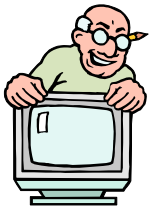
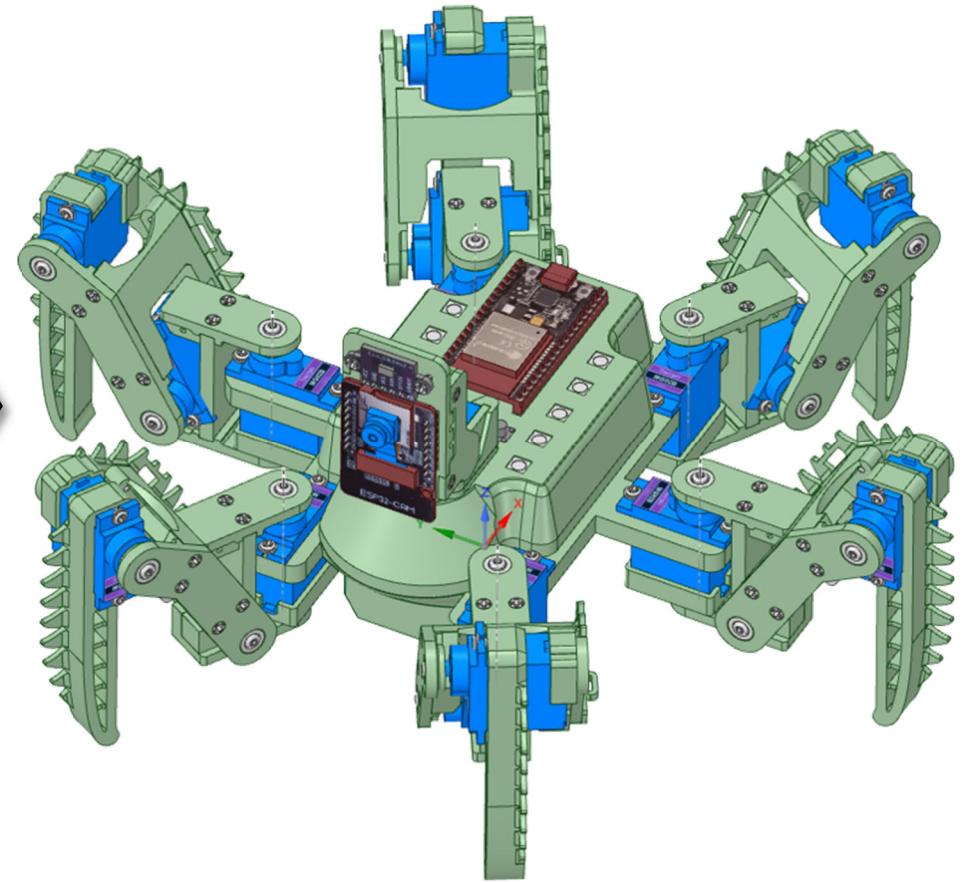
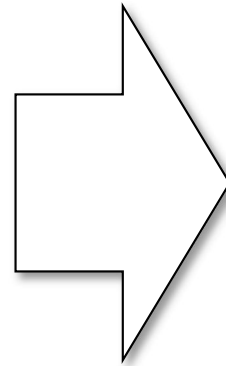
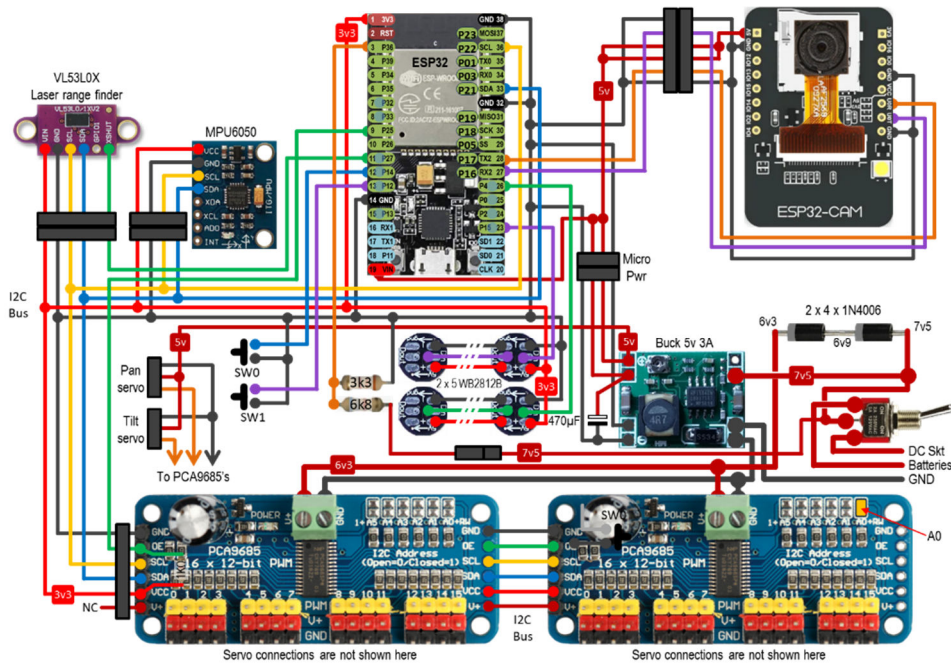


# Project SpidaBot

## Circuits & Wiring



Read through this documentation completely before attempting this project.

# 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.

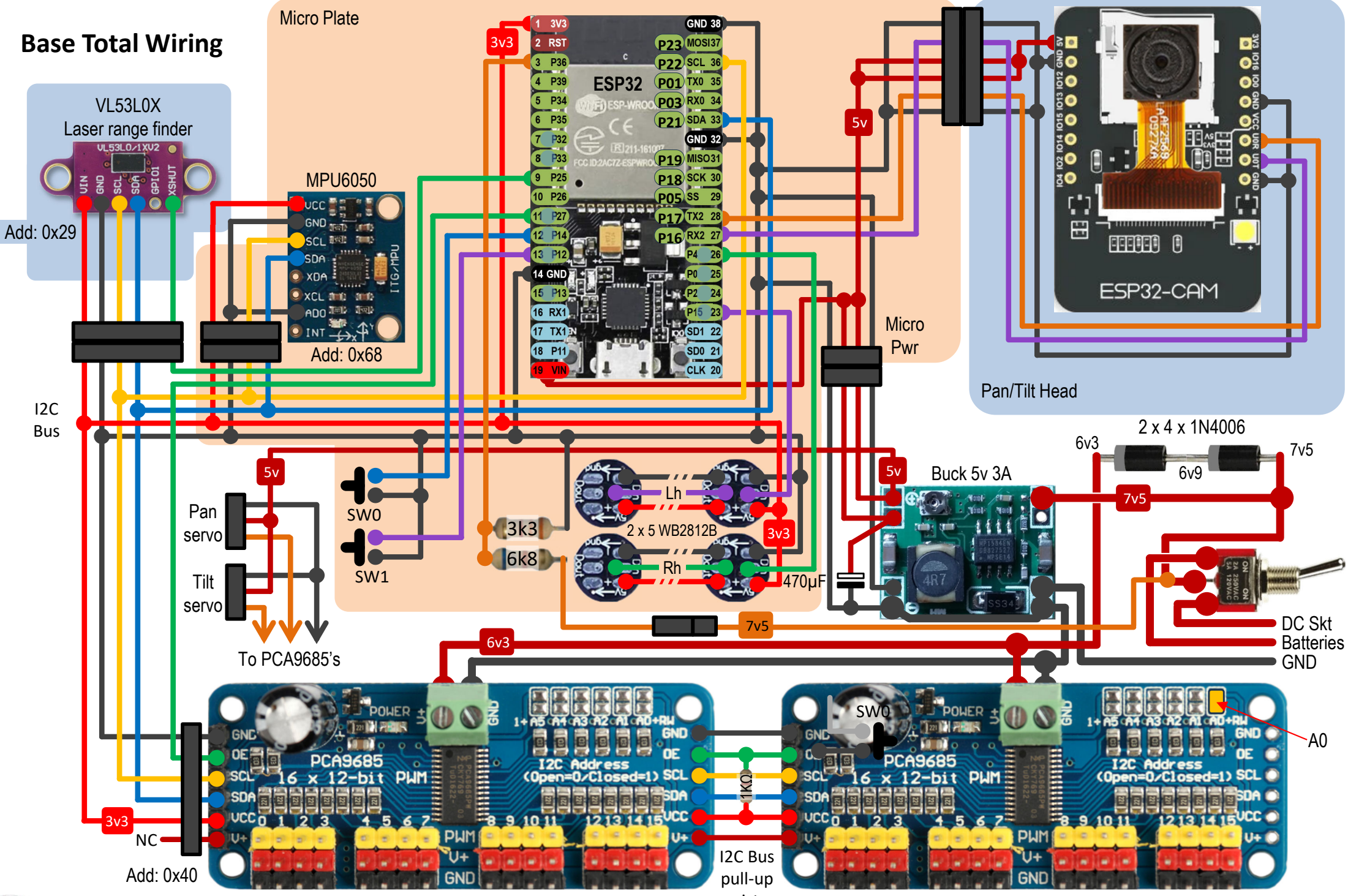
Some printed components act as aids and gauges. Use them.

# 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



# Base Total Wiring



**VL53L0X**  
Laser range finder  
VIN GND SCL SDA GPT01 XSHUT  
Add: 0x29

**MPU6050**  
UCC GND SCL SDA XDA XCL ADO INT  
Add: 0x68

**ESP32**  
GND 38 MOSI37  
P23 SCL 36  
P22 TX0 35  
P01 RX0 34  
P03 SDA 33  
GND 32  
P19 MISO31  
P18 SCK 30  
P05 SS 29  
P17 TX2 28  
P16 RX2 27  
P4 26  
P0 25  
P2 24  
P15 23  
SD1 22  
SD0 21  
CLK 20

**ESP32-CAM**  
VIN GND SCL SDA GPT01 XSHUT  
IO4 IO2 IO14 IO15 IO13 IO12 GND 5V

**Micro Pwr**

SW0 SW1  
Lh Rh  
2 x 5 WB2812B  
3k3 6k8  
470µF

**Buck 5v 3A**  
4R7

2 x 4 x 1N4006  
6v3 6v9 7v5  
7v5  
DC Sckt Batteries GND

Pan servo  
Tilt servo  
To PCA9685's

**PCA9685**  
16 x 12-bit PWM  
I2C Address (Open=0/Closed=1)  
GND DE SCL SDA UCC 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 UCC U+ GND  
Add: 0x40

**PCA9685**  
16 x 12-bit PWM  
I2C Address (Open=0/Closed=1)  
GND DE SCL SDA UCC 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 UCC U+ GND  
ADD: 0x41  
A0

Servo connections are not shown here

Servo connections are not shown here

## PCA9685 power lines

The assignment of servos to PWM channels has been done symmetrically, in such a way as to ease the cable runs to the servos in each leg. See a later slide that details this. As 20 servos are needed in this project, and only 16 channels are available in one PCA9685 board, we need to use two of them in SpidaBot.

The interface to this board is an I2C bus. As more than 16 channels are needed, the two PCA9685 boards are connected in series to extend the I2C bus, and on the 2<sup>nd</sup> board the A0 address bit is set to '1', using a solder blob, which changes the address on the I2C bus of that board from 0x40 to 0x41, with the LSB set.

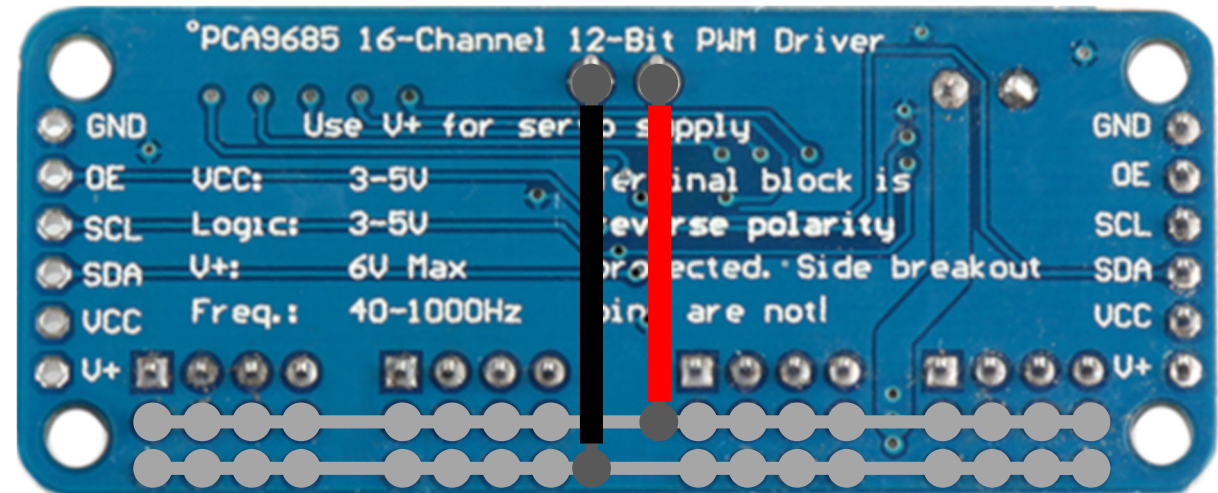
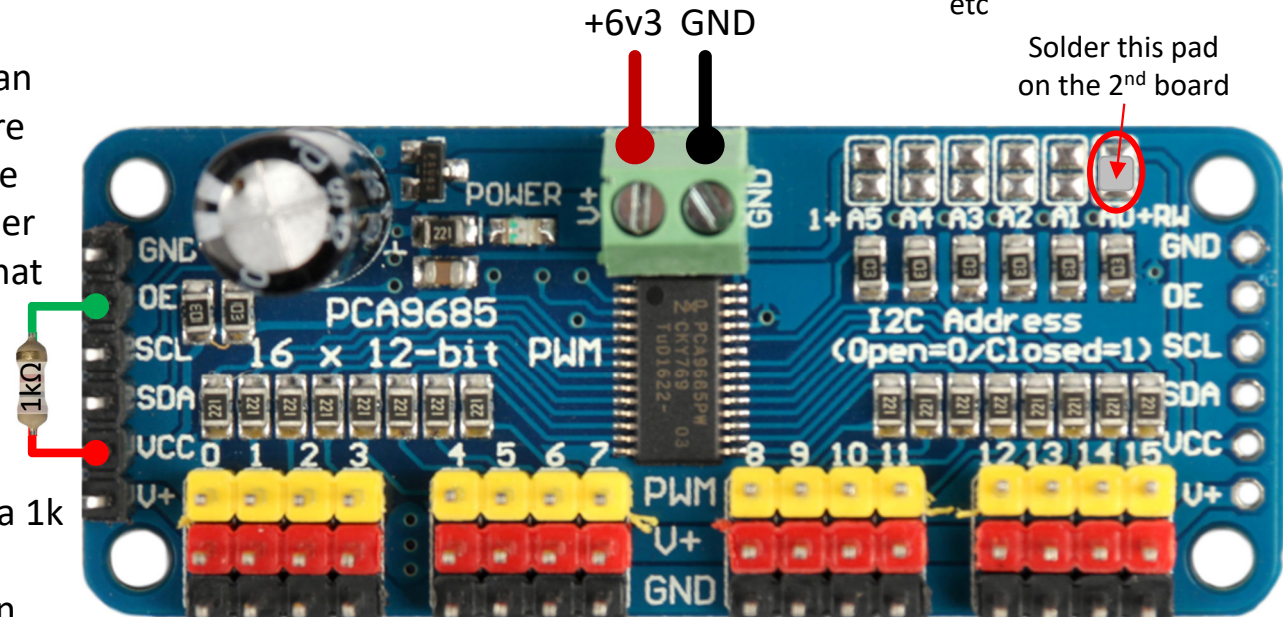
The output enable OE is active LOW, and pulled LOW by a 10k resistor on each board. To prevent servo jump at powerup, we pull this pin HIGH using a 1k resistor, as the output pins of the ESP32 will be high impedance until they are defined in code. The OE pin can then be easily driven LOW, to enable both boards as needed in your code.

All of the signals entering the left-hand side of the board, are carried through to the right-hand edge, so that several boards can be used in cascade, with even more servos. I decided to mount the 1kΩ resistor on the boards inter-connecting harness, as seen later in this document.



Default I2C address = 0x40  
with A0 shorted = 0x41  
with A1 shorted = 0x42  
with A2 shorted = 0x44  
etc

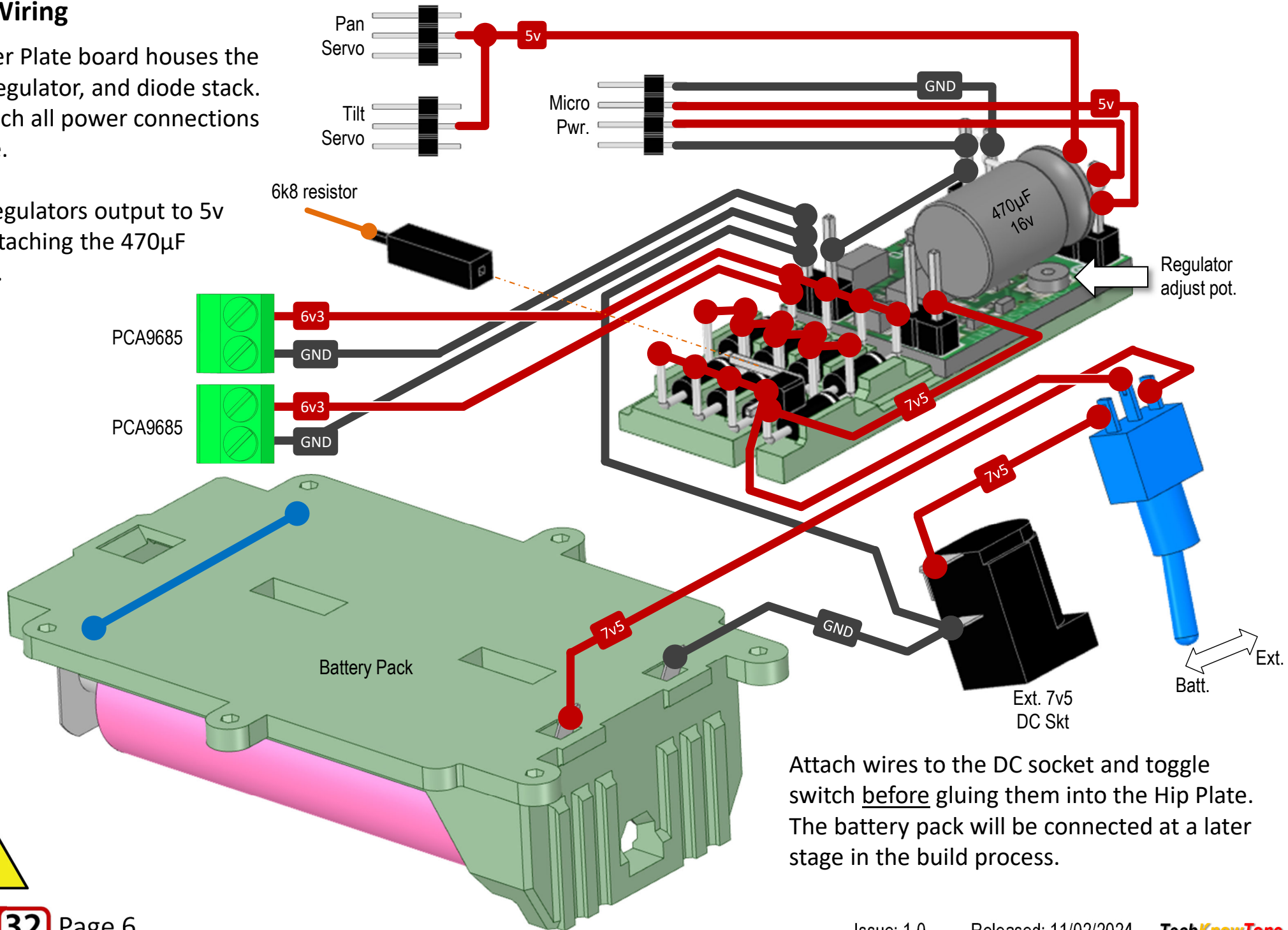
Solder this pad  
on the 2<sup>nd</sup> board



## Power Wiring

The Power Plate board houses the voltage regulator, and diode stack. From which all power connections are made.

