

PCI-68XX 智能 CAN 接口卡使用说明书
(创新的高速智能 CAN 接口卡)
版本：V1.2

目 录

目 录	2
第一章 产品简介	3
1.1 产品概述	3
1.2 性能指标	3
1.3 应用领域	3
1.4 订购信息	4
1.5 产品销售清单	4
1.6 技术支持与服务	4
第二章 设备安装	5
2.1 硬件安装	5
2.2 硬件接口描述	6
2.3 终端电阻	6
2.4 驱动程序安装	7
2.4.1 在WinXP 系统下安装	7
2.4.2 在Win2000 系统下安装	10
2.5 产品使用	10
2.5.1 PCI-68XX接口卡测试软件概述	10
2.5.2 PCI-68XX接口卡测试软件详细介绍	11
第三章 用户编程	12
3.1 函数库数据结构定义	12
3.1.1 初始化CAN的数据类型	12
3.1.2 CAN信息帧的数据类型	13
3.2 接口函数说明	13
3.3 接口库函数使用方法	14
3.3.1 VC 调用动态库的方法（静态链接）	15
3.3.2 C++ Builder 调用动态库的方法（静态链接）	15
3.3.3VB 调用动态库的方法	15
3.4.函数调用流程	16
3.5.接口函数调用例程	16
3.6.验收屏蔽码设置示例	17
3.6.1 单滤波，只能收到ID=1 的标准帧设置	17
3.6.2 双滤波，只能收到ID=1 或者ID=2 的标准帧设置	18
附录	19

第一章 产品简介

1.1 产品概述

PCI-68XX 智能CAN接口卡集成1/2 路CAN通道，可以连接CAN总线并实现CAN2.0B协议（兼容2.0A）的数据通讯。兼容PCI2.2 规范，即插即用。

PCI-68XX 智能CAN接口卡的每路CAN通道都集成完全的电气隔离保护、防浪涌保护，抗干扰能力强，是一款性能稳定、通讯可靠的智能CAN接口卡。

PCI-68XX 智能CAN接口卡采用创新的硬件和软件设计思想，**实测双通道同时达到CAN的理论极限（6500帧/S）**。

PCI-68XX 智能CAN接口卡支持5Kbps ~ 1Mbps之间的波特率，配有可在Win2000/XP下工作的驱动程序，并提供**VB, VB2003, VC, C++Builder, Delphi, Labview**下的应用例程。

1.2 性能指标

PC接口：通用PCI 接口，兼容PCI2.2 规范；

帧流量：**业界最优性能，达到CAN的理论极限，实测每秒钟流量超过 6500 帧。**

传输方式：CAN2.0A 和 CAN2.0B 协议；

通道数目：支持 1-2 路 CAN 控制器，每路均可单独控制；

传输介质：屏蔽或非屏蔽双绞线；

传输速率：CAN 控制器波特率在 5Kbps~1Mbps 之间可选；

通讯接口：CAN-bus 接口采用光电隔离、DC-DC 电源隔离，隔离模块绝缘电压：2500V；

总线长度及节点数：单路总线上最多可接 110 个节点，最长通讯距离 10 公里；

占用资源：即插即用，资源自动分配；

工作温度：PCI-6810 0℃~+70℃

PCI-6820 0℃~+70℃

存储温度：-55℃~+85℃

*注：PCI-68XX接口卡具体性能指标与使用的PC硬件配置及操作系统紧密相关。

1.3 应用领域

CAN-bus 产品开发；

CAN-bus 数据分析；

CAN-bus 主从式网络；

CAN-bus 教学应用；

CAN-bus 网关、网桥；

CAN-bus 工业自动化控制系统；

智能楼宇控制、数据广播系统等 CAN-bus 应用系统；

不同 CAN-bus 网络间的数据转换；

1.4 订购信息

型号	工作温度	说明
PCI-6810	0°C~+70°C	单通道智能 PCI CAN 接口卡
PCI-6820	0°C~+70°C	双通道智能 PCI CAN 接口卡

1.5 产品销售清单

- [1] PCI-68XX 接口卡;
- [2] 光盘 1 张 (包括 PC 驱动 (Dll, Lib)、接口函数、用户手册、VB, VB2003, VC, C++Builder, Dephi, Labview 例程等);

1.6 技术支持与服务

一年免费维修、升级，终身维修。

支持邮箱: embeddedperfect@163.com

支持网站: <http://www.embedded-soc.com>

第二章 设备安装

2.1 硬件安装

PCI-68XX 智能CAN接口卡是属于静电敏感产品，出厂时安放在专用保护袋中。因此，在对接口卡进行操作时，请注意采取必要的防护措施，以保证接口卡不受损坏。

硬件安装时要在PC断电状态下，同样，拆卸PCI-68XX接口卡也应当在PC断电的状态下进行。

PCI-68XX接口卡没有任何开关和跳线用于分配中断和I/O 地址，这些都是由BIOS自动分配的。因此，在安装驱动程序之前板卡必须事先安装到PCI槽上。以下是安装步骤：

