Video Sentinel Users Manual

J C Chavez
Sandia National Laboratories
jochave@sandia.gov
December 12, 2003

Introduction

This manual provides installation instructions and operating instructions for the Video Sentinel system, which is used to detect, record, and display meteor events using an ordinary video camera and a PC. The system includes a video camera, the Video Sentinel box, which takes the video signal from the camera, a PC, which talks to the Video Sentinel box through a cable connected to the printer port, and several software applications that control the works.

PC Requirements

The PC used in the Video Sentinel system should meet the following minimum requirements:

- Pentium class
- 200 MHz or greater
- 64 MB RAM or greater
- 4 GB Hard Drive or greater
- CDROM Drive
- Printer Port supporting EPP mode

The Video Sentinel system requires RTLinux, which is a real-time version of the Linux operating system. Installing RTLinux will require patching the kernel and adding modules to an existing Linux setup (such as Red Hat 8 or 9). This should probably be attempted only by someone with good Linux experience.

Video Sentinel Hardware Installation Instructions

All hardware connections should be made with the PC turned off.

Printer Port:

The current Sentinel software release contains two programs (sentuser and sentuser2) that can operate up to two Sentinel boxes connected to different printer ports on the same PC. The first program uses the printer port at address 0x278 and also uses IRQ 9. The second program uses printer port 0x378 and IRQ 7. Printer ports should be set up in EPP mode.

The first-generation Sentinel box has a 25 pin Canon connector and uses an IEEE-1284 A-A switchbox cable to connect to the PC. These boxes unfortunately suffer from an overly sensitive electronic interface that prevents them from working with many built-in or add-on printer ports. They do appear to work reliably, however, with the **Parallel Pro™** card from **SIIG®** for the ISA bus.

The second-generation Sentinel box features a redesigned electronics interface with a Centronics connector that uses a readily available IEEE-1284 A-B printer cable to connect to the PC. This new design appears to work reliably with built-in and add-on printer ports.

Video Sentinel Box:

Connect the Video Sentinel box to the PC printer port using either an IEEE-1284 A-A switchbox cable (first generation Sentinel) or an IEEE-1284 A-B printer cable (second generation Sentinel).

Connect the incoming video cable to the **middle** BNC connector on the Video Sentinel box (the one next to the power supply input).

Plug in the power supply AC power cord and connect the power supply DC cable to the Video Sentinel box power supply input.

Turn on the PC.

You can optionally connect a video monitor to the video out BNC connector on the Video Sentinel box (the connector furthest from the power supply input). When the Sentinel box is actively looking for meteors, the video out connector provides a delayed version of the input video signal.

Installing RTLinux

Installing RTLinux involves first installing a recent version of Linux on your PC and then patching and rebuilding your kernel. Links for the software patches and instructions can be found at:

http://www.opentech.at/howtos/multiboot-howto/html/index.html

In particular, I used the instructions at:

http://www.opentech.at/howtos/multiboot-howto/html/install_en_2.html

However, while installing on a RedHat 8 or RedHat 9 Linux system, I came up with the following notes for each section:

It is best to log in to your session as root rather than to do a "su" from your user session. This way you have direct access to commands in the /sbin directory (such as lilo).

2.1   Source

I used the following sources:

http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.20.tar.bz2
ftp://ftp.rtlinux.at/pub/rtlinux/rtlinux-3.2-pre2.tar.bz2

2.2   Extracting -- No comments

2.3   Patching -- No comments

2.4   Kernel Configuration

I downloaded the howto directory from:

http://www.opentech.at/howtos/multiboot-howto/download/howto-1.2.3.tar.gz

Then in the configuration menu, I read the file located at:

howto-1.2.3/configs/linux-2.4.20-rtl3.2-pre2/my_config

The only change I made then was to the Processor Type and Features entry to select the correct processor type for my machine.