Set the regulators output to 5v before attaching the 470 $\mu$ F capacitor.

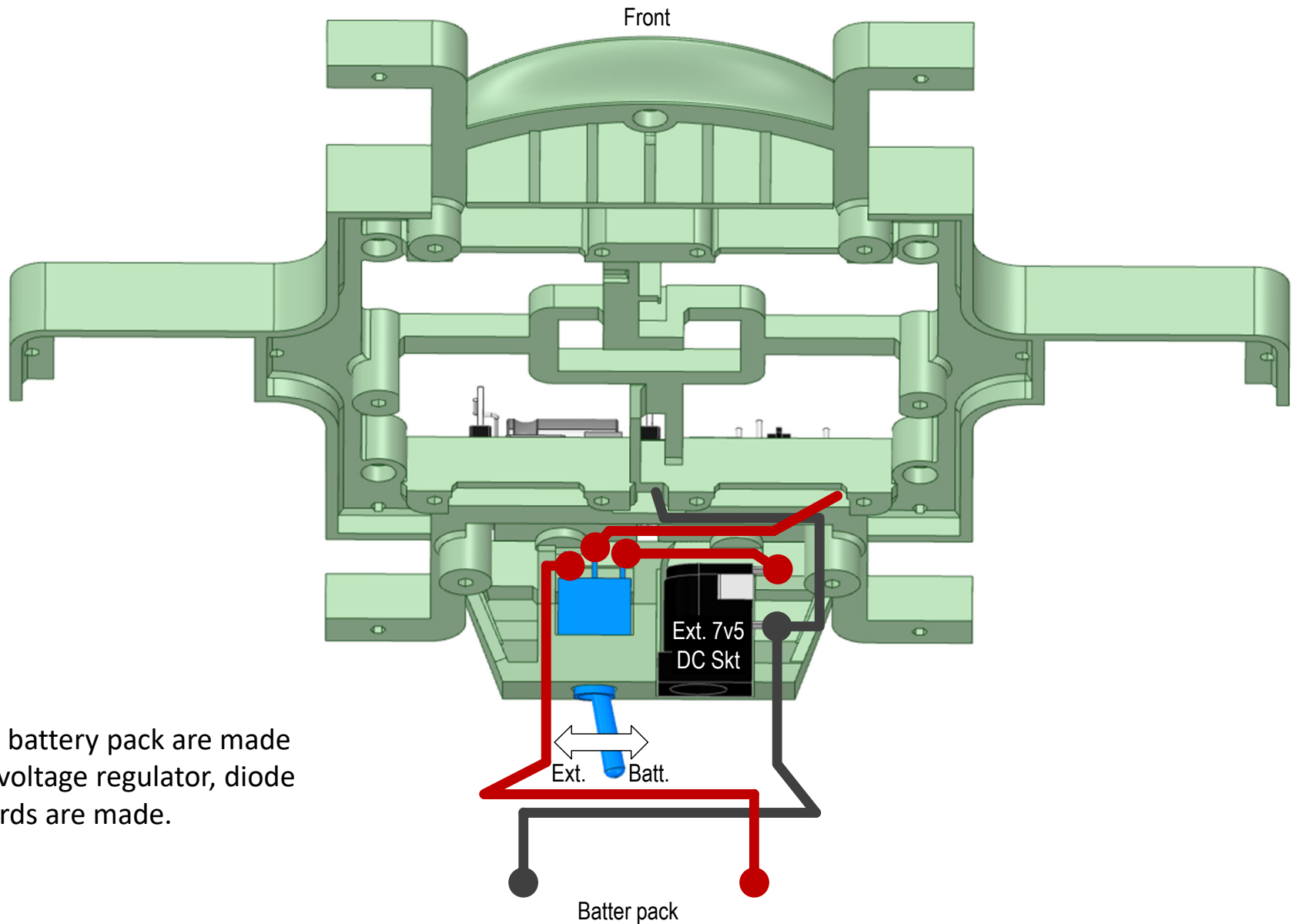


Attach wires to the DC socket and toggle switch before gluing them into the Hip Plate. The battery pack will be connected at a later stage in the build process.



## Power Wiring – underside view

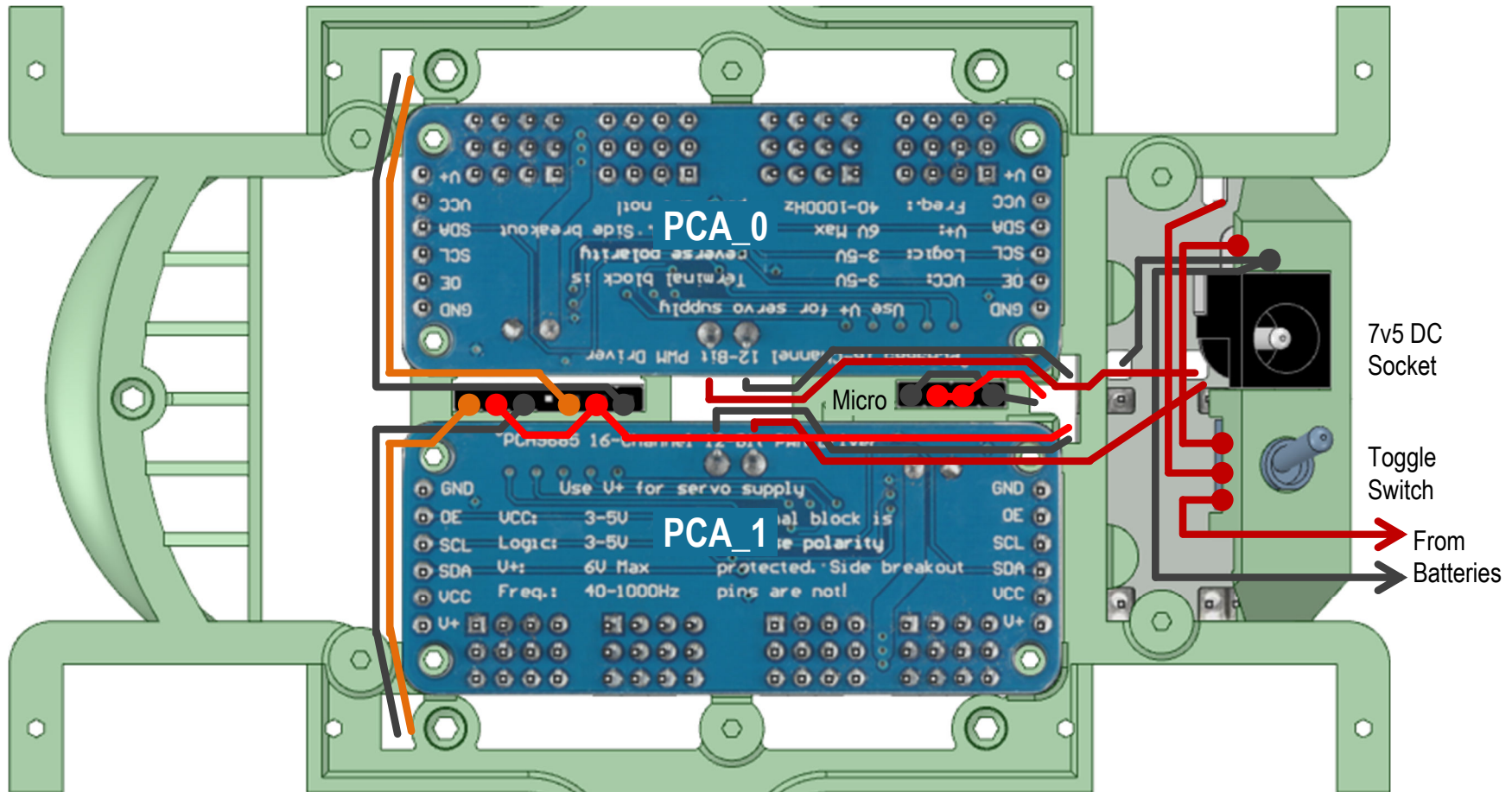
To help your understanding, this is the same information, but view from a different angle.



The connections to the battery pack are made once the wiring to the voltage regulator, diode pack and PCA9685 boards are made.

## Hip Plate Wiring – underside view

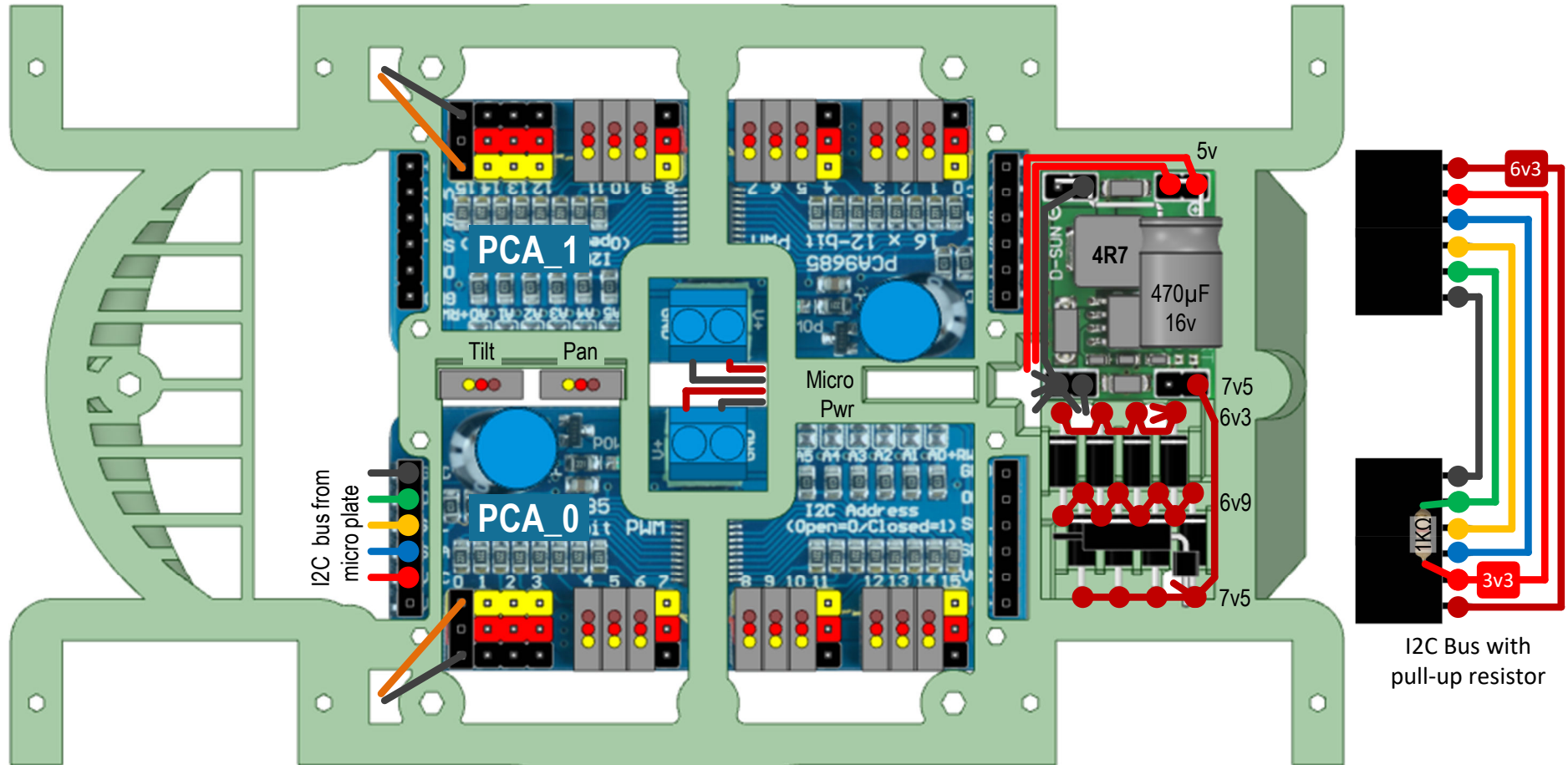
The power is routed from the regulator board as shown. It feeds the 5vDC to the 4-pin strip for the upper body, and 6v3 DC to the two terminal blocks on the PCA9685 boards. A 5v DC feed is also routed to the two 3-pin strips, for the pan and tilt servos. Note that their GND and control signals come from plugs on the other side of the boards.





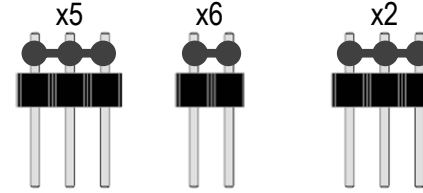
## Hip Plate Wiring – Top View

It shows the chassis plate viewed from above, with the incoming I2C connections on the lower left, fed from the upper micro plate. The wiring of the diode stack and the regulator are shown, along with the short harness that connects the two boards together. Note the mounting of the 1kΩ resistor on this harness.

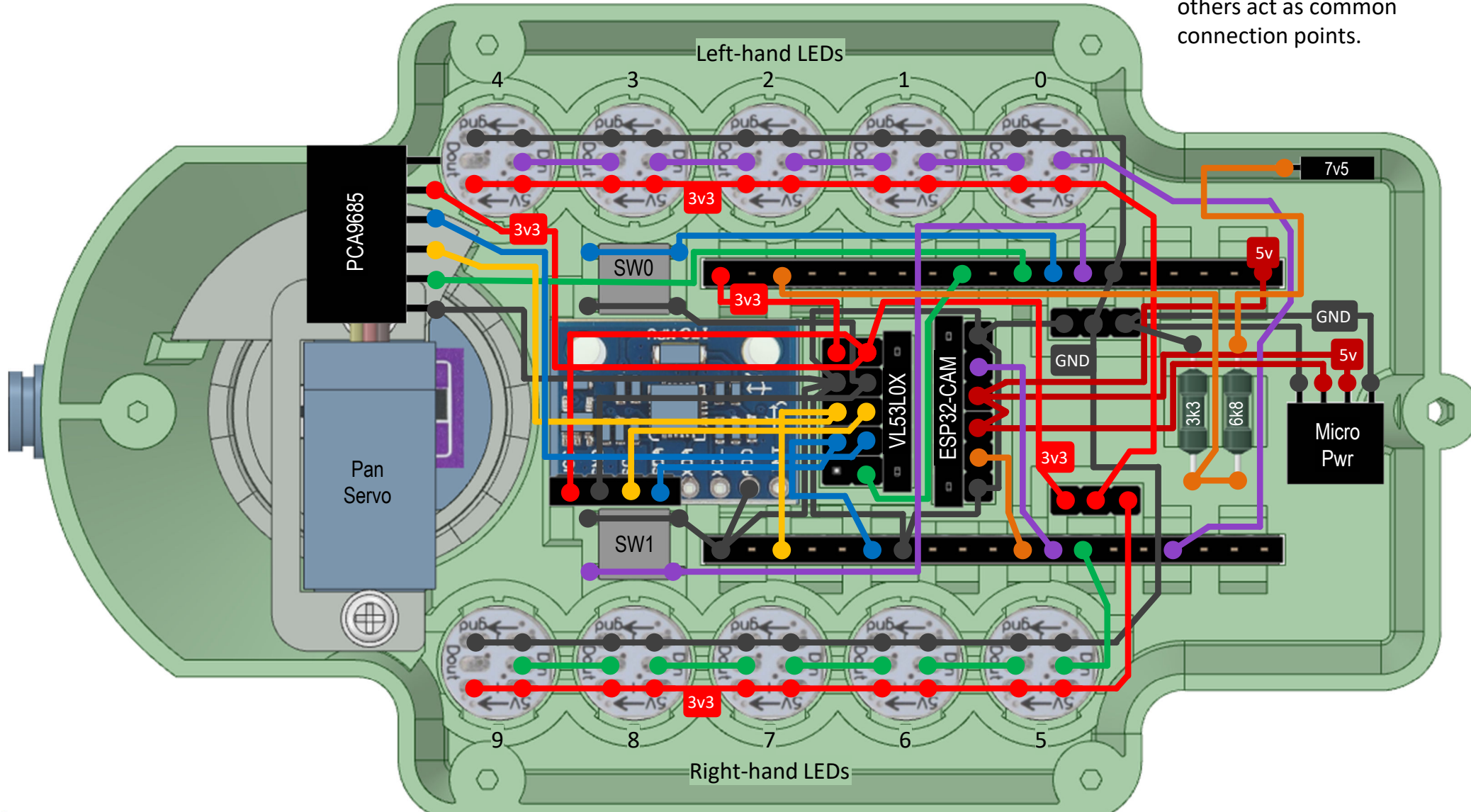


## Micro Plate Total Wiring – underside view

The micro plate houses most of the electronics circuitry, taking its 5v power supply from the Hip plate using a 4 pin socket. The ESP32 regulator then provides 3v3 supply for things like the MPU6050, and RGB LEDs.

