MINIMILL V1.5
Electronics and Firmware

Made in a shed somewhere ... Probably

Millennium

Acknowledgements

Whilst the work that has been put in to this project by the millennium designs team is immense we can't take all the credit for the project as a whole, we had a lot of help along the way and would like to take a moment to recognise a few of the entities outside of the team that helped us make this all a reality.

To the team at openbuilds. As many of you may know we originally began this journey as a mod for the Openbuilds Minimill and whilst we've moved far beyond the capabilities of that machine it would be remiss of us to not to recognise the work that went into that project.

To the Voron team. Voron has also been a huge inspiration to us on the team from the ethos of the community driven development to the layout and content of our manuals and we do owe them a lot (we even use some modified voron parts).

And finally to Fabreeko. Early on Fabreeko had our back with support and donations so that we could afford to buy parts and they helped us guarantee that we could provide a machine that lived up to everyones expectations for that we are eternally grateful.

Introduction

Take a deep breath for a second and realise something.

You... yes **YOU**, are about to commit to wiring and programming a robot that can cut through metal, let alone your squishy human parts...
A machine that can easily electrocute you, cut you or set fire to your whole neighbourhood If it's not given the respect it deserves.
Please give this machine the respect it deserves !!!

Please follow the manual to the letter and perform any additional research you deem necessary before first attempting to use it.
If there is anything, and we mean anything that you are curious or unsure about you are more than welcome to ask us on our discord channel
After all, you are special to us and we don't want you to get hurt due to a lack of knowledge.

Most importantly from everyone at the Millennium design team,
Have fun building your very first milo.

Introduction

What to expect

This manual and associated firmware is for the Milo V1.5 with a Mellow Fly CDY v3 control board installed in the internal electronics enclosure found in the machine base. This manual will not cover external enclosures or the electronics compartment found in the Millenium Casa enclosure, however where possible this manual can be used as a guideline for those other setups. The firmware used in this manual will only consist of the basic 3 Axis functions of the machine, XYZ probes, relays, toolsetters and other related add ons will not be part of this base firmware and therefore will not be present in this manual.

RepRap firmware

RepRap firmware (RRF) is typically considered a 3D printing firmware but in the last few years has begun to see more and more CNC features implemented. RRF boards tend to operate over WIFI or ethernet and do not stream GCODE over the air but rather all GCODE is stored on board keep this in mind when finding a place to store Milo you will need some form of connection to the internet. RRF has the main benefit of being extremely control board agnostic so for the most part firmwares can be shared between different control boards with only a few pin definitions needing to be changed. Pins in RRF act a little differently from firmwares like Klipper and marlin, almost all pins are given a pin number (ex PC_7) and a pin name (ex Xmin) , 90% of the time it is the pin name that will be referenced in the firmware and not the pin number making it so that Xmin on one control board will also be Xmin on others keep this in mind if you want to use a different control board from the one found in this manual. However it is still a good idea to cross reference these pins before changing any settings in firmware most control boards will come with a table similar to the one found here, use this table in combination to a pinout diagram of your board to ensure everything will work as expected.

Introduction

Parts list and printing guidelines

The millennium team has provided a printing list with settings for you as a guideline for printing the parts necessary for the build. This list can be found **Here.** Inside you'll find details on some of the print settings that will give you the best mechanical properties suited for each individual part. Remember these settings are only a guideline and are open to your own interpretations but we do highly recommend following them.

File Naming

With your STL files downloaded and your printer warmed up you may be wondering which parts to print in your favorite colors ? Well have no fear we've got your back we've especially labelled each file to give you a hand.

Primary color

Example Handwheel body x2.stl
These files have no prefix in their filename so are safe to print in your primary color.

Accent color

Example [a] table bolt down bracket A x2.stl
These files are prefixed with "[a]"
And are intended to be printed in your chosen accent color.

Quantity required

Example Handwheel body x2.stl
If any file ends with "_x#", that is telling you the quantity of that part required to build the machine.

Firmware : Installing RRF

Sit back and relax !

For those of you who are less software inclined don't panic we've done most of the work for you. RRF requires you to populate an SD card with a specific file structure and files to go with it, to make this easier for you we have gone ahead and populated the majority of these files for you, all you need to do is follow the very simple steps down below to get your firmware installed on your board. If you don't wish to install the premade firmware and would rather create your own from scratch please follow this tutorial HERE.

Step 1 : Prepare your SD card

You will need a freshly formatted SD card with a capacity up to 32GB, If the capacity of the card is 4GB or lower, use the FAT16 format ,If the capacity is more than 4GB (up to 32GB) then you will have to use the FAT32 format. The SD card should also be class 4 or higher.

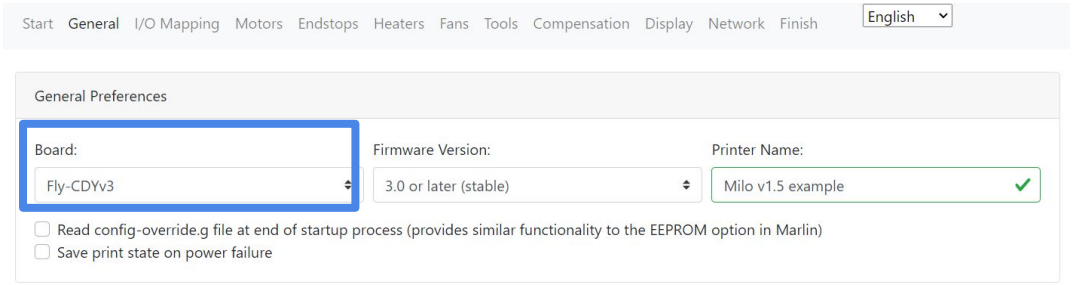
Step 2 : Copy Paste

In the main Milo v1.5 package you will find a folder marked example firmware, in this folder you will find multiple other folders marked with a number to denote the order of events and a descriptor to tell you what needs to be done with these files, starting with the file marked with "1.file structure", copy and paste its contents onto your SD card (copy only the folders contents and not the folder itself)

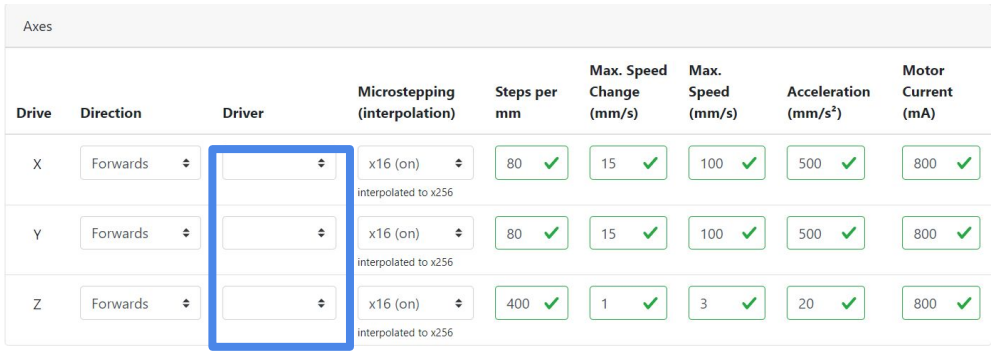
Firmware : Installing RRF

Step 3 : Create SYS files

At this point we will have to use this online tool to configure your SYS files. For the most part the setting you choose won't affect anything as we'll be making those changes later on down the line. There are however 2 crucial settings that need to be selected. The first of which is the board selection. For a standard build select the Fly-CDYv3.



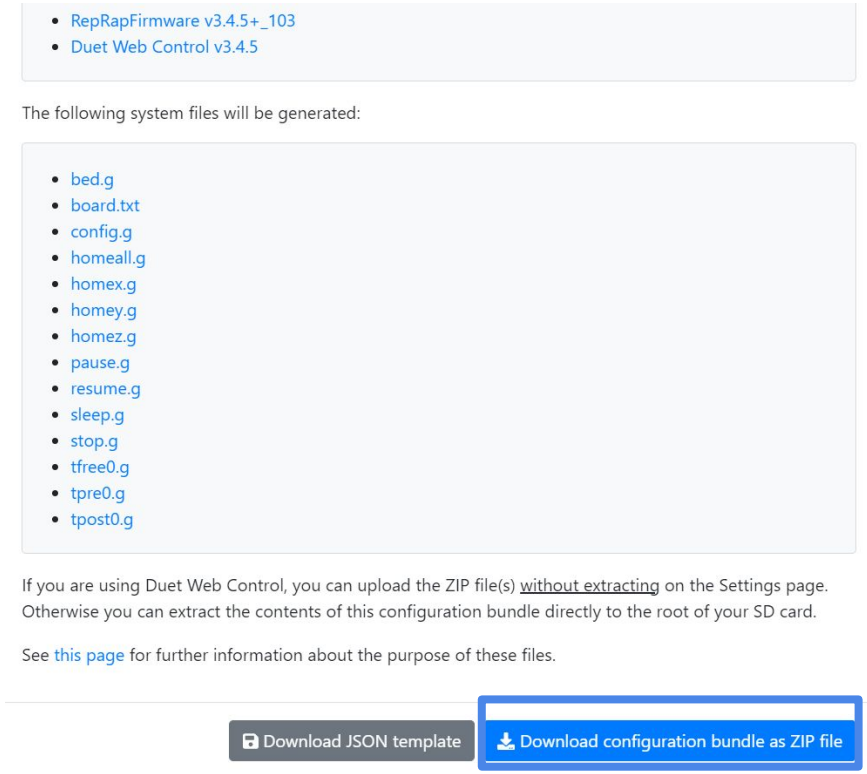
The second is the types of motor drivers you will be using.



