

3D printer Notes

==The printer==

The printer is part of a design by RepRap. It is intended for hobbyists and is entirely self assembled. You can buy kits that send you all the parts, but you will need to have some technical know how to get it operational.

Hardware:

Model: Rostock mini pro (purchased from 3d printer czar)

Steppers: Nema17 () for arms (1.3A, 1.8°)

SM42HT47-1684 nema 17 for extruder (1.68A)

Electronics: RAMPS1.4

Controller: Arduino Mega 2560

Stepper Drivers: Originally A4988, switched to DRV 8255

Power: 12 V laptop adapter.

==Building==

Printer was already built when I was asked to fix it. The rostock mini pro works through stepper motors connected to a spoke that drives a belt. The belt moves an armature up, which in turn moves the extruder unit back and forth. There are three of these armatures and steppers, one for each "axis" (not to be confused with cartesian axes).

Unfortunately, the website for 3d printer czar is down so it was very difficult to collect information related to this specific rostock model.

===Electronics===

My main task was related to the electronics under the hood of the 3d printer, as something wasn't working. I had to find out what was wrong through a process of elimination, as I didnt know what wasn't working.

The first step was to find a 3d printer controller that I could use to test the printer. I chose repeater for this, as it had its own firmware and had pretty useful print controls.

Once I had the software, I started to test each part.

At first, nothing seemed to work, and I had read online that its relatively easy to fry your RAMPS board. Furthermore, the initial problem with the printer was assumed to be with the voltage regulator. So I assumed that the board was fried, and we ordered a new one.

Once we got a new board, I faced the same problem. So the electronics on the board were fine.

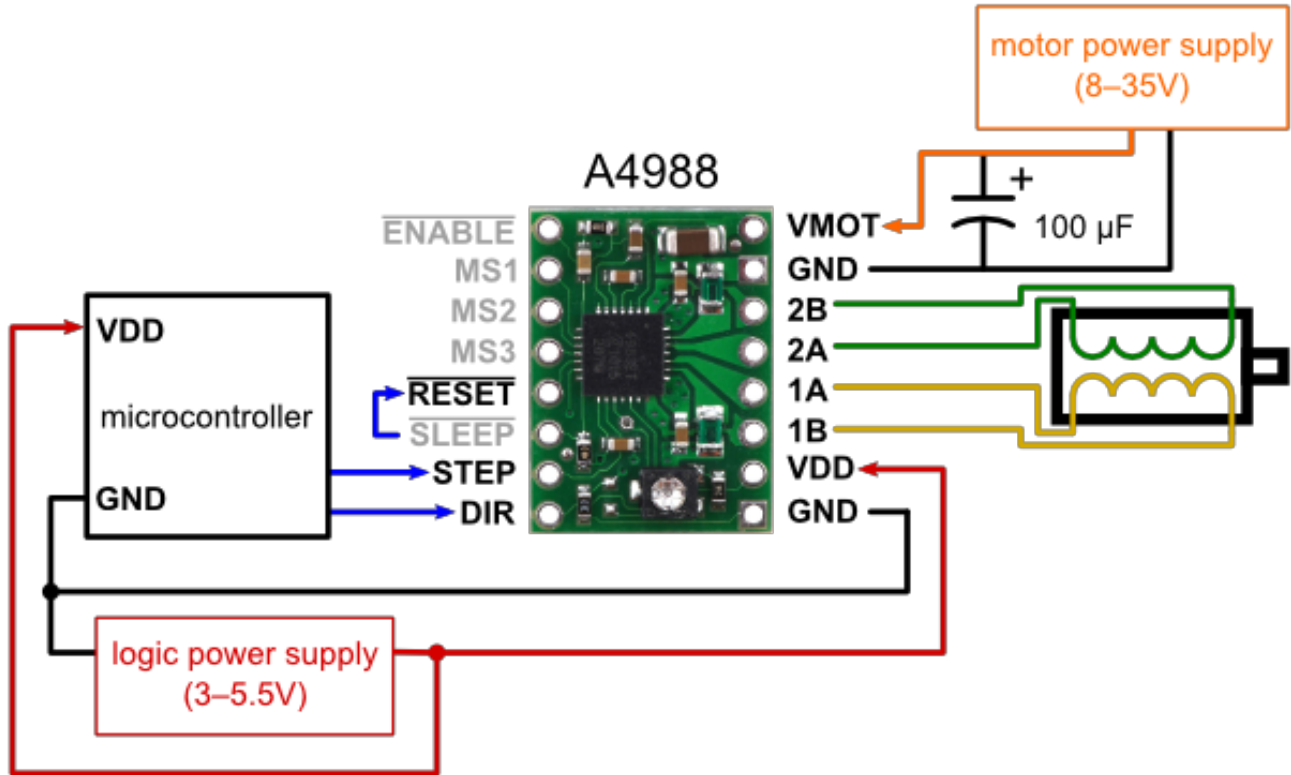
Repetier was able to detect the temperature of the extruder, but could not start the heater up or control the motors. Therefore, the thermistor and its associated circuitry was not the problem.

