# Mjolnir Assembly Instructions

By Thor Hansen, Ben Sanford & a lot of friends.



# 

# Contents

[Mjolnir Assembly Instructions](#)

[Contents](#)

[Foreword](#)

[Introduction](#)

[PARTS:](#)

[Introduction to Brushless Blasters 101](#) - For beginners

[Circuit Diagram](#)

[3D Prints](#)

[3D prints clean up, and Hardware Post Assembly](#)

[Electronics Pre stage, Firmware Flashing](#)

[SUB ASSEMBLIES](#)

[Battery Lid](#)

[Trigger & Micro Switch Sub Assembly](#)

[Magwell, Mag Release Leaver, Trigger Assembly and Grip](#)

[Magazine Detection - Micro Switch](#)

[Rotary Encoder Set Up](#)

[Oled Wires](#)

[Motor Mount Assembly – Top Half](#)

[Mjolnir PCB and Arduino](#)

[First Time power Up](#)

[Mjolnir – No PCB, Discrete Components Method](#)

[Buck PSU Module](#)

[Mosfet and Wires](#)

[How to Use OLED & Solenoid Calibration](#)

[Appendix A: Mjolnir PCB](#)

# Foreword

When Ben and I first started talking about this project, we wanted to make something which was for everybody and we wanted to make it brushless and controllable. We both felt inspired by the PewPew from Pewtech and wanted to make something similar

Neither of us have any professional electronic experience, so this information is all gathered on a hobby level and will have been proof-read by people with experience. Since this project wasn’t something easily obtained, we have needed a lot of help from friends and other modders in the Nerf community. This blaster is the end-result of that collaboration. By the time you read this, we will have released the files for both the firmware (Based on Airzone's brilliant code), the .step files, the wire schematics, bill of materials(BOM) and assembly guides.

We hope you will enjoy, improve and make it your own blaster – You are free to do whatever you like with the design.

If you have any questions, feel free to reach out to us on discord; <https://discord.gg/Edy7qAxdAY>

# Introduction

Mjolnir is a Brushless, Solenoid pusher, Micro controlled Blaster with Oled display and Rotary encoder.

It can be regarded as a medium to advanced electronics project depending on how you approach the build, however there are no reasons beginners cannot start here -this document is to also help beginners get involved.

Options to consider before starting

1. Mjolnir PCB module.

This has Mosfet and required PSU on it. This will be the easiest method for assembly and more suitable for beginners. It does require Manufacture or Sourcing of the assembled PCB.

1. Discrete components, Mosfet module and Buck PSU. This is an alternative method, if Mjolnir PCB is not available. Use this if you are happy with component level electronics

This is a more complex assembly method requiring more advanced skills.

1. Two Solenoid Options are available:

* Neutron from Out of Darts (Links in BOM)
* FJ-Z05

Project files needed from Discord or Github; <https://discord.gg/Edy7qAxdAY> /

- STEP file and/or STL files (3D print files)

- Code (Mjolnir code to load onto Arduino Nano)

You will also need

Arduino IDE - https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing

Blheli\_Suite or Blheli\_suite32 depending on which ESC you bought;

https://github.com/bitdump/BLHeli

Fusion 360

https://www.autodesk.com/products/fusion-360/personal

Basic steps

1. Study documents
2. Decide what options you will use, from above.
3. Source and order parts
4. Print 3D parts
5. . Prep 3D prints test fit and Heat Inserts
6. Electronics prep, Firmware Flashing
7. Assembly
8. How to use, Testing & Calibration

| **Mjölnir Base Components List for use with PCB option** | | | | |
| --- | --- | --- | --- | --- |
|
| **Project files on Github:** | [**https://github.com/Bsanford0916/Cross-Rough-Atlantic-Projects**](https://github.com/Bsanford0916/Cross-Rough-Atlantic-Projects) | | | |
|  | | | | |
| **Item** | | **Qty** | **Notes** | **Links** |
|  | |  |  |  |
| **Hardware** | |  |  |  |
| M3 screw 6mm (*Button Head*) | | 8 | Location??? | <https://a.co/d/f6d6F2C> |
| M3 screw 8mm (*Button Head*) | | 2 | Location??? | See options for sizes in above link |
| M3 screw 10mm (*Button Head*) | | 7 | Location??? | See options for sizes in above link. |
| M2 screw 16mm (*Button Head*) | | 4 | 2 for oled | <https://a.co/d/7gcIDZh> |
| M2 screw 18mm (*Button Head*) | | 2 | Trigger, Micro Switch Mount | See options for sizes in above link. |
| Heat insert (M3 x 3mm Length x 5mm OD) | | 17 | See diagram for location | [Https://a.co/d/2s1DLBN](https://a.co/d/2s1DLBN) |
| Ballpoint pen spring | | 1 | Mag release |  |
| Magnet - 5mm x 2mm (*Battery Lid*) | | 2 | Battery lid |  |
|  | |  |  |  |
| **Electronics** | |  |  |  |
| Iflight Xing2 1404 4600kV | | 2 | Brushless motors | [Https://a.co/d/bhoQbUt](https://a.co/d/bhoQbUt) |
| Blheli\_32 35A ESC *(20A will work)* | | 2 | ESC |  |
| Arduino Nano (*Atmega328*) | | 1 |  |  |
| Solenoid 12V 2.5A (*FJ-Z05 or OOD Neutron*) | | 1 |  | FJ-Z05 Solenoid Link https://a.co/d/eShqyR8 |
| Rotary Encoder (*with push switch*) | | 1 | NO PCB version | KY-040 Without board |
| 0.96in OLED I2C (*SSD 1306*) | | 1 | Color one will work | https://www.amazon.com/SSD1306/s?k=SSD1306 |
| MicroSwitch 10A | | 1 |  | <https://outofdarts.com/products/10a-microswitch-cherry-db2-with-lever> |
| MicroSwitch 1A | | 1 |  | <https://outofdarts.com/products/micro-switch-1a-for-mosfet-builds> |
| Lipo 3s | | 1 |  | 650mAh 75c 3s confirmed |
| 22AWG wire | | N/A |  |  |
| 28AWG Wire | | N/A |  |  |
| XT30 connector set | | 1 |  |  |
|  | |  |  |  |

# PARTS:

**Introduction to Brushless Blasters 101**

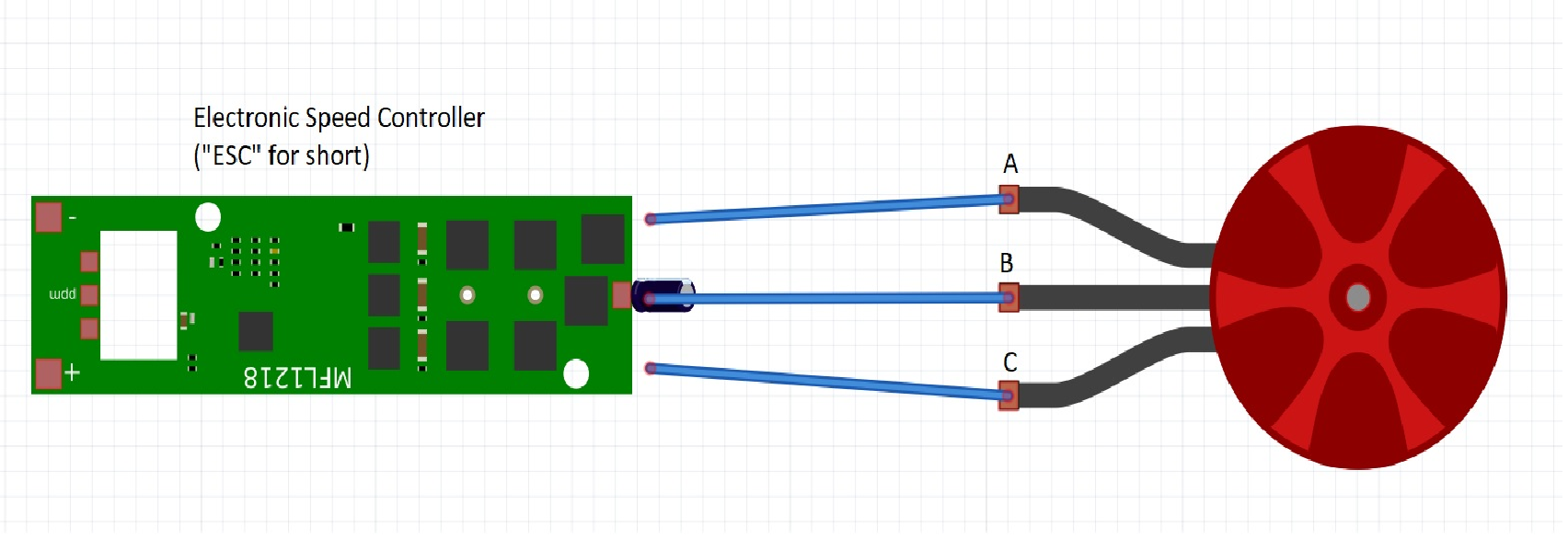
What is a "Brushless Blaster"?

A brushless blaster refers to the main mode of launching darts. In traditional blasters, you'll either have a spring based platform or a motor based platform. The motor based platform utilizes two flywheels to compress and fire foam darts. The motors are usually brushed direct current(DC) motors, which you know from standardized RC cars etc. These motors are running off a direct current making two magnetic fields and rotating the shaft. The brush of a DC motor is a component which provides a conducting path and will be worn down by time.

In brushless motors, you'll have several magnetic fields activating at different times to rotate the shaft. These motor conducting paths don’t physically touch the rotating parts.

Brushless motors are more commonly known as drone-motors, because they are the same motors used in high-end drones.

**Brushless motors comes in two variants:**In-runner variant: The rotating bit is inside a shell and the only rotating thing sticking out is the shaft.

Out-runner variant: The entire outer shell is rotating with the shaft as well! This is great for us, because it allows us to disperse heat faster. We will be focusing on brushless outrunner motors.  
Using brushless motors also complicate things a bit, because you need something to control when the fields activate, which leads me to the first schematic:

If we start on the right hand side, that’s the motor. It has 3 wires coming out and none of them is going to power like a normal DC motor. For simplicity's sake, I’ve called them A, B and C. It doesn't matter which end you'll start naming them, as long as the middle one is called B (I'll get back to that in a second). The wires are going directly to the Electronic Speed Controller(ESC).

The electronic speed controller is what controls which magnetic fields activates at what time and makes the brushless motor work.

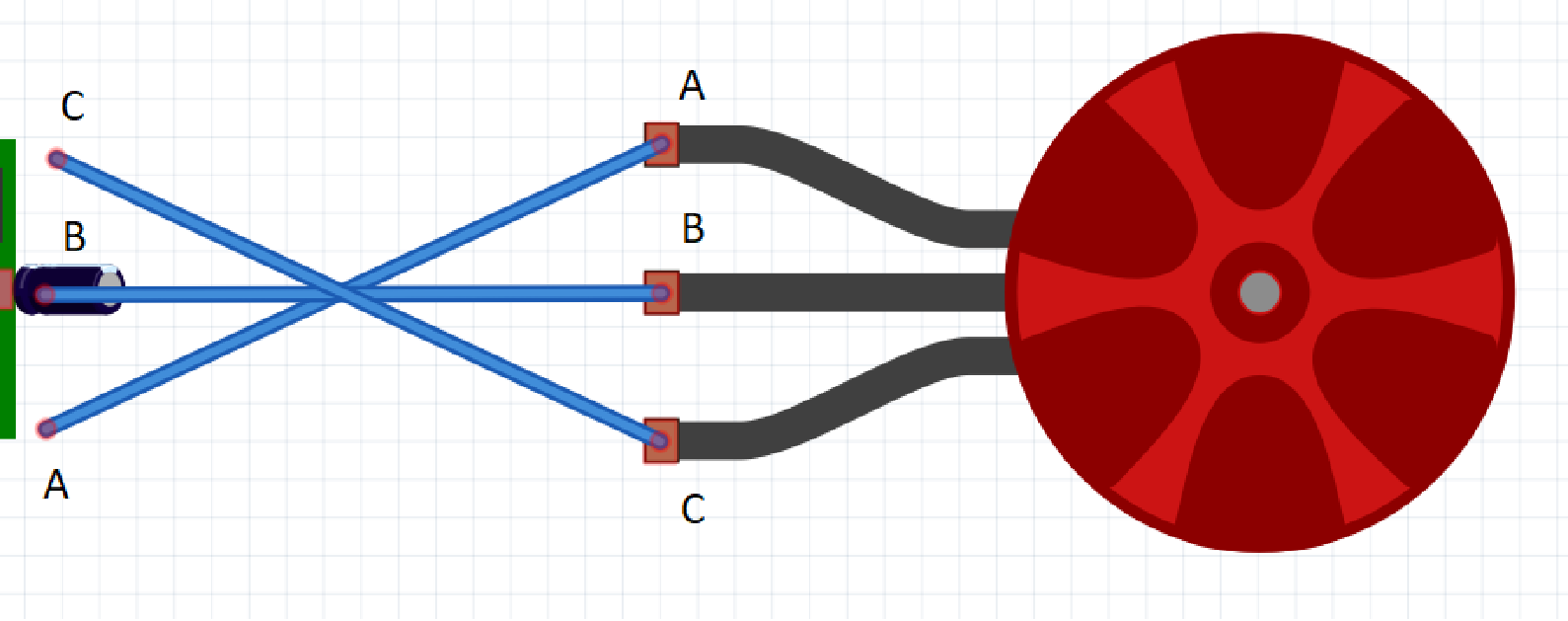
Brushless motors *do not* work without an electronic speed controller and you need *one* ESC per motor.

So, back to the wires. So you've got 3 copper pads at one end of the ESC, which is ment for the motor wires. The middle one always belongs to the middle brushless wire.

For the sake of the example, we'll attach wire A to pad A, wire B to pad B and wire C to pad C. If you had a motor rotating in the wrong direction, you'd have two options to fix that.

A) You can digitally tell the ESC to change the direction of the motor (See preparation guide).

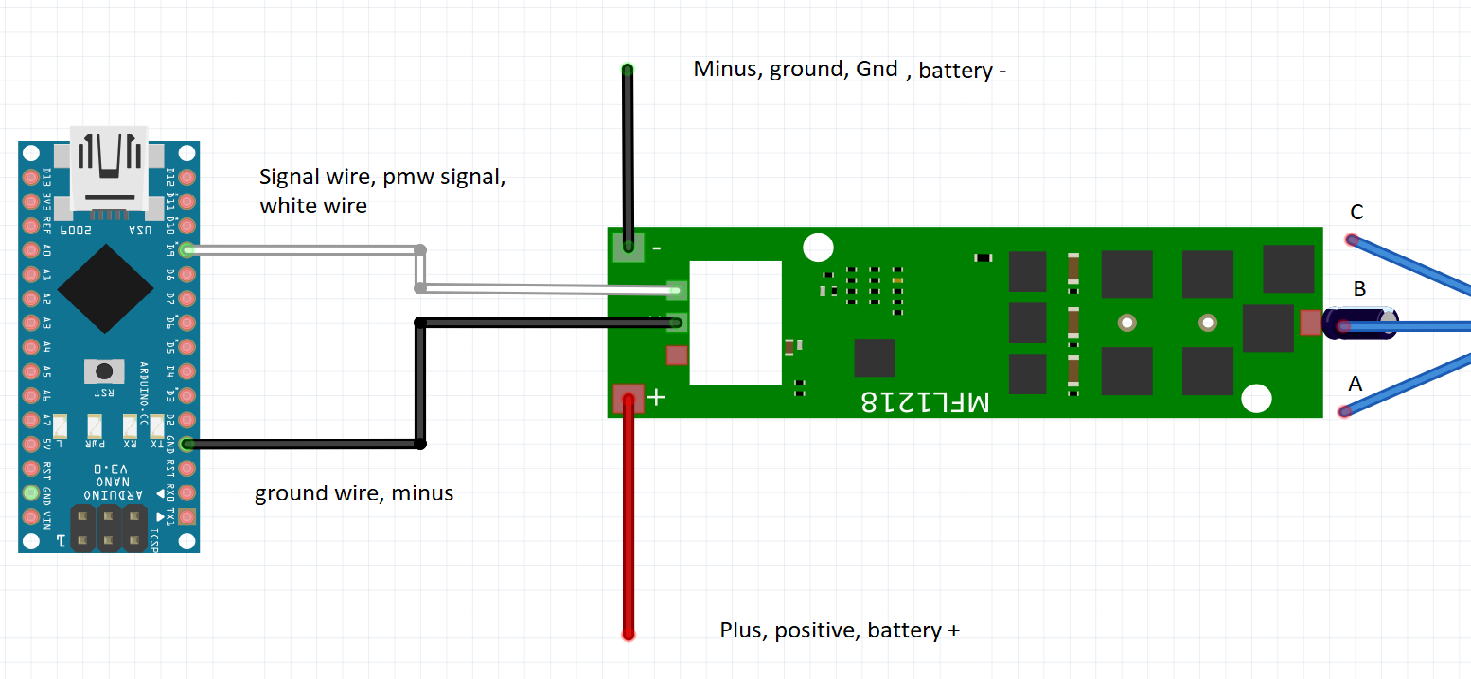
B) You can change direction of the motor spin by switching the A and C wires on the pads, so instead of A-A and C-C, it would be A-C and C-A.



At the other end of the ESC, there will be two pads and 2 wires. The two pads are for plus and minus(ground,gnd) off the battery and will have to be the same as your lipo battery (11,1v or so).

The two other wires are for the control of the electronic speed controller(!). There should be a signal wire, usually white or named something like PMW and a black one called gnd (for ground.) These should go to the micro control unit – In our case, an Arduino Nano 3.0.

