

# 利用 MFC 实现 PC 与三菱 PLC 编程口的通信技术

## Communication Technology Realization between PC and PLC Programming Interface by MFC

上海宝信软件股份有限公司自动化部 尚金瑞  
上海翱腾科技有限公司 闫小雪  
Shang Jinrui Yan Xiaoxue

**摘要:** 本文介绍了利用 Visual C++ 中 MFC 技术实现 PC 与三菱 FX2 系列 PLC 编程口进行串行通信的程序设计方法, 并给出了主要程序代码。

**关键词:** MFC 串行通信 PLC VC++ PC

**Abstract:** The article describes the the method of using MFC to realize communication between PC and PLC programming interface. And gives the main source code.

**Key words:** MFC Serial communication PLC VC++ PC

[中图分类号] TP393 [文献标识码] B 文章编号 1606-5123(2004)07-0060-03

## 1 引言

可编程序控制器作为当代工业自动化的主要支柱之一, 随着其应用范围的不断扩大, PLC 与 PC 之间通信技术的应用也越来越广泛。

通过 PLC 的编程口进行通讯不仅可以减少系统的成本及复杂性, 同时可以减少 PLC 方面软件的复杂程度和编程量。

本文根据实际工程中的经验进行总结, 介绍了利用 VC++ 中 MFC 实现 PC 与三菱 FX2 系列 PLC 编程口进行串行通信的程序设计基本方法。

## 2 通信程序设计

在 PC 机和三菱 FX2 系列 PLC 进行串行通信时, 由于 PC 与 PLC 之间的信息传送是通过 PLC 编程口实现的, 采用的是 PLC 专用协议通信指令。因此, 在 PC 编程方面需要严格的遵循 PLC 编程口通讯协议。具体实现介绍如下。

### 2.1 串口设备的打开

在 Visual C++ 中, 利用 MFC CFile 类来实现串行通讯。这种通信方式与访问磁盘普通文件没有太大不同。打开串口设备需作以下操作:

```
CFile file,
CFileException e,
file.Open (
    portName, //example "COM1", "COM2"
    CFile::modeReadWrite,
    &e);
```

### 2.2 串口设备的初始化

串行端口创建时, 必须对其进行设置以匹配与其对话的设备。FX2 系列 PLC 的波特率为固定的 9600bps, 奇偶校验采用偶校验, 1 位停止位, 7 位数据位。

一般地, 可用如下程序设置它们:

```
DCB dcb,
::GetCommState( (HANDLE)file.m_hFile, &dcb );
dcb.BaudRate = 9600;
dcb.ByteSize = 7 ;
dcb.StopBits = 1 ;
dcb.Parity = 2 ;
::SetCommState((HANDLE)file.m_hFile, &dcb );
为了更好的控制端口可以利用 SetCommTimeouts()函
```

数打开或关闭串口超时功能,具体程序如下:

```
COMMTIMEOUTS cto;
::GetCommTimeouts((HANDLE)file.m_hFile,&cto);
cto.ReadIntervalTimeout =0;
cto.ReadTotalTimeoutMultiplier =0;
cto.ReadTotalTimeoutConstant =0;
cto.WriteTotalTimeoutMultiplier=0;
cto.WriteTotalTimeoutConstant =0;
::SetCommTimeouts((HANDLE)file.m_hFile,&cto);
```

### 2.3 PC 与 PLC 间的通信

在 FX2 系列 PLC 与 PC 机的通信中,数据是以帧为单位发送和接收的。其中字符 ENQ(0x05)、ACK(0x06)和 NAK(0x15)作为单个字符,可以构成单字符帧。其余的字符在发送和接收时必须用字符 STX(0x02)和 ETX(0x03)分别表示该字符帧的起始标志和结束标志,否则将构成帧错。一个多字符帧由字符 S T X、命令码、数据、字符 E T X 以及和校验五部分组成,其中和校验值是将命令码到 ETX 之间的所有字符的 ASCII 码(十六进制数)相加,取所得和的最低二位数。多字符帧格式如图 1 所示。表 1 示出 FX2 系列 PLC 与 PC 通信所用的命令码。

