

VC++ 中 DLL 的编程技术及其在数据采集控制中的应用

谭国平 (南京东南大学机械工程系 210096)

摘要:介绍了 VC++ 中实现 DLL 编程的基本方法及其实现的三种基本形式,并且分析了 DLL 技术的优缺点。以一个实际的数据采集与控制程序为例,编制了相应的 A/D 和 D/A 转换程序,阐述了 VC++ 中 DLL 编程的基本流程和技术,此方法已成功的应用于实际项目中,取得了良好效果。

关键词:VC++ DLL D/A 转换 A/D 转换

一、VC++ 6.0 中实现 DLL 编程的方法

1. DLL 工程中的主要文件介绍

由于 DLL 不能独立运行,只能被其他应用程序调用,所以在开发 DLL 应用时,通常先在 VC++ 中创建一个 EXE 应用程序作为主工程(Project);再创建一个 DLL 应用作为主工程的子工程(Sub-Project),并在主工程中调用 DLL。这样,既可以调试 DLL 本身,又可以调试与应用程序的接口。

开发和使用 DLL 需注意三种文件:

(1) DLL 头文件(.H)。DLL 头文件是指 DLL 中说明输出的类或符号(Symbols,如函数)原型或数据结构的 H 文件。一方面它是类或符号的原型说明文件,另一方面当在其他应用程序中调用 DLL 时,要将该文件包含入应用程序的源文件中。

(2) DLL 的引入库文件(.LIB)。引入库文件是 DLL 在编译、链接成功后生成的文件。它的主要作用是:当其他应用程序调用 DLL 时,要将该文件引入应用程序,否则 DLL 无法引入。引入库文件的方法有许多种,如:可将库文件名放入 VC++ 6.0 中“Projects Settings”对话框的“Linker”栏的“Object/Library Modules”编辑框中(通过菜单项 Build 的子菜单项 Settings 激活);或将库文件直接加入应用程序的工程中。

(3) DLL 文件(.DLL)。DLL 文件是应用程序调用 DLL 运行时真正的可执行代码。DLL 应用程序编译、链接成功后,DLL 文件即存在。开发成功后的应用程序在发布时,只需有 EXE 文件和 DLL 文件,不必有 LIB 和 DLL 头文件。

2. VC++ 6.0 中 MFC 支持的三种 DLL 形式

在 VC++ 6.0 中,MFC 支持三种形式的 DLL,下面

介绍其创建及使用过程:

第一种:通常形式的静态 DLL;

(1)创建通常形式的静态 DLL 工程。使用 VC++ 6.0 的 AppWizard 可以创建通常形式的静态 DLL,首先创建一个新的工程,然后选中(MFC AppWizard(dll))选项,输入工程名称、路径等其他信息,单击“OK”,在“MFC AppWizard - Step 1 of 1”对话框中选“Regular DLL with MFC statically linked”,继续其他选项,即可以创建一个“通常形式的静态 DLL”的工程。

(2)从通常形式的静态 DLL 中输出函数。通常形式的静态 DLL 可被任意的 Win32 的应用程序调用。输出 DLL 内部的符号时使用标准“C”接口。如下所示:

```
extern "C" EXPORT YourExportedFunctionName();
```

(3)从应用程序中调用通常形式的静态 DLL 的输出函数。有两种方法可以实现调用:一是与其他形式的 DLL 相同,将 DLL 的引入库文件(.LIB 文件,包括完整的路径)放入 VC++ 中“Projects Settings”对话框的“Linker”栏的“Object/Library Modules”编辑框中(通过菜单项 Build 的子菜单项 Settings 激活);另一种是在应用程序运行时,使用 SDK 函数 LoadLibrary(),将 DLL 调入。这样就可以调用 DLL 中的输出函数,当然,要将包含函数原型说明的头文件包含在相应的代码文件中。

第二种:通常形式的动态链接 MFC 的 DLL;

这种形式的 DLL 工程创建方法与上述的基本相同,只是在“MFC AppWizard - Step 1 of 1”对话框中选“Regular DLL using shared MFC linked”。

从这种形式的 DLL 输出函数的方法也与上述方法类似,但还需使用宏 AFX_MANAGE_STATE 正确转换 MFC 模块的状态,具体方法是在所有的输出函数前加入

