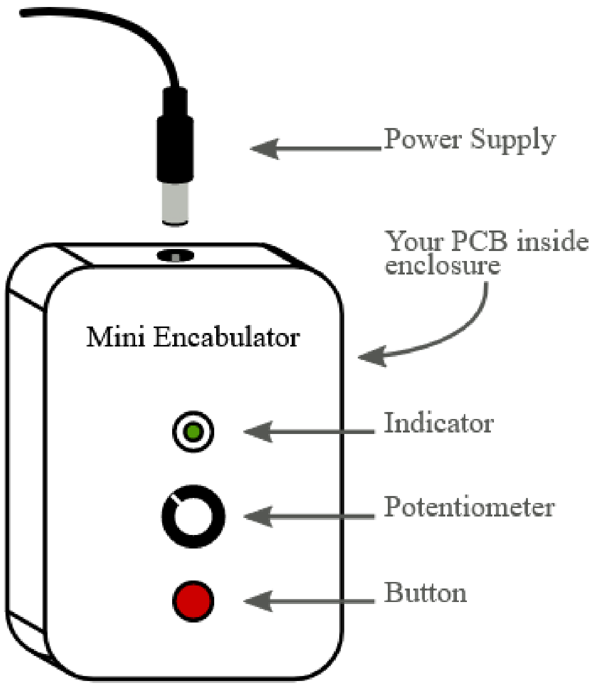
PCB Design Exercise

This assignment was designed for KiCAD 5.1.10. Some menu options may look different in KiCAD 6 and above, but all the steps should generally be the same. If you have any questions, feel free to ask a TA.

In this exercise, you will be designing a PCB for a mini encabulator. This simple device only does two things:

- Blinks an LED when the user holds down a button.
- Sets blink rate proportional to potentiometer position.

A visual aid for the miniencabulator is shown below:



You will be using KiCAD (pronounced KEY-cad), a design tool for creating electrical schematics and board layouts. This assignment will have 3 stages:

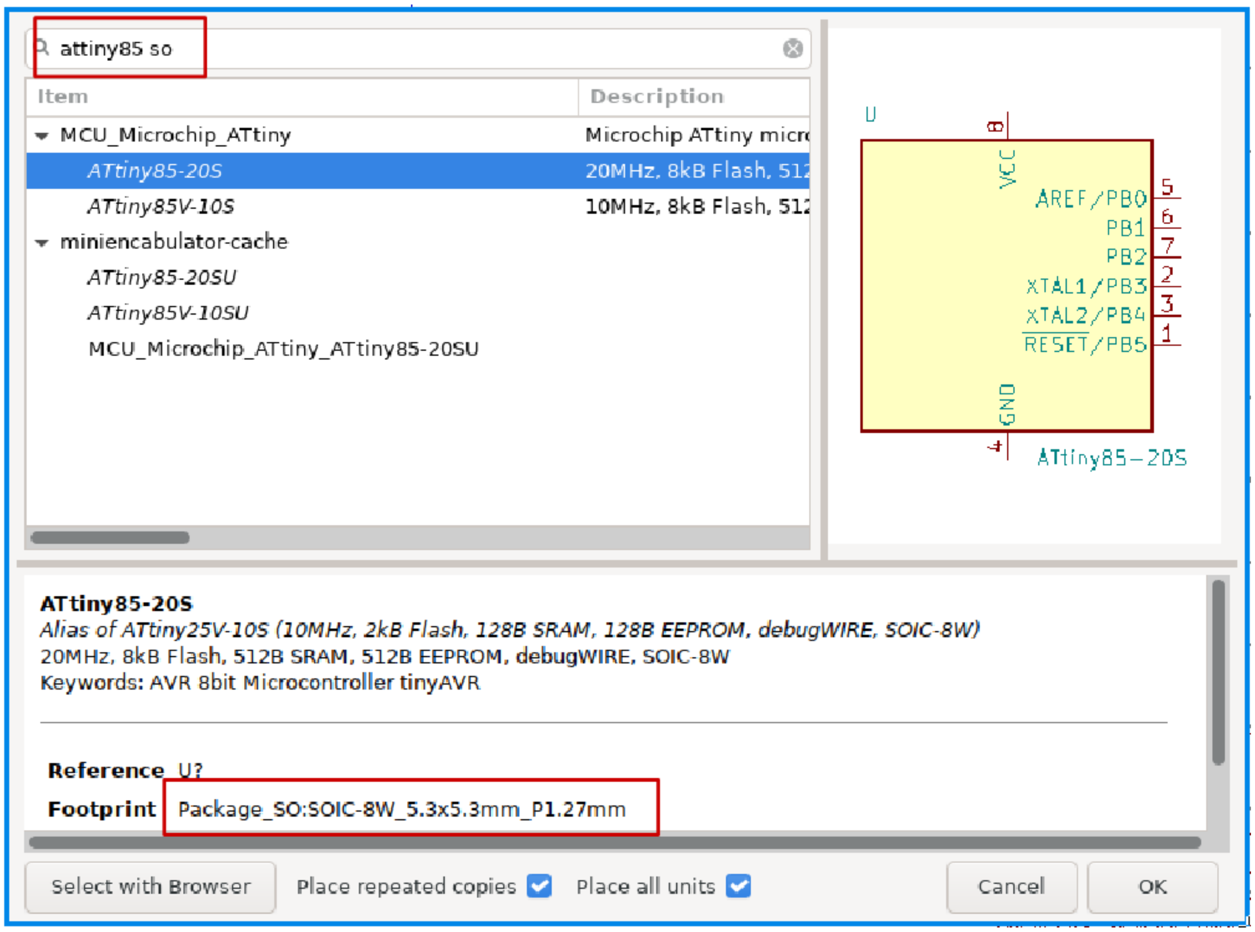
- Add components and define the electrical connections of your design in a schematic.
- Assign packages/footprints to the components in your schematic. (e.g. ATTiny85 can be purchased in a SO-8/SOIC-8 or DIP-8 package).
- Define PCB dimensions, component locations, and copper traces in a layout.

Schematic

Begin by installing KiCAD 5.1.10 or newer. Create a new project called `[netid]_miniencabulator` and open `[netid]_miniencabulator.sch`.

Microcontroller Setup

Hit `a` to open the component add dialog. Search "`attiny85 so`" for an `ATTiny85-20S` in an `SO` package and left click to place the part.



attiny85 so

Item	Description
MCU_Microchip_ATtiny	Microchip ATtiny micro
ATtiny85-20S	20MHz, 8kB Flash, 512
ATtiny85V-10S	10MHz, 8kB Flash, 512
miniencabulator-cache	
ATtiny85-20SU	
ATtiny85V-10SU	
MCU_Microchip_ATtiny_ATtiny85-20SU	

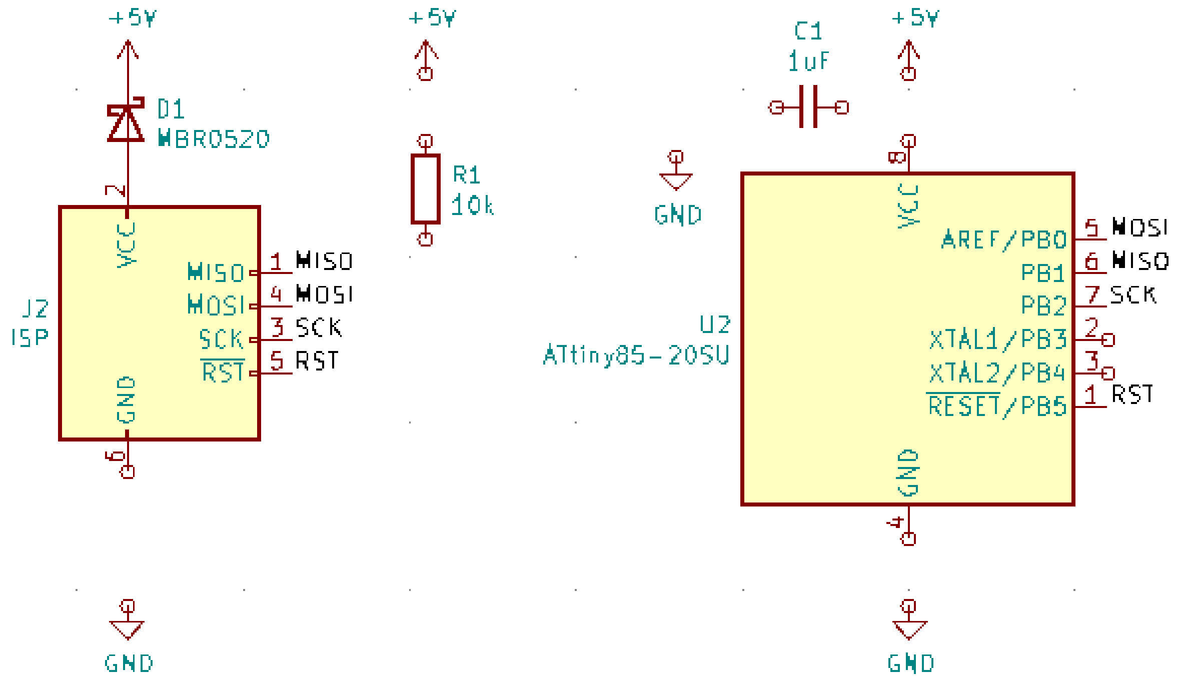
ATtiny85-20S
Alias of ATtiny25V-10S (10MHz, 2kB Flash, 128B SRAM, 128B EEPROM, debugWIRE, SOIC-8W)
20MHz, 8kB Flash, 512B SRAM, 512B EEPROM, debugWIRE, SOIC-8W
Keywords: AVR 8bit Microcontroller tinyAVR

Reference U?
Footprint Package_SO:SOIC-8W_5.3x5.3mm_P1.27mm

Add these other components and arrange like in the picture below:

- `AVR-ISP-6` - programming header.
- `MBR0520` - programmer reverse current protection.
- `+5V` - generic 5V power symbol
- `GND` - generic ground power symbol
- `R` - generic resistor
- `C_small` - generic capacitor

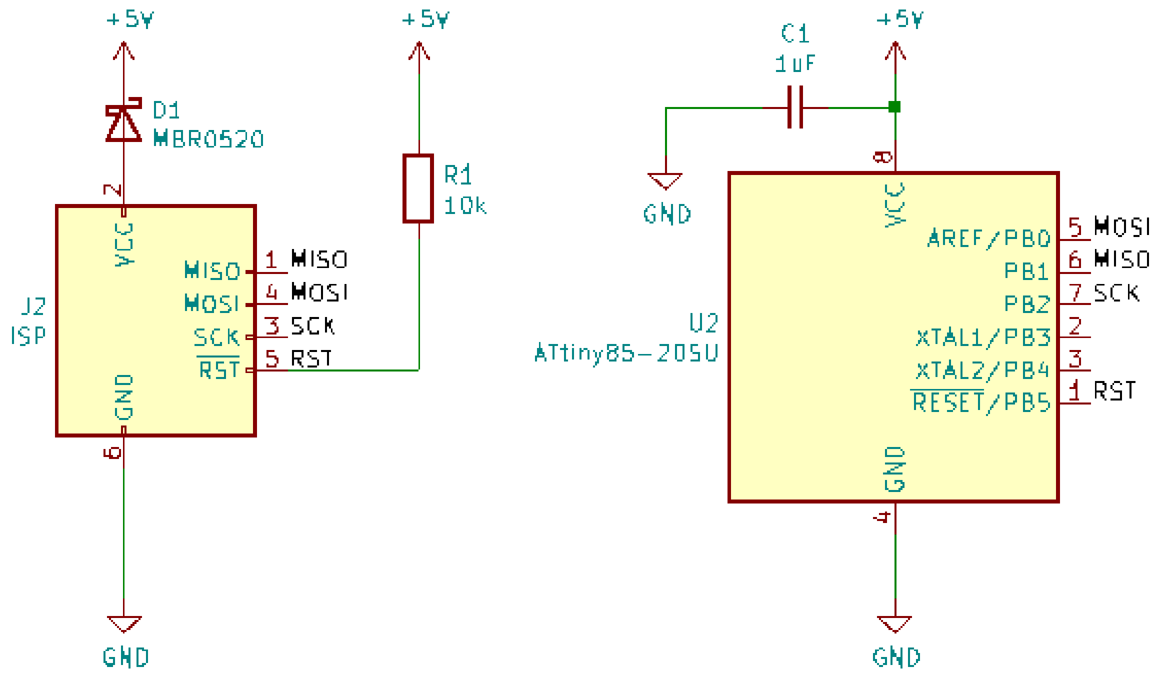
Move components around by hovering over them with the mouse and pressing `g` (grab). Components can be copied with `c`.



