

COM组件技术在现场总线控制系统组态软件中的应用

李正军, 宋晓庆, 王松磊

山东大学控制科学与工程学院(250061)

2008-08-11

摘要: 简要介绍了现场总线控制系统的组成以及组态软件的功能,重点讲述了COM组件技术在组态软件中的应用及VxD驱动程序的开发。

关键词: 现场总线 组态软件 COM VxD

COM(Component Object Model)组件技术是构造二进制兼容软件的规范,通过它可以建立能够相互传输数据的组件,其服务器-客户机结构非常适合工控软件应用程序的开发。由于工控软件不仅包括PC机上的HMI(人-机界面)程序,还包括与各种基于ISA或PCI总线的数据采集卡进行数据交换的程序,这部分程序对开发人员的硬件水平要求较高,而且开发难度较大,与HMI程序是相互独立的,所以可以把工控软件分成两部分,即把HMI程序作为客户端程序,把与硬件进行数据交换的程序作为服务器端程序。基于这种思想,本文将服务器-客户机结构应用到现场总线控制系统的组态软件中,着重介绍客户机和服务器的功能及实现。首先介绍现场总线控制系统的组成。

1 系统组成

现场总线控制系统主要由PC机、ISA或PCI总线智能适配器、智能测控模块、组态软件、HMI软件、COM服务器、用户软件等构成。

现场总线系统中所有信息的传递都是双向的,COM服务器介于智能适配器和上位机软件之间,负责完成数据的传输。上位机软件相当于客户端应用程序,它使用COM服务器提供的接口来操作适配器,对适配器进行初始化及向特定单元写入和读出数据。

由于在Windows保护模式下不能直接访问存储器,所以需要编写VxD驱动程序,将物理地址转换成线性地址,然后COM就可以象使用DLL一样调用VxD的函数,完成对ISA或PCI总线智能适配器的操作。

从测控模块到上位机软件自下而上的数据传输完成了用户对测控模块的监测;而上层软件通过COM将数据送往适配器,再由适配器送往测控模块,实现了用户对测控模块工作参数的设置及工作状态的管理。图1给出了系统软件结构框图。



图1 系统软件结构框图

2 组态软件的功能

现场总线控制系统组态软件是一套基于Windows 98和Windows 2000平台[或更高版本]、用于快速构造和生成上位机监控系统的组态软件,它提供了从数据采集到数据处理、远程控制、报警处理、报表输出等实际工程问题的完整解决方案。它使用COM服务器提供的接口与适配器进行数据交换,是COM客户端的程序。

3 COM组件技术

组件是完成一定功能的软件块,可以被其它程序使用,而且容易替换。为了使每个人编写的组件具有可移植性,必须建立一个标准,保证其兼容性和可互换性。COM正是这样一种标准,遵循COM规则就可以建立能够相互交换数据的组件。

在现场总线控制系统中,COM组件服务器负责组态软件等上位机软件与智能适配器之间的数据传输,因为适配器通过CAN现场总线与测控模块连接,所以对适配器的操作就是对模块的监测与控制。

COM服务器提供的接口中有适配器初始化、模块检查、向模块发送数据及读取模块数据等函数。下面着重介绍数据发送接收格式及如何编写这4个有代表性的函数。

3.1 适配器初始化函数

只有适配器初始化成功后,才能进行其它操作。由于在Windows保护模式下不能直接访问适配器,COM程序需要调用VxD程序将存储器对应的物理地址转换成线性地址指针lpBaseAddress,这样对适配器的操作就转换成对以该指针为首地址的数组的操作。向这个数组的0x3F0、0x3F1和0x3F8单元分别写入上位机节点号以及适配器与模块间的通信波特率和适配器程序规定的命令字0xC6(表示适配器初始化),等待几十ms后,如果适配器接收到上面的数据并做出适当的反应,它会将0x3F8单元清零,这就表示初始化适配器成功;如果该单元不为零,则初始化失败。

