

850GHz LED Array

Jonah Lee

Links

- Jonah's notes:
https://docs.google.com/document/d/1SIDZSCZA1UvchlzY03zprPHowxyhJwNkK9p_C4oSYSE/
- Jonah Kicad Github: <https://github.com/jonahjlee/ccat850-led-mapping>
- NIST Collaboration Notes:
<https://docs.google.com/document/d/1gAiYh7eqyPHMh7sOXutrp0bxpR3vbFxyEPAkcoNrAml/>
- Jordan Wheeler:
 - Github: <https://github.com/Wheeler1711/kicad/tree/main>
 - Slides:
<https://docs.google.com/presentation/d/1-Dt53SovuNfGg163UzySefHOoD23g3hIZrU1PkAwqjA/>

280GHz Array vs. 850GHz

280GHz	850GHz
Complete. [slides] [GitHub]	Our goal. Design to be based on 280 GHz array.
288 pixels/network (24x12)	529 pixels/network (46x11.5)
6 networks	12 networks
1728 LEDs total	6348 LEDs total
0402 LED	0201 LED



Setting Up KiCAD

- KiCAD has excellent documentation: <https://docs.kicad.org/>
- Composed of several different programs...
 - pcbnew
 - schematic editor
 - footprint editor
 - etc.

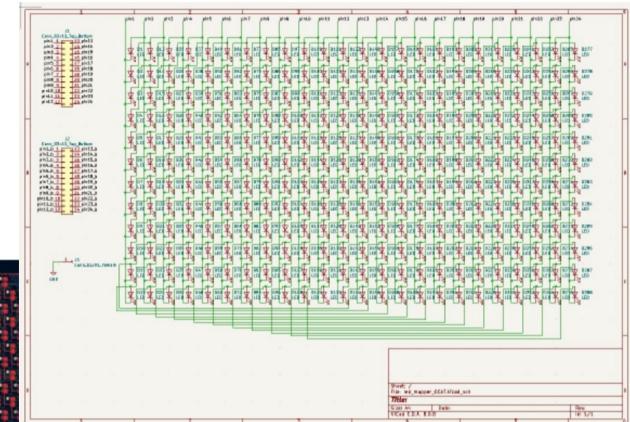
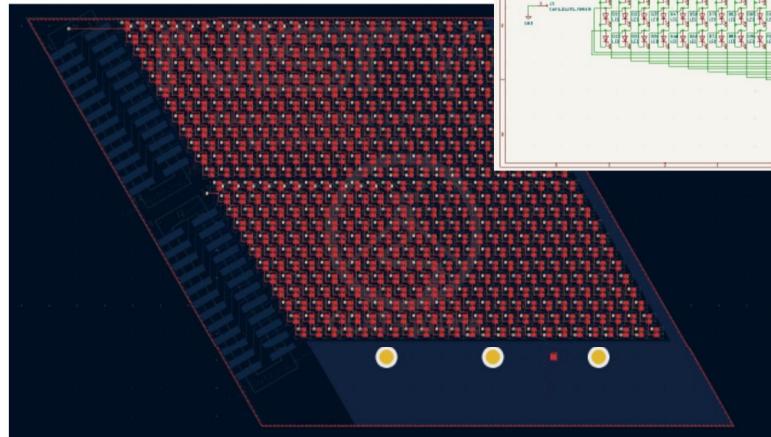
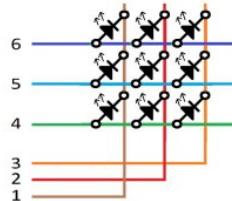


Image source: wikipedia

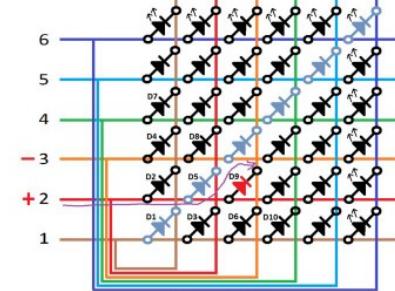
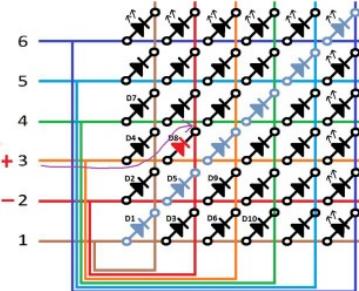
Charlieplexing

- As opposed to multiplexing in a grid, charlieplexing puts an LED in **every network intersection**
- $n(n-1)$ LEDs can be addressed by n pins
- tri-state logic
 - high / low / high-impedance
- <https://en.wikipedia.org/wiki/Charlieplexing>

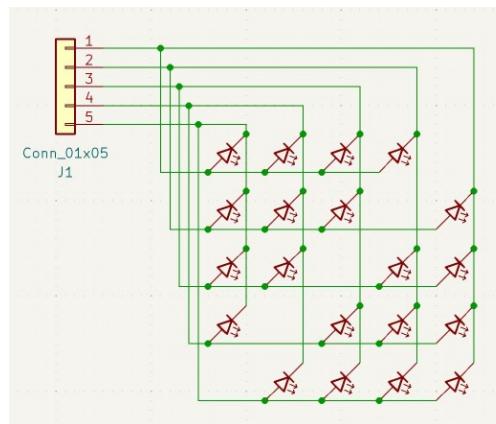
6 (n) i/o forming a standard x/y LED array with $9 ((n/2)^2)$ working intersections .



6 (n) i/o forming a "Charlieplexed" LED array with $30 (n^2 - n)$ working intersections .



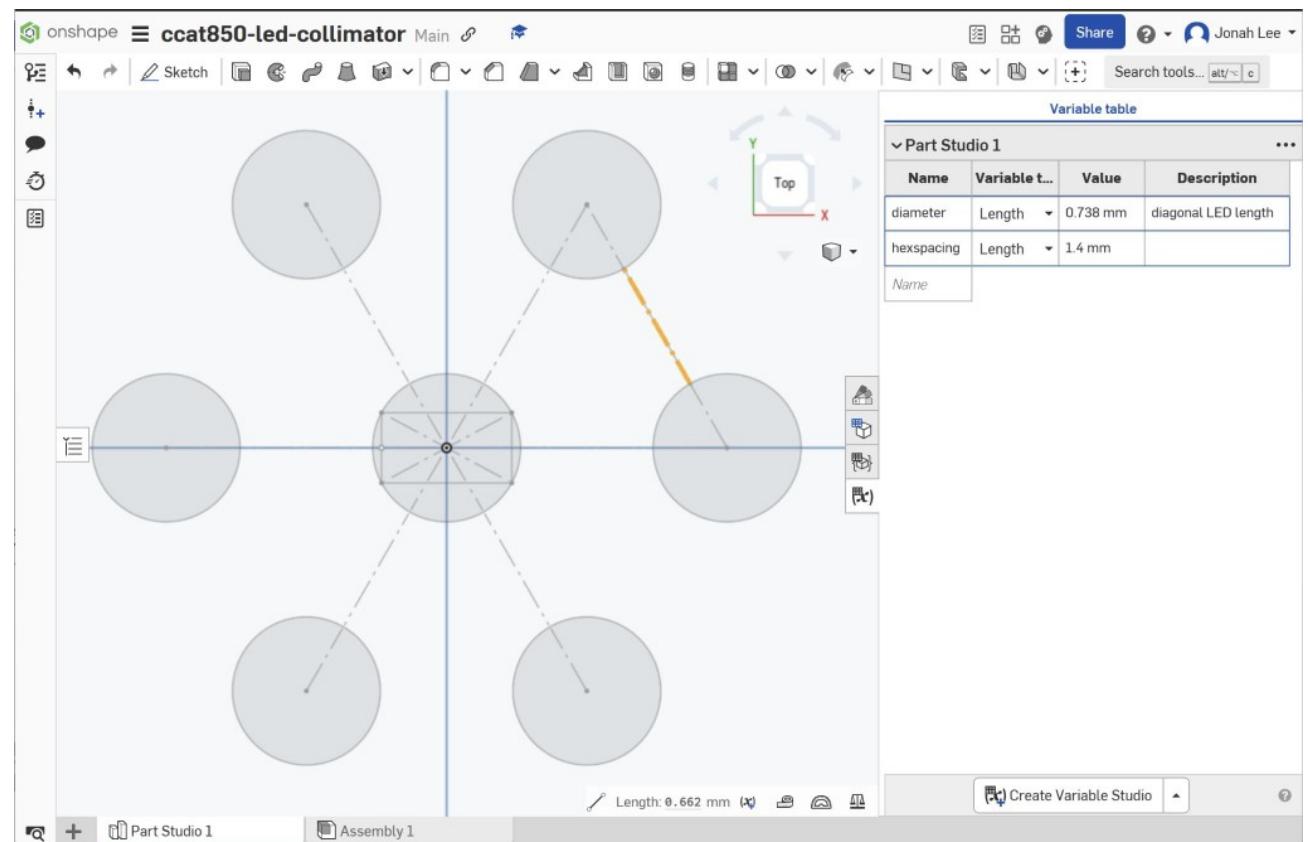
Polarity reversed



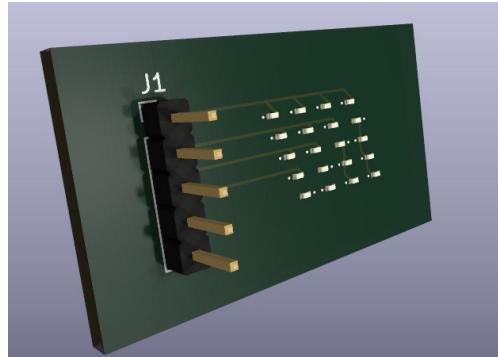
KiCAD practice: 5-pin charlieplexed array

Smallest Collimator Dimensions

<https://cad.onshape.com/documents/4a9874995d3b4039a34f4491/w/a61d1c1f9ee3ea8193c6d7eb/e/afbea4677e4648f6492246a2>



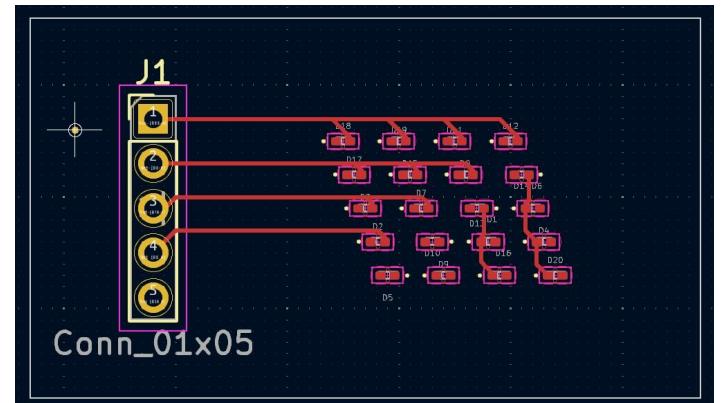
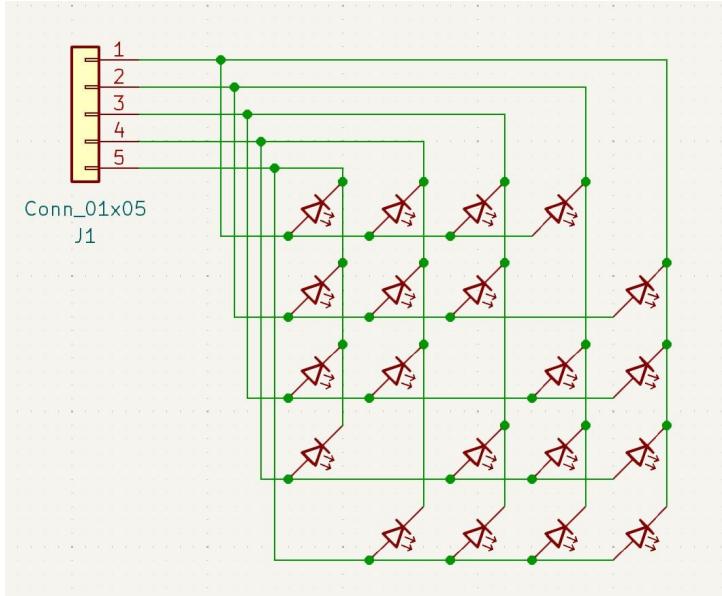
Practice: Charlieplexed Array



