

Beyond 5000SDI 系列

By5000io 标清视音频通道卡 SDK 用户手册 V1.0

武汉思非电子技术有限公司

Seekfit Electronics Tech. Co., Ltd.

文件控制

变更记录

日期	作者	版本	更改说明
2010-10-13	Ljn	V1.0	

前 言

感谢您选用思非电子 Beyond 5000sdi 系列视频板卡产品。

思非电子揉合强大的研发能力和丰富的专业制造经验，充分发挥各项视频图像处理技术，全力为您展示强劲丰富的图像性能，务求为您带来清晰、专业的视频效果。无论电视台播出、专业设计、视频会议、还是多媒体应用，思非电子 Beyond 5000sdi 系列视频板卡都有出色的表现。

如果您在使用我们产品的过程中有更多需要或遇到本手册未提及问题，请浏览我们的网站：www.seekfit.com，查阅相关信息，或加入我们的疑难论坛。

本说明书所述产品内容如有更改恕不另行通知。

版权声明




本手册由武汉思非电子技术有限公司版权所有，保留一切权利。
非经本公司书面许可，任何单位和个人不得擅自摘录、复制本手册的部分或者全部内容。
本手册版本：SF-SG-BY5000I0-1.0

安全警示

在安装视频板卡、打开您的计算机之前，请您务必关闭计算机系统的电源，以免在安装的过程中产生静电等意外损害而造成您的计算机主机难以挽回的伤害。另外如果您没有按照说明书的指示来安装本产品而造成计算机系统的损坏，本公司将不予负责。

静电会严重损害电子元件。在触摸任何电子部件之前请先放掉身上的静电。您可以触摸计算机的金属外壳来放掉静电。在拿取板卡时，请尽量拿板卡的边缘，不要触摸电路部分。

图标说明

图标	说明	
	警告	危险标识，表示您处在一个可能引起身体损害的场合中。在您任何操作之前，您必须意识到这种由电气引起的危险并且尽量按照标准的操作方法去避免这种危害发生。
	注意	提示用户在操作过程中需要注意的地方。
	提示	表示给用户提示的附加说明信息。

目 录

1	简介.....	8
1.1	板卡外观.....	8
1.2	原理介绍.....	8
1.2.1	原理框图.....	8
1.2.2	同步模式.....	9
1.2.3	数据处理.....	9
2	SDK函数说明.....	13
2.1	系统函数.....	13
2.1.1	获取板卡总数量.....	13
	Seekfit_By5000io_GetTotalCardNumber.....	13
2.1.2	打开板卡.....	14
	Seekfit_By5000io_OpenCard.....	14
2.1.3	关闭板卡.....	14
	Seekfit_By5000io_CloseCard.....	14
2.1.4	获取板卡类型号.....	14
	Seekfit_By5000io_GetCardTypeNumber.....	14
2.1.5	恢复板卡缺省设置.....	15
	Seekfit_By5000io_RestoreDefault.....	15
2.1.6	初始化场事件.....	15
	Seekfit_By5000io_InitialFieldEvent.....	15
2.1.7	关闭场事件.....	16
	Seekfit_By5000io_CloseFieldEvent.....	16
2.1.8	获取当前板卡时间戳.....	16
	Seekfit_By5000io_GetCardTimeStamp.....	16
2.1.9	获取视频制式.....	16
	Seekfit_By5000io_GetVideoMode.....	17
2.1.10	设置视频制式.....	17
	Seekfit_By5000io_SetVideoMode.....	17
2.1.11	设置SDI继电器状态.....	17
	Seekfit_By5000io_SetVideoRelayMode.....	17
2.1.12	获取SDI继电器状态.....	18
	Seekfit_By5000io_GetVideoRelayMode.....	18
2.1.13	设置音频继电器状态.....	18
	Seekfit_By5000io_SetAudioRelayMode.....	18
2.1.14	获取音频继电器状态.....	19
	Seekfit_By5000io_GetAudioRelayMode.....	19
2.1.15	获取视频输入状态.....	19
	Seekfit_By5000io_GetVideoInputStatus.....	19

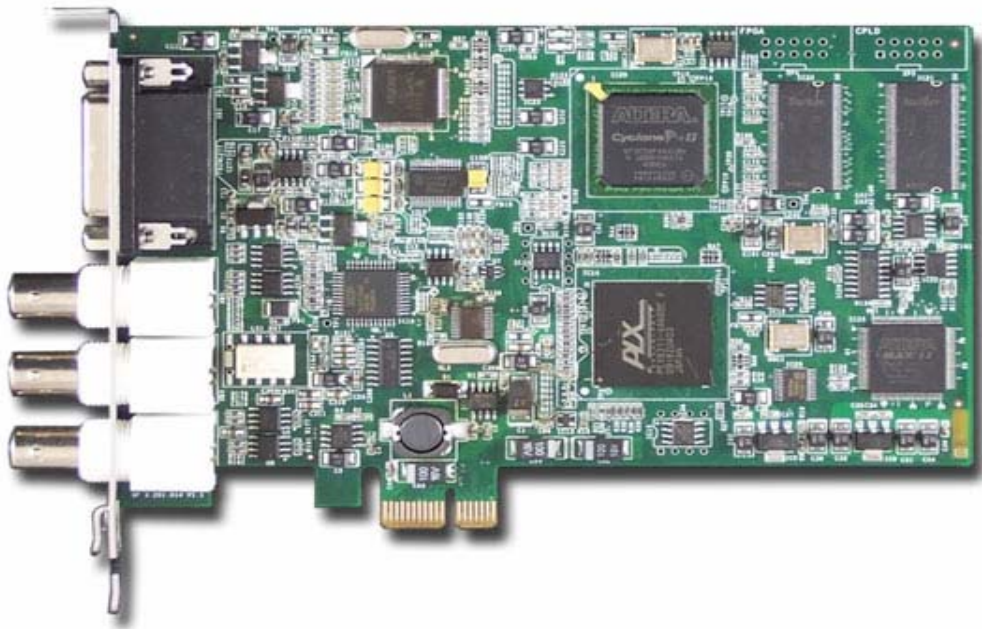
2.1.16	设置板卡同步模式.....	19
	Seekfit_By5000io_SetSyncSource.....	19
2.1.17	获取板卡同步模式.....	20
	Seekfit_By5000io_GetSyncSource.....	20
2.1.18	设置板卡视频源.....	20
	Seekfit_By5000io_SetVideoSource.....	20
2.1.19	获取板卡视频源.....	21
	Seekfit_By5000io_GetVideoSource.....	21
2.1.20	设置板卡音频源.....	21
	Seekfit_By5000io_SetAudioSource.....	21
2.1.21	获取板卡音频源.....	22
	Seekfit_By5000io_GetAudioSource.....	22
2.1.22	写入用户数据.....	22
	Seekfit_By5000io_SetUserData.....	22
2.1.23	校验用户数据.....	22
	Seekfit_By5000io_VerifyUserData.....	22
2.1.24	获取板卡序列号.....	23
	Seekfit_By5000io_GetSerialNumber.....	23
2.2	采集函数.....	23
2.2.1	使能视频采集.....	23
	Seekfit_By5000io_VideoCaptureStart.....	23
2.2.2	取消模拟视频采集.....	24
	Seekfit_By5000io_VideoCaptureStop.....	24
2.2.3	设置采集视频对比度.....	24
	Seekfit_By5000io_SetVideoInputContrast.....	24
2.2.4	获取采集视频对比度.....	24
	Seekfit_By5000io_GetVideoInputContrast.....	24
2.2.5	设置采集视频亮度.....	25
	Seekfit_By5000io_SetVideoInputBrightness.....	25
2.2.6	获取采集视频亮度.....	25
	Seekfit_By5000io_GetVideoInputBrightness.....	25
2.2.7	设置采集视频色度.....	25
	Seekfit_By5000io_SetVideoInputHue.....	25
2.2.8	获取采集视频色度.....	26
	Seekfit_By5000io_GetVideoInputHue.....	26
2.2.9	设置采集视频饱和度.....	26
	Seekfit_By5000io_SetVideoInputSaturation.....	26
2.2.10	获取采集视频饱和度.....	26
	Seekfit_By5000io_GetVideoInputSaturation.....	26
2.2.11	设置采集视频格式.....	27
	Seekfit_By5000io_SetVideoCaptureFrameType.....	27
2.2.12	使能模拟音频采集.....	27
	Seekfit_By5000io_AudioCaptureStart.....	27

