

PCI express
Capture Device

Model 811

Windows SDK User's
Manual

December, 2009

© Sensoray 2009
7313 SW Tech Center Dr.
Tigard, OR 97223
Phone 503.684.8073 • Fax 503.684.8164
sales@sensoray.com
www.sensoray.com



Table of Contents

LIMITED WARRANTY.....	4
INTRODUCTION.....	5
Software Feature Summary.....	5
SOFTWARE.....	6
Feature Summary.....	6
Installation.....	6
Redistribution.....	6
SDK Reference.....	7
Release Notes.....	7
General Notes.....	7
Demo applications.....	8
Functions Reference.....	8

Limited warranty

Sensoray Company, Incorporated (Sensoray) warrants the hardware to be free from defects in material and workmanship and perform to applicable published Sensoray specifications for two years from the date of shipment to purchaser. Sensoray will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The warranty provided herein does not cover equipment subjected to abuse, misuse, accident, alteration, neglect, or unauthorized repair or installation. Sensoray shall have the right of final determination as to the existence and cause of defect.

As for items repaired or replaced under warranty, the warranty shall continue in effect for the remainder of the original warranty period, or for ninety days following date of shipment by Sensoray of the repaired or replaced part, whichever period is longer.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. Sensoray will pay the shipping costs of returning to the owner parts that are covered by warranty. A restocking charge of 25% of the product purchase price, or \$105, whichever is less, will be charged for returning a product to stock.

Sensoray believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, Sensoray reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult Sensoray if errors are suspected. In no event shall Sensoray be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, SENSORAY MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF SENSORAY SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. SENSORAY WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEROF.

Third party brands, names and trademarks are the property of their respective owners.

Introduction

The 811 product is a PCI express video and audio capture device. The 811 product is powered through the PCI express bus and does not require any external power supplies.

Software Feature Summary

- 1 Composite or 1 S-video input.
 - Audio capture
 - Free Windows driver(DirectShow), demo application and Software Development Kit (SDK).
-

Software

Feature Summary

Sensoray's Model 811 is shipped with drivers for Microsoft Windows XP, Vista, Windows 7. A full-featured demo application allows preview and saving of captured video on the host computer, control of the video settings, and snapshot capture.

The included SDK can be used to integrate video capture into any application. The SDK allows maximum flexibility by providing an API for all the 811's functions without knowledge of DirectShow programming. The source code of the demo application is a suggested starting point for custom application development. The DLL source code is available and included in the SDK. Email support@sensoray.com if you require the password.

Since the 811 has an AVStream driver, it is DirectShow compliant. Users familiar with DirectShow programming may use an existing DirectShow program(AMCap) or write their own. Both options (DLL and native DirectShow) are available for maximum flexibility.

Installation

The software may be distributed on a CD or downloaded from the Sensoray's web site.

Run the setup program from the distribution disk or folder. Software components, including a demo application with the source code, will be installed into the /Program Files/Sensoray/811 folder.

During the installation the program will ask whether to install drivers. Select Yes and follow the prompts.

Redistribution

The SDK CD contains the redistributables in the API directory. The midplayback.dll is an extra DLL for playback only. It is provided as a convenience only and users may choose to write their own playback functions or not use playback in their application.

The drivers must also be redistributed to end-users and installed for proper function. They are included in the drivers directory.

SDK Reference

Release Notes

V.1.0.0 (December 1, 2009):

- Initial release

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

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 `S811_Open()` function. Initial default capture settings are loaded.
2. A call to `S811_Open()` may be optionally followed by calls to the functions controlling various settings:
 - input source: `S811_SetSource();`
 - video preview window: `S811_SetPreviewWindow();`
 - video system and geometry: `S811_SetVidSys(), S811_SetVidSize();`
 - video parameters (brightness, contrast, saturation, hue): `S811_SetLevels();`
 - record mode: `S811_SetRecordMode();`
3. A call to `S811_StartPreview()` starts the 811 for preview only. The stream received from the PCI bus is decoded and displayed in the user window specified with `S811_SetPreviewWindow()`.
4. If recording is required, the stream should be started with a call to `S811_StartRecord`, and stopped with a call to `S811_StopRecord()`.
5. During the recording the following function could be used to obtain some useful information:

- `S811_GetStatus()` – current status (record, playback), current recorded file size and path;

6. `S811_Close()` must be called before application terminates to clean up properly.

Demo applications

The SDK includes a demo application provided with the source code to illustrate the use of SDK's functions.

code – an MFC application. Displays the stream in the window, allows modification of board parameters and allows preview and recording to the hard drive.

