Running The Nerd-Cam How-To[1]

T. Nathan Mundhenk
August 17, 2005
Nathan@mundhenk.com

[1] Directions on building the nerd-cam at: http://www.nerd-cam.com/how-to/

1. Hardware Systems

The nerd-cam system can be broken down into two primary hardware components the servomechanisms and the camera devices. Additionally, the software components also can be divided in the same way. It is important to understand at least the basics off all the components since this is an experimental system and many things can malfunction, which requires some expert knowledge in such an event.

A. Servomechanisms

The nerd-cam currently uses the mini-SSC hobby servo controller. In order for this to work it needs:

A. Serial commands from the computer
B. 4.5 Volt connection for servo motor power
C. 9.0 Volt connection for servo controller power
D. A connection to the servo motors

If any of these are absent the servo will not work. One can know if any of the components are absent given the following symptoms:

A. Mini SSC green LED light is on, servo motors are limp – 4.5 volt is disconnected
B. No Mini SSC light – 9.0 volt is disconnected
C. Servo motors are stiff and green light is on – serial command is not being sent or received
D. One of the servo motors is limp, but the other is not – One of the motors is either broken or connected improperly
E. The servos are having trouble moving – Make sure that the cords connected to the camera are not holding the camera down. Additionally, this may be a result of a broken servo.

Thus, without serial input, if the nerd-cam mini-SSC is plugged in correctly, the camera will stand rigid with both servos. If it is not, then the mini-SSC has not been connected properly. Once the servomechanism is connected, it will move on serial command from the computer.

The mini-SCC can be configured to run at 2400 and 9600 baud. It can also be configured to run in a 90 degree arc in increments of .35 degrees or in an arc of 180 degrees in increments of .71 degrees. Currently, the default is to set it for 9600 baud with a 180 degree arc. Changes in these settings will most likely require changes in the software code.

A final note on the nerd-cam is that the cameras do not center perfectly. That is, when given a command of 90 degrees by 90 degrees the camera may in fact be pointing 87 degrees by 92 degrees. This is a manufacturing problem with the cameras but is constant for each camera. As such, center each camera and measure its offset. This can be used until a camera servo is replaced as the permanent offset from center. Once the offset has been computed and plugged in, each camera will center and move properly to the angle requested.

B. Firewire Camera

The firewire camera has a far simpler hardware configuration, but is more apt to failure based on software anomalies. For standard firewire, power to the camera is provided by the firewire bus. On notebook computers and the 4 pin firewire connection, power must be provided externally via the power input connector for the Fire-I camera. This can be done with a standard 12 volt input AC adapter with 1 amp current input.

If the camera is connected to the computer and power is provided then the camera LED above the lens will be either green or yellow. If the LED is yellow that indicates that the camera is not currently talking with the computer. Yellow means that the computer and the camera are talking. It should be noted that the state of the software drivers in linux for the firewire camera cannot be reliably deduced from the state of the LED. The only way to be sure that the drivers are functioning is to insert the kernel modules with the camera plugged in (covered in software systems).

It should also be noted that while firewire is a plug and play standard, the Linux implementation of firewire does not act reliably for this feature. As such, unplugging the camera can cause the system to hang on some occasions and should be avoided. Further, if the camera is unplugged, when it is plugged back in, the firewire drivers will most likely need to be re-loaded.

2. Software Systems

The software systems fall into two categories as mentioned. The mini-SCC is controlled in a semi-stateless manner via serial controller. The camera is has two major components. The first is the firewire driver and the second is the frame grabber which, when the driver is installed, grabs each frame of video and provides it to a C++ program using the iLab Neuromorphic Vision Toolkit (INVT). Other packages may be used to grab video, but INVT provides a simple interface for interacting with the camera.