2.2.13	取消模拟音频采集.....	27
	Seekfit_By5000io_AudioCaptureStop.....	27
2.2.14	获取模拟音频采集状态.....	28
	Seekfit_By5000io_GetAudioCaptureBufferStatus.....	28
2.2.15	读取模拟音频采集数据.....	28
	Seekfit_By5000io_ReadOneAudioFrame.....	28
2.2.16	获取模拟视频采集状态.....	29
	Seekfit_By5000io_GetVideoCaptureBufferStatus.....	29
2.2.17	读取模拟视频采集数据.....	29
	Seekfit_By5000io_ReadOneVideoFrame.....	29
2.3	播出函数.....	30
2.3.1	设置播出视频格式.....	30
	Seekfit_By5000io_SetVideoPlaybackFrameType.....	30
2.3.2	获取模拟音频播出状态.....	31
	Seekfit_By5000io_GetAudioPlaybackBufferStatus.....	31
2.3.3	写入模拟音频播放数据.....	31
	Seekfit_By5000io_WriteOneAudioFrame.....	31
2.3.4	获取模拟视频播出状态.....	32
	Seekfit_By5000io_GetVideoPlaybackBufferStatus.....	32
2.3.5	写入模拟视频播出数据.....	33
	Seekfit_By5000io_WriteOneVideoFrame.....	33
2.3.6	清空视频播出Buffer.....	33
	Seekfit_By5000io_FlushVideoPlaybackBuffer.....	33
2.3.7	清空音频播出Buffer.....	34
	Seekfit_By5000io_FlushAudioPlaybackBuffer.....	34
2.4	字幕函数.....	34
2.4.1	板卡是否支持字幕功能.....	34
	Seekfit_By5kio_IfSupportVideoDualChannelOutput.....	34
2.4.2	使能双通道输出模式.....	34
	Seekfit_By5kio_EnableVideoDualChannelOutput.....	34
2.4.3	取消双通道输出模式.....	35
	Seekfit_By5kio_DisableVideoDualChannelOutput.....	35
2.4.4	设置键延时.....	35
	Seekfit_By5kio_SetKeyDelay.....	35
2.4.5	获取键延时.....	35
	Seekfit_By5kio_GetKeyDelay.....	35
2.5	版本信息函数.....	36
2.5.1	获取板卡版本.....	36
	Seekfit_By5000io_GetBoardVersion.....	36
2.5.2	获取FPGA版本.....	36
	Seekfit_By5000io_GetFPGAVersion.....	36
2.5.3	获取SDK版本.....	36
	Seekfit_By5000io_GetSdkVersion.....	37

2.5.4	获取驱动版本.....	37
	Seekfit_By5000io_GetDriverVersion.....	37
2.6	数据类型定义.....	37
2.6.1	MEDIA_TIME_CODE定义.....	37
3	常见问题.....	38
4	技术支持.....	38

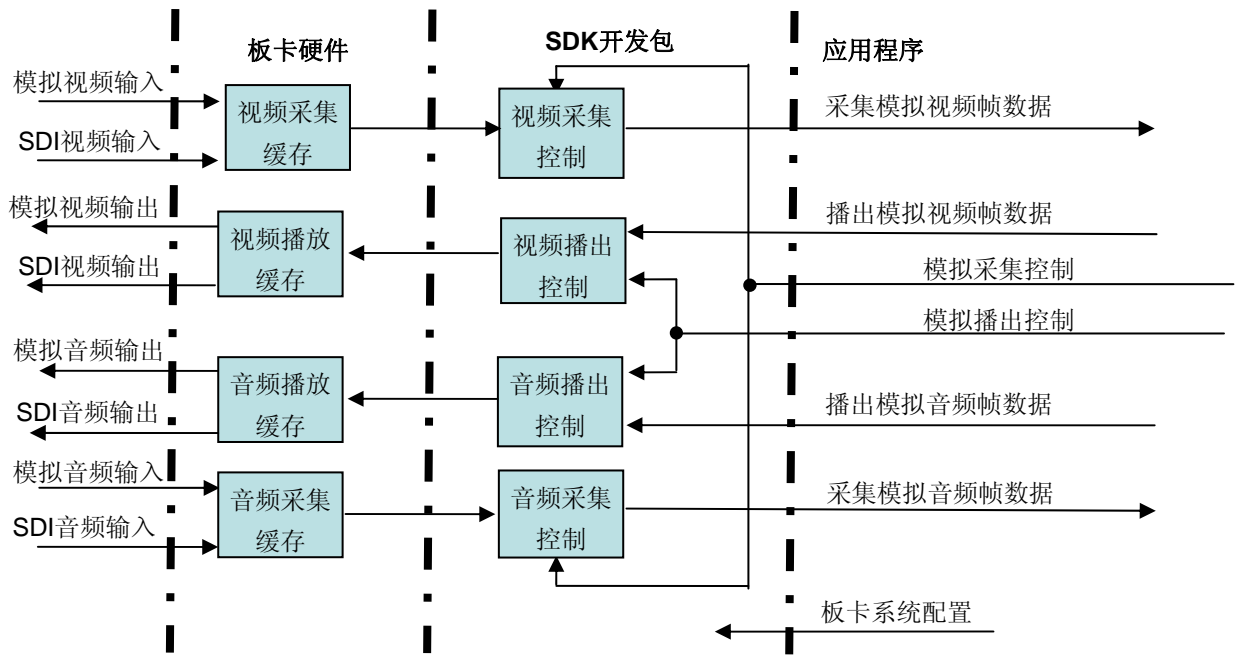
1 简介

1.1 板卡外观



1.2 原理介绍

1.2.1 原理框图



By5000io 视音频通道卡为 PCI-E 总线插卡，数据传输速率可以达到 2GBits/s，是 PCI 速率的 2 倍，极大地提高了板卡的视频数据处理能力。板卡提供 SDI 输入输出接口，可以直接接收和发送 SDI 信号。同时本卡也提供模拟音视频输入输出接口，方便用户进行监视及模拟数字同时播出，输入输出接口详见思非电子公司网站。

用户可以通过 By5000io 视音频通道卡读取采集到的音视频数据，也可以将处理好的音视频数据写入 By5000io 视音频通道卡，板卡将音视频数据输出。从上面原理框图可以看出，本卡只提供一路视频 I/O 通道和一路音频 I/O 通道。视频 I/O 通道的输入源可以选择模拟接口数据源或者 SDI 接口数据源，视频 I/O 通道的输出同时作用于模拟接口输出和 SDI 接口输出，也就是说模拟接口输出和 SDI 接口输出的视频内容是相同的。同样音频 I/O 通道的输入源可以选择模拟接口数据源或者 SDI 接口数据源，音频 I/O 通道的输出同时作用于模拟接口输出和 SDI 接口输出，也就是说模拟接口输出和 SDI 接口输出的音频内容是相同的。SDI 的音频信号是嵌入到视频信号中的，即 SDI 音频和视频在同一根电缆上传送。

