



PCB Design

For the non-expert - Eric Hazen



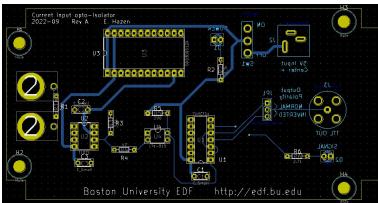
College of Engineering



Eric Hazen (aka Dr. Electron)
Senior Research Engineer

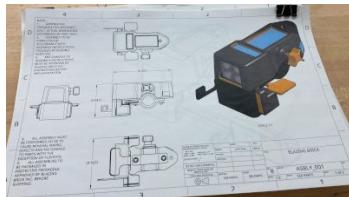
44 Cummington Mall (office, B01)
750 Commonwealth Ave (EPIC)
Boston, MA 02215

dlectron.pro 617-448-9705 hazen@bu.edu

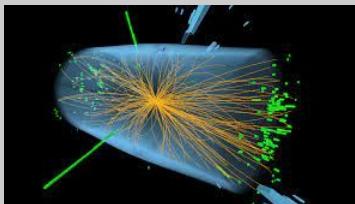


About me...

I consult on electronics for projects, help out in EPIC and design electronics for the Boas and Roblyer labs in BME
e-mail: hazen@bu.edu Appointment: <https://calendly.com/ericshazen/30min> Web: <https://drellectron.pro>



I led a team which built electronics for the CERN experiments which discovered the Higgs Boson. edf.bu.edu (now retired)



In my spare time, I build retro-electronics and fly and drive various impractical vehicles!



What is this presentation?

It is:

- A quick(-ish) introduction to PCB design using KiCAD (free tool)
- A walk-through of a typical (but contrived) design
- Suggestions on where/how to get boards built

It is NOT:

- A course in circuit theory
- A detailed training in PCB theory, esp not high-power, RF, low noise etc
- Instructions for use of any tool other than KiCAD

At the end, you should be able to design a simple PCB

Tutorial: https://docs.kicad.org/8.0/en/getting_started_in_kicad/getting_started_in_kicad.html

3

Deliverable

What you need to turn in (due Thursday 10/23):

- Complete KiCAD project with Arduino analog shield design
(Must be either a GitHub repo I can access or a zip archive of the project)
 - **Schematic must be complete to get full credit**
 - **PCB layout may be done for extra credit.**

NOTE -- Please submit your project in advance of the due date.

*If I find on the due date that I can't access or open it due to an error,
you get no credit.*

Grading is essentially “pass/fail” on this... if you have a mostly complete schematic which shows you know how to use the tools, you will get full credit.

Extra credit for completing the PCB layout

What is a PCB? And, some jargon

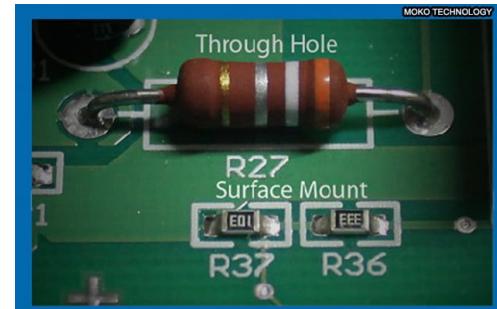
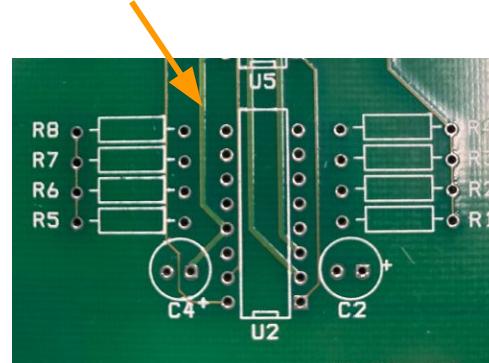
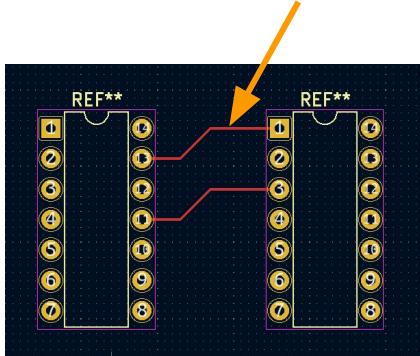
Simply, the “board” is a flat sheet of insulating material (typically “FR-4” fiberglass)

Component “footprints” are places to solder down parts

(“thru-hole” have leads or wires which go through the board,

“Surface mount” or “SMT” parts sit flat on the board)

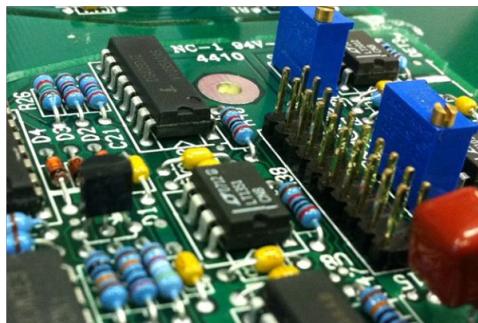
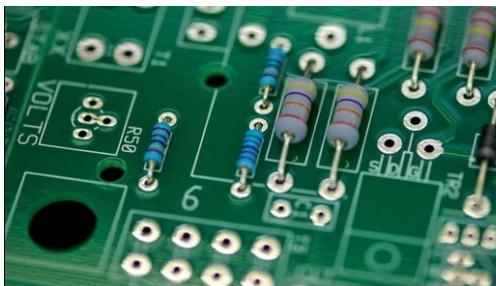
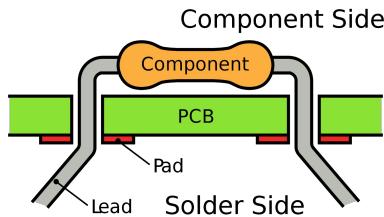
“Tracks” or “traces” are plated copper lines which connect the footprints.



Thru-hole (THT)

vs

Surface Mount (SMT)



Thru-hole is the original PCB technology. Still in widespread use. Simpler to design and solder.

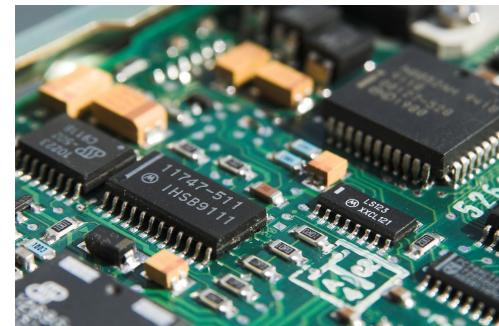
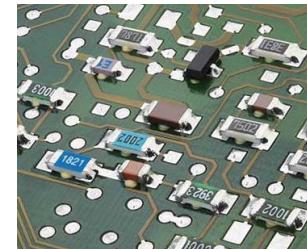
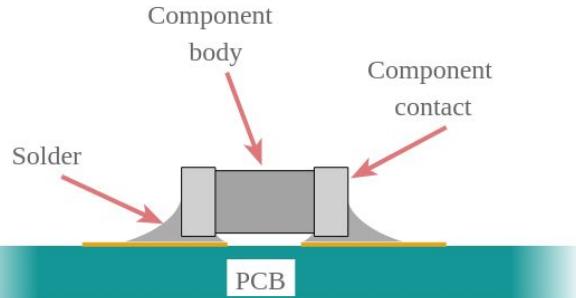
Requires more board space

Parts availability starting to become a problem

Surface-mount was introduced in the 1980s and is used almost exclusively for large volume manufacturing.

Can be challenging to assemble by hand.

But we teach this in EPIC Soldering workshops :)



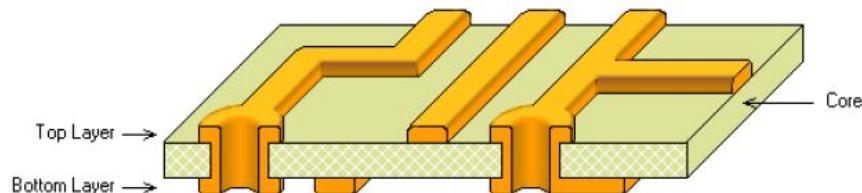
Two-sided vs multi-layer

- “Two-sided” refers to the fact that there is wiring on top and bottom only.
- “Multayer” means that there are inner layers which can also contain wiring.

Inner layers connected by using vias or thru-hole pads.

“via” = hole to make connection from one layer to another

Two-sided

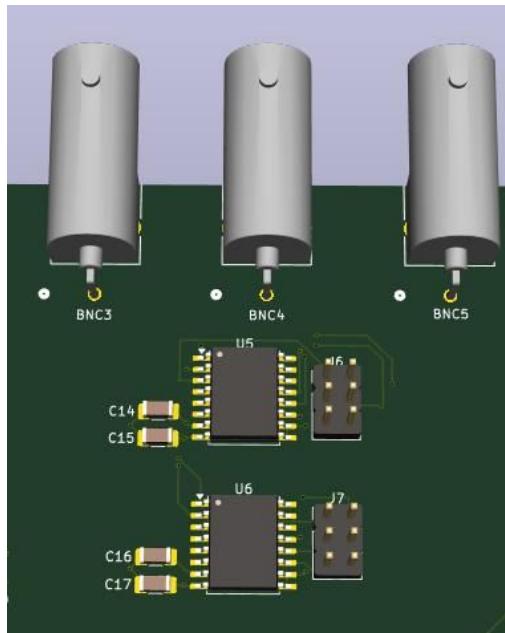
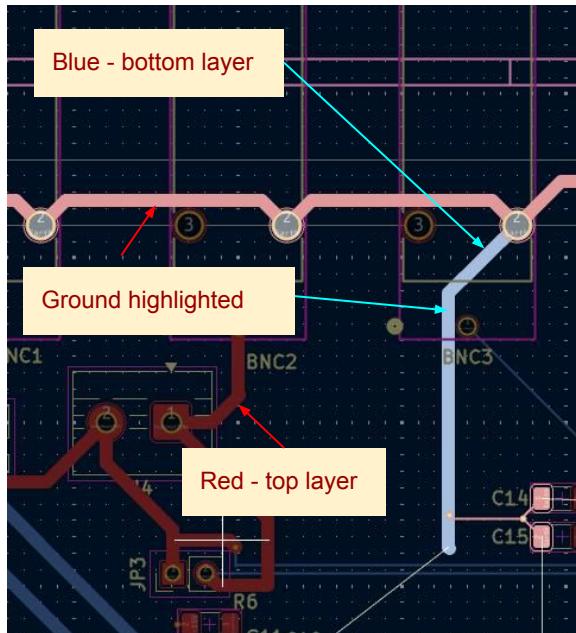


Four layer



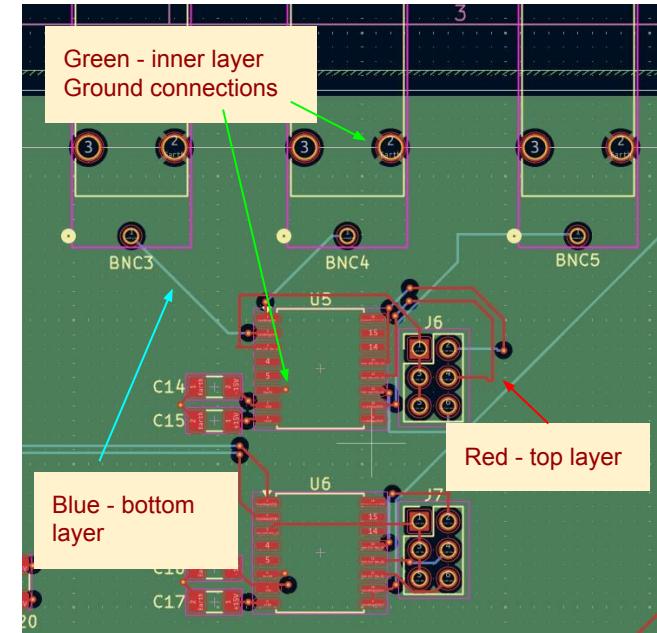
Why use multilayer?

On a two-layer board, power connections must be routed on tracks along with the signals. This can compromise the performance of the circuit



On a multi-layer board, a solid “plane” can be used to distribute power. This can make signal routing on dense boards much easier.

