OK系列图像卡编程手册

Program Handbook for OK Series Frame Grabber

二零一九年六月

北京嘉恒中自图像技术有限公司

Beijing JoinHope Image Technology Ltd.

前 言

北京嘉恒中自图像技术有限公司是国内领先的数字图像产品供应商，总部位于中关村核心地带，是一家聚集了大批业内技术精英，以自主研发为核心竞争力的股份制高新技术企业。我们的前身是中科院自动化所图像部及后来成立的科技嘉仪器仪表有限公司。我公司研发骨干主要来自中科院研究所和重点高校，具有扎实的技术实力，丰富的产品开发经验和良好的用户服务信誉。

嘉恒图像是国内最早的专业图像卡生产商，早在 1988 年就推出了我国首个图像卡产品系列，是国内生产专业图像卡的“老字号”。我公司也是国内为数不多的能够自主研发各种高性能 CCD 和 CMOS 摄像头产品及 DSP、FPGA 图像处理和采集产品的公司之一。目前，我们的主要产品系列有图像采集卡、工业摄像头、嵌入式专用图像采集处理器及基于 DSP 技术的图像采集处理产品等。我们的产品广泛应用于医学影像，生物技术，工业检测，智能交通，保安监控，金融票证，动态分析等领域。我们根据客户的应用需求，提供各种普及档、中档和高档的图像产品，同时提供强大的技术支持 和研发定制服务。

我公司开发图像采集卡已有 20 年以上的历史， 1988 年开发出国内首个 CA 图像卡系列，99 年我们在 CA 图像卡的基础上，推出 OK 系列一代卡，经过不断的技术升级，2003 年后，陆续推出了 OK 系列二代卡，逐步更换了原 OK 系列一代卡。我们图像卡产品不仅在国内处于领先水平，而且与国外产品相比，具有更高的性价比，更丰富的功能，完全可以替代进口产品。国内常见的西门子、飞利浦、GE、岛津等各医疗设备公司生 产的各种 B 超、CT、X 光机、ECT 等影像设备都可以通过我公司的某一图像卡进行采集。

OK 系列二代卡保持了硬件和软件的完全兼容性，但输入输出插头做了统一调整，在可靠性、质量、性能方面得到了全面提升，种类也更齐全，包括 PCI，PCI-E，USB，PC/104+，cPCI 等各种总线接口的全系列图像采集卡。不仅有可以采集标准视频信号的黑白、彩色图像卡和可以接非标准视频信号的高分辨、高灰级的采集卡，而且还开发出一系列具有特殊功能的图像采集处理卡，如 ：递归滤波实时去噪音，实时数据压缩，实时数字减影（DSA），Y/C 和 RGB 分量输入，快速切换的多路采集卡等。OK 系列二代卡继续保持着国内技术领先，国际水平同步的地位。目前的驱动程序可支持 WIN95/98/ME/NT4/2K/XP/Vista/WIN7/WIN8/WIN10。支持 VFW 和 WDM 接口标准。OK 系列二代卡的命名，既保持了 OK 系列卡的连续性又有所区分， 其区分之处是，OK 二代以后的系列卡的名字最后一位是以字母 A、B、C、D 或 K 为序的，如 OK-M10A、OK-C30B、OK-RGB10A 等。

我公司开发的各种基于 PC 的 OK 系列图像采集获取硬件（包括 OK 图像卡，OK 系列 USB 接入、网口接入的采集处理摄像设备等），全部采用统一的硬件无关、操作平台无关的驱动接口标准规范软件，国内首家做到提供给用户的是同一套驱动程序及开发库，使得用户在任一型号的硬件上开发出的软件，执行程序都不用做任何改动， 基本功能都可在其它硬件上使用，从而大大节省了用户在兼容问题上所需花的时间。使得用户一经使用，一劳永逸。OK 系列图像硬件的通用驱动软件开发库具有非常丰富齐备的函数功能，使用更加简单方便。因此，本 OK 系列图像卡编程手册也适用于其他 OK 系列 USB 接入、网口接入的摄像设备的软件编程指导。

我们与广大用户建立起良好的合作关系，以期开发出更多更好的图像产品，为数字图像技术在我国的发展做出贡献。欢迎各界新老用户与我们联系，我们将竭诚为您提供优质的产品和服务。

目 录

亲爱的用户：您好！感谢您使用ＯＫ图像采集卡。

我公司将为您提供免费技术支持及一年免费保修和长期维护服务。若您在使用中遇 到任何问题，或对我公司的产品、技术支持、售后服务有任何建议或意见，欢迎您通过以下方式与我们联系。

电话 ： 400-166-5596 传真 ： 010-82629477 技术支持 ：info@jhi.com.cn 销售咨询 ：sales@jhi.com.cn

并请在主题（Subject）栏务必注明“OK 图像卡”

在联系之前，请您准备好如下材料 ：

1. 使用采集卡的序列号 ： _____
2. 信号源类型 ： _____ （NTSC/PAL/ 非标 黑白 / 彩色等）
3. 主板 厂家 ： _____ 型号 ： _____
4. CPU 型号 ： _____
5. 操作系统 ： _____

（WIN9x/ME/WINNT/WIN2K/WINXP/Vista/WIN7/WIN8/win10）

若您无法准确判断您现在所使用采集卡的型号，请您提供以下信息 ：

1. 购卡日期 ： _____ 2. 购卡单位 ： _____
3. 购 卡 人 ： _____ 4. 购卡时与何人联系 ： _____

若您的问题是在实时采时出现的，请您提供以下信息 ：

1. 显示卡厂家 ： _____ 主芯片型号 ： _____
2. 显 示 模 式 ： _____ 采集数据格式 ： _____

若您的问题是在编程时出现的，请您提供以下信息 ：

1. 编程环境 ： _____ 编程语言及其版本号 ： _____

若有必要，请您留下您的联系方式 ：

1. 联系单位 ： _____ 2. 联系人 ： _____
3. 联系电话 ： _____ 4.E-mail ： _____

您的问题、建议或意见 ：

_____

_____

第 1 章　快速入门

OK 系列图像卡种类齐全，功能丰富，可以满足各种图像应用需求。规范标准的 OK 系列卡开发工具（SDK）为用户提供了一致性、兼容性很强的接口软件库，为 OK 卡用户在应用系统的开发、转换扩展、升级换代、软件保护等方面提供了强有力的支持。由于 OK 系列图像采集卡所应用的领域十分广泛，而不同的用户对硬件系统的结构和软件系统的使用会有很大的差别，所以为了使初用者能快速入门，尽快掌握 OK 系列 PC 机用图像采集处理产品的基本使用，本章介绍了常用库函数的调用，使初用者快速入门。熟悉了这些基本库函数的调用后，用户可根据各自的需要有选择地学习相关的函数使用。本说明书所述内容适合于所有 OK 系列 PC 机用图像产品，驱动程序应是 17.03 版或更新的版本。

1．开发库与驱动程序安装

把安装（SETUP）盘放入光驱 ，在打开的程序中点击“驱动安装”，然后选择需要安装的驱动程序（本说明文档对应的是：OK 系列采集设备驱动及演示程序 ok_setup），按安装程序提示即可方便地安装好开发库、驱动程序及演示程序。

通过安装程序在完全缺省方式下安装以后，所有 OK 系列图像卡的驱动程序都自动安装到了 WINDOWS 的系统目录里。另外在“Program Files”目录下生成一文件夹“OkDemo”，在该文件夹里有如下文件和目录 ：

okdemo.exe	OK 系列图像设备演示程序
okDemoHlp.chm	演示程序在线帮助文件
okdevman.exe	OK 图像设备管理程序
okGigeDemo.exe	OK 千兆网设备演示程序
okPlayback.exe	OK 回显输出卡的参数设置程序
oksetmulnc.exe	多网卡设置程序
okVGACap.exe	VGA、DVI 等高清采集卡参数设置程序
Examples	基本 C 示例源程序文件目录
SDK APIs	用户开发库文件目录

在用户开发库文件目录 SDK APIs 中，有用户编程所需要的文件 ：

okapi32.h(okapi64.h)	库函数的 C 头文件 ，用户编程时嵌入用的头文件
okapi32.lib(okapi64.lib)	接口驱动
dllentry.c	接口驱动
okAutSet.h	自动设置 C 头文件 ，用户编程时嵌入用的头文件
okAutSet.lib	自动设置接口驱动
okUsrHlp.chm	函数在线帮助文件
ok 函数说明和其他使用文档资料	

2.示例程序介绍

在基本 C 示例源程序文件目录中有 OkDemo 的示例程序的源程序、头文件、资源文件及 VC 编译环境文件，用户通过 VC 装入其工作环境后即可进行编译连接，所生成的执行程序即为我们提供的演示程序 okdemo.exe。关于示例源程序的说明可以参见 okdemo.txt 文件中的说明。

在光盘中的 Example 文件夹中，是 OK 设备的常用功能演示，包括实时显示、多卡采集、外触发保存图像、回放文件等，供用户参考。Example 中也有其它编译环境的基本示例程序源代码，如 VC， C++，VB 等各种语言，可供用户编程时参考。用户也可以随时通过我公司的网站 www.jhi.com.cn 下载最新的驱动程序和示例源程序。

3．常用函数介绍

本函数库以尽可能少的函数过程来提供了非常完备的功能。对于初次使用 OK 系列卡的用户，可只阅读下面的常用函数的说明即可。其它的函数可在熟悉了以后或有需要时再去查阅。凡未特别声明的函数，均适用于任何型号的 OK 系列 PC 机用图像采集卡，包括各种 PCI、PCI-E 总线、PC/104+ 总线、USB 串行总线及数字式的图像采集卡，及 USB、网口接口的摄像头。

本节列出了常用函数的简要功能，以便于用户查询和记忆。函数的详细调用方法，请参看函数名后所列页中对该函数的详细介绍。

（1）打开与关闭

打开指定图像卡，返回其句柄，并以之前设置的参数进行初始化。所有对卡操作与控制的函数都要使用该句柄。

关闭指定已打开的图像卡，并存盘当前已设置的参数到初始化文件，然后释放该句柄所用资源。

（2）系统信息

获得为本卡使用的缓存大小及首幅的线性地址，并返回在当前大小设置下，缓存可存放图像的幅数。

获得缓存中指定帧的线性地址。用户可直接使用该地址进行图像处理。

（3）设置采集参数

设置或获得视频源与目标体（如缓存、屏幕等）的窗口大小。

设置并获得视频输入信号的调节选择参数（如源路、对比度、亮度等等）。

（4）视频采集

（5）数据传送

（6）文件读写

（7）信号参数

（8）实用程序

4．编程说明

我公司随卡免费提供的驱动程序与开发库（SDK），具有许多国内外同类产品所不 具有的优越性能。用户不必再担心更换操作系统平台或更换图像卡带来的兼容性问题。 同时还提供有各种编程示例程序，非常方便用户的二次开发。

需要注意的是，无论用户使用哪种型号的 OK 系列图像卡，用户所要直接调用的 动态库都是 OKAPI32.DLL。其它动态库均为内部使用，用户无需也不能直接使用。
我们提供的基本演示源程序为 C 程序，编译环境为 Visual C 5.0。要说明的是，
不要忘记在设置编译环境时设置（Byte Alignment）对齐字节数为 1 字节。还有在函数说明中有很多函数的参数，高字节和低字节分别对应着不同功能的参数，在进行某项设置时，应该将整个参数读出，然后修改对应功能的高字或者低字的部分，然后再将参数设置回去，以防止对设备参数不必要的修改，引起问题。如需要分解的编程示例或其它编译系统的范例可到我公司的网站下载，或者参考用户安装光盘里的 ok_setup\EXAMPLE\sample。

用 C 语言调用动态库 OKAPI32.DLL 中库函数的连接方法有两种 ：一是静态链接，即直接链接输入库 OKAPI32.LIB ；二是动态链接，可以通过加入我们提供的 DLLENTRY.C 源程序来实现。注意，对于非 Visual C 的其它各类 C 编译系统目前则只能用第二种 方法。

如用户需要用 Visual Basic 开发程序，则可通过加入我们提供的为 VB 用的 OKAPI32.BAS 库函数声明文件。由于 VB 无指针功能，因而库函数中个别需要引用指 针的函数，VB 程序不能使用。一般来说 VB 做菜单比较方便，但做图像处理时，相对 C 来说，有些不方便且效率较低。因此建议用户用 C 做复杂的图像处理过程，生成动 态库供 VB 调用，而用 VB 做菜单和控制功能。

如果用户需用我们尚未提供示例的其它编译语言，则需自行根据我们提供的函数 说明（参见 OKAPI32.H）生成相应的声明文件。注意，本说明书是按 C 语言的格式进 行说明的。

还有一用户需要注意的是，由于 WINDOWS 在 256 色模式下的缺省调色板是彩色的，所以当 VGA 设置成 8 位，即 256 色模式时，用户需要在自己的程序里加入一设置 黑白逻辑调色板的函数，否则实时采集的或从文件中装入的黑白图像会成为伪彩色或 黑白分层。具体方法可参见例子 okdemo.c 中的源函数 SetPaletteToWnd。

下面为一 C 的编程基本框架示例，详细请参考提供的演示程序源程序。

```
BOOL BasicProc(HWND hWnd)
{
    long lIndex,num;
    long lRGBForm;
    RECT    rcVideo,rcBuffer; HANDLE    hBoard;
    lIndex=-1;
    //open specified board hBoard=okOpenBoard(&lIndex); if(hBoard)
      { //if success
        Sleep(500); //waiting while for initilizing
```

```
        //-----set basical parameter
        //this exam. select VIDEO 1 (if S-VIDEO 1 than 0x100)
        okSetVideoParam(hBoard,VIDEO_SOURCECHAN, 0x0);
        //get current vga mode lRGBForm=LOWORD(okSetCaptureParam\
        (hBoard, CAPTURE_SCRRGBFORMAT, -1));
        //set video source format to same as current vga
        okSetVideoParam(hBoard,VIDEO_RGBFORMAT, \lRGBForm);
        //set target buffer format to same as current vga okSetCaptureParam(hBoard,\
        CAPTURE_BUFRGBFORMAT, lRGBForm);
        //set video source rect as PAL rcVideo.left=rcVideo.top=0; rcVideo.right=768;
        rcVideo.bottom=576;
        okSetTargetRect(hBoard,VIDEO,&rcVideo);
        //-----1 capture to SCREEN (alive on VGA)
        //set target(here is VAG) rect GetClientRect(hWnd,&rcScreen);
        MapWindowPoints(hWnd,HWND_DESKTOP,\
        (LPPOINT)&rcScreen,2);
        okSetTargetRect(hBoard, SCREEN, &rcScreen );
        //or okSetToWndRect(hBoard,hWnd);
        if( okCaptureToScreen(hBoard) <0 ) MessageBox(NULL,"Can't directly capture \
        on current VGA mode !","Error",MB_OK);
        //or okCaptureTo(hBoard,SCREEN,0,0); Sleep(1000); //just waiting a while  for
        aliving okStopCapture(hBoard);

        //-----2 capture to BUFFER


        rcBuffer.left=rcBuffer.top=0;
        rcBuffer.right=768;
        rcBuffer.bottom=576;
        //set target(here  is buffer) rect okSetTargetRect(hBoard, BUFFER, &rcBuffer);

        //set to not waiting end, return immediately okSetCaptureParam(hBoard, \
        CAPTURE_SEQCAPWAIT, 0);
        num=okGetBufferSize(hBoard,NULL,NULL);//
        //you can here  set your  callback  functions  if necessary
        //okSetSeqCallback(hBoard,BeginCapture,BackDisplay,\ EndCapture);
        okCaptureTo(hBoard,BUFFER,0,num);//
        sequence  capture  to frame  buffer

        //way 1.
        //while( okGetCaptureStatus(hBoard,0) )
        {
            //  SleepEx(5,TRUE); //best do sleep when loop waitting
        //}
        //way 2. okGetCaptureStatus(hBoard,1);

        //close  specified board
        okCloseBoard(hBoard);
        return TRUE;
    }
    return FALSE;
}
```

第 2 章　OK 系列卡型

1. OK 系列卡的兼容性

OK 系列图像卡支持即插即用标准，全 32 位驱动软件支持 WIN95/98、WINME、WINNT4、WIN2000、WINXP、Vista、WIN7。随卡提供给用户的都是同样的安装盘，可安装在上述任一操作系统上。由于所有 OK 系列图像卡均采用统一的用户开发接口标准，所以用户开发的程序不必做任何改动就可在上述任一操作系统上的任一 OK 系列图像卡上运行， 使用户可专心应用开发而不必过多顾及图像卡及操作系统的兼容问题。OK 系列图像卡驱动软件支持一机多卡（同种和不同种均可）同时操作，逐帧并行处理等。即使是同型号的多块卡，它们的参数设置，如对比度，亮度等，也都是完全独立的。

OK 系列图像卡的每一种型号都有唯一的卡型识别码，而在软件中用户使用的是不唯一的类型码。每块卡在出厂前都贴有该卡的序列号，序列号中的前四位就是卡型识 别码。用户通过卡上的序列号就可判别卡型。识别码的定义规则为，第一位为卡的类 型号，第二位为卡的类序号、或与第一位合为类型号、或与后两为合为序列号，后两 位代表卡的序列号。

目前已有类型：（1）黑白采集卡为 10XX，如 OK_M40 卡为 1040；（2） 彩色采集卡为 20XX，如 OK_C30 卡为 2030 ；（3）RGB 分量采集卡为 30XX，如 OK_ RGB10 卡为 3010 ；（4）监控采集卡为 40XX，如 OK_MC30 卡为 4030 ；（5）便携与嵌入 式类采集卡为 5X XX，如 USB 的标准彩色采集卡 OK_USB20A 卡为 5220，PC/104+ 的 RGB 采集卡 OK_PR30A 为 5330 ；（6）数字采集卡，CameraLink 的为 61XX， LVDS 的为 62XX ；（7）线扫描采集卡 OK_LS100A 为 7100 等。PCI-E 卡的第二位为 3 或 4 ；PCI-X 卡的第二位为 5 或 6。

第二位（类序号）为 0 代表基本型，而由于诸如替换、备选、升级等原因，开发了新的同等级类型卡，则类序号依次递增。例如 ：OK_C20 的备选卡 OK_C20N，其识 别码为 2120 ； OK 二代卡的类序号为 2，所以 OK_C20A 卡，其识别码为 2220 ； 但其基 本型的识别码均为 2020。另外，当新设计生产一种新卡用来兼容代替某种停产的卡型时，其识别码的序列号与被兼容代替的卡不同，但为兼容，其类型码是采用同样的码。 如 OK_C20 卡的兼容代替卡 OK_C21，其识别码为 2021，但其类型码与其原基本型 OK_C20 一样仍是 2020。OK_C20 的 OK 二代卡 OK_C20A，其识别码为 2220。为了与 OK 一代卡有所区分，OK 系列二代卡名字的最后一位是以字母 A、B、C、D 或 K 为序的， 如 OK-M10K、OK-C30B、OK-RGB10A 等。

之所以在软件中不直接使用唯一的识别码，而使用不唯一的类型码，是为了用户更换卡时的兼容。同类型但类序号或序列号不同的卡都具有相同的类型码，都等于其原基本型的卡型识别码。所以用户已开发的软件，不用做任何改变，就可以把同等级 的新型号的卡直接当作其原基本型卡一样的使用，保证了向后兼容性，因而用户就不 必担心当前使用的某一卡将来会否停产。

基本型或最初开发出的卡型其识别码就定义为类型码，因此类型码是不唯一确定卡型的码。同类型但类序号或序列号不同的卡都具有相同的类型码，都等于其原基本型的卡型识别码。

用户在编程时，如果按缺省方式打开卡，即根本就不指定卡型，那么任何一块 OK 卡都可以一样地被打开使用。但是如果出于某种需求，所编写的软件只能选用某一类 型的卡，这就需要指定卡型。按我们在宏定义中提供的都是基本型或最初开发出的卡型的定义，因而用户使用的是不唯一的类型码，这样就可保证对用户软件的兼容。因此用户已开发的软件，

不用做任何改变，就可以把同等级的新型号的卡直接当作其原 基本型卡一样的使用，保证了向后兼容性，因而用户就不必担心当前使用的某一卡型将来停产，而不能使用新的替代或升级卡。

2. 常用 OK 系列卡类型定义

每一种类型的 OK 系列图像卡都有自己唯一的识别码。每块卡上都有一个序列号， 序列号的前四位即为该卡的识别码。下面列出了目前常用的各型卡其基本型的识别码， 即类型码的宏定义，详细定义见最新的 OKAPI32.H 文件。

黑白系列卡 ：
OK_M10N 1010（已停产） OK_M10M 1013（已停产） OK_M80 1080（已停产）
OK_M80K 1081（已停产） OK_M20H 1022（已停产） OK_M40 1040（已停产）
OK_M60 1060（已停产）

OK 二代卡型 ：
OK_M10A 1212（基本型 OK_M10M ： 1013）
OK_M10B 1213（基本型 OK_M10L ： 1014）
OK_M10K 1218 (基本型 OK_M80K)
OK_M20A 1222（基本型 OK_M20H ： 1022）
OK_M20B 1223
OK_M40A 1240（基本型 OK_M40 ： 1040）
OK_M40B 1243
OK_M50A 1250
OK_M50B 1253
OK_M50K 1258
OK_M60A 1260（基本型 OK_M60 ： 1060）
OK_M60B 1263
OK_M30A 1230（基本型 OK_M30 ： 1030）
OK_M30B 1233
OK_M30K 1233
OK_M70A 1270（基本型 OK_M70 ： 1070）
OK_M40A_E 1340（已停产）

彩色系列卡 ：
OK_C20 2020
OK_C30 2030
OK_C30S 2031（已停产）

OK 二代卡型 ：
OK_C20A 2220（基本型 OK_C20 ：2020） OK_C30A 2230（基本型 OK_C30 ：2030）
OK_C60A 2260（已停产）
OK_C60B 2263
OK_C80A 2280（基本型 OK_C80 ： 2080）

RGB 系列卡 ：
OK_RGB10 3010（已停产） OK_RGB20 3020（已停产） OK_RGB30 3030（已停产）

OK 二代卡型 ：
OK_RGB10A 3210（已停产）（基本型 OK_RGB10 ： 3010）
OK_RGB10B 3213（基本型 OK_RGB10 ： 3010）

OK_RGB21A 3221（基本型　　OK_RGB20 ： 3020）
OK_RGB20B 3223
OK_RGB30A 3230（基本型　　OK_RGB10 ： 3030）
OK_RGB30B 3233
OK_RGB60A 3260
OK_RGB60B 3263
OK_RGB60C 3265
OK_RGB60C-E　3365（已停产）
OK_VGA40A 3240
OK_VGA40B 3243

多路监控系列卡 ：
OK_MC10　4010（已停产）
OK_MC16　4016（已停产）
OK_MC20　4020（已停产）
OK_MC30　4030
OK_MC30A-E　　4330
OK_MC20B　4223
OK_MC40B　4243

OK 二代卡型 ：
OK_MC10A　4210（基本型　　OK_MC10 ： 4010）
OK_MC16A　4216（基本型　　OK_MC16 ： 4016）

嵌入与便携系列卡 ：
OK_PC10A　5210
OK_PC20A　5221
OK_PR20B　5323
OK_PR30A　5330
OK_PM10A　5410
OK_PM20A　5420
OK_USB20A　5220
OK_USB20B　5223

数字系列卡 ：
OK_CL20A　6120
OK_CL40A　6140
OK_CL60A　6160
OK_LV20A　6220
OK_LV40A　6240
OK_LV60A　6260
OK_CL20B　6123
OK_LV20B　6223
OK_CL20A_E　　6320
OK_LV20A_E　　6420
OK_CL40A_E　　6340
OK_LV40A_E　　6440

线阵系列卡 ：
OK_LS100A　7100
OK_LS120A　7120

3.OK 二代系列图像卡连接线规范

OK 二代卡的 3 排 15 针 (HD15) 综合输入插头针脚定义 ：

脚号	功 能		标号/插头色
	黑白卡/彩色输入卡	RGB输入卡	
1	第1路视频输入	红路分量输入	1/红
2	第2路视频输入	绿路分量输入	2/绿
3	第3路视频输入	蓝路分量输入	3/蓝
4	保留/左声道入	保留	
5	保留/右声道入	保留	
6	接地	接地	
7	接地	接地	
8	接地	接地	
9	空	空	
10	接地	接地	
11	外触发输入	外触发输入	7/黑
12	保留	保留	
13	第4路输入或行/复合同步	行/复合同步输入	4/黄
14	第5路输入或场同步	场同步输入	5/白
15	第6路输入	保留	6/灰

BNC 输入的等同连接 ：
（1）彩色卡 ： 与综合输入第一路视频相连 ；
（2）黑白卡 ： 与综合输入第一路视频相连 ；
（3）RGB 卡 ： 与第一路 RGB 输入的红路输入相连
注 ： 数字卡 (如 CAMERALINK 与 LVDS 系列) 使用的是的专用输入插头。

OK 二代卡的 3 排 15 孔 (HD15) 综合输出插座孔脚定义 ：

脚号	功 能		
	黑白输出	RGB输出	数字(CL/LV系列)卡输出
1	保留	红路分量输出	红路分量输出
2	视频输出	绿路(带同步)分量输出	绿路(带同步)分量输出
3	保留	蓝路分量输出	蓝路分量输出
4	保留	保留	保留
5	保留	保留	保留
6	接地	接地	接地
7	接地	接地	接地
8	接地	接地	接地
9	空	空	输入场同步输出
10	接地	接地	接地

11	保留	保留	外触发输入
12	保留	保留	保留
13	行/复合同步输出	行/复合同步输出	行/复合同步输出
14	场同步输出	场同步输出	场同步输出
15	保留	保留	保留

4. 开发 OK 卡产品须知

开发基于 OK 卡产品的用户和 OEM 厂商应注意 ：

所有的 OK 系列卡的驱动程序 (均以 ok 打头) 都安装在 WINDOWS 系统目录 SYSTEM(WIN95/98/ME) 或 SYSTEM32(WINNT4/2K/XP/Vista/WIn7) 中，这些驱动程序包括用户 开发用的图像卡接口动态库 OKAPI32.DLL，所有的用户用库函数都在此动态库中，它是用户唯一需直接调用的驱动程序 ；图像卡的内部驱动程序及设备驱动程序 okacap. dll，okaux.dll，okm20.dll，okc20.dll，okc21b.dll，okc30.dll，okc50.dll，okc70. dll，okc80.dll，okcn30.dll，okdvi40.dll，okfp60.dll，okGigE20a.dll，okIC20.dll，okie40.dll，okls20.dll，oklv20.dll，okmc20.dll，okmc40.dll，okql20.dll，okxl20.dll，okNet20.dll，okpr60.dll，OKR30.DLL，okARM2.dll，okusb20.dll，okusbtrf.dll，okjpg.dll ，okjpeg.dll，okj2k.dll，okmjpg.dll，okmpg2.dll，okmpg4.dll，oktiff.dll，clseroka.dll，usemfcdll.dll 及 okusbdrv.sys、ok1394.sys 等。还有 WIN95/98/ME 用的 VXD 系统 虚拟设备驱动 okadrv.vxd 与大缓存的 DOS 预分配程序 okalloc.exe，和 WINNT4/2K/XP 用的设备服务驱动 okntdrv.sys。对 WIN95 的早期版本，还需要 32 位公用系统动态库 msvcrt.dll。如果用户利用 OK 系列卡开发出自己的应用系统，并希望把 OK 系列卡所需 的的驱动程序打包到自己的安装程序中，应把 okadrv.inf 即插即用安装信息文件和设备驱动程序 okadrv.vxd、okalloc.exe、okusbdrv.sys 与 okntdrv.sys 考备到安装盘中，然后把其它所有上述驱动程序文件全打包进去。如需自己打包并有特殊要求的，也可与我们直接联系，我们将会提供更为详尽的解决方案和有力的技术支持。

另外，大多数 OK 卡都具有为用户软件加密的功能，以保护用户的软件不被盗版或者用户有 linux 软件开发的需求及 64 位系统下应用的需求，需要驱动程序等可以与我们联系。

第 3 章　OK 卡函数库

1，库函数综述

我公司自从 99 年推出全新 OK 系列图像卡通用标准接口开发库后，得到了用户的 广泛认可，经过我们不断地扩充完善，6.51 以后版的驱动程序均可用于 WIN95/98/ME 和 WINNT4/2K/XP/Vista/WIN7 任一操作平台上，适用于本公司生产的各种型号的图像卡，也就是说开发库是与操作系统及图像卡硬件无关的。 对于一特定的图像卡，如不支持某一硬件功能，则调用该函数后，不会做任何事，而仅返回一不支持码（-1）。因此用 户开发程序时不必顾虑硬件兼容性。当有更新型的卡时，以前的程序不必改动，而只需增加新的功能即可。因而用户最新开发的程序也不必做任何改动，其基本功能仍可 在各种型号的卡上使用。

通过本函数库可以使用户很方便地在一台机器上控制多块不同种和同种型号的图像卡。即使是同型的多块卡，它们的参数设置（如对比度，亮度等）也是完全独立的。

由于调用序列采集函数时，可以立即返回，因而可以进行并行处理。由于序列采集函数可以在采集过程中逐帧回调，因而可以进行逐帧并行处理。

本开发库的驱动程序及设备服务驱动程序均是 32 位编写，因而本函数库是完全 32 位方式。而不是由 32 位库函数首先转换成 16 位调用，然后再去调用 16 位驱动程序的 32 与 16 位的混和方式。因而充分发挥了 32 位的优点。

在开发库中，对经常要涉及的几项硬件通称为目标体 (TARGET)，并分别在 OKAPI32.H 作了宏定义。它们是 ：

a. 视频源 ：

VIDEO（视频输入）：视频信号输入源 ；

b. 采集目的体 ：

SCREEN（屏幕）：VGA 显存 (计算机显示器) ；

BUFFER（缓存）：由设备驱动在主机内存中预申请的序列缓存 ；

FRAME（帧存）：采集卡上自带的存储体 ；

MONITOR（监视器）：模拟视频监视器（此项目标体只对可支持回显的 M40A， M60A，M30A 等有效）。

GRAPH（图形位）：采集卡上用于图形位功能的存储体；

以上这几项采集目标体除 GRAPH 外都是硬件可以直接采入或输出的，采集时调用 okCaptureTo，回显时调用 okPlaybackFrom。而 GRAPH 目标体是只能通过函数对其内容进行设置的。

MEMORY（用户程序申请的内存）和 FILE（图像文件）：这两项是可间接采入或输出的目标体。采集时调用 okCaptureByBuffer，回显时调用 okPlaybackByBuffer。其中用户内存，在有些数据传送函数调用中可以直接用指针，有的则需用 BLOCKINFO 结构来传递， 所定义的结构变量应先全部置零。文件则可以在调用中直接用文件名（带文件类型扩展名）来传递，目前支持的文件类型有自定义序列文件”.SEQ”（文件头为 20 个字节 的 SEQINFO 结构，紧跟其后按行顺序存放各帧原始图像数据）和目前最常用的 BMP 图像格式文件“.BMP”，及裸格式文件“. RAW ”。至于其它格式文件，用户可自行转换。

