USC University of Southern California

EE 105: Introduction to Electrical Engineering

Lab 1.2

Circuit analysis with Python

Yasser Khan

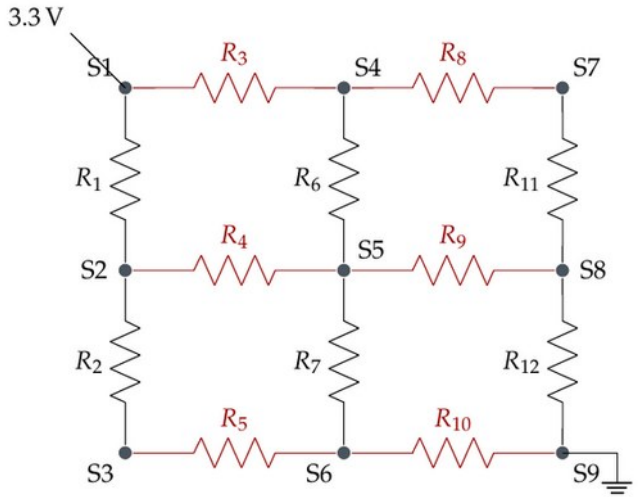
Rehan Kapadia

Today's mission

Write KCL and Ohm's law for the touch-panel grid, solve it two ways, and check it on the board.

By the end of this lab you can:

- Solve small circuits **exactly** with SymPy
- Write **KCL and Ohm's law** for a 3 x 3 resistor grid
- Check with **G v = i** in NumPy: it matches SymPy to 1e-15 V
- Check the predictions with **A0, a multimeter and a divider on A1**



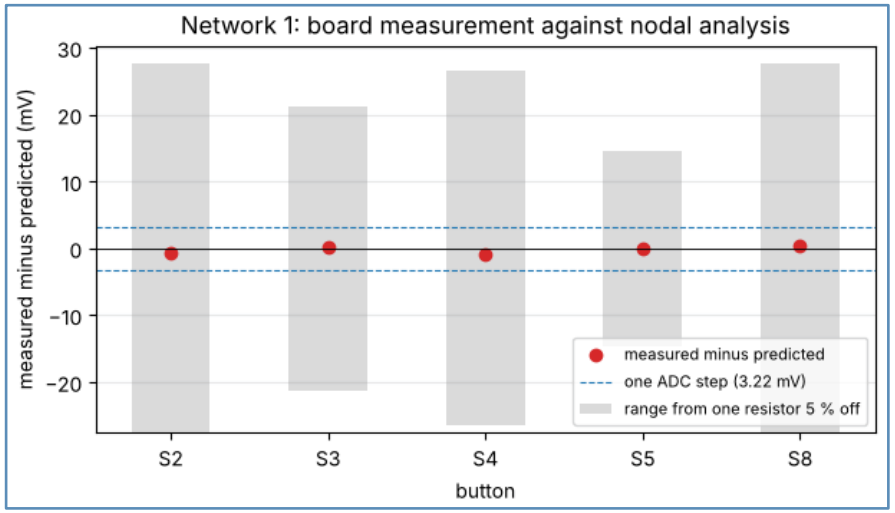
The grid: vertical 100 Ω; horizontal 50 Ω (Network 1) or 100 Ω (Network 2).

A	B	C	D	E	F
Setup	Warm up	KCL and Ohm	Solve	Power	On the board
5 min	**10 min**	**20 min**	**20 min**	**15 min**	**30 min**

Key takeaways

What to remember from this lab, with the numbers that matter.

