Magnetic Generator — Student Build Guide v3.9.2

DEDICATION: To the memory of Andrei Slobodian. This work is dedicated to documenting and preserving his published mechanical concepts. By translating his designs into a reproducible, open-source format, we ensure that his engineering ideas can be rigorously studied, tested, and understood by the world forever.

ACKNOWLEDGMENT: Defense through Disclosure. This project adheres to the principle that the strongest protection for controversial technology is total transparency. We believe that by making knowledge open-source, we ensure that engineering progress—once shared—can never be suppressed.

Introduction

This project is a hands-on effort to replicate and **rigorously test** the physical design and control ideas shown in these three reference videos.

- **Core teardown / mechanical inspiration:** https://youtu.be/rwIEThO5VBw
- **Example of others attempting similar builds:** https://youtu.be/_Z3kQ3HMvuk_
- **Key control concept ("timed push" logic):** https://youtu.be/mUOwgixqWfY

The build is a 55-magnet / 55-coil drum-style machine (5 rows × 11 columns). The core idea being tested is that the device is **not** operated like a conventional motor with continuous commutation. Instead, it is treated as a pulse-timed mechanical oscillator: the controller uses Hall or encoder feedback to apply short, precisely phased "push" pulses to help the rotor pass magnetic center, while the remaining coils are routed for continuous harvesting through rectification and direct current conversion.

The project goal is to reproduce the geometry and switching behavior as faithfully as possible, then evaluate performance using honest accounting—measuring both pulse energy input and harvested output—through clear, repeatable test procedures.

As of version 3.9.2 of the project, this open-source build remains in the design and experimental phase. The intent of this work is strictly educational and investigative. No claims are made regarding verified performance, sustained operation, net energy gain, or “over-unity” behavior. The system is designed solely as an experimental platform to study geometry, timing, magnetic interaction, and control strategies under controlled conditions. Any conclusions must be based on careful measurement, transparent accounting, and repeatable testing.

Release date: January 25, 2026 | Version: v3.9.2

What changed since v3.9.1

- **CRITICAL SAFETY:** PUSH COILS CHANGED TO SERIES-AIDING. Parallel-aiding with 18 AWG wire draws >180 A, which destroys the MOSFETs.
- **ADDED SECTION 3.5:** Fabrication recipe for 80-turn bifilar coils and measured resistance data.
- **Meters:** Shunts specified low-side; Meter #1 measures external input only; Meter #2 measures harvest on return; wording clarifies gross vs net interpretation.
- **BQ25570:** Protected low-voltage tap/clamp required; never connect it to a rail that can rise above 5.5 V.
- **One-hot wiring:** Coil negatives route to Row Trunk (row MOSFET drain). No coil negatives go directly to Star-Ground.
- **High-side P-MOSFET gate safety:** Added 1–2.2 kΩ series current limiter in the gate-sink path (zener clamp protection) plus a small gate stopper near the MOSFET.
- **Timing/physics:** Commissioning timing is ATDC for North-North repulsion; encourages phase sweep because Hall edges are threshold crossings (not guaranteed geometric center).
- **CAD:** Coil-module M3 holes updated to 3.2 mm for print clearance.
- **CABLE LABELING STANDARD (v3.8.1):** Each wire now labeled as {Column}{Row}{Winding}_{End}. Example: Coil A1 has leads A1A_S, A1A_F, A1B_S, A1B_F. Total 220 uniquely labeled wires across 55 coils.
- Added various figures (v3.8.2).
- Added various sections, the Swing analogy, expanded where everything lives on the backplate, and added 10 new figures.
- In v3.8.7 added Visual Electronic Parts Guide (Quantities Required).
- **Version 3.8.8** added a mandatory **Mechanical Assembly** section (9.3) to ensure MOSFETs are mounted and isolated before soldering. It also replaces the confusing visuals with separate **Side** and **Top** view diagrams to clearly show pin bending and wiring paths.

- Version 3.8.9 added a mandatory **Mechanical Assembly** section to ensure proper component isolation, enforces **twisted-pair wiring** to prevent magnetic noise, and updates the **Commissioning** protocols to probe the correct switching nodes.
- Version 3.8.9 introduced **standardized "Switch Node" terminology** to prevent wiring errors and updates **Figure 8-4** with a fully corrected "double-view" protocol, ensuring safety for the logic tap assembly.
- In Version 3.9.0, we created the missing **Section 9.6 (High-Side P-MOSFET Assembly)**, standardizing the "Push Columns" with the same bent-pin safety architecture and twisted-pair wiring protocols used for the Low-Side switches
- In Version 3.9.1, we updated **Sections 10.1 and 10.2** to strictly enforce the 9-bridge layout (removing the "ghost" unit), defined the solid bus bar material requirements, and added the complete mounting and wiring protocols for the Power Distribution and Regulation modules.

Locked items (change only with a new revision)

- **Naming:** Columns A–K and Rows 1–5 (bottom→top). Coil positives labeled +A1…+K5 (positives only on DIN). Cable labeling: {Col}{Row}{Winding}_{End} (e.g., A1A_S, A1B_F).

- **Polarity:** Rotor magnets are all North-out; coils are wired so the rotor-facing face energizes North during a push pulse (North-North).

- **Control:** One-hot intersection firing (only one physical coil pulsed at a time): selected Column high-side + selected Row low-side.

- **Geometry baseline:** Rotor outside diameter 96 mm; stator inside diameter 100 mm; magnets 18 mm × 3 mm stacked to ~6 mm.

- **Air gap:** CAD lip gap is baseline; commissioning targets ~1.0 mm working gap per side using shims/coil seating (measure and record).

- **Metrology:** Log external input, harvest measurements, and scope-based pulse energy (∫v·i dt) before conclusions.

Project status and next step

- **Current status:** Phase 0 complete (printed mechanical parts and baseline inventory in hand; Hall sensing validated on a simple breadboard test).

- **Next step: Phase 1 commissioning (minimum path)**
  - Build rigid, adjustable Hall mount with strain relief; lock a sensor gap and verify clean transitions between magnets.

  - Wire and validate one known-good coil path (one column + one row). Start with bench supply current limit (for example, 2 A).

- Confirm meters and the BQ25570 tap/clamp wiring with power OFF first, then under controlled low-current tests.
- Perform a phase sweep around estimated center to find forward torque (ATDC for repulsion), and log current + pulse energy.

---

0) Read this first (plain-language safety)

- Keep fingers, sleeves, and tools **away from the rotor**. Add a clear shield before high-speed tests.
- Wear **eye protection**. Magnets are brittle; epoxied parts can come off.
- Use **fuses** on every power feed. Start small and step up.
- Treat the supercap and bus bars like a car battery: low voltage, **very high current**. A short can melt metal.
- **Speed limit:** Never exceed 3500 RPM. The firmware includes an overspeed cutoff (>3500 RPM for 0.1 s → disable pulses for 5 s), but this is a backup—if the rotor sounds like it's accelerating uncontrollably, kill external power immediately.
- At high speed, a broken magnet or delaminated epoxy becomes a projectile.

1) What we're building

A 5×11 coil machine (5 rows, 11 columns). The rotor has **11 magnet columns**, all magnets facing **North-out**.

- We use **short push pulses** on two opposite-ish columns (**A and F**) to spin up.
- Other columns are wired to **harvest** electricity.
- At startup, a single hand-flick plus correctly timed push pulses should accelerate the rotor.
- As speed rises, the harvest voltage and available power also rise.

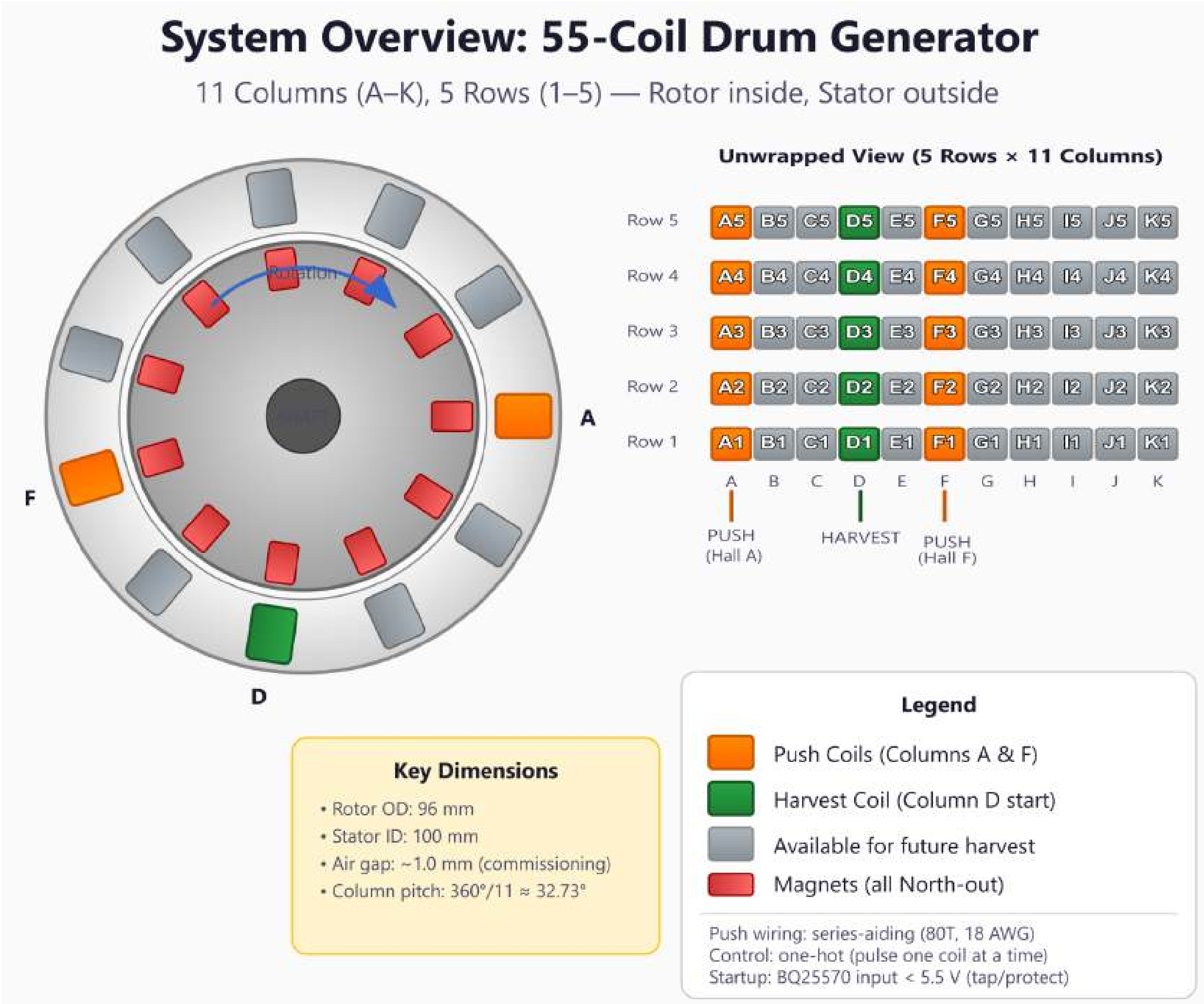
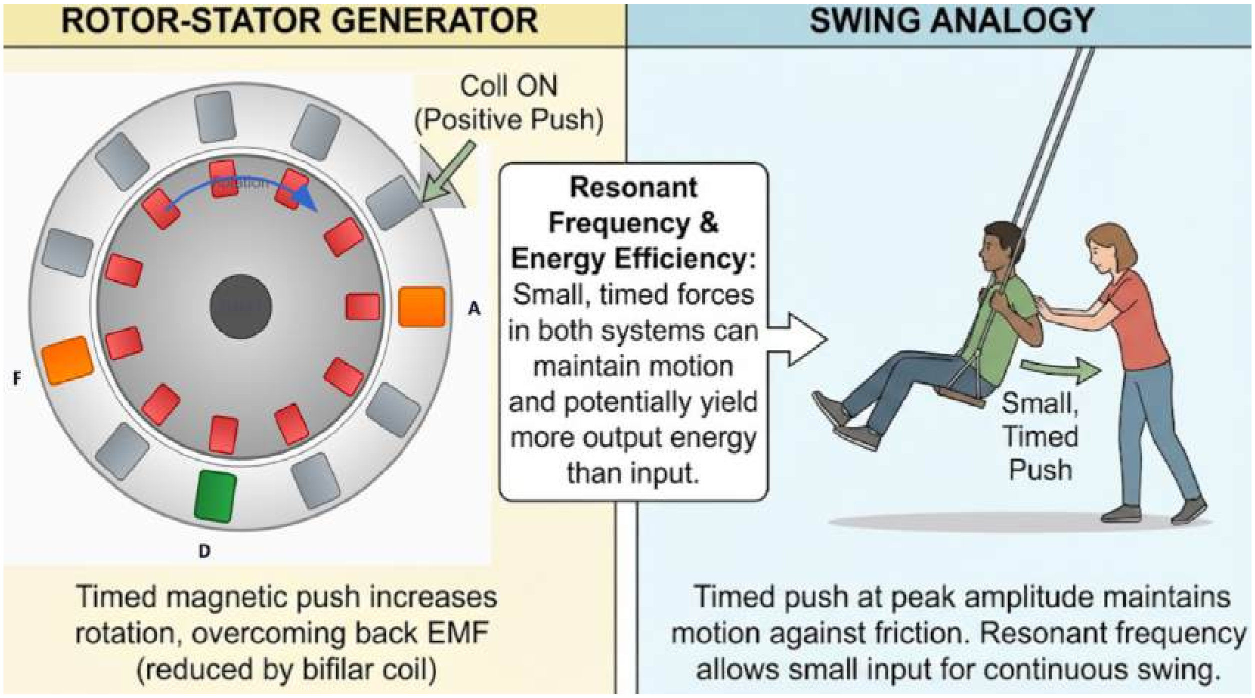


Figure 1-1 — System overview (high-level block diagram).

The "Swing" Analogy: How Your Generator Works:



To understand how we hope to get maximum efficiency from this generator, think of it like pushing a child on a playground swing.

1. The Swing analogy (The Rotor & Resonance)

- **The Analogy:** Your rotor is the swing. It has weight (mass). Once you get it moving, it wants to keep moving because of momentum.
- **The "Perfect Rhythm":** Every swing has a natural rhythm. If you push at the exact right millisecond (just as the swing stops at the top of its arc), a tiny push keeps it going high.
- **The Generator:** Your rotor has a rhythm too. By using the Arduino to fire the coil at the exact right moment (just after the magnet passes center), we tap into the rotor's momentum rather than fighting it.

2. The Friction (Lenz's Law)

- **The Analogy:** Imagine trying to swing with rusty chains or in a strong wind. This friction fights you. No matter how perfectly you time your push, the swing slows down because the friction steals your energy.
- **The Generator:** In physics, this is **Lenz's Law**. As you generate electricity, the magnetic field naturally fights back, creating "magnetic drag." It feels like trying to run through water.

3. The "Oil" (Bifilar Coils)

- **The Analogy:** Your special coils act like **oiling the chains**. They don't remove gravity, but they reduce the friction so the swing glides freely.

- **The Generator:** The "Bifilar" winding creates a special internal electrical environment (Capacitance) that absorbs the "magnetic drag" momentarily. Instead of the drag fighting the rotor immediately, it is stored and delayed. This allows the rotor to glide past the coil with less resistance.

Deep Dive: Your 80-Turn Bifilar Coils

We calculated the specific behavior of your coils based on the **Series-Aiding** wiring, **Ferrite Core**, and **18 AWG wire**. Here is what is happening inside the copper.

A. Inductance: The "Shock Absorber"

- **The Value: ~5 mH (Millihenries)**.
- **The Concept:** Inductance is like the weight of water in a pipe. It creates inertia.
- **Why we need it:** Your wire is thick and has very low resistance (0.27Ω). Without this "magnetic weight," the electricity would rush in too fast (40+ Amps!) and blow your fuse instantly. The Inductance forces the current to ramp up slowly, keeping the system safe during your short pulses.

B. Capacitance: The "Ghost Tank"

- **The Concept:** Even though you didn't solder a capacitor inside the coil, the **Bifilar** winding creates one. Because the wires run side-by-side but carry different voltages, they act like plates of a capacitor.
- **The Tesla Effect:** Nikola Tesla discovered that connecting coils this way (Series-Aiding) drastically increases this storage ability.
- **The Function:** This "Ghost Tank" catches the energy that would normally be wasted as "drag" or heat. It holds it for a split second, smoothing out the system.

C. Resonance: The "Ring"

- **The Frequency: ~100 kHz (100,000 times a second)**.
- **The Concept:** When you stop pushing a swing, it vibrates. When you turn off your coil, it "rings" electrically.
- **The Benefit:** Because of the Ferrite core and the Bifilar winding, your coil rings at a very high pitch (ultrasonic). This ringing represents energy sloshing back and forth inside the coil. Our harvest circuit (the MUR460 diode) catches this sloshing energy and puts it into the battery, recycling what other generators waste.

The Evidence: Why We Believe This Works

This hypothesis is not magic; it is based on documented engineering principles. Here are the sources supporting your design.

1. Evidence: "Neutralizing" the Drag (Tesla’s Patent)

- **The Claim:** A specific winding style can cancel out the coil's own resistance to current (Self-Induction).

- **The Source: Nikola Tesla, U.S. Patent 512,340 ("Coil for Electro-Magnets")**.
- **The quote:** Tesla explicitly states that this winding "gives the coil as a whole a capacity sufficient to neutralize its self-induction."
- **What it means for us:** By using Tesla's Series-Aiding winding, we reduce the "choke" effect, allowing us to pulse the coil faster and cleaner.

2. Evidence: High-Efficiency Energy Storage (LC Resonance)

- **The Claim:** Series-bifilar coils store significantly more energy in their electric field than standard coils.
- **The Source: Vishnuram & Ramachandran (2020)**, Modeling and Analysis of Bifilar Coil for Induction Heating Applications.
- **The Finding:** Their math confirms that the self-capacitance of a series bifilar coil scales by a factor of $N^2/4$. This confirms our "Ghost Tank" is much larger than in a standard motor coil, allowing for better energy capture.

3. Evidence: Reducing Back EMF Effects (Bedini/SSG Research)

- **The Claim:** "Time-shifting" the drag allows the rotor to pass without braking.
- **The Source:** Various replications of **Bedini SSG** systems and **delayed-Lenz** experiments.
- **The Finding:** While Lenz's Law (Conservation of Energy) is never broken, moving the reaction in time (phase shifting) prevents the drag from acting on the rotor at the critical moment. The "Bifilar Delay" creates this phase shift.

Summary: We are building a **Resonant System**.

- **The Input:** Timed perfectly to match the rotor (The Swing).
- **The Coil:** Engineered to store drag energy rather than fighting it (The Oil).
- **The Result:** A machine that spins with minimal friction, allowing us to harvest the maximum amount of energy.

1.5) CAD baseline vs. commissioning settings

The FreeCAD scripts define a baseline geometry. Some parameters are adjusted during commissioning without reprinting parts.

Parameter	CAD baseline	Commissioning target	How adjusted
Lip air gap	0.5 mm	~1.0 mm	Pocket shims + coil seating

Push pulse width	—	1.2–2.0 ms (low RPM)	Firmware; shrink at high RPM
Timing offset (deg)	—	Start with end ≈ 3–4° **after** magnet center	Firmware tuning + calibration (sweep ±10°)
Hall gap	—	0.8–1.5 mm	Physical mounting

Rule: If CAD and commissioning differ, the commissioning value is what you measure and tune to it.^43 CAD is the starting point, not the final spec.

2) Words you'll see (student definitions)

1. **Bus bar:** A thick metal strip that many wires share. We have a **+BUS** (positive) and a **Star‑Ground bus** (all negatives return here).

2. **Star‑ground:** One single spot where all negative/return currents meet. Reduces noise.

3. **Column / Row:** Columns go around the rotor (A…K, 11 total). Rows stack vertically (1…5, bottom to top).

4. **Coil Labeling:** A coil sits at a column∩row intersection and is labeled by its column letter + row number (e.g., A3 = Column A, Row 3).

   - *Legacy mapping:* Older notes may use Row-A…Row-E. Map as: Row-A=1, Row-B=2, Row-C=3, Row-D=4, Row-E=5.

5. **Bifilar coil:** Two equal windings on the same core[51]. We can connect them **parallel‑aiding** (lower resistance; faster current rise) or **series‑aiding** (higher voltage per amp; good for rectifiers).

6. **Hall sensor (A3144):** A tiny switch that flips when it sees a magnetic pole[53]. With our **North‑out** rotor, the **A3144 goes LOW when North passes** if its **flat face points AWAY** from the magnet (its back faces the magnet).

7. **MOSFET:** An electronic switch.

   - **N‑MOSFET (IRLZ44N):** On the **low‑side**, connects a row to ground when ON.

- P‑MOSFET (IRF4905): On the **high‑side**, connects a column to +BUS when ON.

8. **Gate:** The MOSFET's control pin. A driver chip makes it switch fast and safely.
9. **TVS diode (1.5KE33A):** A surge absorber that clamps spikes so parts don't die.
10. **Snubber (R//C):** A resistor + capacitor pair that tames ringing (electrical "ringing" is oscillation after a fast switch).
11. **Ultrafast diode (MUR460):** A fast "freewheel" path for coil current after switching.
12. **Bridge rectifier (KBPC3510):** Turns AC from a coil into DC.
13. **Ideal‑diode module:** A smart one‑way gate with very low loss (MOSFET‑based). Lets harvest power flow **into** +BUS but blocks back‑flow.
14. **LVD (low‑voltage disconnect):** Only connects the supercap to +BUS after the cap is charged enough; disconnects if it sags.
15. **NTC inrush limiter (5D‑11):** A resistor that is high at first and then drops—soft‑charges the supercap.
16. **Buck converter:** Step‑down power supply (e.g., from +BUS down to a clean **5 V** for the Arduino).
17. **TDC / timing offset:** "magnet center" is the point of maximum alignment. Hall sensor edges are threshold crossings, so calibrate center (e.g., midpoint of rising/falling edges) and then apply a timing offset so the pulse helps rotation instead of braking.
18. **Pitch:** Angle between magnet columns. With 11 columns, **pitch = 360°/11 ≈ 32.727°**.

3) Core decisions (locked)

- **Hall sensor columns: A and F** (5 pitches apart ≈ **163.64°**). Two sensors total.
- **Rotor polarity: North‑out**. Therefore the A3144 **flat face points AWAY** from rotor. Gap 0.8–1.5 mm.

  PUSH COILS: START WITH SERIES-AIDING.
  - *Why:* Our 18 AWG coils have extremely low resistance (~0.065 Ω in parallel). Connecting them in parallel would draw **>180 Amps** at 12V, destroying the MOSFETs.
  - Series-Aiding limits current to about 44 Amps at 12 V while providing strong torque.

Note: 44 A is within pulsed operation but near the IRLZ44N continuous rating. A stuck-ON switch can still overheat and damage the row N-MOSFET in seconds. Always use short pulses and confirm turn-off (for example with a scope or by checking that the coil does not heat when it should be OFF).

- **Harvest:** Start with Column **D** (series-aiding → bridge → supercap path). Add other harvest columns later.

- Only positives on DIN. Coil returns must go to the Row Trunk (row N-MOSFET drain) and reach Star-Ground only through the row N-MOSFET. (No black wires on DIN.)

3.5) FABRICATION: WINDING THE BIFILAR COILS

We manufacture our own coils using the **manual coil winding machine** and **18 AWG enameled copper wire**.

- **Setup:** Mount the coil bobbin (printed part) on the winder shaft. Secure two spools of 18 AWG wire behind the machine to feed simultaneously (bifilar).
- **The Recipe:**
  - **Wire:** 2 strands of 18 AWG wound together.
  - **Count: 80 turns** (the counter tracks rotations; 80 turns of *two* wires).
  - **Technique:** Keep high tension. Layer the wire neatly side-by-side. If the winding bulges too wide, it will not fit the pockets.
- **Marking:** Label the starting pair **{Coil}A_S, {Coil}B_S (e.g., A1A_S, A1B_S)** (Start) and the finishing pair **{Coil}A_F, {Coil}B_F (e.g., A1A_F, A1B_F)** (Finish).

MEASURED "TRUTH DATA" (SAFETY BASELINES)

- **Resistance per winding:** ~0.13–0.14 Ω (per 80-turn leg).
- Series-Aiding Resistance (160 turns): ~0.27 Ω. (Current at 12V: ~44 A — lower than parallel, but still requires short pulses and current limiting.)
- **Parallel-Aiding Resistance (80 turns): ~0.065 Ω**. (Current at 12V: **~184 A — DANGER**).

How we measured DCR (Direct Current Resistance): We set the bench supply to constant current = 1.00 A. We measured the voltage directly across one winding with the meter probes placed at the coil leads ({Coil}A_S to {Coil}A_F for the A winding, or {Coil}B_S to {Coil}B_F for the B winding—e.g., A1A_S to A1A_F). Then we compute DCR = V / 1.00 A. We use DCR because it tells us how much current a push pulse can try to pull (I ≈ V / DCR), which protects the MOSFETs, wiring, and fuses. Note: the bench supply's display can read higher because it also includes lead and contact voltage drop; trust the meter reading at the coil leads.

4) Mechanical quick-start

- Mount rotor; verify runout ≤ 0.05 mm. Set the working air gap (magnet face → coil face at the window) to ≈ 1.0 mm per side (±0.1 mm) using pocket shims and coil seating depth.

- *Note:* The CAD stator script uses LIP_AIR_GAP = 0.5 mm for the lip geometry[77]. We achieve the ~1.0 mm commissioning target by shimming, not by reprinting. See Section 1.5.

- **Secure magnets** (epoxy, wrap). Add a clear guard before high speed.

- **Place Halls:**
  - **Hall-A** at Column A; **Hall-F** 5 pitches after A.

  - **Orientation:** A3144 **back toward magnets**; flat face away. Gap 0.8–1.5 mm. Tiny epoxy fillets + tie-mount for relief.

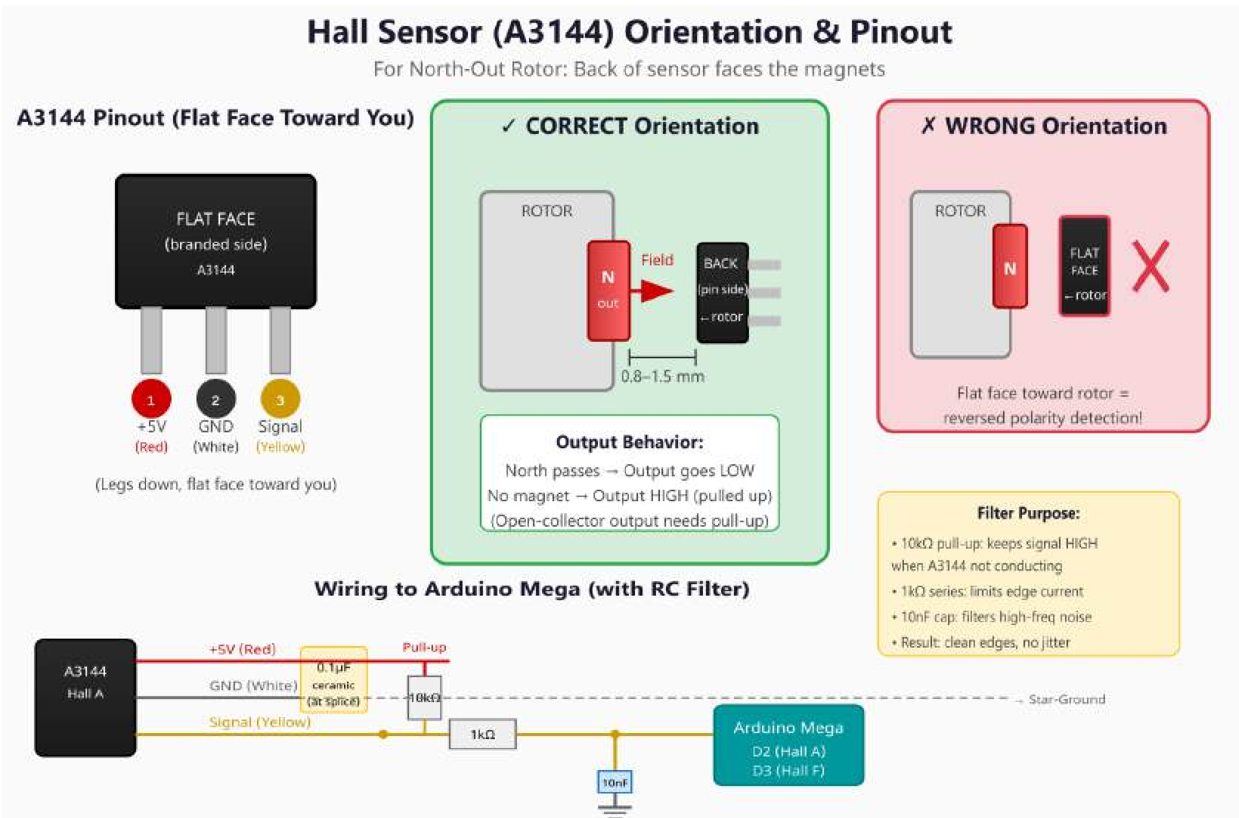


Figure 4-1 — Hall sensor orientation (A3144; back toward magnets).

5) Wiring conventions

- **Red** = positive, **Black** = negative, **Yellow** = Hall signal, **White** = Hall ground, **Red (Hall)** = +5 V.

- Every cable is labeled with wrap labels.

- **Glands:** G1…G11 carry the 11 column harnesses; **G12 carries both Hall cables**.

Master Cabling Map — Commissioning (One Push Column)

Cable Standard: {Col}{Row}{Winding}_{End} — Example: A1A_S, A1B_F, K5A_S, K5B_F

1) CABLE LABELING STANDARD (each raw wire gets a unique ID)

Format: {Column}{Row}{Winding}_{End} (Column=A–K, Row=1–5, Winding=A/B, End=S/F)
Coil A1 (4 raw leads): A1A_S, A1A_F, A1B_S, A1B_F
Coil K5 (4 raw leads): K5A_S, K5A_F, K5B_S, K5B_F
Total: 55 coils × 4 leads = 220 uniquely labeled raw wires

2) Lead definitions (bifilar = 2 windings × 2 ends)

{Coil}A_S = Winding-A Start → becomes POS lead after series link
{Coil}A_F = Winding-A Finish → joins B_S in WAGO (series jumper)
{Coil}B_S = Winding-B Start → joins A_F in WAGO (series jumper)
{Coil}B_F = Winding-B Finish → becomes RET lead after series link

3) Series-aiding internal wiring (MANDATORY) — Example: Coil A1

WAGO W_A1: Join A1A_F ↔ A1B_S (third port = mid-coil test point)
After link, two outgoing leads remain:
- POS = A1A_S → labeled "+A1" at DIN terminal
- RET = A1B_F → labeled "A1_RET" → Row trunk RT_R1

4) Push current path (Column A commissioning)

BUS+24 → F_PUSH (60A) → QP_A (P-FET) → NODE_A → DIN (+A1…+A5)
DIN → coil POS leads (A1A_S…A5A_S) → coil → RET (A1B_F…A5B_F)
RET → RT_R# → QN_R# (N-FET) → SH_PUSH → BUS−

5) Label placement (heat-shrink labels at BOTH ends)

At coil splice: Full ID (A1A_S, A1A_F, A1B_S, A1B_F)
At DIN terminal: Short form +A1, +K5 (POS leads only)
At row trunk: A1_RET → RT_R1, K5_RET → RT_R5
At MOSFET end: NODE_A, QP_A_S, QP_A_D, RT_R#, QN_R#_D, QN_R#_S

6) Wall-3 glands (inside view, right→left)

G1…G11 = Column A…K harness exits (5 coils each)
G12 = Hall sensor / low-voltage wiring bundle (shielded cable)

7) DIN rail allocation (two 35-terminal rails, 70 total)

Rail DIN_R1: +A1…+E5 (25 terminals, Columns A-E)
Rail DIN_R2: +F1…+K5 (30 terminals, Columns F-K) + 15 spares

8) Mechanical identifiers used across all sheets

ENC1 = enclosure; BP1 = backplate; DIN_R1/R2 = rails; GLAND_G1…G12 = M20 glands; STUD_GND = ground

v3.8.0 — Cable Labeling Standard Update

Figure 5-1 — Master Cabling Map: Coil grid, wiring conventions, and DIN allocation.

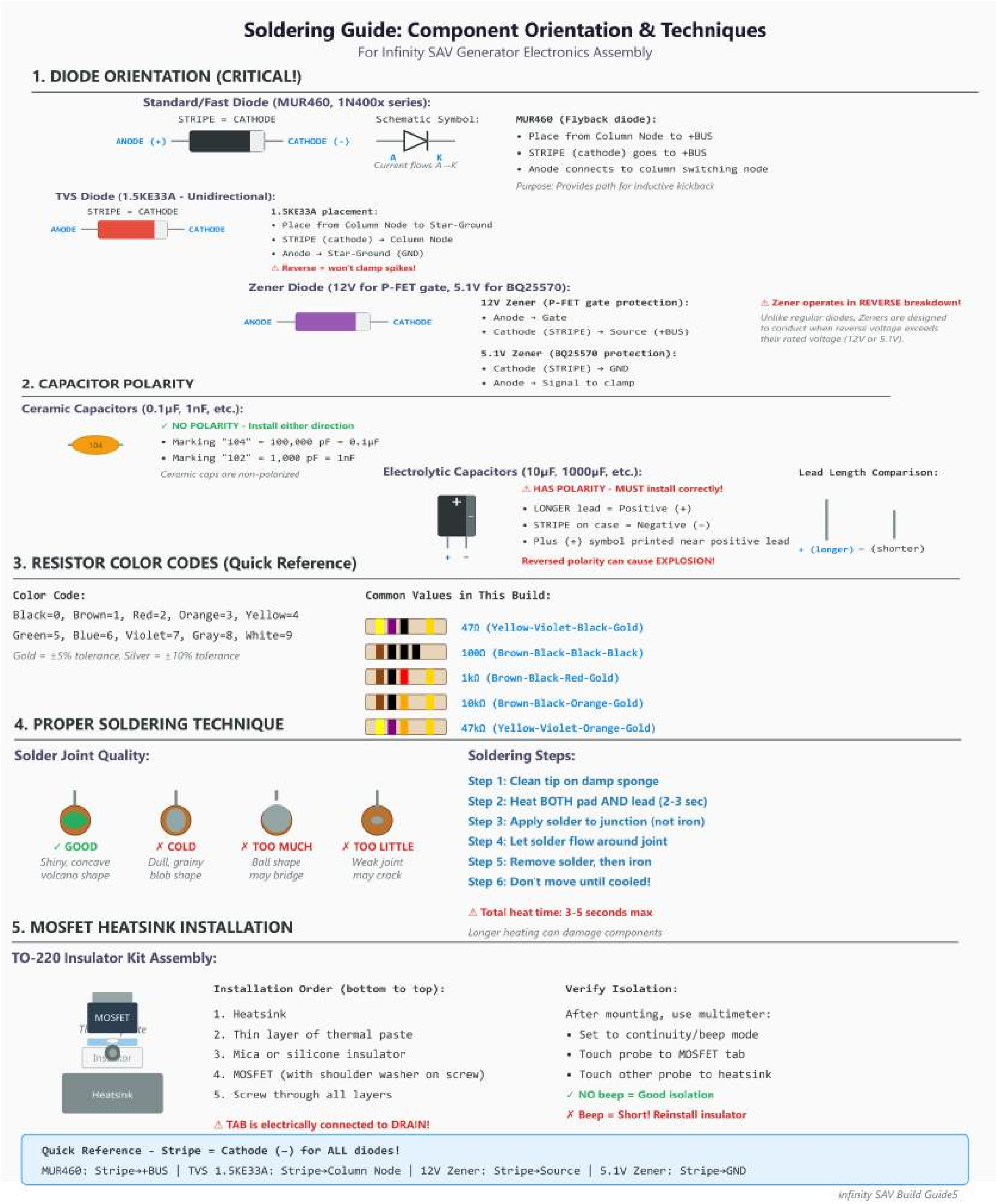


Figure 5-2 — Soldering Guide

STOP AND READ BEFORE SOLDERING: This reference saves you from the three most common mistakes that destroy electronics.

- **Diodes:** The stripe on the plastic body must match the line in the diagram symbol (Cathode). If you reverse this, the protection circuits will not work.
- **Capacitors:** Electrolytic capacitors (the cans) have a stripe on the side for **Negative (-)**. If you wire this backwards to Positive, they can pop or explode.
- **MOSFET Isolation:** You **MUST** use the white plastic insulator washer and the silicone pad between the metal tab and the heatsink. If the metal tab touches the heatsink directly, you will short-circuit the entire generator.
- **Placement Note: Do not improvise mounting.** MOSFETs must be mounted on the specific backplate heatsinks defined in **Sheet 03**. This figure only explains *how* to electrically isolate them, not *where* to put them.

The "Double-View Solder Map" Protocol

For any perfboard or custom electronics assembly, we will provide **three specific distinct artifacts**:

1. The "Placement View" (Top Down)

- **Perspective:** Looking down at the component side.
- **Purpose:** To identify *where* to insert parts before soldering.
- **Rules:**
    - Will look realistic (e.g., black cylinders for caps, green squares for adapters).
    - Will clearly show orientation (Pin 1 dots, Capacitor stripes, Diode bands).
    - **No wiring** is shown here unless it physically exists on top (like jumper wires).

2. The "Solder Map" (Bottom Up - Mirrored)

- **Perspective:** Looking at the underside (copper pads).
- **Crucial Transformation:** This view must be **Horizontally Mirrored**.
    - *Why:* When a student flips a board over, "Left" becomes "Right." If the diagram doesn't mirror, they **will** wire the chips backward.
- **Visualization:**
    - **Power Rails:** Drawn as continuous thick lines (representing bare copper bus wires).
    - **Bridges:** Drawn as short vertical/horizontal lines (representing solder drags).
    - **Legs:** Connections are not drawn as "traces" but as physical destinations (e.g., "Bend resistor leg to Pin 7").

3. The "Physical Netlist" (The Legend)

- **Concept:** We do not use electrical instructions like "Connect Gate to Drain." We use spatial instructions.
- **Numbering System:** Every solder joint gets a unique ID number (#1, #2, #3...) starting from Top-Left to Bottom-Right.

- **The Table Format:**
  - **ID:** The map number.
  - **Action:** What to do (e.g., "Solder Bridge," "Jumper Wire," "Bend Leg").
  - **Source:** The component pin (e.g., "Chip 2, Pin 6").
  - **Destination:** The physical target (e.g., "Top Copper Rail").

Harvest Coil Return Wiring (Clear Method for Students)

This section explains how to wire the negative (return) side of the harvest coils safely and cleanly, using only the parts already in your kit. These return wires connect the coils to the rectifier, which turns their generated AC into usable DC.

Goal
Each coil has two leads:
- The positive (A_S) lead — red wire — already goes to the DIN terminal for that column.
- The negative (B_F) lead — black wire — must reach the bridge rectifier.
We'll show you how to do this by grouping the B_F leads using WAGO 221-415 connectors, without needing any more DIN terminals.

Wiring the Harvest Coil Returns
1. Keep the Red Lead (A_S) on the DIN Rail
   - Every coil's A_S (red, Column-Positive) wire is already routed to a labeled DIN terminal. This does not change.
   - From these terminals, you can wire to the AC~ input of your bridge rectifier.

2. Group the Black Leads (B_F) with WAGO 221-415 Connectors
   - Each coil's B_F (black) wire is the return side of the harvest winding.
   - We do not send these to DIN blocks. Instead, group them using your WAGO 221-415 connectors.

   Here's how to group 11 B_F wires (one from each column):
   - Use one WAGO 221-415 to hold B_F from Columns A–D.
   - Use another WAGO 221-415 for B_F from Columns E–H.
   - Use a third WAGO 221-415 for B_F from Columns I–K plus one jumper wire from each of the two other WAGOs.
   - This gives you:
     • All 11 coil returns tied together.
     • One final output wire (from the last WAGO) that goes into the other AC~ terminal on the rectifier.

3. Bridge Rectifier Wiring
   - AC~ terminal #1: Connects to the output from the DIN terminal chain (the red coil leads).
   - AC~ terminal #2: Connects to the output of the B_F WAGO grouping.
   - DC+ output: Goes through a fuse (15–30 A), then through the ideal-diode module, and into +BUS.

  - DC− output: Goes directly to the Star-Ground.

Why We Do It This Way
- The original InfinitySAV machine also routed all red coil wires to labeled terminals and grouped the black wires away from the DIN rail.
- This matches what we saw in teardown photos.
- This approach keeps your wiring clean and avoids overloading the DIN rail.
- Using 3 × WAGO 221‑415 gives you enough ports to tie all coil returns together safely.

Assembly Summary:
- Coil windings are wired in series-aiding at the coil shelf using a WAGO 221‑413 (see Section 6.3).
- Each coil has:
  - A_S (red) → to DIN terminal.
  - B_F (black) → to WAGO return group.

