UPGRADE TO 3D PRINTABLE JET ENGINE - TURBOFAN DRIVER WITH MAGNETIC COVER

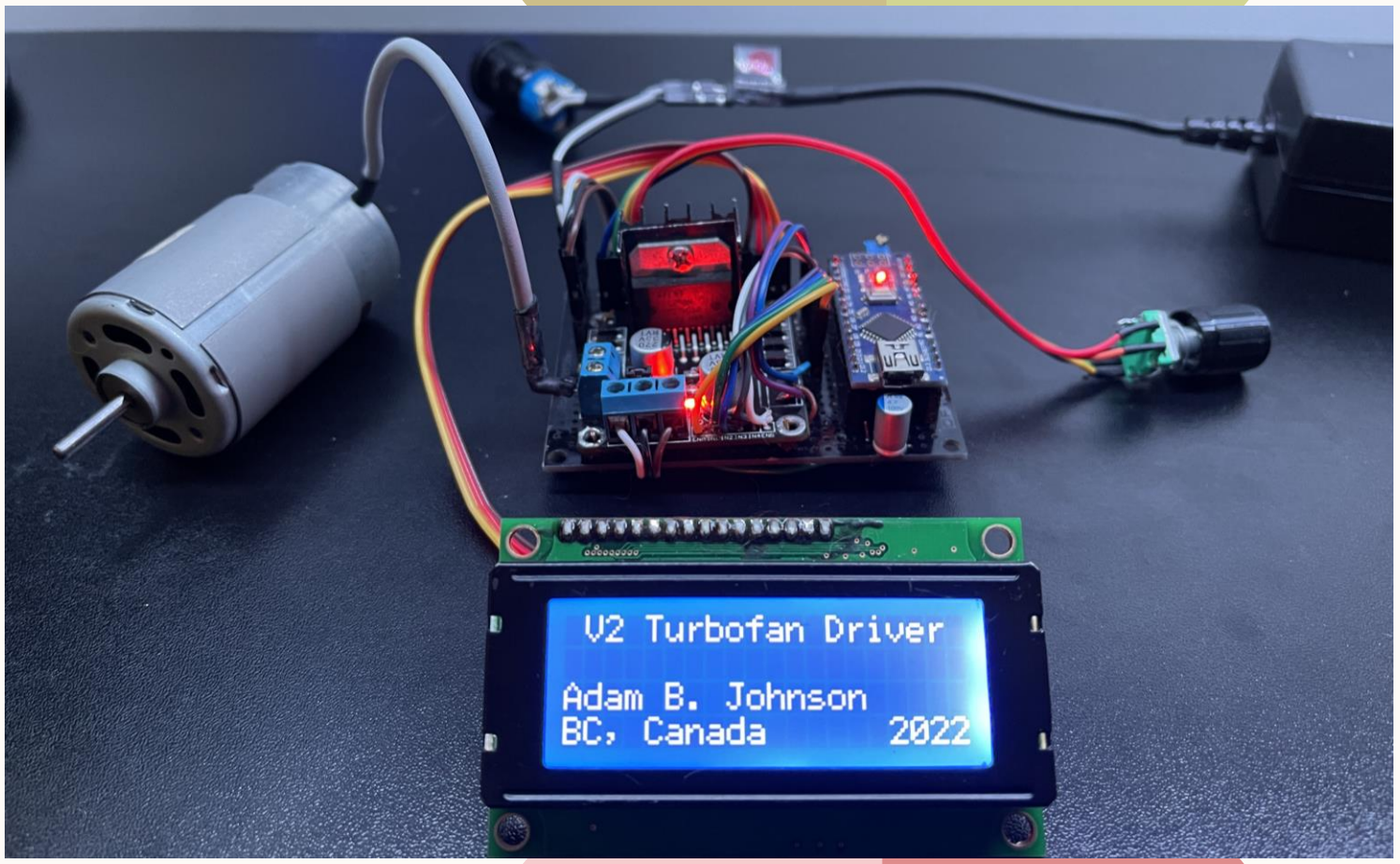
V2

NOV 2022

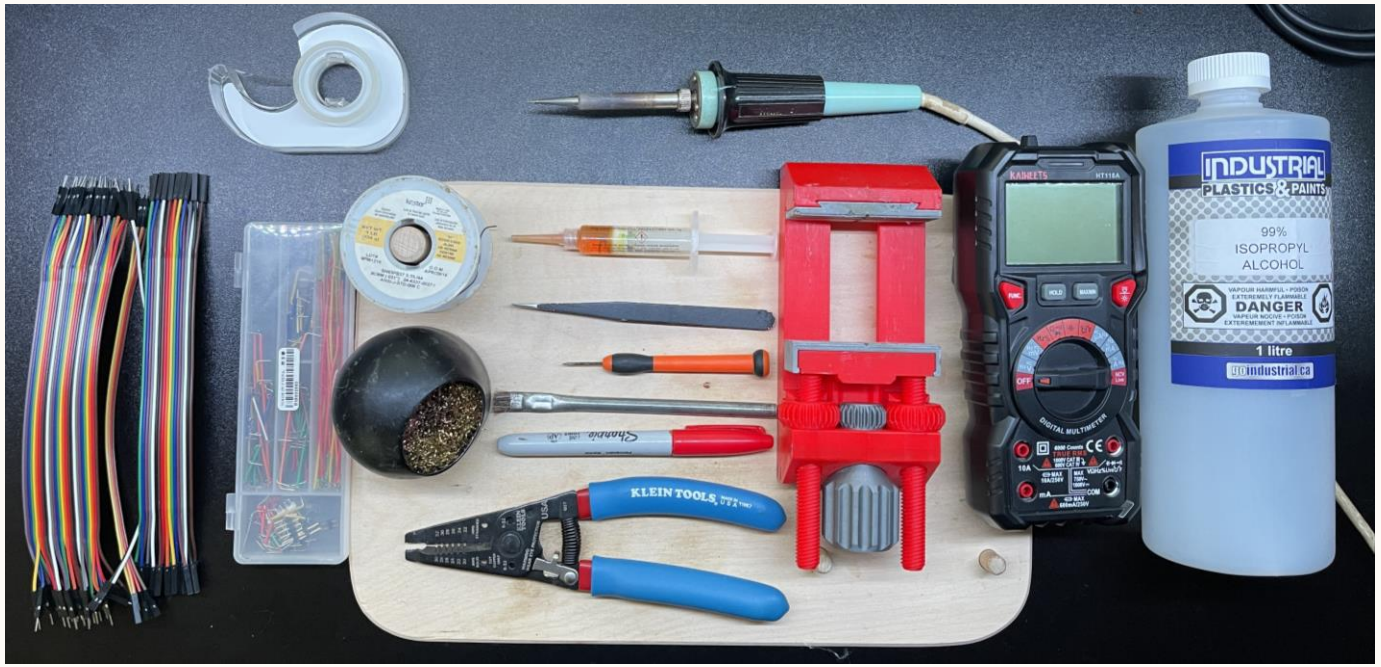
Electronics Assembly Instructions

By Adam B. Johnson

TABLE OF CONTENTS

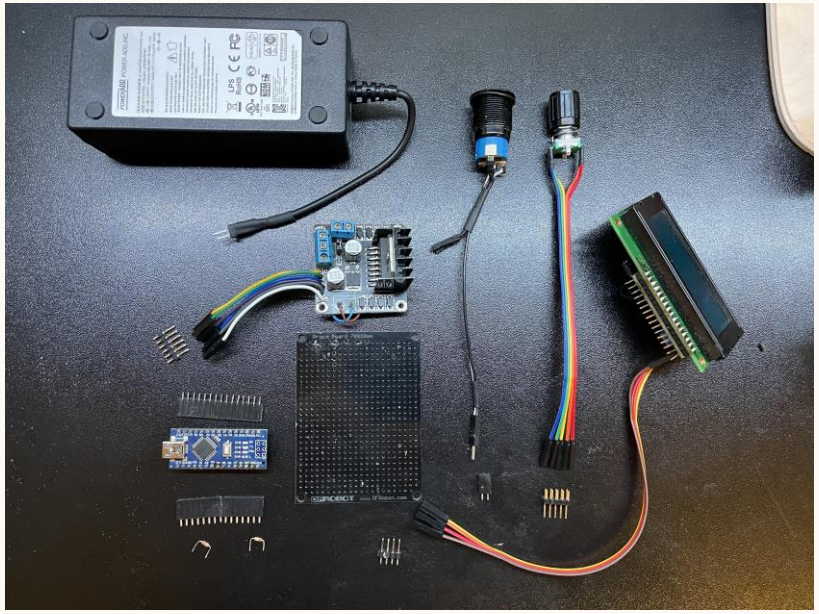


THINGS YOU'LL NEED



Required Equipment

Multimeter with voltage and continuity settings
Soldering iron with solder and flux
Isopropyl alcohol
Small brush
Mini-USB cable for Arduino programming
Wire strippers 16-22AWG
Side cutters
Fine pointed knife
clear tape
red sharpie
GitHub repository: https://github.com/nairck/TurbofanDriver
Control Schematic V2:
https://github.com/nairck/TurbofanDriver/blob/main/Control%20Schematic%20-%20V2.pdf
Arduino Code: https://github.com/nairck/TurbofanDriver/tree/main/Control_Sketch_V2



Off-the-shelf Hardware for Turbofan Driver

1x Arduino Nano - https://www.amazon.ca/Arduino-ELEGOO-ATmega328P-Compatible-Without/dp/B071NMD14Y/ref=sxts_sxwds-bia-wc-rsf-lq2a1_0

1x L298N Motor Driver - https://www.amazon.ca/Neuftech-H-Bridge-Stepper-Controller-Raspberry/dp/B01KBTNHS6/ref=sr_1_3

1x RS-555SH 12V Brushed DC Motor - https://www.parts-express.com/Mabuchi-Type-Motor-RS-555SH-12V-DC-Motor-9-15V-57mm-x-36mm-125-566?quantity=1

1x 20x4 2004D rev. C LCD Screen with I2C module - https://www.buydisplay.com/small-size-arduino-code-lcd-20x4-i2c-character-display-wide-view-angle

1x EC11 Rotary Encoder with Button - https://www.amazon.ca/Cylewet-Encoder-Digital-Potentiometer-Arduino/dp/B07DM2YMT4/ref=dp_prsubs_1

1x 120VAC 19mm Latching Power Button - https://www.amazon.ca/gp/product/B017KP67FY/ref=ppx_yo_dt_b_search_asin_title

1x 58mm x 78mm Proto Board - https://www.robotshop.com/ca/en/prototyping-board.html?gclid=Cj0KCQiAmL-ABhDFARIsAKywVadrBUguidr9uNXcGYmD8eJ5VqHiz2U_HtVTzj7YqXQUhuB1ejnLW4MaAnh7EALw_wcB

22AWG male and female jumper wire – multiple colours. - https://www.amazon.ca/gp/product/B01M1IEUAF/ref=ppx_yo_dt_b_search_asin_image

Breadboard Jumper Cable Wire Kit Jumper Protoboard Cable - https://www.amazon.ca/140pcs-Solderless-Breadboard-Jumper-Shield/dp/B07D32FDMZ/ref=sr_1_4_sspa

1x 10k potentiometer - https://www.amazon.ca/Uxcell-a14052600ux0954-Trimmer-Potentiometer-3296W-103/dp/B00SWJZMM2/ref=sr_1_32

1x 10k resistor – could use a second 10k pot (as pack comes with multiple, or single 10k resistor)

2x 15, 1x 3, 1x 2 socket female header (2.54mm pitch).
1x 6, 1x 5, 1x 4, 1x 2 male header pins (2.54mm pitch).
https://www.amazon.ca/DAOKI-2-54mm-Female-Connector-Arduino/dp/B019WOPOHI/ref=sr_1_2_sspa

