USB MPEG Capture Device
Software Manual (Linux)

Models 2250/2251 | Ver1.20 | September 08

SENSORAY | embedded electronics

Designed and manufactured in the U.S.A

SENSORAY | p. 503.684.8005 | email: info@SENSORAY.com | www.SENSORAY.com

7313 SW Tech Center Drive | Portland, OR 97203

TABLE OF CONTENTS

Software

Feature Summary

The 2250 uses the standard USB drivers for Linux through the libusb library. A full-featured console demo application allows recording of MPEG video on the host computer, control of the compression and video settings and text overlays.

An included C SDK can be used to integrate video capture into any application. The SDK allows maximum flexibility by providing an API for all the 2250's functions. The source code of the demo application is a suggested starting point for custom application development.

Installation

The software may be distributed on a CD or downloaded from the Sensoray's web site. If the file is downloaded, it will need to be unzipped into a folder on the local drive prior to connecting the 2250 to the USB port.

The SDK has been developed on Ubuntu LTS and support is provided for this distribution. The SDK may work on other Linux versions, but this is not guaranteed.

The library libusb must be installed. On Ubuntu, type "apt-get install libusb-dev" if it is not currently installed. On other distrubutions, install libusb-dev from the CD or otherwise.

Some distributions of Linux may not have the usb device filesystem setup. If the demo programs do not run and "lsusb" produces no output, please run "sudo mount -t usbfs none /proc/bus/usb" or try updating/installing usbutils(eg. apt-get install usbutils).

The SDK license is not GPL.

Setup is performed as follows.

1) untar the tgz file. "tar xvfz sdk-2250-linux_vXYZ.tgz" where XYZ is the version of the SDK.
2) "cd sdk-2250-linux_vXYZ" where XYZ is the version.
3) Type "make" (only needs done once to configure libraries and apps)
4) Type "cd ezloader"
5) Type "make"

6) Type "make modules_install". Installs the USB firmware loader.

7) Type "cd .." to change back to make SDK directory.

8) plug in 2250 to USB port.

9) "modprobe s2250_ezloader" loads the firmware driver. (Note: This is not required on subsequent reboots if steps 4-6 are done.)

10) "./sraydemo" runs the demo application. (On 64 bit Linux, use sraydemo64)

11) To encode for example MPEG1 Program Stream (Audio and Video) to test.mpg issue the following commands. Type "h" for a list of commands.

- "encode 0"
- "recvid test.mpg"
- wait 10 seconds
- "stop"

If a non-root user is unable to open the board while running the demo applications due to usb device permissions, create the file "41-usb-permissions.rules" in the /etc/udev/rules.d directory containing the following:

```
# USB devices

SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", \

        ATTRS{idVendor}=="1943", ATTRS{idProduct}=="2250", MODE="0666"
```

The udev daemon will automatically read this file. The next time the 2250 is plugged in, the usb device node will be created with 0666 permissions.

SDK Reference

Release Notes

V.1.20.0

- Driver moved to user space to satisfy new kernel requirements.
- Simplified installation and usage. SDK consists of s2250mid.h, libs2250mid.a, firmware loader driver. Demo programs are linked with libs2250mid.a and include the file s2250mid.h. (64 bit Linux uses libs2250mid64.a)
- Polling and Blocking capability through SN_Polling, SN_Blocking
- SN_GetAudioData modified. SN_GetAudioDataPlusTime added.
- SN_SetGopSize, SN_SetSeqMode added(see instructions before using).

V.1.0.0:

- Multiple boards supported. Specify board number 0 for single board use. Use SN_GetNumBoards to get number of boards detected in system.
- Example "C" console program for multiple boards.
- Example jpg capture demo.

General Notes

The common API flow is described below. Refer to the complete functions reference for the details on individual functions.

1. Initialization. This is performed by a call to `SN_OpenBoard()` function. Initial default capture settings are loaded.
2. A call to `SN_OpenBoard()` may be optionally followed by calls to the functions controlling various settings:
    - input source: `SN_SetVideoSource();`
    - video system and geometry: `SN_SetVideoSource()`
    - `audio source: SN_SetAudioSource();`
    - basic settings and compression mode `SN_SetBasicChipSettings()`

- video parameters (brightness, contrast, saturation, hue): `SN_SetContrast(), SN_SetBrightness(), SN_SetHue, SN_SetSaturation();`
- OSD: `SN_SetCaption(), SN_ClearCaption()`.

3. A call to `SN_StartStream()` starts the 2250. The stream received from the USB is decoded and displayed in the user window specified with `SN_Open()`.

4. If recording is required, video and audio frames can be retrieved with SN_GetAudioData() and SN_GetOneFrame(). The demo application, sraydemo, shows a complete working example.

5. `SN_Close() and SN_StopStream()(if streaming)` must be called before the application terminates to clean up properly.

