RDS 2025 Motor Driver Documentation

Charlie Seifert

Links

- Motor driver design spreadsheet
- Altium project
  - Datasheets should be linked to each part
- PDF of schematics
  - Actually really nice – click on a green block to jump to that sheet

Intro

Motivation

Why bother making a new motor driver?

- Motor driver is a cool thing to try to get working
- Ivor's board has room for improvement, per feedback
- ODrive kinda works
  - Clunky (1 per motor)
  - Difficult to cleanly integrate

Goals

- Drive 2 brushless DC motors with FOC
  a. Number of motors was driven by mechanical packaging, wiring, board space
  b. Support 24V, up to 30A phase current
- Improve current sensing
  a. Faster
  b. Directly measure all 3 phases (not 2 then infer 3rd)
- 40kHz PWM to enable higher-speed control
- Use GaN FET switches (fast and cool)
- Reasonable package size for mechanical teams
- ***Make something work!***

Design decisions to meet goals

- Inline-phase shunt resistor current sensing on all phases
- Dedicated ADCs on board = 20x faster current sensing
- Use GaN FET half-bridge IC
  - GaN = extremely fast switching speeds
  - But using IC avoids gate drive complexity

More design decisions

- Stack driver board with control board
  - Decouples development of both, but doesn't add additional wiring or mounting
  - Allows more advanced logic to be developed in the future, independent of driver
    - People wanted to use STM32, for example
- Hardware dead time enforcement
  - GaN half-bridge IC does not automatically prevent shoot-through
  - My circuitry prevents shoot-through always, does not rely on software
- Connectors
  - Simple headers! Makes it easy to test this board with a breadboard
- MCU
  - Teensy 4.0. Fast and well-supported within CRB, so less risk
- Put test points everywhere

Process and Implementation

Current sensing

- Different approaches: DC link or phase current
  - Measure only DC link current = limit control sophistication

    → *We want to measure individual phase current*

- Phase current sensing
  - High- or low-side sensing, near respective FET
    - Can not see phase current during OFF state of FET

    → *The answer: Inline phase current sensing*

- Inline phase current sensing
  - Shunt
  - Hall effect
    - Isolated (sensor next to conductor) or non-isolated (flows through chip)

Current sensing cont.

- Hall effect nice for isolation: HV, no worry about common-mode voltages
- *Shunt is less complex and very accurate with right amplifier*
  - Amplifier more internally complex and expensive
    - Buuut nowadays "expensive" = $2.53
  - High CMRR = Common-mode rejection ratio. Quantifies a non-ideality of the amp
    - "How much does common-mode voltage sneak into the output?"

- Conclusion: Inline phase shunt resistor current measurement with high-CMRR differential amplifier (made for current sensing)
  - TI INA24X family
  - Must reject common-mode because shunt R sees crazy swings

Current sensing cont.

- How to speed up? → External ADC
  - Dedicated hardware that we can get 1MSPS from = 1us ADC capture

    → 20x improvement

Gate driving and power stage

- We want high switching speeds → high rate of charge and discharge of gate capacitance → high transient currents at gates → *gate drivers needed*
  - Connecting GPIOs straight to FET gates limits slew rate and thus switching speed

- Simplest, quickest to develop: Gate driver within half-bridge IC
  - One chip per leg of driver

Dead time control

7.3.1 Control Inputs

The LMG2100R044's inputs pins are independently controlled with TTL input thresholds and can can support 3.3-V and 5-V logic levels regardless of the VCC voltage.

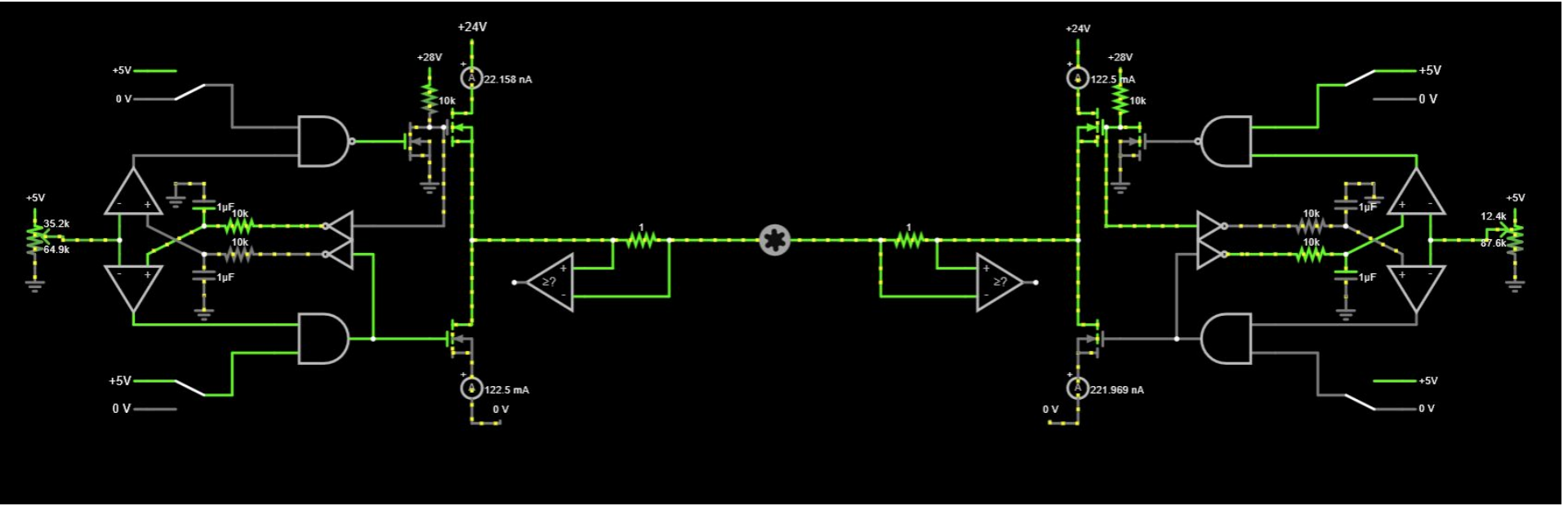
In order to allow flexibility to optimize deadtime according to design needs, the LMG2100R044 does not implement an overlap protection functionality. If both HI and LI are asserted, both the high-side and low-side GaN FETs are turned on. Careful consideration must be applied to the control inputs in order to avoid a shoot-through condition.

- Background
  - Shoot-through is the condition of shorting the power supply by overlapping when high- and low-side switches are closed
  - "Dead time" is time between the opening of one switch and closing of the other
    - Very short, ns for GaN
- Why hardware dead time control?
  - GaN half-bridge does not protect against shoot-through
  - Don't assume software will get it right – cost is very high
    - A fried half-bridge would halt all testing and require risky rework
  - First tests will not be done with software at all (button control of PWM inputs)
  - Also allows us to convert the GaN half-bridge's dual HI and LO inputs into single-ended
    - On top of enforcing dead time, this circuit is basically a decoder: 1 bit to 2 bits

Dead time control - implementation

- Extensively tested in falstad.com
  - See next slide for circuit
- Almost certainly more components than it could be, but works!
- What it does
  - Logic in front of each half-bridge input that checks that BOTH:
    - User wants this input (e.g., low side input LO) to be logic HIGH
    - Other input (e.g., high side input HI) has been logic LOW for sufficiently long dead time
  - Dead time tunable in two ways
    - Potentiometer changes threshold on all comparators that check for the second condition
      - Real-time, and affects all 6 phases!
      - In parallel with R divider for permanent setting, once desirable threshold is found
    - R and C values for RC circuit going into other input of comparator
      - Requires rework

Dead time control - falstad circuit

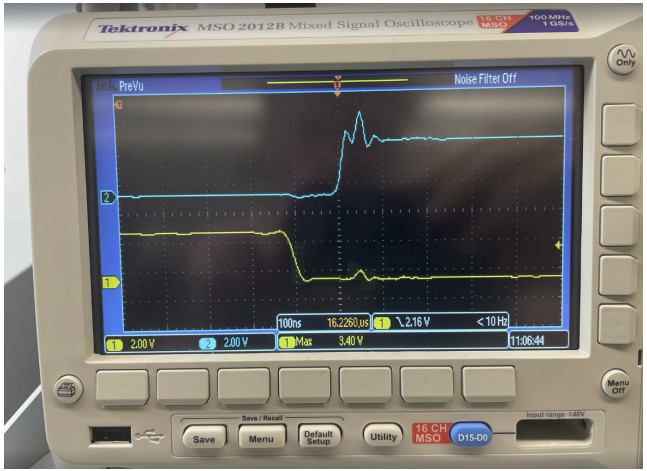


Steps to simulate

1. ← copy this text
2. Go to falstad
3. File → Import From Text

Dead time control - testing

- Testpoints
  - One on output of both inputs' RC circuit (input to comparator) for all phases
  - Single one for threshold
  - Idea was you'd scope both of these simultaneously to see if the circuit was behaving
- Found it more useful to directly measure HI and LO, see scope screenshot
- Findings
  - Works!
  - But much longer delay than expected
    - See "observed" table here
  - Likely unaccounted-for propagation delay