What remained was the endstops (which is unlikely to have any effect on the functioning of the printer as a whole) and the motor drivers.

I took the board apart, and tested each motor driver individually using the following thread.

<https://forum.pololu.com/t/how-to-check-if-a4988-is-fried/3542>

Essentially, this involved connecting the driver to a microcontroller like so;



And testing the outputs with a multimeter. What you want to do is connect Step and Dir to ground, and test 2B against 2A, and 1A against 1B as they act as input outputs for the coils. If these match up to the motor supply voltage, then the driver is functional. If they are significantly higher or lower, it isn't.

I found that 1 of the three motors wasn't functional, so we ordered a new batch. I found out that there was an "upgrade" called the DRV8255 which was capable of 1/32 micro stepping as opposed to the max 1/16 of the A4988, so we got those instead.

Note: when putting stepper drivers in, Double/Triple check that the orientation is correct as putting it in the wrong way will result in both the driver and the RAMPS getting fried. This also applies when you are testing it. Make absolutely sure you don't switch the Vmot and GND by accident as this will also fry the board. It happened to me.

But the motors still refused to budge. Then, as I was looking at the wiring diagram for the kossel mini, I realised that the power input might have been plugged into the wrong socket. There are 2 inputs on the RAMPS board: the 5 A and the 12 A inputs. When I got the printer, it was plugged into the 12 A socket. Since I didn't know anything about the printer and its machinations, I left it as I found it.

However, switching it to 5A allowed the motors to run. SLOWLY. And they sounded horrible.

The next step was to calibrate the steppers to run the motor. The factor that affects the calibration is the max current output of the driver. This needs to be at or under the current rating of the motor you are using.

Calibrating the driver involves turning the potentiometer on the driver (the screw). Clockwise decreases current output, and CCW increases it. It's important to first set the driver to the lowest possible current as apparently the default of chinese DRV8825s might fry it. In order to check the current output, you need to measure the voltage at the potentiometer. The voltage is called the

Vref and can be measured directly from the screw. I recommend using the driver's ground pin as the reference.

The Vref is related to the max current output as follows:

Max current output = V_{ref}^2 .

Adjusting the current therefore is just a matter of turning the potentiometer (CCW from the minimum) until you measure a Vref of 1/2 the steppers current rating. I recommend that this is done without the steppers connected.

Note: DO NOT MESS WITH THE STEPPER MOTOR PINS WHILE THE DRIVER IS CONNECTED TO THE 12 V voltage. This will fry your drivers. First turn off the power, then insert or remove the motor pins.

— — —

Once I had the motors running satisfactorily (now silent, but still slow), it was time to test the printer.

I uploaded a new firmware (Repetier 1.0.2) with a few changes involving

With most 3d printers, there are no sensors for determining the position. They instead rely on a system wherein all the steppers move the extruder in a specific direction until they cant move anymore. This position is called the Home, and acts as a reference for any future coordinates the extruder is moved to. All processes involving movement begin with homing.

