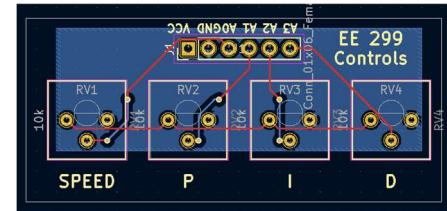
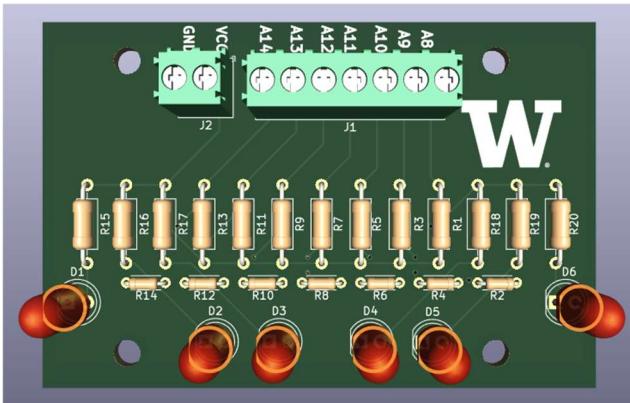


Introduction to PCB Design and Layout



Electrical and Computer Engineering 201

Introduction to PCB Design and Layout

A Schematic Capture and PCB Layout Tutorial in the KiCad software suite

What You Will Need

Materials:

- None

Machinery:

- Computer / Laptop

Software:

- KiCad Software Suite (Installed in Tutorial)

Overview

The goal of this tutorial is to develop a familiarity in Schematic Capture and Printed Circuit Board Layout (PCB) techniques. We will be using [KiCad](#), but skills developed in this tutorial can be applied to analogous Electronic Design Automation (EDA) software suites such as [Eagle](#), [Altium](#), and [Zuken](#) (a few commonly used EDA suites in industry). The process of PCB design can be categorized into two consecutive processes: Schematic Capture and PCB Layout.

This tutorial is split into four portions: **(1) Downloading Software, (2) Schematic Design, (3) PCB Layout, and (4) Export and Fabrication (Optional)**.

We will also cover how to export these designs into a format compatible with fabrication services, and some common places to order parts for your future EE endeavors and side projects!

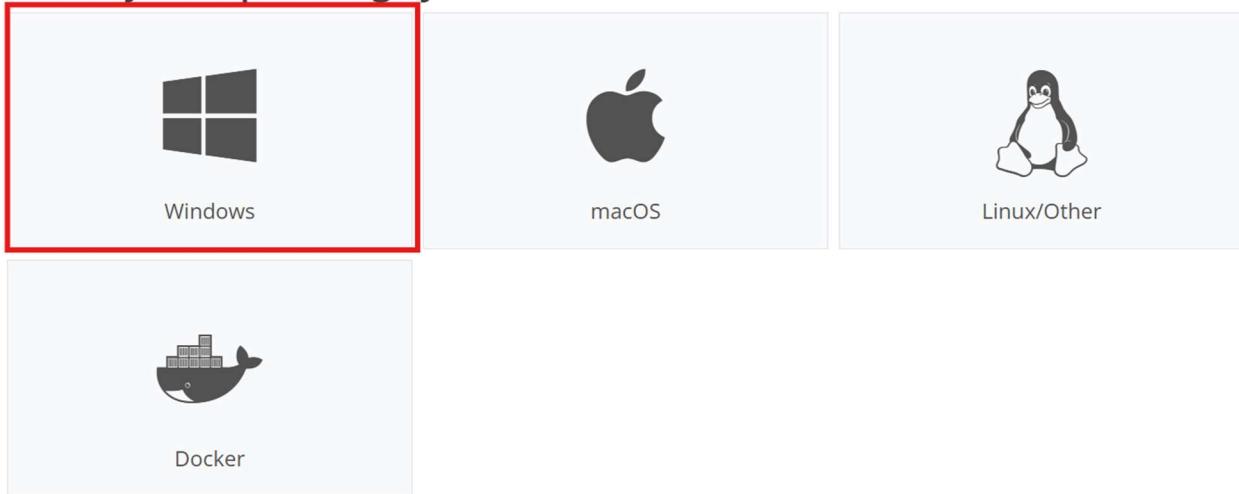
Note: This tutorial is modeled off of the [official KiCad tutorial](#), and is modified specific to the circuits in the line-following robot project. Feel free to refer to the official document if needed. Additionally, KiCad continues to update their software, so this tutorial may resemble newer versions but may not completely match future versions. This version of the KiCad download guide is for the most recent **version 8.0.3**.

Part 1: Download KiCad

1. Go to <https://www.kicad.org/download/>.
2. Select the download package corresponding to your operating system. In this tutorial, we will be using Windows, which may not be specific to you.

Download

Select your operating system

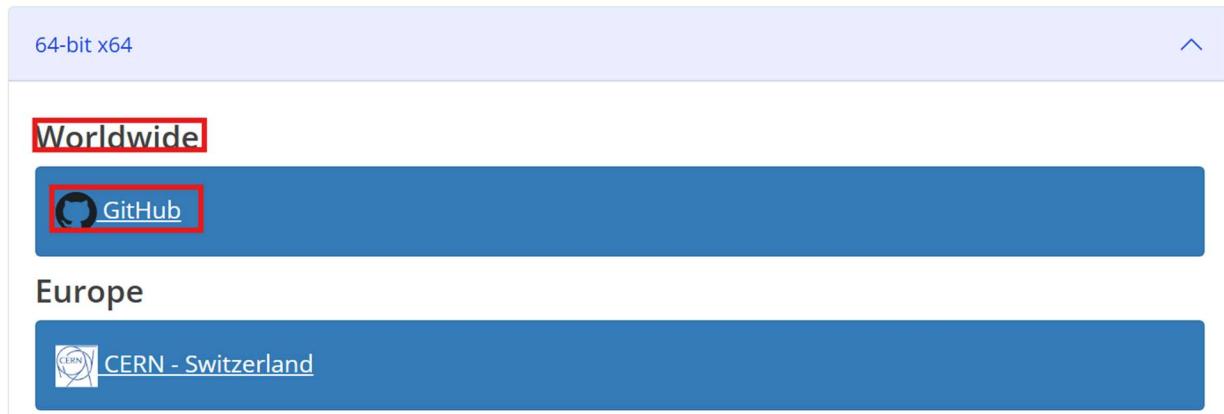


3. You will then be led to a page with specific instructions on downloading. If your operating system is Mac or Windows, you should choose the global release. This will be under “Stable Release” → “Worldwide” → “GitHub.”

Stable Release

Current Version: 8.0.3

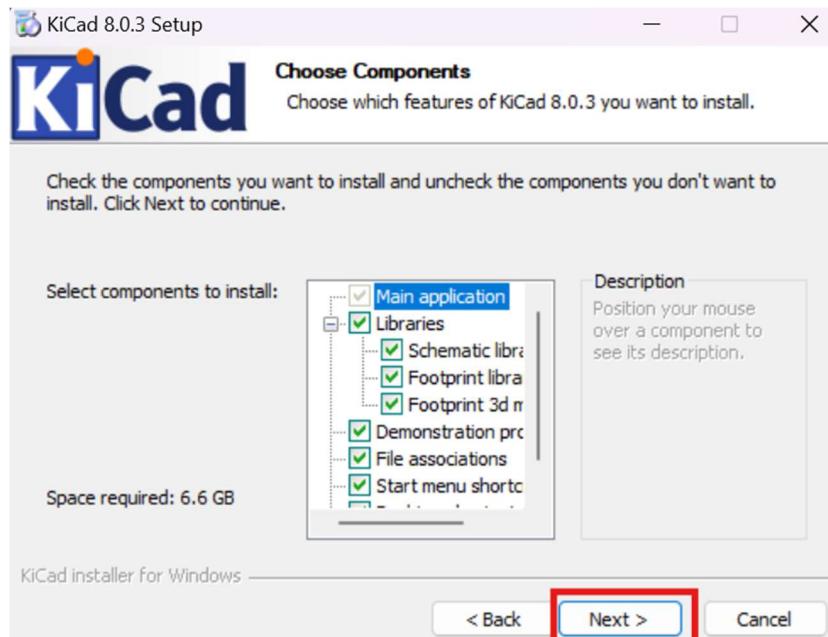
KiCad is available for Windows on the x86-64-bit and arm64 platforms.



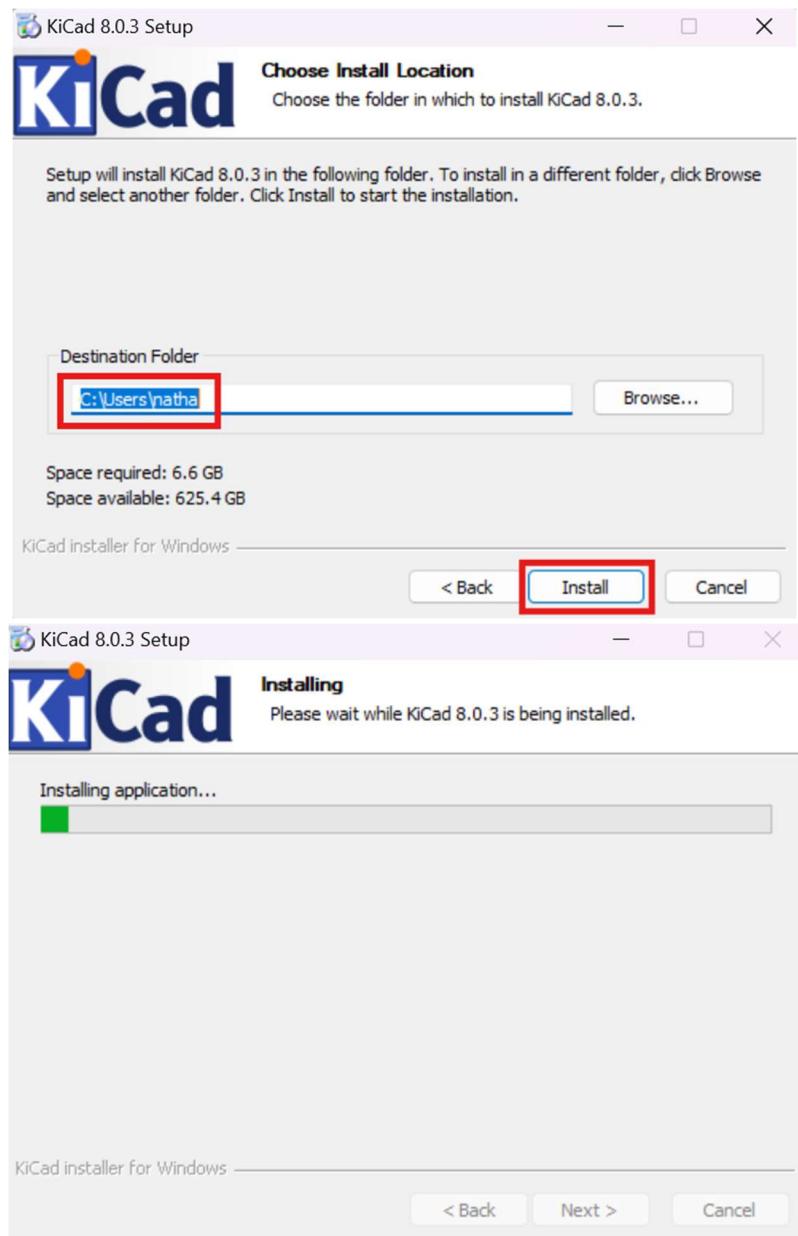
- Clicking GitHub will prompt a download suite. Click on the download prompt, “**kicad-8.0.3-x86_64.exe.**”
- Allow “**Installer for KiCad EDA Suite**” to make changes to your computer by clicking “**Yes.**” This should prompt the installer to begin installing KiCad on your device.
- The KiCad Installer Setup will prompt this screen. Click “**Next.**”



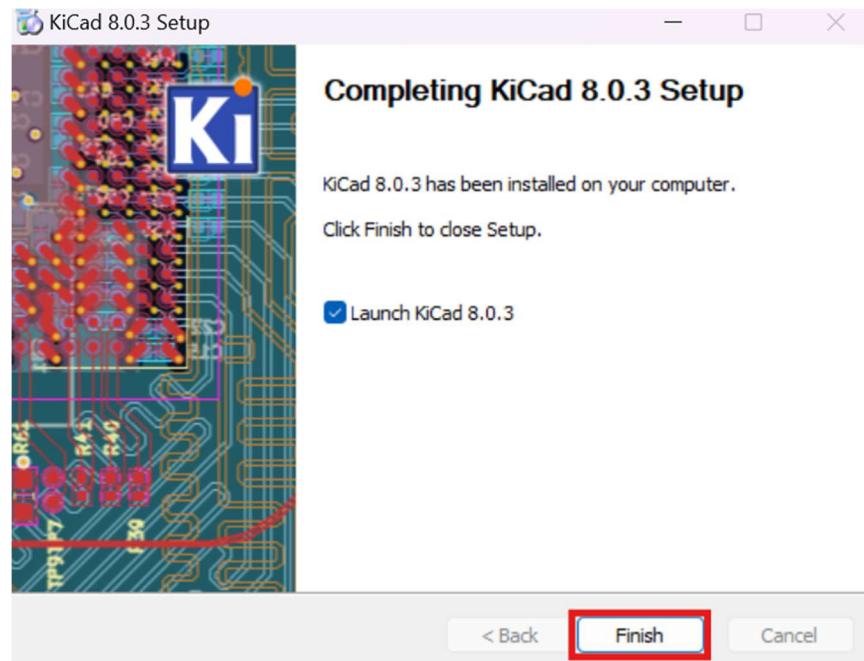
- Next is a screen with recommended components to install. Click “**Next.**”



- Next is a screen with a chosen install location. Make sure the destination is somewhere within your Program Files. Click “**Install.**” Your device will begin installing KiCad. Note, if C++ is not installed on your computer, a C++ installation will run before the KiCad installation. The C++ installer is necessary and supposed to be there!



9. Next is a screen prompting you to finish completing KiCad setup. Click “Finish.”



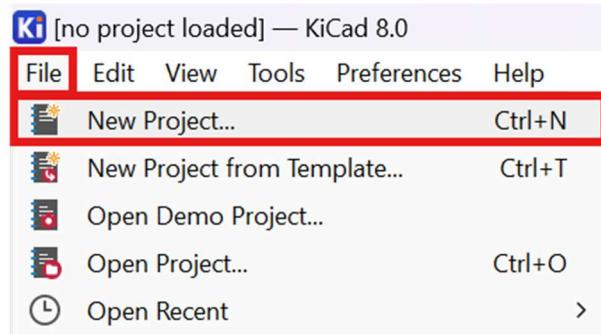
Part 2: Designing the Photoresistor Sensor

The goal of this section is to familiarize yourself with Eschema, the schematic capture software in KiCad. We will be creating a very basic circuit for our photoresistor sensor array.

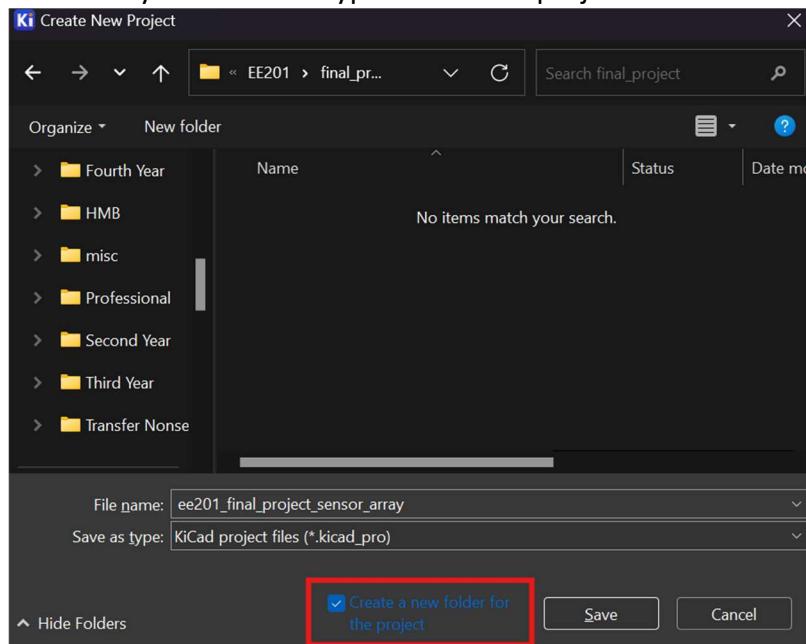
Note: Please find a mouse with a scroll wheel before you attempt this tutorial. It will be very hard to place components without a mouse. You can often find them lying around in EE 137

Step 1 Schematic Capture in Eschema:

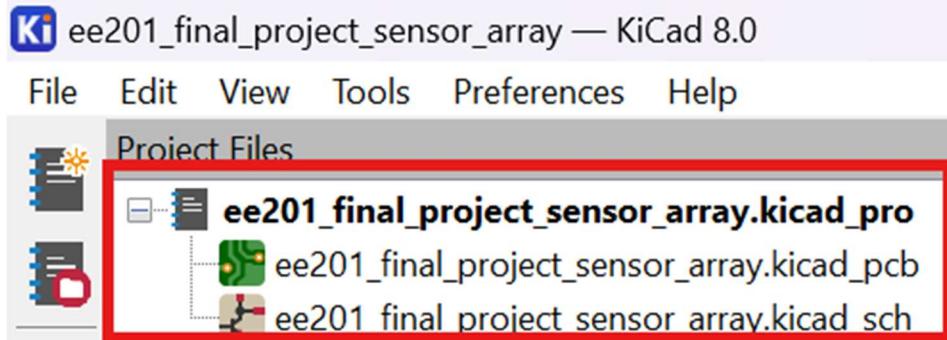
1. With KiCad downloaded, open the app on your device, on the upper ribbon, go to “File” -> “New project.”



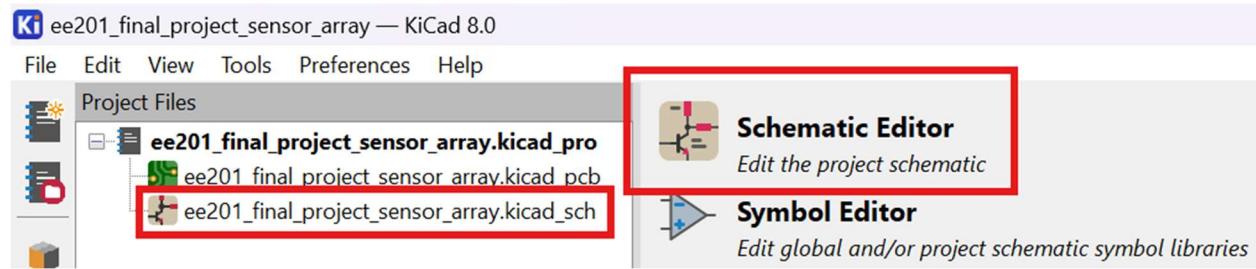
Type in your file name, and click “Save.” Make sure the “Create a new folder for the project” box is checked before saving your project. Creating a project folder will help keep your work organized as there are many files and file types in a KiCad project.



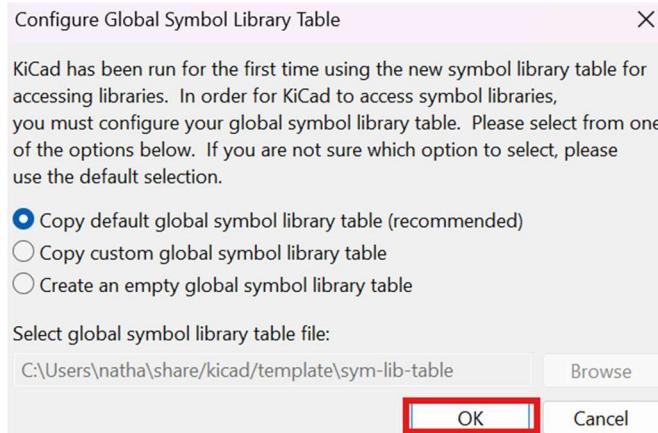
You should see the new project you just created in the “**Project Files**.”



