

# 基于 VC++ 的 PC 与 PLC 的通信软件设计

李丰堂, 谢明红

(华侨大学 机电及自动化学院, 福建 泉州 362021)

**摘要:**目前 PLC 已广泛应用于工业控制的各个领域, 在现代化建设中发挥了重要作用。在 PLC 的实际应用中, 人们需要对其工作状态进行实时监控, 对不符合要求的参数进行及时修改, 以提高产品的质量。正是根据这种需求设计了基于 VC++ 的 PC 与 PLC 的通信软件。文中介绍了 OMRON PLC 与计算机之间的通信协议以及通信参数的设置, 并用 VC++ 6.0 编写的串口通信软件实现了两者之间的通信, 并给出了通信软件的设计方法。利用这个软件, 人们可以方便直观地对 PLC 的工作状态进行随时监控。

**关键词:** MSCOMM; 串行通信; PLC; VC++

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1673-629X(2008)08-0219-04

## Design of Serial Communication Software Between PC and PLC Based on VC++

LI Feng-tang, XIE Ming-hong

(College of Mechanical & Automation of Huaqiao University, Quanzhou 362021, China)

**Abstract:** Now PLC is widely used in the industry control fields and takes an important part in the modern construction. In the using of the PLC, people need to watch and control its working state and amend the parameters which are not accord with request to improve the quality of products. According to this need, designs the serial communication software between PC and PLC based on VC++. Gives a particular introduction about the communication protocols and the communication parameters between PC and OMRON PLC, and the serial communication software based upon Visual C++ 6.0 is completed to realize the communication between them. The communication principle and software design method are given. Using this software people can watch and control the PLC's working state conveniently and directly.

**Key words:** MSCOMM; serial communication; PLC; VC++

### 0 引言

PLC 是基于电子计算机、且适用于工业现场工作的电控制器, 以其稳定的性能、方便的编程而广泛应用于工业控制领域。它采用了可程序的存储器, 用于其内部存储程序, 执行逻辑运算、顺序控制、定时、计数和算术操作等面向用户的指令, 并通过数字式或模拟式输入输出控制各种类型的机械的生产过程。可编程控制器及其有关外部设备, 都按易于与工业控制系统联成一个整体、易于扩充其功能的原则设计。

PLC 与上位计算机的连接称为 Host Link, PLC 可以接受上位机的命令, 并将执行结果送给上位机, 实现系统的集散控制。用 VC++ 6.0 实现 PLC 与微机通

信的目的是利用 VC 的界面编程能力和数据处理手段, 使得 PLC 的应用更加人性化。文中介绍了 OMRON 公司 CPM1A\*/CPM2A\* 系列 PLC 通用串行接口的通信协议, 以及利用 VC++ 6.0 作为工具实现其与 PC 机的串口通信的方法, 以达到 PC 机与 PLC 之间的通信。

### 1 OMRON PLC 描述

OMRON 公司 CPM1A\*/CPM2A\* 系列 PLC 的 CPU 单元或专用通信单元提供 RS-232 或 RS-422A/455 通用串行通信接口, 这使得 PLC 可以直接连接到上位机显示终端以及其他任何具有 RS-232 接口的设备。

#### 1.1 连接方式

CPM1A\*/CPM2A\* 系列 PLC 与其它具有 RS-232 接口设备有 3 种连接方式:

(1) PLC 与上位机连接;

收稿日期: 2007-11-28

基金项目: 福建省自然科学基金计划资助项目 (E0640007)

作者简介: 李丰堂 (1981-), 男, 山东威海人, 硕士研究生, 研究方向为数控技术; 谢明红, 副研究员, 研究方向为数控发展技术。

(2) PLC 与其他系列 PLC 通过网络连接, 实现主一从通信;

