

H.264 720P/1080i Capture Device Software Manual (Windows)

Models 2226 | Ver1.01 | August 2011

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

LIMITED WARRANTY.....	3
INTRODUCTION.....	4
Software Feature Summary.....	4
SOFTWARE.....	5
Feature Summary.....	5
Installation.....	5
Redistribution.....	5
SDK Reference.....	6
Release Notes.....	6
General SDK Usage.....	6
Demo applications.....	7
Function Summary.....	7
Initialization.....	7
Recording.....	8
Stream Capture.....	9
Preview.....	9
Decoding.....	10
Overlay.....	11
Snapshots.....	11
Notifications.....	12
Cleanup/Shutdown.....	12
Functions Reference.....	12
Demo Application.....	48
Board Selection.....	48
Input.....	48
Bitrate and Audio.....	49
Levels.....	49
Preview.....	49
Record from 2226.....	49
Overlay.....	49
Streaming.....	49
Playback on 2226.....	49
Snapshot.....	49
Playback on Media Player.....	49
FAQ.....	49

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 2226 product is a USB 2.0 audio video H.264 capture device. The 2226 supports many different inputs and video format including 1080i and 720p.

Software Feature Summary

- Outputs MPEG Transport stream with H.264 MPEG encoding.
 - Decodes 2226 H264 streams back through hardware decoder.
 - Source code for demo provided.
 - Full featured demo application including recording and UDP streaming of the stream.
 - Free Windows driver(AVStream/DirectShow), 3 demo applications and Software Development Kit (SDK).
 - VB.NET demo application, C# (C Sharp) WPF demo application, MFC C++ demo application.
-

Software

Feature Summary

Sensoray's Model 2226 is shipped with drivers for Microsoft Windows XP, Vista, Windows 7. A full-featured demo application demonstrate capture of the audio/video stream. Control is provided for brightness, bitrate, contrast and other video attributes. Multiple inputs are supported including 480i(NTSC), 576i(PAL), 1080i(@59.94Hz), 1080i(@60Hz), 1080i(@50 Hz PAL), 720P(@59.94 Hz), 720(@60Hz). HDMI is not supported. HD inputs are SDI format.

The SDK allows maximum flexibility by providing an API for all the 2226's functions. The source code of the demo application is a suggested starting point for custom application development.

Since the 2226 has an AVStream driver, it is DirectShow compliant. Unfortunately, due to the wide number of DirectShow programs, Sensoray cannot guarantee operation with any specific third party program. Please note that CPU usage while previewing(decoding) the stream will be very high due to the high compression of the H.264 stream.

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/2226 folder.

During the installation the program will pre-install the drivers using Dpinst. Do not click cancel when the driver. Do not plug or unplug the board during the driver installation process.

Redistribution

The SDK CD contains the redistributable targets in the API directory.

The drivers must also be redistributed to end-users and installed for proper function. They are included in the drivers directory after the SDK (setup.exe) is installed.

The DLL uses the Microsoft Visual C++ 2008 runtime. Sensoray's installer will install the C++ runtime files automatically. For customers creating their own redistribution package, the run time package may be downloaded from Microsoft's website.

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9b2da534-3e03-4391-8a4d-074b9f2bc1bf&displaylang=en>

The files PushFileSource2.ax and sraywrite.ax are registered by the installer using regsvr32. If re-distributing using a custom install, these files must be registered on the target computer.

The DLLs `alfont.dll`, `alleg42.dll`, `mid2226.dll` and `ovlgen.dll` should be included in the same directory as the executable. It is not recommended by Microsoft to install DLLs system wide (in `c:\windows\system32`).

SDK Reference

Release Notes

V1.0.1 (August 18, 2011):

- New firmware release. Please download firmware updater from website. Firmware updater is a standalone program. Firmware fixes known issues on V1.0.0 with some SDI cameras. Firmware update is especially important if using the 59.94Hz frame-rate.
- Overlay update speed improved.
- Down-scaling capability on composite outputs. If an SDI HD input is connected, it is possible to downscale in a viewable format on the composite output. Please see `S2226_SetOutput`.
- Raw preview option on host computer at downscaled resolution. (See `S2226_SetPreviewType`).
- Callback functionality expanded. Callbacks may be used in Preview, Record and Record with Preview streaming as well as standalone callback operation(`S2226_StartCallback`). See `S2226_RegisterCallback`. Raw capture callback also possible (`S2226_RegisterCallbackRaw`).
- Minor bug fixes and new functions.

V.1.0.0 (October 26, 2010):

- Initial release
- V1.0.0 contains decoded preview only. This feature is only available on Windows 7 because XP does not contain the H264 codecs. Please see the section on Preview in the SDK function summary.

A common API flow is described below. Please refer to the Functions Reference and Function summary sections in the manual for more the details on the functions below.

General SDK Usage

1. Enumeration. If using multiple boards, they can be identified with the `S2226_Enumerate` call. If using a single board, it may be opened without enumeration using the index 0.

2. Initialization. This is performed by a call to `S2226_OpenBoard()` function with the board index parameter. Initial default capture settings are loaded and a handle to the board for other functions is returned.
3. A call to `S2226_OpenBoard()` should be followed by calls to the functions controlling various settings. At a minimum, the correct video input, video system, and clock rate(for HD only) should be set:
 - input source: `S2226_SetInput()`;
 - video system: `S2226_SetVidSys()`, `S2226_SetVidSize()`;
 - advanced video system(for 59.94 HZ NTSC): `S2226_SetReducedClock()`;
 - video parameters (brightness, contrast, saturation, hue): `S2226_SetLevels()`;
4. A call to `S2226_StartRecord()` starts the 2226 and records to file. `S2226_StopStream()` stops recording from the 2226. The demo application also shows how to capture the data for other purposes using the callback feature. Do not unplug the input signal before `S2226_StopStream()` is called.
5. During the recording the following function could optionally be used to obtain some useful information.
 - `S2226_GetStatus()` – current status, current recorded file size and path;
6. `S2226_CloseBoard()` must be called before application terminates to clean up properly and close the board handle.

