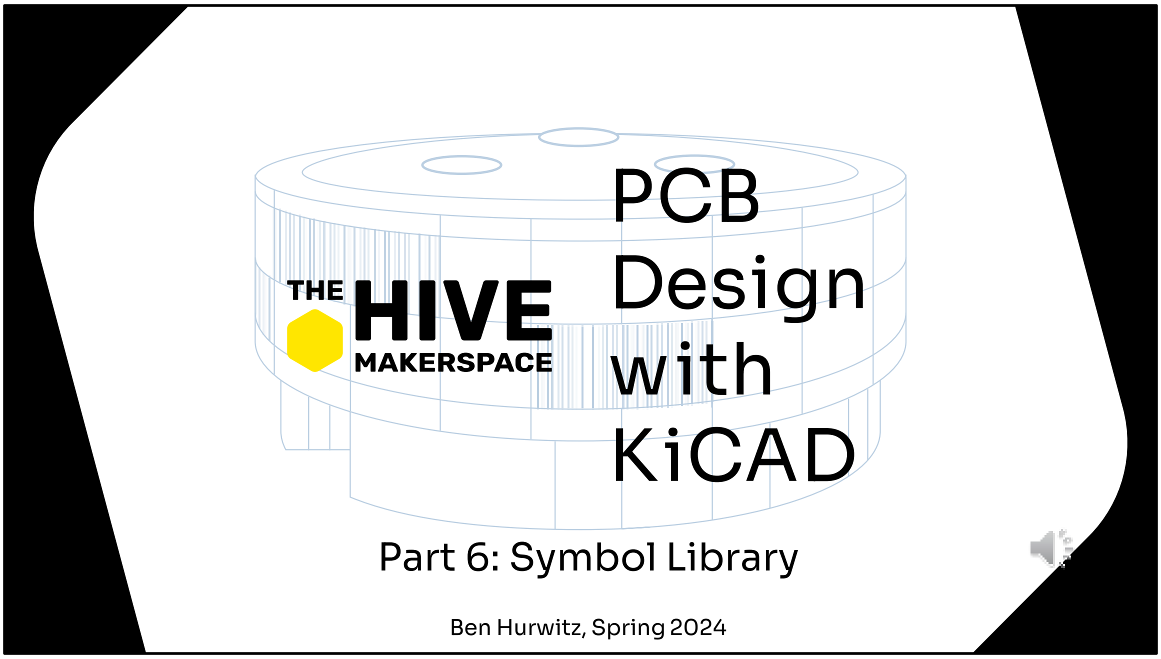


Hi all, welcome to The Hive's series on PCB Design with KiCAD. My name is Ben, and in this series, we've been walking through the PCB design process using KiCAD as our electronics design software.

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.

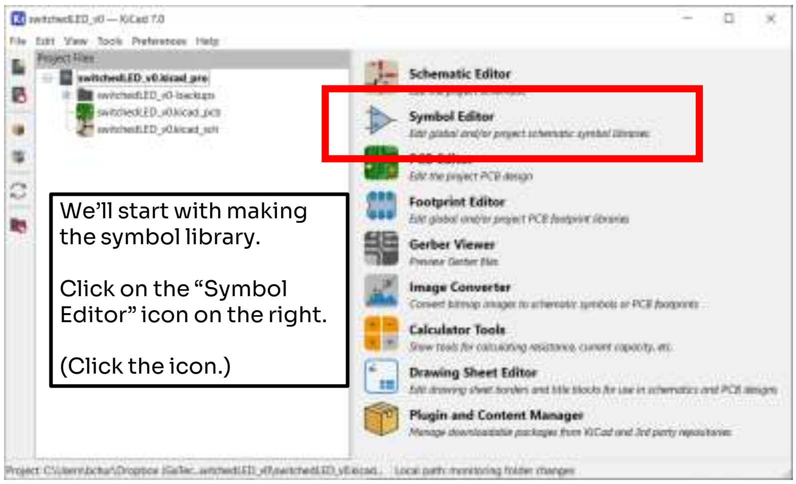
In this video, I will walk you through library selection and generating a single project-scoped symbol library to package with the rest of your project, and keep your work insulated from external changes.

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.

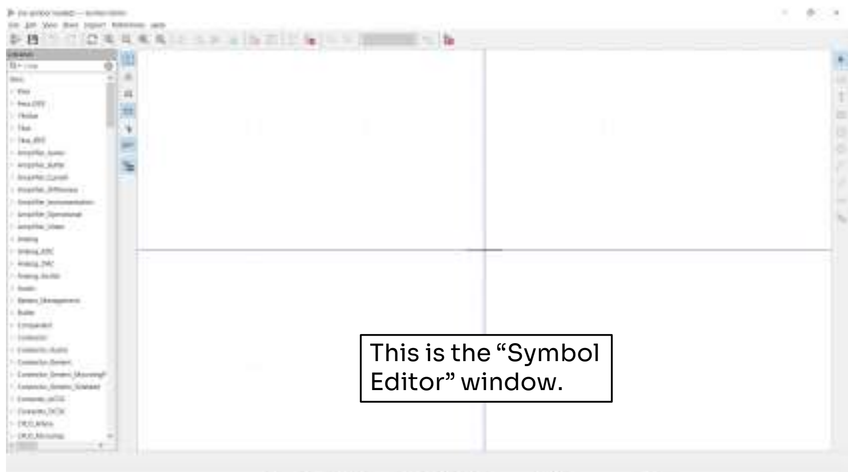
Let’s get started.

Creating the model libraries



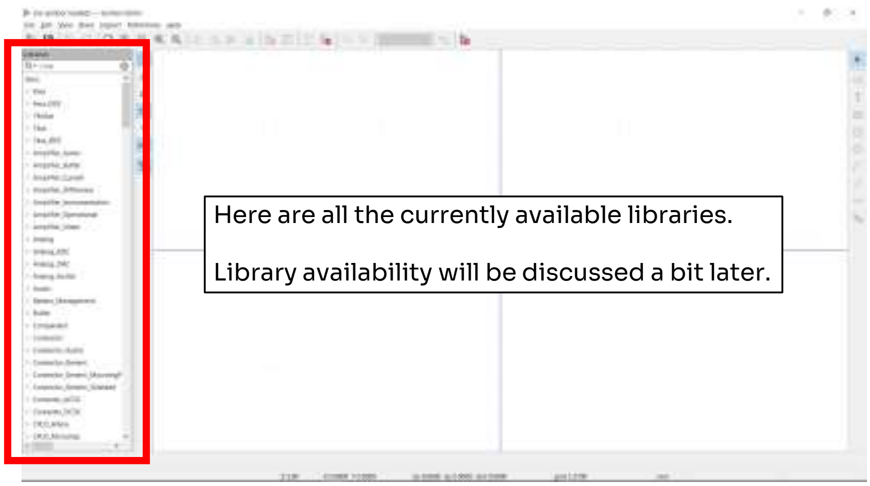
After creating the project, typically the first thing you'd want to do is create a single library where all of your components will live. This will be a living library, as in, components will be added to this library throughout the design iterations. Try not to remove any components since you never know what you'll need again. *

Symbol Library



If you’re familiar with this window, the next few slides will be a review.

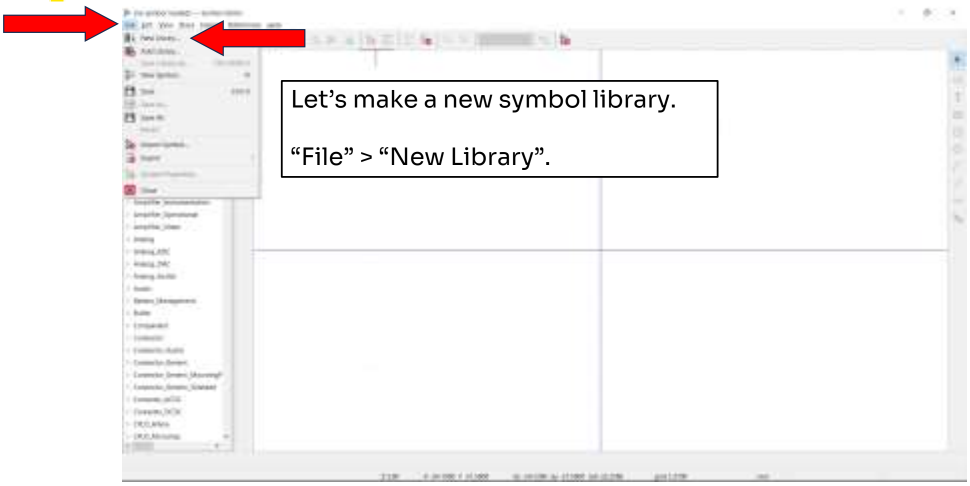
Symbol Library



Here are all the currently available libraries.

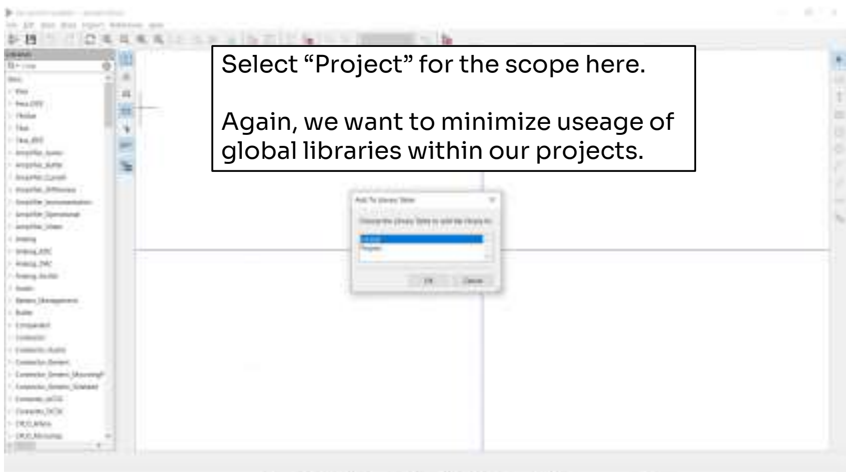
Library availability will be discussed a bit later.

Symbol Library



If you've already gone through the design process in the previous videos, don't make a new library, since we'll just be using the flashlight circuit we developed there.

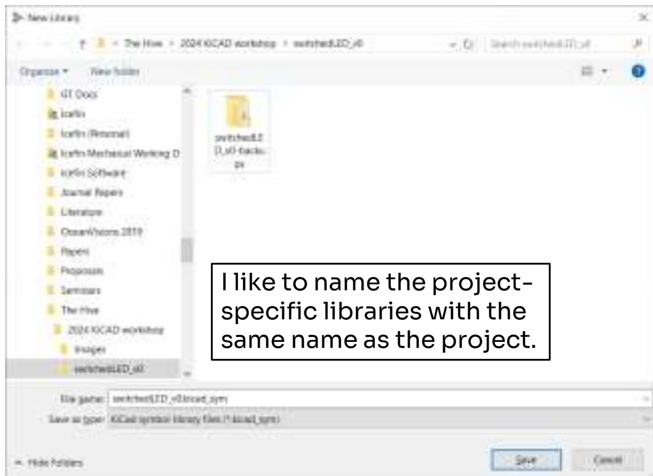
Symbol Library



Select “Project” for the scope here.

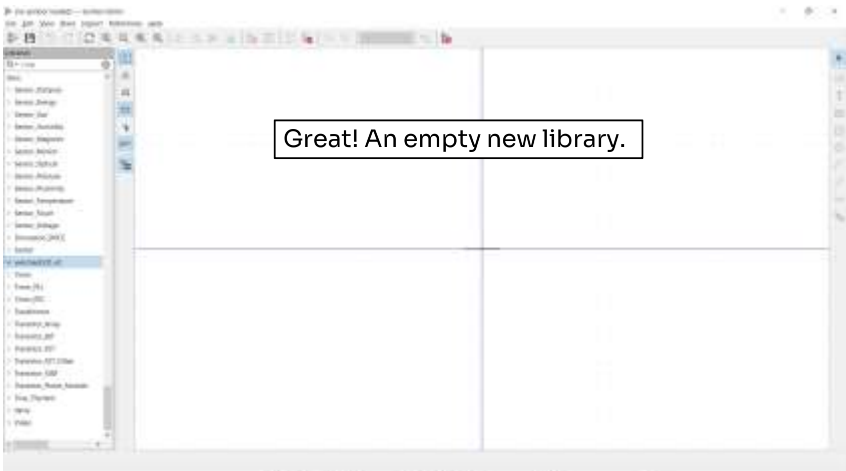
Again, we want to minimize useage of global libraries within our projects.

Symbol Library



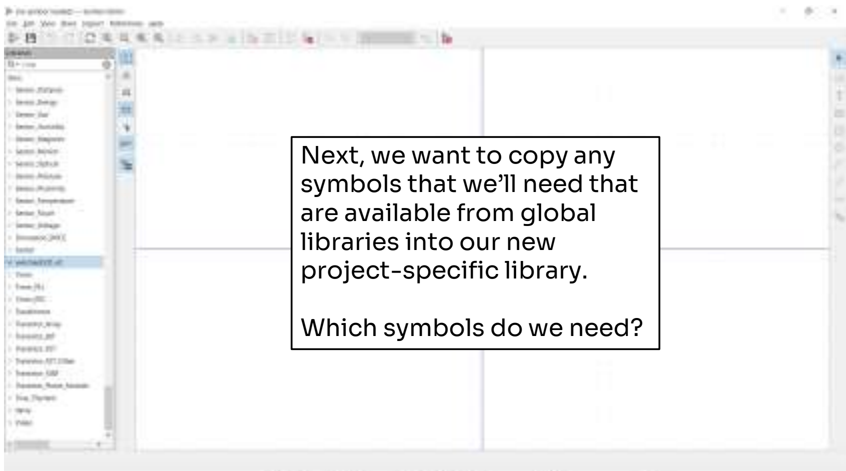
I like to name the project-specific libraries with the same name as the project.

Symbol Library



Again, if you've previously made a library for the flashlight circuit developed in videos 1-5C, move forward here with that library.

Symbol Library

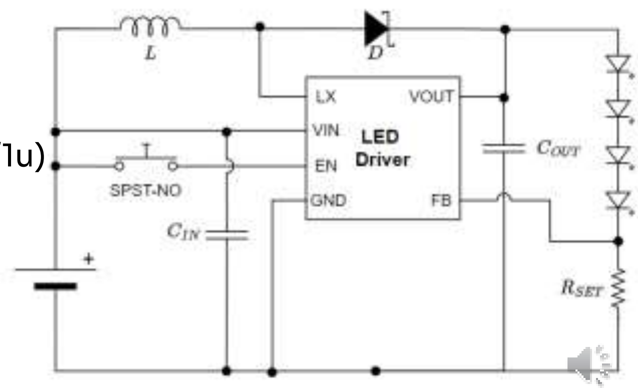


Next, we want to copy any symbols that we'll need that are available from global libraries into our new project-specific library.

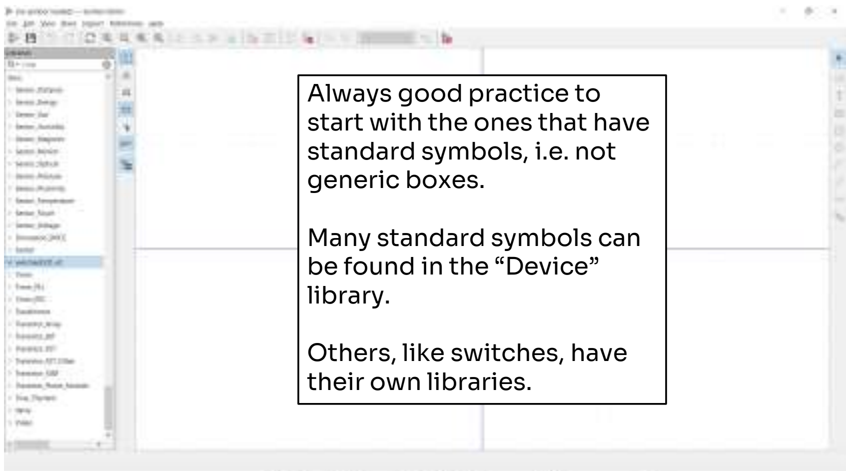
Which symbols do we need?

The flashlight circuit

- LED driver IC
- Battery + holder
- SPST-NO switch
- Cin, Cout (caps) (2.2u/1u)
- L (inductor) (22uH)
- D (Schottky diode)
- Rset (resistor) (30 Ω)
- 4x LEDs



Symbol Library

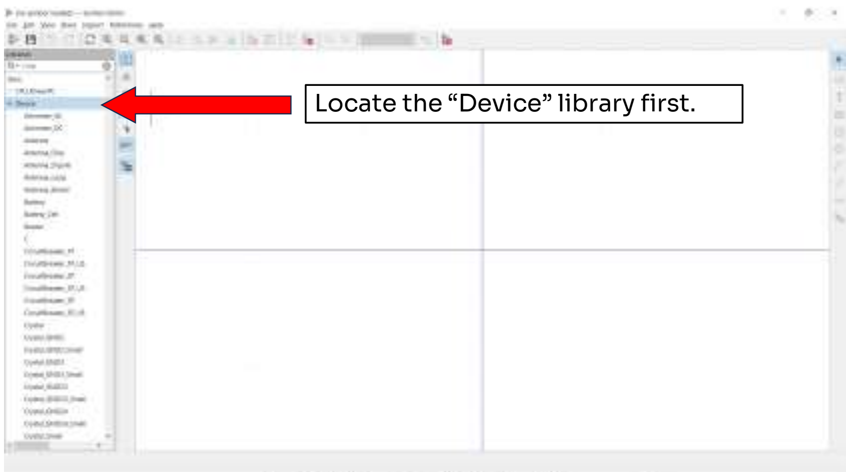


Always good practice to start with the ones that have standard symbols, i.e. not generic boxes.

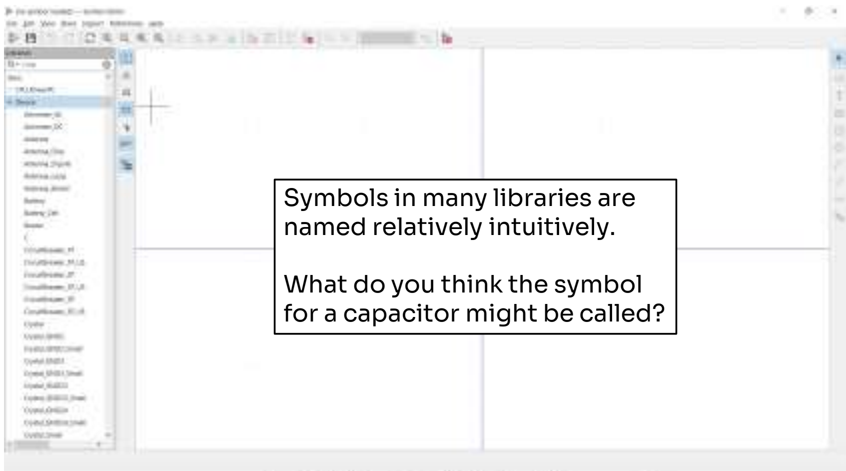
Many standard symbols can be found in the "Device" library.