Demo applications

The SDK includes two demo applications provided with the source code to illustrate the use of SDK's functions. The SDK functions are in the file s2250mid.h

sraydemo: console application for control of one or more 2250's. Demonstrate mpeg recording, snapshots, video settings and more.

jpegdemo: very simple program demonstrating JPEG snapshot retrieval to file. For more advanced JPEG snapshots with standard JPEG quality control, please consider using the 2255 device.

Functions Reference

All API functions are declared as C functions for maximum portability.

```
int SN_OpenBoard( int boardnum );
```

Must be called before any other API functions are called to open 2250 board.

Parameters

Boardnum(0 for first instance).

Returns

0 on success, negative value if error.

```
 int SN_CloseBoard( int boardnum
);
```

Must be called before application terminates for proper clean-up of the SDK and board module.

Parameters

boardnum.

Returns

0 on success, negative value if error.

```
int SN_StartStream(
    int boardnum
);
```

Start video/audio streaming.

boardnum

board number in the system (use 0 for single board setups).

```
int SN_StopStream(
    int boardnum
);
```

Stop video streaming.

```
int SN_SetVideoSource(
   int boardnum,
   int source
);
```

Selects between composite and S-video inputs.

Parameters

source

0 – SVIDEO

1 – Composite video

Returns

0 on success, negative value if error

```
int SN_SetAudioSource(
   int boardnum,
   int source
);
```

Selects between Line and Mic inputs.

Parameters

source

0 – Line input

1 – Microphone input

2 – Microphone input with gain(boost)

Returns

0 on success, negative value if error

```
int SN_SetVideoSystem(
   int boardnum
   int videosystem
);
```

Sets the video system(PAL or NTSC). Will not change the video size. To change video size to new system, please also call SN_SetBasicChipSettings after SN_SetVideoSystem.

Parameters

boardnum

pointer to the value of current input.

videosystem

0- NTSC

1- PAL

2- Returns

0 on success, negative value if error.

```
int SN_SetBasicChipSettingss(
   int boardnum,
   EVideoFormat encodetype,
   int iVidSize,
   MID2250_VIDSYS vidsys,
   int bitrate,
   int bIFramesOnly
);
```

Sets the primary chip settings

Parameters

boardnum

board instance in the system.

encodetype

Compression type MPEG1, MPEG2, MPEG4 or MOTIONJPEG.

Returns

0 on success, negative value if error (see mid2250types.h for error codes list).

```
int SN_SetContrast(
   int boardnum,
   int contrast
);
```

Sets the contrast(0-100)

Parameters

boardnum

board instance in system.

contrast

Video contrast 0-100(50 is default).

Returns

0 on success, negative value if error(see s2250mid.h)

```
int SN_SetHue(
  int boardnum,
  int hue
);
```

Sets the hue (-50 to 50).

Parameters

boardnum

board instance in system.

hue

Video hue -50-50(0 is default).

Returns

0 on success, negative value if error(see s2250mid.h)

```
int SN_SetSaturation(
  int boardnum,
  int saturation
);
```

Sets the saturation(0-100)

Parameters

boardnum

board instance in system.

saturation

Video saturation 0-100(50 is default).

Returns

0 on success, negative value if error

```
int SN_SetBrightness(
  int boardnum,
  int brightness
);
```

Sets the brightness(0-100)

Parameters

boardnum

board instance in system.

brightness

Video brightness 0-100(50 is default).

Returns

0 on success, negative value if error(see s2250mid.h)

```
int SN_SetCaption(
   int boardnum,
   int xstart,
   int ystart,
   char *text
);
```

Sets the text caption.

Parameters

boardnum

board instance in system

xstart

x position in pixels. Position is rounded to the nearest 16x16 pixel block.

Ystart

y position in pixels. Position is rounded to the nearest 16x16 pixel block.

text

text to display (a maximum of 96 characters total for all overlay windows).

Returns

0 on success

```
int SN_ClearCaption(
    int board
);
```

Clears all captions/overlays.

Parameters

board

board number in the system (use 0 for single board setups).

Returns

0 on success, negative value if error

```
int SN_GetOneFrame(
   int board,
   UINT8 *pBuf,
   int buflen,
   frame_info_t *pFI
);
```

Gets one video frame(JPEG if MOTIONJPEG, I,P,or B Frame if MPEG video).

Parameters

board

board number in the system (use 0 for single board setups).

p*Buf*

pointer to buffer to get frame

buflen

length of buffer

pFI

pointer to frameinfo structure(see s2250mid.h for definition)

Returns

0 on success, negative value if error

```
int SN_GetAudioData(
   int board,
   unsigned char *pBuf,
   int*buflen
);
```

Gets audio samples. Call with buflen set to size of sample requested. Returns actual size of samples. See sraydemo.c for an example of SN_GetAudioData.

Parameters

board

board number in the system (use 0 for single board setups).

p*Buf*

pointer to buffer to get samples

buflen

length of buffer

Returns

0 on success, negative value if error

```
int SN_GetAudioDataPlusTime(
    int board,
    UINT8 *pBuf,
    SINT32*buflen,
    int *tv_sec,
    int *tv_usec,
);
```

Gets audio samples plus timestamp. Call with buflen set to size of sample requested. Returns actual size of samples. Also returns audio timestamp in tv_sec and tv_usec.

