Siemens MPI 协议解析

nono95599nono@163.com

摘要：在使用上位机和西门子 s7300 系列 PLC 实现自动化过程控制当中，选择 MPI 协议进行通信时，PLC 可以不用编程，而且可读写所有数据区，快捷方便。但是西门子公司没有公布 MPI 协议的格式，用户如果想使用 MPI 协议监控，就必须购买其监控产品或第三方厂家的组态软件。这样给用户自主开发带来一定困难，特别是自行开发的现场设备就不能通过 MPI 协议接入 PLC。而采用其它通讯方式也存在编程复杂，需要购买软件和授权等局限性。本文通过数据监视、采集、分析的方法，解析出了 MPI 协议的关键报文格式，可用于实现上位机、现场设备与支持 MPI 协议的 CPU 之间通讯，从而提供了一种高效率低成本的通信方式。

关键字： MPI 协议

前言

工业的现代化，很大程度体现在工业生产过程的自动化，其中信息的传输，数据的交换也成为评价工业自动化水平高低的标准。网络通迅方式的多样化和通迅速率的高速化，使信息交换领域从设备控制层延伸到企业管理层。信息技术的飞速发展，促进了自动化系统结构的变革，以网络为主干的分布式控制系统已成为当今自动化系统的主流趋势。因此，网络通迅的实时性和可靠性，以及网络故障的诊断和排除都成为工业网络通信关注的焦点。MPI网络是西门子工业控制系统中经常用到的一种通迅方式，使用RS485物理接口进行数据传输。下面主要阐述西门子MPI协议的解析方法以及关键报文格式。

MPI协议概述

MPI协议，其英文全名为Multi-point-Interface。在PLC之间可组态为主／主协议或主／从协议．如何操作依赖于设备类型：如果控制站都是s7—300／400系列PLC，那么就建立主／主连接关系，因为MPI协议支持多主站通讯，所有的s7—300 CPU都可配置为网络主站，通过主／主协议可以实现PLC之间的数据交换。如果某些控制站是s7—200系列PLC，则可以建立主／从连接关系，因为s7—200 CPU是从站，用户可以通过网络指令实现s7—300 CPU对s7200 CPU的数据读写操作。

分析思路

西门子 Step 7 V5.4 软件是 S7-300 系列 PLC(包括 ET200S)的开发工具，上位机通过其 PCI 插槽上的通讯卡（CP5613A2）接口以及通讯电缆连接到 PLC 的编程口上，并且通讯卡接口和 PLC 编程口都是 RS485 接口标准。这说明，PC 机实际上是可以通过 RS485 串口同 ET200 CPU(IM151-7)通讯,只是我们不知道通讯协议而已。因此，在上位机上运用西门子提

供的 PRODAVE S7 软件读写 PLC 时， 通过监视通讯口上的数据，我们就有可能分析出通信报文格式。然后，撤掉西门子通讯卡，直接通过 RS485 串口向 PLC 发送报文来验证其正确性，并作进一步的操作。本着这一思想，采用以下步骤获得这些报文。

步骤

硬软件需求

硬件：串口分支器及通讯电缆，西门子 CP5613A2 通讯卡，ADVANTECH 公司 PCI-1601A 通讯卡，西门子 ET200S（IM151-7CPU 以及相关模块）。

软件：step7 v5.4 , simatic net 2006 edition， prodave s7 , serial port monitor，PCI1601A driver，visual c++。

