PROFIBUS-OEM2 调试实验系统 用户手册

V 1.0





北京鼎实创新科技有限公司

目 录

第一	·章	PROFIBUS-OEM2 调试实验系统概述	. 4
	1.	什么是嵌入式 PROFIBUS 接口 OEM2	.4
	2.	OEM2 调试实验系统主要用途	. 4
	3.	调试实验系统调试模式	. 4
		(1) 调试模式 1: 实验板调试模式	. 4
		(2) 调试模式 2: OEM2 接口板调试模式	. 4
		(3) 调试模式 3: PC 机调试模式	. 5
	4.	OEM2 调试实验系统主要部件及选件	. 6
		(1) 基本配置	. 6
		(2) 主站部分	. 6
	5.	安全指南	. 6
第二	章	OEM2 开发实验板	.7
	1.	PB-OEM2-SAMPLE 布局图	. 7
	2.	OEM2 开发实验板原理图	. 9
		(1) 原理图-1	14
		(2) 原理图-2	14
		(3) 原理图-3	14
		(4) 原理图-4	14
第三	章	实验板程序、GSD 文件	15
	1.	. 实验板程序	15
		(1) 实验板 CPU	15
		(2) 实验板程序	15
	2.	· 关于 GSD 文件(Electronic Data Sheet)	15
	3.	· PB-OEM2-SE 的 ID 号及 GSD 文件	16
	4.	·用户产品的 ID 号、GSD 文件及产品认证	16
第四	章	实验板程序举例	18
	俪	1. 简单工作模式	18
	1	(1) 技术参数	18
		 (2) SE 初始化报文 	18
		(3) 数据交换状态下实验板与 SE 接口的数据报文	19
		(4) 实验板初始化 SE 接口板的过程	19
		(5) 数据交换状态下实验板简单工作模式	20
		(6) 实验板程序清单	20
		(7) 例 1 的 GSD 文件	29
	例	2.监测 SE 接口状态的工作模式	31
		(1) 技术参数:	31
		(2) SE 初始化报文	31
		(3) 数据交换状态下实验板与 SE 接口的数据报文	32
		(4) SE 接口状态信号 REQ_IT 与 S_RTS/P17 的时序图	32
		(5) 实验板初始化 SE 接口板的过程	33
		(6) 数据交换状态下实验板带监测接口状态功能工作模式	34
		(7) 实验板程序清单	35

	(8) 例 2 的 GSD 文件	
例	3. 带用户参数功能的工作模式	
	(1) 需要使用"用户参数 user_prm"的情况	46
	(2) 实验 I/O 功能及实现方法	
	(3) 具体确定"用户参数"类型、个数、取值范围	46
	(4) 带有"用户参数"描述的 GSD 文件	
	(5) 如何在主站配置中选择用户参数	
	(6) 更便于用户使用的 GSD 文件	51
	(7) 带用户参数功能的 SE 接口的初始化	
	(8) 带用户参数功能数据交换过程	57
	(9) 实验板初始化 SE 接口板的过程框图	59
	(10) 数据交换状态下实验板工作流程	59
	(11) 例 3 实验 I/O 的程序清单	61
第五章	建立一个调试实验系统	73
1.	建立 OEM2 调试实验系统 I	
	(1) 系统 I 设备清单	
	(2) 系统 [结构图	74
	(3) 安装实验调试系统 [74
	(→) 硬件安装	74
	(二) 主站 PC 机软件安装	74
	(4) 按照系统提供的例 2 实现系统连通实验	75
	(→) 在 COM PROFIBUS 配置软件中实现简单连通	75
	〇 自己完成配置例2实现简单连通	80
2.	CP5611+Wincc 做主站实现系统 I 的 PROFIBUS 通信	81
	(1) 设置"Set PG/PC Interface":	
	(2) 建立 WinCC 项目	
	 (2) 建立 WinCC 项目	
3.	 (2) 建立 WinCC 项目	
3.	 (2) 建立 WinCC 项目	
3.	 (2) 建立 WinCC 项目	
3.	 (2) 建立 WinCC 项目	
3.	 (2) 建立 WinCC 项目	81 83 83 83 84 84 84 85 85 85 85
3.	 (2) 建立 WinCC 项目	81 83 83 83 84 84 84 85 85 85 85 85 85
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 85 85 85
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 85 85 86 88
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 85 85 86 88 88 89
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 85 85 86 88 89 90
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 86 88 89 90 90 95
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 86 88 89 90 90 95 95
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 85 85 85 85 85 85 85 86 88 89 90 90 95 95 95
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 85 86 88 89 90 90 95 95 95 95 95
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 85 85 85 85 85 85 85 85 86 88 89 90 90 95 95 95 95 95 96 96
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 85 86 88 89 90 90 95 95 95 95 95 95 95 96 96 96
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 86 88 89 90 90 90 95 95 95 95 95 95 95 96 96 96 96
3.	 (2) 建立 WinCC 项目	81 83 83 84 84 84 85 85 85 85 85 85 85 85 85 86 88 89 90 90 95 95 95 95 95 95 95 95 95 96 96 96 96 96

(3) 实验板程序下载	
① 主要调试功能	
② MODE 开关设置	
③ 操作方式	
(4) 调试模式 3: PC 机调试模式	
① 主要调试功能	
② MODE 开关设置	
③ 操作方式	
附件 A: 下载实验板程序操作指南及下载线制作方法	104
1. 实验板的下载模式选择	104
2. Flash Magic 软件的程序下载操作	104
3. 恢复实验板运行状态	106
4. 下载线的制作及连接方法	106

