Preparing Raspberry Pi with Camera

Background: The Raspberry Pi Camera module can be used to take high-definition video as well as still photographs. It has a five megapixel fixed focus that supports 1080p30, 720p60 and VGA 90 video modes. It attaches via a 15cm CSI port on the Raspberry Pi and can be accessed through the MMAL and V4L APIs. The goal of working with the Raspberry Pi is to master the CSI port, also used by Sony Block and other high performance cameras.

Step 1: Grasp sides and lift up on CSI connector.  
Step 2: Insert Raspberry Pi ribbon cable.  
Step 3: Push down on CSI connector.  
Step 4: insert SD card with NOOBs  
Step 5: install Raspbian  
Step 6: set up language and keyboard  
Step 7: enable camera (sudo raspi-config to configure Raspberry Pi)  
Step 8: set host name (camera for the camera P, receiver from the viewing Pi)  
Step 9: set the memory for the GPU to 256  
Step 10: enable SSH  
Step 11: Username “pi”, password “raspberry”

Setting up Programs for Video Streaming

Step 1: sudo apt-get install mplayer netcat  
Step 2: cd /opt/vc/src/hello\_pi  
Step 3: make –C libs/ilclient  
Step 4: make –C libs/vgfont  
Step 5: cd /opt/vc/src/hello\_pi/hello\_video  
Step 6: make

Setting up Streaming Video from Raspberry Pi to Windows Computer

Step 1: set Raspberry Pi for static IP  
sudo nano /etc/network/interfaces (Set Raspberry Pi for static IP)  
edit eth0 section to read:  
Iface eth0 inet static  
address 192.168.1.85  
netmask 255.255.255.0  
Ctrl-X, then Y to save & exit  
sudo ifdown eth0 (to make change effective)  
sudo ifup eth0 (to make change effective  
Step 2: create streaming script  
nano /home/pi/stream.sh (streaming script)  
Write to read:  
#!/bin/bash  
raspivid –t 0 –w 300 –h 300 –fps 10 –o - -n | nc –l –p 5001 –k  
Ctrl-X, then Y to save & exit  
chmod +x stream.sh  
Step 3: test streaming script  
/home/pi/stream.sh  
Step 4: create file to start streaming on bootup  
mkdir ~/.config/autostart  
nano ~/.config/autostart/stream.desktop  
Type=Application  
Exec=/home/pi/stream.sh  
Reboot to test!  
Step 5: Set up Ground Windows Laptop  
Set static ip to 192.168.1.86 (netmask 255.255.255.0)  
(Control Panel -> View Network Status and Tasks-> click on “Local Area connection” ->Properties-> TCP/IPv4->Properties)  
Set up VLC (Tools->Preferences->Show Settings (lower left):All->Input/Codecs->Demuxers->H264->Frames per second: 10)  
In VLC,Media->Open Network Stream  
URL: tcp://[192.168.1.85:5001](http://192.168.1.85:5001/)  
Show more options  
Edit Options :network-caching=0 :demux=h264

Reducing Latency in Streaming Video from Raspberry Pi to Windows Computer

Step 1: Install Prerequisite software on Pi (requires internet connection)  
"sudo nano /etc/apt/sources.list"  
Add the following line to the end of the file  
deb <http://vontaene.de/raspbian-updates/> . main  
Ctrl+X, then [just] Y to exit  
"sudo apt-get update"  
"sudo apt-get install gstreamer1.0"  
Step 2: Set Raspberry PI for static IP  
Open Terminal, execute “sudo nano /etc/network/interfaces”  
Edit eth0 section to read:  
iface eth0 inet static  
address 192.168.1.85  
netmask 255.255.255.0  
Ctrl-X, then Y to save & exit  
If not rebooting “sudo ifdown eth0” then “sudo ifup eth0” to make change effective  
Step 3: Create streaming script  
“nano /home/pi/stream.sh”  
Write to read:  
#!/bin/bash  
raspivid –t 0 –w 300 –h 300 –fps 10 –o - -n | gst-launch-1.0 -e -vvvv fdsrc ! h264parse ! rtph264pay pt=96 config-interval=5 ! udpsink host=192.168.1.86 port=5001  
Ctrl-X, then Y to save & exit  
“chmod +x stream.sh”  
Test streaming script  
“/home/pi/stream.sh”  
Step 4: Create file to start streaming on bootup  
“mkdir ~/.config/autostart”  
“nano ~/.config/autostart/stream.desktop”  
[Desktop Entry]  
Type=Application  
Exec=/home/pi/stream.sh  
Reboot to test!

