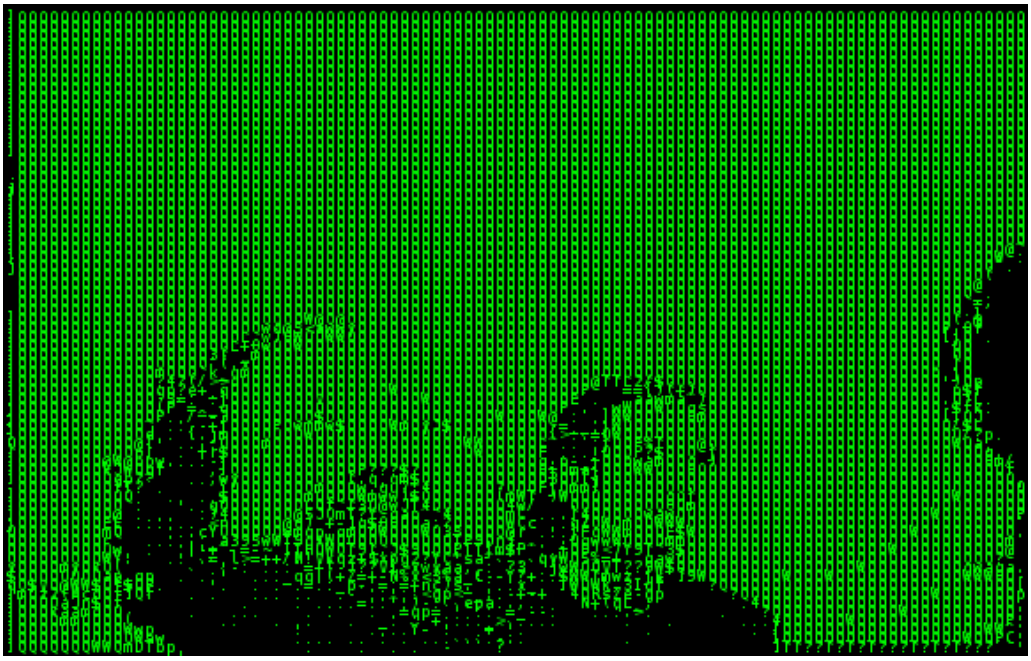
HasciiCAM supposed to stream ASCII video over the network on GNU / Linux

Author : admin



To continue with my lately ASCII centered articles I found **hasciicam**
hasciicam is a program to stream ASCII video over the network on Linux and probably can be easily made working on FreeBSDtoo.

The project concept is interesting in a matter of fun (play) point of view, however not too usable as we all know ASCII character looking faces doesn't look too pretty.

Below is the Debian (Squeeze) package description:

noah:~# apt-cache show hasciicam|grep -i description -A 7
Description: (h)ascii for the masses: live video as text
Hasciicam makes it possible to have live ASCII video on the web. It
captures video from a tv card and renders it into ascii, formatting the
output into an html page with a refresh tag or in a live ASCII window or
in a simple text file as well, giving the possibility to anybody that has a
bttv card, a Linux box and a cheap modem line to show a live ASCII video
feed that can be browsable without any need for plugin, java etc.
Homepage: http://ascii.dyne.org/

On hasciicam Project webpage is it is stated as a hardware you need to have:

> "As hardware you need to have a webcam or a videocard supported by "video 4 linux", most of the gear you can buy around should work well."

To install and test it I run:

noah:~# apt-get --yes install hasciicam

Though it is stated on the project website supposed to work display video fine with most 'linux ready' webcams, it didn't with this very standard one.

Here is the exact WebCamera model as identified to the kernel:

noah:~# dmesg|grep -i camera
[1.433661] usb 2-2: Product: USB2.0 Camera
[10.107840] uvcvideo: Found UVC 1.00 device USB2.0 Camera (1e4e:0102)
[10.110660] input: USB2.0 Camera as /devices/pci0000:00/0000:00:1d.7/usb2/2-2/2-2:1.0/input/input11

By the way, I use the very same CAM daily on for Skype video calls as well as the Camera is working with no problems to save video or pictures inside *Cheese*

Here is the exact WebCamera model as identified to the kernel:

noah:~# dmesg|grep -i camera
[1.433661] usb 2-2: Product: USB2.0 Camera
[10.107840] uvcvideo: Found UVC 1.00 device USB2.0 Camera (1e4e:0102)
[10.110660] input: USB2.0 Camera as /devices/pci0000:00/0000:00:1d.7/usb2/2-2/2-2:1.0/input/input11

The just installed *deb* has one binary file only */usr/bin/hasciicam*. To test it with the camera I issued:

noah:~# hasciicam -d /dev/video0
HasciiCam 1.0 - (h)ascii 4 the masses! - http://ascii.dyne.org
(c)2000-2006 Denis Roio
watch out for the (h)ASCII ROOTS

Device detected is /dev/video0
USB2.0 Camera
1 channels detected
max size w[640] h[480] - min size w[48] h[32]
Video capabilities:
VID_TYPE_CAPTURE can capture to memory
!! error in ioctl VIDIOCGMBUF: : Invalid argument

Unfortunately as you see from the output, it failed to detect the web camera model.
The exact camera besides its kernel detection naminf is a cheap external USB 2.0 *(fake brand / nonanem)* "universal" Web PC Camera (**SUPER .3mega pixel**)

For those who have a further interest in building and installing *hasciicam* on other Linux platforms than Debian and Ubuntu or whoever wants to look in the code check check Project webpage is. For those who are less of programmers (like me) the project is written in C programming language and uses **aa-lib** in order to render the video to ASCII.

On the site you will notice two totally schizophrenic looking pictures of presumably the project head developer ...

As I read in *man hasciicam* manual page it's said to be able to generate *ascii plain text and html* files as well as directly to write the output to console, which later probably can be streamed via the network.
Pitily as it didn't detect my camera I couldn't make some testing of its network capabilities.

A Streaming of ASCII couuld be done through pushing the .html output to a webserver and setting a php or javascript to loop through and refresh the browser over the uploaded files every sec or so.

Also I assume the ASCII video output saved in plain console could be streamed via **netcat** or some tiny scripted *perl* or *bash* script and directly observed via a telnet or ssh connection.
One playful way I can think of checking a stored video without the use of FTP is to login via ssh and do:

```
$ ssh someuser@somehost
$ watch -n 1 "cat video-ascii.html"
```

:)

Well something disturbing about *hasciicam* from a (purely Christian point of view) is it was developed by

some kind of non profit organization called **RastaSoft** on the project website, some of its authors has written **JAH BLESS**.

As I didn't succeeded seeing it working, I'll be interested to hear if someone who red this article and give it a try can report the web camera model used.