

EINFUNC 驱动手册 3S V5.0 SIEMENS S7-200 SMART

V5.0_20170707

1. 35 通讯技术描述

■ 3S 通讯目的

3S 通讯技术是指将 PLC 程序中任意指定的操作数如定时器、计数器、内部变量数据和常量数据,以规定的数据格式如整数、双整数据、HEX/BCD 码,通过 PLC 直流固态输出点组成的 3S 通讯接口串行传送至带有 3S 通讯接口的外部设备上,使这些设备对接收到的数据进行进一步的处理,如图 1-0-1 所示。这项技术由任志兵于 1995 年发明研制,在当年申请并实施了实用新型专利:ZL95226265.7。在图 1-0-1 中:

-U1: 可编程序控制器 PLC, 数字输出点(DO)类型为晶体管型或场效应管型;

-B2: 可编程序控制器 PLC 的输出驱动电源, DC24V;

-H1: 带有 3S 通讯接口的外部设备,它需要接收整数,16Bits;

-H2: 带有 3S 通讯接口的外部设备,它需要接收整数,16Bits;

-H3: 带有 3S 通讯接口的外部设备,它需要接收双整数,32Bits。

■ 3S 基本连接

任选 PLC-U1 三个晶体管输出点作为最基本的 3S 通讯接口,如图 1-0-1 中所示的 Q8.0、Q8.1 和 Q8.2 点:

定义 Q8.0 为"帧同步时钟" 简称: **并行时钟 RCK** 定义 Q8.1 为"串行驱动时钟" 简称: **串行时钟 SCK** 定义 Q8.2 为"数据输出口" 简称: **数据出口 DATA**

将这三点分别与带有 3S 通讯接口的外部设备-H1 的 RCK、SCK 和 D00 端子相连接,再分别将 PLC 的公共端 COM 和外部设备的公共端 COMD 与 DC24V 电源-B2 相连即告完成。这种基本连接方式适合于传送很少量的数据,如 PLC 驱动一台五位数字显示器,只有一个数据通道,每帧数据传送一个字(16Bits)的整数。

■ 3S 增量连接

在最基本的连接基础上只需增加数据输出口 Q8.3、Q8.4、Q8.5……,各个数据通道在时钟的驱动下同时向外部设备的数据输入端 D00、D01、D02……传送数据,每个通道每帧传送一个字(16Bits)的数据码。这样能够保证传送速度不变。如图 1-0-1 中,-H3 外设需要接收双整数据(32Bits),PLC-U1 需要两个数据输出口,一个传送双整数的低字,另一个传送双整数的高字。

■ 3S 扩展连接

扩展型连接是 PLC-U1 同时驱动多台 3S 接口外部设备。在 PLC 侧也是只需增加数据输出口 Q8.3、Q8.4、Q8.5……,将数据分别送至对应的外部设备;而各台外部设备的 RCK 和 SCK 分别并联起来,共用时钟。

以上三种连接方式本质上都是一样的,无论有多少个数据通道,都是在时钟的驱动下同时向外部设备发送数据;在每台3S外设都只需要单通道数据的前提下,n台设备只占用PLC的n+2个直流固态DO点。

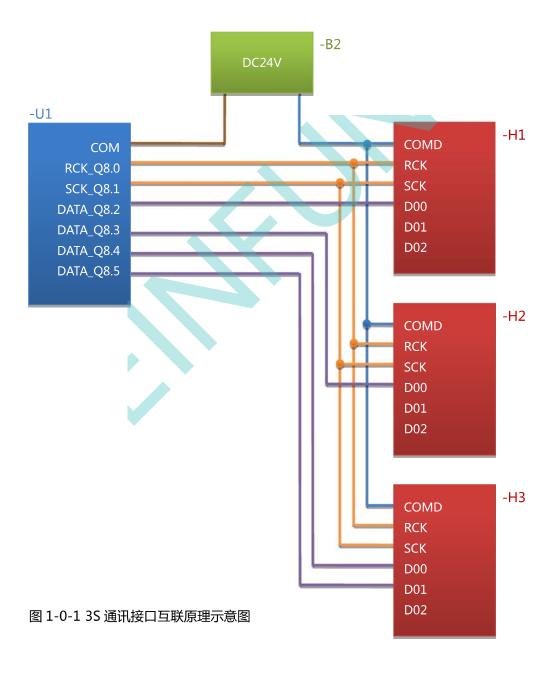
■ 3S 驱动程序

3S 驱动程序不是安装在 PLC 的软件底层上的固件,而是由 PLC 的用户程序来完成,非常简单,驱动单通道数据只需 5 个梯级共 12 个指令就可完成;驱动多通道数据并不会成倍增加梯级和指令步数。请参考后面章节有



关 **3S** 示范驱动程序的详尽描述。在驱动程序中,各个要被传送的数据字由 MOV 指令来捕捉,并行传送到各个对应的字串行移位通道上,字串行移位通道的最高位 MSB 连接指定的数据输出口(晶体管输出点),在串行时钟 SCK 的作用下 SHL 左移位指令驱动数据由低向高移位,从数据输出口逐步移位出去;被传送的数据无论是什么数据类型,总是高位在前,低位在后,由低向高方向串行移位;SCK 每发出 16 个脉冲就完成每个通道一帧数据的传送,每帧数据宽度为 16Bits。

通过后面章节有关 **3S** 示范驱动程序梯形图结构,不难看出这种驱动程序只须五个梯级步骤 "并行时钟、数据 **移位、串行时钟、数据出口、时钟时序**"。为便于记忆,我们把它编成五字口诀,任何品牌的 PLC 都可以按照 五字口诀结构编制 **3S** 驱动程序,前后顺序千万不能排错:**3S_V5.0**: 并、移、串、出、时





■ 3S 通讯电源

3S 通讯线路电源就是图 1-0-1 中的-B2 电源, DC24V。它与各设备的 COM/COMD 的连接方向取决于 PLC-U1 的 DO 晶体管/场效应管的有效电流方向:

表 1-1

| PLC 的 DO 晶体管/场效应管有效电流方向 | PLC DO 的 COM 端 | 3S 接口设备的 COMD 端 |
|-------------------------|--------------------|-----------------|
| 源电流输出,即所谓"高电平有效" | 1L+: +24V , 1M: 0V | 接 DC0V |
| 漏电流输出,即所谓"低电平有效" | 1L+: +24V , 1M: 0V | 接 DC24V |

- ❖ DC24V 电源必须是与强电完全隔离的开关型电源,不可使用线性电源。
- ❖ 尽量做到电源-B2 是独立的 DC24V 电源和独立的 COM 区,即不与其他感性直流负载共用,则能大大提高 3S 通讯的抗干扰能力。同一 COM 区内的其他晶体管输出点最好只接阻性负载,如 DC24V 指示灯;如果接了直流电磁线圈,则必须在直流线圈两端并联反向二极管释能元件,以防止 PLC 和 3S 外设长期受电应力的损害。
- ❖ 推荐按标准电源配置接线,这符合工业通讯技术规范,即相互通讯的设备之间在电气上是相互隔离的,适合长距离通讯,并且抗干扰能力也最强。如果用户出于成本方面的考虑,也可以按图 2-1-1 那样的简洁电源配置接线。

