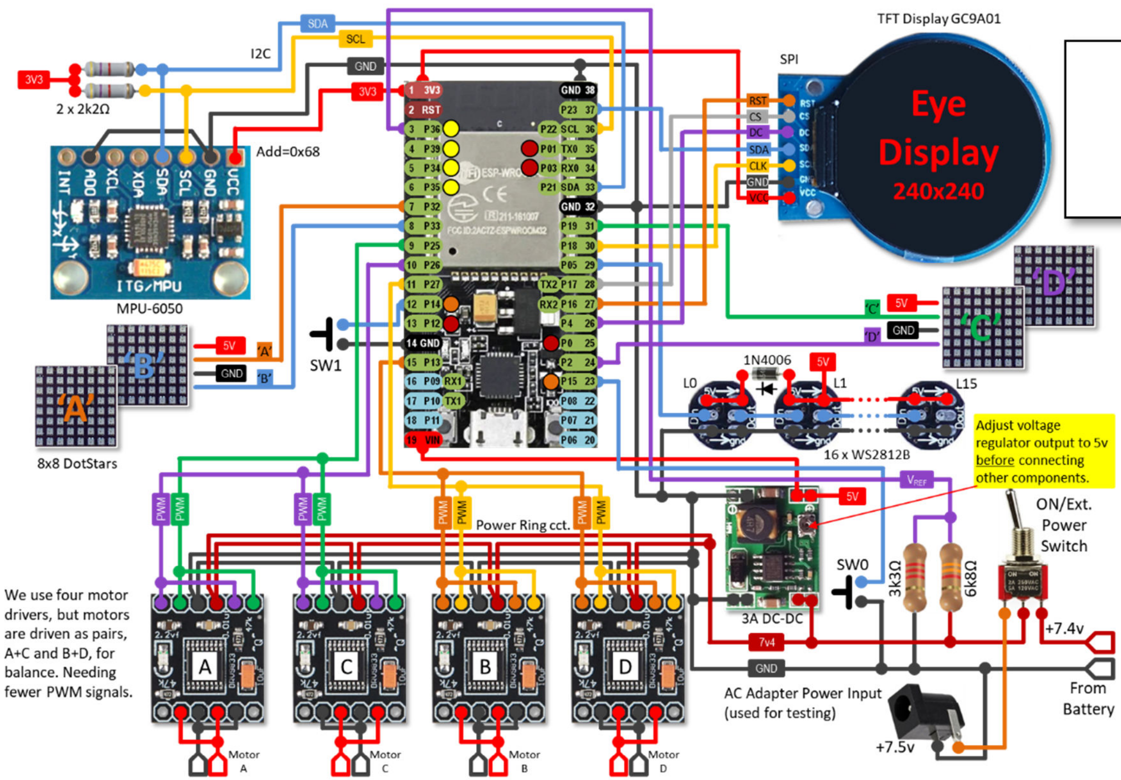
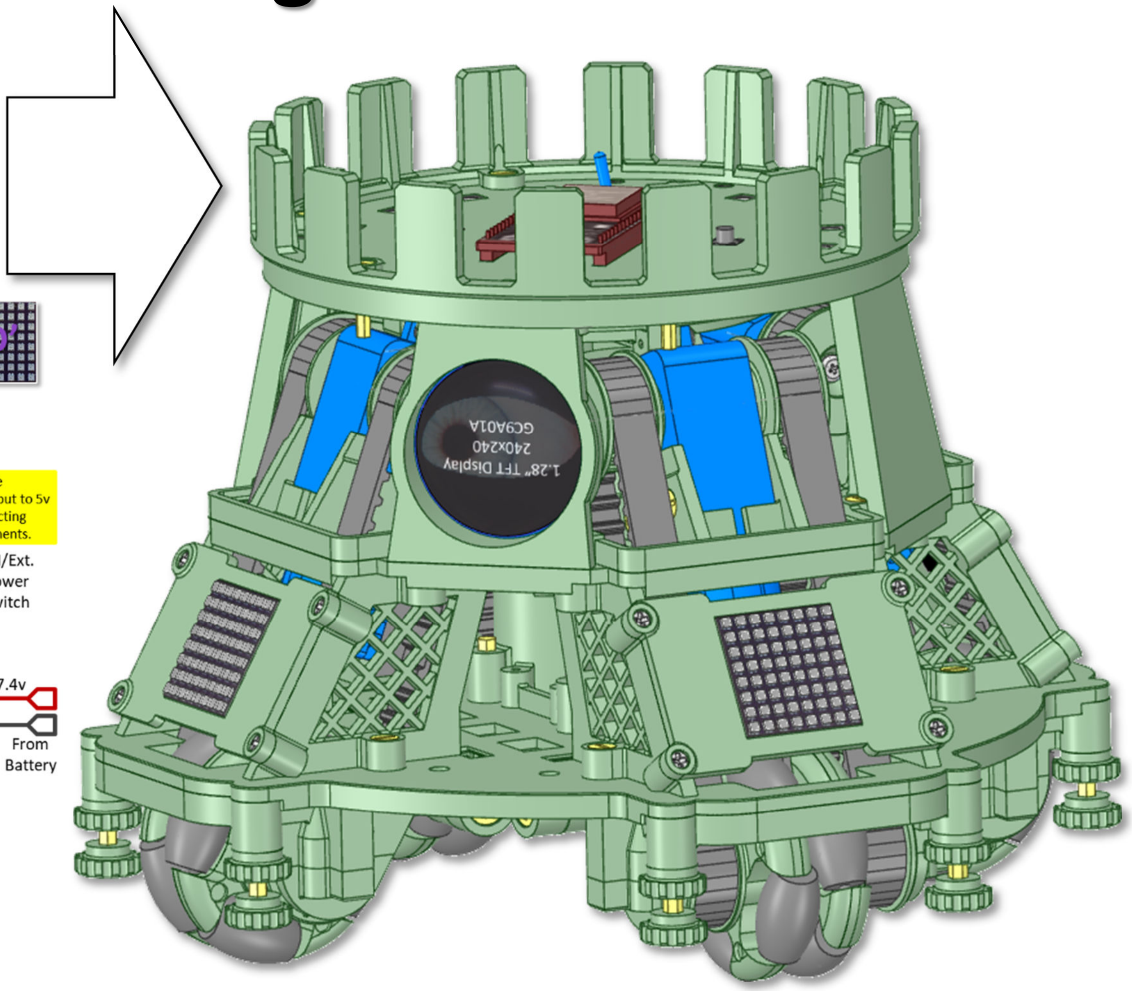
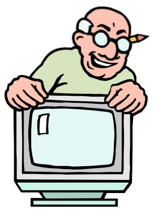
Project BallBot 4x4 (strider)

Circuits & Wiring



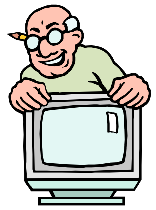




Read through this documentation completely <u>before</u> attempting this complex project.

Issue: 1.0 Released: 12/12/2025 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.

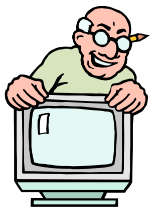
Built-in Monitoring: Most of my project designs include code, and circuitry, to monitor battery voltage, whilst in use. This code then seeks to alert the operator, when the battery has reached a critical low voltage, before shutting down power consuming circuitry; including the micro. Time should therefore 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:

- <u>Fine</u> nosed pliers
- Side cutters
- 1.5 mm Drill
- 2.0 mm Drill
- 4.0 mm Drill
- Needle files
- Screwdrivers
- Craft knife

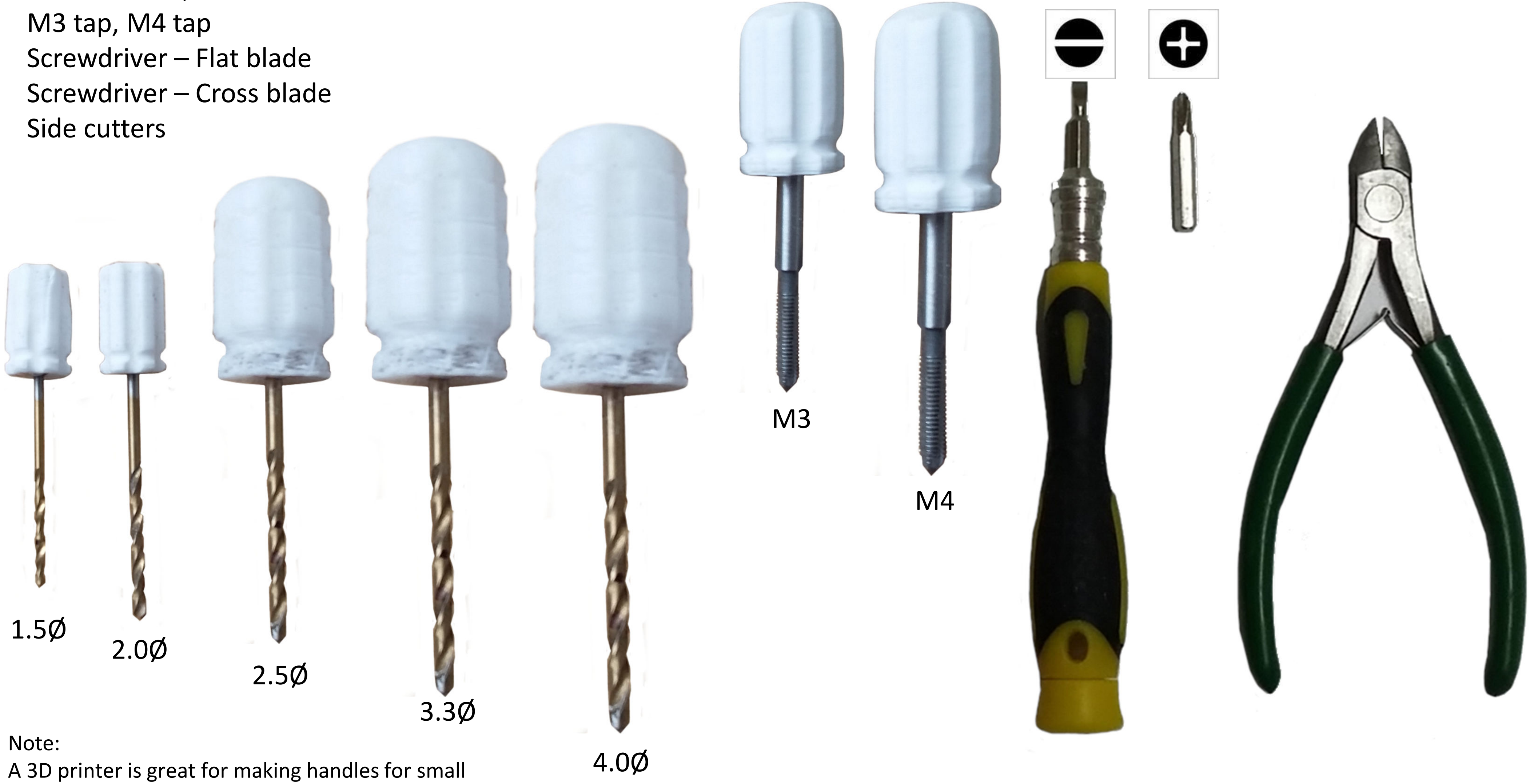


Note: Not all items needed are shown here.

Some printed components act as aids and gauges. Use them.

Construction - Tools:

Drills: 1.5mm, 2.0mm, 2.5mm,
3.3mm, 4.0mm
M3 tap, M4 tap
Screwdriver – Flat blade
Screwdriver – Cross blade
Side cutters

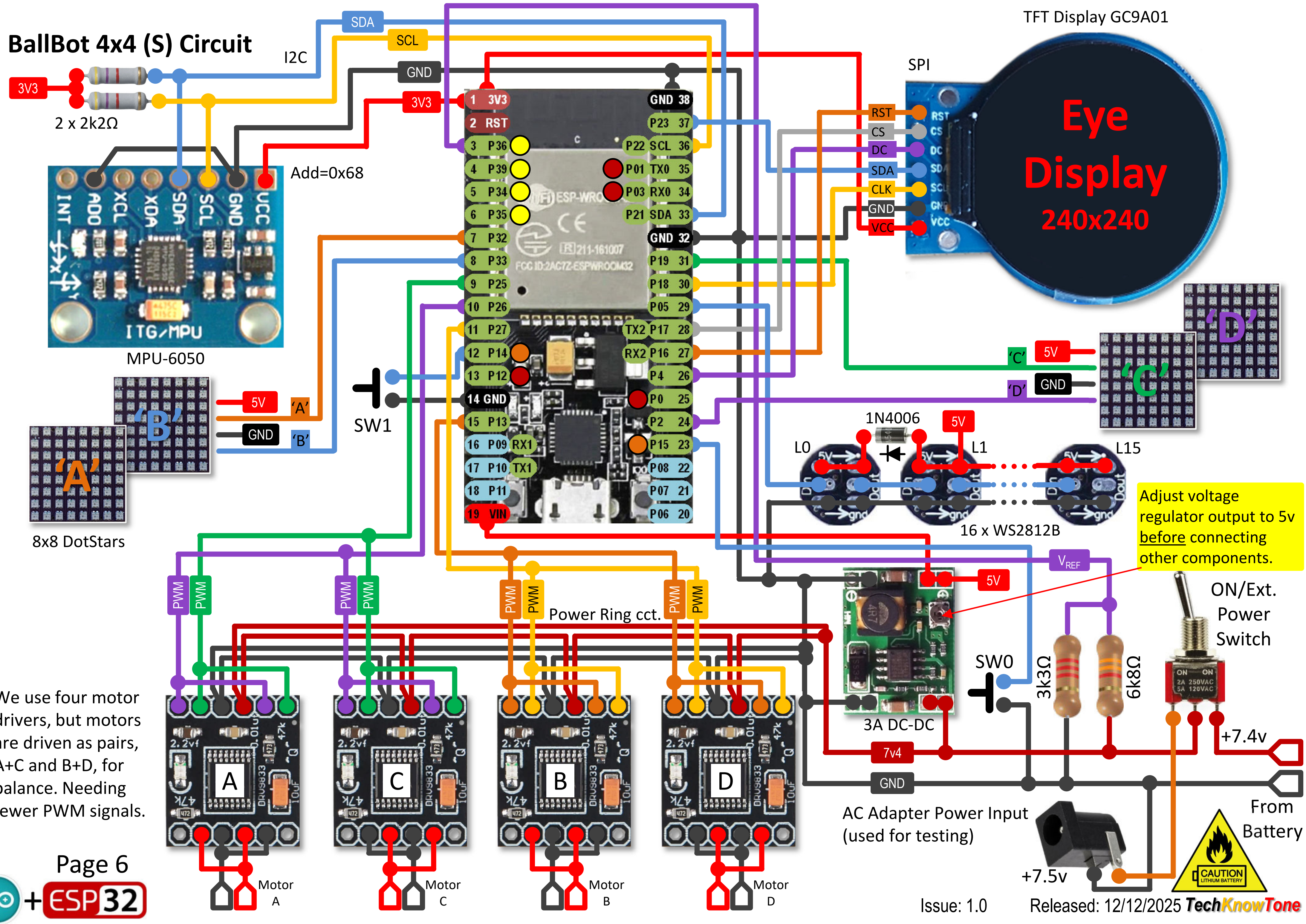


Note:
A 3D printer is great for making handles for small drills and taps.

Tools & Materials:

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

BallBot 4x4 (S) Circuit



We use four motor drivers, but motors are driven as pairs, A+C and B+D, for balance. Needing fewer PWM signals.

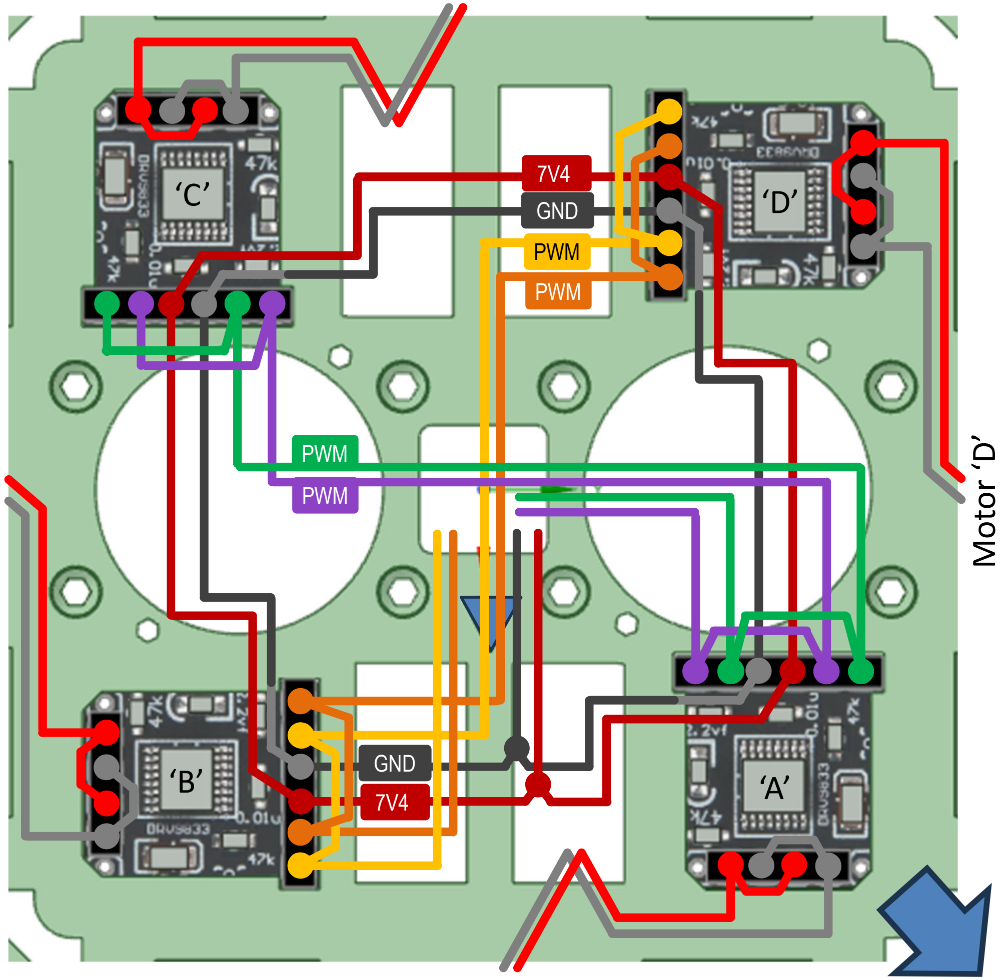
Issue: 1.0
Released: 12/12/2025 TechKnowTone

Motor Plate Driver Wiring

This diagram shows the end-to-end connections for each wire, but the physical routing is dependant on things like the battery top cover. See later photos showing physical routing. Note the construction of the 'power ring' and that it is fed from two wire pairs +7v4 and GND. All feed wires from the micro plate come to the motor driver plate at the centre point, through the aperture in the battery top cover.

Motor 'C'

Motor 'B'



Motor 'D'

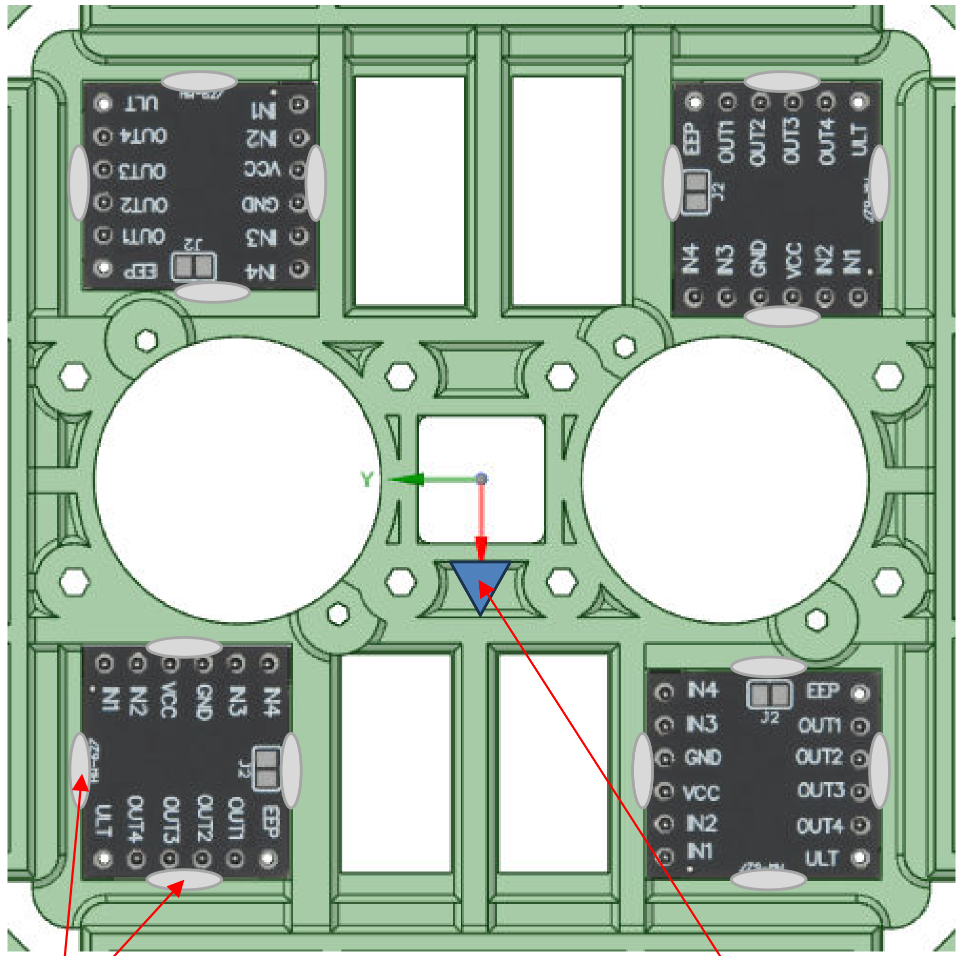
Motor 'A'

Front Eye

Viewed from above

Motor 'C'

Motor 'D'



Motor 'B'

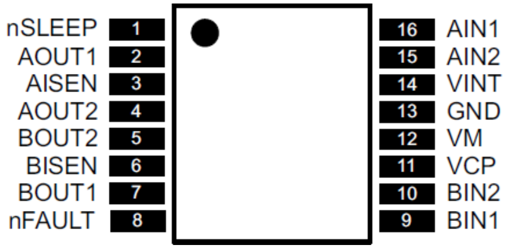
Motor 'A'

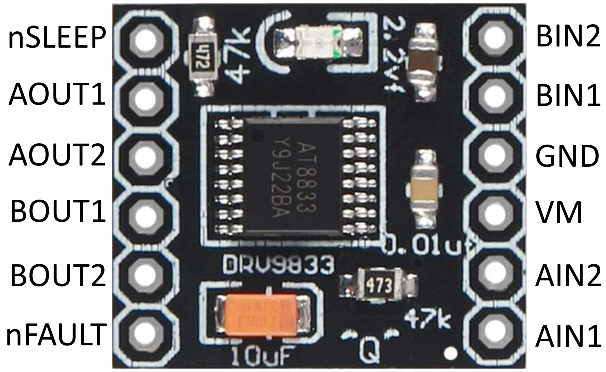
Drivers are glued into position using dabs of epoxy glue.

Viewed from below

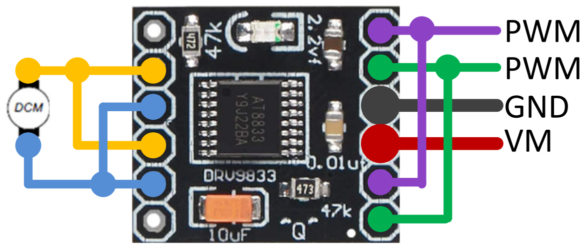
Apply a reference marker.

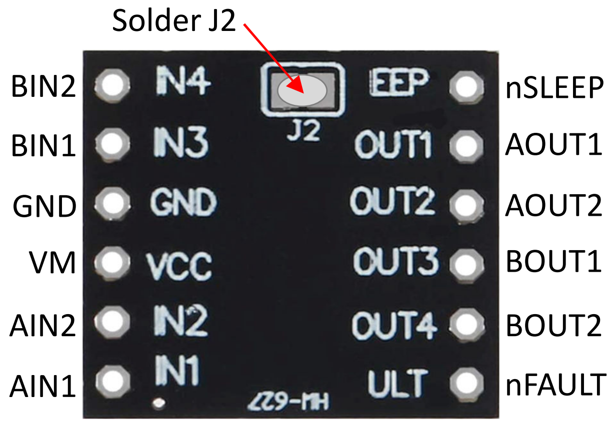
DRV8833 H-bridge Driver

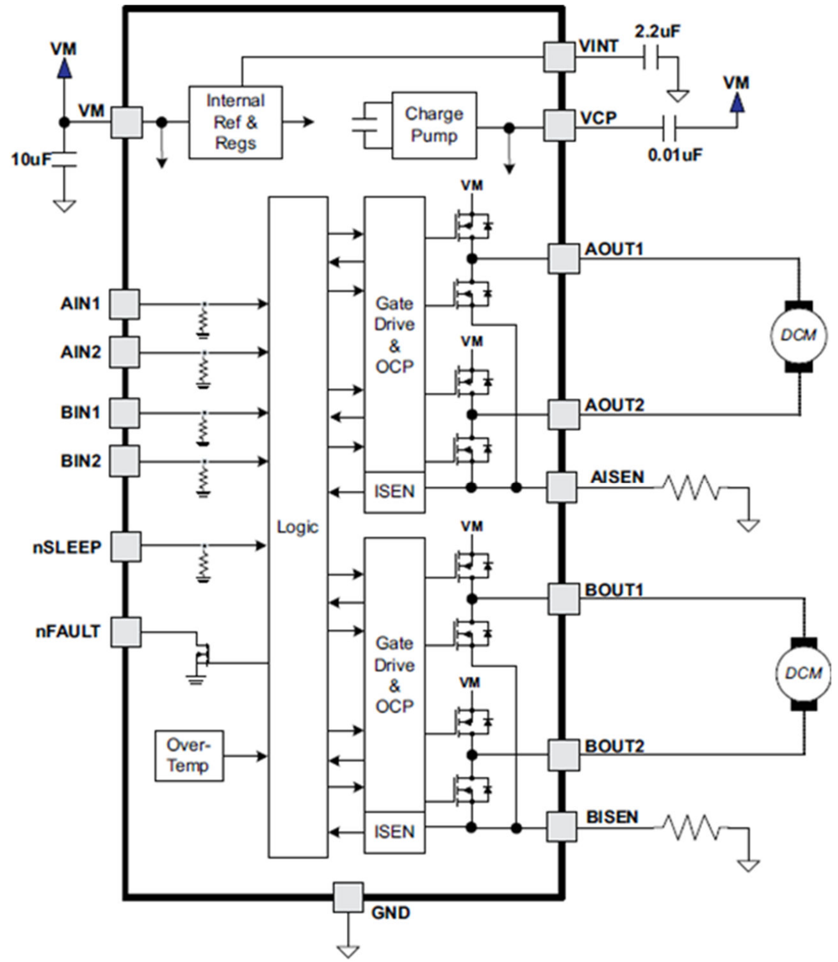




Combined H-bridge.







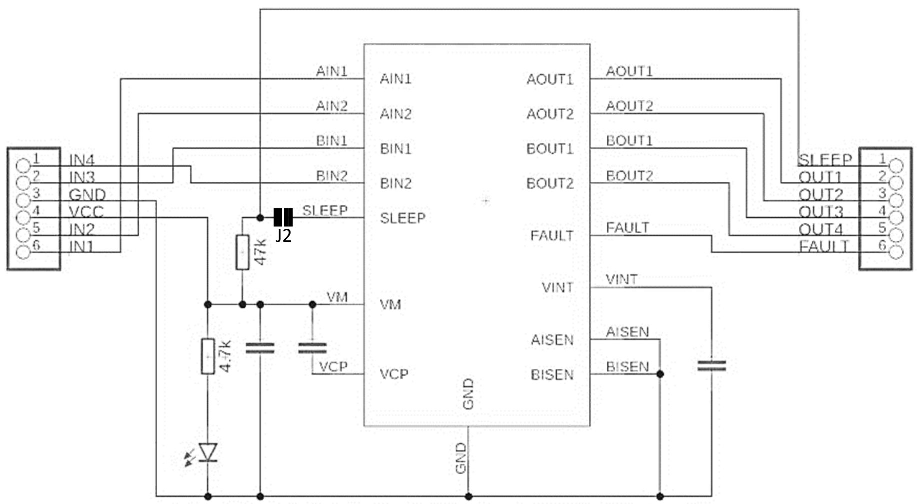


Table 1. H-Bridge Logic

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

Dual H-bridge Wiring

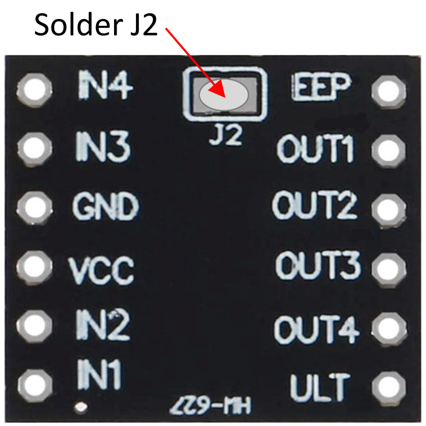
In this design I used one DRV8833 H-bridge dual channel controller for each motor, and combined each channels 2A max current capability, and halved their MOS-FETs on-channel resistance. Giving a more efficient drive signal for each motor.

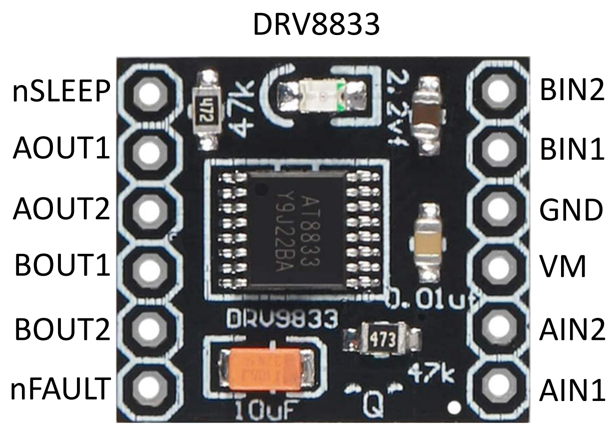
This is achieved by wiring the controllers as shown here. Both the input pins and output pins are connected together as shown. Make sure that you wire the controllers correctly, or you are likely to damage the controller, and draw excessive current from the batteries.