Functions Reference

All API functions are declared using the following definition and the **__stdcall calling convention**:

```
#define MID811_API extern "C" __declspec(dllimport)
```

```
MID811_API int S811_Open( void );
```

Must be called before any other API functions are called to open the SDK.

Parameters

None.

Returns

0 on success, negative value if error (see `mid811types.h` for error codes list).

```
MID811_API int S811_Close(
);
```

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

Parameters

None.

Returns

0 on success, negative value if error (see `mid811types.h` for error codes list).

```
MID811_API int S811_GetNumDevices (
```

```
int *pNumDevices
);
```

Retrieve the number of devices in the system. 4 devices per board.

Parameters

pNumDevices

Returns the number of devices(Note: 4 channels/devices per 811 board).

Returns

0 if success, negative if error.

```
MID811_API int S811_SetSource (
    MID811_SOURCE input,
    int devid
);
```

Selects between composite and S-video inputs.

Parameters

input

enumerated input MID811_SOURCE (see mid811types.h). For model 314 the allowed values are MID811_SOURCE_COMPOSITE_0, MID811_SOURCE_COMPOSITE_1, MID811_SOURCE_COMPOSITE_2, MID811_SOURCE_COMPOSITE_3, MID811_SOURCE_SVIDEO_0 and MID811_SOURCE_SVIDEO_1.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetSource (
    MID811_SOURCE *pSource,
    int devid
);
```

Retrieves current input settings.

Parameters

pSource

pointer to the value of current input.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetVidSys (  
    MID811_VIDSYS vidsys,  
    int devid  
);
```

Sets the input video system (NTSC, PAL).

Parameters

vidsys

video system enumerated type (see mid811types.h).

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetVidSys (  
    MID811_VIDSYS *pvidsys,  
    int devid  
);
```

Gets the input video system (NTSC, PAL).

Parameters

pvidsys

pointer to video system enumerated type (see mid811types.h).

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetStatus(  
    MID811STATUS *pStatus,  
    int devid  
);
```

Retrieves current status information (see MID811func.h for MID811STATUS type definition).
Multibyte (ASCII) filenames.

Parameters

pStatus

pointer to status variable.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetStatusW(  
    MID811STATUS_W *pStatus,  
    int devid  
);
```

Same as S811_GetStatus, but uses Unicode filenames.

```
MID811_API int S811_StartPreview(  
    int devid  
);
```

Starts video and audio streaming for preview only.

Parameters

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_StopPreview(  
    int devid  
);
```

Stops video and audio streaming for preview.

Parameters

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetRecordMode (  
    MID811_REC recMode,  
    int devid  
);
```

Sets the record mode (see mid811types.h for MID811_REC type definition).

Parameters

recMode

record mode.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_StartRecord (  
    char *fileName,  
    int devid  
);
```

Starts recording to a file.

Parameters

fileName

full path to the target file, no extension.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_StartRecordW(  
    wchar_t *fileName,  
    int devid  
);
```

Same as S811_StartRecord, but uses widechar or Unicode filenames.

```
MID811_API int S811_StopRecord(  
    int devid  
);
```

Stops the recording.

Parameters

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_PauseRecord(  
    int devid  
);
```

Pauses the recording. Preview will also be paused if viewing video.

Parameters

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_ResumeRecord(  
    int devid  
);
```

Resumes the recording. Preview will also be resumed if viewing video.

Parameters

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetLevels (  
    int param,  
    int value,  
    int devid  
);
```

Sets brightness, contrast, saturation and hue of the captured video.

Parameters

param

defines the parameter to set (MID811_LEVEL_CONTRAST,

MID811_LEVEL_BRIGHTNESS, MID811_LEVEL_SATURATION, MID811_LEVEL_HUE).

See see mid811types.h for definitions.

value

defines the value of selected parameter

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetLevels (  
    int param,  
    int value,  
    int devid  
);
```

Retrieves video levels.

Parameters

param

defines the parameter to get (MID811_LEVEL_CONTRAST,
MID811_LEVEL_BRIGHTNESS, MID811_LEVEL_SATURATION, MID811_LEVEL_HUE).

See see mid811types.h for definitions.

value

pointer to returned value of selected parameter

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetMute(  
    BOOL bMute,  
    int devid  
);
```

Mutes audio on the host computer. Audio will still be recorded.

Parameters

bMute

TRUE to mute the audio.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetMute(  
    BOOL *bMute,  
    int devid  
);
```

Retrieves current audio playback setting.

Parameters

bMute

pointer to retrieved setting.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetVidSize(  
    MID811_RES eVidSize,  
    int devid  
);
```

Sets the captured video size (resolution).