STX	命令码	数据(多个字符)	ETX	和校验码高位	和校验码低位
-----	-----	----------	-----	--------	--------

图1 多字符帧的组成

表 1 FX2 系列 PLC 与 PC 通信所用的命令码

命令码	操作数	功能
0	X,Y,M,S,T,C,D	读位软设备的状态或字软设备内的数据
1	X,Y,M,S,T,C,D	对位软设备置“0”或“1”和对字软设备写数
7	X,Y,M,S,T,C	对位软设备强制置“1”
8	X,Y,M,S,T,C	对位软设备强制置“0”

在刚开始通信时,PC 机首先要发送一个控制字符 ENQ(0x05),去查询 PLC 是否做好通信准备,同时也检查一下 PC 机与 PLC 的连接是否正确。若通讯正常,则应答字符 ACK(0x06);若通信有错,则应答字符 NAK(0x15)。

(1) 应用 MFC CFile 类实现对串口设备的读写操作的代码

```
// 读串口
char m_ReadBuff[UINT n];
UINT nByte=file.Read (
&m_ReadBuff, //buffer to store byte
UINT nCount //number of bytes to read
);
```

```
// 写串口
char m_WriteBuff[UINT n];
file.Write (
&m_WriteBuff, //buffer to store byte
UINT nCount //number of bytes to write
);
```

(2)PC 机实现与 PLC 通信的程序代码

```
char m_WriteBuff [10];
char m_ReadBuff [10];
UINT nByte=0;
m_WriteBuff [0]=0x05; //ENQ
// 写串口
file.Write (
m_WriteBuff, //buffer to store byte
1 //number of bytes to write
);
// 读串口
nByte=file.Read (
m_ReadBuff, //buffer to store byte
1 //number of bytes to read
);
switch(m_ReadBuff [0])
{
case 0x06: // 应答 ACK, 通信正常
// 添加相关处理代码
break;
case 0x15: // 应答 NAK, 通信故障
// 添加相关处理代码
break;
}
```

### 2.4 PC 对 PLC 内各软设备进行读、写操作

FX2 系列 PLC 的所有开关量输入、输出以及各软设备对 PC 机都是透明的,只有当 PLC 的计时器和计数器的设定值采用常数时,以及文件寄存器内的数据,PC 机不能对其进行读写。不论 PLC 处在“STOP”状态还是“RUN”状态,PC 机都可以按表 1 所列的命令对 PLC 进行读、写操作。各操作数(即各软设备)的地址见接口模块 F X - 232AW 的用户手册。这里仅给出 PC 机与 PLC 通信所用多字符帧的格式及简单示例,实际应用时只需将多字符帧中的字符 ASCII 码(十六进制)按顺序赋予相应的字符数组 m\_WriteBuff, m\_ReadBuff(参见 2.3 PC 机与 PLC 间的

通信),即可实现对PLC的操作。

(1) 读操作

微机对PLC软设备进行读操作的多字符帧的编制格式

如图2所示。

STX	CMD	16 <sup>3</sup>	16 <sup>2</sup>	16 <sup>1</sup>	16 <sup>0</sup>	16 <sup>1</sup>	16 <sup>0</sup>	ETX	16 <sup>1</sup>	16 <sup>0</sup>
0x02	0x30(0)	软设备首地址				字节数		0x03	和校验	
		0x30(0)	0x30(0)	0x41(A)	0x32(2)	0x30(0)	0x32(2)		0x36(6)	0x38(8)

图2 PC对PLC软设备进行读操作的多字符帧的格式

例如:如果要读取线圈Y20~Y37的状态,查FX-232AW用户手册,PLC输出线圈Y20~Y27的软设备地址为0x00A2,而Y20~Y37是2个字节,和校验值SUM是将命令码到ETX的各个ASCII码相加后取后两位,即:  
SUM = 0x30+0x30+0x30+0x41+0x32+0x30+0x32+0x03 = 0x168,取低两位0x68。