Begin connecting components as in the picture below. Use `w` to start routing a new wire under the mouse, left-click to finish routing, and `Esc` to cancel.

To reduce clutter, we will use labels to connect the ISP header to the microcontroller. Use `l` to place labels on pins or wires. Any two points with the same label will behave as if connected by a wire.

Set the value of the resistor to `10k` and the capacitor to `1uF` by hovering over each and pressing `v`.

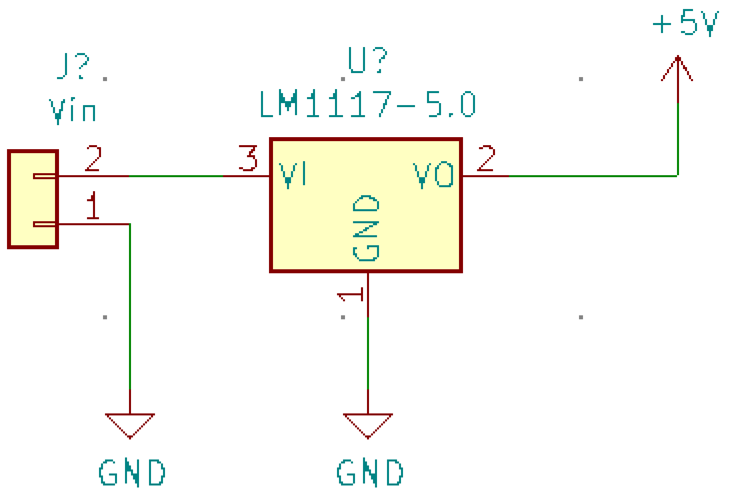


This is the minimum circuitry needed to program an ATTiny85 with an external programmer.

Power

Add these components to a blank area of the schematic and arrange as in the picture below:

- `Conn_01x02` (search "conn gen") - power supply connector
- `LM1117-5.0` - voltage regulator
- `+5V` - generic 5V power symbol
- `GND` - generic ground power symbol



Set the value of the connector to "`Vin`" using `v`. When we layout the PCB later, we will tell KiCAD to display this text on the board silkscreen next to the connector.

The convention to follow when placing components in a block is for power/signal to flow from left to right.

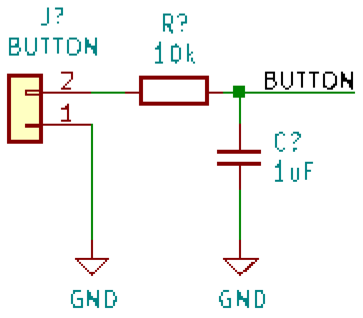
Peripherals

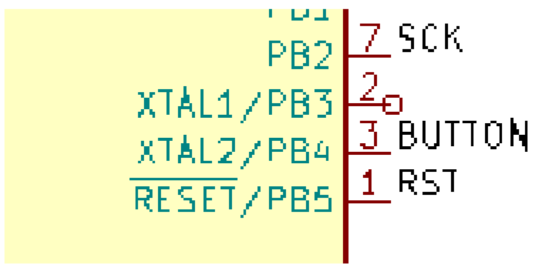
It's now time to add the peripheral electronics to our encabulator. We will connect the button, LED, and potentiometer to the microcontroller.

It's worth mentioning button bounce, which is a problem inherent in all mechanical buttons where the button makes intermittent contact before finally closing. This can be a problem for software which looks for a voltage transition on an IO pin to trigger some action, as there may be multiple transitions when the user presses the button only once. Filtering out these spurious transitions is known as **debouncing**, and can be implemented in hardware or software.

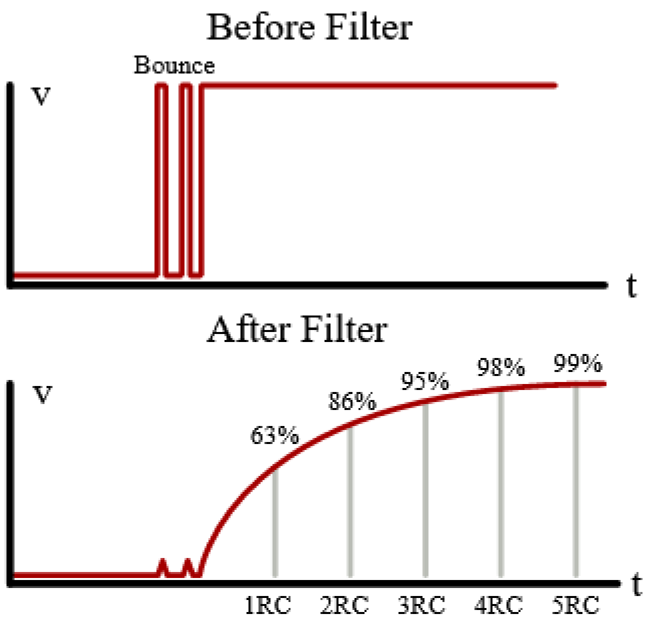
We will add a connector for our button with a simple hardware debouncer. Add these components:

- `Conn_01x02` (search "conn gen") - button connector
- `R` - RC filter resistor
- `C` - RC filter capacitor





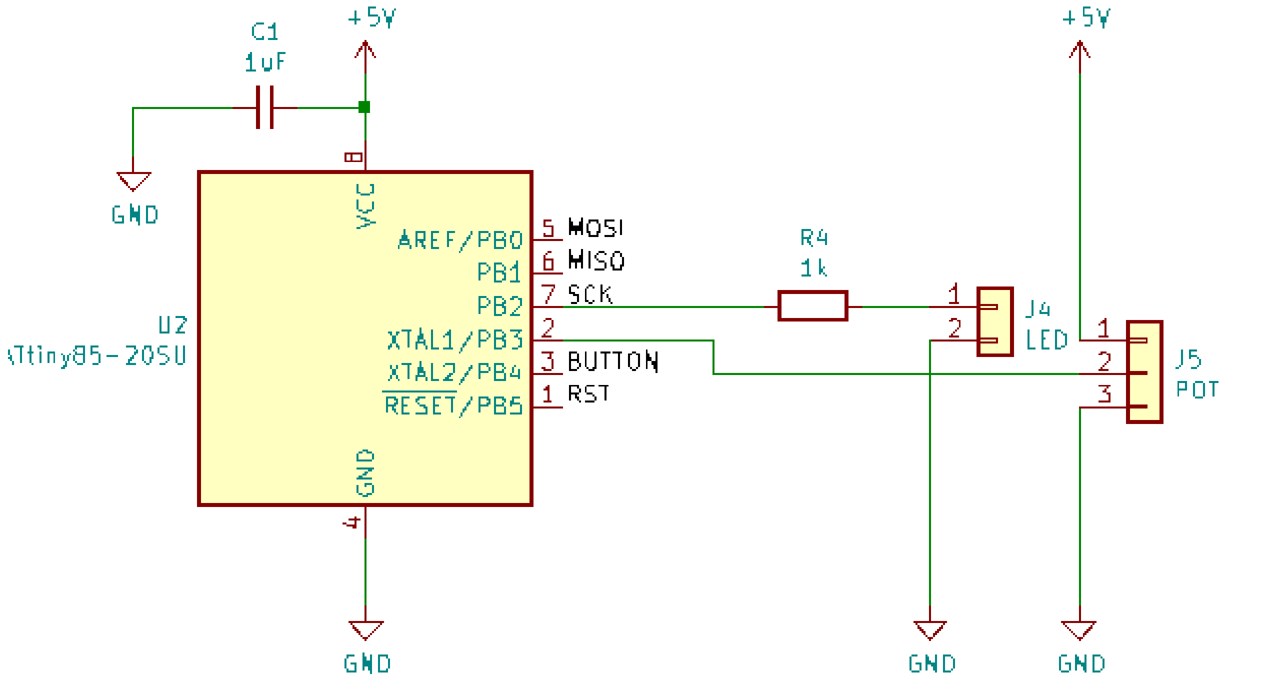
Change the value of the connector to "BUTTON" and place "BUTTON" labels on the output of the filter and input to the microcontroller. Values of 10k and 1uF will give a time constant of about 10ms, which should be long enough to filter out bounces. See the following image.



Next, add these components to connect the potentiometer and LED.

- `Conn_01x03` (search "conn gen") - potentiometer connector
- `Conn_01x02` (search "conn gen") - LED connector
- `R` - current limiting resistor

A value of 1k should suffice for the LED current limiting. It's important that the potentiometer be connected to an ADC pin of the ATTiny85.

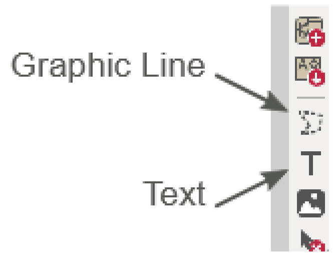


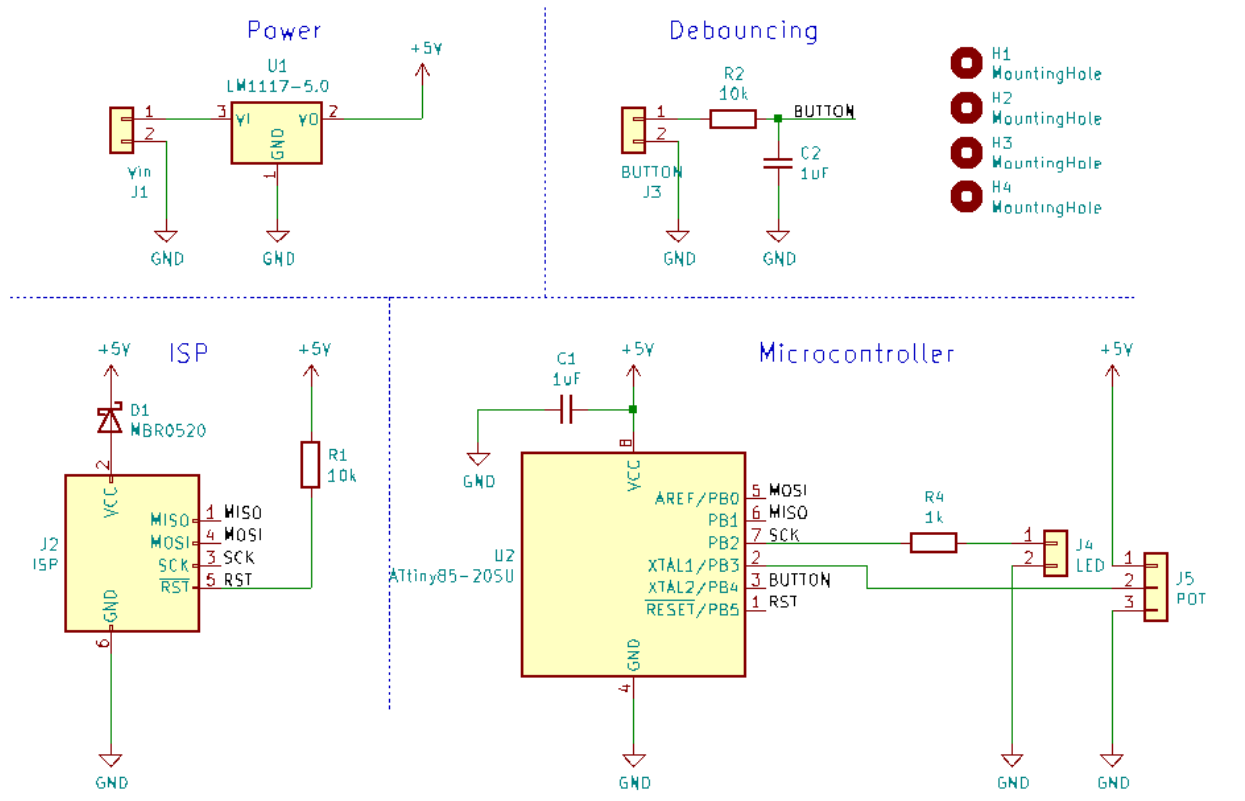
Finalizing the Schematic

Before moving on to the next stage, we will clean up and document our schematic. Since screenshots of your schematics will go into your design document, you should put some effort into keeping them neat and easy to read. Use the graphic line tool to divide up your schematic. Use the text tool to label each of these blocks.