Unfortunately, this wasn't working for me.

The problem was with the wiring of the end stop switches. Firstly, the wires that connected them were old, so they needed to be soldered again. Secondly, since I had removed everything to replace the RAMPS board, the switches weren't hooked up to the correct pins.

There are two different types of endstops: always open, or always closed. Always open switches only allow current to flow when the switch is closed, while always closed switches always have current flowing, and the circuit is broken when the switch closes.

I tested this using the connectivity test of the multimeter. Simply see if the two wires are connected, then press the switch and see how that changes. You need to know this as it needs to be specified in the firmware for the endstops to work properly.

The next step was to figure out if the wiring was correct, as end stop switches are "polarized". They need to be connected to the correct pins in order to work.

The ramps board's 6x3 end stop pins are organised as shown in the following table. The pins need to be connected to Signal and GND, with the black wire of the switch on the Signal Pina and the red wire on the GND pin. Note that this works only for end stop switches with two wires. If you have a endstop-PCB setup with 3 pins, the wiring is slightly different.

I hooked them up as follows.

	XMin	XMax	Ymin	Ymax	Zmin	Zmax
Signal	X black wire		Y black wire		Z black wire	
-/GND	X red wire		Y red wire		Z red wire	
+/VCC						

You can theoretically hook up the end stop wires to either the min or the max pin rows, as long as you specify it in the firmware configuration.

Once everything was hooked up properly, I had to set the details in the firmware. I will go over all details of the firmware later in case anyone reading this has similar problems. The configuration of the firmware takes a lot of effort to understand and set up, so maybe that section might speed things up.

Once the endstops were working, the printer could successfully home, and could therefore start printing. However, it was moving REEAAALLLYY slowly. Since i hadn't worked with a 3d printer before, I ignored it, but my colleagues in the lab said it used to be much faster.

Eventually I found out that it was a firmware problem. In cartesian printers, the Z axis motor is solely responsible for moving up and down the Z axis, which is used to move between layers. Therefore, the extruder doesnt have to move very quickly as the layer height is usually in the order of a few mm.

The default setting for the repeater firmware's Z axis motor speed is therefore set to about 2 mm/s for cartesian printers. Of course, for delta printers, the X, Y and Z "axes" motors are all responsible for the movement of the Z axis in cartesian coordinates. This default setting of the Z motor therefore limited the other two motors to the same speed, causing the entire 3d printer to move incredibly slowly. Setting the Z axis speed to the same as X and Y fixed the problem.

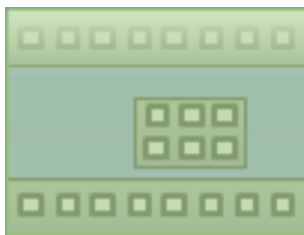
===Note on Microstepping and EEPROM===

While looking for the answer to the slowness problem, I looked to micro stepping.

All stepper motors have a given numerical value for how many degrees of rotation one step is. However, this can be chopped up into even smaller steps by the motor driver to provide increased accuracy and less noise at the cost of higher computational demands and slower motor movement. This is called microstepping.

There are three levels of micro stepping: 1/8, 1/16/ and 1/32. Logically, 1/8 micro stepping cuts the steps into 8ths, and so on for 1/16 and 1/32.

The A4988 motor driver is capable of only 1/8th and 1/16, but the DRV8825 is capable of 1/32 as well. In order to set the maximum number of micro steps, you will need to hook up jumpers on the six pins under the stepper driver on the RAMPS board:



The pins are connected to the MS1, MS2, and MS3 pins on the stepper driver from left to right (the leftmost pins on the ramps are MS1 and the rightmost are MS3). The combination of which pins are connected determines what microstepping the driver is configured for.