*Coffee break and look at next schematic*



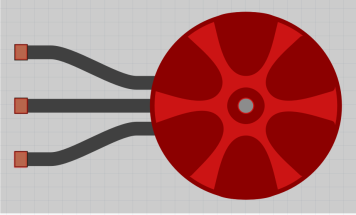
The arduino is basically the brains of the operation. It's where we have the code for what happens when you push X button and what the blaster should do if the battery gets low etc.

So in schematic, the arduino sends a signal to the ESC, which tells it to make the motor go.   
That signal is called a PWM signal, which is short for pulse width modulator. To simplify it, it's a matter of how many small spikes per second it sends and for us, the range is 1040 to 1960 hertz (hz, spikes per second). So if our PWM signal is 1040, the motors would be spinning slowly and 1960 they would be spinning a maximum of speed.

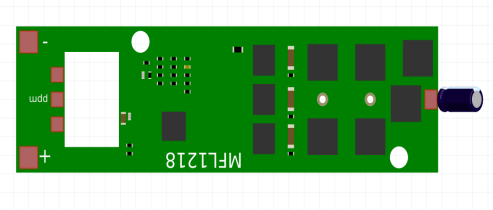
Now, all we have to do is to program the arduino to send these signals when we push the trigger, but don't worry – We've had help to do that for us.

So now you know how a brushless motor works. Well done. I've made a picture dictionary, so you can look at the wire diagram to see what the different items mean and what they do.

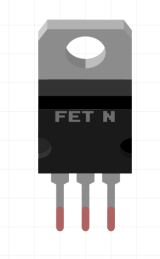
Components/Dictionary of the Mjölnier Brushless Blaster:

Brushless motor: 

Powerhouse of the blaster. We used Xing2 1404 4600kV motors. 14 refers to the stator diameter in milimeters. The 4600kV is the motor contant and this number multiplied by the voltage will give you the rounds per minut of a brushless motor. (Lipo 3s 11,1V x 4600kV = 51,060 RPM)

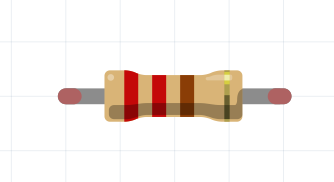
Electronic speed controller:

Used to control a brushless motor. Recieves signal from the arduino, which in turn controls its output signal. The usual ESCs have blheli S or blheli\_32 firmware. Blheli\_32 being the newest generation and the one with lowest latency.

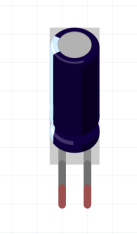


Mosfet:

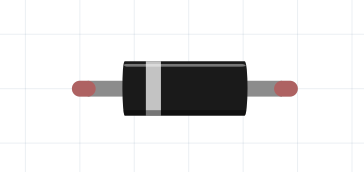
A digital button. Used to let the arduino control high voltage, even though it can only survive low voltage. Arduino sends signal to mosfet, mosfet opens the floodgates. Comes with 3 pins in this order: Gate-Drain-Source.

Resistor: 

Adds resistance to a circuit. Used by us to limit the amount of current going into our arduino. Unit is ohm.

Capacitor: 

Used to stabilize a power current. If the power is suddenly dropped for a mili-second, this will allow a unit to keep functioning for a short while. This could be when the motors rev up or likewise. We use them to keep the arduino and the ESC's alive while the solenoid and motors are firing.

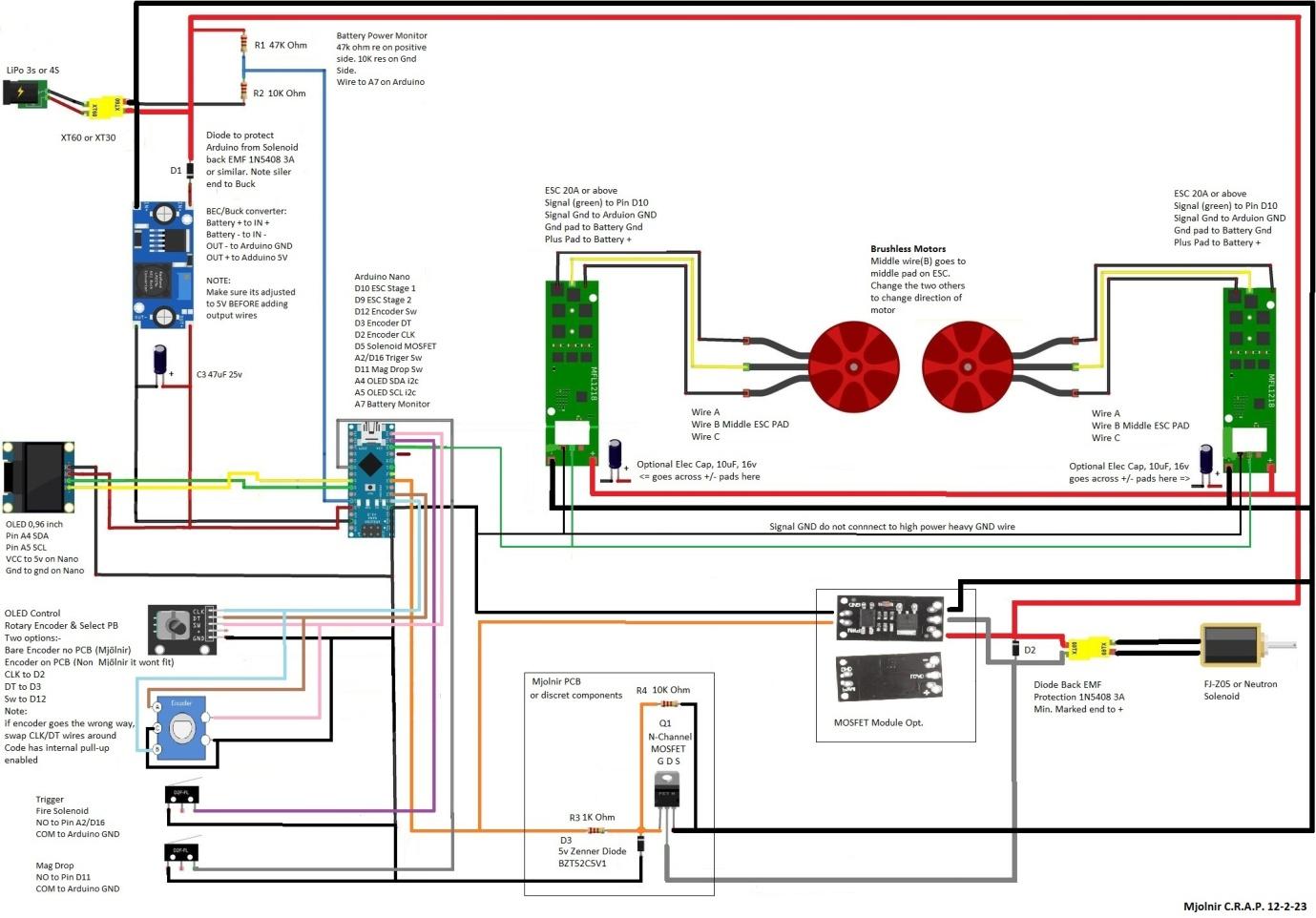
Diode: 

Used to make sure a current only moves in one direction or to make sure that if a current exceeds a certain point, that it will be moved away from fragile components – Which is also why we use it.

Solenoid: 

An electro-magnet which creates a magnetic field, which causes an iron rod to move. It has a spring in one end, which forces the iron rod back when there is no power on it. We use this to push darts into our flywheels. The current solenoid used by us is the FJ-Z05 12v, 2A.

# Circuit Diagram



# 3D Prints

You may have received files as a single STEP or individual STL files. If you have the single STEP file you need to open it with a suitable CAD program. We recommend Fusion 360.

Separate parts:-

From Menu on the left side of Fusion360 screen, select a part name, Right click on the part name and select save as Mesh File, by default it will use that part name as a file name.

Open slicer program to orientate print and produce G-code for printer

This is a very nice link to Thor explaining the correct orientation for the parts, and what supports are needed.(<https://youtu.be/fCTrLMgNpHs>)

Small parts seem to print well together. Print larger parts separately to get a better quality print at the expense of longer overall print times

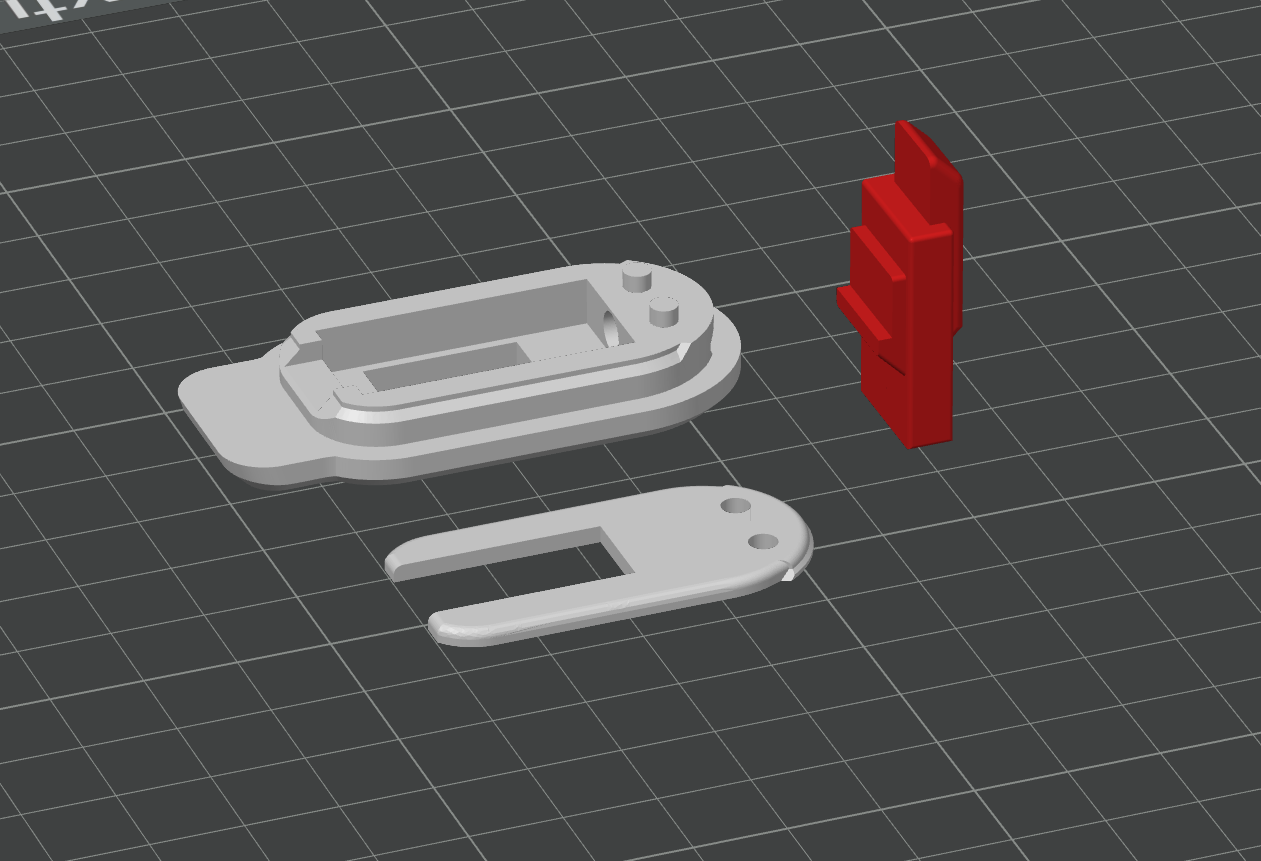
Recommended general printer settings:

* 4 walls
* 20% infill
* 5mm Brim
* 0.2 mm layer height
* 0.1mm layer height for flywheels only

Specific component printing suggestions:

* Fly Wheels: Print separately, set to random seam, and print upside down. 100% infill
* Trigger Guard: will need supports
* Magwell: will need supports
* Top Front: prints best upside down with support

**Print Orientation:**

Fig 1 Battery Lid 

Printing Orientation as shown.

No brim or supports needed

4 Walls, 25% infill for strength

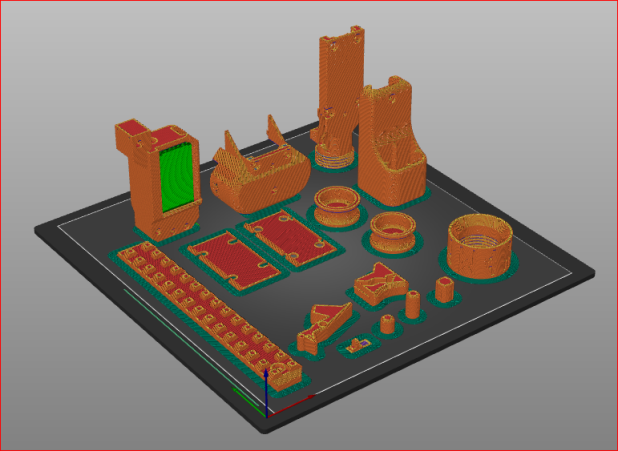
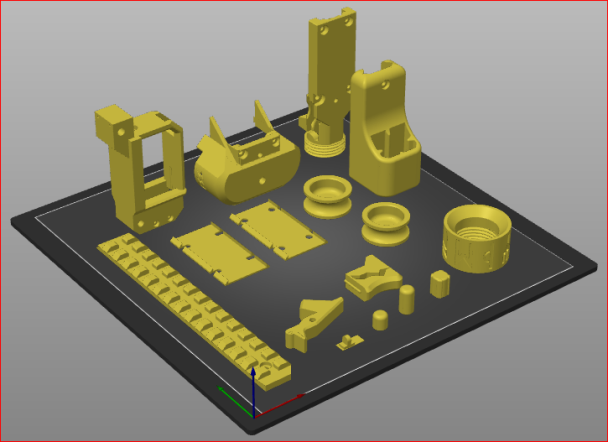


Fig 2 Smaller Parts: Suggest Brim, Trigger requires support (Green square)

Print Flywheels separate Using Random seam, 100% infill

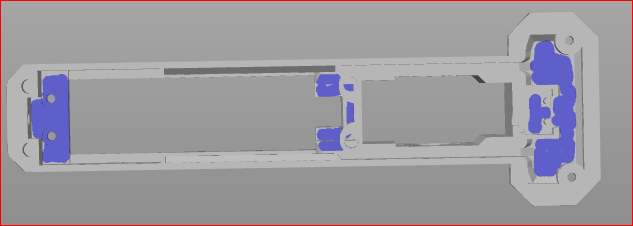
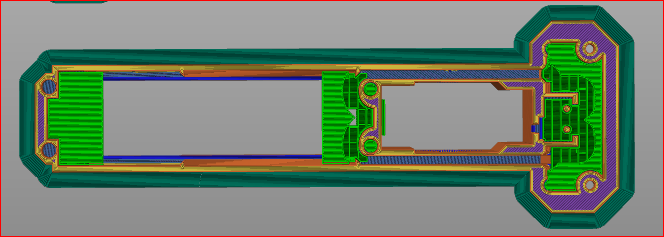
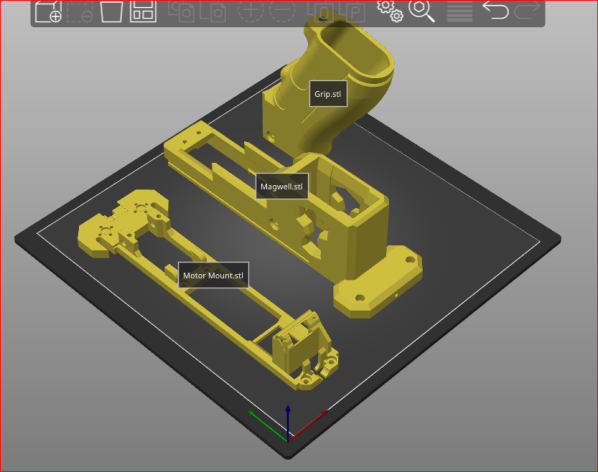


Fig 3 Larger Parts

Magwell prints better with supports; these are between print bed and ‘top’ of upside down part (green)

## 3D print clean up and Hardware Post Assembly

All prints will need some post assembly cleanup work. The more time and attention to detail here the easier it goes later on.

Use a sharp knife or deburring tool to scrape edges and smooth, especially where there has been a Brim.

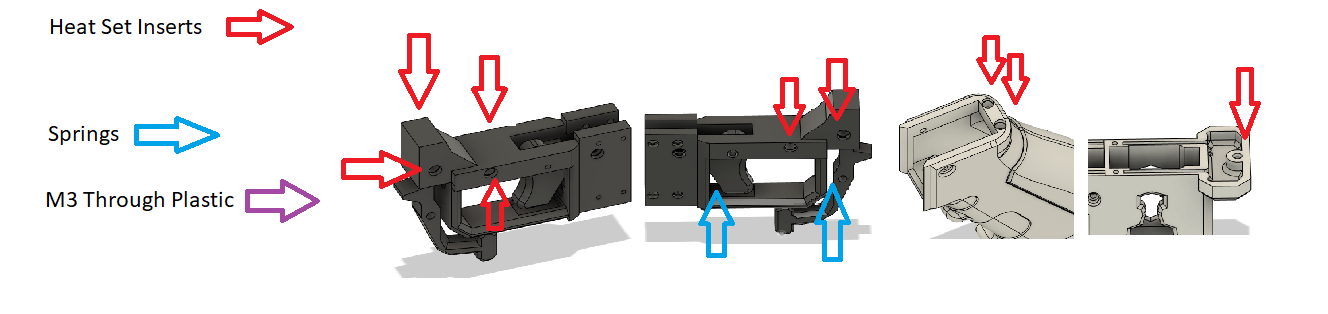
It’s easier to cut, file and sand early to clean parts up then part way through assembly.

