PCI-Express 8-Channel Frame Grabber

Software Manual (Windows)

Model 812 | Ver. 1.0.24 | August 2022

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

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 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 THEREOF.

Introduction

The Sensoray Model 812 is a PCI express 8-channel frame capture device. It also supports the capture of raw PCM audio.

Software Feature Summary

- Video preview
- Audio preview
- AVI recording
- Callback functionality for direct capture of both video and audio streams.
- Full featured C# .NET demo application.
- Windows driver(AVStream/DirectShow) and Software Development Kit (SDK).

Software

Feature Summary

Sensoray's Model 812 is shipped with drivers for Microsoft Windows XP, Vista, Windows 7. A full-featured demo application demonstrates capture of the audio/video stream. Control is provided for brightness, bitrate, contrast and other video attributes. NTSC or PAL inputs are supported.

The SDK allows maximum flexibility by providing an API for all of the 812's functions. The source code of the C# demo application is a suggested starting point for custom application development.

Since the 812 is 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. For D1 capture of all 8 channels, Sensoray recommends a modern and up to date computer.

Installation

The software maybe 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/812 folder or /Program Files(x86)/Sensoray/812.

During the installation the program will pre-install the drivers using Dpinst. Do not click cancel when installing the driver. Also, the 812 device does not support hotswap. Do not plug or unplug the board when the computer is turned on.

Redistribution

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 demo application uses .NET 4.0

SDK Reference

Release Notes

V1.0.24

- Driver update. Increased DMA timeout.

V1.0.23

- Added S812_GetFrame Csharp decorations.
- Reduced maximum number of buffers used in driver for lower memory usage and reduced latency.

V1.0.22

- Fix for affinity mask pop-up error.

V1.0.21

- Fix for record to AVI on Windows 10. Please note recording is in raw format and files will be very large.

V1.0.20.2

- Added VS2019 project files for app-812-csharp. No other changes.

V1.0.20

- DLL fix for choppy video on some systems. Demo and DLL remove the possibility of accidentally configuring NTSC with unsupported 352x288 resolution. Windows 10 driver digitally signed with updated Microsoft cross-signature.

V1.0.19

- Minor demo application updates.

V1.0.18

- Driver update for Windows 10 version 1607+ with secure boot.

V1.0.17

- Additional improvements for lost signal recovery.
- Fixed "export debug" Program Files permissions issue in demo application.

V1.0.16

- Drop-down menu for demo application to change all channel settings at same time.
- Save Settings and Load Settings added to demo applications.
- Export debug settings (DLL version, driver, configuration) for customer support purposes.
- Driver update for loss of video signal recovery on individual channels.
- More robust recovery from lost video signals if signal turned off/on or re-connected while streaming.
- Adds S812_SetDShow and S812_GetDShow as optional replacements for S812_SetDirectShow, which has the opposite parameter order of device index and stream index from other functions.
- Adds S812_SetFlipImage and S812_GetFlipImage. S812_FlipImage is kept for backward compatibility.

V1.0.15

- bug fix for 1.0.14 standards. Channels 1-4 were set incorrectly in some cases in 1.0.14. Recommended upgrade from 1.0.14

V1.0.14

- Improved extra drivers for PAL-only or NTSC-only. Audio PLL value fix to avoid audio drift.

- Demo application adds extra settings for driver version and for changing driver mode to NTSC-only or PAL-only. In these modes, the video standard is limited to either NTSC or PAL. This is useful for third party applications that may not set the standard when the application is first started. This ensures the video standard matches the connected camera at all times if the connected camera is never changed. Please see Appendix A for installation help for this setting.
- S812_GetDriverVersion, S812_SetFixedStandard, S812_GetFixedStandard added.

V1.0.13

- Extra drivers for PAL-only or NTSC-only. Limits video standard in GraphEdit.

V1.0.12

- Extra debug and diagnostic information. Bug fix for Capture pin non-English language issue.

