

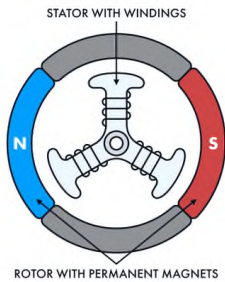
Gardena R50Li Robotic lawnmower fault

den 5 juni 2025

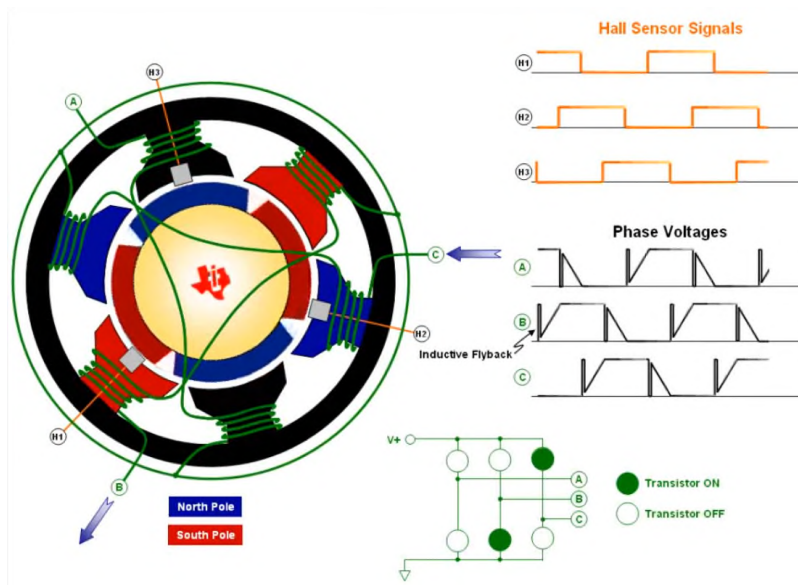
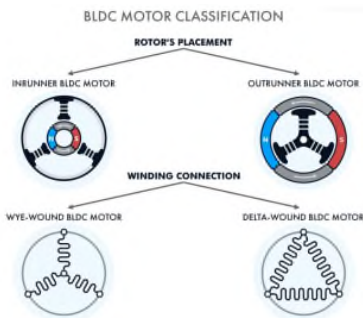
Specification of problem:

The symptom is as follows: Only one wheel motor is turning, the left one. A test was made to shift the left and right wheel motors by changing the connection cables. The problem then moved to the left motor. The conclusion is that there is no problem with the wheel motor and that the problem is related to the main PCB. To be able to reach the test menu (to test wheel motors etc.) enter 7 & 9 concurrently in the main menu. The test menu will now pop up. The unit is a Gardena R50Li Robotic lawnmower.

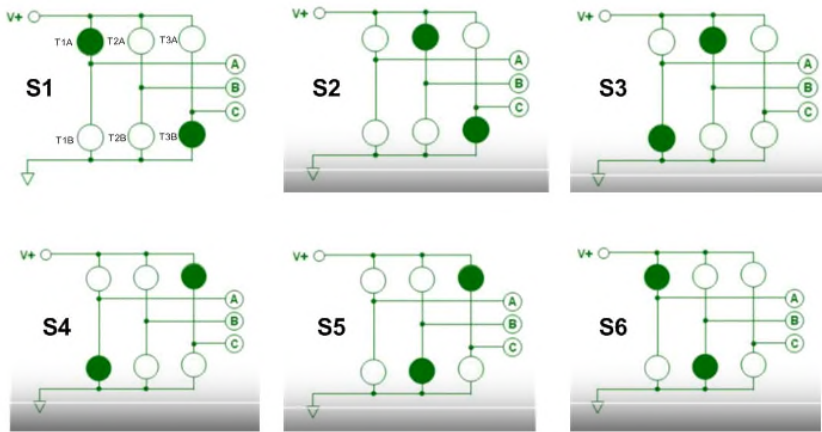
General motor layout/BLDC Brush less motor overview:



Classifications of the motor:

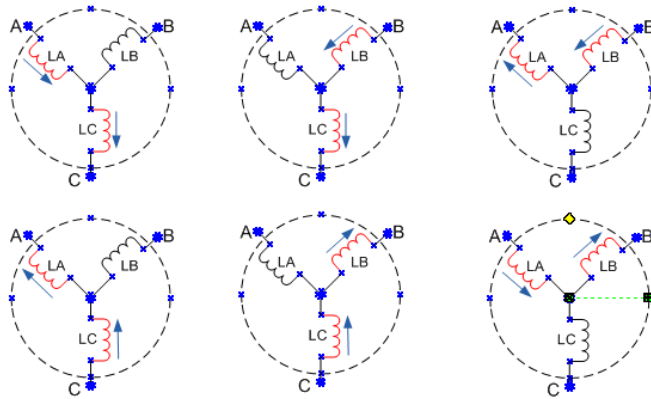


<https://www.youtube.com/watch?v=43JMIuwVrY4>

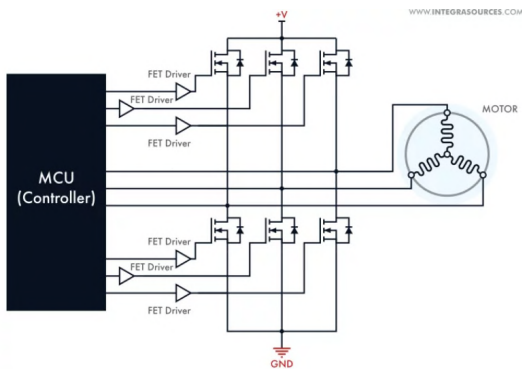


Communication sequence:

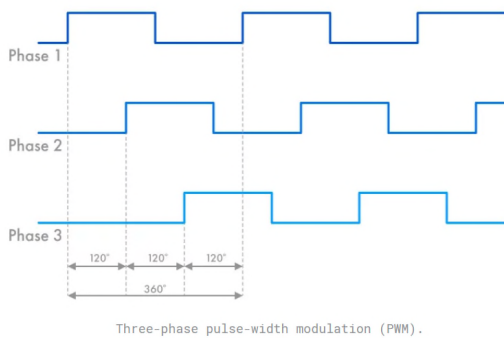
SEQ	T1A	T1B	T2A	T2B	T3A	T3B
S1	1	0	0	0	0	1
S2	0	0	1	0	0	1
S3	0	1	1	0	0	0
S4	0	1	0	0	1	0
S5	0	0	0	1	1	0
S6	1	0	0	1	0	0



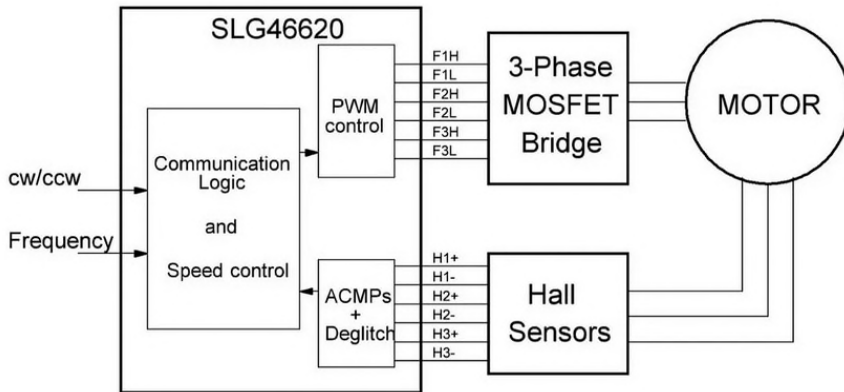
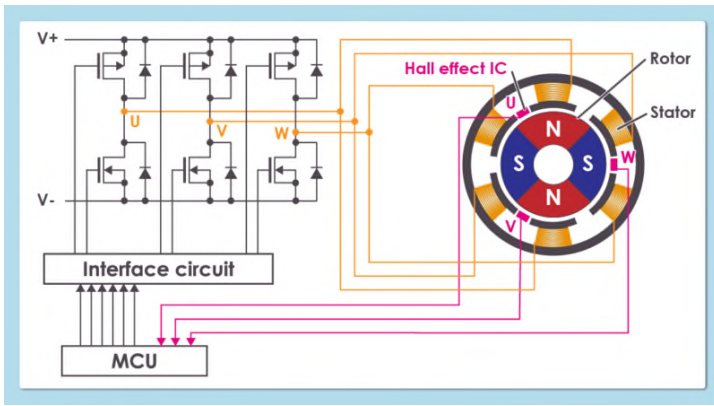
Typical BLDC connection schematics:



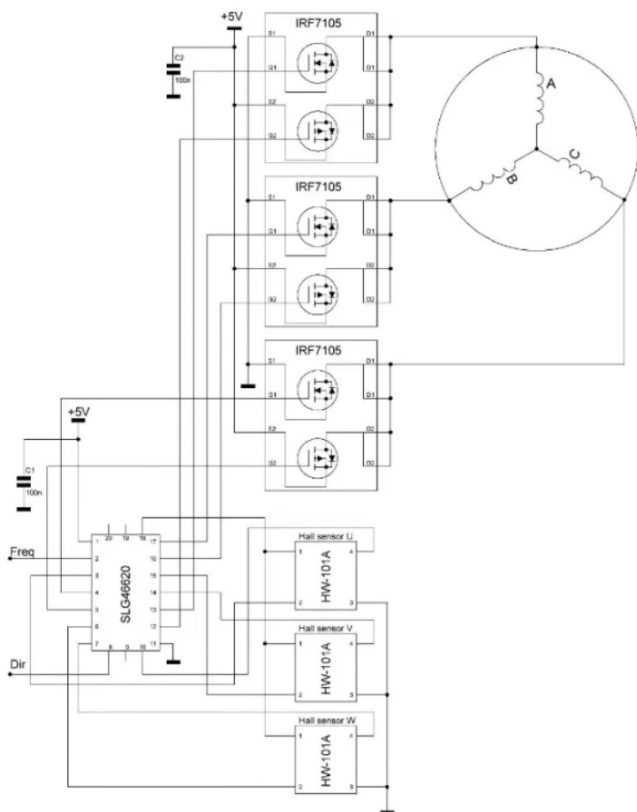
The three phase current to the motor windings phases A, B, C

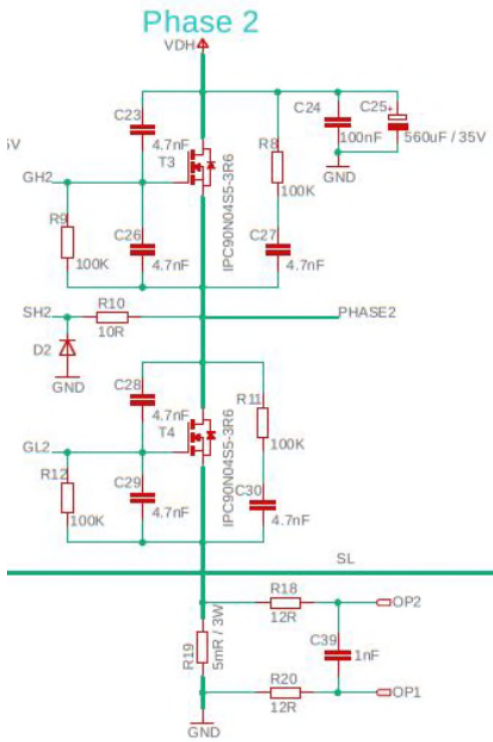


Here is a driver with Dual N & P MOSFET:

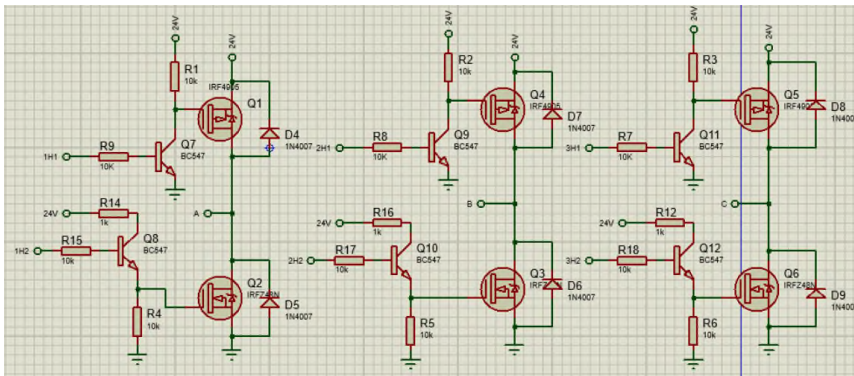


Some schematic examples of one leg H-bridge for BLDC motor (3 Phase):

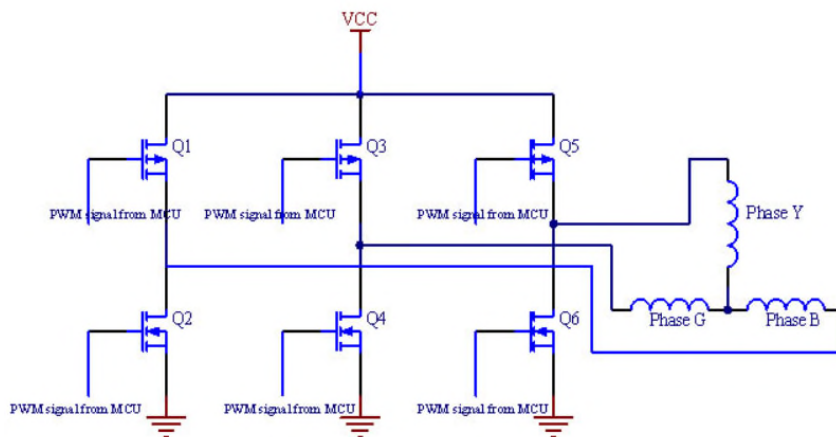




https://www.mouser.se/datasheet/2/196/Infineon_BLDC_shield_UserManual_v01_02_EN-3132092.pdf



