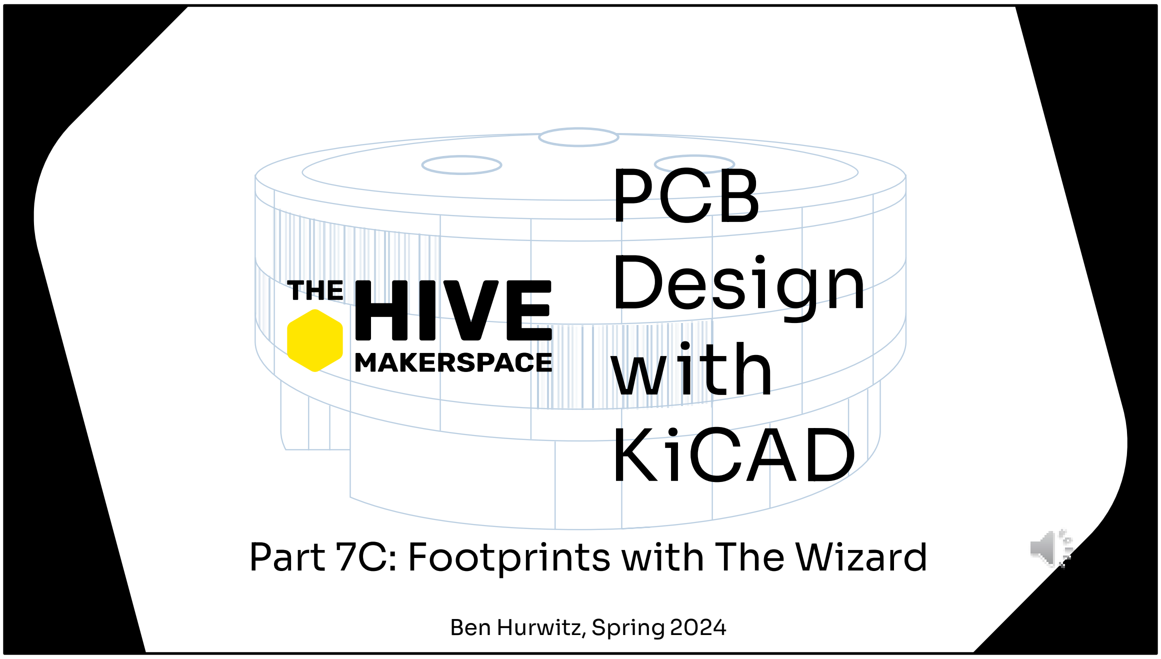


Hi all, welcome to The Hive's series on PCB Design with KiCAD. My name is Ben, and in this series, we've been walking through the PCB design process using KiCAD as our electronics design software.

The previous videos went through the design process all the way through, resulting in a complete PCB ready for fabrication. One thing that I mentioned during that process, and was featured in the original "EDA Design Flow" in part 2, was library management, and the idea of using only project-scope libraries, but when we actually did the design, I ignored this for simplicity and time-constraints.

Part 6 went through a project-specific symbol library.

In parts 7A and B, we've made a custom library and a custom footprint using the "blank slate" method.

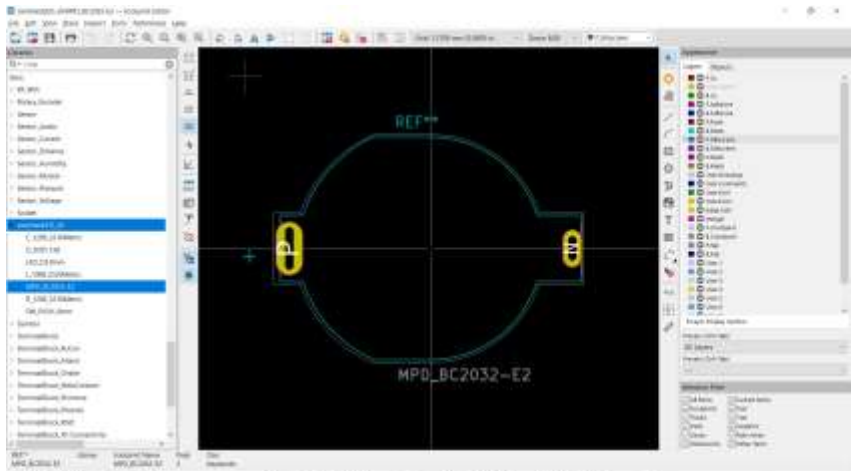
In this video, we'll use the wizard to make a footprint for our IC, then decide never to repeat that if possible.

This material is of course not required for a functional design, but it is good design practice, for KiCAD at least, to keep all your parts in a project-level library.

Because this is not related directly to the design flow of the previous videos, I'll make no assumptions about the state of your system or knowledge. So I apologize if some of this is repetition for some of you.

Let's get started.

Footprint Library



As a review, this is where we left the last video, with seven of the required eight footprints in our project-specific library. Obviously, this battery holder was downloaded, not made by me, so if you saved your personal copy rather than the downloaded one, first, kudos to you, but second, it'll obviously look different unless you spent your precious time making this, in which case, your employer will not be happy. But I digress.

