

# 用 VC++ 编写 USB 人机接口类通信程序

邱 罡<sup>1</sup>, 王伟<sup>2</sup>

(1. 电子科技大学, 四川 成都 610054; 2. 河南工业职业技术学院, 河南 南阳 473009)

**摘 要:**通过对 Visual C++ 6.0 环境下调用 API 函数方法的说明, 来描述如何实现与符合 HID 设备类的 USB 设备接口的通信. 并指出了在 VC 中调用 API 函数的注意事项和方法, 该方法具有很强的通用性, 并经过实践证明具有很好的实用性.

**关 键 词:**通用串行总线; 人机接口设备; API 函数; Visual C++

**中图分类号:** TP311.1

**文献标识码:** A

**文章编号:** 1673-1670(2005)05-0076-03

## 1 引言

在 USB (Universal Serial Bus, 通用串行总线) 出现之前, 计算机的典型接口有并行口、串行口、游戏口等, 与这些接口对应的有各种不同的电缆. 在传输速度方面, 这些接口都存在速度偏低的问题; 在技术方面, 这种设计容易产生 I/O 冲突, 中断请求 (IRQ) 不够用, 以及对于每一种新的外设都必须设计新的接口卡等缺点. 当今的计算机外部设备, 都在追求高速度和高通用性. USB 接口适应了这种要求, 并以其速度快、使用方便、成本低廉等优点, 迅速得到了众多 PC 厂商和半导体厂商的大力支持, 外设向 USB 过渡成为必然的趋势.

但如果主机不知道如何与 USB 外设通信, 那么这个 USB 外设一点用处都没有. Windows 提供了 API 函数来启动应用程序与设备驱动程序之间的通信. 可以使用任何能访问 API 函数的编程语言, 如 VB、C/C++、Delphi 等编写的应用程序, 在设备驱动程序的支持下, 调用 ReadFile、WriteFile、DeviceIoControl 等 API 函数来读/写 USB 设备. 而编写底层总线的驱动程序是一项非常复杂的工程. 为了消除编写设备驱动程序的问题, 对于一些具有相似功能的设备可以组成一类, 分享共有的特性, 便于使用 Windows 提供的共同的类驱动程序. 笔者将介绍在 Visual C++ 6.0 环境下编写与 USB 设备通信的 Windows 程序的编写方法.

## 2 HID 类概述<sup>[1]</sup>

人机接口设备 (HID, Human Interface Device) 类是 Windows 完全支持的第一批 USB 设备类型中的一种, 是指直接和人进行互动的设备, 如鼠标、键盘和游戏柄等. 在 Windows 中, 具有相似属性和提供相似服务的设备被归为一种设备类型, 如音频设备类、通信设备类、人机接口设备类等. 一种类型的设备可以使用一个通用的设备驱动程序. 在运行 Windows 98 或更高版本的 PC 机上, 应用程序可以使用

操作系统内置的 HID 类驱动程序与 HID 通信. 这样使得符合 HID 类的 USB 设备很容易开发与运行. HID 不一定是标准的外设类型, 唯一的要求是交换的数据存储在报表的结构内, 设备固件必须支持报表的格式. 任何工作在该限制之内的设备都可以成为一个 HID, 如温度计、电压计等.

根据 HID 规范, 所有 HID 类设备与主机间的通信均采用控制传输和中断传输这 2 种方式. HID 设备的流数据存放在报表中, 一个 HID 类设备可以支持一个或多个报表, 报表的结构非常有弹性, 可以处理任何类型的数据. 2 种特殊的 HID 类请求: SetReport 和 GetReport 为主机和设备之间的任何类型数据块传输提供了一种方法. 当主机发出 GetReport 请求时, 设备通过控制管道向主机发送相应的流数据; 当主机发出 SetReport 请求时, 主机通过控制管道向设备发送相应的流数据.

## 3 HID 函数介绍

Windows 用来与 HID 设备通信的 API 函数, 包含在 hid.dll、setupapi.dll、kernel32.dll 3 个 dll 文档中, 分别起到与 HID 设备通信、寻找与识别设备、交换数据的作用. hid.dll 与 setupapi.dll 文档内的 API 函数用法可以在 DDK 的帮助文档中找到. kernel32.dll 文档内的 API 函数则是在 MS-DN library 内.

hid.dll 还支持许多 HID 相关函数, 大致可以分为 4 大类:

- 1) 应用程序用来了解 HID 的函数;
- 2) 应用程序用来读/写报表的函数;
- 3) 应用程序设置输入缓冲区来读取报表的函数;
- 4) Windows 与 HID 通信的函数.