下面的代码行:

```
AFX_MANAGE_STATE(AfxGetStaticModuleState  
( ))
```

这种形式的 DLL 也可被任意的 Win32 的应用程序调用。

从应用程序中调用这种形式的 DLL 输出函数时,方法同上所述,但不能使用函数 LoadLibrary()。

第三种:扩展 DLL(动态链接 MFC)

(1)创建扩展 DLL 工程。使用 VC++ 6.0 的 AppWizard 也可以创建扩展 DLL,创建方法同第一种,只是在“MFC AppWizard - Step 1 of 1”对话框中选中间按钮“MFC Extension DLL (Using shared MFC DLL)”。

(2)从 DLL 中输出类和其他符号,如函数。有两种方法达此目的:

①直接输出整个类,不在 .DEF 文件中写出装饰名称(即带@号的名称);

操作方法如下:

在 DLL 的头文件中,在类的定义中添加关键字 AFX_EXT_CLASS,如:

```
class AFX_EXT_CLASS CMyClass: public CDocument  
{  
    //body of class  
};
```

这种方式允许将整个类(CMyClass)输出,不需要在 .DEF 文件中再加入其他装饰名称,其他应用程序即可使用 DLL 中的类(CMyClass)。

②在 .DEF 文件中加入需要输出的变量或函数的装饰名称。这种方式需要在 .DEF 文件写出所有输出函数的装饰名称,适用于输出少量函数的情形。

操作方法是:

在扩展的 DLL 头文件的开始和结束处分别加入如下代码:

```
#undef AFX_DATA  
#define AFX_DATA AFX_EXT_DATA  
//body of your header file  
#undef AFX_DATA  
#define AFX_DATA
```

这几行代码保证扩展 DLL 的正确编译。不使用这些代码可能导致 DLL 编译或链接错误。

然后在 DEF 文件中依次写入输出函数的装饰名称,如:

```
...  
Function1 @1  
Function2 @2  
...
```

这样就可以输出诸如 Function1, Function2 等函数了。

(3)从应用程序中调用 DLL。调用扩展 DLL 的应用程序必须满足以下条件:有一个从类 CWinApp 派生的对象,而且使用 MFC 的共享 DLL 形式(Use MFC In A Shared DLL)。条件满足后,需将扩展 DLL 的引入库文件(.LIB 文件,包括完整的路径)放入 VC++ 中“Projects Settings”对话框的“Linker”栏的“Object/Library Modules”编辑框中(通过菜单项 Build 的子菜单项 Settings 激活)。

注意:引入时应保持版本一致,即在 DEBUG 版本的“Settings”对话框中要加入 DLL 的 DEBUG 版本的引入库;在 RELEASE 版本的“Settings”对话框中要加入 DLL 的 RELEASE 版本引入库。

最后,要在调用扩展 DLL 的文件(如 .CPP 文件)中包含 DLL 工程中包含函数原型说明的头文件,这样才能使用 DLL 的函数。

笔者即是采用主工程(即 EXE 应用程序)与扩展 DLL(动态链接 MFC)子工程并存的方法开发了一个故障诊断应用软件,并利用其中的扩展 DLL 子工程开发出了数据采集与控制模块。

3. 应用程序连接 DLL 时应注意的路径问题

应用程序启动时,它就开始寻找与它连接的 DLL 库,并把它映射到进程的内存空间。为了找到 DLL,操作系统需要查找如下位置:

- (1)EXE 文件运行的目录;
- (2)进程的当前目录;
- (3)Windows 系统目录;
- (4)PATH 环境变量所指目录;

注意:当应用程序在以上四个目录中都找不到所需要的 DLL 文件时,程序就会出错,并且显示相应的出错信息。因此使用时应确保以上四个目录中的其中一个有所需要的 DLL 文件。一般情况下,可以将生成的 DLL 文件拷贝到 Windows 的系统目录或 EXE 文件运行的当前目录。

二、DLL 中数据采集及控制程序编制

在本应用中,硬件采用的是华控自动化系统公司的产品,型号为 HY-1232。这种型号的 A/D、D/A 板可以

直接插入到能与 IBM-PC XT/AT 总线兼容的微型计算机内的任一总线扩展槽中,构成微机数据采集控制系统。A/D 转换触发工作方式采用软件触发方式,转换结果的传输方式有两种:

1. 查询 A/D 完成位然后再读取数据;
2. A/D 转换完成后发中断申请然后由中断服务程序读取数据。在本应用程序中采用的是前一种方法。

在本例中,可以采用上面介绍的三种形式的任何一种,在这里采用的是第三种形式。首先,生成用户的应用程序作为主工程,然后利用 VC++ 的 AppWizard 为主工程添加一个名为 TestDLL 的 DLL 工程(即扩展的 DLL)作为子工程,系统会自动生成 TestDLL.h、TestDLL.cpp、TestDLL.def 等文件,接着自定义一个包含输出函数说明的头文件 TestDLLAPI.h,其内容如下:

```
//TestDLLAPI.h TestDLL Export header file
#ifndef AFX_DATA
#define AFX_DATA AFX_EXT_DATA
extern "C"
{
    float FAR PASCAL EXPORT AcceptData(int basea,
    int num); //数据采集函数
    void FAR PASCAL EXPORT OutputData(int basea,
    float value); //控制模拟输出函数
}
#endif
#define AFX_DATA
```

其中,参数 basea 表示板基地址,num 表示数据采集的通道号,value 表示将要输出的控制电压值。

然后,在 TestDLL.CPP 文件中包括头文件 TestDLLAPI.h,再在文件末尾编写 AcceptData(int basea, int num) 和 OutputData(int basea, float value) 两函数:

```
//数据采集函数的实现:
extern "C"
float FAR PASCAL EXPORT AcceptData(int basea,
int num)
{
    int dbyte;
    float value;
    _outp(basea, num); //切换通道并触发 A/D 转换
    do
    {
        ; //查询 A/D 完成位
    }
}
```

while(!_inp(basea + 5) & 0x80)); //若 A/D 未完成,则继续查询

dbyte = (_inp(basea + 1) & 0xff) | ((_inp(basea + 2) & 0xf) << 8); //读 A/D 结果

value = (float)(dbyte - 2047) * 10/4096; //将 A/D 结果转换成输入电压值

return value; //返回电压值

}
//控制模拟输出函数的实现:

extern "C"

void FAR PASCAL EXPORT OutputData(int basea, float value)

{
int dbyte;

dbyte = (int)((value/5) * 4096); //转换待输出电压值

_outp(basea + 3, (dbyte & 0xff)); //写入低八位数据

_outp(basea + 4, ((dbyte >> 8) & 0xf)); //写入高四位数据

}
在 TestDLL.def 文件中添加输出函数的说明,如下所示:

EXPORTS

AcceptData @1

OutputData @2

这样即完成了 DLL 文件的编写工作,最后编译运行(注意将 TestDLL 设置为当前的活动项目),生成相应的 .DLL 和 .LIB 文件。

显然 DLL 不能单独运行,它必须被某应用程序调用才有意义。在 VC++ 6.0 中调用 DLL 非常简单,按照前面介绍的方法,首先在 EXE 工程(即主工程)项目设置中链接上相应的 .LIB 库(即 TestDLL.lib)文件,其次将 DLL 文件(即 TestDLL.dll)拷贝到 Windows 的 system 目录下或是应用程序的 debug 目录下,最后在程序中包含相应的头文件(即 TestDLLAPI.h),这样就可以直接在程序中引用 DLL 中的函数了。本人编写了一个故障诊断应用软件,成功的引入了该 DLL,在 Windows 98 操作系统和 VC++ 6.0 环境中获得调试通过。

本例子具有较强的实用性,稍加修改可以运用于各种数据采集与控制软件中,而且可以方便自由的扩展、添加它的函数功能,达到实际应用的水平。

三、结论

在 VC++ 6.0 中,可以方便、快捷的生成所需的 DLL 项目,而且 DLL 编程简单、实用,容易掌握。DLL 模块具有很强的通用性和实用性,而且代码的运行效率高,具有开发周期短、软件容易维护等诸多优点,是一种很好的编程方法。在微机数据采集与控制系统中,由于数据采集与控制模块运用频繁,而且要求实时性好,响应快,因此,可以先利用 VC++ 6.0 开发出功能强大的 DLL 模块,然后再在多种不同的用户程序中链接调用,实践证明,这种方法容易实现,而且效果很好,能很好的满足微机数据采集与控制系统的要求。

参考文献

- [1] David Bennett 等著. 徐军等译. Visual C++ 5 开发人员指南. 北京:机械工业出版社,1998.6
- [2] 木林森等著. Windows 95 实用编程与范例. 北京:清化大学出版社,1997.6
- [3] 程耀、宋守许、胡立等编著. Visual C++ 5.0 程序设计教程. 北京:电子工业出版社,1998.9

(来稿时间:1999 年 6 月)