1. 关闭PC电源。
2. 打开PC的盖子。
3. 将PCI-68XX 接口卡插入空闲的PCI 插槽。
4. 拧紧固定板卡的螺钉。
5. 打开PC电源，此时BIOS会自动给PCI-68XX接口卡分配中断和I/O 地址。

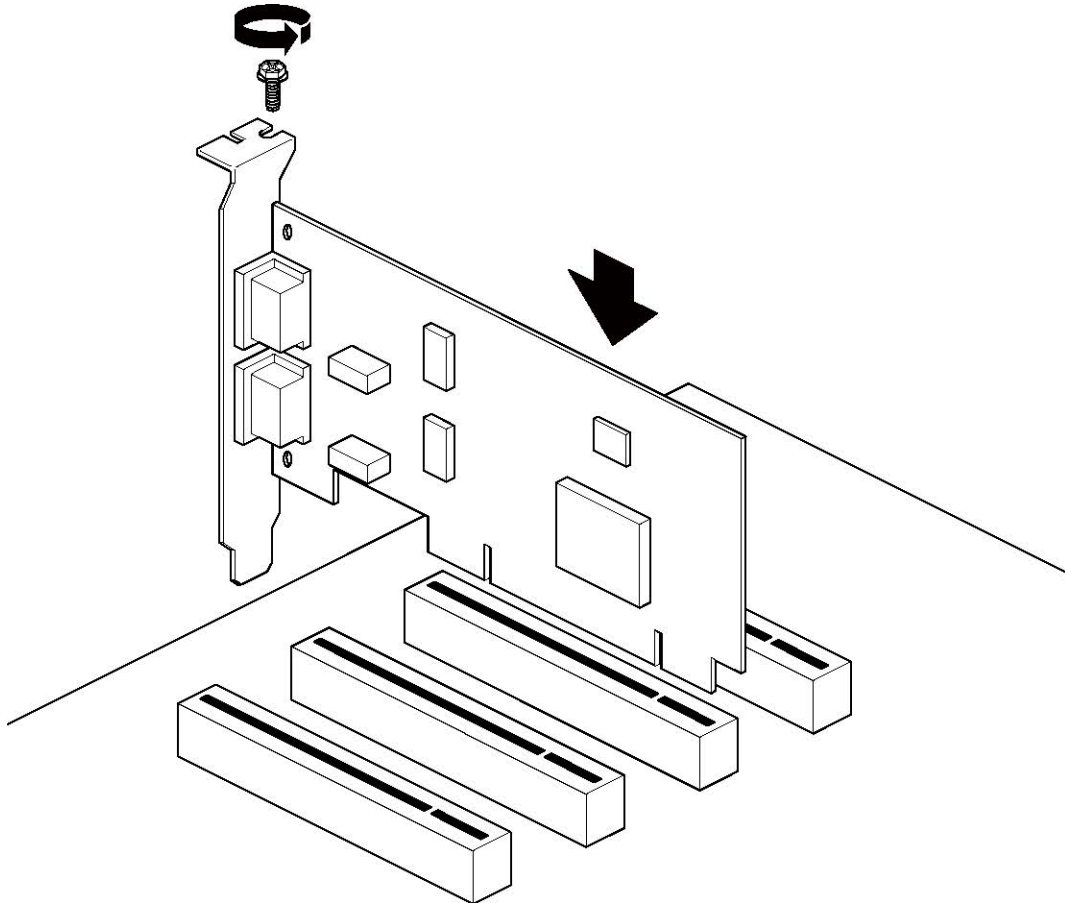


图 2-1 PCI 接口卡的安装

注意事项：请勿带电插拔PCI 接口卡；安装时不要用手触摸器件，防止静电损坏器件。

2.2 硬件接口描述

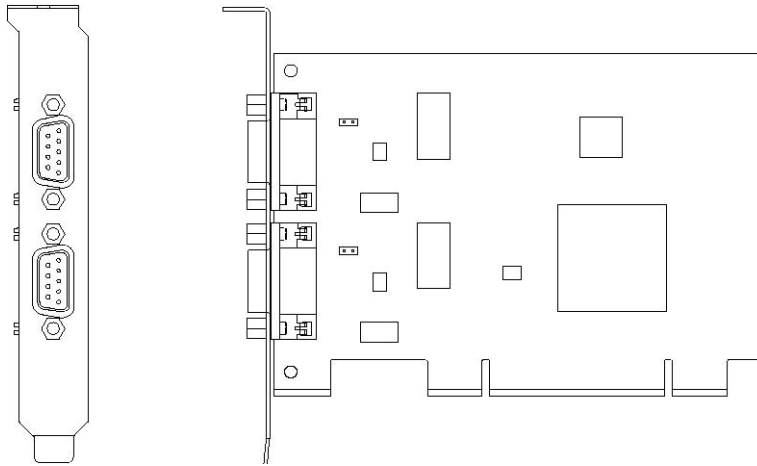


图 2-2 接口卡的DB9 插座位置

PCI-68XX 接口卡提供1/2 个CAN-bus 通道，通过DB9针型连接器与实际的CAN-bus 网络进行连接。DB9针型连接器的引脚信号定义如表所示。引脚定义符合DeviceNet 和 CANopen 标准。