Firmware : Installing RRF

Step 4 : Create sys files continued

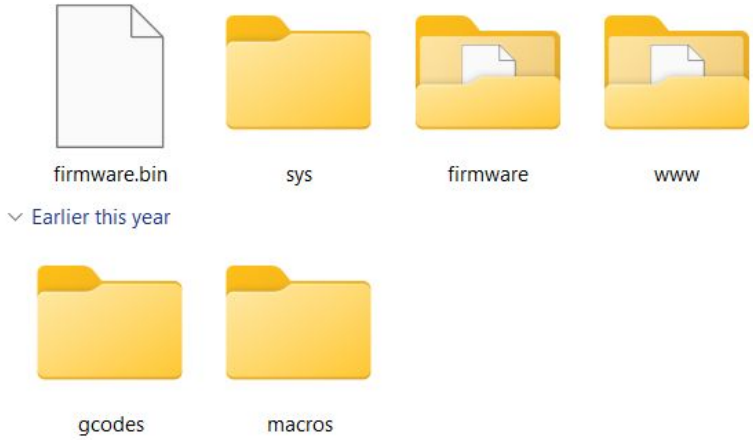
Leave all other settings untouched and skip straight to the FINISH tab at the end of the configurator. Once you click finish a box will pop up with all your generated config files. To continue click “download configuration bundle as ZIP file”, once this file is finished downloading, unzip its contents and copy the “sys” folder found inside straight onto your SD card.



Firmware : Installing RRF

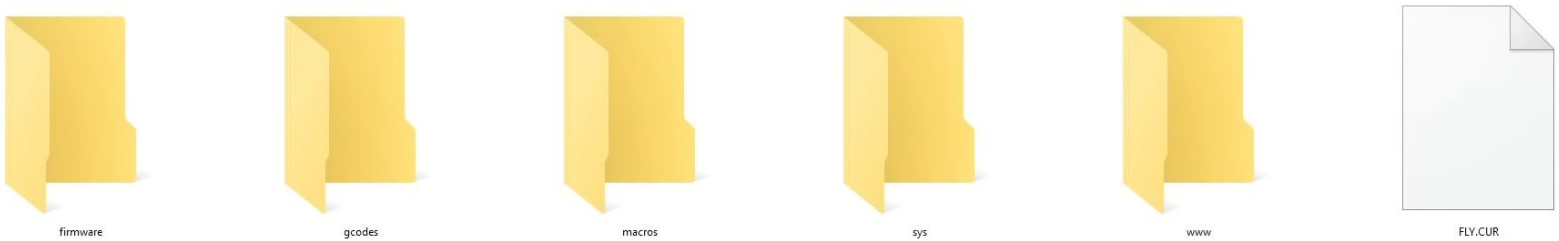
Step 4 : Create sys files (check your work)

Prior to flashing your board double check to make sure your SD card looks like the example seen here.

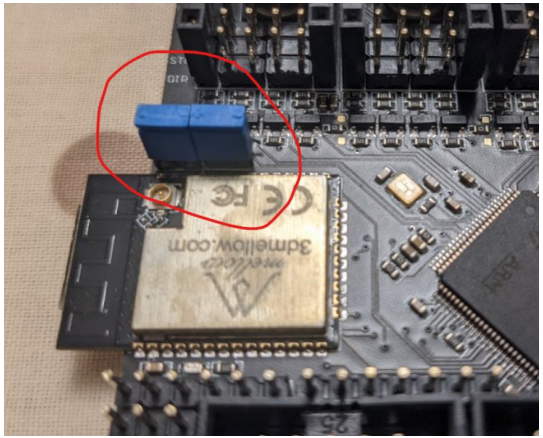


Step 5 : Initial flash

At this point eject your SD card from your computer and mount it into your controller board. Power your board either via USB or via your PSU and let the board rest for a minute or two. Turn the power off and put your SD card back into your computer. You will know this step worked if the firmware.bin file has changed into a FLY.CUR file as pictured below. Once confirmed return your SD card to the board.



Firmware : Installing RRF



Step 6 : Flash your wifi chip

If you are using the mellow FLY CDY V3 add 2 jumpers as shown here. You may need to take similar or different steps depending on your board. Once the jumpers are in place power up the board using 12-24v and connect to the USB port on the board. Connect to the board using a program such as PUTTY. Change the Com port to match the Fly-CDYv3 and connect.
The baudrate doesn't matter. Once connected type M552 S0 into the terminal followed by M997 S1 and wait for the wifi firmware to finish uploading. If you're struggling to get a readable output from PUTTY follow this link to set PUTTY up to work with RRF.

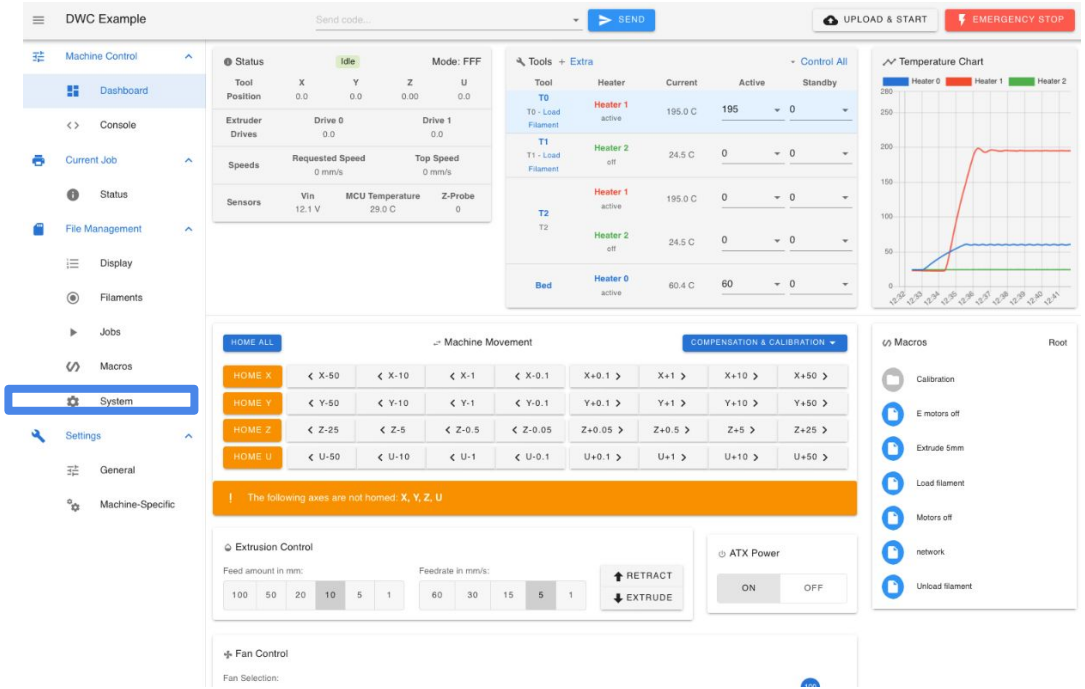
Step 7 : Set up your wifi

Whilst still in PUTTY type M552 S0 again followed by M587 S"your SSID" P"your password" and finish it off by typing M552 S1. Do not close putty at this point. After some time PUTTY should return an IP address for example 192.168.x.xx. Note this IP address down and then exit PUTTY

