Creality K1 Max BTT M8P v2.0 Conversion

The "Moronic Voron" AKA The Moron By: Ghozt

Credit to Pellcorp for the name

Please read the document COMPLETELY before beginning this modification.

For questions or support, please ask in the Devil Design project page

Change-Log

Next Updates: New nozzle wipe macro. Carto 5.0 release updates.
8/21/24 - Updated printer.cfg and gcode_macro.cfg changing output_pins to fan_generic. Updated Carto Tap configuration for latest beta release.
8/20/24 - Added additional pictures and instructions to the SSR steps. Reworded some things for clarification.
8/19/24 - Updated toolhead flash instructions with confirmed process for M8P and stock K1 board. Uploaded copies of configuration files with download links.
8/18/24 - Copy and paste updated printer.cfg/gcode_macro.cfg files. Note toolhead flash process not tested.
8/10/24 - Corrected some instructions, elaborated on many steps for clarification. Added additional pictures.
8/8/24 - Re-organized document structure more logically
8/5/24 - Initial upload

Special Thanks

Big thanks go out to several people who helped me with this conversion and answered my questions along the way! ZIMZ, Pellcorp, Mackins VII, and Nandr0 – you guys were a HUGE help – so thank you! Some portions of text from the K1 M5P Conversion Guide were utilized as well – thank you to Haus and everyone else who worked on that document!

Overview

Summary

This modification replaces the stock Creality K1 Max board with a BTT Manta **M8P** v2.0. This opens up a significant performance increase and headroom for operations, allows the use of pure Klipper, and opens the door to more powerful stepper drivers with different voltages.

The instructions below are meant for the 2209 stepper set to 24v. Please read the entire guide before beginning the installation. There are also a number of modifications to the K1 Max used in this tutorial, with the changes documented in the included printer.cfg file. If your printer is different from the List of K1 Modifications below, then you will need to adjust the printer.cfg as necessary.

Note: You will lose Lidar functionality with this modification, it is highly recommended to replace PRTouch with an Eddy Current sensor (Carto, Beacon, BTT Eddy) or a standard Probe (CRtouch, Biqu, etc).

Disclaimer

Warning: **This modification involves electrical and mechanical hazards, intricate wiring, and detailed steps.** Only attempt this modification if you are comfortable working on complex wiring, electrical troubleshooting, and updating/troubleshooting Klipper configurations. **By proceeding with the installation using this guide, you acknowledge that you have read and understand it and are doing so at your own risk and accept all liability for anything that inadvertently or accidentally occurs during the installation process.**

K1 Max Modifications

This is relevant to the .cfg files if you download the included printer.cfg, sensorless.cfg, and gcode_macros.cfg files fully configured as this guide indicates.

1. Manta M8P with BTT CB2 and TMC2209 drivers
2. BootyGantry linear rail XY gantry with LDO 42STH48-2504 MAC (46mm shaft) steppers
3. Bondtech “Make It Fit” LGX Lite extruder
4. Meanwell 350w power supply (optional: for triple Z)
5. Cartographer Probe v3
6. Mandala 6mm Ultra flat bed
7. Keenovo bed heater
8. 40A 440VAC C-Lin Solid State Relay (stock SSR does not reliably operate the Keenovo heater)
9. Berserker Vindr 5015 K1 Max Bed fans
10. Bento mini air purifiers

Bill of Materials (BOM)

BTT Manta M8P v2.0 Integrated Control Board	Replaces stock Creality board
Raspberry Pi CM4 Compute Module BTT CB2 Compute Module	Compute module. CM4 is HIGHLY recommended.
Raspberry Pi CM4 heatsink BTT CB2 Heatsink [optional]	CM4 a heatsink is required. CB2 a heatsink is optional but recommended.
BTT TMC 2209 Stepper Motor Drivers	3 for X/Y/Z, or 5 for X/Y and triple Z
BTT Smart Filament Sensor 2.0 [optional]	Adds additional capabilities for failure detection
M3 screws (4x 20mm, 4x 3mm, 4x 10mm, 4x self-tapping)	To attach the board mount, Manta, fan mount, and fan
USB adapters [2 total]	Used to connect the camera, USB port, or Carto.
RS232 adapter(s) [2 total, if using Carto 1 total]	Used to connect the toolhead and bed MCU.
Slim 120x15mm PWM Fan	Used for MCU Fan, note this model is 12v
USB hub Adafruit mini USB hub [optional]	To add additional devices or utilize front USB port

Dupont Crimping Tool	To crimp wires for plastic terminals
XH 2.54 pre-crimped connector kit [optional]	Alternative option to crimping
Wire cutters and strippers	For cutting and stripping…obviously
Heat shrink tubing	Don't use electric tape. Please.
USB power supply	For M8P Preparation (most USB ports will work)
USB 2.0 cable 1 to 2 splitter	Need 1 of these. Right angle, 6 inches.
USB 2.0 cable [optional]	Might need 1 of these
BTT TFT50 v2.1 DSI LCD Display [optional]	BTT TFT 50 V2.1 Screen mount
FFC Ribbon Flexible Cable [optional]	For DSI screen connection
BTT HDMI7 Display [optional]	Requires USB and HDMI. Mackins BTT HDMI7 Side Mount
Micro HDMI cable, 3 feet [optional]	For HDMI screen
STLink flash programmer [optional]	One of the methods for flashing the toolhead MCU
16 gauge red/black wire	Primary power. Need ~2-3 feet
20 gauge red/black wire	For various connections. Need ~2-3 feet
Mackins M8P K1 Max mount & 120mm fan shroud	3d print this
My remix of the above mount [optional]	Or 3d print this. Relocated RS232 and added the USB adapter holders.
Mackins D3vils Hooves (feet) [optional]	3d print these to increase air flow
Solid bed mounts [optional]	3d print these if removing PRTouch
BTT SFS 2.0 riser mount [optional]	3d print this if upgrading the filament sensor
K1 Max 3rd party SSR mount	3d print this if upgrading the bed heater
16 gauge fork connectors or WAGO connectors	For making several wire connections. Need 2-4
Soldering iron	Needed if you use pre-crimped wires
Required tool you need to do this mod	Everyone needs this

Choose your Compute Module

The Manta M8P can run on a variety of compute modules including BTT's CB1/CB2, but it was designed to run a Raspberry Pi Compute Module 4 (CM4). **Based on personal experience, and feedback on the Voron forums, I highly recommend going with the Raspberry Pi CM4 (model CM4104032) for 4GB of RAM and 32GB of EMMC flash.** Using the link in the BOM it is only $10 more and MUCH more powerful.

A number of users reported issues with MCU timer too close or MCU timeouts when using the CB2. This was resolved by replacing the CB2 with a CM4 running a 32-bit image of MainsailOS. Whether this resolved the issue due to the more recent 32-bit kernel, or the addition of more raw horsepower, it seems to have solved it. The CB2 only comes with a 64-bit image based on a year old kernel, and there are known bugs with 64-bit OS. Communication timeout that 32-bit image resolved

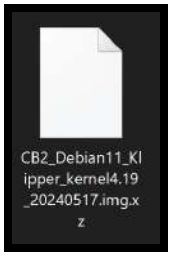
Manta M8P Preparation

Get the M8P Online

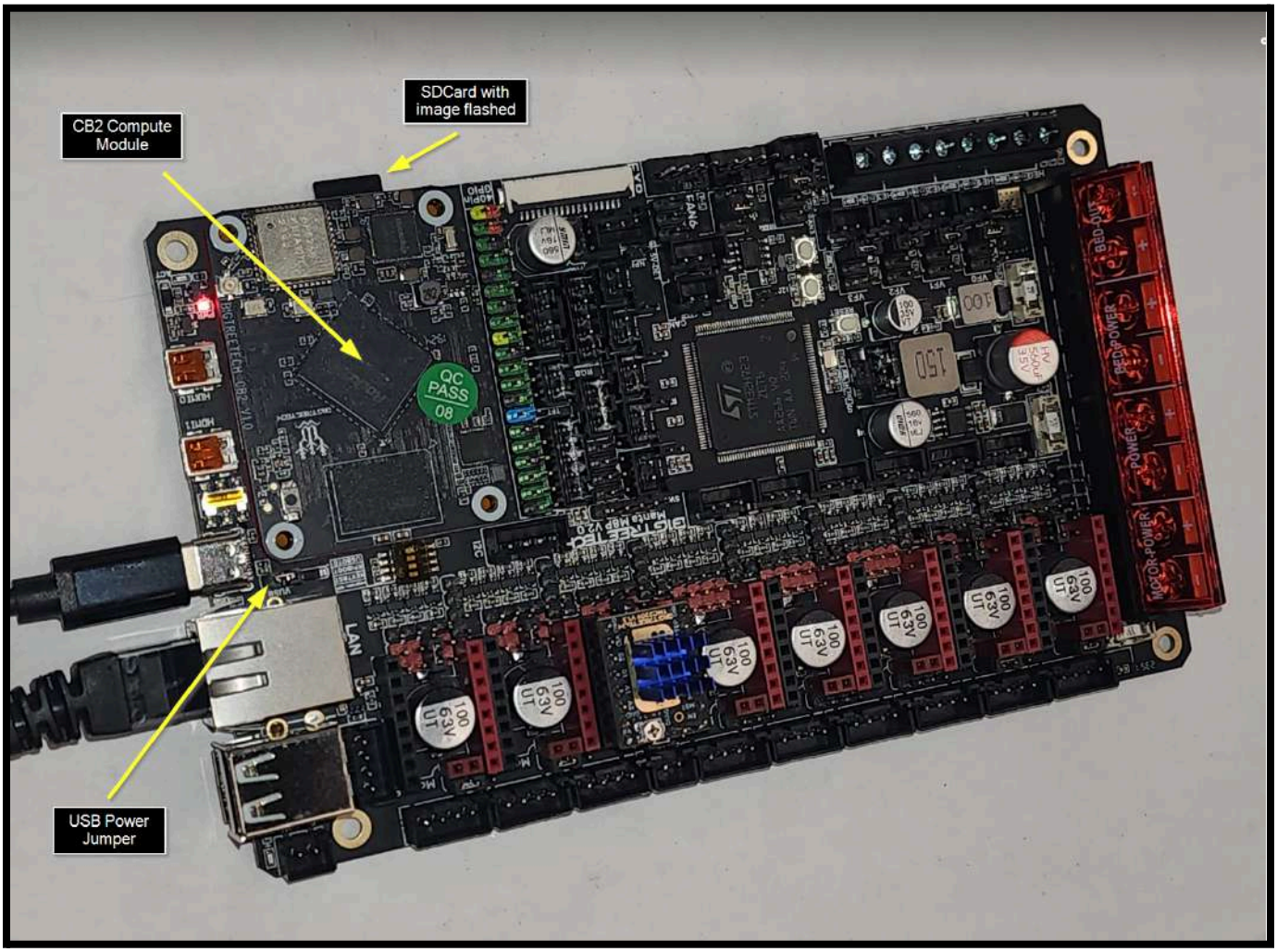
*Do yourself a favor and **GROUND** yourself while handling the M8P and or compute modules (CB1/CB2/etc). Electrostatic discharge (ESD) is real and can damage your components before you even start.*

*Verify you have a working **M8P** and compute module before taking apart your printer. My first **M8P** board had strange issues that required a replacement. It could have been from me handling the board honestly.*

1. Unbox the Manta **M8P** and **CB1**/**CB2**/**CM4** module and install the compute module onto the M8P. Line it up carefully, then press it onto the board. It requires a little bit of force to seat it fully. You will hear it click when fully seated. You do not need screws.
    a. For CB1 and CM4 you will want to install a heatsink. For CB2 it is optional.
2. Flash your compute module. Download Raspberry Pi Imager
    a. CB2: Follow the CB2 instructions to flash the CB2 Klipper image and internal memory. (I extracted the .img file from the .xz using 7zip and selected the .img file itself to burn to the SDCard)
    b. CM4 Lite (no EMMC): Follow the CM4 instructions (page 20) to flash the Raspberry Pi image and internal memory. Alternatively, here is a Video tutorial (not my video)



    c. **CM4 (with EMMC) skip to Flash the EMMC (CM4) section, you do not use an SDCard to flash CM4's EMMC, you use your computer with a USB-C cable**
3. Insert the SDCard in the **M8P** SDCard slot on the long side of the board (SOC)
4. Install the USB power jumper (2 pin next to the USB ports) Note: This **NEEDS** to be removed if you have 24 power connected or you will **DESTROY** the USB hub
5. Connect to your network via Ethernet (or edit the config file to configure your wireless network) and power on the device via the USB-C port using a USB-C power supply (Raspberry Pi power is recommended, but most should be sufficient to program the unit)
    a. I've received some reports that the device only booted using 24v, this may be due to an insufficient USB power supply. If you are running into issues with USB power, then REMOVE the USB power jumper you just installed and try direct 24v power to program the board.



With the SDcard inserted with a correctly flashed image, when powered on, you will see green activity lights blink in one of the M8P's corners. If you do not, please revisit the image flash instructions and/or try a different SDCard.