■ 3S 通讯导线

- ◆ 布线长度≤150m 时,可用普通电线,但仍建议3S信号线穿金属软管、硬管或金属线槽敷设;
- ❖ 布线长度 > 150m 时,推荐用绝缘屏蔽线;屏蔽线的屏蔽层应单点接 EE 或 DC24V 电源的 0V 端;
- ❖ 最长布线距离 300m, 这时可能要加宽 3S 脉冲宽度;
- ❖ 无论何种导线类型 都要避免 3S 外设的电源线和信号线与强电动力线近距离平行布线或穿于同一管槽内。

■ 3S 通讯速度

PLC 每运行 32n 个扫描周期,3S 外设才能刷新一次接收数据结果。3S 通讯速度的快慢与 PLC 的扫描周期 T 和 3S 脉冲宽度 n 有关,n 为 ≥ 1 的正整数,即后面章节提到的 3S 脉冲宽度设定值 SV_3S ,是 PLC 扫描周期的整倍数。下例公式可计算 3S 通讯每秒钟传送的帧数:f 单位:帧/s,T 单位:ms

 $f = \frac{1000}{32nT}$

■ 3S 脉冲宽度

- 3S 外设连接在 PLC 本地主机架的 DO 点和本地扩展 I/O 模块的 DO 点时, 3S 脉冲宽度应大于 PLC 晶体管输出点响应 ON/OFF 的最大时间的两倍以上;
- 3S 外设连接在主从模式现场总线(如 ProfiBus-DP等)的从站节点 I/O 模块的 DO 点上时,3S 脉冲宽度应大于现场总线循环时间的两倍以上。

这两个条件简称为"双时条件",这一点很重要。

■ 3S 时钟负荷

- 若 RCK 和 SCK 所用的晶体管输出点额定负载能力为 0.3A,则它们可驱动最多 38 个 3S 外设;
- 🌣 若 RCK 和 SCK 所用的晶体管输出点额定负载能力为 0.5A , 则它们可驱动最多 69 个 3S 外设。

■ 3S 电磁环境

完全同 PLC 所需电磁环境。PLC 受得了, 3S 外设也受得了。



■ 3S 搭载总线

3S 通讯脉冲完全可以搭载至工业以太网和现场总线上,以实现更远和更广泛的数据传输,比如将 3S 外设接在现场总线 ProfiBus-DP 从站 I/O 模块上,但要注意可能需要调整 3S 脉冲宽度。

这样的搭车技术,早已在 ProfiNet、ProfiBus、Modbus、ControlNet 和 CC-Link 等网络总线上得到证实和应用。

■ 3S 优点缺点

- 优点:3S驱动程序开源,不受串口通讯协议限制,支持任何品牌的PLC、PAC和IPC等,还可搭车工业网络;
- 优点:3S通讯数据线与3S外设一对一,便于故障分析和排查。而RS485总线遇某个劣质节点硬件故障时,不好排查故障,整个网络都受影响;
- ❖ 优点:3S通讯导线在一定范围内无特殊要求,可使用普通电线;
- . 优点:3S通讯技术占用 PLC 硬件资源非常少,比传统的数据并行传输或并行锁存传输方式少得多;
- ❖ 优点:3S 通讯技术可靠性很高,遇到干扰后恢复正确数据的时间只需要 PLC 的 32 个扫描周期,比两线传输方式恢复的快;
- 缺点:3S通讯技术与RS485双绞线通讯技术相比,随着3S外设数量增加,占用PLC硬件资源亦随之递增:
- ❖ 缺点:3S 通讯距离与 RS485 双绞线通讯距离相比,在不增加中继器的前提下还是较短(300m max)。

2. 3S_V5.0 驱动示例

3S 驱动程序不是安装在 PLC 的软件底层上的固件, 而是由 PLC 的用户程序来完成。

3S_V5.0 由爱羽方科技在 2017 年 01 月编制完成,通过了工业现场的应用测试,并向下兼容。比 V2.0~V4.0 版的驱动程序,V5.0 版代码更加优化,更加简洁,更加易读、易懂和易写了。



说明:本文档全面描述了 3S_V5.0 驱动技术在 SIEMENS 公司生产的可编程控制器 S7-200 SMART 上的应用, S7-200 SMART 上的程序结构原理同样适用于 SIEMENS 公司其他系列的 PLC。



1) S7-200 SMART 驱动一台 3S 外设_单数据线

■ 项目任务

将 S7-200 SMART 程序中内部变量 DATA_H1 的有符号整数, 经 3S 接口传送给 3MD3-A15D2 03H 数字显示器,以十进制数形式将数据(-32768~+32767)毫无偏差地显示出来。见图 2-1-1。

■ 硬件连接

表 2-1-1 硬件明细表

| 硬件符号 | 硬件名称 | 硬件规格型号 | 说明 |
|------|----------|---------------------|-----------------------------|
| 111 | 可编程序控制器 | 6ES7 288-1SR20-0AA0 | PLC CPUs |
| -U1 | 扩展输入输出模块 | 6ES7 288-2DT16-0AA0 | 8 DI_DC24V / 8 DO_晶体管,源电流输出 |
| -B2 | 开关电源 | 6EP1331-5BA00 | PLC 输出电源, DC24V 0.6A |
| -H1 | 数字显示器 | 3MD3-A15D2 03H | 单行五位数码,DC24V,3S通讯接口,16Bits |

本示例中的电源配置为"简洁配置"。

■ 程序代码

项目名称: SIEMENS_3S_V5.0_S7-200 SMART_0

软件名称: SIMATIC STEP7-MicroWIN SMART V2.3 国际标

识

本驱动代码用"驱动一台数显+按键操作"来示例,按键操作子程序 SBR_BUTT 不是必须的。

■ 扩宽脉宽

当 CPU 最短扫描周期时间较短,使得 3S 脉宽不能满足第一章所述 "双时条件"时,数据将不能完整传输,这时应人工扩宽脉冲宽度。

参见子程序块 SBR_DRV1 的网络 1~2,3S 脉宽当前值 PV_3S 累计 CPU 的扫描周期的次数,当它小于 3S 脉宽设定值时,退出子程序;只有当它大于等于 3S 脉宽设定值时,自身清零,并进入 3S 驱动网络。这样做可在不改变 CPU 扫描周期的前提下,将 3S 脉宽调整为 CPU 扫描周期的整倍数,增加 3S 脉宽,保证数据完整传送。

3S 脉宽设定值取值 1~2+,示例为 3,0 和 1 不加宽,2 表示 3S 脉宽为 CPU 扫描周期的两倍,3 表示三倍……以此类推。从 2 开始往上调整,直至数字显示器显示数值正确为止。