本说明书常用到的有关采集的几个名词，解释如下 ：

连续采集 ：又叫实时采，是由图像卡连续不断地向指定目的体一个指定的固定位置传送输入视频源当前的图像，这时如果输入视频源的图像是动态的，那末该位置的图像也一直

是动态的、活动的。这种情况下，完全是硬件工作，CPU 是不被占用的。 当目的体是显存时，由于这时可以直接在显示器上看到动态实时的图像，又习惯称为实时显示。

序列采集 ：是由图像卡将输入视频源的图像，逐帧顺序地采向指定目的体的序列存储空间，令目的体可以保留一段动态图像。这种方式要求目的体具有存储多幅图像的空间。这种情况下，除了硬件参与工作外，CPU 也部分参与，用来每采集一幅结束后更换卡所采向的下一幅的地址。线程控制的序列采集对 CPU 的占用就比较多一些， 中断控制的序列采集则只占用非常少的 CPU 时间。存储序列图像的空间是有限的，所 以序列采集是一段过程，采完指定的帧数后就停止采集了。 而循环序列采集就是在指 定的图像存储空间内顺序循环不断的采集。

单帧采集 ：图像卡采集完紧跟采集指令后的一幅图像后，就停止不再采集了。是 序列采集的一个特例。

静态缓存 ：在机器一启动就由设备驱动在主机内存中预申请到的缓存叫静态缓存， 静态缓存在机器启动之后不能改变。在机器启动之后再通 OK 卡的 API 函数申请的缓存称为动态缓存，动态缓存可以随时释放掉，但不一定能随时申请到。我们推荐用户尽可能只使用静态缓存，而只在必须时才使用动态缓存。

回显 ：通过图像卡输出图像到专门的图像显示器（计算机显示器之外）叫回显。可以回显缓存（BUFFER）或帧存（FRAME）中的一幅图像，也可以回显缓存中的一组序列图像。

常用的几个结构变量定义 ：

图像块信息结构 ：

```
typedef struct _blockinfo
{
    SHORT   iType;                  // 图像类型（如 BK、BM）
    SHORT   iWidth;                 // 宽度
    SHORT   iHeight;                // 高度
    SHORT   iBitCount;              // 象元位数
    SHORT   iFormType;              // 图像格式
    union
    {
        struct
        {
            SHORT   lBlockStep;     // 图像块跨距的低字
            SHORT   iHiStep;        // 图像块跨距的高字
        };
        DWORD dwBlockStep;          // 图像块跨距
    };
    union
     {
        struct
         {
            SHORT   lTotal;         // 图像总帧数的低字
            SHORT   iHiTotal;       // 图像总帧数的高字
        };
        DWORD dwTotal;              // 图像总帧数
    };
    SHORT   iInterval;              // 图像帧间隔数
    LPBYTE      lpBits;             // 图像数据指针 / 文件路径名
    LPBYTE      lpExtra;            // 额外数据（如调色板等 ) 指针
} BLOCKINFO, *LPBLOCKINFO;
```

注意：此目标体作为传送数据的目地体或数据源的时候，程序需要按 1 字节对齐来编译才能正常运行，否则会导致传送失败！

序列文件信息结构 ：

```
typedef struct
```

```
{ //file info for seq
    SHORT   iType;                  // 文件类型（SQ、JP）
    SHORT   iWidth;                 // 宽度
    SHORT   iHeight;                // 高度
    SHORT   iBitCount;              // 象元位数
    SHORT   iFormType;              // 图像格式
    union
     {
        struct
        {
            SHORT   lBlockStep;     // 图像块跨距的低字
            SHORT   iHiStep;        // 图像块跨距的高字
        };
        DWORD dwBlockStep;          // 图像块跨距
    };
    union
    {
        struct
         {
            SHORT   lTotal;         // 图像总帧数的低字
            SHORT   iHiTotal;       // 图像总帧数的高字
        };
        DWORD dwTotal;              // 图像总帧数
    };
    SHORT   iInterval;              // 图像帧间隔数
} SEQINFO;
```

常见的错误码意义：

错误代码：	错误原因：
ERR_NOTFOUNDBOARD（1）	没有发现可正常使用的 OK 系列卡 。可能是主机没有插接 OK 卡，或插接的 OK 卡接连接不可靠，或卡有问题。
ERR_NOTFOUNDVXDDRV（2）	没有发现 OK 卡的设备驱动服务程序。可能是没有按即插即用方式正确安装 OK 卡的设备驱动，或没有插 OK 卡，或其它原因造成找不到 OK 卡的设备驱动服务程序。
ERR_NOTALLOCATEDBUF（3）	没有从主机内存中预申请到 OK 设备专用的缓存。可能是没有正确安装 OK 设备的驱动程序，或设置申请缓存的大小为 0。
ERR_BUFFERNOTENOUGH（4）	当前设备的缓存空间不够大。在 demo"帮助"下的"分配缓存"中把预分配的缓存大小的值改大，按照要求重启电脑看能否解决，如果依然报错请确定缓存的大小是否足够，还有请确定在"帮助"下的"锁定缓存"中，是否锁定了该卡使用的缓存大小。
ERR_BEYONDFRAMEBUF（5）	采集图像数据的大小超出了缓存的大小。
ERR_NOTFOUNDDRIVER（6）	没有找所用 OK 卡对应的驱动程序。可能是没有正确安装驱动程序，或某种原因造成了部分驱动程序的丢失。
ERR_NOTCORRECTDRIVER (7)	目前安装的采集卡的驱动不正确。可能是没有正确安装驱动程序。

ERR_MEMORYNOTENOUG（8）	动态库没有分到足够的内存。
ERR_FUNNOTSUPPORT（9）	该卡不支持当前的功能。
ERR_OPERATEFAILED（10）	功能调用异常。
ERR_HANDLEAPIERROR（11）	在调用 okapi32 中的函数时给入卡句柄错误。
ERR_DRVINITWRONG（12）	所用 OK 卡对应的驱动程序初始化时发生了错误。可能是该卡插接不好或硬件故障而导致不能正确控制该卡，或某种原因造成了驱动程序的损坏，建议客户可以把卡拔插一下，用橡皮擦一下金手指以确定卡的硬件连接没有问题，还有重新覆盖安装一遍卡的驱动。
ERR_RECTVALUEWRONG（13）	用户设置的采集目标区域的参数不正确。
ERR_FORMNOTSUPPORT（14）	当前所用的 OK 卡不支持所设置的 RGB 数据位格式。
ERR_TARGETNOTSUPPORT（15）	当前目标体不支持该功能。常见于"实时显"，可能是显卡没有安装适合当前系统的驱动，建议客户更换显卡的驱动。
ERR_NOSPECIFIEDBOARD（16）	在主机中没有插接所指定的某类型的 OK 卡。这是在用户按指定卡类型打开 OK 卡时，在当前主机中没有找到指定的该类型的 OK 卡，或 机器中没有任何 OK 卡。

2．库函数分类

本函数库以尽可能少的函数过程（便于用户记忆）来提供了非常完备的功能。本函数库分为如下几大类 ：（一）基本功能，介绍了各种型号的 OK 卡基本都支持的功能， 最常用的函数都在这一类里，（二）查询卡信息专项功能，列出了一些特殊卡才支持的函数功能， 如屏蔽位、查找表等，（三）多卡控制，主要是为需要一机同时控制多块卡所设置的一些函数，还有多通道、动态缓存有关的函数，（四）实用程序，提供了一些参数设置函数，写字符画图形、编解码等函数。（五）音频采集，为具有音频采集功能的卡提供的音频采集有关的函数。

对于一般用户或刚开始使用图像卡的用户可只阅读基本功能部分即可。为方便用户查询，在本手册的最后一章列有所有这些函数的索引。

第 4 章　库函数详述

1．基本功能

（1）打开与关闭

HANDLE WINAPI okOpenBoard(MLONG *iIndex);

功能：打开指定图像卡。从驱动程序自动生成的初始化文件（在 WINDOWS 目录下的 OKADRV.INI）中读出上次关闭前设置的参数（如是首次则用**系统缺省值**）进行初始化，并获得所需系统设置，创建该卡的引用句柄，以供各功能函数引用。

参数：

iIndex：1、指定要打开的卡在已安装连接在计算机中的所有 OK 系列图像卡的索引号（按插在主板 PCI 插槽中的先后顺序编号），该索引号是 0 起始。

2、可以是要打开卡的类型码，这时其低三字节为类型码，最高字节为同种类型卡顺序号。

3、还可以是 IP 地址字符串的指针，用来指定打开对应 IP 地址的千兆网摄像头。

注意：该变量是以地址方式传递的，原因见下面的说明。

对于初次使用的用户可以简单设其值为 -1（推荐）。这样设置就可以通过我们提供的设备管理器来随时改变用户程序的缺省选用卡。

返回值：如果调用成功，返回指定卡的引用句柄；如果未发现可用卡，返回 0（FALSE），详细错误码可通过调用 okGetLastError() 获得。

说明：如果要打开用户通过在“控制面板”中的“OK 设备管理器”所设置的缺省选用序号（或类型码）（如未设置，缺省值为 0）所对应的卡，可以设定 iIndex=-1。如 iIndex 输入 -1 或类型码，则调用结束 ,iIndex 被赋值为该卡在当前机器中实际的索引（顺序）号。这就是要传递地址的原因。类型码的定义参见头文件 OKAPI32.H。

相关函数：okCloseBoard，okGetLastError，okOpenBoardEx，okCloseBoardEx。

HANDLE WINAPI okOpenBoardEx(MLONG *iIndex, HANDLE hParBoard, LPVOID lParam);

功能：这个是 okOpenBoard 的扩展函数，用来打开指定的图象卡。从驱动程序自动生成的初始化文件（在 WINDOWS 目录下的 OKADRV.INI）中读出上次关闭前设置的参数（如是首次则用系统缺省值）进行初始化，并获得所需系统设置，创建该卡的引用句柄，以供各功能函数引用。

参数：

iIndex：1、指定要打开的卡在已安装连接在计算机中的所有 OK 系列图像卡的索引号（按插在主板 PCI 插槽中的先后顺序编号），该索引号是 0 起始。

2、可以是要打开卡的类型码，这时其低三字节为类型码，最高字节为同种类型卡顺序号。

3、还可以是 IP 地址字符串的指针，用来指定打开对应 IP 地址的千兆网摄像头。

注意：该变量是以地址方式传递的，原因见下面的说明。

对于初次使用的用户可以简单设其值为 -1（推荐）。这样设置就可以通过我们提供的设备管理器来随时改变用户程序的缺省选用卡。

hParBoard ：是想要打开的新卡的父卡句柄,如果打开卡的时候用到了父卡句柄，**一定**要在关闭父卡句柄之前确保该卡句柄已经关闭，该功能目前还没有用到，应给 NULL。

lParam ：保留参数，目前没有用到，应给 NULL。

返回值：如果调用成功，返回指定卡的引用句柄；如果未发现可用卡，返回 0（FALSE），详细错误码可通过调用 okGetLastError()获得。

说明：如果要打开用户通过在“控制面板”中的“OK 设备管理器”所设置的缺省选用序号（或 类型码 ）（如未设置，缺省值为 0）所对应的卡，可以设定 iIndex=-1。如 iIndex 输入 -1 或类型码，则调用结束 ,iIndex 被赋值为该卡在当前机器中实际的索引（顺序）号，这就是要传递地址的原因。类型码的定义参见头文件 OKAPI32.H。

相关函数：okOpenBoard，okCloseBoard，okCloseBoardEx，okGetLastError

BOOL WINAPI okCloseBoard(HANDLE hBoard);

功能 ：关闭所指定的已打开图像卡。并存盘当前设置的参数到初始化文件（在 WINDOWS 目录下的 OKADRV.INI），以便下次打开时初始化之用，然后释放该句柄所用资源。

参数 ：

hBoard ：指定要关闭卡的引用句柄。

返回值 ：返回 1（TRUE）。

说明 ：不再使用某卡时，一定要调用该函数关闭该卡。

注意 ：在采集（回调）过程中不要调用此函数关闭卡句柄。

相关函数 ：okOpenBoard，okOpenBoardEx，okCloseBoardEx

BOOL WINAPI okCloseBoardEx(HANDLE hBoard, DWORD dwSave);

功能 ：这个是 okCloseBoard 的扩展函数，同样用来关闭所指定的已打开图像卡。但是可以由用户指定是否存盘当前设置的参数到初始化文件（在 WINDOWS 目录下的 OKADRV.INI），以便下次打开时初始化之用，然后释放该句柄所用资源。

参数 ：

hBoard ：指定要关闭卡的引用句柄。

dwSave：选择是否保存参数后再释放句柄。=1：保存后再释放，和 okCloseBoard 功能一样；
=0：不保存参数，直接关闭句柄。

返回值 ：返回 1（TRUE）。

说明 ：不再使用某卡时，一定要调用该函数关闭该卡。

注意 ：在采集（回调）过程中不要调用此函数关闭卡句柄。

相关函数 ：okOpenBoard，okOpenBoard，okOpenBoardEx，okCloseBoardEx

（2）错误信息

INT32 WINAPI okGetLastError();

功能 ：获得最后调用的错误码。

参数 ：无

返回值 ：返回错误码。错误码的宏定义参见头文件 OKAPI32.H。

说明 ：该函数只返回最后调用函数可能发生的错误码，前一次的已不存在。

相关函数 ：okOpenBoard。

（3）设置\获得参数

MLONG WINAPI okSetVideoParam(HANDLE hBoard,INT32 wParam, MLONG lParam);

功能 ：设置并获得视频输入信号的调节参数（如对比度、亮度等）。
参数 ：
hBoard ：输入卡句柄。
wParam：指定设置视频项目，所支持的项目如下（可参见 OKAPI32.H 中的宏定义）：
*VIDEO_RESETALL(0) ：重置所有项的参数值为系统缺省值，lParam 为 0。
常用参数设置：
*VIDEO_SOURCECHAN(1) ：视频源路选择，lParam 指定源路。

复合输入	第一路 0x00，第二、三路为 0x01，0x02 以此类推。
Y/C（又称 S-Video）	输入第一路为 0x100，第二、三路为 0x101,0x102 其他类型的视频源的后几路请参考复合和 Y/C 输入源的后几路。
RGB/VGA	输入第一路为 0x200。
YPbPr	输入第一路为 0x400。
DVI/HDMI(RGB)	视频输入为 0x800。
HDMI(YPbPr)	视频输入为 0x900。
SDI	视频输入为 0xa00

对于 MC20，四路合一画面，所以选择源路并不进行切换路，而是作为要调节对比度、亮度、色度和饱和度的当前源路 ；
lParam 的高字等于 1,2,3,4 分别对应四合一画面的左上，右上，左下和右下位置，等于零为不改变原有设置，系统初始的缺省设置是按上面的顺序放置。
lParam=0XFF ， 仅返回本卡可支持的复合输入路数 ；
lParam=0X1FF， 仅返回本卡可支持的 Y/C 输入路数 ；
lParam=0X2FF， 仅返回本卡可支持的 RGB/VGA 输入路数；
lParam=0X4FF， 仅返回本卡可支持的 YPrPb 输入路数；
lParam=0X8FF， 仅返回本卡可支持的 DVI/HDMI(RGB) 输入路数；
lParam=0X9FF， 仅返回本卡可支持的 HDMI(YPbPr) 输入路数；
lParam=0XAFF， 仅返回本卡可支持的 SDI 输入路数。

*VIDEO_BRIGHTNESS(2) ：亮度调节。
lParam 的低字 为亮度调节值，范围 0~255。
lParam 的高字 为基色通道号， 对非 RGB 分量输入卡， lParam 的高字必须为零；对 RGB 分量输入卡（ 如 OK_RGB21A） lParam 的高字为要调节的基色通道号：0 为红、1 为绿、2 为蓝。返回值的低 3 个字节，从低到高分别对应红、绿、蓝的当前值。

*VIDEO_CONTRAST(3) ：对比度调节。
lParam 的低字为对比度调节值，范围 0~255。
lParam 的高字为基色通道号，对非 RGB 分量输入卡， lParam 的高字必须为零 ；对 RGB 分量输入卡（ 如 OK_RGB21A） lParam 的高字为要调节的基色通道号，0 为红，1 为绿，2 为蓝。返回值的低 3 个字节，从低到高分别对应红、绿、蓝的当前值。

*VIDEO_COLORHUE(4) ：色调调节。
lParam 的低字为色调调节值，范围 0~255。对于 YUV/YPbPr 信号。
lParam 的高字=0 表示调节的是 V/Pr 分量；=1 表示调节 U/Pb 分量。只对彩色复合视频输入有效，如 C20A,C30A 卡可以设置此参数，RGB 输入无此设置。

*VIDEO_SATURATION(5) ： 饱和度调节。

lParam 的低字为饱和度调节值，范围 0~255。

对于 YUV/YPbPr 信号，lParam 的高字=0 表示调节的是 V/Pr 分量；

=1 表示调节 U/Pb 分量。只对彩色复合视频输入有效，如 C20A,C30A 卡可以设置此参数，RGB 输入无此设置。

*VIDEO_GAINADJUST(18) ： 增益调节。lParam 为增益调节值，有的卡的范围 是 0~255，有的只有 0~3。 C30N，C30，C80A，C20A，USB20A 等卡有增益调节功能。

*VIDEO_PHASEADJUST (20) ： 采集相位调节，取值为 0~256，一般不用设置。

*VIDEO_RGBFORMAT(6) ： 设置视频 VIDEO 的 RGB 格式，lParam 为格式码，常用码有 FORM_RGB888（24 位）、FORM_RGB565（16 位）、FORM_GRAY8（8 位黑白），详情参见 OKAPI32.H 中的宏定义。此设置只可设置成当前卡支持的格式。当返回时，除 lParam 的低字为格式码，另外 lParam 的高字还放有比特数 / 象元（BitCount），如不支持要设置的格式会返回 -1。

*VIDEO_TVSTANDARD(7) ： 视频输入制式设置。

lParam = 0 为 PAL 制。

lParam = 1 为 NTSC 制。

lParam = 2 为非标准。

注意，当切换成某一标准制式（PAL 或 NTSC）时， 系统会自动设置起始偏移、行采样象元和扫描行数及有效区大小为标准值。

*VIDEO_SIGNALTYPE(8) ：

lParam 的低字：设置视频输入信号类型。

= 0 为逐行；

= 1 为隔行；

= 2 为数字卡的单象元

= 3 为数字卡的双象元；

= 4 是反序双象元；

= 5 当前位宽对应 3 个像素（或者是 bayer 格式）；

= 6 当前位宽对应 4 个像素。

lParam 的高字：设置相机和采集卡的外触发模式。

= 0 为相机连续信号模式，采集卡不开槽；

= 1 为相机沿触发模式，采集设置为开槽；

=2 为脉宽触发模式；

=3 为外同步连续模式；

=4 为触发后保持，在该模式下，触发信号给摄像头后，摄像头曝光产生一帧图像保存在相机的的缓存中，当程序发单帧采命令时，如果有新触发产生的图像，会把新的图像传到计算机端，如果触发的数据已经被读取过了，就不会读到数据。用 okGetSignalParam 的 SIGNAL_CAPFINISHED 来判断究竟有没有数据传回来，有的话函数返回 1，否则返回 0。

=5 为触发后保持，该模式和 4 相似，但是读取数据的时候一直可以读取到之前的数据，不管数据是否曾经被读取过。读取新一帧的数据和已经读过的数据可以用 okGetSignalParam 的 SIGNAL_CAPFINISHED 来判断，新的数据的话函数返回 1，否则返回 0。

当 lParam = -2 时， 获得 lParam 的低字的功能当前卡是否支持，如果返回值为-1 表示不支持。

当 lParam = -3 时， 获得 lParam 的高字的功能当前卡是否支持，如果返回值为-1 表示

不支持。

*VIDEO_SYNCSIGCHAN(10) ： 同步信号所在通道。
lParam 的低字= 0 ： 在红色通道（RED）；
= 1 ： 在绿色通道（GREEN）；
=2 ： 在蓝色通道（BLUE）；
=3 ： 在复合同步通道（SYNC）；
= 4 ： 在行、场分离同步通道。对于 RGB 信号，同步通道一般是=1，3，或 4。
lParam 的高字指定同步通道在哪一个视频输入源上，= 0x000：RGB 输入 1；
= 0x001：RGB 输入 2;
= 0x100 ： 复合视频输入 1；
= 0x101 ： 复合视频输入 2 。
当是复合视频输入时，lParam 的低字即无意义了。当设置了视频输入源路（VIDEO_SOURCECHAN）后，应通过此项设置其同步信号所在通道， 一般同步信号应与视频输入源路是同路，但有的卡也可以不是。

*VIDEO_IMAGESOURCE（36）设置摄像头工作模式。
lParam 的低字节 BYTE0 =0(默认): 实时拍摄(连续)；
BYTE0 =1: 末帧冻结 ；
BYTE0 =2: 黑图；
BYTE0 =3 白图；
BYTE0 =4: 垂直的条带图像；
BYTE0 =5: 垂直动态条带图像。
lParam 的 byte1 选择摄像头当前选用的通道（以 0 为起始）； b16=1：测试光源开， b17：图像上叠加直方图。

*VIDEO_ AUXMONCHANN (11) ：lParam 的低字节 BYTE0 是设置 MC30 卡输出到辅助监视器时用哪一路的输入源作为视频源；BYTE1(b8~15)是 LV20G 卡专用；lParam 的高字用来选择输出视频的通道，（当硬件能够输出的通道不止一个的时候，比如 CCU）。

*VIDEO_RECTSHIFT(9)：视频输入信号有效区域起始位置。
lParam 的低字为水平（X）偏移。
lParam 的高字为垂直（Y）偏移。
对于不同的卡、不同的信号源，这个参数有所不同。对于标准信号源，建议用户使用系统缺省值，而不要改变它。强烈建议用户不要再用此项设置，而要用
VIDEO_RECTSHIFTEX(19) 来设置偏移值。

*VIDEO_RECTSHIFTEX(19) ： 视频输入信号有效区域通用起始位置。
lParam 的低字为水平（X）偏移。
lParam 的高字为垂直（Y）偏移。
对于相同的信号源，各 OK 卡的这个参数应基本一致。对于标准 PAL 信号源， 缺省值都为（168，38）。建议用户不要再用 VIDEO_RECTSHIFT(9)来设置偏移值。

*VIDEO_AVAILRECTSIZE(12) ： 视频输入信号有效区域大小。
lParam 的低字为水平方向（X）的象元数 / 扫描行。
lParam 的高字为垂直方向（Y）行数 / 帧。

*VIDEO_FREQSEG(13): 视频输入信号的频段范围。
当 lParam 的高字 = 0 时， lParam 的低字 = 0 ： 低频段 (7.5~15MHz)；
= 1 ： 中频段 (15~30)；
=2 ： 高频段 (30~60)；

= 3 ： 超高频段 (60~120)；
= 4 ： 甚高频段（>120）。

当 lParam 的高字= 1 时， lParam 的低字代表要设置的以 MHz 为单位的频率。无论哪种设置方式，返回的都是频段值。

*VIDEO_MISCCONTROL(16) ： 设置各种控制位。

bit0:色调开关；
bit1:对比度开关 (对于 c20, c30 卡起效) ；
bit2:自动增益控制的开关；
bit3:gamma 矫正；
bit4:强制 MC10A 等卡采集成黑白的图像，可能黑白图像的效果会更好；
bit5:快速模式（只针对 C30B 卡)；
bit6: 1: 兼容高 bit 模式 (比如 12bit...), 0: 兼容 10bit 模式对于 LVDS 数字输入；
bit7: 1: 当输出和输入不同的时候，关闭自动缩放模式；
bit8: 1:VTR 磁带录像机模式，0:自动模式；
bit9: 1: 关闭语言选择功能, 0:打开；
bit10: 1: 输出视频的时候反转灰度；
bit11: 1: 打开电子圆功能；
bit12: 1: 打开可视测量的区域；
bit13:1:强制 MC10A 卡采集成彩色图像，即使信号中的色彩分量强度很低。

*VIDEO_ENABLEGRAPHS（17）允许 GRAPH 目标体。

lParam 的低字节（Byte0）的 bit0: =0 禁止采集的 GRAPH.；=1：允许采集的 GRAPH；
bit1: =0，禁止输出的 GRAPH；=1：允许输出的 GRAPH。
lParam 的次低字节（Byte1）： 选择 GRAPH 的帧号，0、2 对应输入，1、3 对应输出。

设置摄像头相关参数：

*VIDEO_SETBADPTNUM（25）设置坏点的个数。

lParam 的低字的值表示坏点的个数（以 1 为基准）。
lParam 的高字：=1 表示消除坏点；= 0 不消除。

*VIDEO_SETBADPTPOS（26）设置坏点的位置。

lParam 的低字是坏点的 x 坐标，
lParam 的高字是坏点的 y 坐标。

*VIDEO_TAGFRMCOUNT（27）帧数计数功能设置，用于带有硬件压缩 avi 文件功能的卡的设置，用户一般不需要设置。

lParam 的低字=1: 打开帧数计数功能，并把帧号显示在每一帧的图像上。
=0 关闭该功能。
lParam 的高字=1 重置帧数计数器。

*VIDEO_SETWATCHDOG（28）设置看门狗。

lParam 的低字= 0:关闭看门狗；
>0:开启，并将此值（1~255）设置为定时器的时间，以秒为单位。

*VIDEO_SETDATETIME（29）设置并获得硬件上的保留的日期和时间。lParam 必须是指向 tm 结构体（包括日期和时间）的指针，如果 lParam =0，将提取当前的系统时间和日期，用户可以通过设置 tm.tm_year=-1 来得到当前硬件上的时间、日期。

*VIDEO_RESETSYSTEM（30）更新系统时间。

bits0=1 从新启动系统定时器。
b1=1: 清除系统闪存（flash）中的内容。

*VIDEO_SETDELAYLINES（39）线阵摄像头使用。

低位 LOWORD 为矫正系数，0 到 16。

高位 HIWORD =0，矫正起始行为红路。

HIWORD=1， 矫正起始行为蓝路。

设置采集卡输出参数：

*VIDEO_LINEPERIOD(VIDEO_OUTLINEPERIOD)(14) ： 设置回显时图像卡输出行周期，此项只对有输出功能的卡有效。

当 lParam 的高字=0 时，以 0.54 微秒为计数单位，即：lParam 的低字=1 约为 0.54 微秒， lParam 的低字= 2 约为 1.08 微 秒，以此类推，当 lParam 的低字 = 118 大约为 64 微秒，lParam 的低字最大 127，对应大约为 68 微秒。

当 lParam 的高字=1 时，则以 0.1 微秒为计数单位，即 ：lParam 的低字=1 约为 0.1 微秒，lParam 的低字 = 2 约为 0.2 微秒，以此类推。

当 lParam 的高字=2 时，则以 0.01 微秒为计数单位，即 ：lParam 的低字=1 约为 0.01 微秒，lParam 的低字 = 2 约为 0.02 微秒，以此类推。

*VIDEO _FRAMELINES(15) ：设置回显时图像卡输出每帧的行数，此项只对有输出功能的卡有效，因此可用此功能来确定是否支持输出显示功能。

lParam 的 低字 =0 ： 625 行；

=1 ： 1249 行；

=2 ： 525 行；

=3 ： 1049 行 ；

或直接设置行数，如 lParam 的 低字=1249。对于 M40，M60，M30 只有前 4 种行数有效。

对于有输出功能的 OK 二代卡，lParam 的高字=1 时为强制内同步输出，如设 lParam=0XFFFF，则只返回当前设置的 HIWORD 的值， 如不支持则返回 -1。

*VIDEO_OUTSIGNALTYPE (22) ： 视频输出信号类型。

lParam 的 bit0 为设置输出信号绿路带同步；

bit1 为设置为复合同步；

bit2 为设置为隔行信号。例如 ： 隔行复合输出 ： B2~0 应为 111；

逐行复合输出 ： B2~0 应为 011；

VGA 输出 ： B2~0 应为 000。

bit3 输出视频的水平同步极性是负极性；

bit4~ bit5 为设置垂直方向缩小一倍或扩大一倍：

=0 为自动设置（当 VIDEO_FRAMELINES 的值和 CAPTURE_VERTLINES 的值相差大于 1/3 的时候），

=1 为强制缩小一半

=2 强制放大一倍

=3 强制不缩放，按原始比例输出。此功能目前仅 M50B，M30B 等黑白卡及 CAMERALINK 和 LVDS 带独立输出的数字卡支持。

bit6 输出视频的垂直同步极性为负极性；

lParam 的高字目前针对 CPC50B、PR43A、CPC43C 卡，bit16~bit19:

bit16=1: 关闭输出通道 0；

bit17=1: 关闭输出通道 1 以此类推，（CPC50B、 PR43A 只有两个输出通道，CPC43C 有 4 个输出通道）。

* VIDEO_OUTHORZPIX (23) ： 视频输出（到 MONITOR）水平象元数。值越大输出显示

的图越瘦，值越小输出显示的图越宽。此功能目前仅 M50B，M30B 等黑白卡及 CAMERALINK 和 LVDS 带独立输出的数字卡支持。

*VIDEO_OUTXYOFFSET(24) ：视频输出（到 MONITOR）水平和垂直起始值。
lParam 的低字为水平起始值；
lParam 的高字为垂直起始值。

*VIDEO_OUTHVSYNCWID（41） ： 设置输出水平同步（以像素点为单位）和垂直同步信号的同步宽度（以行为单位），
lParam 的低字为水平同步的值；
lParam 的高字为垂直同步的值。

*VIDEO_OUTWINDOWLEVEL（40） ：设置数据变换窗的窗宽和窗位，对于采集灰度位深大于 8bit 的卡才有用。lParam 的低字是变换窗的中值，lParam 的高字是窗宽，当高字为 0 的时候该功能关闭。

*VIDEO_OUTMODESET(38) ：设置输出信号的模式，目前针对 CPC41C。
lParam 的低字 = 0 ：不输出；
= 1 ：输出暂停；
= 2 ：输出恢复；
= 3 ：当 lParam 的高字>0 时，输出的图像快进；当 lParam 的高字<0 时，输出的图像快退；
= 4 ：跳转到指定帧的图像，指定的帧号为 lParam 的高字中的数值。
= 5 ：基于当前图像帧号的跳转，跳转的帧数由 lParam 的高字来决定，该值可以为负数。

*VIDEO_PARTIALMODE（31）：设置摄像头的局部模式（千兆网摄像头适用）。
0(默认): 正常（全局）模式；
非 0: 局部模式，当是局部模式的时候 lParam 的低字表示局部起始行， lParam 的高字表示局部结束行。