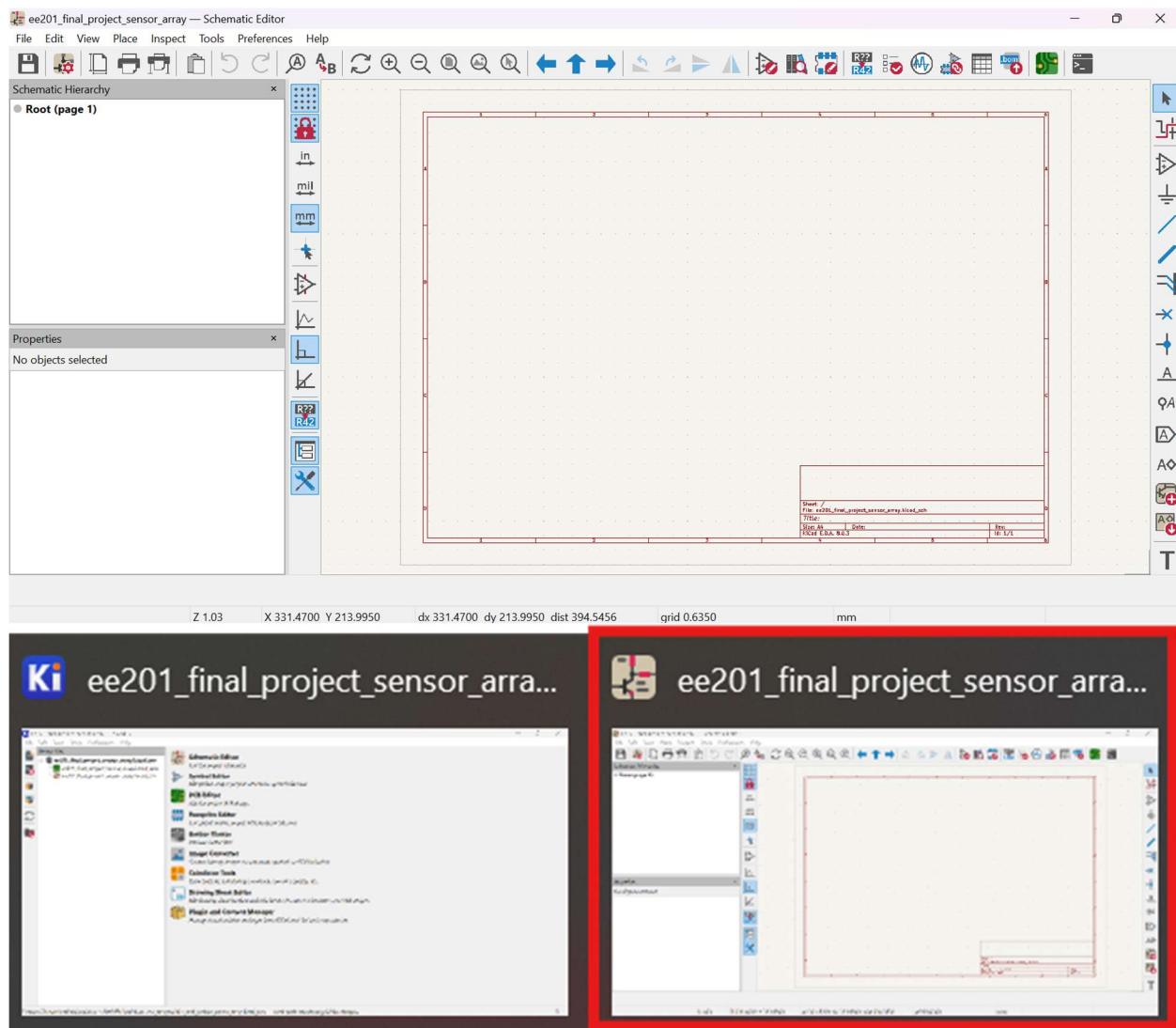
Double click the “**Schematic Editor**” or the “**.kicad_sch**” file to edit.



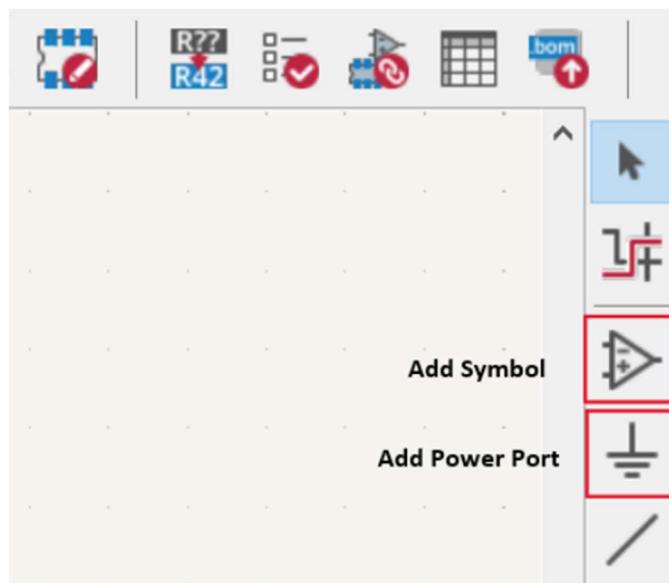
Click “**OK**” if you see the following pop-up window. Keeping the default symbol library will save you time searching for symbols in the future.



A blank schematic should open in a new window. If you do not see the schematic, try hovering over the KiCad logo on your taskbar and clicking on the schematic window.

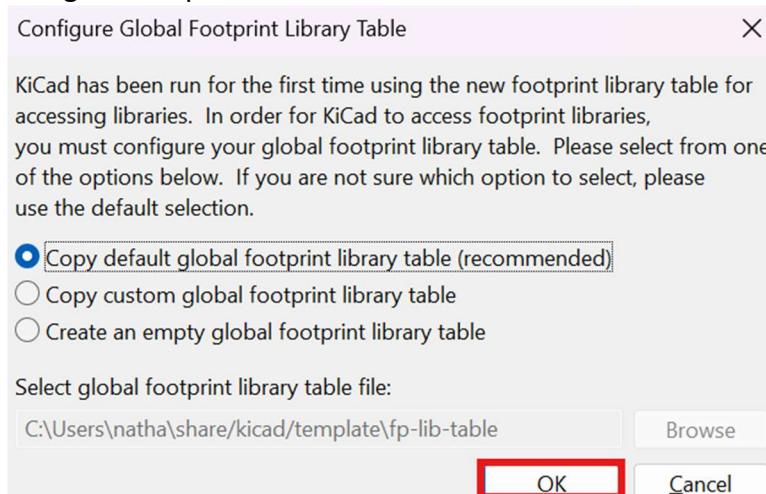


2. Next, refer to the right side bar of your screen to select “Add Symbol.”

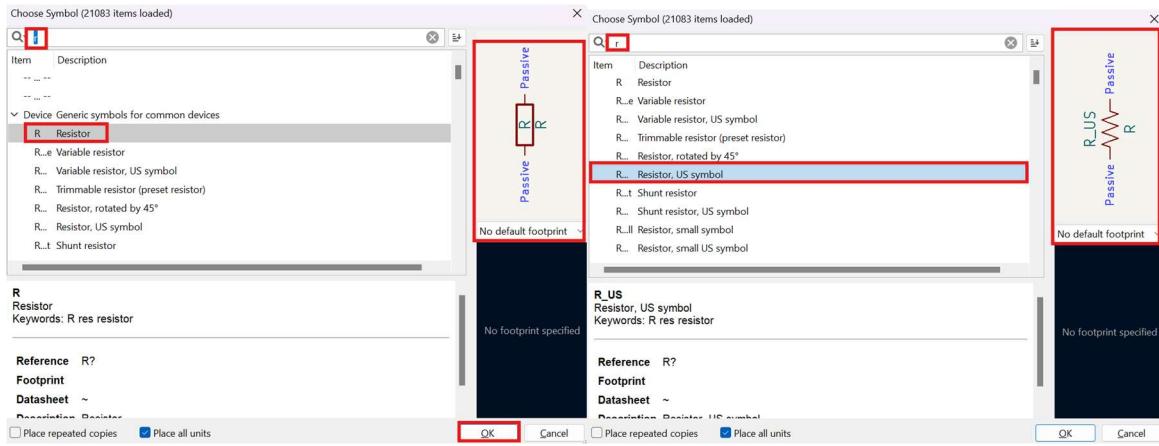


Note: You can use the keyboard shortcut "A" for Add Symbol and "P" for Add Power Port.

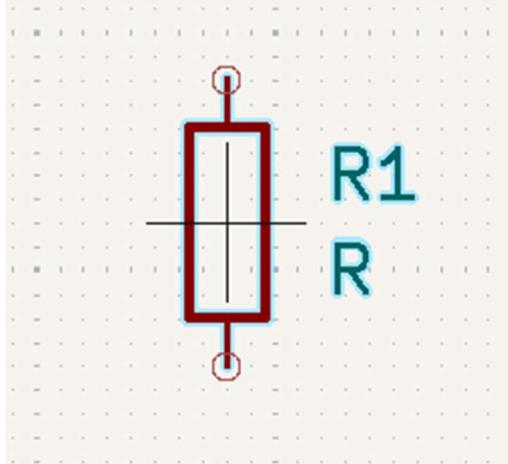
Click “OK” if you see the following pop-up window. Keeping the default footprint library will save you time searching for footprints in the future.



3. You will then be prompted to use the Symbol Library. Type “R” for Resistor. Click “OK” to begin placing a resistor. There are many different resistor symbols you can use, but we will be using the KiCad default resistor symbol (the one that looks like a box with two lines coming out). Some students prefer using the US resistor symbol (the squiggly line most students are familiar with). The resistor symbol chosen does not matter as long as it only has two connection points, or wires going in or out of it.

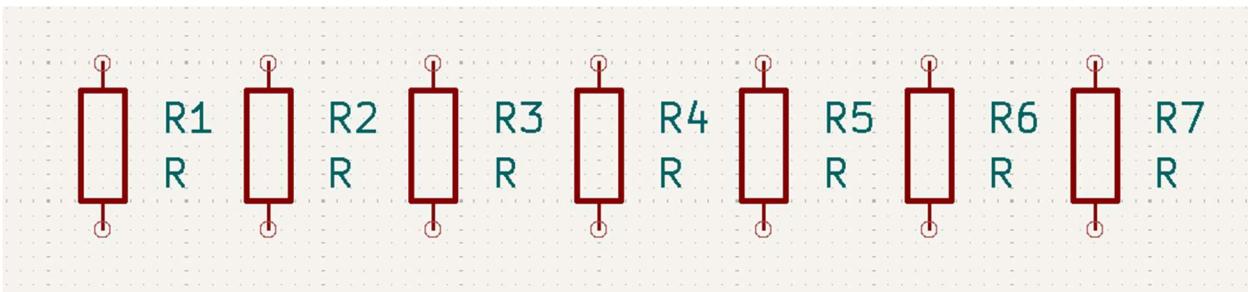


- 4.** Your Resistor will now appear on the schematic and highlighted in blue. Move the resistor by clicking on the component follower and place it in the desired location (somewhere in the middle-ish of the schematic for now).



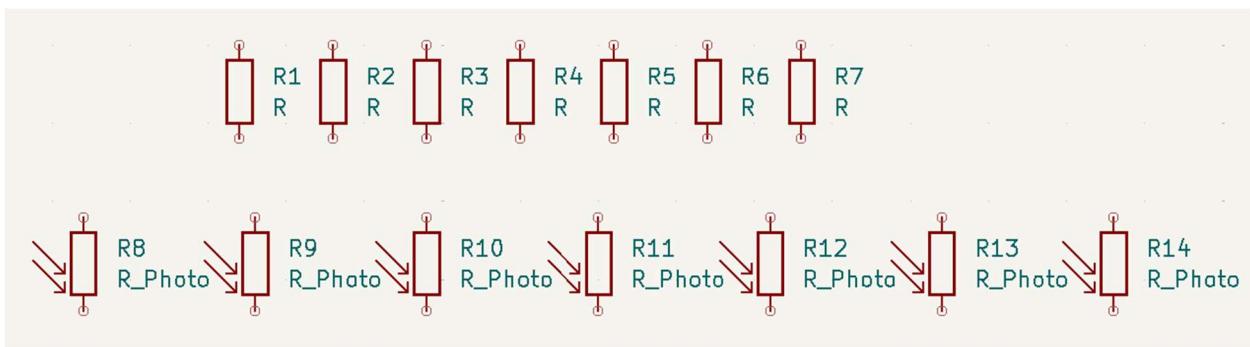
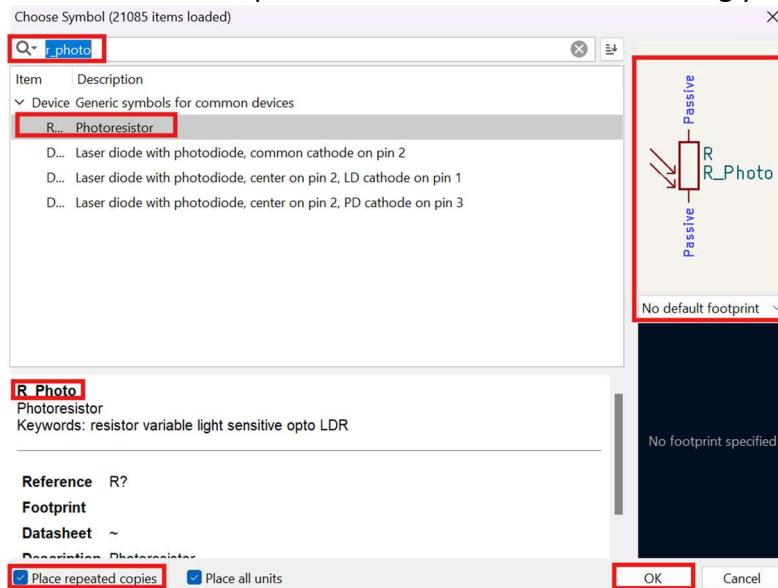
*Note: You can **zoom in** using the scroll wheel to see the symbol better or move the resistor after you've placed it by selecting the resistor and dragging it or using the keyboard shortcut “M.”*

- 5.** Instead of finding the symbol in the “**Add Symbol**” again, select the resistor and use the keyboard shortcuts for copy and paste (“**Ctrl**” + “**C**” to copy and “**Ctrl**” + “**V**” to paste) to place a new resistor adjacent to the original. You should do this for six additional resistors for a total of seven resistors.

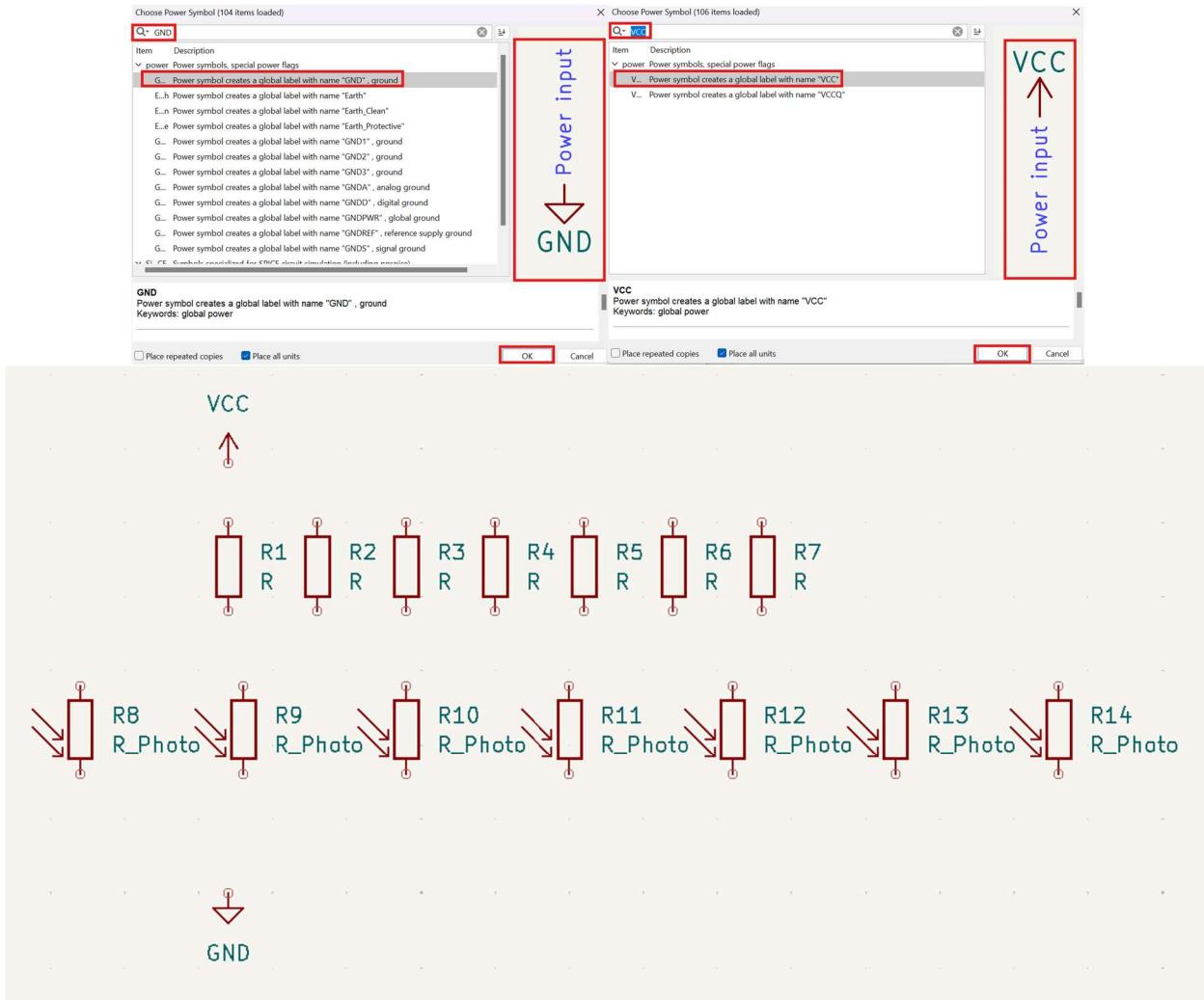


Note: You can also select “Place repeated copies” on the “Add Symbol” menu if multiple of the same component need to be placed. To end the placement (keeping everything placed already), use the keyboard shortcut “Esc” or “Right Click” and click “cancel.”

- After placing these resistors, click “Add Symbol” again and Search “R_Photo” to place a photoresistor symbol. Copy and paste this symbol (or use “Place repeated copies”) to make seven total photoresistors. Place the photoresistors near but not touching your resistors.