Demo applications

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

Function Summary

Initialization

Initialization is done by opening the board. If unsure what board to open (in the case of multiple boards), use the enumerate function `S2226_Enumerate` to find detect the number of boards and associate each board index with a serial number. If using a single board, it is possible just to open it with `S2226_OpenBoard`. Before any other function in the SDK can be used (except `S2226_Enumerate`), the board must be opened with `S2226_OpenBoard`. This function gives a handle to the board, which is used for the other functions. Only one instance of the board should be opened at one time.

After the board is successfully opened, it is necessary to configure it. Generally, it is not necessary to configure the audio settings. The default audio settings are usually adequate. If required, the audio may be adjusted more easily by temporarily changing the audio route to bypassed using `S2226_SetAudioRoute`. This mode passes through the audio from the input to the output audio connector.

The input, clock(if HD), and video system must be correctly configured before any encoding function such as S2226_StartRecord is started. Please see the function reference for a detailed description of these functions.

Encoding

- S2226_SetBitrate

Video

- S2226_SetVidSys, S2226_GetVidSys
- S2226_SetInput, S2226_GetInput
- S2226_SetReducedClock (HD only), S2226_GetReducedClock (HD only)
- S2226_SetLevel, S2226_GetLevel

Audio

- S2226_SetAudioAgc, S2226_GetAudioAgc
- S2226_SetAudioGain, S2226_GetAudioGain
- S2226_SetAudioBalanced, S2226_GetAudioBalanced
- S2226_SetAudioOutputVol, S2226_GetAudioOutputVol
- S2226_SetAudioOutputMono, S2226_GetAudioOutputMono
- S2226_SetAudioOutputStereo, S2226_GetAudioOutputStereo
- S2226_SetAudioOutputHp, S2226_GetAudioOutputHp
- S2226_SetAudioRoute, S2226_GetAudioRoute

Recording

There are two functions to record the stream to file. One version with preview and one without. Please note that Windows7 does not allow recording to the root drive or in the "Program Files" directory unless the application is run as administrator.

The functions for recording are shown below. If using S2226_StartPreviewAndRecord or S2226_StartPreviewAndRecordW, please read the section on preview. Before calling any of these functions, a valid input must be connected to the 2226 and the configuration settings properly configured. The 2226 does not support switching inputs while streaming. Do not unplug the input after recording has started.

- S2226_StartRecord, S2226_StartRecordW(unicode)

- S2226_StartRecordAndPreview, S2226_StartRecordAndPreviewW
- S2226_StopStream

Stream Capture

Stream capture is the process of capturing the encoded H.264 stream to memory. This is done by a callback mechanism. The callback must not block and should return in a timely manner. If the callback does not return fast enough, data will be lost and the stream could be corrupted. If a lot of processing must be done on the data during the callback, it is recommended to use standard software Engineering techniques to work with the data. This could involve, for example, saving the data from the callback to a queue or FIFO to work on in another thread.

An example of stream capture using callbacks is shown in the demo application. It captures the stream and sends it out on a UDP socket. Because the 2226 captures encoded transport stream, no processing or headers need added. The resulting stream can be viewed on the VideoLan media player by opening a network stream with arguments such as `udp://@:1234` (for UDP destination port 1234).

The functions associated with stream capture through callbacks are shown below. The 2226 does not support switching inputs while streaming. Do not unplug the input after streaming has started.

In version 1.0.1, callbacks can be registered for any streaming function such as recording. For instance, you can set a callback with `S2226_RegisterCallback` and then call `S2226_StartRecord`. This will record the stream to file and give a callback to the data. After `S2226_StopStream` is called, the callback is unregistered.

Also in version 1.0.1 is a callback to the raw preview stream. This is available by using the function `S2226_RegisterCallbackRaw`.

- S2226_RegisterCallback
- S2226_RegisterCallbackRaw
- S2226_StartCallback (callback stream only)
- S2226_StopStream

Preview

Preview is the display of the stream on a host computer or PC. The 2226 also has output channels to display the connected input on an external monitor, but these are considered a different feature.

Version 1.0.1 has a raw preview feature on the host computer (at reduced resolution to meet USB bandwidth requirements). This will involve a firmware update, installable locally. Raw preview is at 320x240 resolution and can be set using `S2226_SetPreviewType`. Raw preview is multiplexed with down-scaled output in the hardware. Only one feature is available at a time, either raw preview or down-scaled composite output. If using raw preview, you must disable downscaled output (`S2226_SetOutputType`).

Due to codec issues, decoded preview is only available on Windows 7. Additionally, the following caveats apply to Preview with Recording (S2226_StartPreviewAndRecord). DirectShow has a known issue where the video preview is restarted when the Window is moved from one monitor to another. This is normally not a problem, but if the underlying stream graph is also recording the stream, data will be lost from the recorded file.

There is a workaround to the above problem associated with S2226_StartPreviewAndRecord. The solution is to prevent the user from moving the application Window from one monitor to another while S2226_StartPreviewAndRecord is running. An example is shown in the VB.NET demo. Please note if the stream is stopped, moving the application from one monitor to another is perfectly acceptable.

Additionally, Sensoray has decided that loss of recording data is much more serious than loss of preview on the PC. Therefore, Sensoray has overridden the behavior of DirectShow to prevent this from happen.

Unfortunately, if the user does manage to move the video window from one display to another, the preview may freeze. When this happens, the recording will continue (file size will still increase) but the user may be alarmed by the loss of preview. This at least gives the option of stopping the stream and restarting with a new file without losing recorded data. It is best to prevent this in the first place by using the workaround to prevent the video preview from changing monitors.

