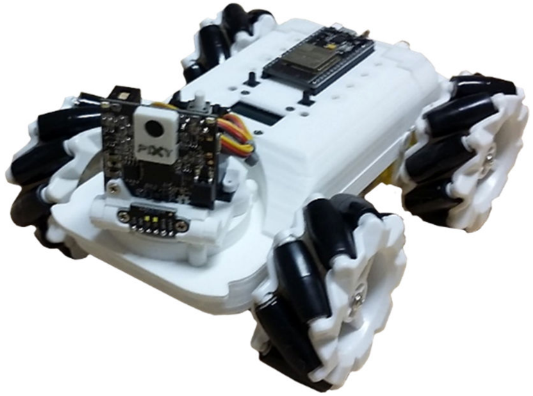
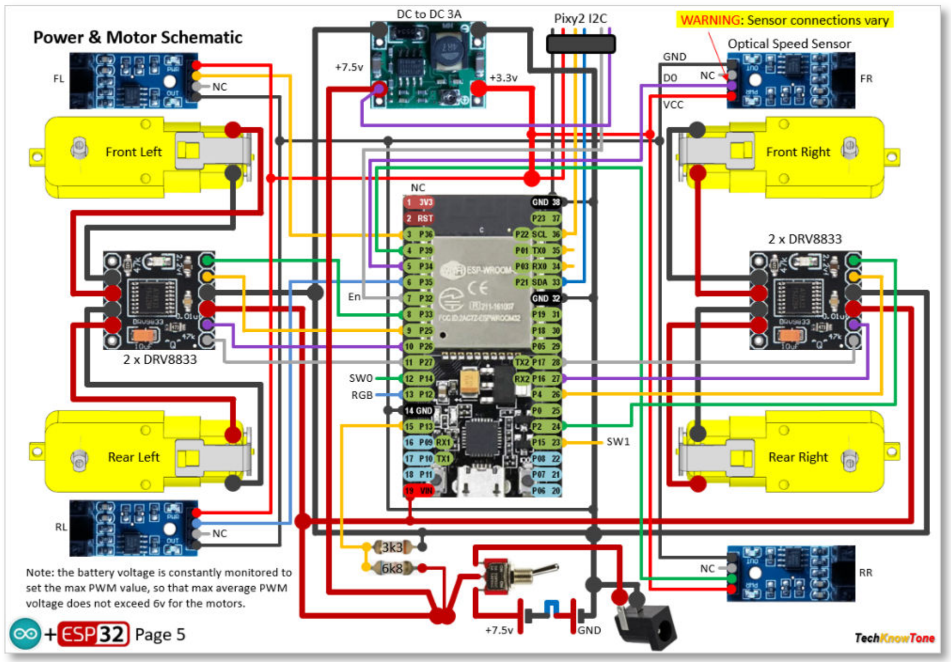
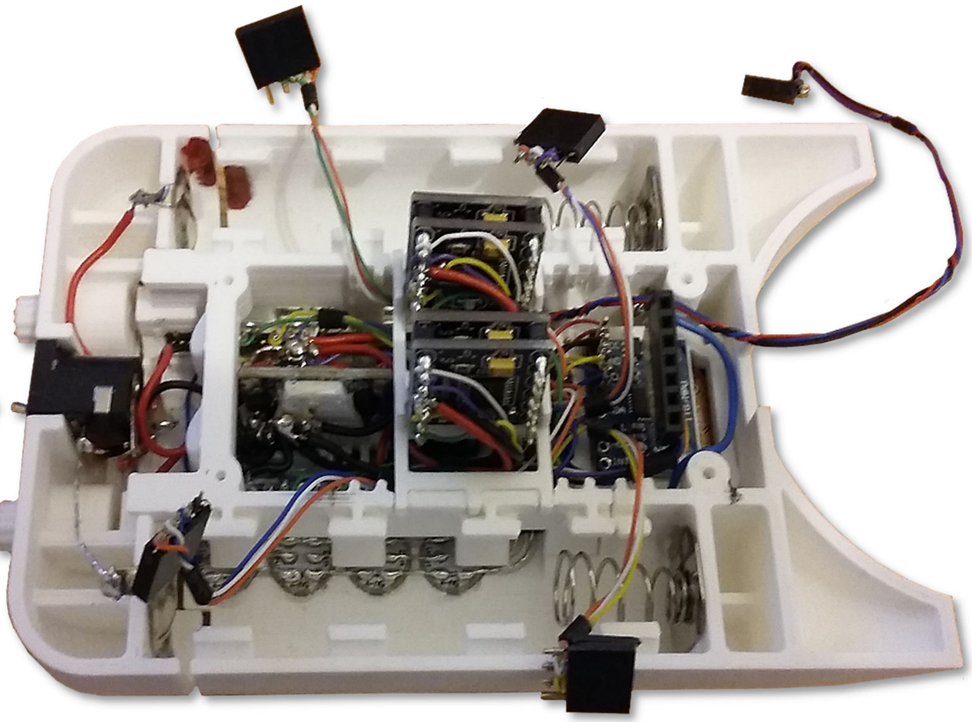
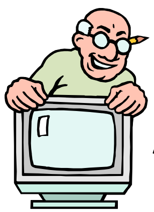
Project PixyBot2

Circuits & Wiring







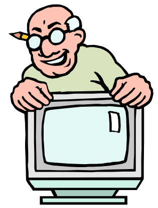


A complex project, please read through this document <u>before</u> starting.

+ESP32

Issue: 1.2 Released: 26/11/2025 TechKnowTone

CAUTION



Lithium batteries can be extremely dangerous, if not handled and cared for properly. This design does not include any form of current limiting circuit, like a fuse. So, care must be taken to ensure that the wiring guidelines are followed accurately, that checks are made for short-circuits, and that battery polarities are marked, and they are inserted the correct way round. Failure to do so, could result in an explosive fire.

Charging Practices: Always remove batteries from your project to charge them. Use a charger, designed for the battery used, and from a trusted supplier. Choose a flat, non-flammable surface to charge on, away from flammable materials. Never leave unattended when charging. Don't charge overnight. Monitor charging to ensure charge characteristics are as expected. Only pair batteries with similar characteristics. Do not overcharge, or leave charging for prolonged periods. This increases the risk of damage and fire.

Battery care & maintenance: Stop using a battery if it is swollen, damaged, dented or leaking. Never charge a damaged battery. Never allow a Lithium battery to discharge below 3.2 volts, as cell damage will occur. Avoid extreme temperatures. Do not charge or store batteries in very hot or cold environments. Don't cover batteries whilst charging, as this can trap heat, causing overheating.

In case of fire: Get out and stay out. If a fire starts, leave immediately, and call the fire brigade. For low voltage Lithium batteries, water is a safe extinguisher.

Built-in Monitoring: Most of my project designs include code, and circuitry, to monitor battery voltage, whilst in use. This code then seeks to alert the operator, when the battery has reached a critical low voltage, before shutting down power consuming circuitry; including the micro. Time should therefore be spent on calibrating this feature, as a precaution, for good battery management and maintenance.

Carefully dispose of batteries that damaged, or discharged below their critical voltage.

Hand Tools:

Recommended:

- Fine Nosed Pliers
- Side Cutters
- 1.5 mm Drill
- 2.0 mm Drill
- 4.0 mm Drill
- Needle Files
- Screwdrivers
- Craft Knife

Note: Not all items needed are shown here.

Tools & Materials:

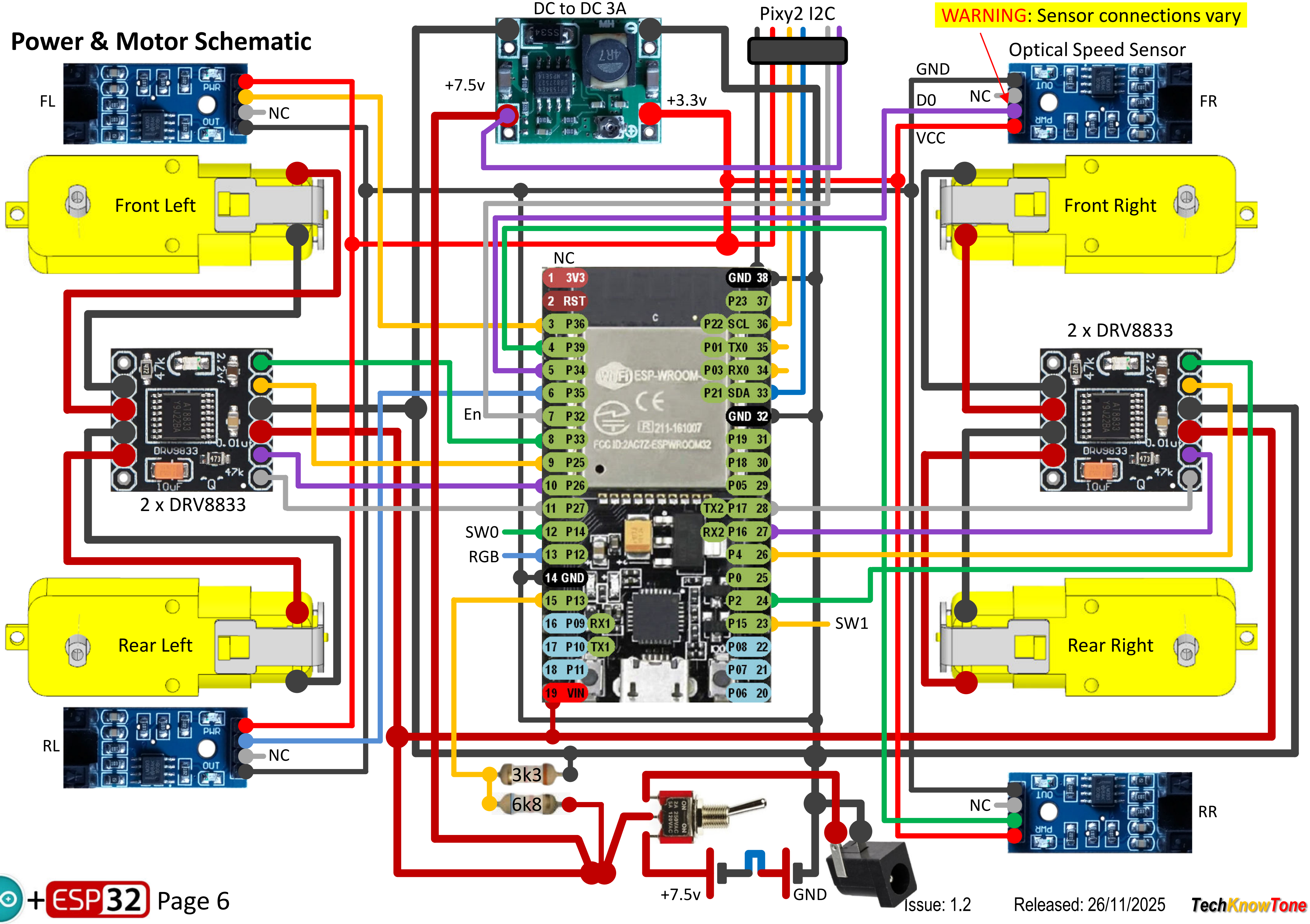
Temperature controlled iron
Solder flux
Resin cored solder
Hot melt glue gun {optional)
2-part epoxy resin glue
Screw drivers
Wire wrapping tool
Wire wrapping wire 30 AWG
24 AWG stranded wire (red & black)
Multimeter

Special Tools

Hot air gun for heat shrink sleeving
Ratchet crimping tool + 2.54mm female connectors

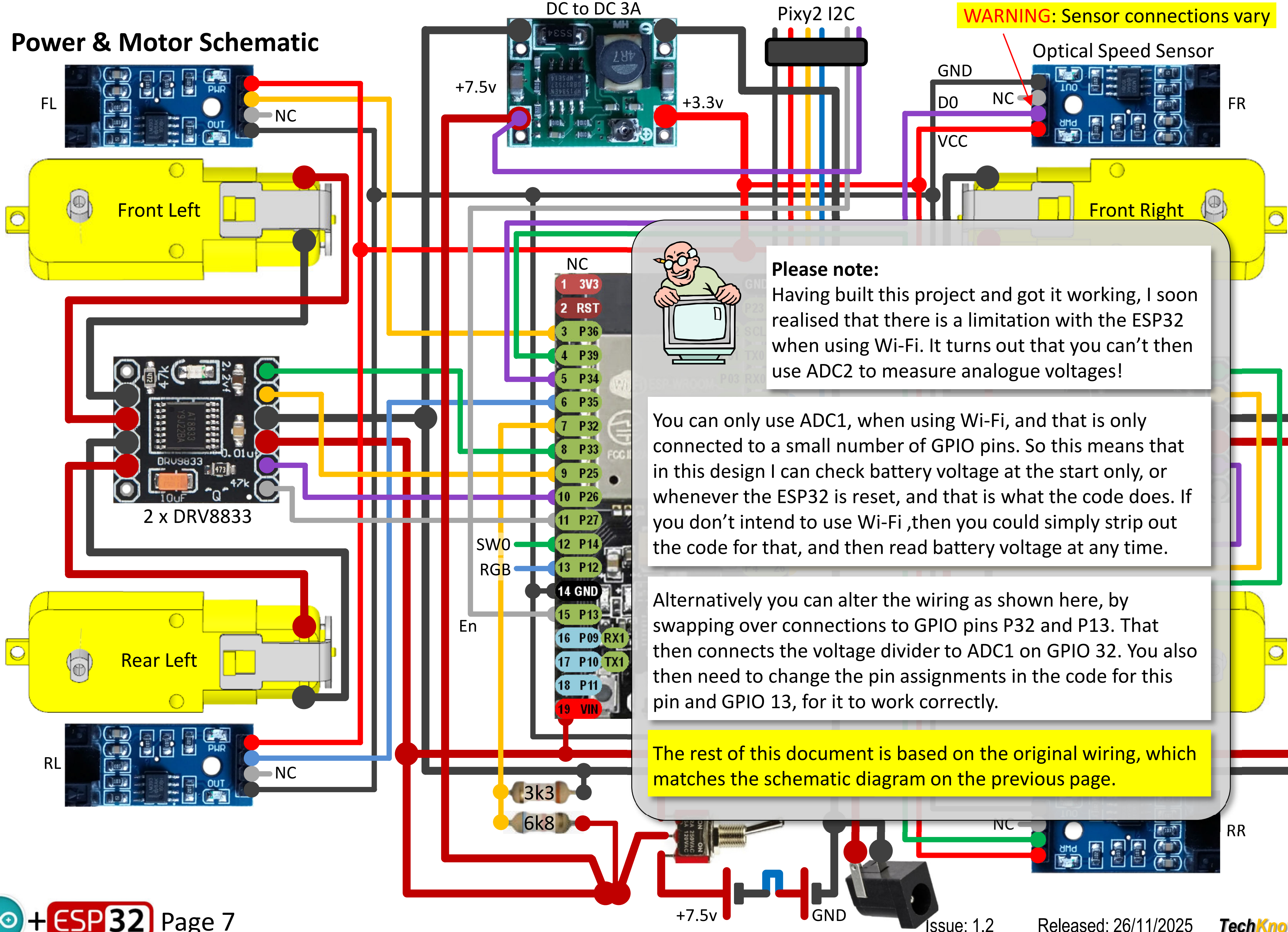
Watch videos on the internet to learn how to best use this tool, <u>before</u> attempting to shorten the servo leads.

Power & Motor Schematic



Issue: 1.2 Released: 26/11/2025 TechKnowTone

Power & Motor Schematic



Please note:
Having built this project and got it working, I soon realised that there is a limitation with the ESP32 when using Wi-Fi. It turns out that you can't then use ADC2 to measure analogue voltages!

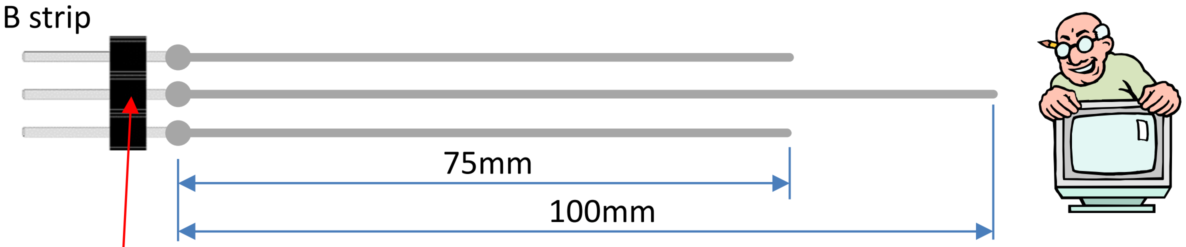
You can only use ADC1, when using Wi-Fi, and that is only connected to a small number of GPIO pins. So this means that in this design I can check battery voltage at the start only, or whenever the ESP32 is reset, and that is what the code does. If you don't intend to use Wi-Fi ,then you could simply strip out the code for that, and then read battery voltage at any time.

Alternatively you can alter the wiring as shown here, by swapping over connections to GPIO pins P32 and P13. That then connects the voltage divider to ADC1 on GPIO 32. You also then need to change the pin assignments in the code for this pin and GPIO 13, for it to work correctly.

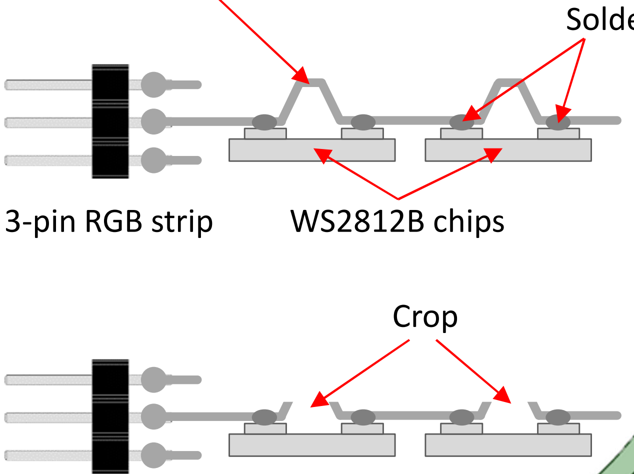
The rest of this document is based on the original wiring, which matches the schematic diagram on the previous page.

Top Plate Pin Strip Wiring

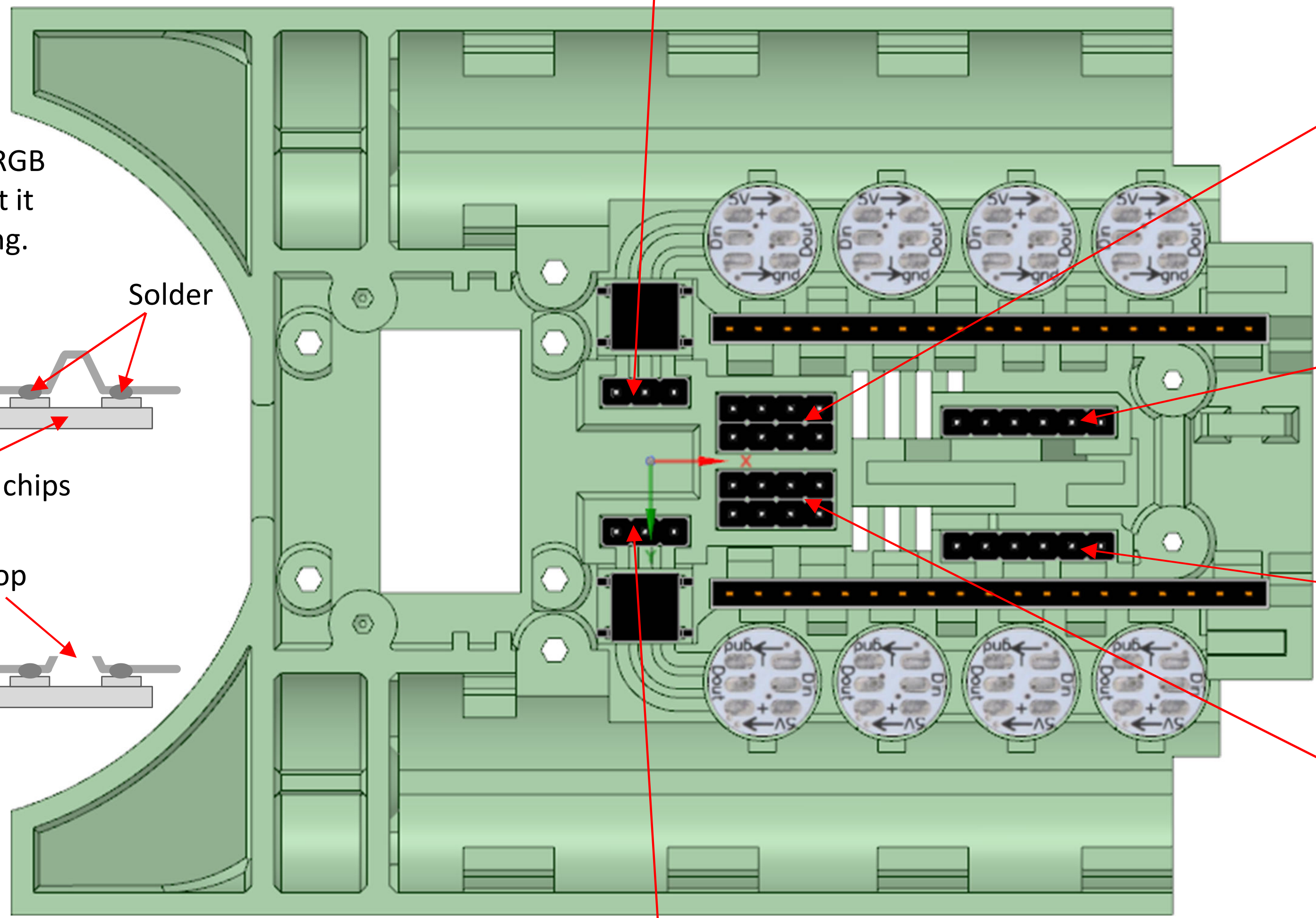
Wiring to the RGB LED chips requires special attention. The chips are glued into the Top Plate prior to wiring, and the wires pass over the back of the button switches, where there is limited clearance, so take care to avoid short circuits.

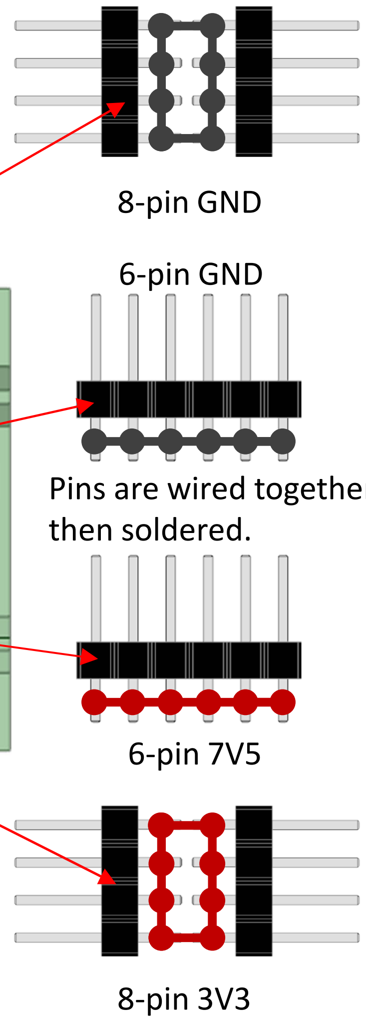


The centre data wire for the RGB chips is longer initially, so that it can be bent up when soldering.

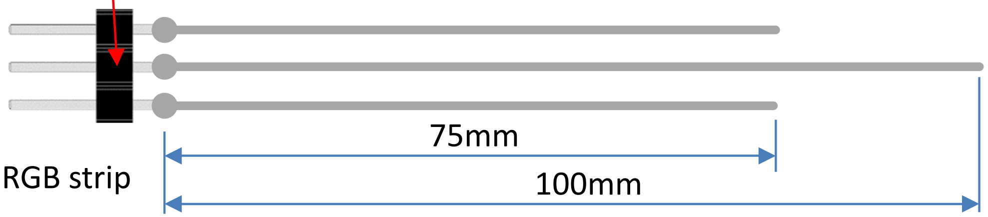


After soldering we crop out the raised wire to break the D_{IN} to D_{OUT} connection.