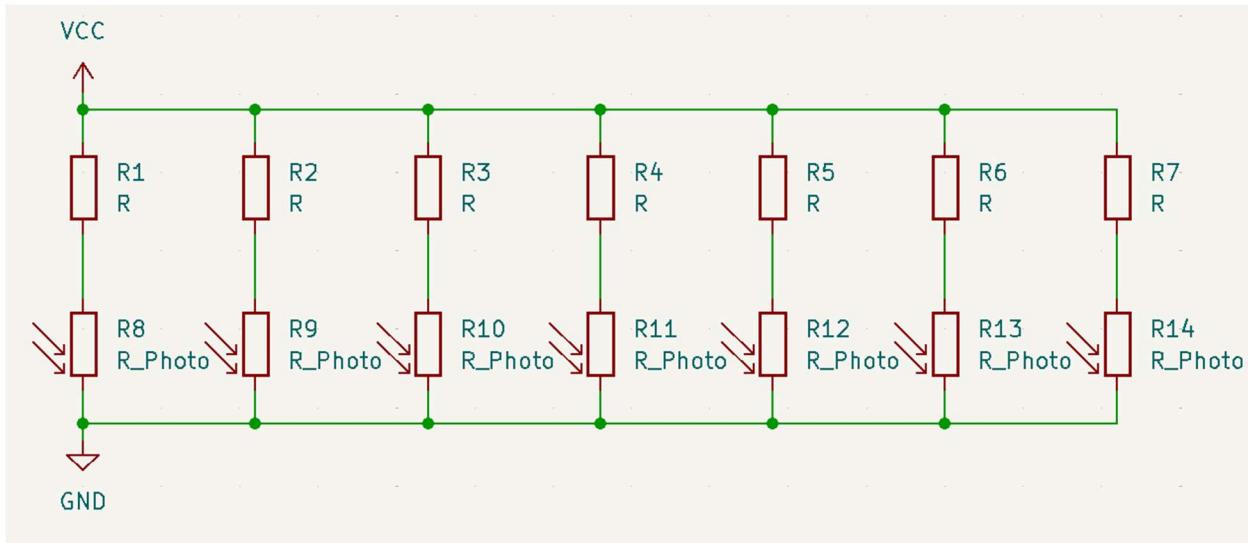


- Next, click the “Add Power Symbol” Icon on the right sidebar, prompting another symbol library. Search “GND” and place a ground symbol directly under the photoresistors. Open up the “Add Power Symbol” prompt again and search “VCC” and place a power symbol above your resistors.



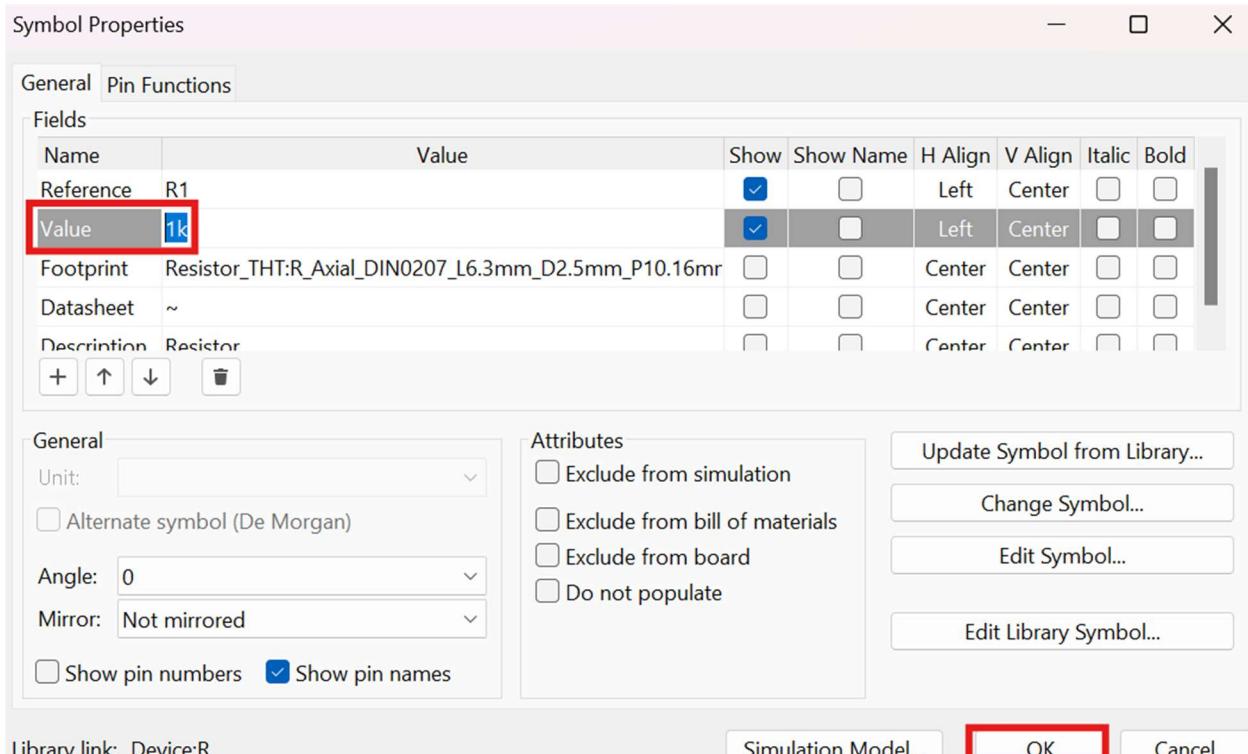
Note: These are considered labels, meaning all areas connected to a "VCC" symbol will be considered connected by the software and all areas connected to a "GND" symbol will be considered connected by the software.

- Once you have all of your symbols placed, select the wire symbol, (or use the keyboard shortcut "W"), and connect all of the components by clicking any area of the screen and guiding each connection with your mouse. Emulate the image below.



Note: You can tell a wire is connected to a component because the small, empty circle (⊕) attached to the component will be removed and replaced with a wire connection (|). If a wire is connected to another wire instead of a component, a green circle will appear at the wire connection (●).

9. In our current schematic, our resistors do not have any resistance value assigned to them; they are all of resistance value “R.” We need to assign the resistors the correct resistance of $1,000\Omega$ (or $1k\Omega$). There are several ways to edit passive component values. The first is to **“Double Click”** on the component to open its properties. Change the **“Value”** field to 1k, 1000, or 1,000. The software recognizes all three as the same value. No units are necessary as the software assumes the units for passive components such as resistors, inductors, and capacitors.



Note: You can also use the keyboard shortcut “E” to open a component’s symbol properties when it is highlighted.

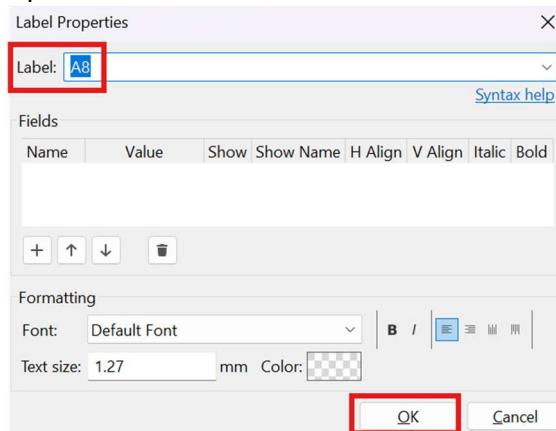
10. Editing each component individually is helpful when you need to change one value, but slow when you need to change multiple components. To edit multiple components at the same time, on the upper ribbon, go to “Tools” → “Edit Symbol Fields...” to open the “Symbol Fields Table.” Change the “Value” field of the “R2-R7” row to 1k. This will change the value (resistance) of resistors R2-R7 to 1k. If you want to change the value of the resistors individually, select the drop-down arrow (“v”) in the “R2-R7” box to display each resistor individually. You can change their resistances by editing their “Value” field.

	R1-R7	1k	~	7
R1	1k	~	1	
R2	1k	~	1	
R3	1k	~	1	
R4	1k	~	1	
R5	1k	~	1	
R6	1k	~	1	
R7	1k	~	1	

The screenshot shows the Altium Designer ribbon with the "Tools" menu selected. A red box highlights the "Edit Symbol Fields..." option under the Tools menu. The "Edit Symbol Fields..." option is also highlighted with a red box in the context menu below the table.

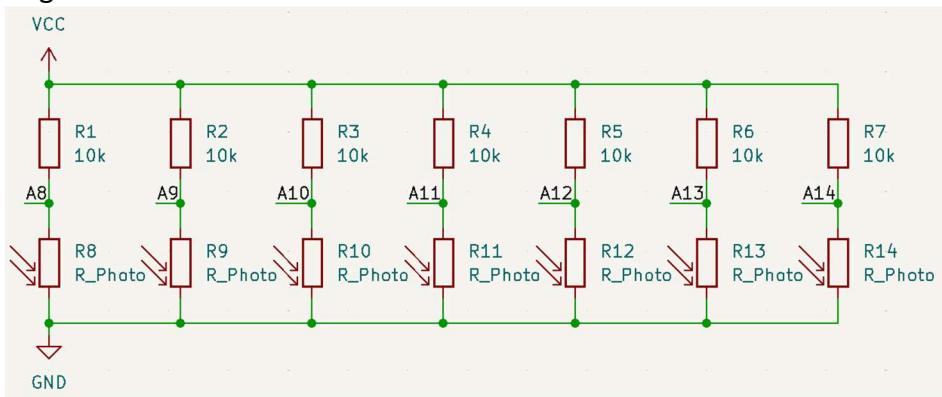
Note: The “Qty” (or quantity) field tells you how many components are in that “Reference.” For the “R2-R7” reference, there are six components. This means that changing the “Value” of the “R2-R7” reference will change the value of six components (R2, R3, R4, R5, R6, and R7).

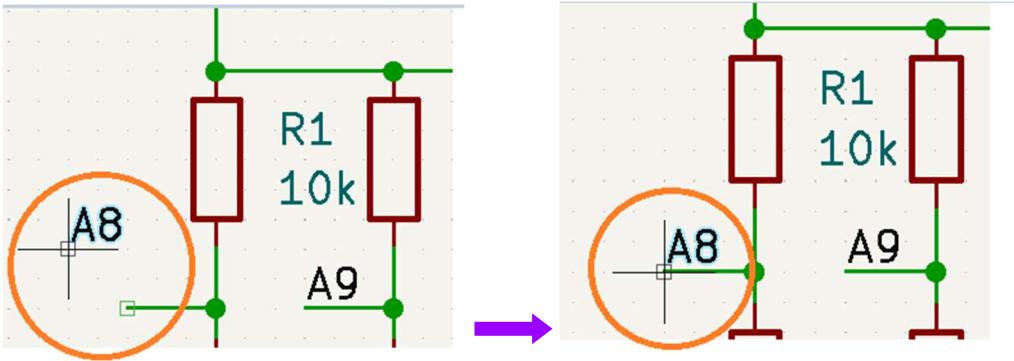
11. Now that the resistors have the correct resistances, we can add the points where we would connect our jumper wires. It is good practice to keep a tidy schematic instead of having wires crossing often. To help keep our schematic readable, we will use “Labels” to mark where our jumper wires will go. Use the “Add Label” button () on the right sidebar to begin adding a label. Type “A8” in the “Label” field. We will be numbering our labels A8-A14 to indicate their corresponding analog input pin on the Arduino.



Note: You can use the keyboard shortcut “L” to create a label as well.

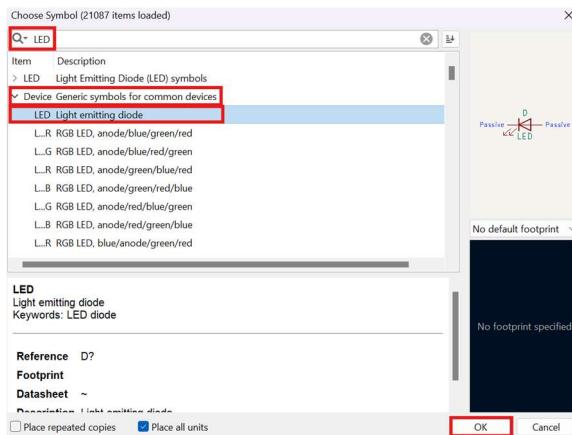
Emulate this figure below.

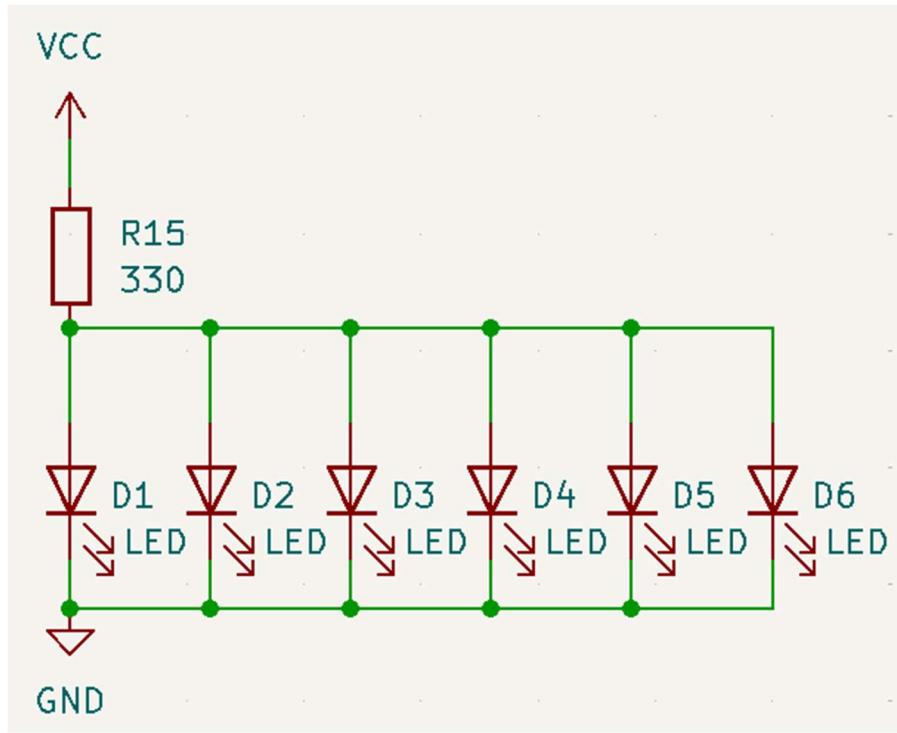




Note: Labels do not display any visual indication that they are connected to a wire, so it is good practice to add another wire connecting the label to the desired location to create a green circle (●) showing visually that the label is connected. You can rotate a label using the keyboard shortcut “R” to adjust its orientation.

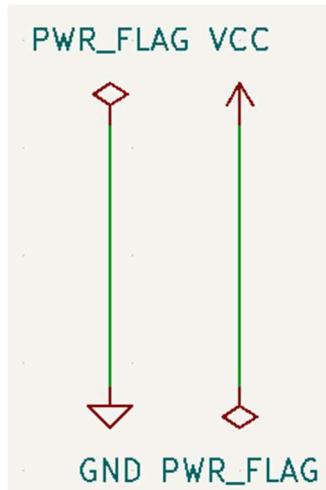
11b (optional). If you are choosing to use a light shield in your project (which is highly recommended), use what you have learned in steps 1-10 to recreate the schematic below. Use the generic “Light emitting diode” component for your LEDs or an equivalent component. Create the light shield schematic in the same KiCad schematic as your photoresistor schematic. Place the components far enough from the photoresistor components that your schematics will not overlap.





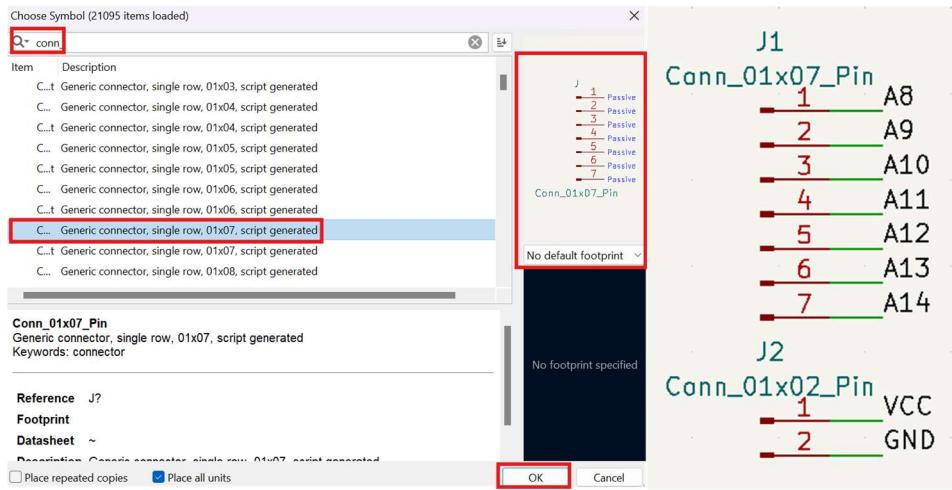
Light Shield LEDs

12. Now that we have placed all our components, it is important to indicate to the software that power will be entering (through “**VCC**”) and leaving (through “**GND**”) our circuit. We indicate to the software that power will be entering or leaving our circuit by connecting the **PWR_FLAG** label to a power flag. In order to do this, select the power symbols library again () and search for “**PWR_FLAG**.” Emulate the connections in the figure below.



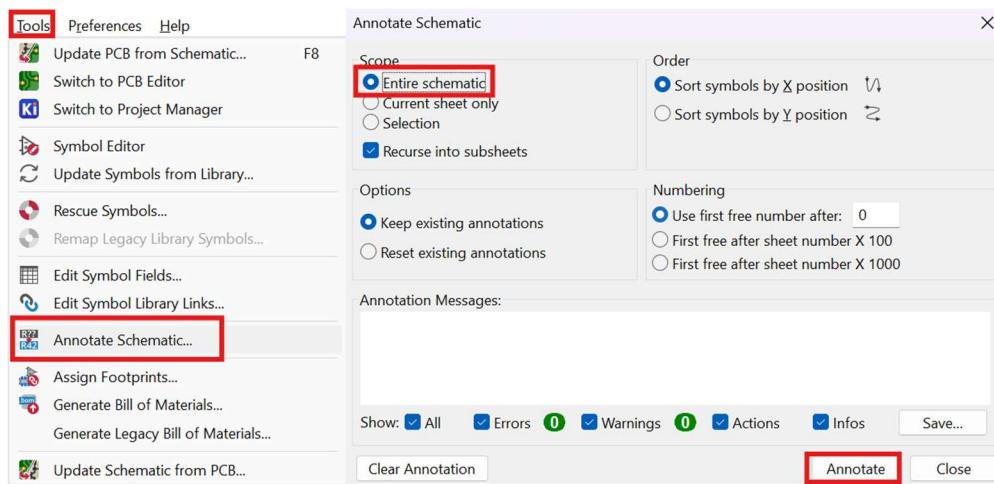
*Note: There are specific power flags available for “**VCC**” and “**GND**,” but we are choosing to use the generic power flags that function for any incoming or outgoing power.*

13. Now that we have all our components placed and labeled, we need to add the connectors that connect our circuit to our Arduino. This tutorial will use female headers (that accept male-jumper wires), but you may choose any other connector (such as screw, male, or any other) that benefits your project. In the component symbol library, search for “**Conn_**” to bring up the list of header pin connectors. Select the “**Conn_01x07_Pin**” component (NOT the “**Conn_01x07_Socket**” component) and place it on schematic. The “**01**” corresponds to the number of columns, the “**07**” corresponds to the number of rows, and the “**Pin**” corresponds to the type of connector. The 01x07 connector will be used to connect our photoresistor labels A8-A14 from the circuit to the Arduino. Also add a “**Conn_01x02_Pin**” to connect our “**VCC**” and “**GND**” labels from our circuit to the Arduino. Connect all labels to their corresponding connector.