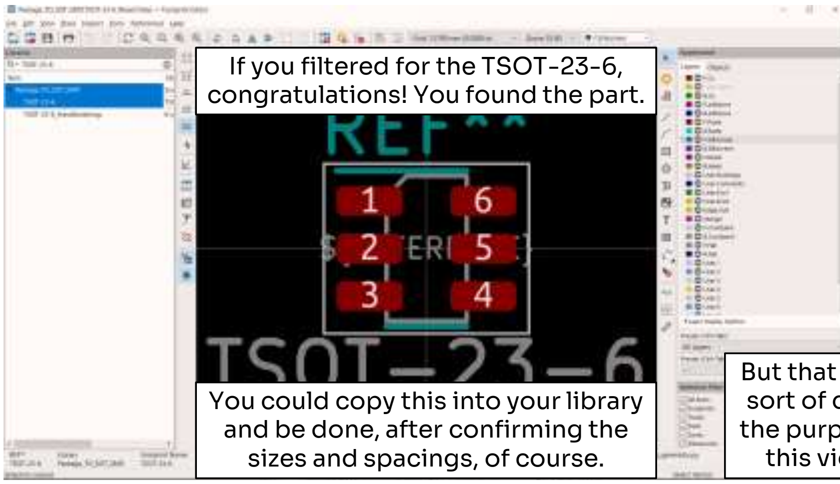
Footprint Library

Finally, onto the IC.

ICs are usually either a standard package with a known footprint, or based off a standard package, *but always confirm the supplier-provided package with the datasheet!*

KiCAD has a large selection of standard footprints that *should* work. So the first place to check for the footprint is in KiCAD itself with the filter box in the Footprint Editor. Filter for the part number and the package name.

Footprint Library



If you filtered for the TSOT-23-6, congratulations! You found the part.

You could copy this into your library and be done, after confirming the sizes and spacings, of course.

But that would sort of defeat the purpose of this video.

Footprint Library

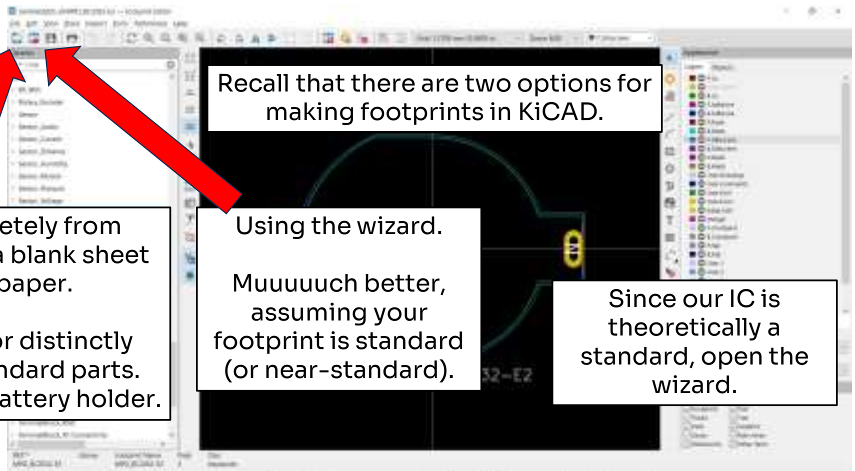
If we didn't find the *correct* package, we would next search online, either from your supplier, the manufacture, Ultra Librarian, or SnapMagic. Then it could be imported like we demonstrated with the battery holder.

If not, we can request it and wait.

The last recourse is to make it, as we saw with the battery holder.

But because this is a tutorial about KiCAD, we'll do that.

Footprint Library



Recall that there are two options for making footprints in KiCAD.

Completely from scratch, a blank sheet of paper.

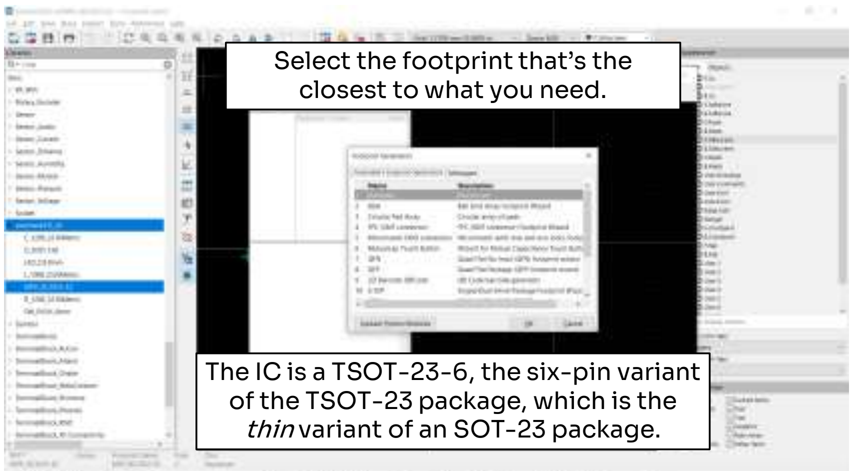
Good for distinctly non-standard parts. Like this battery holder.

