Video surveillance
Single Board Computer
Administrators Guide

Bjorn Hugsted

August 4, 2021

Contents

1

1   Abstract

We describe common tasks that will need to be done for all or any computer
system used for video Surveillance. These systems consist of a variety of
hardware and software placed on different observation stations and viewsta-
tions. Although different there are a still a lot of common tasks related to
the programs and operating systems. This writing decribes tasks that are
related to the install and configuration of operating systems and programs
used with the video surveillance project.

2   Introduction

We have for a long time worked on video and still image transfer from
remotely placed camera servers. Our main focus has been to provide rea-
sonably priced observation stations that send video streams through the use
of the open source mjpg-streamer. The original developer, Andreas Wiklund
[Mjpg-streamer], seem to have left the project altogether. The project was
taken over by Tom Stoeveken [Mjpg-streamer]. What is not good is that
it seems from the SourceForge page that even this developer has no more
time to spend on that project. The project may have been transferred to
[Mjpg-streamer] under the repositori "jacksonliam". We tried this once, but

the build did not complete. Still we use a dirty trick to get hold of a working version.

Around new year 2021 we started working on our own version of capture and network streaming programs. We use the names "mjpeg_streamer" and "h264_streamer". The reason for developing our own sources was both to have a dependable codebase as well as gaining more understanding of the inner workings of the Video for Linux 2 (V4L2) drivers and interface. Currently the mjpeg_streamer is working wery well on the Raspberry Pi 4 and streaming 1280x720 at 30 frames per second is no problem at all. More about our own streamer attempts in section 11.

We also need to install and configure software for the viewstations, here we rely on the well proven OpenCV libraries [OpenCV]. Actually viewing the video in the netbrowser also works fine.

We employ many different types of computers - from different vendors and manufacturers. Most of them are Single Board Computers (SBC) from the UK based Raspberry PI [Raspberry PI] or the Chinese FriendlyElec [FriendlyElec]. All the time we have been using some flavour of Linux, viz. Debian or Ubuntu. We try to document all different observation stations as well as viewstations and found that there were quite a lot of repetitions. Therefore we have placed common tasks in this separate paper.

3   Connecting to 3.3 V serial ports

For machines with no graphics hardware and/or with network access disabled by default one may have to resort to the serial ports brougth out on pin headers on the PIs and NanoPIs. We will therefore discuss this task before the installation of the Operating System (OS).

The logic levels on the NanoPIs and RaspberryPIs are all 3.3 V. Connecting any 5 V logic will most likely make permanent damage. At all times we have used a bitrate of 115200 bit/s. There should be 8 data bits, 1 stop bit, and no parity. As there is no Request To Send or Clear to Send (RTS, CTS) pins the flow control should NOT be RTS/CTS. As terminal emulator program we have been very satisfied with the PuTTY [PuTTY]. One should therefore make sure one has PuTTY [PuTTY] installed on the desktop. For debian or Ubuntu a simple "apt-get install putty" is enough.

When connecting two serial devices the Transmit Data (TxD) on one device should go to Recieive Data (RxD) on the other and vica versa. The ground connection is common. On the host computer these devices will usually have device files like /dev/ttyUSB0, /dev/ttyUSB1, etc... One may notice that the TxD to RxD and vica versa scheme is not as what was used in the old days when a terminal (Date Terminal Equipement) were to be connected to

a modem (Data Connect Equipement). Here the cabling were straight from TxD to TxD and what is an output pin on the DTE is an input pin on the DCE.

3.1 Pin headers on the Raspberry PI

On the Zeros he UART0 is brought out to the header pins 6 - GND, 8 - TxD0, and 10 - RxD0 on the PI 40 pins header. Pins 2 and 4 are +5V power and there are also several ground pins. Remember that the UARTs of the Raspberry PI is not enabled by default.

3.2 Serial ports on the NanoPIs

On the NanoPIs, except Duo, one of the UARTs are brought out on a special four pin header together with a +5 V pin that can be used for powering the NanoPI. Observe that FriendlyElect themselves do not promise that this will work as the NanoPIs usually are specified to require a 2 A power supply. The pins of the debug header are 1: GND, 2: +5 V, 3: UART_TXD0/GPIOA4, 4: UART_RXD0/GPIOA5/PWM0. On the NanoPIs the UART0 is enabled and will give a login prompt. The NanoPi Duo and Duo2 is special. Here there is no debug header and the serial port is brought out on pins on one sied of the computer board.

3.3 FTDI TTL-232R-Rpi

On the FTDI [FTDI] TTL-232R-Rpi USB to serial converter the black lead is GND while the yellow lead is RxD, and then the orange lead is TxD. This converter is specially designed for use with the Raspberry PIs. Connect this converter cable to the Raspberry PI as: black to pin 6, yellow to pin 8, and orange to pin 10. The FTDI cable does not provide any +5 V power lead. This cable may no longer be easily avaliable.

To connect to the NanoPIs use the debug port that has four pins. Pin 1 is marked with a square and possibly the word GND. Do NOT connect any lead to pin 2 as this is +5 V power. Connect the black lead to pin 1, the yellow lead to pin 3, and the orage lead to pin 4.

3.4 PIMORONI CAB0400 USB to UART cable

PIMORONI does provide an USB to UART serial console cable at about Kr 55. At his low price it may even be soldered directly to the SBC without using a pin header. This cable also has the +5 V brought out so it may even

power the machine. This cable has +5 V power supply on the red lead, GND on the black, RxD on white, and TxD on the green lead. When connected to a Linux 4.9.0-3-amd64 machine the dmesg will report that the converter is an PL2303 [Prolific] and what serial device file it is connected to. later cables (after 2017) uses a different hardware, viz. the SiLabs CP210X. Lady Ada [Lady Ada] has written a tutorial that explains this how to connect both versions to different operating systems. Connect this cable to a Raspberry PI as: red to pin 2 or pin 4 (or both), black to pin 6, white to pin 8, and green to pin 10.

3.5 USB to TTL Serial Cable from FriendlyElec

This is actually not a cable at all, but a small circuit board that has an USB connector at one end and a 4-pin header on the other. A short four-lead cable connects the converter to the debug serial port of most NanoPIs. One MUST make sure that the black gound pin is correctly placed. This converter may also power the NanoPI and this may be turned on or off by a sliding switch on the board. As USB power may not be able to power the NanoPIs we have always put this switch in the off position. The on,off markings are hardly visible, but off is when the slider is closest to the crystal can.