Stack

- Upper board = MCU + sensing
  - MCU
  - Interface to system
  - Connectivity to sensors
- Lower board = Motor driver
  - FET ICs
  - Bulk caps
  - Current sensing
  - Power connectors (24V, phases)
  - Motor driver board got the 24V and passed up only 3V3, 5V, and GND through headers

Connectors

- Logic pins were just headers
  - Simple as just placing headers pins, have stacked board mirror and place on bottom
  - Female on motor driver side b/c it receives 24V directly – daughterboard doesn't need 24V
  - Very little tolerance if wrong → communicate requirements!
  - We had no problems because I added the connectors and locked them in stack board design
  - Placement was driven by motor driver (denser layout)
  - SO NICE to prototype with. No need to make a harness or anything
- Power connectors
  - Screw terminals
    - Also very quick to work with
    - But not the most secure connection unless you really torque, then housing shifts and you strain soldered joints (to the detriment of my mental health, see Problems)

Power

- 24V informed by motor selection
- 5V to power GaN half-bridge's logic side, some other stuff because I had it
- 3V3 to power logic
- Regulation: 24V →(DC-DC)→ 5V →(3V3 LDO)→ 3V3
  - DC-DC for 5V regulation for high efficiency (LDO would burn much power)
    - Simple enough to implement – see Schematics
  - LDO for 3V3 regulation for rock-solid logic power
    - 5V line will have noise from DC-DC switching element, LDO does not
    - 3V3 powers sensitive things like the current sensing circuitry; don't want noise on Vdd

Bringup

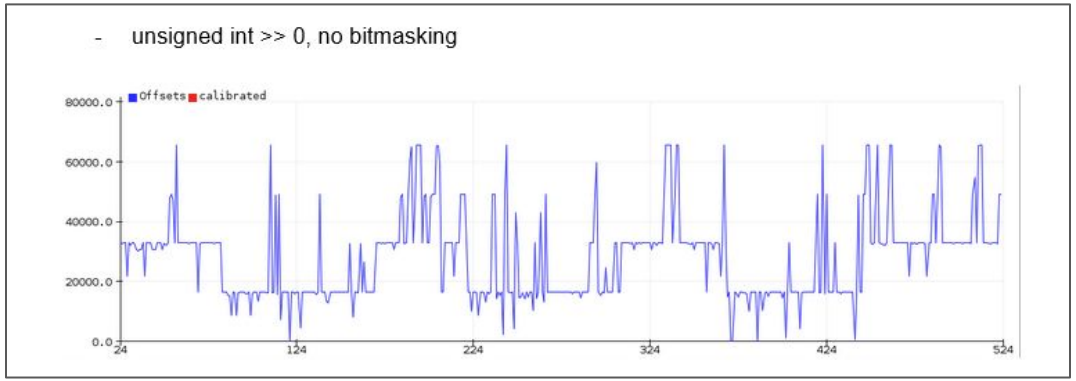
- Soldering issues
  - 12+ hours of rework
  - Possible damage to components due to intermittent power plane short
    - Pretty sure due to cold joints from lead-free solder
- Dead-time enforcement circuitry functions as intended
  - Tunable with potentiometer (real-time), or RC value
  - Testpoints helped greatly
- Half-bridge IC tested to sink 24A = 576W
  - Pulses when sourcing from 24V – unknown reason
- Current sense
  - Analog side (shunt and amplifier) function as intended
  - Had to debug SPI ADC output, eventually functioned as intended

Problems

- Assembly shorts (par for the course)
  - Was a grind. Don't design a dense board if you want to avoid these
- Power plane short
  - Wiggly 24V input connector; 5V and 3V3 would seemingly randomly rise to 19V, then 24V
  - Changed when pressure was applied to 24V input connector
  - Solved by resoldering with not-leadless solder
- SPI ADCs outputting strangely
  - See plots
  - Solved by delaying CS (90% confident)
- Half-bridge not continuously closing high side
  - Really weird pulsing behavior, see waveforms
  - Can continuously close low side and sink 24A
- Caught on fire during calibration
  - MCU was not seeing encoder readings
  - Similar setup to a test that was run just prior, just no scope probes on SPI lines

Tests - SPI ADCs

- Simple test for SPI ADCs:
  - Breadboard wires to MISO, CLK, CS using SPI0 and pin 10 (I think) as CS
  - Scope MISO, CLK
- Didn't work. MISO looked very strange, near-random
- Only flash of hope that the ADCs weren't totally bugging was this plot of the data the Teensy received over SPI from the ADC:
  - Changed in response to current being sent, but in a … weird looking way

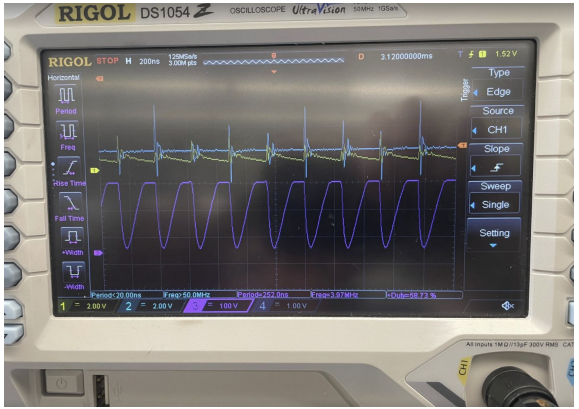


Tests - SPI ADCs - Findings

- Did not scope CS until later! (night before demo)
  - And saw that it was staying low for only ~2 clock cycles
  - It should stay low for the entire transaction
- Was fixed by adding delay:
  - digitalWrite(cs_pin, LOW);
  - raw16 = spix.transfer16(0x0000);
  - **delayMicroseconds(1);**
  - digitalWrite(cs_pin, HIGH);
- Theory: ADC sends actual data only when CS is low, so bringing it prematurely high would corrupt most of the lower bits
- Lessons: check *all* your comm bus lines; looking at received number over serial print is an unreliable observation – scope

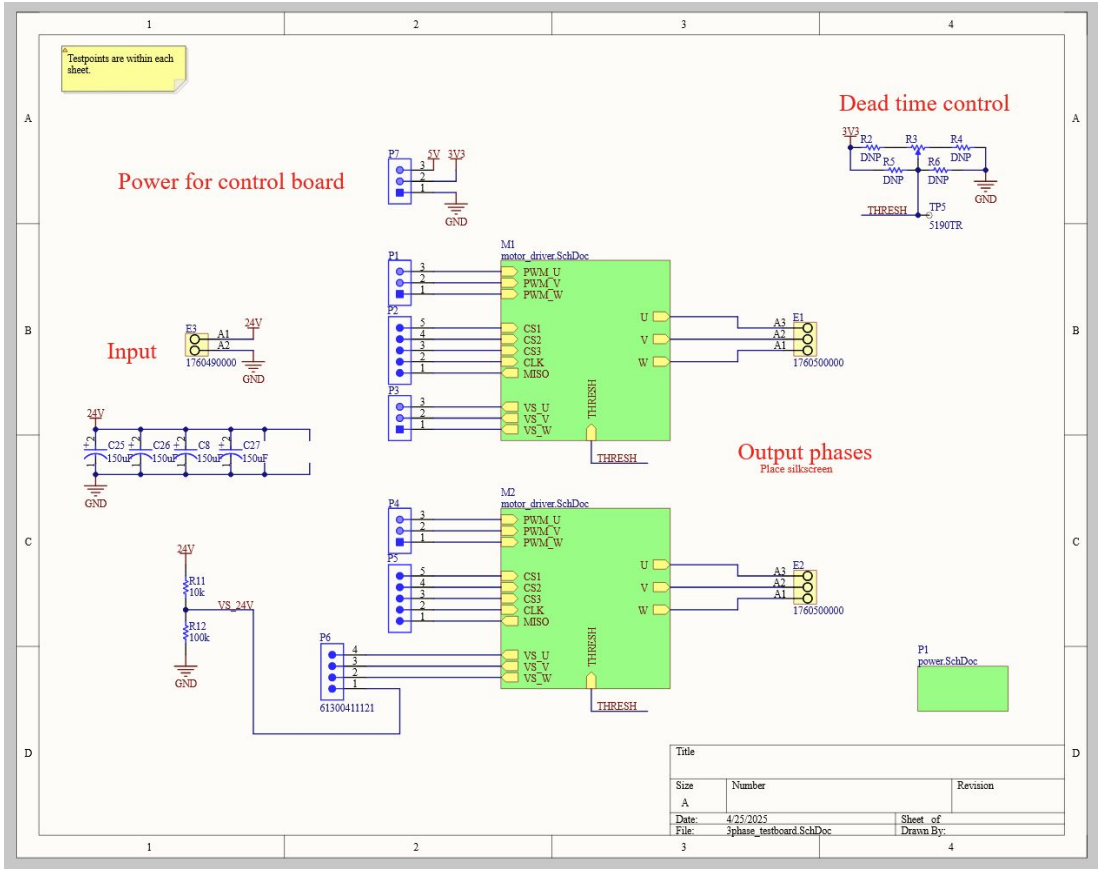
Tests - Driving current and high-side pulsing