2.5   Kernel Installation -- No comments

2.6 LILO Configuration

My RH installation uses the grub loader instead of lilo. Therefore, all I did was modify the grub configuration file at /etc/grub.conf by adding the following lines:

```
title=RTLINUX (2.4.20-rtl3.2-pre2)
    root (hd0,0)
    kernel /vmlinuz-2.4.20-rtl3.2-pre2 ro root=/dev/hda6
```

(hd0,0) corresponds to the first (hd0) physical disk drive's first (0) partition. It should be the partition where the boot file system (/boot) resides. It may be different on your machine.

In my setup /dev/hda6 is where the root file system (/) resides. It may be different on your machine. A "df" command will show you where the file systems reside.

2.7 RTLinux Configuration

I made no changes to the default configuration.

2.8 RTLinux Installation

Issue the following command after "make modules modules_install":

```
make install
```

2.9 Reboot

If you are using the grub loader, skip the lilo command.

2.10 Regression test

I get a large number of warnings and the fp_test fails. I think these can be safely ignored.

Video Sentinel Software Installation Instructions

The following assumes that RTLinux is installed on your PC.

The Sentinel software is contained in a compressed tar file that is available at:

ftp://ftp.sandia.gov/outgoing/meteors/sentinel_source2.tar.gz

or, by e-mailing me at:

jochave@sandia.gov

Copy the sentinel_source2.tar.gz file to your home directory on the Sentinel computer.

On the Sentinel computer, become root (requires root password):

```
su
```

Issue these commands:

```
tar -xzvf sentinel_source2.tar.gz
cd sentinel_source2
make
exit
cd
```

Video Sentinel Checkout and Test

Every time the computer reboots, the RTLinux modules should be re-installed, by first becoming root:

```
su
```

and then typing:

```
cd sentinel_source2
make rtlinux
exit
cd
```

To start the Sentinel control program, type the following:

```
./sentuser
```

(This assumes you are using parallel port 0x278 and IRQ 9. If you are using parallel port 0x378 and IRQ 7, you can type instead: `./sentuser2`)

After a few seconds you should see a "`Command:`" prompt where you can issue commands. If you type "help" you can get a list of available commands.

To see if the Sentinel box is functional, type the following series of commands:

`reset`	This sends a reset pulse to the Sentinel box.
`init`	This loads control code into the Sentinel Video Decoder and Encoder. This takes several seconds to complete.
`test`	This runs a memory test in which 200 frames of pseudo-random data are written to the Sentinel box, whereupon these 200 frames are read and verified. This test takes several minutes to complete.

If you see zero errors at the end of the test, you have verified that the communication interface, the Sentinel memory, and the Sentinel control electronics are in good working order.

To see if the Video Decoder is functional, type the following:

```
capture image.dat
```

This captures a single 640 x 480 gray-scale image to the file image.dat, which can be used to create an image mask (described in the next section). To view this image, first open a new terminal window and convert the image to the BMP format with the following command:

```
convert –size 640x480+0 –depth 8 gray:image.dat image.bmp
```

You should now be able to view image.bmp with a viewing application, like Gimp or GQView.

Making an Image Mask

For its video processing, the Video Sentinel box uses an image mask to control which parts of the input image are noticed and which parts are ignored. The image mask is just a 640 (horizontal) by 480 (vertical) byte array that define a threshold for each pixel of the incoming video image. If the value of the byte is 255 (maximum), then the threshold is too large to ever be crossed and so no changes at that pixel location will ever be noticed. If the value of the byte is 0, then even insignificant changes at that pixel location will be noticed.

The plan, therefore, is to create an image mask file, exactly 640 * 480 bytes in size (307,200 bytes), in which the areas of interest are set to a reasonable threshold and all other areas are set to 255. For our cameras, we have been using threshold values of between 35 and 40. Depending on your ambient lighting condition and how noisy your camera is, you may require higher or lower values.