Firmware : Installing RRF

Step 7 : Configuring SYS files

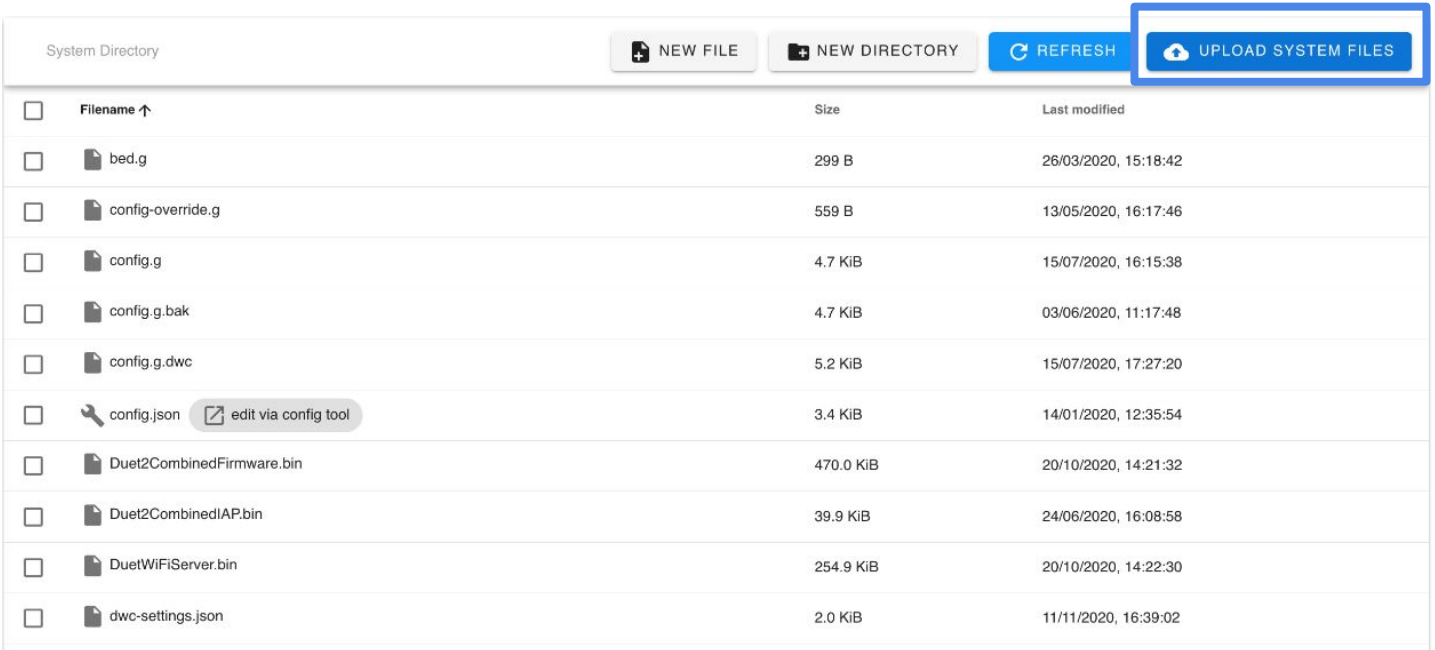
With your board still powered on copy the IP address you noted in the last step into your web browser. Once it loads you should now be face to face with the duet web controller (DWC). In the drop down menu navigate to the tab marked “System”.



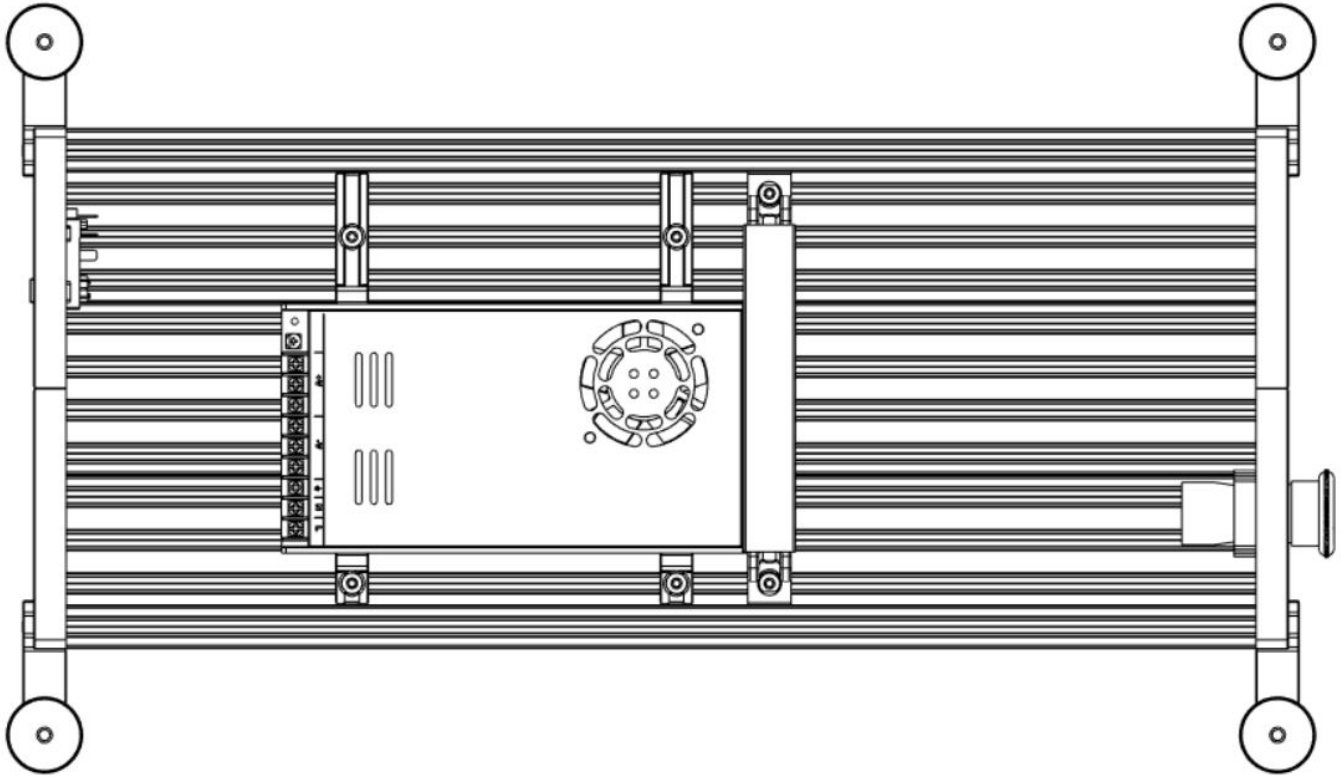
Firmware : Installing RRF

Step 7 : Configuring SYS files continued

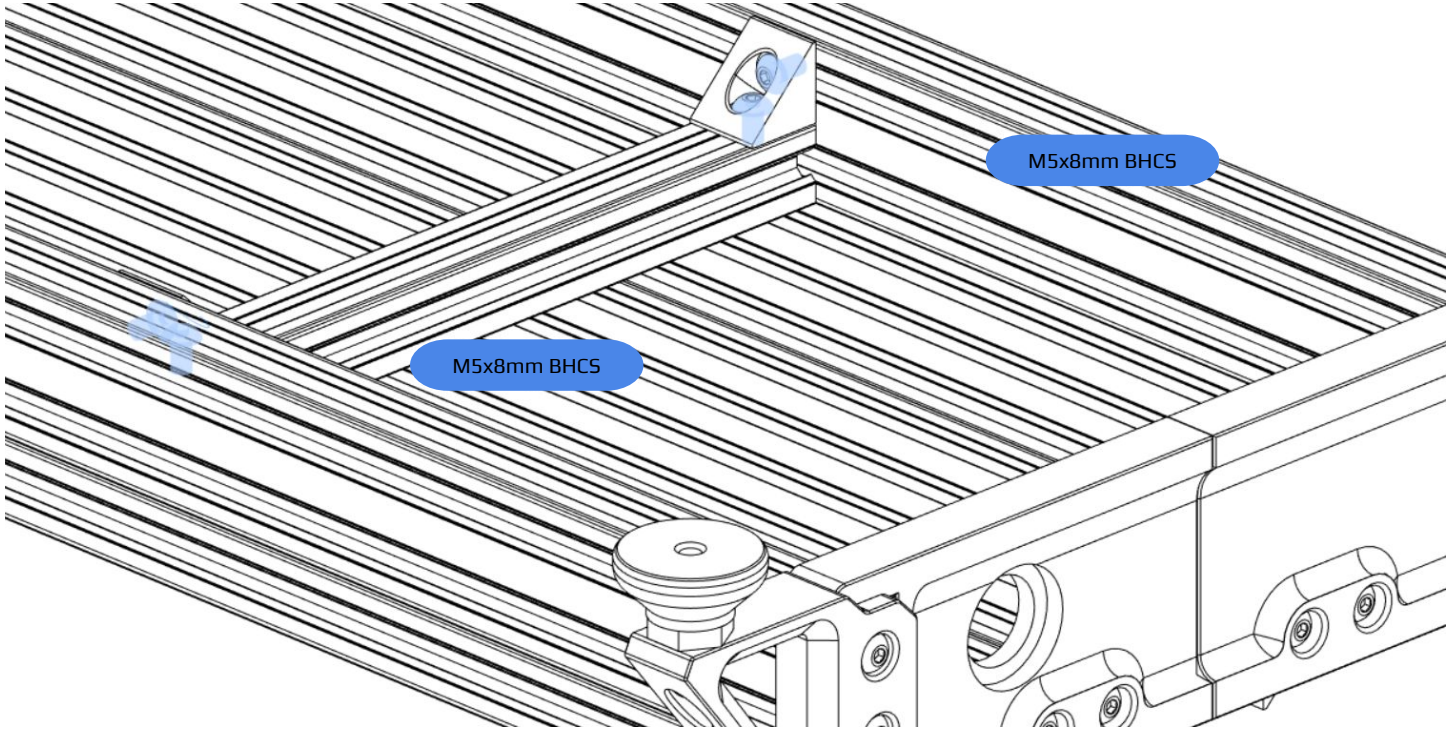
Once in the system tab click on “UPLOAD SYSTEM FILES” and upload all of the files found “2.Example config” to DWC, at this point the basic firmware setup is complete and all that remains are some firmware checks which will be covered later on in the manual. It is now safe to perform the physical install of your board into your mainboard mount.



