**Build Log**

**Streaming Video from Raspicam to web using UV4L Driver**

**Step 1: Material**

- Raspberry Pi or Odroid-W Running Raspbian

- Raspicam

- Router

- Ethernet Cable or wireless usb card

- Windows Computer

- USB Keyboard and Mouse

- HDMI Monitor

**Step 2: Install Raspbian**

Download NOOBS on your computer and Extract to a Folder (<https://www.raspberrypi.org/downloads/noobs/>)

Format SD Card. On Windows, use SD Formatter (with format size adjustment “on”). On linux, use gparted (create one single **Fat32** partition) (<https://www.sdcard.org/downloads/formatter_4/index.html>)

Copy all folders and files in NOOBS to SD Card

Install SD Card in Raspberry Pi

Connect Raspberry Pi to a monitor, keyboard, and mouse and power it up. Change Language to English (US). Install Raspbian.

**Step 3: Set Configuration**

Connect to Internet via WiFi or LAN Cable

Open the terminal and run:

$ sudo raspi-config

Under Boot Options, Select Console Autologin

Under Localisation Options, Select Locale, Timezone, and Wi-Fi Country

Under Enable Camera, Select “Enable”

Under Advanced Options, Select Memory Split and “256”

Under Advanced Options, Select SSH and “Enable”

Under Advanced Options, Select “Update”

Reboot Raspbian

Open the terminal and update the GPU Firmware:

$ sudo rpi-update

**Step 4: Installing Raspicam Driver**

To install the uv4l driver, open the terminal and run the following commands:

$ wget http://www.linux-projects.org/listing/uv4l\_repo/lrkey.asc && sudo apt-key add ./lrkey.asc

Add the following line to the file /etc/apt/sources.list :

$ sudo nano /etc/apt/sources.list

deb http://www.linux-projects.org/listing/uv4l\_repo/raspbian/ jessie main

Or: deb http://www.linux-projects.org/listing/uv4l\_repo/raspbian/stretch stretch main

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo apt-get install uv4l uv4l-raspicam uv4l-raspicam-extras uv4l-server uv4l-uvc uv4l-xscreen uv4l-mjpegstream

$ sudo reboot

$ sudo apt-get autoremove ***#(to cleanup after reboot)***

Source: <http://www.linux-projects.org/uv4l/installation/>

$ sudo ifconfig  
And note the IP address of the device (eth0 if cabled or wlan0 if wireless)  
Use this IP address with browser and ssh client (putty).

$ sudo shutdown now  
Shut down the device

**Step 5: Connect the Camera**

Make sure the device is powered off.

Gently lift the plastic lid on the connector behind the Ethernet port. **Do not use the other connector** on the edge (that is used for the LCD and will fry the camera).

Plug the ribbon cable in with the stripped side aligned with the pins on the connector (see photo). Make sure it fits smoothly.

Gently push the lid back on and power the device back on.



**Step 6: Start the streaming server**

Note the IP address from step 4. You will need it for the browser and ssh client.

The server should start automatically on port 9090. Verify by using browser (http://<raspberry IP address>:9090). You should see the UV4L Streaming Server panel, or (http://<raspberry IP address>:9090/stream) to start streaming.

By default it uses mjpeg. To modify the settings, use the “configuration” option in the browser or edit the config file directly on the PI and then restart the service:

$ sudo nano /etc/uv4l/uv4l-raspicam.conf

$ sudo service uv4l\_raspicam restart

Alternatively, you can run the server manually with different options. ssh into the PI, kill the existing process and start a new one.

$ sudo pkill uv4l

$ sudo uv4l -nopreview --sched-rr --auto-video\_nr --driver raspicam --encoding mjpeg --width 1280 --height 720 --shutter-speed 50 --framerate 30 --server-option '--port=9090' --server-option '--max-queued-connections=30' --server-option '--max-streams=25' --server-option '--max-threads=29'

As an alternative to manually setting the real time scheduling priority to the uv4l process each time, you can run the driver with the --sched-rr option, which requires root privileges by default:

Notes:

For ssh, the default login is ***pi*** / ***raspberry***

The --port=9090 is the local IP port. You can use any port you like.

The --max-streams=25 is the maximum simultaneous streams.

**Step 7: Configure your router**

- Open a Web browser

- Connect to Router (password Kashmir4)

- Type your router IP address (<http://192.168.1.1/>).

- Router login. If you're using Linksys router, the user and pass could be "admin" and “password”

- Port forward your Raspberry Pi IP address.

If you are using Linksys router:

- Click "Applications & Gaming" Tab

- Click "Simple Port Forwarding" Tab

- Configure the following parameters:

External Port: 9090 (Public IP port to reach your Raspberry stream)

\*Make sure your chosen port is Open with <http://mxtoolbox.com/PortScan.aspx>. If it's not, call your internet provider to open the port.