4 Installing Operating Systems

For both the Raspberry PI and The NanoPI the installation of the Operating System (OS) is quite simple. It is is usually done by downloading the distribution that normally is compressed as a zip archive. Then one will uncompress the file and copy it onto a Secure Digital (SD) memory card. The exception is the NanoPI versions with built in embedded MultiMediaCard (eMMC). Here the distribution is a special version that contains the real distribution to be copied to the eMMC.

4.1 Copying an image onto a μSD card

We assume that a normal computer with a Linux destribution is used for download and storage of the different distribution images. One may use the Graphical User Interface and extract the archine to an image or use the unzip command from the terminal. Either way one end up with a large file that shall be copied to the card. When the card is inserted in a cardreader/writer a new device file appears in the /dev dircetory. One may find out what device that has been created by looking at the output of the dmesg command. This command does need superuser privileges and one may either use the sudo

before every command or log in as root with "su -". If we assume the image has the name image.iso and the device is /dev/sdb the command to use for copying to the card is: "dd if=image.iso of=/dev/sdb bs=4M". As superuser one has the power to erase the hard drive of the machine so be very sure that /dev/sdb is the μSD card and not any local disk.

The DD reports the number of bytes written as well as the write speed. The latter may be an indication of the maximum speedf that may be obtained when writing data to the card. Copying friendlycore operatintg system of size 2.5 GB to a 32 GB SanDisk Extreme Pro microSDHC UHS-I Card resulted in the report from DD: "2468347904 bytes (2.5 GB, 2.3 GiB) copied, 126.561 s, 19.5 MB/s". This again leads to assume that the card cannot sustain more than 20 MB/s although this may wery well depend just as much on the writer used.

4.2   Installing Raspbian

For the Raspberry PI there were originally just two distributions to choose from i.e. the strech and the stretch-lite. Later also the buster and buster-lite became avaliabe. And even later the names changed from raspbian to raspios. All raspbian distributions seems to derived from Debian that, of course, is derived from the Linux sources. The "lite" is a small distribution without any window system the "desktop" contains the X-windowing system and there has also appeared a "with selected software". The lite version could be ideal for the obeservation stations that shall not be connected to any display. **OBSERVE**: For the Rasdpberry Pi Zeros we have had problems with connecting a keyboard to the single USB-port. A safe way to gain access to the machine is to enable the serial port and use a serial terminal connection. See the instructions in section 4.2.1 on how to do this.

4.2.1   2018-03-13-raspbian-stretch-lite

OBSERVE: Of the four images I list up this is the only on that I have actually been able to make the mjpg-streamer work on. Although the 2018-04-18 has not been tested. The image is as small as 1.7 GB. It could possibly fit on a 2 GB card if one were avaliable. Once the image has been copied to the μSD card do not remove the card from the writer right away. To make the system start up the sshd server mount he boot partition and create a file with name ssh: "# touch ssh". At the same time one may enable the serial port on GPIO pin. This is done by adding the "enable_uart=1" to the /boot/config.txt. We added the UART line just below the "dtparam=audio=on" and it works. Then mount the root partition and fix up the /etc/network/interfaces to set the correct IP-address for the Ethernet interface. See belov for how to do this. From user pi change thwe

password to SimplePiPass and then the password for root (sudo passwd root) to SimpleRootPass. Log in as root either at the console or through ssh from another machine.

In the raspi-config enable the camera: raspi_config -> Interfacing Options -> PI Camera. This requires reboot or shutdown and start. Once we had problem with reboot so we use shutdown and new start.

Update ther repository list for upgrading and/or installing packages. Here there has been problems due to resolving to IPv6 addresses. Force the apt to use IPv4 addresses instead with: echo 'Acquire::ForceIPv4 "true";' | sudo tee /etc/apt/apt.conf.d/99force-ipv4. The problem with Internet Protocol version 6 has nothing to do with the Raspbian servers themselves, rather somthing in between blocks the version 6

Copy the mjpg-streamer-nanopi.tar to user root's home directory scp mjpg-streamer-nanopi.tar root@surv11:mjpg-streamer-nanopi.tar and untar it. tar -xvf mjpg-streamer-nanopi.tar. Proceed to section 10 for compiling and starting the streamer.

4.2.2 2018-04-18-raspbian-stretch-lite

This distribution has bnot been tested as video server.

4.2.3 2019-04-08-raspbian-stretch-lite

Logged in at the console. Changed pasword for pi and root. Set fixed address in /etc/network/interfaces, set 192.168.1.2 as nameserver in /etc/reolv.conf, set /etc/hostname to surv11. Set allow root login to yes in /etc/ssh/sshd_config. Enabled and started ssh.service with the systemctl command. Finally I did a reboot and logged in as root through ssh. Now I should install package libjpeg-dev, but even the first apt update did not succeed in connecting to raspbian.raspberrypi.org. Probably because the host resolver to internet protocol version 6. As for the 2018-03-13 set the apt to only use IPv4. Still cannot ping the server, but update is working and finally I could install libjpeg-dev. On this distribution the bcm2835-v4l2 was NOT loaded after boot. After all this the image from the mjpg-streamer is garbled.

4.2.4 2019-07-10-raspbian-buster-lite

Image is 2 GiB. After enabling the camera (# raspi-config -> Intefaceing Options -> PI Camera) the machine did not start. A power off and on worked. Still with a rpi-update and power off and reboot the image from the mjpg-streamer was garbled. On this distribution the bcm2835-v4l2 was

loaded after boot. Since it was running nicely on the 2018-03-13-raspbian-stretch-lite I just reinstalled this version and abandoned the buster for the time beeing. Still it is more likely that we ourselves made some mistake than the image is actually not usable.

4.2.5   2019-09-26-raspbian-buster-lite

This image was tested on a Raspberry PI Zero W with camera mounted. We connected the Philis 240V5QDAB display on the HDMI port. Also the pi keyboard and mouse was connected to the OTG micro-USB through the adapter delivered with the Raspberry Pi Zero Adaptor Kit (PIMORONI RPI-010). We need the screen and keyboard to get ther WiFi going. It is in principle possible to arrange for the Zero W to start up and connect to a WiFi network without user intervention. When we applied power the boot messages was visible and everything looked good. In the end a login prompt was presented and we logged in as user pi with the password raspberry. At a later ytime we were not able to get the keyboard working and had to connect to the serial port at /dev/tty1 instead. This may be a special problem for the Rapberry PI Zeros. Also on this distribution the bcm2835-v4l2 was loaded after boot.

