

## Sensoray Model 810/811/911 Quick-Start Instruction for Linux

- 1) Sensoray Model 810/811/911 applied the NXP SAA713xHL chipset as the A/V (Video and Audio) capture device. Therefore, under Linux, the native saa7134 and saa7134\_alsa device drivers come with the kernel, can be used for the Model 810/811/911. The API complies with the V4L2 (Video For Linux Two) API specification. Refer to:  
<http://v4l2spec.bytesex.org/v4l2spec/v4l2.pdf> or  
<http://v4l2spec.bytesex.org/spec/> or  
<http://linuxtv.org/downloads/v4l-dvb-apis/>,  
for the specification details.
- 2) Normally, when user(s) installed any one of the popular/professional Linux distributions like Ubuntu, openSUSE, RedHat, Fedora, CentOS, Linux-Mint, etc., the saa7134 and saa7134\_alsa driver modules should be installed in the system, by default/ and automatically.
- 3) To check if the driver modules are installed, from a terminal window, using command "locate saa7134", you would see that they are in directory:  
/lib/modules/2.6.?-?-generic/kernel/drivers/media/video/saa7134/.
- 4) If not, or you want to get the latest V4L-DVB driver package installed, refer to "[How to Obtain, Build and Install V4L-DVB Device Drivers](http://linuxtv.org/wiki/index.php/How_to_Obtain_Build_and_Install_V4L-DVB_Device_Drivers)" at [http://linuxtv.org/wiki/index.php/How\\_to\\_Obtain\\_Build\\_and\\_Install\\_V4L-DVB\\_Device\\_Drivers](http://linuxtv.org/wiki/index.php/How_to_Obtain_Build_and_Install_V4L-DVB_Device_Drivers), to get the driver installed/updated.
- 5) Once make sure the saa7134 and saa7134\_alsa are installed properly, from root/su, unload and reload the saa7134 and saa7134\_alsa drivers, by using:

```
pulseaudio -k; rmmmod -w saa7134_alsa
modprobe -r saa7134
modprobe saa7134 card=188,188,188,188
(for the installed V4L-DVB has the Model 811/911 card ID adopted)
or
modprobe saa7134 card=73,73,73,73
( for the installed V4L-DVB has not had the Model 811/911 own
  card/device ID adopted in kernel or V4L-DVB. Temporarily,
  card=73,73,73,73 can be used for Model 810/811/911 )
or
modprobe saa7134 card=188,188,188,188,188,188,188,188
or modprobe saa7134 card=73,73,73,73,73,73,73,73
( for the users using two Model 810/811/911 boards )
```

then, exit from root/su and return to user.

- 6) Then, any one of the commonly-used live-video preview and/or A/V capturing applications, including [XawTV](#), [VLC](#), [TV-Time](#), [MPlayer](#), and [GStreamer](#), etc., can be used for preview/capturing video and audio, from the Model 810/811/911.
- 7) For the users using [XawTV](#) and with Ubuntu/Mint Linux distro, followings are the procedures and commands for Quick-Start and preview HOWTO:

```
$ sudo apt-get install xawtv
$ xawtv -c /dev/video0 &          (for Channel-1 preview)
```

```

$ xawtv -c /dev/video1 &          (for Channel-2 preview)
$ xawtv -c /dev/video2 &          (for Channel-3 preview)
$ xawtv -c /dev/video3 &          (for Channel-4 preview)
$ xawtv -c /dev/video4 &          (for Chan-1 of 2nd 810/811/911 board)
$ xawtv -c /dev/video5 &          (for Chan-2 of 2nd 810/811/911 board)
$ xawtv -c /dev/video6 &          (for Chan-3 of 2nd 810/811/911 board)
$ xawtv -c /dev/video7 &          (for Chan-4 of 2nd 810/811/911 board)
( close all XawTV windows, for next step of A/V capturing tests)