设置一些程序的 rect：
*VIDEO_ELLIPSEOFFSET（32）：消隐椭圆（电子圆）的位置偏移值（即沿椭圆外切的矩形的左上角坐标）。
lParam 的低字是水平（X）偏移量。
lParam 的高字是垂直（Y）偏移量。

*VIDEO_ELLIPSESIZE（33）：消隐椭圆（电子圆）的大小。
lParam 的低字是椭圆水平（X）方向最大宽度。
lParam 的高字是椭圆垂直（Y）方向最大高度。

*VIDEO_MEASUREOFFSET（34）：针对 IBS，测量范围的起始偏移。
lParam 的低字是水平（X）偏移量。
lParam 的高字是垂直（Y）偏移量。
lParam=-1 时为读取相机当前设置的测量区域的起始偏移，返回值的低字是水平（X）偏移量，返回值的高字是垂直（Y）偏移量。

*VIDEO_MEASURESIZE（35）：针对 IBS，设置或读取测量区域的宽高。
lParam 的低字是测量区域的宽度。
lParam 的高字是测量区域的高度。

lParam=-1 时为读取相机当前设置的测量区域大小，返回值的低字是宽度，返回值的高字是高度。

*VIDEO_CURRHISTOGRAM（37）：获得指定范围内图像的直方图 (默认是全部图像，有可能有多个直方图，比如 RGB 格式的图像)，返回当前位深的灰度范围 (如果是 gray 8 的话是 256)。

lParam=1： 允许计算直方图;
=0：不允许;
=-1： 只返回灰度范围；
lParam 还可以是指向保存直方图数据地址的 INT32 指针[n1]。

VIDEO_GRAYLEVELPARM（42）：获得当前关于灰度级的参数到 lParam，现在的参数顺序为：平均值，最大值，最小值。当数据是 RGB 格式的时候，顺序为：红色的平均值，绿色的平均值，蓝色的平均值，红色的最大值，绿色的最大值。。。以此类推。

lParam ：对应 wParam 的参数值。意义详见上述。对于对于有些 lParam，他的的高字和低字对应了不同的参数设置的，应该先把整个参数读出，然后只改变需要修改的高字或低字部分然后再进行参数设置。

返回值 ：如果指定的项目或参数不被支持，返回 -1 ；如果失败则返回 -2。如果成功则返回该项目之前的参数值。

注意 ：如果 lParam = -1（GETCURRPARAM），则仅返回该项目当前的参数值。通过此函数设置的视频输入参数，在应用程序调用 okCloseBoard 之后自动存入该卡的初始化配置文件中，在应用程序调用 okOpenBoard 之后，此函数将从对应卡的初始化配置文件中取出视频输入参数来进行图像卡的初始视频输入参数设置，已设置的参数如果没有改变将一直有效，所以当需要不同的设置时就要重新设置相应的参数。

相关函数 ：okSetCaptureParam, okSetTargetRect，okCloseBoard, okOpenBoard。

MLONG WINAPI okSetCaptureParam(HANDLE hBoard,INT32 wParam, MLONG lParam);

功能 ：设置并获得采集控制参数（如采集格式、方式等）。

参数 ：hBoard ：输入卡句柄。

wParam：指定设置采集项目，所支持的项目如下（可参见 OKAPI32.H 中的宏定义）。

* CAPTURE _RESETALL(0) ：重置所有采集项的参数值为系统缺省值，lParam 为 0。

* CAPTURE_INTERVAL(1) ：

lParam 的低字是设置采集帧间隔（设置了之后意味着每 LOWORD+1 帧会采集一帧图像），= 0 为逐帧，=1 为隔一帧，以此类推。

lParam 的高字是表示每 10 秒采集多少帧 (帧率)，只要设置了就起效，但是只有 lParam 的高字= 0 的时候 lParam 的低字设置的值才起效。

* CAPTURE_CLIPMODE(2) ：

当设置的视频源窗与采集目标窗口大小不同时

lParam 的低字用来设置采集裁剪方式，= 0 为缩放方式；
= 1 为中心化方式，即调整源窗到视频源有效区中心；
= 2 为左上角对齐方式，即固定源窗左上角位置而调整右下角位置。

lParam 的高字：如果设置的采集方式是奇偶场交叉采集的方式，当 lParam 的高字=0x40,

函数返回值也是 =0x40, 支持数据场扩展。
当 lParam 的高字=0x80, 函数返回值也是=0x80, 支持图像放大(C30N) 。
当lParam 的高字=0x100, 函数返回值也是=0x100, 支持采集时对图像进行 x 方向缩小 （整体缩小) 。
当lParam 的高字=0x200, 函数返回值也是=0x200, 支持采集时对图像进行 y 方向缩小（整体缩小)。

* CAPTURE_SCRRGBFORMAT(3) ：设置屏幕 （SCREEN） 的 RGB 格式，lParam 为格式码。常用码有 FORM_RGB888（24 位），FORM_RGB565（16 位），FORM_GRAY8（8 位黑白），详情参见 OKAPI32.H 中的宏定义。返回时，lParam 的低字为格式码，lParam 的高字为比特数 / 象元（BitCount/Pixel）。该项不需要用户自行设置，因为系统内部已根据当前 VGA 的模式设置好了，但用户可以通过本函数获得当前屏幕 SCREEN 的格式。

* CAPTURE_BUFRGBFORMAT(4)：设置缓存（ BUFFER ）的 RGB 格式，lParam 为格式码。常用码 有 FORM_RGB888（24 位 ），FORM_RGB565（16 位 ），FORM_GRAY8（8 位黑白），详情参见 OKAPI32.H 中的宏定义。可以设置成任意格式而不论当前卡是否支持。返回时，除 lParam 的低字为格式码外，还有 lParam 的高字为比特数 / 象元（BitCount/Pixel）。

* CAPTURE_FRMRGBFORMAT(5) ：设置帧存（ FRAME） 的 RGB 格式，lParam 为格式码。常用码有 FORM_RGB888（24 位），FORM_RGB565（16 位），FORM_GRAY8（8 位黑白），详情参见 OKAPI32.H 中的宏定义。返回时，lParam 的低字为格式码，lParam 的高字为比特数 / 象元（BitCount/Pixel）。

* CAPTURE_BUFBLOCKSIZE(6) ：设置序列缓存每帧的固定大小，lParam 的低字为缓存宽度 WIDTH（象元数 / 行），lParam 的高字为缓存高度 HEIGHT（象元数 / 列）。要读写缓存图像的大小应通过调用函数 okSetTargetRect(hBoard,BUFFE,&rect)，来设置读写缓存的窗口 rect 区域。如果设置的 rect.right，或 rect.bottom 大于对应的缓存宽度 WIDTH 或缓存高度 HEIGHT， 则程序会自动使 WIDTH= rect.right 或 HEIGHT= rect.bottom。
注意 ：如设置 lParam=0（缺省值既为 0），则缓存每帧的大小随由通过调用函数 okSetTargetRect(hBoard,BUFFER,&rect) 来设置的缓存窗口的大小变化而变化，即 WIDTH=rect.right，HEIGHT=rect.bottom，所以一般此项不用设置。

*CAPTURE_HARDMIRROR(7)：
lParam 的 低字设置采集时是否作镜像变换，最低位（bit0） 代表水平（X）方向，第二位（bit1）代表垂直（Y）方向。
lParam 的 低字=0 ： 无镜像；
lParam 的 低字=1 ： X 镜像；
lParam 的 低字=2 ： Y 镜像；
lParam 的 低字=3 ： X、Y 都镜像。
lParam =-2 的时候，如果函数返回-1，表示该卡不支持输出图像镜像和采集图像镜像独立；
lParam 的 bit7=1 表示设置采集镜像和回放输出镜像是独立的。
当输入和输出独立的时候，bit4:是输出的 x 镜像, bit5 是输出的 y 镜像。
lParam 的高字是特别旋转： bit16~bit17: 特别角度旋转。
bit16~bit17=1: 向左旋转 90 度；
=2: 向右旋转 90 度；
=3: 旋转 180 度。
lParam= -3, 判断当前硬件是否支持特殊角度的旋转，如函数返回 -1 为不支持。

*CAPTURE_ANGLEROTATE（34）：任意角度旋转。

lParam 的低字：bit0~bit11： 要旋转的角度， =1 代表 1/10 度，以此类推；
lParam = -2 的时候，如果函数返回-1： 表示不支持输出图像和采集图像分别独立做旋转。
lParam 的高字：bit31=1 表示设置采集和回显输出的旋转相互独立，这时 bit16~bit27：表示输出图像要旋转的角度，=1 代表 1/10 度，以此类推。如果 CAPTURE_HARDMIRROR 的 lParam 的高字不是 0，这个设置将会被忽略。

* CAPTURE_VIASHARPEN(8) ：lParam 的低字设置采集时通过锐化增强滤波的滤波系数。值的范围为 0~15。目前有 C20N，C30N，C20A，USB20A 支持此项功能。对于 AM1566 相机有效取值为 0-7，输入 7 以上的值实际都是 7。

* CAPTURE_VIAKFILTER(9) ：去除噪声的控制。
lParam 的低字设置采集的递归滤波和平均滤波系数。bit15=0 时使用递归滤波；bit15=1 时使用平均滤波。低字（LOWORD）的低字节（LOBYTE）为递归滤波系数的数值；低字（LOWORD）的高字节（HIBYTE）为平均滤波使用的帧数。
lParam 的高字设置平滑滤波系数，值为 0~15。对于 AM1566 相机：设置硬件实现的帧内滤波算法：=0 为关闭；=1 为均值滤波；=2 是高斯滤波。
当 lParam=-2 时，可以获得当前设置的递归滤波系数，如果返回 -1 为不支持。
当 lParam=-3 时，可以获得当前设置的平滑滤波系数，如果返回 -1 为不支持。
当 lParam=-4 时，可以获得当前设置的平均滤波系数，如果 bit7=1 表示使用平均滤波功能，如果返回 -1 为不支持。

B31…b24	b23…b16	b15	b14…b8	b7…b0
保留	平滑滤波系数	选择递归或平均滤波功能	平均滤波系数	递归滤波系数

获得当前设置的平均滤波系数

```
(okSetCaptureParam(hBoard, CAPTURE_VIAKFILTER, -4))&0x7f
```

* CAPTURE_SAMPLEFIELD(10) ：设置帧采集方式.
lParam =0 逐场采集方式，即按场顺序存放；
lParam =1 逐帧采集方式，即按帧（由两场隔行存放构成） 顺序存放；
lParam =2 按场采集隔行存放方式 ；
lParam=3 按上下场存放；
lParam=4 //in field but interlaced to one frame, from even
lParam=5 //in field just top field
注意，当在某一标准制式（PAL 或 NTSC）下，设置此项参数时，系统会自动设置行采样象元、扫描行数和有效区大小为标准值。

* CAPTURE_HORZPIXELS(11) ：设置水平（X）方向总采集象元数，即总象元数 / 扫描行（含行消隐期）。只有支持可变行采集频率的卡可设置此项。

* CAPTURE_ VERTLINES(12) ：设置垂直（Y）方向信号源的扫描线数，即总行数 / 帧（含消隐行）。对于标准制式（PAL 和 NTSC）此项由系统自动设置； 对于非标准制式，此项则需要由用户根据所接信号源的实际扫描线数来设置此项的参数值。

*CAPTURE_SCROFFSETVID（21）：设置图像在屏幕显示时的偏移，针对图像在屏幕显示的时需要滚动条的时候，如果想要用实时显的方式显示图像，当图像分辨率高，超出屏幕的显示分辨率需要滚动条的时候，可以通过设置这个参数实现显示不同的区域。该参数影响采集到 SCREEN 和 okConvertRect 函数源或目标体是 SCREEN 的时候。lParam 的低字是 X 的偏移值，lParam 的高字是 Y 的偏移值。

* CAPTURE_ARITHMODE(13) ：设置采集 / 回显时实时运算方式，目前只有 M60，M60A

和 CL60A，LV60A 等支持此项设置。

lParam 的 低字为 运算方式， =0 ： 无运算；
=1 ： 源－帧存；
=2 ： 帧存－源；
=3 ： 源＋帧存；
=4 ： 带符号的源－帧存；
=5 ： 带符号的帧存－源。

M60 只支持前 4 项设置。

lParam 的 高字 为运算结果的输出目标，=0 ： 只对输出显示；
=1 ： 只对采集的数据；
=2 ： 同时对两者。

* CAPTURE_TO8BITMODE(14) ：设置高比特转换成 8 bit 的转换方式。对于支持 高 bit（如 10 bit）的采集卡（M30B，M70B 等），当采集的目的体设置为 8 bit 时， 其 10bit 的数据转换成 8bit 数据时的转换方式。

当 lParam 的高字即 HIWORD(lParam) 为 0 时，采用线性变换方式，即把 10 位值域 0~1023 线性缩小（即除 4）到 0~255。

当 lParam 的高字即 HIWORD(lParam) 为非零时，采用截取灰度段（256 个值） 的方式，lParam 的低字即 LOWORD(lParam) 为截取的起始值偏移（offset）。即把 offset~offset ＋ 255 变换到 0~255，低于 offset 的置为 0，高于 offset ＋ 255 的置为 255。但采集的目的体不为 8 bit 时，此设置不起作用。目前有 M30B，M70B 支持此项设置。

* CAPTURE_SELRGBDATA(37) ： lParam 的低字即 LOWORD(lParam) 为设置信号源的 RGB 通道中的某一个通道数据作为采集灰度图像的数据源（**目前针对 CPC61C，且只有在缓存格式设置为 GRAY8 的时候才能起效**）。 =0：取三通道的平均值；=1：红色通道的数据；=2：绿色通道的数据；=3：蓝色通道的数据。

* CAPTURE_SEQCAPWAIT(15) ： 设置采集函数返回方式和回调是否立即返回。

1，设置调用采集（包括间接采集）或回显函数为单帧或序列方式时是否等待结束才返回。
Bit0=0 ： 不等结束立即返回方式。
Bit0=1 ： 等结束后才返回方式。
注意，如果是实时采或循环序列（采集 / 回显）状态，则只能是立即返回方式。

2，设置采集过程中，在回调用户编写的回调函数之前是否等待采集结束。
Bit1=0，不等采集结束就调用回调函数，这是一种并行工作方式，即硬件采集过程与用户回调函数所做工作同时进行 ；
Bit1=1，等采集结束后再调用回调函数。这是串行工作方式，即硬件采集过程完全结束后，才会去执行用户回调函数所做的工作。当采集数据量很大时（如用 RGB20 采集逐行信号），采集需占用 PCI 大量时间，如这时回调函数所做工作也需占用较多 PCI 时间（如涉及显卡（VGA）的操作），就会干扰采集的正常进行，这时就需要通过本设置使回调函数在采集完成后再进行（不过这时就难以实现逐帧并行采集处理了），或回调函数什么也不做。

注意：此设置将控制函数 okCaptureTo、okCaptureByBuffer 的采集等待方式和 okPlaybackFrom、okPlaybackByBuffer 的回显等待方式及是否采集结束才调用回调函数。系统缺省设置为不等待方式。

* CAPTURE_TRIGCAPTURE(17) ： 设置外触发硬件控制采集的方式。

1、 lParam 的低字=0：立即采集，即正常方式；
=1：等到外触发后采集 1 场图像，以此类推，目前支持的最大帧数是 15。

2、 当 lParam 的低字不等于零时，lParam 的高字中的低字节 =0，外触发到来后立即采集 ；

=1 延迟 1 幅图像再采集，以此类推；

lParam 的高字中的高字节表示信号源是触发状态下驱动查询等待时间，=0 为 10ms（默认），可以为 1~255，对应 1~255ms，在序列采集的时候起效，表示驱动等待多久查询一次有没有新的图像来，如果时间间隔长有可能会丢帧，时间间隔短 CPU 占用率会高。

*CAPTURE_TURNCHANNELS(18)：设置序列采集的同时进行通道切换。

bit0~6（共 7 位） 表示要切换的通道数（最大值为 127）；

bit8~23(共 16 位)分别标志 0~15 通道，该位为 1 表示采集此通道，为 0 是屏蔽此通道；

bit7=1 采集的时候，虽然不采集其中的一些通道，但是在缓存中仍保持这些通道的那一帧采集数据的位置，其内容为空；

bit28~31（共 4 位）：切换间隔，表示间隔多少帧切换一次通道。这个设置主要针对 OK 多路卡，如 MC10A 等切换速度比较快的卡，而且要切换的信号源之间必须是同步的，用中断的采集方式可以不丢帧。

*CAPTURE_MISCCONTROL（16）：设置采集控制位。

bit0：=1 强制 okCapturByBuffer 或 okGetSeqCapture 函数用中断控制的方式采集；

bit1：= 1 当使用 okCaptureSequence 、okCapturByBuffer 、okGetSeqCapture 函数的时候强制其使用中断采集的方式，并且在回调的时候顺序帧给用户帧号，使用户可以不丢帧的处理（one by one）数据。当用户处理数据的时间有时过长，超过一帧图像的采集时间，又想要不丢帧的处理所有数据的时候可以设置该位为 1，这样回调函数会每次给用户顺序帧的帧号（默认是给用户最新采集到的一帧的帧号）；

bit2：= 1 强制获得完整的一帧帧的图像数据，只针对 USB20A。（这样可以保证获得每一幅图像都是完整的，但是会导致帧率下降）；

bit3：针对 12 位的采集卡，如 LV20B 等，= 1 采集全部的 12 bits 数据， =0：只采集 10 bits 数据；

bit4：对于 LVDS 系列的采集卡，=1: 水平点数从行同步的上升沿开始计算，= 0：从下降沿开始计算（默认）；

bit5：对于 LVDS 系列的采集卡，=1: 垂直行数从同步的上升沿开始计算，= 0: 从下降沿开始计算（默认）；

bit6：= 1：USB 摄像头启用局部模式，=0：全局模式（默认）；

bit7：=1:强制 okCaptureTo 和 okCaptureActive 转为 okCapturebyBuffer，注意：这一位只有它在高字中的响应位（bit23）是 1 的时候才能设置；

bit10：摄像头的数据是否做 LUT 或 gamma 校正，=1，进行校正。

bit11: =1：使用 8 帧缓存作为 okCaptureByBuffer 函数采集的缓存，默认为 2 帧。

bit12: =1：设置 okCaptureByBuffer 函数保存文件的时候采用不丢帧的并行方式，这样会启动两个线程，一个负责采集回调，一个用于保存图像，加快文件保存的速度。默认为 0，是串行的工作方式。

*CAPTURE_SETLATENCY（24）：设置 PCI 占用期，系统默认是 0xf8, 如果是多卡同时工作的话最好设置成 0x18 。

*CAPTURE_SLEEPTIMES（25）：设置所有序列采集时的空闲时间和最长等待时间。

lParam 的低字: 释放 CPU 级别，= 0: 无效；

=1：系统自动（默认）；

>=2 特殊级别。

lParam 的高字: 最大等待时间（以 10 毫秒为单位），设置了这个参数后，回调的时候在最长等待时间到达之前，如果卡没有采集到一帧图像，可以不进入回调。= 0:系统自动（默认），最大值为 0xFFFF，约为 10 分钟。

*CAPTURE_EXPOSETIME(20)：设置摄像头曝光时间，以微秒（us）为单位。目前 OK 系

列的 USB 摄像头、千兆网摄像头、智能摄像头支持此项功能。lParam=-2: 返回最大有效曝光时间。

*CAPTURE_DELAYEXPOSE（27）：设置摄像头延迟曝光时间，以微秒（us）为单位，摄像头从基准时间延迟 lParam 微秒再开始曝光，常用于摄像头工作在外触发模式下。

*CAPTURE_FRAMERATE（22）：设置摄像头的帧率或行频。

Lparam=-3 时返回面阵摄像头支持的最大帧频或线阵摄像头支持的最大行频。

对于面阵摄像头：
lparam 的高字（HIWORD）用来设置摄像头硬件输出帧率。
lparam 的低字（LOWORD）用来设置摄像头软件采集帧率。设置软件采集帧率的数值需要要大于零且小于硬件输出帧率才有效。
lparam=-2 时返回该摄像头支持的硬件输出帧频共有几档。返回值=1 时：表示只支持全帧频；返回值>1 时：表示除了支持全帧频之外，还支持 1/2 帧频或 1/3 帧频等。

对于线阵摄像头：当 lparam 的高字中的高字节（HIBYTE）=0x80 时，lparam 的低字（LOWORD）用来设置线阵摄像头的行频。

*CAPTURE_SETTEMPER（23）：设置摄像头的工作温度，当 lParam= -2 的时候是得到当前摄像头的实时温度。如果设置的温度值很低，超过了制冷的极限温度（低于环境温度 40 度），摄像头的制冷功能将一直处于开启状态。目前该设置只针对制冷摄像头（如 AM9010 等），主要用于读取当前温度。

*CAPTURE_WHITEBALANCE（26）：设置摄像头白平衡功能。
lParam 的低字=0:关闭该功能；
=1：设置的时候自动平衡一次；
=2：实时进行平衡。
lParam 的高字=0：默认（自动）；
=1：日光环境下；
=2：荧光灯下；
=3 白炽灯下 。
目前只支持 USB 接口的彩色摄像头。

*CAPTURE_SETBINNINGMODE（28）：设置摄像头的 binning 模式。
lParam 的低字：=0：关闭 binning 模式；
不等于 0 时 binning 起效。
当 BYTE1=0 的时候，BYTE0=1: 1x1, =2: 2x2, =3: 3x3, ...；
当 BYTE1!=0 的时候，表示 BYTE0 是 X 方向 binning 的值，BYTE1 是 Y 方向的 binning 值。
lParam 的高字：设置软件 binning 方式（相同帧率）。默认为 0，表示或得全部图像。
当 BYTE3=0 的时候，BYTE0=1: 也是全图模式； =2: 1/2x1/2；…；
当 BYTE3!=0 的时候，表示 BYTE2 是 X 方向 binning 的值，BYTE3 是 Y 方向的 binning 值。

*CAPTURE_ENBCAMFUNC（29）：设置摄像头的特殊功能是否开启，=0：关闭； =1： 开启。
bit0：自动增益模式，如果该功能开启，那么 BYTE2（非 0）用来设置自动增益的目标灰度值(共 8bits)。
bit1： 灰度拉伸。
bit2： 计算直方图数据。

bit3：去除噪声点功能，如果该功能开启，那么 BYTE3（非 0）用来设置目标点和周围的点的灰度值相差多少会被认为是噪点而对其进行处理。

*CAPTURE_GAMMAFACTOR（30）：对（部分）摄像头的 gamma 系数进行设置，设置值是 gamma 系数的 100 倍。如果想要设置的系数是 0.45，那么应该设置的值为 45，对应的摄像头会按照 gamma 系数是 0.45 来对原始数据做校正。

*CAPTURE_FLATFIELDING （36）：平场校正功能（使摄像头或采集卡成像亮度更均匀）。
lParam 的低字=1：采集暗场图像到 BUFFER 0；
= 2：采集亮场图像到 BUFFER 1；
= 3： 生成正确的模板并写入硬件。
lParam 的高字：=1 开启平场校正功能； =0x100：关闭平场校正功能。

*CAPTURE_SETCOMPPARAM（31）：设置智能摄像头压缩参数。
lParam 的低字： 设置 jpg 压缩的质量因子(1~100)，决定图像的压缩比，该值越大压缩比越小，图像保质越好，0 是默认参数，等价于值 50。
lParam 的高字： 设置压缩格式， =1 为 JPG， =2 为 PNG，=3 为 TIFF…目前支持的是 jpg。

*CAPTURE_PACKETINTERVAL（32）：设置网络设备传送数据包的间隔，用于控制传输速度。此项设置和 cpu 性能、网络状况、网卡性能有关。两个数据包间隔过小，可能导致丢帧；两个数据包间隔过大，可能导致摄像头帧速变慢。单位：10 微秒。
当 lParam = 0 时是自动模式（默认）， 通常状态下最大值大约是 6000000(60Seconds)。
当 lParam = -2 时，返回相机能支持的最大值。

*CAPTURE_PACKETSIZE（33）：设置网络设备传送数据的数据包大小，以字节为单位，最小值是 8 字节，最大值是 2048 字节，默认为 1500 字节，一般情况下不需要特别改变，如果网卡支持巨帧（jumbo frame）则可根据实际情况设置 UDP 数据包大小。

*CAPTURE_WAITASKRESEND（35）：设置网络设备每隔多少包后检查一次有没有丢包，如果摄像头支持重发功能的话会自动重发。=0： 不重发；其他值时为检查时间隔的包数。

lParam ：对应 wParam 的参数值。意义详见上述。对于有些 lParam，它的高字和低字对应了不同的参数设置，应该先把整个参数读出，然后只改变需要修改的高字或低字部分然后再进行参数设置。
返回值 ：如果指定的项目或参数不被支持，返回 -1 ；如果失败则返回 -2。如果成功 则返回该项目之前的参数值。
说明 ：如果 lParam = -1（GETCURRPARAM），则仅返回该项目当前的参数值。 通过此函数设置的采集参数，在应用程序调用 okCloseBoard 之后自动存入该卡的初始化配置文件中，在应用程序调用 okOpenBoard 之后，此函数将从对应卡 的初始化配置文件中取出采集参数来进行图像卡的初始视频采集参数设置。已设置的参数如果没有改变将一直有效。所以当需要不同的设置时就要重新设置相应的参数。
相关函数 ：okSetVideoParam, okSetTargetRect, okCloseBoard, okOpenBoard。

MLONG WINAPI okSetDeviceParam(HANDLE hBoard, INT32 wParam, MLONG lParam);

功能：设置、获得网络设备的调节参数。
参数：hBoard：相应设备的句柄。
wParam：指定设置项目，所支持的项目如下 （可参见 OKAPI32.H 中的宏定义） ：
DEVICE_SETDEVMEMORY（0）：设置或获得网络设备（目前主要是摄像头）的全部参数。
lParam=-1：将摄像头的记忆体参数导出为摄像头的当前参数；

=0：储存当前摄像头参数到记忆体作为上电后的默认参数（如果不调用此函数，摄像头断电再上电后所有参数将按照摄像头记忆体中的参数进行设置）；

=1: 将摄像头当前参数恢复为出厂设置的参数（不改变记忆体）。

DEVICE_SETIPADDRESS（1）：设置或获得网络设备的 IP 地址。lParam 是 NETCFGPARAM 结构体的指针，具体定义详见 OKAPI32.H 中的宏定义，

DEVICE_GETMACADDRESS（2）获得网络设备的 mac 地址。lParam 是 NETCFGPARAM 结构体的指针，具体定义详见 OKAPI32.H 中的宏定义。

DEVICE_GETACCEPTTHREAD（3）：得到网络设备接收采集图像的线程的句柄。

DEVICE_GETPARAMCHANGED(4)：得到某组参数被设置过并且发生了改变的标志信息，并且调用后会把该标志清除。

返回值的 b0=1 表示用 okSetVideoParam 设置了参数；

b1=1 表示用 okSetCaptureParam 设置了参数；

b2=1 表示用 okSetConvertParam 设置了参数。

lParam：对应 wParam 的参数值。意义详见上述。

返回值：如果指定的项目或参数不被支持，返回 -1 ；如果失败则返回 0；如果成功则返回 1（TRUE）或是预期的参数值。

说明：NETCFGPARAM 结构体的 bRWFlag 为 0 的时候是读当前参数的值，为 1 的时候是写当前参数的值。

相关函数：okSetCaptureParam，okSetVideoParam，okCloseBoard, okOpenBoard，okOpenBoardEx。

（4）设置\获得采集窗口

LPVOID WINAPI okGetTargetInfo(HANDLE hBoard, TARGET target, MLONG lNoFrame, SHORT *wid, SHORT *ht, MLONG *stride);

功能：获得的目标体（BUFFER、SCREEN，FRAME）中指定帧的线性地址及宽高和行跨距。

参数：

hBoard ：输入卡句柄。

target ：指定目标体 (可以是 BUFFER、SCREEN,FRAME)

lNoFrame ：指定在目标体中的帧序号。

Wid ：返回目标体的宽度 (以象元为单位)。

Ht ：返回目标体的高度 (以象元为单位)。

stride ：返回目标体的行跨距 (以字节为单位)。

返回值：如果调用成功，返回目标体当前设置下指定帧的线性地址。否则返回 0（FALSE）。

说明：如用户需要对某目标体进行直接处理，要用此函数来得到指定帧的线性地址及 其宽高和行跨距（前一行与后一行同一列位置之间的字节数）的信息，才可正确读写该目标体的数据。

相关函数：okGetBufferSize，okGetBufferAddr，okSetCaptureParam。

MLONG WINAPI okSetTargetRect(HANDLE hBoard,TARGET target, LPRECT lpTgtRect);

功能：设置采集 / 回显的目标体的窗口（兴趣区 AOI）。视频源目标体（VIDEO）的源窗 口， 与采集目的目标体（VGA 屏 幕 SCREEN， 帧缓存 BUFFER， 帧存体 FRAME，视频显示 MONITOR，要显示图像的用户窗口句柄）的目的窗口均需用此函数来设置。

参数：

hBoard ：输入卡句柄。

target ：要设置的目标体，可以是源目标体 VIDEO，也可以是目的目标体 SCREEN，

BUFFER，FRAME，MONITOR 及 WINDOWS 窗口句柄其中之一。目标体的宏定义参见 OKAPI32.H。窗口句柄的坐标使用的就是用户区坐标，而非绝对坐标，如果从未设置过窗口句柄的坐标或设置使其坐标的 left=right，则采集到窗口句柄时使用与 SCREEN 相同的位置。

lpTgtRect ：要设置目标体的窗口坐标。如果调用成功且其坐标值不完全正确（X 宽度坐标要求 4 字节对齐），将会被调整为正确值。窗口坐标按 WINDOW 习惯，定义左上为闭坐标，右下为开坐标，即设定的图像窗口区域为，以 （left，top）点为起始，宽度 =right-left，高度=bottom- top。

设置 SCREEN 的窗口坐标时，如果是要设定到用户所打开的某一 WINDOWS 窗口，则要把该窗口的坐标转换成屏幕绝对坐标值。设置 WINDOW 窗口句柄的口坐标时，直接使用该窗口句柄的用户区坐标。

注意，所设置目标体窗口的大小可以比当前卡所允许的大，但 如果调用采集函数时，将自动调整窗口为允许的大小。如果想要没 有调用采集函数之前，就保证设置的窗口符合当前卡所允许的大小。 可以通过调用本函数，使 lpTgtRect=NULL，来实现要设置的目标体窗口与本卡的自动匹配。但当目标体是 WINDOW 窗口句柄时除外，当 lpTgtRect=NULL，意味着清除对 WINDOW 窗口句柄的采集坐标设置，而使用对 SCREEN 设置的坐标。

设置视频显示 MONITOR 的窗口位置 ，仅限于支持视频输出且可设置窗口的卡，如 CL40A，LV40A，M50B，M30B 等。

返回值 ：如果调用成功，返回该目标体在当前窗口设置下可以支持的图像帧数，否则返回 0（FALSE）。

说明 ：如果调用前设置 lpTgtRect 结构中的 .right 或 .bottom 为 -1，则意味着要获得该 目标体在本系统中当前的窗口坐标设置，调用返回后，lpTgtRect 被填写为指定目标体的当前窗口坐标值。