On the Raspbian the superuser has no password so one cannot log in as root. We fixed this with **$ sudo password root** and set our usual root password. Then we logged in as root which is more convenient for system setup. It seemed like the buster lite does not have NetworkManager installed. We used wpa_cli to connect to our local network and saved the configuration. Details are given in section 5.3.3. We did a reboot and the Zero W connected to the same WiFi network on startup.

Since we intend to run the Zero W without any screen or keyboard the Secure Shell must be running and we are so bold as to allow remote root logins. We fixed up /etc/ssh/sshd_config to allow root login and then did the **# systemctl enable ssh.service** and **# systemctl start ssh.service**. Then we were able to log in remotely as root. We set the hostname to suit our local convention, i.e. survxx or tempxx. This time we selected tempxx and arranged for dnsmasq at server to assign the corresponding IP-address, viz. 192.168.1.10.

4.2.6   2020-02-13-raspbian-buster-lite

This distribution was tested at 2020-03-16 and we could never make the keyboard work on the Zero W that we used. Again this may be special for the Zeros as the USB connection is a special On-The-Go (OTG) sort. After enabling uart1, as described in section 4.2.1, we managed to connect through the serial port.

4.2.7 2021-03-04-raspbios-buster-armhf-lite

The armhf probably indicates a ARM architecture with hard built in floating point processor. After writing the uSD-card we connected a HDMI-monitor, USB-keyboard and mouse (we are naturally using the raspberry keyboard and mouse), the Ethernet connecor, and finally power. The screen came up in text only mode (despite using the wrong HDMI connector on the PI 4). We logged in as pi with passowrd raspberry and set the password **SimpleRootPass** for root with: **sudo passwd root**. Logged out and then in again as root when we in the same matter chnaged the password for user pi to **SimplePiPass**. Then we went on the edit the **/etc/ssh/sshd_config** to allow root login through ssh and finally enabled and started sshd with:

```
systemctl enable ssh.service
systemctl start ssh.service
```

Now additional configuration may be done through ssh.

We continued with:

```
apt update
apt upgrade
# reboot

rpi-update
Here it seems like a power on/off was needed

apt install rpi-eeprom
Which told us: already newest version
```

Also one should make sure the camera is enabled by using the raspi-config. After still another reboot we compiled and ran our own mjpeg-streamer.

4.2.8 2021-05-07-raspbios-buster-armhf-lite

For configuration tasks this is same as 2021-03-04-raspbios-buster-armhf-lite

4.3 Installing OS on NanoPIs

The operating systems for the NaniPI series of machines may be downloaded from the Friendlyelec's download page. Follow the instructions given in section 4.1. After the copying it is possible to mount the cards second partition (/dev/sdb2) on /mnt and do some modifications. Remember that

for finding the say /etc directory this will now be <mount point>/etc rather that plain /etc. When the <mount point> is /mnt the /etc will be /mnt/etc.

When the download is the eflasher the procedure is about the same, The system starts as usual, but you should go on to flash the eMMC with the images that are part of the eMMC-flasher image. When logged in as root simply give the command: # eflasher. The eflasher will present a choice of install images to copy to the eMMC. Make your choice and wait for the eMMC to be written to. Then halt the machine and remove the μ-SD card. When power is applied to the NanoPI it will start from the eMMC. It seems to us, through experiments, that the machine will prefer μ-SD card as boot device - when one is present.

4.3.1   nanopi-m1-plus_eflasher_3.4.39_20171102

The image used for observation station with the NanoPI M1 Plus and CAM500B camera.

4.3.2   nanopi-m1-plus_eflasher_4.14.0_20180124

is used for installing the nanopi debian jessie 4.14.0 20180124 to the eMMC of the M1 Plus. After this the "uname -a" reports: "Linux FriendlyELEC 4.14.0 #25 SMP Tue Jan 9 17:15:59 CST 2018 armv7l GNU/Linux".

When inspecting the /etc/rc.local we discovered an iptables command. We could not explain the function so we commented out that line and did a reboot. There were no ill effects. We also disabled a number of services we believe we do not need, viz. bluetooth, isc-dhcp-server, tightvncserver, gpsd. Each of these were disabled with "update-rc.d service disable" where service is one of the listed daemons. The tightvncserver did not disable, the format was wrong.

4.3.3   nanopi-duo2_sd_friendlycore-xenial_4.14_armhf_20191219

This image is for NanoPi Duo2. We have been using this with both Video for Linux (V4L2) and communication through the I2C bus. It seems to work flawlessly.

4.3.4   s5p4418-lubuntu-desktop-xenial-4.4

is the image used for the Fire2A based viewstation. We used the version dated 20180210 and built for armhf. Later we tried out a the 20180615 and 2018-09-06. For the two earlier ones there seems to be problems with the

screensaver. Better disable this from the preferences -> power managment
-> display and set everything to never.

4.4   Setting passwords for root, pi, fa, and ...

Freshly installed the user root may not always be accessible and superuser
privileges must be obtained through the "sudo" command. In the case that
no password is set for user root login is prohibited. One may change this
by setting a password for root when logged in as pi, or fa with the sudo
command: "sudo passwd root" On the NanoPIs there may be a user fa with
password fa, and/or a user pi with password pi. On the Raspbian there
should be a user pi with password raspberry. These users and passwords are
well known and when computers are mounted outside the building or WiFi
is enabled information may be stolen by no-gooders. In the unlikely case the
passwords is compromized they should not be the same as used on computers
inside the building. Also the Raspbian complains when the password for pi
has not been changed when ssh is enabled. As a very simple set of new
passwords one may use SimplePiPass for the user pi and SimpleFaPass for
the user fa. The user root may be assigned the password SimpleRootPass.
Each installation should come up with better passwords.

5   Configuring the OS and Network interfaces

When terminal access is arranged and software has been installed one must
do quite a bit of configuration to get the machine to integrate as a part of
a surveillance system. Here we go through some of the tasks to perform.

5.1   Set the hostname

One should set the host name in the file /etc/hostname. This may also
be done from the rpi-config on Raspbian or the npi-config on most of the
NanoPIs. The name should agree with the name used on the network and
possibly with the name used in an export of filesystems from a fileserver.