If the entire image is of interest, you should create a file in which all the bytes are set to the same threshold value. You can create such a file by typing the following in a terminal window:

```
python -c "open('mask.dat','w').write(chr(40)*640*480)"
```

This creates a file named "mask.dat", exactly 640*480 bytes in size, in which each byte is set to 40.

If only parts of the input image are of interest, then a more complicated image mask must be created. This typically involves two steps:

1) Capture an image from the camera to use as a template.

2) Manipulate the image to form an image mask.

The sentuser program (as described above) can be used to capture a gray-scale image to a file from the camera connected to the Video Sentinel box. This image file is 640*480 bytes in size.

A number of tools can then be used to manipulate the image to form an image mask. I like to use NIH Image for Macintosh. However, on Linux you can use Gimp. Unfortunately, Gimp does not import gray-scale files directly so the file must first be converted to something Gimp can understand, manipulated in Gimp, and then converted back to a raw gray-scale file at the end.

Given that you have captured a gray-scale image called "image.dat", convert this to a bmp file by typing the following in a terminal window:

```
convert -size 640x480+0 -depth 8 gray:image.dat image.bmp
```

You can now open "image.bmp" with Gimp and manipulate it to form a mask. This typically involves painting areas with the draw tools and perhaps using other image processing tools like the threshold tool and the invert function. The thing to keep in mind is that in the end, the areas to be ignored should have a pixel value of 255 (white) and areas to be noticed should have a pixel value equal to the threshold (35-40). Once the mask is complete, save the result back to "image.bmp". Type the following in a terminal window to convert the image back to a raw gray-scale file:

```
convert -size 640x480+0 -depth 8 image.bmp gray:mask.dat
```

Using the sentuser program

Every time the computer reboots, the RTLinux modules should be re-installed, by first becoming root:

```
su
```

and then typing:

```
cd sentinel_source2
make rtlinux
exit
cd
```

The sentuser program is used to interface with the Video Sentinel box.

To start the Sentinel control program, type the following:

```
./sentuser
```

(This assumes you are using parallel port 0x278 and IRQ 9. If you are using parallel port 0x378 and IRQ 7, you can type instead: `./sentuser2`)

After a few seconds you should see a "Command:" prompt where you can issue commands. If you type "help" you can get a list of available commands.

A typical command sequence to begin detecting meteors would be as follows:

`reset`	This sends a reset pulse to the Sentinel box.
`init`	This loads control code into the Sentinel Video Decoder and Encoder which takes several seconds to complete.
`mask mask.dat`	This loads the mask file "mask.dat" into the Sentinel box.
`timeon 18:30:00`	This allows meteor detection to begin at 6:30 PM.
`timeoff 06:30:00`	This allows meteor detection to end after 6:30 AM.
`trigger`	This initiates meteor detection.

Whenever a meteor is detected, the time is displayed in the terminal window and a portion of the video is captured to a file. The application **sdisplay**, which is described below, can then be used to display the captured video.

If you plan to run two Sentinel boxes with the same PC, you can start **./sentuser** in one terminal window and **./sentuser2** in a second terminal window. Each program runs more-or-less independently of the other. The only known restriction is that you should not issue the trigger command in either program until the init command has been issued in both programs. For example, to be safe, you can issue all your non-trigger sentuser commands in one window, then issue all your non-trigger sentuser2 commands in the other window, and finally issue the final trigger commands in both window

sentuser Command Reference

The following section describes the commands available within the `sentuser` (or `sentuser2`) application.

`reset`

This sends a reset pulse to the Video Sentinel Box. The response should be "Reset sent", followed by the command prompt.

`init`

This initializes and loads control codes into the video circuitry. After several seconds, the response should be "Video initialized", followed by the command prompt.

`capture image.dat`

This captures a single gray-scale image to the file "image.dat"

`mask mask.dat`

This sends the mask file "mask.dat" to the Video Sentinel box. After about one second, the response is the command prompt.