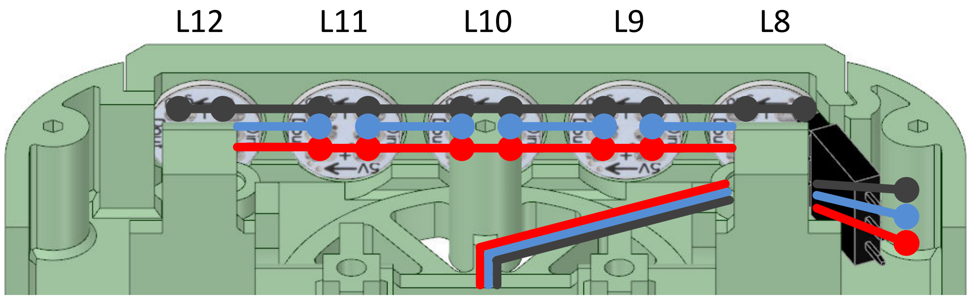
Note: Dimensions are minimum lengths required, any excess wire will be cropped after soldering.

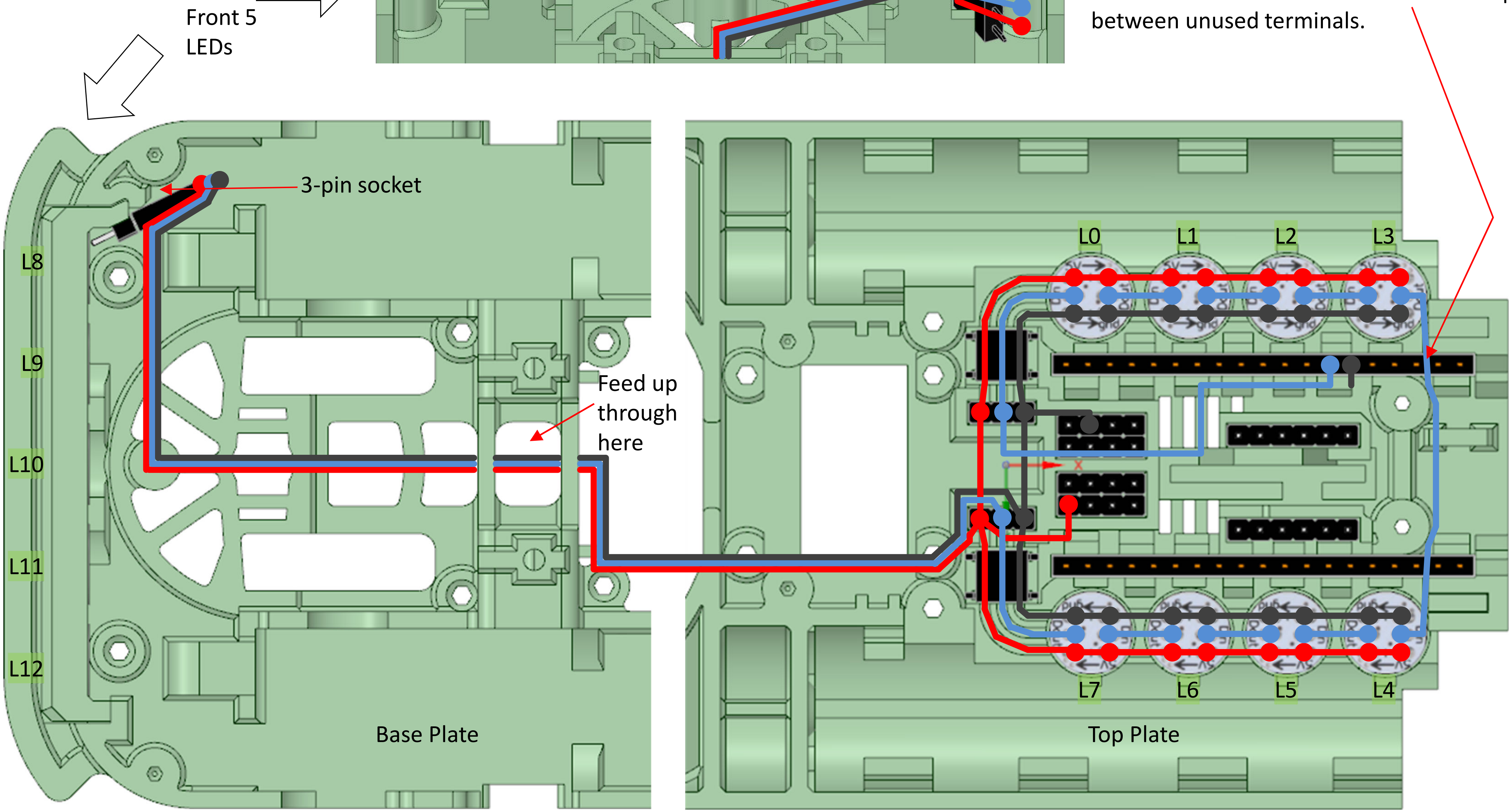


RGB LED Wiring

The front 5 LEDs are pre-wired in a special jig before being glued into the Base Chassis plate.

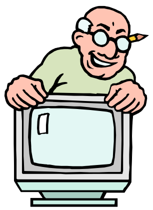
The wires in the top plate are routed over the back of the button switches. Take care not to short the connections. RGB LED is fed from micro GPIO12. LED L3 data wire is connected to LED L4 by passing the insulated wire over the micros socket strip between unused terminals.



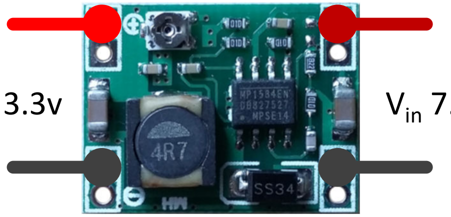


Base Plate is shown sectioned so as not to obscure Top Plate details.

Top Plate Power Wiring

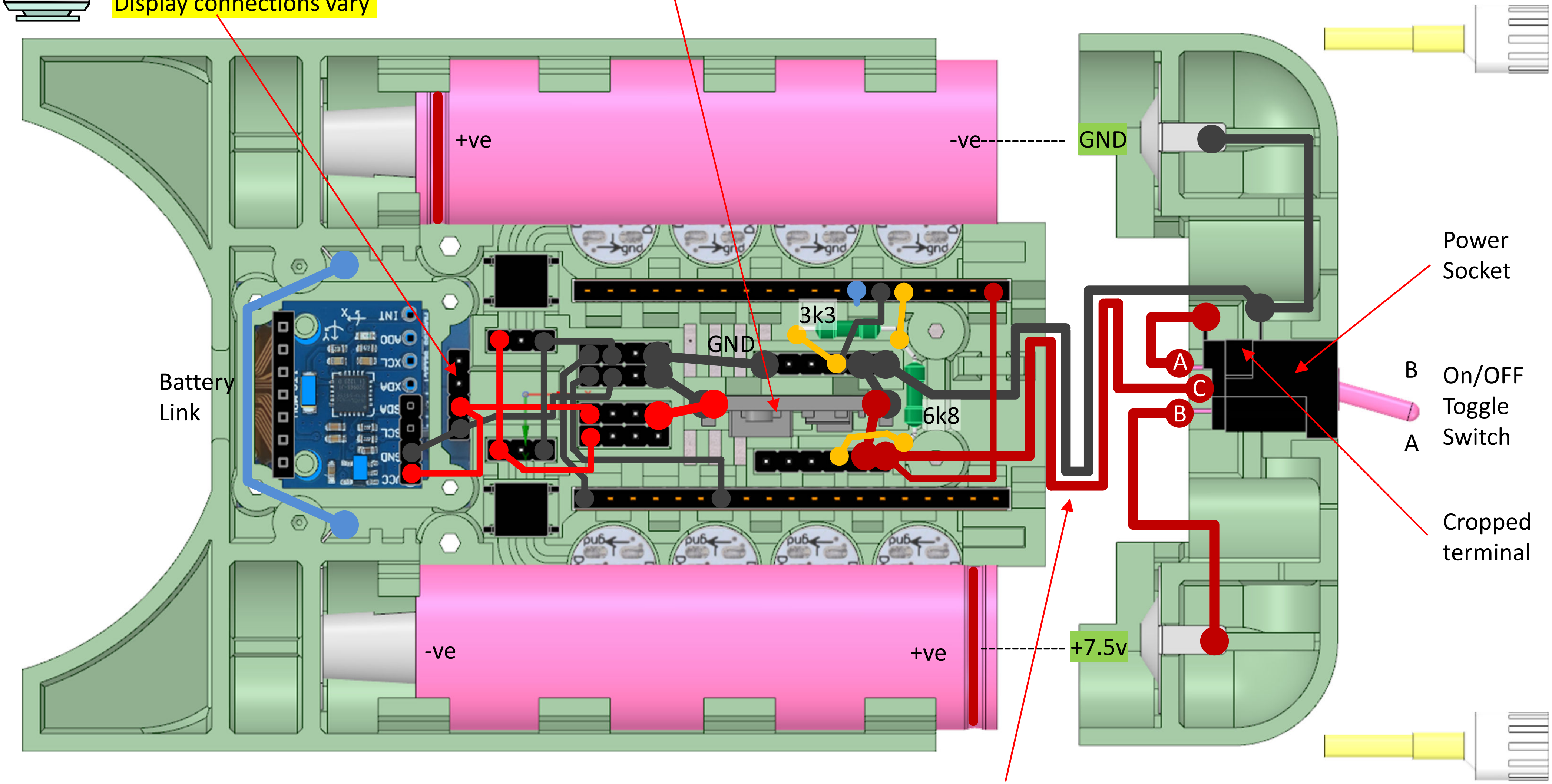


WARNING: Display connections vary



Wire the battery cover connections to the step down converter before gluing it into position in the Top plate.

Battery cover retained by 2 x M3 nylon screws.

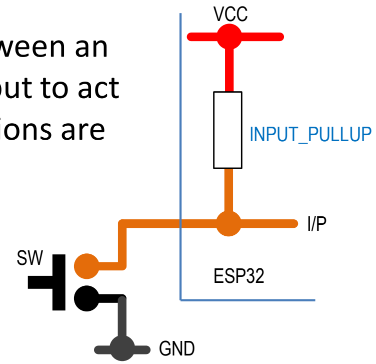


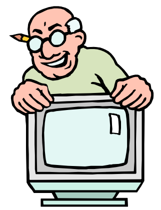
The connecting wires should be of sufficient length to allow the cover to be moved away, so that the batteries can be removed for charging. The batteries will spring out of the case cover.

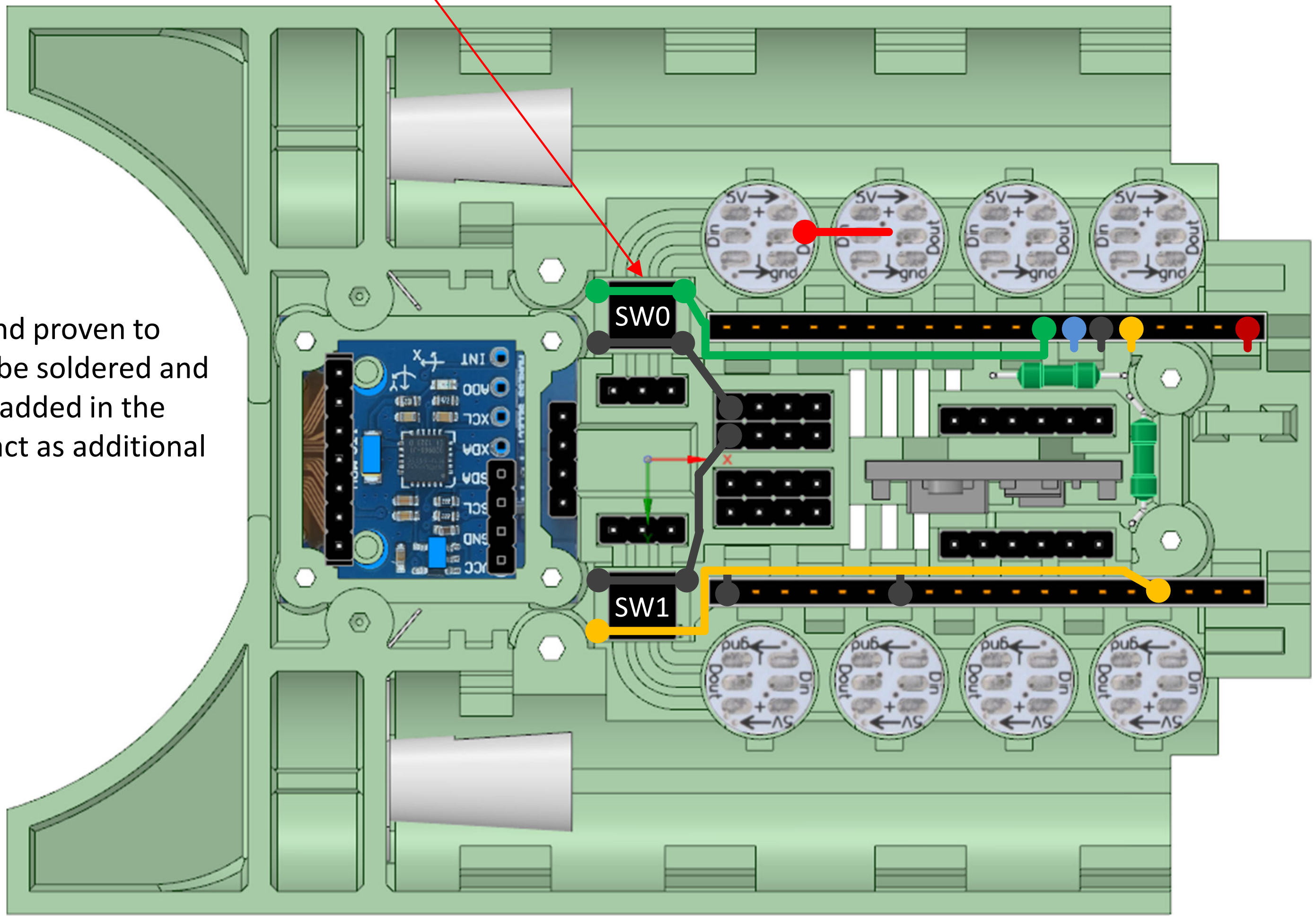
Top Plate Button Switch Wiring

The two button switches SW0 and SW1 are wired up after the RGB LEDs, as their wires pass over the top of the LED wiring. Take care to avoid short circuits.

Both switches are connected between an input and GND. Code set each input to act as a pullup resistor. So button actions are active LOW.

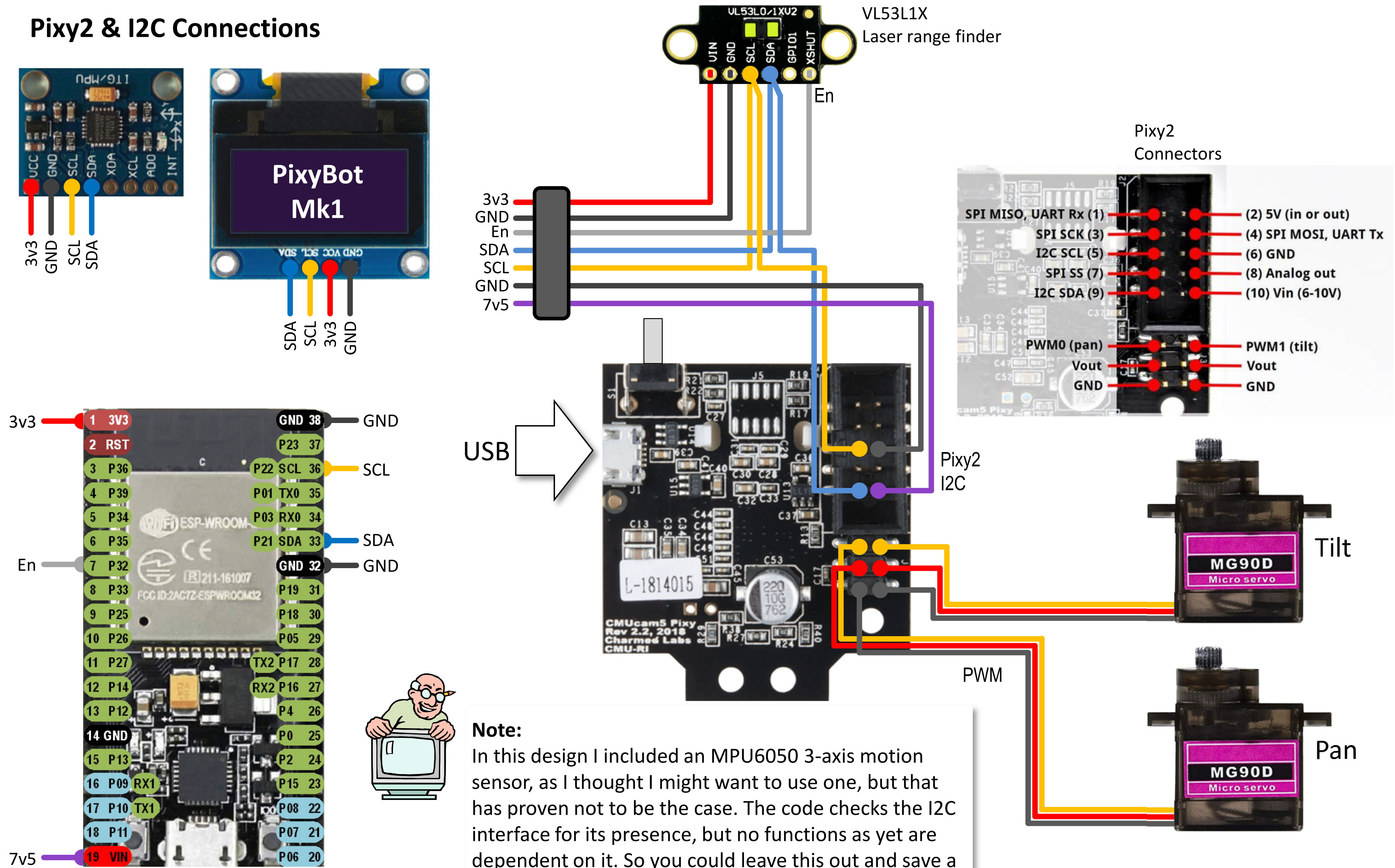






Once the wiring is tested and proven to work, the connections can be soldered and additional glue can also be added in the region of the LED wires to act as additional insulation if needed.

Pixy2 & I2C Connections

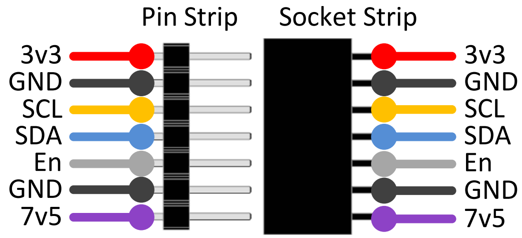


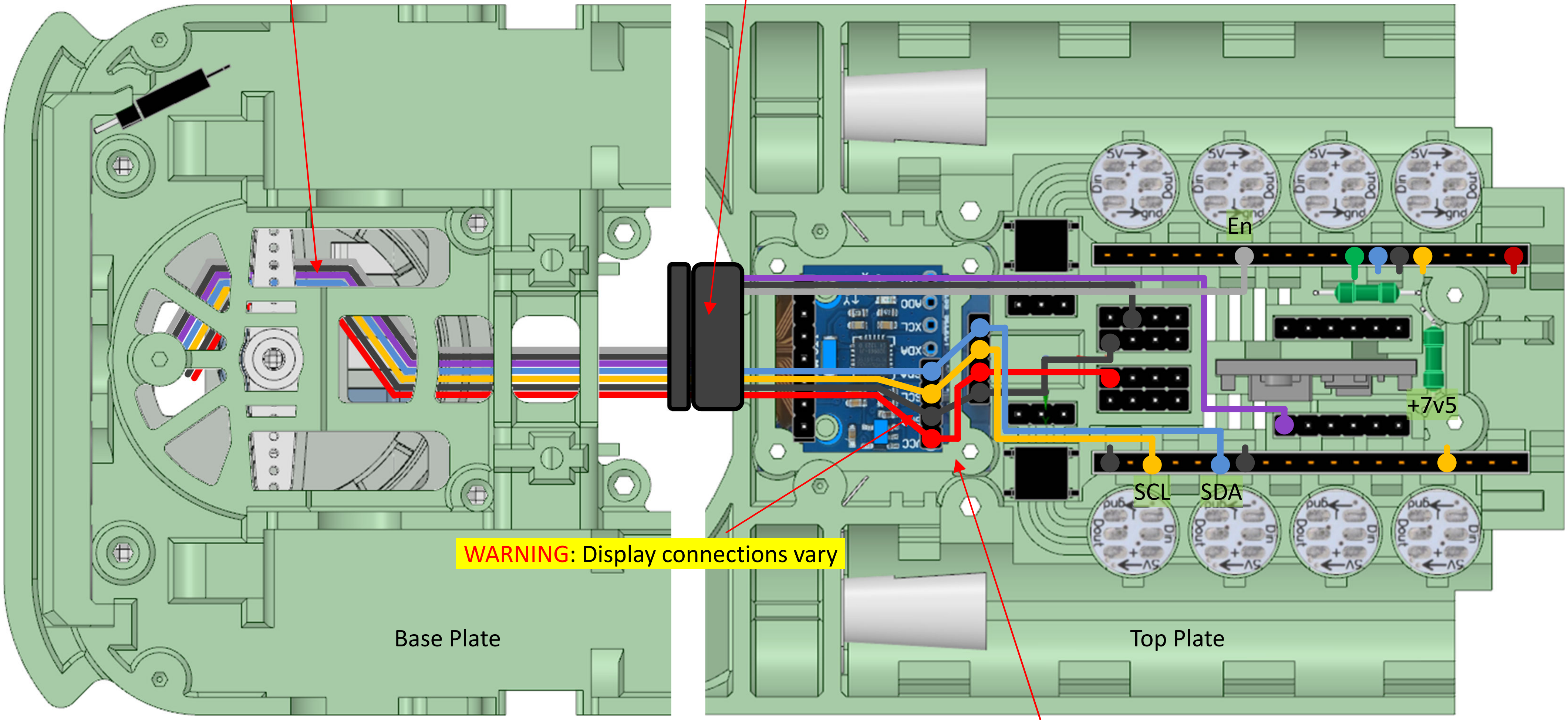
Note:
In this design I included an MPU6050 3-axis motion sensor, as I thought I might want to use one, but that has proven not to be the case. The code checks the I2C interface for its presence, but no functions as yet are dependent on it. So you could leave this out and save a small amount of cost and wiring work.

I2C Wiring

Wiring is looped behind servo drive arm, so that it coils when the Pixy mount is turning left/right.

7 pin strip and socket connection.

