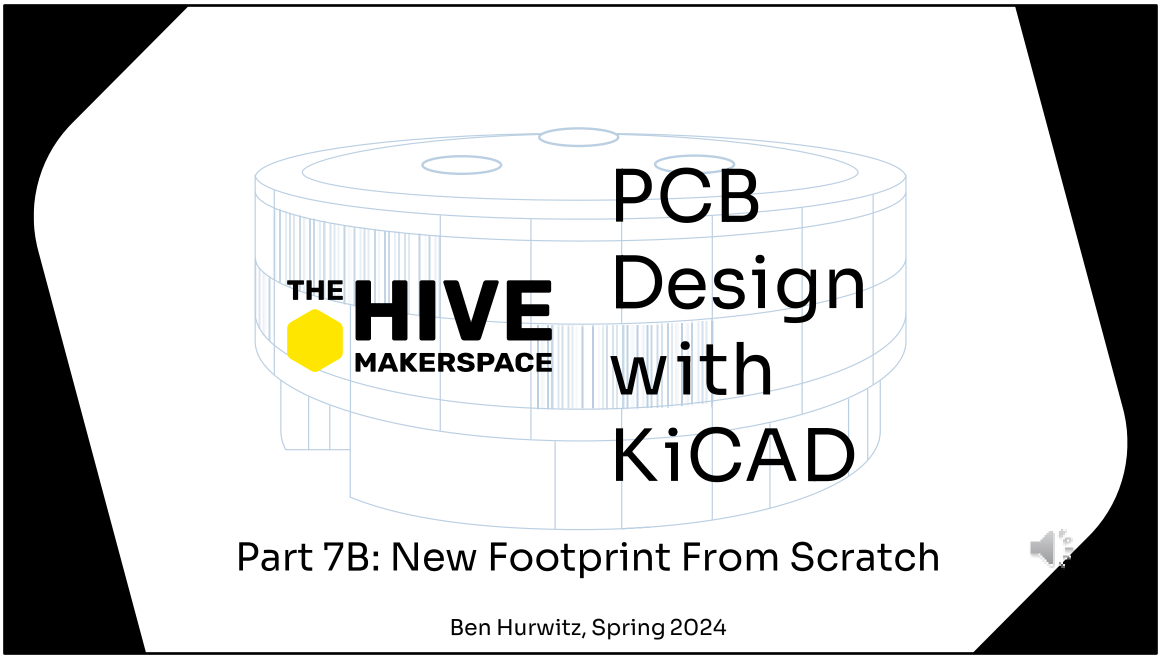


Hi all, welcome to The Hive's series on PCB Design with KiCAD. My name is Ben, and in this series, we've been walking through the PCB design process using KiCAD as our electronics design software.

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.

The last video, part 7A, we made a new project-specific footprint library, and populated it with a bunch of globally-available footprints that we needed.

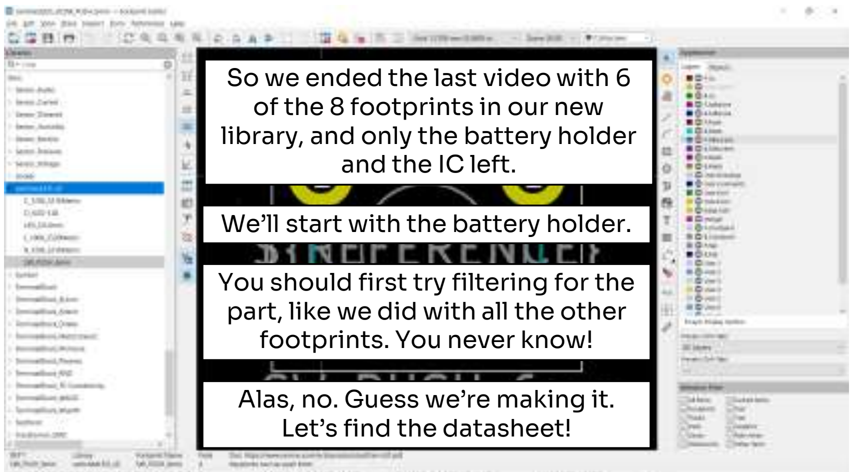
In this video, we'll look at generating a custom footprint from scratch, why you should never do it again, and then how to import an online model.

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 other than that you watched part 7A. So I apologize if some of this is repetition for some of you.

Let's get started.

Footprint Library



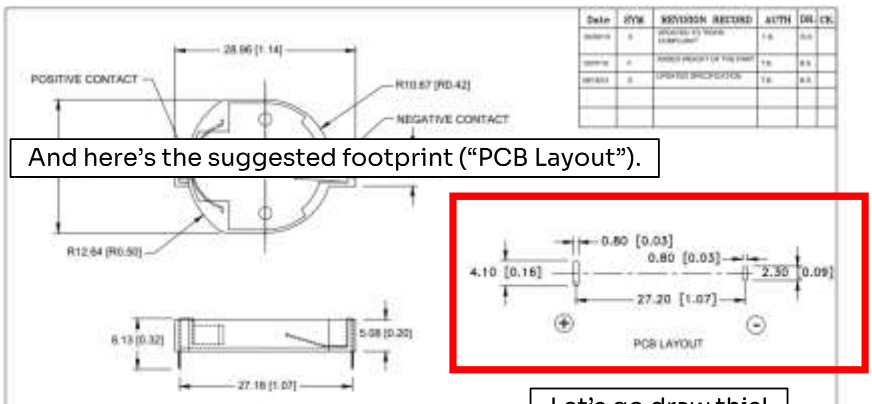
Footprint Library

I had to hunt down the technical drawing for the battery holder on the manufacturer's website because the datasheet link on Digikey gave me their catalogue.

Not helpful.

Footprint Library

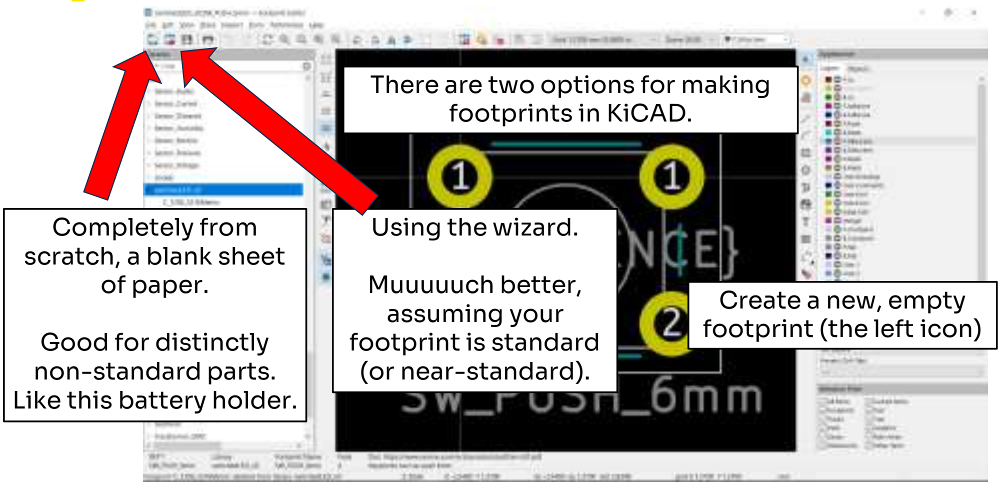
Here’s the technical drawing (the top half, at least).