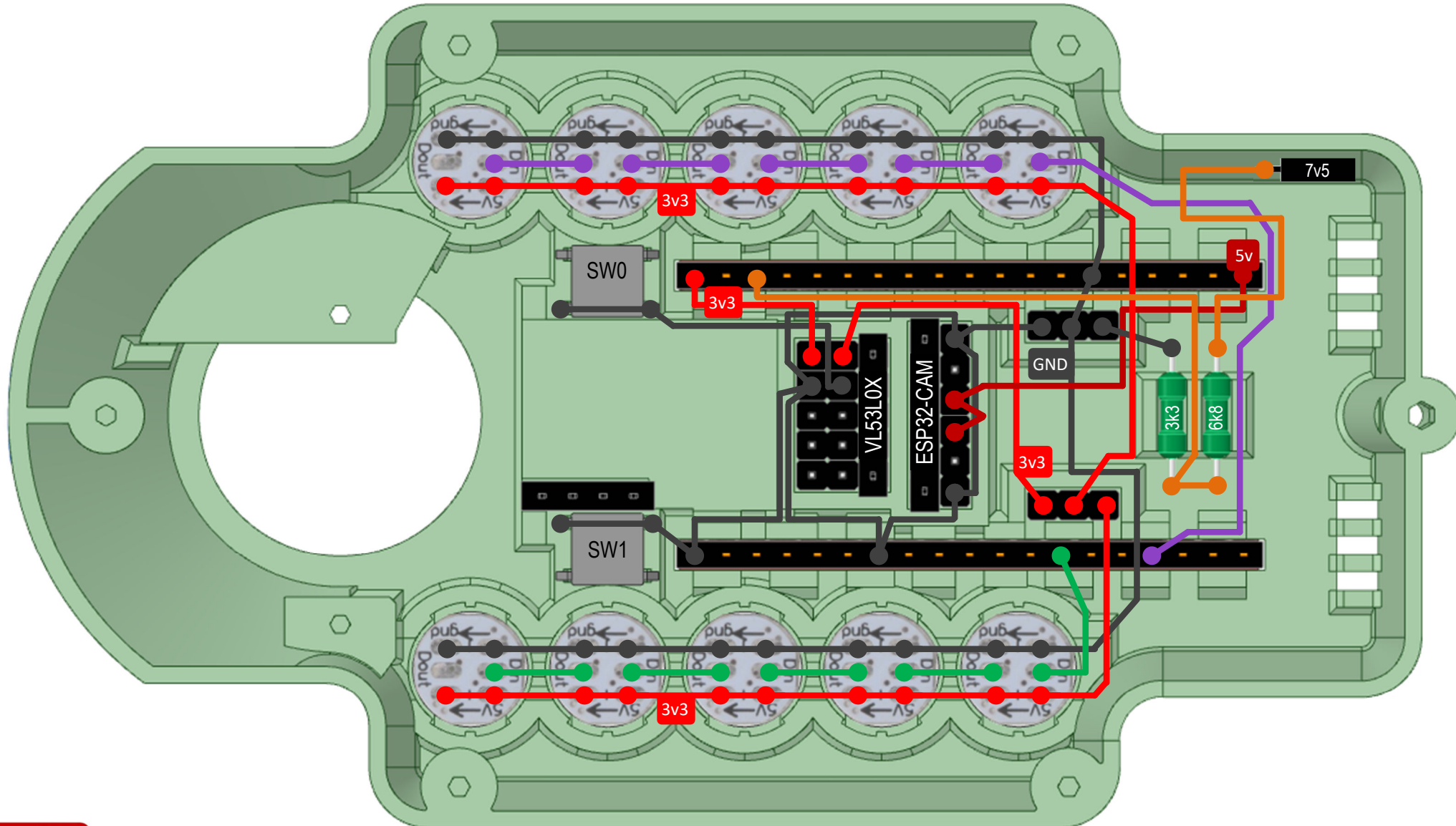
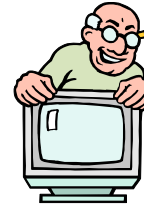


Pin strips are pre-wired, shorting the pins together, before being glued into the micro plate. Some pin strips are then arranged to act as a bus for the I2C devices, whilst others act as common connection points.



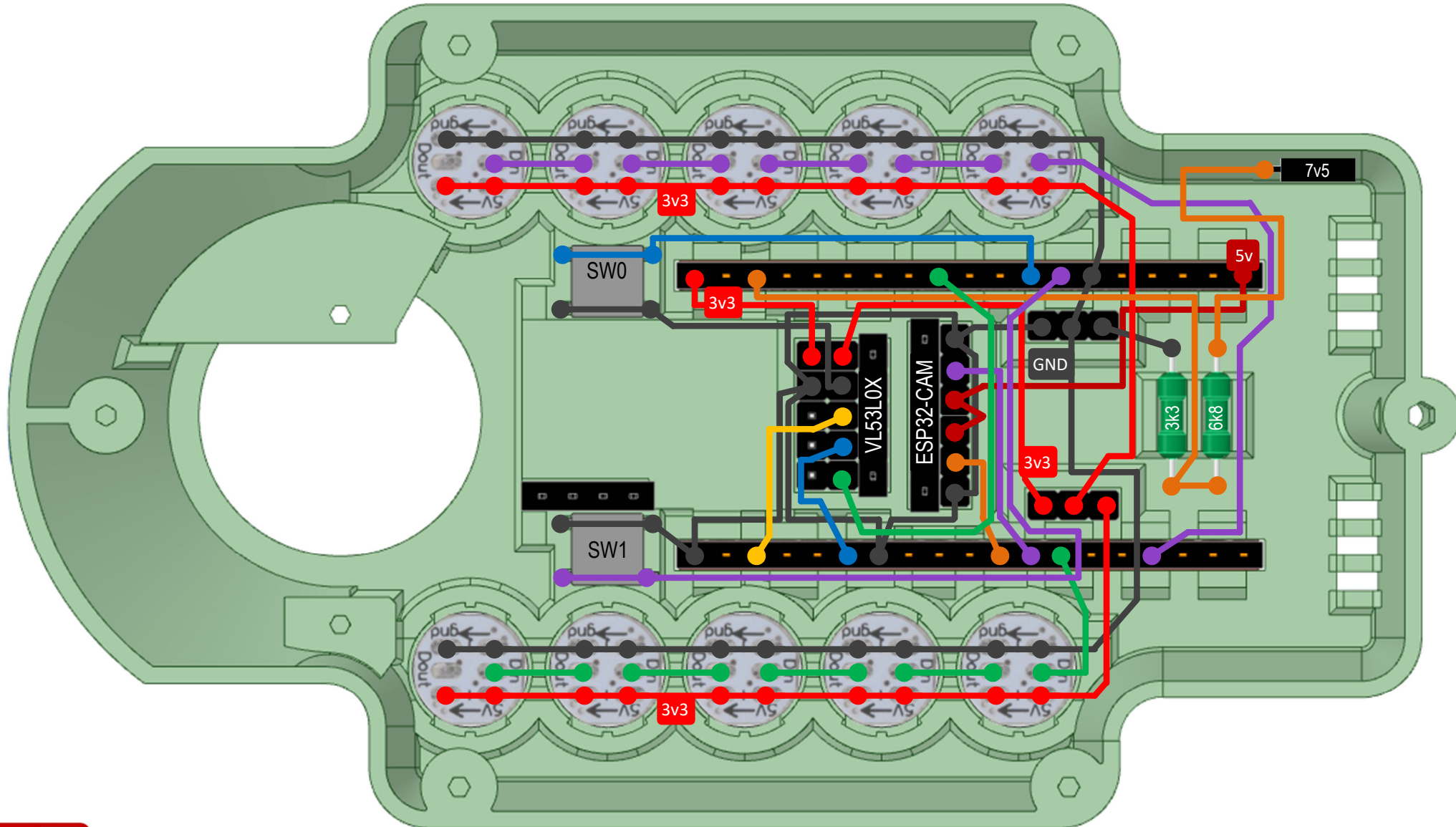
## Micro Plate Total Wiring – Stage 1

The first stage of the build process is to run in the power connections, and to terminate the LED strips, which were pre-wired in the jig. Whilst 30 AWG wire is used for most of the wiring, the power connections are made with thicker wires.



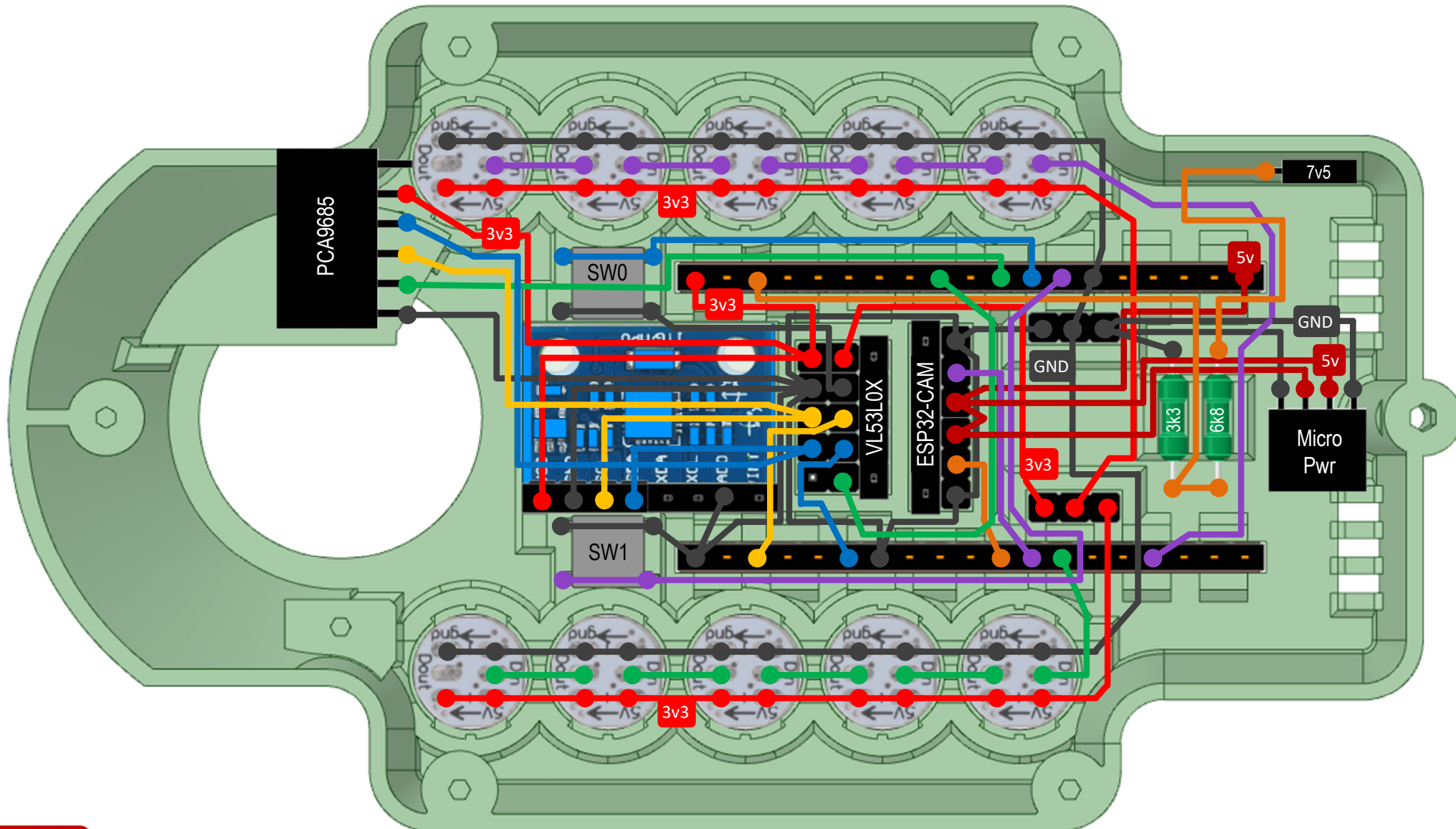
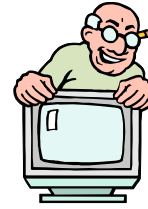
## Micro Plate Total Wiring – Stage 2

The second stage of the build process is to wire in the I2C bus connections, and those being fed to the ESP32-CAM.



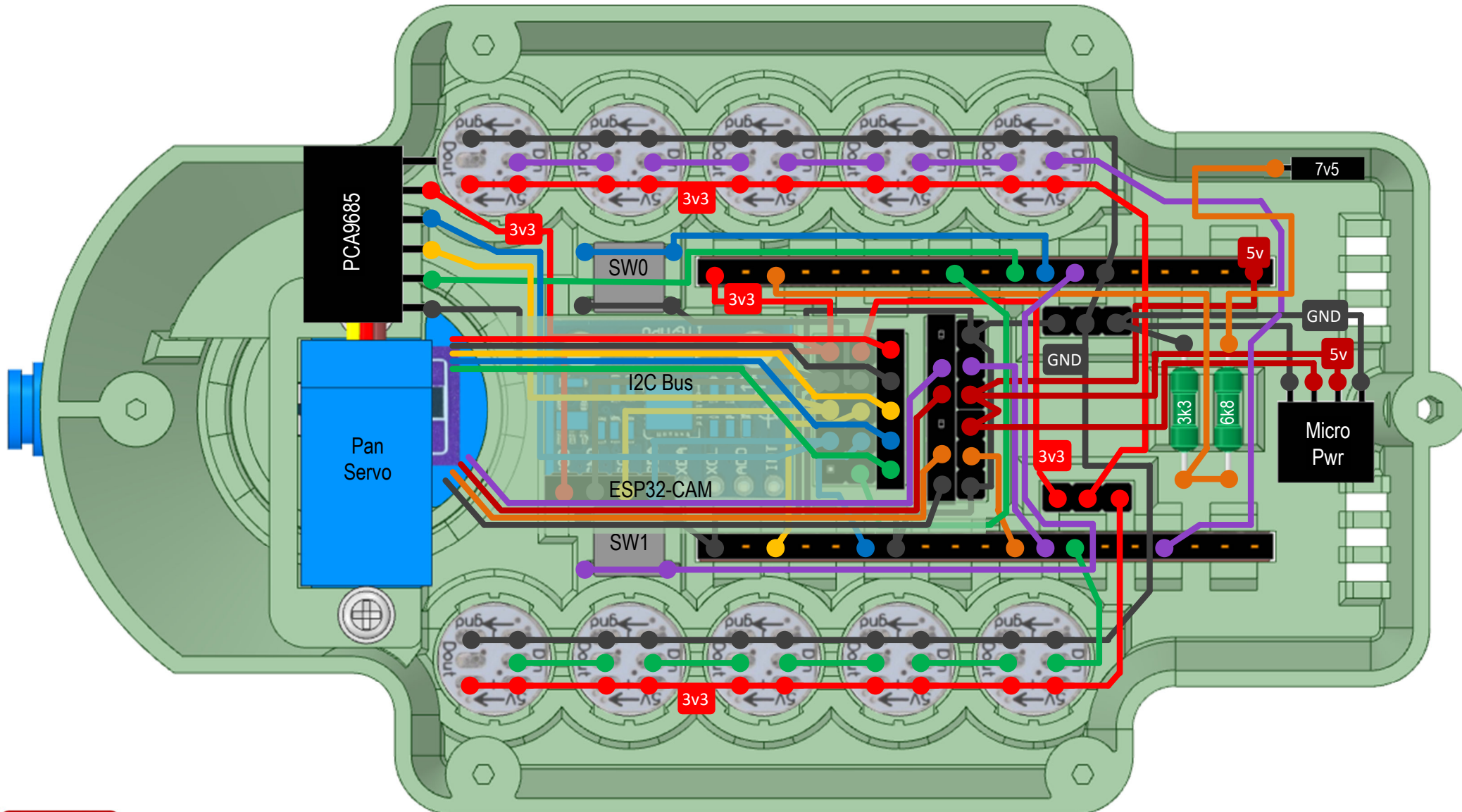
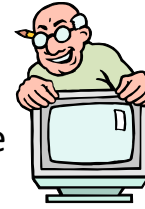
## Micro Plate Total Wiring – Stage 3

The third stage of the build process is to glue in and wire the MPU6050 motion sensor, and the pin-strip plugs for the PCA9685 I2C bus, and the power to the micro plate. Note that the AD0 line on the MPU6050 must be connected to GND.



## Micro Plate Total Wiring – Stage 4

The fourth stage of the build process is to attach the head to the micro plate, using the retaining ring, which is glued on. Then wiring in the VL53L0X range finder, and the ESP-CAM, through the rotary junction to the two pin strip sockets. Previous wiring is faded here to make things clearer.

