FT7190 USB-GPIB 转换器

用户使用手册

费思科技——世界高档仪器中的佼佼者

前言

首先，非常感谢您选择费思科技有限公司 FT7190 型 USB－GPIB 转换器。本手册包含了产品介绍、安装过程说明、库函数使用说明、随机软件使用介绍和售后服务等相关内容。为方便使用和避免因使用不当而造成事故，请您在使用产品前详细阅读本手册。除非另外声明，本手册中所出现的“转换器”均表示 FT7190 型 USB－GPIB 转换器。

产品保证

费思科技有限公司 FT7190 型 USB－GPIB 转换器的性能完全达到手册中所声称的各项技术指标。费思科技有限公司对本产品所采用的原材料和制造工艺都有严格的把关，确保转换器稳定可靠。

保修服务

费思科技有限公司担保所生产的每一台转换器在保修期内且在正常使用与维护状态下的质量。转换器若遇到保修服务范围内的损坏，并在保修期内送回授权维修机构，费思科技有限公司将予以免费维修或更换。

保证限制

本保证仅限于此转换器，不适用被错误使用、无人管理、遭受事故或处于不正常工作环境中使用的转换器。若损坏时因为错误使用、无人管理、事故或不正常工作环境导致，费思科技有限公司将在修理前提交维修估价单。

仅做以上保证，不做其他任何明示或暗示性保证，如适销性、对某种特定应用的合理性与适应性等默示性保证。在合同、民事过失和其它方面，费思科技有限公司不对任何特殊、偶然或间接损害负责。

本手册中提供的相关信息，仅供用户参考。有关保修服务的详细信息，请查阅费思产品保修条例。

包装盒内组件清单

打开 FT7190 转换器包装盒，请确认盒内包括下列组件。如有缺损，请及时联系费思授权经销商。

- USB－GPIB 转换器一个
- 用户使用说明书一本
- USB 电缆一根
- 软件光盘一张
- 保修卡一张
- 合格证一张

通告

目　录

1 产品概述

1.1 产品概述

FT7190 型 USB－GPIB 转换器为计算机提供了通过 USB 端口访问 GPIB 总线的能力，计算机通过该转换器可直接控制具有 GPIB 接口的可编程设备。一台计算机通过 USB Hub 可连接多个 FT7190 转换器，每个 FT7190 转换器能直接控制多达 14 台设备。

FT7190 转换器具有非常小的尺寸和非常轻的重量，安装方法简单，携带方便，特别适用于现场控制类的项目应用。FT7190 转换器不需要外部电源，且容易安装和使用，安装 FaithTech 提供的驱动软件后，操作系统能自动识别 FT7190 转换器，用户无需重新启动计算机，达到了真正的即插即用。

FaithTech 提供的库软件包支持计算机和设备之间进行完全透明的通讯，用户只需要调用 FaithTech 的库接口函数，就可以直接通过 FT7190 转换器操作可编程设备。

1.2 产品特点

- 可同时控制 14 台可编程设备。
- 尺寸小、重量轻，携带方便。
- 无需外部电源，无需连接设备的 GPIB 电缆。

- 具有 3 个状态指示灯，清晰表示转换器的运行状态。
- 支持 Windows 2000/XP/Me/98/SE 和 Vista 操作系统。
- 兼容 USB2.0 协议。
- 兼容 IEEE488.1 和 IEEE488.2 标准。

1.3 物理特性

- 尺寸：90*60*23(mm)
- 重量：220g
- 操作温度：0°C~55°C
- 储存温度：-40°C~70°C
- 电源：USB 总线电源，+5V，500mA(MAX)，200mA(TYPICAL)。

1.3.1 功能子集

其实现的 IEEE488 功能子集见表 1.3.1-1。

表 1.3.1-1　兼容的 IEEE488 功能子集

功能	子集	备注
Source Handshake	SH1	
Acceptor Handshake	AH1	
Talker	T6 或 TE6	可配置
Listener	L4 或 LE4	可配置
Service Request	SR1	
Remote Local	RL1	
Parallel Poll	PP1	
Device Clear	DC1	
Device Trigger	DT1	
Controller	C1	
Electrical Interface	E2	Tri-state driver
注：各功能子集的具体含义请参考 IEEE488.1&2 协议。		

2 安装方法

2.1 软件安装

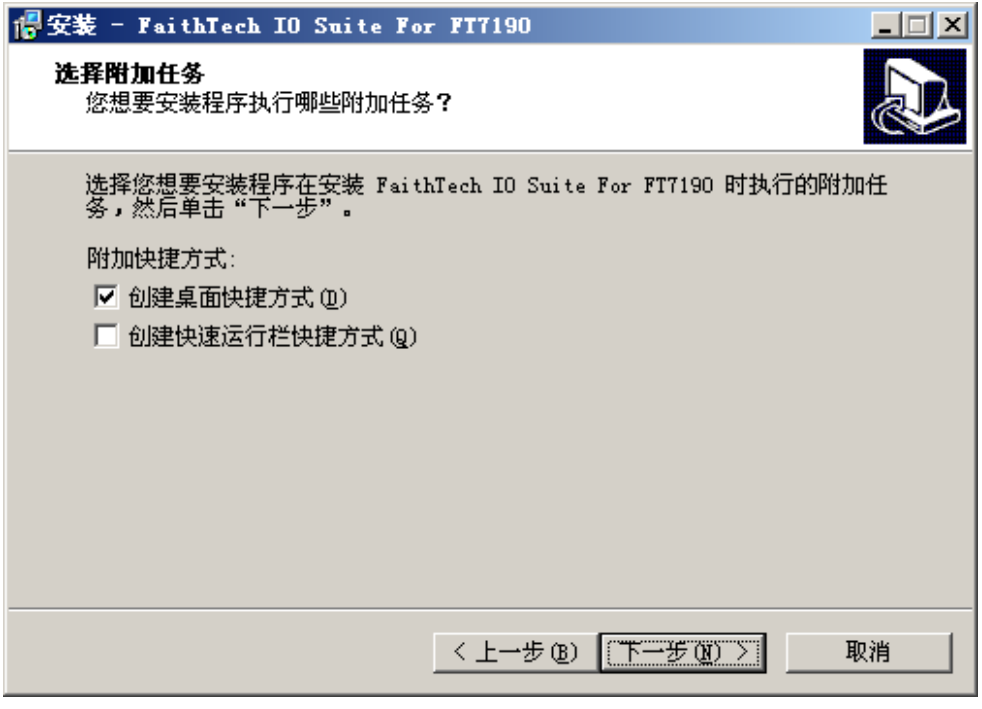
2.1.1 双击光盘中 setup.exe ，启动安装向导。

2.1.2 选择目标位置


安装 - FaithTech IO Suite For FT7190
选择目标位置
您想将 FaithTech IO Suite For FT7190 安装在什么地方？
安装程序将安装 FaithTech IO Suite For FT7190 到下列文件夹中。
单击“下一步”继续。如果您想选择其它文件夹，单击“浏览”。
C:\Program Files\FaithTech IO Suite For FT7190
浏览(R)...
至少需要有 20.0 MB 的可用磁盘空间。
< 上一步(B)
下一步(N) >
取消

2.1.3 选择开始菜单文件夹

2.1.4 选择附加任务


安装 - FaithTech IO Suite For FT7190
选择附加任务
您想要安装程序执行哪些附加任务？
选择您想要安装程序在安装 FaithTech IO Suite For FT7190 时执行的附加任务，然后单击“下一步”。
附加快捷方式:
创建桌面快捷方式(D)
创建快速运行栏快捷方式(Q)
< 上一步(B)
下一步(N) >
取消

2.1.5 准备安装软件

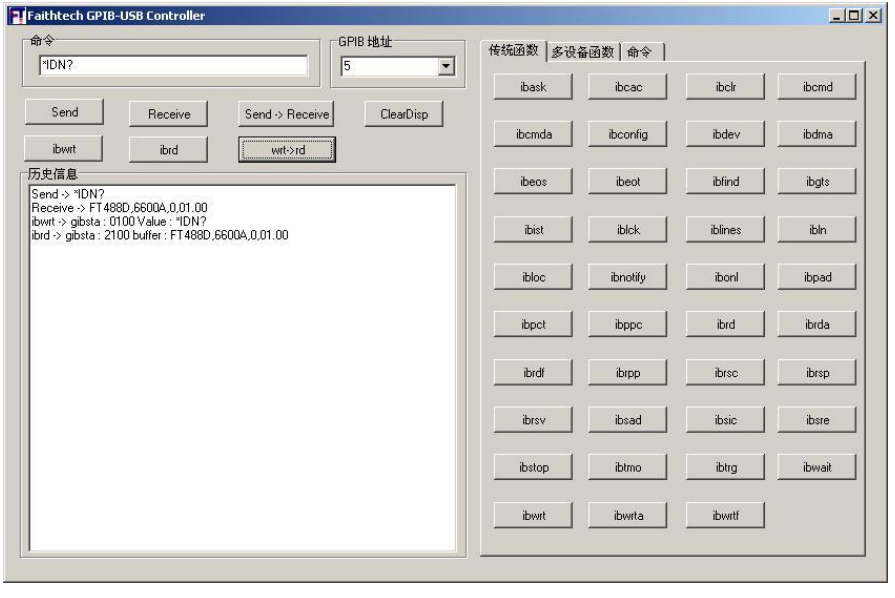

安装 - FaithTech IO Suite For FT7190
准备安装
安装程序现在准备开始安装 FaithTech IO Suite For FT7190 到您的电脑中。
单击“安装”继续此安装程序。如果您想要回顾或改变设置，请单击“上一步”。
目标位置:
C:\Program Files\FaithTech IO Suite For FT7190
开始菜单文件夹:
FaithTech IO Suite For FT7190
附加任务:
附加快捷方式:
创建桌面快捷方式(D)
< 上一步(B)
安装(I)
取消

2.1.6 安装向导完成

安装好后的程序包括演示程序和 GPIB 配置工具两个软件。

2.1.7 演示程序运行实例

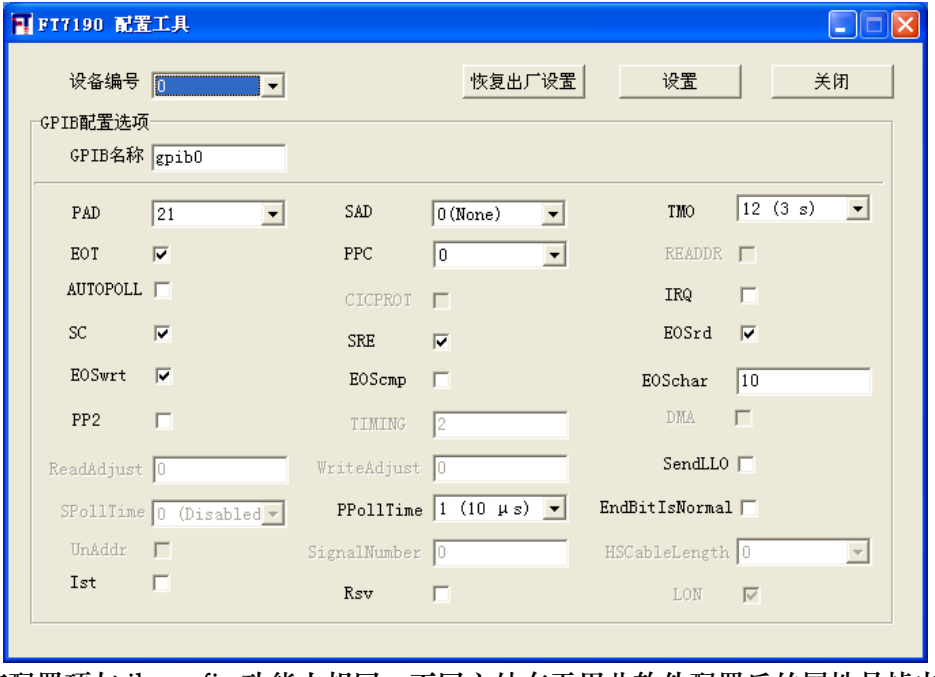
完成应用软件安装和下一节的设备安装后，运行转换器控制演示软件，将出现下图的程序界面。使用前首先选择被控制设备的 GPIB 地址，正确连接后，在文本框内输入设备控制命令，即可控制设备。所有输入的命令和返回信息都会保存在“历史信息”文本框内。