V1.0.1 – V1.0.11

- General SDK improvements

V.1.0.0 (November 9, 2012):

- Initial release

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 S812_GetNumDevices. If using a single board, it may be opened without enumeration using the index 0.
2. Initialization. This is performed by a call to `S812_Open()` function with the board index parameter. Initial default capture settings are loaded.
3. A call to `S812_Open()` may be followed by calls to the functions controlling various settings. At a minimum, the correct video system should be set:
    - video system: `S812_SetVidSys(), S812_SetVidSize();`
    - video parameters (brightness, contrast, saturation, hue): `S812_SetLevel();`
4. A call to `S812_StartPreview()` starts the 812 and previews the data to the monitor. S812_StopStream() stops preview from the 812. The demo application also shows how to capture the data for other purposes using the callback feature.
5. During preview or recording, snapshots or frames may be grabbed using the S812_SnapshotToFile or S812_GetFrame functions. If frames need to be grabbed without recording or previewing, the callback functionality (S812_StartCallback, S812_RegisterCallback) may be used.
6. Audio preview may be started or stopped using S812_StartAudioPreview and S812_StopAudioPreview.
7. Streams (callback, recording or preview) may be stopped with the S812_StopStream function.
8. `S812_Close()` must be called before the application terminates to properly clean up and close the board handle to the driver.

Demo applications

The SDK includes a demo application provided with the source code to illustrate the use of SDK's functions. Right click on the channel in question to preview, record, or view the raw data (callback) from the 812 board. The Demo application is written in C# Windows Forms .NET and provides a wrapper to the "C" DLL in the file s812.cs.

SDK Function Reference

All API functions are declared using the following definition and the **__stdcall calling convention**:
#define MID812_API extern "C" __declspec(dllimport)

```
MID812_API HANDLE S812_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 mid812types.h for error codes list).

```
MID812_API int S812_CloseBoard(int devid);
```

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

Parameters

devid

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

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetNumDevices(int *num_devices);
```

Enumerates all plugged in 812s in the system.

Parameters

num_devices

Returns devices found in system.

Returns

0 if success, negative if error.

```
MID812_API int S812_SetVidSys(
   MID812_VIDSYS vidsys,
   int devid,
   int strmidx
);
```

Sets the input video system (NTSC, PAL).

Parameters

vidsys

video system enumerated type (see mid812types.h).

devid

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

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetVidSys(
   MID812_VIDSYS *pvidsys,
   int devid,
   int strmidx
);
```

Gets the input video system (NTSC, PAL).

Parameters

pvidsys

pointer to video system enumerated type (see mid812types.h).

devid

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

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetPreviewWindow(
   HWND hwnd,
   int devid,
   int strmidx
);
```

Sets the video preview window

Parameters

hwnd

handle to window

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetVideoPosition(
   int xpos,
   int ypos,
   int xsize,
   int ysize,
   int devid,
   int strmidx
);
```

Sets the video position inside the window specified with S812_SetPreviewWindow. This function is strictly optional. If this function is not called, the video will automatically occupy the entire window.

Parameters

xpos

x position

ypos

y position

xsize

video width in the window

ysize

video height in the window

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StartPreview(
   int devid,
   int strmidx
);
```

Starts video preview

Parameters

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StartRecord(
   char *fileName,
   int devid,
   int strmidx
);
```

Starts recording to a file. Please note that the 812 is a uncompressed capture device and this function was designed for recording short clips. Similarly, it is not recommended to record more than one stream at a time using this function.

Parameters

fileName

full path to the target file, no extension.

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StartRecordW(
   wchar_t *fileName,
   int devid,
   int strmidx
);
```

Same as S812_StartRecord, but uses widechar or Unicode filenames.

```
MID812_API int S812_StopStream(
   int devid,
   int strmidx
);
```

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

Parameters

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StartAudioPreview(
   int devid,
   int strmidx
);
```

Starts audio preview on host PC.

Parameters

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StopAudioPreview(
   int devid,
   int strmidx
);
```

Stops audio preview on host PC.