硬件连接如图 1-0

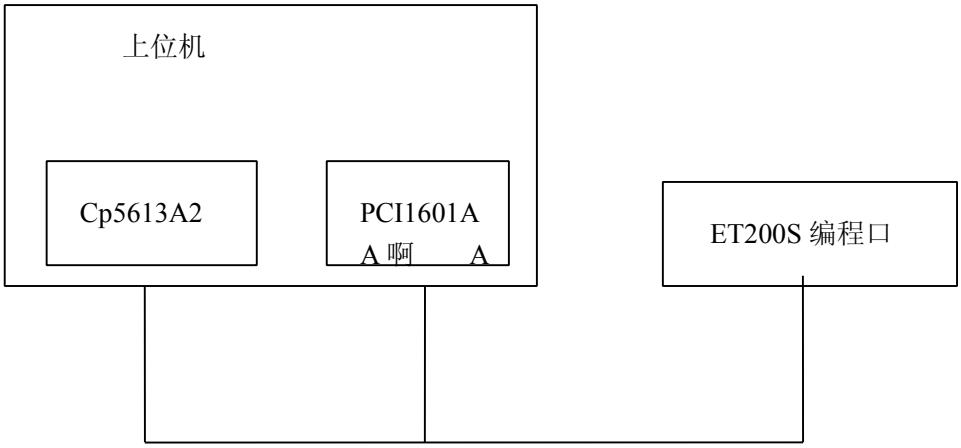


图 1-0

安装完相关软件及驱动程序以后，进行硬件测试以及软件平台搭建

（1） 串口分支器制作及通讯电缆的连接（附录 A）

（2） 运用 STEP 7 V5.4 对 ET200S 组态以及相关初始化设置（附录 B）

（3） PCI1601A 通讯卡的测试（附录 C）

（4） 串口监视软件设置和测试（附录 D）

（5） PRODAVE S7 调试运行（附录 E）

完成设置和调试后，打开串口监视软件，并将 PLC 上电，运行 PRODAVE S7 并在其中进行各种操作（load 、unload、 read 、write 等）时启动数据监视，通过比较分析发现：

（1） 与 S7-200 不同，ET200S 不管出于何种状态（run 或 stop），一经上电，就不断发出数据查找设备，在读写数据过程中也不间断。

（2） 连接、读出、写入和断开时检测到一系列有规律的数据。经过多次监测比较分析，可得到相关操作的数据帧格式，初始化设定 PLC 与上位机的地址分别为 02 和 00；为描述方便，现在对数据帧格式做以下符号约定

SD:(Start Delimiter)开始定界符

LE:（Length）报文长度

LER:（Repeated Length）重复数据长度

SD: (Start Delimiter)开始定界符

DA:（Destination Address）目标地址

SA:（Source Address）源地址

FC:（Function Code）功能码

DSAP:（Destination Service Access Point）目的服务存取点

SSAP:（Source Service Access Point）源服务存取点

FS: （Frame Sequence）帧序列号

UU:（unkown unit）未知操作单元，其数值通常为固定值

GU:(group unit) 分组单元

DU:（Data Unit）数据单元

FCS:（Frame Check Sequence）校验码

END:（End Delimiter）结束分界符

分析结果

连接（load）过程

（1）设备查找

在 PLC(ET200S)上电启动进入 run 状态后，开始不断发出数据查找设备，数据帧格式如下。DA 从 00 到 1F 共 32 个站号，令牌帧和总线访问帧按照严格的帧时序（15 帧/s），交替发出。

令牌帧： DC DA SA

DC DA 02

总线访问帧： SD DA SA FC FCS END

10 DA 02 49 FCS 16

在 prodave s7 中运行 Load 命令后，上位机也开始不断发出数据查找设备，数据帧格式如下。DA 从 00 到 1F 共 32 个站号，令牌帧和总线访问帧按照严格的帧时序（19 帧/s），交替发出。

