Live Streaming Team

CansatII Project

bibliotheca alexandrina
Documentation

Can Sat II

- **Outline**

Content Page

I. How to Connect Raspberry pi to internet with 3g modem

we tried using wvdial and numerous other ways to connect to 3G. None of them worked reliably. In the end, we turned to sakis3g - the All-In-One script for connecting 3G modem.

Put one male USB plug into the PI and the other into a power supply. The dongle fits into the female USB socket.

P-p-p-p-pick Up A PPPD

In order to get our network connected, we need to install the ppp package.

"sudo apt-get install ppp"

installation is very simple:

First, download the latest version. The Raspberry Pi runs on an ARM processor, so this is the version we download.

"wget http://raspberry-at-home.com/files/sakis3g.tar.gz"

The script is compressed. Unzip it.

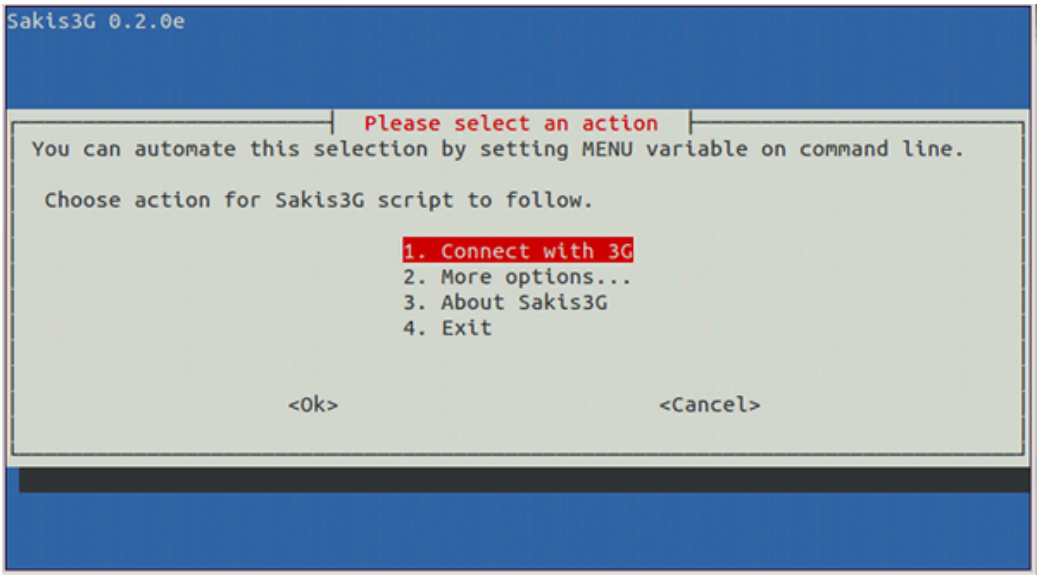
"gunzip sakis3g.gz"

Finally, we want to make the file executable so that we can run it.

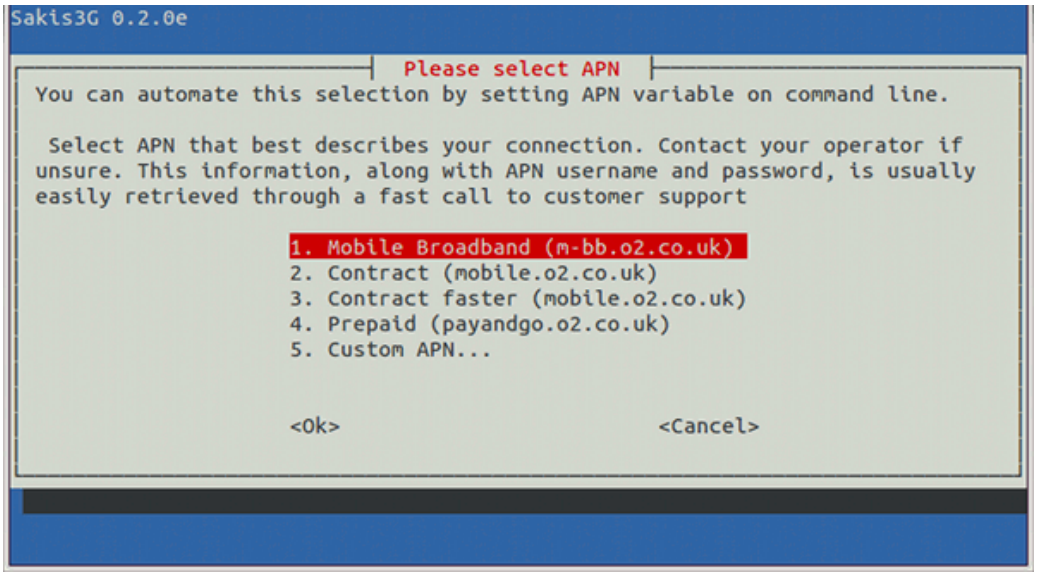
"chmod +x sakis3g"

Running sakis is quite straightforward. It has a basic GUI which will work even if you're just using the command line.

"sudo ./sakis3g –interactive"



Sakis has a fairly comprehensive list of connection details - it should find yours automatically and present you with this screen.