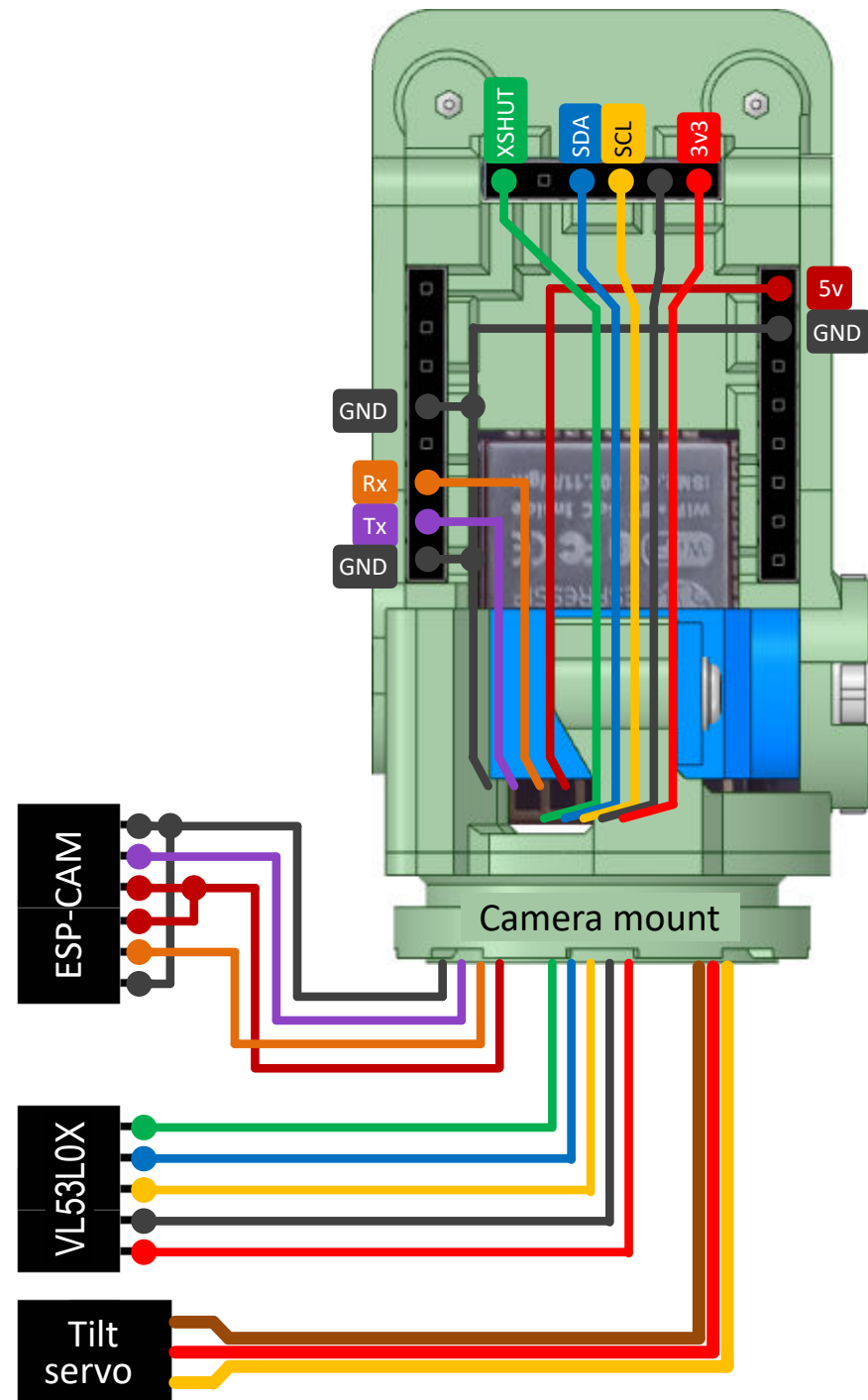


## Head Wiring

With the camera mount retaining ring glued into position, the wiring of the ESP-CAM and VL53L0X sensor can commence. We use 30 AWG wire wrap wire for this, as there is not a lot of space though the rotary junction.

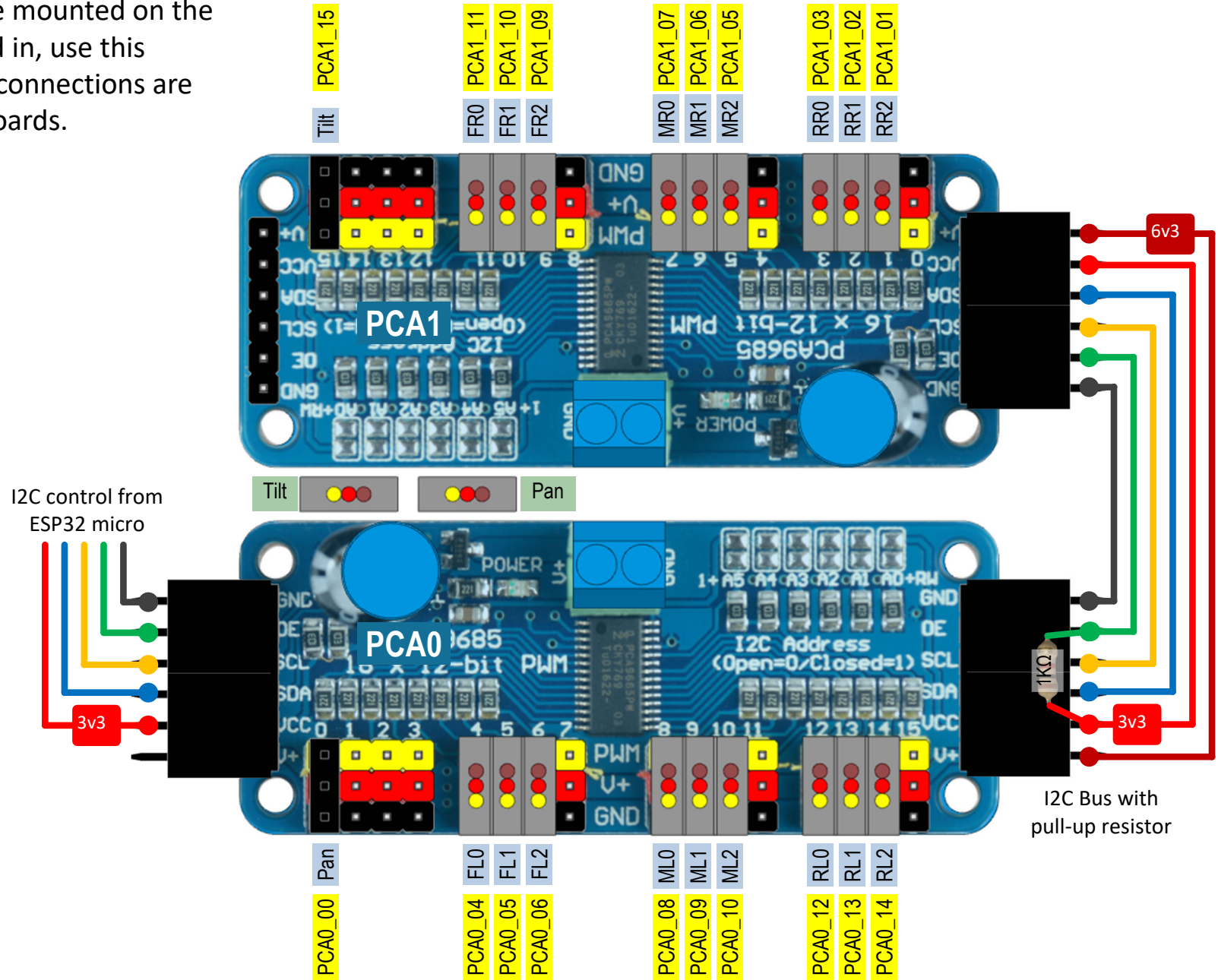
The ESP-CAM wiring is terminated in a 6-pin socket, whilst the VL53L0X sensor is terminated in a 5-pin socket. These then plug into their corresponding pin strips, in the micro plate.

You can check that everything is working, by running the code, before soldering the wire-wrap connections.



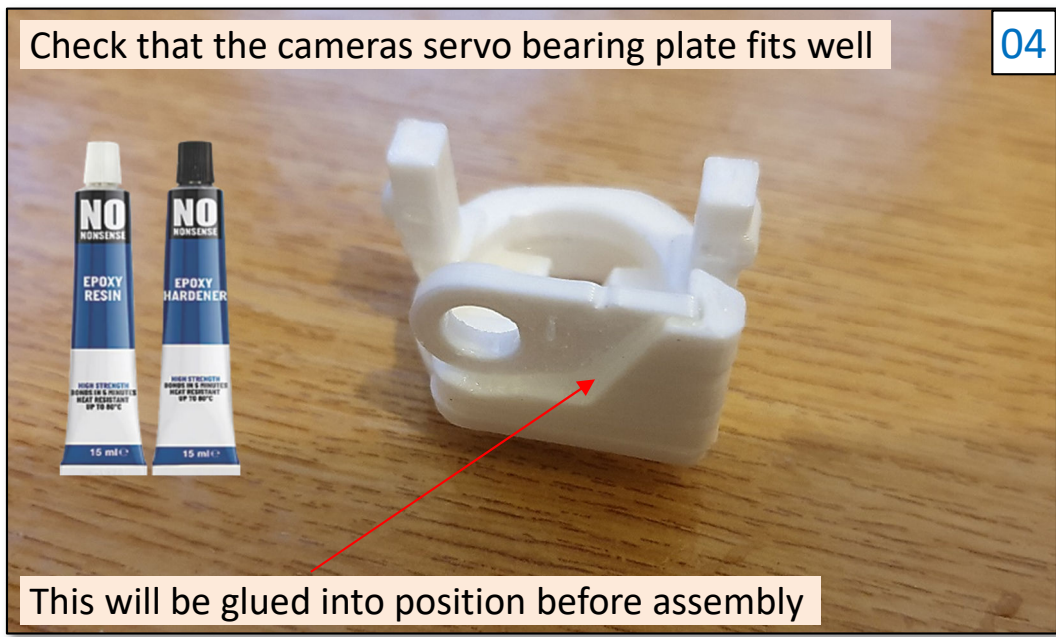
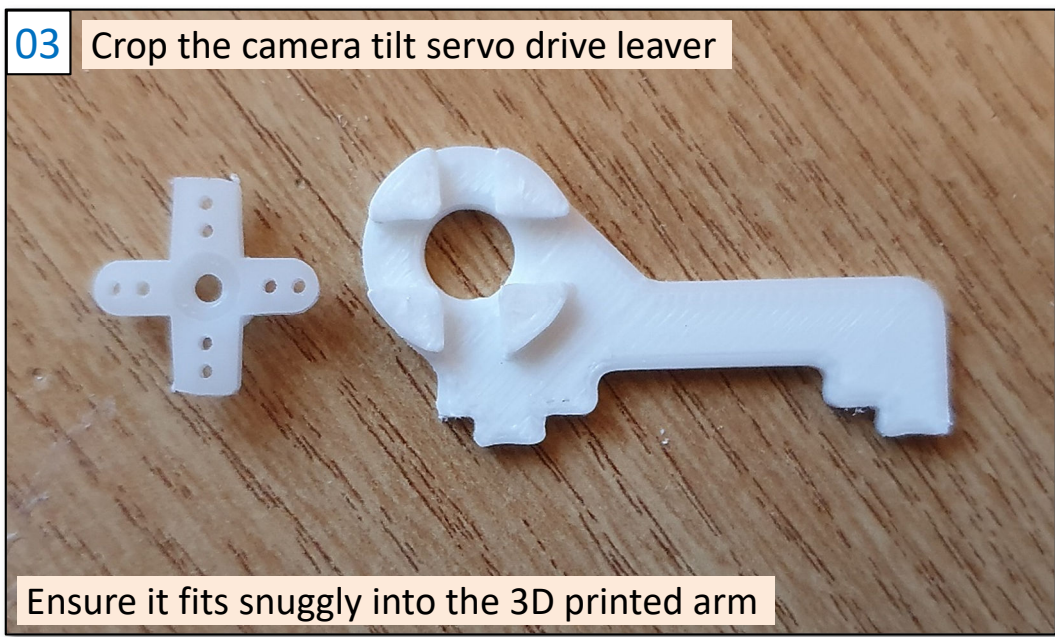
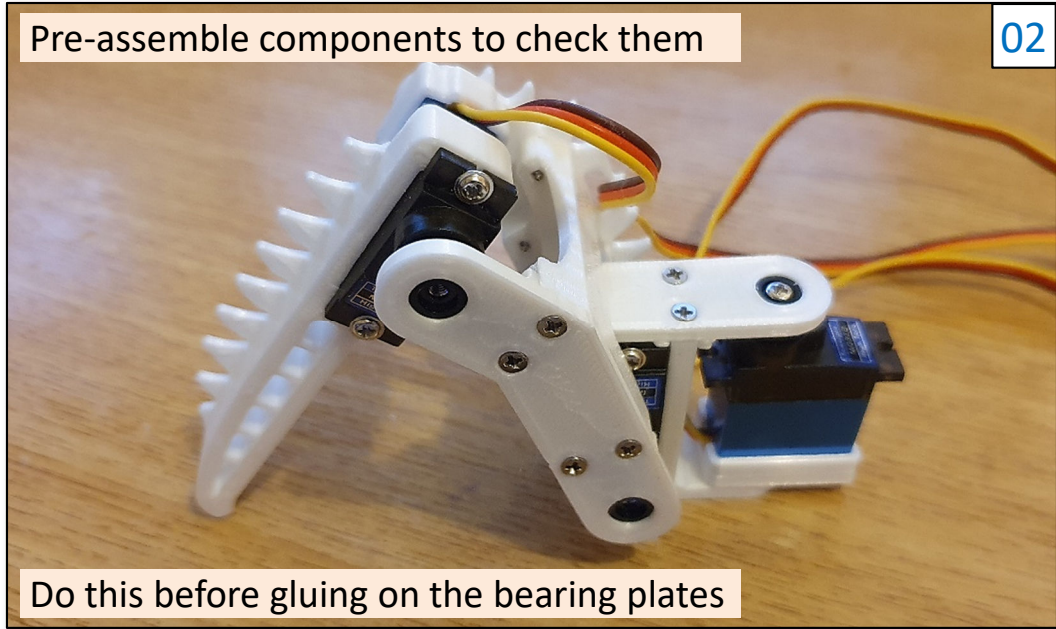
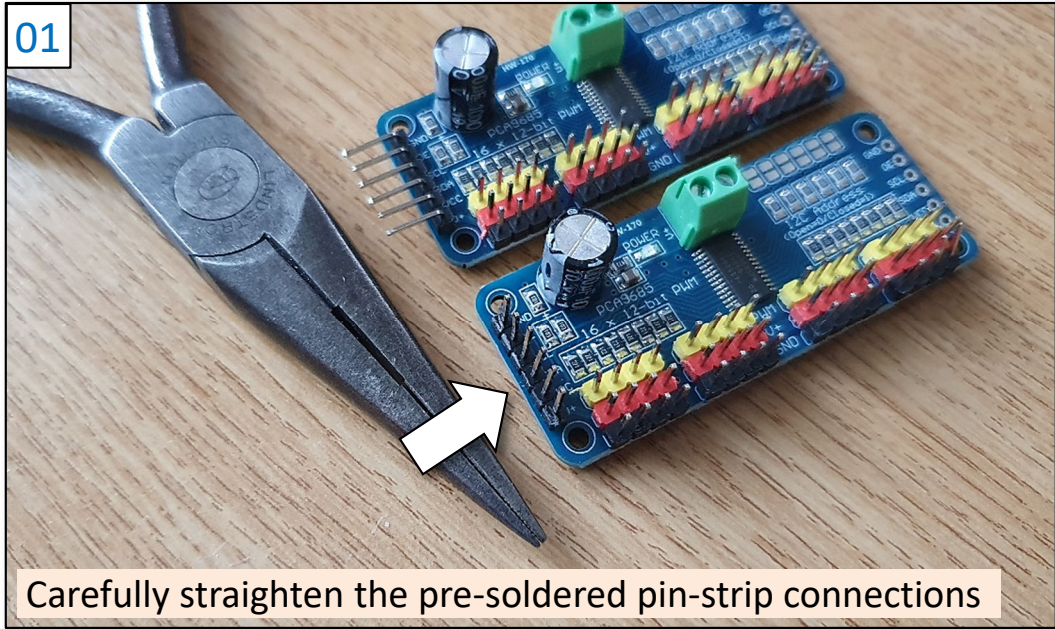
# Servo Channel Assignments

When the MG92B servos are mounted on the legs and ready to be plugged in, use this diagram to show you which connections are used on the two PCA9685 boards.



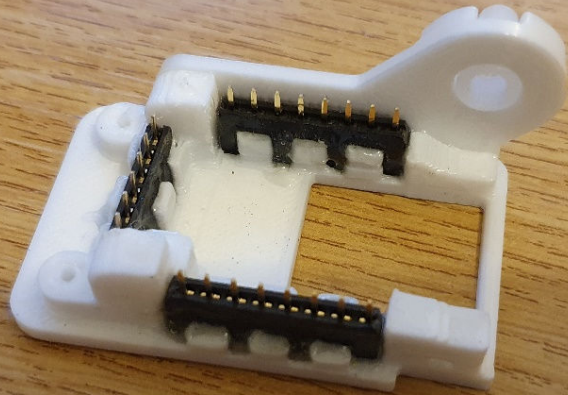


# Build Images



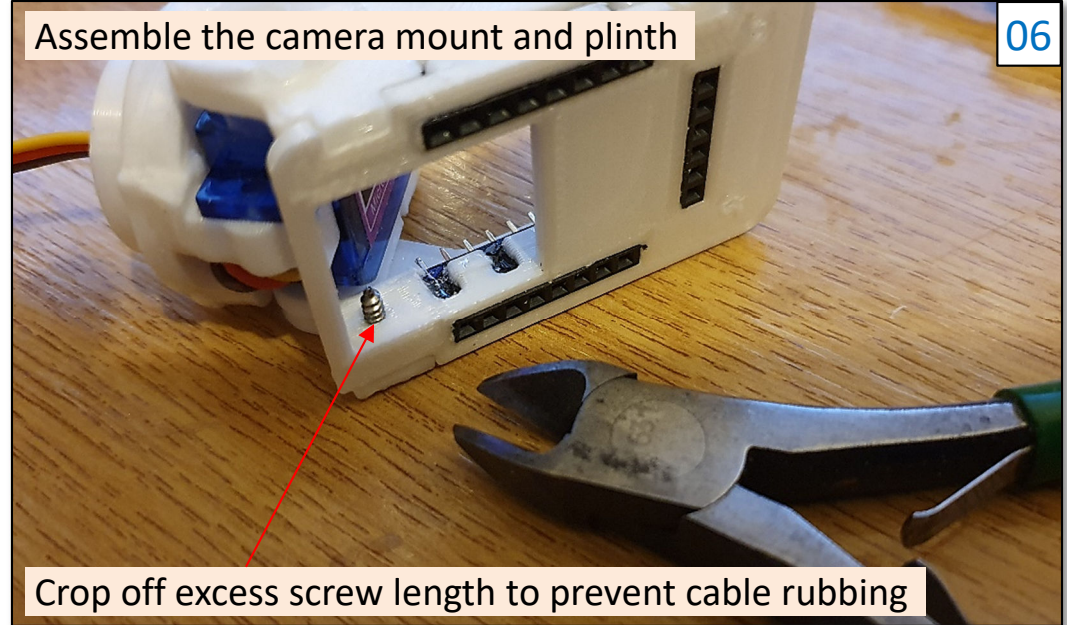
## Build Images

05 Glue the socket strips into the camera mount



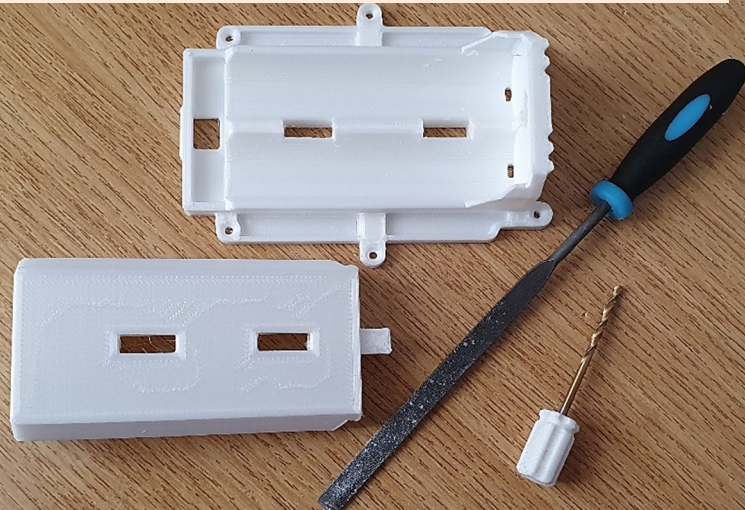
Take care not to get glue on the contacts