- Simple test I did a lot for driving current through an individual phase:
  - Connect one side of a power resistor to phase, other side to PSU 24V or GND
  - Breadboard 2-button setup – pull-up / pull-down, connect either one to PWM input
    - 100% duty cycle or 0% duty cycle, so this was coarse testing
- Used to validate several things
  - How much current can GaN half-bridge deliver? How hot does it get?
  - Current sensing – analog portion and SPI ADC
- Observations
  - Low side could sink current continuously, no appreciable heat
  - High side could not – would pulse in a weird way, see purple
    - Would go back to 24V when it reached some voltage??
    - Driving DC motor: more load increased pulsing frequency



Design

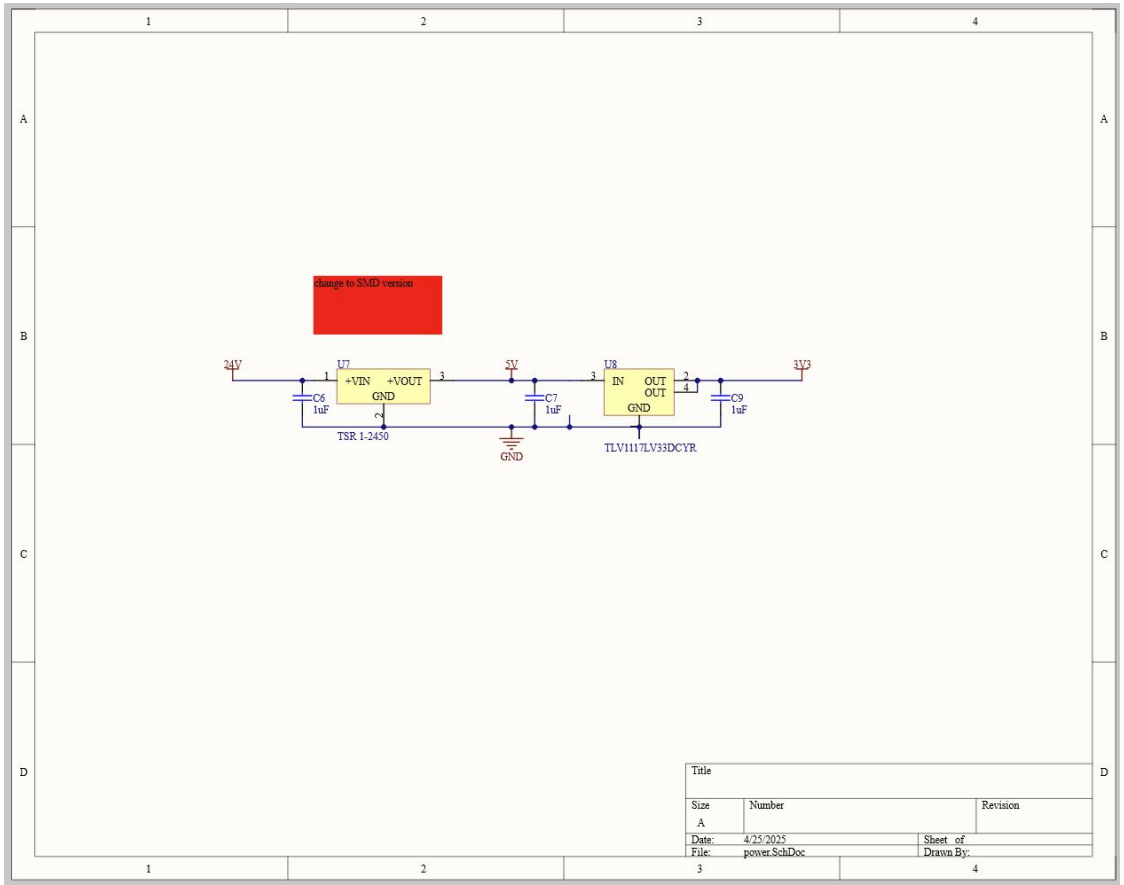
Schematics - Top sheet



Schematics - Top sheet cont.

- Shows **entire design** using hierarchical design structure
  - Each green block represents one instance of a schematic
  - Instead of explicitly laying out three phases twice, I have two motor sheets
  - Same thing for phases: Each phase is identical, so each motor has three phases
  - Not sure if Altium likes this structure, but connectivity worked, which is all you really need
  - Any schematic in the project that is not on this sheet as a green block is not included
    - Learned this with the power sheet (SE corner); has no ports, but still must show up here
- Power and connectors on this sheet
  - Shows which circuits appear only once
  - Nice to see high-level connectivity

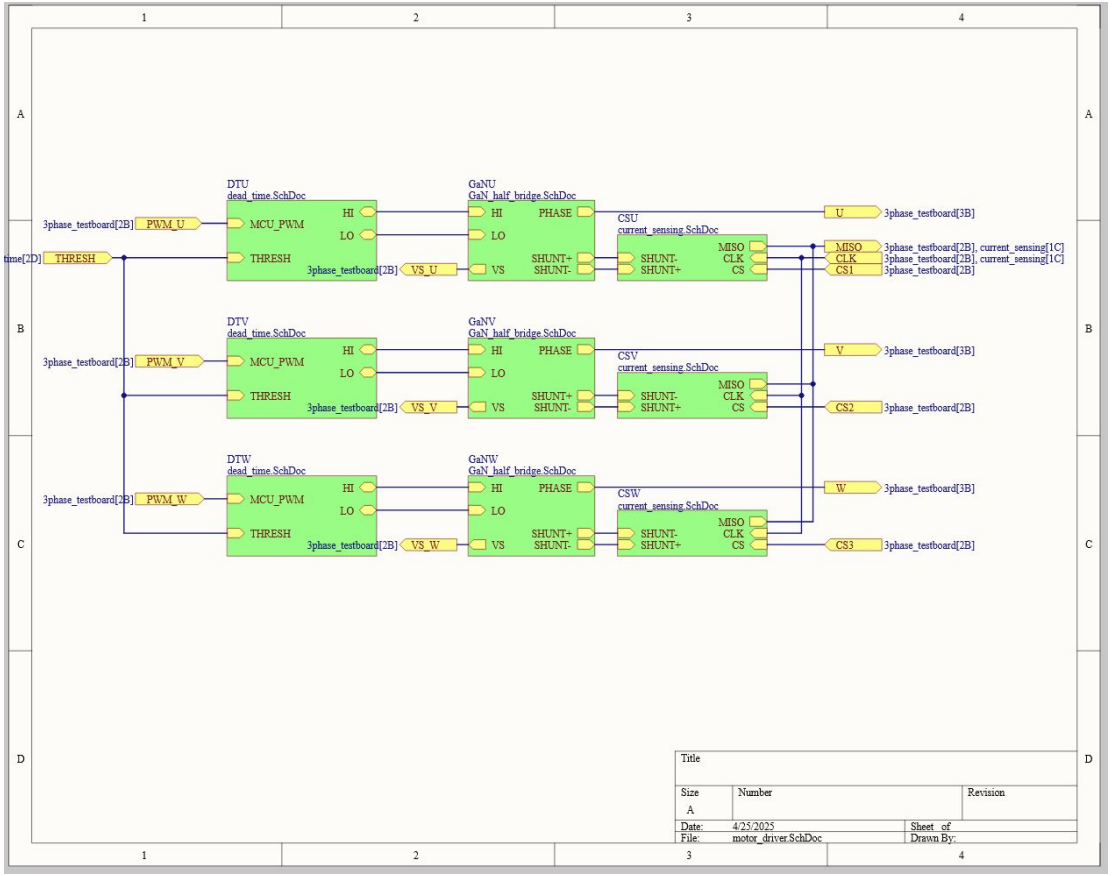
Schematics - Power



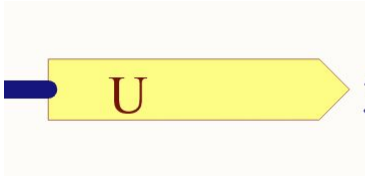
Schematics - Power cont.