3.2 数据传输格式

适配器初始化成功后,就可以同它交换数据了。下面简单说明一下发送数据和接收数据的格式。

适配器初始化得到的线性地址指针lpBaseAddress的1~5单元分别存放上位机节点号、模块节点号、保留字、发送或接收字节长度及模块操作的命令字。lpBaseAddress[6]~lpBaseAddress[256]存放所要发送的数据;从lpBaseAddress[0x106]单元开始存放

热点专题

- 信心09,冬天来了,春天还会远吗?
- 低功耗技术,是鸡还是蛋?
- 华北计算机系统工程研究所(电子六所)总结表彰暨春节联欢会
- Powerwise高效能解决方案
- 2008Security China中国国际社会公共安全产品博览会
- 视频信号处理技术
- 2008嵌入式技术创新及...
- 2008飞思卡尔技术论坛
- Altera公司SOPC...
- 第十届高交会电子展
- 科技闪耀北京奥运
- ADLINK DAY—2008年量测与自动化技术国际高峰论坛
- 中国电子学会Xilinx杯开放源码硬件创新大赛
- 赛灵思公司Virtex-5系列FPGA
- 3G知识
- IPTV
- 触摸屏技术
- RoHS

杂志精华

- 基于CC2430的无线传感器...
- 无线传感器网络应用系统综述
- 无线传感器网络在野外测量中的...
- 基于竞争的无线传感器网络
- 用于矿井环境监测的无线传感器...
- 具有自适应通信能力的无线传感...
- 基于传感器网络技术的深孔测径...
- 基于无线传感器网络的家庭安防...
- 基于ATmega128L与C...
- 无线传感器网络中移动节点设备...

接收到的数据。IpBaseAddress[0x3F8]存放操作适配器的命令字,适配器根据这个单元内容进行处理,如果是0xC6,则初始化适配器和模块上的CAN控制器;如果是0xC7,则将数组里的数送给模块上的E2PROM,模块收到数据后根据IpBaseAddress[5]的命令字进行相应处理:如果是0xB0,则按照接收到的数据配置模块工作状态;如果是0xA5,则将此时的测量值送到适配器上,由COM程序读出。

3.3 模块检查函数

适配器初始化成功后,还要检查适配器与下面的测控模块是否连接好,或者是否存在组态软件要组态的模块,也就是要进行模块检查操作。模块检查的命令字是0xAD,向数组的1~5单元分别写入上位机节点号、模块节点号、保留字、发送数据长度和模块检查命令字0xAD,向0x3F8单元写入0xC7(表示向适配器写入数据),等待几十ms后,如果0x3F8单元清零而且0x100单元被置为0xAA,表示该模块存在而且可以通信;否则,表明该模块不存在或者硬件上有问题。

3.4 写适配器数据函数

在确定了网络中存在哪些可通信的模块之后,就可以向它们发送数据并进行配置。为了实现向适配器发送数据,总共编写了4个函数、SendData([in]BYTE SendBuf[256])、SendFinish([in]BOOL bFinish)、FinishQuery([out]BOOL *bFinish)和ReceiveResult([out]BOOL *bSendFinish)。SendData负责把一个模块所需要发送的数据以数组的形式放到服务器的一个二维数组(Room[64][256])里,每个模块的数据作为一行。由于向适配器发送数据后,要等待一段时间判断模块是否接收成功,所以在SendFinish中开启辅助线程来发送数据并等待结果,这样可不占用COM主程序的时间,使客户调用接口函数后能立即返回,执行其它操作。FinishQuery查询数据发送是否结束。ReceiveResult弹出一个非模式对话框,显示哪些模块接收到数据,哪些没有。

3.5 读适配器数据函数

除了向适配器发送数据,还可以从适配器上读取模块传上来的数据。读取数据的命令字是0xA5。实现该任务的函数是GetPV([in]BYTE bDesNode,[out]float value[8]),第一个参数是模块节点号,第二个参数是返回的测量值数组。

