Techsol Engineering

TEi :: Review of DaisyStepperDriver.kicad_pro

Last Modified Date: 2023-03-10 BEE

Document Purpose

This document discusses the latest (March 2023) **Stepper Driver board** for Dan Royer's **Marginally Clever Robotics** AKA "MCR".

Document Versions

Version	Date	By	Notes
1.0	2023-03-07	BEE	original
1.1	2023-03-09	BEE	Minor tweaks and updates

Confidentiality

Staff and Contractors of TEi and MCR (who are under NDA) only! DO NOT DISTRIBUTE!!

Related Documents

For additional notes on CAN Bus, see [Techsol Engineering :: OBD CAN OBD2 Resources].

Document Organization

This will be simple:

- Schematic Review, going Block by Block
  - With comments and suggestions at the block level.
- PCB Review, following the same blocks
  - Again, with comments for each block.
  - And also general comments.
- Summary
  - Repeat any critical issues
  - Add some suggestions not specific to this application.

Let's go!

Table of Contents

Schematic Review, going Block by Block

Comments and suggestions at the block level.

MCU: ~~STM32f103~~ STM32F405RGTx

All looks very good! But change the block name (at top) to match the new MCU choice. :)
Could run "CAN_ADDR" as a bus, but .. doesn't matter.

PRO TIP
In the voltage divider for VBUS_DET, put an LED in parallel with the top R (R5).
Unless you use LEDs and RESistors that are the same size (I use 0603 R, and 0805 LEDs … and I always had a reel of each or Red, Green, Yellow on my Pick-and-place machine).
An LED that's ON drops 1.8 - 2.2 volts, which brings 5-V (real voltage from USB is often around 4.7-V) down in the 3-volt range, so that puts the Bus Detect into the HI signal range when connected.
It's handy for debugging and testing (to see if you're connected or not).
In production, you can leave the LED off, and just populate the resistor, saving a nickel.

Stepper Driver: TMC2130

I'm not familiar with this particular chip.
But -- it's driving a stepper motor, right?
Which means that it's driving windings = giant inductors.
Which means giant, negative spikes when they are turned off.
I see "Snubber diodes" on MOTOR_Ax/Bx, which is great.
What I don't see is noise protection on the power.

More info in the PS section, below.

PS: 3.3v LDO

Looks good!

Comments:
I see that PG is not connected to anything. For systems where a Power Good signal is available, I've connected that to the MCU's /Reset pin. Then, if the power drops you can force the MCU to restart cleanly, avoiding a brown-out condition. Use an RC network to de-glitch any micro-power-faults (that the bulk CAPs will handle)
NOTE: if you've dealt with brown-outs on an MCU that doesn't handle them well, you know how much of a headache that can be … and service call-outs to have someone go pull the plug on a unit to force a clean boot, after it's **locked-up** from a simple power dip.

PRO TIP:
On designs where I have DIGITAL (MCU), ANALOG (op-amps, etc.), and/or POWER CONTROL such as motors, I always create a power bus for each of them to avoid cross-contamination from noise.
This doesn't have to be elaborate or expensive:

- SIMPLEST: separate 3.3-volt power buses, daisy-chained with ferrite beads.
    - Run the LDO output into a CAP and feed either:
        - the largest load
        - or the one needing the cleanest supply, such as VDD_Analog
    - Then, feed another supply off that one, but through a ferrite bead, and add another bulk capacitor, in addition to smaller ceramics
    - So, if power needs are similar, you could run Analog, then Digital, and finally Power from your PS
        - (*Later on, I saw that this is done for 3V3A beside the MCU. Very good!*)
    - The noise from Power has to get through 2x ferrites to get to the Analog power :)
    - But if Power is a higher draw, consider the 2nd SIMPLEST:
- 2nd SIMPLEST:
    - Run a couple of LDOs off your primary (5V) supply.
    - You're adding the cost of a second LDO, but it will be smaller and cheaper than the one feeding your noisy power devices.
- Remember to segregate the parts on these buses into groups on the PCB placement, so that you can run different power-planes on your inner layers as well, to reduce cross-talk of noise.