```

Using "[streamer](#)" (a companion tool with XawTV) for frame or A/V capturing:

```

$ sudo apt-get install streamer

$ streamer -h
    ( for help and list of command options )

$ streamer -c /dev/video0 -n ntsc -s 640x480 -o image1.pgm
    ( Grab gray-level image from Channel-1,
      and save the data into a .pgm file )

$ streamer -c /dev/video1 -n ntsc -s 640x480 -r 2 -t 10 \
-o imageCh2_00.jpeg
    ( Capture 10 frames from Channel-2, in JPEG format,
      and save the images into .jpeg files )

$ streamer -c /dev/video2 -n ntsc -r 30 -s 640x480 -f mjpeg \
-o video3.avi -t 0:60
    ( Capture/record video from Channel-3, in MJPEG format,
      and save the video data into a .avi file )

$ streamer -C /dev/dsp1 -R 32000 -F stereo -O audio-1.wav -t 0:03:00
    ( Capture/record audio from Audio Channel-1,
      and save the audio data into a .wav file )

$ streamer -c /dev/video2 -n ntsc -r 30 -s 640x480 -f mjpeg \
-C /dev/dsp3 -F stereo -R 32000 -o movie-ch3.avi -t 5:00
    ( Capture & record video+audio from Channel-3, in MJPEG,
      and save the A/V data into an .avi file )

$ streamer -c /dev/video3 -n ntsc -r 30 -s 352x288 -f rgb24
-C /dev/dsp4 -F stereo -R 32000 -o movie-ch4.avi -t 10:00
    ( capture and record raw video + audio from Channel-4,
      and save the A/V data into an .avi file )

```

8) For the users using [VLC](#) and with Ubuntu/Mint Linux distro, followings are the procedures and commands for Quick-Start and preview HOWTO:

```

$ sudo apt-get install vlc

$ vlc v4l2:///dev/video0:standard=NTSC:width=640:height=480 &
    (for Channel-1 preview, with VGA resolution)

$ vlc v4l2:///dev/video1:standard=NTSC:width=720:height=480 &
    (for Channel-2 preview, with D1.ntsc resolution)

$ vlc v4l2:///dev/video2:standard=NTSC:width=352:height=288 &
    (for Channel-3 preview, with CIF resolution)

```

```
$ vlc v4l2:///dev/video3:standard=ntsc --sout "#transcode{vcodec= \
mpeg4,vb=5000}:standard{access=file,dst=./v_ch4_MP4.avi}"
      (recording/transcoding from Channel-4, in D1 size,
      and save the video into an .avi file )
```

- 9) For the users using [MPlayer](#) and with Ubuntu/Mint Linux distro, followings are the procedures and commands for Quick-Start and preview HOWTO:

```
$ sudo apt-get install mplayer

$ mplayer tv:// -tv driver=v4l2:norm=NTSC:width=640:height=480:device=/dev/video0
      (for previewing Channel-1, with VGA resolution)

$ mplayer tv:// -tv driver=v4l2:norm=NTSC:width=720:height=480:device=/dev/video1
      (for previewing Channel-2, with D1.ntsc resolution)

$ mplayer tv:// -tv driver=v4l2:norm=NTSC:width=352:height=288:device=/dev/video2
      (for previewing Channel-3, with CIF resolution)

$ mplayer tv:// -tv driver=v4l2:norm=NTSC:width=352:height=240:device=/dev/video3
      (for previewing Channel-4, with SIF resolution)
```

- 10) For the users using [GStreamer](#) and with Ubuntu/Fedora Linux distro, followings are the procedures and commands for Quick-Start and preview/capture HOWTO:

```
$ gst-launch-0.10 v4l2src device=/dev/video0 ! autovideosink
      ( previewing Channel-1, with D1 resolution)

$ gst-launch-0.10 v4l2src device=/dev/video1 ! ffmpegcolorspace ! \
      video/x-raw-yuv,width=640,height=480 ! xvimagesink
      ( previewing Channel-2, with VGA resolution)

$ gst-launch-0.10 v4l2src device=/dev/video2 ! ffmpegcolorspace ! \
      video/x-raw-yuv,width=640,height=480 ! jpegenc ! \
      multifilesink location=./frame$.4d.jpeg
      ( capturing frames from Channel-3, with VGA resolution,
      and saving the frames/images into .jpeg files      )

$ gst-launch-0.10 v4l2src device=/dev/video3 ! ffmpegcolorspace ! \
      videorate ! 'video/x-raw-yuv,width=640,height=480, \
      framerate=5/1' ! clockoverlay halign=left valign=top \
      text="2011-12-08:" shaded-background=true ! jpegenc ! \
      multifilesink location=./frame$.4d.jpeg
      ( capturing frames from Channel-4, at 5 fps, with clock
      overlay, and saving the images into .jpeg files      )

$ gst-launch-0.10 v4l2src device=/dev/video0 ! ffmpegcolorspace ! \
      video/x-raw-yuv,width=640,height=480 ! clockoverlay ! \
      avimux ! filesink location=./video-raw-ch1.avi
      ( capturing raw video from Channel-1,
      and saving the video into an .avi file      )

$ gst-launch-0.10 v4l2src device=/dev/video1 ! ffmpegcolorspace ! \
      video/x-raw-yuv,width=640,height=480 ! jpegenc ! \
```

```

filesink location=./video-ch2.mjpeg
( capturing video from Channel-2, compressing in mjpeg,
  and saving the video into a .mjpeg file;
  Note: Playbackable with MPlayer directly.
        Or, with VLC, after converting using FFMPEG
        command "ffmpeg -i ./video-ch2.mjpeg -sameq
                  ./video-ch2.avi" )