And here’s the suggested footprint (“PCB Layout”).

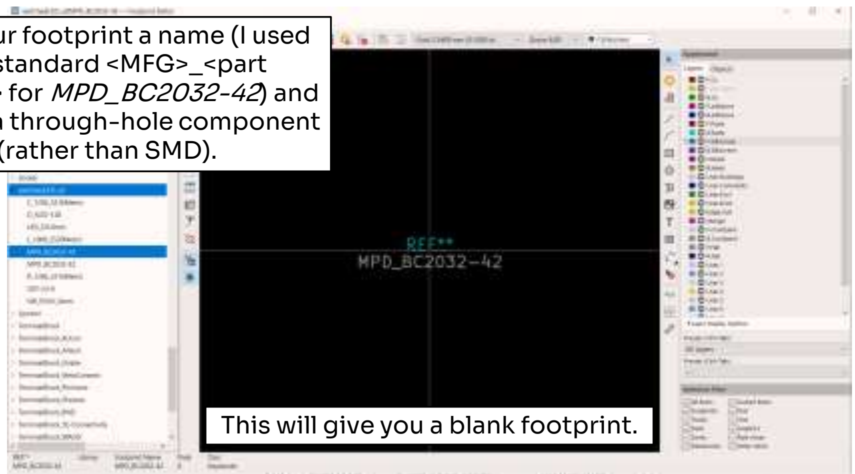
Let’s go draw this!

Footprint Library



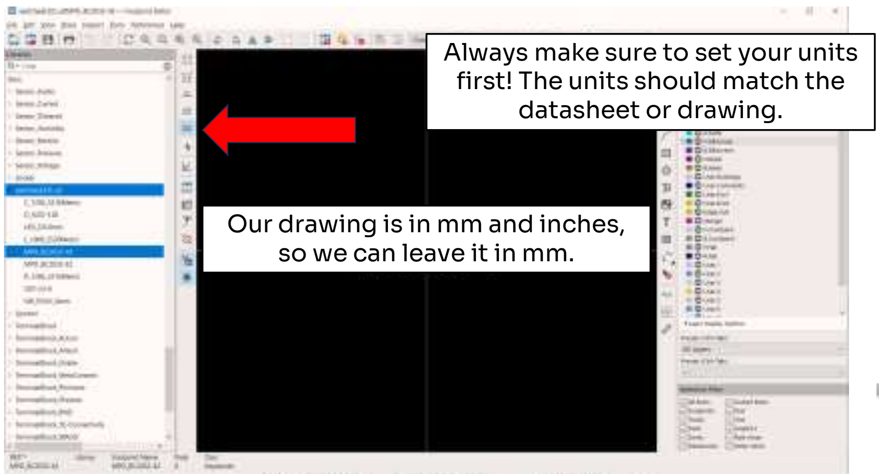
Footprint Library

Give your footprint a name (I used the standard <MFG>_<part number> for *MPD_BC2032-42*) and set it to a through-hole component (rather than SMD).



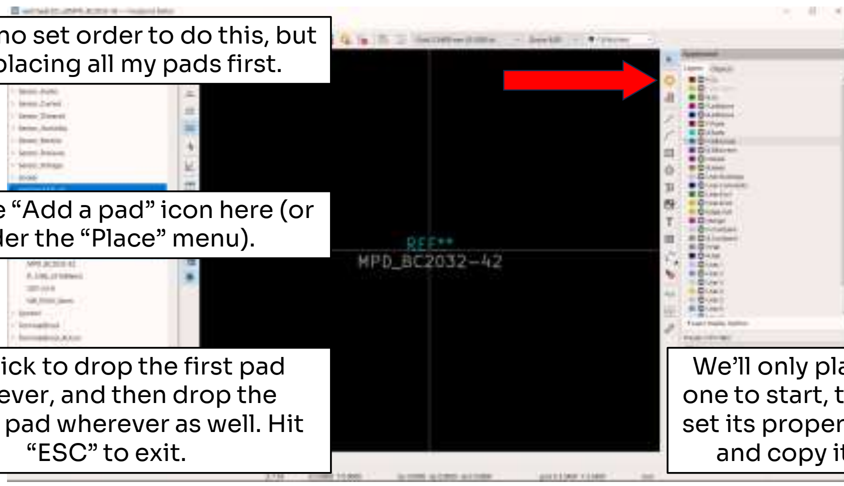
This will give you a blank footprint.

Footprint Library



Footprint Library

There is no set order to do this, but I like placing all my pads first.

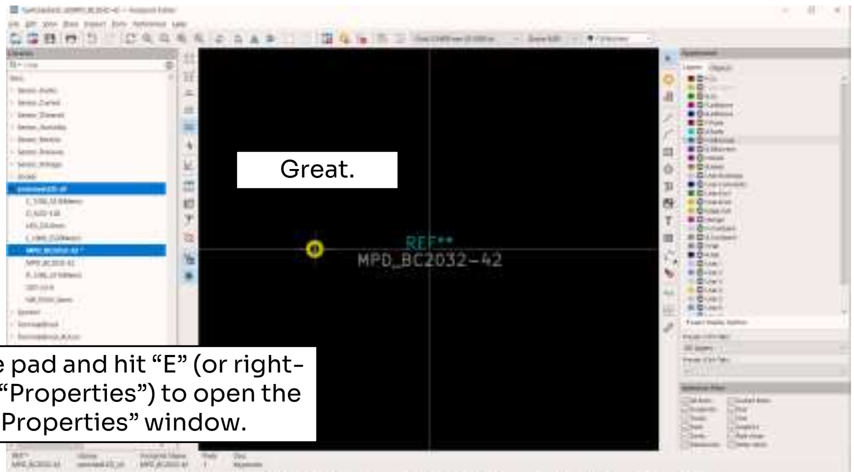


Click the "Add a pad" icon here (or under the "Place" menu).

Left click to drop the first pad wherever, and then drop the second pad wherever as well. Hit "ESC" to exit.

We'll only place one to start, then set its properties and copy it.

