<u>Orange Pi – OctoPrint – Webcam – Gpio – No-IP</u>

Download the latest Ubuntu-armbian for your Orange Pi from https://www.armbian.com/download/ For Orange Pi Pc Plus I took (https://www.armbian.com/orange-pi-pc-plus/) the Armbian_5.25_Orangepipcplus_Ubuntu_xenial_default_3.4.113_desktop.img Login as root apt-get update Create a new User (new user "pi") sudo adduser pi sudo adduser pi sudo (put the new user in sudo group) sudo visudo pi ALL=(ALL) NOPASSWD:ALL (Put this on the last line of the opened file, no passwd for sudo commands) Ctrl + O, Enter, Ctrl + X. (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 sudo su pi (Log in as pi user)

Install OctoPrint

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

cd ~

```
sudo apt-get install python-pip python-dev python-setuptools python-virtualenv git libyaml-dev build-essential
git clone https://github.com/foosel/OctoPrint.git
cd OctoPrint
python -m virtualenv venv
./venv/bin/pip install pip --upgrade
./venv/bin/python setup.py install
mkdir ~/.octoprint
```

Automatic start up

 Adjust the paths to your octoprint binary in both
 ~/OctoPrint/scripts/octoprint.init
 and

 ~/OctoPrint/scripts/octoprint.default
 If you set it up in a virtualenv as described above make sure your /etc/default/octoprint is modified like this:

sudo nano ~/OctoPrint/scripts/octoprint.init DAEMON=/home/**pi**/OctoPrint/venv/bin/octoprint Ctrl + O, Enter, Ctrl + X. sudo nano ~/OctoPrint/scripts/octoprint.default OCTOPRINT_USER=**pi** (Make sure to use the name of new user that you create) DAEMON=/home/**pi**/OctoPrint/venv/bin/octoprint Ctrl + O, Enter, Ctrl + X.

Copy the script files to their respective folders and make the init script executable:

sudo cp ~/OctoPrint/scripts/octoprint.init /etc/init.d/octoprint sudo chmod +x /etc/init.d/octoprint sudo cp ~/OctoPrint/scripts/octoprint.default /etc/default/octoprint sudo update-rc.d octoprint defaults (add the script to autostart)

This will also allow you to start/stop/restart the OctoPrint daemon via sudo service octoprint {start|stop|restart}

~/OctoPrint/venv/bin/octoprint (test - start the OctoPrint server)

Open your browser, go to the machine address at port 5000http://<your Orangepi's IP>:5000If you did everything correct your Octoprint UI should come up (give it a minute or two for the first launch).Here you will be asked to enter admin password – **DO IT (strong password).** That's not your orangepi-pi defaultpassword, that's a password you'll be using to log in to Octoprint web interface. Make something up.

Next, you need log in and go to Settings (top right corner). Here you can set up your server commands and webcam paths:

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

```
stream URL: http://<your Orangepi's IP>:8080/?action=stream snapshot URL: http://127.0.0.1:8080/?action=snapshot
```

You can click save and go back to your terminal. Ctrl + C (stop the test of the OctoPrint server)

Webcam

For webcam and timelapse support, you'll need to download and compile MJPG-Streamer:

~
lo apt-get install subversion libjpeg8-dev libav-tools libv4l-dev cmake
clone https://github.com/jacksonliam/mjpg-streamer.git
mjpg-streamer/mjpg-streamer-experimental
no Makefile
Check if there is "PLUGINS + = input_raspicam.so", and comment it out ("#PLUGINS + = input_raspicam.so";). We do not need this
l + O, Enter, Ctrl + X.

export LD_LIBRARY_PATH=. make	
Compile. Do not connect the camera yet.	
ls /dev/video*	(Theoretically, the output should be empty)
Connecting camera repeat command.	
ls /dev/video*	(If you see the / dev / video0, then everything should be fine, go ahead)

sudo ./mjpg_streamer -i "./input_uvc.so" -o "./output_http.so"

check at http://<your Orangepi's IP>:8080/?action=stream

For some webcams (including the PS3 Eye) or if you get "Unable to set format: 1196444237 res: 640x480 Init v4L2 failed !! exit fatal i: init_VideoIn failed" you'll need to force the YUV mode by using the following start command: ./mjpg_streamer -i "./input_uvc.so -y" -o "./output_http.so"

Please be aware that YUV mode will put additional strain on your Orangepi's CPU which will then lower its performance, possibly up to the point of causing printing issues. If your camera requires the-yparameter to function, consider replacing it with one that doesn't.