If you connect a Vodafone Usb modem

the APN you select is internet.Vodafone.com.eg

Saskis3g will ask you to enter Vodafone APN Username and password

just put Username : internet

Password : internet

So, that should be everything you need to get the Raspberry Pi connected over a USB 3G dongle. Have fun!

II. How to Connect a Wi-Fi dongle to Raspberry pi.

-Wifi dongle is a USB device used in order to make the Raspberry Pi device connect to a Wireless network.

-First of all, we need to make sure that the dongle's device driver module is loaded at the Raspbian Kernel, in order to do that, we need to download the proper module file from the manufacturer website of the wifi dongle, then use the following command to load it:

sudo insmod <module-name>

-After that, we need to identify an interface for our module in order to assign it an IP address and start our internet connection, to do that we need to UP the interface (make it ready to work with our wireless connection(s)) via the command:

sudo ip link set <wireless-interface> up

-Now, we can start the connection between our Wi-Fi dongle and a wireless network, we can search for available networks using the command:

sudo iwlist wlan0 scan

-Finally, we can check if the connection is established successfully or not via pinging any global IP:

ping -c 3 <WAN IP>

III. How to Connect a USB Camera to Raspberry pi.

-Linux Kernel sees any external device as a file, as usual as any file of yours you are working daily with, a USB camera is exactly the same.

-A USB camera file takes the name under the following path:

/dev/videoX

Where X is any positive number larger than or equal zero.

-As soon as you connect your USB device in the Raspberry Pi, try to list all external devices using the command:

*ls –l /dev/video**

-You should find the proper file(s) as many as your connected devices, so for example if we have 4 cameras connected to the Raspberry Pi device, we shall find the following files after connecting each camera:

/dev/video0
/dev/video1
/dev/video2
/dev/video3

IV. How to make Webcam streaming on Raspberry pi localhost using motion package.

1. Get the software ready

we are going to be using a great little application called Motion, this will do a few things for us including accessing the USB cam, getting the images, and streaming them via a built in web server. As the name suggests it will also track and trigger events on motion been detected in the video frames .

```
"sudo apt-get install motion"
```

2. Plug in your web cam

So now the software is on there it's time to plug in the web cam and ensure that everything is working, ensuring that you plug it into the powered hub, and then into the PI. Otherwise the webcam will not get enough power to turn on.

When plugged in type the "lsusb" command, you should see a line there with your web cam manufacture, that proves that you have the basic connectivity working.

ex:

```
lsusb
....
Bus 001 Device 002: ID 04ea:1142 Microsoft Corp.
....
```

3. Configure the software

"sudo nano /etc/motion/motion.conf"

In here there are a few basic changes that you need to perform:

- Daemon = OFF to ON
- webcam_localhost = ON to OFF

You can change other settings but it's recommend you don't take more than 2 frames, and you been the default frame pixel size, for stability.

4. Start the software

To ensure that the motion service will actually start as a daemon we need to change another configuration setting, so enter the following:

|"sudo nano /etc/default/motion"

Then change the value "start_motion_daemon=no" to "yes"

Finally you can start the motion service to stream the web cam images

"sudo service motion start"

Then after about 30 seconds browse to the new web interface, which should be at the below URL (where 192.168.0.100 is your Raspberry PI's IP address)

http://192.168.0.100:8081

5. Final Tweeks

Web Port

You could change the web interface port to 80 (from the default 8081), so that you can just browse to the IP address without having to put :8081 at the end, it’s really simple to do, just:

"sudo nano /etc/motion/motion.conf"

And then change “webcam_port 8081″ to “webcam_port 80″, save the file, and restart the motion service.

"sudo service motion restart"

6. Port Forwarding

This method needs Port Forwarding to go with live stream online ..

we tried it using Adsl connection , it succeeded …

but we need to connect with 3g modem so , we can't use it because 3g connect doesn't allow port forwarding .

V. How to Live Streaming Video using AVConv .

Installation

To transcode and broadcast the video, you will need to use a Linux application called AVConv (similar to FFmpeg). It is a command line program for transcoding multimedia files using the Libav Multimedia Framework. FFmpeg will give you the same result, but I personally prefer AVConv for ease-of-use.

To Install AVConv, run the following command:

"sudo apt-get install avconv"

Find a Broadcasting Platform:

To broadcast live streaming video to the Internet, you will need to create an account with Ustream, Justin.tv, or similar. After creating an account, it's important you find the "Stream Key" or "Stream URL".

Find the Stream URL and Key in Ustream:

- Login to Ustream.tv
- Open your "Dashboard"
- Click "Remote"
- Copy the "RTMP URL". You will need this once you begin broadcasting, in the next step of this tutorial.

Here is an example of the final URL:

rtmp://2.773.fme.ustream.tv/ustreamVideo/77775432

Broadcast Configuration

Once you have everything set up, command-line access to the Raspberry Pi, and your Ustream Stream Key URL, you're ready to initiate the broadcast! This is the fun part!

From this point forward, you only have to input TWO commands.

Open a new screen session:

screen

Enter the AVConv command with ALL of the parameters attached.