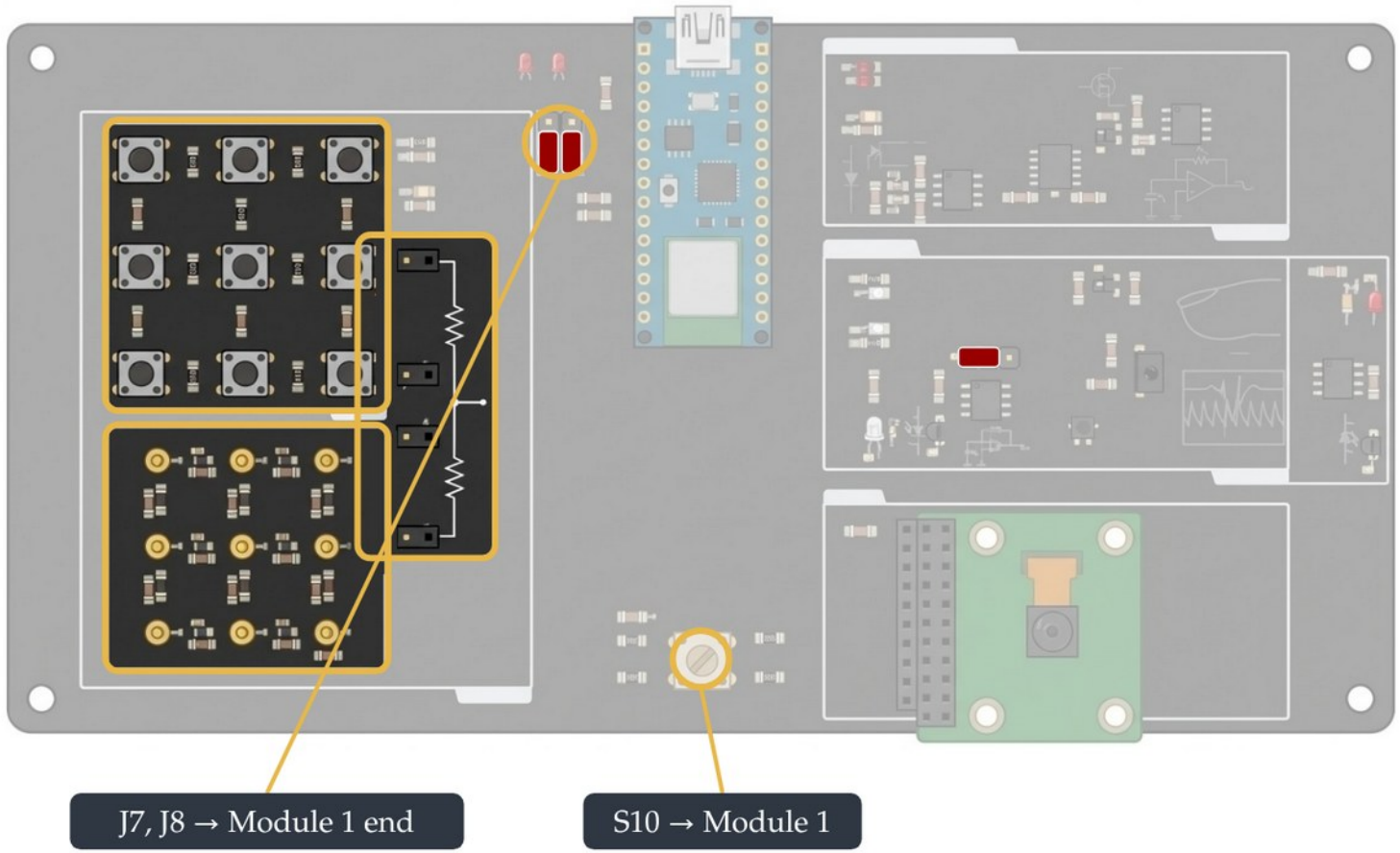
- **A circuit is a set of linear equations:** one KCL per node, one Ohm's law per resistor; 21 for the grid's 20 unknowns
- **SymPy is exact** (s2 = 253/130 V); NumPy's G v = i agrees to about 1e-15 V
- **G by inspection:** conductances at a node on the diagonal, minus the shared ones off it: symmetric
- **One resistor 5 % high moves a node up to 27.5 mV** (8.5 ADC counts): a real board is 10 to 20 mV off
- **The A1 count depends only on the resistor ratio:** divider and ADC share the 3.3 V rail; 2.245 V



Expected result of F1 on the simulated board: every button inside its gray tolerance bar.