1.2.2 同步模式

By5000io 卡有 5 种同步模式，内同步、CVBS 同步、S-VIDEO 同步、YUV 同步、SDI 同步。

内同步模式下，By5000io 卡工作的时基由板卡上的时钟产生，板卡对输入的视频进行时基校正后采集到缓存当中。

其余的四种同步模式属于视频同步，只是所选的输入视频信号不同而已。在视频同步模式下，By5000io 卡工作的时基由外部输入的视频信号产生，板卡输出的视频信号和输入视频信号同步。

1.2.3 数据处理

采集的视、音频数据存放到视频和音频采集缓存中，应用软件通过板卡的 SDK 读取视频

和音频采集缓存中的数据，其中音视频采集控制模块做必要的格式转换及同步处理。

应用软件通过板卡的 SDK 将播放的音视频数据写到视频和音频播放缓存中，板卡直接播出缓存中的音视频数据。其中音视频播放控制模块做必要的格式转换及同步处理。

By5000io 视频通道的视频存储格式为 YUY2，也就是每 4 个字节存放 2 个像素的数据，具体为

字节地址	0	1	2	3
数 据	Y0	Cb0	Y1	Cr0

每帧视频缓存的大小为 (720×576×2) 字节。By5000io 卡的视频采集缓存和视频播放缓存容量都是 8 帧。

By5000io 卡的音频存储格式为双声道 16 比特，也就是每 4 个字节存放 1 个音频样本的数据，具体为

双字节地址	0	1
数 据	左声道 (16 比特)	右声道 (16 比特)

音频的采样速率为 48KHz。为了和视频数据同步，每帧音频缓存的大小为 (48000×0.04=1920) * 4 字节，即 7680 字节。By5000io 卡的音频采集缓存和音频回放缓存容量都是 8 帧。

By5000io 视音频通道卡的视频、音频的采集、回放缓存均为先进先出 (FIFO) 模式。

采集的视频、音频帧都有一个标记采集时刻的时间戳，时间戳以 100ns 为单位。正常情况下，采集到的视频、音频帧的时间戳是严格递增的，通过 SDK 接口函数读取音一帧视频、音频数据，同时可以得到这一帧的时间戳。

播放的视频、音频帧都需要提供一个决定该帧实际播放时刻的时间戳，播放时间戳也是以 100ns 为单位。视频、音频帧保存到板卡的回放缓存中，板卡比较当前的板卡时间和回放缓存最前面一帧的时间戳，如果当前板卡时间小于等于时间戳，音频回放暂停等待，视频重复回放上一帧已经回放过的视频帧；如果当前的板卡时间大于时间戳，继续比较缓存中的下一帧的时间戳，如果当前的板卡时间仍然大于时间戳，就丢弃最前面的一帧，重复这个比较过程，直到当前的板卡时间大于最前面的一帧的时间戳，而当前的板卡时间小于等于下一帧的时间戳，或者回放缓存中只剩下唯一的一帧，才回放最前面的这一帧。也就是说，总是回放缓存中最接近并且小于当前板卡时间的那一帧。

时间戳的概念为软件精确控制视频、音频的回放时刻提供了保证。下面为播放音视频的示例伪代码及设计思路，也可参考 Demo 程序。

/*

播放音视频，时间戳设计思路：

软件开始运行时通过 Seekfit_By5000io_GetCardTimeStamp 获得板卡当前时间戳，称为基准时间戳 `rtCurrentBaseTimeStamp`。

数据处理线程：

软件每处理一帧，在基准时间戳基础上加上一帧的时间得到本帧的时间戳，称为当前帧时间戳 `rtCurrentFrameTimeStamp`。

比如第一帧时间戳 = 基准时间戳 + 1 帧时间；第二帧时间戳 = 基准时间戳 + 2 帧时间，依次

类推。处理好的每一帧放入一个队列，称之为帧队列，等待写卡线程来取走。

写卡线程，下面的代码是写卡线程的参考代码。

获取当前板卡时间戳，和软件维护的软件时间戳比较，看是否有偏差，如果偏差要及时纠正软件时间戳。

从帧队列当中读取处理好的帧，比较当前帧的时间戳和软件时间戳，看是否需要等待，或者丢帧。如果当前帧的时间戳合法，并且板卡缓存有空间可写，就把 软件时间戳+8帧时间 作为写卡时间戳，写入板卡。

*/

```
Void VideoWriteCardThread()
```

```
{  
    DWORD dwWaitResult;  
    UINT uiPlaybackBufferStatus;  
    REFERENCE_TIME rtCurrentCardTimeStamp = 0;//软件时间戳  
    //线程启动时，读取的板卡基准时间戳，运行过程中有可能要及时调整。  
    REFERENCE_TIME rtCurrentBaseTimeStamp = 0;  
    REFERENCE_TIME rtCurrentFrameTimeStamp = 0;//当前帧时间戳  
    REFERENCE_TIME rtNextFrameTimeStamp = 0;//下一帧时间戳  
    UINT uiCurrentFrameNumber = 0;//计算软件时间戳的计数器  
    while(m_hFieldEvent1)  
    {  
        //如果程序退出，退出线程。  
        if (WaitForSingleObject(m_hExitEvent, 0) != WAIT_TIMEOUT)  
            break;  
        //等待板卡场事件，20ms触发一次，此等待过程不是必须的，  
        //只是为了提高程序的运行效率，降低cpu利用率。  
        dwWaitResult = WaitForSingleObject(m_hFieldEvent1, 1000);  
        switch(dwWaitResult)  
        {  
            case WAIT_OBJECT_0://等到场事件  
                //获取板卡当前时间。  
                Seekfit_By5000io_GetCardTimeStamp(rtCurrentCardTimeStamp, uiCardNumber);  
                rtTemp = rtCurrentBaseTimeStamp +  
                TranslateVideoFrameNumberToTime(uiCurrentFrameNumber);  
                //防止数据处理时间过长，处理一帧超过40ms;软件时间戳和板卡当前时间戳同  
                //步，防止偏差太大。  
                if (rtTemp + TranslateVideoFrameNumberToTime(8) < rtCurrentCardTimeStamp)  
                {  
                    //数据处理时间过长，重新纠正软件时间戳  
                    uiCurrentFrameNumber = 0;  
                    rtCurrentBaseTimeStamp = rtCurrentCardTimeStamp;  
                }  
            }  
        }  
    }  
}
```

```
else//由软件自己维护的软件时间戳。
{
    uiCurrentFrameNumber++;
    rtCurrentCardTimeStamp = rtCurrentBaseTimeStamp +
TranslateVideoFrameNumberToTime(uiCurrentFrameNumber);
}
//读取帧队列，获得当前帧时间戳，和下一帧时间戳，
readFrameFromQueue(rtCurrentFrameTimeStamp, rtNextFrameTimeStamp,
VideoBuffer);//伪代码。
if (rtCurrentFrameTimeStamp > rtCurrentCardTimeStamp)
{
    //帧超前，等待,下次仍然处理这一帧，帧队列不清除当前帧。
    break;
}
if (rtNextFrameTimeStamp < rtCurrentCardTimeStamp)
{
    //帧过期，丢弃当前帧
    RmoveFrameFromQueue(); //伪代码。
    break;
}
//帧时间戳合法，写卡
RmoveFrameFromQueue(); //伪代码。
Seekfit_By5000io_GetVideoPlaybackBufferStatus(uiPlaybackBufferStatus,
m_uiCardNumber);
//判断硬件是否有缓存可写。
while(uiPlaybackBufferStatus == VIDEO_BUFFER_STATUS_FULL)
{
    Sleep(1);
    Seekfit_By5000io_GetVideoPlaybackBufferStatus(uiPlaybackBufferStatus,
m_uiCardNumber);
}
//将视频帧数据写卡，加上8帧是为了将硬件填满后再开始播放。
Seekfit_By5000io_WriteOneVideoFrame(rtCurrentCardTimeStamp +
8*40*1000*10, VideoBuffer, m_uiCardNumber);
break;
case WAIT_TIMEOUT:
    TRACE(_T("***** TIMEOUT !!!!!!!!!!!!!\n"));
    break;
default:
    TRACE(_T("***** Anything else ?????????????\n"));
}
Sleep(1);
}
```

```
}  
REFERENCE_TIME TranslateVideoFrameNumberToTime(UINT uiFrameNumber)  
{  
    if (m_uiVideoMode == VIDEO_MODE_PAL)  
    {  
        return (REFERENCE_TIME)uiFrameNumber * 40 * 1000 * 10;  
    }  
    else if (m_uiVideoMode == VIDEO_MODE_NTSC)  
    {  
        return (REFERENCE_TIME)uiFrameNumber * 9009000 / 27;  
    }  
    else  
    {  
        return (REFERENCE_TIME)uiFrameNumber * 40 * 1000 * 10;  
    }  
}
```