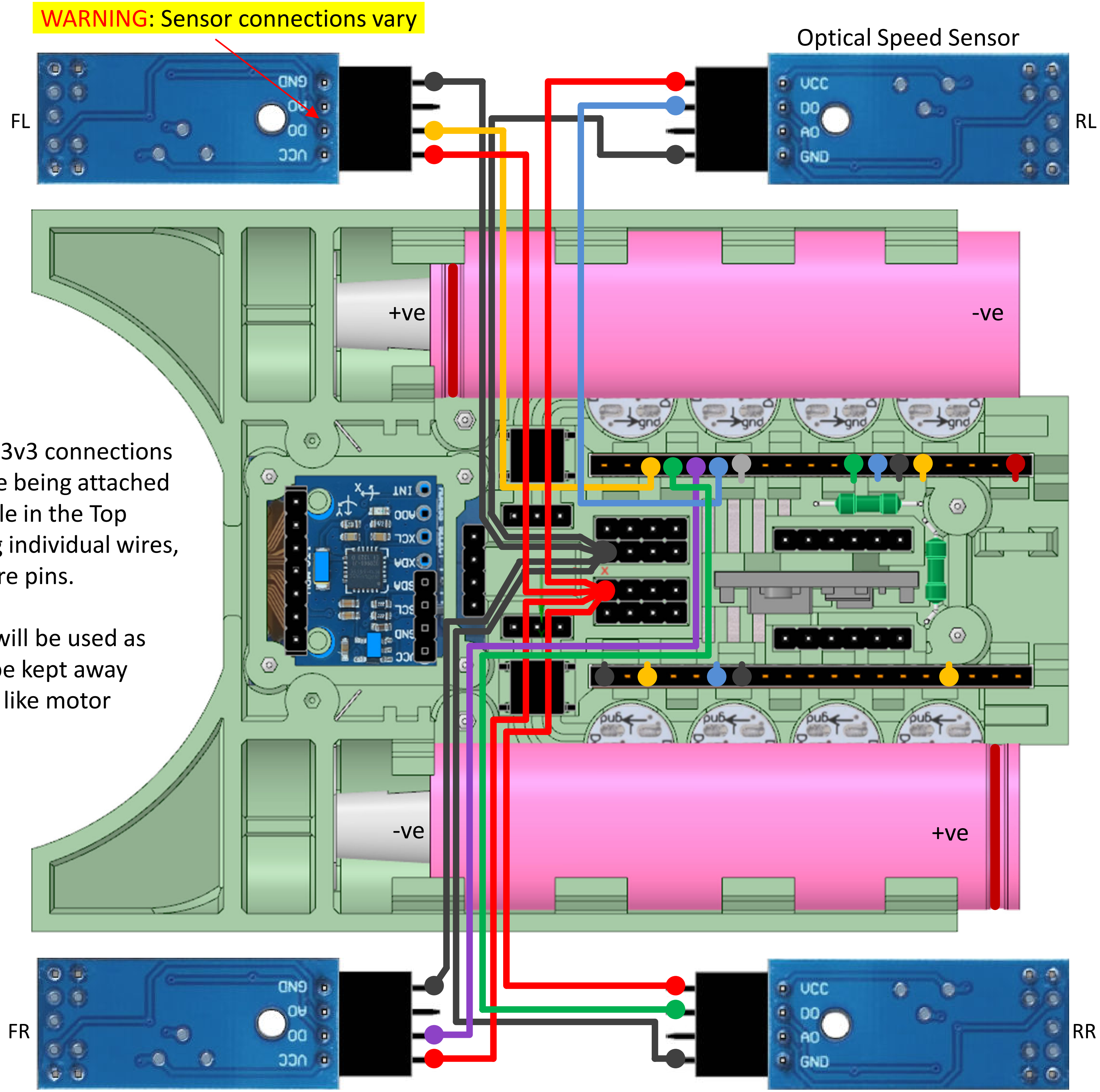




Base Plate is shown sectioned so as not to obscure Top Plate details.

We use wire wrap to make the connections, so that we can test and prove the circuit, before soldering them for improved long term reliability.

Optical Sensor Wiring



Note that the four GND and 3v3 connections are wrapped together before being attached to whatever pins are available in the Top Plate. Rather than wrapping individual wires, which would consume a more pins.

The sensor D0 connections will be used as interrupts, so they need to be kept away from fast rising noisy signals like motor PWM.

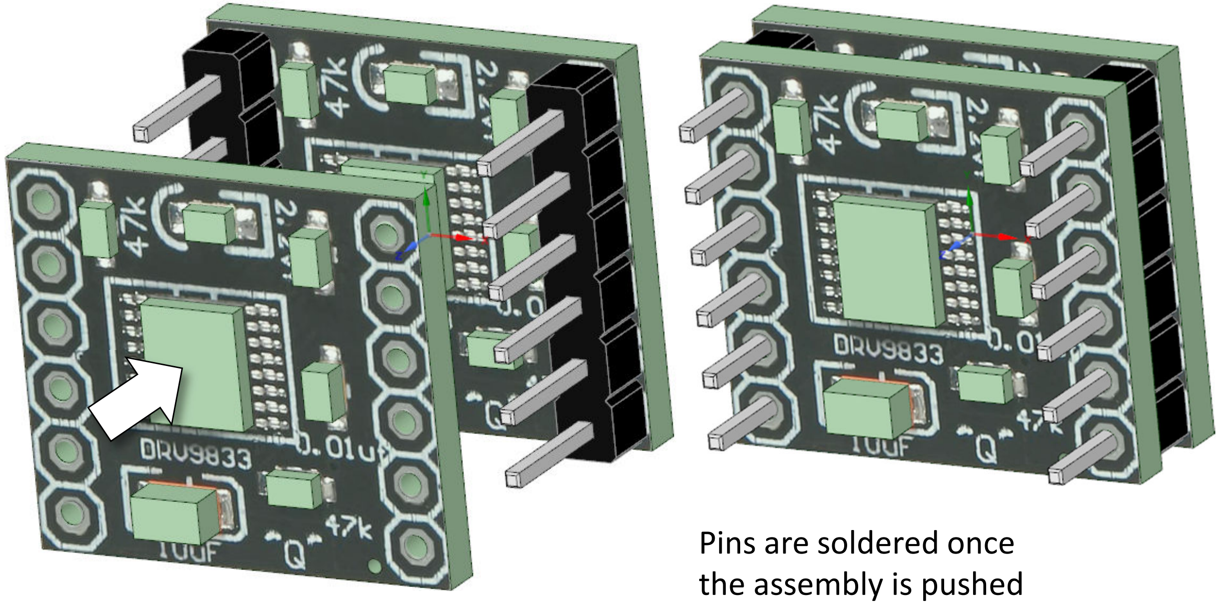
Dual H-bridge Wiring

In this design we run two H-bridge controllers in parallel in order to combined their 2A max current capability, and half their MOS-FETs on-channel resistance.

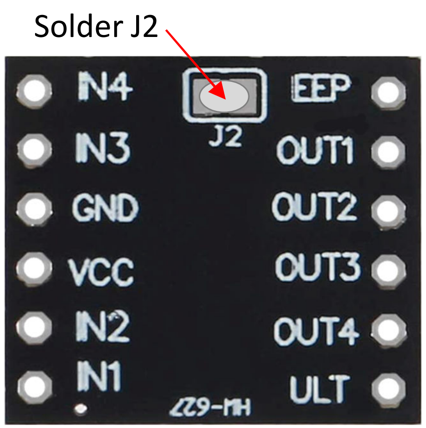
This is achieved by creating a sandwich using two controllers pcb's with 6 pin strips between them. The pins are soldered into each pcb, and the protruding longer pins are used as the connection points.

These sandwich modules are mounted in the design such that cooling airflow can circulate over the DRV8833 driver ICs.

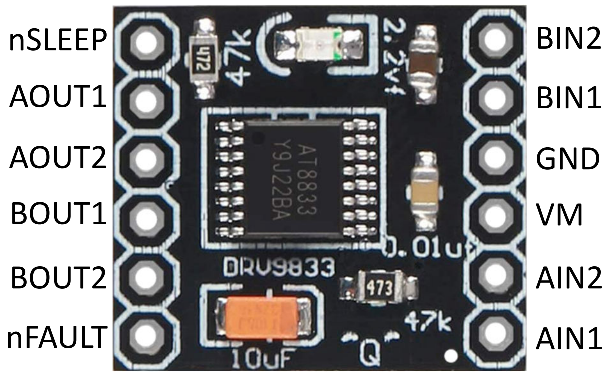
The nSLEEP and nFAULT pins are not connected; however, the nSLEEP pin is pulled HIGH by an on-board 47kΩ resistor when you bridge J2 with a blob of solder.

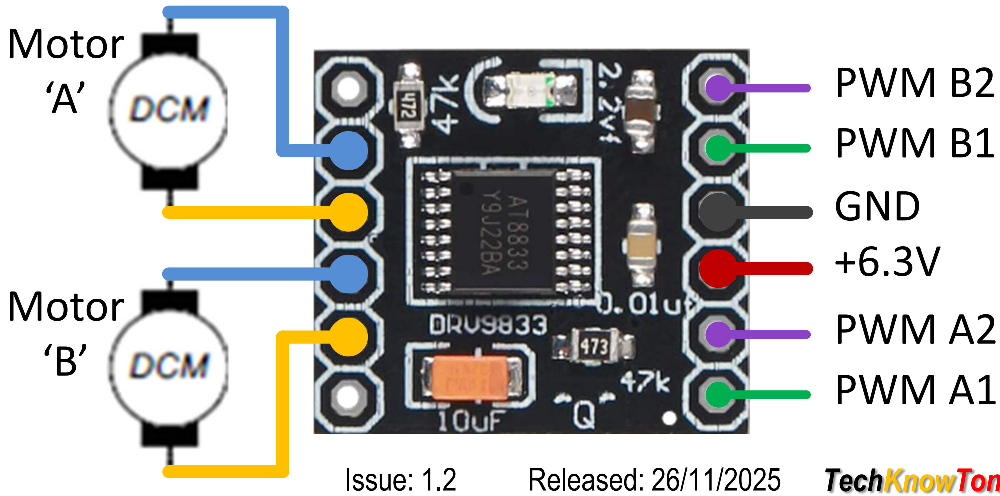


Pins are soldered once the assembly is pushed together.

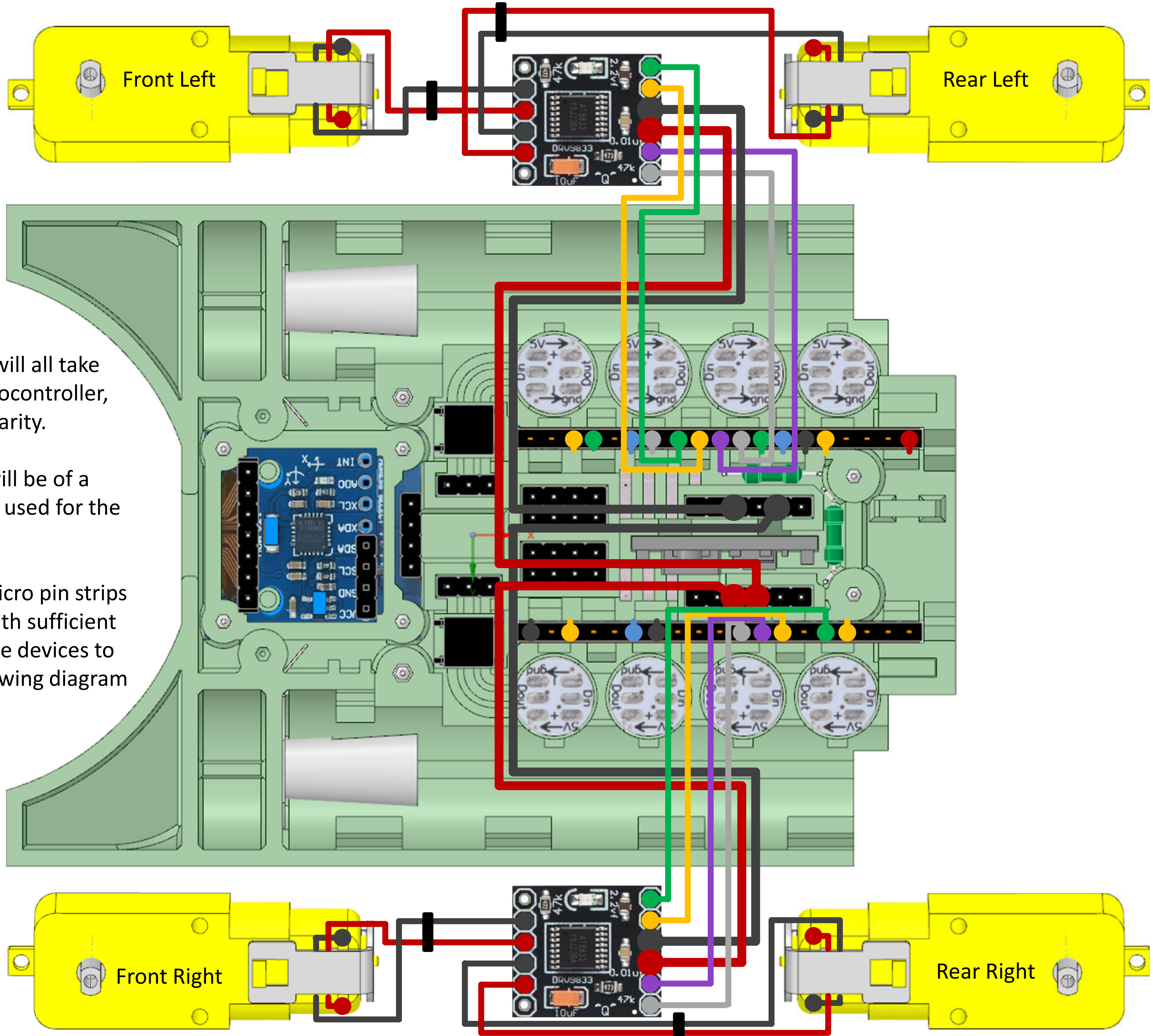


Solder J2 **before** you make the sandwich assembly!





Motor H-bridge Wiring



The actual routing of the wiring will all take place within the area of the microcontroller, but shown here spread out for clarity.

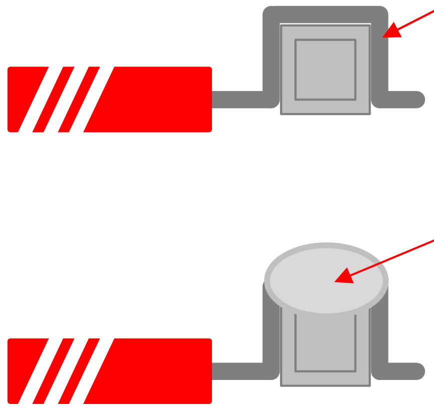
The red and black power wires will be of a thicker grade than the wire wrap used for the PWM signal connections.

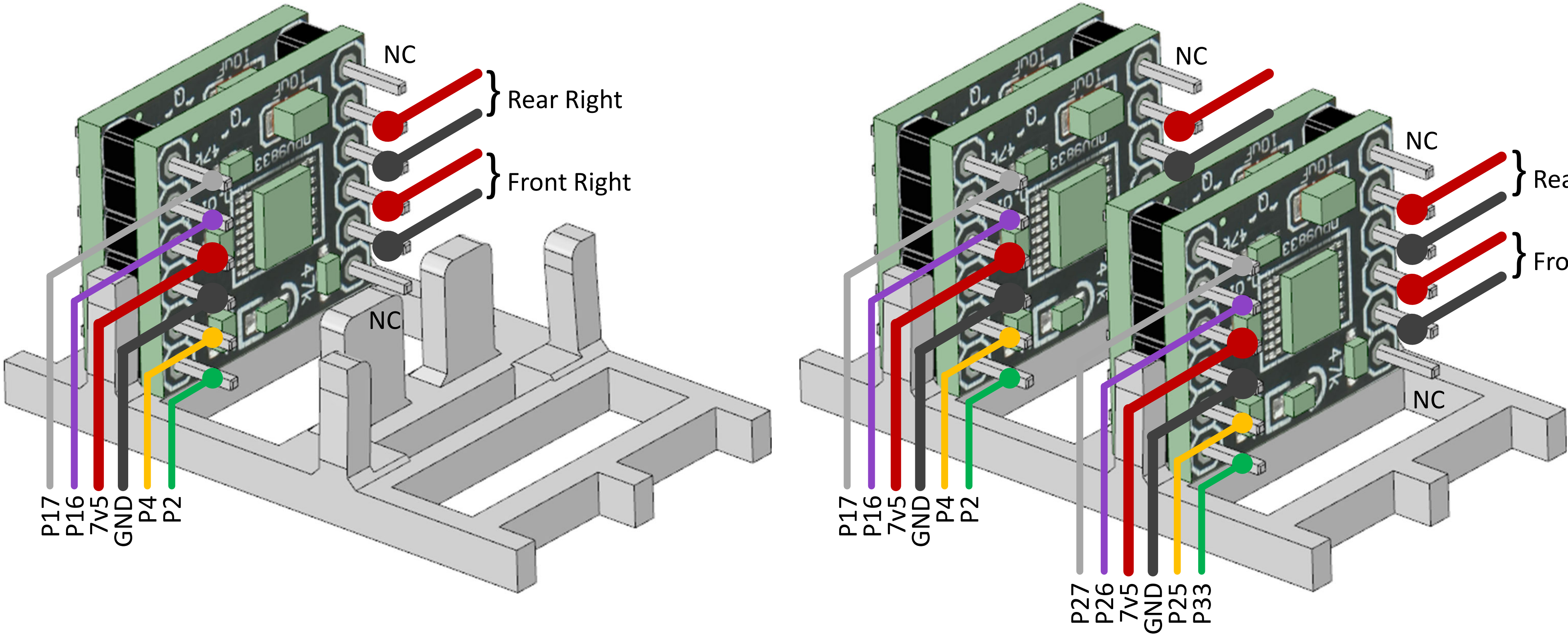
Start by attaching wires to the micro pin strips and power pins GND and 7V5, with sufficient length to reach up to the H-bridge devices to be mounted above. See the following diagram for H-bridge wiring.

Dual H-bridge Wiring

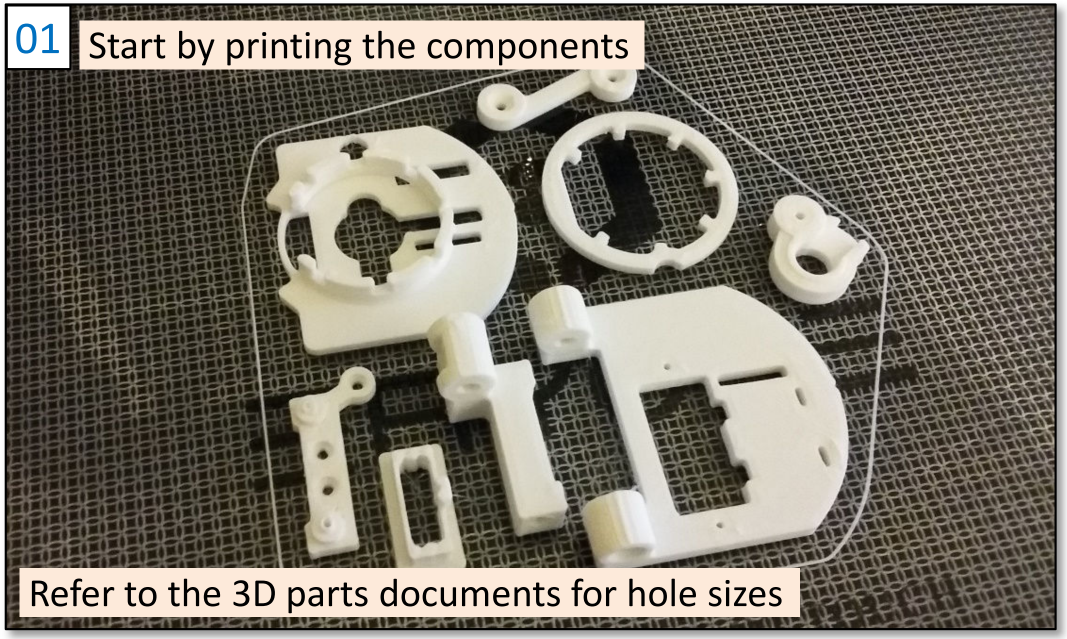
The wires you have connected in the previous diagram will be fed to the respective H-bridge drivers above. Start with the right-hand driver and make off those connections first using wire wrap for the PWM signals and solder the power connections.

I pre-soldered the ends of the power wires, then formed them around a pin as shown here before placing them next to a pin and soldering them together. This method works well for thick wire, which might cause a short between the pins when soldered. It also makes it easier to remove the wires later if one of the H-bridges happens to fail and needs replacing.



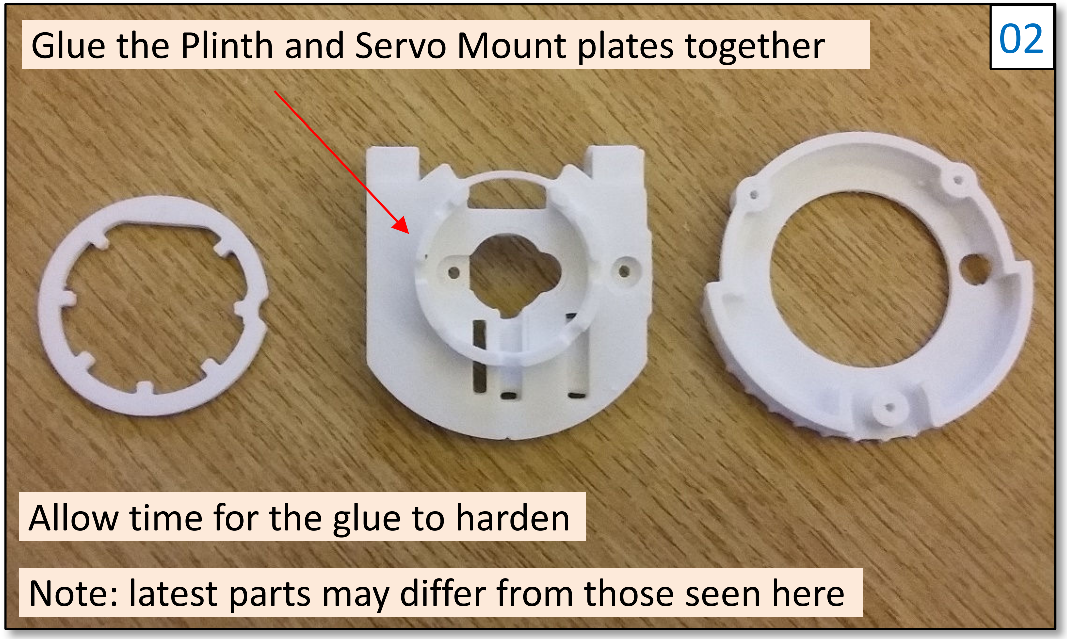


Build Sequence



01 Start by printing the components

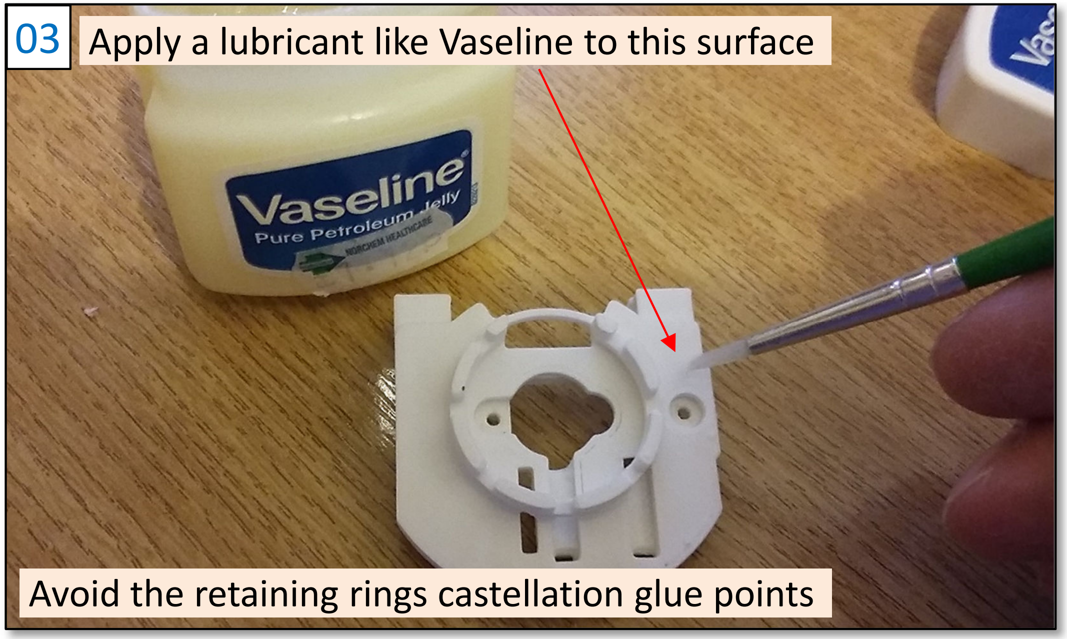
Refer to the 3D parts documents for hole sizes