Takeaways:

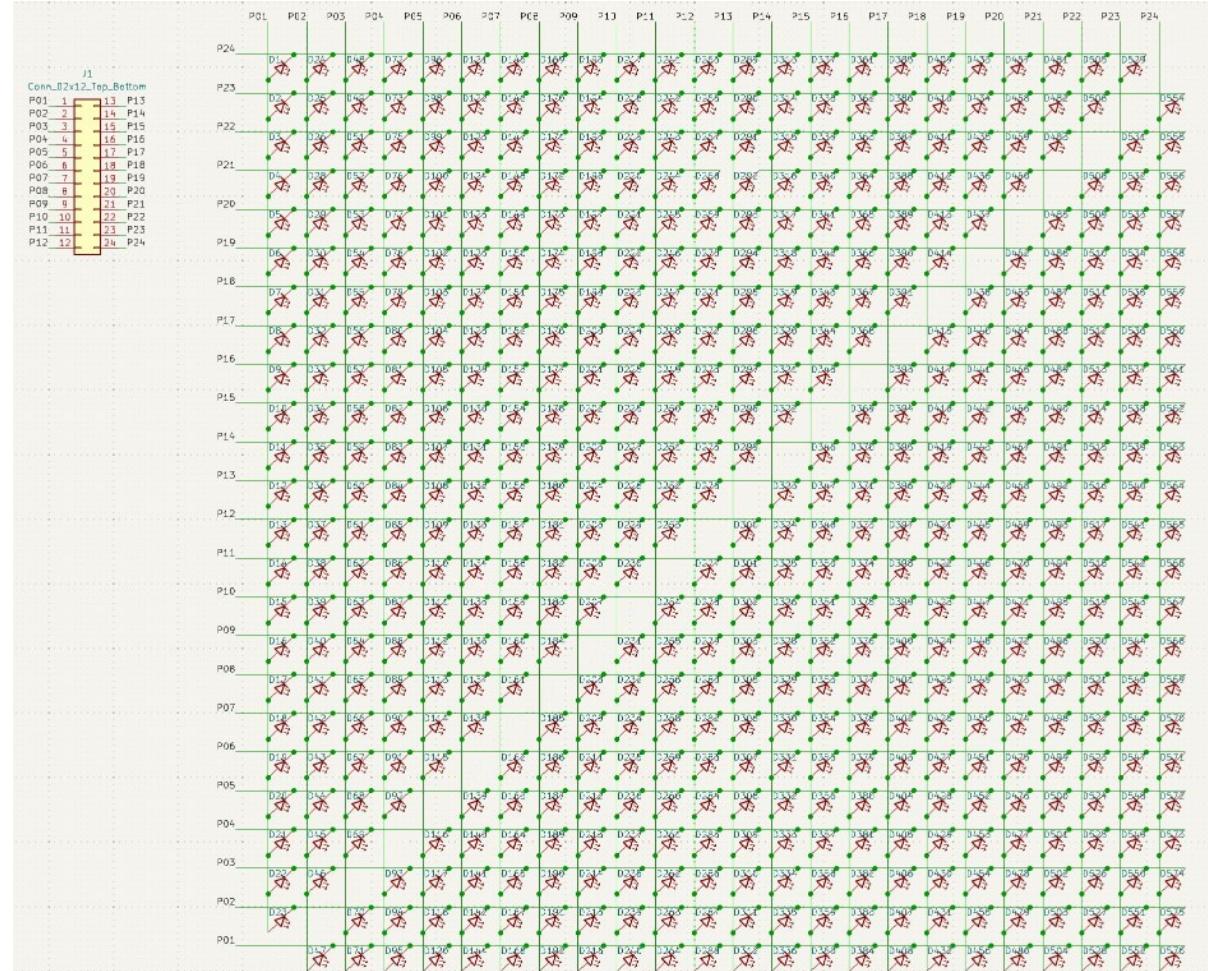
- 45 deg LEDs make schematic easier to read
- Layers / traces:
 - At least 2 layers are necessary
 - 4 layers will be helpful
 - we need to think about space for vias
- Possible trace layout plan:
 - vertical traces on one layer (anode)
 - horizontal traces on another (cathode)
 - short traces to vias
- [Hierarchical schematics](#) will be helpful for keeping organized with the large number of leds / networks

What trace width will we need to deliver enough current?



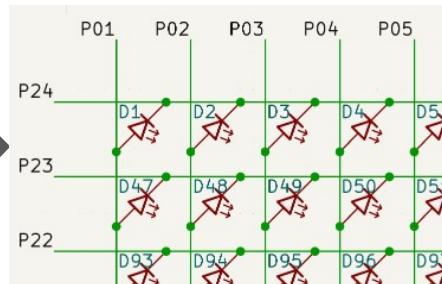
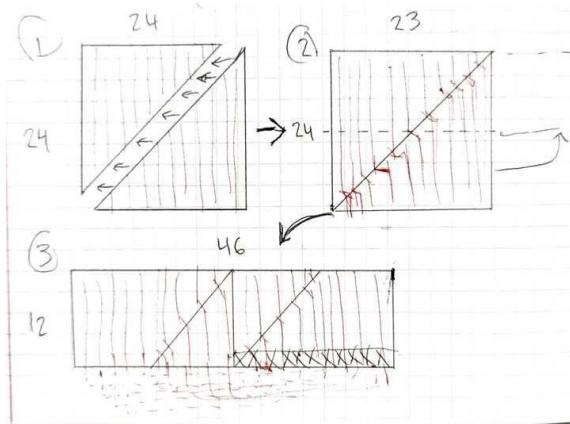
24-pin Array Schematic

- Did not use Python to generate.
Instead used heavy copy/pasting & use of “repeat last item”
 - Each network contains
 $46 \times 11.5 = 529$ LEDs
 - 24 pins is the minimum needed to address a network since $24^*(24-1) = 552$
 - This schematic contains extra LEDs so we can discard some as is convenient by simply removing the LED (wiring should remain the same)



Modified Schematic

- Collapse gap on diagonal → obtain 23x24 grid
- Cut in half & rearrange to get 46x12
- Remove half of last row to get 46x11.5 with exactly 529 LEDs



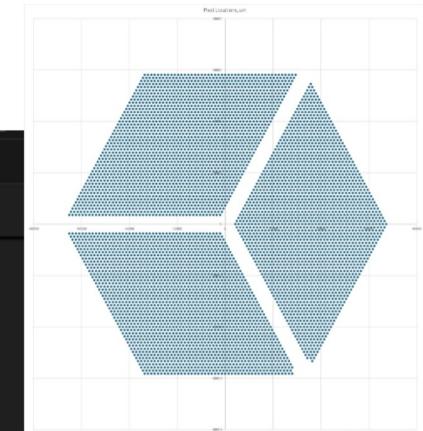
Diode references
arranged neatly in a grid
pattern

Python Code: Generate Layout in Hex Grid

[GitHub repo](#) to
organize KiCad files
& Python scripts

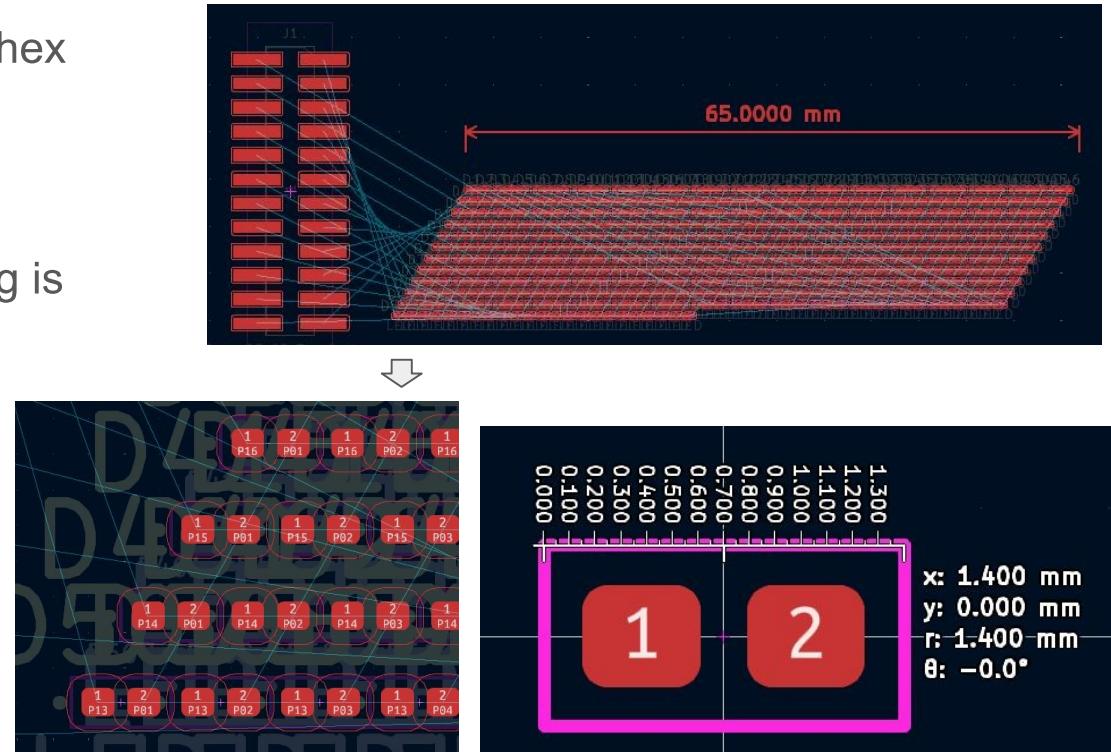
Code only top left
rhombus --
remaining 2 can be
copy/pasted

```
lay_out_pcb.py M X
led_mapper_850 > lay_out_pcb.py > LedPlacer > set_led_positions
  5   class LedPlacer():
  6       def set_led_positions(self) -> None:
  7           for row_idx in range(self.num_rows):
  8               row = row_idx + 1
  9
 10              for col_idx in range(self.num_cols):
 11                  col = col_idx + 1
 12
 13                  # last row is half full
 14                  if (row == self.num_rows and col > self.num_cols // 2): continue
 15
 16                  ref = self.led_ref(row, col)
 17
 18                  fp: pcbnew.FOOTPRINT = self.pcb.FindFootprintByReference(ref)
 19                  assert fp is not None, f'footprint with refdes: "{ref}" not found!'
 20
 21                  if ref == self.reference_led:
 22                      self.set_microns(fp, self.reference_x_um, self.reference_y_um)
 23                      continue
 24
 25                  else:
 26                      x_um = self.reference_x_um + col_idx * self.col_width_um + row_idx * self.row_shift_um
 27                      y_um = self.reference_y_um + row_idx * self.row_height_um
 28                      self.set_microns(fp, x_um, y_um)
```