SETUP

Set the board now; you need it again in Part F.



1. **S10 → Module 1**
   The circuits block: touch panel, Network 2 and the divider header.

2. **J7, J8 → Module 1 end**
   Pins 2-3: A0 reads the touch panel, A1 the divider header.

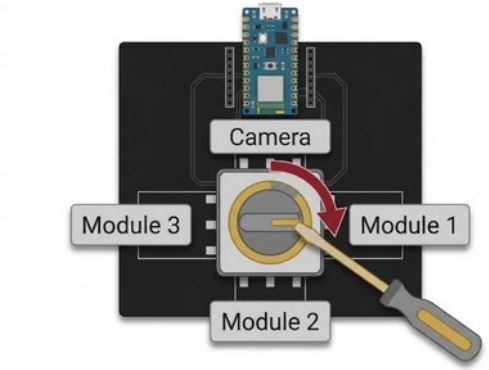
3. **F4: two resistors**
   4.7 kΩ (yellow, violet, red) in J1-J2, top; 10 kΩ (brown, black, orange) in J3-J4, bottom.

4. **F2: a multimeter**
   DC volts, 20 V range or auto.

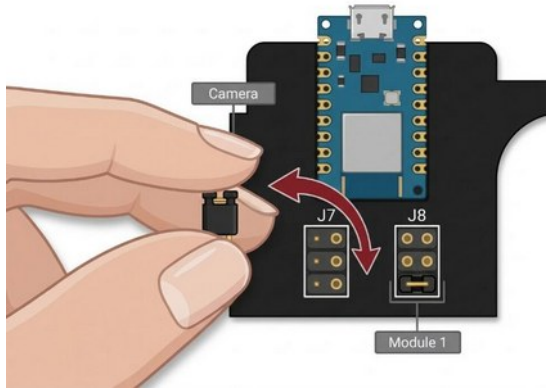
Part A: Setup

Set the board, plug it in, run the first cell. No board with you? Set SIMULATE = True.

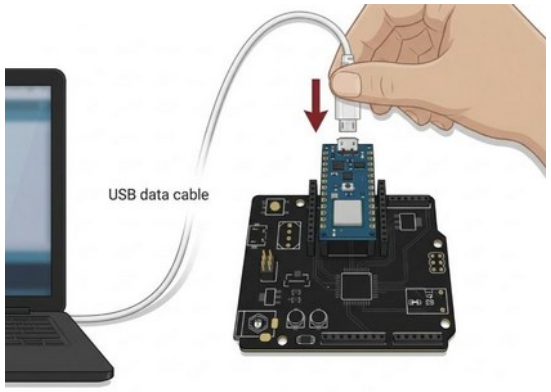
PART A **Goal: A1 · DO** connect to the board and check the selector.



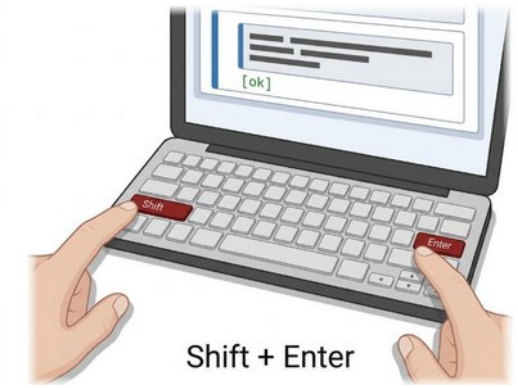
1 Turn S10 to Module 1



2 J7, J8 (3-pin headers): caps on pins 2-3, the Module 1 end



3 Plug the USB data cable into the Nano



4 Run the first cell with Shift+Enter

You should see: both lines start with **[ok]**. A **[!!]** line about the selector is only a warning. Parts B to E need no board.

