

KPCI-1816 16 位多功能数据采集卡

使用说明书

北京科瑞兴业科技有限公司

北京科瑞兴业科技有限公司
邮政编码: 100086

地址: 北京市海淀区知春里 28 号开源商务写字楼 212/213 室
电话: 010-51650651 010-62527214 传真: 010-62657424

<http://www.krxgk.com>

Sales E-mail: sgq@krxgk.com

Tech Support E-mail: lilanzhen007@126.com

目录

第一章 概述

- 1、 介绍
- 2、 应用
- 3、 性能和技术指标
- 4、 软件支持

第二章 主要元件位置图、信号输出插座和开关跳线选择定义

- 1、 主要元件布局图
- 2、 短路套设置
- 3、 信号输入输出插座定义
- 4、 模拟信号输入连接方式及应注意的问题

第三章 函数模块调用说明

1. A/D 采集过程流程图
2. 函数说明

第四章 KPCI-1816 的校准、保修和注意事项

1. 注意事项
2. 校准
3. 保修

第一章 概述

一、介绍

KPCI-1816 是一款 PCI 总线的 16 位多功能采集卡，其先进的电路设计使它具有更高的质量和功能。卡上具有 16 位 16 路模拟量采集通道，16 位 2 路模拟量输出通道和 16 路开关量输入通道以及 16 路开关量输出通道，同时还有 3 路计数通道，特别适合需要采集和控制多种信号的用户使用。

KPCI-1816 卡上的 16 位精度的模拟量采集通道和模拟量输出通道，满足了用户在具有高精度要求的场合，完成对模拟量信号的采集和控制。

KPCI-1816 卡上带有程控增益放大电路，通过软件可以设定不同的输入范围， $\pm 10V$, $\pm 5V$, $\pm 2.5V$, $\pm 1.25V$, $\pm 0.625V$, $0\sim 10V$, $0\sim 5V$, $0\sim 2.5V$, $0\sim 1.25V$, $0\sim 0.625V$ 的多种选择，方便了用户对不同信号的测量要求。

KPCI-1816 卡的最高采集频率可达 200K/S，卡上带有 4K 容量的 FIFO 存储，可以完成大量信号的采集和存储，定时触发、软件触发和外触发三种方式，以适应不同场合的数据采集，方便了用户的使用。

KPCI-1816 卡上具有 2 路 16 位精度的模拟量输出通道，可以通过跳线选择电压和电流输出方式，来控制不同的信号。

KPCI-1816 卡上的 16 路开关量输入通道的输入电平与 TTL 兼容，16 路开关量输出通道的输出电平与 TTL 兼容，3 路 16 位计数通道采用的是 8254 计数器，与 TTL 电平兼容。

KPCI-1816 卡上带有一个 4 位 DIP 拨码开关，当计算机上安装多块板卡时，可使用此开关来定义每块板卡的 ID。用户可以很方便的在硬件配置和软件编程时区分和访问每块板卡。

二、性能及技术指标

2.1 PCI 局部总线性能

PCI 总线宽度 32 位，同步工作频率可达到 33MHz，最高传输速率为 132MB/S
能够完成自动配置，实现设备的即插即用

2.2 模拟信号输入部分

模拟通道数: 16 路单端输入或 8 路差分输入

模拟输入电压范围: $\pm 10V$, $\pm 5V$, $\pm 2.5V$, $\pm 1.25V$, $\pm 0.625V$, $0\sim 10V$, $0\sim 5V$, $0\sim 2.5V$, $0\sim 1.25V$, $0\sim 0.625V$ 。

模拟输入阻抗: 10M Ω

2.3 A/D 转换电路部分

A/D 分辨率: 16 位(65535)

非线性误差: $\pm 3LSB$ (最大)

转换时间: 10 μs

系统测量精度: 0.1%

2.4 A/D 采集方式及周期

三种采样触发方式: 定时触发、软件触发和外触发，可由软件设置使用 FIFO 还是不使用 FIFO。

2.5 FIFO 存储器

深度: 4K Words

宽度: 16 位

标志： 空、有数、半满

2.6 模拟量输出部分：

- D/A分辨率：16位(65535)
- 模拟输出电压、电流范围：±5V、±10V、0~+5V 0~+10V、和4~20mA范围的电流输出
- 输出通道数：2路
- 非线性误差：±2LSB(最大)
- 建立时间：10μS(0.01%精度)
- 输出阻抗：0.2Ω
- 具有加电清零功能

2.7 数字量 DIO：

16 路数字量 DI 输入和 16 路数字量 DO 输出，兼容 TTL 电平。

2.8 定时器：

设计有 3 路定时器输出电路，可实现 1us 到 2s 的定时控制。支持两种工作方式

1、单一定时方式

当没有启动定时和定时结束后输出保持高电平，在启动定时期间，输出为低电平。

5、 方波输出方波

可输出占空比为 1: 1 的任意周期方波，周期范围在 1us 到 1s。兼容 TTL 电平输出。

2.9 计数器：

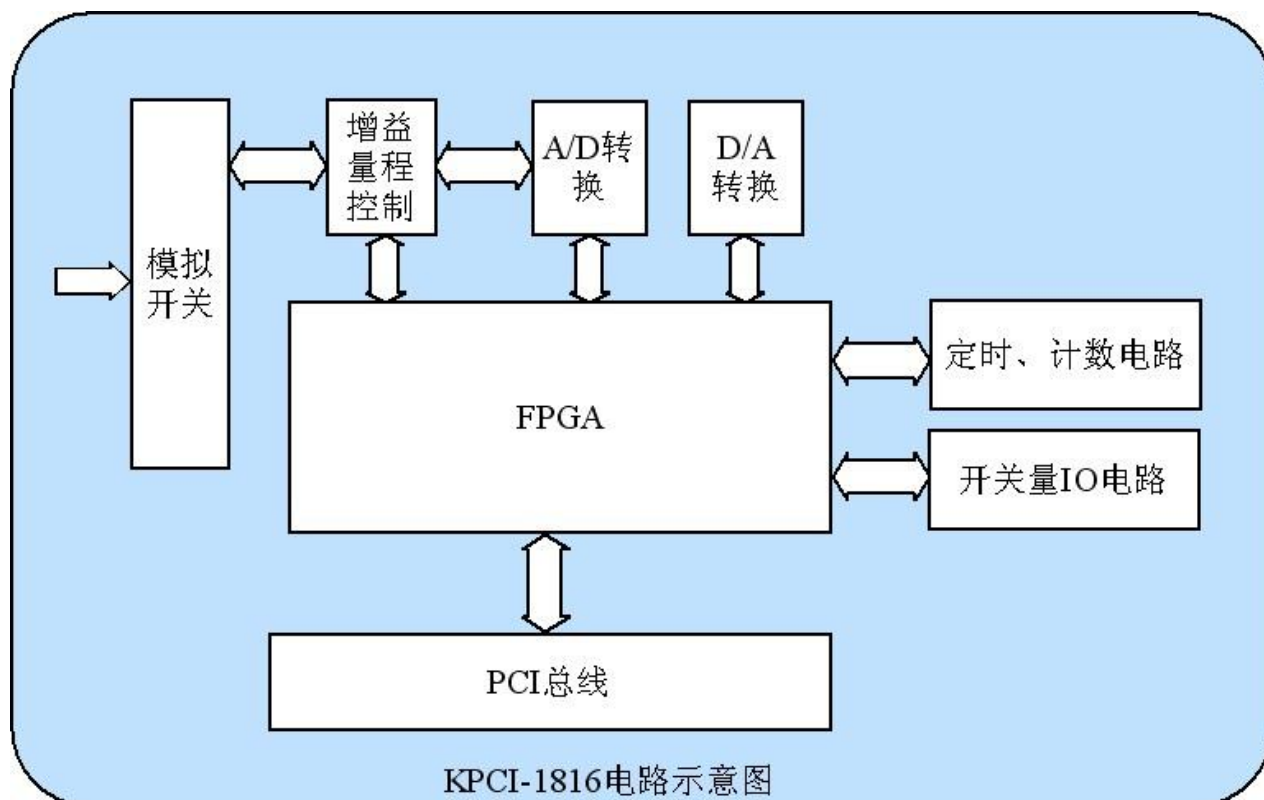
设计有 3 路计数输入电路，计数频率最大可到 5MHz，采用 24 位计数寄存器，最大可计数到十进制的 16777215。可累计上次计数，也可清零重新计数。计满后会有溢出报警。兼容 TTL 电平输入。

三、软件支持

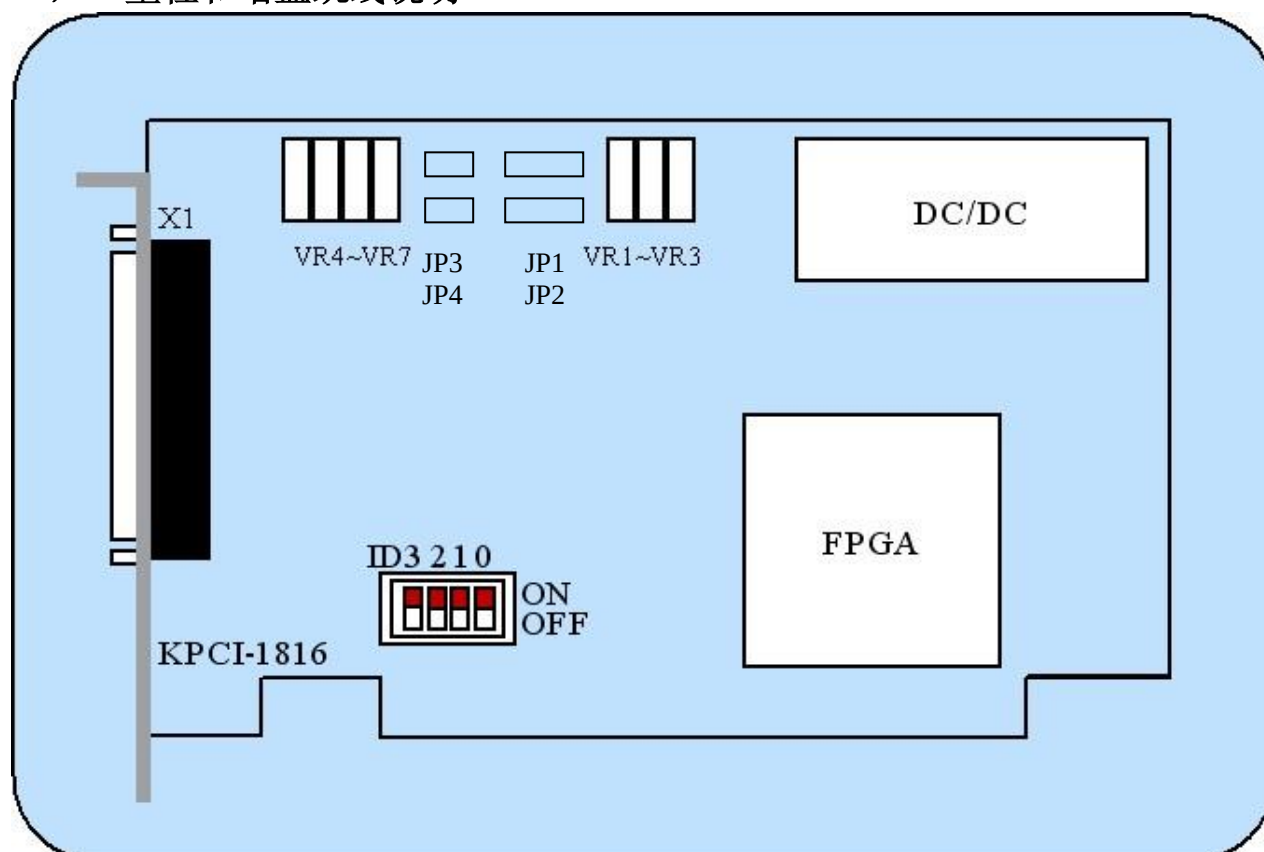
提供 Windows95/98/NT 下的多种语言的驱动，并提供在 VB 和 VC 环境下，开发的示例程序（详见软件说明部分）。

