

USS 协议应用基本概念

Zane

USS 协议简介

USS 协议 (Universal Serial Interface Protocol 通用串行接口协议) 是 SIEMENS 公司所有传动产品的通用通讯协议, 它是一种基于串行总线进行数据通讯的协议。USS 协议是主-从结构的协议, 规定了在 USS 总线上可以有一个主站和最多 30 个从站; 总线上的每个从站都有一个站地址 (在从站参数中设定), 主站依靠它识别每个从站; 每个从站也只对主站发来的报文做出响应并回送报文, 从站之间不能直接进行数据通讯。另外, 还有一种广播通讯方式, 主站可以同时给所有从站发送报文, 从站在接收到报文并做出相应的响应后可不回送报文。

使用 USS 协议的优点

1. 对硬件设备要求低, 减少了设备之间的布线,
2. 无需重新连线就可以改变控制功能,
3. 可通过串行接口设置或改变传动装置的参数,
4. 可实时的监控传动系统

常用 USS 主站的性能对比

产品	通讯接口	最大通讯波特率
CPU 22X	9 芯 D 型插头	115.2K bps
CPU 31XC-PTP	15 芯 D 型插头	19.2K bps
CP 340-C	15 芯 D 型插头	9.6K bps
CP 341-C	15 芯 D 型插头	19.2K bps

可见,S7-200 CPU22X 具有较高的性能价格比。

USS 从站性能对比

产品	PKW 区	PZD 区	Bico	终端电阻	通讯接口	最大通讯波特率
MM3/ECO	3 固定	2 固定	NO	NO	9 芯 D 型插头或端子	19.2K bps
MM410/420	0, 3, 4, 127	0-4	YES	NO	端子	57.6K bps
MM430/440	0, 3, 4, 127	0-8	YES	NO	端子	115.2K bps
Simoreg 6RA70	0, 3, 4, 127	0-16	YES	YES	9 芯 D 型插头或端子	115.2K bps
Simovert 6SE70	0, 3, 4, 127	0-16	YES	YES	9 芯 D 型插头或端子	115.2K bps

USS 通讯硬件连接

1. 条件许可的情况下，USS 主站尽量选用直流型的 CPU（针对 S7-200 系列）
2. 一般情况下，USS 通讯电缆采用双绞线即可（如常用的以太网电缆），如果干扰比较大，可采用屏蔽双绞线。
3. 在采用屏蔽双绞线作为通讯电缆时，把具有不同电位参考点的设备互连会在互连电缆中产生不应有的电流，从而造成通讯口的损坏。要确保通讯电缆连接的所有设备，或是共用一个公共电路参考点，或是相互隔离的，以防止不应有的电流产生。屏蔽线必须连接到机箱接地点或 9 针连接的插针 1。建议将传动装置上的 0V 端子连接到机箱接地点
4. 尽量采用较高的波特率，通讯速率只与通讯距离有关，与干扰没有直接关系。
5. 终端电阻的作用是用来防止信号反射的，并不用来抗干扰。如果在通讯距离很近，波特率较低或点对点的通讯的情况下，可不用终端电阻。多点通讯的情况下，一般也只需在 USS 主站上加终端电阻就可以取得较好的通讯效果。
6. 当使用交流型的 CPU22X 和单相变频器进行 USS 通讯时，CPU22X 和变频器的电源必须接成同相位的。
7. 建议使用 CPU226(或 CPU224+EM277)来调试 USS 通讯程序。
8. 不要带电插拔 USS 通讯电缆，尤其是正在通讯过程中，这样极易损坏传动装置和 PLC 的通讯端口。如果使用大功传动装置，即使传动装置掉电后，也要等几分钟，让电容放电后，再去插拔通讯电缆。

USS 通讯的编程

USS 协议是以字符信息为基本单元的协议，而 CPU22X 的自由口通讯功能和 CPU31XC-PTP 的 RS422/485 串行口正好也是以 ASCII 码的形式来发送接收信息的。利用这些 CPU 的 RS485 串行口的通讯功能，由用户程序完成 USS 协议功能，可实现与 SIEMENS 传动装置简单而可靠的通讯连接。