<https://i.sstatic.net/oULKK.png>

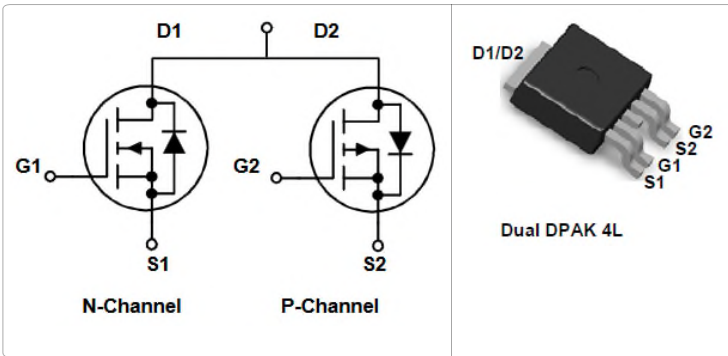


<https://electronics.stackexchange.com/questions/347983/bldc-commutation-and-output-states-of-half-bridges>

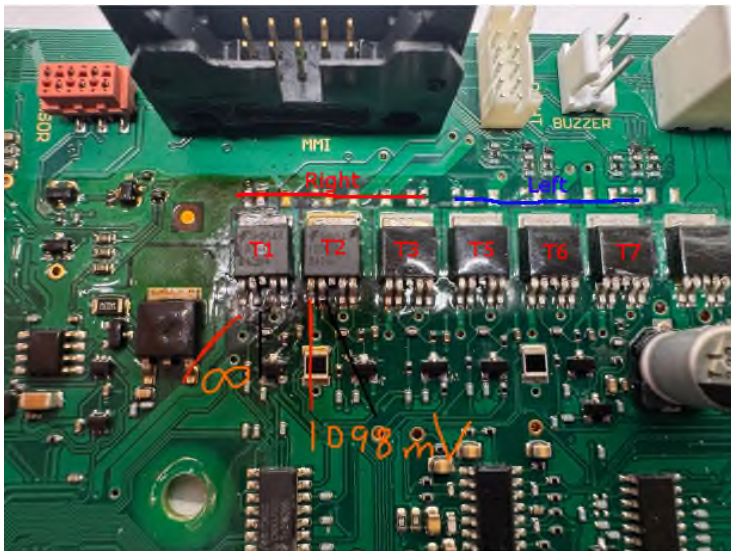
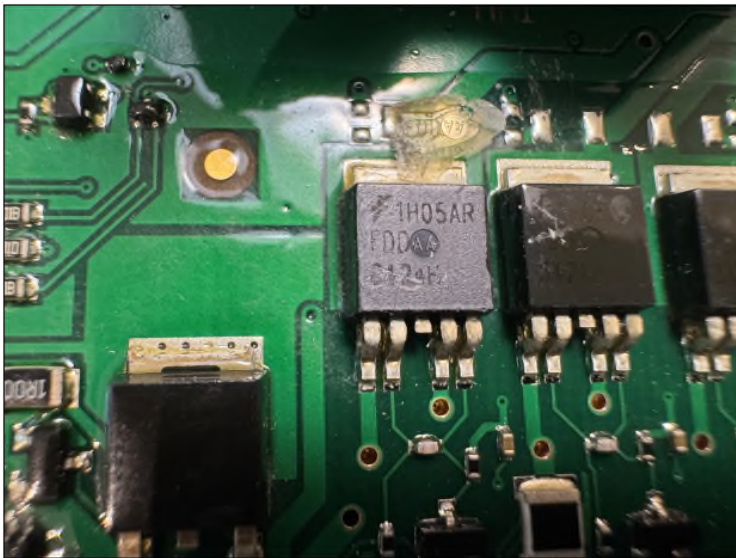
Specific for the Gardena R50Li

Here is specific data for the Gardena R50Li Robotic Lawnmower.

It uses FDD8424H Dual N & P-Channel MOSFET BLDC motor driver that looks like this:



Basic electronic AC DC Fan #Mosfet #Transistor testing, this video shows how to test a FDD8424H with a multimeter. Link: https://www.youtube.com/watch?v=lzgM_D7jj1M



The wheel motor (577 83 99-06 - Hjulmotor komplett) look like this:



Pasted from <<https://translate.google.se/?hl=sv&sl=sv&tl=en&text=Klipparen%20som%20Jacob%20Wigvisthar%20C3%A4r%20en%3A&op=translate>>

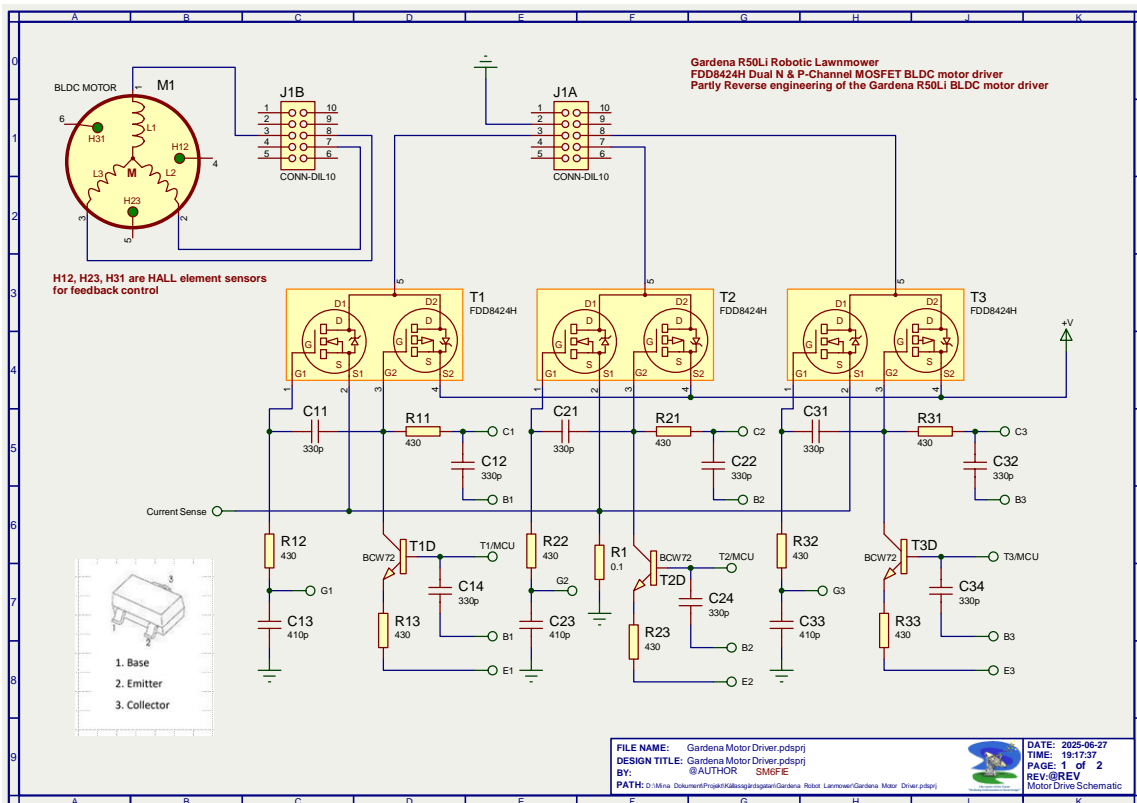
Connectors for motor is probably:

ZH		1.50 mm (0.059 in)	1	1	50	32 to 26	Yes	No	Compatible with ZR (IDC) and ZM (crimp) series.
-----------	---	-----------------------	---	---	----	----------	-----	----	---

Partly Reverse engineering of the Gardena R50Li BLDC motor driver

Belo is a partial schematic of the BLDC driver circuit of the device. This is for just one motor. The circuit is identical for both heel motors, left and right.

T1 is the left most MOSFET on the PCB hen you look at it in the standard orientation. H12, H23, H31 are HALL element sensors but I have not included that circuit in the schematic.



The unit uses a STM32F215Z6T6 micro processor (MCU) as the central brain. It uses a 144 pin LQFP package. Connection to the motor driver circuit is done through pins in the interval 37 to 72. I have not identified the exact pin numbers. The MCU operates at a voltage level of 3.3 volts.

Troubleshooting

I found that T1 (the most left one) was shorted between source and drain as well as the gate. This short persisted as well when the MOSFET was removed for the PCB. T2 and T3 was also removed from the PCB and was found to be ok and working. After the this the PCB was cleaned up with isopropanol alcoholic and new (virgin) FDD8424H MOSFETS was soldered in place on the PCB.

An extensive test of the driving circuit of the T1 MOSFET was done to find any fault in the driving circuit but none was found, everything seemed to be ok with the T1 driving circuit.

Thereafter the PCB was mounted back in the lawnmower and the device was powered on. Thereafter we activated the test menu by enter 7 & 9 at the same time. Thereafter the motor test was selected. Here is the test result:

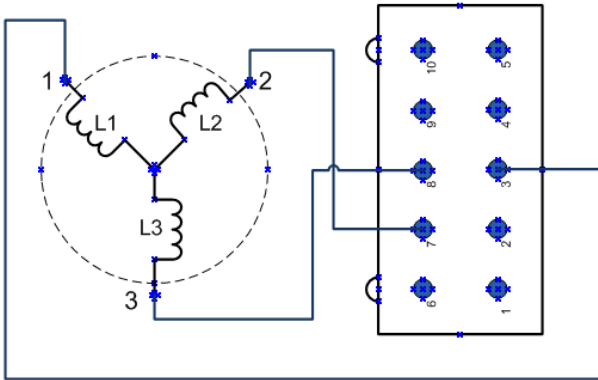
Motor	Connected to connector	Note
Left	Left connector	Runs ok
Left	Right connector	Hacks and does not run well

Right	Left connector	Runs ok
Right	Right connector	Hacks and does not run well

The conclusion is that the right motor is OK, it must be some fault with the gate driving of the right motor or eventually the feedback sensors. I thing is using hall effects sensors.

I will make an effort to measure the windings of the motor to see if I can found any deviation from normal. However Jacob Wiqvist brought up to my attention another possibility and that is that the gearbox of the motor can be faulty. According to Jacob he have seen this problem discussed on forums as a common motor problem.

Here is a measurement of the right motor, se figure below:



Path	Inductance	Resistance
7-8	2.014 mH	3.83 Ohm
8-3	2.061 mH	3.886 Ohm
3-7	2.052 mH	3.826 Ohm

All values seems reasonable so it is probably not any fault with the windings of the motor.

Chat with ChatGPT:

I have a Gardena 50i Robotic Lawnmower. The right wheel driver motor does not work correctly. The left works ok. The lawnmower uses a BLDC motors. The motor driver uses MOSFET FDD8424H. I noted a short between the Drain and the source of one of the MOSFET's, this is MOSFET # T1. The BLDC motor is a three phase BLDC motor. After i replace the T1 MOSFET and test it with the left motor driver it works correctly when the motor is connected to the left driver . The left motor is therefore obviously correct. Now if I connect the left motor that is ok to the right driver it malfunction. It rotates uneven and choppy. After this i find that the T1 MOSFET once again is shorted. Therefore it must be some fundamentally wrong with the right motor driver. I have checked the circuit that drives the T1 MOSFET but can't find any wrong. The signals to the motor driver is generated by a 32 bit micro processor. The motor has a feedback signal, probably using Hall effect sensors. I cant find any error here but I can be mistaken about this, Any ideas about what the fault can be?

Thank you for the detailed description and the image (which confirms the part number: **FDD8424H**, an N-channel MOSFET).

Summary of Your Situation

You have a **Gardena 50i robotic lawnmower** with **three-phase BLDC motors** driven by **MOSFET half-bridges**. The **right wheel driver** repeatedly **destroys the low-side MOSFET (T1, FDD8424H)**.

- The **left motor and left driver work fine**.
- The **right motor itself is also fine** (as tested using the left driver).
- Replacing **T1** temporarily restores operation, but it fails again after test run.
- The **control signals to the gate** of T1 *look normal*.
- Feedback from Hall sensors appears ok, but could be misleading.

Q Most Probable Root Causes

Here are the most likely causes, prioritized:

☐ 1. Gate Drive Timing Problem (Shoot-through)

- If the **high-side and low-side MOSFETs** on **one phase leg** (e.g., T1 and its high-side counterpart) are **on at the same time even briefly**, this causes **direct short from supply to ground** — instantly killing the MOSFET.
- This is the **most common cause** of repeated MOSFET destruction in BLDC drivers.

Even if gate signals look OK on a slow scope, nanosecond overlap (due to gate charge delay mismatch or firmware bug) can cause this.

Test:

- Use an oscilloscope with **both gate signals (high-side and low-side of same phase)** on screen with a **fast time base (e.g., 100 ns/div)**.
- Check for overlapping “on” periods.

Solution:

- Ensure **dead-time insertion** between turning off one transistor and turning on the other.
- If dead-time is too short, increase it slightly.

🔧 2. Gate Driver Malfunction or Weak Drive

- If the MOSFET gate isn't driven fully high or fully low, the MOSFET might operate in the linear region instead of switching sharply, causing overheating and failure.

Test:

- Measure the actual gate voltage swing of T1 with respect to its source.
- Confirm it goes close to 0V (off) and above 8V–10V (on).

Solution:

- Check the gate driver IC or discrete gate drive circuitry.
- Check for excessive gate capacitance, leaky driver output, or gate resistor too large.

3. Floating or Damaged Source Return Path

- The low-side MOSFET's source pin is usually tied to power ground.
- If there's a bad solder joint, broken trace, or lifted pad, the gate voltage becomes referenced to an unstable or floating potential, destroying the device.

Test:

- Confirm good continuity from source pin to ground plane.
- Measure voltage drop during switching with a scope.

4. Incorrect Commutation / Hall Sensor Misread

- If the Hall sensor feedback is incorrectly read or misaligned, the MCU might try to commutate in the wrong phase, causing massive current surges, leading to damage.
- You said Hall feedback seems ok, but it's worth validating in-depth.

