以单片机作为 S7-200 PLC 从站的 PPI 协议的设计

孟 强,梅大成,秦 勃,叶 强
(西南石油大学,四川 成都,610500)

摘 要: 为了将单片机加入到 SIEMENS S7-200 系列 PLC 的 PPI(点对点协议)通信网络中,就需要分析 PPI 通信协议格式。本文利用 CommMonitor6.0(串口监视精灵)工具,监控 PLC 与 PLC 之间的通信,通过总结分析出其数据格式,然后编写相应的单片机 C51 程序,使 PLC 能够使用 NetR/NetW(网络读写命令)与单片机进行数据交换。

关键词: PLC;单片机;PPI;NetR;NetW
中图分类号: TP399 **文献标识码**: A **文章编号**: 1674-7720(2012)17-0057-03

Design of PPI protocol by using single chip microcomputer as the slave station of S7-200 PLC

Meng Qiang, Mei Dacheng, Qin Bo, Ye Qiang
(Southwest Petrolevm University, Chengdu 610500, China)

Abstract: In order to add single chip to the SIEMENS S7-200 series PLC PPI(point to point protocol) communication network, if needs to analyse the PPI communication protocol format. The paper uses CommMontior6.0 to monitor the communication between PLCs, analyzes the data formats, then writes the corresponding C51 program. It achieves the communication between single chip and PLC using NetR/NetW.

Key words: PLC; single chip microcomputer; PPI; NetR; NetW

在工业控制领域,可编程逻辑控制器 PLC(Programmable Logic Controller)以其可靠性高、抗干扰能力强,通用性强、灵活性好、功能齐全、编程简单、使用方便以及安装简便等特点而得到了广泛的应用。现代工业控制系统大都向着分散化、网络化和智能化方向发展,如何实现现场分散的控制设备的网络通信十分重要。

西门子公司的 S7-200 系列 PLC 支持 PPI、MPI、Profibus 和自由口通信等多种通信方式。采用 MPI 协议需要相应的 CP 卡或 MPI 卡支持,如 CP5511 通信卡;若采用 Profibus 协议,则需要 Profibus-DP 模块 EM277;若采用自由口方式,则在 PLC 中需要编写通信程序,占用 PLC 有限的程序存储空间,同时也难以保证在恶劣复杂环境下通信数据的正确可靠性;若采用 PPI 协议,只需在整个通信网络中选定 1 个 PLC 作为通信主站点,其他 PLC 都作为从站点,主站 PLC 通过 NetR/NetW 指令周期性地与从站 PLC 进行数据交换,这种通信方式非常简单可靠,得到了广泛的应用。

在实际应用中通常又需要 PLC 能够与其他设备通信,本文以单片机串口通信为例,详细地分析了 NetR/NetW 指令的通信流程与数据格式,并设计出了单片机串口通信协议,使 PLC 能够使用 NetR/NetW 指令与单片机通信。

1 S7-200 系列 PLC 网络读写指令分析

1.1 PPI 协议简介

PPI 是西门子公司专门为 S7-200 系列 PLC 开发的通信协议,内置于 S7-200 CPU 中。PPI 物理上基于 RS485 接口,通过屏蔽双绞线就可以实现 PPI 通信,是一种主-从通信协议。主站设备发送要求到从站设备,从站设备响应,从站本身不能主动发出信息。为了进行 PPI 通信,S7-200 系列 PLC 专门配备了网络读指令及网络写指令,使用 STEP 7-Micro WIN 中的 NetR/NetW Wizard 可以很方便地配置网络通信。使用该向导可以编辑最多 24 条网络读写指令,每条网络读写指令最多能够读或者写 16 B 的数据。其核心是使用顺序控制指令,