■ 使用按键

3MD3-A15D2 03H 是带按键的数字显示器,由于这些按键在电气上是独立于数字显示器的按钮开关,即直接键,因此可以任意定义这些按键的功能。我们以切换发送 S7-200 SMART 控制的某个温度当前值和修改设定值为例,来说明按键的用法。

■ 按键连接

见图 2-1-2。3S 通讯连接的线路与图 2-1-1 是相同的,只是采用了标准电源配置;四个按键 F1、F2、DN 和UP 的输出接点分别接至 S7-200 SMART 的 I8.0、I8.1、I8.2 和 I8.3 输入点上。

表 2-1-2 硬件明细表

| | - | | |
|------|----------|---------------------|-----------------------------|
| 硬件符号 | 硬件名称 | 硬件规格型号 | 说明 |
| 111 | 可编程序控制器 | 6ES7 288-1SR20-0AA0 | PLC CPUs |
| -U1 | 扩展输入输出模块 | 6ES7 288-2DT16-0AA0 | 8 DI_DC24V / 8 DO_晶体管,源电流输出 |
| -B2 | 开关电源 | 6EP1331-5BA00 | PLC 输出电源, DC24V 0.6A |
| -B3 | 开关电源 | 6EP1331-5BA00 | PLC 输入电源, DC24V 0.6A |
| -B4 | 开关电源 | 6EP1331-5BA00 | 3S 外设工作电源, DC24V 0.6A |
| -H1 | 数字显示器 | 3MD3-A15D2 03H | 单行五位数码,DC24V,3S通讯接口,16Bits |

本示例中的电源配置为"标准配置"。

■ 修改数值

- ❖ 温度当前值存于 PV_H1,温度设定值存于 SV_H1,显示设定范围:-327.68~327.67。
- ❖ 小数点由数字显示器上的 M 键和 S 键人工设定。
- ❖ 修改设定值思路:用按键操作INC和DEC指令来增减SV_H1数值,以达到修改温度设定值的目的。

■ 按键操作

不按任何键, 3MD3-A15D2 03H 数字显示器显示温度当前值; 只按 F1 设定键, 数字显示器显示温度设定值;

按住 F1 设定键,同时点动▲键或▼键可细调温度设定值;

按住 F1 设定键,同时按住▲键或▼键不放,两秒钟以后温度设定值呈较快速度变化;

同时按住 F1 设定键和 F2 加速键,再同时按住▲键或▼键,温度设定值则以最快速度变化。

■ 程序代码

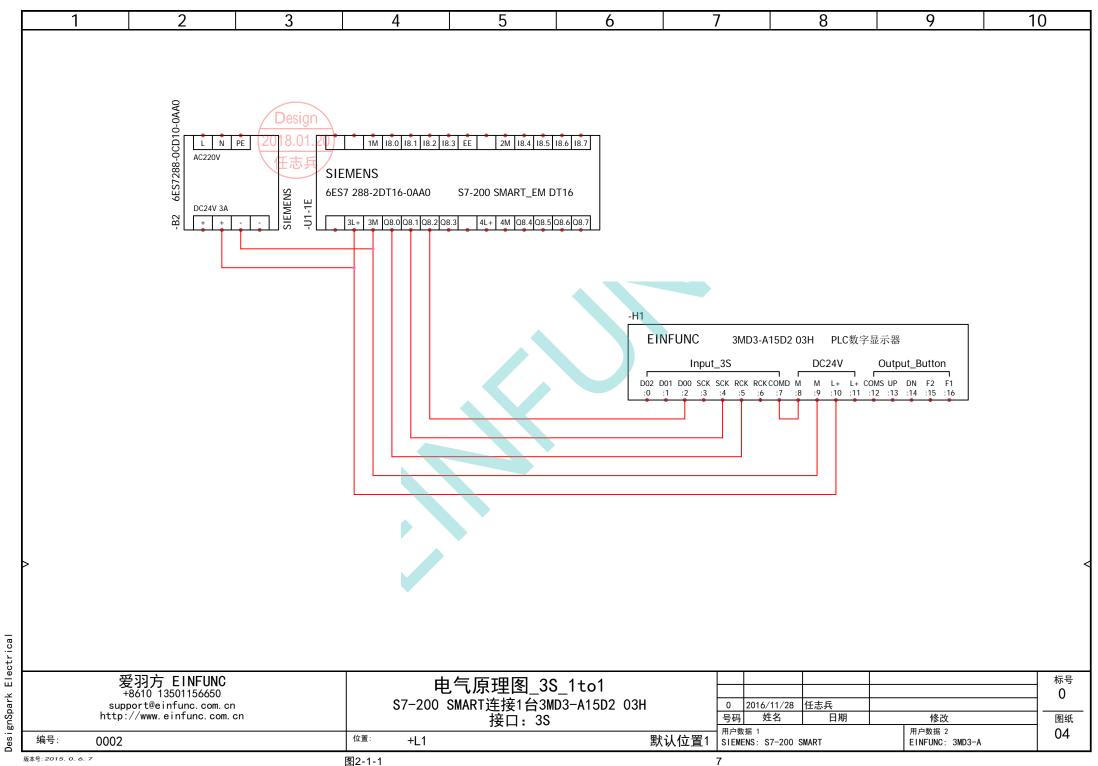
项目名称: SIEMENS_3S_V5.0_S7-200 SMART_0

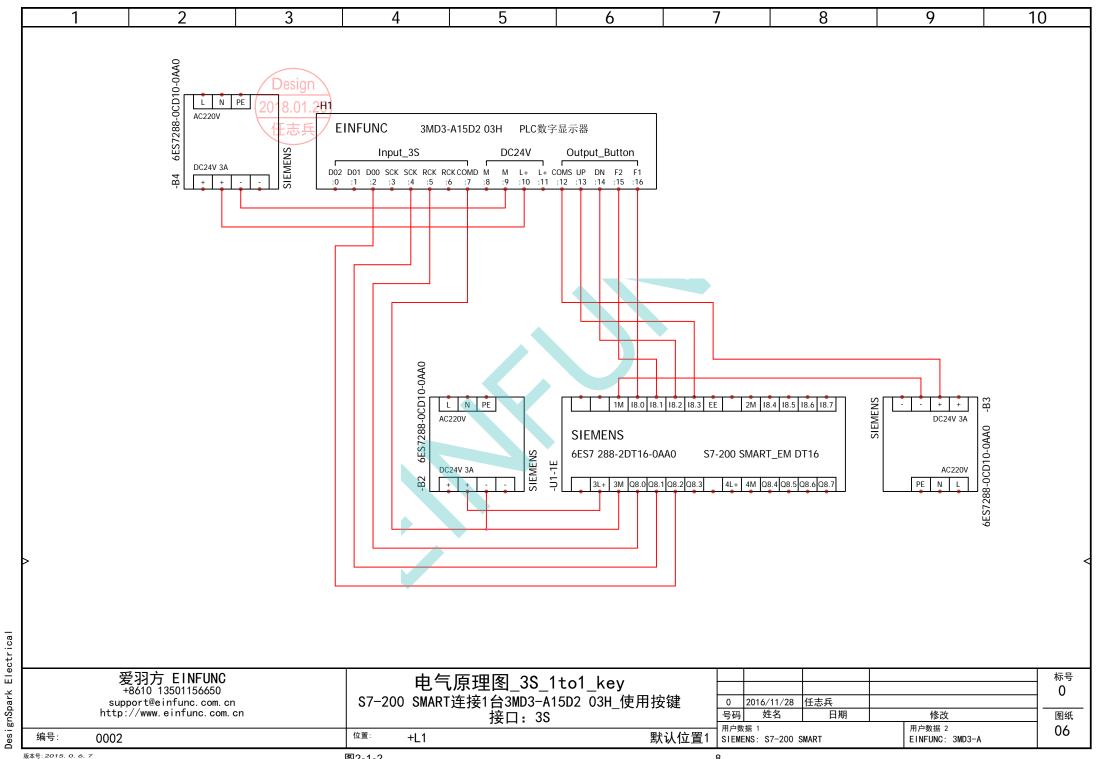
软件名称: SIMATIC STEP7-MicroWIN SMART V2.3 国际标