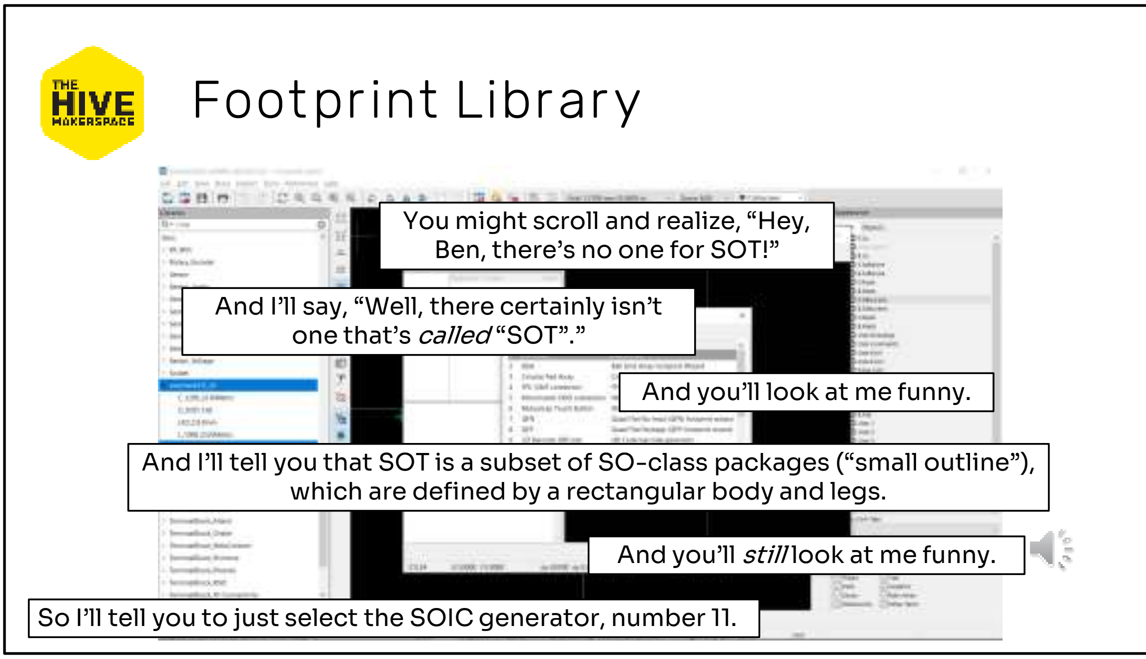
Using the wizard.

Muuuuuch better, assuming your footprint is standard (or near-standard).

Since our IC is theoretically a standard, open the wizard.

Footprint Library





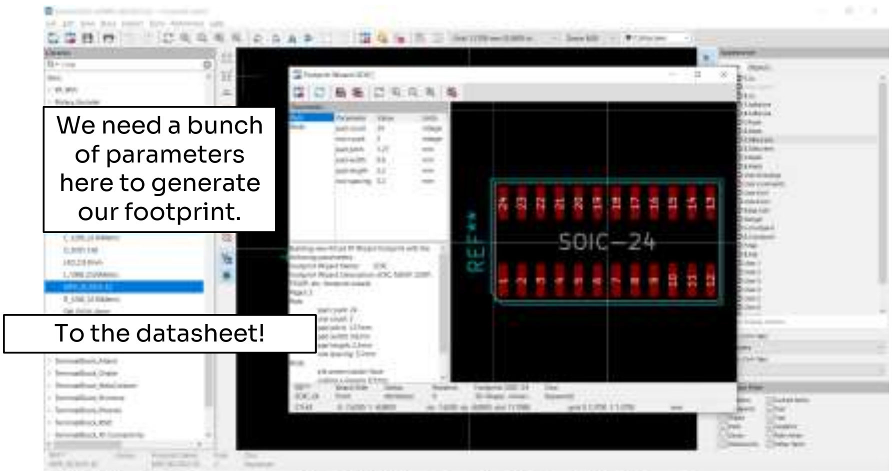
Because SOIC stands for “Small Outline Integrated Circuit” and is the overarching family that includes SOT package types, among others.

Aside: Package nomenclature

- Package nomenclature is *the worst*, and it's something you'll get familiar with as you design boards.
- There are so many standards, and the variation between some are so tiny. It can be really painful.
- The datasheet's mechanical drawings are the *one and only truth* for package sizes.
- Use caution!