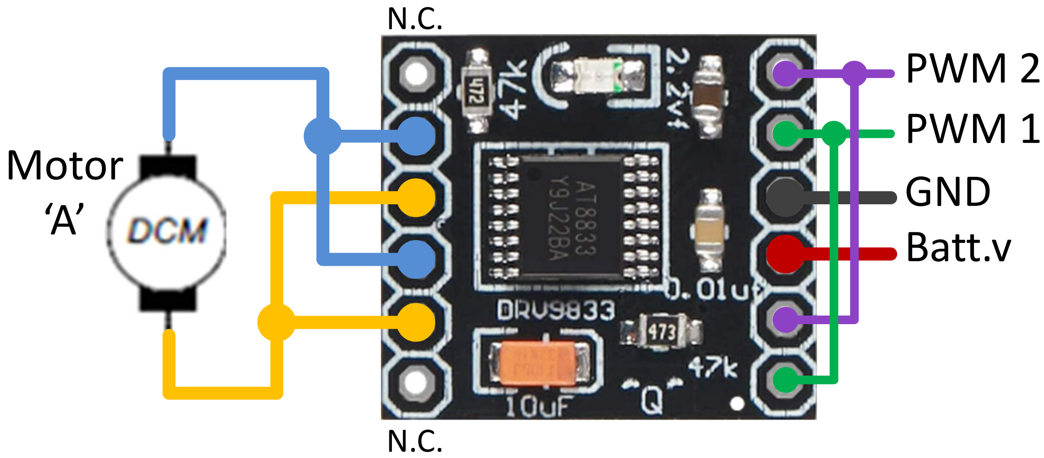
The nSLEEP and nFAULT pins are not connected; however, the nSLEEP pin is pulled HIGH by an on-board 47kΩ resistor when you bridge J2 with a blob of solder, as shown. Do this **<u>first</u>**, before soldering in the pin strips to the boards.

You will find more information on the DRV8833 driver on a later slide. This should enable you to understand how the code producing the PWM signals in the ESP32, controls the motion of the DC motor. For each direction of travel, one PWM input pin is held HIGH, whilst PWM signals are applied to the other.

As the supply voltage affects motor speed and torque, the code monitors the battery voltage and adjusts the applied PWM to give a maximum average voltage of 7.6v.





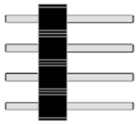


CAUTION – This design does not include any form of current limiting protection (like a fuse). Incorrect wiring can result in large currents, which can cause damage and even the risk of a **fire**. Check and double check your wiring before inserting the batteries.

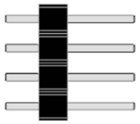
Microplate Total Wiring

This diagram shows all of the micro plates internal wiring, viewed from below. The 5v buck regulator sits beneath the microcontroller, and pin strips are used as common connection points.

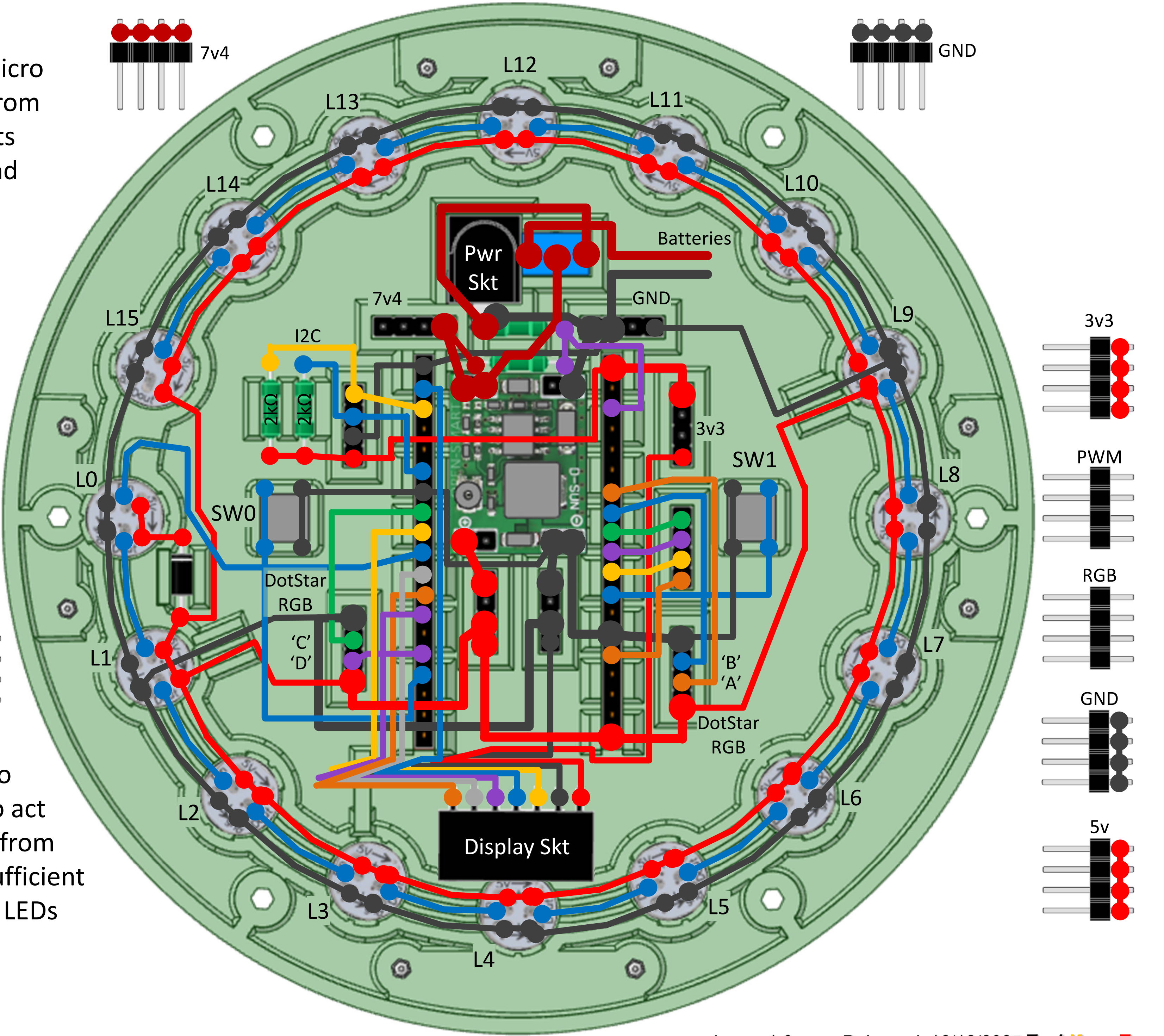
Note that some pin strips are prewired as common connections, where as others are simply used as intermediate termination points.



Wires to the 7-pin socket strip, that connects to the display are looped to providing flexibility in making off the connection.



The diode, feeding 4v4 power to the 1st RGB LED L0, enables it to act as a level shifter. As the output from the ESP32's logic HI may be insufficient to drive it reliably. All the other LEDs work at 5v.



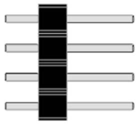
Issue: 1.0 Released: 12/12/2025 TechKnowTone

Microplate Wiring – Step 1

The first step in the micro plate build process is to glue in all of the components. Note that five of the 4-pin strips have their pins wired together before gluing them in place, to act as common connection points. E.g. 5v & GND

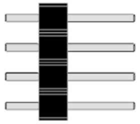
The other pin strips act as connection points for wires coming from the body of the robot.

I2C

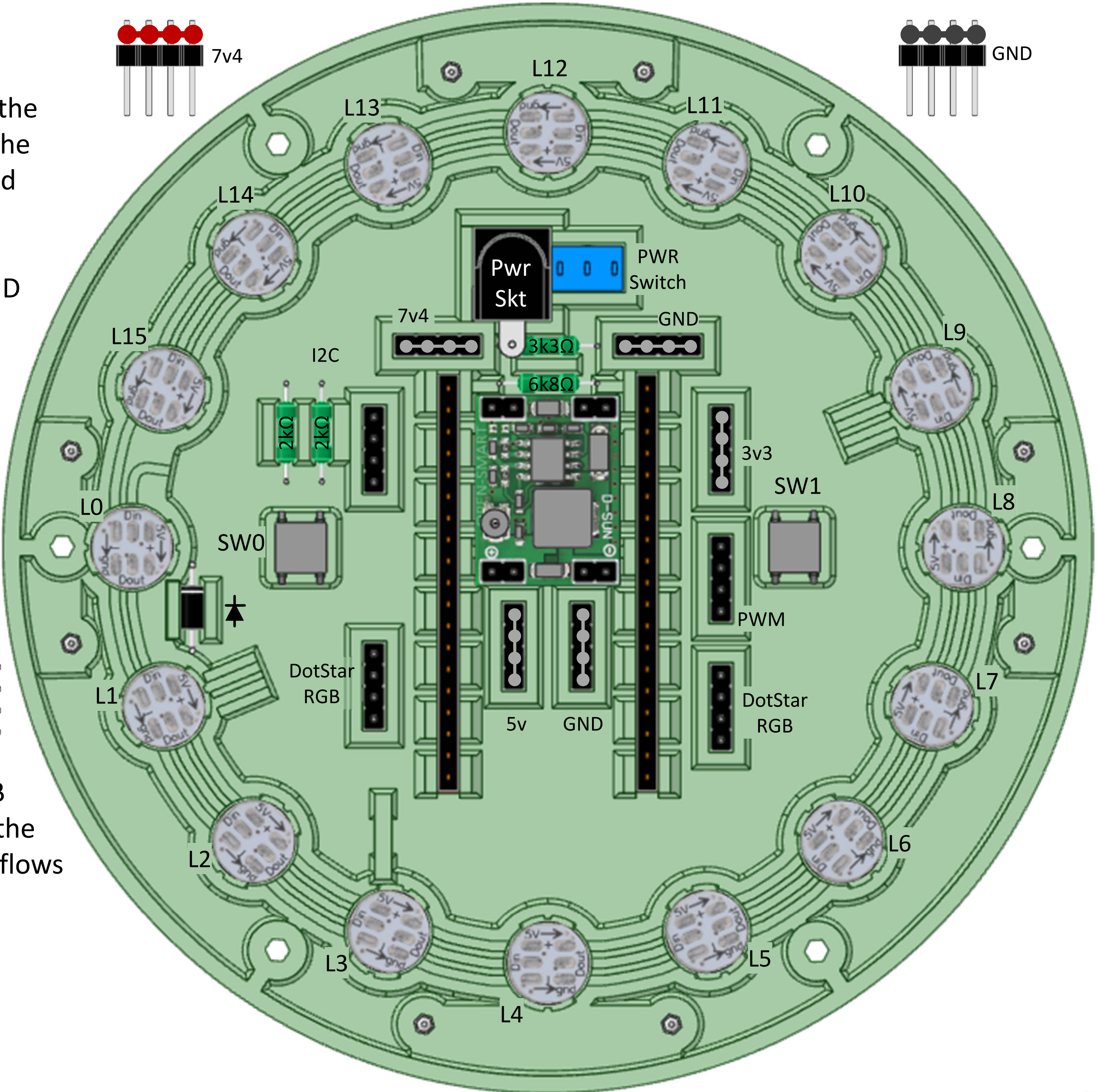


Do not insert/connect active components like the ESP32 micro until you have set the output of the voltage regulator to 5v.

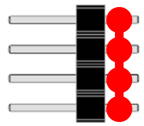
RGB



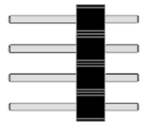
Note the orientation of the RGB LED chips, and the direction of the data flow. It is critical that data flows in an anti-clockwise direction.



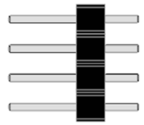
3v3



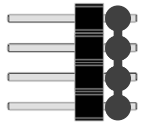
PWM



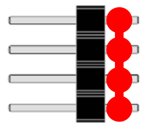
RGB



GND



5v



Microplate Wiring – Step 2

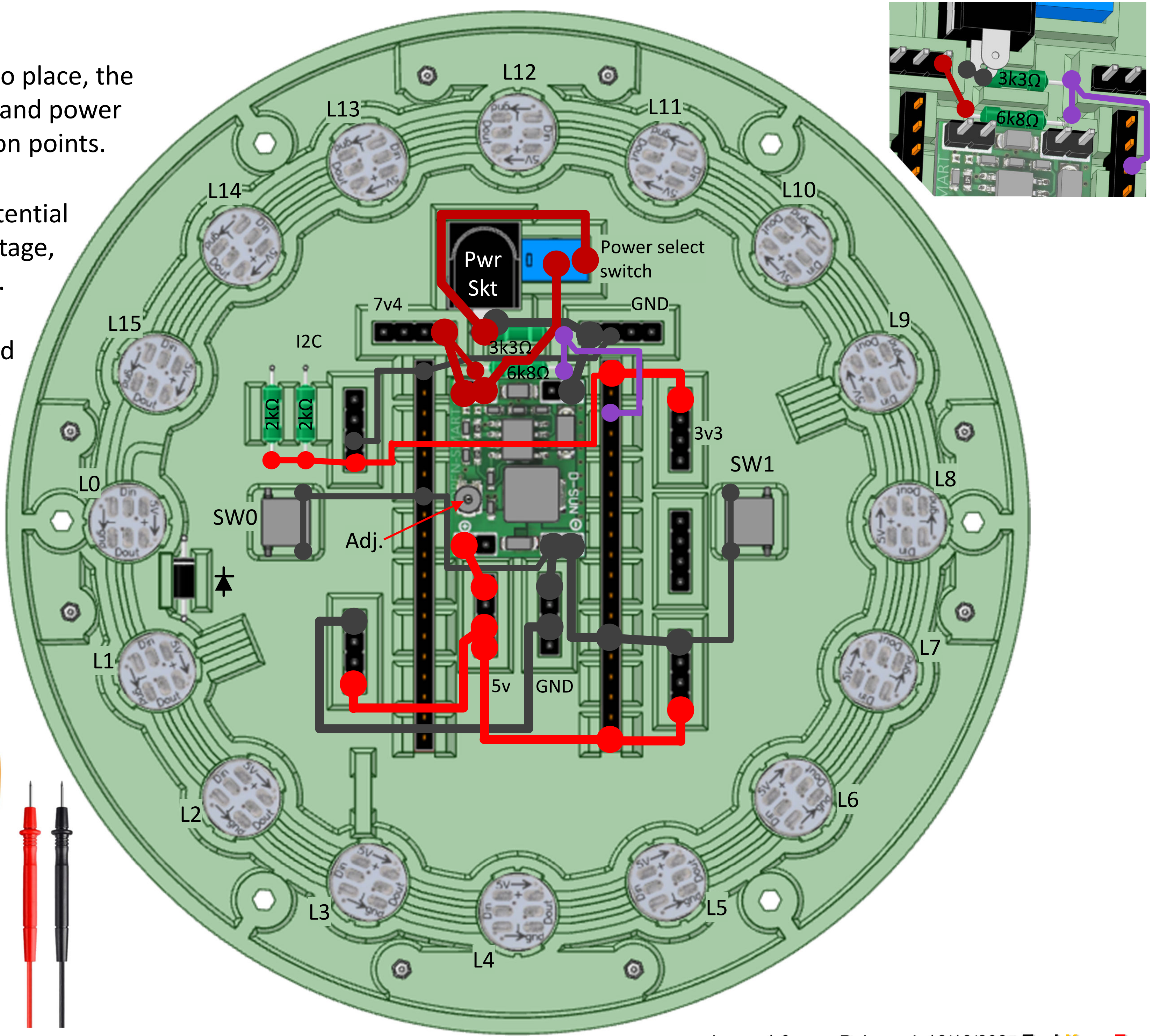
With the components glued into place, the next step is to wire in the GND and power wires to the common connection points.

The two resistors, used as a potential divider to derive the supply voltage, are wired in as shown top right.

The power select switch is wired such that when it is thrown towards the power socket, that becomes the source of power.

Once you have made these connections, you can attach a power source and adjust the voltage regulator output to 5v.

Do this before plugging in the ESP32 micro.



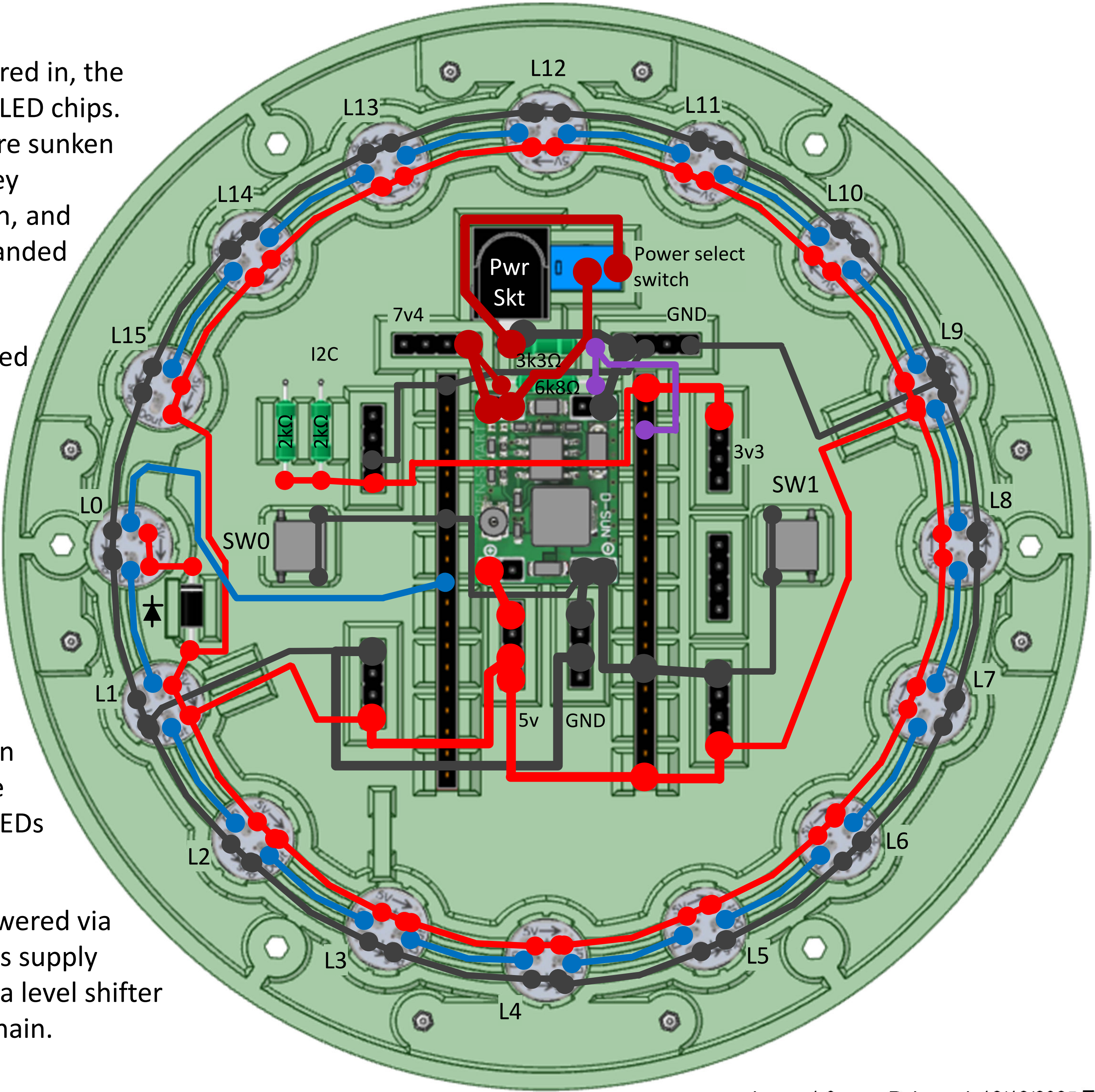
Microplate Wiring – Step 3

With the power connections wired in, the next step is to wire up the RGB LED chips. Moulded into the micro plate are sunken tracks for routing the wires. They Therefore, don’t need insulation, and are formed from bare multi-stranded wire.

The GND and 5v wires are formed in two continuous loops, where as the data wire is broken at each chip, to form a D_{out} to D_{in} connection. The data wire is run in as one continuous loop, but raised at the centre of each chip. Once soldered this raised portion is cropped out, to create the wire break.

Once completed, you can plug in the ESP32 micro and load in the code, to confirm that the RGB LEDs are working as expected.

Note that the first LED, L0 is powered via a 1N4006 diode. This reduces its supply voltage and enable L0 to act as a level shifter for the rest of the LEDs in the chain.



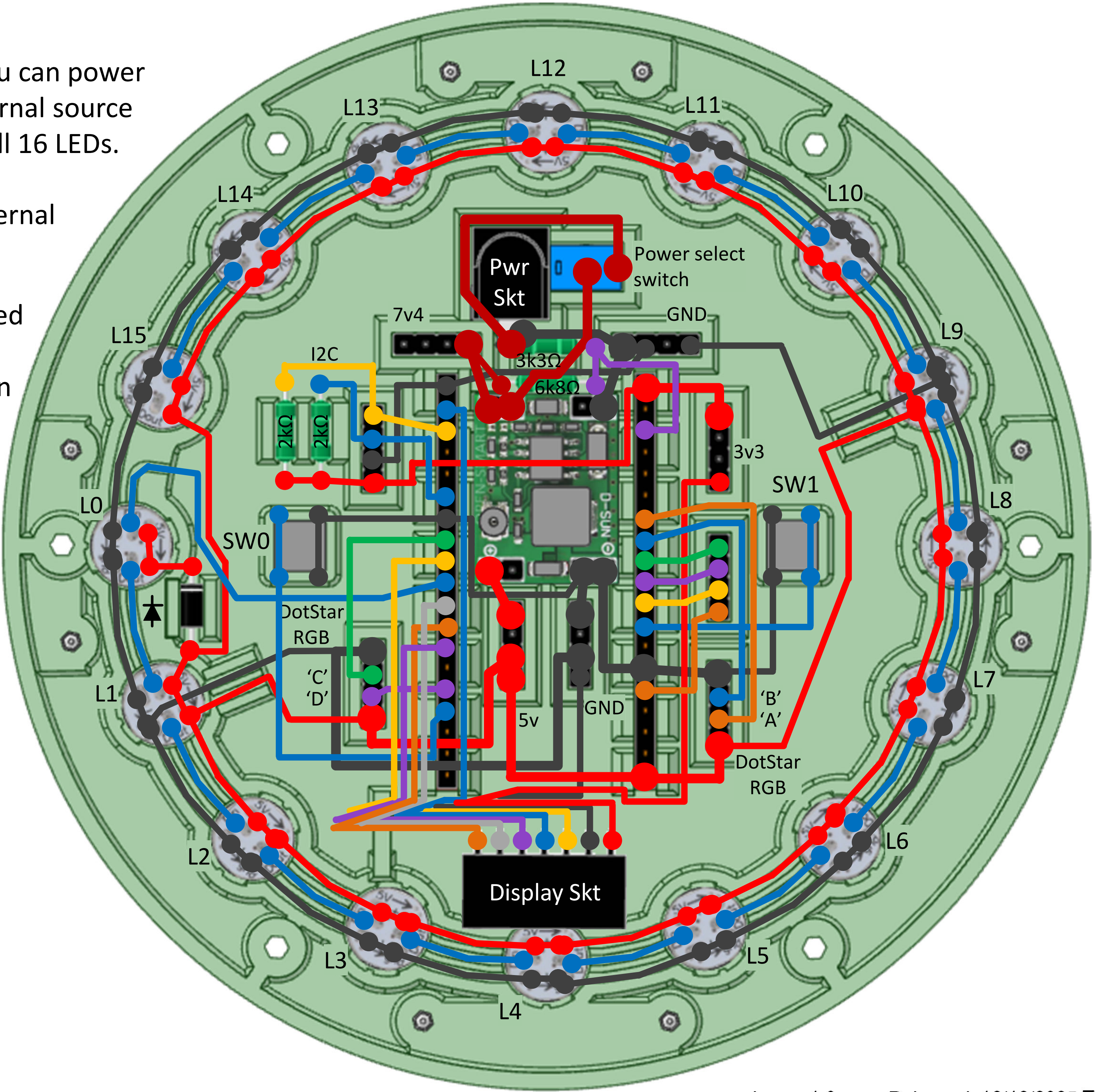
Microplate Wiring – Step 4

With the RGB LEDs wired in, you can power up the micro plate with an external source to ensure that the code lights all 16 LEDs.

Note this works with either external power, or a 5v USB connection.

In this final step we then proceed to wire in the remaining components, including the 7-pin socket for the round display.

The BallBot code loaded into the micro will put something on the display, so you can temporarily power it up off USB and test the interface.



Body Wiring – Step 1

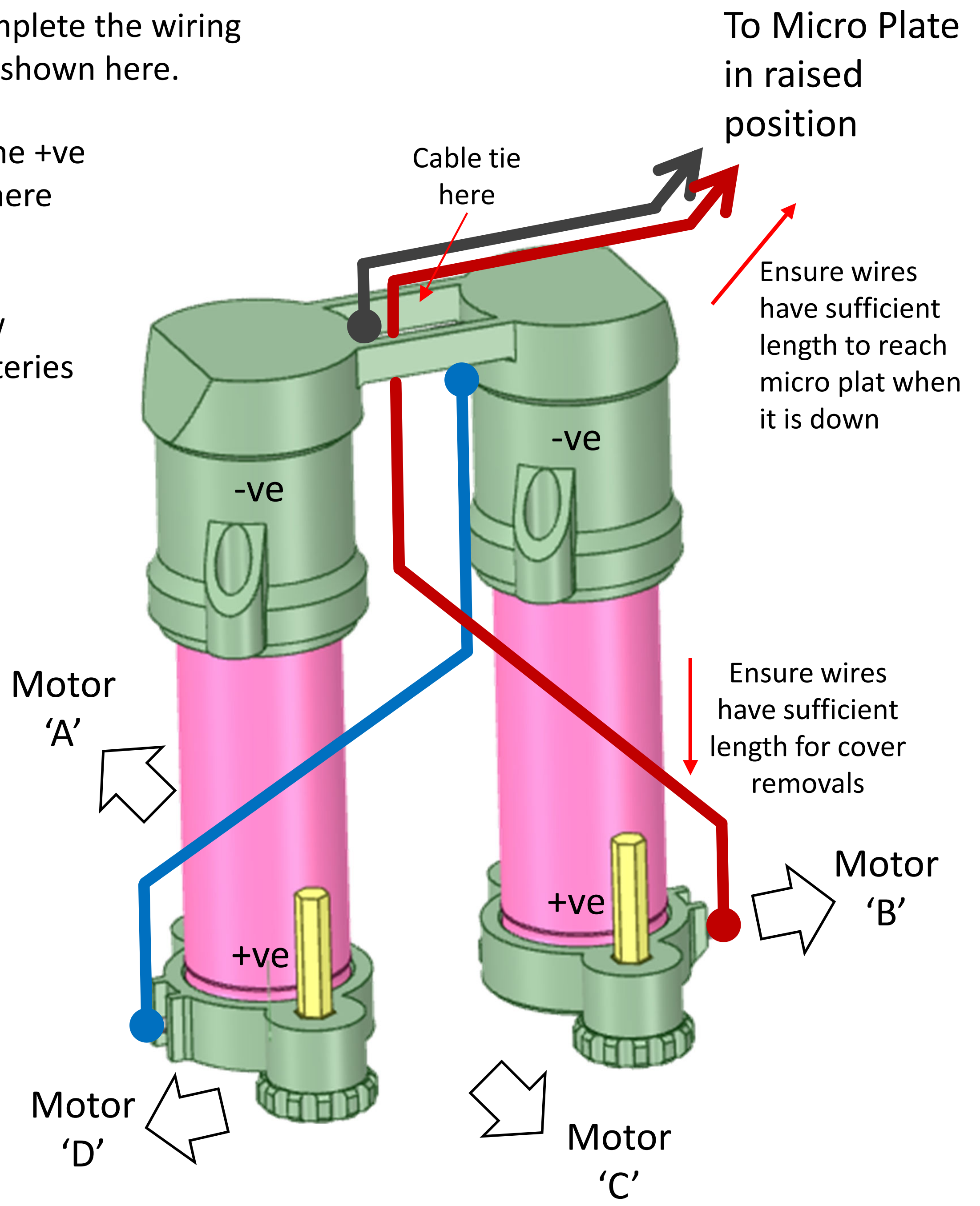
Install the top battery covers, using four 2x10mm screws, and complete the wiring between the battery top covers and lower removeable covers, as shown here.

The two 18650 batteries are connected in series, by connecting the +ve terminal of one, to the negative terminal of the other; as shown here by the blue wire.

The wires to the lower covers must be sufficient in length to allow them to be withdrawn and moved out of the way, so that the batteries can enter the battery tubes.

The batteries must be removed whilst making this wiring. Use a red permanent marker to highlight the positive ends of the battery tubes, to ensure batteries are always inserted correctly. Failure to do so will damage this robot, and could result in a fire.

To get the wires to pass through the centre of the motor plate, it may be necessary to temporary remove one of the lower side plate, to improve access, for feeding the wires through.

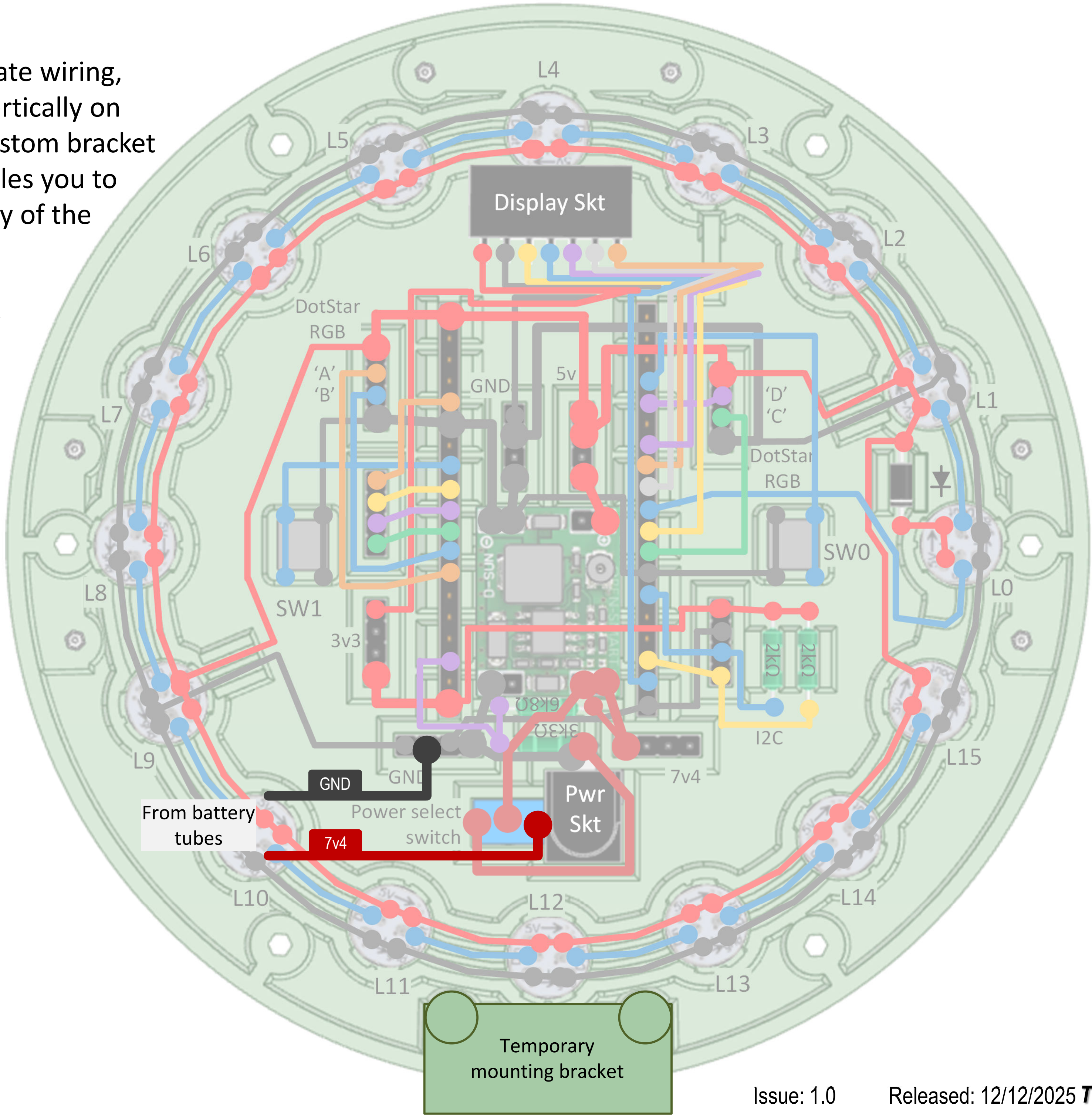


CAUTION – 18650 batteries contain a lot of stored energy, and there is not provision for over-current protection in this design, like a fuse. Only charge batteries using a good quality charger.