Variably sized heat shrink - https://www.amazon.ca/730pcs-Shrink-Electrical-Storage-Portable/dp/B09XXG45Q7/ref=sr_1_1_sspa

1x 120/240VAC to 12VDC Transformer, 4A (wall plug included but detachable) - https://www.amazon.ca/COOLM-Transformer-Doorbell-100-240V-5-5x2-5mm/dp/B07D5BX22W/ref=sr_1_2_sspa

4x 0.25in Square Neodymium Magnet (only the shape is important) - https://www.sparkfun.com/products/8643

35x M3x0.5, 6mm Long Pan Head Screws – I replaced most M2.5's with M3's. - https://www.amazon.ca/Glarks-230Pcs-Stainless-Button-Assortment/dp/B01J7NM9JA/ref=sr_1_2_sspa

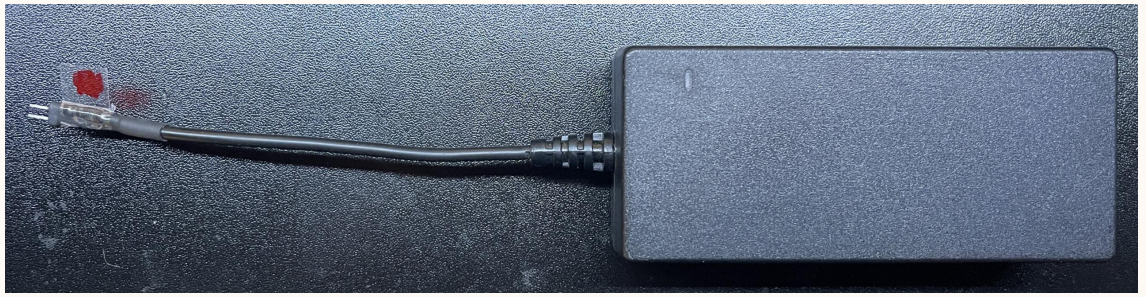
1x M5x0.8, 10-15mm Long Socket Cap Screw (hopefully you can source 1x this instead of buying a whole kit) - https://www.amazon.ca/Glarks-150Pcs-Stainless-Button-Assortment/dp/B01J9865L6/ref=sr_1_8

Optional – 4.7µF electrolytic capacitor (see extra slide at the end) - https://www.amazon.ca/Electrolytic-Capacitors-Appliances-Communication-Electronic/dp/B086VLGRR2/ref=sr_1_7

CABLES TO MAKE

DC motor

Power supply

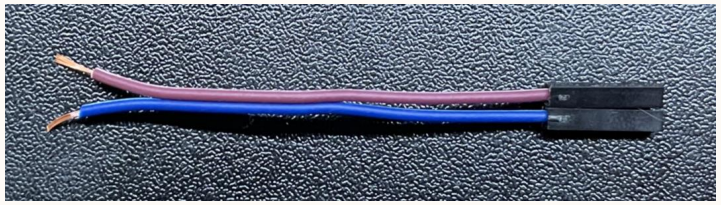


Power button

Encoder/button

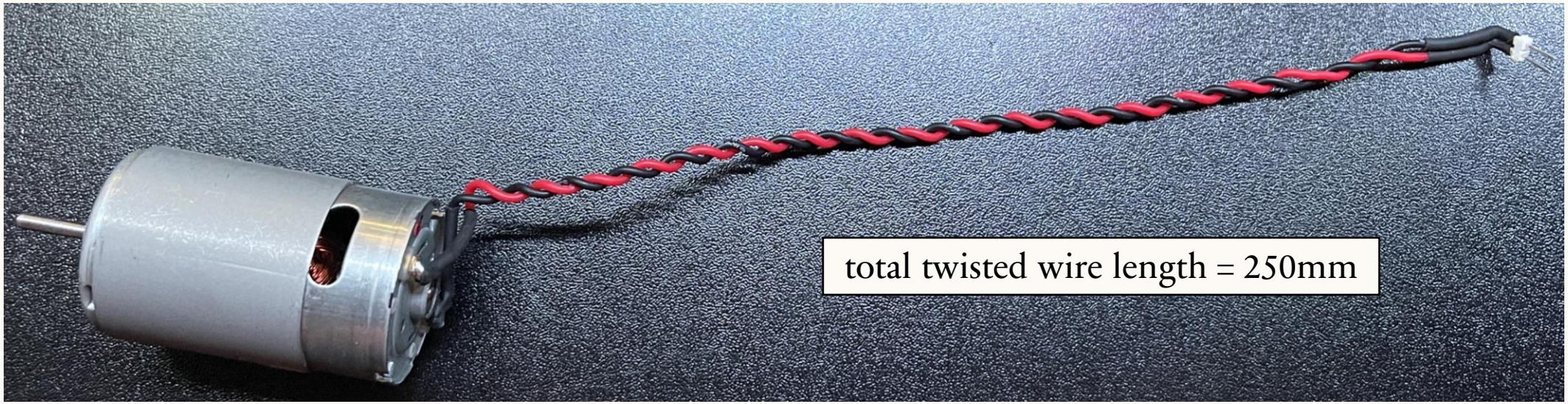
LCD – 4x female - female header
(150mm unmodified)

Motor driver cables



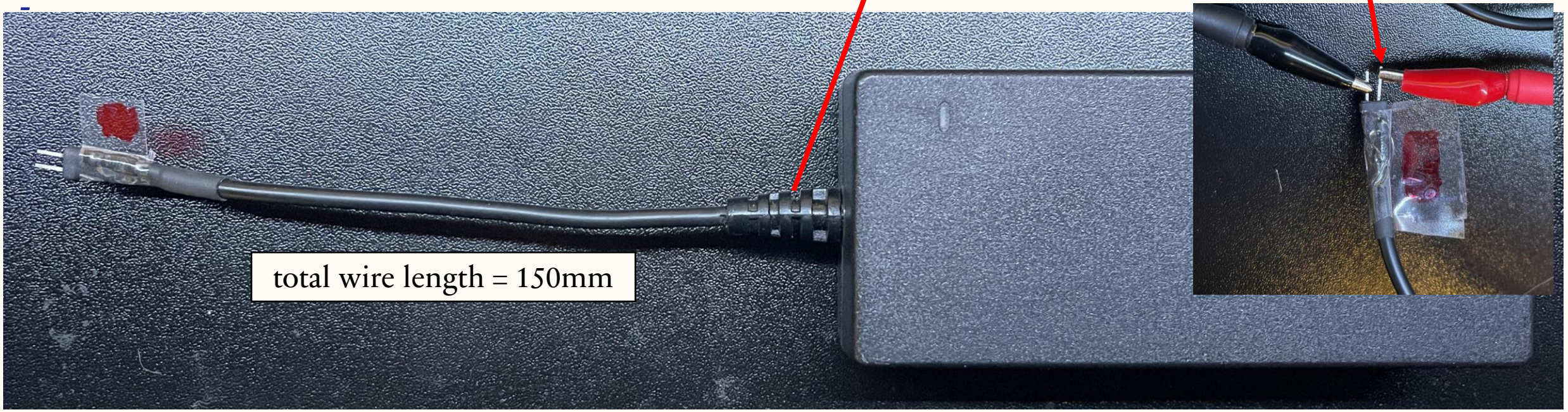
CABLES TO MAKE – DC MOTOR

- Heat shrink all connections
- Twist RED and BLACK wire to get the final 250mm length
  - Using Silicone insulated wire can help longevity of cable
  - Using a drill for twisting can help (YouTube)
- Solder pins sideways as shown side to make final installation easier and maintain the same driving polarity



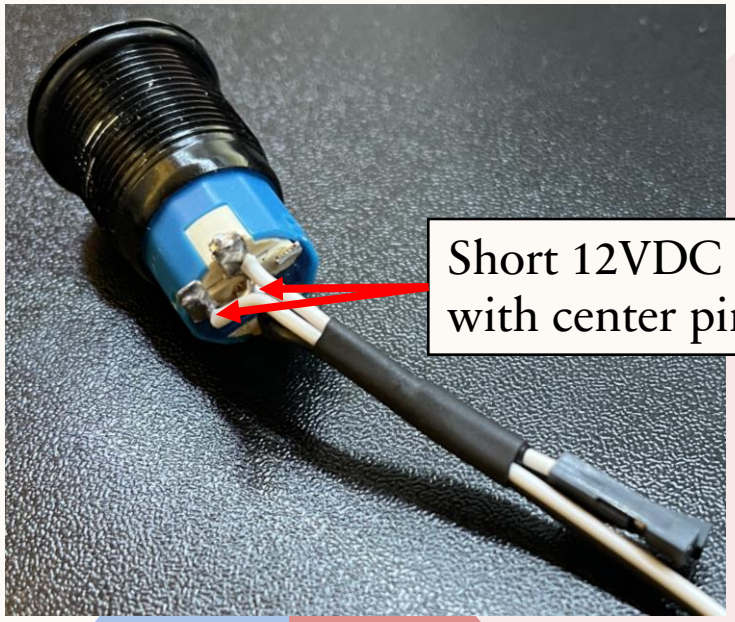
CABLES TO MAKE – POWER SUPPLY

- 2x male header soldered onto transformer output (12V) line
- Carefully trim off strain relief
- verify +12VDC with multimeter and mark +12VDC side with tape and red sharpie

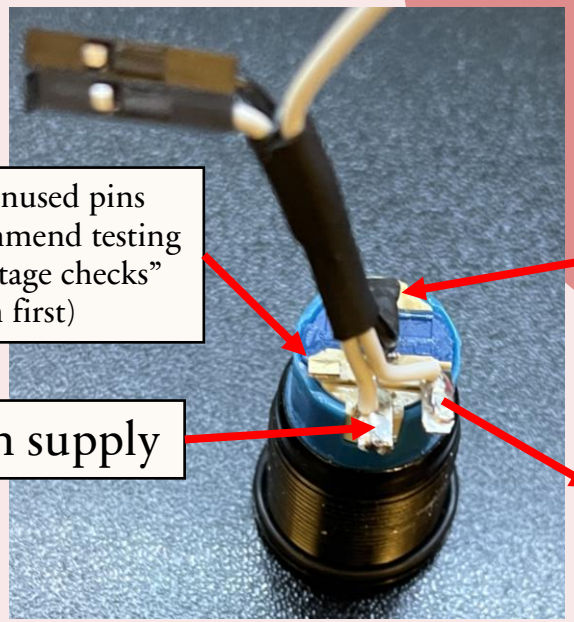


CABLES TO MAKE – POWER BUTTON

Main idea with this button is that the LED is always powered, but the switch opens and closes the 12VDC circuit to the rest of the electronics



Short 12VDC IN with center pin

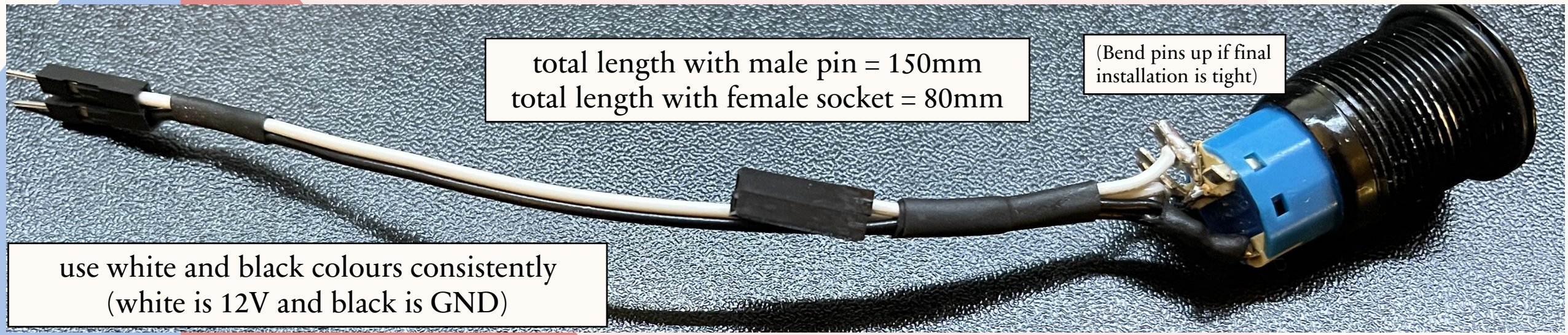


Trim unused pins (recommend testing in "voltage checks" section first)