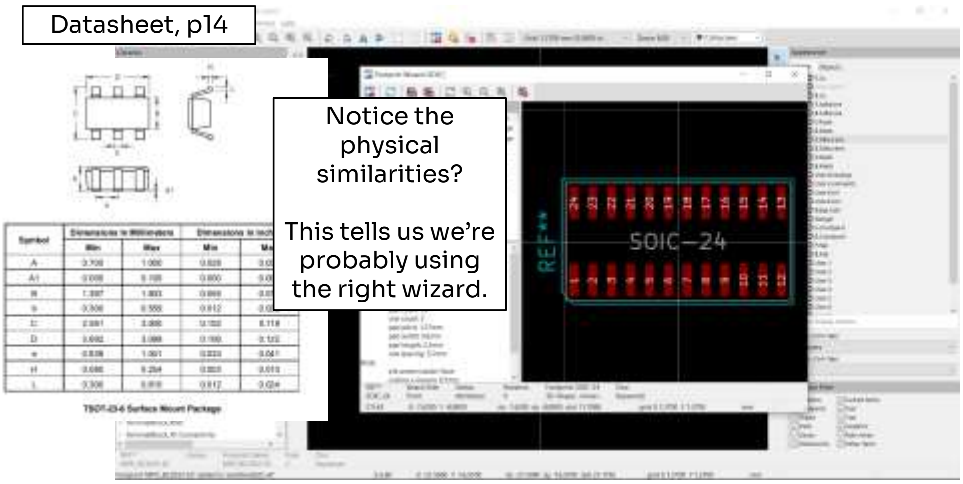
This is a very (very) small subset of package families.

Footprint Library



Footprint Library

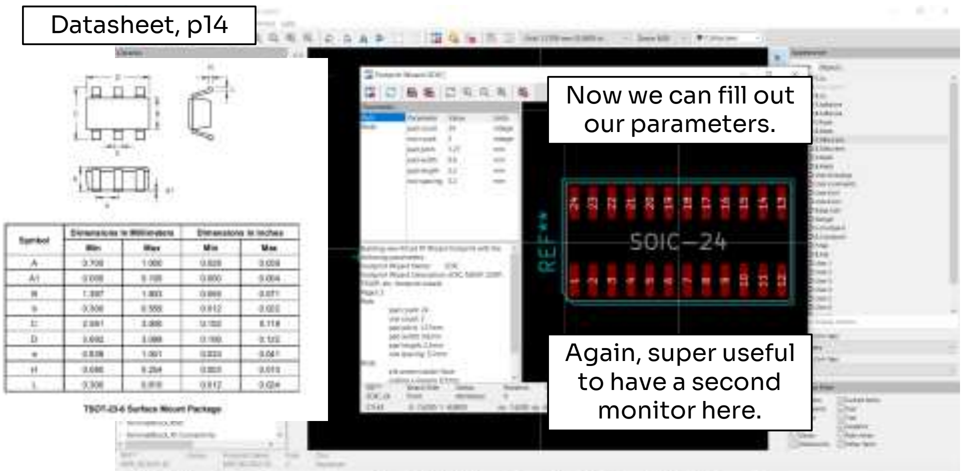
Datasheet, p14



Notice the physical similarities?

This tells us we're probably using the right wizard.

Footprint Library



So what the heck are all those parameters, anyway?

A – Package height

A1 – Seating plane height

B – Body width

b – Leg/pin width

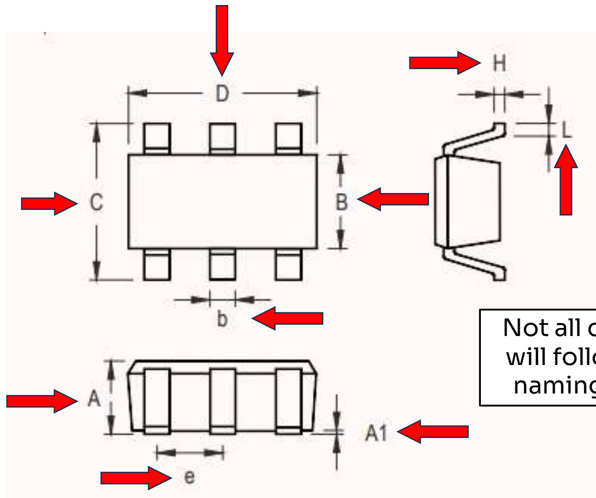
C – Package width

D – Package length

e – Leg/pin pitch

H – Leg/pin thickness

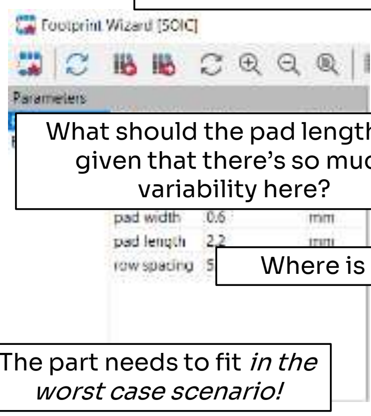
L – Leg/pin length

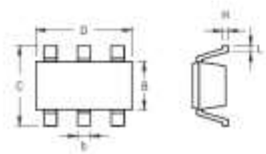


Not all datasheets will following this naming schema!

A few of thes requested parameters are readily available – pad count (6), row count (2), pad pitch (e), and pad width (b).