Additionally, add four mounting holes:

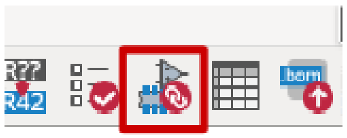
- `MountingHole` - hole for mounting in enclosure




Power
U1
LM1117-5.0
VI
VO
GND
+5V
Vin
J1
GND
Debouncing
R2
10k
BUTTON
C2
1uF
BUTTON
J3
GND
H1 MountingHole
H2 MountingHole
H3 MountingHole
H4 MountingHole
ISP
D1
MBR0520
R1
10k
J2
ISP
VCC
MISO
MOSI
SCK
RST
MISO
MOSI
SCK
RST
Microcontroller
C1
1uF
U2
ATtiny85-20SU
VCC
AREF/PB0
PB1
PB2
XTAL1/PB3
XTAL2/PB4
RESET/PB5
MOSI
MISO
SCK
BUTTON
RST
R4
1k
J4
LED
J5
POT
+5V
GND

Component Association

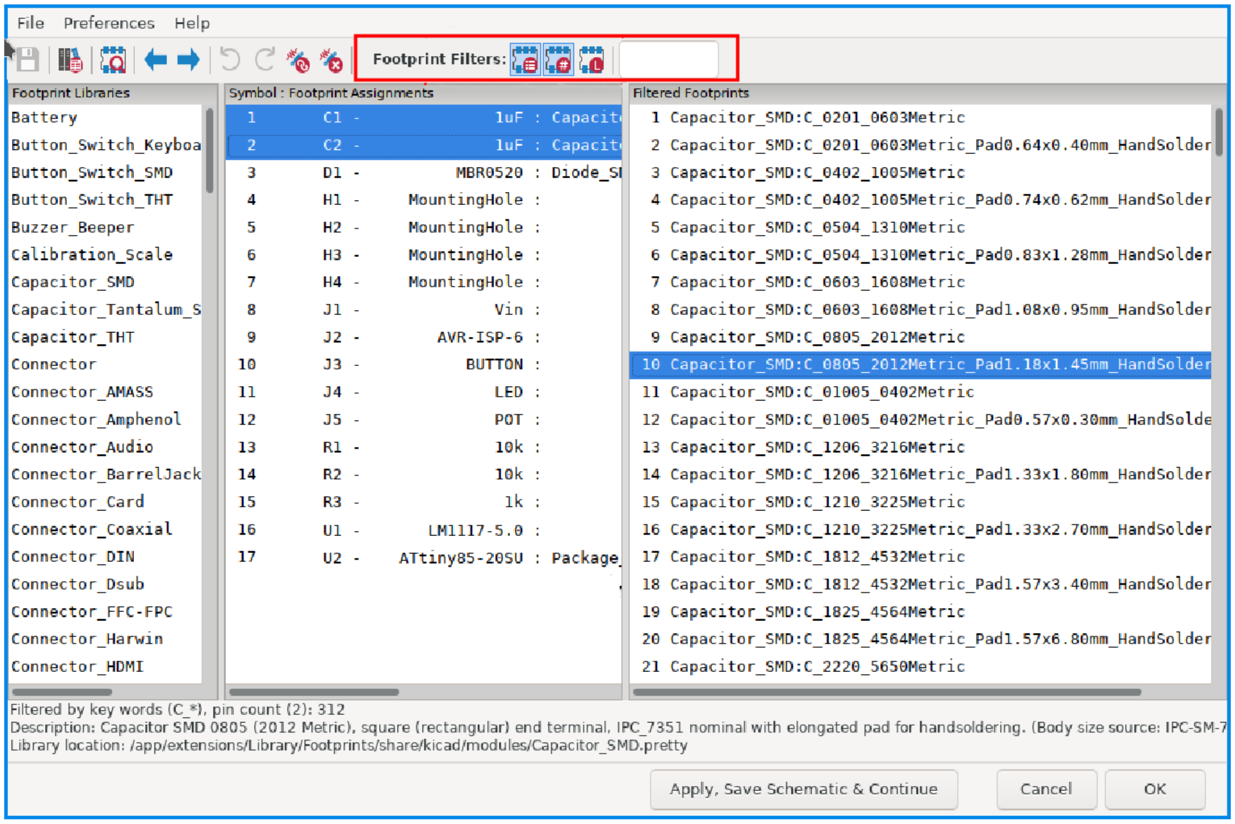
It's time to tell KiCAD what footprints to use for parts. Click the "Run footprint assignment tool" button on the top toolbar:



KiCAD sees that our schematic contains unannotated symbols ("R?", "C?", "U?", etc), and asks how we want to number them. The default options are fine here. Any time you add new symbols to your schematic, KiCAD will bring up this dialog again.

After clicking "Annotate", KiCAD will bring up the footprint association window, containing three panes. The left pane contains the categories of footprints, the middle pane contains your parts that need footprints assigned to them, and the right pane contains available footprints based on your filter options.

We'll start with the capacitors. Hold `Ctrl` and select both capacitors in the middle pane. We have 0805 sized capacitors that we will solder by hand, so double-click `Capacitor_SMD:C_0805_..._HandSolder`. Pay attention to the filter options at the top of the window.



Footprint Libraries
```
Battery
Button_Switch_Keyboa
Button_Switch_SMD
Button_Switch_THT
Buzzer_Beeper
Calibration_Scale
Capacitor_SMD
Capacitor_Tantalum_S
Capacitor_THT
Connector
Connector_AMASS
Connector_Amphenol
Connector_Audio
Connector_BarrelJack
Connector_Card
Connector_Coaxial
Connector_DIN
Connector_Dsub
Connector_FFC-FPC
Connector_Harwin
Connector_HDMI
```

Symbol : Footprint Assignments
```
 1    C1 -          1uF  : Capacit
 2    C2 -          1uF  : Capacit
 3    D1 -      MBR0520  : Diode_SI
 4    H1 -  MountingHole :
 5    H2 -  MountingHole :
 6    H3 -  MountingHole :
 7    H4 -  MountingHole :
 8    J1 -           Vin :
 9    J2 -     AVR-ISP-6 :
10    J3 -        BUTTON :
11    J4 -           LED :
12    J5 -           POT :
13    R1 -           10k :
14    R2 -           10k :
15    R3 -            1k :
16    U1 -    LM1117-5.0 :
17    U2 - ATtiny85-20SU : Package_
```

Filtered Footprints
```
 1 Capacitor_SMD:C_0201_0603Metric
 2 Capacitor_SMD:C_0201_0603Metric_Pad0.64x0.40mm_HandSolder
 3 Capacitor_SMD:C_0402_1005Metric
 4 Capacitor_SMD:C_0402_1005Metric_Pad0.74x0.62mm_HandSolder
 5 Capacitor_SMD:C_0504_1310Metric
 6 Capacitor_SMD:C_0504_1310Metric_Pad0.83x1.28mm_HandSolder
 7 Capacitor_SMD:C_0603_1608Metric
 8 Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder
 9 Capacitor_SMD:C_0805_2012Metric
10 Capacitor_SMD:C_0805_2012Metric_Pad1.18x1.45mm_HandSolder
11 Capacitor_SMD:C_01005_0402Metric
12 Capacitor_SMD:C_01005_0402Metric_Pad0.57x0.30mm_HandSolde
13 Capacitor_SMD:C_1206_3216Metric
14 Capacitor_SMD:C_1206_3216Metric_Pad1.33x1.80mm_HandSolder
15 Capacitor_SMD:C_1210_3225Metric
16 Capacitor_SMD:C_1210_3225Metric_Pad1.33x2.70mm_HandSolder
17 Capacitor_SMD:C_1812_4532Metric
18 Capacitor_SMD:C_1812_4532Metric_Pad1.57x3.40mm_HandSolder
19 Capacitor_SMD:C_1825_4564Metric
20 Capacitor_SMD:C_1825_4564Metric_Pad1.57x6.80mm_HandSolder
21 Capacitor_SMD:C_2220_5650Metric
```

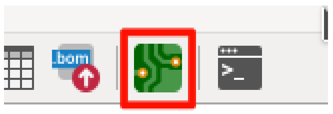
```
Filtered by key words (C_*), pin count (2): 312
Description: Capacitor SMD 0805 (2012 Metric), square (rectangular) end terminal, IPC_7351 nominal with elongated pad for handsoldering. (Body size source: IPC-SM-7
Library location: /app/extensions/Library/Footprints/share/kicad/modules/Capacitor_SMD.pretty
```

Associate the remaining parts with these footprints:

- resistors (R*) - `Resistor_SMD:R_0805_...HandSolder`
- mounting holes (H*) - `MountingHole:MountingHole_3.2mm_M3` (for M3 screws)
- 2 pin connectors (Vin, BUTTON, LED) - `Connector_Molex:KK-254...1x02...`
- 3 pin connectors (POT) - `Connector_Molex:KK-254...1x03...`
- program header (AVR-ISP-6) - `Connector_IDC:IDC-Header_2x03...Vertical`
- regulator (LM1117-5.0) - `Package_TO_SOT_SMD:SOT-223`

Note that `Package_TO_SOT_SMD:SOT-223` has 4 pins, but the `LM1117` symbol only has 3. You must uncheck the pin count filter for it to show up.

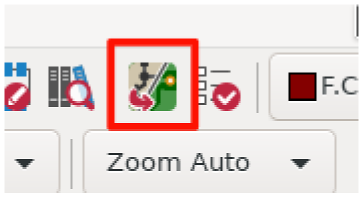
Footprints are now associated with our symbols. Click the "Open PCB in board editor" button to start working on the layout.