BOM Optimization:
(this depends on the total power draw, and "old school USB power" is limited to only 500 mA) but the 2 power-sharing Schottky diodes can be replaced with a single SOT-23.
Which makes sense if these are the only Schottky diodes.
But, if you've got more, then it makes sense to use 2 more of the same here, and avoid procuring/loading yet another component onto the P&P machine.
Although (on the other hand) the dual-diode packages can also be used as just a single diode too.
I used to pay about 1.3 cents for SOT-23 diodes, regardless of the configuration.

Induction Sensor: IPS2200

This is cool! I've never worked with a chip like this before. In fact, I didn't know that they existed!
I see that it has a VDDA going to a CAP, so I presume that it's ideal to not have a lot of noise going into that chip.
And I see that it connects on IIC AKA I2C bus. Which is directly connected to the MCU.
I would put series resistors on the I2C lines to give them some impedance and dampen the sharp edges of the signals, especially the I2C clock. Otherwise you have a high dI/dt on the front edge, plus possibly harmonics, on those little bonding wires in the chip, which can inductively couple noise onto your analog pickup lines. (on their tiny little bonding wires)
I'd throw 1k max on them, or less. Something already in the BOM.
I used to use the 470 R resistors also used for LEDs, but I think you've got 1k3 on the LEDs, but with 2k2 pulling up those lines, a series resistor should be even lower impedance. Any 470 or 330 ohm resistors on the board? 270? 100?

<u>MODULAR "Design Re-use" TIP</u>:
For the I2C lines, put the pull-ups at the master (the IC driving the clock signal) always.
Same if you add small series resistors too. Also place them at the MCU block.
Then you never have to worry about "which peripheral chip has the I2C pull-ups?" and since your peripheral blocks NEVER have the pull-ups, now those blocks are fully portable and re-usable from one design to another.

CAN Bus Driver

Dan, **there's nothing wrong with the existing design**. The circuitry looks correct "as is".

<u>Driver chip choice</u>:
The old-school "single supply" CAN driver chips were 5-V only. .
Those are from the days where all electronics were 5-V.
Next came 5-V Bus drive with 3.3-V logic.
Now there are more 3.3-V only chips. And this is where CAN has diverged from the RS-485 drivers that it started from. In 485, the target was to get a full 5-volt swing on the twisted-pair lines. But, of course, that limits the speed of operation.
CAN is now operating on a smaller voltage differential, which is compatible with 3 volt only systems.

the BEST CAN Driver IC, with 3-V Logic and 5-V Drivers, from Philips/NXP

My original notes on CAN DRIVERS were based on "the best design at the time" almost a decade ago.
Since then, 3-volt CAN drivers are much more plentiful.
So **I'm moving all the CAN driver discussion into an Appendix**.
But one of **the biggest arguments in favour of retaining 5-volt Bus Drive, is noise isolation**.
In the Dominant signaling mode, 2 enhancement-mode FETs in the driver are on - the P-Channel connecting the +Bus power to one wire, and the N-Channel connecting the other wire to GND.
All of the noise on those CAN Bus wires is directed straight onto your supply. You choose which supply.

USB Connector

Looks perfect for the basic connection!
But kinda minimal for performance. Both power and noise.
I always had a tiny CAP across Vbus to GND right at the connector.
Then V-bus and GND-bus EACH go through a ferrite bead to block external noise from coming in (from a PC, etc) and also to block internal noise from going out (essential for FCC and CE emissions approvals).
Then I connected a bulk capacitor to Vbus. Typically a 33uF SMT CAP. (using Niodium Dioxide caps @ U$0.073 each, rather than a Tantalum cap costing several times that).
AFTER the CAP, then Vbus can be connected to other pins.

MISC: Connectors, PBs, SWitches

Everything looks good.

PCB Review, following the same blocks

Again, with comments for each block.

MCU: ~~STM32f103~~ STM32F405RGTx

Looks good.