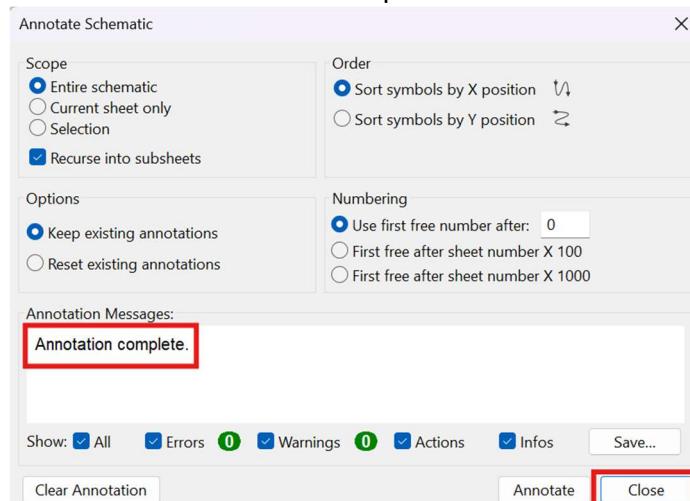


Note: The labels could be connected directly to the connectors, but wires are added to visually show a connection.

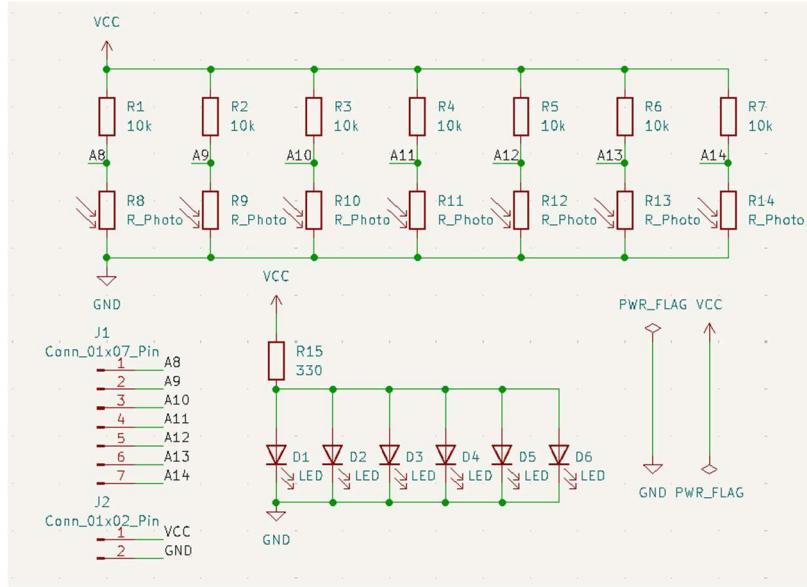
14. Even though our schematic is fully annotated, meaning every component is assigned a number such as R1, D3, or J2, it is good practice to have the software check your annotations in case a component annotation was missed. On the upper ribbon, go to “**Tools**” → “**Annotate Schematic**” (R2). Click “**Annotate**” to begin the automated annotation. Filling symbol reference designators (annotating) will allow your PCB to link schematic symbols to components on the board laying out a PCB.



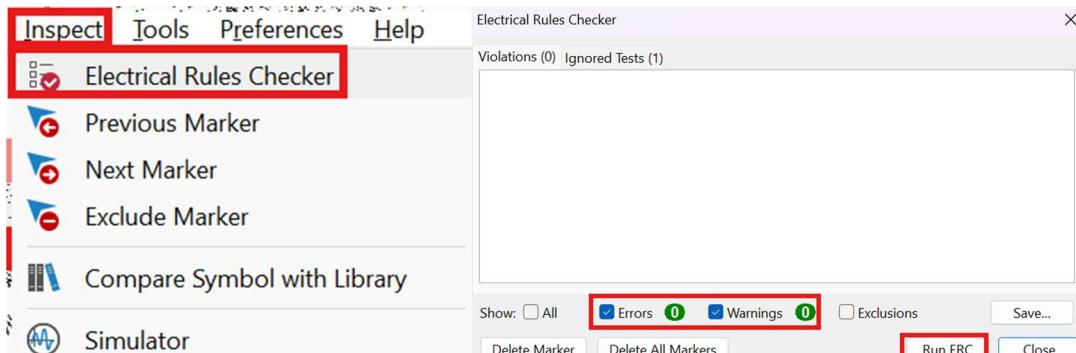
If everything goes well, you will see an “**Annotation complete.**” message. Because our schematic was already fully annotated, no other annotation messages were presented. If any components were to be annotated, the software would include a message indicating the change. Select “**Close**” once the annotation is complete with no errors or warnings.



The final schematic should resemble this:



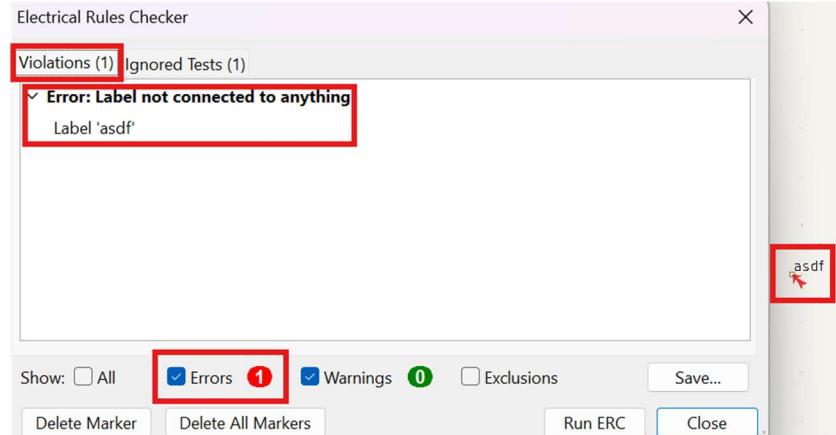
15. With everything annotated, the software can check our schematic for electrical errors. The most common electrical errors are missing connections, so it is incredibly important to run the “Electrical Rules Checker” (or “ERC”) before moving on. On the upper ribbon, go to “Inspect” → “Electrical Rules Checker.” Make sure the “Errors” and “Warnings” boxes are checked, and click “Run ERC.”



*Note: If there are no errors or warnings, there will be no visual indication the ERC has been run.
If you are unsure the ERC has run, run it again!*

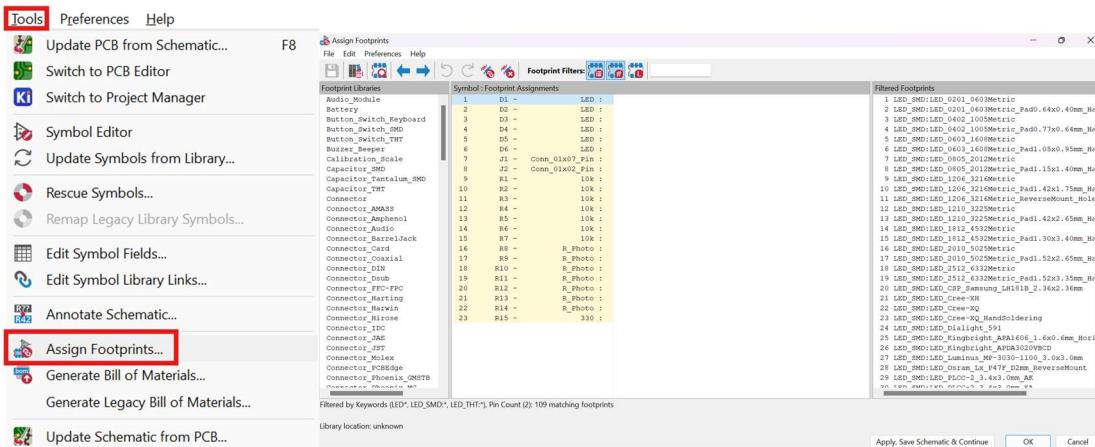
Our circuit did not have any errors or warnings, so it is good to move on! **This does not mean our circuit is correct, it simply means it is complete and does not violate the laws of physics.**

Solving ERC errors is often an iterative and time-consuming process. Errors in the ERC will give a message describing them and an indicator () will show up on the schematic. Errors should always be solved before laying out a PCB, but some warnings may be ignored. **Solve all errors and warnings before moving on with this tutorial.**

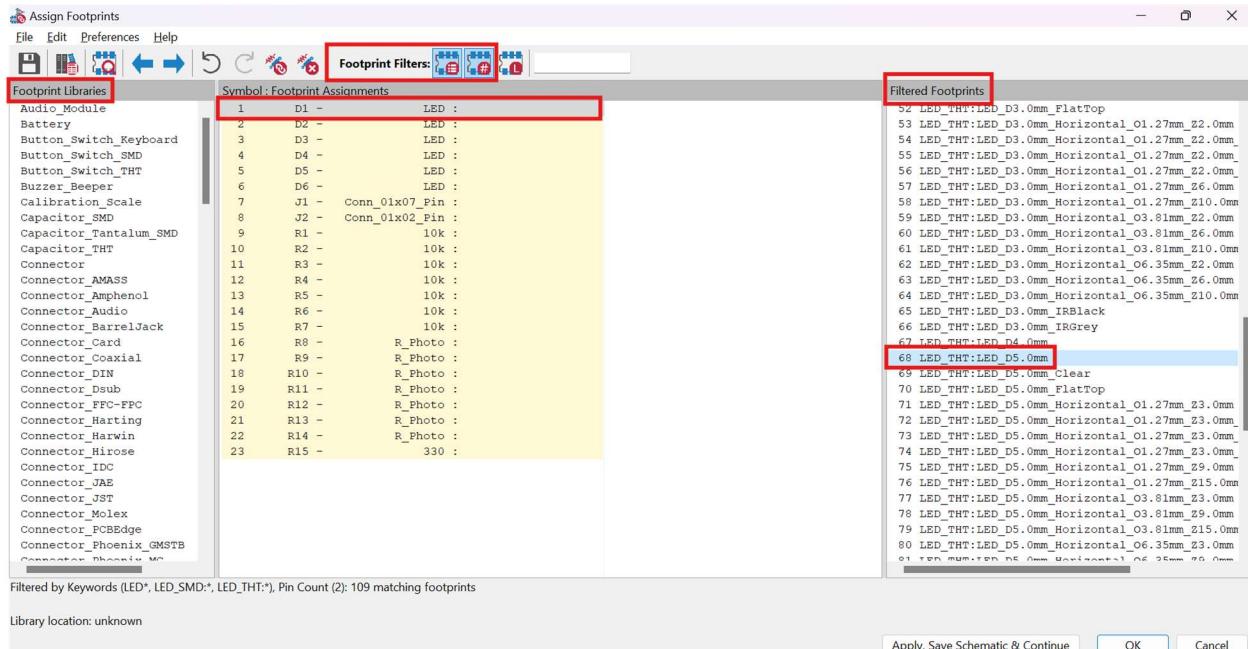


*Note: In this example, “**asdf**” is not connected to anything on the schematic. Resolve this error by deleting the label or connecting it to something. Rerun the ERC to make sure you solved the error.*

16. In order to begin placing components on a circuit board, we need to tell the software what each component will be. The software sees each component as a “**Footprint**” when placing it. The assigned footprint tells the software what modifications are needed to the circuit board to attach the component. The footprint will contain details on the pads, silkscreen (text printed on the board), and any masking or other details that need to be done. In the upper ribbon, go to “Tools” → “Assign Footprints...” (). Components highlighted in yellow still need footprints assigned to them.



To assign a footprint, first select a component by clicking on it (which should select it in a blue color). Make sure the “**Use symbol footprint filter**” and “**Filter by pin count**” options are selected in the “**Footprint Filters:**” section. Selecting these filters will allow the software to show you possible footprints based on the component you selected rather than having to search for the footprints manually in the “**Footprints Library**” section. The filtered footprints will show up under the “**Filtered Footprints**” section.



Note: The first “**Symbol**” (component), “**D1**,” will be assigned “**Filtered Footprint**” number “**68**,” “**LED_THT:LED_D5.0mm**.” In this footprint, “**THT**” means through-hole technology, or that the LED will have legs that need a physical hole in the PCB to be soldered. The “**D5.0mm**” means the LED has a diameter of 5.0mm. Information on the left side of the “**:**” details the mounting type and information on the right side of the “**:**” details the component.

To assign a footprint to a selected component, double click on the footprint or use the keyboard shortcut “**Enter**.” When the component has a footprint assigned to it, it will turn white.

Symbol : Footprint Assignments

1	D1 -	LED : LED_THT:LED_D5.0mm
2	D2 -	LED :

Continue selecting footprints for all components. The footprint names for each component in this tutorial are given in the table below. You are welcome to choose any equivalent footprint as long as it maintains the same functionality. Click “**Okay**” to apply your footprints to the components.

Component	Filtered Footprint Name
LED (D)	LED_THT:LED_D5.0mm
Conn_01x07_Pin (J)	Connector_PinHeader_2.54mm:PinHeader_01x07_P2.54mm_Vertical
Conn_01x02_Pin (J)	Connector_PinHeader_2.54mm:PinHeader_01x02_P2.54mm_Vertical

Resistor (R)	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16_Horizontal
Photoresistor, R_Photo (R)	OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical

The final footprint assignments should resemble this:

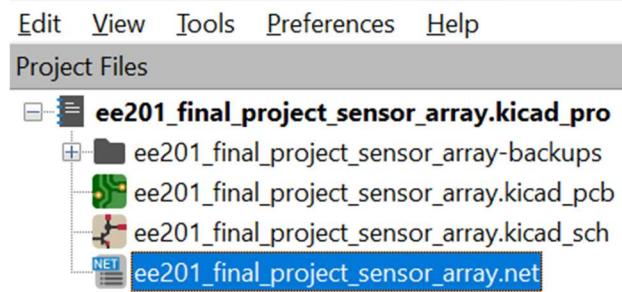
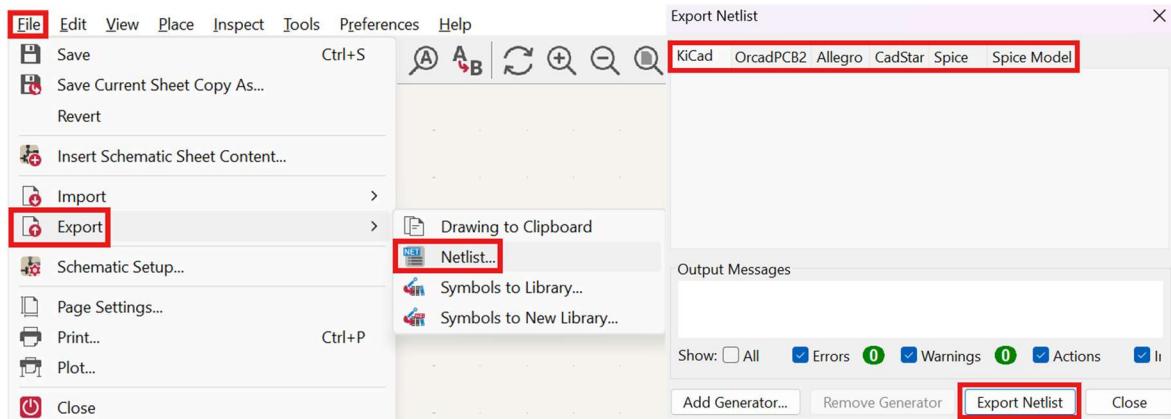
Symbol : Footprint Assignments

```

1      D1 -          LED : LED_THT:LED_D5.0mm
2      D2 -          LED : LED_THT:LED_D5.0mm
3      D3 -          LED : LED_THT:LED_D5.0mm
4      D4 -          LED : LED_THT:LED_D5.0mm
5      D5 -          LED : LED_THT:LED_D5.0mm
6      D6 -          LED : LED_THT:LED_D5.0mm
7      J1 - Conn_01x07_Pin : Connector_PinHeader_2.54mm:PinHeader_1x07_P2.54mm_Vertical
8      J2 - Conn_01x02_Pin : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
9      R1 -          10k : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
10     R2 -          10k : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
11     R3 -          10k : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
12     R4 -          10k : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
13     R5 -          10k : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
14     R6 -          10k : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
15     R7 -          10k : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
16     R8 -          R_Photo : OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical
17     R9 -          R_Photo : OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical
18     R10 -         R_Photo : OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical
19     R11 -         R_Photo : OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical
20     R12 -         R_Photo : OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical
21     R13 -         R_Photo : OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical
22     R14 -         R_Photo : OptoDevice:R_LDR_5.0x4.1mm_P3mm_Vertical
23     R15 -         330 : Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal

```

17 (optional). Many engineers will export their KiCad schematic into something called a “Netlist.” The netlist is a text-file representation of the circuit. Exporting the schematic to a netlist will ensure compatibility with another program if you do not use KiCad’s built in PCB editor for your PCB layout. The tutorial will continue using KiCad’s built in software, but this step is useful in sharing files if you are working with a team or if you are switching between softwares. To export your schematic to a netlist, on the upper ribbon, go to “File” → “Export” → “Netlist.” In the “Export Netlist” pop-up, select the software you want to export to. For this tutorial, we will select KiCad, but many industry engineers use Spice or Allegro. Once your software is selected, click “Export Netlist,” and save the file in your KiCad project folder.



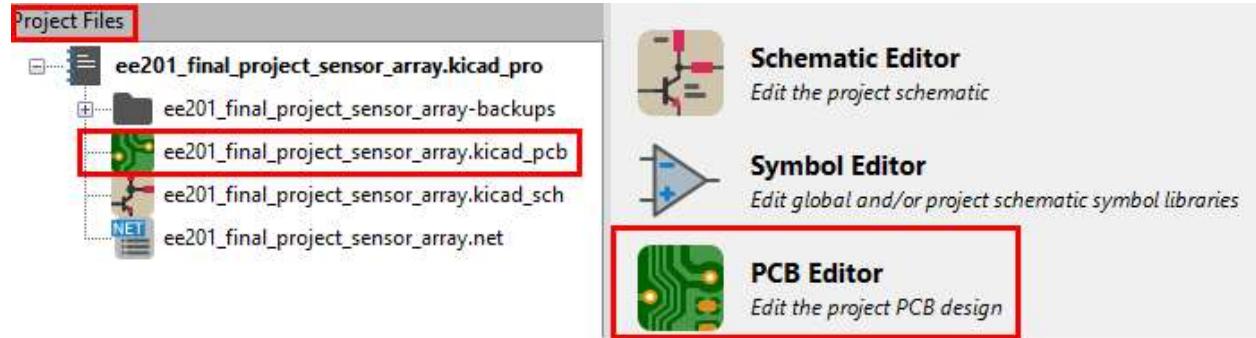
Note: The exported netlist should show up in your project as a ".net file."

You have now exported your schematic to a Netlist, and you are ready to layout your PCB!

