USB-2404

4 通道 24 位 156KHz SPS 同步采集

16 路 DI，16 路 DO 卡

用户手册

北京新超仁达科技有限公司

2012．10

从此文件发布日期起，在此发表的是当前或者拟定的信息。由于我们会不断对产品进行改进和增加特征，此出版物中的信息如有变动恕不另行通知。

一、前言

信息社会的发展，在很大程度上取决于信息与信号处理技术的先进性。数字信号处理技术的出现改变了信息与信号处理技术的整个面貌，而数据采集作为数字信号处理的必不可少的前期工作在整个数字系统中起到关键性、乃至决定性的作用，其应用已经深入到信号处理的各个领域中。实时信号处理、数字图像处理等领域对高速度、高精度数据采集卡的需求越来越大。ISA 总线由于其传输速度的限制而逐渐被淘汰。我公司推出的基于 PCI 总线、USB 总线等数据采集卡综合了国内外众多同类产品的优点，以及使用的便捷、稳定的性能、极高的性价比，获得多家客户的好评，是一系列真正具有可比性的产品，也是您理想与明智的选择。

衷心感谢您选用我公司的产品！

二、概述

USB-2404 板是一款 USB2.0 总线高速、高精度 4 通道同步采集卡，高达 24 位精度、156KSPs 采样率。本卡支持外触发和软件定时触发。4 路模拟信号、1 路外触发信号（下降沿触发）通过板上的 IDC-20 接头 P1 口输入。16 路 DI、16 路 DO 通过 IDC-40 接头 P2 口输入、输出。

USB-2404 模拟量输入信号采用差分输入方式进入 ADC 转换，AD 转换结果存储在大容量存储器 SDRAM（容量 4M 字）中。AD 芯片自带差分放大器和数字滤波器。

USB-2404卡的USB主控芯片采用CYPRESS公司的CY68013A。本板是一款USB2.0总线高速、高精度数据采集板，支持热插拔，即插即用。运用现场可编程门阵列FPGA设计，提高可靠性。

出厂时提供 Win98/2000/XP 下驱动程序和动态链接程序（DLL）及编程指导（DEMO 程序），有 VB/VC/LABVIEW 采集程序例程，并提供两年的质保服务。

三、产品应用

USB-2404 是一款基于 USB2.0 总线的数据采集卡，可直接和计算机的 USB 口相连，构成实验室、产品质量检测中心等各种领域的数据采集、波形分析和处理系统。也可构成工业生产过程监控系统。它的主要应用场合为：

- 电子产品质量检测
- 医学检测
- 高精度信号采集
- 无相差采集
- IO 控制

四、性能特点

- 采用独自四片 AD7765（156KHz）
- 高性能, 24-bit Σ-Δ ADC ，无丢失码
- 信噪比：112 db min @ 156 KHz;
- 用户只需单 5V 供电（在 USB 口供电不足的情况下）
- 双极性输入（±5V*，±10V）
- AD 触发方式：软件定时触发、外触发（下降沿触发）
- 高精度 4.096V 参考电源
- 硬件自带数字滤波器
- 4 路模拟信号差分输入，可软件选择任意几个通道同时工作
- 采样频率可软件设置（范围：1.95KHz～156.25KHz）
- 电源管理芯片，自动决定是外供电还是采用 USB 总线供电
- 外供电源防接反保护，如果外电源接反，二极管正向导通，板载 1A 保险丝将起保护作用
- 板载大容量存储器（4M 字），支持连续采集，实现实时监控

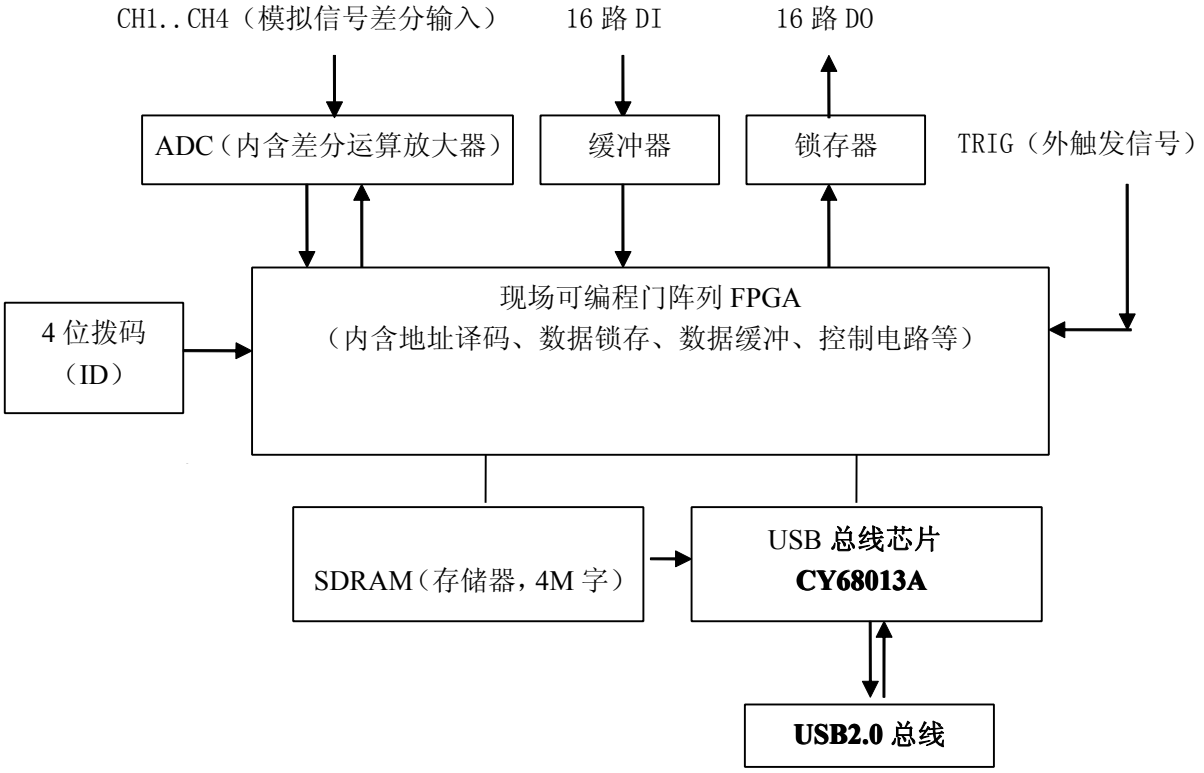
- 用户 4K 字节 EEPROM 空间，方便用户写入自己的参数
- 板卡 ID 识别（0～15），支持一台机器插入多块 USB-2404 卡
- 16 路数字量输入，16 路数字量输出。输出带锁存功能，上电自动清零
- USB2.0 总线，CYPRESS FX2 低功耗主控芯片
- GPIF 胶链接传输数据
- 提供 WIN7/VISTA/2000/XP 下驱动程序及动态连接库
- 尺寸大小（不含外壳）：91(W) × 139(L) (mm)
- *号为出厂默认设置