2.1.8 GPIB 配置工具用法介绍

1、打开开始菜单选择到新安装 FaithTech IO Suit 栏中的 FT7190 配置工具，如图：

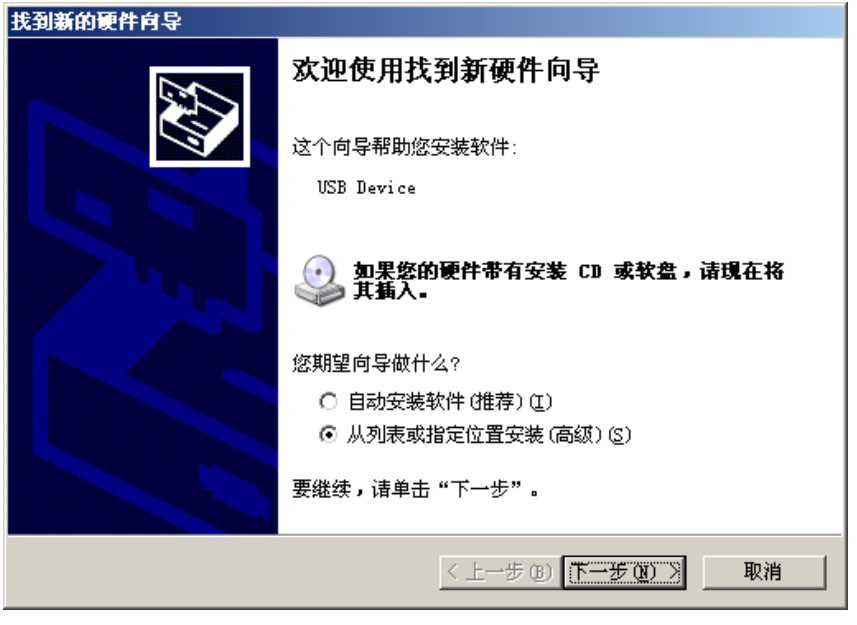
2、配置软件开启后如图所示：



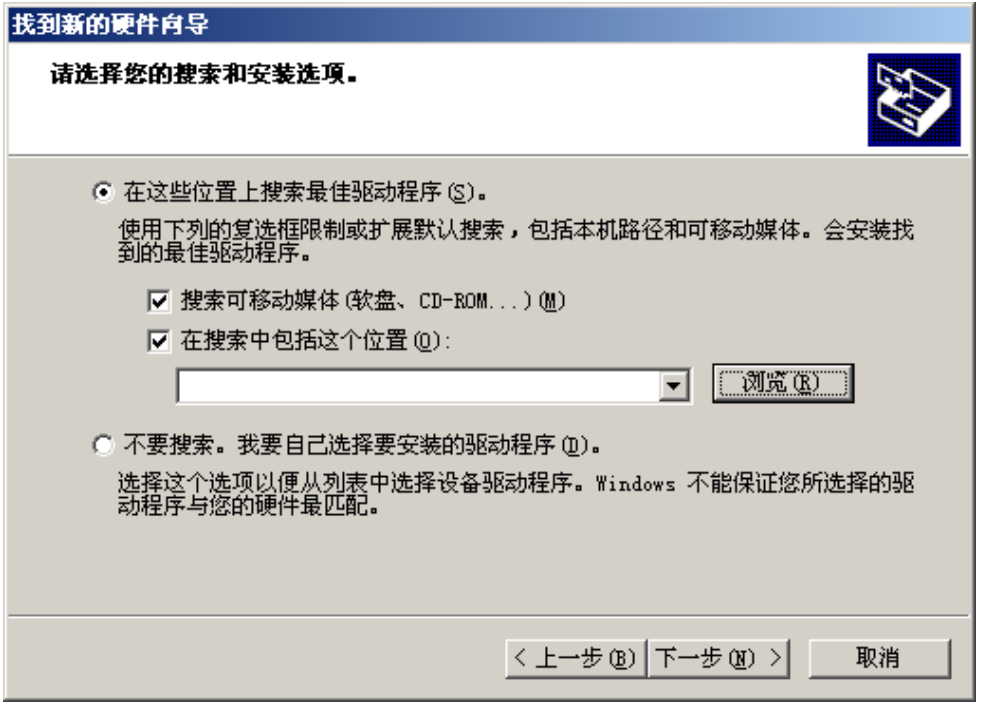
配置软件中所有配置项与 ibconfig 功能上相同，不同之处在于用此软件配置后的属性是掉电有效的。

2.2 驱动安装

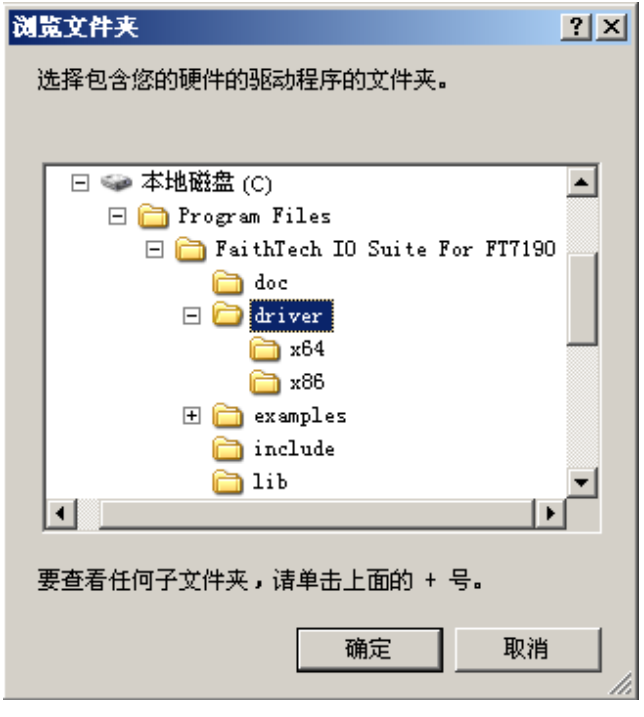
安装应用软件后，将 FT7190 转换器插入计算机的 USB 接口，选择“从列表或指定位置安装(高级)”选项，然后点击“下一步”。



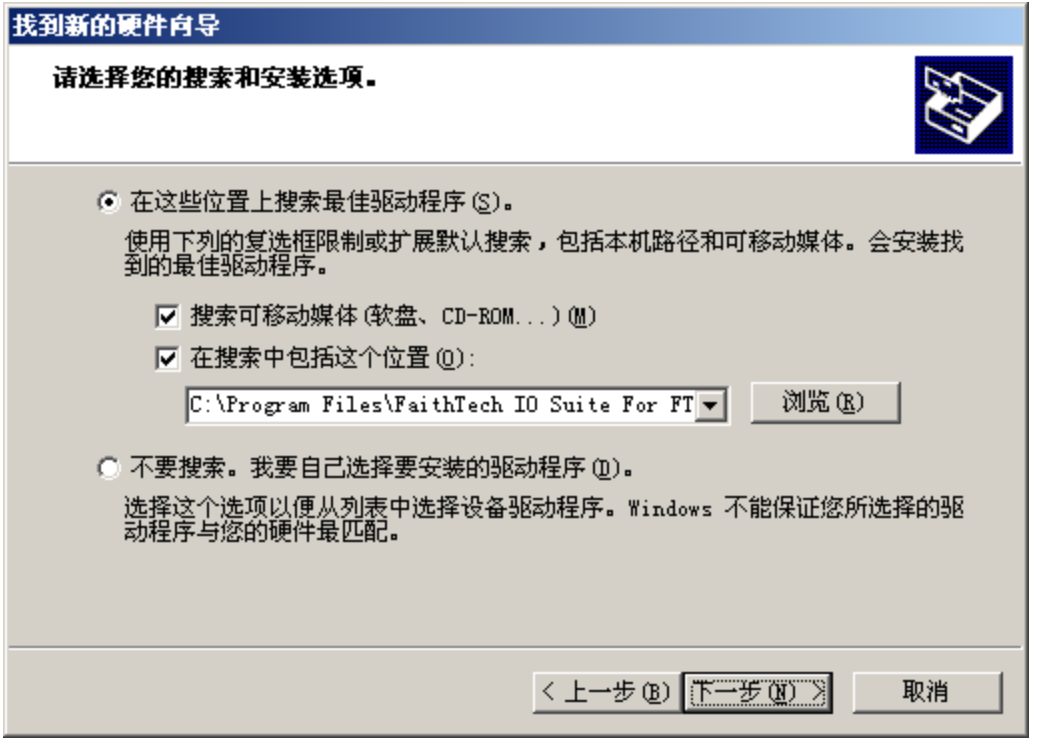
在选择您的搜索和安装选项中选中 ”在搜索中包括这个位置(O)”,点击”浏览”按钮,弹出选择硬件驱动路径对话框



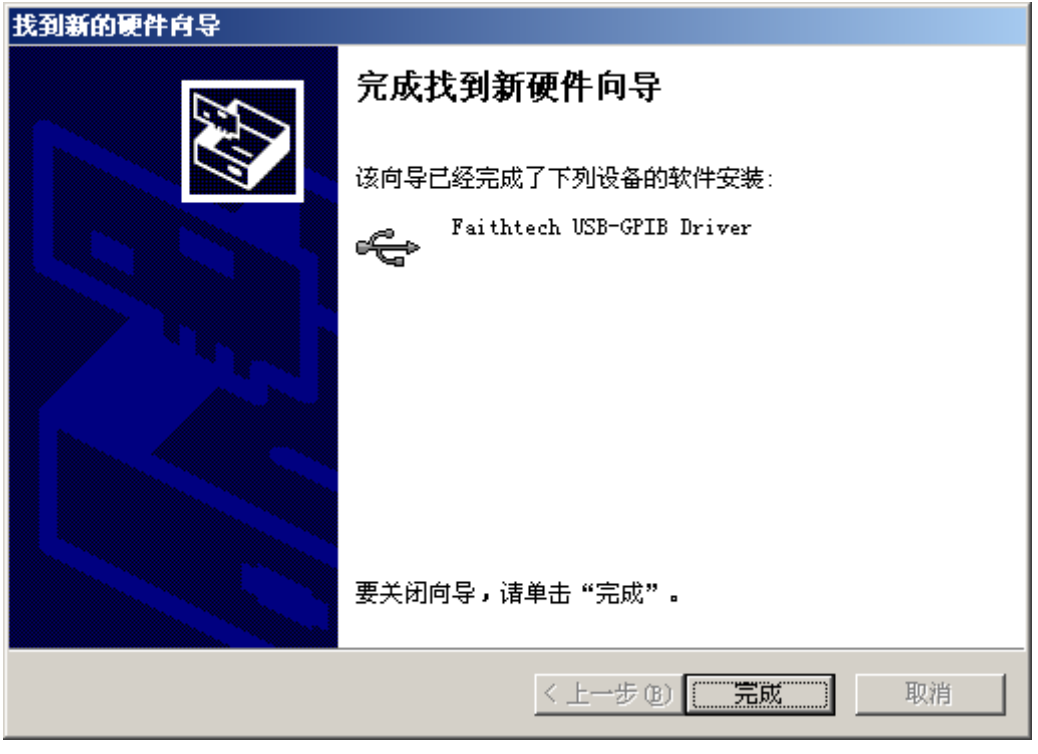
请选装安装目录下驱动文件夹.例如” C:\Program Files\FaithTech IO Suite For FT7190\driver”,点击”确定”按钮



选择好安装路径


找到新的硬件向导
请选择您的搜索和安装选项。
在这些位置上搜索最佳驱动程序(S)。
使用下列的复选框限制或扩展默认搜索，包括本机路径和可移动媒体。会安装找到的最佳驱动程序。
搜索可移动媒体(软盘、CD-ROM...)(M)
在搜索中包括这个位置(O):
C:\Program Files\FaithTech IO Suite For FT
浏览(R)
不要搜索。我要自己选择要安装的驱动程序(D)。
选择这个选项以便从列表中选择设备驱动程序。Windows 不能保证您所选择的驱动程序与您的硬件最匹配。
< 上一步(B)
下一步(N) >
取消

完成安装过程


找到新的硬件向导
完成找到新硬件向导
该向导已经完成了下列设备的软件安装:
Faithtech USB-GPIB Driver
要关闭向导，请单击“完成”。
< 上一步(B)
完成
取消

3 函数说明

表 3-1 IEEE 488 Device Level 函数

函数	说明
ibask	查询指定参数配置项的当前值
ibclr	清除指定设备
ibconfig	设置配置参数
ibdev	打开一个设备并初始化
ibeos	配置字串结束模式或结束字符值（EOS，end-of-string）
ibeot	配置 IO 写操作结束时，是否自动置位 EOI 线
ibln	检查 GPIB 总线上的某设备是否存在
ibloc	返回本地操作模式
ibonl	将设备置为在线（online）或离线（offline）
ibpad	设置主地址
ibpct	向总线上具有 Controller 功能的设备传递控制权
ibppc	配置并行查询参数
ibrd	从设备上读取数据，并将接收到的数据写入缓存区

ibrda	以异步方式从设备上读取数据，并将接收到的数据写入缓存区
ibrdf	从设备上读取数据，并将接收到的数据写入指定文件
ibrpp	执行一次并行查询操作
ibrsp	执行一次串行查询操作
ibsad	设置设备的二级地址
ibstop	取消异步操作
ibtmo	设置 IO 操作的超时时间
ibtrg	触发指定设备
ibwait	等待指定的 GPIB 总线事件发生
ibwrt	将缓冲区中的数据写入到设备
ibwrta	以异步方式将缓冲区中的数据写入到设备
ibwrtf	将指定文件中的数据写入到设备