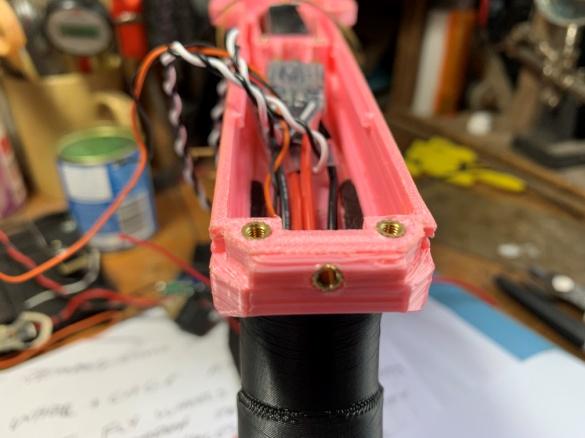
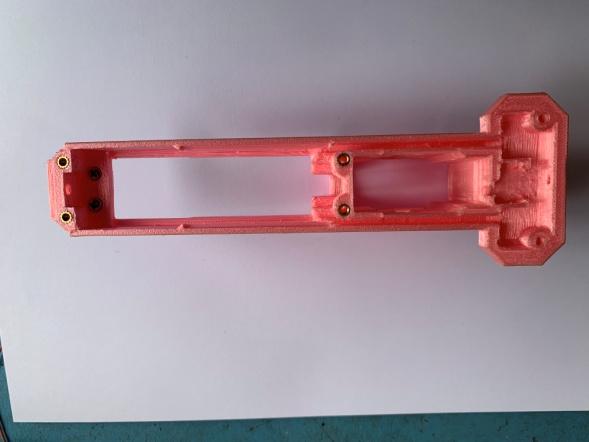
I use a craft knife/scalpel, and small set of jeweller’s files, to scrap printer slag off, and make things fit together nicely.

* Ensure all 3D printed parts fit together nicely, sand and file as required
* The Mag Well part will require a lot of attention
* Clear holes for brass inserts and screws, test fit. Compare to STL via Slicer program
* Check all parts fit in their spots, Motors, Solenoid, OLED, Encoder, Micro switches
* Check Trigger slides smoothly
* Check magazine goes in/out smoothly
* Check mag release movement

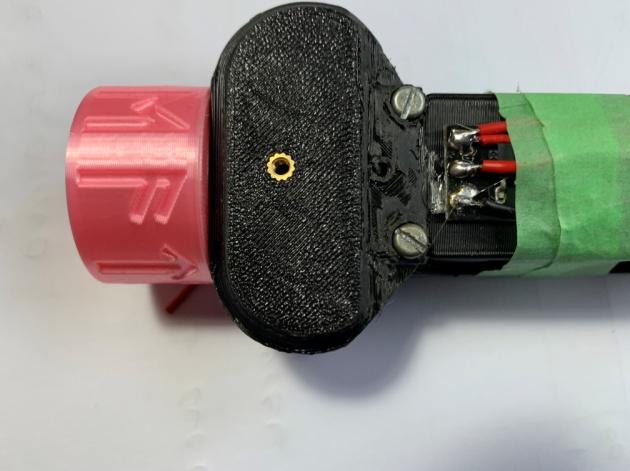
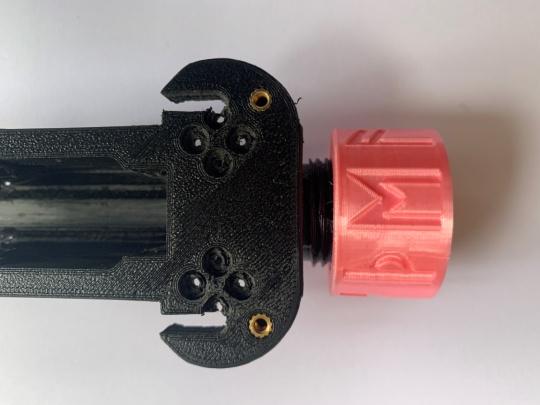
**Performance Tip:** pay particular attention to the inside of the dart guide. File and sand this smooth to ensure the dart has the smoothest surface to fly through and not lose FPS by having rough edges and printer lines.

This work can take some time, but you’ll later appreciate being patient and meticulous now.

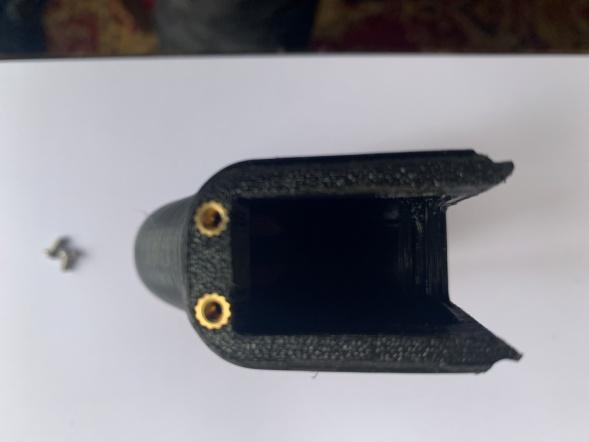
Please reference the following diagrams and links for Brass Inserts and Hardware setup.



Mag Well Inserts



Mort Mount Inserts Top Front Insert

Pic Rain Insert Grip Inserts

Thors: Where things go video (10min ) <https://www.youtube.com/watch?v=9kLfPOxWR3s>

Thors: Installing Brass Heat Inserts part One <https://www.youtube.com/shorts/OJXvqklRar8>

Thors: Installing Brass Heat Inserts part Two <https://www.youtube.com/shorts/8pyeaI89rY0>

# Electronics Pre stage, Firmware Flashing

This section will describe how to flash the ESC and Andruino Firmware

Find a micro-USB cable to connect Arduino nano to your computer. Be warned that micro-USB charging cables probably won’t work for this process as they don’t have everything needed to send data.

**Step 1: Flashing ESC Firmware. This is done *BEFORE* assembly.**

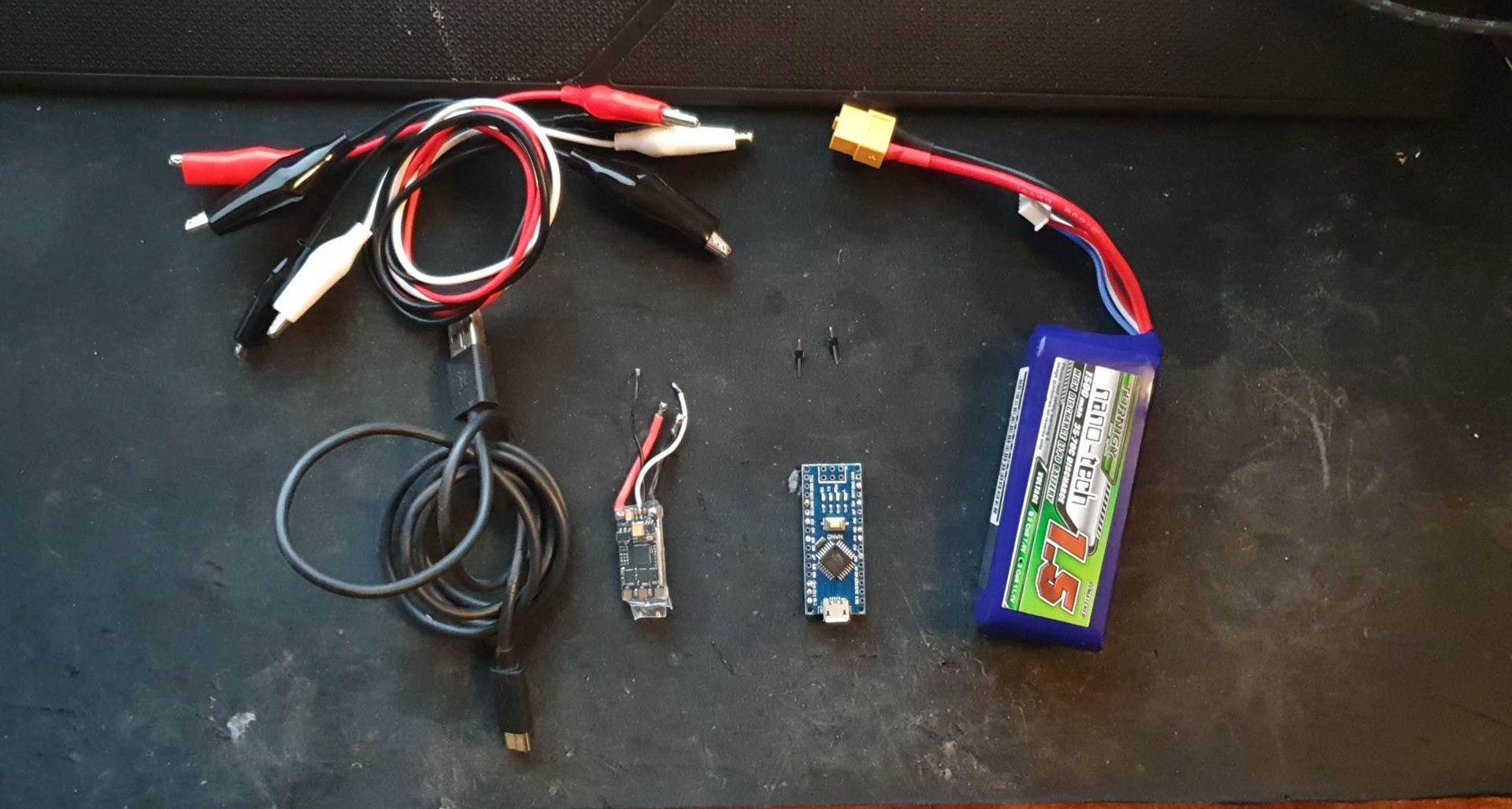
Depending on which ESC you got:

**Download Blheli S suite:** <https://github.com/bitdump/BLHeli>

**Download Blheli\_32 suite:** <https://github.com/bitdump/BLHeli/tree/master/BLHeli_32%20ARM>

(Additional information regarding the flashing of ESC's can be found here:  
Blheli S: https://oscarliang.com/flash-blheli-s-esc-firmware-fc-pass-through/

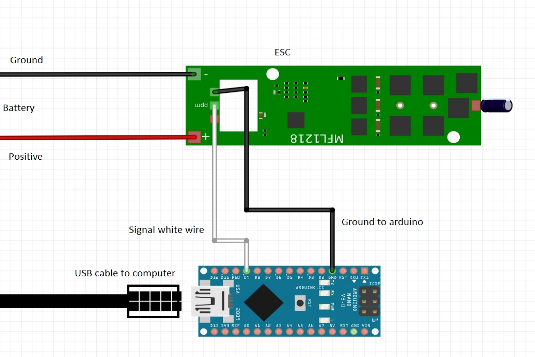
Blheli\_32: <https://oscarliang.com/blheli-32-esc-fc-passthrough/#download>)

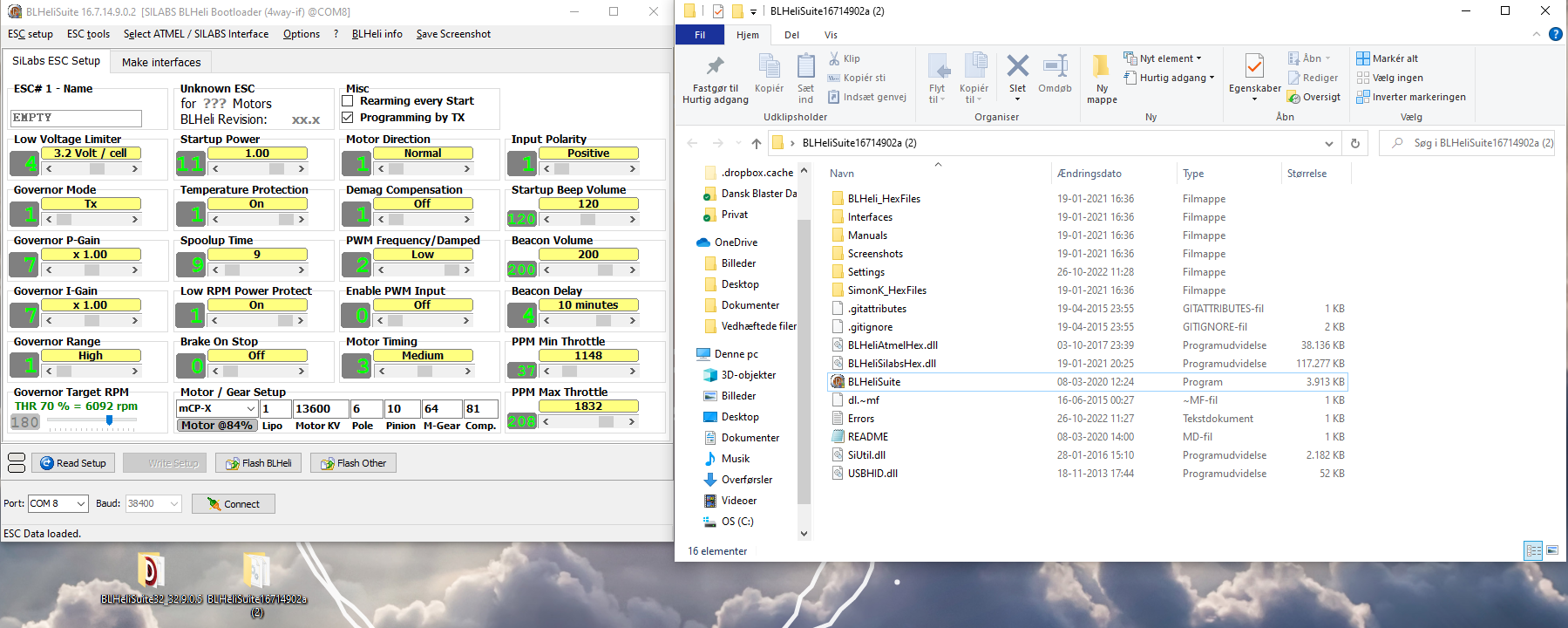


Components I personally use to flash an ESC:

* 4x Crocodile Cables
* 2x metal pins
* Arduino Nano
* ESC Blheli S or Blheli\_32
* USB cable
* LIPO 3s for powering the ESC

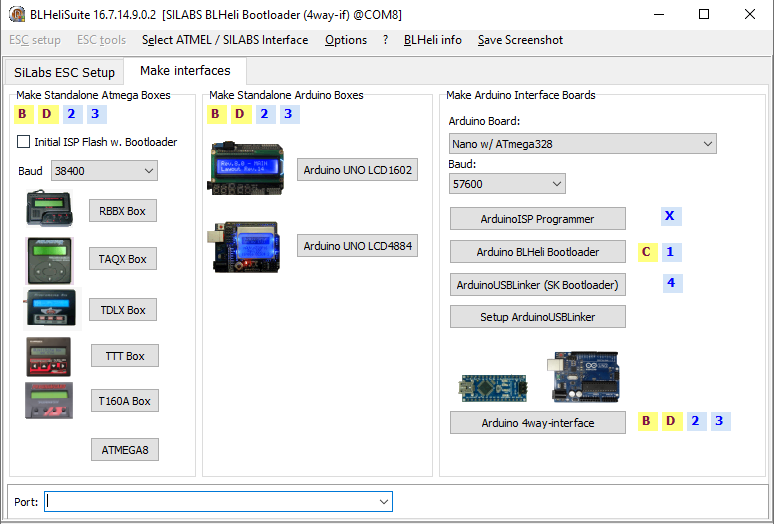
Instructions:

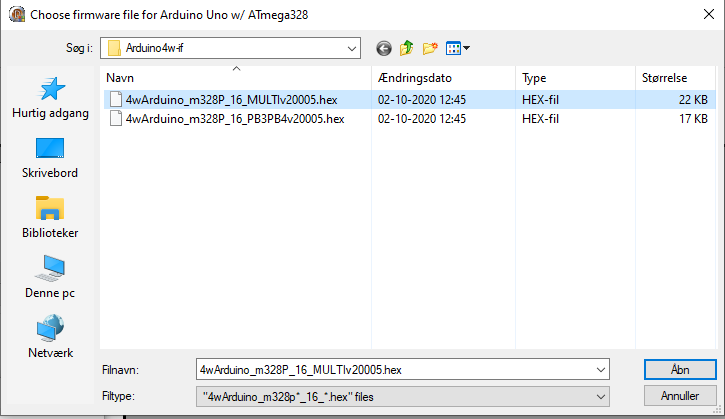
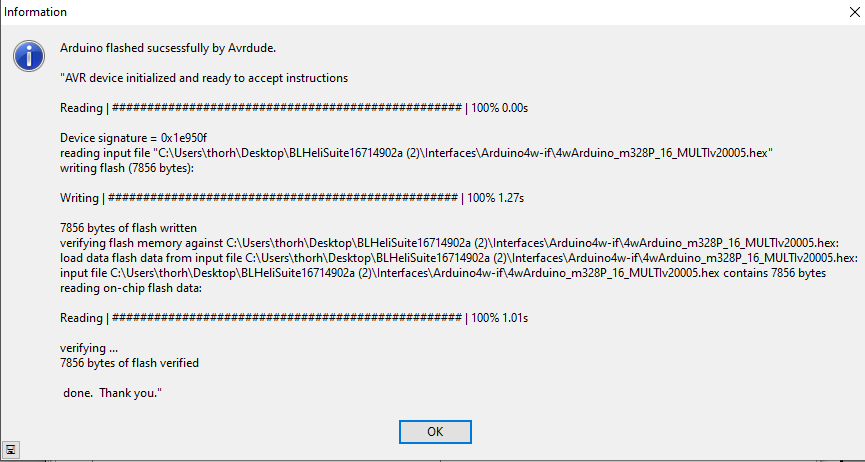
1. Place metal pins in **D9** and **gnd** on Arduino.
2. Use crocodile wire to connect signal wire from ESC to metal pin **D9**.
3. Use crocodile wire to connect ground from ESC to Arduino **gnd**.
4. Connect Arduino to computer (NOTICE: No battery to ESC yet!)
5. Open Blheli suite (Either the 32 or the S version, depending on your ESC.)