Test:

- Swap Hall sensors or cables between left and right to see if the fault moves.
- Log the actual Hall state transitions during rotation and verify they match expected phase sequence.

🔧 5. PCB Damage / Solder Bridges

- Check carefully for microscopic solder shorts, burnt pads, or damaged vias/traces near T1 or its gate driver.
- Reflow suspicious solder joints.

🔧 Suggestions for Action

1. Use a differential probe or isolated scope probe to measure gate-source voltage precisely.
2. Capture high-speed gate timing of both high-side and low-side MOSFETs for the faulty phase.
3. Verify dead-time exists and is adequate.
4. Test gate driver IC output resistance and voltage swing.
5. If you suspect commutation, log Hall states and applied PWM phases to reconstruct sequence.
6. Try swapping driver stage PCBs if modular — this can isolate board-specific issues.

📄 Final Word

Given your symptoms — working motor, failing only in one driver, and MOSFET shorts — the most likely cause is timing-related shoot-through or gate drive malfunction. These are subtle and often only visible under an oscilloscope with fast sampling.

If you can share the circuit schematic or a detailed photo of the driver area, I can help pinpoint test points or vulnerable areas.

Would you like a sample test procedure or waveform diagram to help verify dead-time and shoot-through?

Me:

Can you find a similar motor driver circuit using the FDD8424H? Also note that the FDD8424H has both a N and a P channel MOSFET in a dual package. Yes, the source of the MOSFET is connected to ground. Note that there are three FDD8424H MOSFET's for each motor driver (left and right). Any comments?

Thanks for the clarification and the new information — it's quite helpful.

🔍 Key Observations

- FDD8424H is a dual MOSFET in one package:
 - One N-channel
 - One P-channel
- You say there are 3 x FDD8424H per motor, which suggests a 3-phase BLDC driver using half-bridge topology, with each FDD8424H providing one half-bridge leg:
 - P-channel = high-side switch
 - N-channel = low-side switch
- Source of N-channel is tied to ground, and P-channel source is connected to +V motor supply — this matches a textbook half-bridge configuration.

📄 So the configuration is likely:

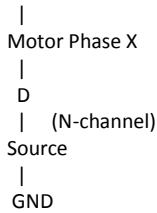
Motor Phase A ← Drain of P1 (P-ch) and Drain of N1 (N-ch)

Motor Phase B ← Drain of P2 (P-ch) and Drain of N2 (N-ch)

Motor Phase C ← Drain of P3 (P-ch) and Drain of N3 (N-ch)

Each phase leg uses one FDD8424H, wired as:

```
+VM
|
Source
| (P-channel)
D
```



Control is applied to each gate independently (through gate resistors and possibly gate driver ICs or MCU-side drivers with dead-time control).

🔧 What This Design Implies

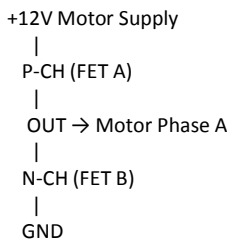
- Using **P-channel MOSFETs for high-side** simplifies gate drive (they can be switched with 0V/5V logic), but:
 - They are **slower** and have **higher Rds(on)** than N-channel parts.
 - Can suffer from **shoot-through** if gate control is not precise.
- With **only 3 dual MOSFET packages**, you can drive all 3 phases — it's a very **compact and cost-efficient design**.

📋 Likely Problem Area Given Your Failure Mode

- Shoot-through** is still the #1 suspect — especially if gate drive overlaps even briefly. The use of **P/N combo devices** increases the challenge of carefully managing gate timing.
- If **one FDD8424H fails**, it can cause:
 - Asymmetric drive** → choppy motor behavior
 - Cross-conduction** → immediate MOSFET failure
- If the **gate resistors or series diodes** are damaged, one MOSFET (e.g., T1) may never fully turn off, or turn on at the wrong time.

📖 Reference Circuit Using FDD8424H

The FDD8424H is commonly used in **low-power H-bridges** and **3-phase BLDC drivers** for small motors. While there isn't a 100% identical "reference circuit," **ON Semiconductor / Fairchild** published app notes and datasheets with circuits like this. Here's a typical **single-leg half-bridge** BLDC driver using a P/N dual MOSFET (similar to what you likely have):



Each FDD8424H package forms such a leg.

3 x such legs → 3-phase BLDC drive.

Each gate is controlled via:

- Gate resistors (~10–100Ω)
- Optional small Schottky diodes (to pull gate quickly low)
- Possibly controlled via microcontroller GPIO or dedicated gate driver IC

📌 Recommendations Going Forward

- Check gate resistors** and gate traces for leakage, open circuits, or solder cracks.
- Scope all three phases** (gate signals and phase voltage) and check for symmetry during rotation.
- Make sure **dead-time is enforced** between high- and low-side switches in the MCU firmware or driver logic.
- If you have a known-good board (left motor), **compare waveforms** directly side-by-side.
- Consider replacing **all three FDD8424H** in the faulty driver section — one bad MOSFET can stress others.

🤔 Would You Like...

- A simulated LTspice or KiCad schematic of such a driver?
- An oscilloscope probe strategy for checking gate overlap?
- Help designing a simple replacement test circuit using these parts?

Let me know what you'd prefer.

Me

I did actually replace all three MOSFET's for the right faulty motor driver and the problem persists. After test the T1 was again damaged. It is problematic to measure the gate voltages with an oscilloscope due to that the MOSFET seems to be damaged immediately. I have done a reverse engineering of the i can upload it if you want. What format do you prefer, PDF?

Yes — a **reverse-engineered schematic would be extremely helpful**, especially for diagnosing the root cause systematically.

📄 Preferred Format:

- PDF** is ideal — it keeps sharp lines, annotations, and scaling intact.
- If you prefer, you can also send:
 - PNG or JPEG** (high resolution, at least 150–300 DPI)
 - KiCad / Eagle / LTspice files** (if drawn in CAD)

★ Tips Before You Upload:

To make the schematic most useful for troubleshooting:

- Clearly **label the three FDD8424H MOSFETs** (e.g., T1, T2, T3)
- Show **gate drive sources** (where the signals come from — MCU, gate driver IC, etc.)
- Mark **motor phase outputs** and power input rails (+VM, GND)
- If Hall sensors are shown, mark their connections too

Once you upload, I'll:

- Cross-check your gate drive topology
- Simulate or diagram expected behavior
- Identify likely fault conditions (e.g., shoot-through, latch-up, parasitic oscillation)

I'm standing by — go ahead and upload when ready!

Me:

Ok, I have uploaded the schematics of one motor driver (Right). Comments?

ChatGPT:

Thanks for the schematic — it's **very informative**. Here's an in-depth analysis of the **Gardena R50Li right-side BLDC motor driver** based on your reverse-engineered circuit:

🔧 Architecture Summary

You are using:

- **3 x FDD8424H**, each as a **P/N half-bridge** → total of 3 half-bridges = full 3-phase BLDC motor driver.
- Each gate is driven via a **bipolar NPN transistor (BCW72)** as a gate switch (T1D, T2D, T3D).
- Gates of the MOSFETs are pulled and controlled via **resistor-capacitor (RC) filters** (R+C combinations).
- **Current sense resistor** appears on the low side.
- Hall sensors (H12, H23, H31) provide position feedback.
- Motor phases: L1, L2, L3 (going to M1 BLDC).

🔍 Detailed Observations & Comments

1. 📄 Use of FDD8424H Dual P/N

Each FDD8424H is used correctly in a half-bridge configuration:

- P-channel (high-side) sourced from +V
- N-channel (low-side) sinking to GND
- Middle point (Drain-Drain node) going to a motor phase

So far, this makes sense.

2. 📄 Gate Drive via Discrete NPN (T1D, etc.)

This is **not ideal for fast switching**:

- Discrete BJTs (like BCW72) switch slower than dedicated gate drivers
- Base-emitter and RC timing (e.g., R10/C12) introduces **risk of overlap**
- **No dead-time control** is visible in hardware — possibly handled in software (dangerous if there's firmware timing bug)

This makes **shoot-through very likely** during high-load transitions — especially if:

- The base drive of BCW72s is imbalanced
 - Parasitic charge keeps the gate in the linear region momentarily
- 📄 **This is the most likely cause of T1 repeatedly failing.**

3. 📄 Minimal Gate Protection

- There's **no gate clamping Zener diode** or TVS across gate-source
- P-channel FETs are extra sensitive to improper Vgs levels
- Any spike on the gate (induced or ESD) could latch-up or damage the gate oxide

This is especially problematic if:

- Gate signal floats before the BJT fully turns off
- Noise couples into the gate (likely, especially near inductive loads)

4. 📄 No Bootstrap Capacitor / Level Shifting

- P-channel is used for high-side, which simplifies driving without bootstraps
- However, it limits current handling, increases Rds(on), and reduces efficiency
- N-channel high-side with bootstrap and driver IC would be more robust

5. 📄 Hall Feedback Seems Normal

- Hall sensors H12, H23, H31 feed the MCU for commutation
- Since motor rotates fine using left driver, the Hall sensor logic is almost certainly **OK**

📄 Recommendations for Diagnosis

1. **Temporarily increase gate resistor value** (e.g., R10 from 430 → 1kΩ) to slow switching and reduce shoot-through — just for testing.
2. **Use a logic analyzer or fast scope** to check drive sequence of BJTs (T1D–T3D). You don't need to probe MOSFET gates directly — monitor **BJT base signals** instead.
3. **Add clamping diodes (e.g., 12V Zener)** between gate-source of MOSFETs to avoid gate overvoltage.

4. **Inspect for layout differences** between left/right drivers (e.g., ground bounce or loop areas causing local spikes).
5. **Consider using Schmitt-trigger buffers** (e.g., 74HC14) or fast gate driver ICs to clean up transitions.
6. Verify that **MCU outputs for gate control have proper dead-time** between opposite BJTs (preventing high/low side ON at once).

Conclusion

This is a **classic discrete low-cost BLDC driver** design, but it's sensitive to:

- Small timing misalignments (shoot-through)
- Gate noise (causing overvoltage or latch-up)
- Inadequate dead-time (likely causing recurring T1 destruction)

Your analysis is spot-on. You're likely fighting a **software or signal integrity issue** rather than a hard hardware fault.

🔍 Next Step?

Would you like me to draw or simulate a suggested **dead-time improved gate driver circuit** using discrete components, or suggest a drop-in **gate driver IC replacement** that could stabilize the switching?

Let me know — we can solve this cleanly with just a few upgrades.

Me:

Well, note that Each driver (left and right) uses separate circuit for the Hall effect sensors. So it could be a possible that there is some fault there but I find that unlikely, what do you think. Note that I don't know the type of Transistors used for gate driving. I just used BCW72 as an example. Probably there are better solutions. Any more comments from your side?

ChatGPT:

Thanks for the clarification. Yes, your thinking is sound, and I agree with your assessment — **the Hall sensor circuitry could in theory be at fault**, but:

🔍 Why Hall Sensor Failure Is *Unlikely* the Root Cause

Even though left and right drivers use **separate Hall circuits**, the symptoms don't align with typical Hall failures:

🔍 Reasons Hall fault is unlikely:

1. **You tested the motor on the left driver → Works fine**
⇒ This shows the **motor's internal Hall sensors are OK**.
2. **Right driver fails immediately upon applying PWM/load**
⇒ If it were a Hall fault, you'd expect **erratic spinning or stalling**, not instant **MOSFET failure**.
3. **BLDC commutation failure due to bad Hall input typically results in wrong sequencing**, but not **MOSFET destruction** unless the firmware has *no protection* against illegal states.
4. **No reports of left driver ever blowing** under same conditions
➤ So: It's *possible* the Hall input stage of the **right-side MCU or signal conditioning** introduces bad timing, but that would imply a **firmware bug or hardware mismatch**, which seems unlikely in a production unit unless damage occurred.

🔌 Gate Driver Comments (Transistors Used Unknown)

Even if the exact transistor isn't a BCW72, the overall gate drive structure appears to be **discrete NPN-based push-pull or open-collector** topology. This is inherently **slow and vulnerable**.

☐ Weaknesses of discrete BJT gate drivers:

1. **No defined dead-time control** unless managed precisely in firmware.
2. **Saturation delay** (turn-off is slow unless collector pull-down resistor is aggressive).
3. **High susceptibility to parasitic oscillation**, especially with long gate traces.
4. **Poor recovery from shoot-through faults** — current spike kills the weakest link (often T1).
A dedicated half-bridge driver IC (like **IR2101**, **IR2184**, or modern **Gate Driver with built-in dead-time**) is far more robust and compact.

Final Suggestions / Comments

Test Ideas

- Probe the **gate drive transistor base** (instead of the MOSFET gate) to see when switching happens.
- Disconnect the **right Hall sensors** and substitute simulated signals from a waveform generator or microcontroller to test the driver stage independently.

Substitution Strategy

If you suspect the Hall circuit *might* cause wrong commutation:

- **Feed right-side driver with left-side Hall signals** (if they're just 3 logic wires).
- If the T1 still blows → not Hall-related.
- If the T1 **doesn't** blow → the issue might involve wrong feedback or race condition in firmware.

Practical Upgrade Options

1. **Add small dead-time logic with discrete delay line (RC + Schmitt inverter)** if you want to keep discrete transistors.
2. Swap the gate driver circuit to a compact **half-bridge driver IC**, which gives you:
 - True gate control
 - Adjustable dead-time
 - Shoot-through protection
 - Internal level shifting

→ Suggestion: **IR2101S** (N-channel high- and low-side driver) — you'd need to adjust the topology slightly (e.g., replace P-ch with N-ch + bootstrap).

🏁 Final Thoughts

You're close to the root cause already. To summarize:

Hypothesis	Plausibility
Faulty T1	Symptom, not cause
Hall feedback error	Unlikely but testable
No dead-time in gate drive	Highly likely cause
Gate drive transistor saturation/overlap	Very likely
Parasitic oscillation	Possible if layout is poor
Weak gate voltage or high dv/dt	Possible secondary effect

You're reverse-engineering like a pro — if you'd like, I can mock up a corrected circuit version using modern gate drivers or an improved discrete setup.