注意 ：如果目标体是帧缓存，即 target=BUFFER, 如果从未通过 okSetCaptureParam 设置过 CAPTURE_BUFBLOCKSIZE 或将其设置为 0，则帧缓存每一帧的大小（宽和 高）将随 lpTgtRect 结构变量中的 right 和 bottom 变化而变化，即帧缓存的宽 = lpTgtRect-> right，高 = lpTgtRect-> bottom。

在采集之前，首先应调用此函数设置源窗口和目的窗口（BUFFER 或 SCREEN）。对于可以支持硬件缩小的采集卡，如：C20A，C30A 等，在缩放方式下（通过 okSetCaptureParam 中的 CAPTURE_CLIPMODE 设置），可以通过设置源窗口和目的窗口的大小不同（目的窗口大小只能小于等于源窗口的大小）来实现硬件采集缩小。如不需要缩小，则必须设置源窗口和目的窗口具有同样的大小，如 果是在源窗的中心剪切或左上角剪切方式，可以仅设置目的窗口，这时的源窗 口会由系统根据当前的剪切方式和目的窗口的大小自动设置。对于不支持硬件直接缩小的采集卡，如：M20A，M40A，M60A，M30，M70，RGB10A，RGB20A 等，则只支持中心剪切和左上角剪切两种方式，此时源窗口和目的窗口应设成同样的大小，如果不同则以目标体的大小为准。

在标准制式下，当 VIDEO_TVSTANDARD 设置为 PAL 制（即值 0）时，最大可设置窗口大小为 768×576 ；当设置为 NTSC 制（即值 1）时，最大可设置窗口大小为 640×480。当需要采集高分辨视频信号时，首先要通过调用 okSetVideoParam 设置 VIDEO_TVSTANDARD 的参数为非标准（即值 2），然后再通过调用 okSetCaptureParam 的 CAPTURE_HORZPIXELS 和 CAPTURE_VERTLINES 的参数为相应的足够大的值（大于要采集窗口的大小），才可通过调用本函数设置视频源窗口（VIDEO）及目的窗口为高分辨的窗口大小。

通过此函数设置的目标体窗口坐标，在应用程序调用 okCloseBoard 之后自动存入该卡的当前通道的初始化配置文件中，在应用程序调用 okOpenBoard 之后， 此函数将用对应卡的零号通道初始化配置文件中的目标体窗口坐标参数来进行图像卡的初始窗口坐标设置。

相关函数 ： okSetVideoParam, okSetCaptureParam, okSetToWndRect ， okOpenBoard, okCloseBoard。

INT32 WINAPI okSetToWndRect(HANDLE hBoard, HWND hWnd);

功能 ：设置 WINDOW 窗口句柄 hWnd 所对应的用户区（Client）作为目标体 VGA 屏幕（SCREEN）的采集 / 回显窗口。
参数 ： hBoard ： 输入卡句柄。
hWnd ： 用户 WINDOWS 窗口句柄。
返回值 ： 如果调用成功，返回 1（TRUE），否则返回 0（FALSE）。
说明 ：该函数将读出 hWnd 的用户区（Client）的窗口坐标，然后转换成 VGA 的绝对 坐标值，再调用 okSetTargetRect 来设置 SCREEN 的窗口坐标。
相关函数 ： okSetTargetRect, okSetVideoParam, okSetCaptureParam。

（5）视频采集

所有的采集函数都将按照目标体的指定格式进行采集，如果当前目标体的格式不为当前卡所支持，则会自动将目标体的格式设置成当前卡所支持的格式，而如果目标体（如 SCREEN）的格式为不可改变的，则函数会调用失败。

BOOL WINAPI okCaptureSingle(HANDLE hBoard,TARGET Dest, MLONG lStart);

功能 ：采集视频输入一帧到指定目标体。这里的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME）。可以不等采集结束立即返回，也可以等 采集结束再返回。
参数 ： hBoard ： 输入卡句柄。
Dest ： 要采集到的目标体，可以是 SCREEN，BUFFER 或 FRAME。目标体的宏定义参见 OKAPI32.H。
lStart ： 采集到目标体的起始帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN，该值只能为 0。
返回值 ： 如果调用成功，返回该目标体所支持的最大帧数。如果失败（如由于格式不支持等）返回 0（FALSE）。如果该目标体不被支持则返回 -1。
说明 ：此函数等价于 okCaptureTo(hBoard, Dest, lStart, 1);
相关函数 ： okGetCaptureStauts, okStopCapture, okCaptureByBuffer, okSetTargetRect, okSetVideoParam, okSetCaptureParam, okCaptureByBufferEx，okCaptureTo。

BOOL WINAPI okCaptureActive(HANDLE hBoard,TARGET Dest, MLONG lStart);

功能 ：实时采集视频输入到指定目标体。这里的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME）。
参数 ： hBoard ： 输入卡句柄。
Dest ： 要采集到的目标体，可以是 SCREEN，BUFFER 或 FRAME。目标体的宏定义参见 OKAPI32.H。
lStart ： 采集到目标体的起始帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN，该值只能为 0。
返回值 ： 如果调用成功，返回该目标体所支持的最大帧数。如果失败（如由于格式不支持等）返回 0（FALSE）。如果该目标体不被支持则返回 -1。
说明 ：此函数等价于 okCaptureTo(hBoard, Dest, Start, 0);
相关函数 ： okGetCaptureStauts, okStopCapture, okCaptureByBuffer, okSetTargetRect, okSetVideoParam, okSetCaptureParam, okCaptureTo, okCaptureByBufferEx。

HANDLE WINAPI okCaptureThread(HANDLE hBoard,TARGET

Dest,MLONG lStart, MLONG lNoFrame);

功能 ：采集视频输入到指定目标体。这里的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME）和监视器（MONITOR）。可以单帧采集也可以进行多帧采集，可以不等采集结束立即返回，也可以等采集结束再返回。
参数 ：hBoard ：输入卡句柄。
Dest ：要采集到的目标体，可以是 SCREEN，BUFFER 或 FRAME。目标体的宏定义参见 OKAPI32.H。
lStart ：采集到目标体的起始帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN，该值只能为 0。
lNoFrame ：采集帧数或采集方式。
1，该值如果大于零（>0），即为采到目标体的帧数，此时为序列采集方式。如果 lNoFram 大于目标体的最大帧数（Total），当逐帧采集到目标体的位置超过 Total 时，将重置回到起始（序号 0）位置继续开始逐帧采集，如此按方式 mode(n%total) 采集，直到采完 lParam 帧为止。
2，如果等于零（=0），为连续采集（即实时采）方式，一直采集视频输入到 wParam 指定的位置直到通过调用 okStopCapture 才会停止。 连续采集时无回调支持。
3，如果等于 -1，为循环序列采集方式，即从 wParam 指定的位置开始序列采集，当达到目标体的最大帧数（Total）时将回到起始（序号 0）位置继续开始逐帧采集，如此无限循环下去，直到通过调用 okStopCapture 才会停止。亦可通过 -N 来指定在 N 帧内循环采集，如 -2， 即在两帧内循环。
返回值 ：如果调用成功，返回该采集函数的线程句柄。如果失败（如由于格式不支持等）返回 0（FALSE）。如果该目标体不被支持则返回 -1。
说明 ：该函数为线程控制方式的序列采集函数。除了只采集一帧的时时候同样支持回调函数外，等价于 okCaptureTo。
相 关 函 数 ：okGetCaptureStauts, okStopCapture, okCaptureByBuffer, okSetTargetRect, okSetVideoParam, okSetCaptureParam, okCaptureByBufferEx, okCaptureTo。

BOOL WINAPI okCaptureSequence(HANDLE hBoard,MLONG lStart, MLONG lNoFrame);

功能 ：中断控制的序列采集视频输入到缓存（BUFFER）。不等采集结束立即返回。
参数 ：hBoard ：输入卡句柄。
lStart ：采集到目标体的起始帧序号（起始为 0）。
lNoFrame ：采集帧数或采集方式。>0 时，要采集的帧数 ；
= 0 时，不启动采集， 仅用来查询所打开的卡是否支持本函数，如不支持，返回 -1，如 支持，则返回可用来序列采集的缓存帧 数 ；
= -1 时，在可用的缓存帧内循环采集。
返回值 ：如果调用成功，返回该目标体所支持的最大帧数。如果失败（如由于格式不 支持等）返回 0（FALSE）。不支持返回 -1。
说明 ：该函数为中断控制方式的后台序列采集函数，功能和参数的设置均类似于 okCatprueThread，并支持回调，支持缓存锁定，但仅在有中断支持的卡上起作用，目前可以使用该函数的卡在 WIN95/98/ME 上只有 M10、M20H、 M30、 M40、M60、M70、RGB20、C30N。在 WINNT4/2K/XP 上则绝大部分卡都支持， 只有部分老卡（如 C20、C80 等）不支持。需要注意的是由于是中断控制方式， 所以序列采集时图像不会丢帧，但回调是线程，当有大量的其它后台、线程操 作时，回调的帧号可能会不连续（跳帧号）。
相 关 函 数 ：okGetCaptureStauts, okStopCapture, okCaptureByBuffer, okSetTargetRect, okSetVideoParam, okSetCaptureParam, okCaptureByBufferEx, okCaptureTo。

MLONG WINAPI okCaptureTo(HANDLE hBoard, TARGET Dest,

MLONG start, MLONG lParam);

功能 ： 采集视频输入到指定目标体。这里的目标体可以是 VGA 屏（SCREEN），帧缓存（BUFFER），帧存体（FRAME）。缺省状态下为不等采集结束立即返回。如果要 立即读取数据，需要调用函数 okGetCaptureStauts，以确认采集结束后，再读取 数据。

参数 ：

hBoard ： 输入卡句柄。

Dest ：要采集到的目标体，可以是 SCREEN，BUFFER 或 FRAME。目标体的宏定义参见 OKAPI32.H。

start ：采集到目标体的起始帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN，该值只能为 0。

lParam ：采集帧数或采集方式。

1， 该值如果大于零（>0），即为采到目标体的帧数，此时为序列采集方式，=1 时为序列采集的特例单帧采集 ，但单帧采集时无回调支持。
如果 lParam 大于目标体的最大帧 b 数（Total），当逐帧采集到目标体的位置超过 Total 时，将 重置回到起始（start）位置继续开始逐帧采集，如此按方式 mode(n%total) 采集，直到采完 lParam 帧为止。

2， 如果等于零（=0），为连续采集（即实时采）方式，一直采集视频输入到 start 指定的位置直到通过调用 okStopCapture 才会停止。连续采集时无回调支持。

3， 如果等于-1， 为循环序列采集方式， 即从 start 指定的位置开始序列采集，当达到目标体的最大帧数（Total）时将回到起始（start）位置继续开始逐帧采集，如此无限循环下去，直到通过调用 okStopCapture 才会停止。亦可通过 -N 来指定在 N 帧内循环采集，如 -2， 即在两帧内循环。

返回值 ： 如果调用成功，返回该目标体所支持的最大帧数。如果失败（如由于格式不支持等）返回 0（FALSE）。如果该目标体不被支持则返回 -1。

说 明 ： 此函数为完全硬件支持的采集。当为循环序列采集方式或序列采集方式，但为不等结束返回方式时，本函数起动一序列采集的线程，然后立即返回调用它的程序，如果需要终止正在进行中的采集过程，可随时通过调用 okStopCapture 终止该过程。当为单帧或序列采集，并且为等待结束返回方式时，则采集全部完后才会返回调用它的程序。通过调用函数 okSetCaptureParam 的 CAPTURE_ SEQCAPWAIT 来设置单帧采集或序列采集时，是不等结束返回方式还是等结束返回方式，以及采集过程中的回调方式是立即回调用户程序还是等采集结束再回调。

注意，如是序列采集，该函数将从被调用时刻之后开始的一帧 / 场进行序列采集。

相关函数 ： okGetCaptureStauts, okStopCapture, okCaptureByBuffer, okSetTargetRect, okSetVideoParam, okSetCaptureParam, okCaptureByBufferEx。

MLONG WINAPI okCaptureToScreen(HANDLE hBoard);

功能 ： 启动连续采集视频输入到 VGA 屏幕（SCREEN），并立即返回。

参数 ： hBoard ：输入卡句柄。

返回值 ： 如果调用成功，返回 1（TRUE），否则返回 0（FALSE）。

说明 ： 此函数是为实时采时书写简捷而设的，实际只是函数 okCaptureTo 的一个特例 ，即 okCaptureTo(hBoard,SCREEN,0,0)。

相 关 函 数 ： okCaptureTo, okGetCaptureStauts, okStopCapture, okSetTargetRect, okSetVideoParam, okSetCaptureParam。

HANDLE WINAPI okCaptureByBuffer(HANDLE hBoard,TARGET dest, MLONG start, MLONG num);

功能 ： 间接（通过帧缓存 BUFFER）采集视频输入到指定目标体。这里的目标体可以是屏幕（SCREEN）、WINDOWS 窗口句柄、用户内存 MEMORY 或文件 FILE。可以单 帧采

集也可以进行多帧采集，可以不等采集结束立即返回，也可以等采集结束再返回。

当直接采到 SCREEN（VGA）有格式问题或冲突时，可以选择此函数 ；或当 需要采到用户程序申请的（虚）内存 MEMORY 时，也可以选择此函数。

参数 ： hBoard ： 输入卡句柄。

dest ： 要采集到的目标体，可以是 SCREEN，BUFFER，窗口句柄，用户内存指针， 或含有用户内存指针的 BLOCKINFO 结构变量指针，或文件名字符串指针。 BLOCKINFO 结构的定义参见 OKAPI32.H ； 各种格式的文件名输入方法详见 okSaveImageFile 中的说明。

start ： 采集到目标体的起始帧序号（起始为 0）。

num：要采集的帧数，当该值大于零时，将从 start 开始直到采集完 num 帧为止。如采到用户内存 MEMORY，该值必须小于用户内存可存放的最大帧数。当该值等于零时，将连续采集到 start 位置，直到通过调用 okStopCapture 终止为止。

注意 ：（1）本间接采集函数不支持循环序列采集方式。num 必须大于等于零。

（2）本间接采集函数支持采集时的回调函数。

返回值 ： 如果调用成功，返回非 0 ，否则返回 0（FALSE）。

说明 ： 此函数为间接采集。通常状态下，它是一线程，通过两帧缓存交替传送到屏幕、窗口、内存或文件的。采集帧的大小和格式取缓存 BUFFER 当前的设置值。如果本卡支持中断控制方式，并通过 okSetCaptureParam 函数 CAPTURE_MISCCONTROL 中的的 BIT0 设置成了中断控制方式，则是用本卡可用所有缓存循环序列采集。如果需要终止正在进行中的采集过程，可随时通过调用 okStopCapture 终止该过程。本函数启动一个序列连续采集进程。

注意：从帧缓存交替传送到目的体是通过调用 okConvertRect 或 okSaveImageFile 来完成的，所以要注意当前的数据（场）传送方式。

如果输入目的体是窗口句柄，采集到目标的坐标位置，就是对该句柄设置 的坐标，如从未通过 okSetTaregtRect 函数设置过或已用 NULL 清除，则使用对 SCREEN 设置的窗口坐标。

如果输入目的体是内存指针，应先清零第一个字节。 如果输入是 BLOCKINFO 结构变量指针，则必须设置结构变量中的元素 iType=BLKHEADER， lpBits 为用户内存指针。其余结构元素将由该函数根据缓存 BUFFER 当前的设置值填写。

如果指定的是文件名，该函数将采图像到该文件中，**注意，如果该文件已经存在，本函数不会删除它，而只是按指定的位置写入。如果用户需要新建文件，应首先删除同名文件，然后再调用此函数。**

如果指定的文件名是”.SEQ”或”.AVI”，则产生一序列格式文件。如文 件名是单幅格式文件”.BMP”或”.JPG”，则自动产生序列编号的多个 BMP 或 JPG 文件。

BLOCKINFO 结构的定义和序列文件 SEQ 的格式参见 OKAPI32.H ； BMP、 AVI、JPG 文件格式参见有关资料。

本函数也可指定目的体为帧缓存（BUFFER），使该函数仅采集到帧缓存并不再传送，而通过设置回调函数来传送图像到自己需要的目的区。

当为序列采集但为不等结束返回方式时，本函数起动一序列采集过程 然后立即返回调用它的程序，如果需要终止正在进行中的采集过程，可随时通过调用 okStopCapture 终止该过程。当为单帧或序列采集，并且为等待结束返回方式时，则采集全部完成后才会返回调用它的程序。通过调用函数 okSetCaptureParam 的 CAPTURE_SEQCAPWAIT 来设置单帧采集或序列采集时， 是不等结束返回方式还是等结束返回方式。

相关函数 ： okConvertRect ， okSaveImageFile， okGetCaptureStauts, okStopCapture, okCaptureTo , okCaptureByBufferEx, okSetCaptureParam。

HANDLE WINAPI okCaptureByBufferEx(HANDLE hBoard,MLONG fileset, TARGET dest, MLONG start, MLONG num);

功能 ： 间接（通过帧缓存 BUFFER）采集视频输入到指定目标体。这里的目标体可以是屏幕（SCREEN）、WINDOWS 窗口句柄、用户内存 MEMORY 或文件 FILE。可以单 帧采集也可以进行多帧采集，可以不等采集结束立即返回，也可以等采集结束再返回。直接采到

SCREEN（VGA）有格式问题或冲突时，可以选择此函数 ；或当需要采到用户程序申请的（虚）内存 MEMORY 时，也可以选择此函数。

参数 ： hBoard ：输入卡句柄。

fileset ：仅当是 JPG 文件时，是质量因子。如此处为 0，则取文件名字符串中 的 JPG 的质量因子设置，如也无或为 0，则取默认值 50。

dest ：要采集到的目标体，可以是 SCREEN，BUFFER，用户内存指针，或含有用户内存指针的 BLOCKINFO 结构变量指针， 或文件名字符串指针。 BLOCKINFO 结构的定义参见 OKAPI32.H ；各种格式的文件名输入方法详见 okSaveImageFile 中的说明。

start ：采集到目标体的起始帧序号（起始为 0）。

num：要采集的帧数，当该值大于零时，将从 start 开始直到采集完 num 帧为止。如采到用户内存 MEMORY，该值必须小于用户内存可存放的最大帧数。当该值等于零时，将连续采集到 start 位置，直到通过调用 okStopCapture 终止为止。

注意 ：（1）本间接采集函数不支持循环序列采集方式。num 必须大于等于零。

（2）本间接采集函数支持采集时的回调函数。

返回值 ： 如果调用成功，返回 1（TRUE），否则返回 0（FALSE）。

说明：此函数为 okCaptureByBuffer 功能扩展，增加存序列 JPG 文件时，图像质量因子选项。

相 关 函 数 ： okConvertRect ， okSaveImageFile, okGetCaptureStauts, okStopCapture, okCaptureTo , okCaptureByBuffer, okSetCaptureParam

MLONG WINAPI okGetSeqCapture(HANDLE hBoard,MLONG start,MLONG count);

功能 ： 获得已采集完的图像并启动采集最近的一帧图像，然后立即返回。

参数 ：

hBoard ：输入卡句柄。

start ： 要使用的双缓存的起始帧号，只有采第一幅即 count=0 时此参数被输入。如果本卡支持中断控制方式，并通过 okSetCaptureParam 函数 CAPTURE_ MISCCONTROL 中的的 BIT0 设置成了中断控制方式，则 start 是起始缓存号，采集在本卡可用所有缓存中循环。

count ： 调用的顺序号。起始调用必须为 0 ；结束时设为 -1 ；过程中为顺序号， 只要不为 0 和 -1 即可。

返回值 ： 返回已采集完的图像所在的帧号。应为 start 或 start+1 之一，但在中断控制 的模式下，由于是在整个缓存中的循环采集，则返回的帧号为 0 至最大帧号。

说明 ： 此函数首次调用必须设 count 等于 0。在正常模式下，此时函数开始采集第一帧，并等采集结束，再让硬件采集下一帧，然后返回结束的帧号。之后的调用，则是首先看上一次调用使硬件采集的一帧是已否结束，如未结束则等结束，如结 束了则再让硬件采集下一帧，然后即返回已结束的帧号。

相关函数 ： okCaptureTo, okCaptureThread, okSetCaptureParam。

（6）回调函数

BOOL WINAPI okSetSeqCallback(HANDLE hBoard, BOOL CALLBACK BeginProc(HANDLE hBoard), BOOL CALLBACK SeqProc(HANDLE hBoard, MLONG No), BOOL CALLBACK EndProc(HANDLE hBoard));

功能 ： 设置或清除序列采集 / 回显时的回调函数。回调函数包括开始（BeginProc），过程中（SeqProc）和结束（EndProc) 三个由用户自编的函数。序列采集 / 回显函数包括

okCaptureTo, okPlaybackFrom,okCaptureByBuffer, okPlaybackByBuffer。 但要注意，当作为单帧采集或连续采集调用 okCaptureTo，或作为单帧回显集调 用 okPlaybackFrom 时，则无回调支持。

参数 ：

hBoard ： 输入卡句柄。

BeginProc ：序列采集函数开始采集时回调的函数。用户程序在序列采集或回显线程开始时，有需要做的事情可有此函数完成。

SeqProc ： 序列采集函数每采集完一帧就回调的函数。用户程序在序列采集或回显线程逐帧循环过程中，有需要做的事情可由此函数完成。

EndProc ： 序列采集函数完成采集后回调的函数。用户程序在序列采集或回显线程结束时，有需要做的事情可由此函数来完成。

返回值 ： 无。

说明 ： 如果在序列采集函数的采集过程中，用户需要做自己的事情（如，回送 SCREEN 显示），可以通过此函数设置自己的回调函数。如果不再需要 ，**必须**通过此函数清除，okSetSeqCallback(hBoard,NULL,NULL,NULL); 回调函数所花的时间，是否会降低序列采集的速度，要看用户编写的回调函数 SeqProc 所耗时间的多少以及机器的快慢。

注意 ：在使用回调函数的过程中不能使用 okCloseBoard 和 okCloseBoardEx 函数关闭当前采集卡句柄。在 EndProc 函数中没有必要再次调用 okStopCapture。

回调函数要按如下方式定义 ：

BOOL CALLBACK BeginProc(HANDLE hBoard);

BOOL CALLBACK SeqProc(HANDLE hBoard, long No);

其中参数 No 为刚完成采集的帧缓存序号或正在回显的帧缓存序号，起始位置为 0。

BOOL CALLBACK EndProc(HANDLE hBoard);

相关函数 ： okCaptureTo, okPlaybackFrom, okCaptureByBuffer, okPlaybackByBuffer。

BOOL WINAPI okSetSeqProcWnd(HANDLE hBoard, HWND hwndMain);

功能：设置序列采集时回调函数中用来接收消息响应的窗口句柄。

参数：

hBoard ： 输入卡句柄。

hwndMain：回调函数中用来响应消息的窗口句柄。

返回值：设置成功返回 1，不成功返回 0。

说明：这个函数是 okSetSeqCallback 的替代函数。在有的编程语言中不能用函数指针再调用函数的方式，因此提供了该函数，用户设置了该函数后还需要在整个程序中设置三个回调函数相对应的三个消息（WM_BEGINSEQPROC 、 WM_SEQPROGRESS 、 WM_ENDSEQPROC）和响应函数才能真正实现回调函数功能。

相关函数： okSetSeqCallback ， okCaptureTo, okPlaybackFrom, okCaptureByBuffer, okPlaybackByBuffer。

BOOL WINAPI okSetEventCallback(HANDLE hBoard, INT32 iEvent, BOOL CALLBACK EventProc (HANDLE hBoard, MWORD dwReserved));

功能：设置某个事件的回调函数，回调函数可以在该事件发生之前（之后）响应。

参数：

hBoard ： 输入卡句柄。

iEvent：需要回调函数的事件，目前可以是：

EVENT_TOCLOSEDEVICE（1）关闭设备事件，回调函数会在关闭卡（设备）之前被调用。

EVENT_NETCONNECTCHANGE（2）网络连接状态改变，回调函数会在网络连接状态改变之后调用。

EventProc:对事件发生响应的回调函数，用户在事件发生时有想要进行的操作可以由此函数

完成。
返回值：如果成功返回 1，不成功返回 0。
说明：用户有想要在某些事件发生时做相应的操作可以通过这个函数设置回调函数来实现，回调函数要按如下方式定义：
BOOL CALLBACK EventProc(HANDLE hBoard, MWORD dwReserved); 其中 dwReserved 是保留参数，应该给 NULL。
相关函数：okSetCloseCallback，okSetCaptureParam, okSetTargetRect，okCloseBoard, okOpenBoard。

BOOL WINAPI okSetCloseCallback(HANDLE hBoard, BOOL CALLBACK CloseProc (HANDLE hBoard, MWORD dwReserved));

功能：设置 okCloseBoard 的回调函数，回调函数会在 okCloseBoard 关闭卡之前被调用。
参数：hBoard ：输入卡句柄。
CloseProc:对关闭卡事件发生响应的回调函数，用户在关闭卡之前有想要进行的操作可以由此函数完成。
返回值：如果成功返回 1，不成功返回 0。
说明：该函数是为用户设置关闭卡回调函数时书写便捷而设的，实际上就是 okSetEventCallback 的一个特例，既 okSetEventCallback(hBoard, EVENT_TOCLOSEDEVICE,, EventProc(HANDLE hBoard, MWORD dwReserved));一般直接用 okSetEventCallback 就可以了，回调函数要按如下方式定义：
BOOL CALLBACK CloseProc(HANDLE hBoard, MWORD dwReserved);
其中 dwReserved 是保留参数，应该给 NULL。
相关函数：okSetEventCallback，okSetCaptureParam, okSetTargetRect，okCloseBoard, okOpenBoard。

（7）采集状态和停止采集

MLONG WINAPI okGetCaptureStatus(HANDLE hBoard, BOOL bWait);

功能：获得当前采集或回显状态。
参数：
hBoard ：输入卡句柄。
bWait ：是否等待标志。=0：不等结束即返回；
=1：okCaptureTo 和 okPlayBackFrom 函数都等待结束后返回；
=2：仅 okCaptureTo 等待结束后返回。
返回值：返回当前采集 / 回显状态：
1，如果不在采集也不在回显状态，或已采集结束，则返回 0（FALSE）。
2，如果在实时采集状态则返回当前目标体，即：-1 屏幕 SCREEN，-2 帧存 FRAME，-3 监示器 MONITOR。
3，如果是在序列或循环序列采集 / 回显状态（目标体可以是帧缓存 BUFFER，用户内存 MEMORY 或文件 FILE）则返回当前正在采集 / 回显的帧位置序号（注意：这里起始为 1。只有 okGetCaptureStatus 和 okStopCapture 这两个函数是这样）。
说明：此函数既用来获得采集状态，也用来获得回显状态。在不等结束即返回方式下的单帧或序列采集 / 回显时，要知道是否完成就可以通过此函数来获悉，并可以通过 bWait=TRUE，使该函数直到在采集 / 回显完成后才返回。但如果是在连续采集 / 回显状态，则不论 bWait 是否 TRUE 均立即返回除 BUFFER 以外的当前采集 / 回显的目标体。在循环序列采集 / 回显时，则立即返回当前采集的帧序号。
注意：如果是循环以不等待的方式调用此函数查询状态，则不要死循环调用，如：while(okGetCaptureStatus(hBoard,0)); 而应使用插入等待的循环调用，如：do{ Sleep(10); }

while(okGetCaptureStatus(hBoard, 0));
相关函数：okCaptureTo, okCaptureByBuffer, okPlaybackFrom, okPlaybackByBuffer, okStopCapture。

MLONG WINAPI okGetLineReady(HANDLE hBoard,MLONG *iCurrFrameNo, MWORD dwRes);

功能：获得当前线阵卡已采集的行数。
参数：hBoard：输入卡句柄。
iCurrFrameNo：返回当前已采集的帧数。帧数是基于 1 的，0 为一帧也没有，1 为第一帧已完成，以此类推。设置的采集高度为每帧的行数。
dwRes：保留，目前未用。
返回值：返回当前已采集完的行数。行数也是基于 1 计数的，0 为当前帧第一行还没 采集完，1 为当前帧第一行已采集完，以此类推。
说明：此函数只对线阵采集卡有效。
相关函数：okCaptureTo, okCaptureByBuffer, okCaptureSequence, okGetCaptureStatus, okStopCapture。

MLONG WINAPI okStopCapture(HANDLE hBoard);

功能：终止当前的采集或回显过程。
参数：hBoard：输入卡句柄。
返回值：返回当前采集 / 回显状态：
1，如果不在采集也不在回显状态，则返回 0（FALSE）。
2，如果是在连续采集或回显状态，则返回当前目标体，即：1 帧缓存 BUFFER，-1 屏幕 SCREEN，-2 帧存 FRAME，-3 监示器 MONITOR；否则返回 0（FALSE）。
3，如果是在序列采集状态（目标体可以是帧缓存 BUFFER，用户内存 MEMORY 或文件 FILE）则返回当前已经采集 / 回显的所有帧数。（注意：这里起始为 1。只有 okGetCaptureStatus 和 okStopCapture 这两个函数是这样））。
说明：此函数既用来终止采集过程，也用来终止回显过程。当在不等结束方式下的单帧或序列（采集 / 回显）过程中，或循环序列及连续（采集 / 回显）过程中，都可以随时通过此函数来终止当前采集 / 回显过程。该函数在序列采集的时候有可能返回了但是采集并没有立即停止。
相关函数：okCaptureTo, okCaptureByBuffer, okPlaybackFrom,okPlaybackByBuffer, okGetCaptureStatus。

MLONG WINAPI okStopCaptureEx(HANDLE hBoard, DWORD dwCmdCode, long lParam);

功能：这个是 okStopCapture 的扩展函数。

（8）数据传送

INT32 WINAPI okReadPixel(HANDLE hBoard, TARGET src, MLONG start, SHORT x, SHORT y);

功能：从源目标体的指定帧图像中的指定位置（X，Y）读出图像的数值。这里的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME），以及带有用户内存 MEMORY 地址的 BLOCKINFO 指针。大量读数据时不建议用此函数。注：对于 M60 卡，其帧存体不能用此函数读出。
参数：hBoard：输入卡句柄。

src ： 数据源目标体 ，
start ： 数据目标体的指定帧序号（起始为 0）。
x ： 一幅图像中指定位置的 X 坐标，
y ： 一幅图像中指定位置的 Y 坐标。
返回值 ： 按目标体当前的数据格式返回该位置的图像数据，例如：对于格式 GRAY8， 返回的是 8 位灰度数据；对于格式 RGB888，返回的则是 RGB 各占 8 位的 24 位数据。如果源目标体不支持则返回 -1。
相关函数 ： okWritePixel, okTransferRect, okReadRect, okWriteRect 。