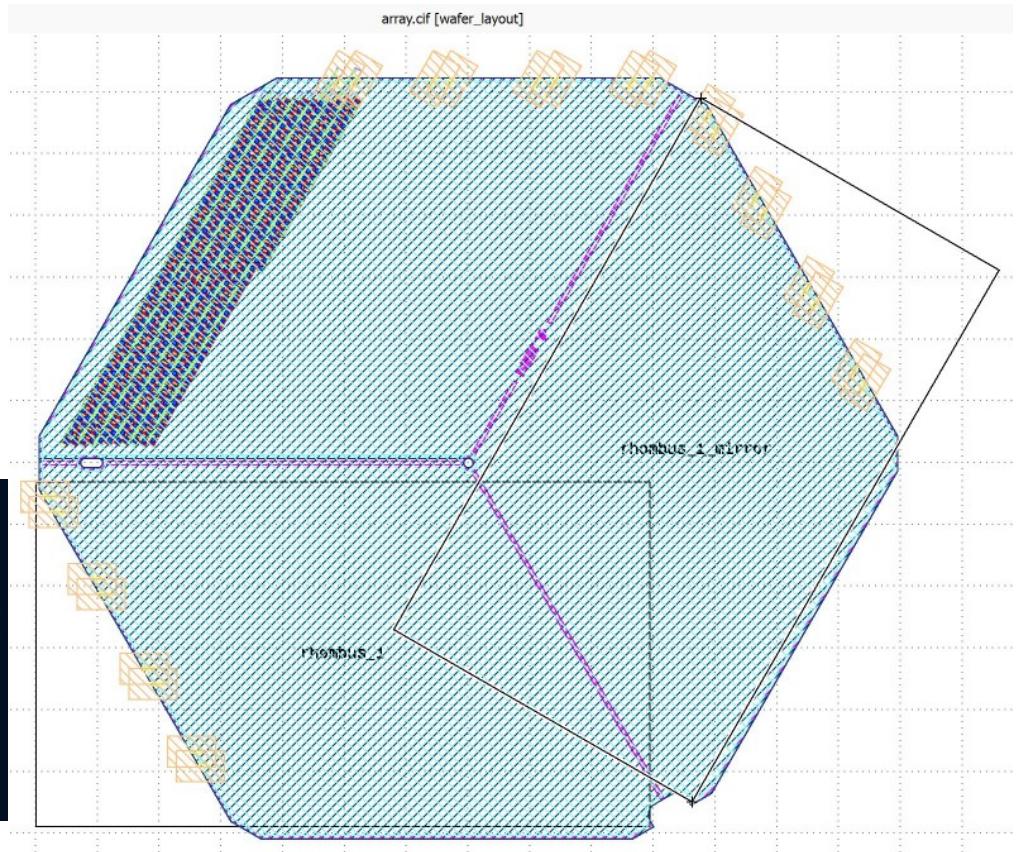
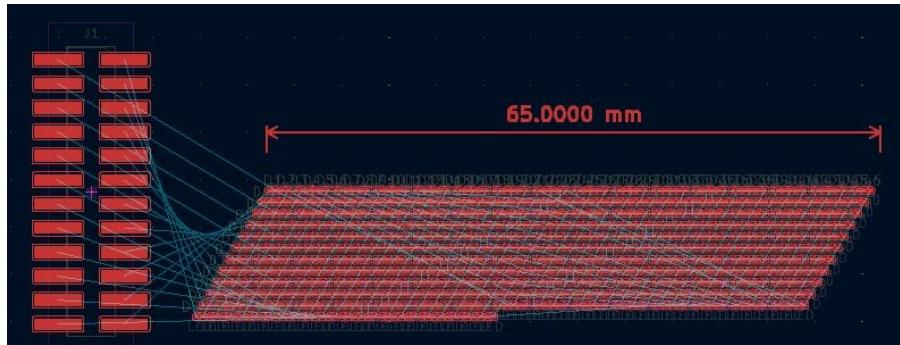
Spacing Concerns

- LEDs are spaced in a 1.4mm hex grid
- Smallest SMD LED ([example](#))
 - '0201' package
 - 0.65 mm x 0.35 mm footprint
- Pads do not touch, but spacing is very tight
- There may not be space for vias/traces
- Rotation of footprints could help save some space?
- **LED density will affect collimator rigidity**
 - max. density requirement?



12 Networks Across 3 “Rhombi”

- LED array needs to be mirrored from detector layout (if viewed from front).
- Network shapes differ slightly, but all have 1.4mm spacing
- Some pixels are missing from bottom right?

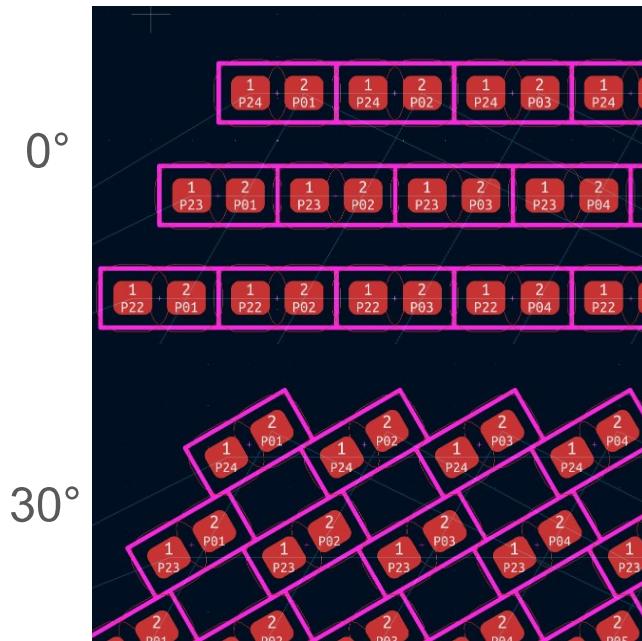


Rotation Experiments

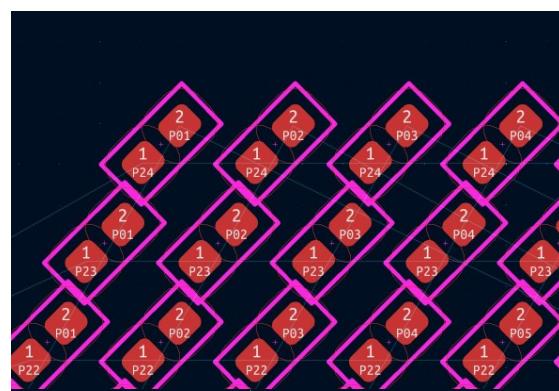
Since we are on a hex grid, angles 60° apart tile similarly. (e.g. 0° vs. 60°)

If we can fit the LEDs and 2 vias per LED, the board should be doable.

The high density may still lead to issues with the collimator.



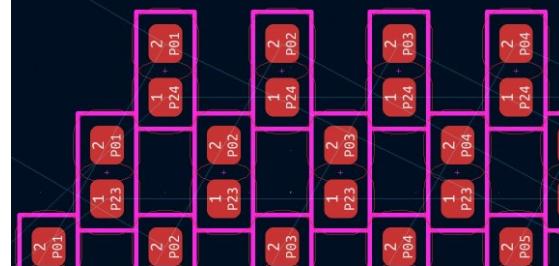
45°



60°

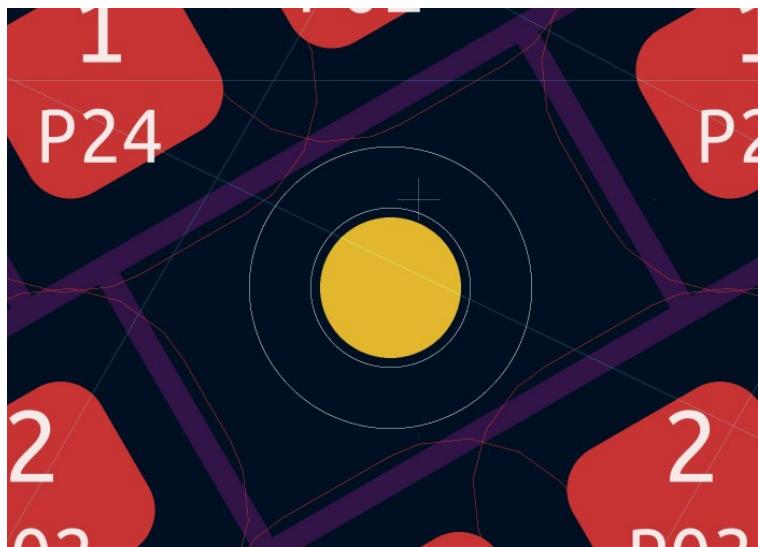


90°



Fitting Vias

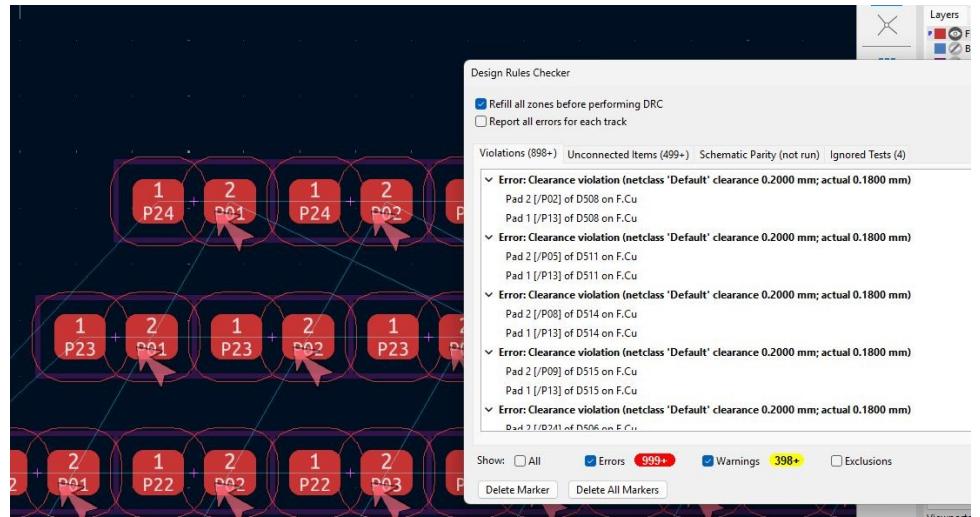
30° grid -- can fit 1 via. (trying to fit 2 leads to DRC violation)



DRC is Not Happy...

Core issue: default clearances are too large for these LEDs. Even 1 LED on its own violates DRC because its pads are so close together.

→ Default clearances do not make sense for this application. We need to check JLC PCB and see what their requirements for manufacturing are.



On a 30° grid, this is the only location allowed for via with default clearance

Change Clearance:
0.2mm → 0.15mm

SMD pad to pad clearance (different nets): 0.15mm

More details of SMD pad spacing:
[SMD Components Minimum Spacing](#)

(no mention of 0201 package
requirement...)



No more DRC errors

Checking JLC PCB Capabilities

- Detailed specs: <https://jlpcb.com/capabilities/pcb-capabilities>
 - SMD pad to pad clearance (different nets) - 0.15mm
- Could use JLC PCB template, e.g.
https://github.com/sethhillbrand/kicad_templates
- Relevant forum post:
<https://forum.kicad.info/t/0201-smd-leds-feasible-project/43714>

bask185 Jul 2023

There is hope!! JLCPCB can manufacture plated via's. So pads in vias can be done

Surface Finish: HASL (with lead) LeadFree HASL

High-spec Options: Vias should be at most 0.55 mm in diameter to avoid incomplete filling. Vias violating this size will be left unfilled.

Outer Copper Weight: 1 oz 2 oz

Via Covering: Tented Untested Plugged Epoxy Filled & Capped Copper Paste Filled & Capped

Vias are filled with epoxy resin and then plated over with copper to create a flat, solderable surface. Ideal for via-in-pad, multilayer boards, and thick boards.

