Installation
of
Octofarm / printerview

Daniel Krah

December 20, 2018

Contents

1 Overview

The topic of this PDF is the installation of a octoprint setup which can handle several printers and provides an overview page which is provided by printerview or octofarm. This Howto is based on Ubuntu Mate.
The main concept is that all printers and webcams are connected to one single computer. To protect the PC and make the installation more easy an active usb hub is used. Each printer will be connected to his own octoprint instance. Each instance is bond to a separate ip.
On each ip the corresponding octoprint and mjpeg-streamer instance is reachable. So the used ports are the same for all ip's. The ip begin at .100 and are above the dhcp managed range. On the first ip (100) runs the caddyserver which delivers printrun or later octofarm. Both give you an overview over the octoprint instances.

1.1 Used hardware

Computer

```
Mainboard   :   ASRock Q1900m
CPU         :   Intel® Quad-Core Processor J1900 (2/2,42 GHz)
GPU         :   Intel® HD-graphics for Intel Atom® (Z3700 series)
RAM         :   4GB DDR3
Connectors  :   3x USB internal, 2x USB 2.0 external, 1x USB 3.0 external 2 x SATA2 3.0
Diskspace   :   160 GB HDD for developement / 16 GB USB 3.0 Stick for production
Network     :   1000m Ethernet
```

Cameras

```
Logitech      :   C270 (USB)
                  C310 (USB)
mjpg-streamer :   input_testpicture (2x)
```

3D printer

```
Prusa i3 mk2 clone   : MKS gen 1.4, E3D V6 (clone), MK42 (Prusa), original Pinda probe,
                       Display
Prusa i3 mk2.x clone : MKS gen 1.4, E3D V6 (clone) on Titan Extruder,
                       MK42 (Orballo Printing modded to 24V), Induktiver Sensor, Display
Software             : Prusa Firmware 3.10
```

Various

```
Power supply : Pico PSU Clone 60w
USB-Hub      : active 7 port USB-Hub
```

Network

```
Router : Fritzbox 7490 with T-DSL 16.000/2.500 (DualStack)
         IP-address:
         192.168.178.1
Q1900m : IP-addresses:
         192.168.178.40  - (static DHCP - for internet access)
         192.168.178.100 - Caddyserver (Printview / later Octofarm)
         192.168.178.101 - Octoprint 1.3.9, Prusa i3 mk2 clone, C310
         192.168.178.102 - Octoprint 1.3.9, Prusa i3 mk2.x clone, C270
         192.168.178.103 - Octoprint 1.3.9, virtual Printer, input_testpicture
         192.168.178.104 - Octoprint 1.3.9, virtual Printer, input_testpicture
```

2 Software installation

Prerequisite is an installation of Ubuntu Mate.
(Any other version of Ubuntu should work as well ...)

2.1 Update the basic system and install the Openssh-Servers

Update of the package list and subsequent update of the respective packages + installation of openssh server

```
sudo apt update
sudo apt upgrade
sudo apt install open-ssh
```

2.2 Network configuration

2.2.1 Set the ip address

The configuration file should have similar content.

/etc/network/interfaces

```
cat /etc/network/interfaces
auto enp4s0
iface enp4s0 inet dhcp
```

Open the file in an editor and insert 5 IP addresses in an area not managed by the DHCP server.

The first IP for the overview page, the rest for the respective Octoprint instances.