06 Assemble the camera mount and plinth



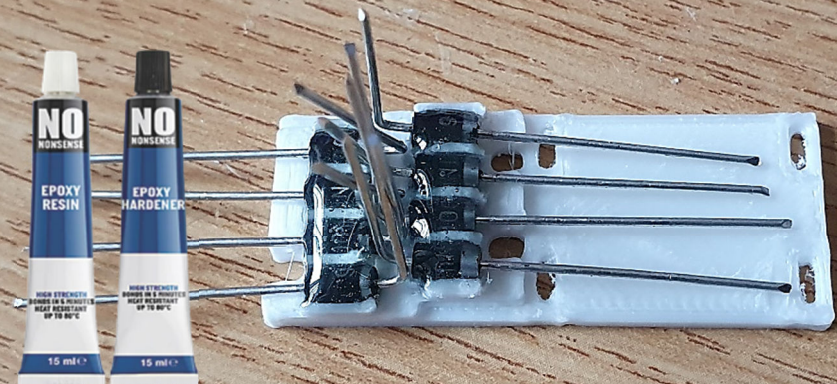
Crop off excess screw length to prevent cable rubbing

07 Visually check the 3D parts for the battery holder



Dress as necessary and clear holes with 2mm drill

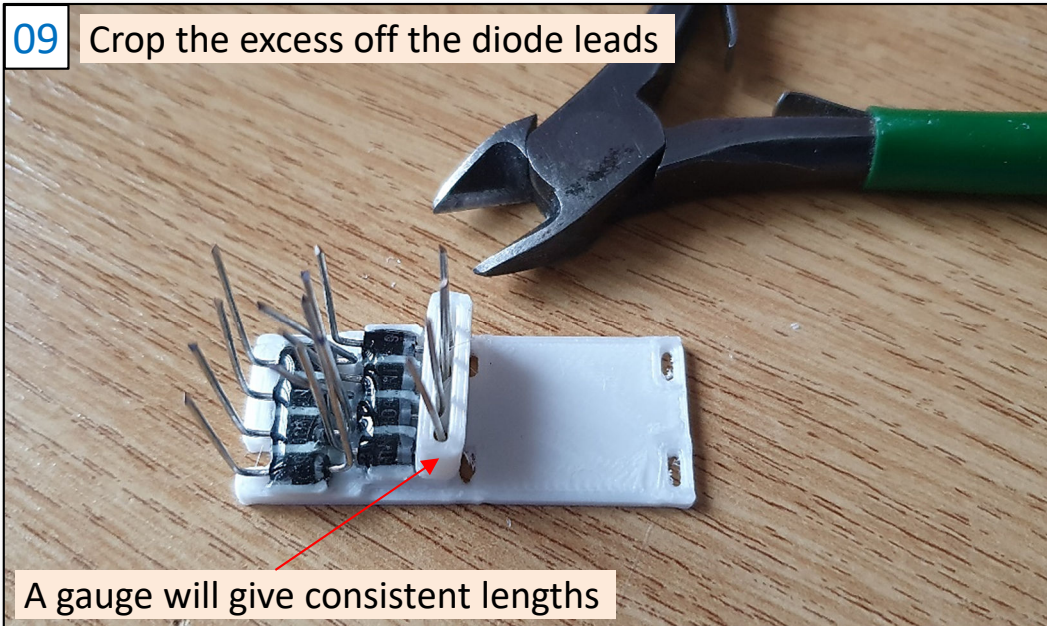
08 Glue the 8 x 1N4006 diodes onto the mounting plate



Once the glue has set, bend their legs at rightangles

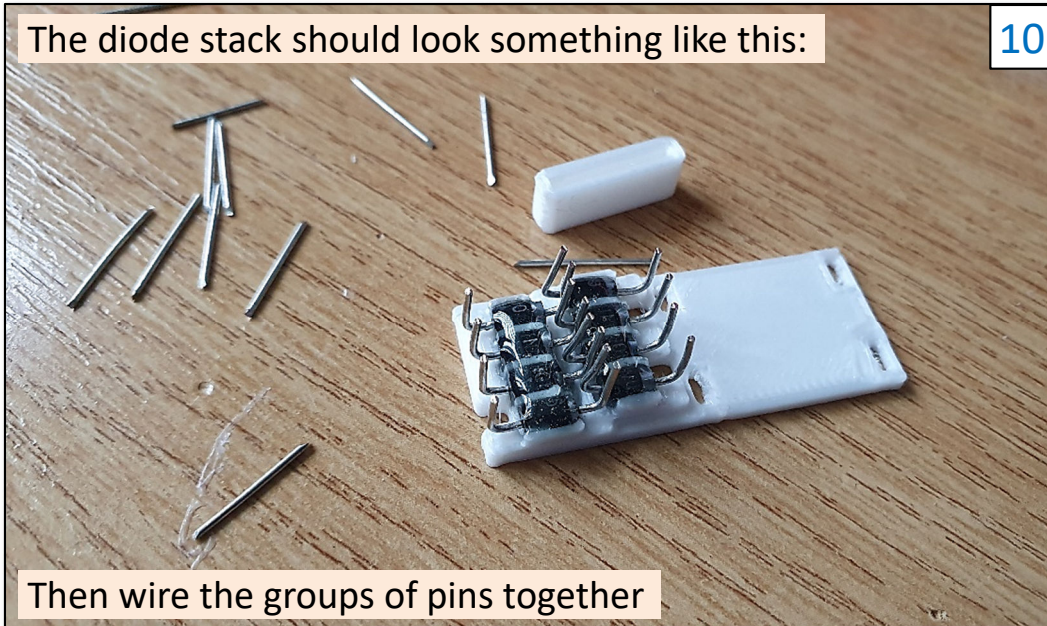
## Build Images

09 Crop the excess off the diode leads



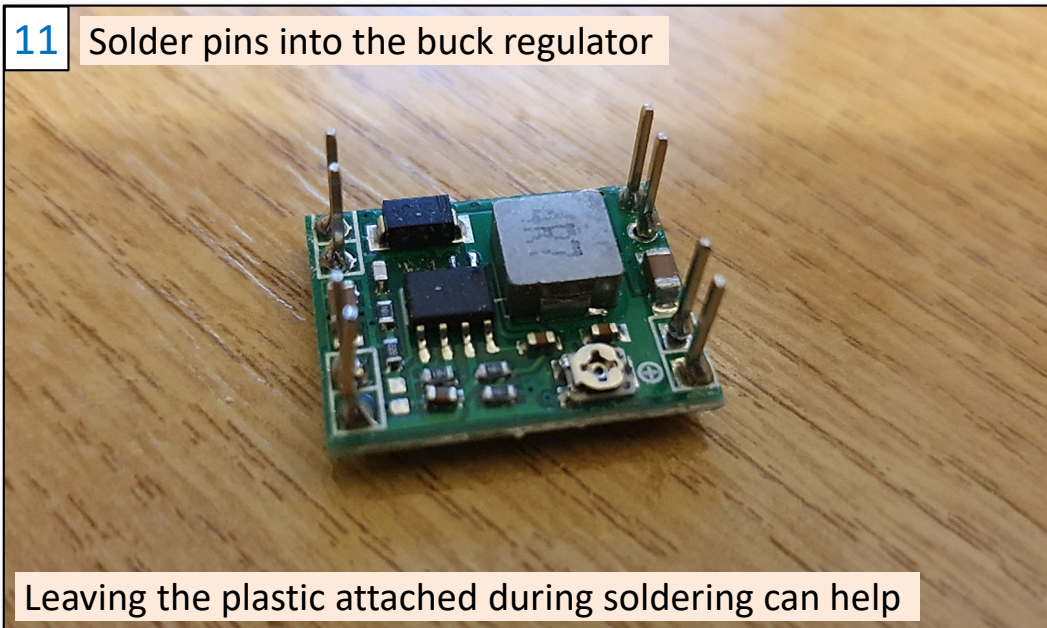
A gauge will give consistent lengths

10 The diode stack should look something like this:



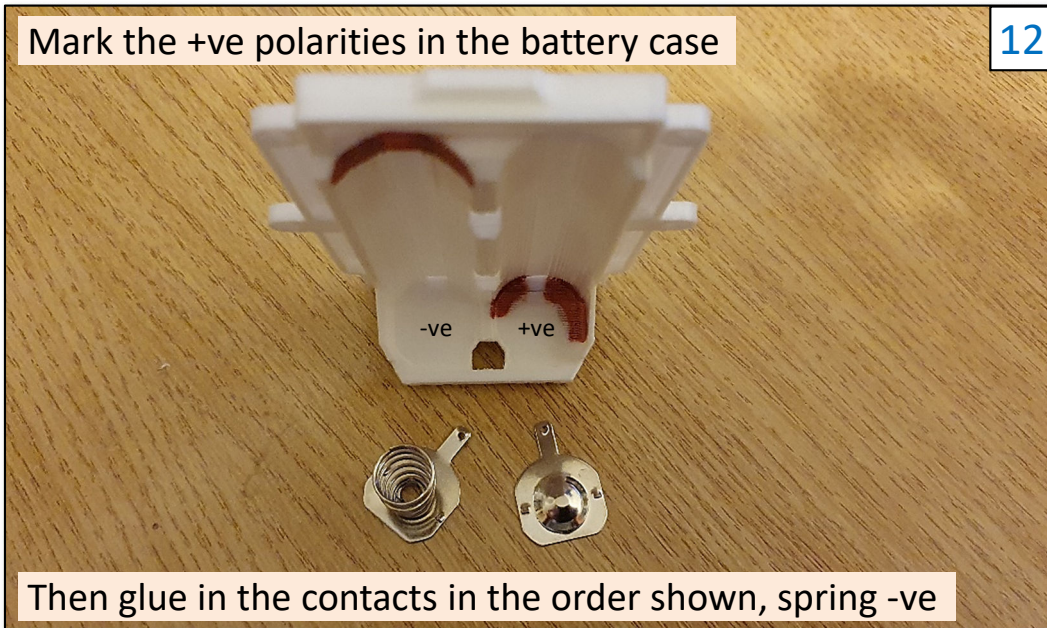
Then wire the groups of pins together

11 Solder pins into the buck regulator



Leaving the plastic attached during soldering can help

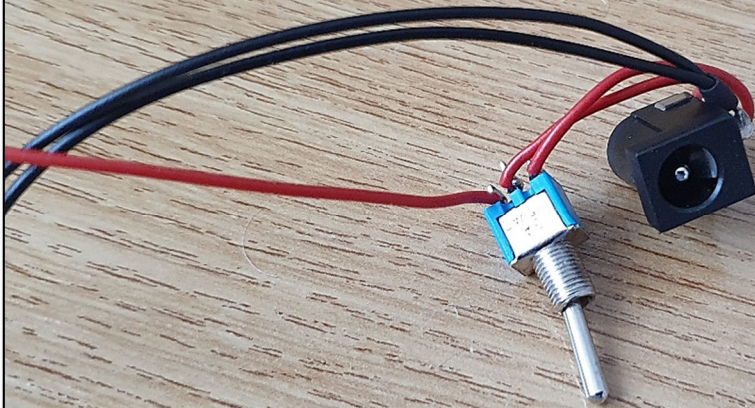
12 Mark the +ve polarities in the battery case



Then glue in the contacts in the order shown, spring -ve

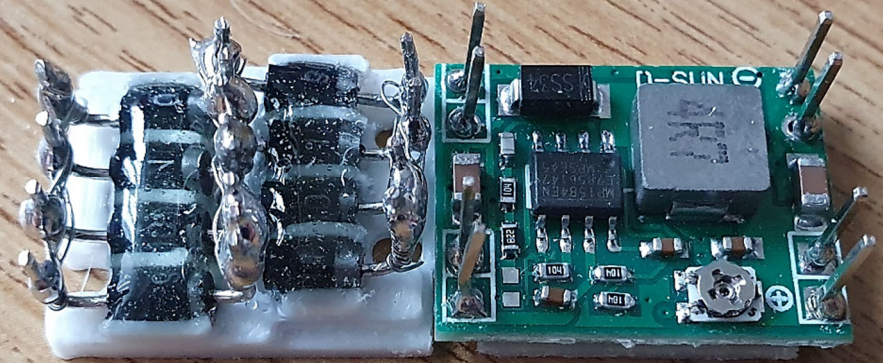
## Build Images

13 Pre-wire the power socket and switch



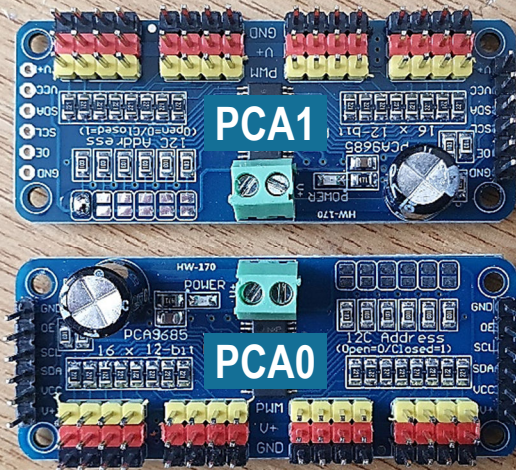
Do this before inserting them into the body plate

14 Glue the buck regulator onto the power plate



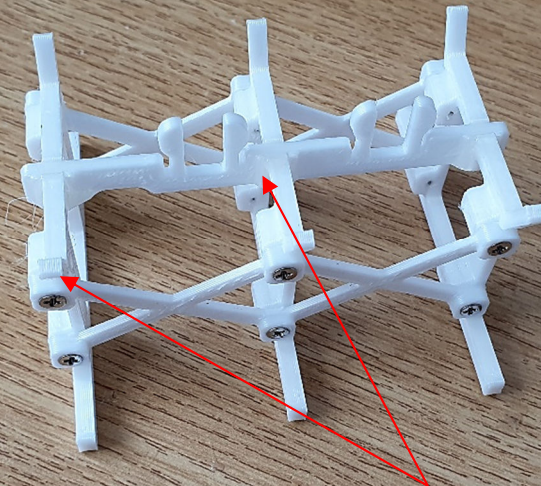
Then attach a power source and adjust for 5v output

15 Note that PCA1 does not have an output pin strip



PCA0 has the original pin strip bent and another attached

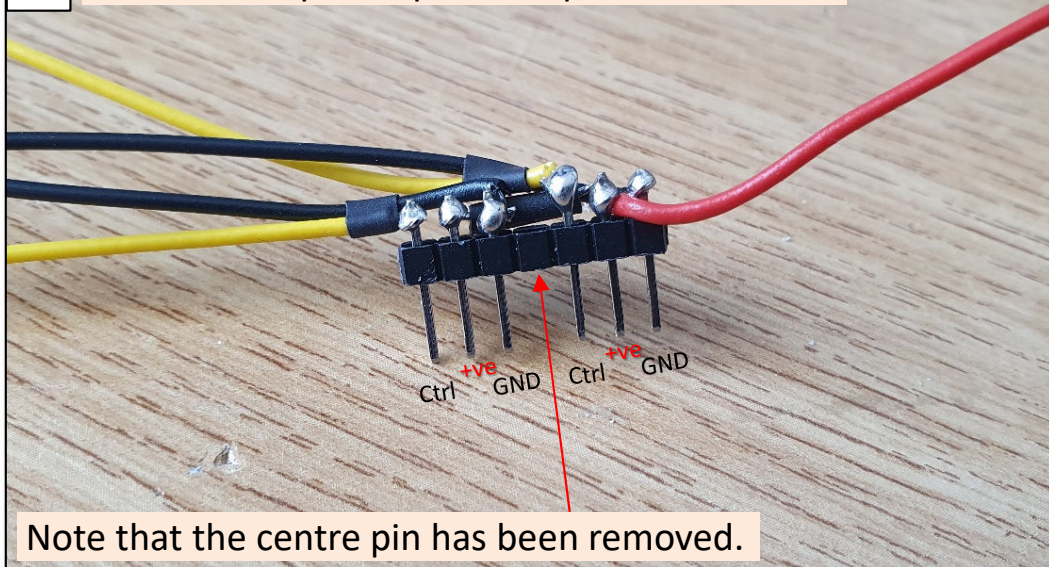
16 The stand side plates are screwed onto the cross pieces



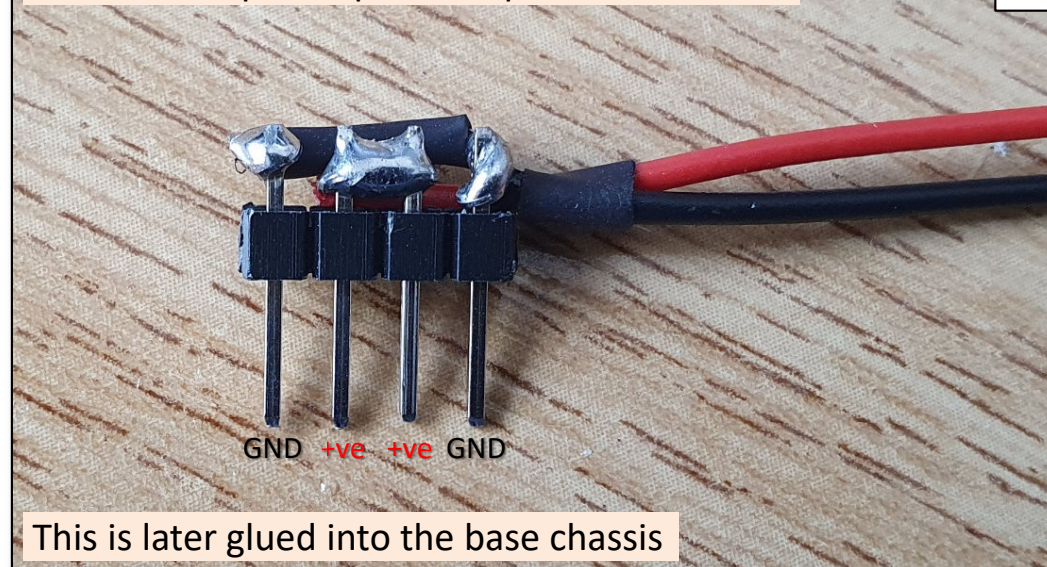
Then glue in the top support bar. Note alignments.