这里,COM是用ATL编写的本地服务器,COM对象的线程是套间线程。接口定义了6个函数,COM程序流程图如图2所示。



图2 COM程序流程图

COM对象接口的函数声明以及适配器初始化的程序如下:

COM接口定义:

```
interface INCardWork : IDispatch
```

```
{
```

```
    [id(1), helpstring(" 适配器初始化函数,返回值为是否成功" )]
```

```
HRESULT NCardInit([in]BYTE bSrcNode, [in]BYTE bIntrAdd, [in]BYTE bRate, [in]long bSegmentAdd, [out]BOOL *flag):
```

```
    [id(2), helpstring(" 将客户端传送的数组赋值给Room[]") ]]
```

```
HRESULT SendData([in]BYTE SendBuf[256]):
```

```
    [id(3), helpstring[" 启动多线程" ]]
```

```
HRESULT SendFinish([in]BOOL bFinish):
```

```
    [id(4), helpstring(" 此函数返回值表示数据是否已向下位机发送完毕,同时可显示哪些模块未被配置,通常在此函数前先用FinishQuery([out]BOOL *bFinish)查询发送是否完毕" )]
```

```
HRESULT ReceiveResult([out]BOOL *bSendFinish):
```

```
    [id(5), helpstring(" 此函数返回值表示数据是否已向下位机发送完毕," 真" 表示发送完毕" )]
```

```
HRESULT FinishQuery([out]BOOL *bFinish):
```

```
    [id(6), helpstring(" 网络检查,用来在发送数据前检测是否有该节点存在" )]
```

```
HRESULT NetCheck([in]BYTE sour, [in]BYTE des, [in]BYTE type, [out]BOOL *flag):
```

```
    [id(7), helpstring(" 读取模块的测量值" )]
```

```
HRESULT GetPV([in]BYTE bDesNode, [out]float value[256]):
```

```
}
```

适配器初始化函数:

```
#include
```

```
#include " winioctl.h"
```

```

//包含其它头文件
.....
STDMETHODIMP NCardWork::NCardInit(BYTE bSrcNode,
    BYTE bIntrAdd, BYTE bRate, long bSegmentAdd, BOOL
    *flag)
{
NCardCtrl cardctrl: //NCardCtrl类的函数调用VxD函数
exbSrcNode=bSrcNode: //给上位机节点赋值
exbRate=bRate: //下位机与适配器间的通信波特率
BOOL transfersign: //初始化是否成功标志
DWORD dwSegmentaddress=bSegmentAdd:
    //适配器段地址
HANDLE hDevice=NULL: //指向线性指针句柄
lpBaseAddress=[PBYTE]cardctrl.MapLinearAddress(dwSeg-
mentaddress, 0x400, hDevice):
    //调用VxD函数,获得指向ISA总线物理地址的线性
地址指针
cardctrl.UnMapLinearAddress(lpBaseAddress, hDevice):
    //关闭VxD调用适配器初始化函数
_outp[0x310, 0x01]: //打开邮箱锁
lpBaseAddress[0x3F0]=bSrcNodeNumber: //上位机节点号
lpBaseAddress[0x3F1]=bRate: //波特率
lpBaseAddress[0x3F8]=0xC6: //适配器初始化命令字
DrvDelay[20, false]: //延时20ms
..... //初始化后其它操作
_outp[0x310, 00]: //关闭邮箱锁
return S_OK:
}

```

4 虚拟设备驱动程序

VxD是虚拟设备驱动程序(Virtual Device Driver)的缩写,中间的x表示某一设备。它能够无限制地访问所有硬件设备、自由地检测操作系统的数据结构(如描述符和页表)以及访问任何内存位置。

