

Pro Capture Series Card

Software Development Guide

Glossary.....	2
1 Overview.....	3
1.1 Main Features	3
1.2 Development Method	5
2 DirectShow Extension Interface	6
2.1 Development Process	6
2.3 Examples List.....	10
3 MWCapture Development.....	11
3.1 Module Introduction.....	11
3.2 MWCapture Functions	12
3.2.1 Function List.....	12
3.2.2 Capture video data at a fixed frame rate.....	15
3.2.3 Capture video data at the frame rate of the input signal	16
3.2.4 Example List	17
4 About	18

Glossary

Capture Device	A computer device for capturing video and audio can be a PCI-e card, USB 3.0 Box/dongle, or Mini PCI-e card.
Product Family	Categorize different capture devices.
Product Model	Using product model to distinguish the different capture device according to their functions.
Hardware Version	If there are different hardware versions of the same model of capture devices, capital letters such as A, B, C are used to distinguish them.
Firmware	The software that stored in capture device chip.
Firmware ID	There can be different firmware for the same model of capture devices to meet the needs of different application. Firmware ID is used to distinguish different firmware.
Hardware Serial NO.	Consists of 13 characters (1 letter and 12 numbers). Unique factory-set number for the capture device.
Capture Channel	Basic unit to capture video/audio signals. A channel usually contains a video channel and an audio channel.
Rotary Switch	The value of rotary switch on the board. The value is from 0 to F.
Channel Index	On a capture device, different capture channels are distinguished by index. Channel index starts from 0.
Input Source	The interface on a capture device to connect to the signal source. Common video input interfaces include HDMI, SDI, VGA and DVI. Common audio input interfaces include HDMI, SDI and Line In.
DMA Transfer	Using the onboard FPGA chip to write data in the target memory area.
Input Source Scanning	If a video channel or audio channel has multiple input sources, it will automatically scan and choose the input source when started or signal changed. The priority order is SDI, HDMI, YPbPr.

1 Overview

SDK is mainly used for developing application software with Pro Capture Series Card. SDK directly access Pro Capture Series card through MWCapture interfaces. By using this SDK, users can use all features of Pro Capture Series Card and have the best performance and better flexibility. DirectShow extension interfaces are also provided for DirectShow users to access extended features.

This document mainly introduces the functions of software development kit. For the type definitions and specific functions, please refer to "Software Development Manual" .

1.1 Main Features

Multi-Streaming Output	One capture channel can output multiple streams in different resolutions.
	Each capture channel's transmission bandwidth is 400 MB.
High-precision Clock	Produced by the onboard crystal oscillator and the smallest unit is 100 nanoseconds.
	Provide universal clock for capture device.
	Timestamps are marked on all frames. It helps to synchronize video and audio.
	Support to set up and adjust clock. It helps in synchronization of multiple capture devices. Those devices can be in one computer or also be in multiple computers in the network.
Asynchronous Event Notification	Set the scheduled time to get the asynchronous event notifications when timeout occurs.
	When input signal is changed, or input signal format is changed, or input video frame has been in the on-board buffer, or input signal is reset, the asynchronous event notification can be gotten.
	When DMA transfer finished, the asynchronous event notification can be gotten.
DMA Transfer	Using the onboard FPGA chip to write capture data in the target storage area. It can reduce the CPU usage.
	Transfer video data to the video memory directly. It will reduce the CPU and memory usage and it is compatible with AMD and NVIDIA graphics cards.
	Transfer video data to physical memory address of the third party device directly.
	Support to transfer data in blocks. It will reduce the capture data time.
Original Device Information	Get all kinds of input signal information, including signal format, signal status and HDMI Info Frame.

Get the device original information, including the on-board buffer state, the accurate time of frame arrival in device, the time of frame filled, etc.

1.2Development Method

DirectShow Interface

If users application software is based DirectShow, using DirectShow interface of SDK is the best choice. It can keep user application software stable on the structure. By calling extension interface, users can use all the extension features of Pro Capture Filmily.

Development based on DirectShow has the following features.