## Build Images

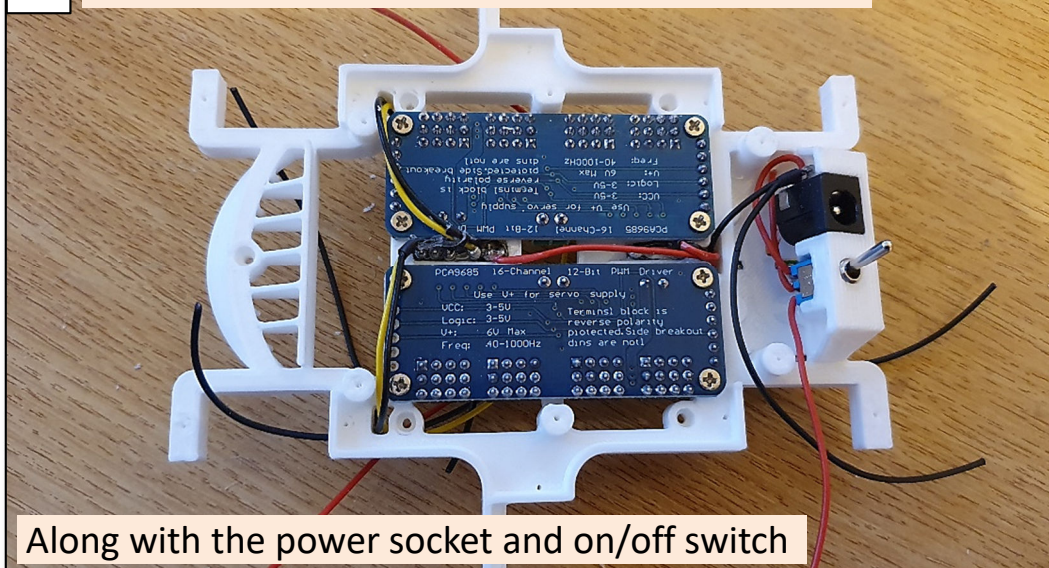
17 Pre-wire a 7-pin strip for the pan & tilt servos



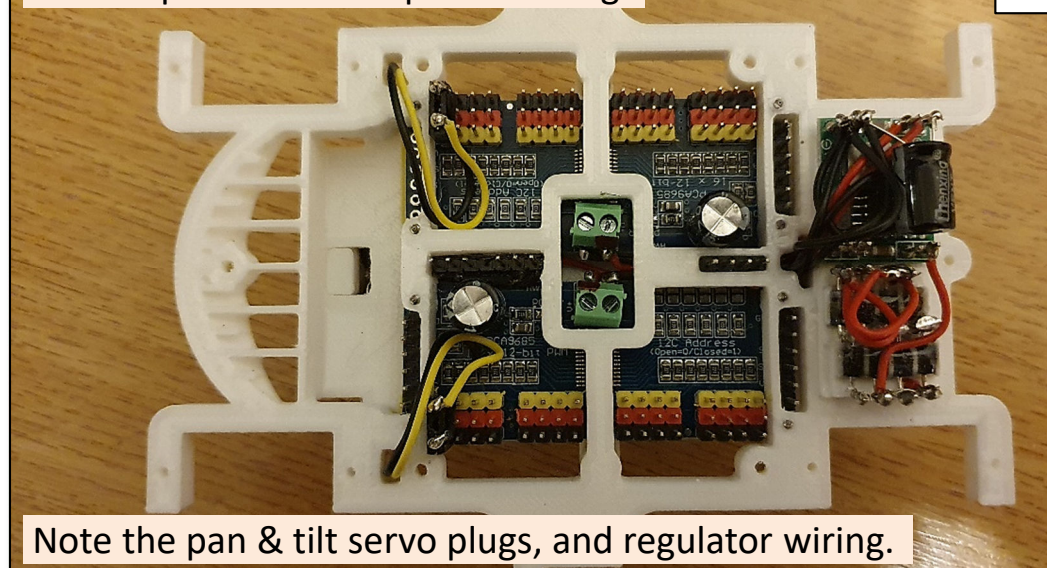
18 Pre-wire a 4-pin strip for the power connector



19 Mount the two PCA9685 boards in the chassis.

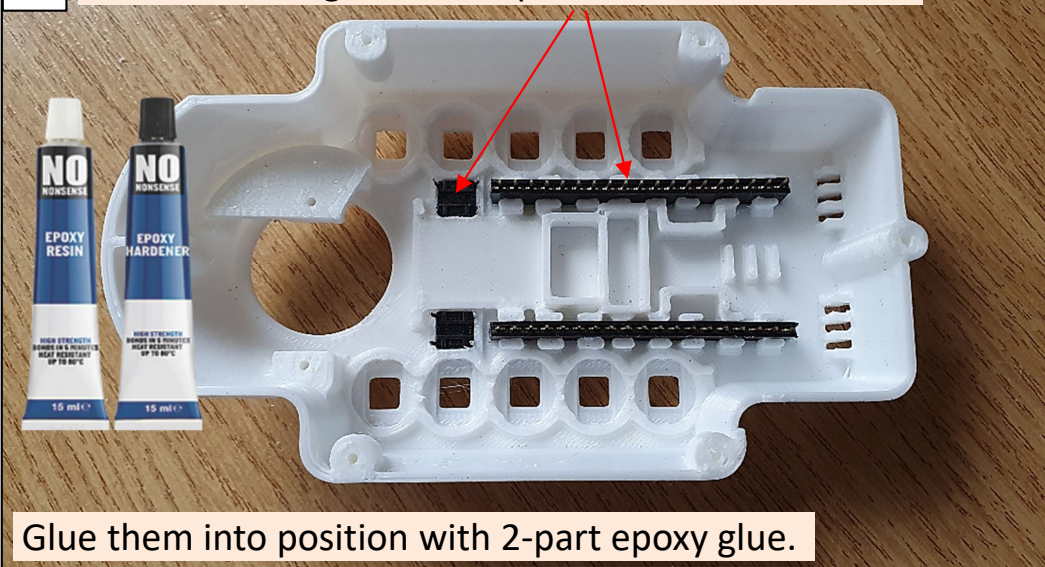


20 Chassis plate with completed wiring.

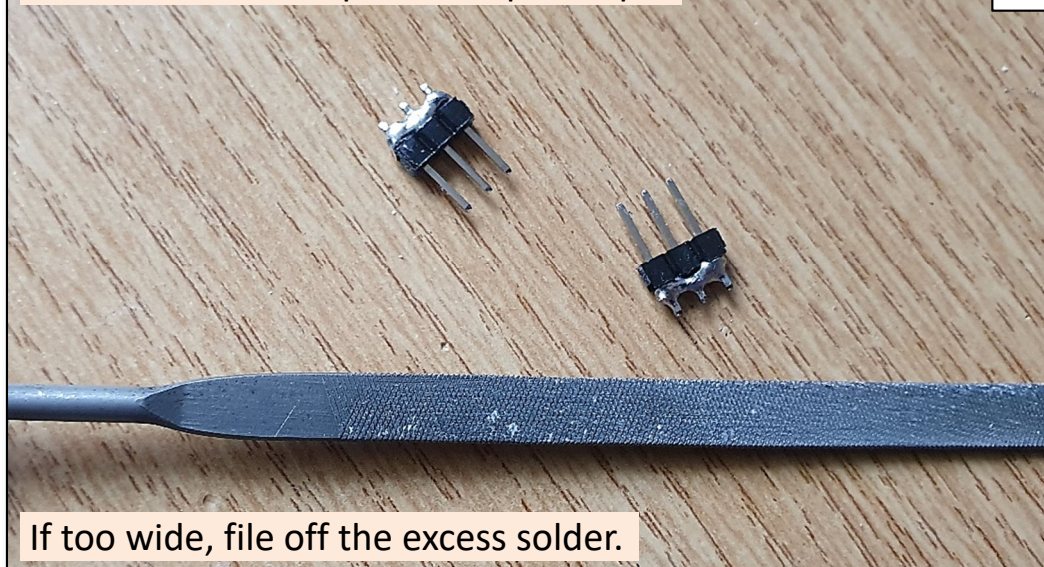


## Build Images

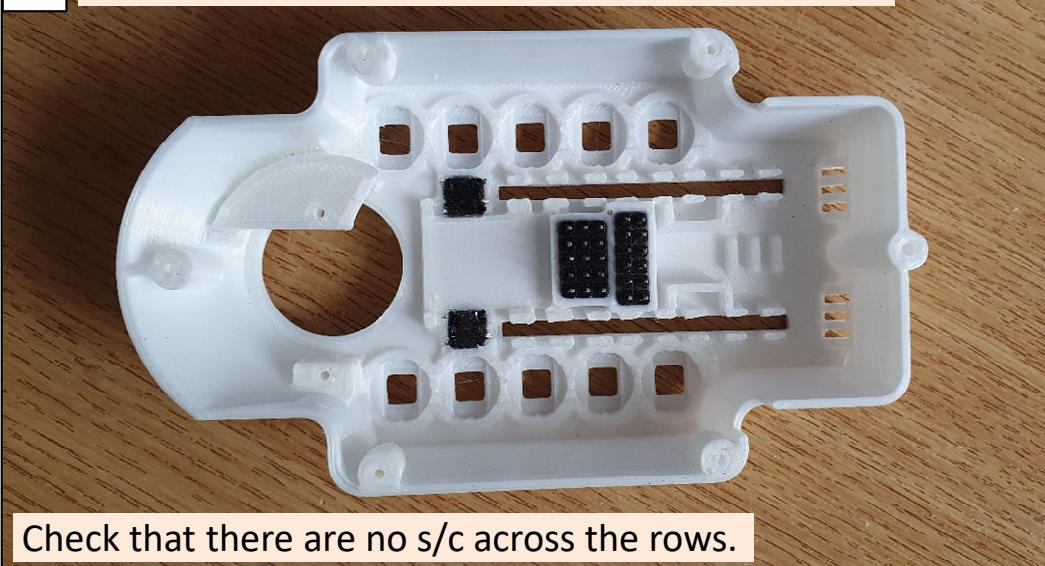
21 Mount the long socket strips and button switches.



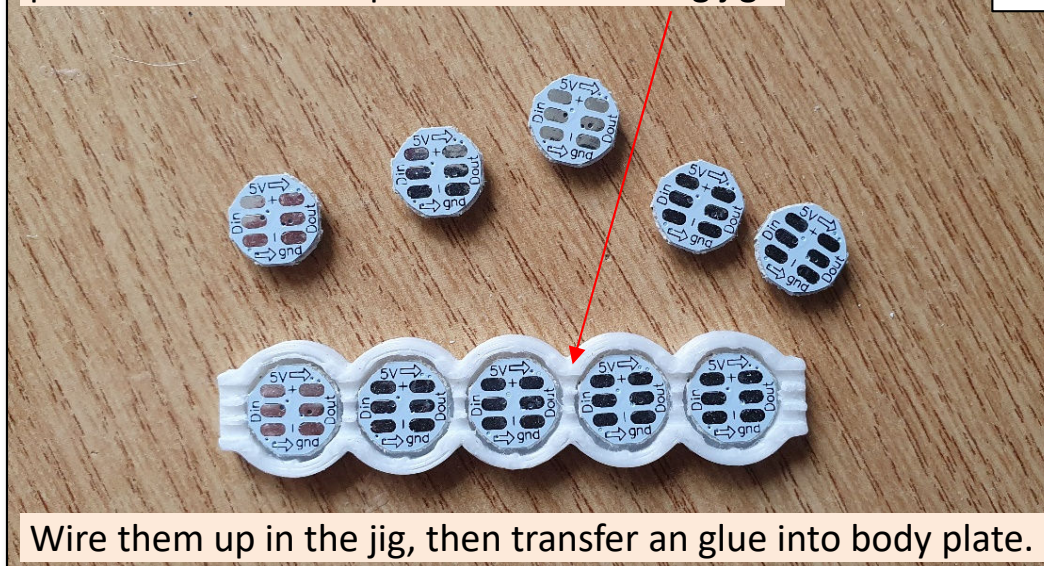
22 Solder wire onto 3-pin and 2-pin strips.



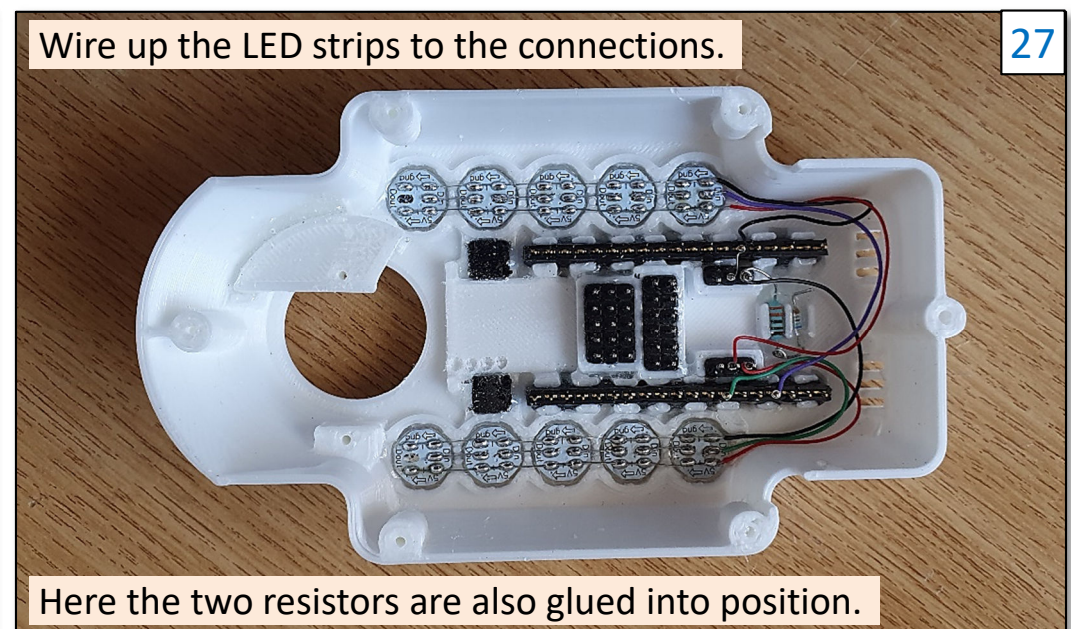
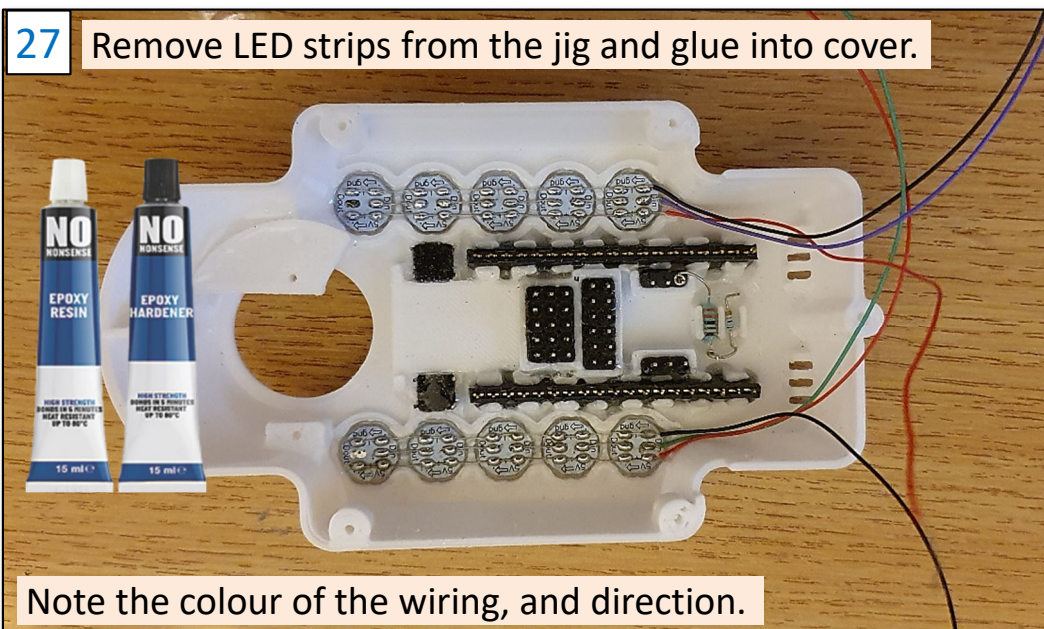
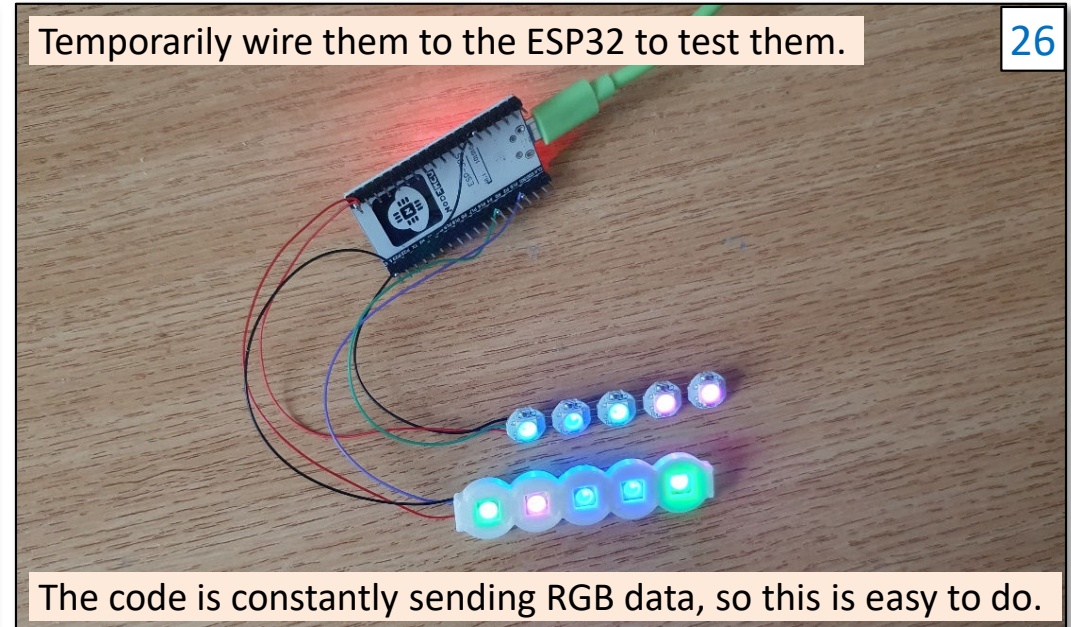
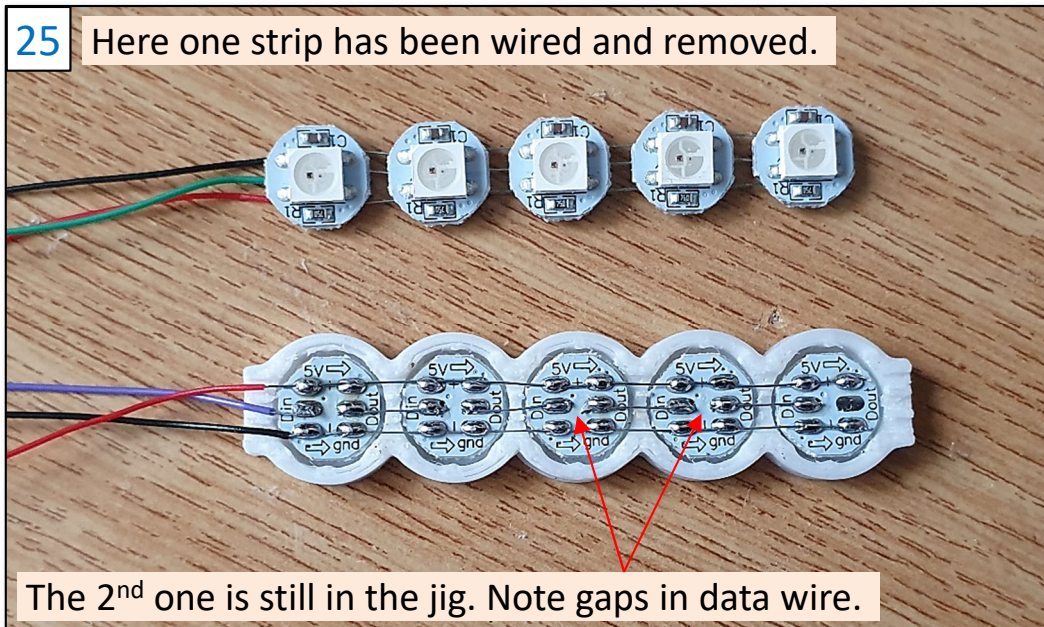
23 Glue them into the holders moulded in the case.