5.2   Static IP-address, but consider using DHCP

First consider wther this is the best solution. One may obtain the same
by using DHCP and arrange for the DHCP server to lease out specific IP-
addresses to specific MAC addresses. On our own systems we decided to go
for static IP-addresses and fill the same informastion into the hosts file on
the local Domain Name Server (DNS).

If one decide to set a fixed IP-address on the Ethernet connection one may edit the /etc/network/interfaces (Remember /mnt if this is done on the download machine) and insert the:

```
# Set fixed IP address on Ethernet
#
auto eth0
iface eth0 inet static
        address 192.168.1.50
        netmask 255.255.255.0
        gateway 192.168.1.1
```

OBSERVE that the Ethernet interface may be given another name, in which case one need to start up the actual machine and have a look. Naturally the adress and the gateway must be adjusted to correct values. Then leave the directory and "unmount /mnt". Remove the card from the reader/writer and insert it in the NanoPI. Then power up the machine.

When the NetworkManager is running it will update the /etc/resolv.conf to reflect the nameserver obtained from a DHCP server. When the interface is static we need to adjust the resolv.conf by hand and at the same time disallow the NetworkManager to update this file. First one should insert "dns=none" in the main section of /etc/NetworkManager/Network-Manager.conf.

```
[main]
plugins=ifupdown,keyfile
dns=none
```

Following that one should "rm /etc/resolv.conf" and then: "echo nameserver 192.168.1.2 > /etc/resolv.conf". Where the IP-address must match your nameserver. We need to first delete the resolv.conf as, when under control of the NetworkManager, this may be a link.

5.3   WiFi connection and WiFi access point

Connecting to the WiFi network is more complicated than simply connecting an Ethenet cable. This is a security feature as the WiFi is accessible to all receivers within its range and one should restrict the access with private keys and encryption.

5.3.1 Check rfkill for blocked device

Before anything can be done with the WiFi one must make sure the device is not blocked by **rfkill**. Use the commend **list** to see the status of radio devices:

```
root@raspberrypi:~# rfkill list
0: phy0: Wireless LAN
        Soft blocked: yes
        Hard blocked: no
1: hci0: Bluetooth
        Soft blocked: no
        Hard blocked: no
root@raspberrypi:~#
```

In this case the Wireless LAN was soft blocked and need to be unblocked with: **rfkill unblock 0**.

5.3.2 Using the nmcli to connect to an access point

First make sure the wlan0 is not mentioned in the /etc/network/interfaces as the NetworkManager will not touch any interface set up through interfaces. Also eventual changes to the /etc/NetworkManager/NetworkManager.conf should be reverted. Especially dns=none should be set to the original dns=dnsmasq as we most likely will use DHCP when connecting WiFi. Remember that the NetworkManager now will control the resolv.conf. Then to find and connect to wireless networks first, as root, find, start, scan, and conncet the wlan0 interface:

```
# nmcli device
DEVICE    TYPE      STATE        CONNECTION
wlan0     wifi      unavailable  --
eth0      ethernet  unmanaged    --
ip6tnl0   ip6tnl    unmanaged    --
gre0      iptunnel  unmanaged    --
sit0      iptunnel  unmanaged    --
lo        loopback  unmanaged    --
tunl0     unknown   unmanaged    --

# nmcli radio wifi on

# nmcli device wifi list
*   SSID                   MODE    SECURITY
```

```
    Maltwhisky2.4         Infra  WPA1 WPA2
    4G-Mobile-WiFi-E25C   Infra  WPA2
    DIRECT-pnC460 Series  Infra  WPA2
    NETGEAR_VISS-Gjest    Infra  WPA1 WPA2
    linksys               Infra

# nmcli device wifi connect MyWiFiNetwork \
    password VerySecretPassword
Device 'wlan0' successfully activated with
    'e9d561a3-1d65-4e24-b00e-fd0716e62c37'.
```

Columns CHAN, RATE, SIGNAL, and BARS have been removed from the list to fit the page. Commands and reply in the connect comamnd have been split over more lines for same reason. The commands can be shortened, i.e. device as dev, radio as r, ...

One more command may be useful with the NtworkManager. That is when a lot of networks has been accessed the list of known networks may grow long. One particular **nmcli connection** may be deleted from the list with:

```
nmcli connection delete id <SSID>
```

Here the <SSID> is the name of the network as shown as first column in the list.

After successful connect with the NetworkManager the settings are stored and at next boot the WiFi interface will be activated automatically. Although we must admit that we have experienced that this does not happen. Presently we have no more knownledge of this problem. WiFi seems now to be a quite reliable solution. Still, when connection is lost you have no way to log in to the machine to do diagnostics.

5.3.3   Connecting WiFi with wpa_cli

We thought the wpa_supplicant was superseeded by the NetworkManager, but when installing 2018-03-13-raspbian-stretch-lite.img on the Raspberry PI Zero W the wpa_supplicant is what I found. Here the control of the wireless connection is done through the wpa_supplicant and the wpa_cli is the command line user interface. Started without parameters the wpa_cli enters interactive mode and you should select the interface, add a network, configure it, connect, and if satisfied save the configuration. These steps are shown below.

```
# wpa_cli
```

```
> interface wlan0
Connected to interface 'wlan0.
> scan
OK
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-RESULTS
<3>WPS-AP-AVAILABLE
> scan_results
bssid /  / ssid
c8:be:19:5d:2a:19      Maltwhisky2.4
08:62:66:8c:a6:58      Kiara
28:28:5d:05:9a:e7      Telenor2754lag
fa:8f:ca:34:e8:15
40:a5:ef:96:be:42      ASUS
> add_network
0
> set_network 0 ssid "Maltwhisky2.4"
OK
> set_network 0 psk "VerySecretPassword"
OK
>
> enable_network 0
OK
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-RESULTS
<3>WPS-AP-AVAILABLE
<3>Trying to associate with c8:be:19:5d:2a:19 \
    (SSID='Maltwhisky2.4' freq=2412 MHz)
<3>Associated with c8:be:19:5d:2a:19
<3>WPA: Key negotiation completed with \
    c8:be:19:5d:2a:19 [PTK=CCMP GTK=TKIP]
<3>CTRL-EVENT-CONNECTED - Connection to \
    c8:be:19:5d:2a:19 completed [id=0 id_str=]
>
> status
bssid=c8:be:19:5d:2a:19
freq=2412
ssid=Maltwhisky2.4
id=0
mode=station
pairwise_cipher=CCMP
group_cipher=TKIP
key_mgmt=WPA2-PSK
wpa_state=COMPLETED
```

```
ip_address=192.168.1.190
p2p_device_address=ba:27:eb:85:68:fe
address=b8:27:eb:85:68:fe
uuid=d520f2c3-1708-5254-9fdf-b4e30f2f2f1b
>
> save_config
> quit
# ifconfig
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 192.168.1.190  netmask 255.255.255.0  \
    broadcast 192.168.1.255
inet6 fe80::b932:dcce:bd82:e94d  prefixlen 64  \
    scopeid 0x20<link>
ether b8:27:eb:85:68:fe  txqueuelen 1000  (Ethernet)
RX packets 28  bytes 3498 (3.4 KiB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 27  bytes 4253 (4.1 KiB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
#
# ping storage
PING storage (192.168.1.3) 56(84) bytes of data.
64 bytes from storage (192.168.1.3): \
    icmp_seq=1 ttl=64 time=16.0 ms
```

