

Redhawk ESP32 EMBEDDED SYSTEMS ROBOT—GET STARTED WITH CIRCUIT BOARD DESIGN

update 1/10/26 - this document is continually updated—email fernandg@seattleu.edu for updates.

Embedded systems design is about building real hardware that works in the physical world—not just drawing circuits or breadboarding. It is the practice of applying theory to products that are manufactured, powered on, and expected to behave reliably.

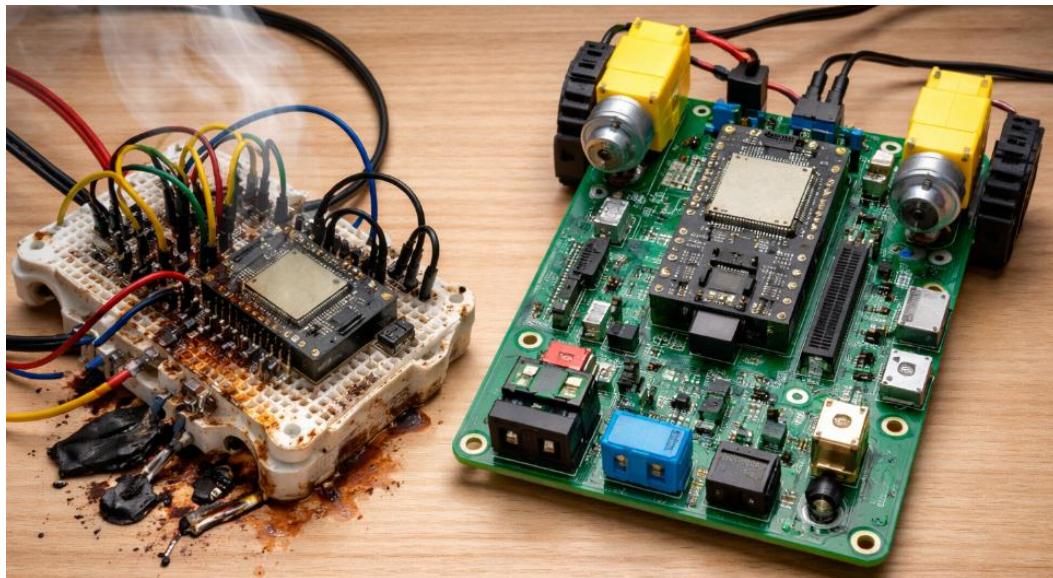
Why Breadboarding No Longer Works

Breadboards are useful for learning basic concepts, but they are not representative of how modern embedded systems are designed, built, or deployed. Breadboarding encourages thinking in ideal terms—wires as perfect connections, short paths, and components behaving exactly as shown in textbooks.

In real products, electronics do not live on breadboards. They live on printed circuit boards where trace length, spacing, ground return paths, power distribution, and component placement all affect behavior. Many of the problems engineers face today—such as noise, timing issues, EMI, and power integrity—do not appear in the **simplified, low-speed circuits typically built on breadboards**. These issues become important when designs scale into real products and are implemented on PCBs.

Because of this, relying on breadboarding as the primary learning tool no longer prepares students for modern embedded engineering. Learning to design and build PCBs early shifts thinking from “does it work on the bench?” to “will it work as a product?”

This course intentionally moves beyond breadboarding and focuses on PCB-based design so students develop the skills and mindset used in industry today.



Each student will design a robotic board. While your boards are being manufactured, we focus on how your designs actually behave once they exist as hardware. Concepts like resistance, inductance, and capacitance (RLC) are no longer abstract symbols; they become properties of the PCB itself. Copper traces, ground planes, power routing, and component placement all contribute real RLC effects.

Many beginner courses introduce RLC circuits using idealized components on breadboards. In real products, every component must be viewed more completely. Even a simple resistor is no longer “just a resistance”—its value changes with temperature, and it also has small but real capacitance and inductance. As boards heat and cool (especially in environments like automotive, aerospace, and medical), these effects matter.

Viewing every component and connection as an RLC system—from the PCB traces down to the silicon—bridges what you learn in class with how real embedded systems are designed.

Here's why experience (doing) matters:

- **Stand out amongst your peers.** Most engineering students have never designed a real circuit board. This is real preparation for your first embedded and circuit engineering job, building a foundation of confidence that comes only from making something yourself.
- **Skills you can demonstrate, not just list.** When in an interview, showing a PCB and working product you designed proves real ability that resume bullet points can't. You'll learn basic schematic design, printed circuit board (PCB) layout, soldering (surface mount devices (SMD) and through-hole), embedded programming, and wireless control—practical skills put to work at engineering firms, startups, and research.

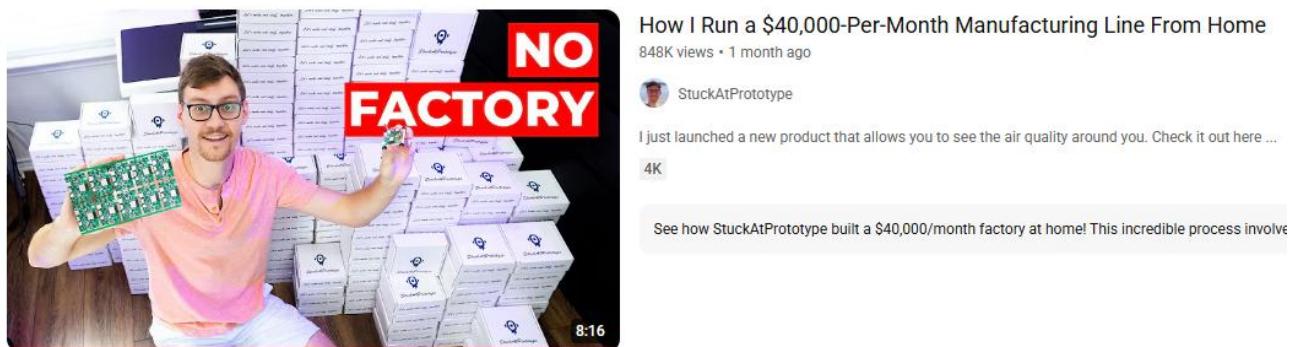
In short: this workshop turns curiosity into capability—and gives you real hardware, real experience, and real proof that you can design and build the future, starting now.

- **Leading the new wave of local innovation.** While much electronics manufacturing moved overseas in the early 2000s, onshoring is now accelerating. This shift is creating strong demand for embedded engineers who can design, prototype, and build technology locally. The next wave of hardware innovation starts with people who can actually build.



- **How you become a hardware creator.** By learning to design your own boards from scratch, you gain the ability to prototype ideas quickly, launch products, and build things people actually want or need. Welcome to the field of embedded engineering!

Example video: <https://youtu.be/RyKvEcgcswg?si=sczsOFtDCqE3MOzw>



WHAT IS KICAD?

KiCAD is a **free, open-source** program that lets you design:

- Electronic schematics (the circuit diagram)
- Printed circuit boards (PCBs) – the actual physical boards

It's powerful enough for most hobbyist, school, and even many professional projects.

Why learn KiCAD?

- It's completely free forever (no licenses or subscriptions)
- It works great for beginner-to-intermediate boards (like the ones we make in class)
- Skills you learn in KiCAD transfer easily to expensive paid tools used in industry

When do engineers use paid software instead? For very advanced, high-speed, or super-complex designs (e.g., 5G phones, aerospace, or high-frequency RF boards), engineers use paid tools that have extra simulation features that KiCAD doesn't include. For everything we do in this workshop, **KiCAD is more than enough** – and it's what many startups and pros actually use!

PCB CAD Tools – The Two Main Parts

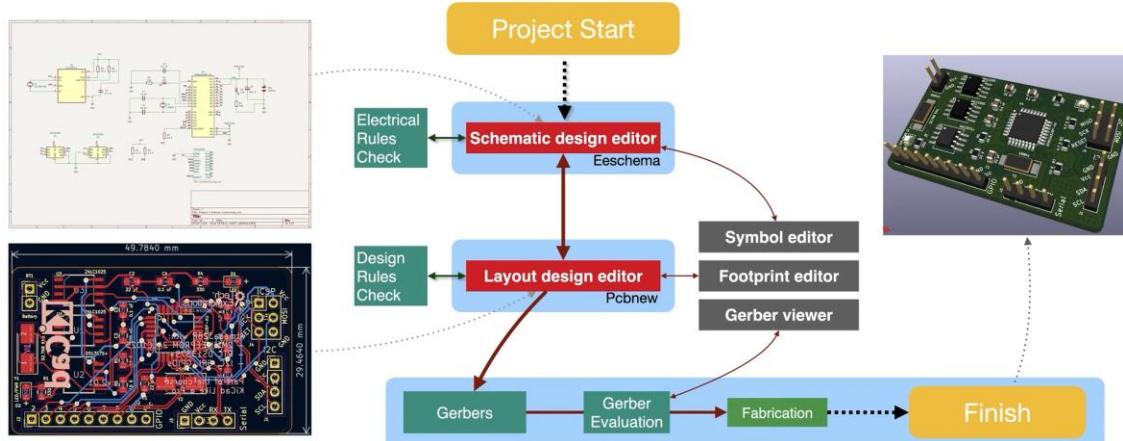
- **Schematic** → The wiring diagram. It shows which components you use and how they are electrically connected (like a circuit diagram on paper).
- **PCB Layout** → The real board design. It shows exactly where each component sits on the physical board and how the copper tracks connect them.

Students may use this guide alongside the official KiCAD documentation. For detailed KiCAD instructions, visit:

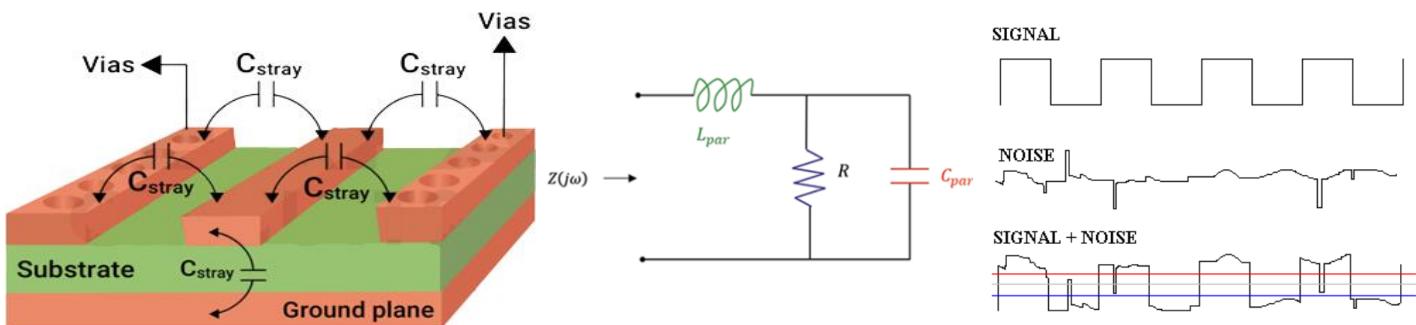
- Get Started: [Getting Started in KiCAD | 8.0 | English | Documentation | KiCAD](#)
- Schematics: [Schematic Editor | 9.0 | English | Documentation | KiCAD](#)
- PCB Layout: [PCB Editor | 9.0 | English | Documentation | KiCAD](#)
- You can download KiCAD at <https://www.kicad.org/download>.
- Full documentation: <https://docs.kicad.org>

Basic Concepts and Workflow

The Kicad design workflow



Why PCBs Must Be Designed for the Real World (No Such Thing as Ideal Circuits)



The Big Idea: In the real world, nothing on a PCB is ideal.

Every trace wire, copper pad, copper via hole, and component behaves not only by its intended function, but also as a resistor, inductor, and capacitor at the same time.

Modern PCB design means thinking beyond symbols.

How Engineers Used to Think (and Why That Changed)

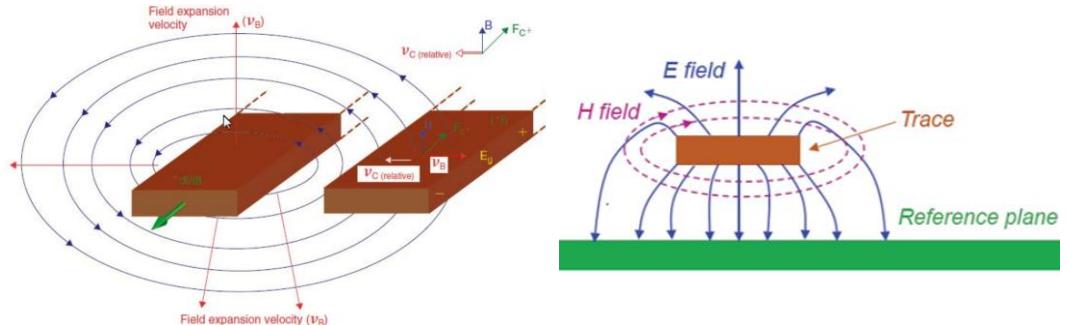
Forty years ago, engineers could often think of PCB design in ideal terms:

- Wires were treated as **perfect connections (0Ω)**
- Switches were simply **on or off**
- Components behaved close to their textbook models

40 years ago, that worked because:

- Speeds were slow

- Frequencies were low
- Currents were modest



That world no longer exists.

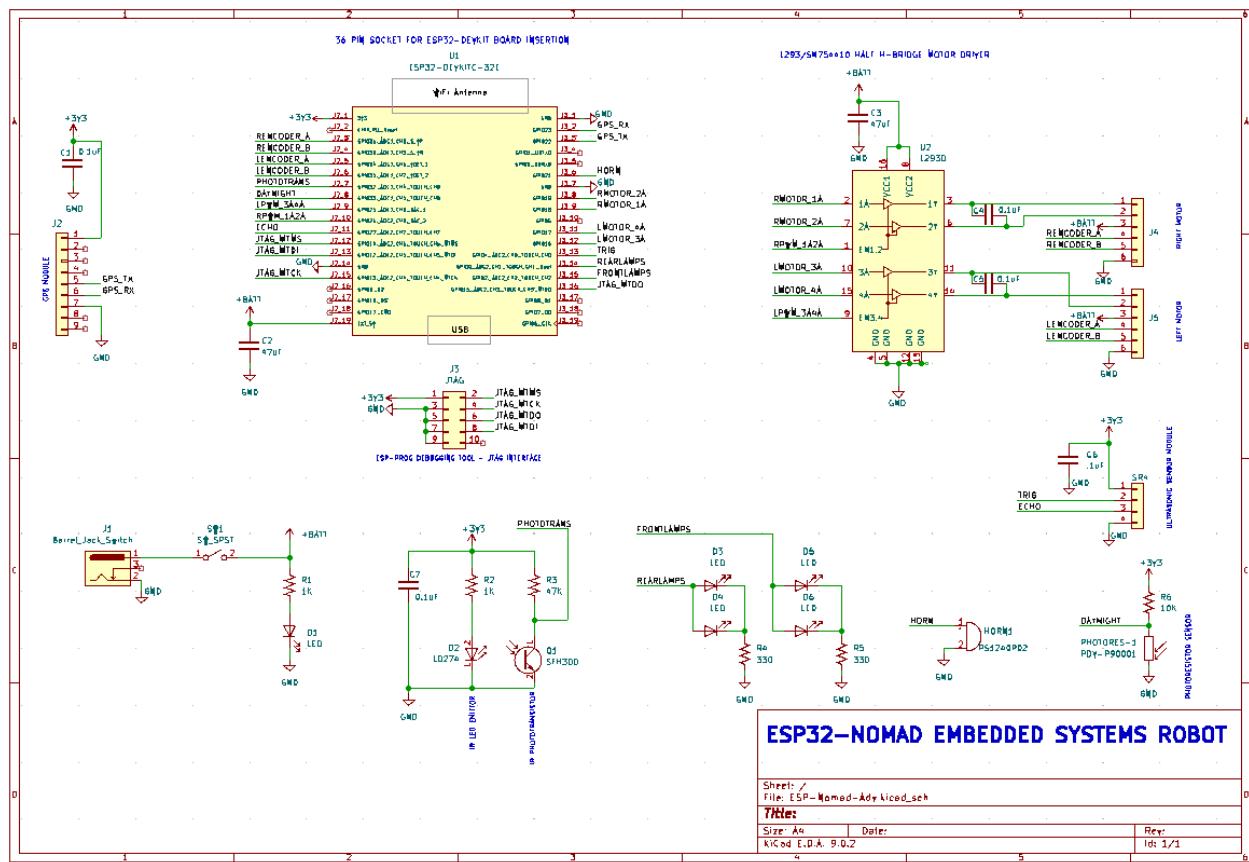
Schematic – Your Circuit Blueprint

In this project, you'll design a real electronic product that works in the real world and is designed with the care and creativity of art—visually expressive.

Your first step is the **schematic** – this is the wiring diagram that tells KiCAD exactly which parts you're using and how they're connected.

Your circuit will include:

- A **36-pin socket** for installing the **ESP32 microcontroller developer board**.
- A **barrel-jack power input, on/off switch, power LED, and front and rear lights**
- A **motor driver** with **plug-in connectors** for motor power and motor encoders (motor encoders measure rotation and speed).
- Built-in **sensors**, including:
 - **GPS** for position tracking
 - **Infrared (IR) sensors** for basic line following
 - An **ultrasonic sensor** for detecting nearby objects and avoidance
 - A **daylight sensor** to measure light levels



PCB Layout – Where Engineering Meets Art

In this project, your PCB layout must do two things:

- Work correctly.**

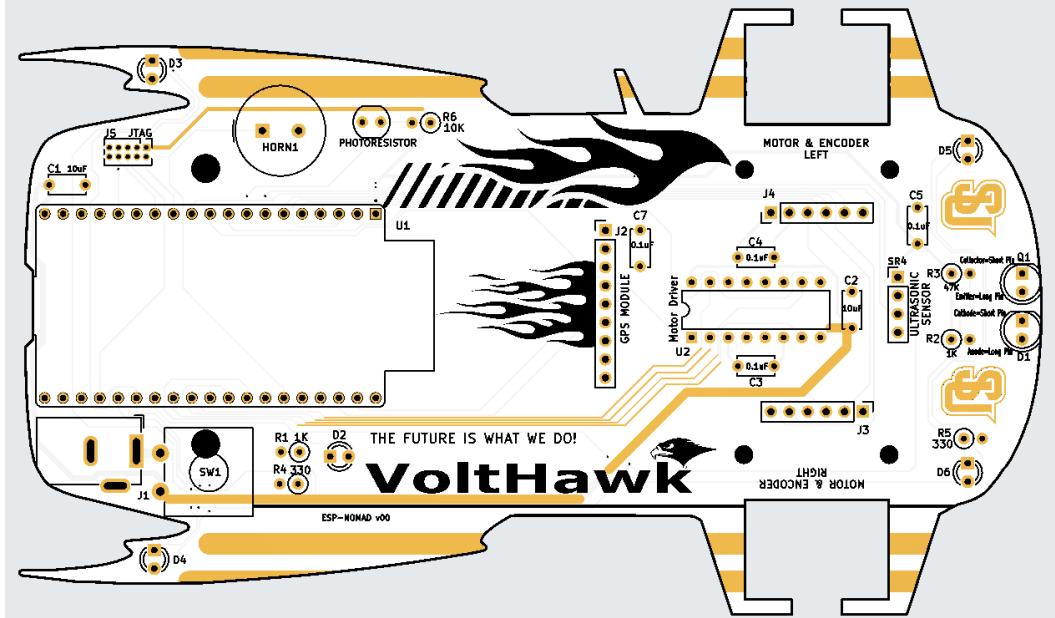
If your schematic is wrong, your board will not work. Double-check every connection before routing.

- Fit the real world.**

Components must be placed so the board actually works as a product: motors at the front, connectors accessible, the power switch easy to reach, and everything fits.

But real engineers don't just design circuit boards that function — they design products people enjoy using. A good design turns a working circuit into something people want to hold, use, and keep.

An Example PCB Layout



A classic example is the first iPhone. The first iPhone wasn't just a better phone — it felt like a piece of jewelry. Every curve, material, and detail showed that engineering and beauty can live together.

For this project, **function comes first**. You're working on a tight schedule, so the goal is a clean, correct, and a completely boring design is acceptable by the deadline. That said, you're the engineer in the seat. You must manage your time, tasks, schedules, and pressures from managers. What can you do if inspired and have extra time to refine layouts, improve aesthetics, and make designs that are both flawless **and** visually impressive.

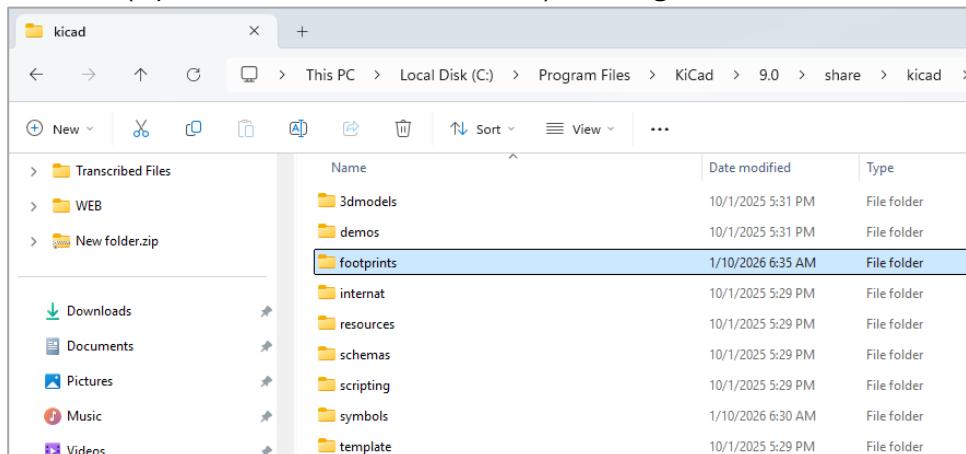


STEP 1. DOWNLOAD THE INNOVATION LAB FOOTPRINT LIBRARY

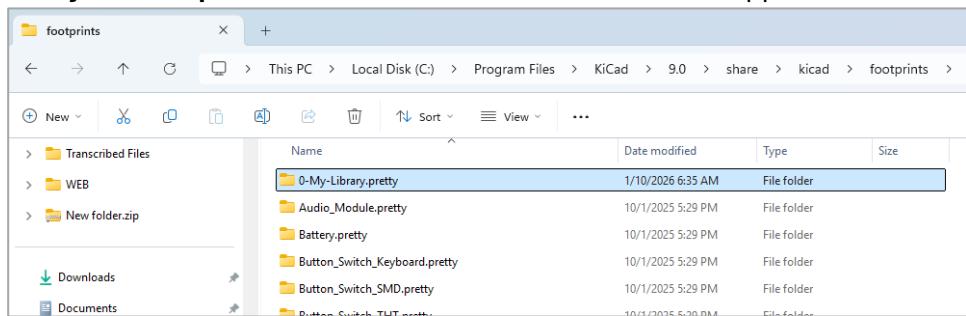
1. In Canvas [ECEGR 3210 01](#) site → **Files**, download the file **0-My-Library.pretty.zip** into the Windows Download folder.
2. Unzip the .zip file using **Windows File Explorer**. Feel free to delete the zip folder.

Footprint Library: 0-My-Library.pretty

3. Copy and paste **.pretty** folder into:
 - (if Bannan lab computers) **C:\KiCAD\9\share\KiCad\footprints**
 - (if personal device-default KiCAD): **C:\Program Files\KiCad\9.0\share\kicad\footprints**



4. **Verify the Footprints folder:** Confirm that the folder now appears inside the **Footprints** directory.

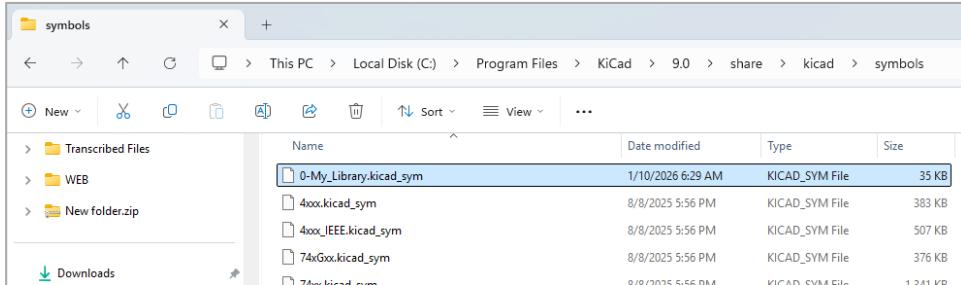


STEP 2. DOWNLOAD THE INNOVATION LAB SYMBOL LIBRARY

1. In Canvas [ECEGR 3210 01](#) site → **Files**, download the file **0-My_Library.kicad_sym** into the Windows Download folder.

Symbol Library: 0-My_Library.kicad_sym

2. Copy and paste .sym folder into:
 - (if Bannan lab computers) **C:\KiCAD\9\share\KiCad\symbols**
 - (if personal device-default KiCAD): **C:\Program Files\KiCad\9.0\share\kicad\symbols**
3. **Verify the Symbols folder:** Confirm that the folder now appears inside the **Symbols** directory.



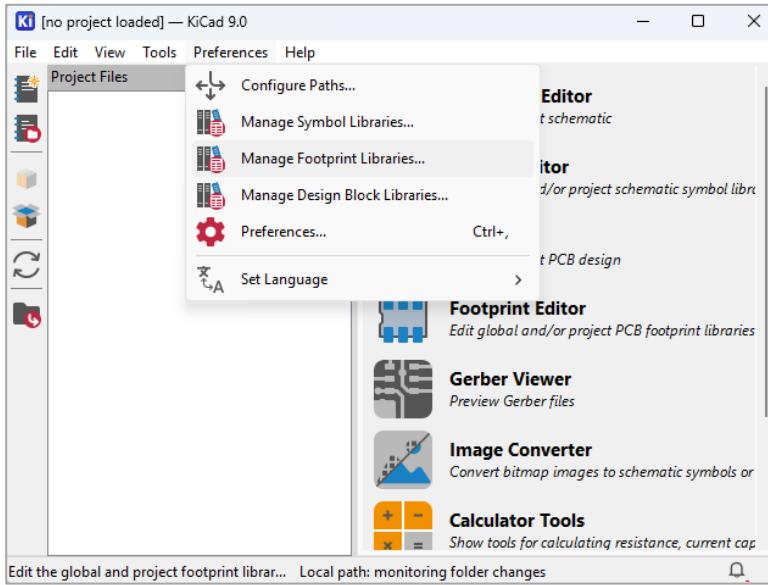
STEP 3. STARTING KICAD ON WINDOWS

1. In the Windows search bar, type KiCAD and launch the application.
2. During first launch, several setup pop-up windows may appear.
 - When prompted to copy default libraries or settings, select **Copy default...** and click **OK**.
 - If you are asked about optional participation or data sharing, you may decline or skip those options.
3. Continue through the remaining prompts, using common sense—accept the default options unless you have a reason or instruction to change them.

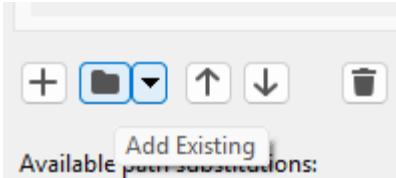
STEP 4. IMPORT THE INNOVATION LAB PROJECT SYMBOL & FOOTPRINT LIBRARY

Manage Footprint Libraries

1. Click Preferences → Manage Footprint Libraries...



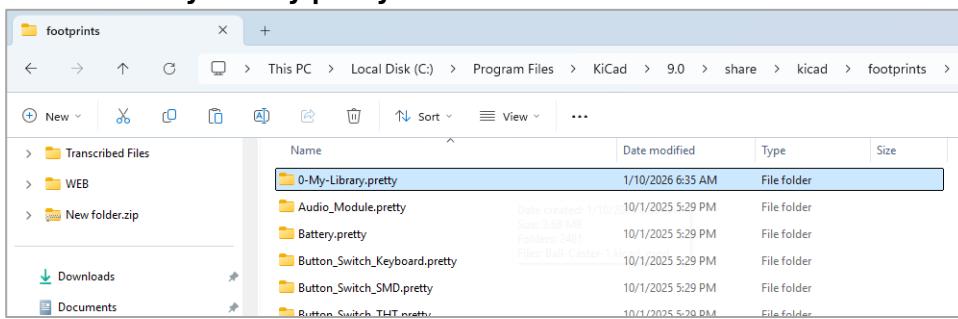
2. In the bottom left window, click the Add Existing icon at the bottom.



3. Navigate to the **0-My-Library.pretty** folder.

- (if Bannan lab computers) **C:\KiCAD\9\share\KiCad\footprints**
- (if personal device-default KiCAD): **C:\Program Files\KiCad\9.0\share\kicad\footprints**

Select the **0-My-Library.pretty**.



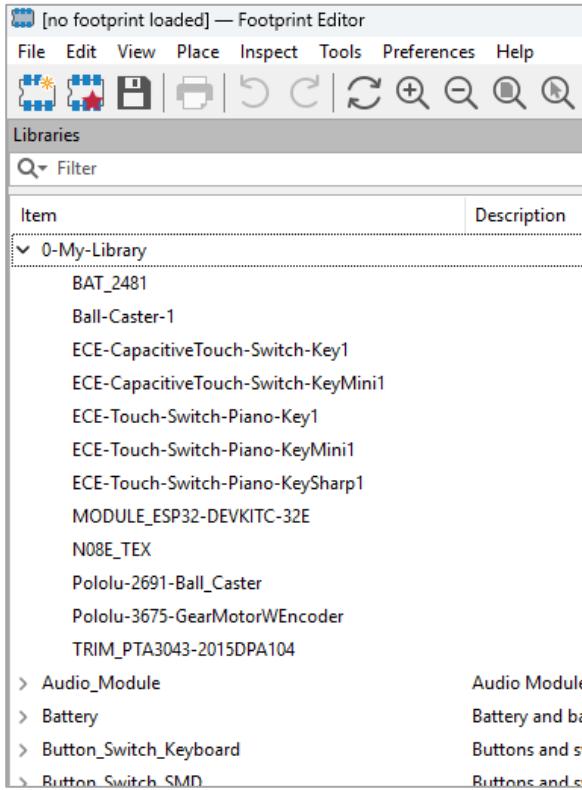
4. The ECE library should now be available to you in KiCAD.

To check the Footprint library is installed:

5. Click the **Footprint Editor** icon and check that the footprints are available.

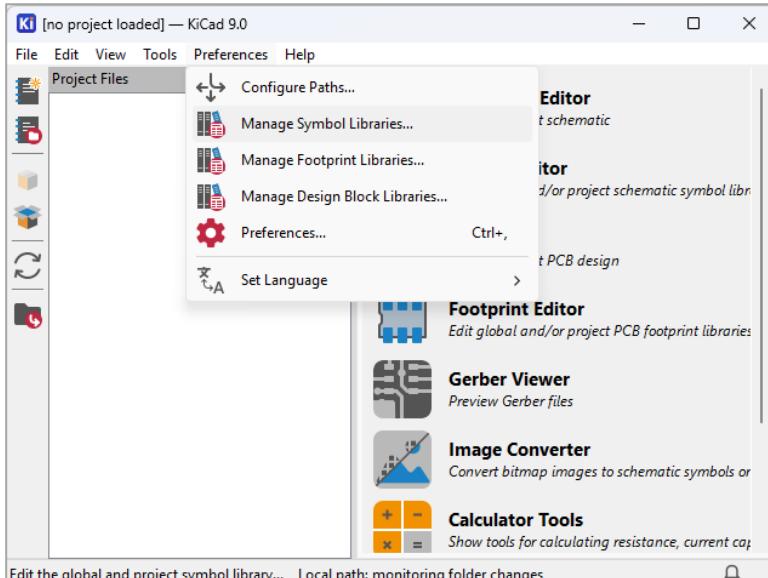


6. 0-My-Library files:

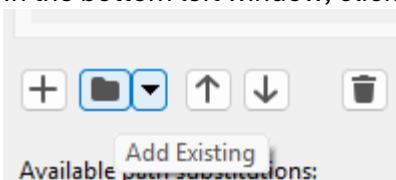


Manage Symbols Libraries

7. Click Preferences → Manage Symbols Libraries...



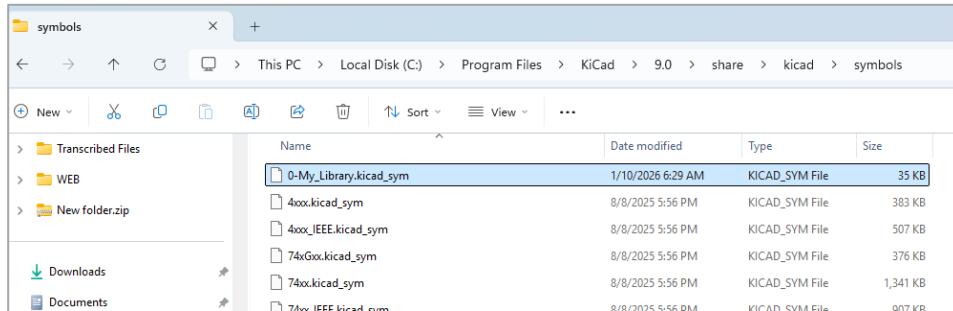
8. In the bottom left window, click the Add Existing icon at the bottom.



9. Navigate to the **0-My-Library.kicad_sym** folder.

- (if Bannan lab computers) **C:\KiCAD\9\share\KiCad\symbols**
- (if personal device-default KiCAD): **C:\Program Files\KiCad\9.0\share\kicad\symbols**

Select the 0-My-Library.kicad_sym



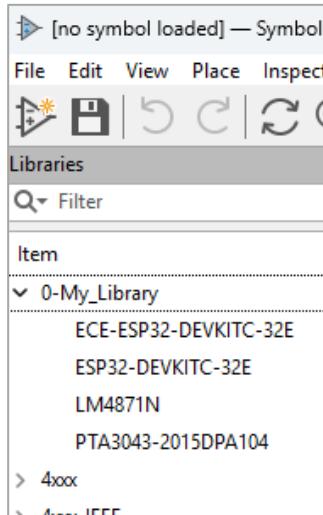
10. The ECE library should now be available to you in KiCAD.

To check the symbol library is installed:

11. Click the **Footprint Editor** icon and check that the footprints are available.



12. 0-My-Library files:



GET STARTED – ESP32-ROBOT PROJECT

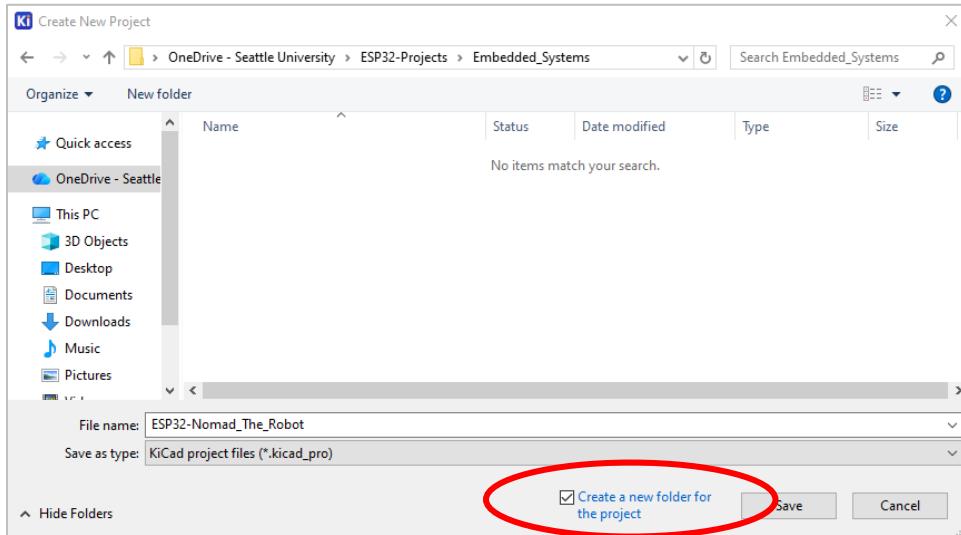
Before beginning, please use dual monitors. Dual monitors let engineers work faster and make fewer mistakes by keeping designs, datasheets, code, and documentation visible at the same time — a clear productivity win for any engineer. Using tools like dual monitors signals that a new engineer is thinking about efficient, professional workflows, which is something managers tend to notice early on.