The current path for harvest looks like this:
[Coil (A_S)] —→ DIN block —→ AC~ terminal on rectifier
[Coil (B_F)] —→ WAGO group —→ AC~ terminal on rectifier
Rectifier DC+ —→ ideal diode —→ +BUS
Rectifier DC− —→ Star-Ground

6) Coil shelf setup (per coil; all 55 the same way)

- **Goal:** Each coil exposes two nodes you can drive/harvest: a **START node** and a **FINISH node**.

- **6.1 Identify four leads** (bifilar): {Coil}A_S, {Coil}A_F, {Coil}B_S, {Coil}B_F. Example for coil A1: A1A_S, A1A_F, A1B_S, A1B_F.

- **6.2 Transition to flexible wire:** Splice each enamel wire to **18 AWG silicone** with butt splices + heat‑shrink:

  - Color: **Starts = Red**, **Finishes = Black**.

6.3 Build two nodes with WAGO 221‑413

- **SERIES-AIDING (REQUIRED FOR PUSH & HARVEST):**
  - **Tie {Coil}A_F to {Coil}B_S (e.g., A1A_F to A1B_S)** together in one WAGO (this is the series jumper).
  - The free end **A‑S** connects to **Column-Positive**.

- The free end **B‑F** connects to **Row-Negative**.

- **PARALLEL-AIDING (WARNING):**
  - *Do NOT use this configuration with 18 AWG wire and standard drivers.* It creates a near-short circuit (>180 A).

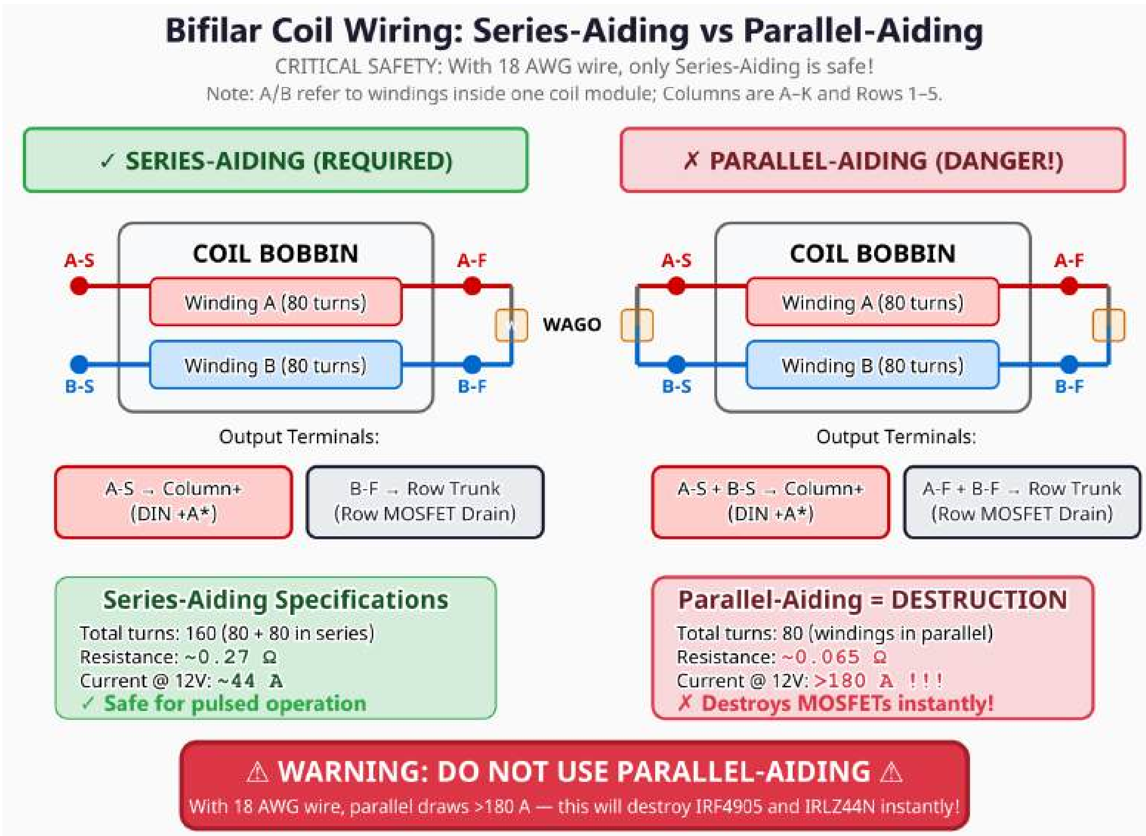


Figure 6-1 — Bifilar wiring: series-aiding (required) vs parallel-aiding (warning).

- **6.4 Polarity check (push coils)**
  - Briefly apply **+** to the node you want as **Column‑Positive** and **–** to the opposite node (3–5 V, 0.5 A limit, 1 s).
  - A North test magnet should **repel** at the coil face.
  - If it attracts, swap which node you call Column‑Positive.

- **6.5 Two outgoing pigtails per coil**
  - **Column‑Positive (red)** → to the DIN terminal of that column.
  - **Row‑Negative (black)** → to the Row Trunk Line for that row (this trunk lands on the DRAIN of that row's N-MOSFET).

- **WARNING:** Do NOT connect coil negatives directly to Star-Ground, or you bypass the row switch and lose one-hot control[98]. We are not paralleling all five coils in a column[99]. The row MOSFET selects ONE coil at a time: a coil only sees current when BOTH its column high-side switch and its row low-side switch are ON.

7) Hall sensors (A & F) — install & filter

- **7.1 A3144 pinout** (flat face toward you, legs down): **Pin1=+5 V**, **Pin2=GND**, **Pin3=Signal**.

- **7.2 Rotor polarity & Hall orientation:** With a North-out rotor, mount the A3144 with the back/pin-side facing the rotor (flat branded face points away from magnets)[102]. This makes the output go LOW when a North pole passes. Gap: 0.8–1.5 mm.

- **7.3 Close-in wiring:** Slide small heat-shrink on each leg. Push **Dupont female sockets** onto the three legs (Red→Pin1, White→Pin2, Yellow→Pin3). Heat-shrink to lock [104]. Tack in place (gap 0.8–1.5 mm), add strain relief.

- **7.4 Splice outside the stator:** Use **24 AWG 3-conductor shielded** cable: splice **Red/White/Yellow** color-to-color with butt splices. Solder **0.1 µF** ceramic from **+5 V (Red) to GND (White)** right at the splice. Large clear heat-shrink over the splice + capacitor.

- **7.5 Through G12 and into panel:** Add a **snap-on ferrite** on the cable inside the panel (two turns if it fits).

  - **Stage-1 (default):** Leave the shield/drain **unconnected at both ends**.

  - Label a short pigtail at the panel end: "**HALL SHIELD (panel end)**".

  - **If jitter/false edges appear:** Clip-lead the pigtail to **Star-Ground**.

  - If it helps, make a **single-point bond**: ring the pigtail under a DIN-rail screw (star washer), then one insulated jumper to the **Star-Ground stud**. Keep the rotor end unconnected.

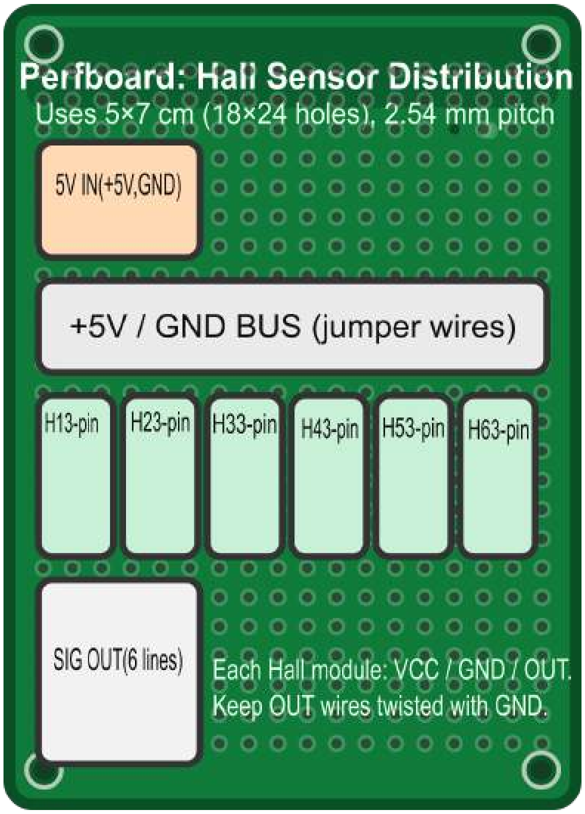


Figure 7-1: Hall Sensor Distribution Board (5x7 cm)

Objective: Create a clean "Hub" for your two Hall Sensors (A & F).
The "Flip" Rule:

- **Top View (Placement):** Insert headers here.
- **Bottom View (Soldering):** Create rails here. **MIRRORED.**

1. Component Placement (Top View)

Use **Figure 7-1a** to insert your components.

- **Inputs:** Two sets of **3-Pin Headers** for Hall A and Hall F.
- **Power:** One **2-Pin Screw Terminal** for main 5V/GND input.
- **Outputs:** A **2-Pin Header** sending just the signals (Yellow wires) to the Arduino.

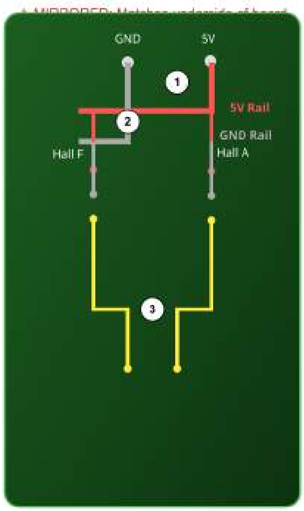
Figure 7-1a: Top View (Placement)

2. Wiring & Soldering (Bottom View)

Flip the board. This view is **MIRRORED**.

- **Red Rail:** Connects the 5V Input to the TOP pin of both sensors.
- **Grey Rail:** Connects the GND Input to the MIDDLE pin of both sensors.
- **Yellow Lines:** Connect the BOTTOM pin of each sensor to the Output Header.

Figure 7-1b: Solder Map (Mirrored)



3. Assembly Netlist (The Instructions)

Table 7-1: Step-by-Step Soldering Guide.

ID	Action	Source (Component Leg)	Destination (Target)	Critical Note
#1	**Create 5V Rail**	Bare Wire (Red Logic)	Across Top Row of Headers (Pin 1)	Connects Input 5V to Sensors A & F.
#2	**Create GND Rail**	Bare Wire (Grey Logic)	Across Middle Row of Headers (Pin 2)	Connects Input GND to Sensors A & F.
#3	**Solder Signals**	Insulated Wires	From Sensor Pin 3 (Bottom)	To Output Header pins. Keep isolated!

Critical Student Notes

- **Pin Order Warning:** This guide assumes standard **A3144 sensors** (Pin 1=VCC, Pin 2=GND, Pin 3=SIG). **Verify your specific sensor datasheet** before soldering; some clones swap VCC and SIG.
- **Color Convention:** The colors (Red/White/Yellow) are used for clarity. If your sensor wires are different colors, just ensure Pin 1 goes to the 5V Rail and Pin 2 goes to the GND Rail.

- **7.6 Arduino pin filter (right at the header):**
  - **Hall-A → D2**, **Hall-F → D3** on the Mega.
  - For each pin: **10 kΩ pull-up to +5 V**, **≈1 kΩ series** from Yellow to the pin, **10–22 nF** from pin to ground (White).
  - **Jitter gate:** With pulses ON, **jitter ≤ 0.5%** and no extra edges = good. Otherwise add the single-end shield bond.

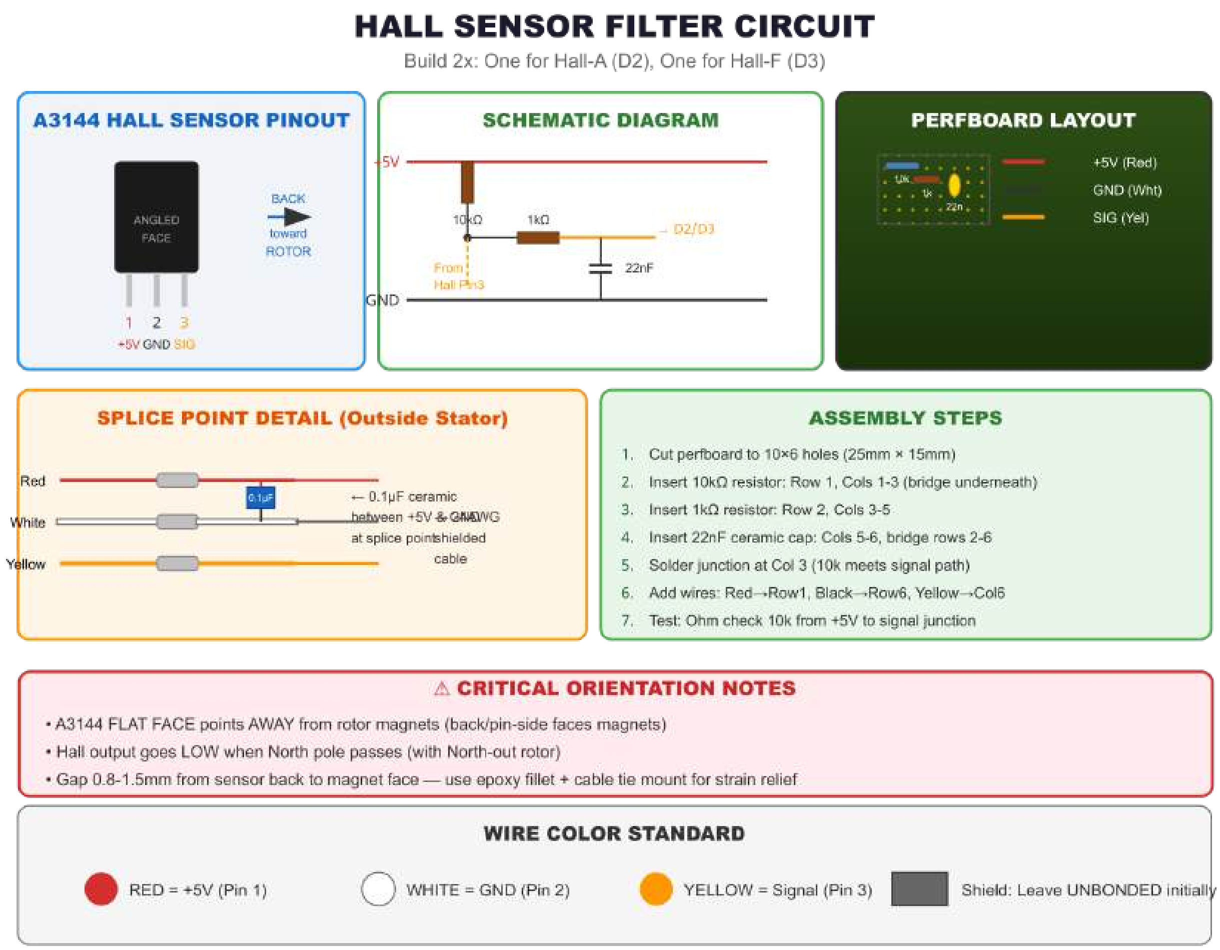


Figure 7-2: Hall Sensor Filter Assembly (Perfboard Layout)

Building the Hall Interface: Instead of twisting wires, build this small circuit on a piece of perfboard to clean up the sensor signals before they reach the Arduino.

- **The Problem:** Long wires pick up electrical noise from the coils, making the Arduino think the rotor is spinning when it isn't.
- **The Fix:** This circuit uses a "Pull-Up" resistor (10kΩ) to keep the signal strong and a capacitor (22nF) to swallow the noise spikes.
- **Wiring Check:** Connect the **Yellow Wire** from the Hall Sensor to the input side, and run a wire from the output side to Arduino pin D2 (for Hall A) or D3 (for Hall F).

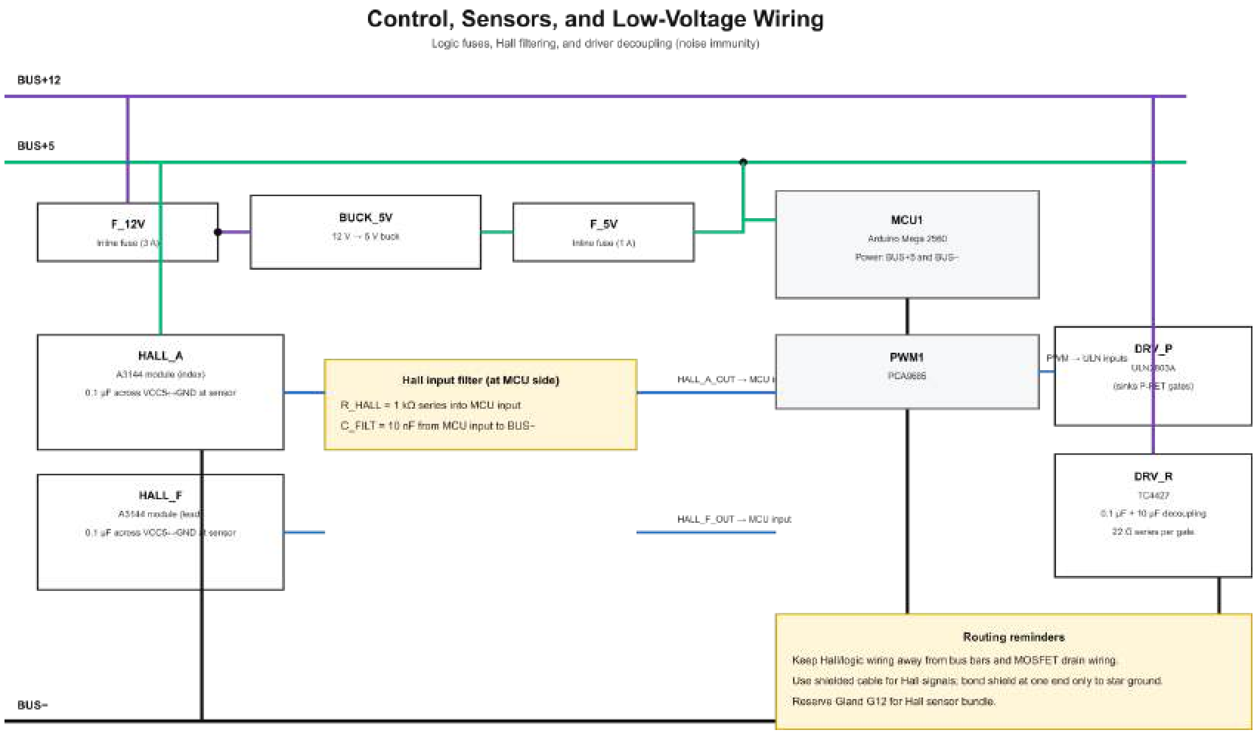


Figure 7-3: Control, Sensors, and Low-Voltage Wiring.

8) Power system & cold-start (no battery)

- **8.1 The Harvest and Cold Start Wiring:**

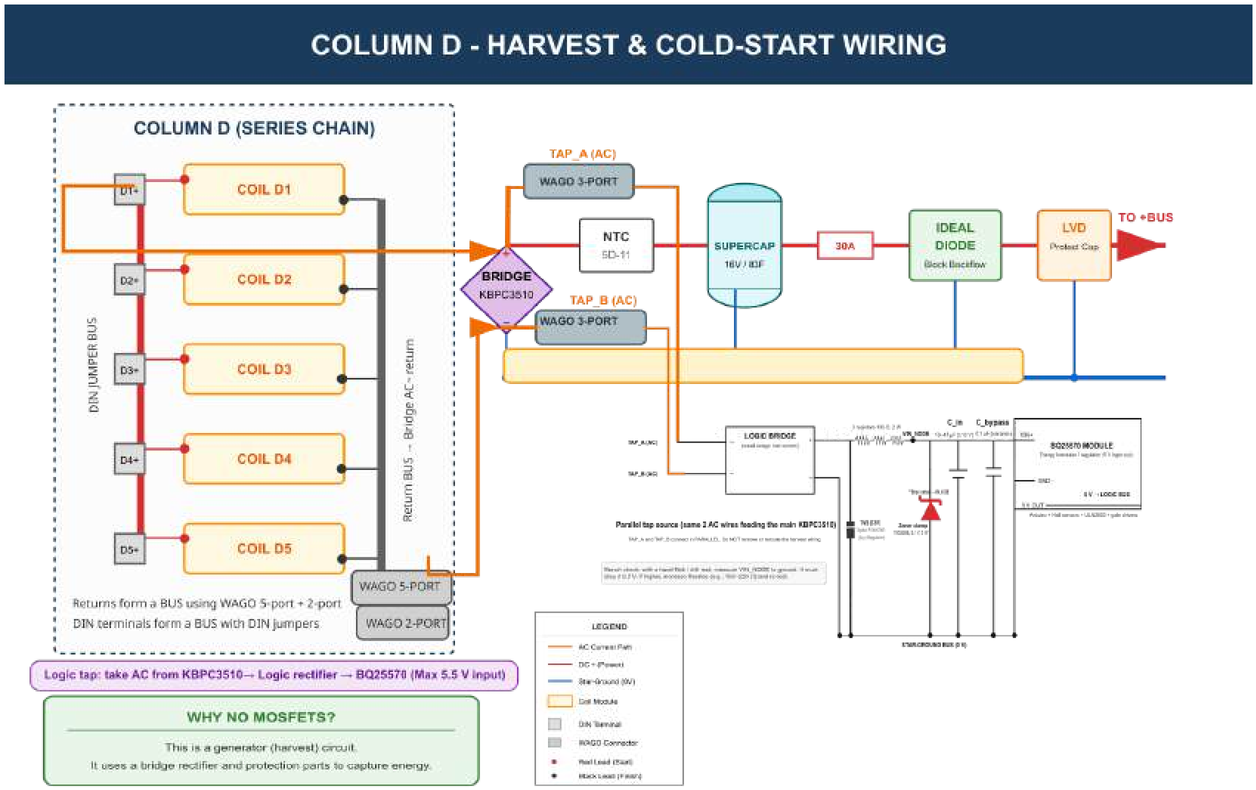


Figure 8-1. Harvest and Cold-start Wiring

8.1 How This Circuit Works Using The "Water Works" Analogy:

Imagine your electrical system is a **City Water Network**.

- **Electricity** = **Water**.
- **Voltage** = **Water Pressure** (PSI).
- **Current** = **Flow Rate** (Gallons per minute).
- **The Generator** = **The Main Water Pump**.

1. The Source (Far Left)

Component: Column D Series Chain (Coils D1–D5)

- **Description:** Five coils connected in a row (Series-Aiding).
- **Function:** This is the heart of the generator. As the magnets spin past these coils, they push and pull electrons back and forth.
- **The Analogy:** The **Main Piston Pump**. As the engine spins, this huge piston slams water forward and backward violently in the pipe. This push-pull motion is called **AC (Alternating Current)**. It is powerful but messy "sloshing" water.

2. The Split (Middle Junction)

Component: WAGO 3-Port Connectors (TAP_A & TAP_B)

- **Description:** Orange connector blocks that act as a splitter.
- **Function:** They take the raw AC power coming from the coils and split it into two separate paths without interrupting the flow.
- **The Analogy:** A **"Y" Splitter** in the main pipe. It directs the main massive flow to the heavy machinery (Top Path) and diverts a tiny "side stream" to the control room (Bottom Path).

3. Path A: The Main Harvest (Top Path)

This path handles the heavy lifting: filling the massive storage tank.

A. Bridge Rectifier (KBPC3510)

- **Description:** The square metal block with 4 pins.
- **Function:** Converts AC (push-pull) into DC (one-way flow).
- **The Analogy:** A **Check Valve Array**. It takes the violent back-and-forth sloshing from the pump and magically routes it so the water always flows out in **one direction**. Now we have a stream, not a slosh.

B. NTC Thermistor (5D-11)

- **Description:** The black disc component.
- **Function:** Limits "Inrush Current" (prevents sparks when plugging in a dead battery).
- **The Analogy:** A **Self-Widening Pipe**. When you first turn on the water, this pipe is very narrow (high resistance). This prevents a massive "water hammer" that could burst the pipes. As the water flows and warms it up, the pipe expands to let the full flow through efficiently.

C. Supercapacitor (16V / 83F)

- **Description:** The large blue cylindrical energy storage unit.
- **Function:** Stores a massive amount of energy to run the heavy push coils.
- **The Analogy:** The **City Water Tower**. It takes a lot of pumping to fill this huge tower up. But once it is full, it can supply a massive volume of water to the city (the bus) instantly.

D. 30A Fuse

- **Description:** A safety device that melts if current gets too high.
- **Function:** Protects the system from dangerous over-current.
- **The Analogy:** A **Designed Weak Link**. If the pressure gets dangerously high, this thin section of pipe bursts safely, stopping the flow immediately before the Water Tower explodes.

E. Ideal Diode Module

- **Description:** A smart circuit board that acts like a one-way valve.
- **Function:** Allows power to leave the Supercap but prevents power from flowing back into it.
- **The Analogy:** A **One-Way Gate**. It lets water leave the tower to go to the city, but it locks shut if the city water tries to flow backward into the tower.

F. Low Voltage Disconnect (LVD)

- **Description:** A smart switch that monitors voltage.
- **Function:** Disconnects the load if the battery/cap gets too low (e.g., below 11V).
- **The Analogy:** A **Smart Sluice Gate**. It keeps the gate locked shut until the Water Tower has enough pressure (e.g., 12 PSI). It prevents the city from draining the tower completely dry, which would crash the system.

Output: To +BUS (The City Main).

4. Path B: The Logic Tap (Bottom Path)

This path is the "Fast Lane." It is designed to wake up the Brain (Arduino) instantly, without waiting for the big Water Tower to fill.

A. Logic Bridge (Small Rectifier)

- **Description:** A smaller version of the main bridge.
- **Function:** Converts the AC side-stream into DC.
- **The Analogy:** Small **Check Valves**. Just like the big ones above, these ensure our little side-stream flows in only one direction.

B. TVS Diode (P6KE33CA)

- **Description:** A fast-acting clamping diode.
- **Function:** Protects against sudden voltage spikes above 33V.
- **The Analogy:** An **Emergency Pressure Relief Valve**. Normally, it stays closed. But if the pump sends a sudden "shockwave" (over 33 PSI), this valve blasts open instantly to dump the excess pressure. It saves the delicate pipes downstream from bursting.

C. Resistor Series Chain (3x 100Ω)

- **Description:** Three resistors connected end-to-end.
- **Function:** Limits current and drops the voltage significantly.
- **The Analogy: Long, Rough Tubing**. The pump pressure is huge (30 PSI), but the Brain can only handle 5 PSI. We force the water through a long, narrow, rough tube. The friction slows the water down, dropping the pressure from "Fire Hose" levels to gentle "Garden Hose" levels. We use three sections so no single section gets too hot from the friction.

D. Zener Diode (5.1V)

- **Description:** A voltage-regulating diode.
- **Function:** Clamps the voltage at exactly 5.1V.
- **The Analogy:** An **Overflow Pipe**. Imagine an open vertical pipe cut exactly at **5.1 inches high**.
  - If the water level is safe (below 5.1), it flows past.
  - If the water level tries to rise to 6 inches, it simply spills out the top of this pipe onto the floor.
  - This guarantees the pressure going to the Brain **never** exceeds 5.1 PSI.

E. Capacitors (0.1μF & 10μF)

- **Description:** Small energy storage buckets.
- **Function:** Filters out noise and smooths the voltage.
- **The Analogy: Buffer Buckets**. The water coming out of the rough tubing might be sputtering or dripping. We put small buckets here to catch the drips. They provide a smooth, steady pour into the final filter.

F. BQ25570 Module

- **Description:** A high-efficiency energy harvester and regulator.
- **Function:** Takes the variable input and outputs a perfect 5.0V.
- **The Analogy:** The **Brita Filter**. It takes the pre-treated water we just cleaned up and polishes it to a perfect, steady 5.0V supply for the delicate Arduino brain.

Output: To 5V Logic (Powers the Arduino and Hall Sensors).

- **8.2 Cold‑start chain (Column D harvest):**

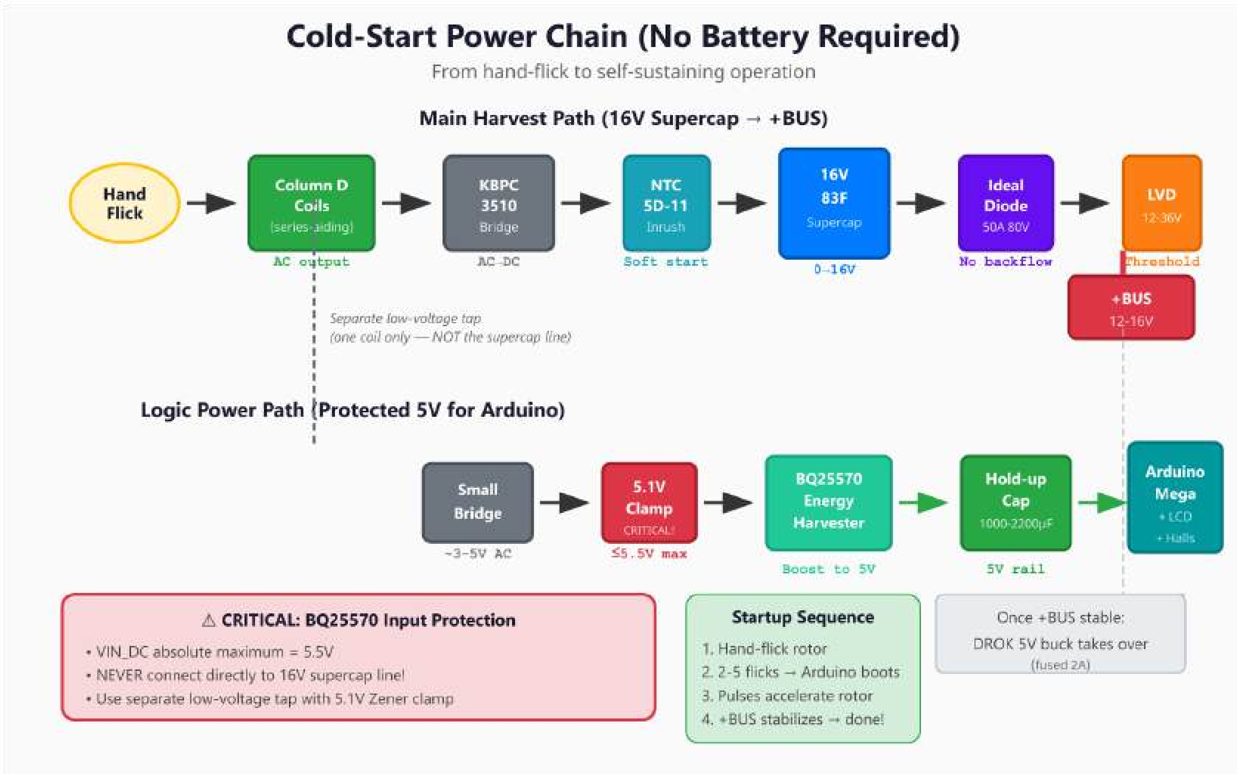


Figure 8-2 — Cold-Start Power Chain (No Battery Required)

Figure 8-2 shows how the system starts with no battery. A hand flick makes the generator coils produce alternating current. The **bridge rectifier** changes that into direct current so it can charge the **supercapacitor.**. The **NTC inrush limiter** (NTC = a resistor that starts high and then drops) prevents a big surge of current into an empty supercapacitor. The **ideal diode** stops +BUS from dumping back into the supercapacitor. The **low-voltage disconnect** keeps the system from turning on too early and then collapsing. At the same time, a **separate low-voltage tap** powers the Arduino safely through a **5.1 V clamp** (clamp = prevents voltage going above a safe limit) and the **BQ25570 energy harvester** (energy harvester = chip that boosts low input voltage to a stable 5 V). Once +BUS is stable, the normal **5 V buck converter** powers the logic rail.

- D-column coils in series-aiding → KBPC3510 bridge → NTC 5D-11 → **16 V / 83 F supercap** → Ideal-diode module (50 A, 80 V) → LVD → +BUS.
- *Why:* Hand-flick creates AC on D; bridge makes DC; NTC soft-charges the cap; ideal-diode lets energy flow **to the bus** but blocks back-flow; LVD only connects the cap to +BUS when it's high enough.

- 8.2 Logic power at startup

First hand-flick induces voltage on the Column-D series string; its bridge rectifier

produces DC.

IMPORTANT: The BQ25570 input (VIN_DC) must stay within its absolute maximum (≈5.5 V). Since the full Column D voltage can exceed 30 V at high speed, you must **not** connect the BQ25570 directly to the bridge output. Instead, feed the BQ25570 from a **Protected Logic Tap** connected in parallel to the Column D AC lines.

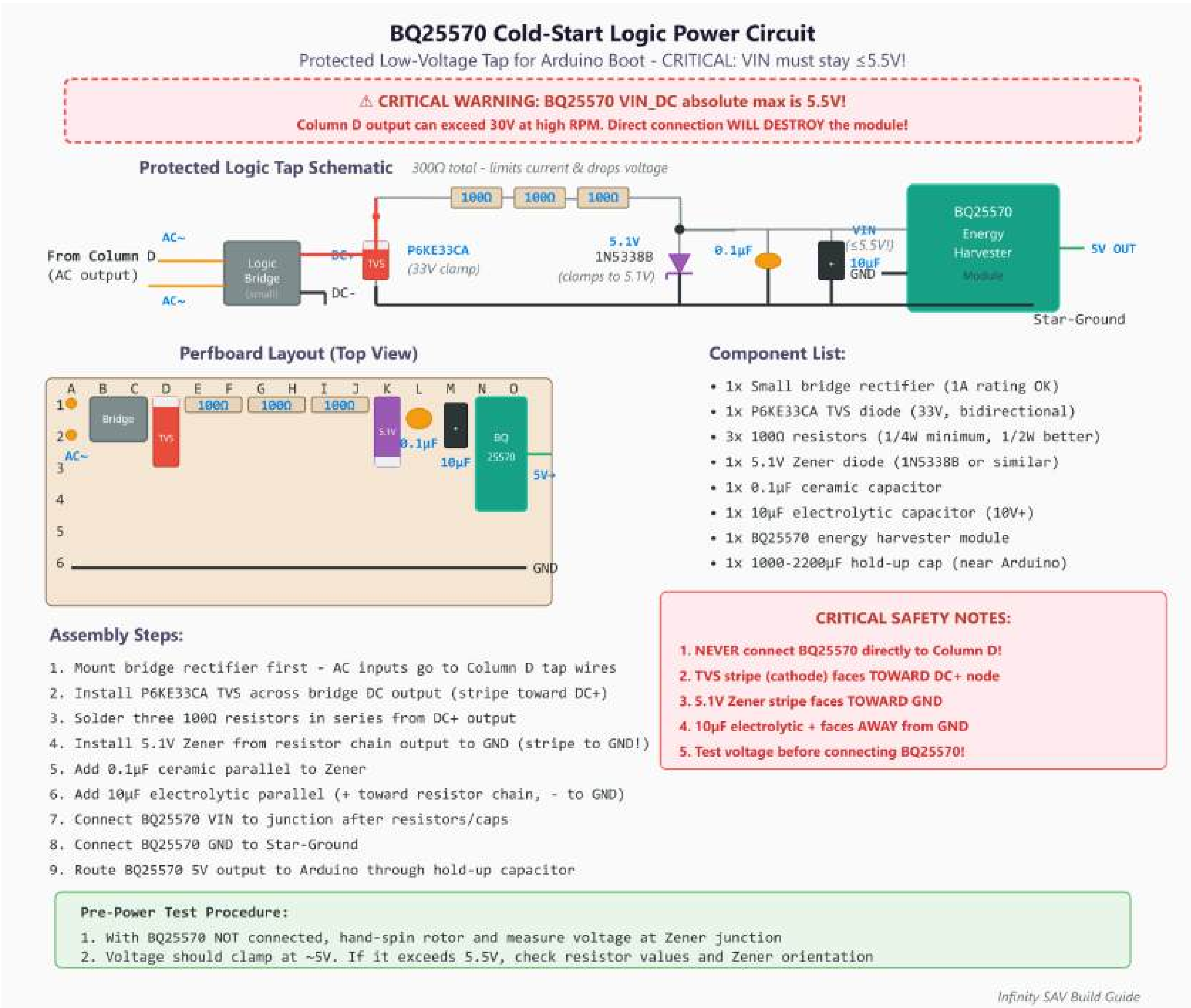


Figure 8-3: Protected Logic Tap Assembly (BQ25570 Protection)

CRITICAL SAFETY CIRCUIT: You must build this "Firewall" before connecting the BQ25570 Energy Harvester.

- **Why:** The main harvest voltage can spike to 30V+, but the BQ25570 chip will die instantly if it sees more than 5.5V.
- **How it works:**
    1. **The Bridge:** Takes AC from the coil tap and turns it into DC.
    2. **The Resistors (3x 100Ω):** These limit the current and slow the voltage rise so the clamp diodes can safely protect the circuit.

3. **The Zener (5.1V):** Acts as an emergency overflow valve. It forces the voltage to stay at 5.1V, keeping the BQ chip safe.

- **Test First:** Before plugging in the BQ module, spin the rotor by hand and measure the voltage at the "OUT" pads. It should never exceed 5.2V.

8.3 Logic Power Protection Assembly (The "Firewall")

Objective: Build the protection circuit that allows the generator to self-start without destroying the sensitive energy harvester.

The Problem: The generator produces high-voltage spikes (30V+). The BQ25570 chip explodes at 5.5V.

The Solution: This board acts as a "Firewall." It clamps the voltage to a safe 5.1V before it reaches the brain.

Critical Rule: The "Flip"

- **Top View (Placement):** You look down at the components. The AC (~) pins are on the **LEFT**.
- **Bottom View (Soldering):** You flip the board over. The AC (~) pins appear on the **RIGHT**.
- **Safety:** Always trust the symbols printed on the Bridge Rectifier body (+, -, ~), regardless of left/right orientation.

Step 1: Component Placement (Top View)

Use **Figure 8-4a** to insert your components. Do not solder yet.

- **Bridge Rectifier:** Align the printed symbols. + goes toward the resistors. ~ goes toward the AC input wires.
- **TVS Diode (P6KE33CA):** This part is **Bidirectional**. It has no polarity band; you can install it either way.
- **Zener Diode (1N5338B):** The **Stripe (Cathode)** must face **UP** (towards the resistors). This connects it to the Positive rail.
- **Resistors:** Leave a small air gap (2mm) under the resistors for cooling.

Figure 8-4a: Top View (Placement)

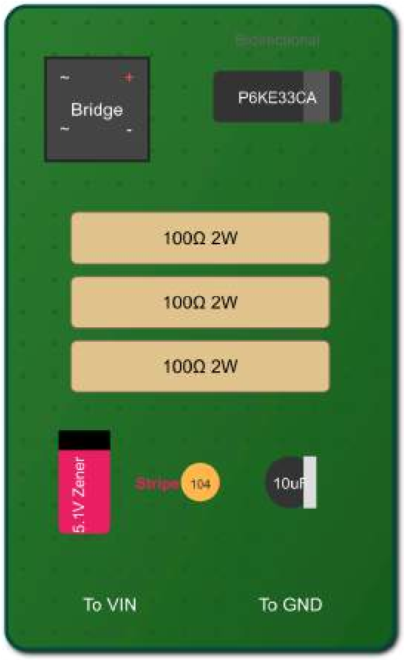


Figure 8-4a: Top View (Placement). Note the Zener Stripe facing the positive rail.

Step 2: Wiring & Soldering (Bottom View)

Flip the board over. Use **Figure 8-4b** to solder the connections. This view is **MIRRORED**.

- **Copper (Orange Lines):** This is the **VIN (+)** path.
- **Grey (Silver Lines):** This is the **GND (-)** path.
- **Warning:** Notice that the AC Inputs (#1a, #1b) are now on the **RIGHT** side.

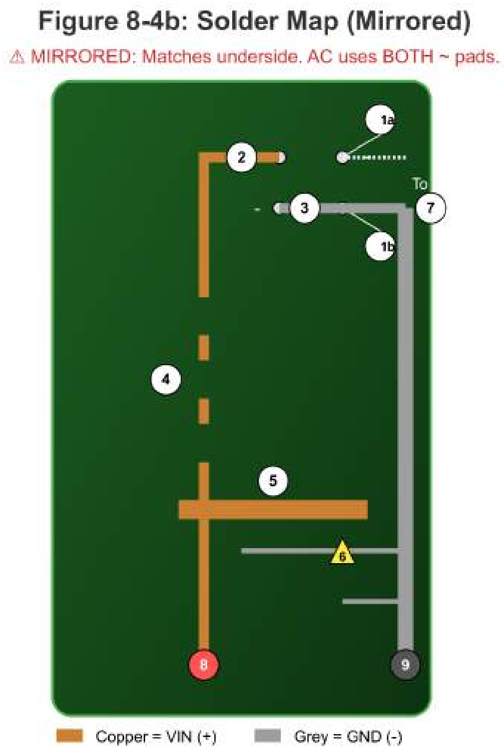


Figure 8-4b: Bottom View (Mirrored Solder Map). Follow the numbered ID tags in the table below.