五、技术参数

- 工作电压： 5V±0.25V
- 工作温度： 0℃～70℃
- 存储温度： -10℃～85℃
- 湿 度： 5%～95%

六、工作原理

6.1、逻辑框图



逻辑方框图

6.2、工作原理简述

USB-2404 由以下功能模块组成：USB2.0 总线桥电路，地址译码及数据锁存，AD 转换，数据存储等功能组成。

地址译码及数据锁存：由现场可编程门阵列 FPGA 芯片控制。由于是 USB2.0 总线，用户可以不关心具体 I/O 地址，直接调用我公司提供的动态连接库即可。8 位数据模式，具体的 IO 地址由 USB 总控芯片解释。

AD 转换：外部模拟信号由 IDC-20 接头 P1 接入，经过高速差分运算放大器，进入 AD 转换，AD 控制时序由 FPGA 完成。

数据存储：AD 转换的结果存储在 SDRAM 中。读、写数据的时序由 FPGA 控制。

数字 IO：16 路数字量输入、16 路数字量输出。输出带锁存功能，且上电自动清零。

USB2.0 总控芯片：负责总线通讯，胶链接传输数据时在 FPGA 控制下完成。CY68013 当成从 FIFO 处理。

6.3、触发模式

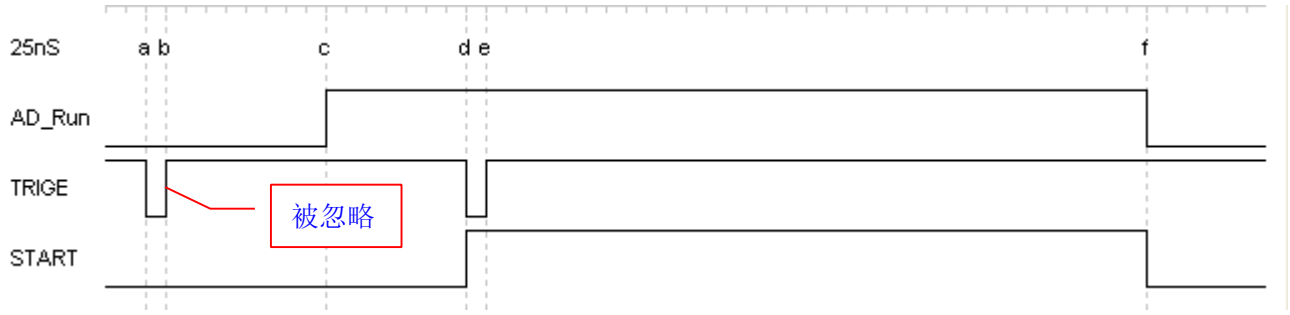
6.3.1、软启动：

在这种模式下，用户配置采样频率和同步通道后，启动 AD，便可进行连续不间断采样，每一通道都能达到设定的采样频率，各通道之间无相差。

6.3.2、外触发：

在这种模式下，需要接入外触发信号（下降沿有效），用户配置采样频率和同步通道后，启动 AD，AD 并不马上开始工作，而是等待外触发信号下降沿到来，下降沿一到，AD 马上工作。

外触发时序图如下：



（注意：如果外触发信号的下降沿早于 AD 启动信号 AD_Run，则此下降沿被忽略）

6.4、SDRAM 中数据存放顺序

本卡采用 SDRAM FIFO 化的技术，数据位宽为 16 位（D0～D15），四路模拟信号（CH1～CH4）通过 AD 转换以后，结果存放在 FIFO 中。以四通道同时工作为例，其他依此类推。软件可任意勾选几个通道同时工作。一个通道的转换结果占用 4 个字节，其中数据占 3 个字节，1 个字节的状态位。

CH1、CH2、CH3、CH4 的转换结果依次存放在 FIFO 中的第 1、2，3、4，5、6，7、8 个位置，第二轮 CH1、CH2、CH3、CH4 的转换结果依次存放在 FIFO 中的第 9、10，11、12，13、14，15、16 个位置。。。依此类推(分两次写，一次写 16bit)

4 路同步的情况，CH1、CH2、CH3、CH4 被同步采样；各通道之间无相差。数据交织存放。具体可参照 DEMO 软件。

6.5、FIFO 容量

本卡采用 1 片 SDRAM，容量为 4M 字，FPGA 将 SDRAM FIFO 化，USB 主控芯片内含 1K 字 FIFO，所以本板总的 FIFO 容量为：硬 FIFO+软 FIFO = 4M 字+1K 字（构成海量 FIFO），上位机一次读取 18432 个字节的数据量，该数据量能被 2、3 整除，分配到每个通道上的数据是一样多。

上位机是一次读取 18432 个字节，也可以更多或更少，但必须是 512 的倍数，因为 USB 端点 2 数据满 512 字节时，才打包提交。同时，为了上位机处理方便，一次读取的字节数最好能被（通道数×4）整除。

6.6、批量数据的传输

USB 的主控芯片的端点 2 我们配置为批量读方式，端点 0 配置为控制传输，当外部 FIFO（SDRAM）不为空且 USB 端点 2 的 FIFO 不满，USB 就自动从外 FIFO 中读取一个数至内 FIFO 中，当端点 2 数据满 512 时，就自动打包提交。