Observe that in the set_network 0 ssid and psk the "" are mandatory, we tried without and got the reply FAIL. The save_config command makes the settings persistent and the WiFi will reconnect to same network at next boot. If you are just exploring the WiFi you may choose to avoid this command. In the scan_results listing we have removed frequency, signal level, and flags to fit the page. Other lines have been broken with a \ for the same reason.

5.3.4   Editing hostapd.conf to set up an access point

We assume the hostapd is installed. The configuration files for hostapd are located in the /etc/hostapd/ directory, but one file that may need to be changed is the /etc/default/hostapd. The last file contains a line that tell hostapd where to find its configuration. You should find the line in the #DAEMON_CONF="" and change it to
DAEMON_CONF="/etc/hostapd/hostapd.conf"
Remember to remove the comment mark #. Also insert the "iface wlan0 inet static" in the /etc/network/interfaces to tell the NetworkManager not to manage the wlan0 interface. The method is discussed in section 5.2. For this to take effect you may do "/etc/init.d/network-manager restart"

and check which interfaces that are managed by the NetworkManager with
"nmcli device" that should mark the wlan0 as unmanaged. The rest of the
configuration is in the /etc/hostapd/hostapd.conf and the current configu-
ration looks like:

```
interface=wlan0
# bridge=brdg0
driver=nl80211

hw_mode=g
channel=1
ssid=SalmoNet

# possible MAC address restriction
#macaddr_acl=0
#accept_mac_file=/etc/hostapd.accept
#deny_mac_file=/etc/hostapd.deny
#ieee8021x=1    # Use 802.1X authentication

# encryption
auth_algs=1
wpa=2
wpa_passphrase=QueenSonja
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ctrl_interface=/var/run/hostapd

# Only root can configure hostapd
ctrl_interface_group=0
```

Here the interface is the one we starts hostapd on. The interfacese will be
listed with the "ifconfig -a" The "bridge=brdg0" should only be used with
bridging and makes hostapd add its interface to the bridge brdg0. The driver
will usually be nl80211, but for the Adafruit hostapd and WiFi dongle it
was rtl871xdrv. The wpa_passphrase is a string of ASCII caracters of length
from 8 to 63 characters. This will be used to generate the 64 hexadecimal
digits for the real key.

For testing you may use:
```
systemctl { start | stop | restart } hostapd.service
```

We had problems during testing the configuration. When the hostapd.conf
was written and we did the "/etc/init.d/hostapd restart" the hostapd did
not run. There seems to be some kind of delay before it can start again.
Check wether the hostapd run with "ps -el | grep hostapd".

Furthermore we could not make hostapd and bridging work together on the NanoPI NEO2 with FriendlyCore Xenial 4.14.0. We tried both the Element14 WiPI as well as the DWL-140. We could never add the WiFi interface to the bridge.

In the end we gave up the NEO2 and put in the M1 Plus. This is a better SBC and even has built in WiFi. An external antenna may be connected for improved reception.

5.4 Bridging Ethernet and WiFi

A bridge has the same functionality as a switch, but is usually implemented on a general computer. It is used for bridging a limited number of physical interfaces. Remember the "bridge=brdg0" in the hostapd.conf. Possibly, but we er are not sure, the bridge-utils need to be installed. We managed to get a bridge set up with the following /etc/network/interfaces:

```
auto lo
iface lo inet loopback

iface eth0 inet manual
iface wlan0 inet manual

auto brdg0
iface brdg0 inet static
        bridge_ports eth0
        address 192.168.1.16
        netmask 255.255.255.0
        gateway 192.168.1.1
```

This is the first attempt so may change, but after a reboot the "brctl show" reported that the brdg0 existed and bridged the interfaces eth0 and wlan0. The wlan0 is not bridged in the interfaces so this is a result of the setting in the hostapd.conf Naturall the IP-address and more must be changed.

5.5 Dnsmasq DHCP server configuration

The file /etc/dnsmasq.conf contains the configuration for the combined DNS- and DHCP-server dnsmasq. There are few changes to this file. Currently we only add dhcp-range at about line 158:

```
# Uncomment this to enable the integrated DHCP server,
# you need to supply the range of addresses available
```

```
#for lease and optionally a lease time.
# If you have more than one network, you will need to
# repeat this for each network
# on which you want to supply DHCP service.
#dhcp-range=192.168.0.50,192.168.0.150,12h
# Inserted dhcp-range 2017-03-24
dhcp-range=192.168.42.50,192.168.42.150,12h


# Assign specific IP-address to a MAC-address
dhcp-host=b0:f1:ec:2d:b8:20,192.168.1.23
dhcp-host=d4:7b:b0:7a:24:ef,e4:6f:13:f3:b8:74,192.168.1.15
```

The dhcp-range reserves a range of addresses to be leased to clients that reqire setup. We believe the range should be within the network used for the wlan0 interface. Below is another way than static addresses to assign known IP-addresses to specific clients. The dhcp-host look at MAC-address of the client and deals out one particular IP-address. The second dhcp-host assign the same IP-address to two different MAC-addresses. A possible solution when a client has both WiFi and Ethernet and they will never be used at the same time.

The dnsmasq writes its current leases to the file /var/lib/misc/dnsmaq.leases. To find out what machines have a leased IP-address and what it is log in to server and "cat /var/lib/misc/dnsmasq.leases". One could use the hardware address to to write a specific lease into the dnsmasq.conf.