Others, like switches, have their own libraries.

Symbol Library



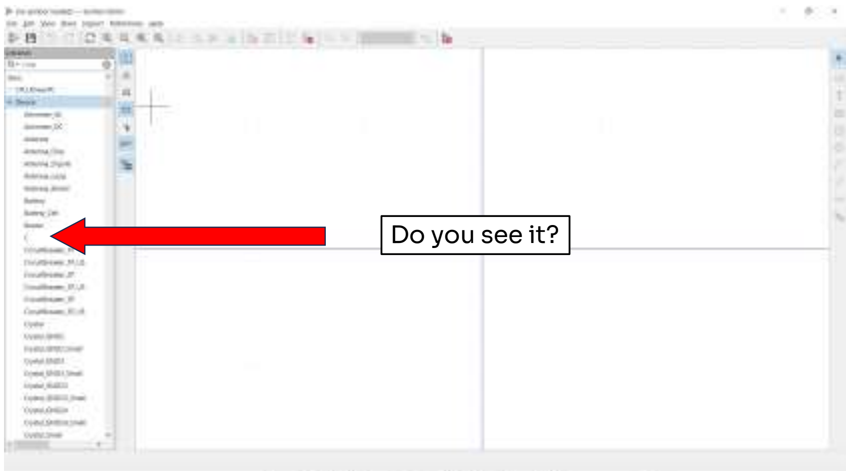
Symbol Library



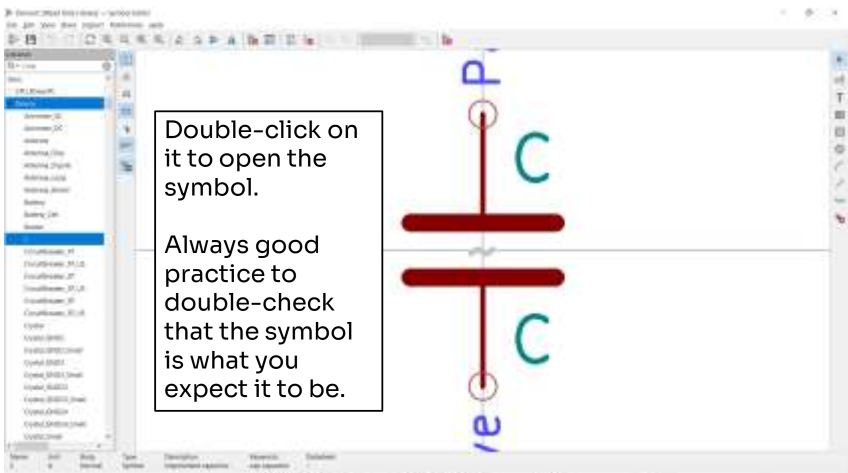
Symbols in many libraries are named relatively intuitively.

What do you think the symbol for a capacitor might be called?

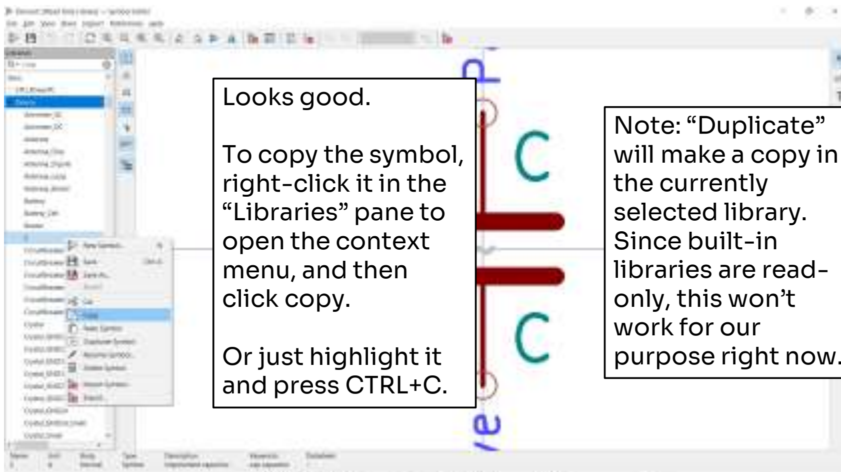
Symbol Library



Symbol Library



Symbol Library



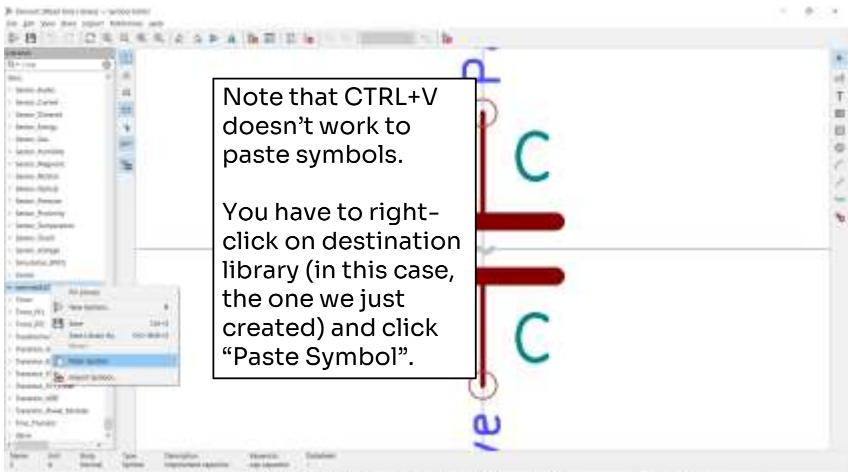
Looks good.

To copy the symbol, right-click it in the "Libraries" pane to open the context menu, and then click copy.

Or just highlight it and press CTRL+C.

Note: "Duplicate" will make a copy in the currently selected library. Since built-in libraries are read-only, this won't work for our purpose right now.

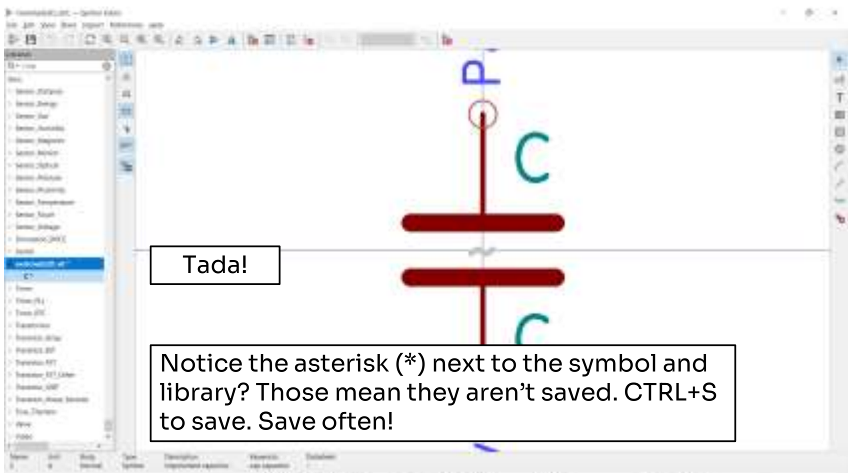
Symbol Library



Note that CTRL+V doesn't work to paste symbols.

You have to right-click on destination library (in this case, the one we just created) and click "Paste Symbol".

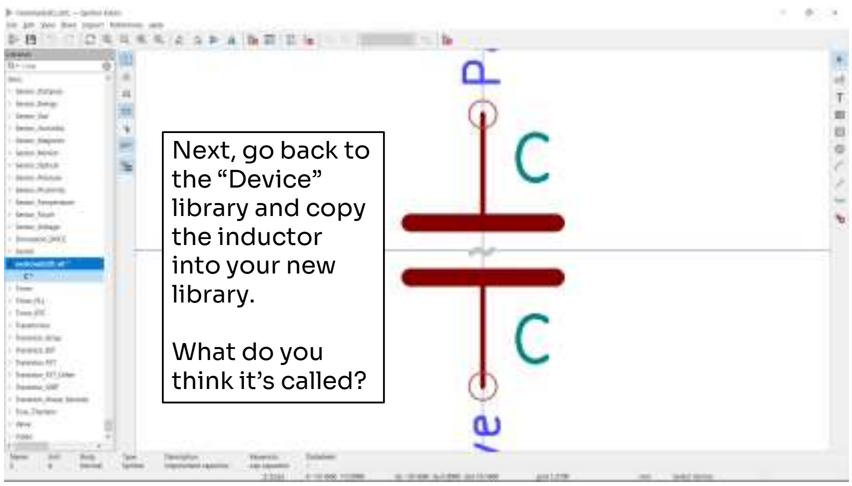
Symbol Library



Tada!

Notice the asterisk (*) next to the symbol and library? Those mean they aren't saved. CTRL+S to save. Save often!

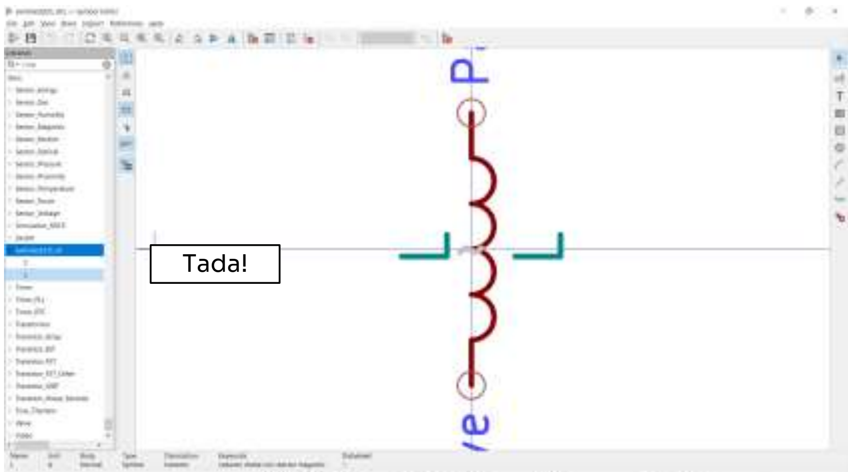
Symbol Library



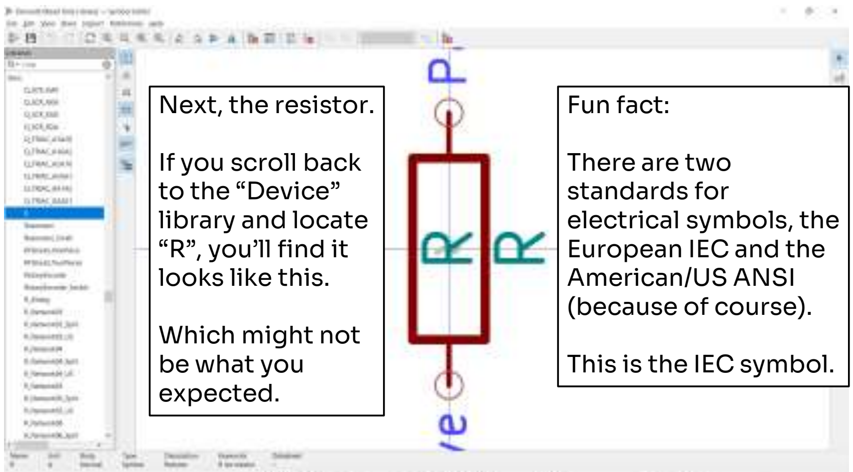
Next, go back to the “Device” library and copy the inductor into your new library.

What do you think it’s called?

Symbol Library



Symbol Library



Next, the resistor.

If you scroll back to the “Device” library and locate “R”, you’ll find it looks like this.

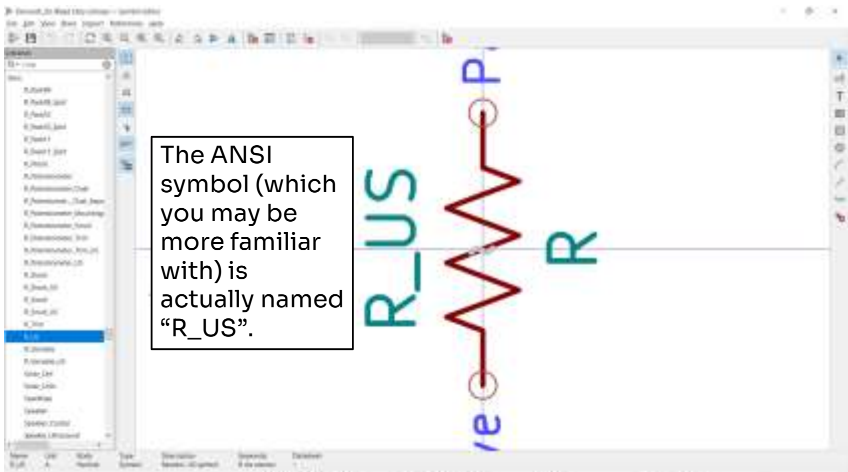
Which might not be what you expected.

Fun fact:

There are two standards for electrical symbols, the European IEC and the American/US ANSI (because of course).

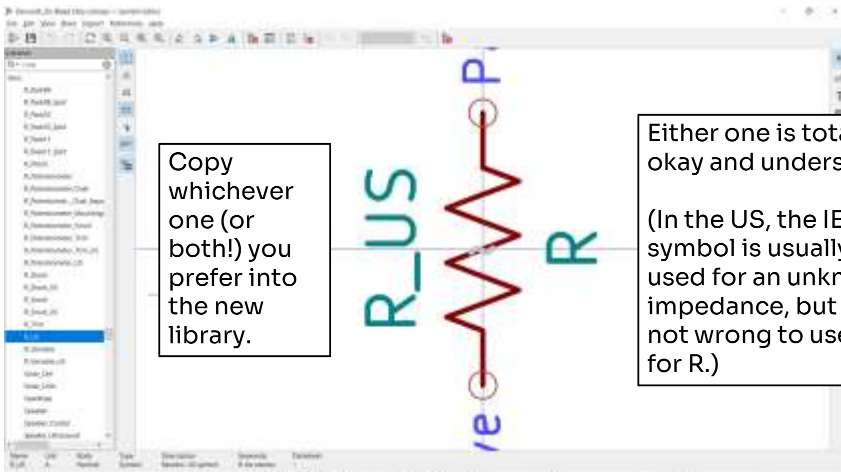
This is the IEC symbol.

Symbol Library



The ANSI symbol (which you may be more familiar with) is actually named “R_US”.

Symbol Library

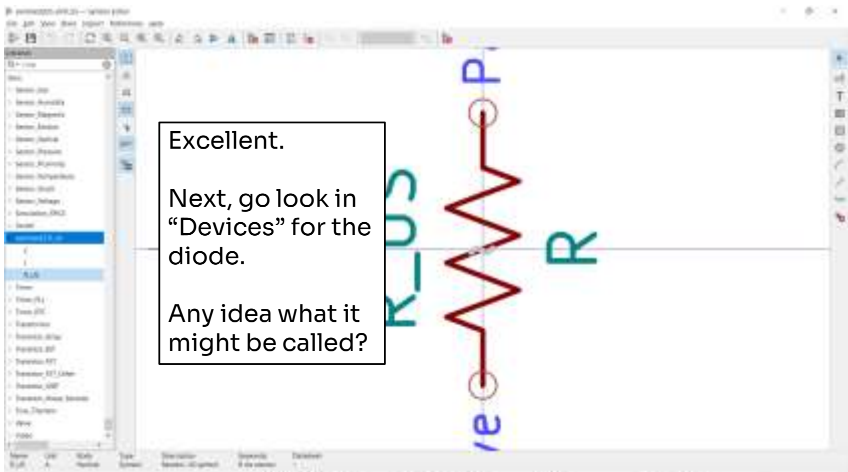


Copy whichever one (or both!) you prefer into the new library.

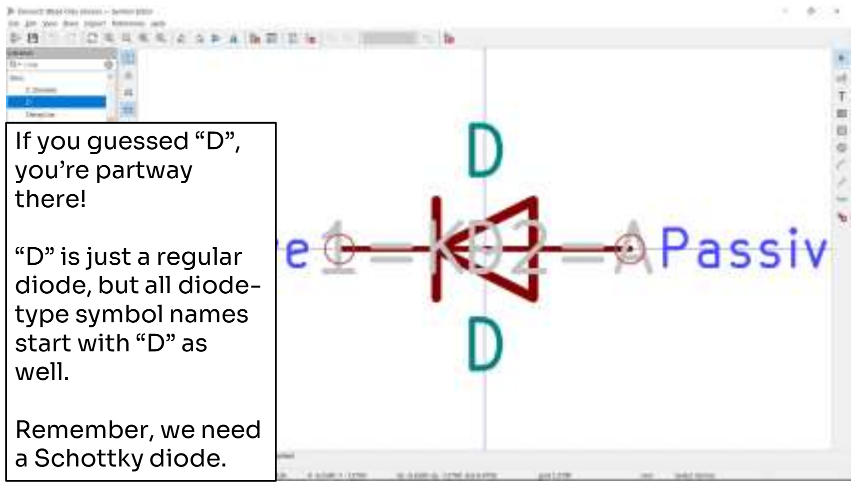
Either one is totally okay and understood.

(In the US, the IEC symbol is usually used for an unknown impedance, but it's not wrong to use it for R.)

Symbol Library



Symbol Library

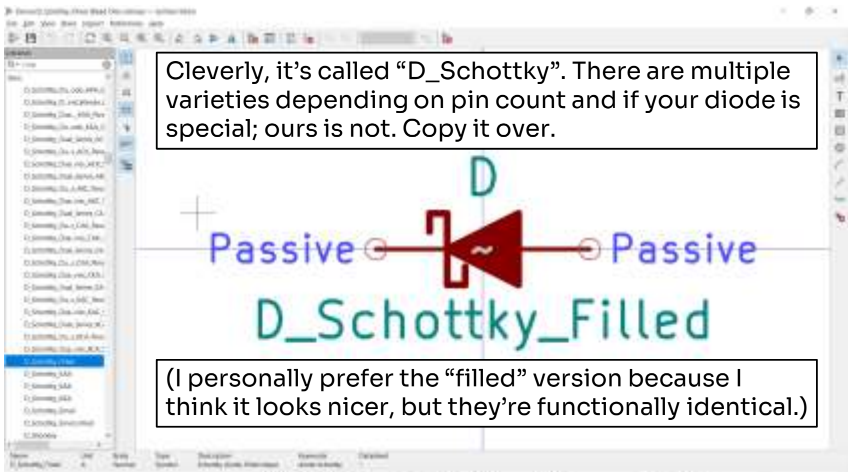


If you guessed “D”, you’re partway there!