外部 FIFO（SDRAM）满标志为高时，代表数据有溢出，应当停止 AD，清空 FIFO 后，重新开始采样。否则 FIFO 中的数据存放顺序会被打乱。

由于 USB 主控芯片采用胶链接，与 16 位宽度的 SDRAM 无缝对接，但 USB 总线是 8 位位宽，所以内部有 16 位转成 2 个 8 位的操作，低 8 位在前，高 8 位在后。

七、信号定义

7.1、模拟输入输出引脚定义

P1：IDC-20芯，模拟输入插头，在输入的插头上标有对应的号码。信号定义如下：

插座引脚号	信号定义	插座引脚号	信号定义
1	$AINA_0+$	2	$AINA_0-$
3	$AINB_0+$	4	$AINB_0-$
5	$AINA_1+$	6	$AINA_1-$
7	$AINB_1+$	8	$AINB_1-$
9	$AINA_2+$	10	$AINA_2-$
11	$AINB_2+$	12	$AINB_2-$
13	$AINA_3+$	14	$AINA_3-$
15	$AINB_3+$	16	$AINB_3-$
17	AGND	18	AGND
19	TRIG	20	GND

AINA*n*+、AINA*n*−：±5V模拟信号差分输入对，n = 0，1，2，3，分别对应4路A/D输入通道；
AINB*n*+、AINB*n*−：±10V模拟信号差分输入对，n = 0，1，2，3，分别对应4路A/D输入通道；

举例说明：如果是通道3，接入的模拟信号范围是±10V，那么模拟信号就应该从$AINB_3+$、$AINB_3-$脚输入；

如果通道3接入的模拟信号范围是±5V，那么模拟信号既可以从$AINB_3+$、$AINB_3-$脚输入；也可以从$AINA_3+$、$AINA_3-$脚输入，但两者只能二选一，为了追求精度，这种情况，推荐从$AINA_3+$、$AINA_3-$脚输入。

17、18脚：AGND，模拟地；19脚：TRIG，为外触发。20脚：GND，为数字地。其中：模拟地和数字地是相通的。

7.2、数字量输入、输出引脚定义

P2：IDC-40（数字量输入、输出）插头，在输入的插头上标有对应的号码。信号定义如下：

插座引脚号	信 号 定 义	插座引脚号	信 号 定 义
1	DI_0	2	DI_1
3	DI_2	4	DI_3
5	DI_4	6	DI_5
7	DI_6	8	DI_7
9	DI_8	10	DI_9
11	DI_{10}	12	DI_{11}
13	DI_{12}	14	DI_{13}
15	DI_{14}	16	DI_{15}
17	AGND	18	AGND
19	AGND	20	AGND
21	DO_0	22	DO_1
23	DO_2	24	DO_3
25	DO_4	26	DO_5
27	DO_6	28	DO_7
29	DO_8	30	DO_9
31	DO_{10}	32	DO_{11}
33	DO_{12}	34	DO_{13}
35	DO_{14}	36	DO_{15}
37	AGND	38	AGND
39	AGND	40	AGND

DI_0～DI_{15}：开关量输入通道0～15；

DO_0～DO_{15}：开关量输出通道0～15；

17～20，37～40：AGND。

7.3、ID设置：（四位拨码开关SW设置）

如果系统插入了两块一样的卡，可通过设置不同的ID来进行区分。具体设置如下表：

ID3	ID2	ID1	ID0	Board ID
ON（1）	ON（1）	ON（1）	ON（1）	0
ON（1）	ON（1）	ON（1）	OFF（0）	1
ON（1）	ON（1）	OFF（0）	ON（1）	2
ON（1）	ON（1）	OFF（0）	OFF（0）	3
ON（1）	OFF（0）	ON（1）	ON（1）	4
ON（1）	OFF（0）	ON（1）	OFF（0）	5
ON（1）	OFF（0）	OFF（0）	ON（1）	6
ON（1）	OFF（0）	OFF（0）	OFF（0）	7
OFF（0）	ON（1）	ON（1）	ON（1）	8
OFF（0）	ON（1）	ON（1）	OFF（0）	9
OFF（0）	ON（1）	OFF（0）	ON（1）	10
OFF（0）	ON（1）	OFF（0）	OFF（0）	11
OFF（0）	OFF（0）	ON（1）	ON（1）	12
OFF（0）	OFF（0）	ON（1）	OFF（0）	13
OFF（0）	OFF（0）	OFF（0）	ON（1）	14
OFF（0）	OFF（0）	OFF（0）	OFF（0）	15

注意：OFF：0，ON：1

7.4、双极性模拟量输入的电压换算

双极性方式工作时，转换后的 24 位数码为补码。此时 24 位数据的最高位(DB_{23})为符号位，“1”表示负，“0”表示正。

如量程为－5～＋5V 时，此时数据与模拟电压值的对应关系为：

如果 DB_{23}=0：

模拟电压值＝数据(24 位) × 4.096 × 1.5781 ÷ 8388607；

如果 DB_{23}=1：

模拟电压值＝（数据(24 位)- 16777215）× 4.096 × 1.5781 ÷ 8388607；

如量程为－10～＋10V 时，此时数据与模拟电压值的对应关系为：

如果 DB_{23}=0：

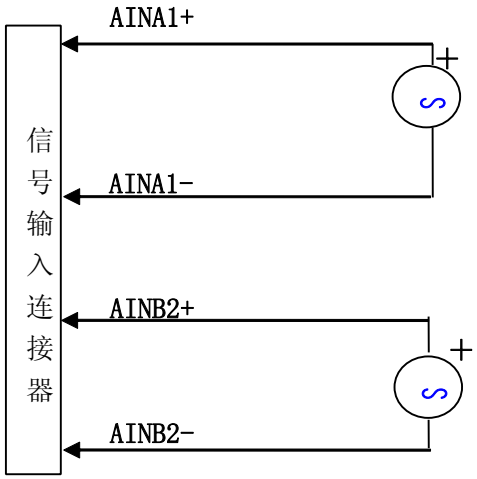
模拟电压值＝数据(24 位) × 4.096 × 3.167 ÷ 8388607；