第二章 元件位置图、信号输出插座和开关跳线选择定义

一， 系统框图



二， 量程和增益跳线说明



JP1、JP3 为第 1 通道 D/A 输出信号量程选择，JP2、JP4 为第 2 通道 D/A 输出信号量程选择每个通道可单独控制。

VR1 为 A/D 通道的满度调整电位器。

VR2 为 A/D 通道的零点调整电位器。

VR3 为程控增益调整电位器。

VR4 为 D/A 通道 1 零点调节电位器。

VR5 为 D/A 通道 1 满度调节电位器。

VR6 为 D/A 通道 2 零点调节电位器。

VR7 为 D/A 通道 2 满度调节电位器。

三、信号输入插座定义

3.1 模拟量输入、输出接口信号定义（输入输出插座为 DB37 孔式弯针插座）

插座引脚号	引脚定义	插座引脚号	引脚定义
1	Ain1 (In1+)	20	Ain 2 (In1-)
2	Ain3 (In2+)	21	Ain 4 (In2-)
3	Ain 5 (In3+)	22	Ain 6 (In3-)
4	Ain 7 (In4+)	23	Ain 8 (In4-)
5	Ain 9 (In5+)	24	Ain 10 (In5-)
6	Ain 11 (In6+)	25	Ain 12 (In6-)
7	Ain 13 (In7+)	26	Ain 14 (In7-)
8	Ain 15 (In8+)	27	Ain 16 (In8-)
9	AGND	28	AGND
10	AGND	29	DA_V_Out1
11	AGND	30	DA_V_Out2
12	AGND	31	未用
13	未用	32	未用
14	未用	33	DA_I_Out1+
15	DA_I_Out1 -	34	DA_I_Out2+
16	DA_I_Out2 -	35	未用
17	未用	36	未用
18	未用	37	TR 外触发信号
19	外触发信号地		

3.2 计数器,开关量输入输出接口信号定义（插座为 40P 双排针插座）

插座引脚号	引脚定义	插座引脚号	引脚定义
1	Din1(开关量输入)	2	Din2(开关量输入)
3	Din3	4	Din4
5	Din5	6	Din6
7	Din7	8	Din8
9	Din9	10	Din10
11	Din11	12	Din12
13	Din13	14	Din14
15	Din15	16	Din16

17	Dout1(开关量输出)	18	Dout2(开关量输出)
19	Dout3	20	Dout4
21	Dout5	22	Dout6
23	Dout7	24	Dout8
25	Dout9	26	Dout10
27	Dout11	28	Dout12
29	Dout13	30	Dout14
31	Dout15	32	Dout16
33	CLK 1	34	CLK2
35	CLK3	36	OUT1
37	OUT2	38	OUT3
39	数字地	40	数字地

Ain 1~Ain 16: 单端模拟信号输入通道

In1+ ~In8+ : 双端模拟信号输入正端

In1- ~ In8- : 双端模拟信号输入负端

AGND: 模拟地

DA_V_Out1, 2: 模拟信号(电压)输出端

DA_I_Out1+, 2+: 模拟信号(电流)输出正端

DA_I_Out1-, 2-: 模拟信号(电流)输出负端

Din1 ~ Din16 : 开关量输入通道

Dout1 ~ Dout16 : 开关量输出通道

CLK1, CLK2, CLK3: 计数器输入端

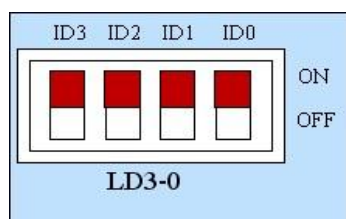
OUT1, OUT2, OUT3: 定时/计数器结束点输出端

TR: 外部触发信号, 当TR有一符合TTL电平的由低至高的变化(上升沿)时,

KPCI-1816卡将按预先设定的采集通道总数进行采集, 直至采集结束。

程序举例见软件说明书相应部分。

四、用四位拨码开关 LD3-0 可用来设置板卡标识 ID, 当用户在同一计算机中使用多块 KPCI-1816 构建自己的系统时, ID 设置功能极为有用。用户可方便的在硬件配置和软件编程过程中区分和访问每块采集卡。



ID3	ID2	ID1	ID0	Board ID
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8

1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

注意： ON 为 0 OFF 为 1

五、模拟输入信号的连接方式

5.1 单端输入方式：

KPCI-1816板均可按图5.1连接成模拟电压单端输入方式，16通道模拟输入信号连接到Ain 1~Ain 16端，其公共地连接到AGND端，注意：为了抑制噪声干扰，所有信号源接地端都应该一点接**AGND**。(见图5.1)

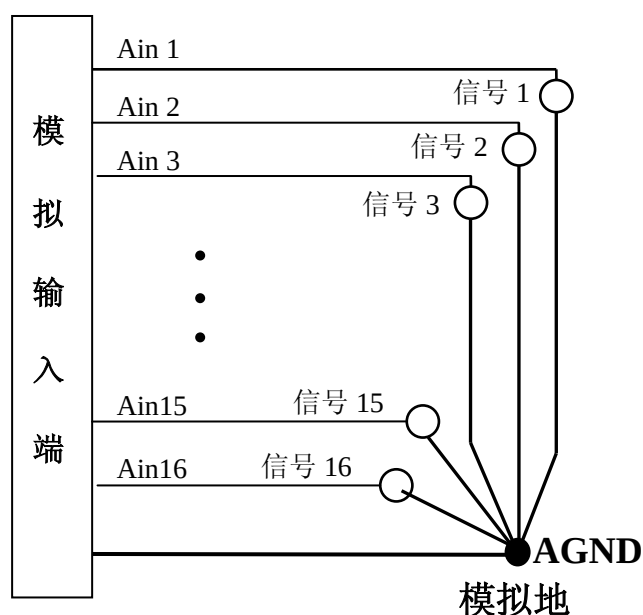


图 5.1 单端输入方式

5.2 差分输入方式：

KPCI-1816板可按图5.2连接成模拟电压差分输入方式，可以有效抑制共模干扰信号，提高采集精度。8通道模拟输入信号正端接到In1+~In8+端，其模拟输入信号负端接到In1-~In8- 端，（见图5.2）

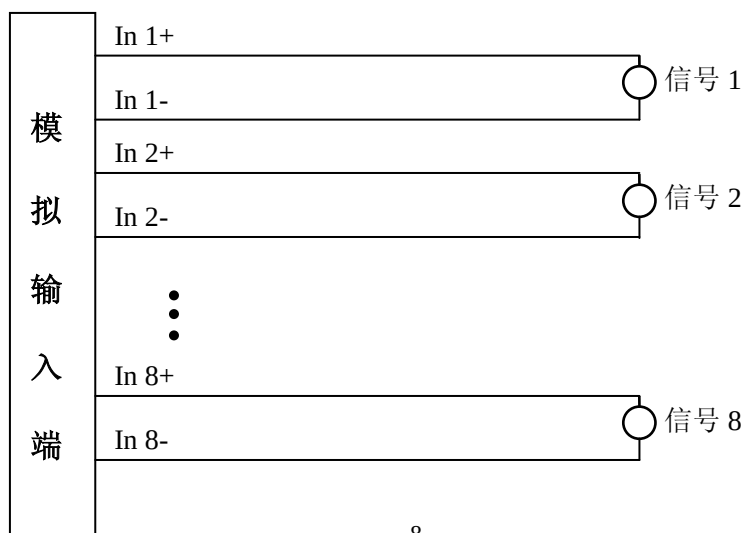
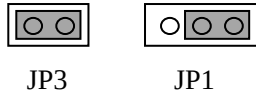


图 5.2 差分输入方式

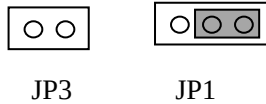
六、模拟量输出信号的连接方式

6.1 模拟量输出量程选择：

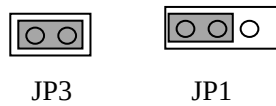
a) D/A 通道 1 输出量程是 0~5V



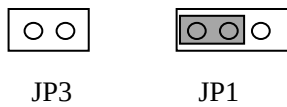
b) D/A 通道 1 输出量程是 0~10V



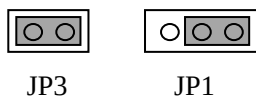
c) D/A 通道 1 输出量程是 -5V~+5V



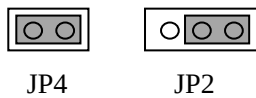
d) D/A 通道 1 输出量程是 -10V~+10V



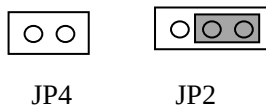
e) D/A 通道 1 输出量程是 4~20mA



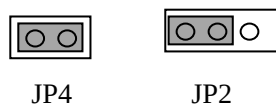
f) D/A 通道 2 输出量程是 0~5V



g) D/A 通道 2 输出量程是 0~10V



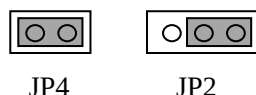
h) D/A 通道 2 输出量程是 -5V~+5V



i) D/A 通道 2 输出量程是-10V~+10V



j) D/A 通道 2 输出量程是 4~20mA



6.2 电压输出连接

KPCI-816 卡用于电压输出时，负载一端接电压输出端（DA_V_Out），另一端接模拟地，提供给负载的最大电流为 10mA，电压输出不需要外接电源。

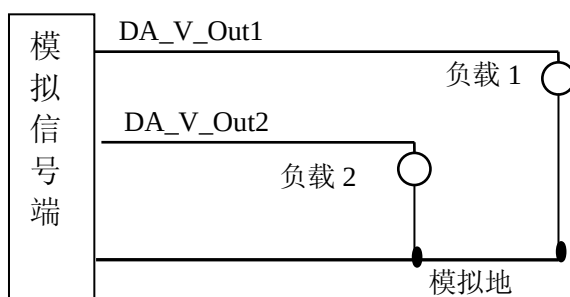


图 6.1 电压输出方式

6.3 电流输出连接

电流输出方式可以直接驱动 II、III 型执行仪表；供电电源外接机外 +24V(或+12V) 电源。其连接使用方法见下图

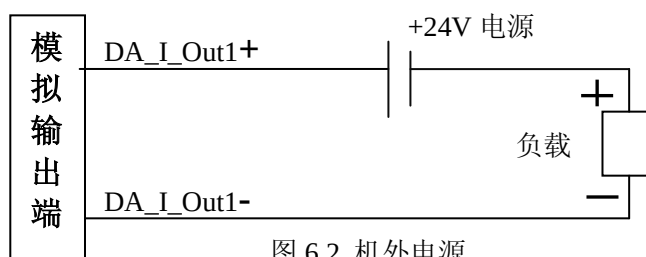


图 6.2 机外电源

特别注意：工作在电流模式下，D/A 通道的跳线必须设在 0~5V 的量程。电流输出时，不要将负载悬空。对于未使用的通道，请将跳线设置成电压输出方式。

6.4 DA 输出工作模式：

D/A 的两个通道有两种工作模式，一种是软件模式，就是操作一次，输出一个电压或电流值，两个通道独立工作。另一种是 FIFO 模式，要输出的数据保存在 FIFO 中，可以输出连续波形，FIFO 大小为 2K，两个 D/A 通道可同时为 FIFO 模式也可单独为 FIFO 模式，当两个通道同时为 FIFO 模式下，共用一个 FIFO。在 FIFO 模式下，D/A 的转换速率可以设定，可以选择输出方波、正弦波、方波三种波形。