1. USS 点对点通讯的编程要点:

- a) USS 主站（PLC）与 USS 从站（传动装置）之间的通讯是异步方式的，负责与传动装置通讯的工作程序应采用后台工作方式，如何发送接收数据应与控制逻辑无关。用户程序通过改变 USS 报文中的 STW 及 HSW 的值，来控制变频器的启停及改变设定频率值。
- b) 利用发送指令（如 XMT, P_SEND, P_SND_RK）发送 USS 报文至传动装置，利用接收指令（如 RCV, P_RCV, P_RCV_RK）接收变频器返回的 USS 报文。同一时刻，只能有一个发送指令或接收指令被激活。
- c) USS 通讯程序包括通讯端口初始化子程序、BCC 校验码计算子程序、数据发送子程序、数据接收子程序、通讯超时响应子程序、通讯流程控制子程序等。可采用中断响应的方式，也可用查询相应标志位的方式来实现。
- d) 设立发送接收数据缓存区与映像区，用户应通过改变映像区的 USS 发送报文值来控制传动装置，或通过读取映像区 USS 接收报文中的状态值来判断

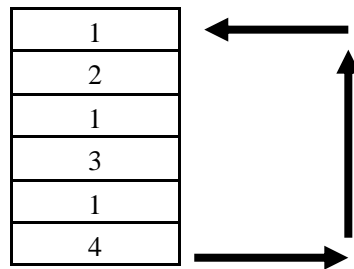
传动装置的当前状态。以防止因干扰而接收到错误数据而使 PLC 做出错误的判断和控制

2. USS 多点通讯的编程要点

- a) 控制通讯的基本流程同上述点对点通讯方式
- b) 对各从站的控制应采取轮询方式，轮询程序同样也是后台工作方式工作的。
- c) 根据对各台传动装置控制任务的轻重，在 PLC 数据区内建立一个从站地址表，按该地址表轮询各传动装置。采用间接寻址的编程方式，可大大节省 CPU 的程序空间。
- d) 轮询地址表示例

虽然，USS 协议的实际物理地址只有 30 个，但轮询地址表的大小无限制，其有效站地址可以在表中根据实际应用需要反复出现。实际轮询站点数越多，其轮询的间隔时间也越大，而表中站地址重复次数越多，其轮询的间隔时间越小，因此必须为每个传动装置设定适当的通讯超时时间以适应这种轮询间隔。

USS 从站轮寻地址表



- e) 不同 USS 从站可以有不同的 USS 报文结构，如 3 PKW + 2 PZD；4 PKW + 4 PZD；0 PKW + 6 PZD 等组合。但整个系统要支持广播方式，则 USS 网络中的所有从站都必须有相同的 PKW 区才行。
- f) 传动装置对以广播方式发送的指令做出响应后，不再回送报文，因此 PLC 可以不再进入数据接收状态。

附 S7-22X USS 点对点通讯演示程序。

S7-22X USS点对点通讯演示程序

声明：

- 本程序为S7-22X USS点对点通讯演示程序，适用于所有支持USS协议的西门子驱动装置。
- 本演示程序只支持USS固定报文格式，无参数读写功能。
- 本程序可以无偿使用于实验及教学目的，对于实际的工程应用，作者不承担任何义务与责任。
- 作者：ZANE

保留资源：

- 通讯口PORT 0
- VB100 -- VB400
- SB0
- T32, T37, T96
- SBR0, SBR1, SBR2, INTO, INT1, INT2
- 其余的系统资源可自由使用