如果 DB_{23}=1：

模拟电压值＝（数据(24 位)- 16777215）× 4.096 × 3.167 ÷ 8388607；

八、常用信号连接

8.1、外部模拟输入差分信号



外部触发信号从 TRIG 和 GND 输入。

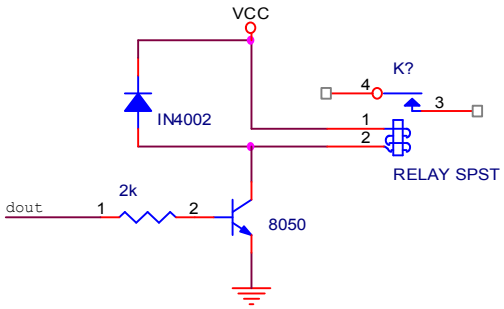
8.2、数字量输入

芯片本身带弱上拉，外部数字量输入信号浮空（即不接）的情况下，采集回来的信号为 1（高电平），当外部数字量输入信号为低电平时，采集回来的信号为 0（低电平）；

开关量输入电平不能低于-0.3V 或高于+5V。

8.3、数字量输出

输出不要对地线、电源短路。由于数字量输出是 TTL 输出，驱动能力有限，不能直接驱动继电器或电磁阀，可按以下电路来驱动继电器：



或直接购买我公司的继电器板 P800 或 PCLD-885。

九、软件

USB-2404 的软件包括 USB-2404 驱动程序，动态链接库及调用例程。

9.1、驱动安装

注意：如果 USB 口的供电能力较弱，则必须外供 5V 电，否则将提示发现“unknow device”。

首先区分自己的操作系统是 32 位还是 64 位。

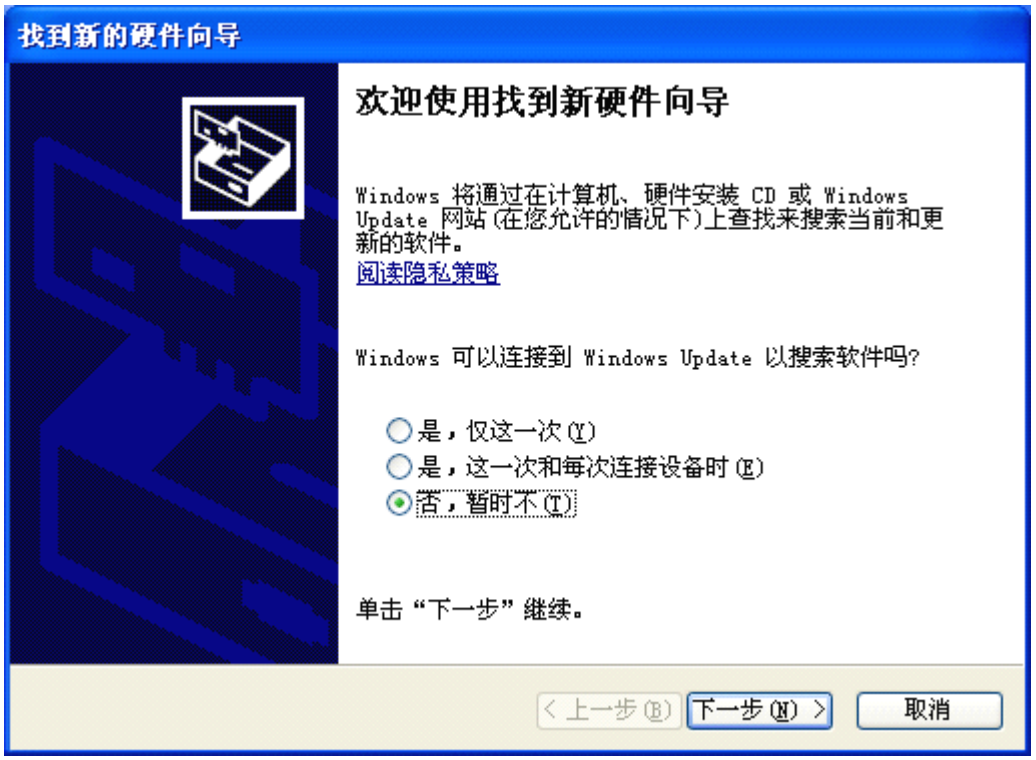
如何判断自己操作系统究竟是 32 位还是 64 位呢？方法其实很简单。具体方法如下：

点击“开始”按钮，右键点击“计算机->属性”，即可打开查看有关计算机基本信息的窗口，在这个窗口中会显示操作系统的版本和类型。如果是 64 位系统，则会在图中的红框处显示“64 位操作系统”。

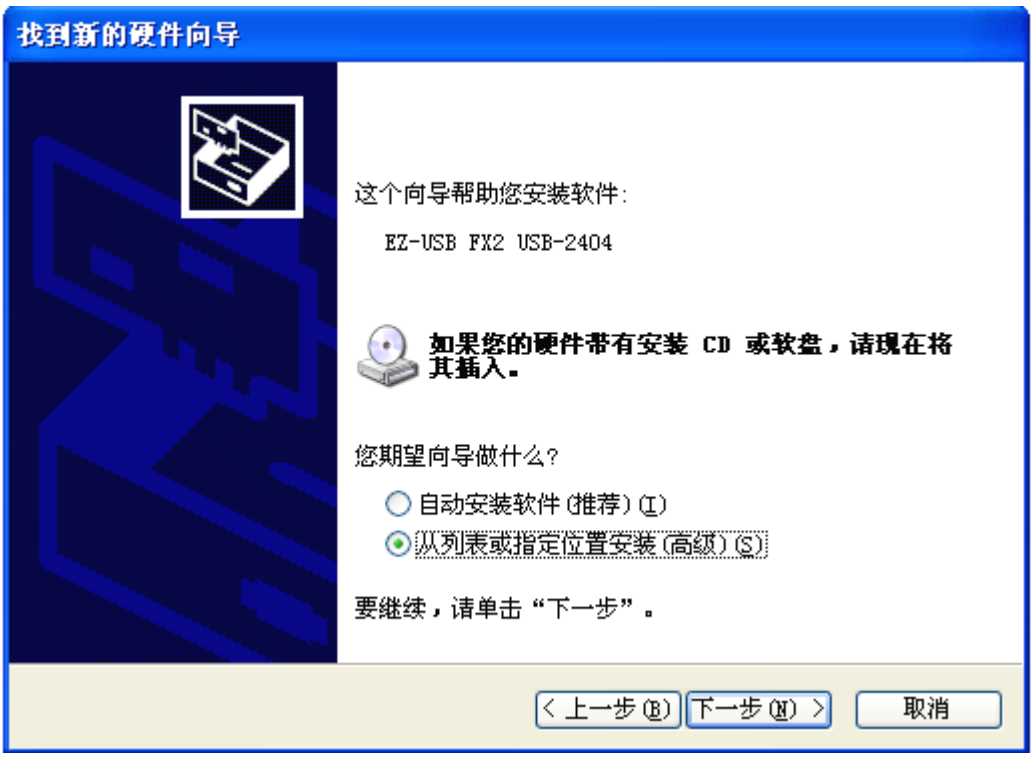
根据自己的操作系统，选择相应驱动文件夹下的 install.bat，双击运行此批处理，完成文件的拷贝。