则该多字符帧的格式如图3所示。

STX	CMD	16 <sup>3</sup>	16 <sup>2</sup>	16 <sup>1</sup>	16 <sup>0</sup>	16 <sup>1</sup>	16 <sup>0</sup>	ETX	16 <sup>1</sup>	16 <sup>0</sup>
0x02	0x30	软设备首地址				字节数		0x03	和校验	

图3 多字符帧的格式

则PLC对图3所示的多字符帧的应答多字符帧如图4所示。如果,线圈当前状态如图5所示。则对应图4中的多字符帧如图6所示。

STX	第一字节	第二字节	.....	最后字节 (最多到64字节)	ETX	16 <sup>1</sup>	16 <sup>0</sup>
0x02	高位 低位	高位 低位	.....	高位 低位	0x03	和校验	

图4 PLC对PC读操作应答的多字符帧的格式

Y37	Y36	Y35	Y34	Y33	Y32	Y31	Y30	Y27	Y26	Y25	Y24	Y23	Y22	Y21	Y20
0	0	1	1	0	1	0	1	1	1	0	0	0	0	1	0
3				5				8				4			

图5 线圈当前状态

STX	第一字节		第二字节		ETX	16 <sup>1</sup>	16 <sup>0</sup>
	高位	低位	高位	低位		0x03	和校验
0x02	0x23(3)	0x35(5)	0x38(8)	0x34(4)	0x33(3)	0x44(D)	0x37(7)

图6 多字符帧

(2) 写操作

微机对PLC软设备进行写操作的多字符帧的编制格式

如图7所示。

STX	CMD	16 <sup>3</sup>	16 <sup>2</sup>	16 <sup>1</sup>	16 <sup>0</sup>	16 <sup>1</sup>	16 <sup>0</sup>	第一字节	第二字节	.....	最后字节 (最多到64)	ETX	16 <sup>1</sup>	16 <sup>0</sup>
0x02	0x04	软设备首地址				字节数		高位	低位	.....	高位	低位	0x03	和校验

图7 PC对PLC软设备进行写操作的多字符帧的格式

PLC接收到写操作多字符帧格式后,若接收到数据有效,则应答ACK(0x06),该字符帧如图8所示。

STX	第一字节		ETX	16 <sup>1</sup>	16 <sup>0</sup>
0x02	0	6	0x03	6	9
	0x30	0x36		0x36	0x39

图8 PLC对PC写操作应答字符ACK的多字符帧

若接收数据无效或和校验出错,则应答字符NAK(0x15),该字符帧如图9所示。

STX	第一字节		ETX	16 <sup>1</sup>	16 <sup>0</sup>
0x02	1	5	0x03	6	9
	0x30	0x36		0x36	0x39

图9 PLC对PC写操作应答字符NAK的多字符帧

2.5 关闭串口设备

PC机与PLC通信完毕后,PC机关闭一个已打开的串口设备只需如下一条语句即可完成。

```
file.Close();
```

3 结束语

以上介绍了利用MFC实现PC与FX2系列PLC串行通讯的基本方法及其关键部分程序代码。以上面的程序代码为基础,不仅可以编制用于以PLC为现场主控机的监控系统软件,而且可编制出微机与其他具有串行通信能力设备的串行通信程序。

参考文献

[1] (美) John E.Swanke 著, 前导工作室译. Visual C++ MFC扩展编程实例[M]. 北京:机械工业出版社,2000.  
[2] 廖常初. 可变程序控制器的编程方法与工程应用[M]. 重庆:重庆大学出版社,2001.

作者简介

尚金瑞 工学硕士 多年来一直从事工厂自动化方面的课题研究及项目开发工作。

电磁兼容技术

电磁兼容(EMC)是指设备或系统在所处的电磁环境中能正常工作且不对该环境中任何其他事物构成不能承受的电磁骚扰的能力。电磁兼容技术是一门迅速发展的交叉学科,涉及电子、计算机、通信、航空航天、铁路交通、电力、军事以至人民生活各个方面。在当今信息社会,随着电子技术、计算机技术的发展,一个系统中采用的电气及电子设备数量大大增加,而且电子设备的频带日益加宽,功率逐渐增大,灵敏度提高,联接各种设备的电缆网络也越来越复杂,因此,电磁兼容问题日显重要。