6. Locate the device's IP (use Advanced IP Scanner to scan for it, the hostname begins with btt)

Flash the Primary MCU

1. Connect via SSH using Putty or similar method to the printer's IP address

2. In the SSH console, login using biqu/biqu and then type the following command:

ls /dev/serial/by-id

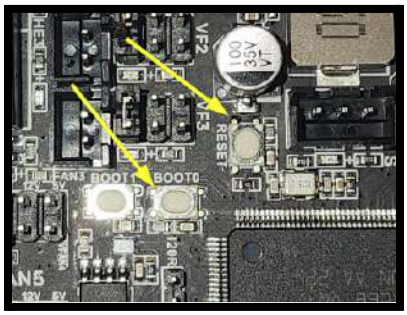
3. You should see a single device with the word MARLIN in the name or similar – this is your compute module (**CB1**, **CB2**, or **CM4**, etc). If you do not, please reseat your compute module on the Manta **M8P**. You should hear a firm click on both sides as it is pressed onto the board.
4. The most reliable method in my experience was DFU mode for flashing the M8P to match the compute module. In the SSH console; type the following commands and select the following configuration options:

cd ~/klipper

make menuconfig

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)
    Processor model (STM32H723)  --->
    Bootloader offset (128KiB bootloader)  --->
    Clock Reference (25 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    USB ids  --->
()  GPIO pins to set at micro-controller startup
```

5. In the SSH console, type the following command:



make

6. On the **M8P** itself, with it still powered on, hold down the **BOOT0** button and press the **RESET** button, then release **BOOT0**
7. In the SSH console, type the following command:

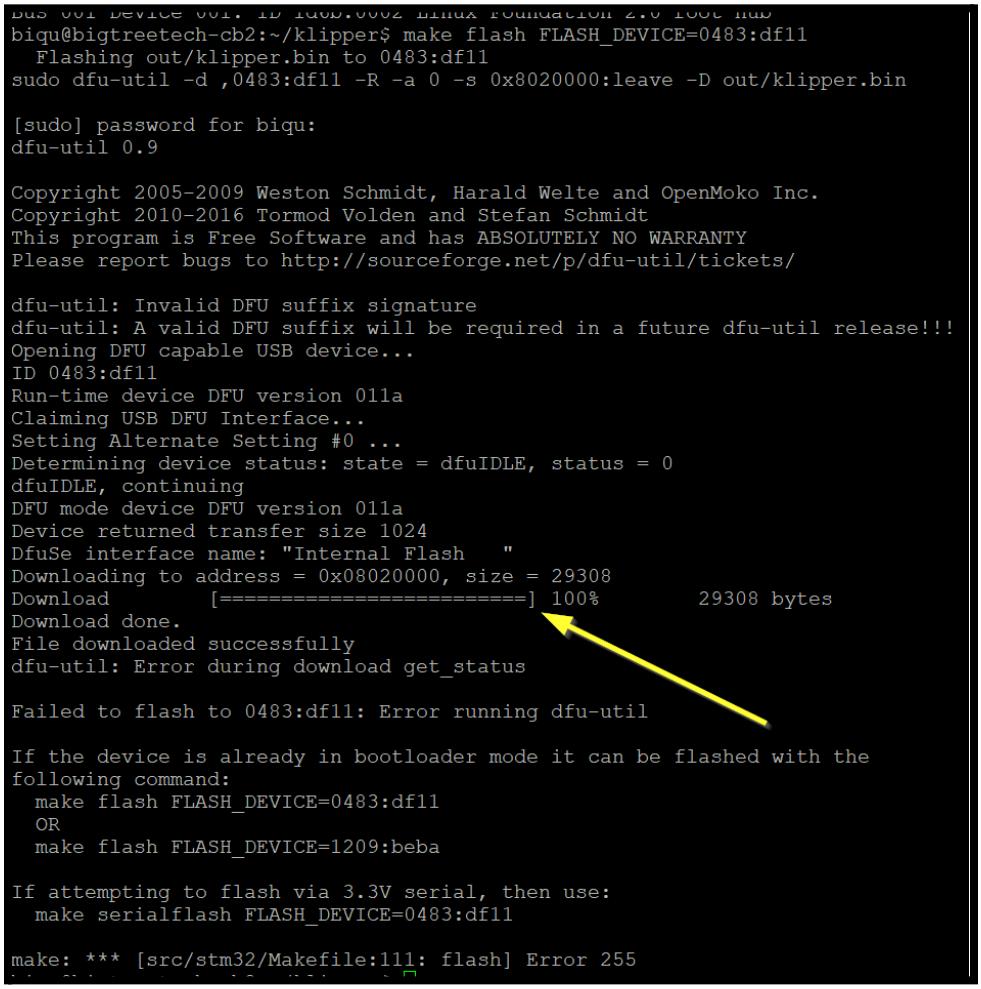
lsusb

8. Make sure you see a device with the words “in DFU Mode” at the end. Confirm the device ID before the device name matches the next step (if not, update the device ID as needed). Then enter the following command via console.

```
biqu@bigtreetech-cb2:~/klipper$ lsusb
Bus 008 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 007 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 004: ID 0483:df11 STMicroelectronics STM Device in DFU Mode
Bus 005 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

make flash FLASH_DEVICE=0483:df11

9. You should see the following screen to indicate the flashing completed successfully. **It will kick out what looks like an error, but as long as the progress bar reaches 100%, then it completed successfully.**

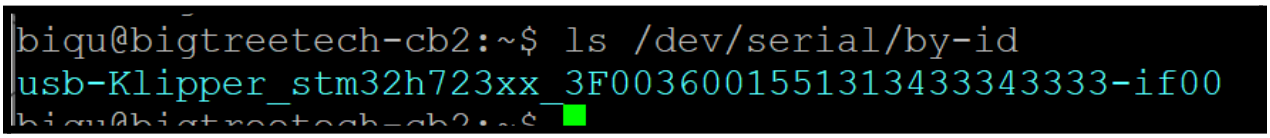


10. Please reset the Manta **M8P**.

If the device does not boot with a blinking green activity light, reseat your CB2 (or similar) on the M8P Manta.

11. Once logged back in, issue the following command and verify a device is listed with KLIPPER in the name now. **This confirms the flash was successful.**

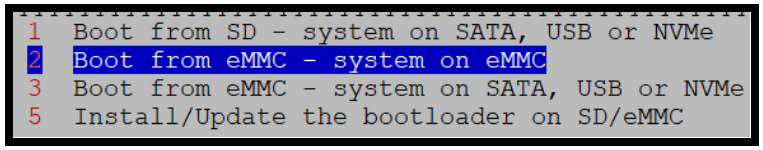
ls /dev/serial/by-id

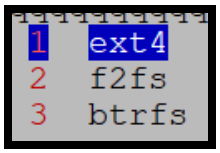


Flash the EMMC (CB1/CB2)

1. For increased performance, it is highly recommended to flash the SDCard to the **M8P's** EMMC (internal memory). This step is technically optional. Use the following command and options via console:

sudo nand-sata-install

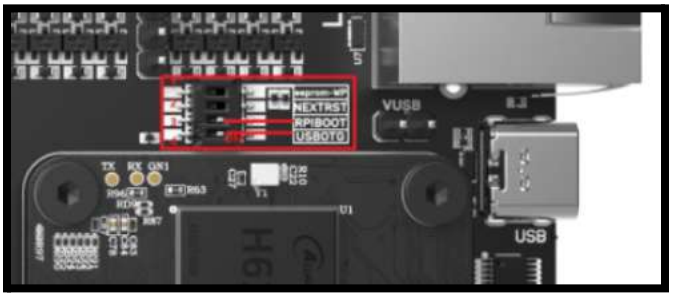




2. Wait for the flash to complete, with a CB2 it took about 2-3 minutes.
3. Once complete, select Power Off in the SSH console and remove the SDCard
4. Power up the Manta **M8P** again to test that it boots up with no SD Card. You should see green activity lights blinking about 5-10 seconds after power is applied. If you do not, then the device is not booting. Connecting a monitor to the mini-HDMI port can assist with troubleshooting.

Flash the EMMC (CM4)

1. Use instructions at this link starting on page 24
2. Download an image
    a. MainsailOS 32-bit
3. Download and install Rpiboot.exe
4. Flip the RPIBOOT and USBOTG DIP switches (3 and 4) to ON



5. Connect your computer to the USB-C port on the Manta M8P using a USB-C cable.
6. Run rpiboot.exe
    a. It installs under C:\Program Files(x 86)\RaspberryPi by default
7. Using Raspberry Pi Imager, flash the image you downloaded to the storage device that appears now
    a. Edit settings
        i. Create your admin account
        ii. Create WIFI configuration
        iii. Update hostname (optional)
        iv. Enable SSH
    b. Wait for flash to complete
    c. Reset DIP switches 3 and 4 from Step 4 to OFF
    d. Reboot **Manta**

Manta M8P Setup

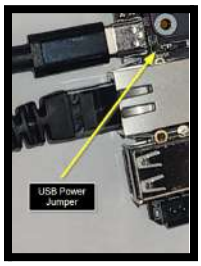
Initial M8P Configuration

1. Use kiauh to install components if you wish, for CB2 I used it to install Mobileraker and change the printer's hostname. For CM4 I used it to install Klipperscreen and Mobileraker.
2. Install Kamp from github directly (optional)
3. Install Cartographer from their website directly (optional)
4. Install Shaketune from their github (optional)

5. Uninstall Obico from your CB2 image (highly recommended if you are not using it)
6. Proceed with uploading the provided sensorless.cfg and gcode_macro.cfg.
7. Review the printer.cfg at the bottom of this document. Use the printer.cfg as a reference point to find PINs/etc, and upload a copy of gcode_macro.cfg to have some basics to start with.

Keep in mind the included configuration files are for wiring things *exactly* as I have shown in the picture. If you use different connections on the board, you will need to adjust the PIN configuration to reflect these changes. You also may need to update device paths for your compute module, toolhead MCU, and/or Cartographer in the printer.cfg file.

8. Save any changes and shut down the printer using the web GUI. Disconnect USB-C power.
9. **THIS STEP IS IMPORTANT – REMOVE THE 2-PIN JUMPER YOU ADDED NEAR THE USB PORTS DURING TESTING/FLASHING. IF YOU DO NOT YOU WILL DESTROY THE USB HUB ON THE M8P WHEN YOU CONNECT 24v POWER**

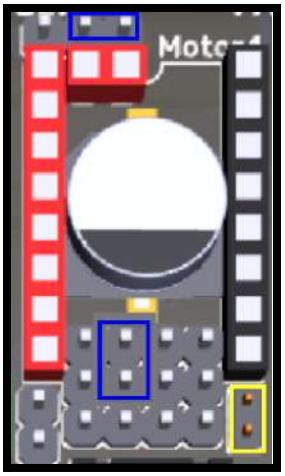


Manta M8P Board Prep

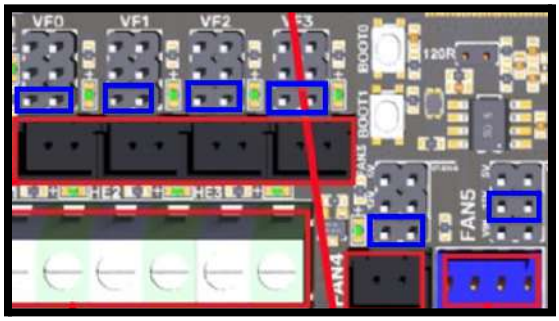
1. Install the jumpers on your stepper and fan outputs. The below instructions assume you are using 24VDC and TMC2209s in UART mode. For more detailed info on the stepper jumpers consult the M8P v2.0 guide.
2. For each motor stepper driver, install a jumper on the pins with blue rectangles around them, and for only the X/Y motors install a jumper on the pins with the yellow rectangle around them (DIAG pins)
3. For the fans and LED strip, set the jumper to 24v. The closest position is 24v, the next is 12v, and the furthest away is 5v
    a. **I have FAN5 (MCU fan) configured for 12v, and the rest for 24v. Adjust as needed for your fan(s).**

Stepper driver jumpers

(using jumpers, bridge blue pins for all used steppers, bridge yellow(diag) pins for X/Y only!)



Fan jumpers



VIN is +24v

Fan Tests

Test each fan with a separate power supply. I used a small spare 24v LED power supply I had laying around. I found that several fans were wired backwards (the black wire with white marks is typically ground and in my case was +V on 2 of my fans) so do not trust Creality's wire colors.

WARNING: It is not recommended to test these using the printer's power supply directly. Death or serious injury may result from attempting to work on your printer while it is plugged into AC power.

K1 Max Disassembly

WARNING: Disconnect your printer from AC power before proceeding. Death or serious injury may result from attempting to work on your printer while it is plugged into AC power.