Parameters

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetLevel(
   int param,
   int value,
   int devid,
   int strmidx
);
```

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

param

defines the parameter to set (`MID812_LEVEL_CONTRAST, MID812_LEVEL_BRIGHTNESS, MID812_LEVEL_SATURATION, MID812_LEVEL_HUE).`

See see mid812types.h for definitions.

value

defines the value of selected parameter. Allowed values:

Brightness: MID812_BRIGHTNESS_MIN to MID812_BRIGHTNESS_MAX, default MID812_BRIGHTNESS_DEF

Contrast: MID812_CONTRAST_MIN to MID812_CONTRAST_MAX, default MID812_CONTRAST_DEF

Saturation: MID812_SATURATION_MIN to MID812_SATURATION_MAX, default MID812_SATURATION_DEF

Hue: MID812_HUE_MIN to MID812_HUE_MAX, default MID812_HUE_DEF

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetLevel(
   int param,
   int value,
   int devid,
   int strmidx
);
```

Retrieves video levels.
Parameters

param

defines the parameter to get (`MID812_LEVEL_CONTRAST, MID812_LEVEL_BRIGHTNESS, MID812_LEVEL_SATURATION, MID812_LEVEL_HUE).`

value

pointer to returned value of selected parameter

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetVidSize(
   MID812_RES vidsize,
   int devid,
   int strmidx
);
```

Sets the video size.
Parameters

vidsize

`MID812_RES_D1, MID812_RES_SIF, MID812_RES_CIF, MID812_RES_VGA, MID812_RES_4SIF, MID812_RES_4CIF.`

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetVidSize(
   MID812_RES *vidsize,
   int devid,
   int strmidx
);
```

Gets the current video size.

Parameters

vidsize

`MID812_RES_D1, MID812_RES_SIF, MID812_RES_CIF, MID812_RES_VGA, MID812_RES_4SIF, MID812_RES_4CIF.`

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_Repaint(
   HDC hdc,
   int devid,
   int strmidx
);
```

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 S812_OpenBoard).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_DisplayChange(
   int devid,
   int strmidx
);
```

Call if display changes for the preview window.

Parameters

hdc

handle to device context.

hdev

handle to device (obtained from S812_OpenBoard).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StartCallback(
   int devid,
   int strmidx
);
```

Starts streaming with data sent to the callback functions registered with S812_RegisterCallback function. Callbacks should be short and return quickly. Do not use blocking functions in the callback.

Parameters

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_RegisterCallback(
   cbfunc_t callback_function_video,
   cbfunc_t callback_function_audio,
   int devid,
   int strmidx
);
```

Registers a callback function for video and/or audio. Use NULL if callback is not desired.

Parameters

callback_function_video

callback function for video or NULL if none. See header file for definition of cbfunc_t function.

callback_function_audio

callback function for audio or NULL if none. See header file for definition of cbfunc_t function.

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StartPreviewAndRecord(
   char *fileName,
   int devid,
   int strmidx
);
```

Starts recording to a file with video preview. Please see preview in the function summary for more information about preview. It is not recommended to record more than one stream at a time using this function.

Parameters

fileName

full path to the target file, no extension.

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_StartPreviewAndRecordW(
   wchar_t *fileName,
   int devid,
   int strmidx
);
```

Same as S812_StartPreviewAndRecord, but uses widechar or Unicode filenames.

```
MID812_API int S812_SnapshotToFile(
   char *filename,
   int filetype,
   int qual,
   int devid,
   int strmidx
);
```

Takes a snapshot and save to file in filename. This function only applies when streaming video using S812_StartPreview, S812_StartRecord or S812_StartPreviewAndRecord
Parameters

filename

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

filetype

file type to save. MID812_FILE_JPEG and/or MID812_FILE_BMP.

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SnapshotToFileW(
   const wchar_t *filename,
   int filetype,
   int qual,
   int devid,
   int strmidx
);
```

Takes a snapshot and save to file in filename (unicode version). Same as S812_SnapshotToFlie but with unicode filename parameter.

```
MID812_API int S812_GetFrame(
   long bufsize,
   unsigned char *pdata,
   BITMAPINFOHEADER *pBMI,
   int devid,
   int strmidx
);
```

Get snapshot to memory function. Retrieves processed image into memery. This function only applies when streaming video using S812_StartPreview, S812_StartRecord or S812_StartPreviewAndRecord. If frequent frame capture is required, it is recommended to use the callback functions.
Parameters

bufsize

allocated size of pdata

pdata

buffer to hold frame

pBMI

pointer to bitmap info. Note: some fields may not be populated.

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetColorSpace(
   MID812_COLORSPACE cspace,
   int devid,
   int strmidx
);
```

Sets the colorspace for preview and for callback streaming. Please note that for preview if the video card does not support the colorspace (typically most modern video cards do not support RGB565), performance problems may result.