表 3-2　IEEE 488 Board Level 函数

函数	说明
ibask	查询指定参数配置项的当前值
ibbna	查询指定接口的别名
ibcac	设置 Controller 进入 Active 状态
ibcmd	发送 GPIB 总线命令
ibcmda	以异步方式发送 GPIB 总线命令
ibconfig	设置配置参数
ibeos	配置字串结束模式或结束字符值（EOS，end-of-string）
ibeot	配置 IO 写操作结束时，是否自动置位 EOI 线
ibfind	打开并初始化 GPIB Board
ibgts	设置 Controller 从 Active 状态进入 Standby 状态
ibist	设置或清除 Board 用于并行查询响应的 ist 位
iblines	返回 GPIB 总线上控制线的状态
ibln	检查 GPIB 总线上的某设备是否存在

ibloc	返回本地操作模式
ibonl	将 Board 置为在线（online）或离线（offline）
ibpad	设置主地址
ibpct	向总线上具有 Controller 功能的设备传递控制权
ibppc	配置并行查询参数
ibrd	从设备上读取数据，并将接收到的数据写入缓存区
ibrda	以异步方式从设备上读取数据，并将接收到的数据写入缓存区
ibrdf	从设备上读取数据，并将接收到的数据写入指定文件
ibrpp	执行一次并行查询操作
ibrsc	请求或释放系统控制权
ibrsv	请求服务并更改串行查询响应字节
ibsad	设置 Board 的二级地址
ibsic	执行一次 GPIB 总线清除操作（即发送 IFC 命令）
ibsre	置位或撤销远程使能 REN 线
ibstop	取消异步操作
ibtmo	设置 IO 操作的超时时间
ibwait	等待指定的 GPIB 总线事件发生

ibwrt	将缓冲区中的数据写入到设备
ibwrta	以异步方式将缓冲区中的数据写入到设备
ibwrtf	将指定文件中的数据写入到设备

表 3-3 IEEE 488 Multi-Device 函数

函数	说明
AllSpoll	串行查询所有设备
DevClear	清除指定设备
DevClearList	清除多个指定设备
EnableLocal	使能设备本地操作，退出远程操作模式
EnableRemote	使能远程编程操作模式
FindLstn	查找 GPIB 总线设备
FindRQS	确定哪个设备正在请求服务
PassControl	向总线上具有 Controller 功能的设备传递控制权
PPoll	执行一次并行查询操作，并返回查询结果
PPollConfig	配置设备的并行查询参数
PPollUnconfig	撤销设备的并行查询参数，不再响应并行查询操作
RcvRespMsg	从已经被寻址为讲者的设备读取数据
ReadStatusByte	执行串行查询操作，获取指定设备的状态字节

Receive	从指定设备读取数据
ReceiveSetup	将指定设备寻址为讲者，并讲 Board 置位听者
ResetSys	复位并初始化所有总线设备
Send	向指定设备发送数据字节
SendCmds	向 GPIB 总线发送总线命令
SendDataBytes	向已经被寻址为听者的设备发送数据字节
SendIFC	向 GPIB 总线发送总线清除命令（IFC），复位 GPIB 总线
SendList	向多个 GPIB 总线设备发送数据
SendLLO	向 GPIB 总线上的所有设备发送本地锁定命令（LLO）
SendSetup	将多个总线设备置为听者
SetRWLS	将设备置为远程锁定状态，禁用设备的本地操作
TestSRQ	获取 GPIB 服务请求线 SRQ 的当前状态
TestSys	使设备进入自检状态并获取自检结果
Trigger	触发指定的总线设备
TriggerList	触发多个总线设备
WaitSRQ	等待总线服务请求线 SRQ 置位或超时

3.1 IEEE 488 基本函数参考

3.1.1 ibask

参数说明

ud: board 或 device 描述符

option: 返回配置项的值，有效选择表 3.1.1-1 和 3.1.1-2 中的配置项。

value:返回指定配置项的值。

返回值

ibsta

函数级别

Board / Device Level

功能描述

返回当前设定值选定的配置项。

可能的错误

EARG, ECAP, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibask (int ud, int option, int *value )
```

Visual Basic

call ibask (ByVal ud As Integer, ByVal opt As Integer, rval As Integer)

表 3.1.1-1 ibask Board 配置参数选择

选择（常数）	选择(值)	返回信息
IbaPAD	0x0001	当前 board 主地址。
IbaSAD	0x0002	当前 board 二级地址。
IbaTMO	0x0003	当前 board I/O 超时时间。
IbaEOT	0x0004	0：在写操作结束时 EOI 线不置位。 1：在写操作结束时 EOI 线置位。
IbaPPC	0x0005	Board 当前的并行查询配置信息。
IbaAUTOPOLL	0x0007	0：禁止自动串行查询。 1：开启自动串行查询。
IbaIRQ	0x0009	0：禁止中断。 1：开启中断。

IbaSC	0x000A	0：board 不是 GPIB 系统控制器。 1：board 是系统控制器。
IbaSRE	0x000B	0：Board 作为系统控制器时不自动置位 REN 线。 1：Board 作为系统控制器时自动置位 REN 线。
IbaEOSrd	0x000C	0：在读取操作时忽略收到的 EOS 字符。 1：收到 EOS 字符则结束读取操作。
IbaEOSwrt	0x000D	0：写操作发送 EOS 字符的同时不置位 EOI 线。 1：写操作发送 EOS 字符的同时置位 EOI 线。
IbaEOScmp	0x000E	0：将接收到的字符与 EOS 字符的低 7 位相比较。 1：将接收到的字符与 EOS 字符的全部 8 位相比较。
IbaEOSchar	0x000F	board 当前 EOS 字符。
IbaPP2	0x0010	0：board 处于并行查询配置远程模式(PP1)。 1：board 处于并行查询配置本地模式(PP2)。
IbaTIMING	0x0011	当前 board 的总线时序。 1：正常时序（T1 延时 2uS)。 2：高速时序（T1 延时 500nS)。 3：超高速时序（T1 延时 350nS)。

IbaDMA	0x0012	0：board 对 GPIB 传输将不使用直接内存访问(DMA)。 1：board 对 GPIB 传输将使用 DMA。
IbaSendLLO	0x0017	0：使用 ibfind 或 ibdev 打开设备时不发送 LLO 命令。 1：使用 ibfind 或 ibdev 打开设备时发送 LLO 命令。
IbaPPollTime	0x0019	0：当进行并行查询时 board 使用标准持续时间（2uS)。 1-17：当进行并行查询时 board 使用可变长度持续时间。持续时间值与 ibtmo 时间值相一致。
IbaEndBitIsNormal	0x001A	0：当接收到 EOI 或 EOI 加上 EOS 字符时，ibsta 的 END 置位。如果只接收到 EOS 字符而没有 EOI 信号，则不置位 END。 1：当已接收 EOI、EOS 或 EOI 加上 EOS 时 END 均置位。
IbaIst	0x0020	Board 的特殊状态（ist）位。
IbaRsv	0x0021	Board 的当前串行查询状态字。

表 3.1.1-2　ibask Device 配置参数选择

选择（常数）	选择（值）	返回信息
IbaPAD	0x0001	当前 device 主地址。
IbaSAD	0x0002	当前 device 二级地址。
IbaTMO	0x0003	当前 device I/O 超时。
IbaEOT	0x0004	0：写入操作结束时 GPIB EOI 线不置位。 1：写入操作结束时 EOI 置位。
IbaREADDR	0x0006	0：在设备级读取和写入操作之间无需执行非必要的寻址操作。 1：在设备级读取或写入操作之前总是执行寻址操作。
IbaEOSrd	0x000C	0：在读取操作时忽略接收到的 EOS 字符。 1：收到 EOS 字符时结束读取操作。
IbaEOSwrt	0x000D	0：当写入操作发送 EOS 字符时，EOI 线不置位。 1：当写入操作发送 EOS 字符时，EOI 线置位。
IbaEOScmp	0x000E	0：将接收到的字符与 EOS 字符的低 7 位相比较。 1：将接收到的字符与 EOS 字符的全部 8 位相比较。

IbaEOSchar	0x000F	当前 board EOS 字符。
IbaSPollTime	0x0018	当正在查询 device 时驱动长时等待串行查询。通过 ibtmo 时间值响应很长时间。
IbaUnAddr	0x001B	0：在每次设备级读写操作之后，不发送 Untalk(UNT) 和 Unlisten(UNL) GPIB 命令。 1：在每次设备级读写操作之后，发送 Untalk(UNT) 和 Unlisten(UNL) GPIB 命令。

3.1.2 ibclr

参数说明

ud: device 描述符

返回值

ibsta

函数级别

Device Level

功能描述

发送所选 Device 清除（SDC）信息给指定 Device。

可能的错误

EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
int ibclr( int ud )
```

Visual Basic

```
call ibclr(ByVal ud As Integer)
```

3.1.3 ibconfig

参数说明

ud: board 或 device 描述符
opt: 想要改变的配置项。有效选择下表 3.1.3-1 和 3.1.3-2 中列出的项目。
value: 想要改变配置项的值。

表 3.1.3-1 ibconfig Board 配置参数选择

选择（常数）	选择（值）	返回信息
IbcPAD	0x0001	当前 board 主地址。
IbcSAD	0x0002	当前 board 二级地址。
IbcTMO	0x0003	当前 board I/O 超时时间。
IbcEOT	0x0004	0：在写操作结束时 EOI 线不置位。 1：在写操作结束时 EOI 线置位。
IbcPPC	0x0005	Board 当前的并行查询配置信息。

IbcAUTOPOLL	0x0007	0：禁止自动串行查询。 1：开启自动串行查询。
IbcIRQ	0x0009	0：禁止中断。 1：开启中断。
IbcSC	0x000A	0：board 不是 GPIB 系统控制器。 1：board 是系统控制器。
IbcSRE	0x000B	0：Board 作为系统控制器时不自动置位 REN 线。 1：Board 作为系统控制器时自动置位 REN 线。
IbcEOSrd	0x000C	0：在读取操作时忽略收到的 EOS 字符。 1：收到 EOS 字符则结束读取操作。
IbcEOSwrt	0x000D	0：写操作发送 EOS 字符的同时不置位 EOI 线。 1：写操作发送 EOS 字符的同时置位 EOI 线。
IbcEOScmp	0x000E	0：将接收到的字符与 EOS 字符的低 7 位相比较。 1：将接收到的字符与 EOS 字符的全部 8 位相比较。
IbcEOSchar	0x000F	board 当前 EOS 字符。
IbcPP2	0x0010	0：board 处于并行查询配置远程模式(PP1)。 1：board 处于并行查询配置本地模式(PP2)。

IbcTIMING	0x0011	当前 board 的总线时序。 1：正常时序（T1 延时 2uS)。 2：高速时序（T1 延时 500nS)。 3：超高速时序（T1 延时 350nS)。
IbcDMA	0x0012	0：board 对 GPIB 传输将不使用直接内存访问(DMA)。 1：board 对 GPIB 传输将使用 DMA。
IbcSendLLO	0x0017	0：使用 ibfind 或 ibdev 打开设备时不发送 LLO 命令。 1：使用 ibfind 或 ibdev 打开设备时发送 LLO 命令。
IbcPPollTime	0x0019	0：当进行并行查询时 board 使用标准持续时间（2uS)。 1-17：当进行并行查询时 board 使用可变长度持续时间。持续时间值与 ibtmo 时间值相一致。
IbcEndBitIsNormal	0x001A	0：当接收到 EOI 或 EOI 加上 EOS 字符时，ibsta 的 END 置位。如果只接收到 EOS 字符而没有 EOI 信号，则不置位 END。 1：当已接收 EOI、EOS 或 EOI 加上 EOS 时 END 均置位。
IbcIst	0x0020	Board 的特殊状态（ist）位。
IbcRsv	0x0021	Board 的当前串行查询状态字。

表 3.1.3-2 ibconfig Device 配置参数选择

选择（常数）	选择（值）	规定值
IbcPAD	0x0001	当前 device 主地址。
IbcSAD	0x0002	当前 device 二级地址。
IbcTMO	0x0003	当前 device I/O 超时。
IbcEOT	0x0004	0：写入操作结束时 GPIB EOI 线不置位。 1：写入操作结束时 EOI 置位。
IbcREADDR	0x0006	0：在设备级读取和写入操作之间无需执行非必要的寻址操作。 1：在设备级读取或写入操作之前总是执行寻址操作。
IbcEOSrd	0x000C	0：在读取操作时忽略接收到的 EOS 字符。 1：收到 EOS 字符时结束读取操作。
IbcEOSwrt	0x000D	0：当写入操作发送 EOS 字符时，EOI 线不置位。 1：当写入操作发送 EOS 字符时，EOI 线置位。