H264 is a highly compressed format. When using a slower PC, high CPU usage and preview stuttering while previewing the stream may occur. In this case, it is best to use an external monitor and the 2226 hardware outputs to preview the stream.

Version 1.0.1 contains raw preview which is less CPU intensive. This eliminates the multi-monitor issue above at the tradeoff of reduced resolution.

Version 1.0.1 also allows callbacks during preview.

The following functions described later in the Functions Reference are associated with Preview. The 2226 does not support switching inputs while streaming. Do not unplug the input after streaming has started.

- S2226_StartPreview
- S2226_StartPreviewAndRecord, S2226_StartPreviewAndRecordW
- S2226_StopStream

Decoding

The 2226 has the capability to decode streams recorded by the same hardware. It is a closed decoder in that other H.264 streams not recorded by the 2226 codec may not be decoded successfully. The decoder must know the format (video size and clock) of the stream. It does not have the capability to auto-detect the stream format parameters. Please see the function reference for the functions below:

- S2226_StartDecode, S2226_StartDecodeW
- S2226_StopStream

Overlay

The 2226 has the capability to add text captions and bitmap images on to the video stream. There are a total of 8 caption windows to work with. Please note that multi-line captions may take up multiple windows. Overlay is supported for encoding only. It is not supported for decoding. The demo application currently only demonstrates one overlay region. Also, the overlay is not updated until S2226_UpdateOverlay is called. If using multiple overlays, set them up first and then call S2226_UpdateOverlay for maximum efficiency.

- S2226_OverlayText, S2226_OverlayTextIdx
- S2226_OverlayImage, S2226_OverlayImageIdx
- S2226_GetOverlayIdx
- S2226_OverlayBackgroundColor
- S2226_MoveOverlay
- S2226_OverlayDel, S2226_OverlayDelXY
- S2226_ClearOverlay
- S2226_UpdateOverlay
- S2226_ClearOverlayRegion
- S2226_SetOverlayRegion

Snapshots

Snapshots are uncompressed grabs from the 2226 in near real-time. They are captured directly in hardware at the same video resolution as the input. They are not decoded from the encoded stream so there will be no loss of quality due to compression. Due to USB bandwidth limitations, the snapshot feature is only intended for intermittent grabs of the stream. The user must not attempt to capture at anywhere near full frame rate using snapshots.

Snapshots are captured field by field. If using an interlaced source, the merge type should be specified. The merge type determines how to put the fields back together. Please see the function reference for more details.

- S2226_SetMergeMethod
- S2226_SnapshotToFile
- S2226_SnapshotToFileW
- S2226_SnapshotToMem
- S2226_SnapshotRaw

Notifications

The 2226 is a USB device. It should not be unplugged during streaming or when the application is running. However, sometimes this may happen. The functions `S2226_SetNotify` and `S2226_TestDeviceRemoval` are used to see if the device was unplugged. The demo application shows how these functions are used. The demo application will be closed if the 2226 is harshly unplugged while the demo is running.

- `S2226_SetNotify`
- `S2226_TestDeviceRemoval`

Cleanup/Shutdown

After all work is done with the 2226, the SDK should be closed for that board. An example is when the demo application closes. The following function closes the SDK for an open board.

- `S2226_CloseBoard`

Functions Reference

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

```
#define MID2226_API extern "C" __declspec(dllimport)
```

```
MID2226_API HANDLE S2226_OpenBoard(int devid);
```

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

Parameters

devid

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

Returns

0 on success, negative value if error (see `mid2226types.h` for error codes list).

```
MID2226_API int S2226_CloseBoard(HANDLE hdev);
```

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

Parameters

hdev

handle to device.

Returns

0 on success, negative value if error (see `mid2226types.h` for error codes list).

```

MID2226_API int S2226_Enumerate (
    int *count,
    DEVINFO2226 *devices
);

```

Enumerates all plugged in 2226s in the system. If *pCount equal to 0, the number of attached devices is set in *pCount. If *pCount != 0 and pDevices != NULL, then pDevices points to a list of at least *pCount DEVINFO2226 structures which get filled in with board number and serial number information. Please see demo application for an example.

Parameters

count

Returns devices found in system (if called with *pCount=0) or size of pDevices to fill with board information.

devices

array of at least *pCount devices. If querying the number of devices (*pCount = 0), pDevices may be NULL.

Returns

0 if success, negative if error.

```

MID2226_API int S2226_SetInput (
    MID2226_SOURCE input,
    BOOL bUpdateNow
    HANDLE hdev
);

```

Selects current input. If bUpdateNow is set to FALSE, the input will not be updated until streaming is started. If changing from an HD input to anything else(including a different HD input), some settings may be reset to default. Please note that input changes are not allowed during streaming. Always stop the stream before changing inputs.

Parameters

input

enumerated input MID2226_INPUT (see mid2226types.h). The allowed values are:

- MID2226_INPUT_COMPOSITE_0 (Main composite input for SD)
- MID2226_INPUT_SVIDEO_0 (Svideo input for SD)
- MID2226_INPUT_COMPOSITE_1 (Alternate composite input via header)
- MID2226_INPUT_SVIDEO_1 (Alternate svideo input)
- MID2226_INPUT_SD_COLORBARS (Not a physical input. Test input for SD)
- MID2226_INPUT_720P_COLORBARS (Not a physical input. Test input for 720p HD)
- MID2226_INPUT_1080I_COLORBARS (Not a physical input. Test input for 1080i HD)
- MID2226_INPUT_SDI_SD (SDI input with standard definition source)
- MID2226_INPUT_SDI_720P (SDI input with 720p HD source. 59.94Hz, 60Hz only)
- MID2226_INPUT_SDI_1080I (SDI input with 1080I HD source: 50Hz, 59.94Hz, 60Hz)

bUpdateNow

Set to TRUE to change input immediately (if stream stopped). Otherwise input changed at next stream start. If *bUpdateNow* is set to TRUE and new input has a different video system, call `S2226_SetVidSys` first or the new video system will not be updated until the stream is started.

hdev

handle to device (obtained from `S2226_OpenBoard`).