Step 3: Assembly Netlist (The Instructions)

Table 8-4: Step-by-Step Soldering Guide

ID	Action	Source (Component Leg)	Destination (Target)	Critical Note
#1	**Solder Wire**	Bridge ~ and ~ pins	To Column D Tap Wires	Use **BOTH** AC pins. Polarity doesn't matter.

#2	**Solder Bridge**	**Bridge DC+ (Marked +)**	Copper Rail (VIN)	**Verify Symbol on Bridge Body.**
#3	**Solder Bridge**	**Bridge DC- (Marked -)**	Grey Rail (GND)	**Verify Symbol on Bridge Body.**
#4	**Solder Chain**	Resistor Legs	R1 $\to$ R2 $\to$ R3	Connect the 3 resistors in a straight line.
#5	**Create Rail**	Copper Wire (Bare)	Across R3, Zener, Caps, Output	This is the "Clean 5.1V" Junction Bar.
#6	**Orientation**	⚠ **CHECK ZENER**	**Stripe $\to$ Junction Bar (#5)**	**CRITICAL:** Stripe (Cathode) $\to$ Positive (#5).
#7	**Create Rail**	Bare Wire (GND Rail)	Down the Right Side	Connects Bridge - and Zener Anode.
#8	**Solder Wire**	Red Wire (Out)	From **Pad #8** (on Copper Rail)	To BQ25570 VIN_DC.
#9	**Solder Wire**	Black Wire (Out)	From **Pad #9** (on Grey Rail)	To Star Ground.

Step 4: System Logic & Troubleshooting

How it works:

1. **Harvest:** The Bridge converts raw AC into DC.
2. **Surge Protection:** The TVS Diode clamps massive spikes (up to 33V).
3. **Current Limiting:** The 3 resistors (300Ω total) limit current and drop voltage.
4. **Voltage Clamp:** The Zener Diode forces the output to stay at **5.1V**.
5. **Bootstrap:** The BQ25570 takes this safe energy and boosts it to 5V to wake up the Arduino.

Troubleshooting:

- **System won't boot:** Check **Step #6**. If the Zener is backwards, it shorts the power to ground.
- **No Voltage at Output:** Check **Step #1**. Did you connect *both* AC wires?
- **Resistors overheating:** Check if the TVS diode is bridging/shorting the rails at **Step #2/#3**.

- **Note: Optional DC/DC for drivers:** A **12 V buck** for the TC4427 drivers (if you choose to drive row MOSFETs at 10–12 V). Fuse 10 A.

8.5 The Master Power Map (System Architecture)

Figure 8-5 (below) is the "Subway Map" of your generator. It shows how the different voltage rails (Harvest, Logic, and Main Bus) interact without blowing each other up.

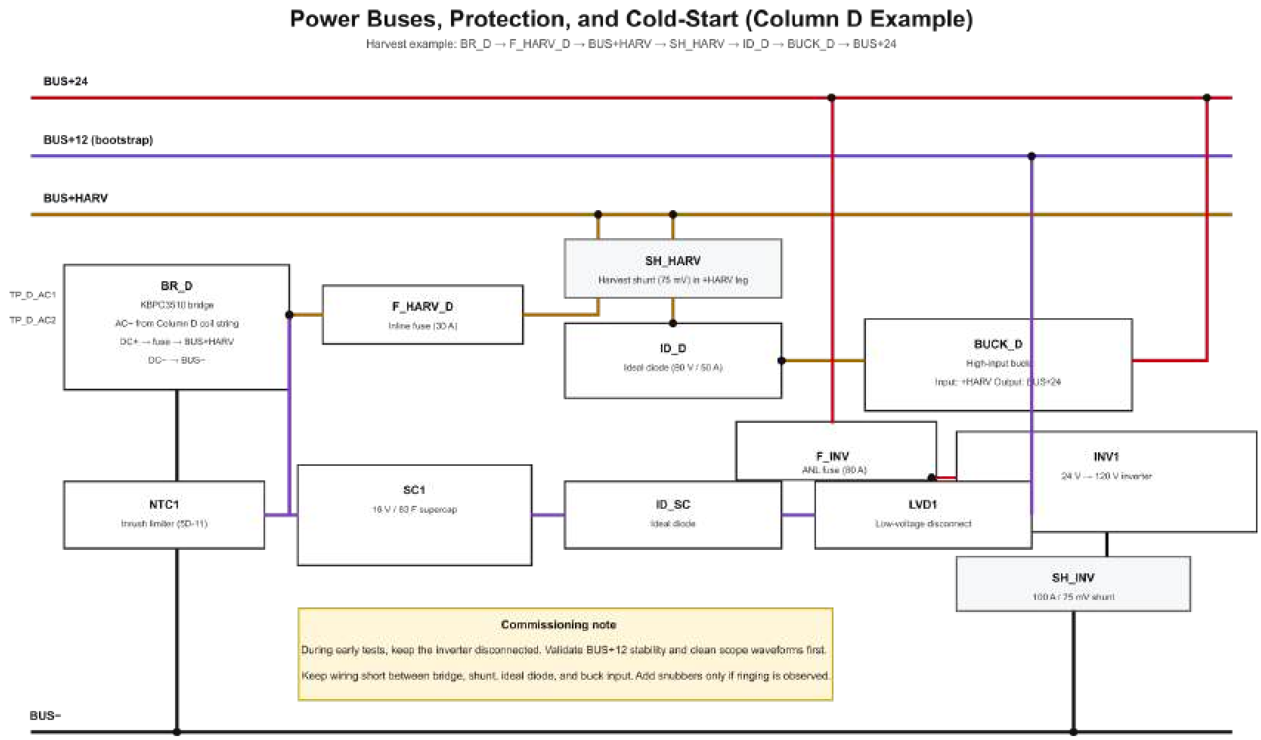


Figure 8-5: Master Power Schematic. This shows the logical flow of energy from Coils to the Main Bus.

How to Read This Map:

1. **The Three "Subway Lines" (Voltage Rails):**
   - **Brown Line (Harvest):** This is raw, wild energy coming straight from the coils/bridges. It is variable and unregulated.
   - **Red Line (+24V Bus):** This is the "City Main." It is the stable, high-power rail that feeds the Inverter and the Push Coils.
   - **Purple Line (+12V Bus):** This is a specialized local rail created by the **Buck Converter** (`BUCK_D`) to power the Gate Drivers (TC4427).
2. **The "Traffic Cops" (Ideal Diodes):**
   - Look at **`ID_D`** and **`ID_SC`**. These are **Ideal Diodes**.
   - **Their Job:** They act like one-way check valves. They allow the Harvest (`ID_D`) and the Supercap (`ID_SC`) to *both* pour water into the main tank (+BUS), but they prevent the water from flowing *backwards* into the wrong pipe.
3. **The Safety Layer (Fuses & Protection):**
   - **Fuses (`F_...`):** Notice that every major block has a fuse *before* it joins the main line. If the Inverter shorts out, `F_INV` blows to save the Supercap.
   - **Inrush Limiter (`NTC1`):** This is the "Soft Start" valve. It sits between the Bridge and the Supercap. Without it, plugging in the empty Supercap would be like opening a floodgate—it would spark and melt the wires.

Student Note: Physical vs. Logical Wiring

- **Shunt Location:** This diagram shows the Harvest Shunt (`SH_HARV`) on the **Positive** line (High-Side).
- **WARNING:** Most student meters (like **Bayite**) require shunts on the **Negative** line (Low-Side) to work correctly.
- **Rule:** Follow this diagram for **Logic** (where power goes), but follow **Section 11 (Meters)** for the specific wiring of your black shunt wires.

9) Switching: how does one coil get a push?

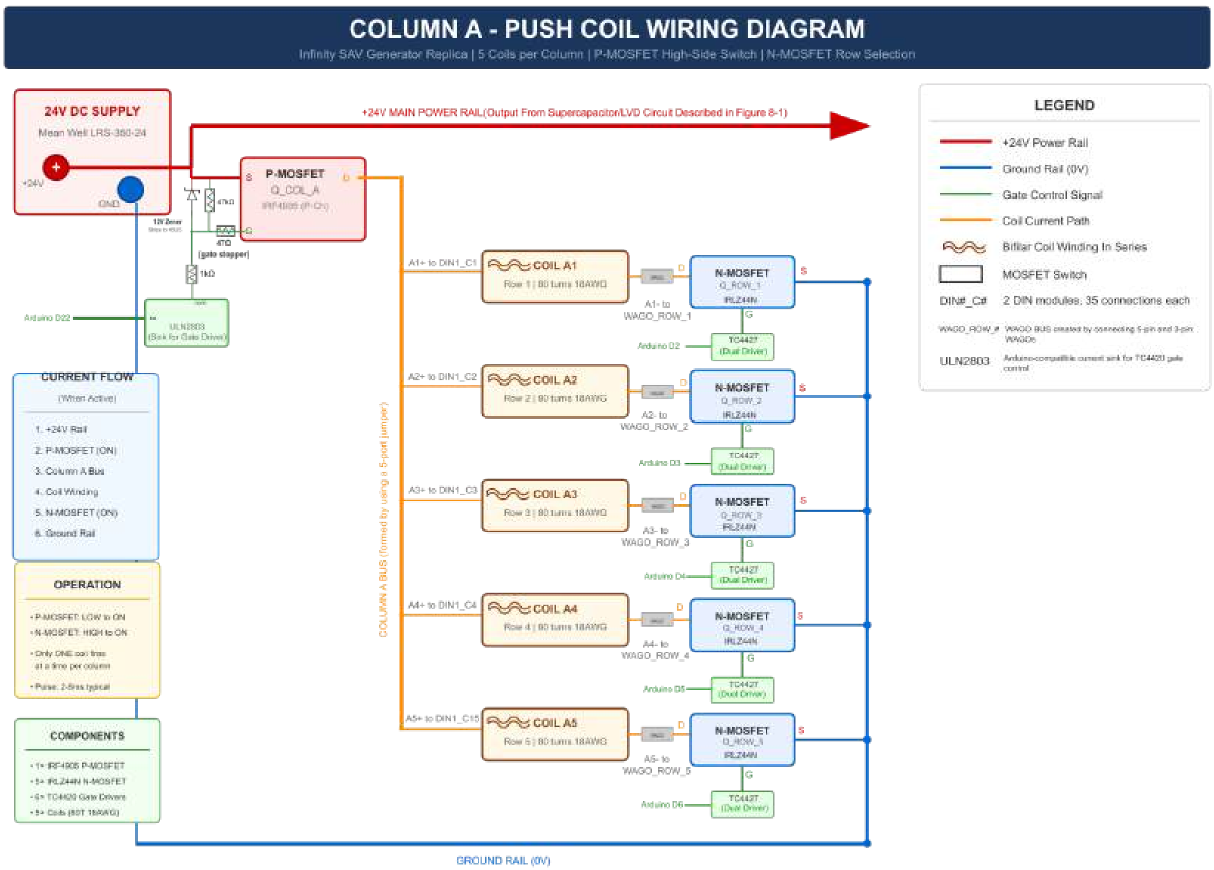


Figure 9-1. Column A - Push Coil Wiring Diagram. (Also applies to any push column like F)

The Push Coil System Analogy: Precision Sniping (Self-Powered).

The Analogy: Imagine your generator is an **Off-Grid Building** (Column A). You rely entirely on your own rooftop **Water Tower** (Supercapacitor) to run the machinery. You must efficiently spray water (electricity) into *one* specific turbine at a time to keep the generator spinning, which in turn refills your tower.

1. The Source (Far Left): The Rooftop Water Tower

- **Component: +24V MAIN POWER RAIL (Red Line)**
- **Description:** Fed directly from the Supercapacitor/LVD Circuit (Harvest System).
- **The Analogy: Your Private Water Tower.**
  - **Startup:** At first, the tower is empty. You spin the rotor by hand ("hand-flick") to pump the first few gallons up into the tank.
  - **Running:** Once the tower has enough pressure (Voltage), the LVD valve opens and pressurizes this red pipe. Now, the building runs on its own stored water.

2. The High-Side Switch (Top Center): The Building Master Valve

- **Component: P-MOSFET (IRF4905) + Gate Network**
- **Function:** Controls power for the *entire* column. If this is OFF, the building is dry.
- **The Analogy: The Building Master Valve.**
  - Even though the Tower is pressurizing the pipe, this valve stays shut tight (held by the **47kΩ Spring**). Water cannot enter the floors until we pull the lever.

Sub-Component: ULN2803 (The Valve Operator)

- **Analogy: The Pilot Valve.** The Arduino is too weak to pull the heavy Master Valve lever. It whispers to the ULN2803, which uses its strength to pull the lever down, letting water rush into the building.

Sub-Component: Zener (12V) + 1kΩ Resistor (The Safety Chain)

- **Analogy: The Safety Chain.** The lever is fragile. If you pull it all the way down (24 inches), it snaps. This chain stops the lever at exactly **12 inches** (12V), letting the valve open fully without breaking the handle.

Sub-Component: 47Ω Resistor (The Gate Stopper)

- **Description:** The tiny resistor right at the MOSFET Gate pin.
- **Analogy: The Vibration Damper.** When the Pilot Valve slams the lever down, the lever might bounce or vibrate ("ring"). This small rubber washer absorbs the vibration so the valve opens smoothly and stays cool.

3. The Column Bus (Vertical Orange Line): The Riser Pipe

- **Component: Column A Bus**
- **The Analogy: The Vertical Riser Pipe.**
  - When the Master Valve opens, water fills this vertical pipe. Every floor (Coils A1–A5) has pressurized water waiting at the inlet, but nothing flows yet because the *drains* are closed.

4. The Loads (Center): The Water Wheels

- **Components: Coils A1–A5 (Bifilar Series-Aiding)**
- **The Analogy: Turbines (Water Wheels).**
  - Water pushes against the wheel, trying to get through. But water can't spin the wheel if the pipe *after* it is blocked. It sits there, holding potential energy.

5. The Low-Side Switches (Right): The Floor Drain Valves

- **Components: N-MOSFETs (IRLZ44N) + TC4427 Drivers**
- **The Analogy: Individual Floor Drains (The Trigger).**
  - This is the moment of action.
  - Arduino signals the **Floor 3 Drain** to **SLAM OPEN** (via the TC4427 Hydraulic Booster).
  - Water rushes from the Tower, down the Riser, through **Turbine 3**, and out the drain.
  - **Result:** The rotor gets a magnetic kick! Turbines 1, 2, 4, and 5 stay dry.

6. The Return (Bottom Blue Line): Recirculation

- **Component: Ground Rail (0V)**
- **The Analogy: The Return Pipe.**
  - The water drains back to the negative side of the capacitor bank to complete the loop.

7. The Protection: Flyback Management (Crucial)

- **Components: MUR460 Diode + TVS Diode** (Located at the P-MOSFET).
- **The Analogy: The Water Hammer Arrestor.**
  - **The Danger:** When you slam the Master Valve shut at the end of a pulse, the rushing water (electricity) crashes into the closed valve. This shockwave ("Flyback") can burst the pipes.
  - **The Fix:**
    - The **MUR460** is a recirculation loop that guides this shockwave back into the tank safely.
    - The **TVS** is an emergency airbag that absorbs any leftover spike.
  - **Student Note:** Without these, your Master Valve (P-MOSFET) will eventually crack from the pressure shocks.

Summary of the Pulse

1. **Flick:** Hand-spin pumps water to the **Tower**.
2. **Ready:** Tower fills. Arduino wakes up.
3. **Fire:**
   - Arduino opens **Master Valve** (P-FET). Pipes pressurize.
   - Arduino slams open **Floor 3 Drain** (N-FET).
4. **Action:** Water blasts through **Coil A3**. Rotor spins.

5. **Safety:** Valves slam shut. The **Arrestor (MUR460)** catches the shockwave.
6. **Refill:** The faster rotor pumps more water back into the Tower.

9.1 High-side column switch (2 columns: A & F)

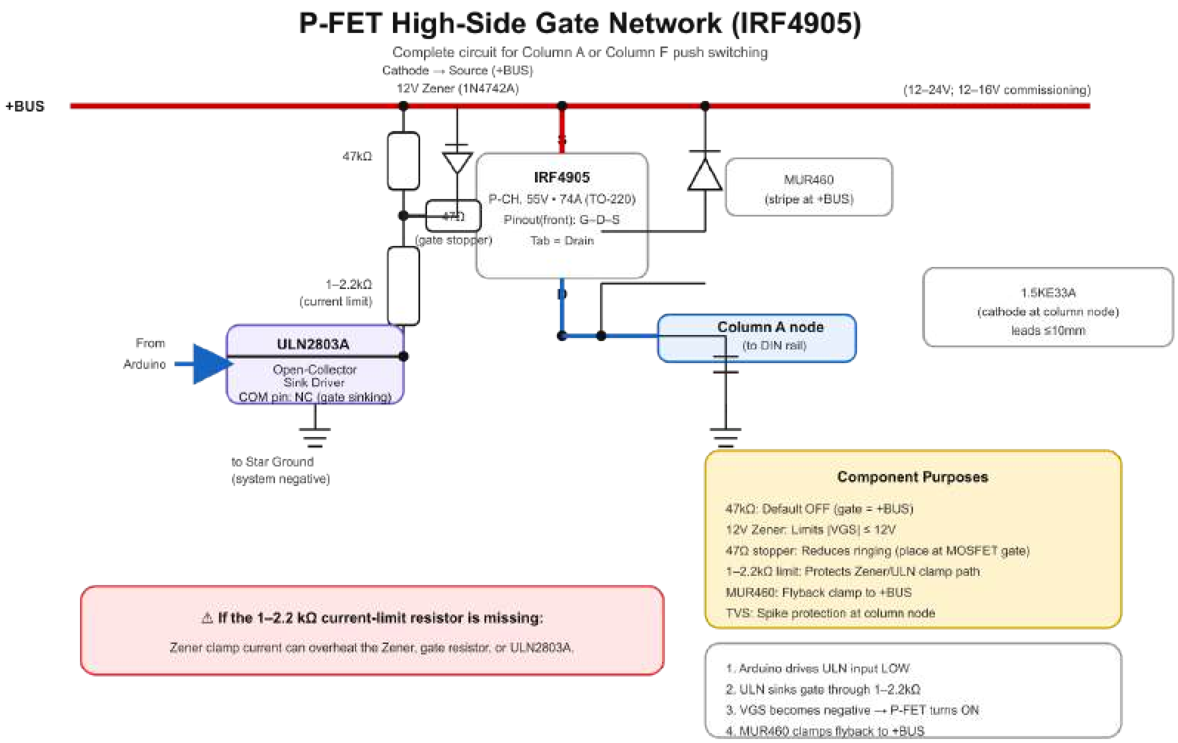


Figure 9-1a: P-FET High-Side Gate Network (IRF4905)

In the main **Push Coil Diagram**, we showed the **P-MOSFET (IRF4905)** acting as the **"Building Master Valve"** that controls all power to Column A.

However, you cannot just wire a switch directly to a high-pressure line. If you pull the lever too hard, it snaps. If you close it too fast, the pipes burst. This detailed schematic shows the **Safety Mechanism** required to operate that Master Valve safely.

1. The Default State: The "Safety Spring" (OFF)

- **Component: 47 kΩ Pull-Up Resistor**.
- **Location:** Connected between the Gate (Control Lever) and the Source (+BUS).
- **The Analogy: The Tension Spring.**
    - This spring holds the valve lever tight against the high-pressure line ($V_{GS} = 0$).
    - **Why:** It ensures the Master Valve stays **locked shut** by default. If the Arduino resets or loses power, this spring snaps the valve closed so the coil never fires accidentally.

2. The Turn-ON Sequence: Pulling the Lever

To open the valve, we must pull the Gate lever *down* (lower voltage than the Source).

- **The Muscle (ULN2803A):**
  - **Analogy: The Pilot Valve.** The Arduino is too weak to pull the heavy Master Valve lever itself. It signals the ULN2803A, which uses its strength to pull the lever down towards the ground.
- **The Safety Chain (12 V Zener Diode):**
  - **Analogy: The Stop Chain.**
  - **The Danger:** The IRF4905's internal lever is fragile. If you pull it all the way down to the floor (24 V difference), it snaps (Gate Oxide Rupture).
  - **The Fix:** The Zener Diode acts like a chain that catches the lever after exactly **12 inches** (12 V). It allows the valve to open fully but prevents the lever from being pulled to the breaking point.
- **The Current Limiter (1 kΩ Series Resistor):**
  - **Analogy: The Shock Absorber.**
  - **Critical Physics:** When the "Safety Chain" (Zener) catches the lever, there is a sudden jerk of tension. Without this resistor, the full force of the 24 V bus would rush through the Zener, burning it out instantly. This resistor limits the tension to a safe level (a few milliamps).
- **The Damper (47 Ω Gate Stopper):**
  - **Analogy: The Rubber Washer.**
  - **Location:** Right at the MOSFET Gate pin.
  - **Why:** When the Pilot Valve slams the lever, it tends to vibrate or "ring" like a bell. This small resistor absorbs the vibration, ensuring the valve opens smoothly without overheating.

3. The Turn-OFF: Managing "Water Hammer"

- **Components: MUR460 Diode** and **TVS (1.5KE33A)**.
- **Analogy: Water Hammer Arrestors.**
  - **The Danger:** When you slam the Master Valve shut after a pulse, the electricity (water) rushing through the column crashes into the closed valve. This creates a massive pressure spike ("Flyback") that can crack the pipe.
  - **The Fix:**
    - The **MUR460** recirculates this shockwave back into the supply tank safely.
    - The **TVS** acts as an emergency airbag to absorb any leftover spikes.

Summary for Assembly (The "Blueprint")

When building the **P-MOSFET Module**, follow these rules to match the diagram:

1. **Placement:** Mount the IRF4905 on a heatsink bar.
2. **Connections:**
   - **Source:** Connects to **+24V BUS**.

    - **Drain:** Connects to the **Column DIN Rail**.
3. **The Gate Network (Soldered to the MOSFET):**
    - **47 kΩ:** Connects Gate $\to$ Source (The Spring).
    - **12 V Zener:** Connects Gate $\to$ Source (Stripe touches Source).
    - **47 Ω:** Connects directly to the Gate pin (The Damper).
    - **1 kΩ:** Connects between the Gate Network and the ULN2803 driver (The Limiter). **Do not omit this.**
4. **Logic:**
    - **To Fire:** Arduino drives ULN input **HIGH**. ULN pulls the gate **LOW**. Valve Opens.
    - **To Stop:** Arduino drives ULN input **LOW**. Resistor pulls gate **HIGH**. Valve Closes.

In more technical language, this circuit controls the flow of power from the +BUS into the top of the coil. We use a P-channel MOSFET (IRF4905) here because it is designed to switch the high-voltage (positive) side of the line.

1. The Default State (Safety OFF)

Component: 47 kΩ Pull-Up Resistor.

Role: Connects the Gate to the Source (+BUS).

Why: By keeping the Gate voltage equal to the Source voltage ($V_{GS} = 0$), the MOSFET stays tightly closed (OFF). This ensures the coil never fires accidentally if the Arduino resets.

2. The Turn-ON Sequence

To open the valve, we must pull the Gate voltage lower than the Source.

The Driver (ULN2803A): The Arduino cannot handle 24 V directly. It signals the ULN2803A, which acts as a "sink" (connecting the circuit to Ground).

The Protection (12 V Zener Diode):

The Danger: The IRF4905 gate insulation ruptures if the voltage difference exceeds 20 V. If your +BUS is 24 V and you pull the gate to 0 V, that 24 V difference destroys the chip.

The Fix: The Zener Diode "clamps" the difference to exactly 12 V—perfect for turning the switch ON safely.

The Current Limiter (1 kΩ – 2.2 kΩ Series Resistor):

Critical Physics: When the Zener activates, it creates a short circuit from +BUS through the driver. Without this resistor, massive current would flow, burning out the Zener instantly. This resistor limits that current to a safe level (a few milliamps) while allowing the gate to trigger.

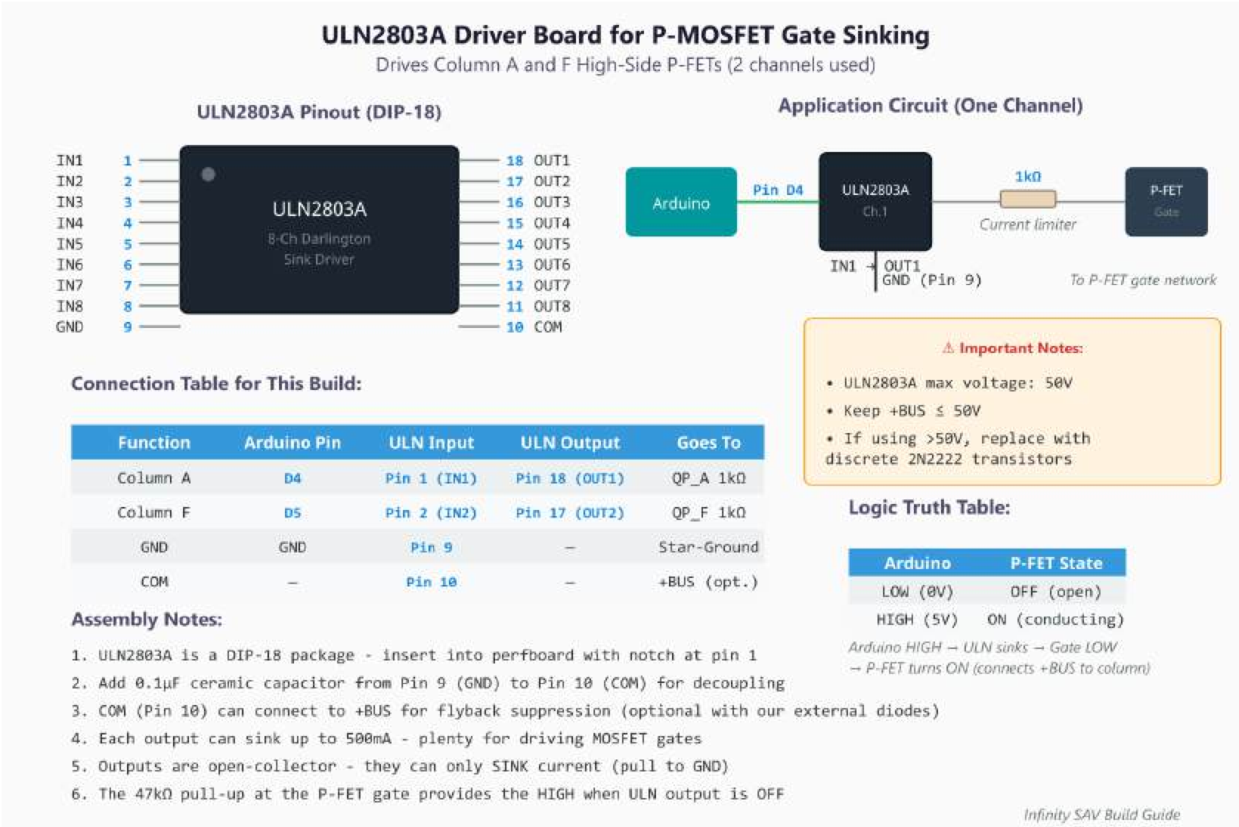
3. Flyback Management (The Turn-OFF)

Components: MUR460 Diode and TVS (1.5KE33A).

Why: When the switch turns OFF, the coil acts like a flywheel—it tries to keep the current moving, creating a massive high-voltage spike backwards. The MUR460 recycles this energy to the bus, and the TVS absorbs any remaining spike to protect the MOSFET.

In Summary:

- **P‑MOSFET IRF4905** per column on a heatsink bar.
- Source → +BUS, Drain → that column's DIN rail.
  Gate network per device:
  - 10–47 kΩ Gate→+BUS (47 kΩ default; 10 kΩ gives stronger pull-up/noise immunity).
  - Gate resistors: (a) Gate stopper 22–47 Ω placed close to the MOSFET gate, and (b) a series **CURRENT‑LIMIT resistor 1 kΩ to 2.2 kΩ** between the gate node and the ULN2803A sink (prevents Zener/ULN over‑current when +BUS > Zener voltage).
  - 12 V Zener Gate→Source (cathode at Source, anode at Gate) to limit |VGS|.
  - *Why the extra series resistor matters:* When the ULN2803A pulls the gate low, the Zener conducts to clamp VGS. Without a current‑limit resistor, a 12–24 V bus can force very high clamp current. Use 1 kΩ–2.2 kΩ so the clamp current stays in the few‑milliamp range.
- **How it turns ON:** A **ULN2803A** channel **sinks the gate** to ground. The ULN's input is driven from the Arduino (or a PCA9685). The ULN can only **pull down**; the 10 kΩ resistor pulls the gate **up** when we want it OFF.
- Keep +BUS ≤ 50 V for ULN2803A; above that, use a discrete 2N2222 sink stage.



Function	Arduino Pin	ULN Input	ULN Output	Goes To
Column A	D4	Pin 1 (IN1)	Pin 18 (OUT1)	QP_A 1kΩ
Column F	D5	Pin 2 (IN2)	Pin 17 (OUT2)	QP_F 1kΩ
GND	GND	Pin 9	–	Star-Ground
COM	–	Pin 10	–	+BUS (opt.)

Arduino	P-FET State
LOW (0V)	OFF (open)
HIGH (5V)	ON (conducting)

Figure 9-1b: ULN2803A Driver Board Assembly (The "Pilot Valve")

The High-Side Trigger: This board connects the weak 5V signal from the Arduino to the powerful P-MOSFET switches.

- **Mounting:** Place this on the **Door** near the Arduino Mega.
- **Wiring:** Connect Arduino Pin D4 to Input 1, and Pin D5 to Input 2. The outputs go to the P-FET gate networks on the backplate.
- **Safety:** The Ground pin (Pin 9) connects to the system Star-Ground. This completes the circuit that allows the chip to "pull down" the P-FET gates.

9.1c Sub-Component: ULN2803A Driver Board (4×6 cm)

Objective: Build the interface that allows the delicate Arduino (5V) to switch heavy-duty Relays or Solenoids (12V/24V).

The "Flip" Rule:

- **Top View (Placement):** The Chip **Notch** faces **UP**. Pin 1 is Top-Left.
- **Bottom View (Soldering): MIRRORED.** Pin 9 (GND) is on the **Bottom-Right**.
- **Safety:** Never power this board until you confirm **Pin 9 is GND** and **Pin 10 is COM (+V)**. Reversing these blows the chip instantly.

1. Component Placement (Top View)

Use **Figure 9-1c(Top)** to insert your components.

- **Socket:** Always solder an **18-Pin DIP Socket** first. Do not solder the chip directly.
- **Orientation:** The **Notch** on the socket/chip must face **UP**.
- **Inputs (Left):** Header for signals coming *from* Arduino.
- **Outputs (Right):** Header for signals going *to* Relays/Coils.

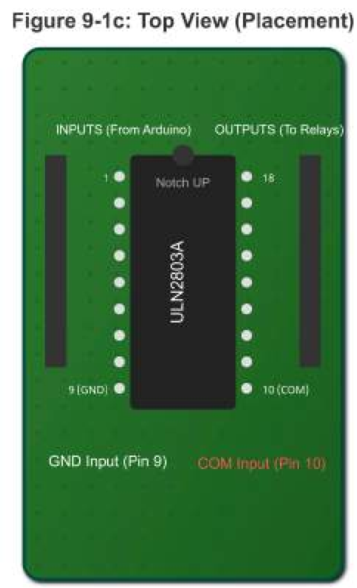


Figure 9-1c (Top): Component Placement. Ensure the Chip Notch faces UP.

---

2. Wiring & Soldering (Bottom View)

Flip the board. This view is **MIRRORED**.

- **Pin 9 (Bottom Right):** Must go to **GND Rail (Grey)**.
- **Pin 10 (Bottom Left):** Must go to **COM Rail (Copper)**.
- **Inputs:** Connect to the Right side pins (1–8).
- **Outputs:** Connect from the Left side pins (18–11).

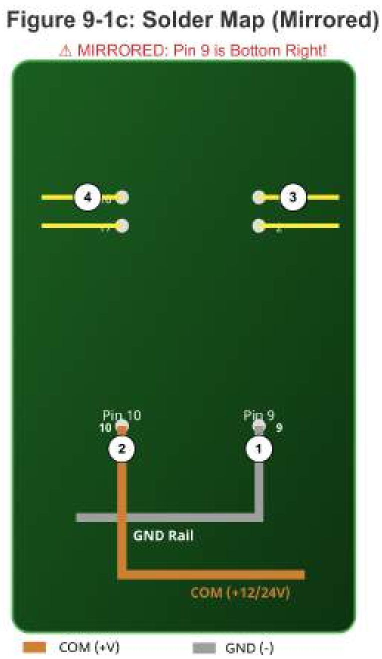


Figure 9-1c (Bottom): Mirrored Solder Map. Critical: Pin 9 is GND, Pin 10 is COM.

3. Assembly Netlist (The Instructions)

Table 9-1c: Step-by-Step Soldering Guide

ID	Action	Source (Component Leg)	Destination (Target)	Critical Note
#1	**Solder GND**	**Pin 9 (Bottom Right)**	To Grey GND Rail	**CRITICAL:** Pin 9 must go to Ground.
#2	**Solder COM**	**Pin 10 (Bottom Left)**	To Copper COM Rail	Connects to Relay Voltage (+12V or +24V).
#3	**Solder Inputs**	Input Header (Right)	To Pins 1, 2... (Right Side)	From Arduino (5V Logic).

#4	**Solder Outputs**	Output Header (Left)	From Pins 18, 17... (Left Side)	To Relay Coil Negatives (Switched Ground).

4. Usage & Safety Note

- **How it Works:** When Arduino sends **HIGH** to Pin 1, the ULN2803A connects Pin 18 to **GND**. This completes the circuit and turns on your relay.
- **The COM Pin (Pin 10):** This pin connects to internal "Flyback Diodes." You must connect the **COM Rail** to the **Positive Voltage** of your relays (e.g., +12V or +24V).
  - **WARNING:** Do **NOT** connect COM to 5V. It must match the high-voltage supply of the load you are switching.

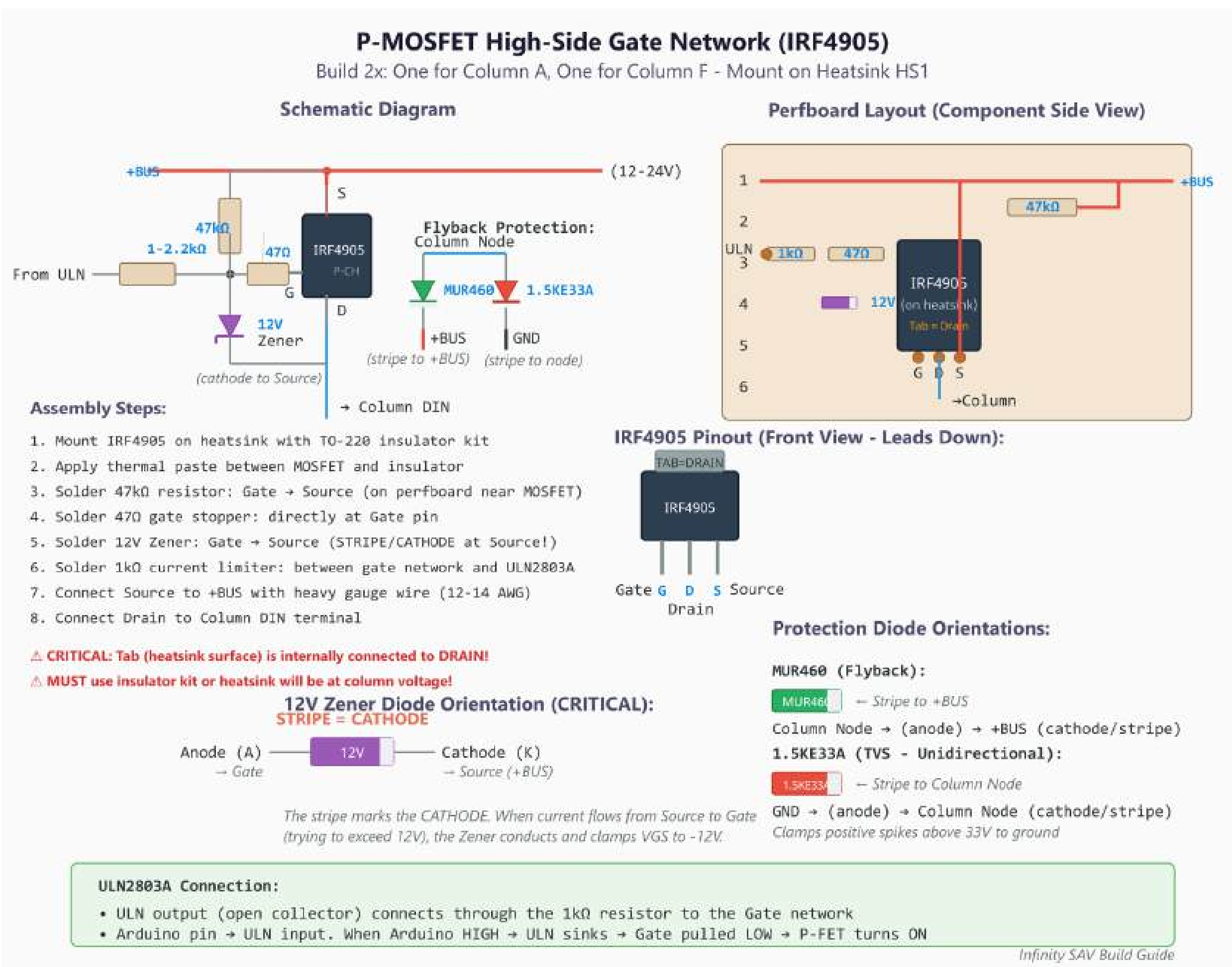


Figure 9-1d: P-MOSFET Gate Safety Network (Mounts on Heatsink HS1)

The "Master Valve" Safety Circuit: You cannot wire the ULN2803 directly to the P-MOSFET. You must solder these protection parts directly onto the MOSFET legs or a small board right next to it.

- **12V Zener:** Prevents the gate voltage from dropping too low and rupturing the chip. **Stripe must face the Source (+24V) pin.**
- **1kΩ Resistor:** Limits the current so the Zener doesn't burn out when it activates.
- **47kΩ Resistor:** The "Return Spring" that keeps the valve closed (OFF) if the Arduino is disconnected.

9.2 Low-side row selector (5 rows: 1…5)

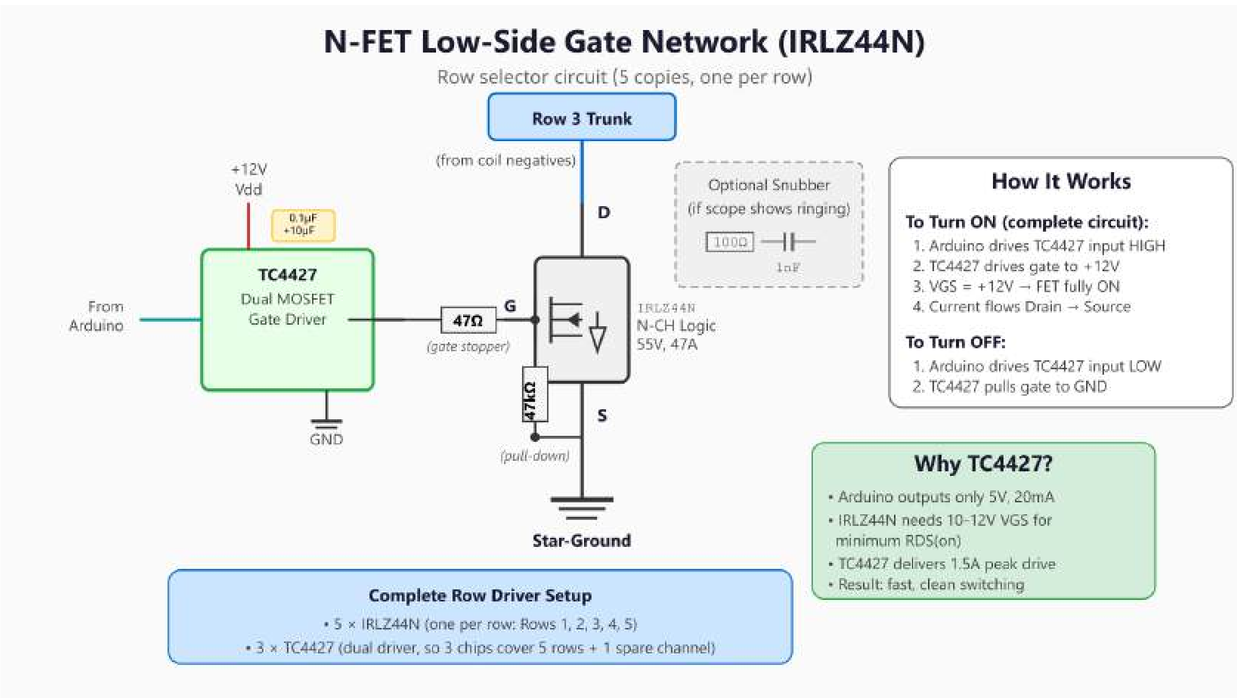


Figure 9-2a — N-FET Low-Side Gate Network

Deep Dive: Building the "Floor Drain Trigger" (N-FET Low-Side Switch)

In the main **Push Coil Diagram**, we showed the **N-MOSFET (IRLZ44N)** as the "Floor Drain" at the bottom of each row.