- Application has good compatibility and can be compatible with various capture devices.
- The demand of the developers is higher. It requires enough background knowledge and experience.
- Development process is relatively complex, but it has better support for C++. It is not friendly to other development language.

MWCapture Functions

Calling functions to develop application software. Development difficulty is relatively low. At the same time, product architecture can keep good flexibility and developers don' t need to understand many principles of multimedia development.

Development based on MWCapture has the following features.

- Development difficulty is low. The development cycle is fast. The product architecture has a good flexibility.
- Support for multiple development language, including C++, VB, C#, Java, etc.
- Developed software can only support some particular capture devices.
- API functions use hierarchical design, suitable for different developers;

Choose a Suitable Development Language

SDK is compiled by Microsoft Visual Studio 2010(Version 10.0.40219.1 SP1Rel). Its develop language is C++. SDK provide several example projects to demonstrate the various functions of SDK interface. SDK supports multiple development languages, including C++, VB, C#, Java, HTML5, WebRTC, etc.

2 DirectShow Extension Interface

2.1 Development Process

The process demonstrates how to call the device driver functions in DirectShow to realization of all sorts of capture device functions.

1	Create video capture filter	IMoniker::BindToObject
2	GetIMWCaptureExtension Interface	IBaseFilter::QueryInterface
3	Call IMWCaptureExtension functions	Call IMWCaptureExtensioninterface functions
4	Release extension interface point	IMWCaptureExtension::Release

2.2Function List

IMWCAaptureExtension implement almost all functions, Please refer to the example code FilterProperty.

Channel Functions

GetChannelInfo	Get capture channel information.
GetFamilyInfo	Get device family information.
GetVideoCaps	Get video channel capacity.
GetAudioCaps	Get audio channel capacity.
GetDeviceInstanceID	Get channel' s system device ID.
RegisterNotify	Register asynchronous event.
UnregisterNotify	Unregister asynchronous event.
GetNotifyStatus	Get event notify status.
GetVideoInputSourceCount	Get number of all the video input interfaces
GetVideoInputSourceArray	Get video input interfaces
GetAudioInputSourceCount	Get number of all the audio input interfaces.
GetAudioInputSourceArray	Get audio input interfaces .
SetInputSourceScan	Set input interface scan mode of the capture channel.
GetInputSourceScan	Get input interface scan mode of the capture channel.
GetInputSourceScanState	Get input interface scan status of the capture channel.
SetAVInputSourceLink	Set the status of link between the current audio channel and video channel
GetAVInputSourceLink	Get the status of link between the current audio channel and video channel
SetVideoInputSource	Set the input interface for the current video channel.
GetVideoInputSource	Get the input interface for the current video channel.
SetAudioInputSource	Set the input interface for the current audio channel.
GetAudioInputSource	Get the input interface for the current audio channel.

EDID Functions

SetEDID	Set EDID data of HDMI interface.
GetEDID	Get EDID data of HDMI interface.

Signal Status Functions

GetInputSpecificStatus	Get the status of the input signal.
GetVideoSignalStatus	Get the status of input video signal.
GetAudioSignalStatus	Get the status of input audio signal.
GetHDMIInfoFrameValidFlags	Get the status of HDMI InfoFrame valid.
GetHDMIInfoFramePacket	Get the status of HDMI InfoFrame data.

Video Process Functions

SetVideoInputAspectRatio	Set the aspect ratio of the video input channel.
--------------------------	--

GetVideoInputAspectRatio	Get the aspect ratio of the video input channel.
SetVideoInputColorFormat	Set the color format of the video input channel.
GetVideoInputColorFormat	Get the color format of the video input channel.
SetVideoInputQuantizationRange	Set the quantization range of the video input channel.
GetVideoInputQuantizationRange	Get the quantization range of the video input channel.
GetVideoCaptureConnectionFormat	Get the video capture connection format of the current channel.
SetVideoCaptureProcessPreset	Set the preset of video capture process.
GetVideoCaptureProcessPreset	Get the preset of video capture process.
SetVideoCaptureProcessSettings	Set the settings of video capture process.
GetVideoCaptureProcessSettings	Get the settings of video capture process.
GetVideoPreviewConnectionFormat	Get the video preview format.
SetVideoPreviewProcessPreset	Set the preset of the video preview process.
GetVideoPreviewProcessPreset	Get the preset of the video preview process.
SetVideoPreviewProcessSettings	Set the video preview process settings.
GetVideoPreviewProcessSettings	Get the video preview process settings.