Parameters

board

board number in the system (use 0 for single board setups).

p*Buf*

pointer to buffer to get samples

buflen

length of buffer

tv_sec

returned timestamp in seconds

tv_usec

returned microsecond component of timestamp

Returns

0 on success, negative value if error

```
int SN_Polling(
    int board)
);
```

Sets the SDK to return immediately on calls to SN_GetAudioData and SN_GetOneFrame.

Parameters

board

board number in the system (use 0 for single board setups).

Returns

0 on success, negative value if error

```
int SN_Blocking(
    int board)
);
```

Sets the SDK to return block(up to one second) on calls to SN_GetAudioData and SN_GetOneFrame.

Parameters

board

board number in the system (use 0 for single board setups).

Returns

0 on success, negative value if error

```
int SN_SetSeqMode(
    int board,
    int seqmode)
);
```

Advanced use only. Sensoray is not responsible for invalid combinations of sequence mode and gop size(see SN_SetGopSize). SN_SetSeqMode will try to determine if the setting is valid and return -EINVAL if the sequence is invalid. The following are allowed settings of sequence mode and gop size.

1) If you change seq_mode = 2, you must set GOP_SIZE to be from 2-15

2) If you change seq_mode = 3, you must set GOP_SIZE to a multiple of 3 up to 15 for MPEG1/MPEG2, and up to 30 for MPEG4_DIVX.

3) If you change seq_mode =1(IFrames only), you should leave GOP_SIZE unchanged. Alternatively, IFrame only mode can be set through SN_SetBasicChipSettings.

Parameters

board

board number in the system (use 0 for single board setups).

seq_mode

Sequence mode. GOP size must be valid for the given sequence mode. If unsure, please do not change the sequence mode or GOP size.

1 = IFrames only (can be set using SN_SetBasicChipSettings)

2 = IP

3 = IPB

Returns

0 on success, negative value if error

```
int SN_SetGopSize(
    int board,
    int gop_size)
);
```

Advanced use only. It is recommended to leave this setting alone. Sensoray is not responsible for invalid combinations of sequence mode and gop size(see SN_SetGopSize). Please see constraints in SN_SetSeqMode to see allowed combinations. If you just want I frames, please see SN_SetBasicChipSettings

Parameters

board

board number in the system (use 0 for single board setups).

gop_size

Size of the group of picture. 1-15 (MPEG1/MPEG2/MPEG4) 1-30 (MPEG4DIVX).

GOP size must be valid for the given sequence mode. If unsure, do not change the GOP size.

Returns

0 on success, negative value if error

```
int SN_SetHeight(
    int board,
    int height)
);
```

Advanced use only. Changes encoded height. Call this function(if desired) after setting SN_SetBasicChipSettings and before starting the stream. Only certain values of height and width are supported. If unsure, please use SN_SetBasicChipSettings to change from full size to half size.

Parameters

board

board number in the system (use 0 for single board setups).

height

supported values:

height=240, width=320 (must use half-size in SN_SetBasicChipSettings)

height=480;width=640 (must use full size SN_SetBasicChipSettings)

height=480;width=704 (NTSC only)

height=480;width=720 (NTSC only)

height=576;width=720 (PAL only)

height=576;width=704 (PAL only)

height=576;width=720 (PAL only)

Returns

0 on success, negative value if error

```
int SN_SetHeight(
    int board,
    int height)
);
```

Advanced use only. Changes encoded Width. Call this function(if desired) after setting SN_SetBasicChipSettings and before starting the stream. Only certain values of height and width are supported. If unsure, please use SN_SetBasicChipSettings to change from full size to half size.

Parameters

board

board number in the system (use 0 for single board setups).

height

supported values:

height=240, width=320 (must use half-size in SN_SetBasicChipSettings)

height=480;width=640 (must use full size SN_SetBasicChipSettings)

height=480;width=704 (NTSC only)

height=480;width=720 (NTSC only)

height=576;width=720 (PAL only)

height=576;width=704 (PAL only)

height=576;width=720 (PAL only)

Returns

0 on success, negative value if error

```
int SN_GetHeight(
    int board,
    int *height)
);
```

Advanced use only. Return encoded height.

Parameters

board

board number in the system (use 0 for single board setups).

height

returned height

Returns

0 on success, negative value if error

```
int SN_GetWidth(
    int board,
    int *width)
);
```

Advanced use only. Return encoded width.

Parameters

board

board number in the system (use 0 for single board setups).

width

returned width

Returns

0 on success, negative value if error

```
int SN_Debug(
    int debug_level)
);
```

Advanced use only. debug_level = 0(default), quiet SDK. debug_level=1(debug messages in SDK for debugging ONLY).

Parameters

debug_level

0 quiet. 1 debug messages

Returns

0 on success, negative value if error