七、定时器

设计有 3 路定时器输出电路，可实现 1 μ s 到 2s 的定时控制。支持两种工作方式

1，单一定时方式

当没有启动定时和定时结束后输出保持高电平，在启动定时期间，输出为低电平。

2，方波输出方波

可输出占空比为 1: 1 的任意周期方波，周期范围在 1 μ s 到 1s，兼容 5V TTL 电平输出。

八、计数器

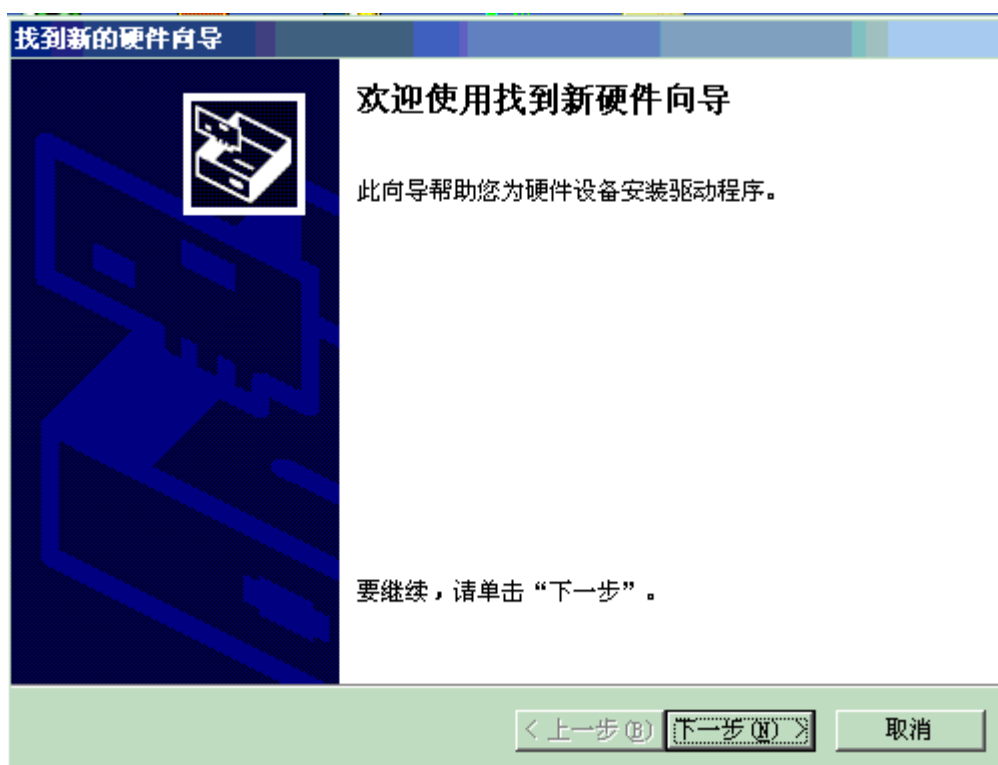
设计有 3 路计数输入电路，计数频率最大可到 5MHz,采用 24 位计数寄存器，最大可计数到十进制的 16777215。可累计上次计数，也可清零重新计数。计满后会有溢出报警。计数输入为 5V TTL 电平输入。

第三章 KPCI-1816 设备驱动程序安装

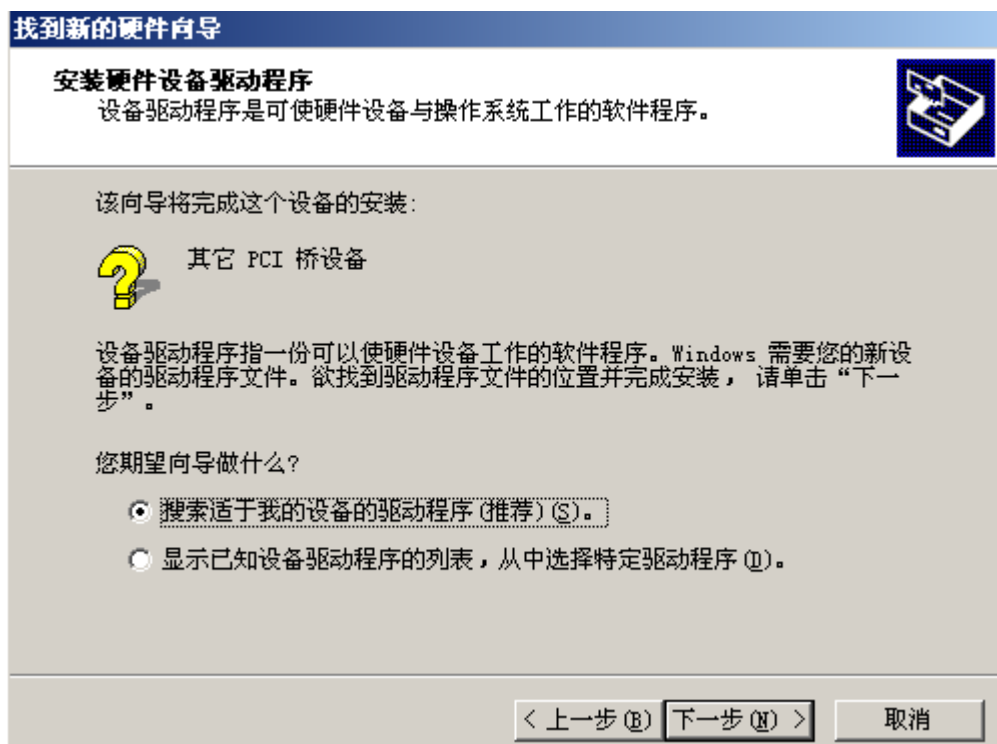
一、安装步骤

第一步 将 KPCI-1816 卡按硬件要求插入计算机主板上的任意一个 PCI 插槽中，并将其固定好，连接好其外接设备后，打开计算机电源，启动 Windows2000(XP)系统。

第二步 如果您正确地插好了 PCI 设备，Windows 系统在启动过程中便会发现这个新的 PCI 设备，并弹出 [欢迎使用找到新的硬件向导] 对话框，单击 [下一步] 按钮



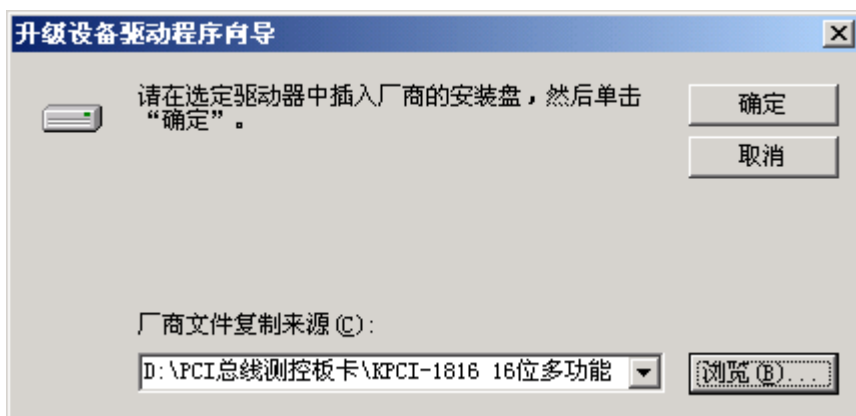
第三步 弹出[安装硬件设备驱动程序]对话框，在对话框中单击 [搜索适用于我的设备的驱动程序(推荐)(S)] 单选框，然后单击 [下一步] 按钮



第四步 选定“指定位置”对话，然后单击“下一步”。



第五步 然后单击“浏览(R)”，将路径定位在光盘上的“PCI 总线测控板卡\KPCI-1816 16 位多功能数据采集卡\驱动程序”路径下，选择 KPCI1816.inf 文件，然后单击“确定”按钮。

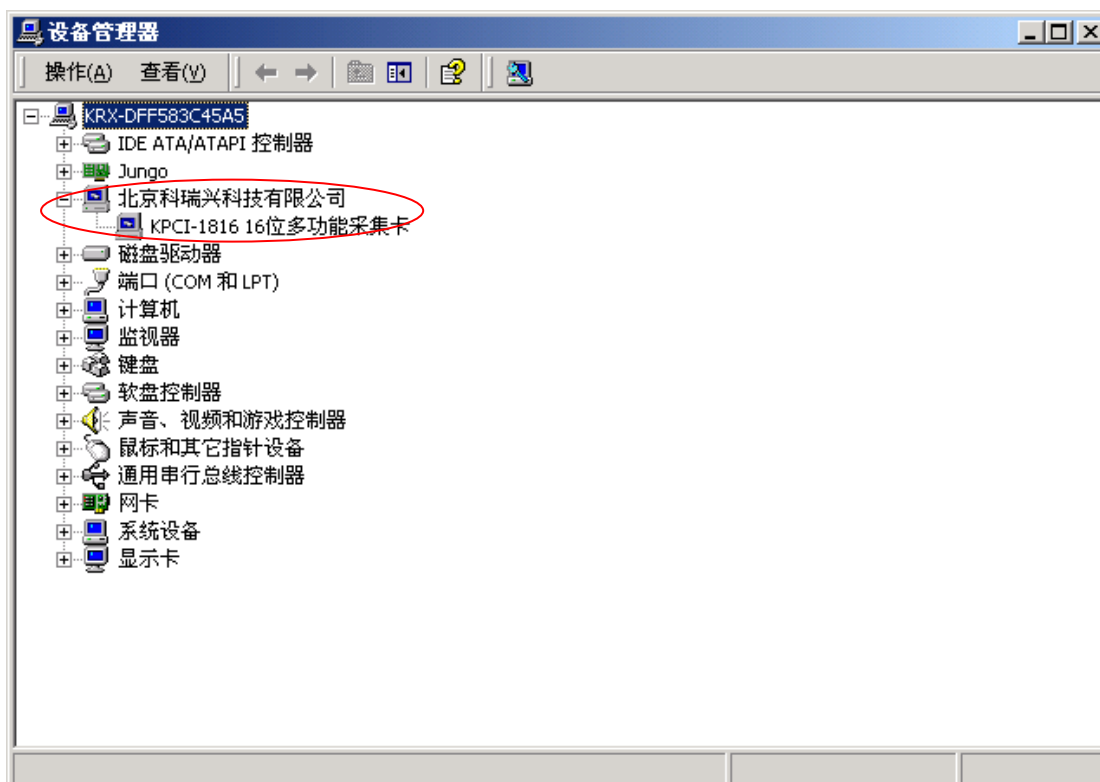


第六步 [找到新的硬件向导], 此步骤可能会出现 Windows 安装驱动程序的进度状态窗口, 用户稍等片刻, 然后出现[完成找到新硬件向导]对话框, 单击 [完成]按钮。



二、安装结果验证

进入 Windows2000 [控制面板] 窗口, 双击 [系统] 图标, 弹出 [系统 特性] 对话框, 在对话框中单击 [硬件] 标签页, 然后单击 [设备管理器]按钮, 进入 [设备管理器] 窗口, 在 [本地计算机] 列表中检查是否有“北京科瑞兴业科技有限公司”显示, 如下图所示。若有, 表示 KPCI-1816 卡的驱动程序已成功安装, 否则, 说明您的安装过程出现了问题, 请试着再安装, 或向硬件供应商求助。



如果在 KPCI-1816 设备上出现黄色感叹号，可运行驱动程序文件夹中的 **SETUP.bat**，或把 **KPCI1816S.SYS** 文件拷贝到 **windows\system32\drivers** 目录下，重新扫描硬件。

第三节 PCI 设备软件测试系统的介绍

当您正确完成了第一或二节中的工作后，您便可以到光盘中“PCI 总线测控板卡\KPCI-1816 16 位多功能数据采集卡\编程示例”文件夹，根据工作模式选择相应的程序，进行板卡测试。当用户自己编制应用程序时，也可以参照这里 VB 和 VC 示例程序来编程。