这样在任一时刻只有一条 NetR/NetW 指令有效。在主程序中必须用 SM0.0 指令来调用该向导生成的子程序，以保证它的正常运行。该子程序有 3 个参数：

（1）Timeout（超时）。0 为不计时；1－36767 为设置以秒为单位的超时延时时间。如果通信有问题的时间超出此延时时间，则会报告错误。

（2）Cycle（周期）。所有网络读/写操作每完成一次切换状态。

（3）Error（错误）。0 为无错误；1 为出错，通过检查 NetR/NetW 指令缓冲区状态字节，可以获取错误代码。

1.2 PPI 协议数据帧分析

利用 CommMonitor6.0 工具监控单主站 PLC 之间的通信，可以获得 4 种不同的数据帧。

（1）令牌帧：SD1，DA SA；

（2）无数据字段的固定长度的请求帧或应答帧：SD2，DA SA FC FCS ED；

（3）有可变数据字段的请求或应答帧：SD3，LE LER SD3 DA SA FC DU FCS ED；

（4）短应答帧：SC。

SD1～SD3 为开始定界符，以区别不同类型的帧格式，SD1＝0xDC，SD2＝0x10，SD3＝0x68；LE＝LER，表示从 DA 至 DU 的数据长度；DA 为目的地址，指示接收该帧的站；SA 为源地址，指示发送该帧的站；FC 为帧控制字节，包含用于该帧服务和优先权等的详细说明；DU 为数据字段，包含有效的数据信息；FCS 为帧校验字节，表示从 DA 到 DU 之间的校验和的 256 余数；ED 为帧结束定界符（0x16）；SC 为单一字符（0xE5），用于从站的确认。

当系统主站 PLC 上电运行后，在一定时间（即用户所设定的 Timeout 时间内）会进行通信网络初始化，首先生成令牌并初始化令牌环，由于是单主站系统，该主站将会一直持有该令牌。接着主站就会不断地搜索它管辖范围的从站，通常从用户所配置的第一条 NetR/NetW 指令的从站地址开始，搜索范围也由用户设定（一般为 0～31）。主站首先发送请求帧 10 DA SA FC FCS ED（FC 功能码为 49H，表示有回答要求的从站状态查询），从站正确接收到后将发送响应帧 10 SA DA FC FCS ED（FC 功能码为 00H，表示应答肯定）。接着主站继续搜索下一个从站，一定时间内如果没有从站响应，则将继续进行下一个网络地址搜索。主站 PLC 一直重复循环此过程，并将从站状态信息记录下来，直到 Timeout 时间到，主站才开始真正执行由用户所配置 NetR/NetW 操作。

1.3 NetR 指令分析

由 NetR/NetW 指令向导创建的指令，最多只能读取 16 B 的信息，而且指令是顺序执行的，完成一条读指令需要两次数据收发。在测试过程中，设定主站 PLC 地址为 01，从站 PLC 地址为 02，主站从 PLC 从站的 VB100～VB115 存储区读取 16 B 的通信过程如下：

（1）首先主站 PLC 发出读命令，数据格式为：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
68	1B	1B	68	02	01	6C	32	01	00	00	02	02	00	0E	00	00

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
04	01	12	0A	10	02	00	10	00	01	84	00	03	20	9F	16	

对于读取固定地址的固定长度的 NetR 命令，其中第 6、11、12、31 字节的数据会发生改变，其他数据保持不变。第 6 字节数据为功能码，只有在首次触发时为 6CH，下次读命令则为 7CH，如果无通信故障，会一直保持不变，否则会在 5CH 与 7CH 之间一直交替执行，通信恢复正常后，保持 5CH 或 7CH 不变。第 11、12 字节数据总是相同的，而且每执行一次 NetR 命令，它们的值会增 1，达到 FFH 后，又从 00H 开始。第 31 字节数据为校验和，表示第 4～第 30 字节的数据和的 256 余数。

（2）从站 PLC 接收判断正确后，则作出响应返回 E5。

（3）主站接到从站响应后，则发出确认读命令 10 02 01 5C 5F 16，其中第 3 字节数据 5CH 为功能码。当首次读命令的功能码为 6CH 或 7CH 时，该字节数据为 5CH；当首次读命令功能码为 5CH 时，该字节数据为 7CH。

（4）从站接收到确认读指令后，才会将有用数据返回给主站 PLC，返回数据格式为：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
68	25	25	68	01	02	08	32	03	00	00	02	02	00	02	00	14

