

## Getting a Trust 120 SpaceC@m to work under Linux.

I originally bought the webcam before I had migrated to Linux completely. The Guy in Bennets said he used Red Hat 9 and was almost certain the Cam was supported. I figured that was good enough for me.

The Trust 120 SpaceC@m is one of the highest quality webcams I have seen, the quality of the stream is definitely superior to many other webcams I have seen,



however it barely works as it should under Windoze, let alone Linux. I plugged it into the USB port whilst booted into Mandrake 10.0 running kernel 2.4.3, and looked at USBView and Hardrake, they both saw the device but didn't know what it was. So I searched for drivers, the Trust website had nothing (unsurprisingly) so I wasted my time by e-mailing them for support. As I expected I got very little help, about the gist of it was that I should convert back to M\$ products. I came across the [Sonix](#) drivers, which I simply could not get to work. The camera just was not recognised. It was not being assigned a file in /dev or making it apparent that it was recognised. In the end I gave up looking or trying, and decided I had wasted my money.

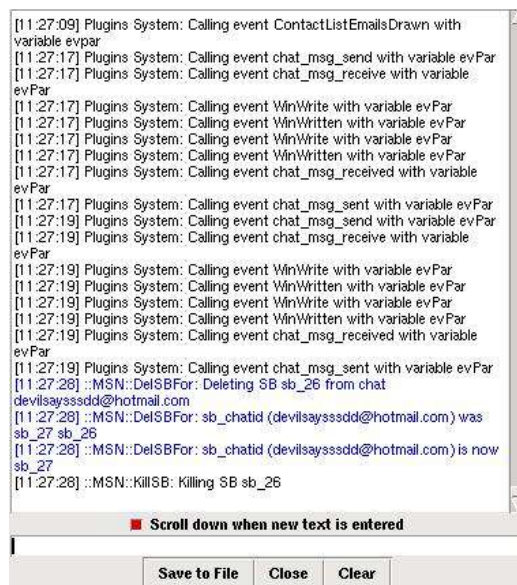
6 months later having upgraded to Mandrake 10.1 and a 2.6 kernel I decided to try again, this time I eventually stumbled upon a site which had a driver based upon the Sonix project. The SPCA5xx driver claimed to support many cameras including mine so I downloaded it. The site can be found at <http://mxhaard.free.fr/index.html> I then copied the tar ball to /usr, extracted it, did make and make install. Then I restarted X, and used the suggested program to view my webcam. [SPCAView](#) did find my webcam, and displayed a stream from it. However it displayed the compressed stream. My understanding is that the compression used is proprietary so it is unlikely there are any Linux suites that can accurately uncompress this. Therefore we need to use the uncompressed stream. I could not for the life of me work out how to tell SPCAvew to read the other stream.

I then accidentally ran gnomemeeting whilst removing its icon which had mysteriously appeared upon my desktop. It found the camera at /dev/video0, so with curiosity getting the better of me I went through the set-up procedure, and then when finally confronted with the gnomemeeting GUI, i found it displaying a reasonably high quality stream of me staring at the screen in amazement.

Now that i knew the webcam worked I had one main problem, how to display it to other people? I am not a general user of gnomemeeting, I prefer to use MSN and Skype, however since going native Linux over a year back, I have not been able to use

a webcam on MSN. Partially because my webcam would not work, and partially because there is very little software to allow you to do this. The [gaim-vv](#) project is developing this, and you can use a webcam over the Yahoo! Protocol, however MSN support is still somewhat Flimsy.

I use [AMSN](#) largely because I like the interface, and the plugin system. However it does not outwardly support webcams. After a little research i discovered that a plugin using linphone-im was being developed in order to support webcams. I downloaded the linphone files from amsn's cvs, compiled it and then proceeded to try and use it. Because it is still in development there is currently no button within the interface. You have to run it from the command window.