Note: If your webcam requires switching to YUV mode in order to work at all, it is strongly recommended to instead use a webcam that natively supports MJPG. For YUV cameras mjpg_streamer will need to transcode all data from the camera to MJPG on your Raspberry Pi, which will put a lot of strain on its CPU (YUV mode at around 30-40% vs MJPG mode at around 1-2%). This MIGHT negatively influence print quality, so better get yourself a cheap MJPG compatible webcam. See this wiki page (<u>https://github.com/foosel/OctoPrint/wiki/Webcams-known-to-work</u>) for a compatibility list and steer clear of cams that require -y to work. Also you may need to run ./mjpg_streamer -i "./input_raspicam.so -fps 5" -o "./output_http.so"

If you now point your browser to http://<your Orangepi's IP>:8080/?action=stream, you should see a moving picture at 5fps. (If you get an error message about missing files or directories calling the output plugin with -o "./output_http.so -w ./www" should help.)

(if running mjpg-streamer)

sudo usermod -a -G video pi (to allow user access to a device video, it is necessary to add it to the appropriate group) sudo make install cd ~ sudo nano webcam-streamer

#!/bin/bash

Daemon=mjpg_streamer DaemonBase=/usr/local DaemonArgs="-i \<mark>"input_uvc.so\" -o \"output_http.so\"</mark>"

case "\$1" in start) eval LD_LIBRARY_PATH=\${DaemonBase}/lib \${DaemonBase}/bin/\${DaemonArgs} >/dev/null 2>&1 & echo "\$0: started" ;; stop) pkill -x \${Daemon} echo "\$0: stopped" ;; *) echo "Usage: \$0 {start|stop}" >&2 ;; esac

Ctrl+O, Enter, Ctrl+X.

sudo chmod +x webcam-streamer sudo mv webcam-streamer /usr/local/bin/ sudo nano ~/.octoprint/config.yaml

(add the following lines - <u>spaces are significant in this file -- a misplaced space</u> <u>here will stop Octoprint from running</u>)

system:

```
actions:

- action: streamon

command: sudo /usr/local/bin/webcam-streamer start

confirm: false

name: Start video stream

- action: streamoff

command: sudo /usr/local/bin/webcam-streamer stop

confirm: false

name: Stop video stream

Ctrl+O, Enter, Ctrl+X.
```

If you also want autostart of the webcam you need to add the following line to /etc/rc.local (Just make sure to put it above the line that reads exit 0). /usr/local/bin/webcam-streamer start

Make everything accessible on port 80 - HAProxy

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** as a reverse proxy instead of configuring OctoPrint to run on port 80. cd ~

sudo apt-get install haproxy sudo nano /etc/haproxy/haproxy.cfg

global	
maxconn 4096	
user naproxy	
daemon	
log 127.0.0.1 local0 debug	
log 121.0.0.1 localo debug	
defaults	
log global	
mode http	
option httplog	
option dontlognull	
retries 3	
option redispatch	
option http-server-close	
option forwardfor	
maxconn 2000	
timeout connect 55	
timeout server 15min	
frontend public	
bind :::80 v4v6	
use_backend webcam if { path_beg /webca	am/ }
default_backend octoprint	
backend octoprint	
reqrep ^([^\:]*)\ /(.*) \1\ /\2	
option forwardfor	
server octoprint1 127.0.0.1:5000	
backend webcam	
regrep $([^:/^]) / webcam/(.*) 1 / 2$	
server webcam1 127.0.0.1:8080	
Ctrl+O, Enter, Ctrl+X.	
sudo service haproxy start	
sudo nano ~/.octoprint/config.yaml	(Edit the following lines. Remember: spaces are significant in this file)
add the following line in "conver" coction:	
add the following line in server section.	
host: 127.0.0.1	(make the server bind only to the loopback interface)
Edit the following lines:	
webcam:	
ffmpeg: /usr/bin/avconv	
snapshot: http://127.0.0.1:8080/?action=snap	shot
stream: /webcam/?action=stream	
Ctrl+O, Enter, Ctrl+X.	