The different options are as follows:

Step size of stepper driver (A4988)

jumper		Yes/No		stepsize
ms1	ms2		ms3	
no	no		no	fullstep
yes	no		no	halfstep
no	yes		no	1/4step
yes	yes		no	1/8step
yes	yes		yes	1/16step

Step size of stepper driver (Drv8825)

jumper		Yes/No		step size
ms1	ms2		ms3	
no	no		no	full step
yes	no		no	half step
no	yes		no	1/4 step
yes	yes		no	1/4 step
no	no	yes		1/16 step
yes	no	yes		1/32 step
no	yes	yes		1/32step
yes	yes	yes		1/32step

Once you've hooked up the jumpers, and the driver is ready to go, you need to set the firmware to acknowledge the micro stepping change. The variable that determines this is "Steps per mm". If you use 1/32 micro stepping and don't double the steps/mm, your printer will move exactly and only half the distance you want it to, as the firmware counts steps, and not how many millimetres the extruder has moved.

I used 160 steps/mm to account for this.

However, when I altered the microstepping, I realised that changing steps/mm on my firmware had no change whatsoever on the printer. It was really frustrating as nothing I changed had any effect.

I eventually realized that the problem was with EEPROM. EEPROM is a type of internal memory. Repetier allows you to use the old EEPROM set that might be left over from the last firmware you used with the printer so you don't have to change all of the configuration options again. However, this also makes all the configuration you do within the new firmware completely useless.

The answer is simply to turn EEPROM off if you want to make your own changes.

=====

Once I had fixed the speed issue, all parts of the printer were working well and harmoniously. I added some filament and tested the extruder and heating element, and those were working fine as well.

Note: You have to heat the extruder in order to pull the filament out when you're done with it. When you pull it out, do it quickly as pulling it slowly might cause it to cool down in the upper section and get stuck. If it gets stuck here, it's nearly impossible to heat up again as the upper section of the extruder doesn't get hot.

Now the printer could print, but it did a horrible job of it. The next step was configuration, which involves tuning the firmware's settings and choosing print settings in order to get the best print.

==Calibration==

A delta printer works by moving its three motors which in turn change the position of the extruder head which prints layers by following a specific path. The path that the extruder head takes is defined by a series of numbers and letters called Gcode. Gcode is a set of specific instructions for how the extruder needs to move.

Once you have a model you would like to print (in stl format), you need to convert it to gcode so the printer can print it. This is done through a software called a slicer.

A slicer "slices" the model into individual layers, and further slices those layers into smaller, more linear units of volume that represent the layers of material that the printer lays down.

How the slicer slices therefore has a huge role to play in the final outcome of the print. The print quality can change drastically depending on what settings you choose.

Furthermore, you often need to change your settings when you change the material you are using, as each material (and each brand of the same material) will have different optimum settings such as temperature and flow rate.

Therefore, calibration is of paramount importance to the final quality of your print.

However, this is often a very tedious process and can be very time consuming especially if you don't know what you are doing.

I will go over some of the problems I encountered and how I solved them.

===Firmware===

The repetier firmware:

<https://www.repetier.com/firmware/v100/index.php> (this might be outdated, so just switch to the latest on the prompt).

The firmware is responsible for converting Gcode into movements, and controlling the various important parts of the 3d printer.

However, the repetier firmware contains code relevant to any and all types of 3d printers.

Therefore, the settings you choose need to be optimal for your printer.

I will go over some of the main settings that are relevant to a rostock mini pro.

Firstly, there are two ways of manipulating repetier's firmware settings. The link above directs to a configuration tool, which is basically a page with all the relevant configuration settings that spits out the firmware with all your selected settings at the end.

All of the settings you choose are stored in a file called configuration.h in the firmware's folder.

Editing this directly is the second way to change the settings. Although its a bit more confusing and intimidating, once you learn which lines change which settings, its a faster way to make small changes when calibrating.

Each setting in the configuration tool has a little line under it that mentions what variable it changes in the configuration file.

Now to go over some of the settings. If i dont mention the setting, assume it was left at the default value.

====General====

Now, starting with the general tab, you need to first specify the hardware details of your printer. This includes the processor, motherboard (RAMPS1.4 in our case) and the type of printer. The Rostock is a delta printer.