Set the ip addresses

```
sudo nano /etc/network/interfaces
... #append at the end
iface enp4s0 inet static
  address 192.168.178.100/24
iface enp4s0 inet static
  address 192.168.178.101/24
iface enp4s0 inet static
  address 192.168.178.102/24
iface enp4s0 inet static
  address 192.168.178.103/24
iface enp4s0 inet static
```

2.2.2 Check the IP addresses

You can check the IP addresses with:

Check the IP addresses

```
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen
 ↪  1000
    link/ether d0:50:66:9c:7f:e0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.178.40/24 brd 192.168.178.255 scope global dynamic noprefixroute enp4s0
       valid_lft 863727sec preferred_lft 863727sec
    inet 192.168.178.100/24 brd 192.168.178.255 scope global secondary enp4s0
       valid_lft forever preferred_lft forever
    inet 192.168.178.101/24 brd 192.168.178.255 scope global secondary enp4s0
       valid_lft forever preferred_lft forever
    inet 192.168.178.102/24 brd 192.168.178.255 scope global secondary enp4s0
       valid_lft forever preferred_lft forever
    inet 192.168.178.103/24 brd 192.168.178.255 scope global secondary enp4s0
       valid_lft forever preferred_lft forever
    inet 192.168.178.104/24 brd 192.168.178.255 scope global secondary enp4s0
       valid_lft forever preferred_lft forever
    inet6 2003:da:fbd6:7a00:f2b7:fb53:57dd:da2f/128 scope global dynamic noprefixroute
       valid_lft 6931sec preferred_lft 3331sec
    inet6 2003:da:fbd6:7a00:8d8b:ad1c:44f0:52b4/64 scope global temporary dynamic
       valid_lft 6929sec preferred_lft 956sec
    inet6 2003:da:fbd6:7a00:fd2e:196a:87b6:c5b3/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 6929sec preferred_lft 956sec
    inet6 fe80::f2b7:fb53:57dd:da2f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: wlp2s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen
 ↪  1000
    link/ether 20:c9:c0:40:60:ae brd ff:ff:ff:ff:ff:ff
```

The second device now shows 6 IP addresses. The first one in this case is the one assigned by the Fritzbox.

2.3 Installation of required packages for (Octoprint, mjpegstreamer, ...)

We install all required packages in one go:

Install required packages

```
sudo apt install python-pip python2.7 virtualenv git subversion libjpeg8-dev imagemagick ffmpeg
 ↪  libavcodec-extra  libav-tools mjpegstreamer gphoto2 cmake pkg-config curl
```

2.4 Cloning the respective repositories of Octoprint, mjpegstreamer, ...

Change to the home folder

```
# go to the home folder
cd ~

# check
pwd
/home/danielkrah
```

Cloning the repositories

```
# Octoprint
git clone https://github.com/foosel/OctoPrint.git
Klone nach 'OctoPrint' ...
# mjpg-streamer
git clone https://github.com/jacksonliam/mjpg-streamer.git
Klone nach 'mjpg-streamer' ...
# bindp for binding applications to their specific IP/port
git clone https://github.com/yongboy/bindp.git
Klone nach 'bindp' ...
```

2.5 Installation von Octoprint

The installation of octoprint is quickly done. The setup follows later ...

Installation of Octoprint

```
# Installation of OctoPrint
# Create virtual enviroment and install
cd OctoPrint/ && virtualenv venv
./venv/bin/pip install pip --upgrade
./venv/bin/python setup.py install
```

2.6 Installation of mjpg-streamer

Now follows the installation of mjpeg-streamer which is the streaming server for the webcams

Installation of mjpg-streamer

```
cd ~
cd mjpg-streamer/mjpg-streamer-experimental
make clean
make
make install
```

2.6.1 Compilation of input_testpicture plugin

This step is basically optional, but may be worthwhile later when using Octofarm ;-)

Compilation of input_testpicture plugin

```
# input_testpicture
cd ~/mjpg-streamer/mjpg-streamer-experimental/plugins/input_testpicture
make
cp input_testpicture.so ../
cp input_testpicture.so ../../
```

I copy the compiled plugin into the 2 folders because I'm not sure where it belongs. Normal should: /mjpg-streamer/mjpg-streamer-experimental/plugins/ fit but darkly I remember that there were problems. So having two strings to one's bow it definitly works and it's not that big ...