STEP 5. OPEN KICAD ON ONE MONITOR AND THIS DOCUMENT GUIDE ON THE OTHER MONITOR.

STEP 6. CREATING A NEW PROJECT

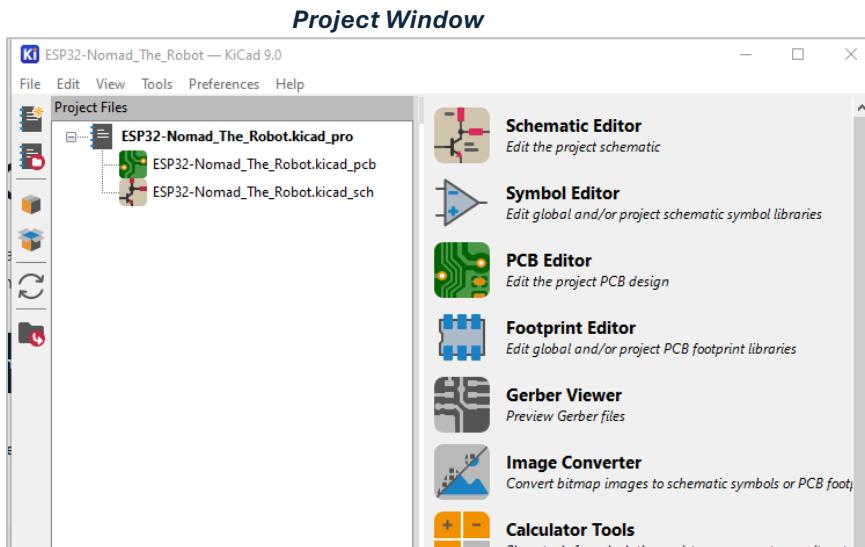
1. Open KiCAD → click File → New Project.
2. Select the OneDrive folder, give your project a name, such as My-Robot.
3. Checkmark the **Create a new folder for the project** box.

4. Click Save.



Your project folder now contains the KiCAD file contents:

1. kicad_pro – the project file
2. .kicad_pcb – the circuit board layout
3. .kicad_sch – the schematic



SCHEMATICS EDITOR

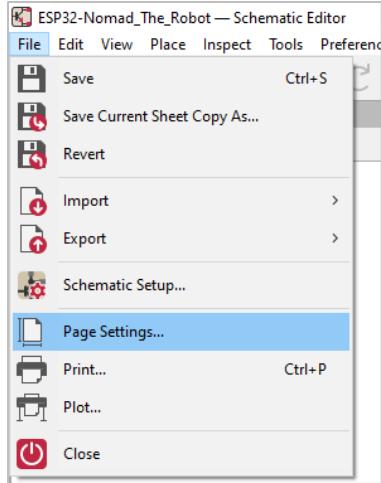
STEP 7. OPEN THE SCHEMATIC EDITOR

1. From the **Project Window**, click on the **Schematic Editor** icon to begin.



STEP 8. SET SCHEMATIC PAGE SIZE

1. File → Page Settings...
2. Set **Paper size** → A3
3. **OK**

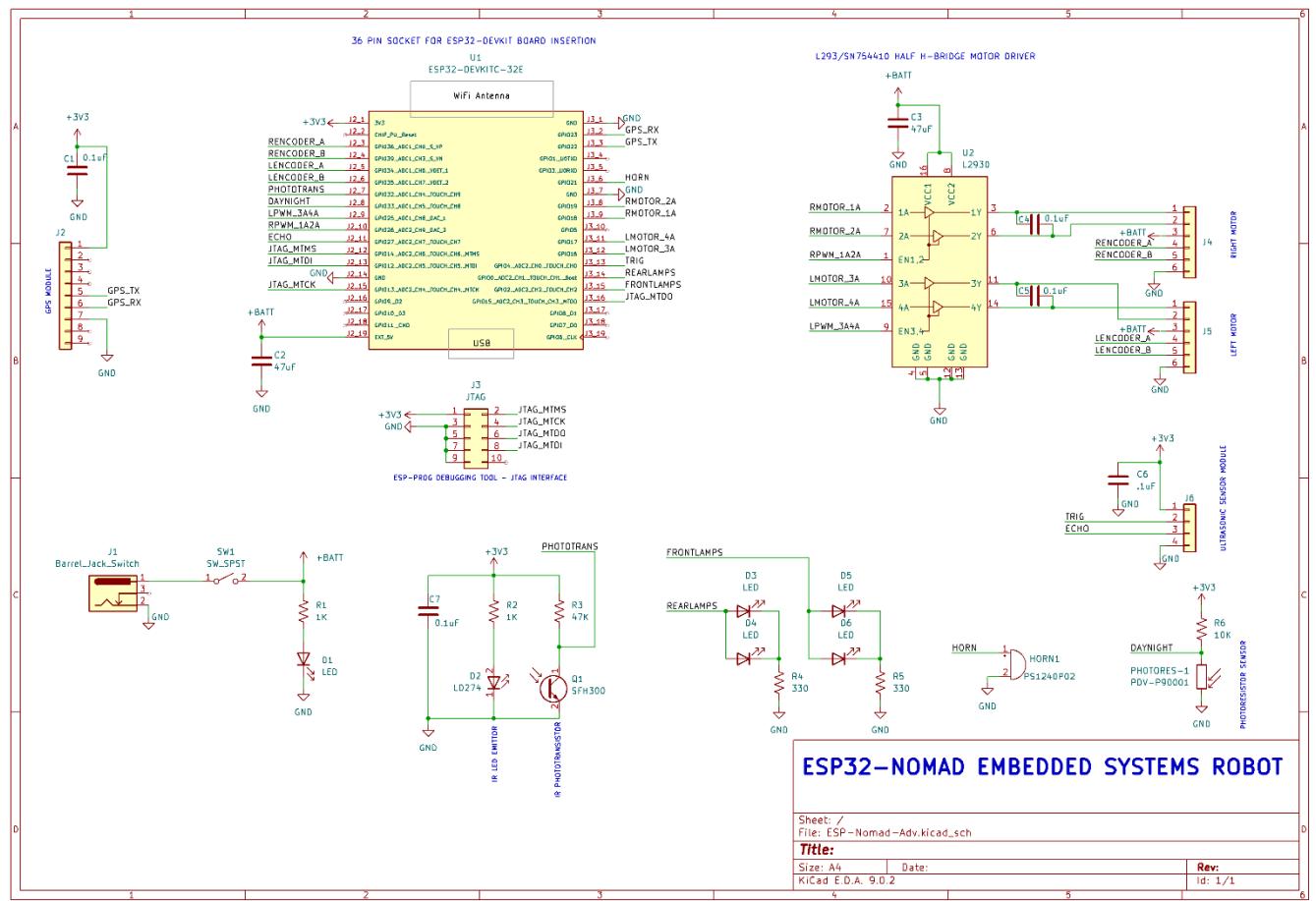


STEP 9. MOUSE POINTER FULL WINDOW CROSSHAIRS

In the **Schematic Editor**, click the **Place Full-Window Crosshairs** icon on the left toolbar. This changes the mouse pointer crosshair style to span the entire window, making it easier to place parts, especially wires and net labels accurately in the next steps.

STEP 10. DRAWING THE SCHEMATIC

Below is the schematic you must build—[link to full-size schematic here](#).

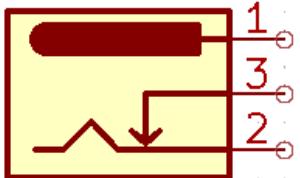
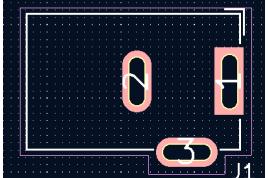
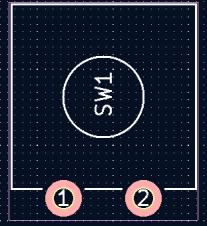


STEP 11. UNDERSTANDING SYMBOLS AND FOOTPRINTS

In PCB design software, every part has two **elements**.

- The **schematic symbol** is the logical drawing you place in your schematic (e.g., a battery symbol).

- The **footprint** is the pattern of copper pads and holes where the real physical parts gets **soldered to the board**, giving it a physical and electrical connection to your circuit board.

	Symbol	Footprint	Physical Part
Barrel Jack Power Input Connector			
Power Switch			

STEP 12. COMPONENTS ARE NOT IDEAL

As you place components on the schematic and PCB, view that **every part, trace, and connection behaves as resistance, inductance, and capacitance (RLC)**. And your physical layout choices directly affect how the circuit behaves in the real world.

For example, the **power switch** shown above is not a perfect on/off device. When closed, there is a small but real resistance between **pin 1 and pin 2**.

Exercise:

- Open the DigiKey link below.
- View the switch datasheet link and open the datasheet.
- Find the **contact resistance** specification.

Now consider the impact:

- If the battery input is **6 V**,
- How much voltage drop occurs across the switch at **100 mA**?
- What about at **1 A**?
- How might this affect the voltage seen by the **ESP32** and the other circuits?
- Start thinking in this way as you begin your schematic.

DigiKey link:

- <https://www.digikey.com/short/83998mfh>

STEP 13. FINDING & SELECTING PARTS – BATTERY INPUT CONNECTOR

The Engineering Process: Finding and Selecting Parts

The first step in any electronics design is choosing the right components. Engineers typically search electronics distributors (such as DigiKey or Mouser) or manufacturer websites to find parts that meet electrical, mechanical, cost, and availability requirements.

For this project, we will use a **4 x AAA battery holder** 6.0-volt battery source. To connect the board to the battery holder we need to add a **Barrel Jack connector**.

4 x AAA Battery Holder + Power Cable Adapter

Barrel Jack Power Connector



Holder: <https://www.digikey.com/short/mj0tjmpc>

Adapter: <https://www.digikey.com/short/t1v9m3d2>



Connector: <https://www.digikey.com/short/r282h4cz>

To see how this process works in practice:

1. Open the **DigiKey search results page**:

<https://www.digikey.com/short/9f9z2pfz>

This page shows many parts that meet the search criteria.

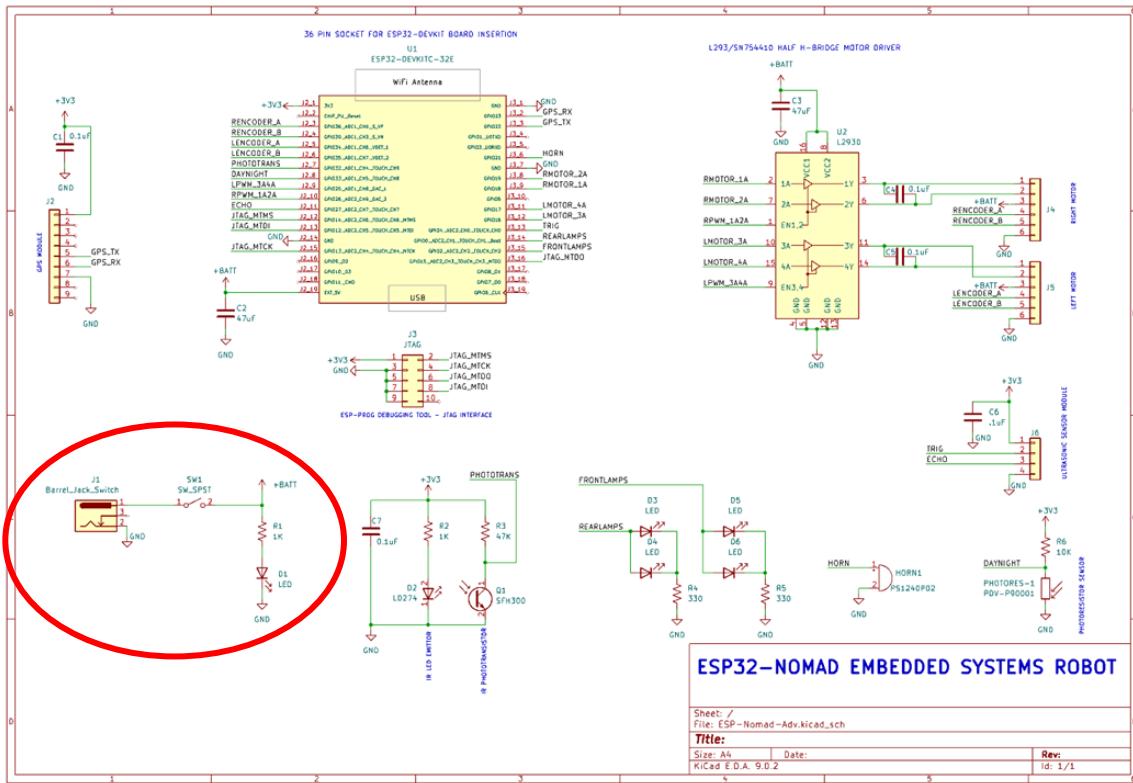
2. Scroll down the page to view the list of available components to choose from.
 3. Hover over the part images to see larger previews and compare different options.
 4. For this workshop, we have already selected a specific battery holder from the list. It was chosen because it is cost-effective, currently in our ECE stock, and already included in the KiCAD symbol and footprint libraries, so no importing is required.
 5. Click this direct DigiKey link to view the selected part:
<https://www.digikey.com/short/r282h4cz>
- Then click the Datasheet link to review detailed specifications.
6. **Keep in mind:** If a part isn't included in the KiCAD symbol or footprint libraries, you can often download it directly from the DigiKey product page using the **EDA/CAD Models** link.
 7. **Try it:** On the DigiKey page for a part, look for and click the **EDA/CAD Models** link to access KiCAD-compatible symbols and footprints.
 8. **If needed:** If the part is not available in KiCAD or on DigiKey, check the component manufacturer's website or other trusted online sources. In some cases, you will need to create the symbol and footprint yourself using the KiCAD Library Editor tools.

STEP 14. REFERENCE DESIGNATORS

Reference designators are short labels (such as **R1**, **C2**, **U1**) that KiCAD automatically assigns to identify each component. These labels link the schematic symbols to the PCB layout footprints and ensure each part is uniquely tracked throughout the design.

STEP 15. ADD THE INPUT POWER CIRCUIT SECTION

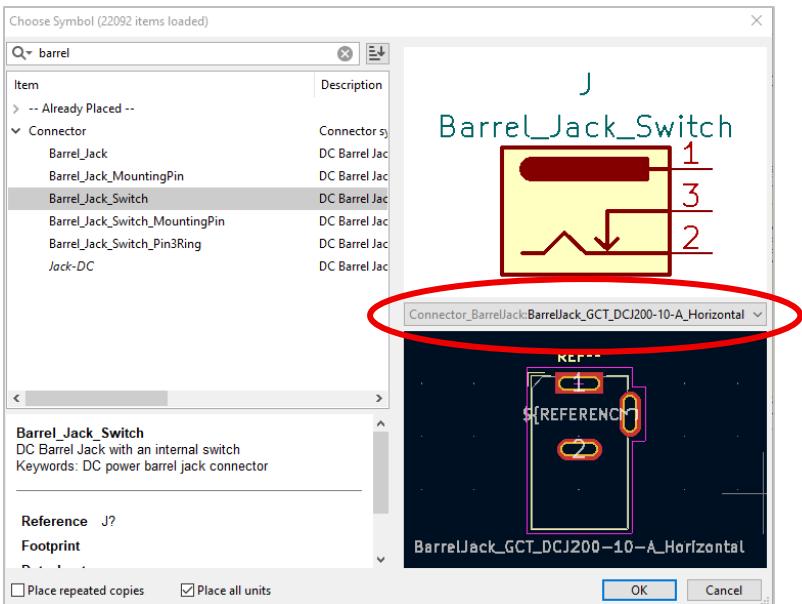
Let's begin building the schematic...



PART 1. PLACE THE BATTERY HOLDER SYMBOL

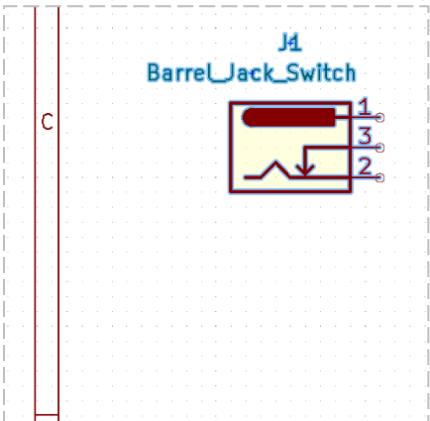
	Symbol	Footprint	Physical Part
Barrel Jack Power Input Connector			

1. In the **Schematic Editor**, navigate to the top menu and select **Place → Place Symbol** or click the **Place Symbol** icon on the right-side toolbar of your schematic, to insert a symbol into your schematic.
2. A search box will appear. Type **barrel** to filter the list.
3. Select the connector **Barrel_Jack_Switch** as shown in the picture below.
4. **Change the footprint field** from a blank to **BarrelJack_GCT_DCJ200-10-A_Horizontal** –see red circle below.
5. Click **OK**.



Design Notes: **Datasheet:** <https://www.digikey.com/short/tjj143hq>

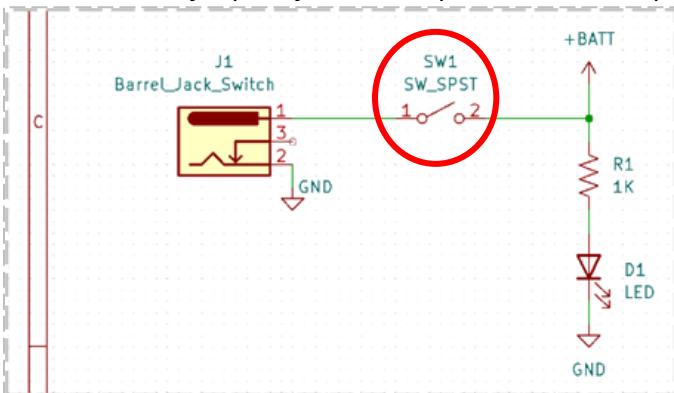
- Click on your schematic sheet to place the symbol—refer to the example schematic for good placement.



- Press **ESC** to exit part placement mode.

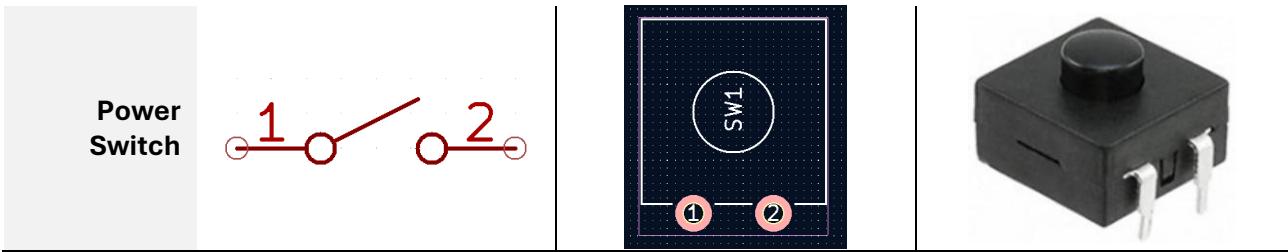
PART 2. ADD THE POWER SWITCH TO THE SCHEMATIC

With the battery input symbol now placed, we must place the on/off switch.

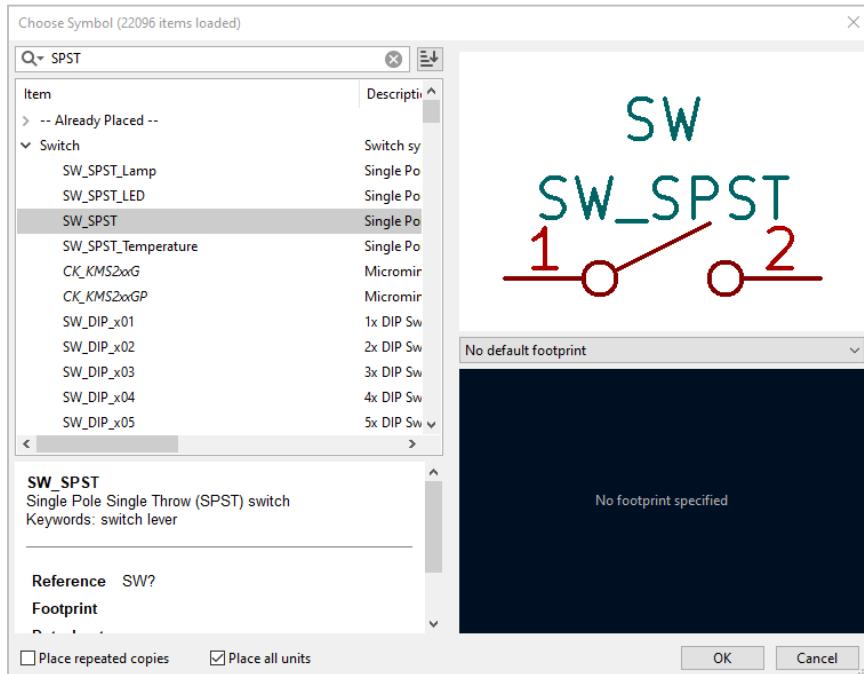


PART 3. PLACE THE POWER SWITCH SYMBOL AND SET THE FOOTPRINT

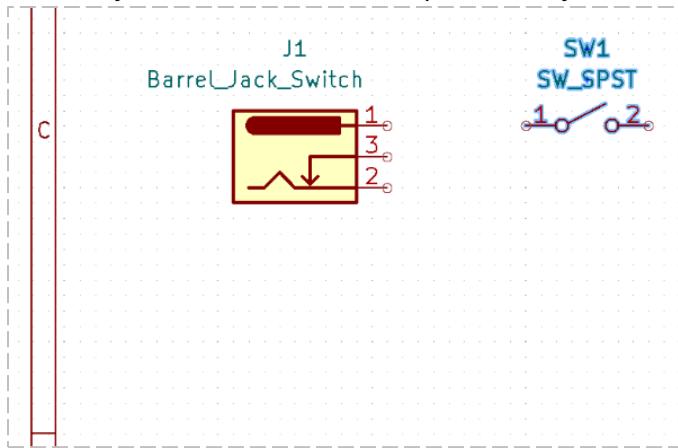
Symbol	Footprint	Physical Part
--------	-----------	---------------



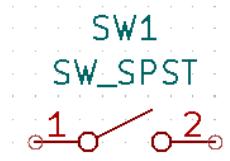
1. In the **Schematic Editor**, navigate to the top menu and select **Place → Place Symbol** or click the **Place Symbol** icon on the right-side toolbar of your schematic, to insert a symbol into your schematic.
2. A search box will appear. Type **SPST** to filter the list.
3. Select the switch **SW_SPST**, as shown in the picture below.
4. **Note:** We will leave the **Footprint** field empty for now, which is why you'll see “**No default footprint**” listed. KiCAD does not include a symbol specifically for our power switch, so we are using a **generic switch symbol** instead. After placing the symbol in the schematic, we'll assign the correct footprint next.



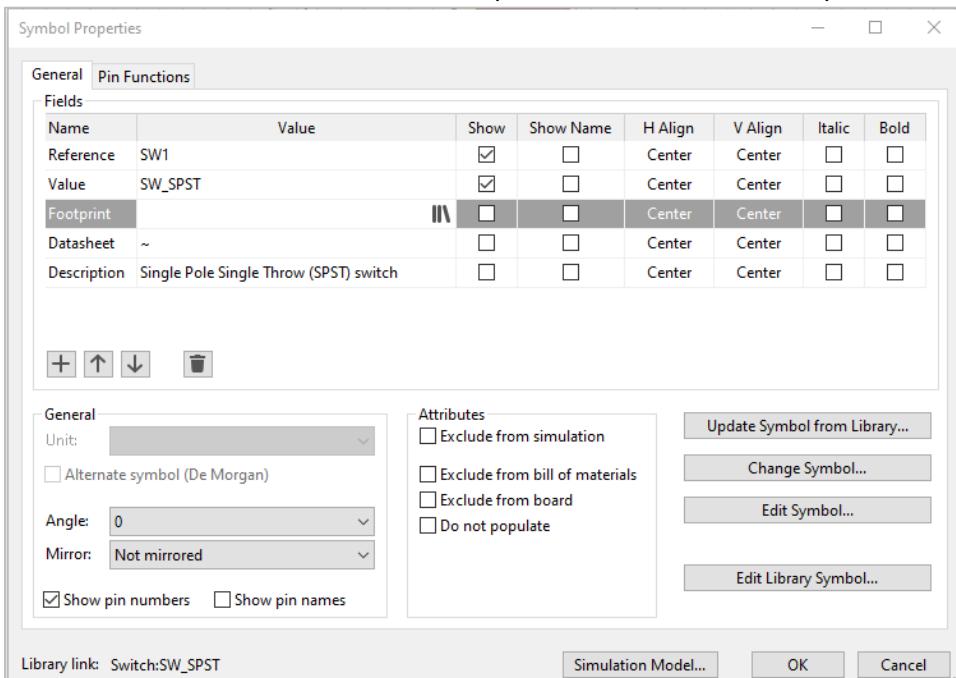
5. Click **OK**.
6. Click on your schematic sheet to place the symbol—refer to the example schematic for good placement.



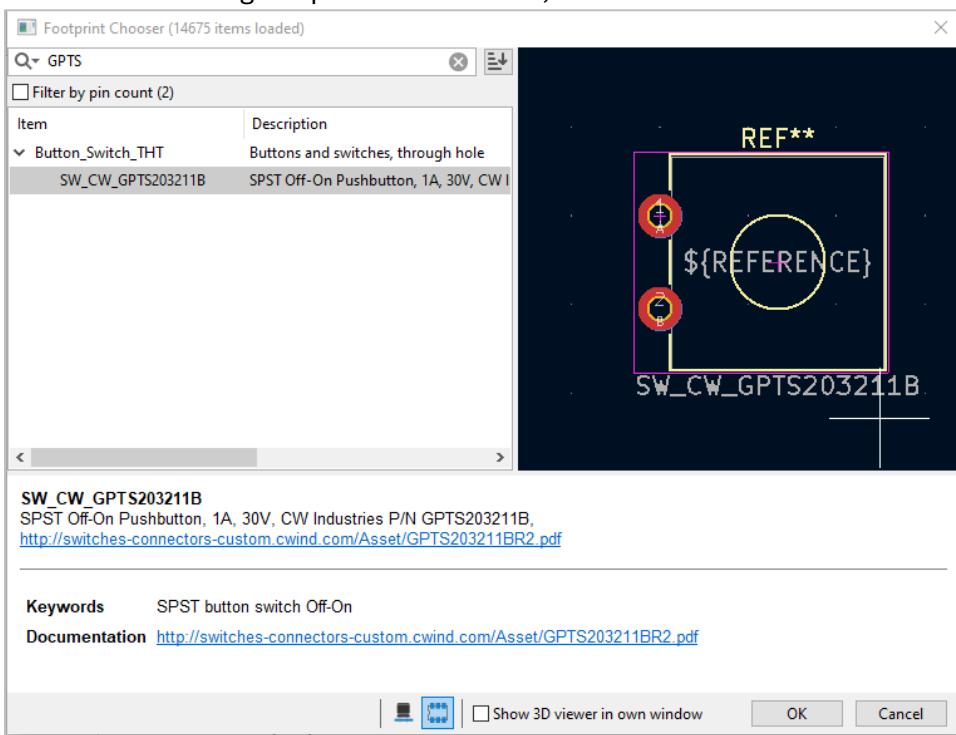
7. Press **ESC** to exit part placement mode.



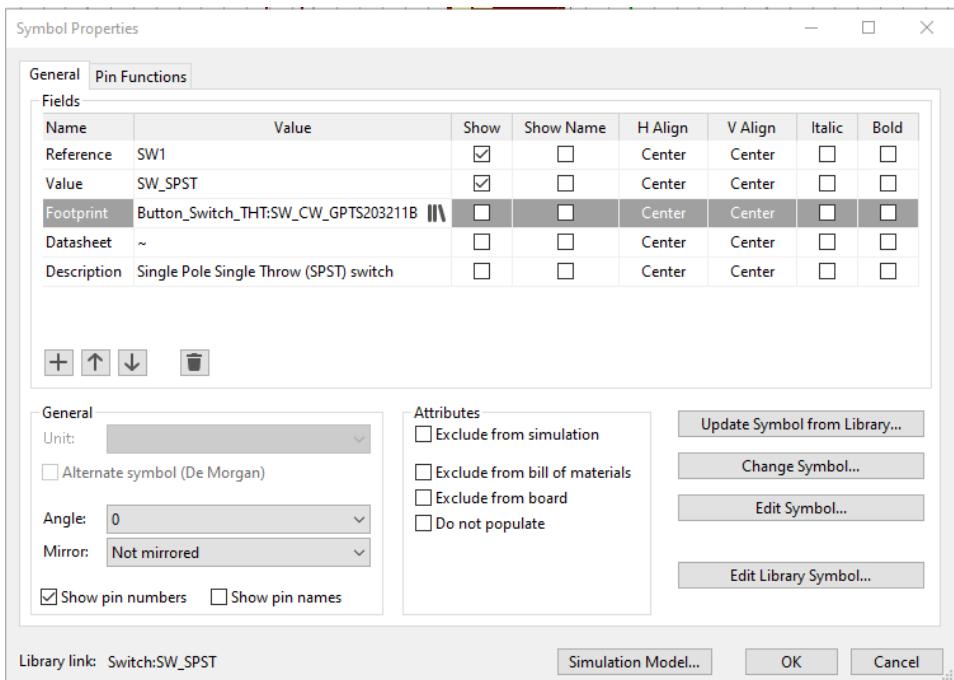
8. Remember, each symbol must have a footprint. Let's add the footprint now. In your schematic, **double click** on the **SW_SPST** symbol to bring up the **Symbol Properties** window.
9. In the **Symbol Properties** window, the **Footprint** field is empty. Let's add the one we need.
10. Click the **book icon**  next to the Footprint field to choose a footprint.



11. In the search filter, type **GPTS** to find the correct footprint.
12. Select the matching footprint shown below, then click **OK**.

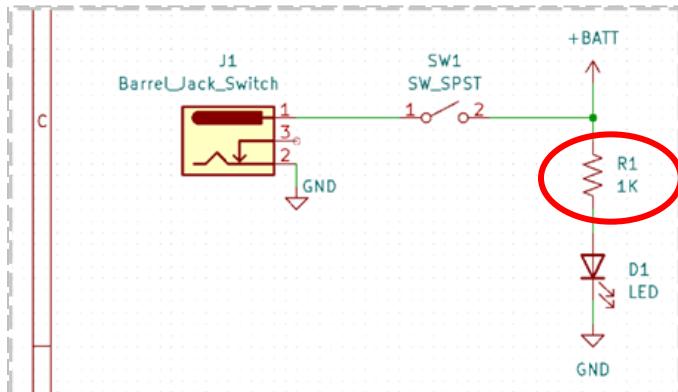


13. The **Footprint** field now shows the correct footprint for the power switch.



14. Click **OK**.

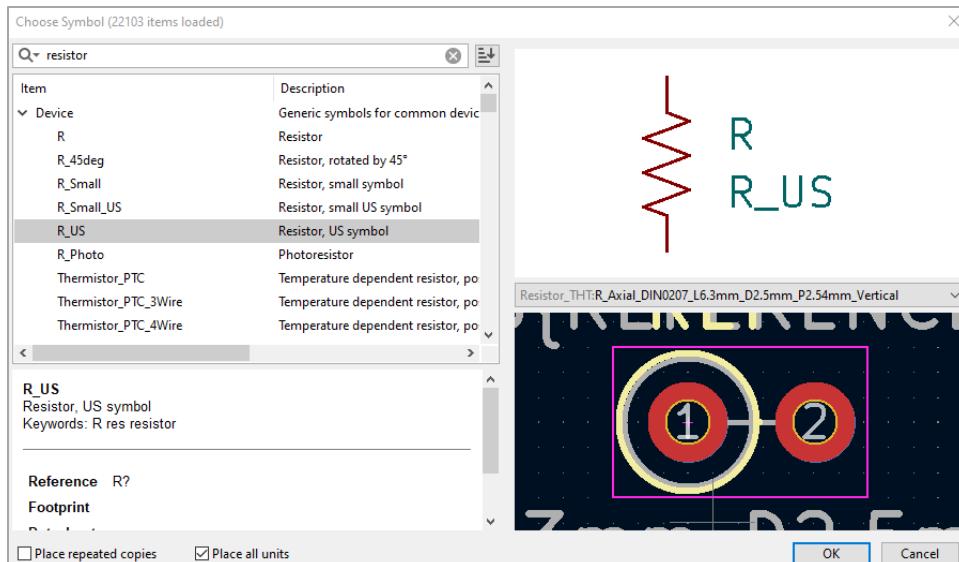
STEP 16. ADD CURRENT LIMITING RESISTOR TO THE LED POWER INDICATOR



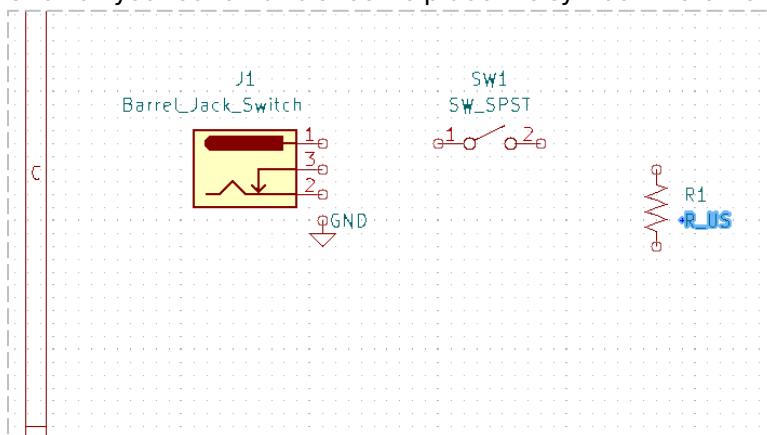
PART 4. PLACE THE RESISTOR

	Symbol	Footprint	Physical Part
Power Switch	 R1 R_US		

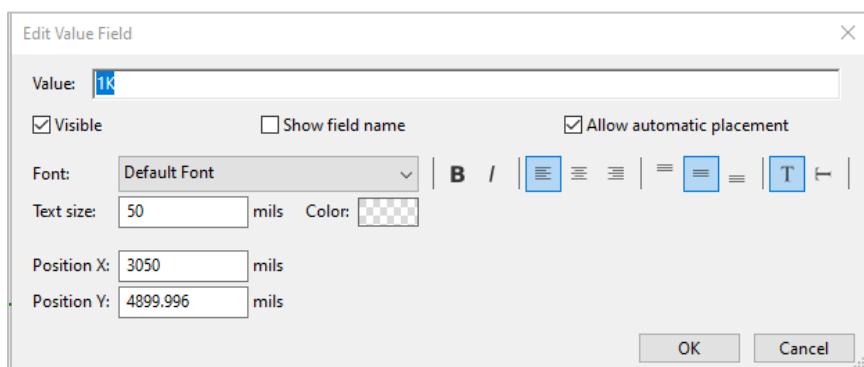
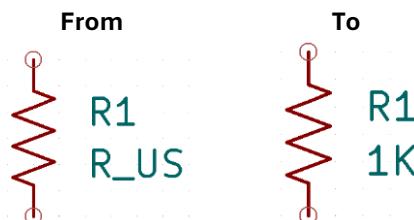
1. In the **Schematic Editor**, navigate to the top menu and select **Place** → **Place Symbol**.
2. A search box will appear. Type **resistor** to filter the list.
3. Select the resistor **R_US** as shown in the picture below.
4. **Change the footprint field** from a blank to **R_Axial_DIN0207_L6.3mm_D2.5mm_P2.54mm_Vertical**.
5. Click **OK**.