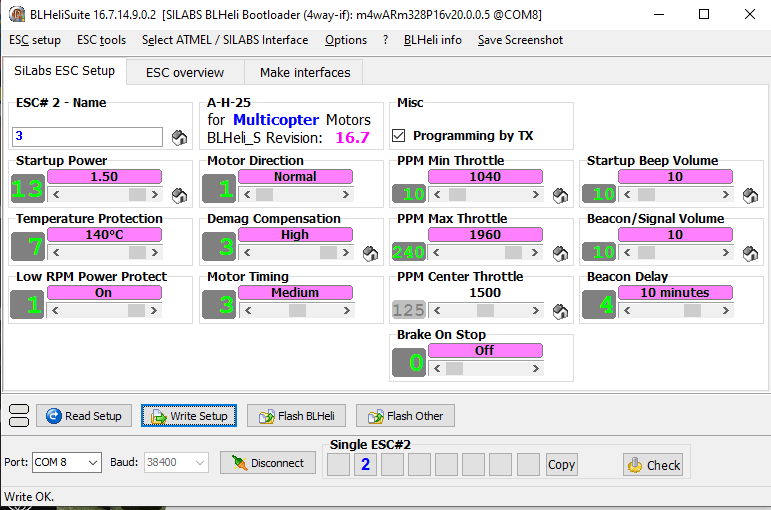


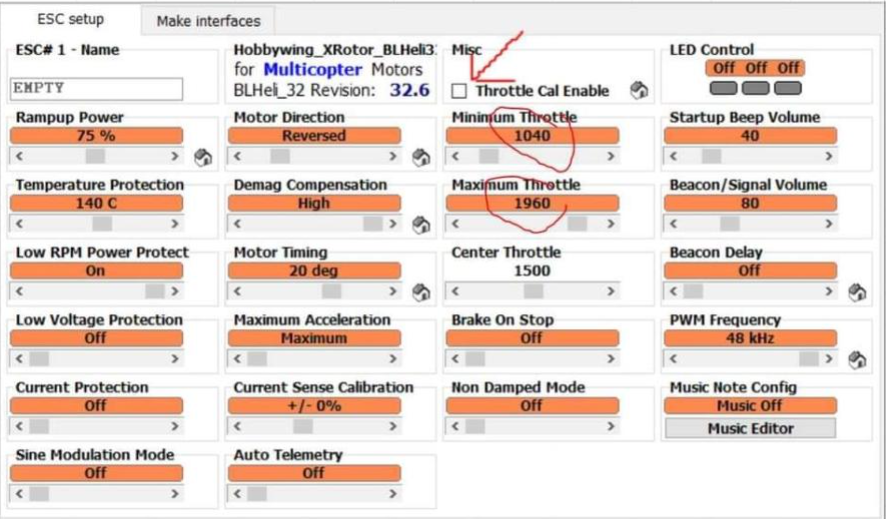
This is how Blheli\_S suite looks when it's just opened.

1. Click the PORTs dropdown at the lower left hand corner and choose the USB port which the arduino is plugged into.
2. Click "Make Interfaces".



1. Choose the Arduino Board used. For the quick people, they'll notice that I'm using an arduino Uno instead of an Arduino Nano. This makes no difference other than you have to pick the right board. I've shown the Arduino Nano board here, with the Atmega328 processor that we recommend for the Mjölnier blaster. NOTICE: I have not connected my usb correctly, so my USB port is not showing at the bottom of the program. This is a common problem with old USB ports. Double check your connections.
2. Press "Arduino 4way-interface"
3. I've picked this one multiple times with Arduino uno. I haven't had a chance to try this with a Nano, but I'll update as soon as I get a chance to do so. Update; Done with a Nano, so it’s doable.
4. Success!
5. Move back to the ESC setup tab
6. Connect 3S Lipo or 11.1v to the ESC’s positive and ground copper pads(Woo!). The red wire from the battery is positive, the black wire is ground. **If you connect this in reverse, you will ruin your ESC**.
7. Make sure it's still the right USB port and press Connect at the bottom of the tab.
8. Press "Read Setup"
9. Update the data to look like your Blheli ESC (S or 32). See pictures at the bottom of this step.





1. Press "Write Setup".
2. Tadaaaa! Well done, the first ESC is flashed.
3. Repeat for the second one with the exception of changing the “Motor Direction” setting to be a “0” so that the second motor will turn in the opposite direction..

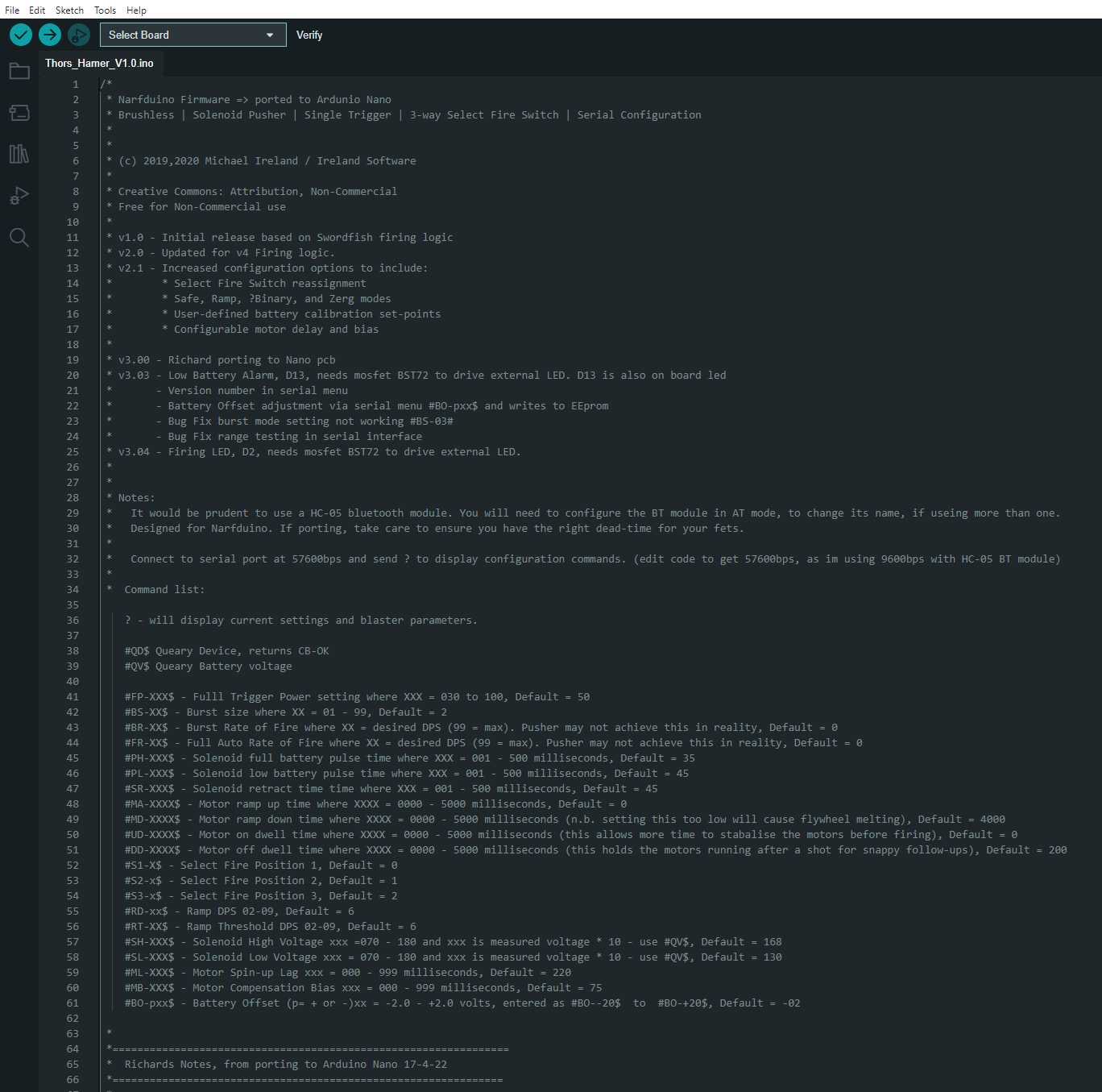
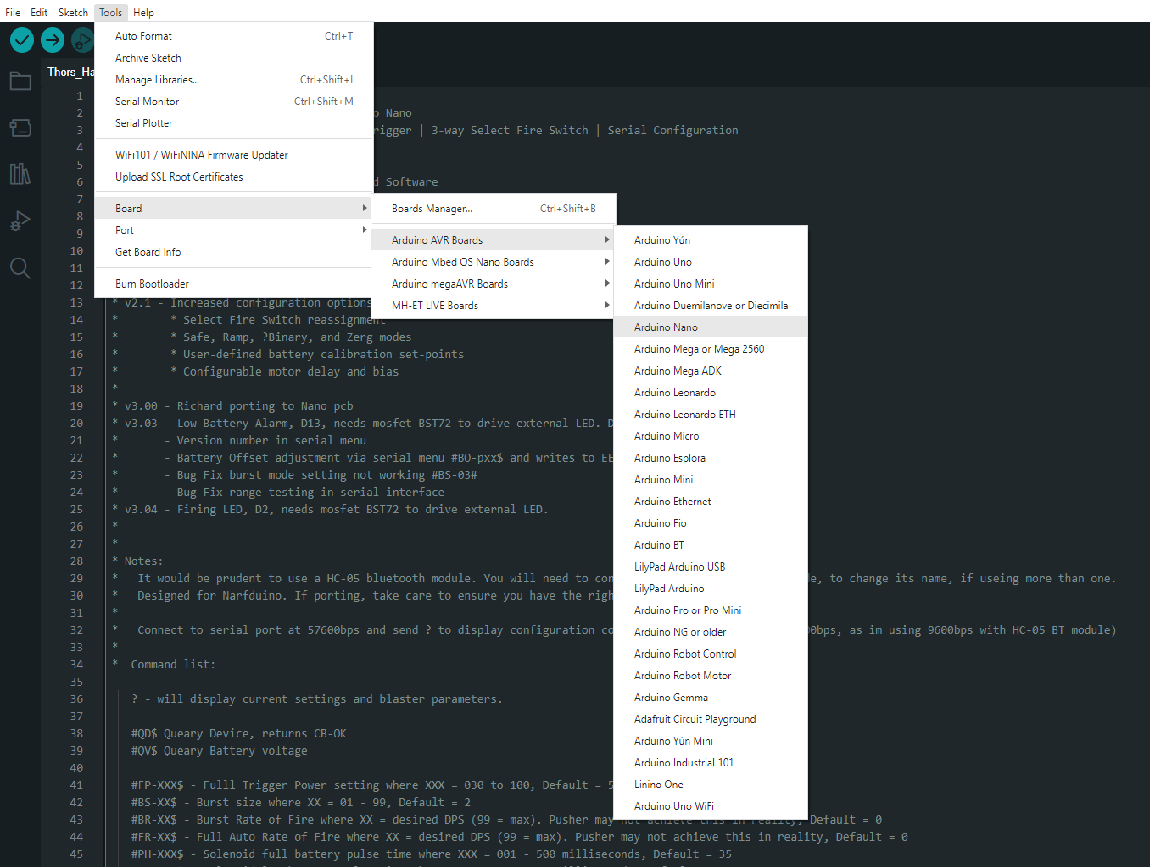
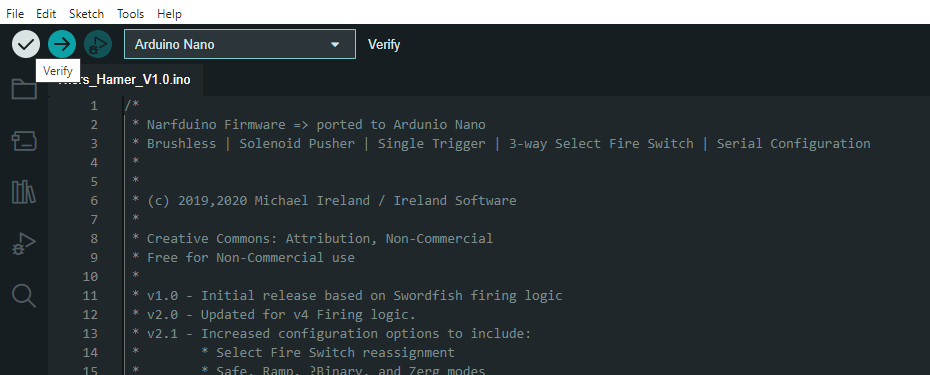
**Step 2: Flashing the Arduino Nano 3.0:**

1. Download the Arduino IDE and install the newest version.   
    Link: <https://www.arduino.cc/en/software>
2. Download the code, ensure these three files are in a folder named Mjolnir

Mjolnir.ino

Logo.h

MenuHelpers.h

1. And open it with the Arduino IDE.   
   It will look something like this: 
2. Plug your Arduino Nano 3.0 into the USB port using a USB cable.
3. Go to Tools --> Boards --> Arduino AVR Boards --> Arduino Nano  
   
4. Go to Tools --> Port --> Pick the port which the Arduino Nano is connected to.
5. Press Verify (Top left corner)  
   
6. Press Upload (right next to it)
7. Once the loading bar in the bottom right corner finishes and it says "upload completed" or so, you're done flashing the Arduino!

**Troubleshooting:**

* Try resetting the Arduino by pressing the button.
* Try to use the old bootloader instead
* Download the proper drivers from the place where you bought the Arduino Nano (different manufacturers can make boards requiring different drivers.)
* DOUBLE CHECK YOUR CABLES. Is it a data-USB cable? Is it loose? Is the USB port working?

# SUB ASSEMBLIES

## Battery Lid

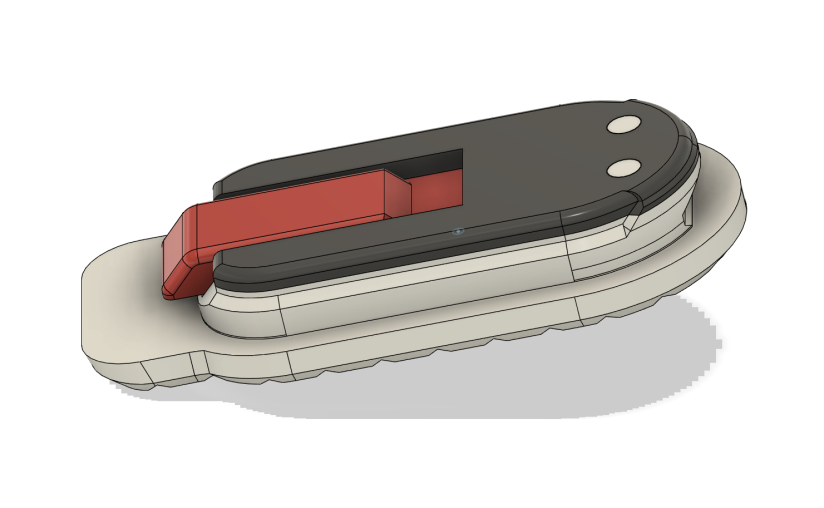


Fig 1 with all parts together



Fig 2 with the top removed

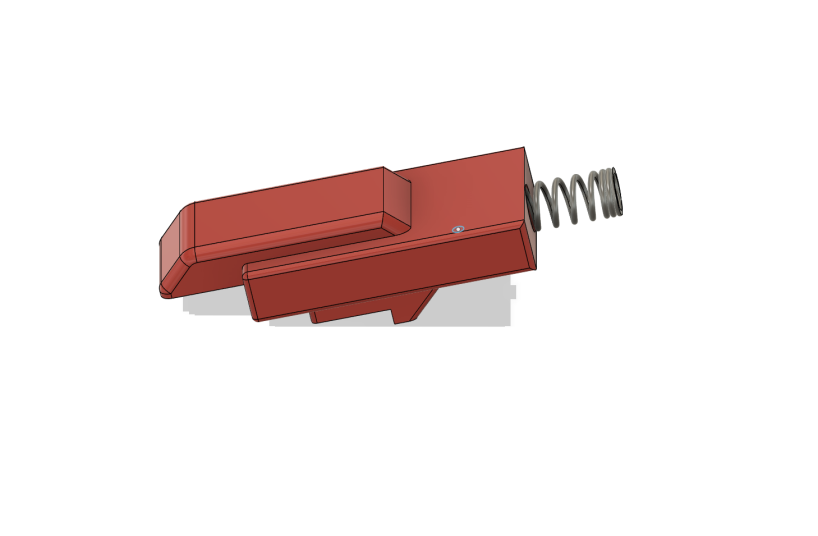


Fig 3 with just slider and spring (spring must be 4mm OD 15mm length)