1. If you are removing PRTouch, please 3D print your solid bed mounts now if you have not. **Double-check the BOM that anything that needs to be 3D printed has been completed (or you have a spare printer to print).**
2. Power off your printer, disconnect power/Ethernet and remove the bottom cover. The bottom cover will need to remain off for everything to fit. It is recommended to print and install Mackins D3vil's Hooves "shoes" or similar to raise the height of the printer, to improve cooling room, but not necessary.
3. [Optional] If you have a Cartographer and have not removed the PRtouch system, you may do so now. Remove the heatbed and peel off the adhesive insulation sheet. Remove all 4 load-cells and the MCU in the middle (from underneath). Install the solid bed mounts for increased rigidity for the heatbed/bed mount.
4. Using a blowdryer (optional; do **NOT** use a heat gun) carefully heat up the connectors on the stock Creality board (to soften the yellow glue) and pick the glue off the connector with a pair of tweezers, small flathead, or needle nose pliers. Disconnect each plug as you clear most of the glue off of it. Attempting to pull the wires, or use pliers to wrestle the connector off will likely result in a wire breaking or connectors becoming damaged – the glue is STRONG. Even if you don't need the board anymore, you can sell it on eBay or something unless you Hulk Smash it.
5. Pull towards the top of the printer on the OEM display on the front to disconnect it. Remove the tape from under the printer that holds the ribbon cable in place. Remove the OEM screen completely as it cannot be controlled by the Manta **M8P**.
6. Some replacement screen options: TFT 5.0v2 (connects via DSI cable) or BTT HDMI7 (utilizes a USB port for power/touch control and mini-HDMI for display).
7. Cut all zip ties holding wires on the bottom of the printer (be careful not to clip any wires)

Manta M8P Installation

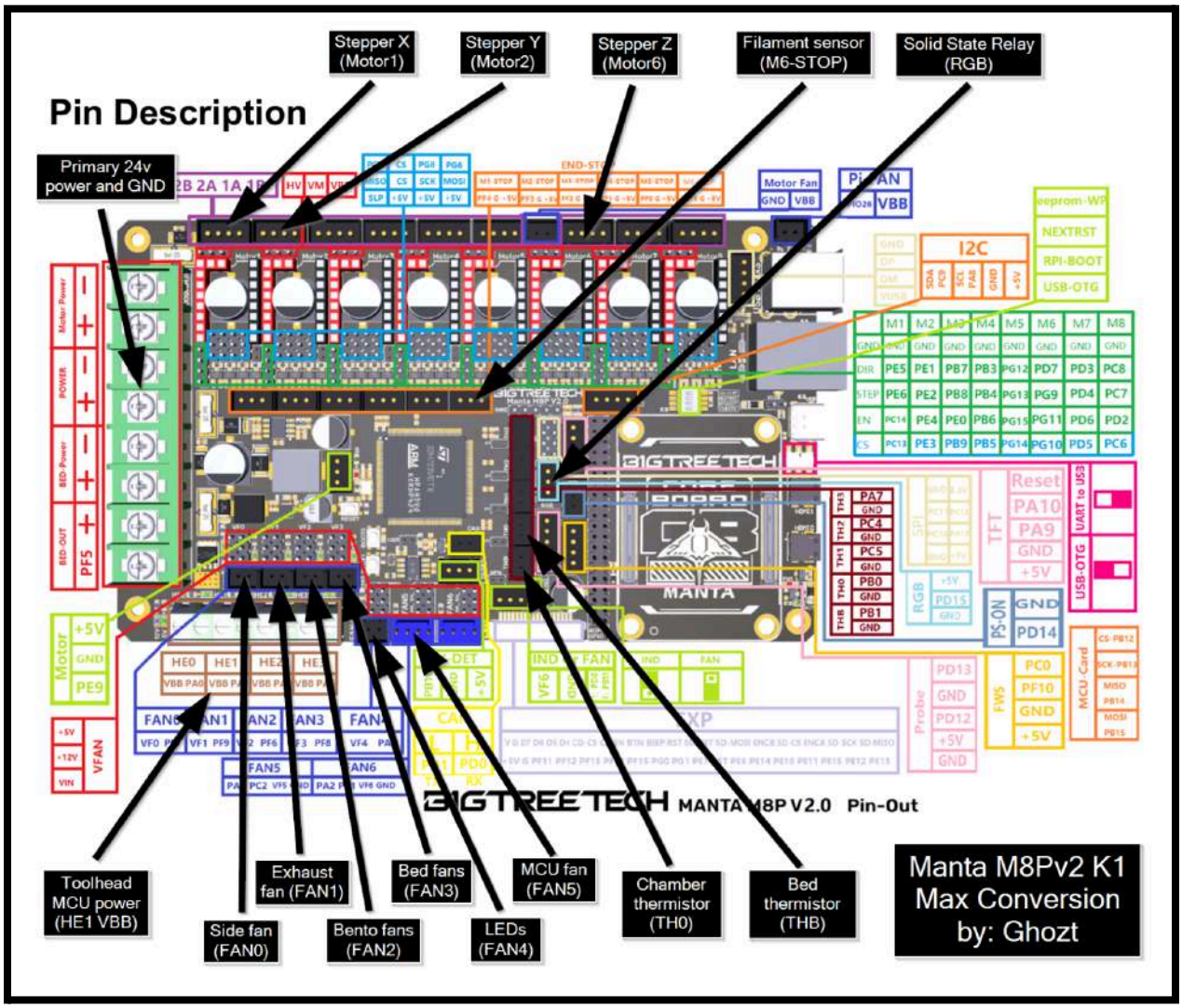
Identify The Wires

Wire locations are based on the printer laying on it's back

LABEL	LOCATION	COMPONENT
J9	Top Right	Camera
USB	Top Right	Front USB port

USB-J55	Bottom Right	LIDAR cable (use for Carto)
232	Bottom Right	Toolhead MCU
232	Bottom Left	Bed MCU
260c	Bottom Right	Chamber thermistor
TB	Bottom Left	Bed thermistor
Filament	Bottom Right	Filament sensor
LED	Top Right	LED strip
FAN	Bottom Right	Side fan
FAN2	Bottom Right	Back (exhaust) fan

Identify The Connections

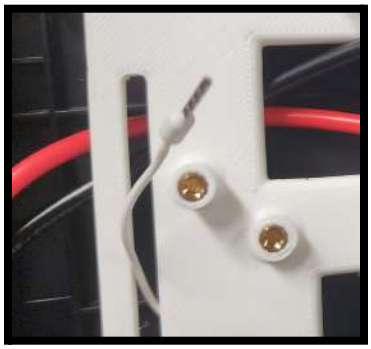


RS232 Serial Adapters (Toolhead/Bed MCU)

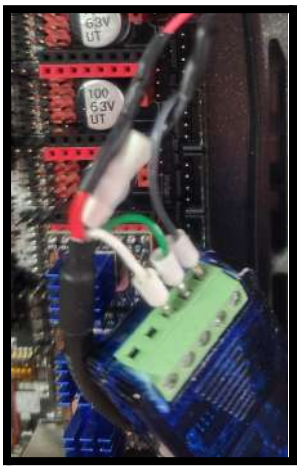
1. RS232 serial adapters are used for interfacing with the toolhead and bed MCU. You need one each for the toolhead MCU and bed MCU. If you are using Cartographer, you only need **one of these** since you can remove the bed MCU entirely.

2. Toolhead MCU: Take the RS232 labeled wire coming from the bottom right of the printer and cut off the connector and strip the exposed wires.
3. The **black** ground wires need to connect in a 3-way
    a. Option A: Solder the wires together: the ground from the toolhead MCU wire, an 18cm length of 20-gauge wire, and a 5cm length of 20-gauge wire. Strip the wires and solder the 3 together at one joint.
    b. Option B: You can strip the toolhead MCU wire, and an 18cm length of 20-gauge wire, twist them together, and insert them into the RS232 adapter on the **GND** pin. I was not able to get these two wires together into a single ferrule that would still fit into the RS232's openings.)
    c. Option C: Strip as above, but join the wires using a Wago lever nut.

4. Connect the available end of the 5cm **black** ground wire into the RS232 adapter on the **GND** pin (the last pin if looking at the screw-terminal side)



5. The longer 20-gauge **black** ground wire's other end will be joined together with the 16-gauge primary **black** ground wire going to the **M8P**'s POWER input in the *Connections:Main Power* step below. **So, leave the other end disconnected for now; but route it under the board mount for ease of installation/removal later.**
6. The **red** power wire on the toolhead MCU wire gets extended using a 22cm length of 20 gauge wire, terminated and connected to the **VBB** pin on **HE01** on the **Manta M8P**. Run this under the board mount and through one of the wire organizer holes for ease of installation/removal.
7. The green/white (TX/RX) wires are connected to the next two pins on the RS232, in that order

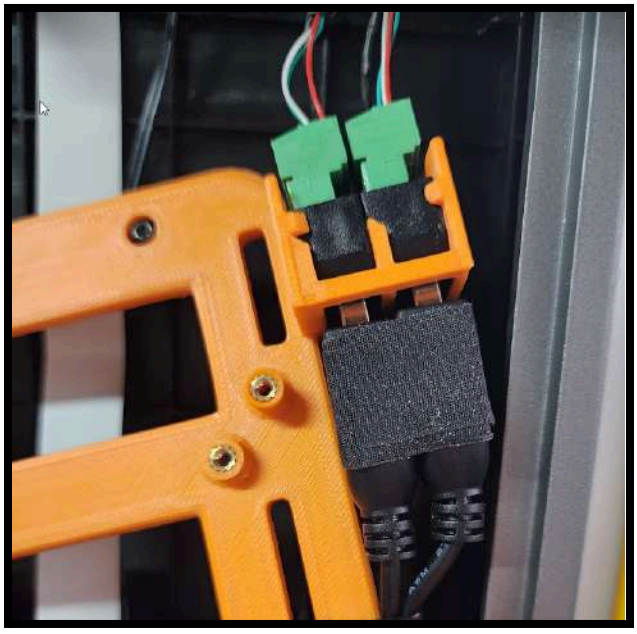


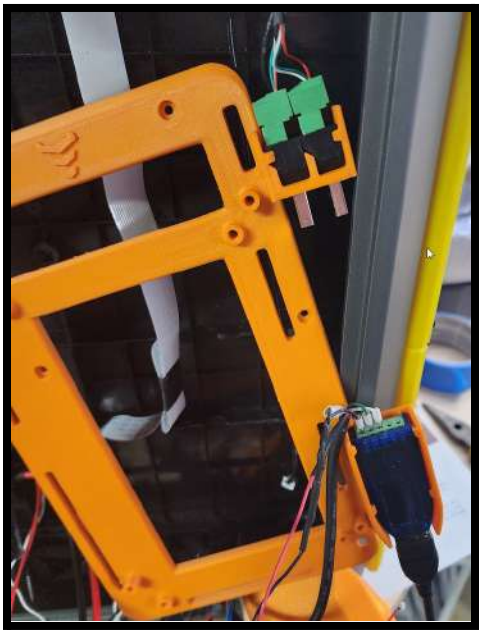
8. For the RS232 adapter that's for the bed PCB (coming from the bottom left of the printer), if you are using Carto you can completely remove this cable. If you are using the stock PRTouch leveling system, please reference the M5P guide for how to wire up the bed PCB
9. Insert the RS232 adapter into the slot on the 3d printed board mount. Since my remix is designed for only 1 RS232, if you are using a second RS232 adapter for the bed MCU, you can try stacking them and holding them together to the board mount with a zip tie.

USB Adapters (Camera, USB Port, Carto)

1. USB adapters are used for the stock camera, the front USB port, and/or the Lidar cable on a K1 Max which typically gets reused as the Carto cable.
2. For the 2 **USB** (4 pin) connectors hanging in the top right corner, simply cut off the connector, strip the wire ends, and insert the wires into the screw terminals as indicated in the picture.
    a. USB pin order: **Ground**, Data+, Data-, +Voltage
3. Tighten the screw terminals and give them a “tug” test to verify they don’t come out easily.

4. Insert the USB adapters into the slots on the 3d printed board mount.
5. Connect your USB splitter and extenders to the USB ports on the adapters you just wired up





Board Mount Installation

1. You will need to trim some of the plastic underneath, as indicated in the picture. If using my remix, there are also some “fins” on the very edge underneath the frame you will need to remove: you can cut them with a wire cutter and use pliers to pull/rip them off.

2. Install the 3d printed board mount using the self-tapping screws that held on the Creality board. Make sure the mount sits flush to your liking and trim additional plastic if desired.
3. Depending on how you are routing your wires, you may want to tuck the USB splitter underneath where the Manta M8P will go, just position the wires carefully to not interfere with installation.

Install Your Display

1. If using a new display then install its mount and the screen now. Route the ribbon cable or HDMI/USB cables underneath the board mount.
4. Connect the DSIO monitor cable to the back of the **M8P**. The ribbon cable has the metal pin contacts facing AWAY from the board (contrary to what you see normally).
5. For DSI, you will need to download the driver using the following command in the SSH console to download and activate DSI1 as the primary display (the default is HDMI).

sudo wget https://datasheets.raspberrypi.com/cmio/dt-blob-disp1-cam1.bin -O /boot/dt-blob.bin

6. For HDMI displays, the **M8P** defaults to HDMI so the only configuration you need is if you need to rotate the screen. You will need to reconfigure the touch screen if you rotate the display.

Mount The M8P

1. Install the **Manta M8P** onto the board mount using 4x 5mm M3 screws (any style).
    a. Keep in mind you may need to remove it in a later step to route some of your wires, but install it now so you can figure out wire lengths and get everything terminated.
2. Celebrate your progress with a **beer** [step not optional]
    a. If you cannot stop at one then please resume work tomorrow

Wire Prep / Termination

Note: The below wiring instructions are for a Manta M8P v2, if you have a v1/1.1 the PINs are different

Tips:

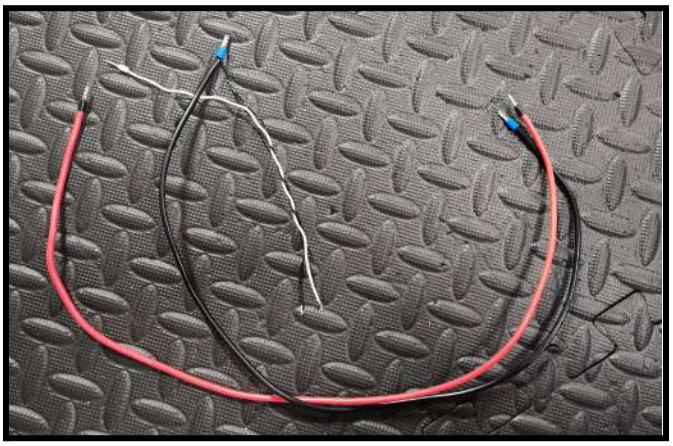
1. If you are going the wire termination route and notice the ends keep breaking off the wires or similar, you may be crimping too hard.
2. Either solder on pre-terminated wires or crimp the wire ends and insert the end carefully into the plastic header making sure they are fully seated (the metal tab will be visible all the way to the end, not halfway through).
3. After crimping/inserting, give them a decent “tug” test to make sure they don’t come right back out and/or break the wire. You want to find these problems now, rather than later.
4. Crimpers with a thinner width for the crimping surface were easier to use, larger ones may be more difficult to use and/or destroy the metal terminal.

