

Setting Up Octoprint Web Interface for 3d Printer on Cubietruck

By [zhenning](#) in [Technology3D-Printing](#)

The screenshot displays the Octoprint web interface. On the left sidebar, the 'Connection' section shows 'Serial Port' and 'Baudrate' both set to 'AUTO'. Below these are checkboxes for 'Save connection settings' and 'Auto-connect on server startup', and a 'Connect' button. The 'State' section shows 'Machine State: Offline' and various print-related metrics. At the bottom of the sidebar is a 'Files' section with a search bar and a file named 'AZ.gcode' (2.1MB). The main content area has tabs for 'Temperature', 'Control', 'GCode Viewer', 'Terminal', and 'Timelapse'. The 'Temperature' tab is active, showing a graph with a green octopus watermark. Below the graph is a control table for 'Hotend' and 'Bed' temperatures.

	Actual	Target	Offset
Hotend	off	off °C Set ▾	0 °C Set
Bed	off	off °C Set ▾	0 °C Set



It would be interesting to be able to manage your 3d printer without having a cable attached to your PC. Most people have tried it out using raspberry pi. Initially, I bought this board for some other projects but it has been lying around for quite sometime. This board has 2 USB 2.0, VGA, HDMI, RS232, ethernet, SATA, SPDIF and some GPIO ports. So, I thought why not try out the Octoprint. Let's get started.

Step 1: Step 1: Setting Up Cubietruck

The cubietruck is a dual core ARM board by default the wifi is disabled. If you are using the ethernet instead, the wifi setup is optional. In order to perform the setup, you need a monitor and a keyboard. The cubietruck supports both VGA and HDMI. However, these options are done when selecting the image for this board.

Wifi:

1. Login to lubuntu: ***linaro/linaro***
2. add ***bcmdhd*** to `/etc/modules` (need to reboot)
3. Add nameserver ***8.8.8.8*** to `/etc/resolv.conf` (without this, the system will be connected to the wifi but no internet access)
4. Reboot

Step 2: Step 2: Adding External Harddisk

The image is loaded into the internal NAND RAM. To speed things up and add more space to the system, I have connected a 2.5" harddisk to the SATA port on-board. The system files can be moved to the harddisk by performing the following steps:

Moving the rootfs to harddisk:

1. `sudo su - root -->` change to root user
2. `mkdir /tmp/1 /tmp/target`
3. `mount /dev/nandb /tmp/1`
4. `fdisk -l ->` to locate your drive, sda? sdb?
5. `fdisk /dev/sda`
4. `mkfs.ext4 /dev/sda1`
5. `mount /dev/sda1 /tmp/target`
6. `(cd /tmp/1; tar --backup -c *) |tar -C /tmp/target -xv`
7. `mount /dev/nanda /mnt/`

8. `vi /mnt/uEnv.txt` (Change `root=/dev/sda1`)
9. `sync`
10. `umount /mnt`
11. `reboot`

Step 3: Step 3: Installing Octoprint

Install subversion at get pyserial for the Ramps 1.4 to connect with Cubietruck:

1. **`sudo apt-get install subversion`**
2. **`svn checkout svn://svn.code.sf.net/p/pyserial/code/trunk pyserial-code cd pyserial-code cd pyserial python setup.py build sudo python setup.py install`**

Install Python related packages and clone the Octoprint devel source for the latest development.

1. `cd ~`
2. `sudo apt-get install python-pip python-dev git git clone https://github.com/foosel/OctoPrint.git`
3. `cd OctoPrint`
4. `git pull git checkout devel sudo python setup.py install`

After that, **`./run`** to run the Octoprint server. It will be hosted in **`0.0.0.0:5000`** locally on the Cubietruck. You can also access it through other devices by replacing the **`0.0.0.0`** with the Cubietruck IP.

Step 4: Step 4: Adding Webcam

Additionally, you can also add webcam to stream the video of the print. Install the related packages and clone mjpg-streamer:

1. `cd ~`
2. `sudo apt-get install subversion libjpeg8-dev imagemagick libav-tools cmake`
3. `git clone https://github.com/jacksonliam/mjpg-streamer.git`
4. `cd mjpg-streamer/mjpg-streamer-experimental`
5. `make`

To test out the video stream: **`./mjpg_streamer -i "./input_uvc.so -y" -o "./output_http.so -w ./www"`** You should be able to get the live feed from your USB webcam using port **`8080`** with the Cubietruck IP or `0.0.0.0` locally in Cubietruck.

