

USB7325 高速光电隔离型模入数据采集模块技术说明书

1. 概述

USB7325 光电隔离型模入数据采集模块适用于提供了 USB 接口的 PC 系列微机，具有真正的热插拔、即插即用 (PnP) 功能。其操作系统可选用目前流行的 Windows 系列、高稳定性的 Unix 等多种操作系统以及专业数据采集分析系统 LabVIEW 等软件环境。在硬件的安装上非常简单，使用时只需将 USB7325 的 USB 接口插入计算机内任何一个 USB 接口插座中，其模入、I/O 信号、脉冲输入及脉冲输出信号均由模块上的双排针插头与外部信号源及设备连接。

模入部分，用户可根据实际需要选择单端或双端输入方式，其 A/D 转换启动方式可以选用车控触发、外部触发两种方式。A/D 转换后的数据结果通过先进先出存储器 (FIFO) 缓存后由 USB 总线读出。本模块的模入部分可以直接与我公司研制的 PS-010B/ PS-011B 等前端信号处理板配套使用，方便地对热电偶、热电阻的微弱信号进行放大，进行温度采集。每块 USB7325 多功能模入接口卡最多可以连接 16 块 PS-010B/ PS-011B 等前端信号处理板，即 256 路模拟信号输入。

开关量部分，本模块有 16 路 TTL 电平的数字量输入和 16 路数字量输出接口。

计数部分，本模块有 3 路 16 位字长的计数/定时器，以及 1MHz 的基准时钟。用户可根据需要时使用计数、定时、测频、脉冲输出等功能。

2. 主要技术指标

2.1 USB 指标:

- 2.1.1 处理器及 USB 接口芯片: CY7C68013
- 2.1.2 通讯方式: USB 接口
- 2.1.3 通讯距离: 小于 5 米
- 2.1.4 通讯协议: USB2.0

2.2 模入部分 (标*为出厂标准状态, 下同):

- 2.2.1 输入通道数: 单端 16 路 *; 双端 8 路
- 2.2.2 输入信号范围: 0~4V, 0~5V, 0~10V*, $\pm 3V$, $\pm 5V$, $\pm 10V$
- 2.2.3 输入阻抗: $\geq 10M\Omega$
- 2.2.4 输入通道选择方式: 单通道程序指定/多通道自动扫描
- 2.2.5 A/D 转换分辨率: A 型 12 位; B 型 16 位
- 2.2.6 A/D 转换器件: ADS7808 (A 型 12 位); ADS7809 (A 型 16 位)
- 2.2.7 A/D 芯片转换时间: $4\mu S$
- 2.2.8 系统最高采集速率: 250KHz / S (所有通道总和)
- 2.2.9 A/D 采样程控频率: 1KHz/5KHz/10KHz/25KHz/50KHz/100KHz/200KHz/250KHz
- 2.2.10 A/D 启动方式: 程控触发/外部触发
- 2.2.11 A/D 转换输出码制: 单极性原码
- 2.2.12 FIFO 存储器容量: $8K \times 16bit$ (全满) / $4K \times 16bit$ (半满)
- 2.2.13 通道切换时间: (模拟开关导通时间 + 放大器建立时间) $\leq 5\mu S$
- 2.2.14 A/D 转换非线性误差: $\pm 1LSB$ (A 型), $\pm 2LSB$ (B 型),

2.3 开关量部分

- 2.3.1 输入路数: 16 路 TTL 电平
- 2.3.2 输出路数: 16 路 TTL 电平

2.4 定时/计数器/测频/脉冲输出部分:

- 2.4.1 16 位字长计数/定时器: 3 路
- 2.4.2 基准时钟: 1MHz, 占空比 50%

- 2.5 电源功耗: $+5V(\pm 10\%) \leq 500mA$ (自供电, 不建议长期使用)
- $+12V(\pm 10\%) \leq 400mA$ (外供电)
- $+24V(\pm 10\%) \leq 200mA$ (外供电)

2.6 使用环境要求：工作温度：10℃~40℃（如需宽温, 订购时说明）

相对湿度：40%~80%

存贮温度：-55℃~+85℃

2.7 尺寸：

裸板尺寸：长×宽=162.9.0mm×102.6mm

外形(模具)尺寸：长×宽×厚=190mm×109mm×36mm

3. 工作原理

USB7325 高速光隔离模入接口模块主要由多路模拟开关选通电路、高精度放大电路、模数转换电路、光电隔离及 DC / DC 电路、先进先出(FIFO)缓冲存储器电路、开关量输入输出电路、定时/计数器电路和接口控制逻辑电路、供电电路等部分组成。

3.1 模入部分

3.1.1 高速多路模拟开关选通电路

本电路由 2 片 ADG508（或同类产品）及跨接选择器 JP5 组成，用以从 16 路单端信号或 8 路双端信号中选择其中一路，送入后端的放大器电路处理。

3.1.2 高速高精度、差分、可编程增益放大器电路

本电路由 2 片高速高精度放大器 AD711(或 LF351)及 1 片高速高精度差分放大器 AMP03 组成，用以对通道开关选中的模拟信号进行变换处理，以提供模数转换电路所需要的信号。

3.1.3 高速模数转换电路

本模块选用 B-B 公司的 A / D 器件 ADS7808（12 bit）或 ADS7809（16 bit）作为本卡的模数转换器件。该器件内部自带采保和精密基准电源。

3.1.4 光电隔离及 DC / DC 电源电路

本卡采用高速光耦对系统总线与模拟信号之间进行光电隔离，以避免相互间的干扰。同时由电源模块及相关的滤波元件组成 DC / DC 电源电路。

3.1.5 先进先出（FIFO）缓冲存储器电路

本电路用于将 A/D 转换的数据结果及通道代码进行缓冲存储。并相应的给出“空”，“半满”和“全满”的标志信号。用户在使用过程中可以随时根据这些标志信号的状态以单次或批量的方式读出 A/D 转换的结果。

3.2 开关量输入输出电路

本模块提供了 16 路非隔离的开关量输入、16 路非隔离的开关量输出信号通道。使用中需注意对这些信号的要求应严格符合 TTL 电平规范。

3.3 计数/定时器部分

计数/定时器电路由 1 片可编程定时/计数器 8254 芯片和基准时钟电路以及有关的跨接选择器组成。可为用户提供 3 个 16 位字长的计数/定时通道和 1MHz、占空比为 50%的基准时钟，用户可外接使用三路计数/定时通道。