(3) PLC 与具有 RS-232 接口的外部设备连接(无协议连接)。

文中是以 CPM1A\*/CPM2A\* 系列 PLC 为例, 作详细阐述, 通信协议采用的是上位机链接通信<sup>[1]</sup>。上位机链接通信通过在上位机与 PLC 之间交换命令和应答实现的。PLC 与上位机连接方式在数据存储区 DM6650, 其字功能设定如表 1 所示。

表 1 DM6650 功能设定

字	位	功能	设定
DM 6650	00~07	端口设置 00: 标准(1 位起始位, 7 位数据位, 2 位停止位, 偶校验, 9600bps) 01: 在 DM6651 里设置	00 (任何值均可)
	08~11	通过外部端口的 1:1 PC 链接区 0: LR00~LR15	00 (任何值均可)
	12~15	通信模式 0: Host Link; 2: 1:1PC 受控链接; 3: 1:1 PC 主控链接; 4: 1:1INT 链接	4

## 1.2 指令格式

每条指令都以标识号(Unit Number)和操作码/头(Header)开始, 以校验码(FCS)和结束标志(Terminator)结束<sup>[2]</sup>。如表 2 所示, 表中@符号必须置于每条命令的开头; 节点号用于识别 PLC, 它设置在 PLC 的 DM6648 的 00~07 中, 最多可以连接 32 台 PLC; 头代码是 2 字符的命令代码, PLC 的命令代码很多, 这里介绍其中几个: WR(写输入输出寄存器)、RR(读输入输出寄存器)、WD(写数据区)、RD(读数据区)、WH(写保持寄存器)、RH(读保持寄存器)、MS(读操作状态)、SC(写状态数据)。正文部分设置命令的参数, 不同的头代码参数内容不同。如: 头代码为 RR, 则后面 8 位十进制数, 前 4 位表示开始字, 后 4 位则表示读开始字后的几个字。FCS 设置两个字符的帧检查校验码, 其值为一条命令中从@开始到正文结束经过异或计算得到的一个 8 位的二进制数经转化而成的 2 位十六进制字符。终止符设置 \* 和回车两个字符, 表示命令结束。PLC 只有在收到终止符才会发出响应。上位机也只有接收到终止符才会发送下一条指令。在一次交换中传输的命令和应答数据称为一帧, 一个帧最多可包含 131 个数据字符。识别码和正文取决于接收到的上位机联结命令, 结束码表示命令完成的状态(即是否有错误发生)。应答帧格式与命令帧格式类似, 如表 3 所示, 也是以标识号(Unit Number)和操作码/头(Header)开始, 以校验码(FCS)和结束标志(Terminator)结束。表中@符号置于帧的开头, 后面是节点号和头代码。头代码后面两位是尾代码, 不同的尾代码表示不同的含义。例如尾代码为 00 表示正常结束, 13 表

示 FCS 错误, 14 表示格式错误, 15 表示入口码数据错误, 18 表示帧长度错误, A3 表示传输数据时因 FCS 错误引起终止, A8 表示传输数据时因帧长度错误引起终止。尾代码后面是返回的内容, 根据命令帧内容的不同而不同。例如: @00RR0010000140 \* + enter, 表示读起始通道为 0010 的后面一个字的内容。40 是 @00RR00100001 异或以后得到的校验码(FCS)。这时将返回 @00RR00198646 \* + enter, 其中 RR 后面的两位 00 表示正常结束, 后面 3~6 位十六进制 1986 是返回的 0010 中的内容, 46 是 @00RR001986 异或以后得到的校验码(FCS)<sup>[3]</sup>。

表 2 命令帧格式

	节点号	头代码	文字	FCS	终止符
@	$\times 10^1$	$\times 10^0$	.....	$\times 16^1$ $\times 16^0$	* enter

表 3 应答帧格式

	节点号	头代码	尾代码	文字	FCS	终止符
@	$\times 10^1$	$\times 10^0$	$\times 16^1$ $\times 16^0$	.....	$\times 16^1$ $\times 16^0$	* enter