Part B: Warm up with SymPy

USC University of Southern California

B1 · RUN solves this example: one equation per law (KVL, KCL, Ohm's law for each resistor).

```
V, R1, R2 = 5, 100, 200
I, V1, V2 = sp.symbols("I V1 V2")
sol = sp.solve([Eq(V, V1 + V2),    # KVL
                Eq(V1, I * R1),    # Ohm, R1
                Eq(V2, I * R2)],   # Ohm, R2
               [I, V1, V2], dict=True)[0]
show(sol)
```

B2 · EDIT

eqs_mixed: 10 V into 100 Ω in series with 200 Ω ‖ 300 Ω; use the node voltage Vm

Expected result · B1

```
I = 1/60 A = 16.67 mA
V1 = 5/3 V = 1.667 V
V2 = 10/3 V = 3.333 V
```

You should see: B1 **16.67 mA, 1.667 V, 3.333 V**. B2: **220 Ω**, 45.45 mA, Vm = 5.455 V.

Parts C and D: KCL, Ohm's law, SymPy, NumPy

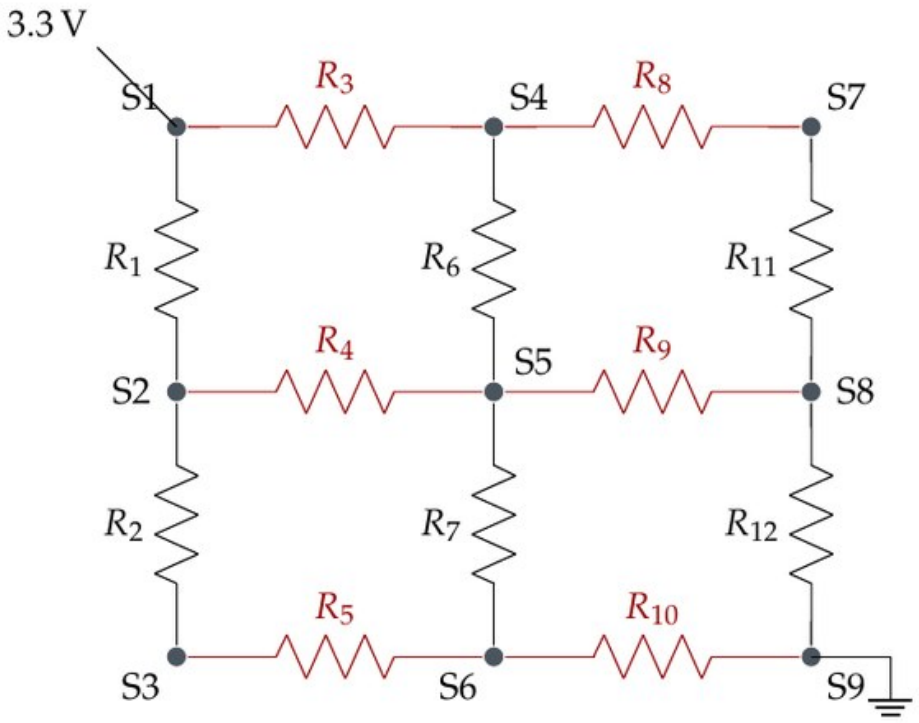
USC University of Southern California

Nine KCL equations and twelve Ohm's law equations: SymPy solves all 20 unknowns exactly.

PARTS C, D

Goal: write the grid's equations once, solve both networks, check with G v = i.

- **C1 · EDIT** KCL at all nine nodes, in = out (two are given)
- **C2 · EDIT** Complete **EDGES**: which two nodes each resistor joins
- **C3 · RUN** Ohm's law printout; the solvers
- **D1 · EDIT R_NET2** (all 100 Ω), then **solve_grid** for both networks
- **D2 · EDIT** G of Network 2 by hand; **np.linalg.solve**



Current in R_k is positive downward or to the right.

You should see: s2 = 253/130 = 1.946 V, s8 = 88/65 = 1.354 V; KCL at S5: **14.38 mA** in and out.

Check (Checkpoints 1, 2): the Ohm's law printout matches yours; exact solutions; NumPy agrees.

