PCB schematic capture & layout best practices

Some tips to make life easier and your design more likely to work!

Schematic design

- There are two audiences for your schematic
    - KiCAD
        - A correct schematic will allow KiCAD to perform checks when you do layout
            - E.g., is the ground pin on the IC physically wired to the circuit ground
    - People
        - To help debug, to instruct, to share
- We thus have two overall goals
    - The schematic should be correct
    - The schematic should be easy to read

    This is just like for code

- Little board for an SGP41 VOC sensor

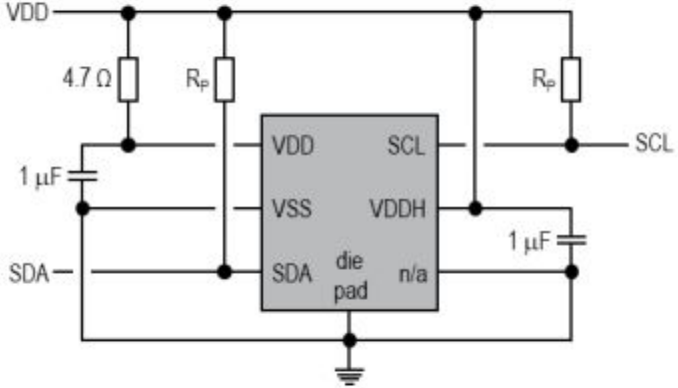
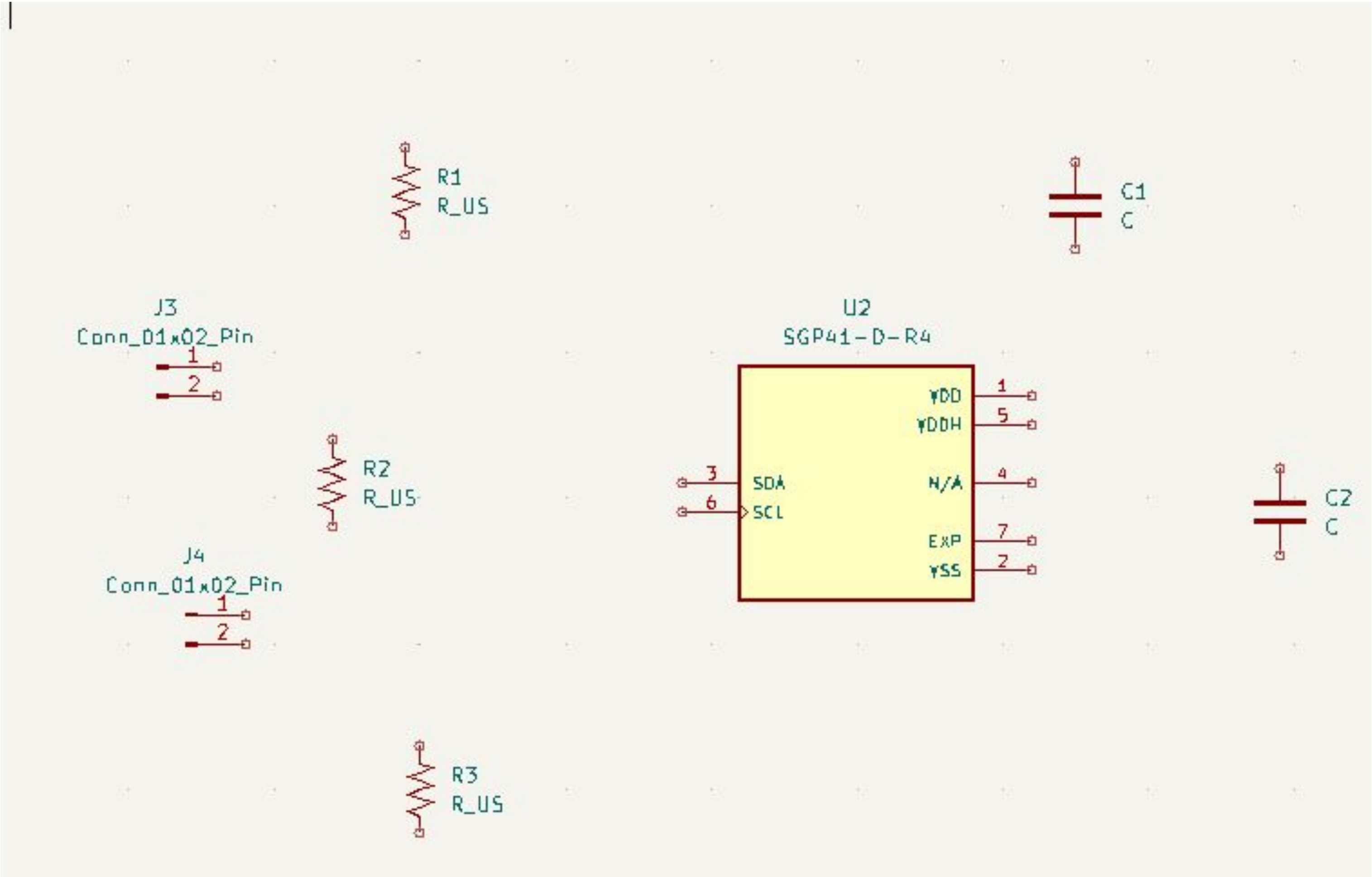
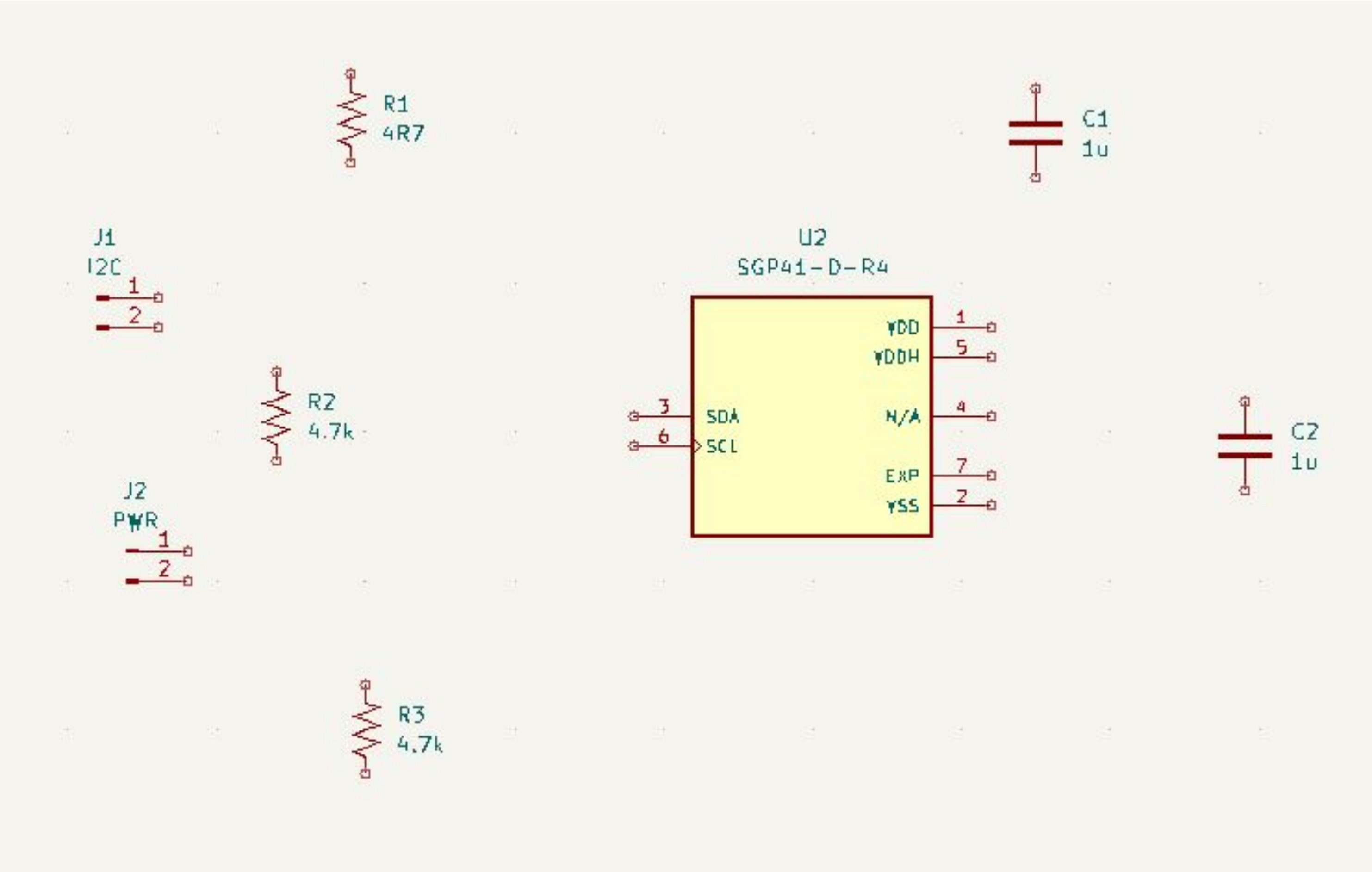