ML9104 Jul 2023

I've been following this thread creating a 1-mm raster LED display with interest (due to a different dormant project). Until now, I didn't realize that 0201 LEDs even exist. 0402, yes. Scanning the interwebs, I found four suppliers for 0201 LEDs: Inolux, Kingbright, LiteON and Lumex. There may be more, please respond if you know any. The KiCAD problem is, that all four suppliers use four different (and incompatible) footprints. The standard "LED_0201_0603Metric" footprint is useless, no such device exists. Doing a PCB layout is perfectly possible with all four, no issue there. So which one to choose? Which footprint will win in the end?

Possible Solution: Via-In-Pad

- Filled vias sit directly in the SMT pad
- Allow for very tight packing of components
- Can lead to increased manufacturing costs
- <https://jlpcb.com/blog/using-via-in-pad-technology>
 - Traditionally, the via-in-pad process has been expensive due to the specialized equipment and materials required. However, JLPCB now provides this advanced capability for free on 6-20 layer PCB orders.
- <https://jlpcb.com/help/article/pcb-via-covering>

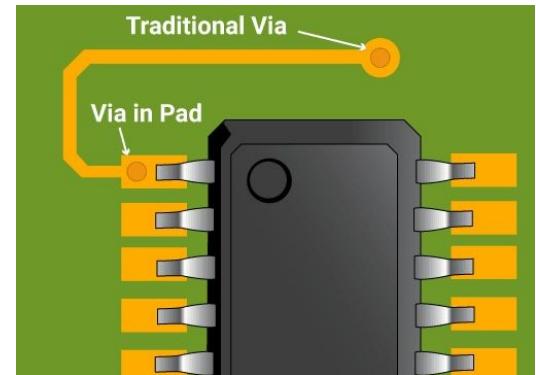
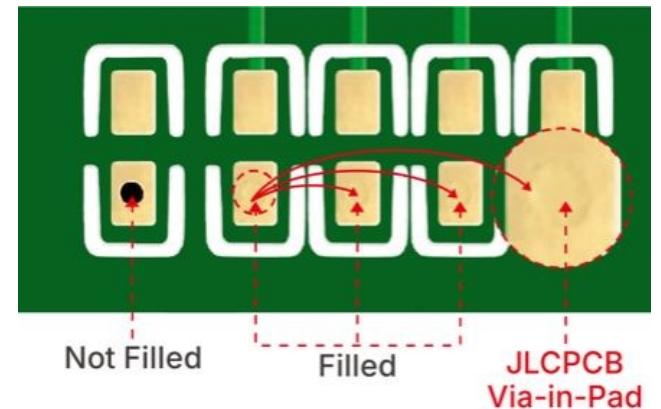


Image source: protoexpress.com



Can we just use an OLED display instead?

Temperature considerations:

- <https://www.sciencedirect.com/science/article/pii/S0379677906000336>
- <https://www.sciencedirect.com/science/article/pii/S0379677915300333>

Misc Thoughts / Things to Keep in Mind

- JLC PCB may not have (enough) LEDs in stock
- Could use filled vias and put the pads directly on top?
- What to set track width to?
- Collimator
 - Xometry?
 - ASU

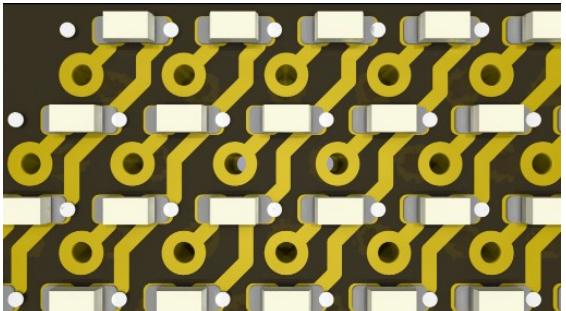
Approach #1: 2 Vias/LED

- 0/60 degree orientation so that LEDs line up
- Smaller vias as necessary to fit 2 per LED
- 3-4 layer pcb
 - front - LED footprints / 2 plated through-hole vias each
 - layer to route parallel “rows”
 - layer to route parallel “columns”
 - optional layer to connect 2 sections of each network
 - (we had to split the 24x23 network into 2 12x23 rectangles to get desired dimensions, but the rectangles need to be wired together)
 - otherwise, this could probably be done on other layers
- Questions/concerns
 - Can I shrink the vias enough to fit them all?

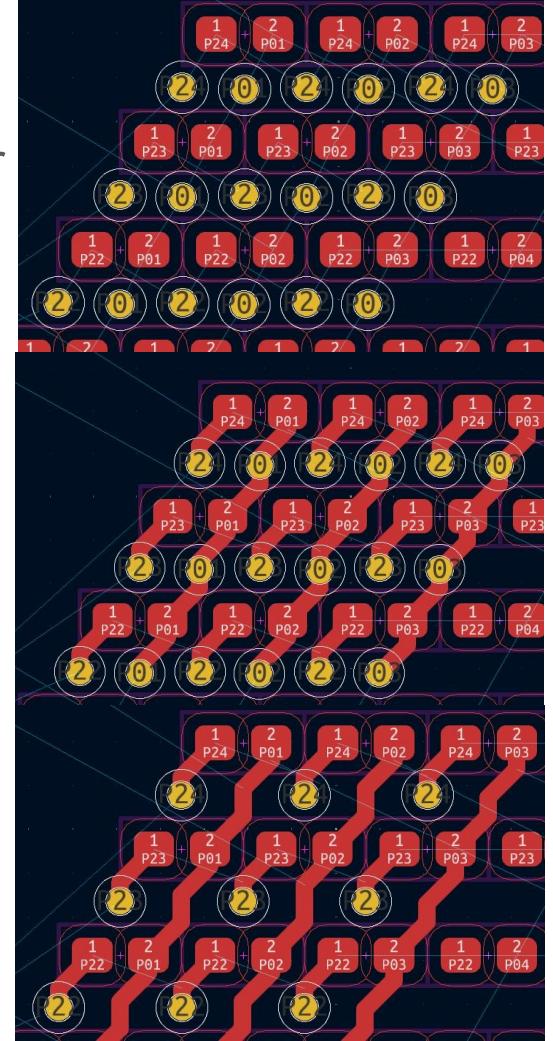


Some Good Results

- Able to fit 2 vias per led without changing via size (only reduced netclass clearance from 0.2 to 0.15mm)
- Because of column arrangement, vias on pin 2 may not even be necessary except to join networks
- This could mean a 2-layer pcb could work well



2 vias per LED



Routing nets

Remove unnecessary vias

Coding Vertical Traces...

```
class LedPlacer():
    def route_columns(self) -> None:
        for col_idx in range(self.num_cols):
            col = col_idx + 1

            if 1 <= col <= 12 or 35 <= col <= 46:
                # left & right rectangular sections -- simply route all the way down
                start_row = 1
                end_row = 12 if col <= 23 else 11
                start_pad: pcbnew.PAD = self.led_footprint(start_row, col).FindPadByNumber(2)
                end_pad: pcbnew.PAD = self.led_footprint(end_row, col).FindPadByNumber(2)
                self.add_track(start_pad, end_pad)

            if 13 <= col <= 22:
                # triangle top left half
                start_row = 1
                end_row = 24 - col # as we go to the right, we end further up
                start_pad: pcbnew.PAD = self.led_footprint(start_row, col).FindPadByNumber(2)
                end_pad: pcbnew.PAD = self.led_footprint(end_row, col).FindPadByNumber(2)
                self.add_track(start_pad, end_pad)

            if 14 <= col <= 23:
                # triangle section 1 bottom left half
                start_row = 25 - col # as we go to the right, we start further up
                end_row = 12 # left half has full height
                start_pad: pcbnew.PAD = self.led_footprint(start_row, col).FindPadByNumber(2)
                end_pad: pcbnew.PAD = self.led_footprint(end_row, col).FindPadByNumber(2)
                self.add_track(start_pad, end_pad)

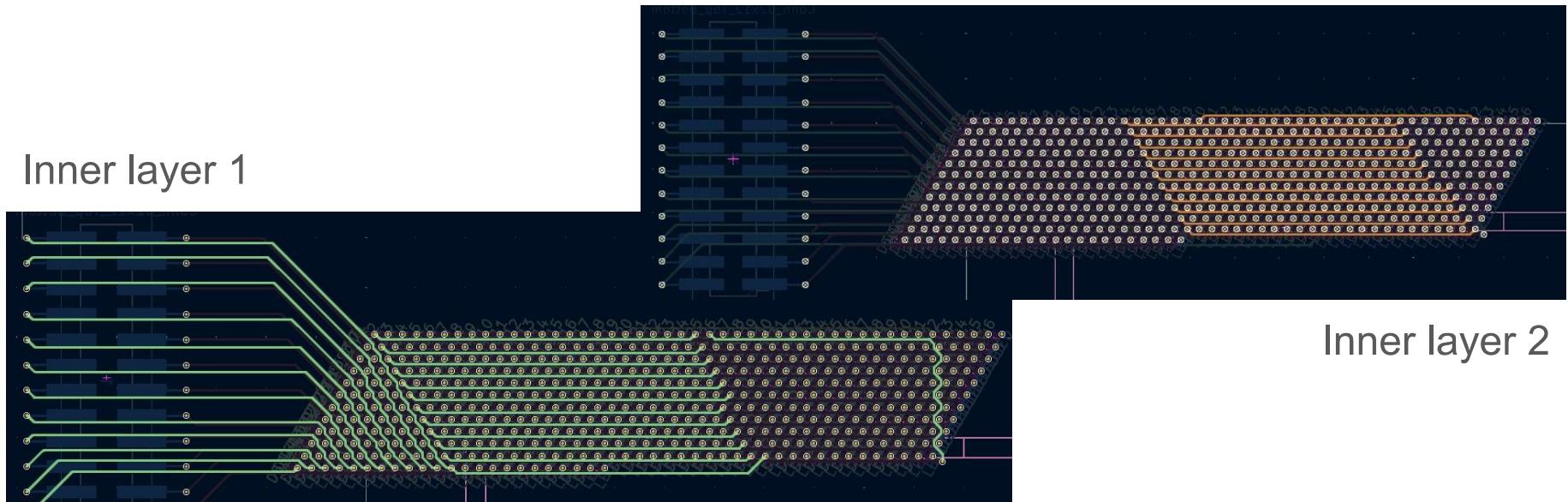
            if 24 <= col <= 33:
                # triangle top right half
                start_row = 1
                end_row = 35 - col # as we go to the right, we start further up
                start_pad: pcbnew.PAD = self.led_footprint(start_row, col).FindPadByNumber(2)
                end_pad: pcbnew.PAD = self.led_footprint(end_row, col).FindPadByNumber(2)
                self.add_track(start_pad, end_pad)

            if 26 <= col <= 34:
                # triangle section 1 bottom right half
                start_row = 36 - col # as we go to the right, we start further up
                end_row = 11 # left half has full height
                start_pad: pcbnew.PAD = self.led_footprint(start_row, col).FindPadByNumber(2)
                end_pad: pcbnew.PAD = self.led_footprint(end_row, col).FindPadByNumber(2)
                self.add_track(start_pad, end_pad)
```



Stitching Sections - 4-layer design

By rotating 60 degrees, we can route pin 1 traces on the front, and pin 1 goes to through-holes. The PTH vias allow us to use additional layers in between to “stitch” the two rectangles together.