表2-1 DB9针型连接器的引脚信号定义

引脚	信号	描述
1	N.C.	
2	CAN_L	CAN_L 信号线
3	CAN_GND	参考地
4	N.C.	
5	CAN_SHIELD	屏蔽线
6	CAN_GND	参考地
7	CAN_H	CAN_H 信号线
8	N.C.	
9	N.C.	

2.3 终端电阻

PCI-68XX 接口卡内建了120欧姆终端电阻，如果设备位于CAN 网络的端点，请将对应CAN 通道的跳线器跳线连上，或者在该设备端口的CAN_H 和CAN_L 之间接上一个约120 欧姆的终端电阻。PCI-68XX 接口卡采用的是PCA82C251 收发器，如果网络上其他节点使用不同的收发器，则终端电阻须另外计算。

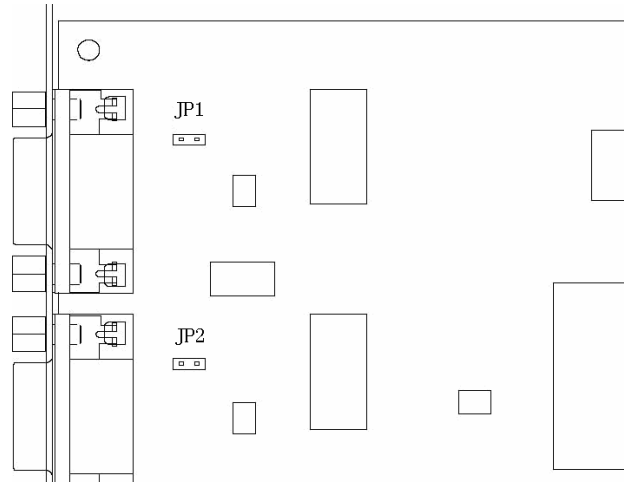


图 2-3 跳线器的位置说明

2.4 驱动程序安装

请将配套光盘<Driver>目录的文件拷贝到硬盘。为了确保任何时候安装都可以正确指定相应的驱动程序，请严格按照以下步骤进行安装处理。

2.4.1 在WinXP 系统下安装

注意：以下的图片截取自PCI-58XX接口卡的安装过程，对于PCI-68XX完全适用。

如果已将PCI-68XX接口卡插入PC 的PCI 插槽，则在重启系统之后，PC 会提示发现新硬件，如图2-4所示，此时应该选择“从列表或指定位置安装（高级）”，然后单击“下一步”：