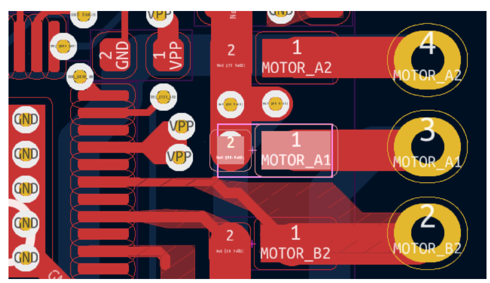
BOM QUESTION: Why the 4-pin crystal?
I usually only see 4 pins on an oscillator.
Crystals are usually 2-pins for low-cost (Abracom, etc)
I buy direct from the California office of an Asian manufacturer of crystals.
I have reels of 8, 12, 24, MHz SMT crystals. (the 8 MHz is HC-49S short can. The others are 5 x 3mm.)
Some others too, but if it's an MCU with a USB port, it normally uses a 12 or 24 MHz crystal, with an internal PLL generating the 48 MHz internal clock for the 12 MHz "Full speed" USB communications circuitry.

Stepper Driver: TMC2130

Looks good! Nice use of "flood" or "area fill" to beef-up the motor drive traces.

1 PRODUCTION ISSUE:

- D7 and D6 connect back to a "sense pin" BRA on the driver.
- On that net, there are 3 power-vias connecting the top traces to the bottom traces carrying the signal back to U4-44.
- **One of those 3 vias is in the middle of the pad of D7**.
- This presents 2 production problems:
    - When screening solder onto the board, some of the solder for the pad can go into the hole, leaving less solder than desired for the connection on the pad
    - In the reflow oven, the via provides a direct thermal connection to the large traces on the bottom, which can suck out the heat, preventing the pad from getting hot enough for proper solder reflow



- These issues, alone or together, greatly increase the possibility of a cold solder joint underneath D7. (via is mostly hidden under the left side of the diode in the image above)

RESOLUTION:

1. Move the PAD out from under D7, and maybe to the right of the other 2 vias on that net.
2. Re-route the top and bottom layers accordingly.
3. Regenerate the internal layers to create clearance around new via.

PS: 3.3v LDO

Very nice!
Fat traces!!

Induction Sensor: IPS2200

No issues. I've never done circles or large ARCs on a PCB with KiCad.
It looks like something that probably took a LONG time to lay out!

CAN Bus Driver

Very nice.
Traces are beefy where they need to be.
Nice use of internal layers for routing "beyond" where the power planes are needed.

USB Connector

The "Micro B" USB connector is one of the worst technology disasters of the past few decades.
At least, the "fully SMT" variants found on most products.
Every connector that a client or user can access should be Pin-Through-Hole (or PTH).
For test gear, I use a full-size USB-B with through-hole pins, and have had boards in service (on production line test) for over a decade with no failures. But the "Micro B" … not so much.
HOWEVER -- **you found the rare Hybrid with SMT signals, but PTH shell connections.**
You don't know how happy that makes me!! 😘

MISC: Connectors, PBs, SWitches

All SMT and switches on top: Check!
SWD pins, on 0.100" connector spacing, on bottom for "bed of nails" or "Pogo Pins" production test / programming jig: Check!
All good.

General PCB comments.

Very nice work! :)

<u>SUGGESTED IMPROVEMENT</u>:
Power vias (on motor and snubber diode connections) appear to be missing thermal reliefs to internal planes.
<u>CORRECTIVE ACTION</u>:
Design or select a power via with thermal reliefs and swap it in, replacing the existing ones.

Summary

Repeat any critical issues

ISSUE	DESCRIPTION	CORRECTION	VERSION / DATE CORRECTED
1	VIA under D7	move via and re-route	
2	missing thermal reliefs on vias	build a pad-stack for the vias with relief	
3	CAN drive only 3 volts	switch to newer driver	