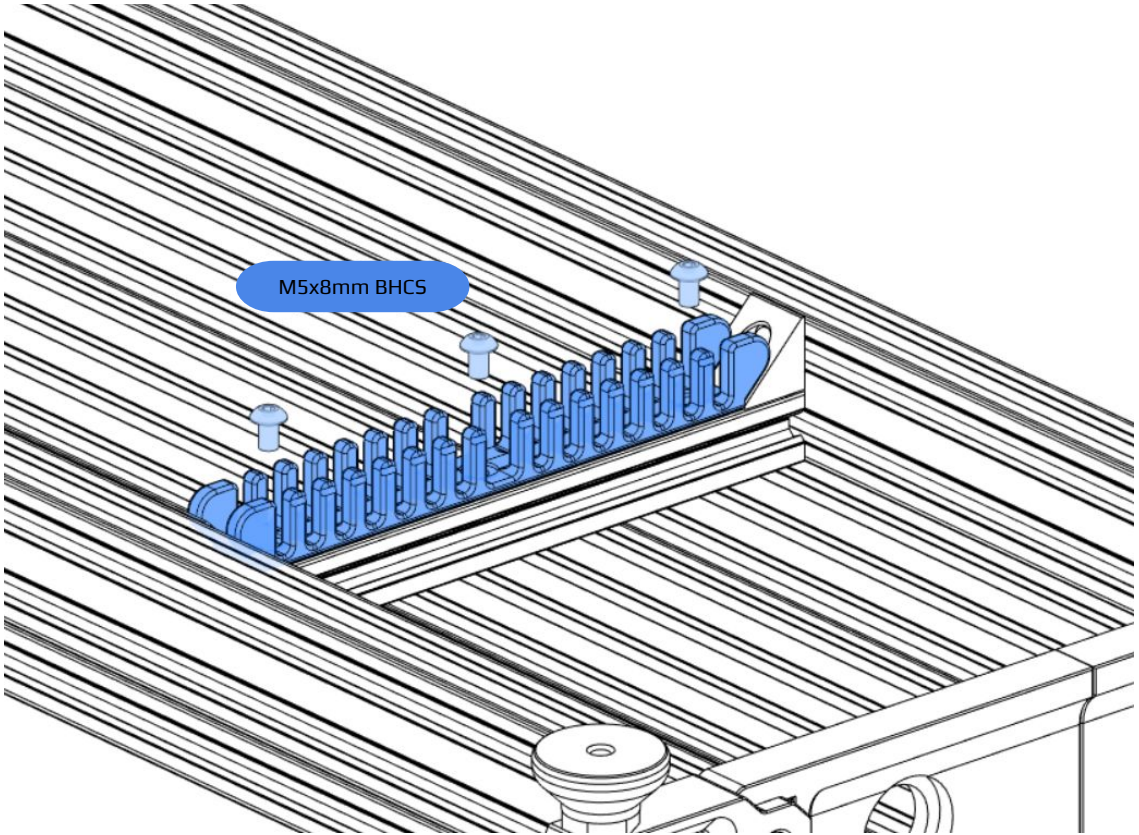
Electronics : AC hardware install



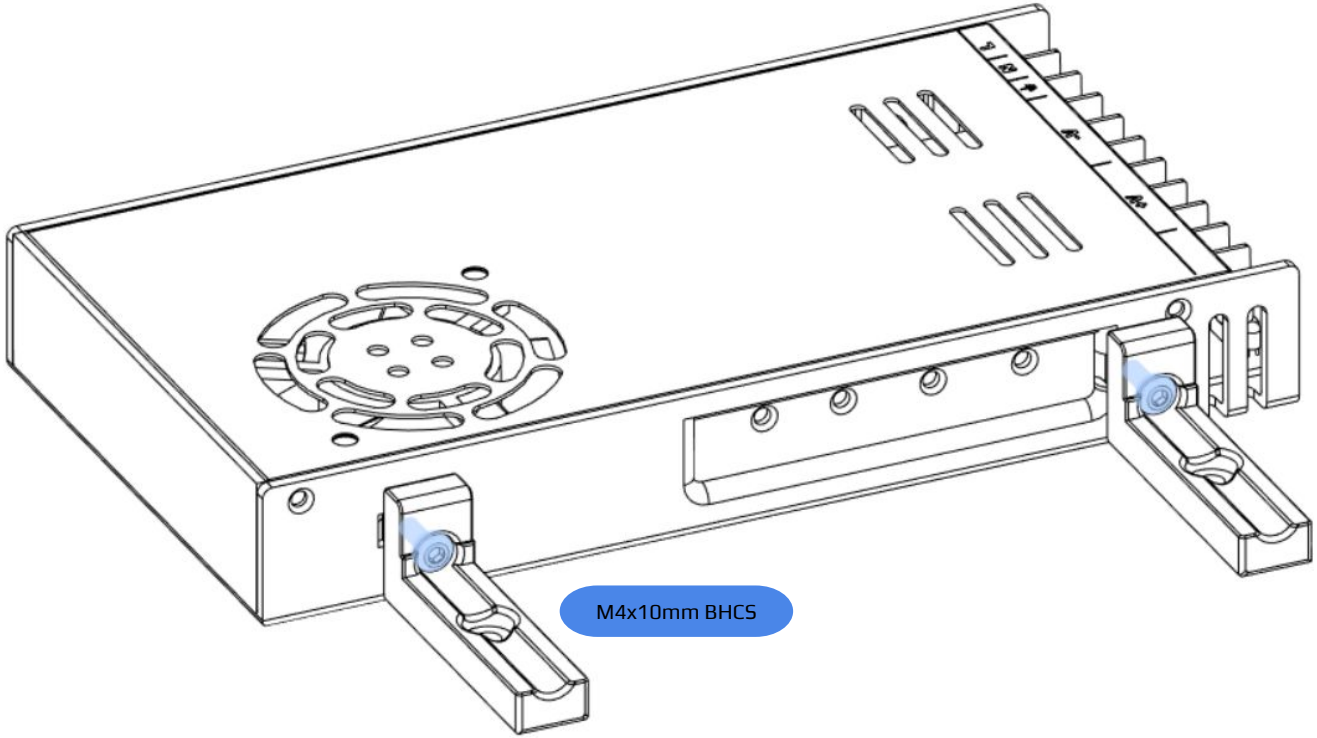
Electronics : Enclosure support



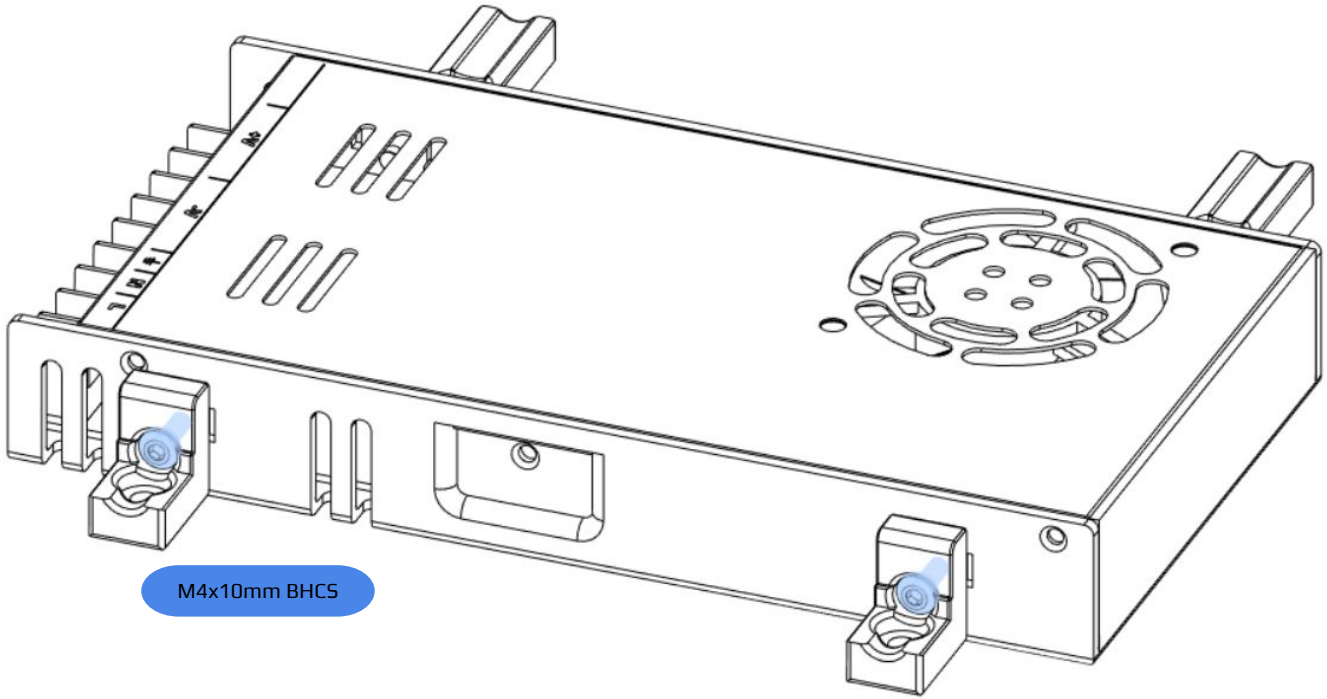
Electronics : Enclosure support



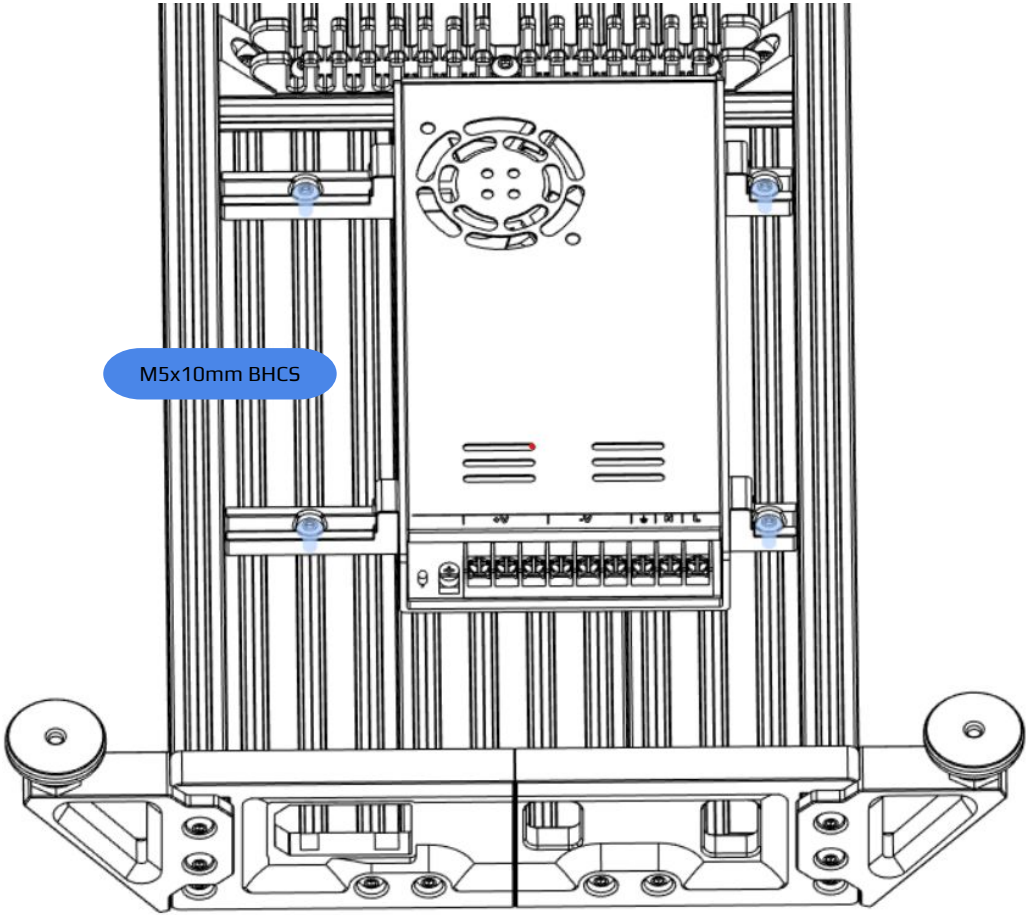
Electronics : PSU



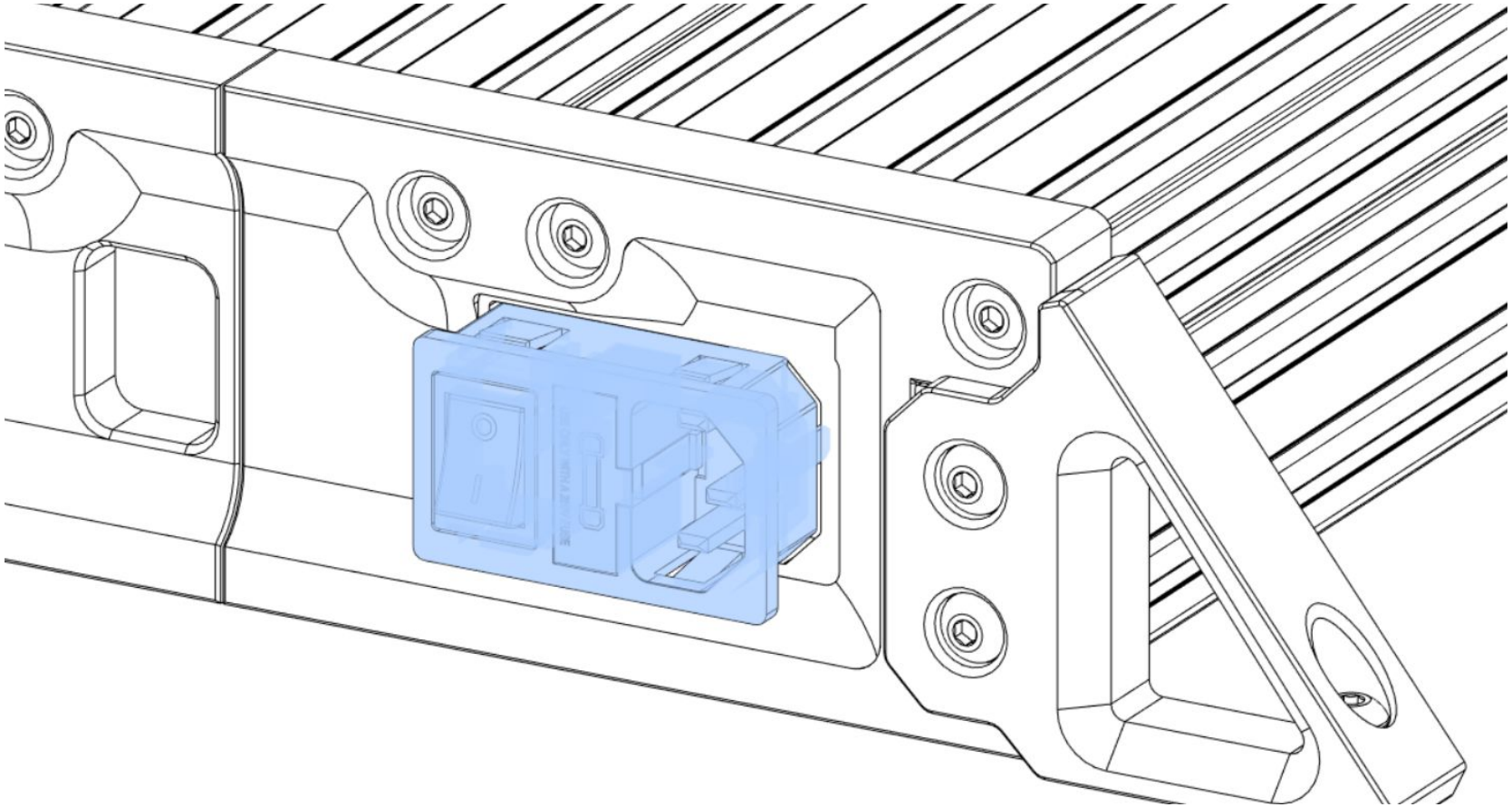
Electronics : PSU



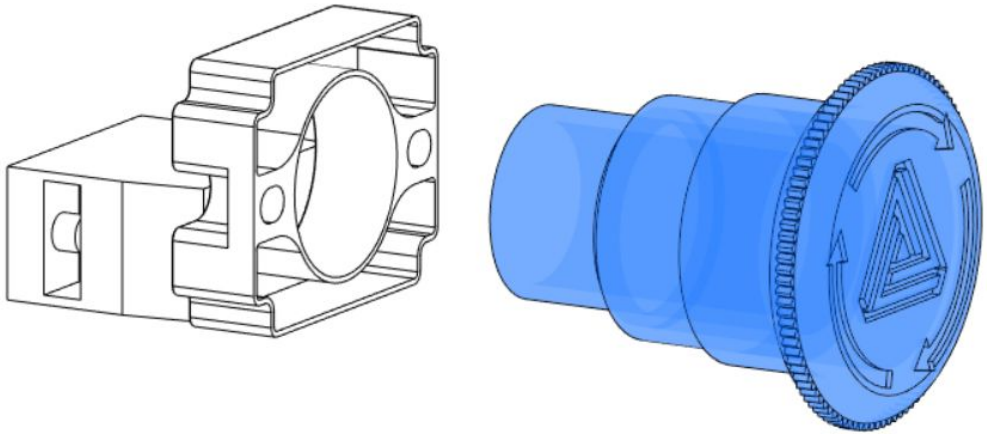
Electronics : PSU