5. I found a nozzle declogging tool was perfect to help seat the metal tab in the plastic header, pushing it from behind, as pushing on the wire can just cause it to bend.
6. A tiny dab of superglue on the wires where they insert into the header can be used to help secure your wires from breaking at the joint.
    a. I found the M8P's included wire connections inserted *very* tightly into the ports and I broke several wires just trying to unplug something. Other 3rd-party pre-terminated plugs fit in more easily and didn't require as much force to remove. Dabs of glue eliminated the issue.

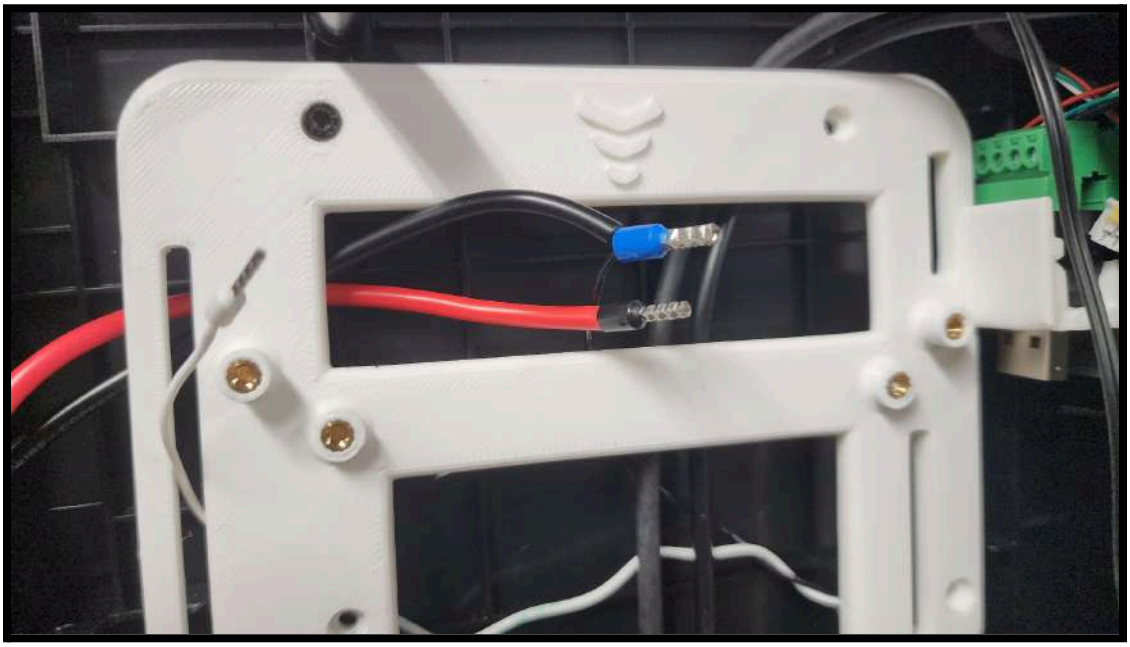
Connections:

Main Power

1. Measure and cut 55cm of 16-gauge **red** and **black** wires to connect to the **M8P** at the POWER input at the top of the board.
    a. Route them along the channels used for the stock wiring.
2. Terminate one end of the **red** wire and one end of the **black** wire each with a fork terminal or ferrule.
3. Terminate the other end of the **red** wire with a fork terminal or ferrule.
4. Twist together the other end of the **black** 16-gauge ground wire with the loose end of the 18cm 20-gauge **black** wire from the toolhead PCB cable prepared in an earlier step.
    a. Run the 18cm **black** ground wire under the board mount before doing this for ease of installation/removal.
5. Terminate the combined **black** wires to a single fork terminal or ferrule.



6. Connect the **red** wire to **V+** on the power supply, and the other end to **POWER+** on the Manta
7. Connect the side with the *single* **black** wire to **V-** on the power supply
8. Connect the side with the *combined* **black** wires to **POWER-** on the **M8P**.
    a. Run these wires under the board mount and have them pop out at the opening at the top.



LED strip

1. Route the 2 pin **LED** connector around the top left edge of the Manta, and plug it directly into **FAN4**

Carto USB

1. Cut off the terminal on the **USB-J55** connector and strip the wires.
    a. Terminate to a 4-pin JST. With the notches on the connector facing up, the wire order will be : **+V**, White, **Green**, **Ground**
    b. Connect this to the USB header directly behind the two USB ports on the **M8P**

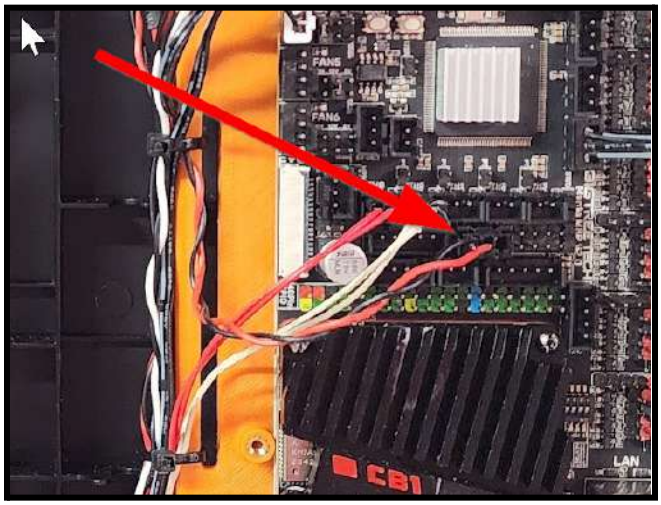
Bed thermistor

1. If you are using a Keenovo bed heater or stock bed heater, trim the thermistor wire labeled **TB** as needed.
    a. Terminate to a 2 pin JST connector (polarity does not matter) and plug into **THB** on the **M8P**

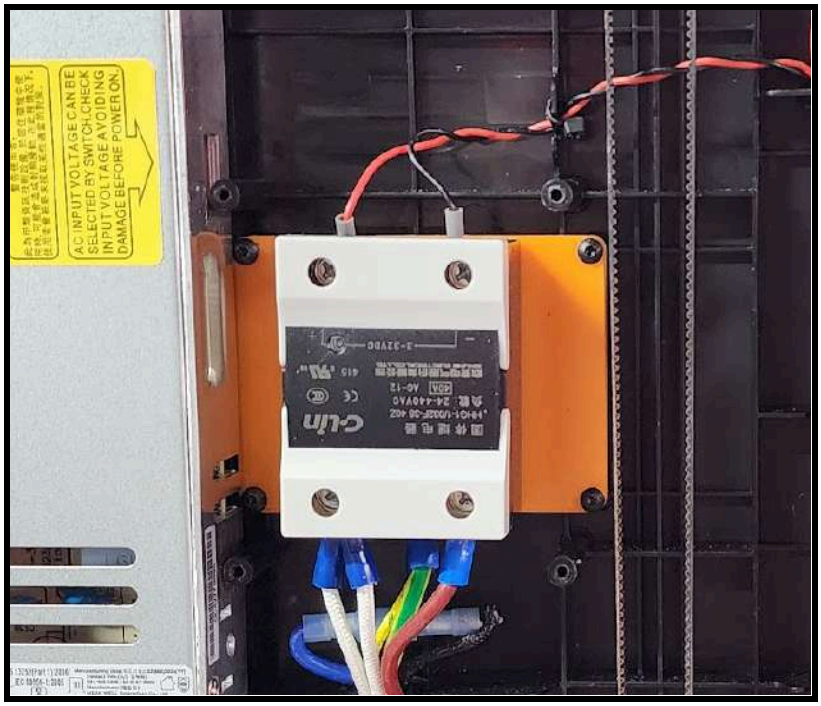
Solid state relay (110v)

Summary: The solid state relay uses a “trigger” wire from the M8P to tell it when to activate. When activated, it will “bridge” the two high voltage connections on the other side of it (the 110v main, and your high voltage heater bed wires)

1. 3rd party solid state relay
    a. You only need 2 wires between the SSR and **M8P**. Measure and cut 30cm of **red** and **black** 20-gauge wire. These are not carrying a load so do not need a higher gauge.
    b. Terminate one end of both wires with a fork terminal or ferrule and connect them to the solid state relay.
    c. Connect the **black** ground to the **-** connection and the **red** trigger wire to the **+** connection. (The solid state relay only has these marked on one side, the low-voltage or “trigger” side)
    d. Terminate the other end to a 3-pin JST and connect to the **RGB** port on the **M8P**.
        i. With the notches facing upwards, the wire order will be: **Ground**, **Control** (red wire in this case), nothing. Leave the 3rd pin empty for a 3rd party solid state relay.



    e. On the side that has no + or - markings (the high voltage side), connect your heater bed high voltage wire(s) to a SINGLE terminal.
        i. For 110v, you should have 2 high voltage wires, for example with Keenovo they are both white. Connect both to the same terminal.
        ii. For 220v, you will only have 1 high voltage wire, connect this wire to the terminal on the SSR
    f. Connect your 110v main power connection to the other terminal
    g. [Optional] Ground any loose ground wires to the bolt holding the SSR to the printer.



2. Stock solid state relay
    a. You only need 3 wires from this cable; you do not use the wire with the white label around it.
    b. Looking at the side connected to the stock solid state relay, the wire order is: Signal (white label), **Control**, **Ground**, **5v**
    c. Trim, strip, and terminate these to a 3-pin JST.
        i. With the notches facing upwards, from left to right, the wire order will be: **Ground**, **Control**, **5v**
    d. Connect the 3-pin JST to the **RGB** port on the **M8P**.
    e. Connect the other wires as they originally were (if you removed them) to the stock SSR (positive, ground, neutral, and bed wires)

Chamber thermistor

1. Measure and trim 65mm from the wire labeled **260c** (including the connector)
    a. Re-terminate it to a 2-pin JST (polarity does not matter) and connect it to **TH0** on the **M8P**.

Side fan

1. Just trim the connector from the wires labeled **FAN**. Strip and re-terminate it to a 2-pin JST and connect it to **FAN0** on the **M8P**.
    c. With the notches on the connector facing to the right, the top wire is **+V** and the bottom is **Ground**. On the original wire, the one with white marks on it is the **Ground**.

Exhaust fan

1. Don't trim anything, just run this up the left side of the **M8P** and connect it to **FAN1** on the **M8P**.

MCU fan

1. Throw away the stock fan. It sucks.

2. Trim the cable on your 120mm fan down to about 150mm and terminate the 4 wires to a 4-pin JST.
    a. With the notches on the connector facing to the right, the wires are **Ground**, **+V**, **tach**, **Control** from the bottom up.
    b. Match up the wires on your fan starting with the **Ground**, moving up.
    c. Strip and terminate the wires to the JST connector
    d. If you have a 2 or 3-pin fan, then simply wire up the pins you have.
3. Alternatively, trim the cable on the fan-side down to about 100mm, trim the cable on the connector-side down to 50mm, and strip/solder the wires together.

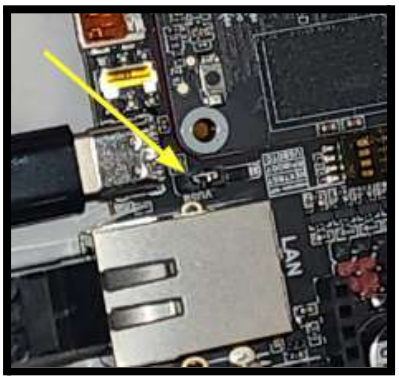
Filament runout sensor

1. Trim 180CM off the **FILAMENT** labeled wire and re-terminate to a 3-pin JST.
    a. With the notches on the connector facing to the right, the top wire is **Control**, middle wire is **Ground**, and the bottom wire is **5v**.
    b. When looking at the stock wire, with the white marks facing upwards, the wire order is: **5v**, **Ground**, **Control**
    c. Terminate as indicated to the 3-pin JST, route up the right side of the **M8P** and connect it to **M6-STOP** on the **M8P**