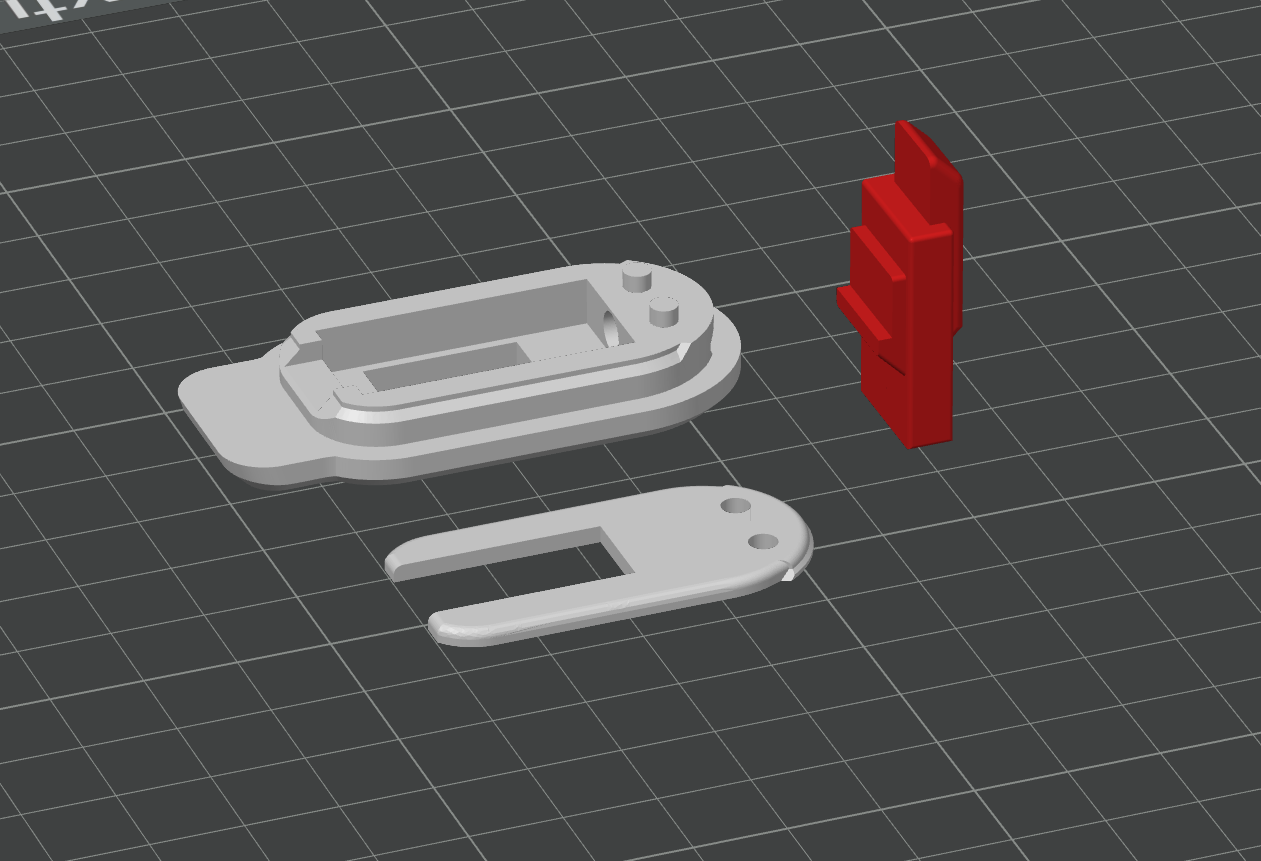


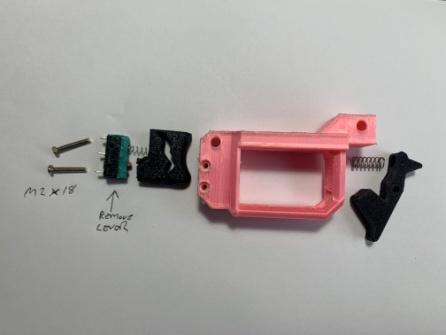
Fig 4 Printing Orientation as shown.

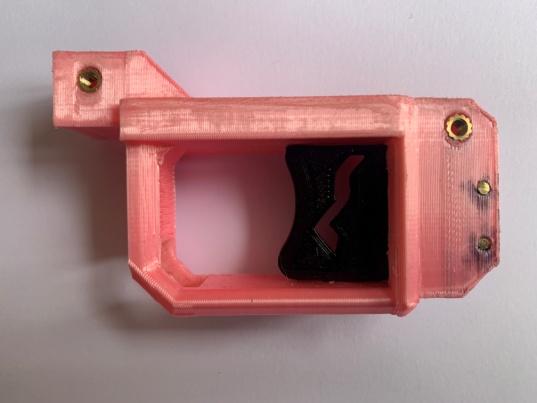
No brim or supports needed

4 Walls, 25% infill for strength

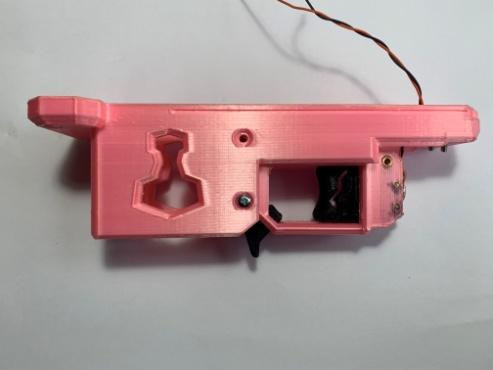
Superglue Top part to bottom using the 2 pins to align

## Trigger & Micro Switch Sub Assembly





## Magwell, Mag Release Lever, Trigger Assembly and Grip

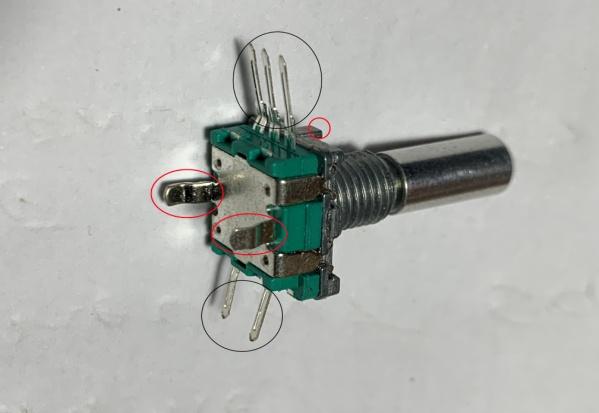
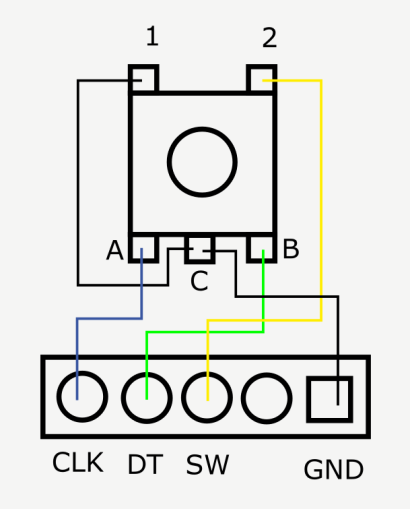


## Magazine Detection - Micro Switch

Set this up now, test fit and sand (do not leave in place as you can lose the small parts.)



## Rotary Encoder Set Up



Cut Red Circled tags off, flush

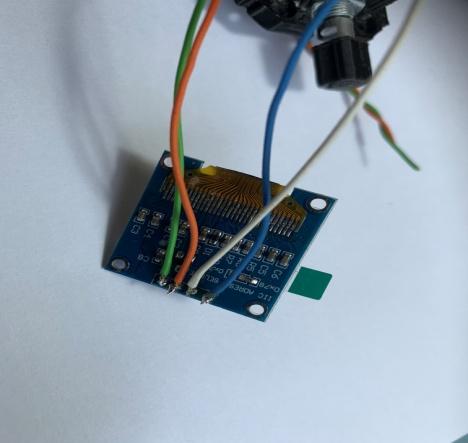
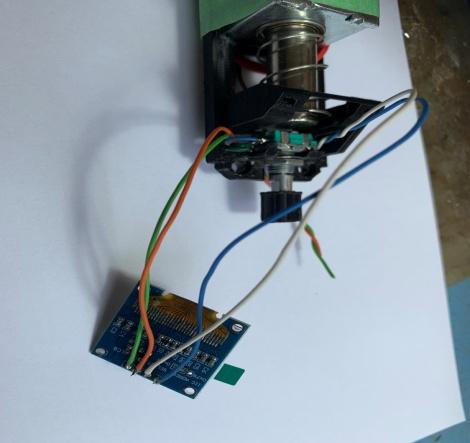
Straighten and Trim Black Circled Legs, they can be about 3mm long when trimmed

Solder wires on as shown, the smaller diameter wire you can use the better. Wires go down the side of Encoder, try to solder so they go this direction see Brown and Whites in Fig n



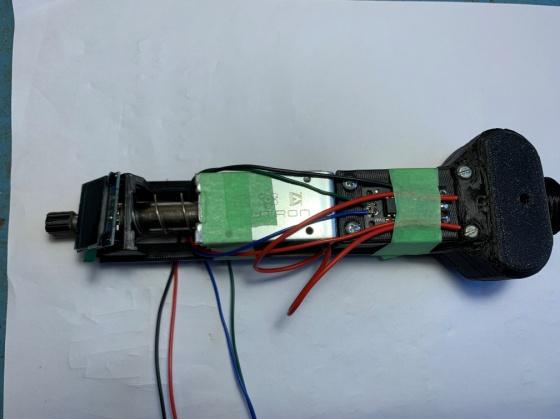
## Oled Wires

Note small diameter wire, soldered in the direction they exit, not through the pcb holes. Please note: These also go down the sides, not behind PCB like this.

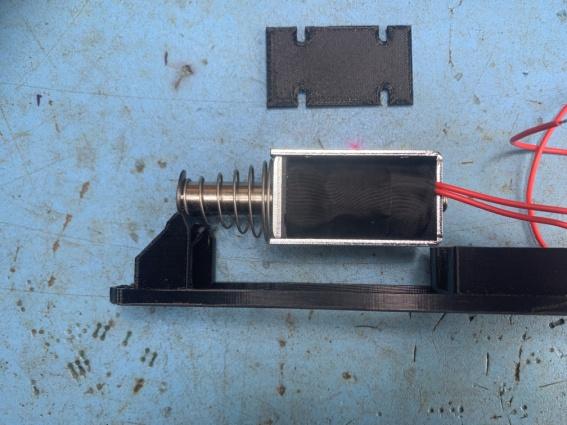


## Motor Mount Assembly – Top Half

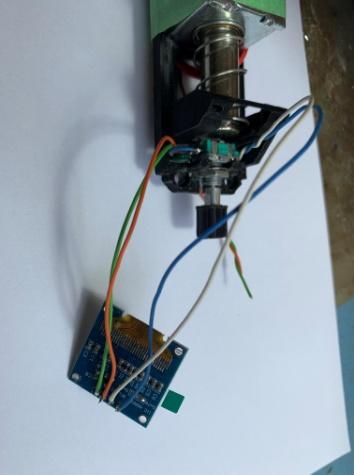
Motors & Flywheel, Solenoid, Encoder & Oled



Finished Top Assembly Finished top & Bottom Assembles  
 (Mag stand link; <https://www.thingiverse.com/thing:4595652> )



Test fit solenoid, you may need to cut a bit at the Oled end of the Motor Mount to get the piston to return correctly. Do not mount it yet. Pull each of the wires into two of the screw holes in the solenoid. Use the other two holes to attach the solenoid mount bracket to the solenoid. (Good way to lead the wires down into the electronics bay)

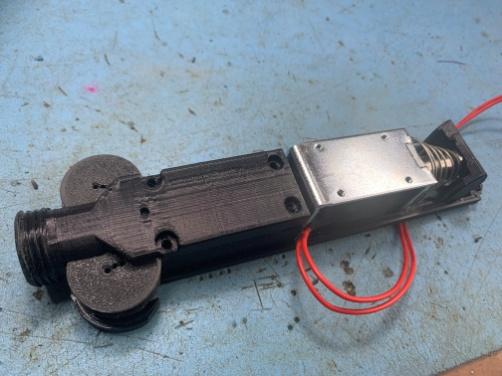


Fit Encoder to Motor Mount, not the Oled yet

Mount fly wheels to motor, use wooden toothpick to align holes. Secure with screws and Loctite. Ensure that the flywheel is sitting fully level on the motor by rotating and watching for wobble.

NOTE: Do not use metal object to align holes. The wires are very fragile (do not pull on them!)

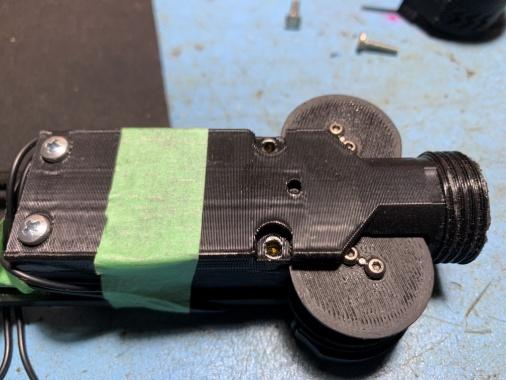


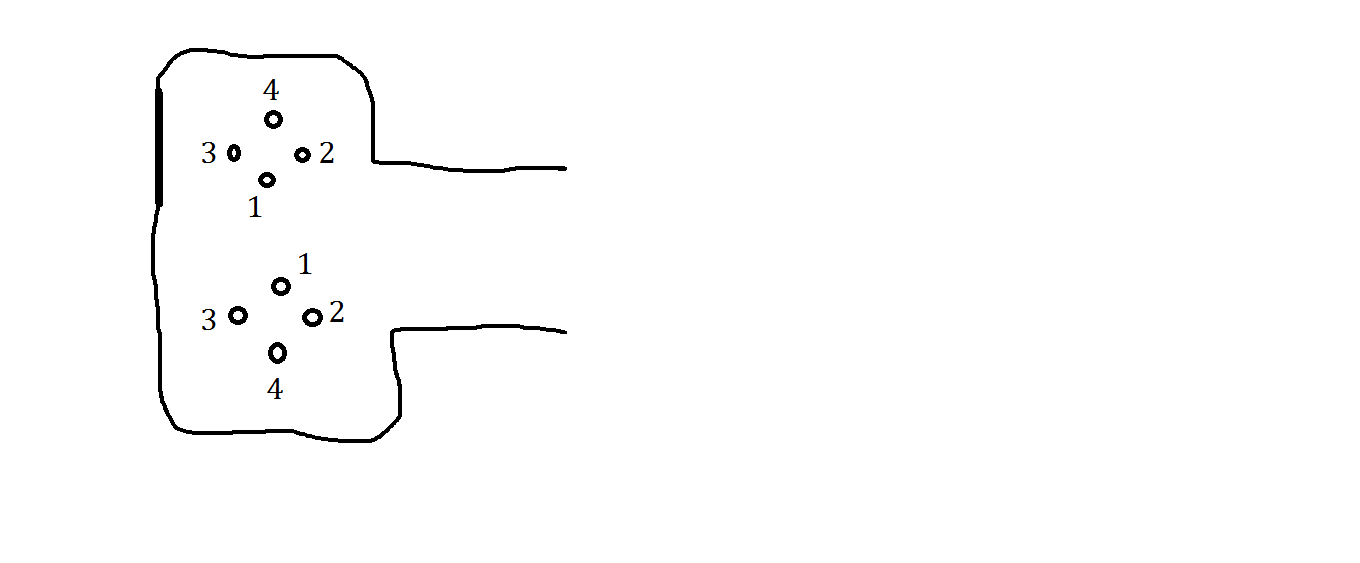
Fit Dart guide to motor mount, no screws yet, but long m3 can be inserted at the rear to help hold it. Fly wheels can be test fitted for clearance.

Carefully move Motor/Flywheel combo into place (without pulling on the wires!)

Ensure Motor sits flat and proper in the cage. Use Tooth picks to align mounting holes.





* + - 1. Insert screw nearest dart guide, push down on motor to ensure it’s seated properly and tighten screw. Screw can be removed later to add lock tight so don’t over tighten
      2. Next do one nearest Mag well
      3. Front edge
      4. Out side edge
* Check motors are seated correctly, flywheels turn easily and are aligned.
* Remove one screw at a time to add Loctite.
* Carefully guide wires down cable guide (do not pull them - they break off. This is the voice of experience talking!)

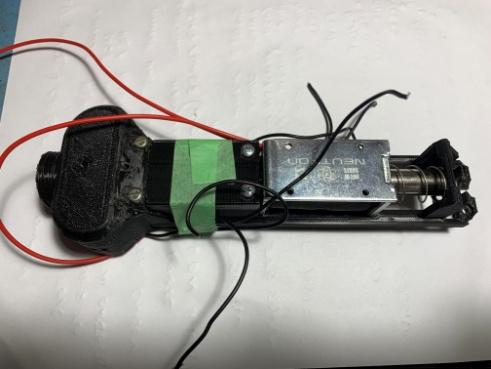


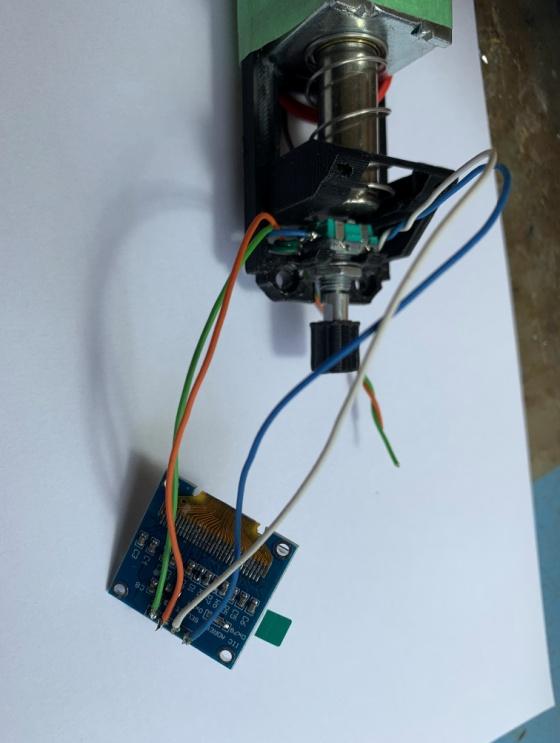
The front cover can now be secured with 2 x m3 screws. There should be brass inserts in the top of this.