02 Glue the Plinth and Servo Mount plates together

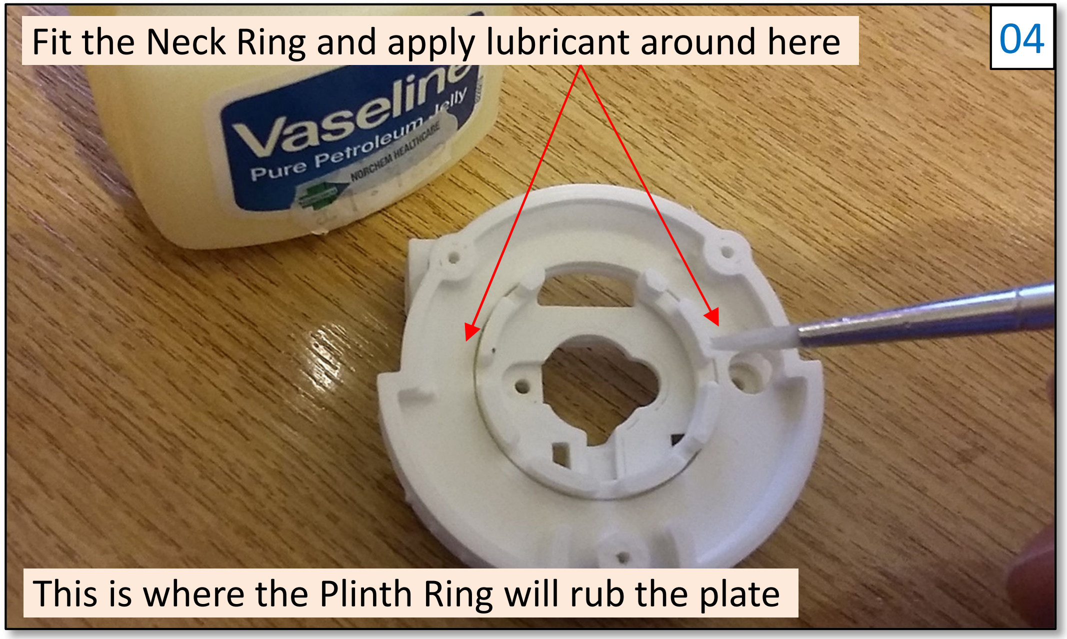
Allow time for the glue to harden

Note: latest parts may differ from those seen here



03 Apply a lubricant like Vaseline to this surface

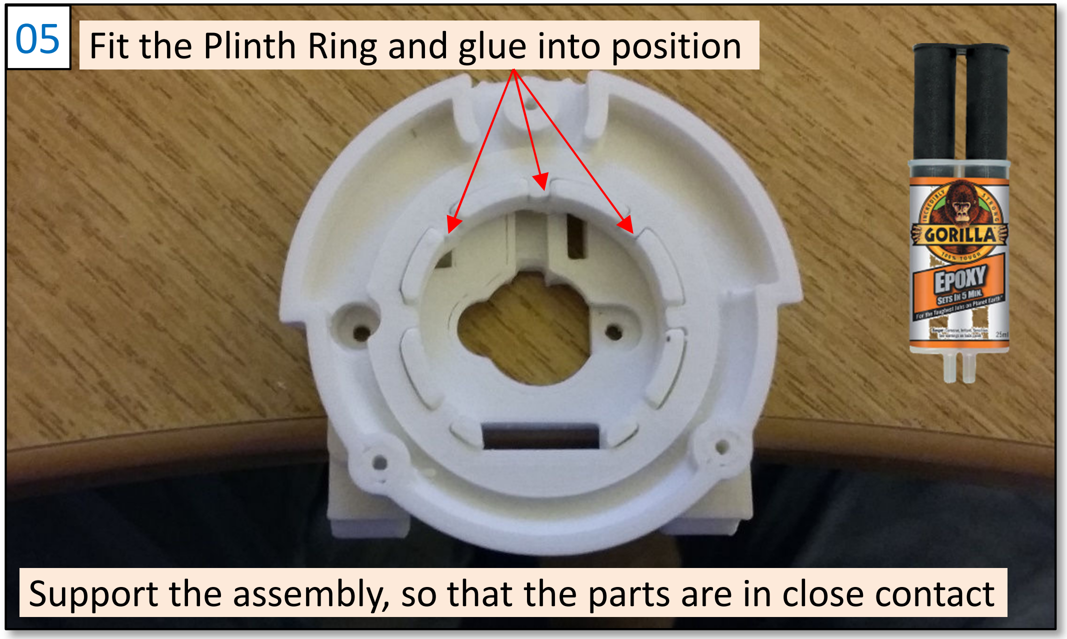
Avoid the retaining rings castellation glue points

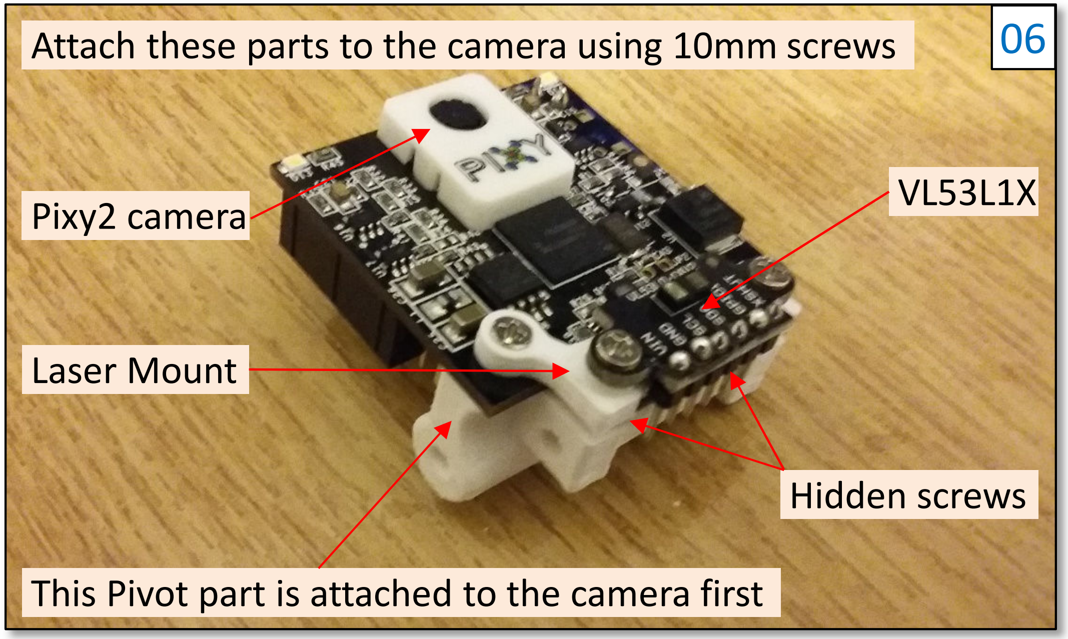


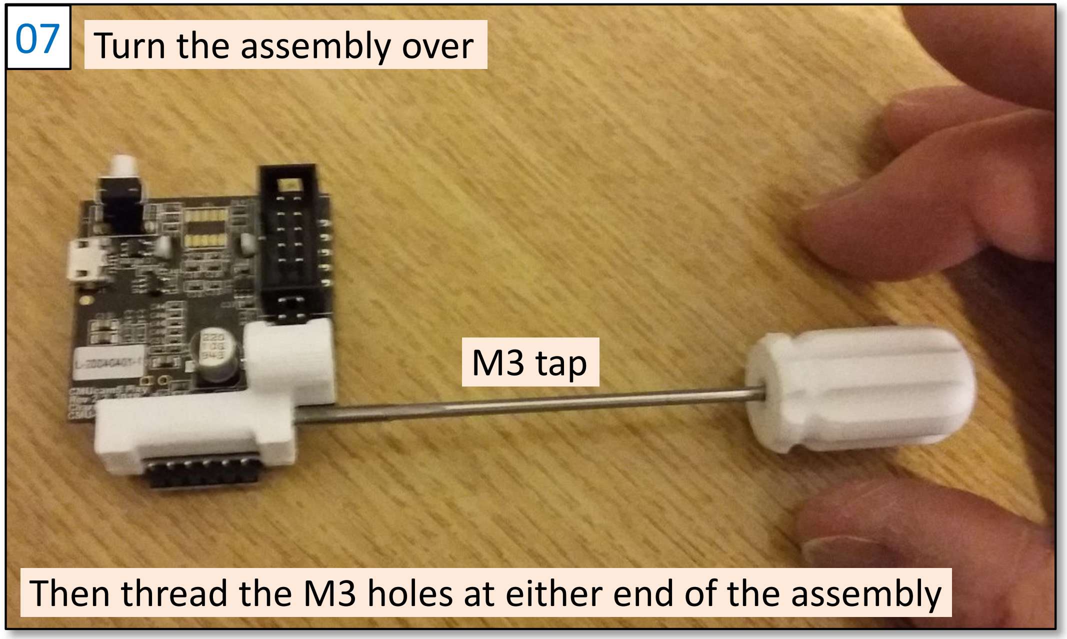
04 Fit the Neck Ring and apply lubricant around here

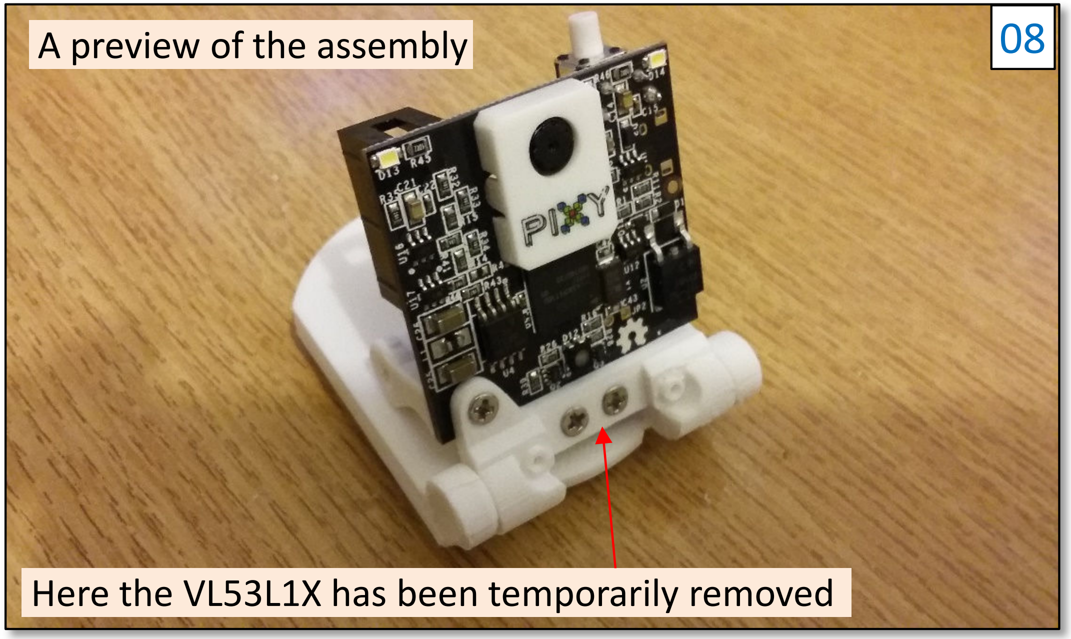
This is where the Plinth Ring will rub the plate