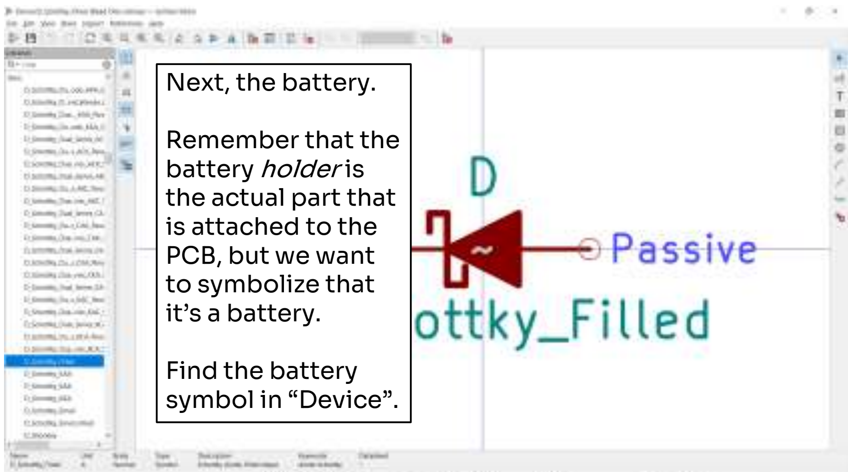
“D” is just a regular diode, but all diode-type symbol names start with “D” as well.

Remember, we need a Schottky diode.

Symbol Library



Symbol Library

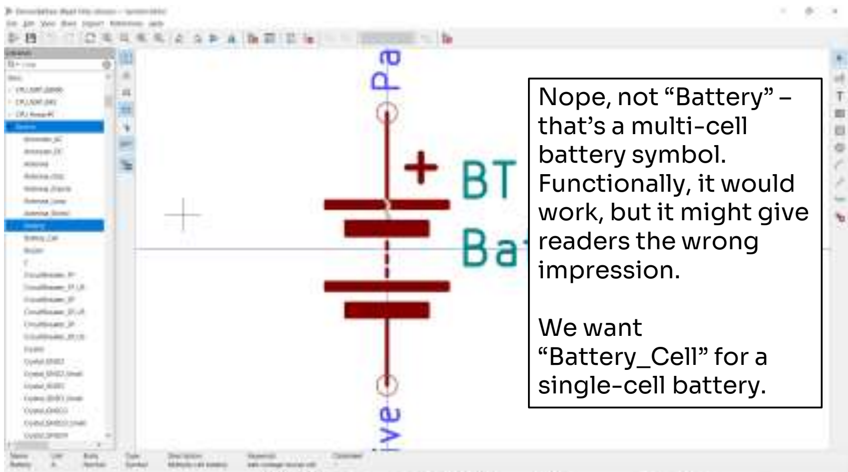


Next, the battery.

Remember that the battery *holder* is the actual part that is attached to the PCB, but we want to symbolize that it's a battery.

Find the battery symbol in "Device".

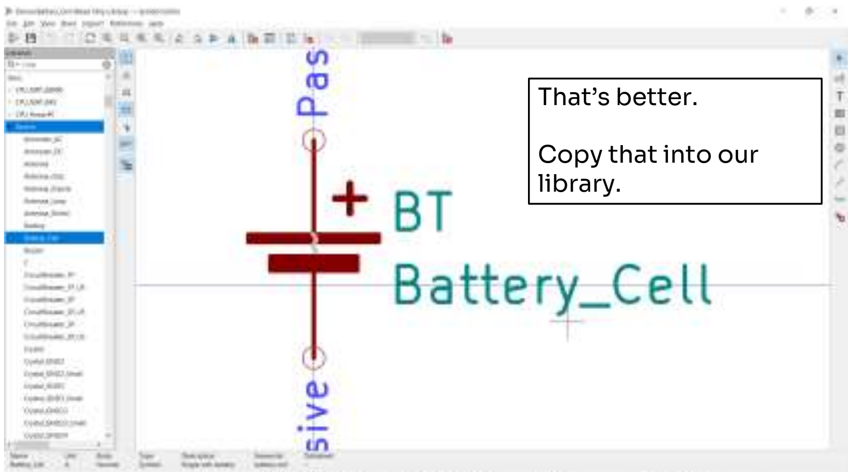
Symbol Library



Nope, not “Battery” – that’s a multi-cell battery symbol. Functionally, it would work, but it might give readers the wrong impression.

We want “Battery_Cell” for a single-cell battery.

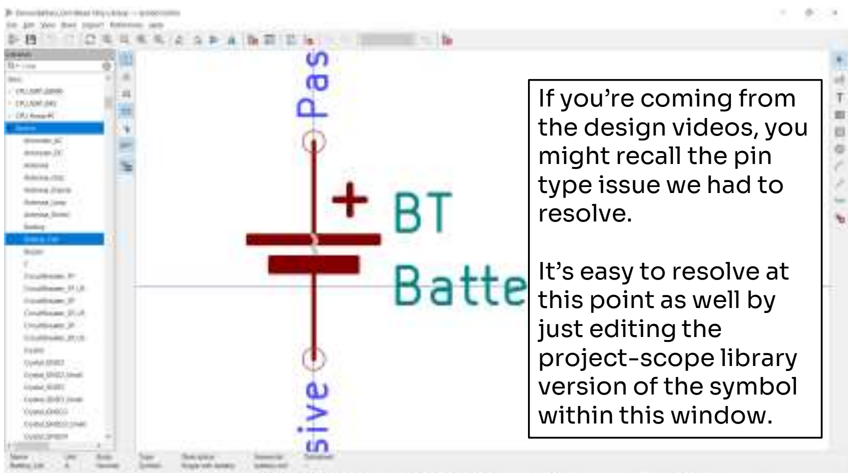
Symbol Library



That's better.

Copy that into our library.

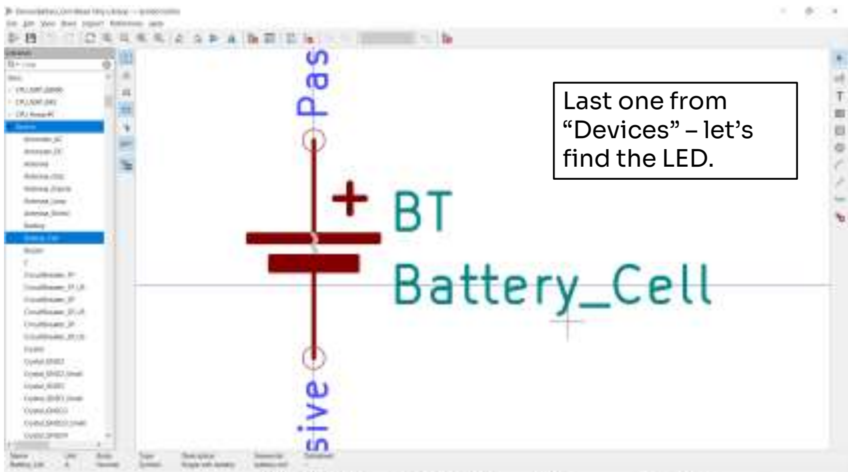
Symbol Library



If you're coming from the design videos, you might recall the pin type issue we had to resolve.

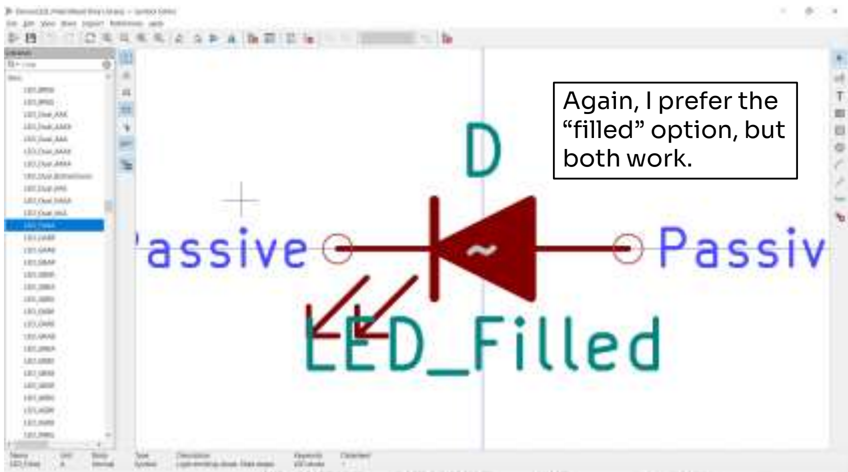
It's easy to resolve at this point as well by just editing the project-scope library version of the symbol within this window.

Symbol Library



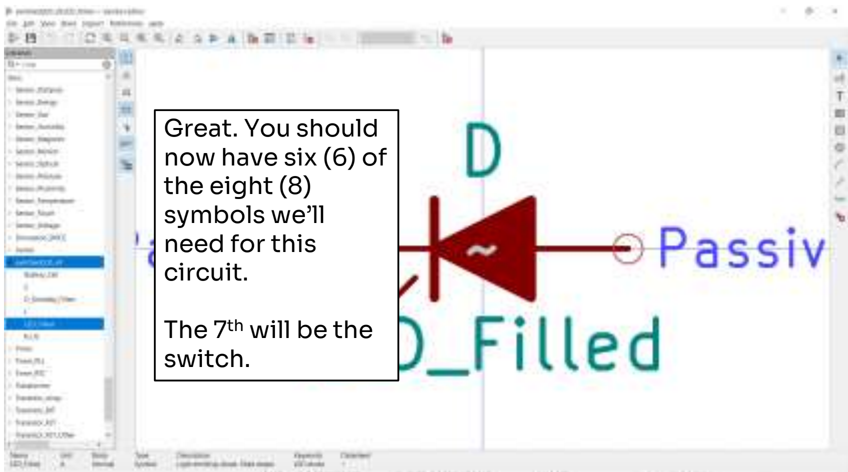
Last one from "Devices" – let's find the LED.

Symbol Library



Again, I prefer the “filled” option, but both work.

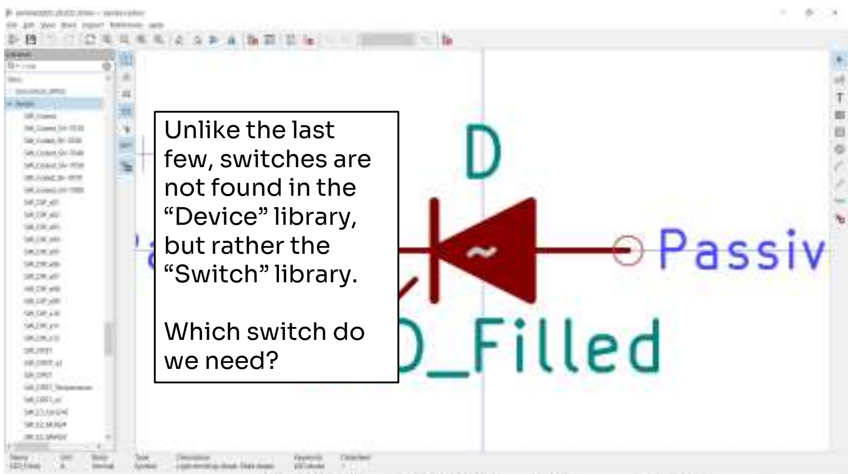
Symbol Library



Great. You should now have six (6) of the eight (8) symbols we'll need for this circuit.

The 7th will be the switch.

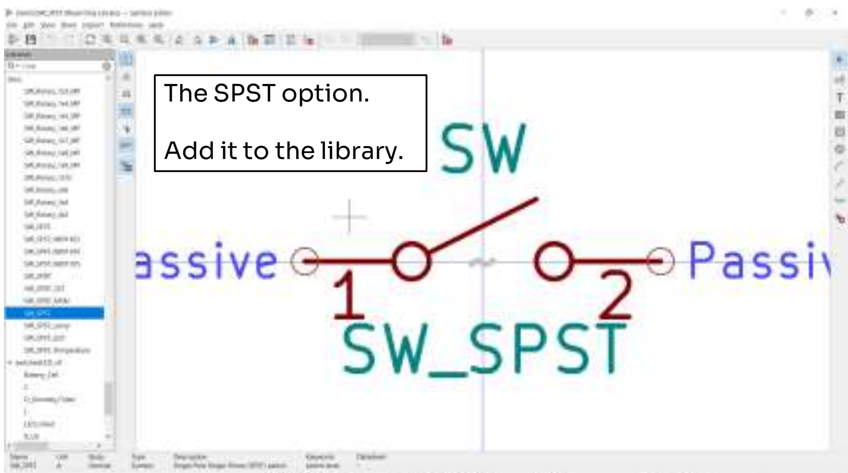
Symbol Library



Unlike the last few, switches are not found in the "Device" library, but rather the "Switch" library.

Which switch do we need?

Symbol Library



If you don’t remember the terminology, SPST stands for single-pole, single-throw, meaning it controls one circuit, or one pathway, with one output.

Symbol Library – Adding the IC

- The last component to add is the IC
- The IC is slightly different than the other components we have put down, for two reasons:
  1. The symbol is “unique”, or at least, a symbol with the same pinout would be difficult to locate in the built-in libraries
  2. The footprint may or may not be in the libraries (although it is a standard package, so it likely is… but we’ll pretend.)

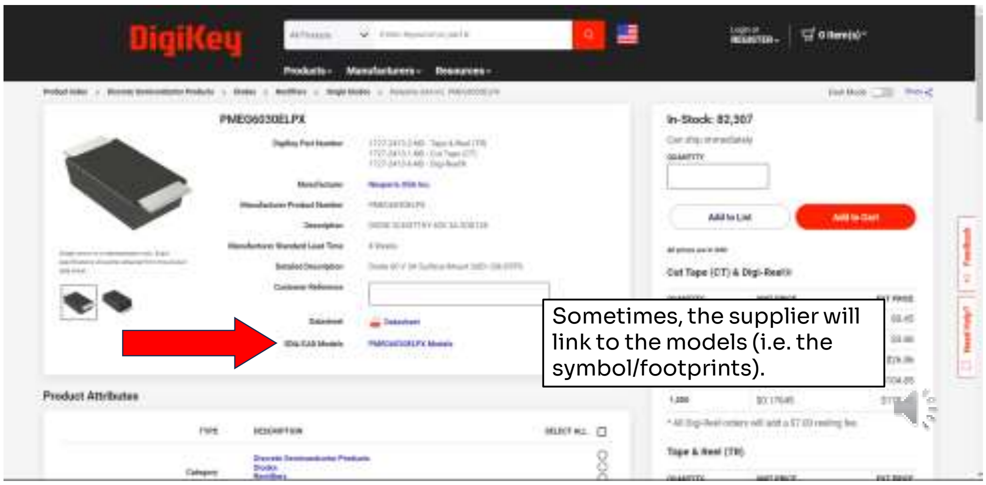
[With an empty slide] If you’ve already generated this symbol during the design videos, congrats! Skip to basically the end of this video, where I’ll show you how to manage which libraries are available to the project. Sorry I can’t be more specific.

For the rest of you…* [and continue with slide]

Symbol Library – Adding the IC

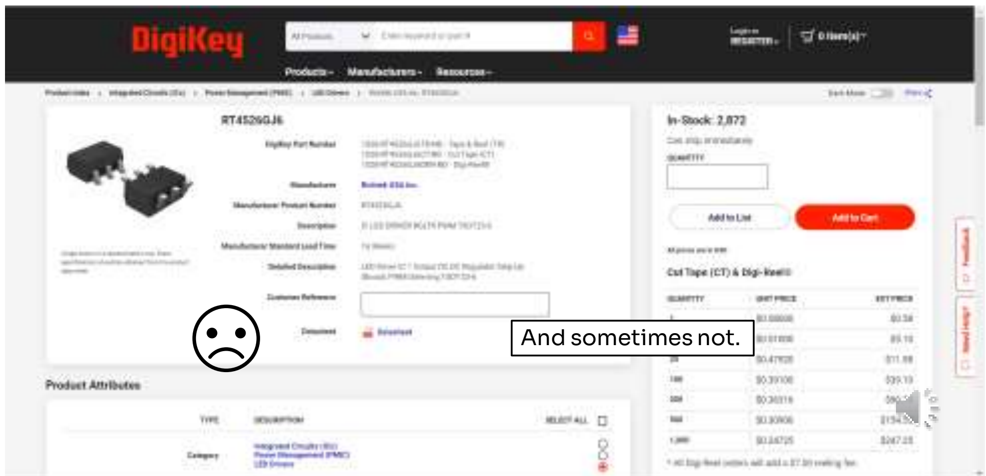
- The first thing to check is whether it's in one of the built-in libraries.
- We can use the filter to search for the RT4526.
- Nope.
  - This is pretty normal for most ICs and non-standard parts because, as I mentioned before, they're pseudo-unique.
- The next option is to see if someone else has already done the work of generating a symbol and footprint

Locating a model for the IC



Sometimes, the suppliers will link to models.

Locating a model for the IC



Sometimes not.

Locating a model for the IC

- There are plenty of places online who can generate (or may already have) these files.
- The two I use are UltraLibrarian and ~~SnapEDA~~ SnapMagic.
- UL does not require an account to download models, but does require one to request new models.
- SnapMagic ~~SnapEDA~~ requires an account to download or request, but you can also generate symbols in-browser.
- Both accounts are free to open.

If the supplier doesn’t link, we can go look for them manually.