识

本驱动代码用"驱动一台数显+按键操作"来示例。







块: MAIN

作者: 任志兵 EINFUNC 创建日期: 2018.01.20 14:10:02 上次修改日期: 2018.02.24 17:27:00

| | 地址 | 符 号 | 变量类型 | 数据类型 | 注释 |
|---|----|----------------|------|------|----|
| 1 | | • | TEMP | | |
| 2 | | • | TEMP | | |
| 3 | | | TEMP | | |
| 4 | | | TEMP | | |

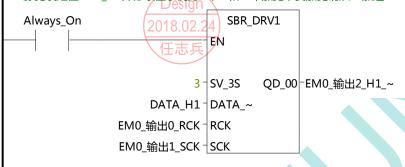
组织流程

程序段 1

发送数据

向一台单数据端口的 3S 外设发送数据

3S 脉宽设定值 #SV_3S 外部取值示例为 3,0 和 1 不加宽,要加宽请从 2 加起

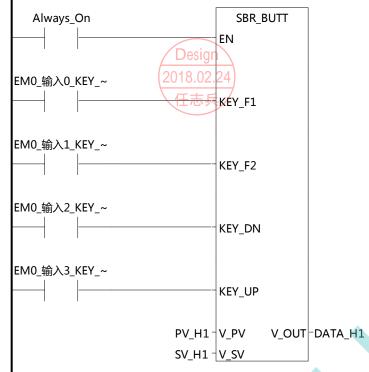


| 符号 | 地址 | 注释 |
|----------------|-------|----------|
| Always_On | SM0.0 | 始终接通 |
| DATA_H1 | VW100 | 被发送数据_H1 |
| EM0_输出0_RCK | Q8.0 | 并行时钟 |
| EM0_输出1_SCK | Q8.1 | 串行时钟 |
| EM0_输出2_H1_D00 | Q8.2 | 数据出口_H1 |

程序段 2

使用按键

本网络并非必须, 当需要使用 3S 外设面板上的自定义按键切换显示数据或修改数据数值时, 才使用本网络



| 符号 | 地址 | 注释 |
|----------------|--------|---------------|
| Always_On | SM0.0 | 始终接通 |
| DATA_H1 | VW100 | 被发送数据_H1 |
| EM0_输入0_KEY_F1 | I8.0 | 设定按键 |
| EM0_输入1_KEY_F2 | I8.1 | 加速按键 |
| EM0_输入2_KEY_DN | 18.2 | 下降按键 |
| EM0_输入3_KEY_UP | I8.3 | 上升按键 |
| PV_H1 | VW200 | 当前值_温度 |
| SV_H1 | VW1000 | 设定值_温度(断电保持型) |
| | | |

块:SBR_DRV1作者:任志兵 EINFUNC创建日期:2018.01.2014:10:02上次修改日期:2018.02.2417:25:03

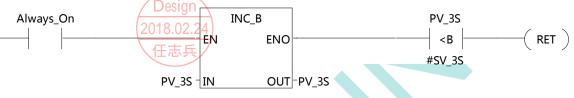
| | 地址 | 符 号 | 变量类型 | 数据类型 | 注释 |
|---|------|----------------|--------|------|----------|
| 1 | | EN | IN | BOOL | |
| 2 | LB0 | SV_3S | IN | BYTE | 设定值_3S脉宽 |
| 3 | LW1 | DATA_00 | IN | WORD | 被发送数据_00 |
| 4 | L3.0 | RCK | IN_OUT | BOOL | 并行时钟 |
| 5 | L3.1 | SCK | IN_OUT | BOOL | 串行时钟 |
| 6 | L3.2 | QD_00 | OUT | BOOL | 数据出口_00 |
| 7 | | | TEMP | | |

数据发送器

程序段 1

判断脉宽

3S 脉宽当前值累计 CPU 的扫描次数,当它小于 3S 脉宽设定值时,返回



 符号
 地址
 注释

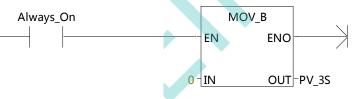
 Always_On
 SM0.0
 始终接通

 PV_3S
 VB0
 当前值_3S脉宽(背景数据)

程序段 2

脉宽清零

否则, 3S 脉宽当前值清零, 并进入下面的 3S 驱动代码

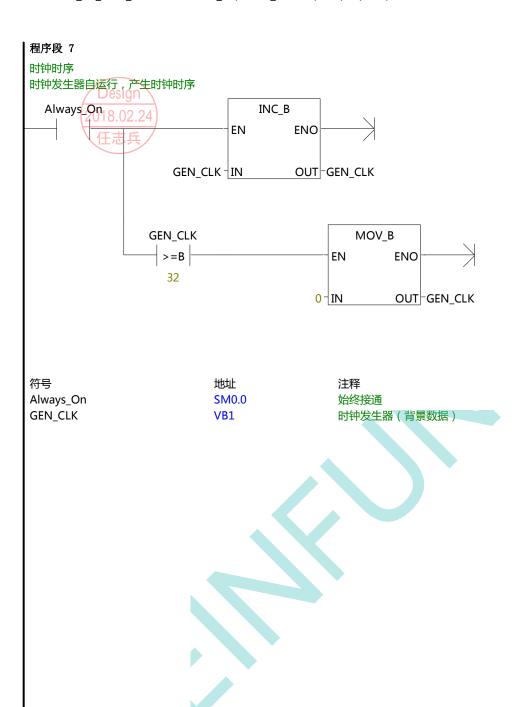


 符号
 地址
 注释

 Always_On
 SM0.0
 始终接通

 PV_3S
 VB0
 当前值_3S脉宽(背景数据)