Next is the EEPROM. Go to the EEPROM section earlier for further details on this. The EEPROM basically overwrites any changes you make in this firmware, so unless you're already happy with your configuration settings, you want to turn this off (EEPROM disabled).

The Baud rate is a property of serial communication (the USB connection to the processor). If you're using a mac, its 115200 ANSI. However, it might be different in your case.

Next comes the dimensions of the printer. The delta 3d printer has a cylindrical print area.

The Z Length (Z_MAX_LENGTH) is the height of this cylinder. This property is extremely important as it determines whether your extruder will drag across the print surface and scratch it or miss it entirely and try to print in the air. It is really important to get this correct. More on this later (in the Bed height and level section). However for the first time you set it, you should push your extruder as far up as it goes, and measure the distance from the tip to the bed.

Next comes the diagonal rod length and horizontal radius when centered. You will need to measure these as well. The measurements are explained in the section above. It is important that you are as accurate as possible here. However, most 3d printers that follow a blueprint have rod lengths that are multiples of 5 mm, which makes it a bit easier to be accurate.

The Floor safety margin is used as a limiter to make sure that the carriages do not go too low. I set this to 15 mm, however, I don't believe it's too important as the carriage will never have to go that low. Just make it a positive non zero value.

The max horizontal radius is the radius of your build area. If you're using a circular plate like I did, just measure its radius. This value also acts like a limit that makes sure the printer does not try to print outside it.

I left everything after this as the default value. Z-correction requires a z-probe (a special device that probes the build area for variations). I didn't have one so I disabled it.

====*Mechanics*====

This part is about how your printer works and moves.

I left everything default until the X axis stepper motor section.

As mentioned before, the Z axis motor in delta printers play the same role as the X and Y motors, so the Z axis settings should be set to the same values as the X and the Y.

Resolution is relevant when you change the microstepping of the printer. For 1/16 microstepping, 80 steps per mm is the value you want. However, if you halve the microstepping, i.e. 1/32, you'll need to double the resolution to 160 or the printer will only move half the distance you want it to. The rest is fine as it is.

Next comes Endstops. I covered this briefly in the last section.

You need to make sure that you specify your endstops as the correct type (Switch on GND, normally closed in my case), and specify the pins you attached it to (Xmin, Ymin, Zmin in our printer). The Rostock I worked with didn't have Xmin, Ymin, Zmin pins, so I left these as not installed.

The rest of the settings were kept at default value.

====*Tools*====

Didn't change anything.

====*Features*====

The only thing I had to change here was the fan pin as the fan in my printer was hooked up to the port normally reserved for the heated bed instead of the fan pin.

The Print cooling fan pin was therefore set to "Heater 2 normally used for extruder 1".

====*The Rest*====

The rest of the configuration was left as is as the printer did not have any display.

To get the files, you have to hit "Download complete firmware incl. these settings" on the download tab.

===**Machine Settings**===

When you first hook up your 3d printer to the printing software (I started with repetier host, but later moved on to simplify3D).

simplify3D costs \$150 ("costs" *cough cough*) and I highly recommend it. It made a huge difference when I switched to it.

When you start up either software, you need to enter the machine settings (i.e. what kind of printer it is, and the dimensions). The firmware will take care of the rest (if it is recognised by the software, repetier host did not recognise marlin).

Setting these is pretty straightforward. Just enter the same values as in the firmware.

===Bed Height and Level===

The first main problem was the height of the bed with respect to the homing position, and how level it was when the extruder moved around it.

The problem arises because the printer never knows where exactly its extruder head is as it doesn't have a sensor to record this information. Instead, the printer relies on starting every print from a zero point called home. Every movement and position that the Gcode then wants the printer to move to is in reference to this point.

The home in a delta printer is when all the extruder carriages (the things that slide along the rails, driven by the steppers) hit the top of the printer, ie the highest position they can all take. Regardless of how well the printer was built, these three points then represent an isosceles triangle (a 2d plane) that acts as the 0 for the cartesian Z axis, and whose centre is the 0 for the cartesian x and y axes. Any movement in the Z direction will be along the normal to this starting plane.