- Aimed for simplicity here
  - DC-DC is a complete module, no external components
  - LDO is easy too
- Just follow datasheets!

Schematics - Motor



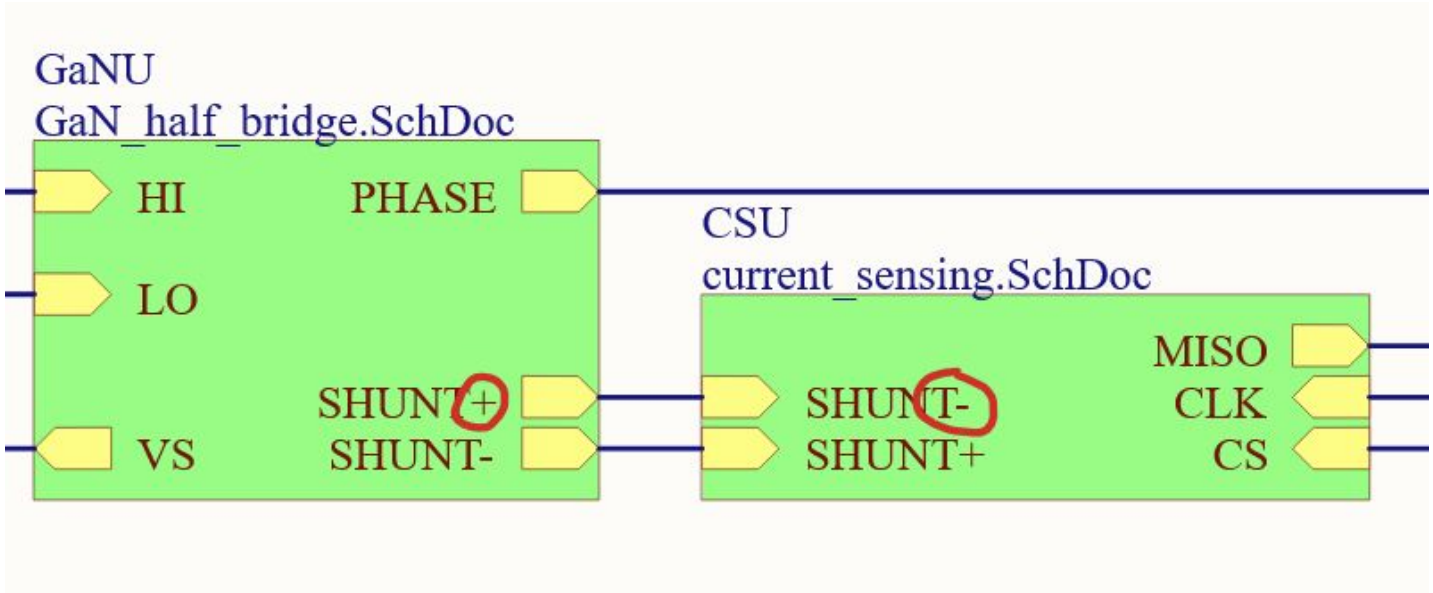
Schematics - Motor cont.



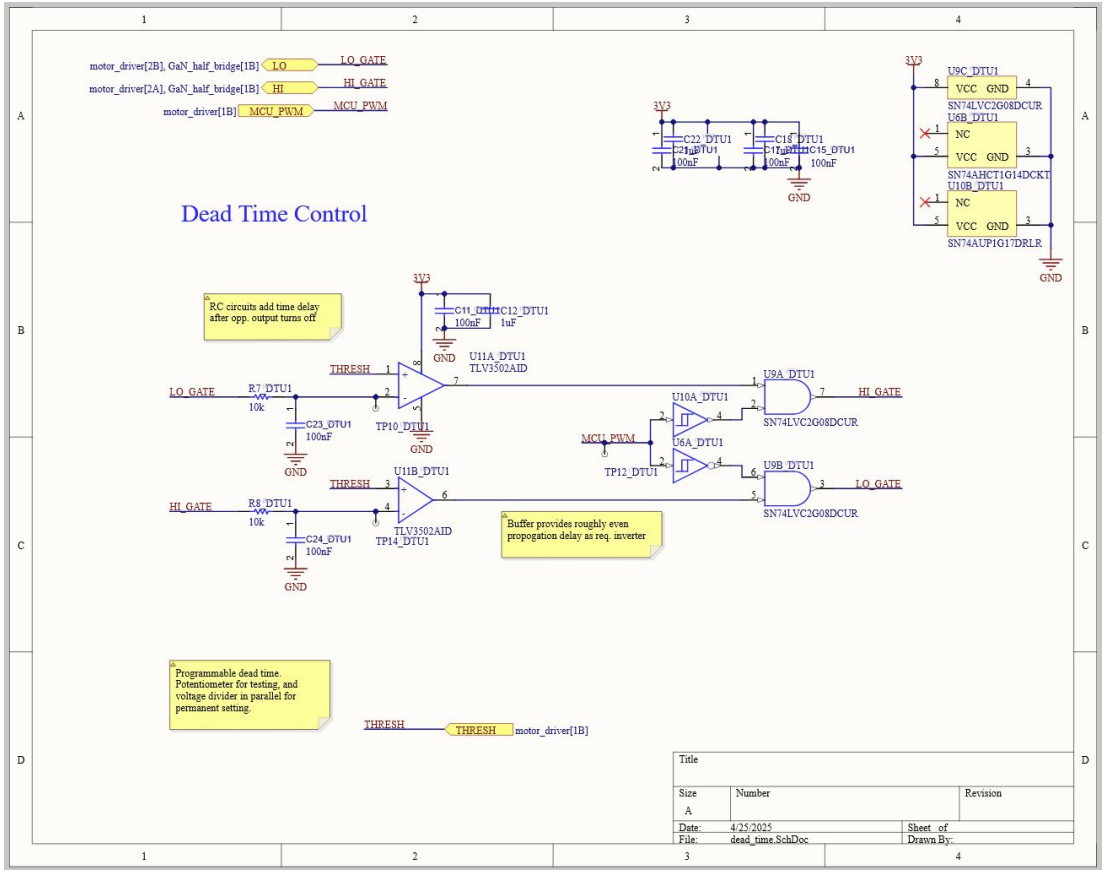
- Note use of hierarchical design again
  - Each row is a phase
  - Speeds up design time, less error-prone
  - Each "port" (shown right) corresponds with a port on the green block in higher-level sheet
- Inputs:
  - PWM, for motor control
  - Threshold, for dead time control
- Outputs:
  - Phase, actual power connection to motor
  - SPI communication for current sensing
  - Voltage sensing

Schematics - Motor cont.

- Oops, just saw this mismatch
- But it's okay – current sensing is bidirectional anyway