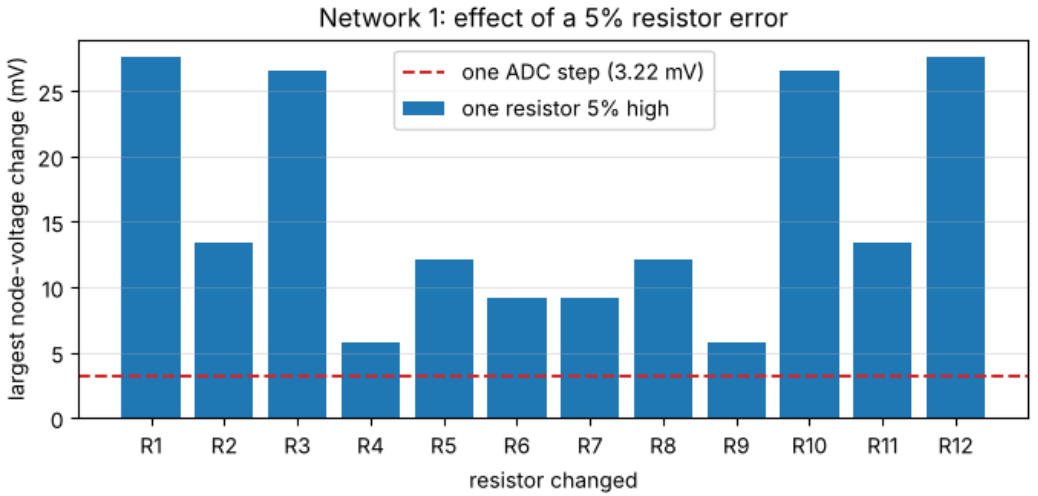
Part E: Power and resistor tolerance

The solver answers design questions: how hot, and how far off with 5 % parts?

PART E

Goal: find the power in every resistor and the effect of one 5 % resistor.

- **E1 · EDIT** P = I^2R for each resistor, and P = 3.3 V × i_t from the rail
- **E2 · EDIT** Each resistor TOL high in turn: 5 %, then TOL = 0.01
- **E3 · RUN** The tolerance plot



Expected E3: the four corner resistors move a node most; every bar is above one ADC step.

You should see: Rail **97.7 mW**; R1 and R12 hottest at **18.3 mW**. 5 %: up to **27.5 mV**; 1 %: **5.6 mV**.