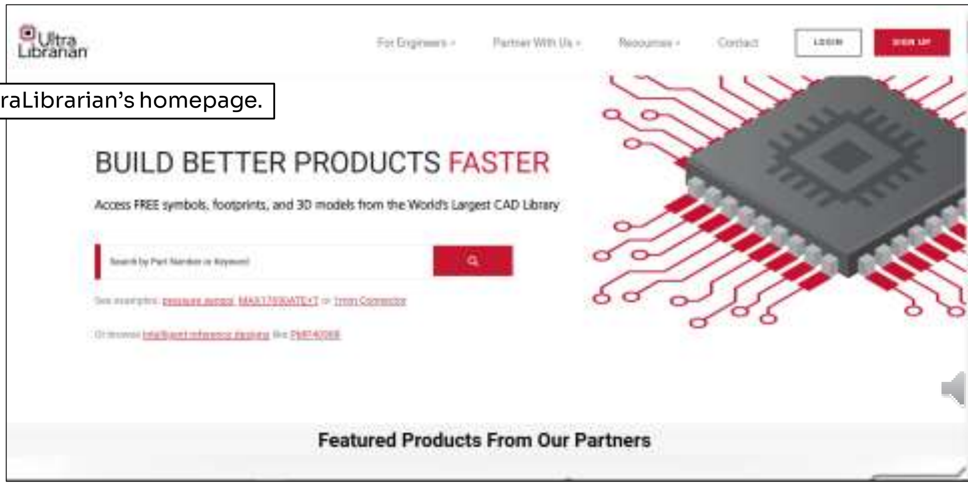
*The two places I have had good results with are Ultra Librarian and SnapMagic, which used to be called SnapEDA.

*Both require free accounts to request new models, but Ultra Librarian allows you to download pre-made ones without one.

*SnapMagic has a few additional tools that are available for registered accounts as well, like in-browser symbol and footprint generation.

**

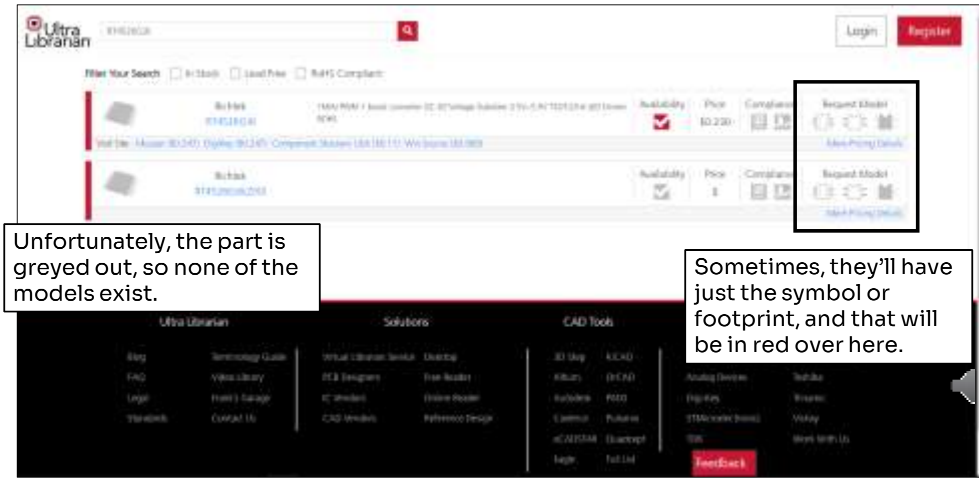
Locating a model for the IC



UltraLibrarian's homepage.

We'll start arbitrarily with Ultra Librarian. Search for the symbol you'd like a model for.

Locating a model for the IC



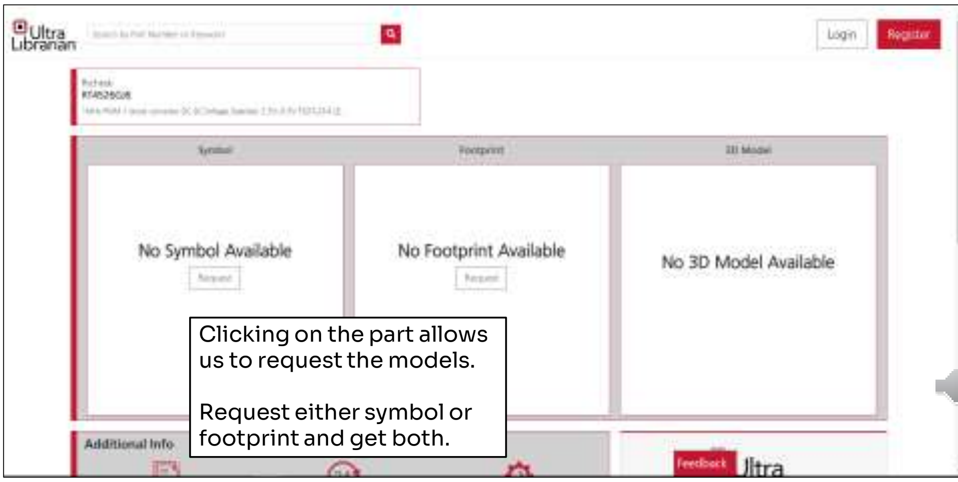
Unfortunately, the part is greyed out, so none of the models exist.

Sometimes, they'll have just the symbol or footprint, and that will be in red over here.

Unfortunately, the part is grayed, so no models exist.

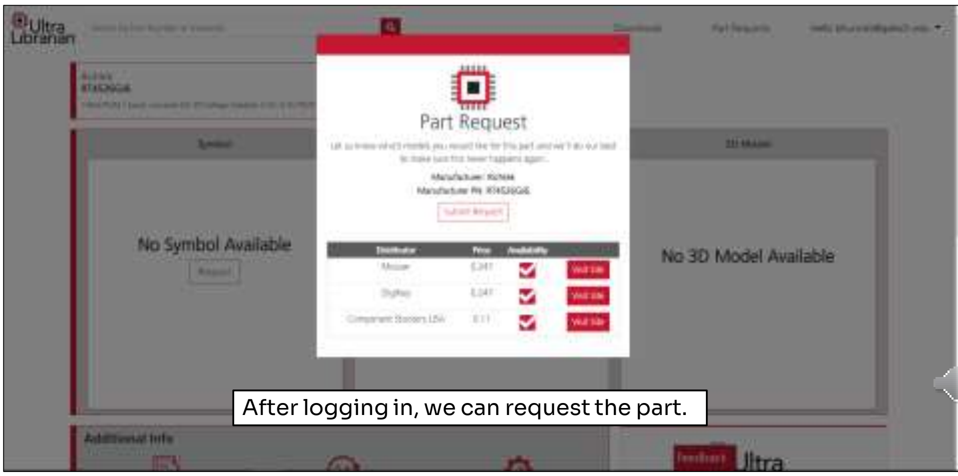
*Sometimes, the icons on the right will be filled in, indicating that they have a footprint or a symbol, but not today.

Locating a model for the IC



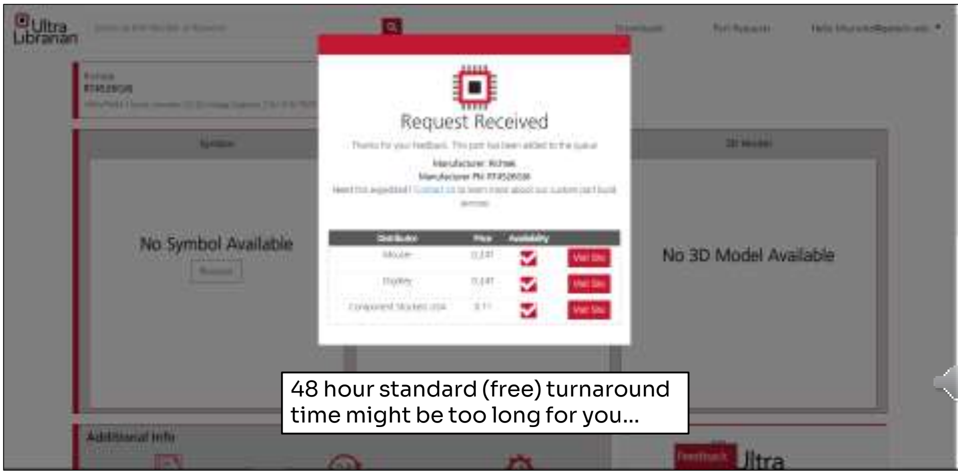
Clicking the part gets us to the part page, where we can ask them to create a model for us. Request either to get both.

Locating a model for the IC

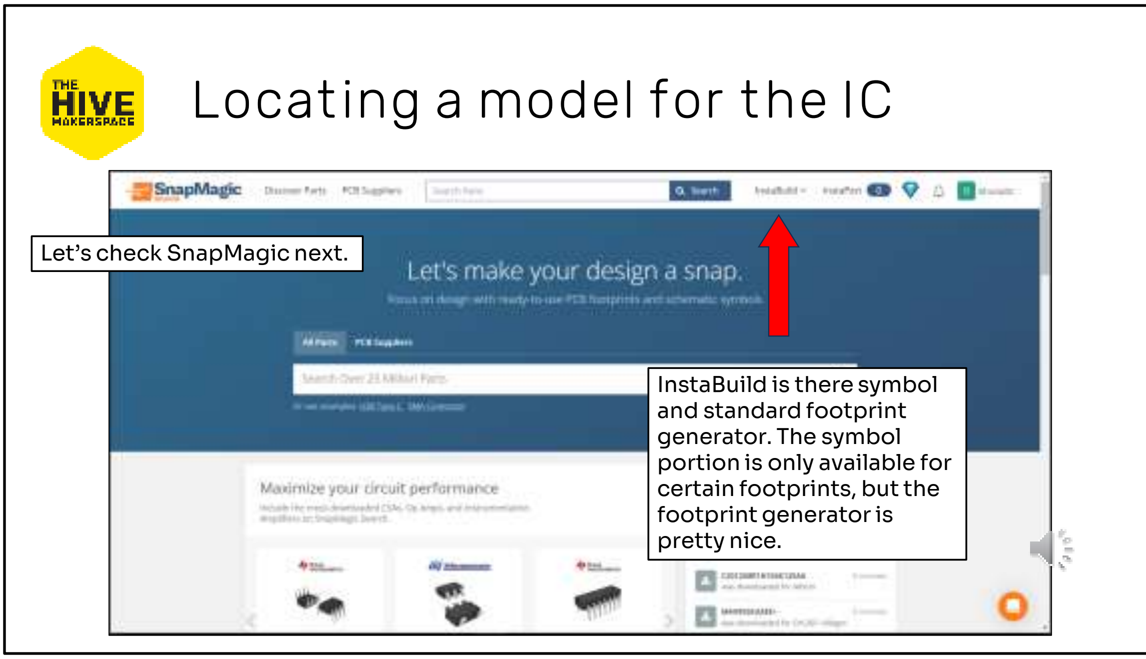


Looks like this

Locating a model for the IC



Standard turnaround guarantee is 48 hours, which may or may not be too long for you.



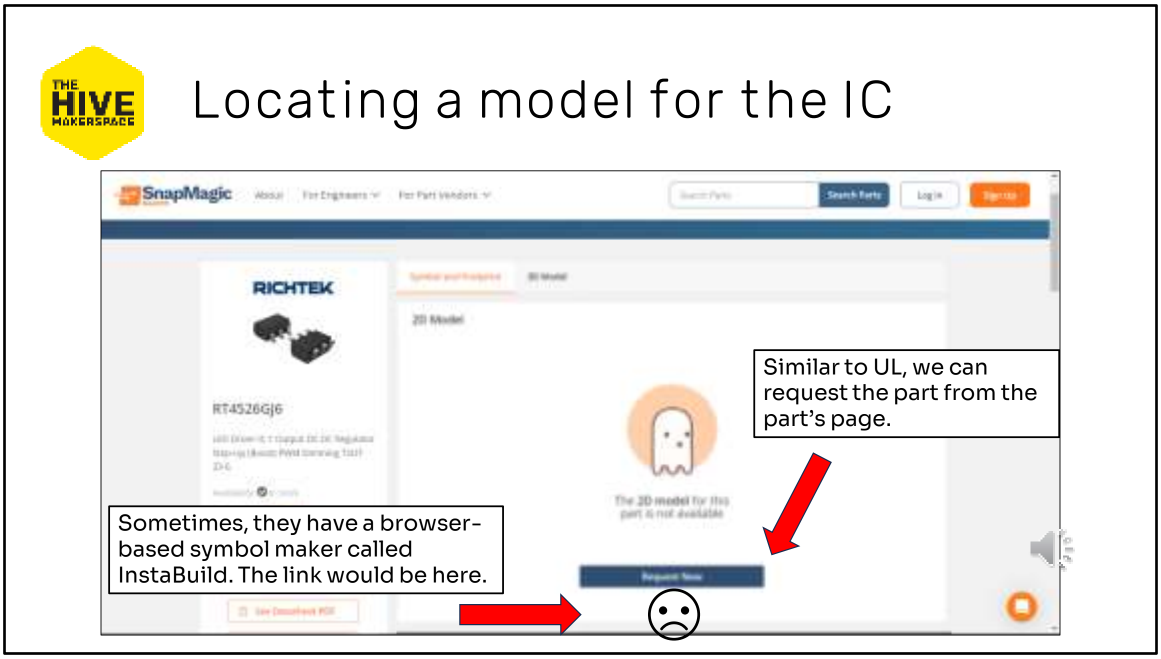
Let's check our other source, SnapMagic.

*SnapMagic's in-browser model generator is linked on their homepage, but while most standard IC footprints can be generated, the symbols are limited to select footprints only.

Locating a model for the IC

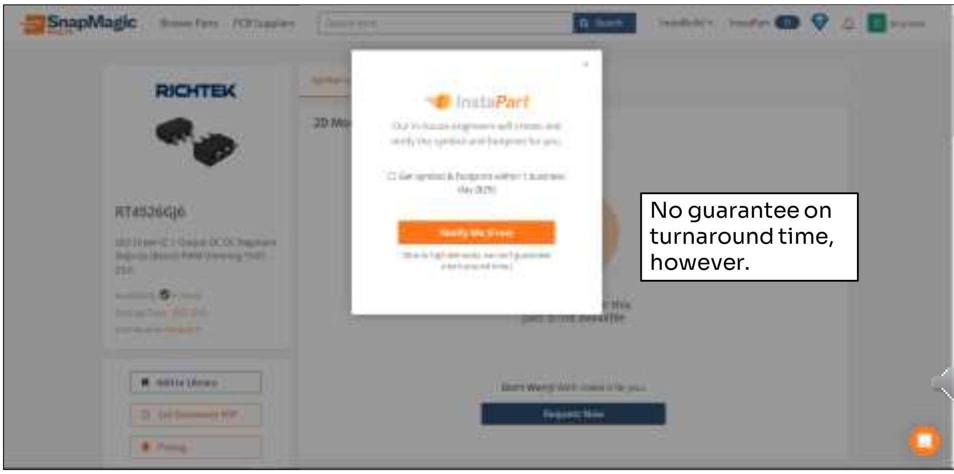
Searching for our part brings us to this page, *but be careful! Sometimes they recommend a part at the top that isn't right.

*Still, no models available – we can see this with the empty icons on the right.



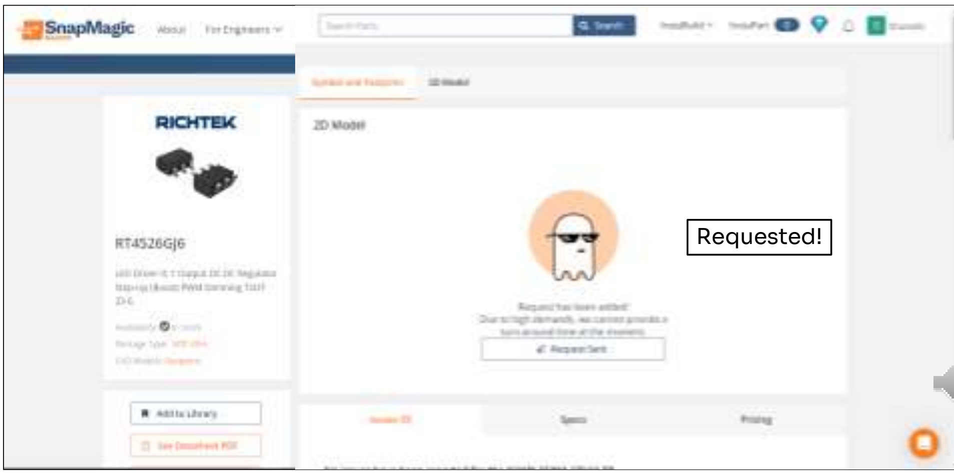
We can request the models from the parts page. *There would also be a link here if the in-brower symbol generator was available for this part, but it's not. Sad.

Locating a model for the IC



Unfortunately, SnapMagic doesn't have a guarantee on turnaround time.

Locating a model for the IC



But you get a cool ghost when you request the part.

~~Locating~~ Creating a model for the IC

- Struck out thrice. What to do?
- Can keep searching (someone *must* have made this before, right?), but returns may be diminishing.
- Because KiCAD stores symbols and footprints in separate libraries, we'll just go ahead and create our own symbol directly in KiCAD.
  - No need to create our own footprint (yet)

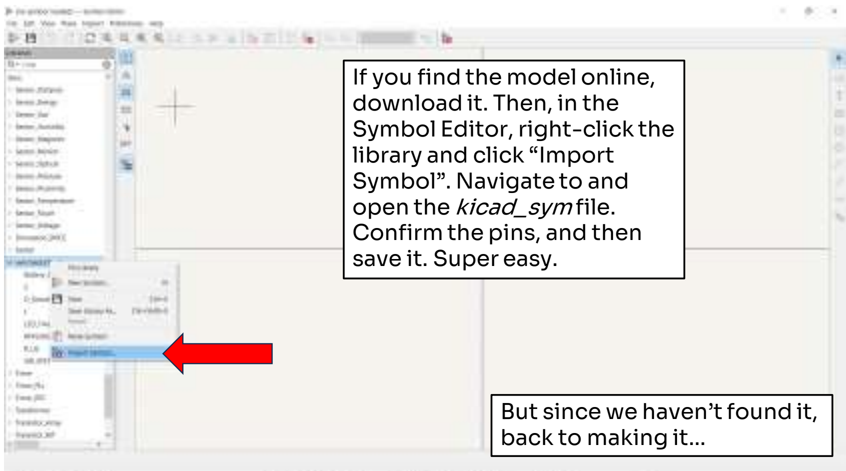
So three strikes. Are we out?

*We can keep searching randomly on the internet, but the returns will likely be diminishing, and the quality might be degraded.

*Instead, we can just go ahead and make the symbol in KiCAD directly.

Symbols are relatively easy to make, and are less prone to errors than a custom footprint, though they can be tedious with many pins, so it's totally doable to make them without waiting for Ultra Librarian or Snap Magic to be done.