Also greatly improves signal integrity (lower noise)

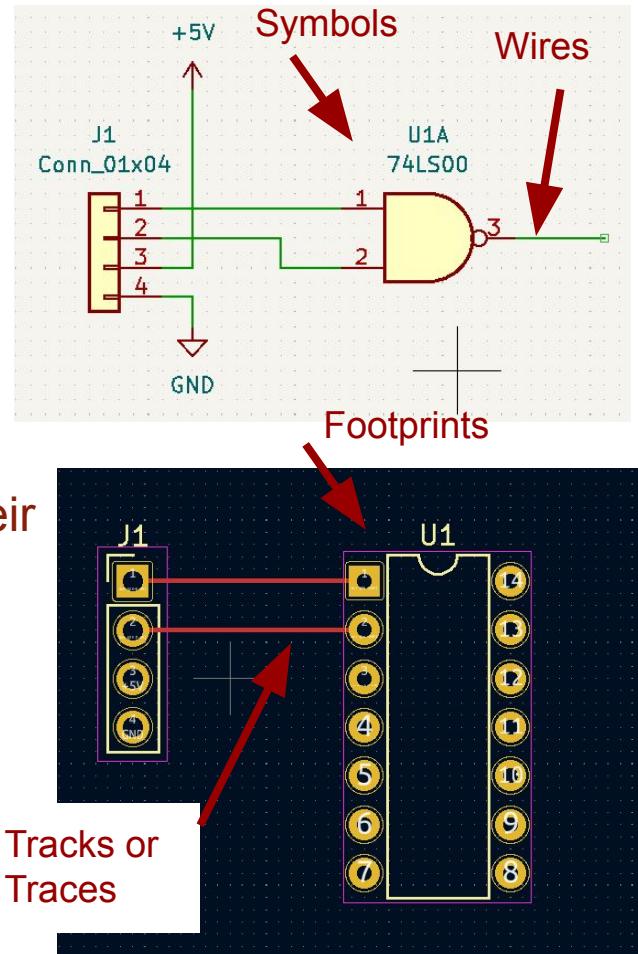


Drawings drawings drawings...

Just as in mechanical design, PCB design is accomplished using CAD tools to create drawings or “artwork”.

Schematic: human-readable drawing which shows the components using “symbols” and their interconnections.

Layout: physical drawing which is used to produce the PCB. Special data formats (e.g. “gerber files” are sent to the manufacturer)



PCB Design Process Steps

1. Write or obtain a specification
2. Sketch initial schematic
3. Select specific parts
4. (build and test prototype, not discussed here)
5. Locate or create schematic **symbols** and **footprints**
(either in supplied libraries; create using editor or download)
6. Draw (“*capture*”) the schematic in schematic editor
7. Assign footprints to symbols
8. Open PCB design tool, *place* components and draw board outline
9. *Route* connections between components
10. Export gerber files for manufacturing

Design Study - automatic electric toothbrush

Problem description:

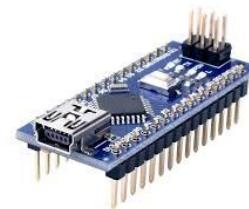
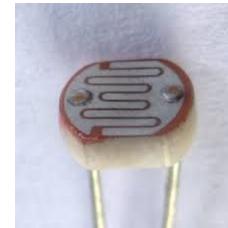
Design an electric toothbrush with these features:

- Automatically turns on when picked up from stand or flat surface
- Provides a battery charge indicator
- Provides a light which illuminates in the dark



Electric Toothbrush - components

- Drive motor with on/off control (power MOSFET transistor)
- LED light with on/off control
- Ambient light sensor
- Rechargeable battery
- Pushbutton for manual on/off
- Microcontroller



Design Steps - preparation

- Install KiCAD (currently v9.0.4). Runs on Windows, Linux, MacOS 
- Install supplied libraries and documentation (included by default) 
- Understand what files and folders (or “directories”) are on your computer (google it!). Figure out where KiCAD files are stored (“Documents?”)
- Create an empty Github repository for your design, and check it out on your computer. (don’t know git - google it!. Optional but *very useful*)
Share with me. Username eshazen on GitHub



KiCAD version note: lab computers have V8 (sorry!)

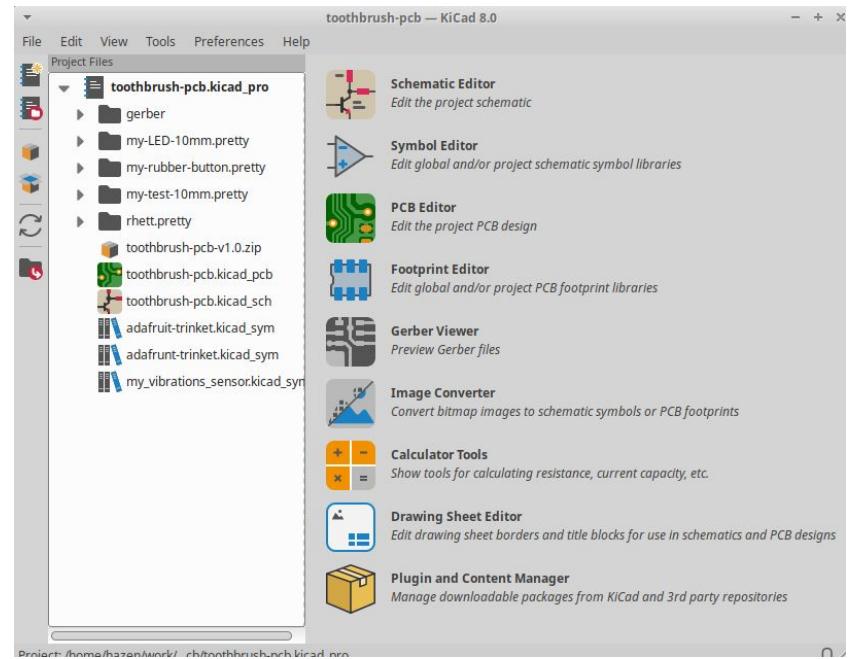
Moving from 8 to 9 is fine, but you can't go backwards

KiCAD - first time startup

- Launch KiCAD
- Choose “Start with Default settings”
- When you open a schematic first time, choose “Copy default global symbol library...”
- Also when prompted choose “Copy default global footprint library...”



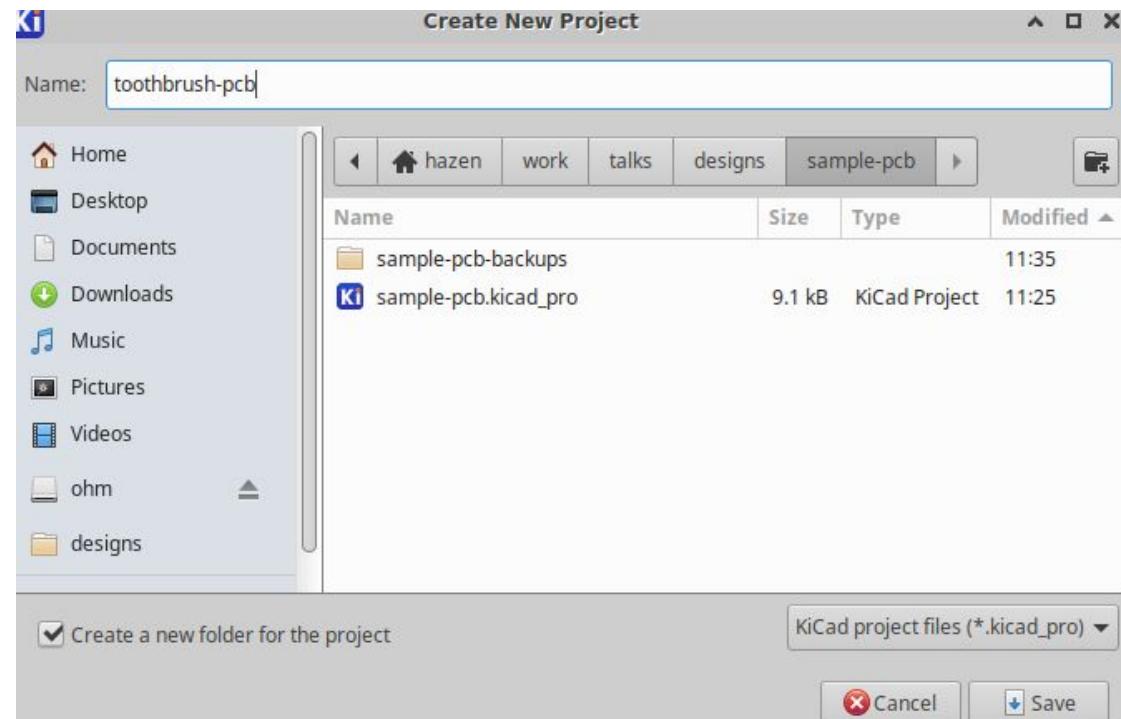
KiCAD Project Manager



Drawing the schematic - I

Start KiCAD. Create a new project using **File→New Project**.

Navigate to the Github repo directory you created or another appropriate place on your computer to store the design. Enter a simple name. We'll use “toothbrush-pcb”.



Note: for maximum compatibility, avoid spaces and odd punctuation in project and file names

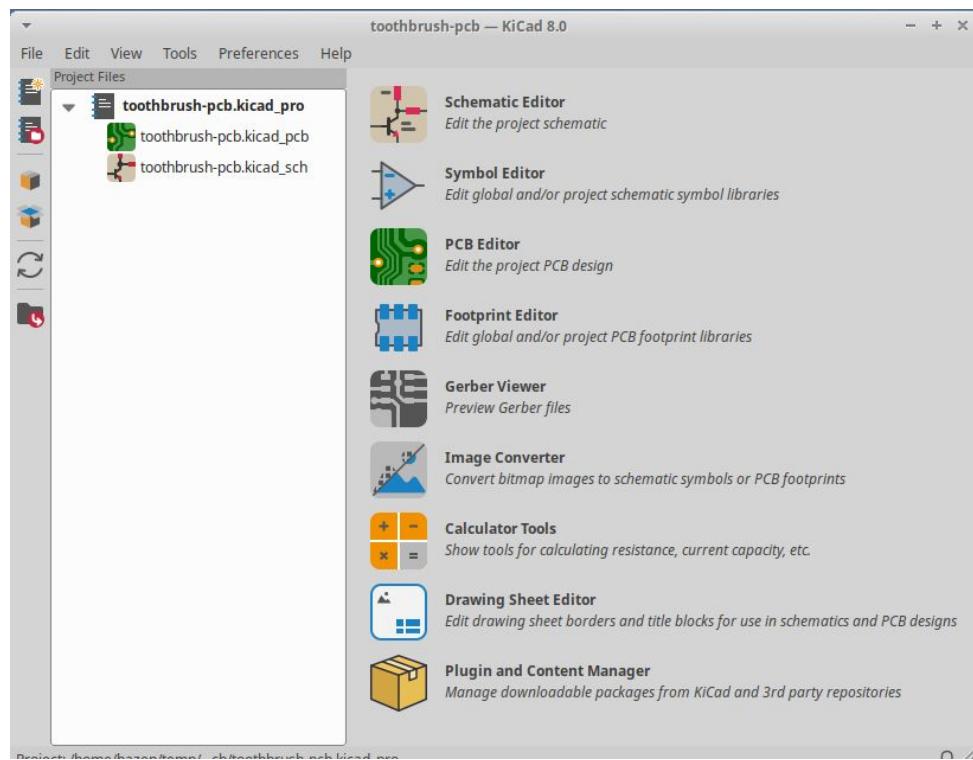
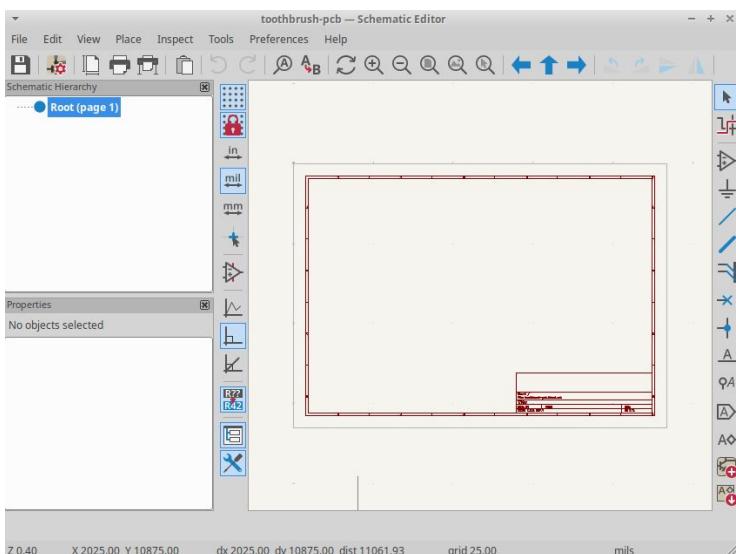
Drawing the schematic - II

This is the KiCAD project manager.



Click “Schematic Editor”
to start the schematic.

This will display a blank sheet with title block.