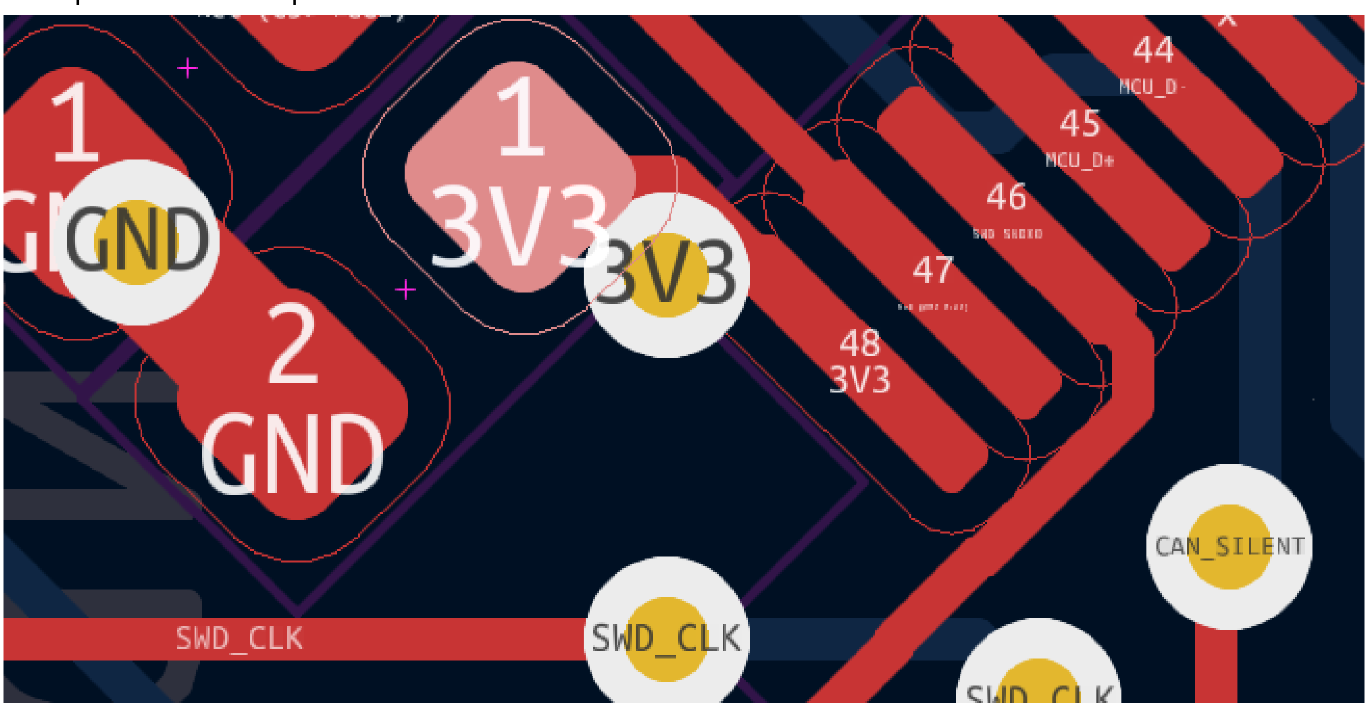
General suggestions not specific to this application.

I have a lot of little tricks to make boards … "more resilient", to use a popular word.
But, I don't have enough time now.
I might have an internal design guide to share (in my thousands and thousands of Google DOCs).

1 little detail that helps reduce noise getting INTO an IC, and also filter noise coming FROM the IC:

- Always run the power-pins of the IC to the bypass CAP first.
- THEN run the trace to the power source
    - such as a via to an internal plane for multi-layer boards.
- This isolates noise on the power-plane from the IC.

Example of where the power comes in BETWEEN the IC and the CAP: MCU Pin 48



If the CAP is moved closer to the TQFP, then there is room for the VIA to go on the other side of the pad.

Appendix 1: CAN Bus Driver notes and discussions

Brian's notes from "best design practices" circa 2018-ish.

A small chip (SOIC-8 package) with a LONG number: TJA1051T3CM118 (on Reel of 2,500)

TJA1051T/CM,118

TJA1051T Series 5.5 V 5 Mbit/s Surface Mount High-Speed CAN Transceiver - SOIC-8

This has 2 power pins:

- "Vcc pin" which is the digital side, which is typically 3.3v, but maybe could be 2.8 or 3.0v if you have a low-voltage micro or FPGA driving it.
    - It's rated for 2.8 to 5.5 volts
- "vIO pin" is the CAN-Bus power and typically 5 volts, so you're getting more drive than from a 3.3v chip
    - It's rated for 4.5 to 5.5 volts

It has fail-safes ... if you drive it active too long it shuts-down.