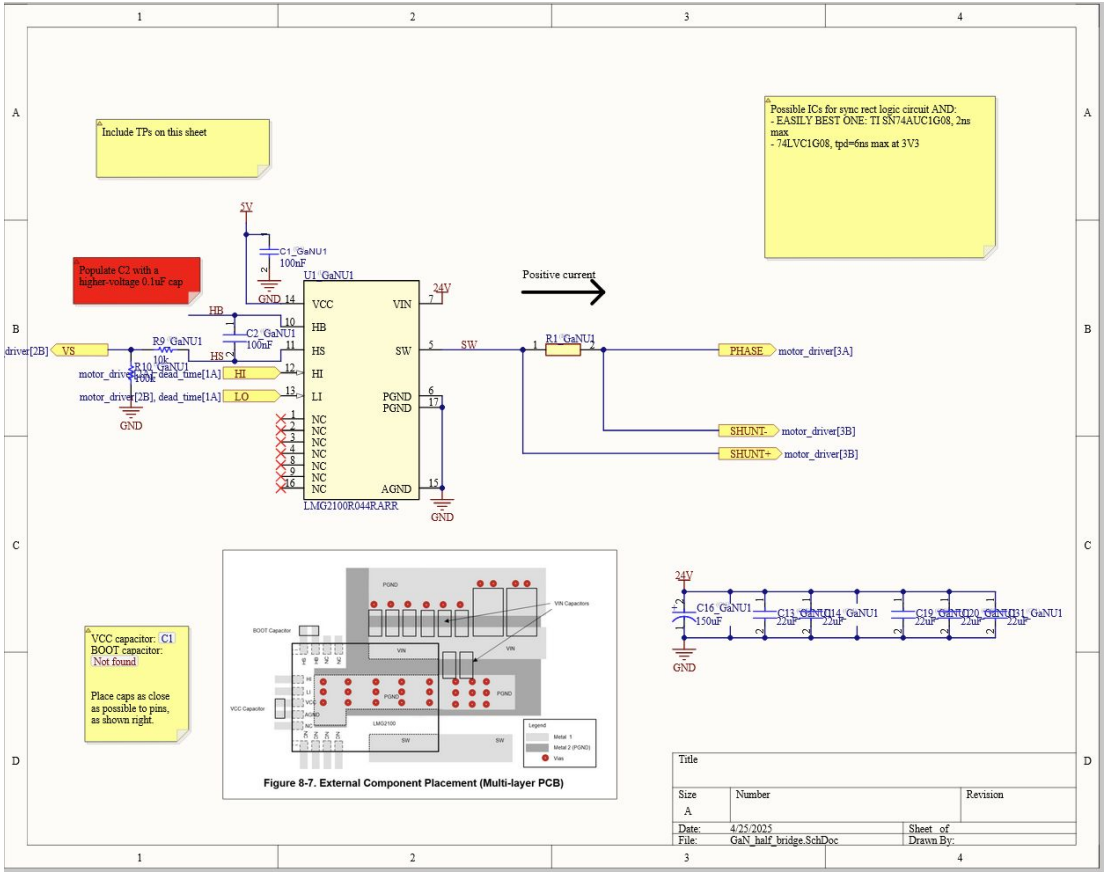
Schematics - Dead time control



Schematics - Dead time control cont.

- Design already explained, so this slide focuses on details of implementation
- Visually "busy" components placed in top right
  - Power blocks for ICs
  - Decoupling caps for ICs
    - Just keep track of how many you need – usually one 0.1u, 1u pair per IC
    - Also refer to typical implementation on datasheet
- Buffer used on PWM input to match propagation delay (pd) of inverter
  - Inverter is required for logic to work, but buffer is not – only there for the above reason
- R7/R8 and C23/C24 make up the (kind of) tunable RC
- MUCH time went into picking these specific components, esp. comparator
  - Minimize pd, small packages that can fit closely with each other

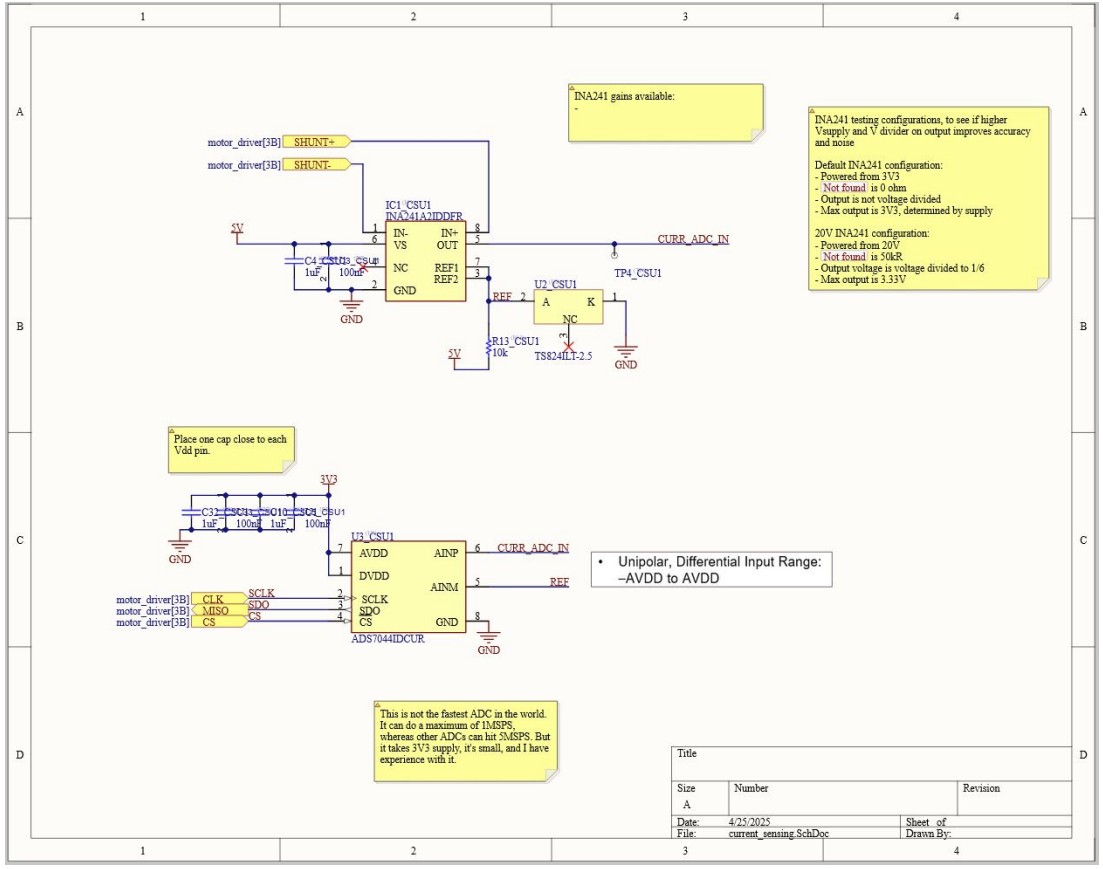
Schematics - GaN half-bridge



Schematics - GaN half-bridge cont.

- Pretty simple to implement, just go off chip datasheet
  - Was deliberate about component choice for C1, C2
    - Cared mostly about value, 0402 package; cared a bit about tolerance and temp rating
- Shunt
  - Choice is tied in with rest of current sensing calcs, see Current sensing schematic
  - Wide package allows for wider traces
- Number of bulk capacitors determined by layout
  - Also smaller caps recommended by datasheet shown here
  - Really should've done all bulk caps on top sheet, and only datasheet-specified on this sheet

Schematics - Current sensing



Schematics - Current sensing cont.

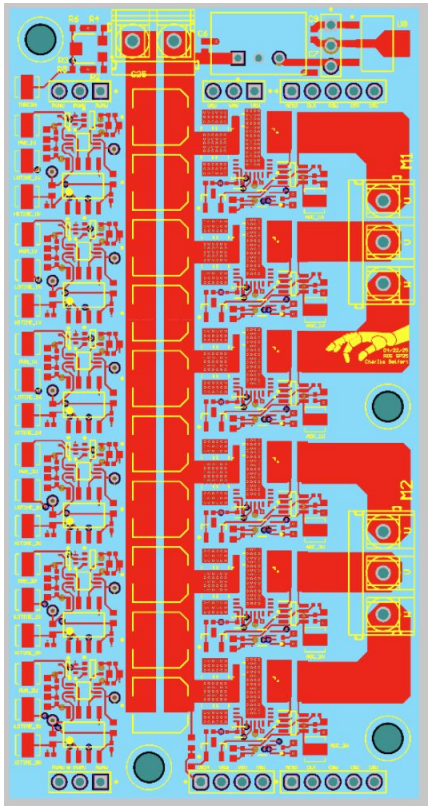
- INA241 used because it's very popular, really good (CMRR), and cheap
  - Flexible Vsupply = 2.7V to 20V
- REF inputs
  - Set zero point of output; This portion of datasheet shows a few ways of doing it
  - Ivor had previously tied this to power rail and GND – but that's unnecessarily noisy
  - Instead, use a precision reference (a zener diode) tied to both REF pins
  - Original plan: adjustable reference up to 10V, with Vsupply switchable from 3V3 to 20V
    - Test if higher resolution was actually helpful
    - Unsure if this would be helpful
- Gain chosen with Motor driver design spreadsheet

Schematics - Current sensing cont.