Parameters

cspace

colorspace of the RAW data. MID812_COLORSPACE_YUY2, MID812_COLORSPACE_UYVY, MID812_COLORSPACE_RGB565.

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetColorSpace(
   MID812_COLORSPACE *cspace,
   int devid,
   int strmidx
);
```

Returns the current colorspace setting.

Parameters

cspace

colorspace of the RAW data. MID812_COLORSPACE_YUY2, MID812_COLORSPACE_UYVY, MID812_COLORSPACE_RGB565.

devid

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

strmidx

channel index (0-7)

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_EnableSnapshots(
   int devid,
   int strmidx,
   BOOL bOn,
);
```

Controls whether snapshots are on or not. Snapshots require intermediate buffering of the data. Enabling snapshots may degrade preview performance because the image can no longer be directly rendered to video RAM.

Parameters

devid

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

strmidx

channel index (0-7).

bOn

TRUE-snapshots on, FALSE, snapshots off.

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetDirectShow(
   int devid,
   int strmidx,
   BOOL bDecomp,
   MID812_DSHOW renderer,
);
```

Advanced setting for more control of the preview. Sets which DirectShow rendering filters are used for preview.

Parameters

devid

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

strmidx

channel index (0-7).

bDecomp

TRUE-add decompressor filter, FALSE-do not add decompressor filter. Adding decompressor is recommended and may result in smoother video preview. Default is TRUE.

renderer

Directshow renderer filter to use.

MID812_VMR9, MID812_VMR7, MID812_DSHOW_LEGACY(not recommended).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetDShow(
   BOOL bDecomp,
   MID812_DSHOW renderer,
   int devid,
   int strmidx
);
```

Advanced setting for more control of the preview. Sets which DirectShow rendering filters are used for preview.

Parameters

bDecomp

TRUE-add decompressor filter, FALSE-do not add decompressor filter. Adding decompressor is recommended and may result in smoother video preview. Default is TRUE.

renderer

Directshow renderer filter to use.

MID812_VMR9, MID812_VMR7, MID812_DSHOW_LEGACY(not recommended).

devid

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

strmidx

channel index (0-7).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetDShow(
   BOOL *bDecomp,
   MID812_DSHOW *renderer,
   int devid,
   int strmidx
);
```

Gets the DirectShow rendering filters currently used for preview and set by S812_SetDShow.

Parameters

**bDecomp*

TRUE-add decompressor filter, FALSE-do not add decompressor filter. Adding decompressor is recommended and may result in smoother video preview. Default is TRUE.

**renderer*

Directshow renderer filter to use.

MID812_VMR9, MID812_VMR7, MID812_DSHOW_LEGACY(not recommended).

devid

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

strmidx

channel index (0-7).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetFlipImage(
   BOOL bFlipH,
   BOOL bFlipV,
   int devid,
   int strmidx
);
```

Advanced setting for flipping the image either horizontally, vertically, or both.

Parameters

bFlipH

TRUE-flip image horizontally. FALSE-default image horizontal orientation. Default is FALSE.

bFlipH

TRUE-flip image vertically. FALSE-default image vertical orientation. Default is FALSE.

devid

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

strmidx

channel index (0-7).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetFlipImage(
   BOOL *bFlipH,
   BOOL *bFlipV,
   int devid,
   int strmidx
);
```

Returns the current settings for flipping the image.

Parameters

**bFlipH*

TRUE-flip image horizontally. FALSE-default image horizontal orientation. Default is FALSE.

**bFlipH*

TRUE-flip image vertically. FALSE-default image vertical orientation. Default is FALSE.

devid

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

strmidx

channel index (0-7).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_FlipImage(
   BOOL bFlipH,
   BOOL bFlipV,
   int devid,
   int strmidx
);
```

For backward compatibily. Same as S812_SetFlipImage. Use S812_SetFlipImage instead for new code.