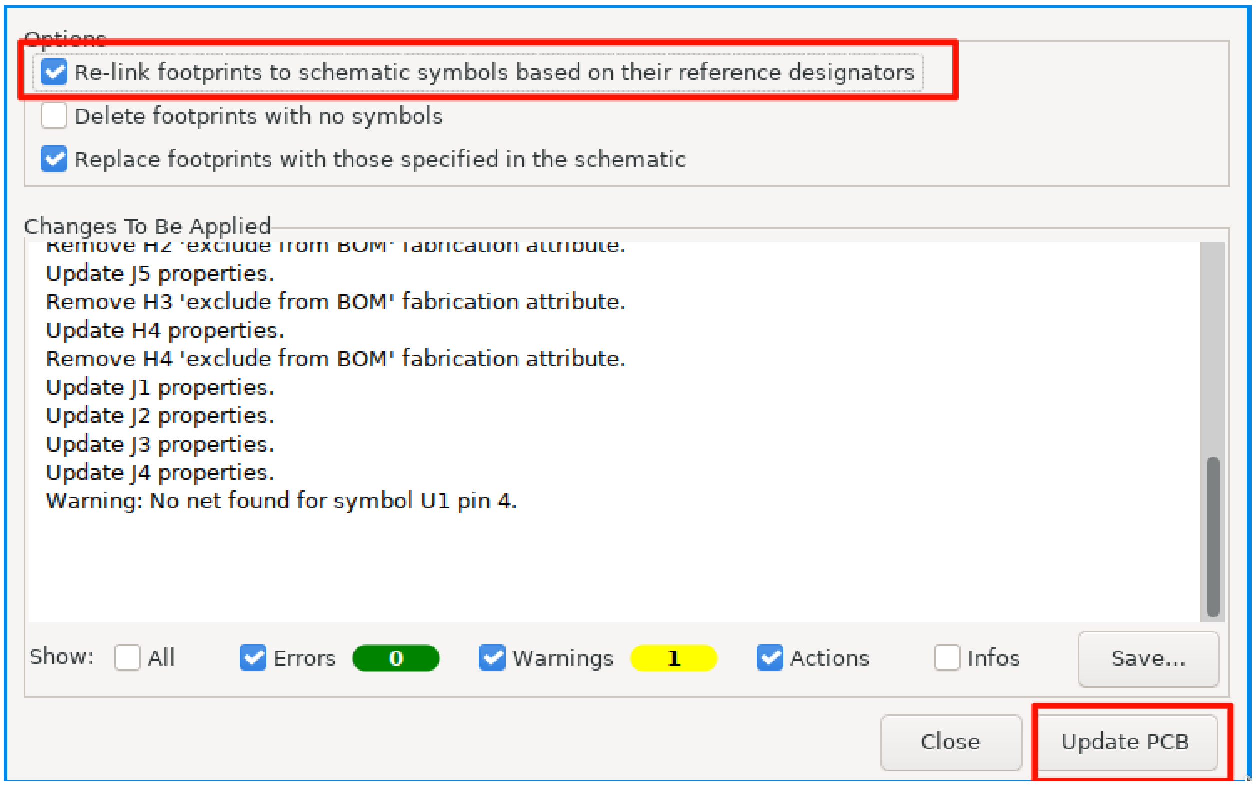


Layout

Component Placement

It's time to start placing components on our board. Start by clicking the "Update PCB with changes made to schematic" button and place the footprints somewhere in the middle of the page. If you make changes to your schematic later on, you should use this button.





Options

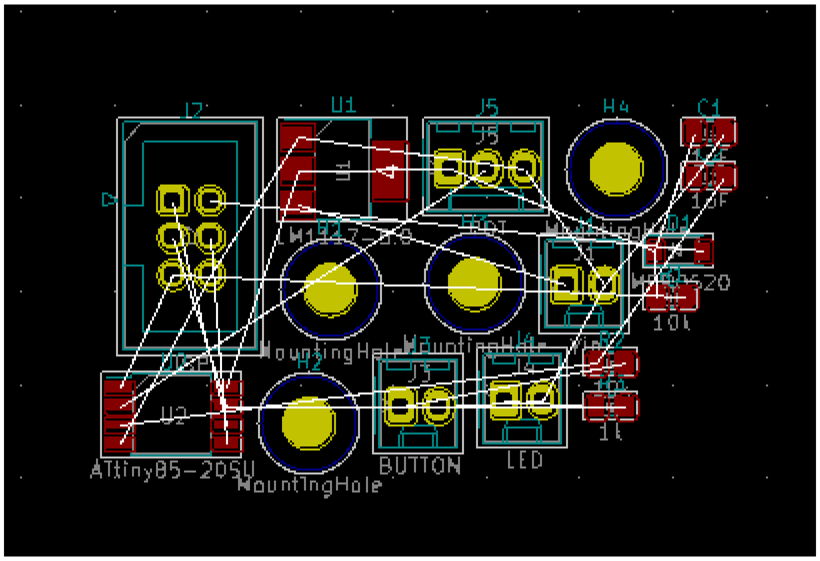
- [x] Re-link footprints to schematic symbols based on their reference designators
- [ ] Delete footprints with no symbols
- [x] Replace footprints with those specified in the schematic

Changes To Be Applied

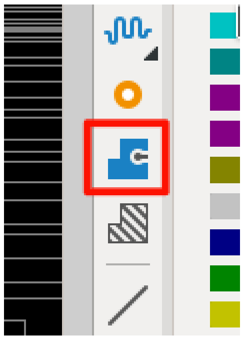
```
Remove H2 'exclude from BOM' fabrication attribute.
Update J5 properties.
Remove H3 'exclude from BOM' fabrication attribute.
Update H4 properties.
Remove H4 'exclude from BOM' fabrication attribute.
Update J1 properties.
Update J2 properties.
Update J3 properties.
Update J4 properties.
Warning: No net found for symbol U1 pin 4.
```

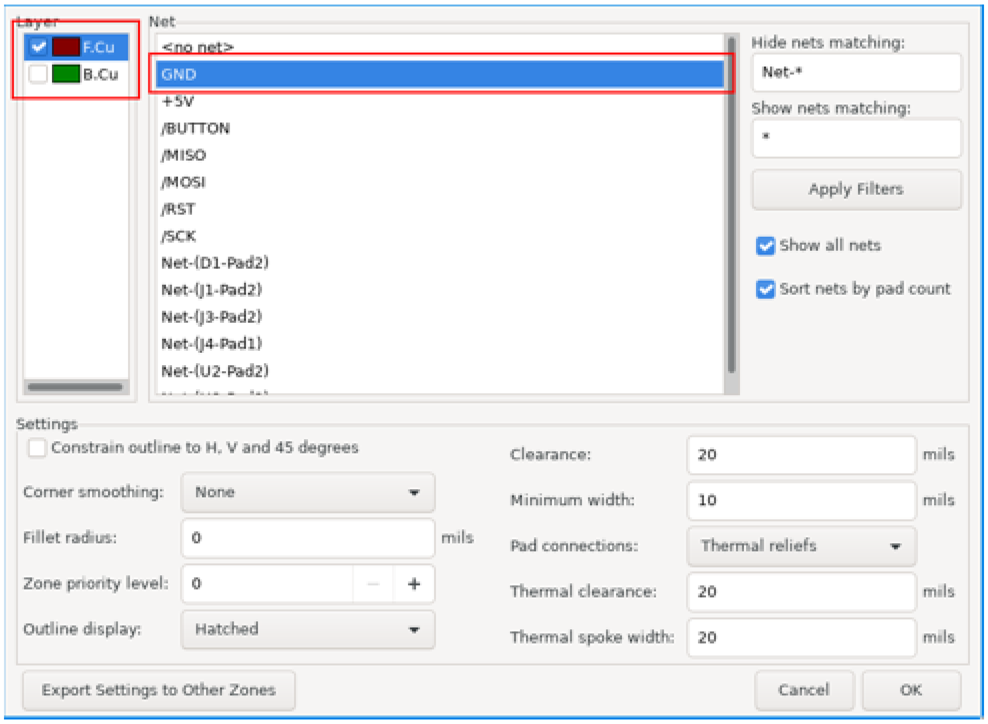
Show: [ ] All [x] Errors 0 [x] Warnings 1 [x] Actions [ ] Infos

The white lines are called **unrouted nets** (also known as "airwires") and indicate what component pads should be electrically connected.



To make things a bit easier, we will add a ground plane to the front and back side of our board that will automatically connect all GND nets together for us. Any areas of filled-in copper like this are known as **zones** in KiCAD.





Layer
- [✓] F.Cu
- [ ] B.Cu

Net
- <no net>
- GND
- +5V
- /BUTTON
- /MISO
- /MOSI
- /RST
- /SCK
- Net-(D1-Pad2)
- Net-(J1-Pad2)
- Net-(J3-Pad2)
- Net-(J4-Pad1)
- Net-(U2-Pad2)

Hide nets matching: Net-*

Show nets matching: *

Apply Filters

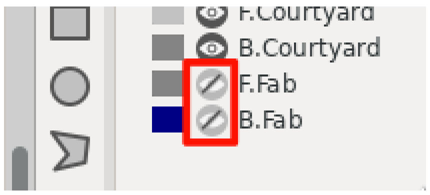
- [✓] Show all nets
- [✓] Sort nets by pad count

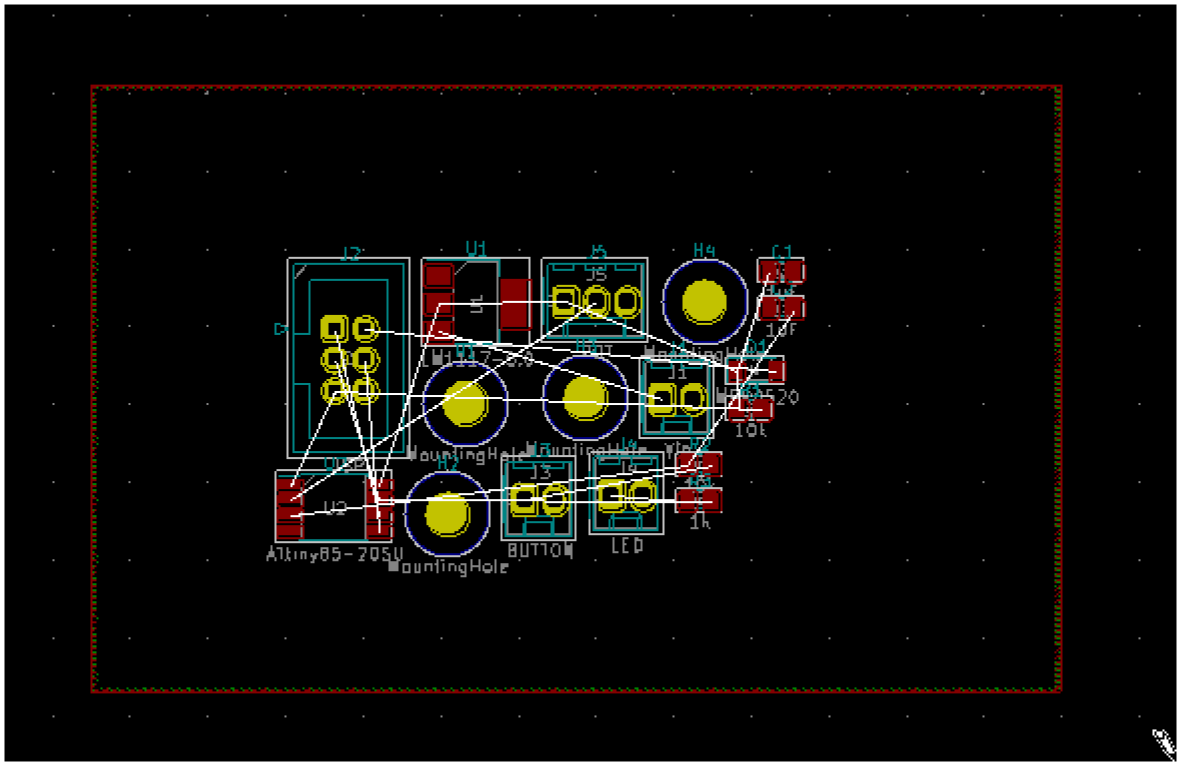
Settings
- [ ] Constrain outline to H, V and 45 degrees

Corner smoothing:	None	Clearance:	20 mils
Fillet radius:	0 mils	Minimum width:	10 mils
Zone priority level:	0	Pad connections:	Thermal reliefs
Outline display:	Hatched	Thermal clearance:	20 mils
		Thermal spoke width:	20 mils