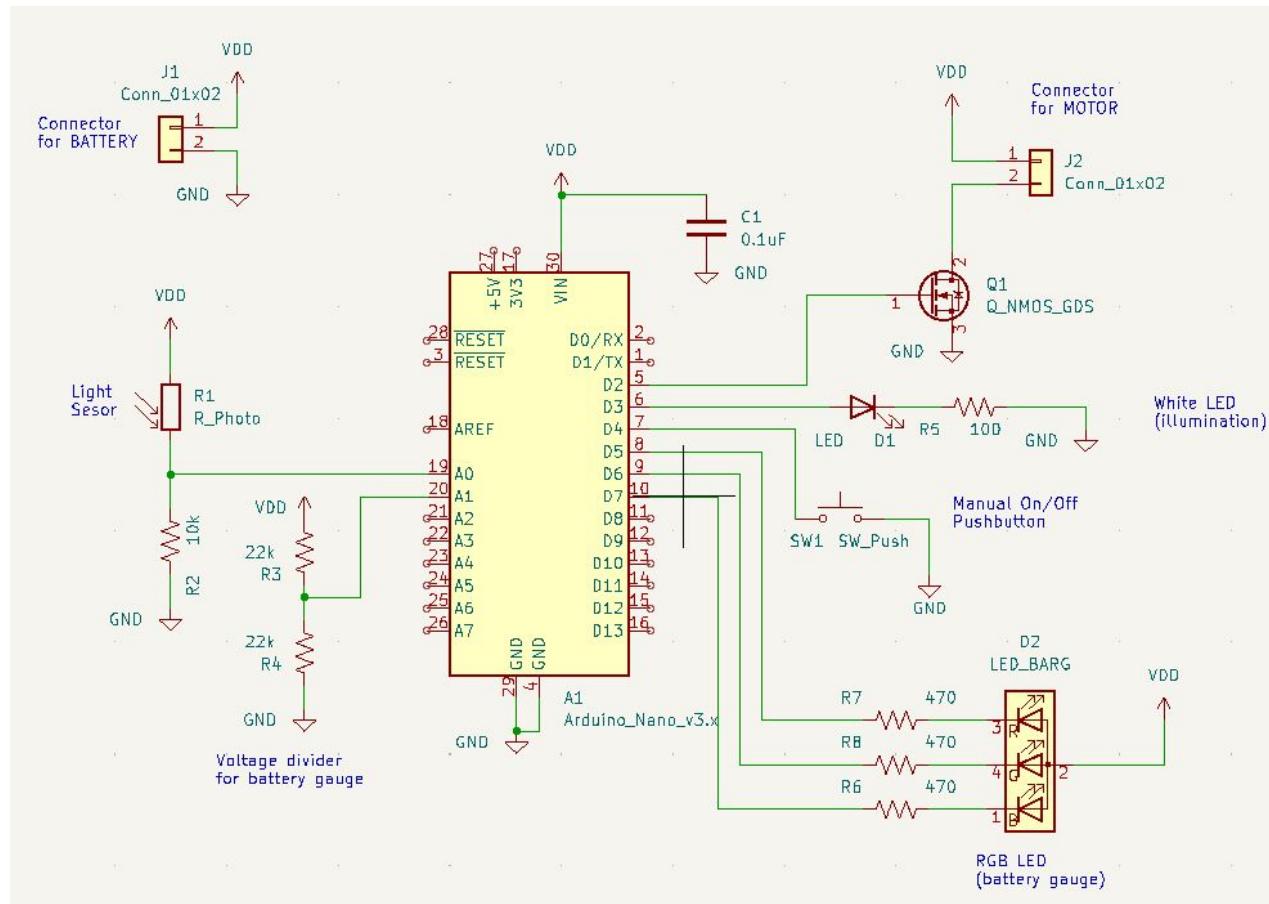
Here's what the finished schematic should resemble...

Notes:

Neat, easy to read
Minimum crossing wires
Everything labeled

All symbols were in the supplied KiCAD libraries

(you can make your own
If needed).



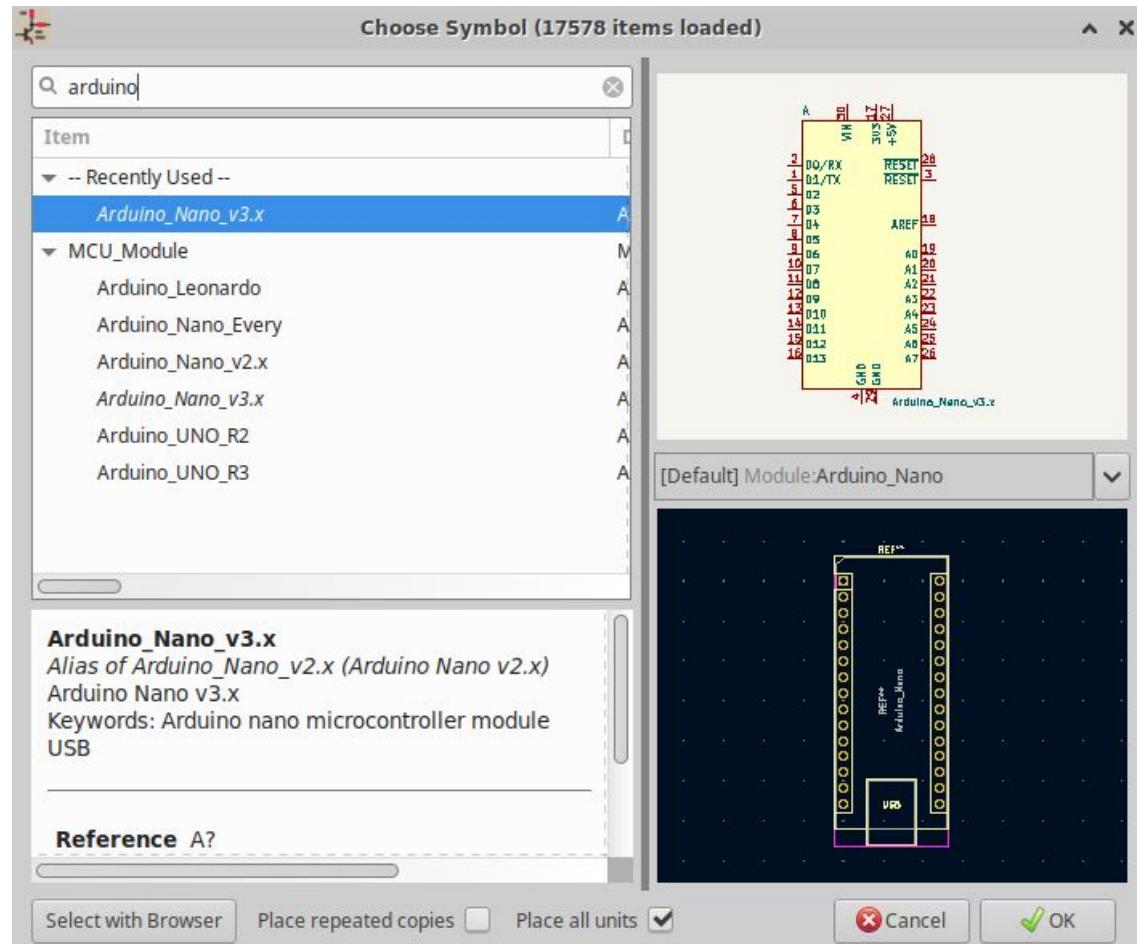
Picking symbols

Find the symbol icon on the right.  Select it and click in the page.

This brings up the browser. Start typing a part name or number, e.g. “Arduino”. Pick the part you want.

If there isn't an appropriate symbol, you'll need to make your own (details later)

Place the symbol on the sheet.



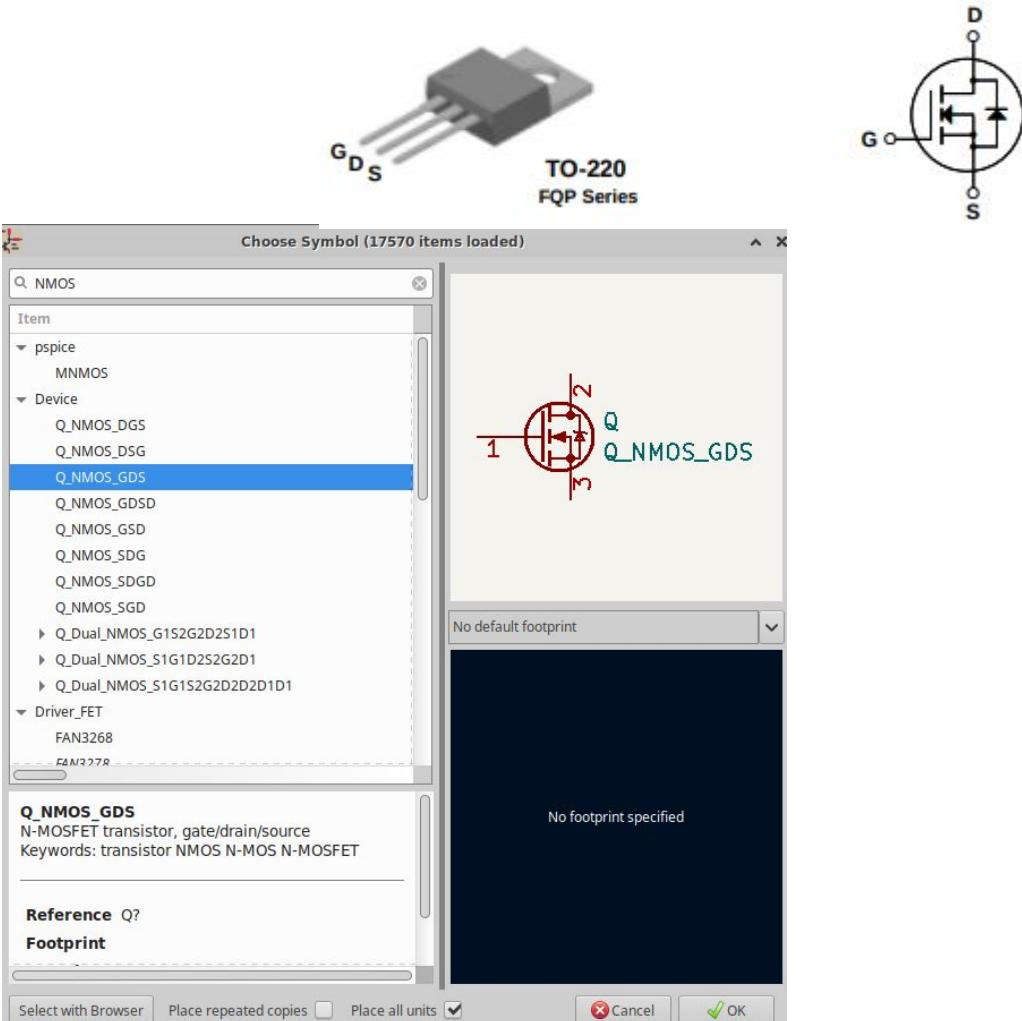
Parts Confusion?

Picking parts can be confusing.
For example, a MOSFET

Typing “MOSFET” in the browser produces many choices! “NMOS” is better.

I chose the “Q_NMOS_GDS”.
(GDS means “gate, drain, source” pin ordering on the part...)

It is not critical to get just the right symbol, just one for the proper type of device and the correct number of pins.



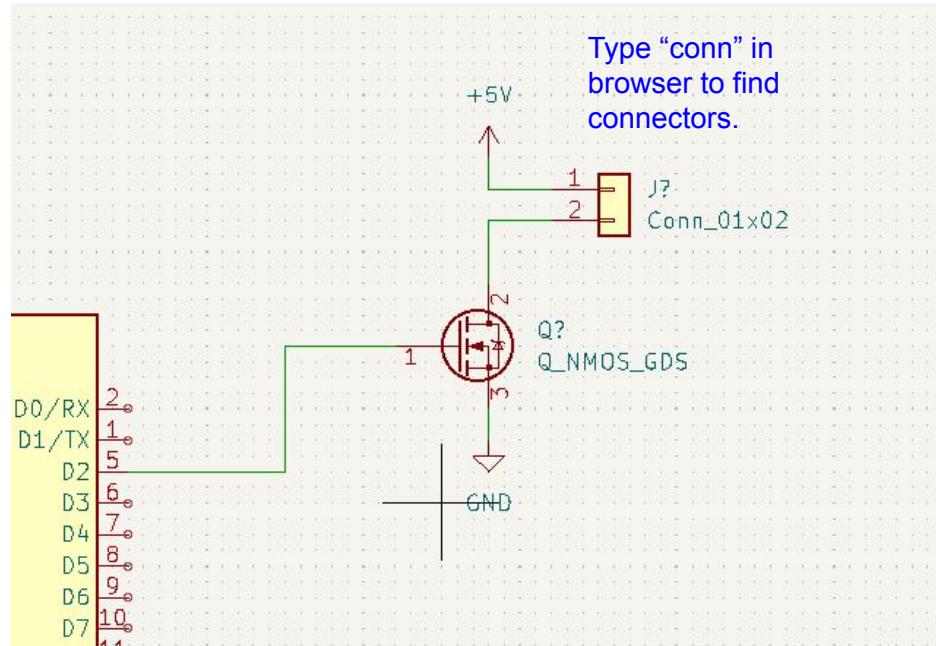
Power, Ground

Power symbols connect together
All points in the circuit with the same
Symbols. This works for GND too.