IbcEOScmp	0x000E	0：将接收到的字符与 EOS 字符的低 7 位相比较。 1：将接收到的字符与 EOS 字符的全部 8 位相比较。
IbcEOSchar	0x000F	当前 board EOS 字符。
IbcSPollTime	0x0018	当正在查询 device 时驱动长时等待串行查询。通过 ibtmo 时间值响应很长时间。
IbcUnAddr	0x001B	0：在每次设备级读写操作之后，不发送 Untalk(UNT) 和 Unlisten(UNL) GPIB 命令。 1：在每次设备级读写操作之后，发送 Untalk(UNT) 和 Unlisten(UNL) GPIB 命令。

返回值

ibsta

函数级别

Board / Device level

功能描述

设置设定值选定的配置项。

可能的错误

EARG, ECAP, EDVR, EHDK, ELCK, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibconfig(int ud, int option, int value)

Visual Basic

call ibconfig(ByVal ud As Integer, ByVal opt As Integer, ByVal v As Integer)

3.1.4 ibdev

参数说明

board_index： 控制该设备的 Board 的索引号。

pad: GPIB 设备主要地址。

sad: GPIB 设备二级地址。

tmo: I/O 超时值。

eot: 启用或禁用设备的 EOI 模式。

eos: 配置的 EOS 字符和 EOS 模式。

返回值

Device 描述符或-1

函数级别

Device level

功能描述

打开并初始化一个设备描述符，如果 ibdev 无法获得一个有效的设备描述符，则返回-1；在错误代码中设置包含 EDVR 的 ibsta 和 iberr。

可能的错误

EARG, EBUS, ECIC, EDVR, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibdev( int board_index, int pad, int sad, int timo, int send_eoi, int eosmode )
```

Visual Basic

```
call ibdev(ByVal bdid As Integer, ByVal pad As Integer, ByVal sad As Integer,
   ByVal tmo As Integer, ByVal eot As Integer, ByVal eos As Integer, ud As Integer)
```

3.1.5 ibeos

参数说明

ud: board 或 device 描述符。

v: EOS 模式和字符信息。如果 v 是 0，则 EOS 字符是禁用的；否则，低字节是 EOS 字符，高字节包含定义 EOS 模式的标志信息。不同的 EOS 配置和对应的 v 值如下表所示。

EOS 模式	V 值		
	位	高字节	低字节
EOS 被检测时终止读取	A	00000100	EOS 字符
设置 EOS 具有 EOS 写功能	B	00001000	EOS 字符
比较所有的 8 位 EOS 字节而不是低 7 位（读和写功能）	C	00010000	EOS 字符

终止读取 I/O 操作时配置位 A 和位 C 确定。如果位 A 是常数和位 C 是清除掉的，接收到的字符与 EOS 低 7 位相匹配时读取结束。如果位 A 和位 C 都是常数，接收到的字符与 EOS 字符的所有 8 位相匹配时读取结束。

写 I/O 操作维护 GPIB EOI 线时配置位 B 和位 C 确定。如果位 B 是固定不变的和位 C 是清除掉的，当书面字符与低 7 位 EOS 字符相匹配时 EOI 被终止。如果位 B 和位 C 都是固定不变的，当书面字符与 EOS 字符的所有 8 位相匹配时 EOI 被终止。

返回值

ibsta

函数级别

Board / Device level

功能描述

配置结束字符串（EOS）终止模式或字符。

注意：给定的 EOS 不会在写 I/O 操作结束时被自动发送。您的配置主要是负责使 EOS 字节结束被指定的数据字符串。

可能的错误

EARG, EDVR, EHGL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibeot (int ud, int v)
```

Visual Basic

```
call ibeos (ByVal ud As Integer, ByVal v As Integer)
```

3.1.6 ibeot

参数说明

ud: board 和 device 描述符

v: 开启或禁用 EOT 模式

返回值

ibsta

函数级别

Board / Device level

功能描述

开启或禁用在写 I/O 操作结束时自动置位 EOI 线。若开启 EOT 模式，当发送完最后的字节时 EOI 置上。

可能的错误

EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibeot ( int ud, int v )
```

Visual Basic

```
call ibeot (ByVal ud As Integer, ByVal v As Integer)
```

3.1.7 ibln

参数说明

ud: board 和 device 描述符。如果 ud 是 board 描述符，总线与 board 进行测试听者。如果 ud 是 device 描述符，ibln 使用 board 与 Device 测试听者。如果检测到听者，返回值非零；如果没有找到听者，则返回值为零。

pad: 主 device 地址（值在 0-30 之间）。

sad: 二级 device 地址（值在 96-126 之间或 NO-SAD 或 ALL-SAD,如果 NO-SAD 不是二级地址在测试，仅仅是 ALL-SAD 主地址在测试，表示所有的二级地址都被测试。）

found_listener: 指明被检测设备是否存在。

返回值

ibsta

函数级别

Board / Device level

功能描述

检查在总线上是否存在某 Device。

可能的错误

EARG, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibln(int ud, int pad, int sad, short *found_listener)

Visual Basic

call ibln (ByVal ud As Integer, ByVal pad As Integer, ByVal sad As Integer,
found_listener As Integer)

3.1.8 ibloc

参数说明

ud: board 或 device 描述符

返回值

ibsta

函数级别

Board / Device level

功能描述

对于接口，如果不处于锁定状态，则 ibloc 使接口处于本地模式。如果接口处于锁定状态，则调用不生效。
对于设备,除非执行 ibsre 撤销 REN 线,否则所有的 device-level 调用自动使指定的 device 处于远程模式。

ibloc 使 device 从远程模式转变为本地模式，并一直维持本地模式至下一个 device 函数被调用。

可能的错误

EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibloc (int ud)

Visual Basic

call ibloc (ByVal ud As Integer)

3.1.9 ibonl

参数说明

ud: board 或 device 描述符

onl: 在线（1）或离线（0）

返回值

ibsta

函数级别

Board / Device level

功能描述

重置 board 或 device,设置所有的软件配置参数使 device 在线或离线。如果 device 或 Board 离线，device 或 Board 描述符不再有效，你必须再次请求 ibdev 或 ibfind 开启 device 或 Board。

可能的错误

EARG, EHDL, ELCK, ENEB

函数语法

Microsoft C/C++ and Borland C++

int ibonl (int ud, int onl)

Visual Basic

call ibonl (ByVal ud As Integer, ByVal onl As Integer)

3.1.10 ibpad

参数说明

ud: board 或 device 描述符

v: GPIB 主地址。有效值范围 0-30。

返回值

Ibsta

函数级别

Board / Device level

功能描述

在 board 或 device 中设置主地址。

可能的错误

EARG, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibpad (int ud, int v)

Visual Basic

call ibpad(ByVal ud As Integer, ByVal v As Integer)

3.1.11 ibpct

参数说明

ud: device 描述符

返回值

Ibsta

函数级别

Device level

功能描述

传递控制权给另一个具有控制功能的 GPIB device。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibpct ( int ud )
```

Visual Basic

```
call ibpct (ByVal ud As Integer)
```

3.1.12 ibppc

参数说明

ud: 描述符

v: 并行查询配置

返回值

ibsta

函数级别

Board / Device level

功能描述

配置并行查询。

若 ud 是设备描述符，ibppc 使能或禁止设备对并行查询作出响应。设备被寻址并接收到并行查询信息：并行查询开启 (PPE) 或禁止 (PPD)。有效并行查询信息范围是 96-126 (十六进制 60-7E) 或者是 0 (PPD)。

若 ud 是接口描述符，ibppc 根据参数 v 执行本地并行查询配置。有效并行查询信息范围是 96-126（十六进制 60-7E）或是 0（PPD)。如果在调用时没有错误发生，iberr 则包含了先前的本地并行查询配置值。

可能的错误

EARG, EBUS, ECAP, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibppc ( int ud, int v )
```

Visual Basic

```
call ibppc (ByVal ud As Integer, ByVal v As Integer)
```

3.1.13 ibrd

参数说明

ud: 描述符

buf:存储从 GPIB 读取到的信息的缓存区

cnt: 从 GPIB 读取到的字节数

返回值

ibsta

函数级别

Board / Device level

功能描述

从设备读取数据到用户指定缓存区。

如果 ud 是 device 描述符，ibrd 从 GPIB 总线上进行寻址，计划读取字节数为 cnt 的数据，并将所读数据存入 buf。当 cnt 指定字节数已经全部接收或 END 标志被接收时，操作正常结束。如果传输没有在超时时间内完成，则操作结束并伴随一个错误。在全局变量 ibcntl 中返回实际接收到的字节数。

如果 ud 是 board 描述符，ibrd 计划读取 cnt 字节的数据，并将数据存入缓存区（buf)。当 cnt 指定字节数已经接收或 END 标志被接收时，操作正常结束。下列情况下操作结束并伴随一个错误:

1、传输没有在超时时间内完成;

2、board 不是 CIC。若 board 不是 CIC，则 CIC 在 GPIB 总线上发送一个设备清除命令。

在 ibcntl 全局变量中返回实际接收数据的字节数。

可能的错误

EABO, EADR, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibrd (int ud, void *buf, long cnt)

Visual Basic

call ibrd (ByVal ud As Integer, buf As String)

3.1.14 ibrda

参数说明

ud: 描述符

buf:数据存储缓存区

cnt:读取的字节数

返回值

Ibsta

函数级别

Board / Device level

功能描述

异步地读取数据并存入用户指定缓存区。

如果 ud 是 device 描述符，ibrd 从 GPIB 总线上进行寻址，计划读取字节数为 cnt 的数据，并将所读数据存入 buf。当 cnt 指定字节数已经全部接收或 END 标志被接收时，操作正常结束。如果传输没有在超时时间内完成，则操作结束并伴随一个错误。在全局变量 ibcntl 中返回实际接收到的字节数。

如果 ud 是 board 描述符，ibrd 计划读取 cnt 字节的数据，并将数据存入缓存区（buf)。当 cnt 指定字节数已经接收或 END 标志被接收时，操作正常结束。下列情况下操作结束并伴随一个错误:

1、传输没有在超时时间内完成;

2、board 不是 CIC。若 board 不是 CIC，则 CIC 在 GPIB 总线上发送一个设备清除命令。

在 ibcntl 全局变量中返回实际接收数据的字节数。

当 I/O 正在进行时，异步 I/O 调用(ibcmda, ibrda, ibwrta)应用程序能执行其它的非 GPIB 操作。一旦异步 I/O 调用开始，其它 GPIB 调用会受到严格限制。任何可能干预正在进行的 I/O 的操作都是不被允许的，在这种情形下驱动返回 EOIP。

一旦 I/O 已经完成，应用程序必须与驱动重新同步。

同步是通过以下三个函数来实现:

ibwait: 如果 ibsta 返回值的 CMPL 位置位，则驱动和应用程序同步。

ibstop: 取消 I/O 操作，驱动与应用程序同步。
ibonl:取消 I/O 操作并重置接口状态，驱动与应用程序同步。

可能的错误

EABO, EADR, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibrda (int ud, void *buf, long cnt)

Visual Basic

call ibrda (ByVal ud As Integer, buf As String)

3.1.15 ibrdf

参数说明

ud: 描述符
filename: 存储读取数据的文件名

返回值

ibsta

函数级别

Board / Device level

功能描述

从 device 读取数据到文件。

如果 ud 是 device 描述符，ibrd 从 GPIB 总线上进行寻址，计划读取字节数为 cnt 的数据，并将所读数据存入文件。当 cnt 指定字节数已经全部接收或 END 标志被接收时，操作正常结束。如果传输没有在超时时间内完成，则操作结束并伴随一个错误。在全局变量 ibcntl 中返回实际接收到的字节数。

如果 ud 是 board 描述符，ibrd 计划读取 cnt 字节的数据，并将数据存入文件。当 cnt 指定字节数已经接收或 END 标志被接收时，操作正常结束。下列情况下操作结束并伴随一个错误：1、传输没有在超时时间内完成；2、board 不是 CIC。若 board 不是 CIC，则 CIC 在 GPIB 总线上发送一个设备清除命令。

在 ibcntl 全局变量中返回实际接收数据的字节数。

可能的错误

EABO, EADR, EARG, EBUS, ECIC, EDVR, EFSO, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibrdf ( int ud, const char *filename )
```

Visual Basic

```
call ibrdf (ByVal ud As Integer, ByVal filename As String)
```

3.1.16 ibrpp

参数说明

ud: 描述符

ppr: 并行查询结果

返回值

ibsta

函数级别

Board / Device level

功能描述

执行并行查询。

可能的错误

EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibrpp ( int ud, char *ppr )
```

Visual Basic

