

Pro Capture 系列采集卡

软件开发指南

目录

术语表	2
1 简介	3
1.1 主要特性	3
1.2 开发方式	4
2 DirectShow 扩展接口	5
2.1 开发流程	5
2.2 函数列表	6
2.3 示例列表	8
3 MWCapture 开发	9
3.1 模块介绍	9
3.2 函数开发	10
3.2.1 函数列表	10
3.2.2 固定帧率采集视频	13
3.2.3 输入信号帧率采集视频	14
3.2.4 示例列表	15
4 关于	16

术语表

采集设备	一个用于采集音视频信号的计算机装置，可以是 PCI-e 采集卡、USB 3.0 采集盒/棒、或者 Mini PCI-e 采集卡。
产品系列	对采集设备进行系列划分，每个产品系列包含多个产品类型。
产品类型	按照不同功能类型对采集设备进行划分，使用不同的 ID 加以区分。
硬件版本	对于同一种类型的采集设备，如果在硬件上有过更新升级，就通过硬件版本来区分，硬件版本使用一个大写英文字母区分:A、B、C.....
固件	存储在采集设备存储芯片上的软件。
固件 ID	对于同一种类型的采集设备，对于不同的应用场景可能存在不同的固件类型，固件类型使用固件 ID 区分。
硬件序列号	由 13 个字符组成（1 个字母加 12 个数字），每个采集设备在出厂时设定唯一的硬件序列号。
采集通道	采集音视频信号的基本单元，一个通道通常包含一个视频通道和一个音频通道。在一个采集设备上可能会有多个通道。
设备开关序号	采集设备上旋钮开关指示的序号，序号从 0 到 F。
通道序号	在一个采集设备上，不同的采集通道通过序号加以区分，采集通道序号均从 0 开始累计。
输入接口	采集设备上用于和信号源相连接的接口，常见的视频输入接口有 HDMI、SDI、VGA、DVI 等，常见的音频输入接口有 HDMI、SDI、Line In 等。
DMA 传输	使用板载芯片实现采集数据向目标内存区域的写入操作。
接口扫描	如果视频通道或音频通道存在多个输入接口，当信号发生变化时，通道会自动扫描并选择有信号的接口，接口的优先级顺序为：SDI，HDMI，YPbPr。

1 简介

Pro Capture 软件开发包主要用于开发 Pro Capture 系列采集卡的应用软件。它直接通过 MWCapture 函数接口访问 Pro Capture 系列采集卡的驱动程序，基于它来开发应用软件可以实现 Pro Capture 系列采集卡的全部功能特性，并且可以获得最佳的性能和更好的灵活性。同时我们也提供 DirectShow 扩展接口，方便基于 DirectShow 开发的应用程序去使用 Pro Capture 系列采集卡的扩展功能。
本文主要对软件开发包的各项功能进行介绍，相关类型定义和函数说明请参阅“软件开发手册”。

1.1 主要特性

多码流输出	同一采集通道支持任意多个分辨率不同的视频流输出。
	每个采集通道的传输总带宽达到 400MB。
高精度时钟同步	由采集设备的板载晶振产生，精度为 100 纳秒。
	为当前采集设备提供统一时钟信息。
	所有音视频帧均标记对应的时间信息，便于进行音视频同步。
	支持设置和校准时钟，便于多个采集设备之间进行时钟同步，这些采集设备可以在同一台计算机上，也可以分布在网络上的多台计算机中。
异步事件通知	可以设置调度时间，在超时发生时获得异步事件通知。
	在输入信号发生改变、输入信号格式发生改变、输入视频帧已填入板载缓冲区、输入信号被重置等情况发生时，可获得异步事件通知。
	当使用 DMA 进行采集数据填充完毕后，可获取异步事件通知。
DMA 数据传输	使用板载 FPGA 芯片完成采集数据向目标存储区域的写入操作，降低对系统 CPU 的占用。
	可以直接向显卡的显存传输采集的视频数据，降低对系统 CPU 和内存资源占用，可兼容 AMD、NVIDIA 系列显卡。
	可以向第三方设备物理内存地址高速传输采集数据。
	支持采集数据分块传输，可以有效降低数据采集的延时。
原始设备信息	可以获取各种信号源的原始信息，包括信号格式、信号状态、HDMI Info frame 等。
	可以获得采集设备上的各种采集原始信息，包括板载缓冲区数据填充状态、数据帧到达采集设备的时间、数据帧填充完毕的时间等。