令牌帧： DC DA SA

DC DA 00

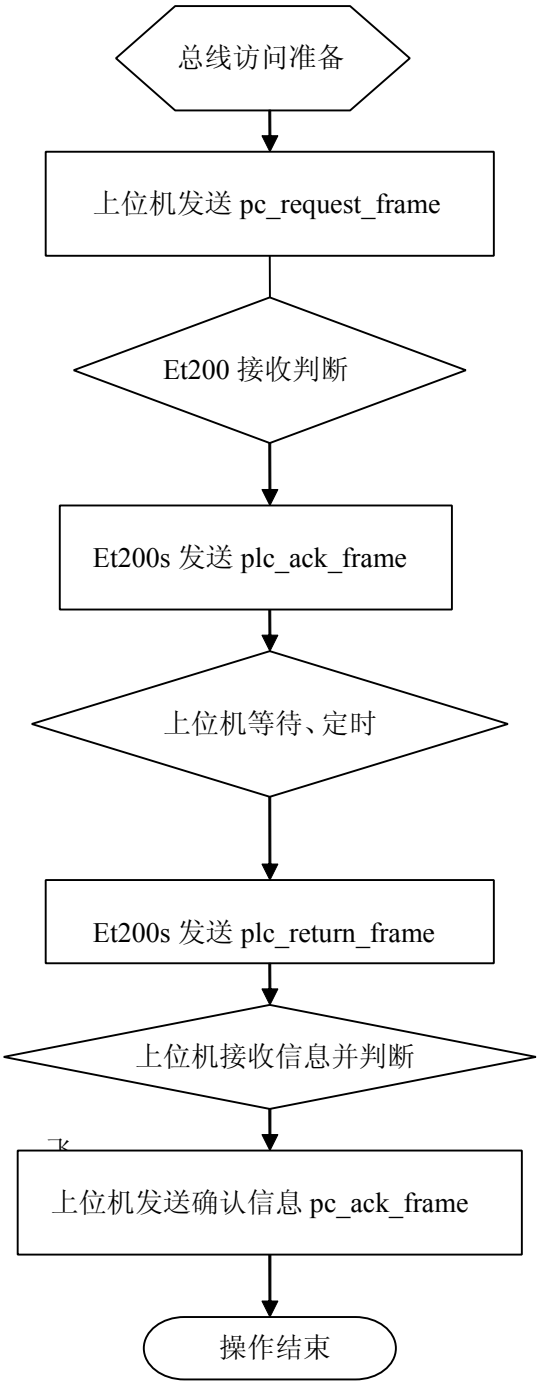
总线访问帧： SD DA SA FC FCS END

10 DA 00 49 FCS 16

SD、DC、FC、END 均占据一个字节长度，为固定数值，分别等于 10、 DC 49、16，FCS 采用求和校验，等于 DA+SA+FC。

(2)握手

上位机在收到 et200s 发出的令牌帧（dc 02 02）后以其令牌帧（dc 00 00）作为回复，等待 et200s 应答，如果收到 dc 00 02，pc 机立即回复 dc 02 00，令牌握手成功。总线访问握手方式与令牌握手一致。在读写操作过程中，应答握手也不间断。



读取操作

一次读操作的步骤包括上位机发出读命令帧（pc_request_frame_read），PLC 作出正确的响应，并将确认信息帧（plc_ack_frame_read）返回给上位机，接着反馈回正确的数据信息帧（plc_return_frame_read）给上位机，上位机接到此帧数据，校验确定后对 PLC 做出确认信息帧（pc_ack_frame_read），这样完成一个读取数据的过程。在读取操作过程中，上位机和 PLC 共进行两次应答。

读取命令

读取数据时上位机的请求帧格式如下，该帧占据 38 字节长度，记作 pc_request_frame_read(38)。

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	FS	UU	GU	DU	FCS	END

SD LE LER SD 占据 4 字节长度，为固定值 。

pc_request_frame_read(0)=68

pc_request_frame_read(1)= 1F，帧长度校验，为 DA+SA+FC+DSAP+SSAP+FS+GU+DU 的字节个数。

pc_request_frame_read(2)=1F 重复帧长度，与帧长度校验记法相同。

pc_request_frame_read(3)=68

DA SA FC DSAP SSAP FS 各占据 7 字节。

pc_request_frame_read(4)=82 数值上等于目标站地址加上 80

pc_request_frame_read(5)=80 数值上等于源站地址加上 80

pc_request_frame_read(6)=5C、7C

pc_request_frame_read(7)=16、15

pc_request_frame_read(8)=02、01

pc_request_frame_read(9)=F1 为分界符，其值不变。

pc_request_frame_read(10)=00~FF，帧序号，对相同操作时自加计数。在应答握手时用来判断当前应答帧是否为本请求的应答。

UU 占据 6 字节长度，均为固定值

pc_request_frame_read(11)=32

pc_request_frame_read(12)=01

pc_request_frame_read(13)=00

pc_request_frame_read(14)=00

pc_request_frame_read(15) =33

pc_request_frame_read(16)=02、01

GU 占据 6 字节长度，混合读写时可以进行操作

pc_request_frame_read(17)=00

pc_request_frame_read(18)=0E

pc_request_frame_read(19)=00

pc_request_frame_read(20)=00

pc_request_frame_read(21)=04

pc_request_frame_read(22)