3.4 接口控制逻辑电路

接口控制逻辑电路用来将 USB 总线控制逻辑转换成与各种操作相关的控制信号。

3.5 供电电路

供电电路由 DC/DC 芯片 LM2675 及其外围电路组成外供电电路，以满足 USB 总线自供电功率的不足。

4. 安装及使用注意

本模块的安装十分简便，无需将主机机壳打开，也无需关电。本模块有两种供电方式，即自供电方式和外供电方式，若选择自供电方式供电，将本模块插入主机的任何一个空余 USB 接口插座即可；若选择外

供电方式，必须给模块提供 8-40V 的直流电源，然后将模块与计算机用 USB 线连接。

本模块采用的模拟开关是 COMS 电路，容易因静电击穿或过流造成损坏，所以在打开模块盒或用手触摸本模块电路板时，应事先将人体所带静电荷对地放掉，同时应避免直接用手接触器件管脚，以免损坏器件。

本模块 USB 接口允许从计算机上带电插拔。模块上的跨接选择器，使用中应严格按照说明书进行设置操作。设置模块跨接套和安装接口带缆时均应在关电状态下进行。

当模入通道不全部使用时，应将不使用的通道就近对地短接，不要使其悬空，以避免造成通道间串扰和损坏通道。

为保证安全及采集精度，应确保系统地线（计算机及外接仪器机壳）接地良好。特别是使用双端输入方式时，为防止外界较大的共模干扰，应注意对信号线进行屏蔽处理。

5. 使用与操作

5.1 主要可调整元件位置及接插件见图 1。

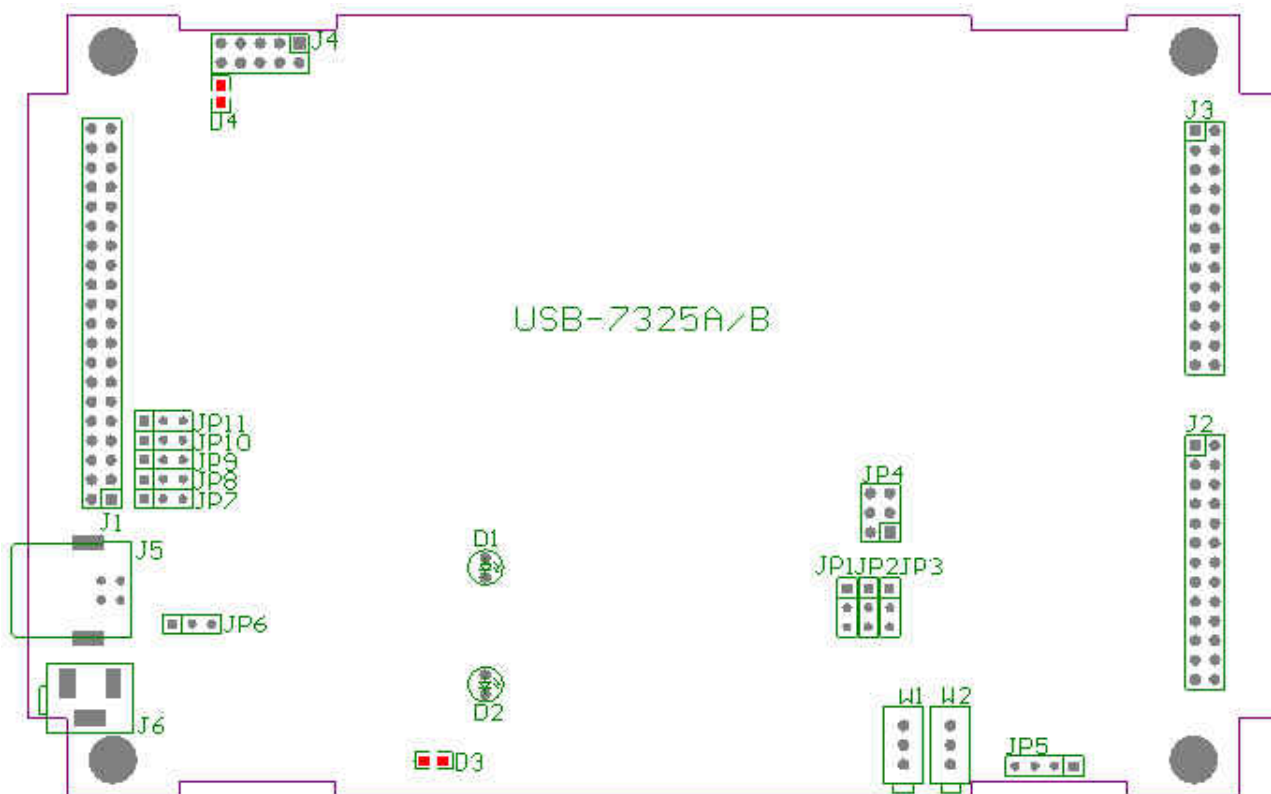


图 1 主要可调整元件位置图

5.2 指示灯 D1、D2、D3、D4:

- D1 亮 指示模块 5V 供电正常
- D2 亮 指示模块 USB 工作正常
- D3 亮 指示模块 3.3V 供电正常
- D4 亮 指示模块 AD FIFO 满溢出(若选择无 FIFO 的 USB7325，此灯状态无意义)

5.3 跨接器的使用:

5.3.1 A/D 量程选择:

本模块由这两个跳线 JP1、JP2、JP3、JP4 组合实现 A/D 量程，其使用方法见图 2。

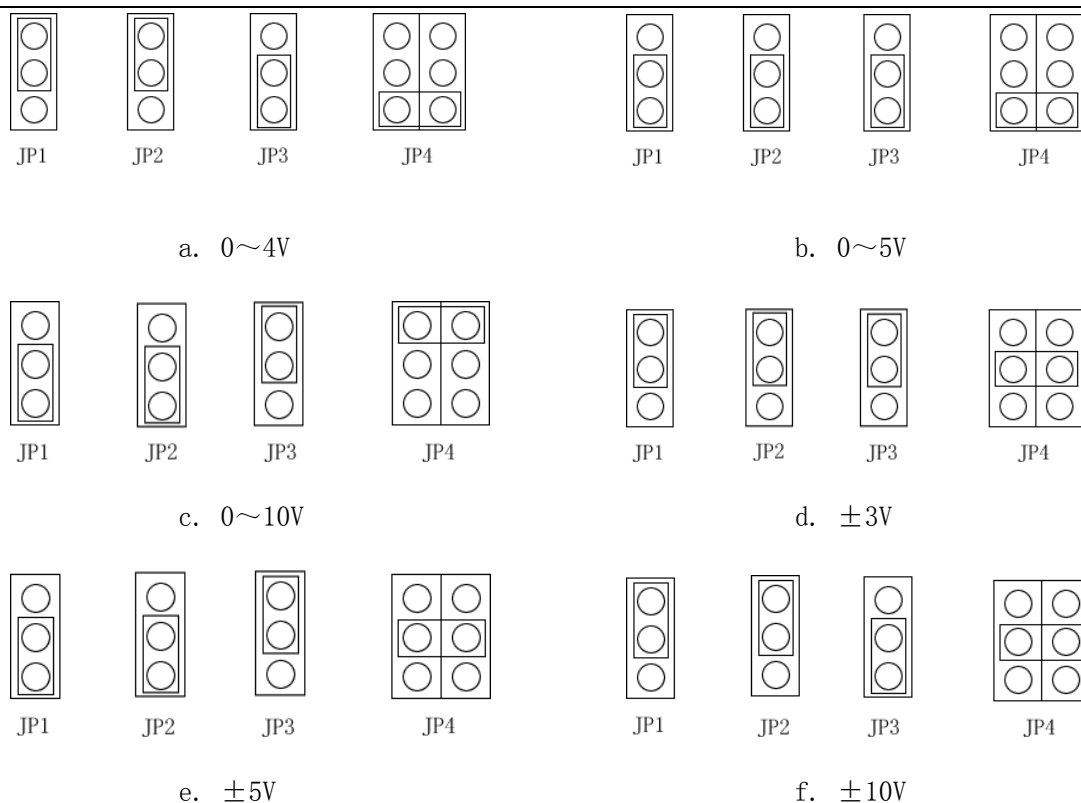


图 2 A / D 量程选择

5.3.2 单端 / 双端方式选择:

JP5 为单端 / 双端方式选择插座, 其使用方法见图 3。



图 3 单 / 双端方式选择

5.3.3 8254 工作方式 (计数、定时、脉冲输出) 的选择

8254 芯片的右下角有一组跨接插座, 即 JP7—JP11, 其作用是为 8254 的 CLK 选择不同的脉冲信号源, 以组成不同的工作方式。JP7—JP11 的定义下图:

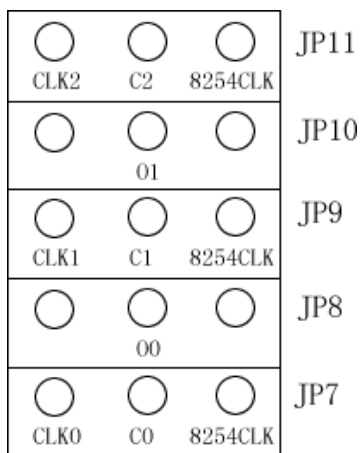


图 4 JP7—JP11 的定义

C_0 、 C_1 、 C_2 表示 8254 芯片的脉冲输入引脚， O_0 、 O_1 、 O_2 表示 8254 芯片的脉冲输出引脚，CLK0、CLK1、CLK2 是外部输入并整形后的现场信号，8254CLK 接板上的 1MHz 内部时钟。比如：用跨接套将 C_1 与 8254CLK 短接，那么表示本组 8254 的第二通道的脉冲输入接到了板上的 1MHz 内部时钟；用跨接套将 C_1 与 O_0 短接，那么表示本组 8254 的第二通道的脉冲输入接到了上一通道的脉冲输出，即所谓的级连；用跨接套将 C_1 与 CLK1 短接，那么表示本组 8254 的第二通道的脉冲输入接到了用户外部输入的脉冲信号源上。

5.3.4 供电方式选择

本模块通过对 JP6 跳线，给用户两种模块供电选择，如图 5：

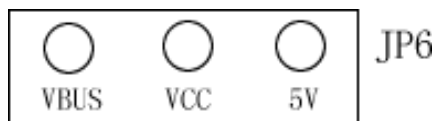


图 5 J6 的定义

将 VCC 针与 5V 针短接，模块通过 JP6 外供电输入插座取电，用户可从 JP6 输入 8 - 40V 的直流电；若将 VCC 针与 VBUS 针短接，模块从计算机的 USB 接口取电。

5.3.5 其它调线

其它调线是调试用的，用户不必关心，请按出厂设置保留原状，不要将它们的任意针短接。

5.4 输入输出接口定义：

5.4.1 模入部分：

本模块 26 芯扁平电缆插座 J2 的信号定义见表 1，用户可根据需要选择连接信号线(单端)或信号线组(双端)。为减少信号杂波串扰和保护通道开关，凡不使用的信号端应就近与模拟地短接，这一点在小信号放大使用时尤其重要。

表 1 J2 模拟输入信号端口定义(括号内表示双端方式)

插座引脚号	信号定义	插座引脚号	信号定义
1	模拟地)	2	模拟地
3	CH1 (CH1+)	4	CH2 (CH2+)
5	CH3 (CH3+)	6	CH4 (CH4+)
7	CH5 (CH5+)	8	CH6 (CH6+)
9	CH7 (CH7+)	10	CH8 (CH8+)
11	CH9 (CH1-)	12	CH10 (CH2-)
13	CH11 (CH3-)	14	CH12 (CH4-)
15	CH13 (CH5-)	16	CH14 (CH6-)
17	CH15 (CH7-)	18	CH16 (CH8-)
19	模拟地	20	模拟地
21	NC (空脚)	22	NC (空脚)
23	外触发输入地(TG)	24	外触发输入地(TG)
25	外触发输入(ET)	26	外触发输入(ET)

5.4.2 接 PS-010B / PS-011 测温专用接口：

本模块的模入部分可以直接与我公司研制的 PS-010B / PS-011B 等前端信号处理板配套使用，方便地对热电偶、热电阻的微弱信号进行放大，进行温度采集。测温专用接口 J3 接 PS-010B / PS-011B 的信号定义见表 2。