Body Wiring – Step 1

Having completed the micro plate wiring, then next step is to mount it vertically on the motor tie ring, using the custom bracket and 2x10mm screws. This enables you to complete the wiring to the body of the robot.

The two wires from the battery tubes can be wired in, to the power switch and GND.

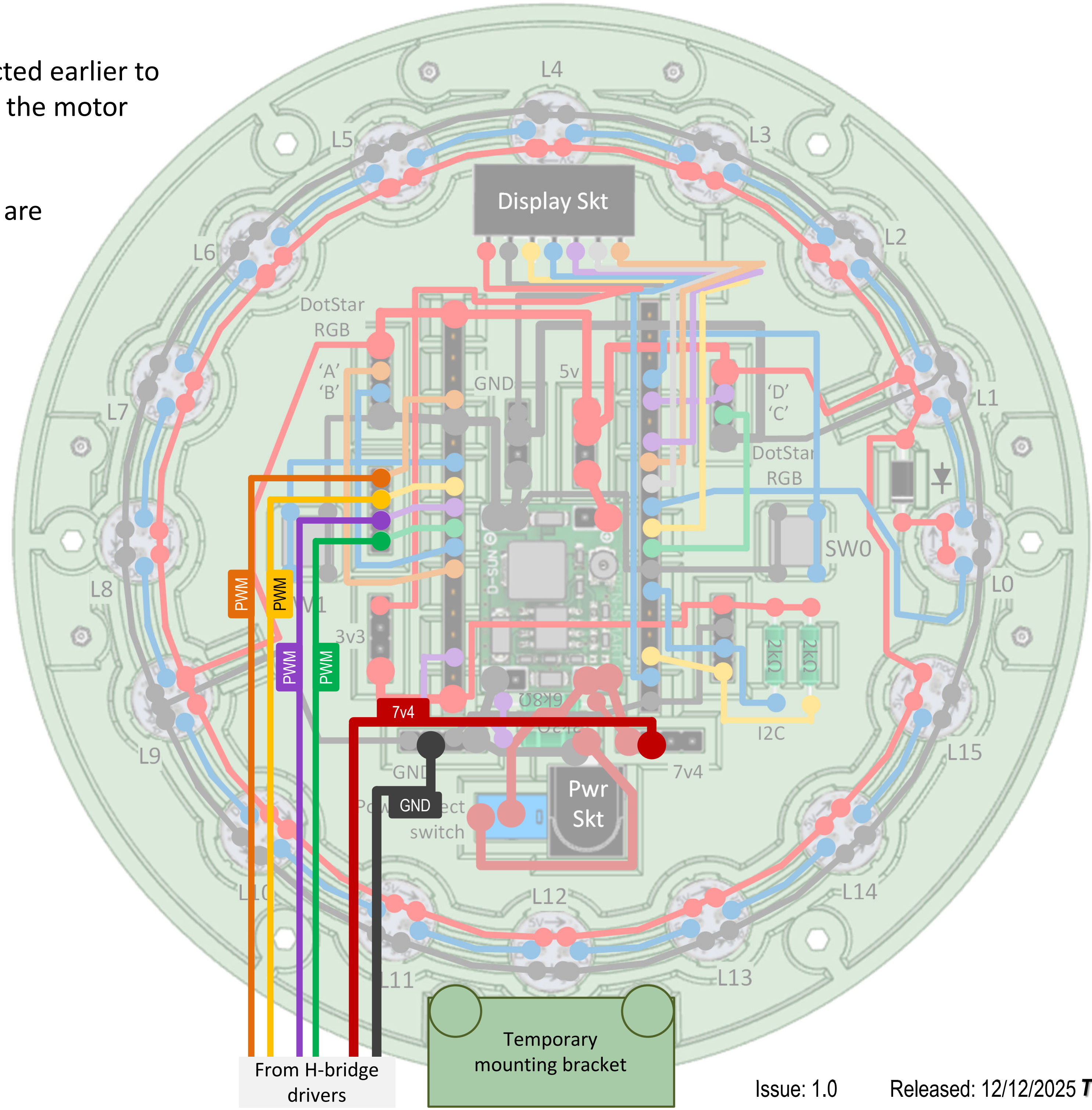


Body Wiring – Step 2

Next connect the wires connected earlier to the H-bridge motor drivers, on the motor plate.

The GND and 7v4 power wires are connected to the common connection pin strips.

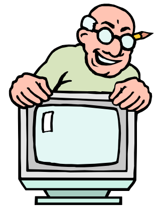
The colour coded PWM wires are connected to the designated 4-pin strip.

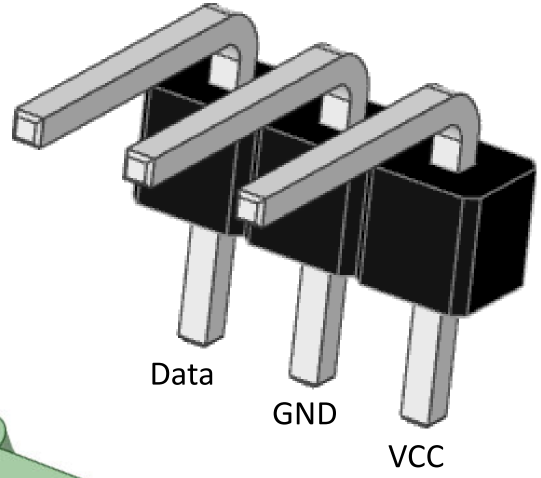


Issue: 1.0 Released: 12/12/2025 TechKnowTone

DotStar Panel Wiring – Step 3

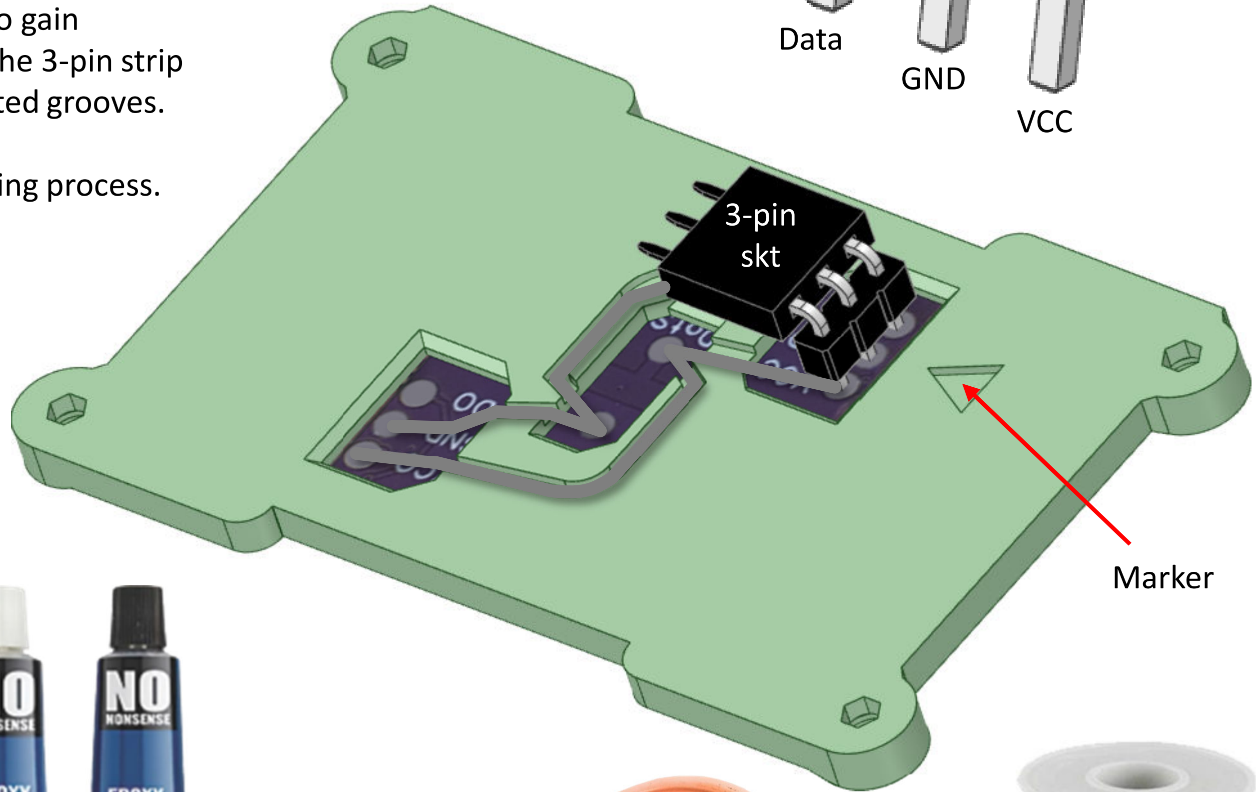
Glue the DotStar arrays into the 3D printed panels, noting their orientation with respect to the printed triangular marker.

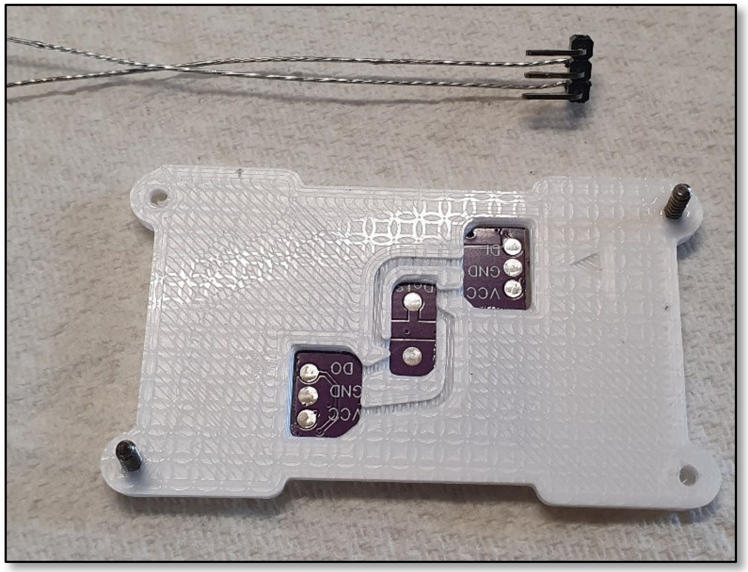




Then solder two wires onto the 3-pin angled strip, prior to soldering it onto the DotStar panel as shown. Note that DotStar panels absorb a lot of heat, and are therefore difficult to solder with a low wattage soldering iron. I recommend that blobs of solder are first applied to the panel connection points, to gain confidence that the iron can supply sufficient heat. Then solder the 3-pin strip and wires to the panel as shown, forming the wires into the printed grooves.

The application of a small amount of flux, should help the soldering process.



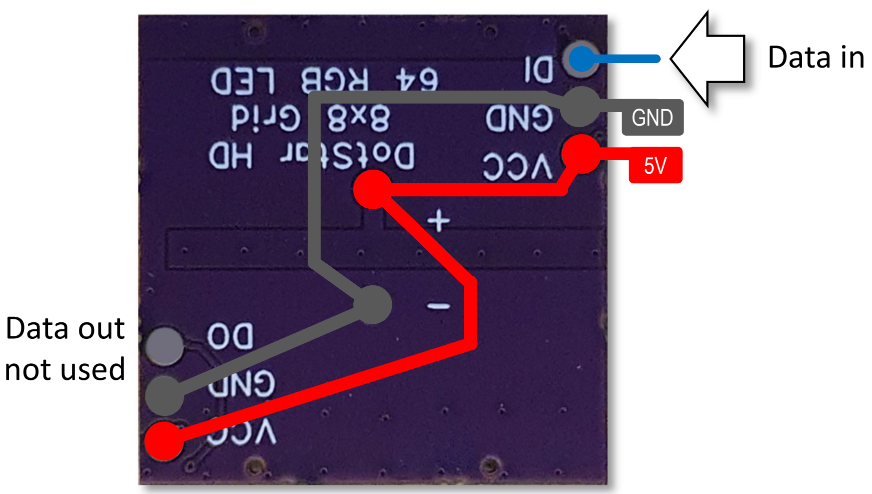


8x8 Panel Wiring

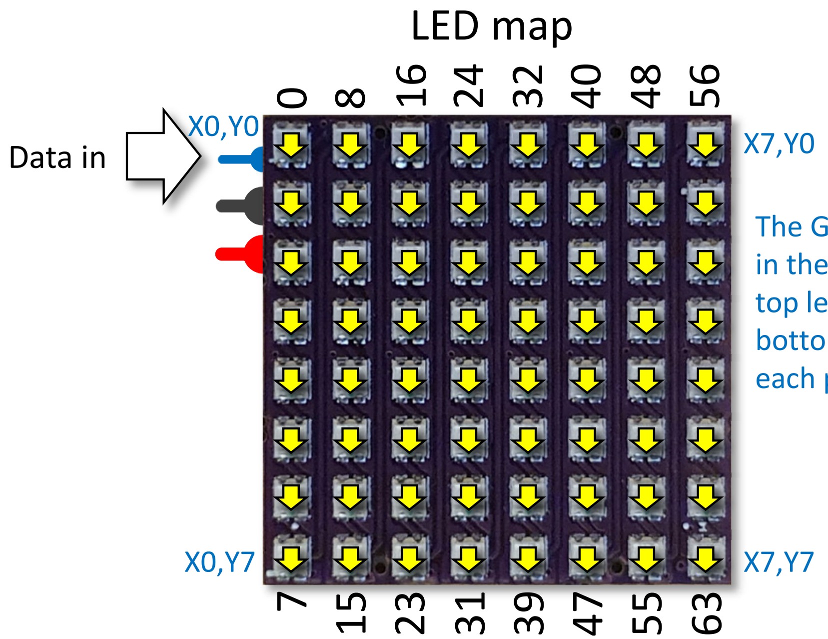
RGB strip code tests, run on a DotStar panel, show that the LEDs will work at 3v3 to 5v, driven via an ESP32C3, running Adafruit NeoPixel. Whilst the ESP32C3 works at 3v3, there does not appear to be any need for a level-shifting LED, as with the WS2812B LEDs on the top micro plate; so, one is not used in this project. The diagrams below show how each panel is wired, and the yellow arrows show how the LEDs are connected in series within each panel.

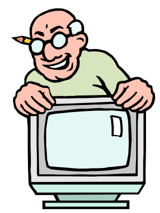
When viewed from the front, with data input is fed from the top left, the top left LED ‘0’ is the first in the chain of LEDs. The next LED ‘1’ is below that, and so on, making the bottom left LED ‘7’. The LED chain then restarts at the top ‘8’, and follows the same sequence down the panel. Such that finally, the bottom right LED is number ‘63’ in the daisy chain.

All of the panels are wired in the same, but from separate independent controller pins, within the ESP32 micro. The DS_GFX functions within the code, provide a means by which coloured graphics can be sent to each panel.



Viewed from the rear





Viewed from the front

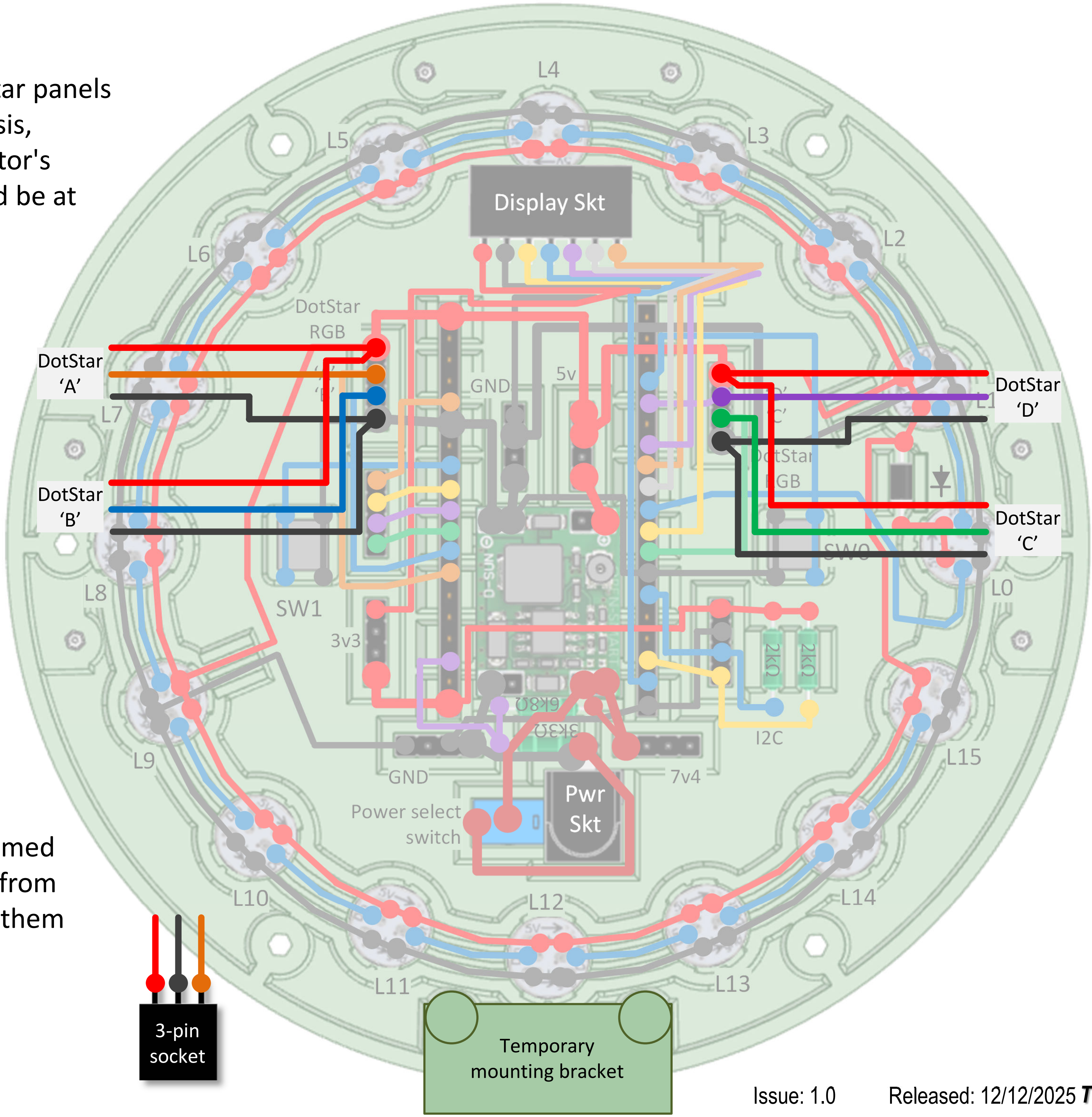
Body Wiring – Step 3

The AWG30 wires for the DotStar panels are fed down through the chassis, following the route of each motor's wires. Each 3-wire group should be at least 28cm long initially.

Wire wrap the 3-pin sockets first, at the motor end, and draw back excess wiring. Use heat shrink sleeving to help group the wired. Pass them through the centre of the upper batter cap, and terminate on the pins on the micro plate.

The great thing about wire wrapping is that you can test the panels as you go, before soldering the joints at the end.

The wire gauge is light, but deemed sufficient to get sufficient light from the DotStar panels. Do not run them on maximum brightness.

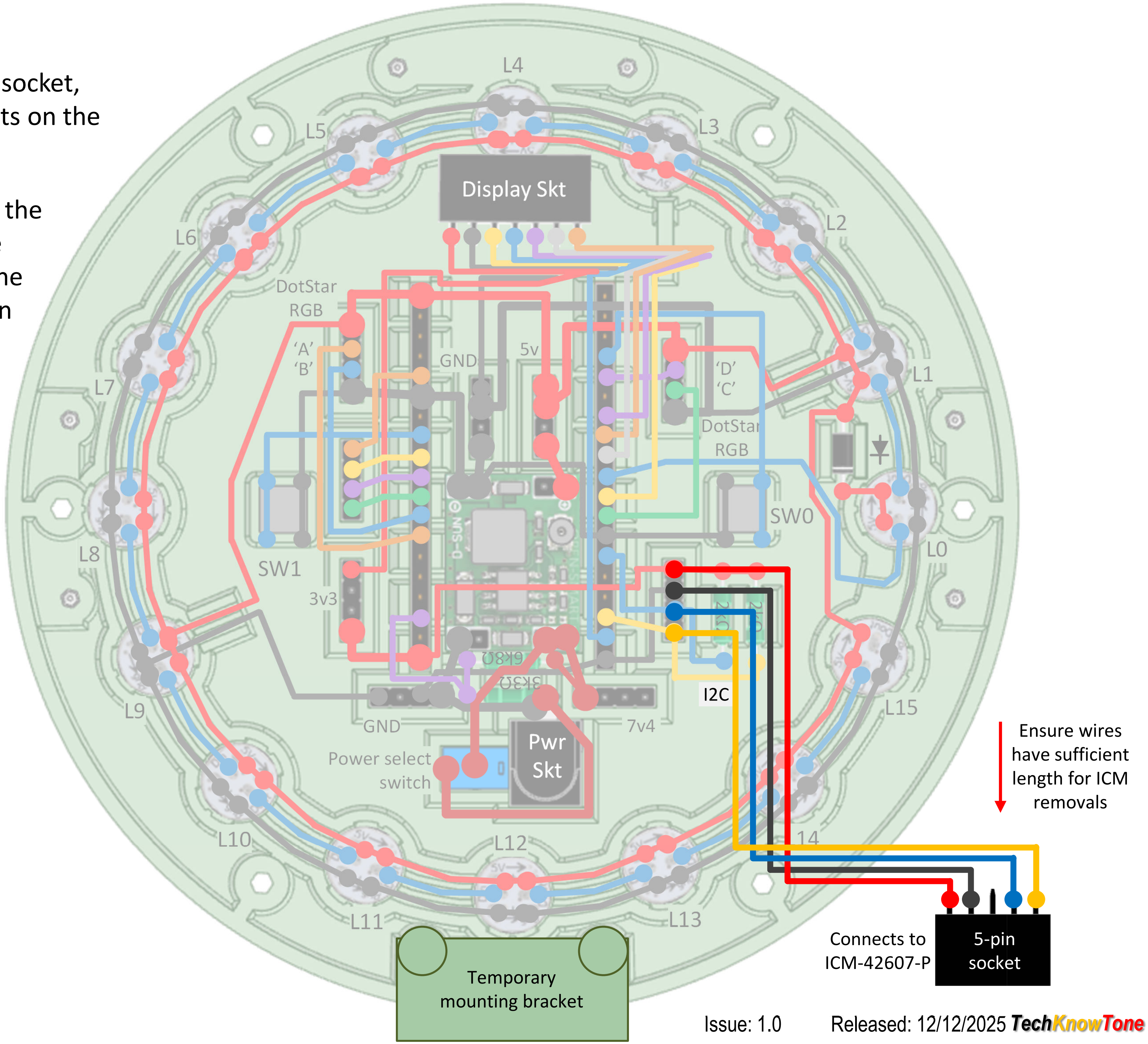


Body Wiring – Step 4

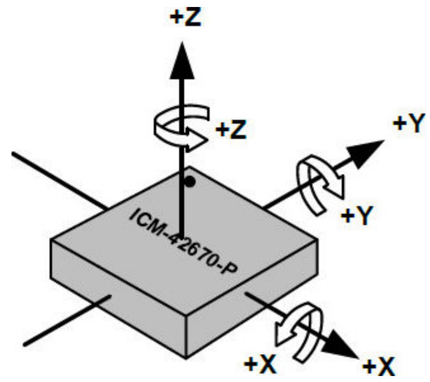
The next step is to wire a 5-pin socket, to the four I2C connection points on the micro plate.

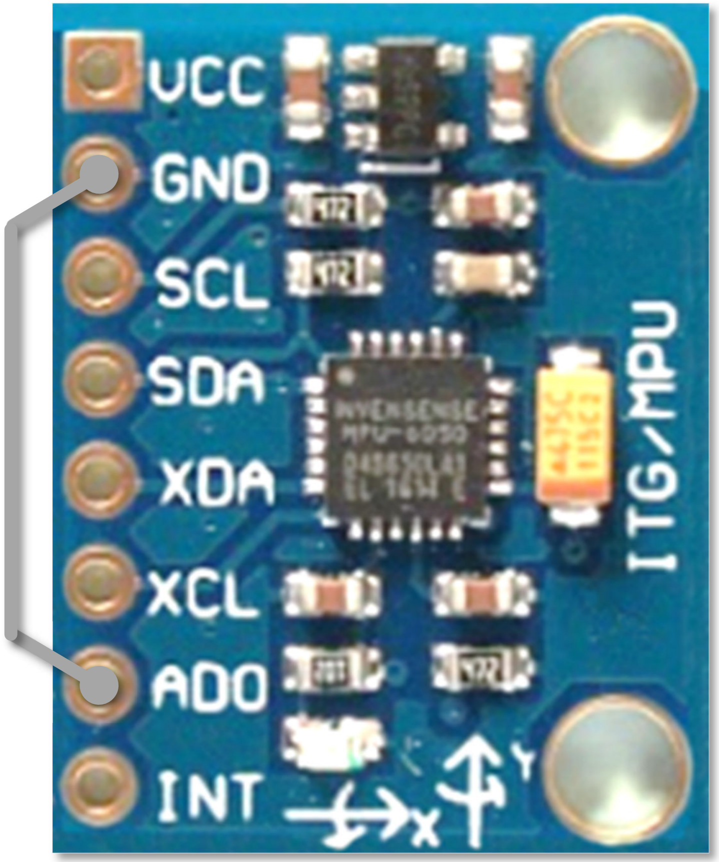
Wire the socket first, then feed the wires through the centre of the chassis plate, and up through the centre of the body, to emerge in the centre of the battery cap. Then wire them to the micro plate.

Allow sufficient length so that the ICM-42607-P can be plugged into the socket, before being screwed onto the chassis plate.



4x4 Ball Balancing Robot
MPU6050 Motion Sensor





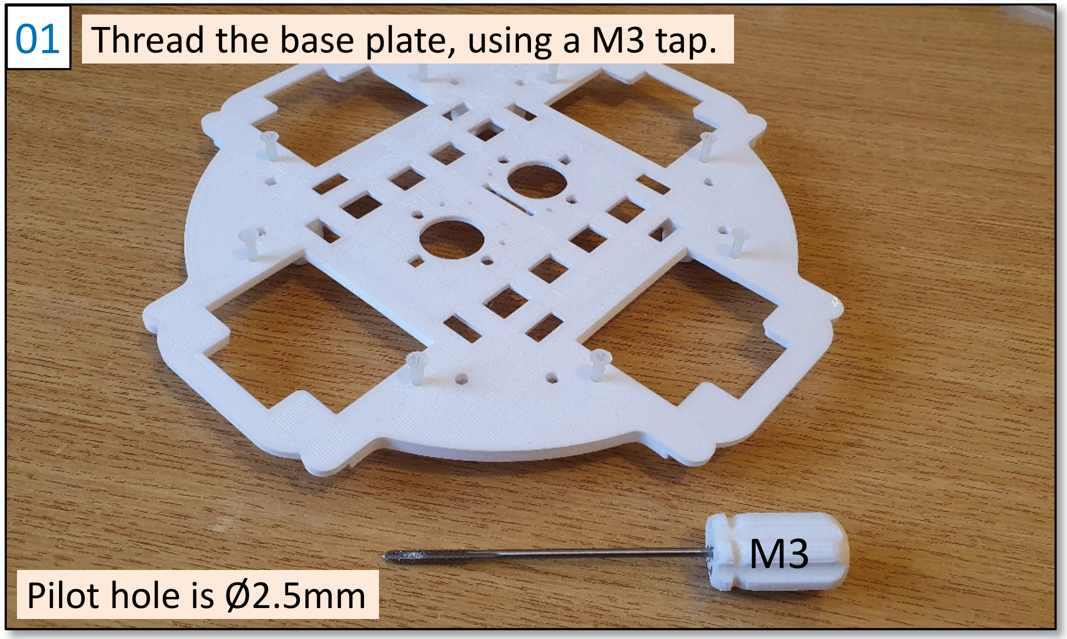
The MPU-6050 is an integrated 6-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package. With its dedicated I2C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis Motion Fusion™ output.

The MPU-6050 Motion Tracking device, with its 6-axis integration, on-board Motion Fusion™, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers.

The MPU-6050 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its auxiliary I2C port.

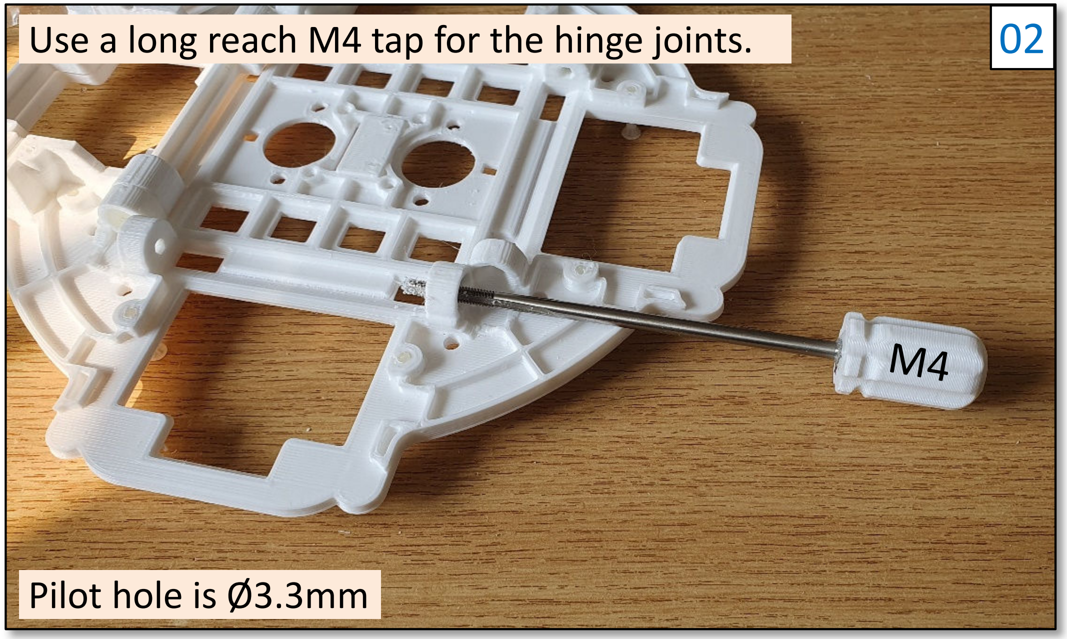
Pull AD0 LOW using a solder bridge, as shown, to set the devices I2C slave address as 0x68 hex. If pulled HIGH, to VCC, it will be 0x69 hex.