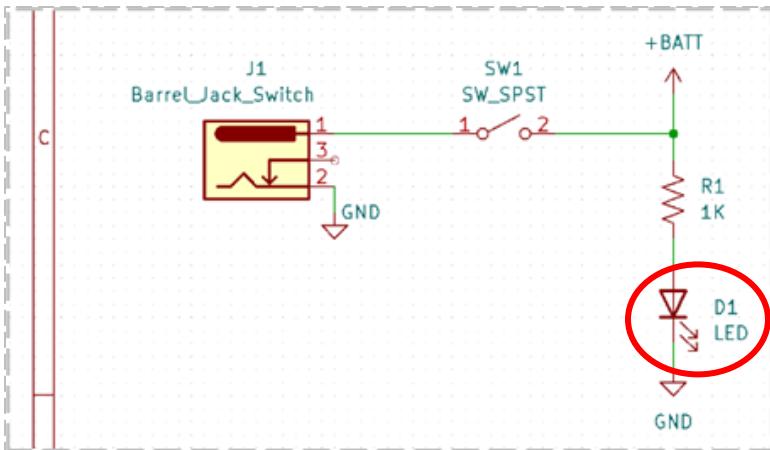
6. Click on your schematic sheet to place the symbol—refer to the example schematic for good placement.



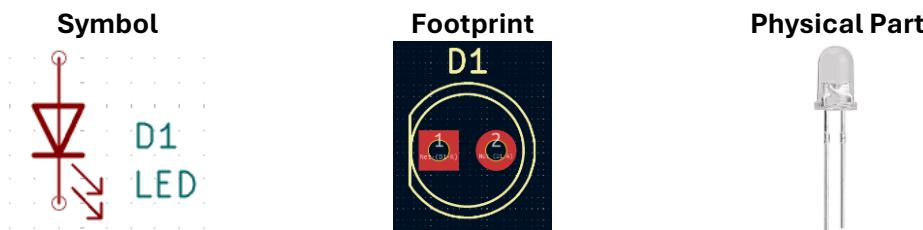
7. Press **ESC** to exit part placement mode.
8. Double-click on the resistor text **R_US** to change the default resistance **Value** field to **1K** for design reference.



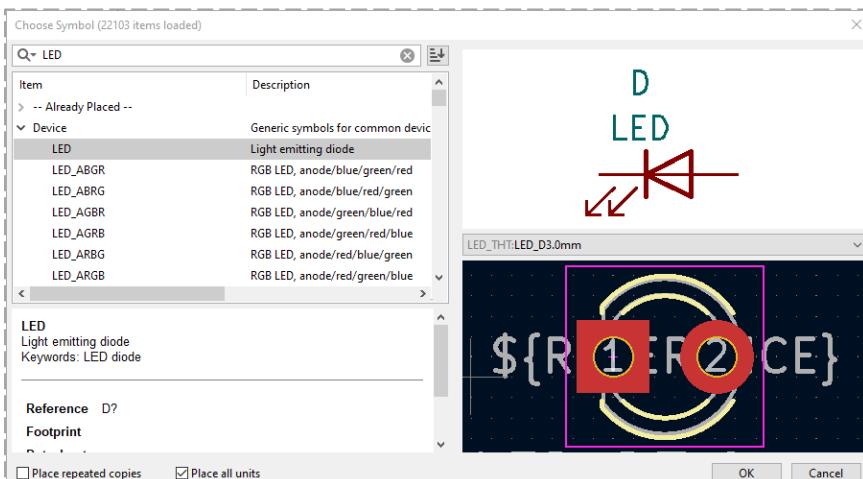
STEP 17. ADD THE LED POWER INDICATOR TO THE SCHEMATIC



PART 5. PLACE THE LED SYMBOL AND SET THE FOOTPRINT



1. In the **Schematic Editor**, navigate to the top menu and select **Place → Place Symbol**.
2. A search box will appear. Type **LED** to filter the list.
3. Select the Switch **LED**, as shown in the picture below.
4. Change the **Footprint** from blank to **LED_THT:LED_D3.0mm**.
5. Click **OK**.



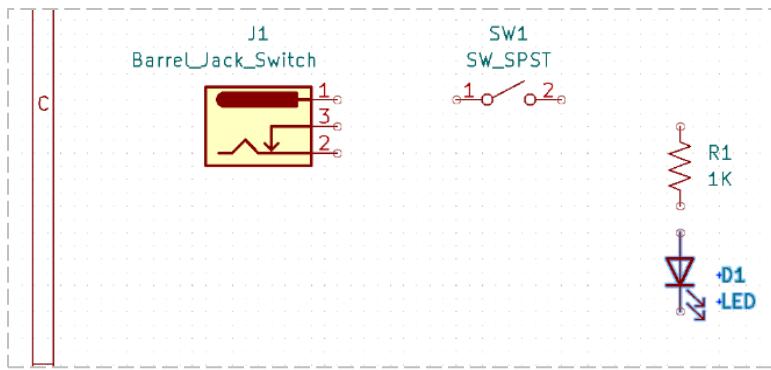
6. Click on your schematic sheet to place the symbol—refer to the example schematic for good placement.
7. Press **ESC** to exit part placement mode.

STEP 18. MOVE, ROTATE, MIRROR COMPONENTS

Note these commands because you will use them when necessary:

- **Move:** Select any symbols to move them.
- **Rotate:** Select the symbol then press **R** to rotate.
- **Mirror:** Select symbol then click the horizontal/vertical mirror icons

8. In this circuit, rotate the LED as shown below.

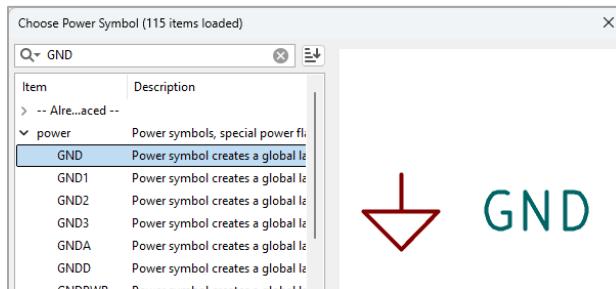


STEP 19. PLACE POWER SYMBOLS

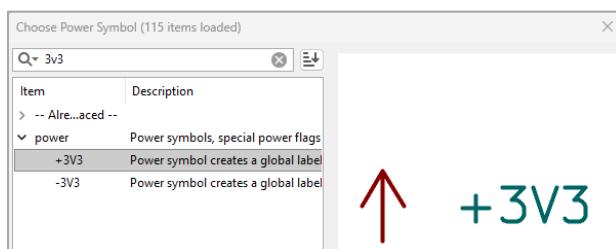
Power symbols act like interconnecting wires but also give name to the connection automatically based on the symbol's label.

1. Select **Place** → **Place Power Symbols** or click the **Place Power Symbols**  icon on the right-side toolbar of your schematic, to insert power symbols into your schematic.

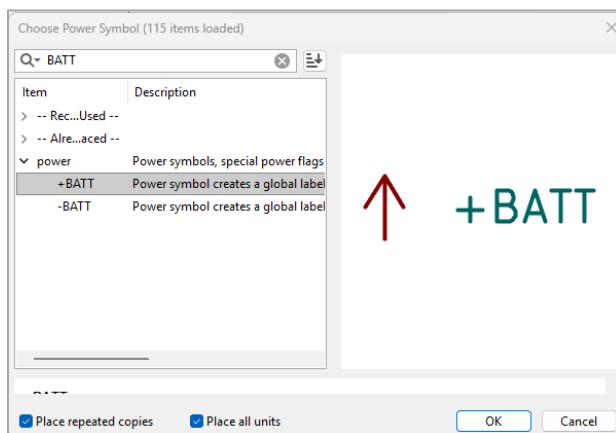
Ground Symbol



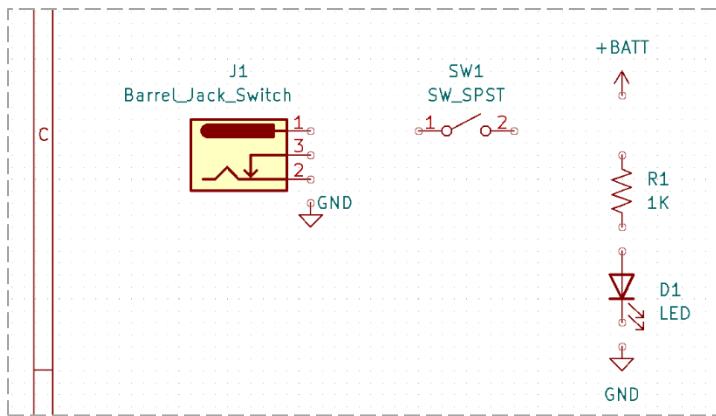
+3.3V Power Symbol



Battery Power Symbol



2. Place all Power Symbols as exemplified below.



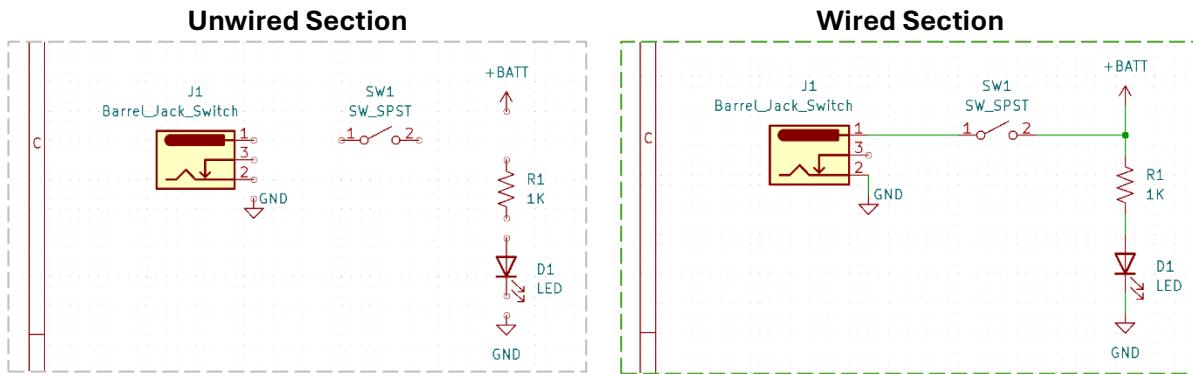
STEP 20. BEGIN DRAWING WIRES TO THE SCHEMATIC

1. To begin connecting pins with a wire, move your mouse and **hover over a pin end** until it highlights with the wire tool , this means the wire will connect correctly.
2. **Click to start** the wire, move to the next pin, **hover until it highlights**, and click again to finish. Zoom in to check connections.
 - When necessary, click the **Draw Wire** tool (the pencil-line icon) icon on the right-side toolbar.
3. Press **Esc** to stop drawing wires.

STEP 21. DELETING WIRES IN THE SCHEMATIC

To delete any wire, select a wire then press the keyboard **Delete** key.

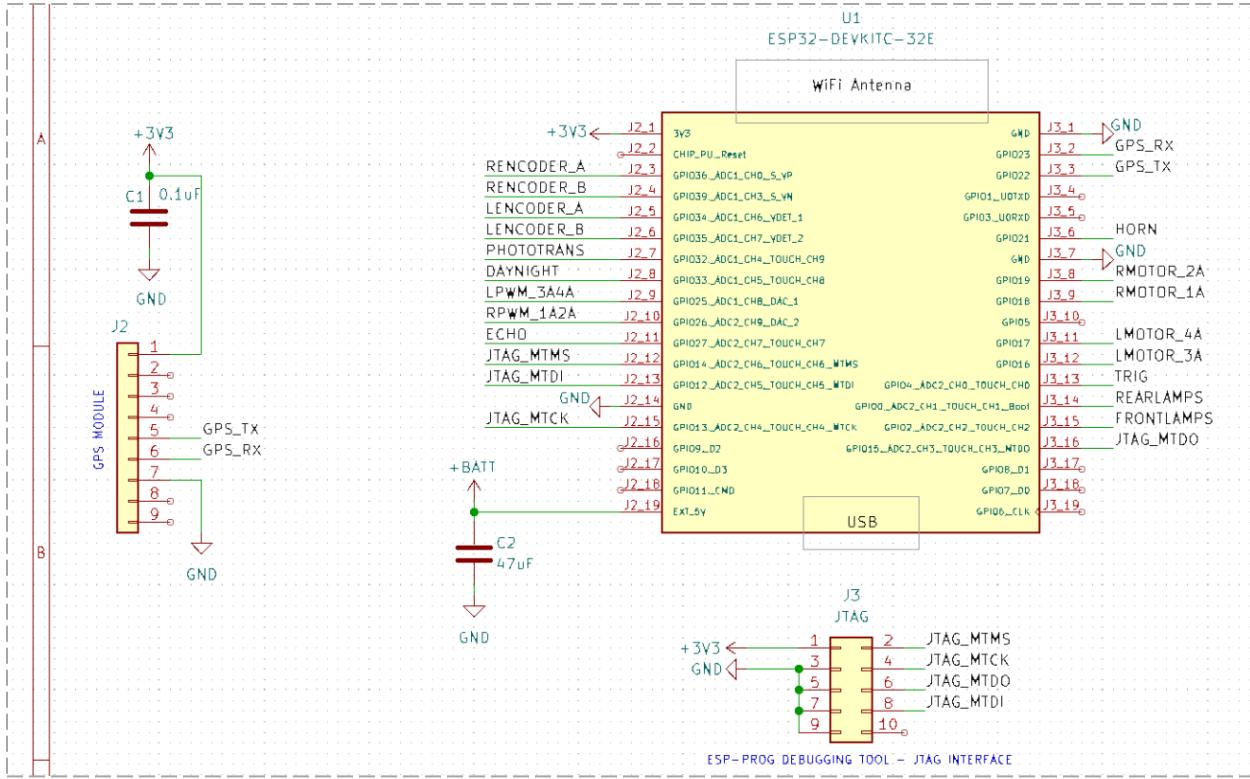
4. Complete the wiring of the drawing



STEP 22. THIS COMPLETES THE POWER INPUT SECTION.

STEP 23. ESP32 & GPS INTERFACE SECTION

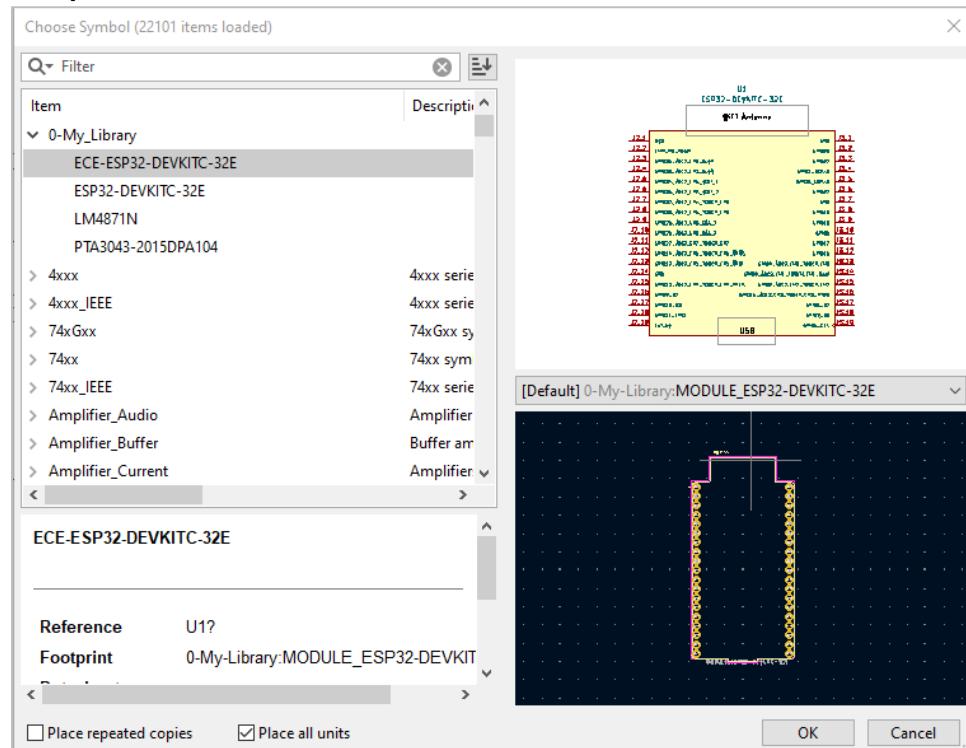
Place the following symbols.



1. Place the ESP32 Module:

Search Filter: [blank] | **Library:** 0_My_Library | **Part:** ECE-ESP32-DEVKITC-32E

Footprint: ESP32-DEVKIT-32E



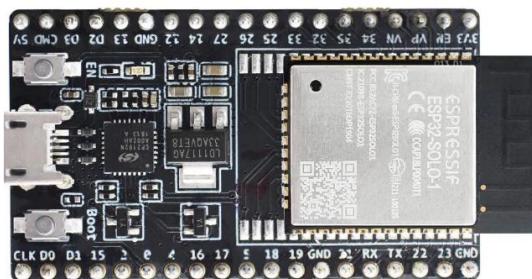
Design Notes: **ESP32-DevKitC Datasheet**

<https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/index.html>

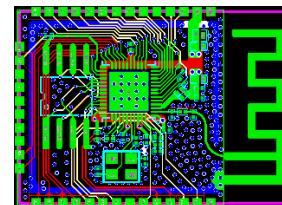
Schematic: https://dl.espressif.com/dl/schematics/esp32_devkitc_v4-sch.pdf

The ESP32-DevKitC is a development board that contains an ESP32-WROOM module:

ESP32 DevKitC



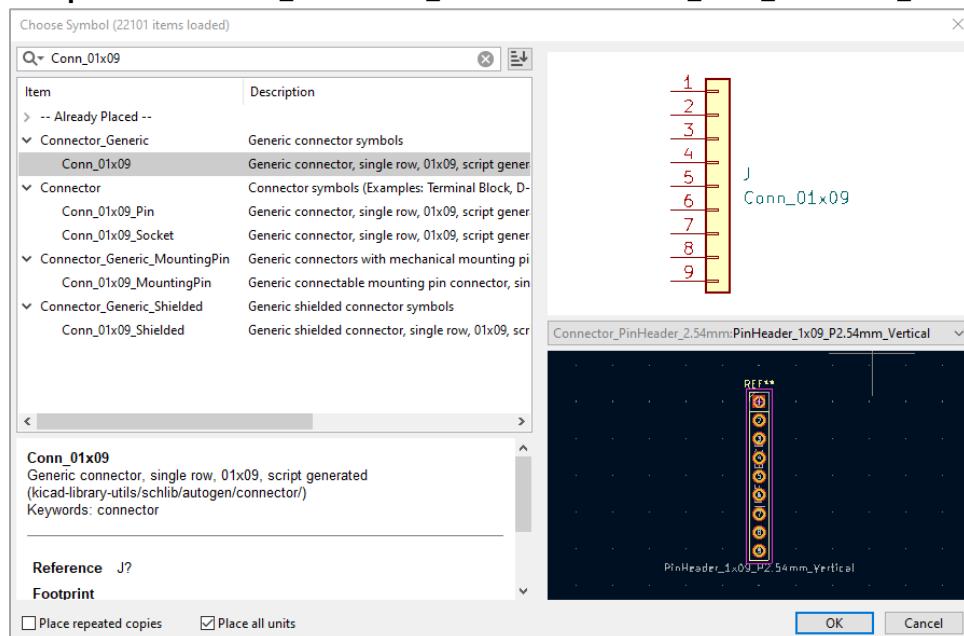
The ESP32-WROOM module is a small, high-density circuit board on its own



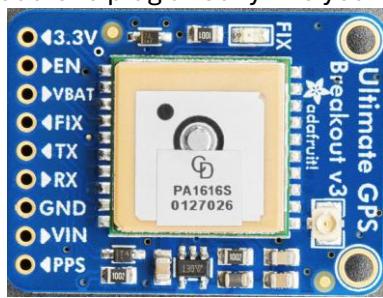
2. Place the GPS Module Connector:

Search Filter: Conn_01x09 | **Library:** Connector_Generic

Footprint: Connector_PinHeader_2.54mm:PinHeader_1x09_P2.54mm_Vertical



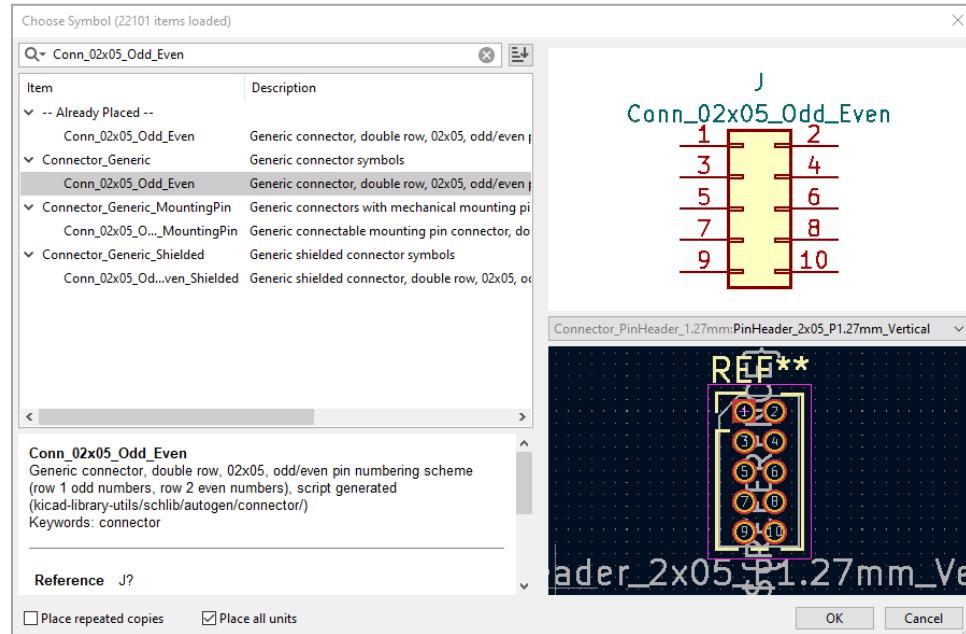
Design Notes: Adafruit Ultimate GPS Module connector: the 9-pin receptacle footprint that allows the GPS module to plug directly into your circuit board.



3. Place the JTAG Debugger Connector:

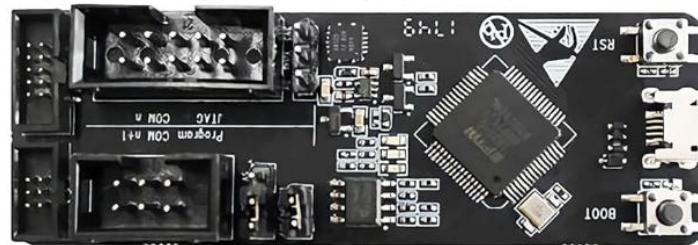
Search Filter: Conn_02x05_Odd_Even | **Library:** Connector_Generic

Footprint: Connector_PinHeader_1.27mm:PinHeader_2x05_P1.27mm_Vertical



Design Notes: **ESP32 ESP-PROG JTAG Interface:** a hardware debug interface that lets you step through code, set breakpoints, and inspect registers while the ESP32 is running.

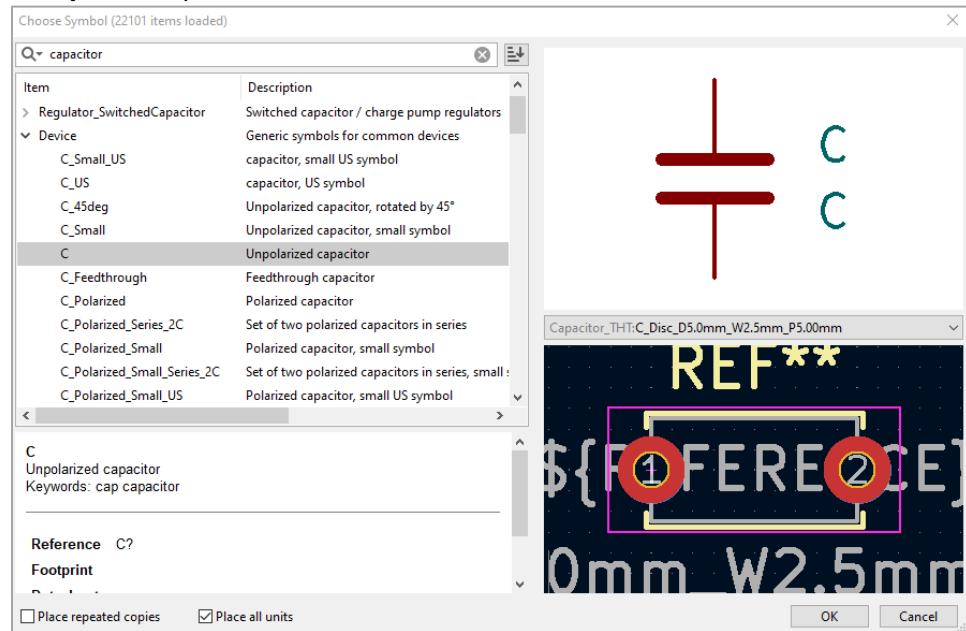
https://docs.espressif.com/projects/esp-iot-solution/en/latest/hw-reference/ESP-Prog_guide.html



4. Place the Decoupling Capacitors:

Search Filter: capacitor | Library: Device

Footprint: Capacitor_THT:C_Disc_D5.0mm_W2.5mm_P5.00mm



Design Notes: **Ceramic capacitor:** a small, non-polarized capacitor with very low internal resistance and

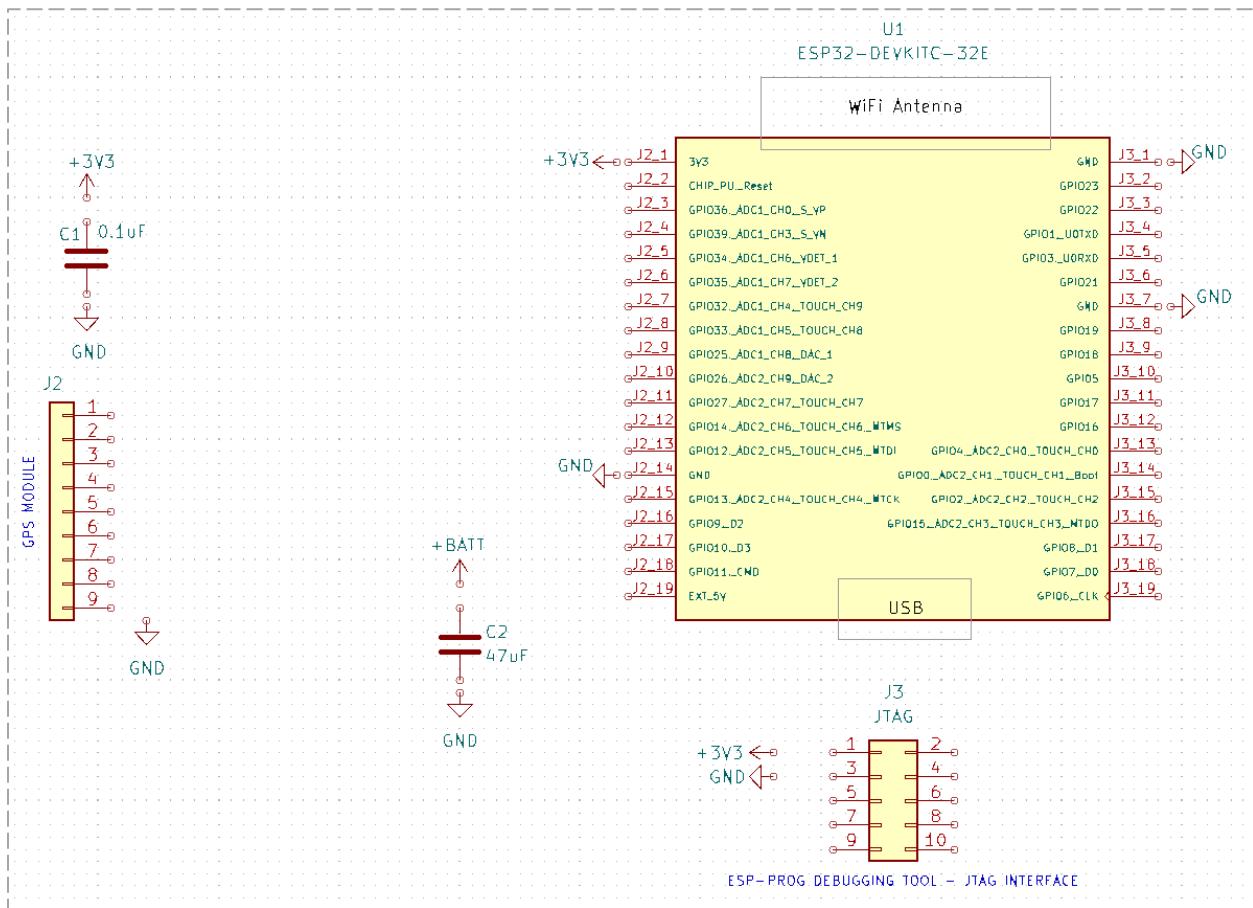
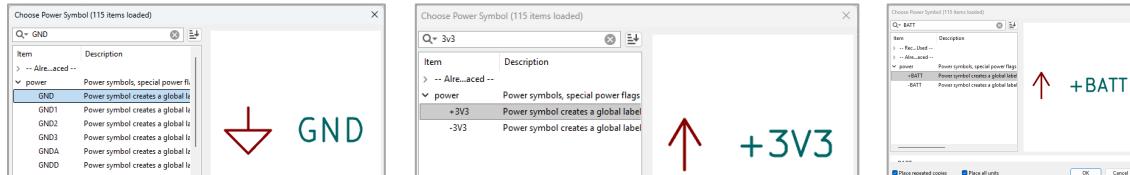
inductance, making it ideal for bypassing noise and fast electrical spikes.

5. Double-click on **C1** to change the default capacitance **Value** field to **0.1uF** for design reference.
6. Double-click on **C2** to change the default capacitance **Value** field to **47uF** for design reference.

STEP 24. PLACE POWER SYMBOLS

7. As shown in the previous section, add the power symbols using the **Place Power Symbols**  icon to insert power symbols.

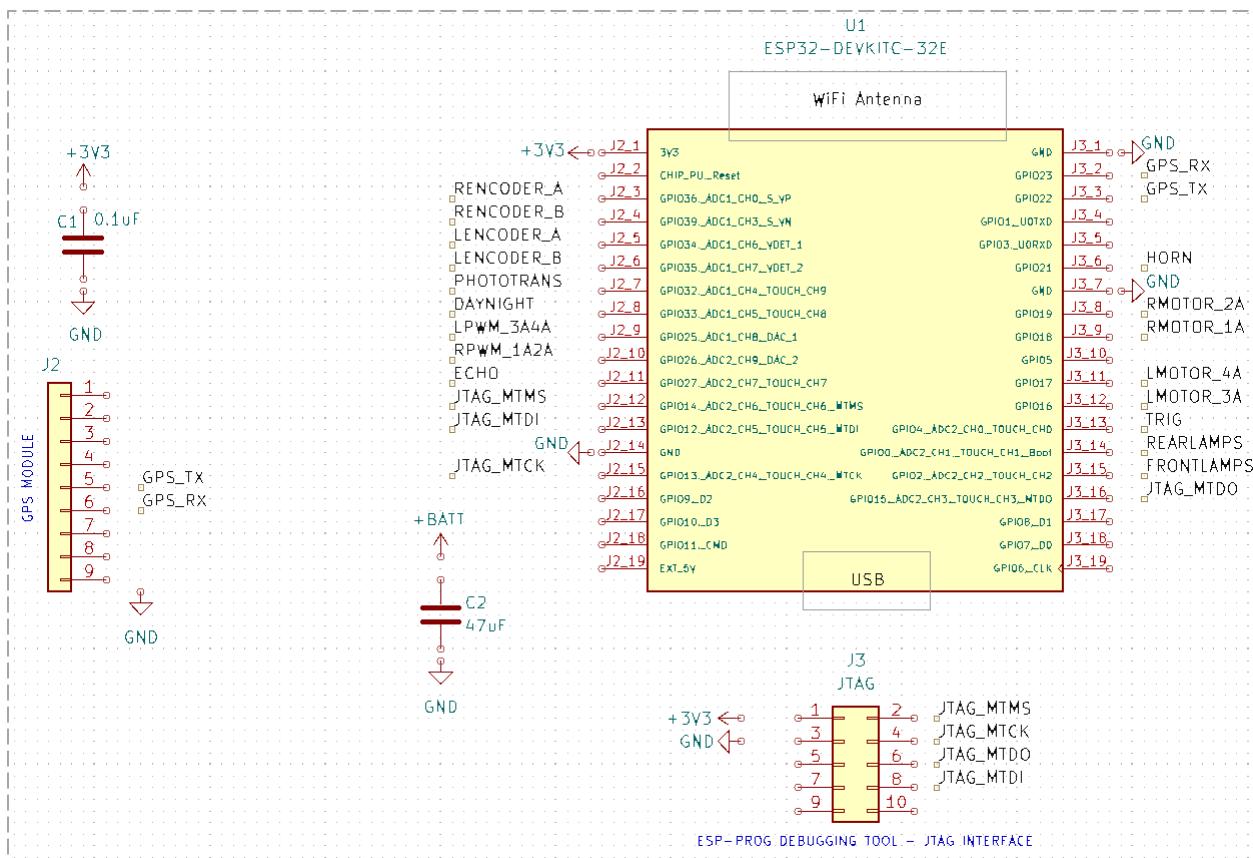
Ground/+3V3/+BATT Power Symbols



STEP 25. PLACE NET LABELS

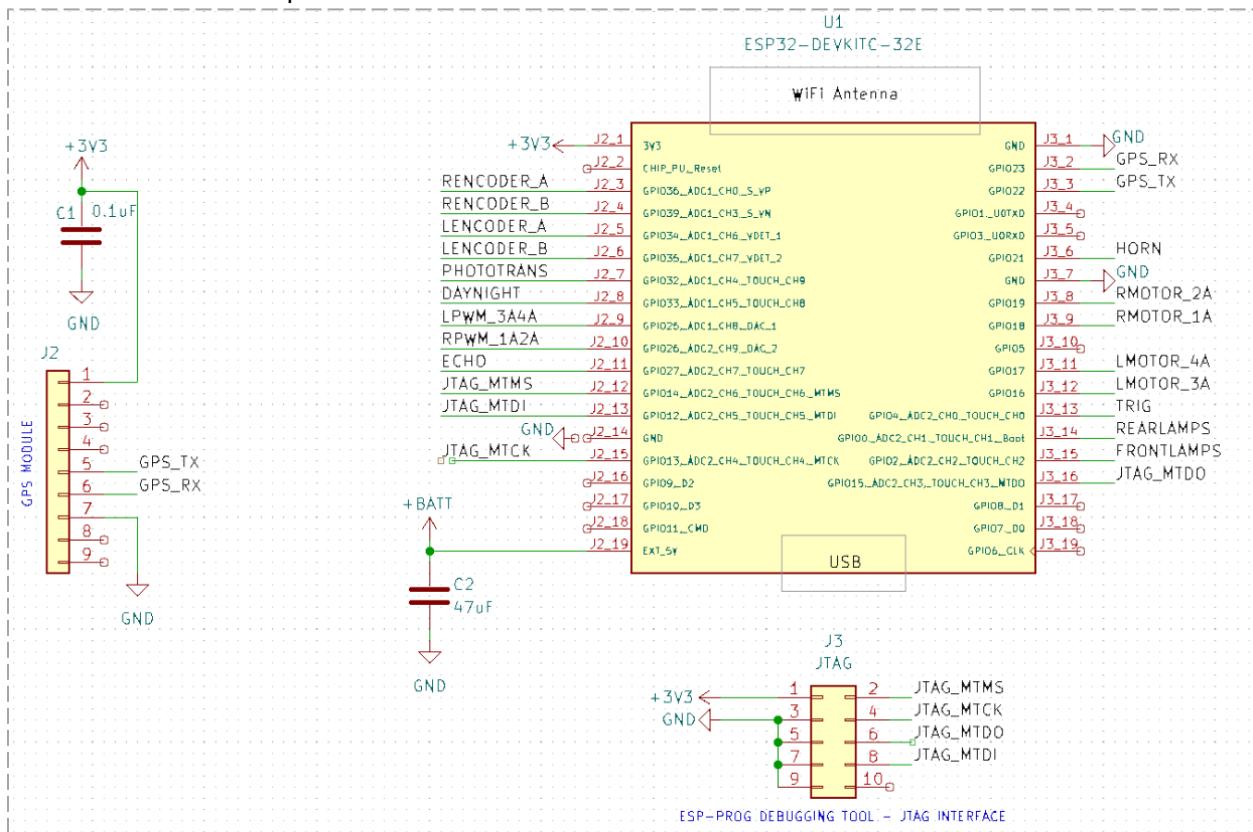
This circuit uses lots of **net labels** to connect component pins instead of drawing a spaghetti mess of crisscrossing wires. In KiCAD, every pin-to-pin electrical connection is given a hidden net name, unless defined with a net label. All these connections form the netlist, which is simply a database that says *which pins on which components are electrically connected*.