Be sure to always use the same one.
("GND" not "AGND").
("5V" not "5VA").

Add green connection wires...
Hover mouse over pin (red circles)
and click to start wiring.

Power/GND
Symbol tool



Tips and tricks -- Hot Keys

“X” and “Y” flip a component horizontally and vertically

“M” is for “move”

“G” is for “drag”

Ctrl+D is for “duplicate”

The **Delete** key does what it says

The **Home** key zooms to the whole drawing

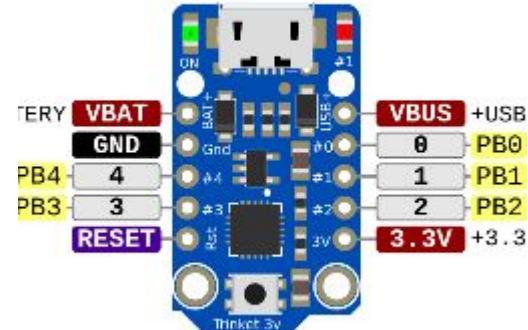
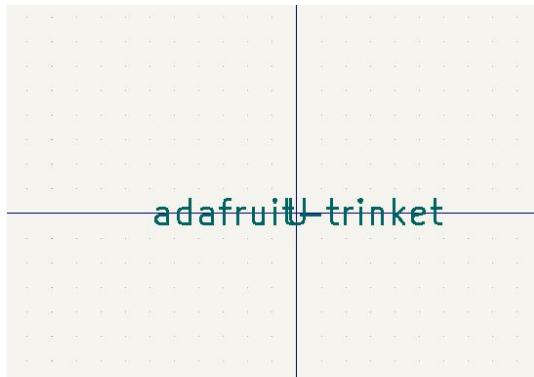
Save Often! (Ctrl+S).

Creating a new Symbol - e.g. Adafruit “trinket”

1. Open the **Symbol Editor** in the design manager
2. Select **File→New Library** (I prefer to put each custom symbol in its own library)
3. In the “Add To Library Table” dialog select “Project”
4. Type a name, e.g. “adafruit-trinket” and **Save**
5. Select **File→New Symbol**. Type a name (can be same as library name, e.g. “adafruit-trinket”). Leave all else set to default, click **OK**



Now you have a blank window with some text:



Drawing the symbol

It is helpful to either find a sample of a symbol in the part's datasheet or online, and sketch it on paper.

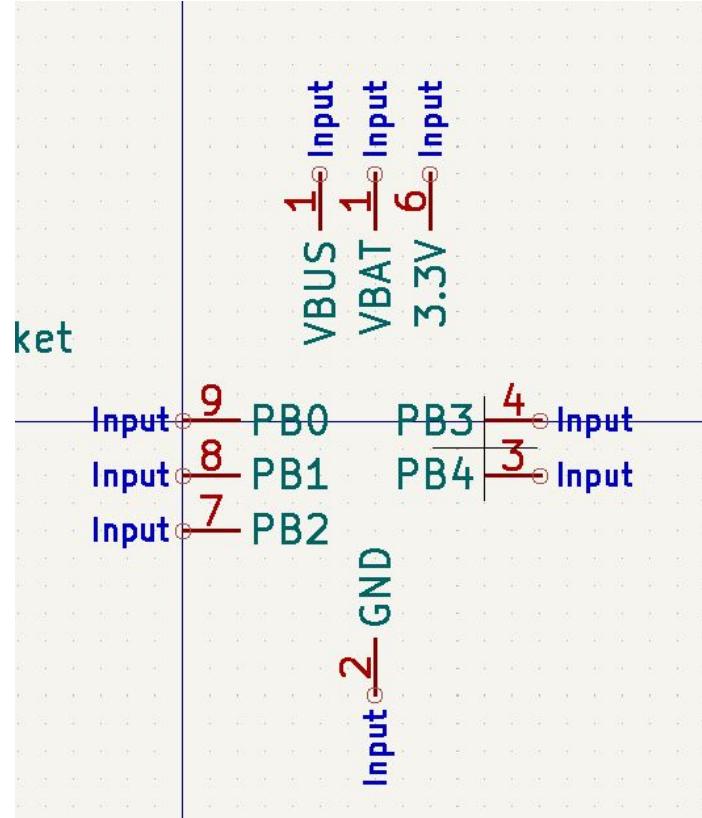
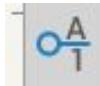
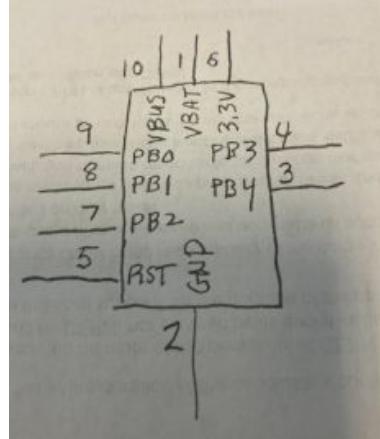
Find a similar example (e.g. another Arduino in the library)

Use the pin tool to add pins. Each pin has a name (e.g. "VBAT") and a number (e.g. 1).

Pin numbers are usually in the datasheet but sometimes you have to make them up.

Pin orientation (up/down/left/right) is obvious.

Pin type (input, output etc) should be set to your best guess. Use "passive" if in doubt.



After placing the pins, draw a rectangle. Select the rectangle, press "e" for edit, Then "Fill with background color"

Finished Symbol

Notes:

GND on bottom, power on top

Inputs on left, outputs on right

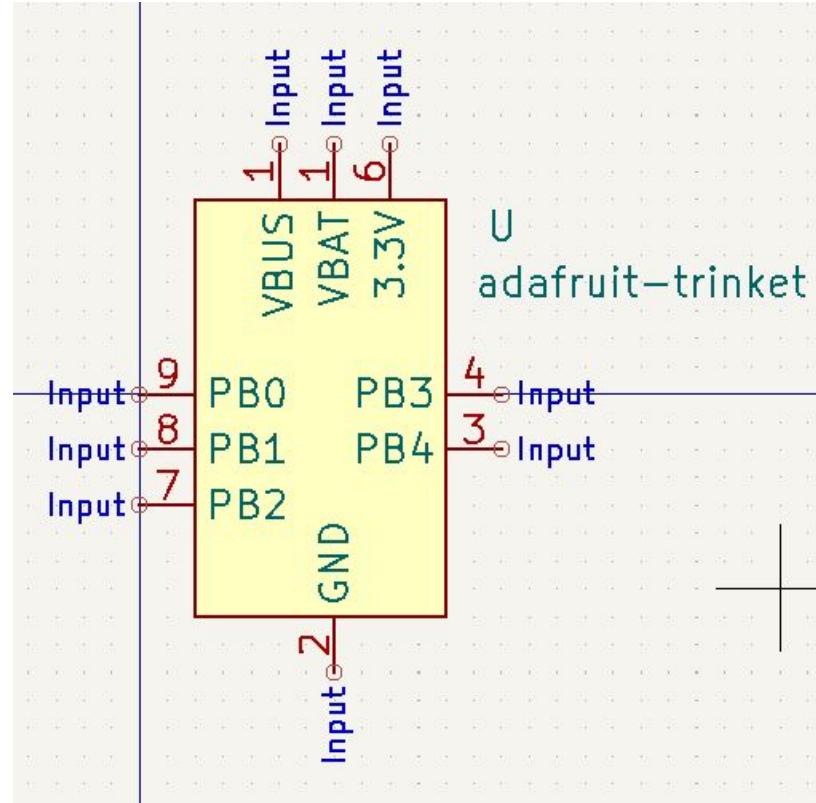
Except for Arduino the pins can be
Either so we just space them evenly

Two grid spaces between pins

(grid must always be set to .050 inch)

Ctrl+s to save

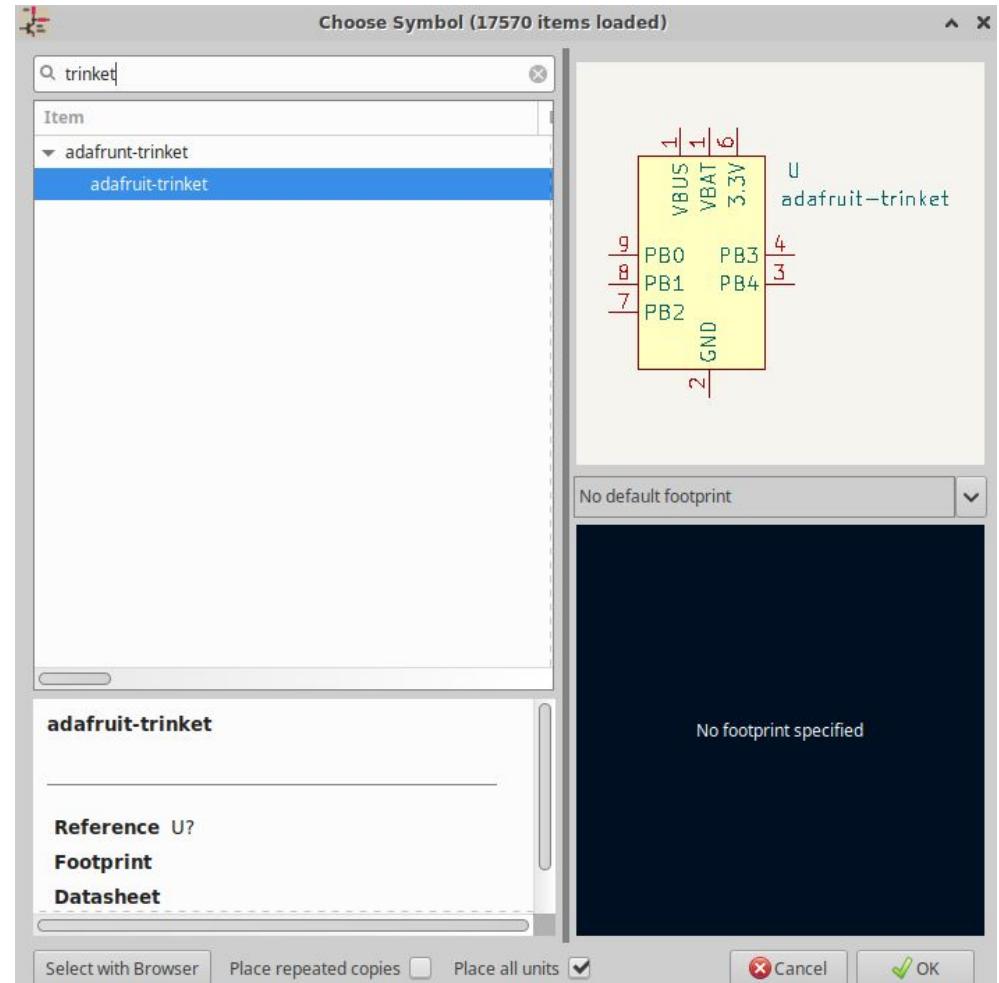
Close the symbol editor



Using the symbol

Now, when you type “trinket” in the part browser, you see your new symbol!

You can place it in the schematic and wire just like any other...



Coffee Break!

Go practice with the schematic and symbol editor until you are comfortable.

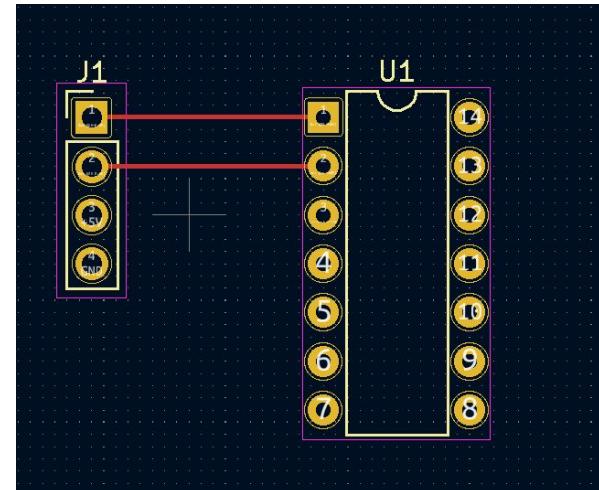
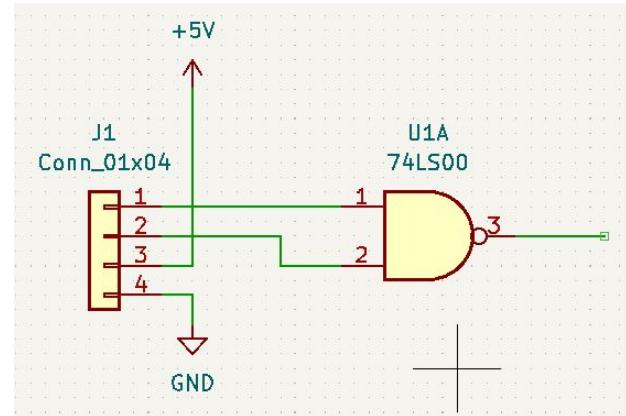


Footprints and moving to PCB layout

Every symbol must have a footprint

Many of these are in the library

Some symbols come with footprints, but you should always check to be sure it is the right one for your part.