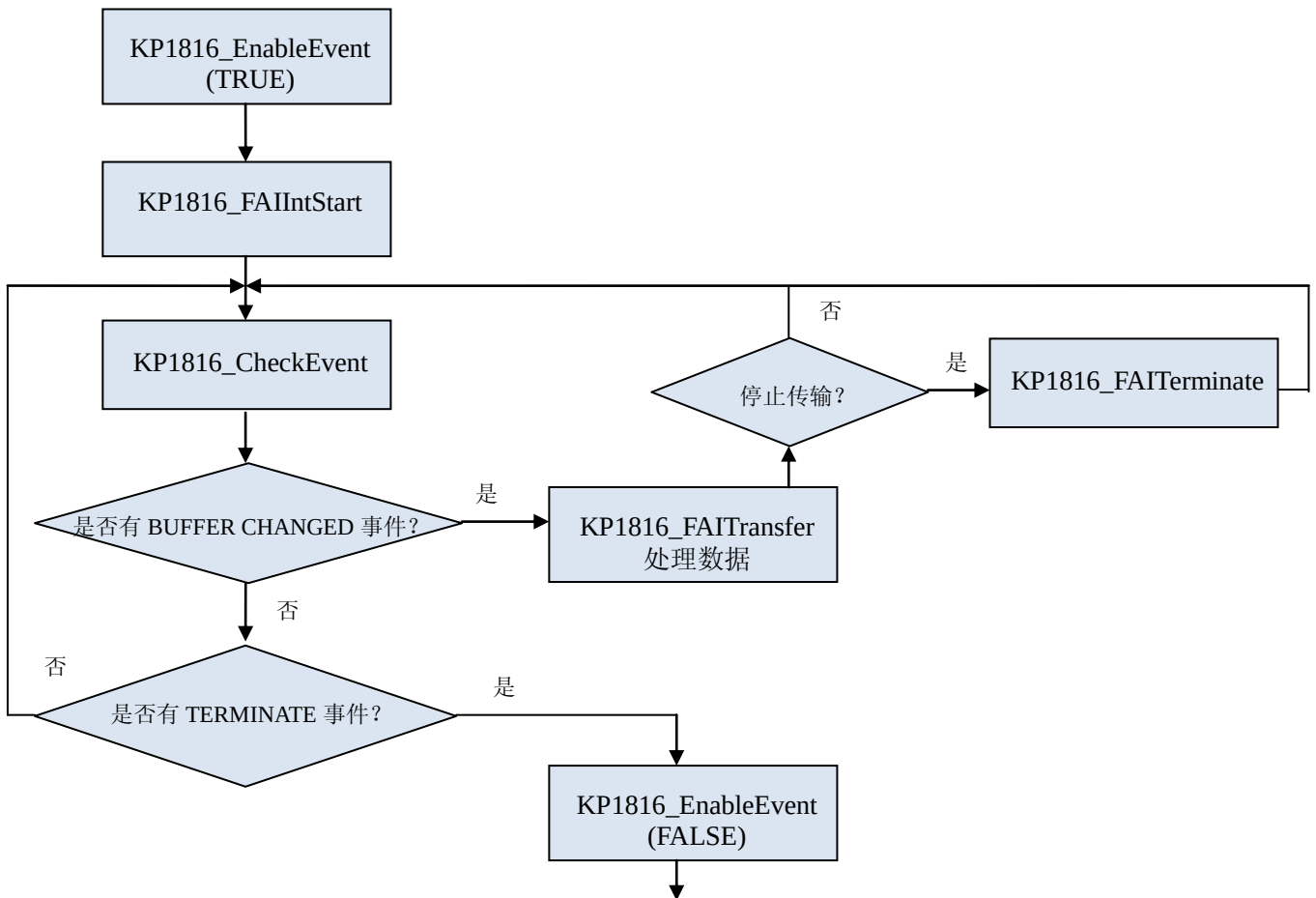
第四章 函数模块调用说明及编程实例

光盘中提供了两个分别在 VB 和 VC 下开发的示例程序，给出了 KPCI-1816 卡的各种触发方式、相关参数的设置过程和数据读取方法。用户可以参照相应程序段根据实际需要利用函数库中提供的函数设计自己的软件，而对数据采集过程中不处理其他任务的用户，也可以直接利用示例程序完成数据。初次使用动态链接库的用户还可以在程序中找到动态链接库的调用方法。为方便用户分析，示例程序以工程的形式提供了所有的资源和代码。用户在编译时需要注意重新指定动态连接库的路径，或把函数库拷到指定的位置。

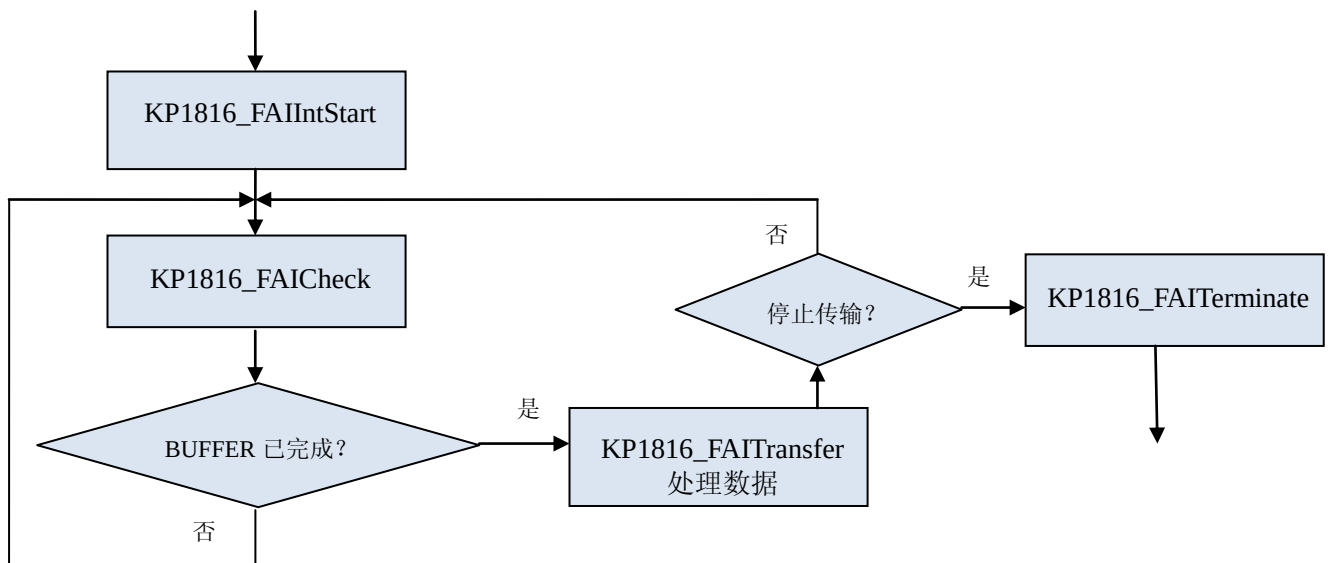
一、函数调用流程：

1、模拟量采集中断触发机制

a) 事件通知模式（推荐使用）



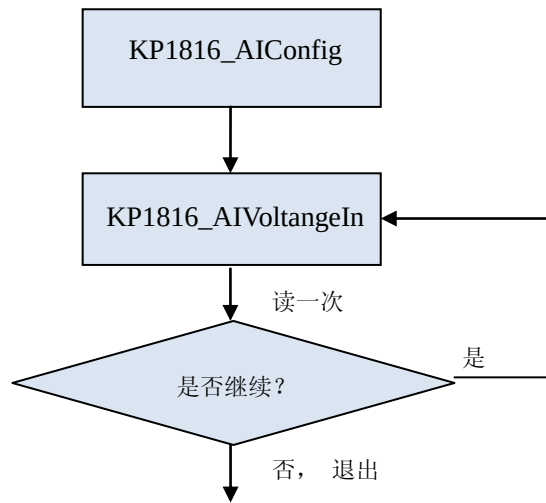
b) 非事件通知模式



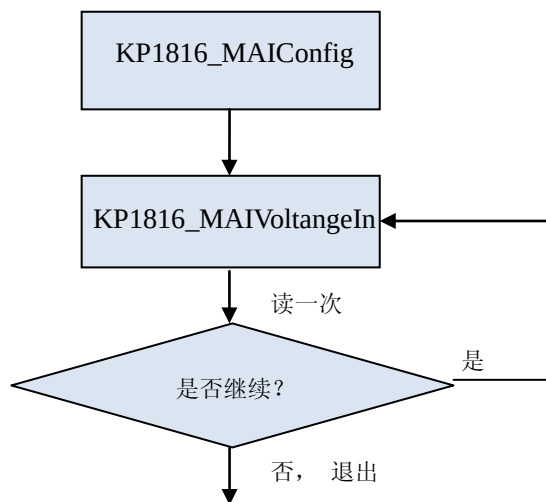
注意：以上流程图是中断触发的单通道的情况下数据采集的流程图，多通道则将KP1816_FAIntStart改成KP1816_FAIntScanStart

2、模拟量采集软件触发机制

a) 单通道采集



b) 多通道采集



二 函数列表

KPCI-1816 函数简表	
函数名	函数功能
KP1816_DeviceOpen	打开要操作的设备
KP1816_SelectDevice	选择要操作的设备的设备号
KP1816_DeviceClose	关闭指定的设备
KP1816_DeviceGetFeatures	获取设备特征
KP1816_DeviceConfig	配置和校验设备
KP1816_AIConfig	配置指定的单个模拟量输入通道
KP1816_AIGetConfig	获取指定的单个模拟量输入通道配置
KP1816_AIBinaryIn	读取指定的单个通道模拟量输入的二进制值
KP1816_AIScale	转换二进制值为电压值
KP1816_AIVoltageIn	读取模拟量输入的电压值
KP1816_MAIConfig	配置指定的多个模拟量输入通道
KP1816_MAIBinaryIn	读取指定的多个通道模拟量输入的二进制值
KP1816_MAIVoltageIn	读取多个模拟量输入通道的电压值
KP1816_ReadPortByte	从指定 I/O 端口读一个字节的数据
KP1816_WritePortByte	写一个字节的数据的到指定的 I/O 端口
KP1816_ReadPortWord	从指定 I/O 端口读一个字的数据
KP1816_WritePortWord	写一个字的数据到指定的 I/O 端口
KP1816_CheckEvent	检查是否有设定的事件发生
KP1816_EnableEvent	启动或停止事件机制
KP1816_GetFIFOSize	获取设备 FIFO 大小
KP1816_FAIIntStart	开始单个通道的模拟量采集
KP1816_FAIIntScanStart	开始多个通道的模拟量采集
KP1816_FAITransfer	把采集到的数据传输到指定的缓存
KP1816_FAICheck	检查模拟量输入的状态
KP1816_ClearOverrun	清除 Overrun 标志
KP1816_FAITerminate	停止当前模拟量输入
KP1816_AOVolCurOut	启动一次模拟量输出操作
KP1816_AOScale	转换模拟量输出数据为二进制值
KP1816_FAOIntScanStart	启动高速模拟量输出
KP1816_FAOCheck	检测高速模拟量输出的当前状态
KP1816_FAOStop	停止高速模拟量输出
KP1816_DioReadBit	读取指定通道的输入开关量
KP1816_DioWriteBit	写指定通道的输出开关量
KP1816_DioReadWord	读取某几个通道的输入开关量
KP1816_DioWriteWord	写入某几个通道的输出开关量
KP1816_DioGetCurrentDOWord	回读输出的开关量
KP1816_StartCounter	启动计数器
KP1816_StopCounter	停止计数器
KP1816_ReadCounterValue	读取计数器的值
KP1816_ClearCounter	清除计数器的值
KP1816_StartTimer	启动定时器

