

My tech Example of the Hardware/Software technician pathway was setting up Octoprint for my 3D printer using a Raspberry Pi 3b. There are many ways of setting up Octoprint in both hardware and software, you can mix and match different cameras, use an old Android phone instead of a Pi, and much more. It is a beginner-friendly tool to set up, but that does not mean there will be no problems. The goal for this project is to get more familiar with Raspberry Pi's and to make my 3D printer a Network Printer.

To set up Octo Print, you have to download the Raspberry Pi Imager and the Octo Pi Imager. There are two Octo Pi OSes: new camera stack and stable. I chose the new camera stack because I want a security camera on the printer in case I'm not in the room to watch it. Once the Pi imager is uploaded to the SD card and in the Pi, you need to connect your choice of camera, a USB or ribbon cable camera. I chose to try both out because I wanted to see which I preferred, and so I could learn how each one performs. I found that the USB is easier and beginner-friendly but expensive, and the ribbon can be a little bit more finicky (probably because mine is so old), but cheaper. I preferred the ribbon cable because it is more customizable when it comes to mounts and wire lengths. Now that you have decided on those two choices, you need to get your parts together.



These are the parts I used, the twesers and screw drivers are for screws I forgot to take photos of. Some of these parts will cause problems in the future.

If you want more information on how exactly I put these parts together or some advice in case you want to do this yourself, look at the linked Dev log. But to put it simply, if you have a USB camera, put it in the USB port; if you have a ribbon cable camera, connect the pins in the camera and Pi (there should be a port on the Pi labeled Camera) to the ribbon cable. Put a USB mini (the cable type that powers the Pi) into one of the USB ports and connect it to your printer once the printer is ready to be powered on. Attach a monitor and keyboard to the Pi. Once those steps are complete, power on the Pi and log in using the username and password you set up previously in the Pi Imager. You will also use the ID you chose in the imager to get into OctoPrint, but if you forgot to set up an ID, type `ifconfig` into the terminal to see your IP address. Once you have your

IP or ID, go to your computer and type in the URL of `http://(your IP or ID).local`. Now you have to set up your wizard.

The wizard is not too complicated, and when you are setting it up, it is self-explanatory.

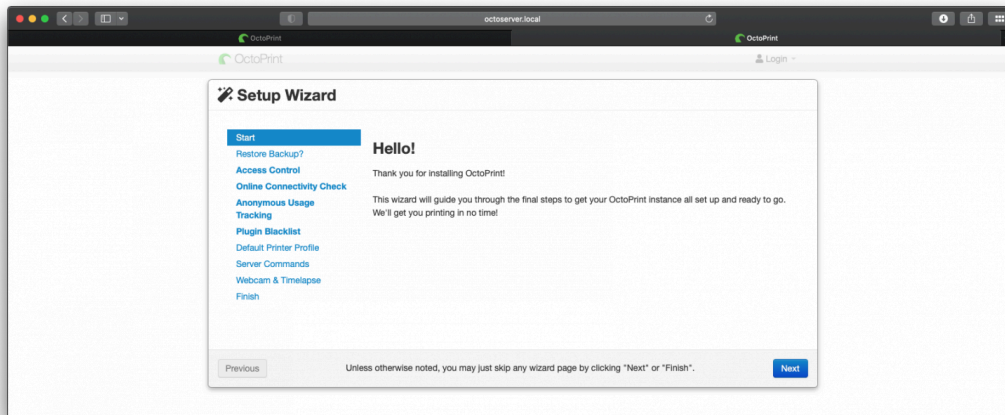
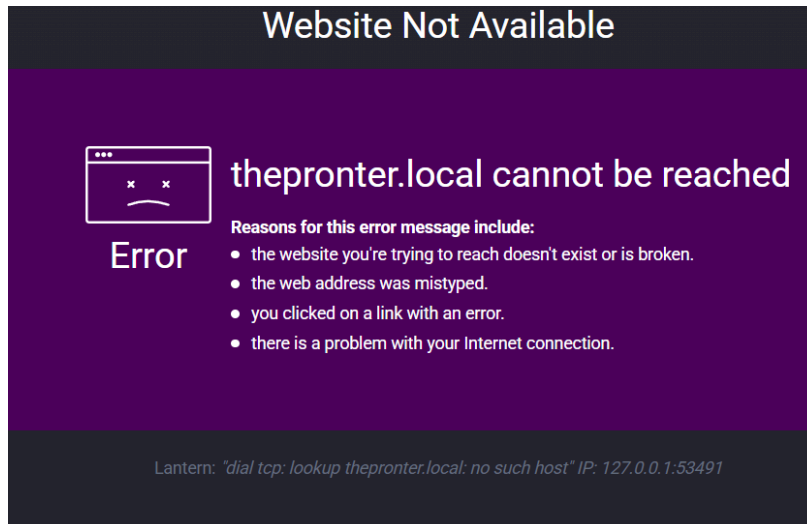


