

基于 LabVIEW 7.0 的 I/O 板卡驱动开发

沈正清 张 弘 北京理工大学化工与环境学院(100080)

Abstract

LabVIEW 7.0 is an excellent development environment of virtual instrument,abroadly applied to project application.This paper mainly introduces two methods about developing drivers for I/O boards in LabVIEW 7.0,including Port I/O function and DLL(Dynamic Link Library).

Keywords: LabVIEW 7.0,drivers for I/O boards,Port I/O function,DLL(Dynamic Link Library)

摘要

LabVIEW 7.0 是一个优秀的虚拟仪器开发环境,被广泛运用于工程应用中。本文主要介绍在 LabVIEW 7.0 环境下利用 I/O 端口函数和 DLL(动态链接库)两种方式开发 I/O 板卡驱动。

关键词: LabVIEW 7.0,I/O 板卡驱动,I/O 端口函数,DLL(动态链接库)

NI 公司在推出 LabVIEW 语言的同时,也推出了一系列数据采集卡。对于这类板卡的操作,可以直接采用 LabVIEW 软件提供的 DAQ 库对端口进行操作。但是,一般这类板卡的价格都十分昂贵,而且无法满足某些特殊场合的数据采集需求,因此有必要借助于第三方的 I/O 板卡。本文主要介绍 LabVIEW 7.0 利用自身提供的 I/O 端口函数和 DLL(动态链接库)两种方式开发北京康拓电脑有限公司的 APCI 系列板卡驱动程序。

1 LabVIEW 7.0 对 I/O 地址的访问

LabVIEW 7.0 支持直接访问 I/O 地址。通过 Functions Palette \ Advanced \ Port I/O 目录下的 In Port.vi 和 Out Port.vi 可以实现对 I/O 地址的读写操作,包括 8 位、16 位和 32 位 I/O 地址的读写操作。

In Port.vi 的功能是实现 I/O 地址的读操作,通过设置 I/O 板卡的 address 参数(设定 I/O 地址)和 Select Type 参数(设定读操作的类型:8 位、16 位和 32 位可选),CPU 就可以从相应的 I/O 板卡读到数据。

Out Port.vi 的功能是实现 I/O 地址的写操作,通过设置 I/O 板卡的 address 参数(设定 I/O 地址)、write data(设定 CPU 向 I/O 板卡写入的数据)和 Select Type 参数(设定写操作的类型:8 位、16 位和 32 位可选),CPU 就可以将数据送到相应的 I/O 板卡。

一般 I/O 板卡的驱动开发主要使用 In Port.vi 和 Out Port.vi 来实现,下面以 APCI5488 板卡来说明 I/O 板卡驱动开发的一般过程。APCI5488 是北京康拓公司开发的一块通用光电隔离型 48 通道 12 位 A/D 采集板,它具有多种输入量程,允许电压或电流输入,单通道最高采集速率可以达到 90KHz。

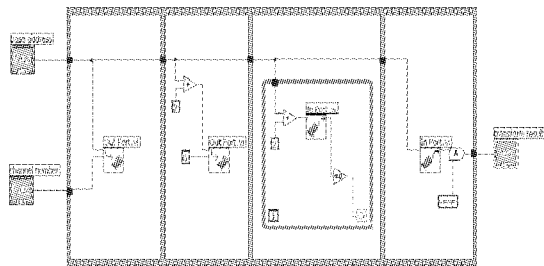


图 1 APCI5488 板卡驱动的开发实例

如图 1 所示,APCI5488 板卡驱动程序的 LabVIEW 7.0 编程如下。

1)设置通道号:写 BASE+0 地址,数据位 D5~D0 译码对应通道号 00H~2FH;

- 2)启动 A/D 转换:写 BASE+2 地址,与数据无关;
- 3)查询 A/D 转换状态:读 BASE+2 地址,D7 位为 1 表示转换结束,D7 位为 0 表示正在转换;
- 4)读 A/D 转换结果:读 BASE+0 地址,D11~D0 为 12 位 A/D 转换结果。

注:BASE 为 APCI5488 板卡的 I/O 基地址。

实践证明,使用 In Port.vi 和 Out Port.vi 开发的 APCI 板卡驱动程序能够很好的满足实际测量测试系统的需要。

2 LabVIEW 7.0 对内存地址访问的局限性

在当今的测量测试系统中,为了满足越来越高的实际要求,智能板卡的使用已经十分普遍。由于绝大多数的智能板上都带有单片机和双口 RAM,因此对内存地址进行访问是驱动程序开发中的需要解决的一个常见技术问题。而 LabVIEW 7.0 不支持内存指针数据类型,即在 LabVIEW 7.0 开发环境下不提供相应的子 vi 来直接访问内存地址,因此有必要通过其他手段使得 LabVIEW 7.0 能够实现内存地址的读写操作。