INT32 WINAPI okWritePixel(HANDLE hBoard, TARGET tgt, MLONG start, SHORT x, SHORT y, INT32 lValue);

功能 ： 向目标体的指定帧图像中的指定位置（X，Y）写一数值。这里的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME），以及带有用户内存 MEMORY 地址的 BLOCKINFO 指针。大量写数据时不建议用此函数。
注 ： 对于 M60 卡，其帧存体不能用此函数写数据。
参数 ：
hBoard ： 输入卡句柄。
tgt ： 写入数据目标体 ，
start ： 数据目标体的指定帧序号（起始为 0）。
x ： 一幅图像中指定位置的 X 坐标，
y ： 一幅图像中指定位置的 Y 坐标。
lValue ： 写入的图像数值，应与目标体当前的数据格式一致。
返回值 ： 如成功则返回写入的数据，如果目标体不支持则返回 -1。 说明 ：
相关函数 ： okReadPixel, okTransferRect , okReadRect, okWriteRect 。

MLONG WINAPI okSetConvertParam(HANDLE hBoard,INT32 wParam, MLONG lParam);

功能 ： 设置传送与变换图像数据时的传送方式。
参数 ： hBoard ： 输入卡句柄。
wParam ： 指定功能项目，所支持的项目如下（可参见 OKAPI32.H 中的宏定义）。
* CONVERT_RESETALL（0）重置所有项的参数值为系统缺省值。
* CONVERT_FIELDEXTEND(1)：设置数据传输场扩展模式，对应的 lParam 可以为以下几种模式 ：

FIELD_JUSTCOPY(0)	逐行取源数据，不扩展地传输。
FIELD_COPYEXTEND(1)	逐行取源数据传输，并逐行扩展，
FIELD_INTERLEAVE(2)	隔行取奇场源数据，不扩展地传输，
FIELD_INTEREXTEND(3)	隔行取奇场源数据，并进行逐行扩展地传输，
FIELD_COPYINTERPOL(4)	逐行取源数据传输，并进行内插行扩展，
FIELD_INTERINTERPOL(5)	隔行取奇场源数据，并进行偶场内插扩展地传输
FIELD_INTEREVEN(6)	隔行取偶场源数据，不扩展地传输
FIELD_INTEREXTEVEN(7)	隔行取偶场源数据，并进行逐行扩展地传输，
FIELD_JUSTCOPYODD(8）	只拷贝奇场源数据到奇场位置，
FIELD_JUSTCOPYEVEN(9)	只拷贝偶场源数据到偶场位置，
FIELD_ODDEVENCROSS(10)	拷贝奇场源数据到偶场位置，拷贝偶场源数据到奇场位置。

FIELD_COPYROWTOODD(11)	拷贝数据到奇场位置
FIELD_COPYROWTOEVEN(12)	拷贝数据到偶场位置

本参数设置影响如下函数 ： okTransferRect, okConvertRect，okConvertRectEx, okReadRect, okWriteRect, okReadRectEx, okWriteRectEx。

* CONVERT_PALETTE(2): 设置转换时使用的调色板，lParam=0 时，设置成缺省的灰度调色 板 ；lParam.>0， 设 置 成 lParam. 指向的调色板，灰度为 256。 当黑白（8、10、12 位 ）转换成彩色（16、24，32 位 ）时， 对 okConvertRect, okConvertRectEx, okReadRect, okWriteRect, okReadRectEx, okWriteRectEx 有效。

* CONVERT_HORZEXTEND(3) 水平方向扩展，整数倍扩大。

* CONVERT_HORZSTRETCH（4）水平方向拉伸，可以是任意倍数的。

* CONVERT_MIRROR(5): 设置软件镜象传送。

lParam=0 无；
lParam=1 左右镜象；
lParam=2 上下镜象；
lParam=3 左右上下都镜象。

只有传送的源和目的之一为缓存 BUFFER 时，用 okConvertRect、okConvertRectEx、okReadRectEx、 okWriteRectEx 及 okCaptureByBuffer、okCaptureByBufferEx 中目标体不是文件 时才有效。另外，对于上述这些函数，当源和目的之一是 BLOCKINFO 所指 定的目的体时，也可以不设置此参数，而通过设置 BLOCKINFO 中的 iWidth 和 iHeight 为负值来完成 X 和 Y 的镜象。

*CONVERT_UPRIGHT（6）图像做 90 度旋转，lParam=1，向右转 90 度；=2，向左转 90 度。

*CONVERT_NOSYNCSCREEN （ 7 ） 为 1 表 示 不 等 待 场 同 步 信 号 ， 只 有 在 使 用 CaptureByBuffer 函数并且目标体是 wnd 的时候起效。

*CONVERT_WINDOWLEVEL（10）设置灰度窗的取值范围，通过设置窗口的起点和宽度实现。lParam 的低字是窗口的中心灰度， lParam 的高字是窗口的宽度，当高字=0 时表示该参数设置无效。主要是当图像位深高于 8 位要转换成 8 位的时候或者转换的目标体格式的某一通道的格式是 8 位的时候才起效，而且当图像的格式发生改变的时候不自动进行调整，也可以是数据源的某一路通道是 8 位的情况，用户需根据实际需要自行调整参数值。该设置只有当下面 CONVERT_LEVELMAPLUT 设置为 0 的时候才有效。

*CONVERT_LEVELMAPLUT （11）设置（拷贝）用户自定义的灰度查找，当转换的时候按照此查找表对数据进行转换。 lParam=0 (默认): 关闭； lParam>0: 设置一个指向 WORD 型的 LUT 查找表的指针。这里必须是 WORD 的查找表，而且 LPWORD[0]中存放的是当前查找表的灰度级的位数（比如：当查找表的数据源格式是 GRAY8, GRAY888, RGB888 的时候，那里面的数值就应该是 8；当查找表的数据源格式是 GRAY16 的时候,那里面的数值就应该是 16）。从 LPWORD[1]开始是查找表数据的实际内容。查找表的最大值不能大于 CONVERT 目标体所能支持的最大值。该项设置**必须**在使用 CONVERT 功能后通过设置 lParam=0 被重置。

lParam ：对应 wParam 的参数值。意义如上所述。

返回值 ： 如果指定的项目或参数不被支持，返回 -1 ；如果失败则返回 -2。 如果成功则返回该项目之前的参数值。

说明 ： 如果 lParam = -1（GETCURRPARAM），则仅返回该项目当前的参数值。通过此 函数设置的数据传送参数，在应用程序调用 okCloseBoard 之后自动存入该卡的初始化配置文件中，在应用程序调用 okOpenBoard 之后，此函数将从对应卡的 初始化配置文件中取出数据传送参数来进行图像卡的初始数据传送参数设置。 已设置的参数如果没有改变将一直有效。所以当需要不同的设置时就要重新设置相应的参数。

相 关 函 数 ： okSetVideoParam, okSetCaptureRect ， okTransferRect, okConvertRect, okOpenBoard, okCloseBoard。

INT32 WINAPI okTransferRect(HANDLE hBoard, TARGET dest, MLONG iFirst, TARGET src, MLONG iStart, MLONG lNum);

功能 ： 从源目标体快速传送窗口（RECT）内图像数据到目的目标体，不进行格式（位数）转换，等传送完成后返回。这里的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME），以及带有用户内存 MEMORY 地址的 BLOCKINFO 指针。调用此函数时目的与源一定要格式相同，否则就可能会出错。
注 ：对于 M60 卡，其帧存体（FRAME）仅是作为模板之用，是一特殊的帧存体，该帧存体只能通过硬件从缓存（BUFFER）中写入，不能变换格式，不能读出。 因而只能通过本函数将设置成同样格式（8 位）的缓存中的某一幅图像写入帧存体。其它数据传送函数，如 okConvertRect 等，则不能使用。
参数 ： hBoard ： 输入卡句柄。
dest ： 目的目标体。
iFirst ： 数据目标体传向的第一帧序号（起始为 0）。
src ： 数据源目标体。
iStart ： 数据源目标体的传送开始帧序号（起始为 0）。
lNum ： 共传送帧数，必须大于 0。
返回值 ： 如果源目标体或目的目标体不支持返回 -1 ；如果失败则返回 0（FALSE）。 如果成功完成传送则返回一块图像的数据长度（以字节为单位）。
说明 ：如果要传送的帧数 lNum 大于源目标体或目的目标体的可容纳的总帧数，将重置 到该目标体的起始位置（序号 0）再继续逐帧传送，如此下去直到传送完 lNum 帧。
本传送函数将按源目标体的格式（字节数 / 象元），源目标体和目的目标体的最小宽高传送数据。也就是说格式不会改变。这样调用本函数前要保证源目标 体和目的目标体的格式是一致的，否则会产生错误。如果需要格式匹配则要调 用函数 okConvertRect。

本函数还将根据通过 okSetConvertParam 对 CONVERT_FIELDEXTEND 参数项的设置，进行逐行 / 隔行取数据，扩展 / 不扩展地传送数据。

传送规则如下 ：a: 如果传送模式 = 0（FIELD_JUSTCOPY）：将逐行传送数据 ；
b: 如果传送模式 = 1（FIELD_COPYEXTEND），实际所传送数据的行数为源 与目的窗口行数中最小的，并将逐行取源数据并扩展行（逐行复制），此模式 主要用来把场图像扩展成帧格式，实际所取源数据的行数为目的窗口行数的一半 ；
c: 如果传送模式 = 2（FIELD_INTERLEAVE）：将隔行取源数据传送，此模 式主要用来把帧图像抽成场图像格式，实际所传送数据的行数为源窗口行数的 一半 ；
d: 如果传送模式 = 3（FIELD_INTEREXTEND）：将隔行取源数据并扩展行（逐行复制行），此模式主要用来把帧图像变成扩展场格式，实际所传送并扩展 的数据总行数为源与目的窗口行数中最小的。

如源或目的之一为 BLOCKINFO，其结构变量 iHeight 为**负值**时，是以**上下颠倒**方式传送的，此功能专为 WINDOWS 中的 DIB 的倒置所设。
注意：此函数参数中有 BLOCKINFO 时，需要程序结构体按照一字节对齐方式编译。

相关函数 ： okConvertRect, okReadRect, okWriteRect，okSetConvertParam

INT32 WINAPI okConvertRect(HANDLE hBoard, TARGET dst, MLONG first, TARGET src, MLONG start, MLONG lParam);

功能 ：从源目标体匹配地传送窗口（RECT）数据到目的目标体，如源与目的格式不同将进行位转换，等传送完成后返回。这里的目标体可以是 VGA 屏幕（SCREEN）， 帧缓存（BUFFER），帧存体（FRAME），以及带有用户内存 MEMORY 地址的 BLOCKINFO 指针。
参数 ： hBoard ： 输入卡句柄。
dst ： 数据传向的目标体。
first ： 数据目标体传向的第一帧序号（起始为 0）。

src ： 数据源目标体。
start ： 数据源目标体的传送开始帧序号（起始为 0）。
lParam ： LOWORD(lParam)=Num ： 共传送帧数，必须大于 0。
HIWORD(lParam)=chann: 为基色通道抽取或单通道扩展的模式。分为两种情况，
A ： 源是彩色（如 24 位）目的是黑白（8 位)：
=0 ： 读取红绿蓝三通道的平均值
=1 ： 只读取红色通道数据
=2 ： 只读取绿色通道数据
=3 ： 只读取蓝色通道数据
B ： 源是黑白而目的是彩色：
=0 ： 扩展成 8 位黑白数据
=1 ： 只读到 lpBuf 的红色通道
=2：只读到 lpBuf 的绿色通道
=3：只读到 lpBuf 的蓝色通道
返回值 ： 如果源目标体或目的目标体不支持返回 -1 ； 如果失败则返回 0（FALSE）。 如果成功完成传送则返回一块图像的数据长度（以字节为单位）。
说明 ： 如果要传送的帧数 lNum 大于源目标体或目的目标体容纳的总帧数，将重置到该目标体的起始位置（序号 0）再继续逐帧传送，如此下去直到传送完 lNum 帧。
本传送函数将按源目标体和目的目标体的最小宽高传送数据。如果两者格式（字节数 / 象元）不同将会进行格式匹配转换。但如需转换速度会有所降低。
本函数还将根据通过 okSetConvertParam 对 CONVERT_FIELDEXTEND 参数项 的设置，进行逐行 / 隔行取数据，扩展 / 不扩展地传送数据。
传送规则如下 ：
a: 如果传送模式 = 0（FIELD_JUSTCOPY）：将逐行传送数据 ；
b: 如果传送模式 = 1（FIELD_COPYEXTEND），实际所传送数据的行数为源与目的窗口行数中最小的 ：将逐行取源数据并扩展行（逐行复制行），此模式 主要用来把场图像扩展成帧格式，实际所取源数据的行数为目的窗口行数的一半 ；
c: 如果传送模式 = 2（FIELD_INTERLEAVE）：将隔行取源数据传送，此模式主要用来把帧图像抽成场图像格式，实际所传送数据的行数为源窗口行数的一半 ；
d: 如果传送模式 = 3（FIELD_INTEREXTEND）：将隔行取源数据并扩展行（逐行复制行），此模式主要用来把帧图像变成扩展场格式，实际所传送并扩展的数据总行数为源与目的窗口行数中最小的。
如源或目的之一为 BLOCKINFO，其结构变量 iHeight 为负值时，是以上下颠倒方式传送的，此功能专为 WINDOWS 中的 DIB 的倒置所设。
注意：此函数参数中有 BLOCKINFO 时，需要程序结构体按照一字节对齐方式编译。

相关函数 ： okTransferRect, okReadRect, okWriteRect，okSetConvertParam。

INT32 WINAPI okConvertRectEx(HANDLE hDstBoard, TARGET dst,MLONG first,HANDLE hSrcBoard,TARGET src,MLONG start, MLONG no);

功能 ： 从源卡的目标体匹配地传送窗口（RECT）数据到目的卡的目标体，如源与目的格式不同将进行位转换，等传送完成后返回。这里的目标体可以是 VGA 屏幕（SCREEN）， 帧缓存（BUFFER）， 帧存体（FRAME）， 以及带有用户内存 MEMORY 地址的 BLOCKINFO 指针。
参数 ： hDstBoard ： 目的卡的句柄。
dst ： 数据传向的目标体。
first ： 数据目标体传向的第一帧序号（起始为 0）。
hSrcBoard ： 源卡的句柄。
src ： 数据源目标体。

start ： 数据源目标体的传送开始帧序号（起始为 0）。
no: LOWORD(no)=Num ： 共传送帧数，必须大于 0。
HIWORD(no)=chann: 为基色通道抽取或单通道扩展的模式。分为两种情况，详见 okConvertRect 中的说明。
返回值 ： 如果源目标体或目的目标体不支持返回 -1 ；如果失败则返回 0（FALSE）。 如果成功完成传送则返回一块图像的数据长度（以字节为单位）。
说明 ：此函数是 okConvertRect 的功能扩展的函数，只增加了数据源对应的源卡句柄， 其余完全一样。详细内容见 okConvertRect 中的说明。
相关函数 ： okConvertRect, okReadRect, okWriteRect，okSetConvertParam。

INT32 WINAPI okReadRect(HANDLE hBoard,TARGET src, MLONG start, LPBYTE lpBuf);

功能：从源目标体指定帧位置（帧序号）读一窗口（RECT）图像数据到用户内存缓存区。这里的源目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME）。
参数 ： hBoard ： 输入卡句柄。
src ： 数据源目标体。
Start ： 数据源目标体的要读窗口所在帧序号（起始为 0）。 LpBuf ： 用户内存缓存区指针。
返回值 ：如果源目标体不支持返回 -1 ；如果失败则返回 0（FALSE）。如果成功则返回 所要读块数据长度（以字节为单位）。
说明 ：如果 lpBuf=NULL，则本函数只是返回所要读块数据长度。 本函数将按源目标体当前的块窗口（RECT）设置和格式（字节数 / 象元），读数据，读出的块数据将逐行顺序存放到 lpBuf 指向的户内存缓存区。 本函数还将根据通过 okSetConvertParam 对 CONVERT_FIELDEXTEND 参数项的设置，进行逐行 / 隔行取数据，扩展 / 不扩展地传送数据。
读取数据规则如下 ： a: 如果传送模式 = 0（FIELD_JUSTCOPY）：将逐行读取数据，实际所读取数据的行数为源窗口所设置的行数 ；
b: 如果传送模式 = 1（FIELD_COPYEXTEND），将逐行读取源数据并扩展行（逐行复制行），此模式主要用来把场图像扩展成帧格式，实际所取源数据的行数为源窗口当前设置的行数 ；
c: 如果传送模式 = 2（FIELD_INTERLEAVE）：将隔行读取数据，此模式主 要用来把帧图像抽成场图像格式，实际所读取数据的行数为源窗口行数的一半；
d: 如果传送模式 = 3（FIELD_INTEREXTEND）：将隔行读数据并扩展行（逐行复 制行），此模式主要用来把帧图像变成扩展场格式，实际所读数据的行数为源 窗口行数的一半，但读到用户内存的数据行数则为源窗口的行数。
相关函数 ： okWriteRect, okWriteRectEx, okReadRectEx, okTransferRect, okConvertRect，okSetConvertParam

INT32 WINAPI okWriteRect(HANDLE hBoard, TARGET dst, MLONG start, LPBYTE lpBuf);

功能 ：把用户内存缓存区的图像数据写到目的目标体的指定帧位置（帧序号）的当前窗口（RECT）。这里的目的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER）， 帧存体（FRAME）。
参数 ： hBoard ： 输入卡句柄。
dst ： 写入的目的目标体。
start ： 目的目标体的写入帧位置序号（起始为 0）。 lpBuf ： 用户内存缓存区指针。
返回值 ：如果目的目标体不支持返回 -1 ；如果失败则返回 0（FALSE）。如果成功则返回所写块数据长度（以字节为单位）。

说明 ：本函数将按目的目标体当前的窗口（RECT）设置和格式（字节数 / 象元）写数据，在 lpBuf 指向的用户内存缓存区存放的数据，应按目的目标体当前的大小设置逐行顺序存放。

本函数还将根据通过 okSetConvertParam 对 CONVERT_FIELDEXTEND 参数项的设置，进行逐行 / 隔行取数据，扩展 / 不扩展地传送数据。

写数据规则如下：a: 如果传送模式 = 0（FIELD_JUSTCOPY）：将逐行写数据，实际所写数据的行数为目的窗口所设置的行数 ；

b: 如果传送模式 = 1（FIELD_ COPYEXTEND），将逐行取用户内存的数据并扩展行（逐行复制行）然后写到目 的窗口，此模式主要用来把场图像扩展成帧格式，实际所写数据的行数为目的 窗口当前设置的行数 ；

c: 如果传送模式 = 2（FIELD_INTERLEAVE）：将隔行读 取用户内存的数据，而逐行写到目的窗口，此模式主要用来把帧图像抽成场图 像格式，实际上从用户内存读取数据的行数为目的窗口的行数 ；

d: 如果传送模 式 = （3 FIELD_INTEREXTEND）：将隔行读取用户内存数据并扩展行（逐行复制行）， 然后写到目的窗口，此模式主要用来把帧图像变成扩展场格式，实际所读取的 用户内存数据的行数为目的窗口行数的一半。

相关函数 ：okReadRect, okWriteRectEx, okReadRectEx, okTransferRect,okConvertRect，okSetConvertParam。

INT32 WINAPI okReadRectEx(HANDLE hBoard,TARGET src, MLONG start, LPBYTE lpBuf,INT32 lParam);

功能 ：从源目标体指定帧位置（帧序号）读一窗口（RECT）图像数据到用户内存缓存区。这里的源目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME）。并可指定在读取时是否进行位转换、基色通道抽取或单通道扩展。

参数 ：

hBoard ：输入卡句柄。

src ：数据源目标体。

start ：数据源目标体的要读窗口所在帧序号（起始为 0）。

lpBuf ： 用户内存缓存区指针。

lParam ：读取方式参数。

1. 其低字 LOWORD(lParam）为用户申请的 lpBuf 的位数格式码（如：FORM_GRAY8 等），如为零则默认为与源目标体相同。

2.HIWORD(lParam) 为基色通道抽取或单通道扩展的模式。分为两种情况 ：

情况 A，即源是彩色（如 24 位）目的是 8 位黑白 ：

=0 为读取红绿蓝三通道的平均值 ；

=1 为只读取红色通道数据 ；

=2 为只读取绿色通道数据 ；

=3 为只读取蓝色通道数据。

情况 B，即源是 8 位黑白而目的是彩色 ：=0 为将三基色都置成相同的黑白数据 ；

=1 为只读源数据到 lpBuf 的红色通道 ；

=2 为只读源数据到 lpBuf 的绿色通道 ；

=3 为只读源数据到 lpBuf 的蓝色通道。

例如：当情况 A 时，lParam=MAKELONG(FORM_GRAY8,1), 数据源的格式为 24 位，即从 24 位数据源中只读出 8 位红基色数据顺序存放在 lpBuf 中。

返回值 ：如果源目标体不支持返回 -1 ；如果失败则返回 0（FALSE）。如果成功则返回所要读块数据长度（以字节为单位）。

说明 ：本函数为 okReadRect 的功能扩展函数。

如果 lpBuf=NULL，则本函数只是返回所要读块数据长度。 本函数将按源目标体当前的块窗口（RECT）设置和格式（字节数 / 象元），
读数据，读出的块数据将逐行顺序存放到 lpBuf 指向的用户内存缓存区。本函数还将根据通过 okSetConvertParam 对 CONVERT_FIELDEXTEND 参数项
的设置，进行逐行 / 隔行取数据，扩展 / 不扩展地传送数据。其读取数据的规则参见 okSetConvertParam 与 okReadRect。
相关函数 ： okWriteRectEx, okWriteRect, okReadRect, okTransferRect, okConvertRect，okSetConvertParam。

INT32 WINAPI okWriteRectEx(HANDLE hBoard,TARGET dst, MLONG start,LPBYTE lpBuf,INT32 lParam);

功能 ： 把用户内存缓存区的图像数据写到目的目标体的指定帧位置（帧序号）的当前窗口（RECT）。这里的目的目标体可以是 VGA 屏幕（SCREEN），帧缓存（BUFFER）， 帧存体（FRAME）。并可指定写入时是否进行位转换、基色通道抽取或单通道扩展。
参数 ： hBoard ： 输入卡句柄。
dst ： 写向的目的目标体。
start ： 目的目标体的写向帧位置序号（起始为 0）。
lpBuf ： 用户内存缓存区指针。
lParam ： 写入方式参数。

1. 其低字 LOWORD(lParam）为用户申请的 lpBuf 的位数格式码（如： FORM_GRAY8 等），如为零则默认为与写目标体的相同。

2.HIWORD(lParam) 为基色通道抽取或单通道扩展的模式。分为两种情况 ：

情况 A，即源是彩色（如 24 位）目的是 8 位黑白 ：

=0 为写源红绿蓝三通道的平均值到目的体 ；
=1 为只取源红色通道数据写入目的体 ；
=2 为只取源绿色通道数据写入目的体 ；
=3 为只取源蓝色通道数据写入目的体。

情况 B，即源是黑白而目的是彩色 ：

=0 为将黑白数据重复写到三基色通道 ；
=1 为只将黑白数据写到红色通道区 ；
=2 为只将黑白数据写到 绿色通道区 ；
=3 为只将黑白数据写到蓝色通道区。

返回值 ： 如果目的目标体不支持返回 -1 ；如果失败则返回 0（FALSE）。如果成功则返回所写块数据长度（以字节为单位）。
说明 ： 本函数为 okWriteRect 的功能扩展函数。 本函数将按目的目标体当前的窗口（RECT）设置和格式（字节数 / 象元）写数据，在 lpBuf 指向的用户内存缓存区存放的数据，应按目的目标体当前的大小设置逐行顺序存放。

本函数还将根据通过 okSetConvertParam 对 CONVERT_FIELDEXTEND 参数项 的设置，进行逐行/ 隔行取数据，扩展/不扩展地传送数据。其读取数据的规则参见 okSetConvertParam 与 okWriteRect。
相关函数 ： okReadRectEx, okWriteRect, okReadRect, okTransferRect, okConvertRect，okSetConvertParam。

（9）图像文件读写

MLONG WINAPI okSaveImageFile(HANDLE hBoard, LPSTR szFileName, MLONG first,TARGET target, MLONG start, MLONG num);

功能 ： 从源目标体存图像窗口（RECT）到硬盘，等存盘完成后返回。这里的源目标体可以

是 VGA 屏幕（SCREEN），帧缓存（BUFFER），帧存体（FRAME），以及带有 用户内存 MEMORY 地址的 BLOCKINFO 指针。

参数 ：

hBoard ： 输入卡句柄。

szFileName：存盘文件名。可以是“.SEQ”、“.AVI”、“.BMP”、“JPG”、“.MP4”或“.RAW”为后缀的文件名。

如果存 **SEQ** 文件，但是希望图像数据是 JPG 压缩的，可以按如下格式输入文件名“AA.SEQ , JPG, 60”。其中“,JPG，60” 表示 图像数据是 JPG 压缩的，60 是 JPG 的质量控制因子，所存图 像的文件名仍为“AA.SEQ”。

如果存 **MP4** 文件，所存图像的文件名为“AA.mp4”，**目前保存的文件不能用通用播放器打开**。

如果存“**.AVI**”，但需要是压缩的，目前支持 MotionJPEG 和 MPEG4 和 h264 编码方式。

如需要存 **MJPEG**， 输入文件名后跟“,MJPG”，缺省质量因子为 60，如需自己指定可再后跟系数，如“AA.AVI,MJPG,70”。

需要存 **MPEG4**，输入文件名后跟“, MPG4”，缺省质量因子为 60，表示编码时候的运算复杂度，如需自己指定可再后跟系数，如“AA.AVI, MPG4，80”。

需要存 **H264**，输入文件名后跟“, h264”，如“AA.AVI, H264”，该编码方式不支持压缩因子，压缩率大约为 1%。

first：对于 SEQ、AVI 文件，该参数是从序列文件中的第几幅开始存（起始为 0）。

对于 **BMP** 文件，first=0，按标准 BMP 格式存放，即把目标体的格式转换为 8 位（黑白）或 24 位（彩色）来存放 ；

first=1，则按支持扩展格式的 BMP 存放，即按目标体当前的的格式存放而不论它是否 8 位或 24 位，注意，如此时目标体的格式是 16 位或 32 位，生成的 BMP 图像文件，一般的应用软件（如 PaintBrush 等）因不支持这些扩展格式而不能读出。

对于 **JPG** 文件，该参数也是质量控制因子（1~100）。（但优先于通过文件名字符串输入方法，只有这里设置为 0 时，文件名字符串的输入才 有效。）=1 ： 保质最差（但压缩比最大），=100 保质最好（压缩比最小）。=0 为缺省设置，等价于值 50。当目标体的格式为非 8 位（黑白）和 24 位（彩色）时，该函数将自动转换成这两种格式再进行压缩。

target ： 源目标体。

start ： 源目标体开始读取的帧序号（起始为 0）。

num ： 要存盘图像幅数，应该大于 0。

返回值 ：如果是“.SEQ”文件，返回文件的总长度（以字节为单位）；如果是“.BMP”、“.JPG”或“.RAW”文件，返回单个文件的长度（以字节为单位）。如果失败返回零。

说明 ：存 SEQ 文件不进行格式转换，因而文件将按目的体的数据位数格式存放。SEQ 序列图像文件的格式为，头 20 字节（即结构变量 SEQINFO，详细参见头文件 OKAPI32.H）是序列图像的格式信息，紧跟其后按行顺序存放各帧原始格式图像数据。

如果是“.SEQ”文件，函数将从序列文件的第 iFirst 幅开始，存 num 幅图像。

如果输入的文件名已存在，将不会被删除。因此，要新建文件需调用前自行删除。

RAW 格式的图像文件，是按行顺序存原始格式的图像数据.。

如果是“.BMP”、“.JPG”或“.RAW” 单幅格式文件，并且 num=1，则 只存一幅单幅格式（如 BMP）的图像文件，如果是存某一单幅格式文件且 num>1，则存多幅单幅格式（如 BMP）的图像文件。多幅的单幅格式（如 BMP） 图像文件的命名规则如下：（1），如果给定的文件名已含有数字（如：”OK1000. BMP”），则函数以第一个出现的数字串，1000 开始顺序加 1，来命名各文件名， 最多可到“OK9999.BMP”；（2），如果给定的文件名不

含有数字（如 ：”OK. BMP”）， 则函数在给定的主名最后自动附加三位数字，从 000 开始顺序加 1， 来命名之后的各文件名，最多可到”OK999.BMP”。注意 ；不要只用数字来定义文件名，应至少含有 1 个字母。

对于“.SEQ”、“.AVI”、“.M2V”等多图像文件，要存储序列格式的图像，包括压缩格式的，如 JPEG，MJPEG，MPEG4，H264 等，可以通过采用“开始，结束” 的头尾结构来提高存入的效率。方法是 ：①先调用本函数一次，给入的文件名后要加上“,beg” ，这时该函数仅创建序列文件，并不实际存图像 ；②然后每实际顺序存入一幅图像时，调用一次本函数，这时每次给入的文件名后都要加 上“,con”，如此连续调用即可实现顺序存图像 ；在这种方式下，文件自动从起始位置开始顺序存储，first 参数变为无效；③在结束序列存入图像后，必须再调用本函数一次，此时给入的文件名后要加上“,end”，此后程序才生成并关闭文件。

相关函数 ： okLoadImageFile，okCaptureByBuffer。

MLONG WINAPI okLoadImageFile(HANDLE hBoard, LPSTR szFileName, MLONG first, TARGET target, MLONG start, MLONG num);

功能 ：把图像文件装入到目标体窗口，等装入完成后返回。这里的目标体可以是 VGA 屏 幕（SCREEN）， 帧缓存（BUFFER）， 帧存体（FRAME）， 以及带有用户内存 MEMORY 地址的 BLOCKINFO 指针。

参数 ：

hBoard ： 输入卡句柄。

SzFileName：图像文件名。可以是“.SEQ”、“.AVI”、“.M2V”、“.BMP”、“.JPG”、“.RAW”、“.TIF”、或“.GIF”，“.MP4”等格式的文件名。（**MP4 只能播放用 ok 驱动录出来的文件**）

first ： 对于 SEQ、.AVI、M2V、MPG 序列文件，该参数是从序列文件中的第几 幅开始读（起始为 0）。对于 RAW 文件，如非零，则 LOWORD 为指定的宽度， HIWORD 为指定的高度。

target ： 目的目标体。

start ： 目的目标体开始装入的帧序号（起始为 0）。

num ： 要装入的图像幅数，应该大于 0 ；如等于 0，则不装入图像，而只返回文件中的图像幅数。

返回值：如果是“.SEQ”，“.AVI”，“.M2V”等格式的序列文件，返回文件中图像的幅数； 如果是“.BMP”、“.JPG”“.RAW”、“.TIF”或“.GIF”等单图像格式的文件， 返回单个文件的图像数据长度（以字节为单位）。如果失败返回零。