Build Images



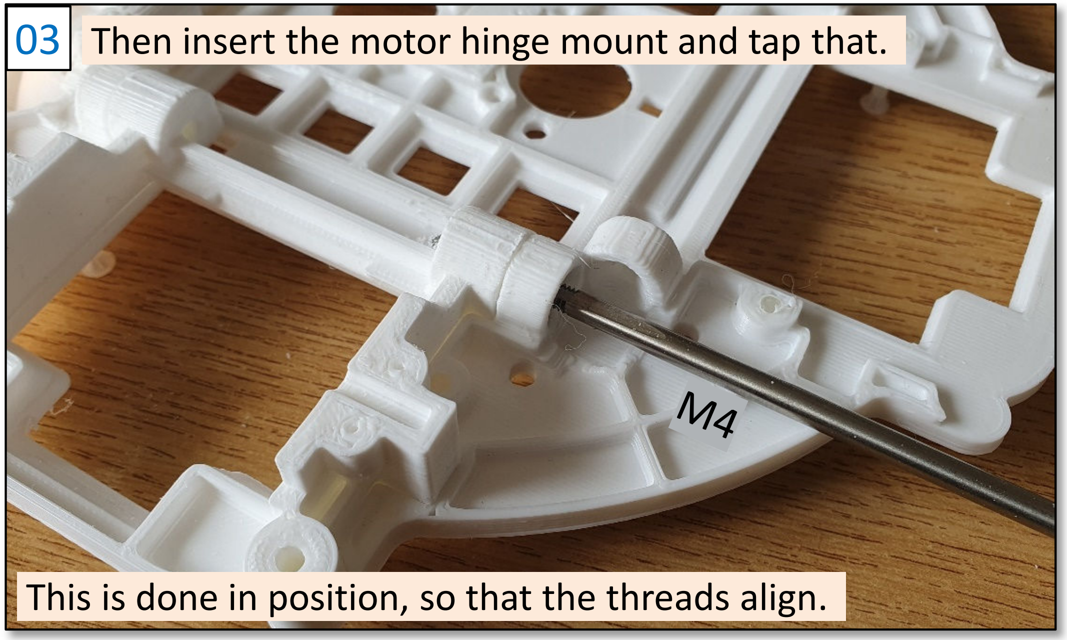
01 Thread the base plate, using a M3 tap.

Pilot hole is Ø2.5mm



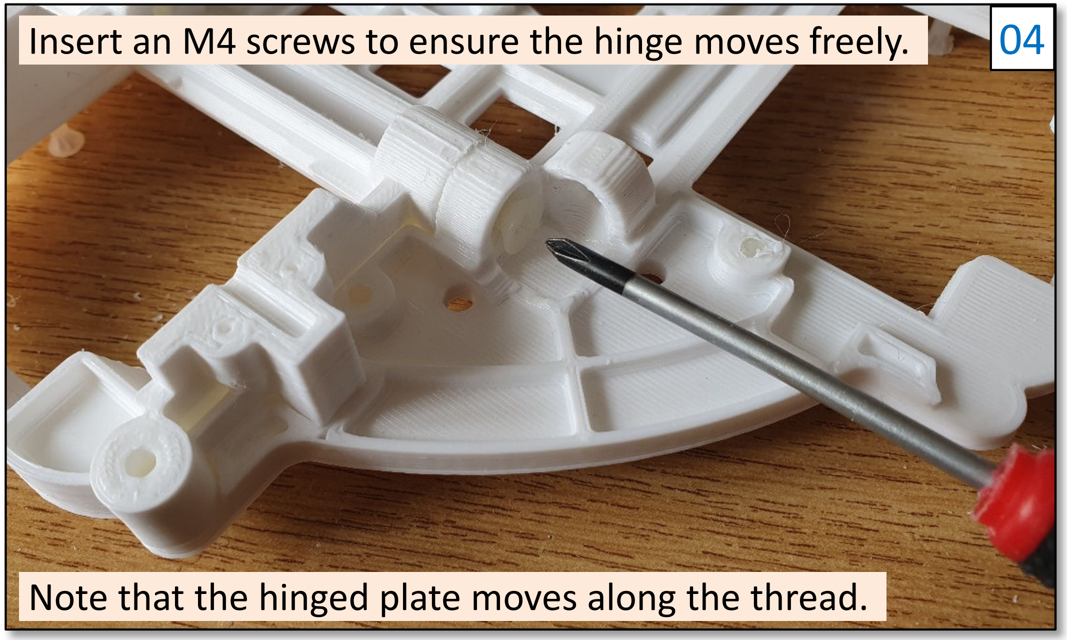
02 Use a long reach M4 tap for the hinge joints.

Pilot hole is Ø3.3mm



03 Then insert the motor hinge mount and tap that.

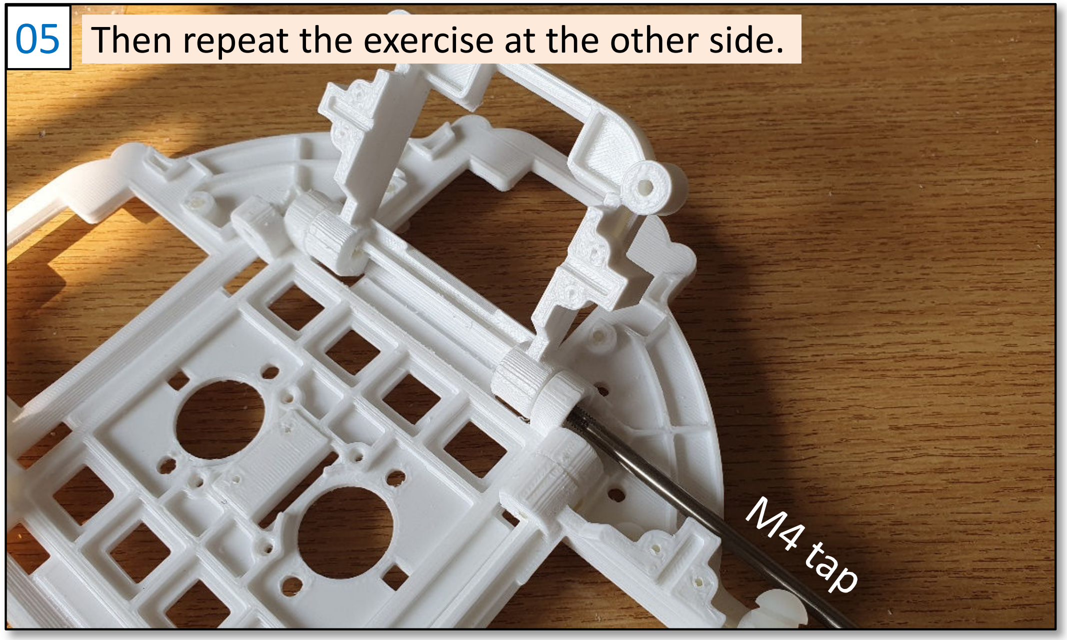
This is done in position, so that the threads align.

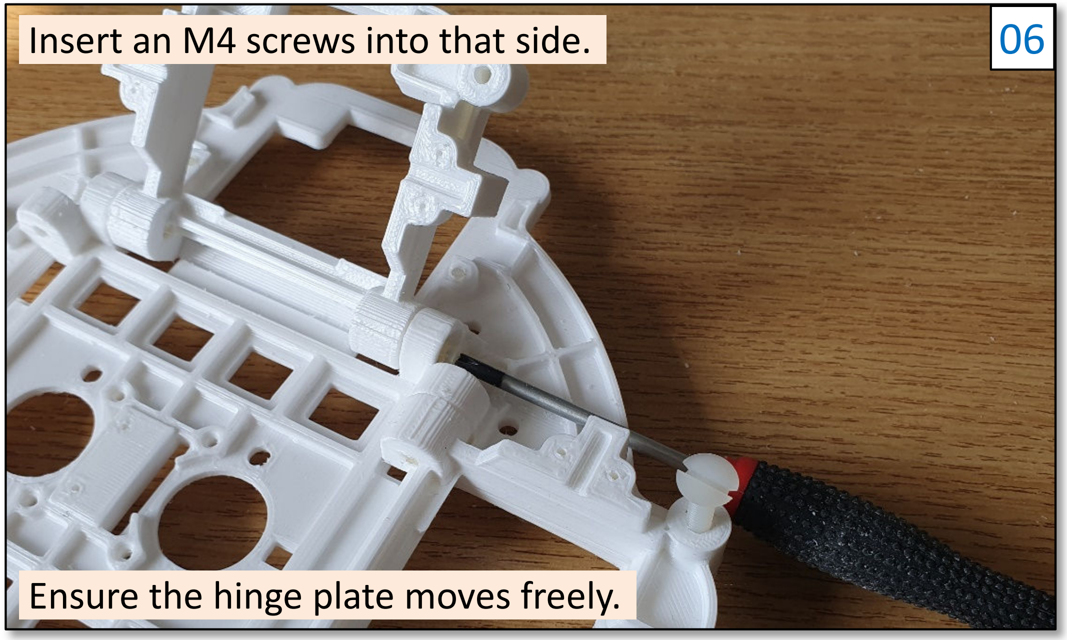


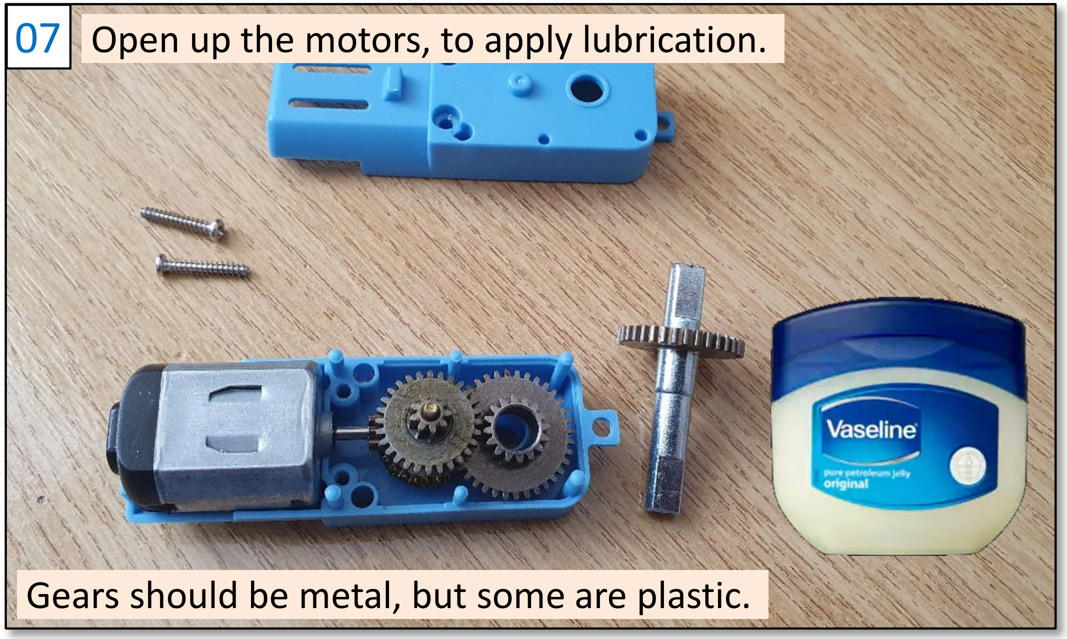
04 Insert an M4 screws to ensure the hinge moves freely.

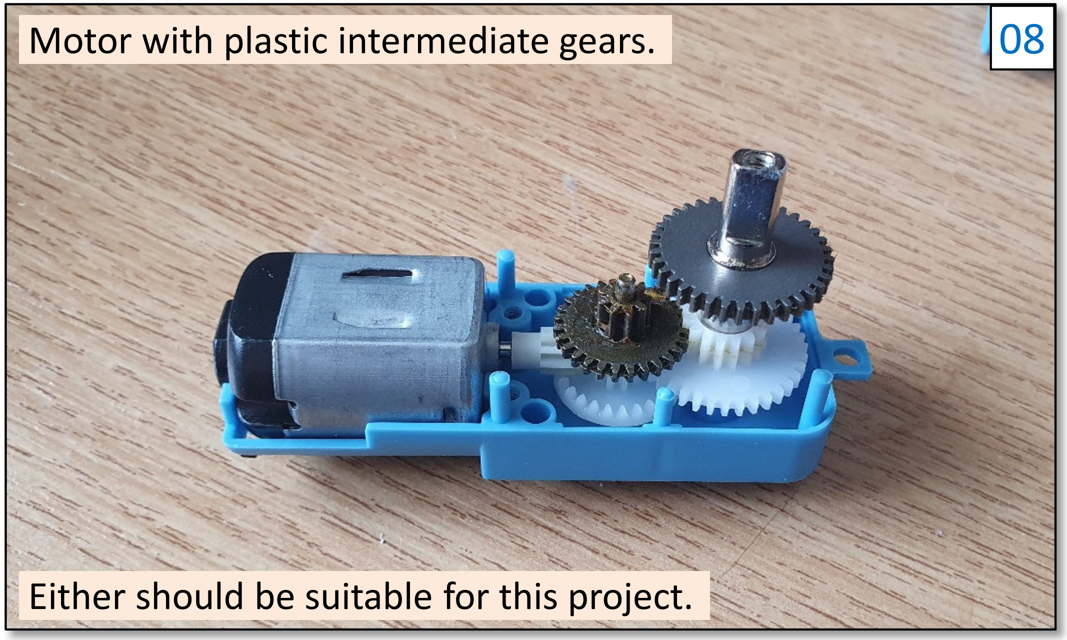
Note that the hinged plate moves along the thread.

Build Images

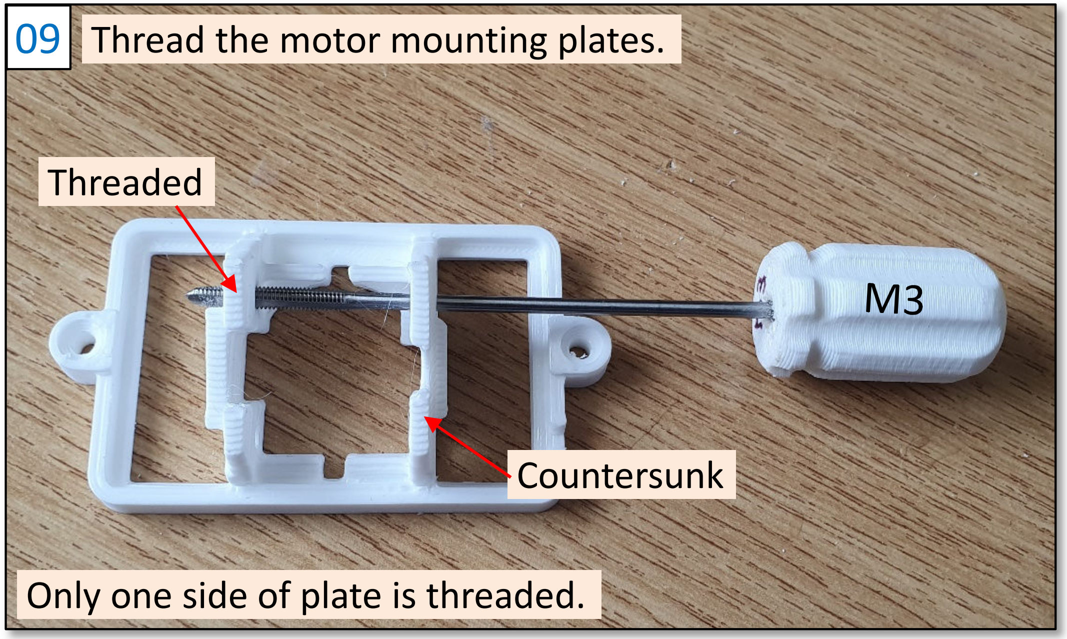


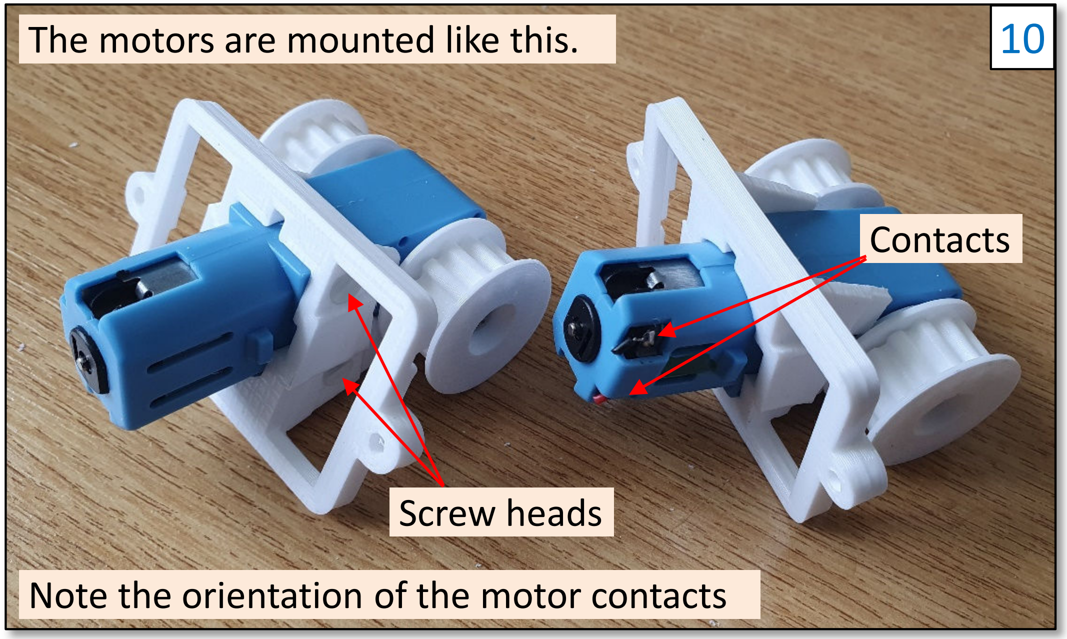


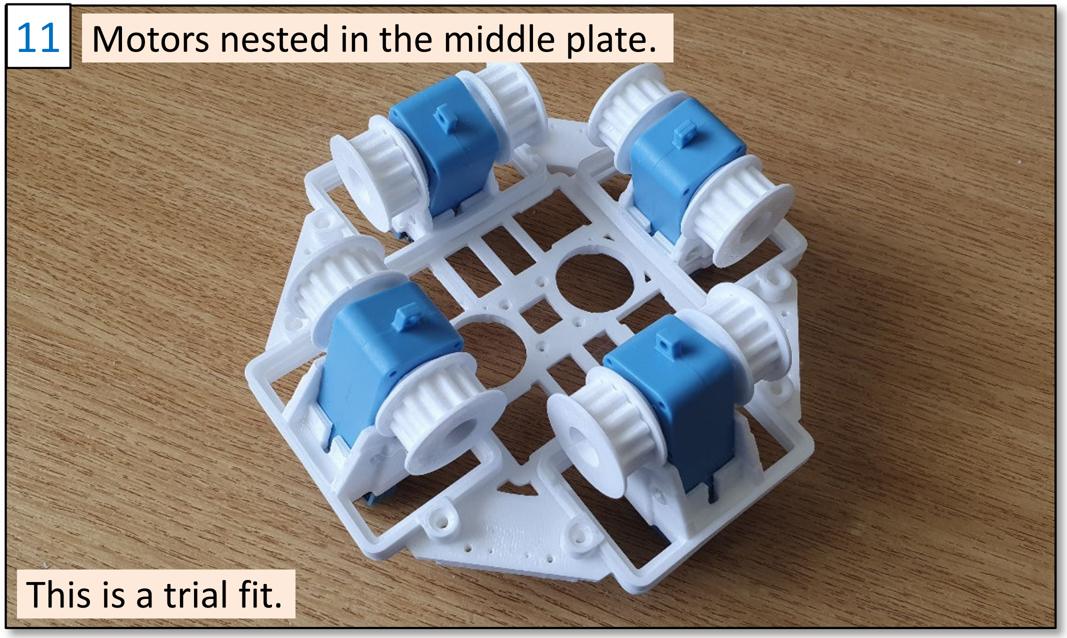


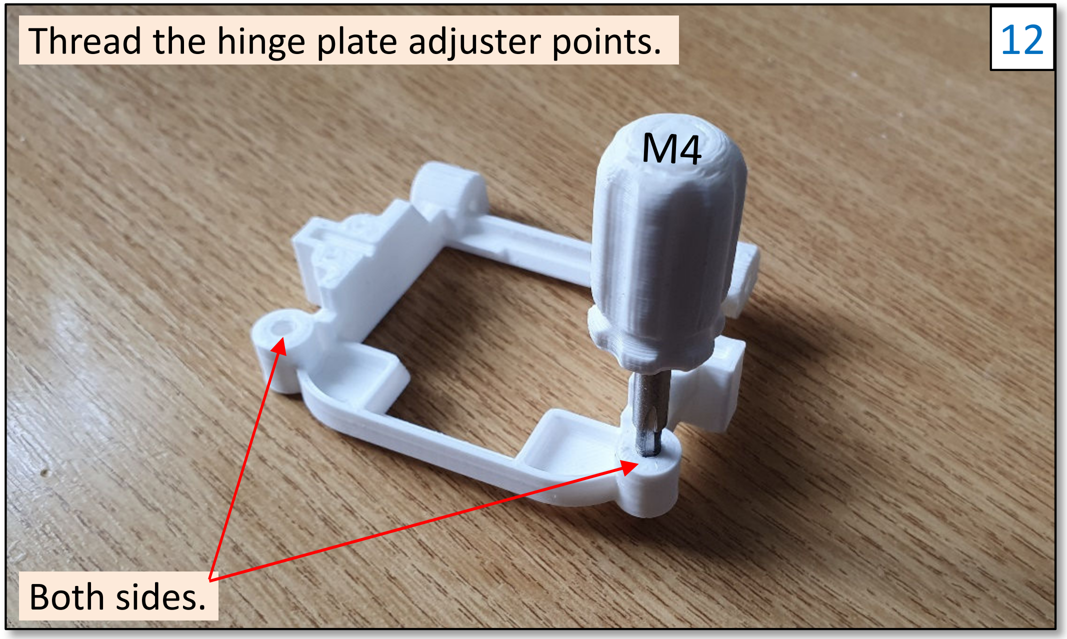


Build Images



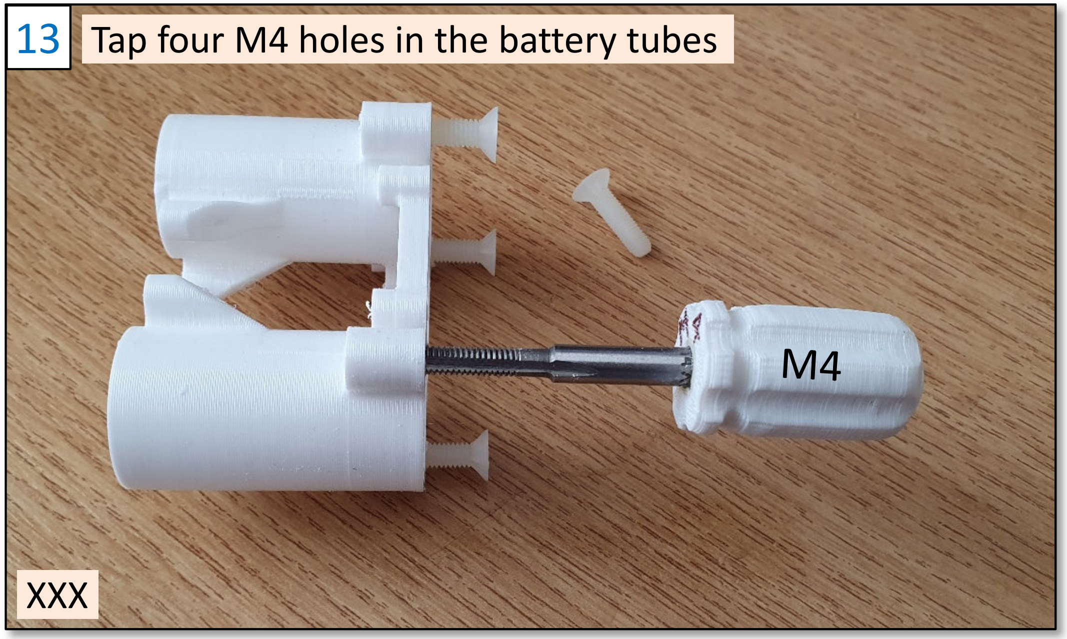






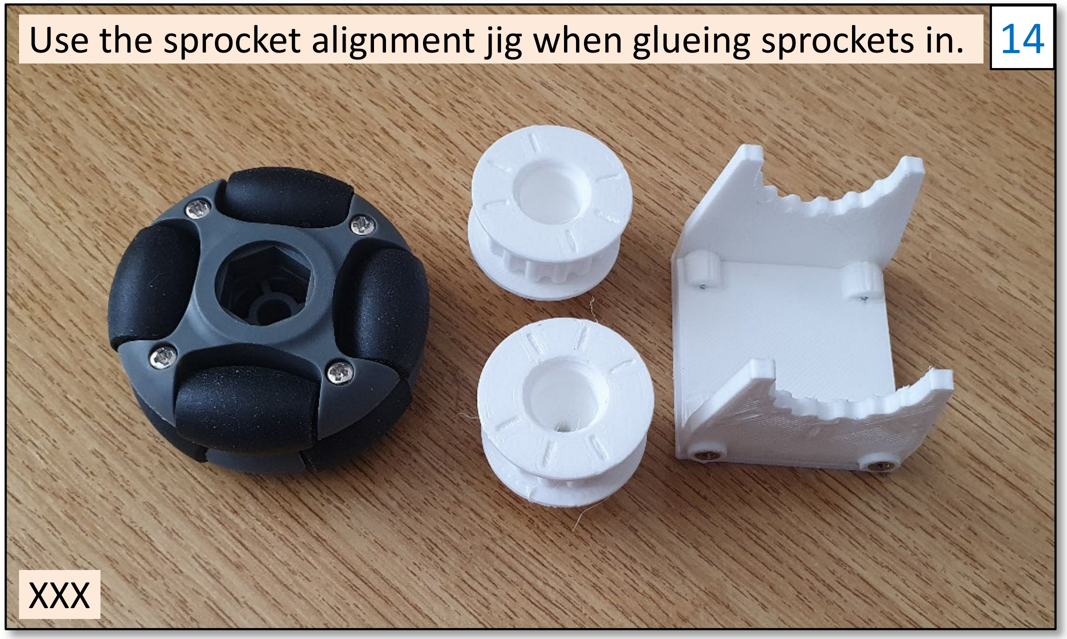
Build Images

13 Tap four M4 holes in the battery tubes



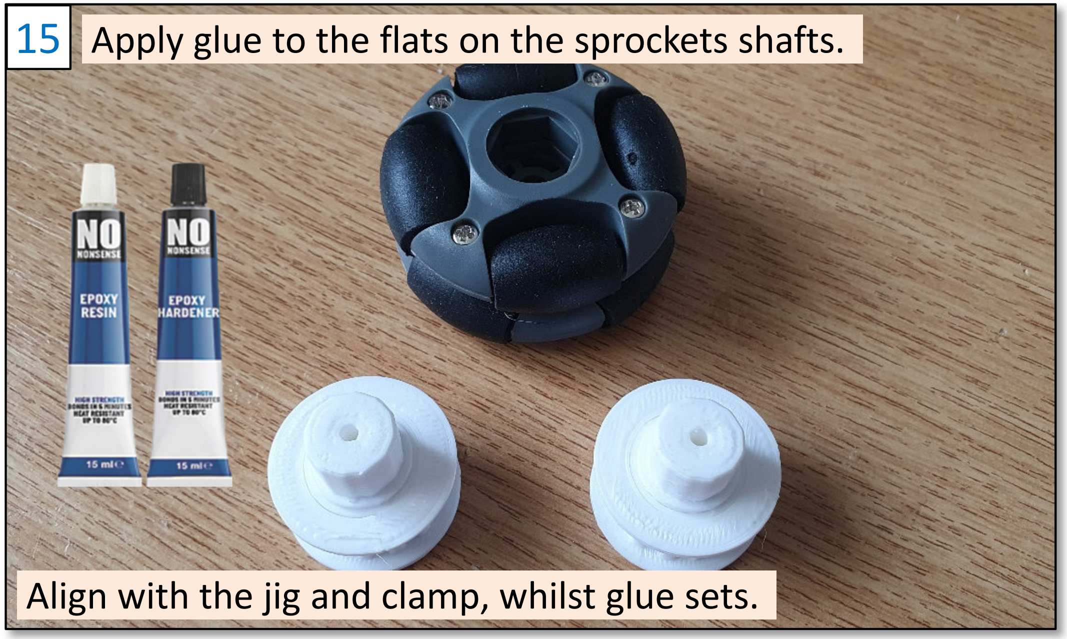
XXX

14 Use the sprocket alignment jig when glueing sprockets in.



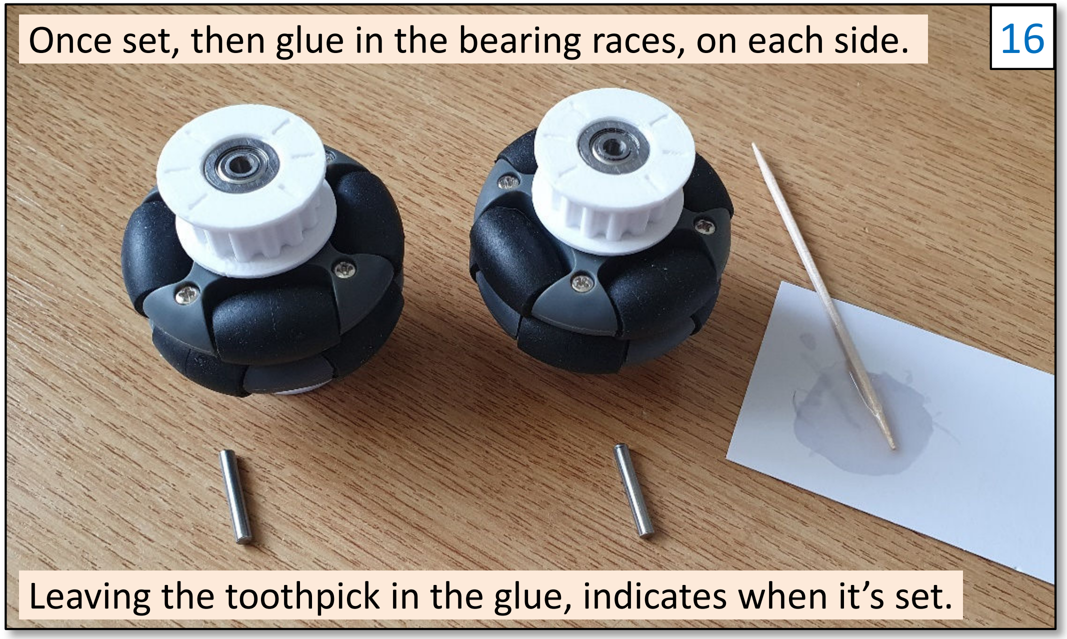
XXX

15 Apply glue to the flats on the sprockets shafts.



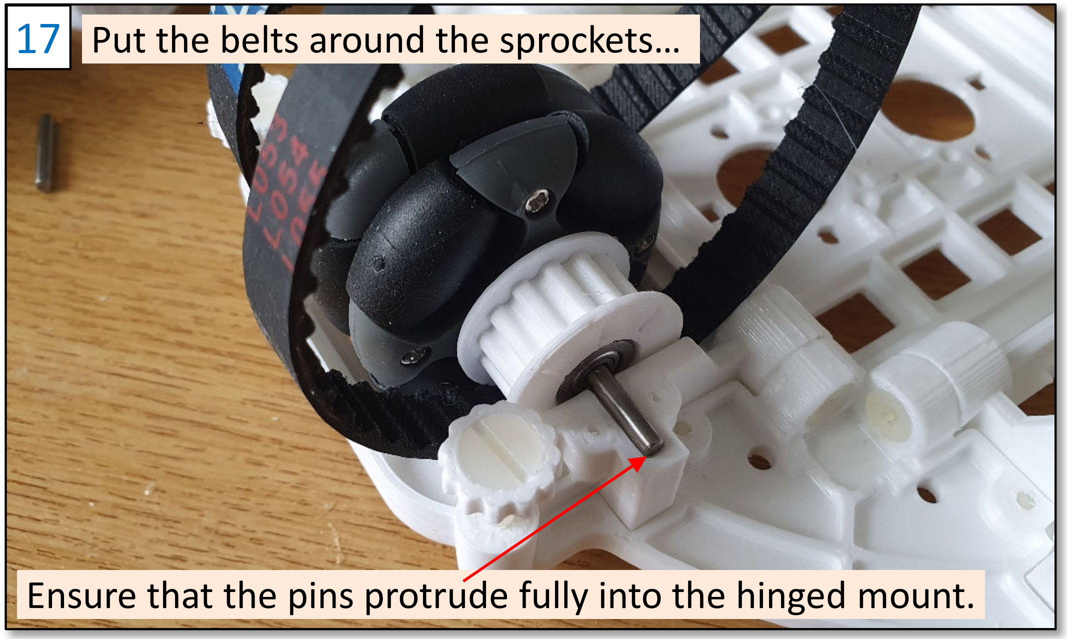
Align with the jig and clamp, whilst glue sets.