1.2 开发方式

DirectShow 开发

如果用户的应用软件是基于 DirectShow 构建，使用 DirectShow 扩展接口是最佳的选择。它可以最大程度保持用户应用软件在结构上的稳定，只需要调用扩展接口，即可使用 Pro Capture 系列采集卡的所有扩展功能特性。

基于 DirectShow 扩展接口开发有以下特点：

- 应用程序的兼容性好，能兼容各种不同采集设备；
- 对开发人员的要求较高，需要足够的背景知识和开发经验；
- 开发过程较复杂，对 C++ 支持较好，其它开发语言支持不够友好；

MWCapture 开发

封装为函数接口调用进行软件开发，其开发难度相对较低，在产品架构上可以保持较大的灵活性，开发人员不需要对多媒体开发的原理有深入的了解，即可开发出功能出众的产品。

基于 MWCapture 开发有以下特点：

- 开发难度较低，开发周期快，产品架构上具有较大的灵活性；
- 支持多种开发语言，包括 C++、VB、C#、Java、HTML5、WebRTC 等；
- 开发出来的软件只能支持特定的采集设备；
- 函数使用分层设计，适用于不同的开发人员；

开发环境

开发包模块使用 Microsoft Visual Studio 2013 (Version 12.0.21005.1) 编译，示例代码使用 Microsoft Visual Studio 2010 (Version 10.0.40219.1 SP1Rel) 编译，开发语言为 C++，SDK 提供了多个命令行和窗口示例展示了开发包各种功能接口的调用，支持多种开发语言，包括 C++、VB、C#、Java、HTML5、WebRTC 等。

2 DirectShow 扩展接口

2.1 开发流程

以下步骤展示了在 DirectShow 中调用扩展接口的设备驱动函数实现各种采集的功能。

1	创建视频采集过滤器	IMoniker::BindToObject
2	获取 IMWCaptureExtension 接口	IBaseFilter::QueryInterface
3	调用 IMWCaptureExtension 接口函数	Call IMWCaptureExtensioninterface functions
4	释放扩展接口指针	IMWCaptureExtension::Release

2.2 函数列表

IMWCaptureExtension 接口实现全部的功能调用，具体请参考示例代码 FilterProperty。

通道函数

GetChannelInfo	获取采集通道的信息。
GetFamilyInfo	获取采集通道的产品系列信息。
GetVideoCaps	获取视频通道的性能属性。
GetAudioCaps	获取音频通道的性能属性。
GetDeviceInstanceID	获取通道对应的系统设备 ID。
RegisterNotify	注册异步事件通知对象。
UnregisterNotify	注销异步事件通知对象。
GetNotifyStatus	获取事件通知的状态。
GetVideoInputSourceCount	获取视频通道的输入接口数目。
GetVideoInputSourceArray	获取视频通道的输入接口。
GetAudioInputSourceCount	获取音频通道的输入接口数目。
GetAudioInputSourceArray	获取音频通道的输入接口。
SetInputSourceScan	设置采集通道的输入接口扫描类型。
GetInputSourceScan	获取采集通道的输入接口扫描类型。
GetInputSourceScanState	获取采集通道的输入接口扫描状态。
SetAVInputSourceLink	设置当前音频通道是否和当前视频通道一致。
GetAVInputSourceLink	获取当前音频通道是否和当前视频通道一致。
SetVideoInputSource	设置视频通道当前使用的输入接口。
GetVideoInputSource	获取视频通道当前使用的输入接口。
SetAudioInputSource	设置音频通道当前使用的输入接口。
GetAudioInputSource	获取音频通道当前使用的输入接口。

EDID 函数

SetEDID	设置 HDMI 接口的 EDID 数据。
GetEDID	获取 HDMI 接口的 EDID 数据。

信号状态函数

GetInputSpecificStatus	获取输入信号的状态。
GetVideoSignalStatus	获取视频输入信号的状态。
GetAudioSignalStatus	获取音频输入信号的状态。
GetHDMIInfoFrameValidFlags	获取 HDMI InfoFrame 有效状态。
GetHDMIInfoFramePacket	获取 HDMI InfoFrame 数据。