PCB Layout Procedure

bold = automated method in
[led_placer.py](#)

FRONT LAYER

- 1. Place LEDs**
- 2. Place & route pad 2 vias**
- 3. Route front rows**
- 4. Short front pins**

BACK LAYER

- 1. Route back columns**
- 2. Stitch diagonals**

INNER LAYERS

- 1. Inner layer 1 - stitch rectangles**
- 2. Inner layer 2 - stitch rectangles**

CONNECTOR

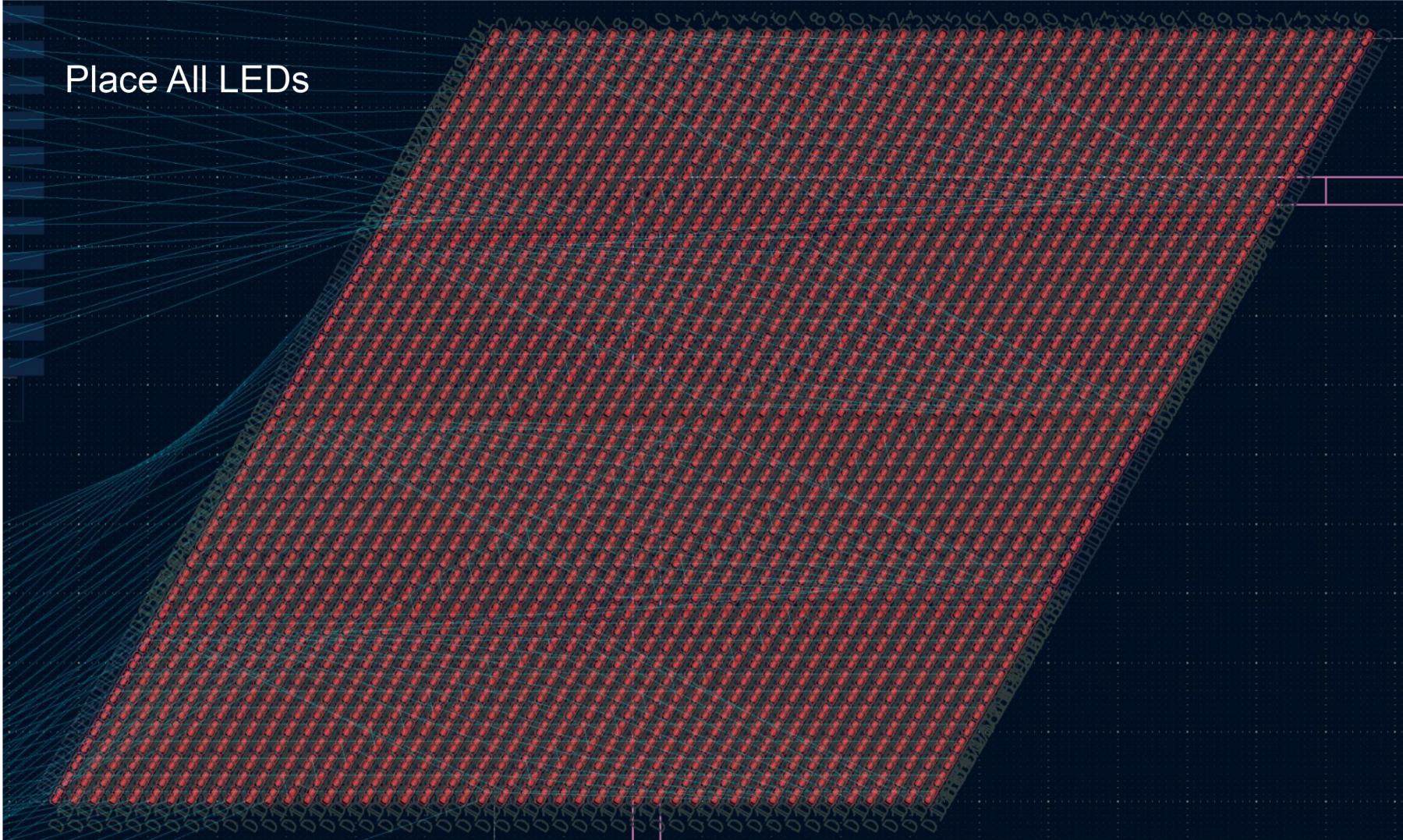
- 1. Add & route connector vias**
- 2. Route pins on front layer**
- 3. Route pins on inner layer 1**

MISC

- 1. Route bottom row right side**
 - a. add new via and route on inner layer 2**
- 2. Add miscellaneous remaining connections**
 - a. inner layer 2 has a decent amount of space, and there are also spaces between networks on the back layer and on inner layer 1**