16 Once set, then glue in the bearing races, on each side.



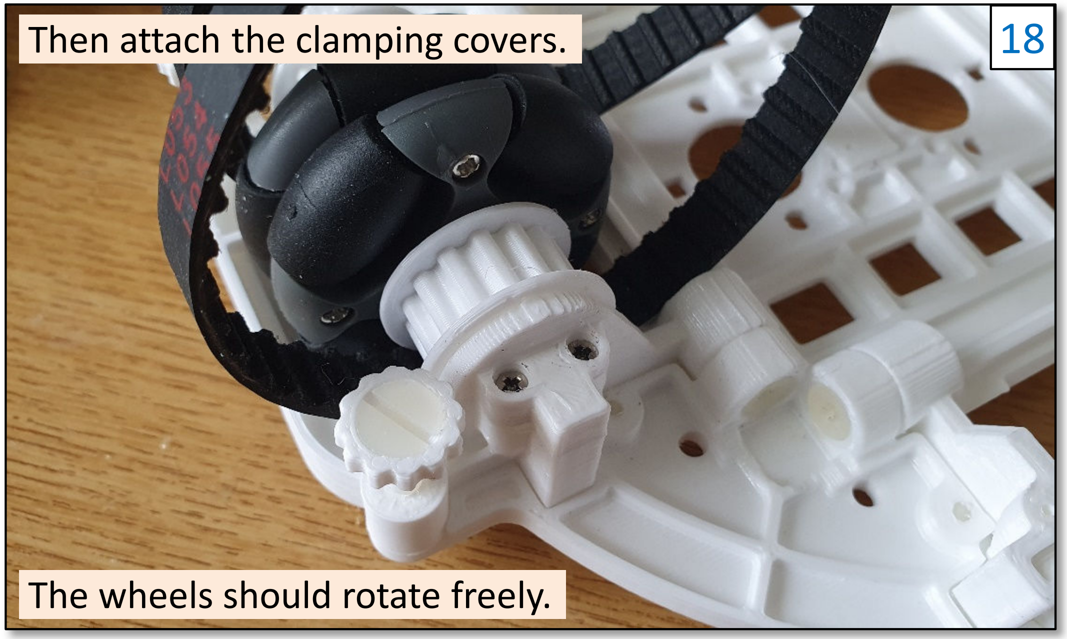
Leaving the toothpick in the glue, indicates when it's set.

Build Images



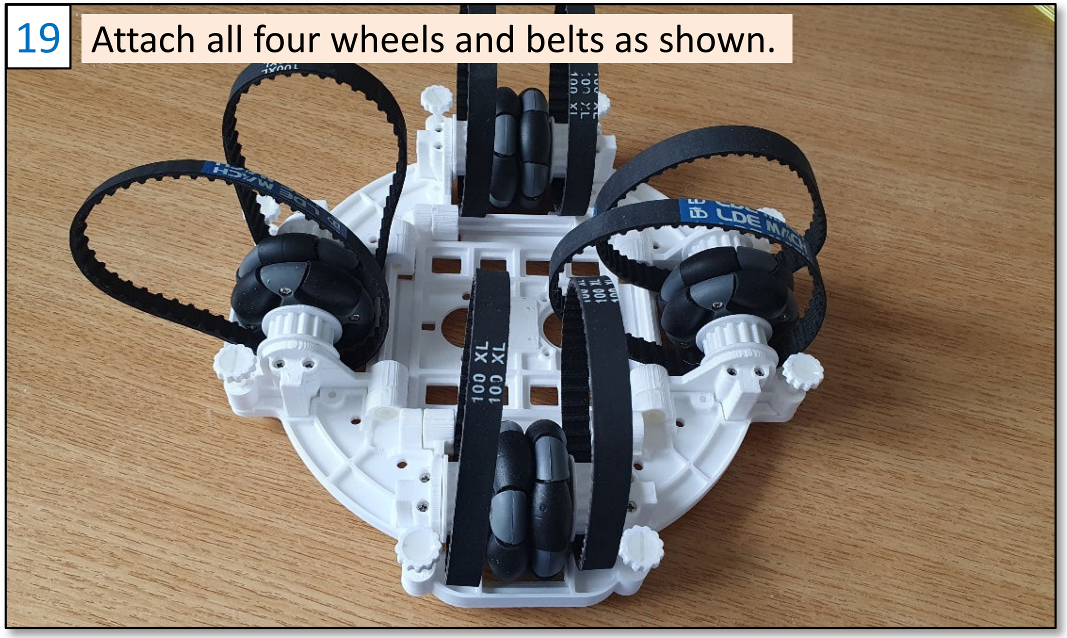
17 Put the belts around the sprockets...

Ensure that the pins protrude fully into the hinged mount.

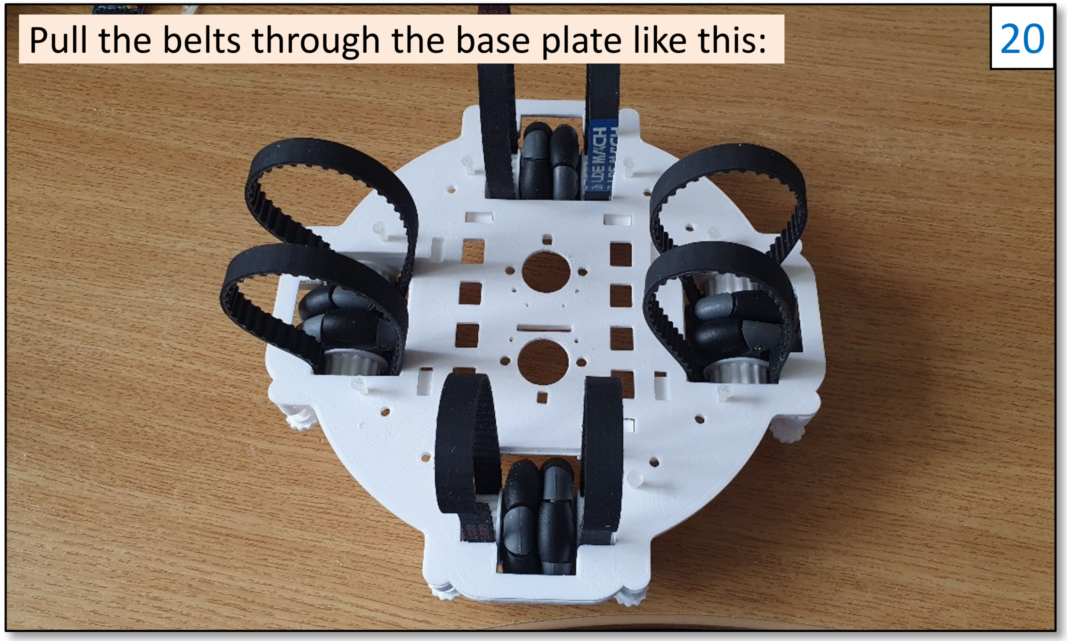


18 Then attach the clamping covers.

The wheels should rotate freely.

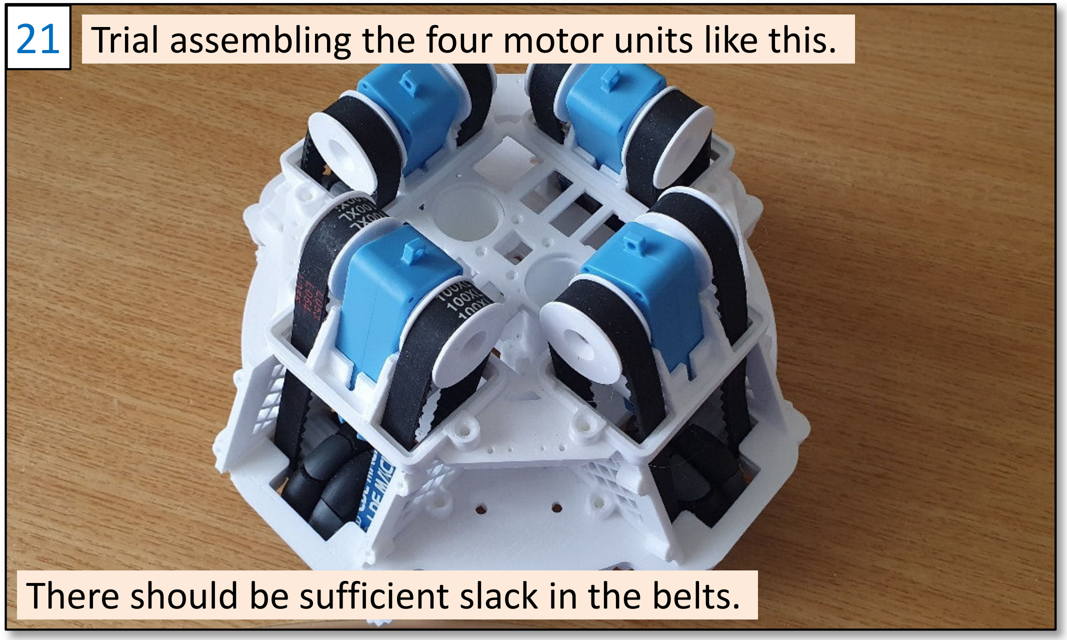


19 Attach all four wheels and belts as shown.



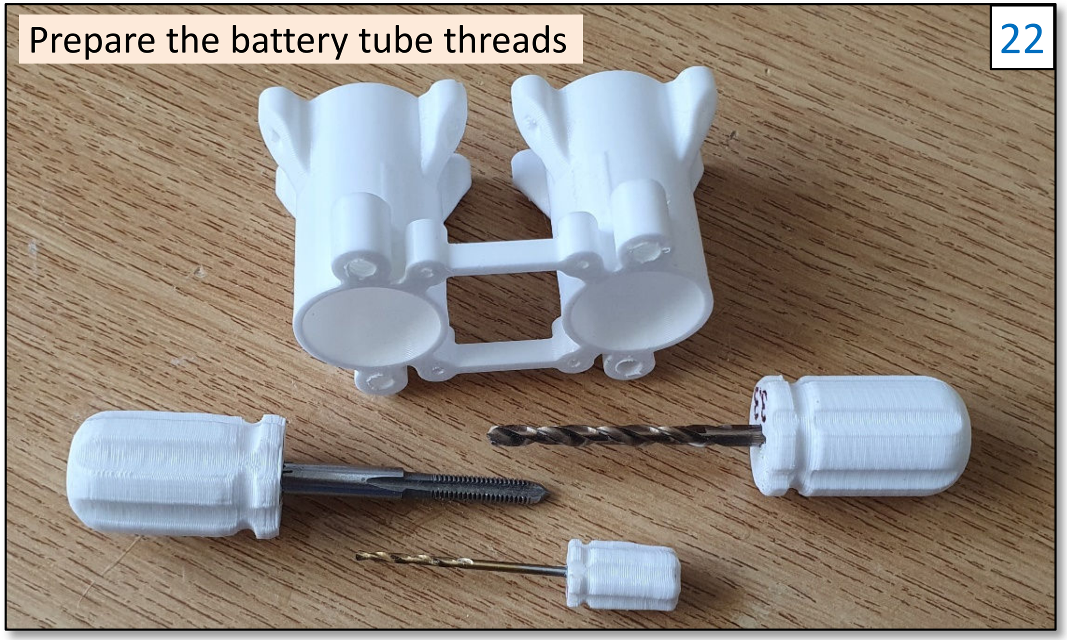
20 Pull the belts through the base plate like this:

Build Images

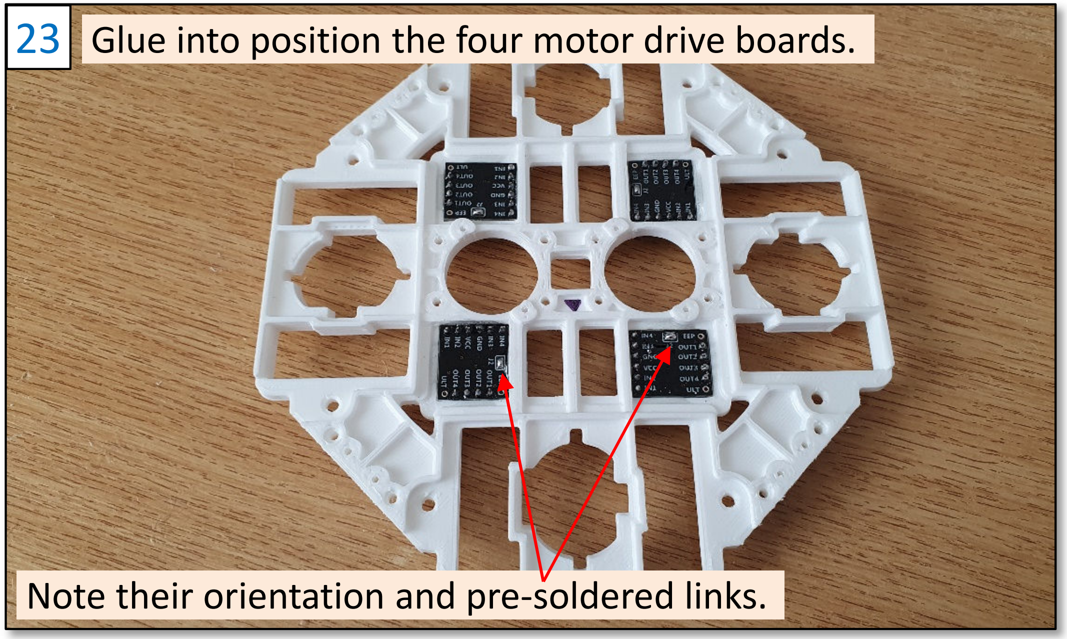


21 Trial assembling the four motor units like this.

There should be sufficient slack in the belts.

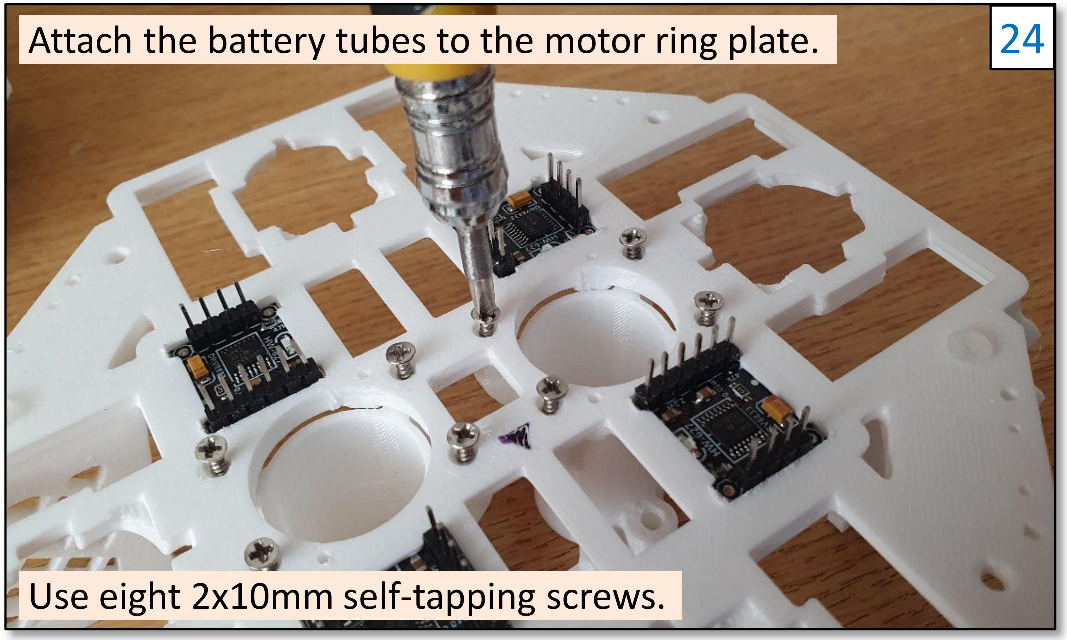


22 Prepare the battery tube threads



23 Glue into position the four motor drive boards.

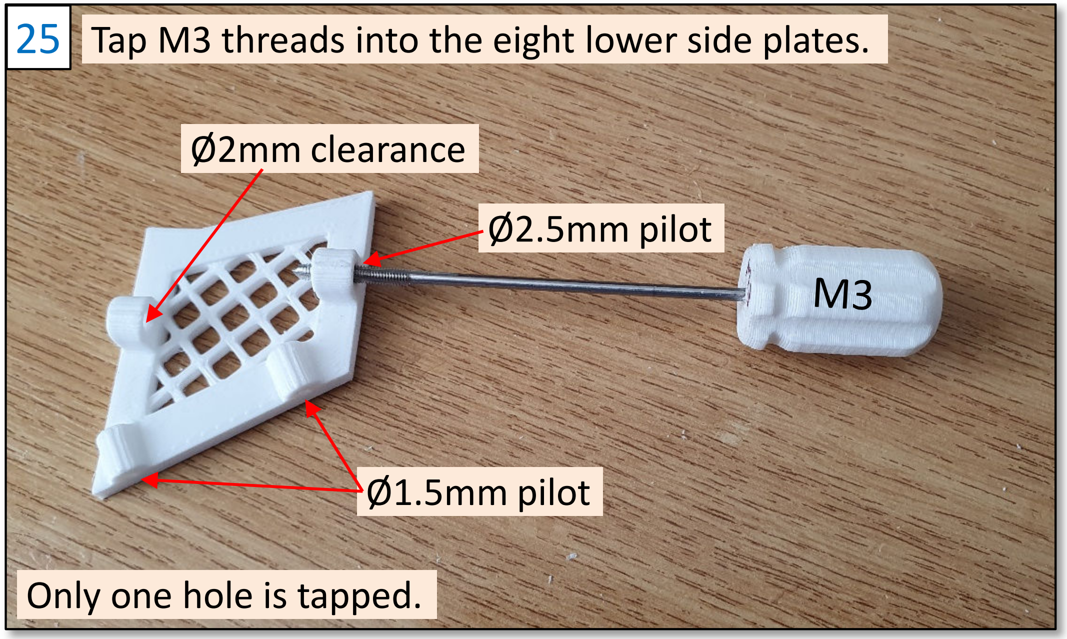
Note their orientation and pre-soldered links.



24 Attach the battery tubes to the motor ring plate.

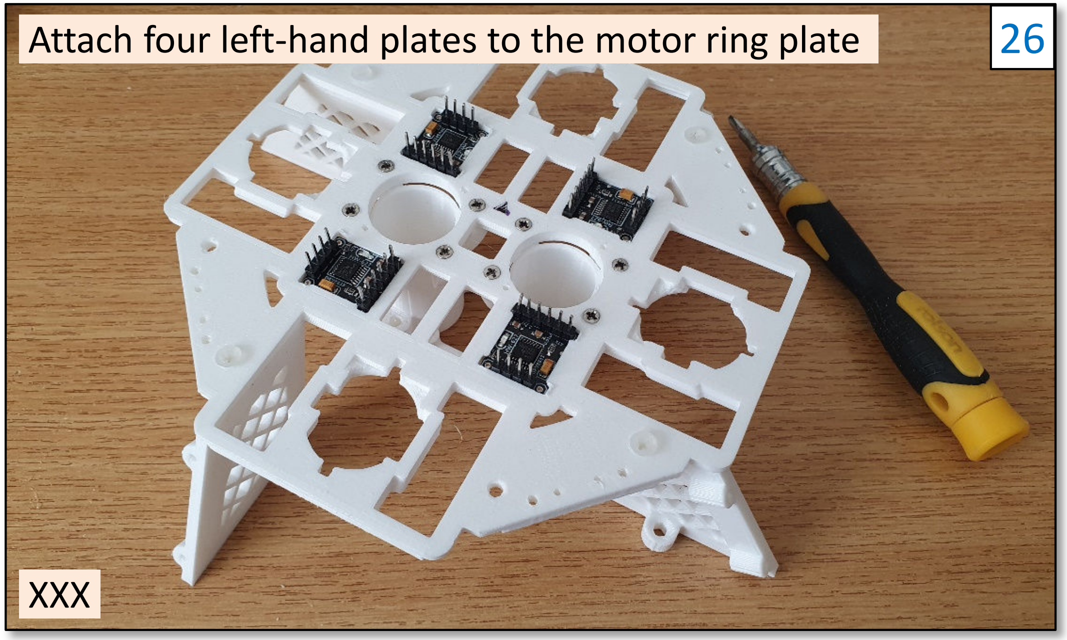
Use eight 2x10mm self-tapping screws.

Build Images



25 Tap M3 threads into the eight lower side plates.

Only one hole is tapped.

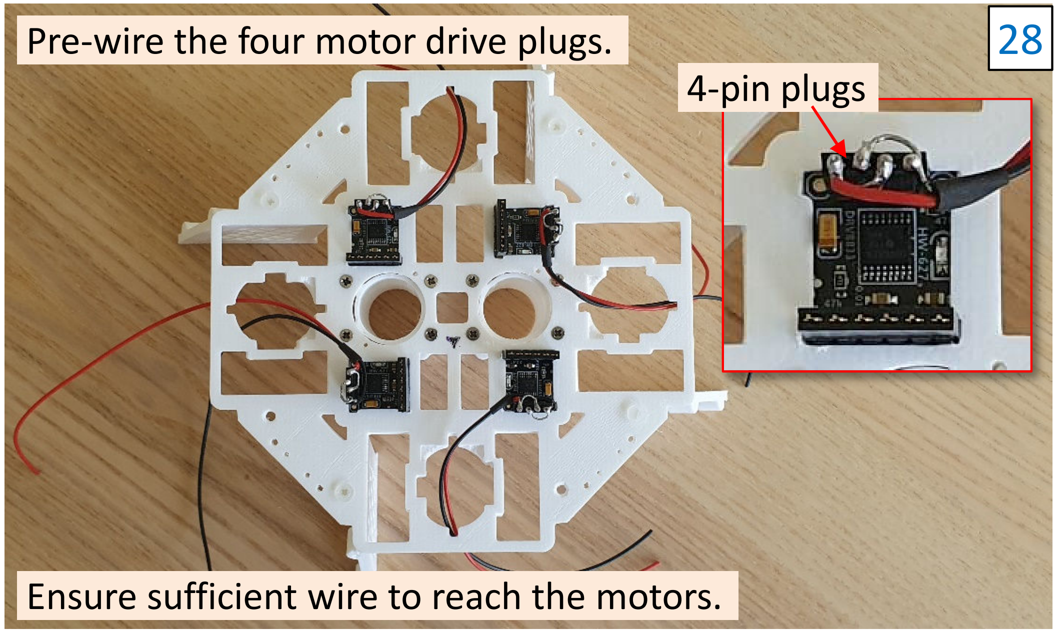


26 Attach four left-hand plates to the motor ring plate

XXX

27 Glue the two battery springs into the battery tubes.

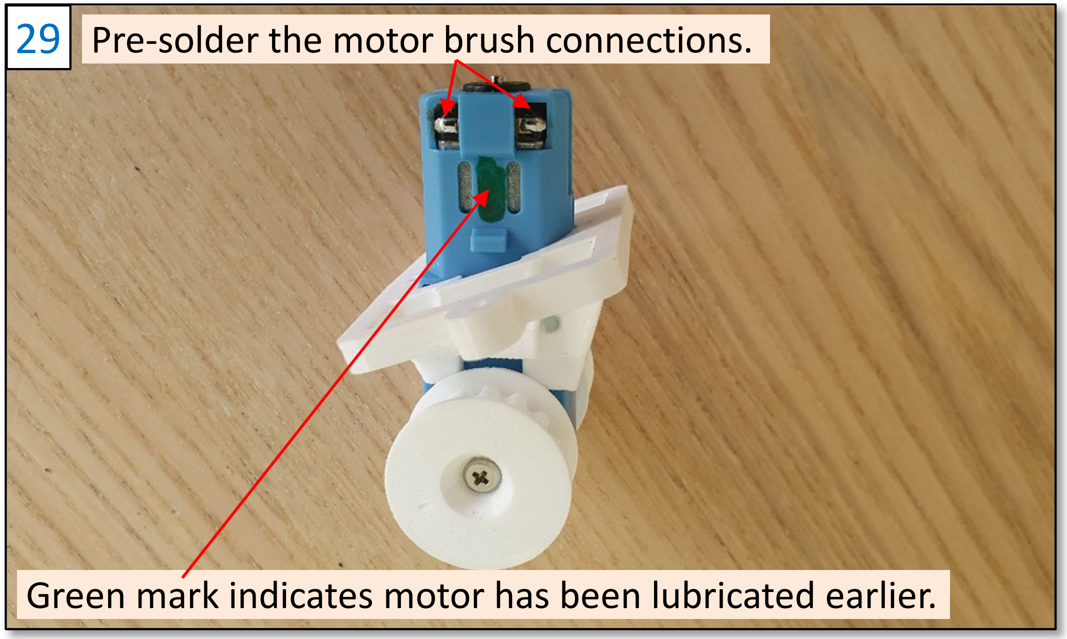
Avoid glue getting on contact areas.



28 Pre-wire the four motor drive plugs.

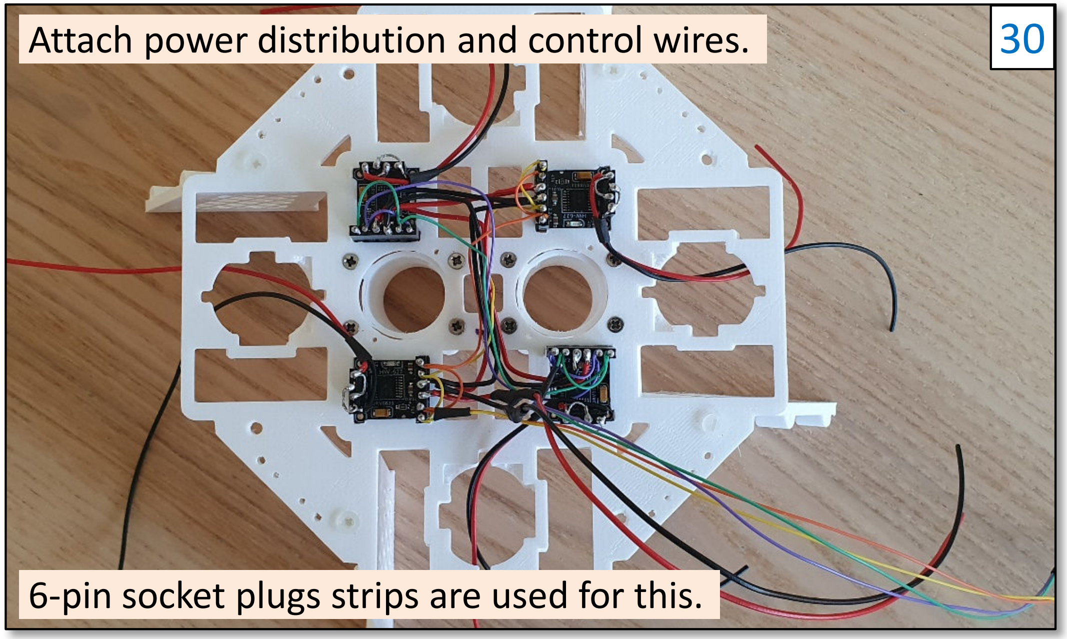
Ensure sufficient wire to reach the motors.

Build Images



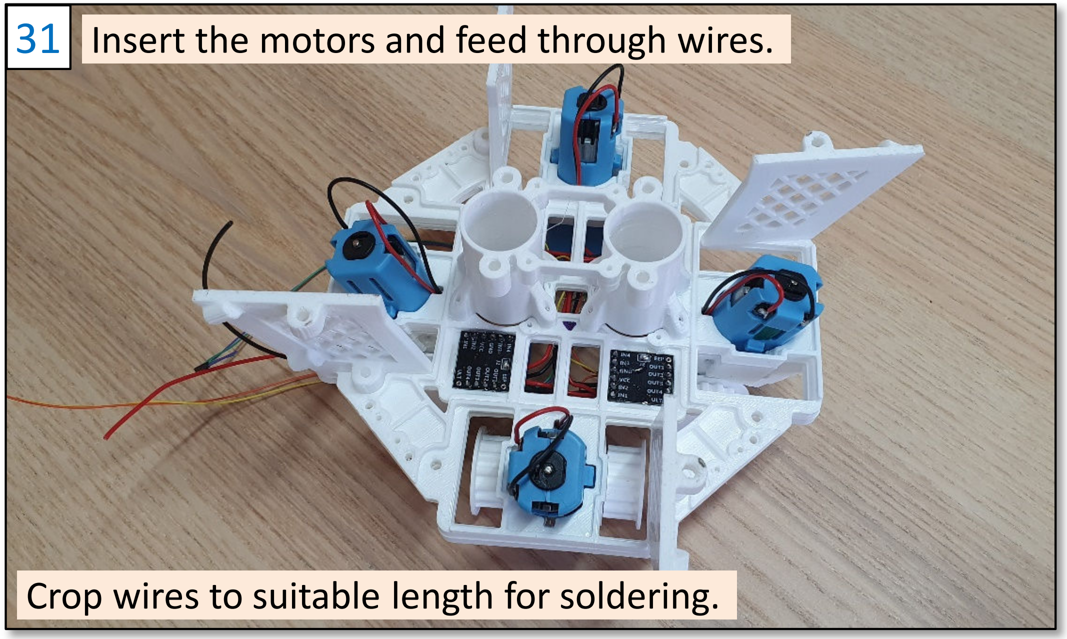
29 Pre-solder the motor brush connections.