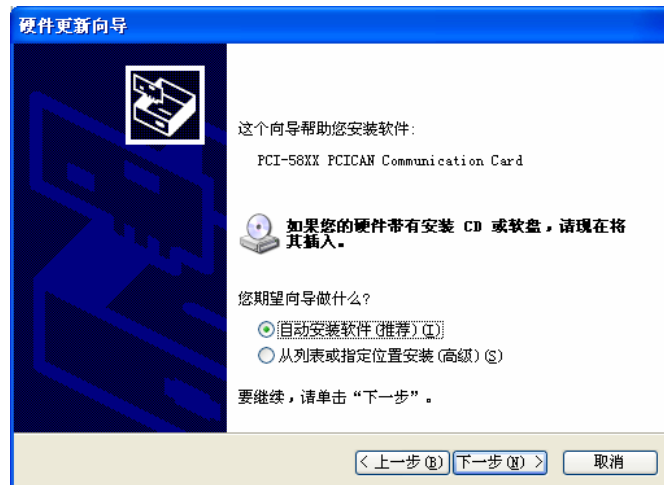


图 2-4 欢迎使用找到新硬件向导

当出现如图2-5所示的对话框，选择“不要搜索，我要自己选择要安装的驱动程序”项，然后单击“下一步”：

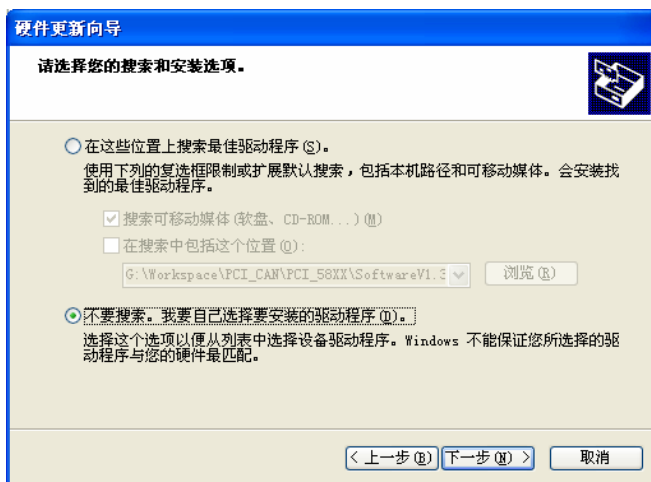


图 2-5 请选择你的搜索和安装选项

当出现如图 2-6 所示对话框时，单击“型号”栏中的空白处，暂时不选中任何型号，接着单击“从磁盘安装”按钮以指定 PCI-68XX 安装目录位置。

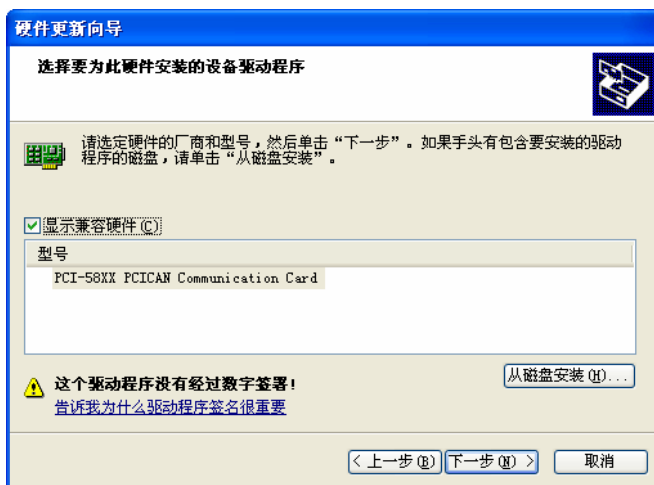


图 2-6 选择要为此硬件安装的设备驱动程序(1)

当出现如图2-7所示对话框时，我们可以通过“浏览”按钮找到驱动程序的inf 文件：

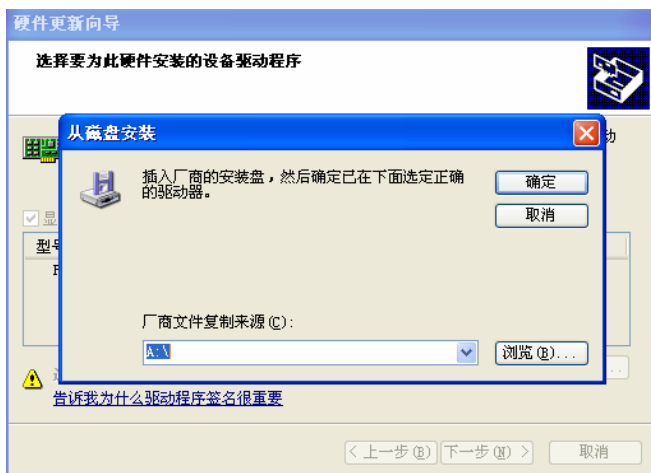


图 2-7 选择”从磁盘安装”

查找文件结果如图 2-8 所示，选中相应文件“PCI96X0.inf” 后单击“打开”：

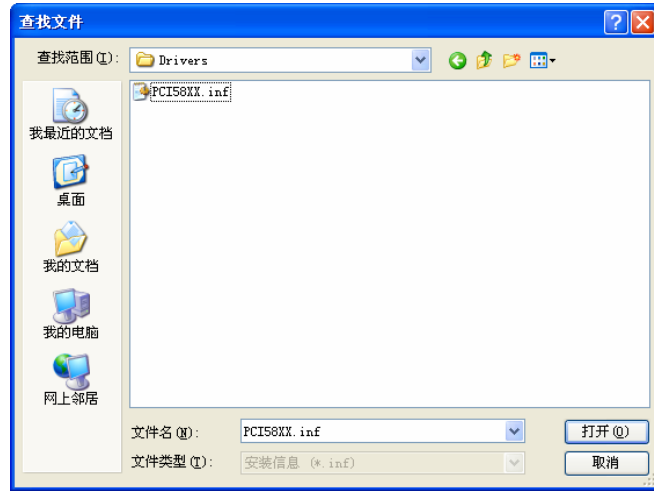


图 2-8 查找文件(1)

当出现如图2-8对话框并确定目录正确后（目录不对的话必须单击“上一步”重新查找到对为止），单击“确定”键：



图 2-9 查找文件(2)