```
call ibrpp (ByVal ud As Integer, ppr As Integer)
```

3.1.17 ibrsc

参数说明

ud: Board 描述符

v: 0: 释放系统控制权。1: 请求系统控制权。

返回值

ibsta

函数级别

Board level

功能描述

通过向设备发送 IFC 和 REN 来请求或释放系统控制权。如果接口释放控制权，则不允许重新请求控制权。

可能的错误

EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibrsc ( int ud, int v )
```

Visual Basic

```
call ibrsc(ByVal ud As Integer, ByVal v As Integer)
```

3.1.18 ibrsv

参数说明

ud: Board 描述符

v: 串行查询状态位

返回值

ibsta

函数级别

Board level

功能描述

请求服务和改变串行查询状态位。

可能的错误

EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

ibrsv (int ud, int v)

Visual Basic

call ibrsv (ByVal ud As Integer, ByVal v As Integer)

3.1.19 ibrsp

参数说明

ud: device 描述符

spr: 串行查询结果

返回值

ibsta

函数级别

Device level

功能描述

执行串行查询。如果 ibsta 的 bit 6 被置位（十六进制 40），则 device 正在请求服务。若自动串行查询开启，则 device 可能已经被查询过，在这种情形下，ibrsp 返回之前已获得的状态字节。

可能的错误

EABO, EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP, ESTB

函数语法

Microsoft C/C++ and Borland C++

```
int ibrsp ( int ud, char *spr )
```

Visual Basic

```
call ibrsp(ByVal ud As Integer, spr As Integer)
```

3.1.20 ibsad

参数说明

ud: board 或 device 描述符

v: 设置或禁用 GPIB 二级地址。若 v 为零，二级地址被禁用。若 v 非零，二级地址被开启。

返回值

ibsta

函数级别

Board / Device level

功能描述

设置或禁用 GPIB 二级地址。

可能的错误

EARG, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibsad( int ud, int v )
```

Visual Basic

```
call ibsad (ByVal ud As Integer, ByVal v As Integer)
```

3.1.21 ibsic

参数说明

ud: Board 描述符

返回值

ibsta

函数级别

Board level

功能描述

若接口是系统控制器，则在 100nS 内置上 IFC 线。然后置位 ATN、并使接口成为 CIC 和活动控制器。

可能的错误

EDVR, EHDL, ELCK, ENEB, EOIP, ESAC

函数语法

Microsoft C/C++ and Borland C++

```
int ibsic ( int ud )
```

Visual Basic

```
call ibsic (ByVal ud As Integer)
```

3.1.22 ibsre

参数说明

ud: board 描述符

v: 0:撤销 REN 线；1:置位 REN 线。

返回值

ibsta

函数级别

Board level

功能描述

置位或撤销远程使能（REN）线。如果远程使能线置位，则 GPIB 远程使能（REN）线置上。如果远程使能线撤销，则 REN 不置上。device 是通过 REN 在本地和远程模式操作之间选择。实际上 device 不应该进入远程模式除非它接收到听者地址并且 REN 置上。

可能的错误

EDVR, EHDL, ELCK, ENEB, EOIP, ESAC

函数语法

Microsoft C/C++ and Borland C++

```
int ibsre( int ud, int v )
```

Visual Basic

call ibsre(ByVal ud As Integer, ByVal v As Integer)

3.1.23 ibstop

参数说明

ud: board 或 device 描述符

返回值

ibsta

函数级别

Board / Device level

功能描述

取消异步 I/O 操作模式。如果异步 I/O 正在进行时，在错误位设置状态字节和返回值为 ibsta 和 EABO，表明 I/O 已成功取消。

可能的错误

EABO, EBUS, EDVR, ENEB

函数语法

Microsoft C/C++ and Borland C++

Int ibstop(int ud)

Visual Basic

call ibstop(ByVal ud As Integer)

3.1.24 ibtmo

参数说明

ud: board 或 device 描述符

v: 超时值。有效超时值如下:

常数	V 值	超时时间
TNONE	0	不超时
T10us	1	10us
T30us	2	30us

T100us	3	100us
T300us	4	300us
T1ms	5	1ms
T3ms	6	3ms
T10ms	7	10ms
T30ms	8	30ms
T100ms	9	100ms
T300ms	10	300ms
T1s	11	1s
T3s	12	3s
T10s	13	10s
T30s	14	30s

T100s	15	100s
T300s	16	300s
T1000s	17	1000s

返回值

ibsta

函数级别

Board / Device level

功能描述

设置 board 或 device 超时时间。超时时间是进行同步 I/O 操作（例如 ibrd 和 ibwrt)、ibwait 或 ibnotify 操作设置了 TIMO 参数时所允许使用的最大时间值。如果操作在超时时间内没有完成，则操作失败。在 ibsta 中返回 TIMO。

可能的错误

EARG, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibtmo(int ud, int v)

Visual Basic

call ibtmo (ByVal ud As Integer, ByVal v As Integer)

3.1.25 ibtrg

参数说明

ud: device 描述符

返回值

ibsta

函数级别

Device level

功能描述

发送触发（GET）信息到指定 device。

可能的错误

EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibtrg (int ud)

Visual Basic

call ibtrg(ByVal ud As Integer)

3.1.26 ibwait

参数说明

ud: board 或 device 描述符

mask: 等待 GPIB 总线事件发生。有效值如下表:

掩码	位	十六进制值	描述
TIMO	14	4000	超时时限
END	13	2000	检测到 END 或 EOS
SRQI	12	1000	SRQ 被置位（仅 board）

RQS	11	800	Device 请求服务（仅 device）
SPOLL	10	400	Board 通过控制器串行查询
——	9	200	未定义
——	8	100	
LOK	7	80	GPIB 接口处于锁定状态
REM	6	40	GPIB 接口处于远程状态
CIC	5	20	GPIB 接口是 CIC
ATN	4	10	ATN 被置位
TACS	3	8	GPIB 接口是讲者
LACS	2	4	GPIB 接口是听者
DTAS	1	2	GPIB 接口处于 device 触发状态
DCAS	0	1	GPIB 接口处于 device 清除状态

返回值

ibsta

函数级别

Board / Device level

功能描述

一直延时等待 mask 中指定的时间或超时状态，直到一个或多个事件发生。如果 mask 为零，ibwait 立即返回更新过的 ibsta。如果 TIMO 设置在 mask 中，当超时时间已经消逝而一个或多个指定事件没有发生，则 ibwait 返回。如果 TIMO 没有设置在 mask 中，函数会无限期的等待一个或多个指定事件发生。现有的 ibwait mask 与 ibsta 各位是相同的。如果 ud 是 device 描述符，有效的 mask 是 TIMO、END、RQS 和 CMPL。如果 ud 是 board 描述符，除了 RQS 外所有的 mask 位都有效。可以利用 ibtmo 函数配置超时时间。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, ENEB, ESRQ, EWIP, ERST

函数语法

Microsoft C/C++ and Borland C++

```
int ibwait ( int ud, int mask )
```

Visual Basic

```
call ibwait (ByVal ud As Integer, ByVal mask As Integer)
```

3.1.27 ibwrt

参数说明

ud:描述符

buf: 缓存区包含要发送的数据字节。

cnt: 已发送的数据字节数。

返回值

ibsta

函数级别

Board / Device level

功能描述

从数据缓存区写入数据到 device。

如果 ud 是 device 描述符，ibwrt 从内存写入 cnt 字节的数据到 GPIB device。当指定字节的数据已发送时操作正常结束。如果数据字节数在超时时间内没有发送成功，则操作结束时有错误伴随。在 ibcntl 全局变量中返回实际发送的字节数。

如果 ud 是 board 描述符，假设 board-level ibwrt GPIB 已经正确设置地址，ibwrt 从缓存区写入 cnt 字节的数据到 GPIB device。当指定数据字节数已发送时操作正常结束。如果指定数据字节数在超时时间内没有发送成功，则操作结束时有错误伴随。如果 board 不是 CIC，则 CIC 在 GPIB 总线上发送设备清除命令。

在 ibcntl 全局变量中返回实际发送的字节数。

可能的错误

EADR, EABO, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ENOL

函数语法

Microsoft C/C++ and Borland C++

int ibwrt (int ud, const void *buf, long count)

Visual Basic

call ibwrt (ByVal ud As Integer, ByVal buf As String)

3.1.28 ibwrta

参数说明

ud: device 描述符

buf: 缓存区包含已发送的数据字节。

cnt: 已发送的数据字节数。

返回值

ibsta

函数级别

Board / Device level

功能描述

从数据缓存区写入异步数据到 device。

如果 ud 是 device 描述符，ibwrt 从内存写入 cnt 字节的数据到 GPIB device。当指定字节的数据已发送时操作正常结束。如果数据字节数在超时时间内没有发送成功，则操作结束时有错误伴随。在 ibcntl 全局变量中返回实际发送的字节数。

如果 ud 是 board 描述符，假设 board-level ibwrt GPIB 已经正确设置地址，ibwrt 从缓存区写入 cnt 字节的数据到 GPIB device。当指定数据字节数已发送时操作正常结束。如果指定数据字节数在超时时间内没有发送成功，则操作结束时有错误伴随。如果 board 不是 CIC，则 CIC 在 GPIB 总线上发送设备清除命令。在 ibcntl 全局变量中返回实际发送的字节数。

当 I/O 正在进行时，异步 I/O 调用(ibcmda, ibrda, ibwrta)能执行其它的非 GPIB 操作。一旦异步 I/O 调用开始，其它 GPIB 调用会严格限制。任何可能干预正在进行的 I/O 的操作都是不被允许的，在这种情形下驱动返回 EOIP。

一旦 I/O 已经完成，应用程序必须与驱动同步。

同步是通过以下四个函数来实现：

ibwait: 如果返回 ibsta 的 CMPL 位被置上，则驱动和应用程序同步。

ibnotify: 如果 ibsta 值通过 ibnotify 返回包含 CMPL，则驱动和应用程序同步。

ibstop: 如果 I/O 被取消，则驱动与应用程序同步。
ibonl: 如果 I/O 被取消和接口被重置，则驱动与应用程序同步。

可能的错误

EADR, EABO, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ENOL

函数语法

Microsoft C/C++ and Borland C++

```
int ibwrta ( int ud, const void *buf, long count )
```

Visual Basic

```
call ibwrta (ByVal ud As Integer, ByVal buf As String)
```

3.1.29 ibwrtf

参数说明

ud: device 描述符
filename: 写入数据的文件名

返回值

ibsta

函数级别

Board / Device level

功能描述

从文件写入数据到 device。

如果 ud 是 device 描述符，ibwrt 从文件写入 cnt 字节的数据到 GPIB device。当指定字节的数据已发送时操作正常结束。如果数据字节数在超时时间内没有发送成功，则操作结束时有错误伴随。在 ibcntl 全局变量中返回实际发送的字节数。

如果 ud 是 board 描述符，假设 board-level ibwrt GPIB 已经正确设置地址，ibwrt 从文件写入 cnt 字节的数据到 GPIB device。当指定数据字节数已发送时操作正常结束。如果指定数据字节数在超时时间内没有发送成功，则操作结束时有错误伴随。如果 board 不是 CIC，则 CIC 在 GPIB 总线上发送设备清除命令。在 ibcntl 全局变量中返回实际发送的字节数。

可能的错误

EABO, EADR, EARG, EBUS, ECIC, EDVR, EFSO, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
int ibwrtf( int ud, const char *file_path )
```

Visual Basic

```
call ibwrtf (ByVal ud As Integer, ByVal filename As String)
```

3.1.30 ibcac

参数说明

ud: Board 描述符

v: 异步或同步方式获取控制权，并进入 Active 状态。

0: 异步方式；1: 同步方式

返回值

ibsta

函数级别

Board Level

功能描述

使指定 GPIB Board 进入 Active 状态并置位 ATN。在调用 ibcac 之前，Board 必须已经是在线控制器(CIC)。使用 ibsic 函数可使 Board 成为在线控制器（CIC)。Board 既可以同步方式也可以异步方式获取控制权。以同步方式获取控制权时，Board 会在不破坏当前数据传输过程的情况下置位 ATN；异步方式获取控制权时，Board 会立即置位 ATN，而不考虑是否破坏当前数据传输。

可能的错误

EARG, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

int ibcac(int ud, int synchronous)

Visual Basic

call ibcac(ByVal ud As Integer, ByVal v As Integer)