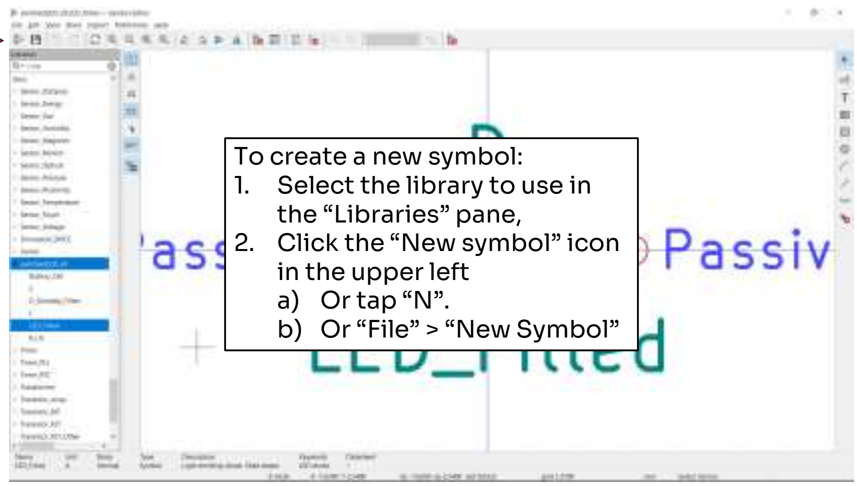
Aside: Importing a model



If you find the model online, download it. Then, in the Symbol Editor, right-click the library and click “Import Symbol”. Navigate to and open the *kicad_sym* file. Confirm the pins, and then save it. Super easy.

But since we haven’t found it, back to making it…

Creating a symbol for the IC



To create a new symbol:
1. Select the library to use in the "Libraries" pane,
2. Click the "New symbol" icon in the upper left
   a) Or tap "N".
   b) Or "File" > "New Symbol"

Next, we can create a new symbol just hitting the "N" key, the icon in the upper-left, or under the "File" menu.

Creating a symbol for the IC

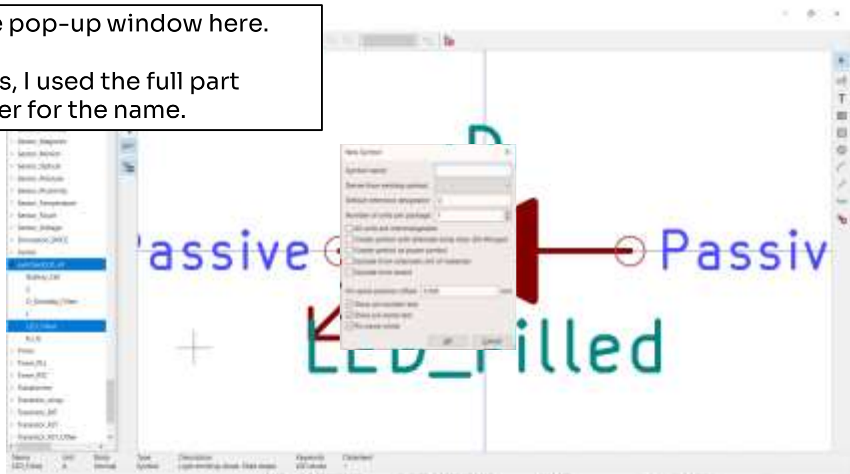
Fill in the pop-up window here.

Let’s fill in the window here.

Creating a symbol for the IC

Fill in the pop-up window here.

- For ICs, I used the full part number for the name.

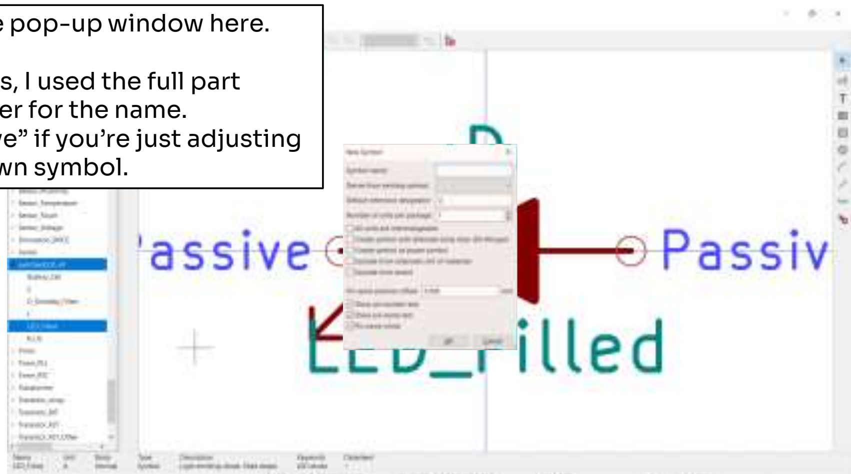


Generally, good policy is to name the symbol with the part number, which in this case would be RT4526. You can leave off the GJ6 in this case because, if you read the datasheet, those are the only options for the final three digits.

Creating a symbol for the IC

Fill in the pop-up window here.

- For ICs, I used the full part number for the name.
- “Derive” if you’re just adjusting a known symbol.

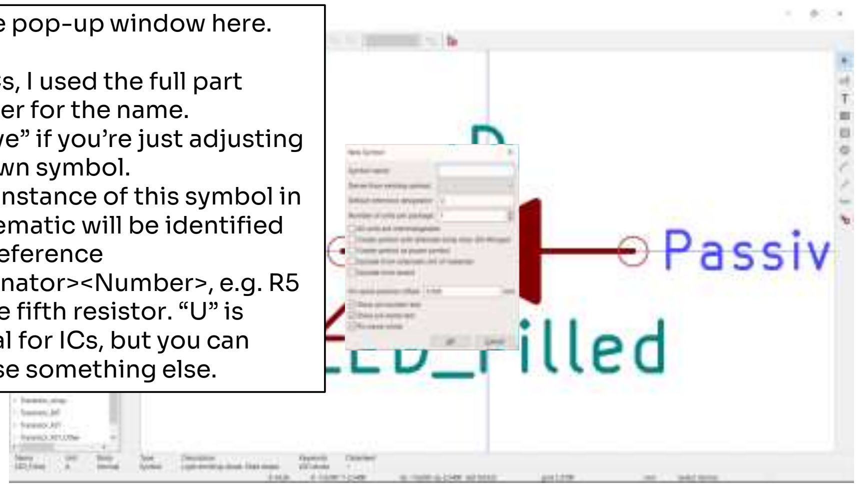


“Derive” refers to if you’re adjusting a known symbol, like creating a polarized capacitor by copying the basic capacitor.

Creating a symbol for the IC

Fill in the pop-up window here.

- For ICs, I used the full part number for the name.
- “Derive” if you’re just adjusting a known symbol.
- Each instance of this symbol in a schematic will be identified by <Reference Designator><Number>, e.g. R5 for the fifth resistor. “U” is typical for ICs, but you can choose something else.



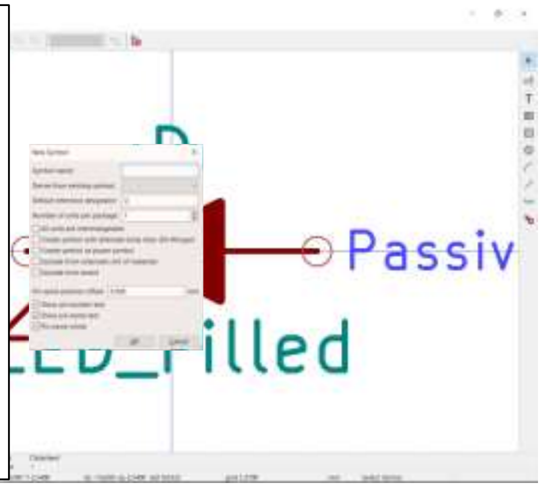
The reference designator is the letter that KiCAD uses to identify all instances of this symbol. Each instance will also get a number, so a resistor might be designated R5 for the fifth resistor of the schematic. “U” is very common for ICs, though you could choose something else if you’d like. It’s fine for multiple symbols to have the same designator character, like U or R; it just means they’re of the same type, so to speak.

Creating a symbol for the IC

Fill in the pop-up window here.

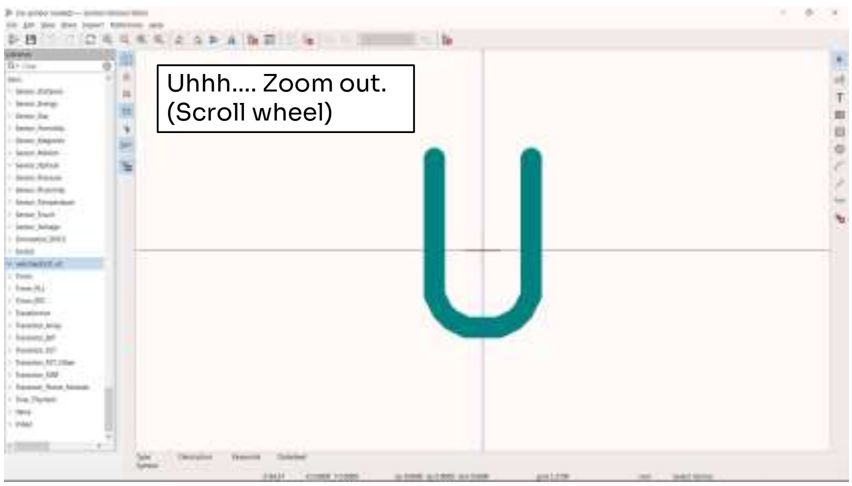
- For ICs, I used the full part number for the name.
- “Derive” if you’re just adjusting a known symbol.
- Each instance of this symbol in a schematic will be identified by <Reference Designator><Number>, e.g. R5 for the fifth resistor. “U” is typical for ICs, but you can choose something else.

The rest is not relevant now. Click OK to continue.

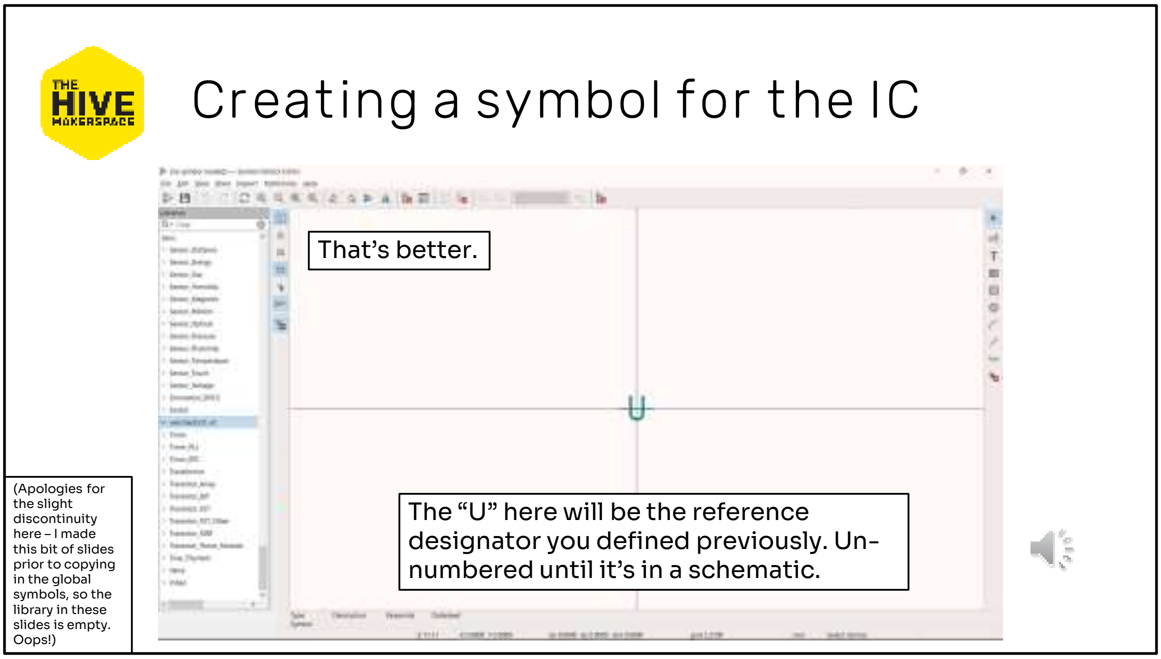


The rest doesn’t matter to us, so click “OK”.

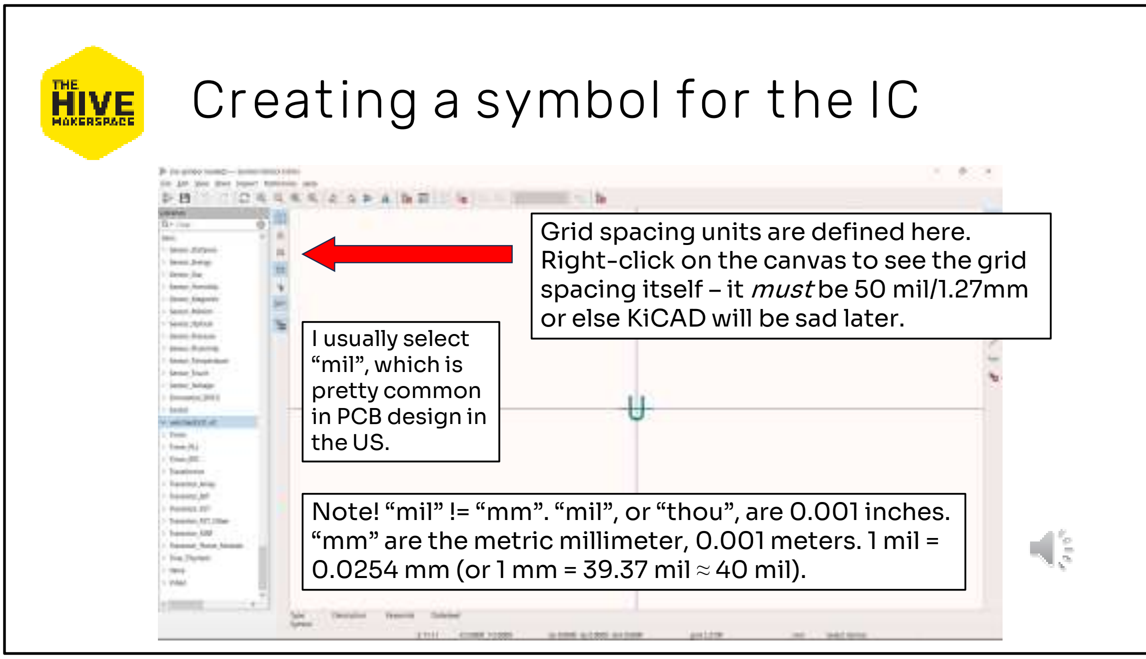
Creating a symbol for the IC



Since there’s nothing on the editor now, it will zoom automatically in to the reference designator.



Something like this is better. The "U" is currently placed at the anchor point, which is the point at which the symbol will be attached to the mouse. We'll move those both later.

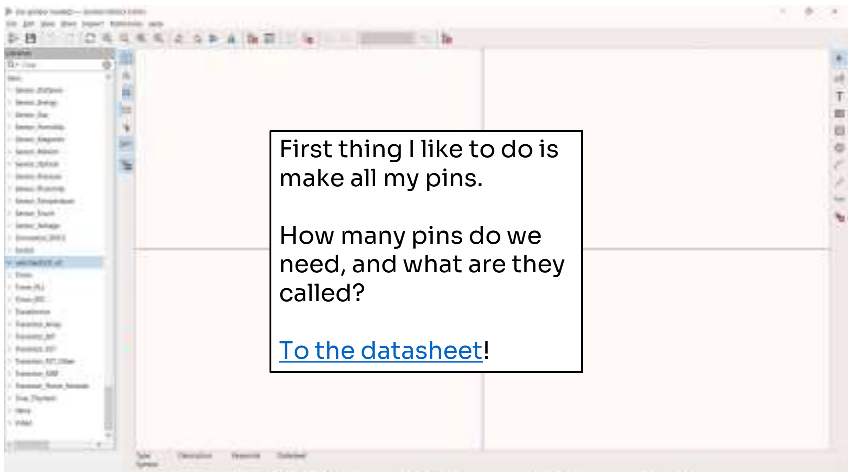


As usual, grid spacing and units are defined on the left. Check the grid by right-clicking. It should read 50 mil or 1.27 mm, or else KiCAD will not be able to attach wires to it and you'll be sad later.

I usually select mils for my units because I'm used to thinking of hole sizes and trace widths in mils, and because I'm used to imperial units. But you can readily use metric units as well. KiCAD is based on metric values, after all.

Note that mils and millimeters are not the same. Mils, which are also known as thous, are a thousandth of an inch, or 0.001 inches. Millimeters are, of course, a thousandth of a meter. One millimeter is about forty mils.

Creating a symbol for the IC



First thing I like to do is make all my pins.

How many pins do we need, and what are they called?

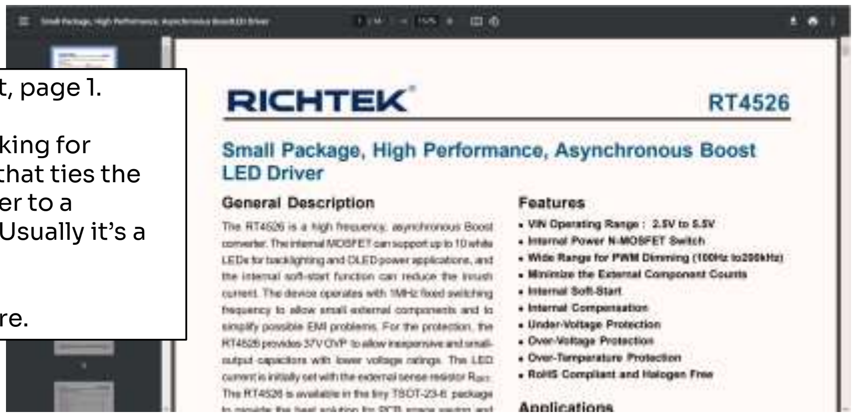
To the datasheet!

Creating a symbol for the IC

Datasheet, page 1.

We're looking for anything that ties the pin number to a function. Usually it's a table.

So not here.