Returns

0 on success, negative value if error (see `mid2226types.h` for error codes list).

```
MID2226_API int S2226_GetInput(
    MID2226_SOURCE *pSource,
    HANDLE hdev
);
```

Retrieves current input settings.

Parameters

pSource

pointer to the value of current input.

hdev

handle to device (obtained from `S2226_OpenBoard`).

Returns

0 on success, negative value if error (see `mid2226types.h` for error codes list).

```

MID2226_API int S2226_SetBitrate (
    unsigned long bitrate,
    HANDLE hdev
);

```

Retrieves current input settings.

Parameters

bitrate

bitrate of encoded stream in kbps. (1000-17000kbps).

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```

MID2226_API int S2226_SetVidSys (
    MID2226_VIDSYS vidsys,
    HANDLE hdev
);

```

Sets the input video system (NTSC, PAL).

Parameters

vidsys

video system enumerated type (see mid2226types.h).

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```

MID2226_API int S2226_GetVidSys (
    MID2226_VIDSYS *pvidsys,
    HANDLE hdev
);

```

Gets the input video system (NTSC, PAL).

Parameters

pvidsys

pointer to video system enumerated type (see mid2226types.h).

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetReducedClock (  
    BOOL bReducedClock,  
    HANDLE hdev  
);
```

For HD inputs only (720p, 1080i) with NTSC video system. Some HD NTSC sources are at 59.94Hz and others at 60Hz. Allows changing between the two types. Note, this is a required setting for HD inputs with the NTSC video system.

Parameters

bReducedClock

if 1, video clock is at 59.94Hz, otherwise clock is 60Hz.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetReducedClock (  
    BOOL *bReducedClock,  
    HANDLE hdev  
);
```

Retrieves reduced clock value.

Parameters

bReducedClock

if 1, video clock is at 59.94Hz, otherwise clock is 60Hz.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetStatus(  
    MID2226STATUS *pStatus,  
    HANDLE hdev  
);
```

Retrieves current status information (see MID2226func.h for MID2226STATUS type definition).
Multibyte (ASCII) filenames.

Parameters

pStatus

pointer to status variable.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetStatusW(  
    MID2226STATUS_W *pStatus,  
    HANDLE hdev  
);
```

Same as S2226_GetStatus, but uses Unicode filenames.

```
MID2226_API int S2226_StartRecord(  
    char *fileName,  
    HANDLE hdev  
);
```

Starts recording to a file.

Parameters

fileName

full path to the target file, no extension.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StartRecordW(  
    wchar_t *fileName,  
    HANDLE hdev  
);
```

Same as S2226_StartRecord, but uses widechar or Unicode filenames.

```
MID2226_API int S2226_StopStream(  
    HANDLE hdev  
);
```

Stops streaming (recording, playing, previewing). Any registered callback is cleared.

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetLevel(  
    int param,  
    int value,  
    HANDLE hdev  
);
```

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

Parameters

param

defines the parameter to set (MID2226_LEVEL_CONTRAST, MID2226_LEVEL_BRIGHTNESS, MID2226_LEVEL_SATURATION, MID2226_LEVEL_HUE). See see mid2226types.h for definitions.

value

defines the value of selected parameter

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetLevel(  
    int param,  
    int value,  
    HANDLE hdev  
);
```

Retrieves video levels.

Parameters

param

defines the parameter to get (MID2226_LEVEL_CONTRAST, MID2226_LEVEL_BRIGHTNESS, MID2226_LEVEL_SATURATION, MID2226_LEVEL_HUE). See see mid2226types.h for definitions.

value

pointer to returned value of selected parameter

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioGain(  
    int gainL,  
    int gainR,  
    HANDLE hdev  
);
```

Sets gain settings for internal PGA audio amp.

Parameters

gainL

gain for left channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input

gainR

gain for right channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioGain(  
    int *gainL,  
    int *gainR,  
    HANDLE hdev  
);
```

Retrieves gain settings for internal PGA audio amp.

Parameters

gainL

gain for left channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input

gainR

gain for right channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioAgc(  
    int bOnL,  
    int bOnR,  
    int gainL,  
    int gainR,  
    HANDLE hdev  
);
```

Sets gain settings for audio automatic gain control.

Parameters

bOnL

toggles AGC gain for left channel. 0(off) recommended for line input.

bOnR

toggles AGC gain for right channel. 0(off) recommended for line input.

gainL

AGC gain for left channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input.

gainR

AGC gain for right channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioAgc(  
    int *bOnL,  
    int *bOnR,  
    int *gainL,  
    int *gainR,  
    HANDLE hdev  
);
```

Retrieves gain settings for audio automatic gain control.

Parameters

bOnL

toggles AGC gain for left channel. 0(off) recommended for line input.

bOnR

toggles AGC gain for right channel. 0(off) recommended for line input.

gainL

AGC gain for left channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input.

gainR

AGC gain for right channel(decibels times 2). 0-118 (0-59dB) 0dB recommended for line input.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioBalanced(  
    BOOL bBalanced,  
    HANDLE hdev  
);
```

Sets whether audio input is balanced(differenced) or not (default).

Parameters

bBalanced

1 if input is balanced, 0 otherwise.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioBalanced(  
    BOOL *bBalanced,  
    HANDLE hdev  
);
```

Retrieves audio balanced setting.

Parameters

bBalanced

1 if input is balanced, 0 otherwise.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioOutputVol (  
    int val,  
    HANDLE hdev  
);
```

Used for S2226_StartDecode only. Set the audio DAC volume.