A screenshot of a command window with a light gray background. The window contains a list of log entries in a monospaced font. The entries are timestamped and describe plugin system events and chat messages. At the bottom of the window, there is a status bar with a red square icon and the text "Scroll down when new text is entered". Below the status bar are three buttons: "Save to File", "Close", and "Clear".

```
[11:27:09] Plugins System: Calling event ContactListEmailsDrawn with variable evPar
[11:27:17] Plugins System: Calling event chat_msg_send with variable evPar
[11:27:17] Plugins System: Calling event chat_msg_receive with variable evPar
[11:27:17] Plugins System: Calling event WinWrite with variable evPar
[11:27:17] Plugins System: Calling event WinWritten with variable evPar
[11:27:17] Plugins System: Calling event WinWrite with variable evPar
[11:27:17] Plugins System: Calling event WinWritten with variable evPar
[11:27:17] Plugins System: Calling event chat_msg_received with variable evPar
[11:27:17] Plugins System: Calling event chat_msg_send with variable evPar
[11:27:19] Plugins System: Calling event chat_msg_send with variable evPar
[11:27:19] Plugins System: Calling event chat_msg_receive with variable evPar
[11:27:19] Plugins System: Calling event WinWrite with variable evPar
[11:27:19] Plugins System: Calling event WinWritten with variable evPar
[11:27:19] Plugins System: Calling event WinWrite with variable evPar
[11:27:19] Plugins System: Calling event WinWritten with variable evPar
[11:27:19] Plugins System: Calling event chat_msg_received with variable evPar
[11:27:19] Plugins System: Calling event chat_msg_send with variable evPar
[11:27:28] ::MSN::DelSBFor: Deleting SB sb_26 from chat devilsayssdd@hotmail.com
[11:27:28] ::MSN::DelSBFor: sb_chatid (devilsayssdd@hotmail.com) was sb_27 sb_26
[11:27:28] ::MSN::DelSBFor: sb_chatid (devilsayssdd@hotmail.com) is now sb_27
[11:27:28] ::MSN::KillSB: Killing SB sb_26
```

Using the command

```
>> load plugins/linphone/Linphone.so
```

I loaded the plugin, then to invite someone to have a video conference with me i used the command

```
>> MSNAV::inviteAV myfriend@hotmail.com AV
```

this sent the invitation to them, they then accepted, and nothing happened, they said they could see a black box where the webcam picture of me should have been. I then realised that in all likelihood my firewall was blocking the connection, so as root i issued the command

```
>> shorewall stop
```

and then opened /etc/shorewall/rules with kate. I then added the ports 9078, 7078 and 6346 in both TCP and UDP.

Having done this I restarted Shorewall with the command

```
>> shorewall start
```

and tried again. When streaming to my girlfriend, who is behind a hardware firewall the image did not get through, and I could not receive her webcam, however when streaming to someone who has a direct connection to the internet They could see me and I could see them.

The simplest solution to the firewall problem would probably have been to get my girlfriend to unblock the appropriate ports, but I figured that there may be another solution.

So I started investigating the possibility of streaming the cam to a web page on my Apache server. I found several programs to do so, such as [webcam\\_server](#) however this insisted on streaming the compressed connection, giving an image like the one below;



So I looked for another way to stream the media. I could use webcam\_server to save the content to a jpeg image at set intervals, and then have the web page refresh at this same interval, however this leads to a very very jolty stream, so I figured as a last resort that it was worth looking at, however because it insisted upon using the compressed connection I had to abandon this approach.

I tried playing around with the Apache video mod but couldn't seem to get it to work in the way I wanted it to, so I eventually gave up on this one. Then I stumbled upon [camsource](#). By this time I was losing hope of getting a half decent stream out of my computer.

I downloaded and extracted the tar ball, went through the `./configure make && make` install process as usual. Thankfully all the dependencies were there because I really was getting fed up of the task I had set myself. I then edited `/etc/camsource` `conf.common` and saved the new file as `camsource.conf` I called `camsource` from the command line

```
>> camsource
```

and saw the following output;

Camsource 0.7.0 starting up...