Wiring Sequence

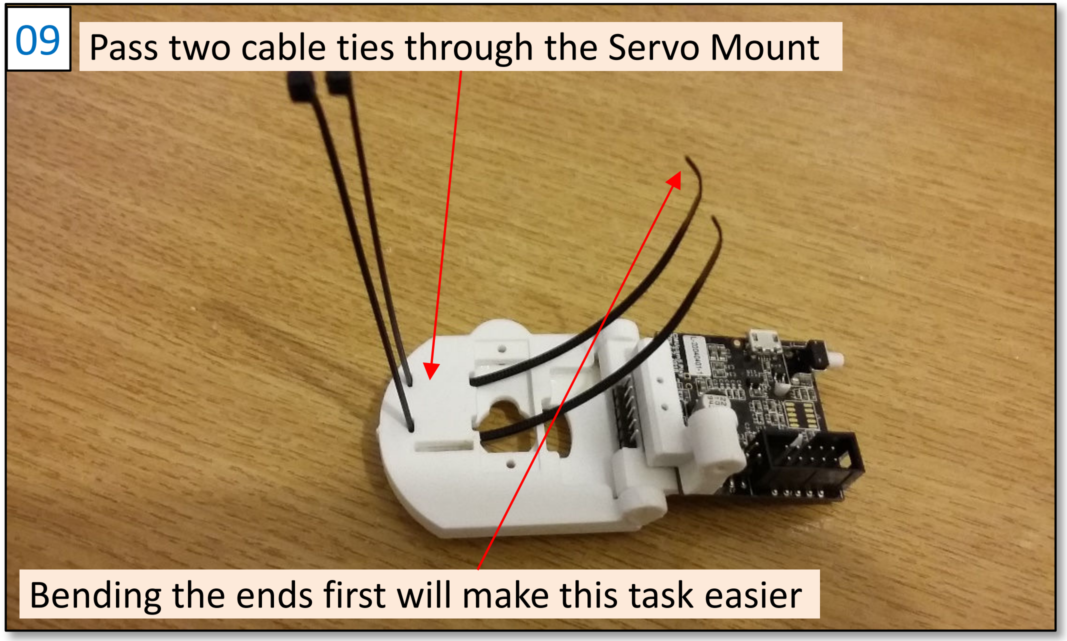


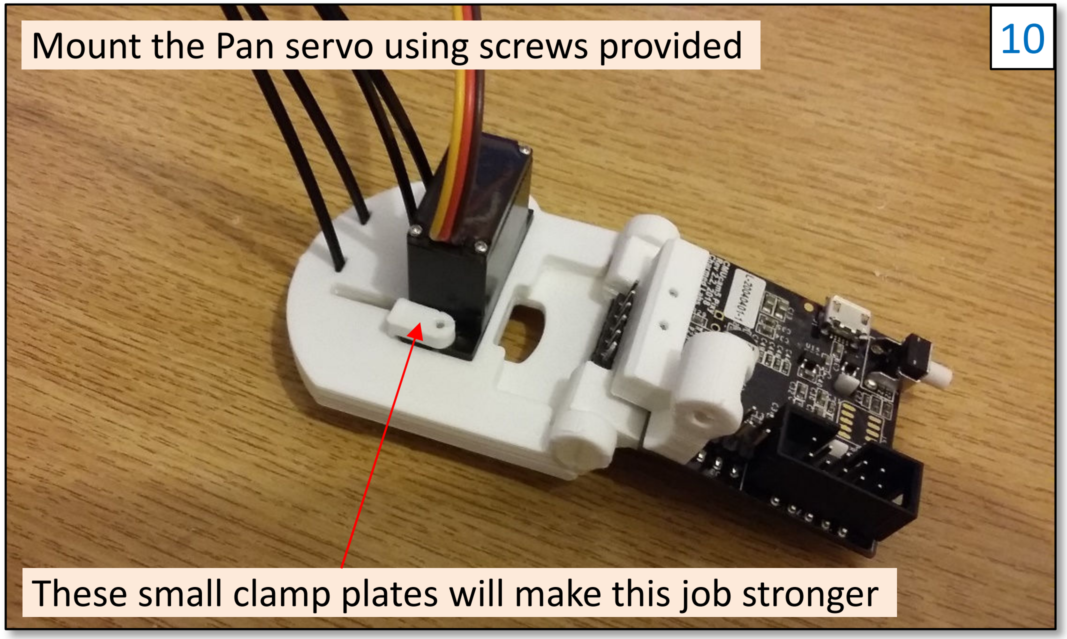


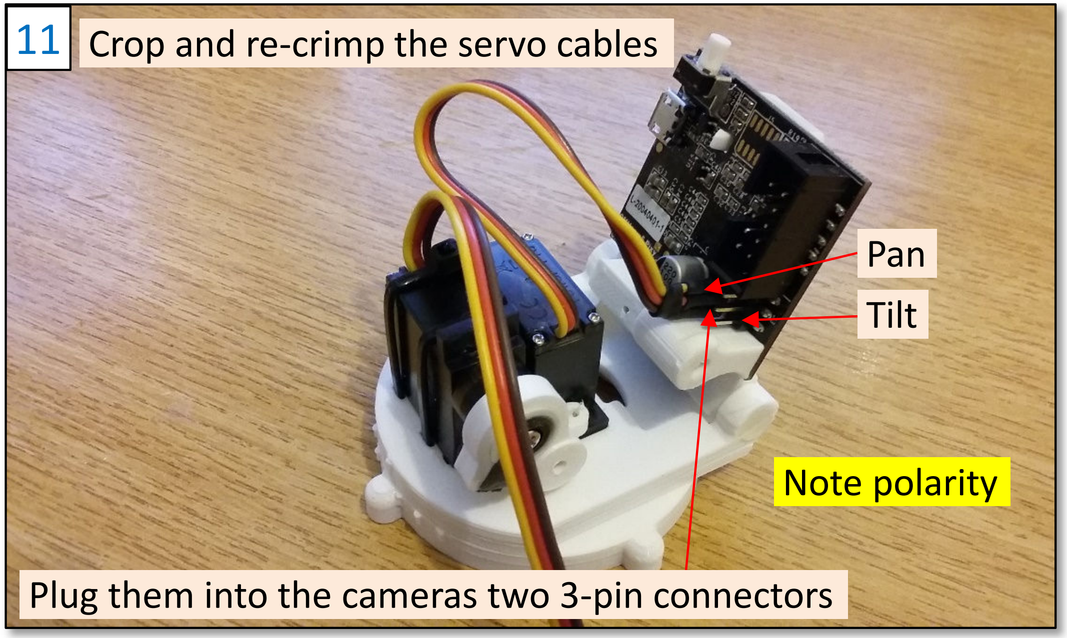


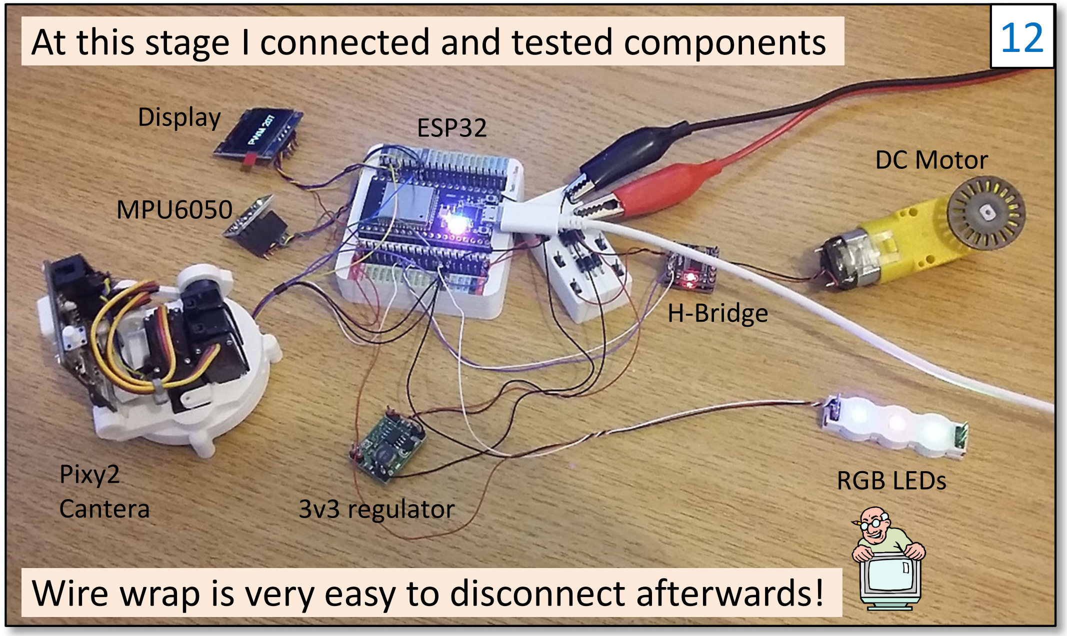


Wiring Sequence

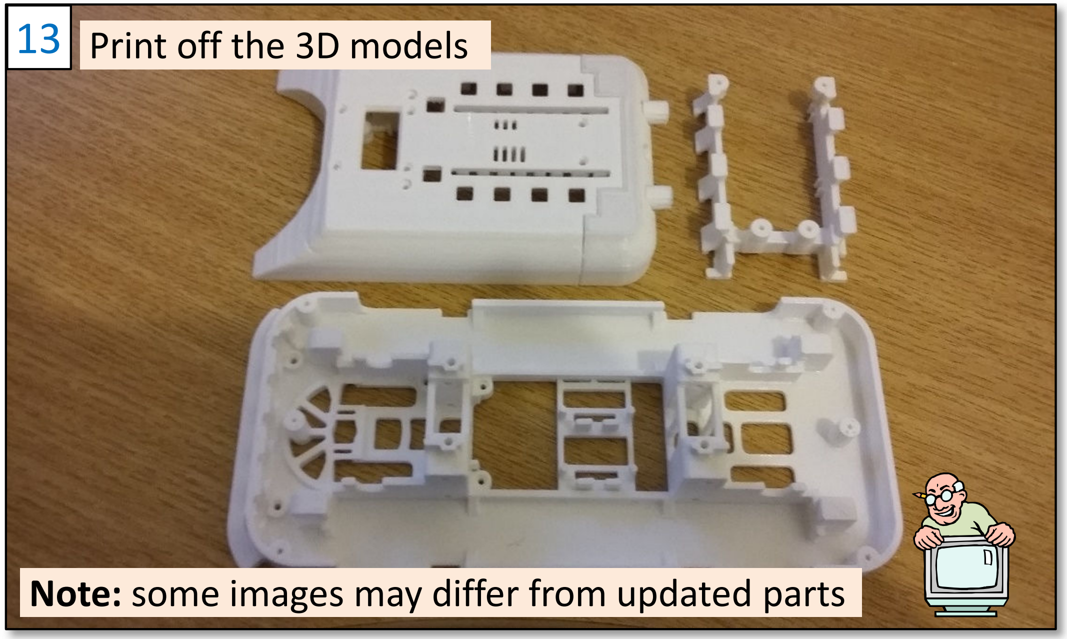


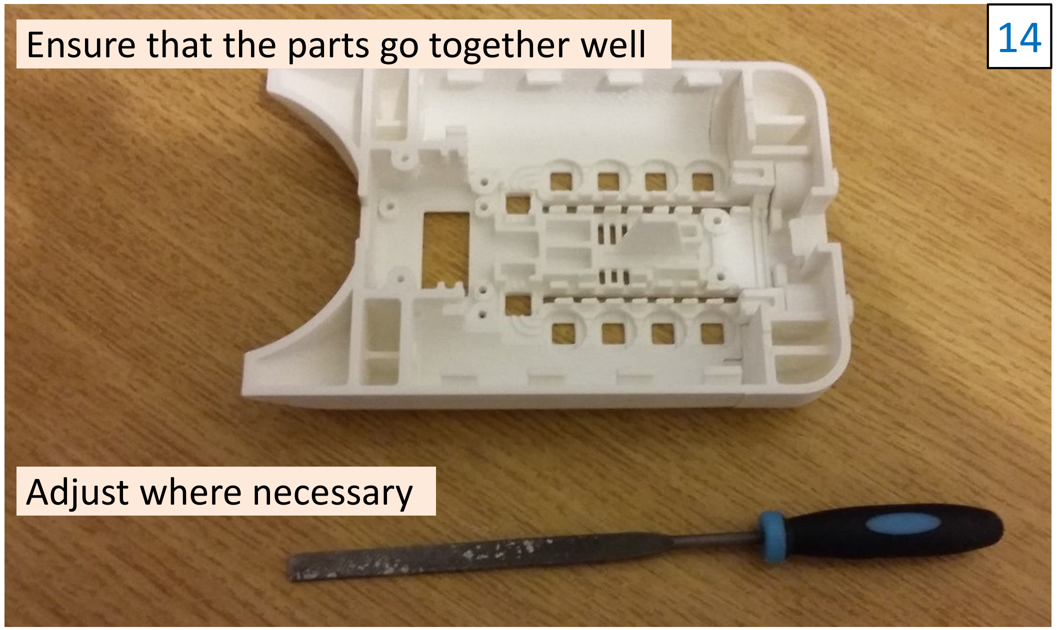


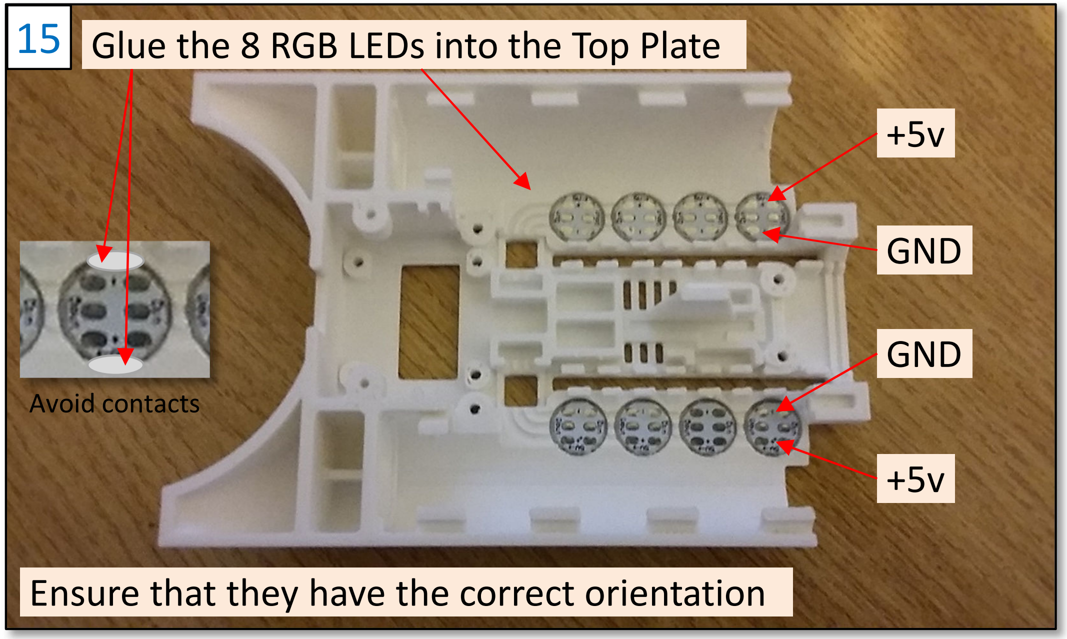


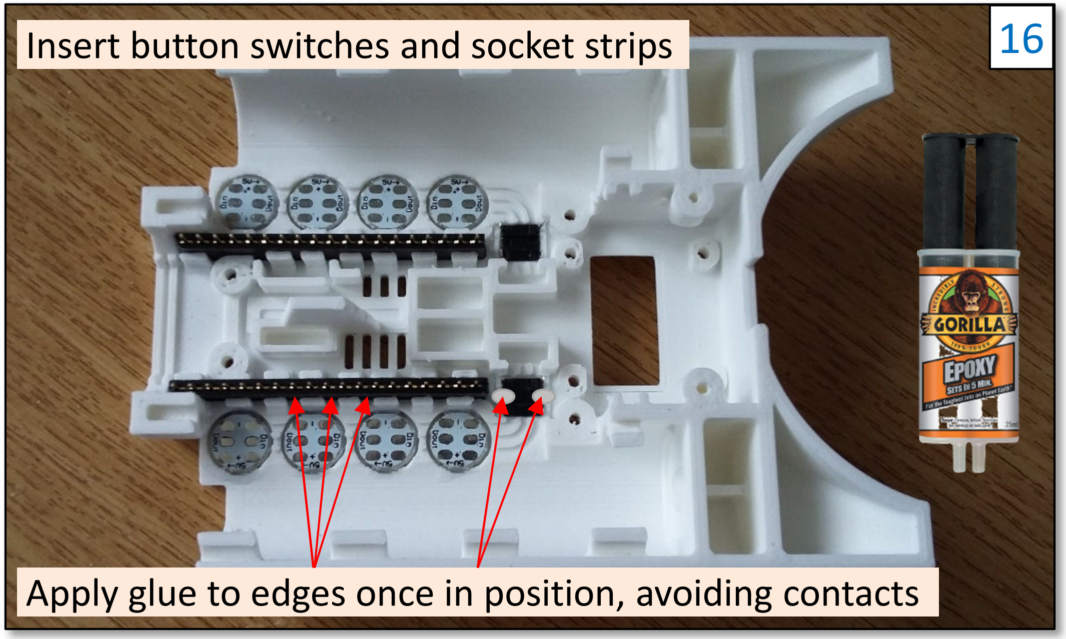


Wiring Sequence

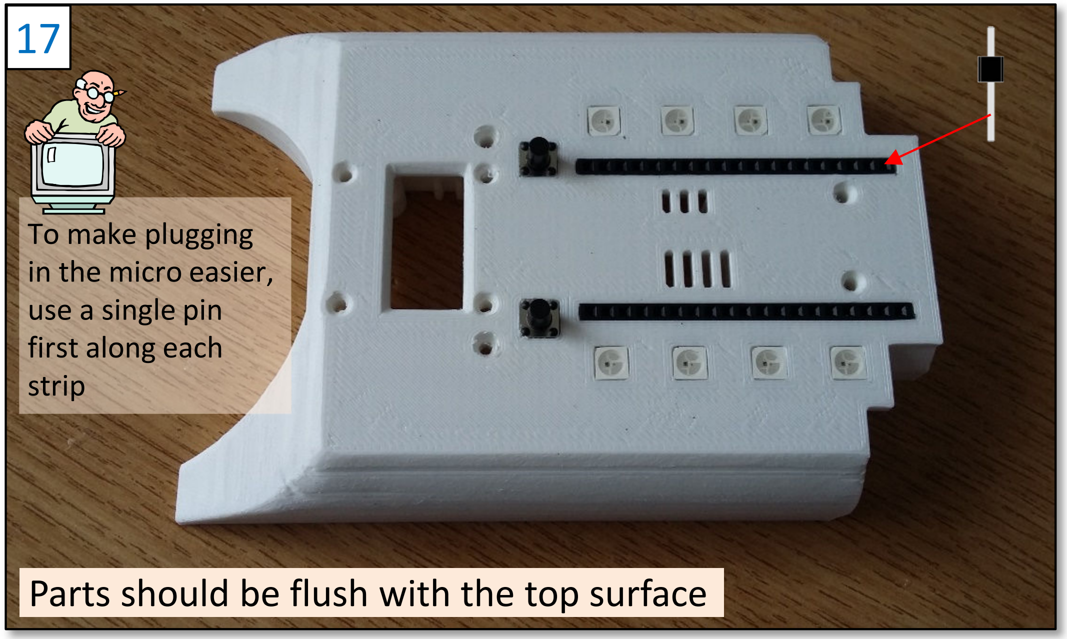


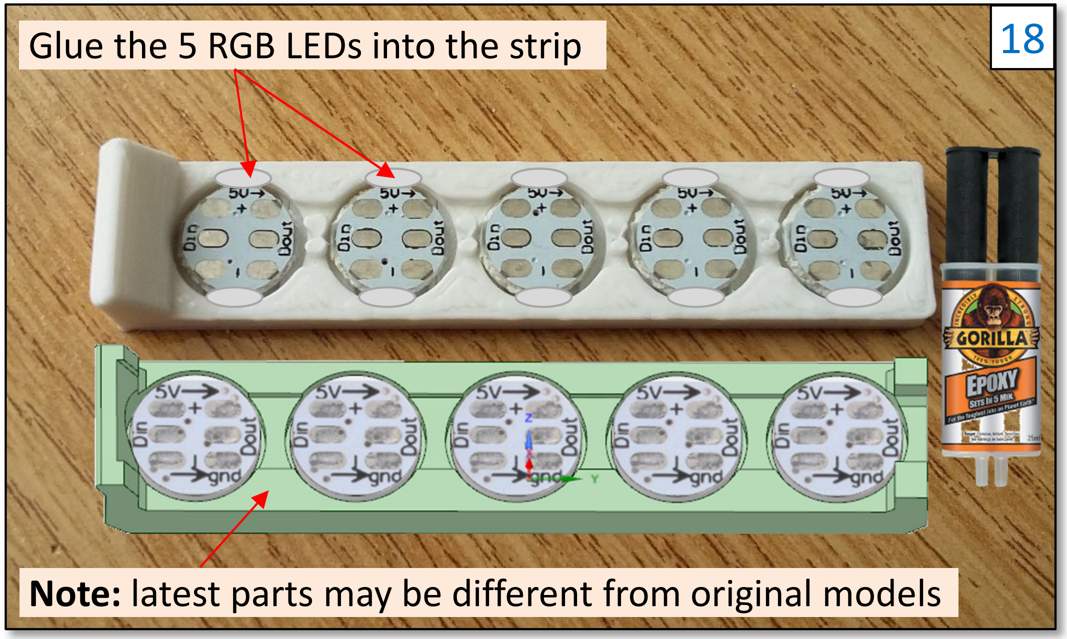


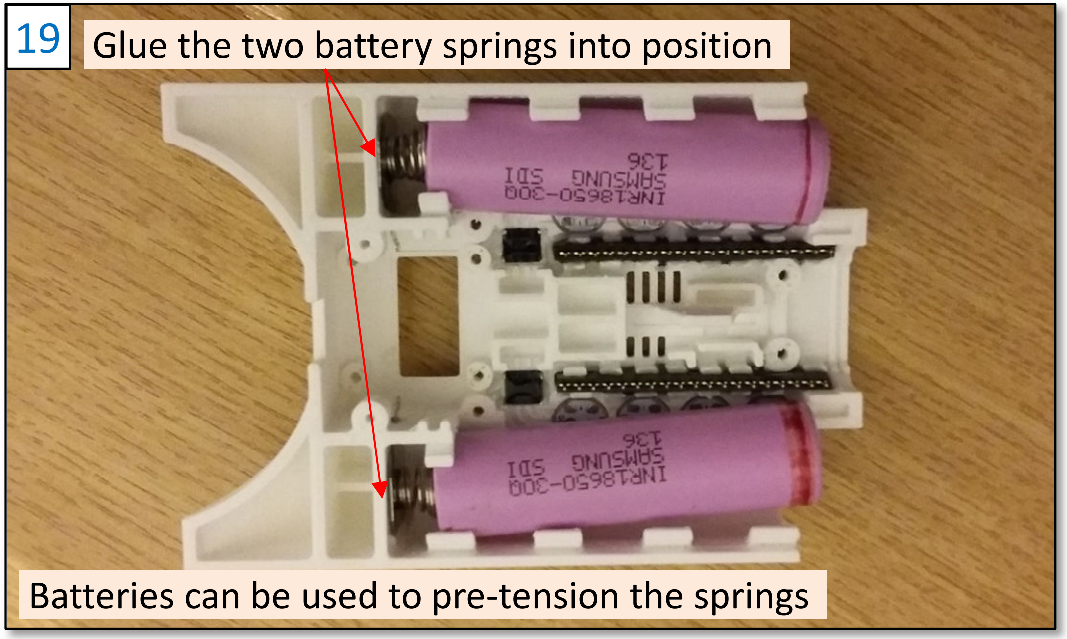


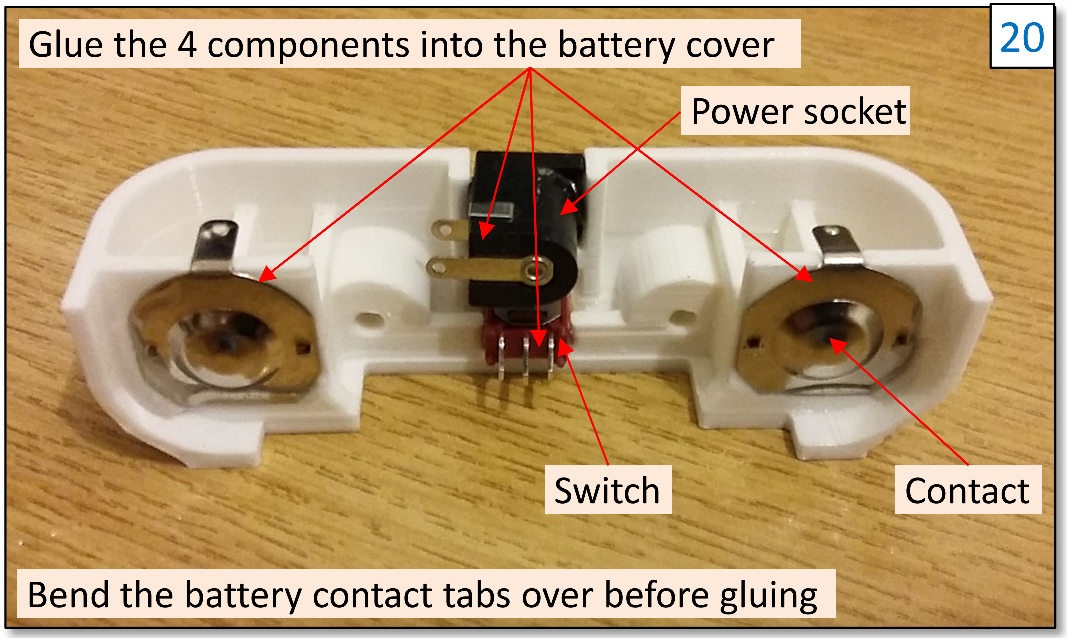


Wiring Sequence



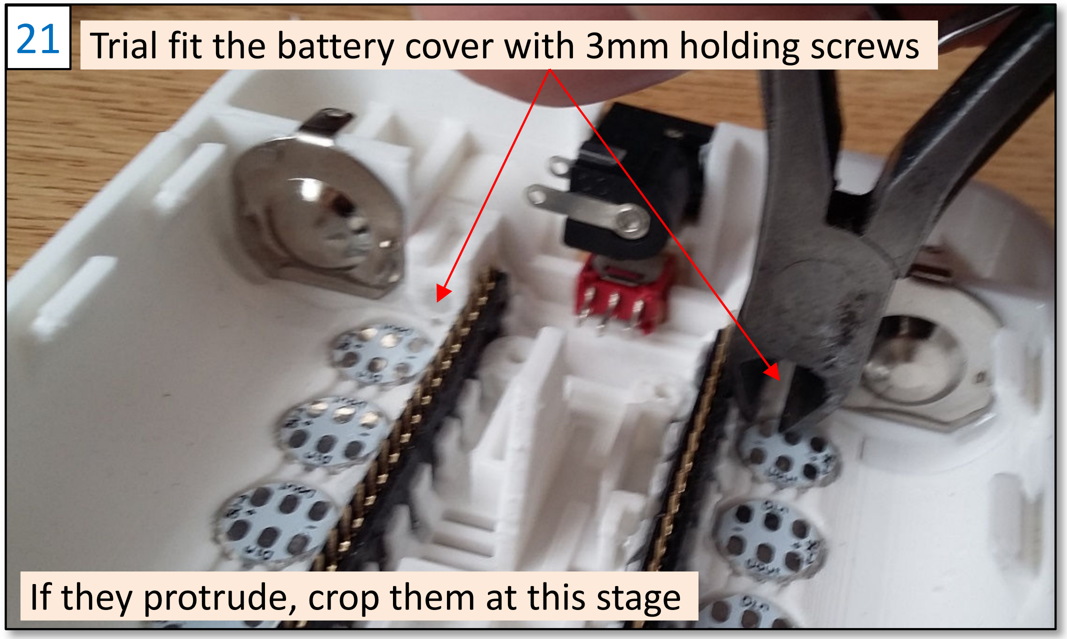






Wiring Sequence

21 Trial fit the battery cover with 3mm holding screws

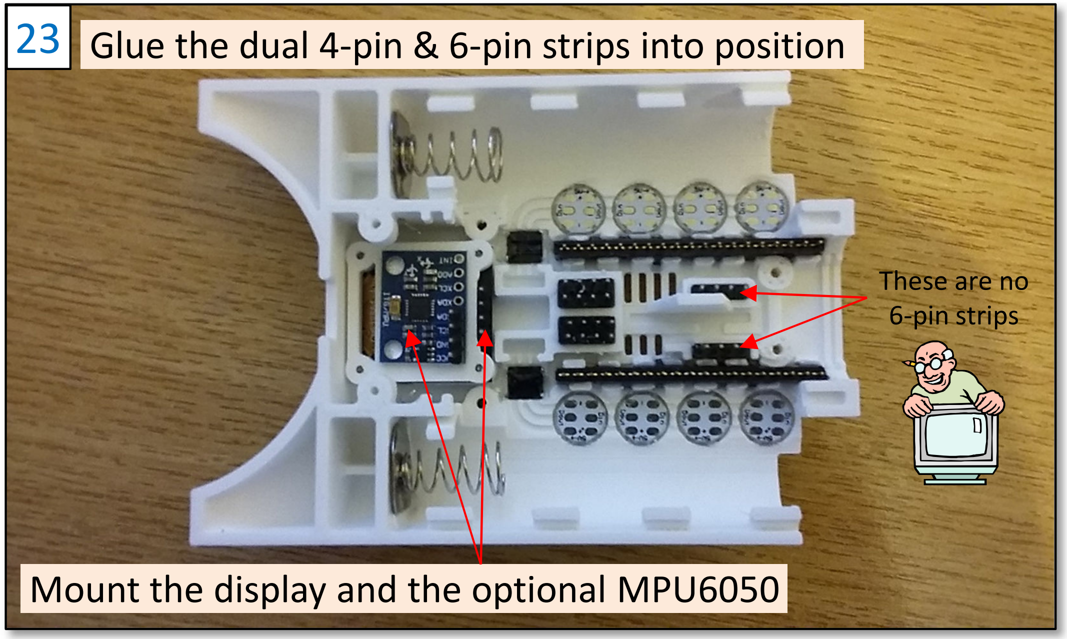


If they protrude, crop them at this stage

22 Make up the pin strips and 3-pin RGB LED leads

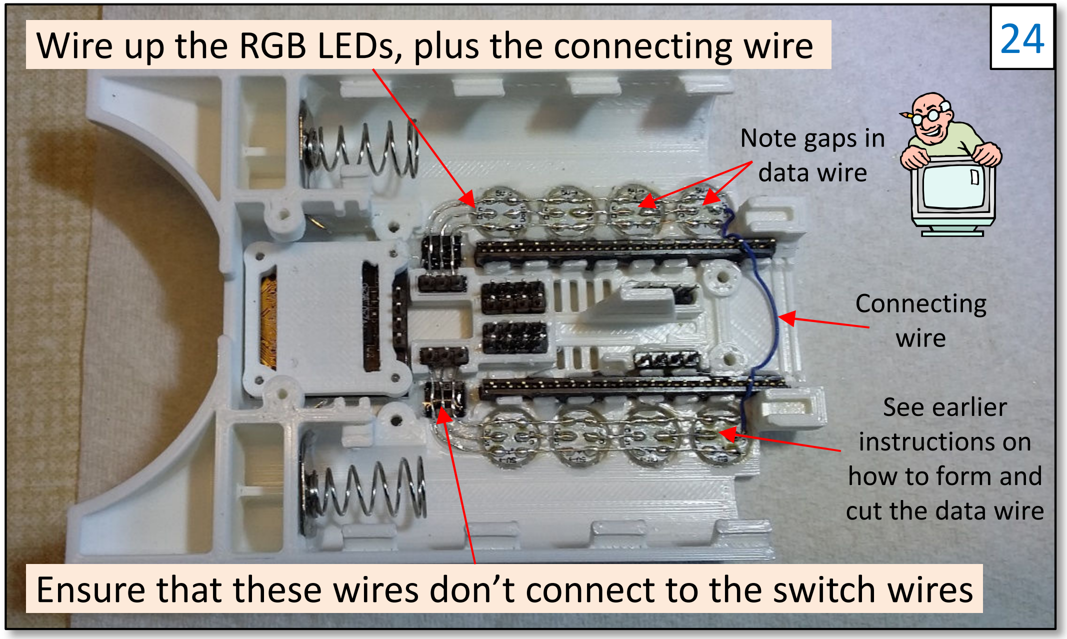
I used 4-pin strips, but these should now be 6-pin

23 Glue the dual 4-pin & 6-pin strips into position



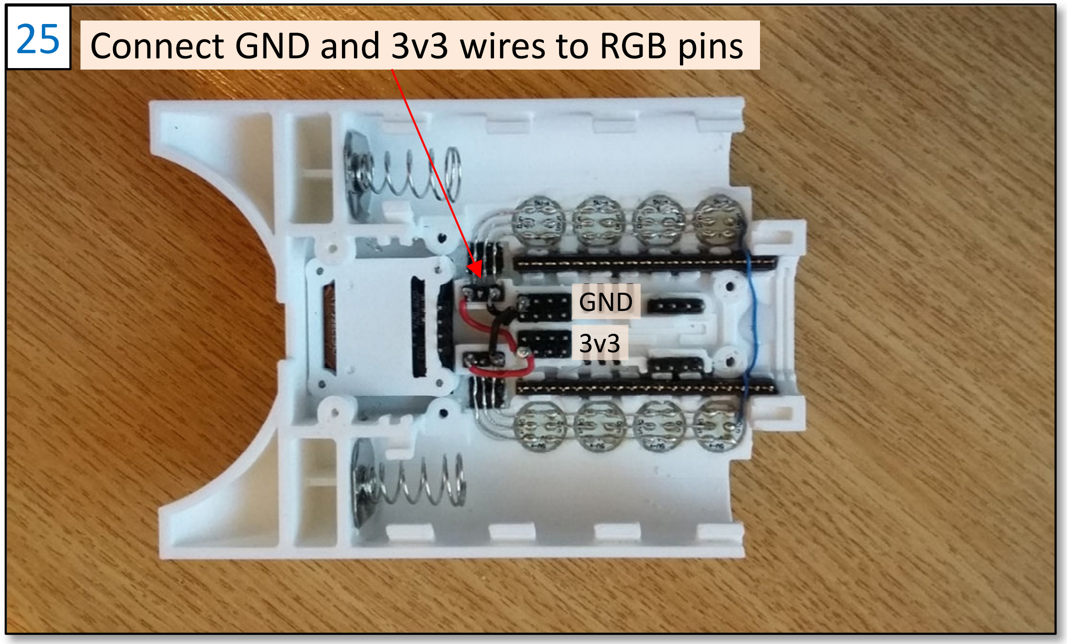
Mount the display and the optional MPU6050

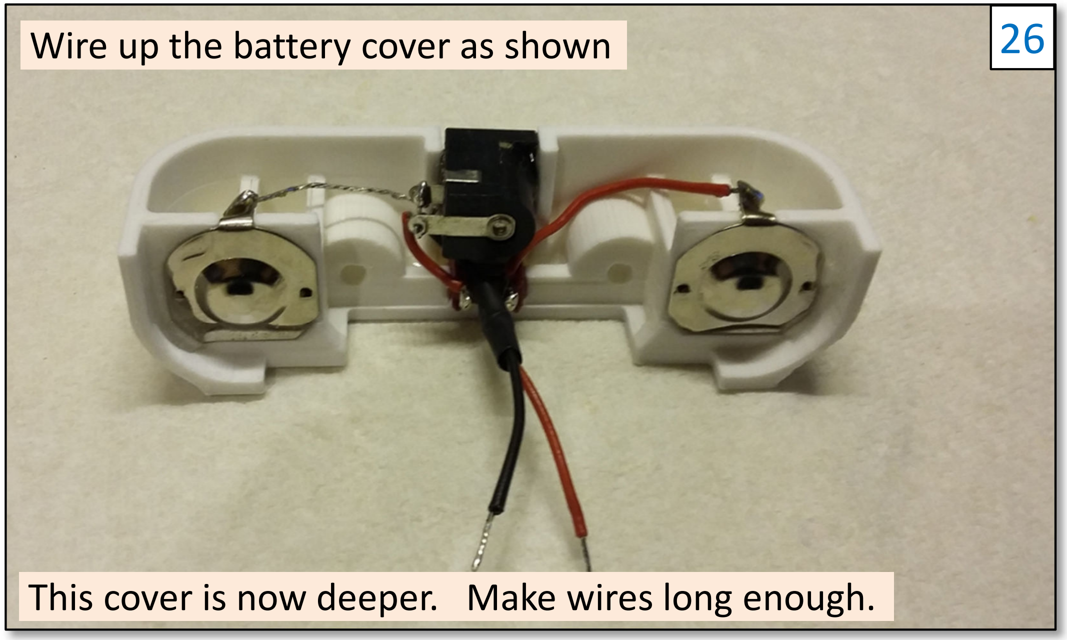
24 Wire up the RGB LEDs, plus the connecting wire