REPEAT FOR ALL 4 ARRAYS

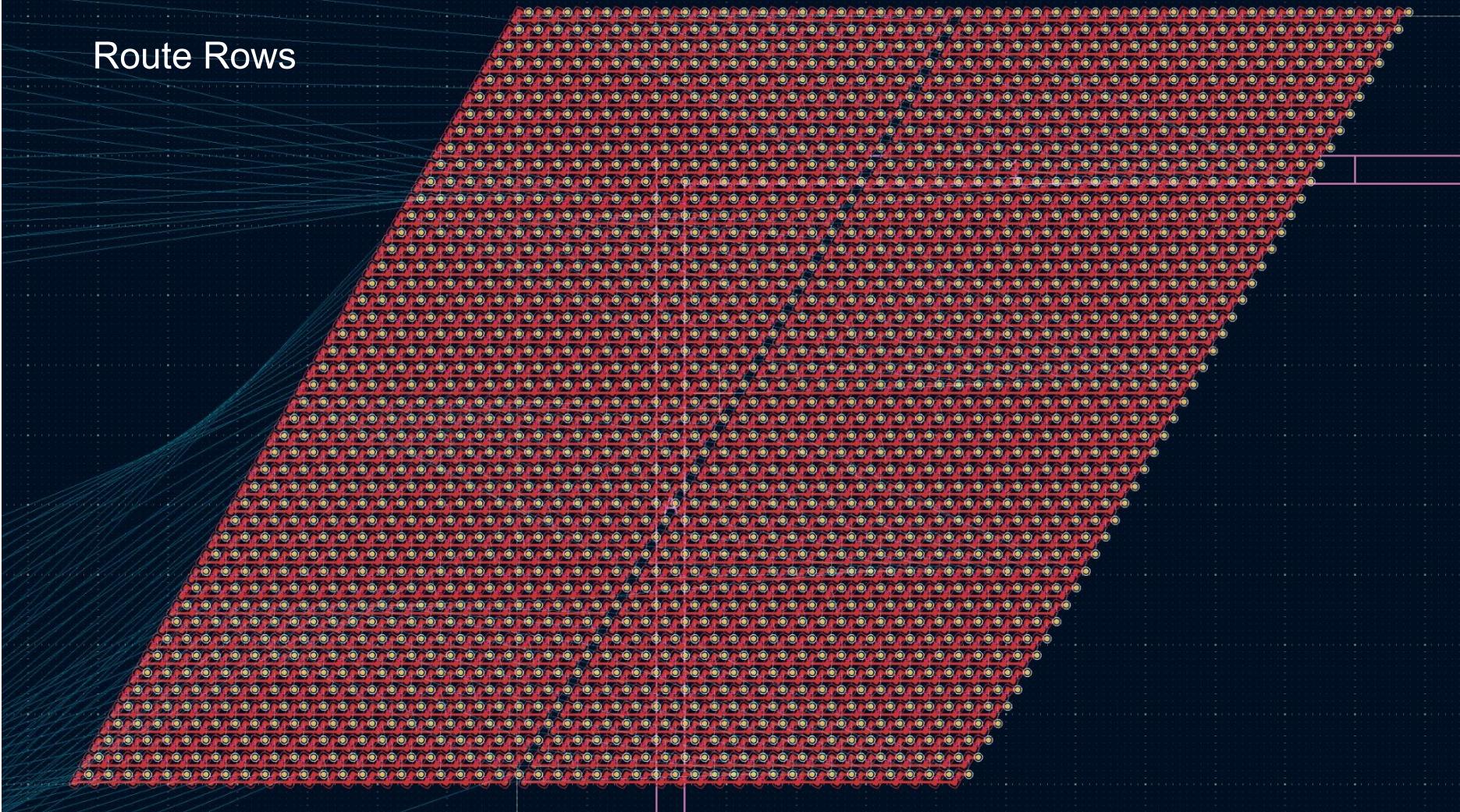
Place All LEDs



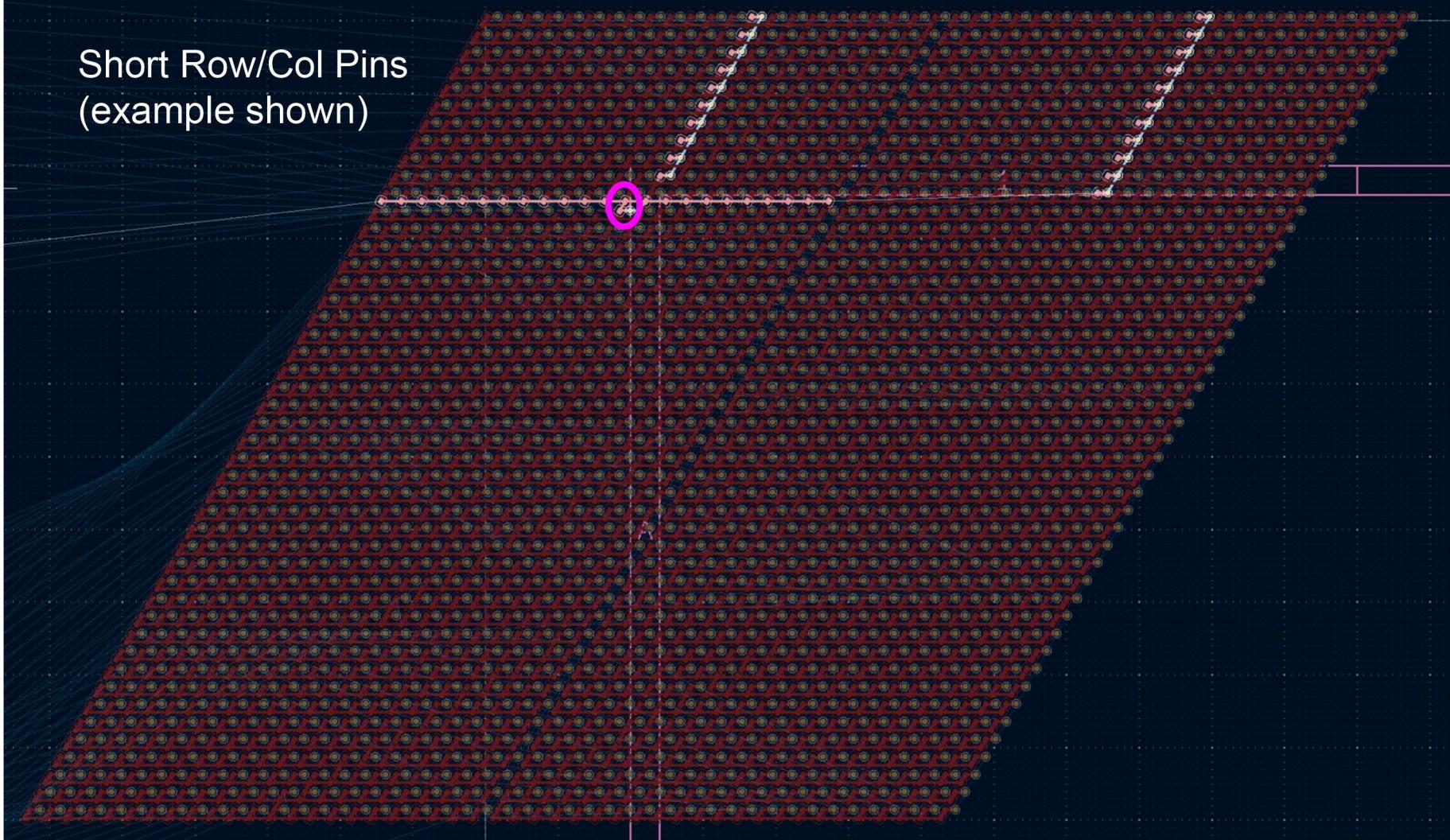
Place Pad 2 Vias



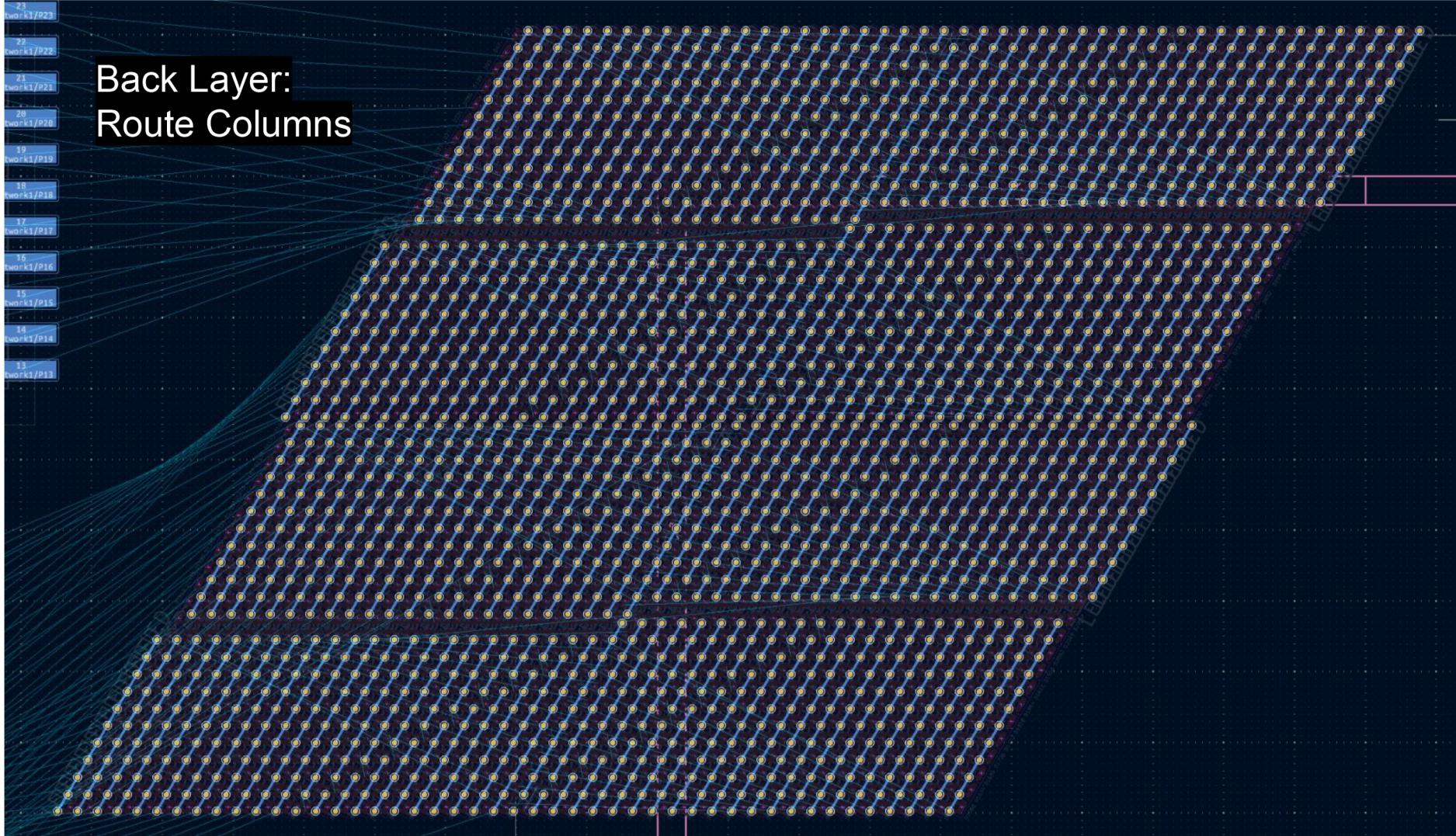
Route Rows



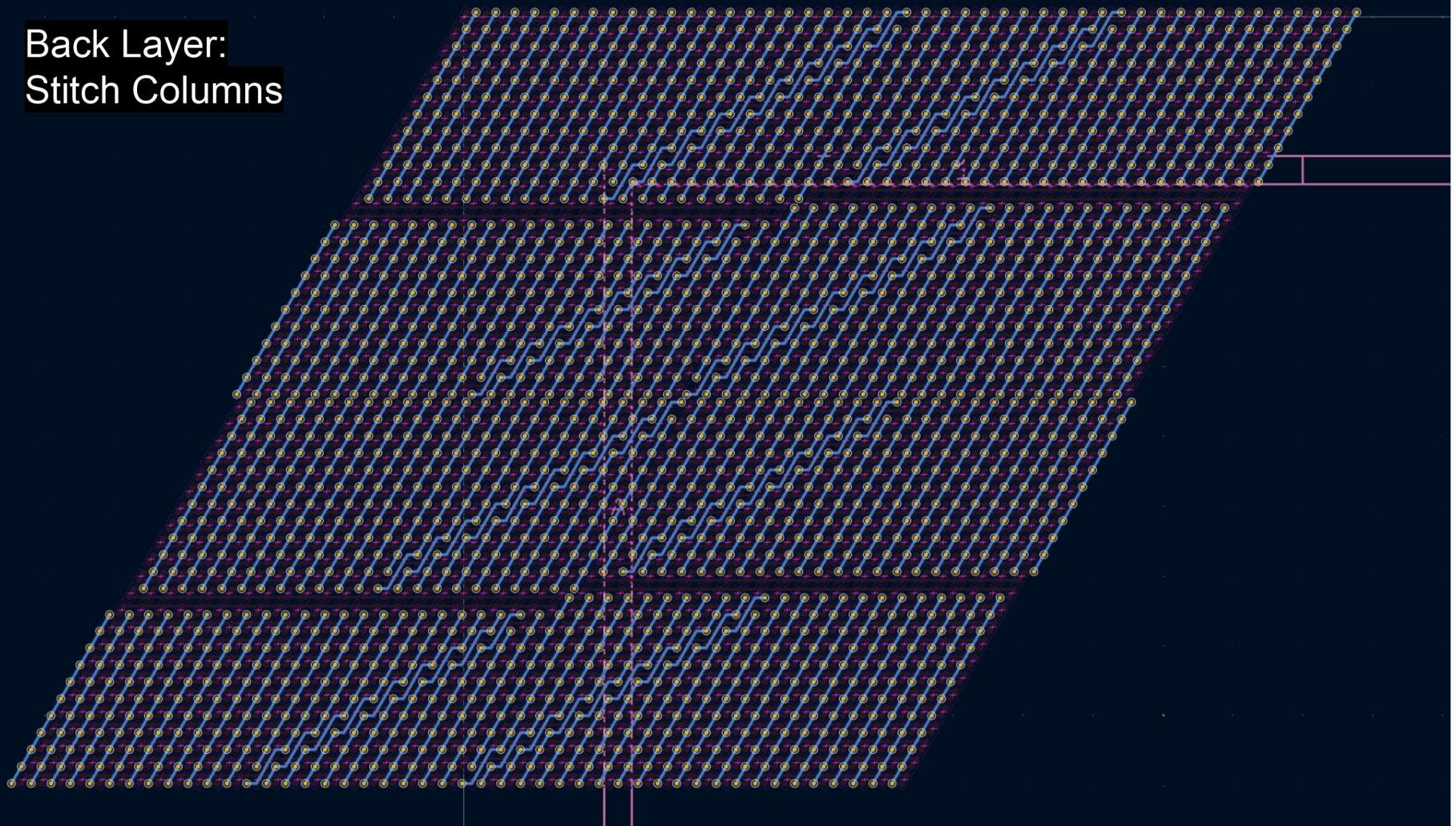
Short Row/Col Pins
(example shown)