2 SDK 函数说明

SDK 当中所有函数的返回值均为：

1 : 表示成功。

0 : 表示失败。

2.1 系统函数

2.1.1 获取板卡总数量

Seekfit_By5000io_GetTotalCardNumber

原 型

```
BOOL Seekfit_By5000io_GetTotalCardNumber(  
    UINT &    uiTotalCardNumber  
);
```

功 能

获取当前检测到的 By5000io 板卡的数量。

参 数

1 uiTotalCardNumber

如果 uiTotalCardNumber 返回 0，表示没有检测到板卡，此时对板卡进行其他操作可能导致系统死机。

如果 uiTotalCardNumber 返回 1，对应的板卡编号为 0，如果 uiTotalCardNumber 返回 2，对应的板卡编号为 0，1，依次类推。

从本函数以下的函数，参数 uiCardNumber 均表示该函数要操作的卡号，0 表示第一块卡，1 表示第二块卡，依次类推，不再单独对此参数做说明。

2.1.2 打开板卡

Seekfit_By5000io_OpenCard

原 型

```
BOOL Seekfit_By5000io_OpenCard(  
    UINT          uiCardNumber = 0  
);
```

功 能

初始化板卡各硬件模块和 SDK 软件模块，分配板卡工作所需的软、硬件资源，只有当此函数执行完后，后面的函数才能调用。

参 数

2.1.3 关闭板卡

Seekfit_By5000io_CloseCard

原 型

```
BOOL Seekfit_By5000io_CloseCard(  
    UINT          uiCardNumber = 0  
);
```

功 能

释放板卡所分配的软、硬件资源，关闭各硬件模块和 SDK 软件模块，调用此函数后，不能再调用下面的其他函数。

参 数

2.1.4 获取板卡类型号

Seekfit_By5000io_GetCardTypeNumber

原 型

```
BOOL Seekfit_By5000io_GetCardTypeNumber(  
    UINT &    uiCardTypeNo,  
    UINT      uiCardNumber = 0  
);
```

功 能

获取 By5000io 板卡类型号。

参 数

1 uiCardTypeNo

返回的板卡类型号，By5000io 对应的类型号为 0x5000。

2.1.5 恢复板卡缺省设置

```
Seekfit_By5000io_RestoreDefault
```

原 型

```
BOOL Seekfit_By5000io_RestoreDefault(  
    UINT      uiCardNumber = 0  
);
```

功 能

如果板卡设置被改写导致无法正常工作，可以恢复板卡缺省设置，使板卡恢复正常工作。

参 数

2.1.6 初始化场事件

```
Seekfit_By5000io_InitialFieldEvent
```

原 型

```
BOOL Seekfit_By5000io_InitialFieldEvent(  
  
LPCTSTR szEvent,  
  
HANDLE & hEvent,  
  
UINT uiCardNumber = 0);
```

功 能

初始化一个场事件。场事件由板卡硬件中断来触发，每 20ms 触发一次，用户可以用场事件来触发读板卡或者写板卡操作。但此事件不是必须要初始化的，用事件来触发硬件相关的操作只是为了提高程序执行效率，降低 cpu 利用率。可以多次调用来生成多个事件，最多允许生成 256 个场事件。

由于事件以名称区分，并且在操作系统范围内有效，所以每个事件的名称必须唯一。

参 数

1 szEvent

事件名称。

2 hEvent

SDK 返回的事件句柄。

2.1.7 关闭场事件

Seekfit_By5000io_CloseFieldEvent

原 型

```
BOOL Seekfit_By5000io_CloseFieldEvent(  
  
HANDLE hEvent,  
  
UINT uiCardNumber = 0);
```

功 能

关闭一个场事件。

参 数

1 hEvent

要关闭的事件句柄，如果此参数为 NULL 值，则关闭所有场事件。

2.1.8 获取当前板卡时间戳

Seekfit_By5000io_GetCardTimeStamp

原 型

```
BOOL Seekfit_By5000io_GetCardTimeStamp(  
  
LONGLONG & llTimeStamp,  
  
UINT uiCardNumber = 0);
```

功 能

获取板卡当前时间戳，以 100ns 为单位。

参 数

1 llTimeStamp

返回的当前板卡时间戳。

2.1.9 获取视频制式

Seekfit_By5000io_GetVideoMode

原 型

```
BOOL Seekfit_By5000io_GetVideoMode(UINT & uiMode, UINT uiCardNumber = 0);
```

功 能

获取板卡当前视频制式。

参 数

l uiMode

返回板卡当前视频制式。

VIDEO_MODE_NTSC NTSC 制式

VIDEO_MODE_PAL PAL 制式

2.1.10 设置视频制式

Seekfit_By5000io_SetVideoMode

原 型

```
BOOL Seekfit_By5000io_SetVideoMode(UINT uiMode, UINT uiCardNumber = 0);
```

功 能

设置板卡视频制式。

参 数

l uiMode

VIDEO_MODE_NTSC NTSC 制式

VIDEO_MODE_PAL PAL 制式

2.1.11 设置 SDI 继电器状态

Seekfit_By5000io_SetVideoRelayMode

原 型

```
BOOL Seekfit_By5000io_SetVideoRelayMode(UINT uiMode, UINT uiCardNumber = 0);
```

功 能

设置 SDI 继电器为直通状态或者正常状态，SDI 信号包括音频和视频，所以此处称视频继电器不太科学。此继电器为 SDI 继电器，板卡只有这一个继电器。

当调用函数 `Seekfit_By5000io_CloseCard` 关闭板卡或者电脑断电，继电器会自动置为直通状态。

当调用 `Seekfit_By5000io_OpenCard` 打开板卡，继电器会自动置为正常状态。

参 数

1 uiMode

RELAY_MODE_LOOPBACK

直通状态，SDI 信号经过板卡不做任何处理直接输出。

RELAY_MODE_NORMAL

正常状态，视频进入板卡，做正常处理后再输出。

2.1.12 获取 SDI 继电器状态

Seekfit_By5000io_GetVideoRelayMode

原 型

```
BOOL Seekfit_By5000io_GetVideoRelayMode(UINT &uiMode, UINT uiCardNumber = 0);
```