重新启动计算机。如果是 WIN7/VISTA 系统，启动系统的时候，按 F8，选“禁用驱动程序签名强制”项进入系统。

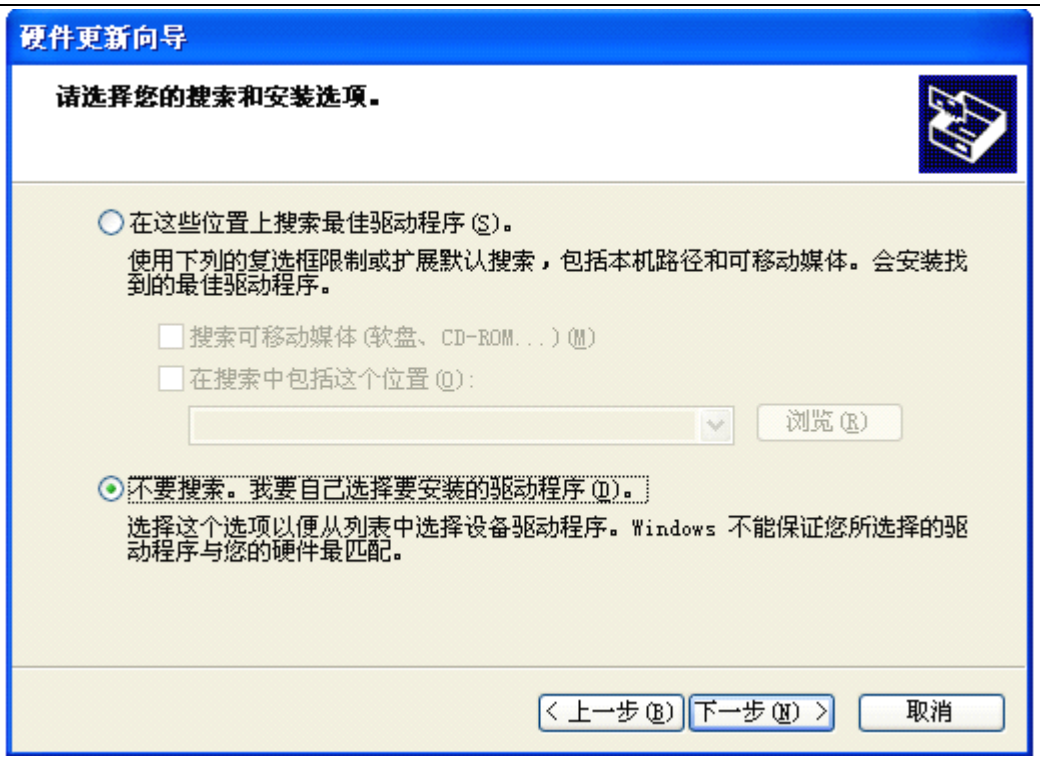
将 USB-2404 模块利用出厂附带的 USB 线缆插入某一 USB 口。操作系统将自行检测新硬件，并弹出“添加新硬件向导”对话框，如下图所示：


找到新的硬件向导
欢迎使用找到新硬件向导
Windows 将通过在计算机、硬件安装 CD 或 Windows Update 网站(在您允许的情况下)上查找来搜索当前和更新的软件。
阅读隐私策略
Windows 可以连接到 Windows Update 以搜索软件吗?
是，仅这一次(Y)
是，这一次和每次连接设备时(E)
否，暂时不(T)
单击“下一步”继续。
< 上一步(B)
下一步(N) >
取消

点击“下一步”

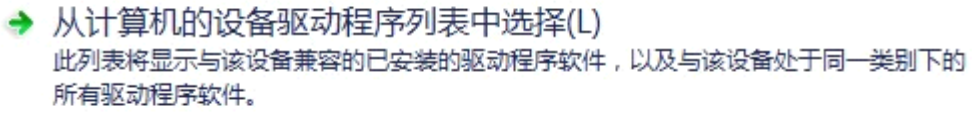

找到新的硬件向导
这个向导帮助您安装软件:
EZ-USB FX2 USB-2404
如果您的硬件带有安装 CD 或软盘，请现在将其插入。
您期望向导做什么?
自动安装软件(推荐)(I)
从列表或指定位置安装(高级)(S)
要继续，请单击“下一步”。
< 上一步(B)
下一步(N) >
取消

点击“下一步”



点击“下一步”

该步：如果是WIN7系统，则点击“从计算机的设备驱动程序列表中选择”，如下图所示：



3、选择“通用串行总线”，点击“从磁盘安装...”按钮，选择“USB2404.inf”所在的文件夹（用户根据自己的操作系统来选择相应的路径，x86（32-bit OS）、 x64（64-bit OS）），点击“确定”按钮，然后点击“下一步”按钮。根据向导提示完成安装。出现如下界面时，

点击“完成”按钮，完成驱动程序的安装。
说明：中途如提示未经微软数字签名，是否继续安装，点“仍然继续”，安装完成后如从（控制面板/系统/设备管理器）中可找到外部设备：USB-2404 驱动程序 新超科技，则可证明硬件驱动安装正确。
4、如果同一机器里插有我方多块 USB2404，则按照上述方法依次安装驱动。

说明：驱动安装完成后，无需重启计算机，除非 windows 要求你这么做，你只需重新插拔一下 USB 设备即可。

9.2、测试程序

提供测试程序为 VB，VC 编写，可对 USB-2404 卡的所有功能进行测试。用户可参照例程自行编程。

9.3、函数调用说明

提供动态链接库作为调用接口，它所封装的函数可以在应用程序运行时调用。任意一种可以调用 DLL 链接库的编程工具均可进行编程。下列函数为 DLL 函数原型，请注意数据格式的匹配及函数的返回类型。