Basic command:

Enter this single command inside the screen session to start broadcasting! Yes, it's that easy!

```
"avconv -f video4linux2 -s 640×360 -r 10 -b 350k -i /dev/video0 -f flv rtmp://live.justin.tv/app/live_ReplaceThisWithYourURL"
```

In the above example,

-s defines the broadcasting resolution. (Widescreen = 640×360)

-r defines the maximum frame-rate

-b defines the maximum bitrate (k= Kbps)

-i defines the location of your USB webcam. In most cases, if you only have one USB video device plugged in, use the /dev/video0 phrase.

-f defines the type of video stream and location. In this example it's FLV transcoding (Flash video) to an RTMP server at Justin.tv

Don’t forget to replace the rtmp:// URL with your own Stream Key URL from the “Find a Broadcasting Platform” step.

An advanced example: (Warning: for experienced users only)

```
avconv -f video4linux2 -s 640×360 -r 10 -b 350k -i /dev/video0 -vf drawtext=”fontfile=/usr/share/fonts/truetype/freefont/FreeSansBold.ttf:text=’some text here ‘:fontsize=16:fontcolor=blue:x=2:y=360″ -acodec copy -metadata title=”24×7 Pi Cam” -f flv rtmp://live.justin.tv/app/live_81777DontStealMyKey6777
```

The above example is what I use on my “Eric’s Pets” live stream. You may notice a big difference from the original AVConv command. The -vf drawtext command allows you to define a font (download to your /freefont/ folder). Once a font is defined, you can add formatted text such as a “watermark” in the bottom left. The “-metadata title” allows me to tell Justin.TV what the broadcast page title should be. It can even get more advanced than this!

Please be very patient with AVConv and Raspbian. It takes time to perfect the stream quality. (My settings may not work for your camera or setup).

AVConv is very similar to FFmpeg in this sense. You can add many different features to your live video broadcast by simply putting new parameters in your initial command. You can learn more about these by reading AVConv/FFmpeg documentation.

AVConv will provide information while streaming, similar to the following string. If you see this, the stream should be broadcasting correctly.

Eric’s Live Bunny Cam via Raspberry Pi

Stream #0:0 -> #0:0 (rawvideo -> flv)

frame=215 fps=8 q=15.5 size=1001kB time=29.20 bitrate=281.0kbits/s dup=0 drop=3

At any time, you can stop the stream by typing Ctrl-C, and exit the screen session by typing “exit”. To close the screen session but leave AVConv running, type Ctrl-A-D. To enter a screen session that’s already running, type “screen -r”.

We have tried this method, but we found problems in the webcam output video format and some problems with Ustream.tv rtmp server.

VI. How to Live Stream a Video on Third party Servers.

-After the failure of the Ustream trial, one team member suggested to host an RTMP server.

-Luckily, we found a server that offers 2 weeks trial, and that server was the Onyx RTMP server.

-We then made a free domain name and a host, with a page that only hosts the RTMP link from the Onyx server.

-Also, this pages served as a client for receiving the sensors reading's from the Raspberry Pi device.

VII. Using FFMPEG tool:

A. Live Streaming

-In order to accomplish a successful live streaming video, we had to use the ffmpeg tool.

-Live streaming is done using the following command:

ffmpeg –f video4linux2 –i /dev/videoX –f flv <RTMP-Link>

Where:

-f : this flag is used in order to specify which video library we need to use in order to encode our video streamed bytes.

-i : this flag is used in order to specify which camera to stream from.

-Here, you can notice our second usage of the (-f) flag, that's because we need to encode the streamed video under the flv encoding extension file format in order to send it to the RTMP server.

B. Video Recording:

-For video recording, and according to Linux best quote :

"Keep It Simple and Stupid"

We wanted to use the same tool as for the live streaming, and the resulting command was super simple:

ffmpeg –f video4linux2 –i /dev/videoX –f <video-encoding-extension> <filename>

C. Image Capturing:

-Again, the lovely ffmpeg was used with this command:

ffmpeg –f video4linux2 –i /dev/videoX –vframe 1 <filename>.jpeg

-vfame : this flag is used to specify the number of frames taken each time, we specified here 1 frame, so that 1 photo is taken with the jpeg encoding extension.

VIII. Synchronizing the cameras:

-In order to work with the three cameras at the same time, and as soon as the Raspberry Pi device is launched, we modified the start up Linux file .bashrc, in order to execute the following commands:

.1 & .2 & .3 &

This command is executing files named (.1 , .2 and .3) ,and forking each execution process so that the next execution doesn't wait for the previous one to terminate to start his own execution.

-The first file (.1) contains the Image Capturing command, inside a loop, in order to handle any crash that might happen during the execution.

-The second file (.2) contains the video recording command, also inside a loop.

-And finally the third file (.3) contains the live streaming command, inside a loop.

IX. References :

1- http://raspberry-at-home.com/tag/sakis3g/

2- http://pingbin.com/2012/12/raspberry-pi-web-cam-server-motion/

3- http://techzany.com/2013/09/live-streaming-video-using-avconv-and-a-raspberry-pi/

4- https://libav.org/documentation.html