Black "GND" wires together

12VDC IN – from supply

12VDC OUT – to motor controller



total length with male pin = 150mm
total length with female socket = 80mm

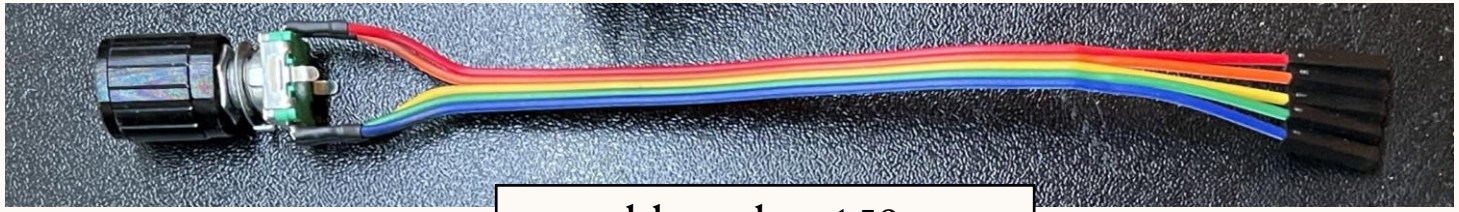
(Bend pins up if final installation is tight)

use white and black colours consistently (white is 12V and black is GND)

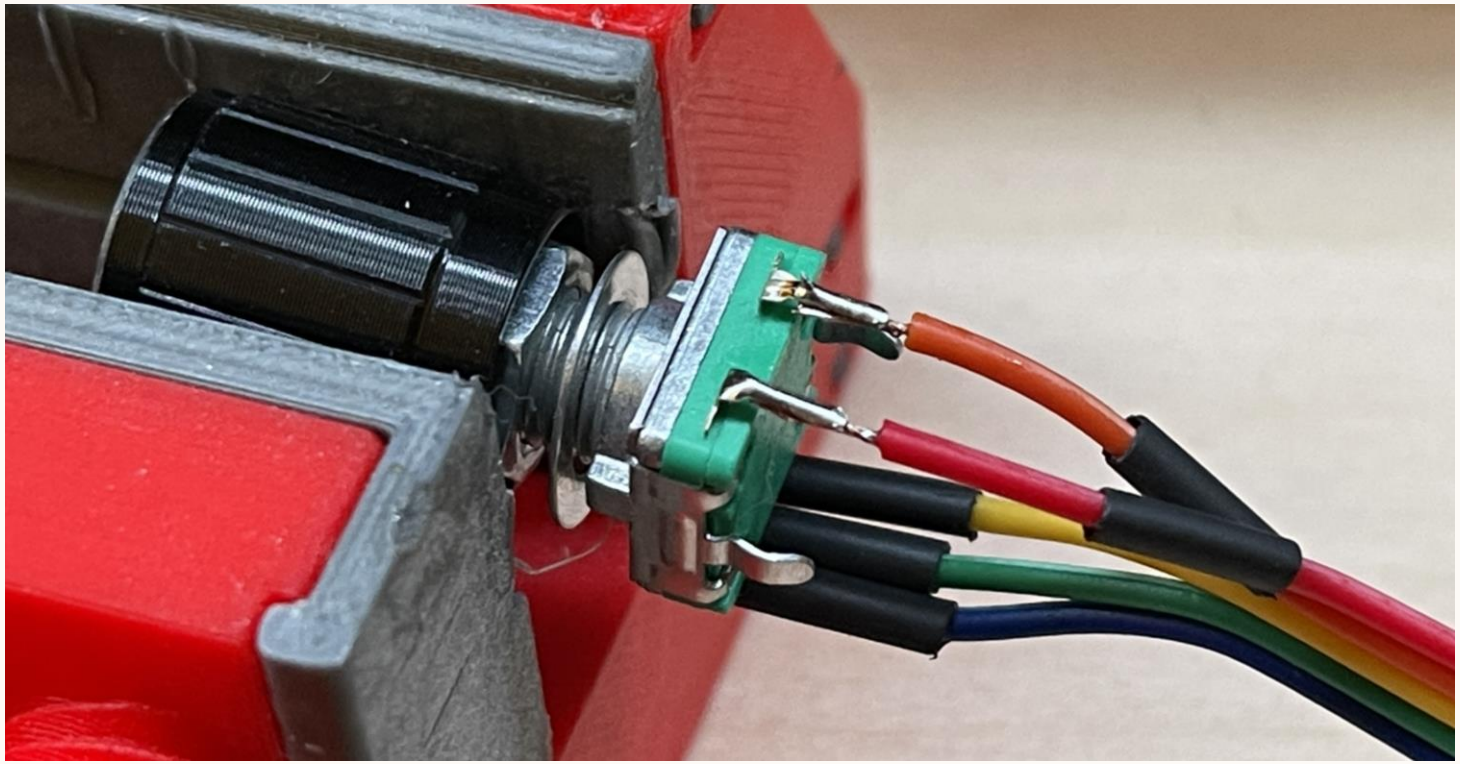
CABLES TO MAKE – ENCODER/BUTTON

1 – BLUE – CH_B Encoder – D2
2 – GREEN – GND
3 – YELLOW – CH_A Encoder – D3
4 – ORANGE – Button – D4
5 – RED – GND

- Place 10mm of heat shrink over each pin

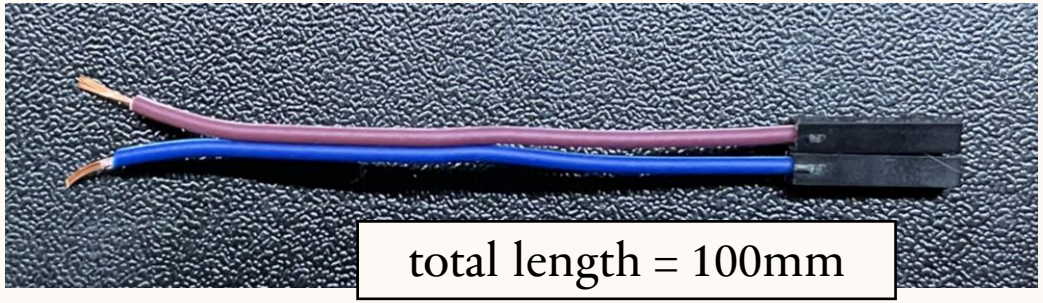


total length = 150mm



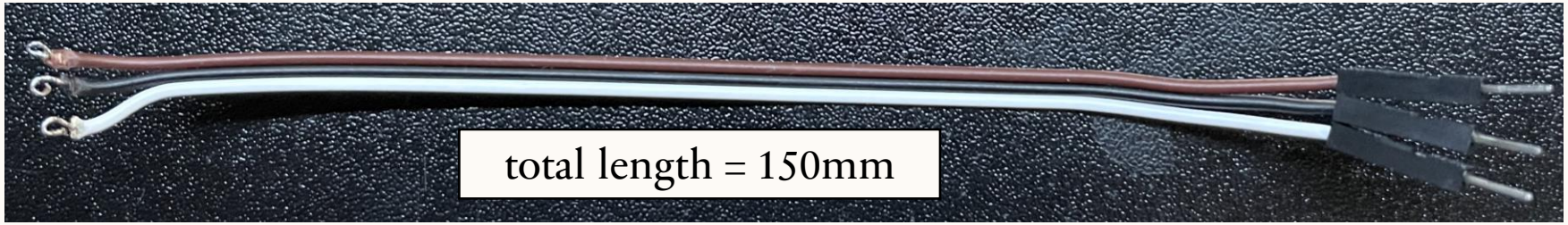
CABLES TO MAKE – MOTOR DRIVER

1 – PURPLE – OUT4 / OUT1
2 – BLUE – OUT3 / OUT2



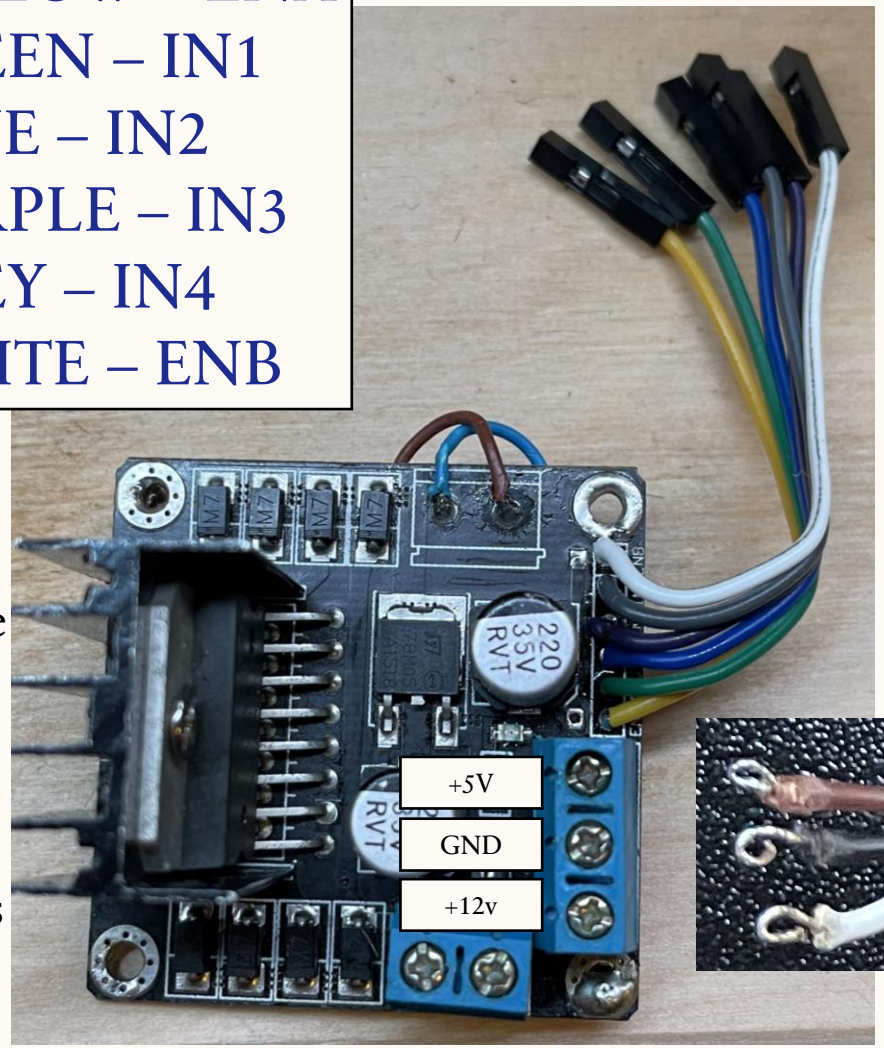
1 – YELLOW – ENA
2 – GREEN – IN1
3 – BLUE – IN2
4 – PURPLE – IN3
5 – GREY – IN4
6 – WHITE – ENB