RICHTEK

RT4526

Small Package, High Performance, Asynchronous Boost LED Driver

General Description

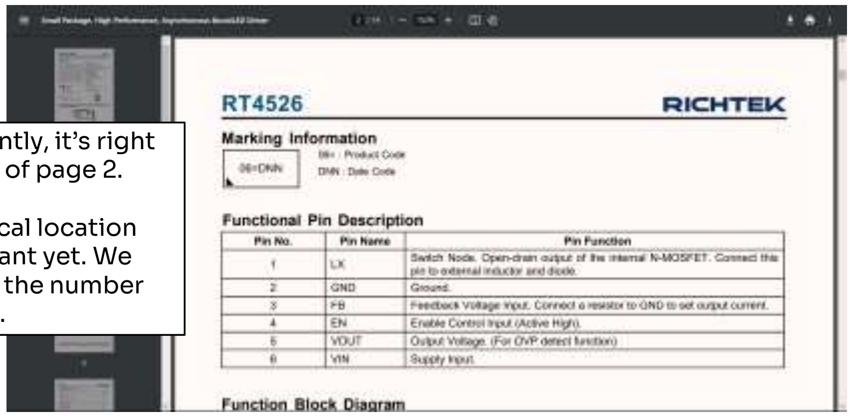
The RT4526 is a high frequency, asynchronous Boost converter. The internal MOSFET can support up to 10 white LEDs for backlighting and OLED power applications, and the internal soft-start function can reduce the inrush current. The device operates with 1MHz fixed switching frequency to allow small external components and to simplify possible EMI problems. For the protection, the RT4526 provides 37V OVP to allow inexpensive and small-output capacitors with lower voltage ratings. The LED current is initially set with the external sense resistor R_{SET}. The RT4526 is available in the tiny TSOT-23-6 package

Features

- VIN Operating Range : 2.5V to 5.5V
- Internal Power N-MOSFET Switch
- Wide Range for PWM Dimming (100Hz to200kHz)
- Minimize the External Component Counts
- Internal Soft-Start
- Internal Compensation
- Under-Voltage Protection
- Over-Voltage Protection
- Over-Temperature Protection
- RoHS Compliant and Halogen Free

Applications

Creating a symbol for the IC


RT4526 RICHTEK

Marking Information

0E=DNN

0E= : Product Code

DNN : Date Code

Functional Pin Description

Pin No.	Pin Name	Pin Function
1	LX	Switch Node. Open-drain output of the internal N-MOSFET. Connect this pin to external inductor and diode.
2	GND	Ground.
3	FB	Feedback Voltage Input. Connect a resistor to GND to set output current.
4	EN	Enable Control Input (Active High).
5	VOUT	Output Voltage. (For OVP detect function)
6	VIN	Supply Input.

Function Block Diagram

Conveniently, it's right at the top of page 2.

The physical location isn't relevant yet. We only want the number and name.

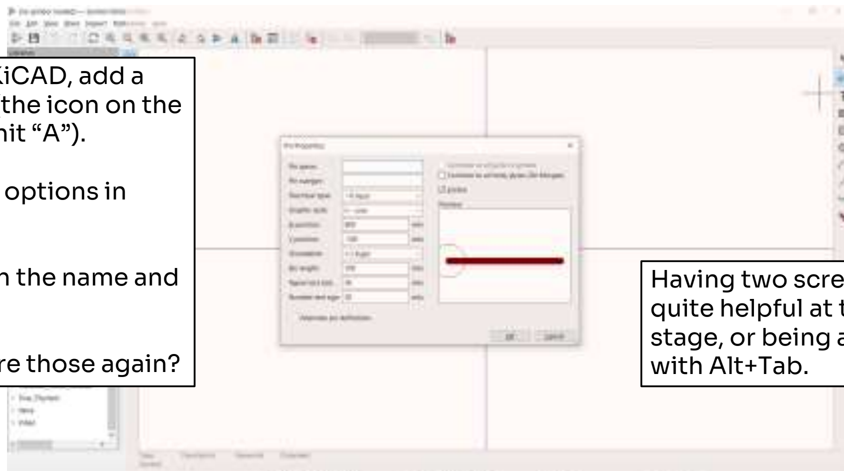
Creating a symbol for the IC

Back to KiCAD, add a new pin (the icon on the right, or hit “A”).

Bunch of options in here.

Start with the name and number.

What were those again?



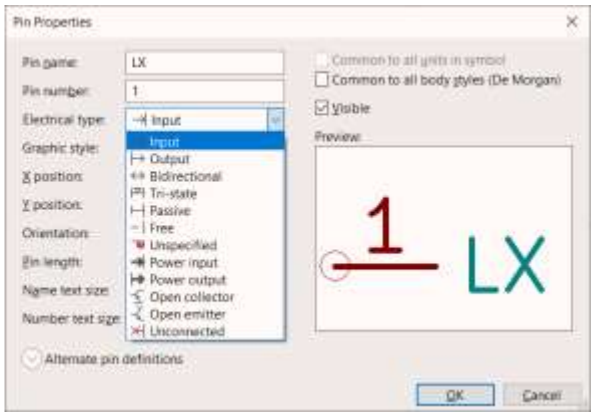
Having two screens is quite helpful at this stage, or being a wizard with Alt+Tab.

Creating a symbol for the IC

Start with pin 1.

By default, it's set as an "input". But is it?

Back to the pinout table!



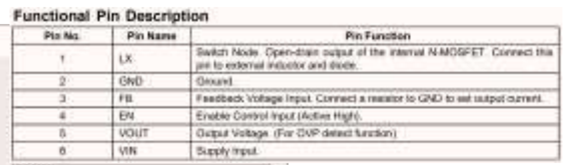
Functional Pin Descripti

Pin No.	Pin Name
1	LX
2	GND
3	FB
4	EN
5	VOUT
6	VIN

Creating a symbol for the IC

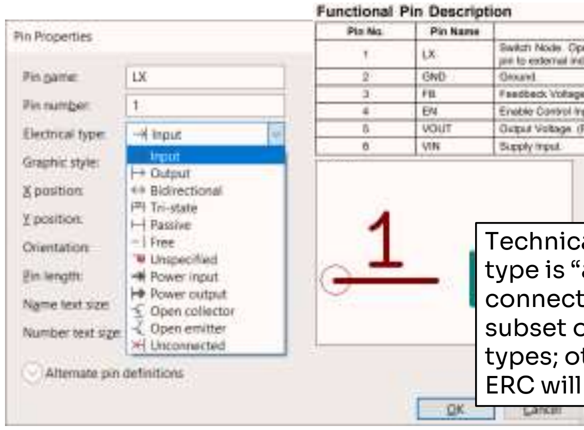
If you're not sure what it is, you can either set it to "Bidirectional" or "Unspecified" or "Passive".

The type won't cause the design to fail, but it might cause you headaches later with the ERC.



Functional Pin Description

Pin No.	Pin Name	Pin Function
1	LX	Switch Node. Open-drain output of the internal N-MOSFET. Connect this pin to external inductor and diode.
2	GND	Ground.
3	FB	Feedback Voltage Input. Connect a resistor to GND to set output current.
4	EN	Enable Control Input (Active High).
5	VOUT	Output Voltage. (For OVP detect function)
6	VIN	Supply Input.

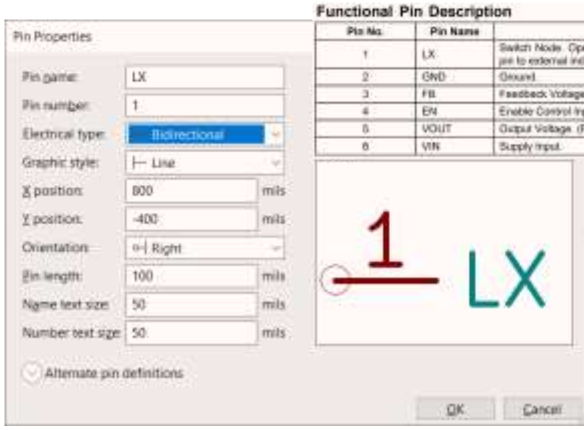


Technically, each pin type is "allowed" connected to only a subset of other pin-types; otherwise, the ERC will throw an error.

Creating a symbol for the IC

The rest of these are graphical choices, so we'll adjust them as needed later.

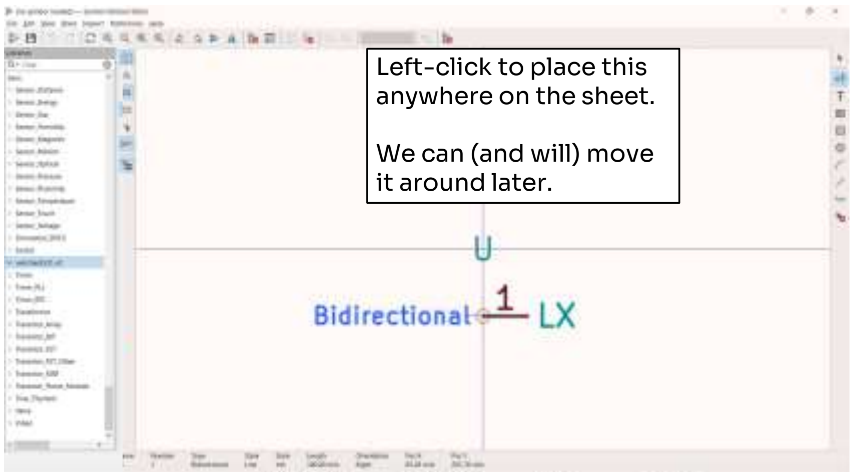
Click OK.



Functional Pin Description

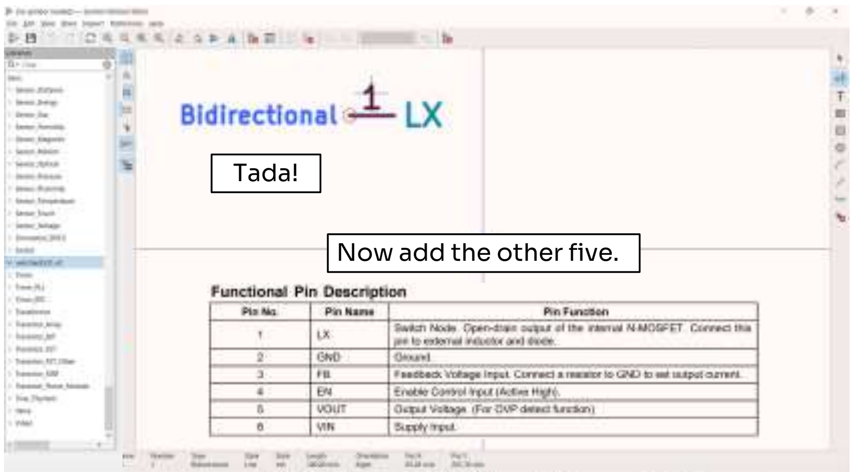
Pin No.	Pin Name	Pin Function
1	LX	Switch Node. Open-drain output of the internal N-MOSFET. Connect this pin to external inductor and diode.
2	GND	Ground.
3	FB	Feedback Voltage Input. Connect a resistor to GND to set output current.
4	EN	Enable Control Input (Active High).
5	VOUT	Output Voltage. (For OVP detect function)
6	VIN	Supply Input.

Creating a symbol for the IC

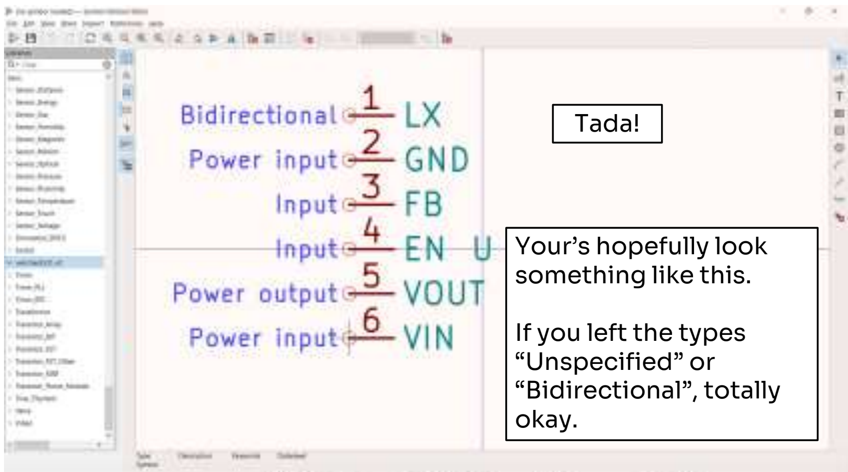


The pin type is in blue there.

Creating a symbol for the IC



Creating a symbol for the IC

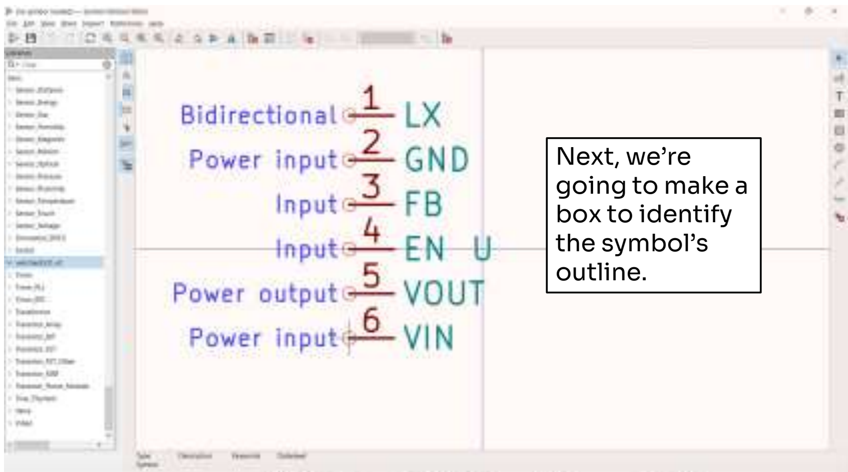


Tada!

Your's hopefully look something like this.

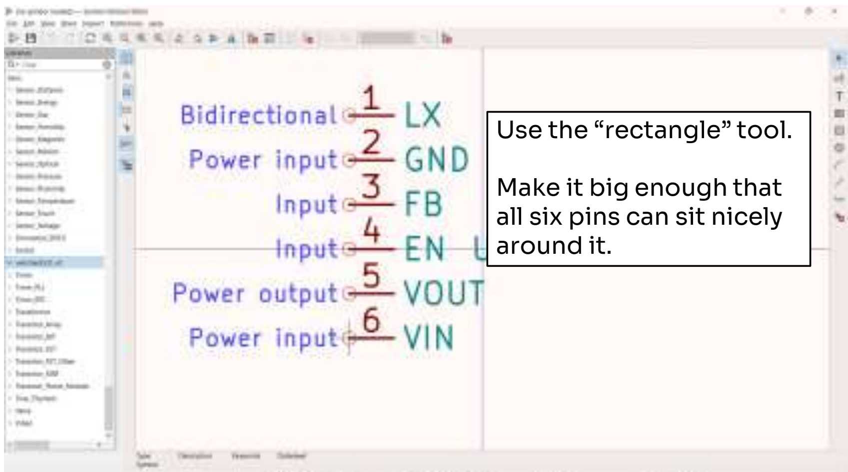
If you left the types "Unspecified" or "Bidirectional", totally okay.

Creating a symbol for the IC



Next, we're going to make a box to identify the symbol's outline.

Creating a symbol for the IC



Use the “rectangle” tool.

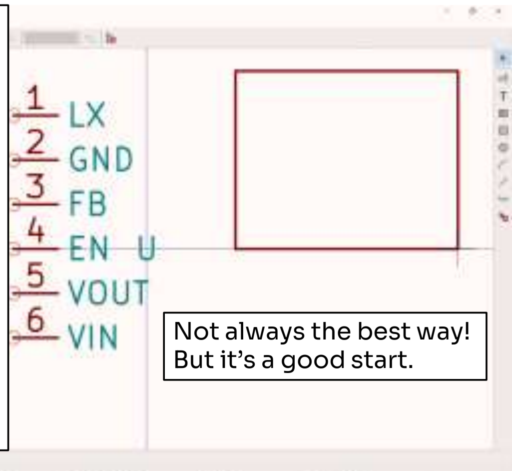
Make it big enough that all six pins can sit nicely around it.

Creating a symbol for the IC

Mine looks like this.

For height, I conceptually split the pins in half and make the box tall enough for half the pins spaced two grids apart plus two grids above and below. In this case, that's 3 pins x 2 grids per pin + 2 extra top grids + 2 extra bottom grids = 10 grids.

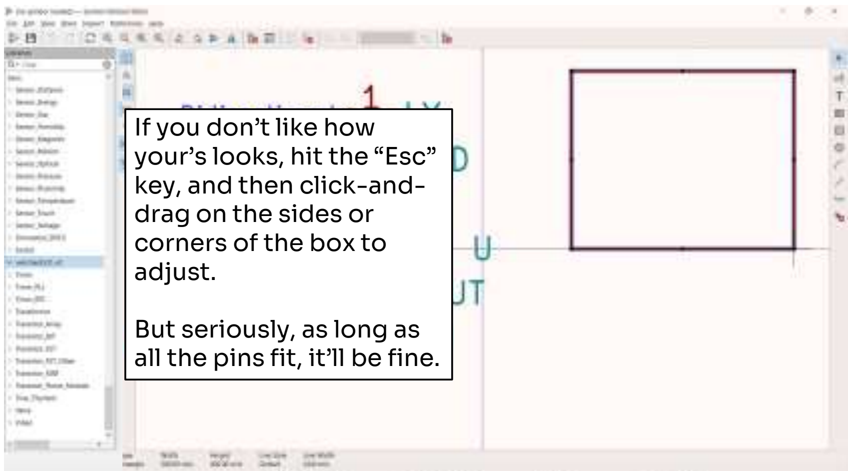
For width, I count the grides for the widest label, double it, and add two. "VOUT" is the widest at 4 grids, so that's 10 grids wide.



Not always the best way!
But it's a good start.

This is the algorithm by which I spec'd my box's size, but I abandoned that size basically in the next few slides, so you're safe to ignore that.

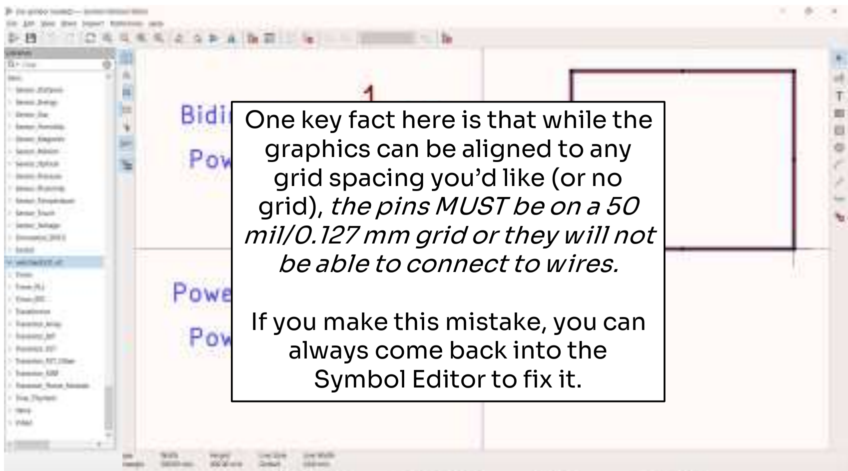
Creating a symbol for the IC



If you don't like how your's looks, hit the "Esc" key, and then click-and-drag on the sides or corners of the box to adjust.