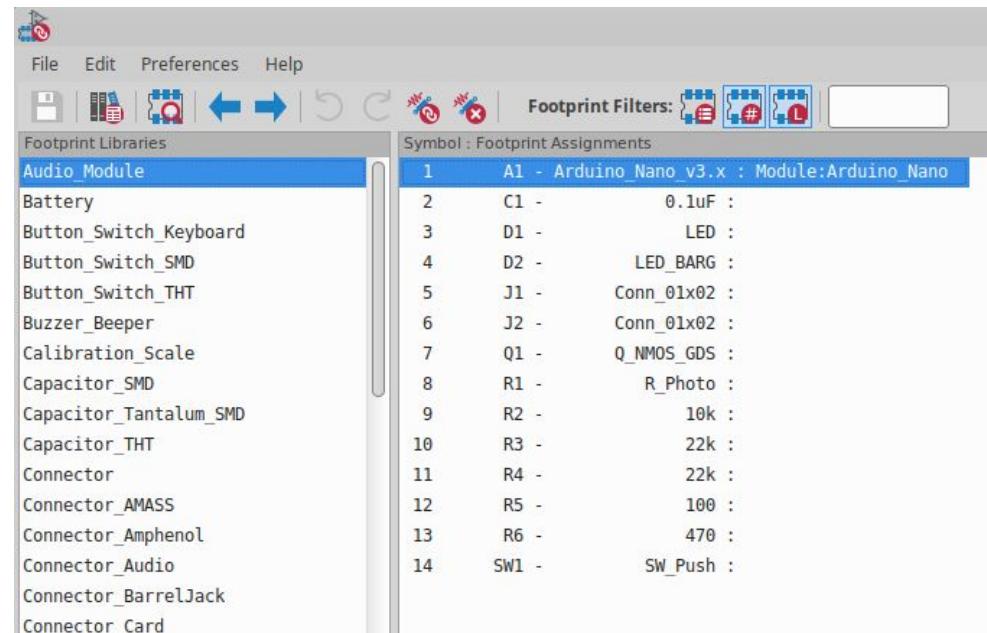


Footprint Assignment

Select Tools→Assign Footprints.

You'll see a list like this.

Only the Arduino Nano came with a pre-assigned footprint, so we have a lot of work to do :)



Symbol : Footprint Assignments		
1	A1 - Arduino_Nano_v3.x	: Module:Arduino_Nano
2	C1 -	0.1uF :
3	D1 -	LED :
4	D2 -	LED_BARG :
5	J1 -	Conn_01x02 :
6	J2 -	Conn_01x02 :
7	Q1 -	Q_NMOS_GDS :
8	R1 -	R_Photo :
9	R2 -	10k :
10	R3 -	22k :
11	R4 -	22k :
12	R5 -	100 :
13	R6 -	470 :
14	SW1 -	SW_Push :

Footprint from Library

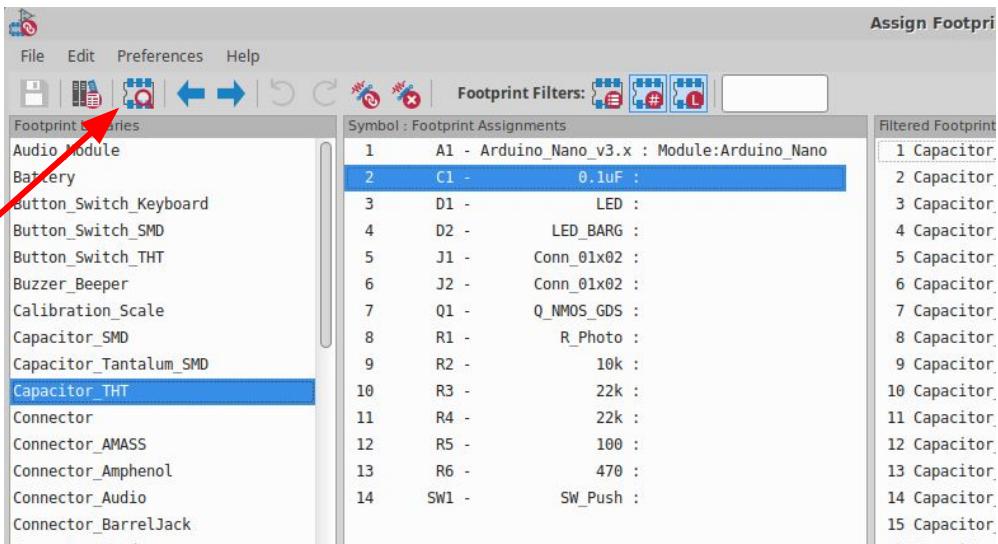
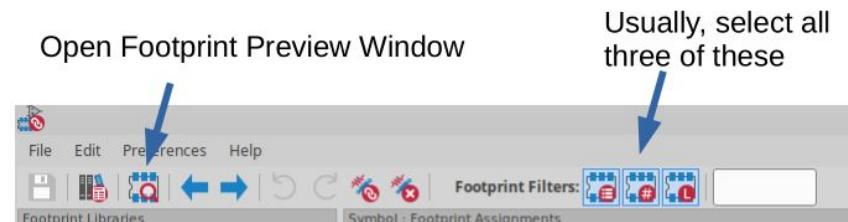
Select the first part with no footprint
(C1 in this case)



This is a capacitor, and we would like to use a through-hole one we found in our parts box which has about 0.2 inch pin spacing.

So, select the library “Capacitor_THT” (thru-hole) on the left.

Open the preview window to see the footprint you have selected in the right-most list.



So many capacitors!

Reading capacitor list...

Capacitor_THT:C_Disc_D4.7mm_W2.5mm_P5mm

C_Disc - disc type capacitor

D4.7mm - disc diameter (not important)

W2.5mm - thickness (width) (not important)

P5mm - pin spacing (**important**)



We have a capacitor which looks like the one above and the pin spacing is about 0.2 in or about 5mm. So this is a reasonable choice.

Double-click on the footprint in the right list to choose this footprint.



Next, the LED

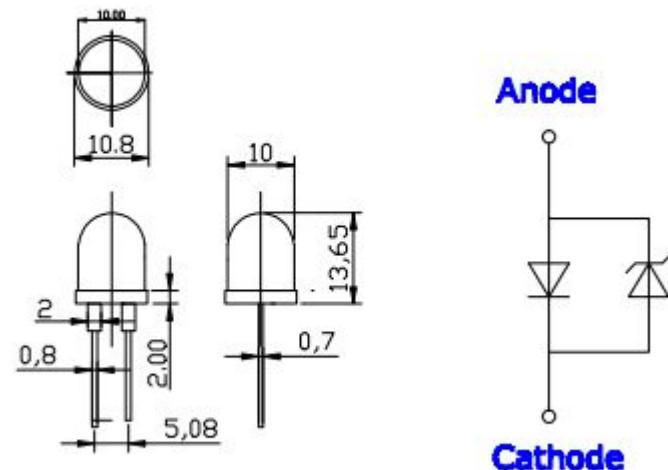
Let's say you picked this one from sparkfun.

Find the drawing in the datasheet
with dimensions.

Most important are the 5.08 mm lead spacing,
and the 10mm diameter.

Look in the “LED_THT” library.
The only real candidate is “LED_D10.0mm”
Which has a 2.54mm lead spacing.

Let's modify it for our use.



Modifying a library footprint

Note the name of the part

Close the footprint assign tool

Open the footprint editor (in the design manager)

Create a new library (**File**→**New Library**).

Select “Project”. Name it “my-LED-10mm” or something.

DO NOT name the library the same as a supplied one

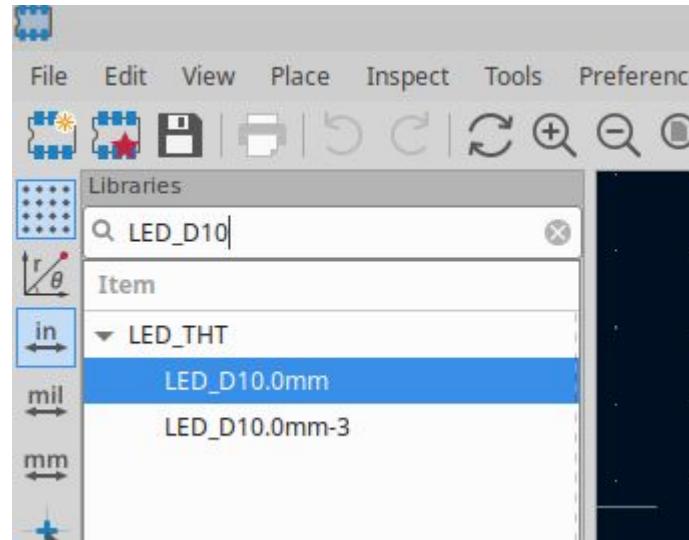
Type the name (“LED_D10”) in the search box below “Libraries”.

Double-click on the part to edit.

Save a copy:

File→**Save As**. Find your library and select it, enter a name such as “**LED_D10.0mm_P5.08mm**”.

Confirm that you are now editing your part, look at the window banner:



Modifying a footprint - II

What we want to do is to move the Pins so they are 0.2 in (5.08mm) spaced.

Set the grid to 0.05 in.

First, switch to Inch mode:

Then right-click on the black Area and select
Grid→0.05in (1.27mm)

Finally, move each pin one grid Space away from each other, So the spacing is 0.2 in (5.08mm)

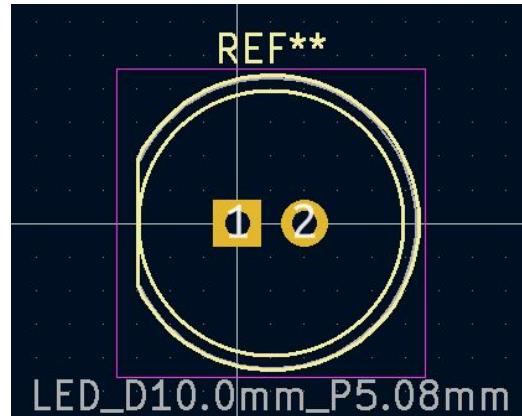
Measure distance by putting the cursor over one item And hitting **SPACE**, then you can read the distance at the bottom Of the screen:

Don't forget
To save!

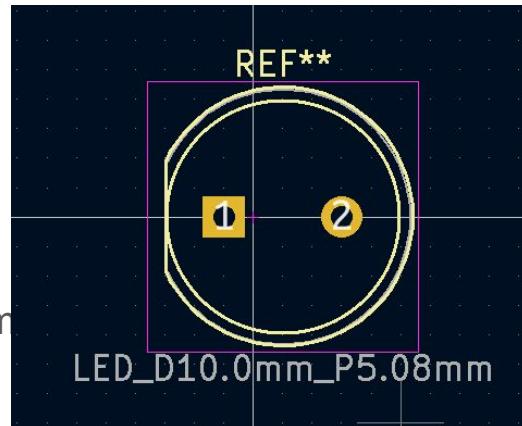
dx 0.2000 dy 0.0000 dist 0.2000



Before:



After:



Parts assignment, onward!

Back to the “Assign Footprints” tool

For the LED, find your part and assign:

Connectors are usually easy. For standard 0.1" headers, use the library

`Connector_PinHeader_2.54mm` and choose the “Vertical” option

The RGB LED should fit in:

`LED_THT:LED_D5.0mm-4RGB`

The screenshot shows two tables of footprint assignments. The top table is titled 'Symbol : Footprint Assignments' and lists the following assignments:

1	A1 -	Arduino_Nano_v3.x : Module:Arduino_Nano
2	C1 -	0.1uF : Capacitor_THT:C_Disc_D4.7mm_W2.5mm_P5.00mm
3	D1 -	LED : my-LED-10mm:LED_D10.0mm_P5.08mm
4	D2 -	LED_BARG :

The bottom table lists footprint assignments for connectors:

D2 -	LED_BARG :
J1 -	Conn_01x02 : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
J2 -	Conn_01x02 : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
Q1 -	Q_NMOS_GDS :

Below the tables is a product listing for an RGB LED:

LED - RGB Diffused Common Anode
COM-10821
\$2.25

ADD TO CART

REF**

LED_D5.0mm-4_RGB

More parts...

Here's a **MOSFET** (overkill!)