- **The Challenge:** The Arduino is like a weak manager. It can whisper a command, but it doesn't have the muscle to slam open a heavy industrial valve (the MOSFET) instantly. If the valve opens slowly ("creaks" open), the rushing water (current) creates friction and burns the valve up.

- **The Solution:** We need a **Hydraulic Booster (TC4427)**.

1. The Muscle (TC4427 Gate Driver)

- **Analogy: The Hydraulic Ram.**
    - **The Problem:** The Arduino (Manager) only has 5V of "strength" and very weak arms (low current). It opens the valve too slowly.
    - **The Fix:** The TC4427 is a powerful hydraulic ram connected to a separate **12V supply**. When the Arduino whispers "Go," the TC4427 uses that 12V muscle to **SLAM** the valve lever fully open in nanoseconds.
    - **Result:** The valve goes from "Closed" to "Wide Open" instantly, skipping the dangerous "half-open" zone where heat is generated.

2. Decoupling Capacitors (0.1 μF + 10 μF)

- **Analogy: The Compressed Air Tanks.**
    - **Why:** When the Hydraulic Ram (TC4427) kicks, it needs a massive gulp of energy *instantly*. If it has to pull that energy through long wires from the main supply, it will choke/stutter.
    - **The Fix:** We place small tanks (Capacitors) right next to the Ram. They hold a "shot" of energy ready to go, ensuring the kick is powerful and immediate.

3. Stability Control

- **Gate Resistor (47 Ω):**
    - **Analogy: The Shock Absorber.** Just like with the Master Valve, slamming the drain open too fast can cause vibration ("ringing"). This resistor dampens the shock just enough to keep it smooth.
- **Safety Pull-Down (47 kΩ):**
    - **Analogy: The Return Spring.**
    - **Why:** If the power goes out or the Arduino reboots, the Ram goes limp. We don't want the drain to float open accidentally. This spring ensures that if nobody is holding the lever, it snaps **shut** (Safe State).

4. The Protection (9.3 High-Speed Safety)

- *Note: These parts are physically located near the P-FET (Top of Column) but work with the N-FET to protect the whole line.*
- **MUR460 (Flyback Diode):**
    - **Analogy: The Recirculation Loop.** When you slam the drain shut, the water in the riser tries to crash backward. This pipe guides that crash safely back to the tank.
    - **Location:** Connected from the **Column Node** to **+BUS**.
- **TVS Diode (1.5KE33A):**
    - **Analogy: The Emergency Airbag.** Absorbs any pressure spikes that the

Recirculation Loop misses.
    - **Location:** Connected from the **Column Node** to **Ground**.

Summary for Assembly

You are building the **"Trigger Module"** for each row:

1. **The MOSFET: IRLZ44N** (Logic Level N-FET).
    - **Drain:** Connects to the **Bottom of the Coil** (Row Trunk).
    - **Source:** Connects to **Star Ground**.
2. **The Driver: TC4427** (Dual Driver).
    - **Input:** From Arduino Pin.
    - **Output:** To MOSFET Gate (through 47 Ω resistor).
    - **Power:** Needs **+12V** (NOT 5V, NOT 24V).
3. **The Safety:**
    - **47 kΩ Resistor:** Gate to Ground.
    - **Capacitors:** Close to the TC4427 chip.

How it connects to the Big Picture:

- Step 1: **P-FET (Master Valve)** Pressurizes the Column.
- Step 2: **N-FET (Floor Drain)** Snaps open to fire the specific coil.
- Step 3: **TC4427 (Muscle)** Ensures the snap happens fast enough to stay cool.

In more technical language, this circuit connects the bottom of the coil to Star-Ground, completing the loop. We use an N-channel MOSFET (IRLZ44N) here because N-FETs are efficient and run cool.

1. The Muscle (TC4427 Gate Driver)

Why not use the Arduino directly? The Arduino only outputs 5 V and very weak current. A MOSFET Gate acts like a capacitor (a bucket) that needs to be filled with electrons instantly to turn ON. If you fill it slowly (using just the Arduino), the MOSFET spends too much time "half-open," generating dangerous heat.

The Solution: The TC4427 takes the weak 5 V command and blasts a strong 12 V signal into the MOSFET. This snaps the switch fully ON instantly, keeping it cool.

Decoupling Capacitors (0.1 μF + 10 μF): These are placed right next to the Driver. They act like local "energy tanks," providing the sudden gulp of power the driver needs to snap the gate open without dipping the voltage rail.

2. Stability Control

Gate Resistor (47 Ω): Placed between the Driver and the MOSFET. It slows down the switching speed *just enough* to prevent "ringing" (electronic echoing) without causing heat.

Safety Pull-Down (47 kΩ): Connects Gate to Ground. If the Driver loses power or the Arduino is rebooting, this resistor ensures the Gate bleeds off any charge and stays OFF.

In Summary:

- **N-MOSFET IRLZ44N** per row.
- Drain → that row's negative trunk, Source → Star-Ground.
- Gate from **TC4427** driver through **22–68 Ω**; add **10–100 kΩ Gate→Source** as a pull-down.
- Decouple each TC4427: **0.1 µF + 10 µF** at its Vdd pins.

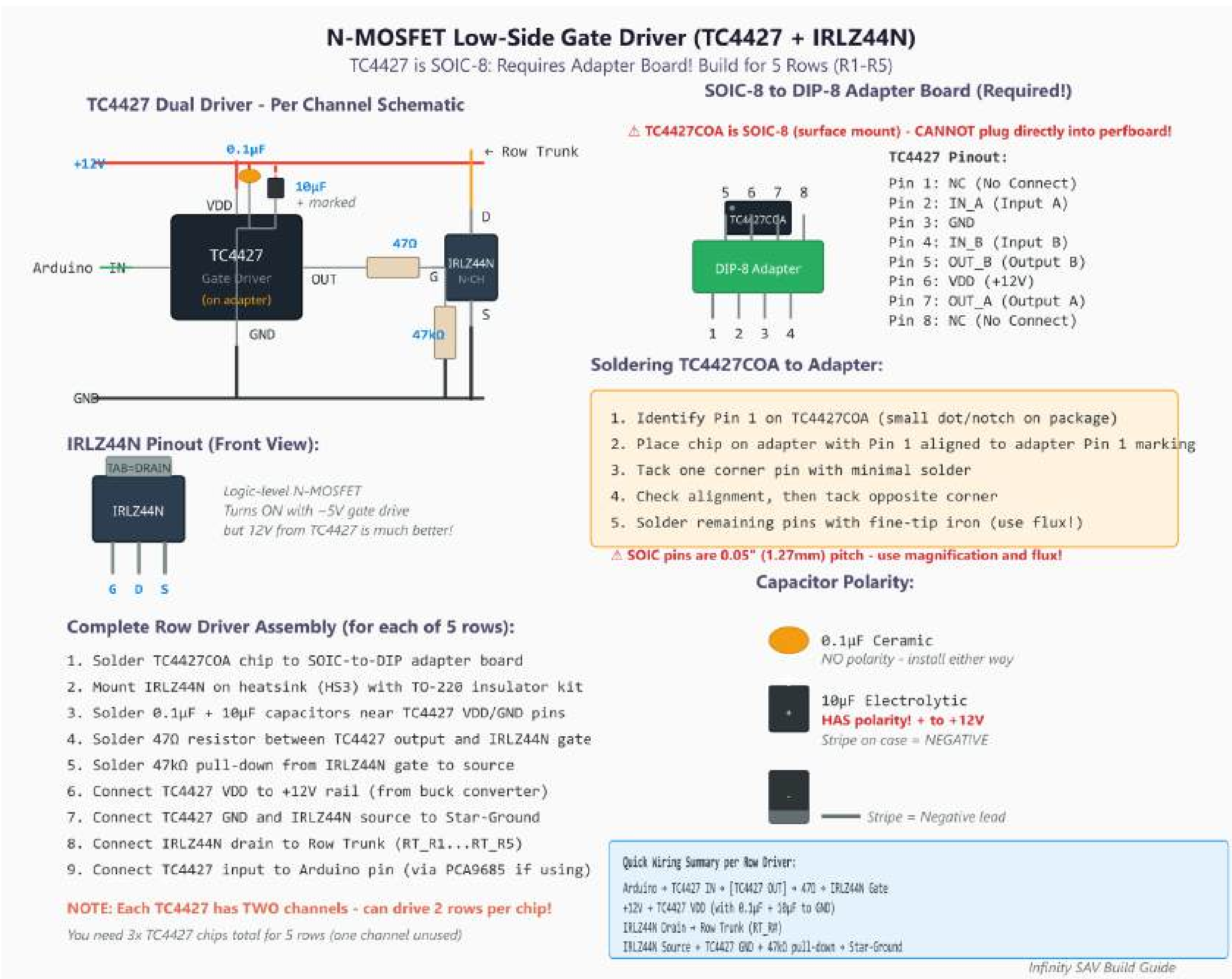


Figure 9-2b: N-MOSFET & TC4427 Driver Assembly (Mounts on Heatsink HS3)

The "Muscle" Board (Row Drivers): This circuit slams the row switches (IRLZ44N) open and closed instantly to prevent heat.

- **Adapter Required:** The TC4427 chip is tiny (Surface Mount). You must solder it to the **DIP-8 Adapter Board** before you can use it. Use flux and a magnifier!

- **Placement:** Mount this board on the **Backplate**, right next to the Row MOSFETs (Heatsink HS3).
- **Distance Rule:** The gate wire between this board and the MOSFET must be as short as possible (**target <10 cm**). Long wires will cause oscillation.
- **Capacitors:** The 0.1μF and 10μF capacitors act as "Air Tanks" for the driver. They must be soldered as close to the chip as possible.

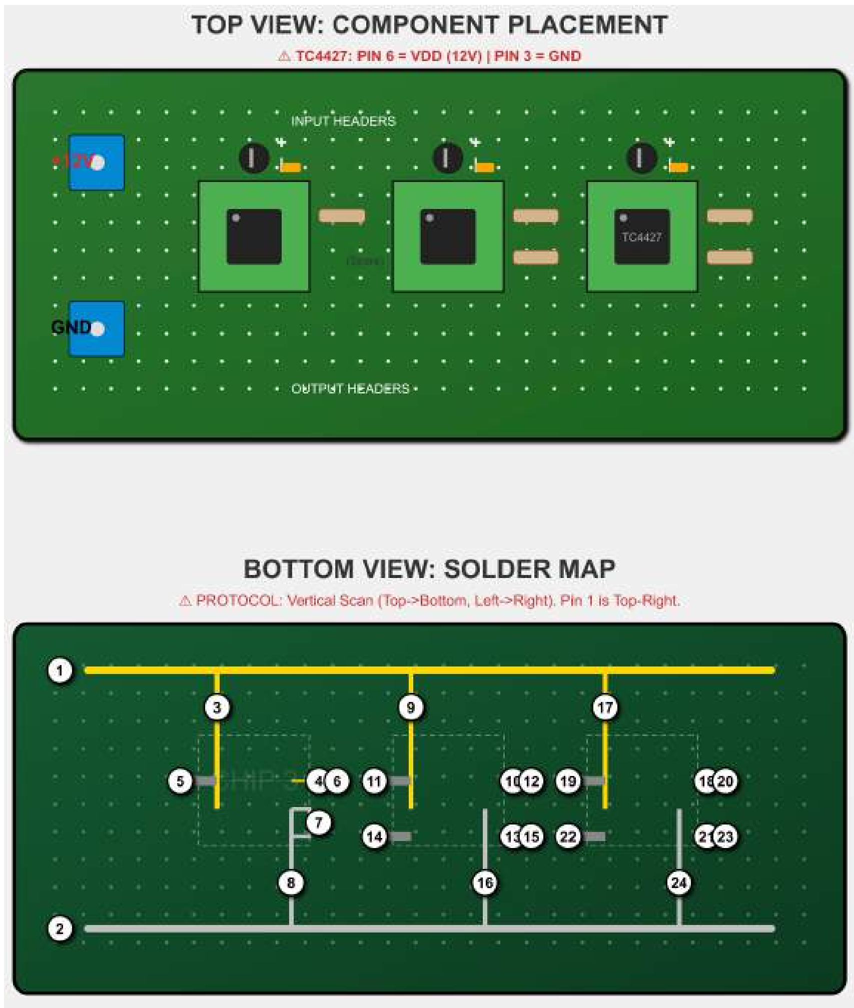


Figure 9-2c: N-FET Gate Driver Perfboard Layout (3x7 cm)

Student Build Guide: N-FET Driver Board Assembly

Standard: IPC Class 2 Reliability **Objective:** Build a high-speed, high-current driver board capable of surviving vibration and thermal stress. **Reference Artifact: Figure 9-2c (Double-View Solder Map)**

Phase 1: The "Mise-en-place" (Preparation)

Professional Standard: 90% of soldering defects are caused by oxidation or dirt.

1. **Clean the Board:** Scrub the copper side of the perfboard with a Scotch-Brite pad or steel wool until it shines bright gold/pink. Wipe with Isopropyl Alcohol (99%). **Do not touch the copper with bare fingers after this.**
2. **Flux is Life:** If the solder won't stick, you aren't using enough flux. Use a flux pen on the pads before applying heat.
3. **The "Third Hand":** Secure the board in a vise or "helping hands." You cannot hold the board and solder professionally at the same time.

Phase 2: Surface Mount First (The Hard Part)

Logic: Always solder the lowest-profile components first. If you solder the tall headers first, you can't flip the board flat to do the chips.

Step 1: Mount the TC4427 Chips to Adapters

- **Orientation Rule:** The **Dot** on the TC4427 chip marks **Pin 1**. All pin numbers in this guide assume the chip is oriented exactly as shown in **Figure 9-2c (Top View)**.
- **Goal:** Solder the tiny SOIC-8 chips onto the Green DIP-8 Adapter boards.
- **Technique:** "Tack and Drag."
    1. Tin **one** pad on the green adapter.
    2. Slide the chip in with tweezers (Align the **Dot** with the **Pin 1** mark).
    3. Re-heat the tinned pad to lock the chip in place.
    4. Solder the remaining 7 pins.
    5. **Inspect:** Use a magnifying glass to ensure no solder bridges connect two adjacent legs.

Phase 3: The Power Grid (Creating the Rails)

Logic: High-current paths must be solid copper to handle the 12V load without voltage sag.

Step 2: Create the Bus Bars (IDs #1 & #2)

- **Material:** Strip 15cm of **Solid Core 18 AWG Wire**.
- **Action:** Lay the bare wire along the **Top Row** and **Bottom Row** as shown in the Solder Map.
- **Execution:**
    - **ID #1 (+12V In):** Solder the Red Terminal Block. Run the bare wire from this block across the entire top row. Solder it to the perfboard every 4-5 holes to secure it.
    - **ID #2 (GND In):** Solder the Black Terminal Block. Run the bare wire across the bottom row. Secure it.

Phase 4: Component Population & The "24-Point" Connection

⚠️ ***IMPORTANT DESIGN CHANGES (Don't Panic!)*** *You may notice differences if you compare this to the theoretical schematic (Figure 9-2a). Follow these **Hardware Updates**:*

- ***Resistors:*** *We are using **22Ω** (Red-Red-Black) instead of 47Ω. This creates faster switching speeds for our magnetic application.*
- ***Pull-Down Resistors:*** *The 47kΩ resistors shown in early concepts are **NOT INSTALLED**. The TC4427 chip handles this function internally, so these are redundant.*

Instructions: Insert components from the Top Side. Flip over. Solder connections on the Bottom Side using this specific checklist.

A. Power & Decoupling (The Energy Tank)

*Technique: The "3-Point Bridge". You are joining the **Chip Pin**, the **Capacitor Leg**, and the **Power Rail** into one solid joint.*

- **[ ] ID #17 (Chip 1 +):** Bridge **Pin 6** + **Capacitor(+)** to the Top Rail (+12V).
- **[ ] ID #24 (Chip 1 -):** Bridge **Pin 3** + **Capacitor(-)** to the Bottom Rail (GND).
- **[ ] ID #9 (Chip 2 +):** Bridge **Pin 6** + **Capacitor(+)** to the Top Rail.
- **[ ] ID #16 (Chip 2 -):** Bridge **Pin 3** + **Capacitor(-)** to the Bottom Rail.
- **[ ] ID #3 (Chip 3 +):** Bridge **Pin 6** + **Capacitor(+)** to the Top Rail.
- **[ ] ID #8 (Chip 3 -):** Bridge **Pin 3** + **Capacitor(-)** to the Bottom Rail.

B. Signal Inputs (The Brains)

Technique: Use thin insulated wire (24 AWG) for these signals to avoid shorting against the bare power rails.

- **[ ] ID #18 (Row 1 In):** Solder wire from **Chip 1, Pin 2** to the Input Header.
- **[ ] ID #21 (Row 2 In):** Solder wire from **Chip 1, Pin 4** to the Input Header.
- **[ ] ID #10 (Row 3 In):** Solder wire from **Chip 2, Pin 2** to the Input Header.
- **[ ] ID #13 (Row 4 In):** Solder wire from **Chip 2, Pin 4** to the Input Header.
- **[ ] ID #4 (Row 5 In):** Solder wire from **Chip 3, Pin 2** to the Input Header.
- **[ ] ID #7 (Safety Jumper):** On Chip 3, create a solder bridge connecting **Pin 4** directly to **Pin 3 (GND)**.
  - *Why? This prevents the unused input from "floating" and causing unintended switching.*

C. Gate Outputs (The Muscle)

Technique: Solder the resistor "inline". One leg goes to the chip, the other goes to the output header.

- **[ ] ID #19 & #20 (Row 1):** Solder **Chip 1, Pin 7** to Resistor Leg. Solder other leg to

Output Header.

- **[ ] ID #22 & #23 (Row 2):** Solder **Chip 1, Pin 5** to Resistor Leg. Solder other leg to Output Header.
- **[ ] ID #11 & #12 (Row 3):** Solder **Chip 2, Pin 7** to Resistor Leg. Solder other leg to Output Header.
- **[ ] ID #14 & #15 (Row 4):** Solder **Chip 2, Pin 5** to Resistor Leg. Solder other leg to Output Header.
- **[ ] ID #5 & #6 (Row 5):** Solder **Chip 3, Pin 7** to Resistor Leg. Solder other leg to Output Header.

Phase 5: Professional Inspection (QC)

Pass/Fail Criteria:

1. **The "Volcano" Test:** Look at every joint. It should look like a shiny, concave volcano (curved in). If it looks like a ball (convex), it is a "cold joint" and will fail. Reflow it with flux.
2. **The "Tug" Test:** Gently tug on your power rail wires. They should not move.
3. **The Beep Test:**
    - **Setup:** Set Multimeter to Continuity Mode.
    - **Short Check:** Probe **Top Rail** and **Bottom Rail**. **Silence = Pass.** (Beep = Short Circuit / FAIL).
    - **Power Check:** Probe **Top Rail** and **Chip Pin 6**. **Beep = Pass.**
    - **Ground Check:** Probe **Bottom Rail** and **Chip Pin 3**. **Beep = Pass.**

Recommended Instructional Videos

Note: These videos are optional but highly recommended to visualize the techniques before you begin.

- **Video: How to Solder SOIC Chips to Adapters**
    - *Focus:* Watch 01:40 to 05:00 for the "tack and drag" method used in Phase 2.
- **Video: Creating Perfboard Solder Bridges**
    - *Focus:* Watch 03:10. This shows exactly how to lay down the bare copper wire
    -

9.3 Heatsink Mechanical Assembly (The Skeleton)

Objective: Physically mount the 5 N-Channel MOSFETs (IRLZ44N) to Heatsink HS3. **Why:** You cannot solder the "Floating Bus Bars" (Section 9.4) until the chips are bolted down rigidly to serve as anchors. **Critical Standard: Electrical Isolation.** The metal tab of the MOSFET must **NOT** touch the aluminum heatsink. If it touches, the entire heatsink becomes electrified, causing a massive short circuit.

1. The Layout Strategy (Spacing & Height)

We need the MOSFETs spaced widely enough to solder the protection diodes between them, but close enough to share the rigid copper bus bars.

- **Vertical Position:** Draw a line horizontally across the **center** of the heatsink face.
- **Horizontal Spacing (The Grid):** Mark 5 drill holes along that center line, exactly **25 mm (1 inch)** apart.
    - *Why 25 mm?* The MOSFET is ~10 mm wide. This leaves a **15 mm air gap** between chips, which is the perfect space to fit the protection diodes later without shorting.

2. The "Isolation Sandwich" (Mounting)

You are building a specific stack of parts to let heat flow out while keeping electricity in.

Parts Per MOSFET:

1. **IRLZ44N** MOSFET.
2. **Silicone Thermal Pad** (Grey/Blue square) **OR** Mica Insulator + Thermal Paste.
3. **Plastic Shoulder Washer** (T-shaped washer). **<-- CRITICAL PART.**
4. **M3 Bolt & Nut.**

Step-by-Step Assembly:

1. **Drill & Deburr:** Drill the 5 marked holes (3 mm bit). **Deburr** the holes (remove sharp metal flakes) so they don't puncture the soft insulation pad.
2. **Apply Interface:** Place the **Silicone Thermal Pad** over the hole on the heatsink.
3. **Place Component:** Place the MOSFET on top of the pad.
4. **Isolate:** Insert the **Plastic Shoulder Washer** into the hole of the MOSFET's metal tab.
    - *Check:* The "neck" of the washer must go *inside* the metal tab. This prevents the screw from touching the MOSFET.
5. **Fasten:** Insert the M3 bolt through the washer and tighten the nut on the back.
    - *Torque:* Finger tight plus 1/4 turn. Do not crush the silicone pad.

Figure 9-3a: Heatsink Side-Profile ("Layer Cake" View)

⚠ NOTE: Visual spacing is expanded for clarity. Real lead lengths must be 10mm.

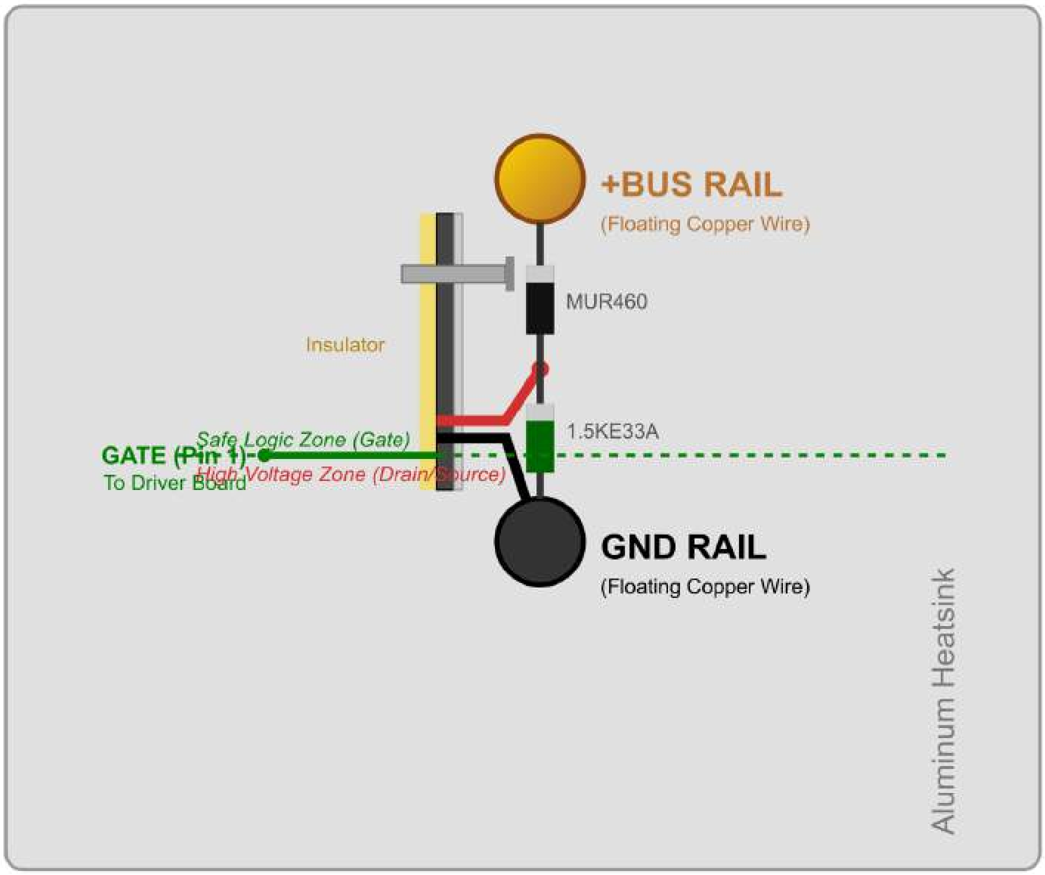


Figure 9-3a: Heatsink Side-Profile (Not to Scale). Note how bending the pins creates safe separation.

3. The "Beep Test" (Safety Verification)

STOP. Do not proceed to wiring until you pass this test for **ALL 5** MOSFETs.

1. Set Multimeter to **Continuity Mode** (Beep).
2. Touch the **Black Probe** to the **Bare Aluminum Heatsink**.
3. Touch the **Red Probe** to the **Center Pin (Drain)** of the MOSFET.
4. **The Verdict:**
    - **Silence:** ✅ **PASS.** The chip is safely floating.
    - **Beep:** ❌ **FAIL.** There is a short. Disassemble, check for metal burrs, and ensure the shoulder washer is seated correctly.

9.4: Protection Circuitry Assembly ("The Bodyguards")

Objective: Install high-speed protection at the "Hot Nodes" to act as shock absorbers for magnetic kickback. **Standard: Zero-Lead-Length.** Components must be mounted within **10mm** of the switching point. **Safety:** These components are **Polarized**. Installing them backward will cause an immediate short circuit and fire.

Critical Definitions

- Switch Node (MOSFET Drain): This is the MOSFET Drain (the metal tab or center pin). It is the "Hot Node" where the magnetic energy collapses.
- **+BUS:** The main positive power rail (Copper Bus Bar).
- **Star Ground:** The main negative return rail.

> **Why the change?** In this project, we use "Low-Side Switching." The MOSFET controls the Ground path for the Row. Therefore, all protection must live at the **MOSFET Drain**.

Assembly Instructions

Step 1: Install the Fast Diode (MUR460)

- **Role:** The "Flyback" Valve. Returns kickback current to the tank.
- **Connection:**
  - Anode (Black Body): Connect to **Switch Node (MOSFET Drain)**. This is the center pin/tab of the MOSFET.
  - **Cathode (Striped End):** Connect to **+BUS**.
- **Verify:** The Stripe must face the **Positive Power Rail**.

Step 2: Install the TVS Diode (1.5KE33A)

- **Role:** The "Shock Absorber." Clamps micro-second voltage spikes.
- **Connection:**
  - Cathode (Striped End): Connect to Switch Node (MOSFET Drain) (MOSFET Drain).
  - **Anode (Unmarked):** Connect to **Star Ground**.
- **Crucial Detail: Keep leads < 10mm.** Every millimeter of wire adds inductance. If the wire is too long, the protection reacts too slowly.

Visual Guide: Polarity Check Use this map to verify your connections. Use Figure 9-3b to align the stripes. Do not guess. Note that the components are drawn spread out for clarity, but the connections match your "Dead Bug" build exactly.

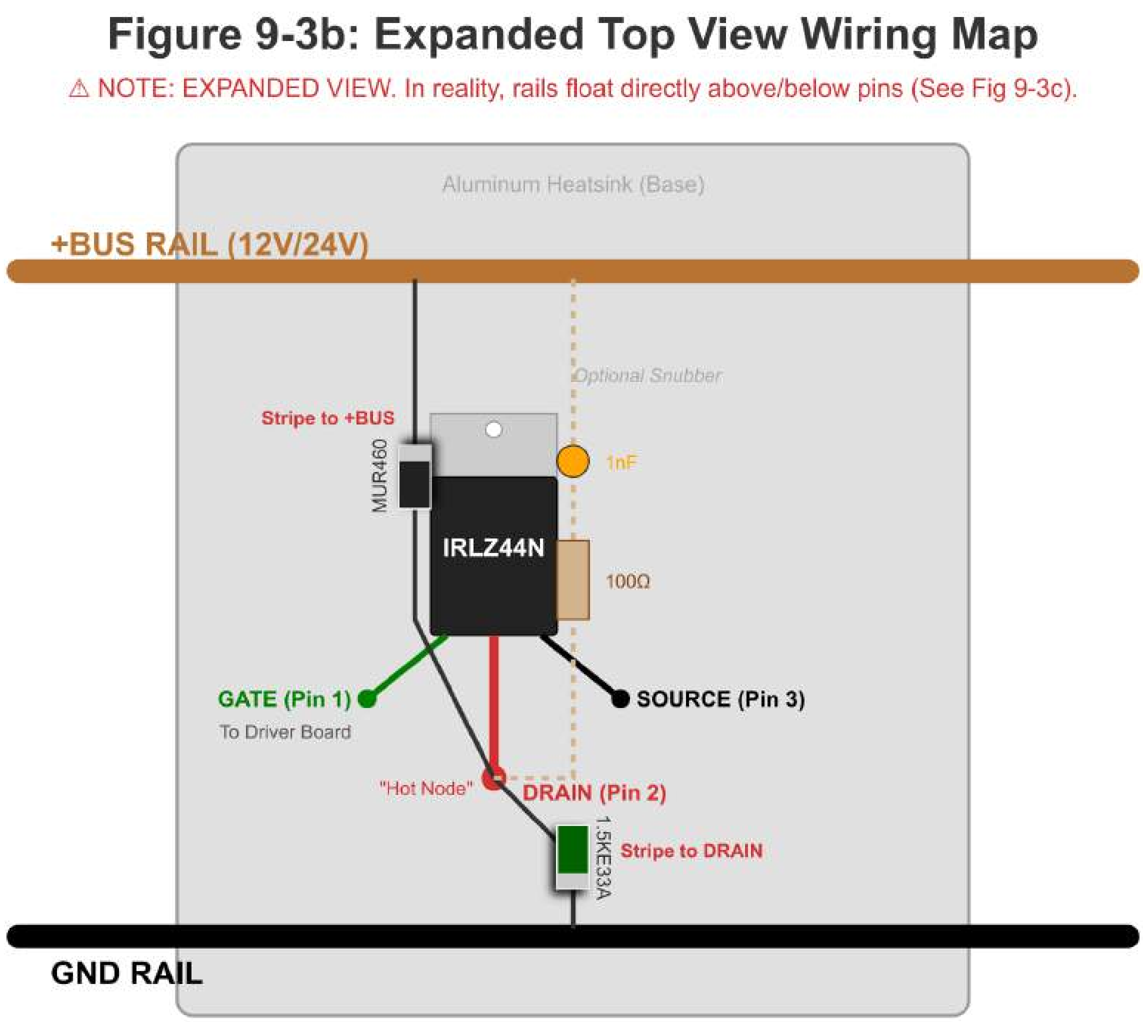


Figure 9-3b: Protection Wiring Map (Expanded View). **CRITICAL:** Ensure the MUR460 stripe faces the +BUS and the TVS stripe faces the Drain.

Step 3: The RC Snubber (Conditional)

- **Status: Do not install initially.** Only install if oscilloscope testing reveals excessive ringing.
- Connection: Series connection (Resistor + Capacitor) from **Switch Node (MOSFET Drain)** to **+BUS**.

9.5 System Integration ("The Umbilical Cord")

Objective: Connect the **Driver Board** (The Brains) to the **Heatsink Assembly** (The Muscle).

The Danger: The wires connecting these two boards carry the "Fire" signals. If these wires are loose, long, or sloppy, they will act like **antennas**. They will pick up magnetic noise from the generator, causing the MOSFETs to "ghost fire" (turn on by themselves), which leads to explosions.

The Solution: We use **Twisted Pair Wiring** to cancel out noise.

1. Material Selection (Use Your Inventory)

You must use the **18 AWG Red/Black Hookup Wire** from your kit.

- **Why 18 AWG?** Ideally, gate signals use thinner wire (22 AWG), but your 18 AWG is robust, holds a twist well, and ensures a solid low-resistance connection.
- **Do NOT use:** Dupont jumper wires (they will vibrate loose) or single loose wires (they act as antennas).

2. The "Twisted Pair" Rule

You are not running single wires. You are running **pairs**.

1. **Cut:** Cut 5 segments of Red/Black wire. Keep them **short** (max 10–12 cm).
2. **Separate:** Pull the Red and Black apart slightly at the ends (2 cm) to strip the insulation.
3. **Twist:** Twist the red and black wires around each other tightly (about 1 twist per inch).
   - *Physics Check:* Twisting the wires forces any magnetic noise to hit both wires equally, cancelling itself out.

3. Wiring the "Umbilical Cords" (Step-by-Step)

Perform this for **Row 1, Row 2, Row 3, Row 4, and Row 5**.

A. Driver Board Side (The Source)

- **Red Wire:** Solder to the **Output Pin** of the TC4427 Driver (via the resistor header).
- **Black Wire:** Solder to the **GND** rail on the Driver Board.
  - *Note:* This creates a dedicated return path for the gate signal.

B. Heatsink Side (The Destination)

- **Red Wire (Signal):** Solder to **Pin 1 (Gate)** of the MOSFET.
  - *Visual Check:* This is the pin you left **straight** in Section 9.3.
- **Black Wire (Return):** Solder directly to the **Source Pin (Pin 3)** where it touches the Ground Bus Bar.
  - *Critical:* Do not run this black wire to the main Star Ground stud. Connect it **directly** to the MOSFET leg. This minimizes "Ground Bounce."

4. Routing & Clearance Check (The Final Exam)

Before you declare victory, look at your "Air Wiring."

1. **No Sagging:** The Copper Bus Bars must be rigid. If you push on them with a finger, they should not touch the heatsink.
2. **Clearance:** Ensure the Red/Black twisted pairs are **floating**. They must not rest against the hot copper +BUS rail.
   - *Tip:* Use a small zip-tie to bundle the 5 twisted pairs together *away* from the high-voltage rails.
3. **The Tug Test:** Gently tug on your twisted pairs. If the Gate pin (Pin 1) moves or feels loose, re-solder it. A disconnected Gate pin is a "floating gate," which guarantees a blown MOSFET.

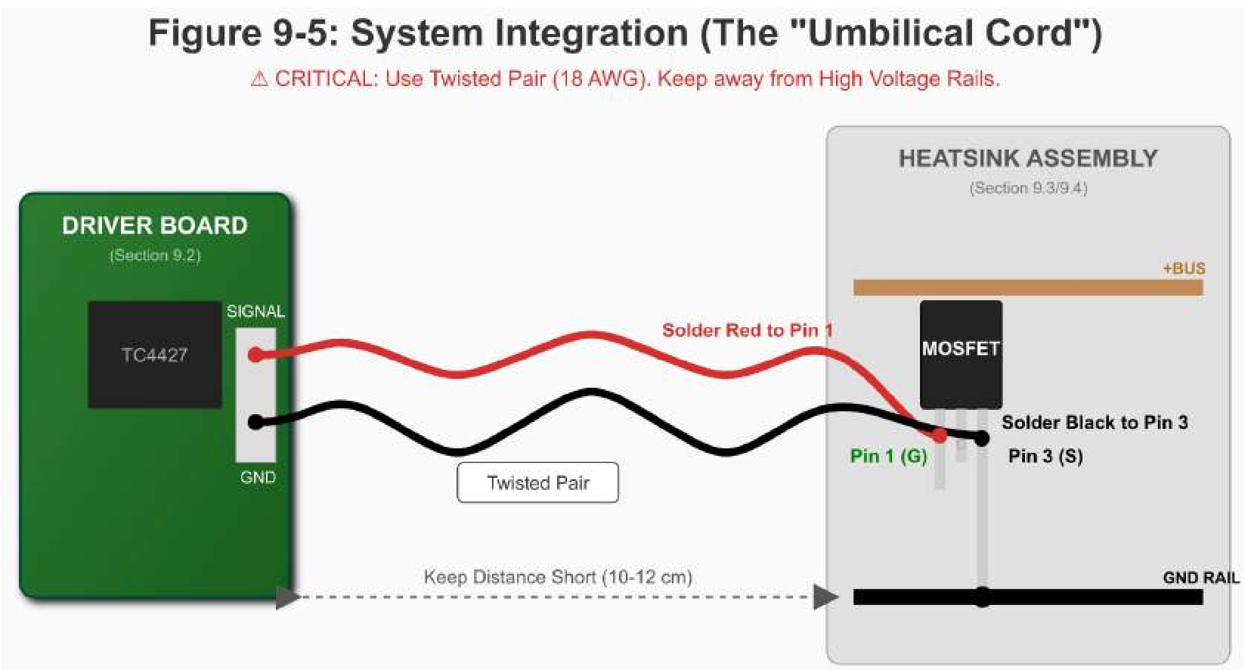


Figure 9-5: System Integration. Note the twisted pairs connecting the Driver Board (Left) to the Heatsink (Right).

Summary: You have now completed the Switching Module.

1. **Skeleton:** MOSFETs mounted and isolated.
2. **Muscle:** Copper rails and protection diodes installed.
3. **Nerves:** Twisted pairs connecting the Driver to the Gates.

If you are running short on time, you may skip “One-Hot Intersection Switching”, and “Push Switching Matrix” and Proceed to Section 9.8: Electronics Build Order Checklist to verify your work.

9.6 High-Side P-MOSFET Assembly (The "Push" Columns)

Objective: Mount and protect the P-Channel "Push" switches (IRF4905) on the horizontal heatsink.

The "Flip" Rule:

- **Side Profile (Figure 9-6a):** Shows how to bend the pins to create the "Safe Zone."
- **Top View (Figure 9-6b):** Shows the wiring layout. **Pin 1 is LEFT** in this view.
- **Critical:** You must perform the **Beep Test** (Heatsink Isolation) *before* soldering.

1. Mechanical Assembly (Side Profile)

Use **Figure 9-6a** to bend your pins correctly.

- **The "Z-Bend":** Pin 1 (Gate) and Pin 3 (Source) must be bent **UP** and **OUT**. They must float in the air, creating a "Safe Zone" away from the heatsink.
- **Pin 2 (Drain):** Keep straight and short. This ensures the metal tab sits flat on the Keratherm pad.

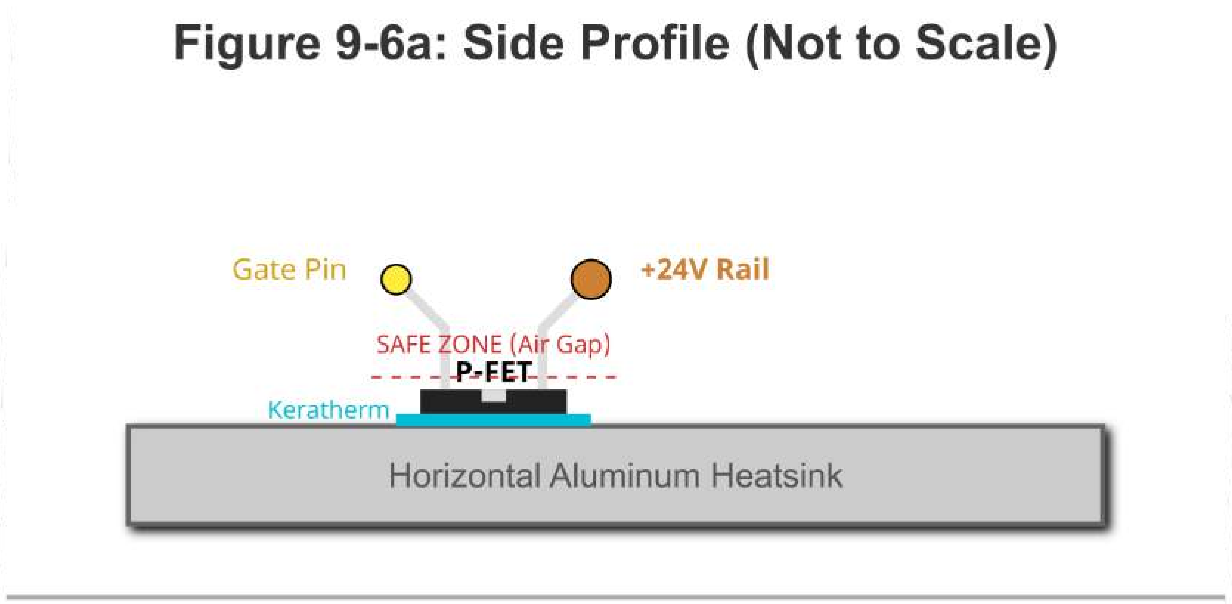


Figure 9-6a: Side Profile (Not to Scale). Note the "Safe Zone" air gap for Pins 1 and 3.

2. Wiring & Protection (Top View)

Use **Figure 9-6b** to wire the "Muscle" and "Nerves."