MM4XX变频器参数设置：

- P700 = 5
- P1000 = 5
- P2010[0] = 8
- P2011[0] = 0
- P2012[0] = 2
- P2013[0] = 4
- P2014[0] = 100

程序清单：

```
ORGANIZATION_BLOCK MAIN:OB1  
TITLE=USS通讯演示程序 VER2.0
```

```
BEGIN
```

```
Network 1
```

```
LD SM0.1
```

```
CALL PORT0_INIT
```

```
Network 2
```

```
LD SM0.0
```

```
CALL COM_PROCEDURE
```

```
END_ORGANIZATION_BLOCK
```

```
SUBROUTINE_BLOCK PORT0_INIT:SBRO
```

```
TITLE=初始化子程序
```

```
BEGIN
```

```
Network 1
```

```
LD SM0.0
```

```
MOVB VB106, SMB30
```

```
BMB    VB100, SMB88, 6
ENI
MOVB   16#01, SBO
END_SUBROUTINE_BLOCK

SUBROUTINE_BLOCK BCC_CAL:SBR1
TITLE= BCC校验码计算子程序
VAR_INPUT
DAT_ADR_P:DWORD;           // Address pointer to data to be calculated
DAT_LEN:BYTE;              // Data length for BCC cal
S_R:BYTE;                  // S=Send Cal  R=RCV Cal
END_VAR
VAR_OUTPUT
BCC_AP:DWORD;              // Address pointer for BCC calculation result
END_VAR
VAR
BCC_CHK:BYTE;              // BCC value buffer
INDEX:INT;
FINAL:INT;
ADDR_P:DWORD;
END_VAR
BEGIN
Network 1
LD     SM0.0
MOVB  0, LB10
MOVD  LD0, LD15
BTI   LB4, LW13
Network 2
LDB=  LB5, 'S'
-I    +1, LW13
Network 3
LDB<> LB5, 'S'
AB<>  LB5, 'R'
CRET
Network 4
LD     SM0.0
FOR    LW11, +1, LW13
XORB  *LD15, LB10
INCD  LD15
Network 5
NEXT
Network 6
LD     SM0.0
MOVB  LB10, *LD15
```

```
MOVD LD15, LD6
END_SUBROUTINE_BLOCK

SUBROUTINE_BLOCK COM_PROCEDURE:SBR2
TITLE=USS通讯处理子程序
VAR
TT:DWORD;
END_VAR
BEGIN
Network 1 // 系统上电延时
LSCR S0.0
Network 2
LD S0.0
TON T37, VW110
Network 3
LD T37
SCRT S0.1
Network 4
SCRE
Network 5 // USS报文准备及计算校验码
LSCR S0.1
Network 6
LD S0.1
MOVB VB300, VB200
BMB VB301, VB201, VB200
CALL BCC_CAL, VD140, VB200, 'S', VD148
ATCH COM_TIMEOUT, 21
S S0.3, 1
SCRT S0.2
Network 7
SCRE
Network 8 // USS报文发送
LSCR S0.2
Network 9
LD SM4.5
XMT VB200, 0
ATCH XMT_COMPLETED, 9
ATCH COM_TIMEOUT, 21
SCRT S0.4
Network 10
SCRE
Network 11 // USS通讯超时检测
LSCR S0.3
Network 12
```

```
LD    S0.3
TON   T32, VW114
Network 13
SCRE
Network 14
LD    S0.4
R     S0.4, 1
Network 15 // USS接收报文校验
LSCR  S0.5
Network 16
LD    SM0.0
CALL  BCC_CAL, VD144, VB250, 'R', VD148
MOVD  VD200, LDO
XORD  VD250, LDO
Network 17
LDB=  *VD148, 0
AD=   LDO, +0
LPS
NOT
INCB  VB119
LRD
MOVB  VB250, VB350
LRD
BMB   VB251, VB351, VB250
LPP
MOVB  0, VB119
Network 18
LDB>  VB119, 200
MOVB  200, VB119
Network 19
LD    S0.5
SCRT  S0.6
Network 20
SCRE
Network 21 // 二次通讯之间的延时间隔
LSCR  S0.6
Network 22
LD    S0.6
TON   T96, VW112
Network 23
LD    T96
SCRT  S0.1
Network 24
SCRE
```