Export Settings to Other Zones | Cancel | OK

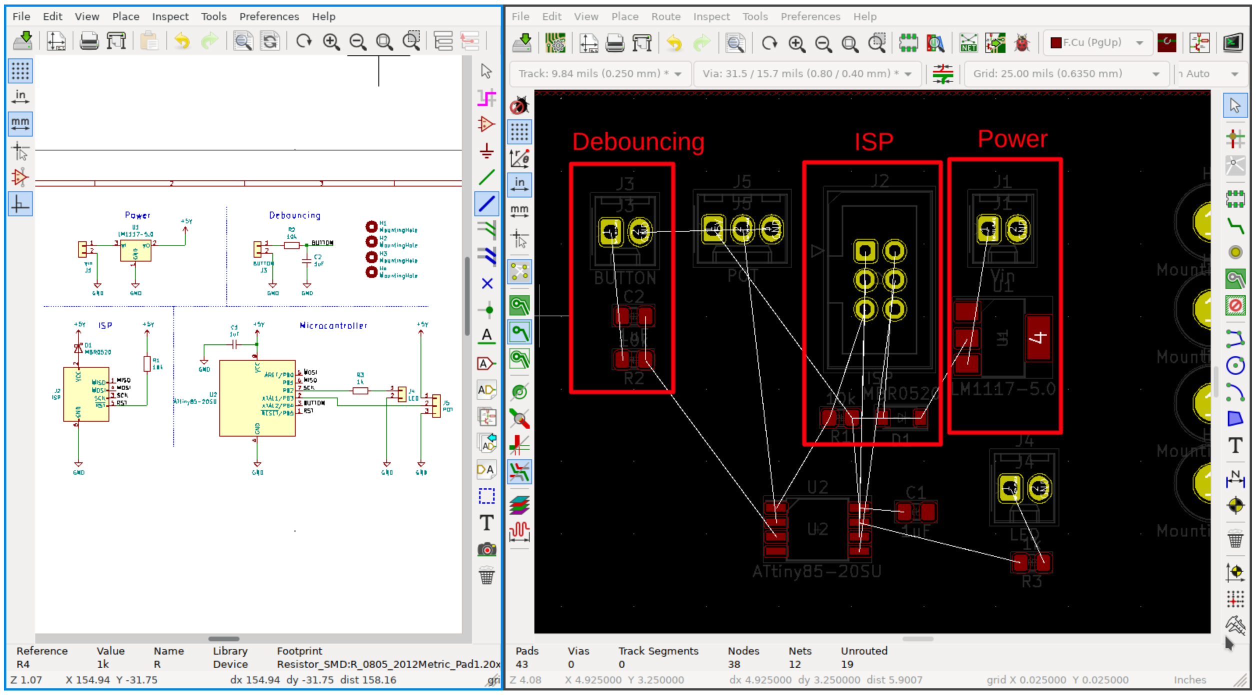
You can also turn off the `F.Fab` and `B.Fab` layers in the right toolbar to hide some unnecessary text:



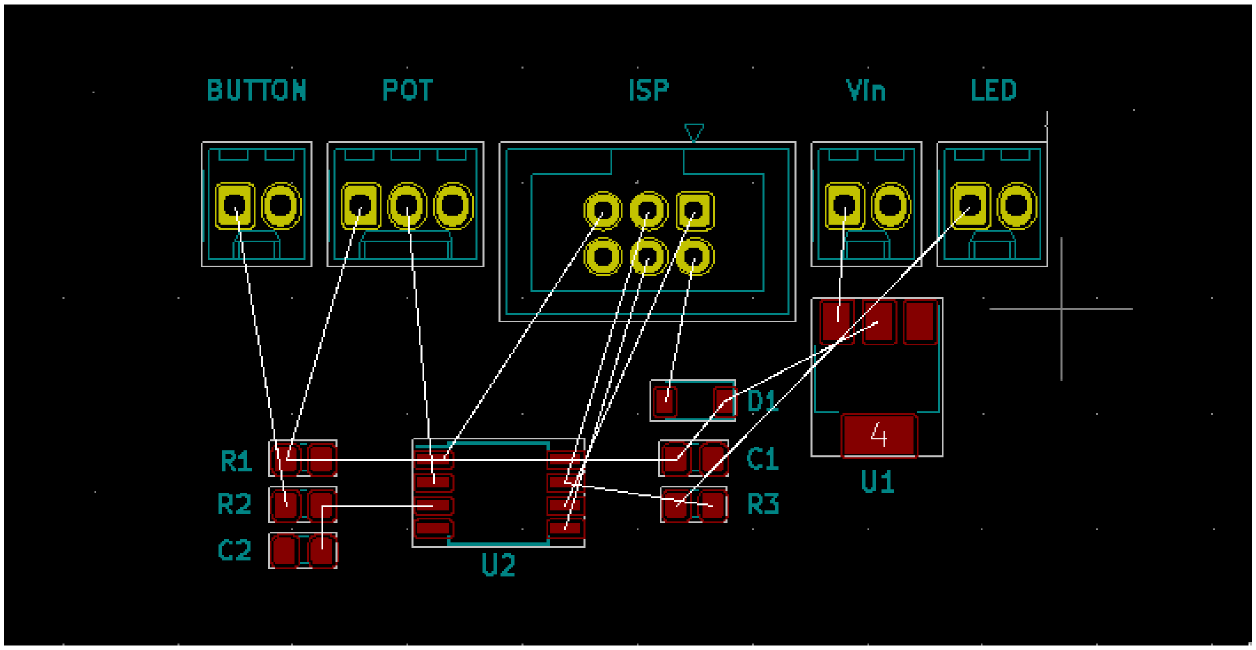


Now we can begin sorting the footprints. This is generally where most of the time designing a PCB should be spent. The goal here is to untangle the footprints so that airwires cross as little as possible. This makes routing in the next section simpler and helps to keep traces short.

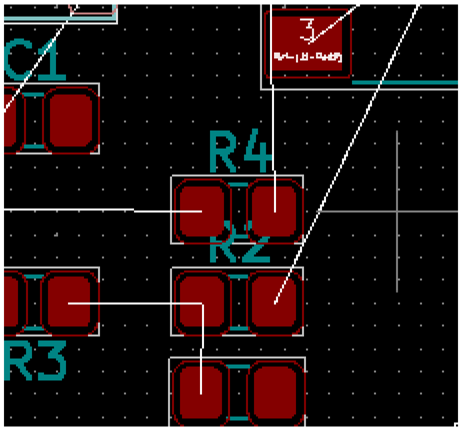
When working on new boards, a good way to start is to first group the footprints together the same way they are grouped in the schematic.



To save some time, below is a suggested layout for your board:



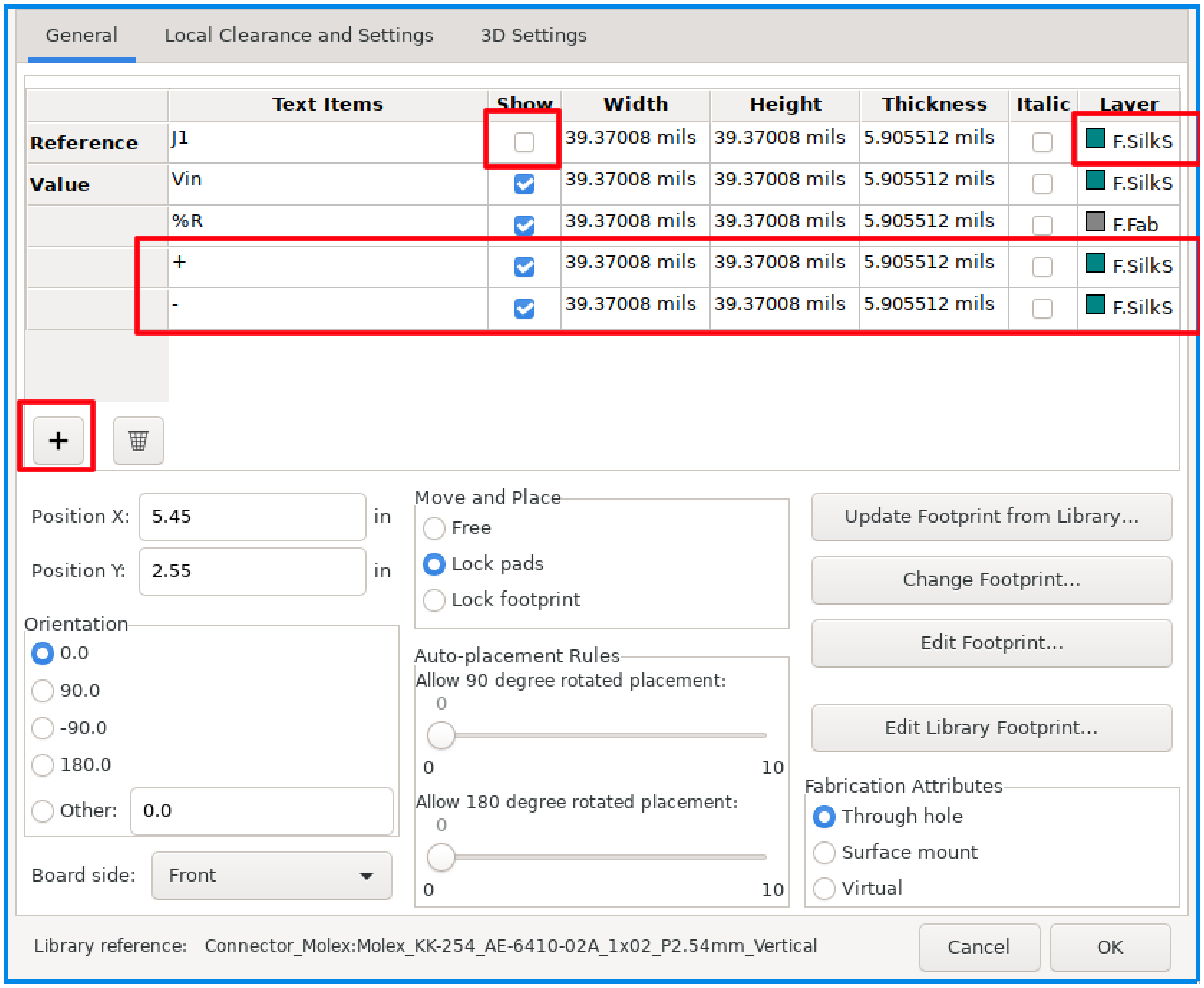
When you've placed all your components, make sure the component references are visible and not obscured by other footprints. You will use these references when populating the board.



Next, we will add some labeling so we know what the connectors are. Click on each connector and press `e` to bring up the footprint properties window. Hide the footprint reference and change the layer of the **value** field to `F.Silkscreen` so it shows up on the front side silkscreen.

For connectors which have polarity, you should label the individual pins. For the `Vin` and LED connectors, add 2 new properties + and - and set their layer to the front silkscreen.

(If your newly-added labels are hidden behind the footprint, you can choose `F.Silkscreen` in the layers pane to get them to show up.)



General | Local Clearance and Settings | 3D Settings

	Text Items	Show	Width	Height	Thickness	Italic	Layer
Reference	J1	☐	39.37008 mils	39.37008 mils	5.905512 mils	☐	🟩 F.SilkS
Value	Vin	✓	39.37008 mils	39.37008 mils	5.905512 mils	☐	🟩 F.SilkS
	%R	✓	39.37008 mils	39.37008 mils	5.905512 mils	☐	⬛ F.Fab
	+	✓	39.37008 mils	39.37008 mils	5.905512 mils	☐	🟩 F.SilkS
	-	✓	39.37008 mils	39.37008 mils	5.905512 mils	☐	🟩 F.SilkS

Position X: 5.45 in

Position Y: 2.55 in

Orientation: [✓ 0.0] 90.0 -90.0 180.0 Other: 0.0

Board side: Front

Move and Place: Free [✓ Lock pads] Lock footprint

Auto-placement Rules

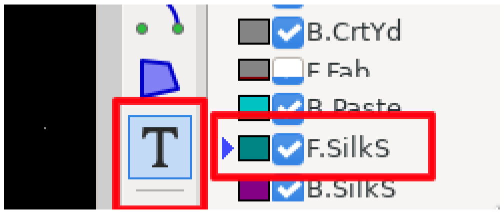
Allow 90 degree rotated placement: 0 (0–10)