1 – BROWN – +5VDC
2 – BLACK – GND
3 – WHITE – +12VDC



CABLES TO MAKE – MOTOR DRIVER

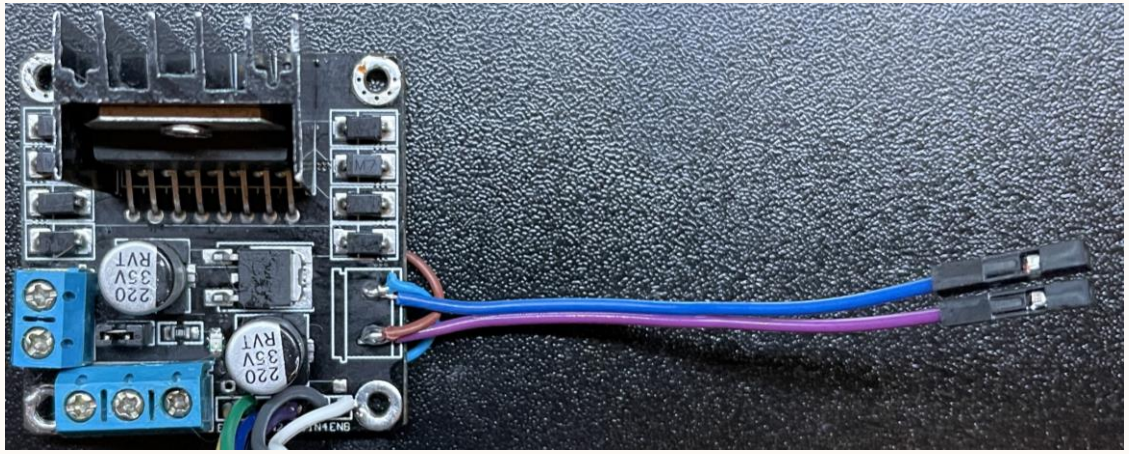
1 – YELLOW – ENA
2 – GREEN – IN1
3 – BLUE – IN2
4 – PURPLE – IN3
5 – GREY – IN4
6 – WHITE – ENB



You can solder to pins but jumpers are too tall. I removed pins and soldered to board holes

1 – BROWN – +5VDC
2 – BLACK – GND
3 – WHITE – +12VDC

Solder OUT4 to OUT1, and OUT3 to OUT2 on bottom of board. Can jump to screw terminals but hard to reach once installed if cable slips out

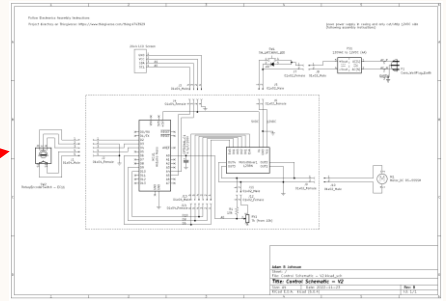


1 – PURPLE – OUT4 / OUT1
2 – BLUE – OUT3 / OUT2

Can solder to board or use screw terminal (I removed mine for another project). Careful not to work joints after

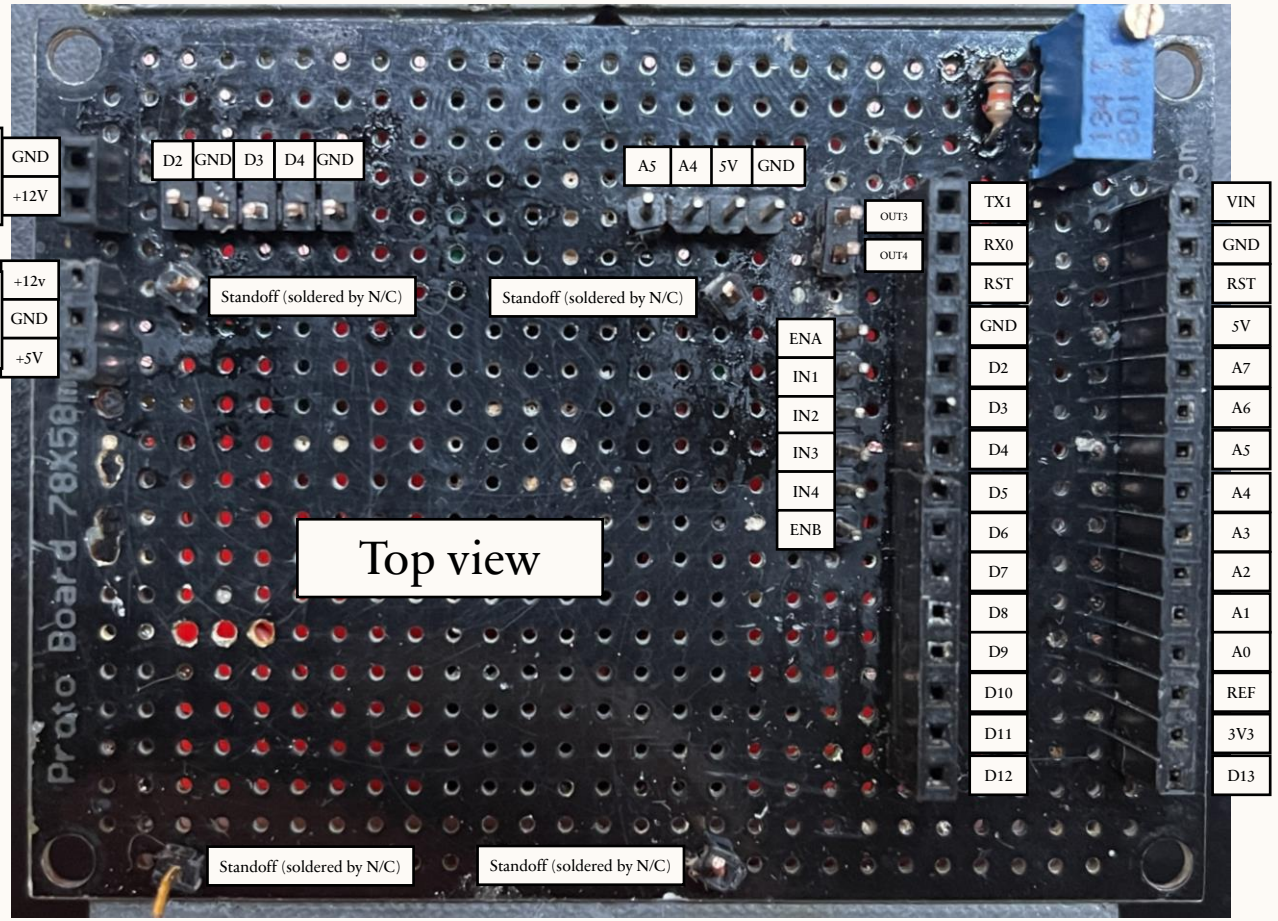
HEADER AND JUMPER SOLDERING

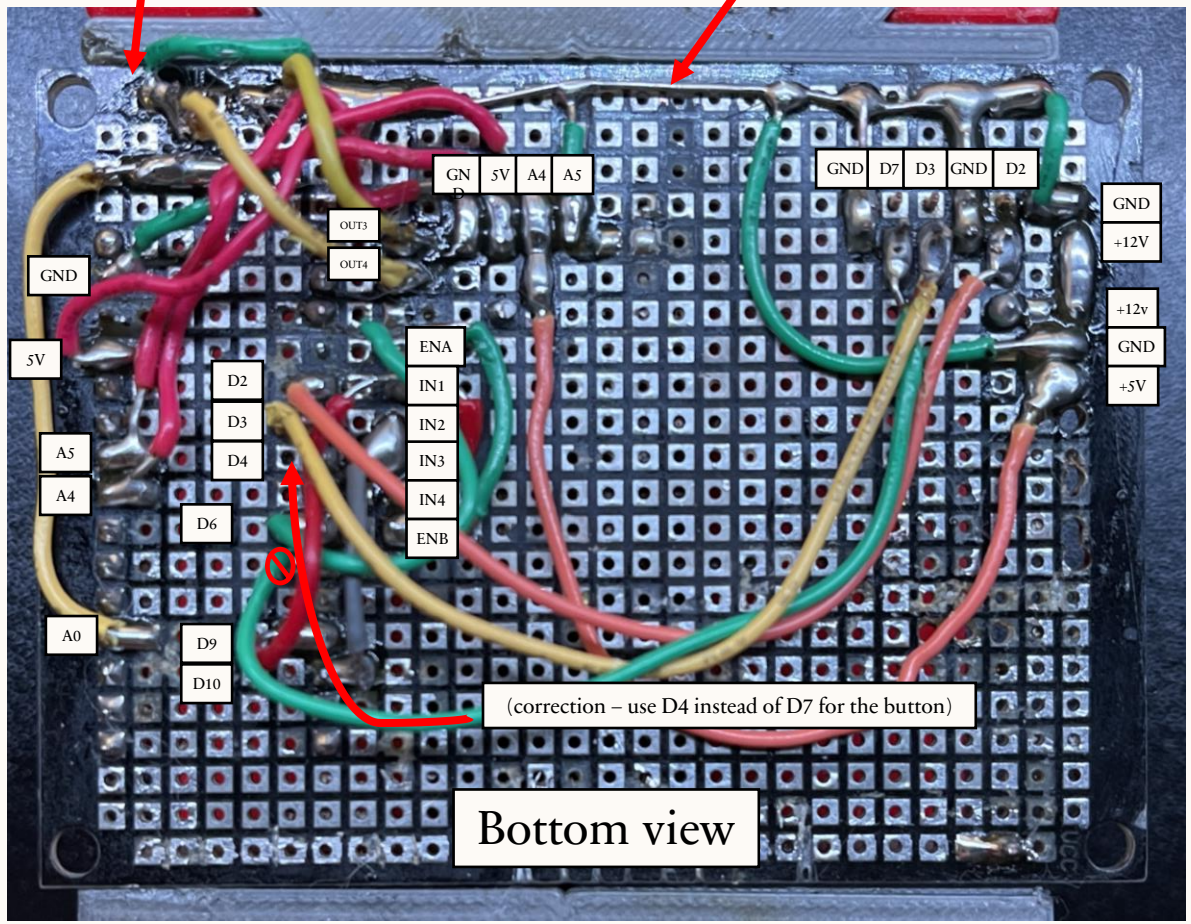
- Verify position/orientation on board before soldering
  - Refer to Control Schematic V2 from GitHub
- Clean all joints with alcohol and brush
- Trim leads coming out of the top after cleanup
- Verify connections with continuity test (for help - Fluke)



use knife to cut between traces on outer rails.
Sometimes the outer rows are all connected