单一读写时 pc_request_frame_read(22)=01，其他不变化；

混合读写时 pc_request_frame_read(22)为其他值。

DU 单元占据 12 字节长度从 pc_request_frame_read(23)到 pc_request_frame_read(26)这 4 字节为固定数值

pc_request_frame_read(23)=12

pc_request_frame_read(24)=0A

pc_request_frame_read(25)=10

pc_request_frame_read(26)=02

pc_request_frame_read(27) 和 pc_request_frame_read(28)这 2 字节共同表示读取的数据个数，当读取的存储区是 I、 Q、M 、DB 时表示字节个数，当存储区是 C、 T 时表示读取的计数器或定时器的个数。

如果读取两个字节，则为：

pc_request_frame_read(27)=00

pc_request_frame_read(28)=02

如果读取一个计数器或者定时器，则为：

pc_request_frame_read(27)=00

pc_request_frame_read(28)=01

pc_request_frame_read(29)、pc_request_frame_read(30)共同表示要操作的 DB 号，如果读取其他区，则二者分别为 00 00。

pc_request_frame_read(31)表示存储区类型，具体参考表 1-1

存储区	I	Q	M	DB	C	T
标示符	81	82	83	84	1C	1D

表 1-1

pc_request_frame_read(32)

pc_request_frame_read(33)

pc_request_frame_read(34)共同表示操作的起始地址，对于 I、 Q、 M 、 DB 存储区按照

bit 计算，对 C 、T 存储区按照其个数计算。

若读取 DB1B1 时则依次为

pc_request_frame_read(32)=00

pc_request_frame_read(33)=00

pc_request_frame_read(34)=08

若读取 C1 或 T1 时则为

pc_request_frame_read(32)=00

pc_request_frame_read(33)=00

pc_request_frame_read(34)=01

pc_request_frame_read（35）是帧校验码，采用和取余算法：

（DA+SA+FC+DSAP+SSAP+FC+UU+GU+DU ）mod 16#100

pc_request_frame_read(36)、pc_request_frame_read(37)是结束符，为固定值，分别等于 16 E5。

PLC 接收到请求命令（pc_request_frame_read）时，确认后返回一个数据帧表示回应，占据 15 字节长度，记作 plc_ack_frame_read(15)，格式如下：

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	UU	FS	FCS	END

SD LE LER SD

plc_ack_frame_read(0)=68

plc_ack_frame_read(1)=08

plc_ack_frame_read(2)=08

plc_ack_frame_read(3)=68

DA、SA

plc_ack_frame_read(4)=80

plc_ack_frame_read(5)=82

FC

plc_ack_frame_read(6)=7C 5C

DSAP、SSAP

plc_ack_frame_read(7)=02

plc_ack_frame_read(8)=16

UU

plc_ack_frame_read(9)=B0

plc_ack_frame_read(10)=01

FS

plc_ack_frame_read(11)

帧序号，和 pc_request_frame_read(10)保持一致。

FCS

plc_ack_frame_read(12)

帧校验,等于（DA+SA+FC+DSSAP+SSAP+UU+FS+FCS）mod 16#100

END

plc_ack_frame_read(13)=16

plc_ack_frame_read(14)=E5

在发送完响应数据帧（plc_ack_frame_read）后 PLC 接着给上位机反馈其所要读取的数据信息帧（plc_return_frame_read）,其长度因读取字节个数而长短不定，格式如下：