- **Expansion:** Mount the two chips on the **Left**, leaving the Right side open for future columns.
- **The Rail (+24V):** A rigid **Solid Copper Wire** connects all Pin 3s (Source). This is your main power feed.
- **The Nerves (Twisted Pairs):**
  - **Red Wire:** Goes to Pin 1 (Gate).
  - **Black Wire:** Goes to Pin 3 (Source/Rail). *This creates the local return path.*

- **Safety Pack:** Solder the **10kΩ Resistor** and **15V Zener** directly to the legs (Pin 1 to Pin 3).

Figure 9-6b: Top View (Wiring)

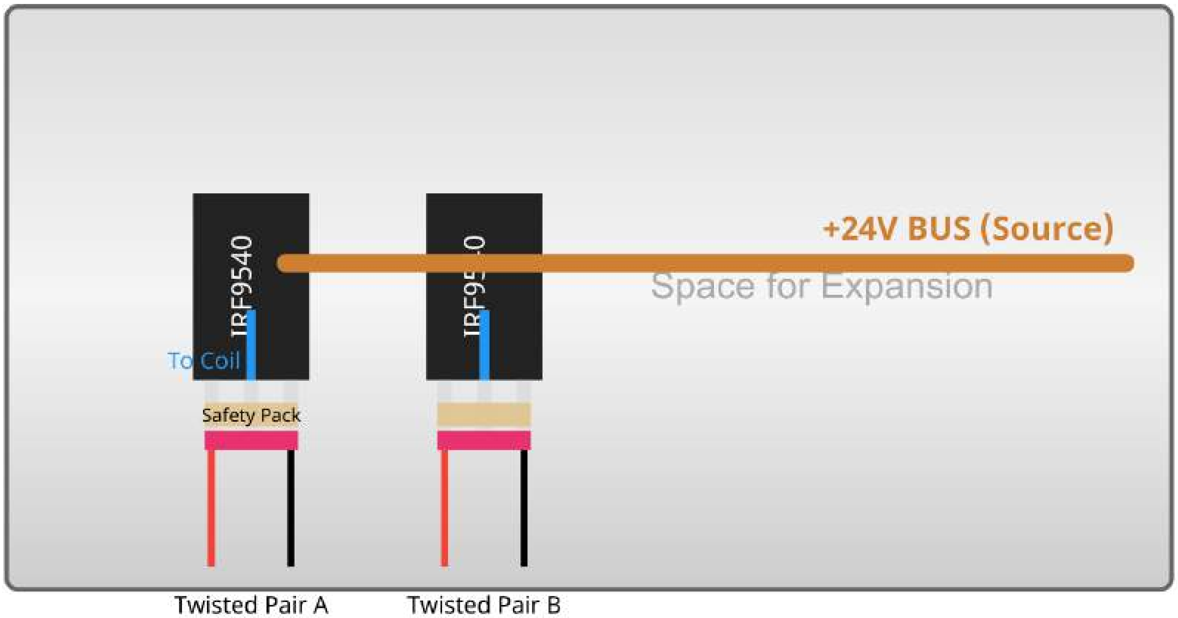


Figure 9-6b: Top View Wiring Map. The "Safety Pack" (Resistor + Zener) is soldered directly to the MOSFET legs.

3. Assembly Netlist (The Instructions)

Table 9-6: Step-by-Step Soldering Guide

ID	Action	Source	Destination	Critical Note
#1	**Create +BUS**	**Solid Copper Wire**	Connect all **Pin 3s** (Right Leg)	This is the +24V Rail. Keep it rigid.
#2	**Safety Pack**	**10kΩ + Zener**	Across **Pin 1** (Left) & **Pin 3** (Right)	Solder directly to legs. **Zener Stripe to Pin 3.**

#3	**Signal Wiring**	**Twisted Pair**	Red $\to$ Pin 1 (Gate) Black $\to$ Pin 3 (Source)	Twist wires tightly! Black connects to the +24V rail.
#4	**Output**	Thick Wire	From **Pin 2** (Center)	To Coil Positive (+).

4. Safety Logic

- **Why Local Resistors?** P-Channel MOSFETs are dangerous. If the gate wire breaks, they turn ON. The 10kΩ resistor on the legs ensures they stay **OFF** by default.
- **Why Twisted Pair?** High-Side switching creates massive noise. Twisting the Signal (Red) with the Reference (Black/+24V) cancels this noise and prevents "ghost firing."

9.7 One-Hot Intersection Switching

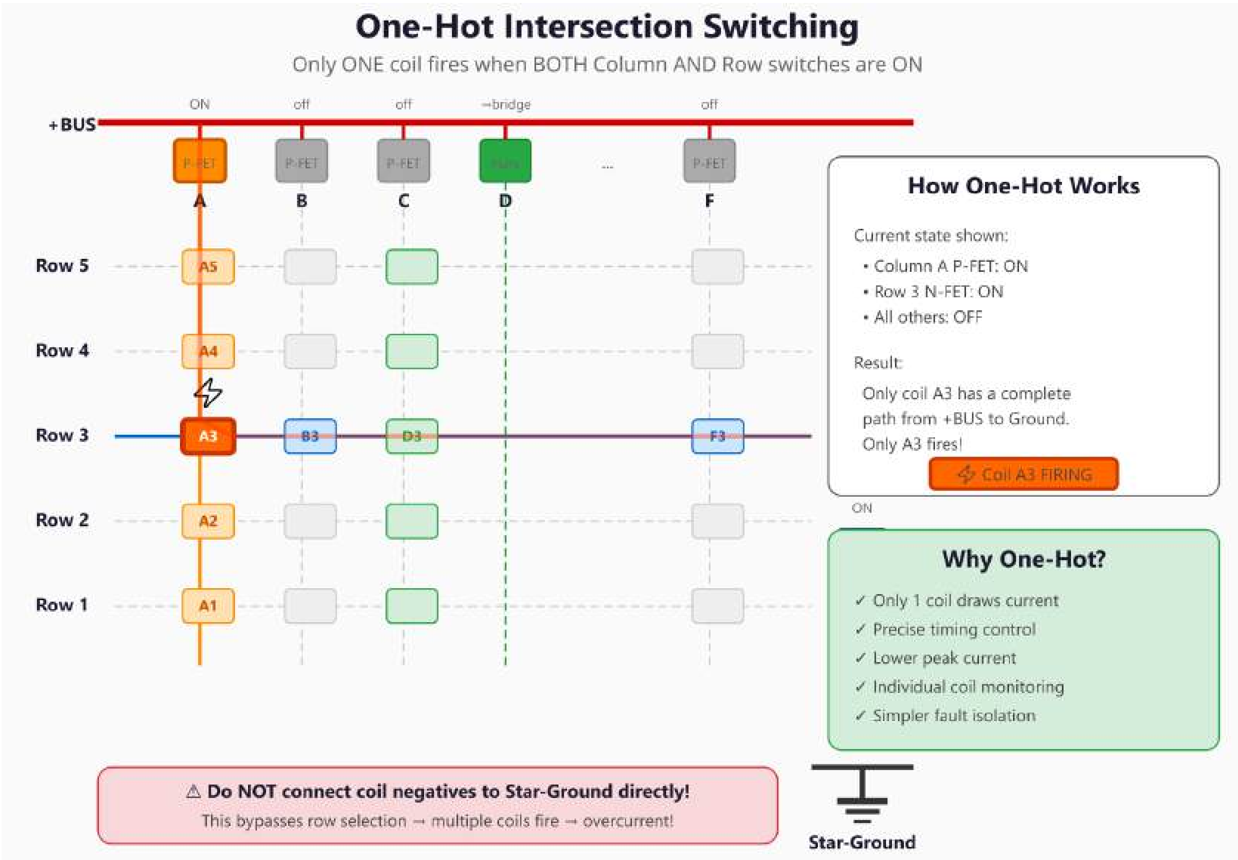


Figure 9-4: One-Hot Intersection Switching (safe single-coil push).

We select **one column** using a high-side P-channel MOSFET, and we select **one row return** using a low-side N-channel MOSFET. Only one row return is allowed ON at a time (“one-hot”). This guarantees that only a single coil is energized during a push pulse. Coil negative leads must not connect directly to star ground, because that would bypass row selection and can energize multiple coils at once.

The "Sniper Scope" Strategy (One-Hot Intersection Switching)

To maximize efficiency, we never fire all the coils at once. Instead, we use a strategy called **"One-Hot Intersection Switching."** Think of it like a sniper scope crosshair or a game of Battleship.

1. The Concept: Coordinates

Imagine your coils are laid out in a grid.

- **Vertical Lines (Columns):** Controlled by the **P-MOSFET** (Building Master Valve).
- **Horizontal Lines (Rows):** Controlled by the **N-MOSFET** (Floor Drain).

To fire a specific coil—for example, **Coil A3**—the Arduino must activate two switches simultaneously:

1. **Column A Switch (ON):** Pressurizes the entire vertical "A" pipe.
2. **Row 3 Switch (ON):** Opens the drain for the entire horizontal "3" pipe.

2. The Intersection

- **Coil A3** sits at the exact intersection of Column A and Row 3.
- Because **both** its supply valve and its drain valve are open, current flows through it violently.
- **Coil A2** is pressurized (because Column A is ON), but its drain (Row 2) is closed. So, it sits silent.
- **Coil B3** has an open drain (because Row 3 is ON), but its supply (Column B) is OFF. So, it sits silent.

Result: Only the single coil at the "crosshairs" fires.

3. Why "One-Hot"?

"One-Hot" is an engineering term meaning "Only one item in the group is allowed to be ON at a time."

- **The Rule:** We never turn on two Rows at the same time.
- **The Benefit:** This guarantees that the push is focused and precise. If we fired multiple coils, the magnetic fields might fight each other or waste energy pushing the rotor when it's not in the perfect position.

4. Critical Warning: The "Short Cut" Danger

> **WARNING: Coil negative leads must NEVER connect directly to Star Ground.**

- **The Mistake:** Wiring a coil's negative wire directly to the ground screw (bypassing the N-MOSFET).
- **The Consequence:** You lose the "Horizontal" control. As soon as you turn on the Column P-MOSFET, *every coil in that column* will fire at once.
- **The Damage:** This draws massive current (5x normal), drains your capacitor instantly, and will likely melt the P-MOSFET. Always route the negative wire to the **Row N-MOSFET**, never directly to ground.

9.8 Push Switching Matrix

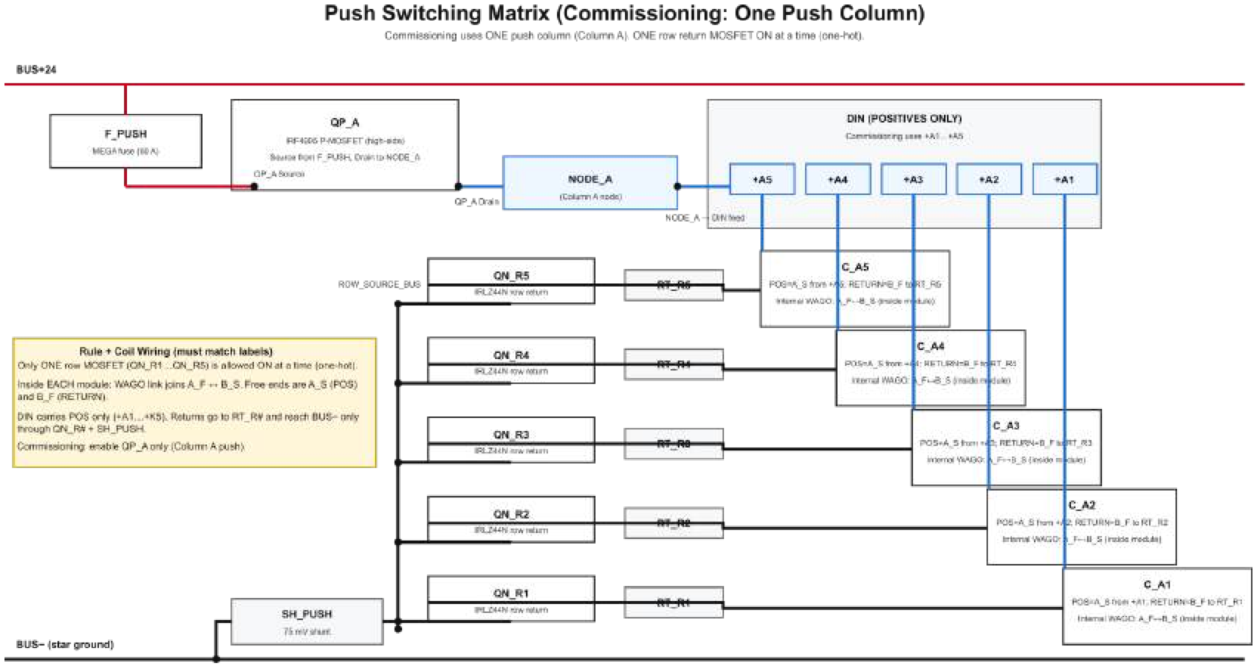


Figure 9-3 — Push Switching Matrix for Commissioning.

The Push Switching Matrix (Commissioning Mode)

This diagram shows exactly how to wire **Column A** for testing. It proves that we can surgically target *one single coil* to fire by opening the correct "Master Valve" (Column Switch) and the correct "Floor Drain" (Row Switch).

1. The Components (Decoding the Blocks)

- **BUS+24 (Top Red Line):** The **Main Water Supply**. It holds the pressure.
- **F_PUSH:** The **Main Circuit Breaker**. If a short circuit happens, this fuse blows to save the wires.
- **QP_A (P-FET):** The **Column Master Valve**.
  - When this opens, it pressurizes the *entire* Column A (all 5 coils get voltage at their input).
  - *Note:* Just because they have voltage doesn't mean they fire. They still need a drain path.
- **DIN (Positives Only):** The **Distribution Manifold**.
  - To keep wiring clean, we only bring the **Positive** wires (A_{Start}) to these top terminals.
- **C_A1 … C_A5:** The **Coils** (The Water Wheels).
- **QN_R1 … QN_R5 (N-FET):** The **Floor Drains**.
  - These determine *which* specific row gets to complete the circuit.
- **SH_PUSH (Shunt):** The **Flow Meter**.
  - This is a precision resistor that measures exactly how much current is flowing. This is critical for commissioning—it proves the coil is actually firing.

2. The "Internal WAGO" (The U-Turn)

This is the most common student wiring error. Your coils have **two separate windings** (Wire A and Wire B). To get maximum power, we must connect them in **Series-Aiding**.

- **The Problem:** You have 4 wires coming out of the coil, but only 2 connection points in the diagram.
- **The Fix (The U-Turn):** Inside the coil housing, you must use a WAGO connector to join **Wire A Finish (A_F)** to **Wire B Start (B_S)**.
- **The Result:**
    - **Input:** You connect voltage to **A_{Start}** (Positive).
    - **Internal:** Current flows through Coil A $\rightarrow$ U-Turn WAGO $\rightarrow$ Coil B.
    - **Output:** Current exits from **B_{Finish}** (Return).

3. Tracing the "One-Hot" Path

Let's trace the path when we want to fire **Coil A3**.

1. **The Source:** Current leaves **BUS+24** $\rightarrow$ through **F_PUSH**.
2. **Column Selection:** We turn **ON** the Master Valve (**QP_A**).
    - *Result:* Voltage sits at the top of Coils A1, A2, A3, A4, and A5.
3. **The Coil:** Current enters **Coil A3** at DIN terminal **+A3**.
    - It travels through the internal windings (A $\rightarrow$ WAGO $\rightarrow$ B).
4. **Row Selection:** We turn **ON** only the Row 3 Switch (**QN_R3**).
    - *Result:* The water (current) finds an open drain at Row 3.
    - *Others:* Coils A1, A2, A4, and A5 are blocked because switches QN_R1, R2, R4, and R5 are OFF.
5. **The Return:** Current flows through **QN_R3** $\rightarrow$ **ROW_SOURCE_BUS** $\rightarrow$ **SH_PUSH** $\rightarrow$ **BUS-**.

4. Student Commissioning Checklist

Before turning on the power for the first time, check these three things:

1. **The U-Turn Check:** Open one coil module. Verify that the **Pink (A_F)** wire is connected to the **White (B_S)** wire via a WAGO. If this is missing, the coil is an open circuit and will do nothing.
2. **The "One-Hot" Code:** Ensure your Arduino code never turns on two **QN_R** switches at the same time. If QN_R3 and QN_R4 are both ON, two coils fire, doubling the current and weakening the push.
3. **The Safety Ground:** Verify that the **Row Trunks (RT)** are NOT connected directly to the Star Ground. They *must* go through the MOSFETs (QN_R). If you bypass the MOSFET, the coil will be "Always ON" as soon as the Column Switch opens.

In more technical terms, this diagram is the **“push” wiring for commissioning** when you only use **one push column (Column A)**. It shows how you can energize **one coil at a time** and measure the push current safely.

What does each labeled block mean?:

- **BUS+24** (top red line): the **positive 24 volt direct current bus**.
- **F_PUSH**: the **main push fuse**. It protects the push circuit if something shorts.
- **QP_A**: the **high-side P-channel metal oxide semiconductor field effect transistor** for **Column A**.
  Think of it as an electronically controlled switch that connects BUS+24 to Column A.
- **NODE_A**: the **Column A node**. It is the “distribution point” for Column A.
- **DIN (POSITIVES ONLY)**: the **terminal blocks** where you land only the **positive coil leads** for easier wiring and debugging.
  - **+A1 … +A5** are the five DIN terminals for Column A, rows 1–5.
- **C_A1 … C_A5**: the **five coils in Column A** (one per row).
- **RT_R1 … RT_R5**: the **row return trunks**. Each row has its own return wire bundle.
- **QN_R1 … QN_R5**: the **row return N-channel metal oxide semiconductor field effect transistor** switches.
  Each one connects its row trunk to the negative side **only when turned on**.
- **ROW_SOURCE_BUS**: the shared node that collects the outputs of the row return switches.
- **SH_PUSH**: the **push current shunt** (a precision resistor). It lets you measure push current.
- **BUS− (star ground)** (bottom black line): the **single negative return point** for the whole system.

The most important wiring rule (prevents mistakes):

Only ONE row return switch (QN_R1…QN_R5) is allowed ON at a time.
That “one-hot” rule guarantees only **one coil** completes a circuit at a time.

What “Internal WAGO: A_F ↔ B_S” means inside every coil module?

Inside each coil module you have four coil wires:

- **A_S** = A winding start
- **A_F** = A winding finish
- **B_S** = B winding start
- **B_F** = B winding finish

For the default **series** setup inside the module:

- A **WAGO connector** joins **A_F to B_S**.
- The two external leads become:
    - **A_S = POSITIVE lead** (goes to the DIN terminal like +A3)
    - **B_F = RETURN lead** (goes to the row trunk RT_R#)

So: **DIN carries A_S only. Returns are B_F only.**

Current path when you "fire" one coil (example: C_A3):

When you want to energize **C_A3**, current flows like this:

1. **BUS+24 → F_PUSH**
2. **F_PUSH → QP_A (high-side switch)**
3. **QP_A → NODE_A**
4. **NODE_A → DIN terminal +A3**
5. **+A3 → coil C_A3 positive (A_S)**
6. Inside the coil: **A_F is linked to B_S** (the internal WAGO link)
7. **Coil return (B_F) → RT_R3**
8. **RT_R3 → QN_R3** (only if QN_R3 is ON)

9. **QN_R3 → ROW_SOURCE_BUS → SH_PUSH → BUS−**

If **QP_A is OFF**, Column A is dead.
If **QN_R3 is OFF**, the coil has no return path, so it cannot turn on.

Why the diagram is built this way?:

- **DIN positives only**: keeps the panel cleaner and makes tracing problems easier.
- **Row return switches (QN_R#)**: let the controller choose *which row coil* completes the circuit.
- **One-hot rule**: prevents two coils from being energized at once (which can cause confusing behavior and high current).
- **Shunt (SH_PUSH)**: lets you measure push current during tests, so you can catch problems early.

Quick student checklist before powering:

- Confirm every Column A coil has: **DIN (+A#) → A_S** and **B_F → RT_R#**.
- Confirm the **internal WAGO link A_F ↔ B_S** exists in every module.
- Confirm **no row trunk (RT_R#) is tied directly to BUS−**. It must go through **QN_R# and SH_PUSH**.
- Confirm your control code/enables ensure **only one QN_R# is ON at a time**.

Before applying power, complete the following one-page build order checklist.

This checklist ties together all electronics figures in Sections 5–9 and enforces correct build sequence, mounting location, and safety constraints.

Do not skip steps. Do not power the system until every item is verified.

9.9 Electronics Build Order Checklist ("The Pre-Flight Check")

STOP. Do not apply 24V or 12V power until you have physically checked every box below.
Rule: If a step fails, you must fix it immediately. Powering up a failed step will destroy the MOSFETs.

Phase 1: Mechanical & Safety (The Skeleton)

- [ ] **The "Beep Test" Passed:** Multimeter in continuity mode. Black probe on Aluminum Heatsink, Red probe on MOSFET Tab (Drain). **Result = SILENCE (No Beep).**
    - *Status:* All 5 chips isolated.
- [ ] **Mounting Rigidity:** MOSFETs are bolted down tight. They do not wiggle.
- [ ] **Rail Clearance:** The Copper Bus Bars (+BUS and GND) are rigid and floating. They do not touch the aluminum heatsink anywhere.

Phase 2: The Driver Board (The Brains)

- [ ] **Chip Orientation:** The "Dot" on the TC4427 chips matches the "Notch" on the adapter board.
- [ ] **Resistors:** The **22Ω** resistors are installed inline with the output.
- [ ] **Capacitors:** The 0.1µF and 10µF capacitors are soldered close to the chips.
- [ ] **No Solder Bridges:** Checked the back of the perfboard with a light. No accidental shorts between tracks.

Phase 3: Protection Circuitry (The Muscle)

- [ ] **MUR460 Polarity:** The **Stripe** faces the **+BUS Rail**.
- [ ] **TVS Polarity:** The **Stripe** faces the **Drain (Center Pin)**.
- [ ] **Lead Length:** All diode legs are shorter than 10mm.

Phase 4: System Integration (The Nerves)

- [ ] **Twisted Pairs:** All 5 Gate/Source wire pairs are tightly twisted (approx. 1 twist per inch).
- [ ] **The "Source Return" Rule:** The **Black Wire** of the twisted pair is soldered to the **MOSFET Source Pin (Pin 3)**, NOT the main Star Ground stud.
- [ ] **The "Tug Test":** Gently pull on the twisted pairs. The connection to Pin 1 (Gate) is solid and does not move.
- [ ] **Safety Gap:** The twisted pairs are routed *away* from the +BUS rail. They are not resting on the high-voltage copper.

Verification Signature: ___________________________ *(Do not sign until all boxes are checked. You are now ready for Section 13: Commissioning.)*

This concludes Chapter 9.

You have successfully documented the construction of the most complex part of the generator.

1. **Section 9.2:** The Driver Board (Brains).
2. **Section 9.3:** Mechanical Assembly (Skeleton).
3. **Section 9.4:** Protection (Muscle).
4. **Section 9.5:** Integration (Nerves).
5. **Section 9.8:** Final Validation (Checklist).

10) Harvest wiring (start with Column D)

- Set each D-coil **series-aiding** at its shelf.
- Free ends **{Coil}A_S (e.g., A1A_S)** and **B-F** → bridge "~" "~".
- At the bridge DC side: **0.1 μF** across **+** ↔ **−** (short leads), then **1000 μF / 35–63 V** across **+** ↔ **−**.
- DC+ → 15–30 A blade fuse → Ideal-diode IN+ → Ideal-diode OUT+ → +BUS.
- DC− → Star-Ground bus.
- Add more harvest columns later the same way.

10.1 Harvest Bridge Bank Assembly (The "Power Plant")

Objective: Install and bus the **9 Bridge Rectifiers (KBPC3510)**. These convert the raw AC from your harvest coils into the DC power that charges the system.

The "Uniformity" Rule:

- **Orientation:** All 9 bridges must be mounted in the **exact same orientation**.
- **The Key:** Look for the **Slanted Corner** (or the tab marked **+**). This corner must always face **Top-Right**.
- **Safety:** If even *one* bridge is rotated incorrectly, it will create a dead short across the main bus when you connect the rails.

1. Mechanical Mounting (3×3 Grid)

Use **Figure 10-1a** to mount the bridges to the Bottom-Left of the Backplate.

- **Count:** Install exactly **9 Bridges**. (One for each of the 9 Harvest columns).
- **No Spares:** Do not install a 10th bridge. There are no spares in this bank.
- **Spacing:** Leave at least **10mm** between cases to allow for airflow and wire routing.
- **Thermal Paste:** Apply a thin layer under each metal case before bolting down.

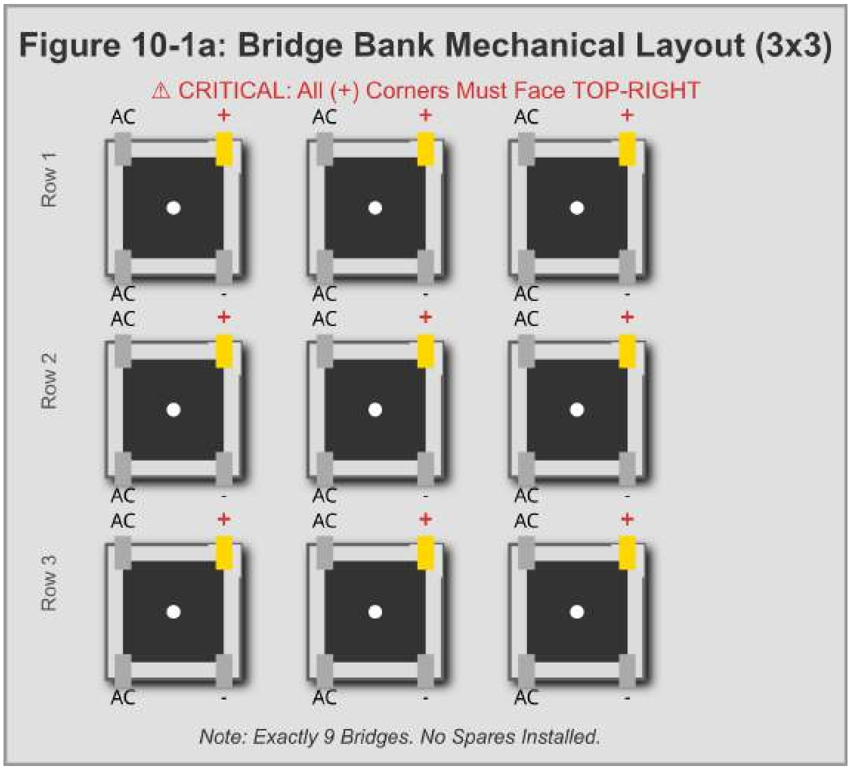


Figure 10-1a: Bridge Bank Mechanical Layout. Mount 9 units in a 3×3 grid. Ensure the (+) corner faces Top-Right. (Note: If your reference image shows a 10th "ghost" bridge, ignore it. Install only 9.)

---

2. DC Bussing (Wiring View)

We will create **Solid Copper Rails** to link the DC outputs. Use **Figure 10-1b** as your map.

- **Material Rule:** The DC+ and DC- rails must each be implemented using a single consistent method: either **Solid Copper Bar** or continuous **14 AWG (or thicker) wire** with insulation stripped at contact points. Do not mix methods within the same rail.
- **The DC+ Rail (Red):** Connects all 9 **(+)** tabs together. Output goes to the **Harvest Fuse (30A)**.
- **The DC- Rail (Black):** Connects all 9 **(-)** tabs together. Output goes to **Star Ground**.
- **The AC Tabs:** Leave these individual. **Do NOT bus the AC tabs.** They connect to specific coils.

Figure 10-1b: DC Bus Wiring (The Rails)

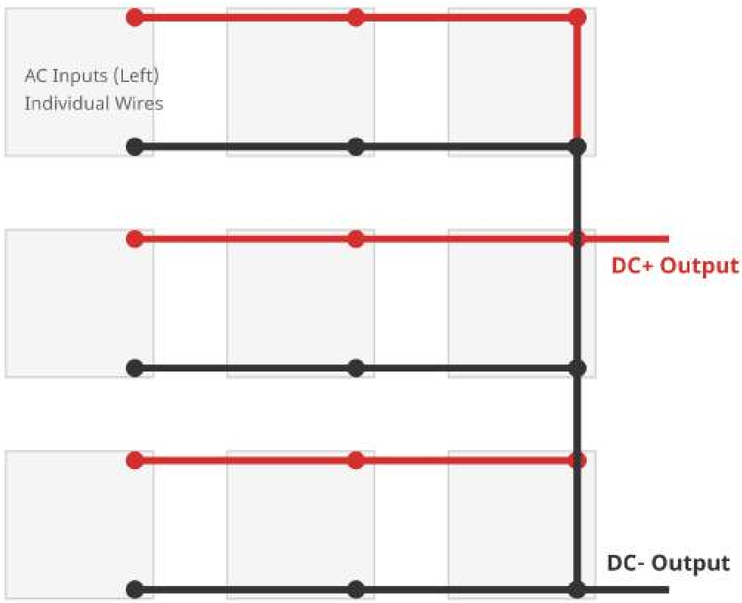


Figure 10-1b: DC Bus Wiring. Create solid rails connecting all Positives together and all Negatives together. AC inputs must remain individual.

3. Assembly Netlist & Mapping

Table 10-1: Bridge Bank Wiring Guide

ID	Action	Source	Destination	Critical Note
#1	**Orient & Mount**	**9x KBPC3510**	Backplate (3×3)	**Chamfered Corner (+)** must be **Top-Right** on ALL units.
#2	**Create DC+ Rail**	**Red Bus**	All 9 **(+)** Tabs	Chain them together. Output to **Harvest Fuse**.
#3	**Create DC-**	**Black Bus**	All 9 **(-)** Tabs	Chain them together.

	Rail			Output to **Star Ground**.
#4	**Connect AC**	Coil Wires	**(~) Tabs** (Left Side)	**Mapping:** Top-Left Bridge = Column C2. Proceed left-to-right, top-to-bottom.

Column Mapping Note: For this build, the 9 Harvest Bridges correspond to the 9 Harvest Columns (typically C2 through C10). Verify your specific coil layout, but ensure one bridge serves exactly one column.

10.2 Power Distribution & Regulation (Ideal Diodes & Bucks)

Objective: Install the "Traffic Cops" (Ideal Diodes) that merge power sources without backflow, and the "Transformers" (Buck Converters) that create stable 12V and 5V rails.

Location: Refer to **Figure 20-1**.

- **Ideal Diode Bank:** Top-Right quadrant of the backplate (2×2 grid).
- **Buck Cluster:** Directly below the Ideal Diodes (2×2 grid).

1. Mechanical Mounting (Top View)

Use **Figure 10-2a** to mount the modules.

- **Orientation:** Mount all modules so their **Input Terminals** face **Left** (towards the bridges) and **Output Terminals** face **Right** (towards the Bus/Door).
- **Standoffs:** Use **M3 Nylon Standoffs** to lift these boards off the aluminum. **Do not bolt them directly to the metal.** The back of these PCBs conducts electricity and will short out if it touches the backplate!

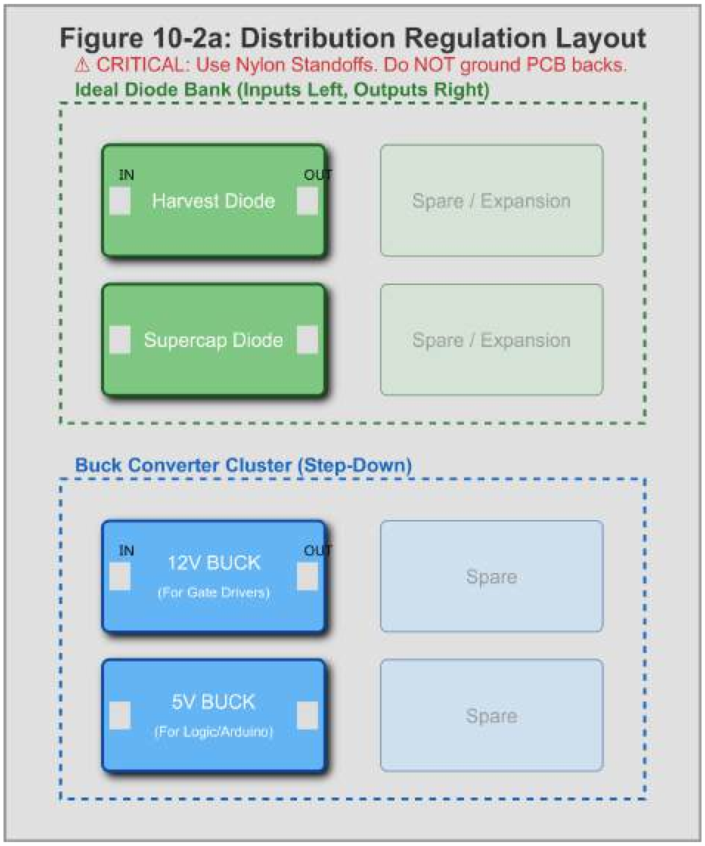


Figure 10-2a: Distribution & Regulation Layout. Inputs on the Left, Outputs on the Right. Use Nylon Standoffs.

---

2. Wiring Logic (The "Merger")

Use **Figure 10-2b** to wire the power flow.

1. **Harvest Path:**
    - **Source:** The **DC+ Rail** from your Bridge Bank (Section 10.1).
    - **Destination:** Input of **Ideal Diode #1**.
2. **Supercap Path:**
    - **Source:** The Positive (+) Terminal of your Supercapacitor Bank.
    - **Destination:** Input of **Ideal Diode #2**.
3. **The Main Bus (+BUS):**
    - Connect the **OUTPUTS** of Diode #1 and Diode #2 together.
    - This combined output becomes your main system **+24V BUS**.
4. **Buck Inputs:**
    - Connect the Inputs of both Buck Converters to the **+BUS**.
5. **Buck Outputs:**
    - **Buck #1:** Tune to **12.0V**. This feeds your **Gate Drivers** (TC4427).
    - **Buck #2:** Tune to **5.1V**. This feeds your **Arduino/Logic**.

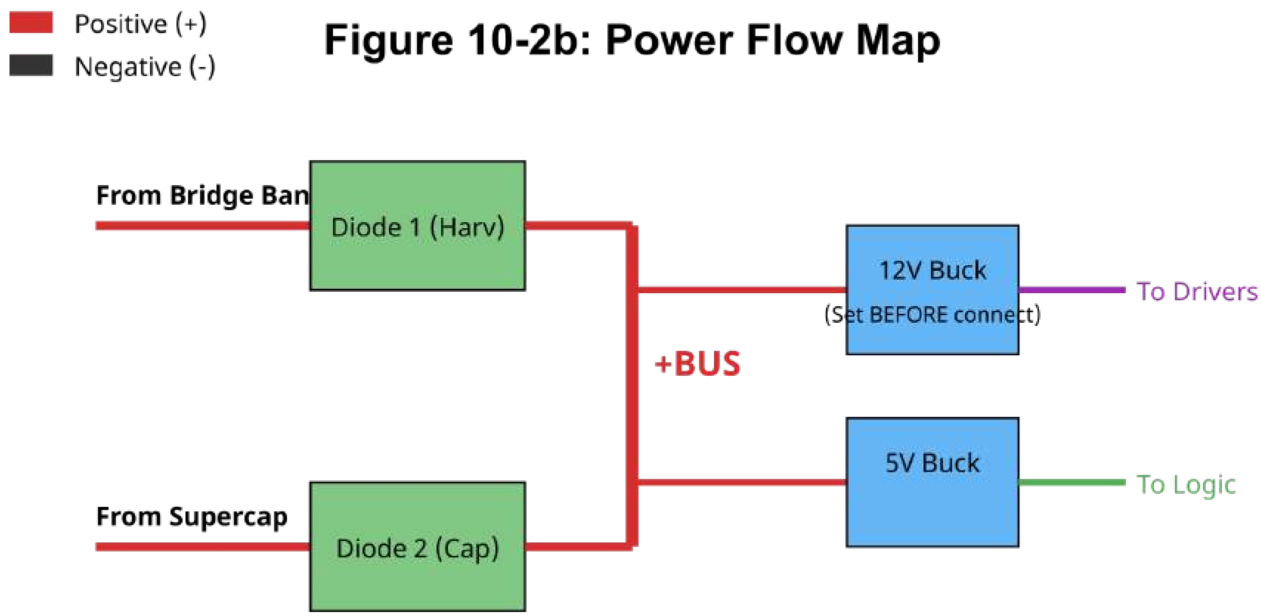


Figure 10-2b: Power Flow Map. Note how the "Ideal Diodes" merge the Harvest and Supercap into a single +BUS.

3. Critical Tuning Procedure

STOP. Do not connect the Buck Outputs yet.

1. Apply 12V-24V to the **Input** of the Buck Converters (using a bench supply).
2. Use a multimeter to measure the **Output Voltage**.
3. Turn the tiny brass screw on the blue potentiometer until:
    - **Buck 1 reads exactly 12.0V.**
    - **Buck 2 reads exactly 5.1V.**
4. **Only after tuning** can you connect the outputs to your Gate Drivers or Arduino.
    - *Why? Factory settings might be 24V or higher, which will destroy your Arduino instantly.*

11) Meters (you own two Bayite Wh meters)

- **Meter #1 (External Input):** Put the shunt in the external supply feed only—either in source positive before it joins +BUS, or in source negative before it reaches Star-Ground. Do not place it in the general Star-Ground bus (that would also count internal recirculation). This meter shows watt-hours imported from the bench supply.

- **Meter #2 (Harvest):** For typical Bayite/DROK meters (non-isolated, low-side shunt), install the shunt on the NEGATIVE return: Bridge Rectifier (−) → Shunt → Star-Ground. Connect the meter's thin voltage-sense lead to Harvest DC+ (after the bridge, before the ideal-diode).

- WARNING: Do NOT place this meter's shunt in Harvest DC+ unless the meter supply is isolated and explicitly supports high-side shunting.

- **Note on energy accounting:** Meter #2 measures gross harvest (energy leaving the D-bridge). For net output, meter the external load and/or the harvest contribution into +BUS. If you add a third meter, place its shunt on a low-side return (for example, load negative) to avoid high-side shunt hazards with common-ground meters.

- **Two-meter option:** If your goal is input-vs-output energy balance, place Meter #2 on the external load feed (+BUS to the load/inverter) so it reads net energy delivered to the load, while Meter #1 remains on the external supply feed.

12) Firmware timing — how the pulses are scheduled

Pitch = 32.727°. At **3000 RPM**, **1 rev = 20 ms ⇒ 55.56 µs per degree**. Use Hall-A and Hall-F edges as timing references (note: a Hall edge is a threshold crossing, not automatically the geometric center).

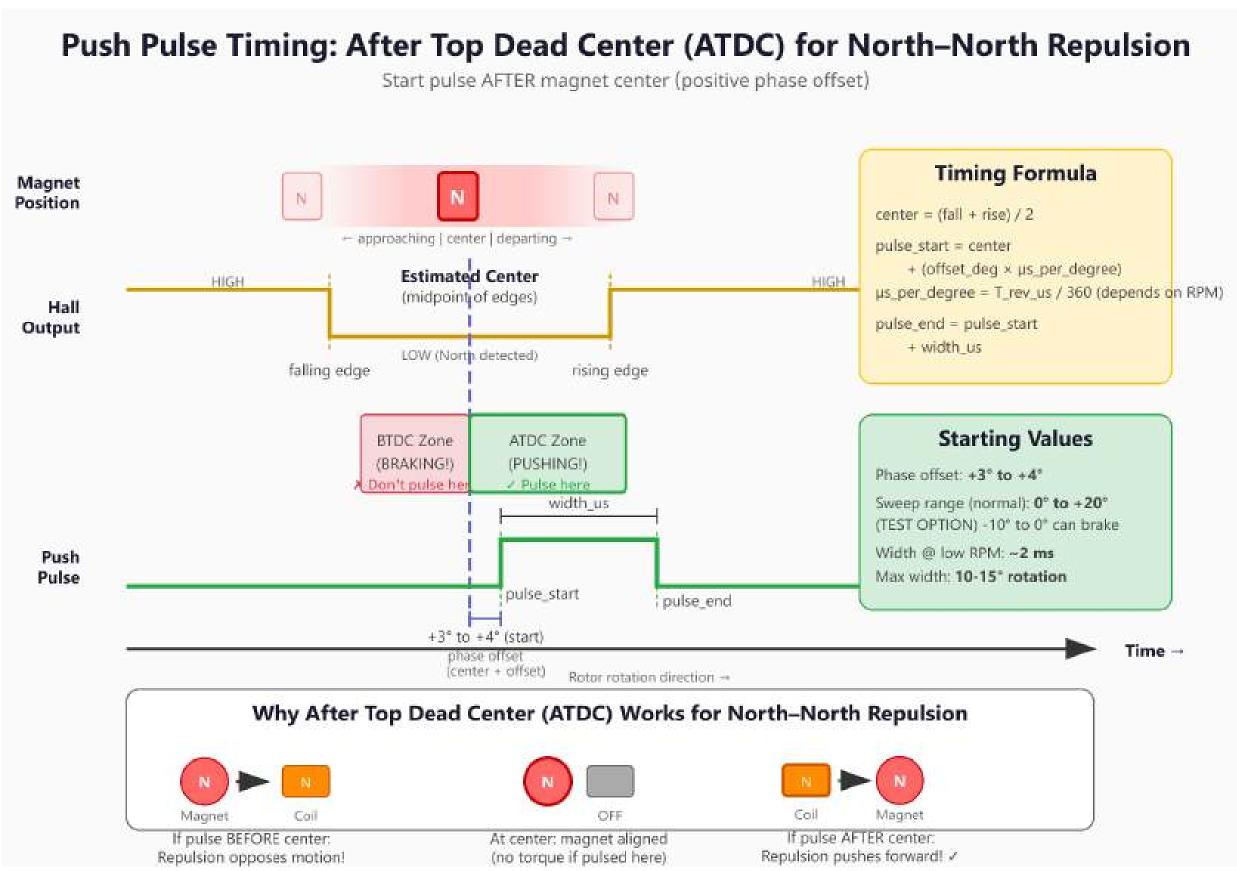


Figure 12-1 — Push Pulse Timing

This diagram shows **how to decide exactly when to turn a coil ON** so it gives the rotor a “push” instead of slowing it down.