Step 5: Finally: Attach Your Printer to Cubietruck Usb

The screenshot displays the Octoprint web interface. On the left, the 'Connection' section shows 'Serial Port' and 'Baudrate' both set to 'AUTO', with options to save settings and auto-connect. Below this, the 'State' section indicates the machine is 'Offline'. The 'Files' section shows a file named 'AZ gcode' with a size of 2.1MB. On the right, the 'Temperature' control panel features a graph with a green octopus overlay and a table for temperature settings.

	Actual	Target	Offset
Hotend	off	off °C Set	0 °C Set
Bed	off	off °C Set	0 °C Set

Now your printer can be access locally through the web interface.

Source : <https://www.instructables.com/id/Setting-Up-Octoprint-web-interface-for-3d-printer/>

Heads-up

If you want to get OctoPrint up and running as fast as possible, it is highly recommended to take a look at [OctoPi 3.0k](#), which is an SD card image based on Raspbian already prepared with OctoPrint, Webcam support, HAProxy and SSL. Just download it, flash it to an SD card and you are good to go -- you can follow [this excellent video guide 2.1k](#) by [Thomas Sanladerer 163](#) who explains all needed steps in detail.

If on the other hand you want to run the latest versions of Raspbian, OctoPrint and all the other packages, and get a sense of how it all fits together, do follow the instructions below (**warning: not for the faint of heart**).

Important: This guide expects you to have a more than basic grasp of the Linux command line. In order to follow it you'll need to know:

- how to issue commands on the shell,
- how to edit a text file from the command line,
- what the difference is between your user account (e.g. pi) and the superuser account root,
- how to SSH into your Pi (so you don't need to also attach keyboard and monitor),
- how to use Git and
- how to use the Internet to help you if you run into problems.

This is **not** a "Linux for Beginners guide", those can be found for example [here 726](#) and [here 249](#). For some Git basics please take a look [here 256](#).

Basic setup

For the basic package you'll need Python 2.7 (should be installed by default) and pip. OctoPrint's dependencies will be installed by pip:

```
cd ~
sudo apt update
sudo apt install python-pip python-dev python-setuptools python-virtualenv git
libyaml-dev build-essential
mkdir OctoPrint && cd OctoPrint
virtualenv venv
source venv/bin/activate
pip install pip --upgrade
pip install https://get.octoprint.org/latest
```

You may need to add the pi user to the dialout group and tty so that the user can access the serial ports:

```
sudo usermod -a -G tty pi
sudo usermod -a -G dialout pi
```

You should then be able to start the OctoPrint server:

```
pi@raspberrypi ~ $ ~/OctoPrint/venv/bin/octoprint serve
```

* Running on <http://0.0.0.0:5000/>

Updating & changing release channels & rolling back

OctoPrint should offer to update itself automatically and also allow you to switch to other Release Channels out of the box.

If for whatever reason you want or need to perform any of this manually however, perform the following commands to install

- the current version from the "Stable" release channel:

```
source ~/OctoPrint/venv/bin/activate
pip install https://get.octoprint.org/latest
```

- the previous version from the "Stable" release channel:

```
source ~/OctoPrint/venv/bin/activate
pip install https://get.octoprint.org/last
```

- the current version from the "Maintenance RC" release channel:

```
source ~/OctoPrint/venv/bin/activate
pip install https://get.octoprint.org/maintenance-rc
```

- the current version from the "Devel RC" release channel:

```
source ~/OctoPrint/venv/bin/activate
pip install https://get.octoprint.org/devel-rc
```

Support restart/shutdown through OctoPrint's system menu

In Settings > Commands, configure the following commands:

- Restart OctoPrint: `sudo service octoprint restart`
- Restart system: `sudo shutdown -r now`
- Shutdown system: `sudo shutdown -h now`