3.1.31 ibcmd

参数说明

ud: Board 描述符

buf: 缓存区包含要发送的命令串。

cnt: 已发送的命令字节数。

返回值

ibsta

函数级别

Board level

功能描述

发送 GPIB 命令。命令字节用于设置 GPIB 状态，它们不是用来发送信息到 GPIB 设备。设备相关信息由

ibwrt 发送。在 ibcntl 全局变量中返回命令字节数。

可能的错误

EABO, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ENOL

函数语法

Microsoft C/C++ and Borland C++

int ibcmd(int ud, const void *cmd, long cnt)

Visual Basic

call ibcmd(ByVal ud As Integer, ByVal buf As String)

3.1.32 ibcmda

参数说明

ud: Board 描述符

buf: 缓存区包含要发送的命令串。

cnt: 已发送的命令字节数。

返回值

ibsta

函数级别

Board level

功能描述

发送异步 GPIB 命令。命令字节用于设置 GPIB 状态，它们不是用来发送信息到 GPIB 设备。设备相关信息由 ibwrt 发送。在 ibcntl 全局变量中返回命令字节数。

当 I/O 正在进行时，异步 I/O 调用(ibcmda, ibrda, ibwrta)能执行其它的非 GPIB 操作。一旦异步 I/O 调用开始，其它 GPIB 调用会严格限制。任何可能干预正在进行的 I/O 的操作都是不被允许的，在这种情形下驱动返回 EOIP。

一旦 I/O 已经完成，应用程序必须与驱动同步。

同步是通过以下四个函数来实现：

ibwait: 如果返回 ibsta 的 CMPL 位被置上，则驱动和应用程序同步。

ibnotify: 如果 ibsta 值通过 ibnotify 返回包含 CMPL，则驱动和应用程序同步。

ibstop: 如果 I/O 被取消，则驱动与应用程序同步。

ibonl: 如果 I/O 被取消和接口被重置，则驱动与应用程序同步。

可能的错误

ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ENOL

函数语法

Microsoft C/C++ and Borland C++

int ibcmda (int ud, const void *cmd, long cnt)

Visual Basic

call ibcmda (ByVal ud As Integer, ByVal buf As String)

3.1.33 ibfind

参数说明

boardname: board 名称；例如 GPIB0。

返回值

Board 描述符或-1

函数级别

Board level

功能描述

打开并初始化 GPIB board 描述符，返回的 board 描述符用于后续调用。ibfind 执行效果与 ibonl（1）等价。通过 ibfind 获得的描述符在对其使用 ibonl（0）之前一直有效。如果 ibfind 不能获得有效描述符，则返回-1。在 ibsta 和 iberr 中设置 ERR 位包含 EDVR。

可能的错误

EBUS, ECIC, EDVR, ELCK, ENEB

函数语法

Microsoft C/C++ and Borland C++

int ibfind (const char *boardname)

Visual Basic

call ibfind (ByVal boardname As String, ud As Integer)

3.1.34 ibgts

参数说明

ud: board 描述符

v: 是否执行接受者握手协议的标志。

返回值

ibsta

函数级别

Board level

功能描述

使控制器从活动状态进入空闲备用状态。ibgts 导致 GPIB 接口成为 Standby 控制器并且 ATN 线不置上。

可能的错误

EADR , ECIC, EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibgts (int ud, int shadow_handshake)

Visual Basic

call ibgts (ByVal ud As Integer, ByVal v As Integer)

3.1.35 ibist

参数说明

ud: board 描述符

v: 表示设置或清除 ist 位。

返回值

ibsta

函数级别

Board level

功能描述

设置或清除 board 用于并行查询的特殊状态（ist）位。

可能的错误

EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int ibist (int ud, int ist)

Visual Basic

call ibist (ByVal ud As Integer, ByVal v As Integer)

3.1.36 iblines

参数说明

ud: board 描述符

line_status: 返回 GPIB 控制线的状态信息。

返回值

ibsta

函数级别

Board level

功能描述

返回 GPIB 控制线的状态。返回值的低字节（位 0-7）表示每一个 GPIB 控制线的状态。高字节（位 8-15）表示对应低字节各位的使能情况。每个字节描述如下表:

7	6	5	4	3	2	1	0
EOI	ATN	SRQ	REN	IFC	NRFD	NDAC	DAV

若要确定某根 GPIB 控制线是否置位，首先检查低字节中对应的位；如果确定线能监听（指示 1 的正确位），则检查高字节中相对应的位。如果高位是（1），则相对应的控制线置位；如果高位是（0），则控制线未置位。

可能的错误

EDVR, EHDL, ELCK, ENEB, EOIP

函数语法

Microsoft C/C++ and Borland C++

int iblines(int ud, short *line_status)

Visual Basic

call iblines(ByVal ud As Integer, lines As Integer)

3.1.37 ibbna

参数说明

ud: Board 描述符

cname: Board 别名

返回值

ibsta

函数级别

Board level

功能描述

返回 Board 的别名到 cname 中，该别名可用作 ibfind 的输入参数。

可能的错误

EARG

函数语法

Microsoft C/C++ and Borland C++

int ibbna(int ud, char* cname)

Visual Basic

call ibbna(By Val ud As Integer, cname As String)

3.2 Multi-Device IEEE 488 函数参考

3.2.1 AllSpoll

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的 device 地址列表。

resultList: 在 addrlist 中串行查询到的 device 地址列表。

功能描述

执行串行查询一个或更多的 device。查询结果储存在 resultList 中且数量在 ibcntl 中。

可能的错误

EARG, EABO, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void AllSpoll( int board_desc, const Addr4882_t addressList[], short resultList[] )
```

Visual Basic

```
call AllSpoll (ByVal board_desc As Integer, addressList () As Integer, resultList () As Integer)
```

3.2.2 DevClear

参数说明

board_desc: board 标识符

address: 想要进行设备清除的 device 地址。

功能描述

发送选定设备清除（SDC）命令清除 device。如果地址是常数 NOADDR，则发送通用设备清除（DCL）命令到所有的 device。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void DevClear( int board_desc, Addr4882_t address)
```

Visual Basic

```
call DevClear(ByVal board_desc As Integer, ByVal address As Integer)
```

3.2.3 DevClearList

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的想要进行设备清除的地址列表。

功能描述

清除多个 device。如果地址是常数 NOADDR，则通用 device 清除（DCL）命令发送到所有的 device。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void DevClearList ( int board_desc, const Addr4882_t addressList[] )
```

Visual Basic

```
call DevClearList (ByVal ud As Integer, addressList () As Integer)
```

3.2.4 EnableLocal

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的希望其返回本地模式的 device 地址列表。

功能描述

通过发送返回本地（GTL）GPIB 信息到多个 device 使能其前面板的返回本地按钮。这使得 Device 进入到本地模式。如果 addrlist 仅包含常数 NOADDR，则远程使能（REN）GPIB 线不置上。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ESAC

函数语法

Microsoft C/C++ and Borland C++

```
void EnableLocal( int board_desc, const Addr4882_t addressList[] )
```

Visual Basic

```
call EnableLocal(ByVal ud As Integer, addressList () As Integer)
```

3.2.5 EnableRemote

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的地址列表。

功能描述

通过置位远程使能（REN）GPIB 线开启设备的远程编程。使 device 处于 listen- active 状态。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ESAC

函数语法

Microsoft C/C++ and Borland C++

void EnableRemote (int board_desc, const Addr4882_t addressList[])

Visual Basic

call EnableRemote (ByVal ud As Integer, addressList () As Integer)

3.2.6 FindLstn

参数说明

board_desc: board 标识符

padList: 以 NOADDR 结束的 GPIB 主地址列表。

resultList: 通过 FindLstn 找到的所有听者 device 地址。

maxNumResults: 最多找到 maxNumResults 个设备。

功能描述

在 GPIB 总线上找到听者 device。

该函数检测 padlist 中所有的主地址：如果 device 的主地址在 padlist 中，则主地址将储存到 resultlist；否则，主地址中的所有二级地址都被检测并且被找到的任何 device 地址将储存到 resultlist。ibcntl 中存储 resultlist 中实际包含的地址个数。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ETAB

函数语法

Microsoft C/C++ and Borland C++

```
void FindLstn( int board_desc, const Addr4882_t padList[], Addr4882_t resultList[],
  int maxNumResults )
```

Visual Basic

call FindLstn (ByVal ud As Integer, padList () As Integer, resultList () As Integer,
ByVal maxNumResults As Integer)

3.2.7 FindRQS

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的 GPIB 主地址列表。

result: 查找到的第一个请求服务的设备的状态字节。

功能描述

串行查询所有设备以确定哪个设备正在请求服务，直到找到正在请求服务的设备为止。将串行查询的回复字节置于 result 中。ibcntl 包含请求服务的设备在 addrList 中的索引值。如果没有 device 请求服务，则在 ibcntl 中返回 NOADDR、iberr 中返回 ETAB。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ETAB

函数语法

Microsoft C/C++ and Borland C++

void FindRQS (int board_desc, const Addr4882_t addressList[], short *result)

Visual Basic

call FindRQS (ByVal ud As Integer, addressList () As Integer, result As Integer)

3.2.8 PassControl

参数说明

board_desc: board 标识符

address: 以 NOADDR 结束的 GPIB 主地址列表。

功能描述

向具有 Controller 功能的另一个 GPIB device 传递控制权，通过发送获取控制权（TCT）GPIB 信息到 device。Device 成为 Controller-In-Charge (CIC)且接口不再是 CIC。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

void PassControl(int board_desc, Addr4882_t address)

Visual Basic

call PassControl (ByVal board_desc As Integer, ByVal address As Integer)

3.2.9 PPoll

参数说明

board_desc: board 标识符

result: 并行查询结果。

功能描述

执行并行查询。Board 发送命令到每个 device（请查看 PPollConfig 和 PPollUnconfig）。Controller 能使用并行查询获得一位设备相关状态信息，一次最多可以获得 8 个设备的状态。

可能的错误

EARG, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

void PPoll(int board_desc, short *result)

Visual Basic

call PPoll(ByVal board_desc As Integer, result As Integer)

3.2.10 PPollConfig

参数说明

board_desc: board 标识符

address: 被配置的设备地址。

dataLine: 为设备分配的对并行查询作出响应的数据线。

lineSense: 并行查询响应标志（0 或 1）。

功能描述

通过置位或不置位 GPIB 数据线来配置设备对并行查询的响应情况。如果 lineSense 与 device 的特殊状态（ist）位相同，则指定的 GPIB 数据线在并行查询时置上；否则，在并行查询时数据线不置上。Controller 能使用并行查询获得一位设备相关状态信息，一次最多可以获得 8 个设备的状态。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

void PPollConfig (int board_desc, Addr4882_t address, int dataLine, int lineSense)

Visual Basic

call PpollConfig (ByVal ud As Integer, ByVal address As Integer, ByVal dataLine As Integer, ByVal lineSense As Integer)

3.2.11 PPollUnConfig

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的 device 地址列表。

功能描述

撤销设备并行查询响应配置。如果 addrlist 仅包含常数 NOADDR，则发送 PPU 到所有 GPIB 设备。被撤销并行查询配置的设备将不参与后续并行查询。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

void PPollUnconfig (int board_desc, const Addr4882_t addressList[])

Visual Basic

call PpollUnconfig(ByVal ud As Integer, addressList () As Integer)

3.2.12 RcvRespMsg

参数说明

board_desc: board 标识符

buffer: 储存读取数据的缓存区。

count: 读取字节数。

termination: 数据结束模式（STOPend 或 8 位 EOS 字符）的描述。

功能描述

从 device 读取数据字节。RcvRespMsg 假设接口已处于 listen-active 状态且 device 地址已成为讲者。函数会一直读数直到 count 个字节数被读取或结束条件被满足。如果结束条件是 STOPend，则当读取某字节

并伴随 EOI 置位时将停止读取数据；否则，当检测到 8 位 EOS 字符时也会停止读取数据。实际读取到的字节数存储在 ibcntl 全局变量中。

可能的错误

EABO, EADR, EARG, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

void RcvRespMsg (int board_desc, void *buffer, long count, int termination)

Visual Basic

call RcvRespMsg (ByVal ud As Integer, buf As String, ByVal termination As Integer)

3.2.13 ReadStatusByte

参数说明

board_desc: board 标识符

address: device 地址。

result: 串行查询回复字节。

功能描述

串行查询单个 device。如果响应字节的 bit 6 置位（十六进制 40），则 device 正在请求服务。

可能的错误

EABO, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

void ReadStatusByte (int board_desc, Addr4882_t address, short *result)

Visual Basic

call ReadStatusByte (ByVal ud As Integer, ByVal addr As Integer, result As Integer)

3.2.14 Receive

参数说明

board_desc: board 标识符

address: 读取数据的设备地址。

buffer: 储存读取数据的缓存区。

termination: 数据结束模式（STOPend 或 EOS 字符)。

功能描述

从 device 读取数据到用户指定缓存区。

通过寻址讲者和设置接口为听者，读取数据字节且使数据存入到缓存区。当已接收到 count 个数据字节或

检测到结束状态时操作正常结束。如果结束状态为 STOPend，当接收到一个字节伴随 EOI 置位时停止读取数据；或者，当检测到 8 位 EOS 字符时也将停止读取数据。在 ibcntl 全局变量中返回实际读取到的字节数。

可能的错误

EABO, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void Receive( int board_desc, Addr4882_t address, void *buffer, long count, int termination )
```