Figure 6 Typical application circuit.

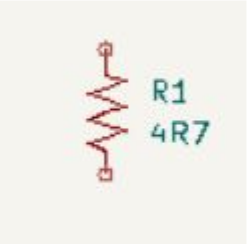


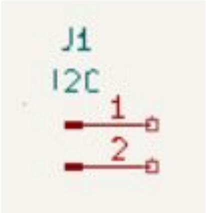
R1
R_US
C1
C
J3
Conn_01x02_Pin
U2
SGP41-D-R4
VDD
VDDH
SDA
SCL
N/A
EXP
VSS
R2
R_US
C2
C
J4
Conn_01x02_Pin
R3
R_US

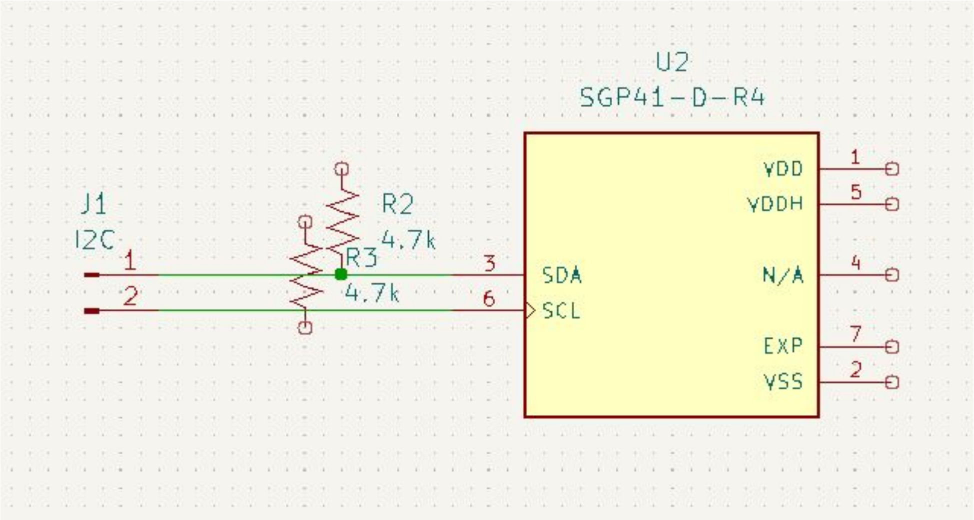

R1
4R7
C1
1u
J1
I2C
1
2
U2
SGP41-D-R4
VDD
VDDH
1
5
R2
4.7k
3
SDA
6
SCL
N/A
4
EXP
7
VSS
2
C2
1u
J2
PWR
1
2
R3
4.7k

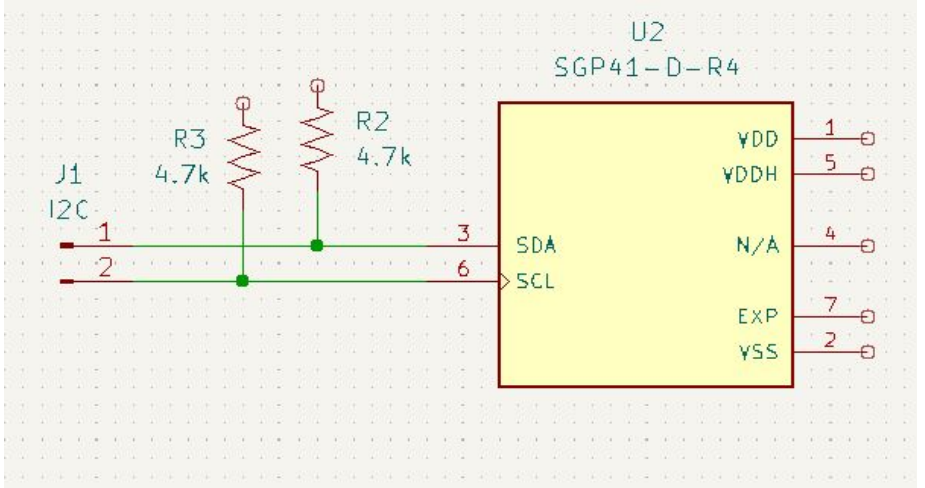
Schematic design

- Use component designators
  - Feel free to renumber starting at 1
- Add component values
- Label your connectors with something that makes sense
  - I2C vs Conn_01x_02Pin