But there are a few tricky bits.





Should we be using min, max, or something else?

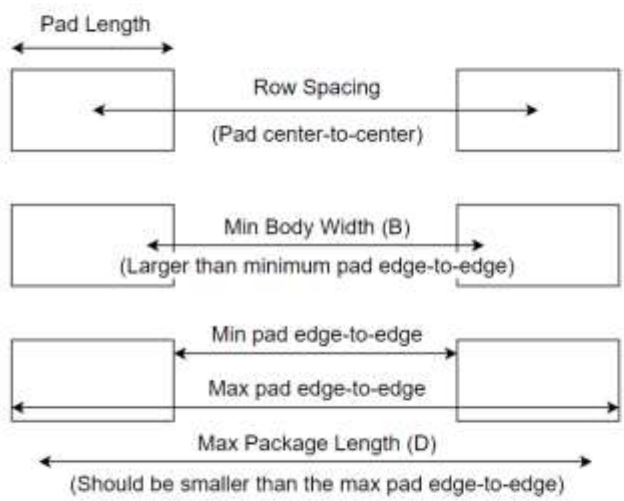
What should the pad length be, given that there's so much variability here?

Where is "row spacing" measured from?

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	0.700	1.000	0.028	0.039
A1	0.000	0.100	0.000	0.004
B	1.397	1.803	0.055	0.071
b	0.300	0.559	0.012	0.022
[illegible]	[illegible]	[illegible]	0.102	0.118
[illegible]	[illegible]	[illegible]	0.106	0.122
e	0.838	1.041	0.033	0.041
H	0.080	0.254	0.003	0.010

The part needs to fit *in the worst case scenario!*

How much bigger should the pads be than the legs themselves?

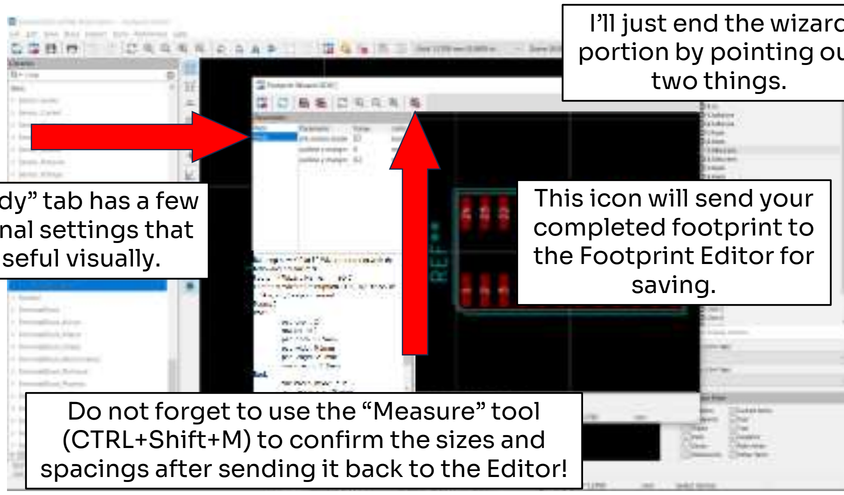


This is a mockup of what the footprint parameters are. The row-spacing is measured center-to-center, but the pad interior edge-to-edge is smaller than the minimum body width, and the maximum package length is shorter than the pad exterior edge-to-edge.

Still doesn’t answer the key question of what the pad length should be.

- Some datasheets are better about giving useful dimensions, or even better, a recommended footprint (like the battery holder did).
- Standard packages that are *not* in KiCAD still have a standard footprint that you can search for and use.
- Otherwise, you'll have to make your best judgement.
- You might be getting a sense for why companies should do this for you.
- Given all this, we're not actually going to make this part. It's not worth our time.

Footprint Library



I'll just end the wizard portion by pointing out two things.

The "Body" tab has a few additional settings that are useful visually.

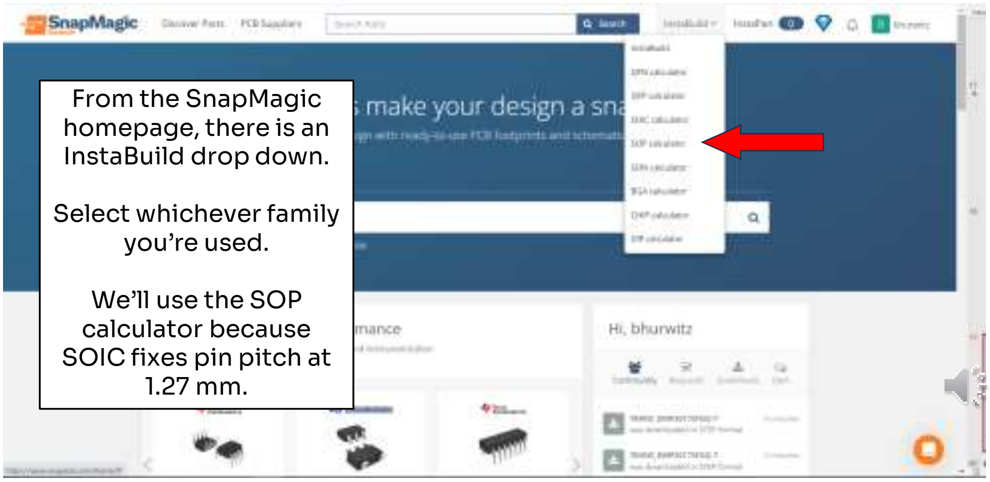
This icon will send your completed footprint to the Footprint Editor for saving.