Parameters

val

0-127 (steps of -0.5 dB, 0=0Db=maximum volume, 127=-63.5dB).

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioOutputVol (  
    int *val,  
    HANDLE hdev  
);
```

Retrieves audio output volume setting.

Parameters

val

0-127 (steps of -0.5 dB, 0=0Db=maximum volume, 127=-63.5dB).

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioOutputMono (
    int extra_gain,
    HANDLE hdev
);
```

Sets extra gain on the audio mono channel.

Parameters

extra_gain

0-9 (in steps of 1 dB). The default and recommended setting is 0.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioOutputMono (
    int *extra_gain,
    HANDLE hdev
);
```

Retrieve extra gain setting for audio mono channel.

Parameters

extra_gain

0-9 (in steps of 1 dB). The default and recommended setting is 0.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioOutputHp (
    int extra_gain,
    HANDLE hdev
);
```

Sets extra gain on the audio high power(HP) channel.

Parameters

extra_gain

0-9 (in steps of 1 dB). The default and recommended setting is 0.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioOutputHp (  
    int *extra_gain,  
    HANDLE hdev  
);
```

Retrieve extra gain setting for audio high power(HP) channel.

Parameters

extra_gain

0-9 (in steps of 1 dB). The default and recommended setting is 0.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioOutputStereo (  
    int extra_gain,  
    HANDLE hdev  
);
```

Sets extra gain on the audio high stereo channel.

Parameters

extra_gain

0-9 (in steps of 1 dB). The default and recommended setting is 0.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioOutputStereo (  
    int *extra_gain,  
    HANDLE hdev  
);
```

Retrieve extra gain setting for audio stereo channel.

Parameters

extra_gain

0-9 (in steps of 1 dB). The default and recommended setting is 0.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetAudioRoute (  
    int route,  
    HANDLE hdev  
);
```

Sets audio route. For debug or setup only. Do not leave audio route on bypassed while recording. If route is bypassed, then the audio from the input is directly connected to the output.

Parameters

route

0 – normal audio input, 1 – unused, 2 - bypassed input, 3 - unused

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetAudioRoute (  

```

```
int *route,  
HANDLE hdev  
);
```

Gets the current audio routing. For debug or setup only.

Parameters

route

0 – normal audio input, 1 – unused, 2 - bypassed input, 3 - unused

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StartCallback(  
HANDLE hdev  
);
```

Starts streaming with H264 data sent to the callback function registered with S2226_RegisterCallback function. After S2226_StopStream, callback in reset. Callbacks should be short and return quickly.

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StopCallback(  
HANDLE hdev  
);
```

Stops streaming to callback function

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_RegisterCallback(  
    cbfunc_t callback_function,  
    HANDLE hdev  
);
```

Registers a callback function. Call before S2226_StartPreview, S2226_StartRecord, S2226_StartPreviewAndRecord or S2226_StartCallback to receive the data as it is captured. Data available in the callback is the compressed MPEG H.264 transport stream. Callback is unregistered after S2226_StopStream. MPEG data is received from the board in a bursty manner. As such, the callback may be called at any time during streaming.

Parameters

callback_function

callback function. See header file for definition of cbfunc_t function.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_RegisterCallbackRaw(  
    cbfunc_t callback_function,  
    HANDLE hdev  
);
```

Registers a callback function. Call before S2226_StartPreview, S2226_StartRecord, S2226_StartPreviewAndRecord or S2226_StartCallback to receive the data as it is captured. Data available in the callback is the raw preview downscaled stream (320x240). The callback is unregistered after S2226_StopStream. Each callback will occur on capture of a single complete frame (but not necessarily synchronized to the MPEG stream).

Parameters

callback_function

callback function. See header file for definition of cbfunc_t function.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StartDecode(  
    char *fileName,  
    MID2226_DECODE_TYPE decode_type,  
    HANDLE hdev  
);
```

Starts decoding a file stream on the 2226 hardware. Output will not be on host or PC. Output will be from physical 2226 hardware connectors. If the stream (decode_type) is HD (720p or 1080i), there will be no valid output from the composite outputs on the 2226. The 2226 is not full duplex so S2226_StartDecode may not be called while recording or performing callback streaming.

Parameters

fileName

full path to the target file, no extension.

decode_type

decode type. See mid2226types.h.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StartDecodeW(  
    wchar_t *fileName,  
    HANDLE hdev  
);
```

Same as S2226_StartDecode, but uses widechar or Unicode filenames.

```
MID2226_API int S2226_PauseDecode(  
    HANDLE hdev  
);
```

If a decode is in progress, this function will pause the output.

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_NormalDecode(  
    HANDLE hdev  
);
```

If a decode is in progress and was paused or running in the slow mode, this function will resume normal output operation at full speed.

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SlowDecode(  
    int hold,  
    HANDLE hdev  
);
```

If a decode is in progress, this function will start slow motion decode. The hold time controls how long each successive frame is displayed.

Parameters

hold

hold time is the number of 1/2 frames to hold each frame on decode. Use 4(half speed) to 255 (slowest).

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetPreviewType (  
    MID2226_PREVIEWTYPE type,  
    HANDLE hdev  
);
```

Sets the preview type (on the host computer, NOT the 2226 physical outputs). Raw preview is downscaled to 320x240 resolution to meet USB2.0 bandwidth requirements (combined with the MPEG stream).

Parameters

type

Preview type (raw or decoded MPEG stream)

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StartPreview (  
    HWND hwnd,  
    HANDLE hdev  
);
```

Starts video preview achieved by decompressing the stream. Available on Windows 7 only. Please note that H264 requires significant CPU resources to decode. Please see preview in the function summary for more information about preview.

Parameters

hwnd

handle to display video in (use NULL for pop-up window).

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StartPreviewAndRecord(  
    HWND hwnd,  
    char *fileName,  
    HANDLE hdev  
);
```

Starts recording to a file with video preview achieved by decompressing the stream. Available on Windows 7 only. Please note that H264 requires significant CPU resources to decode. Please see preview in the function summary for more information about preview.