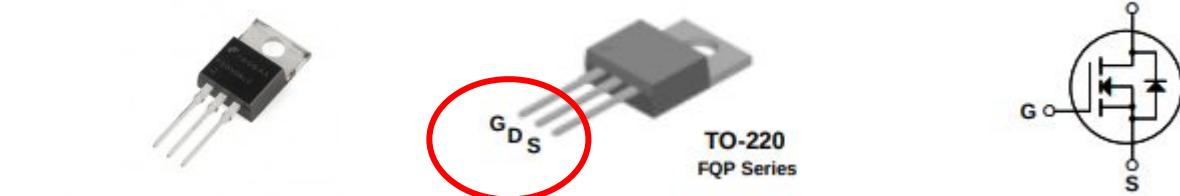
Now you understand why
I chose the “GDS” symbol...

Footprint: Package_TO_SOT_THT:TO-220-3_Vertical

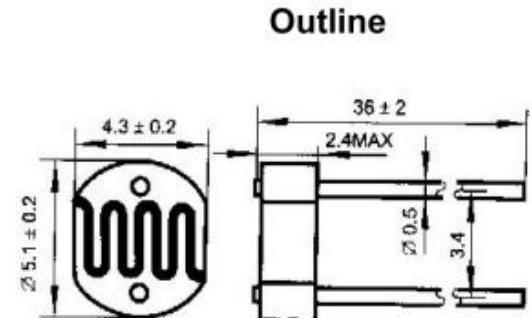
Next the Light Sensor

Footprint:

OptoDevice:R_LDR_5.1x4.3mm_P3.4mm_vertical



Mini Photocell
SEN-09088
\$1.60
4.5 5 5 5 5 7



Resistors



Let's use surface mount "0805" size resistors.
They're big enough to be easy to solder

Footprint: Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder

Note that all resistors get the same footprint even though they have different resistance values

7	Q1 -	Q_NMOS_GDS : Package_T0_SOT_THT:T0-220-3_Vertical
8	R1 -	R_Photo : OptoDevice:R_LDR_5.1x4.3mm_P3.4mm_Vertical
9	R2 -	10k : Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder
10	R3 -	22k : Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder
11	R4 -	22k : Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder
12	R5 -	100 : Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder
13	R6 -	470 : Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder
14	R7 -	470 : Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder
15	R8 -	470 : Resistor_SMD:R_0805_2012Metric_Pad1.20x1.40mm_HandSolder
16	SW1 -	SW Push :

Last, the switch!

Let's use a waterproof one with a rubber cap. The datasheet has the PCB layout and a table of hole sizes. We have to make a new footprint for this one...

PCB LAYOUT, MOUNTING SIDE

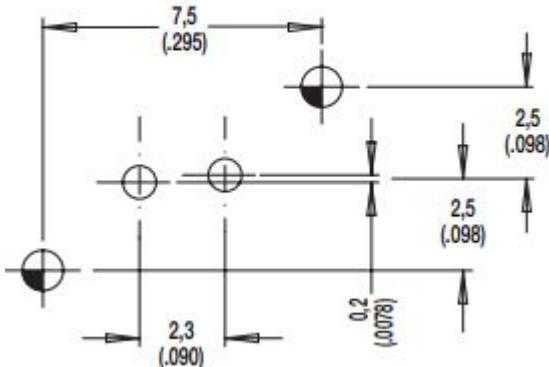


Image shown is a representation only. Exact specifications should be obtained from the product data sheet.



K12C 1 5N

Digi-Key Part Number	401-1159-ND
Manufacturer	C&K
Manufacturer Product Number	K12C 1 5N
Description	SWITCH PUSH S
Manufacturer Standard Lead Time	8 Weeks
Detailed Description	Pushbutton Switch Hole
Customer Reference	Customer F
Datasheet	Datasheet

Hole	Ø	Without LED	Description	Terminal Section	Surface
	1,7 (.069)	2x	snap-in		
	1,6 (.062)	1x	coding hole (L,M,N)		
	1,1 (.043)	2x	center hole		Sn
	0,9 (.035)	2x	LED	m0.5 (.020)	Sn
		2x	switch	0.7 x 0.2 (.028 x .081)	

Custom footprint from scratch

1. Open the footprint editor
2. Create a footprint library in the project (“my-rubber-button”)
3. Create a new footprint: **File→New Footprint**. Select “Through hole”
4. Name it something like “ck-switch-k12c” (just remember it)

5. Select the pad tool in upper right and click on the origin
6. Use “M” (move) to move the text out of the way
7. Cursor over the pad, click, and press “e” to edit
8. Change the hole size to 0.035 in (from the datasheet)
(the pad size is OK at 0.06 in)
9. Place another pad, use “e” edit to move it to
(X=.09, Y=-.0078). See the datasheet.



Custom footprint continued

10. Add two more pads for the other holes (0.043 in dia. from datasheet)
11. Set the locations based on the drawing and *double-check* them
12. Draw a circle of radius 0.25" centered for the button outline
On the “F.Silkscreen” layer
(click on the layer on the right-hand list before adding the circle)
13. Save the footprint in your library (“my-rubber-button” or whatever)

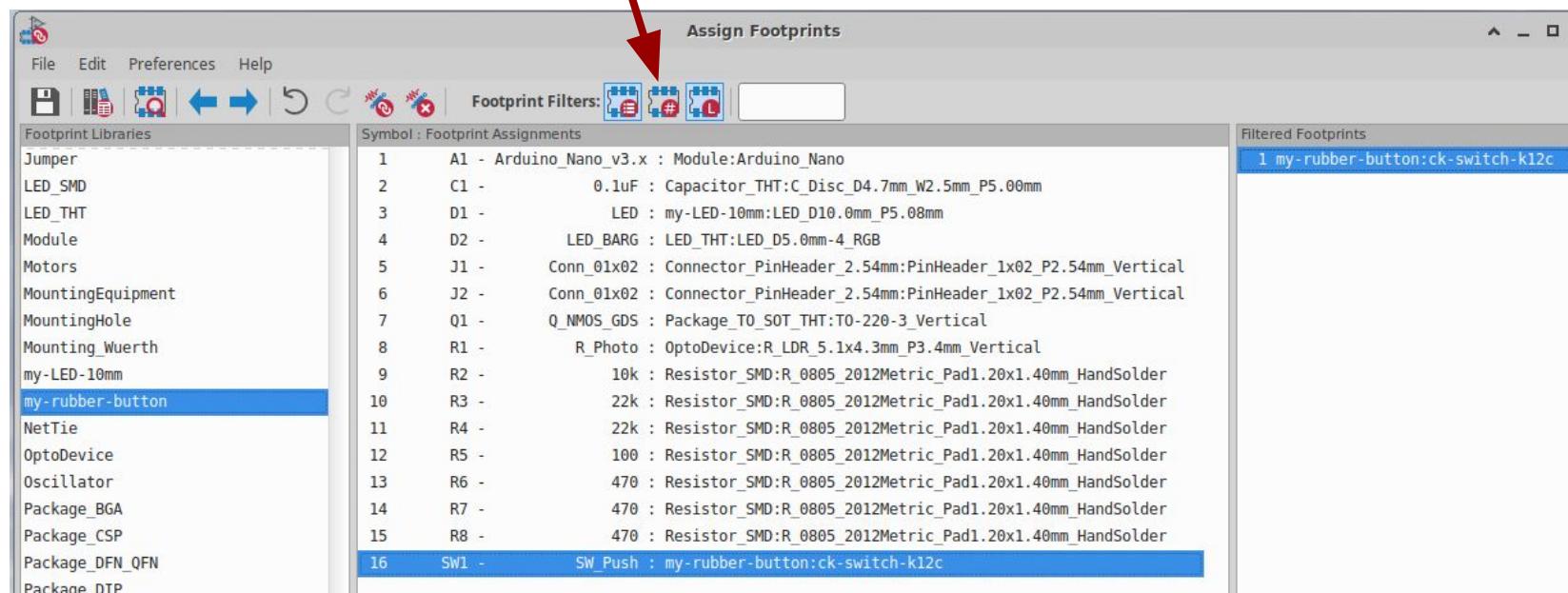
Close the footprint editor

Whew!



Assign your new footprint to the switch

One tricky thing... it won't find your footprint unless you un-select the middle filter, because your new footprint has 4 pins (two are mechanical pegs) so the pin count doesn't match the symbol, which has only 2 pins.



Now all footprints are assigned!

Finding symbols, footprints, 3D models online

For example, DigiKey:

Product Index > Sensors, Transducers > Encoders > Bourns Inc. PEL12D-4225S-S2024



Image shown is a representation only. Exact specifications should be obtained from the product data sheet.



PEL12D-4225S-S2024

DigiKey Part Number	PEL12D-4225S-S2024-ND
Manufacturer	Bourns Inc.
Manufacturer Product Number	PEL12D-4225S-S2024
Description	ENCODER MECH QUAD VERT PC PIN
Manufacturer Standard Lead Time	25 Weeks
Customer Reference	<input type="text"/>
Detailed Description	Rotary Encoder Mechanical 24 Quadrature (Incremental) Vertical
Datasheet	 Datasheet
EDA/CAD Models	PEL12D-4225S-S2024 Models



Finding symbols, footprints, 3D models online

For example, DigiKey:

Product Index > Sensors, Transducers > Encoders > Bourns Inc. PEL12D-4225S-S2024



Image shown is a representation only. Exact specifications should be obtained from the product data sheet.



PEL12D-4225S-S2024

DigiKey Part Number	PEL12D-4225S-S2024-ND
Manufacturer	Bourns Inc.
Manufacturer Product Number	PEL12D-4225S-S2024
Description	ENCODER MECH QUAD VERT PC PIN
Manufacturer Standard Lead Time	25 Weeks
Customer Reference	<input type="text"/>
Detailed Description	Rotary Encoder Mechanical 24 Quadrature (incremental) Vertical
Datasheet	 Datasheet
EDA/CAD Models	 PEL12D-4225S-S2024 Models

Click the link EDA/CAD Models



PEL12D-4225S-S2024 Footprints and Models

PEL12D-4225S-S2024
Bourns Inc.
ENCODER MECH QUAD VERT PC PIN

Manufacturer Provided >

Ultra Librarian >

View on Ultra Librarian

Symbol	Footprint	3D Model
		

Set download format To KiCAD 6 or later



Select Download Format

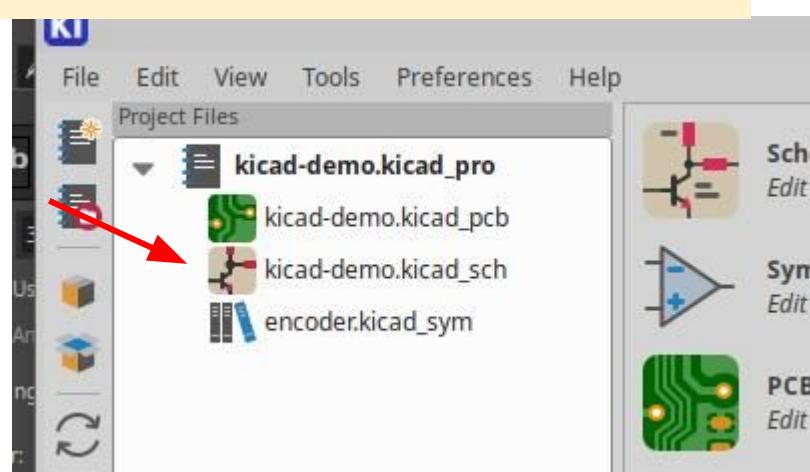
This will usually create a zip file with the Symbol, footprint and often there is a 3D model too

How to install models (symbol)

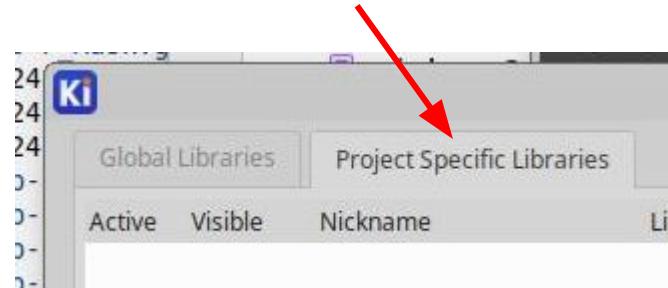
1. Extract the zip file



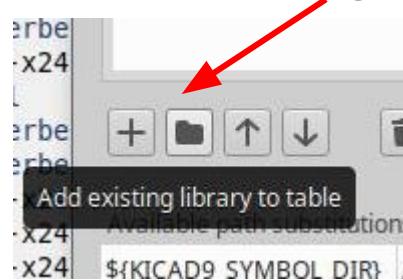
2. Copy the **kicad_sym** file to the project (perhaps giving it a better name like “encoder.kicad_sym”)



3. Select Project Specific Libraries



4. Add existing library...



The symbol should now be available to use in your schematic

Coffee Break!

Take a break before moving on
to the actual board...