Do not forget to use the "Measure" tool (CTRL+Shift+M) to confirm the sizes and spacings after sending it back to the Editor!

SnapMagic Footprint Generator

- Okay, so now we know multiple methods to make a footprint:
  - Make it (from scratch or through wizard)
  - Request/locate it (Ultra Librarian, SnapMagic, manufacture, or supplier) and import it
  - Use the built-in version, if available
- The last method I'll show you as an option is the SnapMagic InstaBuild footprint generator

SnapMagic Footprint Generator

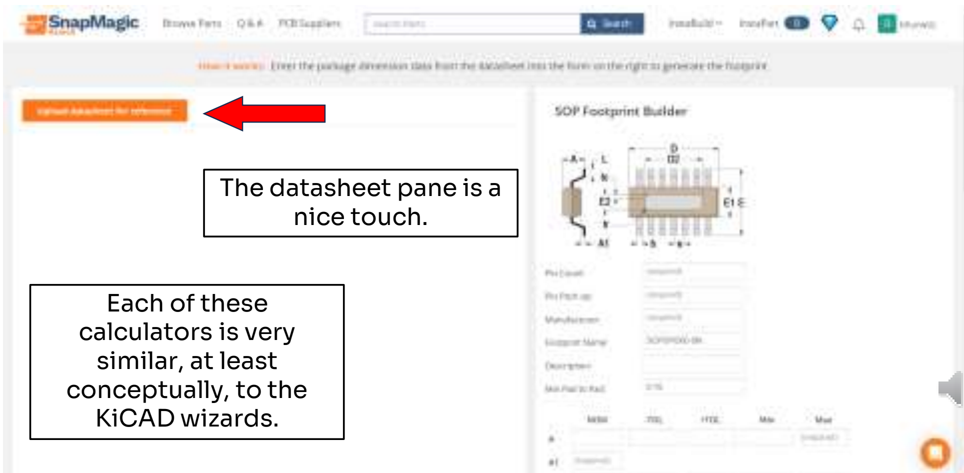


From the SnapMagic homepage, there is an InstaBuild drop down.

Select whichever family you're used.

We'll use the SOP calculator because SOIC fixes pin pitch at 1.27 mm.

SnapMagic Footprint Generator

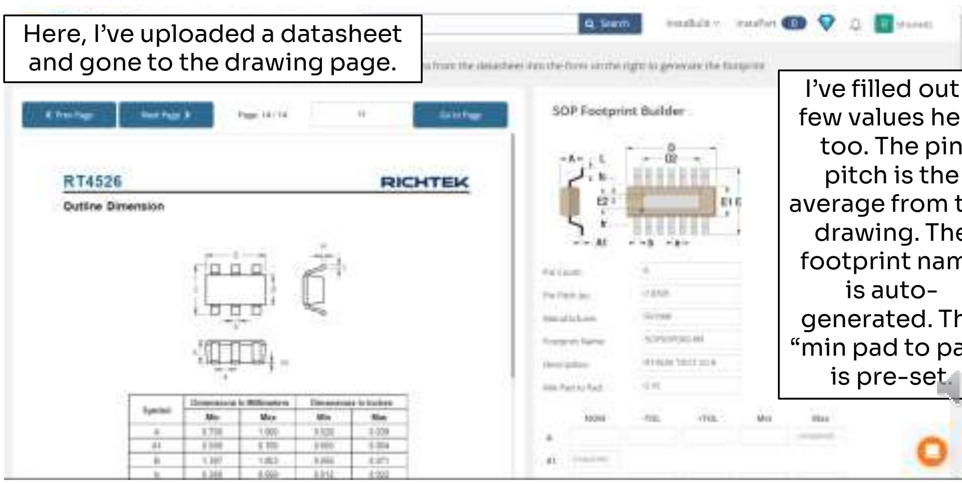


The datasheet pane is a nice touch.

Each of these calculators is very similar, at least conceptually, to the KiCAD wizards.