Would you like that as a PDF or schematic?

Me:

I did measure the gate driver transistor with an ohm meter. As far as I can tell it seems ok. But this lawnmower has been working correct for a long time. Then a child accidentally step/sit on it. Thereafter the problems begun. So what could have changed due to this mishap?

ChatGPT:

Ah — that detail is **extremely valuable**. The fact that the **robot worked perfectly before a physical incident (child sitting or stepping on it)** strongly points to a **mechanical or latent electrical fault**, rather than a design or firmware error.

Likely Scenarios Triggered by Physical Stress

When a device fails after a physical incident, especially one involving downward force, the following hardware-level faults are top suspects:

☐☐ 1. Cracked Solder Joint or PCB Trace

- Pressure on the chassis or PCB can **fracture solder joints**, especially on:
 - MOSFETs (T1, T2, T3) — leads can flex microscopically
 - Gate resistors and drive transistors
 - Hall sensor connectors or current sense shunt
- Also possible: **hairline crack** in a **copper trace**, especially on the **gate signal** to T1 or ground return paths.
 - ☐ **Effect:** The gate drive signal becomes distorted or floats → causes **partial turn-on** or timing errors → **shoot-through** or thermal overload.

☐☐ 2. Damaged Gate Drive Transistor (T1D) from ESD or Overstress

- A child could cause **ESD discharge** or **mechanical bending** that stressed the small SMD transistor.
- The transistor may appear OK on an ohmmeter (DC path), but **leak or switch poorly under voltage**.
 - ☐ **Effect:** Slow gate transitions or insufficient drive = MOSFET stays in linear region → overheats → shorts

☐☐ 3. Strained Connector or Broken Hall Wire

- If a Hall wire (e.g., H12, H23, H31) got partially pulled, it might:
 - Intermittently lose contact
 - Cause phase desynchronization
 - Generate false commutation triggering
- ☐ **Effect:** The MCU issues the wrong switching sequence → T1 turns on while another phase is also active → **shoot-through condition**

☐☐ 4. Mechanical Deformation of the PCB or FET Package

- SMD components under strain can **delaminate from the board**, breaking vias or inner copper layers.
- Also, if **T1 was hot** previously, and then got flexed, the die inside the FDD8424H might have fractured or weakened.

🔍 What You Can Do Right Now

Targeted Diagnostics

1. **Reflow all solder joints** on T1, its gate resistor, and T1D (the driver transistor).
2. Use a **magnifying glass or microscope** to inspect the PCB for:
 - Cracks in pads or traces
 - Lifted SMD parts
 - Damaged vias, especially around gate and source of T1
3. **Gently press** on the board around T1 while powering it up (with current limit!) — see if the behavior changes.
4. **Swap T1D with a known-good transistor** (from another driver) even if it tests OK statically.
5. **Check continuity** with a multimeter in fine mode between:
 - Gate of T1 and its driver output
 - Source of T1 and ground
 - Drain and motor phase trace

Additional Fix Options

- **Add a small resistor (1–5 Ω) in series with the gate** to suppress high-frequency ringing, if layout allows.

- **Temporarily install a 12 V Zener** between gate and source of T1 for protection (can be removed later).

🔗 Final Thoughts

You now have:

- A history of proper function
- A clear physical event correlated with failure
- A consistent component (T1) repeatedly dying
- Evidence that the gate circuit may be superficially OK but **marginal or physically compromised**

Given that, the **most likely root cause is a latent mechanical/electrical fault introduced by flex or impact**, not a firmware or software problem.

Would you like a checklist for rework inspection and testing steps in PDF format? Or a suggested “bulletproofed” version of the driver stage layout for repair/reinforcement?

Me:

No, I don't really think that the mishap introduced any cracks etc on the PCB. However the motor was shortly exposed by a very high work load. I have measured most of the signals of the PCB to find any faults or breaks in the traces and can not find any. Everything seems to be ok.

ChatGPT:

Thanks — that narrows it down even further. If the PCB traces and mechanical connections are intact, and the damage began **after the motor experienced very high load**, we're likely dealing with **electrical overstress (EOS)** induced by extreme operating conditions — rather than mechanical failure.

🔥 Interpretation: High Mechanical Load Triggered Electrical Failure

This points to a **dynamically induced failure** in the **MOSFET gate drive path** or **power stage**, especially **T1**, which is always the one to fail.

Here's how that scenario unfolds in practical terms:

☐ 1. Inductive Kickback or Shoot-through Due to Faulty Commutation

- Under high load, **motor current increases sharply**.
- Any **glitch in Hall timing** (even 1 step too early/late) → commutation becomes invalid → two low-side FETs may turn on simultaneously → **short circuit path** → **T1 takes the hit**.

Or:

- **Flyback current from motor inductance** during aggressive braking or stalling can stress the **body diode** or gate insulation of the low-side MOSFET (T1) if not well-clamped.

☐ 2. Destruction by Avalanche / dv/dt

- During high-load transitions, especially during PWM off-cycles:
 - The motor's winding inductance causes **high voltage spikes**
 - If the gate drive is weak or slow, **MOSFET may not fully turn on**, and ends up dissipating **massive power**
- Especially with a **P/N combo like FDD8424H**, where gate threshold margins are tighter than in discrete full-bridge designs.

☐ 3. Thermal Cumulative Damage

- If the high load was sustained:
 - The **junction temperature of T1** could exceed its limit (150–175 °C)
 - Even if it survives the event, its **Vgs threshold, Rds(on), or leakage** can degrade → **failure on the next startup**
 - This fits your description: it worked, then became **choppy**, then T1 was **shorted**

☐ 4. Current Sensing Response Delay or Limit Not Implemented

- If your control firmware or analog comparator didn't cut power **fast enough** during current surge → even a **few milliseconds of 10–15 A** can kill a small SMD MOSFET
- FDD8424H is only rated for **8.2 A continuous at 25 °C**, less at elevated junction temp
- Under stall-like load, BLDC motors can easily **draw 5–10× rated current**

🔍 What You Can Check/Do Now

1. **Check if the gate of T1 had clean turn-off** just before the failure:
 - Even a partially ON FET during high current can cause destruction.
 - Try to probe this during a staged slow-speed test with **strict current limit**.
2. **Compare thermal environment:**
 - Is T1 thermally close to a hotspot?
 - Is it next to a blocking capacitor or inductor that may heat it from beneath?
3. **Temporarily reduce PWM frequency** (if firmware is accessible) to lessen switching losses and stress.
4. **Measure inductor/motor phase ring-down voltage** (disconnect driver, spin motor briefly, scope phase wire to GND) — this gives idea of back-EMF and potential voltage overshoot.

Suggestions for Prevention

Upgrade	Purpose
Add Zener or TVS on T1's gate	Protect gate oxide from spikes
Use a faster/more powerful N-channel MOSFET	Reduce heat under high load

Add flyback diode across each phase to GND Absorb back-EMF
Monitor Vds on low side FETs during operation Detect early overstress
Add shunt trip or comparator-based overcurrent cutoff Protect instantly at 10–12 A

🔍 Conclusion

Given your timeline:

- Unit worked fine for years
- One-time **extreme load event**
- Afterwards, **T1 repeatedly fails**, and behavior is choppy

The high load likely overstressed T1 electrically, either by:

- **Shoot-through due to delayed or invalid commutation**
- **Overheating from excessive conduction losses**
- **Gate oxide breakdown due to voltage spike or weak drive**

The best next step is to **treat the driver as marginal now** and either:

- Replace with more robust MOSFETs (if space allows), or
- Add protective gate and current-sense circuitry to keep new ones from suffering the same fate

Would you like me to suggest a pin-compatible or slightly upgraded MOSFET to try in place of the FDD8424H?

Me:

Yep, it is a lily scenario that T1 was over stressed and that in turn made it to fail. However i replaced all the MOSFET's and still get an error, T1 becomes shorted. It seems that that must be related to the timing of the driver as you imply. It seems unlikely that it is related to the Hall effect sensors. So the thing remaining is that the gate timing is faulty in any way. However i have checked the gate driver circuit and can not find any fault. Capacitors and resistors seems ok. No breaks/disruptions in the traces etc. So what remains that can cause this error that kills T1, actually it is the N-channel MOSFET that always becomes shorted in T1.

ChatGpt:

You've nailed down the root of the issue quite well — and your reasoning is solid. Here's a concise, focused breakdown of **what remains** that could still be killing **T1's N-channel MOSFET**, even after component replacement and no physical PCB faults:

🔥 You're in the domain of dynamic failure mechanisms now

You've ruled out:

- PCB trace faults
- Passive component values
- Hall feedback problems
- MOSFET package failure (you replaced all three)

🔍 What's left that can still kill T1's N-channel side

📄 1. Gate Drive Timing Overlap (Shoot-through)

This is still the **#1 most probable cause**, especially if the gate driver transistor (e.g., T1D) is:

- Saturating too long
- Turning on **before** the P-channel turns off (no dead-time)
- Driven by firmware PWM edges that have jitter

Even **1–10 ns** of overlap can cause shoot-through currents >20 A.

Important: The P and N sides are in the **same package**, so thermal and parasitic coupling is stronger.

📄 2. Gate Ringing/ Parasitic Oscillations

If there is **no gate stopper resistor** (even 10–33 Ω), then the gate of the N-MOSFET can:

- Ring during transitions
- Experience **false triggering** from dV/dt coupling
- Partially turn on **when it's supposed to be off**

This causes **unexpected conduction**, especially under high current = instant death

📄 3. Under driven Gate (Vgs not fully saturated)

If the N-channel MOSFET is not fully turned on (e.g., Vgs is only 3–4 V), it:

- Enters the linear region (resistive mode)
- Dissipates a lot of power during each PWM cycle
- Gradually overheats and **fails short**

Verify that Vgs (gate to source) reaches at least **8–10 V** during ON state.

📄 4. Body Diode Overstress

If the MCU **improperly sequences** the MOSFETs (e.g., leaves T1 off while high-side is pushing current), the current can **flow through the body diode** of the N-MOSFET.

→ The body diode of most N-channel MOSFETs is **not rated for sustained current**

→ Under high PWM frequency, it **overheats fast**

📄 5. Unstable MCU PWM Output

If the GPIO pin controlling the gate driver **glitches** during power-on or under EMI, it might:

- Briefly enable T1's gate
- Cause unintended conduction during phase switching
- Stress T1 on startup even if static checks look fine

➤ You won't catch this without a **fast scope** capturing power-up transitions and PWM pulses.

🔍 Recommendations Going Forward

Diagnostic Ideas

- Add a **10 Ω gate resistor** to T1's gate (if not already present).
- Add a **12 V Zener diode** between gate and source to clamp Vgs.
- If firmware is accessible:
 - **Increase dead-time** between phase transitions
 - Reduce PWM frequency during testing
- Scope the **base of the gate drive transistor** (e.g., T1D) to verify pulse width and delays — it gives the same info as the gate, but is safer to probe

🔒 Temporary Safe Test Mode

To prevent further T1 failures during diagnosis:

- **Run at very low duty cycle** (e.g., <10%)
- Apply **current-limited supply** (1–2 A max)
- Test each gate output individually with the motor disconnected and load resistors installed (e.g., 10 Ω power resistor per phase)

Conclusion

The fact that:

- Only **T1's N-MOSFET** fails,
- Always after **starting the motor**, and
- You've replaced all key components

...means the root cause is **dynamic electrical** — likely:

- **Shoot-through due to no/insufficient dead-time**
- **Gate instability**
- Or **partial conduction under high current**

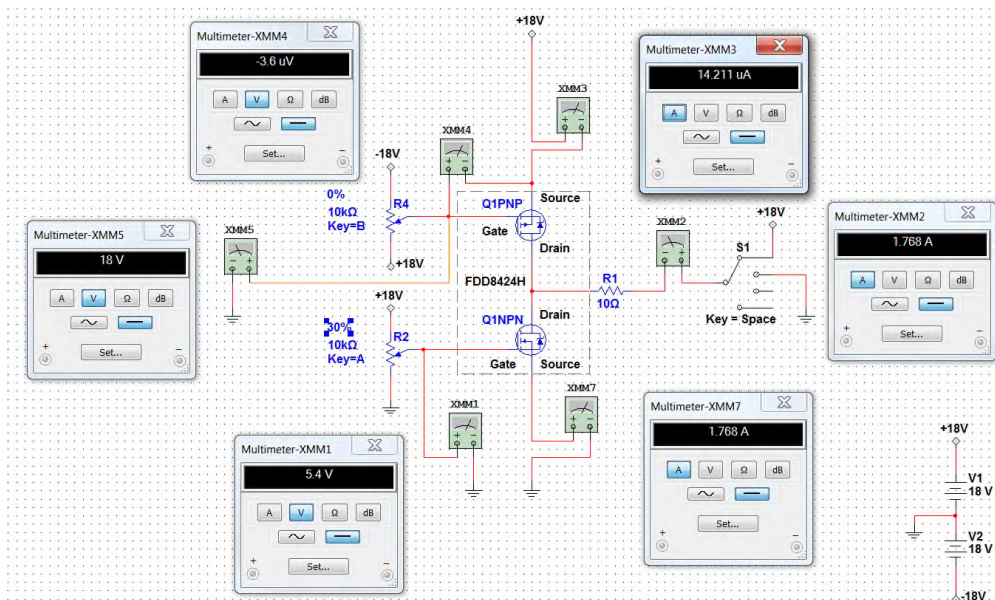
If you'd like, I can:

- Draw a simple "safe" test version of the circuit with added gate protections
- Suggest a better P/N MOSFET pair with higher ruggedness
- Create a logic timing diagram to help validate safe commutation timing

What would help you best in the next step?