On the NanoPI NEO2 with FriendlyCore-Xenial_4.14.0_20171208 we could not get the dnsmasq to start at boot. In the end we resorted to putting the start command in the end of /etc/rc.local. Like this: "/etc/init.d/dnsmasq restart". That actuall works, but we do not know why. Later we abandoned the NEO2 altogether and uses the M1 Plus.

5.6 Allowing user access to device files

The ic2 device files and the serial ports are owned by root and placed in the groups i2c and dialout respectively. To be able to access say the /dev/i2c-0 device file the user need to be in the i2c group (or be root). To add an existing user to an existing group one uses the adduser administrator command. For adding the user bhu to the group i2c do (as root): **adduser bhu i2c**.

5.7 Allow root login through Secure Shell

The Secure Shell (SSH) allows for a terminal window access to a networked computer. It also is the basis for other services as the Secure Copy (SCP)

command. If the SSH is not enabled at all one should du the "systemctl enable ssh.service" command. And follow up with starting the server with "systemctl start ssh.service". Logging in as root throuh SSH is by default not allowed. If one want to allow this just edit the "/etc/ssh/sshd_config" with "sudo vi /etc/ssh/sshd_config". If you are not familiar with vi use your favourite editor. In the file find the line "#PermitRootLogin prohibit-password" and replace it or add a new line "PermitRootLogin yes". Restart the server with "systemctl restart ssh.service".

6 Machines used for development

For amachine that is intended for software develooment we may want to access our fileserver as a trusted user.

6.1 Creating a user for development

The user bhu should be created on machines used for development. This user will then have access to the main fileserver bhu area. THis may be done with the command:

```
useradd --uid 1010 --gid users \
--groups adm,dialout,cdrom,sudo,audio,video,plugdev,games,\
users,input,netdev,gpio,i2c,spi \
--shell /bin/bash --create-home bhu
```

The password is not set so use the "passwd bhu" command to set it.

6.2 Importing directories through the Network File System

To allow access for the machine at all one must edit the /etc/exports on the fileserver and do a "exportfs -r" to rexport all filesystems. On the machine that shall import the dirctories the mount points must be created, e.g. "mkdir /global/storage/home" and a line for mounting the remote directory on this mount point must be added into /etc/fstab. A possible such line could be: *"storage:/home /global/storage/home nfs defaults,nfsvers=3,noauto 0 0"* Here the fileserver only supports version 3 of the NFS and that must be made clear to the mount command. To allow for this one may have to install nfs-common with:

The machine exporting the filesystem is storage, while the importing machine is the one where the fstab recides.

7 Installing software on Debian and Ubuntu

is done through the "apt command" that is a system for keeping track of what "packages" are installed, and what version they are. Information about what servers to connect to and what versions to request are written into files in the directory /etc/apt. Specially the file /etc/apt/sources.list is important. Before the apt-get one must have a working connection to the Internet. Observe: We have had problems with the apt is not able to acquire a lock on the package system. This is due to "unattende-upgrade" running wild.

To make the repository list up-to-date one must first use the "apt update" and normally this will be followed with the "apt upgrade" that brings the operating system and the installed package up to the lates version available. The upgrade may take half an hour and at least at one point requires user interaction. Generally each session of apt should start with an update. Observe that the upgrade take quite some time as there are usuallty many software packages in the distribution. By the way, when in doubt, there is no harm done in trying to install a package that is already the latest version. This will be reported and no unnecessary transfer will be done.

7.1 System software

During configuration and fault finding we need programs for finding out what is going on and how to change it. Lacking a better name we call this system software and it will be software for inspecting network interfaces, bridging network interfaces, setting up packet forewarding and Network Address Translation (NAT), listing hardware, mounting remote file systems, and more.

```
apt update
apt install net-tools
apt install usbutils
apt install bridge-utils
apt install iptables
apt install nfs-common
```

7.2 WiFi accesss point and server software

For using the SBC as a WiFi acces point we also require some sort of software and that is the hostapd to handle the complicated protocol of accessing a WiFi network. For serving the Distributed Host Configuration Protocol (DHCP) on the WiFi we also need a DHCP server and a Domain Name

Server (DNS). We have had good experience with the dnsmasq. Install both hostapd and dnsmasq with:

```
apt update
apt install hostapd
apt install dnsmasq
```

7.3   Compilers and build tools

Later on (section 10) we shall build the mjpg-streamer from sources. This requires some tools and libraries, viz. build-essential, libv4l-dev, and libjpeg-dev. The should all be installed with the apt tool:

```
apt update
apt install build-essential
apt install cmake
apt install libv4l-dev
apt install libjpeg-dev
apt install libopencv-dev
apt install libssl-dev
apt install libi2c-dev
```

Of these the build-essential installs everything needed for conmpiling and linking programs. The cmake is a build configuration tool that is commonly used for building software across different OSs. The libv4l-dev and libjpeg-dev are used by mjpg-streamer. libopencv-dev is used by sveral X_X_Viewers in the Video_Surveillance. The libssl is used by the mechanisms for Web-sockets that is used in the code for sending data to Javascript displays. The libi2c-dev is needed to build programs that access the Inter-Integrated Circuit (IIC or I^{2}C or I2C) bus.

For creating documentation from the tex files you will need the

```
apt update
apt install texlive-full
```

but this package fetches nearly two Gigabytes of data so one will probably only install this on the main computer.

8   V4L2 performance

We have developed a test program for measuring the raw speed (framerate) of video capture devices through the Video for Linux (V4L) framework.

The same program also has the possibility to do a copy of the buffer to local memory, a conversion from the YUYV to RGB888, and writing the image to a file in JPEG format. The program we use is placed in the src.xpl/V4L directory and is named v4l2-speed.cpp. It includes the videodev2 and the central part is a loop that dequeues and queues up buffers i the V4L driver. Fetching a buffer from the q is done with

```
if( ioctl(fd, VIDIOC_DQBUF, &buffers[i].info) != 0) {
    perror("VIDIOC_DQBUF");
    exit(1);
}
```

When this system call returns we do nothing with the buffer and puts it back into the q with:

```
if( ioctl(fd, VIDIOC_QBUF, &buffers[i].info) < 0){
    perror("VIDIOC_QBUF");
    exit(1);
}
```

In both of these calls the buffer[i].info is a

```
struct buffer_t {
    struct v4l2_buffer info;
    void* start;
};
```

Where again the **struct v4l2_buffer info** is declared in videodev2 and start is set as the start of the buffer that is mmapped into user space.