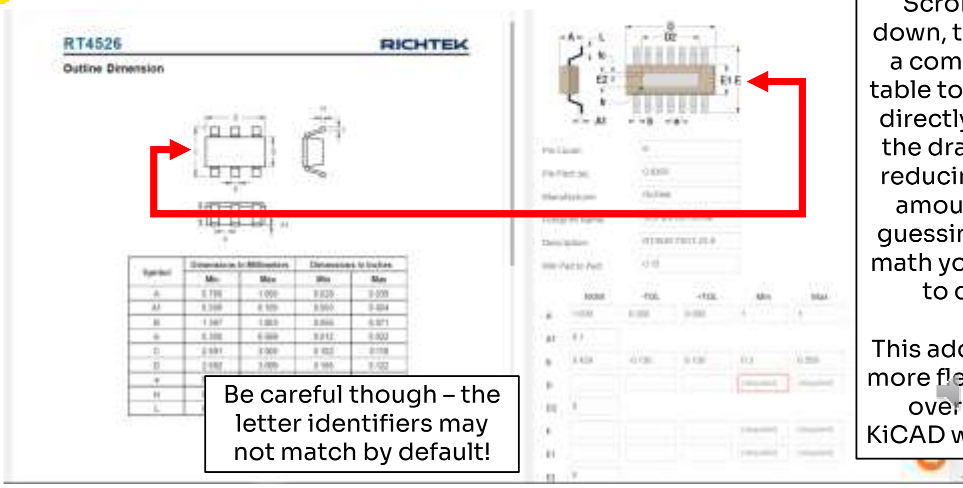
SnapMagic Footprint Generator

Here, I've uploaded a datasheet and gone to the drawing page.



I've filled out a few values here too. The pin pitch is the average from the drawing. The footprint name is auto-generated. The "min pad to pad" is pre-set.

SnapMagic Footprint Generator

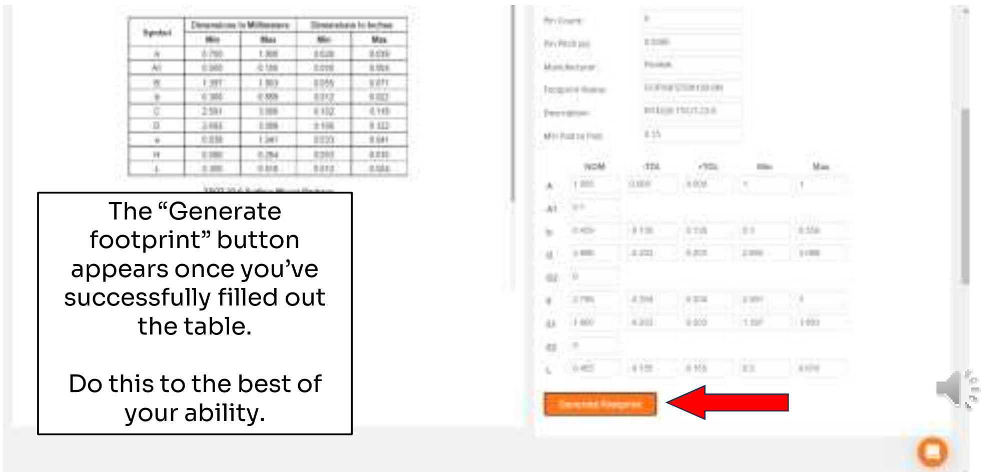


Scrolling down, there is a complete table to fill out directly from the drawing, reducing the amount of guessing and math you have to do.

This adds a lot more flexibility over the KiCAD wizards.

Be careful though – the letter identifiers may not match by default!

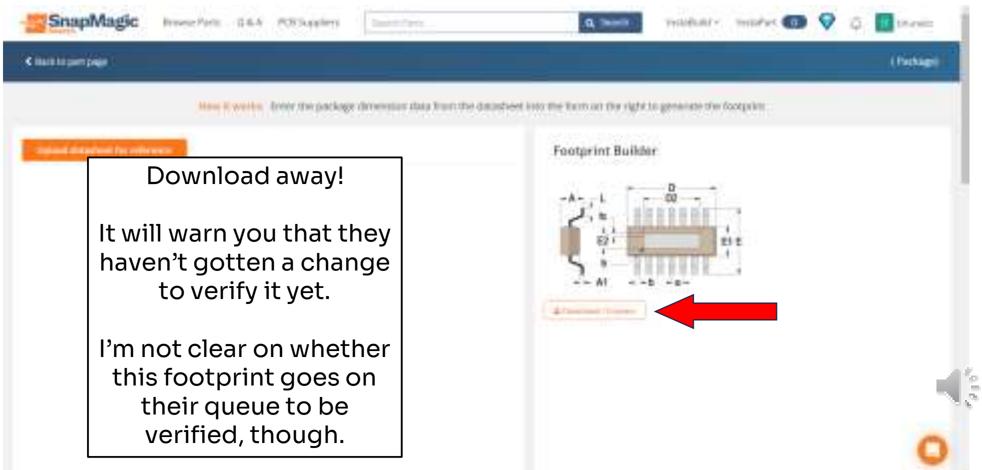
SnapMagic Footprint Generator



The "Generate footprint" button appears once you've successfully filled out the table.

Do this to the best of your ability.

