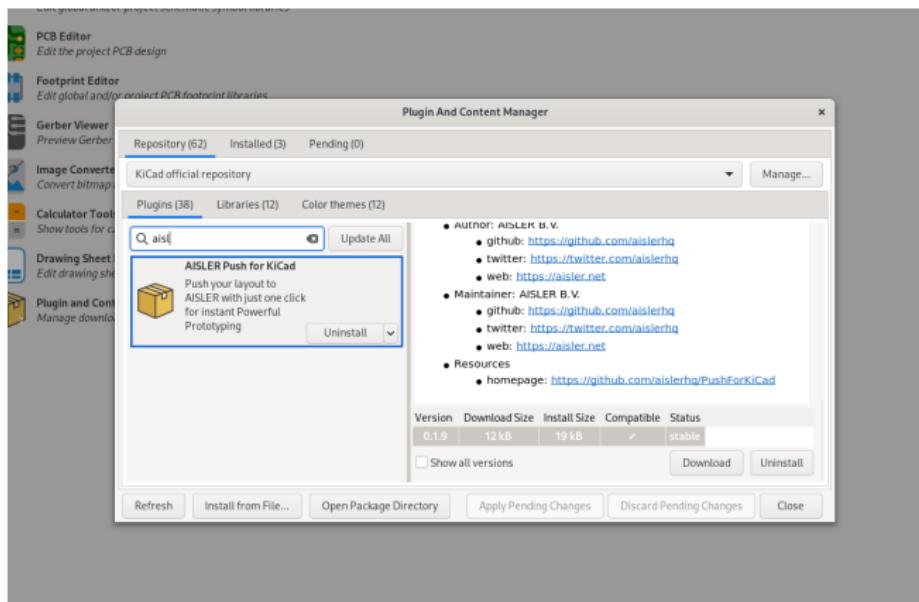
File > Fabrication Outputs > Gerbers



Don't forget the drill files!

AISLER Export

Or alternatively use one of the available direct exports like the AISLER push plugin



Further Topics



Further Topics

Whoop! We created our first PCB, you can be proud :)

There are a lot of further topics which are interesting, these include:

- Library management
- Drawing your own symbols and footprints
- BOM (bill of material) management
- Finding components on Mouser/Digikey
- KiCad and Git integration
- Controlled-impedance routing/length matching
- How to interact with manufacturers

Hi Sir,

Well got your order with many thanks~

Soooo, maybe Crash Course Part 2? I make no promises though

Course Evaluation

This was the first time I did such a course, so please help me improve it!

There are feedback forms, and I'd be really grateful to hear your opinions and constructive criticism to make this better :)