说明 ： 装入文件时，如与目的体格式不一致，且目的体是格式不可变的，如 SCREEN、FRAME，则进行数据格式转换 ；如是可变的，如 BUFFER、BLOKINFO，则不进行 数据格式转换，而是改变目的体格式。

如果是 SEQ、AVI、M2V 文件，函数将从序列文件的第 first 幅开始，读 出 num 幅图像。

如果是单幅格式（如 BMP）的图像文件并且 num=1，则读单幅格式（如 BMP）的图像文件，如果是单幅格式（如 BMP）的图像文件且 num>1，则读多 幅的单幅格式（如 BMP）的图像文件。多幅单幅格式（如 BMP）的图像文件， 名字规则约定如下 ：1，如果给定的文件名已含有数字，如 ：”OK100.BMP”， 则函数以 100 开始顺序加 1，认定以下的各文件名 ；如果给定的文件名不含有 数字（如 ：”OK.BMP”），则函数在给定的主名后自动附加三位数字，从 000 开 始顺序加 1，来认定以下的各文件名。

对于“.AVI”文件，如果存储的图像是序列压缩格式 MJPEG 或 MPEG4， 及 MPEG2 压缩格式的“.M2V”或“MPG”文件，可以通过采用“开始，结束” 的头尾结构来提高读取的速度。方法是 ：①在装入图像前，首先调用本函数一 次，给入的文件名后要加上“,beg”，这时程序仅打开文件并读取图像参数 ；

②然后每顺序读入一幅图像时，调用一次本函数，这时给入的文件名后必须加 上“,con”；

③ 在不需要继续装入图像后，要再调用本函数一次，此时给入的 文件名后要加上“,end”，在此之后程序才关闭文件。

相关函数 ： okSaveImageFile，okPlaybackFromFile。

BOOL WINAPI okGetCurrImageInfo(HANDLE hBoard, MLONG lpImgFrmInfo, MLONG lSize);

功能：和 okLoadImageFile 配合使用，获得最后用 okLoadImageFile 载入的图像文件的额外信息。
参数：
hBoard ： 输入卡句柄。
lpImgFrmInfo:是 EXTRAFRMINFO 结构体的指针，返回的额外的信息放在 EXTRAFRMINFO 结构体里面，结构体的定义可参见 OKAPI32.H 。
lSize：是 EXTRAFRMINFO 结构体的的大小。
返回值：如果 lpImgFrmInfo 非 0，成功返回 1，不成功返回 0。
说明 ：对于 mjpg 压缩的 avi 文件（硬件压缩的或者是驱动是 2011 年 11 月 17 日以后的用 ok 函数保存的 avi 文件），其中会包含额外的压缩该文件时的年月日等信息，如果需要的话，可以在解压文件的时候将其读取出来。先用 okLoadImageFile 函数解压一帧，然后用 okGetCurrImageInfo 得到这一帧的额外信息，用来记录或显示。
相关函数 ： okLoadImageFile。

（10）配置文件读写

BOOL WINAPI okLoadConfigFile(HANDLE hBoard, LPSTR szFileName);

功能：读入指定的 OK 卡配置文件（后缀名指定为“.OKF”），该文件存有各参数的设置值。并按配置文件中的各参数值设置当前卡。
参数 ： hBoard ： 输入卡句柄。
szFileName ： 配置文件名。
返回值 ： 如果成功返回 TRUE，否则返回 FALSE。
说明 ： 当用户已接过某一视频设备，且调好参数后存了 OK 卡配置文件，则可直接读 入该配置文件，使当前卡自动进行同样的设置，而不必再逐一调试。
相关函数 ： okSaveConfigFile，okLoadImageFile，okLoadInitParam。

BOOL WINAPI okSaveConfigFile(HANDLE hBoard, LPSTR szFileName);

功能：把当前卡设置的各参数值(在 11/07/11 ver 4.0 之后还包括硬件压缩参数）存成特定格式的独立硬盘文件，后缀名指定为“.OKF”。该文件称为 OK 卡参数配置文件。
参数 ： hBoard ： 输入卡句柄。
szFileName ： 配置文件名。
返回值 ： 如果成功返回 TRUE，否则返回 FALSE。
说明 ： 当用户对某一所接视频设备调好参数后，可将当前参数存成一独立文件，以备再遇到同类设备时直接使用，而不必再逐一调试。
相关函数 ： okLoadConfigFile，okSaveImageFile，okSaveInitParam。

BOOL WINAPI okLoadInitParam(HANDLE hBoard, SHORT iChannNo);

功能 ： 将指定通道号的初始化参数装入当前卡，并用其初始化卡。
参数 ： hBoard ： 输入卡句柄。
iChannNo ： 要装入初始化参数的指定通道号，以 1 为起始号。
返回值 ： 成功返回 1。

说明 ：装入指定通道号的初始化参数后，该通道即作为当前使用的参数通道。当调用 okCloseBoard 退出时，函数会自动将当前设置的参数存入该通道号。在调用 okOpenBoard 时，程序会自动装入零号通道的初始化参数来初始化卡，用户如需 要选用别的通道号的参数，可调用本函数。
相关函数 ：okSaveInitParam，okSetVideoParam，okSetCaptureParam。

BOOL WINAPI okSaveInitParam(HANDLE hBoard, SHORT iChannNo);

功能 ：将当前设置的参数存入指定通道号，作为该通道的初始化参数。
参数 ：hBoard ：输入卡句柄。
iChannNo ：指定存放初始化参数的通道号，以 1 为起始号。
返回值 ：成功返回 1。
说明 ：存当前的参数到指定的通道号后，该通道同时也作为当前使用的参数通道。当调用 okCloseBoard 退出时，函数会自动将当前最新设置的参数存入当前通道号。
在调用 okOpenBoard 时，程序会自动装入零号通道的初始化参数来初始化卡，用户如需要选用别的通道号设置参数，可调用函数 okLoadInitParam 来完成。
相关函数 ：okLoadInitParam，okSetVideoParam，okSetCaptureParam。

（11）信号参数

INT32 WINAPI okGetSignalParam(HANDLE hBoard, INT32 wParam);

功能 ：获得指定信号 (如场头、外触发等) 的当前状态参数。
参数 ：hBoard ：输入卡句柄。
wParam ：指定要获得信号选项，所支持的项目如下（可参见 OKAPI32.H 中的宏定义）。

SIGNAL_VIDEOEXIST(1)：	测试输入信号是否存在。返回0 不存在，返回1 存在。
SIGNAL_VIDEOTYPE(2)：	测试输入信号的类型，返回0 为逐场（非隔行），1为隔行。
SIGNAL_SCANLINES(3)：	测试输入信号的每帧扫描行数。
SIGNAL_LINEFREQ(4)：	测试计算输入信号的行频（扫描行数/秒）。
SIGNAL_FIELDFREQ(5)：	测试计算输入信号的场频（场数/秒）。
SIGNAL_FRAMEFREQ(6)：	测试计算输入信号的帧频（帧数/秒）。
SIGNAL_EXTTRIGGER (7)：	检测外触发输入信号状态。返回0 无触发，返回1 有触发。
SIGNAL_FIELDID (8)：	检测输入信号奇偶场（0：奇场，1：偶场）。
SIGNAL_VIDEOCOLOR (9)：	检测输入信号是否彩色。（0：黑白， 1：彩色）
SIGNAL_TRANSFERING (10)：	检测图像数据传输状态。1：传输中， 0，传输结束
SIGNAL_CAPFINISHED (11)：	检测是否采集图像完成。返回1，表明已采集完成；并自动清零。返回0，无采集完成。
SIGNAL_HORZSYNCTIME（12）	水平同步头的时间（ns）
SIGNAL_VERTSYNCTIME（13）	垂直同步头的时间（ns）

SIGNAL_SYNCPOLARITY（14）	B0：水平方向的同步极性，B1:垂直方向的同步极性，0是负，1是正。
SIGNAL_HORZVALPOS（15）	DVI信号水平同步沿的极性和水平有效区的宽度，LOWORD 是宽度，HIWORD是沿的极性（0：负极性，1：正极性）
SIGNAL_VERTVALPOS（16）	DVI信号垂直同步沿的极性和垂直有效区的高度，LOWORD 是高度，HIWORD是沿的极性（0：负极性，1：正极性）
SIGNAL_VIDEOCHANGED（17）	自从上一次调用okgetsignalParam之后，视频信号是否改变过
SIGNAL_NETCONNECT（18）	网络设备（或USB设备）连接与否的状态，0：没有连接，1：有连接
SIGNAL_FIELDFREQEX（19）	测试计算输入信号的场频（场数/1000秒）。
SIGNAL_FRAMEFREQEX（20）	测试计算输入信号的帧频（帧数/1000秒）。
SIGNAL_CHECKNETRATE（21）	测试每秒钟网络的传输速率并返回，低字是数据传输速率（M Bytes/S），高字是摄像机的丢帧率（帧/10000 帧）。
SIGNAL_SCANPIXELS(22)	horizental scan pixels per line (only for digital signal)每行的水平扫描点数（只针对数字信号）

注：支持多路外触发的卡（如 MC30 支持 8 路，MC30 只能通过第一块（逻辑）卡来测试），
其返回值的各位中，位 0 对应 1 路，位 1 对应 2 路，以此类推，1 为有触发，0 为 无触发。
返回值 ： 如果指定的项目或参数不被支持，返回 -1 ；如果失败则返回 -2。如果成功则返回该项目当前的参数值。
说明 ： 本函数一般要花一些时间测试输入信号，然后返回。
相关函数 ： okWaitSignalEvent , okSetVideoParam, okSetCaptureParam。

INT32 WINAPI okWaitSignalEvent(HANDLE hBoard, INT32 wParam, MLONG lMilliSecond);

功能 ： 等待指定事件 (如外触发等) 的到来。
参数 ： hBoard ：输入卡句柄。
wParam ：指定要等待的信号事件，所支持的事件如下（可参见 OKAPI32.H 中的宏定义）。
* EVENT_FIELDHEADER(1) ： 等待场头信号。
* EVENT_FRAMEHEADER(2) ： 等待帧头信号。
* EVENT_ODDFIELD(3) ： 等待奇场信号。
* EVENT_EVENFIELD(4) ： 等待偶场信号。
* EVENT_EXTTRIGGER(5) ： 等待外触发信号。
lMilliSecond ：最大等待时间。
返回值 ： 如果指定的事件不被支持，返回 -1 ；如果事件没有等到返回 0，如果事件等到则返回 1。
说明 ： 如果最大等待时间 lMilliSecond=0，函数立即返回当前事件是否到来 ；如果最大等待时间 lMilliSecond=-1（INFINITE），函数将永远等待下去直到该事件到来。
相关函数 ： okGetSignalParam , okSetVideoParam, okSetCaptureParam。

MLONG WINAPI okPutSignalParam(HANDLE hBoard, INT32 wParam, MLONG lParam);

功能 ： 设置要输出到其他设备上的信号 (触发等) 的参数。
参数 ： hBoard ： 输入卡句柄。
wParam ： 指定要输出的信号选项，所支持的项目如下（可参见 OKAPI32.H 中的宏定义）。
* PUTSIGNAL_TRIGGER(1) ： 触发信号。
lParam 的低字：输出一个触发信号。
lParam 的高字：输出一个软件外触发信号。
位 bit0~bit3 依此对应产生触发的 第 1 到第 4 路。
注：以上功能只有 MC20，MC30 支持。MC30 要通过第一块（逻辑）卡来发送。
* PUTSIGNAL_VERTSYNC(2) ： 输出场同步，仅 OK_PC10A 和 GPIO 卡有效。
* PUTSIGNAL_SYNCSTART(3) ： 设定在采集触发之后开始采集的起始行数，仅对线阵卡有效。
* PUTSIGNAL_SYNCCOUNT (4) ： 设定在采集触发之后自采集起始行开始采集的行数，仅对线阵卡有效。
* PUTSIGNAL_DELAYSYNC（5）：设置同步脉冲从基准时间后延迟多少 us 再开始输出，针对相机、CCU。
* PUTSIGNAL_SYNCWIDTH（6）：设置通过 PUTSIGNAL_DELAYSYNC 设置的同步脉冲信号的宽度（以 us 为单位），针对相机、CCU。当 bit31 = 1 的时候，输出的是正脉冲。
* PUTSIGNAL_STARTCONSYNC（11）： 触发以上设置的同步信号的发出。
lParam 的低字是同步信号的持续时长（以 ms 为单位），就是在这个时间段会一直有之前设置的同步信号输出，从基准时间开始算起（对 CCU 有效）。
lParam 的高字（通常都是 0）是特定的外触发后的超长的基准时间后的延迟时间（以 ms 为单位）
* PUTSIGNAL_SETVOLTRANGE（12）： 设置输出电压的幅值（比如 ibs，针对 CCU 和千兆网相机），以 0.1V 为单位。
lParam 的低字是最小值，
lParam 的高字是最大值。
lParam=-2 时为读取输出电压可设置的范围，返回值的低字是最小值，返回值的高字是最大值。
lParam=-1 时为读取相机当前设置的输出电压范围，返回值的低字是最小值，返回值的高字是最大值。

* PUTSIGNAL_FEEDBACKFACT（13）：设置或读取 CCU 的反馈功能或者带有 IBS 功能相机的灵敏度参数，范围从 1 到 16，数值越大反馈越快和灵敏。
lParam=-1 时为读取相机当前设置的灵敏度。
* PUTSIGNAL_SETFLASHMODE（8） 设置闪光灯控制方式，针对 IC1200 摄像头。
lParam 的低字 = 0：自动（默认）；
lParam 的低字 = 1：通过外触发控制；
lParam 的低字 = 2：由 DSP 控制；
lParam 的低字 = 3：一直亮。
lParam 的高字 ： 闪光间隔 1~65535。
* PUTSIGNAL_EXTRIGDELAY（7）设置 GPIO 卡接收到输出触发命令后的延迟时间（以 10us 为单位），延迟后会产生一个下降沿触发（大约 20 微秒）。
- PUTSIGNAL_SETGPIODIR（9）设置 GPIO 卡的输入或输出。
lParam 的 bit0~bit31 对应 32 个 IO 端口。0 表示 输入；1 表示输出。
lParam=-1：函数返回硬件输入和输出分别支持多少位

的数据，lParam 的低字对应输入，lParam 的高字对应输出。

* PUTSIGNAL_SETGPIOLEV（10）设置 GPIO 卡和 OK_IM2366 的高低电平。 lParam 的 bit0~bit31 对应 32 个 IO 端口。0 表示 输入；1 表示输出。（OK_IM2366 支持 3 位）

0 为低电平；

1 为高电平。

lParam=-1：返回当前状态，其他情况下都是要输出的值。

* PUTSIGNAL_SETLENS（14）设置镜头参数，低字节 LOBYTE(BYTE0)为光圈,目前是 0 到 15。

* PUTSIGNAL_SETOPTFILTER（15）设置光学滤光器的参数，LOBYTE(BYTE0)是设置滤光片开关，=0：不使用滤光片，=1：使用滤光片。

lParam ： 对应 wParam 的参数值。

返回值 ： 如果指定的项目或参数不被支持，返回 -1。

相关函数 ： okWaitSignalEvent , okSetVideoParam, okSetCaptureParam

2.查询卡信息

INT32 WINAPI okGetImageDevice(OKDEVTYPE **lpOkDevInfo, MLONG lParam);

功能 ： 查寻以正确安装、连接的 OK 系列图像设备的信息，包括：PCI、PCI-E、PCI-X 等总线，USB 接口和 GigE 接口。

参数 ：

lpOkDevInfo：输入指向 OK 系列图像卡信息结构指针的地址。调用返回后该指针指向当前 OK 系列图像卡信息结构数组。结构 OKDEVTYPE 含有卡的类型码和类型名称，详见 OKAPI32.H 中的定义。

lParam：是输入、输出参数的指针。输入时：b0=1，重新检测；b1=1，包含网络设备。输出时：返回 Ok Image Manager 中是否选择了包含网络设备的选项。

返回值 ： 返回已正确安装的 OK 系列图像设备数量。如果无，返回零。

说明 ： 本函数可以查到所有 OK 系列图像设备。当 Ok Image Manager 中选中了包含网络设备的选项或者 lParam 指定搜索时包含网络设备的时候返回包含网络设备在内的的所有 OK 设备的总数；如果没有指定搜索时包含网络设备，则函数返回不包含网络设备在内的设备总数。如仅需要获得当前卡数，可设 lpOkDevInfo =Null，lParam =0。

相关函数 ： okGetSlotBoard， okGetBoardIndex，okGetTypeCode，okGetBoardName。

SHORT WINAPI okGetSlotBoard(BOARDTYPE **lpOkInfo);

功能 ： 查寻机器插槽中已正确插入的 OK 系列图像卡信息。

参数 ：lpOkInfo ：输入指向 OK 系列图像卡信息结构指针的地址。调用返回后该指针指 向当前 OK 系列图像卡信息结构数组。结构 BOARDTYPE 含有卡的类型
码和类型名称，详见 OKAPI32.H 中的定义。

返回值 ： 返回已正确安装的 OK 系列图像卡数量。如果无，返回零。

说明 ： 本函数只可查到 OK 系列图像卡。如仅需要获得当前卡数，可设 lpOkInfo=Null。

相关函数 ： okGetBoardIndex，okGetTypeCode，okGetBoardName，okGetImageDevice。

SHORT WINAPI okGetBoardIndex(CHAR *szBoardName, SHORT iNo);

功能 ：获取指定卡型及同类型卡的顺序号对应机器中已正确插入的 OK 系列图像卡的索引（顺序）号。
参数 ： szBoardName ： 输入指定卡型的类型码或型号名（都是以字符串形式）。
iNo ： 输入指定卡在机器中已正确插入的同类型卡的顺序号。
返回值 ： 返回所指定卡的索引（顺序）号。如果没有查到，返回 -1。
说明 ： 本函数只可针对 OK 系列图像卡。
相关函数 ： okGetSlotBoard，okGetTypeCode，okGetBoardName。

INT32 WINAPI okGetBoardName(SHORT lIndex, LPSTR szBoardName);

功能 ： 获得指定卡名称。
参数 ： lIndex ： 输入指定卡在当前已连接的所有 OK 卡中的顺序索引号。
szBoardName ： 调用返回后，返回指定卡名称。
返回值 ： 若函数调用成功则返回指定卡类型码，若没有卡则返回 0。
相关函数 ： okGetSlotBoard ，okGetBoardIndex，okGetTypeCode。

INT32 WINAPI okGetTypeCode(HANDLE hBoard, LPSTR lpBoardName);

功能 ： 获得指定句柄对应卡的类型码以及卡型号（字符串）。
参数 ： hBoard ： 输入卡句柄。
lpBoardName ： 如果输入字符串指针，返回当前句柄对应的设备型号名称（字符串）。
如果输入特殊数值，函数返回相应的信息。
返回值 ： 如果调用成功，函数返回当前句柄对应的类型码。
lpBoardName =NULL 时，函数只返回类型码
lpBoardName=-1 时，函数返回当前卡是几代卡。
lpBoardName=-2 时，函数返回识别码。
lpBoardName=-3 时，函数返回主版本号和次版本号。
lpBoardName=-4 时，函数返回所有设备中的序列号;
lpBoardName=-5 时，函数返回 IP 值。
lpBoardName=-6 时，函数返回高位 IP 值。
lpBoardName=-7 时，函数返回硬件逻辑版本号。
lpBoardName=-8 时，函数返回设备类型（=0：PCI、PCI-E、PCI-X；=1：USB；=2：1394；=3：GigE）。
lpBoardName=-9 时，函数返回当前硬件设备是否支持 64 位地址的采集（=0：不支持； =1：支持）。
lpBoardName=-10 时，函数返回线扫描设备。
说明 ： 各卡的类型码定义参见 OKAPI32.H。
相关函数 ： okOpenBoard, okGetBoardIndex，okGetBoardName。

INT32 WINAPI okGetNetDevNumber(INT32 iTypeCode, LPSTR lpNameString);

功能：通过类型码或卡型号（字符串）查询并获得网络设备的数量。
参数：iTypeCode ： 设备的类型码。
lpNameString ： 设备的名称的字符串指针。
返回值：返回 0 说明没有找到 ok 的网络设备，否则返回以 1 为起始的序列号。
相关函数：okGetImageDevice，okOpenBoard，okCloseBoard，okGetLastError。
说明：函数的两个参数至少给其一。

3.查询与锁定缓存

MLONG WINAPI okGetBufferSize(HANDLE hBoard, void **lpLinear,MWORD *dwSize);

功能 ：获得本卡作为图像数据使用的静态缓存 BUFFER 的线性首地址和大小，及在当前设置下可存放图像的幅数（含有动态缓存块的部分）。如未通过 okLockBuffer 进行锁定过，则返回的也是整个缓存的线性基地址。
参数 ：hBoard ：输入卡句柄。
lpLinear ：如 lpLinear 不为 NULL，返回线性基地址。
dwSize ：如 dwSize 不为 NULL，返回本卡可用存图像的大小（以字节为单位）。对于支持屏蔽位的卡，返回的大小不含屏蔽位所用缓存部分。
返回值：如果调用成功，返回缓存（含有动态缓存块部分）可存图像的帧数(在当前缓存宽、高、格式设置下)。否则返回 0（FALSE）。
说明 ：输入参数 lpLinear 与 dwSize 均可以 NULL 指针作为输入，此时将不返回该参数的值。因而用户也可用此函数仅查询当前可用的总帧数。如果要查询设备驱动申请的静态缓存的大小，应调用函数 okGetAvailBuffer。
相关函数 ：okGetAvailBuffer ，okGetBufferAddr，okGetTargetSInfo，okSetCaptureParam。

LPVOID WINAPI okGetBufferAddr(HANDLE hBoard, MLONG lNoFrame);

功能 ：获得序列缓存（含有动态缓存块部分）中指定帧的线性地址。
参数 ：hBoard ：输入卡句柄。
lNoFrame ：指定在序列帧缓存中的帧序号。
返回值：如果调用成功，返回在当前图像帧大小设置（包括宽，高和字节数 / 象元）下， 指定帧的线性地址。否则返回 0（FALSE）。
说明 ：如用户需要直接对缓存进行处理，要用此函数来得到指定帧的线性地址，而不能直接根据起始地址推算。
相关函数 ：okGetBufferSize，okGetTargetSInfo，okSetCaptureParam。

MLONG WINAPI okGetAvailBuffer(void **lpLinear, MWORD *dwSize);

功能 ：获得静态缓存信息，或可锁定使用的静态缓存信息。
参数 ：lpLinear ：如输入非 NULL，调用返回后该指针指向剩余静态缓存的线性首地址。
dwSize ：如输入非 NULL，则返回当前静态缓存的字节数，如未进行过锁定， 则返回的就是申请成功的静态缓存的总大小。
返回值 ：返回剩余缓存的物理首地址，如静态缓存未申请成功，则返回 0。
说明 ：本函数可用来查询系统启动后已申请成功的静态缓存的信息，或需要多卡同时工作时。当用户需要各卡使用不同的缓存区时，可用本函数查询当前可供锁定的剩余缓存情况。
相关函数 ：okGetBufferSize，okLockBuffer，okUnlockAllBuffer。

MLONG WINAPI okLockBuffer(HANDLE hBoard, MWORD dwSizeByte,void **lpBasLinear);

功能 ：锁定指定大小的静态缓存给本卡专用。
参数 ：hBoard ：输入卡句柄。
dwSizeByte ：非零时，该值就是指定要锁定缓存的大小。等于零时，为第二种锁缓存模式。
lpBasLinear ：dwSizeByte 非零时，输入一指针的地址，调用返回后该指针指向锁定缓存 的

线性首地址。模式二，本变量为一顺序存放了指定缓存起始位置和 大小的数组，即 MLONG dwLockBufPo[2]。
返回值 ： 返回实际锁定的缓存大小。
说明 ： 本函数是为需要多卡同时工作而提供的。当用户需要各卡使用不同的缓存区时， 可用本函数第一种模式锁定的所需大小的缓存区给本卡使用，使以后打开的卡不能再使用已锁定的缓存区，而只能使用剩余的缓存部分。
注意 ： 对于支持采集屏蔽位的卡，其屏蔽位要占用一定的缓存，标准信号采集卡占用的大小为
256X768。用第二种模式则完全由用户指定起始和大小，各卡之间可以相互独立，也可以互相重叠。
相关函数 ： okGetBufferSize，okUnlockAllBuffer。

BOOL WINAPI okUnlockAllBuffer(void);

功能 ： 解锁所有锁定的静态缓存，使整个缓存重新公用。同时也使缓存可以重新锁定。
参数 ： 无。
返回值 ： TRUE
说明 ： 本函数是为需要多卡同时工作而提供的。当用户进行了多次锁定解锁，而使缓存区碎块化后，分不到所需的缓存区，而需要重新分配时，或放弃所有的锁定而使缓存都共用时，可以通过调用本函数来实现全部解锁。
相关函数 ： okGetBufferSize，okLockBuffer。

4.多卡同步采集

MLONG WINAPI okMulCaptureTo(HANDLE *lphBoard,TARGET Dest, MLONG start, MLONG lParam);

功能 ： 控制多块卡同时采集到目标体，基本功能与 okCaptureTo 相同。
参数 ： lphBoard ： 多个输入卡的句柄指针，须以零句柄为结束。
Dest ： 要采集到的目标体，可以是 SCREEN，BUFFER 或 FRAME。目标体的宏定义参见 OKAPI32.H。
start ： 采集到目标体的起始帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN，该值只能为 0。
lParam ： 采集帧数或采集方式。详见函数 okCaptureTo。
返回值 ： 如果调用成功，返回该目标体所支持的最大帧数。如果失败（如由于格式不支持等）返回 0（FALSE）。如果该目标体不被支持则返回 -1。
说明 ： 此函数为完全硬件支持的采集。要求各卡的信号源为同步的。并且要分别为每个卡锁定一块缓存。其它详见函数 okCaptureTo 中的说明。由于各卡是同步的，
此时如设置回调只有零号句柄是有效的。
相关函数 ： okCaptureTo, okMulCaptureByBuffer, okStopCapture。

HANDLE WINAPI okMulCaptureByBuffer(HANDLE *lphBoard, TARGET tgt, MLONG start, MLONG num);

功能 ： 控制多块卡同时采集到各自的 BUFFER，再顺序送到目标体，基本功能与 okCaptureByBuffer 相同。
参数 ： lphBoard ： 多个输入卡的句柄指针，以零句柄为结束。
tgt ： 要采集到的目标体，可以是 SCREEN、BUFFER、窗口句柄数组、用户内存指针数组、或含有用户内存指针的多个 BLOCKINFO 结构变量的指针，或中间用单空格分开的多个文

件名的字符串指针（如 :“AA.BMP BB.BMP“）。BLOCKINFO 结构的定义参见 OKAPI32.H。
start ： 采集到目标体的起始帧序号（起始为 0）。
num ： 要采集的帧数，当该值大于零时，将从 start 开始直到采集完 num 帧为止。 如采到用户内存 MEMORY，该值必须小于用户内存可存放的最大帧数。当该值等于零时，将连续采集到 start 位置，直到通过调用 okStopCapture 终止为止。
返回值 ： 如果调用成功，返回该目标体所支持的最大帧数。如果失败（如由于格式 不支持等）返回 0（FALSE）。如果该目标体不被支持则返回 -1。
说明 ： 要求各卡的信号源为同步的，并且各卡已锁定好自己专用的缓存。其它详见函数 okCaptureByBuffer 中的说明。
相关函数 ： okMulCaptureTo, okCaptureByBuffer, okStopCapture。

5. 专项功能

此类函数为专有硬件支持的函数，不是各种卡型都支持。

（1）回显输出

MLONG WINAPI okPlaybackFrom(HANDLE hBoard, TARGET src,MLONG start, MLONG lParam);

功能 ： 从指定源目标体回显到视频模拟监视器（MONITOR）。这里的目标体可以是帧缓存（BUFFER），帧存体（FRAME）。可以单帧回显也可以进行多帧回显，可以不等回显结束立即返回，也可以等序列回显结束再返回。
参数 ： hBoard ： 输入卡句柄。
src ：要回显的源目标体，可以是 BUFFER 或 FRAME。目标体的宏定义参见 OKAPI32.H。
start ： 回显的源目标体的起始帧序号（起始为 0），对于只有一帧的目标体， 如 FRAME，该值只能为 0。
lParam ： 回显帧数或方式。
1，该值如果大于零（>0），即为回显源目标体的帧数，此时为序列回显方式。如果 lParam 大于源目标体的最大帧数（Total），当逐帧回显源目标体的位置超过 Total 时，将重置回到起始（序号 0）位置继续开始逐帧回显，如此按方式 mode(n%total) 回显，直到回显完 lParam 帧为止，并停留在连续回显最后一帧。在特例该值 =1 时，即等于一直回显 start 指定的一帧图像，**注意**，此种特例下无回调支持。
2，如果小于零 (<0) 即为循环序列回显方式，即从 start 指定 的位置开始逐帧回显源，当达到 abs（lParam）代表的帧数或源目标体的最大帧数（Total）时将回到起始（序号 0）位置继续开始逐帧回显， 如此无限循环下去，直到通过调用 okStopCapture 才会停止。
返回值 ： 如果调用成功，返回该回显源目标体所支持的最大帧数。如果失败（如由于格式不支持等）返回 0（FALSE）。如果该目标体不被支持则返回 -1。
说明 ： 本函数只对硬件支持视频回显功能的卡有效（如 ： M40A，M60A，M30 等），并且目标体的格式应是该卡回放功能所支持的，如不是所支持的格式，回放时图像就是乱的。
此函数为完全硬件支持的回显。当为循环序列回显方式，或序列回显但为不等结束返回方式时，本函数起动一序列回显线程然后立即返回调用它的程序，如果需要终止正在进行中的回显过程，可随时通过调用 okStopCapture 终止该过程。当为单帧或序列回显，并且为等待结束返回方式时，则回显全部完成后才会返回调用它的程序。通过调用函数 okSetCaptureParam 的 CAPTURE_ SEQCAPWAIT 来设置单帧回显或序列回显时，是不等结束返回方式还是等结束返回方式。
相关函数 ： okCaptureTo, okPlaybackByBuffer。

HANDLE WINAPI okPlaybackByBuffer(HANDLE hBoard, TARGET src, MLONG start, MLONG num);

功能 ：间接（通过帧缓存 BUFFER）从指定源目标体回显到视频模拟监视器（MONITOR）。这里的源目标体只可以是用户内存 MEMORY 或文件 FILE。可以单帧回显也可以进行多帧回显，可以不等回显结束立即返回，也可以等序列回显结束再返回。