Final Steps

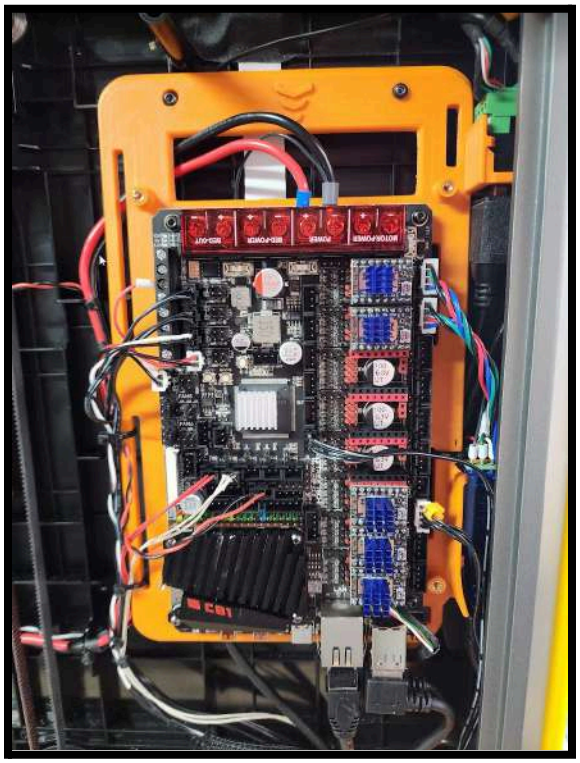
Connect Everything

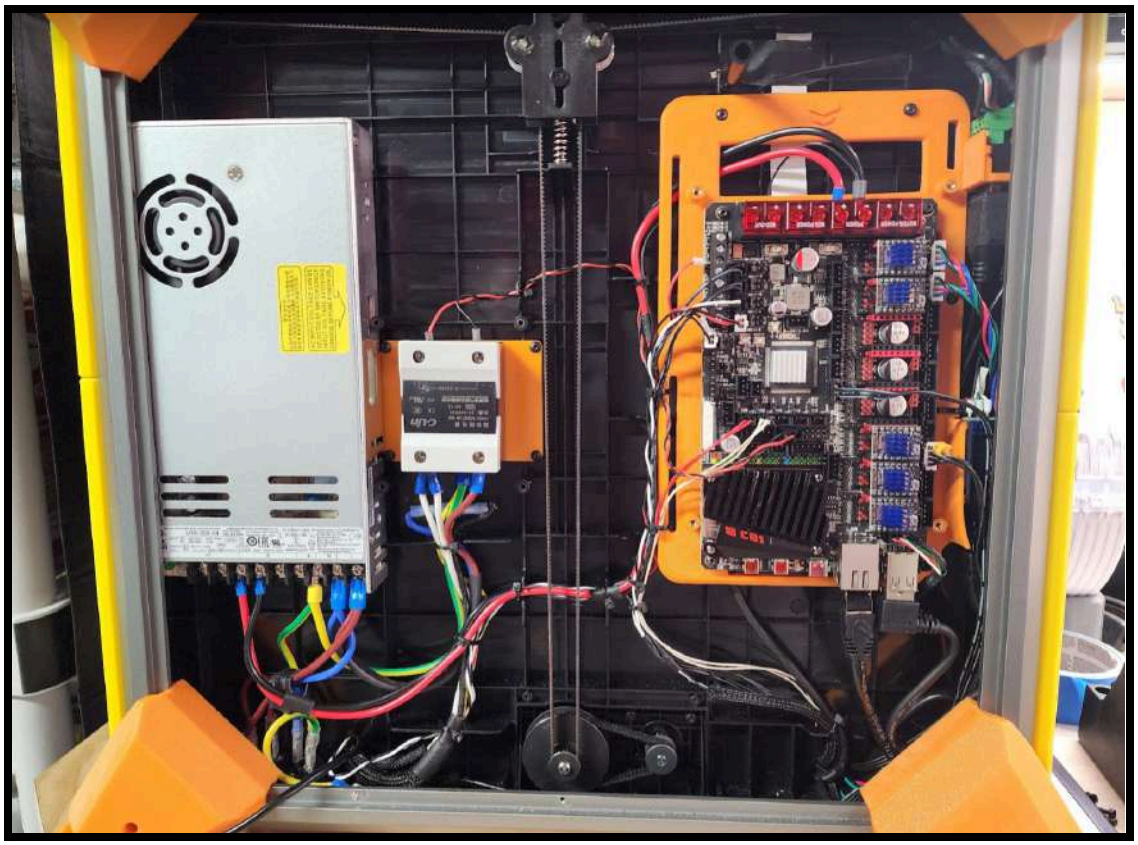
1. **THIS STEP IS IMPORTANT – IF YOU HAVE NOT YET, REMOVE THE USB JUMPER YOU ADDED NEAR THE USB PORTS DURING TESTING/FLASHING. IF YOU DO NOT YOU WILL DESTROY THE USB HUB ON THE M8P WHEN YOU CONNECT 24v POWER**



2. Connect the 16-gauge **red** and **black** power wires to the **M8P's POWER** input at the top. Connect the other end to the **V+** and **V-** on the power supply.
3. Connect your components (fans, thermistors, filament sensor, and SSR) to the indicated connectors from the picture above
4. Connect the **red** power wire for the tool head MCU RS232 connector to **VBB** on **HE1**
5. Connect the wireless antenna to the CB2 directly

6. Carefully route all of your wires using the remaining wire anchors on the bottom of the printer in addition to the locations on the side of the board mount. Verify nothing is touching any belts or anything else. Zip tie all your wires in place.
7. Install the 120mm fan on the shroud using 4 5mm M3 screws
8. Connect the 4 pin MCU fan to **FAN5**.
9. Install the shroud onto the board mount using 4 35mm M3 screws.





Power On The Upgraded Printer

1. Connect the power supply cable
2. Connect the ethernet network cable (if needed)
3. Power on the unit.
4. Double-check your sanity
5. Verify the activity lights blink after a few seconds.
    a. If not, remove everything connected and plug everything in one at a time testing powering on the unit in-between. Troubleshoot problem components by verifying polarity and connectivity. This may require additional assistance in the chat, please use the Devil Design Discord Support Group as needed - everyone is very helpful
    b. I ran into an issue with the device not booting when a DSI screen was plugged in, after troubleshooting and not finding a solution I went with an HDMI7 display instead. Your mileage may vary.
6. Test accessing the printer via a web browser using it's IP address.
7. Review the status of everything. Update any modules in the Update Manager that are dirty/corrupt or need to be updated. With the included .cfg files, as long as everything is connected as indicated you should be able to begin testing fans/heaters to verify they function. You may need to adjust device IDs to get your MCU's to show online. Adjust pin configuration as needed.

Flash the Toolhead MCU

1. Download MobaXTerm to connect via SSH with file transfer capabilities
2. Download the Creality K1 MCU flasher utility and copy it to your **M8P** under /home/biqu/
    a. If you get a permission denied error copying this file, trying saving the .py script to another folder and run it from there instead
3. Download Pellcorp's nozzle MCU firmware, rename the file to nozzle.bin and copy it to your printer to the same folder as above

Method A (*Confirmed working on both Stock board and M8P*)

1. Update your Nozzle MCU configuration in your printer.cfg file to reflect the correct by-path address (I could not get by-id to work)
    a. Type ***ls /dev/serial/by-path/**** to see a list of devices to try

```
[mcu Nozzle_MCU]
#serial: /dev/serial/by-id/usb-1a86_USB_Serial-if00-port0
serial: /dev/serial/by-path/platform-xhci-hcd.4.auto-usb-0:1.2.4:1.0-port0
```

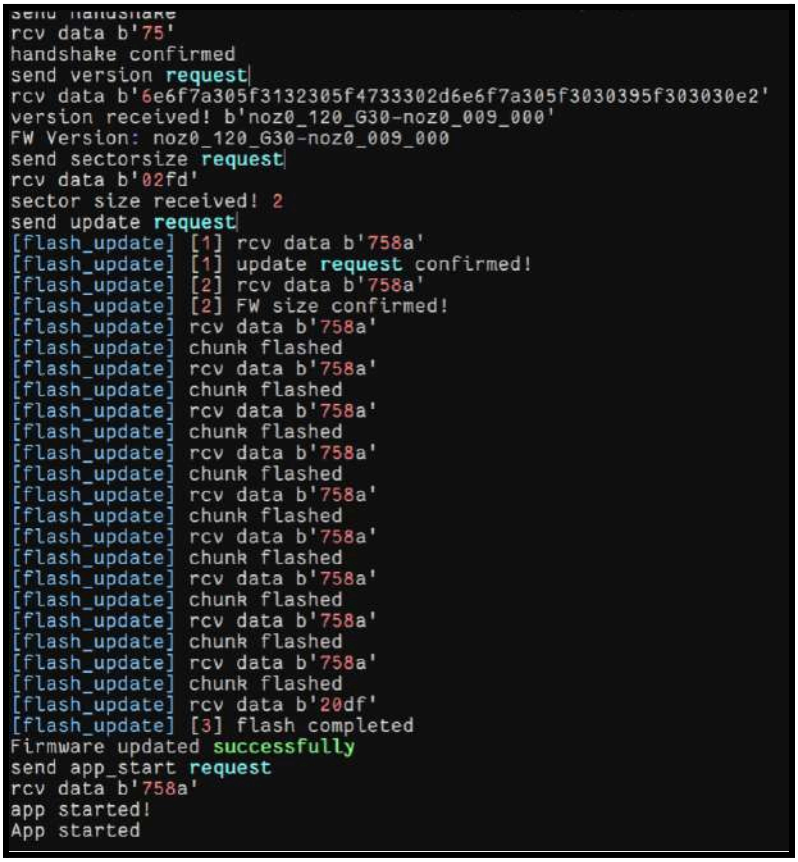
2. Stop the Klipper service in the web GUI by clicking in the top right and hitting the stop button

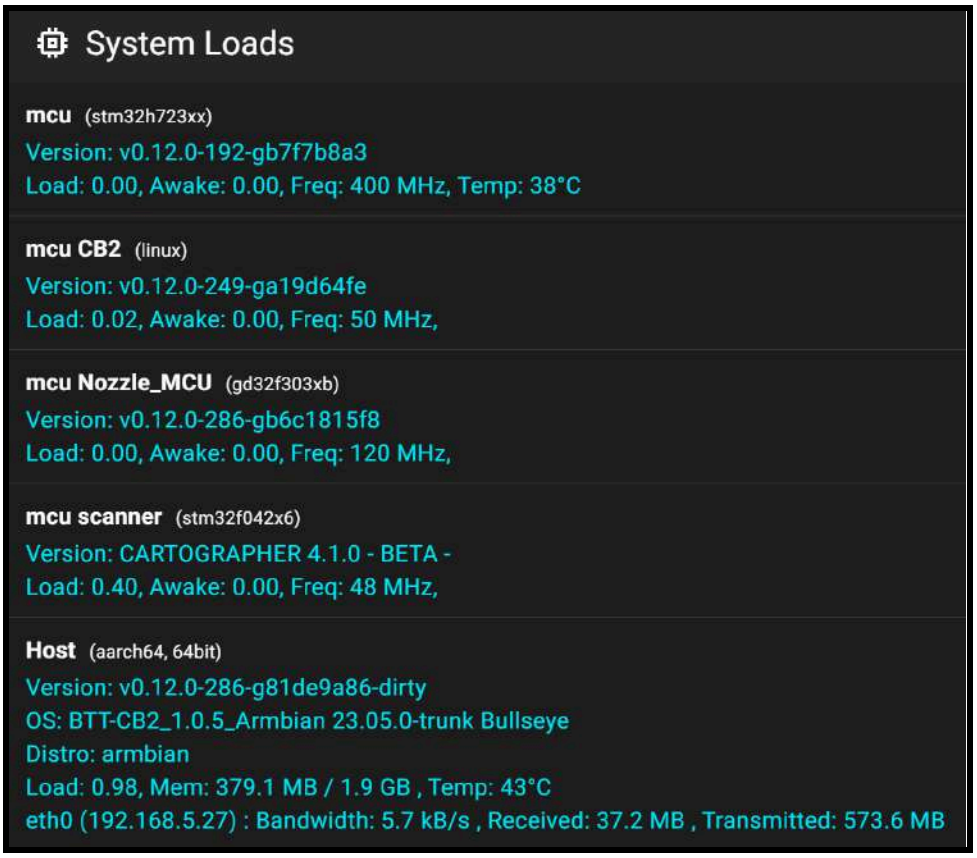
3. Connect via SSH to the Printer and type out the following command but do **NOT** hit enter yet

/home/biqu/klippy-env/bin/python3 /home/biqu/mcu_util.py -v -c -i /dev/ttyUSB0 -g -u -f /home/biqu/nozzle.bin

a. Replace the path if you saved it to a different location
b. Replace the filename nozzle.bin with the correct one if you did not rename your file.
c. The “Device ID” in this command is /dev/ttyUSB0
    i. If this does not work, type ***ls /dev/ttyUSB**** and try one of the options listed. If none are, your printer does not see the toolhead PCB.
    ii. If you are flashing from the **K1** stock board, device ID is /dev/ttyS1
d. If you need to flash back to stock, here is a link to Pellcorp’s stock nozzle MCU firmware, download the noz0 file and download the stock bootloader from here. Follow the instructions to flash the nozzle MCU.

4. Power cycle the toolhead PCB while the printer is still powered on by disconnecting the red power wire connected to the HE01 VBB port (while leaving the printer on)
    a. You have 15 to seconds to hit enter on the command you typed from step 2 (the toolhead PCB takes about 15 seconds to fully boot up, once booted it will not accept a flash)
    b. If it did not work, go back to step 3 and try again.
5. The following screen indicates a successful flash
6. Once verified the flash was successful, restart your printer and then don’t forget to start the Klipper service again by hitting the Play button in the top right corner





Method B (Still testing, not confirmed)

1. Connect via SSH to the Printer and type out the following command

Sudo nano /etc/systemd/system/flash-toolhead-script.service

2. Paste the following code into the text editor that opens

[Unit]

Description=Run Python Script on Early Startup

Before=klipper.service

After=Network.service

[Service]

ExecStart=/home/biqu/klippy-env/bin/python3 /home/biqu/mcu_util.py -c -i /dev/ttyUSB0 -g -u -f /home/biqu/nozzle.bin

WorkingDirectory=/home/biqu

StandardOutput=inherit

StandardError=inherit

[Install]

WantedBy=default.target

3. Press CTRL+X, then Y, then hit enter to save
4. Enter the following commands via console:

sudo systemctl enable flash-toolhead-script.service

sudo systemctl start flash-toolhead-script.service

5. Verify after starting it that no errors were produced. If there is an error, run the following command to determine the problem and adjust whatever is needed (usually the path or filename is incorrect)

sudo systemctl status startup-script.service

6. If needed, replace the /dev/ttyUSB0
    a. On the **M8P**, /dev/ttyUSB0 or /dev/ttyUSB1 is usually the device ID
    b. If you are flashing from the **K1** stock board, device ID is /dev/ttyS1
        i. If you need to flash back to stock, here is a link to Pellcorp's stock nozzle MCU firmware, download the noz0 file and download the stock bootloader from here. Follow the instructions to flash the nozzle MCU.
7. Once verified your script works, reboot the device
8. Verify your toolhead MCU firmware updated to the version you just applied
9. Disable the startup script now that the toolhead is updated:

sudo systemctl disable flash-toolhead-script.service

1. Reboot the device one more time

Test Everything And Print

1. Make sure that everything works as expected:
2. Test that fans spin in the correct direction, at the requested power
    a. Two of my fans polarity was reversed despite following the guideline of the **Grounds** being wires with white markings on them.