Electronics : AC inlet



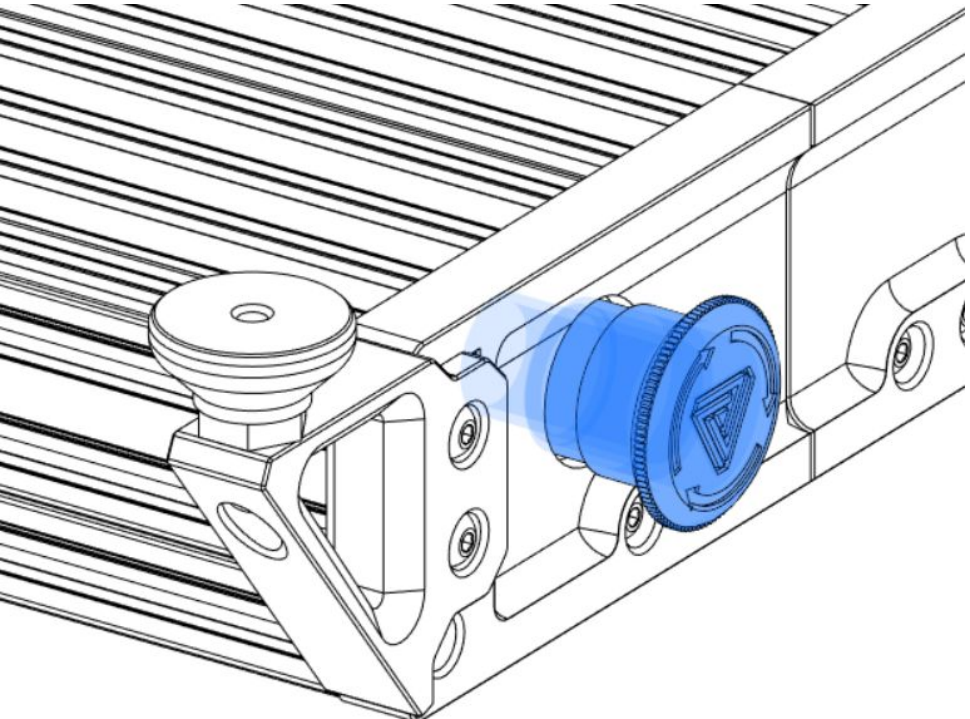
Electronics : E-stop

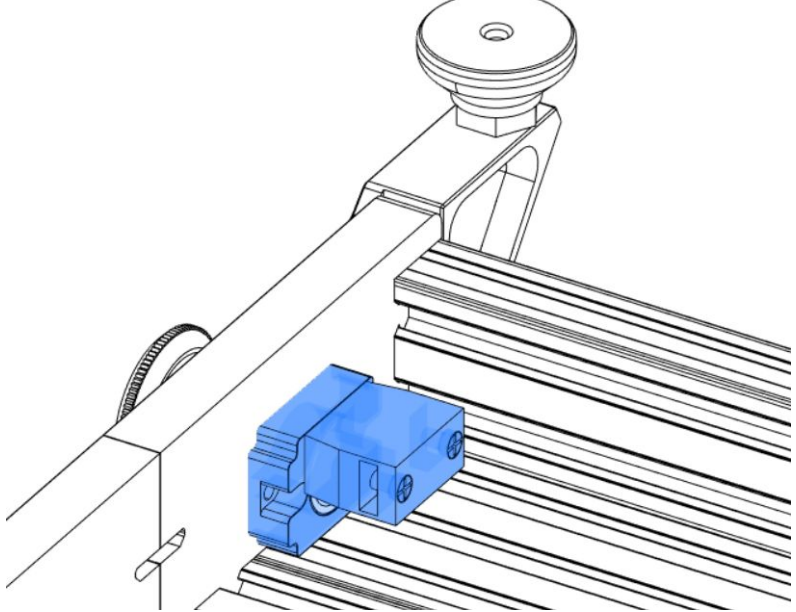


E-stop disassembly

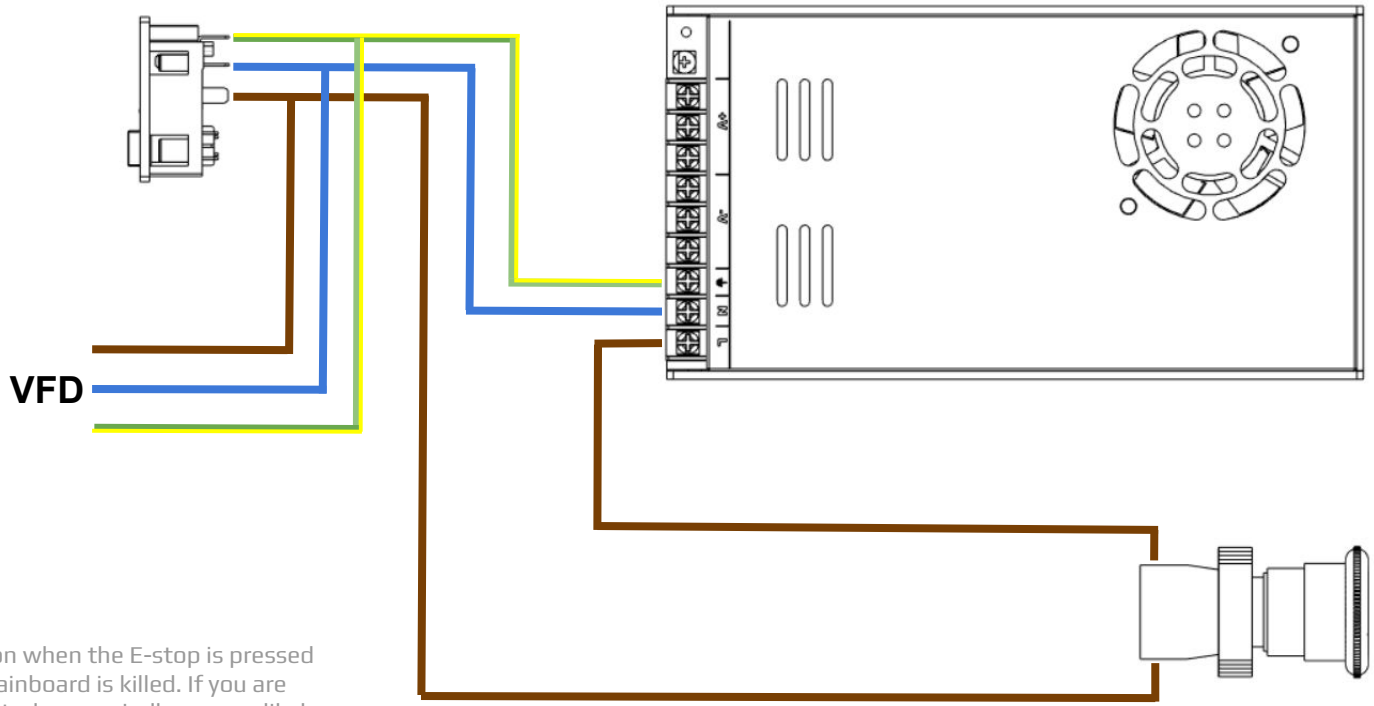
To install your E-stop you must first disassemble it. Most E-stops split in two by spinning the button side whilst keeping the body in place. To reassemble it perform the aforementioned steps backwards.