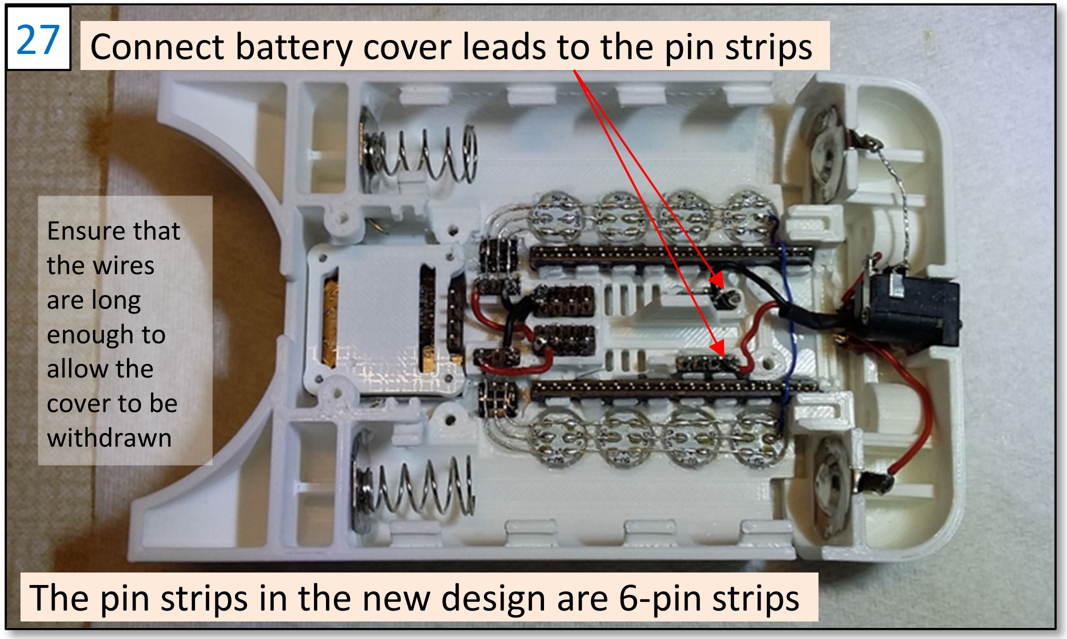


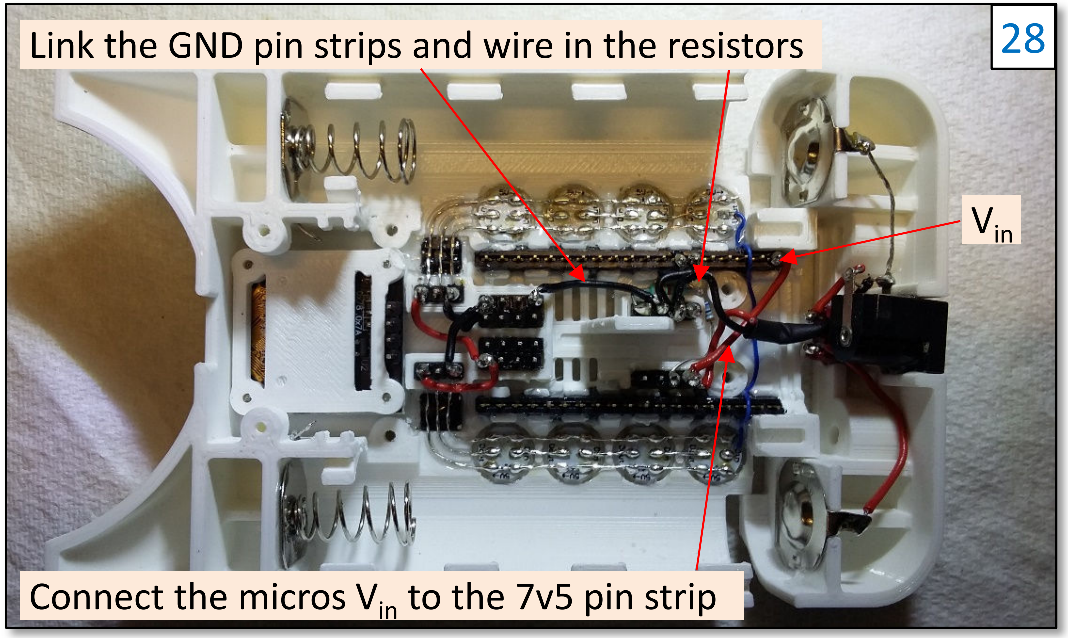
Ensure that these wires don’t connect to the switch wires

Wiring Sequence

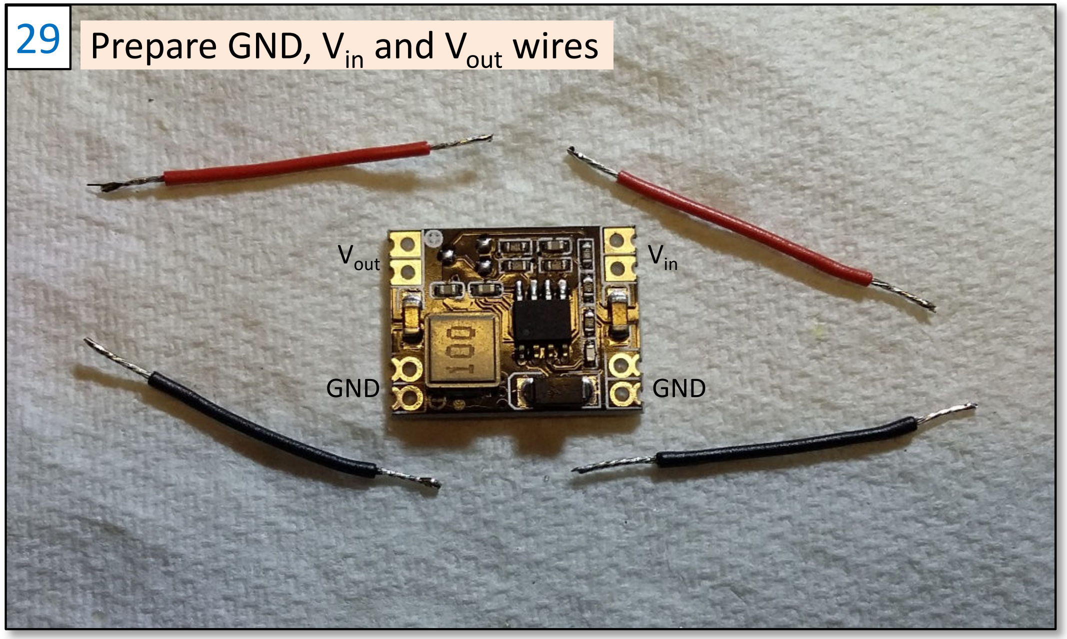


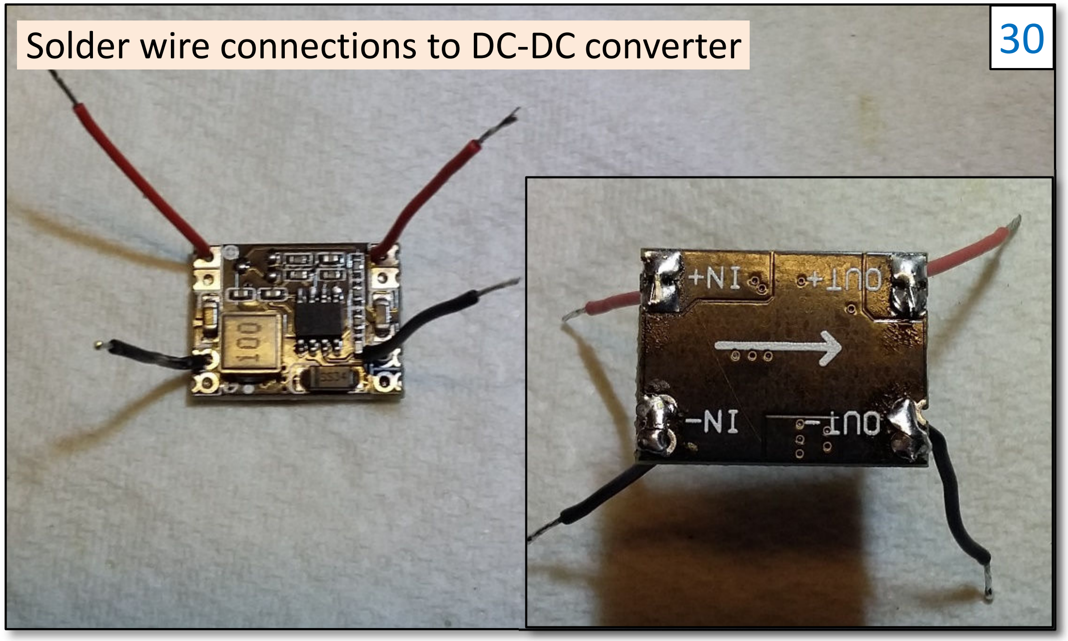


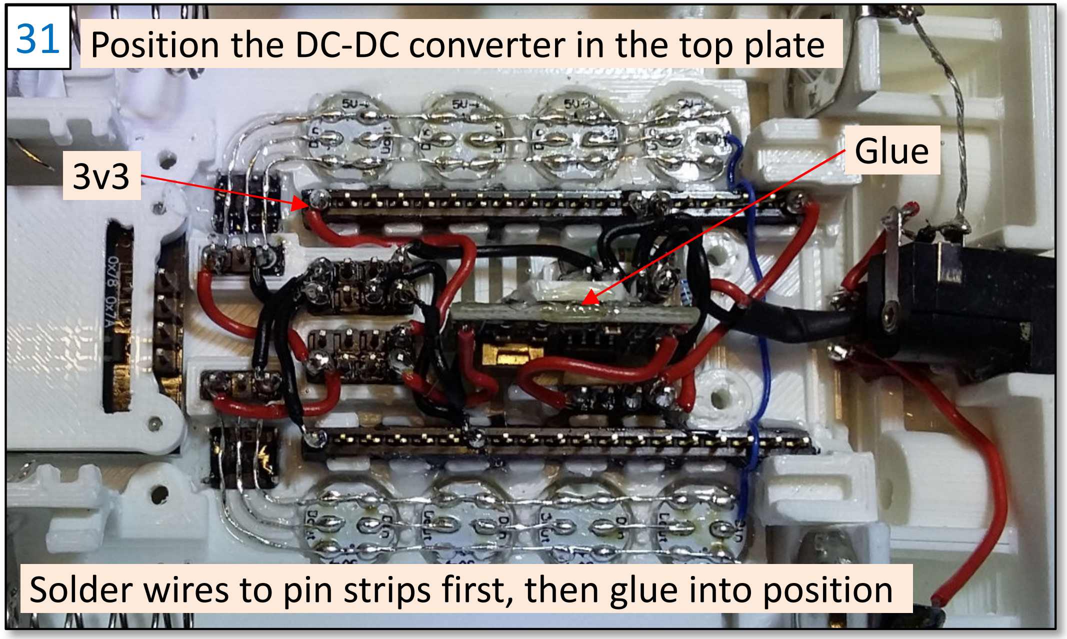


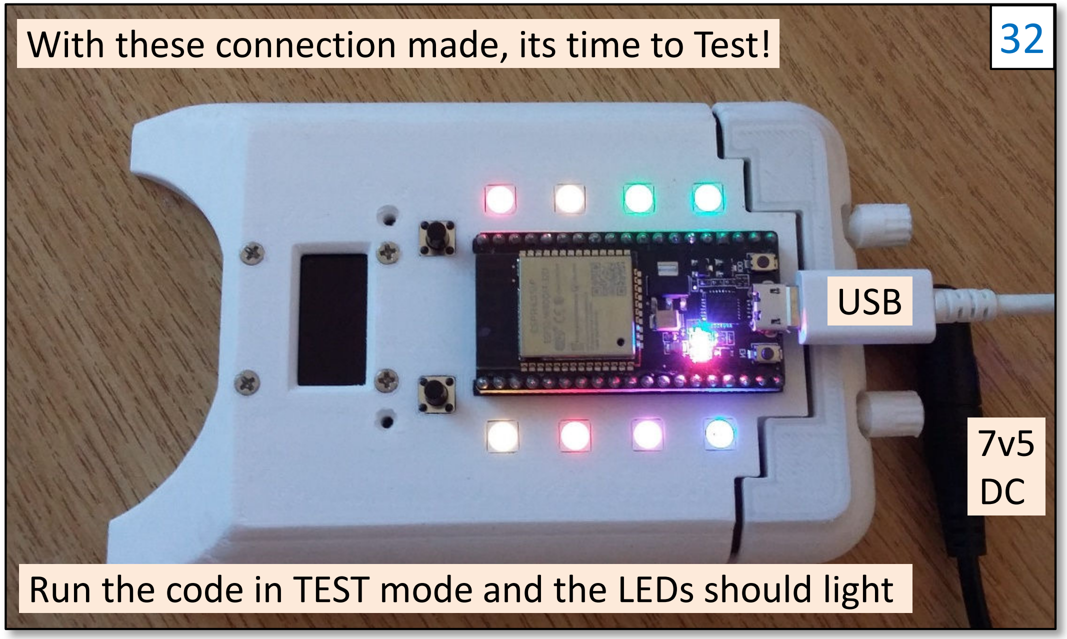


Wiring Sequence

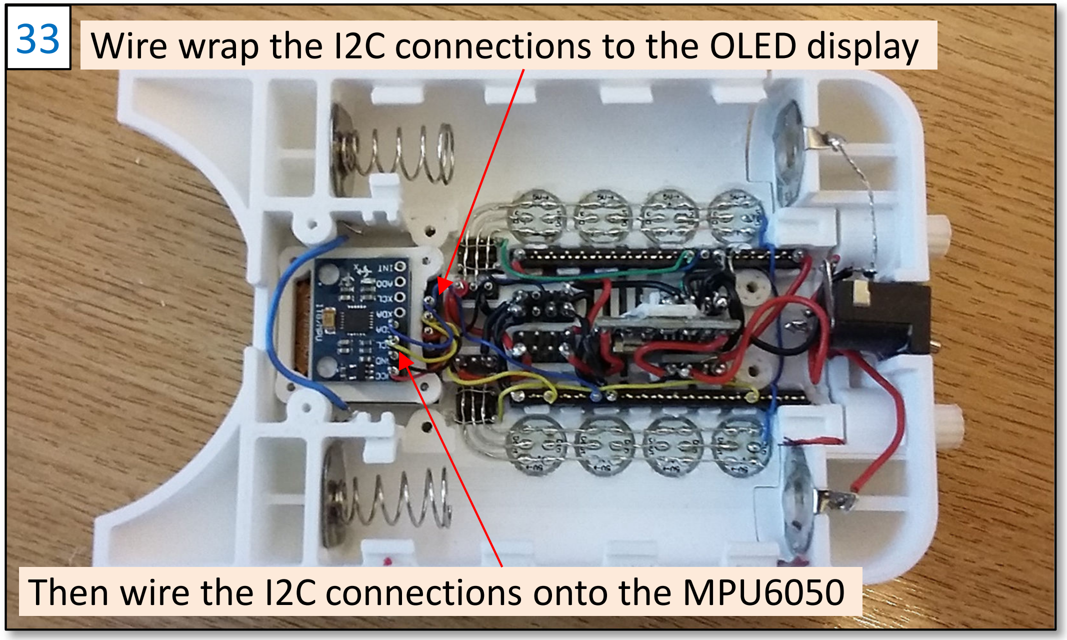






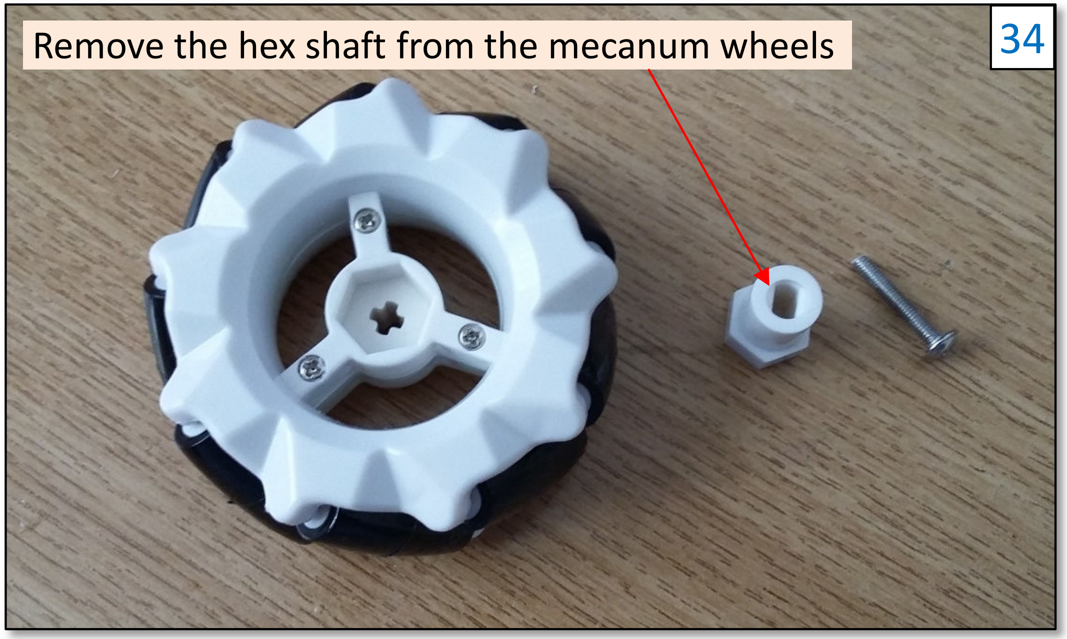


Wiring Sequence

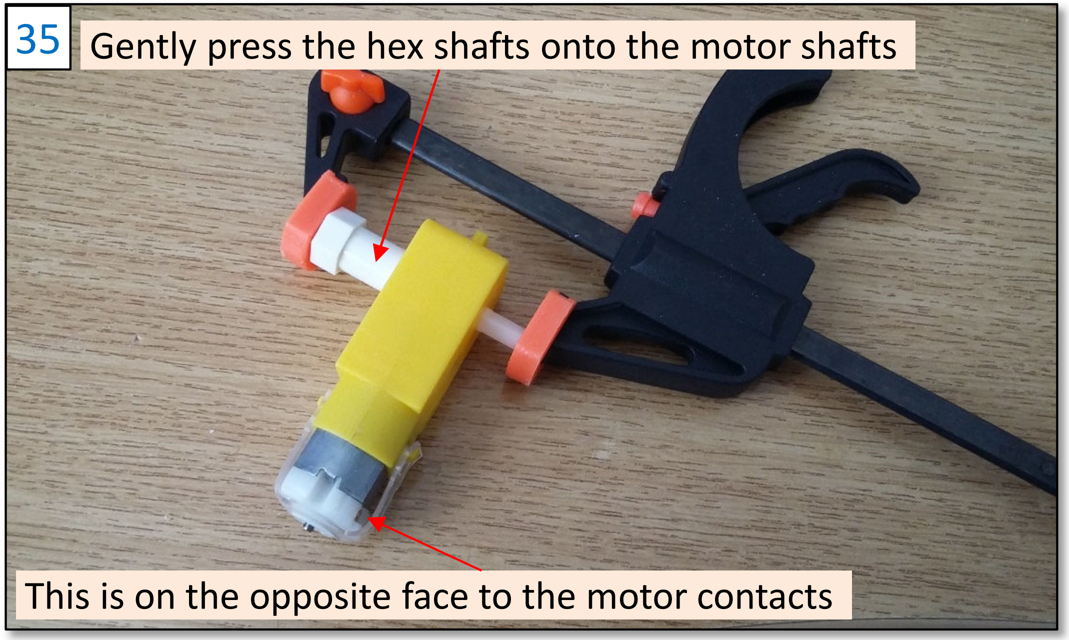


33 Wire wrap the I2C connections to the OLED display

Then wire the I2C connections onto the MPU6050

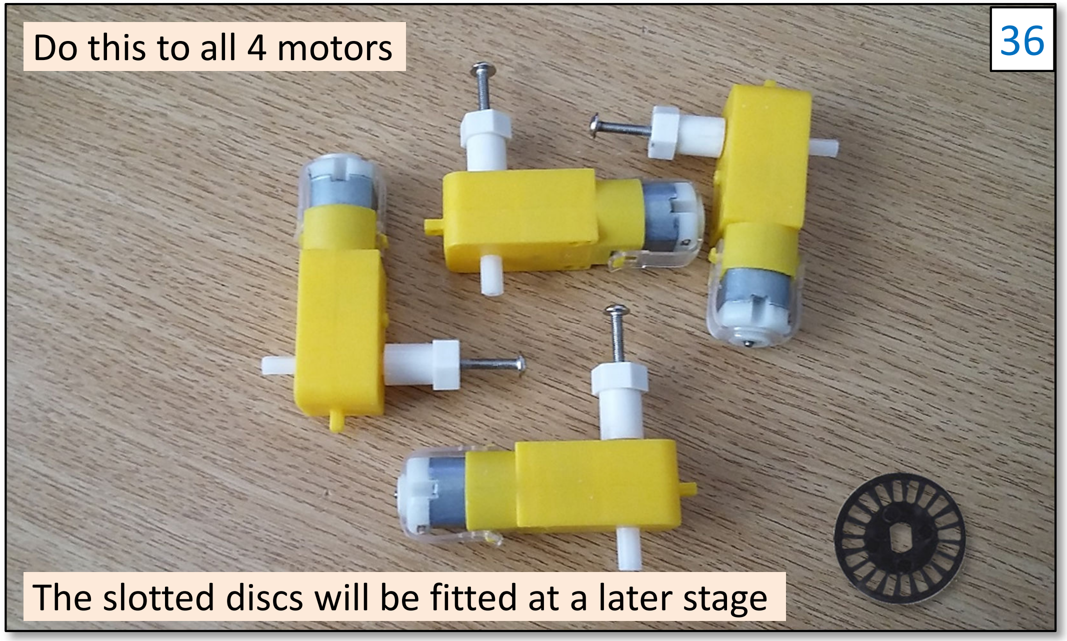


34 Remove the hex shaft from the mecanum wheels



35 Gently press the hex shafts onto the motor shafts

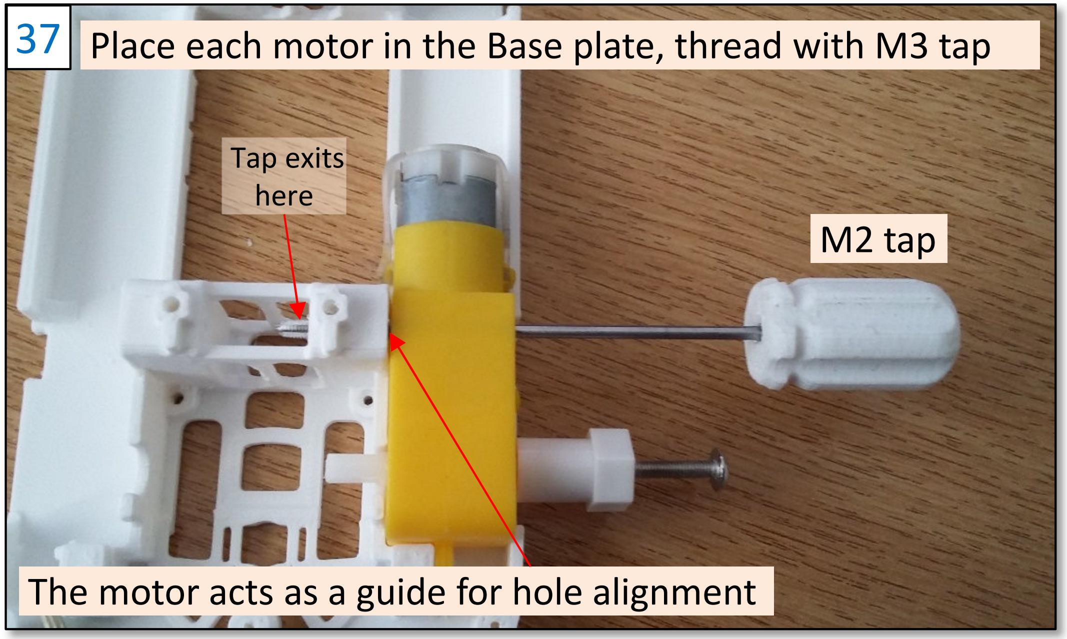
This is on the opposite face to the motor contacts