## 2 VC++ 6.0 中串行通信控件的使用

### 2.1 控件的通信方式

Microsoft Communications Control (以下简称 MSComm) 是 Microsoft 公司提供的简化 Windows 下串行通信编程的 ActiveX 控件, 它为应用程序提供了通过串行接口收发数据的简便方法。具体来说, 它提供了两种处理通信问题的方法: 一是事件驱动(Event-driven)方法, 一是查询法。

(1) 事件驱动方式: 事件驱动通讯是处理串行端口交互作用的一种非常有效的方法。在许多情况下, 在事件发生时需要得到通知, 例如, 在串口接收缓冲区中有字符, 或者 Carrier Detect (CD) 或 Request To Send (RTS) 线上一个字符到达或一个变化发生时。在这些情况下, 可以利用 MSComm 控件的 OnComm 事件捕获并处理这些通讯事件。OnComm 事件还可以检查和处理通讯错误。所有通讯事件和通讯错误的列表, 参阅 CommEvent 属性。在编程过程中, 就可以在 OnComm 事件处理函数中加入自己的处理代码。这种方法的优点是程序响应及时, 可靠性高。每个 MSComm 控件对应着一个串行端口。如果应用程序需要访问多个串行端口, 必须使用多个 MSComm 控件<sup>[4]</sup>。

(2) 查询方式: 查询方式实质上还是事件驱动, 但在有些情况下, 这种方式显得更为便捷。在程序的每个关键功能之后, 可以通过检查 CommEvent 属性的值来查询事件和错误。如果应用程序较小, 并且是自保持的, 这种方法可能是更可取的。例如, 如果写一个简

单的电话拨号程序,则没有必要对每接收一个字符都产生事件,因为唯一等待接收的字符是调制解调器的“确定”响应。

## 2.2 MSComm 控件的常用属性

MSComm 控件有很多重要的属性,但首先必须熟悉以下几个属性:

- (1)CommPort 设置并返回通讯端口号。
- (2)Settings 以字符串的形式设置并返回波特率、奇偶校验、数据位、停止位。
- (3)PortOpen 设置并返回通讯端口的状态。也可以打开和关闭端口。
- (4)Input 从接收缓冲区返回和删除字符。
- (5)Output 向传输缓冲区写一个字符串。

## 3 VC++ 6.0 通信程序的具体操作

### 3.1 在项目中插入 MSComm 控件

选择 Project 菜单下 Add To Project 子菜单中的 Components and Controls 选项,在弹出的对话框中双击 Registered ActiveX Controls 项,则所有注册过的 ActiveX 控件出现在列表框中。选择 Microsoft Communications Control, version 6.0,单击 Insert 按钮将它插入到 Project 中来,接受缺省的选项<sup>[5]</sup>(如果在控件列表中看不到 Microsoft Communications Control, version 6.0,那可能是你在安装 VC6 时没有把 ActiveX 一项选上,重新安装 VC6,选上 ActiveX 就可以了),这时在 ClassView 视窗中就可以看到 CMSComm 类了,并且在控件工具栏 Controls 中出现了电话图标。用鼠标将此图标拖到对话框中,程序运行后,这个图标是看不到的。

### 3.2 在对话框中添加控件

向主对话框中添加两个编辑框:一个用于接收显示数据,另一个用于输入发送数据。再添加一个按钮,功能是按一次就把发送编辑框中的内容发送一次。为发送数据对话框添加 CString 变量 m\_ cstrsend,为接收显示数据对话框添加 CString 变量 m\_ edit\_ receive。m\_ edit\_ receive 和 m\_ cstrsend 分别用来放入接收和发送的字符数据。添加用于显示输入输出端口的单选按钮,用于显示 PLC 的输入输出端口状态。注意不要选中 group 属性。