```

```

$ gst-launch-0.10 v4l2src device=/dev/video2 ! ffmpegcolorspace ! \
  video/x-raw-yuv,width=640,height=480 ! clockoverlay \
  halign=left valign=top text="2011-12-08:" \
  shaded-background=true ! theoraenc quality=32 ! \
  oggmux ! filesink location=./video-ch3.ogg
( capturing video from Channel-3, with clockoverlay,
  compressing with Theora encoder, and saving
  the video into an .ogg file )

```

```

$ gst-launch-0.10 v4l2src device=/dev/video3 ! ffmpegcolorspace ! \
  video/x-raw-yuv,width=640,height=480 ! clockoverlay \
  halign=left valign=top text="2011-12-08:" ! y4menc ! \
  filesink location=./video-ch4.yuv
( capturing video from Channel-4, with clockoverlay,
  and saving the video in y4m format to an .yuv file )
  Note: Playbackable with MPlayer directly.
        Or, with VLC, after converting using FFMPEG
        command "ffmpeg -i ./video-ch4.yuv -sameq
                  ./video-ch4.avi" )

```

11) Audio capturing/recording using "[arecord](#)" and "[aplay](#)" applications:

```

$ arecord -D hw:1,0 | aplay (pre-listening, from audio channel-1)
$ arecord -D hw:2,0 | aplay (pre-listening, from audio channel-2)
$ arecord -D hw:3,0 | aplay (pre-listening, from audio channel-3)
$ arecord -D hw:4,0 | aplay (pre-listening, from audio channel-4)

$ arecord -d 30 -f cd -t wav -r 32000 -D hw:1,0 audio-raw-ch1.wav
(capturing and recording raw audio from audio channel-1,
 and saving the data into a .wav file; duration = 30sec )
$ arecord -d 300 -f cd -t wav -r 32000 -D hw:2,0 audio-raw-ch2.wav
(capturing and recording raw audio from audio channel-2,
 and saving the data into a .wav file; duration = 5min )
$ arecord -d 1800 -f cd -t wav -r 32000 -D hw:3,0 audio-raw-ch3.wav
(capturing and recording raw audio from audio channel-3,
 and saving the data into a .wav file; duration = 30min )
$ arecord -d 3600 -f cd -t wav -r 32000 -D hw:4,0 audio-raw-ch4.wav
(capturing and recording raw audio from audio channel-4,
 and saving the data into a .wav file; duration = 1 hour )

```

12) Audio capturing/recording using "[GStreamer](#)", with Ubuntu/Mint/Fedora Linux distro:

```

$ gst-launch-0.10 alsasrc device=hw:1,0 ! autoaudiosink
or
$ gst-launch-0.10 alsasrc device=hw:1,0 ! audio/x-raw-int,rate=\
  32000,channels=2,depth=16 ! audioconvert! autoaudiosink
( pre-listening, from audio Channel-1 )

```

```

$ gst-launch-0.10 alsasrc device=hw:2,0 ! wavenc ! \
    filesink location=./audio-ch2.wav
( capturing raw audio, from audio Channel-2,
  and saving the data into a .wav file )

$ gst-launch-0.10 alsasrc device=hw:3,0 ! audioconvert ! \
  vorbisenc ! oggmux ! filesink location=./audio-ch3.ogg
( capturing audio, from audio Channel-3, encoding with
  vorbis and saving the data into an .ogg file )

$ gst-launch-0.10 alsasrc device=hw:4,0 ! audio/x-raw-int,rate=\
  32000,channels=2,depth=16 ! audioconvert! avimux ! \
  filesink location=./audio-ch4.avi
( capturing raw audio, from audio Channel-4, using avi
  muxer and saving the audio data into an .avi file )