9.3.1、库中部分函数说明:

打开设备：

函数：HANDLE WINAPI USB2404_OpenDevice(long card_id)
功能：打开USB2404设备,返回硬件操作句柄,出错则为无效
参数：card_id: 指定设备序号（由板上拨码开关SW设置，0-15，用于区分板卡的物理ID）
返回值：NULL 表示打开设备失败；其它值表示打开设备成功
详细参阅板卡ID设置章节。

关闭设备

函数：BOOL WINAPI USB2404_CloseDevice(HANDLE hDevice)
功能：关闭USB设备
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE

软启动AD

函数：BOOL WINAPI USB2404_ADStart(HANDLE hDevice)
功能：软启动AD（如果为外触发模式，则还需要外触发信号的配合，详见《触发模式》章节）
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE

设置采样频率

函数：BOOL WINAPI USB2404_SetFreDiv(HANDLE hDevice,BYTE FreDiv)
功能：设置AD采样频率
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
FreDiv: 取值范围（1-80）

FreDiv：参数取值决定采样频率，系统采用40M晶振时钟，如果FreDiv=1，采样频率为：40M ÷ 256 ≈ 156.25KHz；如果FreDiv = 80，采样频率为：40M ÷ 256 ÷ 80 ≈ 1.95KHz；
固本采集卡支持的采样频率范围为：1.95KHz～156.25KHz。
因为分频因子只能为整数，所以想要的采样频率和实际的采样频率可能有区别，请以实际的采样频率为准。

返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE

注意：如果不调用USB2404_SetFreDiv函数，则FPGA默认是156.25KHz的采样频率。

设置同步通道

函数：BOOL WINAPI USB2404_SetADCh(HANDLE hDevice,BYTE ChSelect)
功能：设置AD同步通道数
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
ChSelect：取值范围（1-15）

ChSelect取值与同步通道数的对应关系如下：

ChSelect取值	同步通道
0	没有一个通道工作
1	通道0
2	通道1
3	通道0、1
4	通道2
5	通道0、2
6	通道1、2
7	通道0、1、2
8	通道3
9	通道0、3
10	通道1、3
11	通道0、1、3
12	通道2、3
13	通道0、2、3
14	通道1、2、3
15	通道0、1、2、3

相应位为1代表该通道有效。
返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE
注：FPGA默认为一个通道也不选择。

设置AD触发模式

函数：BOOL WINAPI USB2404_SetADMode(HANDLE hDevice,BYTE ADMode)
功能：设置AD触发模式
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
ADMode: 取值范围（0-1）

ADMode取值与触发方式的对应关系如下：

ADMode取值	触发方式
0	外触发
1	软启动

返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE
注：FPGA默认为软启动。

将AD二进制原码转换成电压

函数：void WINAPI Fun_DataToV(double* data,BYTE ADRange)
功能：将AD二进制原码转换成电压
参数：data: AD采样数值（0-16777215）
ADRange: 取值范围（0-1），0：±5V；1：±10V；取决于AD的量程设置。
返回值：对应的电压值

停止AD

函数：BOOL WINAPI USB2404_ADStop(HANDLE hDevice)
功能：停止AD
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE

清RAM

函数：BOOL WINAPI USB2404_CLRRAM(HANDLE hDevice)
功能：清空SDRAM
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE

读取DRAM的状态

函数：BYTE WINAPI USB2404_ReadDramState(HANDLE hDevice,BOOL* b_Success)
功能：读取DRAM的状态
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
b_Success，如果*b_Success=TRUE，则函数调用成功，否则函数调用失败
返回值：如果函数调用成功，则返回DRAM的状态，调用失败，返回255
返回值含义如下：

Bit7～Bit3	Bit2	Bit1	Bit0
0	DRAM_FF	DRAM_HF	DRAM_EF

DRAM_FF：该位为1代表DRAM全满；
DRAM_HF：该位为1代表DRAM半满；
DRAM_EF：该位为1代表DRAM空；
注：全满一定半满且不为空，为空一定不半满和全满。

板卡ID

函数：BYTE WINAPI USB2404_GetID(HANDLE hDevice,BOOL* b_Success)
功能：获取板卡的物理ID
参数：hDevice，设备操作句柄，由USB2404_OpenDevice函数返回
b_Success，如果*b_Success=TRUE，则函数调用成功，否则函数调用失败
返回值：如果函数调用成功，则返回物理ID(0-15)，调用失败，返回255

DI

函数：BYTE WINAPI USB2404_DI(HANDLE hDevice,BYTE dich,BOOL* b_Success)
功能：获取八位数字量输入
参数：hDevice，设备操作句柄，由USB2404_OpenDevice函数返回
dich，USB2404共有16位数字量输入，分成两个八组，(0-1)，0对应前8位，1对应后8位
b_Success，如果*b_Success=TRUE，则函数调用成功，否则函数调用失败
返回值：如果函数调用成功，则返回八位数字量输入值(0-255)，调用失败，返回255

函数：WORD WINAPI USB2404_DIALL(HANDLE hDevice,BOOL* b_Success)
功能：获取十六位数字量输入
参数：hDevice，设备操作句柄，由USB2404_OpenDevice函数返回
b_Success，如果*b_Success=TRUE，则函数调用成功，否则函数调用失败
返回值：如果函数调用成功，则返回十六位数字量输入值(0-65535)，调用失败，返回65535

函数：BYTE WINAPI USB2404_DIBIT(HANDLE hDevice,BYTE diBIT,BOOL* b_Success)
功能：获取某位数字量输入
参数：hDevice，设备操作句柄，由USB2404_OpenDevice函数返回
diBIT，USB2404共有16位数字量输入，(0-15)，0对应第1位，15对应16位
b_Success，如果*b_Success=TRUE，则函数调用成功，否则函数调用失败
返回值：如果函数调用成功，则返回某位数字量输入值(1或0)，调用失败，返回1