24 place 5 WS2812B chips into the mounting jig.

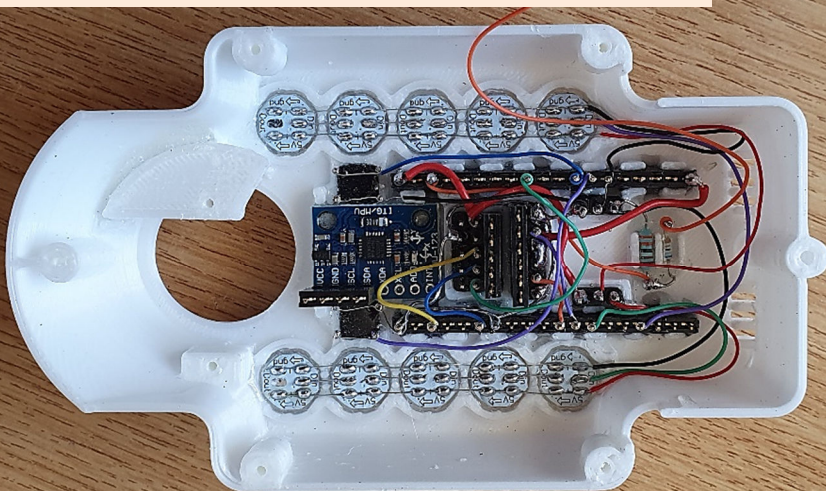


## Build Images



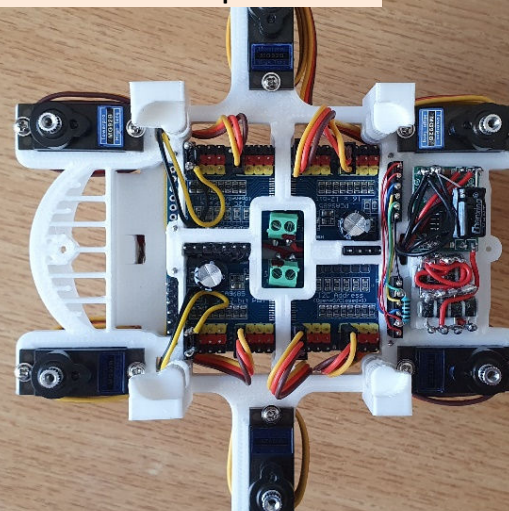
## Build Images

28 Then wire in the button switches and MPU6050



XXX

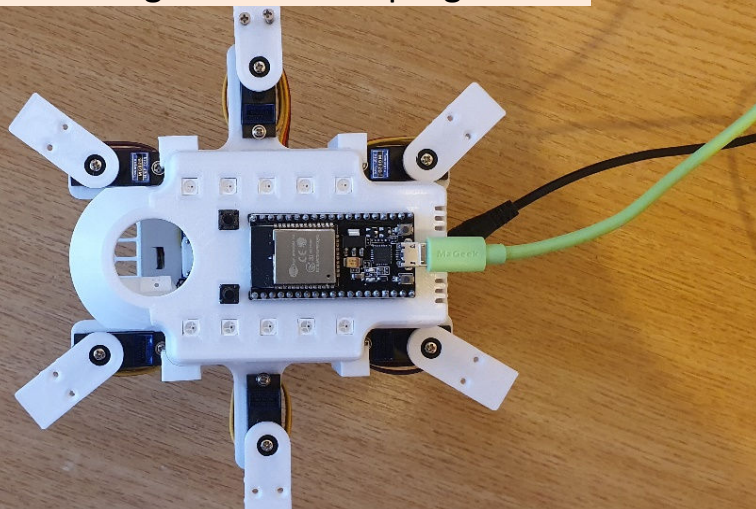
29 Then mount the MG92B hip servos.



29

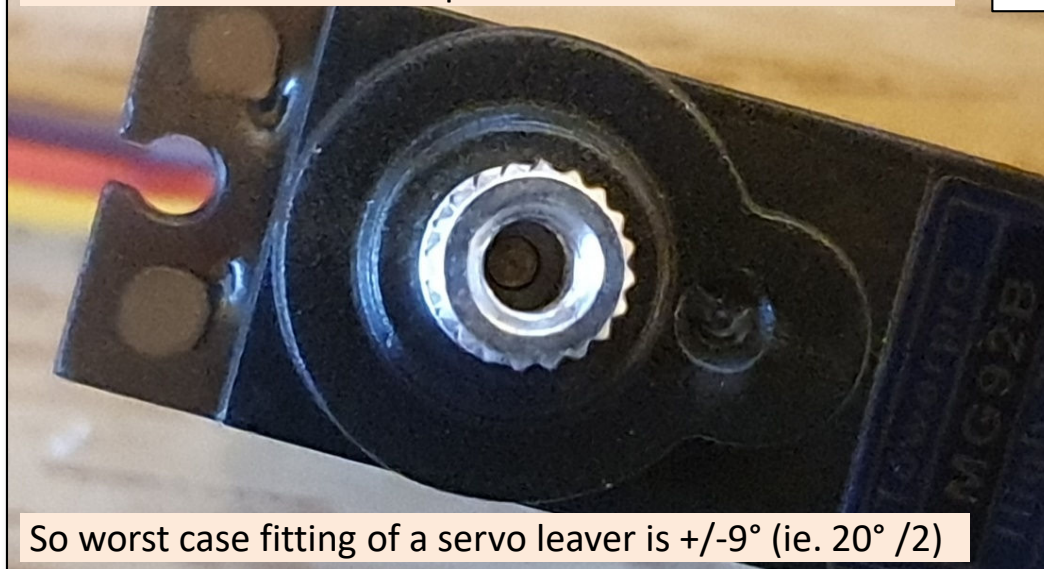
Plug them into the correct PCA9685 positions.

30 We can now test things as the build progresses.



Refer to calibration guide for servo arm fitting.

31 An MG92B servo has 20 splined teeth on its drive shaft.



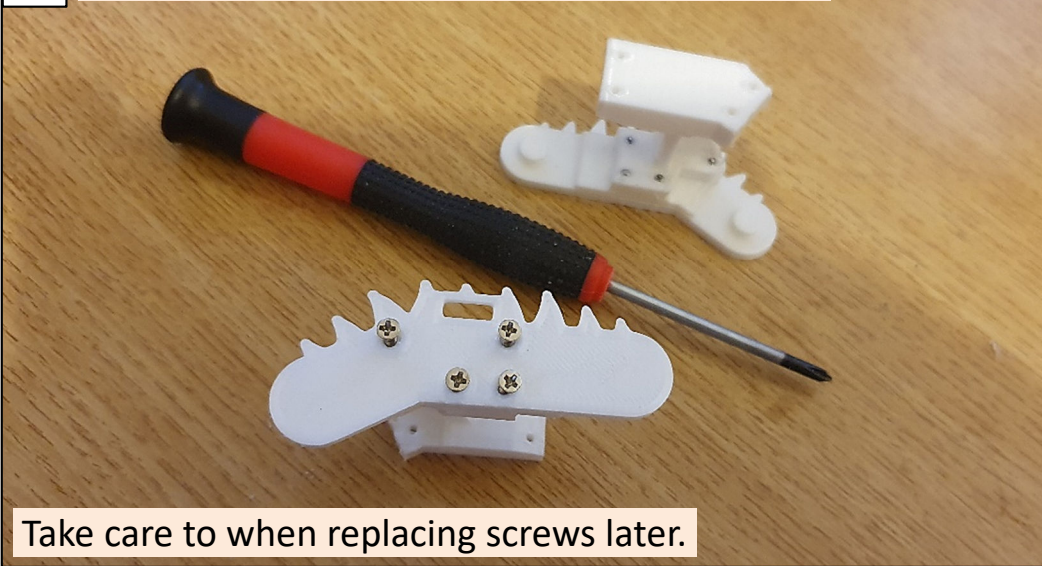
31

So worst case fitting of a servo lever is  $\pm 9^\circ$  (ie.  $20^\circ / 2$ )



## Build Images

32 Pre-assemble the legs, to ensure all parts fit.

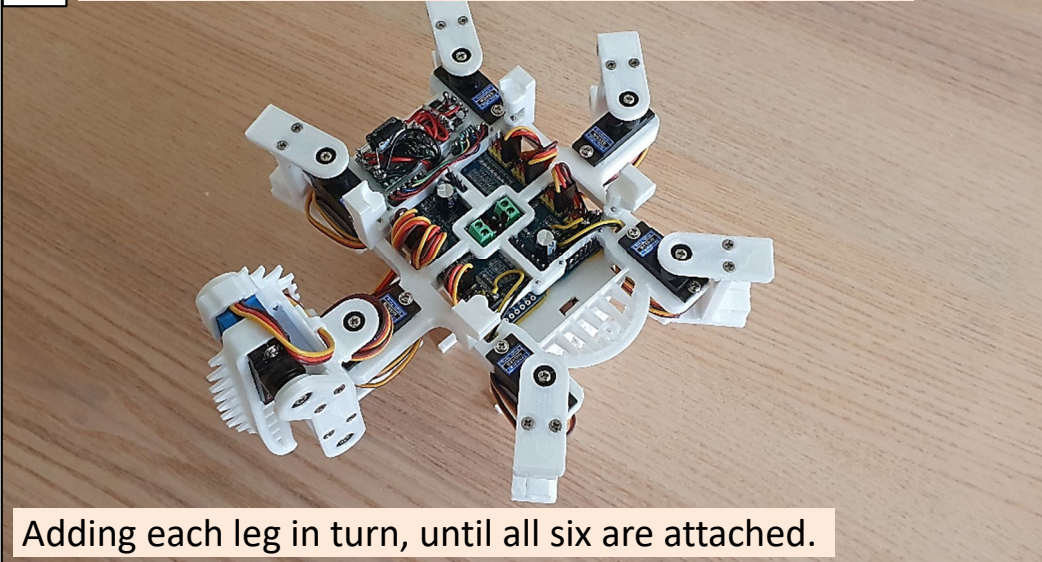


Do this for all leg components.