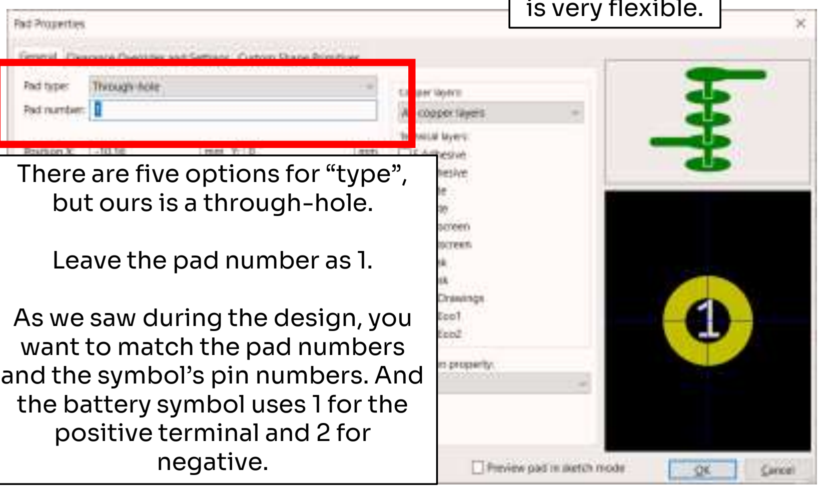
Footprint Library



Select the pad and hit "E" (or right-click and "Properties") to open the "Pad Properties" window.

Footprint Library

Pad generation is very flexible.



There are five options for “type”, but ours is a through-hole.

Leave the pad number as 1.

As we saw during the design, you want to match the pad numbers and the symbol’s pin numbers. And the battery symbol uses 1 for the positive terminal and 2 for negative.

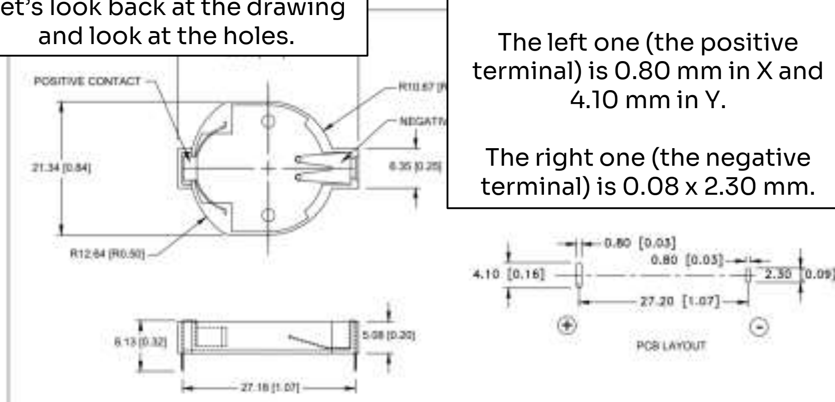
Footprint Library

Let's look back at the drawing and look at the holes.

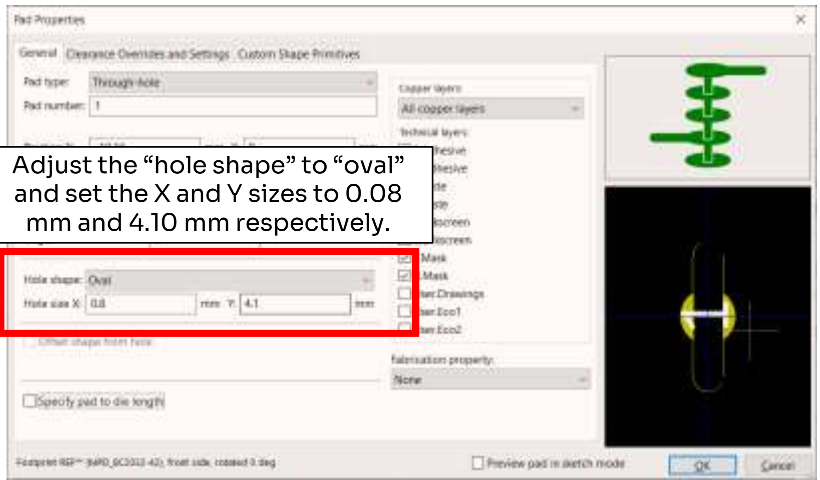
These are slots, so not circular. And they're not the same!

The left one (the positive terminal) is 0.80 mm in X and 4.10 mm in Y.

The right one (the negative terminal) is 0.08 x 2.30 mm.

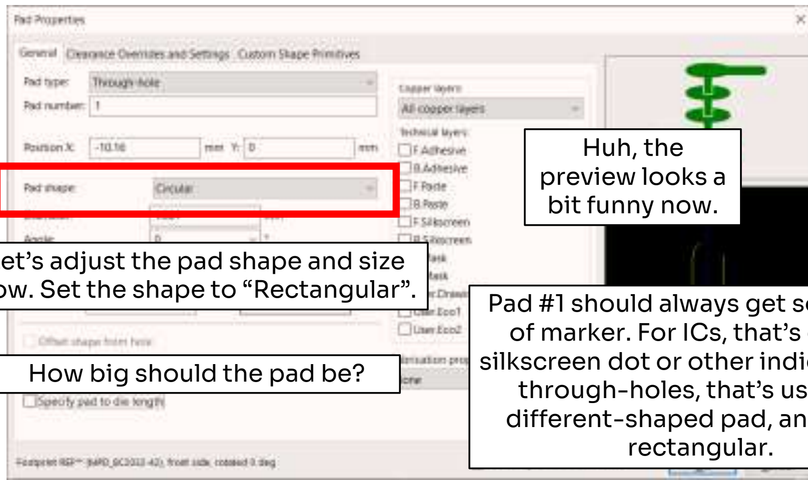


Footprint Library



Adjust the “hole shape” to “oval” and set the X and Y sizes to 0.08 mm and 4.10 mm respectively.

Footprint Library



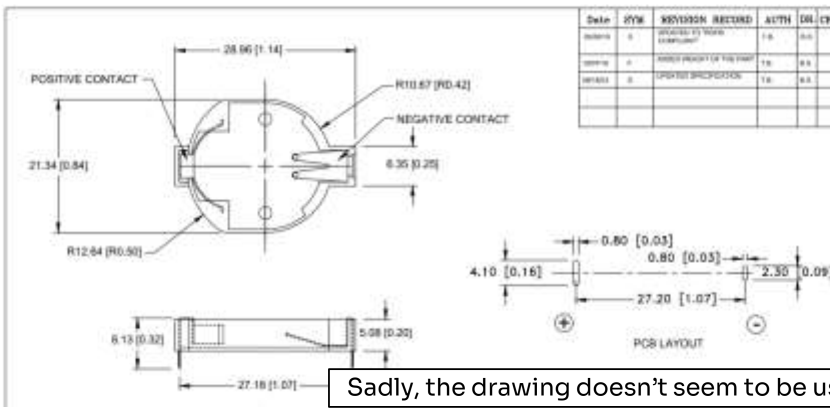
Huh, the preview looks a bit funny now.

Let's adjust the pad shape and size now. Set the shape to "Rectangular".

How big should the pad be?

Pad #1 should always get some sort of marker. For ICs, that's often a silkscreen dot or other indicator. For through-holes, that's usually a different-shaped pad, and often rectangular.