8. To add net labels, click the **Place Net Label**  button on the right-side toolbar. To assign the label to a wire, click along the path of the wire to assign.



STEP 26. DRAW WIRES

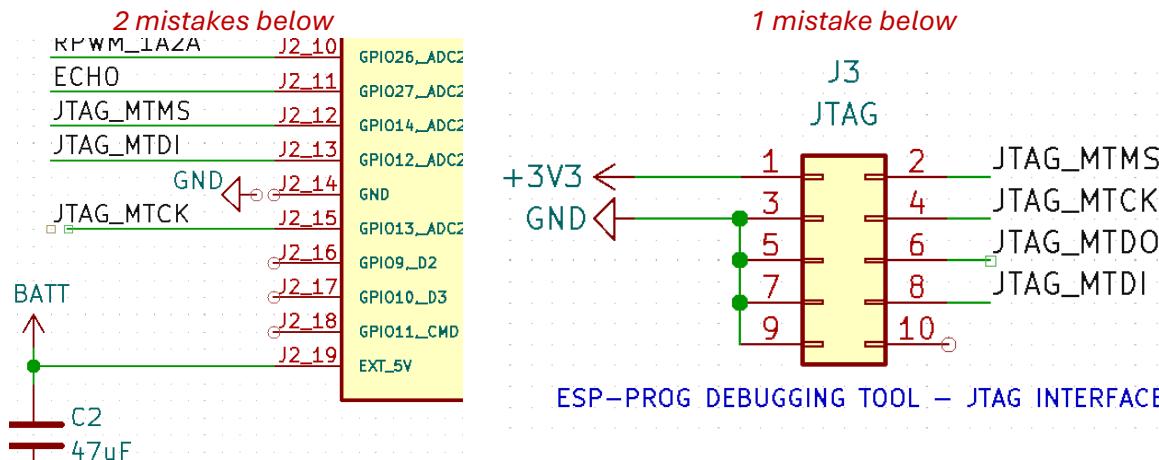
- Draw the wires to complete the circuit as shown below.



STEP 27. CHECK FOR MISTAKES

Did you notice the 3 wiring errors shown above? It's easy to miss—especially when working on a laptop screen or without zooming in. Small mistakes like this can lead to thousands of \$dollars in prototype delays if they aren't caught early.

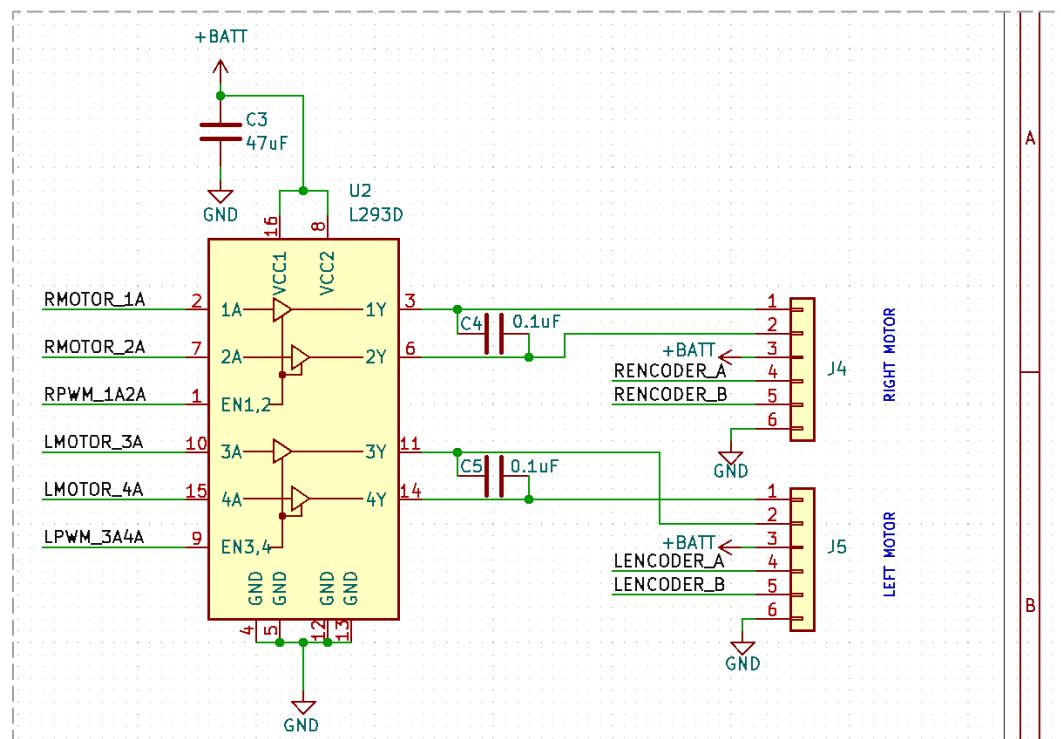
- Check Wires and Net Names



STEP 28. THIS COMPLETES THE ESP32 & GPS CIRCUIT SECTION.

STEP 29. MOTOR DRIVER CIRCUIT SECTION

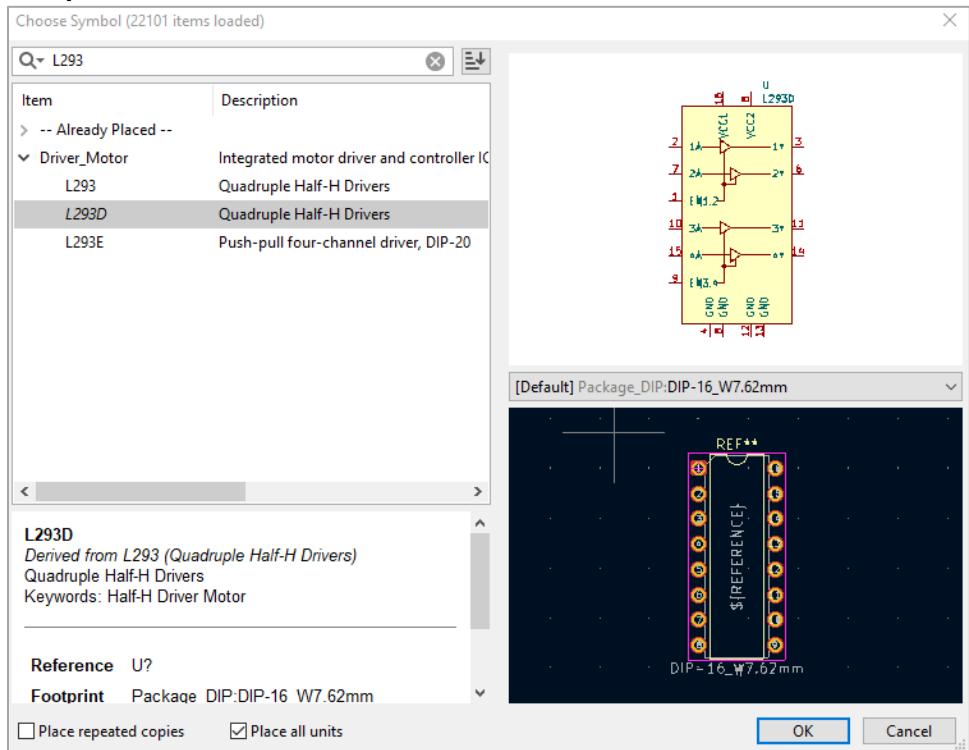
Place the following symbols.



1. Place a L293 Motor Driver Chip:

Search Filter: L293 | Library: Driver_Motor | Part: L293D

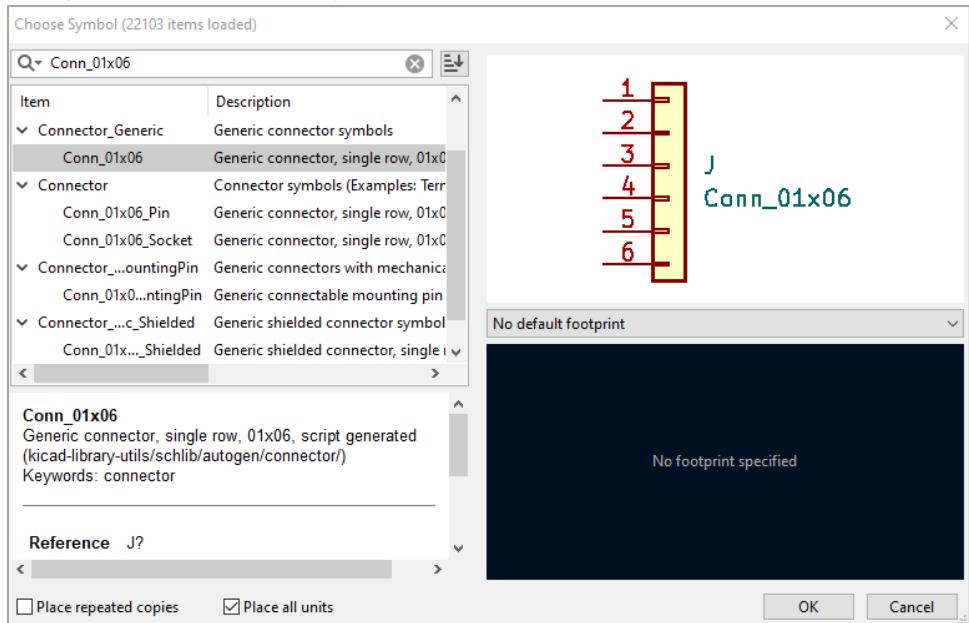
Footprint: Default DIP-16_W7.62mm



2. Place the Left and Right Motors:

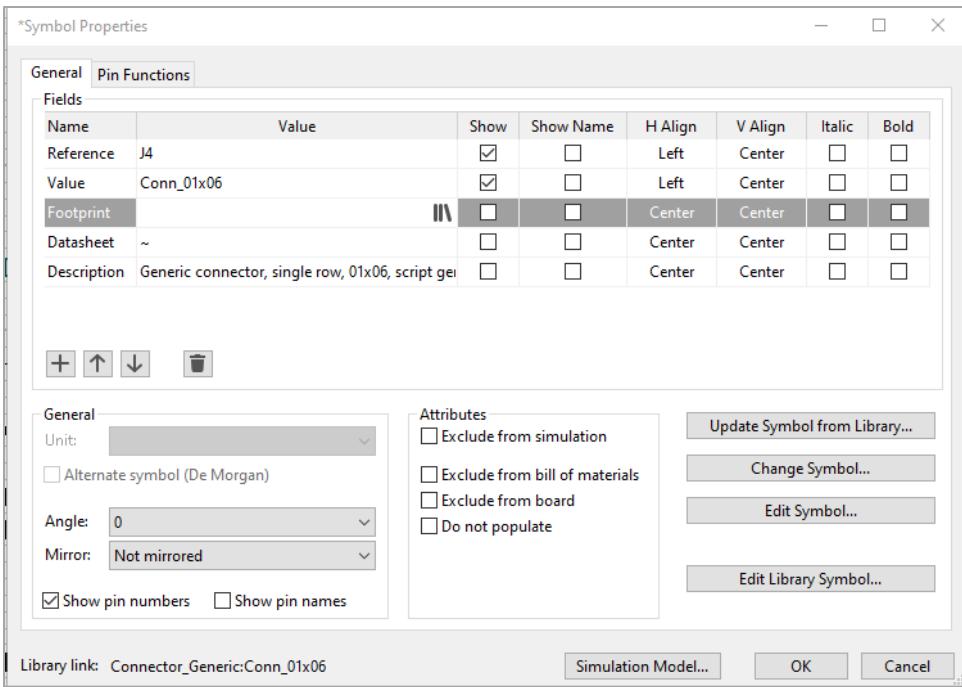
Search Filter: Conn_01x06 | **Library:** Connector_Generic | **Part:** Conn_01x06

Footprint: No default footprint

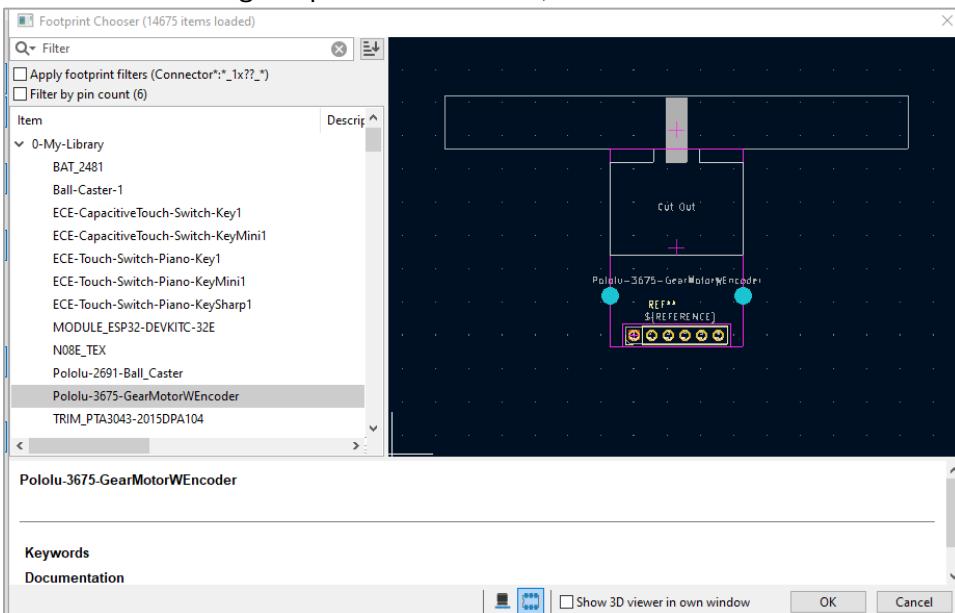


3. Click **OK**.

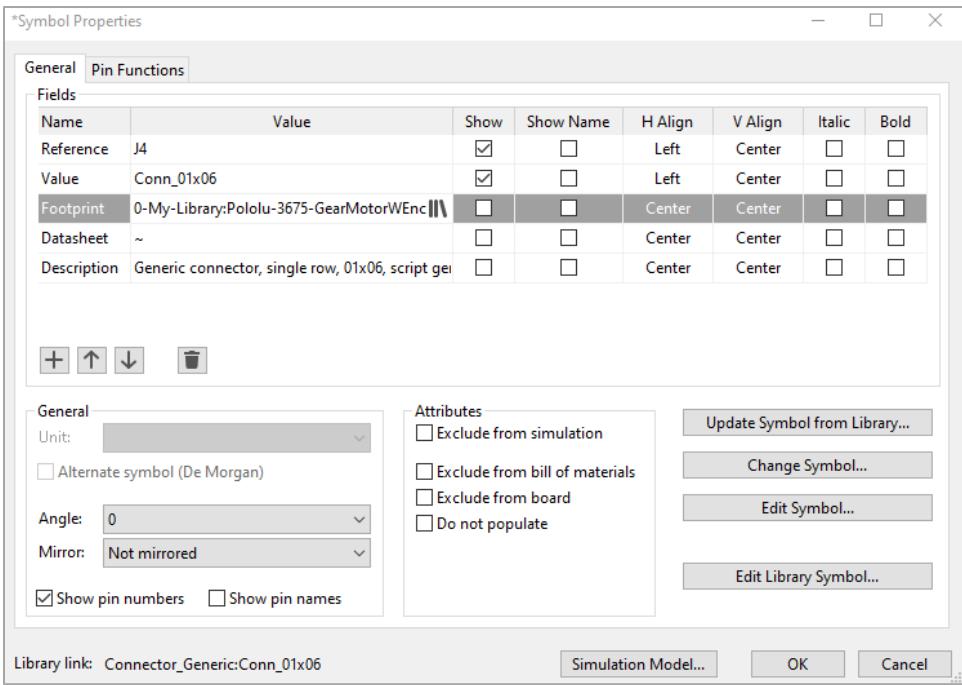
4. Click on your schematic sheet to place the symbol—refer to the example schematic for good placement.
5. Press **ESC** to exit part placement mode.
6. Let's add the footprint now. In your schematic, **double click** on the motor connector symbol to bring up the **Symbol Properties** window.
7. In the **Symbol Properties** window, the **Footprint** field is empty. Let's add the footprint we designed in house.
8. Click the **book icon**  next to the Footprint field to choose a footprint.



9. Navigate to the library **0-My-Library** and select **Pololu-3675-GearMotorWEncoder**.
10. Select the matching footprint shown below, then click **OK**.

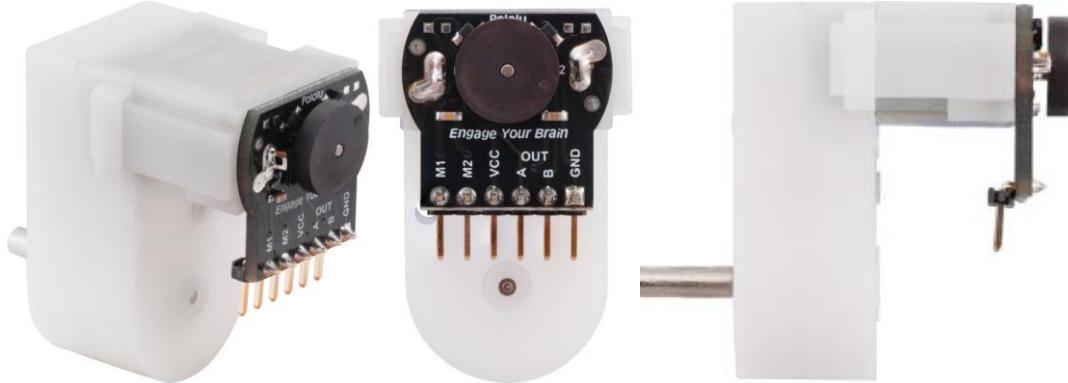


11. The **Footprint** field below now shows the correct footprint.



12. Click OK.

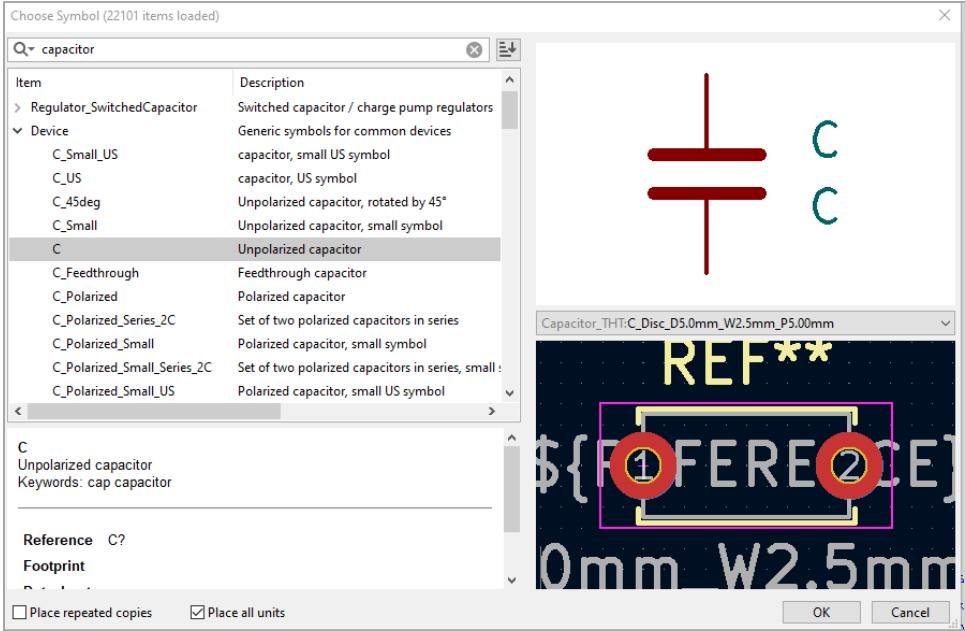
Design Notes: This motor footprint was designed by the ECE department, and both electrically connect and physically holds the motors to the PCB. <https://www.pololu.com/product/3675>



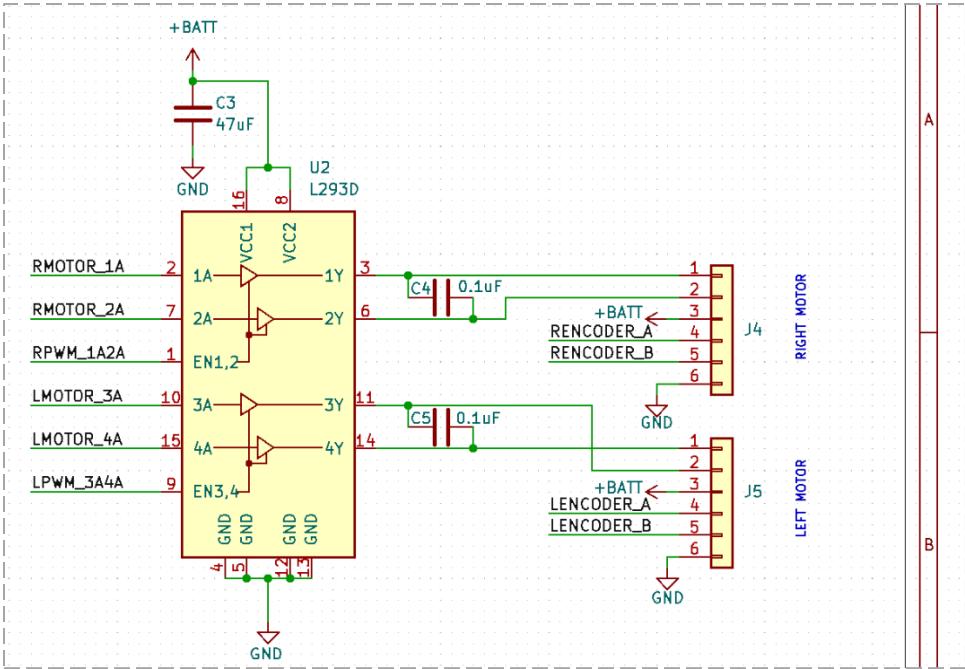
13. Place the L293 decoupling capacitor and the motor EMI suppression capacitors:

Search Filter: capacitor | **Library:** Device | **Part:** C

Footprint: Capacitor_THT:C_Disc_D5.0mm_W2.5mm_P5.00mm



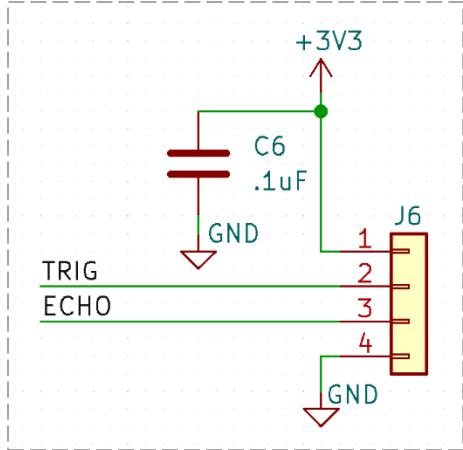
14. Double-click on C3 to change the default capacitance **Value** field to **47uF** for design reference.
15. Double-click on C4 & C5 to change the default capacitance **Value** field to **0.1uF** for design reference.
16. Place the **Net Labels**, **Power Symbols**, and draw the **Wires** to complete the circuit as shown below



STEP 30. THIS COMPLETES THE MOTOR DRIVER CIRCUIT SECTION.

STEP 31. ULTRASONIC SENSOR SECTION

Place the following symbols. Follow the example of placement from the full-size schematic.



1. Place a GPS Module Connector:

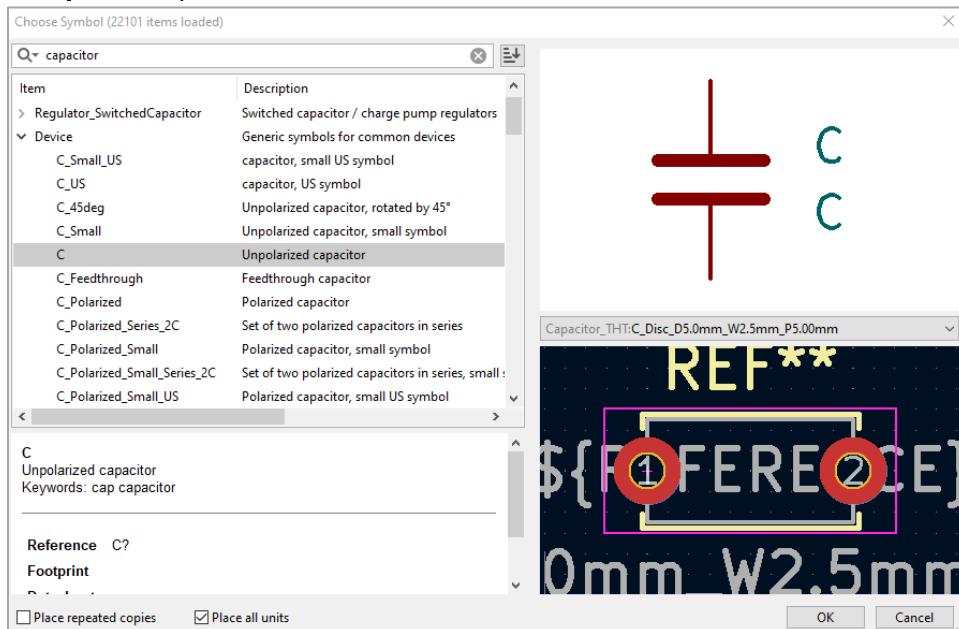
Search Filter: Conn_01x04 | **Library:** Connector_Generic

Footprint: Connector_PinHeader_2.54mm:PinHeader_1x04_P2.54mm_Vertical

2. Place a Decoupling Capacitor:

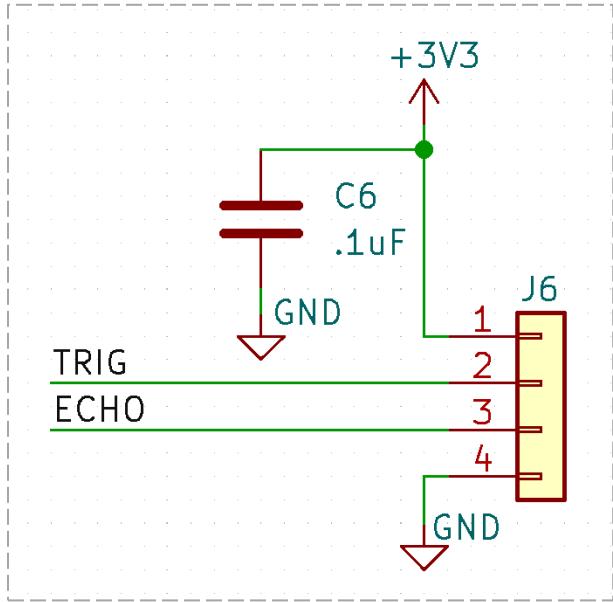
Search Filter: capacitor | **Library:** Device

Footprint: Capacitor_THT:C_Disc_D5.0mm_W2.5mm_P5.00mm



3. Double-click on C6 to change the default capacitance Value field to 0.1uF for design reference.

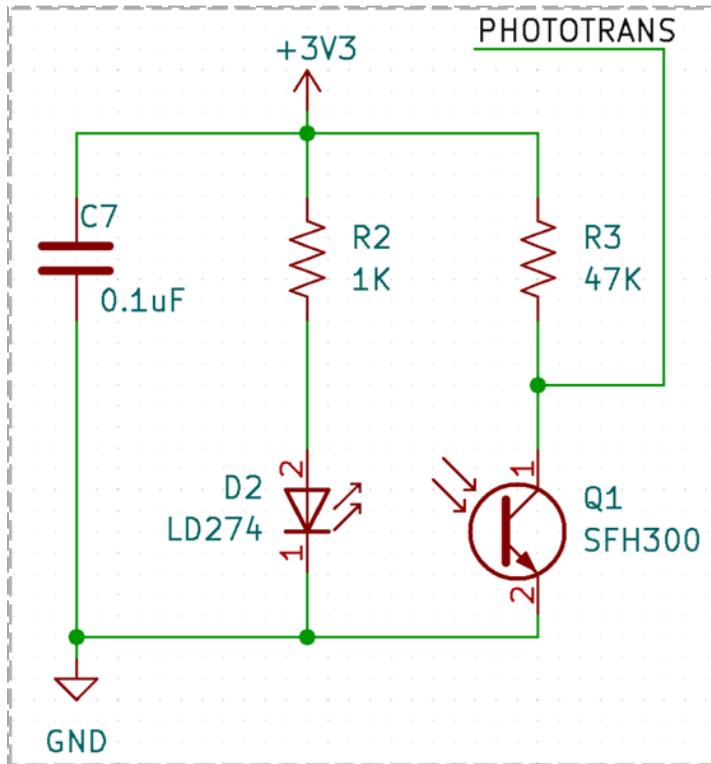
4. Place the **Net Labels**, **Power Symbols**, and draw the **Wires** to complete the circuit as shown below



STEP 32. THIS COMPLETES THE ULTRASONIC SENSOR SECTION.

STEP 33. LINE FOLLOWING CIRCUIT SENSOR SECTION

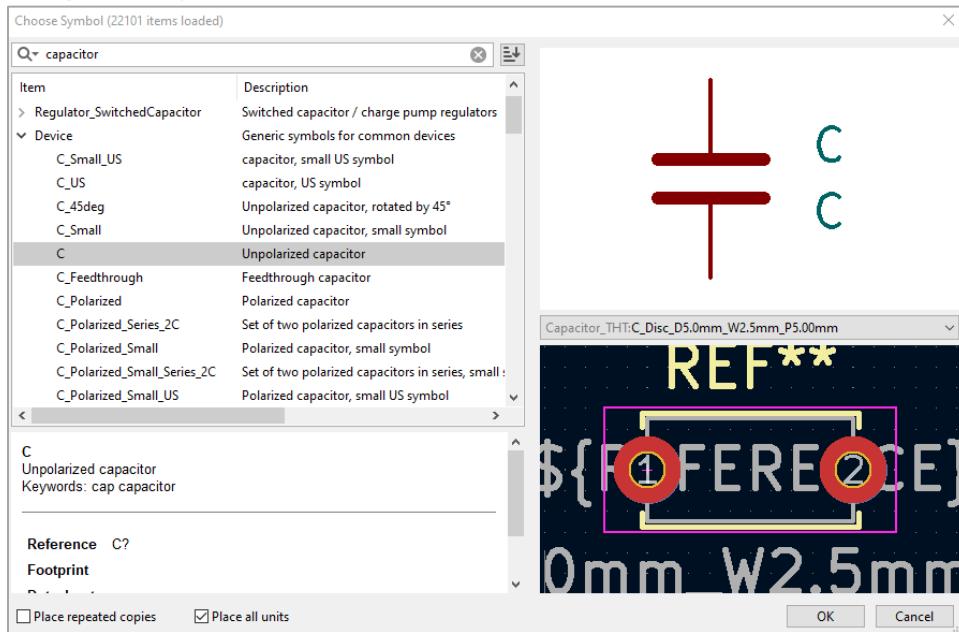
Place the following symbols. Follow the example of placement from the full-size schematic.



5. Place a Decoupling Capacitor:

Search Filter: capacitor | **Library:** Device | **Part:** C

Footprint: Capacitor_THT:C_Disc_D5.0mm_W2.5mm_P5.00mm

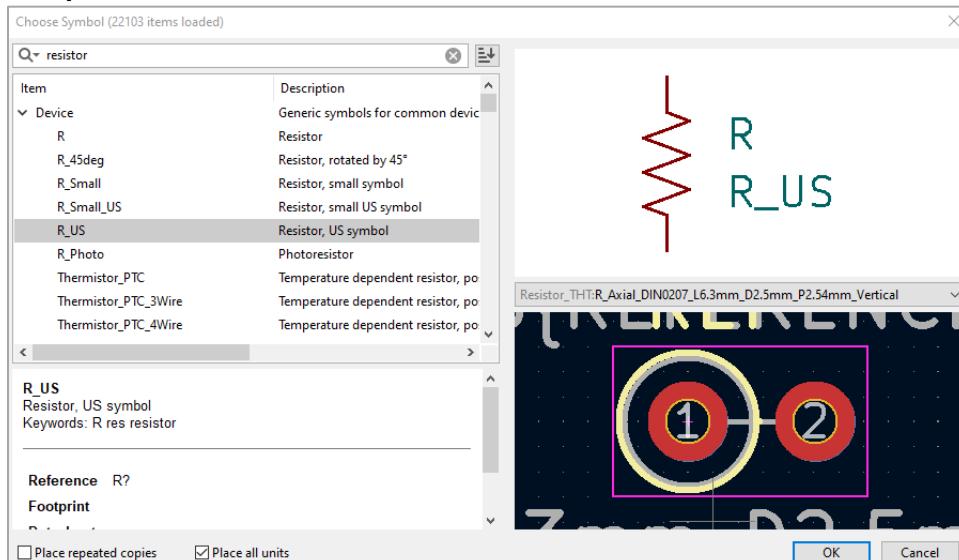


6. Double-click on C7 to change the default capacitance **Value** field to **0.1uF** for design reference.

7. Place Two Current Limiting Resistors:

Search Filter: resistor | **Library:** Device | **Part:** R_US

Footprint: Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P2.54mm_Vertical



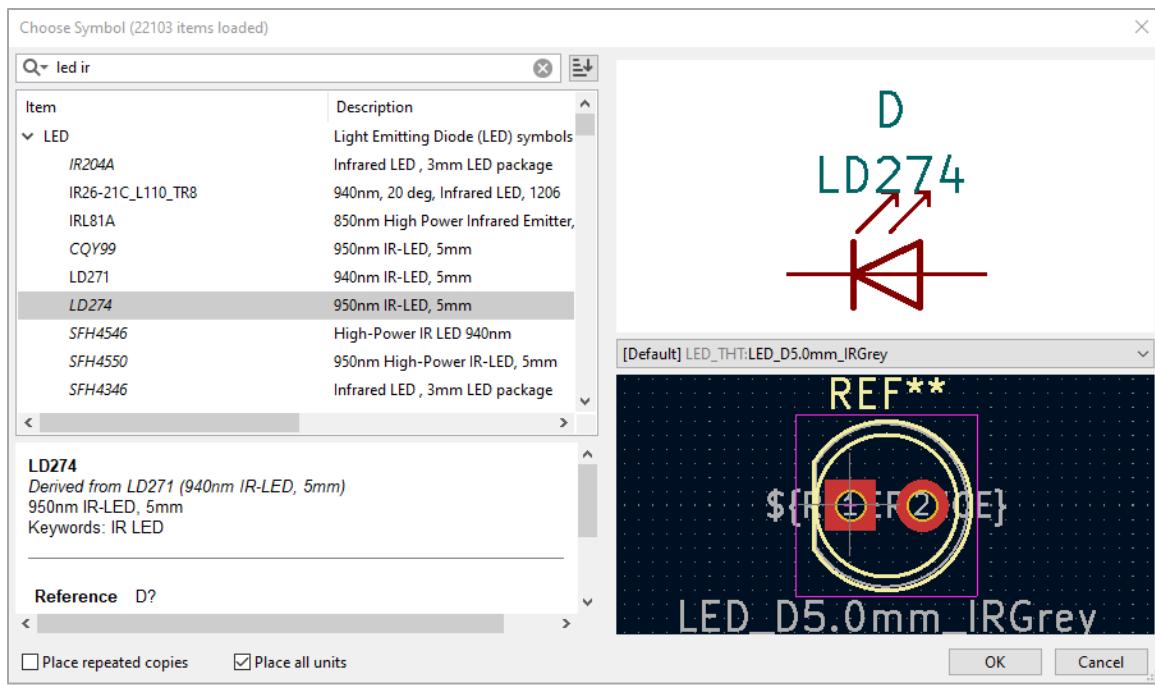
8. Double-click on R2 to change the default capacitance **Value** field to **1K** for design reference.