Green mark indicates motor has been lubricated earlier.



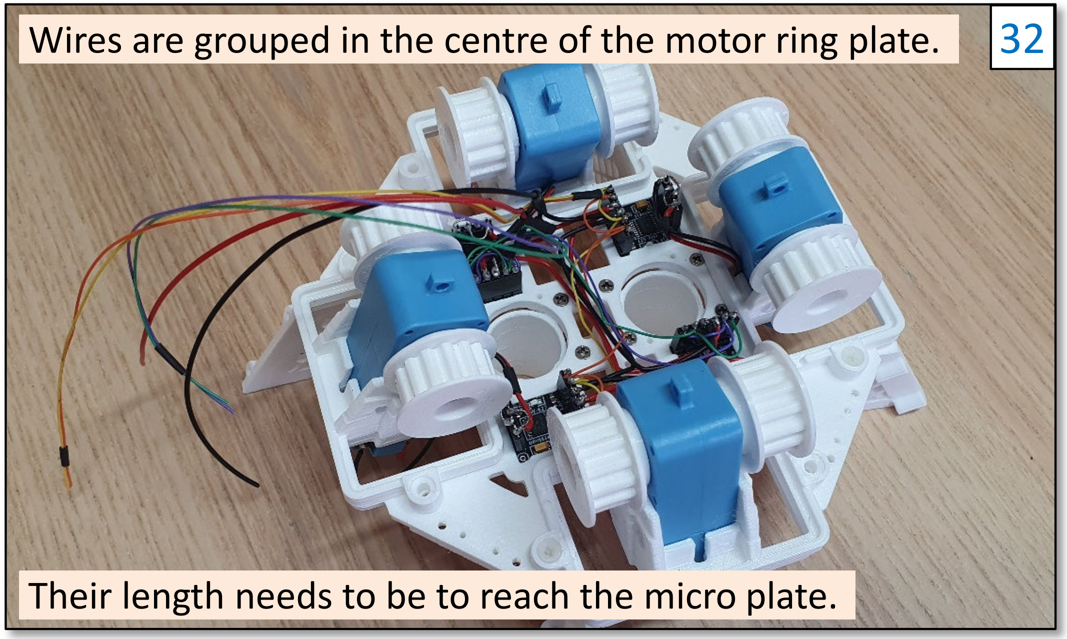
30 Attach power distribution and control wires.

6-pin socket plugs strips are used for this.



31 Insert the motors and feed through wires.

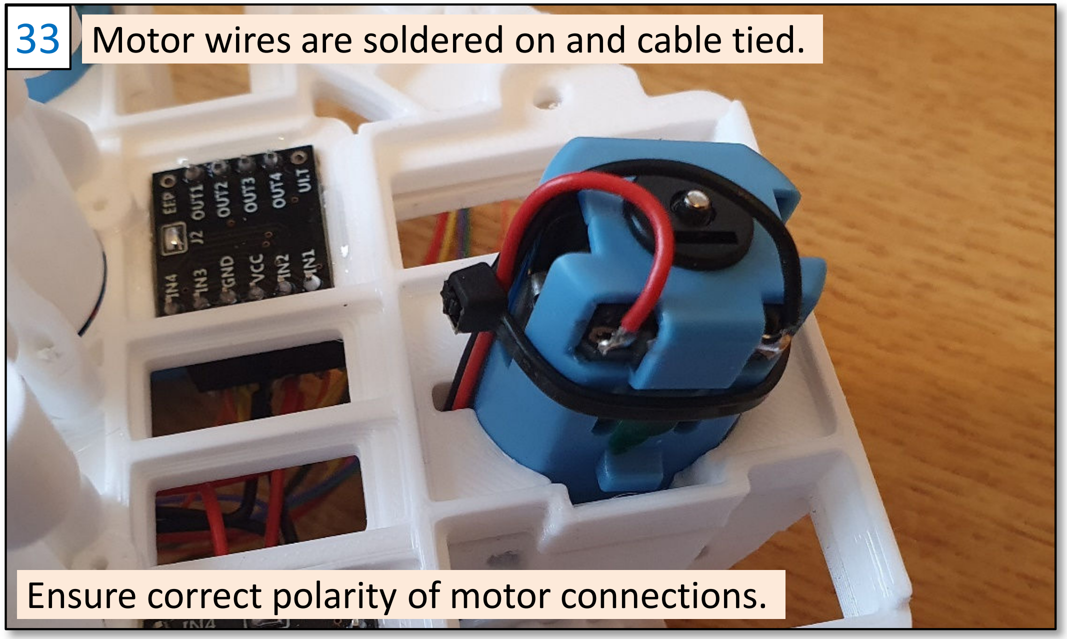
Crop wires to suitable length for soldering.



32 Wires are grouped in the centre of the motor ring plate.

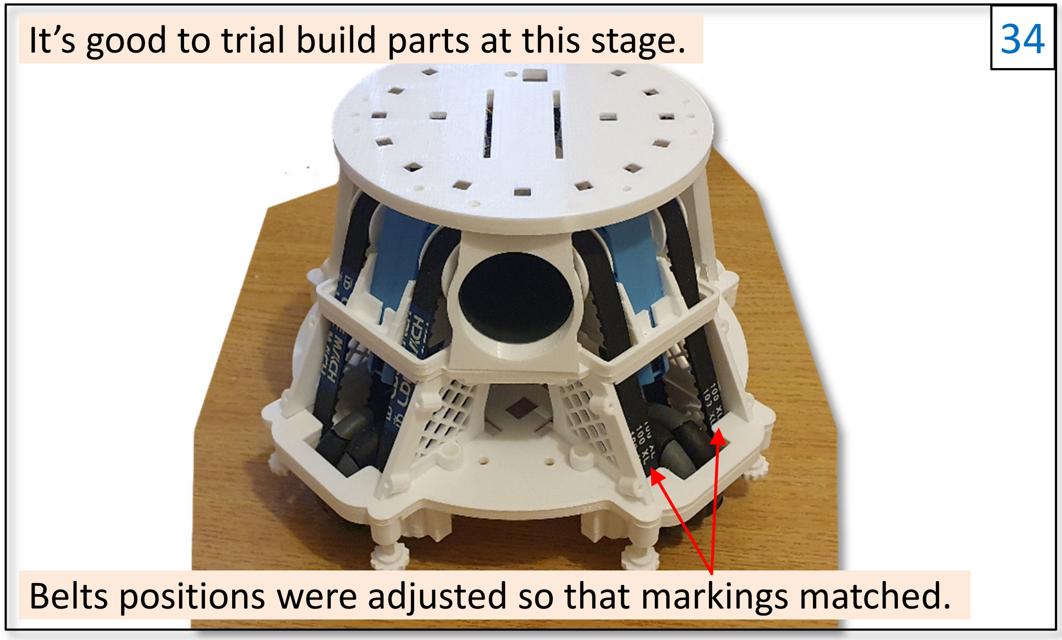
Their length needs to be to reach the micro plate.

Build Images



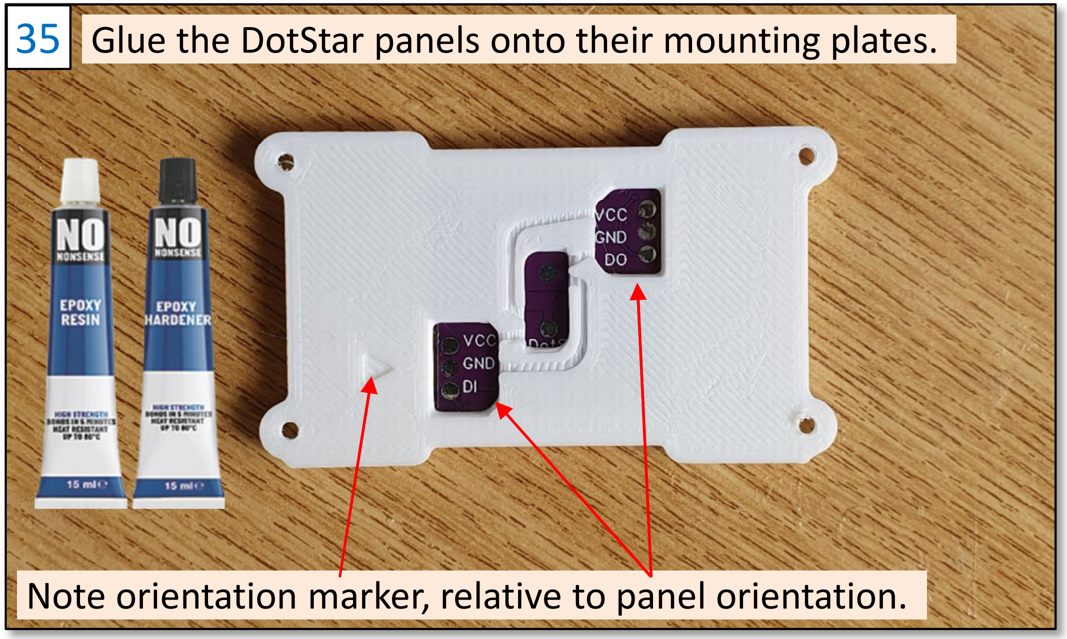
33 Motor wires are soldered on and cable tied.

Ensure correct polarity of motor connections.



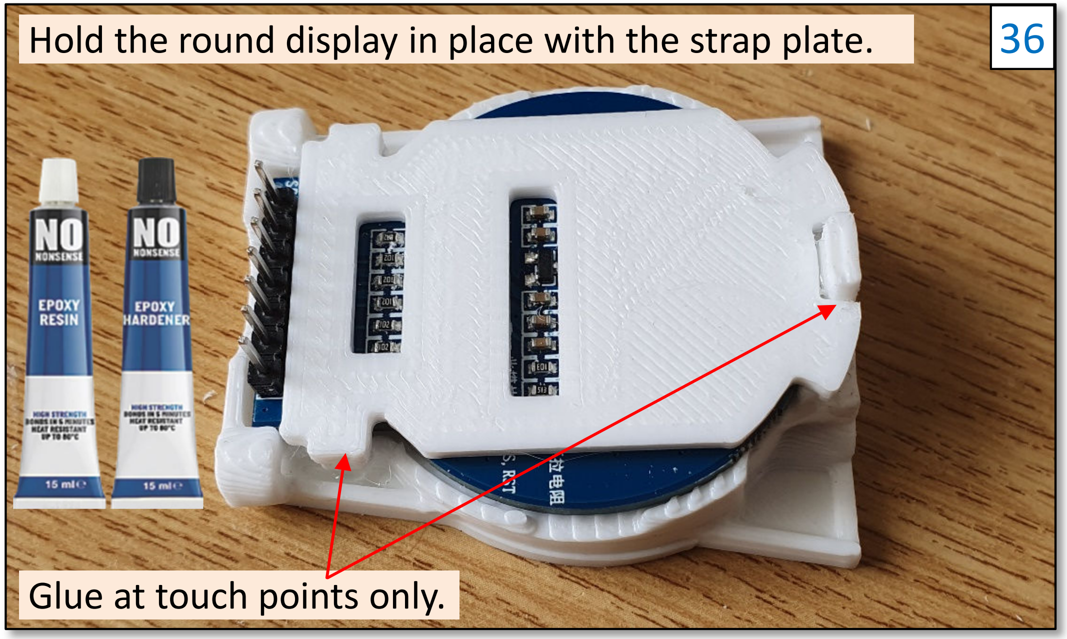
34 It's good to trial build parts at this stage.

Belts positions were adjusted so that markings matched.



35 Glue the DotStar panels onto their mounting plates.

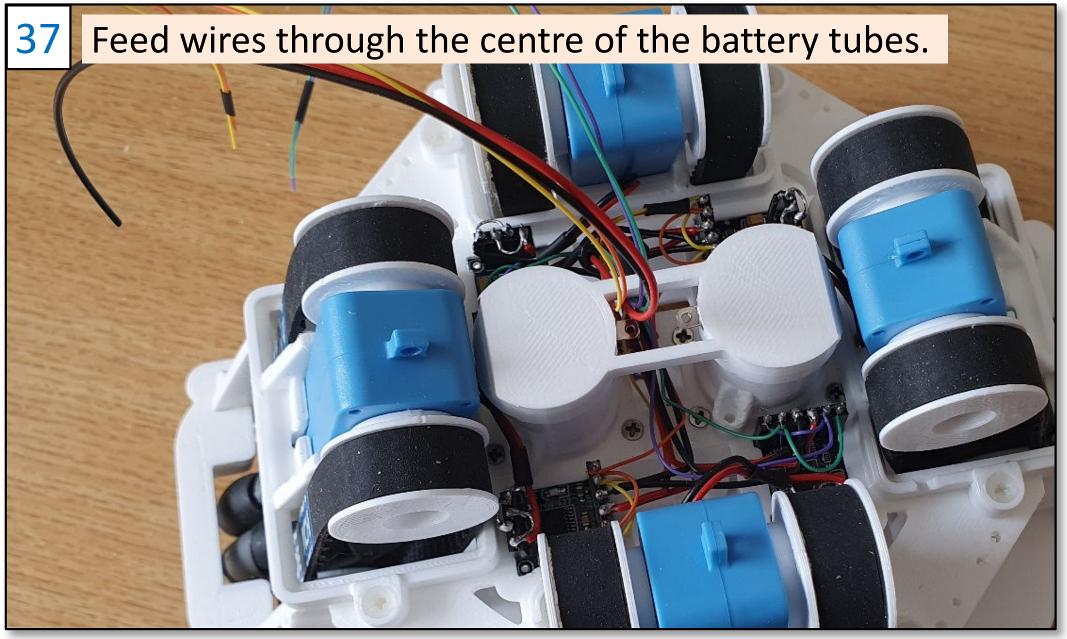
Note orientation marker, relative to panel orientation.



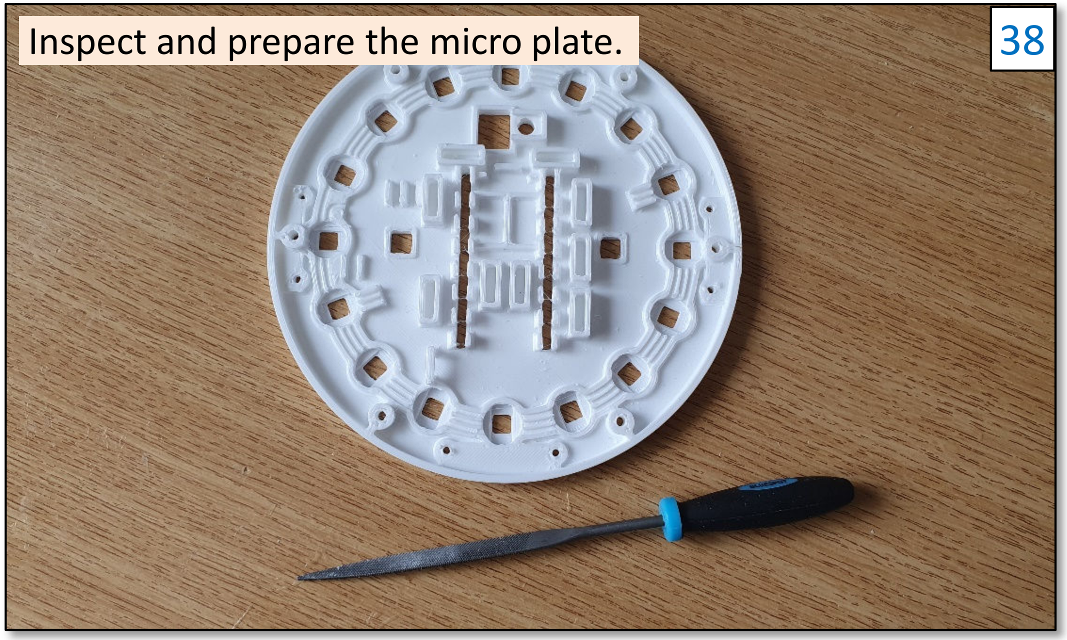
36 Hold the round display in place with the strap plate.

Glue at touch points only.

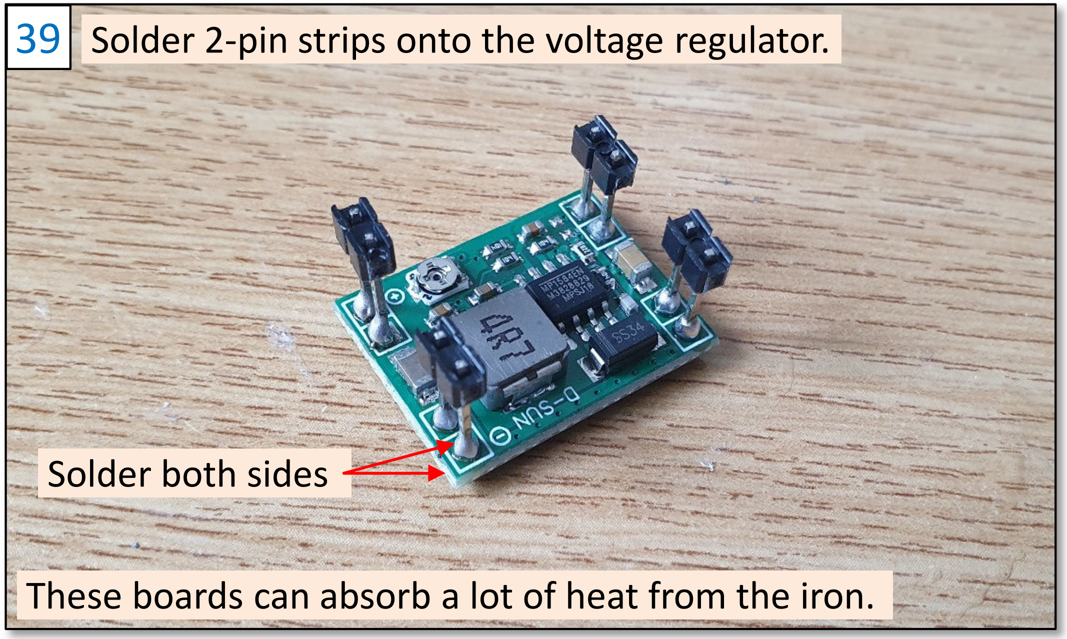
Build Images



37 Feed wires through the centre of the battery tubes.

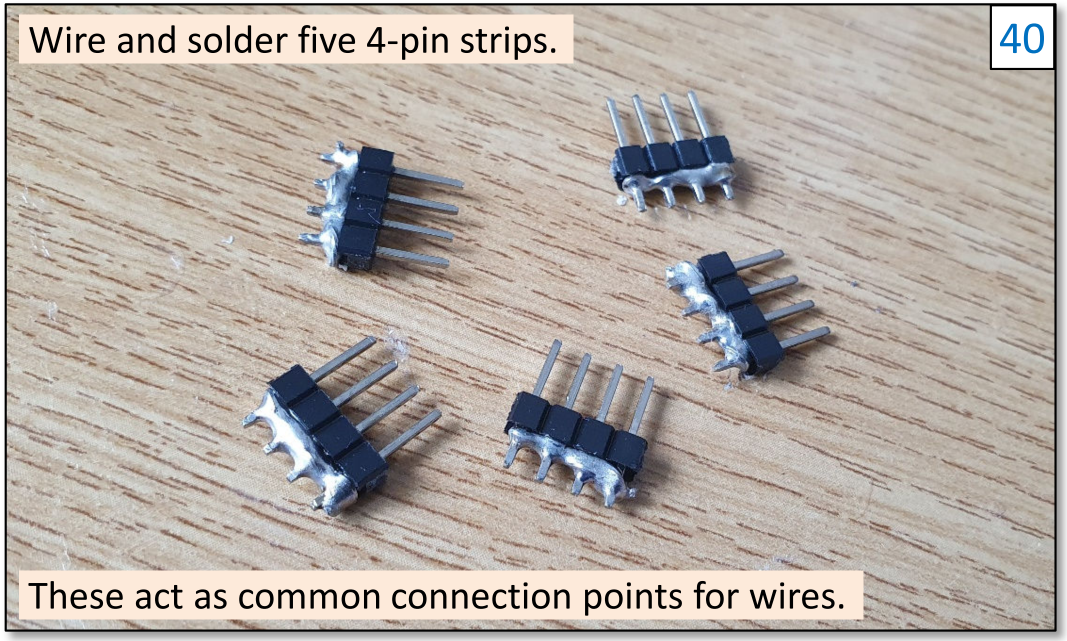


38 Inspect and prepare the micro plate.



39 Solder 2-pin strips onto the voltage regulator.

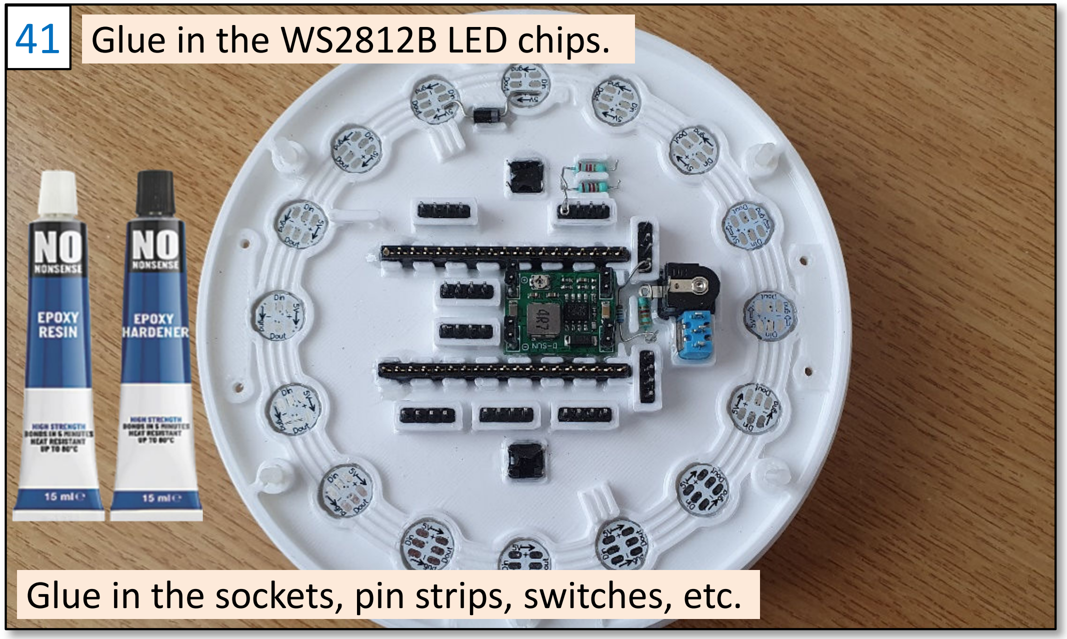
These boards can absorb a lot of heat from the iron.



40 Wire and solder five 4-pin strips.

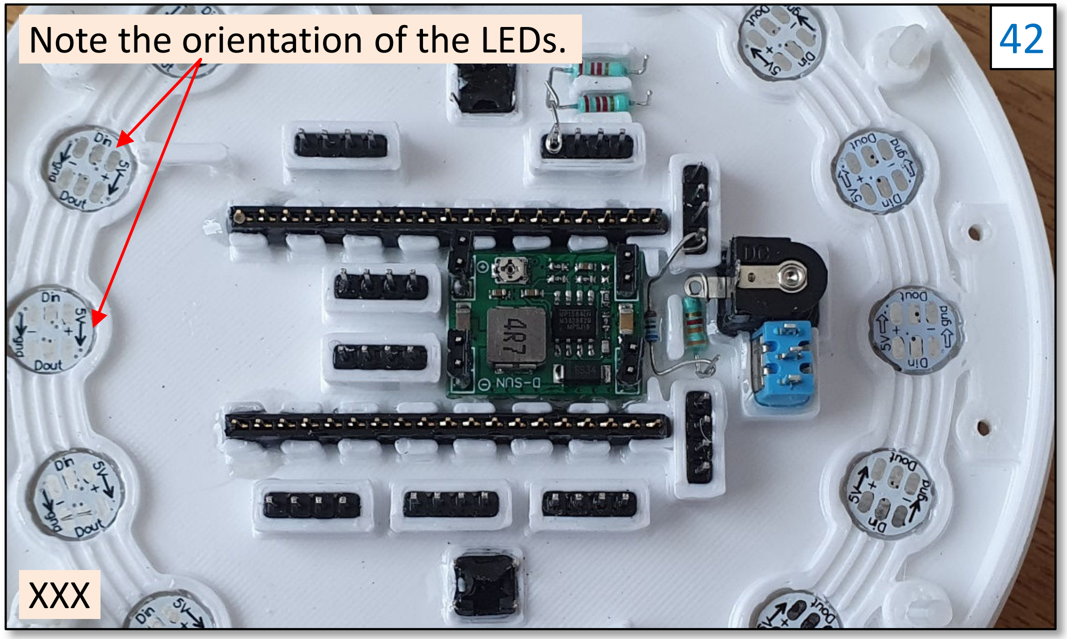
These act as common connection points for wires.

Build Images



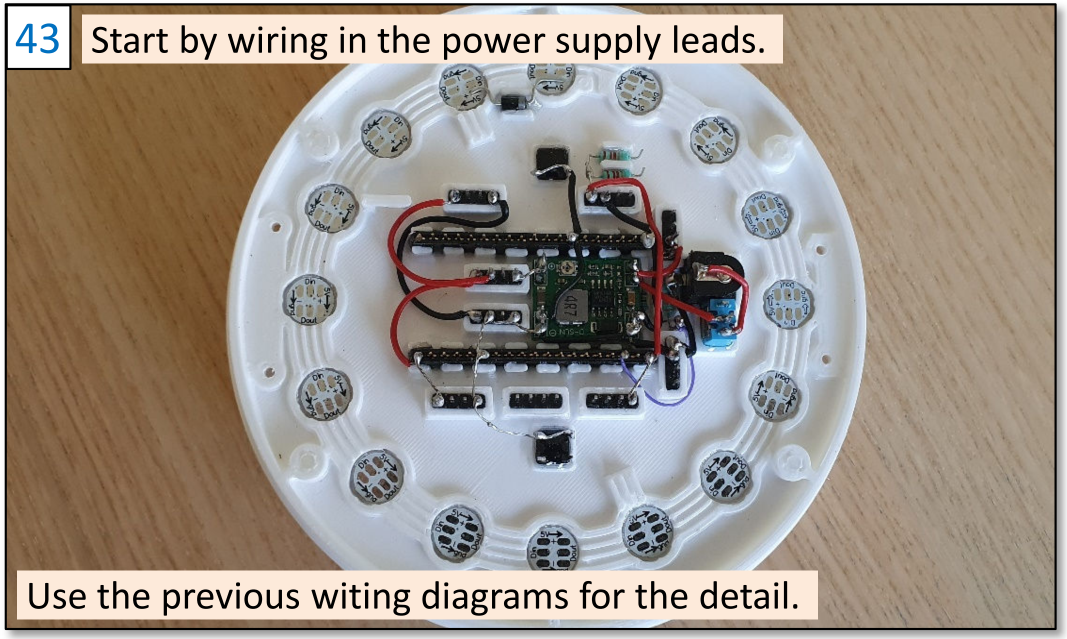
41 Glue in the WS2812B LED chips.

Glue in the sockets, pin strips, switches, etc.



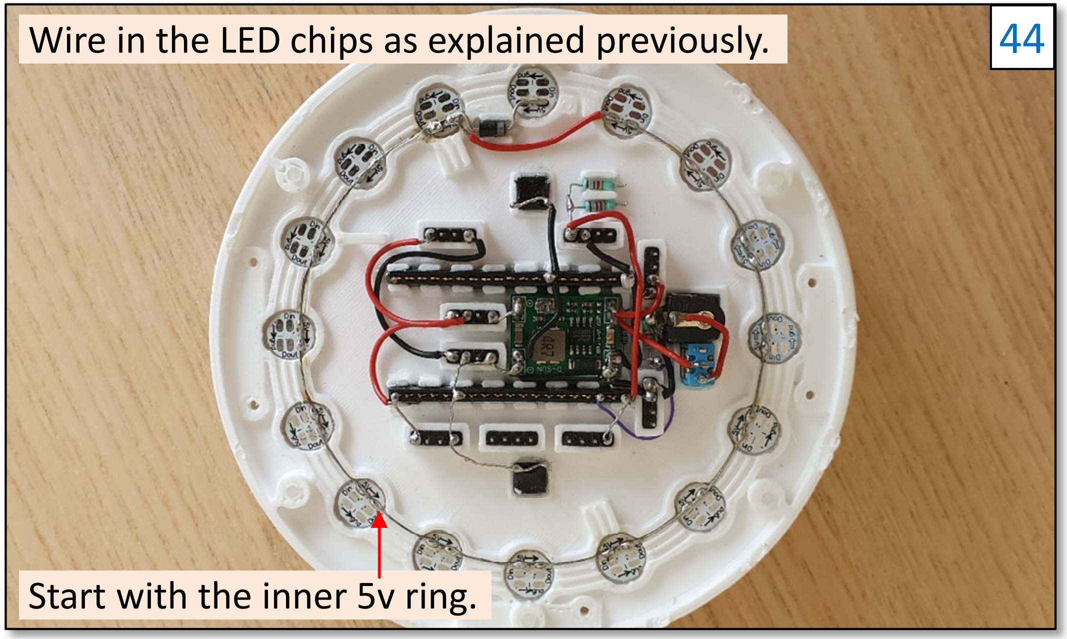
42 Note the orientation of the LEDs.

XXX



43 Start by wiring in the power supply leads.

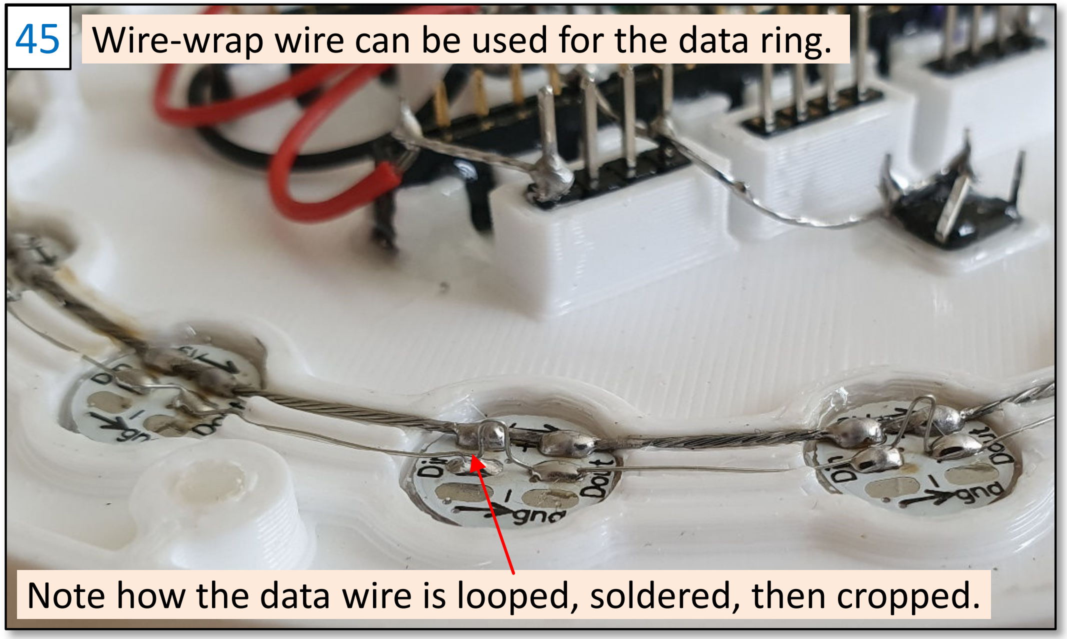
Use the previous witing diagrams for the detail.