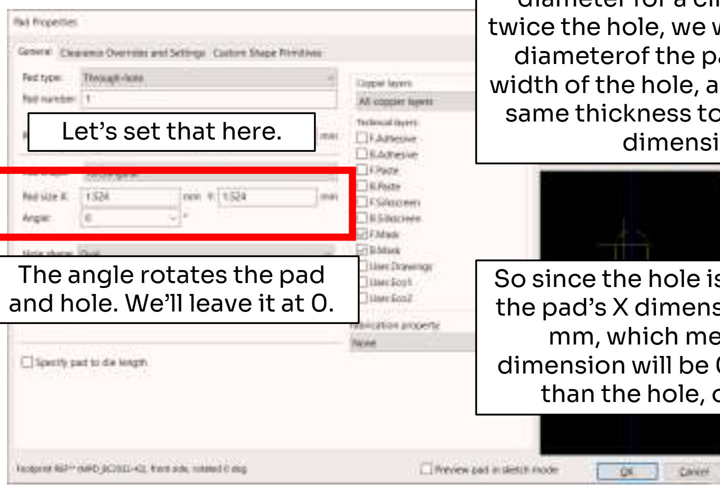
Footprint Library



Sadly, the drawing doesn't seem to be useful here.

Footprint Library

Remembering that the pad diameter for a circular pad is twice the hole, we will make the X diameterof the pad twice the width of the hole, and the use the same thickness to add to the Y dimension.



Let's set that here.

The angle rotates the pad and hole. We'll leave it at 0.

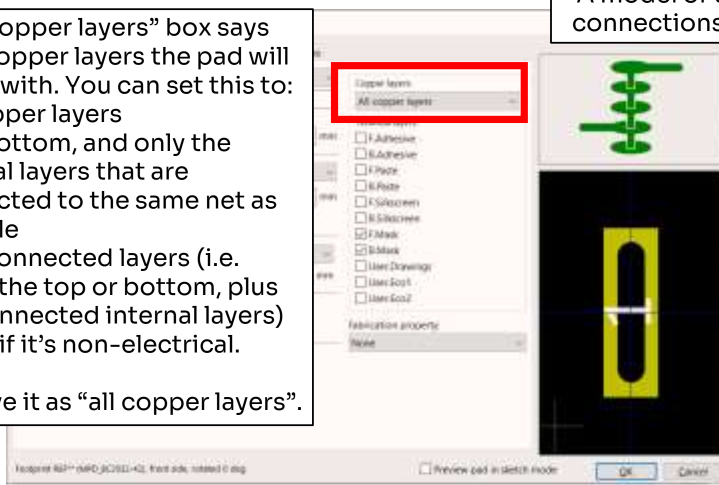
So since the hole is 0.8 x 4.1 mm, the pad's X dimension will be 1.6 mm, which means the Y dimension will be 0.8 mm larger than the hole, or 4.9 mm.

Footprint Library

The "copper layers" box says which copper layers the pad will interact with. You can set this to:

- All copper layers
- Top, bottom, and only the internal layers that are connected to the same net as the hole
- Only connected layers (i.e. either the top or bottom, plus any connected internal layers)
- None, if it's non-electrical.

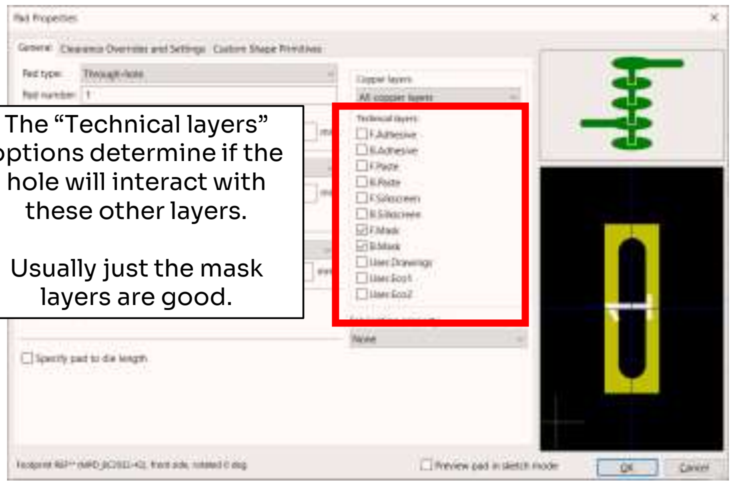
We'll leave it as "all copper layers".



A model of the hole connections is here.

Much better.

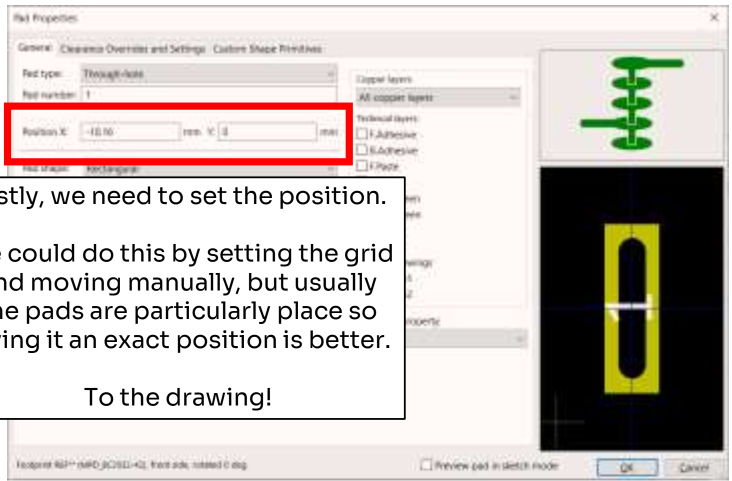
Footprint Library



The “Technical layers” options determine if the hole will interact with these other layers.

Usually just the mask layers are good.

Footprint Library



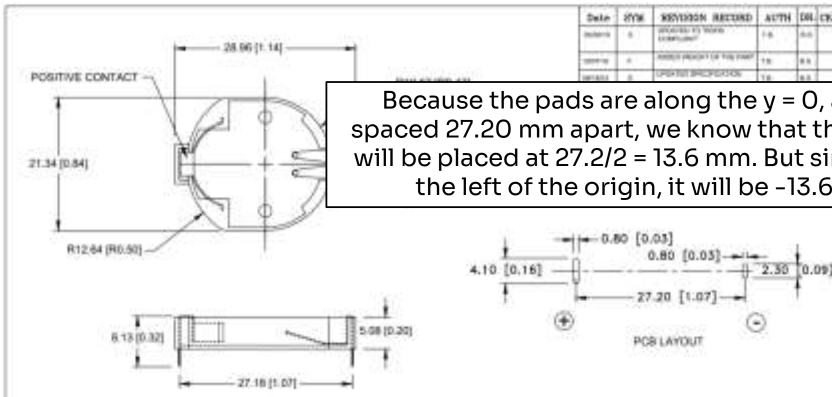
Lastly, we need to set the position.

We could do this by setting the grid and moving manually, but usually the pads are particularly place so giving it an exact position is better.

To the drawing!