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
00	00	04	01	FF	04	00	80	00	11	22	33	44	55	66	77	88

34	35	36	37	38	39	40	41	42								
99	AA	BB	CC	DD	EE	FF	DA	16								

返回的有用数据为第 25～第 40 字节的共 16 B，第 41 字节为第 4～第 40 字节的数据的校验和，而且第 11、12 字节的数据必须与主站读命令的第 11、12 字节保持一致。

这样经过两次收发数据，才能正确完成一次数据的读操作。

1.4 NetW 指令分析

（1）首先主站 PLC 发出写命令，数据格式为：68 2F 2F 68 02 01 6C 32 01 00 00 02 02 00 0E 00 14 05 01 12 0A 10 02 00 10 00 01 84 00 03 20 00 04 00 80 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 30 16。其中，第 6、11、12 字节数据规则与 NetR 命令一致，第 51 字节数据为校验和，第 35～50 字节的数据为真正的要写入从站的有用数据。

（2）从站 PLC 正确接收后，则作出响应，返回 E5。

（3）主站接收到此响应后，则发出确认写命令 10 02 01 5C 5F 16，第 3 字节数据规则与 NetR 命令一致。

（4）从站接收到确认写命令后，返回确认命令 68 12 12 68 01 02 08 32 03 00 00 02 02 00 02 00

01 00 00 05 01 FF 4C 16。第 11、12 字节数据与主站写命令应保持一致，这样收发两次数据，才能完成一次数据的写操作。

2 单片机串口通信协议设计

2.1 串口通信方式选择

PPI 协议物理上采用 RS485 标准，每个字符扩展成 11 bit，采用 NRZ（不归零）编码。首先是 1 bit 开始位，它总是二进制“0”，接着是 8 bit 信息位，之后是 1 bit 奇偶校验位（PPI 协议规定为偶检验），最后是 1 bit 停止位，它总是二进制“1”。

因此，应将单片机串口通信设置为工作方式 3：9 bit UART 通信模式，8 bit 数据位与 1 bit 奇偶检验位，奇偶校验方式使用偶校验；定时器 1 用作波特率发生器，选择工作方式 2，8 bit 自动重装模式，在这里使用 9 600 b/s 波特率，由式（1）、（2）计算可得，TL1=0xFD。

$$baudrate=2^{SMOD}\times T1\ 溢出率/32 \quad (1)$$

$$T1\ 溢出率=f_{osc}/(12\times(256-TL1)) \quad (2)$$

串口通信初始化程序：

```
TMOD|=0x20;
          //定时器 1 选择方式 2，8 bit 自动重装模式
TH1=0xFD;
TL1=0xFD;
PCON &=0x7F;          //SMOD=0，波特率不加倍
SCON=0xD0; //串口通信选择方式 3，9 bit UART 模式，8 bit 数据位，1 bit 校验位
TR1=1;
```

2.2 接收信息起始条件和结束条件选择

在串口通信过程中，单片机有可能从一个字符的中间开始接收字符，从而导致校验错误和接收信息功能终止，为避免出现此类问题，就需要在接收开始前，对信息的起始和结束条件进行定义。

由于 PLC 会发送 3 种不同类型的数据帧，并且单片机需要及时做出正确的响应，因此，单片机在接收到不同的数据帧时应作出不同的响应。单片机采用中断的方式接收数据，而由于在 PPI 协议中，并没有固定的起始字符，经过分析，采用断点检测的方法来作为接收起始条件。断点是指在小于一个完整字符传输时间的一段时间内，接收数据一直为 0，只有在断点之后接收到的字符才会存入到信息缓冲区，任何在断点之前接收到的字符都被忽略。一个完整字符传输时间定义为传输起始位、数据位、校验位和停止位的时间总和。在本系统中，通信波特率为 9 600 b/s，因此传输一个完整的字符（11 bit）时间为 $t=11/9\ 600$，即为 1.145 83 ms，为了方便，断点检测时间可以设定为 2 ms。