9. Double-click on R3 to change the default capacitance **Value** field to **47K** for design reference.

10. Place Infrared LED Emitter:

Search Filter: LED IR | **Library:** LED | **Part:** LD274

Footprint: LED_THT:LED_D5.0mm_IRGrey



Design Notes:

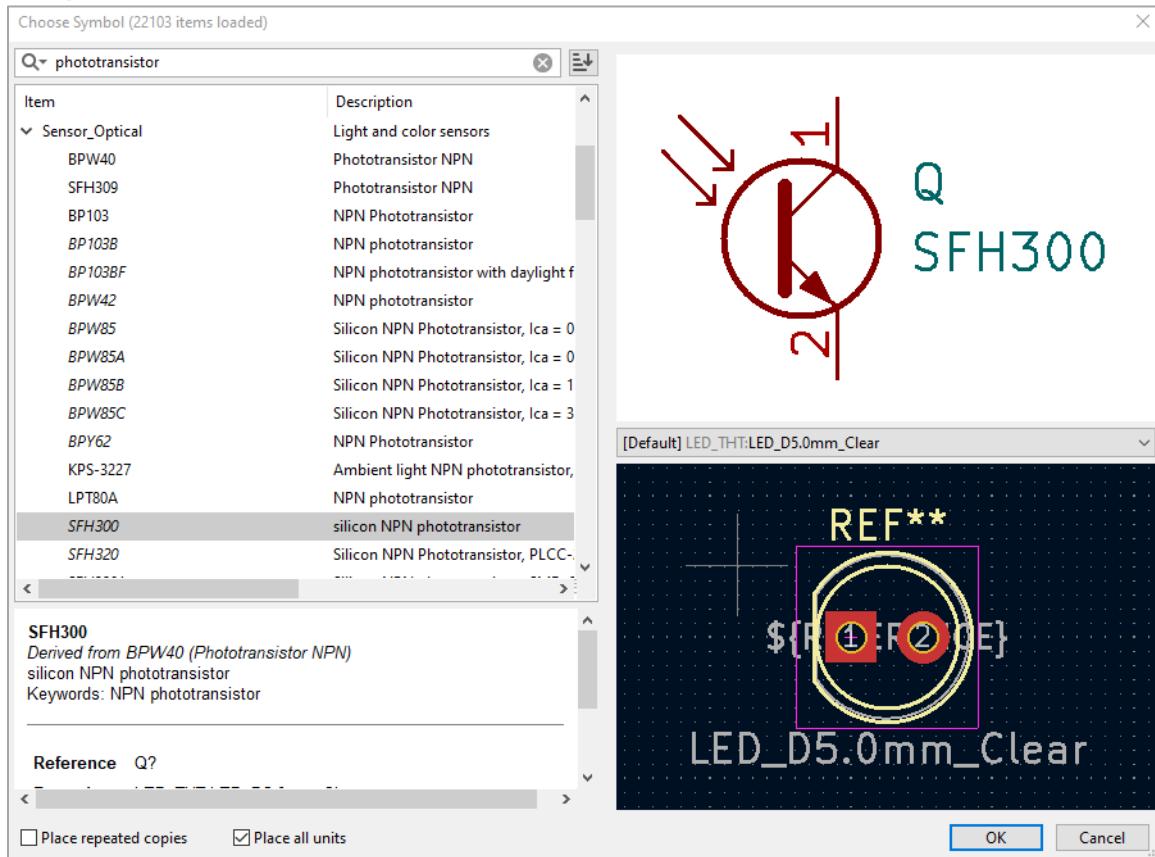
Description: Infrared (IR) Emitter LED, 940nm 50ma 5mm

Datasheet: Actual part used: <https://www.digikey.com/short/0jfm4j84>

11. Place Infrared Phototransistor:

Search Filter: phototransistor | **Library:** Sensor_Optical | **Part:** SFH300

Footprint: LED_THT:LED_D5.0mm_Clear

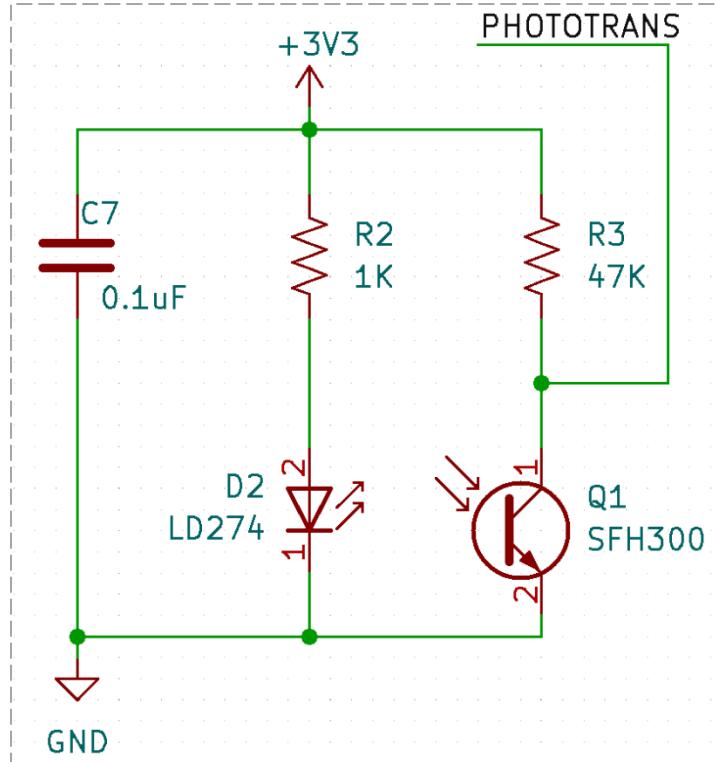


Design Notes:

Description: Infrared (IR) Emitter LED, 940nm 50mA 5mm

Datasheet: Actual part used: <https://www.digikey.com/short/mwdm0t4p>

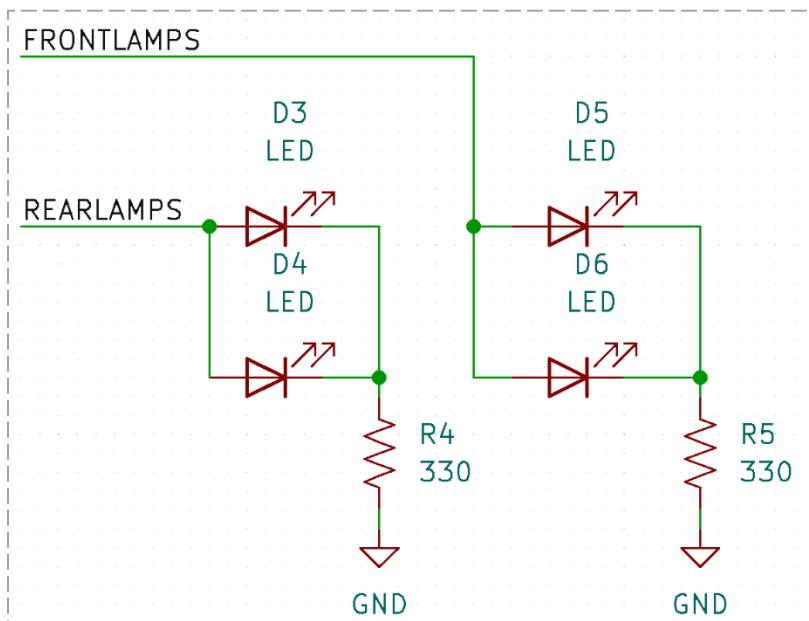
12. Place the **Net Labels**, **Power Symbols**, and draw the **Wires** to complete the circuit as shown below.



STEP 34. THIS COMPLETES THE LINE FOLLOWING CIRCUIT SECTION.

STEP 35. VEHICLE HEADLAMP/REAR LAMPS CIRCUIT SECTION

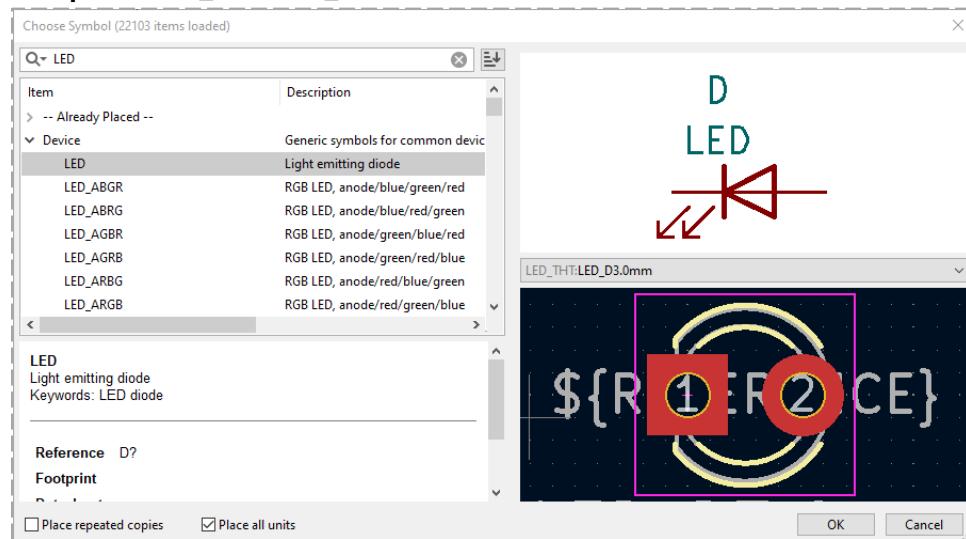
Place the following symbols. Follow the example of placement from the full-size schematic.



13. Place Vehicle Front/Rear Lights:

Search Filter: LED | Library: Device | Part: LED

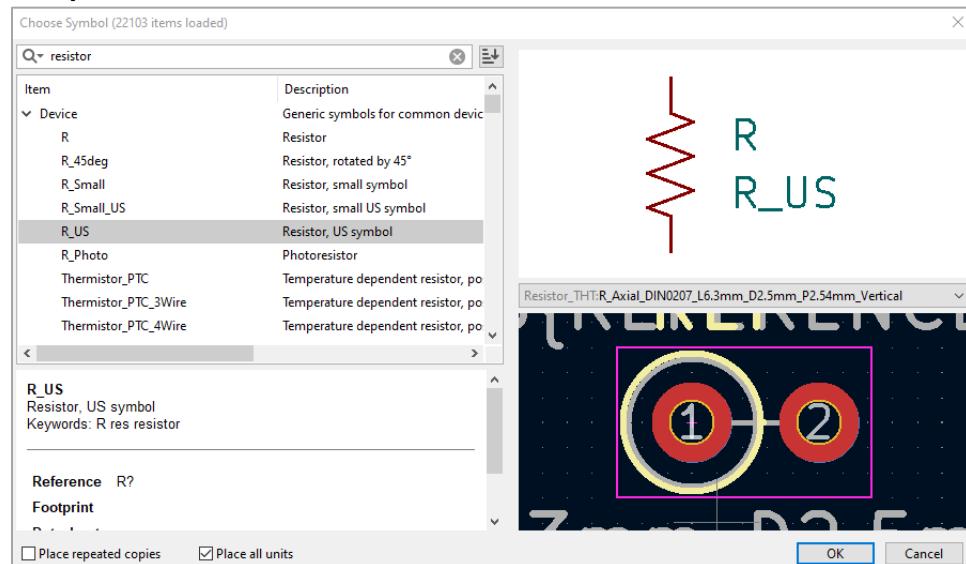
Footprint: LED_THT:LED_D3.0mm



14. Place Two Current Limiting Resistors:

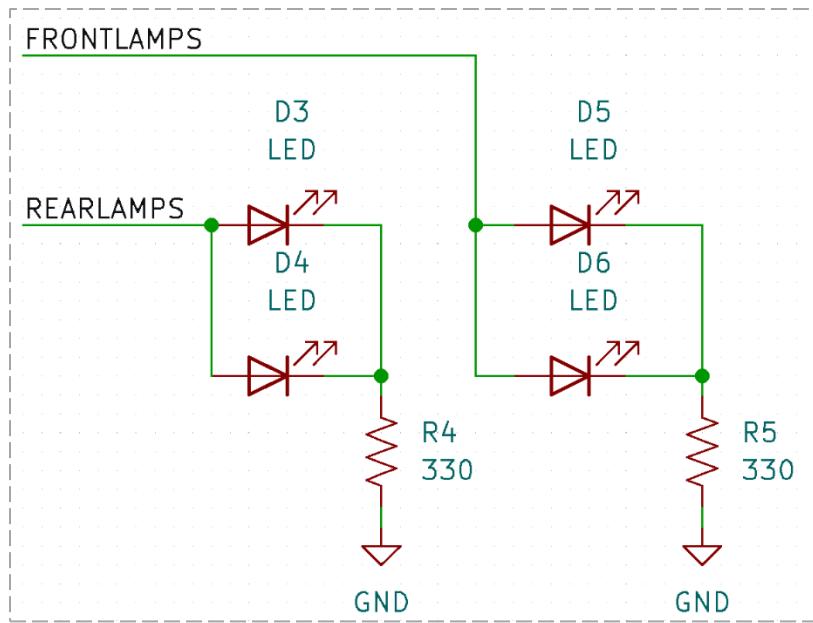
Search Filter: resistor | Library: Device | Part: R_US

Footprint: Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P2.54mm_Vertical



15. Double-click on R4 and R5 to change the default resistance Value field to **330** for design reference.

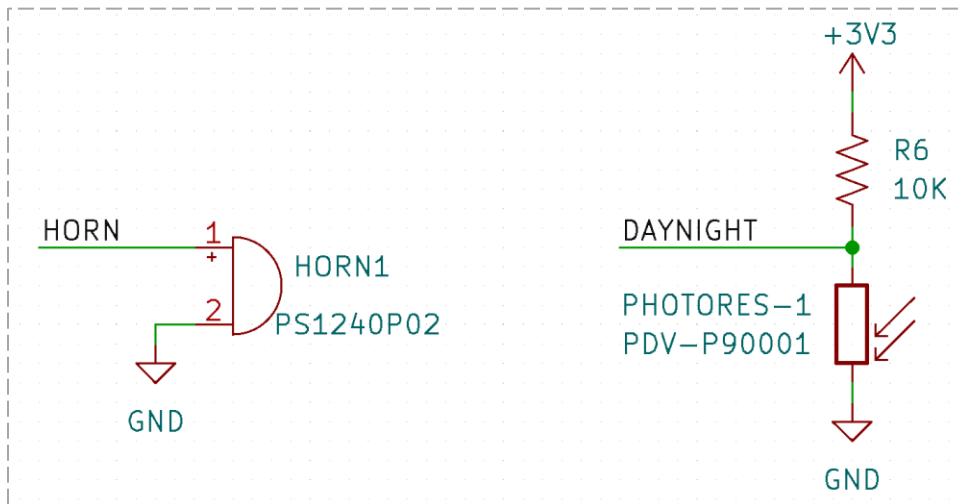
16. Place the **Net Labels**, **Power Symbols**, and draw the **Wires** to complete the circuit as shown below.



STEP 36. THIS COMPLETES THE FRONT/REAR LIGHTS CIRCUIT SECTION.

STEP 37. VEHICLE DAY/NIGHT SENSOR AND HORN CIRCUIT SECTION

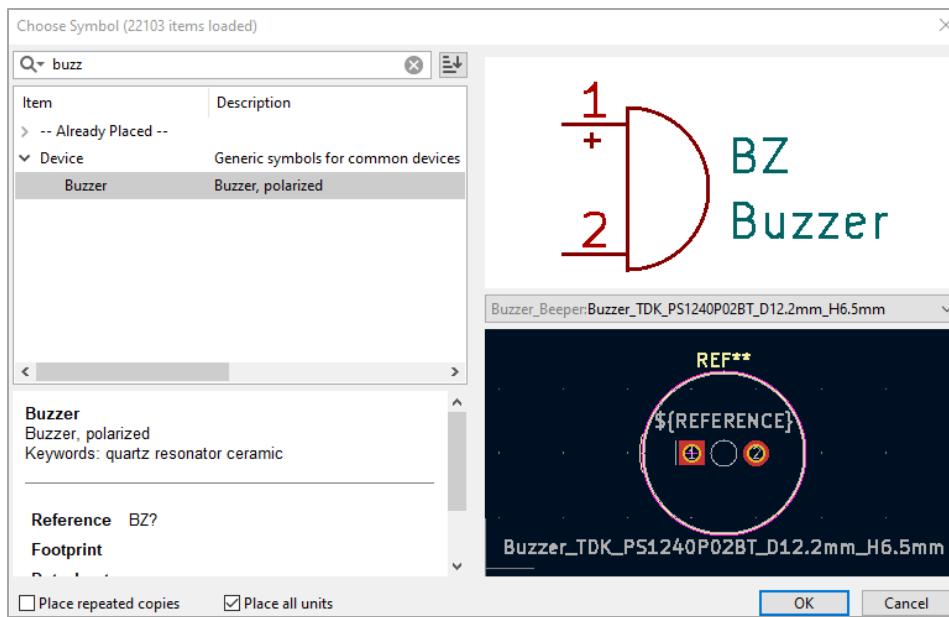
Place the following symbols. Follow the example of placement from the full-size schematic.



17. Place Vehicle Horn:

Search Filter: buzzer | **Library:** Device | **Part:** Buzzer

Footprint: Buzzer_Beeper:Buzzer_TDK_PS1240P02BT_D12.2mm_H6.5mm



Design Notes:

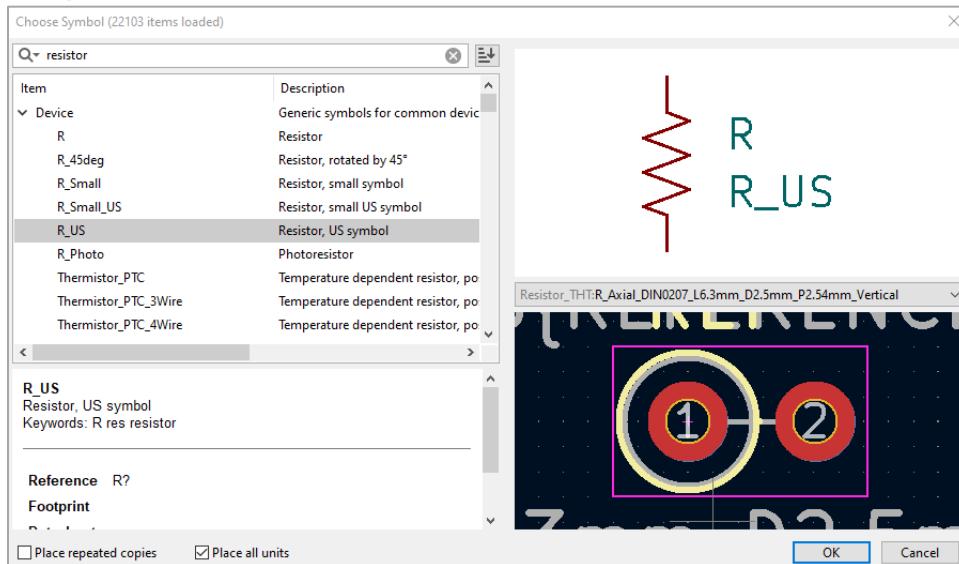
Description: Infrared (IR) Emitter LED, 940nm 50ma 5mm

Datasheet: Actual part used: <https://www.digikey.com/short/mwdm0t4p>

18. Place Current Limiting Resistor:

Search Filter: resistor | **Library:** Device | **Part:** R_US

Footprint: Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P2.54mm_Vertical

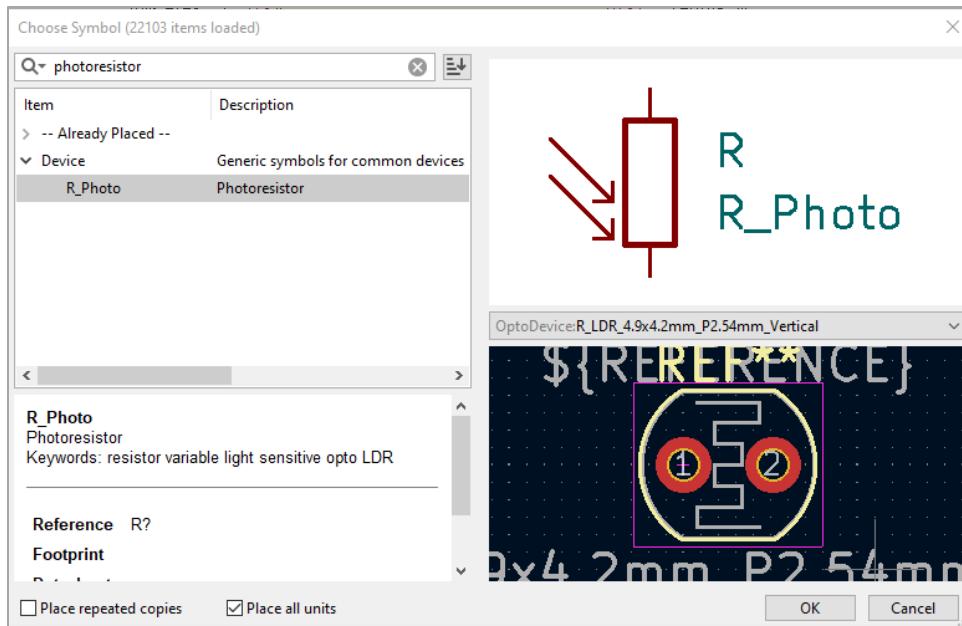


19. Double-click on R6 to change the default resistance Value field to 10K for design reference.

20. Place Photoresistor Day/Night Sensor:

Search Filter: Photoresistor | **Library:** Device | **Part:** R_Photo

Footprint: OptoDevice:R_LDR_4.9x4.2mm_P2.54mm_Vertical



Design Notes:

Description: Cds Cell Photoresistor 4K-11K OHM @ 10 lux 4.20MM 570nm

Datasheet: <https://www.digikey.com/short/m0q22q8z>

21. Place the **Net Labels**, **Power Symbols**, and draw the **Wires** to complete the circuit as shown below.



STEP 38. THIS COMPLETES THE VEHICLE DAY/NIGHT SENSOR & HORN CIRCUIT SECTION.

STEP 39. DESIGN REQUEST NOTICE

- In **Canvas** → **Files** → folder **Design Change Request** check for any Design Change Notices. Be sure to complete these before proceeding.

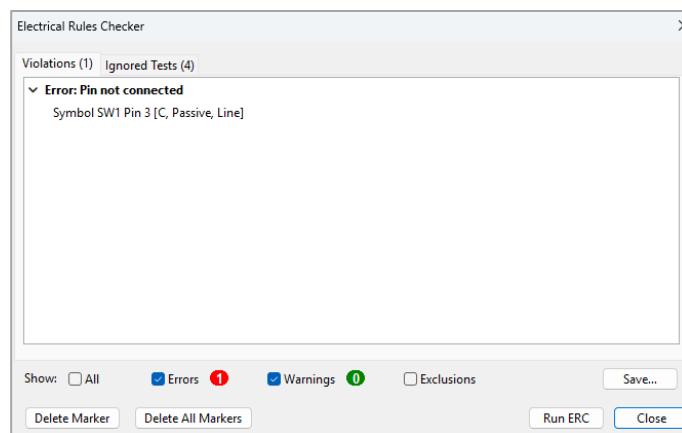
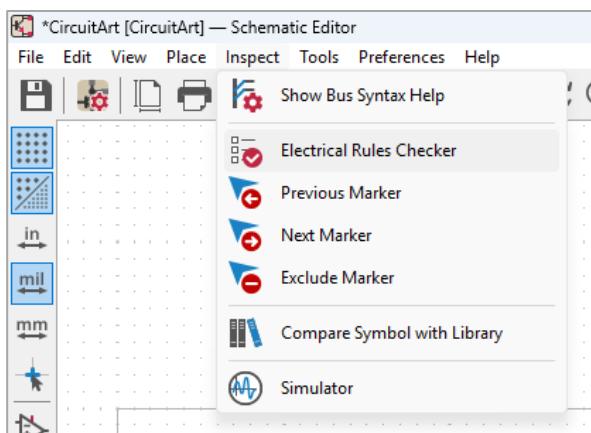
STEP 40. CHECK FOR ELECTRICAL ERRORS

Before moving on, verify your schematic connections:

- In the **Schematic Editor**, Click **Inspect** → **Electrical Rules Checker (ERC)**.

Electrical Rules Checker

✓ Error ✓ Warnings



Errors to Ignore:

- Symbol '+3V3' doesn't match copy in library 'power'
 - Symbol 'GND' doesn't match copy in library 'power'
 - This is a free open-source program, many of the errors can be ignored and there may also be the means to fix them. We can discuss any errors of concern in the workshop or lab.
2. Use the mouse scroll wheel to review and address any errors or warnings. **Unconnected pins or nets and multiple parts with the same reference designator are the most critical issues.** The **ERC doesn't find your design mistakes**, but it helps you double-check for common errors.
 3. Save your schematic once you consider it error-free.

STEP 41. SCHEMATIC PEER REVIEW & QUALITY CHECK

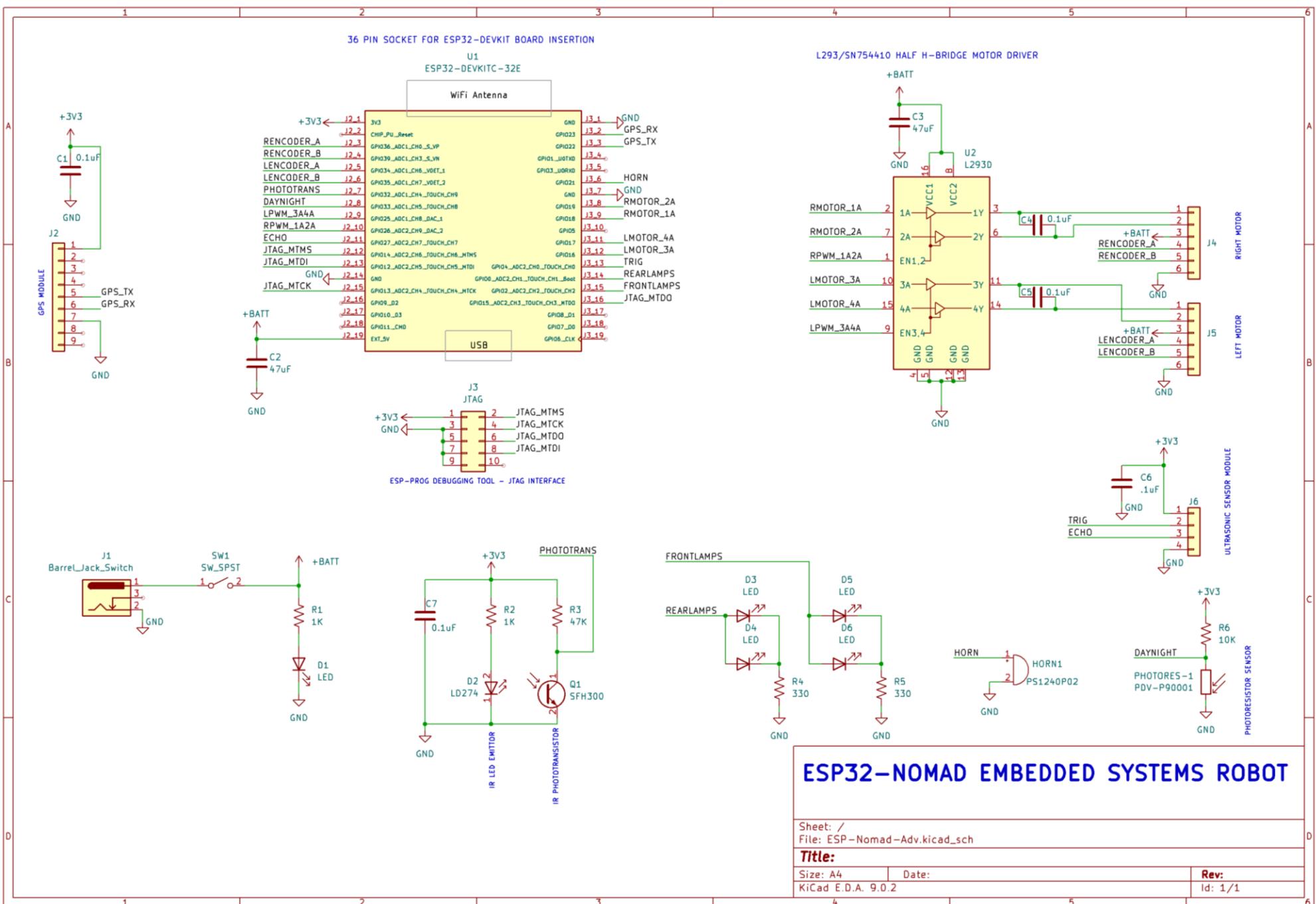
2. In Canvas → Files, complete the schematic peer review before you proceed to far along with your PCB layout.
3. Two students from Highline College came in to solder and assemble PCBs they designed here a month ago. Their boards did not work—not because of soldering or code, but because a simple schematic mistake was never caught.
4. This is an important lesson: DRC and ERC only check rule violations, not design intent. They cannot tell you if your circuit makes sense. That's why human review and careful schematic checking matter.
5. Download/open the Schematic Peer Review form. The form seems to work in Edge and Chrome.
6. Try to find one classmate who has not reviewed anyone's schematic yet or someone that has and willing to review yours. (everyone should try to take a turn as a reviewer).
7. Have them review your schematic on your screen or give them your schematic to check as a print or .sch file.
8. The reviewer completes the checklist, writes any notes, and signs the form.
9. You fix anything marked "Needs Fix," then submit your completed .sch file on Canvas.

💡 Tip: If you're having trouble finding a reviewer, ask someone new first

💡 Tip: KiCAD can crash occasionally. As you work on your schematic and PCB layout, make regular backups. Use Save As and add a clear prefix such as "Backup-" to the filename (for example: Backup-before-wiring-ESP32-Robot

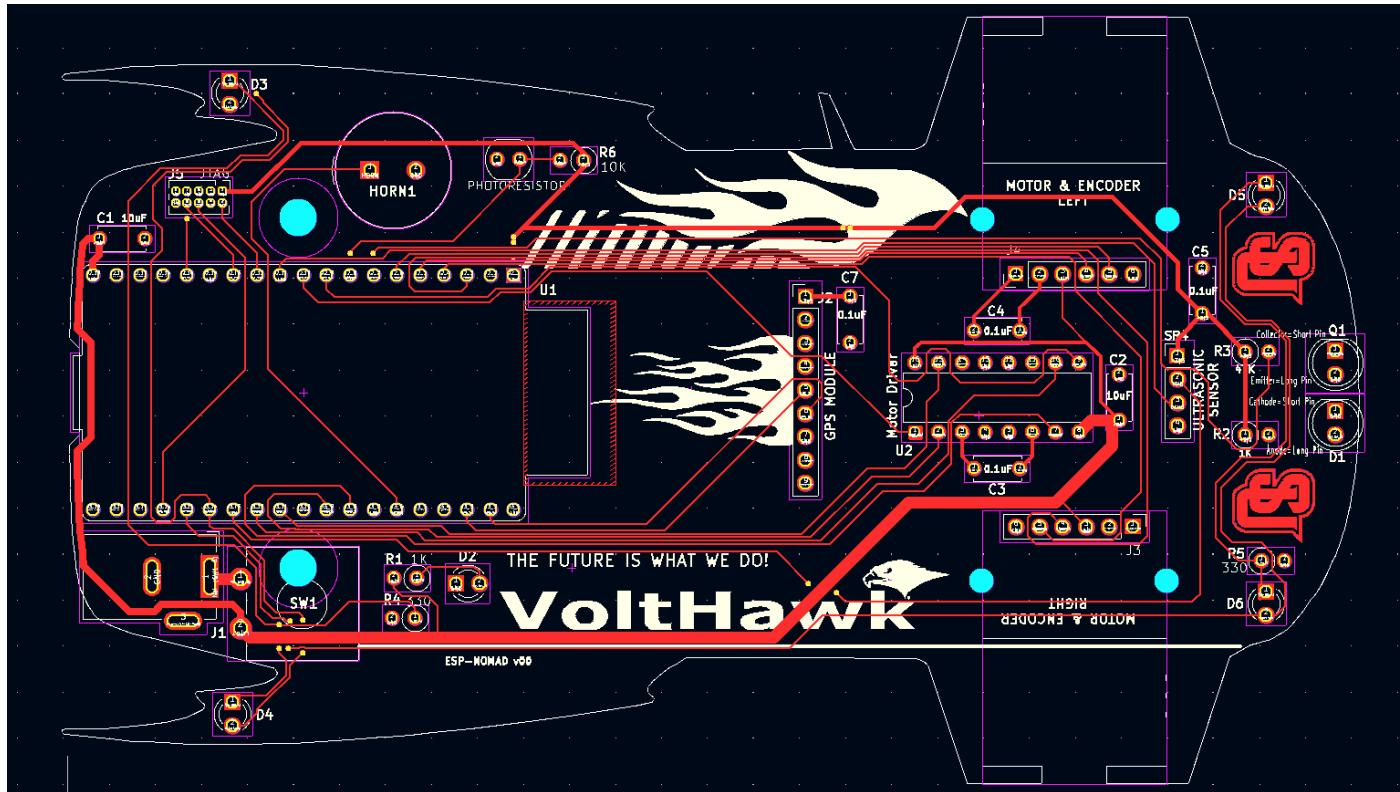
For your reference, I've included a large size image of the schematic on the next page.