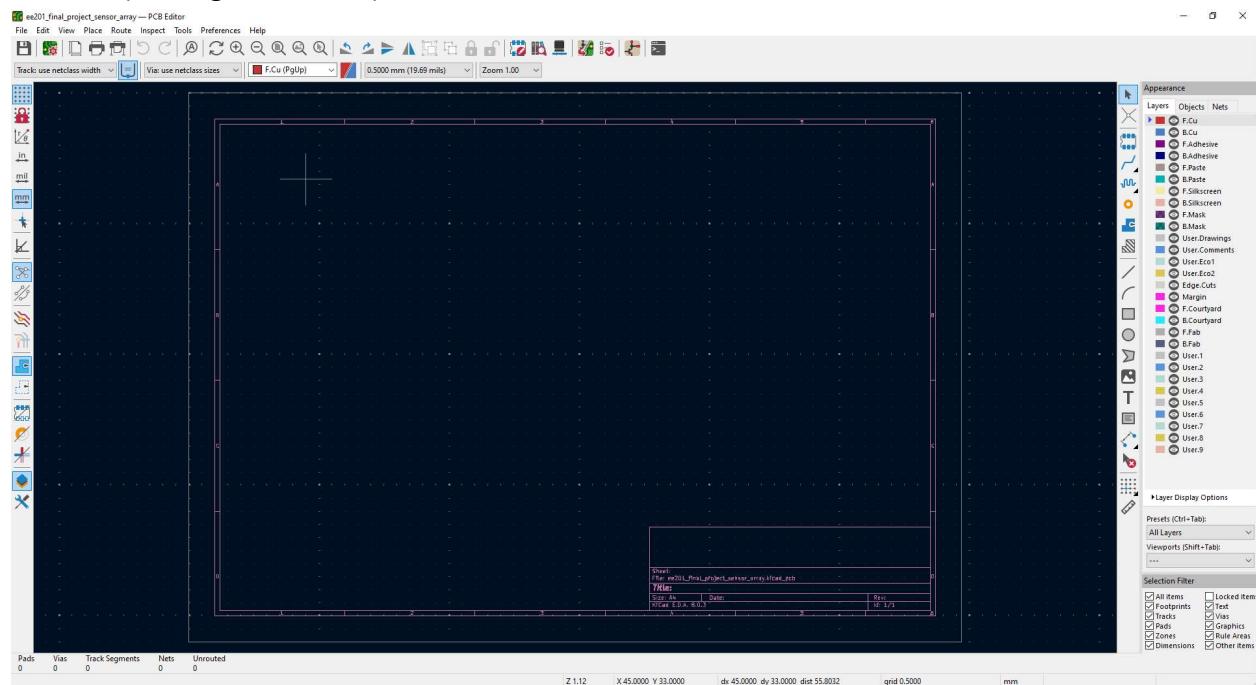
Step 2 Layout in Printed Circuit Board (PCB) editor:

The goal of this section is to familiarize yourself with KiCad's built in PCB editor. Laying out a PCB is a difficult and time-consuming task. Ask for help often!

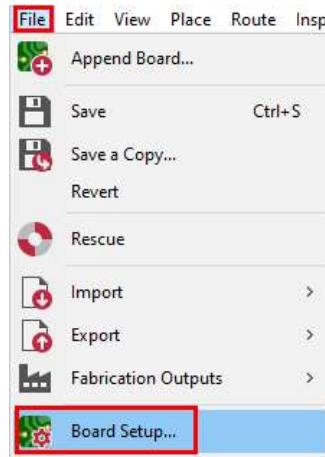
1. With the KiCad project open, “**Double Click**” on the “**PCB Editor**” or the “**.kicad_pcb**” file to edit.



2. A blank layout editor should open in a new window. You should see a similar-looking layout window (if using dark mode).



3. Before we begin laying out our PCB, we want to make sure our board is set up correctly for manufacturing. On the upper ribbon, go to “**File**” → “**Board Setup...**” to open the “**Board Setup**” menu. This part is tedious but important to ensure our board can be manufactured properly.



Ensure the “Text & Graphics” → “Defaults” and “Design Rules” → “Constraints” match the images below.

Board Setup

- Board Stackup
- Physical Stackup
- Board Finish
- Solder Mask/Paste
- Text & Graphics**
- Defaults
- Formatting
- Text Variables
- Design Rules
- Constraints
- Pre-defined Sizes
- Teardrops
- Length-tuning Patterns
- Net Classes
- Custom Rules
- Violation Severity

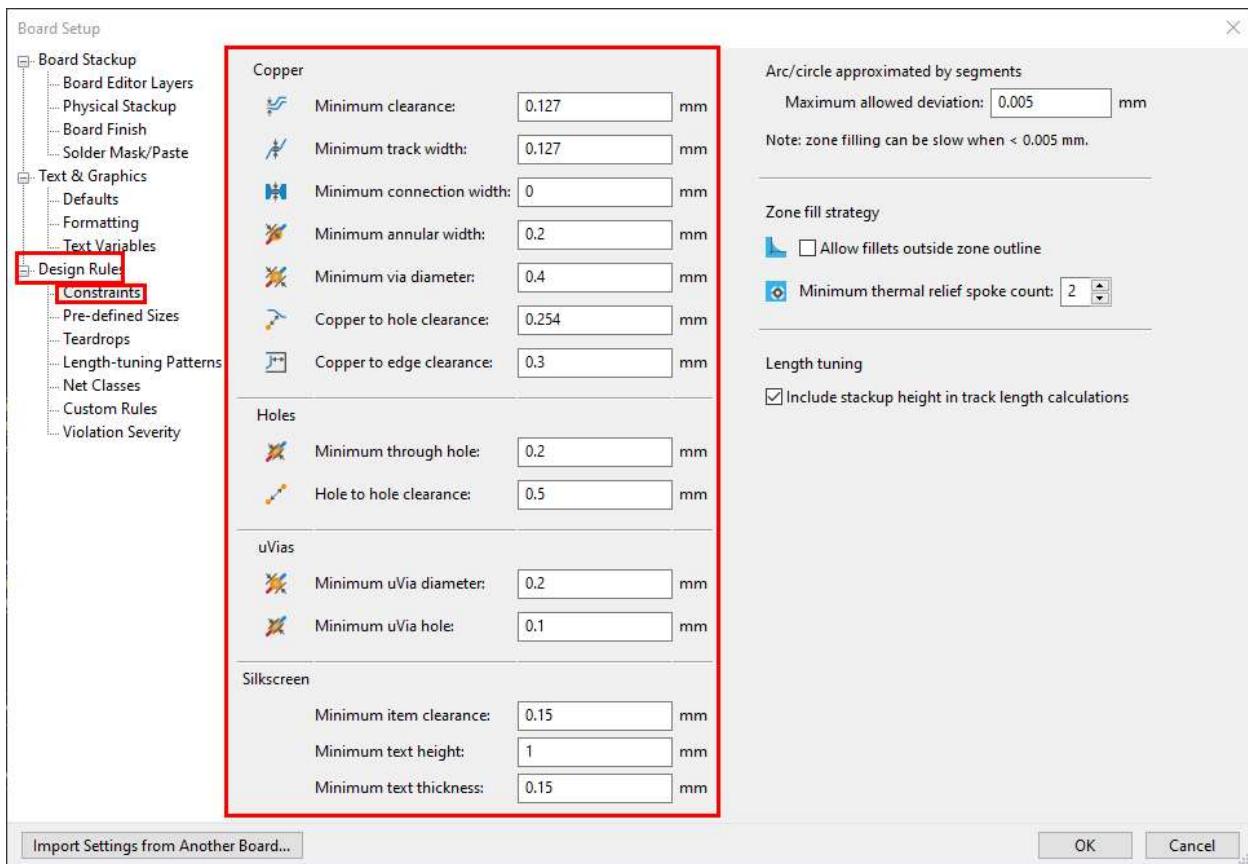
Default properties for new graphics and text:

	Line Thickness	Text Width	Text Height	Text Thickness	Italic	Keep Upright
Silk Layers	0.153 mm	1 mm	1 mm	0.153 mm	<input type="checkbox"/>	<input type="checkbox"/>
Copper Layers	0.2 mm	1.5 mm	1.5 mm	0.3 mm	<input type="checkbox"/>	<input type="checkbox"/>
Edge Cuts	0.1 mm					
Courtyards	0.05 mm					
Fab Layers	0.1 mm	1 mm	1 mm	0.15 mm	<input type="checkbox"/>	<input type="checkbox"/>
Other Layers	0.15 mm	1 mm	1 mm	0.15 mm	<input type="checkbox"/>	<input type="checkbox"/>

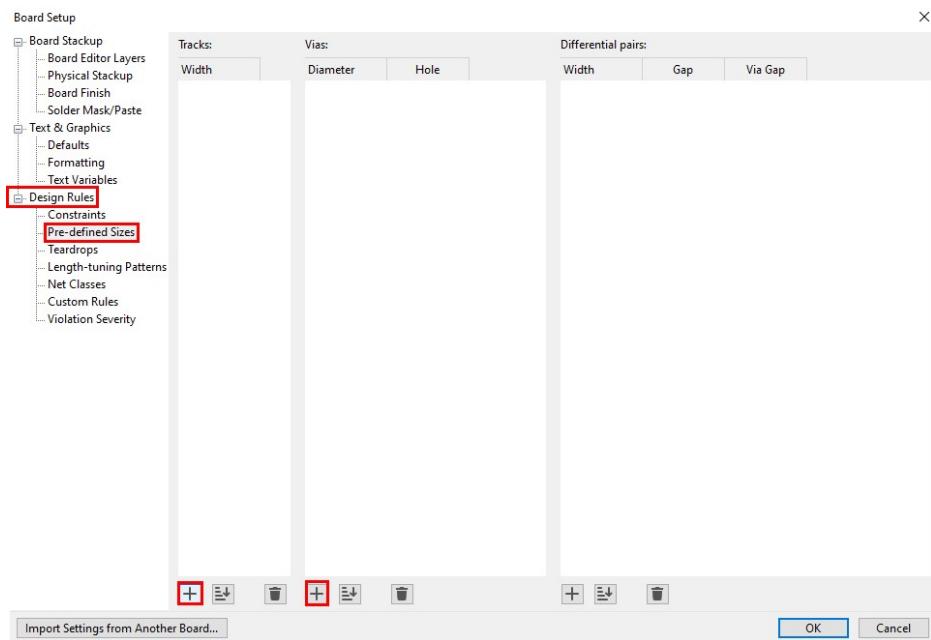
Default properties for new dimension objects:

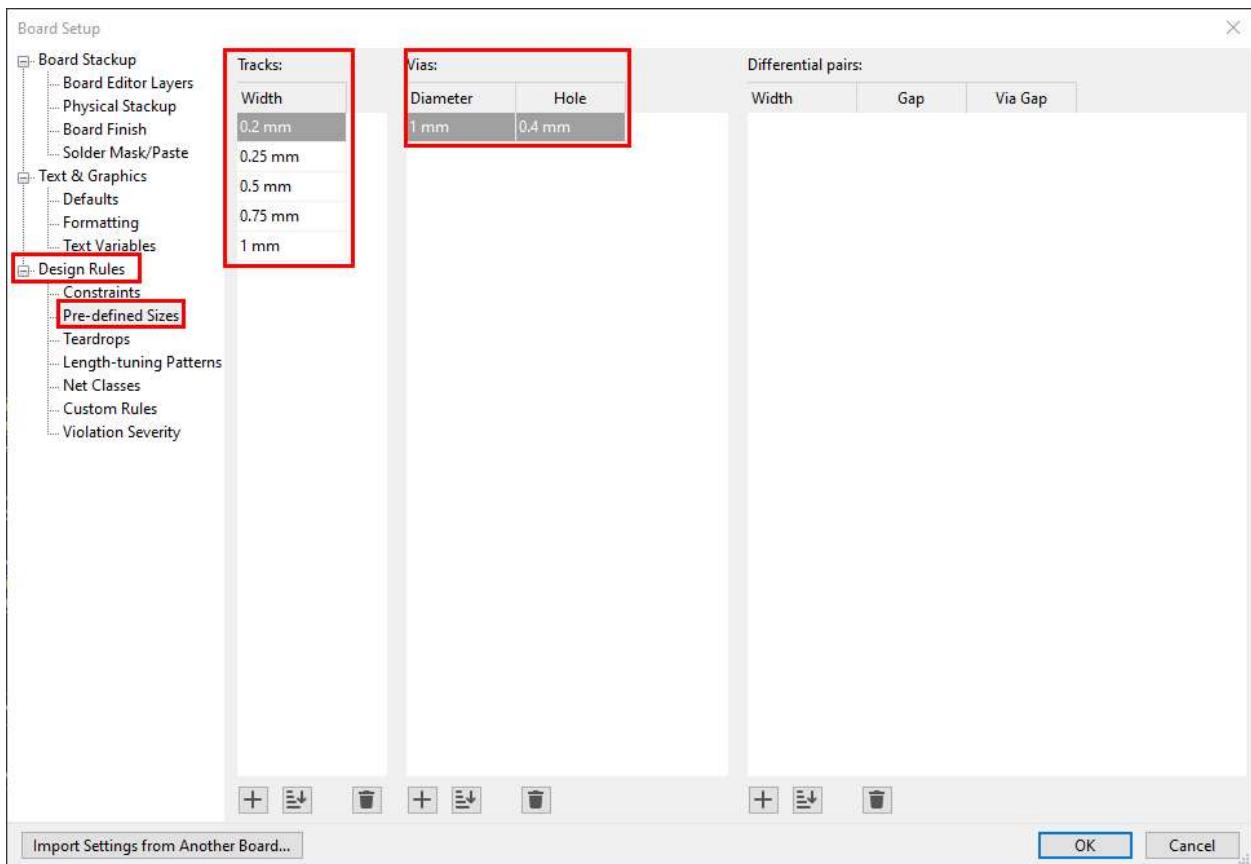
Units:	Automatic	Text position:	Outside
Units format:	1234 mm	<input checked="" type="checkbox"/> Keep text aligned	
Precision:	0.0000	Arrow length:	1.27 mm
<input type="checkbox"/> Suppress trailing zeroes		Extension line offset:	0.5 mm

Import Settings from Another Board... OK Cancel

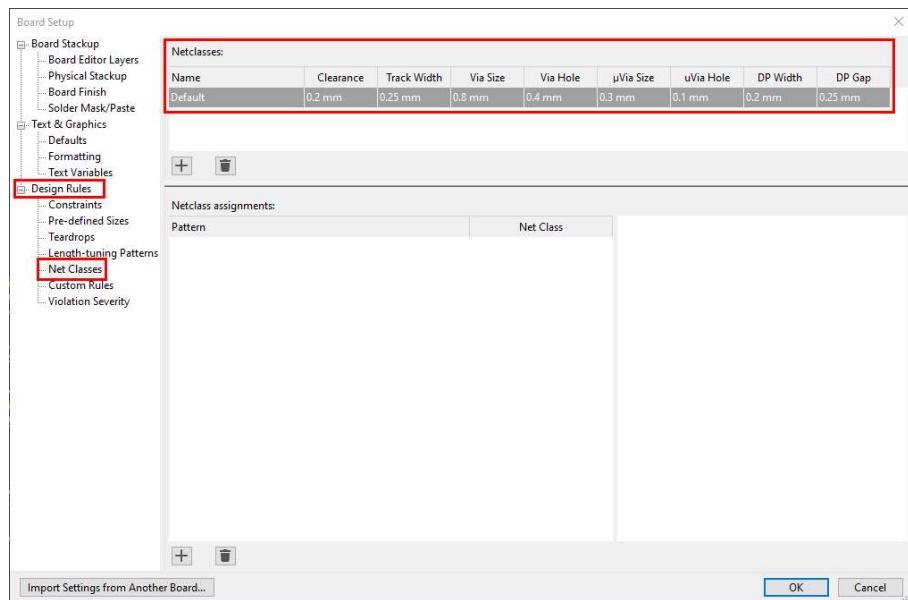


The next part is to edit the “**Design Rule**” → “**Pre-defined Sizes**” section. We need to create five default “**Tracks:**” and one default “**Vias:**.” Click the “+” icon to add a default track or via. Emulate the “**Tracks:**” and “**Vias:**” sections in the images below.





Next is the “**Design Rules**” → “**Net Classes**” section. Change the defaults to emulate the image below.

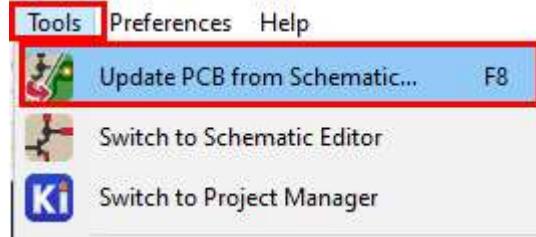


Finally, navigate to “Design Rules” → “Violation Severity.” Emulate the images below. Click “OK” to apply the new board setup settings.

The screenshot shows the "Board Setup" dialog with the "Design Rules" section highlighted by a red box. Within this section, the "Violation Severity" subsection is also highlighted by a red box. The "Electrical" and "Design for Manufacturing" sections are visible, containing various rules and their severity levels (Error, Warning, Ignore). The "Schematic Parity" section is partially visible at the bottom. At the bottom right of the dialog are "OK" and "Cancel" buttons.

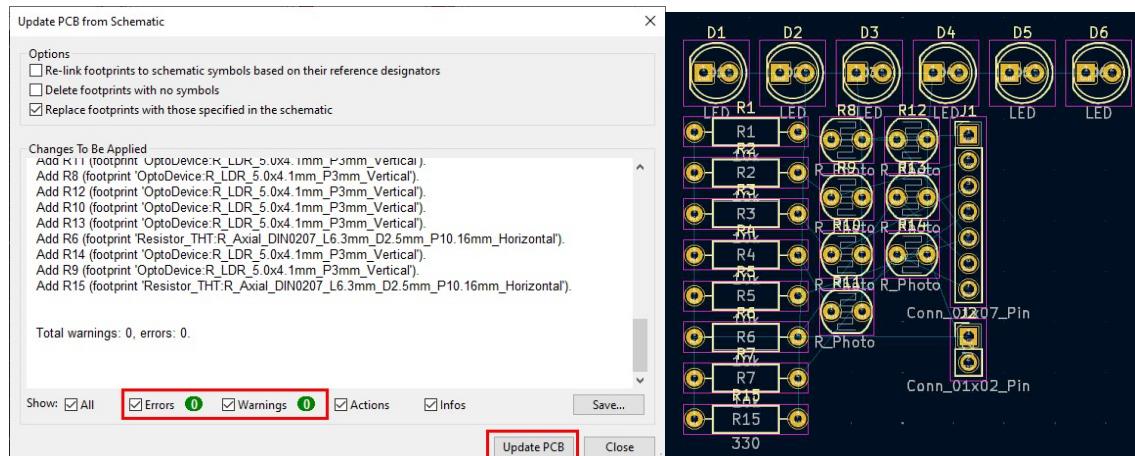
Category	Rule	Error	Warning	Ignore
Electrical	Items shorting two nets:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Tracks crossing:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Clearance violation:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Via is not connected or connected on only one layer:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Track has unconnected end:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Design for Manufacturing	Thermal relief connection to zone incomplete:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Board edge clearance violation:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Hole clearance violation:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Drilled holes too close together:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Drilled holes co-located:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Schematic Parity	Track width:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Annular width:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Hole size out of range:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Micro via hole size out of range:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Courtyards overlap:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Miscellaneous	Footprint has no courtyard defined:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	Footprint has malformed courtyard:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Board has malformed outline:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Copper sliver:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Solder mask aperture bridges items with different nets:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Readability	Copper connection too narrow:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Duplicate footprints:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Missing footprint:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Silkscreen overlap:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Silkscreen clipped by solder mask:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Signal Integrity	Silkscreen clipped by board edge:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Text height out of range:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Text thickness out of range:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Trace length out of range:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Skew between traces out of range:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Miscellaneous	To many or too few vias on a connection:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Differential pair gap out of range:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Differential uncoupled length too long:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Items not allowed:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Copper zones intersect:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Readability	Isolated copper fill:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Footprint is not valid:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Padstack is not valid:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	PTH inside courtyard:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	NPTH inside courtyard:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Signal Integrity	Item on a disabled copper layer:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Unresolved text variable:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Footprint component type doesn't match footprint pads:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	Footprint not found in libraries:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	Footprint doesn't match copy in library:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Miscellaneous	Through hole pad has no hole:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. We set up our board, but our current layout is empty. We need to import our circuit! This can be done using the optional netlist we exported earlier, but we will be using KiCad's built in functionality that links our schematic and layout. On the upper ribbon, go to “Tools” → “Update PCB from Schematic.” This will import your most recently saved schematic into the layout editor.