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	FS	UU	GU	DU	FCS	END

SD LE LER SD

plc_return_frame_read(0)=68

plc_return_frame_read(1)

plc_return_frame_read(2)

plc_return_frame_read(3)=68

DA SA FC DSAP SSAP FS 各占一个字节

plc_return_frame_read(4)=80

plc_return_frame_read(5)=82

plc_return_frame_read(6)=5C

plc_return_frame_read(7)=16

plc_return_frame_read(8)=02

plc_return_frame_read(9)分界符，为固定值 F1

plc_return_frame_read(10)，与 plc_ack_frame（11）保持一致。

UU 占据 8 个字节长度，均为固定值

plc_return_frame_read(11)=32

plc_return_frame_read(12)=03

plc_return_frame_read(13)=00

plc_return_frame_read(14)=00

plc_return_frame_read(15)=33

plc_return_frame_read(16)=02

plc_return_frame_read(17)=00

plc_return_frame_read(18)=02

GU 占据 6 字节长度

plc_return_frame_read(19)=00

plc_return_frame_read(20)

等于读取的自己个数加 4，如果读取一个字节时为 05

plc_return_frame_read(21)=00

plc_return_frame_read(22)=00

plc_return_frame_read(23)=04

plc_return_frame_read(24)

单一读取时为 01，分组读取时为其他值。

DU 占据（4+pc_request_frame_read(27)+ pc_request_frame_read(28)）字节长度

plc_return_frame_read(25)=FF

plc_return_frame_read(26)=04

plc_return_frame_read(27)、plc_return_frame_read(28)共同表示返回所读取的数据位数，按照 bit 计算；如果读取了一个字节，则他们分别为 00、08。

PLC 返回所要读取的数据，按照从低地址到高地址的顺序依次存放。

plc_return_frame_read(29)

plc_return_frame_read(30)

.

.

plc_return_frame_read(n)

n=28+读取的字节数

plc_return_frame_read(n+1)为 FCS,采用和取余校验。

END

plc_return_frame_read(n+2)=16

plc_return_frame_read(n+3)=E5

上位机在接收到数据经过确认以后向 PLC 发送一个确认帧（pc_ack_frame_read）,一共 15 字节长度，记作 pc_ack_frame_read（15）。

格式如下：

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	UU	FS	FCS	ED

SD　LE　LER　SD

pc_ack_frame_read（0）=68

pc_ack_frame_read（1）=08

pc_ack_frame_read（2）=08

pc_ack_frame_read（3）=68

DA　SA

pc_ack_frame_read（4）=82

pc_ack_frame_read（5）=80

FC

pc_ack_frame_read（6）=5C

DSAP　SSAP

pc_ack_frame_read（7）=16

pc_ack_frame_read（8）=02

UU

pc_ack_frame_read（9）=B0

pc_ack_frame_read（10）=07

FS

pc_ack_frame_read（11） 和 plc_return_frame(10) 保持一致

FCS

pc_ack_frame_read（12）

采用求和取余校验算法，等于（DA+SA+FC+DSSAP+SSAP+UU+FS）mod 16#100

END

pc_ack_frame_read（13）=16

pc_ack_frame_read（14）=E5

读数据过程完成。

写入操作

一次完整的写入操作步骤包括：首先上位机发出写命令信息帧（pc_request_frame_write），PLC 接收以后判断，若正确，则做出响应，并将确认信息（plc_ack_frame_write）帧返回给上位机，并反馈回正确的数据（plc_return_frame_write）帧给上位机，上位机接到此帧数据，校验正确后对 PLC 做出确认信息（pc_ack_frame_write），这样就完成一个读取数据的过程。在读取操作过程中，上位机和 PLC 共进行两次应答。

写入时上位机的请求帧（pc_request_frame_write）所占据字节长度不确定，跟写入的数据个数有关。记作 pc_request_frame_write（），格式如下：