This is the point from which problems with level and bed height arise. Assuming the home point's plane is exactly orthogonal (ie 90 degrees) to the rails of the printer, your bed needs to be perfectly parallel to the home point's plane.

If it is not, your print will be skewed. For example, the extruder could be dragging and scratching the bed when printing on one side of it, but print "in the air" when printing on the other side.

This is a huge problem for your print as the first few layers often determine the quality of the rest of the print.

The best way to fix it is to use the "jog controls" feature of your printer software. I suggest using a higher Z_MAX_LENGTH (the build height) than necessary, and manually moving your extruder down to meet the bed. Once it has met the bed in the middle, put a piece of paper between the bed and the extruder tip. The paper should be held down by the tip, but should still be able to slide around.

Next, move the extruder around in the X and Y directions without changing the Z and do the paper test at different points on the bed. Since the bed is perfectly flat, you might find that one side will often be too high (the extruder isn't touching the bed), and another might be too low (the paper can't move around under the extruder). Find the lowest point (sometimes the extruder presses down on this point so be careful), and increase the Z until the paper can move around. Note the Z height at this point as it will be the final Z height value you use.

You then need to level the bed out. From the lowest point, move the extruder around in the X and Y directions to manually probe the bed as you did before, but this time prop up those sides of the bed to meet the extruder. I used folded paper to prop up different sides of the bed.

Sometimes the original lowest point you start from will move up or down when you prop up the bed, so keep double checking if everything is in line with the paper test. Repeat the process as many times as necessary.

Also make sure all parts of your printer are tight and fit properly. If anything is loose, the whole process will be way off and print quality can change from one day to the next simply because some part of the mechanism has slipped a mm.

Once your done, your bed will probably no longer be parallel to the surface its resting on, but will be parallel to the home point, so your print will now be even along the bed.

Set the Z_MAX_HEIGHT in your configuration.h file and the same in your softwares machine settings to the height you recorded during the levelling process.

If your print is still a bit too high or low, you can always increase the Z height values by small increments. What is important is that the bed is perfectly level.

===Stickiness===

Another important factor that influences prints is the stickiness of the bed. Too sticky, and you'll have trouble getting the print off afterwards (which means you might have to do the levelling all over again because you moved the bed trying to pry the model off). Not sticky enough and the filament wont even stick to the bed and will instead get stuck to the extruder in a big blob.

The balance here is important. While my colleagues in the lab recommend glue stick (the one they use in offices) as thats what the last guys used, I found that it wasn't nearly sticky enough. I therefore instead used double sided tape.

I started with an industrial grade one I found somewhere, but it clung on to the model like a cat suspended above a body of water. This was way too strong.

So I moved on to a slightly less strong one. Namely:

Mario Brand D/s Tissue Tape Acrylic solvent Base Item No. SP 110

This worked very well. you could pry the model off easily, but the filament still stuck to it when printing.

===Movement of the bed===

One massive problem I faced was that sometimes the printed layers weren't exactly uniform and had little bits sticking out. The extruder would snag onto these bits and move the bed around as it wasn't secured. So, as the print progressed, layers of the model would slowly shift little by little. For example, if I was printing a cube, it would instead resemble a trapezoid in the end.

To fix this, I made three structures to secure the bed in place, as shown below.

It worked.

===Flow Rate, Speed and Temperature===

These are all settings that relate specifically to the type of model you are printing and the material you are using.

There is no set values that will give you the best print as even if you are using the printer I worked on, it might change over time due to a variety of factors.