33

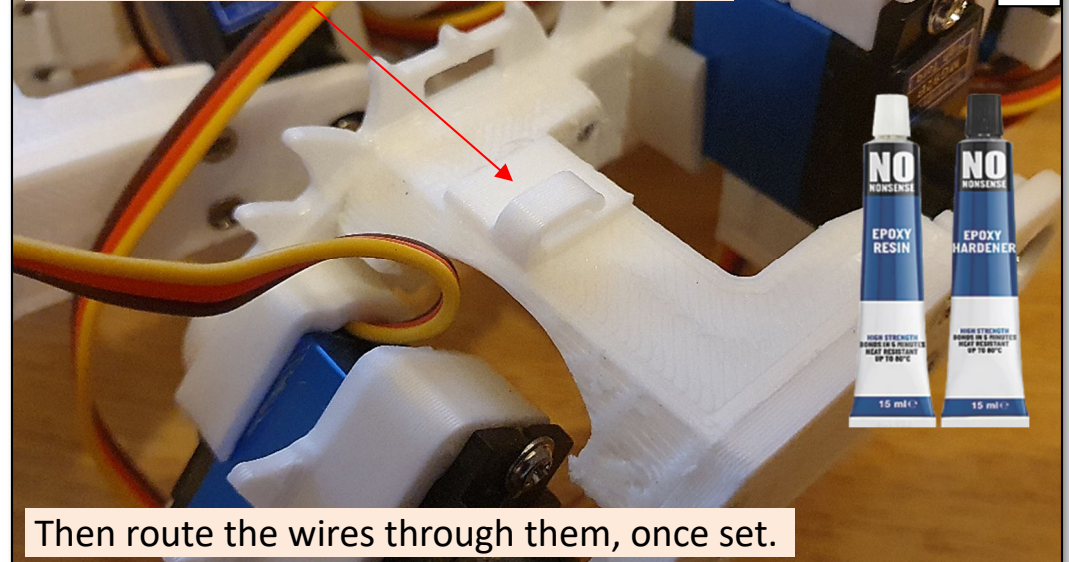


34 Progress the build and course calibration process.

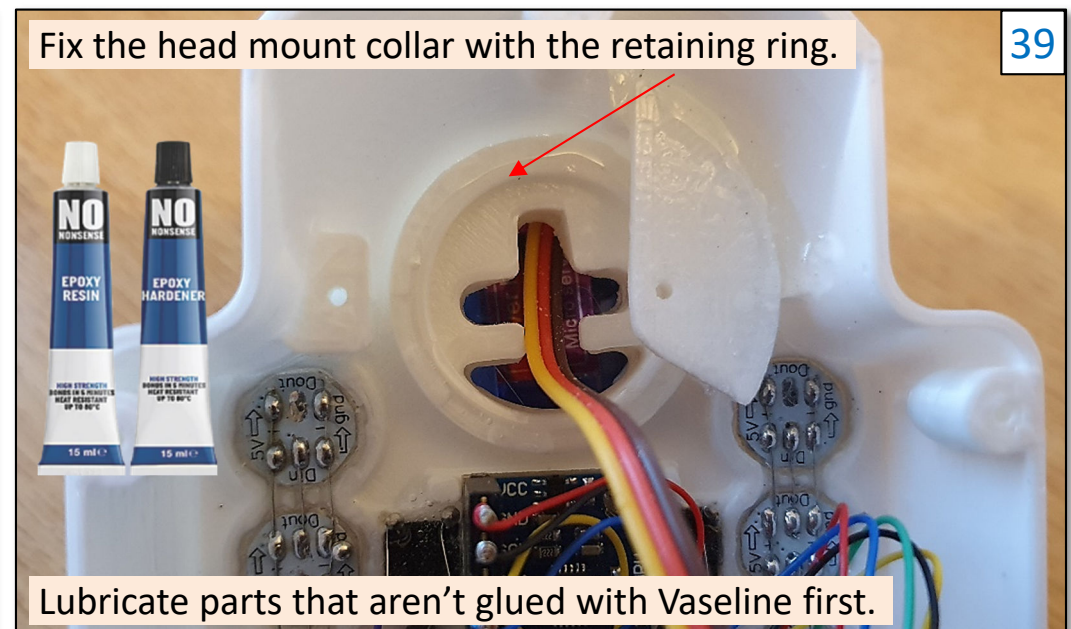
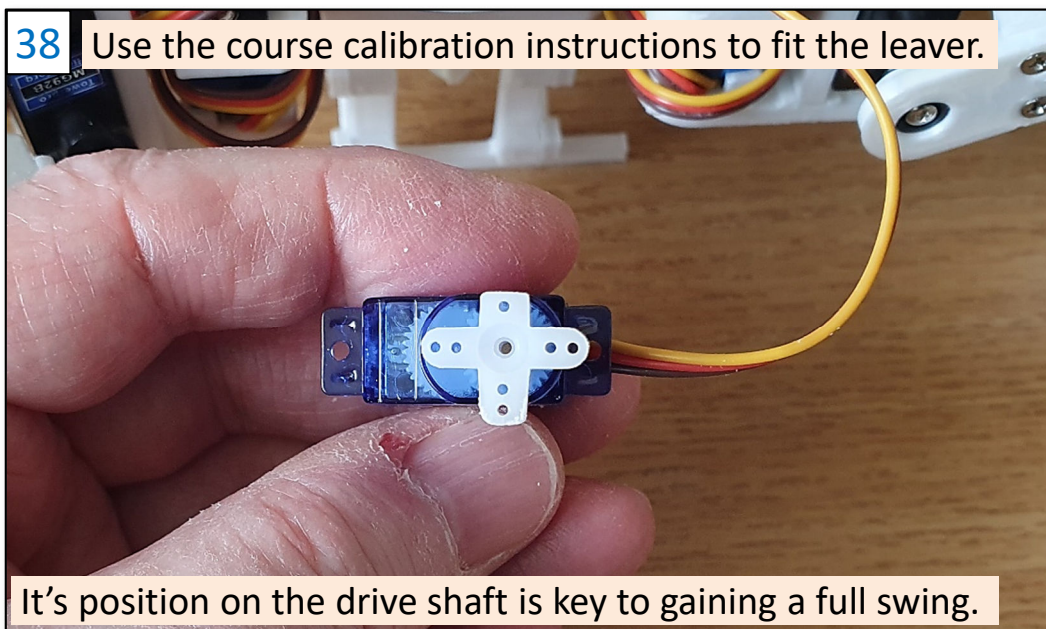
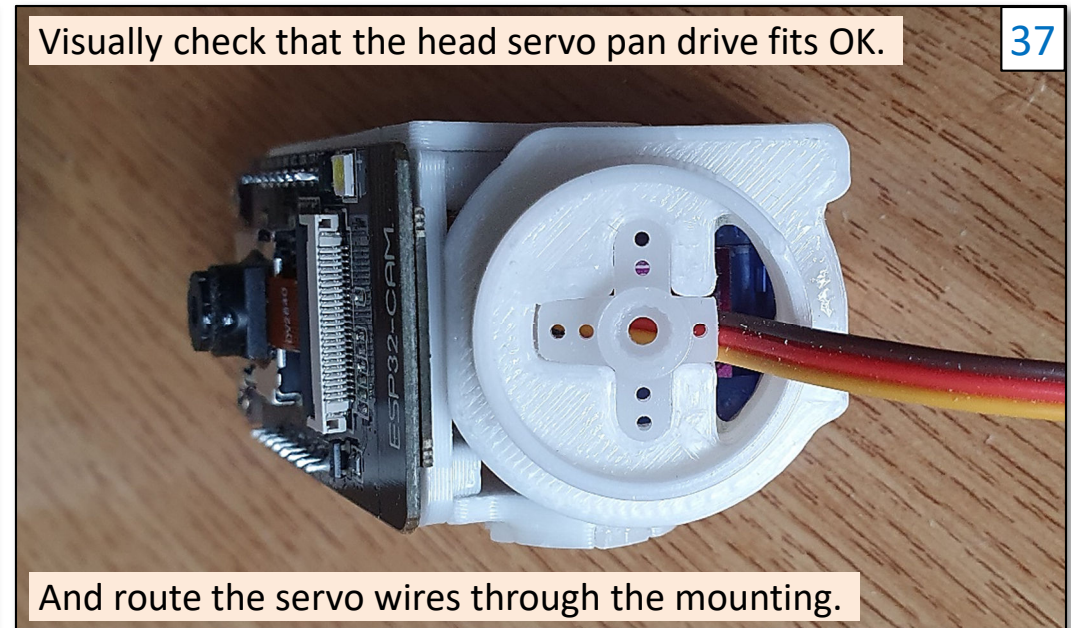
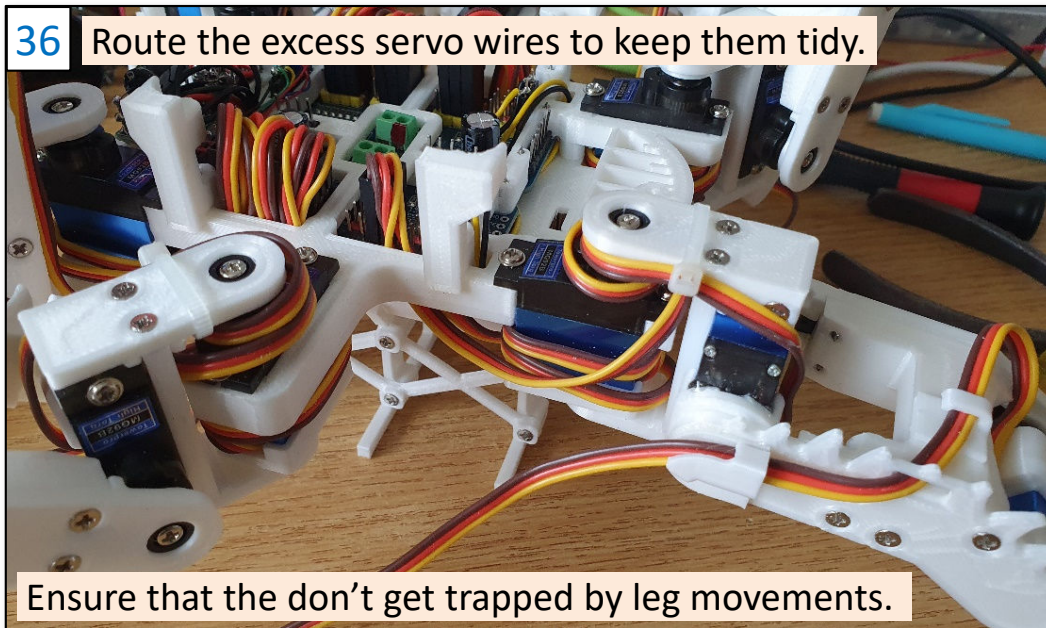


Glue the wire retaining clips into position.

35

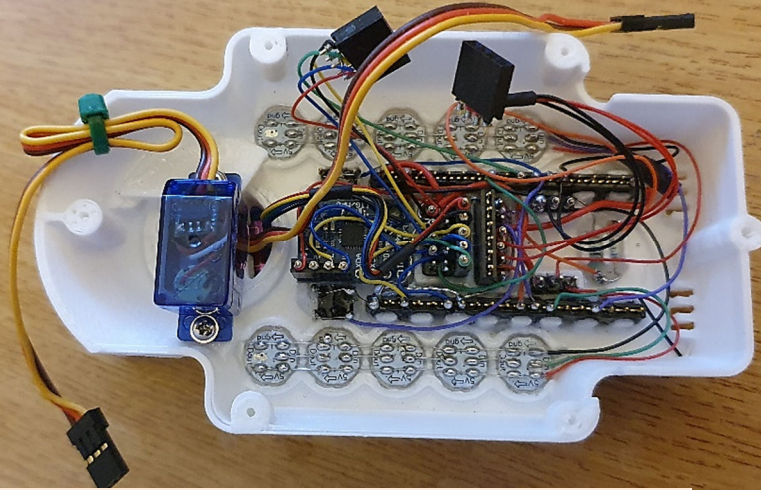


## Build Images



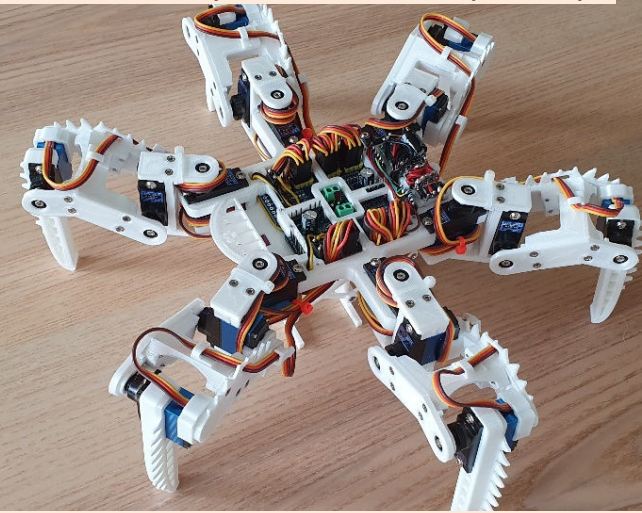
## Build Images

40 Micro plate with pan servo mounted.



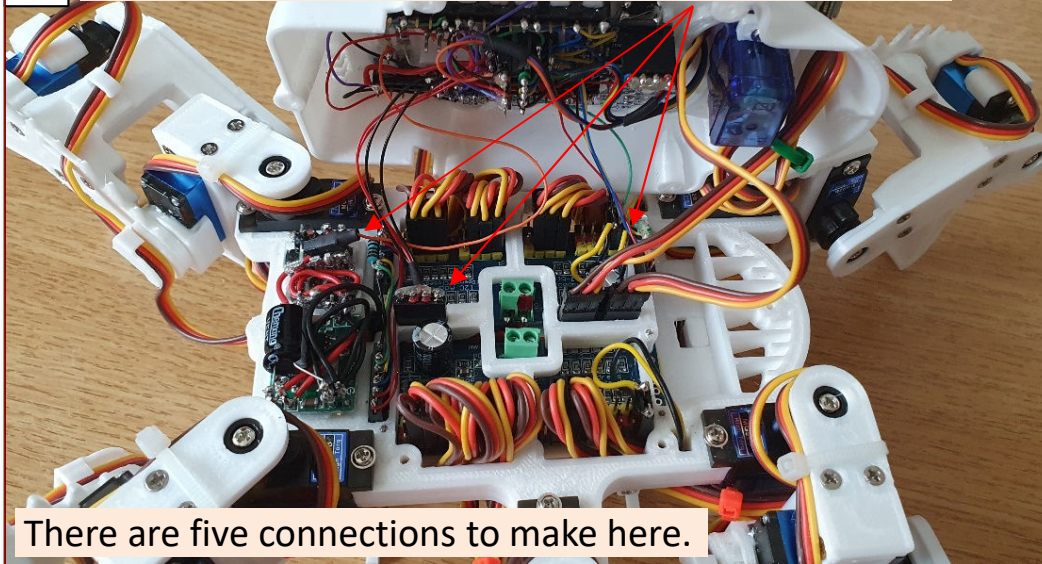
This is now ready to be attached to the base plate.

The base plate assembly is also wired up ready.



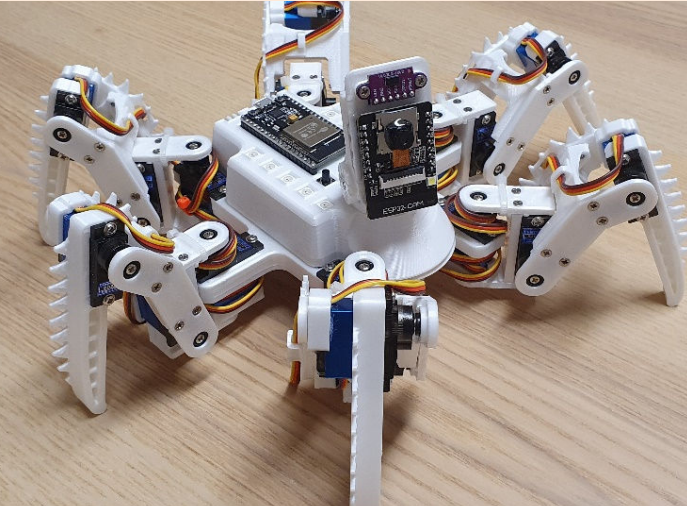
The two parts are held together with 2x10mm screws.

42 Bring the two halves together and plug in the leads.



There are five connections to make here.

All ready for the process of fine calibration to begin.



Use the Windows apps provided to complete this task.

# Battery Voltage Health Monitoring

See Lithium discharge curve obtained from the internet. In this analysis the lipo battery consists of three identical batteries connected in series. Assume fully charged 8.2v battery max voltage is  $V_{BM} \geq 8.4v$  max  
 Set battery warning point at  $V_B = 7.0v$  (2 x 3.5v)  
 Set battery critical point at  $V_{BC} = 6.6v$  (2 x 3.3v)

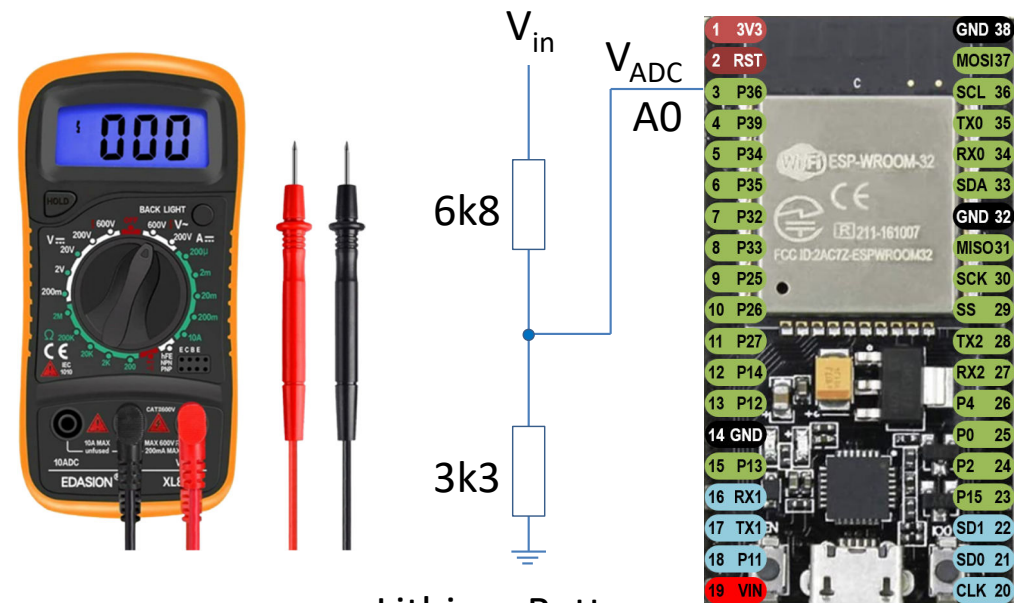
The ESP32 is powered via a voltage regulator connected to the 3.3v pin.  $V_{ADC} == 4095$  on 12-bit converter (4095 max).  
 If we use a 6k8 resistor feeding A0 and a 3k3 resistor to GND, we get a conversion factor of  $10.1v == 4095$ , or  $2.47mV/bit$ , or  $405.4 bit/v$   
 Using a Multimeter I determined the following  $V_{ADC}$  values for corresponding threshold voltages:

- MAX: (100%)  $V_M = 8.2v$ , gave A0 = 3032 on  $V_{ADC}$  (2 x 4.1v)
- HIGH: (80%)  $V_H = 7.6v$ , gave A0 = 2773 on  $V_{ADC}$  (2 x 3.8v)
- WARNING: (20%)  $V_B = 7.2v$ , gives A0 = 2611 on  $V_{ADC}$  (2 x 3.6v)
- CRITICAL: (0%)  $V_{BC} = 6.6.0v$ , gives A0 = 2372 on  $V_{ADC}$  (2 x 3.3v)

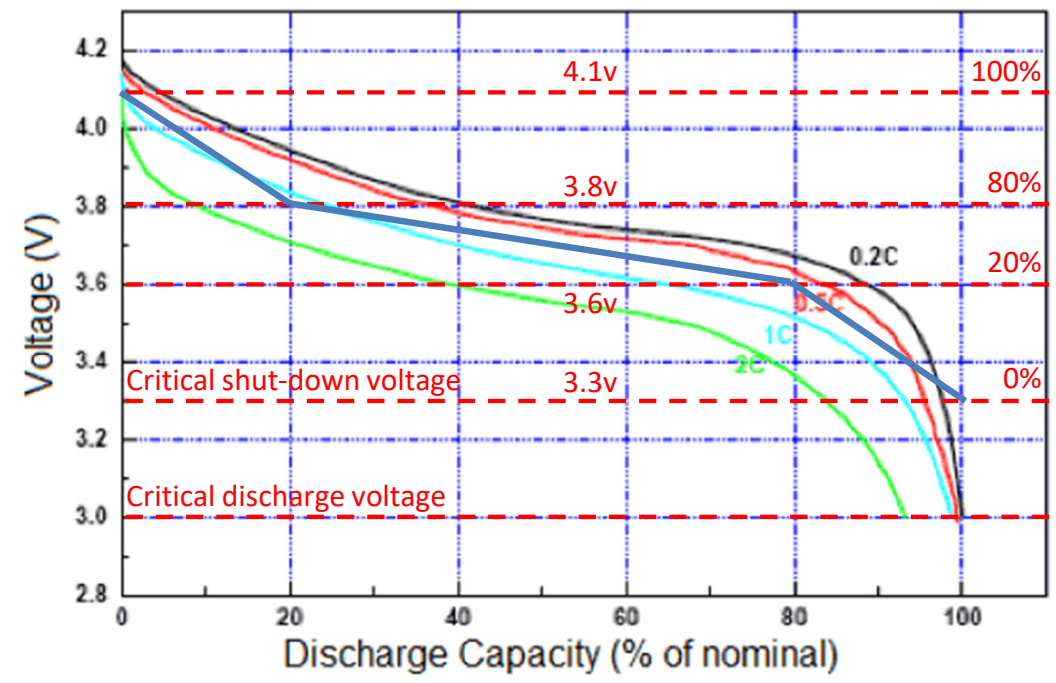
The code will sample the battery voltage on power-up to ensure it is sufficient, then at every 40ms interval, calculating an average (1/50) to remove noise. It also detects no battery as USB mode.

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.



Lithium Battery Discharge Profile



Discharge: 3.0V cutoff at room temperature.

## ESP32-CAM Flashing

To flash this device:

- Hold in the IO0 button before plugging in the USB cable.
- With US plugged in and power light on, release IO0 button.
- You should now be able to flash the part using the IDE:
  - Board: AI Thinker ESP32-CAM
  - See settings opposite

Connect using:

SSID: ESP32-CAM Access Point

Password: 123456789

Browser URL: <http://192.168.4.1>

### Arduino IDE settings:

Board: "AI Thinker ESP32-CAM"	▶
Port: "COM6"	▶
Get Board Info	
CPU Frequency: "240MHz (WiFi/BT)"	▶
Core Debug Level: "None"	▶
Erase All Flash Before Sketch Upload: "Disabled"	▶
Flash Frequency: "80MHz"	▶
Flash Mode: "QIO"	▶
Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"	▶
Programmer	▶
Burn Bootloader	