SET_PIN PIN=Bed_Fans VALUE=255

    a. part_fan, Side_Fan, Chamber_Fan, MCU_Fan, Bento_Air_Scrubbers, Bed_Fans, and LED_Lights are all options for the PIN, with the range going from 0-255 for off to maximum power.)
    b. If a fan does not work, check its continuity using a voltmeter
    c. If the fan operates backwards, re-pin it and reverse the polarity
3. Test that steppers move in the correct direction when both issuing direct move commands (G1) or homing.

G1 X-149 Y149 Z5 F3000
#Centers printhead

G1 X10 Y10 Z5 F3000
#*M*oves print head to corner

    a. If you can issue movement commands successfully but homing does not move as expected, your DIAG pin is likely misconfigured for that stepper or you forgot to bridge the diag pins for that stepper.
4. Verify that accurate temperatures are displayed for all thermistors
5. Run hot end and heater bed PID tunes

PID_CALIBRATE HEATER=extruder TARGET=200
PID_CALIBRATE HEATER=heater_bed TARGET=100

6. Test heater elements by setting the hot end to 150 and the bed to 60

SET_HEATER_TEMPERATURE HEATER=extruder TARGET=100
SET_HEATER_TEMPERATURE HEATER=heater_bed TARGET=60
#Once verified, set both of those targets back to 0.

7. Test the extruder by heating the hot end up to the desired temp (for example, 220 for ABS) and test extruding filament

SET_HEATER_TEMPERATURE HEATER=extruder TARGET=220
#Wait for temp to be reached
G1 X-149 Y149 Z20
#Centers printhead
G1 E10 F100
SET_HEATER_TEMPERATURE HEATER=extruder TARGET=0

8. Endstops should show X/Y as open and Z as triggered until the Carto is configured. Z will show open as well if PRtouch is still installed.

QUERY_ENDSTOPS

9. If you do not have Carto, verify home X/Y/Z works and perform a full bed mesh to verify PRTouch is working.
10. If you do have Carto, home X/Y then issue the following commands via console (in the web GUI is fine). Without an existing mesh it will not proceed with Z homing and will likely show the endstop as “triggered” until it has a configuration.

G1 X149 Y149 F6000

a. For Carto tap only users

Probe_calibrate

b. Then, for all Carto users

Carto_calibrate

11. Generate a fresh bed mesh.
12. Adjust the printer and repeat as necessary until you are happy with your tramming and mesh variance
13. Run input shaping and select your best shaper options
14. Send a test print!

Optional Installs

Upgrade The Power Supply (for triple Z)

WARNING: Disconnect your printer from AC power before proceeding. Death or serious injury may result from attempting to work on your printer while it is plugged into AC power.

1. Purchase a Meanwell LRS-350
2. Print the K1 Max Meanwell LRS-350 bracket
3. Print the Meanwell LRS-350 terminal cover
4. Disconnect all wires from the power supply using a screwdriver.
5. Remove the 2 screws holding the power supply to the bottom.
6. Install the bracket onto the power supply using short M4 screws. Slide the PSU into the cavity on the left side of the frame and use 2x 25mm M3 screws to attach the bracket to the stock Solid State Relay cover mount locations.
7. Connect up power/neutral/ground and the 24v/ground wires to the indicated terminals in the pictures below.
8. Connect any loose ground wires to either the chassis or the solid state relay case.
    a. On the stock SSR, connect to the GND location
    b. On a 3rd party SSR the ground is part of the mounting plate so secure your grounds to the center screw holding the SSR onto the printer.
9. Secure any wires as needed with zip ties.

Upgrade The Filament Sensor

1. Purchase a BTT SFS 2.0
2. Print the corner riser bracket for the SFS 2.0 (requires this style riser installed)
3. Remove screws holding stock filament sensor and uninstall sensor + wire.
4. Run included wire to Manta M8P and connect (connectorA) to M6-STOP and (connectorB) to M5-STOP.
5. In the included printer.cfg file, comment out the filament sensor config and uncomment out the two configurations directly below it.

Upgrade The Bed and Bed Heater

WARNING: Disconnect your printer from AC power before proceeding. Death or serious injury may result from attempting to work on your printer while it is plugged into AC power.

1. Print the 3rd party solid state relay mount for the K1 Max
2. Purchase a Keenovo 750w or similar bed heater
    a. Make sure to include a 40VAC C-Lin (or comparable) solid state relay. The stock solid state relay does not reliably operate this bed heater; it will intermittently not work.
3. Purchase the ultra flat Mandala 6mm bed or similar
4. Follow Keenovo's Install instructions and install the Keenovo bed heater carefully centering it while leaving a small area exposed at the rear.
5. Remove the 2 screws from the top rear of the stock heated bed and install them on the matching available holes on the new bed
6. Using a small screw, connect the available ground wire to the heated bed underneath, using the available screw hole to ground it to the heated bed itself.

7. Remove the stock bed and stock solid state relay.
    a. This includes removing the stock bed thermistor/wires from the printer entirely.
8. Install the printed SSR mount and 3rd party solid state relay in the location the original was. The SSR requires M4 or M5 screws to mount it properly. The C-Lin SSR had to be installed upside-down for the terminals to be oriented correctly with the trigger wires on top.
9. Terminate the neutral wires from the power supply and Keenovo using blade connectors, or a WAGO connector and connect them together. You can also connect the Keenovo neutral wire to the power supply directly if you feel like cutting up the stock wire harness to clean up the extra neutral wire hanging there.
10. Connect any loose ground wires to the solid state relay case. On a 3rd party SSR the ground is part of the mounting plate so secure your grounds to the center screw holding the SSR onto the printer.
11. Re-terminate and connect the thermistor from the Keenovo to the same location the stock bed thermistor will be installed on the Manta M8P (THB)
12. Connect the two white wires from the Keenovo to the lower left terminal on the 3rd party SSR
13. Connect the primary power wire from the power supply to the lower right terminal
14. Create and terminate a 2 pin wire that will run from the top terminals on the 3rd party SSR to the Manta M8P.
15. Connect the control wire from the M8P to the upper left terminal
16. Connect the ground wire from the M8P to the upper right terminal
17. In the included printer.cfg file, under the bed_heater section, comment out the thermistor type and uncomment the line directly below it.

Configuration Files

These .cfg files are a work-in-progress. These configuration files are 100% functional when combined with the above installation. The below configuration also includes sections for Carto Tap, which at the time of writing this document is still in beta. There are numerous additional fans added that you may need to comment out.

Configuration files download links: printer.cfg, gcode_macro.cfg
Updated 8/21/2024

Printer.cfg

```
# K1 MAX
# Configuration creation date: 7/8/2024

#===MODIFICATIONS===
# Relevant to below configurations, if your printer is different please adjust as necessary

# Manta M8P with CB2, using 2209 steppers at 24v
# BootyJones linear rail gantry with 2504MAC steppers
# Cartographer
# Bondtech extruder
```

```
# Keenovo bed heater with upgraded SSR (Stock K1 Max SSR does not reliably
operate the Keenovo heater)
# Bed fans
# Bento mini air purifiers

#===INCLUDES===

# [include fluidd.cfg]
[include mainsail.cfg]
[include sensorless.cfg]
[include gcode_macro.cfg]
[include timelapse.cfg]
[include KAMP_Settings.cfg]
[include moonraker_obico_macros.cfg]
[include bedfans.cfg] # Comment this out if you do not have bed fans installed as
it overwrites some default commands
# [include K-ShakeTune/*.cfg]
# [include aux_fan.cfg]
# [include shell_command.cfg]
# [include printer_params.cfg]

[exclude_object]

#===MCUs===

[mcu]
serial: /dev/serial/by-id/usb-Klipper_stm32h723xx_05001E001851313434373135-if00

[mcu Nozzle_MCU]
serial: /dev/serial/by-path/platform-xhci-hcd.4.auto-usb-0:1.2.4:1.0-port0
baud: 230400
restart_method: command

[mcu CB2]
serial: /tmp/klipper_host_mcu

[temperature_sensor Manta M8P]
sensor_type: temperature_mcu
min_temp: 0
max_temp: 100

[virtual_sdcard]
path: /home/biqu/printer_data/gcodes
```

```
#===SAFETY SETTINGS===

[verify_heater extruder]
[verify_heater heater_bed]
check_gain_time: 120
heating_gain: 1.0
hysteresis: 10

[idle_timeout]
gcode =
    M104 S0 # turn off hotend
    M140 S0 # turn off bed heater
    M107 P0 # turn off part fan
    M107 P2 # turn off side fan
    M107 P3 # turn off chamber fan
    {action_respond_info("Idle timeout reached. Heaters and fans powered down.")}
# Let the user know what is happening
    M84 # unlock steppers
timeout = 1200

[gcode_arcs]
resolution: 0.3

#===CARTOGRAPHER===

[scanner]                   # uncomment if using cartographer
serial: /dev/serial/by-id/usb-Cartographer_614e_380021001443565039363120-if00
speed: 40.0
lift_speed: 5.0
backlash_comp: 0.5
x_offset: 0 # Must match your probes position
y_offset: 15 # Must match your probes position
trigger_distance: 2.0
trigger_dive_threshold: 1.5
trigger_hysteresis: 0.006
cal_nozzle_z: 0.1
cal_floor: 0.1
cal_ceil:5.0
cal_speed: 1.0
cal_move_speed: 10.
default_model_name: default
mesh_main_direction: x
```

```
mesh_cluster_size: 1
mesh_runs: 2
calibration_method: touch
z_offset: 0
probe_speed: 2.0
touch_location: 149,149 # set to center of bed
sensor: cartographer

#===MOTORS===

[stepper_x]
step_pin: PE6
dir_pin: PE5
enable_pin: !PC14
microsteps: 32
rotation_distance: 40
endstop_pin: tmc2209_stepper_x:virtual_endstop
position_endstop: 304
position_min: 0.5
position_max: 304
homing_speed: 80
homing_retract_dist: 0
full_steps_per_rotation: 400

[tmc2209 stepper_x]
uart_pin: PC13
interpolate: False
run_current: 1.4
#hold_current:1.0
#stealthchop_threshold: 0
diag_pin: PF4
sense_resistor: 0.100
driver_SGTHRS: 145

[stepper_y]
step_pin: PE2
dir_pin: PE1
enable_pin: !PE4
microsteps: 32
rotation_distance: 40
endstop_pin: tmc2209_stepper_y:virtual_endstop
position_endstop: -2
position_min: -2
```

```
position_max: 284
homing_speed: 80
homing_retract_dist:0
full_steps_per_rotation: 400

[tmc2209 stepper_y]
uart_pin: PE3
interpolate: False
run_current: 1.4
#hold_current:1.0
#stealthchop_threshold: 0
diag_pin: PF3
sense_resistor: 0.100
driver_SGTHRS: 147

[stepper_z]
step_pin: PG9
dir_pin: PD7
enable_pin: !PG11
microsteps: 16
rotation_distance:8
gear_ratio: 64:20
endstop_pin: probe:z_virtual_endstop # use cartographer as virtual endstop
homing_retract_dist: 0 # cartographer needs this to be set to 0
#position_endstop: 0
position_max: 305
position_min: -10
homing_speed: 10
second_homing_speed: 1
#homing_retract_dist: 2.0

[tmc2209 stepper_z]
uart_pin: PG10
run_current: 0.8
diag_pin: PC15
stealthchop_threshold: 0
driver_SGTHRS: 0
sense_resistor: 0.100

[tmc2209 extruder]
uart_pin: Nozzle_MCU:PB11
tx_pin: Nozzle_MCU:PB10
uart_address: 3
```

```
# run_current: 0.5 #stock extruder
run_current: 0.7 #bondtech extruder
sense_resistor: 0.150
stealthchop_threshold: 0
# driver_IHOLDDELAY: 8
# driver_TPOWERDOWN: 20
# driver_TBL: 2
# driver_TOFF: 3
# driver_HEND: 0
# driver_HSTRT: 5

# ===HEATERS===

[extruder]
max_extrude_only_distance: 1000.0
max_extrude_cross_section: 80
step_pin: Nozzle_MCU:PB1
dir_pin: Nozzle_MCU:PB0
enable_pin: !Nozzle_MCU:PB2
microsteps: 16
#rotation_distance: 6.9 #stock extruder
rotation_distance: 5.7 #bondtech extruder
nozzle_diameter: 0.400
filament_diameter: 1.750
heater_pin: Nozzle_MCU:PB7
sensor_type: EPCOS 100K B57560G104F
sensor_pin: Nozzle_MCU:PA0
pressure_advance: 0.05
pressure_advance_smooth_time: 0.040
#control: pid #Uncomment this line and the 3 below on first setup
#pid_Kp: 25.013
#pid_Ki: 2.566
#pid_Kd: 60.966
min_temp: 0
max_temp: 320

[heater_bed]
heater_pin: PD15
#sensor_type: EPCOS 100K B57560G104F #stock bed thermistor
sensor_type: Generic 3950 #Keenovo bed thermistor
sensor_pin: PB1
#control: watermark #Uncomment this line and the 4 below on first setup
#control: pid
```

```
#pid_kp: 27
#pid_ki: 0.08
#pid_kd: 0
min_temp: 0
max_temp: 120

#===FAN CONFIGS===

[duplicate_pin_override]
pins: PB0, PF9, PA2, PC5, PF8

#[multi_pin heater_fans] # The MCU and hotend fan have been split and operate
independently now
#pins:Nozzle_MCU:PB5,PB2

[heater_fan hotend_fan]
#pin: multi_pin:heater_fans
pin: Nozzle_MCU:PB5
tachometer_pin: ^Nozzle_MCU:PB4
heater: extruder
heater_temp: 40

[temperature_sensor chamber_temp]
sensor_type: EPCOS 100K B57560G104F
sensor_pin: PB0
min_temp: 0
max_temp: 125

[temperature_sensor power_supply_temp] # Comment out this section if not needed
sensor_type: Generic 3950
sensor_pin: PC5
min_temp: 0
max_temp: 125

[static_digital_output my_fan_output_pins]
pins: Nozzle_MCU: PB6

[fan_generic Part_Fan]
pin: !Nozzle_MCU: PB8
tachometer_pin: ^Nozzle_MCU: PB3
max_power: 1.0
shutdown_speed: 0.0
kick_start_time: 0.300
```

```
cycle_time: 0.010
hardware_pwm: False

[fan_generic Side_Fan]
pin: PF7
max_power: 1.0
shutdown_speed: 0.0
kick_start_time: 0.100
cycle_time: 0.010
hardware_pwm: False

[fan_generic Chamber_Fan]
pin: PF9
max_power: 1.0
shutdown_speed: 0.0
kick_start_time: 0.100
cycle_time: 0.010
hardware_pwm: False

[fan_generic Power_Supply_Fan] # Comment out this section if not needed
pin: PA2
max_power: 1.0
shutdown_speed: 0.0
kick_start_time: 0.100
cycle_time: 0.010
hardware_pwm: False

[temperature_fan Chamber_Fan]
# Temperature based control over chamber fan, use the following gcode in your
machine start g-code
# SET_TEMPERATURE_FAN_TARGET TEMPERATURE_FAN=Chamber_Fan TARGET=65
pin: PF9
cycle_time: 0.0100
hardware_pwm: false
max_power: 1
shutdown_speed: 0
sensor_type: EPCOS 100K B57560G104F
sensor_pin: PB0
min_temp: 0
max_temp: 100
control: watermark
max_delta: 2
target_temp: 50
```

```
max_speed: 1.0
min_speed: 0.0

[temperature_fan Power_Supply_Fan] # Comment out this section if not needed
pin: PA2
tachometer_pin: PC1
cycle_time: 0.0100
hardware_pwm: false
max_power: 1
shutdown_speed: 0
sensor_type: Generic 3950
sensor_pin: PC5
min_temp: 0
max_temp: 70
control: pid
pid_Kp: 4.0
pid_Ki: 1
pid_Kd: 0.2
pid_deriv_time: 2.0
target_temp: 40.0
max_speed: 1.0
min_speed: 0.0

[temperature_fan MCU_Fan]
pin: PA6
tachometer_pin: PC2
cycle_time: 0.01
hardware_pwm: false
max_power: 1
shutdown_speed: 0
sensor_type: temperature_host
min_temp: 0
max_temp: 110
control: pid
pid_Kp: 4.0
pid_Ki: 1
pid_Kd: 0.2
pid_deriv_time: 2.0
target_temp: 45.0
max_speed: 1.0
min_speed: 0.0

[temperature_fan Air_Purifier] # Comment out this section if not needed
```

```
pin: PF8
cycle_time: 0.0100
hardware_pwm: false
max_power: 1
shutdown_speed: 0
sensor_type: EPCOS 100K B57560G104F
sensor_pin: PB0
min_temp: 0
max_temp: 70
control: watermark
max_delta: 2
target_temp: 50
max_speed: 1.0
min_speed: 0.0

[fan_generic Air_Purifier] # Comment out this section if not needed
pin: PF8
max_power: 1.0
shutdown_speed: 0.0
kick_start_time: 0.100
cycle_time: 0.010
hardware_pwm: False

[fan_generic Bed_Fans] # Comment out this section if not needed
pin: PF6
max_power: 1.0
shutdown_speed: 0.0
kick_start_time: 0.100
cycle_time: 0.010
hardware_pwm: False

[output_pin LED_Lights]
pin: PA4
pwm: true
hardware_pwm: false
scale: 255
shutdown_value: 0.0
value: 255
cycle_time: 0.001

#===BED MESH===

[bed_mesh]
```

```
speed: 350
mesh_min: 20,27             # x / y offsets for carto.
mesh_max: 280,270           # added a little space from the back of the bed to
prevent scanning screws or crashing into the motor mounts
probe_count: 40,40
#fade_start: 5.0
#fade_end: 50.0
algorithm: bicubic
bicubic_tension: 0.1        # required for above 5x5 meshing

#===PRINTER SETTINGS===

[adxl345]
cs_pin: scanner:PA3 #If not using Cartographer, comment out this line and
uncomment the line below
#cs_pin: Nozzle_MCU:PA4
spi_bus: spi1

[resonance_tester]
accel_chip: adxl345
probe_points:
   150,150,10

[input_shaper]
shaper_freq_x: 48.6 # center frequency for the X axis filter
shaper_type_x: zv # filter type for the X axis
shaper_freq_y: 44.8 # center frequency for the Y axis filter
shaper_type_y: mzv # filter type for the Y axis
damping_ratio_x: 0.091 # damping ratio for the X axis
damping_ratio_y: 0.033 # damping ratio for the Y axis

[display_status]

#[filament_switch_sensor filament_sensor] # Stock filament sensor
#pause_on_runout: true
#switch_pin: !PC15

[filament_switch_sensor runout_sensor] # BTT SFS2.0 only
switch_pin: PC15
pause_on_runout: False
runout_gcode:
  PAUSE # [pause_resume] is required in printer.cfg
  M117 Filament runout detected
```

```
  {action_respond_info("Filament runout detected")}
insert_gcode:
  M117 Filament inserted
  {action_respond_info("Filament inserted")}

[filament_motion_sensor jam_sensor] # BTT SFS2.0 only
switch_pin: !PF0
detection_length: 2.88
extruder: extruder
pause_on_runout: False
runout_gcode:
  PAUSE # [pause_resume] is required in printer.cfg
  M117 Extrusion issues detected
  {action_respond_info("Extrusion issues detected")}
insert_gcode:
  M117 Extrusion issues resolved
  {action_respond_info("Extrusion issues resolved")}

[printer]
kinematics: corexy
max_velocity: 800
max_accel: 20000
minimum_cruise_ratio: 0.1
max_z_velocity: 20
square_corner_velocity: 5.0
max_z_accel: 300

[pause_resume]

#[axis_twist_compensation]
#speed: 300
#horizontal_move_z: 5
#calibrate_start_x: 20
#calibrate_end_x: 200
#calibrate_y: 112.5

[firmware_retraction]
retract_length: 0.4
retract_speed: 60
unretract_extra_length: 0
unretract_speed: 60
```