程序段 3 并行时钟 时钟发生器清零时,被发送数据装载至移位通道上,并使能并行时钟 GEN CLK MOV_W #RCK ==B ΕN ENO 0 OUT CH_00 #DATA_00 ⁻IN 符号 地址 注释 CH_00 VW10 移位通道_00(背景数据) GEN_CLK VB1 时钟发生器(背景数据) 程序段 4 数据移位 在串行时钟的驱动下,移位通道上的数据左移 #SCK #RCK SHL_W ΕN **ENO** CH_00 IN OUT-CH_00 1 N 符号 地址 注释 CH_00 VW10 移位通道_00(背景数据) 程序段 5 串行时钟 时钟发生器最低位使能串行时钟 V1.0 #SCK 程序段 6 数据出口 移位通道的最高位使能数据出口 #QD_00 V10.7



块:SBR_BUTT作者:任志兵 EINFUNC创建日期:2018.01.20 14:23:07上次修改日期:2018.01.26 21:20:19

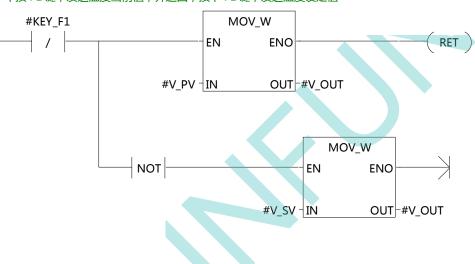
| | 地址 | 符 号 | 变量类型 | 数据类型 | 注释 |
|------------|--------------|----------------|--------|------|------|
| Design | \. | EN | IN | BOOL | |
| Design | L 0.0 | KEY_F1 | IN | BOOL | 设定按键 |
| (2018.03.2 | 240.1 | KEY_F2 | IN | BOOL | 加速按键 |
| 任志年 | L0.2 | KEY_DN | IN | BOOL | 下降按键 |
| 5 | L0.3 | KEY_UP | IN | BOOL | 上升按键 |
| 6 | LW1 | V_PV | IN | INT | 当前值 |
| 7 | LW3 | V_SV | IN_OUT | INT | 设定值 |
| 8 | LW5 | V_OUT | OUT | INT | 输出值 |
| 9 | | | TEMP | • | |

数值修改器

程序段 1

切换温度值

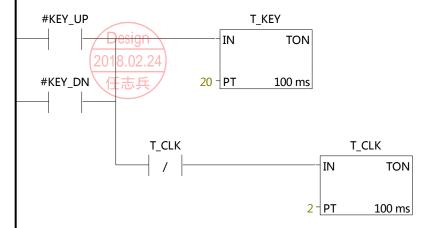
不按 F1 键,发送温度当前值,并返回;按下 F1 键,发送温度设定值



程序段 2

修改设定值:延时两秒

按住升降键不动,两秒钟后定时器 T_KEY 动作 T_CLK 定时器自激,产生 0.2 秒时钟脉冲



符号 地址 注释

 T_CLK
 T255
 定时器_时钟脉冲

 T_KEY
 T254
 定时器_按键按住

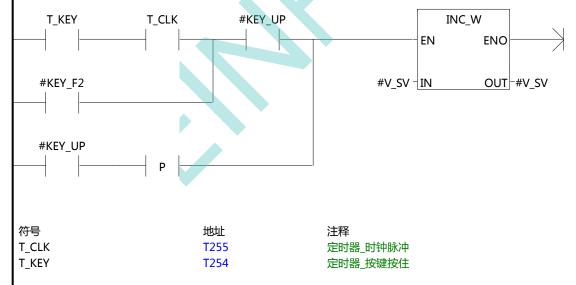
程序段 3

修改设定值:设定值增加

按住 F1 设定键,同时点动 UP键,可细调增加设定值

按住 F1 设定键,同时按住 UP 键不放,两秒钟以后设定值呈较快速度增加

按住 F1 设定键,同时按住 F2 加速键和 UP键,设定值则以最快速度增加



程序段 4 修改设定值:设定值减少 按住 F1 设定键,同时点动 DN 键,可细调减少设定值 按住 F1 设定键,同时按住 DN 键不放,两秒钟以后设定值呈较快速度减少 按住 F1 设定键,同时按住 F2 加速键和DN键,设定值则以最快速度减少 T_KEY T_CLK #KEY_DN DEC_W ΕN ENO #KEY_F2 #V_SV IN OUT #V_SV #KEY_DN 符号 地址 注释 T_CLK T255 定时器_时钟脉冲 T_KEY T254 定时器_按键按住

| | 🦳 💂 筏 | 符号 | 地址 | 注释 |
|---|-------|---------|------------------------|----------------|
| 1 | T. | _KEY | T254 | 定时器_按键按住 |
| 2 | T. | _CLK | Design T255 | 定时器_时钟脉冲 |
| 3 | P | PV_3S | VPU | 当前值_3S脉宽(背景数据) |
| 4 | G | GEN_CLK | (2018.02.2 √ β1 | 时钟发生器(背景数据) |
| 5 | C | CH_00 | 任志兵 YW10 | 移位通道_00(背景数据) |
| 6 | D | DATA_H1 | VW100 | 被发送数据_H1 |
| 7 | P | V_H1 | VW200 | 当前值_温度 |
| 8 | S | SV_H1 | VW1000 | 设定值_温度(断电保持型) |



SBR_DRV1 2018.02.SBR_BUTT 任志丘MAIN

1

2

3

地址 SBR0 SBR1 OB1

注释 数据发送器 数值修改器 组织流程



| | | 符号 | 地址 | 注释 |
|----|----------|----------------|------|---------|
| 1 | | CPU_输入0 | I0.0 | |
| 2 | _ | CPU_输入1 | I0.1 | |
| 3 | _ | CPU_输入2 | I0.2 | |
| 4 | _ | CPU_输入3 | I0.3 | |
| 5 | _ | CPU_输入4 | I0.4 | |
| 6 | _ | CPU_输入5 | I0.5 | |
| 7 | _ | CPU_输入6 | I0.6 | |
| 8 | _ | CPU_输入7 | I0.7 | |
| 9 | = | CPU_输入8 | I1.0 | |
| 10 | = | CPU_输入9 | I1.1 | |
| 11 | _ | CPU_输入10 | I1.2 | |
| 12 | = | CPU_输入11 | I1.3 | |
| 13 | | EM0_输入0_KEY_F1 | I8.0 | 设定按键 |
| 14 | | EM0_输入1_KEY_F2 | I8.1 | 加速按键 |
| 15 | | EM0_输入2_KEY_DN | I8.2 | 下降按键 |
| 16 | | EM0_输入3_KEY_UP | I8.3 | 上升按键 |
| 17 | = | EM0_输入4 | I8.4 | |
| 18 | | EM0_输入5 | I8.5 | |
| 19 | = | EM0_输入6 | I8.6 | • |
| 20 | = | EM0_输入7 | I8.7 | |
| 21 | = | CPU_输出0 | Q0.0 | |
| 22 | | CPU_输出1 | Q0.1 | |
| 23 | _ | CPU_输出2 | Q0.2 | |
| 24 | _ | CPU_输出3 | Q0.3 | |
| 25 | _ | CPU_输出4 | Q0.4 | |
| 26 | _ | CPU_输出5 | Q0.5 | |
| 27 | _ | CPU_输出6 | Q0.6 | |
| 28 | _ | CPU_输出7 | Q0.7 | |
| 29 | | EM0_输出0_RCK | Q8.0 | 并行时钟 |
| 30 | | EM0_输出1_SCK | Q8.1 | 串行时钟 |
| 31 | | EM0_输出2_H1_D00 | Q8.2 | 数据出口_H1 |
| 32 | _ | EM0_输出3 | Q8.3 | |
| 33 | _ | EM0_输出4 | Q8.4 | |
| 34 | | EM0_输出5 | Q8.5 | • |
| 35 | = | EM0_输出6 | Q8.6 | |
| 36 | | EM0_输出7 | Q8.7 | |
| | | | | |