Restart the server. OctoPrint should still be available on port 80, including the webcam feed (if enabled). sudo service octoprint restart

Install WIRINGOP library to manage gpio(s)

git clone https://github.com/zhaolei/WiringOP.git -b h3 cd WiringOP chmod +x ./build sudo ./build gpio readall (--- and check grio.7) gpio mode 7 out gpio write 7 1 gpio readall (--- and check again grio.7)

+		+	+	+	++	-Orange Pi++	+	+	+	+
1	BCM	wPi	Name	Mode	V	Physical V	Mode	Name	wPi	BCM
+		+ 	3.3v			1 2		5v		+
i	12	8	SDA.0	ALT5		3 4		5V		Ì
i	11	9	SCL.0	ALT5		5 6		0v		Ì
i	6	7	GPIO.7	OUT	1	7 8 0	ALT3	TxD3	15	13
Í		ĺ	0v			9 10 0	ALT3	RxD3	16	14
	1	0	RxD2	ALT3	0		ALT3	GPIO.1	1	110
	0	2	TxD2	ALT3	0	13 14		0v		1
	3	3	CTS2	ALT3	0	15 16 0	ALT3	GPIO.4	4	68
			3.3v			17 18 0	ALT3	GPIO.5	5	71
	64	12	MOSI	ALT4	0	19 20		0v		1
	65	13	MISO	ALT4	0	21 22 0	ALT3	RTS2	6	2
	66	14	SCLK	ALT4	0	23 24 0	ALT4	CE0	10	67
		1	0v			25 26 0	ALT3	GPIO.11	11	21
	19	30	SDA.1	ALT4	0	27 28 0	ALT4	SCL.1	31	18
	7	21	GPIO.21	ALT3	0	29 30		0v		
	8	22	GPIO.22	ALT3	0	31 32 0	ALT3	RTS1	26	200
	9	23	GPI0.23	ALT3	0	33 34		0v		
	10	24	GPIO.24	ALT3	0	35 36 0	ALT3	CTS1	27	201
	20	25	GPIO.25	ALT3	0	37 38 0	ALT3	TxD1	28	198
			0v			39 40 0	ALT3	RxD1	29	199
	BCM	wPi	Name	Mode	V	Physical V	Mode	Name	wPi	BCM

sudo nano ~/.octoprint/config.yaml

(Edit the following lines. Remember: spaces are significant in this file)

server:

commands:

serverRestartCommand: <u>gpio write 7 1 &&</u> sudo service octoprint restart systemRestartCommand: <u>gpio write 7 1 &&</u> sudo shutdown -r now systemShutdownCommand: <u>gpio write 7 1 &&</u> sudo shutdown -h now #for OctoPrint #for OrangePi #for OrangePi

#for Printer

and add

system: actions: - action: Shutdown Printer command: gpio write 7 0 name: Printer Off - action: Start Printer command: gpio write 7 1 name: Printer On Ctrl+O, Enter, Ctrl+X.

sudo service octoprint restart

Force the gpio output at startup

sudo nano /etc/init.d/octoprint After the line #Author put: gpio write 7 1 gpio mode 7 out Ctrl+O, Enter, Ctrl+X. sudo nano /etc/default/octoprint After the line OCTOPRINT_USER put: gpio write 7 1 gpio mode 7 out Ctrl+O, Enter, Ctrl+X.

sudo update-rc.d octoprint defaults

This way you can control your Printer status – On or Off, but with the Orange pi On. When Shutdown system the printer will be On.

DDNS Client - No-IP

Sign Up at www.no-ip.com to Create Your Free Hostname

cd /usr/local/src sudo wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz sudo tar xzf noip-duc-linux.tar.gz cd no-ip-2.1.9-1 sudo make sudo make install If you get "make not found" or "missing gcc" then you do not have the gcc compiler tools on your machine. You will need to install these in order to proceed.

> **To Configure the Client** sudo /usr/local/bin/noip2 -C

You will then be prompted for your username and password for No-IP, as well as which hostnames you wish to update. Be careful, one of the questions is "Do you wish to update ALL hosts". If answered incorrectly this could affect

hostnames in your account that are pointing at other locations.