接着如图 2-10 所示，这时我们才选中相应的板卡型号，确定选中正确型号后单击“下一步”：

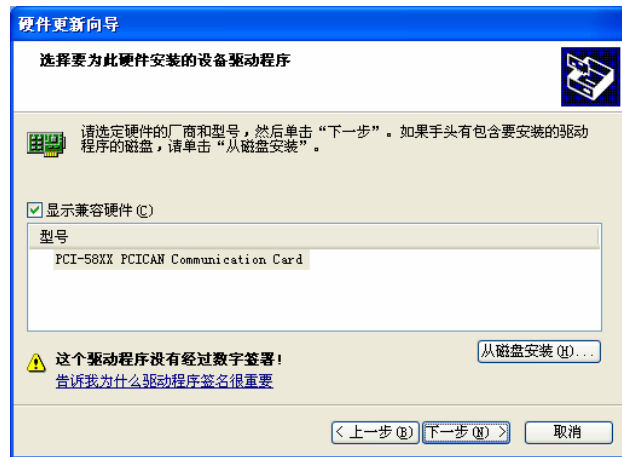


图 2-10 选择要为此硬件安装的设备驱动程序（2）

继续安装，直到出现图 2-11 所示对话框；此时，单击“完成”，即完成了驱动的安装。



图 2-11 完成找到新硬件向导

安装成功后，“设备管理器”中将列出所安装的PCI-68XX 接口卡。图 3-9 为安装完成1块PCI-68XX 接口卡后的设备管理器界面。



图 2-12Win XP 的设备管理器

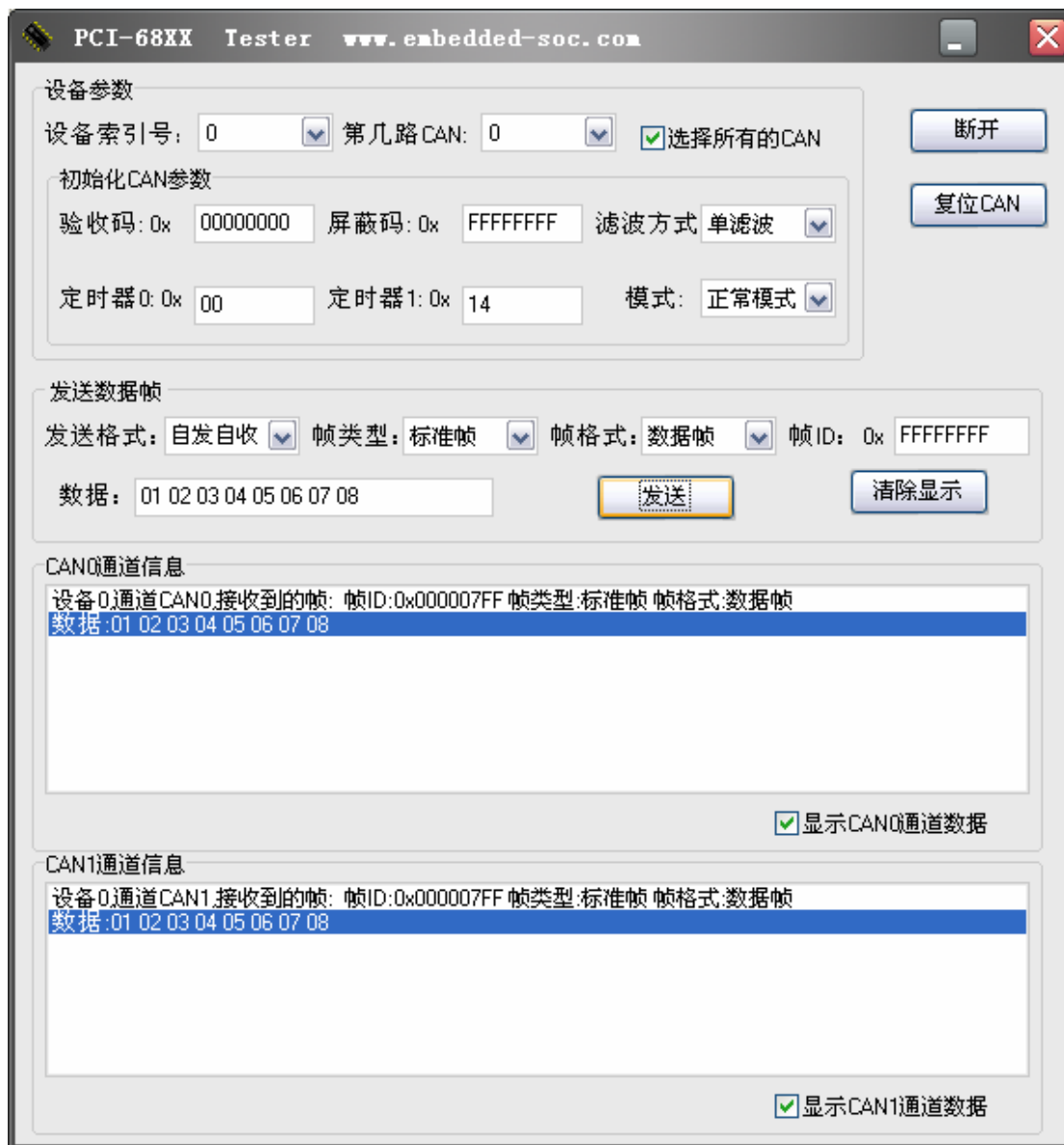
2.4.2 在Win2000 系统下安装

在Win2000 系统下安装PCI-68XX接口卡的过程与WinXP系统类似，这里不再多讲，需要注意的是：任何时候安装都要尽可能手工选择，而不要使用自动安装。

2.5 产品使用

2.5.1 PCI-68XX接口卡测试软件概述

该测试软件非常简洁明了，使用极为方便，通过打开设备命令后就可以发送和接受数据了，发送和接受的数据和状态在下面的信息提示框中有很清楚的显示。



2.5.2 PCI-68XX接口卡测试软件详细介绍

- [1] 在设备参数部分用户可以选择打开第几个设备的第几路CAN通道，当用户点击了选择所有的CAN时将同时打开二路CAN。（注意PCI6810支持CAN0通道一个通道，PCI6820才支持CAN0和CAN1二个通道）。
- [2] 在设备参数部分用户可以设置验收码，屏蔽码，滤波方式，以及波特率和工作模式
- [3] 在连接设备后，程序自动启动接受线程。
- [4] 在发送数据帧部分组合了各种可能情况由用户来进行测试。帧ID根据11位还是29位ID从低位截取。（注意从那个CAN通道发送出去由设备参数部分的用户选择的第几路CAN决定。）
- [5] 当插入多个设备时，设备索引号选择对应的设备。

第三章 用户编程

用户如果只是利用接口卡进行 CAN 总线通信测试，可以直接利用随机提供的测试软件，进行收发数据的测试。如果用户打算编写自己产品的软件程序，请仔细阅读本章节。

PCI-68XX 智能 CAN 接口卡的函数接口非常简单，主要由：[打开设备](#)，[初始化](#)，[发送数据](#)，[接受数据](#)，[关闭设备](#)以及一些辅助函数组成，函数接口类似与 ZLG CAN 接口卡。[光盘](#)中附带 VB、VC、C++Builder、Delphi、Labview 的完整例程。

3.1 函数库数据结构定义

3.1.1 初始化CAN的数据类型

```
typedef struct _ PCI68XX_INIT_CONFIG{
    DWORD    AccCode;
    DWORD    AccMask;
    DWORD    Reserved;
    UCHAR    Filter;
    UCHAR    Timing0;
    UCHAR    Timing1;