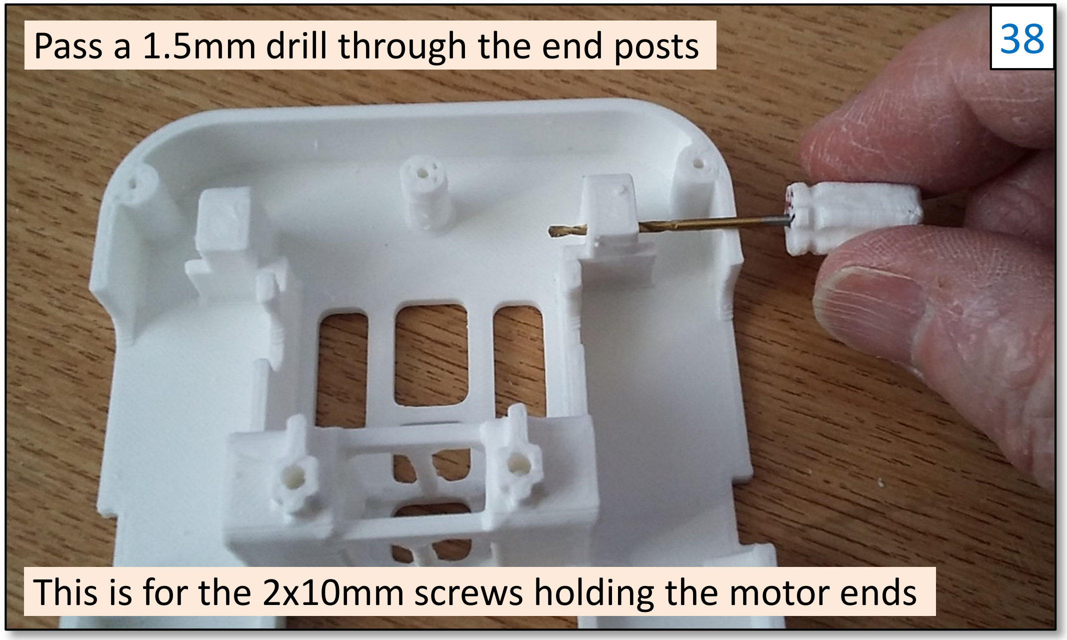


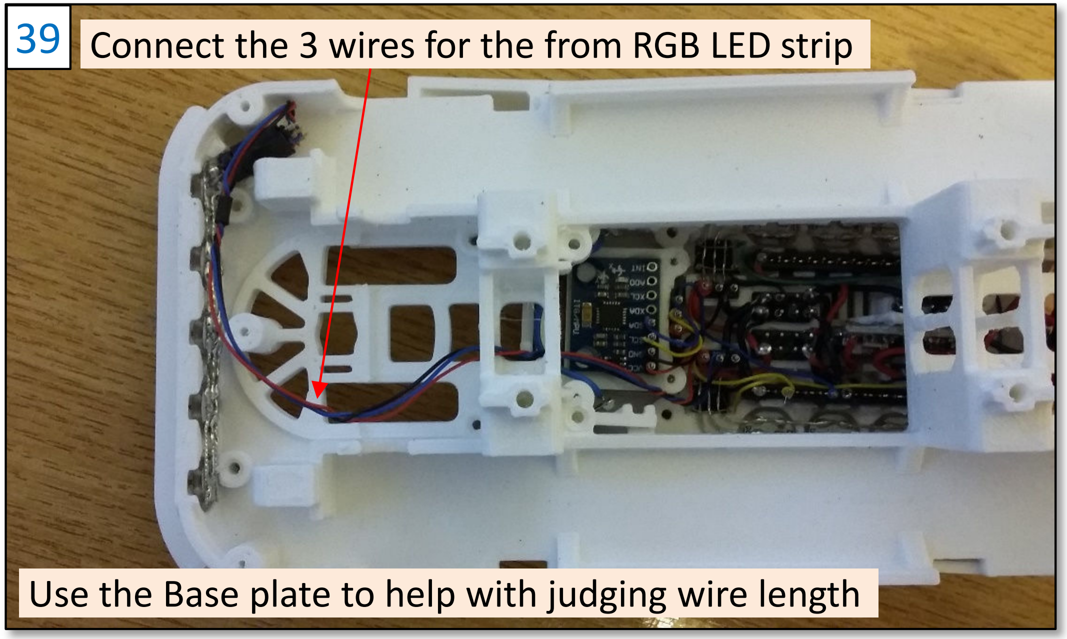
36 Do this to all 4 motors

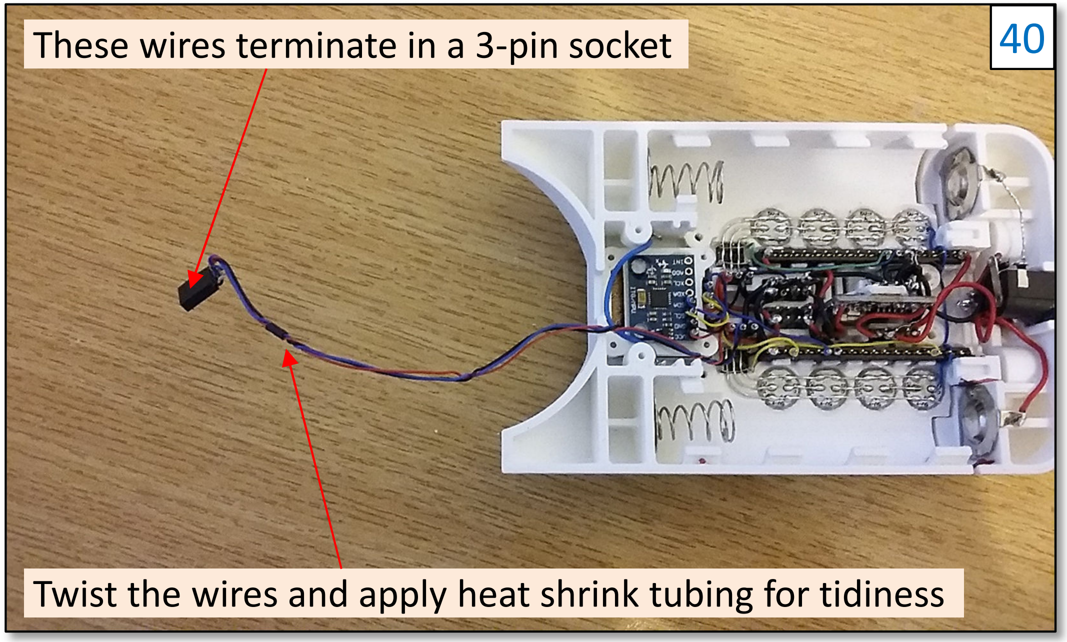
The slotted discs will be fitted at a later stage