功 能

获取视频继电器状态。

参 数

1 uiMode

返回当前继电器状态。

RELAY_MODE_LOOPBACK

直通模式，视频经过板卡不做任何处理直接输出。

RELAY_MODE_NORMAL

正常模式，视频进入板卡，做正常处理后再输出。

2.1.13 设置音频继电器状态

Seekfit_By5000io_SetAudioRelayMode

原 型

```
BOOL Seekfit_By5000io_SetAudioRelayMode(UINT uiMode, UINT uiCardNumber = 0);
```

功 能

在 By5000io 板卡中暂时没有此继电器，此函数暂时无效。

参 数

1 uiMode

RELAY_MODE_LOOPBACK

直通状态，音频经过板卡不做任何处理直接输出。

RELAY_MODE_NORMAL

正常状态，音频进入板卡，做正常处理后再输出。

2.1.14 获取音频继电器状态

Seekfit_By5000io_GetAudioRelayMode

原 型

```
BOOL Seekfit_By5000io_GetAudioRelayMode(UINT &uiMode, UINT uiCardNumber = 0);
```

功 能

在 By5000io 板卡中暂时没有此继电器，此函数暂时无效。

参 数

1 uiMode

返回当前继电器状态。

RELAY_MODE_LOOPBACK

直通状态，音频经过板卡不做任何处理直接输出，可以做到掉电直通。

RELAY_MODE_NORMAL

正常状态，音频进入板卡，做正常处理后再输出。

2.1.15 获取视频输入状态

Seekfit_By5000io_GetVideoInputStatus

原 型

```
BOOL Seekfit_By5000io_GetVideoInputStatus(UINT &uiStatus, UINT uiCardNumber = 0);
```

功 能

获取视频输入状态，此函数目前只能获取模拟视频输入的状态，对 SDI 视频输入状态无法获取。

参 数

1 uiStatus

返回当前视频输入状态。

VIDEO_INPUT_UNLOCKED

锁定状态，有视频信号输入。

VIDEO_INPUT_UNLOCKED

正常模式，无视频信号输入。

2.1.16 设置板卡同步模式

Seekfit_By5000io_SetSyncSource

原 型

```
BOOL Seekfit_By5000io_SetSyncSource(UINT uiSource, UINT uiCardNumber = 0);
```

功 能

设置板卡同步模式，参考 1.2.2。

参 数

1 uiSource

要设置的同步模式。

BY5000IO_SYNC_SOURCE_INTERNAL	内同步
BY5000IO_VIDEO_SOURCE_CVBS	CVBS 同步
BY5000IO_VIDEO_SOURCE_SVIDEO	SVIDEO 同步
BY5000IO_VIDEO_SOURCE_YUV	YUV 同步
By5000io_VIDEO_SOURCE_SDI	SDI 同步

2.1.17 获取板卡同步模式

```
Seekfit_By5000io_GetSyncSource
```

原 型

```
BOOL Seekfit_By5000io_GetSyncSource(UINT &uiSource, UINT uiCardNumber = 0);
```

功 能

获取板卡同步模式，参考 1.2.2。

参 数

1 uiMode

返回的同步模式。

BY5000IO_SYNC_SOURCE_INTERNAL	内同步
BY5000IO_VIDEO_SOURCE_CVBS	CVBS 同步
BY5000IO_VIDEO_SOURCE_SVIDEO	SVIDEO 同步
BY5000IO_VIDEO_SOURCE_YUV	YUV 同步
BY5000IO_VIDEO_SOURCE_SDI	SDI 同步

2.1.18 设置板卡视频源

```
Seekfit_By5000io_SetVideoSource
```

原 型

```
BOOL Seekfit_By5000io_SetVideoSource(UINT uiSource, UINT uiCardNumber = 0);
```

功 能

设置板卡视频源，本板卡提供多种视频输入源，可以从其中选择一种作为视频输入源。

参 数

l uiSource

要设置的视频源。

BY5000IO_VIDEO_SOURCE_CVBS	CVBS 视频源
BY5000IO_VIDEO_SOURCE_SVIDEO	SVIDEO 视频源
BY5000IO_VIDEO_SOURCE_YUV	YUV 视频源
BY5000IO_VIDEO_SOURCE_SDI	SDI 视频源

2.1.19 获取板卡视频源

Seekfit_By5000io_GetVideoSource

原 型

```
BOOL Seekfit_By5000io_GetVideoSource(UINT & uiSource, UINT uiCardNumber = 0);
```

功 能

获取板卡当前选择的视频源。

参 数

l uiSource

返回的板卡视频源。

BY5000IO_VIDEO_SOURCE_CVBS	CVBS 视频源
BY5000IO_VIDEO_SOURCE_SVIDEO	SVIDEO 视频源
BY5000IO_VIDEO_SOURCE_YUV	YUV 视频源
BY5000IO_VIDEO_SOURCE_SDI	SDI 视频源

2.1.20 设置板卡音频源

Seekfit_By5000io_SetAudioSource

原 型

```
BOOL Seekfit_By5000io_SetAudioSource(UINT uiSource, UINT uiCardNumber = 0);
```

功 能

本板卡音频源有两种，一种是模拟音频(RCA)，另外一种为 SDI 数字音频。

参 数

l uiSource

要设置的音频源。

BY5000IO_AUDIO_SOURCE_RCA	莲花模拟音频
BY5000IO_AUDIO_SOURCE_SDI	SDI 数字音频

2.1.21 获取板卡音频源

Seekfit_By5000io_GetAudioSource

原 型

```
BOOL Seekfit_By5000io_GetAudioSource(UINT& uiSource, UINT uiCardNumber = 0);
```

功 能

获取板卡当前音频源设置。

参 数

1 uiSource

返回板卡当前的音频源。

BY5000IO_AUDIO_SOURCE_RCA 莲花模拟音频

BY5000IO_AUDIO_SOURCE_SDI SDI 数字音频

2.1.22 写入用户数据

Seekfit_By5000io_SetUserData

原 型

```
BOOL Seekfit_By5000io_SetUserData(UINT uiAddr, UCHAR ucData, UINT uiCardNumber = 0);
```

功 能

将一些特定数据写入板卡中，既可以作为保存数据用，也可以用来将软件和板卡硬件一对一绑定，起到保护软件的作用，和 Seekfit_By5000io_VerifyUserData 配合使用。Seekfit_By5000io_SetUserData 将数据写入板卡，每次写入一个字节，可以多次写入；使用 Seekfit_By5000io_VerifyUserData 来校验，判断板卡是否和软件对应。

参 数

1 uiAddr

要写入的地址，取值范围[0, 0x2f]，每次写入一个字节，可以多次写入。

2 ucData

写入的一个字节的数据。

2.1.23 校验用户数据

Seekfit_By5000io_VerifyUserData

原 型

```
BOOL Seekfit_By5000io_VerifyUserData(UINT uiAddr, UINT uiLength, UCHAR *  
    pucData, UINT uiCardNumber = 0);
```