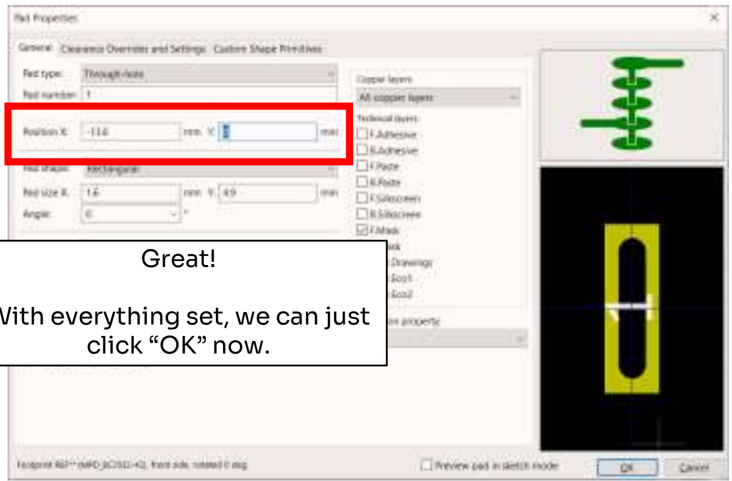
Footprint Library

It’s always a good idea to center the part at the origin.



Because the pads are along the y = 0, and are spaced 27.20 mm apart, we know that the left pad will be placed at 27.2/2 = 13.6 mm. But since it’s to the left of the origin, it will be -13.6mm.

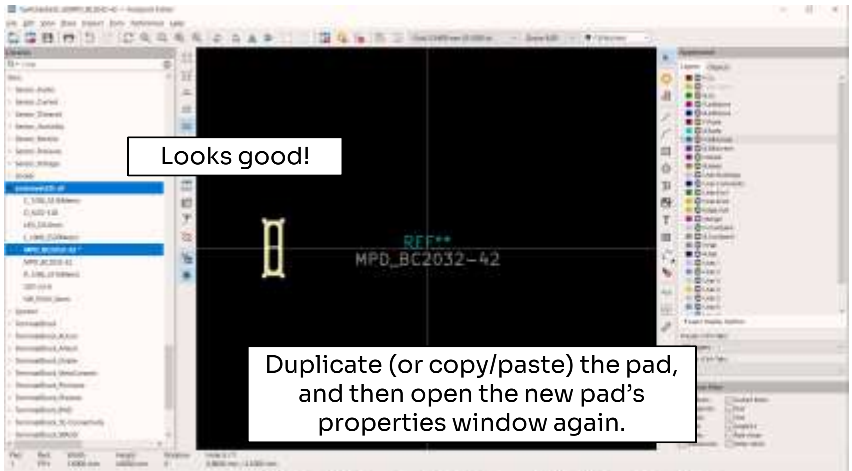
Footprint Library



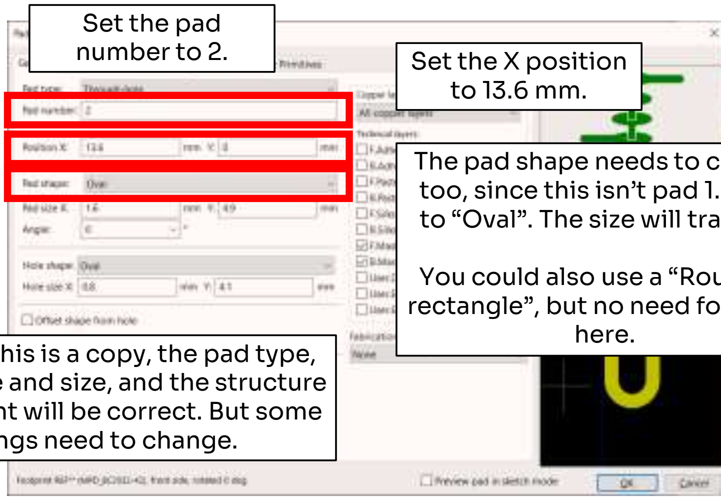
Great!

With everything set, we can just click “OK” now.

Footprint Library



Footprint Library



Set the pad number to 2.

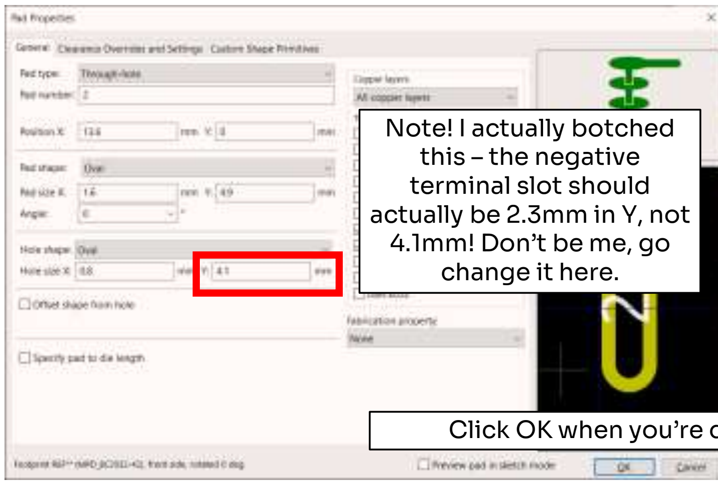
Set the X position to 13.6 mm.

The pad shape needs to change too, since this isn't pad 1. Set it to "Oval". The size will transfer.

You could also use a "Rounded rectangle", but no need for fancy here.

Because this is a copy, the pad type, hole shape and size, and the structure on the right will be correct. But some things need to change.