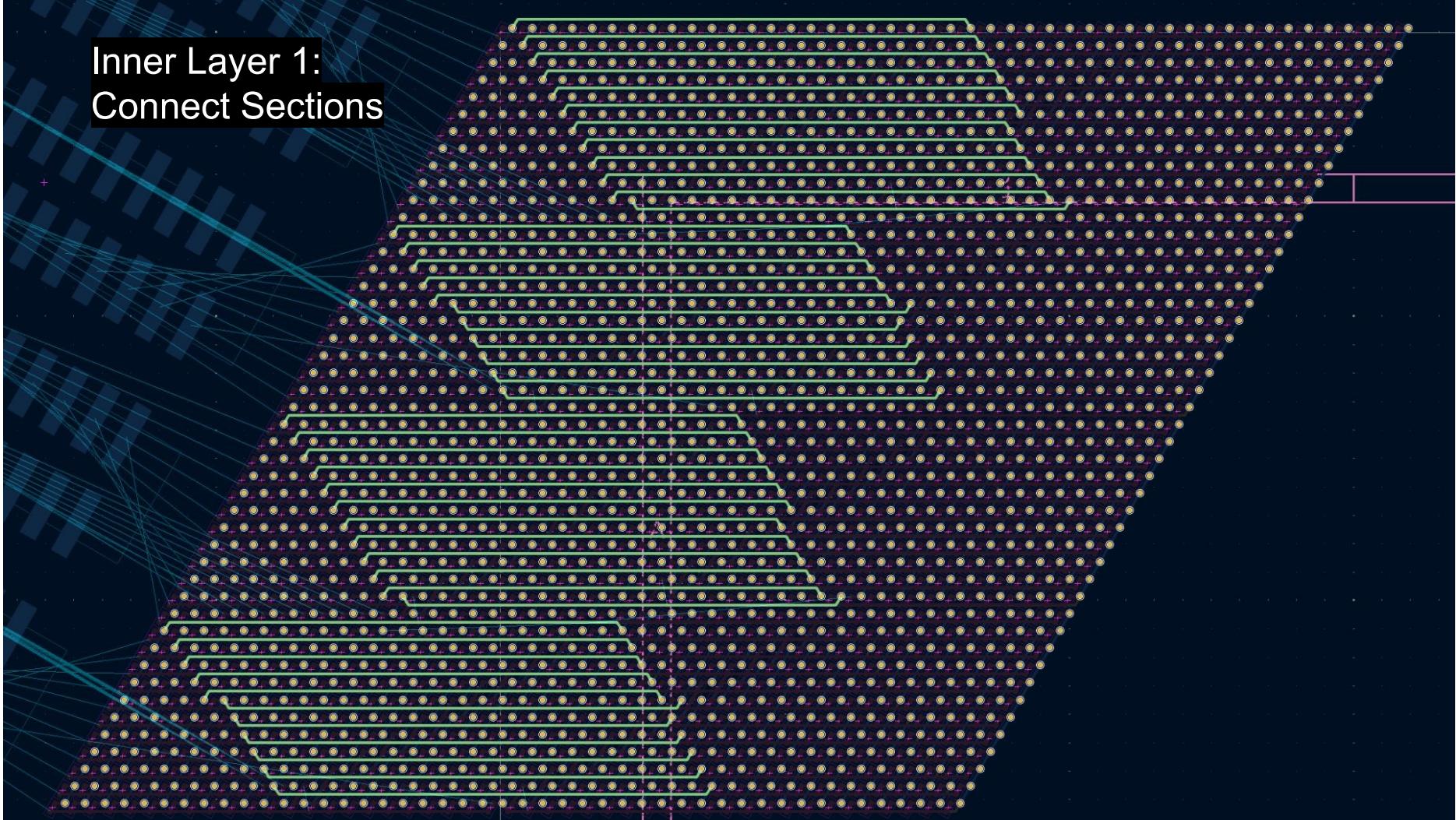
Back Layer: Route Columns



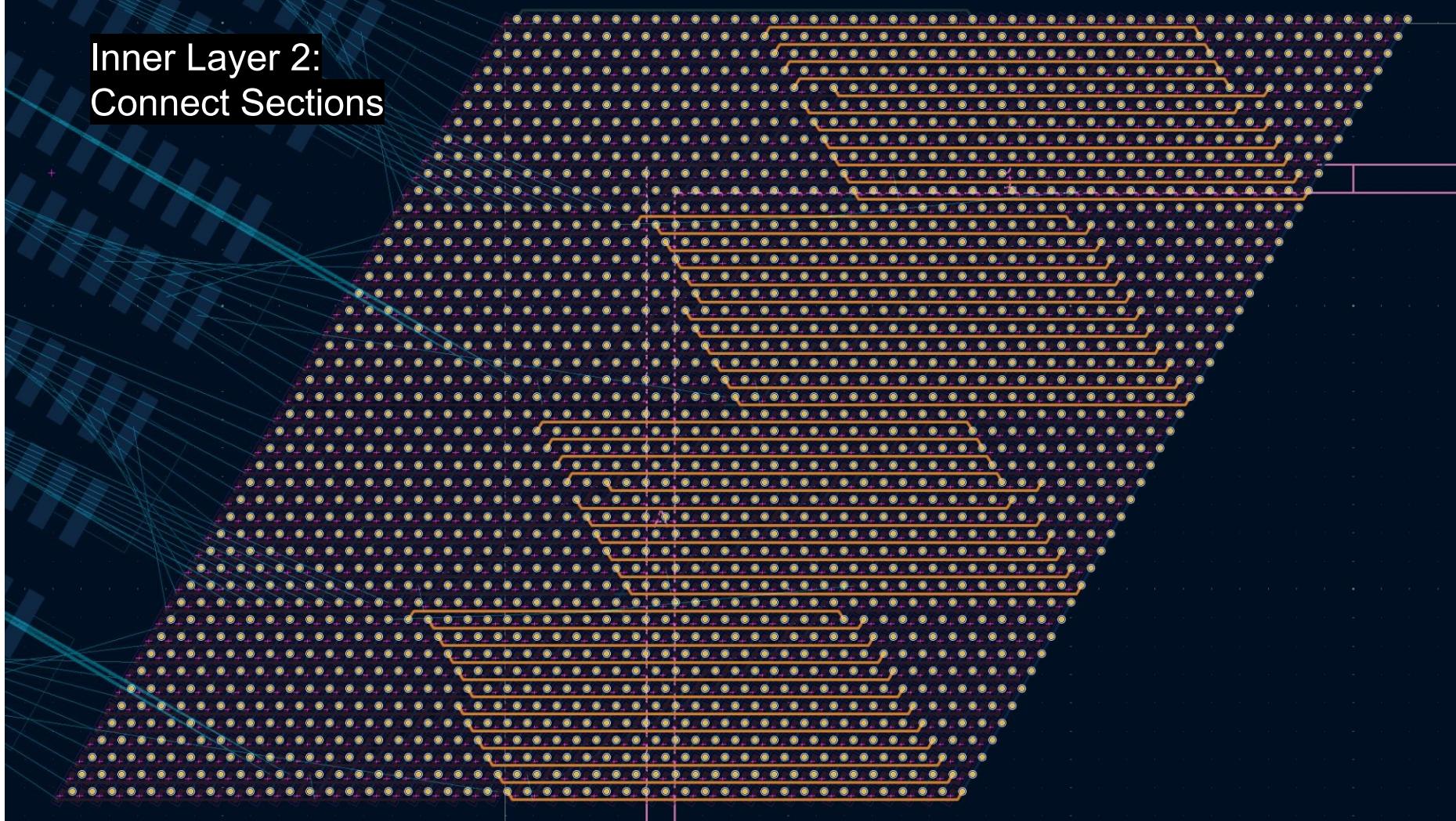
Back Layer: Stitch Columns



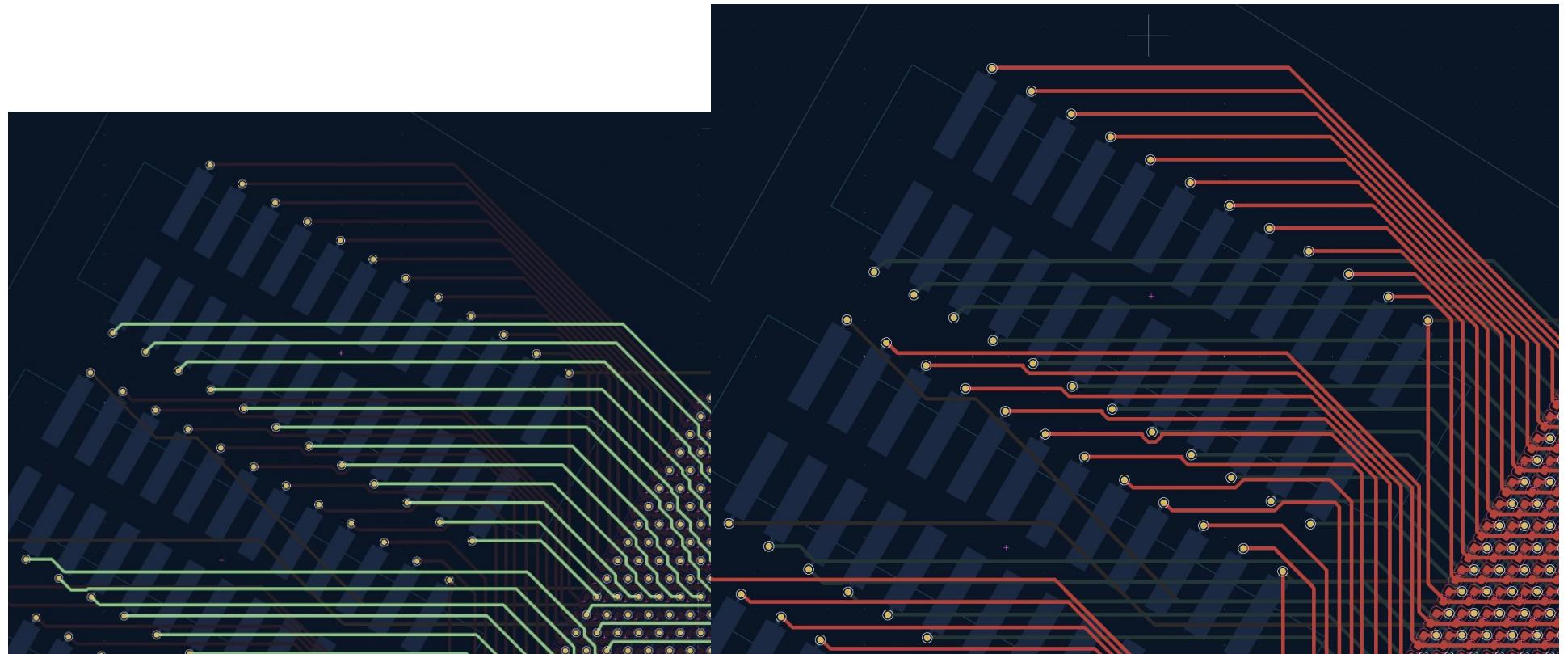
Inner Layer 1:
Connect Sections



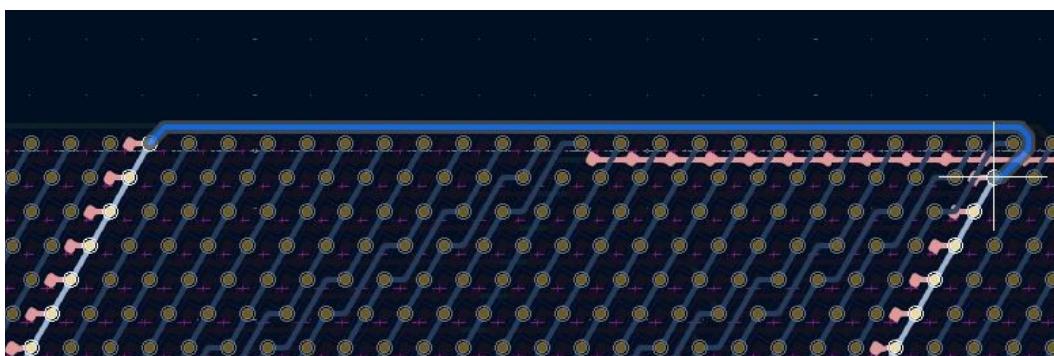
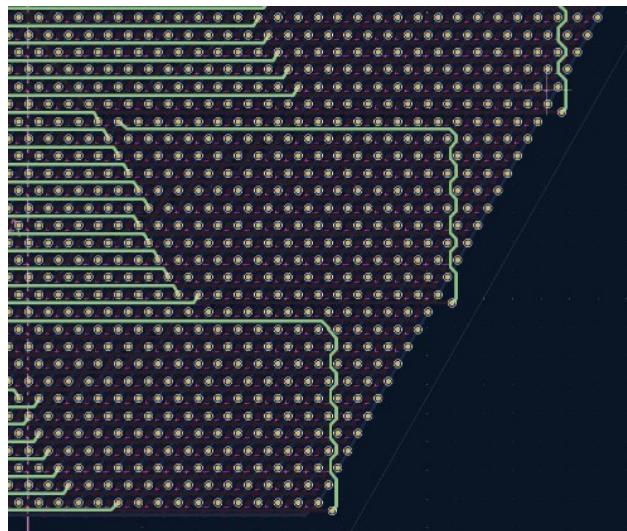
Inner Layer 2: Connect Sections