功 能

校验写入板卡的用户数据，可以一次校验多个字节。参考 2.1.22。

参 数

1 uiAddr

要开始校验的地址，取值范围[0, 0x2f]，并且满足条件： $uiAddr+uiLength - 1 < 0x2f$ 。

2 ucLength

要校验的字节个数。

3 pucData

保存待校验数据的地址。

2.1.24 获取板卡序列号

Seekfit_By5000io_GetSerialNumber

原 型

```
BOOL Seekfit_By5000io_GetSerialNumber(ULONG & ulSerialNumber, UINT  
    uiCardNumber = 0);
```

功 能

获取板卡序列号，每个板卡对应唯一的一个序列号，该序列号和板卡背面的条形码对应。

参 数

1 ulSerialNumber

返回的板卡序列号，格式如下。

按 16 进制，如 0x09100073，表示 09 年 10 月份第 73 号卡。

2.2 采集函数

2.2.1 使能视频采集

Seekfit_By5000io_VideoCaptureStart

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_VideoCaptureStart(UINT uiCardNumber = 0);
```

功 能

使能板卡模拟视频采集功能，板卡缺省为使能状态。

参 数

2.2.2 取消模拟视频采集

Seekfit_By5000io_VideoCaptureStop

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_VideoCaptureStop(UINT uiCardNumber = 0);
```

功 能

取消板卡模拟视频采集功能。

参 数

2.2.3 设置采集视频对比度

Seekfit_By5000io_SetVideoInputContrast

原 型

```
BOOL Seekfit_By5000io_SetVideoInputContrast(UINT uiCardNumber, INT uiContrast, UINT uiCardNumber = 0);
```

功 能

设置板卡模拟接口采集视频对比度。

参 数

1 uiContrast

要设置的对比度，取值范围[0, 255]。

2.2.4 获取采集视频对比度

Seekfit_By5000io_GetVideoInputContrast

原 型

```
BOOL Seekfit_By5000io_GetVideoInputContrast(UINT uiCardNumber, INT & uiContrast, UINT uiCardNumber = 0);
```

功 能

获取板卡模拟接口采集视频对比度。

参 数

1 uiContrast

返回的对比度，范围[0, 255]。

2.2.5 设置采集视频亮度

Seekfit_By5000io_SetVideoInputBrightness

原 型

```
BOOL Seekfit_By5000io_SetVideoInputBrightness(UINT uiBrightness, UINT  
        uiCardNumber = 0);
```

功 能

设置板卡模拟接口采集视频亮度。

参 数

1 uiBrightness

要设置的亮度，取值范围[0, 255]。

2.2.6 获取采集视频亮度

Seekfit_By5000io_GetVideoInputBrightness

原 型

```
BOOL Seekfit_By5000io_GetVideoInputBrightness(UINT & uiBrightness, UINT  
        uiCardNumber = 0);
```

功 能

获取板卡模拟接口采集视频亮度。

参 数

1 uiBrightness

返回的亮度，范围[0, 255]。

2.2.7 设置采集视频色度

Seekfit_By5000io_SetVideoInputHue

原 型

```
BOOL Seekfit_By5000io_SetVideoInputHue(UINT uiHue, UINT uiCardNumber = 0);
```

功 能

设置板卡模拟接口采集视频色度。

参 数

1 uiHue

要设置的色度，取值范围[0, 255]。

2.2.8 获取采集视频色度

Seekfit_By5000io_GetVideoInputHue

原 型

```
BOOL Seekfit_By5000io_GetVideoInputHue(UINT & uiHue, UINT uiCardNumber = 0);
```

功 能

获取板卡模拟接口采集视频色度。

参 数

1 uiHue

返回的色度，范围[0, 255]。

2.2.9 设置采集视频饱和度

Seekfit_By5000io_SetVideoInputSaturation

原 型

```
BOOL Seekfit_By5000io_SetVideoInputSaturation(UINT uiSaturation, UINT  
uiCardNumber = 0);
```

功 能

设置板卡模拟接口采集视频饱和度。

参 数

1 uiSaturation

要设置的饱和度，取值范围[0, 255]。

2.2.10 获取采集视频饱和度

Seekfit_By5000io_GetVideoInputSaturation

原 型

```
BOOL Seekfit_By5000io_GetVideoInputSaturation(UINT & uiSaturation, UINT  
uiCardNumber = 0);
```

功 能

获取板卡模拟接口采集视频饱和度。

参 数

1 uiSaturation
返回的饱和度，范围[0, 255]。

2.2.11 设置采集视频格式

Seekfit_By5000io_SetVideoCaptureFrameType

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_SetVideoCaptureFrameType(  
UINT uiType,  
UINT uiCardNumber = 0);
```

功 能

设置采集视频的格式。用户通过此函数可以设置采集上来的视频格式。

参 数

1 uiType

要设置的采集视频格式，取值如下。

VIDEO_FRAME_TYPE_YUY2	采集视频为 YUY2 格式
VIDEO_FRAME_TYPE_ARGB32	采集视频为 ARGB32 格式
VIDEO_FRAME_TYPE_ARGB32_UPSIDEDOWN	采集视频为上下颠倒 ARGB32 格式

2.2.12 使能模拟音频采集

Seekfit_By5000io_AudioCaptureStart

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_AudioCaptureStart(UINT uiCardNumber = 0);
```

功 能

使能板卡模拟音频采集功能，板卡缺省为使能状态。

参 数

2.2.13 取消模拟音频采集

Seekfit_By5000io_AudioCaptureStop

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_AudioCaptureStop(UINT uiCardNumber = 0);
```

功 能

取消板卡模拟音频采集功能。

参 数

2.2.14 获取模拟音频采集状态

Seekfit_By5000io_GetAudioCaptureBufferStatus

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_GetAudioCaptureBufferStatus(  
  
UINT & uiStatus,  
  
UINT uiCardNumber = 0);
```

功 能

获取板卡模拟音频采集状态，用户通过此函数可以得知当前硬件 Buffer 是否有等待采集的音频数据。

参 数

1 uiStatus

返回的硬件采集音频 Buffer 状态，取值如下。

AUDIO_BUFFER_STATUS_NORMAL Buffer 正常，没有满，也没有空，用户可以读取采集到的音频数据。

AUDIO_BUFFER_STATUS_EMPTY Buffer 空，此状态下，用户要等待，等到 Buffer 不为空时，再读取采集的音频数据。

AUDIO_BUFFER_STATUS_FULL Buffer 满，此状态下用户可以读取采集到的音频数据。

2.2.15 读取模拟音频采集数据

Seekfit_By5000io_ReadOneAudioFrame

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_ReadOneAudioFrame(  
  
LONGLONG & IIFrameTimeStamp,  
  
PULONG pulBuffer,  
  
UINT uiCardNumber = 0);
```

功 能

读取一帧模拟音频采集数据。为了和视频同步，将音频也以帧为单位来处理。

音频数据按照采样频率 48k、双声道、采样大小 16bit 格式，按照每秒 25 帧的频率计算，每次读取的音频帧数据量大小为 $(48000/25)*2*2 = 1920*4$ 字节。

参 数

1 llFrameTimeStamp

SDK 返回采集该音频帧的板卡时间戳，时间戳以 100ns 为单位，参考 1.2.3。

2 pulBuffer

存放音频数据的内存地址。此内存空间要用户自己申请，自己释放，内存大小要求大于等于 $1920*4$ 字节。

2.2.16 获取模拟视频采集状态

Seekfit_By5000io_GetVideoCaptureBufferStatus

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_GetVideoCaptureBufferStatus(  
UINT & uiStatus,  
UINT uiCardNumber = 0);
```