Note : If you disabled Raspbian's default behaviour of allowing the pi user passwordless sudo for every command, you'll need to explicitly allow the pi user passwordless sudo access to the `/sbin/shutdown` program for the above to work. You'll have to add a sudo rule by creating a file `/etc/sudoers.d/octoprint-shutdown` (as root) with the following contents:

```
pi ALL=NOPASSWD: /sbin/shutdown
```

Automatic start up

Download the init script files from OctoPrint's repository, move them to their respective folders and make the init script executable:

```
wget https://github.com/foosel/OctoPrint/raw/master/scripts/octoprint.init &&
sudo mv octoprint.init /etc/init.d/octoprint
wget https://github.com/foosel/OctoPrint/raw/master/scripts/octoprint.default &&
sudo mv octoprint.default /etc/default/octoprint
sudo chmod +x /etc/init.d/octoprint
```

Adjust the paths to your octoprint binary in `/etc/default/octoprint`. If you set it up in a virtualenv as described above make sure your `/etc/default/octoprint` is modified like this:

```
DAEMON=/home/pi/OctoPrint/venv/bin/octoprint
```

Note also the removed `#` at the start of the line, uncommenting it and making it effective!

Then add the script to autostart using `sudo update-rc.d octoprint defaults`.

This will also allow you to start/stop/restart the OctoPrint daemon via

```
sudo service octoprint {start|stop|restart}
```

Make everything accessible on port 80

If you want to have nicer URLs or simply need OctoPrint to run on port 80 (http's default port) due to some network restrictions, I recommend using [HAProxy 212](#) as a reverse proxy instead of configuring OctoPrint to run on port 80. This has the following advantages:

- OctoPrint does not need to run with root privileges, which it would need to be able to bind to port 80 thanks to Linux privileged port restrictions
- You can make mjpg-streamer accessible on port 80 as well
- You can add authentication to OctoPrint
- Depending on the HAProxy version you can also use SSL to access OctoPrint

Setup on Raspbian is as follows:

```
pi@raspberrypi ~ $ sudo apt install haproxy
```

I'm using the following configuration in `/etc/haproxy/haproxy.cfg`, for further examples take a look

<https://community.octoprint.org/t/setting-up-octoprint-on-a-raspberry-pi-running-raspbian/2337>

Raspberry Pi 3D printer

Hi guys, this time I would like purpose you how to control a 3D printer with Raspberry Pi.

For this project you can start with the installing of Raspbian OS

How to install Raspbian

Install Raspbian in pretty easy, you have to download the Raspbian image from the official Raspberry's website at [this link](#). If you want tu use PiTFT screen use [this linkfor resistive](#) and [this link for capacitive](#).

After the download extract the zip file and write the .img file on Raspberr's SD card, if you use windows you can use a programm called win32diskimager or if you use another OS you can follow all the passage in the home of this site.After the installation of Rappbian to control 3D printer we need another software caled OCTOPRINT.

What is Octoprint?

Octoprint is a software that allows you to wirelessly controll and monitor 3D printer for Raspberry Pi (and other platforms), It's host software that tourns Raspberry Pi into a serveradding a web interface.

How to install Octoprint on Raspbian

After the Raspbian installation go to the terminal and go to the config menù with:

```
"sudo raspi-config"
```

When you are in the menù expand the file sistem and reeboot the Pi, after the rebooting connect the Raspberry to a internet connection and start to install octoprint.

To install Octoprint open the terminal an put in the same sequences:

- 1) `cd ~`
- 2) `sudo apt-get install python-pip python-dev python-setuptools python-virtualenv git libyaml-dev build-essential`
- 3) `git clone https://github.com/foosel/OctoPrint.git`
- 4) `cd OctoPrint virtualenv venv`
- 5) `./venv/bin/pip install pip --upgrade`
- 6) `./venv/bin/python setup.py install`
- 7) `mkdir ~/.octoprint`

You may need to add the pi user to the dialout group and tty so that the user can access the serial ports:

- 1) `sudo usermod -a -G tty pi`
- 2) `sudo usermod -a -G dialout pi`

You should then be able to start the OctoPrint server:

```
~/OctoPrint/venv/bin/octoprint
```


* Running on your browser <http://0.0.0.0:5000/> (if you use raspberry pi through VNC or putty and you want to visit the octoprint page with your pc replace all the zero before :5000 with Raspberry Pi's IP address) Now you need to insert a username and a password to set an Octoprint account and after this you need to be logged in and you are able to use Octoprint. Now we are ready to set the automatic start up.