END_SUBROUTINE_BLOCK

INTERRUPT_BLOCK XMT_COMPLETED: INTO

TITLE= USS报文发送完成中断子程序

BEGIN

Network 1

LD SM0.0

DTCH 9

MOVB 16#CC, SMB87

MOVB VB200, SMB94

RCV VB250, 0

ATCH RCV_COMPLETED, 23

Network 2

LDN V118.0

= V118.0

END_INTERRUPT_BLOCK

INTERRUPT_BLOCK RCV_COMPLETED: INT1

TITLE= USS报文接收完成中断子程序

BEGIN

Network 1

LD SM0.0

DTCH 21

DTCH 23

MOVW T32, VW116

R T32, 1

R S0.3, 1

S S0.5, 1

Network 2

LDN V118.1

= V118.1

END_INTERRUPT_BLOCK

INTERRUPT_BLOCK COM_TIMEOUT: INT2

TITLE= USS通讯超时中断子程序

BEGIN

Network 1

LD SM0.0

DTCH 9

DTCH 21

DTCH 23

MOVB 16#00, SMB87

RCV VB250, 0

INCB VB119


```
R    T32, 1
R    S0.3, 1
S    S0.6, 1
Network 2
LDB>  VB119, 200
MOVB  200, VB119
Network 3
LDN   V118.2
=     V118.2
END_INTERRUPT_BLOCK

DATA BLOCK
//COMM PORT INITIAL (Default for Port0)
VB100 16#02      //-->SMB88
VB101 16#00      //-->SMB89
VW102 16#0000    //-->SMW90
VW104 16#0064    //-->SMW92
VB106 16#41      //-->SMB30

//USS PROCEDURE PARAMETER
VW110 30          //START DELAY      T37
VW112 5           //INTER DELAY      T96
VW114 100        //COM TIME OUT    T32
VW116 0          //READ T32
VB118 16#00      //STATUS FLAGS
VB119 0          //ERR_CNT_COM

//ADDR_POINTER
VD140 16#080000C9 //ADDRESS POINTER FOR SEND BUFFER MESSAGE START BYTE
VD144 16#080000FB //ADDRESS POINTER FOR RCV BUFFER MESSGAE START BYTE
VD148 16#00000000 //ADDRESS POINTER FOR BCC CALCULATION RESULT REGISTER

//***USS SEND BUFFER***
VB200 16#00      //XMT_LEN
VB201 16#00      //STX
VB202 16#00      //LGE
VB203 16#00      //ADR
//USS NET DATA FOR SEND
//BCC

//***USS RCV BUFFER***
VB250 16#00      //RCV_LEN
VB251 16#00      //STX
VB252 16#00      //LGE
```

```
VB253 16#00 //ADR
//RECEIVED USS NET DATA
//RECEIVED BCC
//RCV BCC CHECK BYTE

//SLAVE
//***XMT_MESSAGE*** (Data structure is based on different USS node)
VB300 16#10 //XMT_LEN
VB301 16#02 //STX
VB302 16#0E //LGE
VB303 16#00 //ADR
//**PKW AREA**
VW304 16#0000 //PKE
VW306 16#0000 //IND
VW308 16#0000 //VAL PWE1
VW310 16#0000 //VAL PWE2
//**PZD AREA**
VW312 16#047E //STW PZD1
VW314 16#2000 //HSW PZD2
//**BCC**
VB316 16#00 //BCC
//***RCV_MESSAGE***
VB350 16#00 //RCV_LEN
VB351 16#00 //STX
VB352 16#00 //LGE
VB353 16#00 //ADR
//**PKW AREA**
VW354 16#0000 //PKE
VW356 16#0000 //IND
VW358 16#0000 //VAL PWE1
VW360 16#0000 //VAL PWE2
//**PZD AREA**
VW362 16#0000 //ZSW PZD1
VW364 16#0000 //HIW PZD2
//**BCC**
VB366 16#00 //BCC
```