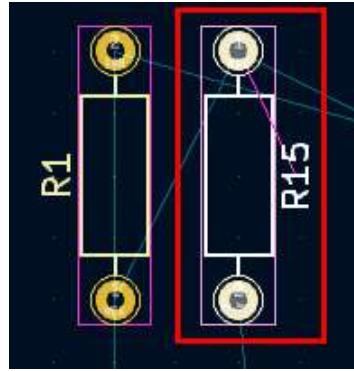
Note: You can also update your PCB using the keyboard shortcut “F8.” Your PCB will not update to match your schematic if you make changes unless you save your schematic and update the PCB in the layout editor!

In the updater, make sure the “Errors” and “Warnings” boxes are checked. Because we have no errors or warnings, we can click “Update PCB” and then “Close” to import our schematic components and connections. The components come grouped together in what is called a “Rat’s Nest.” A rat’s nest is simply jargon that means a grouping of all the circuit components and their connections.



Note: The initial location placement of the components does not matter as they will be moved in future steps.

5. Now that the components are imported, we need to place them in their proper locations. If we left the components in their current locations, our PCB would not be very effective! For example, our photoresistors are not in order and are not in a single line (meaning they would not perform as expected). To move a component, select it, by clicking on it, and then drag it to a new location. A component will turn white when selected.



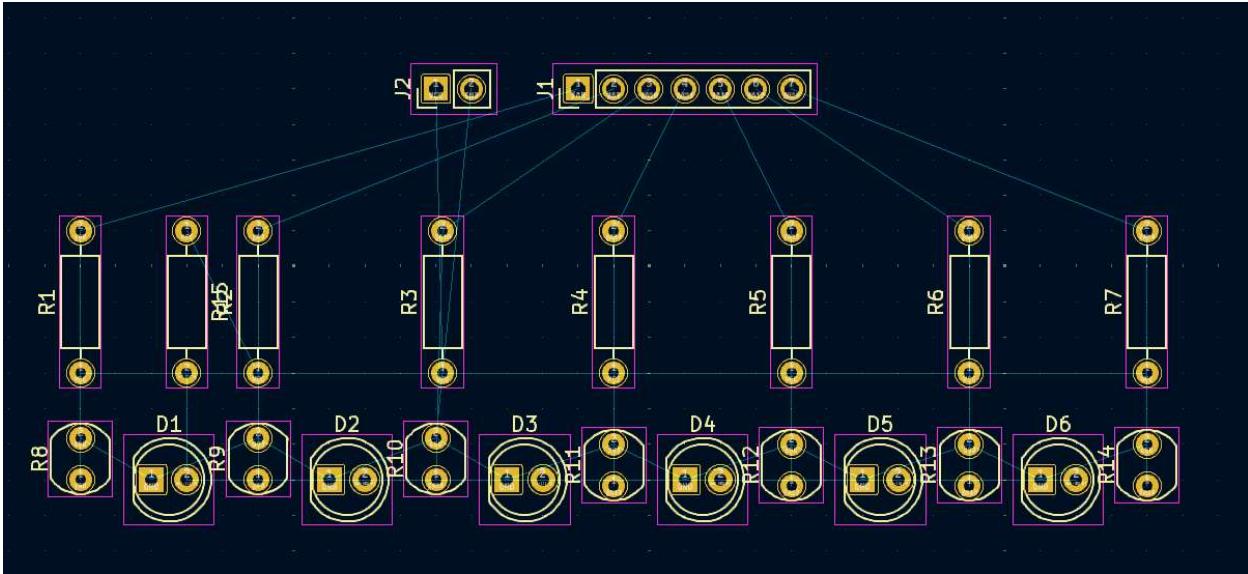
Note: You can also move a selected component by using the keyboard shortcut “M.” You can rotate a selected component by pressing “R.”

You may be wondering why the previous image does not have the gray text. You can change which characteristics are displayed by going to the “**Appearance**” menu. To turn off the gray text, deselect “**F.Fab**” and “**B.Fab**.” The “**F**” stands for front (of the board) and “**B**” stands for back (of the board). The “**Fab**” means fabrication, or info only sent to the manufacturer but not printed onto the board. Turning off fabrication elements will help visually declutter your layout, hopefully making placing your components easier.

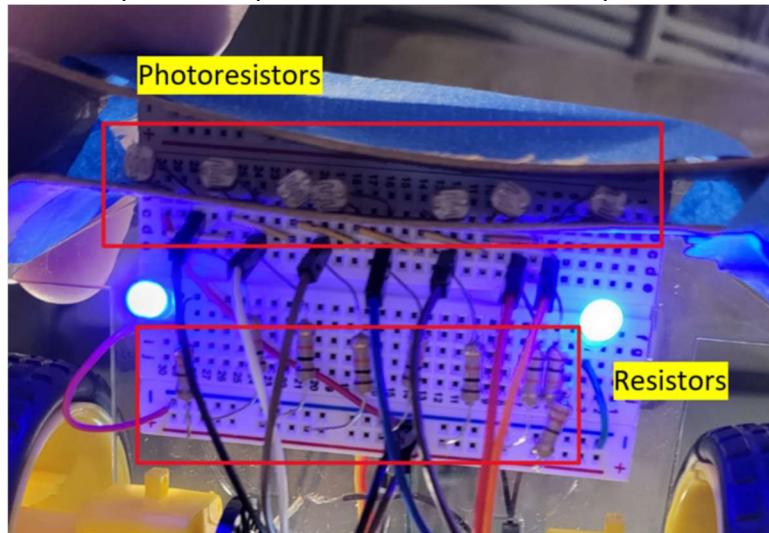


The goal is to place your components roughly where you think they should go. In integrated circuits (ICs), a datasheet will often provide suggestions on where to place related components. For this project, component placement does not matter greatly if you are able to correctly connect every component in the layout editor and you believe you will be able to solder the

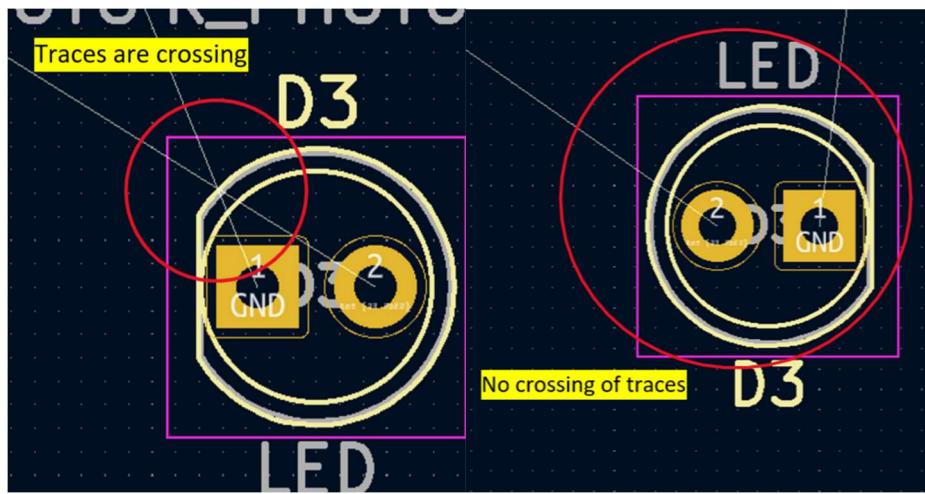
components once the board is ordered and manufactured. A rough suggestion of initial component placement is given in the image below.



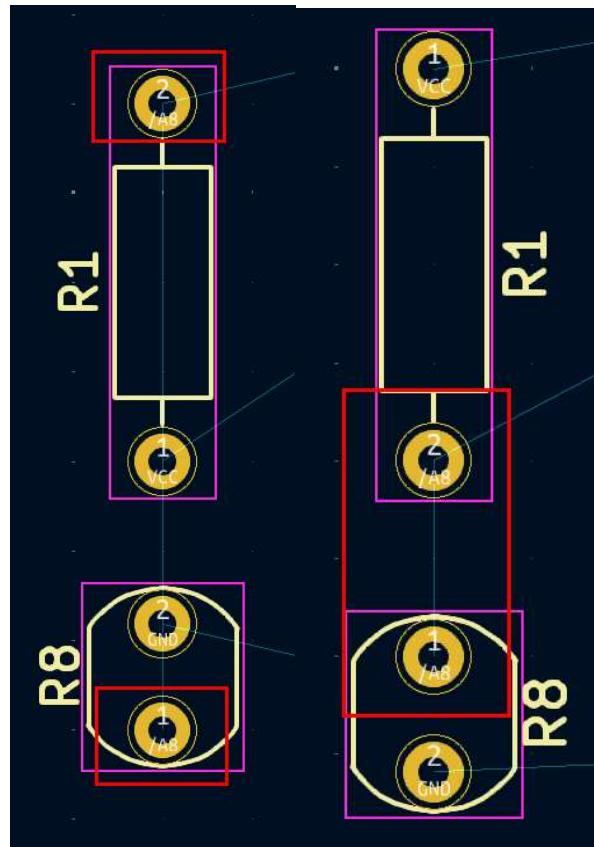
The component placement loosely follows the structure of the initial breadboard prototype. The LEDs and photoresistors are placed together near the edge of the board. The resistors are placed near the photoresistors and the connectors are placed on the opposite side of the board. There are several ways we can improve our placement from our initial placement.



The most obvious, and often easiest to implement, improvement is to reduce the number of traces that cross other traces. Reducing crossing traces will reduce the number of vias on the board and make for a simpler final board.

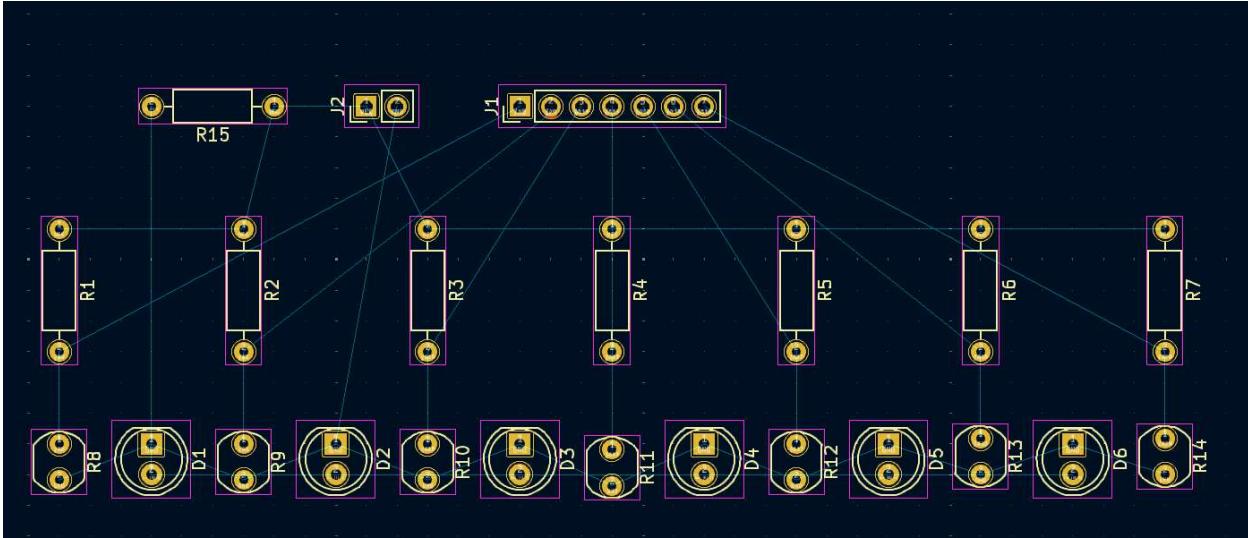


Another easy change is to rotate components so that connections are closer to each other. The goal is to connect every component as it is in the schematic. For example, rotating “R1” and “R8” 180 degrees each puts the “A8” connection closer to each other. Ignore “GND” and “VCC” connections for now as we will handle them all at once with a cool trick!

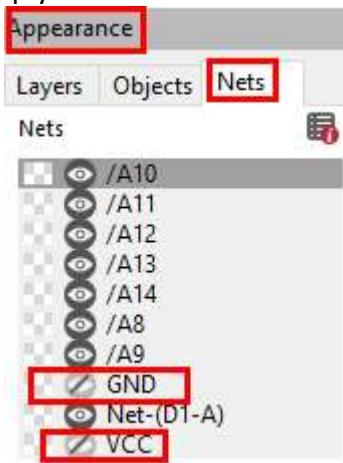


An improved initial component layout is shown below. It is important to remember that your board does not need to look like the example board going forward. It is more important to understand the concepts behind placement than to blindly copy the example board (which

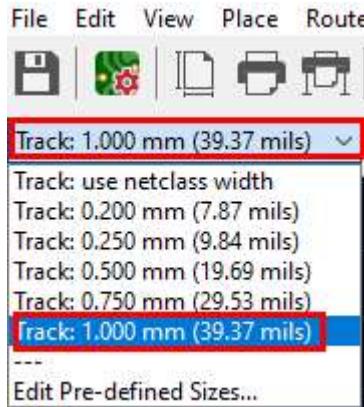
might not make sense since the designer is looking ahead and using some higher-level concepts).



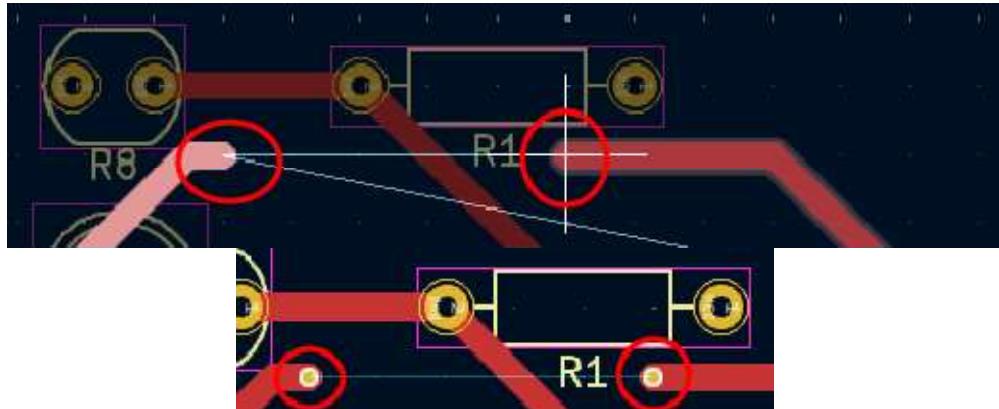
6. Now that we have our initial components placed, we can begin connecting our layout. Connecting components consists of routing Copper traces from the pad of one component to the pad of another component. Right now, there are an overwhelming number of traces, so we will hide “VCC” and “GND” connections and only handle the rest of the connections. We will complete the “VCC” and “GND” traces all at once at the end. To turn off certain connection visibility, go to the “Appearance” menu and navigate to the “Nets” menu. Turn off visibility for “GND” and “VCC” nets. A net is simply the reference name of connected components.



Now we can connect our components with a “Trace.” Select the “Route Tracks” () tool on the right taskbar. Change the “Default Width” to use for your trace. This tutorial will use “**Track: 1.000 mm (39.37 mils)**,” but the trace width you choose does not matter a great deal for this project. Larger trace widths are usually used for power connections such as “VCC” and “GND” as they can support more power and act as smaller wire resistances, allowing your PCB to function more like an ideal circuit; however, our circuit is not constrained by size, so the tutorial will use the largest trace width for all traces.

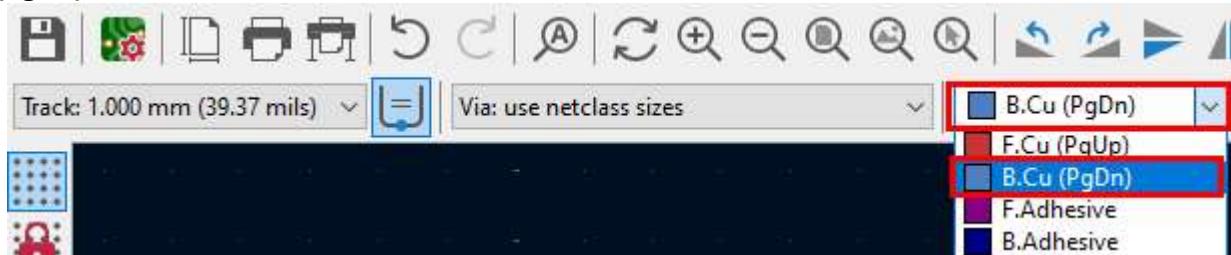


Connect all visible suggested traces. If you run into a situation where you want to route a trace but it crosses another trace, you will have to create a “**Via**.” A via is a hole that connects the front of the board to the bottom of the board. Use the keyboard shortcut “V” (while in the “**Route Tracks**” interface) to place a via at the two locations you want to connect. The vias will allow you to run a trace on the back side of the board that doesn’t intersect the existing trace!

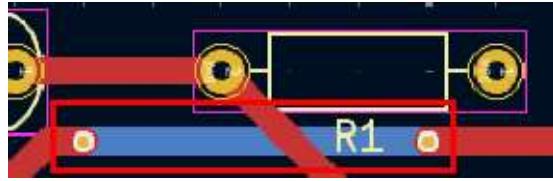


*Note: You can use the keyboard shortcut “X” to open the “**Route Tracks**” interface.*

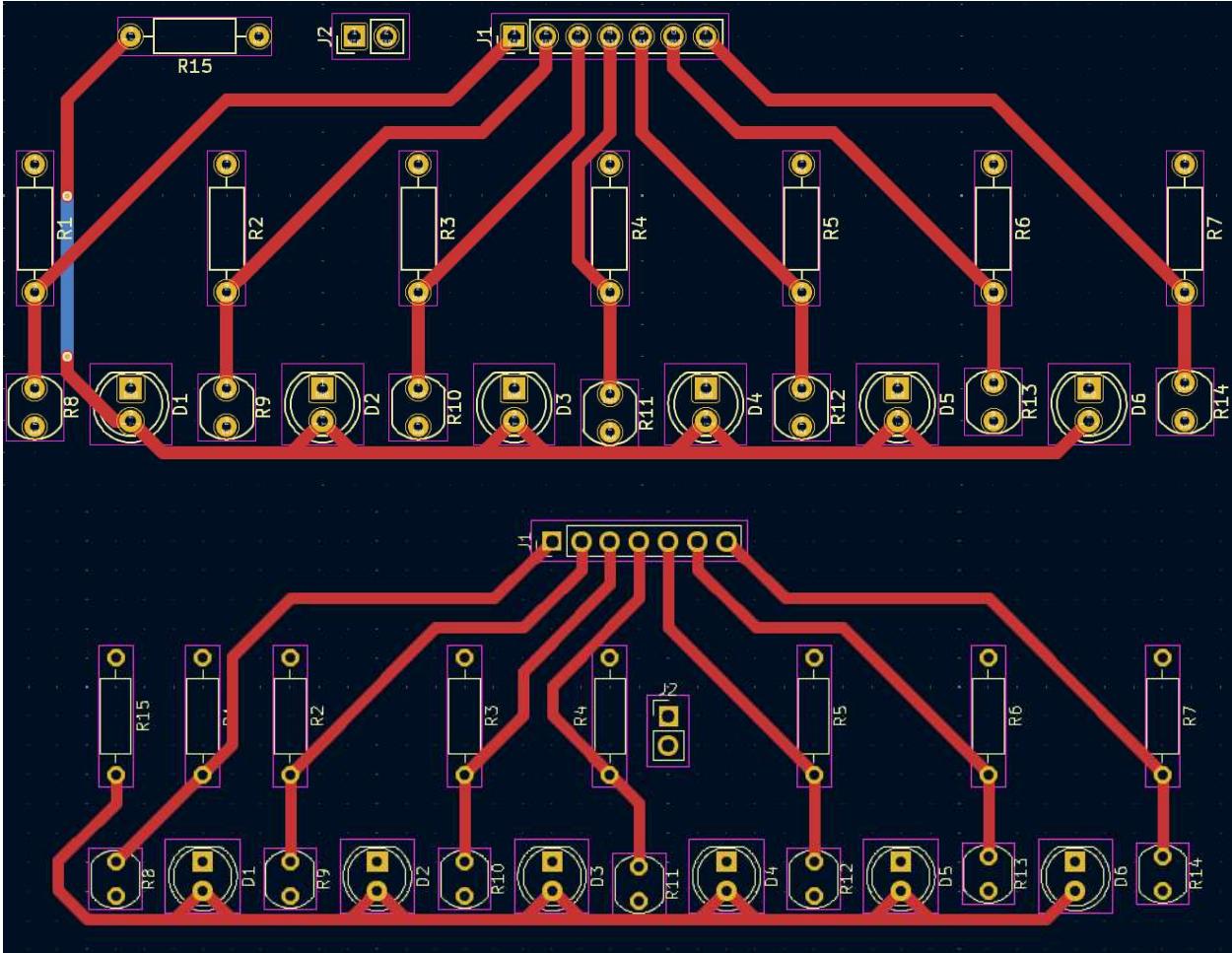
After the vias are placed, you need to switch from routing traces on the front side of the board to the back side. Use the dropdown on the upper toolbar to change your active layer to “B.Cu (PgDn).”