Gcode_macro.cfg

```
#DEFINED VARIABLES

[gcode_macro PRINTER_PARAM]
variable_z_safe_pause: 0.0
variable_z_safe_g28: 3.0
variable_max_x_position: 300.0
variable_max_y_position: 300.0
variable_max_z_position: 300.0
variable_fans: 3
variable_auto_g29: 0
variable_fan0_min: 25
variable_fan1_min: 50
variable_fan2_min: 180
variable_fan2_speed: 0
variable_hotend_temp: 0
variable_e_min_current: 0.27
gcode:

[gcode_macro _CLIENT_VARIABLE]
variable_use_custom_pos  : True
variable_custom_park_x   : 285
variable_custom_park_y   : 275
variable_custom_park_dz   : 50.0
variable_park_at_cancel   : True
gcode:

#GENERAL MACROS (START,END,CANCEL,LOAD,UNLOAD,PAUSE,RESUME)
[gcode_macro QUICK_CLEAN_NOZZLE]
gcode:
    # Move nozzle to cleaning position
    m104 S160
    G1 X1 Y1 Z10 F3000  ; Move to cleaning position
    M400  ; Wait for moves to finish

    # Clean nozzle
    G1 Z0 F300  ; Lower nozzle to 0mm above the bed
    M106 P0 S1  ; Set nozzle fan to max
    G1 X75 Y1 F300  ; Wipe nozzle on the bed
    G1 Z20 F3000  ; Raise nozzle
```

```
[gcode_macro CLEAN_NOZZLE]

variable_start_x: 130
variable_start_y: 226
variable_start_z: 3.8
variable_wipe_dist: 40
variable_wipe_qty: 60
variable_wipe_spd: 250
variable_raise_distance: 15

gcode:
 {% if "xyz" not in printer.toolhead.homed_axes %}
   G28
 {% endif %}

 ## Reset max accel and velocity in case it's been overwritten
 SET_VELOCITY_LIMIT VELOCITY={printer.configfile.settings.printer.max_velocity}
ACCEL={printer.configfile.settings.printer.max_accel}

 ## Move nozzle to start position
 G90
 G1 X{start_x} Y{start_y} F6000
 G1 Z{start_z} F1500



 ## Heat up the nozzle, cool, and turn on fans while wiping
 M109 S180 # 180c
 M109 S0
 M106 S1
 M106 P2 S1



 ## Wipe nozzle
 {% for wipes in range(1, (wipe_qty + 1)) %}
   G1 X{start_x + wipe_dist} F{wipe_spd * 60}
   G1 X{start_x} F{wipe_spd * 60}
 {% endfor %}

 ## Raise nozzle
 G1 Z{raise_distance}


[gcode_macro START_PRINT]
```

```
variable_bed_temp: 105
variable_extruder_temp: 245
gcode:
    {% set BED_TEMP = params.BED_TEMP|default(105)|float %}
    {% set EXTRUDER_TEMP = params.EXTRUDER_TEMP|default(245)|float %}
    m104 S{150}
# start heating nozzle dont wait
    M140 S{BED_TEMP}
# start heating bed to desired bed temps and wait
    G28
    BED_MESH_CLEAR
# clear current mesh
    G4 P500
# wait required to prevent MCU overload / inconsistant meshing
    SET_VELOCITY_LIMIT ACCEL=15000 ACCEL_TO_DECEL=10000 SQUARE_CORNER_VELOCITY=25
# drop accels to prevent hyper agressive meshing
    QUICK_CLEAN_NOZZLE
    G28
    M107 P0
# Set nozzle fan to off
    M140 S{BED_TEMP}
# start heating bed to desired bed temps and wait
    CARTOGRAPHER_TOUCH # CALIBRATE=0
    BED_MESH_CALIBRATE
# start bedmesh calibrate
    G4 P500
# wait required to prevent MCU overload / inconsistant mesh calculation
    SMART_PARK
# park the printhead near the print area
    SET_VELOCITY_LIMIT ACCEL=18000 ACCEL_TO_DECEL=10000 SQUARE_CORNER_VELOCITY=5
# up acceleration back to slightly below regular travel speeds
    M190 S{BED_TEMP}
# wait for bed temperature before next step
    M109 S{EXTRUDER_TEMP}
# wait for nozzle temperature before next step
    LINE_PURGE
# create purge line near the print area of the part
    {action_respond_info("3d model received. Printing initiated.")}
# happy printing~

[gcode_macro END_PRINT]
gcode:
```

```
  G91
# set to reletive positioning
  G1 E-5 F300
# Retract 5mm of filament at 300mm/min
  G1 Z15
# move z down 15mm to avoid contacting nozzle
  G90
# set to absolute positioning
  {% set POSITION_X = printer.configfile.settings['stepper_x'].position_max *0.99
%}  # Calculate 99% position of X
  {% set POSITION_Y = printer.configfile.settings['stepper_y'].position_max /2 %}
# calculate 50% of total Y
  G1 X{POSITION_X} Y{POSITION_Y} F6000
# move toolhead to the front of the K1 aux fan like stock
#  M220 S100
# set feedrate to 100
#  M204 S500
# set starting acceleration to 500
  TURN_OFF_HEATERS
# disable heaters
  M84
# power off motors
    {action_respond_info("Print complete, cooling that nozzle 150c.")}
# End Print Notification
  M106 P0 S1
# Set nozzle fan to max
  M106 P2 S1
# Set side fan to max
  M109 S150
# Cool nozzle to 150c to reduce oozing
  M107 P0
# turn toolhead fan off
  M107 P2
# turn sidefan off
  SET_FAN_SPEED FAN=Bed_Fans SPEED=0.0
  SET_TEMPERATURE_FAN_TARGET TEMPERATURE_FAN=Chamber_Fan TARGET=40
  SET_TEMPERATURE_FAN_TARGET TEMPERATURE_FAN=Air_Purifier TARGET=40
  SET_PAUSE_NEXT_LAYER ENABLE=0
  SET_PAUSE_AT_LAYER ENABLE=0 LAYER=0
  M84
# disable stepper motors
  M104 S0
# turn off hot end
```

```
  EXCLUDE_OBJECT_DEFINE RESET=1
# reset bed mesh boundaries


[gcode_macro CANCEL_PRINT]
description: Cancel the actual running print
rename_existing: CANCEL_PRINT_BASE
gcode:
  ##### get user parameters or use default #####
  {% set client = printer['gcode_macro _CLIENT_VARIABLE'] | default({}) %}
  {% set allow_park = client.park_at_cancel | default(false) | lower == 'true' %}
  {% set retract = client.cancel_retract | default(5.0) | abs %}
  ##### define park position #####
  # restore idle_timeout time if needed
  {% if printer['gcode_macro RESUME'].restore_idle_timeout > 0 %}
    SET_IDLE_TIMEOUT TIMEOUT={printer['gcode_macro RESUME'].restore_idle_timeout}
  {% endif %}
  _CLIENT_RETRACT LENGTH={retract}
  G91
  G1 Z15
# move z down 15mm to avoid contacting nozzle
  G90
# set to absolute positioning
  {% set POSITION_X = printer.configfile.settings['stepper_x'].position_max *0.99
%}  # Calculate 99% position of X
  {% set POSITION_Y = printer.configfile.settings['stepper_y'].position_max /2 %}
# calculate 50% of total Y
  G1 X{POSITION_X} Y{POSITION_Y} F6000
# move toolhead to the front of the K1 aux fan like stock
  TURN_OFF_HEATERS
  M106 S0
  M106 P0 S1
# Set nozzle fan to max
  M106 P2 S1
# Set side fan to max
  M109 S150
# Cool nozzle to 150c to reduce oozing
  M107 P0
# turn toolhead fan off
  M107 P2
# turn sidefan off
  SET_FAN_SPEED FAN=Bed_Fans SPEED=0.0
  SET_TEMPERATURE_FAN_TARGET TEMPERATURE_FAN=Chamber_Fan TARGET=40
```

```
  SET_TEMPERATURE_FAN_TARGET TEMPERATURE_FAN=Air_Purifier TARGET=40
  {client.user_cancel_macro | default("")}
  SET_GCODE_VARIABLE MACRO=RESUME VARIABLE=idle_state VALUE=False
  # clear pause_next_layer and pause_at_layer as preparation for next print
  SET_PAUSE_NEXT_LAYER ENABLE=0
  SET_PAUSE_AT_LAYER ENABLE=0 LAYER=0
  CANCEL_PRINT_BASE
  EXCLUDE_OBJECT_DEFINE RESET=1
# reset bed mesh boundaries

[gcode_macro LOAD_FILAMENT]
description: Load filament into the hotend
gcode:
  {% set EXTRUDE_LENGTH = params.EXTRUDE_LENGTH|default(50)|float %}
  {% set EXTRUDE_SPEED = params.EXTRUDE_SPEED|default(5)|float %}
  {% set RETRACT_LENGTH = params.RETRACT_LENGTH|default(2)|float %}
  {% set TEMPERATURE = params.TEMPERATURE|default(220)|float %}
  M117 Heating up to load filament
  M104 S{TEMPERATURE}
  M109 S{TEMPERATURE}
  M117 Loading filament
  G92 E0
  G1 E{EXTRUDE_LENGTH} F{EXTRUDE_SPEED * 60}
  G1 E-{RETRACT_LENGTH} F{EXTRUDE_SPEED * 60}
  G4 P5000  # Wait 5 seconds
  M104 S0 # turn off extruder heater
  M117 Filament load complete
  {action_respond_info("Mmm, delicious!")}

[gcode_macro UNLOAD_FILAMENT]
description: Unload filament from the hotend
gcode:
  {% set RETRACT_LENGTH = params.RETRACT_LENGTH|default(100)|float %}
  {% set RETRACT_SPEED = params.RETRACT_SPEED|default(5)|float %}
  {% set FAST_RETRACT_LENGTH = params.FAST_RETRACT_LENGTH|default(20)|float %}
  {% set FAST_RETRACT_SPEED = params.FAST_RETRACT_SPEED|default(40)|float %}
  {% set TEMPERATURE = params.TEMPERATURE|default(220)|float %}
  M117 Heating up to unload filament
  M104 S{TEMPERATURE}
  M109 S{TEMPERATURE}
  M117 Unloading filament
  G92 E0
  G1 E-{RETRACT_LENGTH} F{RETRACT_SPEED * 60}
```

```
  G1 E-{FAST_RETRACT_LENGTH} F{FAST_RETRACT_SPEED * 60}
  M104 S{160}
  M117 Filament unload complete
  {action_respond_info("That filament sucked anyway.")}

[gcode_macro PAUSE]
rename_existing: BASE_PAUSE
gcode:
    # Parameters
    {% set z = params.Z|default(10)|int %}
; z hop amount

    {% if printer['pause_resume'].is_paused|int == 0 %}
        SET_GCODE_VARIABLE MACRO=RESUME VARIABLE=zhop VALUE={z}
; set z hop variable for reference in resume macro
        SET_GCODE_VARIABLE MACRO=RESUME VARIABLE=etemp
VALUE={printer['extruder'].target}    ; set hotend temp variable for reference in
resume macro

        SET_FILAMENT_SENSOR SENSOR=runout_sensor ENABLE=0
; disable filament sensor
        SAVE_GCODE_STATE NAME=PAUSE
; save current print position for resume
        BASE_PAUSE
; pause print
        {% if (printer.gcode_move.position.z + z) <
printer.toolhead.axis_maximum.z %}       ; check that zhop doesn't exceed z max
            G91
; relative positioning
            G1 Z{z} F900
; raise Z up by z hop amount
        {% else %}
            { action_respond_info("Pause zhop exceeds maximum Z height.") }
; if z max is exceeded, show message and set zhop value for resume to 0
            SET_GCODE_VARIABLE MACRO=RESUME VARIABLE=zhop VALUE=0
        {% endif %}
        G90
; absolute positioning
        G1 X{printer.toolhead.axis_maximum.x/2}
Y{printer.toolhead.axis_minimum.y+5} F6000   ; park toolhead at front center
        SAVE_GCODE_STATE NAME=PAUSEPARK
; save parked position in case toolhead is moved during the pause (otherwise the
return zhop can error)
```

```
        M104 S0
; turn off hotend
        SET_IDLE_TIMEOUT TIMEOUT=43200
; set timeout to 12 hours
    {% endif %}

[gcode_macro RESUME]
rename_existing: BASE_RESUME
variable_zhop: 0
variable_etemp: 0
gcode:
    # Parameters
    {% set e = params.E|default(2.5)|int %}
; hotend prime amount (in mm)

    {% if printer['pause_resume'].is_paused|int == 1 %}
        SET_FILAMENT_SENSOR SENSOR=runout_sensor ENABLE=1
; enable filament sensor
        #INITIAL_RGB
; reset LCD color
        SET_IDLE_TIMEOUT
TIMEOUT={printer.configfile.settings.idle_timeout.timeout}  ; set timeout back to
configured value
        {% if etemp > 0 %}
            M109 S{etemp|int}
; wait for hotend to heat back up
        {% endif %}
        RESTORE_GCODE_STATE NAME=PAUSEPARK MOVE=1 MOVE_SPEED=100
; go back to parked position in case toolhead was moved during pause (otherwise
the return zhop can error)
        G91
; relative positioning
        M83
; relative extruder positioning
        {% if printer[printer.toolhead.extruder].temperature >=
printer.configfile.settings.extruder.min_extrude_temp %}
            G1 Z{zhop * -1} E{e} F900
; prime nozzle by E, lower Z back down
        {% else %}
            G1 Z{zhop * -1} F900
; lower Z back down without priming (just in case we are testing the macro with
cold hotend)
        {% endif %}
```

```
        RESTORE_GCODE_STATE NAME=PAUSE MOVE=1 MOVE_SPEED=60
; restore position
        BASE_RESUME
; resume print
    {% endif %}

#ADDITIONAL MACROS (M106, M107, etc)

[gcode_macro M106]
description: Set Fan Speed. P0 for part, P2 for auxiliary, P3 chamber
gcode:
  {% set fan_map = {0: "Part_Fan", 2: "Side_Fan", 3: "Chamber_Fan"} %}
  {% set fan_id = params.P|default(0)|int %}
  {% set fan = fan_map[fan_id] %}
  {% set speed_param = params.S|default(255)|int %}
  {% if speed_param > 0 %}
    {% set speed = (speed_param|float / 255) %}
  {% else %}
    {% set speed = 0 %}
  {% endif %}
  SET_FAN_SPEED FAN={fan} SPEED={speed}

[gcode_macro M107]
description: Set Fan Off. P0 for part, P2 for auxiliary, P3 chamber
gcode:
  {% set fan_map = {0: "Part_Fan", 2: "Side_Fan", 3: "Chamber_Fan"} %}
  {% set fan_id = params.P|default(0)|int %}
  {% set fan = fan_map[fan_id] %}
  SET_FAN_SPEED FAN={fan} SPEED=0

[gcode_macro M109]
description: Wait for hot end temp before proceeding (does not wait to stabilize)
rename_existing: M99109
gcode:
    #Parameters
    {% set s = params.S|float %}

    M104 {% for p in params %}{'%s%s' % (p, params[p])}{% endfor %}  ; Set hotend
temp
    {% if s != 0 %}
        TEMPERATURE_WAIT SENSOR=extruder MINIMUM={s} MAXIMUM={s+1}   ; Wait for
hotend temp (within 1 degree)
    {% endif %}
```

```
[gcode_macro M141]
description: Set Chamber temperature with slicers
gcode:
  SET_TEMPERATURE_FAN_TARGET TEMPERATURE_FAN=chamber_fan
TARGET={params.S|default(35)}

[gcode_macro M900]
gcode:
  {% if 'K' in params %}
    {% if 'E' in params %}
      SET_PRESSURE_ADVANCE EXTRUDER={params.E} ADVANCE={params.K}
    {% else %}
      SET_PRESSURE_ADVANCE ADVANCE={params.K}
    {% endif %}
  {% endif %}

[gcode_macro SET_E_MIN_CURRENT]
gcode:
  {% set e_current = printer['gcode_macro PRINTER_PARAM'].e_min_current %}
  M400
  SET_TMC_CURRENT STEPPER=extruder CURRENT={e_current} HOLDCURRENT={e_current}
  G4 P2000

[gcode_macro RESTORE_E_CURRENT]
gcode:
  {% set e_current = printer.configfile.settings['tmc2209 extruder'].run_current
%}
  M400
  SET_TMC_CURRENT STEPPER=extruder CURRENT={e_current} HOLDCURRENT={e_current}
  G4 P2000

[gcode_macro z_home_test]
gcode:
  {% set STEPS = params.STEPS|default(5)|int %}
  M400
  {% for step in range(STEPS) %}
    M400
    G91
    G1 Z+10 F1500
    G90
    G28 Z
  {% endfor %}
```