`threshold 37`

This modifies the image mask in the Video Sentinel box in the following way: All pixels that are set to 255 are unchanged but all other pixels are set to the specified value (in this case 37). This allows us to adjust the threshold in the image mask without forcing us to create a new image mask. This command would normally be issued after the mask command but before the trigger command. After about one second, the response is the command prompt.

`timeon 18:30:00`

This sets the daily turn-on time of the Video Sentinel box, in this case to 6:30 PM. Normally, we want to turn on in the evening and turn off in the morning. The default turn-on time is 6:00 PM. You may want to change the turn-on time as the seasons change and when changing to or from Daylight Savings Time. The response is "Time On: 18:30:00" followed by the command prompt.

`timeoff 05:30:00`

This sets the daily turn-off time of the Video Sentinel box, in this case to 5:30 AM. Normally, we want to turn on in the evening and turn off in the morning. The default turn-off time is 6:00 AM. You may want to change the turn-off time as the seasons

change and when changing to or from Daylight Savings Time. The response is "Time Off: 05:30:00" followed by the command prompt.

`trigger`

This causes trigger to be enabled. The response is the command prompt. (However, if the command is issued during the "Off" duration, the response is "Turn Off: <time>, where <time> is the current time, followed by the command prompt.)

Once trigger is enabled and the PC clock is within the "On" duration (from turn-on to turn-off), the Video Sentinel box looks for above-threshold changes in the video image. If it sees enough of a change, an event is generated and up to 6 seconds of video are captured in the Video Sentinel box. The PC then transfers a subset of the video data (centered on where the change occurred) from the Video Sentinel box to a file in the current directory. This file can be viewed with the sdisplay program described below.

`stop`

This causes trigger to be disabled. The response is the command prompt. (Unfortunately, there is a bug in the code that sometimes prevents the command prompt from reappearing. Then, the only recourse is to force the application to quit by pressing control c.)

`test`

This runs a test of the Sentinel Box memory. 200 frames of pseudo-random data are written to the Sentinel Box memory. Then, these frames are read from the Sentinel Box and verified. If an error is detected, the expected value and the actual value are displayed. At the end of the test, the total error count is reported.

`maskcapture testmask.dat`

This writes the current mask within the Sentinel Box to the file: testmask.dat
This command can be used to verify that the mask (which was set with the "mask" command) has not been corrupted.

`perhour 20`

This command sets the event rate limit to the specified value (in this case 20). The default is 12 events per hour. Whenever the number of events in the last hour exceeds the event rate limit, the sentuser program will go to sleep after each subsequent event long enough to strictly limit the event rate to the specified value. This helps to keep us from filling up the disk with erroneous events during lightning or wind storms.

```
pixel+
```

This command increments the number of pixels above threshold that are required to indicate an event. The default is 12.

```
pixel-
```

This command decrements the number of pixels above threshold that are required to indicate an event. The default is 12.

```
clock
```

This command prints the current sentuser clock time. The sentuser clock time is based on the PC's system time but also uses a time correction value in a file named "TimeOffset" in the current directory (if it exists). This correction value (which is a text string) specifies the number of seconds that must be added to the system time to produce the correct time. A separate application (like the sclock application described below) can be used to periodically update the TimeOffset file so that accurate time is maintained.

```
setclock
```

This command sets the PC's system time by adding the correction value found in the file named "TimeOffset" (if it exists).

Using the sdisplay Program

The sdisplay program is used to view video files generated by the Video Sentinel box and the sentuser program. These files have file names that indicate when the file was created. They look like this:

`v20020115_034512.45`

The format is `vYYYYMMDD_hhmmss.ss`, where `v` is just `v`, `YYYY` is the year, `MM` is the month, `DD` is the day of the month, `hh` is the hour, `mm` is the minute, and `ss.ss` is the second (to a resolution of 0.01 seconds).