功 能

获取板卡模拟视频采集状态，用户通过此函数可以得知当前硬件 Buffer 是否有等待采集的视频数据。

参 数

1 uiStatus

返回的硬件采集视频 Buffer 状态，取值如下。

VIDEO_BUFFER_STATUS_NORMAL Buffer 正常，没有满，也没有空，用户可以读取采集到的视频数据。

VIDEO_BUFFER_STATUS_EMPTY Buffer 空，此状态下，用户要等待，等到 Buffer 不为空时，再读取采集的视频数据。

VIDEO_BUFFER_STATUS_FULL Buffer 满，此状态下用户可以读取采集到的视频数据。

2.2.17 读取模拟视频采集数据

Seekfit_By5000io_ReadOneVideoFrame

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_ReadOneVideoFrame(  
LONGLONG & llFrameTimeStamp,
```

PULONG pulBuffer,

UINT uiCardNumber = 0);

功 能

读取一帧模拟视频采集数据。读取的一帧视频大小和函数 Seekfit_By5000io_SetVideoCaptureFrameType 的设置有关系，如果设置为 YUY2 的采集视频格式，则一帧视频大小为 PAL 制式 720*576*2 字节，NTSC 制式大小 720*480*2 字节；如果设置为 ARGB32 或者 ARGB32_UPSIDEDOWN 格式，则一帧视频大小为 PAL 制式 720*576*4 字节，NTSC 制式大小 720*480*4 字节。

参 数

1 llFrameTimeStamp

SDK 返回采集该视频帧的板卡时间戳，时间戳以 100ns 为单位，参考 1.2.3。

2 pulBuffer

存放视频数据的内存地址。此内存空间要用户自己申请，自己释放，申请的内存大小参考本函数的“功能”说明，不管是 PAL 视频源还是 NTSC 制式视频源，都按照 PAL 制式大小来申请。

2.3 播出函数

2.3.1 设置播出视频格式

Seekfit_By5000io_SetVideoPlaybackFrameType

原 型

BOOL CALL_TYPE Seekfit_By5000io_SetVideoPlaybackFrameType(

UINT uiChannel,

UINT uiType,

UINT uiCardNumber = 0);

功 能

设置播出视频的格式。通过此函数可以设置用户写入板卡的视频格式，SDK 根据用户的设置，做相应的视频处理。

参 数

1 uiChannel

By5000io 卡可以作为字幕卡用，具体用法如下：

本卡的 SDI 输出支持双通道输出模式：主输出 (SDI_OUT1) 和键输出 (SDI_OUT2/KEY)，模拟接口只支持单通道输出，模拟接口输出的内容和 SDI_OUT1 相同。

作为字幕卡用，用户将包含键信号信息的 BGRA 格式视频数据写入板卡，SDK 提取视频数据当中的键信号，将视频数据从主输出播出，将键信号从键输出播出。

首先要使能双通道输出模式，然后通过本函数设置播出的视频类型。

设置主输出视频格式：uiChannel 置为 0，uiType 置为 VIDEO_FRAME_TYPE_ARGB32 或者 VIDEO_FRAME_TYPE_ARGB32_UPSIDEDOWN。

设置键输出视频格式：uiChannel 置为 1，uiType 置为 VIDEO_FRAME_TYPE_KEY。

具体使用参考 By5000ioDemo 程序源代码。

2 uiType

要设置的播放视频格式，取值如下。

VIDEO_FRAME_TYPE_YUY2	用户写入板卡视频为 YUY2 格式
VIDEO_FRAME_TYPE_ARGB32	用户写入板卡视频为 ARGB32 格式
VIDEO_FRAME_TYPE_ARGB32_UPSIDEDOWN	用户写入板卡视频为上下颠倒 ARGB32 格式
VIDEO_FRAME_TYPE_KEY	作为键模式输出

2.3.2 获取模拟音频播出状态

Seekfit_By5000io_GetAudioPlaybackBufferStatus

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_GetAudioPlaybackBufferStatus(  
UINT & uiStatus,  
UINT uiCardNumber = 0);
```

功 能

获取板卡硬件模拟音频播出状态。用户通过此函数可以得知当前硬件是否有空闲 Buffer 可以写入。

参 数

1 uiStatus

返回的硬件播出音频 Buffer 状态，取值如下。

AUDIO_BUFFER_STATUS_NORMAL Buffer 正常，没有满，也没有空，用户可以向板卡写入要播放的音频数据。

AUDIO_BUFFER_STATUS_EMPTY Buffer 空，此状态下，用户可以向板卡写入要播放的音频数据。

AUDIO_BUFFER_STATUS_FULL Buffer 满，此状态下用户要等待 Buffer 不为 FULL，然后向板卡写入要播放的音频数据。

2.3.3 写入模拟音频播放数据

Seekfit_By5000io_WriteOneAudioFrame

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_WriteOneAudioFrame(  
  
LONGLONG llFrameTimeStamp,  
  
PULONG pulBuffer,  
  
UINT uiCardNumber = 0);
```

功 能

向板卡写入一帧音频播出数据。为了和视频同步，将音频也以帧为单位来处理。
音频数据按照采样频率 48k、双声道、采样大小 16bit 格式，按照每秒 25 帧的频率计算，每次写入的音频帧数据量大小为 $(48000/25)*2*2 = 1920*4$ 字节。

参 数

1 llFrameTimeStamp

该音频帧播出的板卡时间戳，时间戳以 100ns 为单位，硬件会根据用户写入的时间戳来播出该音频数据，参考 1.2.3。

2 pulBuffer

存放音频数据的内存地址。此内存空间要用户自己申请，自己释放，内存大小要求大于等于 $1920*4$ 字节。

2.3.4 获取模拟视频播出状态

```
Seekfit_By5000io_GetVideoPlaybackBufferStatus
```

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_GetVideoPlaybackBufferStatus(  
  
UINT & uiStatus,  
  
UINT uiCardNumber = 0);
```

功 能

获取板卡硬件模拟视频播出状态。用户通过此函数可以得知当前硬件是否有空闲 Buffer 可以写入。

参 数

1 uiStatus

返回的硬件播出视频 Buffer 状态，取值如下。

VIDEO_BUFFER_STATUS_NORMAL Buffer 正常，没有满，也没有空，用户可以向板卡写入要播放的视频数据。

VIDEO_BUFFER_STATUS_EMPTY Buffer 空，此状态下，用户可以向板卡写入要播放的视频数据。

VIDEO_BUFFER_STATUS_FULL Buffer 满，此状态下用户要等待 Buffer 不为 FULL，然后向板卡写入要播放的视频数据。

2.3.5 写入模拟视频播出数据

Seekfit_By5000io_WriteOneVideoFrame

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_WriteOneVideoFrame(  
LONGLONG llFrameTimeStamp,  
PULONG pulBuffer,  
UINT uiCardNumber = 0);
```

功 能

向板卡写入一帧模拟视频播放数据。一帧视频大小和函数 Seekfit_By5000io_SetVideoPlaybackFrameType 的设置有关系，如果设置为 YUY2 的视频格式，则一帧视频大小为 PAL 制式 720*576*2 字节，NTSC 制式大小 720*480*2 字节；如果设置为 ARGB32 或者 ARGB32_UPSIDEDOWN 格式，则一帧视频大小为 PAL 制式 720*576*4 字节，NTSC 制式大小 720*480*4 字节。

参 数

1 llFrameTimeStamp

该视频帧播出的板卡时间戳，时间戳以 100ns 为单位，硬件会根据用户写入的时间戳来播出该视频数据，参考 1.2.3。

2 pulBuffer

存放视频数据的内存地址。此内存空间要用户自己申请，自己释放，申请的内存大小参考本函数的“功能”说明，不管是 PAL 视频源还是 NTSC 制式视频源，都按照 PAL 制式大小来申请。

2.3.6 清空视频播出 Buffer

Seekfit_By5000io_FlushVideoPlaybackBuffer

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_FlushVideoPlaybackBuffer(  
UINT uiCardNumber = 0);
```