A. Servomechanisms

The servomechanism is simple from a driver standpoint. To move the camera a set of bytes is written to the serial port, which contains the camera to be moved, and what position the camera is to be moved to. The mini-SSC has 8 servo ports and the camera uses two of these for its pan and tilt. If one wishes, one can connect up to 4 nerd-cam cameras to a single mini-SSC controller. To test the cameras one can use the *test-cameraMotion* program included in INVT.

Since the mini-SSC does not provide feedback to the computers serial connection, one must in software keep track of the servo position. Additionally, when providing movement commands, one has to compute the trip time for the movement. The software should wait the amount of time it takes for the servo to move before issuing another command. Additionally, one cannot assume the position of the camera to be correct until the travel time has passed. That is, you do not know what position the camera is in during the travel time. It can be a handy short cut to issue several shorter movements if it is needed to change to cameras movements frequently or if it is needed to be known what position it is in.

A final note on the use of the serial port. On many systems the serial port has restricted write access. As such, one may have to change to permissions of the serial port as root in order to write to the serial port. If one is unable to move the mini-SSC and it is connected properly, this is one of the most common causes.

B. Firewire Camera

i. Drivers

The firewire camera uses two software components. The first component is the firewire driver, the other is the frame grabber. The driver consists of *raw1394* and *dc1394* driver modules. When inserted correctly into the kernel, using *lsmod* as root one will see *video1394*, *ohci1394* and *raw1394* modules inserted. If these are missing, the camera will not operate to grab video.

To insert the firewire modules use the *firewire script* (appendix B) to automatically insert the kernel modules and start the drivers. The script does not work reliably to remove the kernel modules. Instead, to restart the firewire driver, as root type *rmmod video1394* then *rmmod raw1394*. After this run the firewire script with the start command. If the camera is plugged in and the LED on the camera is lit, but you are unable to grab video, most likely the firewire needs to be reset by re-inserting the kernel modules as just described.

ii. Grabing software

The software grabber uses the model manager in order to take in command line arguments. An example of a simple program to test to video grabber is *test-grab*. For grabbing options one can type *test-grab --fg-type=1394 --framegrabber*. This will give one a variety of option on controlling the camera. The most simple way to test the camera is by typing *test-grab –fg-type=1394*. This starts the frame grabber and specifies that firewire camera is to be used. It will default to *320 x 240* resolution at 30 frames per second. It is recommended that one increases the number of frame buffers with the *--framegrabber-nbuf* option. For grabbing in color at 640 x 480 resolution, one must use 15 frames per second with the *--framegrabber-fps* option.

If the camera fails to grab frames with the default settings using the Fire-I camera, then most likely the drivers need to be re-inserted into the kernel. The grabber software itself is fairly reliable on any a computer that is fast enough. Typically, the failure occurs at the level of the driver.

Appendix A: Step by step for camera usage

Basic usage of nerd-cam follows several basic steps which will be expanded.

1. Connect the firewire camera and turn the computer on
2. Download and compile iLab Neuromorphic Vision Toolkit (INVT)
3. Connect mini-SSC to serial, servo and power
4. Make sure 1394 drivers have been installed, if not install them
5. Test using INVT test programs

1. Connect camera

In general since the plug and play capabilities of firewire are still somewhat dubious, it is recommended that the camera be connected before the system is powered on. It is up to the individual to determine if this is necessary, but it is recommended that for the first trail of the camera this is done. This way, if the camera fails for some reason, a person has a reduced number of possible causes.

2. Download and compile INVT

The code can either be obtained from iLab's subversion repository or from the archive on iroom.info. On iroom.info, the latest code is stored in */home/raid1/local_svn_copy/*. For the SVN version check with the ilab home page for the latest info on how to obtain the code at *http://ilab.usc.edu/toolkit/*.

Once the code has been downloaded and unpacked change into the saliency directory. Compile the code in this way:

1. type "./configure" – this will configure the code for your system as well as tell you which libraries you might be missing
2. A note on compiling: The saliency code is designed to treat warnings as errors by default. You can turn this off by using "./configure --disable-werror" when executing step 1. Additionally, a less verbose output can be seen by using the "--enable-quietcompile" option. Check with "--help" for more options.
3. type "make tprogs" – this will make the test programs test-grab and test-cameraMotion. Alternatively, a "make all" will also work, but will make more code than is needed.
4. The binaries are compiled into a directory called /saliency/bin. This is where you will find the test programs once INVT has finished compiling.