But seriously, as long as all the pins fit, it'll be fine.

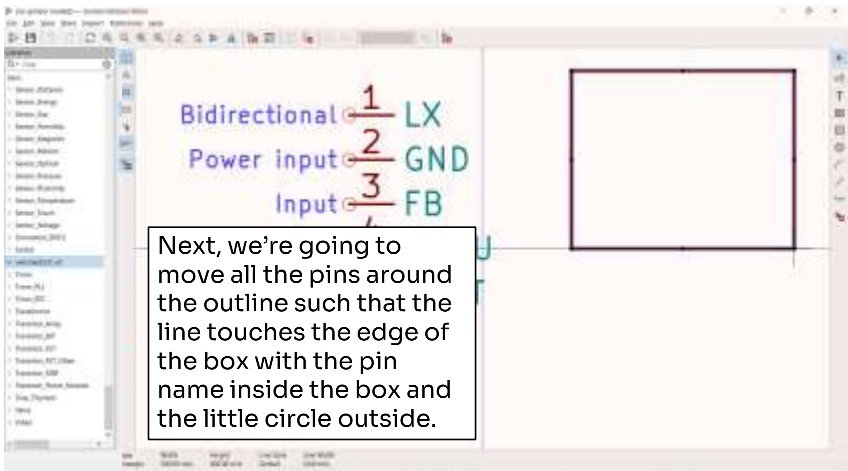
Creating a symbol for the IC



One key fact here is that while the graphics can be aligned to any grid spacing you'd like (or no grid), *the pins MUST be on a 50 mil/0.127 mm grid or they will not be able to connect to wires.*

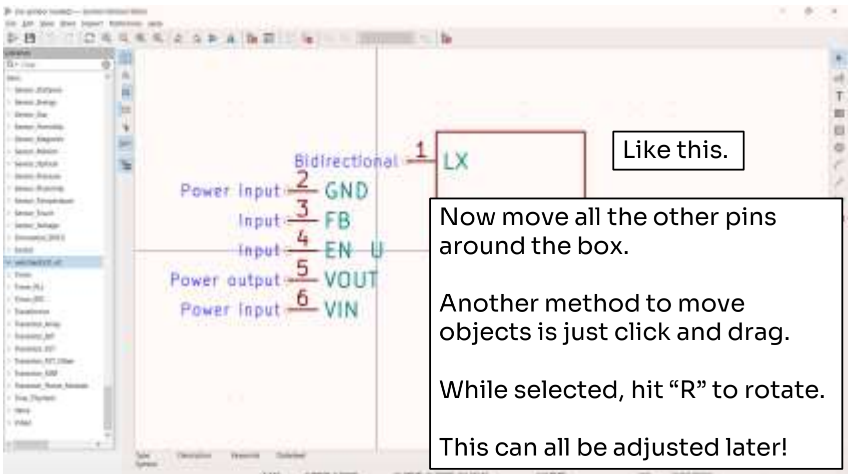
If you make this mistake, you can always come back into the Symbol Editor to fix it.

Creating a symbol for the IC



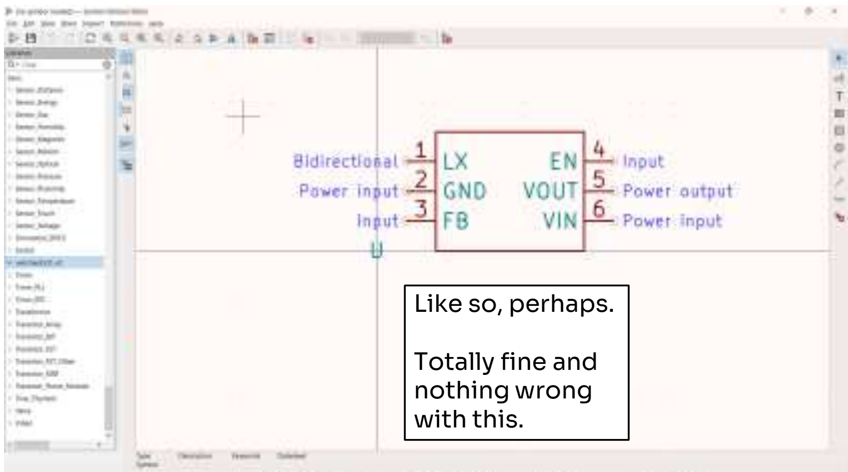
Next, we're going to move all the pins around the outline such that the line touches the edge of the box with the pin name inside the box and the little circle outside.

Creating a symbol for the IC



You might consider pausing the video here while you place your pins before seeing what I did.

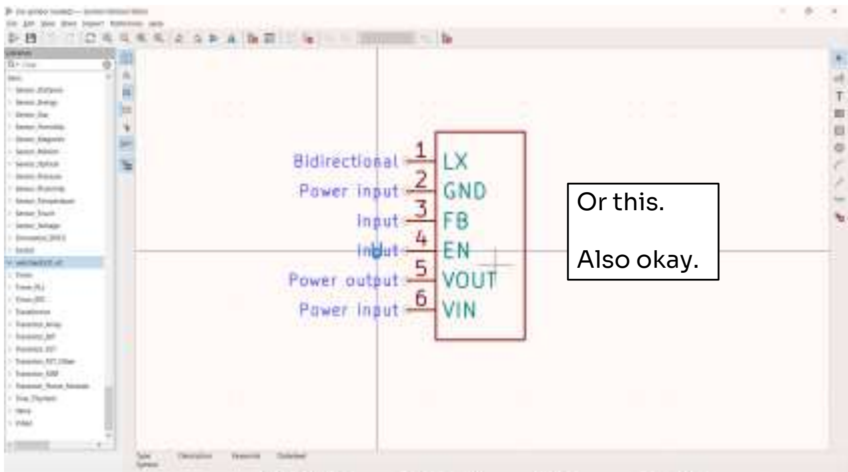
Creating a symbol for the IC



Like so, perhaps.

Totally fine and nothing wrong with this.

Creating a symbol for the IC



Or this.

Also okay.

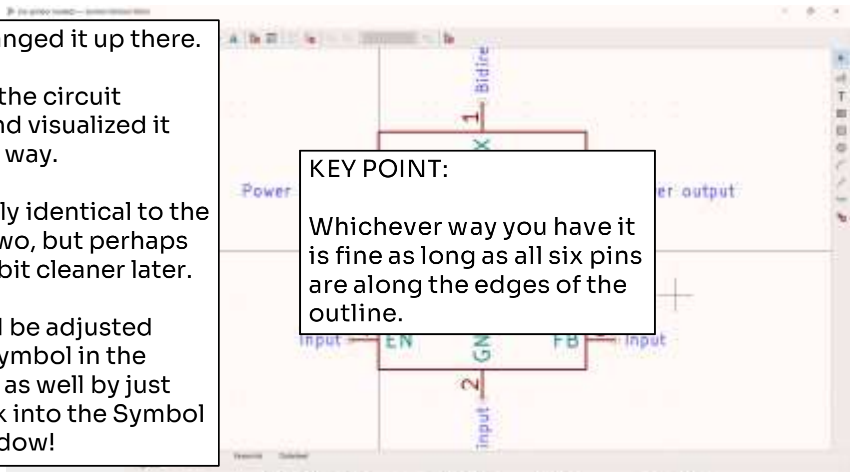
Creating a symbol for the IC

Totally changed it up there.

Looked at the circuit drawing and visualized it better this way.

Functionally identical to the previous two, but perhaps will look a bit cleaner later.

This can all be adjusted once the symbol in the schematic as well by just going back into the Symbol Editor window!



KEY POINT:

Whichever way you have it is fine as long as all six pins are along the edges of the outline.

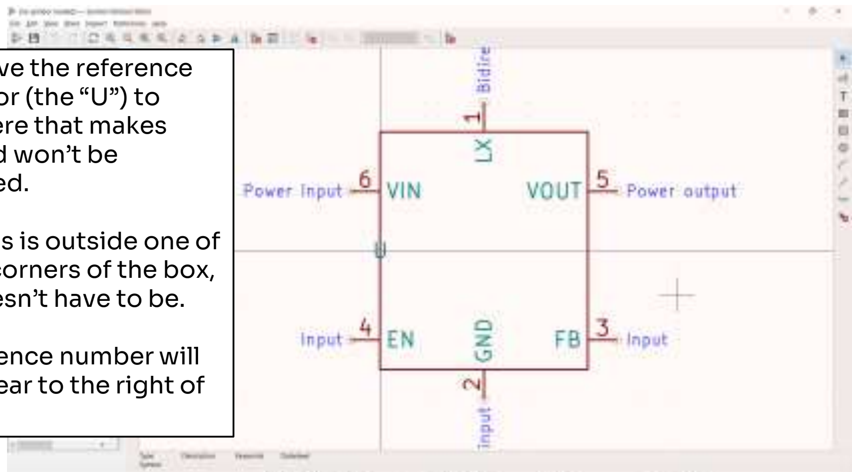
Additionally, when we learn about nets in the next video, you might understand why it doesn't matter how the symbol is laid out so much – pins can connect anywhere without a direct line.

Creating a symbol for the IC

Next, move the reference designator (the "U") to somewhere that makes sense and won't be obstructed.

Often, this is outside one of the four corners of the box, but it doesn't have to be.

The reference number will later appear to the right of the "U".



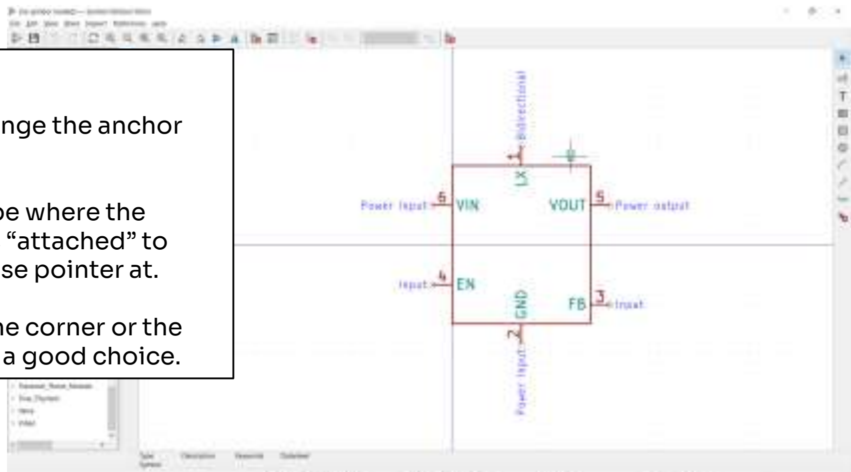
Creating a symbol for the IC

Great.

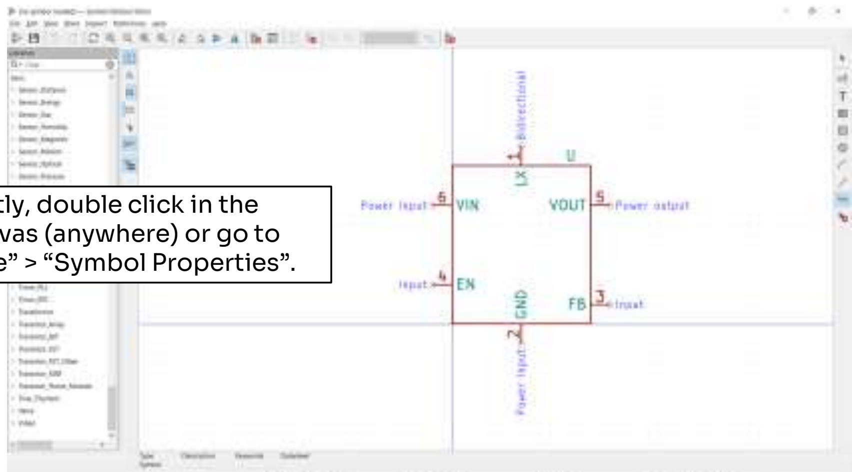
Next, change the anchor point.

This will be where the symbol is “attached” to your mouse pointer at.

Usually the corner or the middle is a good choice.



Creating a symbol for the IC



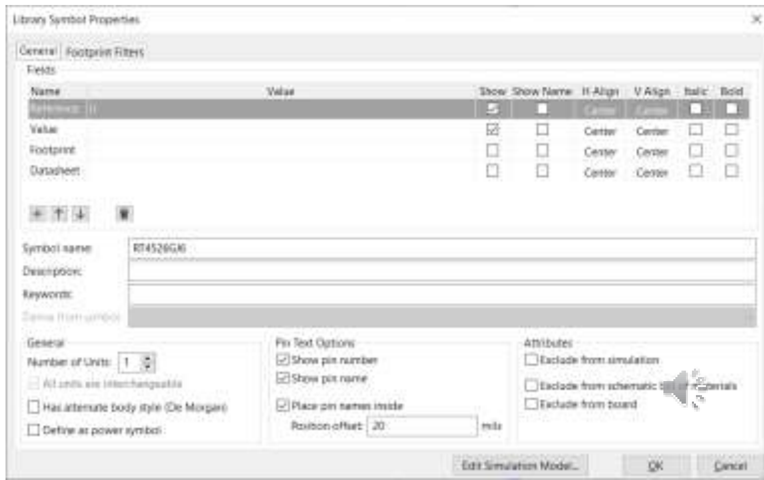
Lastly, double click in the canvas (anywhere) or go to “File” > “Symbol Properties”.

Creating a symbol for the IC

These windows often have a lot more options than we need right now, but they do offer a lot of flexibility.

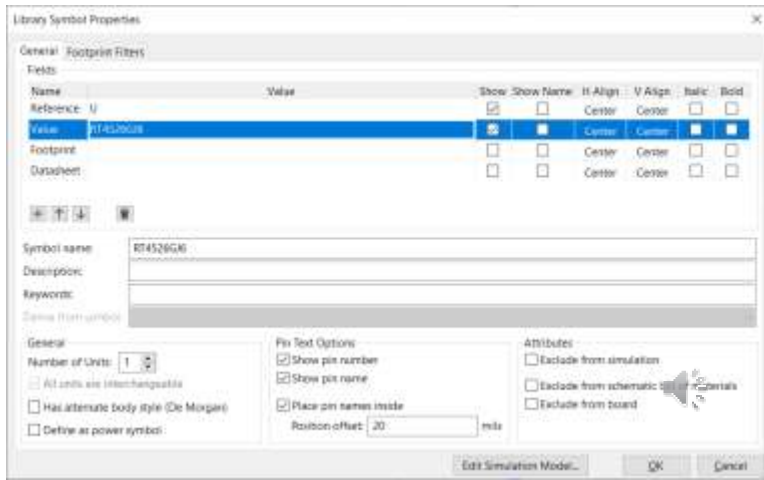
For now, just add the part number (RT4526GJ6) to the "Value" field's value.

Adding a footprint here can be done if there's a single footprint for this part. We don't have a footprint yet, so we won't add it, but later maybe.

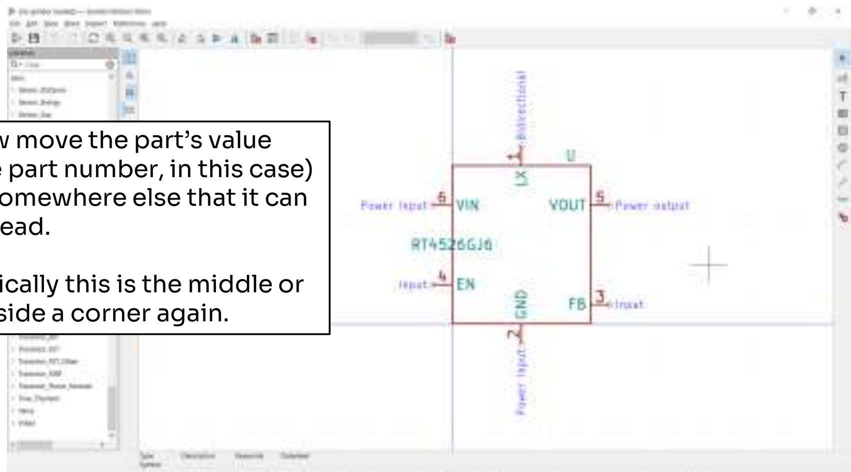


Creating a symbol for the IC

Click OK once your done.



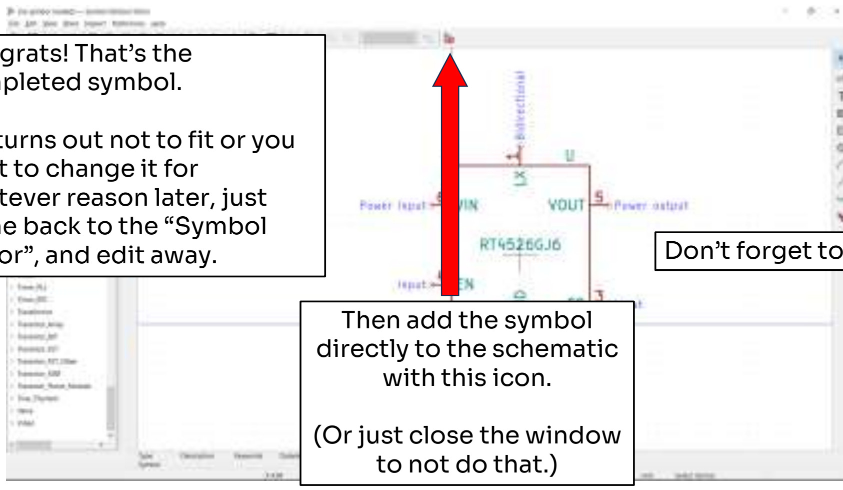
Creating a symbol for the IC



Now move the part's value (the part number, in this case) to somewhere else that it can be read.

Typically this is the middle or outside a corner again.

Creating a symbol for the IC



Congrats! That's the completed symbol.

If it turns out not to fit or you want to change it for whatever reason later, just come back to the "Symbol Editor", and edit away.

Don't forget to save!

Then add the symbol directly to the schematic with this icon.