DO

函数：BOOL WINAPI USB2404_DO(HANDLE hDevice,BYTE dodata,BYTE doch)
功能：控制八位数字量输出
参数：hDevice，设备操作句柄，由USB2404_OpenDevice函数返回
dodata，要输出的八位数字量值(0-255)
doch，USB2404共有16位数字量输出，分成两个八组，(0-1)，0对应前8位，1对应后8位
返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE

函数：BOOL WINAPI USB2404_DOALL(HANDLE hDevice,WORD dodata)
功能：控制十六位数字量输出
参数：hDevice，设备操作句柄，由USB2404_OpenDevice函数返回
dodata，要输出的十六位数字量值(0-65535)
返回值：如果函数调用成功，则返回TRUE，调用失败，返回FALSE

函数：WORD WINAPI USB2404_DOBIT(HANDLE hDevice,WORD dodata,BYTE BITValue,BYTE whichBIT,BOOL* b_Success)
功能：数字量按位输出（可以指定某位为 1 或 0）
参数：hDevice，设备操作句柄，由USB2404_OpenDevice函数返回
dodata，函数调用之前十六位数字量输出的状态(0-65535)
BITValue，决定 whichBIT 指定的位输出高还是低，(0 或 1)，0 输出低电平，1 输出高电平
whichBIT，指定哪位，(0-15)
b_Success，如果*b_Success=TRUE，则函数调用成功，否则函数调用失败
返回值：如果函数调用成功，则返回新的十六位数字量输出的状态，调用失败，返回原值

微秒延时函数

函数：void WINAPI DelayUs(int dly)
功能：微秒延时
参数：dly，微秒数
返回值：无

获取动态链接库的版本：

函数：double WINAPI xc_GetVersion()

功能：读当前计数器的值

参数：无

返回值：动态链接库的版本号

往 USB 总控芯片发送请求命令

函数：BOOL WINAPI USB2404_IOCTL_VENDOR_REQUEST(HANDLE hDevice,BYTE request)

功能：往 USB 总控芯片发送请求命令

参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回

request: 请求命令(0xB2、0xB3、0xBA)

B2命令含义：USB总控芯片允许传输并复位USB端点FIFO

B3命令含义：USB总控芯片初始化

BA命令含义：USB总控芯片禁止传输

返回值：如果函数调用成功，则返回 TRUE，调用失败，返回 FALSE

从 USB 块读数据

函数：BOOL WINAPI USB2404_IOCTL_BULK_READ(HANDLE hDevice,unsigned char* Buffer,long BufferSize,unsigned long* nBytes)

功能：从 USB 块读数据

参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回

Buffer: 用来接收读取的数据

BufferSize: 需要读取的字节数

nBytes: 指针，指向实际读取的字节数

返回值：如果函数调用成功，则返回 TRUE，调用失败，返回 FALSE

注意： 一次 AD 结果 24 位分成 3 个字节，外加一个状态字节，共四个字节，其中状态字节在四个字节的最前，再次为 AD 结果的低字节，再次为 AD 结果的中字节，最后为 AD 结果的高字节。以四通道同步为例，数据存放顺序为：Buffer[0]，Buffer[1]，Buffer[2]，Buffer[3]，分别为第一路 AD 的状态、结果低、结果中、结果高字节；Buffer[4]，Buffer[5]，Buffer[6]，Buffer[7]，分别为第二路 AD 的状态、结果低、结果中、结果高字节；Buffer[8]，Buffer[9]，Buffer[10]，Buffer[11]，分别为第三路 AD 的状态、结果低、结果中、结果高字节，分别为第四路 AD 的低、高字节；Buffer[12]，Buffer[13]，Buffer[14]，Buffer[15]，分别为第四路 AD 的状态、结果低、结果中、结果高字节；。。。依此类推。

其中，状态寄存器结构如下：

Bit7	Bit6	Bit5	Bit4	Bit3～Bit0
1：数据有效 0：数据无效	1：超量程 0：不超量程	0	0	Don't care

读写 EEPROM 的操作

函数：BOOL WINAPI USB2404_RWEEp(HANDLE hDevice,WORD addr,long len, unsigned char* buffer,BOOL bIsRead = TRUE)

功能：读写板上 EEPROM

参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回

addr: 从0X1000开始，范围0X1000～0X1FFF

len: 需要读取的字节数，最大0X1000（即4096个字节）

buffer: 读、写数据缓冲区指针

bIsRead: 读还是写，默认是读操作

返回值：如果函数调用成功，则返回 TRUE，调用失败，返回 FALSE

从 EEPROM 读出一个字节

函数：BOOL WINAPI USB2404_REEp_B(HANDLE hDevice,WORD addr,unsigned char* buffer)
功能：从 EEPROM 读出一个字节
参数：hDevice: 卡的操作句柄，由USB2404_OpenDevice函数返回
addr: 从0X1000开始，范围0X1000～0X1FFF
buffer: 读、写数据缓冲区指针
返回值：如果函数调用成功，则返回 TRUE，调用失败，返回 FALSE

写一个字节至 EEPROM

函数：BOOL WINAPI USB2404_WEEp_B(HANDLE hDevice,WORD addr,unsigned char* buffer)
功能：写一个字节至 EEPROM
参数：hDevice: 卡的操作句柄，由 USB2404_OpenDevice 函数返回
addr: 从0X1000开始，范围0X1000～0X1FFF
buffer: 读、写数据缓冲区指针
返回值：如果函数调用成功，则返回 TRUE，调用失败，返回 FALSE
（用户可利用以上三个 EEPROM 操作函数，写入自己的参数到板载的 EEPROM 上）

9.3.2、函数调用注意事项
调用函数的正确顺序为：
1、加载驱动，打开设备。（应用程序在初始化时调用一次 USB2404_OpenDevice 即可）
2、对板卡进行访问，可进行 AD 转换操作。
3、关闭设备。(推荐放在退出整个应用程序的时候调用 USB2404_CloseDevice)

9.4、DLL函数全部是WINAPI调用约定的，即__stdcall接口

在使用各种编程语言时应注意选择，
Visual C++/C++ Builder/Delphi
可以使用两种类型的调用约定。要在函数定义中明确指出__stdcall 还是__cdecl；
Visual Basic/PowerBuilder等语言
应该使用__stdcall调用接口。