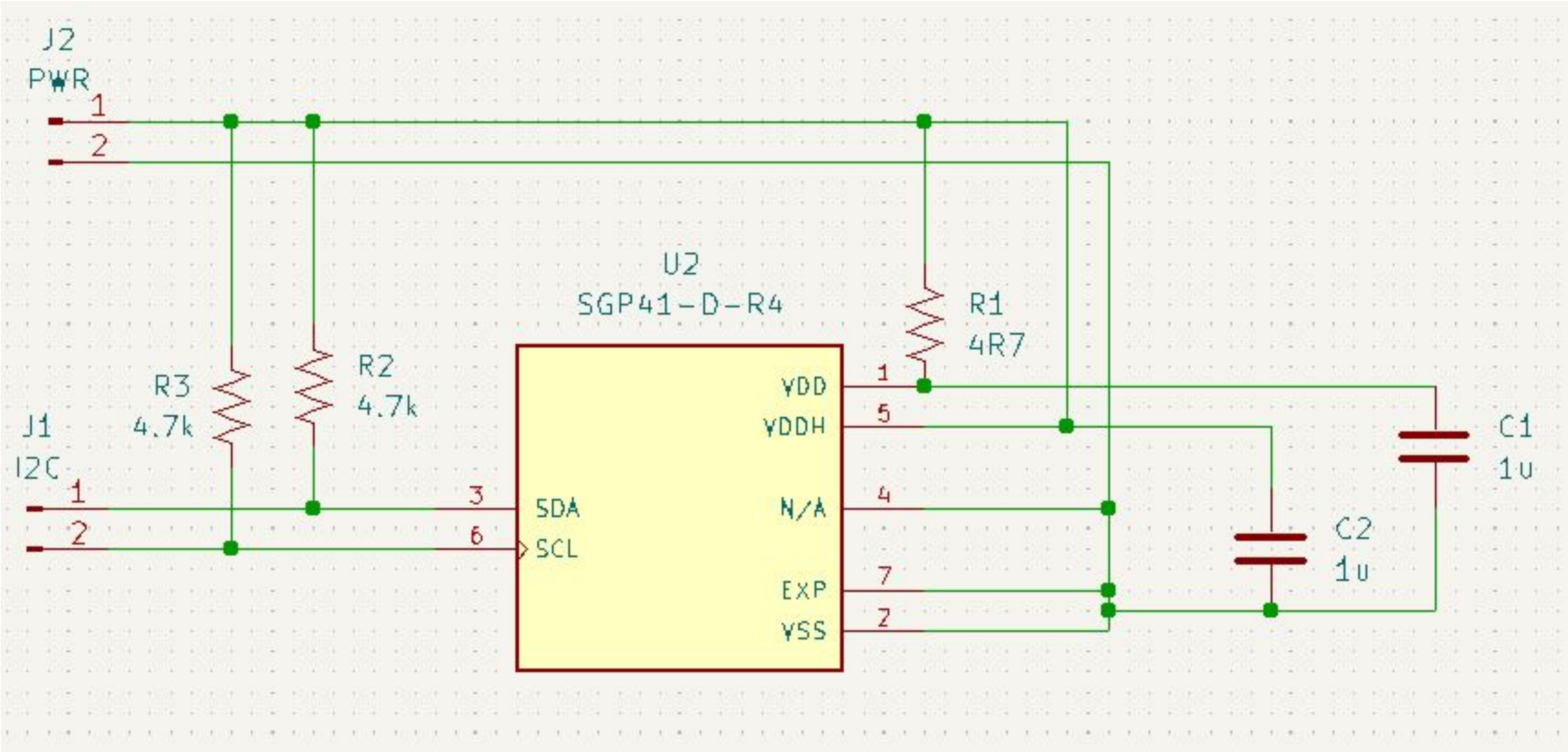




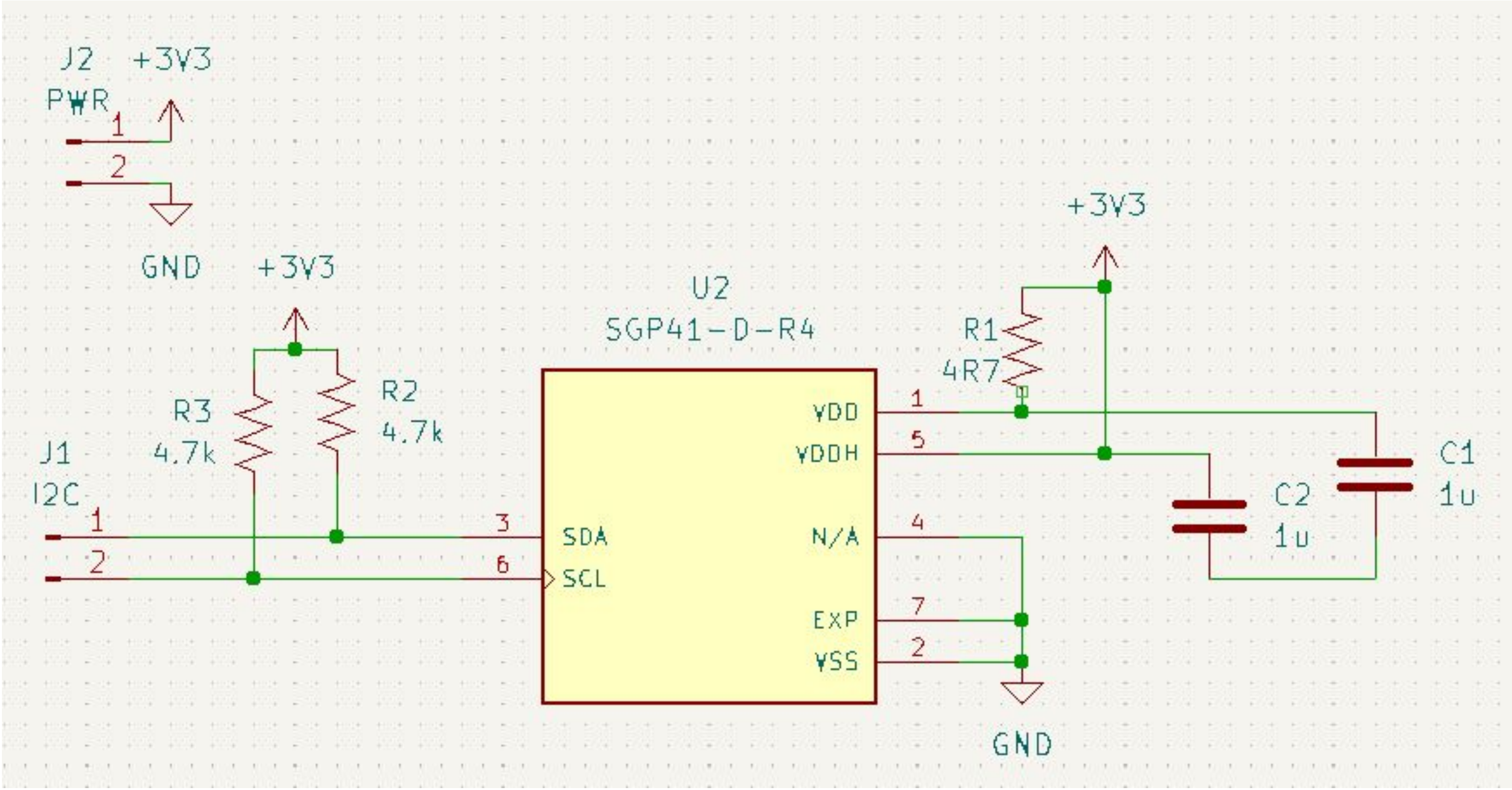




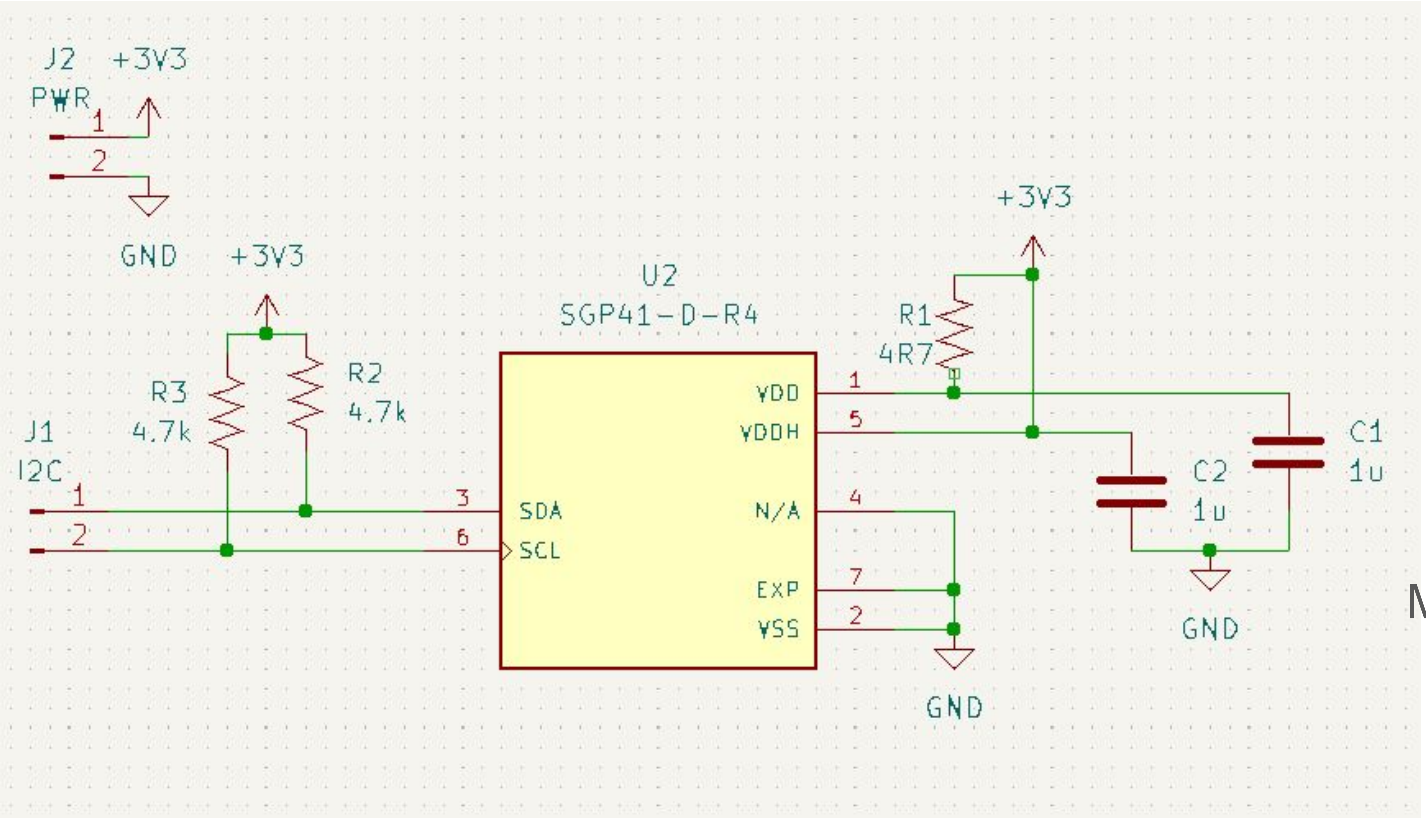
- R2 and R3 text is overlapping → hard to read