3. Connect the mini-SSC

The mini-SSC can be connected as follows:

1. Connect the standard 9 volt battery type connector to a standard 9 volt DC input. The power usage is low, so probably any inverter will do. The green LED will light when this is connected. Additionally, a 9 Volt battery will do since the power draw is rather low.
2. Connect the two black and red leads to a standard 4.5 volt DC input. This should have something close to 1 amp current ability. 800 mA seems to be sufficient.
3. Connect the serial port to the RJ-12 phone plug input.
4. Connect the servomotors to the mini-SSC. This is the row of 3 x 8 pins on the top of the board. CAUTION: connecting the polarity backwards can burn out the servo motors.
5. Once the servos are connected and power is connected the cameras should be stiff and unmovable. The serial does not need to be connected in order for the servos to be this way.

6. Install or check the IEEE1394 drivers

This is the weakest link in the process. As mentioned, if the frame grabber fails to grab, this is the most likely point of failure. To insert or remove firewire drivers you must either be logged in as root or be granted permission to run it using the command *sudo*.

You should see the following lines with an *lsmod* command:

video1394	18168	0
ohci1394	31108	1 video1394
raw1394	24600	0
ieee1394	355672	4 video1394,ohci1394,raw1394,sbp2

As a side note, make sure that if you compile the code to run on a 64 bit machine then you link against the 64 bit libraries. As such, the first time, it is not recommended to compile to the code on a 32 bit machine if the target it a 64 bit machine.

7. Test using INVT test programs

If the code compiled alright and you haven't pulled out all your hair hunting down all the required libs you can now run the test programs. There are several test programs you may use. Each one is found in /saliency/bin once compiled.

1. test-grab – Use this with the –fg-type=1394 option. This takes the video from the camera and displays it in a window. The code is simple, so it should give you an idea of how to integrate the grabber into other programs.
   a. Source located in /saliency/src/AppDevices
2. test-grabVideo – This does the same as test-grab except also dumps each frame to the hard drive as a ppm.

    a. Source located in /saliency/src/AppMedia

3. test-cameraMotion – This calls the SSC driver and will move the camera to different positions. It also does pixel to angel conversion automatically. This is a good program to look at to understand the camera control.
    a. Source located in /saliency/src/AppDevices

Appendix B: Firewire script

```
#!/bin/bash
#
# chkconfig: 2345 85 15
# description: configure and start firewire for beobots

# source function library
. /etc/rc.d/init.d/functions

case "$1" in
  start)
	gprintf "Starting Beobot firewire: "
	insmod ieee1394 > /dev/null 2>&1
	insmod ohci1394 attempt_root=1 > /dev/null 2>&1
	modprobe raw1394
	modprobe video1394
	if [ ! -c /dev/raw1394 ]; then
	  mknod /dev/raw1394 c 171 0
	fi
	if [ ! -c /dev/video1394/0 ]; then
	  rm -rf /dev/video1394
	  mkdir /dev/video1394
	  mknod /dev/video1394/0 c 171 16
	fi
	chmod 777 /dev/raw1394 /dev/video1394/0
	chmod 777 /dev/tts/*
	success "firewire startup"
	echo
	;;
  stop)
	gprintf "Shutting down Beobot firewire: "
	rmmod video1394
	rmmod raw1394
	rmmod ohci1394
	rmmod ieee1394
	success "firewire shutdown"
	echo
	;;
  restart|reload)
	$0 stop
	$0 start
	;;
  status)
	lsmod
```

```
        ;;
  *)
        gprintf "Usage: firewire {start|stop|status|restart|reload}\n"
        exit 1
esac

exit 0
```