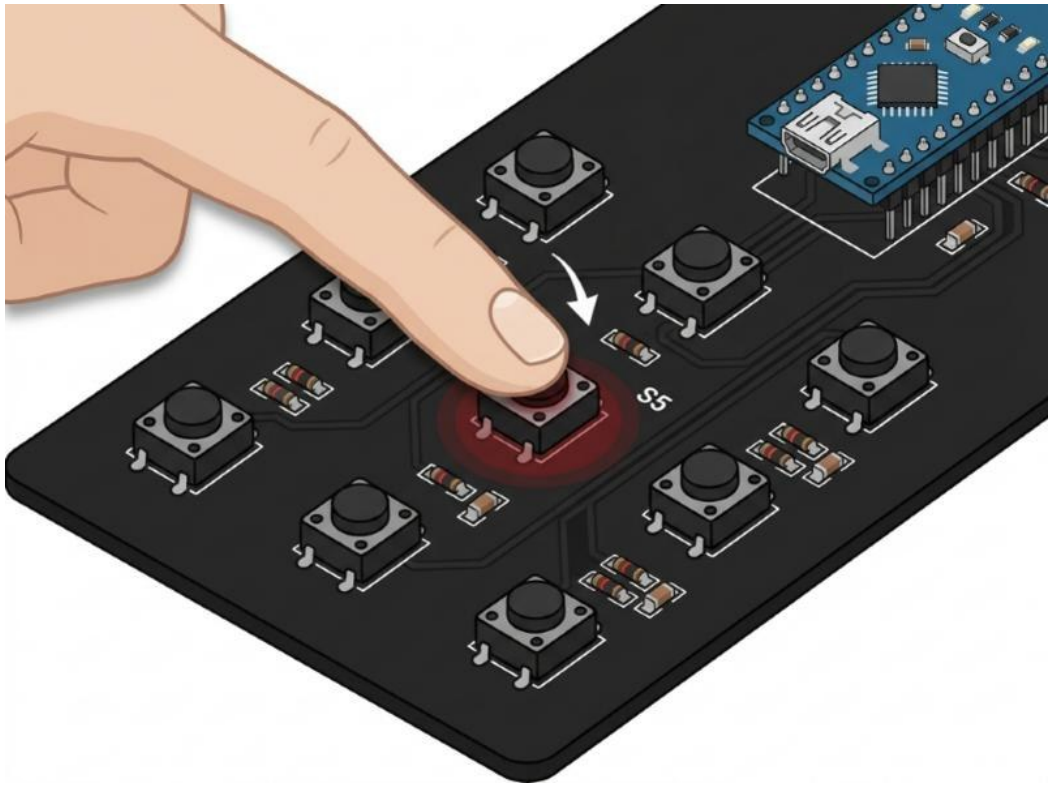
Part F: Network 1 on the board

Your solution predicts the voltage each button puts on A0. Test it.

PART F

Goal: measure five buttons and compare them with your prediction.

- **F1 · DO** Run the cell: it asks for **S2, S3, S4, S5 and S8** in turn
- **Hold the button down, then press Enter**; keep holding until it prints
- The cell then draws the differences; the gray bars show 5 % tolerance



S2/S4 and S3/S8 are the pairs to tell apart.

You should see: measured minus predicted within about **10 to 20 mV** on a real board: inside the gray tolerance bars. Your Lab 1.1 readings appear as an extra column.

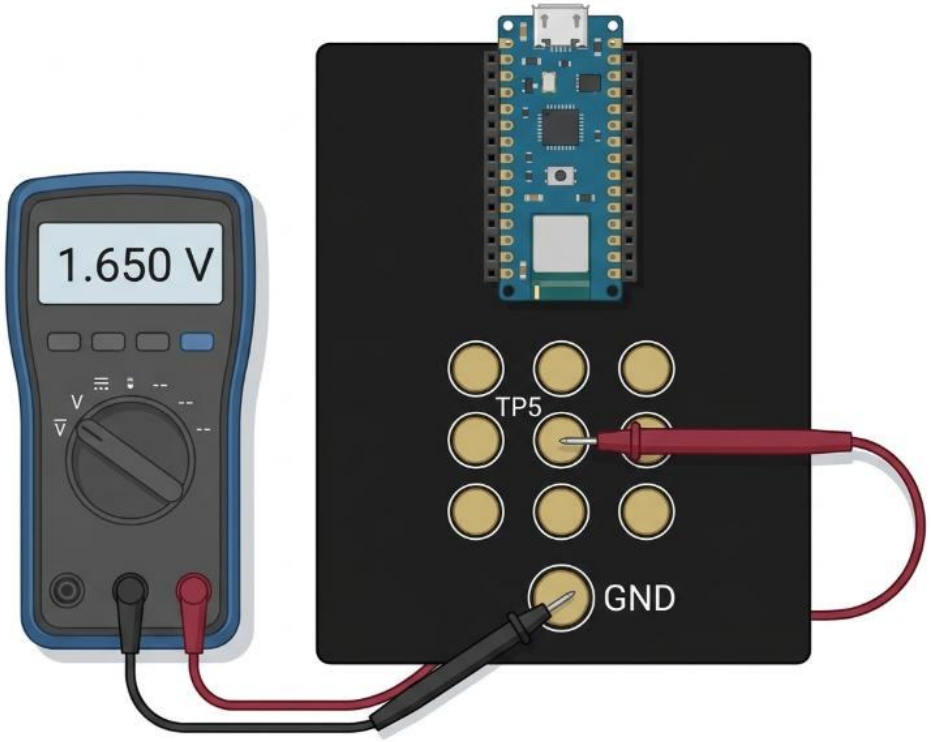
Part F: Network 2 with a multimeter

Network 2 is symmetric: mirror-image nodes read the same voltage.