- R3 is not actually connected to anything



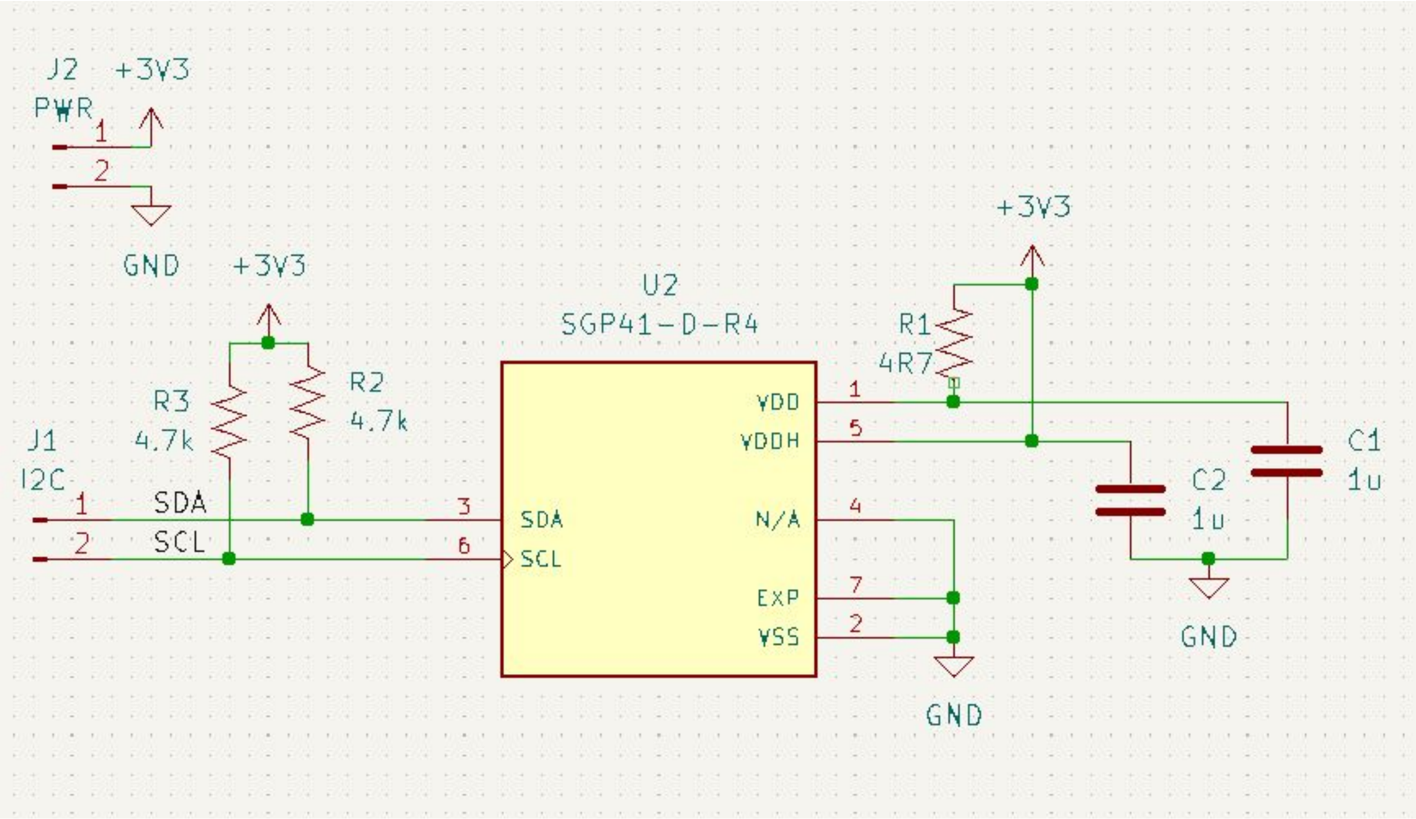
This is correct, but hard to read
Lots of overlapping wires
Hard to tell what's VDD, what's ground (VSS) by just looking