其中, 以 HidP 名称开头的函数, 应用程序与驱动程序都可以使用. 以 HidD 名称开头的函数, 只适用于应用程序.

## 4 VC++ 实现与 USB 接口通信实例

下面介绍用 Visual C++ 编写应用程序调用 API 函

收稿日期: 2005-06-15

作者简介: 邱 罡 (1973-), 男, 河南省南阳市人, 电子科技大学自动化工程学院在读硕士.

数,从而实现与硬件通信的过程。

#### 4.1 建立工程

1) 在 VC++ 6.0 下建立一个工程文档。<sup>[2]</sup>

2) 执行 Visual C++ 的命令 Project| settings,即出现一个 settings 对话框。打开 Link 选项卡,在 Category 下拉列表框中选择 Input。然后在 Object/library modules 文本框中输入 hid.dll 与 setupapi.dll。

3) 调用 API 函数。

#### 4.2 VC 中调用 API 函数详述<sup>[3]</sup>

1) 获得 GUID (Globally Unique Identifier),需要调用函数 HidD\_GetHidGuid:

```
VOID HidD_GetHidGuid (OUT LPGUID HidGuid);
```

通过调用它可以得到 HID 类设备的 GUID,应用程序在与 HID 设备通信之前,必须获得 HID 类的独特标志符 GUID,它是一个 128 位值,每一位唯一表示了一个对象。通过这个 API 函数就可以从系统中读取该值,得到 HID 设备句柄。

2) 取得 HID 信息的结构数组

当 SetupDi\_GetClassDevs 函数调用成功时,将返回一个包含所有已连接并检测过的 HID,包含其信息的结构数组的地址,该值在下一个将要调用的函数 SetupDi\_EnumDeviceInterfaces 中将使用到。

SetupDi\_GetClassDevs 函数声明如下:

```
HDEVINFO SetupDi_GetClassDevs(
    IN LPGUID ClassGuid, OPTIONAL
    IN PCTSTR Enumerator, OPTIONAL
    IN HWND hwndParent, OPTIONAL
    IN DWORD Flags
```

```
);
```

3) 识别 HID 接口

SetupDi\_EnumDeviceInterfaces 函数用来读取识别一个接口的结构的指针,每一次调用必须传递一个数组的索引来指定一个接口。

SetupDi\_EnumDeviceInterfaces 函数的声明如下:

```
BOOL SetupDi_EnumDeviceInterfaces
    (IN HDEVINFO DeviceInfoSet,
    IN PSP-DEVINFO-DATA DeviceInfoData,OPTIONAL
    IN LPGUID InterfaceClassGuid,
    IN DWORD MemberIndex,
```

```
OUT PSP-DEVICE-INTERFACE-DATA DeviceInterfaceData);
```

上例中 PSP-DEVICE-INTERFACE-DATA 包含的结构用来识别每一个 HID 的接口。要与设备通信还需要一些更详细的信息,其中最重要的是设备路径,它可以通过函数 SetupDi\_GetDeviceInterfaceDetail 得到。

4) 取得设备路径

SetupDi\_GetDeviceInterfaceDetail 函数用来传回另外一个与前一个函数所识别的接口有关的结构,其中 DevicePath 成员是一个设备路径,应用程序通过此路径来开启与该设

备的通信。

SetupDi\_GetDeviceInterfaceDetail 函数声明如下:

```
BOOL SetupDi_GetDeviceInterfaceDetail
    (IN HDEVINFO DeviceInfoSet,
    IN PSP-DEVICE-INTERFACE-DATA DeviceInterfaceData,
    OUT PSP-DEVICE-INTERFACE-DETAIL-DATA DeviceInterfaceDetailData, OPTIONAL
    IN DWORD DeviceInterfaceDetailDataSize,
    OUT PDWORD RequiredSize, OPTIONAL
    OUT PSP-DEVINFO-DATA DeviceInfoData,OPTION-AL);
```

第 1 次调用该函数时,其中的 DeviceInterfaceDetailDataSize 无法预知,故可以两次调用该函数,第 1 次调用出错,但可以返回正确的 DeviceInterfaceDetailDataSize,第 2 次调用时传递此函数返回值,调用即可成功。通过上述步骤应用程序基本可以建立与设备的连接了。