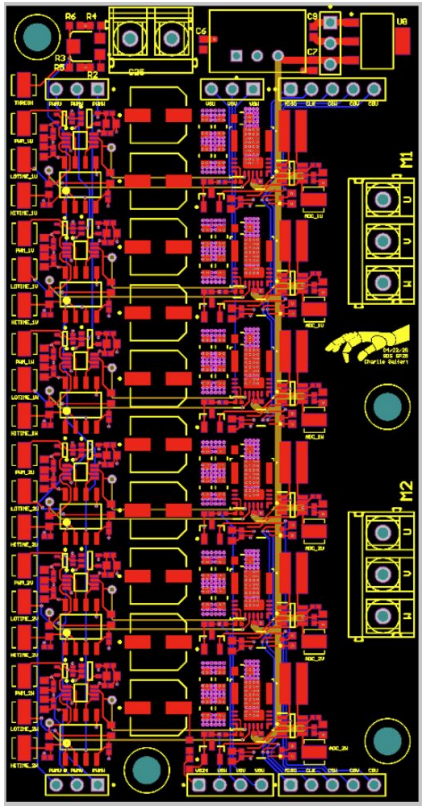
- ADC choice
  - LOT of research
  - There are better ADCs
    - This one is 1MSPS, there are 4+MSPS
  - But I had experience with this one, and code that worked
  - Easy enough to design around, especially compared to others (supply voltage, package, etc.)
- Note use of REF as AINM ("analog in minus")
  - ADC reads 0V differential when no current, even though INA241 outputting 2.5V
  - Maximizes use of ADC's voltage input range