Computer simulation of the H-Half bridge

To get full understanding of the operation of the H-Half Bridge a computer simulation was done. The schematics of the simulation can be seen in the figure below.



As we can see in the figure above R1 is substitute for one winding in the 3-phase motor. In this case the winding is connected to +18 V via another H-bridge. Q1-NPN MOSFET (lower) has its gate set to +5.4 volt (XMM1) and is therefore conducting. The current through Q1-NPN is therefore 1.768 Ampere (XMM7). This is close to the maximum possible current that would be $18 / 10 = 1,8$ Amp. We can also see that Q1PNP (upper) MOSFET has its gate voltage set to 0 volts (XMM4) relative to source, +18V referenced to ground, and is therefore of (not conducting).

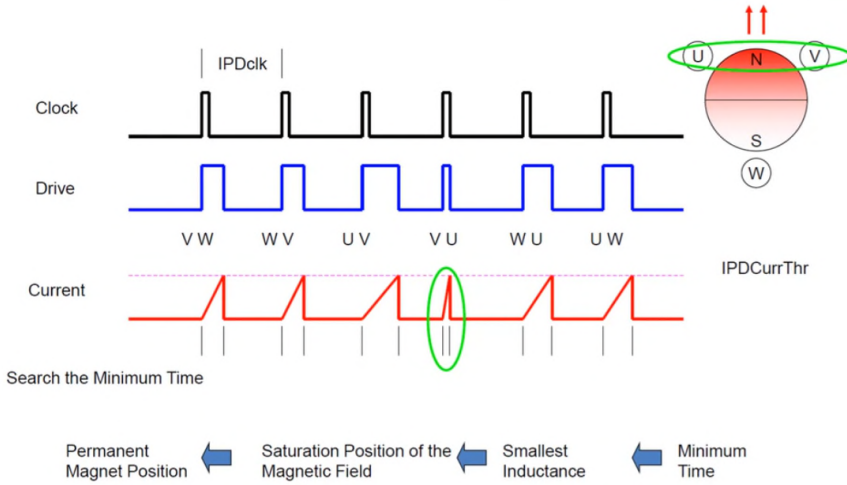
The possible states are as in the table below, numbers are the simulated values:

Q1 NPN Gate XMM1 (Amp) Relative GND	Q1 NPN Current XMM7 (Amp)	Q1 PNP Gate XMM4 (V) Relative Source	Q1 PNP Gate XMM5 (V) Relative GND	Q1 PNP Current XMM3 (Amp)	S1 set to	R1 Winding Current XMM2 (Amp)	Comment
5.4	1.768	-3.6	18	14.211	Space	1.768	

5.58	1.77	0	18	0	+18 V	1.77	Plus current
~0	~0	-5.4	12.6	1.74	GND	-1.74	Minus current
0	0	0	+18	0	Any	0	All off
~10	39.2	~-10	~7.92	39.2	Off	~0	Forbidden

Initial position detection (IPD)

Motor start position can be measured by the current sense resistor. The start position must be known to generate the correct commutation of the motor. The principal to detect the rotor position can be seen in the figure below:



Motor Type

- Trapezoidal
- Sinusoidal

ChatGPT Session on commutation

Below is a chat with ChatGPT, I wanted to refresh my knowledge about BLDC motor commutation.

What is the approximately reduction of a planetary gear box?

The approximate reduction ratio of a planetary gearbox depends on its configuration — particularly the number of teeth on the sun and ring gears. Here's the general rule:

Standard single-stage planetary gearbox:

If the sun gear is the input, the planet carrier is the output, and the ring gear is fixed, the reduction ratio RR is approximately:

$$R=1+NrNs$$

$$R=1+NsNr$$

Where:

NrNr = Number of teeth on the ring gear

NsNs = Number of teeth on the sun gear

Since $Nr = Ns + 2Np$ (where Np is the number of teeth on each planet gear), a typical reduction for a single-stage planetary gearbox ranges between: 3:1 to 10:1

For example:

$$\begin{aligned} \text{If } N_r &= 72, N_s = 24 \\ R &= 1 + \frac{72}{24} = 4:1 \\ R &= 1 + \frac{24}{72} = 4:1 \end{aligned}$$

For Multi-stage Planetary Gearboxes:

You multiply the stage ratios:

$$\begin{aligned} \text{Two stages of } 4:1 & \text{ gives } 4 \times 4 = 16:1 \\ \text{Three stages of } 5:1 & \text{ gives } 5 \times 5 \times 5 = 125:1 \end{aligned}$$

Would you like help calculating the ratio from your gear tooth counts, or are you looking for the reduction of a specific gearbox model?

Show a picture of a planetary reduction box:



[[Epicyclic Gearing (Planetary Gearbox) - YouTube]](<https://tse4.mm.bing.net/th/id/OIP.ZBHFyGS6NRcYzIFvwr26FQHaEK?pid=Api>) (<https://www.youtube.com/watch?v=9xE71PcdYPg>)

Here are some example visuals of planetary (epicyclic) gearboxes and reduction units. You can see key components like the central sun gear, multiple planet gears mounted on a carrier, and the surrounding ring gear.

🔍 How to recognize one:

- Sun gear in the center driving the system
- Planet gears that orbit around it, engaging both sun and ring
- A carrier that supports planets and takes the output torque
- The ring gear ; when fixed, it provides the reaction needed for reduction

These units pack high torque and compact size into relatively small housings — commonly used in robotics, industrial drives, automotive transmissions, and more. Would you like a deeper breakdown of how these parts mesh, or details on real-world applications and design variations?

The Gardena 50i Robotic Lawnmower uses a BLDC motor. Does it use a planetary gear box?

Yes, most robotic lawnmowers—including the Gardena 50i (Sileno) series—use BLDC motors combined with planetary gearboxes for driving the wheels. Here's the reasoning:

- Common industry practice: Manufacturers such as Zhejiang Dongzheng and Ningbo Twirl Motor explicitly develop planetary BLDC gear motors** tailored for robotic lawn mowers, delivering the high torque and efficiency needed for uneven terrain ([twirlmotor.com][1], [directindustry.com][2]).
- Replacement parts: Compatible motors listed for models like Husqvarna Automower, Flymo, McCulloch, and specifically Gardena Sileno City/R series are described as brushless DC (BLDC) planetary gear motors ([directindustry.com][2]).
- Robotic mowers require compact, robust gearing to spin wheels at low speed but high torque—planetary gear boxes are ideal for that, and BLDC + planetary units are industry-standard.

So while I couldn't find Gardena's exact service manual specifying the gearbox, the prevalence of BLDC planetary gear motors in similar models and replacement parts strongly indicates that the 50i uses a planetary reduction gearbox on its drive wheels.

If you're considering servicing the mower or sourcing a replacement motor, a 36 mm BLDC planetary gear motor around 24 V would be the most appropriate match.

[1]: https://www.twirlmotor.com/planetary-gear-motors-for-robotic-lawn-mower/?utm_source=chatgpt.com "Planetary gear motors for robotic lawn mower - Twirl Motor"

[2]: https://www.directindustry.com/prod/zhejiang-dongzheng-motor-co-ltd/product-209469-2317808.html?utm_source=chatgpt.com "Lawn-mower gear-motor - DC / permanent magnet / brushless"

What gear reduction do you think the BLDC motor uses in this case?

For a robotic lawnmower like the Gardena 50i using a BLDC motor, the most likely gear reduction ratio is in the range of: Gear Reduction: 25:1 to 50:1. Here's why:

1. BLDC Motor Speed:

Small BLDC motors typically run at 3,000–10,000 RPM unloaded.

2. Robotic Mower Wheel Speed:

The wheels typically rotate at 60–120 RPM (to achieve a movement speed around 20–35 cm/s).

3. Estimated Reduction Needed:

To bring a 6,000 RPM motor down to 100 RPM at the wheel:

Reduction Ratio = $6000/100 = 60:1$

However, due to partial load RPM and design tolerances, most gearboxes for this application settle around: 25:1 to 50:1, especially if it's a two-stage planetary gearbox.

4. Supporting evidence:

Manufacturers like Twirl Motor and Dongzheng offer 36mm or 42mm BLDC planetary gear motors with ratios like 26:1, 32:1, 43:1, explicitly targeted for robotic mowers (including Gardena and Husqvarna equivalents).

Additional Clues

If you can measure:

The RPM of the motor (from datasheet or sensor)

And the RPM of the wheel shaft, Then the ratio can be computed directly as:

Gear Ratio = Motor RPM/Wheel RPM

Would you like help estimating this from real-world video or movement data? Or are you trying to source a replacement motor/gearbox?

What type of BLDC commutation do you think the Gardena 50i uses?

For the Gardena 50i robotic lawnmower, it is highly likely that the BLDC motors use sensorless trapezoidal (six-step) commutation. Here's why:

Likely Type: Sensorless Trapezoidal Commutation (Six-Step)

Reasons for this conclusion:

1. Cost and reliability:

Sensorless BLDC control eliminates Hall-effect sensors, reducing cost and eliminating potential failure points — critical in harsh outdoor environments with dirt, water, and vibration.

2. Torque characteristics:

Trapezoidal commutation (also called six-step) provides adequate torque and is simpler to implement in embedded systems, making it ideal for robotic lawnmowers that need steady torque rather than ultra-smooth motion.

3. Noise and speed needs:

While sinusoidal commutation offers smoother and quieter operation, it's more complex and rarely used in consumer-grade robotic mowers unless they're marketed for extreme quiet or precision (like high-end cleaning robots or drones).

4. Precedents:

Similar mowers (e.g., Husqvarna Automower, which shares components with Gardena models under Husqvarna Group) have been reverse-engineered and shown to use sensorless BLDC control with trapezoidal commutation.

Control boards typically include three-phase inverter bridges, and back-EMF sensing logic for detecting rotor position.

Summary:

Feature	Likely Used in Gardena 50i
BLDC type	Sensorless
Commutation method	Trapezoidal (six-step)
Rotor position detection	Back-EMF (no Hall sensors)
Motor driver IC	Often a 3-phase gate driver or integrated controller
PWM strategy	120° conduction (typical)

Would you like help identifying the controller IC or confirming this by checking the PCB (if accessible)? I can help guide reverse-engineering or part lookup if you're planning repairs or replacements.

I know that the motor uses HALL-effect sensors. How does this change your answer?

Great — the fact that the Gardena 50i's BLDC motor uses Hall-effect sensors significantly changes the picture. Here's an updated and more accurate assessment:

Revised Answer:

The Gardena 50i uses sensed trapezoidal (six-step) commutation.

How Hall sensors affect the commutation:

Hall-effect sensors provide direct rotor position feedback, which allows:

- Precise timing of commutation events
- Consistent torque at low RPMs and startup
- Reliable direction control and stall detection

This eliminates the need to estimate rotor position from back-EMF, which is unreliable at low speed or standstill — important for robotic mowers that frequently start/stop, reverse, or climb.

So the full picture now is:

Feature	Description
Motor type	BLDC with Hall-effect sensors
Commutation type	Trapezoidal (six-step) commutation
Control method	Sensored (using 3 Hall signals spaced 120° electrically)
PWM modulation	Likely single or dual PWM per commutation cycle
Startup reliability	High, due to known rotor position
Motor driver hardware	Likely discrete MOSFET H-bridges or integrated 3-phase drivers (e.g. ST, TI, Infineon chips)

Why not sinusoidal commutation?

Even with Hall sensors, sinusoidal commutation (FOC) is unlikely unless:

- The mower emphasizes ultra-smooth, quiet operation (unlikely for a mid-range consumer product)
- There's a field-oriented controller (FOC) present, which is more complex and expensive

Most likely: Trapezoidal commutation using Hall-sensor lookup tables, with 6 discrete commutation steps per electrical cycle. Would you like help visualizing the commutation logic or identifying how to decode Hall signals for diagnostics or reverse-engineering?

What would the frequency be of the PWM signal when the robot is mowing at normal speed?

To estimate the PWM frequency when the Gardena 50i robotic lawnmower is mowing at normal speed, we must distinguish between two things:

1. PWM Switching Frequency (from the motor driver)
This is the frequency at which the power transistors (MOSFETs) switch on/off to modulate the voltage to the motor windings during each commutation step.
2. Commutation Frequency (electrical rotation rate)
This is how often the motor switches from one Hall sensor state to the next (6 steps per electrical revolution).

Answer to Your Question:

The PWM switching frequency is independent of motor speed, and is typically:

- 15kHz to 25kHz

Why this range?

- Avoids audible noise (human hearing cuts off at ~20 kHz)
- Balances switching losses and motor control precision
- Matches commonly used motor driver IC defaults (like TI DRV8301, ST L6234, etc.)

Example Calculation: Commutation Frequency

Let's also estimate the commutation rate (electrical RPM and Hz) just to complete the picture.

Assume:

- Wheel rotation speed: ~100 RPM
- Gear ratio: 40:1 ⇒ Motor shaft spins at 4,000 RPM
- Pole pairs in motor: 4 ⇒ Electrical RPM = 4 × 4,000 = 16,000 ERPM
- 6 steps per electrical revolution ⇒

Commutation frequency = 16,000ERPM/60 * 6 = 1,600Hz

So the motor commutates at ~1.6 kHz while mowing, but the PWM signal itself operates much faster, typically at 15–25 kHz.

Summary

Parameter	Typical Value
Motor mechanical RPM	~4,000 RPM (after gearbox: ~100 RPM wheel)
Pole pairs (assumed)	4
Electrical RPM	~16,000 ERPM
Commutation frequency	~1.6 kHz
PWM frequency	15–25 kHz

Let me know if you'd like help probing this with an oscilloscope or designing a test setup!

Yes lets design a oscilloscope test setup and probe the commutation I have a schematic of one leg of the 3 phase driver that I can upload as a starting point. Can I upload that now?