Parameters

hwnd

handle to display video in (use NULL for pop-up window).

fileName

full path to the target file, no extension.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_StartPreviewAndRecordW(  
    HANDLE hwnd,  
    wchar_t *fileName,  
    HANDLE hdev  
);
```

Same as S2226_StartPreviewAndRecord, but uses widechar or Unicode filenames.

```
MID2226_API int S2226_Repaint(  
    HDC hdc,  
    HANDLE hdev  
);
```

Only used for preview and when rendering to a non-NULL hwnd. Call this function in your OnPaint routine. See demo application for example usage.

Parameters

hdc

handle to device context.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetNotify(  
    HWND hNotifyWnd,  
    UINT mNotifyMessage  
    HANDLE hdev  
);
```

Use to set up a callback to the HWND when device message occurs. Use in conjunction with S2226_TestDeviceRemoval to see when device removed (see demo application for implementation details).

Parameters

hNotifyWnd

Window to notify on device event (removal)

mNotifyMessage

Window message to use for notification.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_TestDeviceRemoval(  
    HANDLE hdev  
);
```

After receiving a message (set up by S2226_SetNotify), call S2226_TestDeviceRemoval to see if device was removed (unplug).

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

```

MID2226_API int S2226_OverlayText(
    int xpos,
    int ypos,
    overlay_text_t *ovltext,
    int regionmask,
    HANDLE hdev
);

```

Adds overlay text. If overlay text already exists at that x,y position, deletes windows before adding. Overlay active on regions defined by regionmask. If text contains embedded newline characters (\n = 10 dec), then each line of text will be created on in a new window, AtIndex sub-window position, directly below the preceding line. Each sub-window position only consumes enough overlay memory needed to hold the individual line. If text contains embedded character 30 dec (entered programmatically or by holding down Alt- and typing "030" on the numeric keypad), then each line following a char(30) will be on a new line. Multi-line text created this way will be created as one large graphic at one "index" location.

Parameters

xpos

start x position.

ypos

start y position.

ovltext

pointer to overlay text.

regionmask

MID2226_REGION_MONITOR, MID2226_REGION_MPEG, MID2226_REGION_STILL.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, -1 if too many regions, negative value if error (see mid2226types.h for error codes list).

```

MID2226_API int S2226_OverlayTextIdx(
    int AtIndex,
    int xpos,
    int ypos,
    overlay_text_t *ovltext,

```

```

    int regionmask,
    HANDLE hdev
);

```

Adds overlay text. If overlay already exists at index WinIndex, deletes window before adding. See S2226_OverlayText for more details about the overlay and multi-line support.

Parameters

AtIndex

(0-7) sub window position to update/add text x position.

xpos

start x position.

ypos

start y position.

ovltext

pointer to overlay text.

regionmask

MID2226_REGION_MONITOR, MID2226_REGION_MPEG, MID2226_REGION_STILL.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```

MID2226_API int S2226_OverlayImage(
    int xpos,
    int ypos,
    char *imageFile,
    int regionmask,
    HANDLE hdev
);

```

Adds overlay image. If overlay already exists at that x, y position, deletes window before adding.

Parameters

xpos

start x position.

ypos

start y position.

imageFile

full path to image file. Must be a 24bit BMP or PCX bitmap image only. See demo application and Logo.bmp.

regionmask

MID2226_REGION_MONITOR, MID2226_REGION_MPEG, MID2226_REGION_STILL.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, -1 if too many regions, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_OverlayImageIdx(  
    int AtIndex,  
    int xpos,  
    int ypos,  
    char *imageFile,  
    int regionmask,  
    HANDLE hdev  
);
```

Adds overlay image. If overlay already exists at that x, y position, deletes window before adding.

Parameters

AtIndex

(0-7) sub-window position to update/add text.

xpos

start x position.

ypos

start y position.

imageFile

full path to image file. Must be a 24bit BMP or PCX bitmap image only. See demo application and Logo.bmp.

regionmask

MID2226_REGION_MONITOR, MID2226_REGION_MPEG, MID2226_REGION_STILL.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetOverlayIdx(  
    int WinIndex,  
    int *type,  
    int *region,  
    int *group,  
    int *xpos,  
    int *ypos,  
    char **value,  
    HANDLE hdev  
);
```

Adds overlay image. If overlay already exists at that x, y position, deletes window before adding.

Parameters

WinIndex

(0-7) sub-window position to query.

type

1=image, 0=text

region

MID2226_REGION_MONITOR, MID2226_REGION_MPEG, MID2226_REGION_STILL.

group

text with common group number is kept together.

xpos

start x position.

ypos

start y position.

value

pointer to text or image file path.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_OverlayBackgroundColor (  
    int red,  
    int green,  
    int blue,  
    HANDLE hdev  
);
```

Sets the color used for the background regions.

Parameters

red

red component 0-255

green

green component 0-255

blue

blue component 0-255

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_MoveOverlay(  
    int WinIndex,  
    int xpos,  
    int ypos,  
    HANDLE hdev  
);
```

Changes the position of an overlay sub-window

Parameters

WinIndex

index of window to move

xpos

new x position

ypos

new y position

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_OverlayDelXY(  
    int xpos,  
    int ypos,  
    HANDLE hdev  
);
```

Delete an overlay at x, y location if one exists at that location.

Parameters

xpos

x position

ypos

y position

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, -1 if overlay does not exist, -2 if that sub-window already used.

```
MID2226_API int S2226_OverlayDel(  
    int WinIndex,  
    HANDLE hdev  
);
```

Delete overlay with index WinIndex.

Parameters

WinIndex

(0-7) sub-window position to query.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, -1 if overlay does not exist.