表 2 J3 接 PS-010B / PS-011 测温专用接口定义

插座引脚号	信号定义	插座引脚号	信号定义
1	CH1	2	模拟地
3	CH2	4	模拟地
5	CH3	6	模拟地
7	CH4	8	模拟地
9	CH5	10	模拟地
11	CH6	12	模拟地
13	CH7	14	模拟地
15	CH8	16	模拟地
17	模拟地	18	模拟地
19	数字地	20	数字地
21	CD0	22	CD1
23	CD2	24	CD3
25	数字地	26	数字地

CD0 - CD3 为本模块接 PS-010B/ PS-011B, 控制 PS-010B/ PS-011B 板上 16 路输入信号的 4 位通道选择信号; 如果不接 PS-010B/ PS-011B, CD0 - CD3 可用来作为一般的开关量输出, 以弥补用户应用过程中 J1 16 路开关量输出的不足。

5.4.3 开关量及定时计数部分:

本模块 40 芯扁平线插座 J1 的信号定义见表 3。

表 3 J1 开关量及脉冲输入输出端口定义

插座引脚号	信号定义	插座引脚号	信号定义
40	D016 开关量输出	39	D015 开关量输出
38	D014 开关量输出	37	D013 开关量输出
36	D012 开关量输出	35	D011 开关量输出
34	D010 开关量输出	33	D09 开关量输出
32	D08 开关量输出	31	D07 开关量输出
30	D06 开关量输出	29	D05 开关量输出
28	D04 开关量输出	27	D03 开关量输出
26	D02 开关量输出	25	D01 开关量输出
24	DI16 开关量输入	23	DI15 开关量输入
22	DI14 开关量输入	21	DI13 开关量输入
20	DI12 开关量输入	19	DI11 开关量输入
18	DI10 开关量输入	17	DI9 开关量输入
16	DI8 开关量输入	15	DI7 开关量输入
14	DI6 开关量输入	13	DI5 开关量输入
12	DI4 开关量输入	11	DI3 开关量输入
10	DI2 开关量输入	9	DI1 开关量输入
8	OUT2 脉冲输出	7	OUT1 脉冲输出
6	OUT0 脉冲输出	5	CLK2 脉冲输入
4	CLK1 脉冲输入	3	CLK0 脉冲输入
2	数字地	1	数字地

5.5 模入码制以及数据与模拟量的对应关系

5.5.1 A 型卡

本接口卡在单极性方式工作时, 即输入的模拟量为 0~10V 时, 转换后的 12 位数码为二进制原码。此 12 位数码表示一个正数码, 其数码与模拟电压值的对应关系为:

$$\text{模拟电压值} = \text{数码}(12\text{位}) \times 10(\text{V}) / 4096 \quad (\text{V})$$

即：1LSB=2.44mV

本接口卡在双极性方式工作时，转换后的12位数码为二进制偏移码。此时12位数码的最高位(DB₁₁)为符号位，“0”表示负，1”表示正。偏移码与补码仅在符号位上定义不同，此时数码与模拟电压值的对应关系为：

输入信号为-5~+5V时：

$$\text{模拟电压值} = \text{数码} \times 10(\text{V}) / 4096 - 5 \quad (\text{V})$$

即：1LSB=2.44mV

输入信号为-10~+10V时：

$$\text{模拟电压值} = \text{数码} \times 20(\text{V}) / 4096 - 10 \quad (\text{V})$$

即：1LSB=4.88mV

5.5.2 B型卡

本接口卡在单极性方式工作时，即输入的模拟量为0~10V时，转换后的16位数码为二进制原码。此16位数码表示一个正数码，其数码与模拟电压值的对应关系为：

$$\text{模拟电压值} = \text{数码}(16\text{位}) \times 10(\text{V}) / 65536 \quad (\text{V})$$

即：1LSB=0.1526mV

本接口卡在双极性方式工作时，转换后的16位数码为二进制偏移码。此时16位数码的最高位(DB₁₅)为符号位，“0”表示负，1”表示正。偏移码与补码仅在符号位上定义不同，此时数码与模拟电压值的对应关系为：

输入信号为-5~+5V时：

$$\text{模拟电压值} = \text{数码} \times 10(\text{V}) / 65536 - 5 \quad (\text{V})$$

即：1LSB=0.1526mV

输入信号为-10~+10V时：

$$\text{模拟电压值} = \text{数码} \times 20(\text{V}) / 65536 - 10 \quad (\text{V})$$

即：1LSB=0.3052mV

5.6 外触发启动信号的使用说明

本卡的A/D采样可以在外启动方式下工作。在本卡设置好外部信号启动方式，当外部启动信号为高电平，本卡即开始正常采样。此时只要检测FIFO的状态标志即可知道是否开始采样并按需要读出A/D转换结果。外部启动信号默认的输入电平是TTL电平，也可以是其他电平，但需要改变相应的限流电阻，这一点切记。**如外触发启动信号是非TTL电平，订购时请向我们技术人员咨询。**