视频处理函数

SetVideoInputAspectRatio	设置视频输入信号的宽高比。
GetVideoInputAspectRatio	获取视频输入信号的宽高比。
SetVideoInputColorFormat	设置视频输入信号的色彩空间格式。
GetVideoInputColorFormat	获取视频输入信号的色彩空间格式。

SetVideoInputQuantizationRange	设置视频输入信号的量化范围。
GetVideoInputQuantizationRange	获取视频输入信号的量化范围。
GetVideoCaptureConnectionFormat	获取视频采集格式信息。
SetVideoCaptureProcessPreset	设置视频采集处理预设参数。
GetVideoCaptureProcessPreset	获取视频采集处理预设参数。
SetVideoCaptureProcessSettings	设置视频采集处理配置参数。
GetVideoCaptureProcessSettings	获取视频采集处理配置参数。
GetVideoPreviewConnectionFormat	获取视频预览格式信息。
SetVideoPreviewProcessPreset	设置视频预览处理预设参数。
GetVideoPreviewProcessPreset	获取视频预览处理预设参数。
SetVideoPreviewProcessSettings	设置视频预览处理配置参数。
GetVideoPreviewProcessSettings	获取视频预览处理配置参数。

OSD 函数

SetVideoCaptureOSDPreset	设置视频采集 OSD 预设参数。
GetVideoCaptureOSDPreset	获取视频采集 OSD 预设参数。
SetVideoCaptureOSDSettings	设置视频采集 OSD 配置参数。
GetVideoCaptureOSDSettings	获取视频采集 OSD 配置参数。
SetVideoPreviewOSDPreset	设置视频预览 OSD 预设参数。
GetVideoPreviewOSDPreset	获取视频预览 OSD 预设参数。
SetVideoPreviewOSDSettings	设置视频预览 OSD 配置参数。
GetVideoPreviewOSDSettings	获取视频预览 OSD 配置参数。
GetCoreTemperature	获取板载卡的当前温度。

视频时序函数

SetVideoAutoHAlign	设置视频水平方向是否自动调整。
GetVideoAutoHAlign	获取视频水平方向是否自动调整。
SetVideoSamplingPhase	设置视频采样相位大小。
GetVideoSamplingPhase	获取视频采样相位大小。
SetVideoSamplingPhaseAutoAdjust	设置视频采样相位是否自动调整。
GetVideoSamplingPhaseAutoAdjust	获取视频采样相位是否自动调整。
SetVideoTiming	设置视频时序参数。
GetPreferredVideoTimings	获取预设置视频时序参数。
SetCustomVideoTiming	设置自定义视频时序参数。
GetCustomVideoTimingsCount	获取自定义视频时序个数。
GetCustomVideoTimingsArray	获取自定义视频时序接口数目。
SetCustomVideoTimingsArray	设置自定义视频时序接口。
GetCustomVideoResolutionsCount	获取自定义视频分辨率个数。
GetCustomVideoResolutionsArray	获取所有自定义视频分辨率。
SetCustomVideoResolutionsArray	设置所有自定义视频分辨率。

2.3 示例列表

FilterProperty	示例展示了 DirectShow 架构下调用 IMWCaptureExtension 接口访问所有采集设备属性的过程。
DShowCapture	示例展示了调用 DirectShow 进行音视频设备的枚举、采集和预览 ,并调用 Windows Media SDK 压缩生成 ASF 文件的全过程。

3 MWCapture 开发

3.1 模块介绍

LibMWCapture	功能介绍	主要实现设备枚举、视频通道采集等功能。
	头文件	LibMWCapture \ MWCapture.h
	库文件	LibMWCapture(d).lib
	执行文件	LibMWCapture(d).dll

3.2 函数开发

设备访问函数基于 MWCapture 实现，封装了对采集设备驱动程序的访问，主要实现设备枚举、显示信号状态、板载时钟、通道采集、固件升级等功能。

3.2.1 函数列表

初始化函数

MWCaptureInitInstance	初始化 SDK。
MWCaptureExitInstance	退出。

设备枚举函数

查找当前计算机上是否有可用的采集设备，并可获取设备相关信息。

MWRefreshDevice	枚举当前计算机上的采集设备。
MWGetChannelCount	获取采集设备的采集通道数量。
MWGetChannelInfoByIndex	通过采集通道序号获取采集通道的详细信息。
MWGetFamilyInfoByIndex	通过采集通道序号获取采集设备的产品系列信息。
MWGetDevicePath	获取设备路径。