参数 ： hBoard ： 输入卡句柄。

src ： 要回显的用户内存指针，或含有用户内存指针的 BLOCKINFO 结构变量指针，或文件名字符串指针。BLOCKINFO 结构的定义参见 OKAPI32.H。

start ： 回显的源目标体的起始帧序号（起始为 0）。

num ： 要回显的帧数。

1，该值如果大于零（>0），则为回显源目标体的帧数，将从位置 start 开始，逐帧回显源目标体直到回显完 num 帧为止。并停留在连续回显最后一帧。

2，如果小于零 (<0) ，则为循环序列回显方式，即从 start 指定的位置开始逐帧回显源，当达到 abs（num）代表的帧数时将回到起始（序号 0）位置继续开始逐帧回显，如此无限循环下去，直到通过调用 okStopCapture 才会停止。

注意 ：（1）num 的绝对值必须小于等于源目标体的帧数。（2）本间接回显函数支持回显时的回调函数。

返回值 ： 如果调用成功，返回线程句柄，否则返回 0（FALSE）。

说明 ： 本函数只对硬件支持视频回显功能的卡有效（如 ： M40A，M60A，M30 等）。 此函数为间接回显，它是通过把内存或文件的图像传送到两帧帧缓存来完成回显的。如果需要终止正在进行中的采集过程，可随时通过调用 okStopCapture 终 止该过程。本函数启动一序列回显进程。

注意 ：如果回显源目标体是 BLOCKINFO 结构变量指针或文件，帧缓存 BUFFER 的格式将被改变为与源目标体一样，大小不变。如果回显源目标体是用户内存指针，则回显帧存的大小和位数取帧缓存 BUFFER 当前的设置值。

如果输入是 BLOCKINFO 结构变量指针，结构变量中的各元素必须正确填写， 如 iType=BLKHEADER，lpBits 为用户内存指针。

如果输入的是”.SEQ”或”.AVI”文件名，该函数将顺序回显该序列文件中的图像。如果指定的是”.BMP”或”.JPG”文件名，则自动生成以该文件为第一个文件的顺序序列编号的”.BMP”或”.JPG”文件名，然后顺序回显各文件中的图像。

BLOCKINFO 结构的定义和序列文件 SEQ 的格式参见 OKAPI32.H ； BMP、AVI 及 JPG 文件格式参见有关资料。

当为循环序列回显方式，或序列回显但为不等结束返回方式时，本函数起 动一序列回显线程然后立即返回调用它的程序，如果需要终止正在进行中的回 显过程，可随时通过调用 okStopCapture 终止该过程。当为单帧或序列回显，并 且为等待结束返回方式时，则回显全部完成后才会返回调用它的程序。通过调用函数 okSetCaptureParam 的 CAPTURE_SEQCAPWAIT 来设置单帧回显或序列回显时，是不等结束返回方式还是等结束返回方式。

相关函数 ： okPlaybackFrom, okLoadImageFile, okGetCaptureStauts, okStopCapture, okCaptureTo。

BOOL WINAPI okPlaybackSequence(HANDLE hBoard,MLONG lStart, MLONG lNoFrame);

功能 ： 中断控制方式从指定缓存 (BUFER) 回显到视频模拟监视器（MONITOR）。可以单帧回显也可以进行多帧回显，不等回显结束立即返回。

参数 ：

hBoard ： 输入卡句柄。

lStart ： 回显的源目标体的起始帧序号（起始为 0）。

lNoFrame ： 要回显的帧数。图像

1，该值如果大于零（>0），则为回显缓存的帧数，将从位置 lStart 开始，逐帧回显源目标体直到回显完 lNoFrame 帧为止。并停 留在连续回显最后一帧。

2，如果小于零 (<0) ，则为循环序列回显方式，即从 lStart 指定的位置开始逐帧回显源，当达到 abs（lNoFrame）代表的帧数时 将回到起始（序号 0）位置继续开始逐帧回显，如此无限循环下去， 直到通过调用 okStopCapture 才会停止。

注意 ：（1）lNoFrame 的绝对值必须小于等于源目标体的帧 数。（2）本回显函数不支持回显时的回调函数。

返回值 ： 如果调用成功，返回 1（TRUE），否则返回 0（FALSE）。

说明 ： 本函数只对硬件支持视频回显功能的卡有效（如 ：M40，M60，M30 等）。如果需要终止正在进行中的采集过程，可随时通过调用 okStopCapture 终止该过程。

相 关 函 数 ： okPlaybackFrom, okLoadImageFile, okGetCaptureStauts, okStopCapture, okCaptureTo, okPlaybackByBuffer。

（2）采集屏蔽

INT32 WINAPI okEnableMask(HANDLE hBoard, BOOL bMask);

功能 ： 允许或禁止采集屏蔽。

参数 ： hBoard ： 输入卡句柄。

bMask ： 0 禁止采集屏蔽，1 允许正向模式采集屏蔽，2 允许负向模式采集屏蔽。

返回值 ： 返回之前的屏蔽模式。

说明 ： 本函数只对硬件支持屏蔽功能的卡有效。所谓正向模式采集屏蔽，即屏蔽位为 1 的区域，采入视频输入信号，屏蔽位为 0 的区域，保留原有数据（如在 VGA（SCREEN）上的 WINDOWS 菜单等），而不被视频输入所覆盖。负向模式采集屏蔽与之相反，即屏蔽位为 0 的区域采入视频输入信号，屏蔽位为 1 的区域，保留原有数据，而不被视频输入所覆盖。

相关函数 ： okSetMaskRect , okCaptureTo。

INT32 WINAPI okSetMaskRect(HANDLE hBoard, LPRECT lpRect, LPBYTE lpMask);

功能 ： 设置屏蔽窗口与屏蔽位。

参数 ： hBoard ： 输入卡句柄。

lpRect ：设置采集目标屏蔽窗口位置。窗口座标是相对采向目标体所设置的窗 口左上角的。

lpMask ： 设置采集目标屏蔽位图。其格式为 ： 每象元对应一字节（置 0 或 1）， 一行的宽度等于所设置 lpRect 的宽度 （lpRect->right-lpRect->left）， 逐行顺序存放。

如果屏蔽位图是简单矩形，也可直接设 lpMask=1，等于对应 lpRect 所设窗口的矩形区域的屏蔽位都是 1 ；设 lpMask=0，等于对应 lpRect 所设窗口的矩形区域的屏蔽位都是 0。

返回值 ： 对于不支持屏蔽位的卡，返回零。支持的卡返回非零。

说明 ：本函数只对硬件支持屏蔽功能的卡有效。所谓正向模式采集屏 蔽，即屏蔽位为 1 的区域，采入视频输入信号，屏蔽位为 0 的区域，保留原有数据（如在 VGA（SCREEN）上的 WINDOWS 菜单等），而不被视频输入所覆盖。负向模式采集屏蔽与之相反，即屏蔽位为 0 的区域采入视频输入信号，屏蔽位为 1 的区域，保留原有数据，而不被视频输入所覆盖。

相关函数 ： okEnableMask , okCaptureTo。

（3）查找表控制

INT32 WINAPI okFillOutLUT(HANDLE hBoard, LPVOID bLUT, INT32 start, INT32 num);

功能 ： 填写视频输出查找表。
参数 ： hBoard ： 输入卡句柄。
bLUT ：存放 RGB 输出查找表数据，格式为 r0,g0,b0; r1,g1,b1; ...。对于 8 位查找表，bLUT 应为字节指针，每个数据为 8 位。对于大于 8 位的查找表， 如 10 位、12 位等。bLUT 应为字指针，每个数据相应为 10 位，或 12 位等。设 bLUT=NULL，不填写查找表，仅用来查询支持否 ；通过输出值可判断是否支持，及是多少位的查找表。
start ： 填写查找表数据 bLUT 到输出查找表的起始位置。范围 0~255。
num ：填写查找表数据 bLUT 的项数（每项对应三个字节，顺序为 R，G，B）， 范围 1~256。
返回值 ： 返回 -1 ：不支持。填写查找表时，成功后返回 1。查询时返回支持的表位数， 如 8，即 8 位查找表，10，即 10 位查找表等。
说明 ： 本函数只对硬件支持视频回显输出查找表功能的卡有效（如 ：M40A，M60A，LV40A，CL40A 等等）。打开图像卡时，系统缺省初始化为正常灰度查找表。
相关函数 ： okPlaybackFrom, okPlaybackByBuffer，okFillInputLUT。

INT32 WINAPI okFillInputLUT(HANDLE hBoard, LPVOID bLUT, INT32 start, INT32 num);

功能 ： 填写视频输入查找表。
参数 ： hBoard ： 输入卡句柄。
bLUT ：存放 RGB 输出查找表数据，格式为 r0,g0,b0; r1,g1,b1; ...。对于 8 位查找表，bLUT 应为字节指针，每个数据为 8 位。对于大于 8 位的查找表， 如 10 位、12 位等。bLUT 应为字指针，每个数据相应为 10 位，或 12 位， 等。设 bLUT=NULL，不填写查找表，仅用来查询支持否 ；通过输出值可判断是否支持，及是多少位的查找表。
start ： 填写查找表数据 bLUT 到输入查找表的起始位置。范围 0~255。
num ：填写查找表数据 bLUT 的项数（每项对应三个字节，顺序为 R，G，B）， 范围 1-256。
返回值 ： 返回 -1 ：不支持。填写查找表时，成功后返回 1。查询时返回支持的表位数，如 8，即 8 位查找表，10，即 10 位查找表等。
说明 ：本函数只对硬件支持视频输入查找表功能的卡有效（如 ：M10B，M10K，M20A，M20B，M40A，M50A，M60A，LV20A 等等）。打开图像卡时，系统缺省初始化 为正常灰度查找表。
相关函数 ： okFillOutLUT。

（4）串口操作

INT32 WINAPI okSetSerial(HANDLE hBoard, INT32 wParam, INT32 lParam);

功能：设置设备上有内部串口连接设备的串口配置参数。
hBoard ： 输入卡句柄。
wParam：要设置的项目：
SERIAL_COMMINDEX（0）当前想要进行读写操作的串口号，默认值是 0。
SERIAL_BAUTRATE（1） 串口的波特率（比如 38400, 19200, 9600, 4800, 2400, 1200...) 默认值是 9600。

SERIAL_DATABITS（2） 数据长度（bit）比如 8,9...默认值是 8。
SERIAL_STOPBITS（3）停止位长度，为 0 是 1bit；为 1 是 1.5bit,为 2 是 2bit, 默认值是 0
SERIAL_PARITY（4） 校验位，为 0 无校验；为 1 是奇校验，为 2 是偶校验,默认值是 0
lParam：相应项目的值。如果为-1：只返回当前设置值。
返回值 ：返回之前该项设置的值。如果要设置的项不支持返回-1；如果项目支持但设置的时候错误返回-2.
相关函数 ：okReadSerial，okWriteSerial
说明：该函数主要用于 camaralink 接口的卡或者 CCU 上，因为该卡的接口中同时包含串口协议，可以控制和它相连的摄像头等设备。

INT32 WINAPI okReadSerial(HANDLE hBoard, LPVOID lpBuffer, INT32 lSize, MLONG lTimeOut);

功能：从当前卡上连接的串口设备读取数据。
hBoard ：输入卡句柄。
lpBuffer：存放读数据的缓存的指针。
lSize：读数据的缓存的大小。
lTimeOut：超时等待时间。
返回值 ：返回实际读出了多少字节。
相关函数 ：okSetSerial，okWriteSerial
说明：该函数主要用于 camaralink 接口的卡或者 CCU 上，因为该卡的接口中同时包含串口协议，可以控制和它相连的摄像头等设备。

BOOL WINAPI okWriteSerial(HANDLE hBoard, LPVOID lpBuffer, INT32 lSize, MLONG lTimeOut);

功能：将数据写入当前与卡连接的串口设备。
hBoard ：输入卡句柄。
lpBuffer：要写入数据的缓存指针。
lSize：写入数据的缓存的大小。
lTimeOut：超时等待时间。
返回值 ：如果成功返回非 0，失败返回 0。
相关函数 ：okSetSerial，okReadSerial
说明：该函数主要用于 camaralink 接口的卡或者 CCU 上，因为该卡的接口中同时包含串口协议，可以控制和它相连的摄像头等设备。

（5）平场校正

BOOL WINAPI okSaveFlatModelFile(HANDLE hBoard, LPSTR szFile);

功能：从 BUFFER 的第 0 帧和 BUFFER 的第 1 帧中保存当前平场校正模板数据到文件。
hBoard ：输入卡句柄。
szFile：文件名字符串指针，保存的文件名后缀必须是“.fla”的。
返回值：成功返回 1，失败返回 0。
相关函数 ：okLoadFlatModelFile，okSetCaptureParam
说明：平场校正模板可以在 okSetCaptureParam 函数的操作下生成，并且生成后马上就可以使用，但是硬件并不能掉电之后依然保存校正模板，当再上电使用的时候还想要用这个功能的话就要重新生成模板，生成模板会浪费一定的时间和精力，而且大部分情况下客户使用的

环境都是相对固定的，只要最初生成一次模板以后都用其来校正图像就可以了，因此就需要将最开始生成的模板数据保存下来，以后打开卡之后将参数从新装载进硬件进行校正，该函数就是用来实现保存模板的功能。

BOOL WINAPI okLoadFlatModelFile(HANDLE hBoard, LPSTR szFile);

功能：从计算机指定位置装载平场校正模板文件到 BUFFER 的第 0 帧和 BUFFER 的第 1 帧中，同时将数据设置到硬件以备平场校正功能时使用。
hBoard ：输入卡句柄。
szFile：文件名字符串指针，指定的文件名后缀必须是“.fla”的。
返回值：成功返回 1，失败返回 0。
相关函数 ：okSaveFlatModelFile，okSetCaptureParam

6.实用函数

（1）对话框

BOOL WINAPI okOpenSetParamDlg(HANDLE hBoard, HWND hParentWnd);

功能 ：打开设置各视频输入和采集控制参数的模式对话框。
参数 ：hBoard ：输入卡句柄。
hParentWnd ：输入父窗口句柄。
说明：在调用此函数之前，通过调用 SetWindowLong 函数，对其中的 GWL_USERDATA 参数进行设置，可以改变设置参数打开之后的模式等信息。
如果 GWL_USERDATA 的 bit0 = 1：打开的设置参数对话框为无模式对话框，函数返回打开的对话框的句柄，不过此方式会导致打开的对话框失去焦点，需要客户自己把焦点从新设置到对话框上；bit1 = 1：新打开的对话框会直接进入实时显状态；bit2 = 1：打开的窗口不允许最小化。
返回值 ：如果成功，返回 1（TRUE）或窗口句柄，否则返回 0（FALSE）。
相关函数 ：okOpenSeqCaptureDlg，okOpenReplayDlg，okOpenReplayDlgEx

BOOL WINAPI okOpenSeqCaptureDlg(HANDLE hBoard，HWND hParentWnd);

功能 ：打开可用来序列采集 / 回显的模式对话框。
参数 ：hBoard ：输入卡句柄。
hParentWnd ：输入父窗口句柄。
返回值 ：如果成功，返回 1（TRUE），否则返回 0（FALSE）。
相关函数 ：okOpenSetParamDlg，okOpenReplayDlg，okOpenReplayDlgEx。

BOOL WINAPI okOpenSetLUTDlg(HANDLE hBoard, HWND hWnd, LPVOID lpLUT);

功能：如果当前选用的卡支持查找表功能，打开可以用来设置输入输出查找表的模式对话框。
参数：hBoard ：输入卡句柄。
hWnd：输入父窗口句柄

lpLUT：当点击 OK 关闭对话框时用来存放查找表数据的内存指针，如果不需要查找表的具体数据可以给 NULL。
返回值 ：如果成功，返回 1（TRUE），否则返回 0 （FALSE）。
说明：该函数只对有查找表功能的卡起效，查找表的内容对采集的图像是有影响的，可以是线性、线性负像、γ 矫正等。
相关函数 ：okFillOutLUT，okFillInputLUT。

LPDIBINFO WINAPI okOpenReplayDlg(HANDLE hBoard,HWND hWnd, TARGET src, MLONG total);

功能 ：打开可用来回放序列图像的无模式对话框。当通过对话框调换当前帧图像时，本函数将自行把当前帧的图像显示在窗口 hWnd 的用户区左上角。
参数 ：hBoard ：输入卡句柄。
hWnd ：输入父窗口句柄。
src ：序列图像源目标体，可以是帧缓存 BUFFER，用户内存 MEMORY，或文件 FILE。
total ：源目标体存放序列图像的帧数。
返回值 ：如果成功，返回带有对话框信息的 DIBINFO 结构指针。否则返回 0（FALSE）。关于 DIBINFO 结构，参见头文件 OKAPI32.H。
说明 ：可以通过此打开的无模式对话框控制序列图像的逐帧前进放，后退放，任意帧显示。如果序列图像源目标体是以 BLOCKINFO 结构形式给出的用户内存。
MEMORY，或文件 FILE，则帧缓存的格式将会按源目标体的改变而改变。如源目标体就是用户内存指针，则函数把其图像格式和大小默认为当前帧缓存的设置。
相关函数 ：okOpenSetParamDlg，okOpenSeqCaptureDlg，okOpenReplayDlgEx。

HWND WINAPI okOpenReplayDlgEx(HANDLE hBoard,HWND hWnd, TARGET src, MLONG total, LPBITMAPINFOHEADER lpbi, LPBYTE lpdib);

功能 ：打开可用来回放序列图像的无模式对话框。当通过对话框调换当前帧图像时，本函数将把新的图像数据拷贝到 lpdib，并向用户窗口 hWnd 发送刷新消息 WM_ PAINT，由用户自己的程序决定显示位置，并自己实现当前帧图像的刷新显示。
参数 ：hBoard ：输入卡句柄。
hWnd ：输入父窗口句柄。
Src ：序列图像源目标体，可以是帧缓存 BUFFER，用户 MEMORY，或文件 FILE。
Total ：源目标体存放序列图像的帧数。
lpbi: 用于显示的 BITMAPINFOHEADER 结构的指针。
lpdib: 用于显示的图像数据指针。
返回值 ：如果成功，返回对话框窗口句柄。否则返回 0（FALSE）。
说明 ：可以通过此打开的无模式对话框控制序列图像的逐帧前进放，后退放，任意帧显示。如果序列图像源目标体是以 BLOCKINFO 结构形式给出的用户内存。
MEMORY，或文件 FILE，则帧缓存的格式将会按源目标体的改变而改变。如源目标体就是用户内存指针，则函数把其图像格式和大小默认为当前帧缓存的设置。
相关函数 ：okOpenSetParamDlg，okOpenSeqCaptureDlg，okOpenReplayDlg。

（2）统计分析

float WINAPI okGetFocusMeasure(HANDLE hBoard, TARGET target, MLONG start, LPRECT lpRect, SHORT iMethod, SHORT iPreDegree);

功能：通过算法判断当前图像的对焦清晰程度。

参数：
hBoard ：输入卡句柄。
Target:：数据源目标体，可以是 GRAPH, SCREEN, BUFFER, FRAME,和 BLOCKINFO，目标体的宏定义参见 OKAPI32.H。
lStart ：目标体的帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN,该值只能为 0。
lpRect：要用于测量图像内容的感兴趣区域。
iMethod：选择判断的依据，iMethod=0: 灰度差分；
=1: 罗伯特；
=2: 4X 拉普拉斯；
=3: 8X 拉普拉斯；
=4: brenner
iPreDegree：预处理方法, iPreDegree=0: 无预处理；
=1: 3x3 十字滤波；
=2: 3x3 矩形滤波；
=3: 5x5 矩形滤波；
=4: 7*7 矩形滤波。
返回值：函数返回当前图像的清晰度测量值，值越大图像越清晰，对焦越好。
说明：该函数不支持虚拟卡。

INT32 WINAPI okMultiFrmMean(HANDLE hBoard, TARGET target, MLONG start, MLONG number);

功能：对选定的数据区中的数据进行多帧平均运算，将计算结果存放到数据源的 lStart 中。
参数：
hBoard ：输入卡句柄。
Target：数据源目标体，可以是 GRAPH, SCREEN, BUFFER, FRAME,和 BLOCKINFO，目标体的宏定义参见 OKAPI32.H。
lStart ：目标体的起始帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN,该值只能为 0。
Number：要参与平均运算的帧数。
返回值：成功返回 1，当不支持的时候返回 -1。

INT32 WINAPI okTakeRectMean(HANDLE hBoard, TARGET target, LPRECT lpRect, MLONG start, short iMode);

功能： 计算并返回指定区域的灰度值。
参数：
hBoard ：输入卡句柄。
Target：目的目标体。
lpRect：是在当前的区域中要进行计算时的感兴趣区域，如果所有的当前范围都要进行计算则此参数给 NULL。
start：目标体的帧序号（起始为 0）。
iMode：给 0。
返回值：成功返回目标区域的灰度平均值，不支持返回-1。
说明：本函数不支持虚拟卡，如果是彩色图像，会分别统计。则返回值的低字节是蓝路的平均值，次低字节是绿路的平均值，高字是红路的平均值。
相关函数 ：okEvaluateHistogram、okGetLinExtenMaplut、okGetHistEquaMaplut。

INT32 WINAPI okEvaluateHistogram(HANDLE hBoard, TARGET target, LPRECT lpRect, MLONG start, LPDWORD lpHist);

功能： 对指定区域进行直方图数据统计。
参数：
hBoard ：输入卡句柄。
Target：目的目标体。
lpRect：是在当前的区域中要进行计算时的感兴趣区域，如果所有的当前范围都要进行计算则此参数给 NULL。
start：目标体的帧序号（起始为 0）。
lpHist：直方图指针。它的大小必须大于目标体的灰度级。如果目标体中存放的是彩色图像，那么函数会将图像中的 R、G、B 分量的值分别统计，然后将他们的直方图的数据按照 B、G、R 的顺序存放在 lpHist 中。即目标体是 GRAY8 的图像，那么申请的内存大小必须要大于 256*sizeof(DWORD)字节，如果目标体是 RGB888 的图像，那么申请的内存大小必须要大于 256*3*sizeof(DWORD)字节。
返回值：如果调用成功则返回直方图的信息。返回值的低字是直方图统计的位数（如 8,10,16…），返回值的高字是直方图统计的通道数（如是黑白图像为 1，彩色图像为 3）。
说明：本函数不支持虚拟卡。
相关函数 ：okTakeRectMean、okGetLinExtenMaplut、okGetHistEquaMaplut。

INT32 WINAPI okGetLinExtenMaplut(LPDWORD lpHist, short iNumBits, short iNumChann, short iIntensity, LPWORD lpwLUT);

功能： 根据图像直方图统计的数据得到要对图像线性拉伸的查找表。
参数：
lpHist: 指向图像直方图统计后的数据的指针。
iNumBits：图像位数。
iNumChann：图像通道数（为 1 或 3）。
iIntensity：增强程度（0：无变化，10：最强）。
lpwLUT: 内存区指针，调用成功时用于存放查找表数据。它的大小必须大于目标体的灰度级。即目标体是 8 位深的图像，那么申请的内存大小必须要大于 256*sizeof(WORD)字节。
返回值：如果成功返回值大于 0，并会将计算后的查找表数据放到 lpwLUT 中，失败返回 0。
说明：如果直方图数据是彩色图像的，3 个通道同时给入，函数会对三个通道进行综合计算，得出相应的查找表。如果用户想要对彩色图像的三个分量进行分别的处理，可以将直方图的数据指针分别指向相应分量的起始位置，通道数给 1 即可。
相关函数 ：okTakeRectMean、okEvaluateHistogram、okGetHistEquaMaplut。

INT32 WINAPI okGetHistEquaMaplut(LPDWORD lpHist, short iNumBits, short iNumChann, LPWORD lpwLUT);

功能： 根据图像直方图统计的数据得到要对图像进行直方图均衡的查找表。
参数：
lpHist: 指向图像直方图统计后的数据的指针。
iNumBits：图像位数。
iNumChann：图像通道数（为 1 或 3）。
lpwLUT: 内存区指针，调用成功时用于存放查找表数据。它的大小必须大于目标体的灰度级。即目标体是 8 位深的图像，那么申请的内存大小必须要大于 256*sizeof(WORD)字节。
返回值：如果成功返回 1，并会将计算后的查找表数据放到 lpwLUT 中，失败返回 0。
说明：如果直方图数据是彩色图像的，3 个通道同时给入，函数会对三个通道进行综合计算，得出相应的查找表。如果用户想要对彩色图像的三个分量进行分别的处理，可以将直方图的数据指针分别指向相应分量的起始位置，通道数给 1 即可。
相关函数 ：okTakeRectMean、okEvaluateHistogram、okGetLinExtenMaplut。

（3）图像处理

BOOL WINAPI okUnifyFields(HANDLE hBoard, TARGET target, MLONG start);

功能：使奇偶两场的数据亮度一致。
参数：
hBoard ：输入卡句柄。
Target ：数据源目标体，可以是 GRAPH, SCREEN, BUFFER, FRAME,和 BLOCKINFO，目标体的宏定义参见 OKAPI32.H。
start ：目标体的帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN,该值只能为 0。
返回值 ：成功返回 1。
说明：当有的客户需要用闪光灯又不能保证闪光灯和信号源同步的时候，如果是隔行信号源可能会导致图像的奇场和偶场的亮度不一致，图像看起来会有条纹，如果用这个函数把图像处理一下就可以得到亮度均匀的图像了。

INT32 WINAPI okGuassFilter(HANDLE hBoard, TARGET target, MLONG start);

功能：通过高斯滤波器去除图像上的噪声。
参数：
hBoard ：输入卡句柄。
Target:：数据源目标体，可以是 GRAPH, SCREEN, BUFFER, FRAME,和 BLOCKINFO，目标体的宏定义参见 OKAPI32.H。
lStart ：目标体的帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN,该值只能为 0。
返回值：如果成功返回 1，不成功返回-1。
说明：该函数用于去除高斯噪声，但是图像的边缘也会相应的变模糊，不支持虚拟卡。

INT32 WINAPI okDiffusionFilter(HANDLE hBoard, TARGET target, MLONG start, SHORT nLoop);

功能：通过梯度扩散法去除图像上的噪声。
参数：
hBoard ：输入卡句柄。
Target:：数据源目标体，可以是 GRAPH, SCREEN, BUFFER, FRAME,和 BLOCKINFO，目标体的宏定义参见 OKAPI32.H。
lStart ：目标体的帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN,该值只能为 0。
nLoop：迭代的次数，应大于 0，推荐用 2。
返回值：如果成功返回 1，不成功返回-1。
说明：该函数用于去除噪声，相对于高斯滤波，图像的边缘会得到更好的保留，不支持虚拟卡。

INT32 WINAPI okSharpFilter(HANDLE hBoard, TARGET target, MLONG start, SHORT iClass);

功能：通过边缘增强算法使图像的边缘锐化，图像边缘更清晰。
参数：
hBoard ：输入卡句柄。
Target:：数据源目标体，可以是 GRAPH, SCREEN, BUFFER, FRAME,和 BLOCKINFO，目标体的宏定义参见 OKAPI32.H。

lStart ： 目标体的帧序号（起始为 0），对于只有一帧的目标体，如 SCREEN,该值只能为 0。
iClass：锐化滤波的级数，可以是 1~4，1 是锐化程度最小，4 是最大。
返回值：如果成功返回 1，不成功返回-1。
说明：该函数用于图像的边缘增强，不支持虚拟卡。

INT32 WINAPI okSetCustomFilter(HANDLE hBoard, LPRECT lpRect, WORD wChann, WORD *wKernSize, LPWORD lpwKernel, SHORT *iDivisor, SHORT *iOffset);

功能：设置用户自定义的滤波模板（IM2366 支持）。
参数：
hBoard ： 输入卡句柄。
lpRect:: 图像有效区域，lpRect=-1: 关闭滤波模板； =-2：只获得当前模板的参数（3*3，5*5）。同时 lpwKernel 会返回当前模板里的数值，函数的返回值为 1 表示当前此滤波模板是启用状态，0 表示关闭状态。
wChann：选择哪个通道的图像需要被处理，0 是所有通道都处理；1、2、3 分别对应 R、G、B 通道。
wKernSize：低字节（LOBYTE）是模板 X 方向的大小，高字节（HIBYTE）是模板 Y 方向的大小。
lpwKernel：自定义的模板参数的指针。
iDivisor：一般是 0，是模板运算后结果的除数。
iOffset：一般是 0，是结果的偏移值。
返回值：如果成功返回 1，不支持返回-1。
说明：该函数用于图像的自定义处理，不支持虚拟卡。

（4）图形操作

BOOL WINAPI okSetTextTo(HANDLE hBoard, TARGET target, LPRECT lpRect, LOGFONT *lfLogFont, SETTEXTMODE*textmode LPSTR lpString, INT32 lLength);

功能 ： 把字符串以指定的字体图形写到目的体内，目的体可以是 GRAPH，SCREEN，BUFFER，FRAME，或 BLOCKINFO 指定的内存区。
参数 ：
hBoard ： 输入卡句柄。
target ： 写向的目的体。
lpRect ： 写向目的体的指定区域的起始位置，用其中的（left,top）变量来控制。
lfLogFont ： 定义逻辑字体的结构变量，详见 WINDOW 的 API 说明。
textmode：指定输入字符的前景和背景色，和设置方式，详见 okapi32.h 中的定义。
lpString ： 指定输入的字符串指针。
lLength ： 输入字符串的长度。
返回值 ： 返回 1。
相关函数 ： okDrawEllipsTo, okWriteRect, okTransferRect, okConvertRect。

INT32 WINAPI okFillCircleTo(HANDLE hBoard, TARGET target, MLONG lStart, LPRECT lpRect, INT32 lForColor, INT32 iMode);

功能：填充任意大小的圆形（或椭圆形）到指定目标体，目标体可以是 GRAPH，SCREEN，BUFFER，FRAME，或 BLOCKINFO 指定的内存区，可以填充圆形的外侧或内侧。

参数：
hBoard ： 输入卡句柄。
Target:写向的目的体，可以是 GRAPH，SCREEN，BUFFER，FRAME，或 BLOCKINFO 指定的内存区。
lStart ： 采集到目标体的指定帧号。
lpRect ： 指定（椭）圆形的外切矩形坐标，圆形的直径取 rect 值最大的方向。
lForColor：所画（椭）圆形颜色的值。
该值的 b7~b0 是 B 的值
b15~bb8 是 G 的值
b23~b16 是 R 的值。
iMode：b0=0: 表示把值填充到（椭）圆形外面的区域;
=1: 把值填充到（椭）圆形里面的区域;
b1=0: 填充圆形区域;
=1: 填充椭圆形区域。
返回值 ： 返回具体填充数据的象元数。
相关函数 ： okSetTextTo, okWriteRect, okTransferRect, okConvertRect。

INT32 WINAPI okDrawLineTo(HANDLE hBoard, TARGET target, MLONG lStart, POINT ptS, POINT ptE, INT32 iForeColor);