Full Schematic



Redhawk ESP32 EMBEDDED SYSTEMS ROBOT—PCB LAYOUT

[PCB Editor | 9.0](#) | [English](#) | [Documentation](#) | [KiCad](#)



Example Embedded Robot Board Layout

While your boards are being manufactured, we'll dig into how each circuit section actually behaves in the real world. This includes resistance, inductance, and capacitance (RLC)—not just as abstract components, but as *properties* of the PCB itself.

PCB LAYERS EXPLAINED - A BEGINNER-FRIENDLY GUIDE

A printed circuit board (PCB) is built like a multi-layer sandwich. Each layer has its own job. Once you understand them, your KiCAD designs will look much more professional!

Typical PCB Layers

1. Fiberglass Core Center
 - Rigid, non-conductive material that gives the board its strength.
2. Copper Layers
 - Very thin copper foil bonded to the top and bottom of the fiberglass.
 - The copper forms the conductive tracks that carry all electrical signals and power, and provide the pads where components are soldered to the board.
 - KiCAD names:
 - F.Cu → Front (top) copper
 - B.Cu → Back (bottom) copper
3. Solder Mask Layers
 - Thin colored protective coating (lacquer) is applied over the copper.
 - This is the colored surface you see on a finished board.
 - Purposes:
 - Prevents accidental solder bridges (shorts)
 - Protects copper from oxidation
 - Makes soldering easier and cleaner
 - Common colors:

Solder mask	Silkscreen
Green	White
Blue	White
Red	White
Yellow	White
Black	White
White	Black

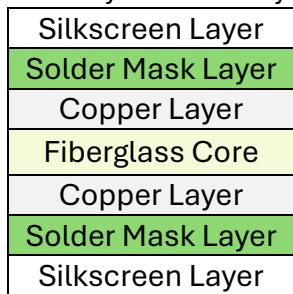
4. Silkscreen Layers

- Ink printed on top of the solder mask.
- Used for component labels (R1, C3, etc.), logos, text, outlines, polarity marks and simple artwork.
- Usually white ink.
 - On white solder mask → manufacturers use black ink for better contrast.

KiCAD names:

- F.Silkscreen (top side)
- B.Silkscreen (bottom side, when needed)

5. That's it — only four main layers to remember!

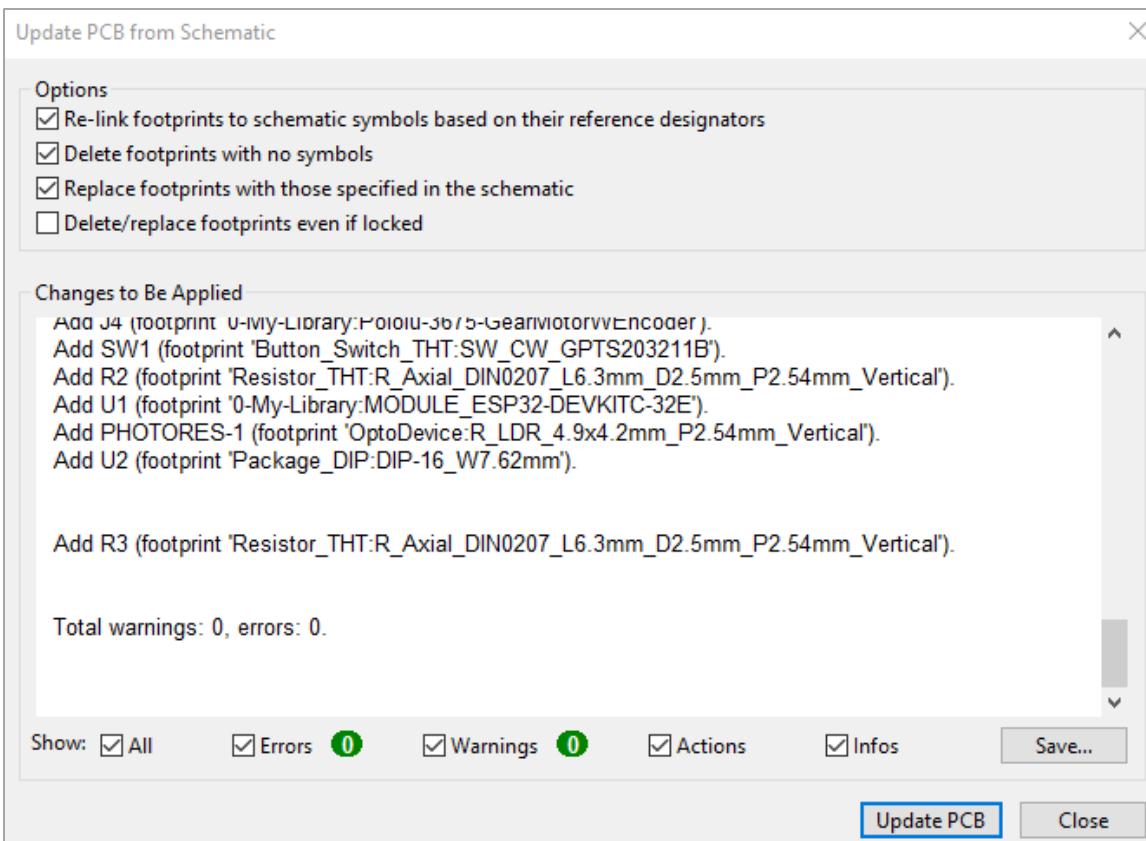


STEP 42. GENERATING THE NETLIST AND PREPARING FOR PCB LAYOUT

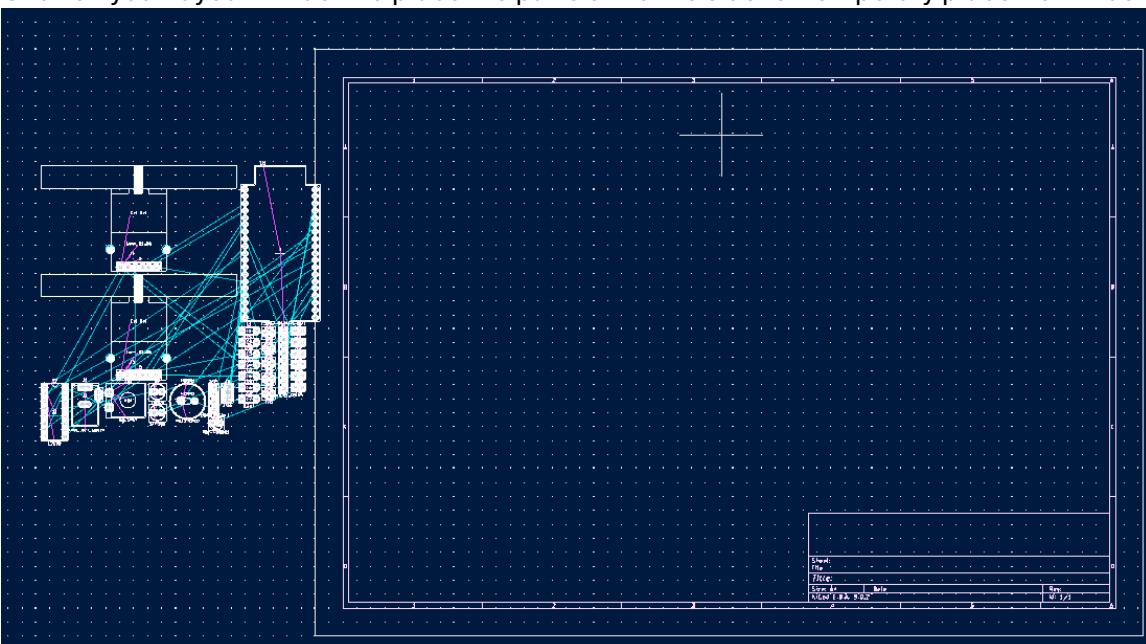
Once your schematic is complete and error-free, it's time to move from the logical circuit (schematic) to the physical board (layout).

The next step links each symbol's pins in the schematic to the matching footprints on the circuit board. The **netlist** is the information that transfers those connections from the schematic to the board layout, describing exactly which component pins are electrically connected.

1. Let's transfer your schematic data into the PCB Editor:
2. Under the **Tools** menu, click **Update PCB from Schematic**.



3. Be sure you have no warnings or errors. Resolve any errors or warnings (unconnected pins nets, etc.) before proceeding.
Total warnings: 0, errors: 0.
4. If there are no errors or warnings, in the dialog box, click **Update PCB**.
5. After the Update PCB, check that it continues to show no errors or warnings shown. Resolve any errors or warnings (missing footprints, unconnected pins nets, etc.) before proceeding.
6. All footprints you placed earlier will now appear stacked together in the PCB Editor workspace.
7. Click on your layout window to place the parts off to the side for temporary placement—as shown below.

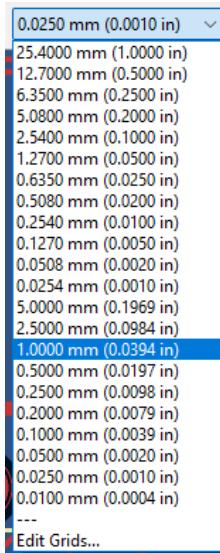


STEP 43. PCB LAYOUT EDITOR

The PCB file should already be open, if not from the KiCAD Project window, double-click the **.kicad_pcb** file or select **PCB Editor icon**.

STEP 44. ABOUT THE KICAD GRID (WHY AND WHEN TO CHANGE IT)

KiCad uses a **grid** to help you place and route things accurately. You don't need to stay on one grid size the whole time—changing it is normal and expected.



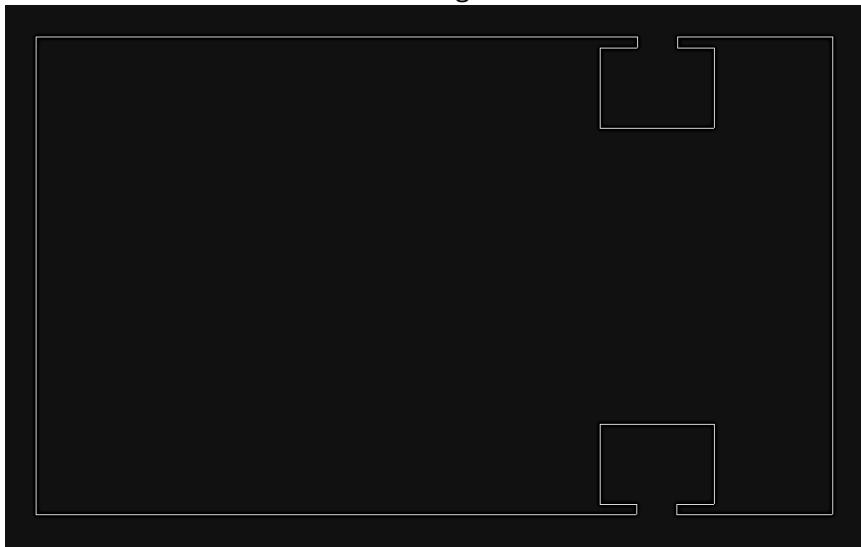
- Start with a larger grid.
- **Use a larger grid** (for example, 1.0 mm or 0.5 mm) when **placing components**. This helps parts line up cleanly and keep layouts neat and organized and for easier routing of traces later.
- **Use a smaller grid** (for example, 0.25 mm or 0.1 mm) when **routing traces**. A finer grid gives you more control when navigating tight spaces between pins and pads. You can change the grid at any time using the **Grid dropdown** in the toolbar. Think of the grid as a helper—not a rule. It's there to make your work cleaner and easier, not to limit you.

STEP 45. CREATING A CIRCUIT BOARD OUTLINE

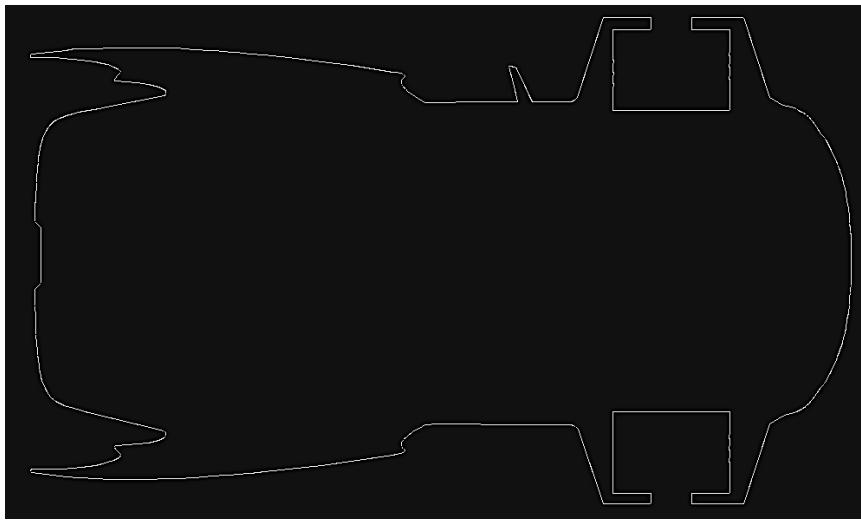
The board shape and graphics you choose can significantly affect how a design is perceived, turning a classroom project into something that looks like a professional product.

Due to time constraints, a basic board outline footprint is provided for you to use if you wish, though you are free to modify this basic outline or create your own.

Robot Board Outline—Basic Rectangular



Robot Board Outline—Graphically Shaped to Rule the Road!



STEP 46. CREATING A BOARD OUTLINE IN KICAD

Every PCB needs a board outline — a closed shape that tells the manufacturer where the board should be cut. In KiCAD, board outlines are always drawn on the **Edge.Cuts** layer—see pic below. Without this outline, your PCB cannot be fabricated.

KiCad PCB Layers (Common & Important)

Layers	Objects	Nets	Layer Name	Type	Purpose
█	█		F.Cu	Copper	Top copper layer for traces and pads
█	█		B.Cu	Copper	Bottom copper layer for traces and pads
█	█		F.Adhesive		Adhesive glue locations for temporarily holding surface-mount parts on bottom side during soldering.
█	█		B.Adhesive		
█	█		F.Paste	Top Solder Paste	Top stencil openings for SMT
█	█		B.Paste	Bottom Solder Paste	Bottom stencil openings for SMT
█	█		F.Silkscreen	Top Silkscreen	Top text, labels, reference designators
█	█		B.Silkscreen	Bottom Silkscreen	Bottom text and markings
█	█		F.Mask	Top Solder Mask	Openings in top solder mask
█	█		B.Mask	Bottom Solder Mask	Openings in bottom solder mask
█	█		User.Drawings		
█	█		User.Comments		
█	█		User.Eco1		
█	█		User.Eco2		
▶	█		Edge.Cuts	Mechanical	Board outline (defines board shape)
█	█		Margin		
█	█		F.Courtyard	Courtyard	Component clearance (top)
█	█		B.Courtyard	Courtyard	Component clearance (bottom)
█	█		F.Fab	Fabrication	Assembly drawings (top)
█	█		B.Fab	Fabrication	Assembly drawings (bottom)
█	█		User.1		
█	█		User.2		
█	█		User.3		
█	█		User.4		

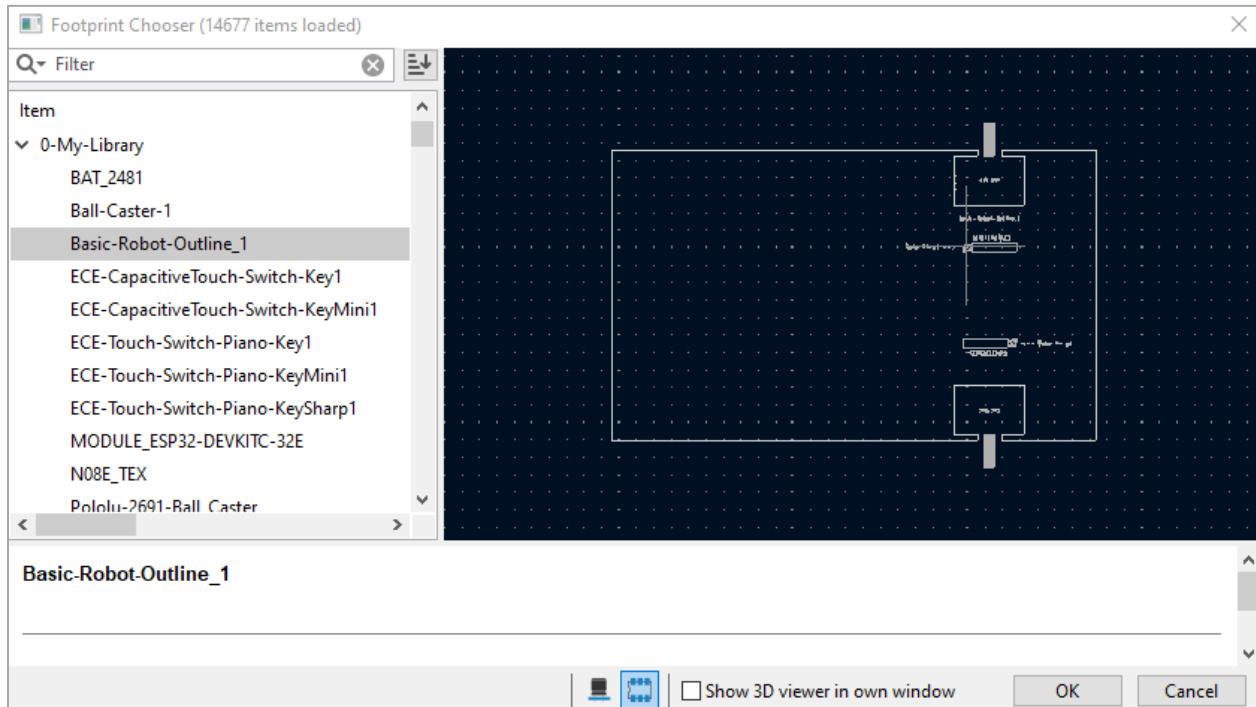
STEP 47. DEFAULT BOARD OUTLINE

Often, the board outline is created after placing components, but here we are setting the size and providing a generic shape of the robotics board ahead of time. If preferred, you can customize the shape later.

A basic board outline has been provided as a footprint in the 0-My-Library library to help you get started quickly. Move all components inside this outline. You are welcome to modify the outline to your liking. The footprint also includes helpful motor pin-1 alignment indicators.

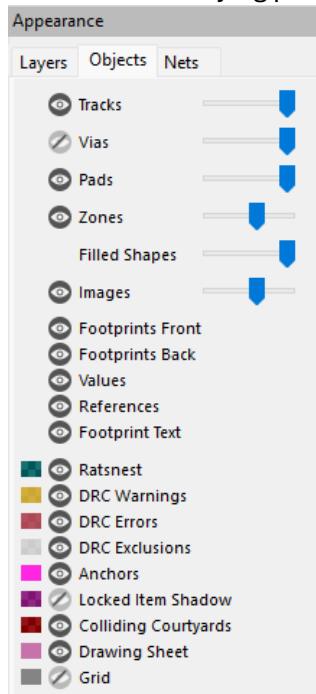
1. Place the Basic Board Outline into the Layout Workspace:

Search Filter: blank | **Library:** 0-My-Library | **Part:** Basic-Robot-Outline_1



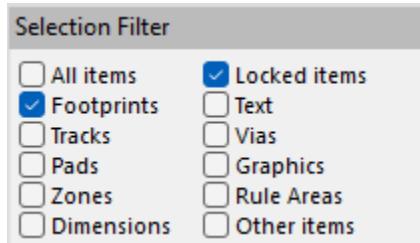
STEP 48. LOCKING UNLOCKING THE BOARD OUTLINE

1. Before moving the schematic component footprints into the board outline, let's lock the board outline so that it is not automatically selected and accidentally moved when we are trying to select other footprints, preventing confusion.
2. Double click on the board outline and checkmark the Lock field. The board outline should now be highlighted in purple-violet.
3. To hide the annoying purple highlight, in the **Objects** Tab, unselect the **Locked Item Shadow**-see below.



4. If you choose to customize your board outline, you will have to unlock it later.

- a. To unlock the board outline, in the **Selection Filter** checkmark both the **Footprints** and **Locked items**.

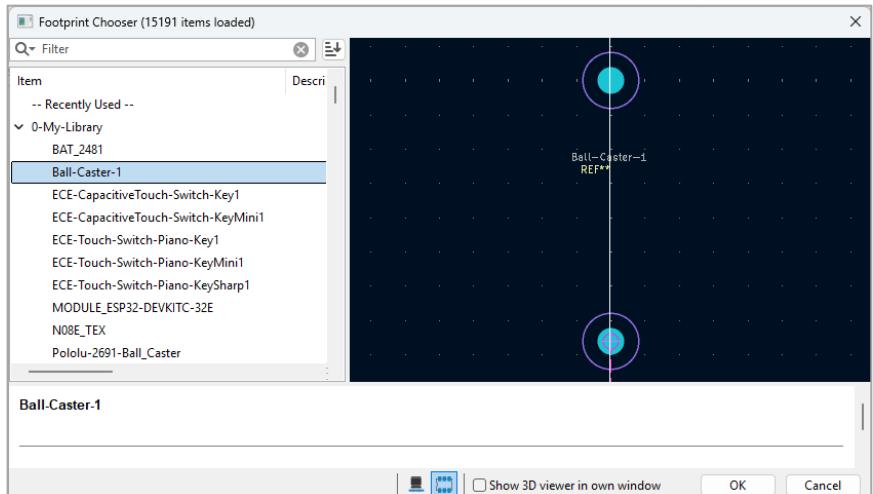


- b. Select the board outline and uncheck mark the **Lock** field box.

STEP 49. ADD REAR BALL CASTOR WHEEL FOOTPRINT

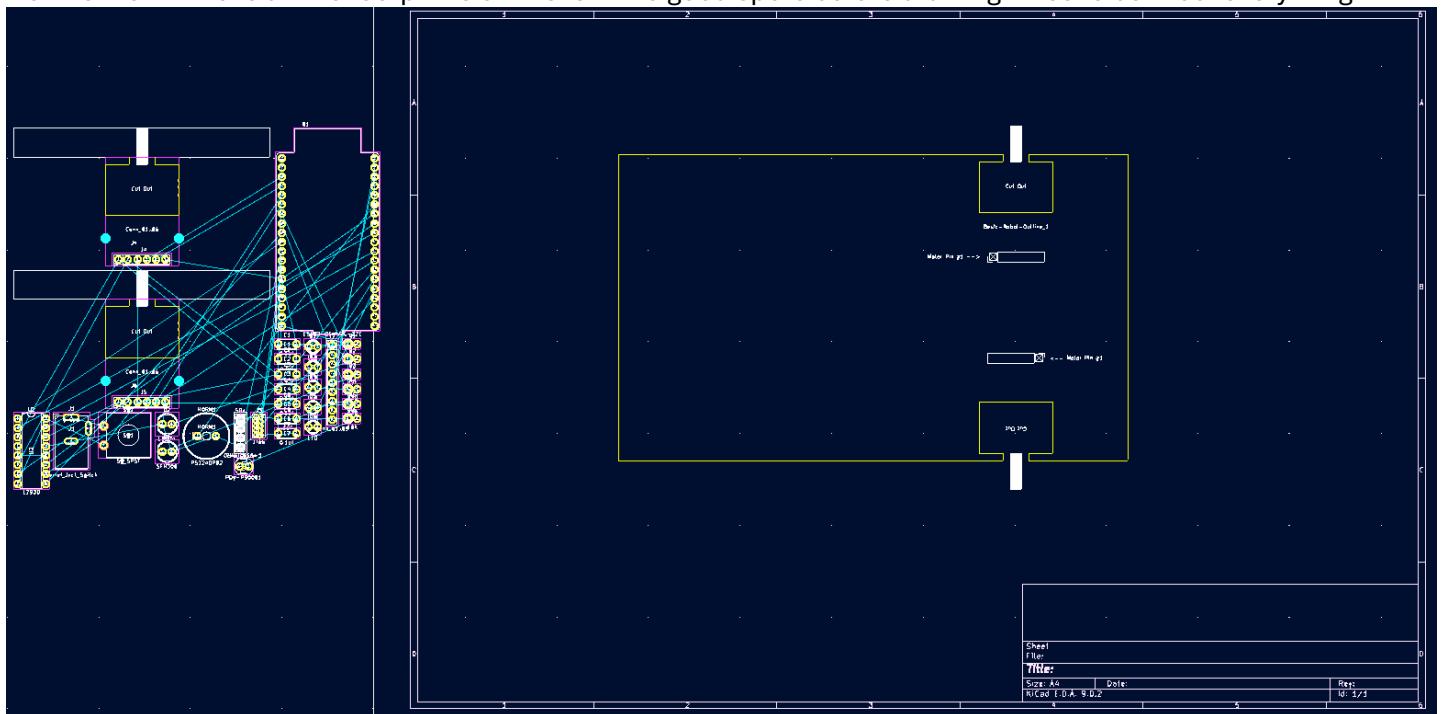
- Place the Basic Board Outline into the Layout Workspace:

Search Filter: blank | **Library:** 0-My-Library | **Part:** Ball-Caster-1



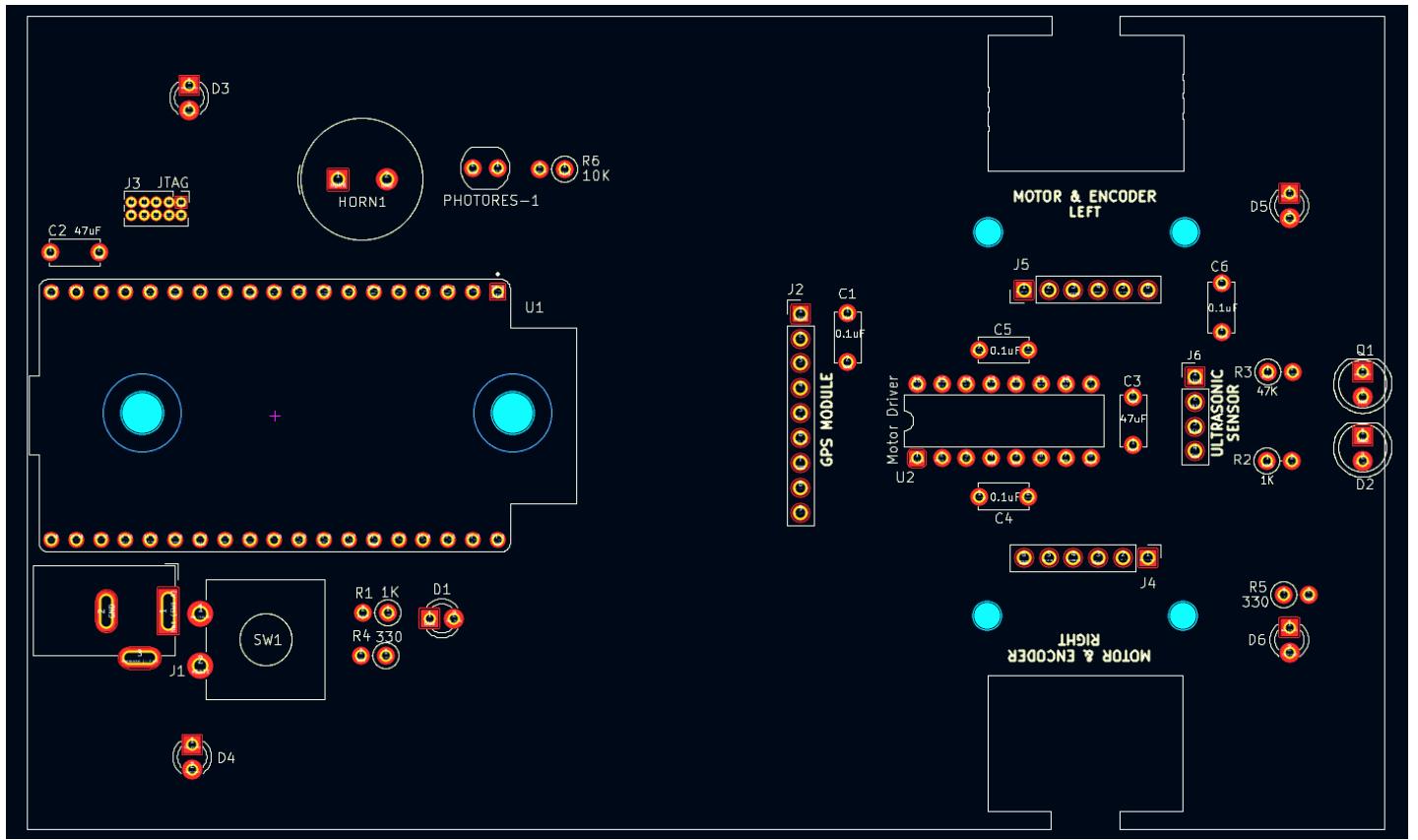
STEP 50. MOVE THE ELECTRONIC COMPONENTS INTO THE BOARD OUTLINE

Do this first — move all the footprints on the left into good spots before drawing wires to connect everything.



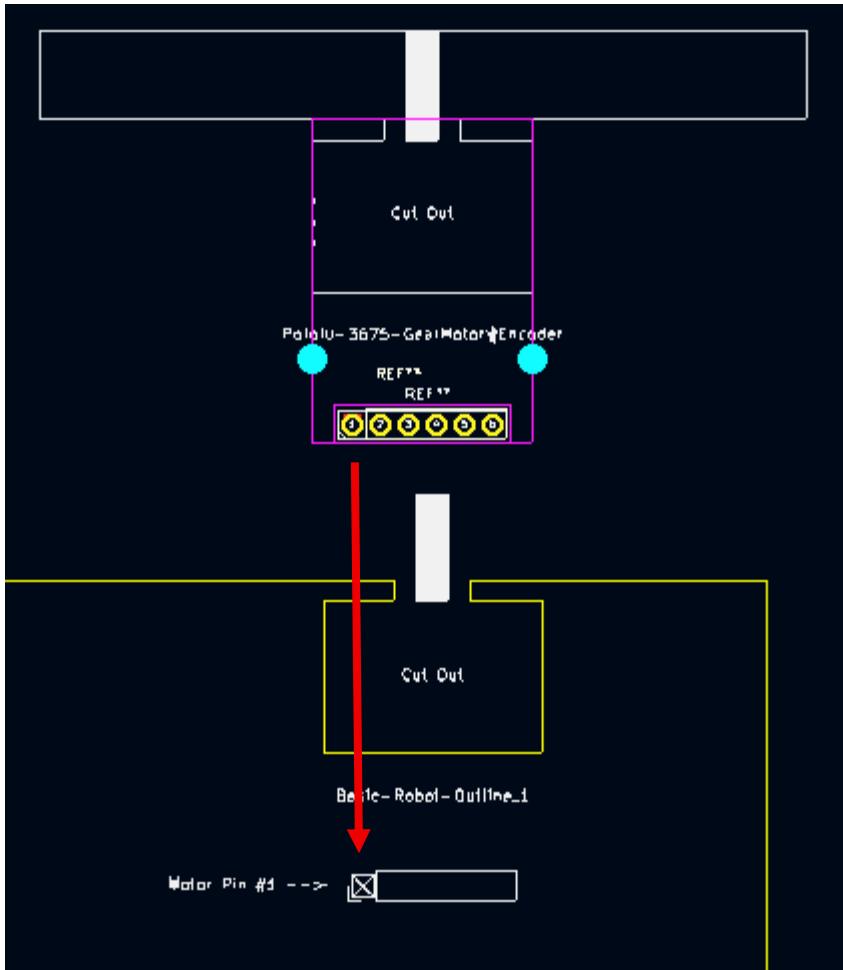
Example Placement

- Temporarily turn **F.Fab** and **F.Courtyard** off to make the layout easier to view and turn them back **on** when you need the information on those layers.
- You are not restricted to this example layout, if you can do a better, hats off to you.

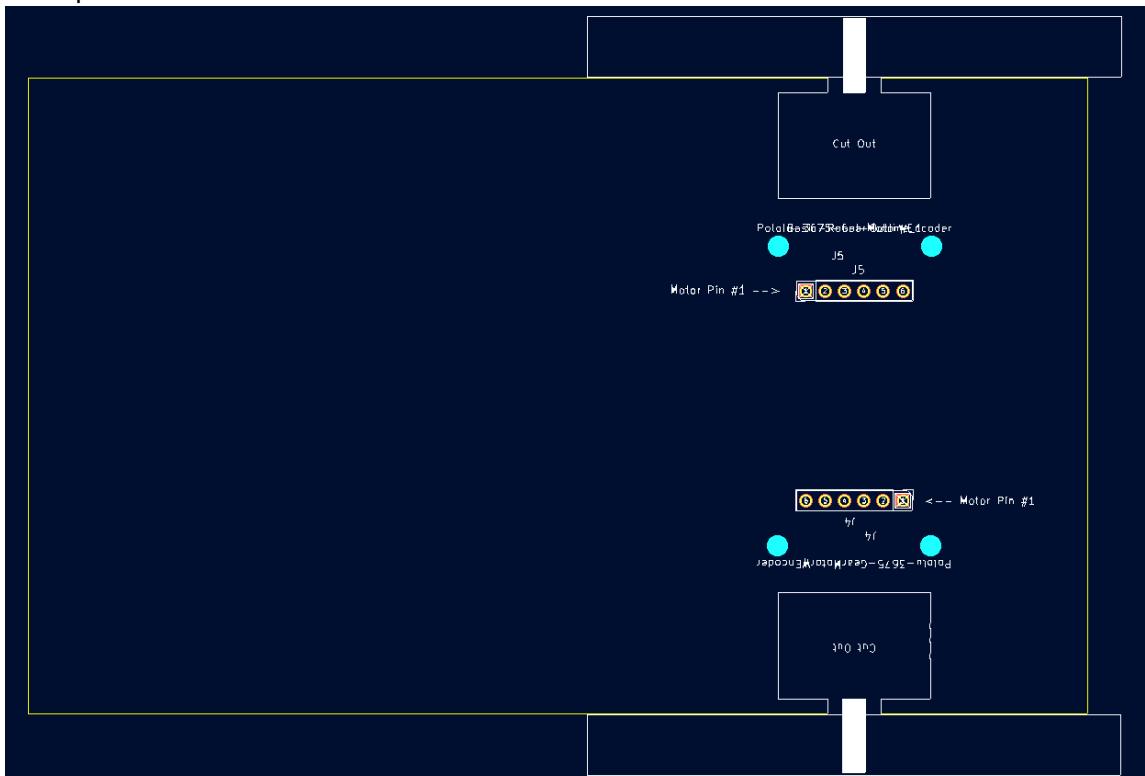


Suggest you start with:

1. Place the left and right motors, aligning the pin-1 motor connector with the provided board outline reference markers.
Pin-1 reference X marker:
2. With **Pads** enabled in the **Selection Filter**, hover over **pin 1** of the motor connector and click. KiCAD will pick up and move the footprint using that pin as the grab point, making it easier to align the connector accurately to the reference marker.
3. **Rotate:** When necessary, select the footprint then press **R** to rotate.