PART F

Goal: measure TP1 to TP9 with a multimeter and compare with your Network 2 solution.

- Set the meter to **DC volts** (20 V range or auto)
- Black probe on **TP9** (ground), red probe on each test point in turn
- **F2 · ANSWER** Type the nine readings into **tp_measured**
- Which test points read the same, and why? (Question 5)



TP1 to TP9 run row by row; TP1 on 3.3 V, TP9 on ground.

You should see: TP1 to TP9: **3.3, 2.2, 1.65, 2.2, 1.65, 1.1, 1.65, 1.1, 0 V**.

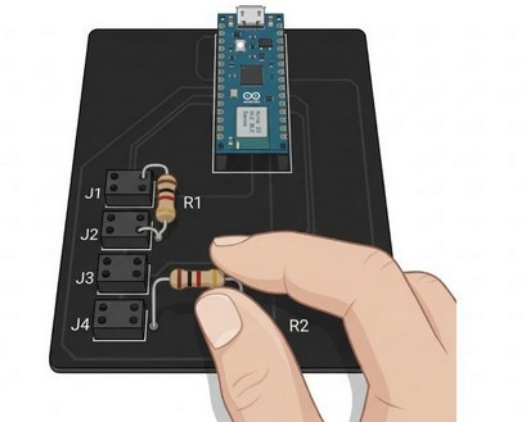
Part F: A voltage divider on A1

Two resistors in the divider header; the midpoint reaches A1 through J8.

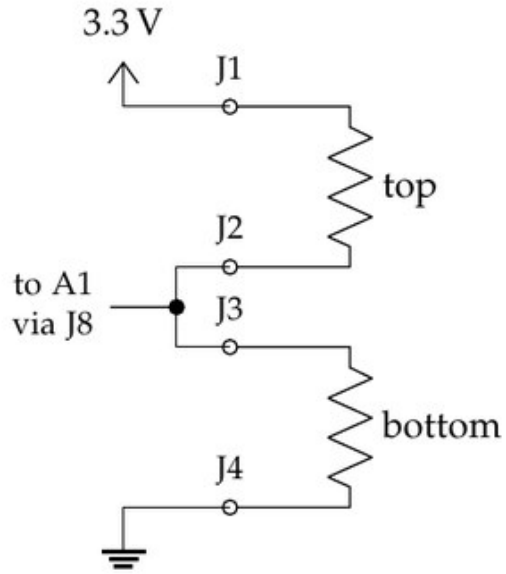
PART F

Goal: predict the divider voltage, measure it, and infer the top resistor.

- **F3 · EDIT** Predict V_A1 = 3.3 V × R_bottom / (R_top + R_bottom)
- **F4 · DO 4.7 kΩ** in J1-J2 (top), **10 kΩ** in J3-J4 (bottom); read A1 1000 times
- **F5 · EDIT** R_top = R_bottom (3.3 V / V_A1 - 1): compare with the color bands



R1 top (J1-J2), R2 bottom (J3-J4)



Midpoint J2-J3 to A1 via J8

You should see: predicted **2.245 V**; the inferred top resistor within about **5 %** of 4.7 kΩ.

Check (Checkpoint 3): board and meter within a few tens of mV of your solutions, the divider close to your prediction.

Stuck?

Check these first. If the fix does not help, raise your hand.

solve_grid: No unique solution

A mistake in your KCL list (C1) or EDGES (C2): check each node and resistor.

Run All stops at F1

The cell waits for you: hold the named button, then press Enter.

One button is more than 30 mV off

Released too early: rerun F1 and hold until it prints.

The multimeter reads 0 V or jumps

DC volts; black probe on TP9; press the red probe on the test pad.

A1 reads 0 V or 3.3 V

Seat both resistors; J8 cap on the Module 1 end; S10 on Module 1.