Layout - Full view

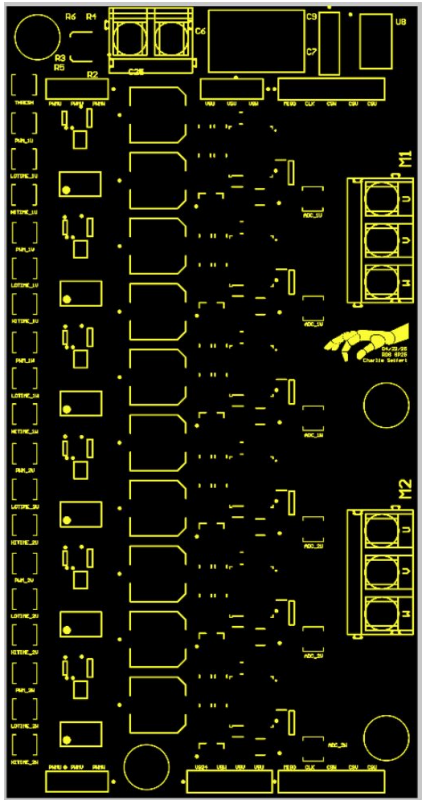
full



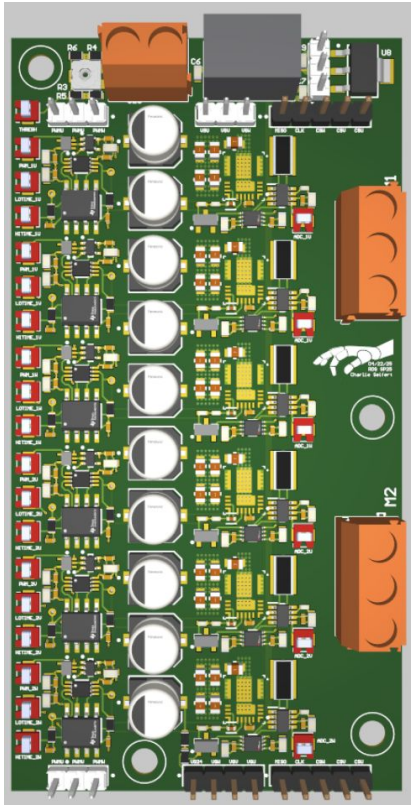
polygons shelved



top silkscreen

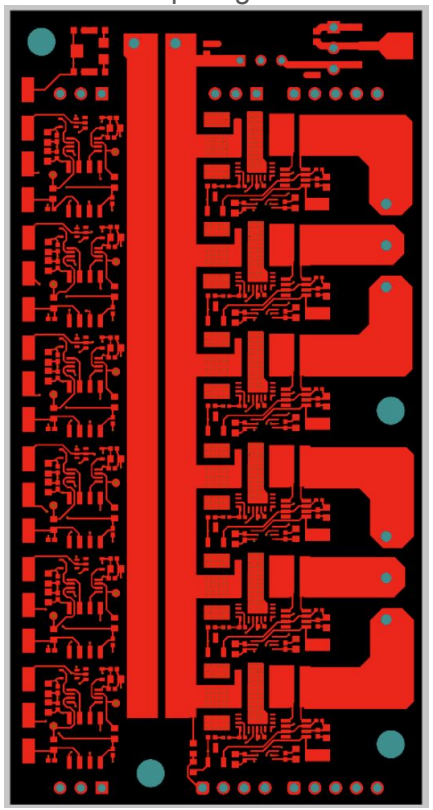


3D

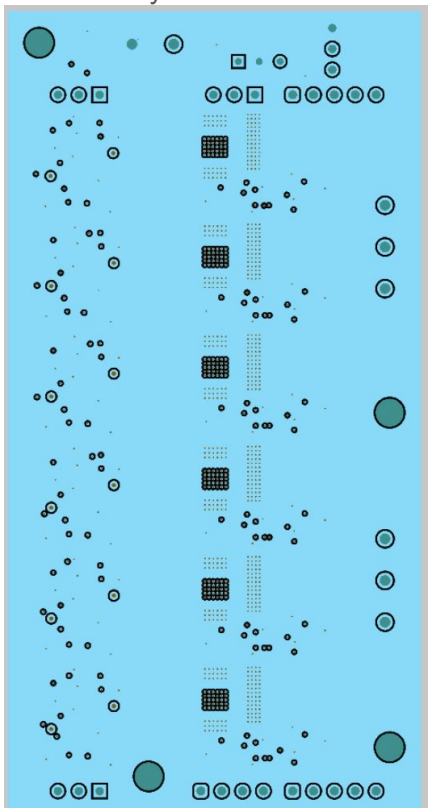


Layout - Layers

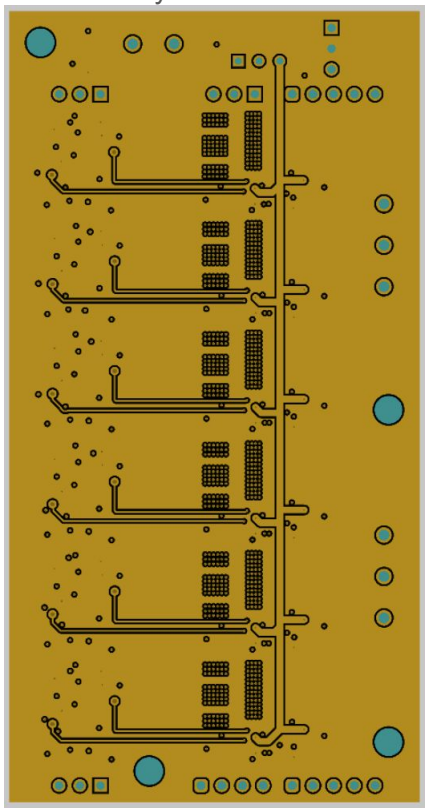
top - signal



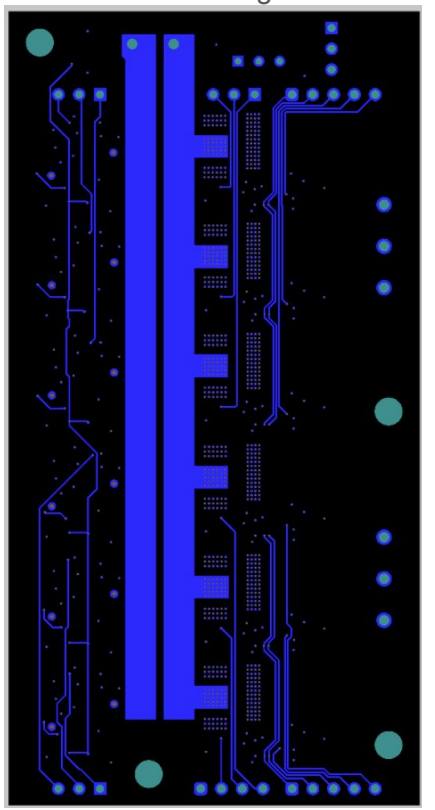
layer 2 - GND



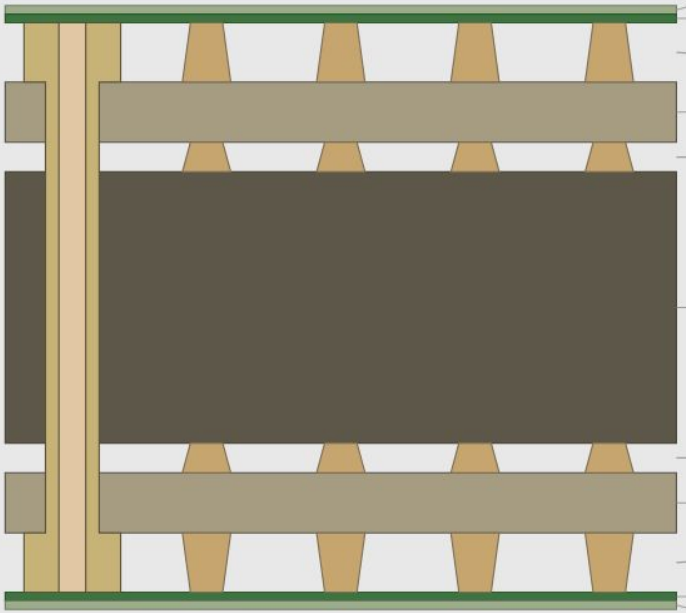
layer 3 - 3V3



bottom - signal



Layout - Layer stack



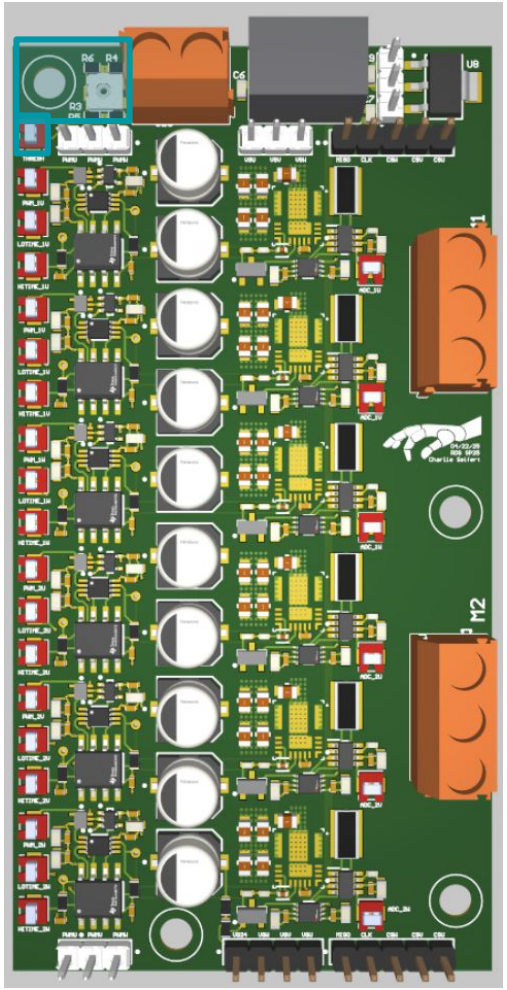
	Layer	Type	Thk (mm)
-	Top Overlay	Overlay	
-	Top Solder	Solder Mask	0.0102
1	Top Layer	Signal	0.0700
-	Dielectric 2	Prepreg	0.0711
2	Layer 2	Signal	0.0350
-	Dielectric 1	Dielectric	0.3200
3	Layer 3	Signal	0.0350
-	Dielectric 3	Prepreg	0.0711
4	Bottom Layer	Signal	0.0700
-	Bottom Solder	Solder Mask	0.0102
-	Bottom Overlay	Overlay	
		Total	0.6926

Layout - Layer stack cont.

- 4-layer board
  - Was a little difficult to fit but doable without much compromise
  - Still cheap from JLCPCB
- 2oz top and bottom layers
  - Power pours lived here
  - Made it take forever to melt solder, but oven solves that hassle
- 1oz on inner layers sufficient
  - Don't want high power on inner layers anyway – less heat dissipation so less current capacity
  - Used for logic power, a few straggling signals, and a large ground pour

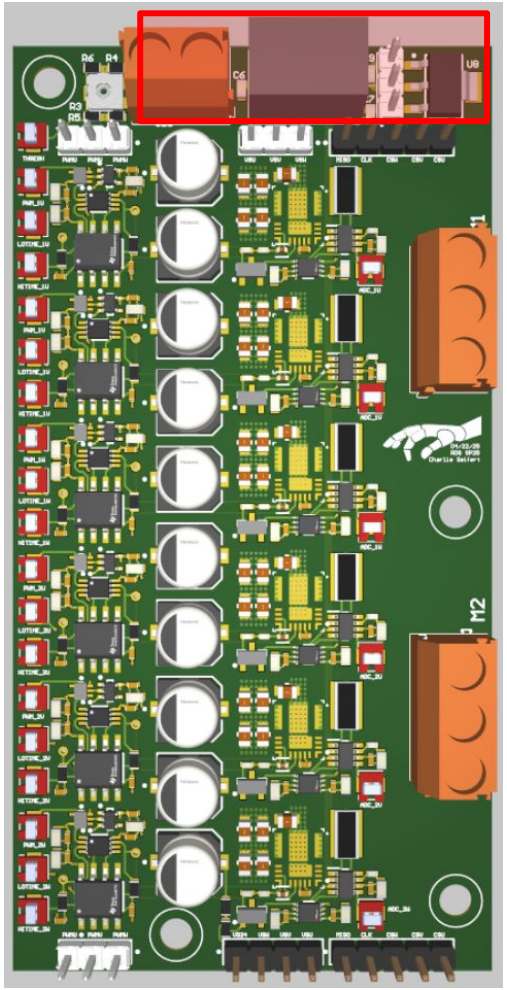
Layout - Regions

- Dead time threshold



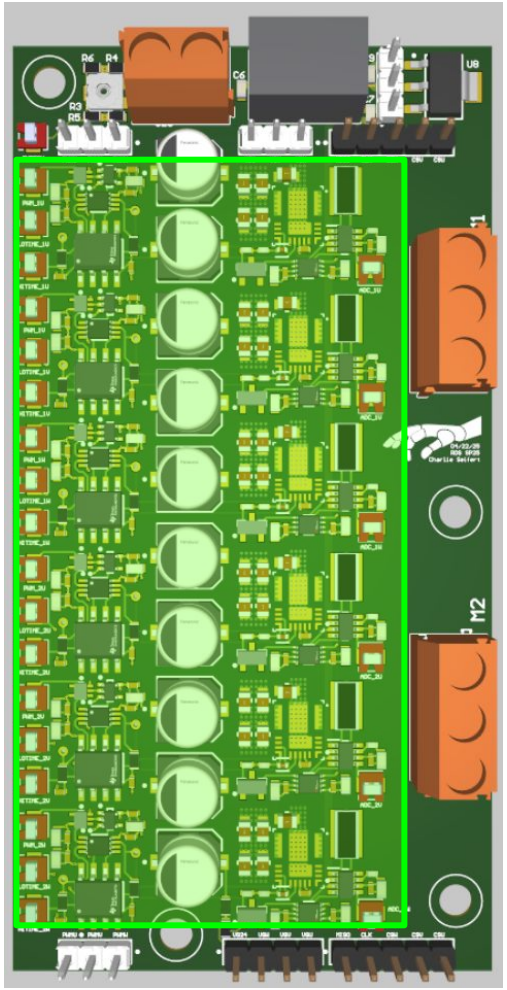
Layout - Regions

Power input, regulation



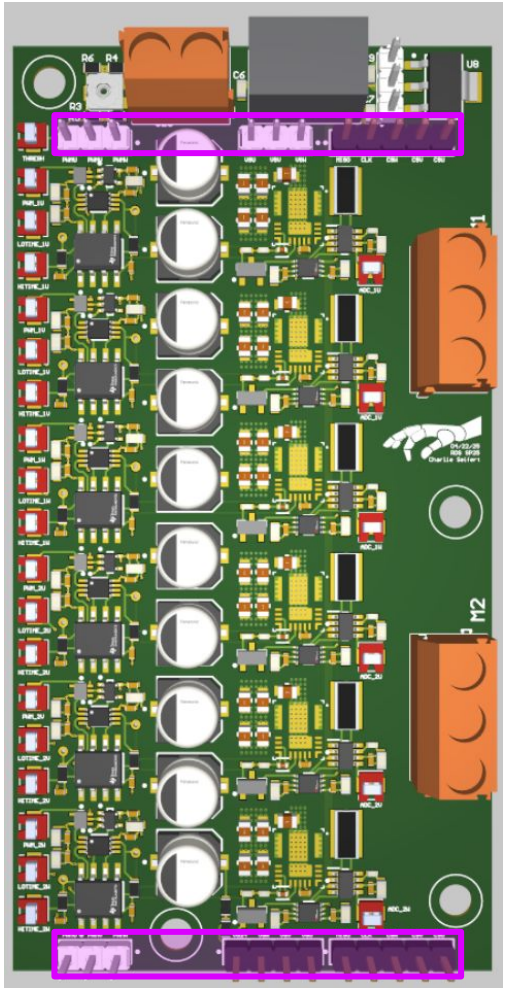
Layout - Regions

Phases (6)