Parameters

eVidSize

see header file for supported resolutions.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetVidSize(  
    int *eVidSize,  
    int devid  
);
```

Gets the captured video size (resolution).

Parameters

eVidSize

returns pointer to vid size.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetRenderVideo(  
    BOOL bDisplayVideo,  
    int devid  
);
```

Turns the preview on and off.

Parameters

bDisplayVideo

TRUE to enable preview.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetRenderVideo(  
    BOOL *bDisplayVideo,  
    int devid  
);
```

Retrieves the preview setting.

Parameters

bDisplayVideo

pointer to retrieved setting.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_Repaint(  

```

```
HDC hdc
);
```

Repaint callback, should be called when application receives a WM_PAINT event. This is necessary to notify the video renderer of a repaint event.

Parameters

hdc

device context handle or NULL if unknown.

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_DisplayChange (
);
```

Informs the video renderer that a WM_DISPLAYCHANGE message has been received by the application. This callback should be called when the application has a WM_DISPLAYCHANGE event.

Parameters

none

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetPreviewWindow (
    HWND hwnd,
    int devid,
);
```

Sets the window for video streaming. Overrides S811_SetRenderVideo setting to TRUE.

Parameters

hwnd

Handle to video display window.

devid

current device id.

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```

MID811_API int S811_SetVideoPosition (
    int xpos,
    int ypos,
    int xsize,
    int ysize,
    int left_clip,
    int top_clip,
    int right_clip,
    int bottom_clip,
    int devid
);

```

Sets the video position in the clipping window.

Parameters

xpos, ypos

x and y coordinates of the top left corner.

xsize, ysize

horizontal and vertical sizes of video display.

left_clip, top_clip, right_clip, bottom_clip

number of pixels to clip of the left, top, right and bottom sides of source video.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```

MID811_API int S811_EnableSnapshots (
    BOOL bOnOff,
    int devid,
);

```

Enables snapshot capture filter. This will increase CPU usage, but it is required to use the S811_GetFrame and S811_SnapshotToFile functions. Must be called before starting the stream(S811_StartRecord or S811_StartPreview).

Parameters

bOnOff

control value to turn snapshots on/off.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_SetColorSpace(  
    MID811_COLORSPACE colorspace,  
    int devid,  
);
```

Changes default colorspace mode. Setting depends on capture mode. For display, YUY2(default is the most efficient). If capturing BMPs, it may be better, depending on the application, to set to RGB24 to capture the images without a SW transform.

Parameters

colorspace

MID811_COLORSPACE_YUY2, COLORSPACE_RGB32, COLORSPACE_RGB24, COLORSPACE_UYVY. S811_SnapshotToFile requires YUY2 or RGB24.

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

0 on success, negative value if error (see mid811types.h for error codes list).

```
MID811_API int S811_GetFrame(  
    long inSize,  
    unsigned char *pFrame,  
    BITMAPINFOHEADER *pBMI,  
    int devid  
);
```

S811_GetFrame retrieves current frame for a streaming graph. Snapshots must be enabled.

Parameters

inSize

size of pFrame buffer supplied (should be at least 768*576*3). If too small, an error will be returned.

pFrame

pointer to buffer of size inSize where frame is stored

pBMI

pointer to bitmap info.

devid

device id in system.

Returns

0 if success, negative if error.

```
MID811_API int S811_SnapshotToFile(  
    char *fileName,  
    int filetype,  
    int qual,  
    int devid  
);
```

Takes a snapshot and saves it to a file. Please see S811_EnableSnapshots and S811_SetColorspace.

Parameters

fileName

full path to the target file, no extension.

filetype

file type(s) to save to (MID811_FILE_JPEG, MID811_FILE_BMP, see mid811types.h).

Please note that jpegs are compressed on the host computer, not on the 811 card. BMPs capture will be less CPU intensive.

qual

JPEG quality (25-100, higher value yields better quality and bigger files).

devid

device id in the system (use 0 for first channel with a single board installed).

Returns

Image size in bytes or a negative value in case of an error.

```
MID811_API int S811_SnapshotToFileW(  
    wchar_t *fileName,  
    int filetype,  
    int qual,  
    int devid  
);
```

Same as S811_SnapshotToFile, but uses Unicode filenames

```
MID811_API int S811_GetVideoSize(  
    RECT *prect,  
    int devid,  
);
```

Retrieves the current video rectangle size.

Parameters

prect

Retrieved pointer to rectangle.

devid

current device id.

Returns

0 on success, negative value if error (see mid811types.h for error codes list).