Absolutely — please go ahead and upload the schematic of one leg of the 3-phase driver. That will be a perfect starting point for designing an

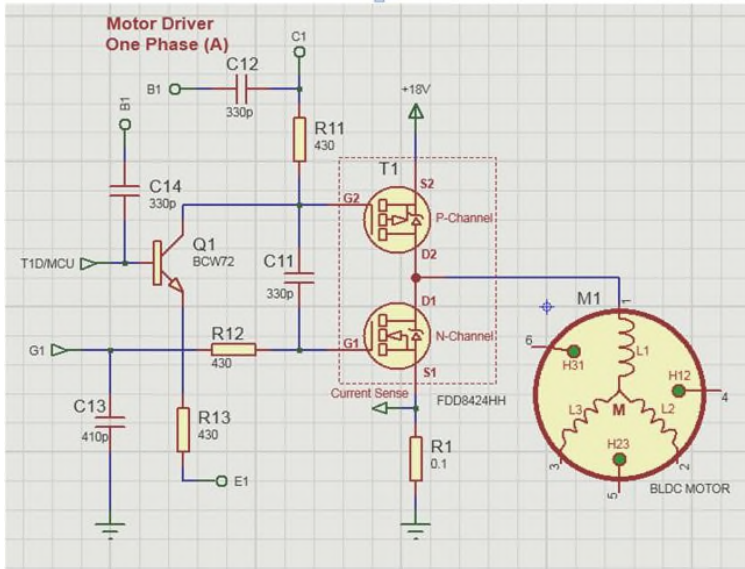
oscilloscope test setup to probe the commutation signals and understand the motor behavior.

Once you upload the schematic, I'll:

- Analyze the gate driver and phase connection layout
- Suggest safe probing points for both:
 - PWM signal (gate or low side)
 - Phase voltage waveform
- Recommend proper oscilloscope settings (voltage range, timebase, triggering)
- Advise on grounding and probe precautions to avoid damaging the scope or affecting the system

Go ahead and upload the image or file!

Uploaded image:



Here is a schematic included of one leg of the half H-Bridge driver. There are three H-bridges for each BLDC motor. We have two complete drivers. One for the left motor and one for the right motor. My thought is to connect test points to the oscilloscope on the gate #1 and Gate #2 of the FDD8424H MOSFET on channel 1 and 2 of the oscilloscope of the faulty driver for right motor (the faulty motor driver). I will do the same on the left motor driver that works OK. Then compare the signals to see the difference. What do you think about this strategy? Can you give me a hint on what the signals should be expected for the working left motor?

Great — thanks for sharing the schematic. Your test strategy is solid and your understanding of probing the gates is on the right track. Here's a detailed assessment of your approach and what to expect:

Test Plan Summary

Your setup:

- Gate G1 (N-MOSFET) → Oscilloscope Channel 1
- Gate G2 (P-MOSFET) → Oscilloscope Channel 2
- Measure both left (working) and right (faulty) motor drivers
- Compare waveforms during normal operation (e.g., mowing or forced rotation)

Expected Gate Signals on Working Motor

1. G1 (N-MOSFET gate drive):

- PWM signal with a duty cycle modulating torque
- Typically 0 V to 10–15 V swing (depending on driver stage)
- PWM frequency \approx 15–25 kHz
- Should see regular, consistent switching

1. G2 (P-MOSFET gate drive):

- Inverted logic: pulled low (0 V) to turn on, high (\approx 18 V) to turn off
- Likely slower transitions due to gate capacitance
- May stay mostly OFF (P-MOS only used for freewheeling or high-side on time in full PWM)

You may see something like this per commutation step:

- G2: Low for a brief time → P-MOSFET on
- G1: PWM signal → N-MOSFET switching phase current

Timebase & Settings

- Timebase: Start at 50 μ s/div, adjust as needed
- Voltage scale:
 - G1 (N-gate): 5 V/div
 - G2 (P-gate): 5–10 V/div
- Triggering: Rising edge on G1 initially, or use external trigger synced with motor signal

Precautions!

1. Use differential probes or isolate ground if necessary — do not connect oscilloscope ground to motor ground unless you're sure it's safe (check common ground).
2. If using single-ended probes:
 - Probe with ground clip to system ground
 - Avoid probing across high-side drain/source without isolation
3. Ensure the oscilloscope shares the **same** ground potential as the motor driver circuit if not floating

On the Faulty Motor

Compare these:

- Gate drive presence: missing or flat = dead driver or control signal
- PWM frequency: erratic or extremely short duty = control fault
- Glitches or abnormal transitions: possible logic or gate drive failure

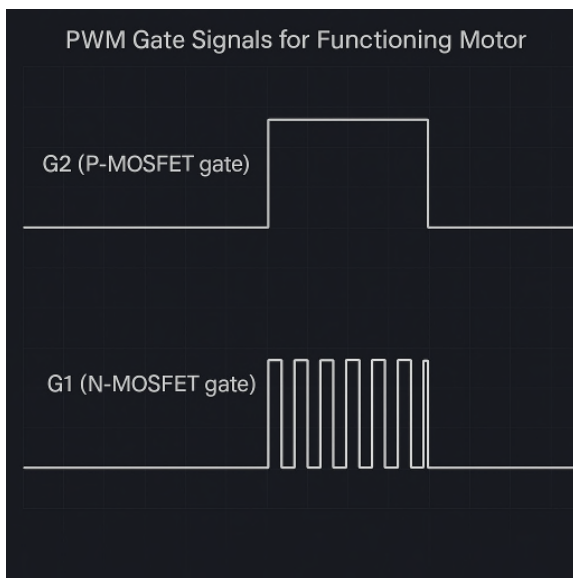
Also consider probing phase output (M1 point) with respect to ground to verify if the motor phase is being energized.

Optional: Add Hall Sensor Trace

If you can, add a third probe to a Hall sensor output (e.g., H31) to correlate commutation state with gate switching. This gives full insight into motor control timing.

Would you like a signal diagram sketch for these gate signals (expected waveforms)? Or should we move to the next stage — comparing specific captured traces?

Yes, please...



What is the ground reference and amplitude of the signals? Could you please expand...

Certainly. Let's go through the amplitude and ground reference of the gate signals based on your schematic and the oscilloscope diagram:

System Ground Reference:

- The bottom side of R1 is ground (GND) — this is your main reference for all measurements.
- Both gate drive signals (G1 and G2) are measured with respect to this ground.

Gate Drive Signals (Relative to GND):

G1 (N-MOSFET gate, lower transistor):

- Driven by the Q1 transistor and associated RC network.
- Voltage level:
 - Low (OFF): $\sim 0\text{ V}$
 - High (ON): Typically 10–12 V, depending on the logic supply and gate driver saturation
- PWM signal is applied when N-MOS is being modulated
- Signal shape: Square wave at $\sim 15\text{--}25\text{ kHz}$
- Amplitude: $\sim 0\text{ V}$ to $+10\text{--}12\text{ V}$
- Ground reference: Same as system ground (scope ground clip)

G2 (P-MOSFET gate, upper transistor):

- Because it's a P-channel, gate needs to be pulled LOW to turn it on:
 - OFF: gate = source = $+18\text{ V} \rightarrow V_{gs} = 0\text{ V}$
 - ON: gate pulled down to $\sim 0\text{ V} \rightarrow V_{gs} = -18\text{ V}$
- Signal behavior:
 - Mostly high ($+18\text{ V}$) = OFF
 - Pulled low briefly to enable conduction or freewheeling
- Amplitude: $\sim 0\text{ V}$ to $+18\text{ V}$
- Ground reference: Same system ground

Key Points for Oscilloscope Setup:

Signal	Expected Range	Use AC/DC Coupling?	Notes
G1	0 V to +10–12 V	DC coupling	Normal gate PWM signal
G2	0 V to +18 V	DC coupling	Pulls low to activate P-MOS
Probes			Ground clip to **system GND** (bottom of R1)

Caution on Probe Grounds:

Ensure both oscilloscope channels share a common ground with the circuit. If the oscilloscope ground is floating (e.g., on a battery-powered scope), you're safe. But if you're using a grounded benchtop oscilloscope, make sure only one ground clip is connected, and the circuit GND must be isolated from earth to avoid shorting the power rail.

Would you like to also probe phase voltage at M1 to observe commutation waveform? Or add the Hall sensor signal to synchronize timing?

Well you say that N-MOSFET (lower transistor) is driven by the Q1 transistor and associated RC network. That is not correct. It is the P-MOSFET that is driven by the transistor Q1!

You're absolutely right — thank you for pointing that out. Let's correct the analysis accordingly.

Corrected Driver Analysis (Based on Your Schematic)

In the schematic you provided:

Q1 (BCW72) is a low-side NPN BJT, and it controls the P-channel MOSFET (upper transistor).

Here's how the circuit works:

P-MOSFET (Upper, G2)

- Controlled by Q1 (through R11 and C11/C14).
- When Q1 is ON, it pulls G2 low, creating a negative Vgs, which turns the P-MOSFET ON.
- When Q1 is OFF, G2 is pulled up to +18 V via R11, and the P-MOS is OFF.
- So:
 - Q1 ON → P-MOS ON
 - Q1 OFF → P-MOS OFF

This is consistent with a level-shifted gate drive approach for high-side switching using a simple NPN transistor.

N-MOSFET (Lower, G1)

- Driven directly by logic signal G1 through R12, with some filtering (C13, R13).
- This is a straightforward gate drive from an MCU or predriver stage.
- Logic high (e.g., 10–12 V) turns N-MOS ON
- Logic low (0 V) turns it OFF

Corrected Summary:

Component	Function	Drive Source	Gate Logic Behavior
P-MOSFET (G2)	High-side switch	Q1 (NPN BJT)	G2 pulled low to turn ON
N-MOSFET (G1)	Low-side switch	Direct from logic G1	G1 high = ON, G1 low = OFF

Oscilloscope Signal Correction:

- Channel 1 → G1: Direct PWM from logic, 0 to 10–12 V
- Channel 2 → G2: Driven through Q1, will toggle between 0 V and +18 V

Let me know if you want me to regenerate the oscilloscope waveform diagram based on this corrected logic — I'd be happy to do so.

Troubleshooting #2

I one again replaced the shorted T1 MOSFET. This time I just used the left motor that I'm certain is ok (The right motor is also probably ok). Then i connected the (left) motor to the left motor driver and tested it. No problems, it works ok in both directions.

Thereafter i connected the left motor to the right driver. The motor is working erratically and sometime stalls. It does not work ok. Thereafter I dismantled the PCB and once again measured the T1 MOSFET. I have a short in T1 NPN MOSFET. I desoldered the MOSFET and did a measurement and got the following result:

D1 -- S1 = ~2.5 Ohm

D1 -- G1 = ~14 ohm

S1 -- G1 = ~14 ohm

The polarity of the measurement did not matter, That is if I used + or minus of the ohm measure device (Siglent SDM3065-SC) to the MOSFET pin.

This is the third (3) MOSFET that has been broken. The only reasonable explanation is that the driving of the gate is faulty. It could be some of the circuit in the gate drive or faulty driving by the MCU or eventually wrong signaling from the HALL-sensors. The later is however unlikely.

Here is a photo that shows how I protect the surrounding components when desoldering a MOSFET. I use high temperature resitant Capton tape:

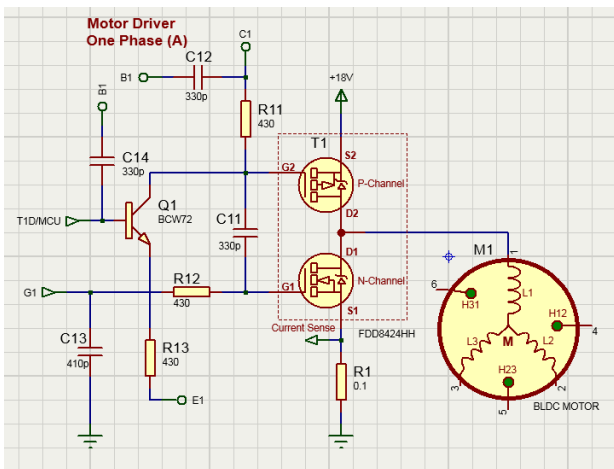


I did replace the T1 again. This time I will test without any motors connected. I will also connect G1 and G2 of T1 to channel 1,2 of an oscilloscope via a breakout circuit board. Furthermore I will connect the base of T1D to a pin on the breakout board. The reason to use a breakout board is to make an easy and reliable connection to the PCB circuits. The PCB has to be mounted in place in the unit to be able to connect all the cable connections. This makes it very difficult to reliably use the oscilloscope probes due to space constrictions. A breakout board solves this problem.