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	FS	UU	GU	DU	FCS	END

SD　LE　LER　SD

pc_request_frame_write（0）=68

pc_request_frame_write（1）pc_request_frame_write（2）写入一个字节时为 24，写入两个字节时为 25…

pc_request_frame_write（3）=68

DA　SA　FC　DSAP　SSAP　FS

pc_request_frame_write（4）=82

pc_request_frame_write（5）=80

pc_request_frame_write（6）=5C

pc_request_frame_write（7）=16

pc_request_frame_write（8）=02

pc_request_frame_write（9）=F1

pc_request_frame_write（10）为帧序号，从 00 到 FF

UU 占据 6 字节长度

pc_request_frame_write（11）= 32

pc_request_frame_write（12）=01

pc_request_frame_write（13）=00

pc_request_frame_write（14）=00

pc_request_frame_write（15）=43

pc_request_frame_write（16）=02

GU 占据 6 字节长度

pc_request_frame_write（17）=00

pc_request_frame_write（18）=0E

pc_request_frame_write（19）、pc_request_frame_write（20）共同表示写入的字节个数加 4。如果要写入 2 个字节，则依次为 00 、06。

pc_request_frame_write（21）=05

pc_request_frame_write（22）=01

DU 占据的长度和要写入的字节个数有关，其长度为（16+ pc_request_frame_write（21）+pc_request_frame_write（22）- 4）个字节，置复位操作按照一个字节计算。

pc_request_frame_write（23）=12

pc_request_frame_write（24）=0A

pc_request_frame_write（25）=10

pc_request_frame_write（26）与存储区有关，当写 C 区时，为 1C, M 区置复位时为 01，写其他存储区为 02

pc_request_frame_write（27）=00

pc_request_frame_write（28）=01

pc_request_frame_write（29）、pc_request_frame_write（30）共同表示所要写入的 DB 号，写其他存储区时，为 00 00，注意 I 区和 T 区不能进行写操作。

pc_request_frame_write（31）表示存储区类型，参考表 1-2

存储区	Q	M	DB	C
标示符	82	83	84	1C

表 1-2

pc_request_frame_write（32）

pc_request_frame_write（33）

pc_request_frame_write（34）

以上三个字节表示要写入的起始地址。对于 C 区，则为起始计数器的编号，若对 C2 写入，

则依次为 00 00 02。对其他区写操作时则表示起始位地址，如果要写 DB1B1,则依次为 00、00、08。

pc_request_frame_write（35）、pc_request_frame_write（36）与存储区有关，写入 C 区时依次为 00　09，置复位操作时依次为 00 03，写入其他区（包括对 M 区写入字节）是依次为 00、04 。

pc_request_frame_write（37）、pc_request_frame_write（38）共同表示写入的数据量，与存储区类型有关。写 C 区时表示写入的字节个数，例如对一个计数器进行写操作，则其值依次为 00、02；写其他存储区区时表示要写入的数据位数，按照 bit 计算，若写 DB1B0,则依次为 00、08。

pc_request_frame_write（39）

pc_request_frame_write（40）

pc_request_frame_write（41）

.

.

pc_request_frame_write（n）

以上若干字节表示要写入的数据，按照低地址到高地址的顺序排列,其中对 C 区进行写操作时，数据表示为 BCD 码，对其他其他存储区写入时，数据均为十六进制格式表示。