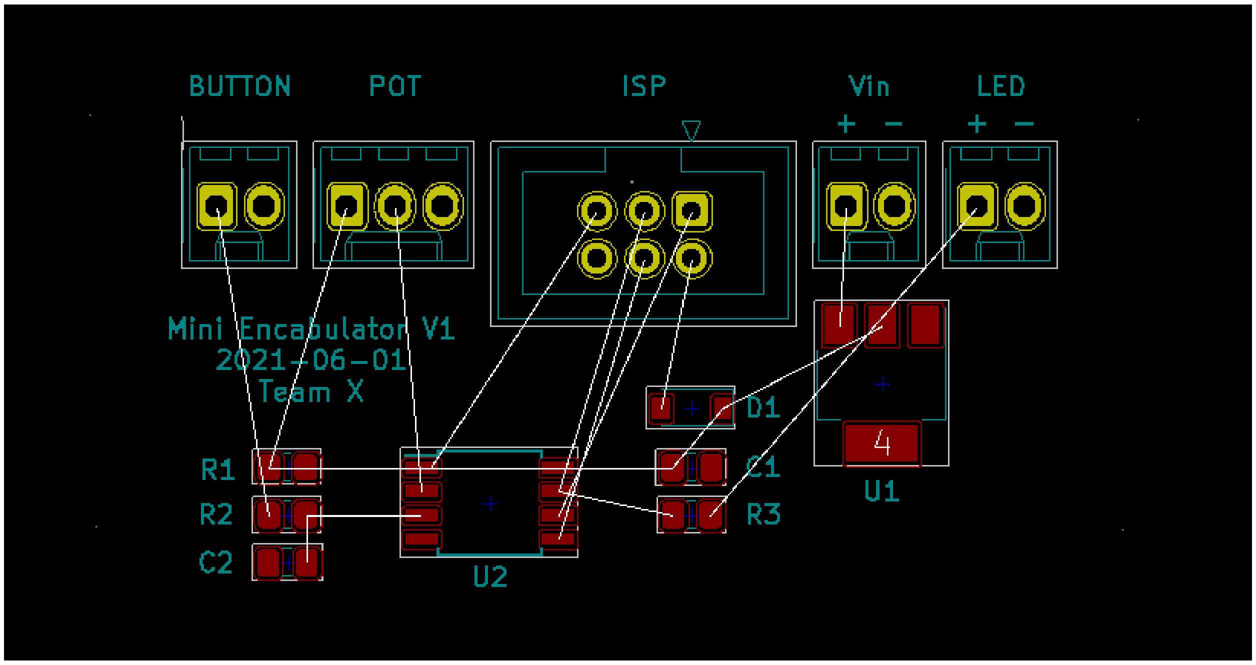
Allow 180 degree rotated placement: 0 (0–10)

Fabrication Attributes: [✓ Through hole] Surface mount Virtual

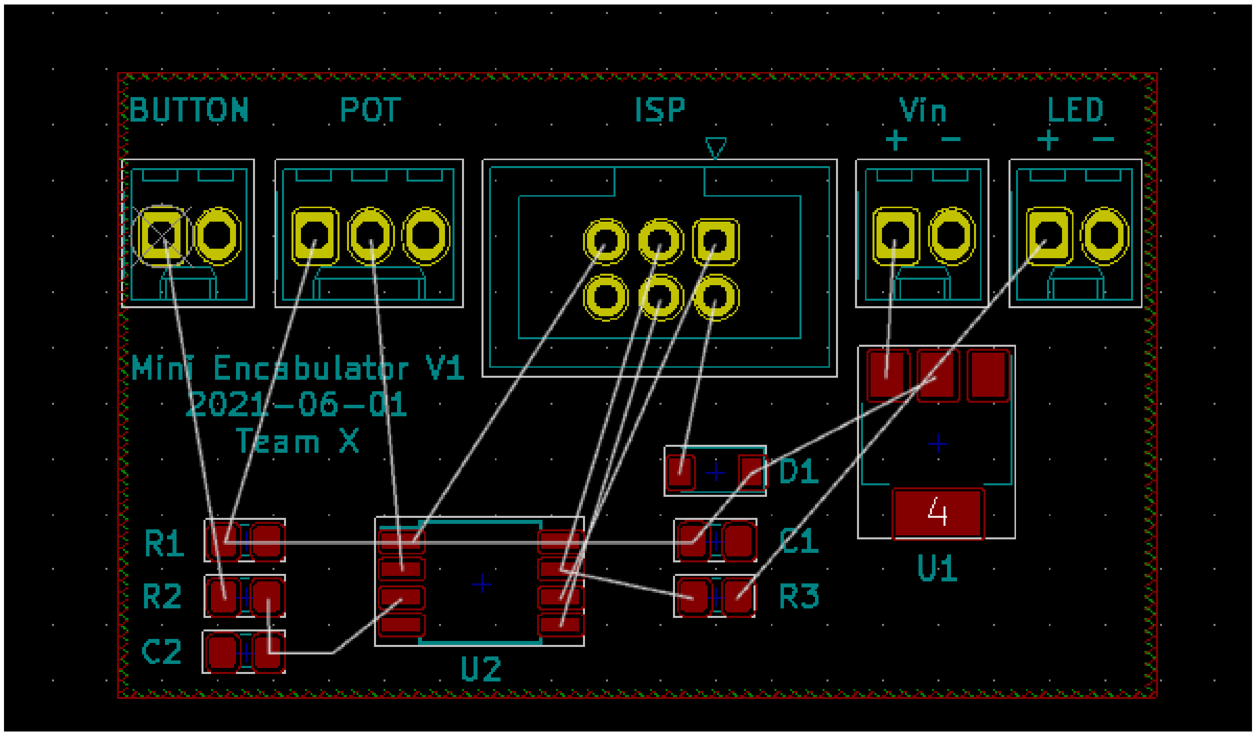
Library reference: Connector_Molex:Molex_KK-254_AE-6410-02A_1x02_P2.54mm_Vertical

Use the text tool to add information about your board to the silkscreen layer. Include the date, your team number, and the board version.



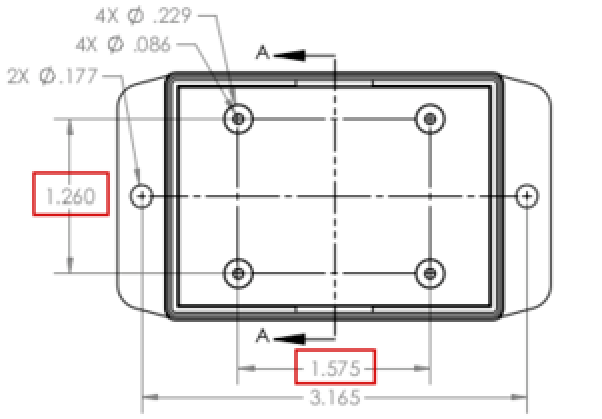


Then drag your front and back ground zones to surround the component area. It's a good idea to make these zones as small as possible to limit RF noise picked up from external sources.



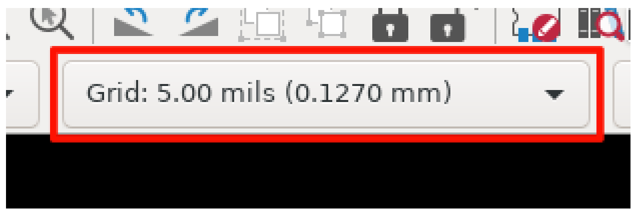
It's time to pick an enclosure. Generally you want to pick one which is slightly wider/longer than your PCB, has sufficient height for PCB connectors and panel mount components and comes with PCB mounting bosses. Check the enclosure datasheet.

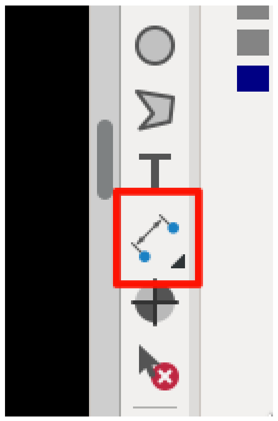
For example, the SN25 enclosure from Polycase is shown below:

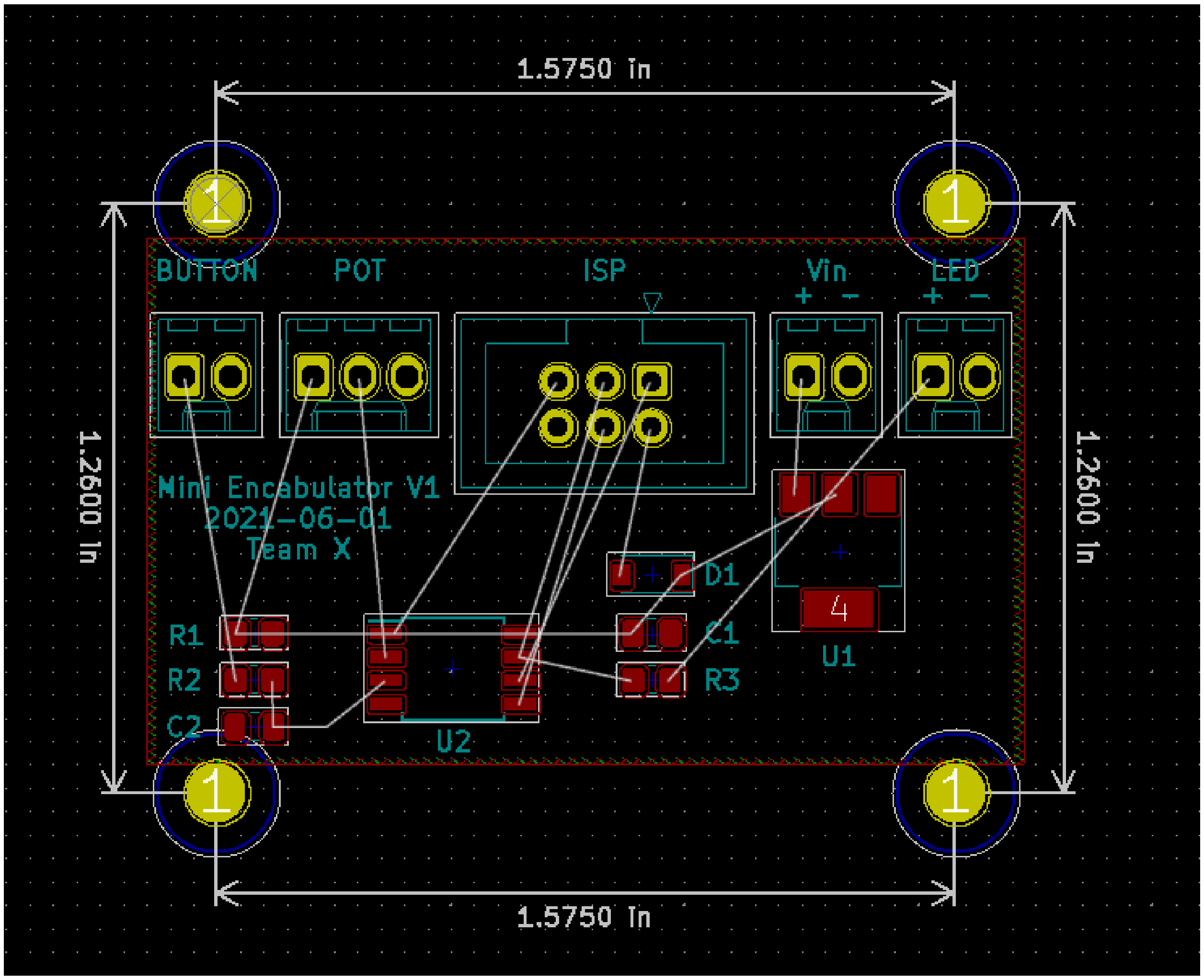


So the mounting bosses are separated by 1.26in vertically and 1.575in horizontally. Change the grid size to 5mils (0.005in) and move the mounting holes to the corners of the board. Use the dimensioning tool to verify that the separation between the mounting holes is correct. The dimensioning lines should go on the `User.Drawings` layer.

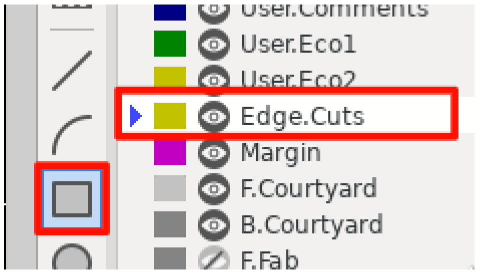
When your mouse is above the mounting hole, snapping crosshairs should appear. These dimensions must be exact to receive full credit.