No board with you

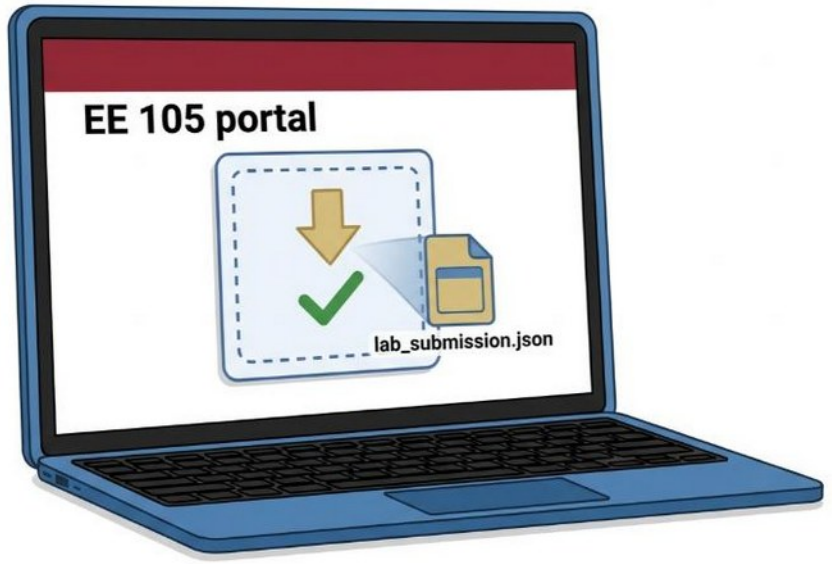
SIMULATE = True runs Part F on a simulated board, for practice.

Submit your results

The last cell saves your results in one file. Upload it on the course portal.

`lab1.2_submission.json`

1. **Save · RUN** writes **lab1.2_submission.json** next to the notebook
2. Log in at **ee105.org/portal** with your student ID and portal PIN; open Lab 1.2
3. Upload the file. It is graded automatically; you can upload again after a fix
4. Keep **grid_solution.csv** and the notebook for your report



ee105.org/portal

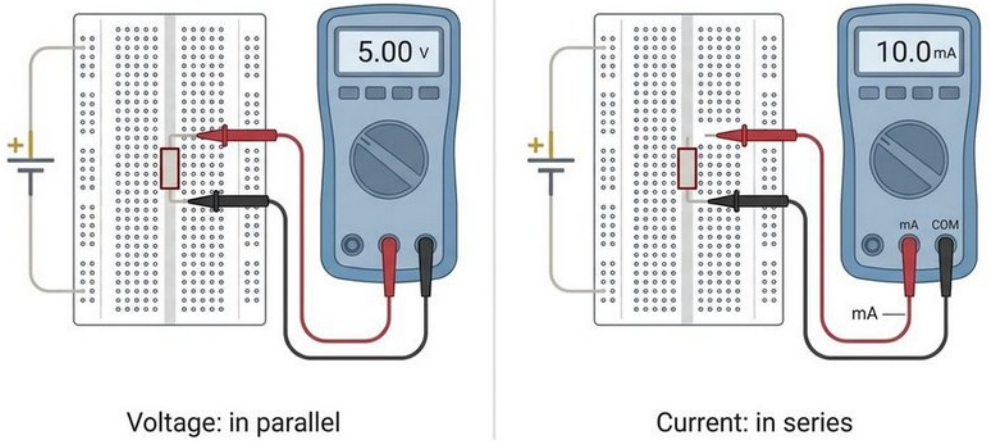
Check: the portal shows your grade. A file from a simulated run is practice only and earns no credit.

Lab complete

You solved a 20-unknown circuit two ways and checked it with three instruments.

You built:

- Exact SymPy solutions for both touch-panel networks
- A nodal G v = i solver, built resistor by resistor
- A power and 5 % tolerance study
- Board, multimeter and divider checks of your predictions



A voltmeter goes in parallel with what it measures.

Did you know?

Circuit simulators such as SPICE work like your nodal_system: they build G from the netlist one resistor at a time, then solve (Chapter 1).

Next

Lab 2.1, diode and transistor I-V characteristics: sweep a diode and a MOSFET (S10 on Module 2).