Better: Use GND and 3V3 power ports (P)
Fewer crossing wires, easier to read
Now I can see an error I made

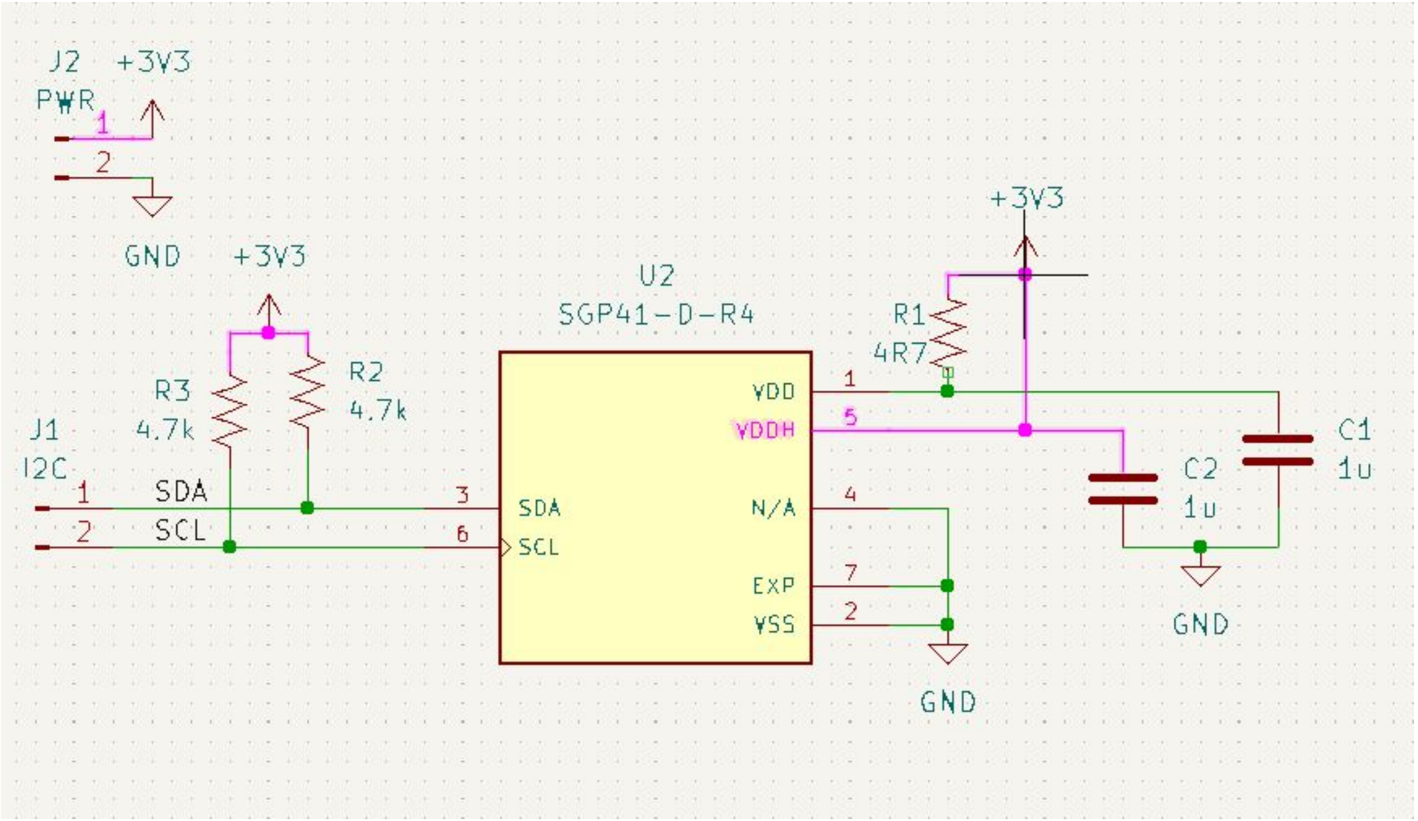


Missing a ground

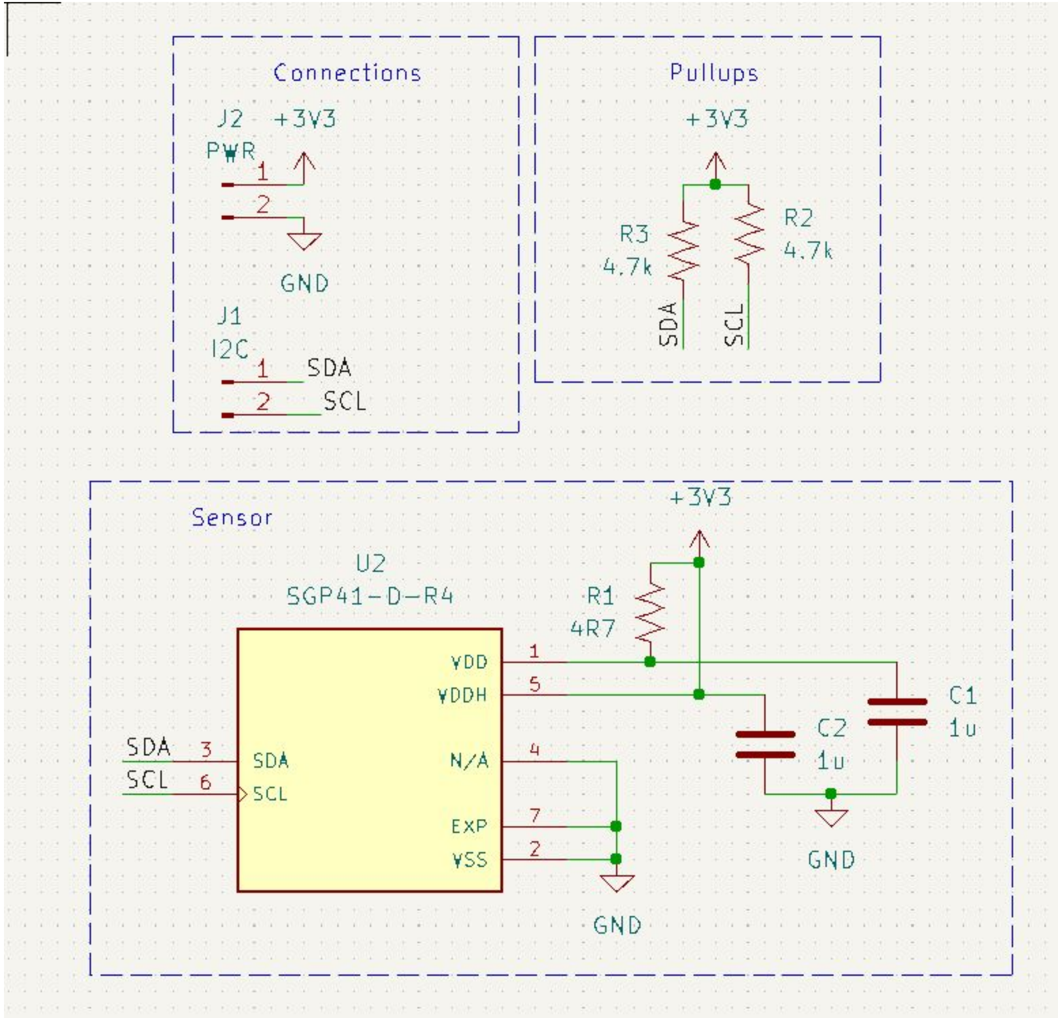


Assign SDA and SCL wires to SDA and SCL nets
This will be really useful when we lay out the PCB
Assign all important signals to nets

Now we can use