Use tape around the dart guide to hold motor wires while you work on the rest of it

Fit nose to solenoid and fit solenoid to Motor Mount, and attach mounting plate with M3 screws, use tape to hold in place

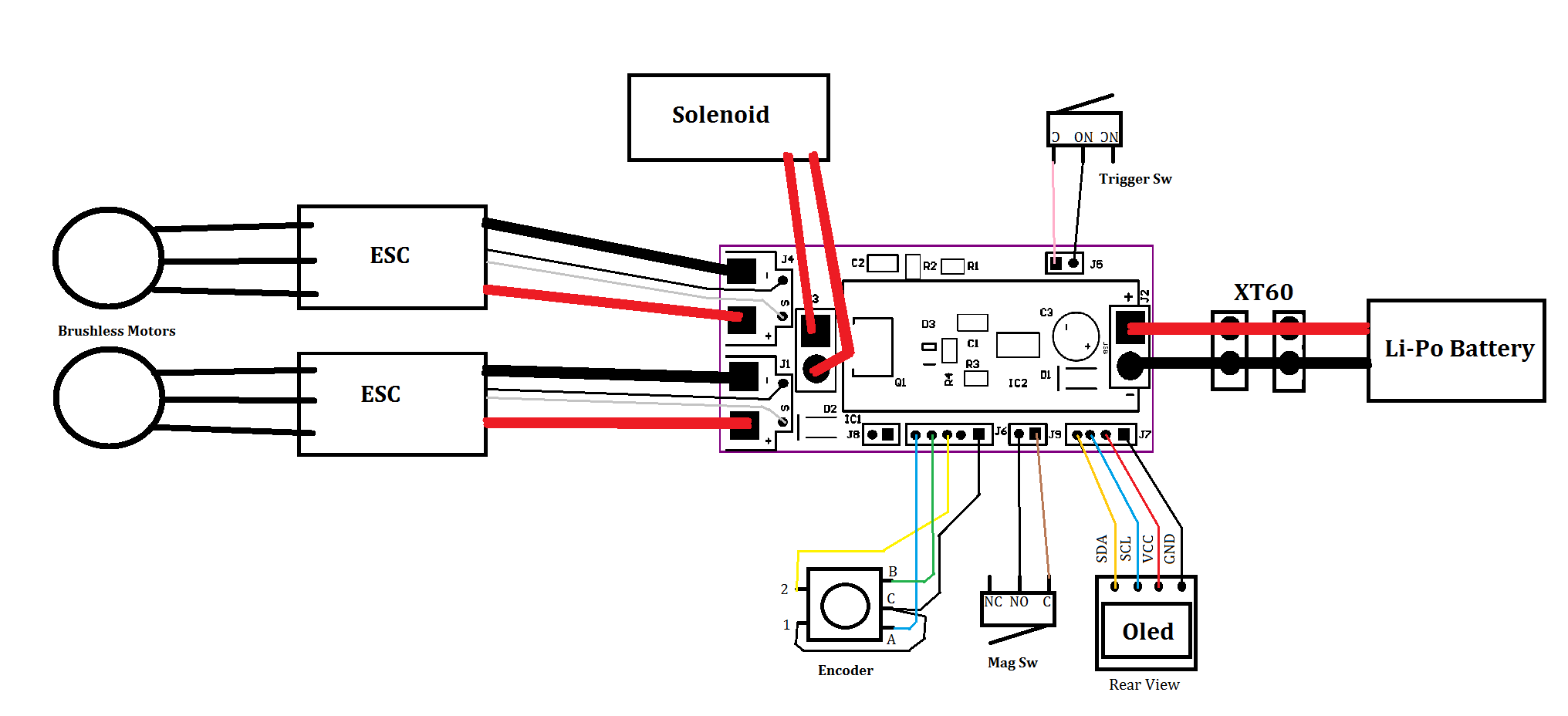


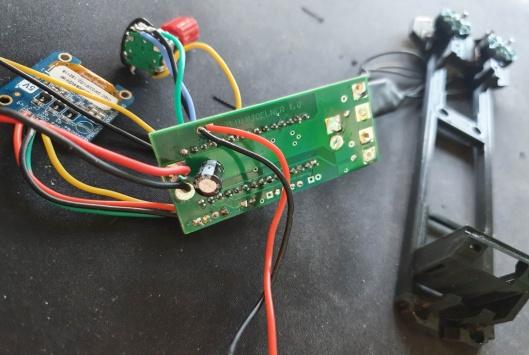
Mount Oled. Pay attention to getting wires down the sides so as to not get stuck in solenoid pusher



## Mjolnir PCB and Arduino

The following Diagram shows how the devices are connected together using the Mjolnir PCB.

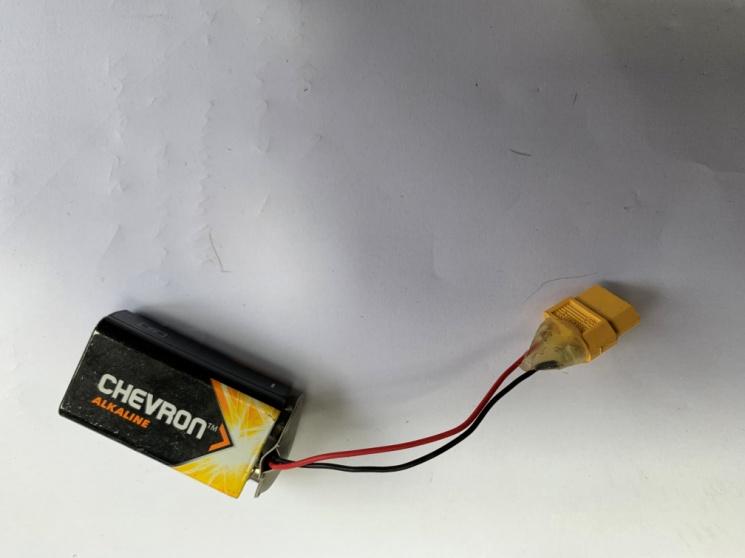




Showing Top side wire connections to PCB Bottom Side connections to PCB

Please note wire colors do not affect performance, diagram colors and photos may not match each other exactly.

## First Time power Up

To avoid Magic smoke, It is strongly recommended to power up for the first time using a 9v battery with suitable XT30 lead adapter. If there is a wiring fault this will help ensure components can not draw too much power and blow up.

LCD will come on and it should be ready to use.

Constantly check for components getting noticeably hot to the touch or unusual smells. Turn off immediately if you encounter either.

First check the Encoder is working, by long click and scrolling through the onscreen menu.

If all is well, test a single shot with no magazine, to start.

Still with no magazine, test Burst and Auto modes.

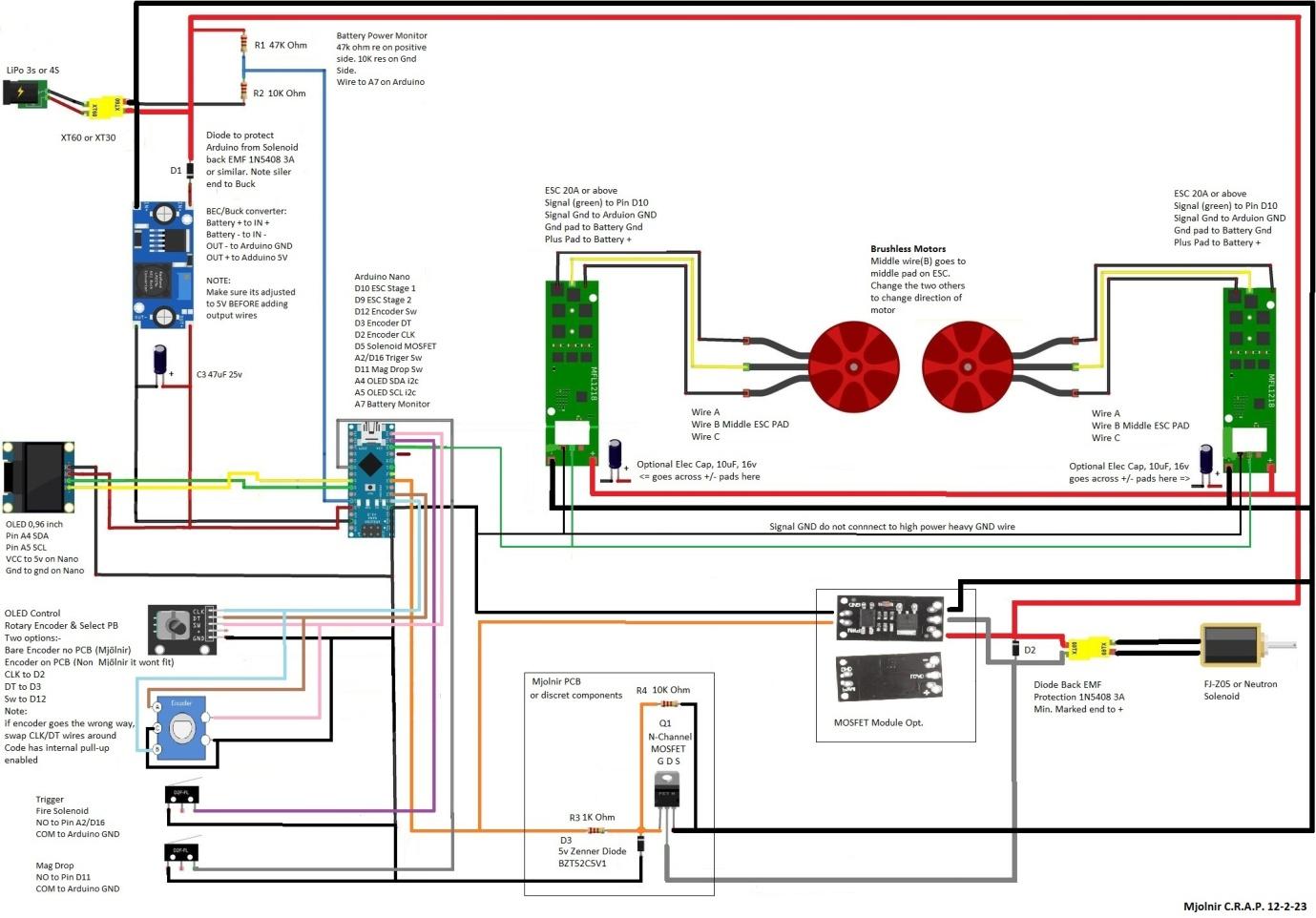
Ready for Calibration

# Mjolnir – No PCB, Discrete Components Method

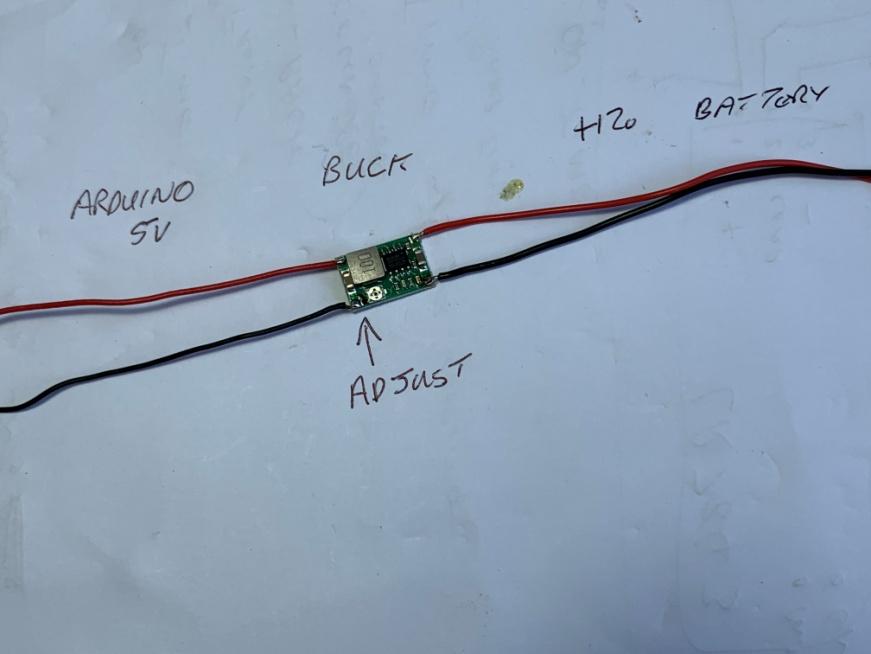
This method uses a Mosfet PCB (or discrete components) and Buck PSU module.

Sub assemblies are completed as per above; this option replaces the Mjolnir PCB

Please reference this diagram



## Buck PSU Module

Solder Tails onto the Buck module like so

It is important to set the output voltage on this **before connecting** it to anything else, MAGIC SMOKE will appear if this is not done. Despite the name, you do not want magic smoke.

Connect the Buck input side tails to an XT60 connector, pay attention to polarity.

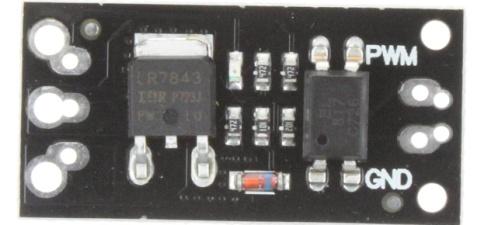
Via a screw terminal block, connect buck output to a multimeter set to DC volts.

Connect a suitable Li-Po battery and adjust potentiometer so the output reads 5.00v on the multimeter display.

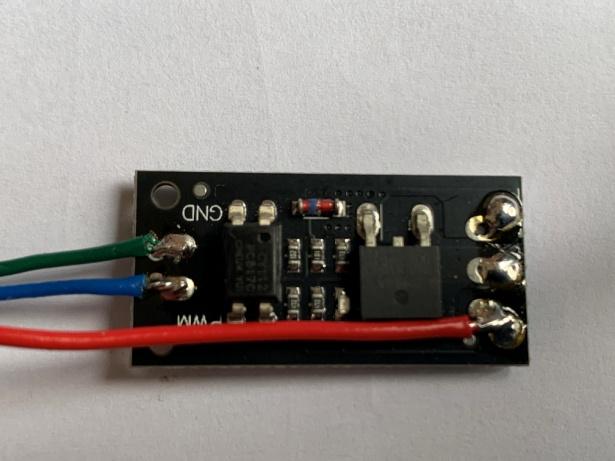
Disconnect battery/XT60/Multi-meter



## Mosfet and Wires

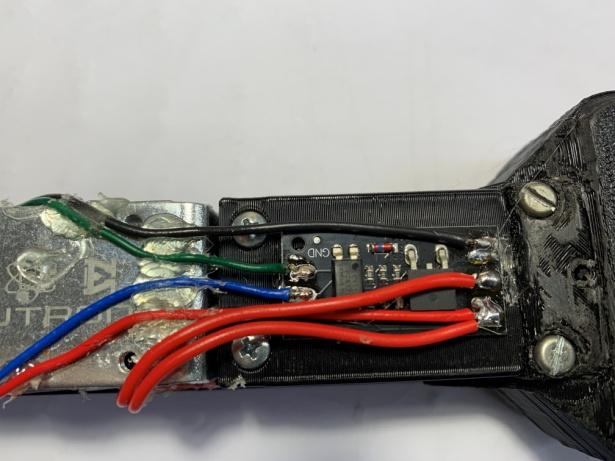






Note direction of soldered wires.

Negative is not shown.



Mosfet PCB. Hot melt guided to Dart Guide.

Solenoid wires are Red

Power = Red/Blk

Grn = signal GND

Blu = Pusher trigger signal

Assembly of the electronics into the electronics bay is a bit tricky and will come down to the individual.

Things to consider

There is not much room in the grip, suggest one pair of wires with XT60 on end.

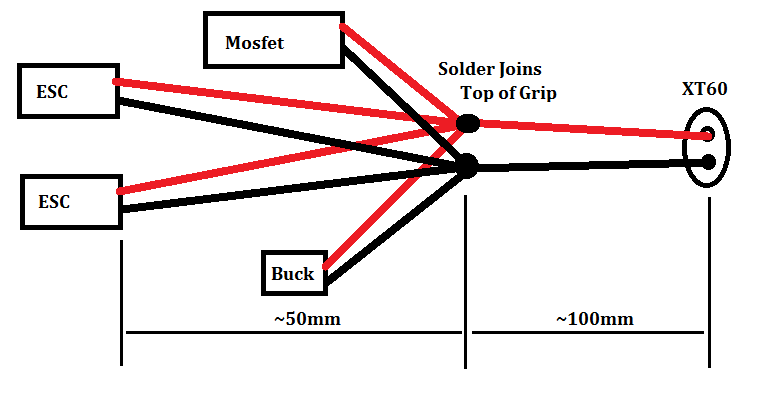
You need to join the power wires together; this is best done at the top of the grip, in space behind trigger.