A) What the Hall sensor signal means

The Hall sensor output is a digital signal (HIGH or LOW).

- It stays **HIGH** most of the time.
- It goes **LOW** when the rotor magnet is close enough that the sensor detects **North**.

That LOW section has two important moments:

- **falling edge**: the signal changes HIGH → LOW (North detection begins)
- **rising edge**: the signal changes LOW → HIGH (North detection ends)

B) Why we estimate the magnet center

The Hall sensor does **not** tell you the exact center of the magnet directly.
It switches when the magnetic field crosses the sensor’s internal threshold.

So we estimate the “center time” like this:

estimated_center_time = (falling_edge_time + rising_edge_time) / 2

That is why the dashed line is labeled **Estimated Center (midpoint of edges)**.

C) Why pulsing BEFORE center can brake

Think of the magnet moving past the coil:

- **Before center (BTDC)**: the force from the energized coil tends to oppose the motion → **braking risk**
- **After center (ATDC)**: the energized coil’s force tends to help the magnet move away → **pushing**

That is why the diagram marks:

- **BTDC zone: “Don’t pulse here”**
- **ATDC zone: “Pulse here”**

D) What “phase offset” means (and why +3° to +4° is only a starting value)

Phase offset means how far after the estimated center you start the pulse.

The build guide currently says to start with:

- **phase_offset_deg = +3° to +4°**

That is not "the correct value forever." It is a **safe starting guess** that should usually land the pulse on the "pushing" side.

Then you **sweep timing** to find the best point for your build:

- normal sweep: **0° to +20°**
- **(TEST OPTION)**: **−10° to 0°** can brake

E) Converting degrees into time (this is the part students often miss)

A degree is a slice of one full rotation.

To convert degrees into microseconds, you need the rotor speed:

- **μs_per_degree = T_rev_us / 360**
- where **T_rev_us** is the time for one full revolution in microseconds

So the same "+4 degrees" means **different microseconds** at different speeds.

F Pulse width (how long the coil stays ON)

The diagram also shows pulse width:

- **pulse_end = pulse_start + width_us**
- The build guide rule is to keep width ≤ **10° to 15° of rotation**, so the push is short and does not turn into heating or braking.

For each column:

- Capture both Hall edges for that magnet pass if available (falling_edge and rising_edge).
- Estimate magnet_center_time = (falling_edge + rising_edge) / 2. If you only capture one edge, treat it as edge_time and apply a tuned offset.
- Compute pulse_start = magnet_center_time + (phase_offset_deg × μs_per_deg). For North-North repulsion, start with phase_offset_deg = **+3 to +4 degrees** (start slightly **AFTER** your estimated center), then sweep from −10° to +20° to find the phase that produces forward acceleration with the lowest current. (If you only captured one Hall edge,

magnet_center_time is approximate—expect a larger sweep).

- Compute pulse_end = pulse_start + width_us. Start width_us at 1.2–2.0 ms at low RPM; shrink width_us as RPM rises (keep the pulse ≤10–15° of rotation).
- Assert column + row HIGH at pulse_start, LOW at pulse_end.

Interleave A and F; if the same row would be hit at the same moment, skip one pulse.

Pulse width scaling: Keep push pulse width ≤10–15° of rotation. Convert to time:

- **300 RPM:** 556 µs/deg → 8.3 ms max
- **1000 RPM:** 167 µs/deg → 2.5 ms max
- **3000 RPM:** 55.6 µs/deg → 0.83 ms max

Start with ~2 ms at low RPM and reduce as speed increases.
Safety: Minimum OFF-time per rev, overspeed cutoff (e.g., >3500 RPM for 0.1 s ⇒ disable pulses 5 s).

12.1 Row selection strategy: The switching is intersection-based: a coil fires only when both its column high-side P-FET and its row low-side N-FET are ON. We enforce one-hot selection—only one physical coil is pulsed at a time.

- **Bring-up:** Start with a single coil (e.g., Column A, Row 3).
- After polarity and timing are verified, test additional rows or cycle through rows in firmware. Do not enable multiple rows simultaneously on the same column during initial commissioning.

13) Commissioning (The "First Breath")

Objective: Safely power up the machine for the first time.

Golden Rule: Do not rush. If a test fails, STOP. Do not "try it again" without finding the root cause.

A) Unpowered Checks (Multimeter Only)

- [ ] **The Final Beep Test:**
    - Set Multimeter to **Continuity Mode**.
    - Probe 1: **Aluminum Heatsink**.
    - Probe 2: **Copper +BUS Rail**.

    - **Result:** MUST be Silence (Open Circuit).
    - *Why:* If this beeps, your "Dead Bug" rail is touching the heatsink. Do not power on.
- [ ] **Short Circuit Check:**
    - Probe 1: **+BUS**.
    - Probe 2: **Star Ground**.
    - **Result:** Silence (or brief beep while capacitors charge). Resistance should climb to >1kΩ.
- [ ] **Coil Isolation:** Verify no shorts between any Coil Positive (+A1...K5) and the Frame/Heatsink.

B) 5V Logic Check (No High Voltage Yet)

- [ ] **Power Up Arduino:** Plug in USB. LCD should light up.
- [ ] **Check Hall Sensors:**
    - Rotate rotor by hand.
    - Verify the **LCD RPM** counter changes or the debug LED blinks.
    - Ensure clean HIGH $\to$ LOW $\to$ HIGH transitions. No "flickering" or double-counts.

C) Low-Voltage Power Injection (The "Soft Start")

Setup: Set your Bench Power Supply to **12V** and **Current Limit 2A**. Connect to +BUS and Star Ground.

- [ ] **Quiescent Current Check:**
    - With NO coils firing (Idle), the current draw should be very low (< 200mA).
    - *If it draws >1A instantly:* You have a short. Kill power immediately.
- [ ] **Manual Fire Test (One Coil Only):**
    - Use the "Test Mode" in your firmware to fire **Column A + Row 3** for **50 milliseconds**.
    - **Result:** The rotor should "kick" forward.
    - **Current:** The Bench Supply should show a momentary pulse, then return to zero.

D) Scope & Protect (The Waveform Check)

- [ ] **Probe Location:** Connect Oscilloscope Probe to the **Row 3 Switching Node** (The N-FET Drain / Center Pin).
    - *Note:* Do not probe the Column. The action is at the Row Drain.
- [ ] **Ground Clip:** Connect Scope Ground to **Star Ground**.
- [ ] **The Pulse:** Spin the rotor by hand or low speed. Watch the pulse.
- [ ] **The Flyback:** Look at the moment the switch turns OFF.
    - **Good:** The voltage spikes up but is "clamped" flat by the TVS/Zener.
    - **Bad:** You see crazy "ringing" (oscillations) that go higher than 60V.
    - *Fix:* If ringing > 60V, check your **Twisted Pair** routing and ensure the **MUR460** diode leads are short (<10mm).

E) Spin Up (Gradual Acceleration)

- [ ] **Enable Column A Only:** Start with one push column.
- [ ] **Monitor Heat:** Touch the **Heatsink (HS3)** every 30 seconds. It should be cool or warm, never hot to the touch.
- [ ] **Add Column F:** Once stable, enable the second push column.
- [ ] **Harvest Check:** Only enable harvest (Column D) once the machine acts predictably.

14) Troubleshooting (symptom → action)

- **No push at all:** The per-coil polarity is flipped—repeat the quick magnet test and swap which node is Column-Positive.

- **Hall flickers / double counts:** Increase the Arduino pin capacitor to **22–47 nF**; add the **single-point shield bond**; route Hall cable away from the FET/bridge loop.

- **MOSFET warm:** Confirm **TC4427** is feeding the row MOSFET; check **gate resistors 22–68 Ω**; ensure **zener** and **10 k gate pull-up** are installed on P-FETs; verify snubber.

- **Harvest reads zero:** Check bridge tabs (~, ~, +, −) wiring and the fuse/ideal-diode orientation.

- **Cold-start flaky:** Add a stronger hand-flick; confirm NTC, LVD thresholds, and the **5 V hold-up cap** near the Arduino.

15) Why short pulses work (physics in one paragraph)

A quick current pulse makes the selected stator coil a North pole. That repels the rotor's North magnet. For North-North repulsion, the useful "kick" is AFTER the magnet passes center (ATDC): start the pulse slightly after the estimated center and end shortly after. If you pulse before center, the repulsion tends to oppose the approaching magnet (braking).

Because Hall edges are threshold crossings (not guaranteed geometric center), the correct phase must be found by a short phase sweep and confirmed by forward acceleration and reasonable current. As you harvest more watts, Lenz's law increases braking, so it is normal to tune timing with harvest OFF first, then measure with harvest ON.

16) Minimal diagrams

16.1 Push path

+BUS → IRF4905 (A) → DIN A+ → [Coil (series-aiding)] → Row 3 trunk → IRLZ44N (Row 3) → Star-Ground

16.2 Gate networks (text)

- **P‑FET (IRF4905):** Gate→+BUS (47 kΩ), gate stopper 22–47 Ω at MOSFET, 12 V Zener Gate→Source (cathode at Source), plus 1–2.2 kΩ series current‑limit between Gate node and ULN2803A sink. ULN2803A sinks Gate to turn ON.

- **N‑FET (IRLZ44N):** Gate from TC4427 through 22–68 Ω, 10–100 kΩ Gate→Source.

16.3 Protection

- (Drain) node ──► MUR460 ► +BUS (stripe at +BUS)

- Switch Node (Drain) ──► MUR460 ► +BUS (stripe at +BUS)

- Row node ──► [100 Ω in series with 1 nF (≥1 kV)] ► +BUS (snubber if needed)

16.4 Cold‑start

D-column AC → KBPC3510 → NTC 5D-11 → 16 V/83 F supercap → Ideal-diode → LVD → +BUS └──► (separate low-voltage tap, VIN≤5.5 V) BQ25570 → 5 V hold-up cap → Arduino/Halls

17) Parts you already own that this plan uses

- **Switching:** (2) IRF4905 (P-FETs), (5) IRLZ44N (N-FETs), (≥1) TC4427, (≥1) ULN2803A.

- **Protection:** 1.5KE33A TVS, MUR460 diodes, 100 Ω/2 W + 1 nF/1 kV for snubbers.

- **Harvest:** KBPC3510 bridges, ideal‑diode module(s), LVD module, NTC 5D‑11.

- **Power:** BlueStars bus bars, ANL fuse + holder, DROK 5 V buck (and optional 12 V buck).

- **Cold‑start storage:** 16 V / 83 F supercap module.

- **Control & IO:** Arduino Mega kit, LCD1602, Hall A3144 sensors, cabling, ferrites, WAGO 221‑413, DIN terminals & rails, labels.

- **Meters:** (2) Bayite Wh meters + 100 A shunts.

- *Note:* If a **5 V supercap** isn't on hand, use **1000–2200 µF** electrolytic at the Mega as the hold‑up; you can add a true supercap later.

18) Student checklists (print these)

SVG Wiring Diagram Audit Checklist (v3.8.4)

Check every box before students build. Goal: no missing parts, no ambiguous paths, no clipped drawings.

A) Inventory checks

- □ Coils labeled A1…K5 exist in the project (Column A–K, Row 1–5); commissioning sheet may show only a subset
- □ Each coil has 4 leads labeled like: A1A_S, A1A_F, A1B_S, A1B_F (replace "A1" with the coil ID)
- □ Internal WAGO link is explicitly stated: A_F ↔ B_S (inside each module)
- □ Only the A_S lead lands on DIN labeled +A1…+K5. Returns (B_F) do NOT land on DIN
- □ Two DIN rails of 35 terminals each: 70 total, 55 used, 15 spare (per locked constraints)
- □ Shunts installed for: push return, inverter return, and harvest output
- □ Fuses installed for: push, inverter, harvest, 5 V logic, and 12 V drivers
- □ Commissioning mode uses ONE push column only (start with Column A). All other columns disabled for push

B) Continuity checks (trace end-to-end)

- □ Push path: +24 V bus → push fuse → P-FET (push column) → DIN +A# → coil → row MOSFET → push shunt → − bus
- □ Row MOSFET sources are explicitly wired to a common return that feeds the push shunt (no hidden connection)
- □ Harvest path: column → bridge rectifier → harvest fuse → +HARVEST bus → harvest shunt → ideal-diode → buck → +24 V bus (bridge negative → − bus)
- □ Inverter path: +24 V bus → inverter fuse → inverter; inverter return goes through inverter shunt to − bus
- □ Cold-start: generator tap → inrush limiter (NTC) → supercap → ideal-diode → low-voltage disconnect → +12 V bus (correct direction)
- □ Hall signals: HALL_A and HALL_F are named; 1 kΩ series + 10 nF to − bus are shown at the microcontroller side

C) Visual clarity checks

- □ No label box covers a junction dot, symbol, or pin label
- □ All junction dots are visible and not hidden by boxes

If all boxes are checked, the diagram set is approved for commissioning documentation.

Figure 18-1 — System Audit Checklist (Labeling, Polarity, and Safety Invariants)

Coil shelf (one coil)

- [ ] Four pigtails labeled {Coil}A_S, {Coil}B_S (red), {Coil}A_F, {Coil}B_F (black). Example: A1A_S, A1B_S (red), A1A_F, A1B_F (black).
- [ ] Two WAGOs built as series-aiding (REQUIRED for both push & harvest).
- [ ] Column-Positive and Row-Negative pigtails labeled and strain-relieved.
- [ ] North test shows **repel** when Column-Positive is powered.

Halls

- [ ] A at column A, F at column F (5 pitches = ~163.6°).
- [ ] A3144 back toward magnets; flat face away; gap 0.8–1.5 mm.
- [ ] 0.1 μF at splice; ferrite in panel; RC at Arduino pins D2/D3.
- [ ] Shield unbonded at first; pigtail labeled.

Power & meters

- [ ] ANL fuse installed (cover on).
- [ ] Meter #1 shunt in external supply feed (+ or − before bus); Meter #2 shunt in Bridge Negative (low-side) with voltage sense to Harvest DC+.

Push path

- [ ] P-FET gate network: 47 kΩ pull-up Gate→+BUS, 22–47 Ω gate stopper at MOSFET, 12 V Zener Gate→Source, plus 1–2.2 kΩ series current-limit between Gate node and ULN2803A sink.
- [ ] ULN2803A sinks gate; +BUS ≤ 50 V (or use 2N2222 sink).
- [ ] Row N-FETs via TC4427; gate resistors 22–68 Ω; pull-downs in place.
- [ ] MUR460 + TVS close to nodes; add snubber if scope shows ringing.

Run

- [ ] Pulse starts about 3-4 degrees AFTER your calibrated center reference (positive phase offset for North-North repulsion), then tune by phase sweep; revolutions per minute rises smoothly.
- [ ] Pulse end near magnet center (start ~3–4° before your calibrated center reference, then tune by phase sweep); RPM rises smoothly.
- [ ] Harvest ON shows DC and expected Lenz drag.
- [ ] Jitter ≤ 0.5% with pulses ON (or add single-point shield bond).

19) FAQ (for students)

- **Why two Halls (A & F)?** 11 columns have no exact opposite. A→F is 5 pitches (~163.6°), giving two well-spaced timing marks.
- **Why stop the pulse near center?** If you drive too far past the best-torque point, the field can start opposing motion. We give a short push, then let it coast. The exact phase depends on how your Hall reference maps to magnet center, so tune by a small phase sweep.
- **Why don’t we leave coils ON longer?** Longer pulses heat the coil and waste energy; short pulses usually give better torque per energy.

- **Why measure with harvest OFF first?** Harvest adds magnetic drag (Lenz effect). First verify push timing and stability, then enable harvest and account for the extra load.

- **Is series or parallel better for push?** With our 18 AWG coils, **Series-Aiding** is mandatory to protect the MOSFETs. Parallel draws too much current (>180 A).

20) Where everything lives on the backplate (325×275 mm)

20.0 Enclosure Orientation & Wall Naming

The "Rotated Map" Rule: To ensure alignment with the diagrams, we orient the enclosure so the **Hinge is at the Top**. Use **Figure 20-0** to orient yourself.

- **Wall 1 (Top / Hinge):** The Pivot Point.
  - *Function:* Hinge mechanism; anchor for the door.
- **Wall 2 (Right):** Power Distribution.
  - *Function:* **Bus Bars** (+24V/GND) and **Inverter Cable Entry**.
- **Wall 3 (Bottom):** Input Zone.
  - *Function:* **Cable Glands** (Coils, Halls, Harvest wiring enter here).
- **Wall 4 (Left):** The User Interface.
  - *Function:* **Meter**, **E-Stop**, and **Switches** mount on this wall.
- **Center (Bottom):** The Aluminum Back Panel.
  - *Function:* The main mounting surface for electronics.

Figure 20-0: Wall Numbering Map

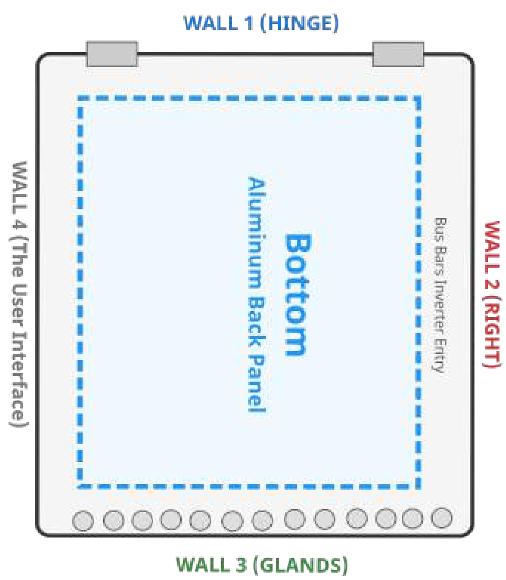


Figure 20-0: Enclosure Wall Map. Wall 4 (Bottom Wall) holds the User Interface.

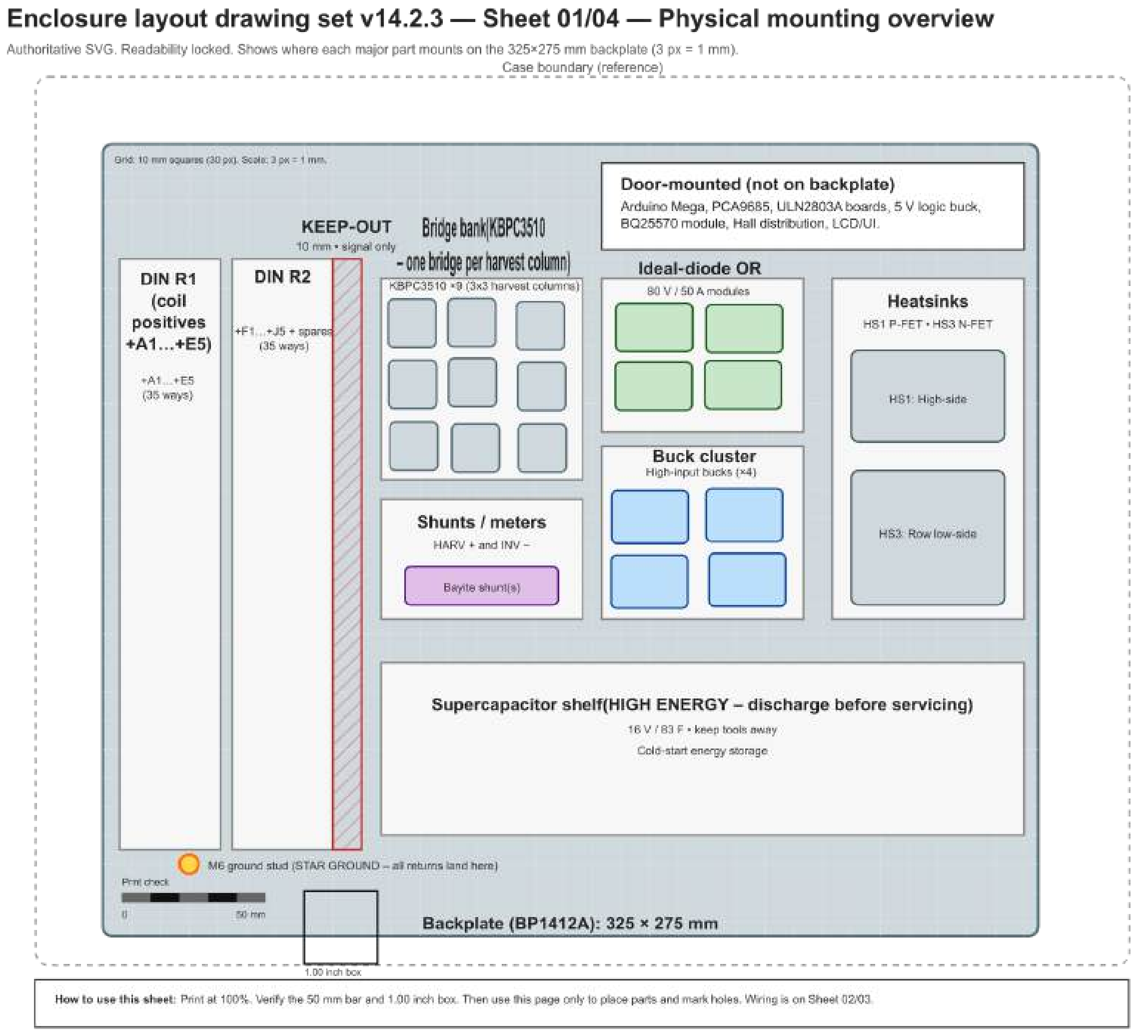


Figure 20-1. Enclosure Master Layout — Physical Mounting Overview (Sheet 01)

This sheet answers one question only: **“Where does each major part physically mount inside the enclosure?”** It does not show detailed wiring. Its purpose is to prevent drilling mistakes and overcrowding before any wires are installed.

What This Drawing Represents: The blue rectangle is the metal backplate mounted inside the enclosure. The size shown (325 × 275 mm) matches the real backplate. The grid is a 10 mm reference grid to help with spacing and alignment. The dashed outline shows the enclosure boundary for reference only. This drawing is printed at 100% scale and must be verified before drilling.

Scale Check (Do This First)

Before drilling anything, measure the 50 mm scale bar and the 1 inch square. If either measurement is incorrect, do not drill and reprint the page. This prevents permanent mounting errors.

DIN Rails (Left Side)

DIN R1 and DIN R2 are installed first. These rails are where all coil positive leads (+A1…+K5) terminate. Each rail has enough terminals for the coils plus spares. Landing coil wires on DIN rails keeps wiring organized and allows future troubleshooting.

Red Hatched Strip — Keep-Out Zone

The red striped area is a 10 mm keep-out strip. Do not run thick power wires through this area. Only thin signal or logic wires may cross it. This strip separates noisy, high-current wiring from sensitive electronics and reduces electrical interference.

Bridge Rectifier Bank

Each square represents a KBPC3510 bridge rectifier. Each harvest column uses one bridge. Bridges convert coil alternating current (AC) into direct current (DC). Mount all bridges first and verify spacing before tightening.

Ideal-Diode OR Bank

These modules prevent current from flowing backward between sources. They protect the system when multiple harvest paths are combined. Current can flow toward the bus but cannot return to another source.

Buck Converter Cluster

These high-input DC-DC converters step harvested voltage down to controlled levels. Grouping them together keeps wiring short and improves electrical stability.

Shunts and Meters

Shunts measure current flow for harvest and bus/inverter paths. Their placement is fixed. Do not swap shunt locations, as measurements will be incorrect.

Heatsinks

Heatsinks mount the power MOSFETs. Separate heatsinks are used for high-side switches and row (low-side) switches. MOSFETs must be electrically insulated from the heatsink using the supplied insulator kits.

Supercapacitor Shelf

This area holds the cold-start energy storage capacitor rated at 16 V / 83 F. It is labeled HIGH ENERGY. Always discharge the supercapacitor before servicing and keep metal tools away from its terminals.

Door-Mounted Components

Anything labeled “door-mounted” does not mount on the backplate. These include the microcontroller, driver boards, logic power supplies, Hall sensor distribution, and user interface components. Keeping logic on the door prevents overcrowding and improves service safety.

Ground Stud

The orange marker shows the star ground location. All system returns connect here. Do not daisy-chain grounds. Every return must land at this single point.

How to Use This Sheet During Assembly

Print at 100% scale and verify measurements. Place the page on the backplate and mark hole locations. Drill and mount hardware. Only after all parts fit correctly should wiring begin. If anything does not physically match this drawing, stop and fix it before proceeding.

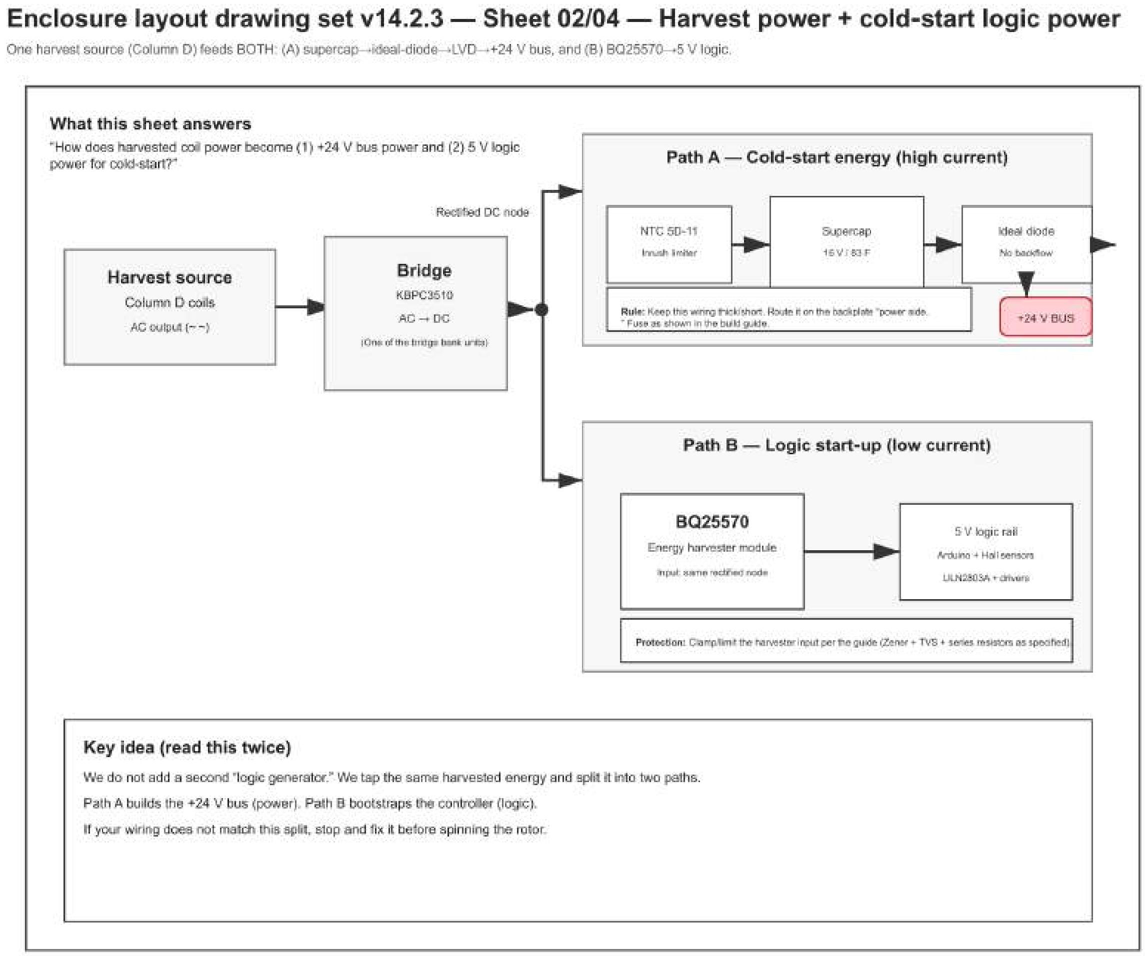


Figure 20-2. Harvest Power and Cold-Start Logic Power (Sheet 02)

This sheet explains how power flows through the system, not where parts physically mount; use Sheet 01 for physical placement.

Purpose: This page answers one question only: how harvested coil power becomes (1) main bus power and (2) logic power.

Diagram type: This is a functional block diagram, not a backplate or drilling layout.

Power flow direction: Power starts at the generator coils and flows left to right through the system.

Harvest source: Column D coils generate alternating current (AC).

Bridge rectifier: AC from Column D enters a KBPC3510 bridge, which converts it to direct current (DC).

Rectified node: After the bridge there is a single DC node; all downstream power comes from this same point.

Critical split: From the rectified DC node, power intentionally splits into two paths; no separate logic generator is used.

Path A (cold-start energy): This path handles higher current and builds the main power bus.

Path A order: Bridge → NTC 5D-11 → Supercapacitor → Ideal Diode → Low-Voltage Disconnect → +24 V bus.

NTC function: The NTC limits inrush current when the supercapacitor is empty.

Supercapacitor function: The supercapacitor stores energy for cold-start operation.

Ideal diode function: The ideal diode prevents energy from flowing backward into the system.

LVD function: The low-voltage disconnect protects the bus from deep discharge.

Path A result: The output of Path A is a stable +24 V bus.

Path A wiring rule: This wiring must be short, thick, and routed on the backplate power side.

Path B (logic start-up): This path provides low-current power for control electronics.

Path B order: Same bridge node → BQ25570 energy harvester → regulated 5 V logic rail.

BQ25570 role: The BQ25570 is designed for ultra-low-power start-up.

Logic powered: The 5 V rail powers the microcontroller, Hall sensors, and logic drivers.

Startup advantage: This allows the controller to wake up before the main power bus is active.

Input protection: Series resistors, Zener diodes, and TVS devices limit voltage and protect the harvester.

Key idea: We do not add a second generator for logic power.

System concept: One harvested source is safely split into two functional paths.

Functional summary: Path A builds power; Path B boots control.

Student warning: If wiring does not clearly match this split, stop and fix it before spinning the rotor.

Physical placement warning: Blocks on this sheet do not represent physical positions in the enclosure.

Authoritative placement: Physical mounting is defined by Sheet 01 (overall layout) and Sheet 03 (switching and thermal).

Example placements: The bridge lives in the bridge bank; the BQ25570 is door-mounted; the supercapacitor mounts on the bottom shelf.

Final rule: Never drill or mount parts using Sheet 02 alone.

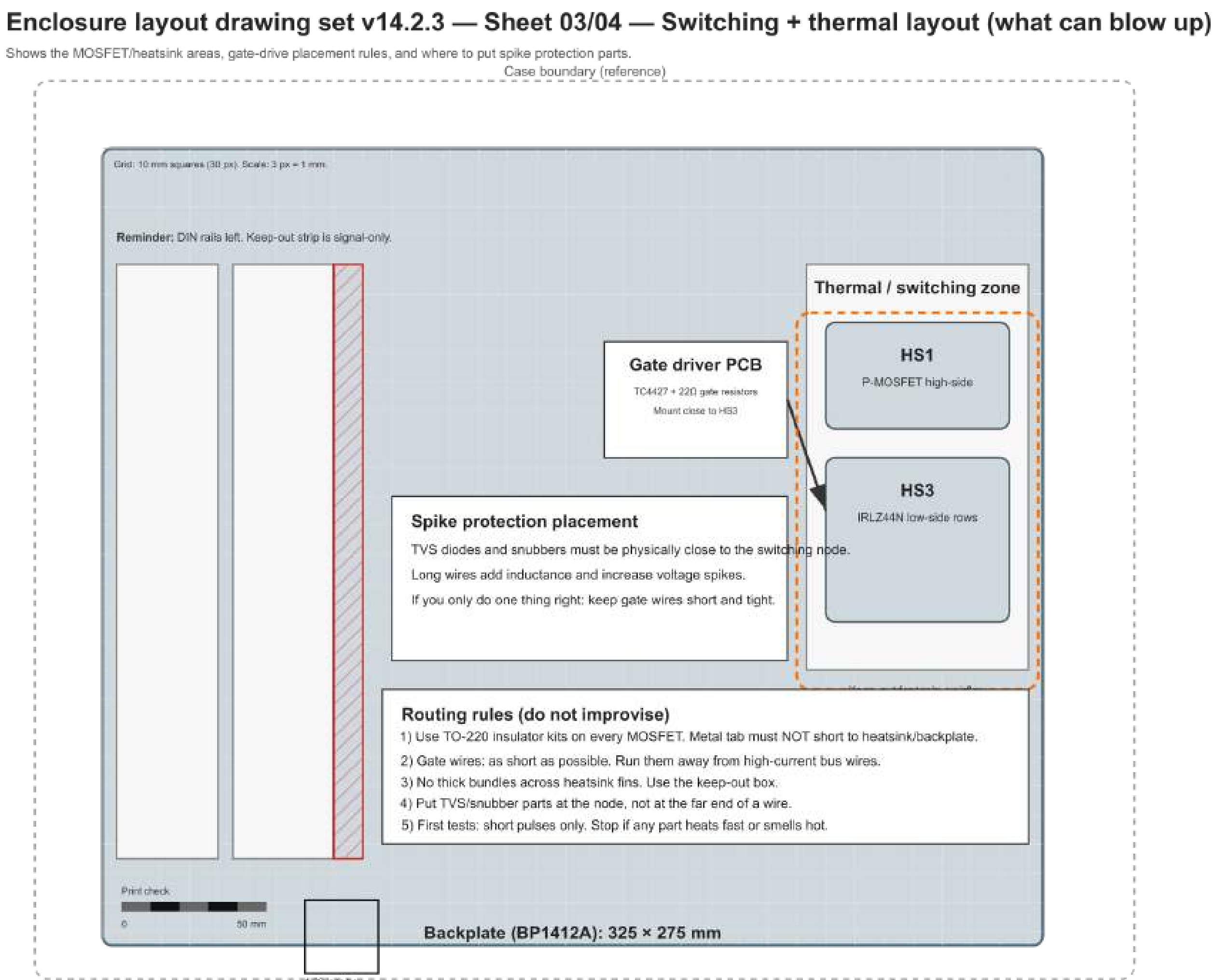


Figure 20-3. Switching and Thermal Layout (Sheet 03)

1. This page shows where the high-power switching parts go and how to prevent

overheating or electrical failure.

2. **Purpose:** This sheet answers "Where do the MOSFETs, heatsinks, and gate drivers mount, and how must they be wired safely?"
3. **Thermal / switching zone:** The dashed zone contains all high-power switches and heatsinks; do not place unrelated parts here.
4. **Heatsinks:** HS1 holds high-side P-MOSFETs; HS3 holds low-side row N-MOSFETs; both switch high current and can become hot.
5. **Insulation:** Every MOSFET must use a TO-220 insulator kit; the metal tab must not contact the heatsink or backplate.
6. **Gate driver PCB:** The TC4427 gate driver board must be mounted close to the MOSFETs to keep gate wires short.
7. **Gate wiring:** Gate wires must be as short as possible and routed away from high-current bus wiring to prevent ringing and false switching.
8. **Spike protection:** TVS diodes and snubbers must be placed physically close to the switching node, not at the end of long wires.
9. **Keep-out rules:** Do not route thick wire bundles across heatsink fins; use the keep-out strip for signal wiring only.
10. **Routing rules:** Use TO-220 insulators; keep gate wires short; avoid routing across heatsinks; place spike protection at the node; stop testing if parts heat rapidly.
11. **Why this matters:** Most early failures occur here due to shorts, oscillation, or voltage spikes; correct layout prevents MOSFET damage.

Enclosure layout drawing set v14.2.3 — Sheet 04/04 — Build sequence + lock-in checklist

Use this as the pre-drill and pre-power checklist. Lock it in: no cosmetic iterations unless a real assembly blocker is found.

Build sequence (high-confidence order)

1) Print Sheet 01 at 100% scale. Verify the 50 mm bar and 1.00 inch box.
2) Mark and drill the big items first: DIN rails, bridge bank, supercap shelf, heatsinks.
3) Dry-fit every part with screws/standoffs. Fix fit issues now (before wiring).
4) Mount buses and shunts. Then mount bridges and ideal-diode modules.
5) Wire harvest path (Sheet 02). Only then wire the switching nodes (Sheet 03).
6) Door-mount the logic stack (Arduino, PCA9685, ULN2803A, BQ25570). Connect last.
7) First spin test: no inverter, short push pulses, measure voltages before adding loads.

Lock-in checklist (stop if any item fails)

- □ Print scale verified (50 mm bar measures 50 mm; 1 inch box measures 25.4 mm).
- □ DIN rails placed at locked X positions (R1 at 5 mm, R2 at 45 mm from backplate edge).
- □ 10 mm keep-out strip is clear of thick bundles (signal only across that gap).
- □ Bridge bank mounted (KBPC3510 ×9). Screw heads accessible with a screwdriver.
- □ Ideal-diode modules mounted. Orientation verified (no reverse installation).
- □ Supercap shelf mounted and shielded. No exposed metal can short its terminals.
- □ Every MOSFET has an insulator kit installed. Continuity test: tab is isolated from heatsink.
- □ Gate resistors installed (22 Ω). Gate wires are short and routed away from bus wires.
- □ TVS/snubber parts are located at the node (not at the end of long wires).
- □ Cold-start wiring matches Sheet 02 (shared harvest source; two-path split).
- □ Door-mounted logic is separated. No wiring pinches when closing the door.
- □ Team sign-off: two people cross-check against SVG sheets before first power test.

Gland numbering reminder (inside view)

Wall 3 glands: G1 is the right-most gland (inside view). Numbering increases right-to-left. G12 reserved for Hall sensors.

Figure 20-4. Build Sequence and Lock-In Checklist (Sheet 04)

This sheet tells you **what order to build things in and when to stop if something is wrong**.

Purpose: Use this page as a pre-drill, pre-power checklist before committing to wiring or testing.

Build sequence: The numbered list shows the safest order to assemble the enclosure so parts do not block each other later.

Scale check: Always verify the printed scale (50 mm bar and 1 inch box) before drilling anything.

Mount big parts first: DIN rails, bridge bank, supercapacitor shelf, and heatsinks must be mounted before smaller electronics.

Dry fit rule: Every part should be test-mounted with screws or standoffs before any wiring begins.

Wiring order: Build the harvest power path first, then switching nodes, then logic wiring last.

Door-mounted logic: Arduino, PCA9685, ULN2803A, and BQ25570 are mounted on the door and connected only after backplate wiring is complete.

First spin rule: Initial tests must be done with no inverter and short pulses only.

Lock-in checklist: If any checklist item fails, stop and fix it before continuing.

Electrical safety: Every MOSFET must be electrically isolated from the heatsink using insulator kits.

Thermal safety: Gate wires must be short, and no wire bundles may cross heatsink fins.

Cold-start verification: Confirm that the harvest source is split correctly into power and logic paths before spinning the rotor.

Gland numbering rule: Inside view of Wall 3 — G1 is the right-most gland; numbering increases right-to-left; G12 is reserved for Hall sensors.

Final rule: Do not proceed to power testing until two people cross-check the build against the SVG sheets.

Closing note

Build it like LEGO: finish one small block, test it, then adjust timing/phase in small steps until it spins up cleanly from a flick.

Appendix — MOSFETs 101 (Why N-MOSFET vs P-MOSFET, and where we use each)

Plain-language goal: We need electronic “switches” to connect a coil to +BUS (high side) and to connect a coil to GND (low side). MOSFETs are those switches. We pick **P-channel** parts for the **high side** and **N-channel** parts for the **low side**.

1) What a MOSFET is (defined simply)
MOSFET = Metal-Oxide-Semiconductor Field-Effect Transistor. It’s a voltage-controlled switch.

- **Pins: G**ate (control), **D**rain (one side of the switch), **S**ource (the other side).
- **V_GS:** How hard we push the switch "ON".
- **R_DS(on):** The resistance when ON (lower = cooler, better).
- **Body diode:** Every power MOSFET has a built-in diode between Drain and Source. Its direction matters for where the part lives in the circuit.
- **Logic-level MOSFET:** Turns fully ON with small V_GS (≈4–10 V). We use these on the low side.