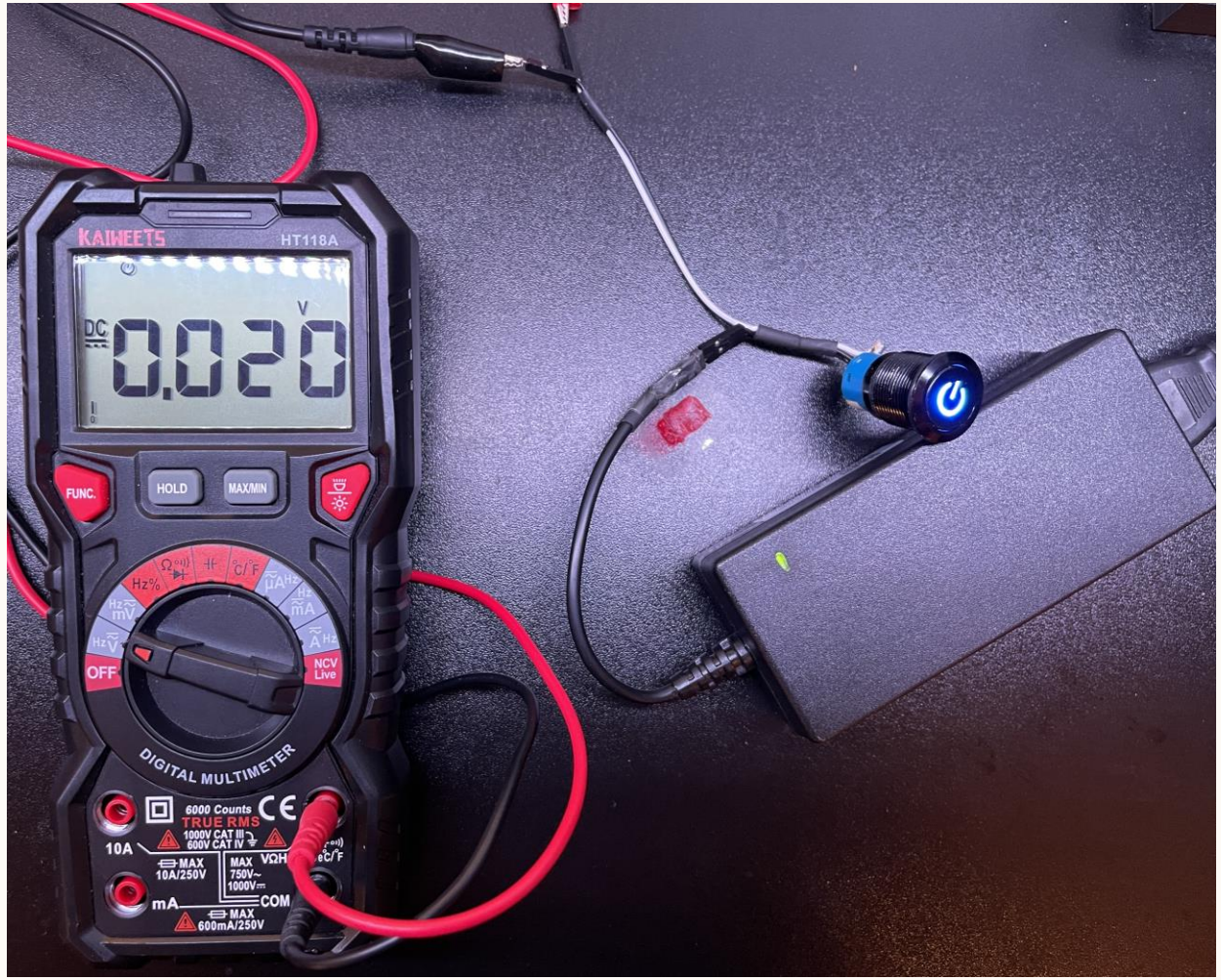
Rule: no ground loops – use ground tree to branch out to each device from the power supply source

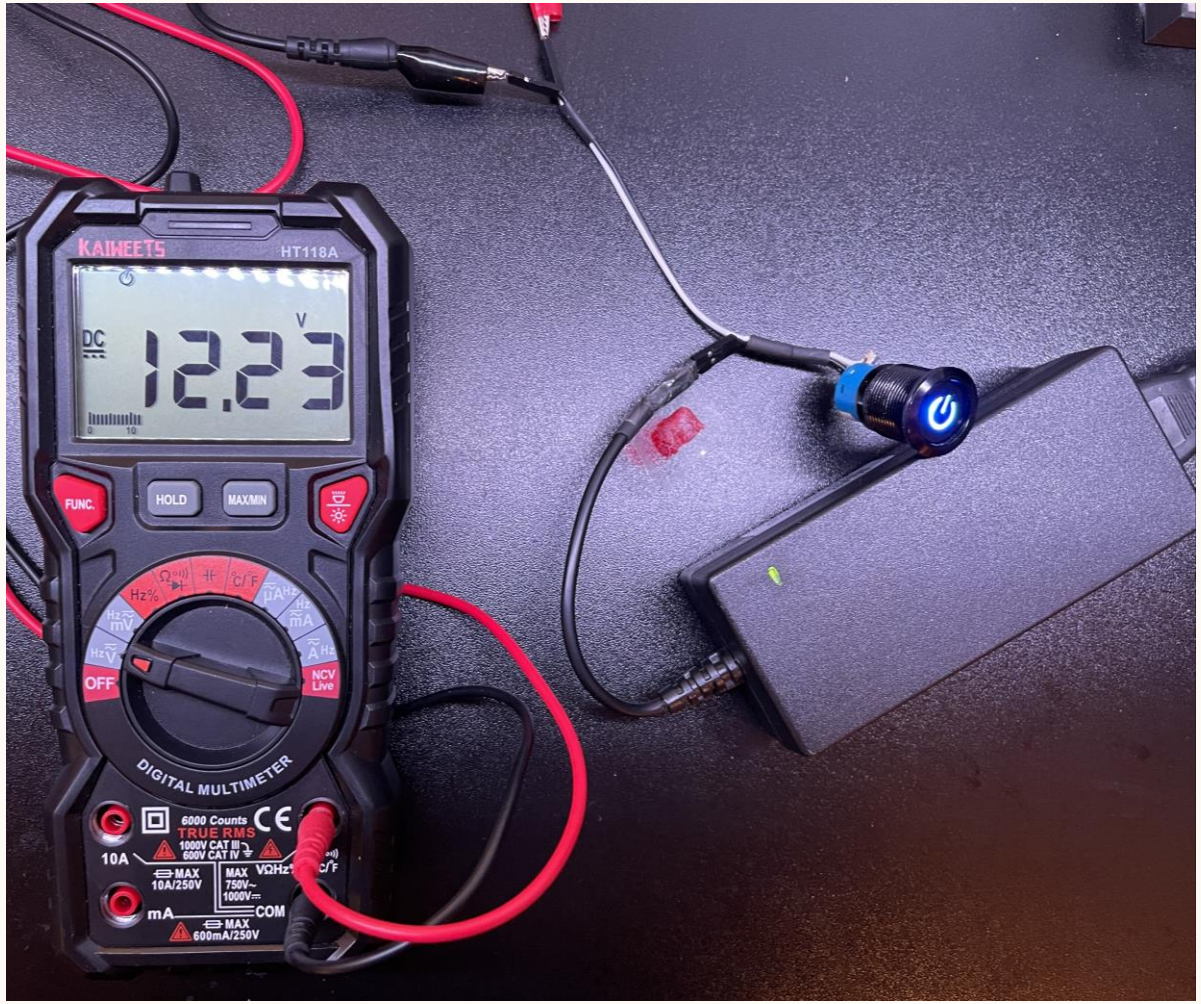




VOLTAGE CHECK – POWER BUTTON

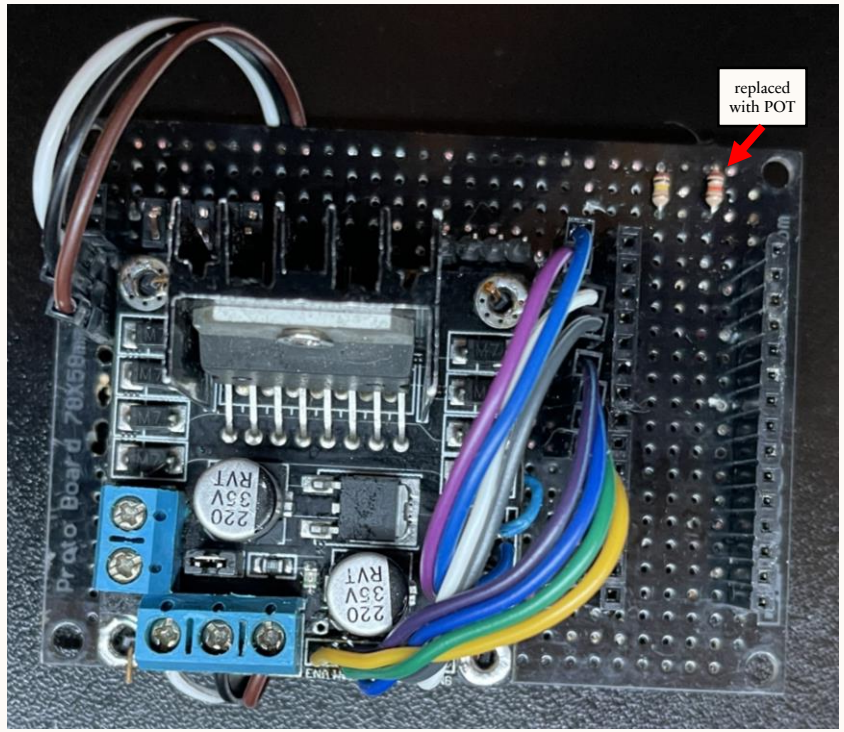
- Plug in power button (white wire to +12VDC).
  - with button unpressed (circuit open), button should illuminate but voltage should be ~0VDC
  - with button depressed (circuit closed), button should stay illuminated and voltage should go to ~12VDC