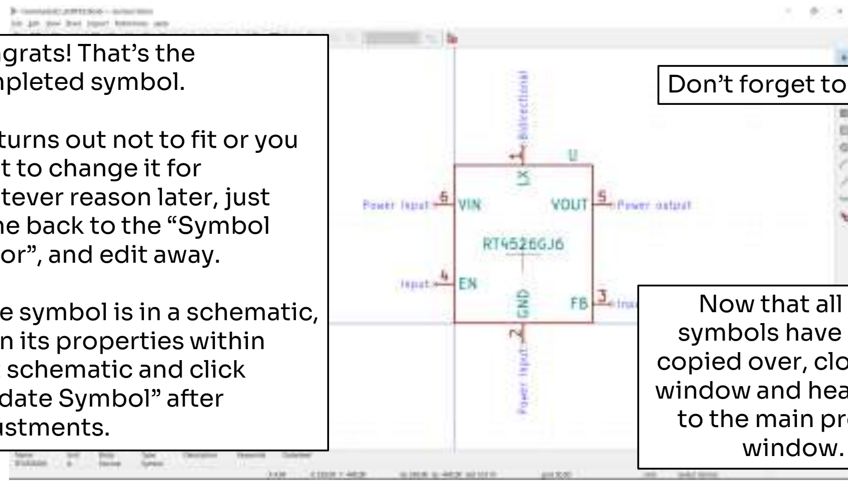
(Or just close the window to not do that.)

Symbol Library – Creating Models

Congrats! That's the completed symbol.

If it turns out not to fit or you want to change it for whatever reason later, just come back to the "Symbol Editor", and edit away.

If the symbol is in a schematic, open its properties within that schematic and click "Update Symbol" after adjustments.

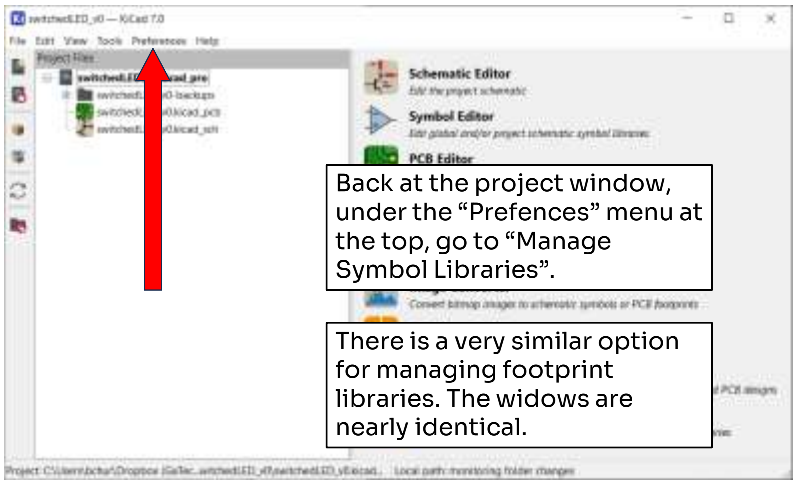


Don't forget to save!

Now that all the symbols have been copied over, close this window and head back to the main project window.

Now that all the symbols have been added to the library, we can close this window. When adding symbols to your schematic, now just use the symbols in this library instead of in the built-in libraries.

Symbol Library – Creating Models



Back at the project window, under the “Prefences” menu at the top, go to “Manage Symbol Libraries”.

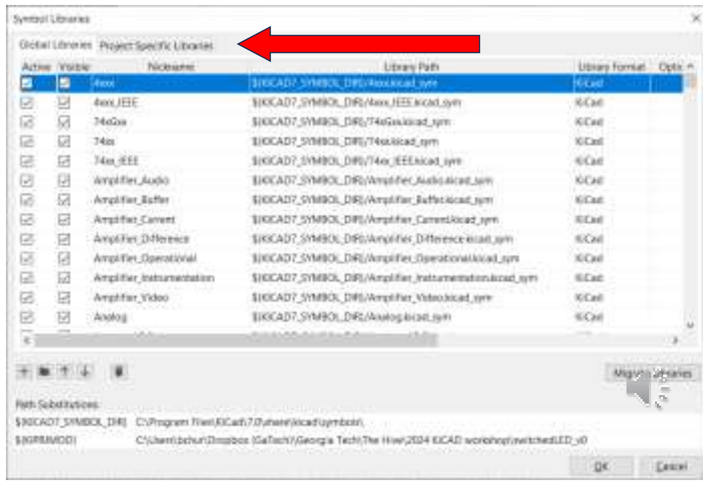
There is a very similar option for managing footprint libraries. The widows are nearly identical.

Symbol Library – Creating Models

The window that opens is where you can manage which libraries are active and visible.

I'm not sure if a library can be deactivated but still visible, but it can be active and invisible.

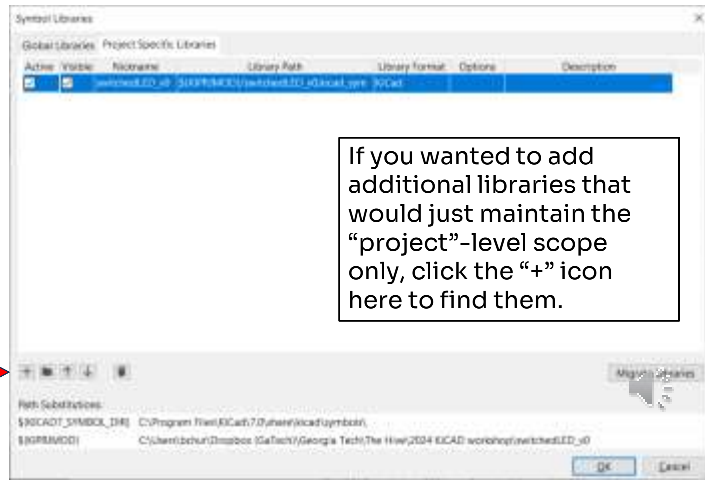
At the top, the two tabs let you switch between library scopes. Let's switch to "Project Specific Libraries".



Symbol Library – Creating Models

Conveniently, the library we just made is automatically added as both active and visible. Excellent – we can now use the symbols from within it.

Job well done.

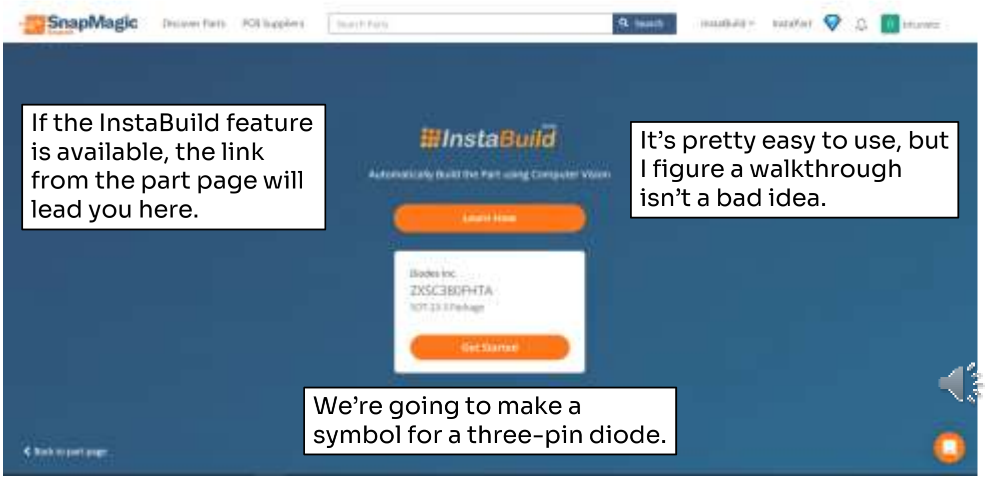


If you wanted to add additional libraries that would just maintain the “project”-level scope only, click the “+” icon here to find them.

SnapMagic's InstaBuild

- I'm going to spend the last few minutes demonstrating SnapMagic's InstaBuild symbol-generator.
- This is a browser-based "computer-vision-based" symbol generator.
- It's only available for symbols that specifically offer it on their parts page on the SnapMagic website.

Symbol Library – Creating Models

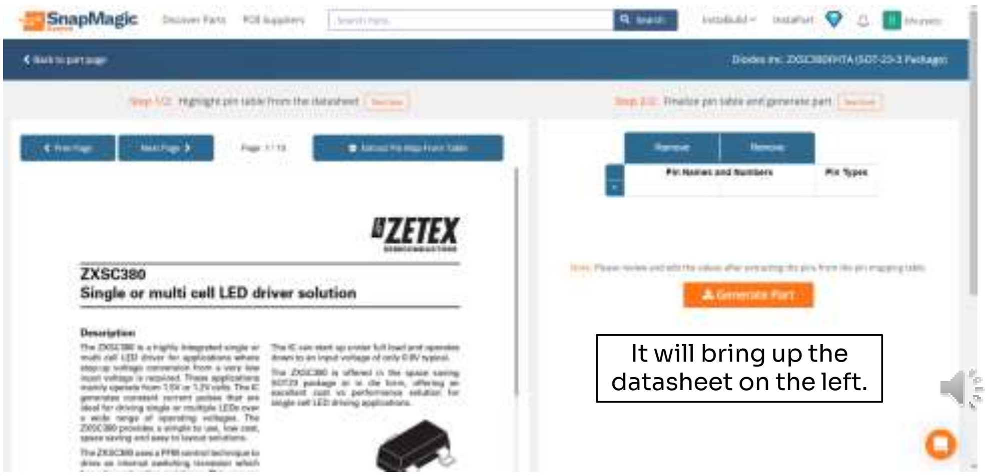


If the InstaBuild feature is available, the link from the part page will lead you here.

It's pretty easy to use, but I figure a walkthrough isn't a bad idea.

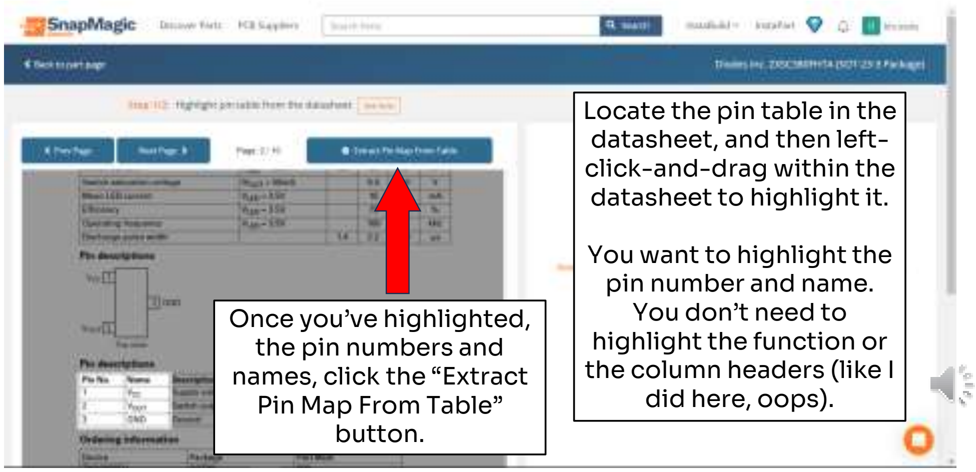
We're going to make a symbol for a three-pin diode.

Symbol Library – Creating Models

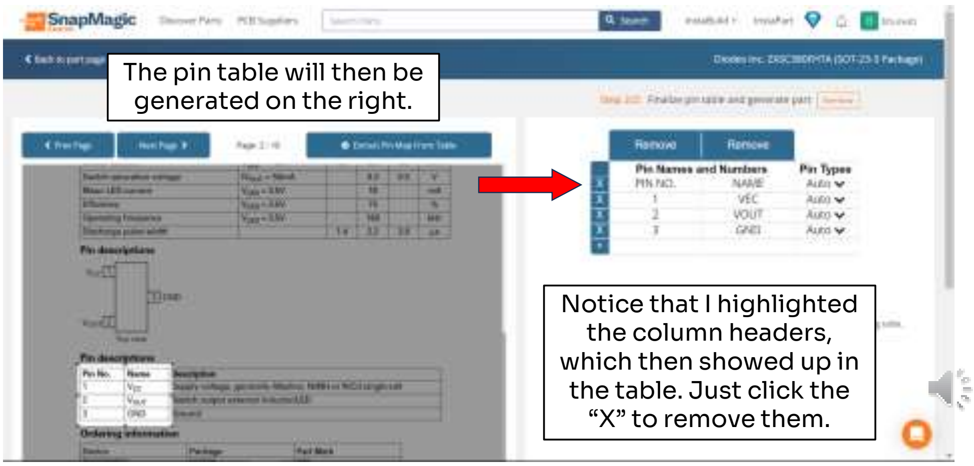


It will bring up the datasheet on the left.

Symbol Library – Creating Models



Symbol Library – Creating Models



The pin table will then be generated on the right.

Notice that I highlighted the column headers, which then showed up in the table. Just click the "X" to remove them.

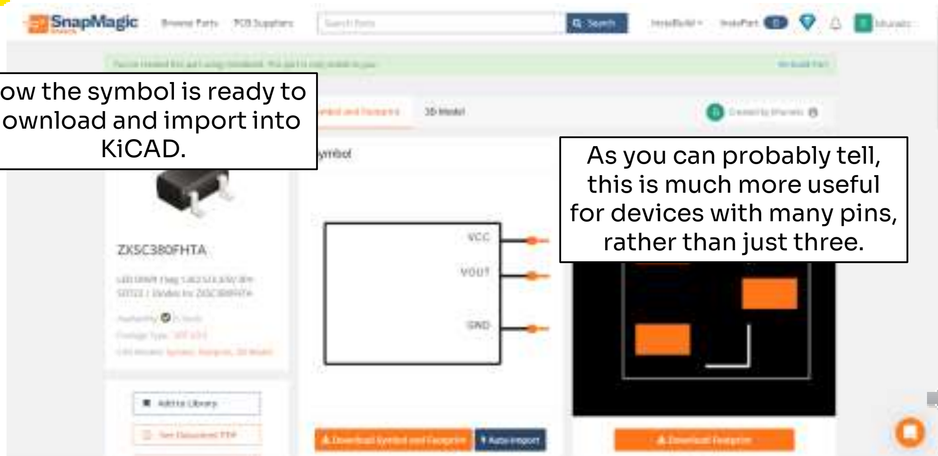
Symbol Library – Creating Models

Voilà!

Now just click “Generate Part” and the symbol will be generated.

Don’t expect anything fancy.

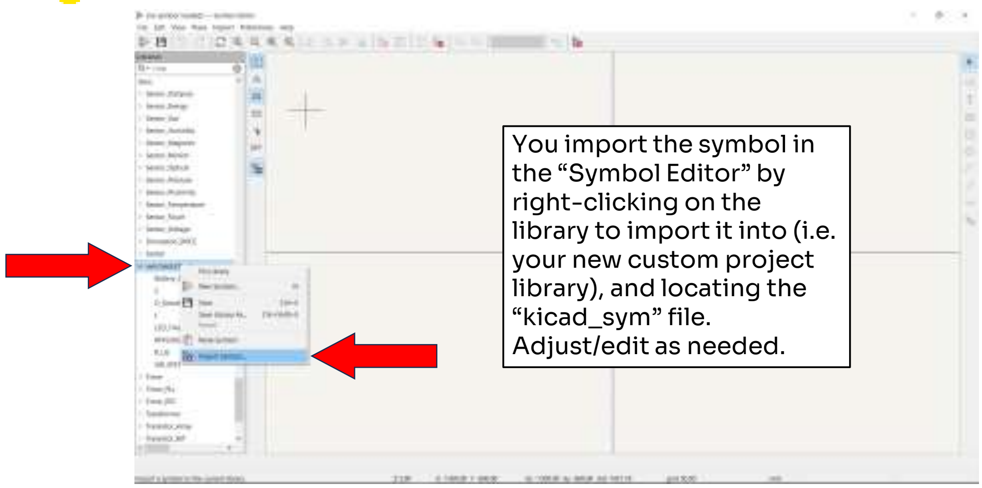
Symbol Library – Creating Models



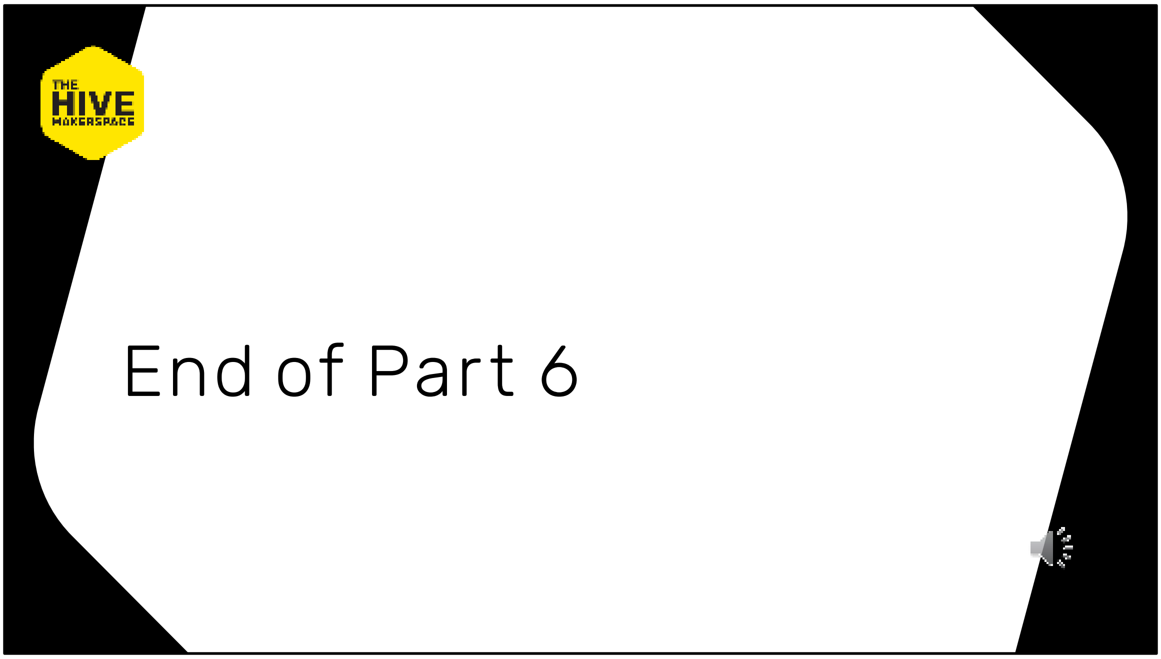
Now the symbol is ready to download and import into KiCAD.

As you can probably tell, this is much more useful for devices with many pins, rather than just three.

Symbol Library – Creating Models



I explained this process earlier, but to import a downloaded symbol, right-click the library from within the Symbol Editor, and select “Import Symbol”. Then navigate to, and open, the “kicad_sym” symbol model file you downloaded. Edit and adjust as needed or wanted, and then save.



And that ends part 6 of this video series, in which we covered creating and filling our own project-specific library. A PDF of this video is available as well, linked in the description and hosted on The Hive's Wiki.

In the next video in the series, part 7A, we'll cover creating a footprint library and populating it with globally-available models from KiCAD's built-in libraries.

See you then!