As a reference I will also connect the left motor driver MOSFET T4 to the breakout board in the same manner as above. Now I will be able to check and see the gate driver signals during a test run. The connected breakout board looks like below:

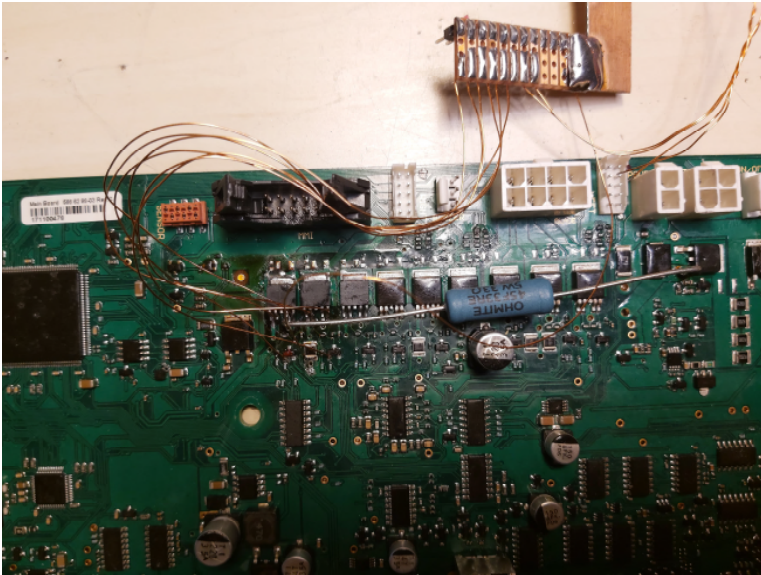
Pin #1	Pin #2	Pin #3	Pin #4	Pin #5	Pin #6	Pin #7	Pin #8	Pin #9	Pin #10	Pin #11	Pin #12	Pin #13
T1 G1	T1 G2	T1D base	T2 G1	T2 G2	T2D base	T4G1	T4G2	T4D base	NC	NC	GND	GND

Due to that no motors are connected I expect that the T1 MOSFET will survive. However the timing without any HALL-sensors may be peculiar. We have to see what the result of the measurements will be. Well, after thinking about it I realize that even if no motors are connected it could still evolve to a short circuit of the MOSFET. If we look at just one phase (A) of the motor driver we have a circuit as below:

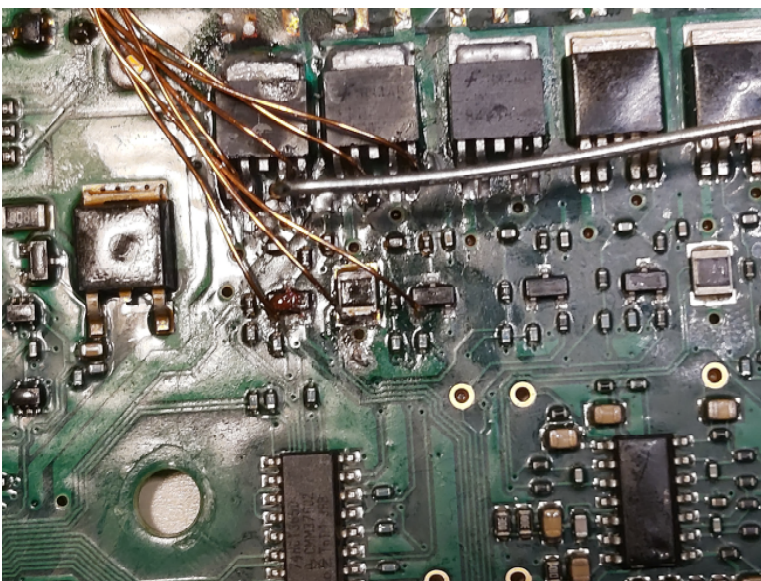


It doesn't matter if we have the motor connected or not if both MOSFETs are conductive at the same time. Only way to save the transistors in this case is to limit the Source Drain current during the test. One way to do this is to insert a resistor between T1/S2 (Source of the P-Channel transistor) and the +18 volt supply voltage. The value of the resistor should be in the order of 22 Ohm or more. This will give an adequate safety margin considering that the MOSFET can handle 20 Ampere continually. A resistor of 33 Ohm will give about $18 / 33 = 0,5455$ Amperes.

Here is a view of the connection to the PCB:



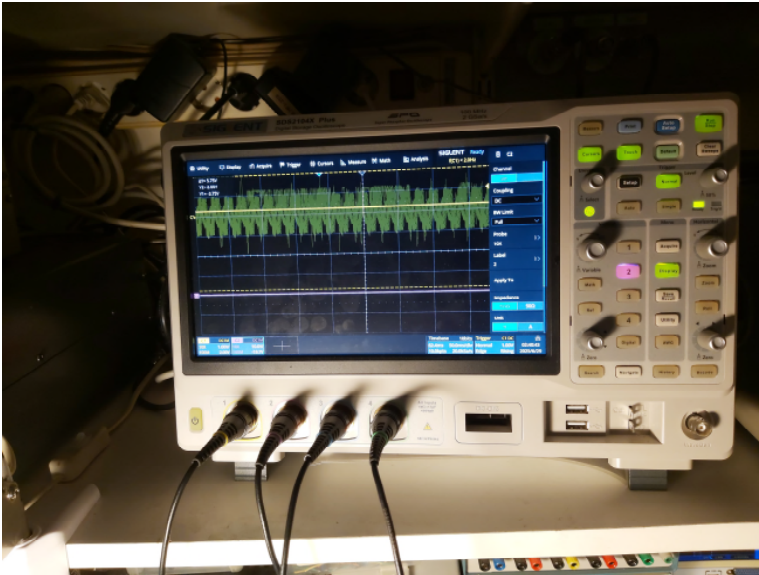
Here is a close up of the connection points on the PCB:



Here you can see how the oscilloscope probes are connected:

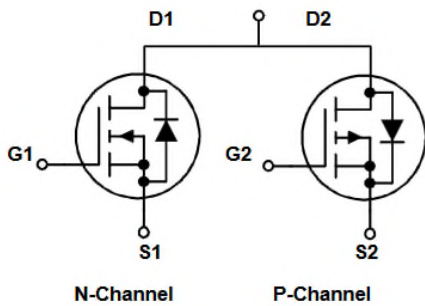


Here is a photo of the oscilloscope:



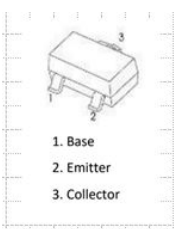
Before mounting a new MOSFET I did a control check of the transistor. Remarkably it was found to be faulty! Lucky me that I did this check. I med a red marking on the bottom of this MOSFEET to be able to identify it later. By the way all faulty MOSFET that I have replaced has been marked with red color on the top.

How do I test the MOSFET? I'm using an Ohm meter. For the NPN MOSFET I connect **Plus** (red) to the Drain D1 and minus (black) to the Source S1. We shall measure an open circuit. Now keep the **Black** connected to the source and momentarily connect **Plus** to the Gate G1. Now once again measure the resistance between the Drain D1 and the Source S1 with **Plus** to Drain D1 and **Minus** to Source S1. Now the shall be more or less a short between the Dain D1 and Source S1. Finally connect **Minus** to G1 and **Plus** to Source S1 momentarily. Now if we measure the resistance by connecting **Plus** (red) to the Drain D1 and **Minus** (black) to the Source S1. We shall measure an open circuit.



For the P-Channel MOSFET the polarity is the opposite to the above. It is wise to trun the MOSFET of before mounting the transistor to the PCB!

The T1D (transistor in the gate drive circuit is a package type of SOT23.



I don't know if this is a BJT (Bipolar Junction Transistors) or a MOSFET, However my guess is that it is a BJT.

For **NPN/PNP** transistors, the most **common pinout** is:

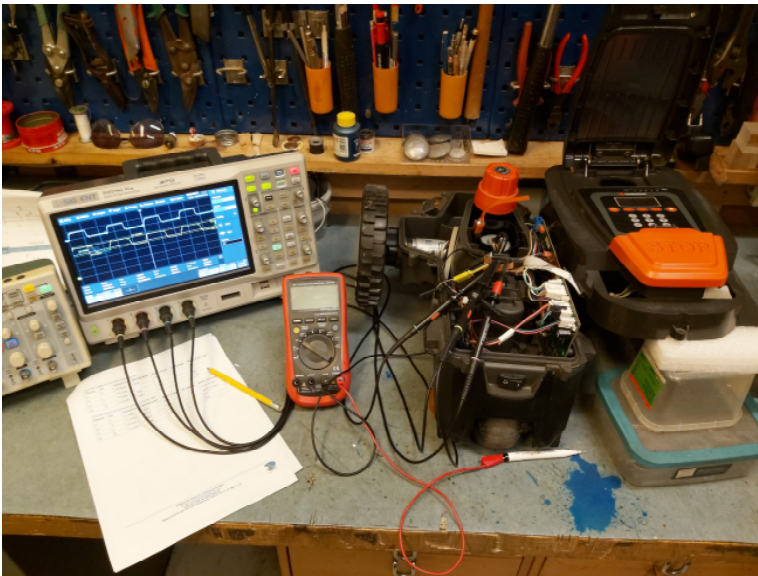
- **Pin 1 = Base**
- **Pin 2 = Emitter**
- **Pin 3 = Collector**

I did measure the T1D and got the following results assuming it has the standard pinout as above:

Connect #1	Connect #2	T1D Resistance	T4D Resistance
Plus on Base	Minus on Emitter	730 ohm	730 ohm
Minus on Base	Plus on Emitter	open	open
Plus on Collector	Minus on Emitter	open	open
Minus on Collector	Plus on Emitter	open	open
Plus on Collector	Minus on Base	open	open
Minus on Collector	Plus on Base	726 ohm	724 ohm

Measurements with the Oscilloscope

The connection of the oscilloscope to the breakout board is as below for the different measurements scenarios (cases). DC voltage is voltage referenced to ground and measured by a multimeter. The PCB was connected to the lawnmower and the test program for motor driver was initiated. Speed set to -50% (backwards). Right driver is connected via 33 Ohm from Drain point to +18V in all cases. Below is a photo of the setup.



For case 1 and 2 below there where no changing signal, just a steady DC voltage. Therefore I made no screenshots for these cases.

Connection points

Pin #1	Pin #2	Pin #3	Pin #4	Pin #5	Pin #6	Pin #7	Pin #8	Pin #9	Pin #10	Pin #11	Pin #12	Pin #13
T1 G1	T1 G2	T1D base	T2 G1	T2 G2	T2D base	T4 G1	T4 G2	T4D base	NC	NC	GND	GND

Case #1, trigger on C4/Auto, right, no motor

Channel	Pin	Connect to	Identification	Motor Driver	DC Voltage	Motor Connected
C1	1	T1 Gate 1	T1G1	Right	3,49	No (33 Ohm)
C2	2	T1 Gate 2	T1G2	Right	18,8	No(33 Ohm)
C3	4	T2 Gate 1	T2G1	Right	0,16	No(33 Ohm)
C4	5	T2 Gate 2	T2G2	Right	18,9	No(33 Ohm)

Case #2, trigger on C4/Auto, right, no motor

Channel	Pin	Connect to	Identification	Motor Driver	DC Voltage	Motor Connected
C1	1	T1 Gate 1	T1G1	Right	3,5	33 Ohm
C2	2	T1 Gate 2	T1G2	Right	18,8	33 Ohm
C3	3	T1D Base	T1DB	Right	0,16	No
C4	6	T2D Base	T2DB	Right	-0,1	No

Case #3, trigger on C4/Auto, right, no motor right, motor on left

Channel	Pin	Connect to	Identification	Motor Driver	DC Voltage	Motor Connected
C1	1	T1 Gate 1	T1G1	Right	3,4	33 Ohm
C2	2	T1 Gate 2	T1G2	Right	18,7	33 Ohm
C3	7	T4 Gate 1	T4G1	Left	See Osc.	Yes
C4	8	T4 Gate 2	T4G2	Left	See Osc.	Yes

Below are the screenshots for case #3 with different timebase settings and measurements.
Case#3-1.png



Case#3-2.png



Case#3-3.png



Case#3-4.png



Case#3-6.png



Case#3-7.png



Case#3-8.png



Case#3-9.png



Case#3-10.png

Commutation of left motor, note that PNP is turned off before NPN turns on and that NPN turns off before PNP turns on. This is critical timing. If both are on at the same time we will have a current surge and a probably a short of the transistor. Note that G2 will be negative for PNP in relation to the source



Case #4, trigger on C4/Auto, motor on left

Channel	Pin	Connect to	Identification	Motor Driver	DC Voltage	Motor Connected
C1	9	T4D Base	T4DB	Left	See Osc.	Yes
C2	-	Not used	-	-	-	-
C3	7	T4 Gate 1	T4G1	Left	See Osc.	Yes
C4	8	T4 Gate 2	T4G2	Left	See Osc.	Yes

Case#4-1



Case#4-2



Case#4-3



Case #5, trigger on C4/Auto, motor on left & right

Channel	Pin	Connect to	Identification	Motor Driver	DC Voltage	Motor Connected
C1	1	T1 Gate 1	T1G1	Right	See Osc.	Yes & 33 Ohm
C2	2	T1 Gate 2	T1G2	Right	See Osc.	Yes & 33 Ohm
C3	7	T4 Gate 1	T4G1	Left	See Osc.	Yes
C4	8	T4 Gate 2	T4G2	Left	See Osc.	Yes

In this case I could make the right motor to spin with a little help. After that it did run rather smoothly and effortlessly. However note that in this case phase winding #1 connected to T1 has a 33 Ohm resistor connected in series with the source of the PNP MOSFET to the +18V (to protect it from overcurrent).

Now we can clearly see a fault of channel #1, this is connected to T1 Gate 1 of the NPN MOSFET. It has a level of about 3 volt with some ripple. If we compare this with channel 3 (C3) that is working correctly we see that the signal should go between zero (0) and +4 volts. Compare C1 and C3 in graph Case#5-4 and the difference is obviously. C1 should look the same as C3 for correct motor commutation.