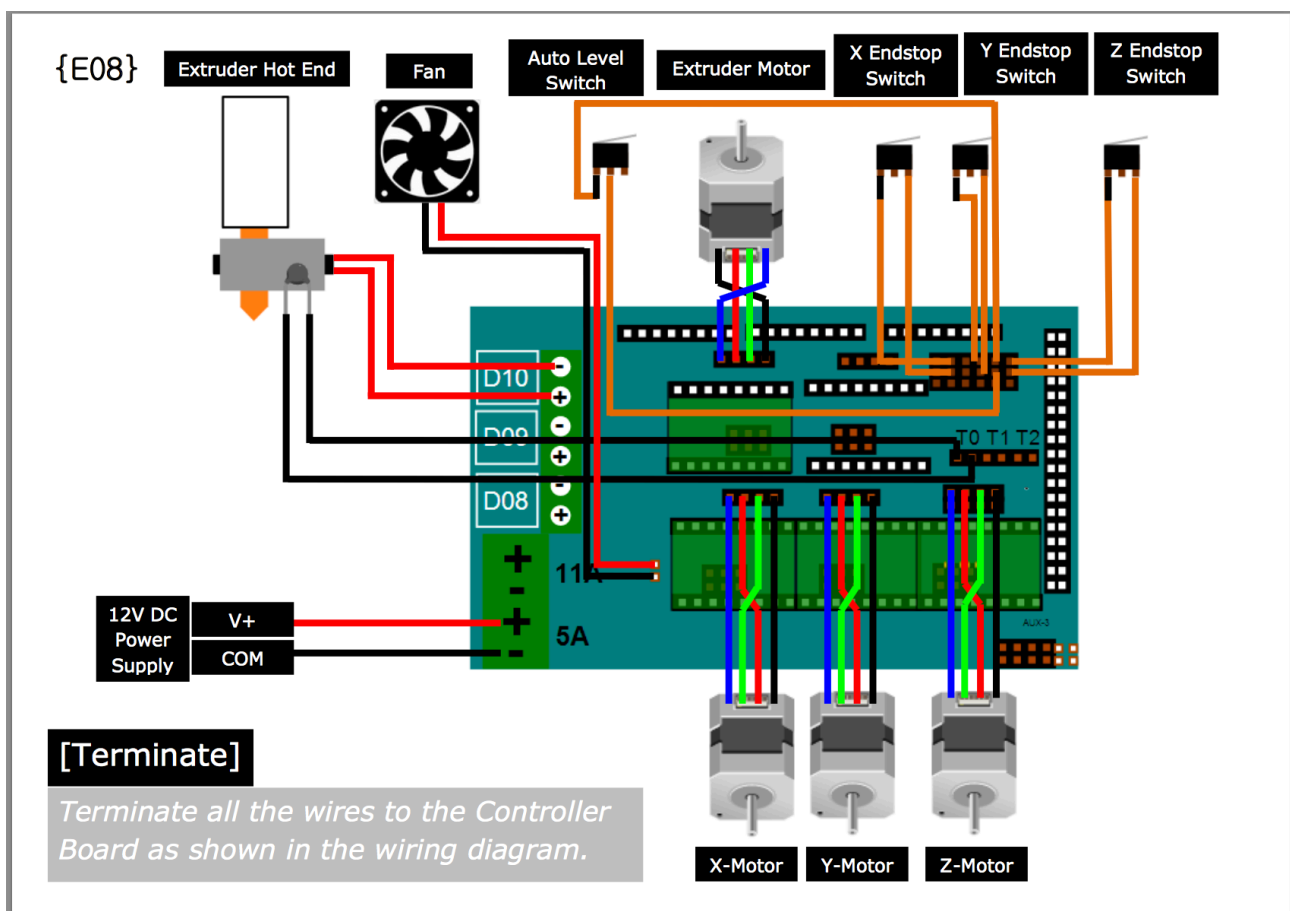
However, this guide form Simplify3d covers most problems you might encounter.

<https://www.simplify3d.com/support/print-quality-troubleshooting/>

However, this process is a slow and steady type deal. It will take time to get the perfect print. There are a bunch of calibration prints out there that are good ways to test the different processes involved in printing such as bridges and supports.

=====

I could not find the wiring diagram for the rostock, but the configuration of the kossel is very similar.



The wiring diagram is exactly the same, but the lab's rostock doesn't have a auto leveller, and the fan was connected to D09 instead of the fan pin.

=====