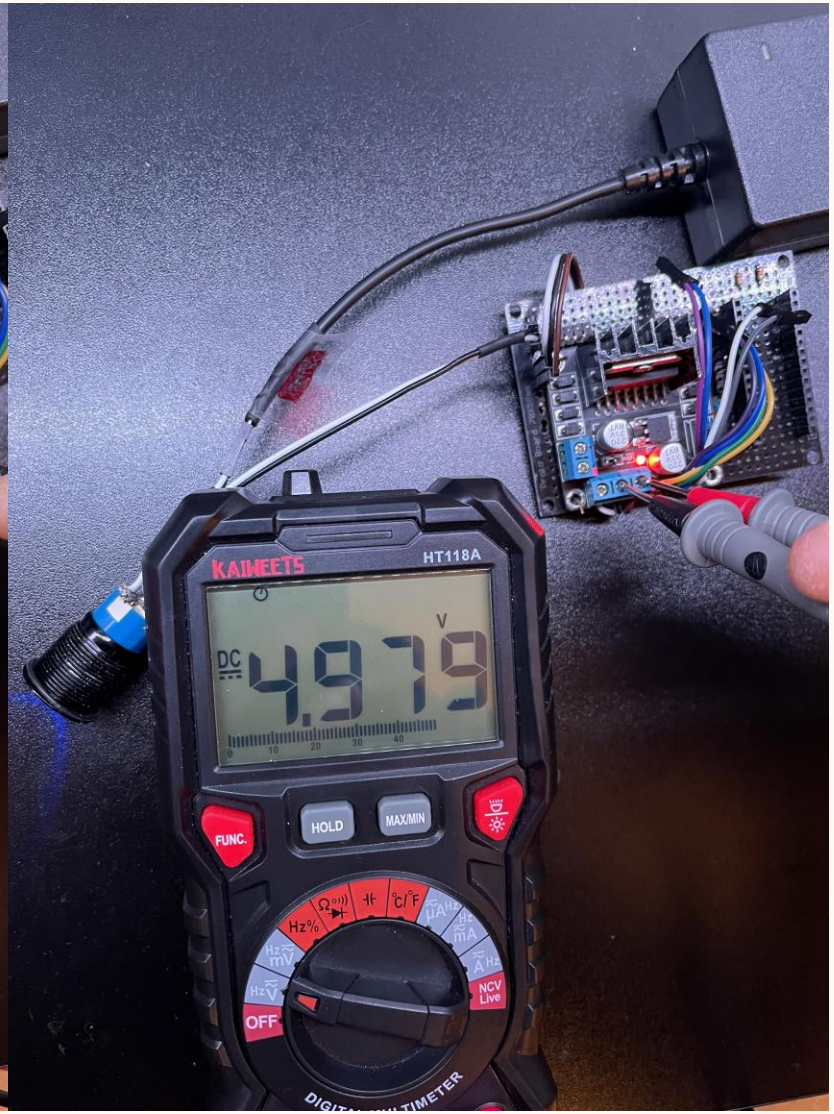




VOLTAGE CHECK – MOTOR DRIVER

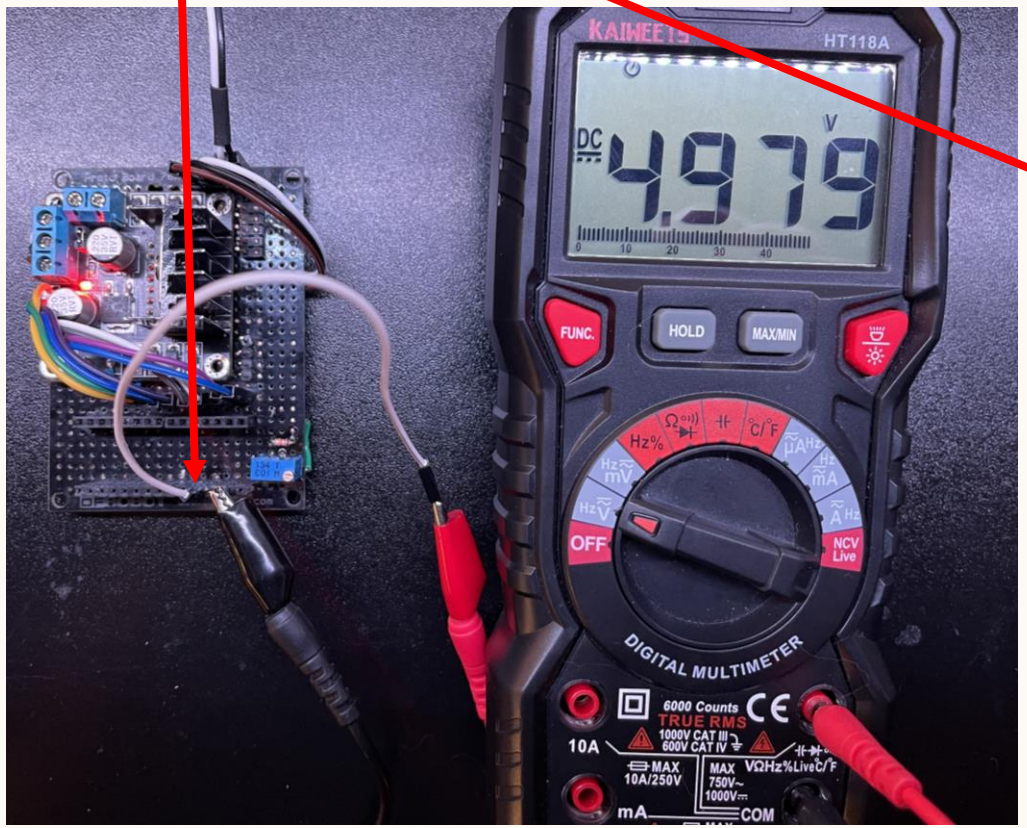
- Mount motor driver to protoboard assembly (bend mounting pins so it is easy to remove later if needed)
- With motor driver power cable plugged in, and power button ON, LEDs on motor driver should illuminate
  - check +12VDC and +5VDC
  - using power button should de-energize entire board

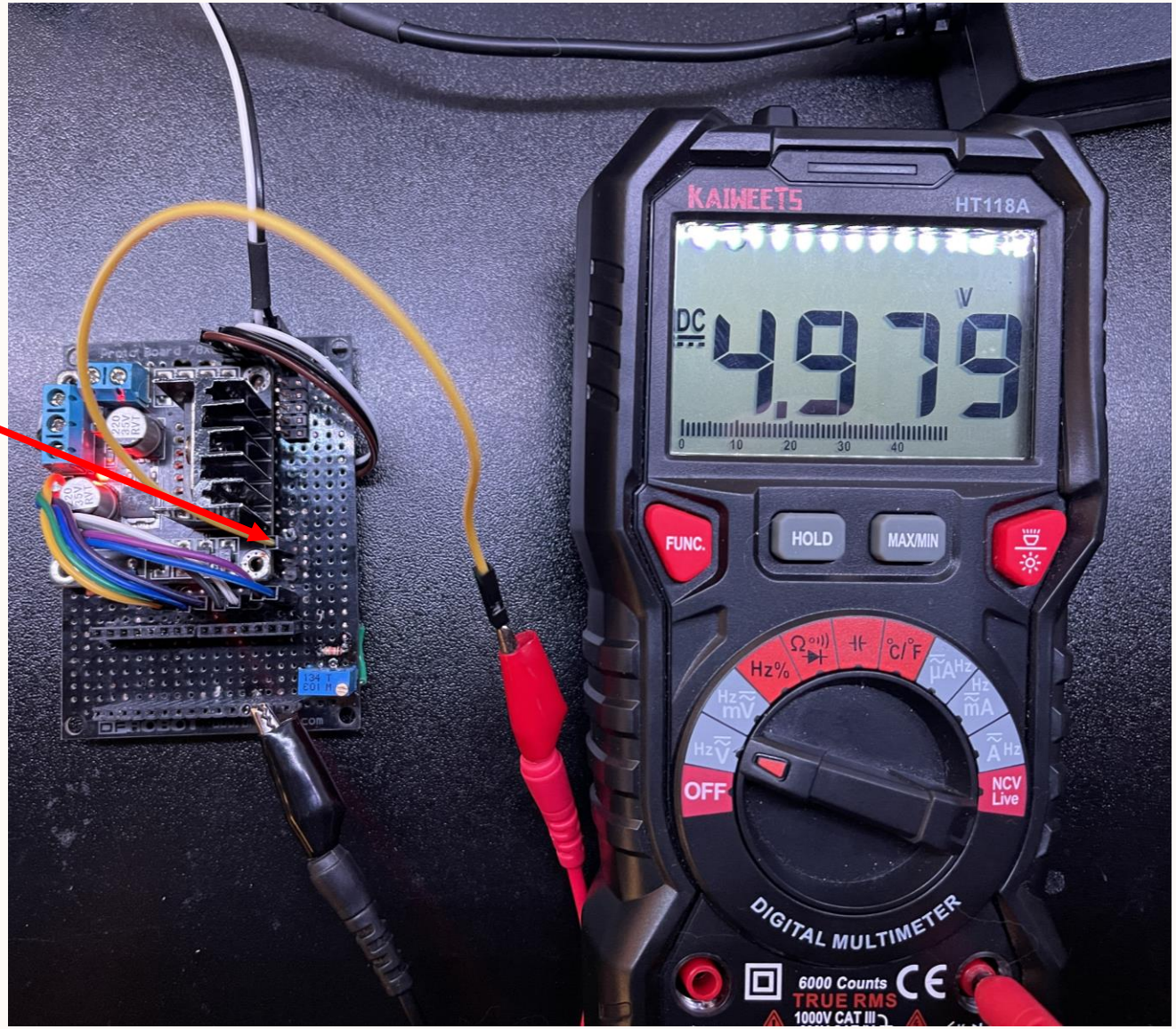




VOLTAGE CHECK – 5V SUPPLIES

- Using jumpers, measure power pins with respect to the GND:
  - the 5V Arduino pin
  - the 5V LCD pin

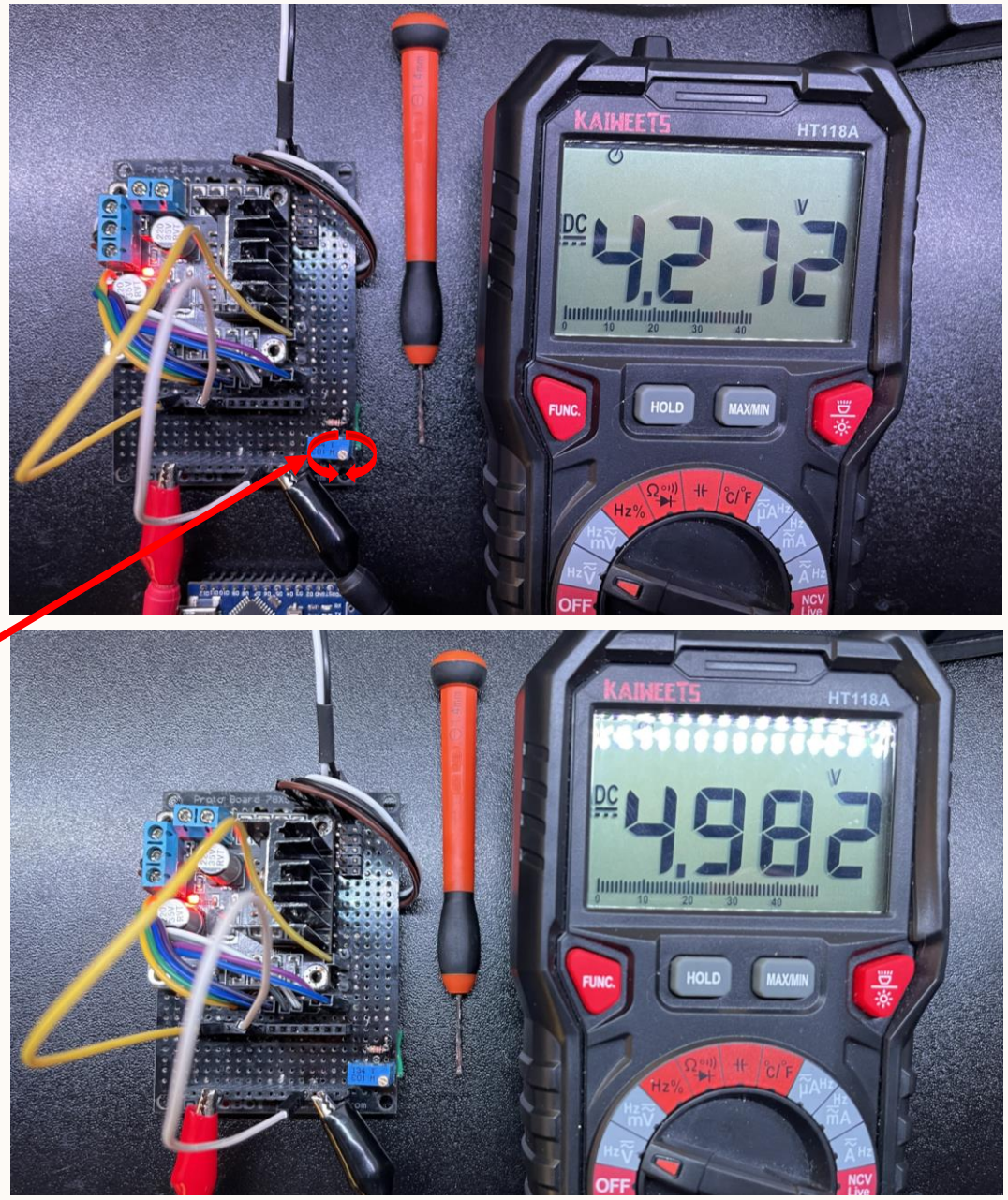




- Each should be approximately 5V

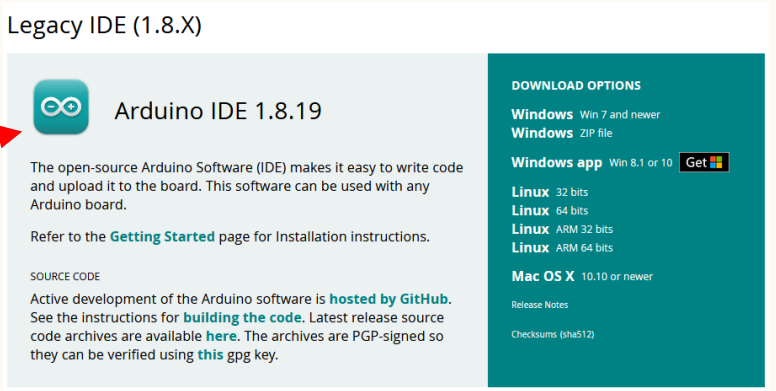
VOLTAGE CHECK – ANALOG INPUT

- Motor speed is measured by measuring the voltage across the motor terminals.
  - A high voltage indicates a high speed (without considering current)
  - The Arduino pins **ARE ONLY 5V TOLERANT,** this is why we have two resistors to split the voltage
  - Adjusting the potentiometer will bring the voltage measured across A0 and GND to a safer level
- Using jumpers, power:
  - the D9 pin with 5V from the LCD header
  - the D6 pin with 5V from the Arduino 5V socket
- Adjust the potentiometer with a screwdriver until A0 pin reads at least ≤ 5VDC, but try to match what you measured previously for Vs (~4.979V for me)
  - If you can only get between 5 and 12V, your resistors need to be switched with each other. If they aren't swapped, the Arduino will be damaged
  - If you can't get a reading, recheck your soldering work with a continuity check and verify there are no shorts or ground loops.