    UCHAR    Mode;
} PCI68XX_INIT_CONFIG, *P_PCI68XX_INIT_CONFIG;
```

AccCode 验收码。

AccMask 屏蔽码。

Reserved 保留。

Filter 滤波方式。

Timing0 定时器0（BTR0）。

Timing1 定时器1（BTR1）。

Mode 模式

备注 Timing0 和 Timing1 用来设置 CAN 波特率，几种常见的波特率设置如下：

CAN 波特率	定时器 0	定时器 1
5Kbps	0xBF	0xFF
10Kbps	0x31	0x1C
20Kbps	0x18	0x1C
40Kbps	0x87	0xFF
50Kbps	0x09	0x1C
80Kbps	0x83	0Xff
100Kbps	0x04	0x1C
125Kbps	0x03	0x1C
200Kbps	0x81	0xFA
250Kbps	0x01	0x1C
400Kbps	0x80	0xFA
500Kbps	0x00	0x1C

666Kbps	0x80	0xB6
800Kbps	0x00	0x16
1000Kbps	0x00	0x14

3.1.2 CAN信息帧的数据类型

```
typedef struct _PCI68XX_CAN_OBJ{
    DWORD    ID;
    UCHAR    SendType;
    UCHAR    ExternFlag;
    UCHAR    RemoteFlag;
    UCHAR    DataLen;
    UCHAR    Data[8];
} PCI68XX_CAN_OBJ, *P_PCI68XX_CAN_OBJ;
```

ID 报文ID。

SendType 0: 正常发送, 1: 自发自收

RemoteFlag 是否是远程帧。

ExternFlag 是否是扩展帧。

DataLen 数据长度(<=8), 即Data 的长度。

Data 报文的数据。

3.2 接口函数说明

[1] 打开设备

```
BOOL __stdcall PCI68XX_OpenDevice(DWORD DevIndex);
```

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

返回值 为1 表示操作成功, 0 表示操作失败。

[2] 关闭设备

```
BOOL __stdcall PCI68XX_CloseDevice(DWORD DevIndex);
```

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

返回值 为1 表示操作成功, 0 表示操作失败。

[3] 初始化CAN

```
BOOL __stdcall PCI68XX_InitCan(DWORD DevIndex, DWORD CANIndex,
P_PCI68XX_INIT_CONFIG pInitConfig);
```