功 能

将播出视频的硬件 Buffer 清空，播出画面会停留在当前帧，直到用户写入新的视频帧，板卡从新写入的视频帧开始播出。

参 数

2.3.7 清空音频播出 Buffer

Seekfit_By5000io_FlushAudioPlaybackBuffer

原 型

```
BOOL CALL_TYPE Seekfit_By5000io_FlushAudioPlaybackBuffer(  
UINT uiCardNumber = 0);
```

功 能

将播出音频的硬件 Buffer 清空，板卡输出静音，直到用户写入新的音频帧，板卡从新写入的音频帧开始播出。

参 数

2.4 字幕函数

2.4.1 板卡是否支持字幕功能

Seekfit_By5kio_IfSupportVideoDualChannelOutput

原 型

```
BOOL CALL_TYPE Seekfit_By5kio_IfSupportVideoDualChannelOutput(UINT uiCardNumber  
= 0);
```

功 能

老版本的 By5000io 卡不支持字幕卡功能，此函数可以判断一下板卡是否可以作为字幕卡用，函数返回 0 表示不支持，返回 1 表示支持。

参 数

2.4.2 使能双通道输出模式

Seekfit_By5kio_EnableVideoDualChannelOutput

原 型

```
BOOL CALL_TYPE Seekfit_By5kio_EnableVideoDualChannelOutput(UINT uiCardNumber =  
0);
```

功 能

使能 By5000io 卡的双通道输出功能，只有使能此功能后，才可以作为字幕卡用，参考 2.3.1。

参 数

2.4.3 取消双通道输出模式

Seekfit_By5kio_DisableVideoDualChannelOutput

原 型

```
BOOL CALL_TYPE Seekfit_By5kio_DisableVideoDualChannelOutput(UINT uiCardNumber = 0);
```

功 能

取消 By5000io 卡的双通道输出功能，取消此功能后，不能作为字幕卡用，参考 2.3.1。

参 数

2.4.4 设置键延时

Seekfit_By5kio_SetKeyDelay

原 型

```
BOOL CALL_TYPE Seekfit_By5kio_SetKeyDelay(  
UINT keyDelay,  
UINT uiCardNumber = 0);
```

功 能

设置键延时，可以微调键信号的偏移，用来和视频对齐。

参 数

1 keyDelay

要设置的键延时值，取值范围[0, 255]，缺省值 128。

2.4.5 获取键延时

Seekfit_By5kio_GetKeyDelay

原 型

```
BOOL CALL_TYPE Seekfit_By5kio_GetKeyDelay(  
UINT &keyDelay,  
UINT uiCardNumber = 0);
```

功 能

获取板卡当前键延时值。

参 数

l keyDelay

SDK 返回的键延时值，取值范围[0, 255]。

2.5 版本信息函数

2.5.1 获取板卡版本

Seekfit_By5000io_GetBoardVersion

原 型

```
BOOL Seekfit_By5000io_GetBoardVersion(UINT * puiBuffer, UINT uiCardNumber = 0);
```

功 能

获取板卡版本号。

参 数

l puiBuffer

返回的版本数据，取 puiBuffer 指针指向的整形数据，按 16 进制表示，只用了低 24 位，分为三部分，每 8 位表示一部分数值，如：

0x00010400，表示 1.4.0。

2.5.2 获取 FPGA 版本

Seekfit_By5000io_GetFPGAVersion

原 型

```
BOOL Seekfit_By5000io_GetFPGAVersion(UINT * puiBuffer, UINT uiCardNumber = 0);
```

功 能

获取 FPGA 版本号。

参 数

l puiBuffer

返回的版本数据，参考 2.5.1。

2.5.3 获取 SDK 版本

Seekfit_By5000io_GetSdkVersion

原 型

```
BOOL Seekfit_By5000io_GetSdkVersion(UINT * puiBuffer);
```

功 能

获取软件 SDK 版本号。

参 数

1 puiBuffer

返回的版本数据，参考 2.5.1

2.5.4 获取驱动版本

Seekfit_By5000io_GetDriverVersion

原 型

```
BOOL Seekfit_By5000io_GetDriverVersion(UINT * puiBuffer, UINT uiCardNumber = 0);
```

功 能

获取驱动程序版本号。

参 数

1 puiBuffer

返回的版本数据，参考 2.5.1

2.6 数据类型定义

2.6.1 MEDIA_TIME_CODE 定义

原 型

```
struct MEDIA_TIME_CODE
```

```
{
```

```
    UINT uiHour;
```

```
    UINT uiMinute;
```

```
    UINT uiSecond;
```

```
    UINT uiFrame;
```

};

功 能

用于表示素材的时间单位。

成 员

1 uiHour

小时时间。

2 uiMinute

分钟时间。

3 uiSecond

秒时间。

4 uiFrame

帧时间。

方 法

3 常见问题

问题：Beyond 系列视音频播出卡和显卡是否会有冲突？

解答：部分厂家的视音频播出卡采用显卡芯片作为 PCI 桥，使用时主板会将其识别为显卡，因而容易与电脑上已经安装的显卡产生硬件上的冲突。Beyond 系列视音频播出卡采用专用的高速 PCI 桥芯片，与显卡芯片没有任何关系，因而不会造成视音频播出卡与显卡的硬件冲突。

问题：视频信号输入 BY5000IO 标清视音频播出卡，视音频播出卡的输出接到电视机上，在软件没有启动时视频图像显示正常，运行软件后电视机无显示，是否视音频播出卡损坏？

解答：请检查视音频播出卡的 SDI 输入、输出是否接反。因为 BY5000IO 标清视音频播出卡 SDI 接口具有断电直通功能，软件没有运行之前，视频信号通过板卡上的继电器直通，所以即使输入、输出接反也能看到图像。软件运行之后，直通继电器断开，如果视频输入、输出接反，自然就没有图像显示了。

问题：视频信号输入 BY5000IO 标清视音频播出卡，视音频播出卡的输出接到电视机上，在软件没有启动时视频图像显示正常，运行软件后电视机显示图像不稳定，有黑白图像上下不停滚动，是否视音频播出卡损坏？

解答：请检查视音频播出卡的当前视频制式与视频源、电视机是否一致。因为 BY5000IO 标清视音频播出卡 SDI 接口具有断电直通功能，软件没有运行之前，视频信号通过板卡上的继电器直通，相当于直接把视频源连接到电视机，视音频播出卡对视频信号没有任何影响。软件运行之后，直通继电器断开，如果板卡工作在 PAL 制，而视频源是 NTSC 制的，或者板卡工作在 NTSC 制，而视频源是 PAL 制的，板卡无法与视频源同步，图像就会不停滚动。

4 技术支持

如果您在使用我们产品的过程中有更多需要或遇到本手册未提及问题，请浏览我们的网

页: www.seekfit.com, 查阅相关信息, 或加入我们的疑难论坛。

技术支持热线, 工作时间: 027-87598453, 非工作时间: 13971152853。



推荐您使用更方便、周到的网络技术支持方式:

QQ: 179483467

MSN: eebyte@hotmail.com

任何问题, 请随时与我们取得联系, 满足您的需求是我们公司全体员工不懈追求的目标!