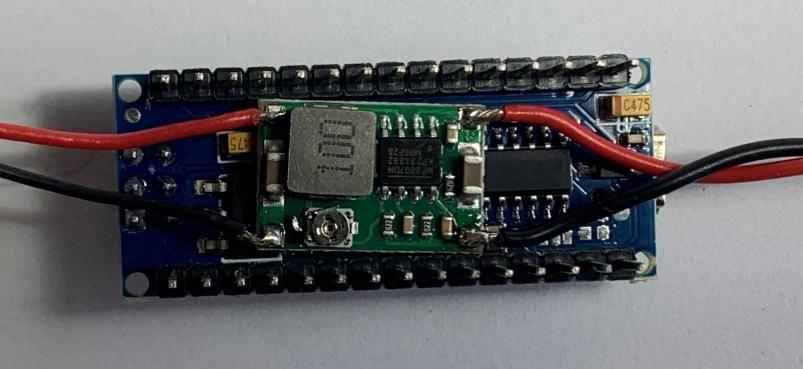
You will also need a common negative point for signal grounds to go to

The Buck will fit between the pin lines of the Arduino PCB

The Arduino will sit best with its pins facing down, USB to the rear

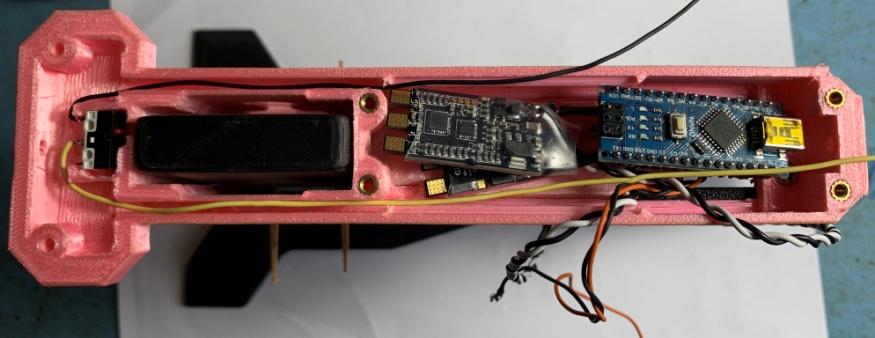
**Power wiring loom Drawing**





Arduino with Buck sitting between pins

The Buck will require heat shrink wrapping before installation



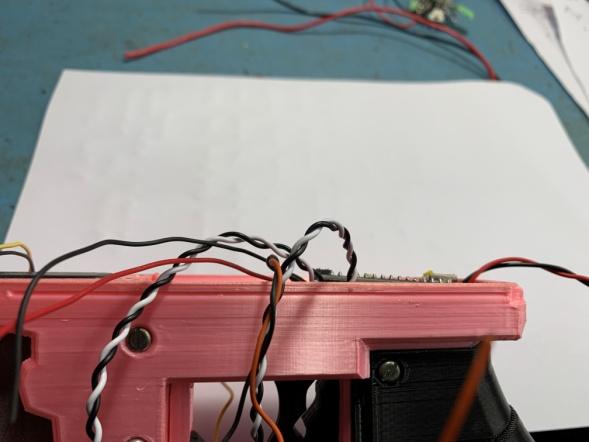
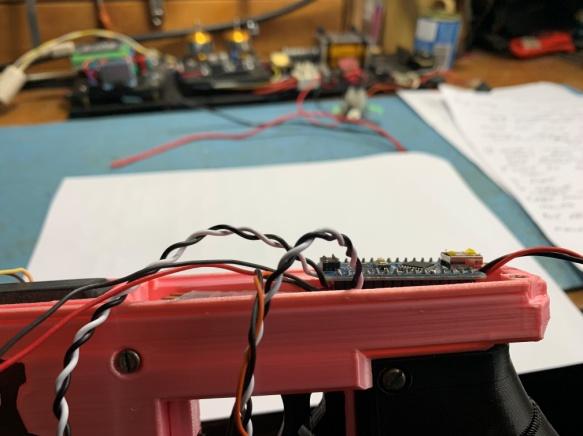
Electronics Bay

ESCs Back to Back

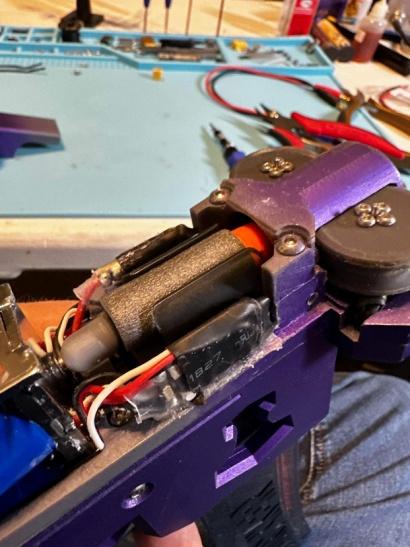
Arduino Pins down

Power wiring loom goes under PCBs

The Electronics must sit down below the lip of the Mag Well/Electronics Bay



**Example Pic’s of other Builds**



# How to Use OLED & Solenoid Calibration

On first time power up it will make some sounds, a logo screen will appear for a short time, and then the main Home screen is displayed. Default profile settings will be loaded.

OLED - Home Screen

* 1st Line: Top Left: Bat volts: 11.4v, Top Right: Game Timer
* 2nd Line: Select fire mode: Single, Burst: 2, Full Auto
* 3rd Line: ROF: MAX
* 4th Line: Pwr: 50%
* Ammo Counter (Large Digit)

NOTE:

ROF and Pwr change relative to select Fire Mode and your profile settings

Pwr is at 50% by default change, to get higher FPS

Ammon counter is reset by changing the Mag. Zero ammo keeps firing

Select Fire Mode

* Short click, on Push Button (PB), to change Mode
* Mode is on home display (Single / Burst: 2 / Full Auto)

Configuration Screen

* Long Click, on PB, to get into the Configuration screen
* Short click to get out when on the EXIT tag, top or bottom of menu
* Use Knob to move up/down configuration menu
* Short Click to select a item, pointer changes from > to #, use rotary knob to change value
* Short Click again to accept value

* EXIT - Exit config back to Home screen
* GameT: - Game Timer Default 0:00
* Power1: - Flywheel Stage1 Power 30 - 100% Default: %50
* Power2: - Flywheel Stage2 Power 30 - 100% Default %50
* A ROF : - Full Auto, Rate Of Fire 0(Max) - 150 Default: Max
* B ROF: - Burst, Rate Of Fire 0(Max) - 150 Default: Max
* Burst : - Burst Rate (number of darts to fire) 2-99 Default: 2
* MagS : - Mag Size capacity 0-99 Default: 18
* Ramp U: - Fly Wheel Start Ramp Up time 0 – 5000 Default: 0 (Fastest)
* Ramp D : - Fly Wheel Stop Ramp Down time 0 – 5000 Default: 4000
* Dwel U : - Ramp up Dwell time (hold rev) 0 - 5000 Default: 0
* Dwel D : - Ramp down Dwell time(hold rev) 0 - 5000 Default: 200
* SP Hi : - Solenoid Pulse High Time 0 - 1000 Default: 35
* SP Low : - Solenoid Pulse Low Time 0 - 1000 Default: 45
* SP Ret : - Solenoid Pulse Retract Time 0- 1000 Default: 45
* Bat Type : - 3S or 4S Battery Default 3S
* Bat Off : - Battery Offset for calibration, increments of 0.1 displayed as interger
* MagSw : - Mag Sw sensor On/Off Default On
* EXIT - Exit config back to Home screen

NOTES

* To load factory default settings in EEPROM, Hold trigger ON during power up or reset, let go when screen says too.
* Fleywheel Power2:

(Stage 2) cannot be set below Power1 (Stage 1) due to the possibility of a dart jam, the system will auto correct a low value set here.

* Game Timer:

This is an operator preset countdown timer, representing the game mode time to be played. Set in minutes. Eg 00:20 = 20 minute Game Time. On exit from Config screen, if this is not at zero, timer will start counting down from value set. At 1 minute from timer expiry, screen will flash, to indicate end of game coming in less than a minute.

* Bat Type:

Select 3S or 4S battery Type. Set battery low limits and performace setting for ROF

* Bat Off:

Battery offset to correct for error in Olded reading comparted to a volt meter. +/- in 0.1 increments

* MagSw:

Turn Mag Sw Detector On/Off . Disables ammo counter and ‘mag dropped’ display

SOLENOID Calibration

To get best performance and ROF you should Tune or Calibrate the Solenoid.

In order to get a good solenoid tune, you need to have 3 things:

- Push-out Time while LiPO is fully charged, ~12.5v or similar

- Push-out time while LiPO is depleted, storage voltage 11.5v or similar

- and retract time.

It is advisable to disconnect the ESC signal wire so that you don't have the motors draining power, and affecting your tune.

We need to edit these values in the Configuration screen

SP Hi Solenoid Pulse On Time High default = 35;

SP Low Solenoid Pulse On Time Low default = 45;

SP Ret Solenoid Pulse Retract Time default = 45;

Focus on the return rate first

Via config menu, change the following values:-

A ROF Max It is on this by default

SP Hi 150 Solenoid Pulse On Time High

SP Low 150 Solenoid Pulse On Low

SP Ret 100 Solenoid Pulse Retract Time

Exit Config

Set select fire to Full Auto

* Start with Pulse Retract Time, at 100ms,to see if the solenoid is fully retracting between shots.
* You might be able to use your finger to feel it.
* Increase or reduce this number until you have a consistent and reliable retraction.
* n.b. solenoids will retract slower with a fully loaded magazine, so once you have found the number, add a bit extra to it (like 10ms)
* To achieve the above may require multi edits

Then focus on the fully charged pulse time.

* Fully charge your LiPO.
* Start reducing both Pulse On Time High and Pulse On Low
* keep the numbers the same, until it's quick, but not so quick as it's short-stroking
* Use your finger up the magwell to determine how hard the solenoid is hitting it.
* Keep reducing the time until it starts to feel weaker, then back off.
* Add a bit extra to it for safety margin. Write down this value.

For the low battery discharge time, bring the LiPO to storage charge.

Then do the same thing as above.

Once you have both the Pulse On Time High and Pulse On Low value, key them in and test the solenoid over a range of battery values.

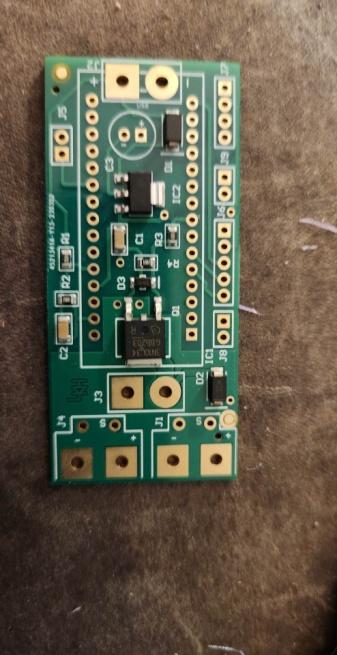
Reducing the timing will make it faster, at the expense of reliability.

Thous values: SP HI = 47, SP Low = 47, SP Ret = 27

# Appendix A: Mjolnir PCB

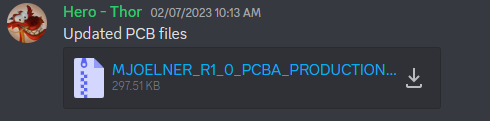
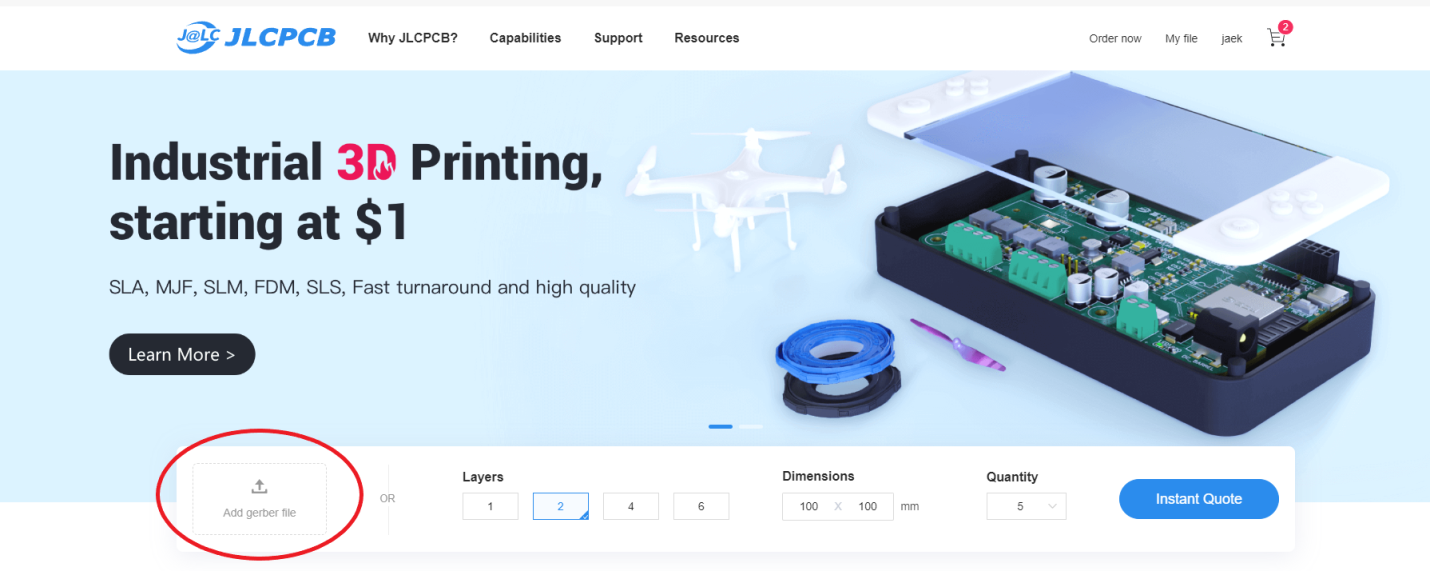
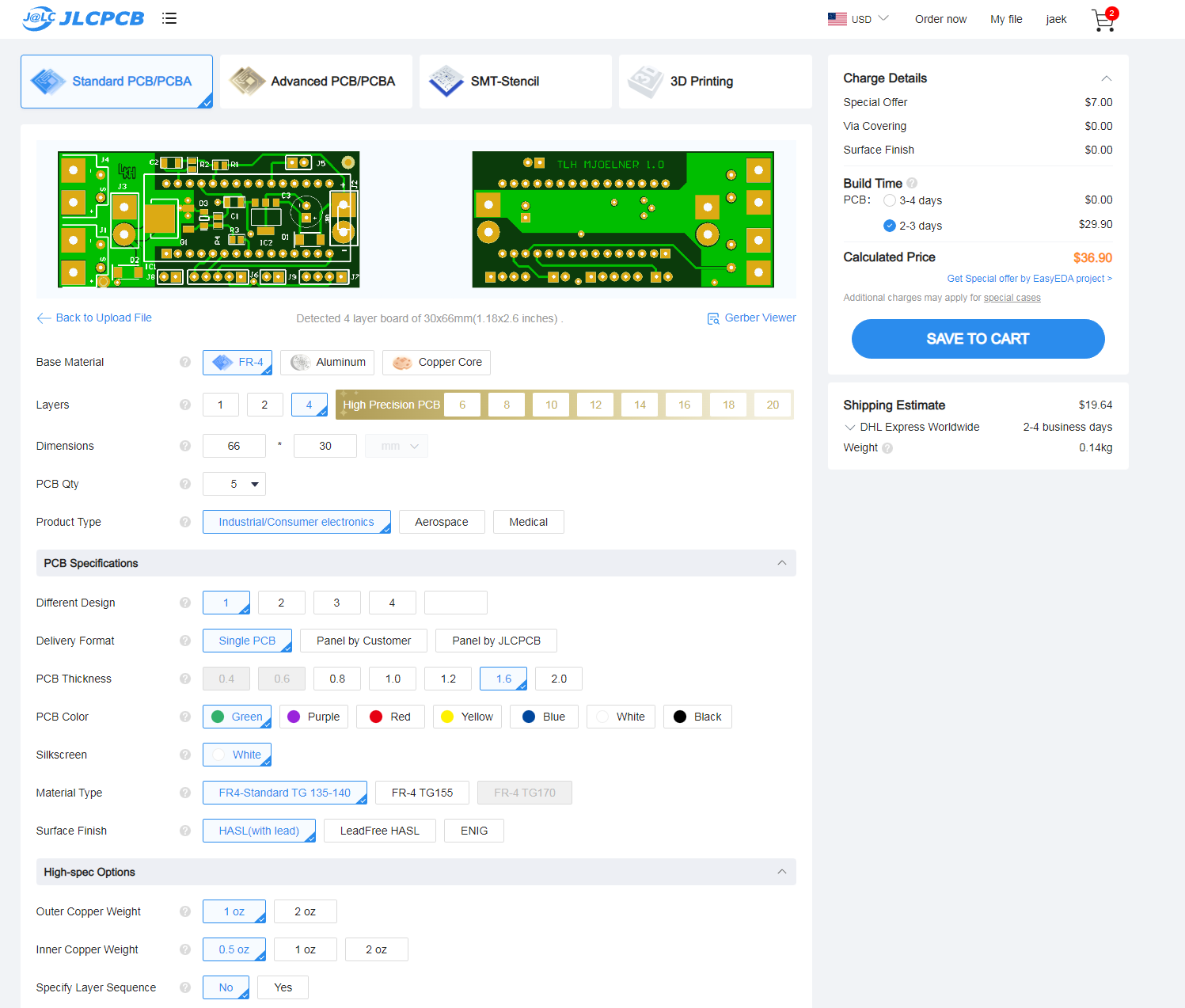
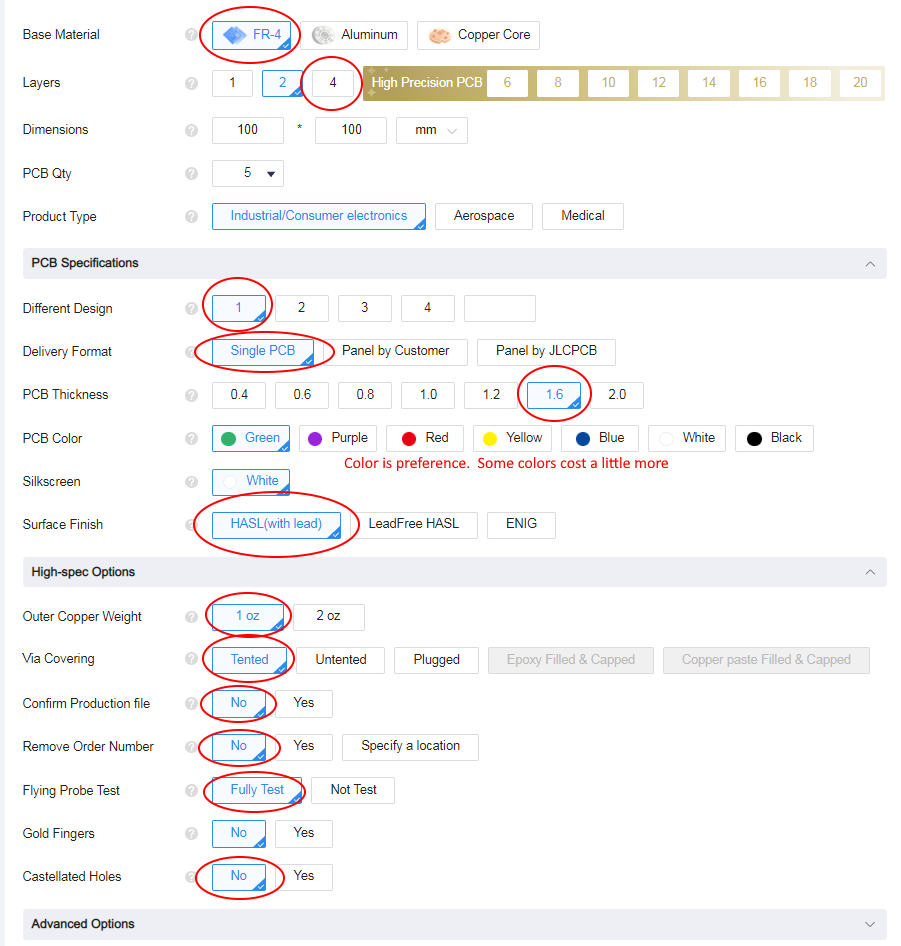
The Mjolnir files come with a folder containing all that is needed to make a functional PCB.