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

CANIndex 第几路CAN。

InitConfig 初始化参数结构。

成员	功能描述
pInitConfig->AccCode	AccCode 对应SJA1000 中的四个寄存器ACR0, ACR1, ACR2, ACR3, 其中
pInitConfig->AccMask	高字节对应ACR0, 低字节对应ACR3; AccMask 对应SJA1000 中的四个寄存器AMR0, AMR1, AMR2, AMR3, 其中高字节对应AMR0, 低字节对应AMR3 。

pInitConfig->Reserved	保留
pInitConfig->Filter	滤波方式, 0 表示单滤波, 1 表示双滤波
pInitConfig->Timing0	定时器0
pInitConfig->Timing1	定时器1
pInitConfig->Mode	模式, 0 表示正常模式, 1 只听模式

返回值 为1 表示操作成功, 0 表示操作失败。

[4] 复位CAN设备

BOOL __stdcall PCI68XX_ResetCan(DWORD DevIndex , DWORD CANIndex);

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

CANIndex 第几路CAN。

返回值 为1 表示操作成功, 0 表示操作失败。

[5] 发送数据

BOOL __stdcall PCI68XX_Transmit(DWORD DevIndex, DWORD CANIndex, P_PCI68XX_CAN_OBJ pSend);

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

CANIndex 第几路CAN。

pSend 要发送的数据帧。

返回值 为1 表示操作成功, 0 表示操作失败。

[6] 接受数据

DWORD __stdcall PCI68XX_Receive(DWORD DevIndex, DWORD CANIndex, P_PCI68XX_CAN_OBJ pReceive, DWORD Len, INT WaitTime);

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

CANIndex 第几路CAN。

pReceive 接收的数据帧数组的首地址。

Len 需要读取多少帧数据。

返回值 返回实际读取到的帧数。

[7] 获取缓冲区中尚未读取的帧数

DWORD __stdcall PCI68XX_GetReceiveNum(DWORD DevIndex, DWORD CANIndex);

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

CANIndex 第几路CAN。

返回值 返回缓冲区中尚未读取的帧数。

[8] 清空缓冲区中的数据

BOOL __stdcall PCI68XX_ClearBuffer(DWORD DevIndex, DWORD CANIndex);

DevIndex 设备索引号, 有一个设备时索引号为0, 有两个可以为0或1。

CANIndex 第几路CAN。

返回值 为1 表示操作成功, 0 表示操作失败。

3.3 接口库函数使用方法

首先，把库函数文件都放在工作目录下。总共有三个文件PCI68XX.h, PCI68XX.lib (For VC) , PCI68XXbc.lib (For BCB)。

上述库文件支持采用VB, VC, C++Builder,Delphi, C#, Labview等工具进行编程，当用户采用动态链接时不需要用使用PCI68XX.lib (For VC) , PCI68XXbc.lib (For BCB)。

3.3.1 VC 调用动态库的方法（静态链接）

- (1) 在.CPP 中包含PCI68XX.h头文件；
- (2) 在工程文件中加入PCI68XX.lib 文件。

3.3.2 C++ Builder 调用动态库的方法（静态链接）

- (1) 在.CPP 中包含PCI68XX.h 头文件；
- (2) 在工程文件中加入PCI68XXbc.lib文件。

3.3.3VB 调用动态库的方法

通过以下方法进行声明后就可以调用了。

语法:

```
[Public | Private] Declare Function name Lib "libname" [Alias "aliasname"] [(arglist)] [As type]
```

Declare 语句的语法包含下面部分:

Public (可选)

用于声明在所有模块中的所有过程都可以使用的函数。

Private (可选)

用于声明只能在包含该声明的模块中使用的函数。

Name (必选)

任何合法的函数名。动态链接库的入口处 (entry points) 区分大小写。

Libname (必选)

包含所声明的函数动态链接库名或代码资源名。

Alias (可选)

表示将被调用的函数在动态链接库 (DLL) 中还有另外的名称。当外部函数名与某个函数重名时，就可以使用这个参数。当动态链接库的函数与同一范围内的公用变量、常数或任何其它过程的名称相同时，也可以使用 Alias。如果该动态链接库函数中的某个字符不符合动态链接库的命名约定时，也可以使用 Alias。

Aliasname (可选)

动态链接库。如果首字符不是数字符号 (#)，则 aliasname 是动态链接库中该函数入口处的名称。如果首字符是 (#)，则随后的字符必须指定该函数入口处的顺序号。

Arglist (可选)

代表调用该函数时需要传递参数的变量表。

Type (可选)

Function 返回值的数据类型；可以是 Byte、Boolean、Integer、Long、Currency、Single、Double、Decimal (目前尚不支持)、Date、String (只支持变长) 或 Variant，用户定义类型，或对象类型。

arglist 参数的语法如下:

[Optional] [ByVal | ByRef] [ParamArray] varname[()] [As type]

部分描述:

Optional (可选)

表示参数不是必需的。如果使用该选项,则 arglist 中的后续参数都必需是可选的,而且必须都使用 Optional 关键字声明。如果使用了 ParamArray,则任何参数都不能使用 Optional。