KP1816_StopTimer	停止定时器
KP1816_ReadTimerFlag	读取定时器的标识位
KP1816_GetErrorMessage	根据错误代码返回错误信息
KP1816_GetAddress	该函数用于 VB，用于返回数据地址

三、函数说明：

1、设备操作

a. KP1816_DeviceOpen

Visual C++:

LRESULT KP1816_DeviceOpen (ULONG ulDeviceNum, PVOID *lpDev)

Visual Basic:

Declare Function KP1816_DeviceOpen Lib "KPCI1816.dll" (ByVal ulDeviceNum As Long, lpDev As Long) As Long

功能：该函数打开要操作的设备。

参数：ulDeviceNum: 是 Device Number（板卡上拨码开关的数值）。

lpDev: 是返回的设备指针（后面的函数调用中都要用到此设备指针）

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

b. KP1816_SelectDevice

Visual C++:

LRESULT KP1816_SelectDevice(HWND hCaller, PULONG ulDevNum)

Visual Basic:

Declare Function KP1816_SelectDevice Lib "KPCI1816.dll" (ByVal hCaller As Long, ByVal ulDevNum As Long) As Long

功能：该函数选择要操作设备的设备号。

参数：hCaller: 调用者的窗口句柄；

ulDevNum: 是返回的设备号（拨码开关）。

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

注意：当知道 **Device Number**（板卡上拨码开关的数值）时可以不用调用此函数。

c. KP1816_DeviceClose

Visual C++:

LRESULT KP1816_DeviceClose (PVOID lpDev)

Visual Basic:

Declare Function KP1816_DeviceClose Lib "KPCI1816.dll"(ByVal lpDev As Long) As Long

功能：该函数关闭设备。

参数：lpDev: 设备指针。

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

d. KP1816_DeviceGetFeatures

Visual C++:

LRESULT KP1816_DeviceGetFeatures(PVOID lpDev, LPT_DeviceGetFeatures lpDevFeatures)

Visual Basic:

Declare Function KP1816_DeviceGetFeatures Lib "KPCI1816.dll" (ByVal lpDev As Long, lpDevFeatures As PT_DeviceGetFeatures) As Long

功能: 该函数获取设备特征。

参数: lpDev: 设备指针;

lpDevFeatures: 是 PT_DeviceGetFeatures 结构的指针, 该结构包含了设备特征参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

e. KP1816_DeviceConfig

Visual C++:

LRESULT KP1816_DeviceConfig (PVOID lpDev, HWND hCaller)

Visual Basic:

Declare Function KP1816_DeviceConfig Lib "KPCI1816.dll" (ByVal lpDev As Long, ByVal hCaller As Long) As Long

功能: 该函数配置和校验设备。

参数: lpDev: 设备指针;

hCaller: 是调用该函数程序的窗口句柄。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

2、模拟量输入操作

a. KP1816_AIConfig

Visual C++:

LRESULT KP1816_AIConfig (PVOID lpDev, LPT_AIConfig lpConfig)

Visual Basic:

Declare Function KP1816_AIConfig Lib "KPCI1816.dll" (ByVal lpDev As Long, ByRef lpAIConfig As PT_AIConfig) As Long

功能: 该函数配置指定的单个模拟量输入通道。

参数: lpDev: 设备指针;

lpConfig: 是 PT_AIConfig 结构的指针, 该结构包含模拟量输入通道配置参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

b. KP1816_AIGetConfig

Visual C++:

LRESULT KP1816_AIGetConfig (PVOID lpDev, LPT_AIGetConfig lpAIGetConfig)

Visual Basic:

Declare Function KP1816_AIGetConfig Lib "KPCI1816.dll" (ByVal lpDev As Long, ByRef lpAIGetConfig As PT_AIGetConfig) As Long

功能: 该函数获取指定的单个模拟量输入通道的配置

参数: lpDev: 设备指针;

lpAIGetConfig: 是 PT_AIGetConfig 结构的指针, 该结构包含要模拟量输入通道配置的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

c. KP1816_AIBinaryIn

Visual C++:

LRESULT KP1816_AIBinaryIn (PVOID lpDev, LPT_AIBinaryIn lpAIBinaryIn)

Visual Basic:

Declare Function KP1816_AIBinaryIn Lib "KPCI1816.dll" (ByVal lpDev As Long, ByRef pAIBinaryIn As PT_AIBinaryIn) As Long

功能: 该函数读取指定的单个通道模拟量输入的二进制值

参数: lpDev: 设备指针;

lpAIBinaryIn: 是 PT_AIBinaryIn 结构的指针, 该结构包含要获取输入的通道、二进制值等的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

d. KP1816_AIScale

Visual C++:

LRESULT KP1816_AIScale (PVOID lpDev, LPT_AIScale lpAIScale)

Visual Basic:

Declare Function KP1816_AIScale Lib "KPCI1816.dll" (ByVal lpDev As Long, lpAIScale As PT_AIScale) As Long

功能: 该函数转换读取到的二进制值为电压值

参数: lpDev: 设备指针;

lpAIScale: 是 PT_AIScale 结构的指针, 该结构包含转换需要的参数和转换后的数值, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

e. KP1816_AIVoltageIn

Visual C++:

LRESULT KP1816_AIVoltageIn (PVOID lpDev, LPT_AIVoltageIn lpAIVoltageIn)

Visual Basic:

Declare Function KP1816_AIVoltageIn Lib "KPCI1816.dll" (ByVal lpDev As Long, lpAIVoltageIn As PT_AIVoltageIn) As Long

功能: 该函数读取模拟量输入的电压值

参数: lpDev: 设备指针;

lpAIVoltageIn: 是 PT_AIVoltageIn 结构的指针, 该结构包含单个模拟量通道电压输入的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

f. KP1816_MAIConfig

Visual C++:

`LRESULT KP1816_MAIConfig (PVOID lpDev, LPT_MAIConfig lpMAIConfig)`

Visual Basic:

`Declare Function KP1816_MAIConfig Lib "KPCI1816.dll" (ByVal lpDev As Long, lpMAIConfig As PT_MAIConfig) As Long`

功能: 该函数配置指定的多个模拟量输入通道

参数: lpDev: 设备指针;

lpMAIConfig: 是 PT_MAIConfig 结构的指针, 该结构包含配置多个模拟量通道的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

g. KP1816_MAIBinaryIn

Visual C++:

`LRESULT KP1816_MAIBinaryIn (PVOID lpDev, LPT_MAIBinaryIn lpMAIBinaryIn)`

Visual Basic:

`Declare Function KP1816_MAIBinaryIn Lib "KPCI1816.dll" (ByVal lpDev As Long, lpMAIBinaryIn As PT_MAIBinaryIn) As Long`

功能: 该函数读取指定的多个通道模拟量输入的二进制值

参数: lpDev: 设备指针;

lpMAIBinaryIn: 是 PT_MAIBinaryIn 结构的指针, 该结构包含要获取输入的起始通道、通道数以及二进制值数组等的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

h. KP1816_MAIVoltageIn

Visual C++:

`LRESULT KP1816_MAIVoltageIn (PVOID lpDev, LPT_MAIVoltageIn lpMAIVoltageIn)`

Visual Basic:

`Declare Function KP1816_MAIVoltageIn Lib "KPCI1816.dll" (ByVal lpDev As Long, lpMAIVoltageIn As PT_MAIVoltageIn) As Long`

功能: 该函数读取多个模拟量通道输入的电压

参数: lpDev: 设备指针;

lpMAIVoltageIn: 是 PT_MAIVoltageIn 结构的指针, 该结构包含多个模拟量通道同时电压输入的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

3、I/O 端口操作

a. KP1816_ReadPortByte

Visual C++:

`LRESULT KP1816_ReadPortByte(PVOID lpDev, LPT_ReadPortByte lpReadPortByte)`

Visual Basic:

`Declare Function KP1816_ReadPortByte Lib "KPCI1816.dll" (ByVal lpDev As Long, lpWritePortByte As PT_ReadPortByte) As Long`

功能：该函数读一个字节的 I/O 端口数据

参数：lpDev: 设备指针;

lpReadPortByte: 是 PT_ReadPortByte 结构的指针, 该结构包含读取一字节端口数据的参数, 结构定义请参照数据结构说明。

返回值：如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

b. KP1816_WritePortByte

Visual C++:

LRESULT KP1816_WritePortByte(PVOID lpDev, LPT_WritePortByte lpWritePortByte)

Visual Basic:

Declare Function KP1816_WritePortByte Lib "KPCI1816.dll" (ByVal lpDev As Long, lpWritePortWord As PT_WritePortByte) As Long

功能：该函数写一个字节数据到 I/O 端口

参数：lpDev: 设备指针;

lpWritePortByte: 是 PT_WritePortByte 结构的指针, 该结构包含写一个字节数据到端口的参数, 结构定义请参照数据结构说明。

返回值：如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

c. KP1816_ReadPortWord

Visual C++:

LRESULT KP1816_ReadPortWord(PVOID lpDev, LPT_ReadPortWord lpReadPortWord)

Visual Basic:

Declare Function KP1816_ReadPortWord Lib "KPCI1816.dll" (ByVal lpDev As Long, lpWritePortByte As PT_ReadPortWord) As Long

功能：该函数从 I/O 端口读一个字的数据

参数：lpDev: 设备指针;

lpReadPortWord: 是 PT_ReadPortWord 结构的指针, 该结构包含从指定的 I/O 端口读取一个字数据的参数, 结构定义请参照数据结构说明。

返回值：如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

d. KP1816_WritePortWord

Visual C++:

LRESULT KP1816_WritePortWord(PVOID lpDev, LPT_WritePortWord lpWritePortWord)

Visual Basic:

Declare Function KP1816_WritePortWord Lib "KPCI1816.dll" (ByVal lpDev As Long, lpWritePortWord As PT_WritePortWord) As Long

功能：该函数写一个字节数据到 I/O 端口

参数：lpDev: 设备指针;

lpWritePortWord: 是 PT_WritePortWord 结构的指针, 该结构包含写一个字的数据到指定 I/O 端口的参数, 结构定义请参照数据结构说明。

返回值：如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

4、高速数据采集操作

a. KP1816_CheckEvent

Visual C++:

LRESULT KP1816_CheckEvent (PVOID lpDev, LPT_CheckEvent lpCheckEvent)

Visual Basic:

Declare Function KP1816_CheckEvent Lib "KPCI1816.dll" (ByVal lpDev As Long, lpCheckEvent As PT_CheckEvent) As Long

功能: 该函数检查是否有设定的事件发生

参数: lpDev: 设备指针;

lpCheckEvent: 是 PT_CheckEvent 结构的指针, 该结构包含检测事件状态的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

b. KP1816_EnableEvent

Visual C++:

LRESULT KP1816_EnableEvent (PVOID lpDev, LPT_EnableEvent lpEnableEvent)

Visual Basic:

Declare Function KP1816_EnableEvent Lib "KPCI1816.dll" (ByVal lpDev As Long, lpEnableEvent As PT_EnableEvent) As Long

功能: 该函数启动或停止事件机制

参数: lpDev: 设备指针;

lpEnableEvent: 是 PT_EnableEvent 结构的指针, 该结构包含使能事件状态的参数, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

c. KP1816_GetFIFOSize

Visual C++:

LRESULT KP1816_GetFifoSize(PVOID lpDev, PULONG lSize)

Visual Basic:

Declare Function KP1816_GetFifoSize Lib "KPCI1816.dll" (ByVal lpDev As Long, lSize As Long) As Long

功能: 该函数获取设备 FIFO 大小

参数: lpDev: 设备指针;

lSize: 是 long 的指针, 该参数返回 FIFO 的大小。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

d. KP1816_FAIntStart

Visual C++:

LRESULT KP1816_FAIntStart (PVOID lpDev, LPT_FAIntStart lpFAIntStart)

Visual Basic:

Declare Function KP1816_FAIntStart Lib "KPCI1816.dll" (ByVal lpDev As Long, lpFAIntStart As PT_FAIntStart) As Long

功能: 该函数开始单个通道的模拟量采集

参数: lpDev: 设备指针;

lpFAIntStart: 是 PT_FAIntStart 结构的指针, 该结构包含启动单通道模拟量数据采集的参数, 结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

e. KP1816_FAIntScanStart

Visual C++:

LRESULT KP1816_FAIntScanStart (PVOID lpDev, LPT_FAIntScanStart
lpFAIntScanStart)

Visual Basic:

Declare Function KP1816_FAIntScanStart Lib "KPCI1816.dll" (ByVal lpDev As Long,
lpFAIntScanStart As PT_FAIntScanStart) As Long

功能：该函数开始多个通道的模拟量采集

参数：lpDev: 设备指针；

lpFAIntScanStart: 是 PT_FAIntScanStart 结构的指针，该结构包含启动多个通道模拟量数据采集的参数，结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

f. KP1816_FAITransfer

Visual C++:

LRESULT KP1816_FAITransfer (PVOID lpDev, LPT_FAITransfer lpFAITransfer)

Visual Basic:

Declare Function KP1816_FAITransfer Lib "KPCI1816.dll" (ByVal lpDev As Long,
lpFAITransfer As PT_FAITransfer) As Long

功能：该函数把采集到的数据传输到指定的缓存

参数：lpDev: 设备指针；

lpFAITransfer: 是 PT_FAITransfer 结构的指针，该结构包含传输数据到缓存的参数，结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

g. KP1816_FAICheck

Visual C++:

LRESULT KP1816_FAICheck (PVOID lpDev, LPT_FAICheck lpFAICheck)

Visual Basic:

Declare Function KP1816_FAICheck Lib "KPCI1816.dll" (ByVal lpDev As Long,
lpFAICheck As PT_FAICheck) As Long

功能：该函数检查模拟量输入的状态

参数：lpDev: 设备指针；

lpFAICheck: 是 PT_FAICheck 结构的指针，该结构包含一些状态的指针，结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

h. KP1816_ClearOverrun

Visual C++:

LRESULT KP1816_ClearOverrun(PVOID lpDev)

Visual Basic:

Declare Function KP1816_ClearOverrun Lib "KPCI1816.dll" (ByVal lpDev As Long)
As Long

功能：该函数清除 Overrun 标志

参数：lpDev: 设备指针；

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

i. KP1816_FAITerminate

Visual C++:

LRESULT KP1816_FAITerminate (PVOID lpDev)

Visual Basic:

Declare Function KP1816_FAITerminate Lib "KPCI1816.dll" (ByVal lpDev As Long)
As Long

功能：该函数停止当前模拟量输入

参数：lpDev: 设备指针

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

5、模拟量输出操作

a. KP1816_AOVolCurOut

Visual C++:

LRESULT KP1816_AOVolCurOut (PVOID lpDev, LPT_AOVolCurOut
lpAOVolCurOut)

Visual Basic:

Declare Function KP1816_AOVolCurOut Lib "KPCI1816.dll" (ByVal lpDev As Long,
lpAOVolCurOut As PT_AOVolCurOut) As Long

功能：该函数启动一次模拟量输出操作

参数：lpDev: 设备指针；

lpAOVolCurOut: 是 PT_AOVolCurOut 结构的指针，结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

6、高速模拟量输出

a. KP1816_AOScale

Visual C++:

LRESULT KP1816_AOScale (PVOID lpDev, LPT_AOScale lpAOScale)

Visual Basic:

Declare Function KP1816_AOScale Lib "KPCI1816.dll" (ByVal lpDev As Long,
lpAOScale As PT_AOScale) As Long

功能：该函数转换模拟量输出数据为二进制值

参数：lpDev: 设备指针；

lpAOScale: 是 PT_AOScale 结构的指针，结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

b. KP1816_FAOIntScanStart

Visual C++:

LRESULT KP1816_FAOIntScanStart (PVOID lpDev, LPT_FAOIntScanStart lpFAOIntScanStart)

Visual Basic:

Declare Function KP1816_FAOIntScanStart Lib "KPCI1816.dll" (ByVal lpDev As Long, lpFAOIntScanStart As PT_FAOIntScanStart) As Long

功能: 该函数启动高速模拟量输出

参数: lpDev: 设备指针;

lpFAOIntScanStart: 是 PT_FAOIntScanStart 结构的指针, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;
如果未成功, 则返回错误代码。

c. KP1816_FAOCheck

Visual C++:

LRESULT KP1816_FAOCheck (PVOID lpDev, LPT_FAOCheck lpFAOCheck)

Visual Basic:

Declare Function KP1816_FAOCheck Lib "KPCI1816.dll" (ByVal lpDev As Long, lpFAOCheck As PT_FAOCheck) As Long

功能: 该函数检测当前高速模拟量输出状态

参数: lpDev: 设备指针;

lpFAOCheck: 是 PT_FAOCheck 结构的指针, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;
如果未成功, 则返回错误代码。

d. KP1816_FAOStop

Visual C++:

LRESULT KP1816_FAOStop (PVOID lpDev)

Visual Basic:

Declare Function KP1816_FAOStop Lib "KPCI1816.dll" (ByVal lpDev As Long) As Long

功能: 该函数停止高速模拟量输出

参数: lpDev: 设备指针;

返回值: 如果执行成功, 则返回 0;
如果未成功, 则返回错误代码。

7、开关量输入/输出操作

a. KP1816_DioReadBit

Visual C++:

LRESULT KP1816_DioReadBit(PVOID lpDev, LPT_DioReadBit lpDioReadBit)

Visual Basic:

Declare Function KP1816_DioReadBit Lib "KPCI1816.dll" (ByVal lpDev As Long, lpDioReadBit As PT_DioReadBit) As Long

功能: 该函数读取指定通道的输入开关量。

参数: lpDev: 设备指针;

lpDioReadBit: 是 PT_DioReadBit 结构的指针, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;
如果未成功, 则返回错误代码。

b. KP1816_DioWriteBit

Visual C++:

LRESULT KP1816_DioWriteBit (PVOID lpDev, LPT_DioWriteBit lpDioWriteBit)

Visual Basic:

Declare Function KP1816_DioWriteBit Lib "KPCI1816.dll" (ByVal lpDev As Long, lpDioWriteBit As PT_DioWriteBit) As Long

功能: 该函数写指定通道的输出开关量。

参数: lpDev: 设备指针;

lpDioWriteBit: 是 PT_DioWriteBit 结构的指针, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

c. KP1816_DioReadWord

Visual C++:

LRESULT KP1816_DioReadWord (PVOID lpDev, LPT_DioReadWord lpDioReadWord)

Visual Basic:

Declare Function KP1816_DioReadWord Lib "KPCI1816.dll" (ByVal lpDev As Long, lpDioReadWord As PT_DioReadWord) As Long

功能: 该函数读取某几个通道的输入开关量。

参数: lpDev: 设备指针;

lpDioReadWord: 是 PT_DioReadWord 结构的指针, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

d. KP1816_DioWriteWord

Visual C++:

LRESULT KP1816_DioWriteWord(PVOID lpDev, LPT_DioWriteWord lpDioWriteWord)

Visual Basic:

Declare Function KP1816_DioWriteWord Lib "KPCI1816.dll" (ByVal lpDev As Long, lpDioWriteWord As PT_DioWriteWord) As Long

功能: 该函数写入某几个通道的输出开关量。

参数: lpDev: 设备指针;

lpDioWriteWord: 是 PT_DioWriteWord 结构的指针, 结构定义请参照数据结构说明。

返回值: 如果执行成功, 则返回 0;

如果未成功, 则返回错误代码。

e. KP1816_DioGetCurrentDOWord

Visual C++:

LRESULT KP1816_DioGetCurrentDOWord(PVOID lpDev, USHORT far * value)

Visual Basic:

Declare Function KP1816_DioGetCurrentDOWord Lib "KPCI1816.dll" (ByVal lpDev As Long, ByRef value As Integer) As Long

功能: 该函数回读输出的开关量。

参数: lpDev: 设备指针;

value: 是所读出的输出开关量的值。

返回值: 如果执行成功，则返回 0；
如果未成功，则返回错误代码。

8、定时器/计数器操作

a. KP1816_StartCounter

Visual C++:

LRESULT KP1816_StartCounter(PVOID lpDev, USHORT usChanMask)

Visual Basic:

Declare Function KP1816_StartCounter Lib "KPCI1816.dll" (ByVal lpDev As Long, ByVal usChanMask As Integer) As Long

功能: 该函数启动计数器。

参数: lpDev: 设备指针；

usChanMask: 是启动计数器的掩码，0、1、2 有效。为 1 时启动计数。

返回值: 如果执行成功，则返回 0；
如果未成功，则返回错误代码。

b. KP1816_StopCounter

Visual C++:

LRESULT KP1816_StopCounter (PVOID lpDev, USHORT usChanMask)

Visual Basic:

Declare Function KP1816_StopCounterLib "KPCI1816.dll" (ByVal lpDev As Long, ByVal usChanMask As Integer) As Long

功能: 该函数停止计数器。

参数: lpDev: 设备指针；

usChanMask: 是启动计数器的掩码，0、1、2 有效。为 1 时停止计数。

返回值: 如果执行成功，则返回 0；
如果未成功，则返回错误代码。

c. KP1816_ReadCounterValue

Visual C++:

LRESULT KP1816_ReadCounterValue(PVOID lpDev, USHORT usChannel , ULONG far * usCount)

Visual Basic:

Declare Function KP1816_ReadCounterValue Lib "KPCI1816.dll" (ByVal lpDev As Long, ByVal usChannel As Integer, ByRef usCount As Long) As Long

功能: 该函数读取指定的计数器的值。

参数: lpDev: 设备指针；

usChannel: 是指要读取的计数器；

usCount: 读取出来的计数器的值。

返回值: 如果执行成功，则返回 0；
如果未成功，则返回错误代码。

d. KP1816_ClearCounter

Visual C++:

LRESULT KP1816_ClearCounter (PVOID lpDev, USHORT usChannel)

Visual Basic:

Declare Function KP1816_ClearCounter Lib "KPCI1816.dll" (ByVal lpDev As Long, ByVal usChannel As Integer) As Long

功能：该函数清除指定的计数器的值。

参数：lpDev: 设备指针;
usChannel: 是指要清除的计数器。

返回值：如果执行成功，则返回 0;
如果未成功，则返回错误代码。

e. KP1816_StartTimer

Visual C++:

LRESULT KP1816_StartTimer(PVOID lpDev, LPT_StartTimer lpStartTimer)

Visual Basic:

Declare Function KP1816_StartTimer Lib "KPCI1816.dll" (ByVal lpDev As Long, lpStartTimer As PT_StartTimer) As Long

功能：该函数启动定时器。

参数：lpDev: 设备指针;
lpStartTimer: 是 PT_StartTimer 结构的指针，结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0;
如果未成功，则返回错误代码。

f. KP1816_StopTimer

Visual C++:

LRESULT KP1816_StopTimer (PVOID lpDev, USHORT usChanMask)

Visual Basic:

Declare Function KP1816_StopTimer Lib "KPCI1816.dll" (ByVal lpDev As Long, ByVal usChanMask As Integer) As Long

功能：该函数停止定时器。

参数：lpDev: 设备指针;
usChanMask: 是定时器的掩码，0、1、2 有效。为 1 时停止定时器定时。

返回值：如果执行成功，则返回 0;
如果未成功，则返回错误代码。

g. KP1816_ReadTimerFlag

Visual C++:

LRESULT KP1816_ReadTimerFlag(PVOID lpDev, USHORT far * usFlag)

Visual Basic:

Declare Function KP1816_ReadTimerFlag Lib "KPCI1816.dll" (ByVal lpDev As Long, ByRef usFlag As Integer) As Long

功能：该函数读取定时器的标识位。

参数：lpDev: 设备指针;
usFlag: 是指定时器的完成标志。

返回值：如果执行成功，则返回 0;
如果未成功，则返回错误代码。

9、其他操作

a. KP1816_GetErrorMessage

Visual C++:

LRESULT KP1816_GetErrorMessage(LRESULT lError, TCHAR * lpMsg)

Visual Basic:

Declare Function KP1816_GetErrorMessage Lib "KPCI1816.dll" (ByVal lError As Long, ByVal lpMsg As String) As Long

功能：该函数根据错误代码返回错误信息

参数：lError: 错误代码, lpMsg: 指向错误信息字符串的指针。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

b. KP1816_GetAddress

Declare Function KP1816_GetAddress Lib "KPCI1816.dll" (lpVoid As Any) As Long

功能：该函数用于 VB，返回数据地址

参数：lpVoid: 可以是任意的类型，需要返回其地址

返回值：返回该数据的地址。

四、数据结构说明

1、设备操作

a) PT

_DeviceGetFeatures

typedef struct tagPT_DeviceGetFeatures

{

LPDEVFEATURES buffer; USHORT size;

}PT_DeviceGetFeatures, FAR * LPT_DeviceGetFeatures;

Member Description:

名称	方向	类型	描述
buffer	Output	DEVFEATURES 的指针	指向设备特征结构的指针
size	Input	USHORT	特征结构的数据长度

b) DEVFEATURES

typedef struct tagDEVFEATURES

{

USHORT usMaxAIDiffChl; USHORT usMaxAISiglChl;

USHORT usMaxTimerChl; USHORT usMaxAlarmChl;

USHORT usNumADBit; USHORT usNumADByte;

USHORT usNumGain ; GAINLIST glGainList[32];

DWORD dwPermutation[4];

} DEVFEATURES, FAR * LPDEVFEATURES;

Member Description:

名称	方向	类型	描述
usMaxAIDiffChl	Output	USHORT	差分模拟量输入通道的最大个数
usMaxAISiglChl	Output	USHORT	单端模拟量输入通道的最大个数
usMaxTimerChl	Output	USHORT	计数器/时钟的最大个数
usMaxAlarmChl	Output	USHORT	报警通道的最大个数
usNumADBit	Output	USHORT	A/D 转换的位数
usNumADByte	Output	USHORT	A/D 转换所需字节个数
usNumGain	Output	USHORT	输入范围的最大个数
glGainList	Output	GAINLIST 数组	所有支持输入范围
dwPermutation	Output	DWORD	设备所支持功能

Note:

dwPermutation 的定义:

Bit 0: Software AI
 Bit 1: DMA AI
 Bit 2: Interrupt AI
 Bit 3: Condition AI
 Bit 4: Software AO
 Bit 5: DMA AO
 Bit 6: Interrupt AO
 Bit 7: Condition AO
 Bit 8: Software DI
 Bit 9: DMA DI
 Bit 10: Interrupt DI
 Bit 11: Condition DI
 Bit 12: Software DO
 Bit 13: DMA DO
 Bit 14: Interrupt DO
 Bit 15: Condition DO
 Bit 16: High Gain
 Bit 17: Auto Channel Scan
 Bit 18: Pacer Trigger
 Bit 19: External Trigger
 Bit 20: Down Counter
 Bit 21: Dual DMA
 Bit 22: Monitoring
 Bit 23: Qcounter

c) GAINLIST

```
typedef struct tagGAINLIST
```

```
{
    USHORT usGainCde;    FLOAT fMaxGainVal;
    FLOAT fMinGainVal;   CHAR szGainStr[16];
} GAINLIST;
```

Member Description:

名称	方向	类型	描述
usGainCde	Output	USHORT	该模拟量输入范围的代码
fMaxGainVal	Output	Float	该模拟量输入范围的最大值
fMinGainVal	Output	Float	该模拟量输入范围的最小值
szGainStr	Output	Char 数组	该模拟量输入范围的字符描述

2、模拟量输入操作

a) PT_AIConfig

```
typedef struct tagPT_AIConfig
```

```
{
    USHORT DasChan;    USHORT DasGain;
} PT_AIConfig, * LPT_AIConfig;
```

Member Description:

名称	方向	类型	描述
DasChan	Input	USHORT	采样通道
DasGain	Input	USHORT	采样值范围的代码

b) PT_AIGetConfig

```
typedef struct tagPT_AIGetConfig
```

```
{
    LPDEVCONFIG_AI buffer;    USHORT size;
} PT_AIGetConfig, * LPT_AIGetConfig;
```

Member Description:

名称	方向	类型	描述
buffer	Output	DEVONFIG_AI 指针	指向设备配置结构的指针
size	Input	USHORT	该配置结构的大小

c) PT_AIBinaryIn

```
typedef struct tagPT_AIBinaryIn
```

```
{
    USHORT chan;    USHORT TrigMode;    USHORT *reading;
} PT_AIBinaryIn, * LPT_AIBinaryIn;
```

Member Description:

名称	方向	类型	描述
chan	Input	USHORT	采样通道
TrigMode	Input	USHORT	触发模式。0 为内部触发；1 为外部触发
reading	Output	USHORT 指针	从采样通道读取的原始数据

d) PT_AIScale

```
typedef struct tagPT_AIScale
```

```
{
    USHORT reading;    FLOAT MaxVolt;
    USHORT MaxCount;    USHORT offset;    FLOAT *voltage;
} PT_AIScale, * LPT_AIScale;
```

Member Description:

名称	方向	类型	描述
reading	Input	USHORT	原始数据
MaxVolt	Input	float	该输入范围的最大电压值
MaxCount	Input	USHORT	最大范围值（4095）
offset	Input	USHORT	0 伏特电压的偏移
voltage	Output	（float）浮点数指针	转换后的电压值（伏特）

e) PT_AIVoltageIn

```
typedef struct tagPT_AIVoltageIn
```

```
{
    USHORT chan;    USHORT gain;
    USHORT TrigMode;    FLOAT *voltage;
} PT_AIVoltageIn, * LPT_AIVoltageIn;
```


Member Description:

名称	方向	类型	描述
chan	Input	USHORT	采样通道
gain	Input	USHORT	该模拟量输入范围代码
TrigMode	Input	USHORT	触发模式
voltage	Output	(float) 浮点数指针	输入的电压, 已转换成伏特值的形式

f) PT_MAIConfig

```
typedef struct tagPT_MAIConfig
```

```
{
    USHORT NumChan;          USHORT StartChan;   USHORT *GainArray;
} PT_MAIConfig, * LPT_MAIConfig;
```

Member Description:

名称	方向	类型	描述
NumChan	Input	USHORT	要配置的通道数
StartChan	Input	USHORT	起始通道
GainArray	Input	USHORT 指针	模拟量输入范围代码的数组

g) PT_MAIBinaryIn

```
typedef struct tagPT_MAIBinaryIn
```

```
{
    USHORT NumChan;          USHORT StartChan;
    USHORT TrigMode;         USHORT *ReadingArray;
} PT_MAIBinaryIn, * LPT_MAIBinaryIn;
```

Member Description:

名称	方向	类型	描述
NumChan	Input	USHORT	采样的通道数
StartChan	Input	USHORT	采样的起始通道
TrigMode	Input	USHORT	触发模式
ReadingArray	Output	USHORT 指针	模拟量输入的原始数据数组

h) PT_MAIVoltageIn

```
typedef struct tagPT_MAIVoltageIn
```

```
{
    USHORT NumChan;          USHORT StartChan;
    USHORT *GainArray;       USHORT TrigMode;   FLOAT *VoltageArray;
} PT_MAIVoltageIn, * LPT_MAIVoltageIn;
```

Member Description:

名称	方向	类型	描述
NumChan	Input	Unsigned short	采样的通道数
StartChan	Input	USHORT	采样的起始通道
GainArray	Input	USHORT 指针	各个通道的模拟量输入范围代码
TrigMode	Input	USHORT	触发模式
VoltageArray	Output	(float)浮点数指针	模拟量输入的电压数据数组

3、I/O 端口操作

a) PT_ReadPortByte

```
typedef struct tagPT_ReadPortByte
```

```
{
    USHORT port;          USHORT *ByteData;
} PT_ReadPortByte, * LPT_ReadPortByte;
```

Member Description:

名称	方向	类型	描述
port	Input	USHORT	端口地址
ByteData	Output	USHORT 指针	从端口读取的字节数据

b) PT_WritePortByte

```
typedef struct tagPT_WritePortByte
```

```
{
    USHORT port;          USHORT ByteData;
} PT_WritePortByte, * LPT_WritePortByte;
```

Member Description:

名称	方向	类型	描述
port	Input	USHORT	端口地址
ByteData	Input	USHORT 指针	写入端口的字节数据

c) PT_ReadPortWord

```
typedef struct tagPT_ReadPortWord
```

```
{
    USHORT port;          USHORT *WordData;
} PT_ReadPortWord, * LPT_ReadPortWord;
```

Member Description:

名称	方向	类型	描述
port	Input	USHORT	端口地址
WordData	Output	USHORT 指针	从端口读取的一个字的数据

d) PT_WritePortWord

```
typedef struct tagPT_WritePortWord
```

```
{
    USHORT port;          USHORT WordData;
} PT_WritePortWord, * LPT_WritePortWord;
```

Member Description:

名称	方向	类型	描述
port	Input	USHORT	端口地址
WordData	Input	USHORT 指针	写入端口的一个字的数据

4、高速数据采集操作

a) PT_CheckEvent

```
typedef struct tagPT_CheckEvent
```

```
{
    USHORT *EventType;      DWORD Milliseconds;
} PT_CheckEvent, * LPT_CheckEvent;
```

Member Description:

名称	方向	类型	描述
EventType	Output	DWORD	驱动支持的事件类型，请参考 PT_EnableEvent
Milliseconds	Input	USHORT 指针	事件超时的毫秒数

b) PT_EnableEvent

```
typedef struct tagPT_EnableEvent
```

```
{
    USHORT EventType;          USHORT Enabled;  USHORT Count;
} PT_EnableEvent, * LPT_EnableEvent;
```

Member Description:

名称	方向	类型	描述
EventType	Input	USHORT	希望触发的事件类型
Enabled	Input	USHORT	使能或取消触发的事件类型。1 为使能；0 为取消
Count	Input	USHORT	希望事件触发的次数

注意：EventType 的定义：

- 0 位：中断事件
- 1 位：buffer change 事件
- 2 位：termination 事件
- 3 位：overrun 事件

c) PT_FAIIntStart

```
typedef struct tagPT_FAIIntStart
```

```
{
    USHORT TrigSrc;            DWORD SampleRate;
    USHORT chan;               USHORT gain;
    USHORT *buffer;            ULONG count;
    USHORT cyclic;             USHORT IntrCount;
} PT_FAIIntStart, * LPT_FAIIntStart;
```

Member Description:

名称	方向	类型	描述
TrigSrc	Input	USHORT	触发源: 1 位外部触发源, 0 位内部触发源
SampleRate	Input	DWORD	采样频率
chan	Input	USHORT	采样通道
gain	Input	USHORT	该通道输入范围代码
buffer	Output	USHORT 指针	用户分配的缓存
count	Input	ULONG	采样的次数
cyclic	Input	USHORT	是否循环模式: 1 位循环模式, 0 位非循环模式
IntrCount	Input	USHORT	采样多少次一中断, (只有 FIFO 和中断两种采样模式, 中断为 1, FIFO 则为半满 FIFO 大小)

```
d) PT_FAIntScanStart
typedef struct tagPT_FAIntScanStart
{
    USHORT TrigSrc;           DWORD SampleRate;
    USHORT NumChans;          USHORT StartChan;
    USHORT *GainList;          USHORT *buffer;
    ULONG count;               USHORT cyclic;    USHORT IntrCount;
} PT_FAIntScanStart, * LPT_FAIntScanStart;
```

Member Description:

名称	方向	类型	描述
TrigSrc	Input	USHORT	触发源: 1 位外部触发源, 0 位内部触发源
SampleRate	Input	DWORD	采样频率
NumChans	Input	USHORT	采样通道数
StartChan	Input	USHORT	采样起始通道
GainList	Input	USHORT 指针	各个通道输入范围代码
buffer	Output	USHORT 指针	用户分配的缓存
count	Input	ULONG	采样的次数
cyclic	Input	USHORT	是否循环模式: 1 位循环模式, 0 位非循环模式
IntrCount	Input	USHORT	采样多少次一中断, (只有 FIFO 和中断两种采样模式, 中断为 1, FIFO 则为半满 FIFO 大小)

```
e) PT_FAITransfer
typedef struct tagPT_FAITransfer
{
    USHORT ActiveBuf;          PVOID DataBuffer;
    USHORT DataType;           ULONG start;
    ULONG count;               USHORT *overrun;
} PT_FAITransfer, * LPT_FAITransfer;
```

Member Description:

名称	方向	类型	描述
ActiveBuf	Input	USHORT	Buffer 的类型, 本设备总是为 0
DataBuffer	Output	指向浮点数或 USHORT 的指针	如果 buffer 指针不为空, 当数据类型为 USHORT, 则该 buffer 的长度应该不小于 2*count 。当数据类型为浮点数时, 则该 buffer 的长度应该不小于 4*count 。
DataType	Input	USHORT	数据类型, 0 为原始数据, 1 为电压值 (浮点数)
start	Input	ULONG	从源 buffer 复制到用户 buffer 的起始点
count	Input	ULONG	从源 buffer 复制到用户 buffer 的采样个数
overrun	Output	ULONG 指针	overrun 状态 0- 无 overrun 发生 1- Overrun 发生了

```
f) PT_FAICheck
typedef struct tagPT_FAICheck
{
    USHORT far *ActiveBuf; USHORT far *stopped;
    ULONG far *retrieved; USHORT far *overrun; USHORTfar *HalfReady;
} PT_FAICheck, * LPT_FAICheck;
```

Member Description:

名称	方向	类型	描述
ActiveBuf	Output	USHORT 指针	本设备总是为 0
stopped	Output	USHORT 指针	说明操作是否完成，0 代表未完成，1 代表完成。
retrieved	Output	ULONG 指针	A/D 转换的次数
overrun	Output	USHORT 指针	是否有 overrun 发生，0 无 overrun，1 有 overrun。
HalfReady	Output	USHORT 指针	说明 buffer 状态，0 为数据没准备好；1 为第一次半满 buffer 准备。2 为第二次半满 buffer 准备。

5、模拟量输出操作

```
a) PT_AOVolCurOut
typedef struct tagPT_AOVolCurOut
{
    USHORT chan; FLOAT fAOMaxVal;
    FLOAT fAOMinVal; FLOAT fValue;
}PT_AOVolCurOut, *LPT_AOVolCurOut;
```

Member Description:

名称	方向	类型	描述
chan	Input	USHORT	AO 通道
fAOMaxVal	Input	FLOAT	输出的最大电压或电流值
fAOMinVal	Input	FLOAT	输出的最小电压或电流值
fValue	Input	FLOAT	要输出的电压或电流值

6、开关量输入/输出操作

```
a) PT_DioReadBit
typedef struct tagPT_DioReadBit
{
    USHORT bit; USHORT far * state;
} PT_DioReadBit, * LPT_DioReadBit;
```

Member Description:

名称	方向	类型	描述
bit	Input	USHORT	要读取的 DI 的位
state	Output	USHORT 指针	所读出的 bit 位的值

Member Description:

```
c) PT_DioReadWord
typedef struct tagPT_DioReadWord
{
    USHORT * mask;           USHORT * value;
} PT_DioReadWord, * LPT_DioReadWord;
```

Member Description:

```
d) PT_DioWriteWord
typedef struct tagPT_DioWriteWord
{
    USHORT mask;           USHORT value;
} PT_DioWriteWord, * LPT_DioWriteWord;
```

Member Description:

名称	方向	类型	描述
mask	Input	USHORT	要写入的 DO 值的掩码
value	Input	USHORT	所写入 DO 的值

```
a) PT_StartTimer
typedef struct tagPT_StartTimer
{
    USHORT usMode;           USHORT usMask;
    double  dwMillisecond[3];
} PT_StartTimer, * LPT_StartTimer;
```

Member Description:

名称	方向	类型	描述
usMode	Input	USHORT	定时器的输出模式。低三位有效。改为为 1，输出为方波；为 0，输出高电平。
usMask	Input	USHORT	定时器启动掩码。低三位有效。为 1 时启动定时器。
dwMillisecond	Input	Double 数组	为定时器的时间，单位毫秒。数组 0、1、2 分别代表 0、1、2 三个定时器。

第五章 KPCI-1816 多功能数据采集卡的校准、保修和注意事项

一 应用注意事项：

在公司售出的产品包装中，用户将会在资料光盘中找到本卡的说明书。

在使用KPCI-1816板时，应注意以下问题：

- ① KPCI-1816卡正面的IC芯片不要用手去摸，防止芯片受到静电的损害。
- ② 在使用KPCI-1816卡时，可通过K-801E等信号调理端子板与现场信号连接。
- ③ 用户务必注意电源的开关顺序，使用时要求先开主机电源，后开信号源的电源；先关信号源的电源，后关主机电源。

二、校准：

KPCI-1816板出厂时已经校准，只有当用户使用一段时间后，或者用户认为需要时才做校准。下面以0~10V量程为例，说明校准过程：

准备一块4位半精度以上数字电压表，安装好KPCI-1816，打开主机电源，预热10分钟。

A/D的校准

(一) 单极性输出的校准

- 1) ① 零点调整
选模拟输入的任意两个通道，比如Ain1、Ain2通道，将Ain1通道输入接0伏电压(AGND),在Windows下运行KPCI-1816 测试程序,选择1通道，选择单通道数据采集方式，采集开始后，调整电位器VR1，使显示值为 0。
- ② 满度调整
将 Ain2接正满度电压+9998 mV，再选择2通道，采集程序开始采集后，调整电位器VR2，使显示值为 +9998mV。
重复以上步骤，直到满足要求为止。
- 2) 如果放大器增益选择其他倍数，在相应的通道上，也要接入相应的满度电压。(5 ,2.5, 1.25 或 0.625V)

(二) 双极性输出的校准

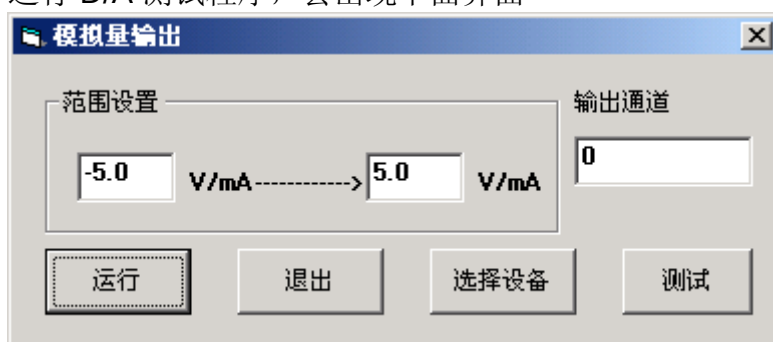
- 1) ① 零点调整
选模拟输入的任意3个通道，比如Ain1、Ain2、Ain3通道，将Ain1通道输入接0伏电压(AGND),在Windows下运行KPCI-1816 测试程序,选择1通道，选择单通道数据采集方式，采集开始后，调整电位器VR3，使显示值为 0。
- ② 满度调整
将 Ain2接正满度电压+4998 mV，再选择2通道，采集程序开始采集后，调整电位器VR2，使显示值为 +4998mV。
将 Ain3接负满度电压-4998 mV，选择3通道，采集程序开始采集后，看显示值是否为 -4998mV。
重复以上步骤，直到满足要求为止。

- 2) 如果放大器增益选择其他倍数，在相应的通道上，也要接入相应的满度电压。（5,2.5, 1.25或 0.625V）

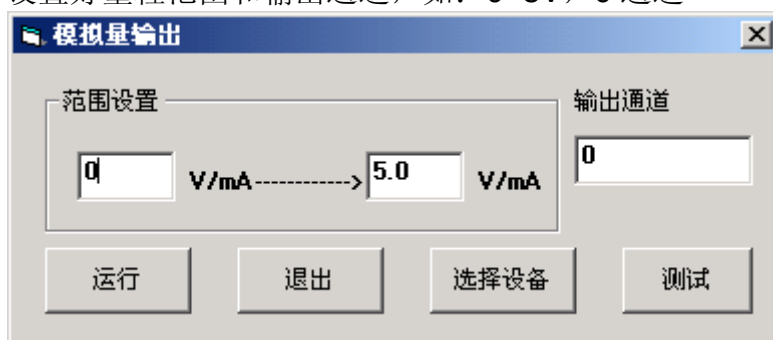
D/A的校准:

凡改变模出的工作方式和量程范围后，如果输出结果误差较大时，需要对模出进行调整。调整时需要注意，一般情况下不需要调整零点，应该首先进行满度调整，尤其是在由 0~10V 方式改变为 0~5V 或 4~20mA 方式时，更应进行此项调整。待满度调整完毕后再观察零点情况并决定是否进行调整。具体调整方式如下：

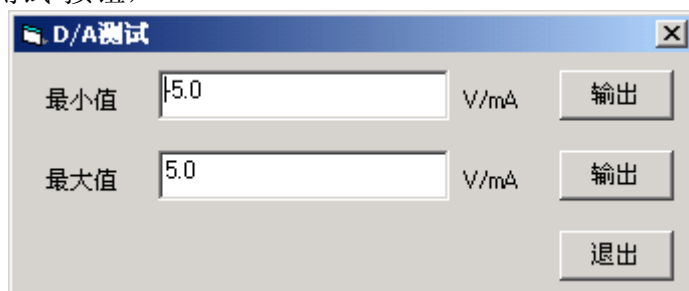
运行 D/A 测试程序，会出现下面界面



设置好量程范围和输出通道，如：0~5V，0 通道



按“测试”按钮，



- ① 零点调整：在单极性方式或双极性输出方式时，最小值输入 0，测量 D/A 输出通道 1，调整电位器 VR4 使其偏差最小。
- ② 电压输出方式满度调整：在零点调整正常情况下，如果满度偏差较大，最大值输入 5，测量 D/A 输出通道 1，调整电位器 VR5 使满度符合要求。
- ③ 用同样方法可以完成其他 3 个输出通道的调整。
零点调整：通道 2 调整电位器 VR6。
满度调整：通道 2 调整电位器 VR7。

三、保修：

KPCI-1816自出厂之日起，两年内凡用户遵守贮存，运输和使用要求，而产品质量低于技术指标的，凭保修卡免费维修。因违反操作规定和要求而造成损坏的，需交纳器件维修费。

四、产品配套清单：

- 4.1 KPCI-1816 16位多功能数据采集卡壹块。
- 4.2 北京科瑞兴业公司产品光盘壹张。
- 4.3 37芯D型插头壹套。
- 4.4 40P扁平带缆插头壹套。