功能 ：画一直线（含起始和结束点）到指定的目的体内，目的体可以是 GRAPH，SCREEN，BUFFER，FRAME，或 BLOCKINFO 指定的内存区。
参数 ：
hBoard ： 输入卡句柄。
target ： 写向的目的体。
lStart: ： 目标体的指定帧号。
ptS ： 所画直线的起始点坐标。
ptE ： 所画直线的结束点坐标。
iForeColor ： 所画直线的颜色的值。该值的 b7~b0 是 B 的值，b15~bb8 是 G 的值，b23~b16 是 R 的值。
返回值 ： 返回所画直线象元数。
相关函数 ： okSetTextTo, okWriteRect, okDrawEllipsTo, okConvertRect。

INT32 WINAPI okDrawEllipsTo(HANDLE hBoard, TARGET target, MLONG lStart, LPRECT lpRect, INT32 iForeColor);

功能 ：画一（椭）圆到指定的目的体内，目的体可以是 GRAPH，SCREEN，BUFFER，FRAME，或 BLOCKINFO 指定的内存区。
参数 ： hBoard ： 输入卡句柄。
target ： 写向的目的体，
lStart: ： 目标体的指定帧号，
lpRect ： 所画椭圆的外切矩形坐标，
iForeColor ： 所画椭圆的颜色的值。该值的 b7~b0 是 B 的值，b15~bb8 是 G 的值，b23~b16 是 R 的值。
返回值 ： 返回所画椭圆的象元数。
相关函数 ： okSetTextTo, okWriteRect, okTransferRect, okConvertRect。

HDC WINAPI okCreateDCBitmap(HANDLE hBoard, TARGET target, HANDLE *hDCBitmap);

功能 ： 创建一个 WINDOWS 的 GDI 兼容的与目的体大小相同的内存 DC，并返回与

DC 所对应的存放位图等内容的句柄。目的体可以是 SCREEN，BUFFER， FRAME， 或 BLOCKINFO 指定的内存区。
参数 ：hBoard ：输入卡句柄。
target ： 写向的目标体，
hDCBitmap: 返回存放位图等内容的 DC 位图句柄，用于把 GDI 函数所写的内容映射到目的体。
返回值 ：返回 GDI 兼容的内存 DC。
说明 ：通过该函数创建的 DC 可以利用各种 WINDOWS 的 GDI 函数，如画图、写字符等等， 写某种内容到与指定目的体大小相同的位图里，然后再通过 okMapDCBitmapTo 转换到目的体中。
相关函数 ：okMapDCBitmapTo, okFreeDCBitmap, okSetTextTo, okDrawEllipsTo。

BOOL WINAPI okMapDCBitmapTo(HANDLE hDCBitmap, MLONG lStart);

功能 ：映射由 okCreateDCBitmap 创建的与 DC 和目标体兼容的位图句柄的内容（非零的部分）到目标体的指定的图像帧。对应零值的区域则维持原值不变。
参数 ：hDCBitmap: DC 位图句柄，
lStart: 目标体的指定帧号。
返回值 ：返回非零。
说明 ：通过该函数，可以将各种 WINDOWS 的 GDI 函数所写的图形等，转到指定目的体里。
相关函数 ：okCreateDCBitmap, okFreeDCBitmap, okSetTextTo, okDrawEllipsTo。

BOOL WINAPI okFreeDCBitmap(HANDLE hDCBitmap);

功能 ：释放由 okCreateDCBitmap 创建的与 DC 和目标体兼容的位图句柄等资源。
参数 ：hDCBitmap: DC 位图句柄。
返回值 ：返回 1。
相关函数 ：okCreateDCBitmap, okMapDCBitmapTo, okSetTextTo, okDrawEllipsTo。

（5）自动设置参数

INT32 WINAPI okAutoSetCapRect(HANDLE hBoard, INT32 Width, INT32 Height);

功能：根据当前信号信息自动设置所有采集参数。
参数：hBoard ：输入卡句柄。
Width：指定图像的宽度，在不知道的时候可以给 0。
Height：指定图像的高度，在不知道的时候可以给 0。
返回值 ：如果信号源有信号，成功返回 INT32 型变量，其中的 HIWORD 是图像的高度，LOWORD 是图像的宽度，失败返回 0。
说明 ：对于一些 VGA 信号，我们的卡默认的参数不能直接采集到合适的图像，当客户想要直接调整参数使图像卡可以直接采集到的相应参数来采集图像的时候，用这个函数把所有的参数自动设置后，可以直接得到基本上合适的图像，省却了每个参数都要设置的麻烦。
相关函数 ：okGetSignalParam。

INT32 WINAPI okAutoAdjustBright(HANDLE hBoard, LPRECT lpRect, LPVOID dwRes);

功能：自动设置亮度、对比度，增益和饱和度（如果可以设置）使图像达到最好的视觉效果。
参数：hBoard ： 输入卡句柄。
LpRect:是在当前的缓存区中要进行亮暗自动设置时的感兴趣区域，必须小于当前 BUFFER 的 rect，并且是一块大于 8*8 的范围。如果所有的当前 BUFFER 范围都要进行亮暗自动调节则此参数给 NULL。
dwRes：应该给 0，保留参数，暂时没有用。
返回值：如果成功，返回自动设置之后图像的平均灰度值，在 0~255 之间。
说明：当用户对当前图像感觉不满意时，如果直接手动调节各种参数会比较麻烦，可以通过这个函数进行自动调节以达到较好的图像效果，该函数在图像采集卡上调整效果明显，对亮度、对比度不可调的设备无效。

（6）编码与解码

HANDLE WINAPI okBeginEncode(HANDLE hBoard, DWORD dwCodeWay, MLONG lpImageInfo);

功能 ：启动指定编码方式的编码，输入相应编码方式的输入参数。
参数 ：hBoard ： 输入卡句柄。
dwCodeWay ： 输入指定的编码方式，定义参见下表。
lpImageInfo ： 指定编码方式输入参数对应的结构指针，如下表所示，结构具体定义参见头文件 OKAPI32.H。

编码方式	相应结构体
CODE_JPEG（1）	JPEGPARAM
CODE_MJPG（3）	MJPGHEADER
CODE_MPEG4（4）	MPEG4HEADER
CODE_JPG2K（5）	JPG2KHEADER
CODE_H264（6）	H264HEADER

返回值 ：如果成功，返回编码器句柄，否则返回 0（FALSE）。
说明 ：目前该函数支持 JPEG、MJPEG，MPEG4，JPEG2000 编码方法。
相关函数 ：okEncodeImage，okEndEncode，okBeginDecode。

INT32 WINAPI okEncodeImage(HANDLE hCoder, TARGET src, MLONG start, LPBYTE lpData, MLONG maxlen);

功能 ：指定一幅图像，对其进行编码。
参数 ：hCoder ： 输入编码器句柄。
Src ： 指定数据源，可以是 SCREEN，BUFFER，或含有用户内存指针的 BLOCKINFO 结构变量指针。BLOCKINFO 结构的定义参见 OKAPI32.H。
Start ： 指定数据源的第几帧。
lpData ： 输入用来存放编码数据的内存区指针。
maxlen ： 指定输入内存区的最大长度，以字节为单位。
返回值 ：如果成功，返回编码数据的实际长度，以字节为单位。
说明 ：目前该函数支持 JPEG、MJPEG，MPEG4，JPEG2000 编码方法。
相关函数 ：okBeginEncode，okEndEncode，okDecodeImage。

INT32 WINAPI okEndEncode(HANDLE hCoder)；

功能 ：结束编码，并释放编码器资源。
参数 ：hCoder ： 输入编码器句柄。

返回值 ：如果成功，返回 1。
说明 ：目前该函数支持 JPEG、MJPEG，MPEG4，JPEG2000 编码方法。
相关函数 ：okBeginEncode, okEncodeImage, okEndDecode。

HANDLE WINAPI okBeginDecode(HANDLE hBoard, DWORD dwCodeWay, LPBYTE lpData, MLONG lpImageInfo);

功能 ：启动指定编码方式的解码，并获得编码数据的参数。
参数 ：hBoard ：输入卡句柄。
dwCodeWay ：输入指定的编码方式，定义参见下表。
lpData ：输入含有编码头的编码数据。
lpImageInfo ：指定解码方式输入参数对应的结构指针，如果其不为 NULL ，输出编码数据的原图像参数。对应关系如下表所示，结构具体定义参见头文件 OKAPI32.H。

编码方式	相应结构体
CODE_JPEG（1）	IMAGESIZE
CODE_MPEG2（2）	MPEG2HEADER
CODE_MJPG（3）	MJPGHEADER
CODE_MPEG4（4）	MPEG4HEADER
CODE_JPG2K（5）	JPG2KHEADER

返回值 ：如果成功，返回解码器句柄，否则返回 0（FALSE）。
说明 ：目前该函数支持 JPEG、MJPEG，MPEG2，MPEG4，JPEG2000 解码方法。
相关函数 ：okDecodeImage，okEndDecode，okEncodeImage。

INT32 WINAPI okDecodeImage(HANDLE hCoder, LPBYTE lpData, MLONG *length, TARGET target, MLONG start);

功能 ：对给定的编码数据进行解码。
参数 ：hCoder ：输入解码器句柄。
lpData ：输入存放编码数据的内存指针。
Length ：输入编码数据的长度，输出实际用的长度。
target ：指定输出目的体，可以是 SCREEN，BUFFER，或含有用户内存指针的 BLOCKINFO 结构变量指针。BLOCKINFO 结构的定义参见 OKAPI32.H。
Start ：指定输出到目的体的第几帧。
返回值：如果对已输入的编码数据，解出了一幅图，则返回 1，如果还不够一整幅图， 则返回 0。
说明 ：目前该函数支持 JPEG、MJPEG，MPEG2，MPEG4，JPEG2000 解码方法。
相关函数 ：okBeginDecode，okEndDecode，okEncodeImage。

INT32 WINAPI okEndDecode (HANDLE hCoder) ；

功能 ：结束解码，并释放解码器资源。
参数 ：hCoder ：输入解码器句柄。
返回值 ：如果成功，返回 1。
说明 ：目前该函数支持 JPEG、MJPEG，MPEG2，MPEG4，JPEG2000 解码方法。
相关函数 ：okBeginDecode，okDecodeImage，okEndEncode。

（7）其它

INT32 WINAPI okGetProgramInfo(INT32 iItem, LPSTR lpString, INT32 iSize);

功能：获得驱动程序的版本信息。
参数：iItem：指定想要获得的信息。
lpString ：字符串的指针，用来存放版本号的信息。
iSize：字符串的大小。
返回值：0。

BOOL WINAPI okWaitVerticalSync(HANDLE hBoard, BOOL bHeader);

功能：等待显卡输出的场同步信号。
参数：hBoard ：输入卡句柄。
bHeader：为 0，等待显卡输出场同步信号的开始，为 1，等待场同步信号的结束。
返回值 ：返回 1。
说明：当用经缓存显示的时候有的电脑在显示的时候图像会出现横条似的闪烁，显示的不稳定，用这个函数可以让图像刷新的时候和显卡显示同步，就可以避免图像闪烁，该函数需要显卡硬件支持。
相关函数 ：okSetConvertParam。

INT32 WINAPI okSetLangResource(INT32 langcode);

功能：设置语言类型。
参数：
langcode：语言代码，1252 是英语，936 为简体中文。=-1：返回当前的语言代码。
返回值 ：设置时语言代码。
说明：当客户调用各种实用对话框的时候可能需要显示不同的语言，那么在调用对话框前可以先用此函数把语言设置成英文或简体中文，然后再调用对话框就可以显示相应的语言了。不能识别的语言代码都默认设置成英语。
相关函数 ：okOpenSetParamDlg，okOpenSeqCaptureDlg，okOpenReplayDlg，okOpenReplayDlgEx，okOpenSetLUTDlg。

MLONG WINAPI okSetUserData(HANDLE hBoard, MLONG lUserData);

功能：设置或获得用户的数据到 hboard 句柄中。
参数：hBoard ：输入卡句柄。
lUserData：=-1：不设置数据；其他值时为用户想要在卡句柄中保存的数据。
返回值：返回之前用户设置的数据。

DWORD WINAPI okGetTickCountMicro(LPDWORD lpHiDWord);

功能：和 GetTickCount 函数功能一样，但是可以准确到微妙。
返回值：返回当前的系统时间。

DWORD WINAPI okGetTickCount(void);

功能：和 GetTickCount 函数功能一样，但是在 WIN2K 的系统下会比 GetTickCount 更精确。
返回值：返回当前的系统时间。

void WINAPI okSleepMicro(DWORD dwMicro);

功能：和 Sleep 函数功能一样，以微秒为单位。
参数：
dwMicro 想要 sleep 的微秒数。
返回值：无

void WINAPI okSleep(DWORD dwMill);

功能：和 Sleep 函数功能一样，但是在 WIN2K 的系统下会比 Sleep 更精确。
参数：
dwMill 想要 sleep 的毫秒数。
返回值：无

7.硬件压缩采集

HANDLE WINAPI okOpenStream(HANDLE hBoard, DWORD lParam);

功能 ：打开指定的视频流压缩设备。
参数 ：hBoard ：输入带有视频流压缩设备的图像卡的句柄。
lParam ：正常打开视频流压缩句柄的时候**必须**设为 0；当 bit0 = 1 的时候调用该函数用来确认当前图像采集卡是否支持硬件压缩功能。
返回值 ：如成功，返回视频流压缩设备句柄，否则返回 0。
相关函数 ：okCloseStream, okCaptureStream,okOpenBoard。

BOOL WINAPI okCloseStream(HANDLE hStream);

功能 ：关闭指定的视频流压缩设备。
参数 ：hStream ：输入视频流压缩设备句柄。
返回值 ：返回 TRUE。
说明 ：不再进行视频流压缩时调用此函数关闭视频流压缩设备。
相关函数 ：okOpenStream, okCaptureStream,okCloseBoard。

HANDLE WINAPI okCaptureStream(HANDLE hStream, TARGET target,FARPROC lpfnUserProc, MLONG lMiliSeconds);

功能 ：开始视频流压缩。
参数 ：hStream: 输入视频流压缩设备句柄。
Target: 采集目标可以为 NONE(无)、BUFFER(缓存)或文件名，目前只可为 M2V (MPEG2)或 mjpg 压缩的 AVI 文件格式。
lpfnUserProc: 进行视频流压缩时，用户读取视频流压缩数据的回调函数。当不使用回调函数时该项必须设为 NULL。
lMiliSeconds: 指定要采集压缩的时间，以毫秒为单位，如果是 0，则为无限长采集压缩，直到通过调用 okStopCaptureStream 停止为止。
返回值 ：返回 0，失败 ；返回非 0（句柄） 成功开始压缩。
说明 ：如果采集目标是 NONE(无)或 BUFFER(缓存)，则应设置回调函数通过回调函数将已完成的视频流压缩数据读走，否则后面的数据就会将前面的数据覆盖掉。读取视频流压缩数据的回调函数要按如下方式定义 ：
BOOL CALLBACK WriteStreamProc (HANDLE hStream, LPBYTE lpStreamBuf,

INT32 length);

其中 hStream ：输入视频流压缩设备句柄，lpStreamBuf ：指向可存放读取的视频流压缩数据内存区，length ：为要读取的视频流压缩数据长度，以字节为单位。每当采集到一定数量的数据后，就会调用本回调函数，送出已采集到的数据，length 为数据长度。采集停止后，会再调用一次本回调函数， 这时 length=0，表示采集结束。

相关函数 ： okOpenStream, okCloseStream,WriteStreamProc。

BOOL WINAPI okStopCaptureStream(HANDLE hStream, DWORD dwCmdCode);

功能 ： 停止视频流压缩。

参数 ：

hStream ： 输入视频流压缩设备句柄。

dwCmdCode ： =0 ： 停止采集压缩，

=1 ： 暂停压缩，

=2 ： 恢复压缩。

返回值 ： 如果正在采集，则返回视频流压缩过程中已读出的数据总长度。

相关函数 ： okOpenStream, okCloseStream。

MLONG WINAPI okSetStreamParam(HANDLE hStream, INT32 wParam, MLONG lParam);

功能 ： 设置视频流压缩时的设备参数。

参数 ：

hStream ： 输入视频流压缩设备句柄。

wParam ： 指定设置的项目。参见 okapi32.h。

STREAM_RESETALL (0) ： 重置所有项的参数值为系统缺省值。

STREAM_RESOLUTION (1) ： 用于 C20B、C21B，录像文件的分辨率，

=0 ： 全尺寸大小 (720)，

=1 ： 2/3 全尺 寸 (480)，

=2 ： 1/2 全 尺 寸 (352)，

=3 ： 1/2 全 尺寸 (仅奇场)(352)。

缺省设置为全尺寸大小。

STREAM_BITRATEMODE(2) ： 用于 C20B、C21B，压缩码位率

=0 ： 固定码位率方式 ；

非 0 ： 变码 位率方式（保证压缩质量，为缺省设置）的期望码率。

缺省设置为 3000（kbit/s）。

STREAM_MAXBITRATE (3) ： 用于 C20B、C21B，固定码位率方式的码位率 ；变码位率方式的最大 码位率。缺省设置为 9000（kbit/s）。

STREAM_CALLINTERVAL(4) ： 回调的最小时间间隔（以毫秒为单位），缺省设置为 120（毫秒）。

STREAM_RECTSELECT (5) ： 用于 RGB60C、RGB61C-4E 等，是否使用独立的采集压缩窗口。

=0（缺省设置）：不使用独立窗口，使用和采集函数所设置的相同的窗口；

=1 ： 使用视频流独立的窗口，具体窗口位置由 STREAM_SOURCERECT 来进行设置。

STREAM_RECTSOURCE 是曾用名。

STREAM_SOURCERECT (6) ： 用于 RGB60C、RGB61C-4E 等，独立的窗口参数 RECT 设置，只有当 STREAM_RECTSOURCE 参数为 1 的时候起效。

STREAM_PICKFRMRATE（7）：用于 RGB60C、RGB61C-4E 等，设置每 100 帧图像中取多少帧用来压缩成文件，和信号源的频率一起决定压缩文件的帧率，在硬件压缩达不到实时的情况下，可以均匀的降低压缩文件的帧率。

STREAM_QUALITY（8） 用于 RGB60C、RGB61C-4E 等，设置硬件压缩时图像的质量因子，可以是 10~100, =100: 图像质量最好； =0(default) 表示 50。
STREAM_MAXFILELEN（9）设置文件长度的最大限制，以 MB 为单位。在 ntfs 或 fat32 文件系统下都起效。当文件的长度超过这个数值后驱动会自动进行文件切换，新的文件名是源文件名+“-1.XXX”。如果没有预先设置长度限制，而录制的文件是“AVI”、“seq”文件或系统的文件格式是“FAT32 ”时，如果文件大于 3.5G，驱动也将会自动进行文件切换。
lParam ： 对应 wParam 的参数值。
返回值：如调用成功，返回新设置的值，不支持返回 -1，错误返回 -2。如果 lParam 等于 -1 则不设置新值，而只返回当前设置的值。
说明 ： 可对比参照函数 okSetVideoParam 与 okSetCaptrueParam。
相关函数 ： okOpenStream, okCloseStream。

INT32 WINAPI okReadStreamData(HANDLE hStream, LPBYTE lpStreamBuf, INT32 lReadSize);

功能 ： 读取采集到视频流压缩数据。
参数 ：
hStream ： 输入视频流压缩设备句柄。
lpStreamBuf ： 用户用于保存视频流压缩数据所申请的内存的指针。
lReadSize ： 用户所要读取的视频流压缩数据的长度，以字节为单位。
返回值 ： 返回实际读取的视频流压缩数据长度，以字节为单位。
说明 ： 用户在调用视频流压缩数据函数后，就需要每隔一定时间，通过本函数把已完成的视频流压缩数数据读出来。
相关函数 ： okOpenStream, okCloseStream,okCaptureStream。

8.音频采集

HANDLE WINAPI okOpenAudio(HANDLE hBoard, MLONG lParam);

功能 ： 打开指定的音频采集设备。
参数 ： hBoard ： 输入图像卡的句柄。
lParam ： 正常打开音频采集设备句柄时**必须**设为 0；当 bit0 = 1 的时候调用该函数用来确认当前图像采集卡是否支持音频采集功能。
返回值 ： 当 lParam = 0 时：函数成功，返回音频设备句柄，否则返回 0；当 lParam 的 bit0 = 1 时：设备支持，返回 1，否则返回 0。
说明 ：
相关函数 ： okCloseAudio, okCaptureAudio,okOpenBoard。

BOOL WINAPI okCloseAudio(HANDLE hAudio);

功能 ： 关闭指定的音频采集设备。
参数 ： hAudio ： 输入音频设备句柄。
返回值 ： 返回 TRUE。
说明 ： 不再采集声音时调用此函数关闭采集采集设备。
相关函数 ： okOpenAudio, okCaptureAudio,okCloseBoard。

MLONG WINAPI okCaptureAudio(HANDLE hAudio, TARGET target, FARPROC lpfnUserProc, MLONG lParam);

功能 ： 开始采集声音数据。
参数 ： hAudio ： 输入音频设备句柄。
target ： 采集目标可以为 BUFFER(缓存)，或文件名，目前只可为 .WAV 文件格式。
lpfnUserProc: 采集声音时，用户读取声音数据的回调函数，当用户不使用回调函数时必须设为 NULL。
lParam ： 保留参数，必须设为 0。
返回值 ： 音频缓存的时间长度，就是音频设备的内部缓存最大可以存放声音数据的时间长度，以毫秒为单位。
说明 ： 如果采集目标是 BUFFER，则在音频缓存的时间长度内，必须通过回调函数，或函数 okReadAudioData 将声音数据读走，否则后面的数据就会将前面的数据覆盖掉。 读取声音数据的回调函数要按如下方式定义 ：
BOOL CALLBACK WriteAudioProc(HANDLE hAudio, LPBYTE lpAudBuf, INT32 length);
其中 hAudio ： 输入音频设备句柄，lpAudBuf ： 指向可读取的采集到的声音数据内存区，length ： 为要读取的声音数据长度，以字节为单位。 每当采集到一定数量的声音数据后，就会调用本回调函数，送出已采集到的声音数据，length 为数据长度。采集停止后，也会调用本回调函数，这时 length=0。
相关函数 ： okOpenAudio, okCloseAudio,WriteAudioProc。

BOOL WINAPI okStopCaptureAudio(HANDLE hAudio);

功能 ： 停止采集声音。
参数 ： hAudio ： 输入音频设备句柄。
返回值 ： 返回采集过程中读出的音频数据总长度。
说明 ：
相关函数 ： okOpenAudio, okCloseAudio。

INT32 WINAPI okSetAudioParam(HANDLE hAudio,INT32 wParam,\ INT32 lParam);

功能 ： 设置声音采集的参数。
参数 ： hAudio ： 输入音频设备句柄。
wParam ： 指定设置的项目。参见 okapi32.h。
AUDIO_RESETALL (0) ： 重置所有项的参数值为系统缺省值 。
AUDIO_SAMPLEFRQ (1) ： 采样频率，即每秒采样次数。采样频率越高，声音越真实，相应的数据量也越大。可任意设置，常见的 有 8000，11025，22050，44100 等。
AUDIO_SAMPLEBITS (2) ： 采样数据位数，即 bits 数。可设为 8 或 16，16bits 采样的声音效果强于 8bits，但数据量上升一倍。lParam=-2:获得当前信号源的数据位数。
AUDIO_INVOLUME (3) ： 输入音量控制，调节值范围为 0~255。音频输入不可太大，否则会由于失真而产生噪音。
AUDIO_CALLINTERVAL （4）设置回调的时间间隔，以毫秒为单位。
AUDIO_SOURCECHAN（5） 当有多个信号源时，lParam 的高字是指定采集的信号源，lParam 的低字选择指定的信号源中多个声道中的某一个（中文、英文）。（暂不使用，不发布）
AUDIO_SOUNDCHAN（6） 信号源是多通道的时候，lParam 的低字设置采集指定通道的数据，bit0：左声道(单声道)；bit1：右声道；bit2：中间声道；bit3：左后声道；bit4：右后声道；bit5：重低音…。lParam 的高字返回信号源本来支持的通道数。 lParam = -1，返回当前信号源的通道数。
AUDIO_SIGNALINFO（7）用来或得信号源是否存在，b0=1: 检测到了信号源。
lParam ： 对应 wParam 的参数值。
返回值：如调用成功，返回新设置的值，不支持返回 -1，错误返回 -2。如果 lParam 等于 -1 则不设置新值，而只返回当前设置的值。
说明 ： 可对比参照函数 okSetVideoParam 与 okSetCaptrueParam。

相关函数 ： okOpenAudio, okCloseAudio。

INT32 WINAPI okReadAudioData(HANDLE hAudio, LPBYTE lpAudioBuf, INT32 lReadSize);

功能 ： 读取采集到的声音数据。
参数 ： hAudio ： 输入音频设备句柄。
lpAudioBuf ： 用户用于保存声音数据所申请的内存的指针。
lReadSize ： 用户所要读取的声音数据的长度，以字节为单位。
返回值 ： 返回实际读取的声音数据长度，以字节为单位。
说明 ： 用户在调用采集音频数据函数后，就需要在音频缓存的时间长度内，也就是在音频缓存数据满之前，通过本函数把音频数据读出。
相关函数 ： okOpenAudio, okCloseAudio,okCaptureAudio。

9.系统控制

MLONG WINAPI okSetDriverParam(INT32 lWhich, MLONG lParam);

功能：设置驱动的参数。
参数：lWhich：指定设置项目，所支持的项目如下（可参见 OKAPI32.H 中的宏定义）。

STATICVXD	1 静态加载 vxd
INSTNTDRV	2 安装驱动
ALLOCBUFFER	3 申请缓存（会调用 okSetAllocBuffer）
MEMBLOCKSIZE	4 内存快的大小
MEMBLOCKCOUNT	5 内存块的数量
UNREGISTER	6 卸载

lParam：根据设置项目的不同给入相应的参数。
说明 ： 如果 lParam = -1（GETCURRPARAM），则仅返回该项目当前的参数值。
相关函数 ： okSetStaticVxD

MLONG WINAPI okSetAllocBuffer(MLONG lSize);

功能 ： 设定或获得希望申请的缓存大小。
参数 ： lSize ： 希望设定的缓存大小。
如果 = -1：则只获得之前的设定；
= -2：获得当前的系统的最大可用内存；
= -num：表示要申请|num|大小，并且修改后不提示对话框。
返回值 ： 新设定前的设定大小，如 lSize =-1，则仅返回之前的设定值。
说明 ： 若 lSize 为正数，函数调用成功，并且与之前的设定值不同，则会提示要求重新启动 Windows。
相关函数 ： okSetStaticVxD, okSetNTDriver。

BOOL WINAPI okSetStaticVxD(INT32 lMode);

功能 ： 设定 VxD 驱动的加载模式，只适用于 WIN95/98/ME。
参数 ： lMode ： 驱动加载模式。
1，该值为 0 时，查寻驱动安装的状态，当驱动程序为静态加载时返回 TRUE，否则返回 FALSE。
2，该值为 1 时，设定当前的驱动程序加载模式为静态加载。

3，该值为 2 时，删除驱动静态加载的模式。
4，该值为 10 时，功能同值为 1 时一样，但不弹出要求重启动系统的提示。
返回值 ：函数调用成功则返回 TRUE，或函数调用失败则返回 FALSE。
说明 ：当加载模式为静态加载时，无论计算机中是否插有 ok 系列图像采集卡，Windows 启动时均会将驱动程序加载入内存，也就是说，无论是否插有图像卡 Windows 均会分配相应的物理内存给驱动程序。当动态加载时，仅当计算机中插有 ok 系 列图像采集卡时，Windows 才会将驱动程序加载入内存。该函数调用后，需要 重新启动 Windows 后才会生效。
相关函数 ：okSetAllocBuffer, okSetNTDriver。

BOOL WINAPI okSetNTDriver(BOOL bCmd);

功能 ：设置安装驱动程序，只适用于 WINNT4/2K/XP。
参数 ：bCmd ：驱动加载模式。
1. 该值为 0 时，查寻驱动安装与否，如驱动程序安装好，返回 TRUE， 否则返回 FALSE。
2. 该值为 1 时，安装 NT 驱动，安装成功后会提示要求重启系统。
3. 该值为 2 时，删除 NT 下驱动。
4. 该值为 10 时，功能同值为 1 时一样，但不弹出要求重启动系统的提示。
返回值 ：函数调用成功则返回 TRUE，或函数调用失败则返回 FALSE。
说明 ：该函数仅在 WINNT4/2K/XP 下发生作用。
相关函数 ：okSetAllocBuffer, okSetStaticVxD。

BOOL WINAPI okUnRegister(DWORD dwCmd);

功能 ：卸载驱动程序和安装信息。
返回值 ：函数调用成功则返回 TRUE，或函数调用失败则返回 ALSE。
相关函数 ：okSetNTDriver, okSetStaticVxD。

10.附：IO 卡相关函数

SHORT WINAPI okGetGPIOPort(SHORT index, WORD *wPortBase);

功能：该函数只针对 OK PCI GPIO20，用来得到 GPIO 卡的端口基地址和端口数量。
index：GPIO 卡的索引号（以 0 为起始）。
wPortBase：返回端口的基地址。
返回值 ：如果成功返回 GPIO 卡端口数量，否则返回 0。
相关函数 ：okOutputByte，okInputByte，

BOOL WINAPI okSetPortBase(WORD wPortBase, SHORT iPortCount);

功能：预先设置非 PCI 的 IO 卡需要使用的端口基地址和端口数量。
wPortBase：端口基地址。
iPortCount：端口数量。
说明：对于在 WinNT/Win2000 下使用的 PCI 总线的 IO 卡，为了使用一些端口（默认基地址是 0x300，数量是 4），您必须先用该函数设置端口基地址和数量，而且这些端口只有系统重启之后才能正常使用。
返回值 ：返回 GPIO 卡端口数量。
注意：在使用以下 I/O 函数之前，您必须已经使用过以下函数之一：okOpenBoard, okGetGPIOPort（推荐）, okGetImageDevice。

BOOL WINAPI okOutputByte(WORD wPort, BYTE data);

功能：在指定端口输出 1 字节的数据。
wPort：端口地址。
Data：数据。
返回值：成功返回 1，失败返回 0。

BOOL WINAPI okOutputShort(WORD wPort, SHORT data);

功能：在指定端口输出 2 字节的数据。
wPort：端口地址。
Data：数据。
返回值：成功返回 1，失败返回 0。

BOOL WINAPI okOutputLong(WORD wPort, MLONG data);

功能：在指定端口输出 4 字节的数据。
wPort：端口地址。
Data：数据。
返回值：成功返回 1，失败返回 0。

BYTE WINAPI okInputByte(WORD wPort);

功能：在指定端口输入 1 字节的数据。
wPort：端口地址。
返回值：返回读到的数据。

SHORT WINAPI okInputShort(WORD wPort);

功能：在指定端口输入 2 字节的数据。
wPort：端口地址。
返回值 ：返回读到的数据。

MLONG WINAPI okInputLong(WORD wPort);

功能：在指定端口输入 4 字节的数据。
wPort：端口地址。
返回值：返回读到的数据。

第5章 库函数索引