Parameters

bFlipH

TRUE-flip image horizontally. FALSE-default image horizontal orientation. Default is FALSE.

bFlipH

TRUE-flip image vertically. FALSE-default image vertical orientation. Default is FALSE.

devid

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

strmidx

channel index (0-7).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetDLLVersion(
   int *v1,
   int *v1,
   int *v3
);
```

Retrieves the current active running DLL version.

Parameters

v1

pointer to version major number

v2, v3

pointer to version minor numbers

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetDriverVersion(
   int *v1,
   int *v1,
   int *v3,
   int *v4,
   int devid,
   int strmidx,
);
```

Retrieves the current driver version for the 812 device and channel.

Parameters

v1, v2, v3, v4

pointer to driver version numbers

devid

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

strmidx

channel index (0-7).

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_SetFixedStandard(
   int fixed,
   int devid,
);
```

Limits ALL channels of all devices to a single video standard. Devid is a parameter, but this affects all boards in the system. The devid parameter should be an active device in the system. This driver setting is useful if the video standard of all cameras in the system are always the same. It is also useful if a third-party DirectShow application is used, which may not correctly set or save the video standard. See the Appendix for more information. This function is only available for SDK and driver version 1.0.14 or higher.

Parameters

fixed

0-disable this function, 1-all channels/devices NTSC, 2-all devices/channels PAL.

devid

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

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