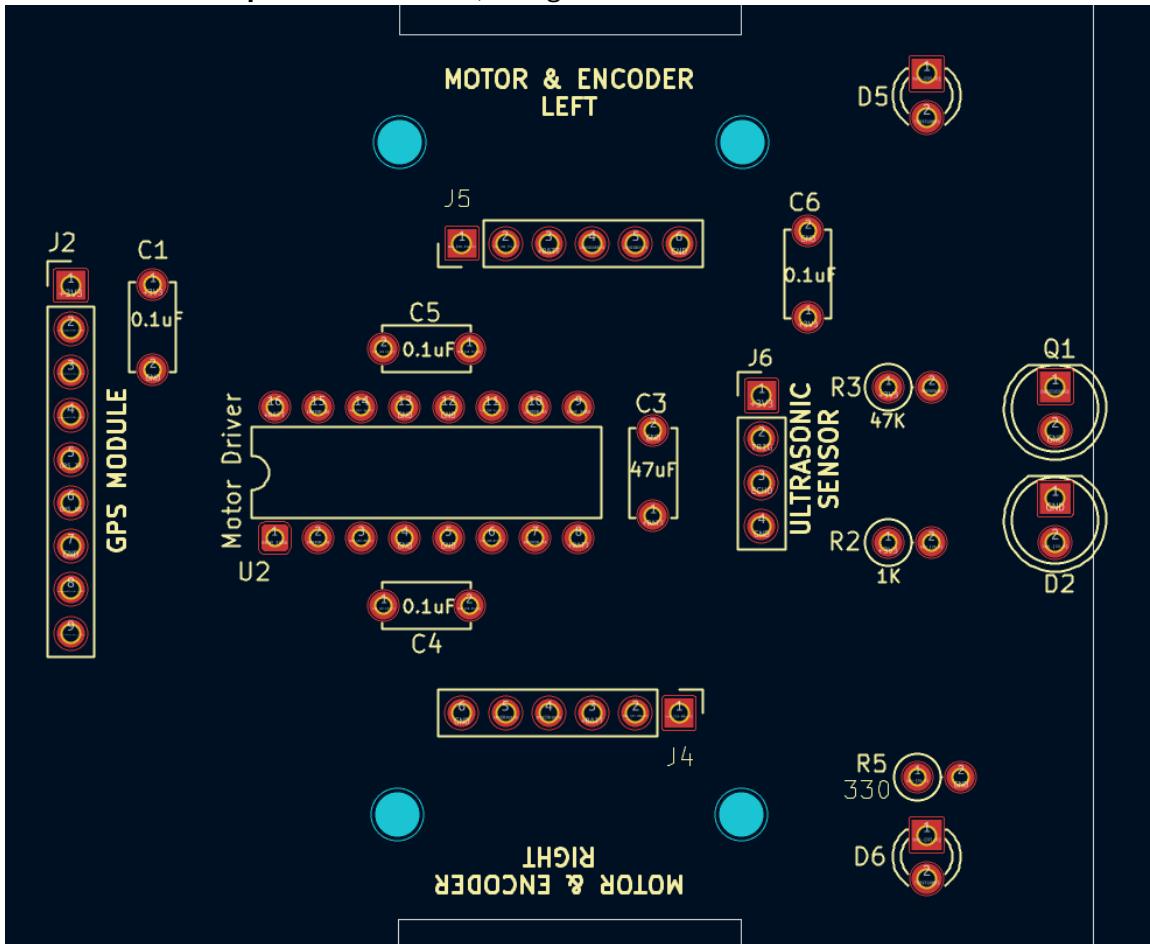
Example—Motors Placed



💡 Tip: When positioning the motors, you may need to reduce the grid setting to its smallest value for precise alignment. Grid settings were discussed earlier in this document. After placing the motors, it is strongly recommended to return to a larger grid setting when placing the remaining components.

If your **schematic reference designators** differ from these, adjust the steps as needed.

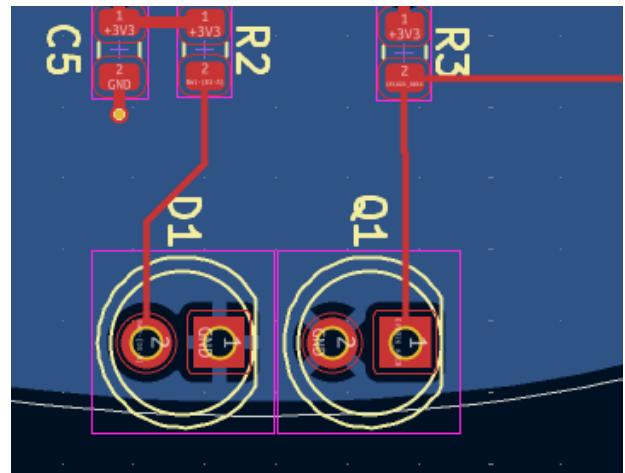
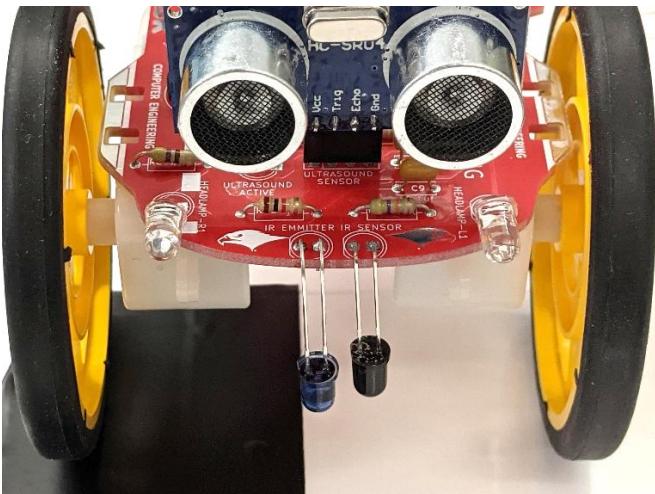
1. Place the **U2 motor driver** along with capacitors **C3, C4, and C5** (see example).
2. Place the **J6 ultrasonic sensor connector** and capacitor **C6**.
3. Place the **J2 GPS module** and capacitor **C1**.
4. Place the **line-following circuit components: Q1, D2, R2, and R3**.
5. Place the **headlamp LEDs D5 and D6**, along with resistor **R5**.



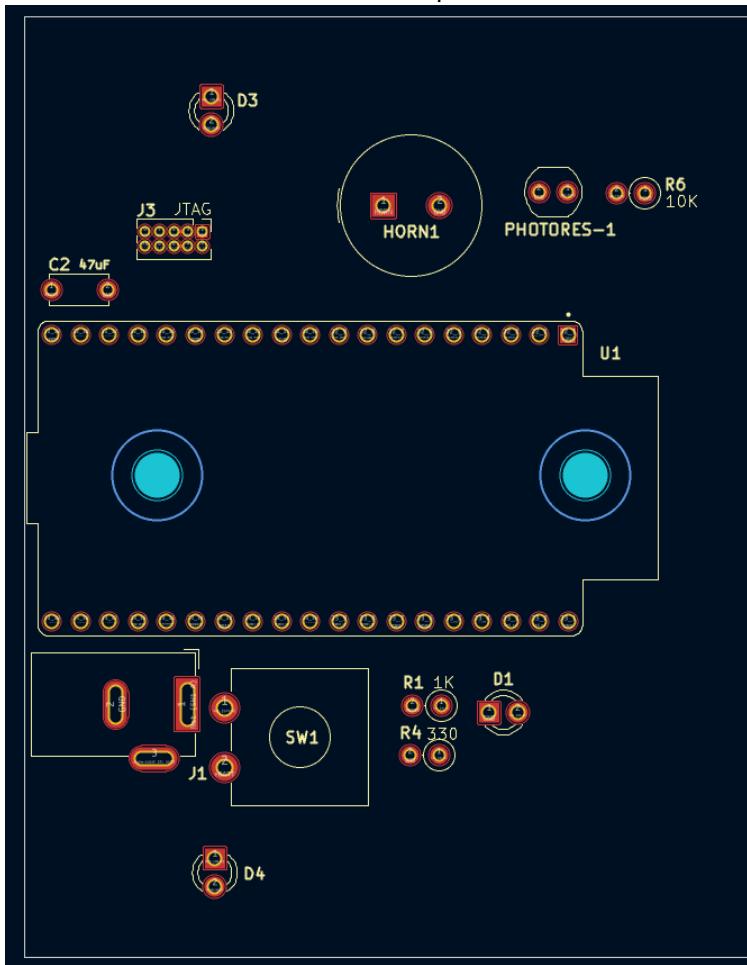
Line Following Circuit Section

Layout Key Points

- Place the line following IR emitter and phototransistor on the front edge, so that they can point downward toward the floor.



6. Place the remaining footprints as you see fit.
7. Be sure the Ball Caster Wheel footprint

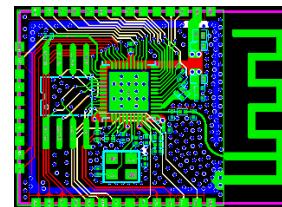
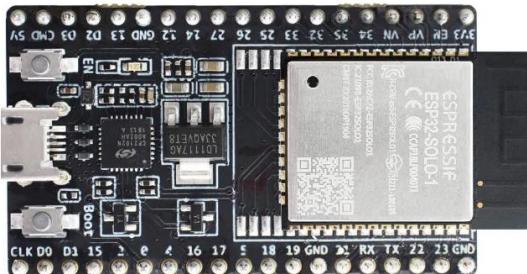


8. When placing the ESP32, make sure the **USB connector remains accessible** so the programming cable can be easily plugged in.
9. When placing the ESP32 footprint, do not place components near the antenna.

The ESP32-DevKitC is a development board that contains an ESP32-WROOM module:

ESP32 DevKitC

The ESP32-WROOM module is a small, high-density circuit board on its own

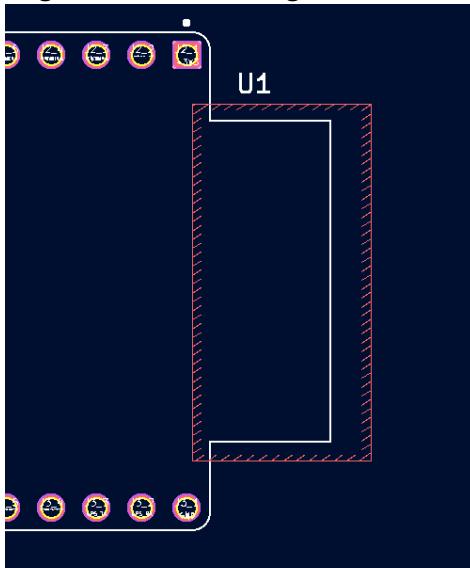


Add a copper keepout around the ESP32 antenna

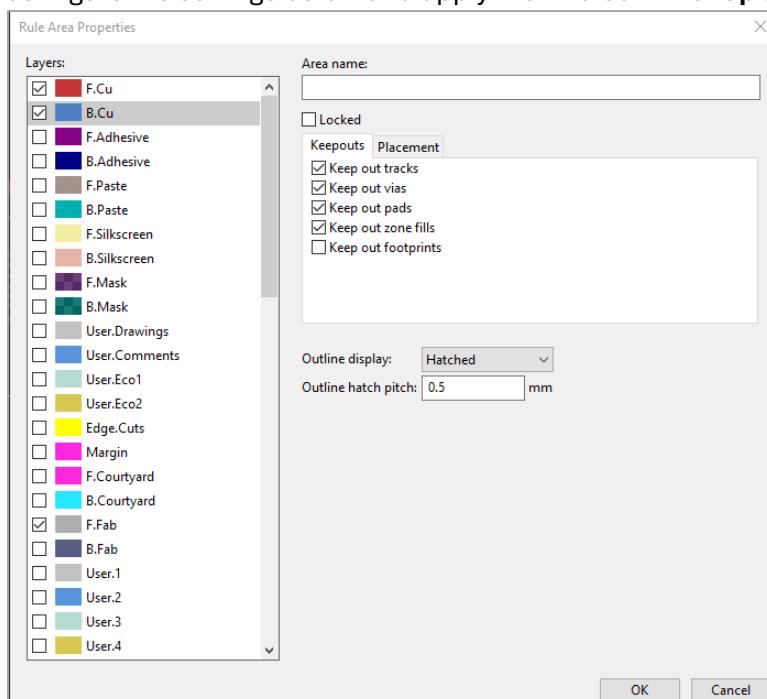
10. Select the **Draw Rule Area** tool

- Right toolbar → **Draw Rule Area**

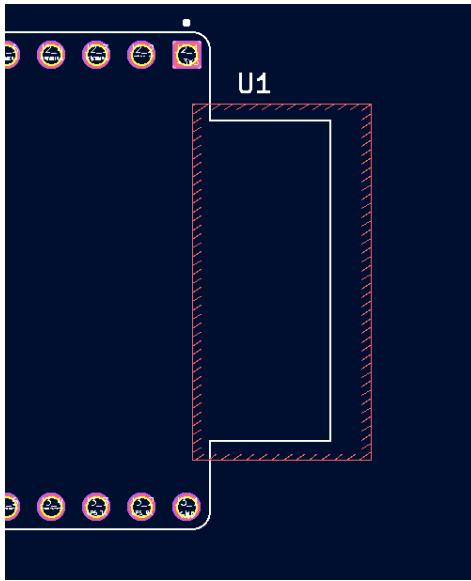
11. Begin to draw a rectangle around the **antenna end** of the U1 ESP32 module.



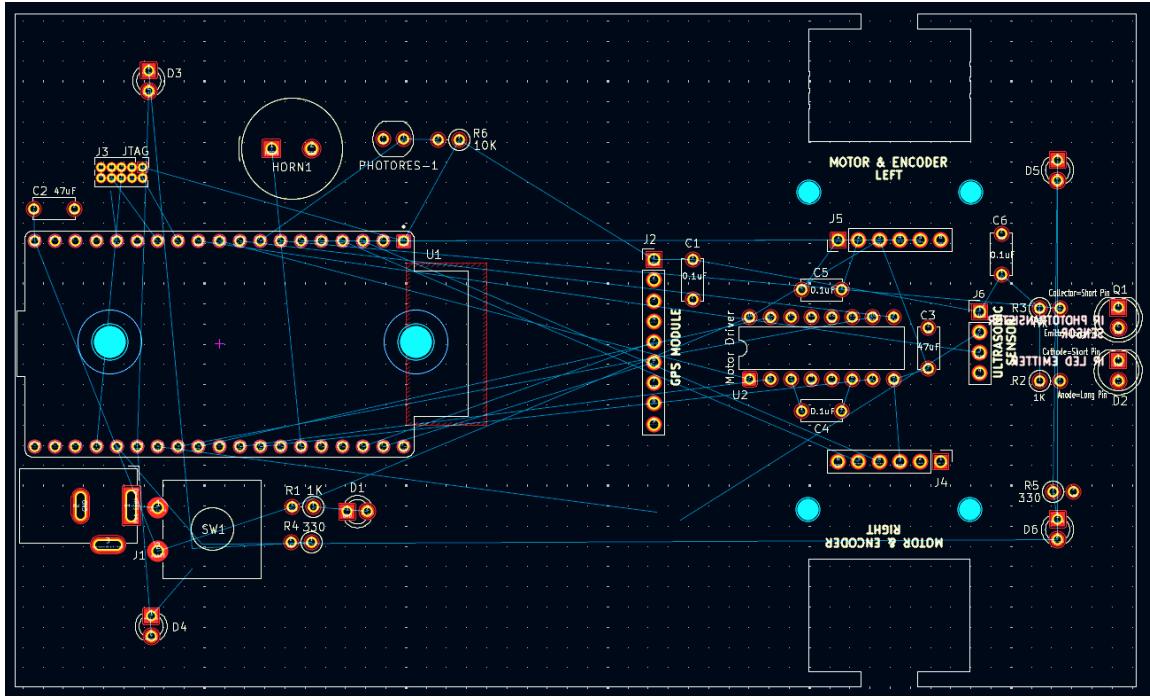
12. Click to place the starting corner of the keepout area. When the **Rule Area Properties** dialog appears, configure the settings below and apply them to both the **top and bottom copper layers**.



13. Draw a rectangle around the **antenna end** of the U1 ESP32 module
Example:



15. Place all components and review their positions to your liking. Once component placement is complete, you can begin routing the traces.

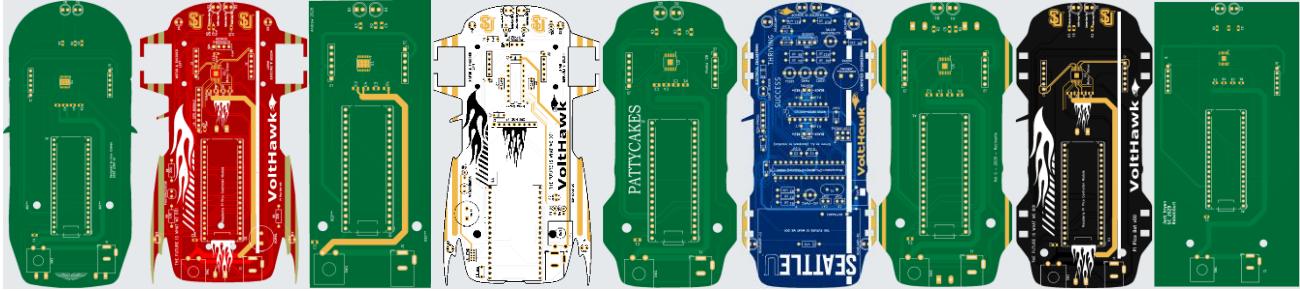


STEP 51. RATSNEST

Blue ratsnest lines show which pins still need to be connected. They are guides—not real wires—and disappear as you route the actual copper traces. When all the blue ratsnest lines are gone, it means every required electrical connection has been routed. Your PCB is fully connected according to the schematic, though it still needs design-rule checks and review.

STEP 52. SIMPLE BOARD EDGE OUTLINE OR CUSTOM?

PCB Design Club embedded robotic board lineup



STEP 53. BOARD SHAPE: SPEED VS. STATEMENT

At this point, your board outline becomes a design decision, not a requirement. A simple rectangular board gets you to a working robot faster, great for prototyping new designs and beginners. A custom shape turns your project into something memorable.

Look back at how long your schematic took to design and use that as a reality check: time spent on board shape is time taken away from finishing, or time traded from testing, debugging, or refinement. That tradeoff is something real engineers make every day.

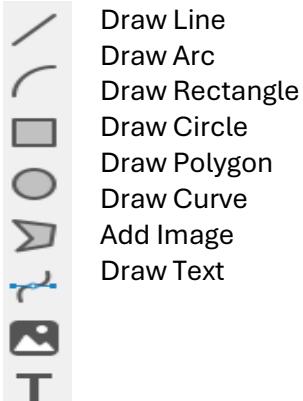
You're free to experiment—your robot doesn't have to look like a car. It could be triangular, futuristic, or unexpected. Just remember: any shape still has to respect motor placement, connectors, and real physical parts. Creativity matters—but so does execution.

With all that said, it's probably best to continue with the basic rectangular board outline and come back to this section after you've gotten familiar with adding the wire connections.

STEP 54. BOARD OUTLINE: USE PROVIDED OR CUSTOM (OPTIONAL)

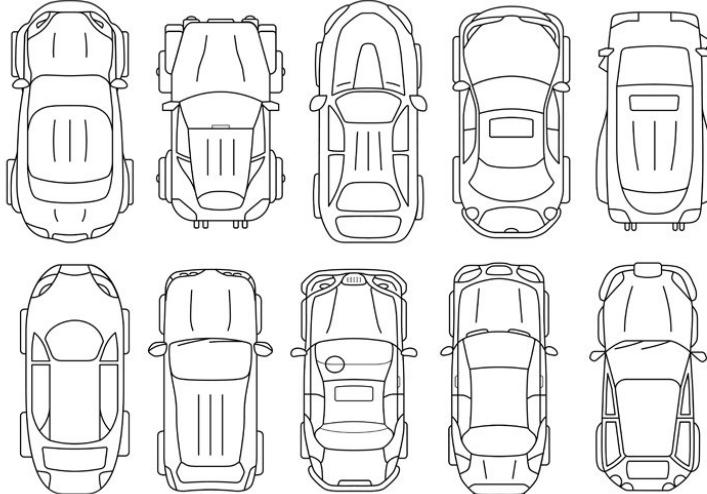
You may **use the provided Basic-Robot-Outline_1 footprint as-is**, or **optionally** choose to create a custom board outline.

1. If you decide to customize the outline, draw or modify it directly on the **Edge.Cuts** layer.
2. In the PCB Editor, make sure **Edge.Cuts** is the active layer.
3. Use one or more of the **drawing tools** from the right toolbar to create your custom outline.



4. Your board doesn't need a complex shape. Most designs use simple outlines (circles, rectangles, or basic polygons) because they're quick and easy to draw — no tracing required.

5. By searching google for “automobile topside outlines” or be specific “Porche topside outline” you can find board outlines that can then be traced on the Edge.Cuts layer.



See the section at the end of this document to add graphics and board outline images that can be traced: [Link-Adding Silkscreen Graphics & Board Outline Graphics](#)

STEP 55. ADD A BOTTOM-SIDE GROUND PLANE

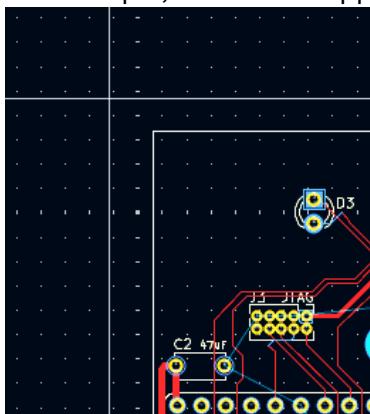
A **power or ground plane** is a large, continuous sheet of copper on a PCB that is connected to a single net, such as a power rail or ground. Instead of routing many separate traces, the board uses one shared copper area that multiple components can connect to.

In professional designs, planes are often used for both power and ground because nearly every component needs them. A large copper plane provides a clean, low-noise path and keeps other layers uncluttered for signal routing.

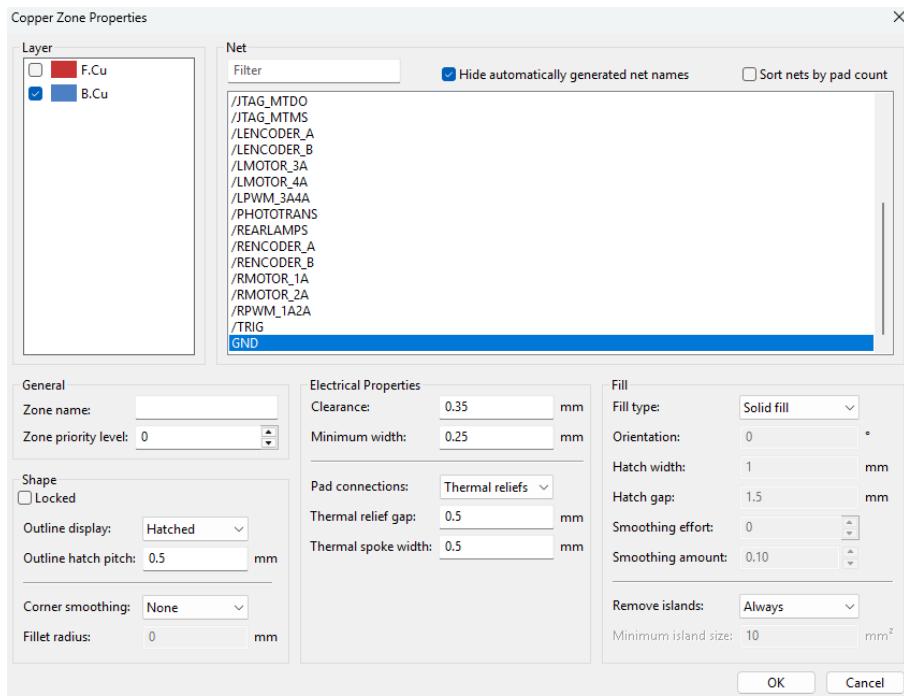
In this design, we will only add a ground plane on the bottom layer. This gives us the electrical benefits of a plane while still allowing the bottom layer to be used for some signal traces if space becomes tight—the ground pour will automatically clear around those traces.

Steps

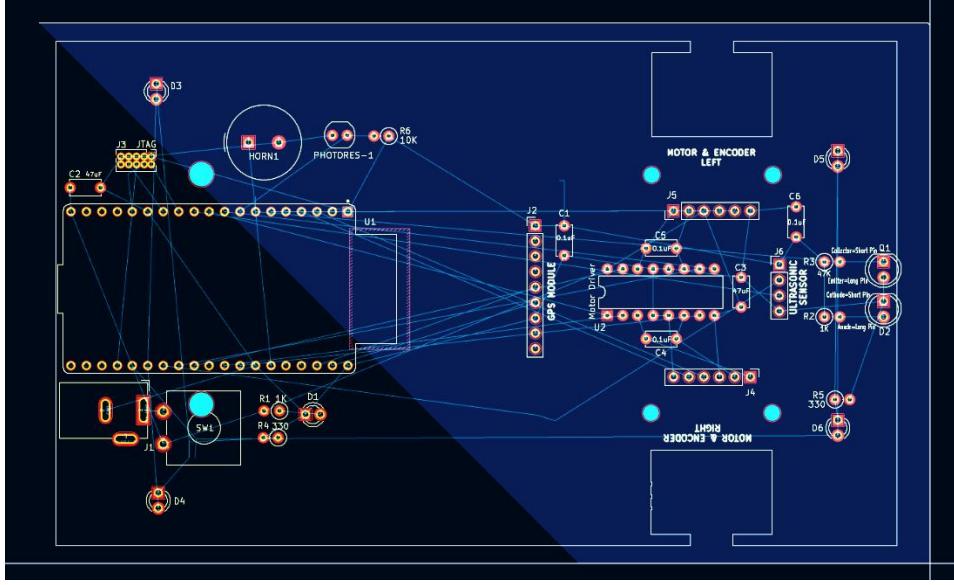
1. In the **Layers panel**, select **B.Cu** (bottom copper).
2. From the right toolbar, click **Draw Filled Zone** (icon).
3. Click in an area outside a corner of your board edge to start drawing a copper filled zone.
For example, the outside upper left corner of your board:



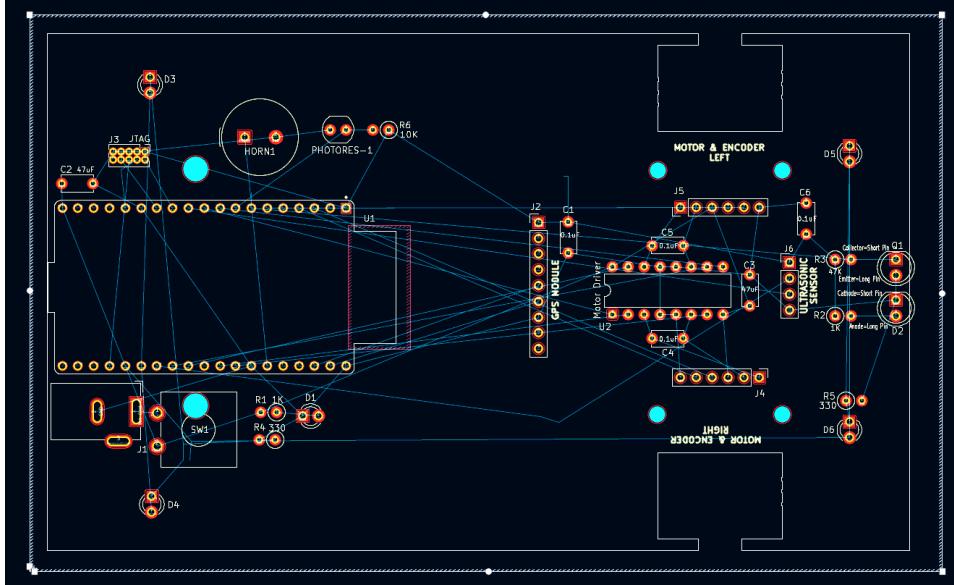
4. In the **Zone Properties** window:
 - o Set **Layer** to **B.Cu**
 - o Set **Net** to **GND**
 - o Leave the default clearance and thermal relief settings
 - o Click **OK**



5. Continue drawing to form a rectangle around your board outline on the outside.



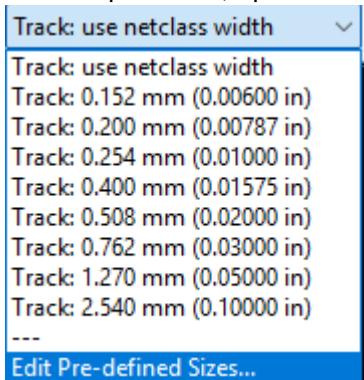
6. Zoom in to finish the last corner and close the shape—rectangle around board outline.



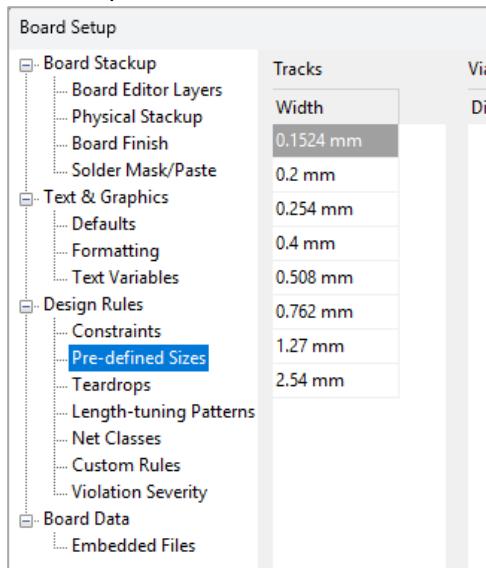
7. Select **Edit → Fill All Zones** or press **B** to **fill the zone with copper on the bottom layer—a ground plane.**
8. Your bottom layer is now a ground plane. Ground pins will connect automatically, and any bottom-layer signal traces you add later will create clearances through the plane.
9. After moving components or adding traces to the bottom layer, press **B** at any time to refill the zone.

STEP 56. BEFORE DRAWING YOUR COPPER WIRES (CALLED “TRACKS” OR “TRACES”)

1. In the top toolbar, open the Track Width dropdown and select Edit Pre-defined Sizes.



2. Add the pre-defined track widths to the list.



3. These sizes will be available for quick selection while routing nets in the next step.
4. Use 0.1524 mm (6 mil) primarily for signal traces and select wider traces for power and ground connections.

Finally! Let’s route the board!

STEP 57. DRAW THE COPPER WIRES (CALLED “TRACKS” OR “TRACES”)

Now you connect the components with thin copper lines that carry electricity.

1. Click the **Route Single Tracks** tool



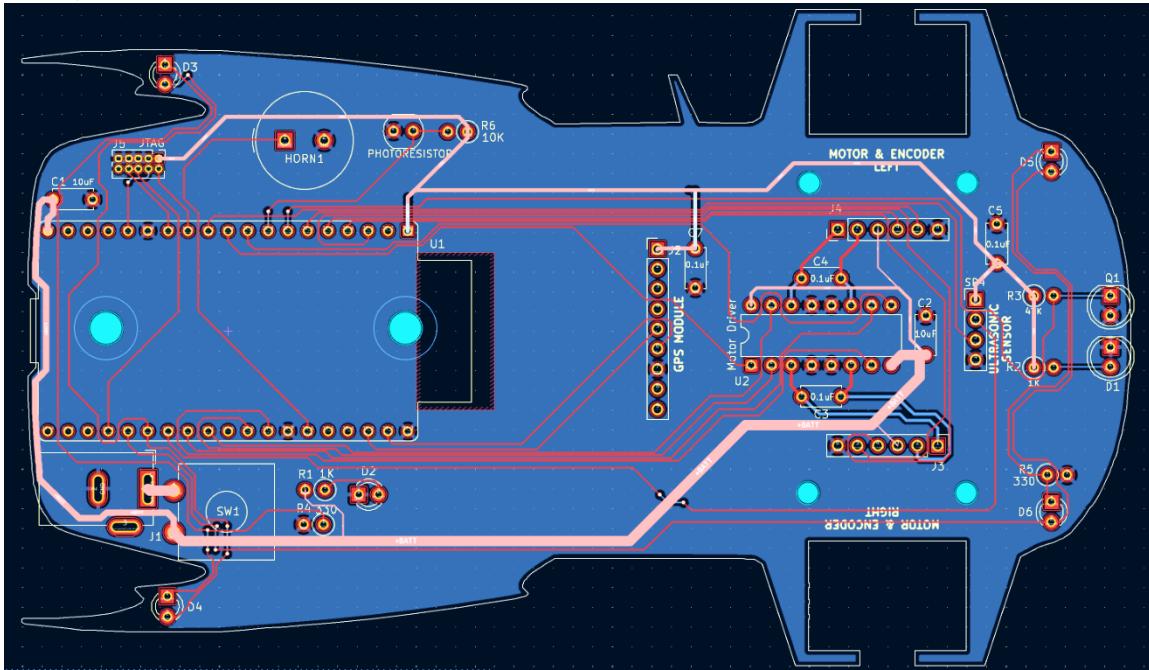
2. Click on one metal pad (the shiny circle or square around a component pin) and drag to another pad it needs to connect to. A copper line will follow your mouse.
3. KiCAD shows thin gray “guide lines” (called the ratsnest) – these tell you exactly which pads still need to be connected. Your job is to draw copper wires along those gray lines until every single ratsnest disappears.
4. You have two copper layers to draw tracks (wires) on:
 - Top side = F.Cu (red by default)

- Bottom side = B.Cu (blue by default)

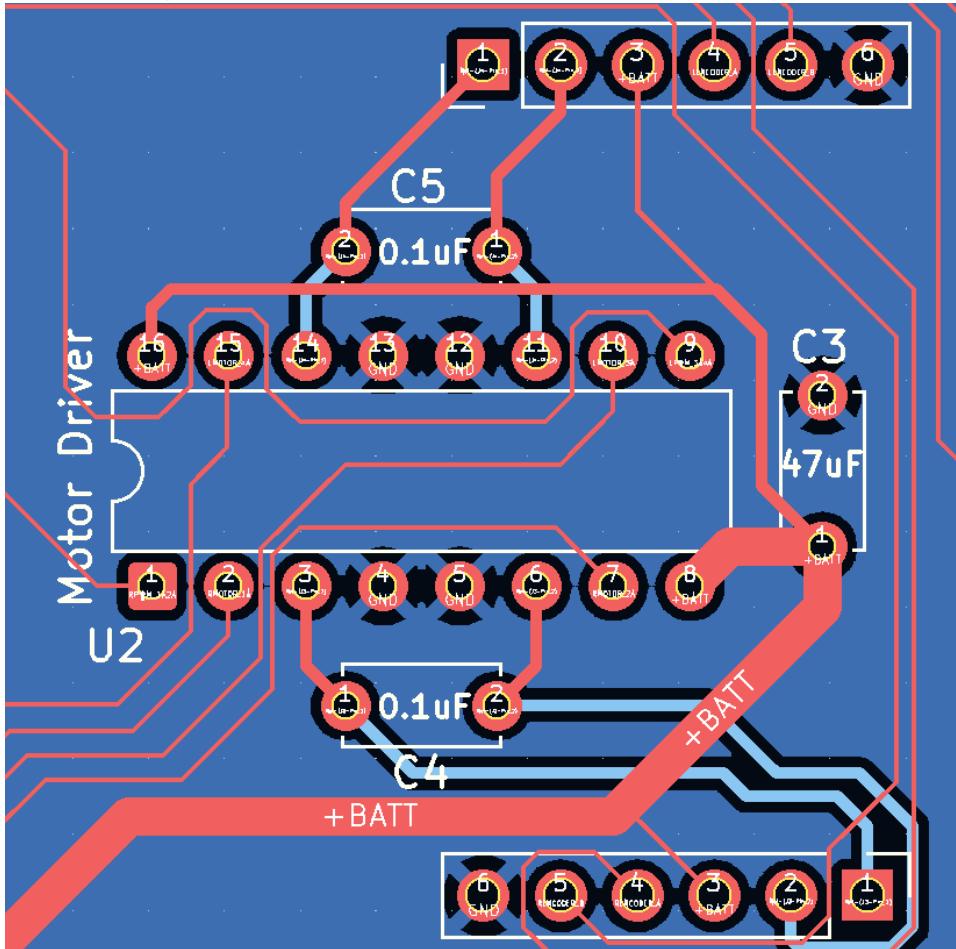
Best beginner strategy:

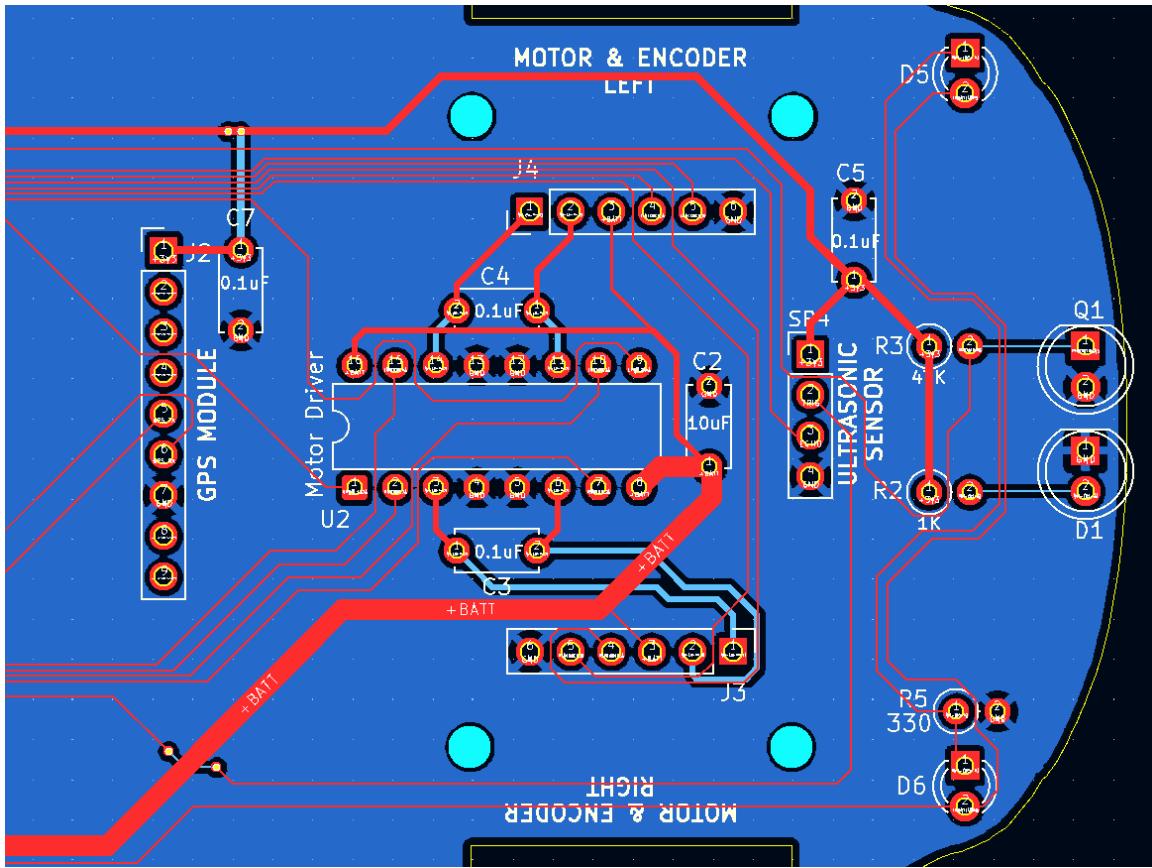
5. Start by routing the **power connections** on the top copper layer—see the pink highlighted nets connected to **J1** Barrel Jack input, **SW1** Power Switch, and **+BATT**, **+3V3** nets.