5.7 调整与校准

本卡出厂时已进行了调整与校准，如无必要，请不要进行此项工作。如果长期使用后发现零点或满度偏移，请按下述方法进行调校。

5.7.1 零点校准

将任一通道对模拟地短接(单端方式时)同时对该通道进行A/D转换，调整W1电位器，使其转换结果为“0”或“1”(稍微偏正1个码为好)。

5.7.2 满度校准

在任一通道接入一接近正满度的稳定正电压信号，运行程序对该通道采样。调整W2使A/D转换读数等于或接近外加信号电压。

5.7.3 双极性校准

如果测量双极性信号时偏差较大，应在零点和满度已校准好的基础上分别加入正、负信号并调整W1使其符合要求。

6. 软件

6.1 软件安装：

USB7325模块在硬件安装完成后还需进行软件安装，其具体安装步骤如下：

1. 将模块USB接口插入计算机USB插座。
2. 启动计算机，操作系统将自行检测新安装的硬件，并弹出“添加新硬件向导”对话框，在“添加新硬

件向导”对话框出现时，点击“下一步”按钮。

3. 选中“显示指定位置的所有驱动程序的列表...”单选按钮，点击“下一步”按钮。

4. 点击“下一步”。

5. 点击“从软盘安装”按钮，弹出“从磁盘安装”对话框。

6. 点击“浏览”按钮，选择“Usb7kC.inf”所在的目录（光盘\USB7000\Usb7kC），点击“确定”按钮。

7. 回到“添加新硬件向导”，点击“下一步”按钮。

8. 点击“完成”按钮，第一块模块安装完成。

当要安装多块模块时，应先安装第一块模块，软件安装中会出现上述 1~8 的步骤，按以上步骤操作安装。在安装第二块模块时，系统会自动检测安装，不再经过上述 1~8 的步骤，只是模块的索引值不同。索引值是 USB 板模块使用时非常重要的参数，USB7000 系列模块是靠索引值来区分的。一般情况下，第一块安装的模块的索引值为 0，以后安装的模块索引值自动依次加 1。用户通过调用动态链接库(Usb7kC.d11)中的 OpenUSB7kC 函数，将 USB 模块的索引值赋给此函数，即可操作相应的 USB 模块（详细过程请见后面的“函数介绍”）。

6.2 测试程序说明：

本程序(光盘\USB7000\Usb7kC\7325\测试程序\7325test_vb\7325test.exe)是为 USB7325 专门编写的一个测试工具，它可以对 USB7325 采集模块的所有功能进行测试。如图 6 所示：



图 6 USB7325 测试程序

6.2.1 选择模块

运行“USB7325 数据采集模块测试程序”，点击“选择模块”菜单，在弹出的对话框中，输入所要操作模块的索引值，然后即可在程序的界面中进行选中的 USB7325 模块的测试（倘若计算机中只有一块 USB7325 模块，可略过这步，本程序默认对索引值为 0 的模块测试）。如图 7 所示：



图 7 选择模块：

本程序界面根据 USB7325 模块各功能的不同，分为以下几个部分：

6.2.2 模拟量输入测试：

该部分程序用于模拟量输入测试，如图 8 所示：



图 8 模拟量输入测试

在“模拟量输入测试”功能块, 16 文本框对应着 16 路模拟量输入通道。点击“开始采集”按钮, 16 路模拟输入电压值, 将输入到计算机, 并通过相应的文本框显示出来。

在“采集方式”框架中, 可进行模拟信号采集方式设置。“单路采集”单选钮, 用来对 16 路单端模拟量进行实时采集; “单路双端”单选钮, 用来对 8 路双端模拟量进行实时采集; “16 路循环”单选钮, 用来对 16 路模拟信号循环扫描, 所以采集速度较前者快, 且为批量采集。

在“显示设置”框架中, 可进行输入模拟量的显示设置。

USB7325A: 选中“原码值(十六进制)”单选钮, 则文本框中显示模拟量的十六进制原码值(000~FFF); 选中“0-4V”单选钮, 则显示 0 到 4000 毫伏的单极性输入信号的电压值; 选中“0-5V”单选钮, 则显示 0 到 5000 毫伏的单极性输入信号的电压值; 选中“0-10V”单选钮, 则显示 0 到 10000 毫伏的单极性输入信号的电压值; 选中“-3~+3V”单选钮, 则显示-3000 毫伏到+3000 毫伏的双极性输入信号的电压值; 选中“-5~+5V”单选钮, 则显示-5000 毫伏到+5000 毫伏的双极性输入信号的电压值; 选中“-10~+10V”单选钮, 则显示-10000 毫伏到+10000 毫伏的双极性输入信号的电压值。

USB7325B: 选中“原码值(十六进制)”单选钮, 则文本框中显示模拟量的十六进制原码值(0000~FFFF); 选中“0-4V”单选钮, 则显示 0 到 40000 X 100 微伏的单极性输入信号的电压值; 选中“0-5V”单选钮, 则显示 0 到 50000 X 100 微伏的单极性输入信号的电压值; 选中“0-10V”单选钮, 则显示 0 到 100000 X 100 微伏的单极性输入信号的电压值; 选中“-3~+3V”单选钮, 则显示-3000 X 100 微伏到+3000 X 100 微伏的双极性输入信号的电压值; 选中“-5~+5V”单选钮, 则显示-50000 X 100 微伏到+50000 X 100 微伏的双极性输入信号的电压值; 选中“-10~+10V”单选钮, 则显示-100000 X 100 微伏到+100000 X 100 微伏的双极性输入信号的电压值。

在“选择型号”框架中, 实现 USB7325A 与 USB7325B 模块的选择, 选择模块型号后, 再点击“开始采集”按钮, 进行模拟量采集。

以上的一些设置, 均可在按下“开始采集”按钮后, 进行动态的参数设置。

6.2.3 开关量测试：

该部分程序既可测试开关量输出，亦可测试开关量输入。如图 9 所示：

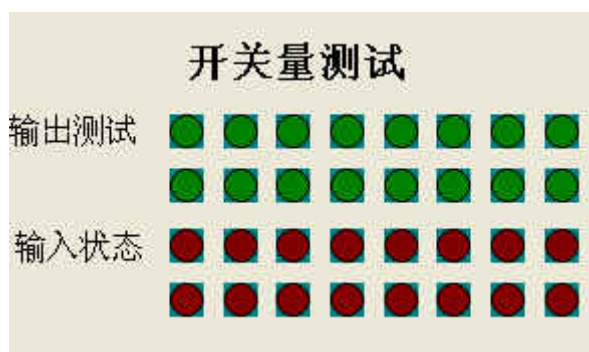


图 9 开关量测试

在测试界面的“开关量测试”功能块中，“输入状态”后面的 16 个灯，从左至右、从上到下（通道 1 至通道 16）依次指示了开关量输入通道的通断状态。该功能是自动运行的，无需用户参与。

点击“输出测试”后面的灯，可进行开关量的输出测试。同样，这 16 个灯，从左至右、从上到下依次对应着通道 1 至通道 16 开关量输出通道。点击某个灯，它的状态（亮—高电平或灭—低电平）就会切换，同时输出到对应的开关量输出通道，进行相应的输出通断状态的切换操作。