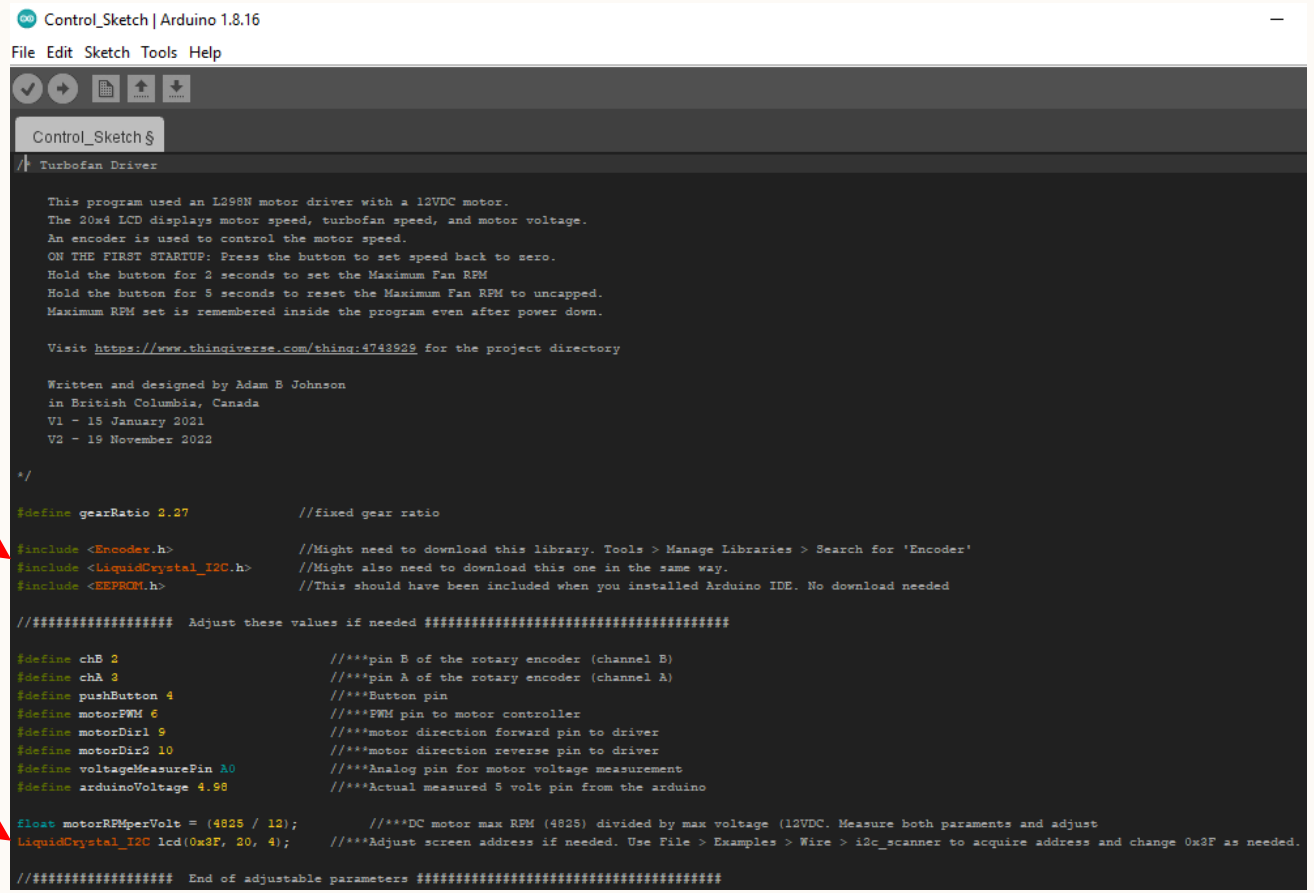


ARDUINO PROGRAMMING

- Download Arduino IDE (I use version 1.8.16) here
  - Any version should be fine, like the shown 1.8.19 version
  - **Arduino forum and website are a great repository for help



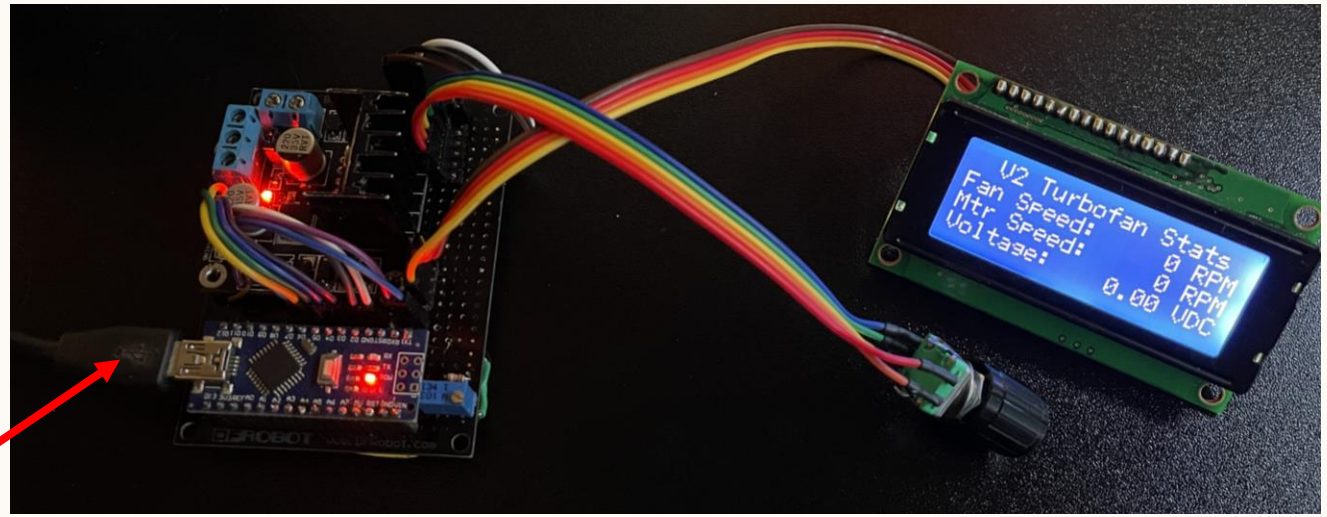
- Go to GitHub directory and download sketch folder:
  - https://github.com/nairck/TurbofanDriver/tree/main/Control_Sketch_V2
- Open Control_Sketch.ino in the Arduino IDE
  - File > Open > 'Control_Sketch.ino'
- You don't need to understand the code, but you should:
  - make sure you have the three libraries
    - Encoder.h
    - LiquidCrystal_I2C.h
    - EEPROM.h
    - Download libraries through the IDE (if needed)
      - Tools > Manage Libraries > *Search for these titles*
  - Check and change the I2C address for the screen
    - File > Examples > Wire > i2c_scanner
    - Will need to wire the Arduino and screen together
      - Use LCD female jumper cable and plug directly into Arduino pins
      - Program Arduino with i2c_scanner sketch (follow tutorial)



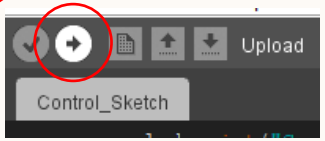
- (I have heard the replacement screen has a different address. This is normal for devices to have different addresses)

ARDUINO PROGRAMMING AND SCREEN CONTRAST

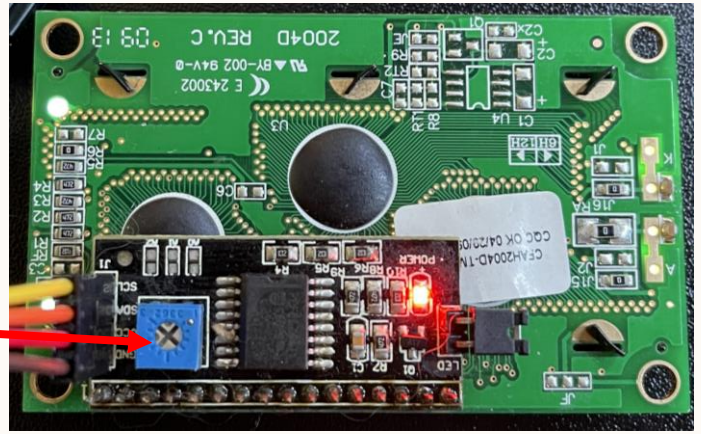
- Select the correct board to program
  - Tools > Board: "___" > Arduino AVR Boards > Arduino Nano
- Select the correct programming port
  - Tools > Port > COMX
    - Arduino must be plugged in with USB – see here for help



- Once you are happy with your connections, plug in all the devices and connectors **except the 12V supply or power switch.**

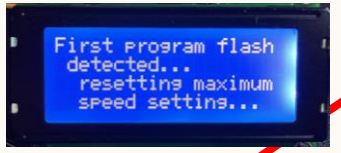


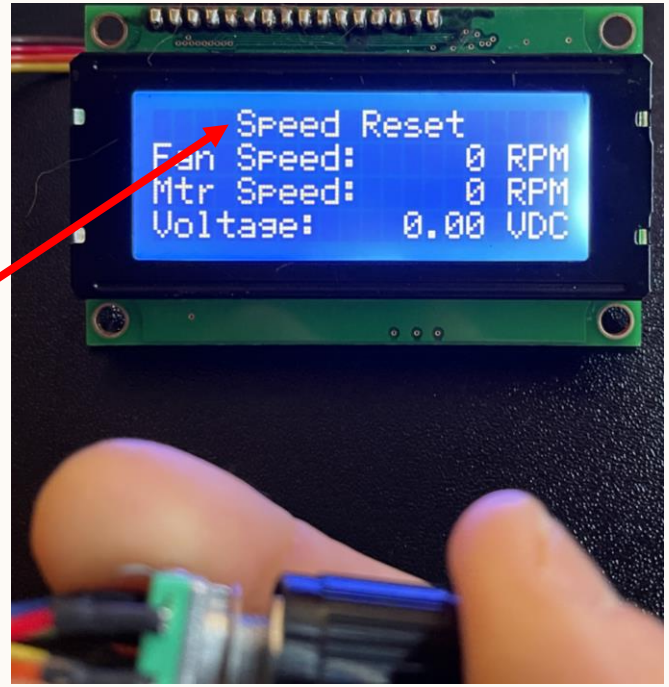
- Program the Arduino over USB
  - If it worked, you should see a startup screen for a few seconds, then the status screen!
- If the screen contrast is too low, you can adjust it on the I2C module potentiometer, located on the back of the LCD assembly
  - When we plug in the 12 VDC, it will likely need to adjust it again