Example:

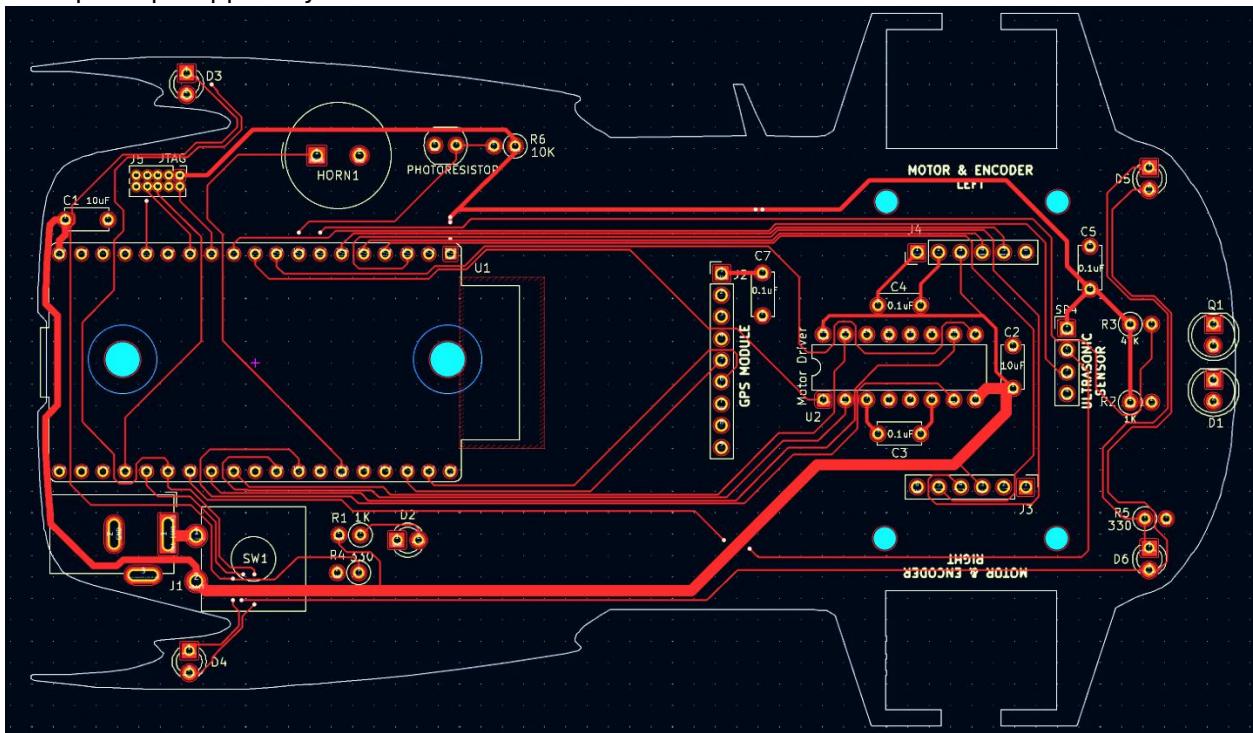


6. Route the caps C3, C4, C5, U2 motor driver à to J4 & J5 motor connectors.

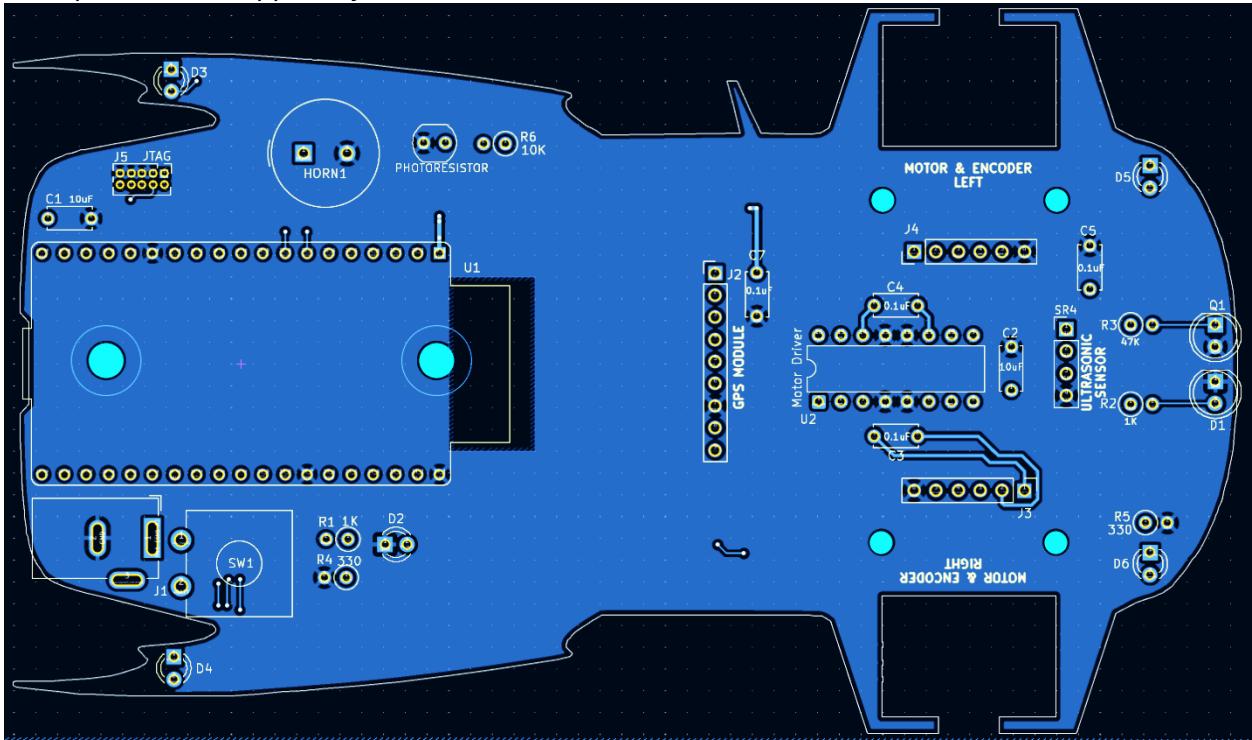




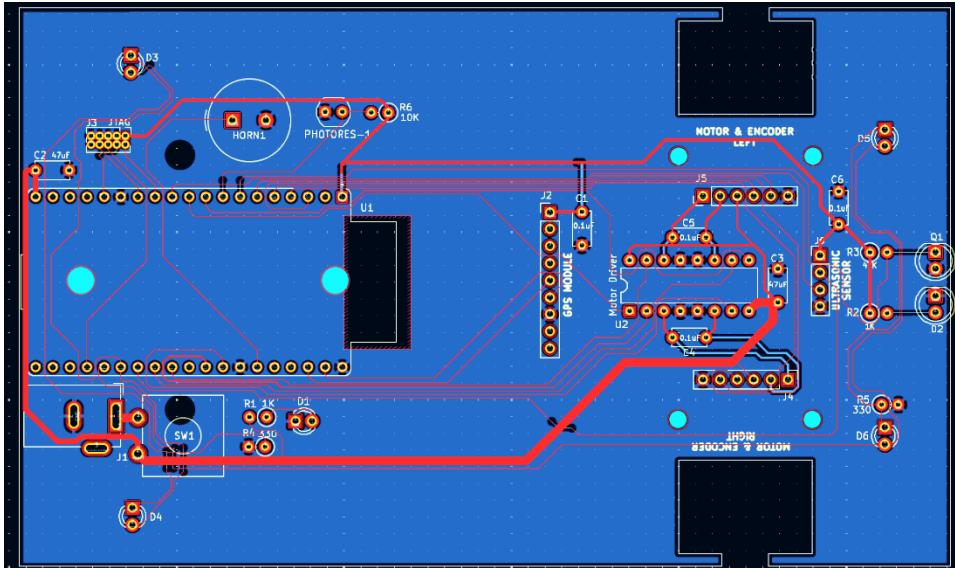
7. If two copper lines absolutely must cross: While drawing, press V (via) this places a via (a tiny plated hole). The track automatically jumps through the via to the other layer, so the lines never touch or short-circuit.
8. Goal: When all the gray ratsnest lines are gone, your circuit is fully connected and ready!
9. Example Top Copper Layer



10. Example Bottom Copper Layer



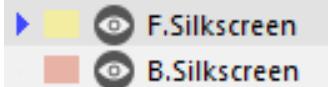
11. Example Rectangle Board



Almost done!

STEP 58. ADD YOUR NAME AND MESSAGE TEXT TO THE BOARD

1. Add your board ID Place your first name + last initial or full name. Include your graduation year. Put this on the F.Silkscreen layer. Or you can add it to the bottom (B.Silkscreen) if you like.
2. Add extra fun text or graphics, get creative!
3. Use the Text tool and place the text on F.Silkscreen (or B.Silkscreen for the back).



4. That's it — your board now has your personal signature and message!

5. ARE YOU DONE?



STEP 59. CHECK YOUR BOARD FOR MISTAKES (RUN DRC – DESIGN RULES CHECK)

Before you send your board to be manufactured, KiCAD can automatically find common errors (like wires touching, parts too close, missing connections, etc.).

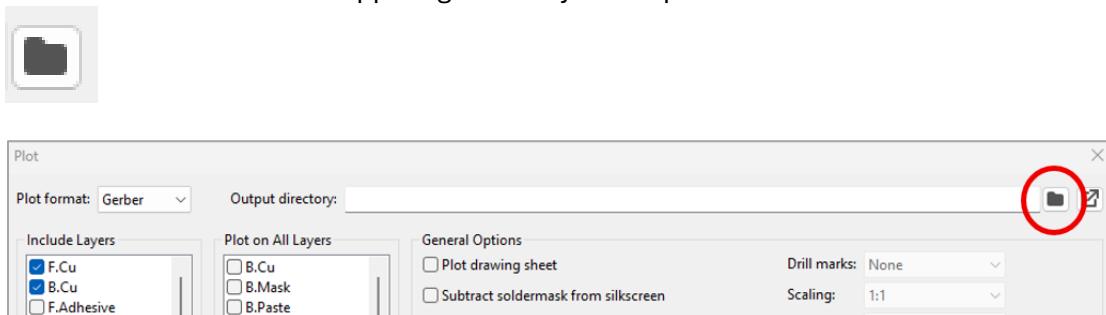
1. Save your project first (always a good habit!).
2. Go to the menu **Inspect → Design Rules Checker**.
3. A window opens — just click **Run DRC** (big green button).
4. Wait a few seconds. When it finishes, look at the bottom part of the window:
 - o If it says, “**0 unconnected**” and “**0 errors**” → Perfect! Your board is ready.
 - o If you see red or orange messages:
 - **Errors** (red): Must fix these (example: two tracks touching, missing connection).
 - **Warnings** (orange): Usually okay for simple student boards (ignore for now).
5. To fix an error:
 - o Zoom in on your board, click on the error line in the list → KiCAD zooms and puts an arrow on the problem.
 - o Fix it (move a track, add a missing connection, etc.), then click **Run DRC** again.

Goal: 0 errors and 0 unconnected items → your board will work and will be accepted by the factory!

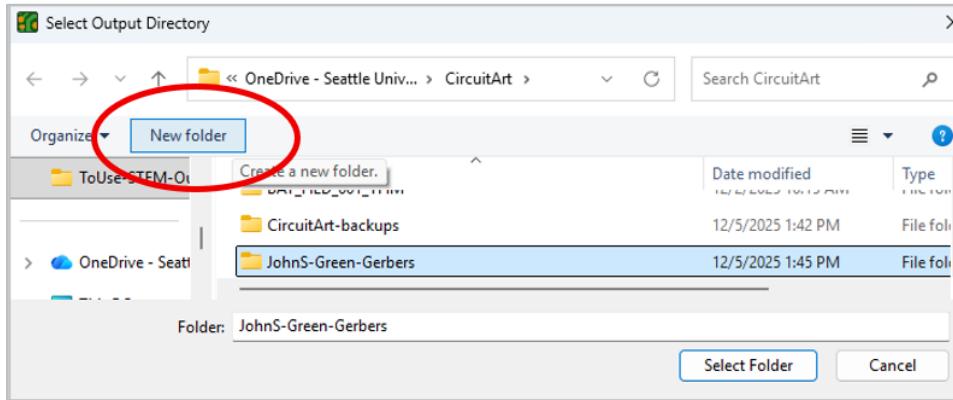
STEP 60. GENERATE FILES FOR THE FACTORY (GERBER FILES)

You’re done designing — now create the files the factory needs!

1. In PCB Editor, go to File → Fabrication Outputs → Gerbers.
2. Click the folder icon in the upper right to set your output folder.



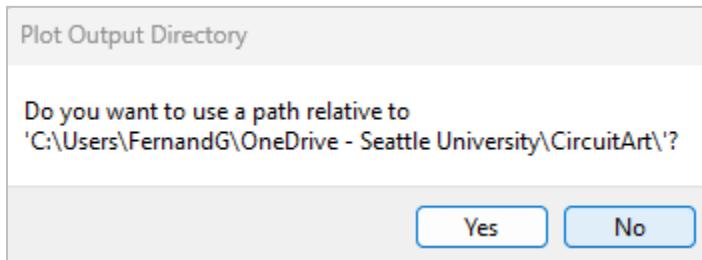
- In the Select Output Directory popup, click **New folder** button.



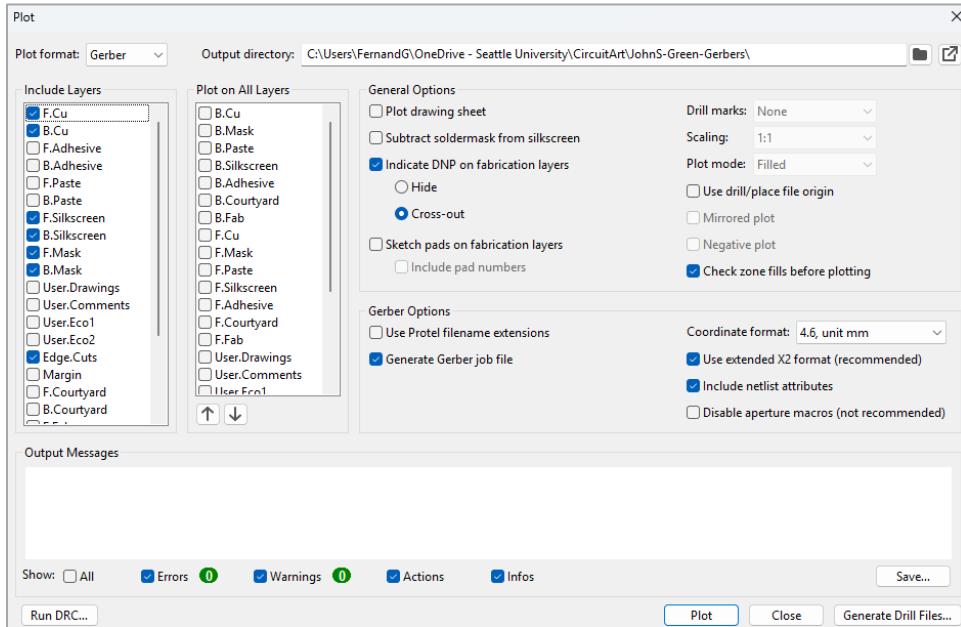
- Name the folder FirstnameLastinitial-BoardColor-Gerbers

Example: JohnS-Green-Gerbers (pick your solder mask color: Green, Blue, Red, Yellow, Black, or White)

- Click Select Folder button.
- Click the No button.



- Make sure these layers listed below in the Include Layer panel are checked (the usual ones for a 2-layer board):

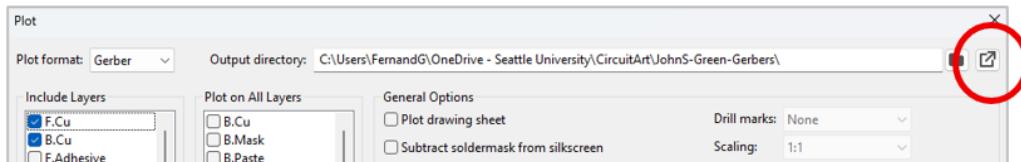


- At the bottom, click **Plot**. KiCad will create a bunch of gerber files that end in .gbr in your gerber folder.
- Now click **Generate Drill Files** (same window at the bottom). Failure to take this step will prevent your board from being assembled because it will not have any holes to mount the parts.
- Close the window.

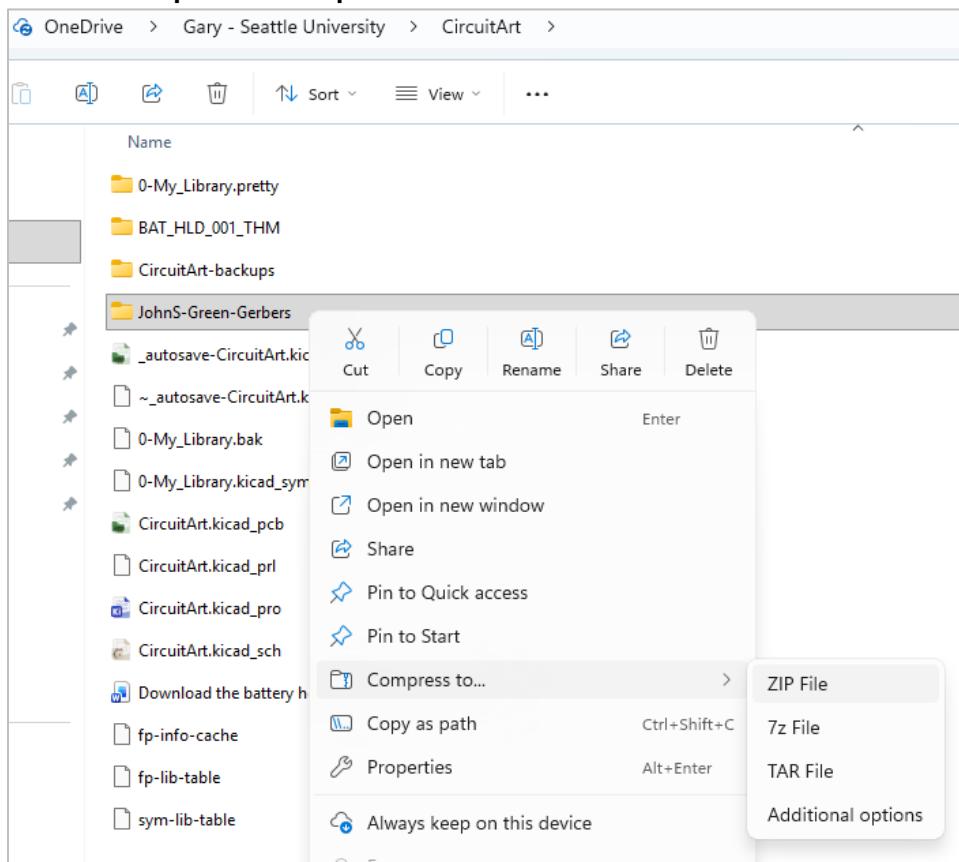
STEP 61. ZIP AND NAME YOUR FILES (IMPORTANT!)

- Open the folder you just created (the one named like JohnS-Green-Gerbers).

The easiest way: in the KiCad Plot window, click the small folder icon in the top-right corner – it opens the folder for you automatically!



2. Right-click on the Gerber folder inside your project.
3. Choose **Compress to... Zip File**



4. Your zip file should already be named something like JohnS-Green-Gerbers.zip

STEP 62. CHECK YOUR FILE ON THE MANUFACTURE'S WEBSITE

1. Go to the manufacturer's website <https://jlpcb.com/>
2. Click the button.

Instant Quote

3. Upload your **zip** file.

Add gerber file

4. Wait to get up to 100%.
 5. Select the various PCB Colors
- | | | | | | | | | | | | | | | |
|-----------|--|-------|--|--------|--|-----|--|--------|--|------|--|-------|--|-------|
| PCB Color | | Green | | Purple | | Red | | Yellow | | Blue | | White | | Black |
|-----------|--|-------|--|--------|--|-----|--|--------|--|------|--|-------|--|-------|
6. Be curious, check out the pricing for various quantities.
 7. LOOK AT THOSE TARIFF AND DUTIES FEES—Oh No 😊
 8. On the webpage, click on the Gerber Viewer and check out your board with more details.

Gerber Viewer

STEP 63. PCB PEER REVIEW & QUALITY CHECK

1. In Canvas → Files, complete the PCB peer review before you proceed.
2. The reviewer completes the checklist, writes any notes, and signs the form.
3. You fix anything marked “Needs Fix,” then submit your completed .pcb file and gerbers zip file on Canvas.

DONE! If this is the first board you’ve designed—congratulations embedded engineer!

ADDING GRAPHICS AND BOARD OUTLINES

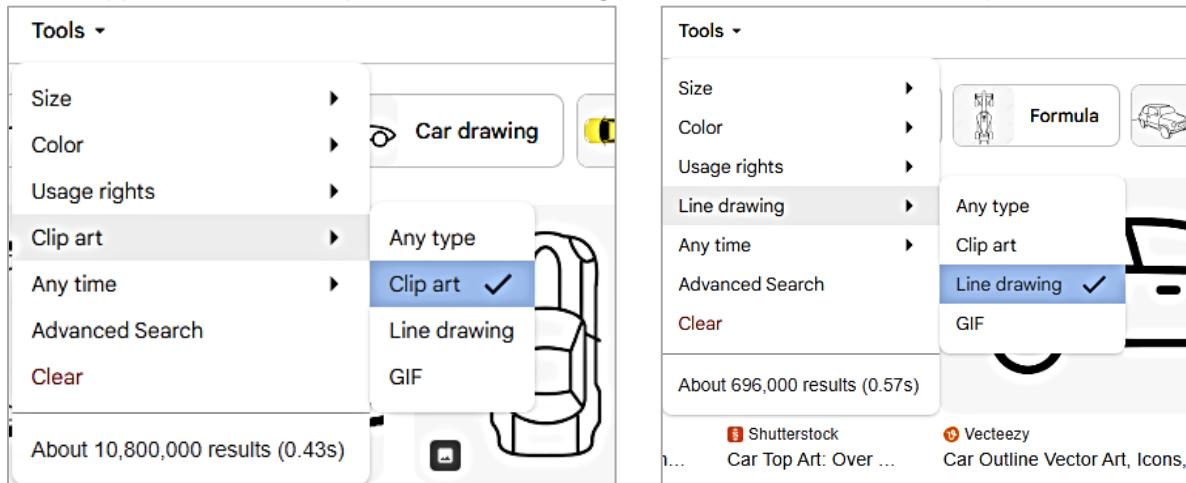
<unfinished section>

STEP 64. CHOOSE YOUR ARTWORK

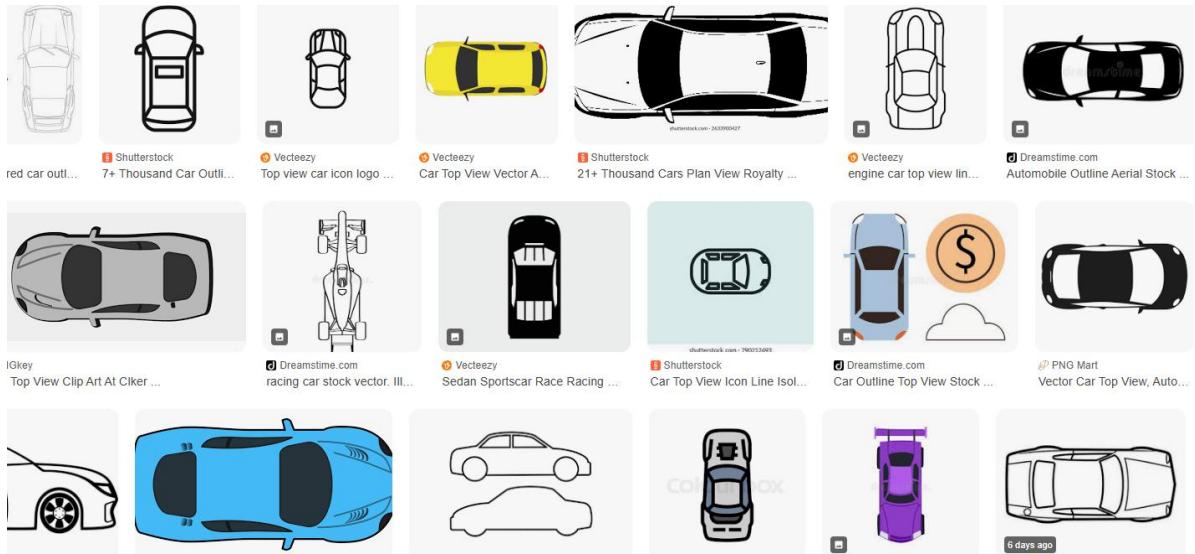
1. For example, go to the google image search [Google Images](#). In the search field, type the image you are looking for, such as “**automobile** (or be specific to the model) **topside topview outline**”.
2. In the google results, you may also get better results by clicking the **Tools** dropdown menu-see below



3. Select **Type** then select **Clip art** or **Line drawing** to filter the results. Each one produces different results.



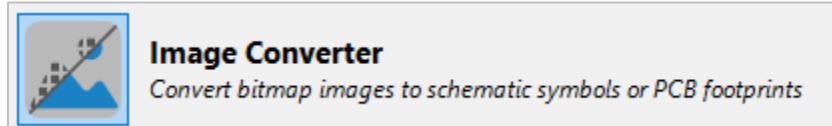
Filtered Results



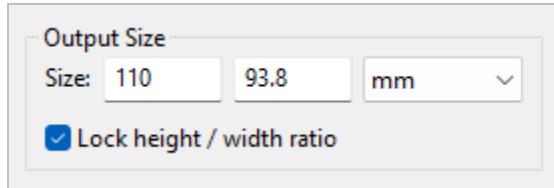
4. Choose **non-copyrighted images**.
5. For better quality, use **higher-resolution images** by clicking the thumbnail to open the original source page. Thumbnail images are usually **low resolution**, but they may still work for this project.
6. Using the **Windows Snipping Tool** to capture and save the image produces good results. Or right-click an image and choose **Save As**. Your file must be saved as **.JPG** or **.PNG**. If the image is saved as a **.webp** file, convert it to **.JPG** or **.PNG** using Paint or any online converter, or use the Windows Snipping Tool.
7. If your picture needs any editing or cropping, use Windows Paint or the photo viewer-editor.

STEP 65. KICAD IMAGE CONVERTER

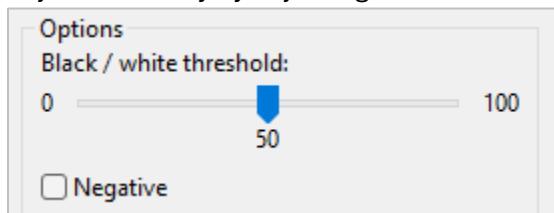
1. Open KiCad Image Converter (found under Tools → Bitmap to Component Converter).
2. From the **Project Window**, open the Image Converter by clicking on the **Image Converter** icon

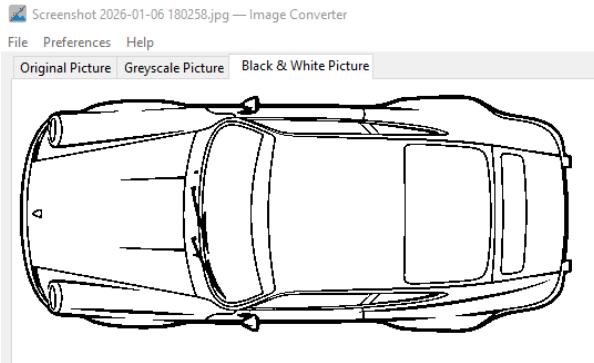


3. Load your image file.
4. Adjust the size. Try starting at about 130 x 85. Experiment for best results.

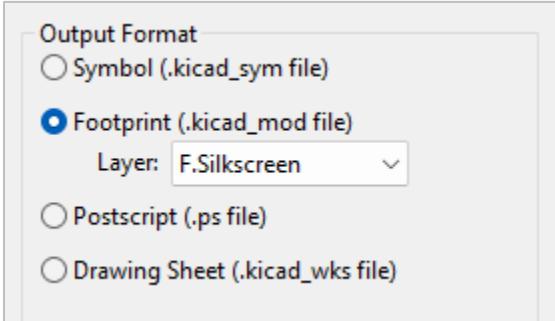


5. Adjust for clarity by adjusting the black/white threshold as needed.

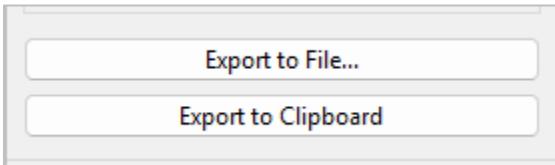




- Set the Output Format to **Footprint Layer**, then choose **F.Silkscreen** from the dropdown. This is the layer that prints text and graphics on the top of your circuit board.

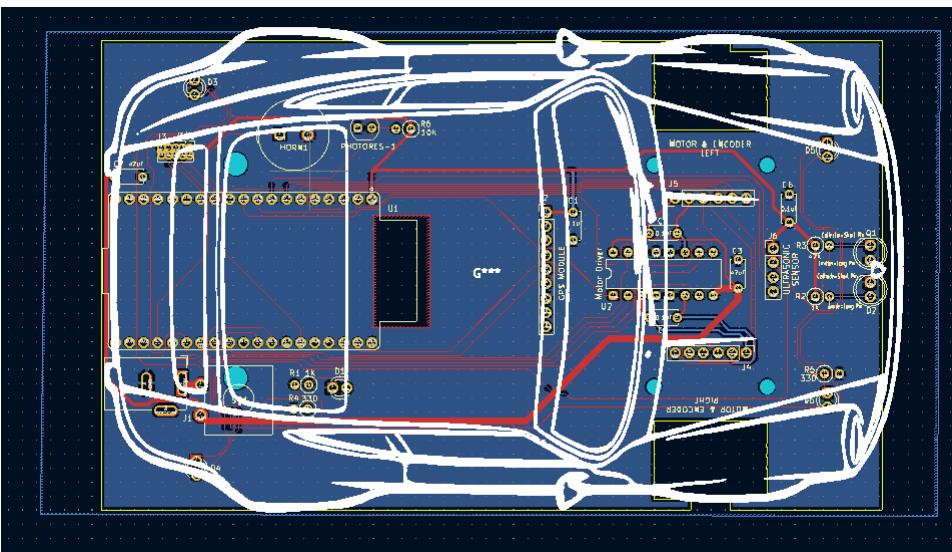


- Export your image to the **Clipboard**



STEP 66. PASTE IMAGE FROM THE CLIPBOARD

- In the **PCB Editor**, press **Ctrl + V** to paste the image into your layout area inside the fab box.



- If needed, rotate the image or the fab box by selecting them and pressing the **R** key.
- If the image size still needs adjustment, resize it in the KiCad **Image Converter** or another image-editing tool before re-importing.

4. Multiple images can be imported into your board.
5. You can then trace an outline on the Edge.Cuts layer, if this was for a board outline. Otherwise, it is graphics on the silkscreen layer.

Tricks—How to Create Shiny Metallic Copper Artwork (Very Cool Effect!)

Steps:

1. Draw a solid copper area on the copper layer (use a filled zone or copper pour).
2. Draw your text, logo, or shape on the solder mask layer exactly over that copper.
3. The factory removes solder mask wherever you drew on solder mask → bright tinned copper shines through!
4. Don't overlap text on both silkscreen and solder mask in the same spot — one will hide the other. Silkscreen can only be printed over solder mask.
5. Popular metallic design ideas
 - Company or project logo
 - Geometric patterns or borders
2. Quick checklist
 - Normal white text/logos → place on silkscreen
 - Shiny copper features → place on solder mask over copper
3. That's it! Once you master these four layers (Copper → Mask → Silkscreen), your boards will look as good as they work.
4. Have fun designing!



Design a gift,
Design an award,
Design a product to sell,
Design your dream,

Design a plaque with a famous phrase or poem,
Design name tag or business card,
Design your expression and passion!