```
MID2226_API int S2226_UpdateOverlay(  
    HANDLE hdev  
);
```

Refresh or update the overlay to the hardware.

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_ClearOverlay(  
    HANDLE hdev  
);
```

Clears all overlays for every region.

Parameters

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_ClearOverlayRegion(  
    int regionmask,  
    HANDLE hdev  
);
```

Clears all overlays for specified region.

Parameters

regionmask

MID2226_REGION_MONITOR, MID2226_REGION_MPEG, MID2226_REGION_STILL.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetOverlayRegion(  
    int regionmask,  
    HANDLE hdev  
);
```

Sets all overlays for specified region.

Parameters

regionmask

MID2226_REGION_MONITOR, MID2226_REGION_MPEG, MID2226_REGION_STILL.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetMergeMethod(  
    MID2226_FIELDALG merge_method,
```

```

    void *unused,
    HANDLE hdev
);

```

Sets the interlaced field merging algorithm. Used for snapshots/stills.

Parameters

merge_method

MID2226_FIELDALG_NONE, MID2226_FIELDALG_DUP, MID2226_FIELDALG_MERGE,
MID2226_FIELDALG_INTER

unused

possible future use. Set to NULL.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```

MID2226_API int S2226_SnapshotToFile(
    char *filename,
    int filetype,
    int freetime,
    int wait,
    int qual,
    int unused,
    HANDLE hdev
);

```

Takes a snapshot and save to file in filename.

Parameters

filename

fully qualified file with path (without extension, extension will be added by the SDK).

filetype

file type to save. MID2226_FILE_JPEG and/or MID2226_FILE_BMP.

freetime

time in milli-seconds to freeze the image(freezing is done on the video output channels of the 2226).

wait

wait = 1 will wait if operations pending(board is busy), wait = 0 will return err code if board busy.

unused

Unused. For future possible use. Value will be ignore.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SnapshotToFileW(  
    const wchar_t *filename,  
    int filetype,  
    int freezetime,  
    int wait,  
    int qual,  
    int unused,  
    HANDLE hdev  
);
```

Takes a snapshot and save to file in filename (unicode version)

Parameters

filename

fully qualified unicode filename with path (without extension, extension will be added by the SDK).

filetype

file type to save. MID2226_FILE_JPEG and/or MID2226_FILE_BMP.

freezetime

time in milli-seconds to freeze the image(freezing is done on the video output channels of the 2226).

wait

wait = 1 will wait if operations pending(board is busy), wait = 0 will return err code if board busy.

unused

Unused. For future possible use. Value will be ignore.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SnapshotToMem(  
    BYTE *image,  
    int size,  
    int freetime,  
    int wait,  
    HANDLE hdev  
);
```

Get snapshot to memory function. Retrieves processed image into memory. Image is converted to RGB (1 byte per color) and the fields are merged using the algorithm set by S2226_SetMergeMethod.

Parameters

image

pointer to retrieved snapshot. (image must be preallocated).

size

size of the image buffer (image).

freetime

time in milli-seconds to freeze the image(freezing is done on the video output channels of the 2226).

wait

wait = 1 will wait if operations pending(board is busy), wait = 0 will return err code if board busy.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SnapshotRaw(  
    BYTE *image,  
    int size,  
    image_raw_t *image_raw,  
    int freetime,  
    int wait,  
    HANDLE hdev  
);
```

Get snapshot data from hardware only. This function does no processing on the image, it just retrieves a pointer to the raw fields(field 1 will be null for 720p format) in YCrCb format. Image itself stored in image parameter. image_raw parameter used to indicate where the fields start and their size.

Parameters

image

pointer to pre-allocated space for image.

size

size of the image buffer (image).

image_raw

pointer to the returned raw image fields.

freetime

time in milli-seconds to freeze the image(freezing is done on the video output channels of the 2226).

wait

wait = 1 will wait if operations pending(board is busy), wait = 0 will return err code if board busy.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetFirmwareVersions(  

```

```

    unsigned int *fpga,
    unsigned int *usb,
    unsigned int *cpu,
    unsigned int *boardid,
    HANDLE hdev
);

```

Retrieves hardware firmware information. For support information.

Parameters

fpga

FPGA firmware version.

usb

USB firmware version.

CPU

Embedded CPU firmware version

boardid

Board revision identifier.

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```

MID2226_API int S2226_GetSDKVersions (
    unsigned int num,
    unsigned int *ver,
    HANDLE hdev
);

```

Retrieves version for SDK software. Useful for debugging Windows installation issues. See demo app for example usage.

Parameters

num

number of version to receive

ver

array of versions (driver core version[0], driver proxy version[1], dll version[2])

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_GetSN(  
    SN2226 *sn,  
    HANDLE hdev  
);
```

Retrieves device serial number.

Parameters

sn

device serial number structure

hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

```
MID2226_API int S2226_SetOutput(  
    int type,  
    HANDLE hdev  
);
```

Sets the output control for the device for the composite outputs. Multiplexed with the raw preview feature.

Parameters

type

0-scaled output off(raw preview may be used), 1-scaled output on (raw preview not available).