2) S7-200 SMART 驱动三台 3S 外设_多数据线

■ 项目任务

将 S7-200 SMART 中的变量 DATA_H1 和 DATA_H2 的有符号整数,和 DATA_H3 的有符号双整数,通过 3S 接口分别传送给三台 3MD3-A 系列数字显示器,以十进制数形式将数据-32768~+32767 或±99999999 毫无偏差地显示出来。

■ 硬件连接

表 2-2-1 硬件明细表

| 硬件符号 | 硬件名称 | 硬件规格型号 | 说明 |
|------|----------|---------------------|-----------------------------|
| -U1 | 可编程序控制器 | 6ES7 288-1SR20-0AA0 | PLC CPUs |
| -01 | 扩展输入输出模块 | 6ES7 288-2DT16-0AA0 | 8 DI_DC24V / 8 DO_晶体管,源电流输出 |
| -B2 | 开关电源 | 6EP1331-5BA00 | PLC 输出电源, DC24V 0.6A |
| -B4 | 开关电源 | 6EP1331-5BA00 | 3S 外设工作电源,DC24V 0.6A |
| -H1 | 数字显示器 | 3MD3-A15D2 03H | 单行五位数码,DC24V,3S通讯接口,16Bits |
| -H2 | 数字显示器 | 3MD3-A15D2 03H | 单行五位数码,DC24V,3S通讯接口,16Bits |
| -H3 | 数字显示器 | 3MD3-A18D2 03H | 单行八位数码,DC24V,3S 通讯接口,32Bits |

本示例中的电源配置为"标准配置"。

从图 2-2-1 中不难看出,这三台数字显示器是共用 RCK 和 SCK 的,这正是 3S 通讯技术的优势所在。三台显示器并不是占用 S7-200 SMART 的 10 个输出点,而是只占用 6 个输出点。

■ 程序代码

项目名称: SIEMENS_3S_V5.0_S7-200 SMART_1

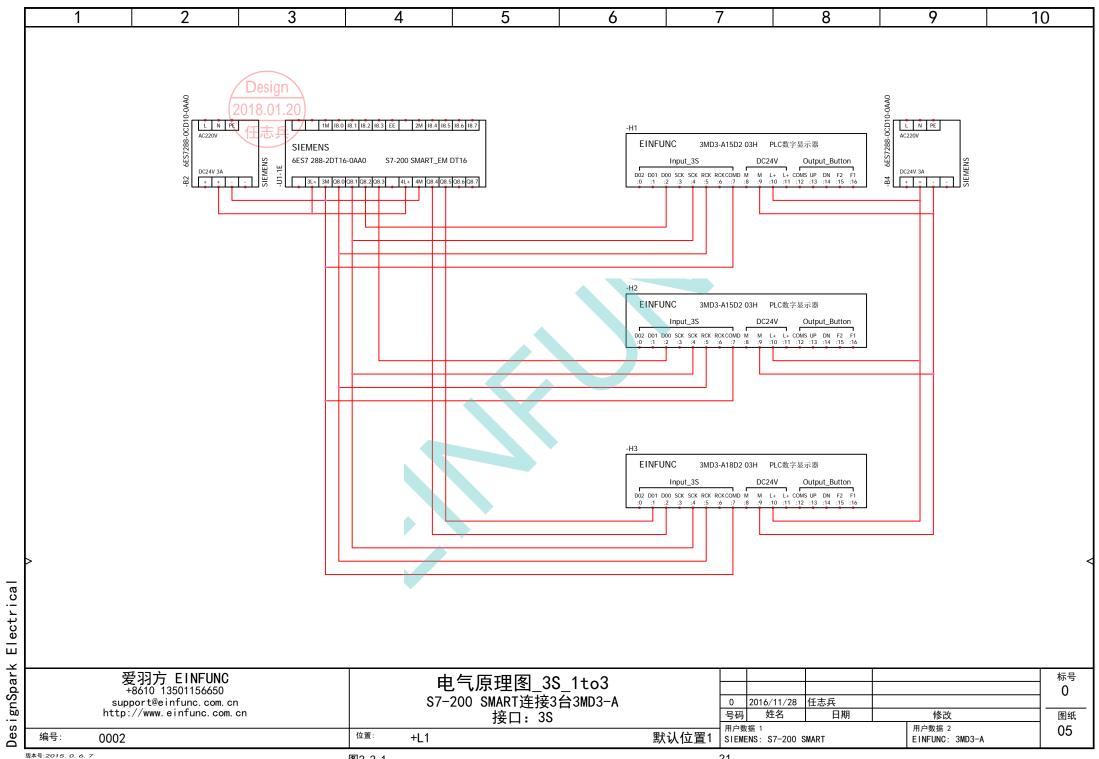
软件名称: SIMATIC STEP7-MicroWIN SMART V2.3 国际标

识

■ 结构对比

对比 SBR_DRV3 与 SBR_DRV1,程序结构并没有改变,只是在并行时钟网络增加了被装载的数据,在数据移位网络增加了数据移位通道;在数据出口网络增加了数据出口输出点。所以,驱动程序量并不因显示器台数的增加而成倍增加。这个例子也说明无论爱羽方产品型号如何,只要具备 3S 通讯接口,都可以按照这种"时钟共用"的方式扩展 3S 通讯。





块: MAIN

作者: 任志兵 EINFUNC 创建日期: 2018.01.20 14:10:02 上次修改日期: 2018.02.24 15:03:55

| | 地址 | 符 号 | 变量类型 | 数据类型 | 注释 |
|---|----|----------------|------|------|----|
| 1 | | | TEMP | | |
| 2 | | | TEMP | | |
| 3 | | | TEMP | | |
| 4 | | | TEMP | • | |