信息结束采用字符间隔定时器的方式来判断一条信息的结束。字符间隔时间是指从一个字符的结尾（停止位）到下一个字符的结尾（停止位）之间的时间。在数据传输过程中，如果两个字符之间的时间间隔超过了所设定的时间，则表示这条信息接收完成。由于定时器总是包含接收一个完整字符的时间，因此该时间值应设置为大于在指定波特率下传输一个字符的时间（在此为 1.145 83 ms），在这里设置为 2 ms。单片机在每接收到一个字符后，都要重启字符间隔定时器，如果超时，则表示信息接收完成。

由于单片机硬件资源有限，只提供 2 个定时器，定时器 1 用作波特率发生器，断点检测和字符间隔定时器的时间都为 2 ms，因此可以共用定时器 0。为了计算方便，定时器 0 选择工作方式 1（16 bit 定时器），初值为 TH0=0xFF，TL0=0xFD。

2.3 数据字符检验程序

接收校验程序如下：

```
ACC=SBUF;
if(RB8==P)
{
  rxd_buf[rxd_count]=ACC;          //暂存到接收缓冲区
  rxd_count++;
}
else                    //接收校验错，则需要重新开始接收
{
  rxd_en=0;                                  //停止接收标志
  rxd_count=0;
  return;
}
```

发送校验程序如下：

```
ACC=txd_buf[txd_count];
TB8=P;
SBUF=txd_buf[txd_count];
```

单片机在接收到一条完整的信息后，首先会进行数据帧分析，通过比较，判断主站 PLC 发送的数据帧类型，并对判断正确的请求帧给予正确的响应，返回给 PLC 正确的数据格式。使用 Keil 开发工具编写 C51 程序代码，采用结构化程序设计思想，程序流程图如图 1 所示。

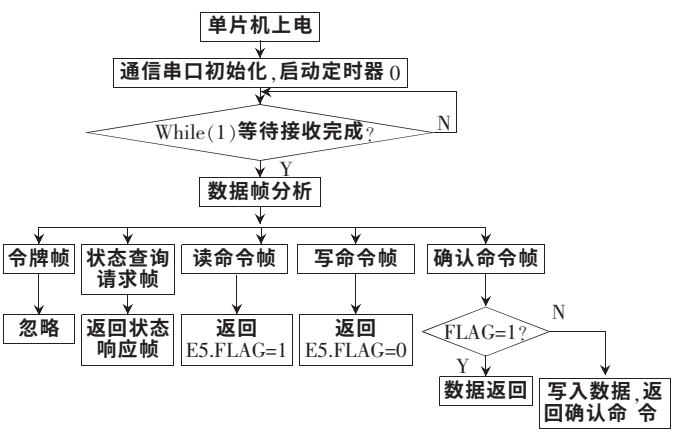

图 1 程序流程图

3 结果验证

最后通过通信测试验证，PLC 主站能够使用 NetR/NetW 指令很方便地读取单片机的数据或向单片机写入给定的数据。而且在具有多个 PLC 从站的 PPI 网络中，通过设定不同从站地址，将多个单片机接入到该网络中，作为主站的 PLC 也能够正常地访问各个从站 PLC 与单片机从站，它们之间的通信稳定可靠，且互不影响，这也为以后在 PPI 网络中扩展其他智能设备提供了可行性。

参考文献

[1] 张扬，蔡春伟，孙明建.S7-200 PLC 原理与应用系统设计[M].北京：机械工业出版社，2007.

[2] 孙鹤旭，梁涛，云利军.Profibus 现场总线控制系统的设计与开发[M].北京：国防工业出版社，2007.

[3] 马忠梅，籍顺心，张凯，等.单片机的 C 语言应用程序设计[M].北京：北京航空航天大学出版社，2005.

[4] 廖常初.PLC 编程及应用[M].北京：机械工业出版社，2008.

（收稿日期：2012-03-12）

作者简介：

孟强，男，1985 年生，硕士研究生，主要研究方向：计算机应用技术，嵌入式系统。

梅大成，男，1965 年生，教授，主要研究方向：嵌入式系统开发与应用，计算机实时检测及控制。

秦勃，男，1984 年生，硕士研究生，主要研究方向：计算机应用技术，嵌入式系统。