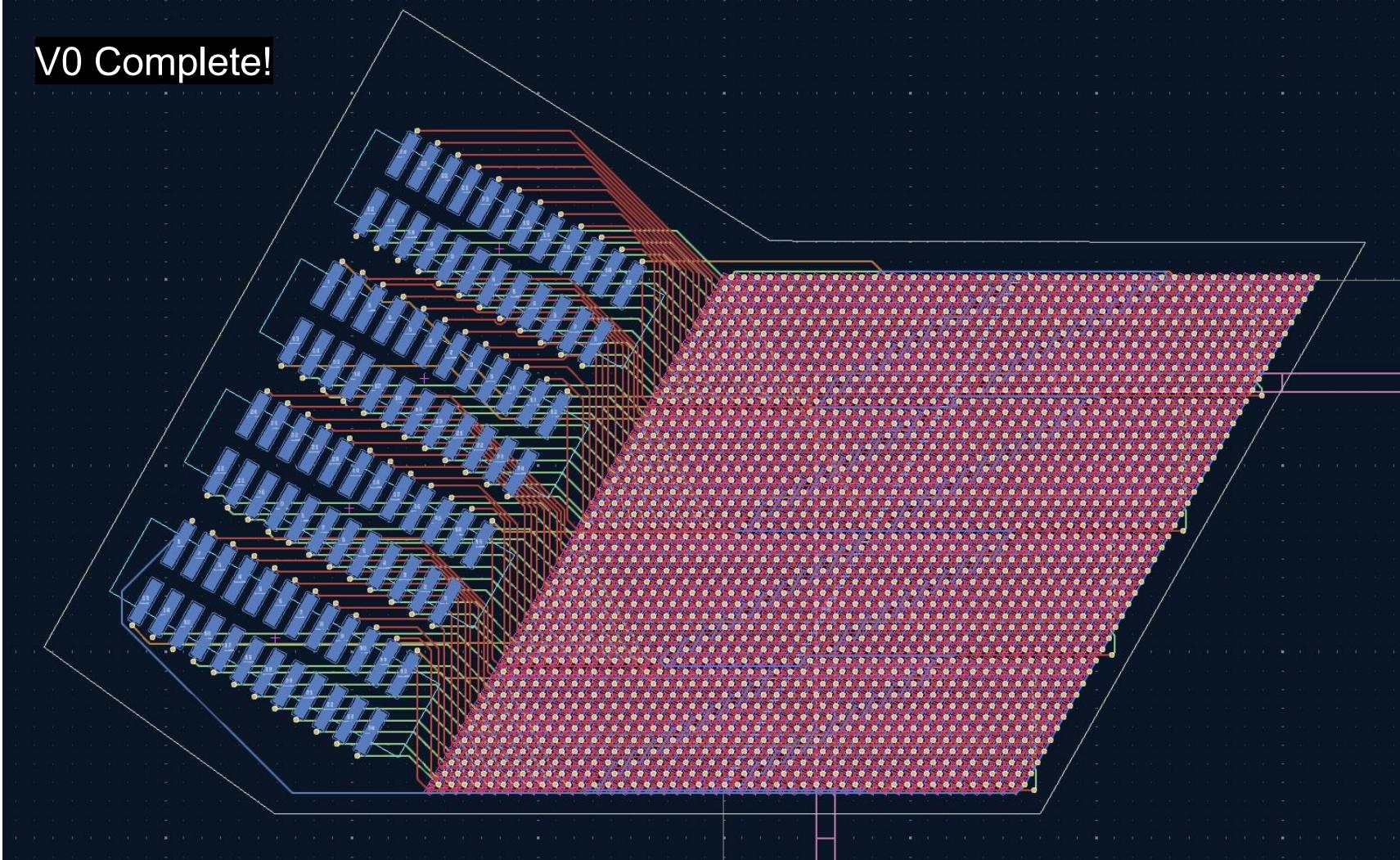
**Connector: Place Vias
& Wire on Front & Inner Layer 1**



Misc. Manual Cleanup

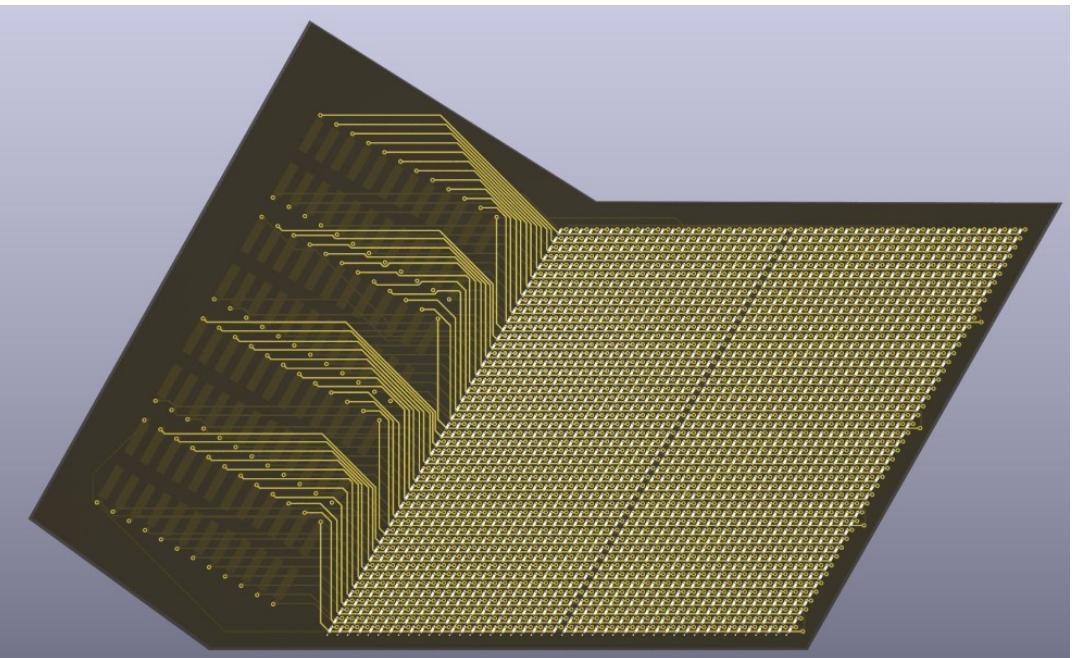
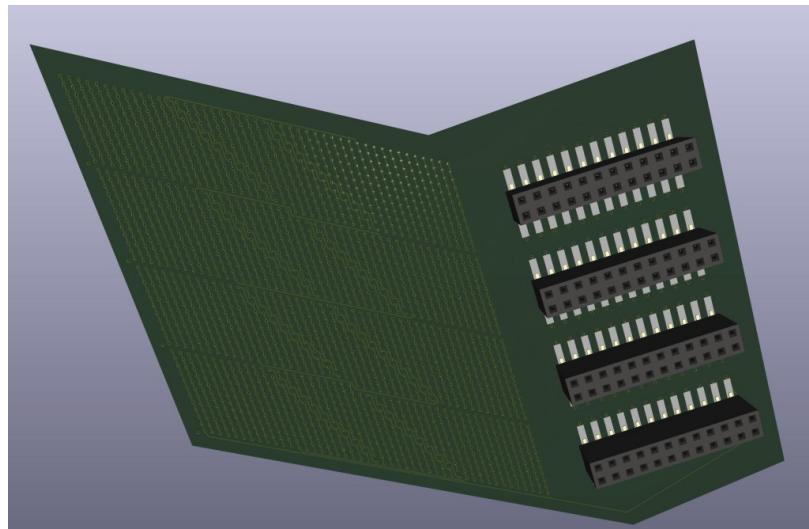


V0 Complete!



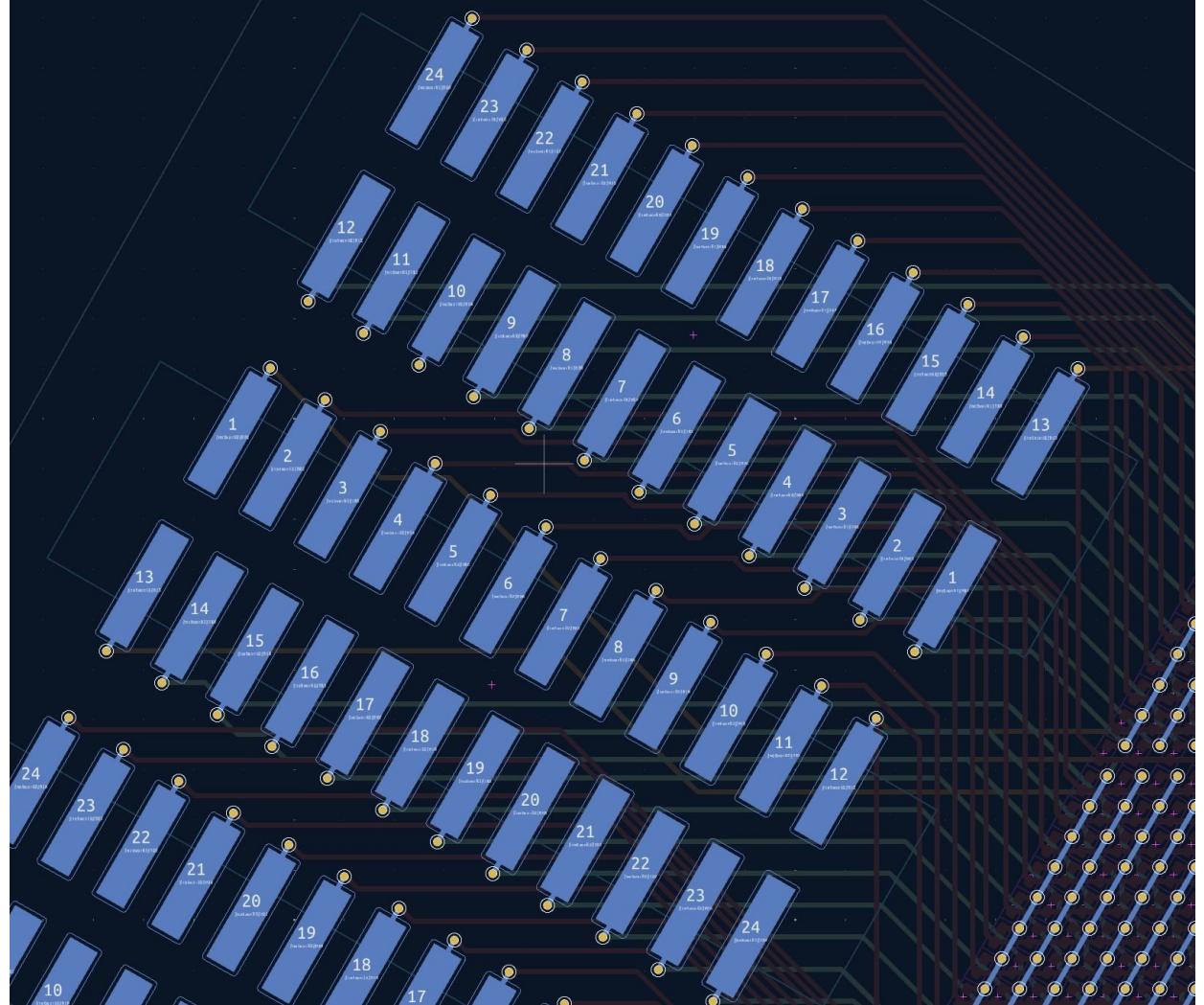
Board Renders

Wires are a bit messy for now, but will be cleaned up when connector positions /edge cuts are finalized



Connector Orientation

In order to simplify wiring, connectors were placed in alternating direction. If this is problematic, it is possible to rearrange the LEDs such that this is not the case.



Uploading Gerber Files to JLCPCB

5 boards = \$30 USD

PCBA is probably going to be much more expensive,
but I need to log in to view the price.

0201 packages are not supported for economic PCBA; standard is required

(<https://jlcpcb.com/capabilities/pcb-assembly-capabilities>)

The screenshot shows the JLCPCB website interface for uploading Gerber files. At the top, there are tabs for Standard PCB/PCBA, Advanced PCB/PCBA (highlighted with a yellow 'Limited Time Offer' banner), SMT-Stencil, and 3D Printing/CNC. Below the tabs, there are two 3D models of a green PCB assembly. A message indicates 'Detected 4 layer board of 86.4x142mm(3.4x5.59 inches)'. To the right, there's a 'Gerber Viewer' button. On the left, there are several configuration options: 'Base Material' (FR-4 selected), 'Layers' (4 selected), 'Dimensions' (142 * 86.4 mm), 'PCB Qty' (5 selected), and 'Product Type' (Industrial/Consumer electronics selected). On the right, the 'Charge Details' section lists costs: Engineering fee (\$24.00), Via Covering (\$0.00), Surface Finish (\$0.00), and Board (\$6.30). The 'Build Time' section shows options for 3-4 days (selected) and 3 days, both costing \$0.00. The 'Calculated Price' is \$30.30. A 'SAVE TO CART' button is available. Below it, a 'Shipping Estimate' section shows DHL Express at \$17.50 with a 2-4 business days delivery time and a weight of 0.34kg. There are also 'Coupons' and discount buttons.