Electronics : E-stop





Electronics : AC wiring



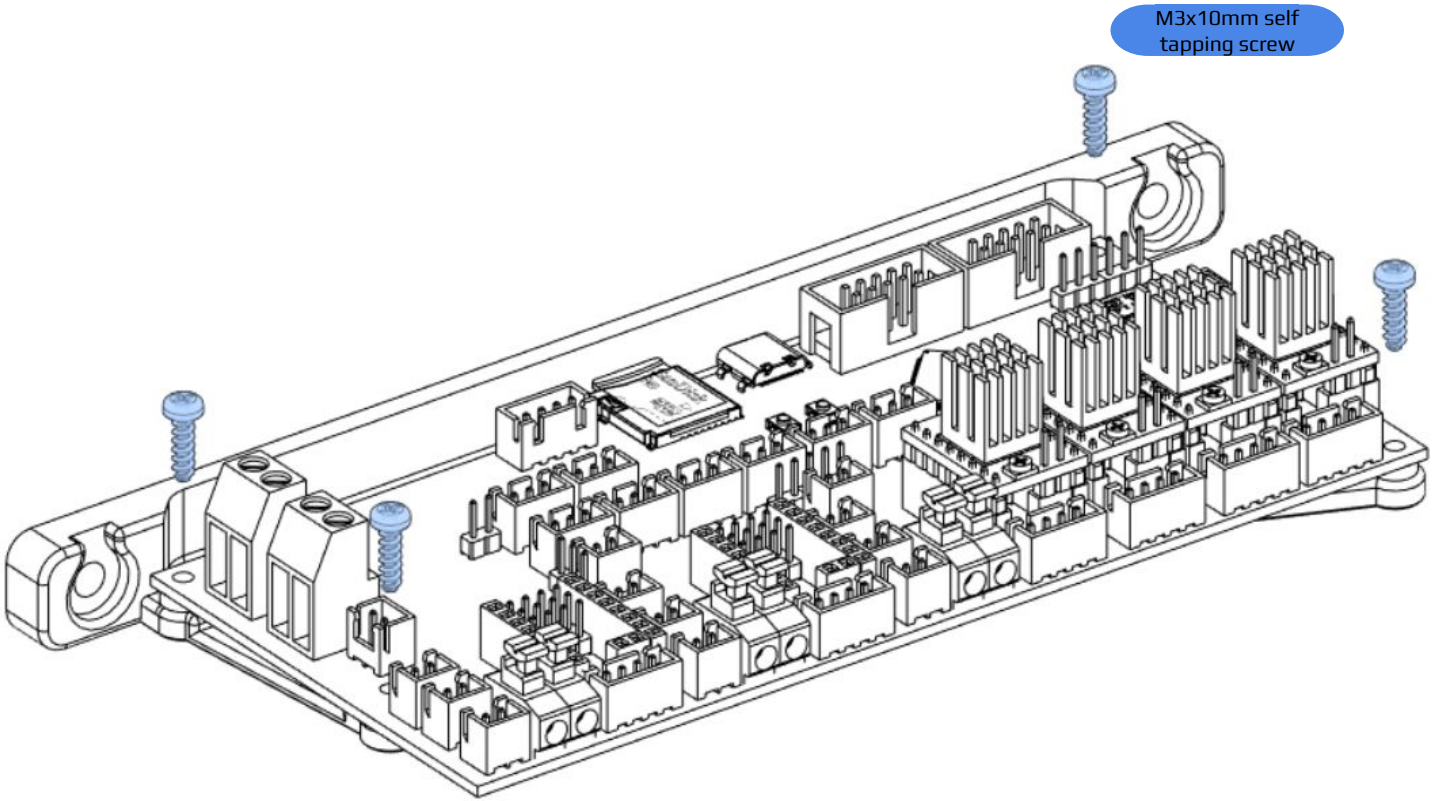
VFD AC power

If you plan to use a VFD to power your spindle you can split the AC input going to the PSU to power your VFD as seen in this diagram. Make sure that if you have a fused inlet and that the fuse is rated to your handle the load of both and make sure to consult your VFDs manual for its correct wiring.