Wiring Sequence

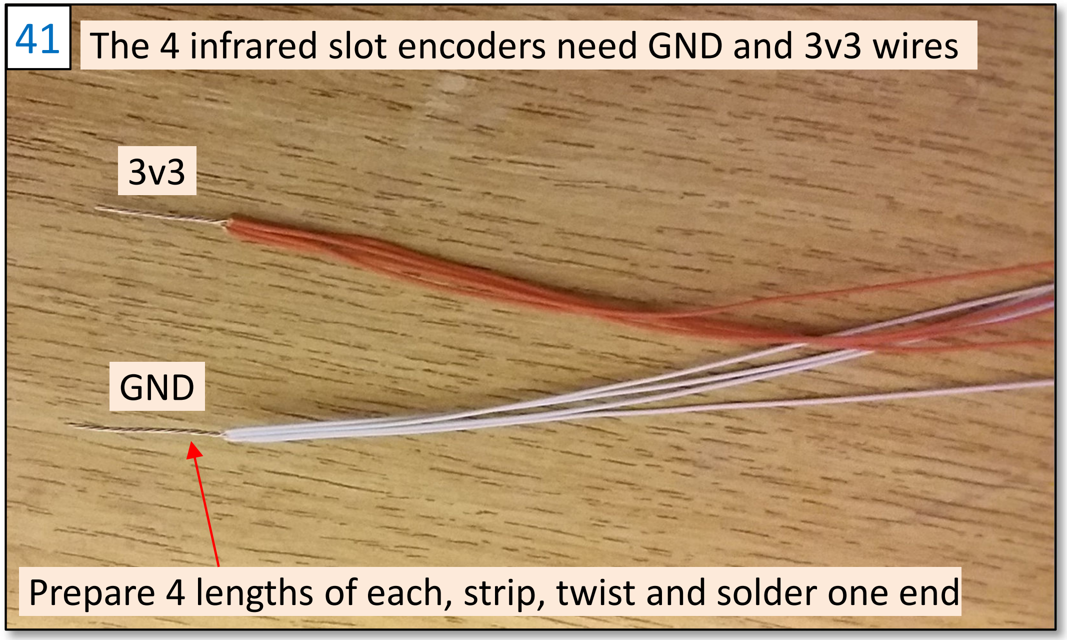


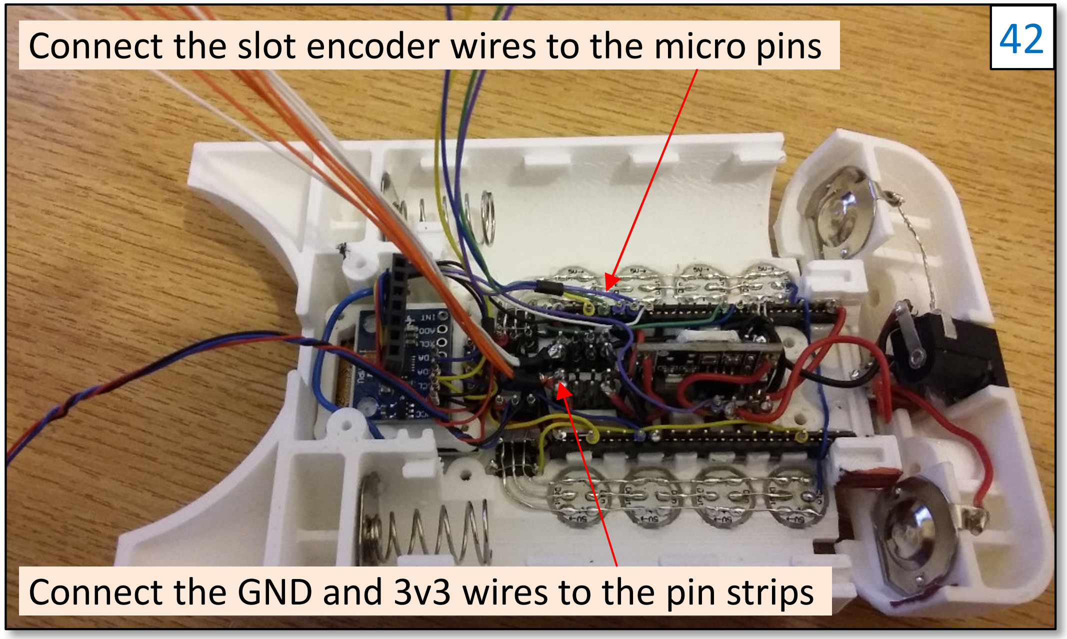


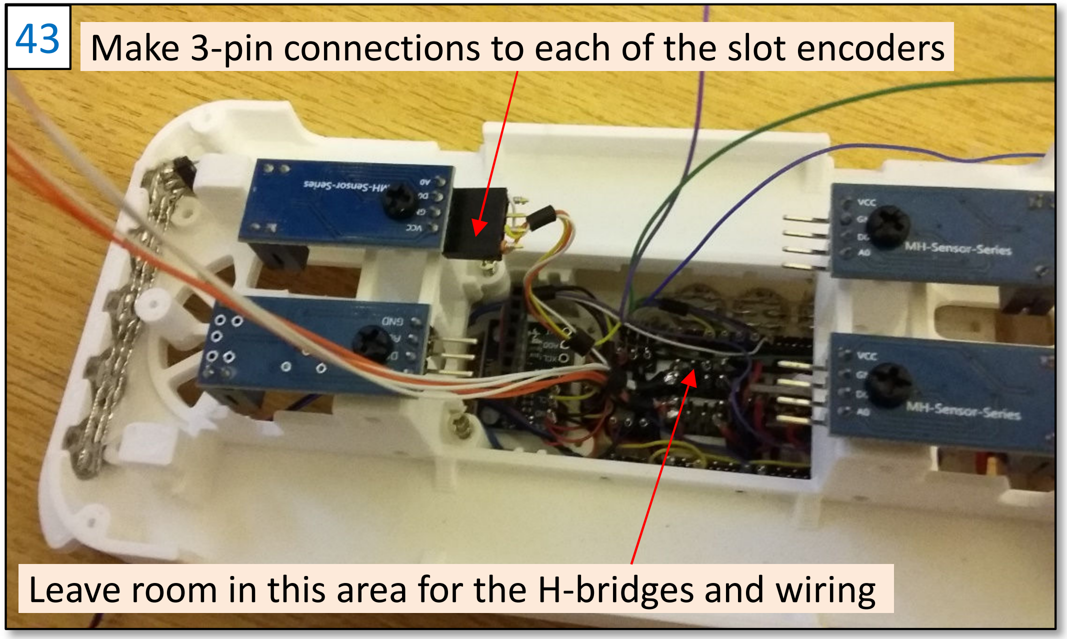


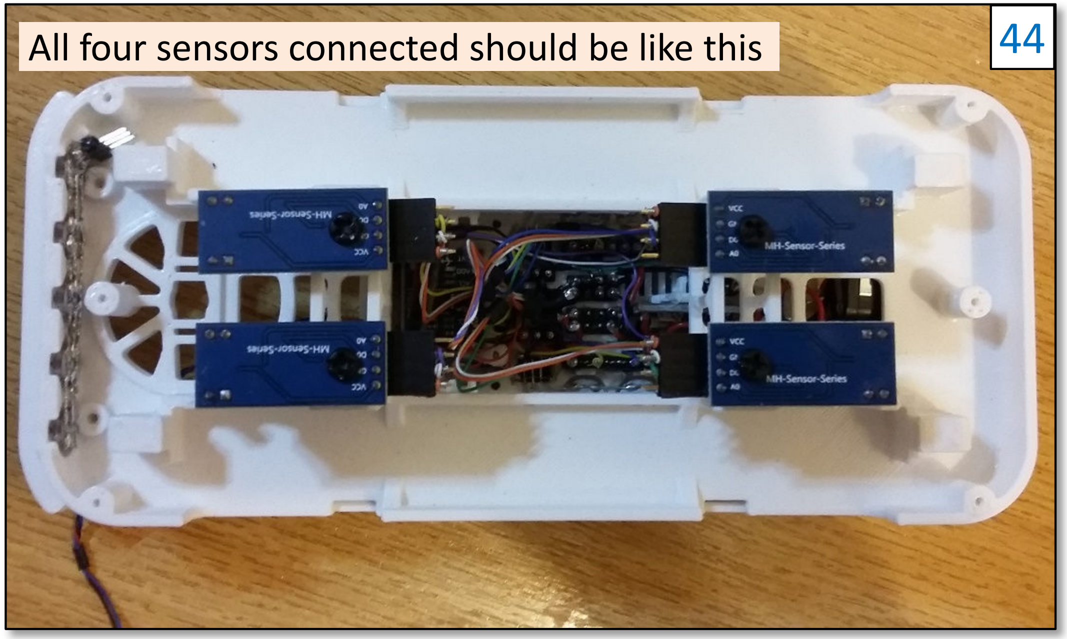


Wiring Sequence

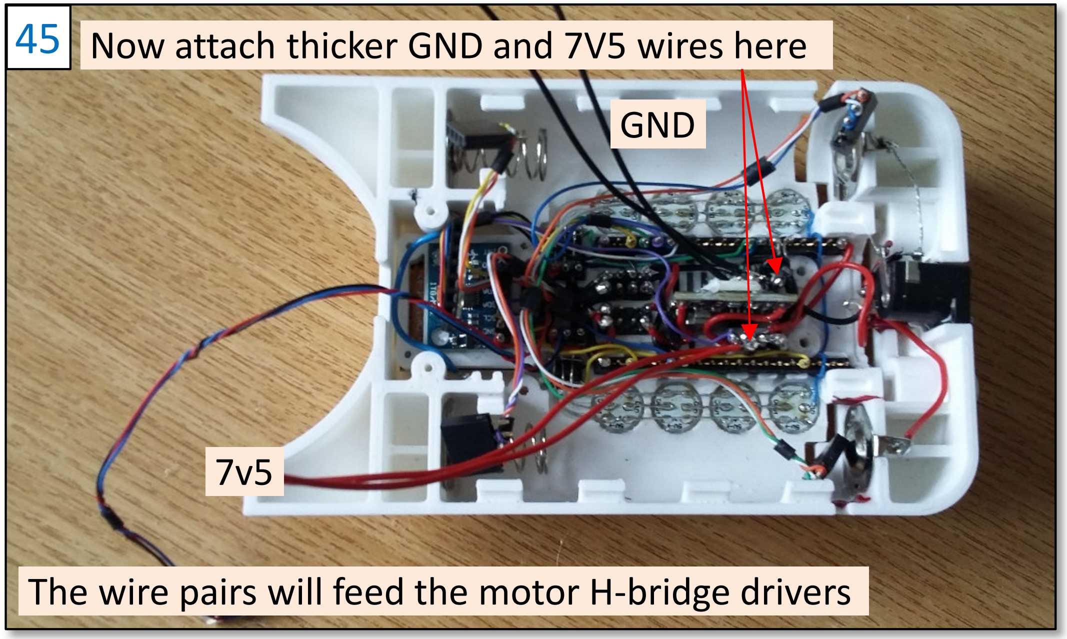


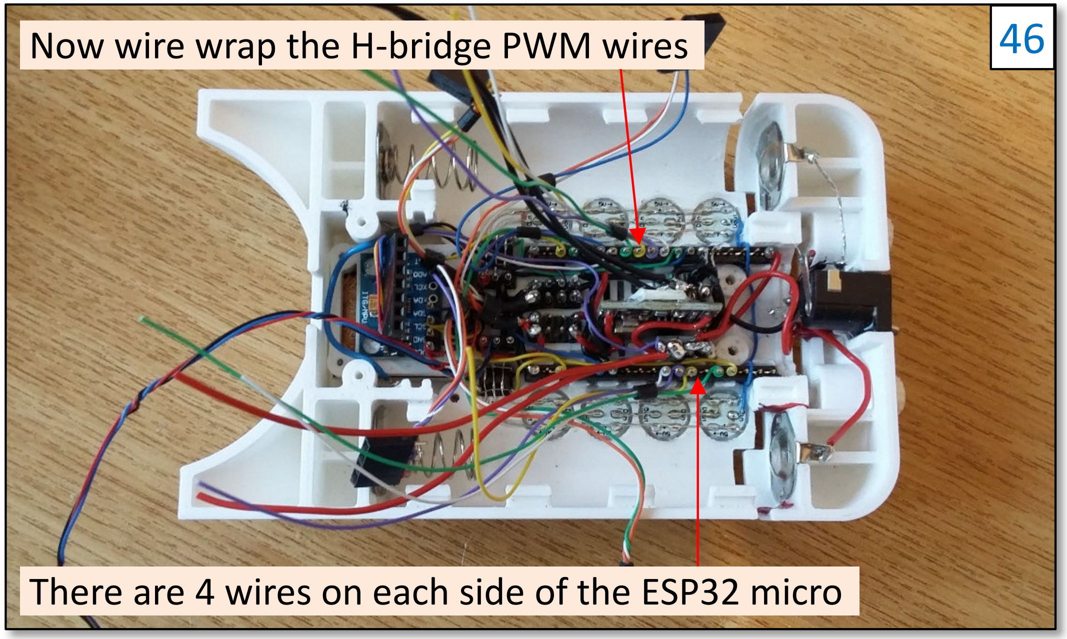


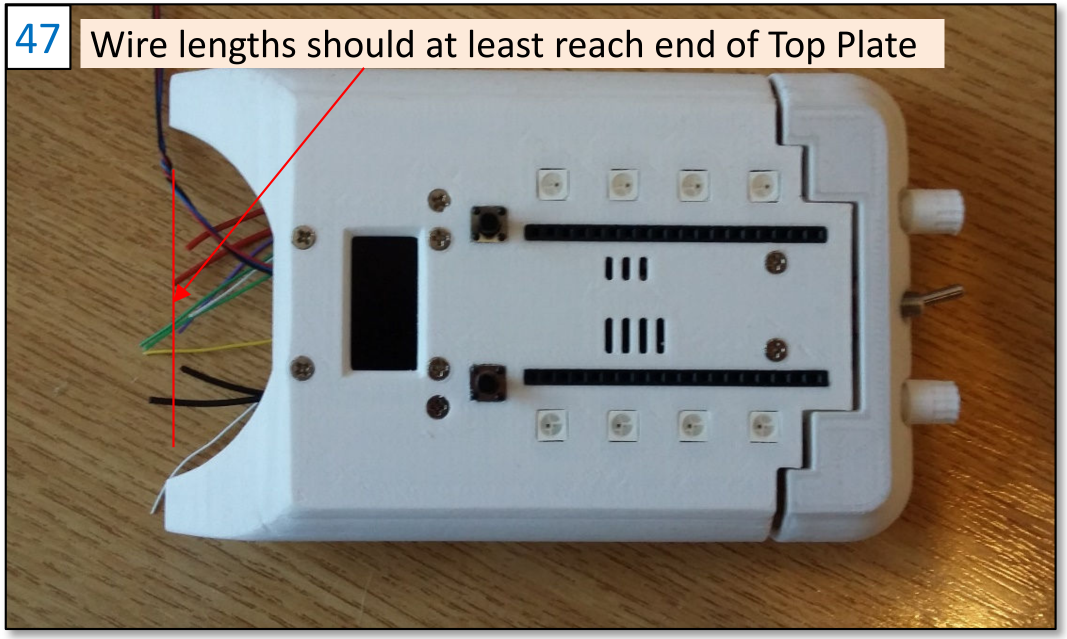


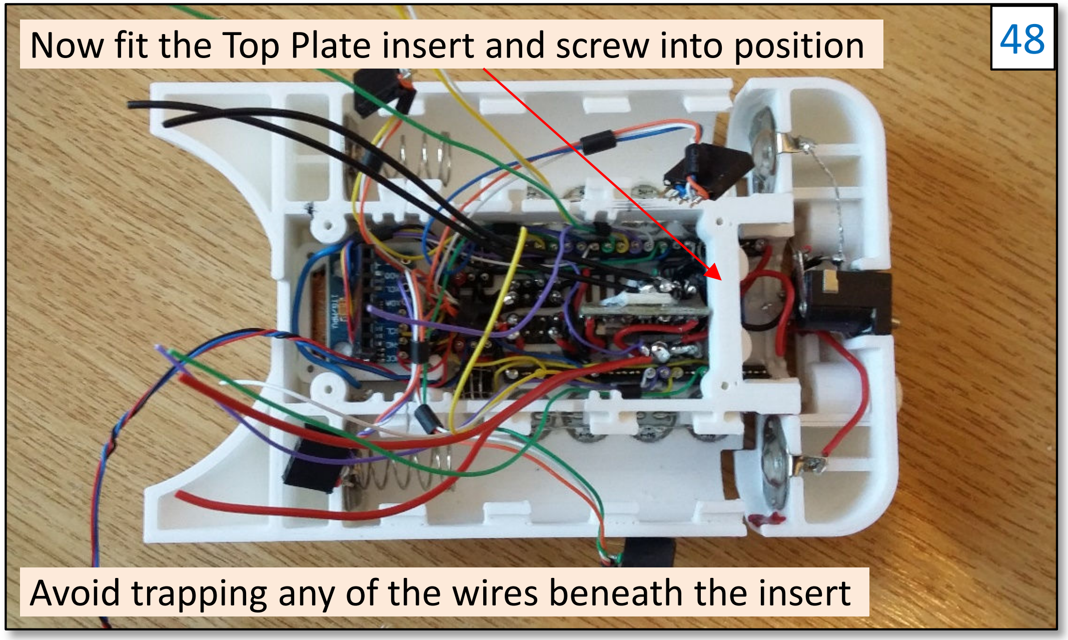


Wiring Sequence

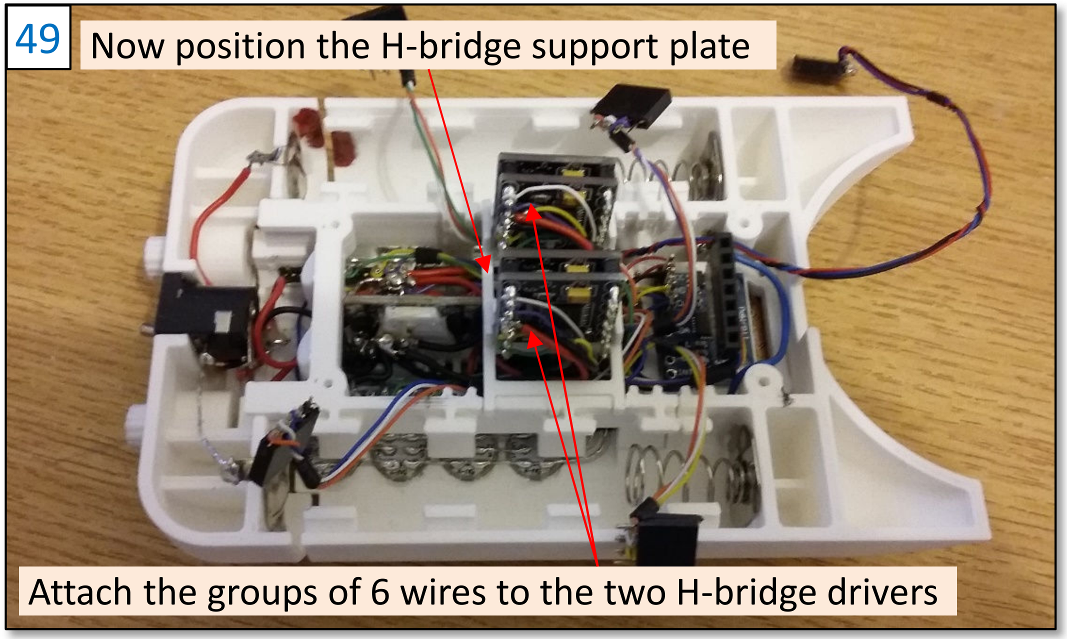


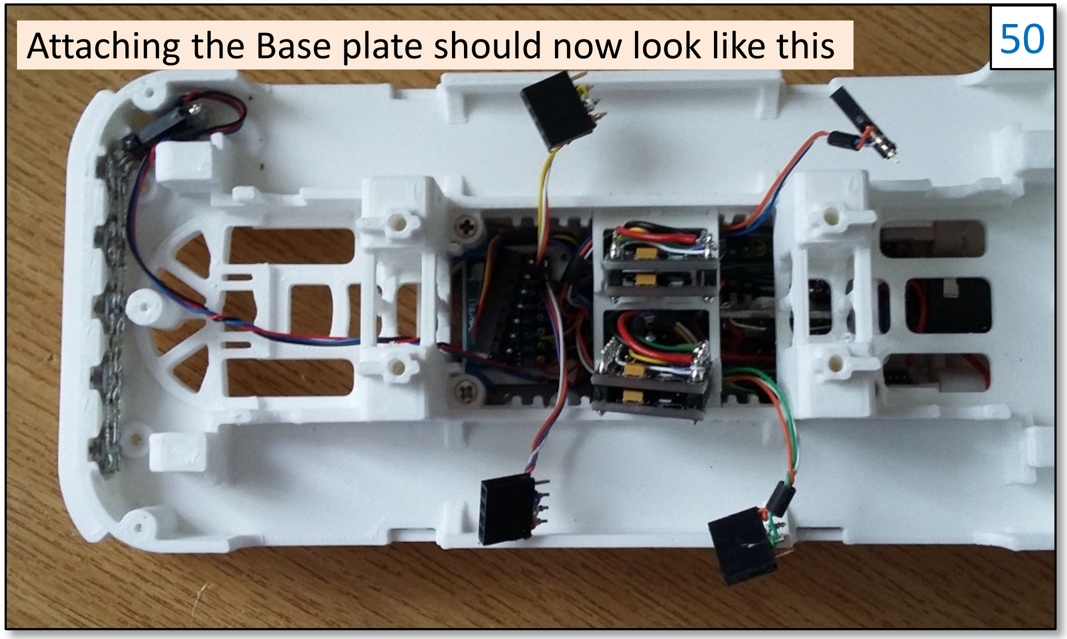


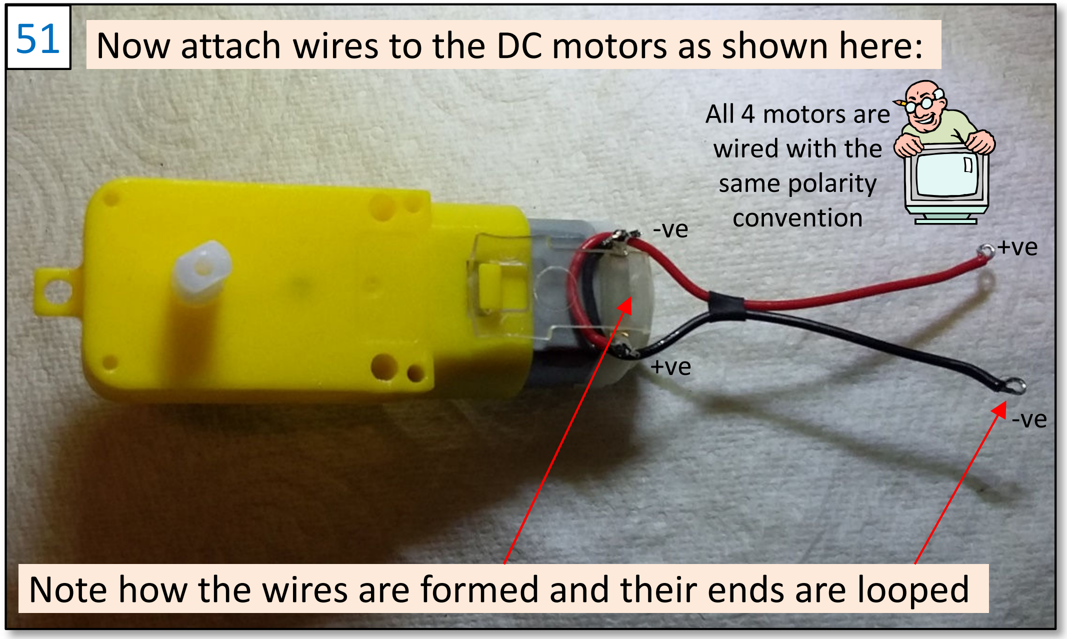


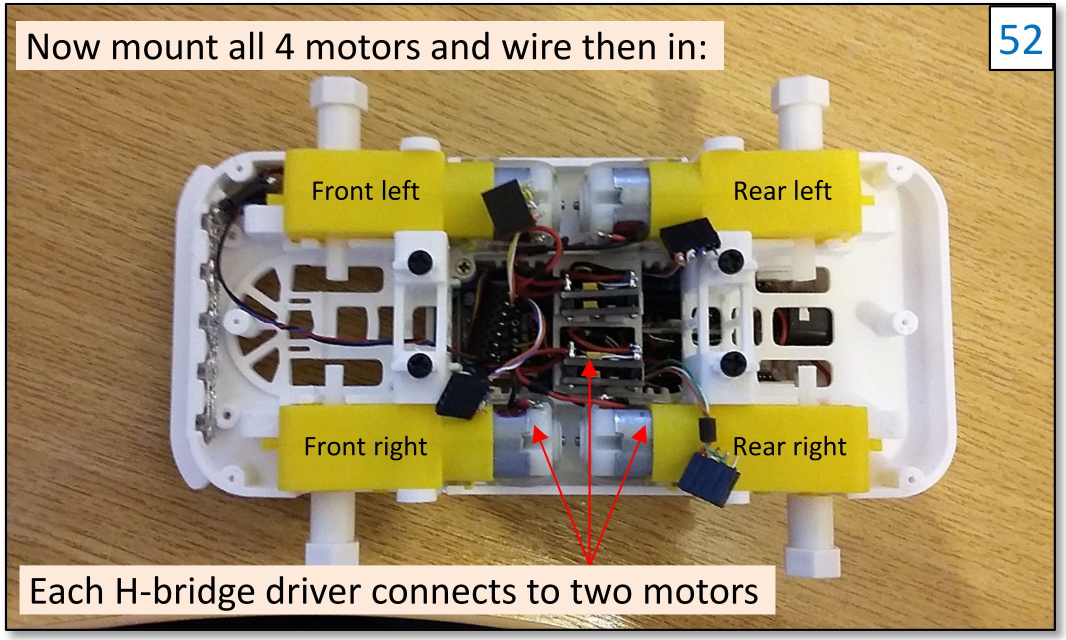


Wiring Sequence



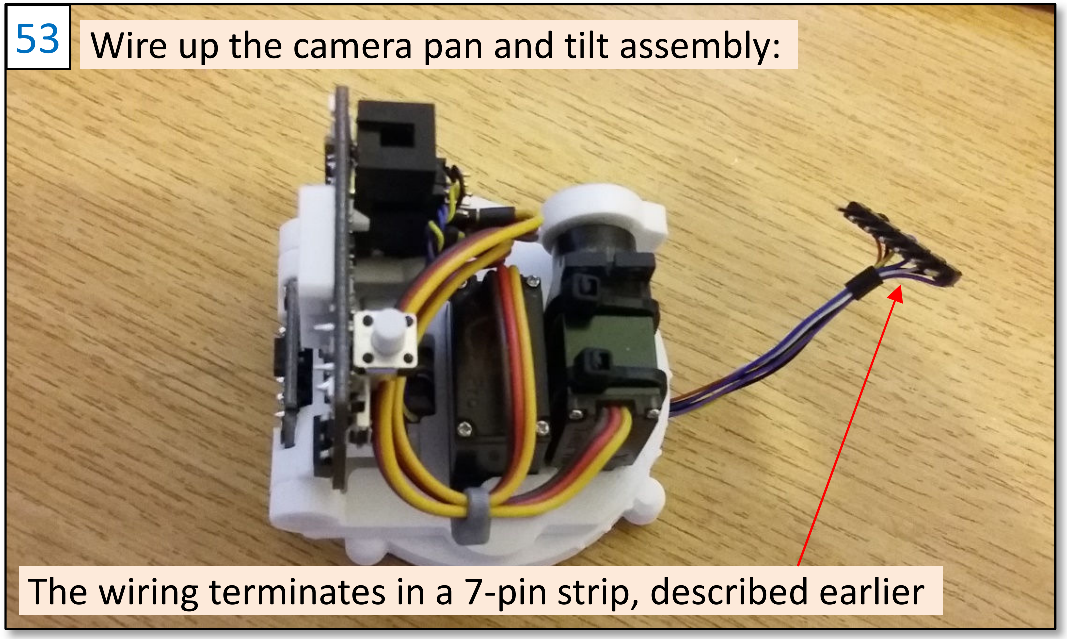






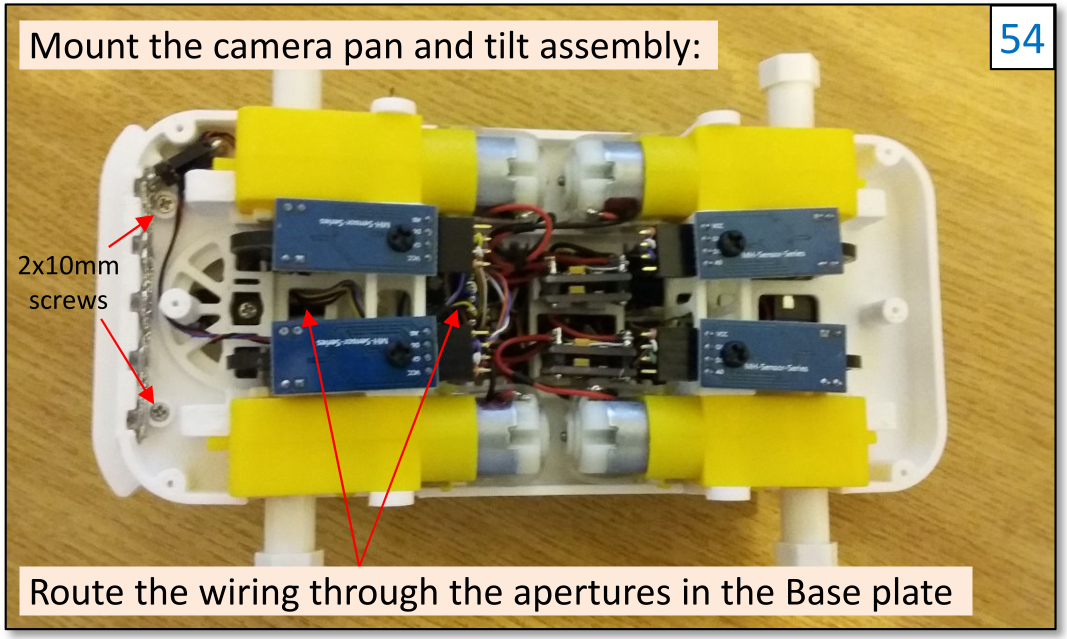
Wiring Sequence

53 Wire up the camera pan and tilt assembly:



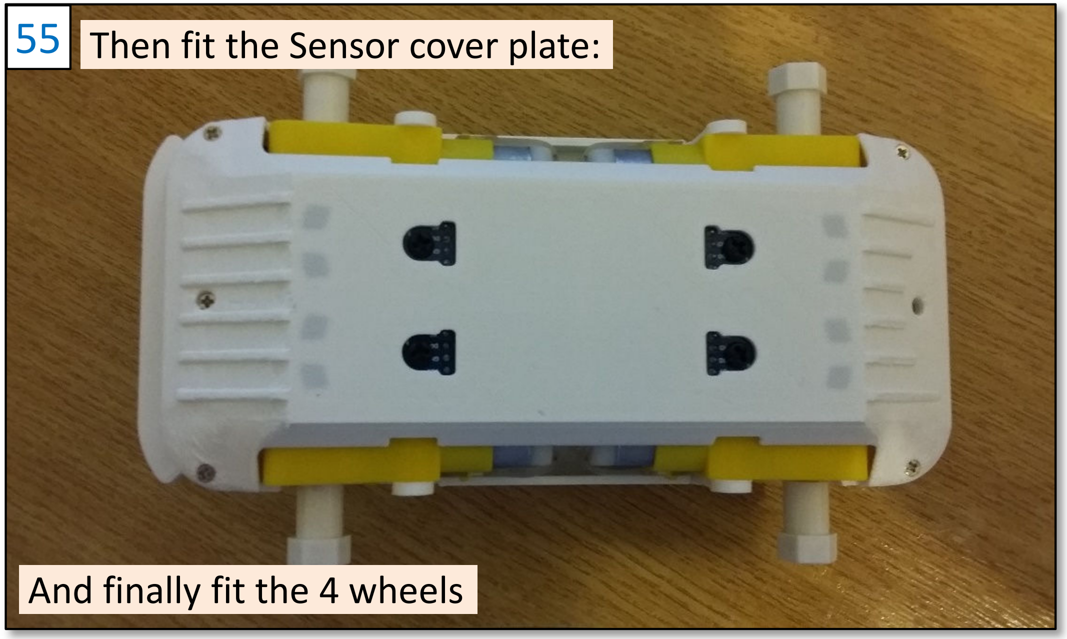
The wiring terminates in a 7-pin strip, described earlier

54 Mount the camera pan and tilt assembly:



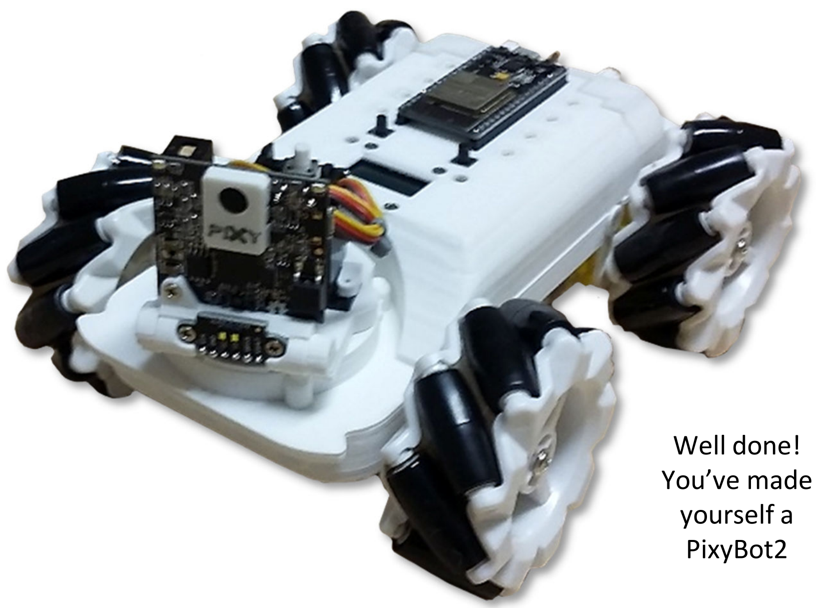
Route the wiring through the apertures in the Base plate

55 Then fit the Sensor cover plate:

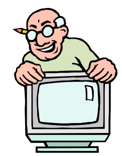


And finally fit the 4 wheels

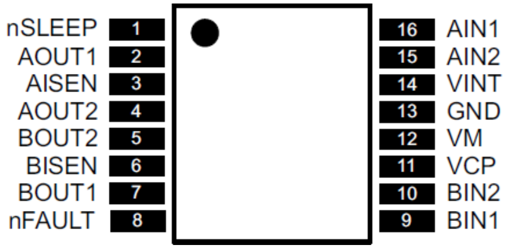
56

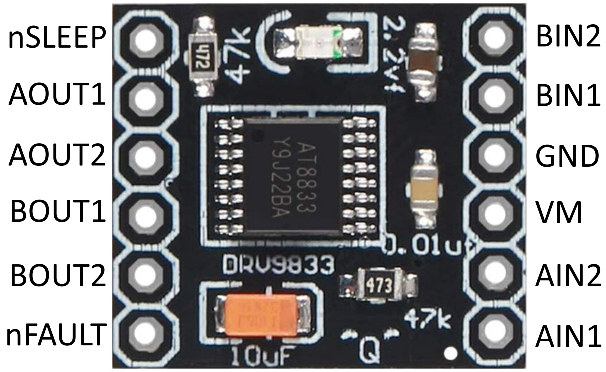


Well done!
You've made
yourself a
PixyBot2

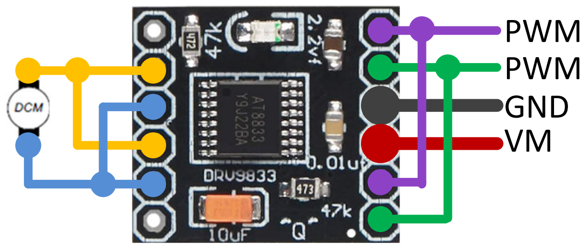


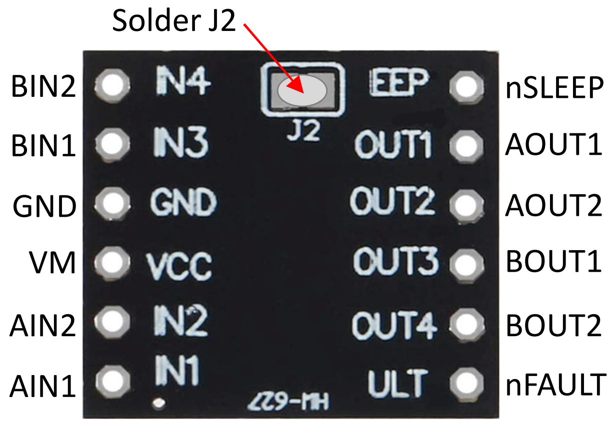
DRV8833 H-bridge Driver





Combined H-bridge.





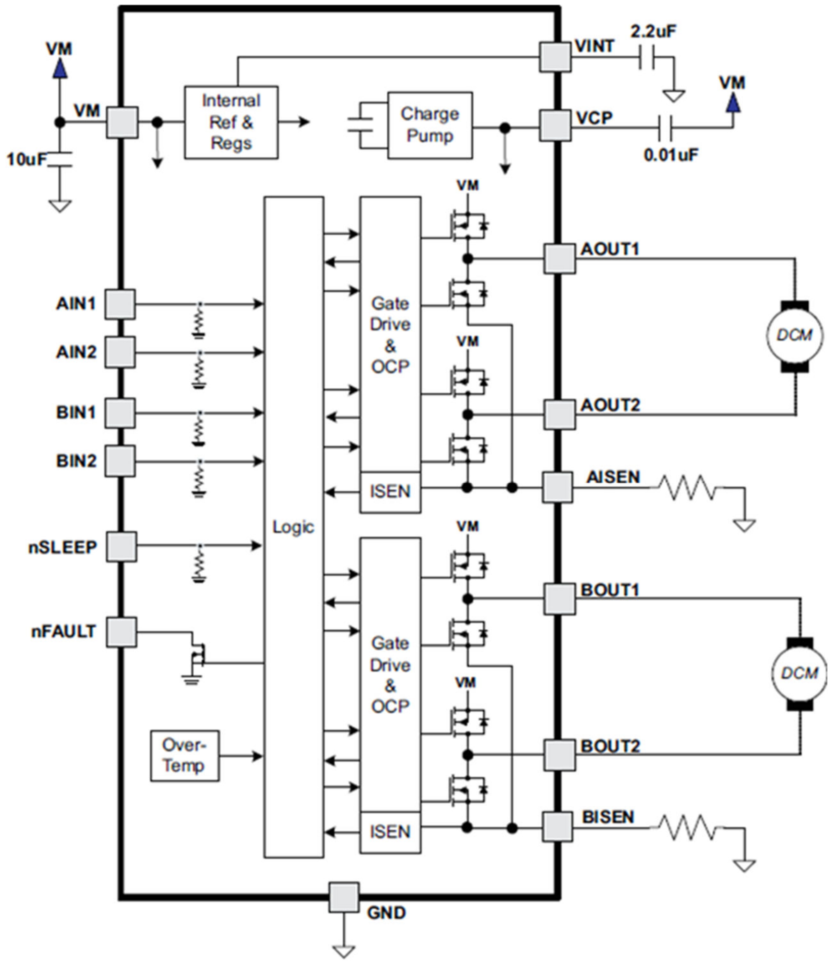
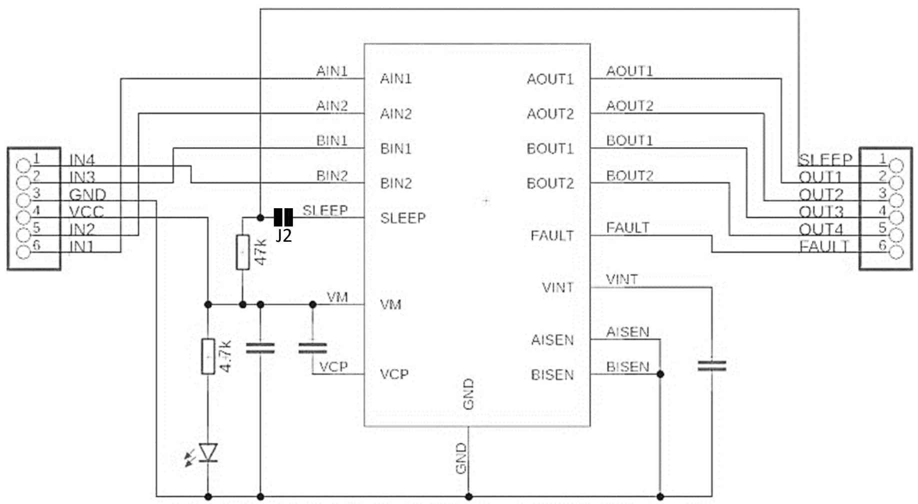


Table 1. H-Bridge Logic

xIN1	xIN2	xOUT1	xOUT2	FUNCTION
0	0	Z	Z	Coast/fast decay
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	L	L	Brake/slow decay

Table 2. PWM Control of Motor Speed

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay



Battery Voltage Health Monitoring

See 18650 discharge curve obtained from the internet. In this analysis both batteries are identical and connected in series,
Assume fully charged batteries max voltage is V_{BM} >= 8.2v max
I measured my fully recharge 18650 at 8.4v when connected and ON.
Set battery warning point at V_B = 7.00v
Set battery critical point at V_{BC} = 6.60v

ESP32 is powered from batteries connected to V_{in}.
3.3v at VADC == 4095 on 12-bit converter (4095 max).
If we use a 6k8 resistor feeding A0 and a 3k3 resistor to GND, we get a conversion factor of 10.1v == 4095 or 2.47mV/bit or 404.85
Using a Multimeter I determined the conversion factor needed to be reduced to 383.9 to display voltage correctly.

MAX: V_M = 8.2v, gives A0 = 3148 on ADC (V_M* 383.9)

WARNING: V_B = 7.0v, gives A0 = 2687on ADC (V_B* 383.9)

CRITICAL: V_{BC} = 6.6v, gives A0 = 2534 on ADC (V_{BC}* 383.9)

The code will sample the battery voltage on power-up to ensure it is sufficient, then at every 40ms interval, calculating an average (1/20) to remove noise.

Given the relatively light current drawn I have assumed a linear discharge curve ranging from 8.2v (100%) to 6.6v (0%) capacity. The rate of discharge is monitored and used to actively predict the life of the battery in use.

Note: If connected to USB port with internal battery switched OFF the ADC will read a value 5 volts (A0 = 1919) or less. So if the micro starts with such a low reading it knows that it is on USB power.