### 3.3 MSComm 的初始化

添加串口事件消息处理函数 OnComm() 打开 ClassWizard -> Message Maps,选择类 CCommTestDlg,选择 IDC\_ MSCOMM1,双击消息 OnComm,在弹出的对话框中将函数名改为 OnComm。在 CCommDlg::OnInitDialog() 中添加代码如下:

```
if(m_ ctrlComm. GetPortOpen())
m_ ctrlComm. SetPortOpen(FALSE);
m_ ctrlComm. SetCommPort(1); //选择 com1
if(! m_ ctrlComm. GetPortOpen())
m_ ctrlComm. SetPortOpen(TRUE); //打开串口
else
AfxMessageBox("cannot open serial port");
m_ ctrlComm. SetSettings("9600, e, 7, 2"); //波特率 9600,
有校验, 7 个数据位, 2 个停止位
m_ ctrlComm. SetRThreshold(1);
//参数 1 表示每当串口接收缓冲区中有多于或等于 1 个字符时将引发一个接收数据的 OnComm 事件
m_ ctrlComm. SetInputLen(0); //设置当前接收区数据长度为 0
ms_ ctrlComm. GetInput(); //先预读缓冲区以清除残留数据
```

### 3.4 校验程序 FCS 的实现

在发送指令(如:@00RR001986)时,需要在指令后面加校验码(FCS)。校验码(FCS)是将指令异或以后得到的。具体实现代码如下:

```
CString CCommDlg::FCS(CString m_ strFrame)
{
int i, count;
unsigned int m_ intMidData = 0, m_ zuo = 0;
count = m_ strFrame. GetLength();
for(i = 0; i < count; i++)
m_ intMidData = m_ intMidData ^ m_ strFrame. GetAt(i);
m_ strFrame. Format("%02X", m_ intMidData);
return m_ strFrame;
}
```

### 3.5 发送数据

首先为发送按钮添加一个单击消息,即 BN\_CLICKED 处理函数,打开 ClassWizard -> Message Maps,选择类 CCommTestDlg,选择 IDC\_ BUTTON\_ MANUALESEND,双击 BN\_CLICKED 添加 OnButtonManualsend() 函数,具体实现代码如下:

```
void CCommDlg:: OnButtonManualsend ()
{
CString pp;
char cEnter = 13;
char xing = 42;
pp = FCS(m_ cstrsend);
UpdateData(TRUE);
m_ cstrsend = m_ cstrsend + pp + xing + cEnter;
m_ ctrlComm. SetOutput(COleVariant(m_ cstrsend)); //发送数据
}
```

### 3.6 接收数据

这个函数是用来处理串口消息事件的,如每当串

口接收到数据,就会产生一个串口接收数据缓冲区中有字符的消息事件,刚才添加的函数就会执行,在 OnComm()函数加入相应的处理代码就能实现自己想要的功能了。具体实现代码如下:

```
void CCMYCommDlg::OnComm()
{
    VARIANT variant_inp;
    COleSafeArray safearray_inp;
    LONG len,k;
    BYTE rxdata[2048]; //
    设置 BYTE 数组 An 8 - bit
    integer that is not signed.
    CString strtemp;
    if(bReceive)
    {
        if ( m_ ctrlComm. Get-
        CommEvent() == 2) //事件
        值为 2 表示接收缓冲区内有
        字符
        { variant_inp = m_ ctrl-
        Comm. GetInput(); //读缓冲
        区
        safearray_inp = variant-
        inp; //VARIANT 型变量转
        换为 ColeSafeArray 型变量
        len = safearray_inp. GetOneDimSize(); //得到有效数据长度
        for(k=0;k<len;k++)
        safearray_inp. GetElement(&k, rxdata + k); //转换为 BYTE
        型数组
        for(k=0;k<len;k++) //将数组转换为 CString 型变量
        {
            BYTE bt = *(char*)(rxdata + k); //字符型
            strtemp. Format("%c",bt);
            m_ edit_receive + = strtemp; //加入接收编辑框对应字符
        }
        UpdateData(FALSE);
    }
}
```