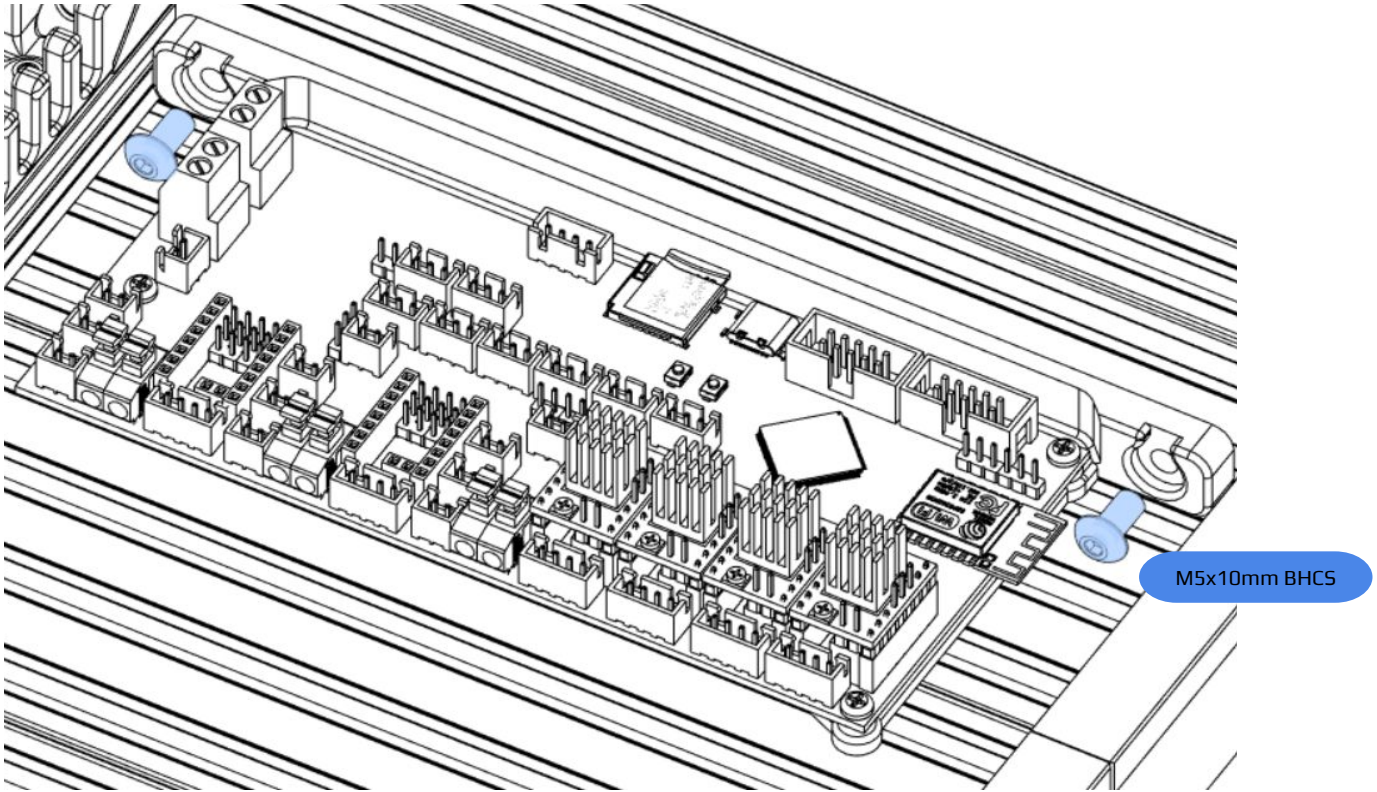
E-stop wiring

In this configuration when the E-stop is pressed all power to the mainboard is killed. If you are using a VFD to control your spindle you are likely to be using a PWM or analog signal from your mainboard to communicate with your VFD. When the power is killed the PWM/analog signal will drop to 0 and the VFD will bring your spindle to a halt.

Mainboard mounting



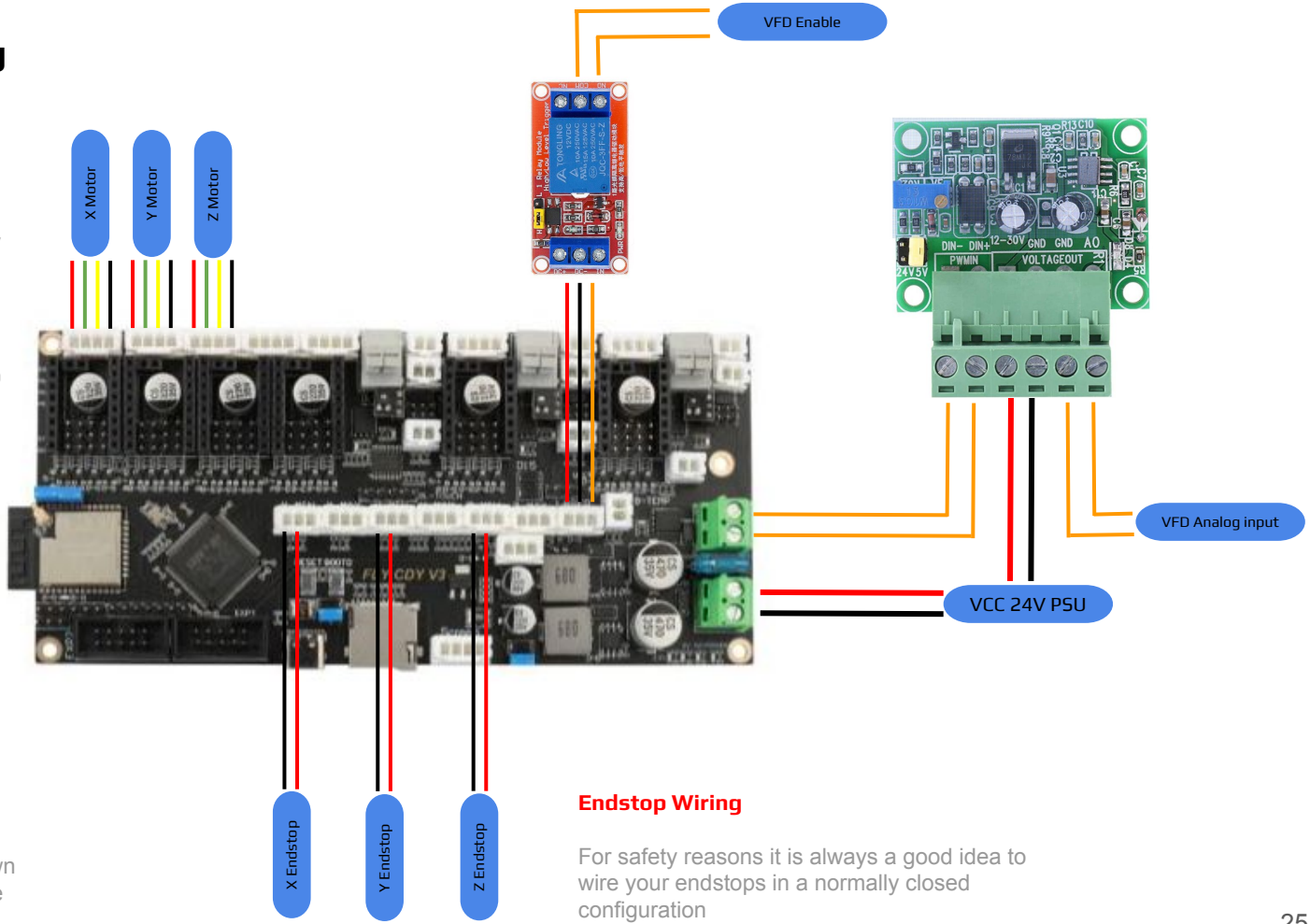
Mainboard mounting



Electronics : DC wiring

Daughter Boards

Depending on how you want to control your spindle you may need to add a few extra PCBs. This example is for a VFD controlled spindle that accepts a 0-10V analog signal to control spindle speed. You may also require a relay to send an enable signal to your VFD. If you wish to control your VFD manually ignore these extra components.