Image from [Octoprint® – Multiple Printers -](#)

I didn't have any backups because this was my first time. In the connectivity check, I set the 1.1.1.1 to 8.8.8.8 because it was the recommended number by the wizard itself. Then you can disable/enable the anonymous usage tracking and public blacklist program, I disabled both because I didn't want to have either on. I put my printer's model and name in for the printer setup. I have found that the website still works with a printer if it is a knockoff of a supported printer. The setup of the printing area size is 220\*220\*250. Then I left the Server Commands and Web cam settings alone because it was unnecessary for me to change them. Now that that was done, Octo Print was up and running. I could move my printer with my phone, or could I?

This is where my problems would all start in the project. Both of the wires were too short to reach the printer from where I wanted to set it up with its new networking capabilities, and one of them was broken because it was from 2013. So I got new wires that were longer. Now I could, nope. The screw for the camera mount was too short, so I got a new one. Then, the ribbon cable camera stopped working. It turns out the ribbon cable camera got corroded after 1 year of not being used, so I cleaned it. Once it was clean, I tested it out, and it was still not working. Then I tried to mess with the raspi-config, and this happened:

1. If you want to see my steps layed out in more detail look at this document:  
[https://docs.google.com/document/d/1jul0TfH3X\\_IIINPLrYRWzWDFSR9Uv2rTuk-kJp3uw5g/edit?usp=sharing](https://docs.google.com/document/d/1jul0TfH3X_IIINPLrYRWzWDFSR9Uv2rTuk-kJp3uw5g/edit?usp=sharing)



DO NOT do the raspi-config stuff in this video: [Integrating camera module with raspberry pi | troubleshooting](#). It is unsupported now and will break OctoPi.

As of now, the camera is working, I had to mess with some settings in the Boot.txt. The ribbon cable was my second biggest problem because my biggest problem was balancing home life, game design, and homework without losing time to work on this project. I did eventually overcome it. Here is a photo of the final product.

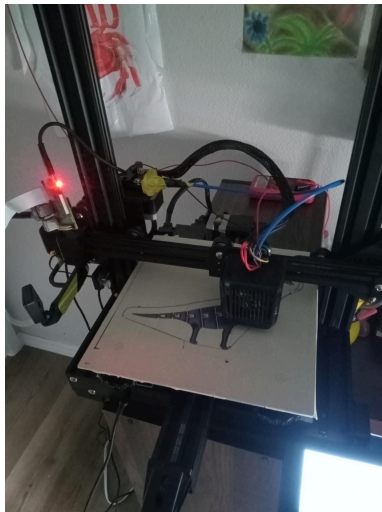


Photo of the final product set up in my room.

This project aligns with my focus area, the hardware/software technician focus area, because it involves hardware (the Pi, the printer, the cameras, and the computers) and software (the stuff I did in Octopi to get the ribbon cable to work). It leans more towards the hardware side, unlike my last tech example, a C # script. Turning my Knock-off Ender 3 into a Knock-off Ender 3 with networking capabilities by using a Raspberry Pi involves hardware and software.

1. If you want to see my steps layed out in more detail look at this document:  
[https://docs.google.com/document/d/1jul0TfH3X\\_IIINPLrYRWzWDFSR9Uv2rTuk-kJp3uw5g/edit?usp=sharing](https://docs.google.com/document/d/1jul0TfH3X_IIINPLrYRWzWDFSR9Uv2rTuk-kJp3uw5g/edit?usp=sharing)