板载时钟函数

MWGetDeviceTime	从板载时钟获取当前时间。
MWSetDeviceTime	设置时钟的当前时间。
MWRegulateDeviceTime	校准时钟的时间。
MWRegisterTimer	注册时间异步对象。
MWUnregisterTimer	注销时间异步对象。
MWScheduleTimer	调度时间异步对象。

通道函数

MWOpenChannel	打开指定的采集通道。
MWCloseChannel	关闭采集通道。
MWOpenChannelByPath	使用设备路径打开视频通道。
MWGetChannelInfo	获取采集通道的信息。
MWGetFamilyInfo	获取采集通道的产品系列信息。
MWRegisterNotify	注册异步事件通知对象。
MWUnregisterNotify	注销异步事件通知对象。
MWGetNotifyStatus	获取事件通知的状态。
MWGetInputSpecificStatus	获取输入信号的状态。
MWGetVideoSignalStatus	获取视频输入信号的状态。
MWGetAudioSignalStatus	获取音频输入信号的状态。
MWGetVideoCaps	获取视频通道的性能属性。
MWGetAudioCaps	获取音频通道的性能属性。

MWGetVideoInputSourceArray	获取视频通道的所有输入源。
MWGetAudioInputSourceArray	获取音频通道的所有输入源。
MWGetInputSourceScan	获取采集通道的输入源扫描状态。
MWSetInputSourceScan	设置采集通道的输入接口扫描状态。
MWGetVideoInputSource	获取视频通道当前使用的输入源。
MWGetAudioInputSource	获取音频通道当前使用的输入源。
MWSetVideoInputSource	设置视频通道当前使用的输入源。
MWSetAudioInputSource	设置音频通道当前使用的输入源。
MWGetAVInputSourceLink	获取当前音频通道是否和当前视频通道一致。
MWSetAVInputSourceLink	设置当前音频通道是否和当前视频通道一致。

视频采集函数

MWStartVideoCapture	开始视频通道的采集操作。
MWStopVideoCapture	关闭视频通道采集操作。
MWPinVideoBuffer	锁定目的地址以减少 CPU 占用。
MWUnpinVideoBuffer	解锁目的地址。
MWCaptureVideoFrameToVirtualAddress	使用 DMA 采集数据到指定的虚拟内存地址中。
MWCaptureVideoFrameToPhysicalAddress	使用 DMA 采集数据到指定的物理内存地址中。
MWCaptureVideoFrameWithOSDToVirtualAddress	使用 DMA 采集合成 OSD 图片后的数据到指定的虚拟内存地址中。
MWCaptureVideoFrameWithOSDToPhysicalAddress	使用 DMA 采集合成 OSD 图片后的数据到指定的物理内存地址中。
MWCaptureVideoFrameToVirtualAddressEx	使用 DMA 采集数据到指定的内存地址中，支持纵宽比、隔行处理、裁剪等处理。
MWCaptureVideoFrameToPhysicalAddressEx	使用 DMA 采集数据到指定的物理内存地址中，支持纵宽比、隔行处理、裁剪等处理。
MWGetVideoCaptureStatus	获取当前采集状态。
MWGetVideoFrameInfo	获取视频采集通道的缓冲区信息。

音频采集函数

MWStartAudioCapture	开始音频通道的采集操作。
MWStopAudioCapture	关闭音频通道采集操作。
MWCaptureAudioFrame	采集音频数据。

视频处理函数

MWSetVideoInputAspectRatio	设置输入图像的纵宽比。
MWGetVideoInputAspectRatio	获取已设置的输入图像的纵宽比。
MWSetVideoInputColorFormat	设置输入图像的色彩空间。
MWGetVideoInputColorFormat	获取输入图像的色彩空间。
MWSetVideoInputQuantizationRange	设置输入图像的量化范围。
MWGetVideoInputQuantizationRange	获取输入图像的量化范围。

固件升级函数

MWGetFirmwareStorageInfo	获取当前固件存储信息。
MWEraseFirmwareData	擦除固件数据。
MWReadFirmwareData	读取固件数据，用于数据校验和固件导出。
MWWriteFirmwareData	写入固件数据。
MWSetPostReconfig	设置加载固件。