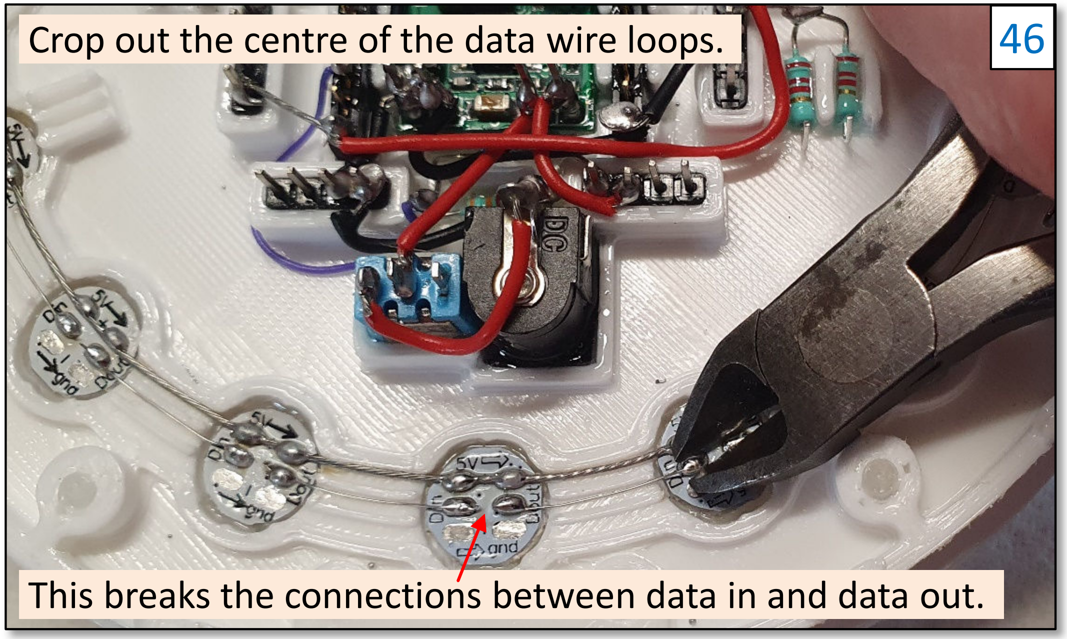


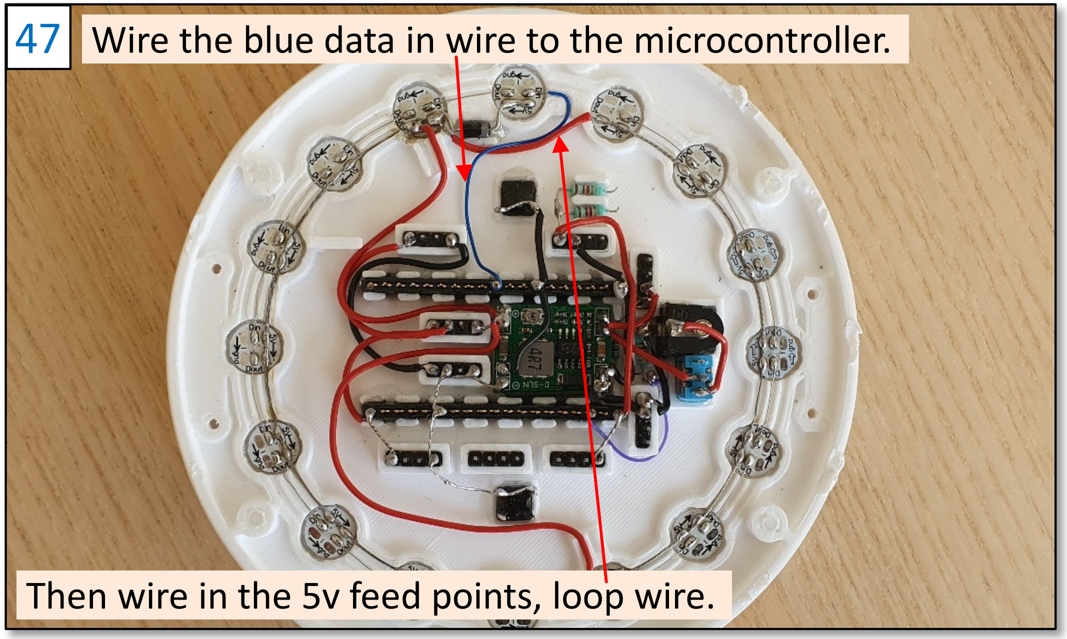
44 Wire in the LED chips as explained previously.

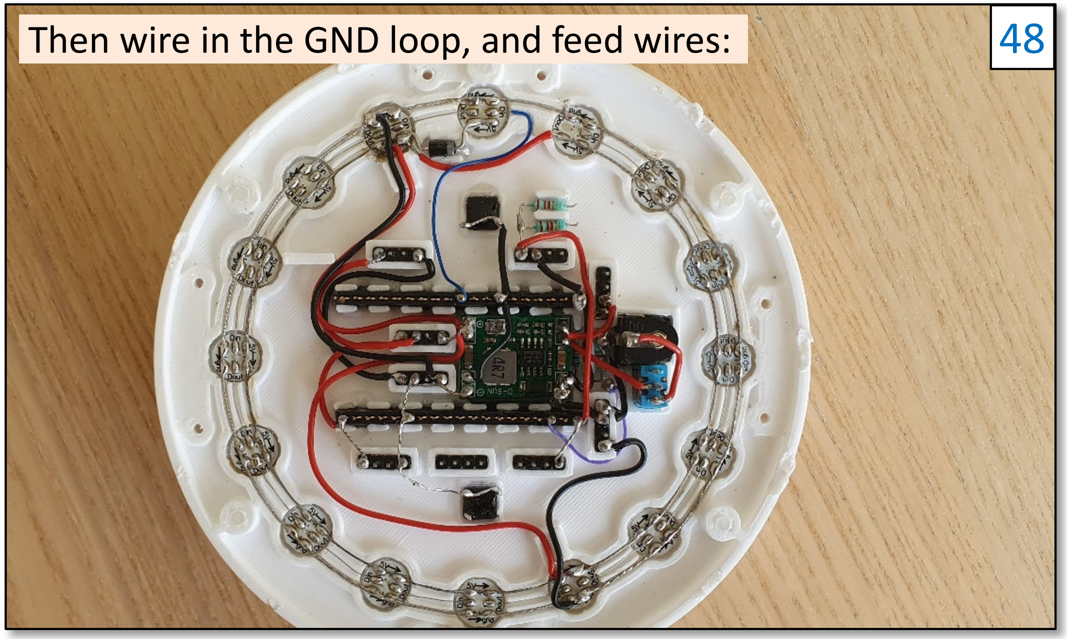
Start with the inner 5v ring.

Build Images

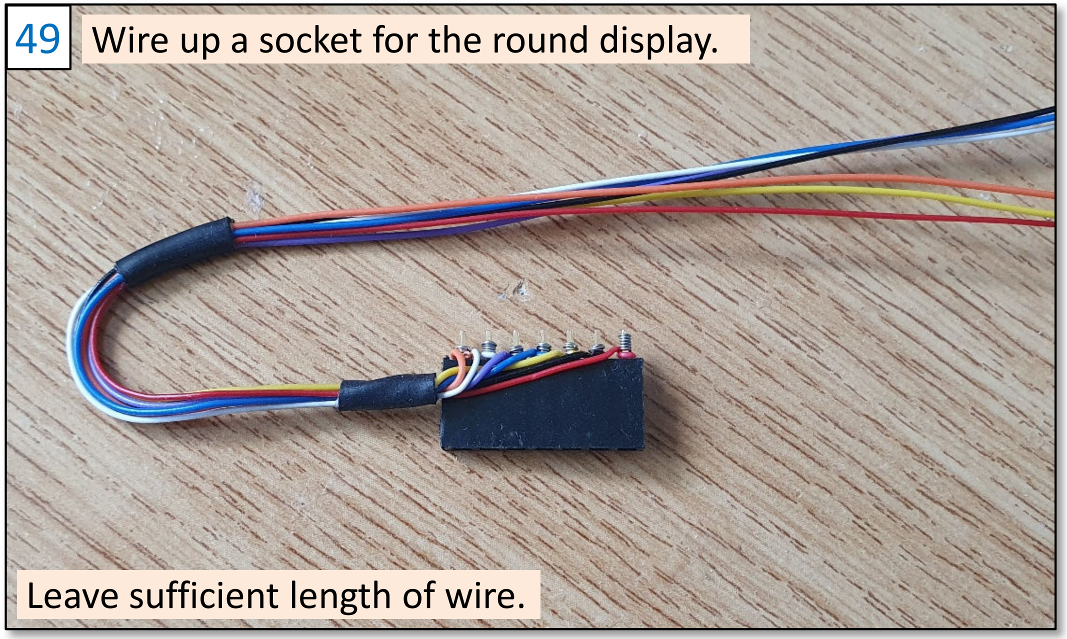




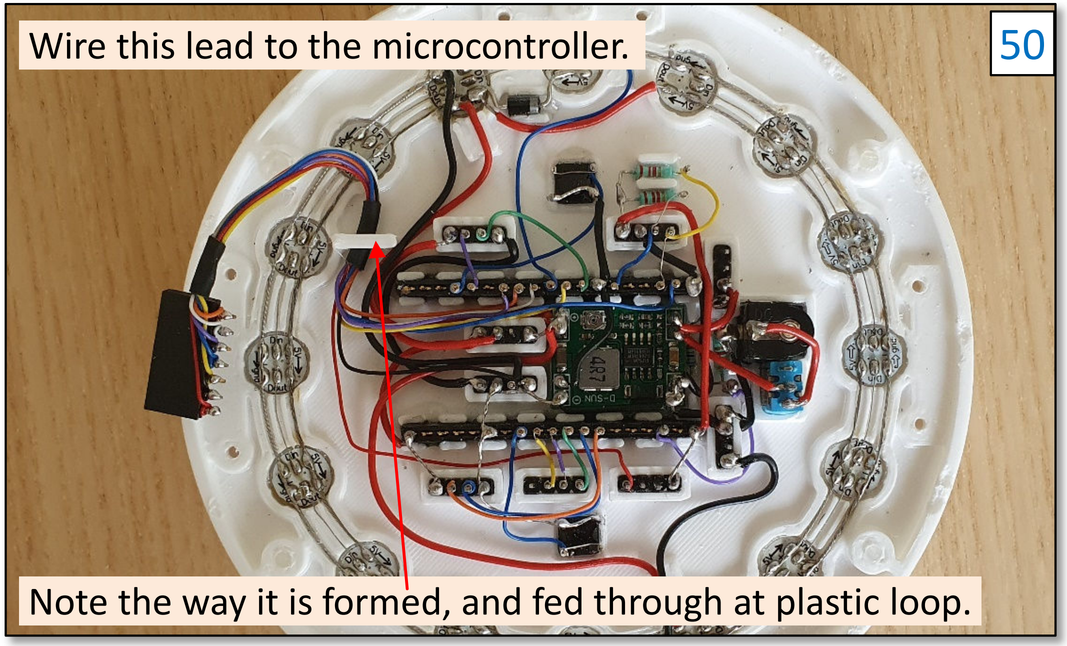




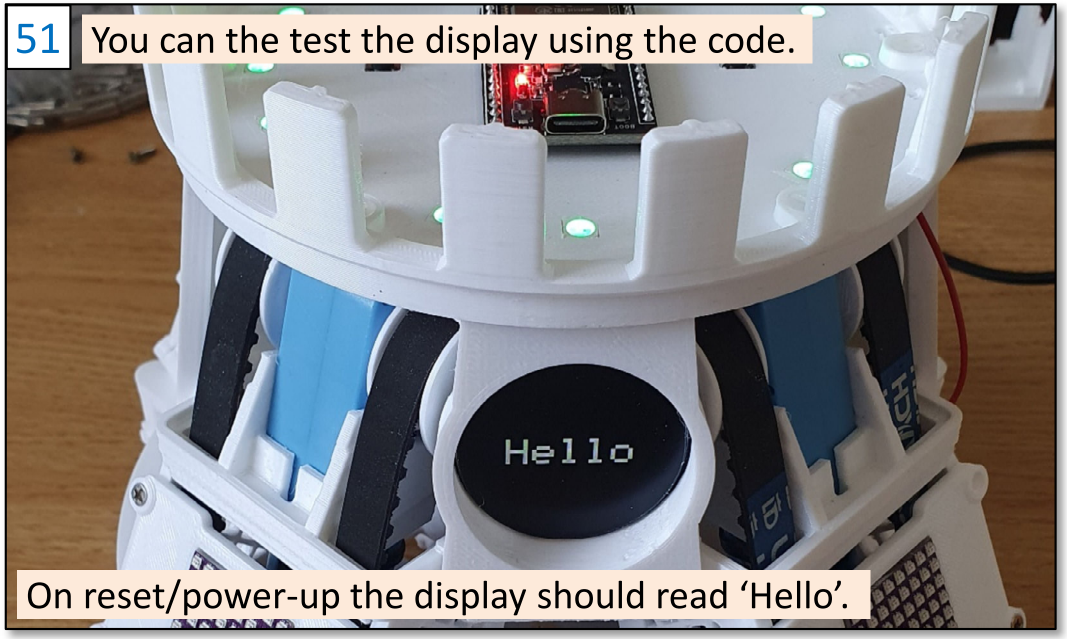
Build Images


49 Wire up a socket for the round display.

Leave sufficient length of wire.

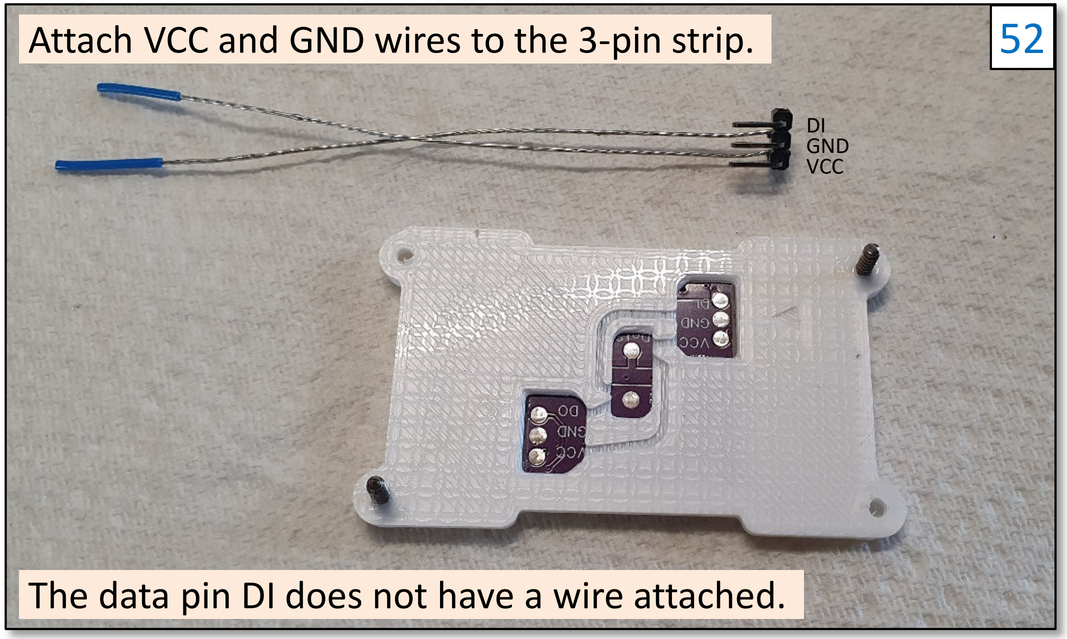

50 Wire this lead to the microcontroller.

Note the way it is formed, and fed through at plastic loop.



51 You can the test the display using the code.

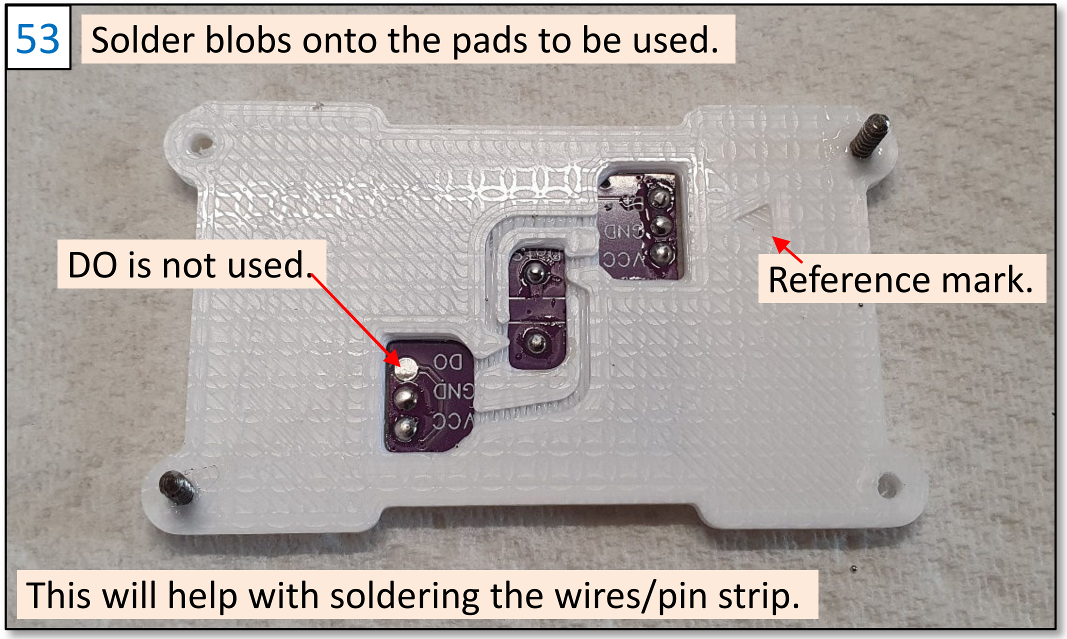
On reset/power-up the display should read ‘Hello’.



52 Attach VCC and GND wires to the 3-pin strip.

The data pin DI does not have a wire attached.

Build Images

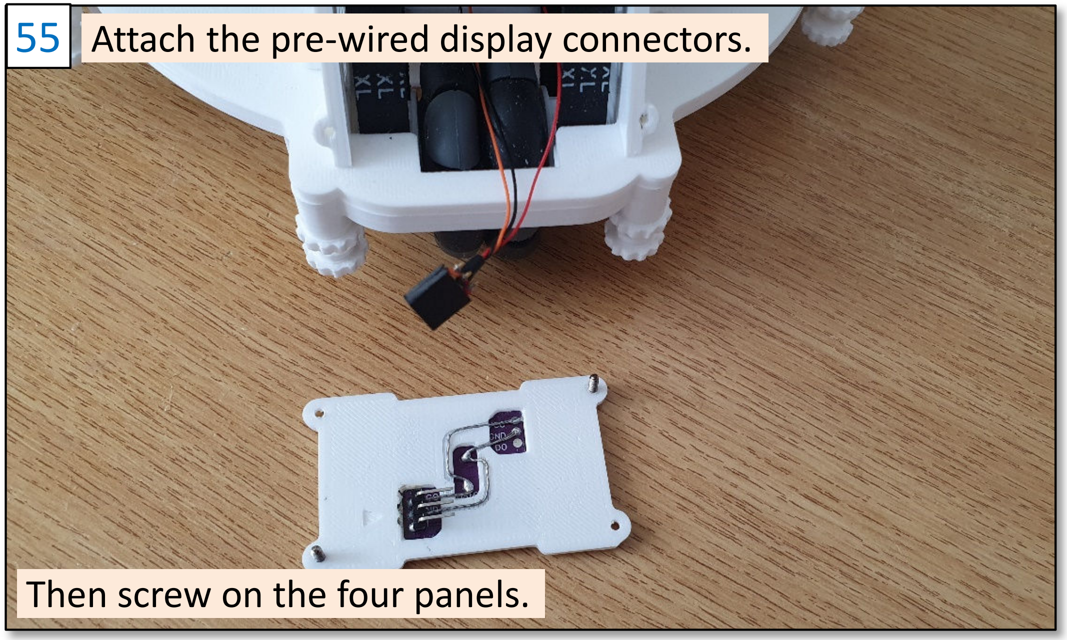


53 Solder blobs onto the pads to be used.

This will help with soldering the wires/pin strip.

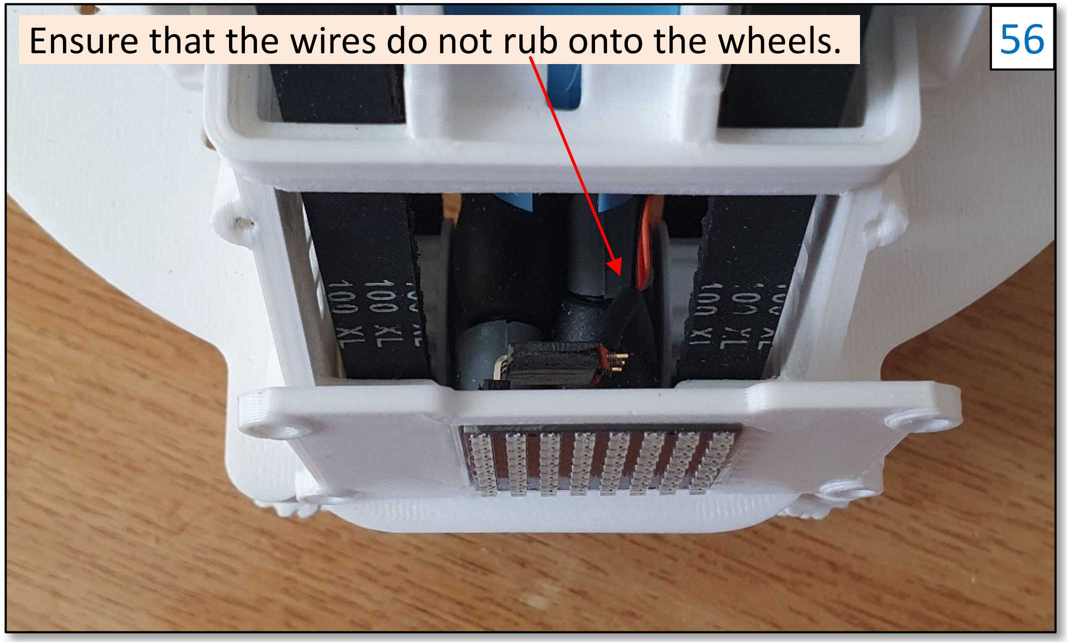
54 Then form and solder the wires onto the pads.

Flux, brushed onto the solder blobs will help.



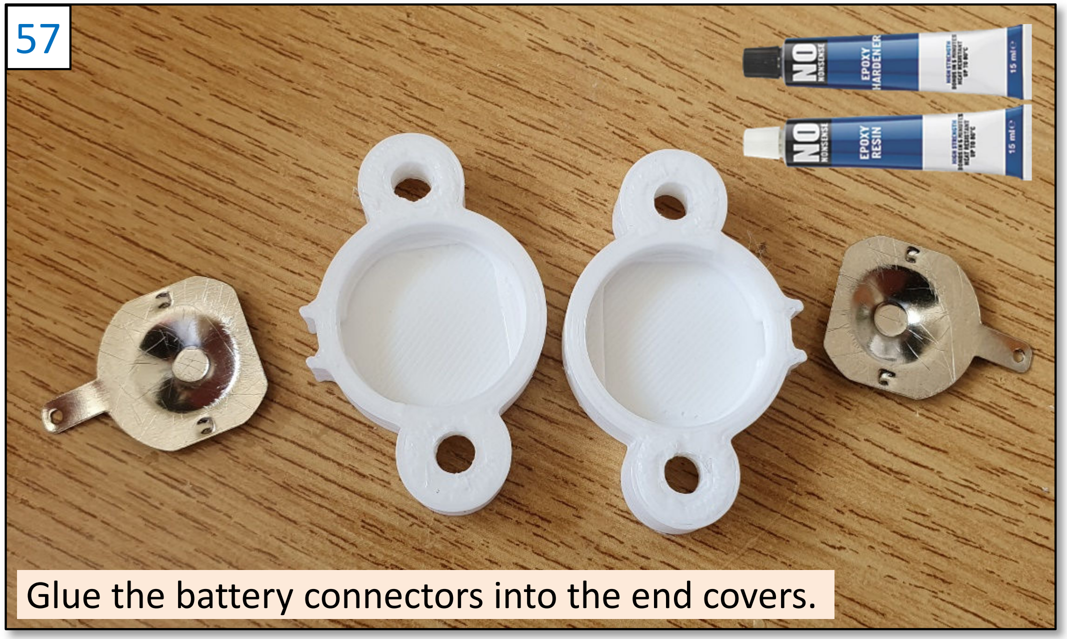
55 Attach the pre-wired display connectors.

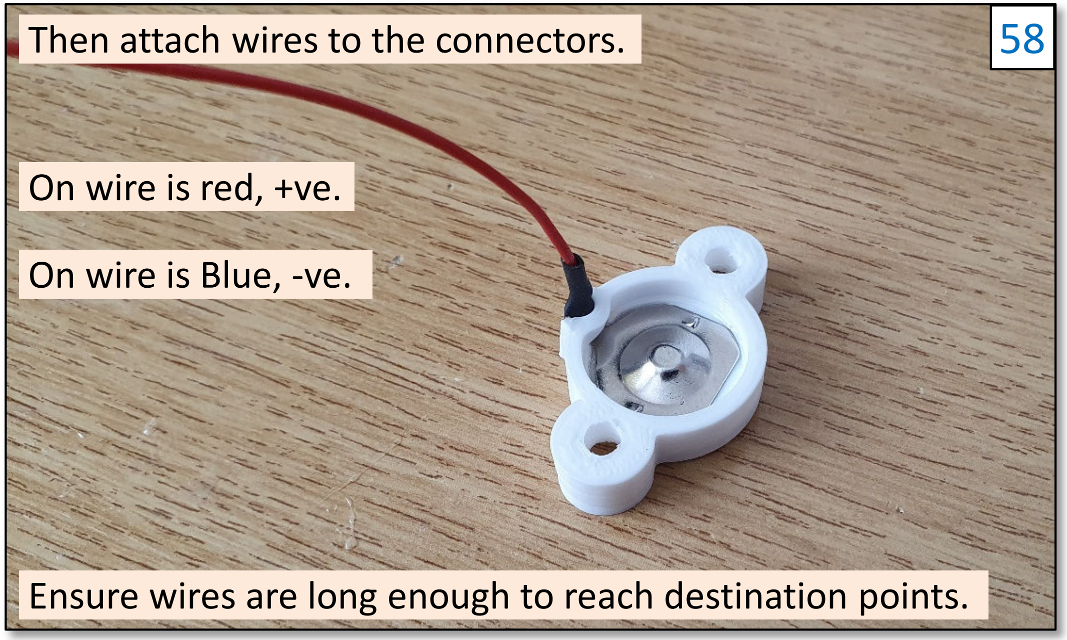
Then screw on the four panels.

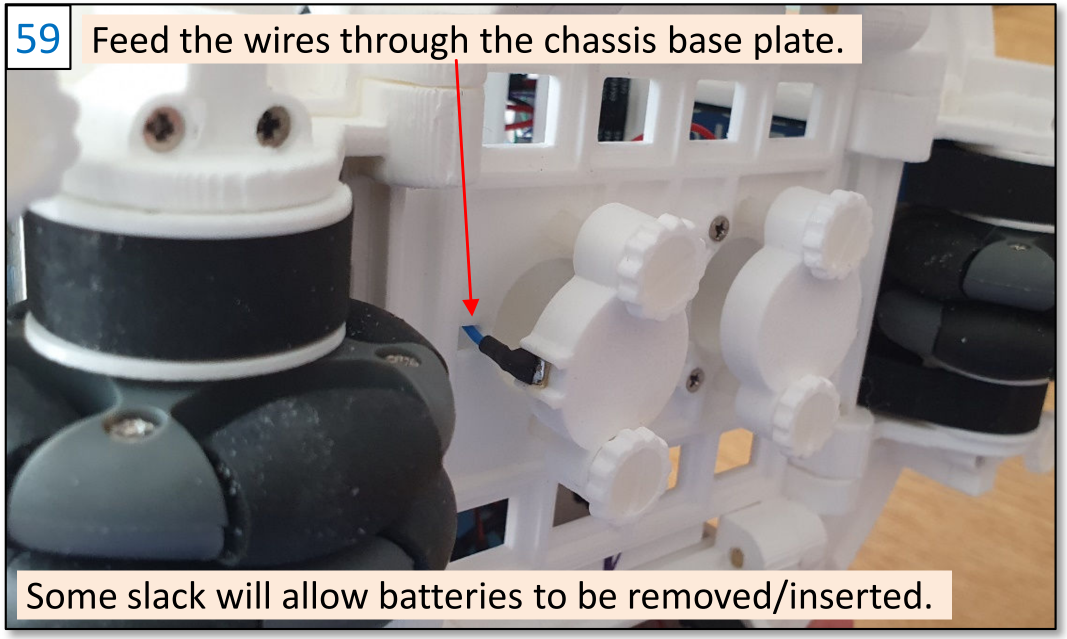


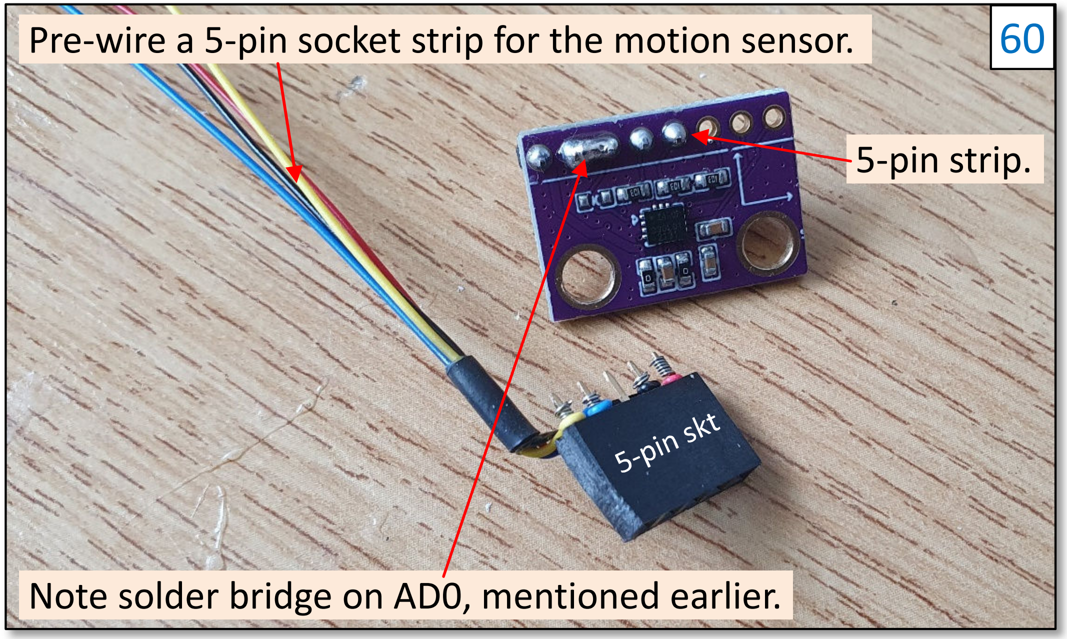
56 Ensure that the wires do not rub onto the wheels.