OSD Functions

SetVideoCaptureOSDPreset	Set video capture OSD preset.
GetVideoCaptureOSDPreset	Get video capture OSD preset.
SetVideoCaptureOSDSettings	Set video capture OSD settings.
GetVideoCaptureOSDSettings	Get video capture OSD settings.
SetVideoPreviewOSDPreset	Set video preview OSD preset.
GetVideoPreviewOSDPreset	Get video preview OSD preset.
SetVideoPreviewOSDSettings	Set video preview OSD settings.
GetVideoPreviewOSDSettings	Get video preview OSD settings.
GetCoreTemperature	Get the card current temperature.

Video Timing Functions

SetVideoAutoHAlign	Set whether to conduct horizontal alignment of the video signal automatically.
GetVideoAutoHAlign	Get the setting of whether to conduct horizontal alignment of the video signal automatically.
SetVideoSamplingPhase	Set the video sampling phase.
GetVideoSamplingPhase	Get the video sampling phase.
SetVideoSamplingPhaseAutoAdjust	Set whether to conduct adjustment of the video sampling phase automatically.
GetVideoSamplingPhaseAutoAdjust	Get whether to conduct adjustment of the video sampling phase automatically.
SetVideoTiming	Set the preset video timing.
GetPreferredVideoTimings	Get the preset video timings.
SetCustomVideoTiming	Set the custom video timing.
GetCustomVideoTimingsCount	Get the number of the custom video timings.

GetCustomVideoTimingsArray	Get all the custom video timings.
SetCustomVideoTimingsArray	Get all custom video timings.
GetCustomVideoResolutionsCount	Get the number of resolutions of custom video.
GetCustomVideoResolutionsArray	Get resolutions of all the custom video channels.
SetCustomVideoResolutionsArray	Get resolutions of all the custom video channels.

2.3 Examples List

FilterProperty	The example demonstrated how to access all capture device property by calling IMWCaptureExtension interface.
DShowCapture	The example demonstrates how to call DirectShow API to enumerate, capture and preview. It also demonstrates calling Windows Media SDK to compress video and audio to ASF files.

3 MWCapture Development

3.1Module Introduction

Functions consist of the following modules.

LibMWCapture	Functions	Device enumeration and capture data from the video channel
	Head File	LibMWCapture \ MWCapture.h
	Lib File	LibMWCapture(d).lib
	DLL File	LibMWCapture(d).dll

3.2 MWCapture Functions

Device Access Functions is based on MWCapture. It encapsulates the access to capture device driver, mainly to conduct the device enumeration, show signal status, use onboard clock, capture A/V data from the channel, upgrade firmware, etc.

3.2.1 Function List

Initialize Functions

MWCaptureInitInstance	Initialize SDK
MWCaptureExitInstance	Exit

Enumerate Devices Functions

These functions are used to check whether there is available capture device on the current computer and to get the device information.

MWRefreshDevice	Enumerate capture device on the current computer.
MWGetChannelCount	Get number of the capture channels.
MWGetChannelInfoByIndex	Get capture channel information by capture channel index.
MWGetFamilyInfoByIndex	Get capture device family information by capture channel index.
MWGetDevicePath	Get device path.

Onboard Clock Functions

MWGetDeviceTime	Get the current time from onboard clock.
MWSetDeviceTime	Set the current time of the clock.
MWRegulateDeviceTime	Adjust the time of clock.
MWRegisterTimer	Register timeout asynchronous event.
MWUnregisterTimer	Unregister timeout asynchronous event.
MWScheduleTimer	Schedule timeout asynchronous event.