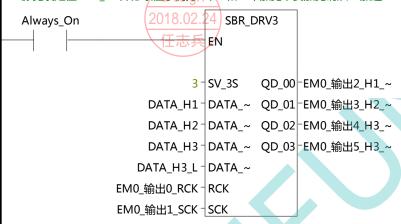
组织流程

程序段 1

发送数据

向两台单数据端口和一台双数据端口的 3S 外设发送数据

3S 脉宽设定值 #SV_3S 外部取值示例为 3,0 和 1 不加宽,要加宽请从 2 加起



| 符号 | 地址 | 注释 |
|----------------|-------|------------|
| Always_On | SM0.0 | 始终接通 |
| DATA_H1 | VW100 | 被发送数据_H1 |
| DATA_H2 | VW102 | 被发送数据_H2 |
| DATA_H3 | VW104 | 被发送数据_H3 |
| DATA_H3_L | VW106 | 被发送数据_H3_L |
| EM0_输出0_RCK | Q8.0 | 并行时钟 |
| EM0_输出1_SCK | Q8.1 | 串行时钟 |
| EM0_输出2_H1_D00 | Q8.2 | 数据出口_H1 |
| EM0_输出3_H2_D00 | Q8.3 | 数据出口_H2 |
| EM0_输出4_H3_D00 | Q8.4 | 数据出口_H3_低字 |
| EM0_输出5_H3_D01 | Q8.5 | 数据出口_H1_高字 |
| | | |

块:SBR_DRV3作者:任志兵 EINFUNC创建日期:2018.01.2014:10:02上次修改日期:2018.02.2415:00:51

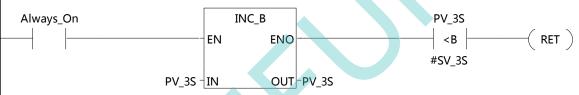
| 地址 | 业 符号 | 变 | 量类型 数据类型 | <u></u> 注释 |
|-------------------------|-------------|----|-----------|------------|
| 1 . | EN | IN | BOOL | |
| 2 LB0 | 0 SV_3S | IN | BYTE | 设定值_3S脉宽 |
| Design Lw | /1 DATA_00 | IN | WORD | 被发送数据_00 |
| 2018. 4 2.24 | /3 DATA_01 | IN | WORD | 被发送数据_01 |
| 5 LW | /5 DATA_02 | IN | WORD | 被发送数据_02 |
| 性态 KW | /7 DATA_03 | IN | WORD | 被发送数据_03 |
| 7 L9. | 0 RCK | IN | _OUT BOOL | 并行时钟 |
| 8 L9. | 1 SCK | IN | _OUT BOOL | 串行时钟 |
| 9 L9. | 2 QD_00 | Ol | JT BOOL | 数据出口_00 |
| 10 L9. | 3 QD_01 | Ol | JT BOOL | 数据出口_01 |
| 11 L9. | 4 QD_02 | Ol | JT BOOL | 数据出口_02 |
| 12 L9. | 5 QD_03 | Ol | JT BOOL | 数据出口_03 |
| 13 . | | TE | MP . | |

数据发送器

程序段 1

判断脉宽

3S 脉宽当前值累计 CPU 的扫描次数, 当它小于 3S 脉宽设定值时, 返回

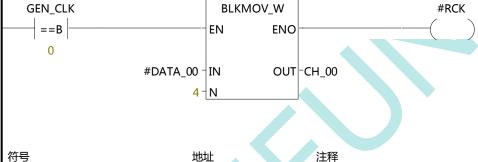


符号 Always_On PV_3S 地址 SM0.0 VB0

注释 始终接通

当前值_3S脉宽(背景数据)

程序段 2 脉宽清零 否则, 3S 脉宽当前值清零,并进入下面的 3S 驱动代码 2018.02.24 MOV_B Always_On ΕŃ ENO OUT PV_3S 0 IN 符号 地址 注释 Always_On SM0.0 始终接通 PV_3S VB0 当前值_3S脉宽(背景数据) 程序段 3 并行时钟 时钟发生器清零时,被发送数据装载至移位通道上,并使能并行时钟 GEN_CLK BLKMOV_W ==B ΕN **ENO** 0



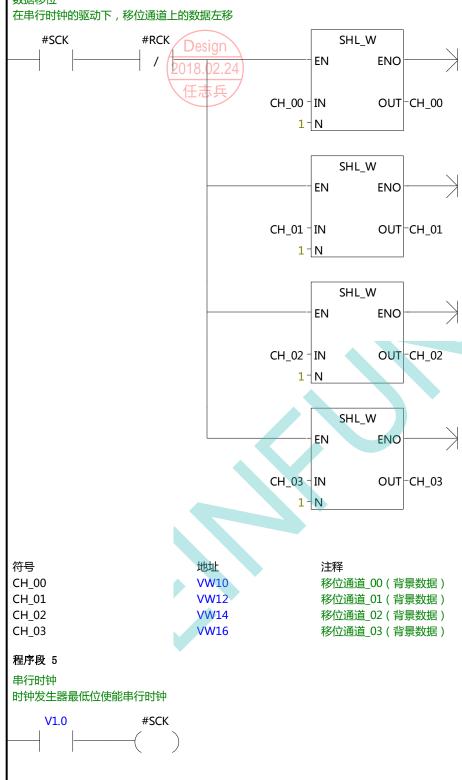
 符号
 地址
 注释

 CH_00
 VW10
 移位通道_00 (背景数据)

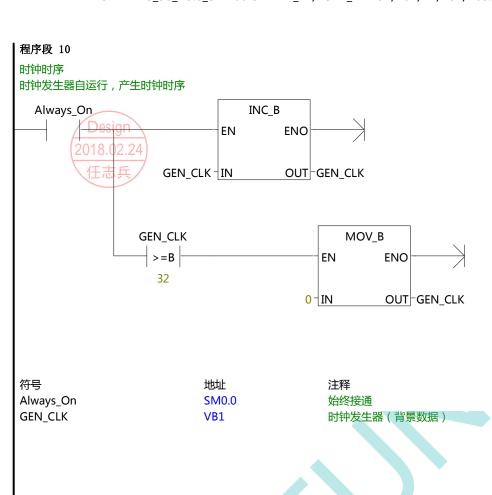
 GEN_CLK
 VB1
 时钟发生器(背景数据)



数据移位



程序段 6 数据出口 移位通道的最高位使能数据出口 #QD_00 程序段 7 数据出口 移位通道的最高位使能数据出口 V12.7 #QD_01 程序段 8 数据出口 移位通道的最高位使能数据出口 #QD_02 V14.7 程序段 9 数据出口 移位通道的最高位使能数据出口 #QD_03 V16.7



| | 🤁 🔙 符号 | 地址 | 注释 |
|----|---------------|-------|----------------|
| 1 | PV_3S | VB0 | 当前值_3S脉宽(背景数据) |
| 2 | GEN_CLK | VB1 | 时钟发生器(背景数据) |
| 3 | Design сн_00 | VW10 | 移位通道_00(背景数据) |
| 4 | 2018 02 SH_01 | VW12 | 移位通道_01(背景数据) |
| 5 | CH_02 | VW14 | 移位通道_02(背景数据) |
| 6 | 性态共€H_03 | VW16 | 移位通道_03(背景数据) |
| 7 | DATA_H1 | VW100 | 被发送数据_H1 |
| 8 | DATA_H2 | VW102 | 被发送数据_H2 |
| 9 | DATA_H3 | VW104 | 被发送数据_H3 |
| 10 | DATA_H3_L | VW106 | 被发送数据_H3_L |