Visual Basic

```
call Receive(ByVal ud As Integer, ByVal addr As Integer, buf As String,
  ByVal termination As Integer)
```

3.2.15 ReceiveSetup

参数说明

board_desc: board 标识符

address: 讲者的 device 地址。

功能描述

设置 device 为讲者且接口为听者。该函数通常后跟一个从 device 读取数据的函数——RcvRespMsg。此次

调用有利于多次调用 RcvRspMsg，接收每块数据时无需重新寻址。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

void ReceiveSetup(int board_desc, Addr4882_t address)

Visual Basic

call ReceiveSetup(ByVal ud As Integer, ByVal addr As Integer)

3.2.16 ResetSys

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的 device 地址列表。

功能描述

重置并初始化 device。它包含三个步骤：第一个步骤是通过中断远程使能（REN）线和接口清除（IFC）线重置 GPIB；第二个步骤是通过发送通用 device 清除（DCL）GPIB 信息清除所有的 device；最后一个步骤会导致 device 执行 device-specific 重置并初始化。这些步骤是通过 addrlist 发送信息“*RST\n”到

device 完成的。

可能的错误

EABO, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, ENOL, EOIP, ENEB, ESAC

函数语法

Microsoft C/C++ and Borland C++

void ResetSys (int board_desc, const Addr4882_t addressList[])

Visual Basic

call ResetSys (ByVal ud As Integer, addressList () As Integer)

3.2.17 Send

参数说明

board_desc: board 标识符

address: device 地址。

buffer: 要发送的数据字节。

count: 数据字节数。

eot_mode: 数据结束模式：DABend、NULLend 或 NLend。

功能描述

从数据缓存区写入数据到 device。

当 count 个字节已发送时操作正常结束。如果 eot 模式是 DABend，则在发送完最后字节时置位 EOI；如果 eot 模式是 NULLend，则在发送完最后字节时不置位 EOI；如果 eot 模式是 NLend，则在发送完最后字节时自动添加’\n’并置位 EOI。在 ibcntl 全局变量中包含实际发送的字节数。

可能的错误

EABO, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, ENOL, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void Send ( int board_desc, Addr4882_t address, const void *buffer, long count, int eot_mode )
```

Visual Basic

```
call Send (ByVal ud As Integer, ByVal addr As Integer, ByVal buf As String,
  ByVal eot_mode As Integer)
```

3.2.18 SendCmds

参数说明

board_desc: board 标识符

cmdbuf: 要发送的命令字节。

count: 数据值。

功能描述

发送 GPIB 命令。在 ibcntl 全局变量中返回已发送的命令字节数。

可能的错误

EABO, ECIC, EDVR, EHDL, ELCK, ENOL, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void SendCmds ( int board_desc, const void * cmdbuf, long count )
```

Visual Basic

```
call SendCmds(ByVal ud As Integer, ByVal cmdbuf As String)
```

3.2.19 SendDataBytes

参数说明

board_desc: board 标识符

buffer: 要发送的数据字节。

count: 数据值。

eot_mode: 数据结束模式：DABend、NULLend 或 NLend。

功能描述

从缓存区发送数据到设备。函数假设接口处于讲状态且总线上有设备已成为听者。若 eot 模式是 DABend，则发送完最后字节时置位 EOI；若 eot 模式是 NULLend，则发送完最后字节时不置位 EOI；若 eot 模式是 NLend，则发送完最后字节时自动添加’\n’并置位 EOI。在 ibcntl 全局变量中包含实际发送的字节数。

可能的错误

EABO, EADR, EARG, EHDL, ELCK, EDVR, ENOL, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void SendDataBytes ( int board_desc, const void *buffer, long count, int eotmode )
```

Visual Basic

```
call SendDataBytes(ByVal ud As Integer, ByVal buf As String, ByVal term As Integer)
```

3.2.20 SendList

参数说明

board_desc: board 标识符

addresslist: 向其发送数据的 device 地址列表。

buffer: 要发送的数据字节。

count: 数据值。

eot_mode: 数据结束模式：DABend、NULLend 或 NLend。

功能描述

发送数据字节到多个设备。该函数先寻址 addrlist 描述的所有设备成为听者，然后将数据发送到设备。若 eot 模式是 DABend，则发送数据后置位 EOI；若 eot 模式是 NULLend，则发送数据后不置位 EOI；若 eot 模式是 NLend，则发送完数据时自动添加’\n’并置位 EOI。在 ibcntl 全局变量中包含实际发送的字节数。

可能的错误

EABO, EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void SendList( int board_desc, const Addr4882_t addressList[], const void *buffer,
  long count, int eotmode )
```

Visual Basic

call SendList(ByVal ud As Integer, addressList () As Integer, ByVal buf As String,
ByVal term As Integer)

3.2.21 SendIFC

参数说明

board_desc: board 标识符

功能描述

通过发送接口清除命令重置GPIB。SendIFC是GPIB初始化的一部分。它将强制GPIB成为 Controller-In-Charge。同时也确保所有连接到 GPIB 总线的 device 进入未被寻址状态且接口调用处于空闲状态。

可能的错误

ENEB, ESAC, EHDL, ELCK, EDVR, EOIP

函数语法

Microsoft C/C++ and Borland C++

void SendIFC(int board_desc)

Visual Basic

call SendIFC(ByVal ud As Integer)

3.2.22 SendLLO

参数说明

board_desc: board 标识符

功能描述

发送本地锁定（LLO）信息到所有的 device。当本地锁生效时，仅有 Controller-In-Charge 能通过发送适当 GPIB 信息更改 device 状态。

本地锁用于特殊的本地/远程环境中。在特殊情况下应用 SetRWLS 使 device 处于具有本地锁的远程环境。

可能的错误

EBUS, ECIC, EHDL, ELCK, ENEB, ESAC, EDVR, EOIP

函数语法

Microsoft C/C++ and Borland C++

```
void SendLLO ( int board_desc )
```

Visual Basic

```
call SendLLO (ByVal ud As Integer)
```

3.2.23 SendSetup

参数说明

board_desc: board 标识符

addresslist: 以 NOADDR 结束的 device 地址列表。

功能描述

设置 device 接收数据。SendSetup 使 addresslist 描述的设备进入 listen-active 且使接口进入 talk-active。该函数通常从接口使用 SendDataBytes 发送数据。在多次调用 SendDataBytes 之前用 SendSetup 寻址，则后续每块数据的发送无需重新寻址。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void SendSetup ( int board_desc, const Addr4882_t addressList[] )
```

Visual Basic

```
call SendSetup(ByVal ud As Integer, addrs() As Integer)
```

3.2.24 SetRWLS

参数说明

board_desc: board 标识符

addresslist: 以 NOADDR 结束的 device 地址列表。

功能描述

使 device 处于远程锁定状态。SetRWLS 通过置位远程使能（REN）线使 addrList 描述的设备进入远程模式，然后通过本地锁（LLO）使设备处于锁定状态，不能在设备前面板控制设备直到 Controller-In- Charge 释放本地锁使能本地调用。

可能的错误

EARG, EBUS, ECIC, EDVR, EHDL, ELCK, EOIP, ENEB, ESAC

函数语法

Microsoft C/C++ and Borland C++

```
void SetRWLS ( int board_desc, const Addr4882_t addressList[] )
```

Visual Basic

```
call SetRWLS (ByVal ud As Integer, addressList () As Integer)
```

3.2.25 TestSRQ

参数说明

board_desc: board 标识符

result: SRQ 线状态：如果线置上则非零，否则为零。

功能描述

检查 GPIB 服务请求（SRQ）线的当前状态。如果 SRQ 置上，结果为非零值；否则为零。使用 TestSRQ 获得 GPIB SRQ 线的当前状态；使用 WaitSRQ 等待 SRQ 置上。

可能的错误

ECIC, EDVR, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void TestSRQ ( int board_desc, short *result )
```

Visual Basic

```
call TestSRQ (ByVal ud As Integer, result As Integer)
```

3.2.26 TestSys

参数说明

board_desc: board 标识符

addrlist: 以 NOADDR 结束的 device 地址列表。

resultList: 测试结果列表；在 addrlist 中每个条目对应一个地址。

功能描述

使 device 进行自检。TestSys 发送“*TST?”信息到 device；“*TST?”信息能使它们进入自检程序。从每个 device 的 16 位测试结果代码都是可读的。测试结果 0\n 表明该 device 通过自检。请参阅 device 附件已确定故障代码的含义。任何其它结果表明 device 没有通过自检；ibcntl 的内容取决于返回的错误。如果在超时时间内 device 不发送响应，则测试结果为？且返回错误 EABO。

可能的错误

EABO, EARG, EBUS, EDVR, ECIC, EHDL, ELCK, EOIP, ENEB, ENOL

函数语法

Microsoft C/C++ and Borland C++

```
void TestSys ( int board_desc, Addr4882_t * addrlist, short resultList[] )
```

Visual Basic

```
call TestSys (ByVal ud As Integer, addrlist () As Integer, resultList () As Integer)
```

3.2.27 Trigger

参数说明

board_desc: board 标识符

address: 要被触发的 device 地址。

功能描述

发送 GET 信息到单个 device。如果地址是常数 NOADDR，则 GET 信息发送到当前连接到 GPIB 上的所有处于 listen-active 的 device。

可能的错误

EARG, EBUS, EDVR, ECIC, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void Trigger( int board_desc, Addr4882_t address
```

Visual Basic

```
call Trigger (ByVal ud As Integer, ByVal address As Integer)
```

3.2.28 TriggerList

参数说明

board_desc: board 标识符

addressList: 以 NOADDR 结束的 device 地址列表。

功能描述

发送 GET 信息到多个 device。如果地址列表 addrlist 仅存在常数 NOADDR，则寻址不执行且 GET 信息发送到当前连接到 GPIB 上的所有处于 listen-active 的 device。

可能的错误

EARG, EBUS, EDVR, ECIC, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void TriggerList ( int board_desc, const Addr4882_t addressList[] )
```

Visual Basic

```
call TriggerList (ByVal ud As Integer, addressList () As Integer)
```

3.2.29 WaitSRQ

参数说明

board_desc: board 标识符

result: SRQ 线状态：如果线置上则非零，否则为零。

功能描述

等待直到 device 置位 GPIB 服务请求（SRQ）线。当 WaitSRQ 返回时，如果 SRQ 置位则结果为非零；否则结果为零。使用 TestSRQ 获得 GPIB SRQ 线的当前状态；使用 WaitSRQ 等待 SRQ 置位。

可能的错误

EDVR, ECIC, EHDL, ELCK, EOIP, ENEB

函数语法

Microsoft C/C++ and Borland C++

```
void WaitSRQ( int board_desc, short *result )
```

Visual Basic

```
call WaitSRQ (ByVal ud As Integer, result As Integer)
```

4 注意事项

1、在开启应用软件之前，确保将要用到的所有转换器都已可靠地连接到 PC。
2、在多于一张 FT7190 转换器同时接入 PC 的情况下，确保连接的稳定。
3、在使用 FT7190 转换器的情况下，请勿将其他同类转换器从 PC 上拔下或连接到 PC。

5 附录

5.1 附录 A：状态码

所有调用更新全局状态字，ibsta 包含关于 GPIB 状态和硬件的信息。在每个函数调用后，能够通过 ibsta 的 ERR 位检测是否有错误发生。

ibsta 是 16 位值。在状态发生情况下相应位置 1；在状态不发生情况下相应位置 0。

助记符	位	十六进制值	级别类型	描述
ERR	15	8000	device, board	GPIB 错误
TIMO	14	4000	device, board	超时
END	13	2000	device, board	检测到 END 或 EOS
SRQI	12	1000	Board	发生 SRQ 中断
RQS	11	800	device	Device 请求服务
——	10	400	board	未定义

——	9	200	board	
CMPL	8	100	device, board	完成 I/O 操作
LOK	7	80	board	锁定状态
REM	6	40	board	远程状态
CIC	5	20	board	在控制器
ATN	4	10	board	ATN 被置位
TACS	3	8	board	讲者
LACS	2	4	board	听者
DTAS	1	2	board	Device 触发状态
DCAS	0	1	board	Device 清除状态