Connect the two vias on the back side using the “**Route Tracks**” interface again. The default color scheme is that traces on the front side of the board will show up red and traces on the back side of the board will show up blue.

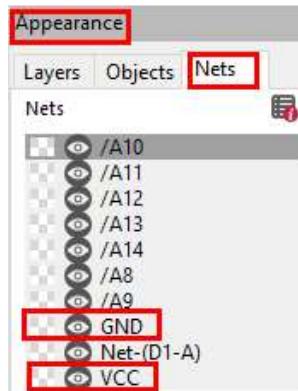


Finish connecting all visible suggested traces using vias as you see fit. It is good practice to reduce the number of vias on a board. Two example connections are shown below (once again, your circuit does not need to look like the examples!).



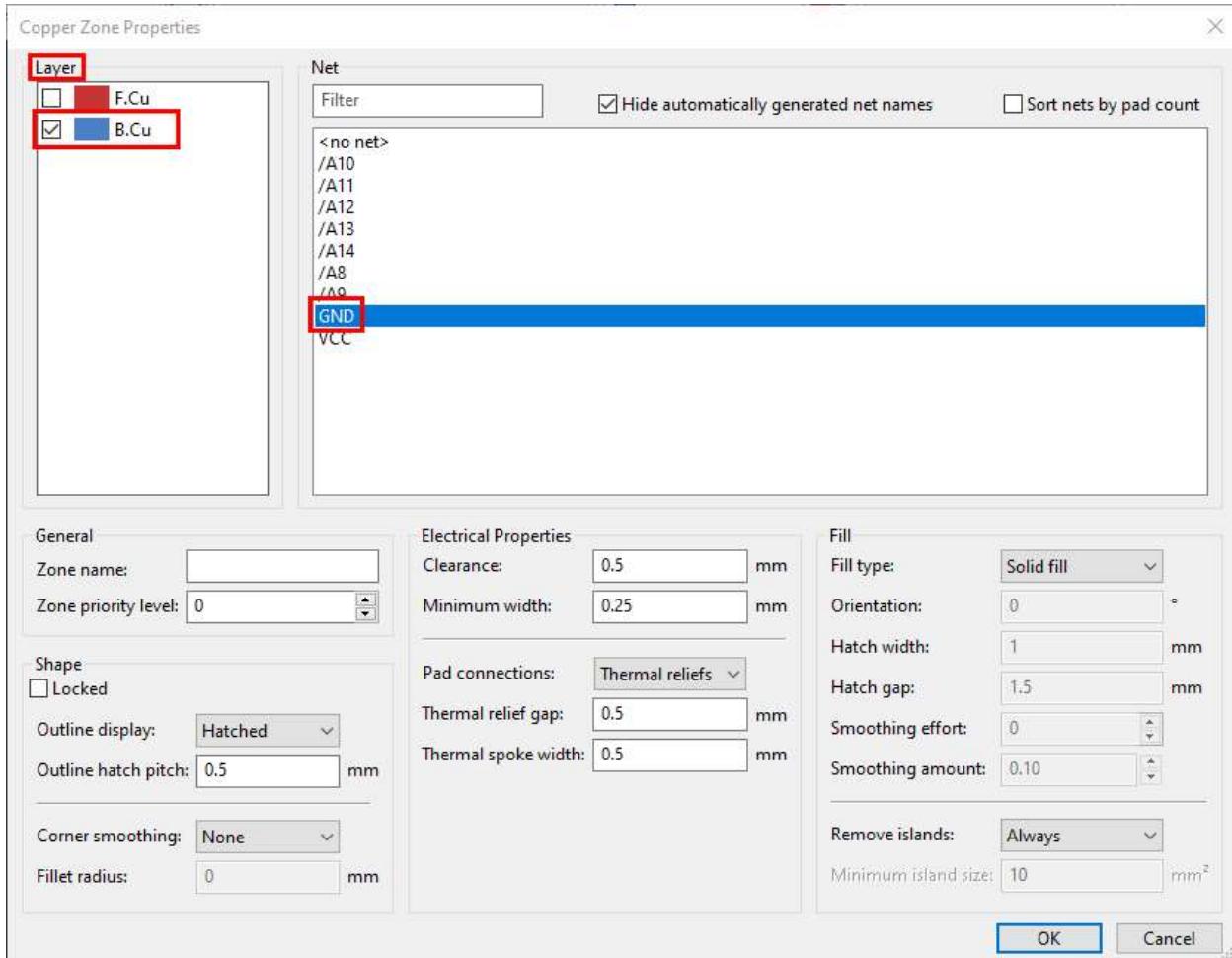
Note: The circuit can be connected with or without vias. It is the designer's decision where to place components and how to connect them. Both examples are acceptable.

7. Now we will connect our “**GND**” and “**VCC**” nets using “**Pours**” (or “**Filled Zones**”) instead of “**Traces**.” Using a pour negates the need to manually connect nets with traces. Pours have many electrical benefits, but for this tutorial, their main benefit is aiding in the simplicity of the board. To begin placing a pour, first turn back on the visibility of the “**GND**” and “**VCC**” nets by using the “**Appearance**” menu and navigating to the “**Nets**” menu. Make sure the “**GND**” and “**VCC**” nets are set to visible.

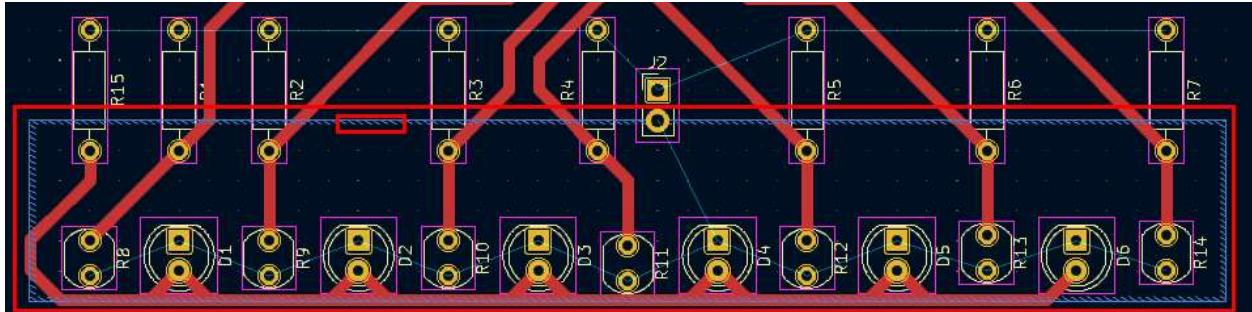


Note: You can do “GND” or “VCC” one at a time instead of doing both at the same time.

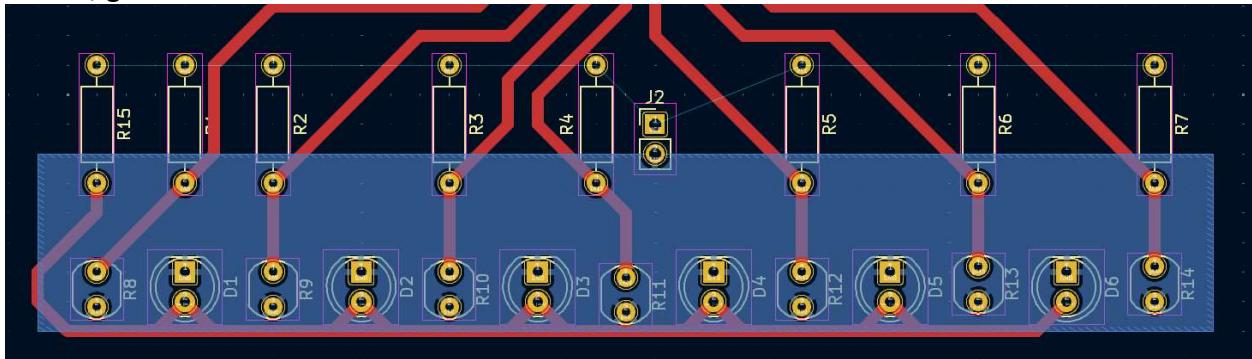
On the right taskbar, click on “Add Filled Zone” (). Click to open the “Copper Zone Properties” menu. Select the “Layer” as “B.Cu” and the “Net” as “GND.” This tells the software we want to make a Copper pour on the back layer of our board that will connect our “GND” nets.



Draw a polygon (of any shape) that encloses all of your “**GND**” connections by using your mouse and clicking where you want the vertices to be. Make sure you close the shape! The polygon will show blue hashing around the lines you just drew when it becomes a closed shape.



We have created a zone, but now we need to fill it with Copper. To fill the zone, on the upper ribbon, go to “Edit” → “Fill All Zones.” The zone should now fill and be shaded blue.



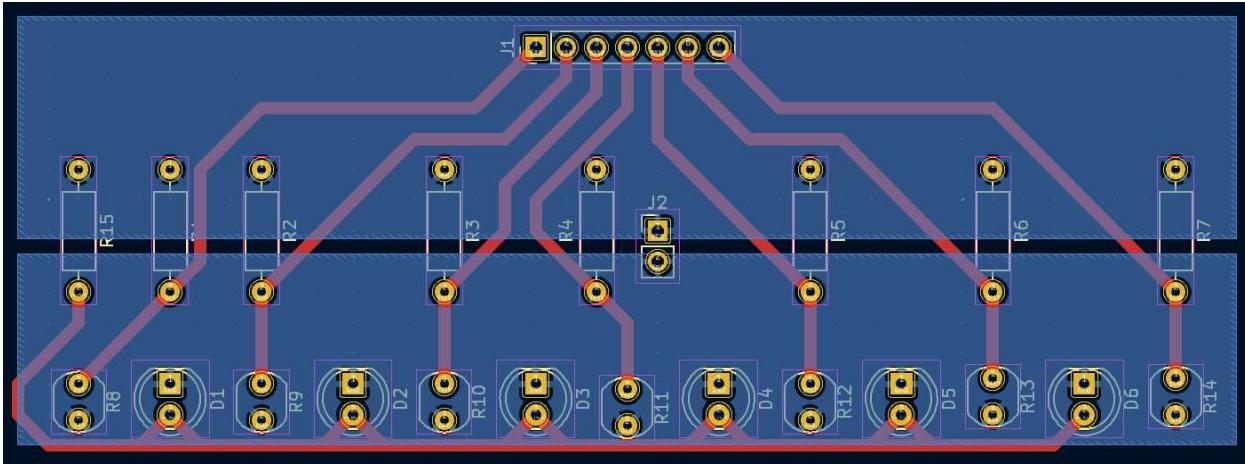
Note: You can also use the keyboard shortcut “B” to fill zones.

The software will automatically connect all pads labeled with the net “**GND**” that are within the filled zone. You can check the connection by zooming in on the individual pads to see the “**Spokes**” connecting the pads to the pour.

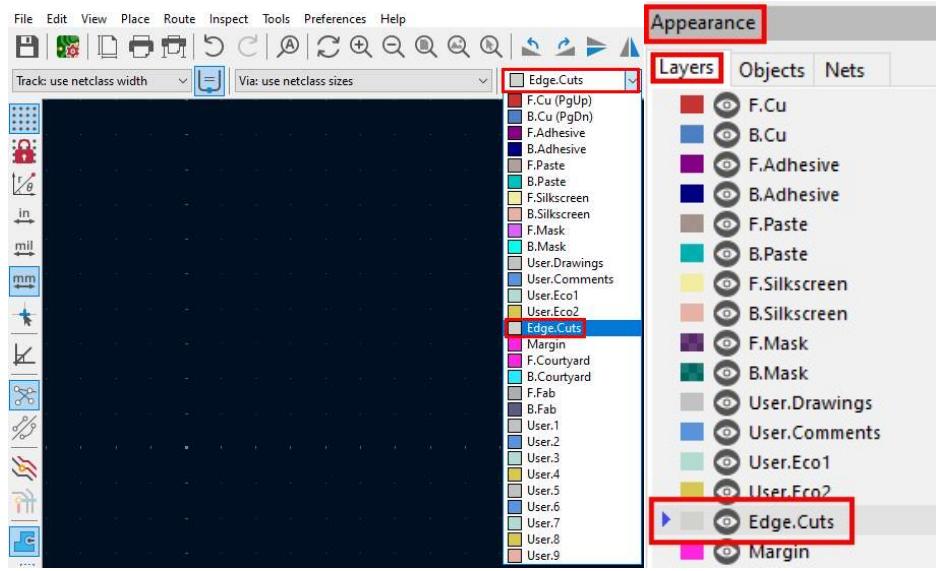


Note: Connections may have different numbers of spokes. This does not impact performance for our specific board.

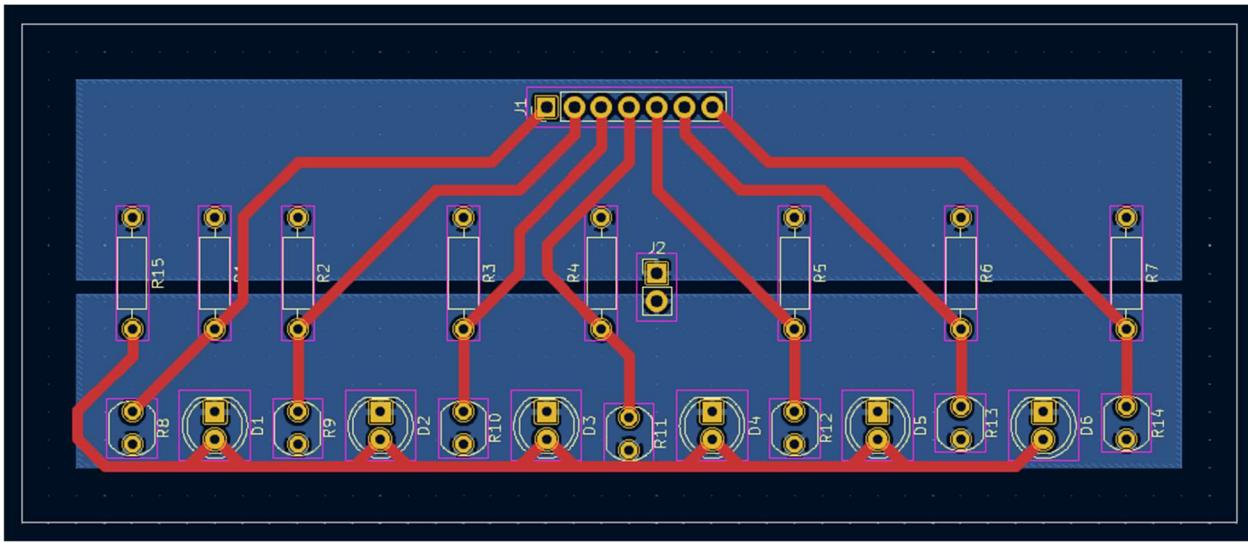
Repeat step 7 (this step) again for the “**VCC**” net.



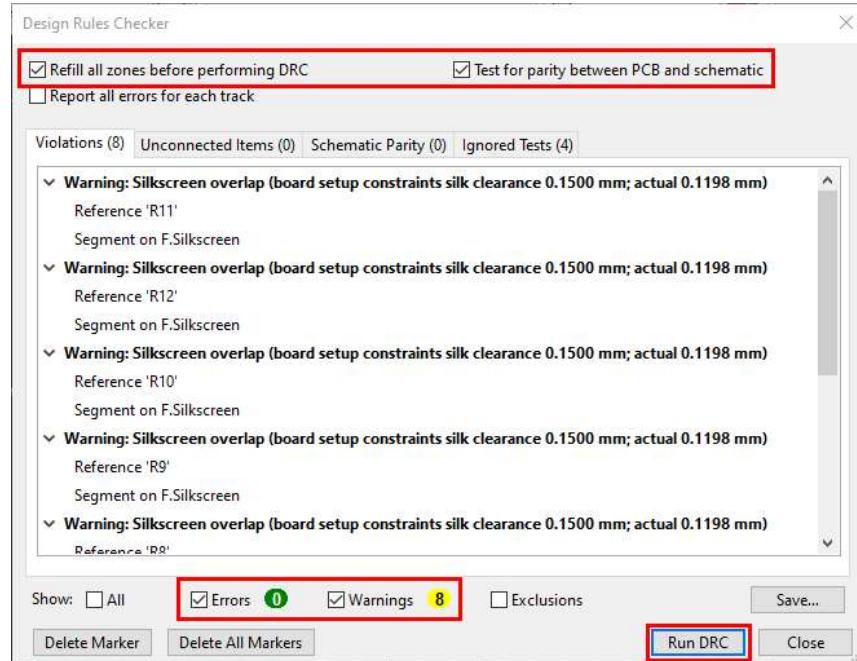
8. We now need to create the physical outline for the board to be manufactured. To do this, we need to tell the software we want to create a physical board shape. On the “**Appearance**” menu, navigate to “**Layers**,” and select “**Edge.Cuts**.” The “**Edge.Cuts**” layer tells the software where the edge of the board will be. You can also select the “**Edge.Cuts**” layer by using the dropdown on the upper taskbar.



With the layer selected as “**Edge.Cuts**,” draw the outline of your board. You can use any of the drawing tools on the right taskbar: “**Draw Line**” (/), “**Draw Arc**” (⌂), “**Draw Rectangle**” (□), “**Draw Circle**” (○), or “**Draw Graphic Polygon**” (▶). Make sure there aren’t any traces or circuit components outside of your board outline! An example board outline is given in the image below.



9. Now that we have our board completed, we need to check the validity of our design. Similar to the “ERC” we used for our schematic, we will use the “Design Rules Checker” (or “DRC”) to check our board design. It is crucial that our board setup is done as outlined in step 3 to ensure our “DRC” checks correctly for our manufacturing settings. On the upper ribbon, go to “Inspect” → “Design Rules Checker” to open the “Design Rules Checker.” Ensure the “Refill all zones before performing DRC,” “Test for parity between PCB and schematic,” “Errors,” and “Warnings” boxes are checked before clicking “Run DRC.”