FIRST TIME STARTUP AND SPEED RESET

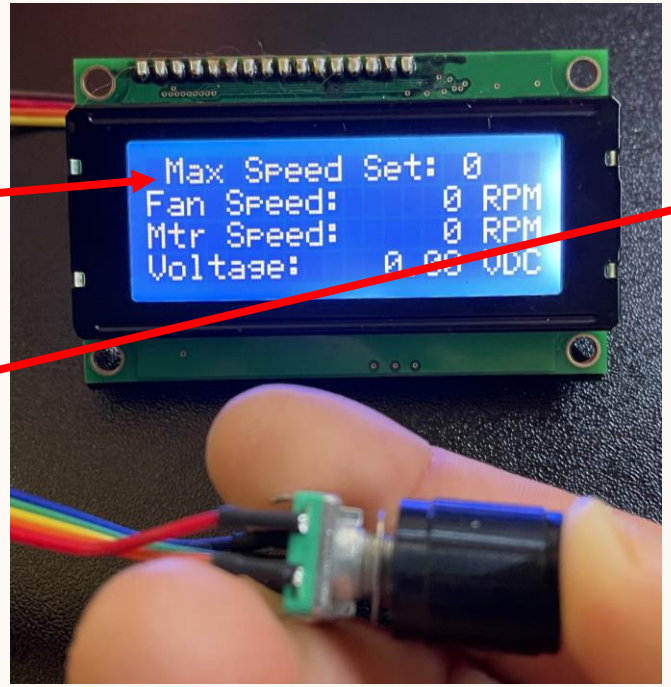
- ** The code now detects a first-time startup, so this procedure isn't required, but is recommended just incase a zeroed EEPROM system isn't detected***

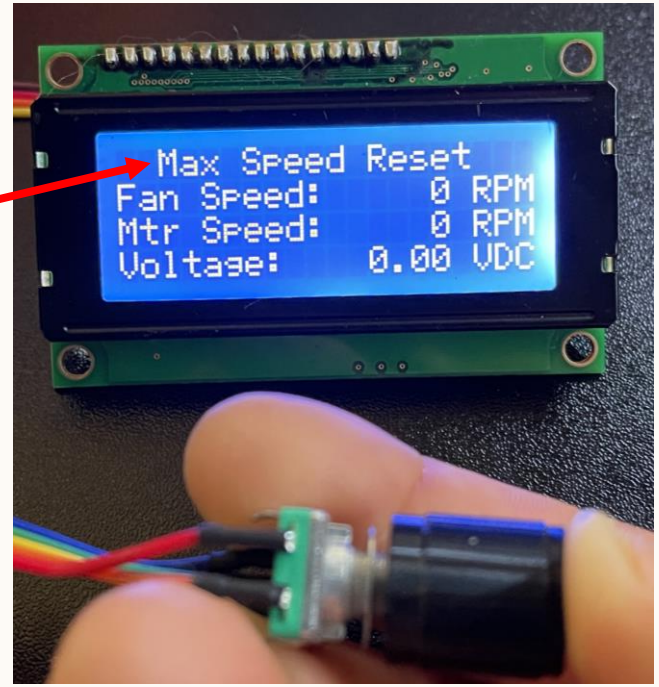




With the system still only powered by USB

- Press the button for less than 0.5 seconds
  - This will zero the driving voltage, stopping the fan instantly
- Hold the encoder button for 2 seconds
  - This will set the maximum speed of the system so you can limit your system if you so choose
- Hold the encoder button for 5 seconds
  - This will reset the maximum speed





The motor driver will do nothing if the maximum speed is set to zero

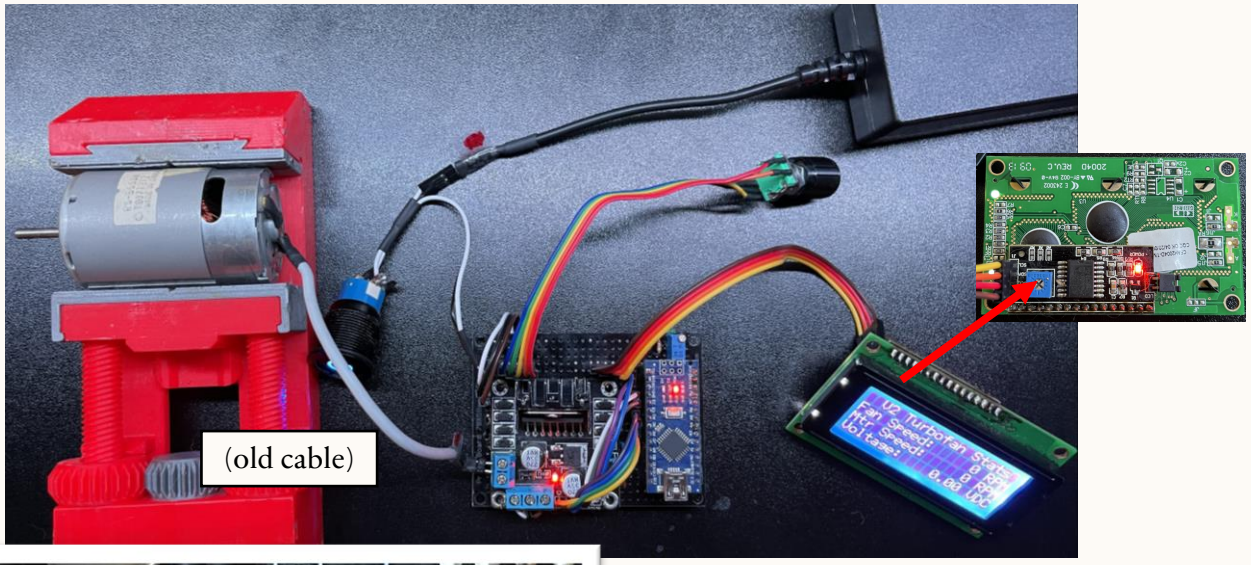
12V CIRCUIT TEST

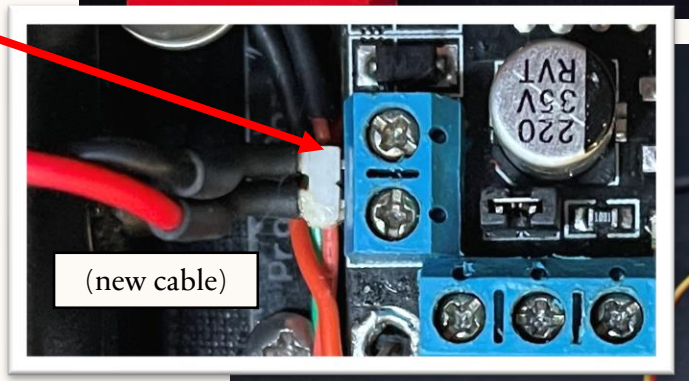
- If everything has gone well so far, we can test the system with the 12VDC supply
  - Unplug the USB for programming
  - Plug in the motor to the motor driver screw terminal
    - Secure it (without squishing) in a clamp or vice
  - Make note of motor polarity for final assembly
  - Plug in the 12VDC power supply and power button
  - Adjust LCD contrast on back of LCD

- Slowly adjust encoder button and make the motor spin
  - Hold it in hand so you get a feel for the adjustability
  - Don't let motor rip away from electronics and pull on any wires
  - If the system works as expected, resting on a cloth can be good enough

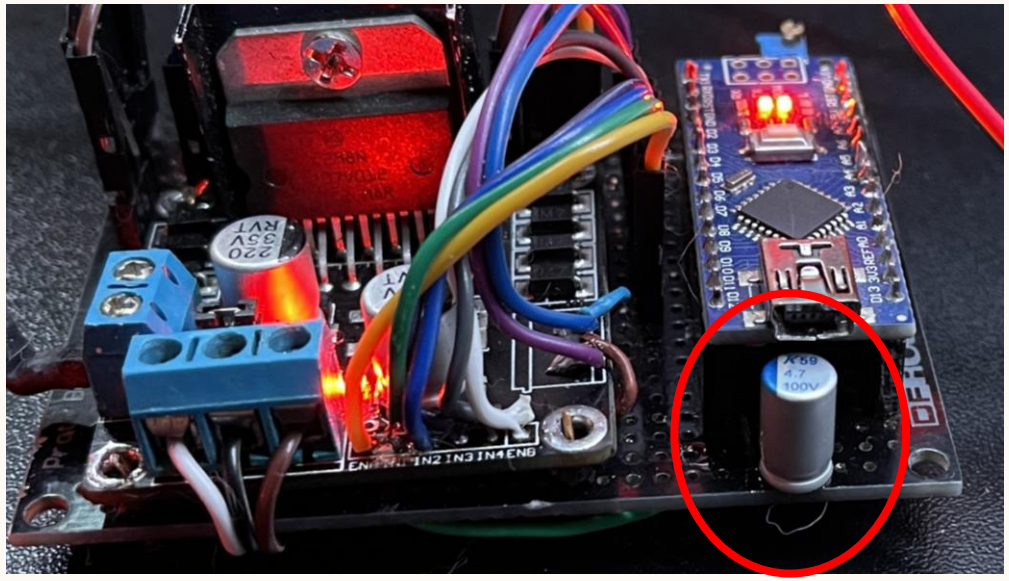
Great work, you're done!!! Everything is ready to install into the electronics bay.

One extra step (next slide) can help voltage reading noise if you have a capacitor kicking around.



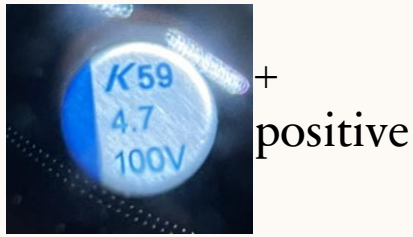


SIGNAL CONDITIONING - EXTRA



- To filter out AC noise on the motor voltage reading, a 4.7µF/100V electrolytic capacitor was used (which I had laying around
  - Anything between 0.1µF and 100µF will be helpful
  - Other capacitor would also be okay (like non-polarized ceramic)
  - Make sure voltage rating is at least 10V
  - Make sure to get the polarity correct (if applicable)

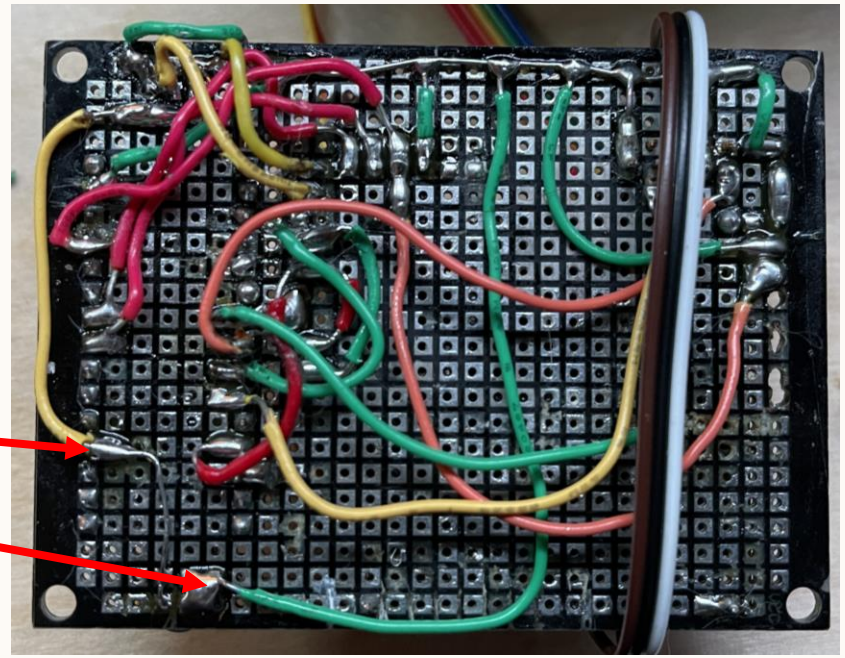
- negative



+ positive

+ positive

- negative



- Solder to the breadboard near/under the USB port
  - Positive terminal connected to A0 (voltage read pin)
  - Negative terminal connected to GND
  - (make sure capacitor is connected physically close to the A0 pin)

- Using decoupling capacitors on other circuit locations can also minimize noise, but experiment showed minimal impact

THANK YOU, AND GREAT WORK!

Continue the build with the YouTube assembly videos on my channel!
https://www.youtube.com/channel/UCdyHOoZkerU78A9CjtsDdRQ

Follow the project directory on Thingiverse for the most recent updates:
https://www.thingiverse.com/thing:4743929