5.2 附录 B：错误码

下表是 GPIB 错误代码，仅当 ERR 置位时 iberr 有意义。查看错误描述可确定错误原因并寻求解决方案。

错误助记符	iberr 值	描述
EDVR	0	操作系统错误
ECIC	1	函数的执行需要 Board 是在线控制器(CIC)
ENOL	2	在 GPIB 总线上没有听者
EADR	3	GPIB 接口没有被正确寻址
EARG	4	函数所用参数无效
ESAC	5	GPIB 卡不具备函数所需要的系统控制能力
EABO	6	I/O 操作异常终止（或超时）
ENEB	7	不存在的 GPIB board
EDMA	8	DMA 错误

EOIP	10	异步 I/O 操作正在进行
ECAP	11	无进行此项操作的能力
EFSO	12	文件系统错误
EBUS	14	GPIB 总线错误
ESRQ	16	SRQ 处于 ON 状态
ETAB	20	列表错误
ELCK	21	接口被锁定
EARM	22	ibnotify 调用 callback 函数失败
EHDL	23	接口或设备描述符无效或超出范围
EWIP	26	函数 ibwait 正在执行
ERST	27	取消了某个操作导致接口复位
EPWR	28	接口掉电

5.3 附录 C：通过 LabVIEW 控制转换卡

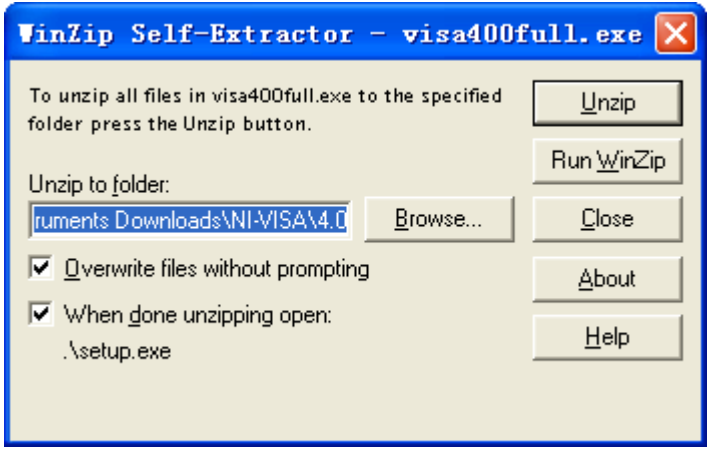
（参见附属光盘上 example 文件夹中的 LabVIEW 实例）

5.4 附录 D：通过 NI Max 控制转换卡的流程简介

5.4.1 安装 NI Max

安装过程中不要更改任何默认配置。

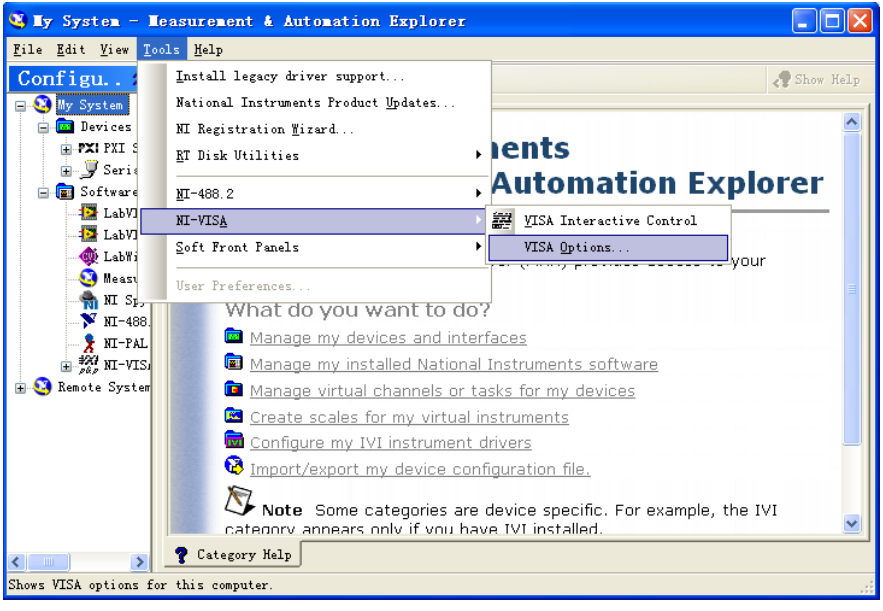
5.4.2 安装 VISA 插件



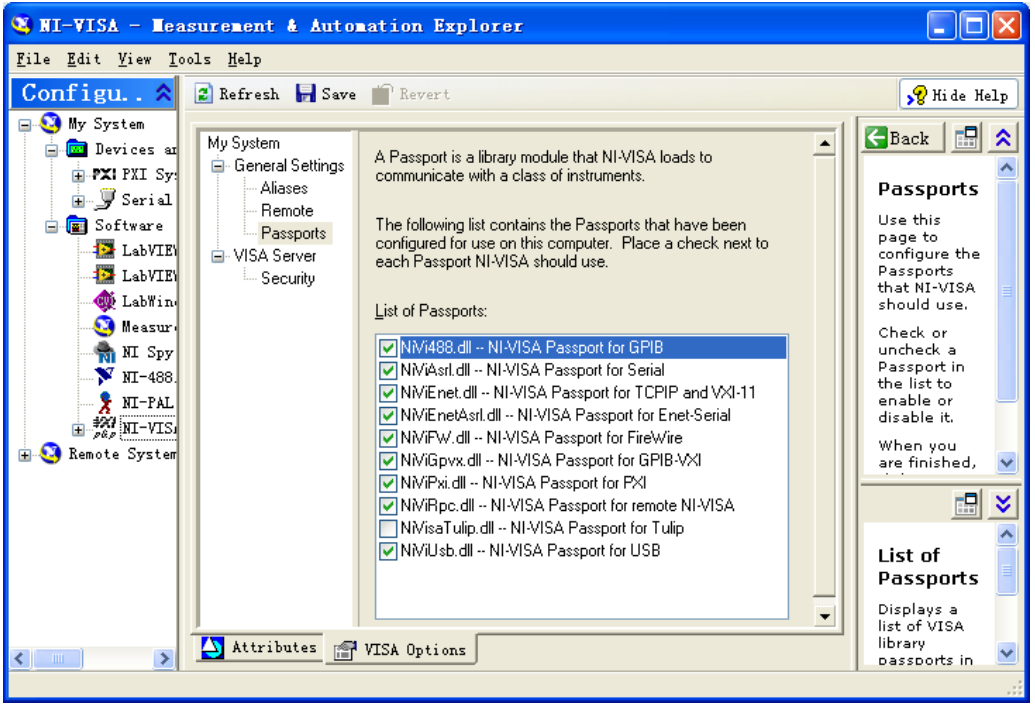
点击上图中的 Unzip 按钮。出现如下图所示安装界面，点击“Next>>”直至安装完成即可。


NI-VISA 4.0
NATIONAL INSTRUMENTS
NI-VISA 4.0
Exit all applications before running this installer.
Disabling virus scanning applications may improve installation speed.
This program is subject to the accompanying License Agreement(s).
NI-VISA™
National Instruments VISA Software
ni.com/visa
© 2006 National Instruments. All rights reserved.
<< Back
Next >>
Cancel

5.4.3 NI Max 的配置

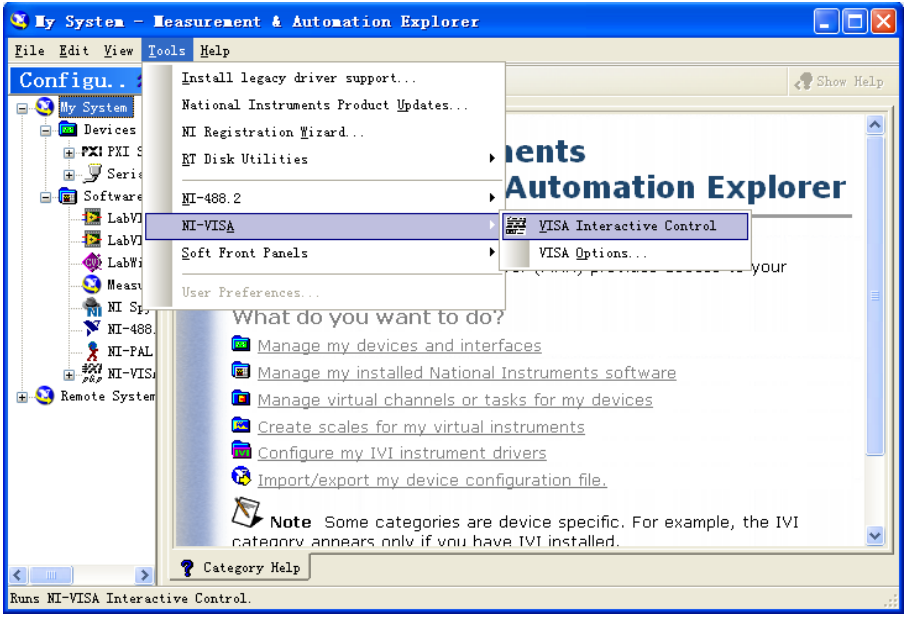


选择菜单栏中的“Tools -> NI-VISA -> VISA Options”出现如下图所示界面。

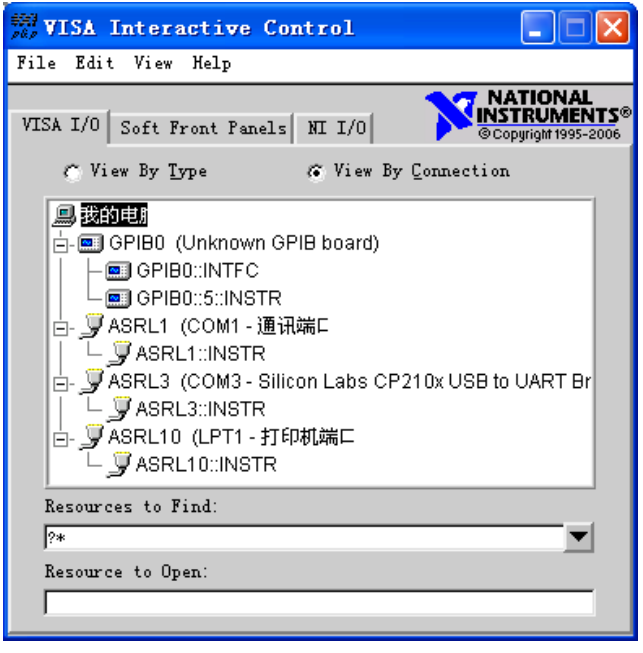


选中图示所有复选框，单击“Save”保存。

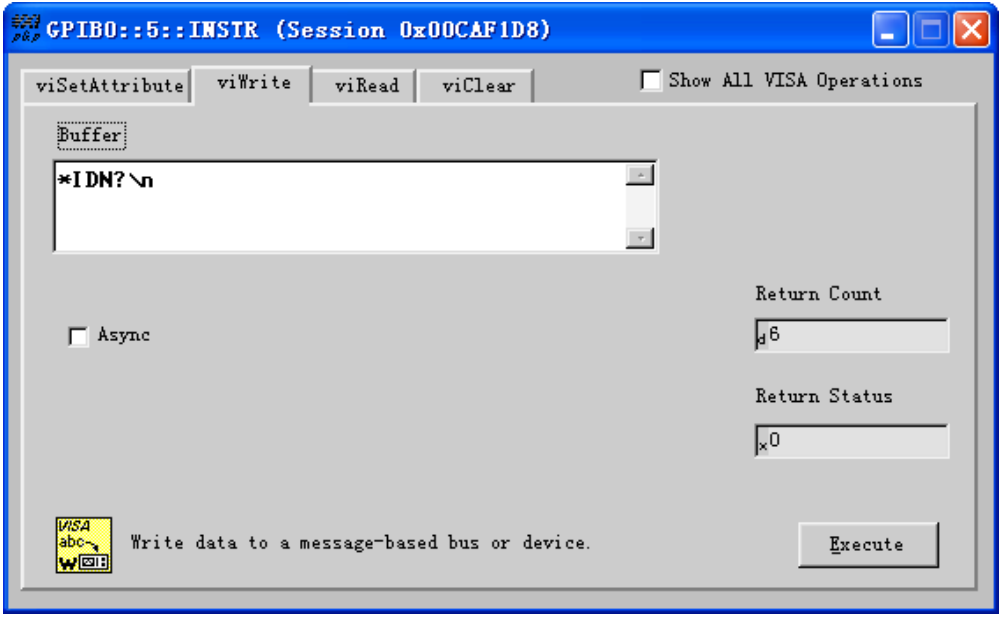
5.4.4 使用 NI Max 操作 FT7190 转换卡



选择菜单栏中的“Tools -> NI-VISA -> VISA Interactive Control”出现如下图所示界面。



图中“GPIB0::5::INSTR”为厂家检测到的示例设备，实际情况可能有所不同。双击此设备即可进行通信。通信示例如下图。



此界面下点击“Executed”可向设备发送信息。在“viRead”标签页下点击“Executed”可从设备读取响应数据。