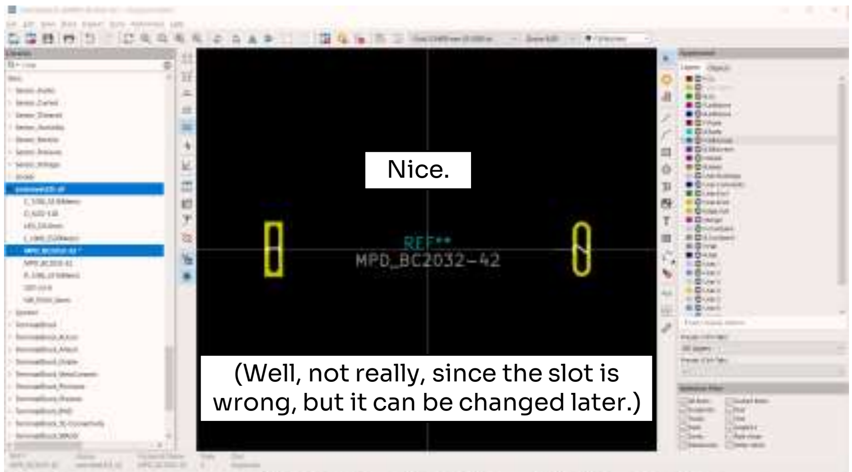
Footprint Library



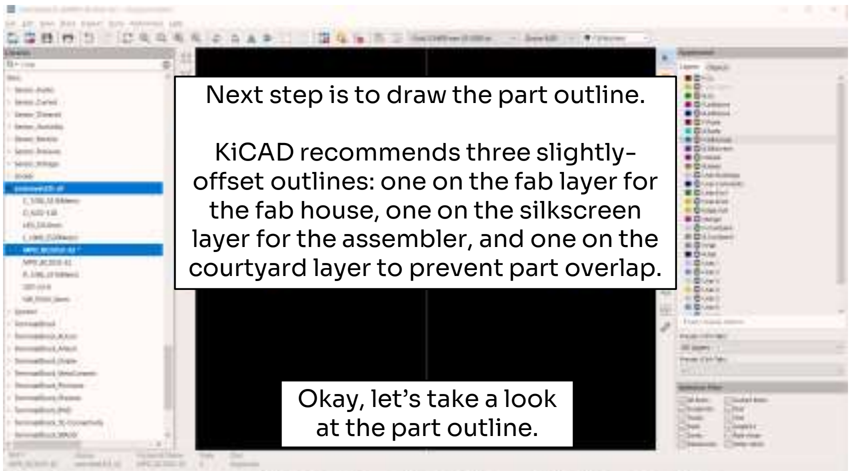
Note! I actually botched this – the negative terminal slot should actually be 2.3mm in Y, not 4.1mm! Don't be me, go change it here.

Click OK when you're done.

Footprint Library

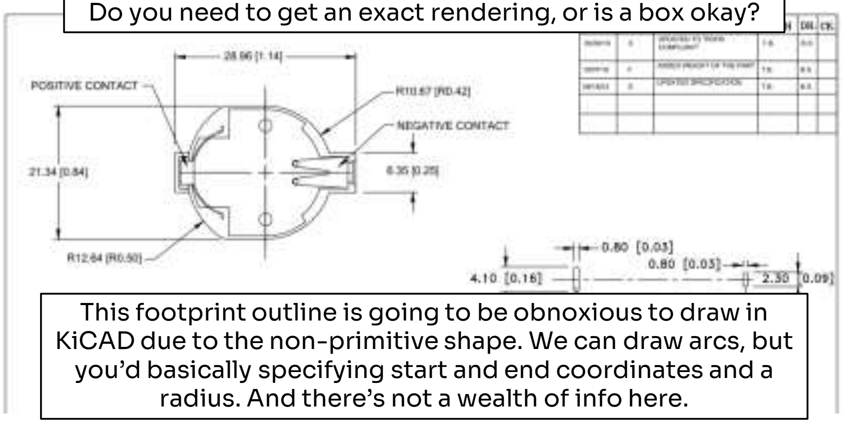


Footprint Library



Footprint Library

How accurately you draw the outline is sortof up to you. Do you need to get an exact rendering, or is a box okay?



This footprint outline is going to be obnoxious to draw in KiCAD due to the non-primitive shape. We can draw arcs, but you'd basically specifying start and end coordinates and a radius. And there's not a wealth of info here.

Footprint Library

I didn't bother with the exactly outline because, frankly, that would've been a pain and I don't care that much. So I made three circles.

The fab-layer circle approximates the part outline with a circle of diameter equal to the maximum X-dimension of 28.96 mm.

You don't want to overlap graphics because it's really hard to see them, so KiCAD recommends placing the silkscreen outline, which will be printed, immediately outside the fab outline. The circle is sized to be larger than the fab circle by half the linewidth of the fab layer plus half the linewidth of the silkscreen layer because the sizing is measured from the middle of a line. The linewidth can be found by looking at the properties of the circles drawn on those layers.

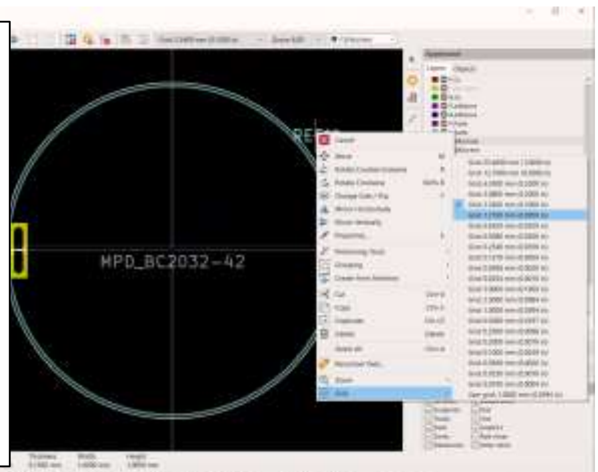
Lastly, the courtyard outline should be sized to have a 0.25mm clearance around the silkscreen outline, which is added to the radius.

These sizes are written in the image in purple on the User.Comments layer, so they won't show up anywhere that actually gets made.

Footprint Library

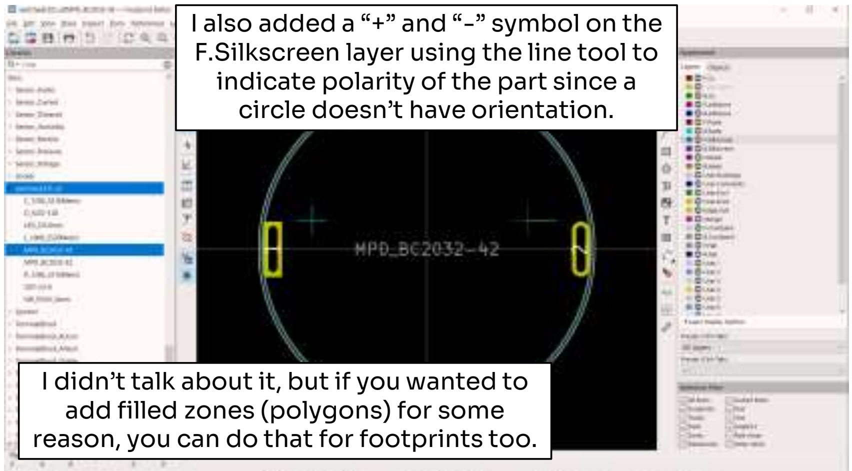
We want to also move the reference designator, REF**, outside the outline so that the assembler can see it after the part is populated.

My grid was a bit too large, so while moving the REF** text, I right-clicked to open the context menu and adjust the grid through the Grid submenu.



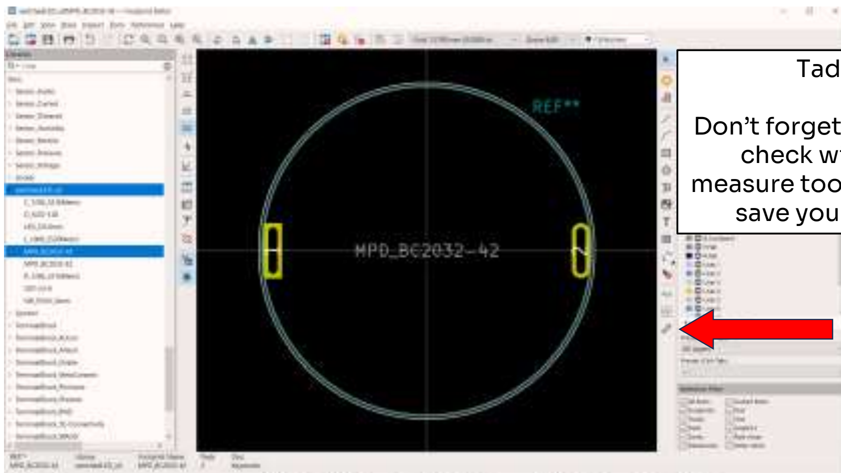
Footprint Library

I also added a "+" and "–" symbol on the F.Silkscreen layer using the line tool to indicate polarity of the part since a circle doesn't have orientation.