Build Images

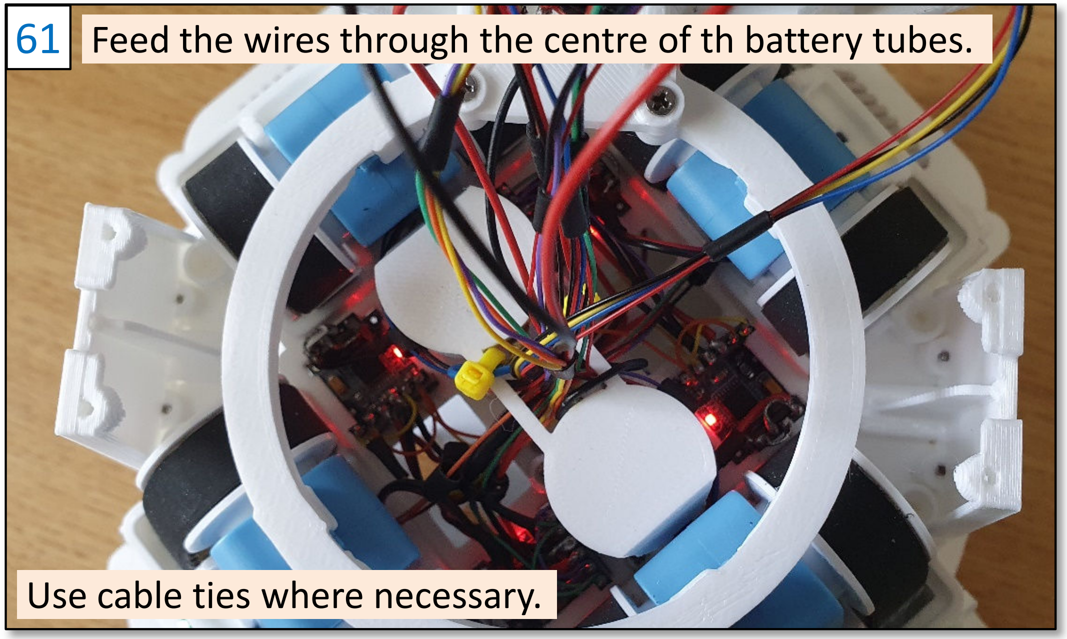






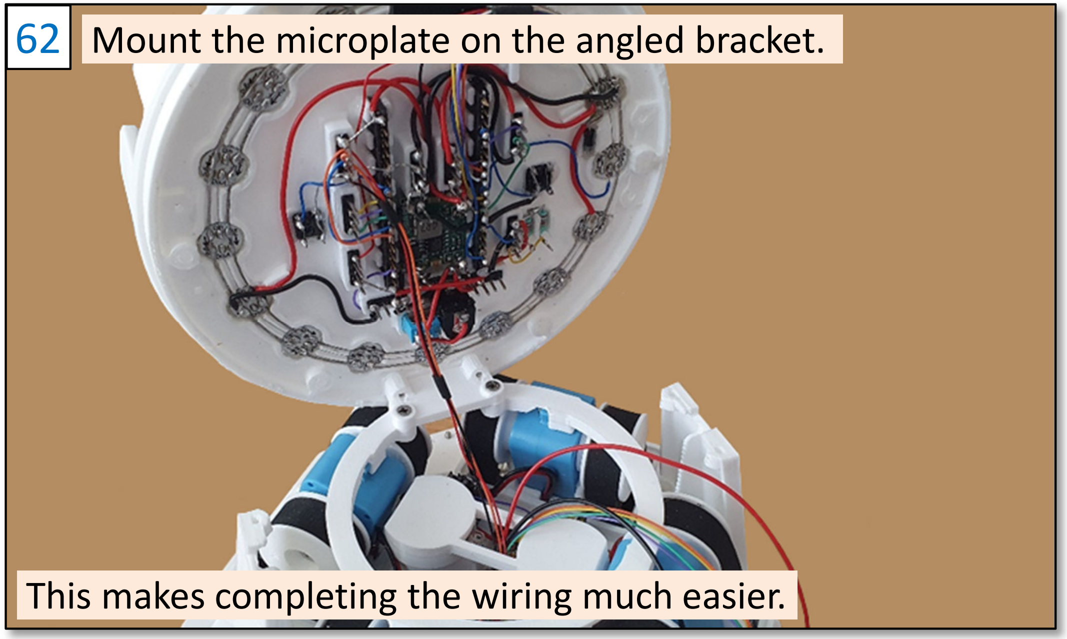


Build Images



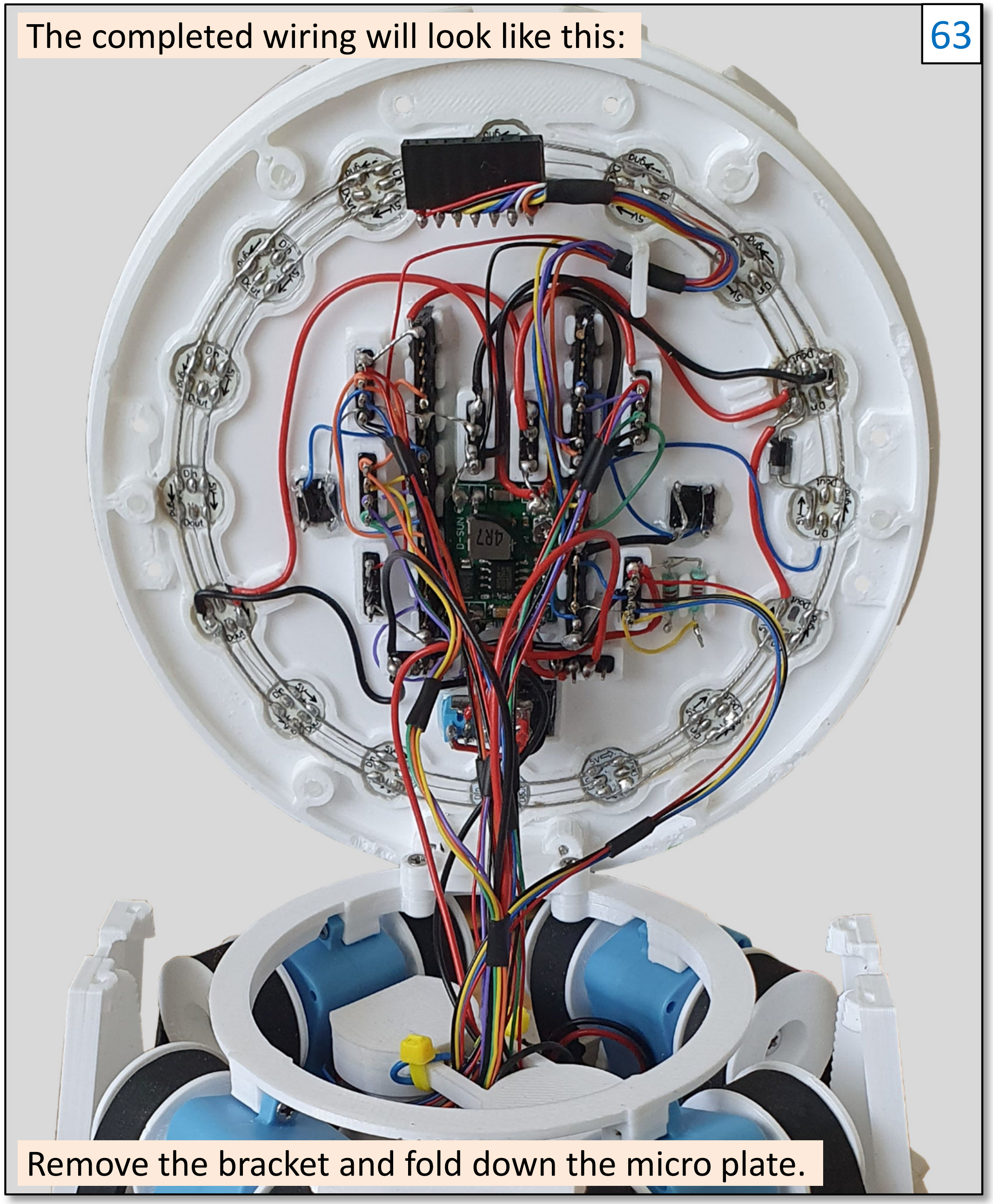
61 Feed the wires through the centre of th battery tubes.

Use cable ties where necessary.



62 Mount the microplate on the angled bracket.

This makes completing the wiring much easier.



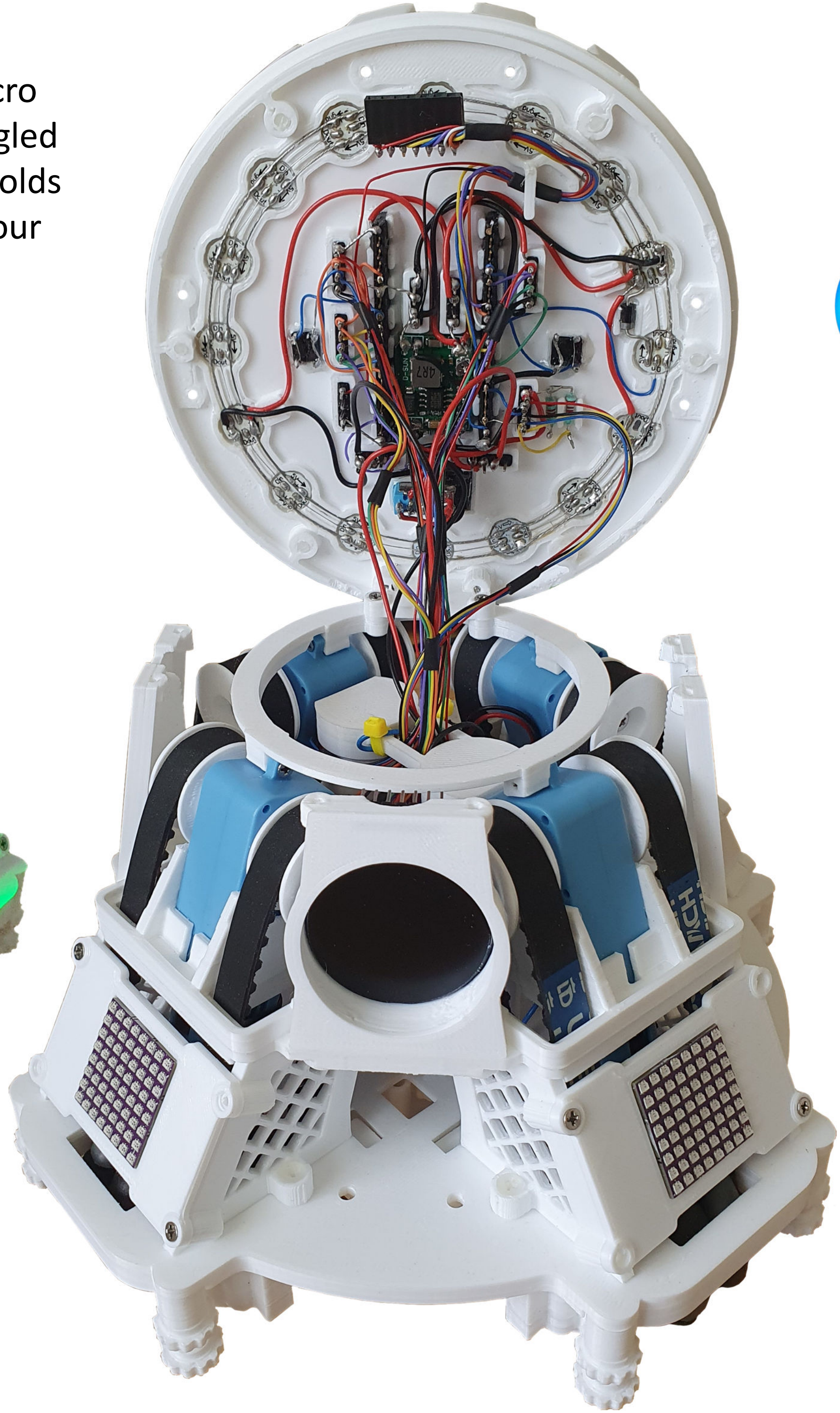
63 The completed wiring will look like this:

Remove the bracket and fold down the micro plate.

Body Wiring – Completed

This image shows the wiring to the micro plate; with it mounted off the right-angled support bracket. The microplate then folds down the wiring and attached to the four vertical plates.

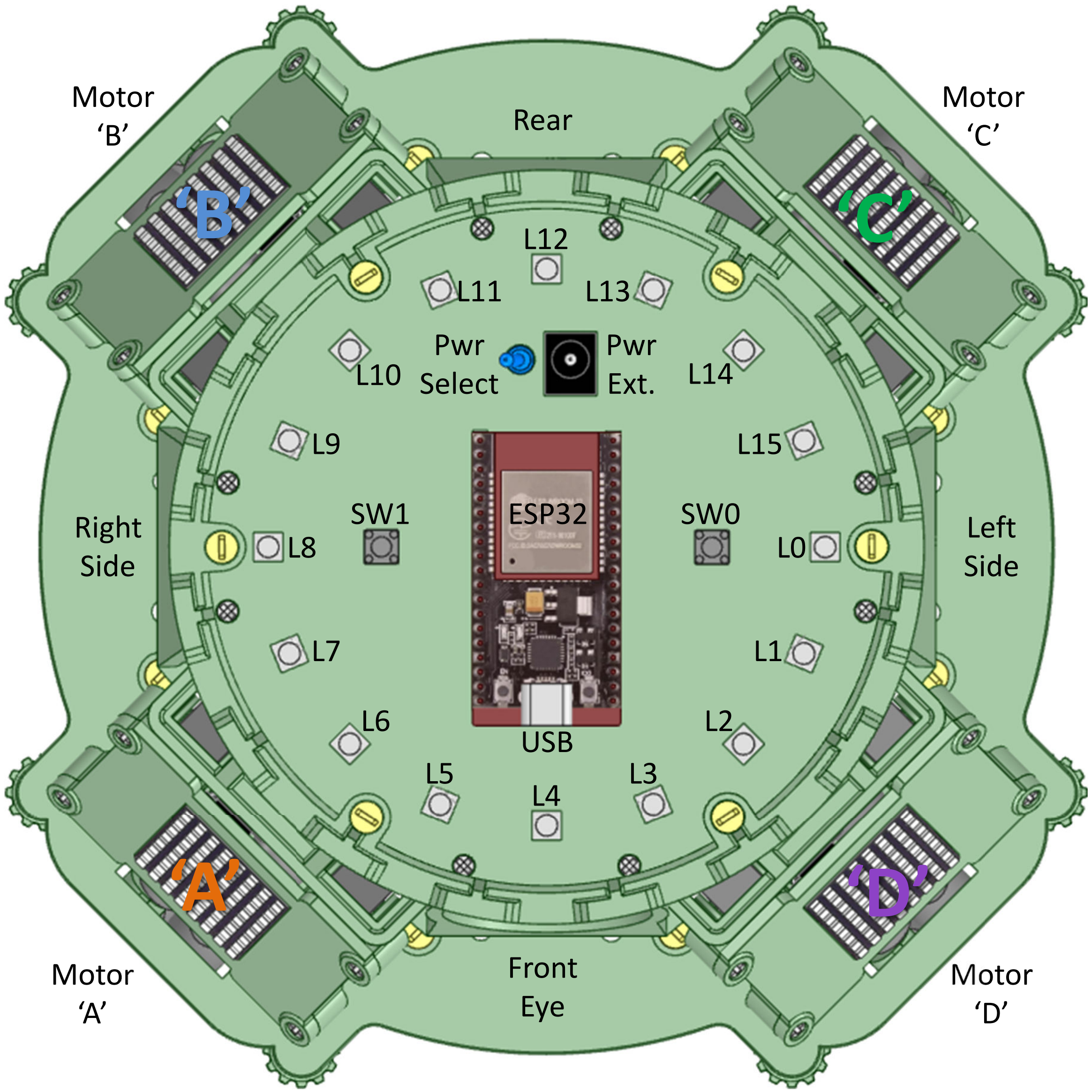
BallBot powered from an external supply.



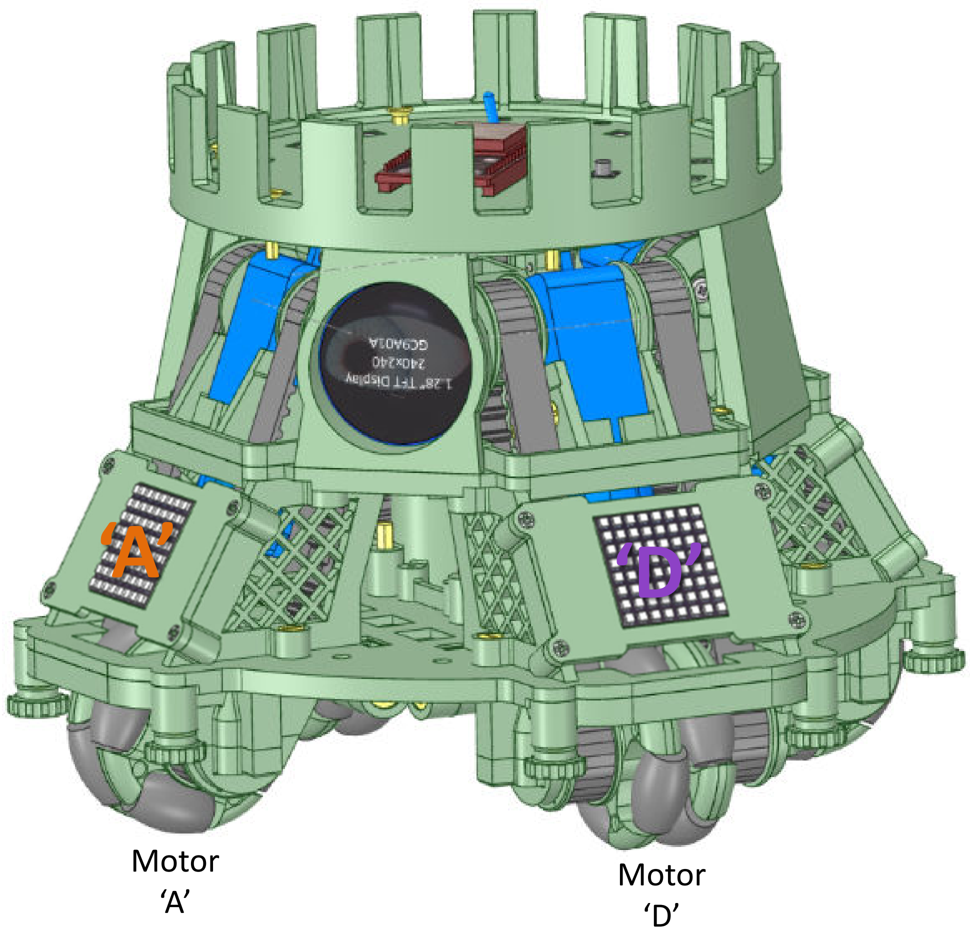
This project is designed to work with a WEMOS D1 transceiver

A Wii Classic controller, and the Monitor+ app for tuning the PID controllers

Component Reference



Top View



Front View

Motor Drive

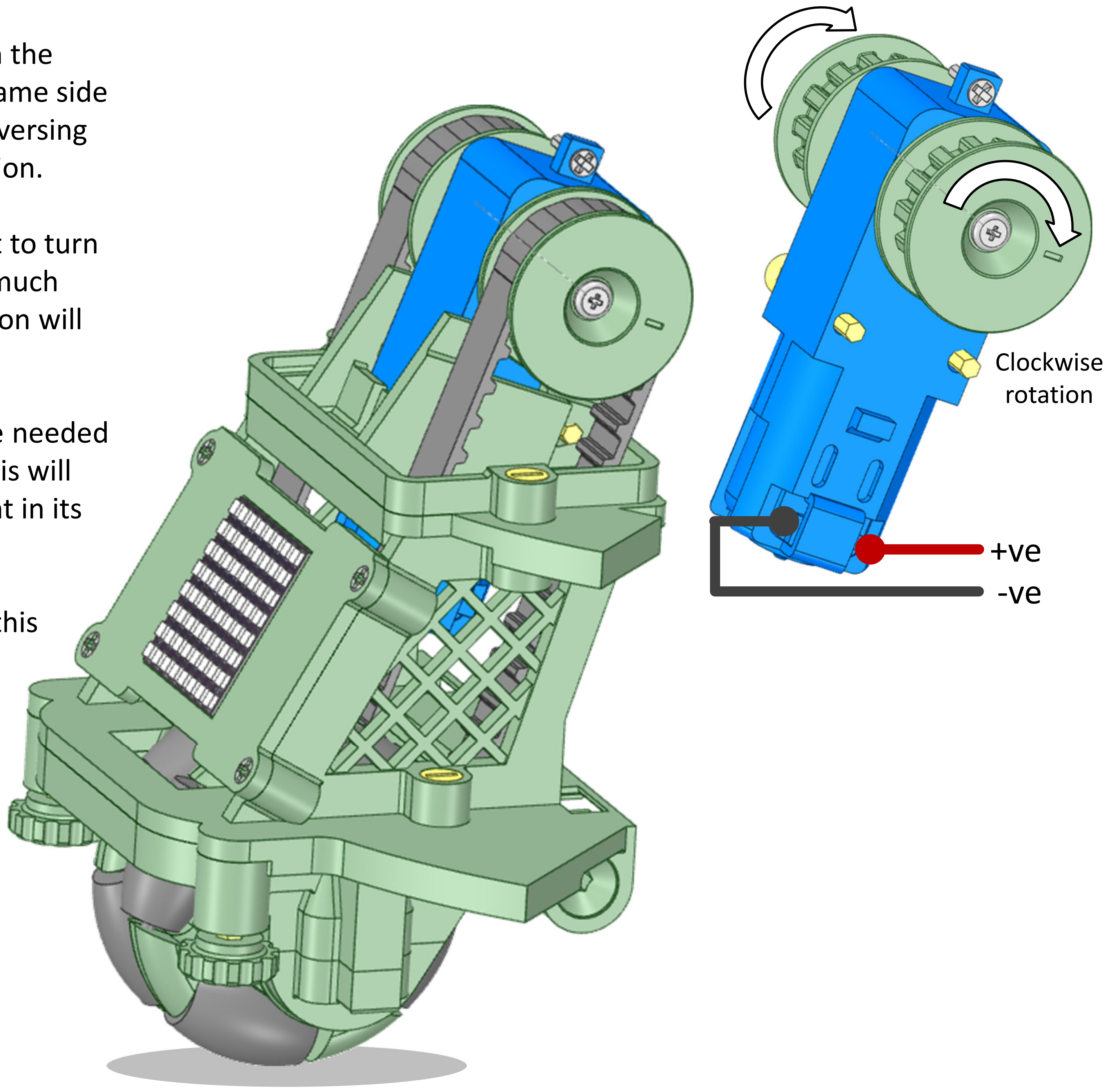
When a DC voltage is applied to a motor, with the polarity as shown here, the sprocket on the same side as the motor connections turns clockwise. Reversing the polarity will reverse the direction of rotation.

Whilst the motor is rated at 6v DC, it will start to turn at a much lower voltage, depending on how much friction there is in the drive system. Belt tension will play a significant role in this effect.

It is useful to determine the minimum voltage needed to initiate movement; as voltages less than this will simply cause the motor to draw excess current in its stalled state, which is wasteful.

The code monitors PWM demands to detect this condition and prevent it from persisting.

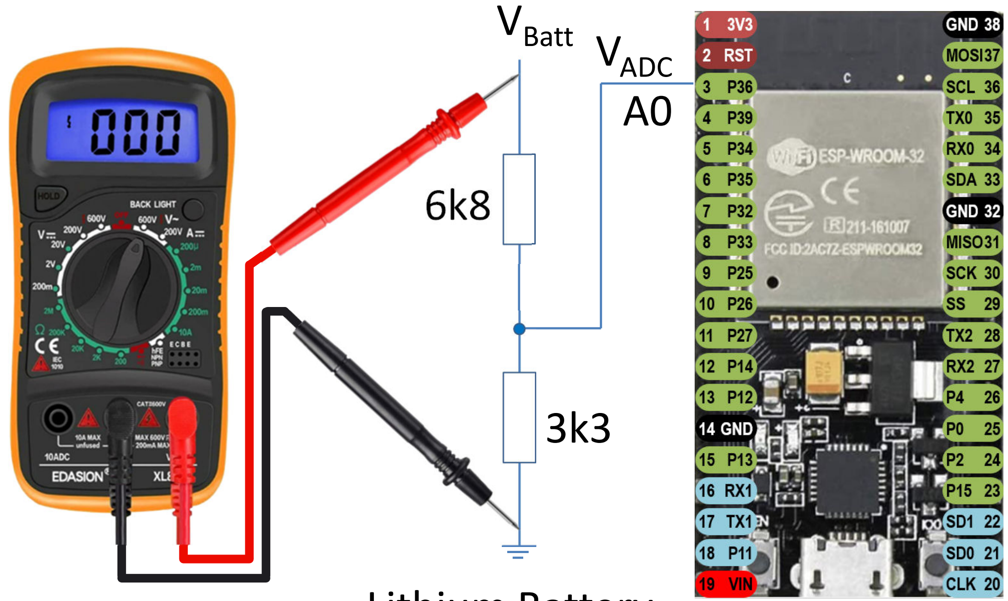
Opposite motor pairs are wired such that they drive in the same direction, even though they are reversed mechanically due to the way in which they are mounted. Motors have the own H-bridge drivers, but they share common PWM signals.



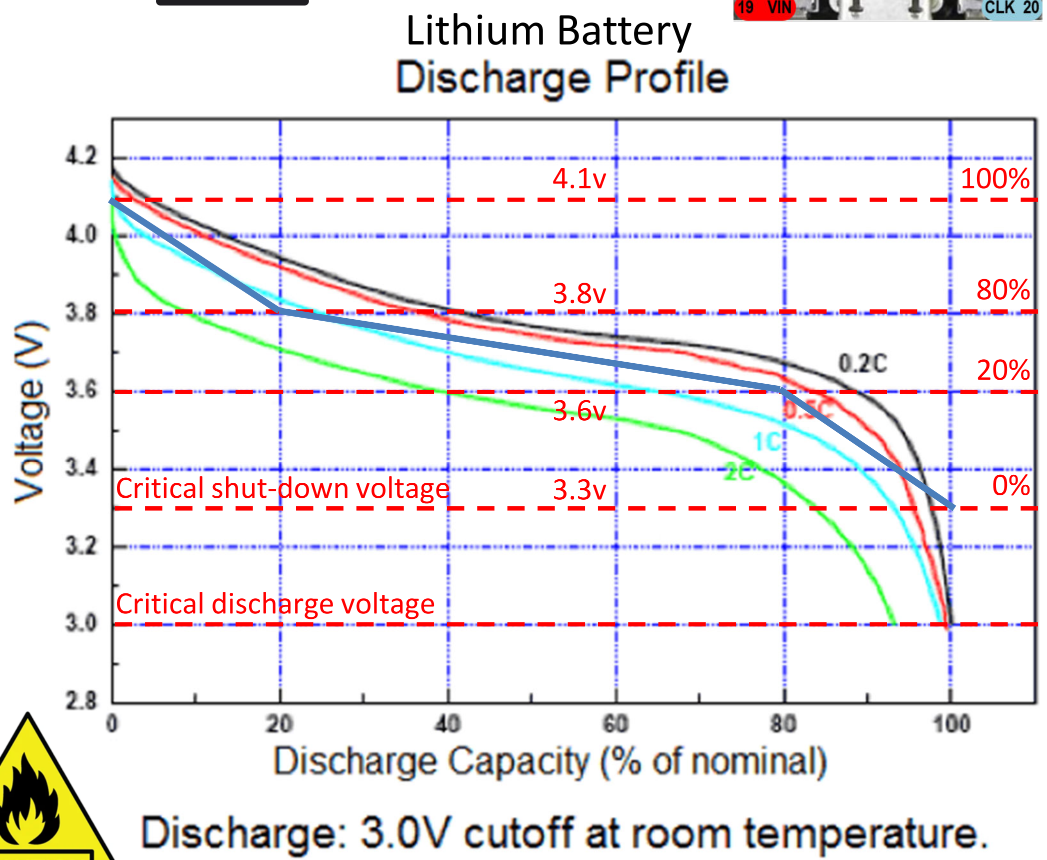
Battery Voltage Calibration

See Lithium discharge curve obtained from the internet. In this analysis the lipo battery consists of two identical batteries connected in series.
Assume fully charged 8.2v battery max voltage is V_{BM} >= 8.4v max (charging)
Set battery warning point at V_{BW} = 7.2v (2 x 3.6v)
Set battery critical point at V_{BC} = 6.6v (2 x3.3v)

The ESP32 is powered via a 5v voltage regulator, connected to the V_{in} pin, but the 6k8 supply sampling resistor is connected to source V_{Batt}.
For ESP32 V_{ADC} == 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 405.4 bit/v
Using a Multimeter and a variable DC supply, I determined the following V_{ADC} values for corresponding threshold voltages:



MAX. O.C	V_{OC} = 8.4v, gave A0 = 3363 On V_{ADC} (2 x 4.2v)
MAX: (100%)	V_M = 8.2v, gave A0 = 3260 on V_{ADC} (2 x 4.1v)
HIGH: (80%)	V_H = 7.8v, gave A0 = 3083 on V_{ADC} (2 x 3.9v)
WARNING: (20%)	V_{BW} = 7.2v, gives A0 = 2820 on V_{ADC} (2 x 3.6v)
CRITICAL: (0%)	V_{BC} = 6.6v, gives A0 = 2559 on V_{ADC} (2 x 3.3v)



The code will sample the battery voltage on power-up to ensure it is sufficient, then at every 40ms interval, calculating an average (1/50) to remove noise. It also detects no battery as USB mode.

In the code I have assumed a discharge curve ranging from 8.2v (100%) to 6.6v (0%) capacity, using the overlay lines shown. The rate of discharge is monitored and used to 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.