2) The two jobs in our generator

- **High-side column switch:** Connects a chosen **column positive** line to **+BUS**. Lives **above** the coil.
- **Low-side row switch:** Connects a chosen **row return** line to **GND/Star-Ground**. Lives **below** the coil.

3) Why P-channel on the high side (columns A…K)
The high side's Source sits at +BUS. To turn a P-MOSFET ON, we pull its Gate lower than its Source. That's easy to do with a simple sinking driver (ULN2803A): it pulls the gate low to turn ON, and a resistor pulls it back up to +BUS to turn OFF.

- The **body diode** in a P-MOSFET (in high-side orientation) points **from Drain → Source** (from load toward +BUS). In normal operation, current flows **Source → Drain**, so the diode is **reverse-biased**.
- **Protection detail:** Because +BUS can be > the Zener voltage, the Zener will conduct during turn-on to clamp VGS. Use a 1–2.2 kΩ series current-limit resistor to keep clamp current safe.

Parts we use:

- **P-MOSFET:** IRF4905 (55 V, P-channel, TO-220).
- **Gate sink:** ULN2803A (8-channel open-collector sink).
- **Network:** 47 kΩ pull-up, 22–47 Ω gate stopper, 12 V Zener, 1–2.2 kΩ current-limit resistor.

4) Why N-channel on the low side (rows 1…5)
The low side's Source sits at GND. To turn an N-MOSFET ON, we drive its Gate higher than

Source by ~10–12 V. N‑channel parts have much lower R_DS(on) for the same size vs P‑channel, so they run cooler on the current‑heavy low side.

- The **body diode** points **from Source → Drain**. In normal operation current flows **Drain → Source**, so the diode is **reverse‑biased**.

Parts we use:

- **N‑MOSFET:** IRLZ44N (logic‑level N, TO‑220).
- **Gate driver:** TC4427 (dual) at 12 V supply.

5) Quick pictures in words (current & diode directions)

- **High side (P‑MOSFET):** +BUS → Source → (when ON) Drain → Column → Coil → Row → (when ON) N‑MOSFET → GND.
- **Low side (N‑MOSFET):** Coil current enters Drain, exits Source to GND when ON.

6) Why not N‑channel on the high side here?

You can use an N-MOSFET high side, but you need a high-side gate driver with a charge-pump/bootstrapping (to drive Gate ≈ +BUS + 10–12 V). That adds parts and complexity. For our student build, P-channel + ULN2803A is simpler and safer.

7) Choosing values (student cheat‑sheet)

- **Gate pull‑up (P‑FET):** 47 kΩ to +BUS → default OFF.
- **Gate resistors:** 22–47 Ω gate stopper, 1–2.2 kΩ series current‑limit.
- **Zener (P‑FET):** 12 V Gate→Source → limits |VGS|.
- **Gate driver (rows):** TC4427 at 12 V supply.
- **TVS:** 33 V TVS from column node → GND.
- **Flyback diode:** MUR460 from column node → +BUS.

8) Common mistakes to avoid

- **Exceeding V_GS:** Forgetting the Zener on P‑FET gates when +BUS > 15 V.
- **Wrong MOSFET type/side:** Putting an N‑MOSFET on the high side without a proper

high-side driver.

- **Body-diode the wrong way:** Flipping Source/Drain on a high-side P-FET.
- **Weak drive on low side:** Driving IRLZ44N directly from Arduino at 5 V (use TC4427 at 12 V).

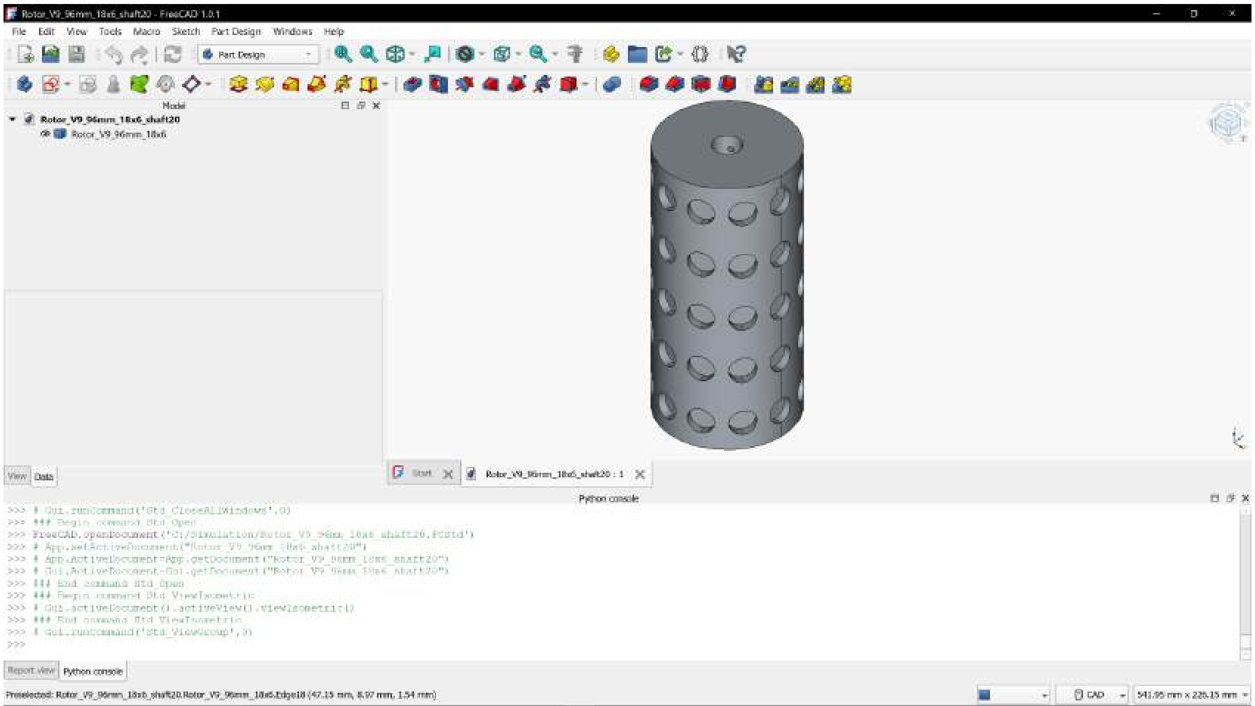
10) Mini decision table

Task	Best device	Why
Connect a column to +BUS (high side)	**P-MOSFET (IRF4905)** + ULN2803A sink	Easiest safe gate drive; body diode reverse-biased; minimal parts
Connect a row to GND (low side)	**N-MOSFET (IRLZ44N)** + TC4427 driver	Lowest R_DS(on), cool running; easy 0/12 V drive
Maximum efficiency upgrade later	N-MOSFET high side + high-side driver	Lower loss than P-FET, but adds complexity

Takeaway: Use **P-channel** for the **high-side column** switches (simple, safe gate drive with ULN2803A + Zener clamp). Use **N-channel** for the **low-side row** switches (cool, strong switch with the TC4427 driver). Keep the body-diode directions and gate-to-source limits in mind, and your switches will run cool and reliable.

Final CAD — FreeCAD Console Scripts

A) Rotor — V9 (Ø96, 18×6 pockets, Ø20 shaft, 3×M4)



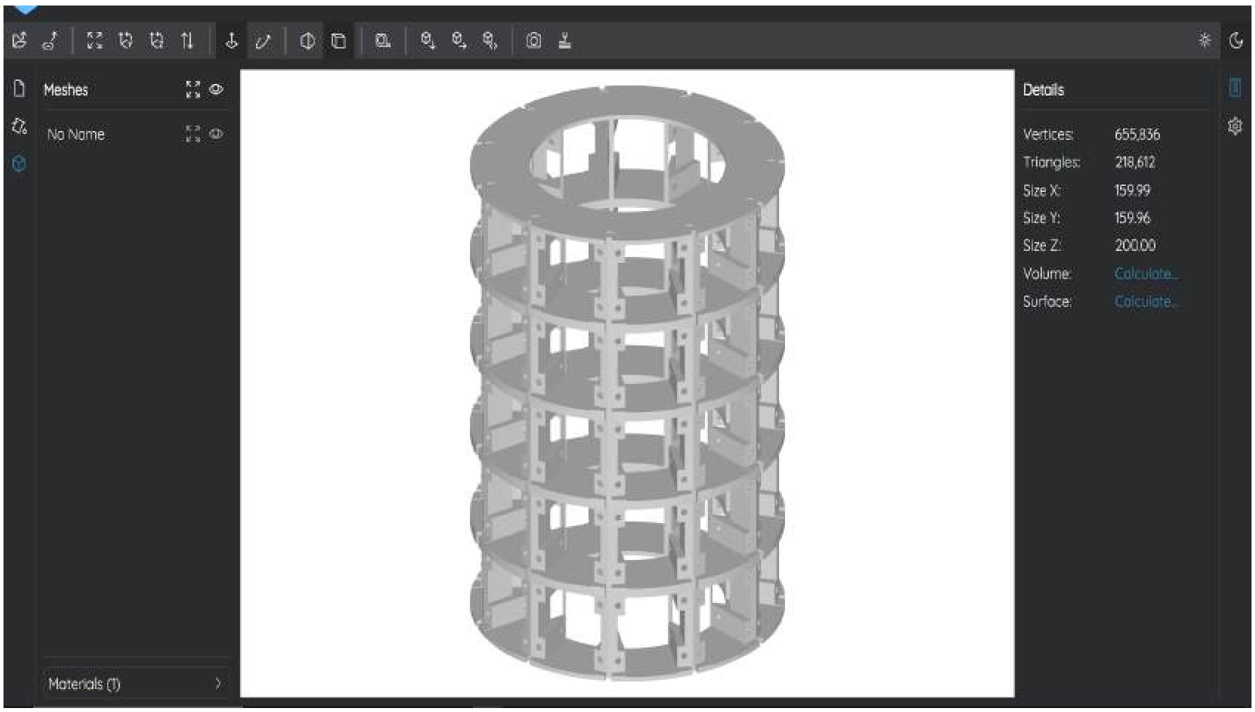
```
# Rotor_V9_96mm_18x6_shaft20.py

import FreeCAD, Part, math

from FreeCAD import Base

App=FreeCAD



DOC="Rotor_V9_96mm_18x6_shaft20"

ROTOR_D=96.0; L=200.0

SHAFT_D=20.0; SHAFT_BORE_EXTRA=0.10

MAG_D=18.2; MAG_T=6.2

ROWS=5; COLS=11

ADD_SETSCREWS=True

SCREW_CLEAR=4.2; NUT_AF=7.0; NUT_TH=3.2; NUT_SLACK=0.30;
HEX_DEPTH=NUT_TH+0.6
```

```
Z_FEATURE=(L/2.0)-10.0; ANGLES=[0.0,120.0,240.0]


docs=App.listDocuments(); doc=docs[DOC] if DOC in docs else
App.newDocument(DOC)

App.setActiveDocument(DOC); _=[doc.removeObject(o.Name) for o in list(doc.Objects)
if o.Name.startswith("Rotor")]


R=ROTOR_D*0.5; r_sh=(SHAFT_D+SHAFT_BORE_EXTRA)*0.5

rotor=Part.makeCylinder(R,L); bore=Part.makeCylinder(r_sh,L)

rotor.translate(Base.Vector(0,0,-L*0.5)); bore.translate(Base.Vector(0,0,-L*0.5))

rotor=rotor.cut(bore)


p=Part.makeCylinder(MAG_D*0.5, MAG_T+2.0)

p.rotate(Base.Vector(0,0,0),Base.Vector(0,1,0),90)

p.translate(Base.Vector(R-MAG_T,0,0))


row_pitch=L/float(ROWS); col_step=360.0/float(COLS); cuts=[]

for i in range(ROWS):

    z=(i*row_pitch)+(row_pitch*0.5)-(L*0.5)

    for j in range(COLS):

        ang=j*col_step

        c=p.copy(); c.translate(Base.Vector(0,0,z));
c.rotate(Base.Vector(0,0,0),Base.Vector(0,0,1),ang)

        cuts.append(c)
```

```
if ADD_SETSCREWS:

    Rhex=(NUT_AF+NUT_SLACK)/(2.0*math.cos(math.radians(30)))

    pts=[Base.Vector(Rhex*math.cos(math.radians(60*k)),
Rhex*math.sin(math.radians(60*k)),0) for k in range(6)]; pts.append(pts[0])

    hex_face=Part.Face(Part.makePolygon(pts));
prism=hex_face.extrude(Base.Vector(0,0,HEX_DEPTH))

    for ang in ANGLES:

        hp=prism.copy(); hp.rotate(Base.Vector(0,0,0),Base.Vector(0,0,1),ang)

        hp.translate(Base.Vector(R*math.cos(math.radians(ang)),
R*math.sin(math.radians(ang)), Z_FEATURE-HEX_DEPTH*0.5)); cuts.append(hp)

        sc=Part.makeCylinder(SCREW_CLEAR*0.5, R-(r_sh-1.0)+2.0, Base.Vector(0,0,0),
Base.Vector(1,0,0), 360.0)

        sc.rotate(Base.Vector(0,0,0),Base.Vector(0,0,1),ang)

        sc.translate(Base.Vector((r_sh-1.0)*math.cos(math.radians(ang)),
(r_sh-1.0)*math.sin(math.radians(ang)), Z_FEATURE)); cuts.append(sc)


rotor=rotor.cut(Part.Compound(cuts))

Part.show(rotor,"Rotor_V9_96mm_18x6"); doc.recompute()


r_center=(ROTOR_D*0.5)-(MAG_T*0.5)

pitch=(2.0*math.pi*r_center)/COLS; clear=pitch-MAG_D

print("ROTOR: Ø%.1f x %.1f, shaft bore=%.2f; pockets Ø%.1f x
%.1f"%(ROTOR_D,L,2*r_sh,MAG_D,MAG_T))

print("Layout %dx%d; tangential pitch≈%.2f; clearance≈%.2f
(OK>0)"%(ROWS,COLS,pitch,clear))
```

print("NOTE: ALL magnets NORTH-out; epoxy; add 3x M4 set-screws on shaft.")

B) Stator — (with holes)




```
exec(r'''
# Stator_V11_OD160_ID100_CoilV19_LOCKED23G_LIP05_M3VISIBLE.py
# OD=160, ID=100, L=200 (5x11) - Coil V19
# Change: end-cap screw circle to M3 at the same radius and mid-sector angles as the endcap.
# Holes are made visible by adding shallow face counterbores on both faces.

import FreeCAD, Part, math
from FreeCAD import Base
App = FreeCAD

# ------------------------ Core parameters ------------------------
ROTOR_OD     = 96.0
STATOR_ID    = 100.0
STATOR_OD    = 160.0
STATOR_LEN   = 200.0
NUM_ROWS     = 5
NUM_PER_ROW  = 11

# Air gap control
```

```
LIP_AIR_GAP  = 0.5    # mm
SHIM_DELTA   = 0.0

# End-cap screw circle (must match endcap script)
SCREW_CIRCLE_R   = 73.0         # mm
HOLE_ANGLES_MODE = "MID"        # "MID" uses (j+0.5)*ANG like the endcaps

# M3 hole spec (and make them visible at both faces)
M3_MODE     = "TAP"   # "TAP" = M3 tap pilots (2.5 mm), blind from both faces; "CLR" = 3.2
mm blind from both faces
PILOT_DEPTH = 8.0     # depth from each face
FACE_Z_MARGIN = 1.0   # uncut skin at face before pilot starts
M3_TAP_D   = 2.5
M3_CLR_D   = 3.2
FACE_SPOT_BORE_D = 5.5   # shallow face counterbore diameter, for visibility
FACE_SPOT_BORE_H = 0.6   # shallow face counterbore height

# Coil V19 geometry
TOP_W, TOP_L = 44.2, 35.0
POCKET_WIDTH, POCKET_HEIGHT, POCKET_DEPTH = TOP_W+0.2, 35.2, 6.0
BOTTOM_LIP_W, BOTTOM_LIP_L = 34.0, 14.0
WINDOW_W_SLACK, WINDOW_H_SLACK = 0.6, 1.0
M3_D, SPAN_W, SPAN_L, SAFE_WEB = 3.2, 34.0, 25.0, 1.0
THROUGH_W_CENTER = BOTTOM_LIP_W + WINDOW_W_SLACK     # 34.6
CENTER_BAND_H    = BOTTOM_LIP_L + 2.0                # 16.0
THROUGH_W_SIDEBAND = 2.0*(SPAN_W*0.5 - (M3_D*0.5 + 2.0))  # 27.0
THROUGH_H_TOTAL    = TOP_L + WINDOW_H_SLACK               # 36.0

# Grooves
WIRE_DIA, WIRE_DEPTH = 3.0, 3.0   # grooves ON

# -------------------------- Derived --------------------------
R_i, R_o, L = STATOR_ID/2.0, STATOR_OD/2.0, STATOR_LEN
ROW_SP = L/float(NUM_ROWS)
ANG    = 360.0/float(NUM_PER_ROW)

WALL_THK       = R_o - R_i
R_pocket_floor = R_o - POCKET_DEPTH
hole_depth     = max(0.2, WALL_THK - POCKET_DEPTH - SAFE_WEB)
R_hole_bottom  = R_pocket_floor - hole_depth
R_groove_floor = R_o - WIRE_DEPTH

# Enforce lip target radius
LIP_TARGET_R = (ROTOR_OD*0.5) + LIP_AIR_GAP + SHIM_DELTA
```

```
INNER_CLR, OUTER_OVS = 0.20, 0.80
INNER_OVS   = max(INNER_CLR, R_i - LIP_TARGET_R + INNER_CLR)
RAD_LEN     = WALL_THK + INNER_OVS + OUTER_OVS

assert abs((STATOR_ID - ROTOR_OD)*0.5 - 2.0) < 1e-6, "Rotor to stator radial gap must be 2.0
mm."
if not (0.0 <= LIP_TARGET_R <= R_i):
    raise ValueError("Lip target radius outside bore.")

# ------------------------ Helpers ------------------------
def refine_solid(s):
    if s.isNull(): return s
    try: s = s.removeSplitter()
    except: pass
    if len(s.Solids) == 1: return s.Solids[0]
    if len(s.Solids) > 1:
        f = s.Solids[0]
        for t in s.Solids[1:]:
            try: f = f.fuse(t)
            except: pass
        return f
    if len(s.Shells) == 1:
        try: return Part.Solid(s.Shells[0])
        except: return s
    return s

def safe_cut(base, tool):
    try:
        r = base.cut(tool)
        return base if r.isNull() else r
    except:
        return base

# ------------------------ New document ------------------------
DOC = "Stator_V11_OD160_ID100_CoilV19_LOCKED23G_LIP05_M3VISIBLE"
docs = App.listDocuments()
doc  = docs[DOC] if DOC in docs else App.newDocument(DOC)
App.setActiveDocument(DOC)
_ = [doc.removeObject(o.Name) for o in list(doc.Objects)]

# ------------------------ Base ring ------------------------
outer = Part.makeCylinder(R_o, L)
inner = Part.makeCylinder(R_i, L)
stator = outer.cut(inner)
```

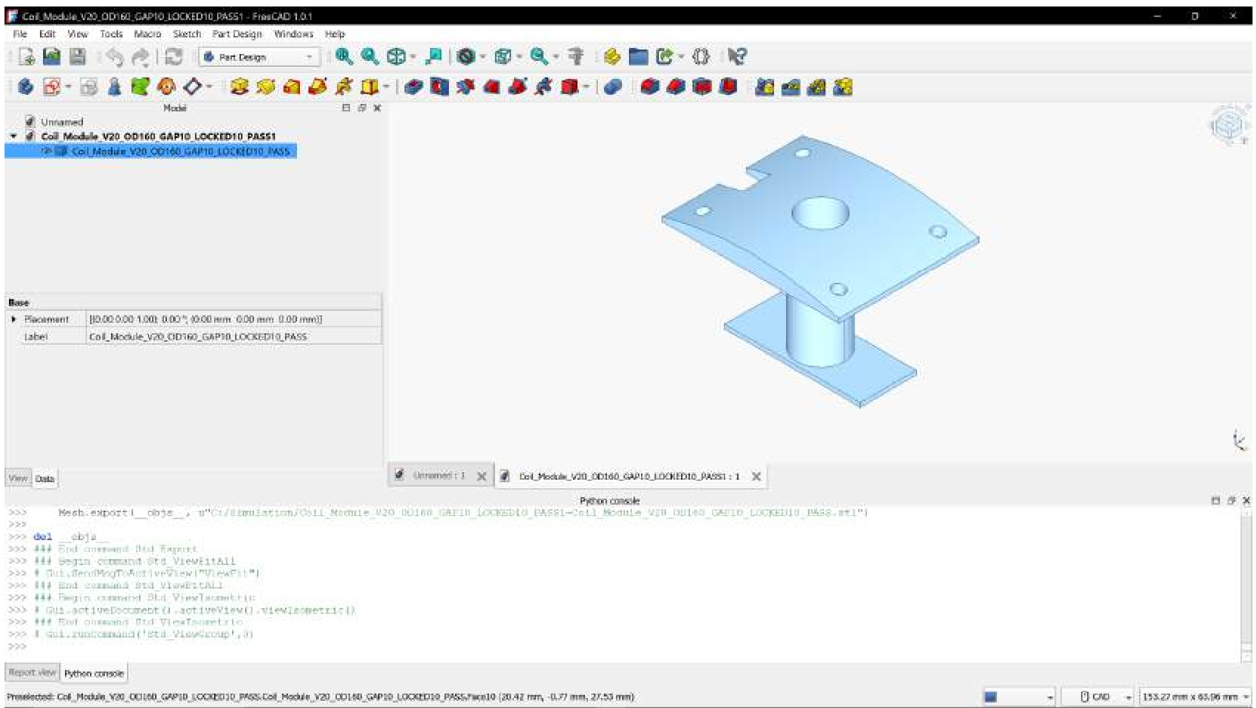
```
stator.translate(Base.Vector(0,0,-L*0.5))
stator = refine_solid(stator)

# ------------------------- Coil pockets + windows -------------------------
pocket_p = Part.makeBox(POCKET_DEPTH+1.0, POCKET_WIDTH, POCKET_HEIGHT)
pocket_p.translate(Base.Vector(R_pocket_floor, -POCKET_WIDTH*0.5,
-POCKET_HEIGHT*0.5))

x_start = R_i - INNER_OVS
w_center_p = Part.makeBox(RAD_LEN, THROUGH_W_CENTER, CENTER_BAND_H)
w_center_p.translate(Base.Vector(x_start, -THROUGH_W_CENTER*0.5,
-CENTER_BAND_H*0.5))

side_h = max(0.0, (THROUGH_H_TOTAL - CENTER_BAND_H)*0.5)
w_top_p = w_bot_p = None
if side_h > 0:
    w_top_p = Part.makeBox(RAD_LEN, THROUGH_W_SIDEBAND, side_h)
    w_bot_p = w_top_p.copy()
    w_top_p.translate(Base.Vector(x_start, -THROUGH_W_SIDEBAND*0.5,
CENTER_BAND_H*0.5))
    w_bot_p.translate(Base.Vector(x_start, -THROUGH_W_SIDEBAND*0.5,
-CENTER_BAND_H*0.5 - side_h))

m3_p = Part.makeCylinder(M3_D*0.5, hole_depth-0.05, Base.Vector(R_hole_bottom+0.05,0,0),
Base.Vector(1,0,0))
m3_offsets = [(-SPAN_W*0.5, -SPAN_L*0.5),
              ( SPAN_W*0.5, -SPAN_L*0.5),
              (-SPAN_W*0.5,  SPAN_L*0.5),
              ( SPAN_W*0.5,  SPAN_L*0.5)]

for i in range(NUM_ROWS):
    zc = (i*ROW_SP) + ROW_SP*0.5 - L*0.5
    for j in range(NUM_PER_ROW):
        ang = j*ANG
        # pockets + center window
        for tool in [pocket_p.copy(), w_center_p.copy()]:
            tool.translate(Base.Vector(0,0,zc))
            tool.rotate(Base.Vector(0,0,0), Base.Vector(0,0,1), ang)
            stator = safe_cut(stator, tool)
        # side bands
        if w_top_p:
            for tool in [w_top_p.copy(), w_bot_p.copy()]:
                tool.translate(Base.Vector(0,0,zc))
                tool.rotate(Base.Vector(0,0,0), Base.Vector(0,0,1), ang)
```

```
            stator = safe_cut(stator, tool)
        # 4x M3 pattern for the pocket plate
        for (dy, dz) in m3_offsets:
            h = m3_p.copy()
            h.translate(Base.Vector(0,dy, dz+zc))
            h.rotate(Base.Vector(0,0,0), Base.Vector(0,0,1), ang)
            stator = safe_cut(stator, h)

# ------------------------ End-cap screw circle (M3 and visible) ------------------------
theta = ANG
mid_off = 0.5*theta if HOLE_ANGLES_MODE.upper()=="MID" else 0.0

if M3_MODE.upper() not in ("TAP","CLR"):
    raise ValueError("M3_MODE must be TAP or CLR.")

cuts = []
face_spots = []

for j in range(NUM_PER_ROW):
    ang = j*theta + mid_off
    x = SCREW_CIRCLE_R*math.cos(math.radians(ang))
    y = SCREW_CIRCLE_R*math.sin(math.radians(ang))

    # visible shallow spot bores at both faces
    for z_face in (-L*0.5, L*0.5 - FACE_SPOT_BORE_H):
        spot = Part.makeCylinder(FACE_SPOT_BORE_D*0.5, FACE_SPOT_BORE_H)
        spot.translate(Base.Vector(x,y,z_face))
        face_spots.append(spot)

    if M3_MODE.upper()=="TAP":
        # blind pilots from each face
        z_bot = -L*0.5 + FACE_Z_MARGIN
        z_top =  L*0.5 - FACE_Z_MARGIN - PILOT_DEPTH
        hb = Part.makeCylinder(M3_TAP_D*0.5, PILOT_DEPTH);
hb.translate(Base.Vector(x,y,z_bot))
        ht = Part.makeCylinder(M3_TAP_D*0.5, PILOT_DEPTH);
ht.translate(Base.Vector(x,y,z_top))
        cuts += [hb, ht]
    else:
        # close-clearance 3.2 mm blind from each face
        z_bot = -L*0.5 + FACE_Z_MARGIN
        z_top =  L*0.5 - FACE_Z_MARGIN - PILOT_DEPTH
        hb = Part.makeCylinder(M3_CLR_D*0.5, PILOT_DEPTH);
hb.translate(Base.Vector(x,y,z_bot))
```

```
        ht = Part.makeCylinder(M3_CLR_D*0.5, PILOT_DEPTH);
ht.translate(Base.Vector(x,y,z_top))
        cuts += [hb, ht]

# apply visible spot bores first, then the pilots
stator = safe_cut(stator, Part.makeCompound(face_spots))
stator = safe_cut(stator, Part.makeCompound(cuts))

# ------------------------- Grooves (vertical + short horizontal joiners) -------------------------
vg = Part.makeBox(WIRE_DEPTH+1.0, WIRE_DIA, L+2.0)
vg.translate(Base.Vector(R_groove_floor, -WIRE_DIA*0.5, -L*0.5-1.0))
vtools=[]
for k in range(NUM_PER_ROW):
    g = vg.copy()
    g.rotate(Base.Vector(0,0,0), Base.Vector(0,0,1), (k+0.5)*ANG)
    vtools.append(g)
stator = safe_cut(stator, Part.makeCompound(vtools))

seg_outer = Part.makeCylinder(R_o+1.0, WIRE_DIA, Base.Vector(0,0,0), Base.Vector(0,0,1),
ANG)
seg_inner = Part.makeCylinder(R_groove_floor, WIRE_DIA, Base.Vector(0,0,0),
Base.Vector(0,0,1), ANG)
hseg = seg_outer.cut(seg_inner); hseg.rotate(Base.Vector(0,0,0), Base.Vector(0,0,1), -ANG*0.5)
hcuts=[]
for i in range(NUM_ROWS):
    z = (i*ROW_SP) + ROW_SP*0.5 - L*0.5
    for j in range(NUM_PER_ROW):
        hs = hseg.copy()
        hs.translate(Base.Vector(0,0, z - WIRE_DIA*0.5))
        hs.rotate(Base.Vector(0,0,0), Base.Vector(0,0,1), j*ANG)
        hcuts.append(hs)
stator = safe_cut(stator, Part.makeCompound(hcuts))

# ------------------------- Show / report -------------------------
stator = refine_solid(stator)
Part.show(stator, "Stator_V11_OD160_ID100_CoilV19_LOCKED23G_LIP05_M3VISIBLE")
doc.recompute()

bb = stator.BoundBox
rad_margin_to_groove = (R_groove_floor) - (SCREW_CIRCLE_R + max(M3_TAP_D,
M3_CLR_D)*0.5)
print("LOCKED23G_LIP05_M3VISIBLE - OK")
print("BBox: %.1f x %.1f x %.1f mm" % (bb.XLength, bb.YLength, bb.ZLength))
```

```
print("Lip target radius: %.3f  | Window inner edge R ~= %.3f (overshoot %.3f)" %
(LIP_TARGET_R, R_i - INNER_OVS, INNER_OVS))
print("End-cap screw circle R=%.1f, angles=(j+0.5)*theta, mode=%s, pilot depth=%.1f from each
face." %
    (SCREW_CIRCLE_R, M3_MODE.upper(), PILOT_DEPTH))
print("Face spot bores: %.1f x %.1f mm (both faces) - visible alignment markers." %
    (FACE_SPOT_BORE_D, FACE_SPOT_BORE_H))
print("Radial margin to groove floor: %.2f mm" % rad_margin_to_groove)
''')
```

C) Coil Module —




```
exec(r'''

# Coil_Module_V20_OD160_GAP10_LOCKED10_PASS.py
# Adds a rectangular wire pass-through on a chosen edge of the curved top plate.
# SLOT_FACE: "OD" (top), "ID" (bottom), "LEFT", or "RIGHT".
# Default places the slot in the CENTER of a SHORT SIDE ("LEFT").

import FreeCAD, Part, math
from FreeCAD import Base
App = FreeCAD
```

```
DOC = "Coil_Module_V20_OD160_GAP10_LOCKED10_PASS"

# ---- curvature / pocket fit ----
R_STATOR  = 80.0
TOP_W, TOP_L = 44.2, 35.0
TOP_THK   = 4.2
REACH     = 26.0

# ---- lip & tube ----
LIP_W, LIP_L = 34.0, 14.0
BOTTOM_LIP_H = 1.0
FERRITE_D    = 10.0
HOLE_CLEAR   = 0.2
TUBE_WALL    = 1.0
TUBE_OD      = FERRITE_D + HOLE_CLEAR + 2.0*TUBE_WALL
TUBE_R       = TUBE_OD*0.5

# ---- mounting (M3 countersunk) ----
M3_D = 3.2; SPAN_W = 34.0; SPAN_L = 25.0
CSK_D = 6.0; CSK_ANGLE = 90.0

# ---- lead grooves (disabled by default to avoid weak points) ----
ENABLE_TOP_GROOVES = False
CH_W = 3.0
CH_R = CH_W*0.5
TOP_CH_DEPTH = 1.6

# ---- NEW: rectangular pass-through slot (for 4x 18 AWG) ----
SLOT_FACE   = "LEFT"   # "OD" | "ID" | "LEFT" | "RIGHT"
SLOT_W      = 7.0      # width across the edge
SLOT_INSET  = 6.0      # depth into plate
SLOT_Z_SLACK= 2.0      # extra height so cut clears

# ---- build doc ----
docs = App.listDocuments()
doc  = docs[DOC] if DOC in docs else App.newDocument(DOC)
App.setActiveDocument(DOC)
for o in list(doc.Objects): doc.removeObject(o.Name)

# ---- Z layout ----
TOTAL_H      = TOP_THK + REACH
Z0           = 0.0
Z_WIND_START = BOTTOM_LIP_H
Z_WIND_H     = max(0.1, REACH - BOTTOM_LIP_H)
Z_WIND_END   = Z_WIND_START + Z_WIND_H
```

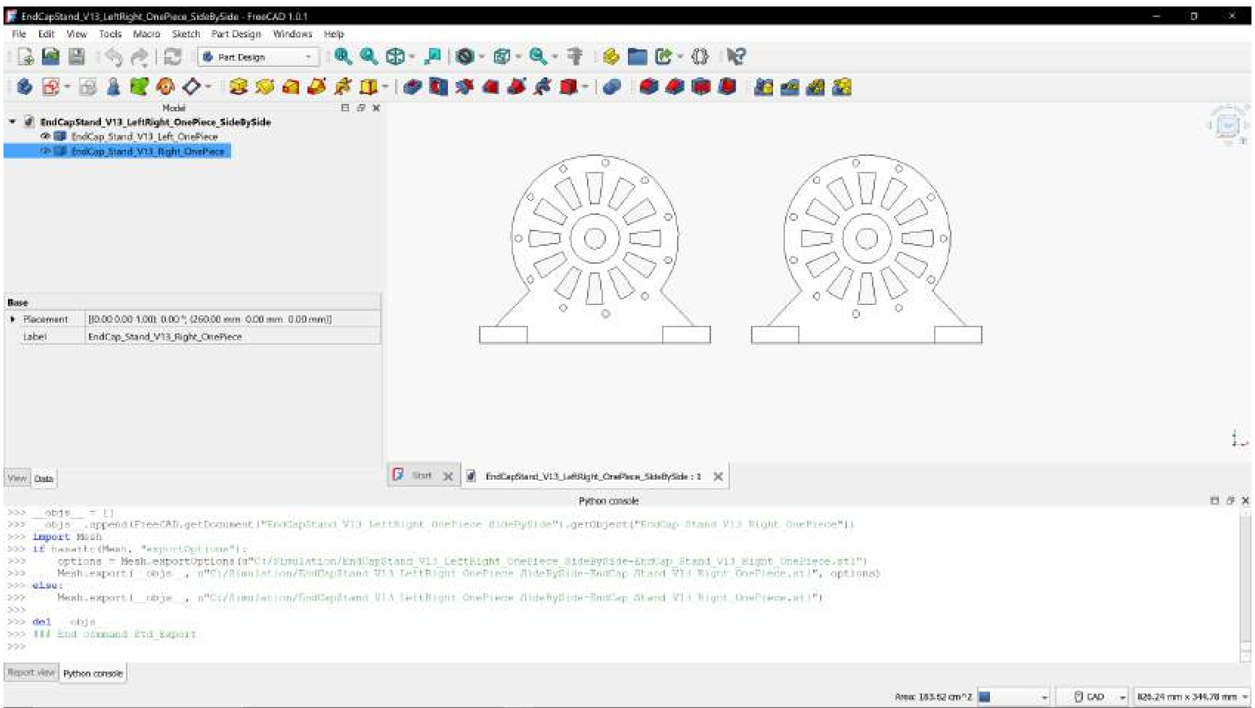
```
Z_PEAK      = TOTAL_H
Z_CYL_CENTER = Z_PEAK - R_STATOR

# ---- curved top plate ----
convex = Part.makeCylinder(R_STATOR, TOP_L + 60.0)
convex.rotate(Base.Vector(0,0,0), Base.Vector(1,0,0), 90)
convex.translate(Base.Vector(0, (TOP_L + 60.0)*0.5, Z_CYL_CENTER))
block = Part.makeBox(TOP_W, TOP_L, TOP_THK + 1.0)
block.translate(Base.Vector(-TOP_W*0.5, -TOP_L*0.5, Z_WIND_END))
curved_top = block.common(convex)

# ---- tube + lip ----
tube = Part.makeCylinder(TUBE_R, Z_WIND_H)
tube.translate(Base.Vector(0, 0, Z_WIND_START))
lip  = Part.makeBox(LIP_W, LIP_L, BOTTOM_LIP_H)
lip.translate(Base.Vector(-LIP_W*0.5, -LIP_L*0.5, Z0))
mod = lip.fuse(tube).fuse(curved_top)

# ---- ferrite bore Ø10.2
core = Part.makeCylinder((FERRITE_D + HOLE_CLEAR)*0.5, TOTAL_H + 4.0,
Base.Vector(0,0,-2.0), Base.Vector(0,0,1))
mod = mod.cut(core)

# ---- optional half-round grooves
if ENABLE_TOP_GROOVES:

    for sx in (-1, 1):

        x_edge = sx*(TOP_W*0.5)

        top_groove = Part.makeCylinder(CH_R, TOP_THK + 2.0,

                    Base.Vector(x_edge - sx*CH_R, -1.0, Z_WIND_END - CH_R),

                    Base.Vector(0,1,0))

        top_groove.translate(Base.Vector(sx*(-TOP_CH_DEPTH), 0, 0))

        mod = mod.cut(top_groove)

    bot_groove = Part.makeCylinder(CH_R, BOTTOM_LIP_H + 2.0,

                Base.Vector(-1.0, -CH_R, Z0 + BOTTOM_LIP_H*0.5 - CH_R),

                Base.Vector(1,0,0))
```

```
    mod = mod.cut(bot_groove)


# ---- rectangular slot (based on SLOT_FACE)
def cut_slot(mod):
    sz = TOP_THK + SLOT_Z_SLACK
    if SLOT_FACE.upper() == "OD":
        sx, sy = SLOT_W, SLOT_INSET + 2.0
        x0 = -SLOT_W*0.5; y0 =  TOP_L*0.5 - SLOT_INSET; z0 = Z_WIND_END - 1.0
    elif SLOT_FACE.upper() == "ID":
        sx, sy = SLOT_W, SLOT_INSET + 2.0
        x0 = -SLOT_W*0.5; y0 = -TOP_L*0.5 - 2.0; z0 = Z_WIND_END - 1.0
    elif SLOT_FACE.upper() == "LEFT":
        sx, sy = SLOT_INSET + 2.0, SLOT_W
        x0 = -TOP_W*0.5 - 2.0; y0 = -SLOT_W*0.5; z0 = Z_WIND_END - 1.0
    elif SLOT_FACE.upper() == "RIGHT":
        sx, sy = SLOT_INSET + 2.0, SLOT_W
        x0 =  TOP_W*0.5 - SLOT_INSET; y0 = -SLOT_W*0.5; z0 = Z_WIND_END - 1.0
    else:
        raise ValueError("SLOT_FACE must be one of: OD, ID, LEFT, RIGHT")
    slot_box = Part.makeBox(sx, sy, sz)
    slot_box.translate(Base.Vector(x0, y0, z0))
    return mod.cut(slot_box)


mod = cut_slot(mod)
```

```
# ---- countersunk holes
holes = [(-SPAN_W*0.5, -SPAN_L*0.5), ( SPAN_W*0.5, -SPAN_L*0.5),
         (-SPAN_W*0.5,  SPAN_L*0.5), ( SPAN_W*0.5,  SPAN_L*0.5)]
for (x,y) in holes:
    cyl = Part.makeCylinder(M3_D*0.5, 50.0, Base.Vector(0,0,-25.0))
    cs_h = ((CSK_D - M3_D)*0.5) / math.tan(math.radians(CSK_ANGLE*0.5))
    cone = Part.makeCone(CSK_D*0.5, M3_D*0.5, cs_h); cone.translate(Base.Vector(0,0,25.0))
    cutter = cyl.fuse(cone)
    z_surf = Z_CYL_CENTER + math.sqrt(max(0.0, R_STATOR*R_STATOR - x*x))
    n = Base.Vector(x, 0, z_surf - Z_CYL_CENTER)
    if n.Length != 0:
        zhat = Base.Vector(0,0,1)
        axis = zhat.cross(n); ang = zhat.getAngle(n)
        if axis.Length > 0:
            cutter.rotate(Base.Vector(0,0,0), axis, math.degrees(ang))
    cutter.translate(Base.Vector(x,y,z_surf))
    mod = mod.cut(cutter)

try: mod = mod.removeSplitter()
except: pass
Part.show(mod, "Coil_Module_V20_OD160_GAP10_LOCKED10_PASS")
doc.recompute()
```

```
print("LOCKED10_PASS — V20 with rectangular slot on", SLOT_FACE.upper())

print(f"Slot W={SLOT_W} mm, Inset={SLOT_INSET} mm, Plate height={TOP_THK} mm (grooves {'ON' if ENABLE_TOP_GROOVES else 'OFF'}).")

''')
```