Sensorless.cfg

```
# thanks to zarboz for the original configs and shima for editing to speed up and
make more universal to creality printers :)
[homing_override]
axes: xyz
set_position_z: 0
gcode:
   {% set home_all = 'X' not in params and 'Y' not in params and 'Z' not in
params %}
   G90
   G0 Z5 F600
   {% if home_all or 'X' in params %}
    _HOME_X
   {% endif %}
   {% if home_all or 'Y' in params %}
    _HOME_Y
   {% endif %}
   # the duplication of these conditions is intentional
   {% if home_all or 'X' in params %}
    _HOME_X
   {% endif %}
   {% if home_all or 'Y' in params %}
    _HOME_Y
   {% endif %}
   {% if home_all or 'Z' in params %}
    _HOME_Z
   {% endif %}

[gcode_macro _HOME_X]
gcode:
    # Always use consistent run_current on A/B steppers during sensorless homing
    {% set RUN_CURRENT_X = printer.configfile.settings['tmc2209
stepper_x'].run_current|float %}
    {% set RUN_CURRENT_Y = printer.configfile.settings['tmc2209
stepper_y'].run_current|float %}
    {% set HOME_CURRENT = 0.8 %}
    SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CURRENT}
    SET_TMC_CURRENT STEPPER=stepper_y CURRENT={HOME_CURRENT}

    # Lift Z before Home X
```

```
    G1 Z10 F1200
    # Home
    G28 X
    # Move away
    G91
    G1 X-10 F1200

    # Wait just a second… (give StallGuard registers time to clear)
    G4 P1000

    # Set current during print
    SET_TMC_CURRENT STEPPER=stepper_x CURRENT={RUN_CURRENT_X}
    SET_TMC_CURRENT STEPPER=stepper_y CURRENT={RUN_CURRENT_Y}

[gcode_macro _HOME_Z]
gcode:
    # Always use consistent run_current on A/B steppers during sensorless homing
    G90
    {% set cartographer = ('cartographer' in printer.configfile.settings) %}
    {% set POSITION_X = printer.configfile.settings['stepper_x'].position_max/2
%}
    {% set POSITION_Y = printer.configfile.settings['stepper_y'].position_max/2
%}
    G0 X{POSITION_X} Y{POSITION_Y} F2800
    {% if cartographer %}
    G4 P2000
# wait required to prevent camera restarting too quickly causing mcu crash
    RUN_SHELL_COMMAND CMD=cam-stop
# stop camera services to prevent mcu overload
    {action_respond_info("Camera Stopped For Probing")}
# notify user of camera state
    {% endif %}

    G28 Z
    G90
    {% if cartographer %}
    RUN_SHELL_COMMAND CMD=cam-start
    {action_respond_info("Camera re-started!")}
# notify user of camera state
    {% endif %}

    {% if not cartographer %}
    G1 Z10 F1200
```

```
    {% endif %}

    # Wait just a second… (give StallGuard registers time to clear)
    G4 P1000

[gcode_macro _HOME_Y]
gcode:
    # Set current for sensorless homing
    {% set RUN_CURRENT_X = printer.configfile.settings['tmc2209
stepper_x'].run_current|float %}
    {% set RUN_CURRENT_Y = printer.configfile.settings['tmc2209
stepper_y'].run_current|float %}
    {% set HOME_CURRENT = 0.8 %}
    SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CURRENT}
    SET_TMC_CURRENT STEPPER=stepper_y CURRENT={HOME_CURRENT}

    # Home
    G28 Y
    # Move away
    G91
    G1 Y+10 F1200

    # Wait just a second… (give StallGuard registers time to clear)
    G4 P1000

    # Set current during print
    SET_TMC_CURRENT STEPPER=stepper_x CURRENT={RUN_CURRENT_X}
    SET_TMC_CURRENT STEPPER=stepper_y CURRENT={RUN_CURRENT_Y}
```