SnapMagic Footprint Generator



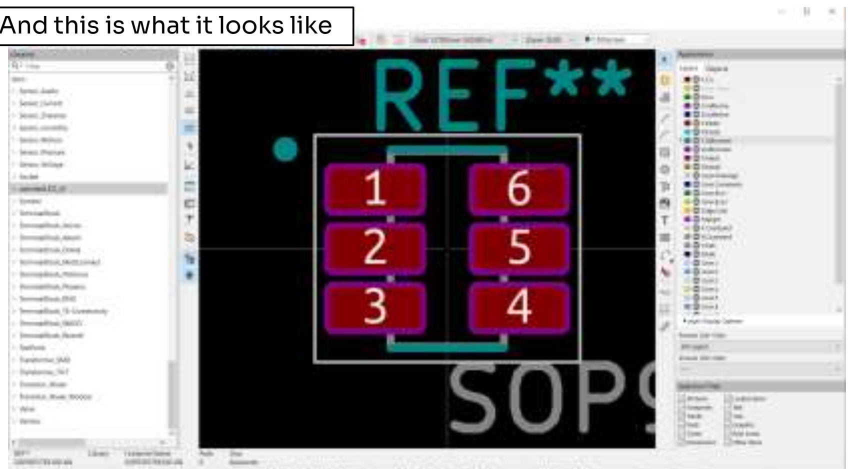
Download away!

It will warn you that they haven’t gotten a change to verify it yet.

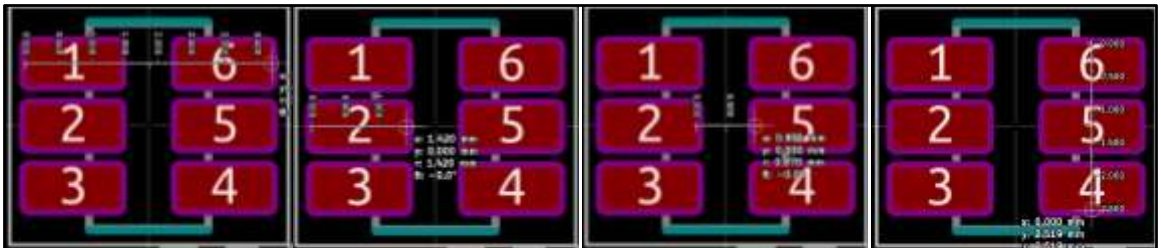
I’m not clear on whether this footprint goes on their queue to be verified, though.

SnapMagic Footprint Generator

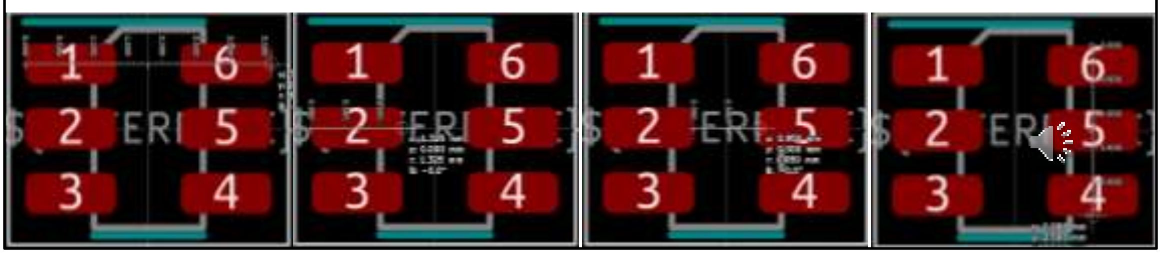
And this is what it looks like



The pink is the mask layer, which is used for defining when soldermask openings are on the board. The other parts have those polygons too, but they're just the same size as the copper pad, which takes visual priority.



	Model Width	Pad Width	Int. Gap	Pad-Col Height
SnapMagic	3.71	1.42	0.87	2.519
KiCAD	3.6	1.325	0.95	2.5



Here’s a comparison between some dimensions between the SnapMagic model, on top, and the KiCAD model, on the bottom. It’s close, but they’re not not quite the same. Will one not work? We’d have to put them on a board and print a 1-to-1 copy of the board to find out. But the differences are small, so hopefully.

Either way, this is a great example of how error-prone and variable the footprint generation process can be! Especially given the variation between datasheets. It’s critical to confirm your parts fit before sending your designs off for fab.

End of Part 7C

And with that, we conclude part 7C, and, as of spring 2024, the end of the PCB Design with KiCAD tutorial series.

If you have further design questions and you're on the Georgia Tech campus, feel free to stop by The Hive during open hours, normally during the semester from 11-6. There is usually a PI, MPI, or staff member available to help you, even if they don't know KiCAD. Design questions transcend software choices.

If you're not on campus, the internet is your friend here. For KICAD-specific questions, there are probably hundred of tutorials on KiCAD, especially the basics we've covered in this series, the forums are quite active, and the documentation is pretty good (though not complete). For design-specific questions, well, you'll probably be doing a lot of trial and error. Make some small boards to demo the concepts you're working on, read a lot, and decide how much error is tolerable to your system.

Design is a never-ending topic, and there is always more to learn.

Thanks for watching, and good luck!