Internal Port: 9090 (Raspberry Pi streaming port)

Protocol: Both

To IP Address: 192.168.1.104 (Your local Raspberry Pi IP Address)

Enabled: checked

Resources

How to find your router IP address: [http://portforward.com/networking/routers\_ip\_addre...](http://portforward.com/networking/routers_ip_address.htm)

Port Forwarding: [http://setuprouter.com/router/cisco/linksys-e1200/...](http://setuprouter.com/router/cisco/linksys-e1200/port-forwarding.htm)

**Step 8: Test**

To see the streaming follow the next steps:

- Open a browser

- Type your Raspberry Pi IP followed by the external/public port (http://Your Raspberry Pi IP:9090/stream)

**Step 9: Set Up for Stream on Boot**

To set up your Raspberry to Stream on boot, add the following lines to etc/rc.local

sudo nano /etc/rc.local

sudo pkill uv4l

sudo uv4l -nopreview --sched-rr --auto-video\_nr --driver raspicam --encoding mjpeg --width 1280 --height 720 --shutter-speed 50 --framerate 30 --server-option '--port=9090' --server-option '--max-queued-connections=30' --server-option '--max-streams=25' --server-option '--max-threads=29'

**Step 10: Stream/Record Over Mission Planner HUD**

Open Mission Planner

Right Click on HUD

Under Video, Select MJPEG Source

Enter the URL to the mjpeg source:

<http://10.0.0.39:9090/stream/video.mjpeg>

**Step 11: Stream/Record Over VLC Playe**r

Open VLC Player

Media

Open Network Stream

<http://10.0.0.27:9090/stream/video.mjpeg>

**Step 12: Record Video to micro SD Card**

dd if=/dev/video0 of=video.mjpeg bs=1M & pid=$! ; sleep 10; kill $pid

**Optional USB Camera**

If you are using a UTC compatible USB Camera

* Replace --driver raspicam with --driver uvc
* Determine camera make/model with *lsusb* command
* Add --device-id 05a3:9520 (for 05a3:9520 camera)
* Remove options that are not compatible with UVC
  + --width 640
  + --height 480
  + --framerate 20
  + --encoding mjpeg

**Appendix A: Diagnostics**

If you are unable to see the stream, run the command:

$ sudo raspistill -v -o test.jpg

If you see:

***mmal: Camera is not detected. Please check carefully the camera module is installed correctly***

* Make sure you enabled the camera in raspi-config
* Make sure the ribbon cable is plugged in the right way

If you see:

**mmal: No data received from sensor. Check all connections, including the Sunny one on the camera board**

* Make sure you power down the PI before performing any of the actions below and power it back up afterwards to retest.
* Make sure the ribbon cable is plugged in firmly into the connector.
* Make sure the other side is plugged in firmly into the camera board.
* Gently clean the cable end going into the PI.
* If you have an extra cable, replace it.
* If all else fails, you may have a bad camera :(

**Appendix B: Backup/Restore SD Card**

You can backup the SD card after you are done with the configuration and then flash another SD card as needed, to replace the original card. This can help save a lot of time. Power down the PI, remove the mini SD card and plug it into your PC (using SD or USB adapter).

**On Windows:**

Source: <http://raspberrypi.stackexchange.com/questions/311/how-do-i-backup-my-raspberry-pi>

**On Linux:**

In the example below, SD card is /dev/sdg - double check and modify as necessary

To Backup

$ sudo dd if=/dev/sdg of=pi\_sd\_image

To restore on a FAT32 formatted SD card:

$ sudo dd if=pi\_sd\_image of=/dev/sdg

Alternatively, you can use compression to save some space on your hard drive (will take longer):

Backup:

$ sudo dd if=/dev/sdg | gzip > pi\_sd\_image.gz

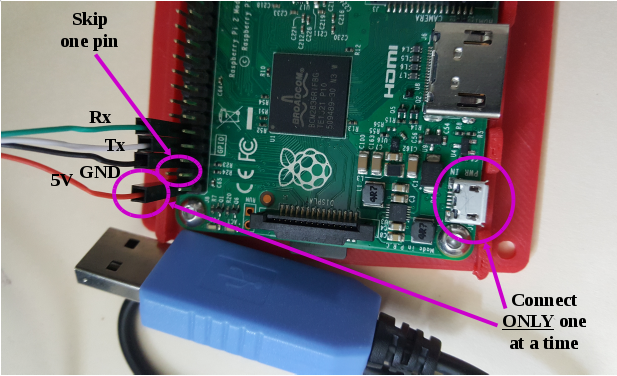
Restore:

$ sudo gzip -dc pi\_sd\_image.gz | dd of=/dev/sdg

**Appendix C: Serial Console into PI**

Using a USB to TTL cable (<https://www.adafruit.com/product/954>) can eliminate the need for a monitor/keyboard to initially configure the PI. Connect the cable as shown in picture and plug into USB on you PC.

Note: The red wire provides power to the PI. Make sure the mini USB is unplugged before connecting it. Alternatively, leave the red wire unconnected and lug in the mini USB to power the PI.



**On Windows:**

1. Download and install the drivers for the cable (<https://cdn-shop.adafruit.com/product-files/954/PL2303_Prolific_DriverInstaller_v1_12_0.zip>).
2. Download and install putty.
3. Plug the cable in.
4. Search “Devices and Printers” to find the COM port the cable is connected under (assume COM6 for this example)
5. Launch putty:
   * Under **Connection type:** click **Serial**
   * Under **Serial line** type **COM6** (or whatever port is listed in 4.)
   * Under **Speed** type **115200**
   * Optionally, save the parameters for easy access in the future
     + Under **Saved Sessions** type **PI Serial** (or another meaningful name)
     + Click **Save**
   * Click **Open**

**On Linux:**

Use putty as above or launch terminal and use:

$ screen /dev/ttyUSB0 115200