Channel Functions

MWOpenChannel	Open the specified capture channel
MWCloseChannel	Close capture channel.
MWOpenChannelByPath	Open capture channel by device path.
MWGetChannelInfo	Get capture channel information.
MWGetFamilyInfo	Get device family information.
MWRegisterNotify	Register asynchronous event.
MWUnregisterNotify	Unregister asynchronous event.
MWGetNotifyStatus	Get status of the event notify.

MWGetInputSpecificStatus	Get input signal status.
MWGetVideoSignalStatus	Get input video signal status.
MWGetAudioSignalStatus	Get input audio signal status.
MWGetVideoCaps	Get video channel capacity.
MWGetAudioCaps	Get audio channel capacity.
MWGetVideoInputSourceArray	Get all the input sources of video channels.
MWGetAudioInputSourceArray	Get all the input sources of audio channels.
MWGetInputSourceScan	Get capture channel input source scan status.
MWSetInputSourceScan	Set status of capture channel input source scan.
MWGetVideoInputSource	Get the current input source of the video channel.
MWGetAudioInputSource	Get the current input source of the audio channel.
MWSetVideoInputSource	Set the current input source of the video channel.
MWSetAudioInputSource	Set the current input source of the audio channel.
MWGetAVInputSourceLink	Get the status of the link between the current audio channel and video channel.
MWSetAVInputSourceLink	Set the status of the link between the current audio channel and video channel.

Video Capture Functions

MWStartVideoCapture	Start to capture video data.
MWStopVideoCapture	Stop capturing video data.
MWPinVideoBuffer	Pin the virtual memory of the video buffer to reduce CPU usage.
MWUnpinVideoBuffer	Unpin the virtual memory previously pinned.
MWCaptureVideoFrameToVirtualAddress	Capture and transfer data to virtual memory address by DMA transmission.
MWCaptureVideoFrameToPhysicalAddress	Capture and transfer data to physics address by DMA transmission.
MWCaptureVideoFrameWithOSDToVirtualAddress	Capture video frame with OSD and transfer to virtual memory address by DMA transmission
MWCaptureVideoFrameWithOSDToPhysicalAddress	Capture video frame with OSD and transfer to physics address by DMA transmission
MWCaptureVideoFrameToVirtualAddressEx	Capture data to memory address by DMA transmission. Support aspect ratio, deinterlace mode, image cropping, etc.
MWCaptureVideoFrameToPhysicalAddressEx	Capture data to physics address by DMA transmission. Support adjustment of aspect ratio, choosing deinterlace mode, image cropping, etc.

MWGetVideoCaptureStatus	Get video capture status.
MWGetVideoFrameInfo	Get frame buffer information of video capture channel.

Audio Capture Functions

MWStartAudioCapture	Start to capture audio data.
MWStopAudioCapture	Stop capture audio data.
MWCaptureAudioFrame	Capture audio data.

Video Processing Functions

MWSetVideoInputAspectRatio	Set video input aspect ratio.
MWGetVideoInputAspectRatio	Get video input aspect ratio.
MWSetVideoInputColorFormat	Set video input color format
MWGetVideoInputColorFormat	Get video input color format
MWSetVideoInputQuantizationRange	Set video input quantization range.
MWGetVideoInputQuantizationRange	Get video input quantization range.

Upgrade Firmware Functions

MWGetFirmwareStorageInfo	Get current firmware storage information.
MWEraseFirmwareData	Erase the onboard firmware data.
MWReadFirmwareData	Read firmware data for data verification and firmware exporting.
MWWriteFirmwareData	Write firmware data.
MWSetPostReconfig	Set to load new firmware.

Other Functions

MWGetVersion	Get SDK version.
MWSetLEDMode	Set onboard LED status.
MWGetEDID	Get EDID data from HDMI interface.
MWSetEDID	Set EDID data to HDMI interface.
MWGetHDMIInfoFrameValidFlag	Get HDMI info frame valid flag.
MWGetHDMIInfoFramePacket	Get HDMI info frame data
MWCreateImage	Create an image
MWOpenImage	Open an image.
MWCloseImage	Close an image.
MWUploadImageFromVirtualAddress	Upload an image from virtual address
MWUploadImageFromPhysicalAddress	Upload an image from physical address.
MWCreateExtendObject	Get extend information from the card.