```
MID812_API int S812_GetFixedStandard(
   int *fixed,
   int devid,
);
```

Returns the value of the video standard from S812_SetFixedStandard.

Parameters

fixed

0-disable this function, 1-all channels/devices NTSC, 2-all devices/channels PAL.

devid

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

Returns

0 on success, negative value if error (see mid812types.h for error codes list).

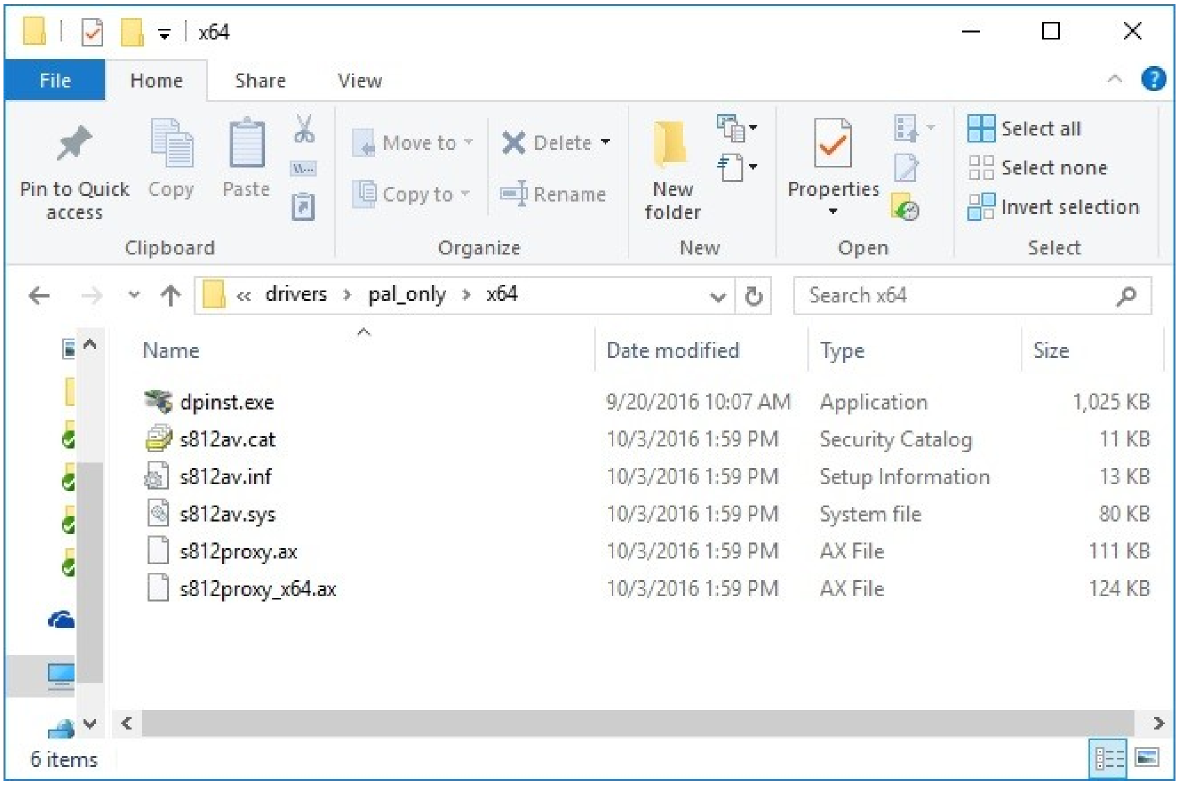
Appendix A: Fixed video standard

The function S812_SetFixedStandard limits the permissible video standard to a fixed value. If all cameras are PAL, this setting will prevent the video standard from changing to NTSC. This is useful for some third party applications, among other uses.

There are two ways to install the driver for this feature, described below. The first method is only supported for the PAL video system.

Method 1) PAL-only driver installation

Before installing the SDK, change to the directory pal_only under drivers in the installation package.



« drivers › pal_only › x64

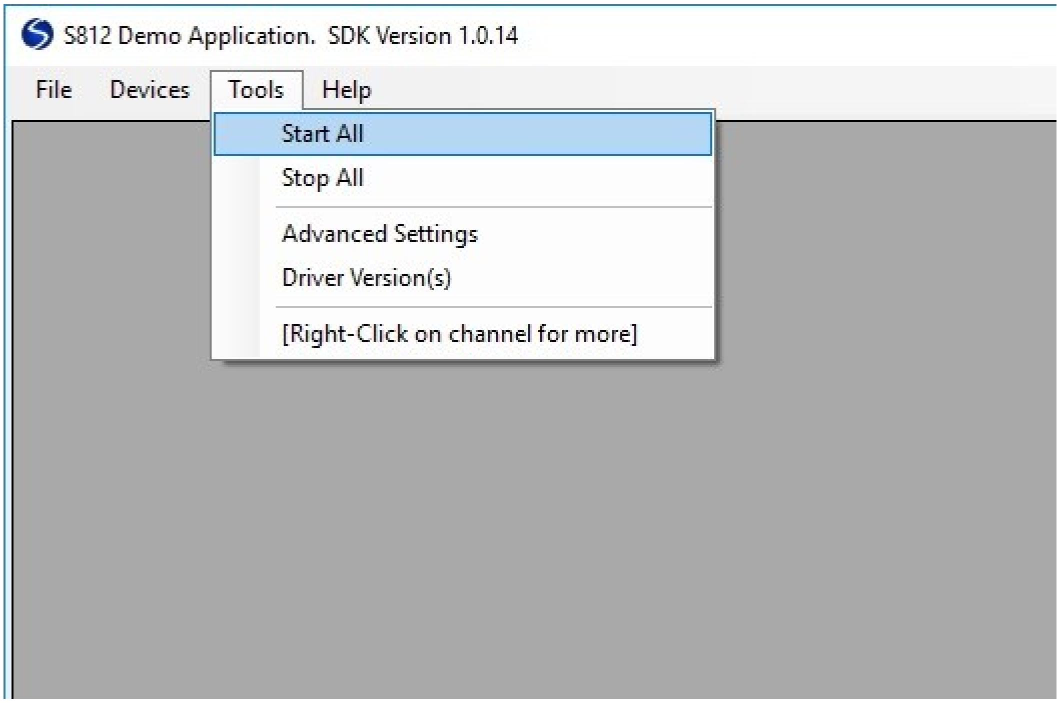
Name	Date modified	Type	Size
dpinst.exe	9/20/2016 10:07 AM	Application	1,025 KB
s812av.cat	10/3/2016 1:59 PM	Security Catalog	11 KB
s812av.inf	10/3/2016 1:59 PM	Setup Information	13 KB
s812av.sys	10/3/2016 1:59 PM	System file	80 KB
s812proxy.ax	10/3/2016 1:59 PM	AX File	111 KB
s812proxy_x64.ax	10/3/2016 1:59 PM	AX File	124 KB

6 items

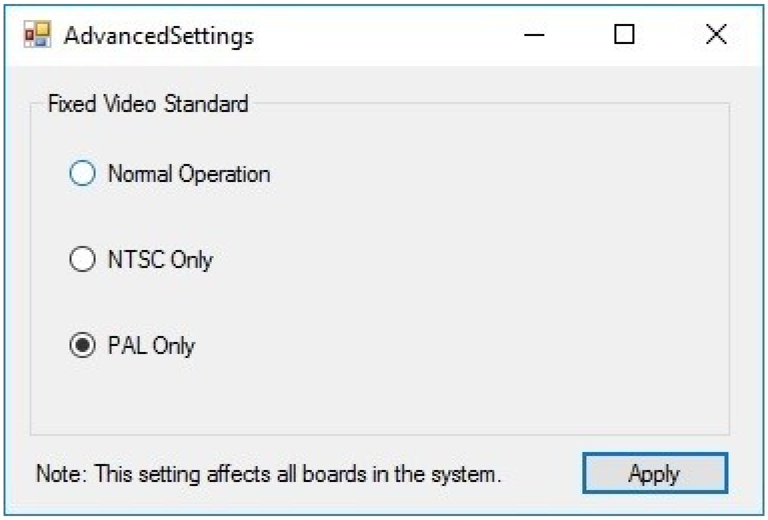