Please enter the login/email string for no-ip.com: <your_login>

Please enter the password for user 'your_login': <your_password> Please enter an update interval:[30] <interval time in minutes>

Do you wish to run something at successful update?[N] <N>

Now that the client is installed and configured, you just need to launch it. Simply issue this final command to launch the client in the background:

sudo /usr/local/bin/noip2

This way the application is restarted automatically when the system boots. sudo nano /etc/init.d/NoIp

#! /bin/sh
/etc/init.d/noip
BEGIN INIT INFO
Provides: noip
Required-Start: \$remote_fs \$syslog
Required-Stop: \$remote_fs \$syslog
Default-Start: 2 3 4 5
Default-Stop: 0 1 6
Short-Description: Simple script to start a program at boot
Description: A simple script from www.stuffaboutcode.com which will start / stop a program a boot / shutdown.
END INIT INFO

If you want a command to always run, put it here

Carry out specific functions when asked to by the system case "\$1" in start) echo "Starting noip" # run application you want to start /usr/local/bin/noip2 ;; stop) echo "Stopping noip" # kill application you want to stop killall noip2 ;; *) echo "Usage: /etc/init.d/noip {start|stop}" exit 1 ;; esac exit 0

Ctrl+O, Enter, Ctrl+X.

sudo chmod 755 /etc/init.d/NoIp

f you also want autostart/restart of the NoIp you need to add the following lines to /etc/rc.local just before exit 0. sudo nano /etc/rc.local

sudo /etc/init.d/NoIp stop sudo /etc/init.d/NoIp start exit 0 Ctrl+O, Enter, Ctrl+X.

Static ip

sudo nmtui (change ip4 to manual and add a <static address>/24, also add the <default gateway>) sudo /etc/init.d/networking restart sudo reboot

From your router, PORT Forwarding to your OrangePi local ip, like port 80.

For me the settings was: Router's local ip \rightarrow login \rightarrow Advanced Setup \rightarrow NAT \rightarrow Virtual Server :

Application	HTTP_Server	
Protocol	ALL(UDP & TCP)	
Start Port Number	80	
End Port Number	80	
Local IP Address	192.168.1.100	(static local ip of my OrangePi)
Start Port Number(Local)	80	
End Port Number(Local)	80	

Check if the port is open from portchecktool.com. If not, change a router and do the same.

Check from your browser your Noip-hostname:	http://hostname.ddns.net
if you set different port just check	http://hostname.ddns.net: <portnumber></portnumber>

if you see your router's configuration page to your public ip, just test from an outside network - not your home's wifi

```
my file ~/.octoprint/config.yaml :
..
••
server:
 commands:
 serverRestartCommand: gpio write 7 1 && sudo service octoprint restart
 systemRestartCommand: gpio write 7 1 && sudo shutdown -r now
 systemShutdownCommand: gpio write 7 1 && sudo shutdown -h now
 firstRun: false
 host: 127.0.0.1
 secretKey: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 seenWizards:
 corewizard: null
 cura: null
 softwareupdate: null
system:
 actions:
 - action: Shutdown Printer
 command: gpio write 7 0
 name: Printer Off
 - action: Start Printer
 command: gpio write 7 1
 name: Printer On
 - action: streamon
 command: sudo /usr/local/bin/webcam-streamer start
 confirm: false
 name: Start video stream
 - action: streamoff
 command: sudo /usr/local/bin/webcam-streamer stop
 confirm: false
 name: Stop video stream
webcam:
 ffmpeg: /usr/bin/avconv
 snapshot: http://127.0.0.1:8080/?action=snapshot
 stream: /webcam/?action=stream
```

REMEMBER: MAINS ELECTRICITY CAN KILL YOU



http://www.orangepi.org/orangepibbsen/forum.php?mod=viewthread&tid=594

https://github.com/foosel/OctoPrint/wiki/Setup-on-a-Raspberry-Pi-running-Raspbian

http://www.noip.com/support/knowledgebase/installing-the-linux-dynamic-update-client/

http://www.stuffaboutcode.com/2012/06/raspberry-pi-run-program-at-start-up.html

http://fabrogg.unblog.fr/2016/04/18/installer-octoprint-sur-une-orange-pi-one-pour-piloter-son-imprimante-3d/

https://www.youtube.com/watch?v=b6ZagKRnRdM Using Relays and Relay Boards with the Raspberry Pi

http://3dtoday.ru/blogs/eta4ever/orange-octopus-an-ordinary-webcam/