8.1   Raspberry PI Zero W

The Pi model Zero W is a very small SBC computer with a single-core CPU running at 1 Ghz and sporting 512 MB of RAM. The testing was done on the 2019-09-26-raspbian-buster-lite.img with the Raspberry PI Camera. The resulting printout on the console is:

```
$ time ./v4l2-speed
Requested width, height: 640, 480, Obtained w, h: 640, 480
VIDIOC_S_PARM for setting frame rate is NOT supported
old bufrequest.count = 5, new bufrequest.count = 5
 1000 frames captured
real    0m33.809s
user    0m0.028s
sys     0m0.203s
```

1000 frames in 33.8 s computes to 29.6 frames/s. Allowing for some time to start the program we can assume 30 frames/s.

8.2   NanoPi Duo2 with OV5640 camera

The duo is a rather small SBC from FriendlyElec. It has an Allwinner H3 processor with Quad-core Cortex-A7 Up to 1.2GHz and 512 MB DDR3 RAM. Testing was done on nanopi-duo2_sd_friendlycore-xenial_4.14_armhf_20191219.img. The results from timing one run of v2l2-jpeg is:

```
Requested width, height: 1280, 720, Obtained w, h: 1280, 720
use_width*use_height*2*sizeof(unsigned char): 1843200
old bufrequest.count = 3, new bufrequest.count = 3
Starting to capture and process the requested 1000 images
Captured and processed 1000 images
real    0m34.533s
user    0m0.018s
sys     0m0.064s
```

This computes to a frame rate of 29.0 frames/second. In our next test we had the Duo2 copy each frame into a local buffer. We used the memcpy library function for the copy. The buffer was preallocated and the same buffer was overwritte for each frame. In this test the framerate ended up as 19.4 frames/second.

9   libJPEG performance

For storage or transmission of images one may consider converting the raw images (usually YUYV pixel format) into JPEG. The libJPEG will be used for such conversions. As a start we checked out the obtainable framerate for two different frame sizes while running on the NanoPi Duo2. We have found several examples that may be a starting point for our own conversion code. The speeds we obtain for a 640x480 image is about 33 frames/second and for 1280x720 we obtain about 11 frames/second. It may be possible to run the conversion on more processor core to obtain higher framerates.

10   Install, compile, and run the mjpg-streamer

The mjpg-streamer is our workhorse for getting images and video from the camera and onto the network. The mjpg-streamer acts as a Hypertext server with its own web pages and will reply with single images or streams of Motion

JPEG (MJPEG). The mjpg-streamer is not a precompiled package, rather one need to install the sources and build the program. During the build process the libv4ldev, and libjpeg-dev will have to be available. Naturally also the usual C++ compiler and tools.

10.1 Installing the sources

On the NanoPI the sources are installed in the home dictory of root, viz. /root/mjpg-streamer, but we may prefer another user. For the Raspberry PI the situation is more difficult as the mjpg-streamer at sourceforge seems to be inactive [Mjpg-streamer]. Still there is an alternative in the github repositori [Mjpg-streamer]. We tried this once, but the build did not complete. What we did was to make a tar file from the sources installed by default on the NanoPIs. To copy a file from one machine to another the most convenient way is probably to use the scp. To copy file.tar into user pi's home directory at host use: **scp file.tar pi@host:file.tar**.

10.2 Compiling the mjpg-streamer

10.2.1 NanoPI

Log in as root and find the mjpg-streamer directory. Simply enter the directory "cd mjpg-streamer" and do "make clean" for good measure, then a make. There are a lot of output and even some warnings. The build may be tested at once with ./start.sh. This file contains a lot of examples on how to start the streamer.

10.2.2 Raspberry PI

Use the tar file mjpg-streamer-nanopi.tar that contains the mjpg-streamer directory from the NanoPI - after a "make clean". Copy this file into the home directory of the user you prefer and do a: "tar -xvf mjpg-streamer-nanopi.tar" and after that "cd mjpg-streamer". do "make clean" and "make'. On the 2018-03-13 version of raspbian the compile completes with a lot of output and even a warning about unused variables. This version is the only one that we have have been able to get going, but the 2018-04-18 has not been tested. We do need to install the jpeglib.h from package libjpeg-dev. On the 2019-04-08-raspbian-stretch-lite the NanoPI version of mjpeg-streamer compiled without warnings, but the image was garbled. On the 2019-07-10 and 2021-03-04-raspios-buster-armhf-lite the compile fails due to redefinition of struct statex_timestamp included from utils.c line 32. I removed this line to make it compile, but as noted in the subsection on

operating systems the mjpg-streamer still did not run. Also we needed to install libjpeg-dev to complete the compile with a lot of warnings. The 720x480 image was garbled, but 1280x702 at 5 frames/s was OK.

10.3 Starting mjpg-streamer

The mjpg-streamer need the module bcm2835-v4l2 and this module needs to be loaded if it is not built into the kernel. On later versions of the operating system this module is present as default. One the earlier Raspberry PIs one needed to do a "modprobe bcm2835-v4l2". Assuming we are in the home directory where the mjpg-streamer has been built the commands for starting the streamer are:

```
./mjpg_streamer \
-i "./input_uvc.so -y 1 -r 1280x720 -f 5 -q 90 -n" \
-o "./output_http.so -w ./www"
MJPG Streamer Version: svn rev:
 i: Using V4L2 device.: /dev/video0
 i: Desired Resolution: 1280 x 720
 i: Frames Per Second.: 5
 i: Format............: YUV
 i: JPEG Quality......: 90
 o: www-folder-path...: ./www/
 o: HTTP TCP port.....: 8080
 o: username:password.: disabled
 o: commands..........: enabled
```

This is the output when the mjpg-streamer is started from the terminal. And again we needed to break a few lines to fit the page This works both for the console and for a SSH login. There has been some changes and now the "-y 1" option seems to be important. The "-n" option means "do not initalize dynctrls of Linux-UVC driver" and removes a lot of "UVCIOC_CTRL_ADD - Error" messages from the output. The "-q 90" has to do with JPEG quality. One could consider changing the resolution "-r" and/or frame rate "-f". FriendlyElec has put several examples of how to start the mjpg-streamer into the file start.sh. Edit this file and fill in an uncommented line that contains something like the line shown above. To start the mjpg streamer simply do: "./start.sh" in the mjpg-streamer directory. When you want to shut down the stream simply hit CNTRL-C.