2.7 Creation of UDEV-rules for the webcams

2.7.1 Determining the necessary values

to be sure that the assignments of the cameras to the respective printers are still correct after a restart or accidental disconnection, UDEV rules must be created.

For this you need the respective product and device ID's which are given in the format ID XXXX: XXXX.

In my case, I use 2 Logitech USB cameras (C270 and C310).

The corresponding values are:

- 046d:081b ⇒ C310
- 046d:0825 ⇒ C270

These values are obtained by taking the vendor and device id from lsusb.

Output of lsusb

```
lsusb

...
Bus 001 Device 007: ID 046d:0825 Logitech, Inc. Webcam C270
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass          239 Miscellaneous Device
  bDeviceSubClass         2 ?
  bDeviceProtocol         1 Interface Association
  bMaxPacketSize0        64
  idVendor           0x046d Logitech, Inc.
  idProduct          0x0825 Webcam C270
  bcdDevice            0.10
  iManufacturer           0
  iProduct                0
  iSerial                 2 AF1943F0
  bNumConfigurations      1
  Configuration Descriptor:



  Bus 001 Device 002: ID 046d:081b Logitech, Inc. Webcam C310
  Device Descriptor:
    bLength                18
    bDescriptorType         1
    bcdUSB               2.00
    bDeviceClass          239 Miscellaneous Device
    bDeviceSubClass         2 ?
    bDeviceProtocol         1 Interface Association
    bMaxPacketSize0        64
    idVendor           0x046d Logitech, Inc.
    idProduct          0x081b Webcam C310
    bcdDevice            0.10
    iManufacturer           0
    iProduct                0
    iSerial                 2 4B8254A0
    bNumConfigurations      1
    Configuration Descriptor:

...
```

2.7.2 Create a new UDEV rule file for the webcams

Create a new UDEV rule file as root with sudo

```
sudo nano /etc/udev/rules.d/01-webcam-usb.rules
```

The content is inserted on the basis of the determined data with the following content ...

Create a new UDEV rule file as root with sudo

```
SUBSYSTEMS=="usb", ATTR{idVendor}=="046d", ATTR{idProduct}=="081b", ATTRS{serial}=="4B8254A0",
 ↪  SYMLINK+="logitechC310dollyMK3"
SUBSYSTEMS=="usb", ATTR{idVendor}=="046d", ATTR{idProduct}=="0825", ATTRS{serial}=="AF1943F0",
 ↪  SYMLINK+="logitechC270dollyMK2"
```

- SUBSYSTEMS==”usb”,
  - Subsystem USB both cameras are connected via USB.
- ATTRidVendor==”046d”,
  - Is identical for both cameras because they are both Logitech with id 046d.
- ATTRidProduct==”081b” / ATTRidProduct==”0825”,
  - 081b and 0825 for C310 and C270
- ATTRSserial==”4B8254A0” und ATTRSserial==”AF1943F0”,
  - The specification of the serial number is only necessary if 2 cameras of the same model are used.
    In my case, this would not be necessary, but I still add them in case if another camera would be added.
- SYMLINK+=”logitechC270dollyMK2” and SYMLINK+=”logitechC310dollyMK3”
  - You can freely choose the name of the symlink.
    In my case I use the schema manufacturer + model + printer name.
    In my case 2 Prusa type printers.

2.7.3 Apply the new UDEV rules without rebooting

If you want to apply the rule without rebooting, you can do that with the following command.

Update UDEV rules

```
sudo udevadm trigger
```

2.7.4 Check the new UDEV rules

Check the UDEV rules

```
ls /dev/ | grep -i logit
# Output
logitechC270dollyMK2
logitechC310dollyMK3
```

”logit” is part of the symlink and since both are Logitech cameras and the names have been chosen accordingly, both are output.

2.8 Installation of Caddy-Server

Caddy Server installation

```
# caddy
curl https://getcaddy.com | bash -s personal
 ↪  http.cors,http.git,http.minify,http.nobots,http.prometheus,http.upload
```