This folder can be provided to a local PCB prototype service to be manufactured, as ether

* Blank PCB – You source and populate with components
* Loaded PCB – Arrives as per left pic, just need to add Arduino

**Mjolnir PCB Ordering @ JLCPCB**

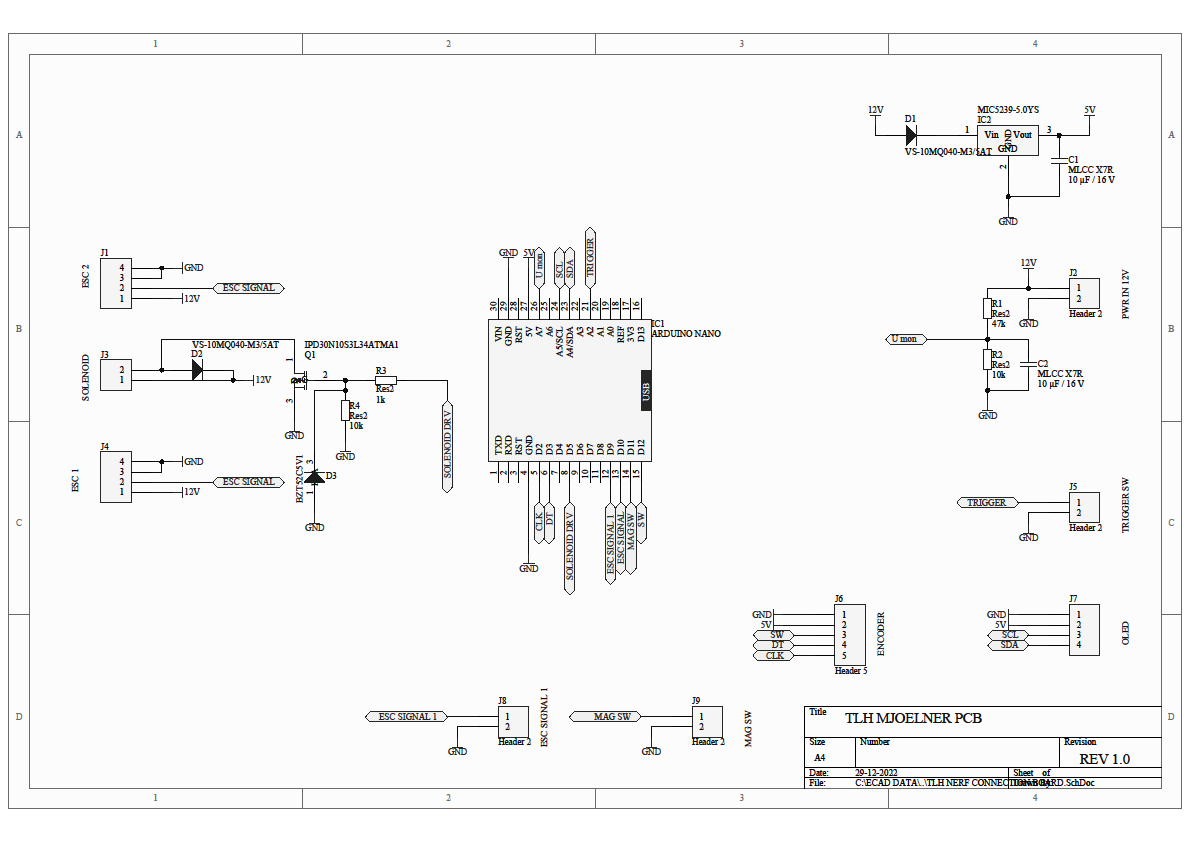
It is highly recommended to make an account with JLC for repeat ordering.

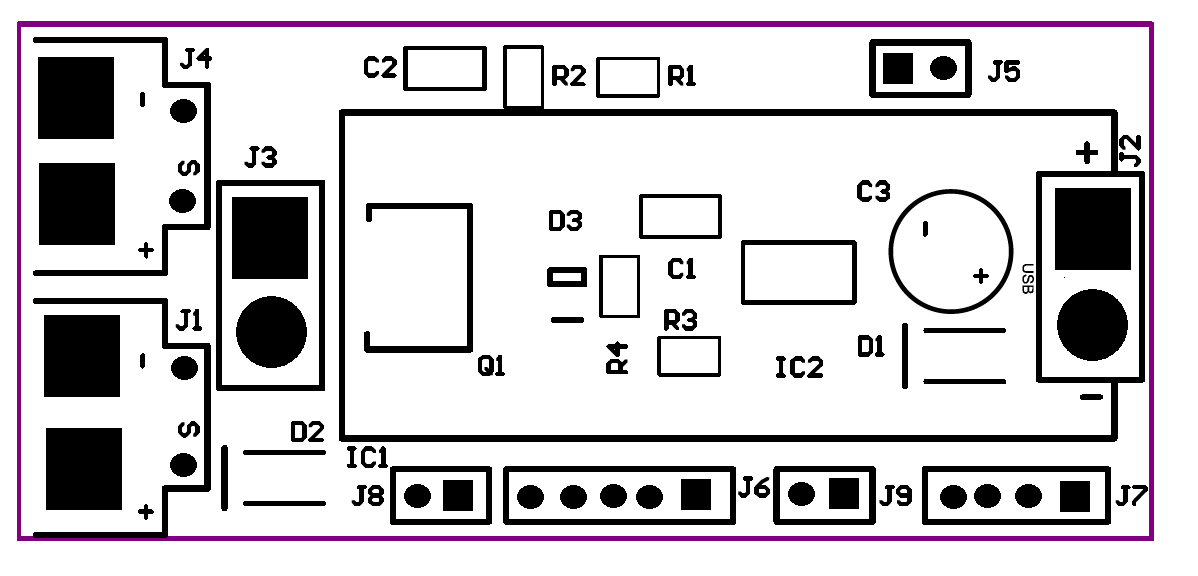
1. Retrieve the PCB files from the #mjolnir-files text channel in the discord. Leave the file as a .zip
2. 
3. Navigate the JLC’s homepage, and upload your file.
4. After uploading, this page should pop up with all the default PCB settings.
5. Leave everything at the default settings, but you can change the PCB color if you wish. If the default settings do not save correctly, use the image below. 
6. Leave assembly unchecked, and stencil unchecked.
7. Done! Place your order!

**Components**

If you chose to only buy the unassembled PCB, you’ll need to source the components yourself. Components can be found at Digikey, Mouser Electronics, and other online stores. Each component is paired with a designator and quantity required.

BOM for the pcb:

Mjolnir PCB Circuit Diagram



**Mjolnir PCB Headers**

J1 ESC2 (STAGE 1)

J2 PWR 1 = +12v 2 = Gnd. From XT60

J3 Solenoid Pusher

J4 ESC1 (Stage 1)

J5 Trigger - Micro Switch

J6 Rotary Encoder

J7 Oled

J8 ESC Signal

J9 Mag Detect - Micro Switch

| **Mjölnir PCB Electronics Components** | | | | |
| --- | --- | --- | --- | --- |
|
| **Project files on Github** | [**https://github.com/Bsanford0916/Cross-Rough-Atlantic-Projects**](https://github.com/Bsanford0916/Cross-Rough-Atlantic-Projects) | | | |
| **Item** | **Quantity** | **Link** | **Notes** | |
|  |  |  |  | |
| **PCB** |  |  |  | |
| Capacitor 25v, 47uF | 1 | https://www.mouser.com/ProjectManager/ProjectDetail.aspx?AccessID=42ba671b8e | All in one Order BOM | |
| Ceramic Capacitor 16v, 10uF | 2 |  | MLCC X7R 1206 | |
| Zener Diode 5v1 (*BZT52C5V1*) | 1 |  | SOD-123 | |
| Diode (*VS-10MQ040-M3/5AT)* | 2 |  | Do-214AC (SMA) | |
| Voltage Regulator (*UA7805M05CDCY*) | 1 |  | SOT223-S | |
| P-channel Mosfet (*IPD30N10S3L34ATMA1)* | 1 |  | Dpak (TO-252) | |
|  |  |  |  | |
| 47k 1% 0.125W Resistor | 1 |  | R0805 | |
| 10k 1% 0.125W Resistor | 2 |  | R0805 | |
| 1k 1% 0.125W Resistor | 1 |  | R0805 | |
|  |  |  |  | |
| 2pin Header, 100mil | 2 |  |  | |
| 4pin Header, 100mil | 1 |  |  | |
| 5pin Header, 100mil | 1 |  |  | |
|  |  |  |  | |
| **Optional Electronics Components** |  | **These components are used in place of PCB and Arduino** | | |
|  |  |  | |  |
| Micro Dual BEC Board, 7-21V to 5V/12V ADJ | 1 | [Https://a.co/d/8TOD3Te](https://a.co/d/8TOD3Te) | | Alternate Component |
| Isolated Mosfet Relay Board 30V 161A | 1 | [Https://a.co/d/1EoDc6Q](https://a.co/d/1EoDc6Q) | | LR7843 |
|  |  |  | |  |

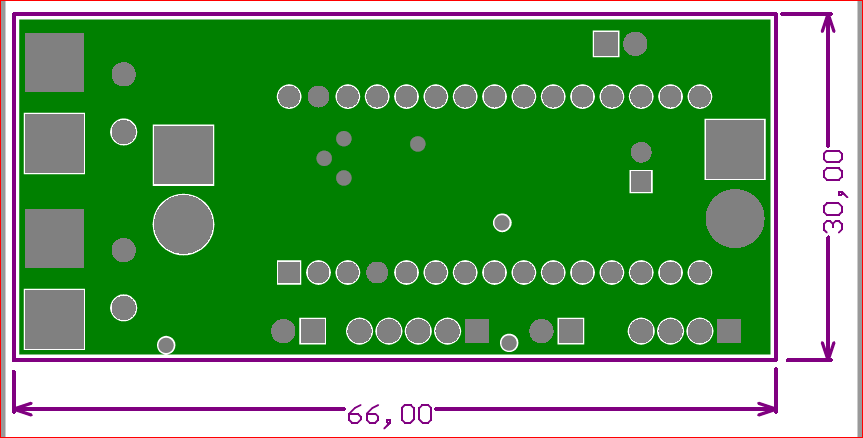
PCB LAYERS

Fig 1. Bottom Layer 1

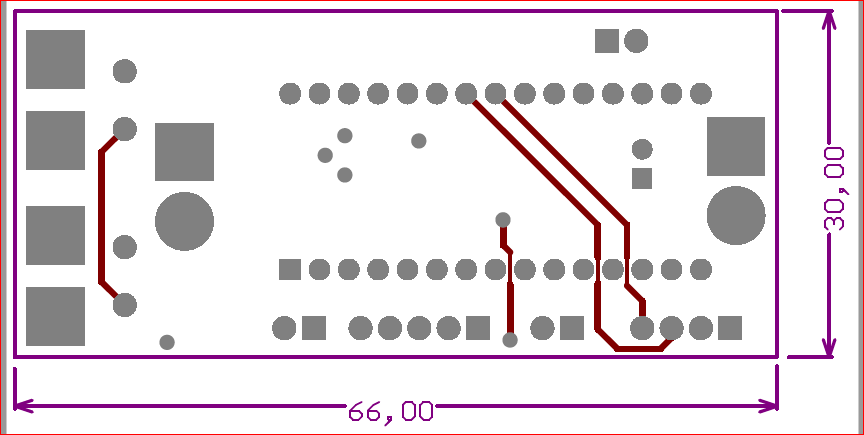


Fig 2. Mid Layer 2

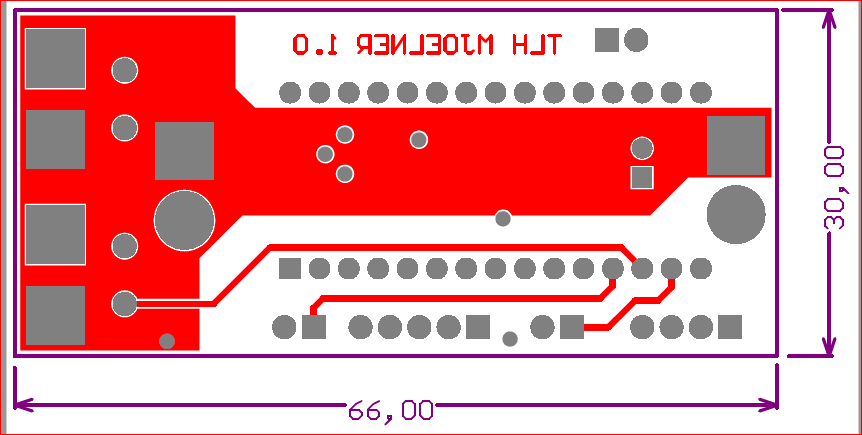


Fig 3. Mid Layer 3



Fig 4. Top Layer 4

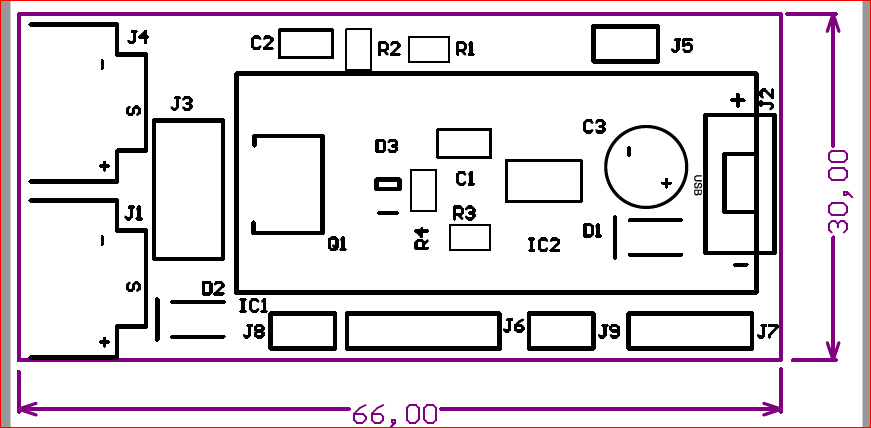


Fig 5. Overlay

Firmware:

Code used to control a processor. It could be firmware for the ESC's or firmware for an arduino.

BOM:

Bill Of Materials, also commonly referred to as a Parts list

**Useful links:**

Facebook Groups:

<https://www.facebook.com/groups/nothinbutnerf>

<https://www.facebook.com/groups/115961665148248>

**Shops:**

<https://blastersbyairzone.com/>

<https://outofdarts.com/>

<https://www.blaster-time.eu/>

**Guides:**

In my personal opinion, Airzone, Torukmakto4 and Ultrasonic2 are the founding fathers of brushless foam blasters. Their respective youtube/blogspots are filled with priceless information:

Torukmakto's blogspot: <https://torukmakto4.blogspot.com/>

Airzones youtube channel: <https://www.youtube.com/@AirzonesBlasters>

Ultrasonic2's youtube channel: <https://www.youtube.com/@Ultrasonictwo>

How to add settings/Flash an ESC: <https://www.youtube.com/watch?v=i6lhMcQLRSU>