6.2.4 计数测试：

该部分程序用于 8254 计数（方式 2）的测试，如图 10 所示：



图 10 模拟量输入测试

在“计数器测试”功能块，每个计数通道都有一组“设置时钟初值”的按钮、设置初值的“设置值”文本框、显示当前值的“当前值”标签、“停止计数器”的按钮。

在文本框中输入初始值，点击对应通道的“设置时钟初值”按钮，即可启动相应的计数器，随后计数器的当前值即显示在“当前值”标签中。

点击“停止计数器”即可停止相应通道的计数器。

6.3 函数介绍：

Usb7kC.dll 是为 USB7000 系列数据采集模块配制的工作在中西文 Windows 95 / 98 / 2000/NT 环境下的一个动态链接库，它所封装的函数可以被其它应用程序在运行时直接调用。用户可以用任何一种可以使用 DLL 链接库的编程工具来编写。所列函数的说明格式为 C++ 应用程序中调用 DLL 库函数时的常用格式，无论使用哪一种开发工具，务必请注意数据格式的匹配及函数的返回类型。

为了将对模块的操作简单化。动态链接库（Usb7kC.dll）中所有的函数的参数均通过一个结构体（ZT_USBBOARD）来传递。

现将该结构体及 USB7000 的有关函数说明如下：

```
struct ZT_USBBOARD
{
    long lIndex;           /* USB模块索引值, 该值在安装时被系统分配*/
    HANDLE hHandle;       /* USB模块的操作句柄, 这个参数用户不必关心*/
    short nCh;           /* 通道号*/
    long lData;           /*输入输出数据变量, 不同函数, 不同意义*/
    unsigned long * plData; /*输入输出数据指针, 不同函数, 不同意义*/
    long lCode;           /*设备控制字,, 不同函数, 不同意义, 具体含义见函数说明*/
};
```

6.3.1 设备操作部分：

◆ 打开设备：

函数： `__declspec(dllexport) long _stdcall OpenUSB7kC(ZT_USBBOARD* bs)`

功能：打开某一中泰研创USB7325模块

入口有效参数：lIndex USB模块索引值

出口返回参数：hHandle USB模块的操作句柄（只有涉及中断时，返回才有意义）

返回值： 0 打开设备成功

-1 打开设备失败

◆ 关闭设备：

函数： `__declspec(dllexport) long _stdcall CloseUSB7kC(ZT_USBBOARD* bs)`

功能：关闭某一中泰研创USB7325模块

入口有效参数：lIndex USB模块索引值

出口返回参数：无

返回值： 0 关闭设备成功

-1 关闭设备失败

6.3.2 开关量部分：

◆ 开关量输入：

函数： `__declspec(dllexport) long _stdcall USB7325DI(ZT_USBBOARD* bs)`

功能：采集 USB7325 某一通道开关量输入信号的状态。

入口有效参数：lIndex USB 模块索引值

lCode 设备控制码： 0 单通道操作

1 多通道操作

nCh 通道号：1-16（单通道模式使用）

出口返回参数：lData 某一通道读取值：0或1（单通道操作）

该模块所有开关量输入状态值（多通道操作）

返回值： 0 采集成功

-1 采集失败

◆ 开关量输出：

函数： `__declspec(dllexport) long _stdcall USB7325DO(ZT_USBBOARD* bs)`

功能：进行某一个通道的开关量数据输出操作。

入口有效参数：lIndex USB 模块索引值

lCode 设备控制码： 0 单通道操作

1 多通道操作

nCh 通道号：1-16（单通道模式使用）

lData 某一通道输出设定值：0 或 1（单通道操作）

该模块所有开关量输入设定值（多通道操作）

出口返回参数：无

返回值: 0 开关量数据输出成功
 -1 开关量数据输出失败

◆ 开关量输出:

函数: `__declspec(dllexport) long _stdcall USB7325D0_T(ZT_USBBOARD* bs)`

功能: 进行 *CD0 - CD3* 某一个通道的开关量数据输出操作。

入口有效参数: lIndex USB 模块索引值

lCode 设备控制码: 0 单通道操作
 1 多通道操作

nCh 通道号: 1-4 (单通道模式使用)

lData 某一通道输出设定值: 0 或 1 (单通道操作)
 该模块所有开关量输入设定值 (多通道操作)

出口返回参数: 无

返回值: 0 开关量数据输出成功
 -1 开关量数据输出失败

注: “USB7325D0_T” 开关量输出函数, 仅用来控制 JP9 (见表 2) CD0 - CD3 4 路开关量输出。

6.3.3 模拟量输入部分

◆ 单通道采集

函数: `__declspec(dllexport) long _stdcall USB7325AI(ZT_USBBOARD* bs)`

功能: 对某一通道模拟量进行单步数据采集。

入口有效参数: lIndex USB 模块索引值

nCh 通道号: 1-16

lCode 设备控制字

nZTUSB.lCode = display * 256 + USB7325AB * 128 + trigger * 32 + diffence * 16

display = 返回值设置

USB7325AB = 模块型号

trigger = 外部触发

diffence = 差分输入

单双端设置 diffence

0 单端采集

1 双端采集

外部触发: trigger AD启动方式

0 程控触发 (软件启动)

1 外部触发

模块型号 USB7325AB

0 7325A

1 7325B

lData返回值设置: display 返回值形式

0 原码值

1 0 - 4 伏

2 0 - 5 伏

3 0 - 10 伏

4 -3 - +3 伏

5 -5 - +5 伏

6 -10 - +10 伏

出口返回参数: lData 某一通道模拟量的数值

返回值: 0 采集成功

-1 采集失败

注: 该函数无需调用 USB7325AIIInit 初始化

◆ 温度采集:

函数: `__declspec(dllexport) long _stdcall USB7325AITemp(ZT_USBBOARD* bs)`

功能: 对某一块 ps010/ps011 上某一通道温度进行数据采集。

入口有效参数: lIndex USB 模块索引值

nCh ps010/ps011板卡选择: 1-16

lCode 设备控制字

'*****'

```
'nZTUSB.lCode = tempch      * 4096
                + display    * 256
                + USB7325AB  * 128
                + trigger    * 32
                + diffence   * 16
```