3 LabVIEW 7.0 对内存访问的实现方法

在 LabVIEW 7.0 开发环境下提供了大量与其他外部代码和应用软件的接口方式,本文论述的内存访问方法是基于 CLFN(Call Library Function Node)调用 DLL 来实现的,其基本思路可以由以下两步来实现。

1)在 Visual C++ 环境下,结合 WINDRVR 编写访问 I/O 端口和内存地址的外部接口函数,并将其编译成 ACCESS.dll 的形式;

2)在 LabVIEW 7.0 环境下,通过 Functions Palette \ Advanced 目录下的 Call Library Function Node 函数来实现对 ACCESS.dll 的调用,如图 2 所示。



图 2 使用 Call Library Function Node 调用 DLL

右键点击 Call Library Function Node,在下拉菜单中选择 Configure 选项,进入配置窗口,如图 3 所示。

Library Name or Path:用于选择用户指定的动态链接库(ACCESS.dll)。

Function Name:列出用户选定的动态链接库中所包含的所有可用的外部接口函数,即 API 函数(W_Mem_8_Read)。

Calling Convention:该列表框有 stdcall(WINAPI)和 C 两个选项,若用户选定的是 Windows API 函数,则选用 stdcall

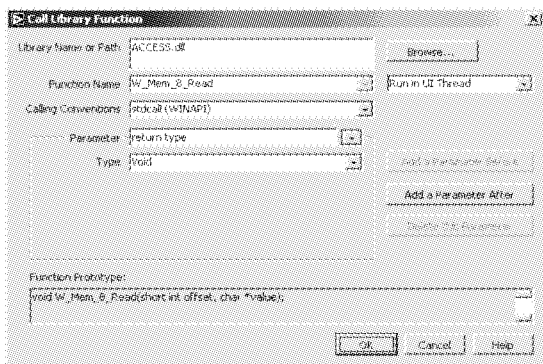


图3 Call Library Function Node 的配置图

(WINAPI)选项;若用户选用的是非 Windows API 函数,则应选择 C 选项。

Parameter、Type:用于设定参数的名称和数据类型。

利用上述方法,将 ACCESS.dll 动态链接库提供的外部接口函数做成子 vi 的形式,以供开发板卡驱动时直接调用。

4 LabVIEW 7.0 内存地址访问实例

下面以 APCI5458 板卡驱动程序为例来说明 LabVIEW 7.0 如何实现对内存地址的访问。APCI5458 采用 89C51 单片机作为智能元件,板上带有双口 RAM 芯片,使其可以在不占用主 CPU 时间的情况下,独立完成 A/D 采集及数据处理,主 CPU 可以在需要的时候随时读取 A/D 采集数据。

如图 4 所示,APCI5458 板卡驱动程序的软件编程如下。

1)初始化 APCI5458 板卡:设置 APCI5458 板卡的 I/O 基地址为 150H,内存基地址为 D0000H;

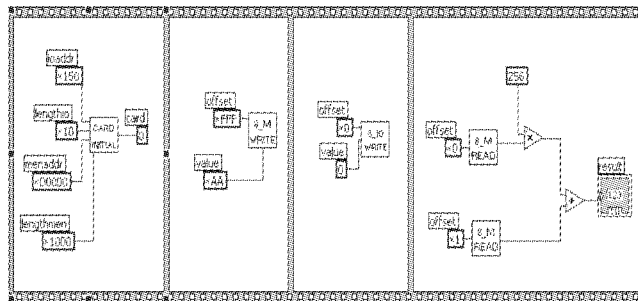


图4 APCI5458 板卡驱动开发实例

2)启动 A/D 转换:向 APCI5458 板卡的 D0FFFH 内存地址写 AAH;

3)主 CPU 向 89C51 单片机发送中断申请:向 I/O 基地址 150H 写入 0;

4)读取 A/D 转换结果:内存地址 D0000H 为转换结果的高 8 位,内存地址 D0001H 为转换结果的低 8 位,并通过前面板显示给用户。

实践证明,利用 Call Library Function Node 调用 DLL 的方式不仅实现了 LabVIEW 7.0 对内存的访问,而且也能够在对 I/O 地址的访问,因此这是一种更加通用的 I/O 板卡驱动开发的方式,能够实现不同类型 I/O 板卡驱动的开发。

参考文献

- 1 National Instruments Corporation. LabVIEW User Manual. April 2003 Edition. Part Number 320999E-01
- 2 杨乐平,李海涛,赵勇等编著. LabVIEW 高级程序设计.北京:清华大学出版社,2003

[收稿日期:2004.6.7]