Which means it won't work as a UART line driver below 40k baud.

We use them for both CAN Bus and "RS-485" networks, but have to run 57k baud to avoid protection shut down. This is the driver chip recommended by ARROW Electronics, largest distributor of electronic components on the planet. Modern cars have dozens of these.

BONUS: This chip also includes 8kV of static protection rated IEC 61000-4-2 at pins CANH and CANL.

<u>PINOUT</u>:

NOTE: Digital is on the LEFT of the chip.

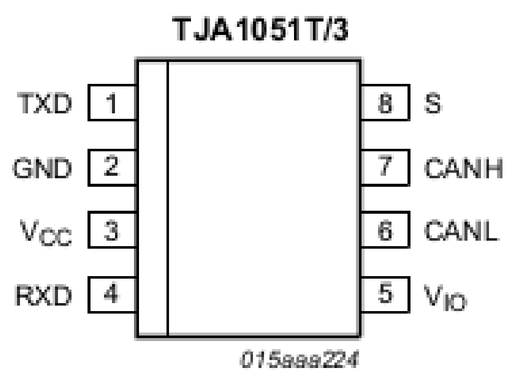
ANALOG (CAN Bus) is on the right.

Except 'S' = Silent = Shut-down TX (low-power).

Vcc = 3.3v

Vio = 5.0v

Easy-peasy! And they're pretty much bulletproof.



c. TJA1051T/3: SO8

Fig 2. Pin configuration diagrams

<u>DATASHEET</u>: https://www.nxp.com/docs/en/data-sheet/TJA1051.pdf

I used to pay $0.45 @ 1k from Arrow.

You must get the "/3" part to get the 3-v digital supply and 5-v bus. Only 1 Megabaud speed, though.

Notes from the Field

IF you were running a system with LOTS of CAN devices, over long cables, in a very noisy environment (like a semi-tractor truck, or worse, a train engine) then I'd want the extra muscle of the 5V system. Though I can't tell you for certain that it would be better, because I've never tested 5V vs. 3V CAN bus.

I did do a LOT of communications with RS-485 systems, including credit card payment terminals at Whistler (with devices on both sides of the mountain connected by vintage phone cabling, but all on the same network) and airport parking systems, such as Ottawa Airport, where the "main lot" by the terminal was wired to the "economy lot" a few kms away, and again, all one single network with over a dozen devices connected.
And **CAN drivers are just RS-485 drivers with a wiring optimization**.
The early CAN BUS networks took an RS-485 driver, tied the TX pin to GND, and then fed the UART's TX pin to the /EN enable pin of the driver. Thus, the bus is driven when the start-bit comes out, and all other "0" bits after, but goes passive/recessive state on the stop bits and idle time.

CAN Bus Design Detail: Bus Termination

In those systems, we were pressured to move cars through the payment lanes as fast as possible, because airport traffic is very "lumpy" … a big plane or 2 come in, and 40 minutes later half the vehicles are exiting the parking lot, seemingly at once! Our system, in the 1980s, could process a car through a lane in under 50 seconds, and that includes the credit-card payment authorization, which took less than 1 second, but operated over that twisted-pair RS-485 network, with 4 messages between the "network master" and the payment terminal (a "fee computer" which is just a cash-register with specialized SW to calculate charges based on entry and exit times, and with extra-fast, higher volume/life-span printers).

One of my targets in the design was that all units had to be identical, in order to facilitate easy servicing. If we had to have termination resistors on units at the end of the bus, it would be a nightmare when people moved stuff around, whether trouble-shooting, or actually reconfiguring the parking lots.
So, imagine your typical network with 20 units on it. (the RS-485 spec goes up to a maximum of 32 units. I'm not sure what the limits are for CAN).
OLD SCHOOL:

- each end has a 120 ohm resistor, creating a net shunt across the wires of 60 ohms.
- there are no resistors from the data lines to the power bus anywhere

NEW SCHOOL: (after testing on many installations across North America, and deploying equipment to Europe, South America, and Australia)

- If you wanted to distribute the load resistors (termination resistors) across 20 units, you would put 20 * 60 ohms at each unit.
  - so all 20 in parallel present the same 60 ohm "bus load" to the drivers
- So, putting a 1.2k ohm resistor on EVERY unit puts you at the same load with 20 units.
  - at 5 units, you only have 240 ohms across the wire pair
  - at 30 units, impedance drops to 40 ohms … starting to push the drivers' current

capabilities … at least if one of them is transmitting near continuously, which is not the case with a CAN Bus.