Double click on dpinst.exe to install the pal_only driver. The driver/SDK will be fixed to the PAL video standard. If done in error, the pal_only setting may be removed from the demo application (see below).

Method 2) NTSC-only or PAL-only mode using demo application

- Install SDK version 1.0.14 or higher. Make sure to install the driver when prompted.
- After successfully installing the SDK, run the demo application.
- In the demo application, click on Tools→Advanced settings.


S812 Demo Application. SDK Version 1.0.14
File
Devices
Tools
Help
Start All
Stop All
Advanced Settings
Driver Version(s)
[Right-Click on channel for more]

- In the Advanced settings menu, change from disabled to NTSC-only OR PAL-only depending on the attached video cameras.



Fixed Video Standard

○ Normal Operation

○ NTSC Only

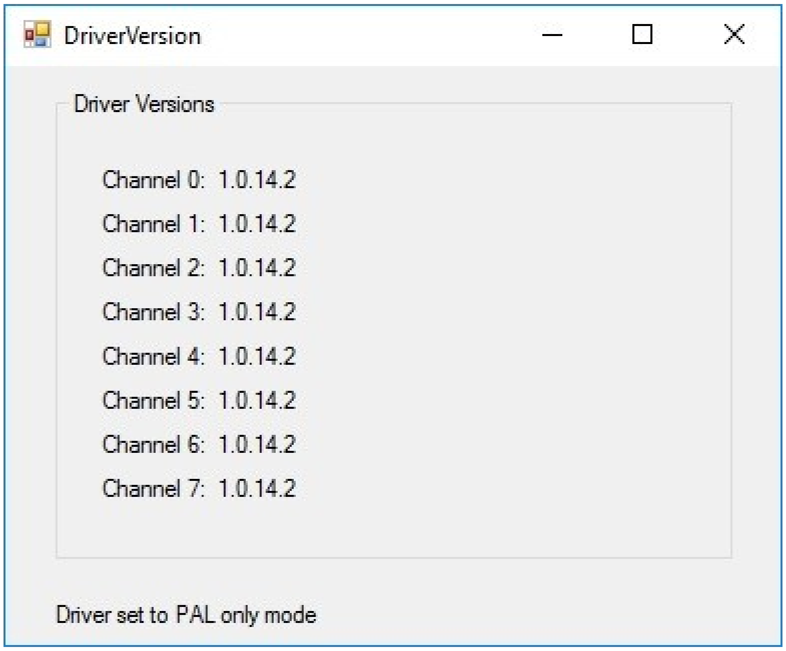
● PAL Only

Note: This setting affects all boards in the system.

- Click Apply to change the settings

Verifying PAL-only or NTSC-only setting

- In the demo application, click on Tools→Driver Settings.



Driver Versions

Channel 0: 1.0.14.2

Channel 1: 1.0.14.2

Channel 2: 1.0.14.2

Channel 3: 1.0.14.2

Channel 4: 1.0.14.2

Channel 5: 1.0.14.2

Channel 6: 1.0.14.2

Channel 7: 1.0.14.2

Driver set to PAL only mode

- PAL only or NTSC only setting will be shown at the bottom of the dialog. The displayed driver version should be 1.0.14.2 or higher. If not, the setting can not be verified. Re-install the SDK by running setup.exe, and follow the prompts to update the driver.