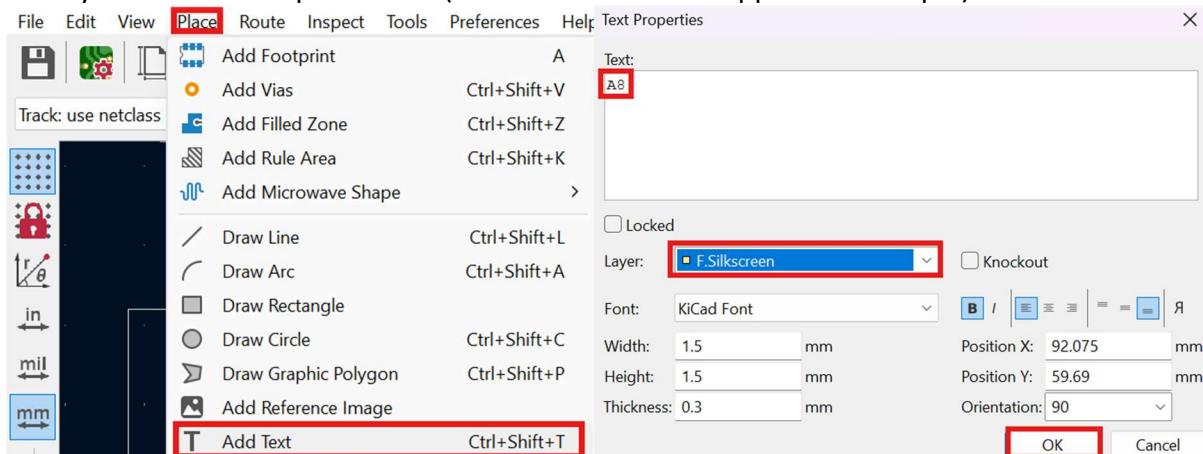
Because our DRC has no errors (even though there are some warnings), we could order this board, and it would function properly electrically!

Congratulations on completing the most basic, functioning PCB for your project :)

There are several things we recommend to improve our board, but they are not explicitly required. The rest of this tutorial will focus on optional, but recommended, improvements to the PCB.

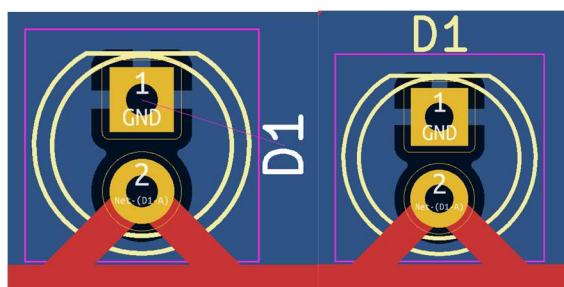
10a (optional). An incredibly helpful addition to any PCB is proper labeling and images. Labeling can help with placing components correctly on prototypes and helping the users understand a board's functionality. Images add creativity and can impart a unique ownership of the board.

The most basic form of labeling comes from using text to create a “**Silkscreen**.” The silkscreen is paint applied to the board. A good use of text is to label our header pins with their corresponding nets. To create a text, on the upper ribbon, go to “Place” → “Add Text.” Type whatever text you want into the “Text” field. Make sure to select the layer as “F.Silkscreen” to ensure your text will be painted on (instead of made of Copper for example).



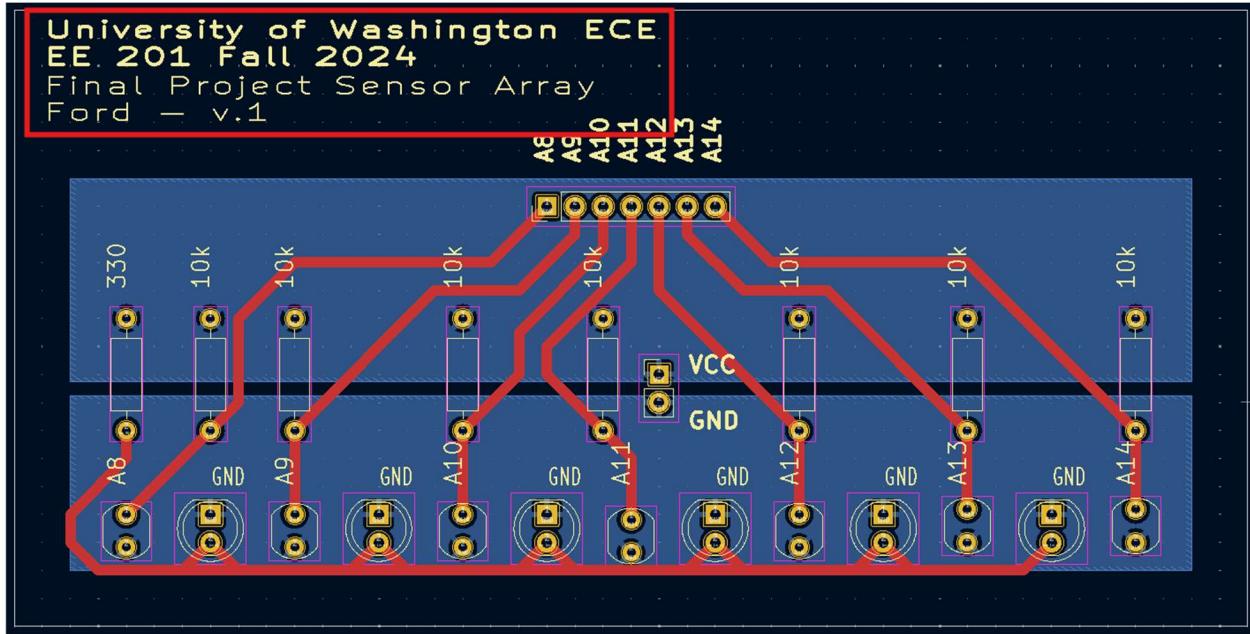
Note: You can also use the keyboard shortcut “Ctrl” + “Shift” + “T” to add text. Silkscreens are viewable even when placed over traces or pours.

It is also helpful to move the component labels to a more visible location rather than their default component location. To move a component's silkscreen, click on the text only, and then move the silkscreen to a desired location. The silkscreen will change to white when it is selected.

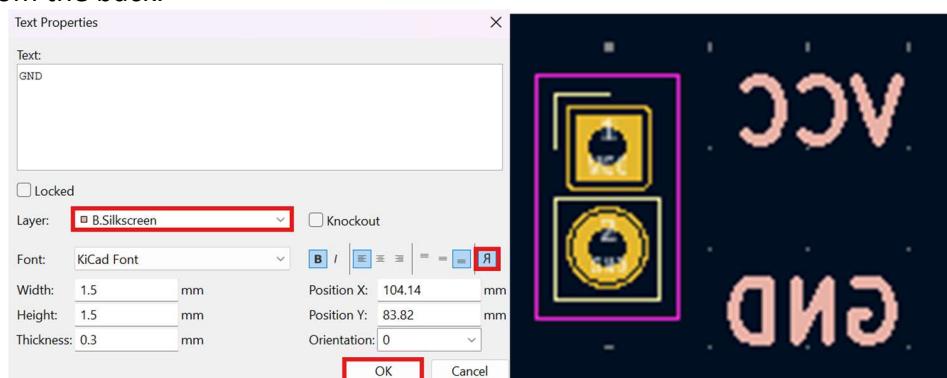


Note: You can also delete silkscreens you don't want by selecting it and using the keyboard shortcut “Delete.”

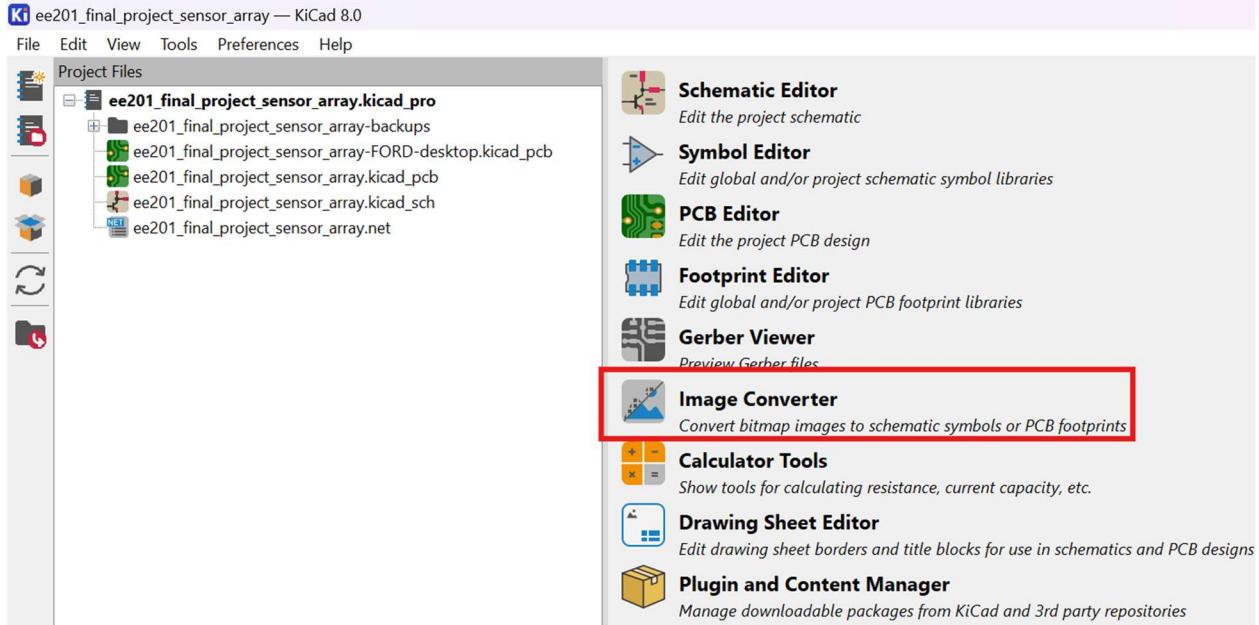
It is also useful to include information about the board itself such as company, functionality, version, date, creator, or anything else that may help a user understand the board. An example is shown in the image below.



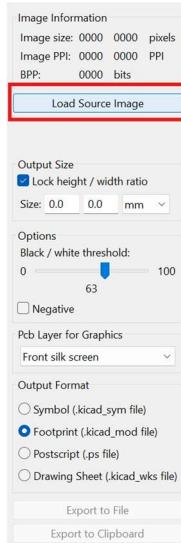
Our information is currently only displayed on the front of the board, but what if we want to display the information on the back of the board? To put text on the back of the board, we need to change our text layer to “**B.Silkscreen**.” We also need to reverse the text so it is legible when we turn the board around. Make sure to select the “**я**” option to ensure the text printed is legible from the back.



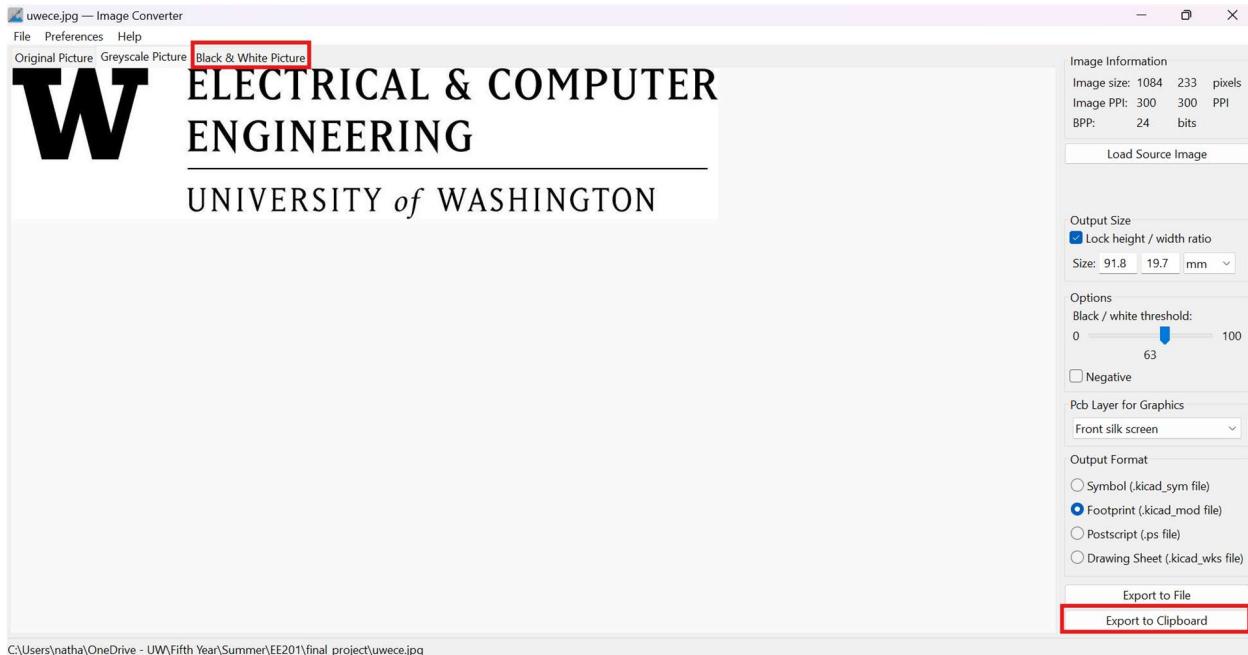
You can also add images to your design. Images can only be silkscreened in one color - either ink or no ink. This can be represented in a file type called a “**Bitmap**.” The easiest way to make a bitmap is to use KiCad’s Built in “**Image Converter**.” On the KiCad project screen, select the “**Image Converter**” tool to open the “**Image Converter**” interface.



On the “**Image Information**” sidebar, select the “**Load Source Image**” option and select the image you want to upload. The file must be an image file, so a “**.jpg**” or “**.jpeg**” file is best. There are other acceptable file formats, but the standard photo format is usually fine.



With your file uploaded, you can now modify your image if you want. Make sure the “**Black & White Picture**” tab is selected when creating your bitmap. It’s recommended to leave the default settings (shown in the image below) before using your bitmap. You can save the bitmap, but for this example, the bitmap will not be used again, so copying it by clicking “**Export to Clipboard**” is sufficient.

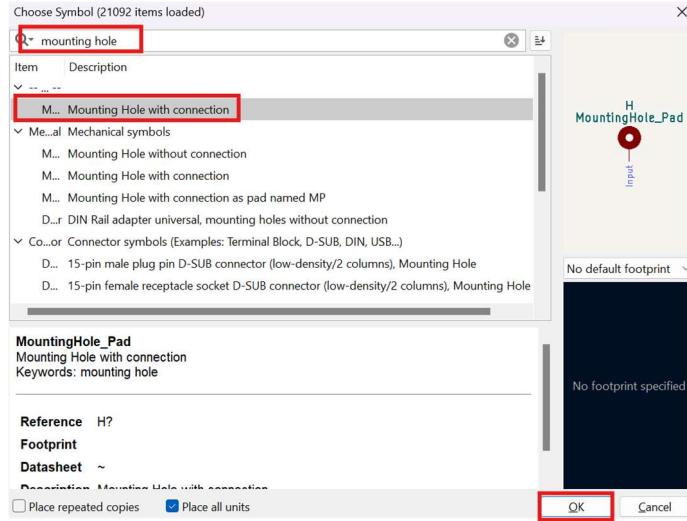


In the PCB editor, **“Right Click”** and **“Paste”** the copied bitmap. The bitmap can now be manipulated like any other footprint.

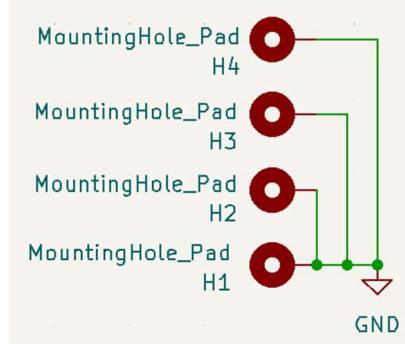


10b (optional). A helpful addition to the PCB for use in your final project is to add mounting holes to attach the PCB to your cart. There are several ways to add mounting holes, but the most straightforward is to go to the schematic editor and add mounting holes then manipulate them in the PCB editor.

In the schematic editor, use the **“Add Symbol”** button to open the **“Symbol Library.”** Search for **“Mounting Hole”** and select a mounting hole. It is good practice to ground mounting holes, so the example will use a mounting hole with connection, but it is not necessary for this PCB.



Create four mounting holes (or however many holes you desire) and place them on your schematic. If using a mounting hole with connection, connect your mounting holes to a “**GND**” symbol.

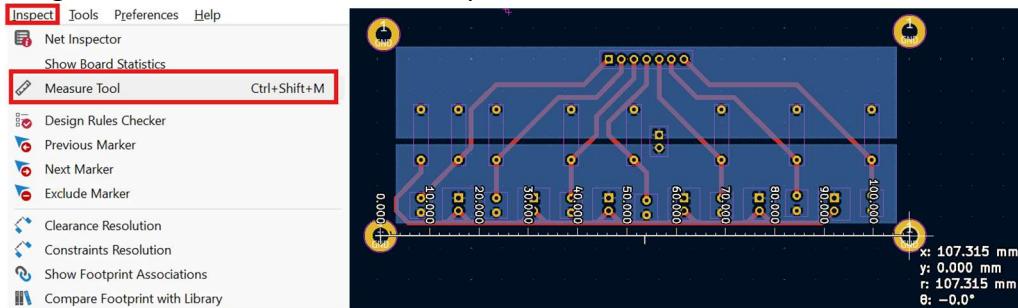


Make sure to assign the mounting holes a footprint! It is recommended to use M3 screw holes. This example uses the footprint: “**MountingHole:MountingHole_3.2mm_M3_Pad**,” but you can choose any equivalent footprint.

Return to the PCB editor (make sure to click “**Update PCB from Schematic**” or press “**F8**” to get the mounting holes to show up). Place the mounting holes on your schematic and connect them to ground if needed.

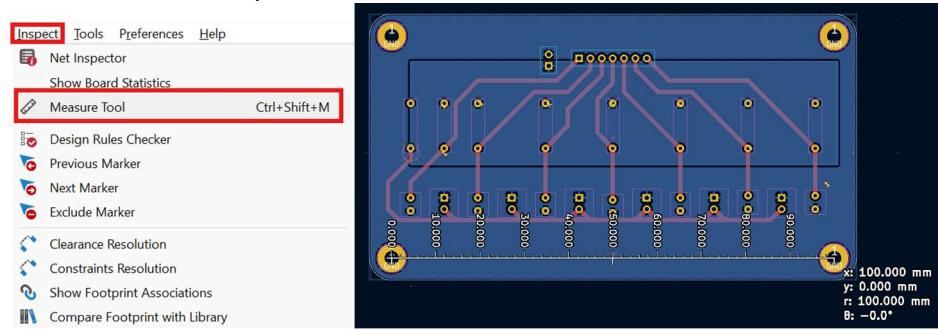
The goal is to mount your PCB to a chassis. The chassis will likely be 3D printed, so you need to know how far apart the mounting holes are in order to create corresponding holes on your chassis! To check how far apart your mounting holes are, on the upper ribbon, go to “**Inspect**” → “**Measure Tool**” to open the ruler. Click on two points to measure the distance between them. Conveniently, the measure tool will snap to the center of the mounting holes when measuring distance between them. There are two different strategies for placing mounting holes.

i) The first is to place the mounting holes wherever seems best on your PCB editor and then design the chassis to match the holes. One downside of this method is that the mounting hole distances might be non-standard distances apart.



Note: You can use the keyboard shortcut “Ctrl” + “Shift” + “M” to use the measure tool.

ii) The second method is to place your mounting holes a standard distance apart first. One downside to this method is that you may need to move board components to get the holes in the correct location. This example has chosen hole distances of 100.000mm and 50.000mm.



10c (optional). The next step in improving your PCB would be adding components to improve functionality! Components to include could be the potentiometer circuit for changing the values of “S,” “P,” “I,” and “D” for your code. Another possible addition is to include a display for showing the robot’s status or variable values. The IR remote and sensor can be used to make wireless changes to your robot’s setup. The possibilities are endless, but building or improving upon a lab from this quarter is an easy way to improve your functionality.