n=38+ pc_request_frame_write（19）+pc_request_frame_write（20）-　4)。

pc_request_frame_write（n+1）为 FCS，采用求和取余算法，等于

（DA+SA+FC+DSAP+SSAP+FS+UU+GU+DU）mod 16#100

END

pc_request_frame_write（n+2）=16

pc_request_frame_write（n+3）=E5

PLC 在接收到请求数据帧确认后，返回确认信息帧（plc_ack_frame_write）

占据 15 字节长度，格式如下：

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	UU	FS	FCS	ED

SD　LE　LER　SD

plc_ack_frame_write（0）=68

plc_ack_frame_write（1）=08

plc_ack_frame_write（2）=08

plc_ack_frame_write（3）=68

DA　SA　FC　DSAP　SSAP

plc_ack_frame_write（4）=80

plc_ack_frame_write（5）=82

plc_ack_frame_write（6）=5C

plc_ack_frame_write（7）=02

plc_ack_frame_write（8）=16

UU

plc_ack_frame_write（9）=B0

plc_ack_frame_write（10）=01

FS

plc_ack_frame_write（11） 与 pc_request_frame_write（10）保持一致。

FCS

plc_ack_frame_write（12）,采用求和取余校验算法。

END

plc_ack_frame_write（13）=16

plc_ack_frame_write（14）=E5

plc_return_frame_write 帧格式

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	FS	UU	FCS	ED

记作 plc_return_frame_write（29）

SD　LE　LER　SD

plc_return_frame_write（0）=68

plc_return_frame_write（1）=16

plc_return_frame_write（2）=16

plc_return_frame_write（3）=68

DA　SA　FC　DSAP　SSAP　FS

plc_return_frame_write（4）=80

plc_return_frame_write（5）=82

plc_return_frame_write（6）=7C

plc_return_frame_write（7）=02

plc_return_frame_write（8）=16

plc_return_frame_write（9）=F1

plc_return_frame_write（10），FS，同 plc_ack_frame_write（11）保持一致。

UU 占据 15 字节长度

plc_return_frame_write（11）=32

plc_return_frame_write（12）=03

plc_return_frame_write（13）=00

plc_return_frame_write（14）=00

plc_return_frame_write（15）=43

plc_return_frame_write（16），和 pc_request_frame_write（16）保持一致。

plc_return_frame_write（17）=00

plc_return_frame_write（18）=02

plc_return_frame_write（19）=00

plc_return_frame_write（20）=01

plc_return_frame_write（21）=00

plc_return_frame_write（22）=00

plc_return_frame_write（23）=05

plc_return_frame_write（24）=01

plc_return_frame_write（25）=FF

FCS

plc_return_frame_write（26），采用求和取余校验算法。

END

plc_return_frame_write（27）=16

plc_return_frame_write（28）=E5

pc_ack_frame_write 帧格式

SD	LE	LER	SD	DA	SA	FC	DASP	SSAP	UU	FS	FCS	ED

记作 pc_ack_frame_write（15）

SD　LE　LER　SD

pc_ack_frame_write（0）=68

pc_ack_frame_write（1）=08

pc_ack_frame_write（2）=08

pc_ack_frame_write（3）=68

DA SA FC DSAP SSAP

pc_ack_frame_write（4）=82

pc_ack_frame_write（5）=80

pc_ack_frame_write（6）=7C

pc_ack_frame_write（7）=16

pc_ack_frame_write（8）=02

UU

pc_ack_frame_write（9）=B0

pc_ack_frame_write（10）=07

FS

pc_ack_frame_write（11）plc_return_frame_write（29）保持一致。

FCS

pc_ack_frame_write（12）采用求和取余校验算法。

END

pc_ack_frame_write（13）=16

pc_ack_frame_write（14）=E5

写数据过程完成。

对于读写数据帧的 FC 以及 DSAP 和 SSAP 做以说明

	FC		DSAP		SSAP	
Pc_request_frame_read	7C	5C	15	01	16	02
Plc_ack_read	7C	5C	01	15	02	16
Plc_return_read	5C	7C	01	15	02	16
Pc_ack_read	5C	7C	15	01	16	02

在解析读写数据报文以后，可以撤掉 cp5613 通讯卡，以验证其正确性

测试源码见附录 F。

结束语

本文运用串口监视的方法，通过简单易行的操作解析出 siemens MPI 协议的报文格式，其结果具有很大的使用价值(1)使用户不用购买西门子专用的通讯处理卡就可以让上位机和 PLC 的通信，从而实现所需控制功能，节约成本。(2) 降低了用户自主开发的难度，使通讯编程变得简单明了，无需购买软件和授权等就可以使用 MPI 协议监控 PLC 的工作状态。

参考文献

STEP 7V5.4 编程手册

Siemens s7300 系列硬件手册

Visual C++串口编程实践

http://www.serial-port-communication.com/serial-monitor/

http://www.csdn.net/

http://www.ad.siemens.com.cn/