Grid: 5.00 mils (0.1270 mm)

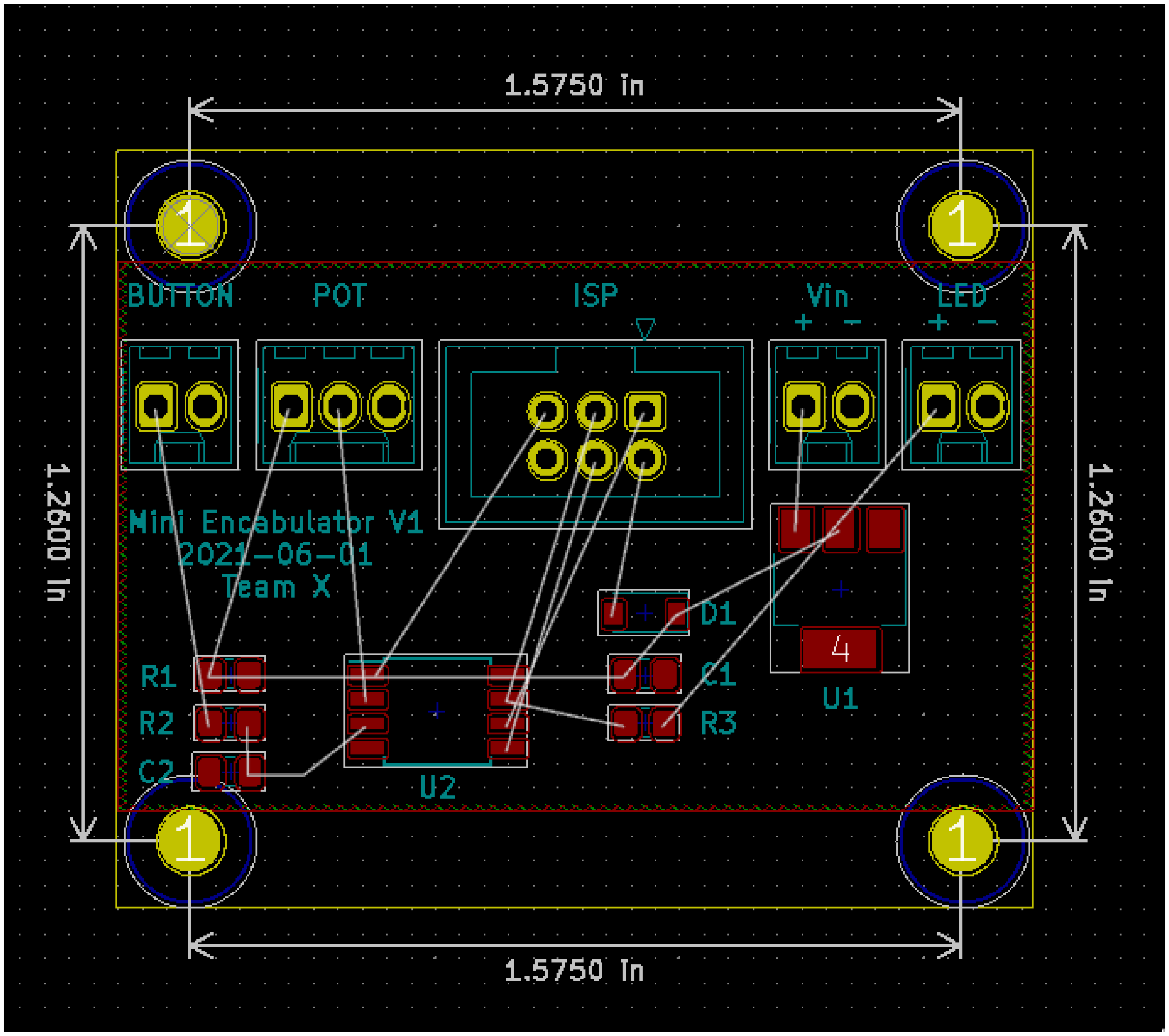




Finally, use the rectangle tool with the `Edge.Cuts` layer selected and draw a rectangle just outside of the mounting holes to define the edge of the board.



Below is a fully populated board ready for routing.



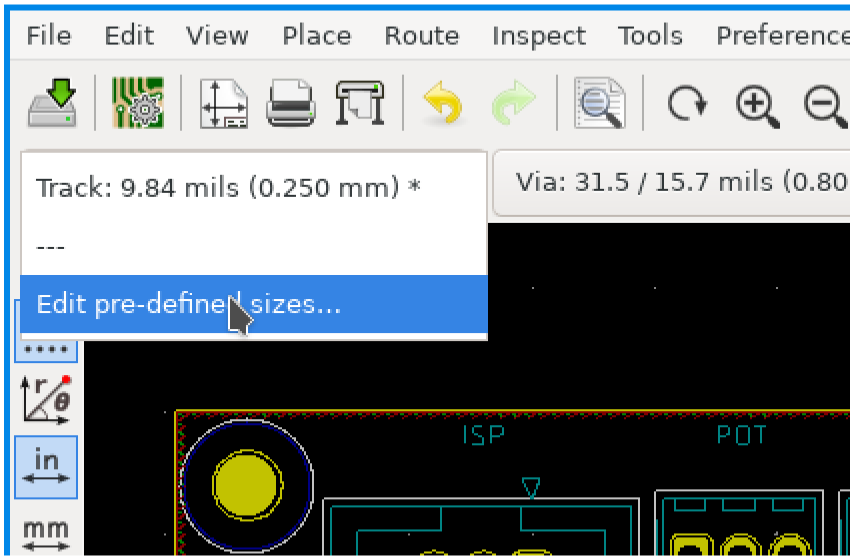
Routing

Keyboard shortcuts used in this section:

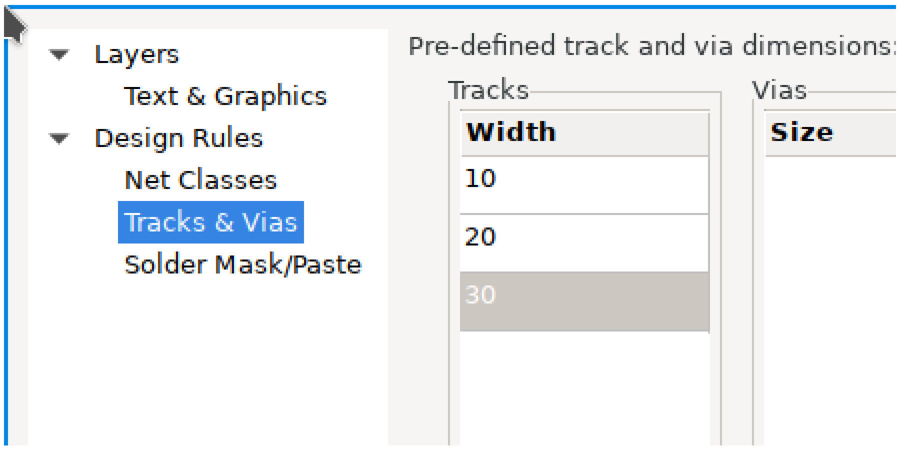
- x - Add track
- v - Toggle front/back side of board
- u - Select all segments of a track
- `Del` - Delete selected track segments
- b - redraw zones (after moving footprints or tracks)

It's time to start connecting our components together with tracks (also known as traces). KiCAD only comes with a single track width specified, so we will add more.

Click the dropdown `Track: ... > Edit pre-defined sizes...` and add track widths for 10, 20, and 30 mils. These larger tracks are useful if you need to carry more current.


File
Edit
View
Place
Route
Inspect
Tools
Preference
Track: 9.84 mils (0.250 mm) *

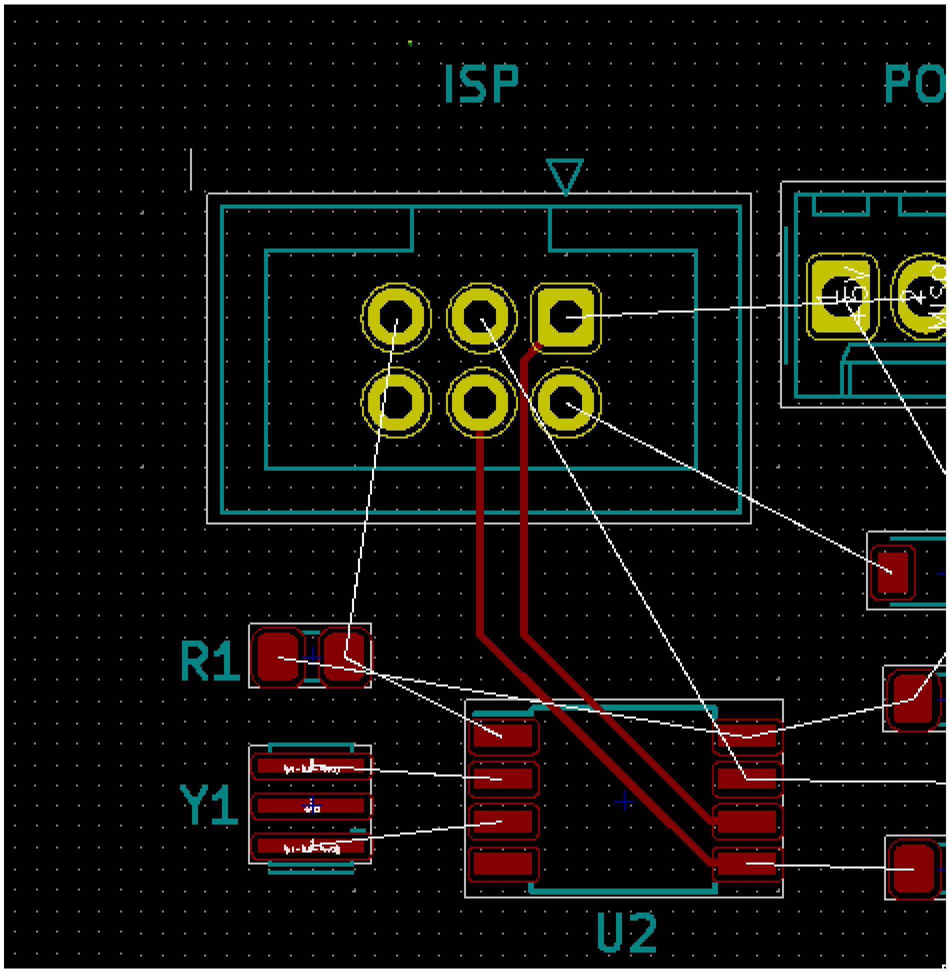
Edit pre-defined sizes...
Via: 31.5 / 15.7 mils (0.80
in
mm
ISP
POT



Width
10
20
30

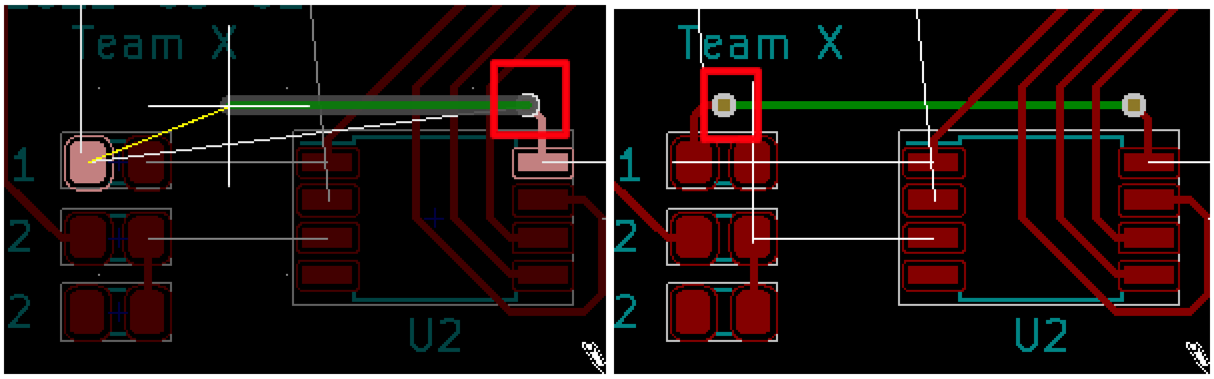
When doing routing, it's good to prioritize traces which are high-speed, high-power, or highly parallel over other traces like slow data lines or simple GPIO in order to keep them short.