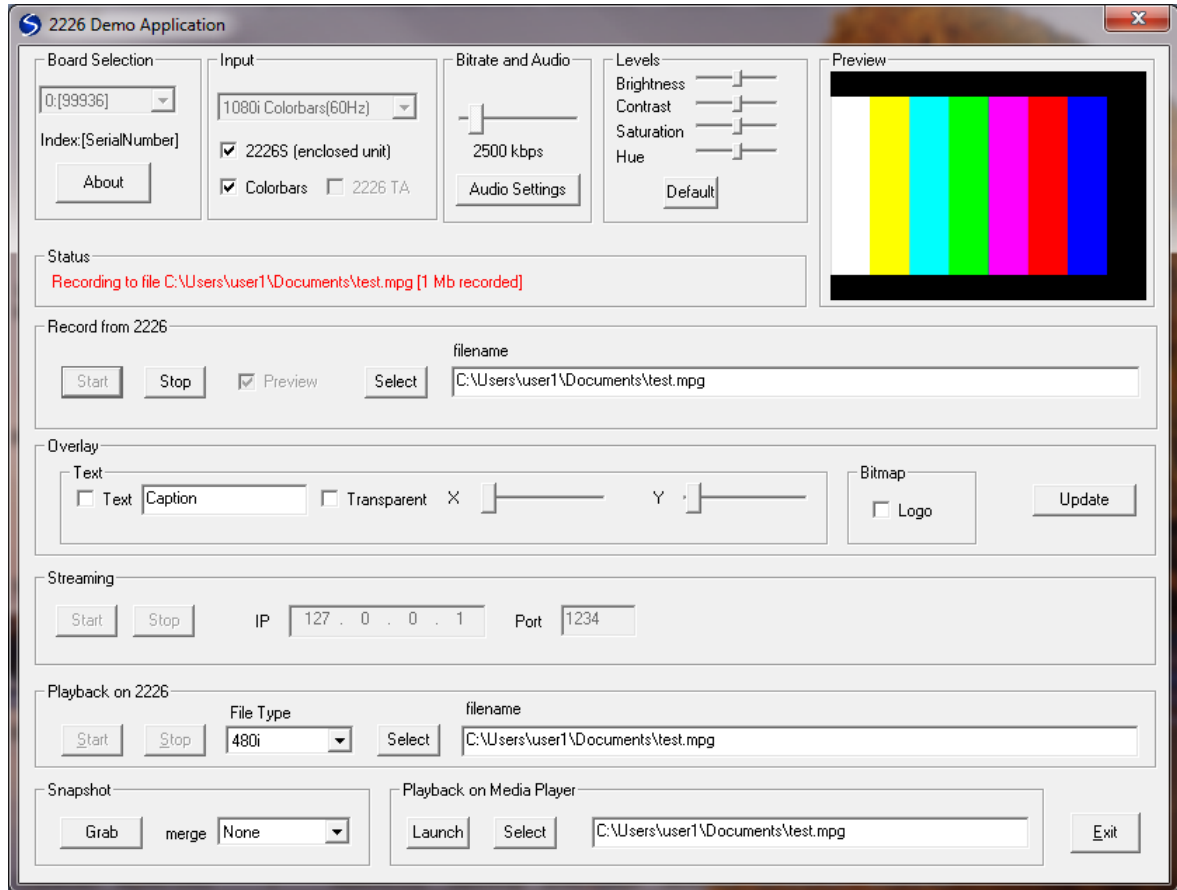
hdev

handle to device (obtained from S2226_OpenBoard).

Returns

0 on success, negative value if error (see mid2226types.h for error codes list).

Demo Application



The demo application is shown above.

Board Selection

The board selection allows the user to select between multiple boards in the system. The First number represents the zero indexed board number. The second number following the colon is the serial number.

The About button pops up a dialog showing the current board's firmware revisions.

Input

The input must be selected before performing any streaming operation. Select from the drop down. The "Colorbars" checkbox is used to record internally generated colorbars on the 2226. Checking "Colorbars" adds the color bar inputs to the Input drop down. If color bars are desired, "Colorbars" must

be selected AND the color bar input selection made from the drop down list. "2226TA" adds the auxiliary inputs available using the 2226 termination board (2226TA). "2226S" is the default setting and displays the inputs on the boxed 2226 system.

Bitrate and Audio

Bitrate is the encoded bitrate used when streaming and recording from the 2226. Audio settings brings up the audio gain and output settings.

Levels

The brightness, contrast, hue and saturation may be changed at any time with the slider bars.

Preview

The preview window displays the preview if selection with the "Preview" checkbox while recording from the 2226.

Record from 2226

Record from the 2226 to file. Files are encoded as H.264 MPEG with the selected bitrate.

Overlay

Demonstrates some of the 2226 overlay capability. Click the Update button to refresh the current overlay settings on the 2226.

Streaming

Streams the encoded H264 transport stream over UDP.

Playback on 2226

Decodes the recorded H264 stream on the 2226 and output to hardware outputs. Please make sure file type is correct before starting.

Snapshot

Grabs an snapshot from the 2226. When the file dialog appears, do not enter an extension. A bitmap and JPG file will be created with the required file extensions.

Playback on Media Player

Launches file on the default media player. WMP support for Win7 only. If running on XP, the demo will check for the VLC player.

FAQ

Q1) Can the 2226 record one file and decode another at the same time?

A1) The 2226 is half duplex. Only one operation (encode/decode) may be performed at a time.

Q2) I need to pause the video. Why does the pause button in GraphEdit not work?

A2) The hardware does not support pausing of the stream. If pause is required, it may be possible to construct a custom callback function to drop packets. At this time, Sensoray does not provide support for such a feature. The recommended way to support pause is to record with separate clips.

Q3) I can't play back the recorded file in Windows media player on XP?

A3) XP and Vista do not necessarily have the H264 decoders to decode the stream. The recorded stream may be played back with WMP under Windows 7. A player that works under XP is VideoLan.

Q4) What are the adapter inputs in the demo application?

A4) An optional termination board is available for the 2226 (2226-TA) to provide S-Video and an extra composite video channel. This does not apply for the boxed 2226S unit.

Q5) Why do the image snapshots show 2 images.

A5) The S2226_SnapshotToFile and S2226_SnapshotToFileW functions work in conjunction with S2226_SetMergeType. For maximum flexibility in the API, the user is given full control over how the fields are presented. Change the merge type in the demo application to merge the fields into one frame.