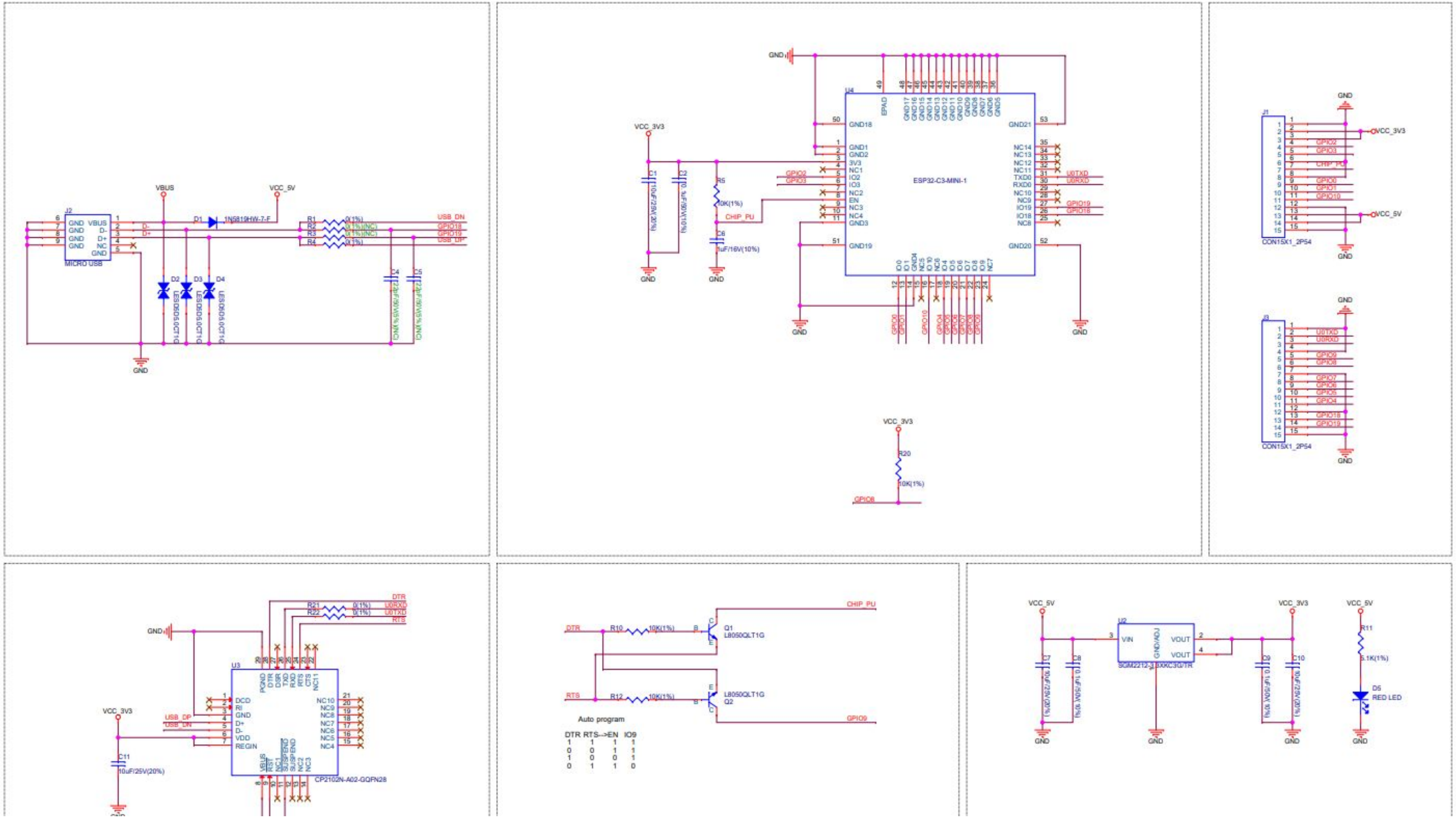
To highlight all wires of same net



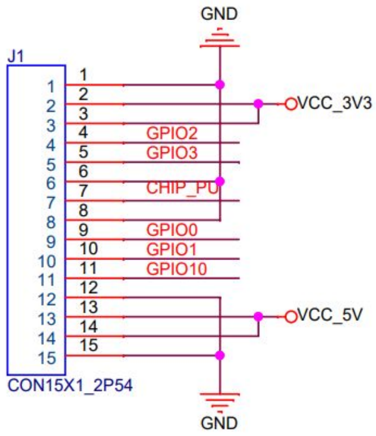
Too far?
I find this harder to read than the previous version

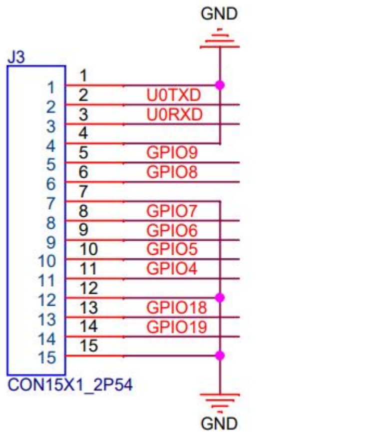
Balance use of direct connections with use of net labels

This is part of the schematic for our ESP32C3 dev board
See how they put different parts in different sections?