You start the sdisplay program by opening a new terminal window and typing:

`./sdisplay &`

When the program starts, it opens a 640 x 480 pixel window, looks for the first file in the current directory that begins with "v2", and plays the video from that file. The file name is shown in the window caption. The file contains the complete reference frame from the original captured video but only a 128 x 128 pixel subset of each subsequent frame, centered on wherever changes were seen. The movie consists of a fixed background image upon which a smaller sub-image movie is shown like a moving sprite. As the movie is played, 4 white pixels mark the corners of the 128 x 128 pixel sub-image.

The interface for the sdisplay program is extremely simple, consisting entirely of single keystroke commands. These commands are listed below:

<space-bar>   Replay the current video file.

<right-arrow> Load and play the next video file that begins with "v2" in the current directory. If there are no more files, replay the current video file.

<left-arrow> Load and play the previous video file that begins with "v2" in the current directory. If there are no previous files, replay the current video file.

<shift right-arrow>   (Press right arrow while holding down shift key.) Load the next video file and show only the first frame. This is much quicker than playing the entire movie so it can be useful when you want to quickly scan through a number of files.

<shift left-arrow>   (Press left arrow while holding down shift key.) Load the previous video file and show only the first frame. This is much quicker than playing the entire movie so it can be useful when you want to quickly scan through a number of files.

<home> Load and play the first file that begins with "v2" in the current directory.

<end> Load and play the last file that begins with "v2" in the current directory.

<del> or k If the current video file is not the last, move the current file to the Trash directory, then load and play the next video file. This is how you can remove uninteresting files. The files still exist in the Trash directory, however, so they must be removed occasionally with the unix command: `rm -f Trash/*`

<shift del> (Press del key while holding down the shift key.) If the current video file is not the last, move the current file to the Trash directory, then load and show only the first frame of the next video file. This is much quicker than playing the entire movie so it can be useful when you want to quickly remove a number of files.

1   Set movie speed to normal speed and play current video file.

2   Set movie speed to half speed and play current video file.

3   Set movie speed to quarter speed and play current video file

4   Set movie speed to eighth speed and play current video file.

5   Set movie speed to sixteenth speed and play current video file.

0   Set movie speed to triple speed (100 frames/sec) and play current video file.

c   Use the amber color look-up table defined in the file: CLUT.txt This shows the movie in an amber pseudo-color that may allow you to discern amplitude gradations better than with the normal gray-scale image.

g   Use the normal gray-scale color table. This shows the movie in the normal gray-scale.

<esc> Exit the sdisplay program.

Using the sclock program

The sclock program is used with an **Ultra-Link** Model 325 WWVB Receiver/Decoder to provide accurate time to the sentuser and sentuser2 programs. The Ultra-Link unit monitors WWVB transmissions that allow it to maintain accurate time. The sclock application talks to the Ultra-Link through a serial interface and periodically updates a time correction value in the file "TimeOffset" that indicates the number of seconds that must be added to the PC system time to produce an accurate time. This correction value is subsequently used by the sentuser and sentuser applications to accurately time-tag meteor events.

In a separate terminal window, start the sclock program by typing:

`./sclock`

This assumes you are using COM1 to connect to the Ultra-Link. If you are using COM2 to connect to the Ultra-Link, then you should start sclock by typing:

`./sclock 2`

The program periodically outputs the message string from the Ultra-Link to the terminal. Depending on the strength of the WWVB signal at your location, it may take several minutes for the Ultra-Link to acquire the correct time.

Making a Quicktime Movie

To allow wider dissemination of captured meteor events, I have developed applications to convert video event files into Quicktime movies and JPEG composite images. Versions of the application that run under Windows are available at:

ftp://ftp.sandia.gov/outgoing/meteors/RTMeteorMovie.exe
ftp://ftp.sandia.gov/outgoing/meteors/CRTMeteorMovie.exe

The first application produces a grayscale movie while the second application produces an amber tinted pseudo-color movie.

You can also e-mail me if you would like a version of the application that runs on a Macintosh.