```

- 13) A/V capturing/recording using "[GStreamer](#)", with Fedora Core 12 Linux distro:

```

$ gst-launch-0.10 -v v4l2src device=/dev/video0 ! queue ! \
  ffmpegcolospace ! videoscale ! video/x-raw-yuv,\
  width=640,height=480 ! queue ! mux. \
  alsasrc device=hw:1,0 ! audio/x-raw-int,rate=32000,\
  channels=2,depth=16 ! queue ! audioconvert ! mux. \
  avimux name=mux ! filesink location=./avtest-ch1.avi
( capturing raw video+audio, from Channel-1,
  and saving the A/V data into an .avi file )

$ gst-launch-0.10 avimux name=mux ! filesink location=avcapCh2.avi \
  v4l2src device=/dev/video1 ! video/x-raw-yuv,width=320,\
  height=240,framerate=\(fraction\)30000/1001 ! \
  ffenc_mpeg4 ! queue ! mux. alsasrc device=hw:2,0 ! \
  audio/x-raw-int,rate=32000,channels=2,depth=16 !
  audioconvert ! lame ! mux.
( capturing video+audio, from Channel-2, compressing
  the A/V in MPEG4+MP3, and saving the compressed data
  into an .avi file )

```

- 14) In addition to the 3<sup>rd</sup>-party OSS tools/utilities introduced above, Sensoray provides a "Linux-811-1.0.zip" package to the Model 810/811/911 Linux customers, to help the customers get quick-started and/or follow the supplied demo/sample programs to develop their own program(s) for their particular applications.

- 15) In the package, you will find following files:

sxlxCapV4L2.c	A demo application program to demonstrate capturing raw frames from one channel on the Sensoray Model 810/811/911, using saa7134 driver and v4l2 API
avcapture.c	A demo application program to demonstrate capturing both Video+Audio from a channel on the Model 810/811/911, and save the raw A/V data into an .avi file

"Sensoray Model 810&811&911 Quick-Start Instruction -- Linux.pdf"  
This instruction manual .pdf file

16) Compile sample program, "sxlxCapV4L2", using:

```
gcc -o s811cap sxlxCapV4L2.c
```

17) Run demo: "s811cap"

In the demo program s811cap (sxlxCapV4L2), the default capturing channel is set to the Channel-1, with opening device "/dev/video0".  
If user would like to capture from different channel, simply,

```
# ./s811cap -d /dev/video0      (for Channel-1)
# ./s811cap -d /dev/video1      (for Channel-2)
# ./s811cap -d /dev/video2      (for Channel-3)
# ./s811cap -d /dev/video3      (for Channel-4)
# ./s811cap -d /dev/video4      (for Chan-1 of 2nd 810/811/911 board)
# ./s811cap -d /dev/video5      (for Chan-2 of 2nd 810/811/911 board)
# ./s811cap -d /dev/video6      (for Chan-3 of 2nd 810/811/911 board)
# ./s811cap -d /dev/video7      (for Chan-4 of 2nd 810/811/911 board)
```

18) Any one of the following image viewing/processing utilities like GNOME image viewer, Kuickshow, imageMagick, GIMP, etc., can be used to display/process the captured image saved into the output files.

19) Compile sample program, "avcapture", using:

Ubuntu:

```
su -c "apt-get install libavformat-dev libasound2-dev" (one time)
make avcapture LDLIBS="-lavformat -lavcodec -lavutil -lasound"
```

RedHat/Fedora:

```
su -c "yum install ffmpeg-devel" (one time)
make avcapture LDLIBS="-lavformat -lavcodec -lavutil -lasound \
-lpthread" CFLAGS="-I/usr/include/ffmpeg"
```

20) Run demo: "avcapture"

Usage: ./avcapture <video\_device> <output\_file> [duration] [clip\_length]

```
video_device  V4L2 device, ex: /dev/video0 for Channel-1,
               /dev/video1 for Channel-2, ....., and
               /dev/video3 for Channel-4, etc.
output_file   filename, ex: output.avi
duration      duration in seconds, or with a time suffix;
               ex: 30s 60m 2h ( default=3600s )
clip_length   optional clip_length in seconds ( default=600s )
               or with a time suffix; ex: 20s 10m 1h
               ( <30m is suggested, to insure good A/V sync for
                 all A/V clips )
```

21) The captured .avi A/V clips can be played back with the Movie Player,

VLC, MPlayer media players.

- 22) For capturing video/audio and compressing the A/V stream into a MJPEG or MPEG-1/2/4 format, using FFmpeg, Sensoray will provide an application note:

AN811.03 -- Howto Capture and Compress A/V stream into MPEG-1/2/4 or MJPEG, using [FFmpeg](#)