This is part of the schematic for our ESP32C3 dev board
See how they put different parts in different sections?
Net labels are useful for labeling the signals at all the connectors

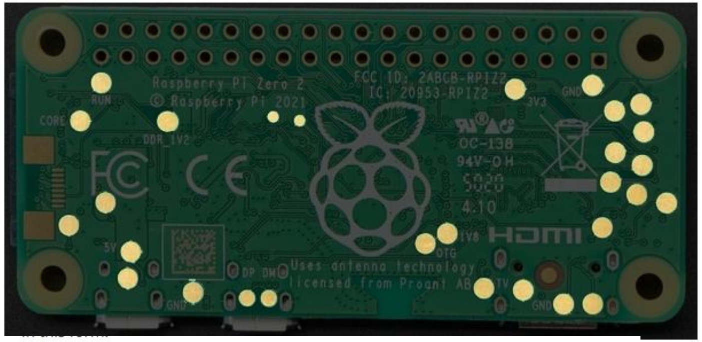




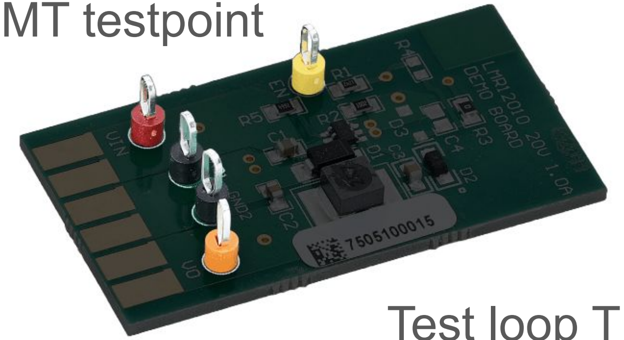
Don’t forget about debugging!

Here I’ve added 4 testpoints

Make sure every important signal has a TP

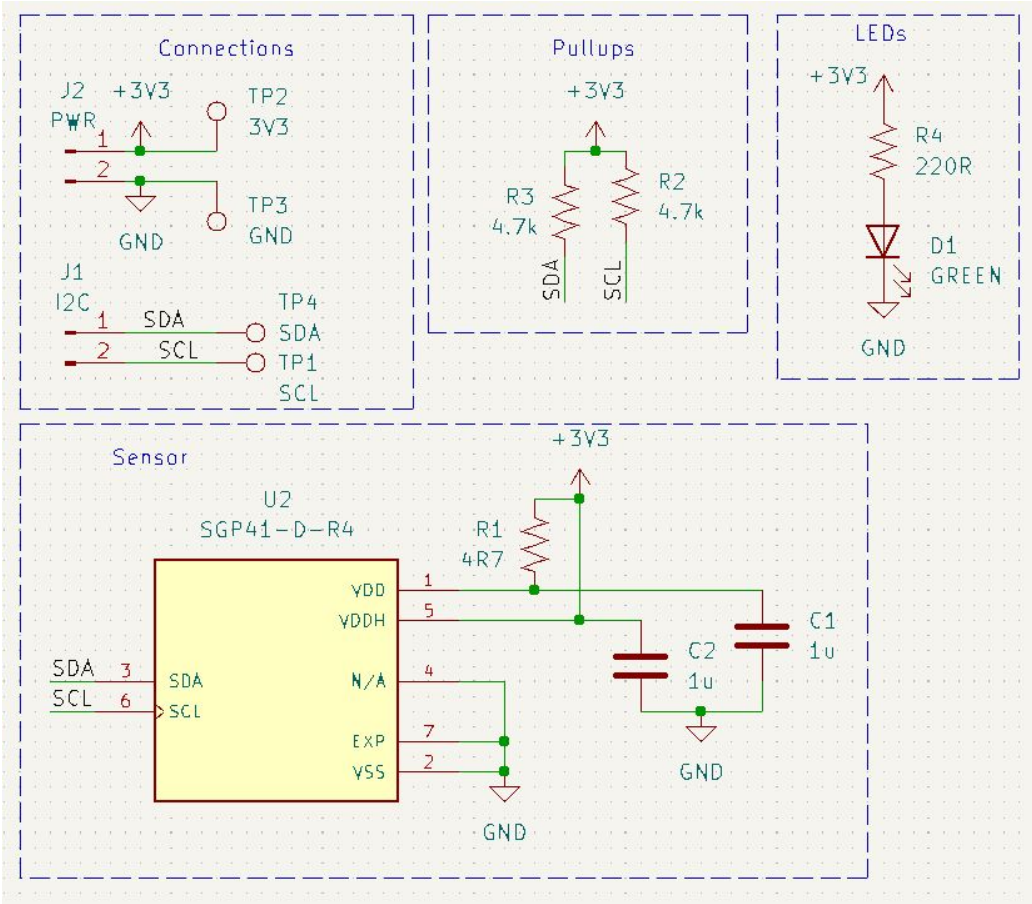


SMT testpoint



Test loop TP

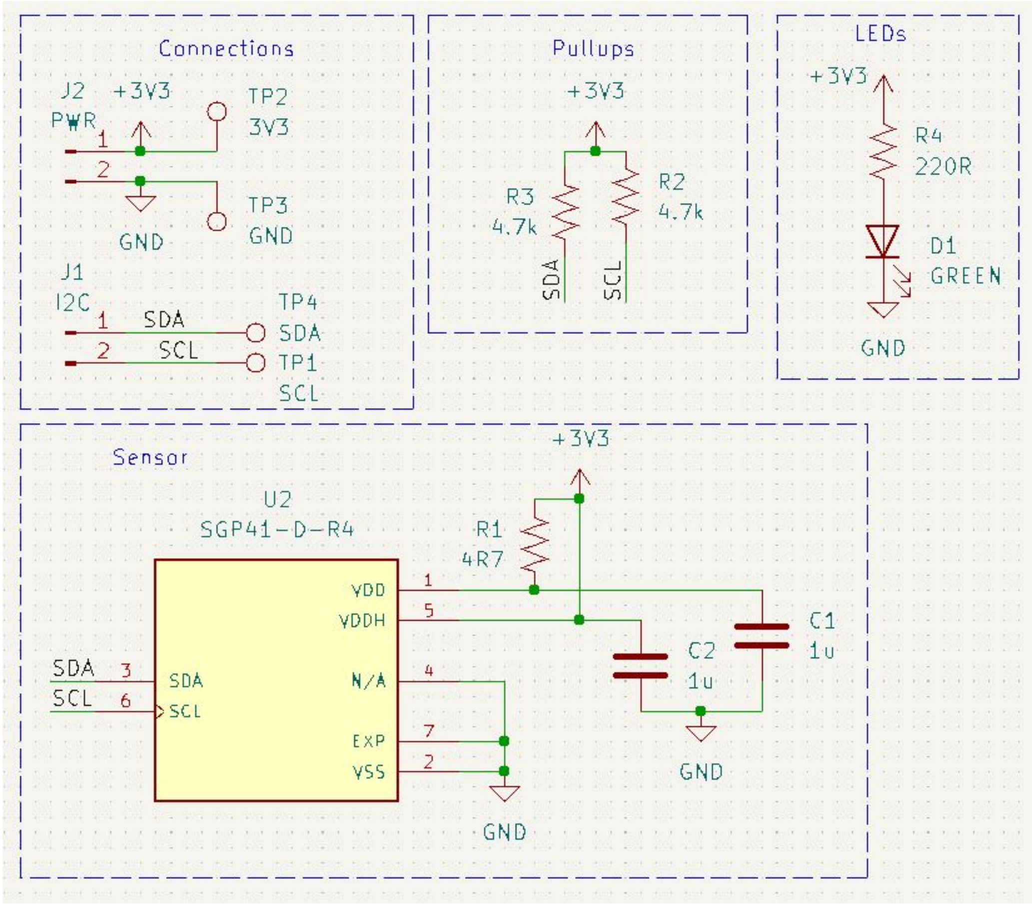
Image credit: TI, Editing credit: Winnie Szeto