10.4 Testing the mjpg-streamer, and how to stop it

Go back to you machine with a display and keyboard, start the web-browser, and fill in the "http://camera-host:8080" in the address field. Here the

camera-host is the name of the machine running the mjpg-streamer. If You do not have any nameserver on the network, You should use the IP-address. The mjpg-streamer page should come up and it should be possible to navigate to the static or streaming pages.

When you want to be able to log out the terminal and still have the mjpg-streamer running you should use the nohup command that intercepts the HangUP (HUP) signal when starting another program. The command line "nohup ./start.sh > mjpg.log &" will disable hangup, redirect output to mjpg.log, and arrange for the command to run i the background. If You do not want to keep any log redirect into /dev/null. The start.sh is in the mjpg-streamer directory of user root on the NanoPI. In this case you cannot stop the program with CNTRL-C. Instead find the process ID from the ps command and use the "kill -term PID".

10.5   Starting mjpg-streamer at boot

may be done by commands placed in the /etc/rc.local. Commands in this file are for use by the system administrator and is traditionally executed after all the normal system services are started. If the bcm2835-v4l2 is not built into the kernel it must be loaded first. Insert the command "modprobe bcm2835-v4l2" in the rc.local before the commands used to start the streamer.

Remember the streamer does not terminate so the line that starts it should end with an & to create a new process. Otherwise the boot sequence would halt here. As the mjpg-streamer startup expects the current directory to be the one mjpg-streamer is built in one shuold "cd /root/mjpg-streamer". Here this would be /root/mjpg-streamer. Typically the /etc/rc.local would end in

```
# Lines for running Tom Stoevecken mjpg_streamer
#
# For older Raspbian versions load the module for v4l2
# At about raspios this seems to be loaded by default
# modprobe bcm2835-v4l2

# Enter mjpg build directoy at user pi or elsewhere
cd /home/pi/mjpg-streamer

# Start the streamer - REMEMBER & at end or rc.local
# will not return and hang up system initialization
#
./mjpg_streamer \
-i "./input_uvc.so -y 1 -r 1280x720 -f 5 -q 90 -n" \
```

```
-o "./output_http.so -w ./www" &

# The very end of rc.local that must return 0
exit 0
```

For the later verions of sapbian and raspios the bcm2835-v4l2 seems to be
loaded as default. Again the \ has been used to break a long line. Among the
thing one may change are the framerate "-f 5", the resolution "-o 1280x720".

11   The Homebrew mjpeg_streamer

As we saw the original mjpg-streamer producing more and more warnings
during compile we decided to create a mjpeg_streamer of our own. This
will be considerably simpler than the one designed by Tom Stoevecken
[Mjpg-streamer], but we have the opertunity to learne more about the V4L2
driver.

11.1   Developing for the V4L2 on Linux

The V4L2 is not simple to use, but regarding flexibility and the complex-
ity of the actions performed by the driver that is expected. As sources
of information we have looked deeply into the source code for the Mjpg-
streamer originally by Tom Stoevecken [Mjpg-streamer]. The sources we
have available is currently the version provided in the /root directory of
Ubuntu on images for the NanoPi SBCs. Further information is found in
the kernel sources for Linux [Linux kernel]. Here we look into "The Linux
kernel user-space API guide" and further to "Linux Media Infrastructure
userspace API" that finally contains "Part I - Video for Linux API". Even
this is quite difficult to navigate, but again this is as expected.

11.2   URLs with special meaning for the streamers

The normal command for starting streaming from the mjpeg_streamer as it is
entered in the address field of the web-browser will be: **http://host:8080/-
stream**. Here the host is the hostname or IP-address for the computer that
the mjpeg_streamer runs on. The port number **8080** is the default port for
the streamers, but this can be changed. With this form the speed of the
V4L2 capture part is not changed. The framerate will be as it was last set.
Setting the framerate may be done before starting mjpeg_streamer with the
command **v4l2-cnt –set-parm 20** that in this case sets the framerate to 20
frames/s. The currently set framerate may be obtained with the command
v4l2-ctl –get-parm that prints the framerate and more.

We have also implemented special URLs of the form **http://host:8080/-slow** where the address may end with **slow**, **med**, or **fast** to indicate a requested framerate. Currently the rates corresponding to slow, med, fast are 3, 10, 30 respectively. There is also a special link for getting one single image: **http://host:8080/single**.

Currently we also implements the target **info** that is intended to give information about the running mjpeg_streamer. This is very much a work in progress.

11.3   Starting the homebrew at boot

We use the same method as with the Tom Stoevecken mjpg-stremer and for the time beeing put the commands into the **/etc/rc.local**. Remember that this file must have excute permission to be run at all. The few lines to start the mjpeg_streamer is:

```
# =====  Lines for running the homebrew mjpeg_streamer
#
cd /home/pi/Projects/Video_Surveillance/V4L2_Streamers
./mjpeg_streamer > /dev/null 2>&1 &
```

This will change to the build directory of the mjpeg_streamer. Start the mjpeg_streamer in background and at the same time redirect both standard output and standard error to **/dev/null**.

References

[FriendlyElec]   FriendlyElec, FriendlyArm
                 www.friendlyarm.com

[FTDI]   Future Technology Devices International Ltd.
                 http://www.ftdichip.com

[Lady Ada]   https://learn.adafruit.com/ adafruits-raspberry-pi -lesson-5-using-a-console-cable/ software-installation-mac

[Linux kernel]   https://www.kernel.org/doc/html/latest/

[Mjpg-streamer] The mjpg-streamer team
                 Design Andreas Wiklund
                 mjpg-streamer.sf.net

[Mjpg-streamer] Mjpg-streamer (sourceforge)
Tom Stoeveken
WARNING: ABANDONED as 2018-02-18
https://sourceforge.net/projects/mjpg-streamer/

[Mjpg-streamer] Mjpg-streamer (Github)
Jacksonliam
https://github.com/jacksonliam/mjpg-streamer

[OpenCV] Open Computer Vision
www.opencv.org

[PIMORONI] PIMORONI
Tech Treasure for Tinkerers
https://shop.pimoroni.com

[Prolific] Prolific
http://www.prolific.com.tw/

[PuTTY] PuTTY
www.putty.org

[Raspberry PI] Raspberry PI foundation
www.raspberrypi.org

[ST] STMicroelectonics
http://www.st.com