I didn't talk about it, but if you wanted to add filled zones (polygons) for some reason, you can do that for footprints too.

Footprint Library



Tada!

Don't forget to double check with the measure tool, and then save your work!

The measure tool icon is highlighted by the arrow.

Footprint Library

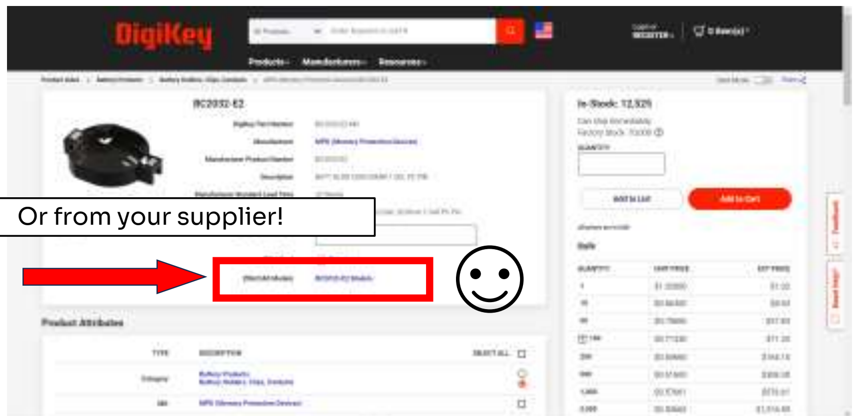
Holy moly.

So that wasn't so bad, but mostly because I opted to draw circles for the outline.

That won't always be possible, nor will datasheets always be any good, potentially making custom footprint generation nightmarish.

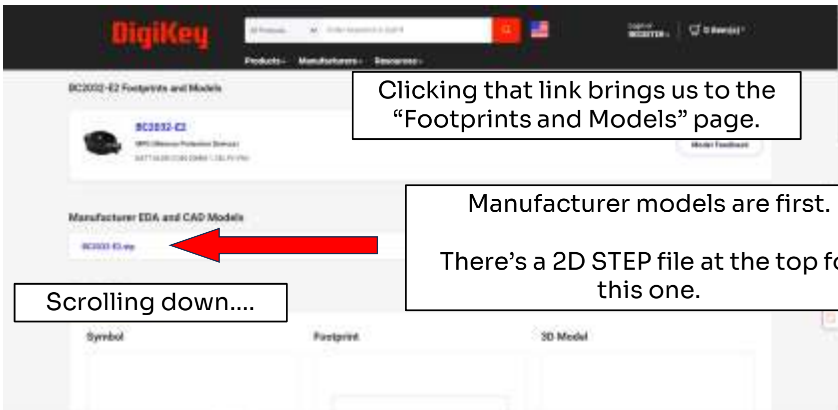
Thankfully, most of the time we don't have to do this because we can either locate or request the models from UltraLibrarian or SnapMagic.

Footprint Library



Or from your supplier!

Footprint Library



Clicking that link brings us to the “Footprints and Models” page.

Manufacturer models are first.

There’s a 2D STEP file at the top for this one.

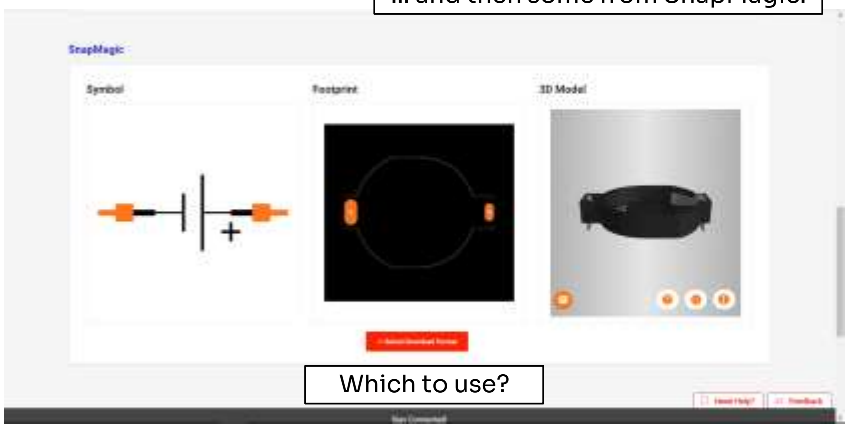
Scrolling down....

Footprint Library

… There's a set of models from UltraLibrarian…

Footprint Library

... and then some from SnapMagic.



Which to use?

Footprint Library

There is no right or wrong answer here generally. Both models are extremely likely to be sufficient.

It's on you to confirm that though!

I'm going to use SnapMagic here because I like the footprint better (and there's a 3D model).

Footprint Library

Click "Select Download Format" and select "KiCAD (V6 or later)".

(Obviously select a different format if you're using different software.)

I also selected the 3D CAD model because a 3D view of the designed PCB is useful.

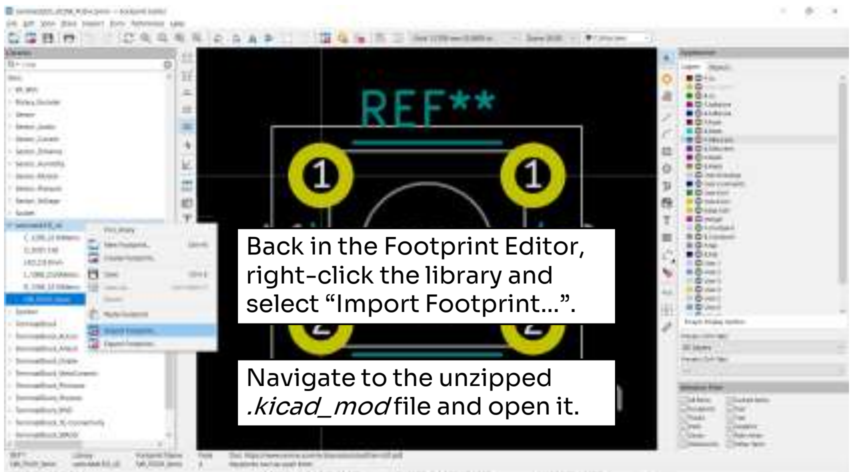
Footprint Library

Unzip the download somewhere accessible.