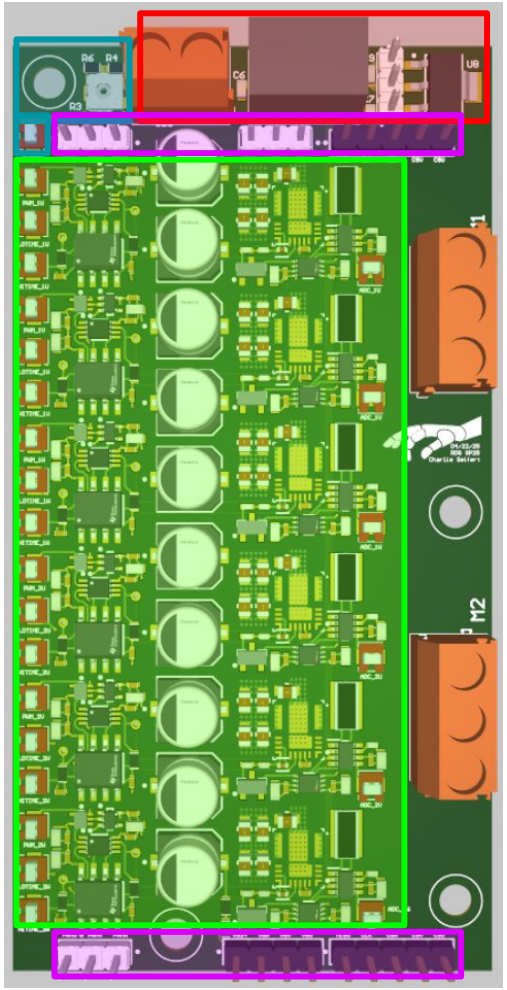
Layout - Regions

- Connectors for PWM, Vsense, SPI

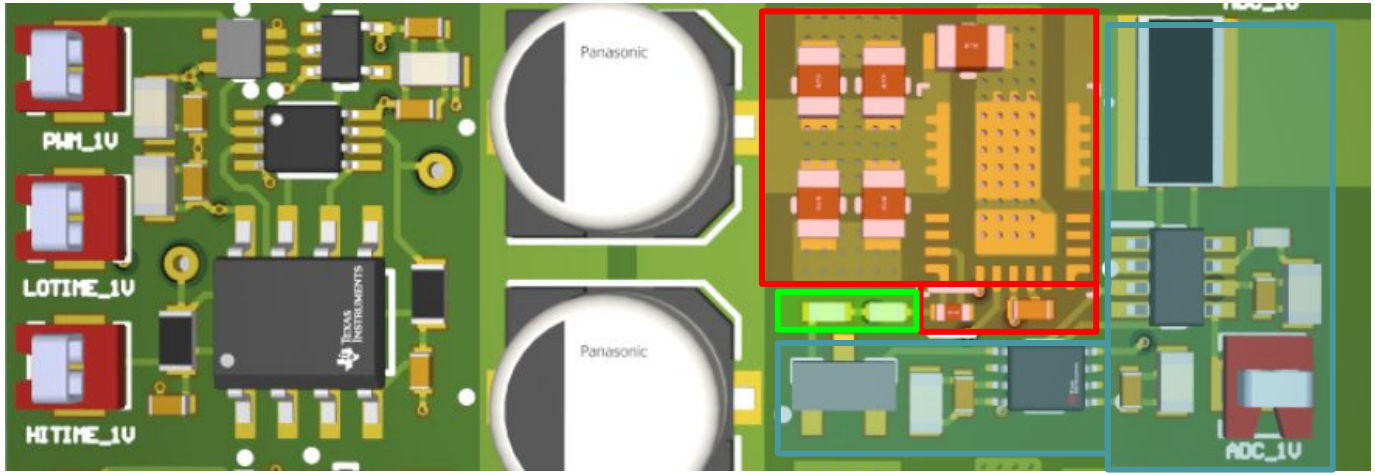


Layout - Regions

- Dead time threshold
- Power input, regulation
- Phases (6)
- Connectors for PWM, Vsense, SPI



Layout - Regions of phase



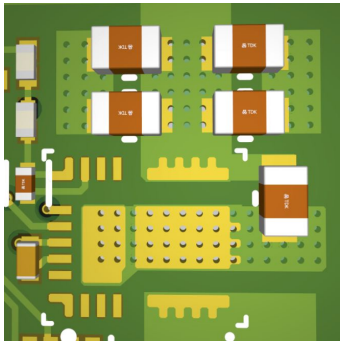
Dead time control

Shunt and current sense circuitry

GaN half-bridge + caps

Voltage sensing (a R divider)

Layout - General thoughts



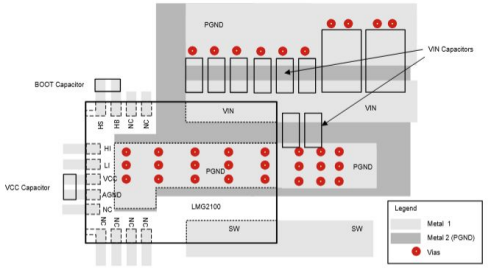


Figure 8-7. External Component Placement (Multi-layer PCB)

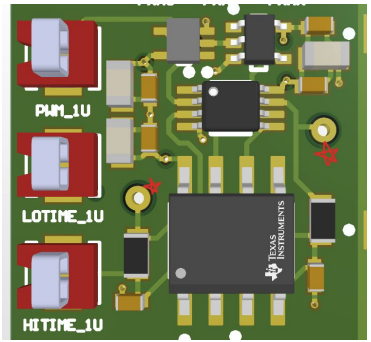
- Components only on top
  - Simplifies assembly
- Lots of vias for power connections
  - Good solid connection to ground plane for heat dissipation
- Decoupling caps close to supply pins for ICs
- Follow datasheets for layout guidelines as much as possible
  - Be deliberate when you deviate – be able to explain why
- One way I deviated from datasheet: Placement of GaN half-bridge caps
  - See above right; I wanted a more square and tileable layout, so bent the pours and deleted some caps; assumed probably-excess bulk capacitance would be sufficient
- Try to keep sensitive signal lines from traveling parallel to high-current paths
  - If you have to cross, cross perpendicularly

Layout - General thoughts cont.

- SPI routed on the bottom layer
- Separated connectors per motor: top row controls M1, bottom row M2
  - Wanted SPI lines to not be so long, and allow multiple buses to control in case 6 devices on one bus was not desired
- SPI shared pins (MISO, CLK) are not connected between two header rows
  - In our case, daughterboard connected the same SPI bus to both because of pin budget, but that does not have to be the case
- Grouped the connectors so that misalignment when soldering (which is super easy) would not make it impossible to install the stacked daughterboard
  - Would be worse if you had a bunch of independent headers, like right next to each ADC
  - Ended up being really easy to install!

Layout - General thoughts cont.

- Made testpoints accessible and readable
  - Actually wanted a few more, but decided to delete for space
    - In the case of HI and LO inputs for each GaN half-bridge, I put in big untented vias
  - Labels were too small (15mil) to see (esp. U vs. V); increase silkscreen text size (~20mil)
  - Current sensing line would've been stupid to pull out to the edge (sensitive analog)
- Large untented vias for HI and LO inputs (see right)
  - Actually ended up being a fine way to get a scope probe on them
- 3V3 plane ended up routing 5V; HI and LO inputs
  - Didn't care much about cutting up this power plane; had solid GND pour



Reflection

- When in doubt, apply less solder paste
  - Especially for dense designs
- Don't use lead-free solder
  - I have no clue if this is what caused the problem
  - But it adds uncertainty for basically no benefit
- Buy female headers of certain lengths
  - Don't cut short rows off a longer "stock" piece
  - Can do that for male headers, but cut females just look bad and wasteful to cut (you lose one)
- Question packaging requirements
  - Would've been nice to decide we wouldn't mount electronics on forearm
  - It would be far more advantageous to **use more space and make not so dense**
    - Bringup, debugging SO much easier

Reflection cont.

- Make it easier to measure HI and LO inputs closer to the chip
  - Problem was noise on the input, but that was a good inch from the chip – had to be really careful when probing on the chip's pins, not very safe
  - But large untented vias are really not a terrible way to break out a signal!
- Add decoupling to HI and LO inputs
  - Have not tested, but very suspicious of this causing the pulsing problem… or something else in the dead time control circuit

Reflection on RDS

- Difficult project with interesting problems
  - A bunch of people learned a lot of things, both technical and how to work on a team
- Structure with testbench in winter worked well
  - If rushed, it served us well this quarter, especially dealing with integration problems on a smaller scale
- Lack of solid, visible timeline = bad
  - Instead, work with teams individually to make their timelines
  - Everything took too long
- **Take inspiration from Ilya's structure of EE326**
  - Fully functional prototype by week 5, then spend the rest of the time making it better
  - Yes, that means you will compromise design – that is the point, *simplify* and you will learn more after you make the thing

Reflection on RDS

- This class desperately needed high-level direction
  - Lack of centralized leadership causes fragmented design processes, changing requirements, and pushed deadlines
  - Time is obviously an issue, so would've been best if project scope was defined better before the quarter started
  - **Undirected effort is wasted effort** – and direction must come from above
- Not sure what the answer is there, but there needs to be someone on the hook if the project flops
  - Otherwise everyone says "not my problem" and no urgency

Contact

I am glad to answer any questions on the board or the decisions I made in developing it. You can reach me at e36@duck.com.