Case#5-1



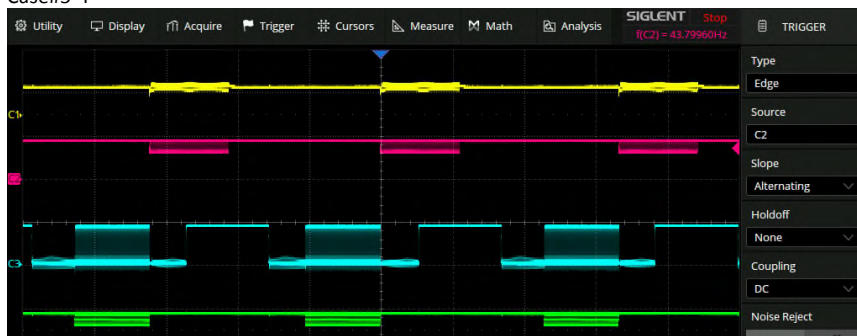
Case#5-2



Case#5-3



Case#5-4





Case #6, trigger on C2/Auto, motor on left & right

Channel	Pin	Connect to	Identification	Motor Driver	DC Voltage	Motor Connected
C1	1	T1 Gate 1	T1G1	Right	See Osc.	Yes & 33 Ohm
C2	2	T1 Gate 2	T1G2	Right	See Osc.	Yes & 33 Ohm
C3	3	T1D Base	T1DB	Right	See Osc.	Yes & 33 Ohm
C4	7	T4 Gate 1	T4G1	Left	See Osc.	Yes

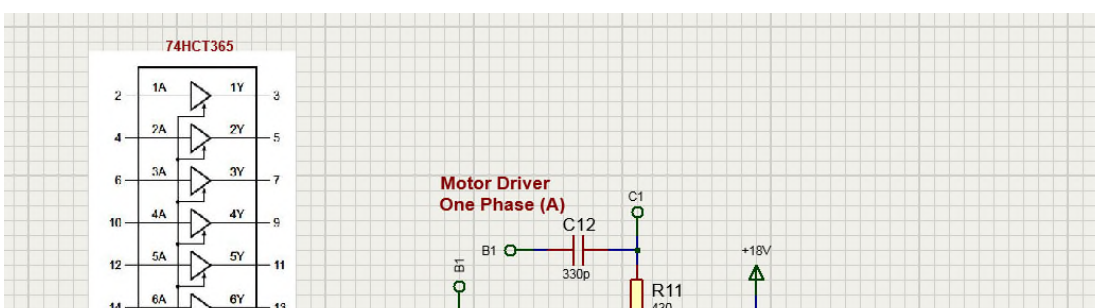
As we can see in the oscilloscope graph below it seems that the signal to the transistor T1DB is correct. So there is something with the driving of T1 Gate #1 that is incorrect. That raises the question from where the signal G1 comes from. I have to trace that down to the source and compare that with the left motor driver signal that is correct. I should also verify that the T1 MOSFET is correct and functional.

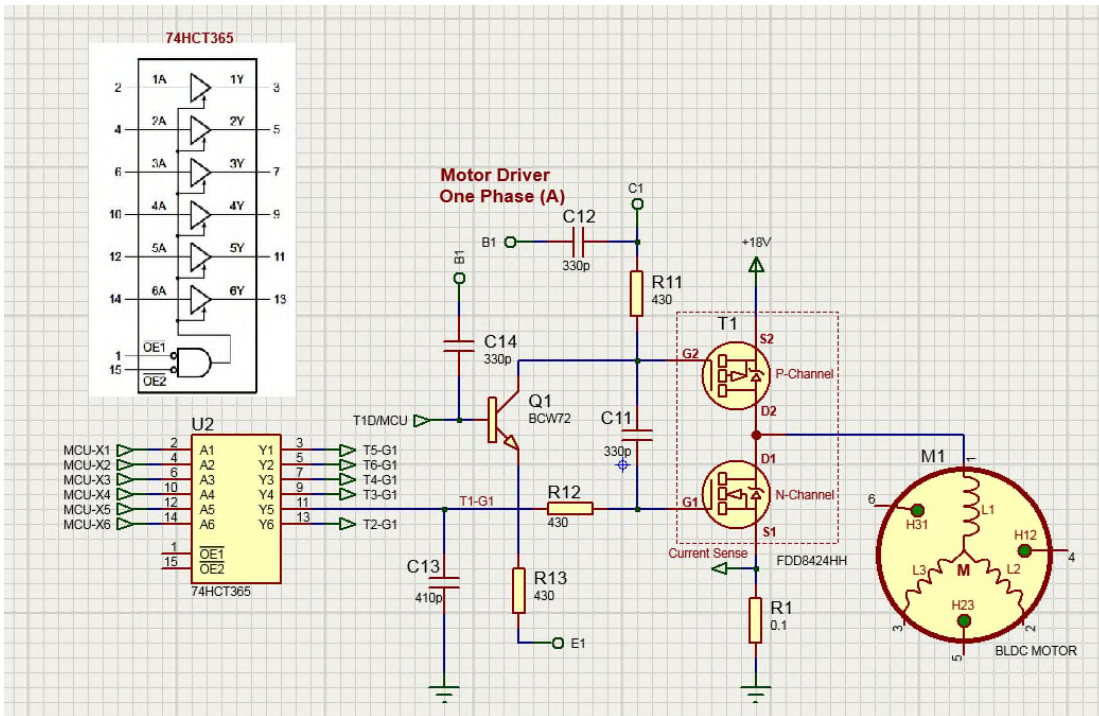


New approach for troubleshooting

We now know that the problem is with the T1 NPN Gate 1 signal, it has the incorrect amplitude, and are just high for the whole commutation cycle. To find the root cause it was discovered that the signal to the NPN gate is generated via a IC 74HCT265.

Here is an updated version of the schematics for one motor phase:





U2 is the 74HCT365 IC that drives the Gate 1 of the transistors of the MOSFET drive stage. We have concluded that there is a fault in the drive signal to Gate 1 (G1) of T1. So the only remaining possibilities are:

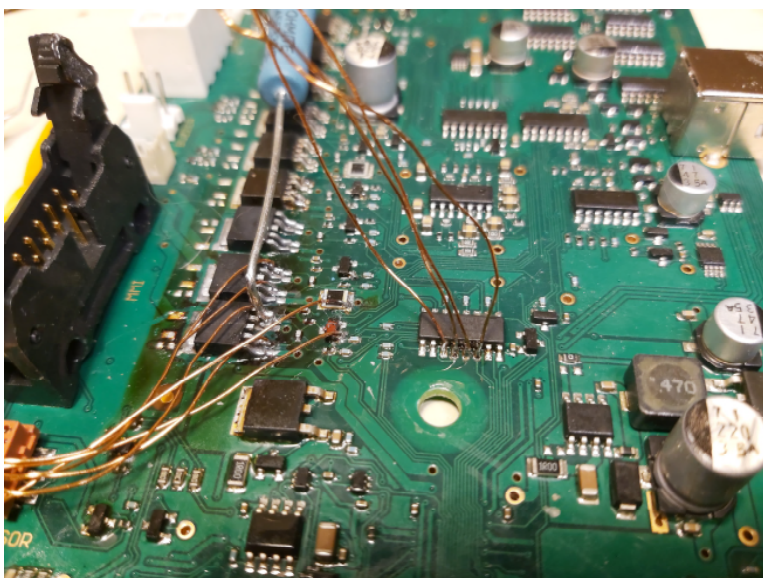
- The 74HCT365 IC is faulty, pin 11 is not delivering a correct signal
- The signal from the MCU, MCU-X5 is faulty
- There is an error in capacitor C13, unlikely because the capacitor seems correct when measured
- R12 is faulty, unlikely because seems correct when measuring the resistor
- Connectivity issues with the PCB traces, unlikely because everything seems OK when measuring the traces

A reconfiguration of the breakout connection points was done as below to be able to check the signal levels of the 74HCT365.

Connection points

Pin #1	Pin #2	Pin #3	Pin #4	Pin #5	Pin #6	Pin #7	Pin #8	Pin #9	Pin #10	Pin #11	Pin #12	Pin #13
T1 G1	T1 G2	T1D base	T2 G1	T2 G2	U2-A5 MCU-X5 MCU signal	U2-Y5 Output 74HCT365 To T1G1	U2-A6 MCU-X6 MCU signal	U2-Y6 Output 74HCT365 To T2G1	NC	NC	GND	GND

Below is a photo of the new connections of the breakout board, in this photo you can also see the location of the 74HCT365 IC circuit:



Case #7, trigger on C2/Auto, motor on left & right

Channel	Pin	Connect to	Identification	Motor Driver	DC Voltage	Motor Connected
C1	6	U2-A5 (12)	MCU-X5	Right	See Osc.	Yes & 33 Ohm
C2	7	U2-Y5 (11)	T1G1 res.	Right	See Osc.	Yes & 33 Ohm

C3	8	U2-A6 (14)	MCU-X6	Right	See Osc.	Yes & 33 Ohm
C4	9	U2-Y6 (13)	T2G1 res,	Right	See Osc.	Yes & 33 Ohm

Case #7-1



Case #7-2



Case #7-3



As we can see from the oscilloscope images (channel 1) above we have the input signal from the MCU to U1 MUX-X5 (A5) and it looks completely correct. However the output from this tristate buffer (Channel 2) is high all the time with some minor noise. If we compare this to the signals to T2, that are correct, we see the difference. In this measurement the test menu was activated and a test run of the wheel motors was performed. Both the wheels was rotating smoothly. However, note that a 33 ohm resistor was connected between the source of the PNP MOSFET and the +18V supply. This to protect the MOSFET to be shorted due to faulty commutation.

So the verdict is that IC U2 is faulty and needs to be replaced. Hopefully this is the only problem and the unit will work correctly after replacement. I hold my fingers crossed.

Final solution and repair

I did change the 74HCT365 IC and removed the 33 Ohms test resistor. To be able to remove the 74HCT365 I made a special tool. The reason was that there was a number of resistor and capacitance components close to the IC. I was afraid that if I used hot air I would disconnect these components due to that the where so close to the IC. With the special tool that I made I minimized the risk for this to happen. The tool that I made can be se in the section below. Thereafter I cleaned up the board and did check for short s and I also did continuities tests. After all that passed OK I once again mounted the PCB in the robot and initiated the motor test as described above. Now both motors run as they should in both directions without any problems. **The problem is solved!** The robot was ready to get back to service. I did not add protective varnish right away. I wanted the unit to be fully teste in operation for a day or so in case of any additional problem. If that works out of protection varnish will be added by the owner.

Desoldering Tool

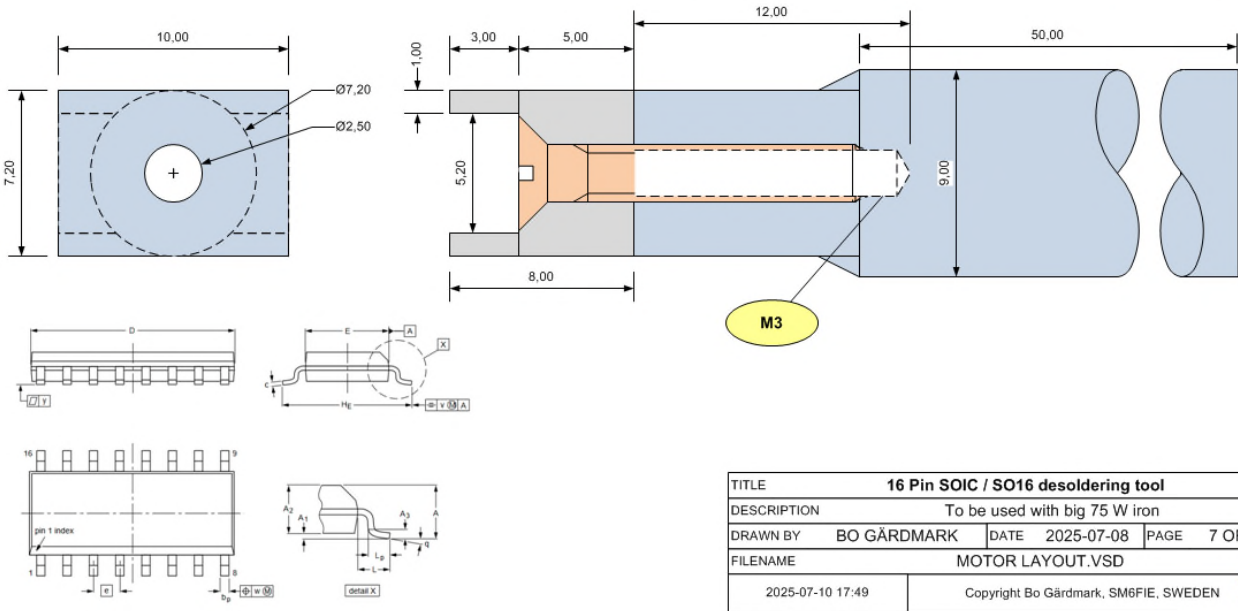
A desoldering tool for 16 pin SOIC package was made in aluminum. This is an very effective and fast method to remove IC circuits from the PCB with minimum efforts. It just takes a few seconds. The method also has minimum impact on surrounding components like resistor, capacitors etc.

Dimensions (inch dimensions are derived from the original mm dimensions)

Unit	A	A ₁	A ₂	A ₃	b _p	c	D ⁽¹⁾	E ⁽¹⁾	e	H _E	L	L _p	v	w	y	θ
mm	max 1.75	0.25			0.51	0.25	10.0	4.0		6.2		1.27	0.2	0.25	0.1	8°
	nom			0.25					1.27		1.05	0.4				
	min	0.10	1.25		0.31	0.10	9.8	3.8		5.8						0°
inches	max 0.069	0.010			0.020	0.010	0.394	0.16		0.244		0.05	0.008	0.01	0.004	8°
	nom			0.01					0.05		0.041	0.016				
	min	0.004	0.049		0.012	0.004	0.386	0.15		0.228		0.016				0°

Note
1. Plastic or metal protrusions of 0.15 mm (0.006 inch) maximum per side are not included.

Outline version	References			European projection	Issue date
	IEC	JEDEC	JEITA		
SOT109-1		MS-012			03-02-19-23-10-27



TITLE			
16 Pin SOIC / SO16 desoldering tool			
DESCRIPTION		To be used with big 75 W iron	
DRAWN BY	BO GÄRDMARK	DATE	2025-07-08
PAGE		7 OF 7	
FILENAME MOTOR LAYOUT.VSD			
2025-07-10 17:49		Copyright Bo Gärdmark, SM6FIE, SWEDEN	

Below is a photo of the tool mounted on a 75 watt soldering iron.



Links:

- BLDC Motor Controller: How It Works, Design Principles & Circuit Examples <https://www.integrasources.com/blog/bldc-motor-controller-design-principles/>
- FDD8424H Dual N & P-Channel MOSFET [FDD842H MOSFET 40V 20A](#)
- STM32F215Z6T6 micro processor (MCU),
<https://www.st.com/content/ccc/resource/technical/document/datasheet/51/9b/66/ba/d4/a8/49/4a/CD00263874.pdf/files/CD00263874.pdf/jcr:content/translations/en.CD00263874.pdf>