1 2 地址 SBR0 OB1 注释 数据发送器 组织流程



| | 9 | | 符号 | 地址 | 注释 |
|----|---|-----------|----------------|------|--------------|
| 1 | | | CPU_输入0 | IO.0 | <i>1</i> ⊥1+ |
| 2 | | Ö | CPU_输入1 | IO.1 | • |
| 3 | | Ö | CPU_输入2 | IO.2 | • |
| 4 | | Ö | CPU_输入3 | IO.3 | • |
| 5 | | Ö | CPU_输入4 | I0.4 | |
| 6 | | Ö | CPU_输入5 | I0.5 | |
| 7 | | Ö | CPU_输入6 | I0.6 | |
| 8 | | | CPU_输入7 | I0.7 | |
| 9 | | | CPU_输入8 | I1.0 | |
| 10 | | | CPU_输入9 | I1.1 | |
| 11 | | | CPU_输入10 | I1.2 | |
| 12 | | | CPU_输入11 | I1.3 | |
| 13 | | | EM0_输入0 | I8.0 | |
| 14 | | | EM0_输入1 | I8.1 | |
| 15 | | | EM0_输入2 | I8.2 | |
| 16 | | | EM0_输入3 | I8.3 | |
| 17 | | \Box | EM0_输入4 | I8.4 | |
| 18 | | \Box | EM0_输入5 | I8.5 | |
| 19 | | \Box | EM0_输入6 | I8.6 | |
| 20 | | \Box | EM0_输入7 | I8.7 | |
| 21 | | \Box | CPU_输出0 | Q0.0 | |
| 22 | | \Box | CPU_输出1 | Q0.1 | |
| 23 | | \Box | CPU_输出2 | Q0.2 | |
| 24 | | \Box | CPU_输出3 | Q0.3 | |
| 25 | | \Box | CPU_输出4 | Q0.4 | |
| 26 | | \square | CPU_输出5 | Q0.5 | |
| 27 | | \square | CPU_输出6 | Q0.6 | |
| 28 | | \Box | CPU_输出7 | Q0.7 | |
| 29 | | | EM0_输出0_RCK | Q8.0 | 并行时钟 |
| 30 | | | EM0_输出1_SCK | Q8.1 | 串行时钟 |
| 31 | | | EM0_输出2_H1_D00 | Q8.2 | 数据出口_H1 |
| 32 | | | EM0_输出3_H2_D00 | Q8.3 | 数据出口_H2 |
| 33 | | | EM0_输出4_H3_D00 | Q8.4 | 数据出口_H3_低字 |
| 34 | | _ | EM0_输出5_H3_D01 | Q8.5 | 数据出口_H1_高字 |
| 35 | | | EM0_输出6 | Q8.6 | |
| 36 | | W | EM0_输出7 | Q8.7 | |



3) 关于驱动例程背景数据的存储分布

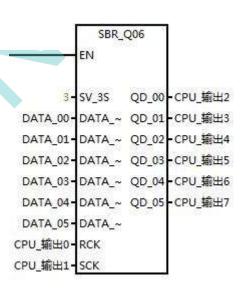
SIMATIC STEP7-MicroWIN 编程软件系统的结构符合二十世纪九十年代的 IEC1131-3 标准,这在那个年代是非常先进的,没有哪家公司的小型 PLC 编程软件系统能比得上 SIMATIC STEP7-MicroWIN。到了二十一世纪一十年代 SIMATIC STEP7-MicroWIN SMART 完全继承了这种系统结构,没有本质的变化。

由于 SIMATIC STEP7-MicroWIN SMART 子程序所使用的传递参数总数不能超过 16 个,而 PLC 硬件系统 又没有为每个子程序分配专用的存储器,只能使用 L 型临时存储器。块内部使用的变量如 3S 脉宽当前值 PV_3S、时钟发生器 GEN_CLK 和 CH_xx 等都是背景数据,存储程序运算过程中的数据(反复运算时数据不能丢失),所以 对子程序中使用的无关块输入输出的背景数据变量的存储分布需要仔细斟酌考虑。下面以 3S 驱动例程块为例,对 使用背景数据变量的存储分布方式做一下对比,供您在编程时参考。

■ 背景数据使用全局变量

如本章第 1 节和第 2 节所示,将背景数据分配在全局变量 V 存储器中(只要分配好 VB/VW/VD 号即可),将您最关心的数据如 DATA_xx、RCK、SCK 和 Hxx_D00 等分配在块的输入输出管脚上,如右图所示。

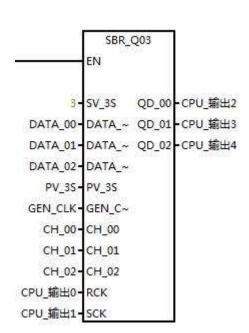
- ❖ 优点:例程块外观最标准,最美观,最直观,一目了然;
- ❖ 缺点:只能驱动六个 3S 数据出口,且不能简单任意移植, 移植时要重新分配背景数据的 V 存储器。



■ 背景数据使用 IN-OUT 参数

背景数据分配在 IN_OUT 传递参数里,读写至全局 V 存储器,如右图所示。

- ❖ 优点:例程块可以任意移植;
- ❖ 缺点:最多只能驱动三个 3S 数据出口,并且不美观,干扰了您查看最关心的输入输出变量。



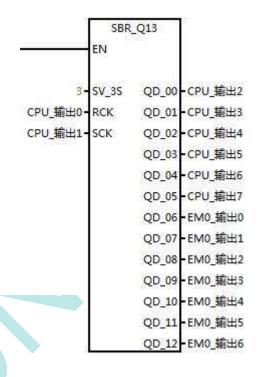


■ 背景数据和输入数据用全局变量

例程块的主要输入数据(如 DATA_H00~DATA_H12)和背景数据均分配在全局变量 V 存储器中,您只需将要发送的数据写在全局符号表指定的 VW 地址上,例程块的输入侧几乎没有输入管脚,外观上看不到被发送数据,这种例程块最多可以驱动十三个 3S 数据出口,如右图所示。

❖ 优点:广普适用;

❖ 缺点:不能简单任意移植。



■ 所有数据均使用全局变量

例程块全部变量不分内外,均分配在全局变量存储器中,例程 块两侧除 EN 外看不到任何管脚,如右图所示。

❖ 优点:可驱动的 3S 数据出口无限制;

缺点:例程块外观不直观,不能简单任意移植,移植时要 重新分配大量的数据存储地址。 SBR_more EN



北京爱羽方模块科技发展中心 EINFUNC

地址:中华人民共和国 北京市 海淀区 上地十街1号院 辉煌国际4号楼1902室

邮编:100085

电话:+8610-62175465 移动:+86 13501156650

官网: http://www.einfunc.com.cn 支持: support@einfunc.com.cn 商务: business@einfunc.com.cn





爱羽方_官方网站

爱羽方_官方微信