如想获得更多关于设备能力的信息,还可以使用 HidD\_GetAttributes 函数、HidD\_GetPreparsedData 函数、HidP\_GetCaps 函数,它们都包含在 hid.dll 文件中,分别实现获得厂商 ID、产品 ID 与版本号码,设备的 Usage、Usage Page、报表长度等,在此不再赘述。

5) 取得设备的代号

取得设备的路径后,就可以准备开始与设备通信。首先使用 CreatFile 函数来开启一个 HID 设备,并且取得此设备的代号,并用此代号来与设备交换数据。

CreatFile 函数的声明如下:

```
HANDLE CreatFile
    (LPCTSTR lpFileName,
    DWORD dwDesire Access,
    DWORD dwShareMode,
    LPSECURITY_ATTRIBUTES lpSecurityAttributes,
    DWORD dwCreationDisposition,
    DWORD dwFlagsAndAttributes,
    HANDLE hTemplateFile);
```

当应用程序取得 HID 设备的代号时,就可以利用 ReadFile 函数和 WriteFile 函数读写报表。

6) 与设备交换报表

ReadFile 函数和 WriteFile 函数声明如下:

```
BOOL ReadFile
    (HANDLE hFile,
    LPVOID lpBuffer,
    DWORD nNumberOfBytesToRead,
    LPDWORD lpNumberOfBytesRead,
    LPOVERLAPPED lpOverlapped);
BOOL WriteFile
    (HANDLE hFile,
    LPVOID lpBuffer,
```

DWORD nNumberOfBytesTo Write ,  
LPDWORD lpNumberOfBytesWrite ,  
LPOVERLAPPED lpOverlapped) ;

读写报表缓冲区时,第一个字节是 Report ID,其后是报表数据.报表缓冲区默认是 8 个字节,并且环状排列.因为数据读写发生在主机轮询设备时,并不是由设备触发产生硬件中断,所以,如不能及时读写,新的数据将会覆盖旧的数据,导致报表丢失.当数据读写频繁时应使用特征报表,它可以保证当报表数据没有变化时,HID 不会传送新的数据.

#### 7) 关闭通信

当应用程序结束与 HID 的通信后,必须释放所有之前保留的资源.所涉及的几个 API 函数如下:

BOOL HidD\_FreePreparedData  
(IN PHIDP\_PREPARED\_DATA PreparedData) ;

其作用是释放由 HidD\_GetPreparedData 函数传回的 PreparedData 缓冲区.

函数 SetupDiDestroyDeviceInfoList 的定义如下:

BOOL SetupDiDestroyDeviceInfoList  
(HDEVINFO DeviceInfoSet) ;

其作用是释放由 SetupDiGetClassDevs 函数传回的 hDevInfo 数组.最后还要使用 CloseHandle 函数来关闭通信,释放所有之前保留的资源.

## 5 结论

Windows 提供的用来与 HID 设备通信的 API 函数,可以被应用程序通过任何的计算机语言来调用.除了 HID 以外,大部分的 USB 设备也可以使用这些函数来通信.在 Windows 环境下开发 USB 应用程序可以归结如下:在应用程序可以开始与 HID 交换数据前,应用程序必须先识别该设备并且读取它的报表信息,这些动作需要调用一堆 API 函数.应用程序首先需要寻找连接到系统上的是哪些 HID 设备,然后检验每个信息直到找到所需的属性.如果是客户化的设备,应用程序可以寻找特定的厂商与产品 ID.或者应用程序可以寻找特定类型的设备.最后运用缓冲区进行数据通信.

## 参考文献:

- [1]陈逸. USB 大全[M]. 北京:中国电力出版社,2001: 110 - 115
- [2]刘小石,郑淮,马林伟,等.精通 Visual C++ 6.0 [M]. 北京:清华大学出版社,2000:37 - 41
- [3]萧世文. USB 2.0 硬件设计[M]. 北京:清华大学出版社,2002:300 - 329

## Programming USB Human Interface Device communication with VC++

QIU Gang<sup>1</sup>, WANG Wei<sup>2</sup>

(1. University of Electronic Science and Technology, Chengdu, Sichuan 610054, China ;

2. Henan Polytechnic Institute, Nanyang, Henan 473009, China)

**Abstract :**By introducing how to call the Application Program Interface functions under Visual C++ 6.0, this paper describes a method of communication with the USB interface which accords with the HID class specification. The special points and detail steps of using API function under Visual C++ are stated. This method is generally used and proved to be feasible by practice.

**Key words :**Universal Serial Bus (USB) ; Human Interface Device (HID) ; API Function ; Visual C++