- In addition, I added a PULL-UP and a PULL-DOWN on the comms wires, to force the recessive (idle) state. Not super strong, but they were there.
- So, my BUS INTERFACE was a totem-pole of 3 resistors from +5 to one wire, to the other wire, and to ground.
- And, I had small CAPs across the CAN driver too, just to de-glitch noise spikes.
  - They were very small … might have been ones we used on crystals too

The NET EFFECT is that I had systems that were deployed and operated for years, in the presence of lots of "industrial noise", for years at a time, without being reset. And on the twisted-pair-bus, EVERY unit was the same!

Driver Sourcing and Availability

The **TJA1051T/3/1J** is available from lots of places, but not the largest distributors.
I suspect that they are sucked-dry supplying their automotive clients.

From the datasheet, the T/3 is the one with 3v logic, apparently:

Functional description

The TJA1051 is a high-speed CAN stand-alone transceiver with Silent mode. It combines the functionality of the TJA1050 transceiver with improved EMC and ESD handling capability. Improved slope control and high DC handling capability on the bus pins provides additional application flexibility.
The TJA1051 is available in three versions, distinguished only by the function of pin 5:
• The TJA1051T is backwards compatible with the TJA1050
• The ***TJA1051T/3 and TJA1051TK/3 allow for direct interfacing to microcontrollers with supply voltages down to 3 V***

In today's wacky world, I would lean towards using parts that are super available from a LOT of suppliers. The **Texas Instruments SN65HVD230DR** was not available before June in 2022, but since then has had good availability.
You can buy direct from Ti, who have > 222,000 parts in stock (today's numbers)
https://www.ti.com/product/SN65HVD230/part-details/SN65HVD230DR
Digikey has only 803 in stock, which is worrying, because that might be zero tomorrow morning.
But Mouser have a lot more. (almost 91k parts)
Lots of distis carry this part, and total inventories are running over 2 million parts now.
Average price is U$1.445

Going back to the NXP parts,
TJA1051T/3/1J is a 5 Megabit part, average cost U$0.78, 1.2 million parts in warehouses
TJA1051T/3,118 is a 1 Mbps part (like the T.i. part) costing U$0.76, with 575k parts available
DIFFERENCE: both run a 5-V CAN bus. But T/3 runs a digital side down to 2.8 volts.
For the T/3/1J parts, the digital side operates down to 1.3 volts.
So, if you're running your automotive engine controller with a 1.8-volt core, you can wire directly to the CAN driver, without the additional circuitry in your MCU for the 3.3v IO pins (which also add a small timing and power penalty).

Here's something cool: The **Texas Instruments SN65HVD1050DR** is a possible drop-in for the TJA1051T/3,118 but at a higher price (U$0.96) however, it operates from -40c - 125c, and there's over 2.9 million in warehouses!
Actually, it was presented to me as a replacement, but its a 5-V only part.
<u>WHY NOT USE IT</u>:

- Its 5-V power only with a 3.3-V MCU.
  - The MCU to TX will be fine, because 5-V electronics operate more on TTL levels
    - LO is anything under 0.8 volts.
    - HI is over 2.8 v, so its 3.3v CMOS MCU compatible.
  - The driver to RX looks like a problem, like you'd need a voltage divider
    - you can do that, but you can also just put a series resistor between Driver RX and MCU-CAN-RX, and let the I/O pin's over-protection deal with it (though, its likely a TTL high signal, so no issue).

<u>WHY USE IT</u>:

- Supply noise: its on 5 V, which has a benefit that its drivers are not throwing noise onto your 3.3v bus (just the 5v bus, which is typically power devices)
- Its really cheap direct from Tex Ins direct:
  - Q=100 its under U$0.65
  - **Q=1k its about U$0.406** !!!
- 3 million parts in inventory (across many distributors)