3.2.2 Capture video data at a fixed frame rate

The procedure demonstrates how to capture video data at a fixed frame rate. Please refer to the example code "Examples \ CmdLineTools \ CaptureByTimer".

1	Enumerate Devices	MWRefreshDevice MWGetChannelCount MWGetChannelInfo MWGetDevicePath
2	Open Capture Channel	MWOpenChannel MWOpenChannelByPath
3	Startto Capture	MWStartVideoCapture
4	Register Timer	MWRegisterTimer
5	Schedule Timer	MWScheduleTimer
6	Wait for Timer Event	WaitForSignalObject / WaitForMultiObject (Win32 Functions)
7	Capture Video Data	MWCaptureVideoFrameToVirtualAddress MWCaptureVideoFrameToPhysicalAddress
8	Get video Capture Status	MWGetVideoCaptureStatus
9	Unregister Timer	MWUnRegisterTimer
10	Stop Video Capture	MWStopVideoCapture
11	Close Capture Channel	MWCloseChannel

3.2.3 Capture video data at the frame rate of the input signal

The procedure demonstrates how to capture video data at the frame rate of the input signal. Please refer to the example code "Examples \ CmdLineTools \ CaptureByInput".

1	Enumerate Devices	MWRefreshDevice MWGetChannelCount MWGetChannelInfo MWGetDevicePath
2	Open Channel	MWOpenChannel MWOpenChannelByPath
3	Startto Capture	MWStartVideoCapture
4	RegisterNotify Event	MWRegisterNotify
5	Wait for Notify Event	WaitForSignalObject / WaitForMultiObject (Win32 Functions)
6	Capture Video Data	MWCaptureVideoFrameToVirtualAddress MWCaptureVideoFrameToPhysicalAddress
8	Get video capture status	MWGetVideoCaptureStatus
9	UnregisterNotify Event	MWUnRegisterNotify
10	Stop Video Capture	MWStopVideoCapture
11	Close Capture Channel	MWCloseChannel

3.2.4Example List

CaptureByInput	The example demonstrates how to use video capture functions to capture video data at the input signal frame rate and create a snapshot.
CaptureByTimer	The example demonstrates how to use video capture functions to capture video data at a fixed signal frame rate and create a snapshot.
HDMIInfoFrame	The example demonstrates how to use signal status functions to get HDMI signal' s InfoFrame data.
InputSignal	The example demonstrates how to use signal status functions to get input signal status.
InputSource	The example demonstrates how to use signal source functions to get information of signal source.
ReadWriteEDID	The example demonstrates how to use EDID functions to read EDID information from a file or write EDID information to a file.
AudioCapture	The example demonstrates how to use audio capture functions to capture dual-channel audio data and record it into a wav file of 4 seconds .
AVCapture	The example demonstrates how to capture video and display video in windows. It also demonstrates how to clip video in capturing process.
LowLatency	The example demonstrates how to capture video deal with low latency and display video in windows.
MultiAudioCapture	The example demonstrates how to capture multi audio channel and record it as a wav file.
MultiStreaming	The example demonstrates how to capture multi video and display them in windows. It also can set different resolution ratio.
NDISender	The example demonstrates how to capture video, audio and display video in windows. It also can send the data to the NDI receiver by the internet.
OSDPreview	The example demonstrates how to capture video with OSD image and display video in windows
XICaptureQuad	The example demonstrates how to synchronous capture multi video channels and display video in windows

4 About

Please visit the following website to get the latest version.

<http://www.magewell.com/sdk>

If you have any questions, please send an email to support@magewell.net.

DirectShow SDK (<http://msdn.microsoft.com>) are compatible.

NDI (<http://NDI.NewTek.com>) are compatible.