Legend

VCC + : ——

Ground : ——

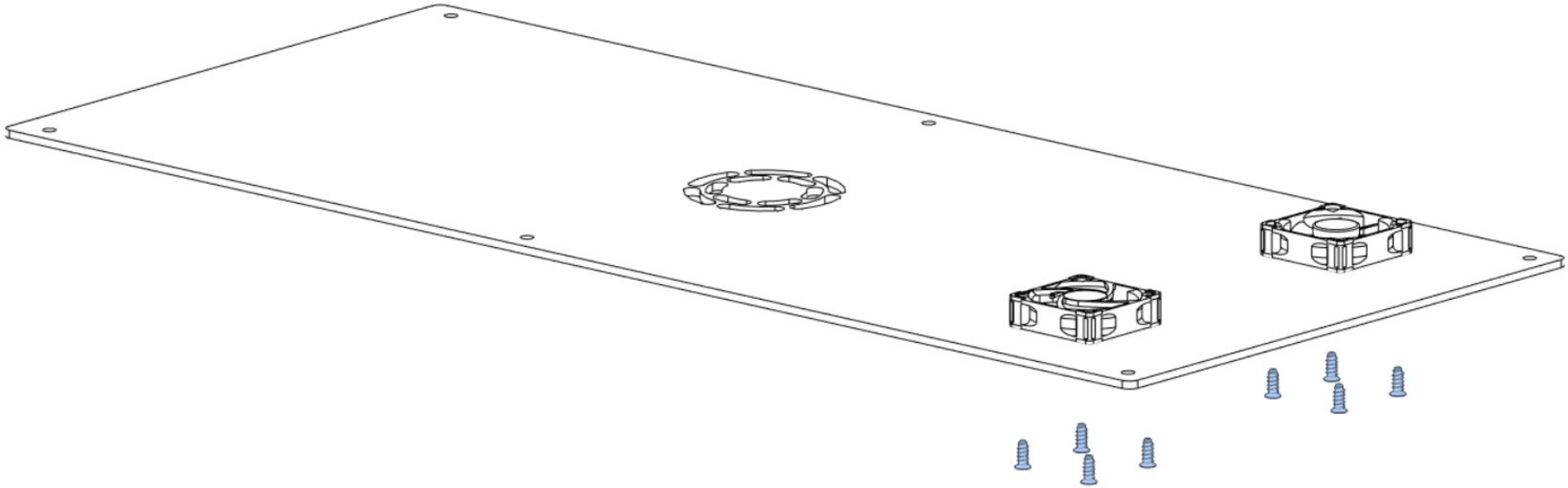
Signal : ——

Motor wiring : There is no standard for motor wire colors the motor wires shown in this example do no correlate with the rest of the legend

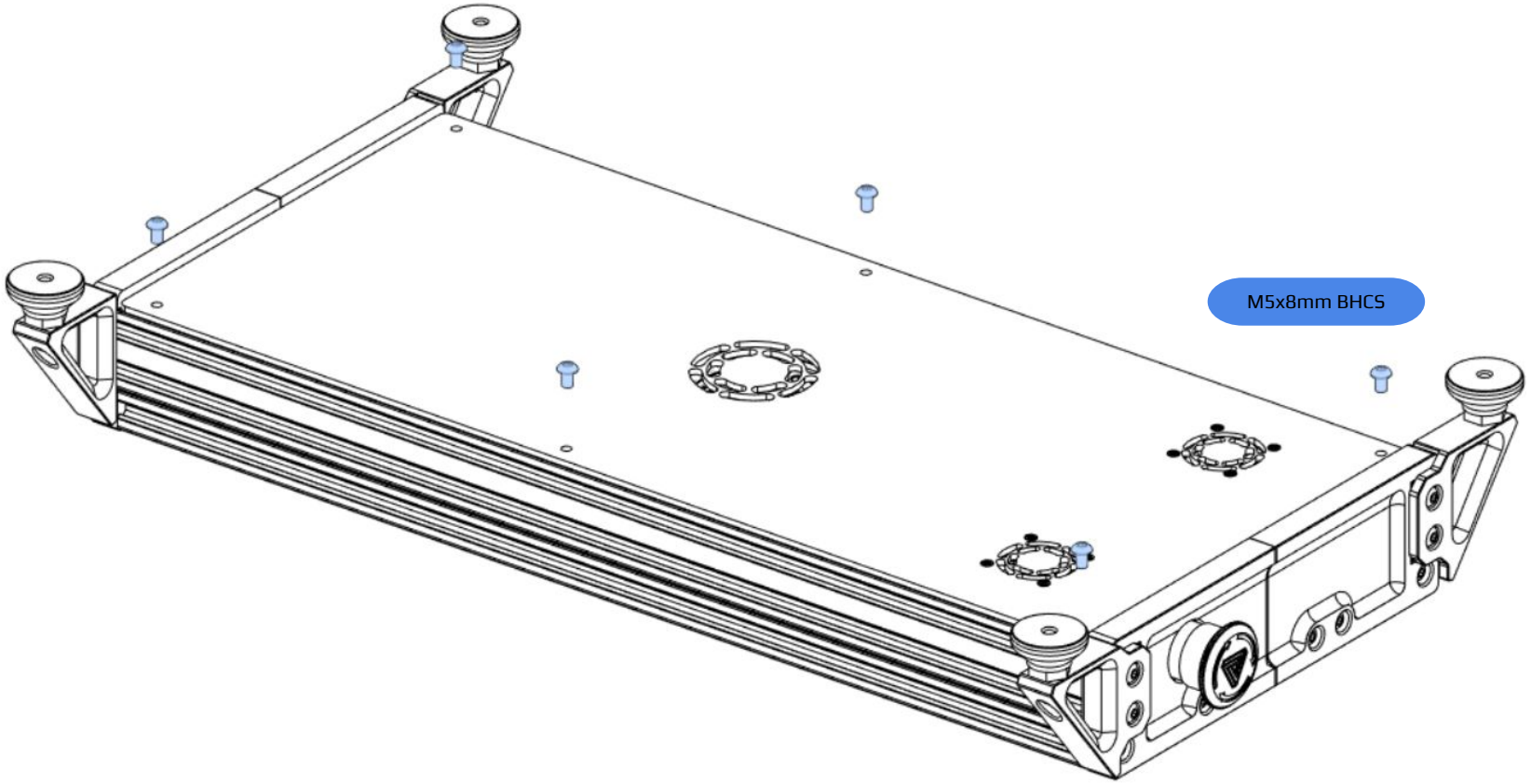
Endstop Wiring

For safety reasons it is always a good idea to wire your endstops in a normally closed configuration

Fan installation



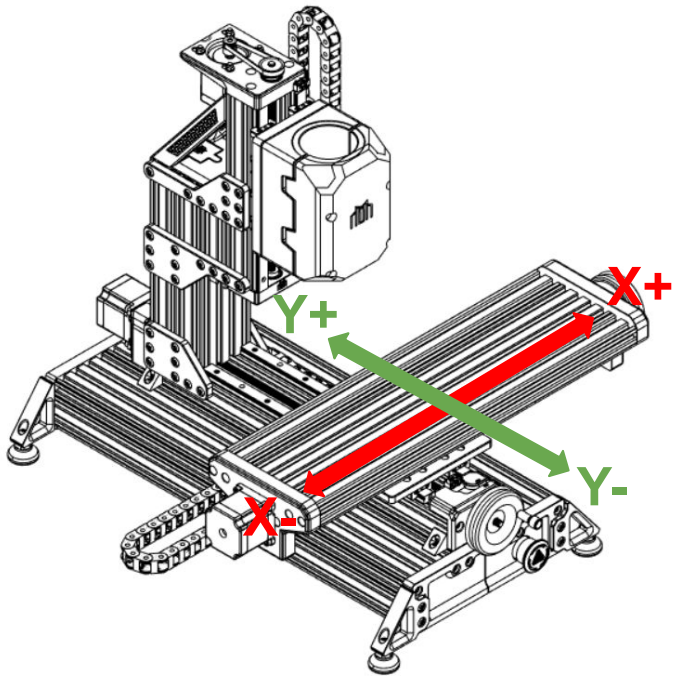
Electronics cover



Pre-flight checks

Axis Directions

Axes movements are based on relative spindle movements and not the movement of the work table, this may be confusing to new users but to simplify the setup process we've created a diagram based on the table movement instead, use this as a reference to setup your axes direction in your config,g.
For your Z axis the spindle should move up for Z+ and down for Z-



Endstop query

Ensure that all your endstops are working correctly, to do this press each endstop individually and enter M119 in the Duet web control console if it does not return as triggered check your wiring and or your endstop setup in config.g.

Homing check

Ensure that all you homing routines in homeall.g , homex.g , homey.g and homez.g all work appropriately.

Steps per mm calibration

You must ensure that when you move your axis X amount of millimeters it actually moves X amount of millimeters if you do not know how to do this you can follow this guide HERE.

Check backlash compensation

Using an indicator, check that all your axes for backlash, if you are not happy with the amount of backlash they have you can snug up the antibacklash nuts.

Pre-flight checks

Tram your work surface.

Make sure you tram your spindle appropriately to your work table. If you are using a fixture of some kind tram your spindle with reference to that instead.

Spindle speed control

First ensure your VFD is set up correctly according to it's manual. Then make sure to set your max spindle speed in config.g. Under the spindle section of your config find the line beginning with M950 and change the value found after "L" to your maximum spindle speed e.g. L24000.

Grease everything !

If you have not done so yet make sure that all leadscrews and rails are appropriately greased and or oiled. A good quality lithium grease is recommended for this job.