

## **AVT Cameras and Linux**

Ulf-Erik Walter  
Allied Vision Technologies GmbH  
<http://www.alliedvisiontec.com>  
November 2004

### **INHALTSVERZEICHNIS**

1. Preliminary Remark .....	2
2. Suppositions .....	2
2.1. Linux-Kernel.....	2
2.2. Kernel-Module .....	3
2.3. Problems .....	3
3. Libraries.....	4
3.1. libraw1394.....	4
3.2. libdc1394 .....	5
3.3. Rights for the Device File .....	5
4. The AVT-Library .....	6
5. Pictures on the Monitor: Viewer .....	6
5.1. Examples of the libdc1394.....	6
5.2. coriander.....	6
5.3. The AVT-Viewer: cc1394 .....	7
6. Follow-Up Projects .....	7

## 1. Preliminary Remark

With this document we want to explain the general procedure of how to get an AVT camera running under a Linux system. Therefore we abstained using a special procedure for a certain distribution like Redhat, SuSE or Debian. We would like to mention that the system that we used for all testing was a Debian 3.1pre Distribution.

I have put commands that need to be executed in the command line in Typewriter in an extra box and marked them with an upfront "\$". Should the appropriate command need root<sup>1</sup>-rights than we would replace "\$" with "#"

The most common function of all computers looks like this:

```
$echo Hello World
```

## 2. Suppositions

### 2.1. Linux-Kernel

Of course you will need a computer(x86-architektur) which has Linux installed as well as an AVT Camera. The Kernel may not be too old. (Younger than 2.4.22 or 2.6.1)

To find out which Kernel version you will need, just use

```
$uname -a
```

Should the Kernel be too old, you need to contact the distribution manufacturer and ask for a newer Kernel. Or download the source code free of charge from <http://www.kernel.org> and install it.

A description of the procedure of generating a new Kernel is a part of the Kernel source.

---

<sup>1</sup> System Administrator

## 2.2. Kernel-Module

The next step would be to make sure that the Kernel supports the IEEE 1394<sup>2</sup>. If the 1394-function is integrated in the Kernel than the 1394-System would be

```
$dmesg | grep 1394
```

Should you have a re-loadable Kernel-Device, you can look up the status of the device either with

```
$cat /proc/modules
```

or

```
#lsmod
```

Devices can be reloaded as root with

```
#insmod /lib/modules/2.../raw1394.o
```

after having adjusted the path. You will need the OHCI1395, VIDEO1394 and RAWIO1394 devices. Should there be no IEEE 1394 support, you would need to get the source code from Kernel, compile it and install the new Kernel. Currently no distribution is known where this problem has appeared.

## 2.3. Problems

Until now we had rare problems with the assignment of the Interrupts / IRQs. The behavior ranged from misleading error messages while re-loading the Kernel-Device (the IRQ was divided with the IDE Controller) up to confusing breaks within the photo telegraphy (IRQ was caused by the mouse at the USB-Bus) Should it come to inexplicable problems, you should first check

```
$cat /proc/interrupts
```

and check if it has come to an unfavorable overlapping.

---

<sup>2</sup> Also known as FireWire.

### 3. Libraries

By now any larger Linux-Distribution includes the necessary libraries.  
But it could still be that you need to exchange it with a newer version

#### 3.1. *libraw1394*

This library can be found under <http://sourceforge.net/projects/libraw1394>

An update of this library should be relatively easy.

Should you have the .tar.gz package (of course with the appropriate version number) you may use

```
$tar -xzf libraw1394_0.9.0.tar.gz
$cd libraw1394-0.9.0
$./configure
$make
```

or as root

```
#make install
#make dev
```

With the last commando the necessary devices<sup>3</sup> will be created. The rights are very restricted in this step and need to be adjusted according to the operating range of your system. You may test this library with

```
#src/testlibraw
```

This program reads and writes some data into the 1394 Subsystem.

---

<sup>3</sup> Device Files, that represent the devices

### **3.2. libdc1394**

The website <http://sourceforge.net/projects/libdc1394> is the place of recovery for the library which includes the application programming interface for IIDC conformable cameras. Please take care to use a newer version of the library than 0.9.5. Version 1.0.0 has been available since September 2004. If you make a manual update, you proceed as explained in libraw. Afterwards you need to create the devices manually.

```
#mkdir /dev/video1394
#mknod -m 600 /dev/video1394/0 171 16
#mknod -m 600 /dev/video1394/1 171 17
```

You may test the function of the libdc by using the attached example programs. For example is the dc1394 multiview included in the examples subdirectory as a complete Viewer, which source code relatively easily shows how the pictures get out of the camera.

### **3.3. Rights for the Device File**

Giving everybody and everything access to the devices would be a little bit too easy, because the rights at the first step are also for the CD-Rom driver and similar devices which are connected via FireWire. You should neither give rights to the device file, which could be used by everyone, nor should only "root" have access to the cameras which are used on a daily business term. A practicable solution would be to apply a new user group<sup>4</sup> and to assign the devices to this group. Now you only need to give this group read and write authorization and to allocate those users who should be working with the AVT cameras in the system.

---

<sup>4</sup> refer to /etc/group

## 4. The AVT-Library

The libavt1394 is an extension of the libdc1394. Her use is to make all extended features of the AVT-Dolphin, AVT-Marlin and AVT-Oscar camera families easier accessible. Currently this library is available as source code. It is compiled with

```
$tar -xzf libavt-1.0.2.tgz
$cd libavt-1.0.2
$.configure
$make
```

Now Root needs to call up the installation with

```
#make install
```

At the moment we do not have a manual available which would give an overview over all options that this library offers, but the Header File offers a short description about all functions.

## 5. Pictures on the Monitor: Viewer

### 5.1. *Examples of the libdc1394*

A small Viewer is included in the examples. This Viewer may not be very flexible at the first step, but it is for some projects the starting point of development. Small but easy to use!

### 5.2. *coriander*

This project : <http://www.tele.ucl.ac.be/PEOPLE/DOUXCHAMPS/ieee1394/coriander> should not be missing on any PC which uses 1394 cameras. With this Viewer you are able to use all features of the cameras, which are defined under IIDC Standard.

### 5.3. *The AVT-Viewer: cc1394*

This Viewer, which shows the possibilities of how to demonstrate the libavt194, shall give a clear picture of all features of the AVT- Cameras and also offer all defined influence possibilities under IIDC Standard. We made it especially easy to use, because some features are rather complex. This Viewer is absolutely essential for using those features of AVT- Cameras. The newest Version of this Viewer will be available at <http://www.alliedvisiontec.com>.

## 6. Follow-Up Projects

The following list can only show a small section from the many different options under Linux. She may be used as a starting point for own investigations.

- The Source of all Knowledge: <http://www.linux1394.org/>
- Various APIs
  - IIDC API: <http://sourceforge.net/projects/iidcapi>
  - <http://unicap.sourceforge.net/>
  - <http://kauri.auck.irl.cri.nz/~johanns/camwire/>
  - Qt-Bindings: <http://sourceforge.net/projects/libgmuvision/>
  - Java-Framework: <http://sourceforge.net/projects/jlibdc1394/>
  - Network-Middleware <http://graphics.cs.uni-sb.de/NMM/>
- Image Processing Libraries
  - <http://siptoolbox.sourceforge.net/>
  - <http://www.ee.uwa.edu.au/~braunl/improv/>
  - OpenCV: <http://www.intel.com/research/mrl/research/opencv/>
  - MATLAB: <http://www.ph.tn.tudelft.nl/DIPLib/>
  - HALCON: <http://www.mvtec.com>