Starting the board

1. Open the PCB Editor
2. Click **Tools**→**Update PCB from Schematic**
You should get a dialog box with a list of parts added and no warnings/errors
Click **Update PCB**, then **Close**
3. Click somewhere to drop the pile of components.
4. Create a board outline. Best is to import a DXF, or you can draw it using lines on the “Edge.Cuts” layer. To import:
File→**Import Graphics**
Select the layer “Edge.Cuts”. Click OK and place the outline on the page.
(location is not important).

Set PCB design rules

Select **File**→**Board Setup**.

On the left, click **Physical Stackup**. Change “Copper layers” from 2 to 4.

Click **Adjust Dielectric Thickness** and enter “0.063 in”.

(boards can have from 2 layers [top/bottom only] up to 32 or more.

4 layers is a good compromise as you have 2 layers for your signals, and two other for ground and power).

There are many other settings under Design Rules / Constraints.

The defaults are OK for simple boards.

Board Outline and parts

I drew the board outline in A 2D CAD and saved as a DXF file. Then imported Into KiCAD on the **Edge.Cuts** layer.

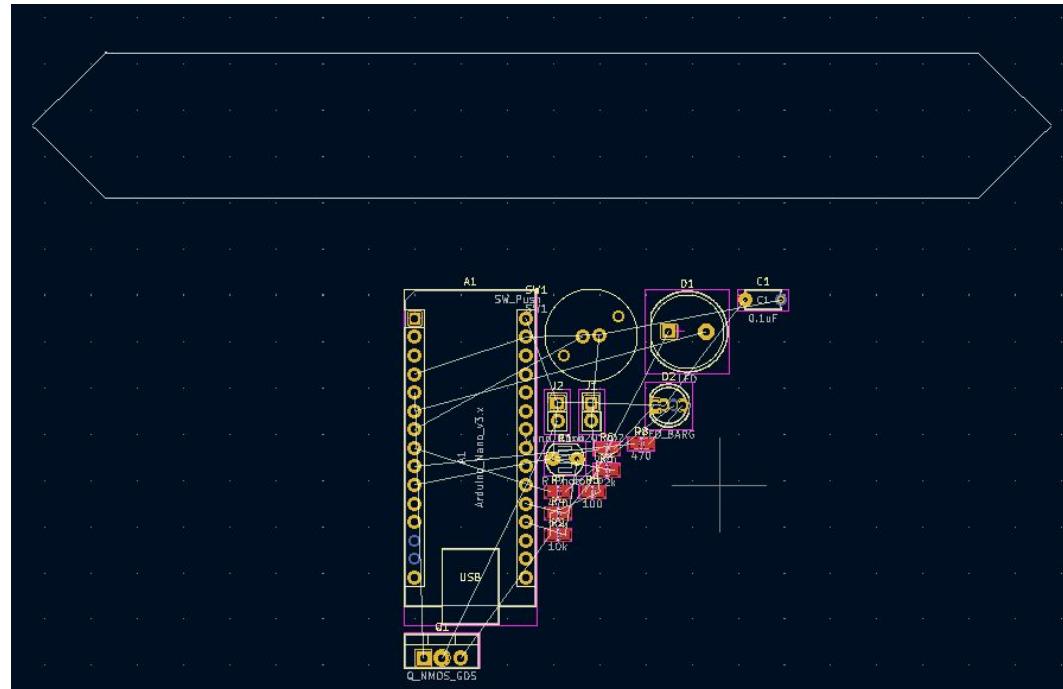
You can also just draw it in KiCAD using the line tool on **Edge.Cuts**



The next task is to place all the parts in appropriate locations.

Make sure they don't conflict with any mechanical features.

Parts can go on both sides of the board, but this can get hard to understand visually



Component Placement

...is an art, but here are some suggestions:

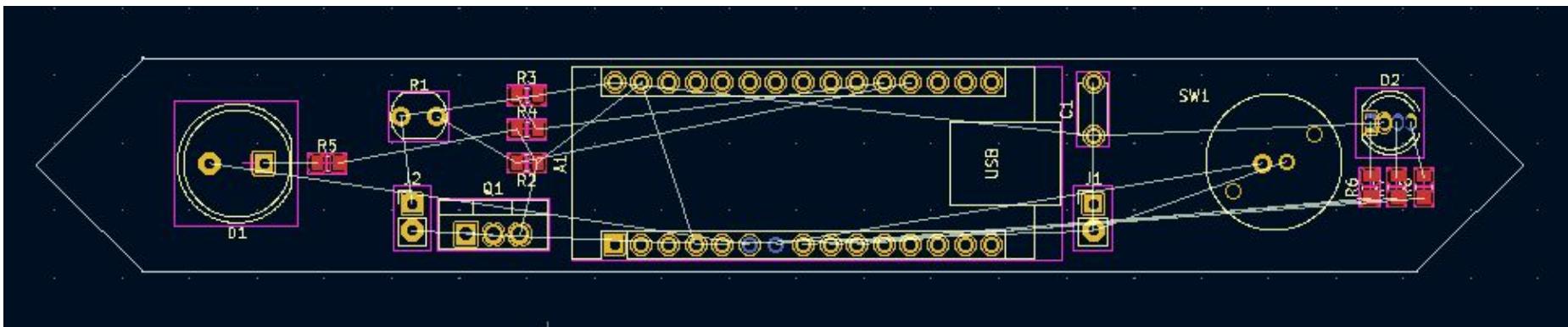
- Set a grid, 0.025 in or 0.5 mm so things line up
- Place parts which have mechanical constraints first (switches, connectors)
- Components can be placed on either side (front or back)
- Point cursor at parts, “m” to move, “r” to rotate, “f” to flip to back side
- Keep the schematic editor open at the same time (two monitors helps) so you can see which component you have selected
- Mind the “rats nest” of white lines which show connections
- Place components in neat rows, not too close together, for easier soldering

Here is my placement for the toothbrush board

White LED

Light sensor

RGB LED



MOSFET

Arduino

Push
button

Layers Layers Layers...

There are lots of drawing layers, but here are the most important:

Edge.Cuts the board outline

F.Cu, B.Cu copper traces and solder pads on front and back

In1.Cu, In2.Cu copper traces on inner layer 1 and 2
(usually In1 and In2 are power and ground)

F.Silkscreen (B) front and back silkscreen (white text / lines)

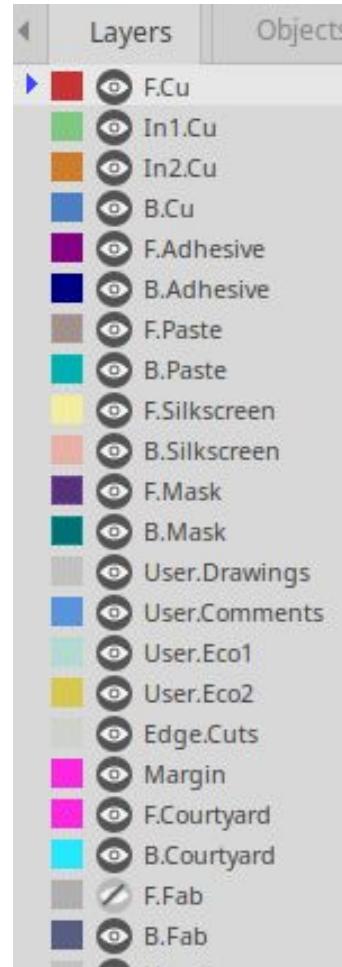
F.Mask, B.Mask solder mask (typically green)

F.Paste, B.Paste solder paste areas (for stencils)

Most of these are handled “automatically”... footprints contain graphics and text on many layers.

Traces (wires) are usually drawn only on F.Cu and B.Cu

Text notes and logos (Rhett!) should be drawn on F.Silkscreen

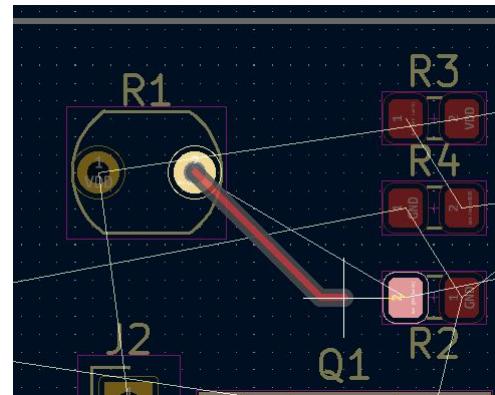


Routing (wiring)

“Routing” is the process of connecting the wires between parts. Use the routing tool.

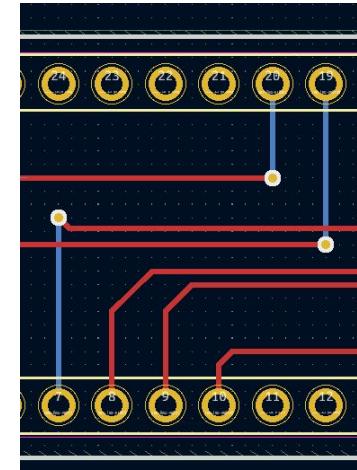
Click the routing tool. Click on a pad at one end of the track and drag to the other end. If it’s a short connection it will be made automatically. Click on intermediate points to run a track.

Do not route power (e.g. VDD) or GND tracks now; they will be handled later on inner layers.



Routing II

- Two tracks cannot cross on the same layer or they make a short circuit.
- Change layers using “vias” which are a plated hole to the other side
- Type “v” and click to make a via. You will automatically be switched to the opposite side of the board.
- A useful strategy is to run mainly horizontal tracks on one layer and vertical on the other. Vias are “free” so no need to avoid them.
- Surface mount parts (red or blue pads) can only connect on the layer they are mounted on. Use a via nearby to get to other layers if needed.



Power and Ground

On a 4-layer board with only one power supply like this one, it is “easy”.

You have to create “filled zones” on layers “In1.Cu” and “In2.Cu”.

Select the Filled Zone tool.

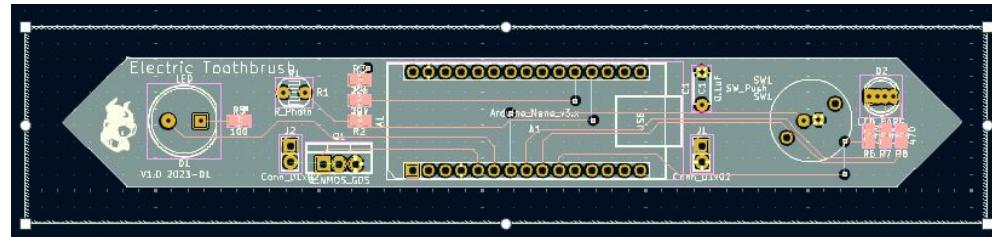
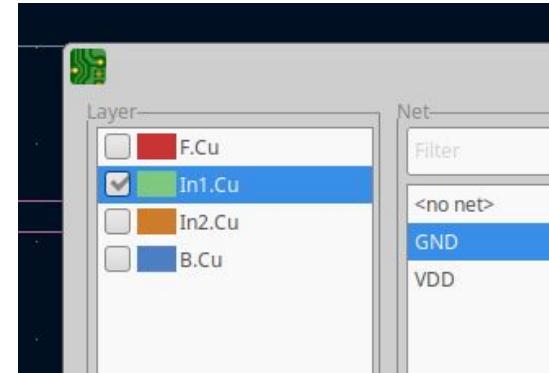
Draw a rectangle enclosing the board.



For the GND zone: check In1.Cu only
And select the GND net. Click **Ok**.

Type “B” to update the filled zone.

The zone connects to all thru-hole GND pads.

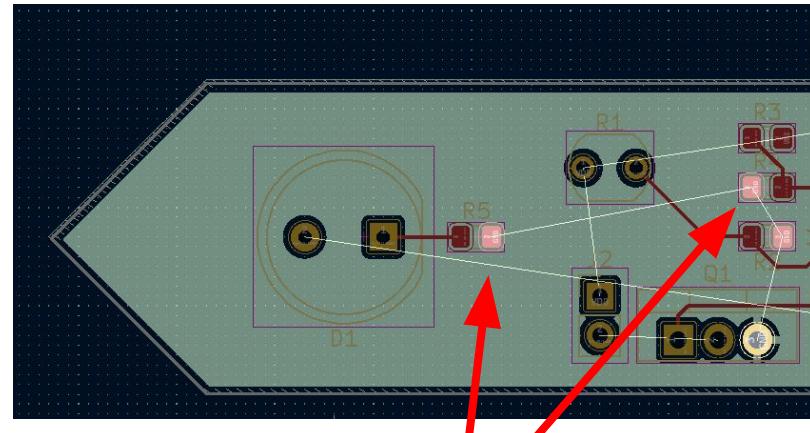


Power and Ground II

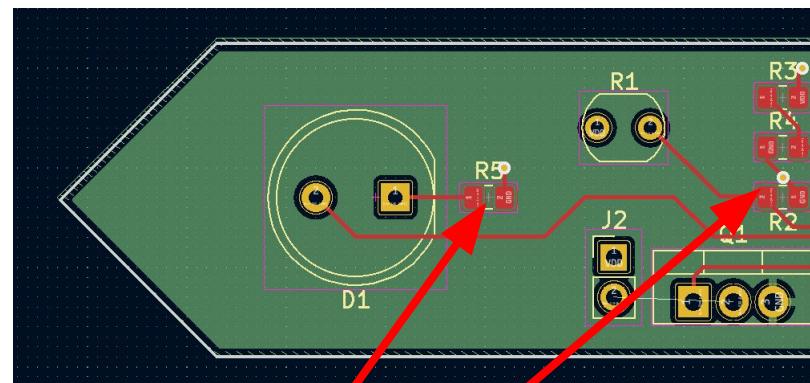
The filled zone only connects on one (inner) layer, so SMT pads aren't connected.