We'll start by routing the tracks from the microcontroller to the ISP header. Choose the 10 mil track width from the dropdown menu, and press x to begin routing. Route the MOSI, MISO, SCK, and RST pads on the microcontroller to their respective pins on the programming header by left-clicking, as in the picture below.



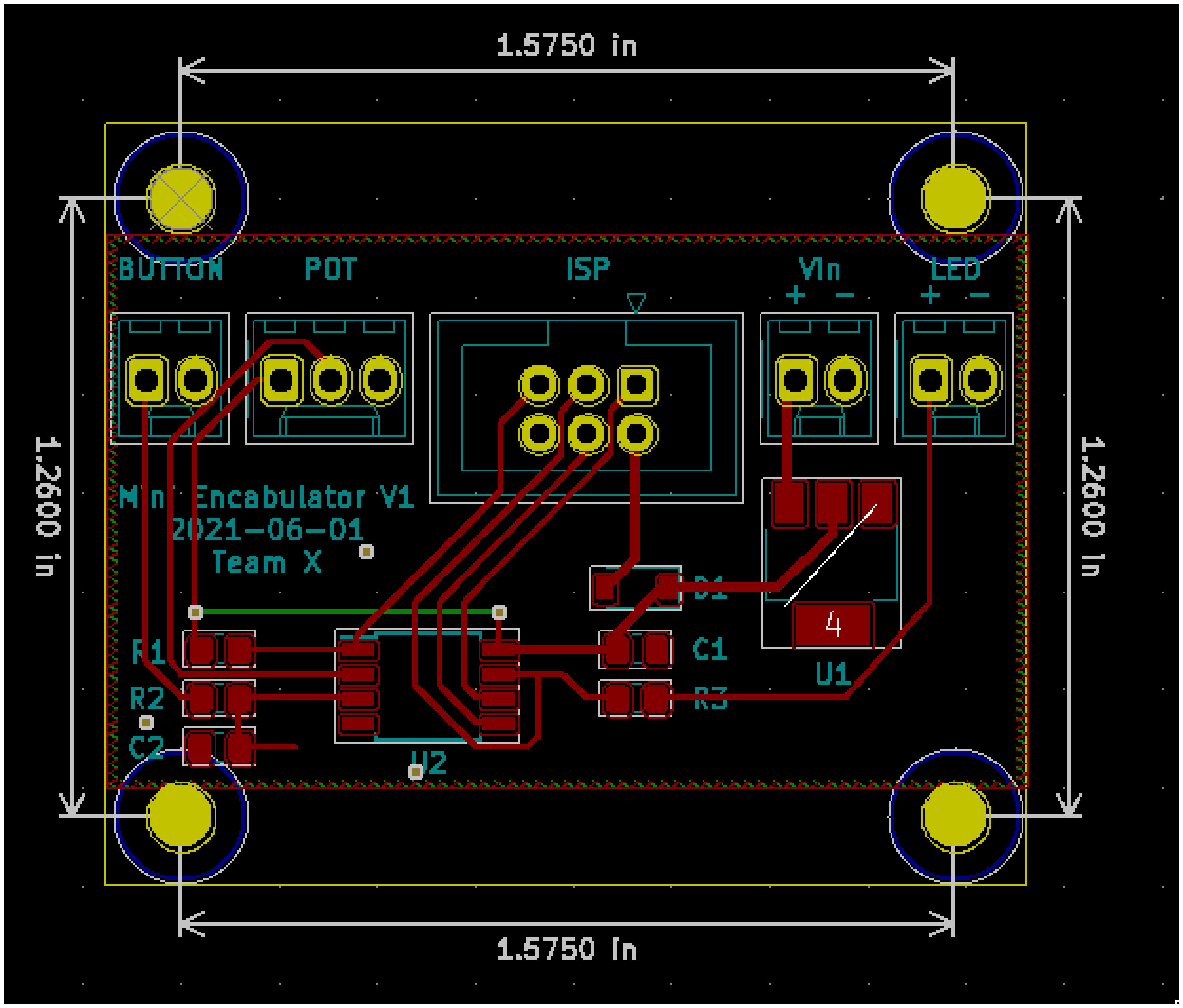
We'll use the back side of the board to connect the RST pullup resistor to 5V.

Start routing the +5V pin on the microcontroller, then hover over the position indicated in the first picture below and press v to switch to the back side of the board and left-click to place a via. Repeat this process in the position indicated in the second picture to come to the front side of the board and finish connecting the track.



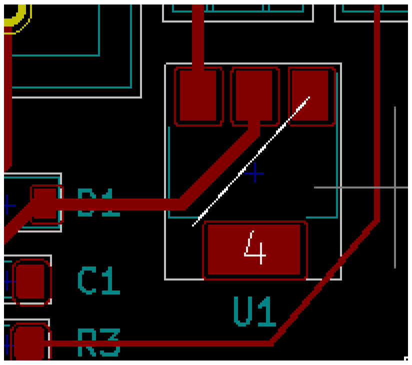
The red/green segments of this track are on the front/back sides of the board and are connected by a **via**.

Continue routing the rest of the tracks. Use 20 mil tracks when connecting Vin to the regulator, and the regulator to the microcontroller, as in the picture below.

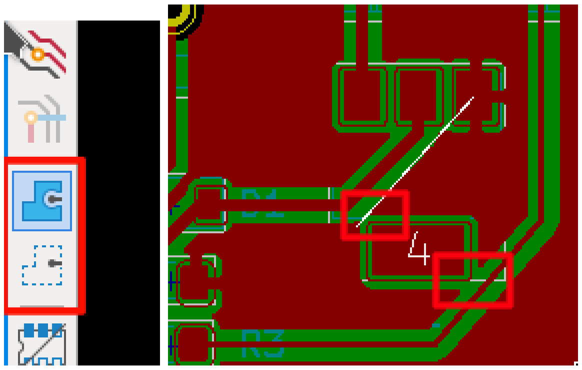


Copper Island and Ground Stitching

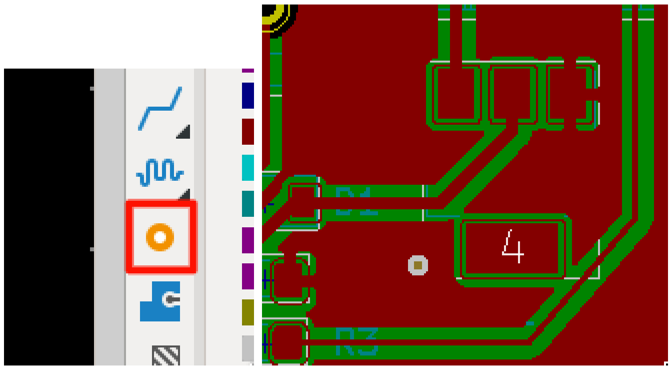
You may notice there is still a remaining airwire in the previous picture. This may or may not be present in your board depending on how you laid down your tracks. (continue reading even if you don't have this!)



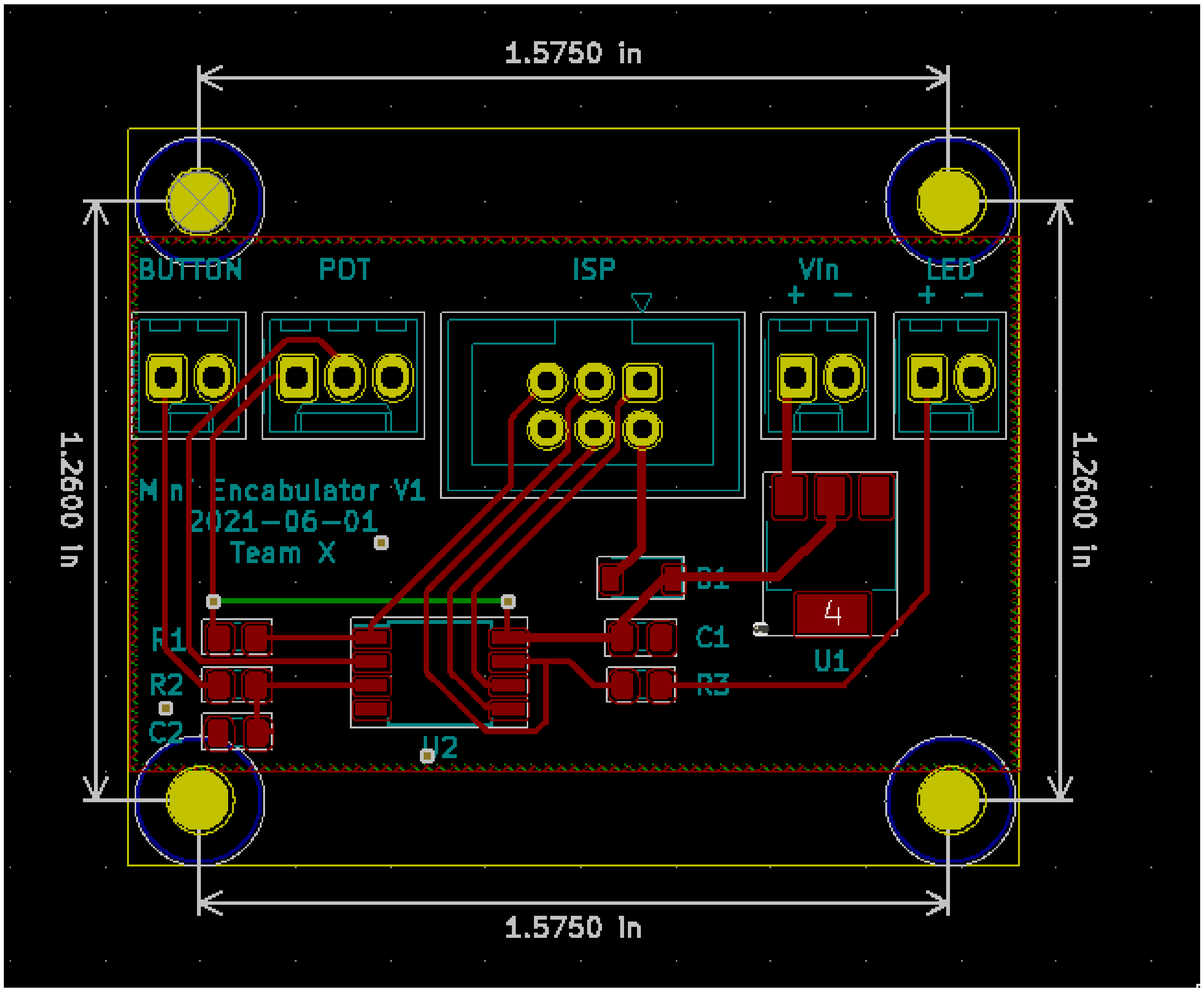
This is because there is not enough space around the regulator to allow the GND zone to flow in around it. You can use the zone visibility settings to see this more clearly. (hit b after changing visibility settings to redraw the zones.)



The unconnected bit of copper on the left is known as a **copper island**. Instead of moving components around to get more space, we can manually add a via.



While you're at it, scatter a few GND vias around other areas of your board. This is known as **ground stitching** and helps keep the current return path short between components. The completed PCB should look something like below:



Submit your assignment by uploading the `.kicad_sch` and `.kicad_pcb` files.

Appendix A: Generating Gerbers

While the PCB is finished, you can't just send your KiCAD files directly to a PCB board house (PCB production company). Instead, we will tell KiCAD to compile our design into a set of files containing low-level instructions that can be understood by the fab house. These files are known as **Gerbers**, and have file extension .grb and .drl.

Choose `File > Plot` to bring open the menu for generating Gerbers. Create a new folder in your project's folder called `gerbers/`, and choose this as the output directly.

The default options should be OK for most board houses. Choose `Generate drill files > Generate drill file` to create the `.drl` file, and `Plot` to create the `.grb` files.

Finally, zip the `gerbers/` folder. This file is what you will upload when ordering a PCB.