'tempch = ps010/ps011通道选择

'display = 返回值设置

'USB7325AB = 模块型号

'trigger = 外部触发

'diffence = 差分输入

'*****'

```
单双端设置  diffence
              0      单端采集
              1      双端采集
```

```
外部触发:   trigger  AD启动方式
              0      程控触发 (软件启动)
              1      外部触发
```

```
模块型号   USB7325AB
              0      7325A
              1      7325B
```

```
lData返回值设置: display 返回值形式
                  0      原码值
                  1      0 - 4 伏
                  2      0 - 5 伏
                  4      0 - 10 伏
                  4      -3 - +3 伏
                  5      -5 - +5 伏
                  6      -10 - +10 伏
```

ps010/ps011通道选择: tempch 1-16

出口返回参数: lData 某一块ps010/ps011上某一通道温度对应的原码值或电压值

返回值: 0 采集成功

-1 采集失败

注: 该函数无需调用 USB7325AIIInit 初始化

◆ 初始化模拟量采集

函数: `__declspec(dllexport) long _stdcall USB7325AllInit(ZT_USBBOARD* bs)`

功能: 初始化模拟量数据采集。

入口有效参数: lIndex USB 模块索引值

nCh 通道号: 1-16

nCh 在指定一个通道采集时为该通道的通道号;

nCh 在多路扫描采集时为循环采集的末通道号。

lCode 设备控制字

```

',*****
' lCode = speed      * 512
',   + ch1_flag     * 256
',   + USB7325AB    * 128
',   + scan         * 64
',   + trigger      * 32
',   + diffence     * 16
',
', speed           = 程控频率
', ch1_flag       = 通道1标志
', USB7325AB      = 模块型号
', scan          = 多路扫描
', trigger        = 外部触发
', diffence       = 差分输入
',*****
单双端设置      diffence
                0          单端采集
                1          双端采集

外部触发:      trigger    AD启动方式
                0          程控触发（软件启动）
                1          外部触发

多路扫描:      scan      采集方式
                0          单通道采集
                1          多路扫描采集

模块型号      USB7325AB
                0          7325A
                1          7325B

通道1标志      ch1_flag
                0          不返回通道1标志
                1          返回通道1标志

程控采集频率: speed
                0          1KHz
                1          5KHz
                2          10KHz
                3          25KHz
                4          50KHz
                5          100KHz
                6          200KHz
                7          250KHz

```

出口返回参数：无

返回值： 0 初始化成功

-1 初始化失败

说明：1) *lCode*在不同的函数有不同的意义，且设备控制字 (*lcode*) 所包含的各种参数的设置是该参数在*lcode*中“权”的体现。拿函数USB7325AIInit来说，比如：硬件采集速度100K、返回通道1标志、USB7325B、自动（多路）扫描、外触发启动、双端采集，那么使用函数USB7325AIInit：
 $lCode = 5 * 512 + 1 * 256 + 1 * 128 + 1 * 64 + 1 * 32 + 1 * 16$

2) 通道1标志的作用：当用户设定一个固定的比较快的硬件采集速度，比如100K，那么如果PC机用用户应用程序，读取（抽取）FIFO的速度比较慢的话，就会出现FIFO满而溢出（指示灯D5亮，指示模块AD FIFO满溢出），数据出现断续的现象。这种现象在“USB7325B、多路扫描采集”是尤为糟糕：本模块连接FIFO与USB的总线为16位，USB7325B为16位AD采集，多路扫描采集时就没有多余的数据位指示从FIFO读出的AD值具体是哪个通道的AD值，如果FIFO溢出自动扫描的规律就会“乱”（FIFO溢出后再次抽取FIFO中的数据，第一个数据就不会在是第一个通道的AD值）。通道1标志的设置就是为这一情况“USB7325B、多路扫描采集”设置的。如果将通道1标志*ch1_flag*置为1，那么从FIFO读出的AD值的最低位，通道1为1，其余通道为0，以便让用户在FIFO溢出的情况下仍能找到通道1。但这样做的后果牺牲了1位精度（实际应用过程中，这点影响可以忽略不计），如果用户不想这样做可以将*ch1_flag*置为0（但用户须保证抽取FIFO的速度足够快，不使FIFO溢出）。通道1标志只在“USB7325B、多路扫描采集”这种情况下是有用的，其他情况是无效的。

◆ 设置FIFO数据地址

函数：__declspec(dllexport) long __stdcall GetFifo7325 (short nIndex, unsigned long* lAdd)

功能：获取用户存放 FIFO 数据的地址。

参数：nIndex USB 模块索引值

lAdd 存放 FIFO 数据的地址

返回值： 0 设置成功

-1 设置失败

◆ FIFO数据批量采集

函数：__declspec(dllexport) long __stdcall USB7325AIFifo(ZT_USBBOARD* bs)

功能：读取 FIFO 中的一批模拟量数据。

入口有效参数：lIndex USB 模块索引值

pldata 存放 FIFO 数据的指针

ldata 从 FIFO 中读取数据的字节数，且必须大于等于 512，即至少取 256 个数；必须小于 65536，即至最多取 32K 个数

出口返回参数：无

返回值： 0 读取成功

-1 读取失败

A/D 转换结果，即 FIFO 中的数据格式见表 4 和表 5

表 4 USB7325A (12 bit) A/D 转换结果的数据格式

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CB3	CB2	CB1	CB0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

注：CB0 - CB3 返回对应 AD0 - AD11 ad 值的通道 CH1 - CH16。

表 5 USB7325B (16 bit) A/D 转换结果的数据格式

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

注：“USB7325B、多路扫描采集”，*ch1_flag*置为0时，A/D 转换结果如上表所示；*ch1_flag*置为1时，通

道1 A/D 转换结果D0 为1, 其余通道A/D 转换结果D0 为0。

注: 1. 使用前, 需调用 USB7325AIIInit 初始化。

2. 初始化后, 用户需调用 GetFifo7325 设置存放数据的数组地址 (该函数是为不支持指针的计算机语言设置的, 比如: Visual Basic)。

3. 使用支持指针的计算机语言, 可不用 GetFifo7325 函数, 直接将存放 FIFO 数据的地址, 赋给 pData 即可。

◆ 读取FIFO状态

函数: `__declspec(dllexport) long _stdcall USB7325FIFO_Status(ZT_USBBOARD* bs)`

功能: 读取 FIFO 的状态。

入口有效参数: lIndex USB 模块索引值

出口返回参数: lData FIFO 的状态

0 空

1 有数未半满

3 半满未全满

7 全满

返回值: 0 读取成功

-1 读取失败

◆ 停止采集

函数: `__declspec(dllexport) long _stdcall USB7325ADStop(ZT_USBBOARD* bs)`

功能: 停止批量采集的硬件时钟, 不在往 FIFO 里放数。

入口有效参数: lIndex USB 模块索引值

出口返回参数: 无

返回值: 0 停止成功

-1 停止失败

◆ 清空FIFO

函数: `__declspec(dllexport) long _stdcall USB7325ClrFIFO (ZT_USBBOARD* bs)`

功能: 清空 FIFO。

入口有效参数: lIndex USB 模块索引值

出口返回参数: 无

返回值: 0 清空成功

-1 清空失败

6.3.5 计数器部分:

◆ 计数器初始化

函数: `__declspec(dllexport) long USB7325CTStart (ZT_USBBOARD* bs)`

功能: 设置计数器工作方式及某一通道初始值, 并启动计数器。

入口有效参数: lIndex USB模块索引值

nCh 通道号: 1-3

lCode 设备控制码: 0 8254以方式0工作

1 8254以方式1工作

2 8254以方式2工作

3 8254以方式3工作

4 8254以方式4工作

5 8254以方式5工作

lData 通道初始值 (0 - 65535)

出口返回参数: 无

返回值: 0 设置成功

-1 设置失败

◆ 读取计数寄存器的值

函数: `__declspec(dllexport) long USB7325CTRead(ZT_USBBOARD* bs)`

功能: 读取计数器某一通道计数器当前值。

入口有效参数: `lIndex` USB模块索引值

`nCh` 通道号: 1-3

出口返回参数: `lData` 某一通道计数器当前值 (0 - 65535)

返回值: 0 读取成功

-1 读取失败

◆ 停止计数

函数: `__declspec(dllexport) long USB7325CTStop (ZT_USBBOARD* bs)`

功能: 停止计数器某一通道计数。

入口有效参数: `lIndex` USB模块索引值

`nCh` 通道号: 1-3

出口返回参数: 无

返回值: 0 停止成功

-1 停止失败

◆ 读取频率寄存器的值

函数: `__declspec(dllexport) long _stdcall USB7325FreRead(short nIndex, unsigned long* pData)`

功能: 读取3路计数器当前频率值。

入口有效参数: `nIndex` USB模块索引值

出口返回参数: `pData` 存放3路计数器当前频率值的地址指针

返回值: 0 读取成功

-1 读取失败

注: 所有函数可根据用户的具体要求定制。

附 A. 名词注释

1. 单端输入方式

各路输入信号共用一个参考地电位, 即各路输入信号共地, 这是最常用的接线方式。使用单端输入方式时, 地线比较稳定, 抗干扰能力较强, 建议用户尽可能使用此种方式。

2. 双端输入方式

各路输入信号各自使用自己的参考电位, 即各路输入信号不共地。如果输入信号来自不同的信号源, 而这些信号源的参考电位(地线)略有差异, 可考虑使用这种接线方式。使用双端输入方式时, 输入信号易受干扰, 所以, 应加强信号线的抗干扰处理, 同时还应确保模拟地以及外接仪器机壳接地良好。而且特别注意的是, 所有接入的信号, 不论是高电位还是低电位, 其极限电平相对于本模块的模拟地电位应不超过 $\pm 15V$, 以避免电压过高造成器件损坏。

3. 单极性信号

输入信号相对于模拟地电位来讲只偏向一侧, 如输入电压为 $0 \sim 10V$ 。

4. 双极性信号

输入信号相对于模拟地电位来讲可正可负, 如输入电压为 $-5V \sim +5V$ 。

5. 码制

模拟量信号转换为数字量后, 形成一组从 0 至 4095 的连续数字, 每一个数字对应着一个特定的模拟值, 这种对应关系称为编码方法或码制。依据输入信号的不同分为单极性原码与双极性偏移码。单极性输入信号对应着单极性原码, 双极性信号对应着双极性偏移码。

6. 单极性原码

以 12 位 A/D 为例, 输入单极性信号 $0 \sim 10V$ 。转换后得到 $0 \sim 4095$ 的数字量, 数字量 0 对应模拟量为 $0V$, 数字量 4095 对应模拟量为 $10V$, 这种编码方法称为单极性原码, 其数字量值与模拟电压值的对应关系可描述为:

$$\text{模拟电压值} = \text{数码(12位)} \times 10(V) / 4096 \quad (V)$$

$$\text{即: } 1\text{LSB(1个数码位)} = 2.44\text{mV}$$

7. 双极性偏移码:

以 12 位 A/D 为例, 输入双极性信号 $-5 \sim +5V$ 。转换后得到 $0 \sim 4095$ 的数字量, 数字量 0 对应模拟量为 $-5V$,

数字量 4095 对应模拟量为+5V，这种编码方法称为双极性偏移码，其数字量值与模拟电压值的对应关系可描述为：

$$\text{模拟电压值} = \text{数码 (12 位)} \times 10(\text{V}) / 4096 - 5 \quad (\text{V})$$

$$\text{即：1LSB (1 个数码位)} = 2.44\text{mV}$$

此时 12 位数码的最高位(DB11)为符号位，此位为 0 表示负，1 表示正。偏移码与补码仅在符号位上定义不同，如果反向运算，可以先求出补码再将符号位取反就可得到偏移码。

8. A/D 转换速率：

表明 A/D 转换芯片的工作速度。如对 BB774 来讲，完成一次转换所需要的时间是 8.5 微秒，则它的转换速率为 117.65 KHz。

9. 通过率：

指 A/D 采集模块对某一路信号连续采集时的最高采集速率。

附 B. 产品清单及保修

产品清单：

1. USB7325 光电隔离型模入数据采集模块壹块。
2. USB7325 光电隔离型模入数据采集模块技术说明书壹份。
3. USB 接口电缆壹根。
4. 40 芯扁平线壹根。
5. 26 芯扁平线两根。
6. 铜柱及螺丝 4 套。
7. 安装光盘壹张

本产品自售出之日起三年内，凡用户遵守贮存、运输及使用要求，而产品质量低于技术指标的，凭保修单免费维修。因违反操作规定和要求而造成损坏的，需交纳器件和维修费。