Step 5: On ground Windows laptop:  
Download GStreamer for Windows from <http://gstreamer.freedesktop.org/data/pkg/windows/1.2.4/gstreamer-1.0-x86_64-1.2.4.msi>  
Open the file and perform a "Complete" installation  
Right click on the Desktop and select New->Text Document. Name it "View Stream.cmd"  
Right click on it and select "Edit". Enter the following  
C:\gstreamer\1.0\x86\_64\bin\gst-launch-1.0 -e -v udpsrc port=5001 ! application/x-rtp, payload=96 ! rtpjitterbuffer ! rtph264depay ! avdec\_h264 ! fpsdisplaysink sync=false text-overlay=false  
Save & exit  
Set static ip to 192.168.1.86 (netmask 255.255.255.0)  
(Control Panel->View Network Status and Tasks-> click on “Local Area Connection”->Properties->TCP/IPv4->Properties)  
Viewer can be launched by double-clicking on the “View Stream” command file created in the earlier step.

Set Up Streaming Over Wi-Fi (Ad-Hoc Network/Proof-of-Concept)

Step 1: Configure Ground Laptop for Ad-Hoc Network  
Make sure wireless switch is on (right side of the laptop – move switch towards the rear)  
Click on the wi-fi icon in the system tray and click “Open Network and Sharing Center”  
Click “Set up a new connection or network”  
Click “Set up a wireless ad hoc (computer-to-computer) network”  
Enter network name (i.e. “airlink2”)  
Select security type “No authentication (open)”  
Step 2: Configure Ground Laptop for static IP  
Click on the wi-fi icon in the system tray and click “Open Network and Sharing Center”  
Click on “Wireless Network Connection (network name from step 1])”  
Select “Internet Protocol Version 4”  
Click Properties  
Enter IP address 192.168.2.86, Subnet Mask 255.255.255.0  
Click Ok  
Step 3: Configure Pi Networking  
Step 1: Plug in wi-fi stick  
Step 2: Open Terminal.  
“sudo nano /etc/network/interfaces”  
Modify the “wlan0” section to:  
allow-hotplug wlan0  
iface wlan0 inet static  
address 192.168.2.85  
netmask 255.255.255.0  
wireless-essid network name from step  
wireless-mode ad-hoc  
(Leave the line directly below the wlan0 section untouched – “iface default inet dhcp”)  
Ctrl-X, then Y to exit  
Step 4: Modify the Pi to send its stream over the wireless -link  
In Terminal,  
“nano /home/pi/stream.sh”  
Change 192.168.1.86 to 192.168.2.86  
Ctrl-X, then Y to exit  
Step 5: Reboot (easiest way to make all the wireless settings take effect)

Advanced:

* new pi
* change keyboard layout, enable camera, expand file system, change password, change host name
* Edit /etc/network/interfaces, /etc/wpa\_supplicant/wpa\_supplicant.conf, /etc/default/ifplugd, and /boot/config.txt

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install python-stdeb

sudo apt-get install python-dev

sudo easy\_install pyzmail

sudo easy\_install pymavlink

sudo easy\_install pygsm

//Fails for some reason, install with pip//

sudo apt-get install python-pip

sudo pip install git+<https://github.com/adammck/pygsm>

sudo apt-get install screen

sudo apt-get install gstreamer1.0

sudo apt-get install openvpn

copy openvpn files client.config, ca.crt, PI\_NAME.crt, PI\_NAME.key to /etc/openvpn

edit client.config to correct .crt filenames

sudo wget <https://raw.github.com/txt3rob/hamachi-pi/master/hamachi.sh>

sudo chmod 777 hamachi.sh

sudo ./hamachi.sh

sudo apt-get install tinc

sudo mkdir -p /etc/tinc/myvpn/hosts

sudo vi /etc/tinc/myvpn/tinc.conf

Name = PI\_NAME

Interface = tun0

sudo vi /etc/tinc/myvpn/hosts/PI\_NAME

Subnet = [10.0.0.10/32](http://10.0.0.10/32)

sudo tincd -n myvpn -K4096

sudo vi /etc/tinc/myvpn/tinc-up

#!/bin/sh

ifconfig $INTERFACE 10.0.0.10 netmask 255.255.255.0

sudo vi /etc/tinc/myvpn/tinc-down

#!/bin/sh

ifconfig $INTERFACE down

sudo chmod 755 /etc/tinc/myvpn/tinc-\*

sudo nano /etc/tinc/nets.boot

myvpn

* copy tinc keys to /etc/tinc/myvpn/hosts/PI\_NAME and other computers.