本文中, VxD将ISA总线对应的物理地址转换成一段线性地址,供应用程序使用。VxD的开发工具是VtoolsD,转换时用的函数为MapPhysToLinear。以下是部分程序代码:

```

//定义结构体
typedef struct _MapDevRequest
{
    PVOID mdr_PhysicalAddress: DWORD mdr_SizeInBytes:
    PVOID mdr_LinearAddress: WORD mdr_Status:
}MAPDEVREQUEST, *PMAPDEVREQUEST:
#include
//包含其它头文件
.....
DWORD LdrvDevice::OnW32DeviceIoControl(PIOCTL-
    PARAMS pDIOPParams)
{
    PMAPDEVREQUEST pRea: //自己定义的结构体
    switch(pDIOPParams->dioc_IocCode)
    {
        case IOCTL_OPEN:
        case IOCTL_CLOSEHANDLE: break:
        case MDR_SERVICE_MAP:
            pReq=(PMAPDEVREQUEST*)pDIOPParams->
                dioc_InBuf:
            pReq->mdr_LinearAddress=MapPhysToLinear
                (pReq->mdr_PhysicalAddress, pReq->
                mdr_SizeInBytes, 0):
            if(pReq->mdr_LinearAddress==NULL)
                pReq->mdr_Status=MDR_STATUS_ERROR:
            else
                pReq->mdr_Status=MDR_STATUS_SUCCESS:
            break:
        case MDR_SERVICE_UNMAP: break:
        default:

```

```
return ERROR_INVALID_FUNCTION;
```

```
}
```

```
return DEVIOTL_NOERROR;
```

```
}
```

在现场总线控制系统中使用COM组件技术,不仅可以使数据传输部分的功能独立于客户端程序,减小开发难度,而且使其可以被任何支持二进制代码的程序如Excel 电子表格等直接调用。当系统中采用服务器和客户端方式时,代码更加易于维护。即使要升级服务器端程序,只要接口不变,其客户端程序也完全不需修改,大量后续工作被减轻。象服务器端一样,客户端也只需关心服务器的接口,而不必考虑其如何实现数据交换。也就是说,COM服务器或客户机中的一端功能发生改变,只要其接口保持不变,另一端不需修改就可以工作。本文所介绍的技术已在胜利油田某注水站等实际工程项目中得到成功的应用。

参考文献

- 1 李正军. LON总线在毫伏信号测量智能节点中的应用. 济南: 山东大学学报[工学版], 2002: [32]4
- 2 李正军. 基于LonWorks的热电偶温度测量智能节点的设计. 自动化博览[增刊], 2002(5)
- 3 Ash Rofail, Yasser Shohoud. COM与COM+从入门到精通. 北京: 电子工业出版社, 2000. 4
- 4 阳宪惠. 现场总线技术及其应用. 北京: 清华大学出版社, 1999
- 5 武安河, 周利莉. Windows设备驱动程序(VxD与WDM). 北京: 电子工业出版社, 2001

在线联系

[添加到收藏夹](#)

关于“COM组件技术在现场总线控制系统组态软件中的应用”, 我有如下需求或意向:

用户名:

密码:

验证码:

5829

[欢迎注册](#)

相关应用

- RS-485总线的理论与实践
- 消息中间件在数据交换中的应用研究及其面临的挑战
- 基于COM的OPC技术研究及其接口实现
- 基于CC-Link的炉窑网络控制系统的设计
- Windows平台下压铸机控制系统的研究
- 多媒介通信系统链路管理的设计和实现

[版权声明](#) | [投稿须知](#) | [《电子技术应用》投稿](#) | [网站地图](#) | [帮助中心](#) | [广告中心](#) | [关于我们](#) | [管理员信箱](#)

[回到顶端](#)

《电子技术应用》编辑部版权所有

地址: 北京海淀区清华东路25号电子六所大厦

联系电话: 82306084 / 82306085 传真: 62311179 京ICP备05053646号

推荐分辨率1024*768 IE6.0版本