D) Endcap + Stand —



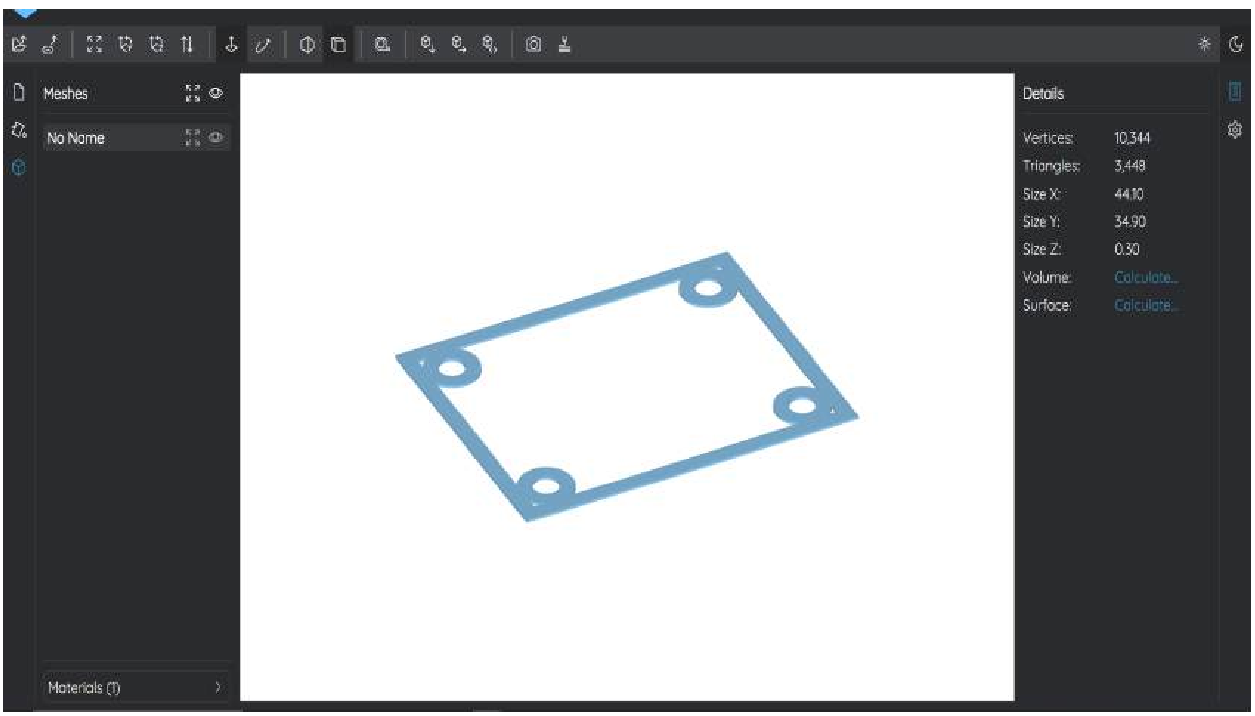

```
CODE = r'''
# EndCap_Stand_V13_LeftRight_OnePiece_SideBySide.py
# Goal: One-piece endcap/stand in the proven V13 geometry, plus a MIRRORED twin for the opposite stator face.
# Why a mirror? With 11 holes at mid-sector ((j+0.5)*θ), a 180° rotation misaligns by θ/2 ≈ 16.3636°. Mirroring preserves hole azimuths.

import FreeCAD as App, Part, math
from FreeCAD import Base

DOC="EndCapStand_V13_LeftRight_OnePiece_SideBySide"

# ----- Parameters (V13) -----
STATOR_OD=160.0
```

```
SCREW_CIRCLE_R=73.0; NUM_SCREWS=11
SCREW_D_CLR=6.4; CSK_D=12.0; CSK_ANGLE=90.0

SHAFT_D=20.0; BEARING_OD=47.0; BEARING_W=14.0
THK=18.0; BASE_W=220.0; H_TO_CENTER=100.0

FOOT_W=45.0; FOOT_H=16.0; FOOT_EXT=12.0
VENTS=12; MARGIN=10.0; SPOKE=0.5

def refine(s):
    try: s = s.removeSplitter()
    except Exception: pass
    return s

docs=App.listDocuments()
doc=docs[DOC] if DOC in docs else App.newDocument(DOC)
App.setActiveDocument(DOC)
for o in list(doc.Objects): doc.removeObject(o.Name)

Rout=STATOR_OD*0.5; Rsc=SCREW_CIRCLE_R; Htot=H_TO_CENTER+Rout

# ---------- Outline (proven V13) ----------
P1=Base.Vector(-BASE_W*0.5,0,0); P2=Base.Vector(BASE_W*0.5,0,0)
yt=H_TO_CENTER-50.0
Xt=(Rout**2 - (H_TO_CENTER-50.0)**2)**0.5 if Rout**2>(H_TO_CENTER-50.0)**2 else Rout
P3=Base.Vector(Xt,yt,0); P4=Base.Vector(-Xt,yt,0)
wire=Part.Wire([Part.makeLine(P1,P4),
                Part.Arc(P4, Base.Vector(0,Htot,0), P3).toShape(),
                Part.makeLine(P3,P2),
                Part.makeLine(P2,P1)])
plate=Part.Face(wire).extrude(Base.Vector(0,0,THK))

# ---------- Feet ----------
f1=Part.makeBox(FOOT_W,FOOT_H,THK+2*FOOT_EXT);
f1.translate(Base.Vector(-BASE_W*0.5,0,-FOOT_EXT))
f2=Part.makeBox(FOOT_W,FOOT_H,THK+2*FOOT_EXT);
f2.translate(Base.Vector(BASE_W*0.5-FOOT_W,0,-FOOT_EXT))
plate=plate.fuse([f1,f2])
for x in (-BASE_W*0.5+FOOT_W*0.5, BASE_W*0.5-FOOT_W*0.5):
    h=Part.makeCylinder(6.2*0.5, FOOT_H+2.0)
    h.rotate(Base.Vector(0,0,0),Base.Vector(1,0,0),90)
    h.translate(Base.Vector(x,-1.0,(THK+2*FOOT_EXT)*0.5-FOOT_EXT))
    plate=plate.cut(h)
```

```
# ---------- Bearing + Shaft ----------
ctr=Base.Vector(0,H_TO_CENTER,0)
shaft=Part.makeCylinder(SHAFT_D*0.5, THK+4.0); shaft.translate(ctr+Base.Vector(0,0,-2.0))
pocket=Part.makeCylinder(BEARING_OD*0.5, BEARING_W);
pocket.translate(ctr+Base.Vector(0,0,THK-BEARING_W))
plate=plate.cut([shaft,pocket])

# ---------- Vent ring ----------
Rin=(BEARING_OD*0.5)+MARGIN; RoutV=Rsc-MARGIN
if RoutV>Rin:
    ring=Part.makeCylinder(RoutV, THK+4.0).cut(Part.makeCylinder(Rin, THK+4.0));
ring.translate(Base.Vector(0,0,-2.0))
    step=360.0/float(VENTS); slot_ang=step*SPOKE
    wedge=Part.makeCylinder(RoutV+5.0, THK+6.0, Base.Vector(0,0,-3.0), Base.Vector(0,0,1),
slot_ang)
    slot=ring.common(wedge); slot.rotate(Base.Vector(0,0,0), Base.Vector(0,0,1), -slot_ang*0.5)
    cuts=[slot.copy()]
    for i in range(1,VENTS):
        c=slot.copy(); c.rotate(Base.Vector(0,0,0),Base.Vector(0,0,1), i*step); cuts.append(c)
    cr=Part.Compound(cuts); cr.translate(ctr); plate=plate.cut(cr)

# ---------- Screw circle (11 holes @ mid-sector) ----------
theta=360.0/float(NUM_SCREWS)
csk_h=(CSK_D - SCREW_D_CLR)*0.5 if CSK_ANGLE==90.0 else (CSK_D*0.5 -
SCREW_D_CLR*0.5)/math.tan(math.radians(CSK_ANGLE*0.5))
holes=[]
for i in range(NUM_SCREWS):
    ang=i*theta+0.5*theta
    x=Rsc*math.cos(math.radians(ang)); y=Rsc*math.sin(math.radians(ang))
    cyl=Part.makeCylinder(SCREW_D_CLR*0.5, THK+4.0);
cyl.translate(ctr+Base.Vector(x,y,-2.0)); holes.append(cyl)
    if csk_h>0:
        cone=Part.makeCone(CSK_D*0.5, SCREW_D_CLR*0.5, csk_h);
cone.translate(ctr+Base.Vector(x,y,THK - csk_h)); holes.append(cone)
plate=plate.cut(Part.Compound(holes))
plate = refine(plate)

# ---------- Handed pair ----------
left_one  = plate.copy()
right_one = plate.copy().mirror(Base.Vector(0,0,0), Base.Vector(1,0,0))  # mirror across YZ
plane
right_one.translate(Base.Vector(260.0, 0.0, 0.0))                    # move aside so they don't
overlap
```

```
Part.show(left_one,  "EndCap_Stand_V13_Left_OnePiece")
Part.show(right_one, "EndCap_Stand_V13_Right_OnePiece")

App.ActiveDocument.recompute()

print("="*60)
print("Built handed pair.")
print("Use LEFT on one stator face, MIRRORED RIGHT on the opposite face.")
print("Odd-symmetry note: with 11 holes, rotating a single part 180° misaligns by θ/2 ≈ %.4f°." % (180.0/11.0))
print("Each part has 11 through holes on SC R=%.1f, countersunk on the outer face." % SCREW_CIRCLE_R)
print("="*60)
'''
exec(CODE, globals(), locals())
```

E) Pocket Shims — (Used to increase air gap)




```
exec(r'''

# Pocket_Shim_V11_SINGLE_for_LOCKED23G.py

# Single shim for Stator_V10..._LOCKED23G pockets (44.4 × 35.2), with real M3 holes.
```

```
import FreeCAD, Part
from FreeCAD import Base
App = FreeCAD

DOC = "Pocket_Shim_V11_SINGLE"

# --- Geometry locked to your build ---
POCKET_W = 44.4     # tangential
POCKET_H = 35.2     # axial
FIT_SLACK = 0.30    # total; ~0.15 mm each side
W = POCKET_W - FIT_SLACK
H = POCKET_H - FIT_SLACK

# --- Shim thickness (edit this only) ---
THK = 0.30          # mm  (try 0.10 / 0.20 / 0.30 / 0.50)

# --- Details ---
FRAME = 2.0         # frame ring width
PAD_D = 7.0         # round pad Ø
M3_CLR = 3.2        # through clearance
SPAN_W = 34.0       # M3 pattern span (tangential)
SPAN_L = 25.0       # M3 pattern span (axial)
```

```
# ----- reset doc -----
docs = App.listDocuments()
doc  = docs[DOC] if DOC in docs else App.newDocument(DOC)
App.setActiveDocument(DOC)
for o in list(doc.Objects):
    doc.removeObject(o.Name)


# ----- build shim -----
# outer (centered)
outer = Part.makeBox(W, H, THK); outer.translate(Base.Vector(-W/2.0, -H/2.0, 0.0))
# inner cut to leave a frame
inner = Part.makeBox(W - 2*FRAME, H - 2*FRAME, THK)
inner.translate(Base.Vector(-(W - 2*FRAME)/2.0, -(H - 2*FRAME)/2.0, 0.0))
shim = outer.cut(inner)


# four round pads
pad = Part.makeCylinder(PAD_D*0.5, THK)
pads=[]
for x,y in [(-SPAN_W/2.0,-SPAN_L/2.0), (SPAN_W/2.0,-SPAN_L/2.0),
            (-SPAN_W/2.0, SPAN_L/2.0), (SPAN_W/2.0, SPAN_L/2.0)]:
    p = pad.copy(); p.translate(Base.Vector(x,y,0.0)); pads.append(p)
shim = shim.fuse(Part.Compound(pads))
```

```
# through holes Ø3.2 at pad centers

hole = Part.makeCylinder(M3_CLR*0.5, THK+1.0, Base.Vector(0,0,-0.5))

holes=[]

for x,y in [(-SPAN_W/2.0,-SPAN_L/2.0), (SPAN_W/2.0,-SPAN_L/2.0),

            (-SPAN_W/2.0, SPAN_L/2.0), (SPAN_W/2.0, SPAN_L/2.0)]:

    h = hole.copy(); h.translate(Base.Vector(x,y,0.0)); holes.append(h)

shim = shim.cut(Part.Compound(holes))


try: shim = shim.removeSplitter()

except: pass


Part.show(shim, "Pocket_Shim_T%0.2f" % THK)

doc.recompute()

print("Shim done: %.2f x %.2f mm, THK=%.2f; holes Ø%.1f at 34 x 25 mm." % (W, H,
THK, M3_CLR))
''')
```

Visual Electronic Parts Guide (Electronic and Control parts Required To Populate The Heartland Enclosure)

⚡ Power Electronics

×1 16V 83F Supercapacitor Module ⚡ HIGH CURRENT

1 bootstrap energy store for cold start

×9 KBPC3510 Bridge Rectifier

1 per harvest column × 9 harvest columns

×2 IRF4905 P-Channel MOSFET

1 per push column × 2 push columns (A and F)

×5 IRLZ44N N-Channel MOSFET

1 per row × 5 rows

×1 80V 50A Ideal Diode Module

1 cold-start path (Column D) between supercap and +BUS

×2 Aluminum U-Channel Heatsink

X1 for P-MOSFETs, x1 for N-MOSFETs

×2 Bus Bars

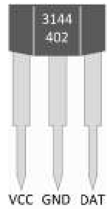
Go on wall 2

Control & Logic Electronics

×1 Arduino Mega 2560

One controller for the build

×2 A3144 Hall Effect Sensor

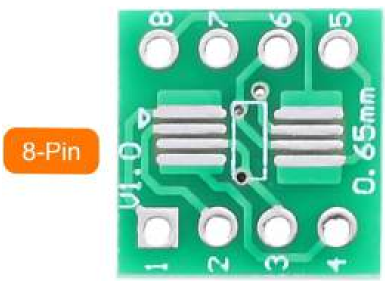


2 push columns (A and F) use 2 Hall sensors total

×3 TC4427COA Gate Driver (SOIC-8)

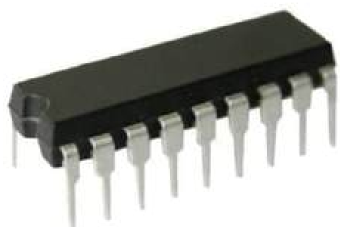
5 row gates need 5 driver channels; TC4427 has 2 channels → ceil(5/2)=3

×3 SOIC-8 to DIP-8 Adapter



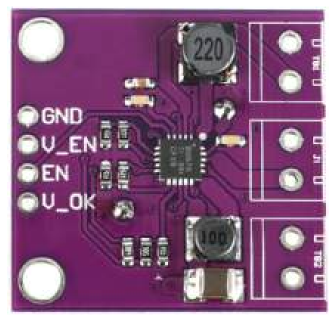
One adapter per TC4427 chip → 3

×1 ULN2803A Darlington Array



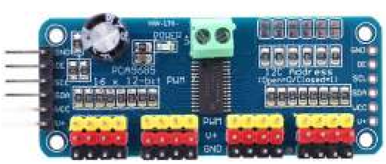
Used to sink P-MOSFET gates (2 channels needed); ULN2803A provides up to 8 → 1

×1 BQ25570 Energy Harvester ⚠ 5.5V MAX



1 logic-power harvester in the single cold-start subsystem

×1 PCA9685 16-CH PWM Driver



One PWM expander board provides all row timing/PWM channels (≤16)

×1 Low Voltage Disconnect (LVD)

1 cold-start/bus protection stage in the single bootstrap chain

Protection Devices

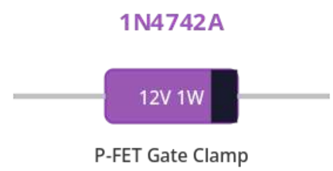
×2 MUR460 Ultrafast Recovery Diode

1 per push column × 2 (A and F) (column node → +BUS clamp)

×2 P6KE33CA TVS Diode (Bidirectional)

1 per IRF4905 × 2 (drain–source clamp at IRF4905)

×2 1N4742A 12V Zener Diode



1 per IRF4905 × 2

×1 1N5338B 5.1V 5W Zener Diode

1 clamp diode for the logic-tap input (explicit in guide)

×1 80A ANL Fuse (Gold Plated)

One main inverter feed protection fuse = 1

×1 ANL Fuse Holder Kit

One holder for the ANL fuse = 1

×1 1.5KE33A TVS

1 main bus clamp for the whole system (+BUS ↔ star ground)

×1 5D-11 NTC Inrush Limiter

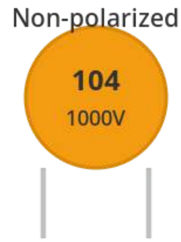
1 inrush limiter in the single cold-start chain

📦 Capacitors & Resistors

×9 1000µF 63V Electrolytic Capacitor

1 per harvest bridge × 9

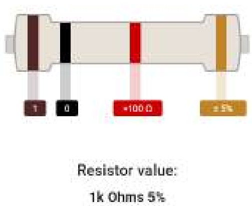
×50+ 0.1µF (104) Ceramic Disc Capacitor



1 per harvest bridge × 9

×Pack 1nF 2000V Film Capacitor (may be used)

X2 1KΩ Resistor

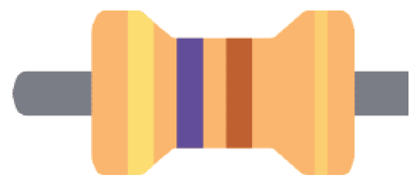


Per push column clamp chain uses 1 kΩ → 2

×3 100Ω 2W Metal Film Resistor

Series in the logic-tap chain (explicit in guide)

×2 47KΩ Resistor



1 per IRF4905 × 2 (yellow – purple – orange – gold)

Intentionally excluded: Tools (crimpers, meters, bench power supply), consumables that vary by wiring (wire lengths, ferrules, ring lugs, heatshrink), and any "optional / add if needed" suppression (snubbers, extra TVS) unless the guide makes them mandatory. This keeps the list strictly "explicitly determined" for the student build end state.

Parts Inventory For The Project:

1. 10 Pcs A3144 3 Terminal Sensitive Hall Effect Switch Sensor https://www.amazon.com/dp/B00SWK15ZE
2. Eventek DC Power Supply, 30V/5A Variable Power Supply with Encoder Adjustment Knob, Memory Switching Bench Power Supply with 4-Digits LED Display, 5V/3.6A USB Output & Banana Plug Lead https://www.amazon.com/dp/B0BFCWSXJK
3. DC 24V Step Down to 12V 10A 120W Voltage Regulator Buck Converter DC-DC Reducer Power Converter Waterproof Module Transformer for Golf Cart Club Car https://www.amazon.com/dp/B08D876FTC

4. DC Buck Converter 24V to 12V 10A 120W Step Down Transformer with Fuse Waterproof, Voltage Reducer for Golf Cart LED Light Motor Truck Vehicle Boat Solar System etc. (Accept DC 15-40V Inputs) https://www.amazon.com/dp/B08YYKSRJW
5. Power Supplies Module, DC-DC DC-DC Buck Converter 6.5V-60V to 1.25-30V 10A Adjustable Step Down Converter Power Supplies Module https://www.amazon.com/dp/B07TLHM71W
6. Power Supplies Module, DC-DC DC-DC Buck Converter 6.5V-60V to 1.25-30V 10A Adjustable Step Down Converter Power Supplies Module https://www.amazon.com/dp/B07TLHM71W
7. BQ25570 Boost Power Converter Energy Collector Module Solar Energy Collector Converter Boost Charging Buck Conversion Low Power Wireless Network Voltage Converter https://www.amazon.com/dp/B0BBLV85L8
8. 100 Non-Insulated Butt Splice Connectors 22-18 AWG by A Plus Parts House https://www.amazon.com/dp/B01ICSTUP2
9. 100FT 18 AWG Gauge Electrical Wire, DC 12V Hookup Red Black Copper Stranded Auto 2 Cord, Flexible Extension Cable with Spool for LED Ribbon Lamp Light or Low Voltage Products by MILAPEAK https://www.amazon.com/dp/B07CWQ6JPB
10. Absolute ANL80-2, 2 Pack ANL Fuses 80 Amp Gold Plated https://www.amazon.com/dp/B005EUTU5W
11. High Current 80V 50A Ideal Diode Module Solar Controller Anti-Reverse Charging Anti-Reverse Current Protection with Terminal https://www.amazon.com/dp/B0DDJFBF3B
12. WZRELB 1200W Pure Sine Wave Inverter 24V DC to 110V 120V AC Power Converter with LED Display, Off-Grid Portable Power Inverter with 2 AC Outlets for Home, RV, Truck, Camping https://www.amazon.com/dp/B0DF55LKX7
13. Two of BlueStars 2-Pack Bus Bar 12-48V DC 180A, 12 Terminal Power Distribution Block with 32 Wire Lugs - Compatible with Car Golf Cart RV Boat Solar Cell - Positive & Negative, Stainless Steel https://www.amazon.com/dp/B0D9GNWQLQ
14. 14 awg Silicone Electrical Wire 2 Conductor Parallel Wire line 50ft [Black 25ft Red 25ft] 14 Gauge Soft and Flexible Hook Up Oxygen Free Strands Tinned Copper Wire https://www.amazon.com/dp/B07FMTCHC1
15. High Current 80V 50A Ideal Diode Module Solar Controller Anti-Reverse Charging Anti-Reverse Current Protection with Terminal https://www.amazon.com/dp/B0DDJFBF3B
16. 1000uF 63V Capacitor, Aluminum Electrolytic Capacitors 105C High Temp 20pcs https://www.amazon.com/dp/B08MZ51MJR
17. 60 AMP ANL Fuse Holder Kit, 3 Pack https://www.amazon.com/dp/B0C6P6PWMX
18. 12 AWG Inline Fuse Holder with 30A ATC/ATO Blade Fuse, 5 Pack https://www.amazon.com/dp/B081DHT8Y7
19. Silicon Bridge Rectifier,10pcs KBPC3510 35A 1000V Metal Case Silicon Bridge Rectifier Designed for Single Phase Rectification https://www.amazon.com/dp/B07WQY2VJ1
20. 6 Gauge AWG (5 Feet Black + 5 Feet Red) Power Inverter Battery Cable Wire Kit for DC to AC Inverters RV, Car, Solar, Marine, Off-Grid https://www.amazon.com/dp/B01N0XLW2V

21. Thermal Pad 13W/mK [0.5~3mm], 85 x 45 x 0.5mm Silicone Cooling Pad Non Conductive Heat Resistance Extreme Odyssey Cover with Dual Self-Adhesive Films for PC Laptop PS4/PS5/CPU/GPU/Computer https://www.amazon.com/dp/B09XJTZYNV
22. Thermistor Resistors 5D-11 4A 5 Ohm Inrush Current Limiter Temperature Sensors Pack of 40 https://www.amazon.com/dp/B07KZZ8WV8
23. Ferrule Crimping Tool Kit, AWG 23-7 Self-adjustable Ratchet Wire Ferrule Crimper Plier Set with 1200pcs Wire Terminals Crimping Connectors Wire Ends Ferrules https://www.amazon.com/dp/B0CV4P1QXD
24. Car Heavy Duty Relay Switch 12V 30A SPDT 5Pin Wire Socket Plug Harness Waterproof Electrical Automotive Pack of 5 https://www.amazon.com/dp/B00RWV470O
25. Converter Module, DC-DC Buck Step Down Converter Adjustable Power Source Module 6.5V-60V to 1.25-30V 10A for Lithium or -Acid Battery https://www.amazon.com/dp/B07R1WHFMM
26. 150 Pcs ATO/ATC Standard Auto Fuse Kit - Blade Fuse Assortment Kit with Fuse Puller for Car, Truck, SUV, RV, Boat (2A 3A 5A 7.5A 10A 15A 20A 25A 30A 35A) https://www.amazon.com/dp/B07VWRK2VD
27. 24 AWG Control Cable Copper Wire Shielded Audio Cable Headphone Cable Signal Line 3-core 6 Meter Black https://www.amazon.com/dp/B0728HB6SM
28. 12 AWG Inline Fuse Holder with 30A ATC/ATO Blade Fuse, 5 Pack https://www.amazon.com/dp/B081DHT8Y7
29. 20 Pieces Clip-on Ferrite Ring Core RFI EMI Noise Suppressor Cable Clip for 3mm/5mm/7mm/9mm/13mm Diameter Cable, Black https://www.amazon.com/dp/B07CWCSNW9
30. 2000V 0.001UF 1NF P15 Metallized Film Capacitor Pitch=15mm https://www.amazon.com/dp/B0D112YLJS
31. 50 Pieces 22 Ohm 1/4W (0.25W) Metal Film Resistor 1% Tolerance https://www.amazon.com/dp/B00V9TIWQK
32. 22Pcs 320-2500 Grit Sanding Sponge Set, Ultra Fine Sanding Pads Soft Foam Sanding Block Wet Dry Sandpaper for Wood, Metal, Wall Polish https://www.amazon.com/dp/B0DZ69Z16Q
33. ABS Filament 1.75mm White 3D Printer Filaments, 1kg(2.2lbs) Neatly-Wound Spool, Strong, Durable, Good Heat Resistant, Less Odor, Dimensional Accuracy±0.03mm https://www.amazon.com/dp/B0F9NZDP3P
34. 120pcs Multicolored Dupont Wire 40pin Male to Female, 40pin Male to Male, 40pin Female to Female Breadboard Jumper Ribbon Cables Kit Compatible with Arduino Projects https://www.amazon.com/dp/B01EV70C78
35. 32 Pcs Double Sided PCB Board Prototype Kit for DIY Soldering with 5 Sizes Compatible with Arduino Kits https://www.amazon.com/dp/B072Z7Y19F
36. Quick Disconnects Vinyl Insulated Female Spade Wire Connectors, 16-14 AWG-6.3mm 1/4" Electrical Crimp Terminal https://www.amazon.com/dp/B0CWTH25M4
37. (Pack of 50 Pieces) Chanzon TVS Diodes P6KE33CA 600W 33V DO-15 (DO-204AC) Axial Bidirectional Channel 600 Watt 33 Volt https://www.amazon.com/dp/B079KK5FL8
38. 10 Sets C45 PCB DIN Rail Mounting Adapter 35mm 15mm Circuit Board Mounting Bracket Holder Carrier Clip https://www.amazon.com/dp/B09KZHY8G4

39. 50pcs PCB Mount Screw Terminal Block Connectors 5mm Pitch (2P 30pcs,3P 10pcs,4Pin 10pcs) https://www.amazon.com/dp/B09BQ78XWF
40. 15Pcs SOIC-8 to DIP-8 Narrow Adapter,0.5x0.5 Inch 8 Pin Double Side Prototype Connector Green PCB Converter Pin Board Adapter SO SOP8 to DIP8 https://www.amazon.com/dp/B0DGPFR9SB
41. 40 Pin Header - 2.54mm Single Row Male Header Connector for Arduino, Breadboard, PCB Connectors (Short Needle, 12) https://www.amazon.com/dp/B0DSKD99QY
42. 20pcs 100 ohm 2W 100R Metal Film Resistor 2W Resistance https://www.amazon.com/dp/B0DR2GKMYT
43. two of Aluminum U Type to 3P Heat Sink Dense Tooth Heat Cooler Fin Cooling 100 X 40 X 40mm for Applications Where is Whether Building Custom Concern Compact Cooling is Choice https://www.amazon.com/dp/B0DRYYQR8W
44. 5PCS TC4427COA713 TC4427COA TC4427C TC4427 SOP8 https://www.amazon.com/dp/B0BG97HQ7Y

Note: TC4427COA is an SOP-8 / SOIC-8 surface-mount package. Solder it to a SOIC-to-DIP adapter before using on perfboard/stripboard.

45. 2.7V 500F 16V 83F Automotive Super Farad Capacitor Module https://www.amazon.com/dp/B0CTMNN7S7
46. 1000 PCS Resistor Kit 1/4 W +-1% Premium Metal Film Resistors Assortment Kit, 25 Values (1 Ohm-1M Ohm) 40 Each, Breadboard & DIY Electronics, Includes 10k Ohm, 1k Ohm, 220 Ohm Resistors Pack https://www.amazon.com/dp/B0F4P352BB
47. DROK DC Buck Converter Adjustable, DC Voltage Regulator Adjustable 4.5-40V to 1.25-37V (36V 24V 12V 9V 5V 3V) Variable Voltage Stabilizer Power Supply Volt Reducer Step Down Transformer https://www.amazon.com/dp/B00SY37S00
48. Four of DIN Rail Terminal Blocks Gray Kit by International Connector 20 Dinkle DK2.5N 12 AWG Gauge 20A 600V Ground DK4N-PE Jumper DSS2.5N-10P End Covers End Brackets Grey https://www.amazon.com/dp/B07RWF97XT
49. Two of 12 Packs TO-220 Heatsink + Insulator/Mounting Kits for LM317 LM317t L7805 L7812 L78XX Voltage Regulator, IRF Z44N 3205 520N 630 1404 MOSFET Transistor (20mm x 15mm x 10mm) (Silver 12 Pack) https://www.amazon.com/dp/B08MW1YNJM
50. 100PCS 1W 12V 1N4742A 1N4742 DO-41 Zener diode https://www.amazon.com/dp/B07L87RVNP
51. (Pack of 20 Pieces) Chanzon MUR460 Super Fast Recovery Rectifier Diode 4A 600V 25-50ns DO-201AD (DO-27) Axial 4 Amp 600 Volt Electronic Diodes https://www.amazon.com/dp/B079KCJS5S
52. Pack of 8 1.5KE33A Diode TVS Single Uni-Dir 28.2V 1.5KW 2-Pin DO-201, Cut Strip, RoHS https://www.amazon.com/dp/B0CFGJ6LBG (no longer valid)
53. Genuine Loctite 638 x 50ml Maximum Strength Retaining Compound https://www.amazon.com/dp/B01FU86SF2 (no longer valid)
54. 210 Pieces Hex Standoff Spacers M4 Carbon Steel Male Female Metric Thread Spacer Motherboard Prototyping Accessories for PCB, Quad Copter Drone, Computer & Circuit Board Assortment Kit https://www.amazon.com/dp/B0BB6S86YZ

55. IRF4905P MOSFET Transistors 55V 74A IRF4905PBF P-Channel Power Mosfets Transistor 55 Volt 74 Amp TO-220 (Pack of 10Pcs) https://www.amazon.com/dp/B0CBKH44K5
56. 2Pcs Bridge Rectifier Diode, KBPC3510, AC to DC, 1000V 35A, 4 Pin, Full Wave Single Phase Metal Housing Bridge Rectifier (2) https://www.amazon.com/dp/B0D41DLM94
57. 40-Pack Metric M4 x 30mm Machine Screws for Cabinet Drawer Knob Pull Handle, 304 Stainless Steel, Truss Head Bolts, Phillips Drive https://www.amazon.com/dp/B06Y1WSSSD
58. New 10Pcs ULN2803A ULN2803 2803 Receiver Darlington Transistor Array-8 NPN IC https://www.amazon.com/dp/B09ZL3CJXR
59. 350PCS M4 Black Male Female Nylon Hex Spacer Standoff Screw Nut Set Assortment Kit (Black) for PCB Motherboard Circuit Board https://www.amazon.com/dp/B0BXT5Q5B3
60. 10Pcs IRLZ44N IRLZ44NPBF 47A 55V TO-220AB Power Mosfet N-Channel https://www.amazon.com/dp/B0BW8K31BF
61. 50PCS White Adhesive PCB Spacer Adhesive Bracket 0.25 Inch Support Height Adhesive Plastic Insulated Bracket Nylon Plastic Screw Stick PCB Spacers for Computer Motherboard Support https://www.amazon.com/dp/B0F1D3K19D
62. Voltage Module, Low Voltage Disconnect Automatic Cutoff Battery Module Board DC 12-36V for Accumulator for Lithium Battery https://www.amazon.com/dp/B09MP4TCLB
63. JZK PCA9685 16 Channel 12-Bit PWM Servo Motor Driver IIC Module for Arduino Robot https://www.amazon.com/dp/B06XSFFXQY
64. IRF4905P MOSFET Transistors 55V 74A IRF4905PBF P-Channel Power Mosfets Transistor 55 Volt 74 Amp TO-220 (Pack of 10Pcs) https://www.amazon.com/dp/B0CBKH44K5
65. High Current 80V 50A Ideal Diode Module Solar Controller Anti-Reverse Charging Anti-Reverse Current Protection with Terminal https://www.amazon.com/dp/B0DDJFBF3B
66. Polycarbonate Clear Plastic Sheet 12" X 12" X 0.118" (1/8") Exact with EasyRuler Film, Shatter Resistant, Easier to Cut, Bend, Mold Than Plexiglass. https://www.amazon.com/dp/B07VVGL6GV
67. Ceramic Capacitors Set, 50 Pack of Set 0.1uF 100nF High Voltage 1000V Disc 104M Ceramic Capacitors Set https://www.amazon.com/dp/B07VMN8G3D
68. 680 Pcs White Nylon Washers for Bolts, Plastic Flat Washers for Screws, 7 Sizes Assorted Hardware Assortment Kit, Nylon Sealing Washers Gaskets Spacer, M2.5 M3 M4 M5 M6 M8 M10 https://www.amazon.com/dp/B0DZHKYR44
69. Two bayite DC 6.5-100V 0-100A LCD Display Digital Current Voltage Power Energy Meter Multimeter Ammeter Voltmeter with 100A Current Shunt https://www.amazon.com/dp/B013PKYILS
70. RED WOLF 8 AWG 1/4" Battery Cable Lug 12 PCS w/3:1 Heat Shrink Tubing for Wiring Insulation Black Red 12 PCS for Automotive Electrical 8 Gauge Wire Copper Lug Marine Grade M6 Ring Terminal Ends https://www.amazon.com/dp/B0DWRV7CPS
71. HS Cable Tie Mounts Self Adhesive with Screw Hole Zip Wire Tie Mounting Base Holder (100 Pack) Squares 1x1 Inch,Black https://www.amazon.com/dp/B078P8TPPZ

72. 8 Gauge Silicone Wire 5 ft Red and 5 ft Black Flexible 8 AWG Stranded Tinned Copper Wire https://www.amazon.com/dp/B01A45YFME
73. Mr-Label Self-Laminating Wrap Around Cable Labels – Letter Sheet Laser Printer Only – for Wire Marking Identification (25 Sheets (800 Labels), White) https://www.amazon.com/dp/B078XR72Y7
74. 1200PCS Assorted Wire Ferrules Terminals Kit, AWG 22-10 Insulated Copper Crimp Connectors for Electric, Industrial, and Power Control Systems https://www.amazon.com/dp/B0DZJ1NV9T
75. 240 Pieces M4 Stainless Steel Button Head Hex Socket Head Cap Bolts Screws Nuts Assortment Kit + Wrench https://www.amazon.com/dp/B07CYM1N8R
76. Heartland AH14126C Polycarbonate Enclosure with Hinged, Locking, Clear Cover, 14 x 12 x 6 Inches https://www.amazon.com/dp/B00JJV1CT4
77. BP1412A Aluminum Back Panel Fits 14 x 12 Inch Centurion, Heartland, and Freedom Series Enclosures https://www.amazon.com/dp/B00JJV1HFS
78. 304 Stainless Steel Rods, 20mm x 254mm Metal Solid Shaft Round Rod Lathe Bar Stock for Model Plane Cars Ship DIY Craft Metal Working https://www.amazon.com/dp/B0DKXRJMHK
79. Loctite Instant Adhesive, 401, Prism, 20 Gram Bottle https://www.amazon.com/dp/B006GOKRSY
80. Digital Tachometer DT6236E Non Photo Tachometer 5 Digits 18 mm RPM Tach LCD Rotation Meter Tester https://www.amazon.com/dp/B0B5Y7DLC2
81. 50ml Threadlocker Blue M243, Thread Lock Medium Strength Lock Tight for Nuts, Bolts, Fasteners, and Metals to Prevent Loosening and Corrosion https://www.amazon.com/dp/B0DCJVZ2N5
82. Heat Tape, 2 Rolls Heat Resistant Tape, 108ft (10mm x 33m) Heat Tape for Sublimation & Transfer Tape, No Residue, Perfect for Heat Press, Electronics, Soldering, and Circuit Boards, Brown https://www.amazon.com/dp/B0D8T1JLLD
83. 10 of 25/64" x 1" 10x25mm Ferrite Rod Bar 6 Pcs for Antennas https://www.amazon.com/dp/B0087ZS2BY
84. Two of 6204-2RS Deep Groove Ball Bearings P5 20x47x14mm Lubricated Bearings Durable Black Silver Tone Scooter Skateboard Wheel https://www.amazon.com/dp/B0DG5WH1KP
85. M4 Grub Screws 304 A2-70 Stainless Steel Cup Point Hex Socket Set Screws (20Pcs, M435mm) https://www.amazon.com/dp/B07YSKYFWM
86. J-B Weld Pro Size ClearWeld 5 Minute Set Epoxy https://www.amazon.com/dp/B07TTTYKGS
87. Two of 18 AWG - 5 lb (1000 feet) 99.9% Pure Copper Wire, Enameled Magnetic Wire for Motor, Transformer, Magnetic Coil, & Electroculture Gardening, Winding Magnet Wire - 220°C (428°F) Thermal Class https://www.amazon.com/dp/B0BWL3HFZD
88. Desoldering Wick and Soldering Flux Paste for Electronics Repair, No Clean Flux(10CC), Solder Removal Kit https://www.amazon.com/dp/B0D3H8NZLP
89. Tip Tinner For Soldering Iron Tip Cleaning, Supplied By Wonderway https://www.amazon.com/dp/B0DGPP78NY

90. 63-37 High Purity Tin Lead Rosin Core Solder Wire for Electrical Soldering, Content 1.8% Solder flux (0.6mm, 50g) https://www.amazon.com/dp/B09KM28JT2
91. Four of 30 Pack Fridge Magnetic, 12x3mm Refrigerator Magnets, Neodymium Magnet, Small Magnets, Magnetic for Fridge, Magnets for Whiteboard, Magnet for Crafts https://www.amazon.com/dp/B0B5G83M1W
92. Klein Tools 1005 Cutting / Crimping Tool, Made in USA, Cutter / Crimper for 10-22 AWG Terminals and Connectors, Terminal Crimper for Insulated and Non-Insulated Terminals https://www.amazon.com/dp/B0006M6Y5M
93. Two of 90pcs Reusable For Wago 221 Electrical Connectors Wire Block Clamp Terminal Cable
94. 600/2600pcs 130 Values Resistor Kit 1/4W, 1 Ohm-3M Ohm RoHS Compliant Melk Temn Resistor Assortment Kit Metal Film Fixed 1% Electrical Resistors Set Assortment Resistance Optional 100% New
95. 100nf (104 & 101) + 10nf (103) - High Precision ±0.1% Tolerance, Low Loss & High Stability, Axial Lead Wire, Wide Temperature Range (-40°C to +105°C) for Electronics, Filters, Power Supplies, Audio, RF, Automotive - Durable Ceramic & Metal Construction
96. 1pc 20/22/24/26 AWG PVC Insulated Wire, 2 Core Cable, 12V Low Voltage Flexible Wire For Automotive, LED Lighting, Marine Use
97. Expandable Black PET Cable Sleeve - 393.7inch Length, Insulated Braid for Wire Protection, Fits 2-35mm Diameter, Wire Gland, Tight
98. Two of 8pcs Nylon M8-M20 Gland Head Cable Connectors - Metric Threaded Fittings with Sealed Design, Black/White, Easy Assembly for Automotive, Plumbing & Industrial Use
99. (25pcs) 1 Amp Standard Car Fuses, 1A Automotive Fuse (ATO/ ATC), Blade Fuse for Car/ RV/ Truck/ SUV/ Motorcycle/ Boat
100. 800PCS Solder Seal Wire Connectors Tubings - Insulated Waterproof Electrical Butt Terminals for Marine Automotive Motorcycle Wiring Boat.
101. Mini Survival Compass - Waterproof, Acrylic, Essential Outdoor Adventure and Exploration Tool with Lanyard, Suitable for Camping, Hiking, Hunting, and Survival - Includes Hook, Convenient to Hang on Backpack or Gear, Without Battery
102. 1pc Versatile Welding Stand with Flexible Arms & Stable Base - Ideal for Electronic Repairs, Jewelry Assembly, Model Building, Hobby Crafts & Workshop Tasks - Precision Clamping Base, Uncharged, No Battery Needed, Precision Tool | Versatile Design | Plastic Construction
103. Heavy-Duty Circuit Board Repair Mat with Non-Slip Surface, Precision Scale & Component Guides - Heat-Resistant Durable Plastic Workbench for Soldering
104. Dual purpose Manual/ Coil Winder Hand Coil Winding Machine 0-99999 Replacement Wheel Maintenance
105. Four of 480pcs M3 Flat Head Self Tapping Screws, Small Cross Drive Head Self Tapping (no longer available)
106. Full Version Starter Learning Kit MEGA 2560 With LCD1602 Servo Motor
107. Oscilloscope Multimeter Signal Generator Three-in-one Tester
108. Digital Caliper, IP54 Stainless Steel Caliper Measuring Tool, Vernier Caliper (no longer available)

109. 20Pcs 1N5338B 1N5338 5W 5.1V Zener Diode
https://www.amazon.com/dp/B0CGM9GCPP

Open-Source and Prior-Art Notice:

Scope of Disclosure:

This work discloses a generator architecture in which **individual stator coils are dynamically switched between continuous energy harvesting and angle-timed impulse drive**, while **all remaining coils continue harvesting**, using a **one-hot selection scheme**, **uniform rotor magnet polarity**, and **explicit electrical energy accounting** for all drive and harvest paths.

Authorship verification (commitment hash):
240a1b3f0e70eac2ca9bd54af3d5a86fd23ad0882aac0e39b162b04a75d20265 | ID:
6522d8ac-fd02-4025-b117-4638b9f0bba6