We have to modify two text files called `~/OctoPrint/scripts/octoprint.init` and `~/OctoPrint/scripts/octoprint.default`, for this put in the terminal the command:

- 1) `sudo nano ~/OctoPrint/scripts/octoprint.init`
- 2) `sudo nano ~/OctoPrint/scripts/octoprint.default`

and in each file copy and paste this line of code:

```
DAEMON=/home/pi/OctoPrint/venv/bin/octoprint7
```

(In each file a line of code about DAEMON is already existing so add an # at the start of line and copy the previous line of code under it)

Now copy the file in the respective folder and make it executable with:

- 1) `sudo cp ~/OctoPrint/scripts/octoprint.init /etc/init.d/octoprint`
- 2) `sudo chmod +x /etc/init.d/octoprint`
- 3) `sudo cp ~/OctoPrint/scripts/octoprint.default /etc/default/octoprint`

Then add the script to autostart using in the terminal the following command:
`sudo update-rc.d octoprint defaults`

This will also allow you to start/stop/restart the OctoPrint service using in the terminal:

```
sudo service octoprint {start|stop|restart}
```

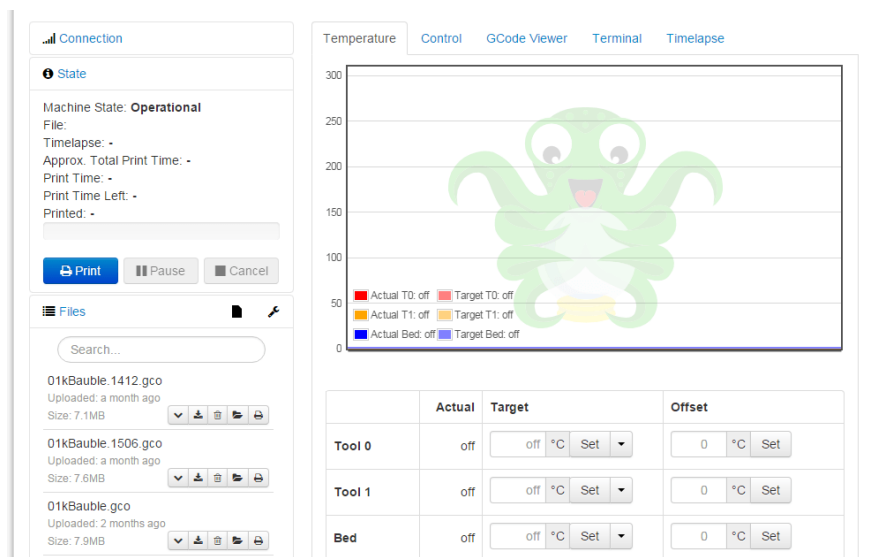
(you have to use this command in this way:

```
sudo service octoprint start
```

```
sudo service octoprint stop
```

```
sudo service octoprint restart )
```

The web page when you will work it will be like this



The screenshot shows the Octoprint web interface. On the left, the 'State' section indicates the machine is 'Operational'. Below this, there are buttons for 'Print', 'Pause', and 'Cancel'. A 'Files' section lists three G-code files: '01kBauble.1412.gco', '01kBauble.1506.gco', and '01kBauble.gco'. On the right, the 'Temperature' section features a graph with a green octopus mascot. Below the graph is a table for temperature control:

	Actual	Target	Offset
Tool 0	off	off °C Set	0 °C Set
Tool 1	off	off °C Set	0 °C Set
Bed	off	off °C Set	0 °C Set

If you use PiTFT screen

On PiTFT screen It isn't easy use octoprint service on web page so you need to use a touch UI interface so lets start with the last step.

Go to Octoprint web page with your computer (so you need to insert in the search bar Raspberry Pi's IP address and add :5000 at the end of address) and go to setting, go to plugin menager click get more and search TouchUI; install it and after restart Octoprint server with "sudo service octoprint restart".You are done! Enjoy with your Raspberry Pi and 3D printer!!

<https://raspiproject.altervista.org/3d-printer-and-raspberry-pi/amp/>