#### 4 结束语

所介绍的利用 MSComm 控件基于 VC++ 6.0 文档视图体系结构开发的 CPM2A \* 系列 PLC 系统监控软件已经成功地运用在 CPM2AE PLC 上。在 PLC 运行长达 8 小时的监控运行中稳定可靠。PC 机定时读取 PLC 36 个输入端口、24 个输出端口的工作状态,并实时显示。图 3 是笔者开发的 PC 机与 OMRON PLC 之间的串行通信软件界面。

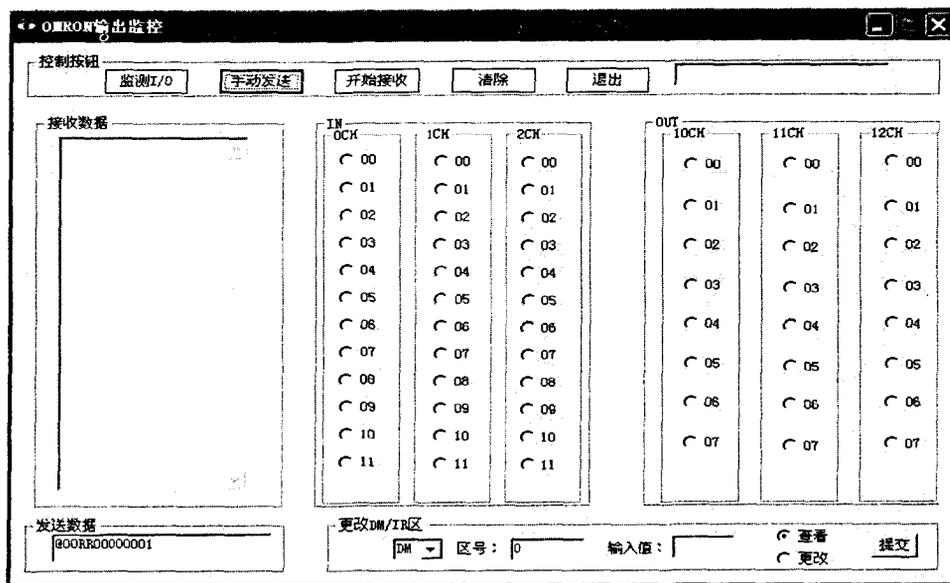


图 1 PC 机与 OMRON PLC 间的串行通信软件界面

#### 参考文献:

- [1] 朱善君. 可编程控制系统原理、应用和维护[M]. 北京: 清华大学出版社, 1992: 15-34.
- [2] OMRON 公司. C200HX/C200HG/C200HE 编程手册[M]. [s. l.]: OMRON 公司, 1997: 75-96.
- [3] 李东晓, 李晓明, 李留振, 等. PC 与 PLC 实时通信的 VisualC++ 6.0 实现[J]. 计算机应用研究, 2002(1): 111-113.
- [4] 何华东, 赵喜荣, 王程远, 等. PLC 与上位计算机的串行通信程序设[J]. 机电工程, 2002, 19(2): 24-26.
- [5] Nelson M. 串口通讯开发指南[M]. 北京: 中国水利水电出版社, 1999: 20-45.

(上接第 218 页)

型微型计算机系统, 2000(11): 1173-1177.

- [2] Nilssen E J. Feature Extraction and Classification of Earth Sounds[D]. Diploma Thesis in Applied Physics. Institute of Mathematical and Physical Sciences, University of Tromsø, 1996.
- [3] Foote J. Content-based Retrieval of Music and Audio[J]. SPIE, 1997, 3229: 138-147.
- [4] Foote J. A Similarity Measure for Automatic Audio Classifica-

tion[C]//AAAI 1997 Spring Symposium on Intelligent Integration and Use of Text, Image, Video and Audio Corpora. USA: Stanford University, 1997: 1-7.

- [5] Foote J. ARTHUR: Retrieving Orchestral Music by Long-term Structure[C]//Proceedings of the International Symposium on Music Information Retrieval. [s. l.]: [s. n.], 2000: 1-7.