其他函数

MWGetVersion	获取 SDK 版本号。
MWSetLEDMode	设置板载 LED 灯状态。
MWGetEDID	获取 HDMI 接口的 EDID 数据。
MWSetEDID	设置 HDMI 接口的 EDID 数据。
MWGetHDMIInfoFrameValidFlag	获取 HDMI 信息帧的有效状态。
MWGetHDMIInfoFramePacket	获取 HDMI 信息帧率数据。
MWCreateImage	创建 OSD 图像。
MWOpenImage	打开 OSD 图像。
MWCloseImage	关闭 OSD 图像
MWUploadImageFromVirtualAddress	从虚拟内存上载图像到采集设备。
MWUploadImageFromPhysicalAddress	从物理内存上载图像到采集设备。
MWCreateExtendObject	获取板载扩展信息。

3.2.2 固定帧率采集视频

本流程展示了按指定帧率进行视频采集的函数调用过程。请参考示例代码“Examples \ CmdLineTools \ CaptureByTimer”。

1	设备枚举	MWRefreshDevice MWGetChannelCount MWGetChannelInfo MWGetDevicePath
2	打开采集通道	MWOpenChannel MWOpenChannelByPath
3	开始视频通道采集	MWStartVideoCapture
4	注册定时器	MWRegisterTimer
5	调度定时器	MWScheduleTimer
6	等待事件通知	WaitForSignalObject / WaitForMultiObject (Win32 Functions)
7	采集视频图像数据	MWCaptureVideoFrameToVirtualAddress MWCaptureVideoFrameToPhysicalAddress
8	获取采集状态	MWGetVideoCaptureStatus
9	注销定时器	MWUnRegisterTimerEvent
10	停止视频通道采集	MWStopVideoCapture
11	关闭采集通道	MWCloseChannel

3.2.3 输入信号帧率采集视频

本流程展示了按信号输入帧率进行视频采集的函数调用过程。请参考示例代码“Examples \ CmdLineTools \ CaptureByInput”。

1	设备枚举	MWRefreshDevice MWGetChannelCount MWGetChannelInfo MWGetDevicePath
2	打开采集通道	MWOpenChannel MWOpenChannelByPath
3	开始视频通道采集	MWStartVideoCapture
4	注册通知事件	MWRegisterNotify
5	等待事件通知	WaitForSignalObject / WaitForMultiObject (Win32 Functions)
6	采集视频图像数据	MWCaptureVideoFrameToVirtualAddress MWCaptureVideoFrameToPhysicalAddress
7	获取采集状态	MWGetVideoCaptureStatus
8	注销通知事件	MWUnRegisterNotify
9	停止视频通道采集	MWStopVideoCapture
10	关闭采集通道	MWCloseChannel

3.2.4 示例列表

CaptureByInput	示例使用视频采集函数，按照输入信号的帧率采集视频数据，并生成截图。
CaptureByTimer	示例使用视频采集函数，按照固定帧率采集视频数据，并生成截图。
HDMIInfoFrame	示例使用信号状态函数获取 HDMI 的 InfoFrame 信息。
InputSignal	示例使用信号状态函数获取输入信号状态。
InputSource	示例使用信号源函数获取信号源状态。
ReadWriteEDID	示例使用 EDID 函数进行 EDID 读写操作。
AudioCapture	示例使用音频采集函数，采集双声道音频，并录制生成 4 秒钟的 wav 文件。
AVCapture	示例展示了采集视频信号，并显示输出到窗口中。同时展示了如何对输入信号进行裁减。
LowLatency	示例采集视频信号进行低延迟处理，并显示到输出窗口中。
MultiAudioCapture	示例采集多路音频信号，并且合成 wav 文件。
MultiStreaming	示例展示多路视频流的采集，可以设置不同分辨率并显示到窗口上。
NDISender	示例采集视音频信号显示到窗口中，同时可以通过网络发送到 NDI 接收端。
OSDPreview	示例将 OSD 图像合成到采集的视频帧中，显示到窗口上。
XICaptureQuad	示例同步采集多路视频帧进行显示。

4 关于

访问以下网站以获取最新版本：

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

如有任何问题，请邮件联系 support@magewell.net

* 本产品兼容 DirectShow SDK (<http://msdn.microsoft.com>)。

* 本产品兼容 NDI (<http://NDI.NewTek.com>)。