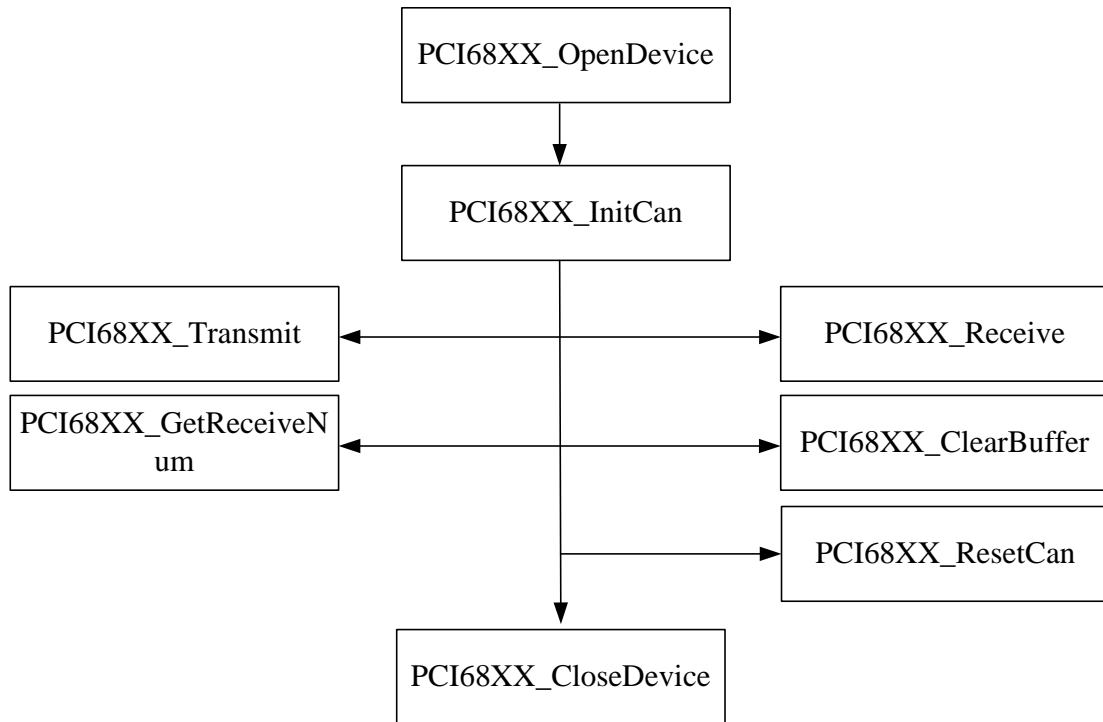
ByVal (可选)

表示该参数按值传递。

ByRef (可选)

表示该参数按地址传递。

3.4.函数调用流程



3.5.接口函数调用例程

附带光盘中的 Examples 目录下有 VB,VC,BC,Dephi,VB2003,Labview 的调用例子。

3.6.验收屏蔽码设置示例

3.6.1单滤波，只能收到ID=1的标准帧设置



3.6.2 双滤波，只能收到ID=1或者ID=2的标准帧设置



附录

1. CAN2.0B标准帧

CAN 标准帧信息为11个字节，包括两部分：信息和数据部分。前3个字节为信息部分。

	7	6	5	4	3	2	1	0	
字节1	FF	RTR	X	X	DLC (数据长度)				
字节2	(报文识别码)				ID.10-ID.3				
字节3	ID.2-ID.0			X	X	X	X	X	
字节4	数据1								
字节5	数据2								
字节6	数据3								
字节7	数据4								
字节8	数据5								
字节9	数据6								
字节10	数据7								
字节11	数据8								

字节1 为帧信息。第7 位 (FF) 表示帧格式，在标准帧中，FF=0；第6 位 (RTR) 表示帧的类型，RTR=0 表示为数据帧，RTR=1 表示为远程帧；DLC 表示在数据帧时实际的数据长度。字节2、3 为报文识别码，11 位有效。字节4~11 为数据帧的实际数据，远程帧时无效。

2. CAN2.0B扩展帧

扩展帧CAN 扩展帧信息为13 个字节，包括两部分，信息和数据部分。前5个字节为信息部分。

	7	6	5	4	3	2	1	0	
字节1	FF	RTR	X	X	DLC (数据长度)				
字节2	(报文识别码)				ID.28-ID.21				
字节3	ID.20-ID.13								
字节4	ID.12-ID.5								
字节5	ID.4-ID.0					X	X	X	
字节6	数据1								
字节7	数据2								
字节8	数据3								
字节9	数据4								
字节10	数据5								
字节11	数据6								
字节12	数据7								
字节13	数据8								

字节1 为帧信息。第7 位 (FF) 表示帧格式，在扩展帧中，FF=1；第6 位 (RTR) 表示帧的类型，RTR=0 表示为数据帧，RTR=1 表示为远程帧；DLC 表示在数据帧时实际的数据长度。字节2~5 为报文识别码，其高29 位有效。字节6~13 为数据帧的实际数据，远程帧时无效。