Loading modules:

```
input_v4l (alias input_v4l) version 0.7.0 OK
jpeg_comp (alias jpeg_comp) version 0.7.0 OK
socket (alias socket) version 0.7.0 OK
wc_serv (alias wc_serv) version 0.7.0 OK
wc_serv (alias noresizewc) version 0.7.0 OK
http (alias http) version 0.7.0 OK
flip (alias flip) version 0.7.0 OK
resize (alias resize) version 0.7.0 OK
text (alias text) version 0.7.0 OK
```

Main init done and logfile opened.

Closing stdout and going into background...

It was actually loading up! That's the easy part done. I then pointed a web browser at `127.0.0.1:9192` and there it was an image of me! It was a still image admittedly but at least it was an image and not a psychedelic nightmare.



The difficulty now was in trying to get a moving image, I figured that in all likelihood what was needed was a meta command to refresh the content every second or every half second. I went back to the `camsource` website and after a few minutes of scanning through the site found something very useful. `Camsource` is compatible with `webcam_server`. To put it simply the Java applet that comes with `webcam_server` works for `camsource` as well!

So I changed to the directory I had extracted the webcam\_server tar ball to, luckily I am a messy bugger and had not removed the files yet. I recompiled the Java applet just to be sure with the command

```
>> javac *.java
```

and then recompiled it into the java applet

```
>> jar xvf applet.jar#
```

I then copied them across to the directory on my web server

```
>> su
>> cp applet.jar /var/www/html/webcam
>> cp webcam.html /var/www/html/webcam
```

On the edge of my chair now, I went into Firefox and entered the URL

127.0.0.1/webcam/webcam.html

and to my delight not only was there a picture of me, but the page was refreshing every second and my movement was obvious. However it was still very jerky, so I decided to alter the refresh rate. I opened webcam.html with kate and changed the lines

```
// "var speed" is the refresh rate adjustment in seconds.
var speed = 5;
var y = 5;
var x = speed + y;
```

I tried setting it to 1, and this seemed to have the desired effect. Whilst in this area I noticed something equally important. The line

```
var camImg = "http://localhost:9192/small" + "?" + now.getTime();
```

means that the applet will only work on this machine. Anybody who tries to use the applet on another machine will simply try to access port 9192 on their machine. So I changed this to the URL of my website.

```
var camImg = "http://benscomputer.no-ip.org:9192/small" + "?" + now.getTime();
```

this will point anyone who accesses it to my machine. However this leaves yet another problem. The router I am behind has an interesting method of resolving IP addresses. If I access 192.168.0.x then it will point me to the appropriate machine, including myself. However if I enter the address allocated by our ISP then rather than forwarding the query on to my machine as it would with all outside queries it simply directs me to the router. This means that I cannot view the image of myself using this page, and similarly no-one on the LAN can use this page either. So I decided to make a localised page for the network as well, I simply copied the webcam directory to one of my samba shares, and edited the web page again to make the line

```
var camImg = "http://benscomputer.no-ip.org:9192/small" + "?" + now.getTime();
```

```
read
```

```
var camImg = "http://benspc:9192/small" + "?" + now.getTime();
```

this allows anyone on my local area network to view the webcam.

The most obvious drawback with this method is that whilst camsource is running, anybody can view the webcam at any time. In order to restrict this i created a button which runs the command

```
camsource -k
```

this will stop the running camsource process, and stop people from being able to view the images. Then I created a shell script to run camsource and also to open a box that shows me the streamed image.

```
#!/bin/bash  
camsource  
konqueror /home/ben/.cam/webcam.html
```

this simply opens the local page in konqueror so that I can see the image. I used konqueror for this because it uses less memory, and I found that using Firefox worked, but if I already have a Firefox window open it will reuse that one.

I then changed the permissions of this script within

```
>> chmod 755 webcam
```

and placed it in my ~/bin folder

Now to run the webcam I simply have to click a button on my sidebar which will run this script, and to stop it I click the button below it.