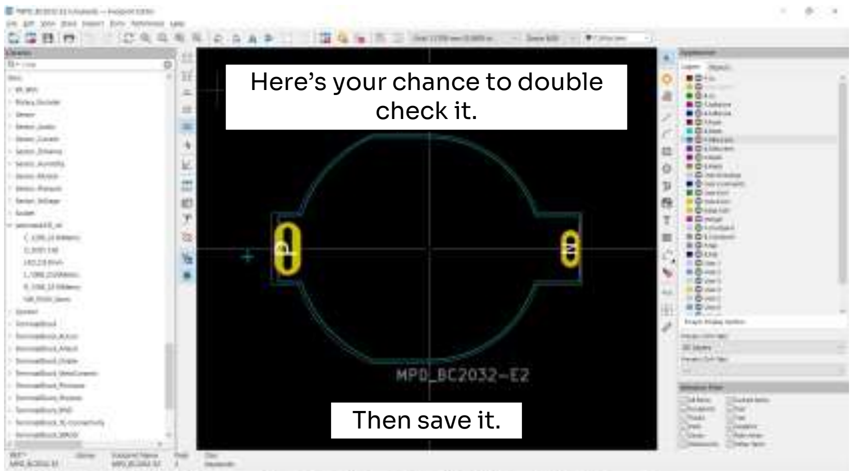
There will be five five files:

- a *.kicad_sym* symbol library
- a *.kicad_mod* footprint model library
- a *.step* 3D model
- a *.htm* link to a “How to import” webpage
- a license textfile

Footprint Library

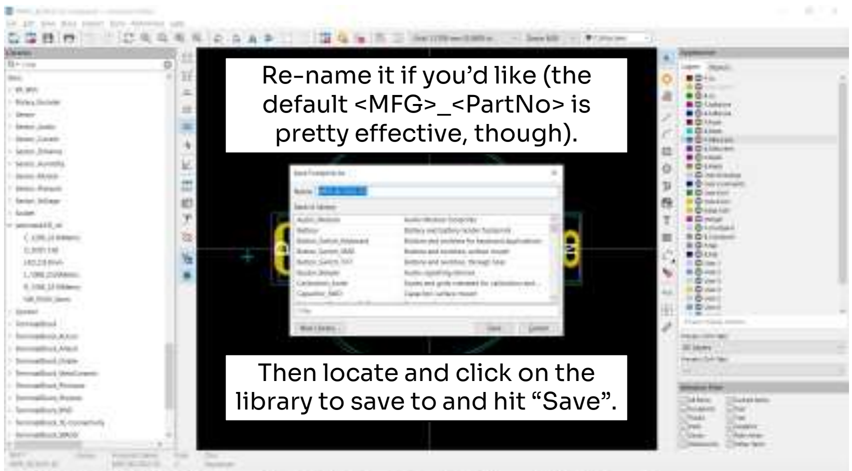


Footprint Library

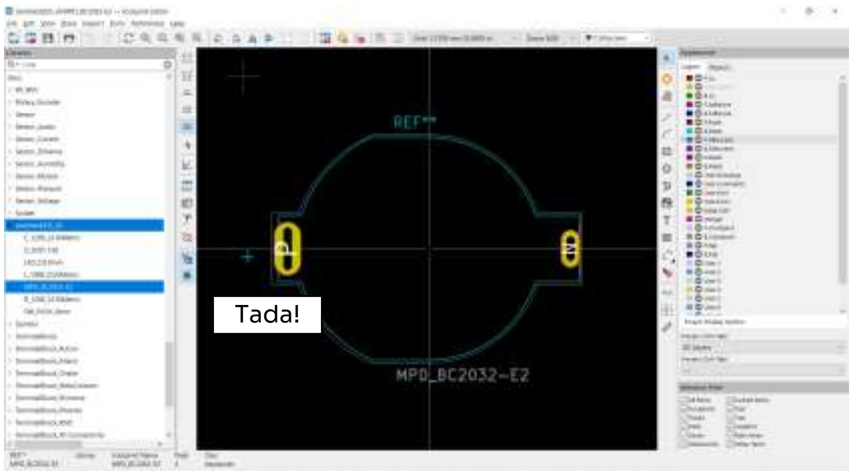


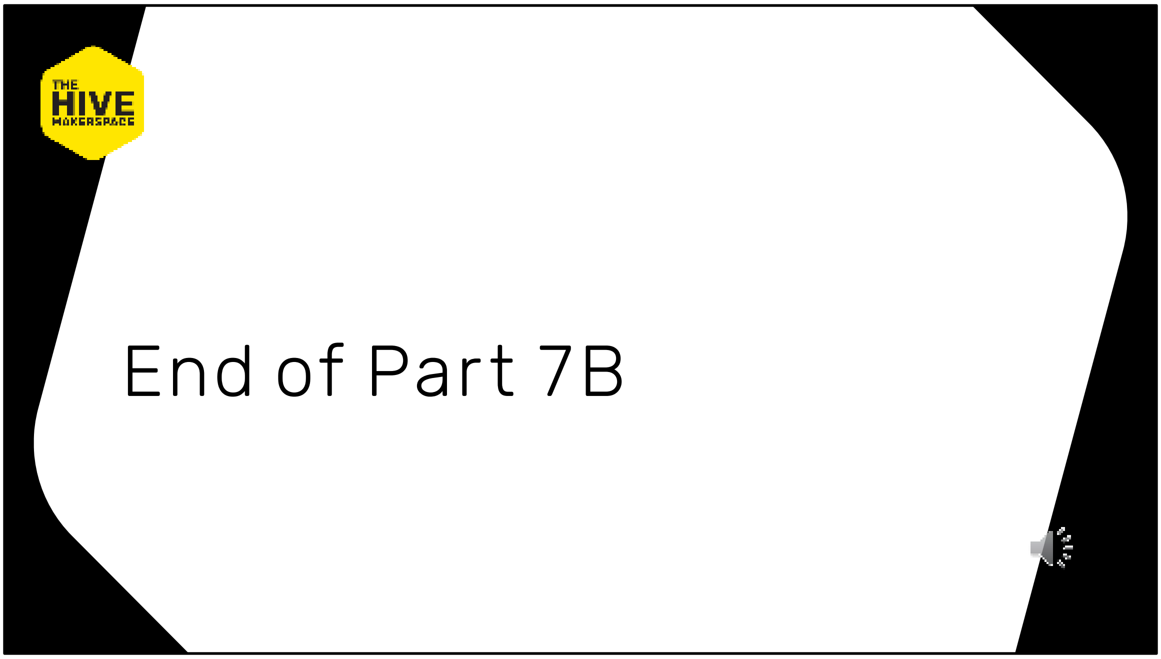
See how much nicer their looks? And notice the three outlines. Nice. They also intelligently put the positive indicator outside the part outline, meaning the user can see it after assembly.

Footprint Library



Footprint Library





And that ends part 7B and our discussion of making footprints from scratch. As you've seen, that's not the best process, and it's really (really) error-prone if you're not super (super) careful, so it's highly advised to find the model online, or request it to be made by UltraLibrarian or SnapMagic. A PDF of this video is available as well, linked in the description and hosted on The Hive's Wiki.

In the next video, we'll cover using the footprint wizard, which can make generating standard or pseudo-standard footprints easier. Or more difficult. Guess we'll find out!

See you there!