Connect them by routing a short track And placing a via ("v"). This will connect To the GND zone on In1.Cu.

Do the same (create a zone on In2.Cu) for the power (VDD) net.



Unconnected GND pads

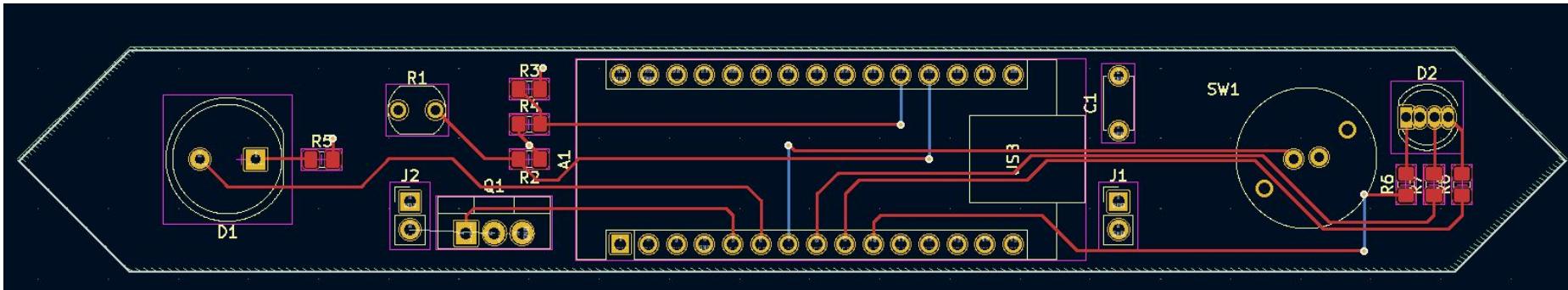


Connected GND pads

Finished routing

Here is how the board looks with all the connections routed.

Note the convention of using mostly 45 degree bends... this is historical but makes the board look more professional.



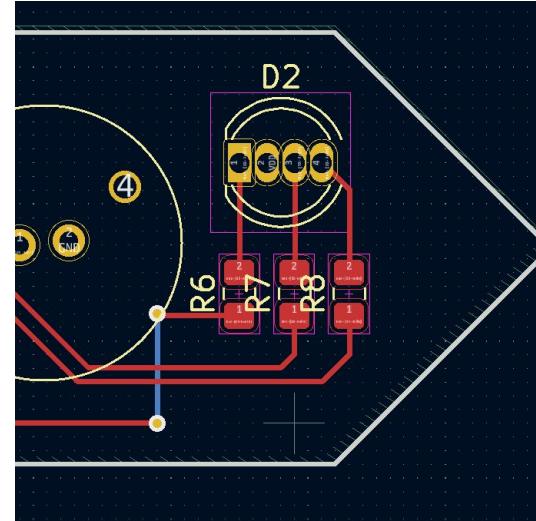
Silkscreen cleanup

The silkscreen is the white text on the board for humans.

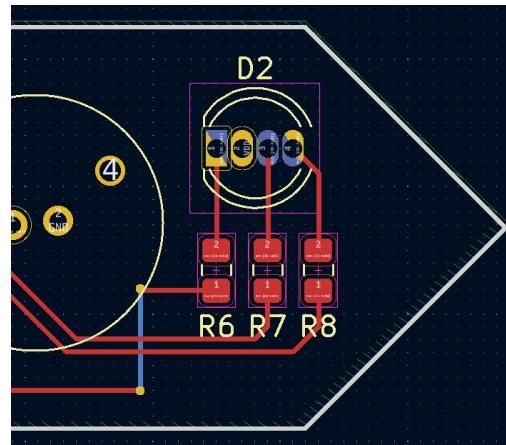
Each component has a reference (e.g. “R3”) which should be placed near the component, not overlapping any other component.

And, ideally right-side up.

Before:



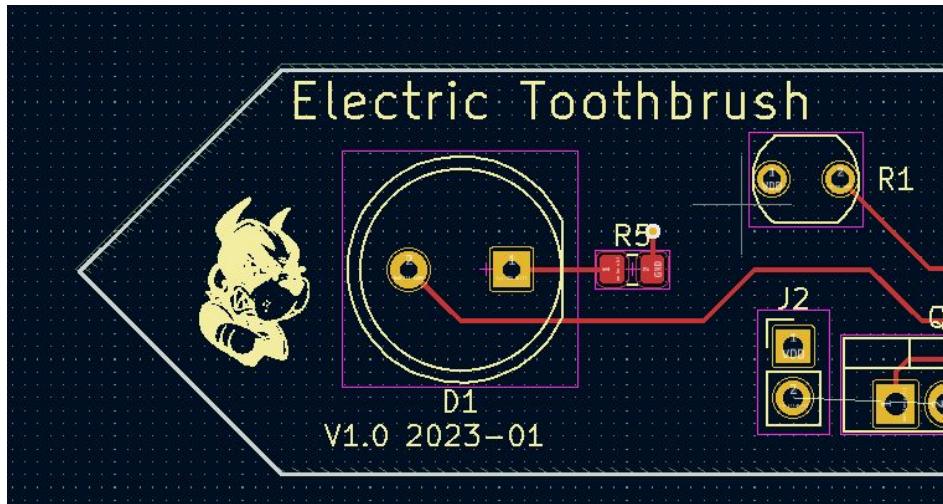
After:



Final things on silkscreen

You can and should label the board, including a date and revision.

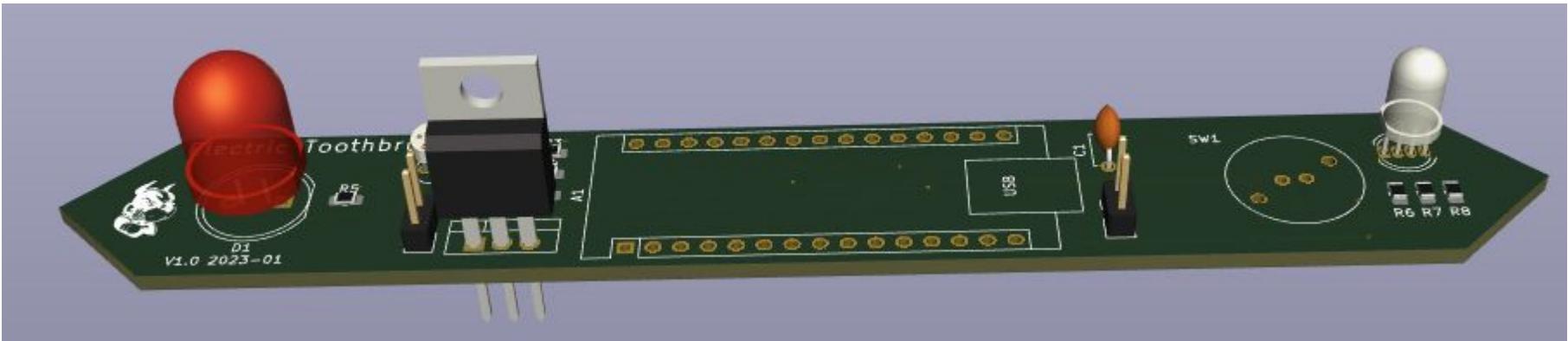
The “Image converter” allows you to put other things like Rhett on the board!



3D view!

Click **View→3D Viewer**.

Note that some parts don't have 3D models... you can Google or ask me how to add them.



Coffee Break!

Take a break and check everything
On the board very carefully (esp the
mechanics) before proceeding.

The Plot function  allows you to
Export specific layers as DXF so you can
Check against mechanical models.



Generating manufacturing files (Gerber, Drill)

Select the Plot function.



Select the layers shown.

Note **Edge.Cuts** selected too!

Create a directory “gerber” for output.

Click **Plot** then

Generate Drill Files,

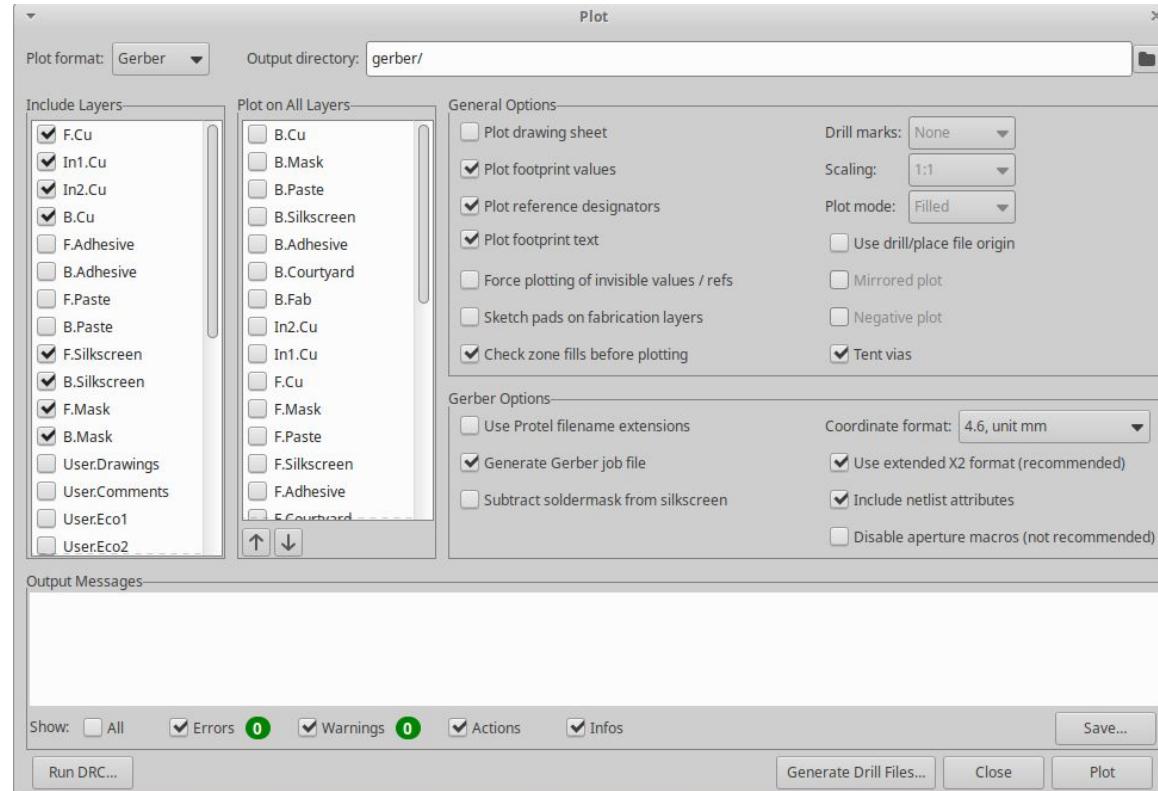
Generate Drill File and Close

Use the “Gerber Viewer” to check everything.

Make a zip with all the gbr and drl files.

This is what you will upload for manufacturing.

Name it e.g. **toothbrush-v1.0.zip**



Manufacturing options

These prices are just examples... they were correct as of mid 2025...

<u>Manufacturer</u>	<u>General Info</u>	<u>Price for Toothbrush board</u>
Osh Park:	\$10 / in ² for 4-layers Ship within 14 days but usually faster	e.g. \$50 for lot of 3 for this board
JLCPCB (China)*:	100x100mm \$7 Build time 2-3 days plus ~1 week ship	4 layer \$25 plus \$24 shipping 2 layer \$ 5 plus \$24 shipping
PCBWay (China):	Build time 4-5 days plus shipping	\$47.00 plus \$24 shipping
ExpressPCB (USA):	start at \$110	\$230 plus shipping

“Full service” PCB houses (Advanced, Sunstone) in USA will charge from \$300-\$1000 for these, but you can get them in 1-2 days if needed.

Notes: 2-layer is cheaper, though you have to route power and GND on the front/back layers.
ExpressPCB has a relatively cheap fixed 3.8x2.5 inch size
JLCPCB is very cheap for 100x100mm and the price varies with size (try the website!)

Additional Notes

Trace Widths

The default 0.007" width is fine for signal traces

(Much) wider traces or a plane are needed for GND

Best is a solid plane

Somewhat wider traces are needed for power.