9.5、驱动文件

文件名	文件类型及功能	适用的操作系统
cyusb.Sys	Win32/64标准设备驱动WDM 模式的设备驱动程序库	Win7，Vista Win2000，WinXP
USB2404.Dll	底层驱动程序库的用户级函数接口封装所用的动态库。	所有操作系统
USB2404.Lib	基于Microsoft Visual C++工程开发环境的驱动程序函数接口输入库。	所有操作系统
USB_2404.h	基于Microsoft Visual C++工程开发环境的函数调用头文件。	所有操作系统
PcMod.Bas	基于Microsoft Visual Basic 工程开发环境的驱动程序函数接口输入模块文件	所有操作系统

十、编程指导

本公司提供的动态链接库经精心设计，支持所有的高级语言，高级语言如何调库请参阅相关书籍.本卡是中高速数据采集卡，在 windows 下编程，为了避免丢数，建议采用多线程。

10.1、VC 程序编程说明

编程前，将 USB2404.lib 及 USB_2404.h 程序拷贝到用户当前目录中。（需要的文件在 VC 目录中）

VC 编程的基本流程：

利用显式调用加载函数。USB2404.lib、USB_2404.h 文件必须在当前工作目录中。方法，程序的开始处加入如下语句：

```
#pragma comment(lib, “USB2404.lib”)
#include “USB_2404.h”
```

详细可以参考 VC 目录中的程序，USB_2404.h 文件包含了需要的函数的声明过程。

利用 USB2404_OpenDevice 函数获得板卡的操作句柄。

在退出程序时必须执行如下操作：利用 USB2404_CloseDevice 函数关闭句柄。

例：

```
//获得所有 USB2404 的操作函数
#pragma comment(lib, “USB2404.lib”)
#include “USB_2404.h”

HANDLE hDevice = INVALID_HANDLE_VALUE;  //硬件操作句柄

Main()
{

    //获得 USB2404 硬件操作句柄
    hDevice = USB2404_OpenDevice(0); //创建设备驱动句柄，设备号为 0，取决于 4 位拨码开关

        …… //用户程序

    //退出
    USB2404_CloseDevice(hDevice); //关闭操作句柄
}
```

详细可以参考光盘上的 USB2404 的 VC 目录下的例子。

在编程时必须注意，硬件操作句柄 HANDLE 必须为全局变量或必须传递给有相应硬件操作的函数。硬件句柄只要在程序启动时打开一次即可，不需要每次打开或关闭。

10.2、VB 程序编程说明

编程前，请将 USB2404.dll 动态链接库程序拷贝到用户当前目录中或 windows 系统的 system32 目录中。

VB 编程的基本流程：

在工程菜单中选择添加模块，将 PcMod.bas 模块添加进来（该模块在光盘中\USB\USB2404\vb 目录中，应用时将文件拷贝到当前工作目录），此文件为所有函数的声明文件。

在模块中定义一个硬件操作句柄，为一个 long 属性的全局变量，这样可以被用户程序中的所有 form 调用（例：PcMod.bas 中声明的句柄 hDevice）。

利用 USB2404_OpenDevice 函数获得板卡的操作句柄。

在退出程序时必须执行如下操作：

利用 USB2404_CloseDevice 函数关闭句柄

注：PcMod.bas 模块已经包含了所有必要的 USB2404 函数的声明语句。

例：

```
DIM hDevice as long
Private Sub Form_Load()
    hDevice = USB2404_OpenDevice(0)  ‘打开设备 0 号，获得驱动句柄
    ……  ‘其他操作
End Sub
……
Private Sub Form_Unload(Cancel As Integer)
    USB2404_CloseDevice hDevice  ‘关闭操作句柄
End Sub
```

注：VB 中如果设备操作句柄不等于&H0 为有效句柄。

10.3、LabVIEW 程序编程说明

本公司生产的所有采集卡的相关接口函数，均以动态链接库的形式提供给用户。在使用 LabVIEW 对本公司采集卡进行开发时，只需通过 LabVIEW 中的 Call Library Function Node 节点来调用我们所提供的动态链接库函数即可对硬件进行相关操作。

10.4、Delphi 程序编程说明

在 Delphi 中调用动态链接库的方式分为静态调用和动态调用，本公司所提供的例程均采用静态调用方式。

编程前，请将 USB2404.dll 动态链接库程序拷贝到用户当前目录中或 windows 系统的 system32 目录中

Delphi 编程的基本流程：

1. 在.pas 文件中的 implementation 处声明动态连接库中的函数。
2. 定义一个硬件操作句柄，为一个 ulong 属性的全局变量。
3. 利用 USB2404_OpenDevice 函数获得板卡的操作句柄。

在退出程序时必须执行如下操作：

利用 USB2404_CloseDevice 函数关闭句柄

例：

```
var
hDevice:ulong;//句柄
……
procedure TForm1.FormCreate(Sender: TObject);
var
i:ulong;
begin
   hDevice:= USB2404_OpenDevice(0);
end;
……//其他操作
procedure TForm1.Formdestroy(Sender: TObject);
begin
```

```
    USB2404_CloseDevice(hDevice);  //关闭操作句柄
  end;
```

注：Delphi 中如果设备操作句柄不等于$0 为有效句柄。

有关用户其他方面的应用请参考光盘中的例程。

提供的 DEMO 程序由 VC6.0 编写，采用线程的办法取数。详细请参阅 DEMO 程序。

十一、维修服务

11.1、产品完整性

USB-2404 产品应包括以下内容，请检查其完整性：

1、 USB-2404一块（贴有出厂日期）。
2、 USB连接电缆一条。
3、 20、40芯扁平线缆各一条。
4、 软件光盘一张（含驱动软件及说明书）。
5、 配套端子板P-500，需单独购买。

11.2、维修

本产品自售出之日起两年内，凡用户正确使用下，出现产品质量问题的，免费维修。（出厂日期的贴条撕毁无效）因违反操作规定和要求而造成损坏的，收取元器件成本费和维修费。

11.3、服务

当您购买 USB2404 之后，软、硬件及其它技术上使用问题均可通过电话或 E-mail 与我们联系,我们将提供令您满意的服务。

厂家保留对手册的更新和修改的权利，如有改动，恕不另行通知，请到公司的官方网站(www.xckz.com)下载，感谢您的支持和理解。