Images from pcb.mit.edu

I also added an LED so we know if the board is even powered

You can add LEDs for other signals Just be aware they may cost you power, which can be annoying for battery-powered systems



You are not obligated to use any particular component or value

Example: many people put pull-up resistors on every board with I2C lines.

But your MCU may have internal pull-ups, or another board may have pull-ups on those lines already. You only need one set.

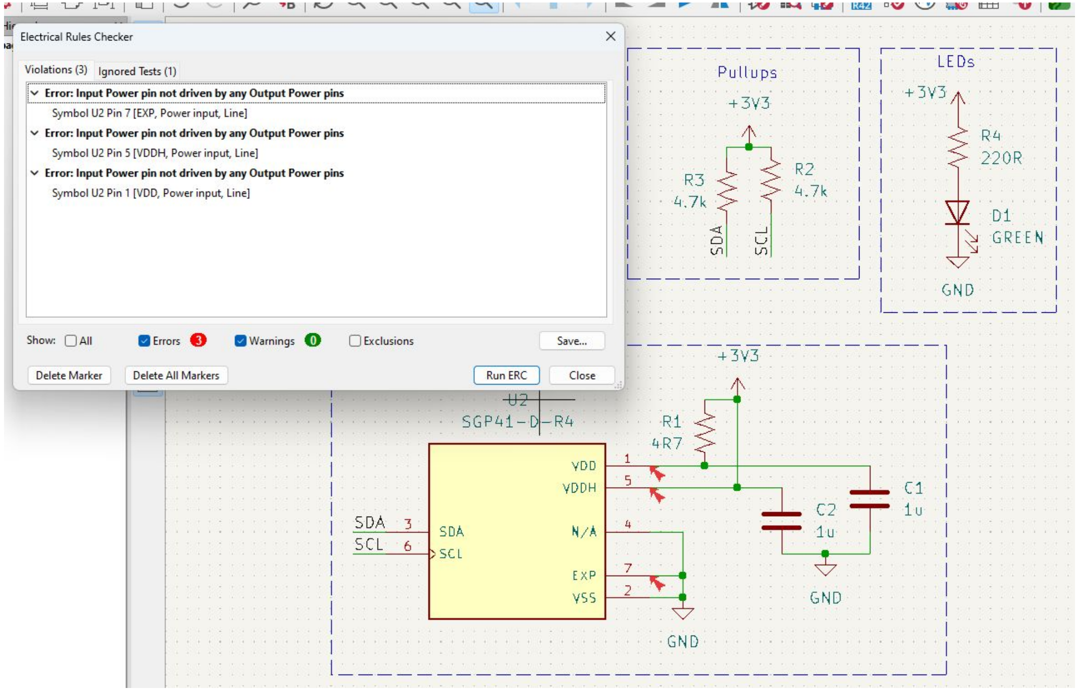
There is no problem to place components in the schematic and PCB but not install them in the actual board

You are not obligated to use any particular component or value

Example: You spec a 220R resistor for your LED. But then you decide to use a 470R resistor to reduce power consumption. That’s fine.

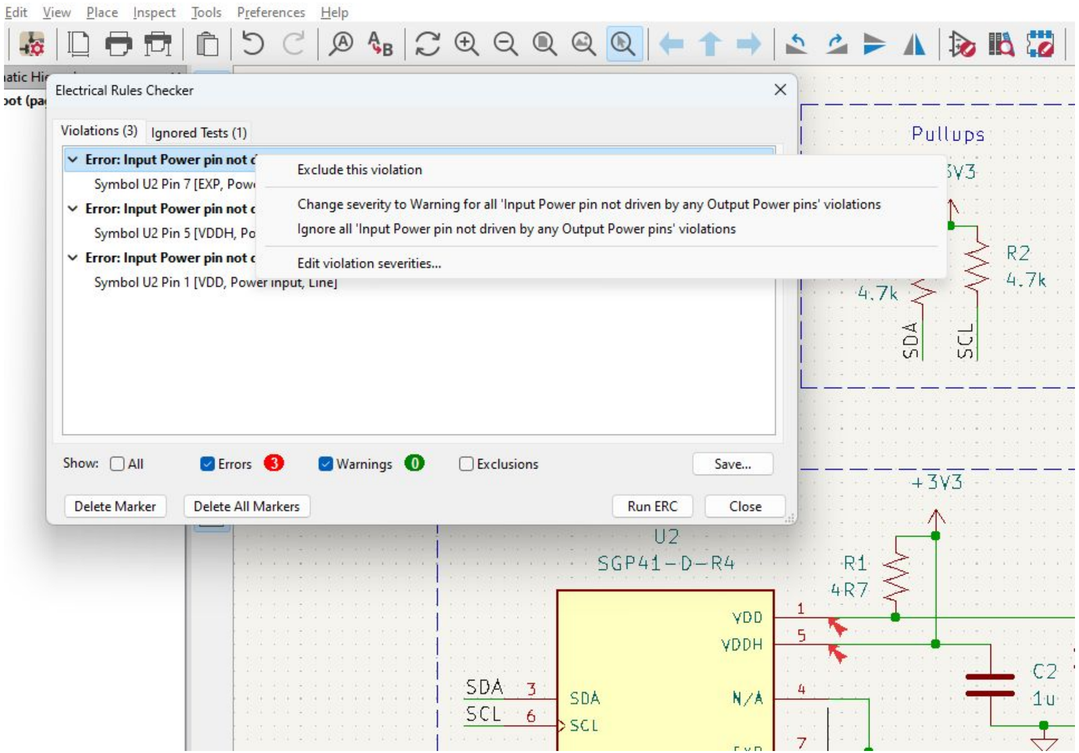
All that really matters is the footprint of the resistor on the PCB. If you place a 220R resistor with a 0805 footprint, you must use a 470R 0805 resistor.

That said, it’s good idea to have the correct resistor values in your schematic because you will refer to that when assembling your board.



Always run and pass your ERC

Here KiCAD is upset because my IC has power and ground pins and they don't seem to be connected to a voltage source



Always run and pass your ERC

Here KiCAD is upset because my IC has power and ground pins and they don’t seem to be connected to a voltage source

But we know they are, so this is one error we can “Ignore all…”