第一章 PROFIBUS-OEM2 调试实验系统概述

OEM2 调试实验系统(以下简称"调试系统"),专为应用嵌入式 PROFIBUS 通信接口 OEM2 的用户,提供一个完整、最小化、最便捷的调试实验系统平台。

1. 什么是嵌入式 PROFIBUS 接口 OEM2

嵌入式 PROFIBUS 接口 OEM2(以下有时称 OEM2 卡)专为自主开发具有 PROFIBUS-DP 通信功能 产品的用户,以 OEM 方式提供从站的通信接口。产品型号是 PB-OEM2-SE,《PB-OEM2-SE 产品手册》 中有详细使用及原理说明。

2. 0EM2 调试实验系统主要用途

- (1)为 PROFIBUS 产品开发用户提供一个完整、最小化、最便捷的调试实验平台。
- (2) OEM2 调试实验系统包括:主站、从站实验板、PROFIBUS 电缆及接插件、相关配置组态和调试软件、 产品开发例程等。调试实验系统共提供3种调试模式。
- (3) 系统包括一块实验板。这是 OEM2 卡的评估板,其硬件原理图和 C 代码源程序完全公开,为用户产品开发提供样板。

3. 调试实验系统调试模式

(1) 调试模式 1: 实验板调试模式

主要调试功能:建立主站、从站的配置、PROFIBUS 系统的连通、PROFIBUS I/O 通信。用户可以 先使用实验板按照本手册完成与 PROFIBUS 主站的连接,然后再将自主开发的用户板与 PROFIBUS 主站 连接,见图 1-1。



图 1-1 调试模式 1—实验板调试模式

(2) 调试模式 2: OEM2 接口板调试模式

主要调试功能: 监测实验板与 OEM2 接口板的通信数据,可检验初始化数据、PROFIBUS I/O 数据、用户参数数据等,是学习、调试 OEM2 接口板的主要方法,见图 1-2。

_	OME2	接口板调试模式	
	PROFIBUS		
PROFIBUS 主站		实验板及 OEM2 卡	PC 机监测数据

图 1-2 调试模式 2—OEM2 接口板调试模式

实验板程序下载: 可将实验板 CPU 的 C 源程序编译、连接生成 .HEX 目标程序,通过软件 WINSIP 下载到实验板中。此工作可以配合调试模式 2 使用,用户可以自行修改实验板的程序,实现不同 PROFIBUS 配置的连接,见图 1-3。



图 1-3 实验板程序下载

(3) 调试模式 3: PC 机调试模式

主要调试功能:"调试模式 3"可替代"调试模式 2"方式。使用 PC 机 OEM2 调试软件: oem2_test.exe, 用 PC 机代替实验板 CPU, 实现与 OEM2 接口板的数据交换。这种调试模式可方便设定不同初始化数据、设定 PROFIBUS 输入数据,监测主站的 PROFIBUS 输出数据、用户参数数据,是学习、调试 OEM2 接口板的另一种方法,见图 1-4。



图 1-4 调试模式 3—PC 机调试模式

4. 0EM2 调试实验系统主要部件及选件

(1) 基本配置

	PROFIBUS-OEM2 调试实验系统基本配置(从站部分必选)				
序号	名称	制造商	数量	备注	
1	PB-OEM2-SE	鼎实科技	1块		
	嵌入式 PROFIBUS 接口				
2	PB-OEM2-SAMPLE OEM2 开发实验板	鼎实科技	1块		
2	PROFIBUS 电缆插头: 10米电缆+2插头	西门子	1套		
3	文件资料(光盘)包括: 手册、例程 1~3(COM PROFIBUS 配置、实验板 C 源 及目标程序、系统 I WinCC 项目文件、系统 II Step7 项 目文件、GSD 文件)及 GSD 文件、PC 机调试模式软件、 单片机程序下载软件、实验板硬件原理图、SE 外转接口 原理图。	鼎实科技	1 张 CD	随系统赠送	
4	手册(印刷品): 《PB-OEM2-SE 产品手册》 《PROFIBUS-OEM2 调试实验系统使用手册》	鼎实科技	1套	随系统赠送	

(2) 主站部分

主站部分有2种选择:

① 选择 CP5611+PC 机做主站,优点:价格低

选择 CP5611+PC 机做主站					
序号	名称	制造商	数量	备注	
1	CP5611 PROFIBUS 主站网卡	西门子	1块		
3	组态软件(光盘):	西门子	3CD	随系统赠送	
	COM PROFIBUS V5.1(DEMO版);				
	SOFTNET-DP V5.2(DEMO版);				
	WinCC V5.1(DEMO 版)				

② 选择 PLC 做主站,优点:系统可靠,可做为批量生产产品时,产品出厂的测试系统。

选择 PLC 做主站					
序号	名称	制造商	数量	备注	
1	S7-300PLC/CPU313-2DP	西门子	1块		
2	MPI 编程电缆	西门子	1根		
3	组态软件(光盘):	西门子	1CD	随系统赠送	
	STEP 7 V5.1+SP1(DEMO版)				

5. 安全指南

本手册包括应该遵守的注意事项,以保护产品和所连接的设备免受损坏。



警告 (Warning):

禁止带电插拔 RS232 电缆和 PROFIBUS 电缆。

第二章 OEM2 开发实验板

OEM2 开发实验板是调试系统中 PROFIBUS 的从站部件,是 OEM2 接口板的评估板。型号: PB-OEM2-SAMPLE。

开发实验板为用户提供了一块开发样板。实验板的全部硬件原理图和C源码程序可在本手册中找到。 用户使用 OEM2 开发实验板可以实现:

- ① 与 PROFIBUS 主站连接,了解如何实现 PROFIBUS 组态、配置;
- ② 依照实验板硬件原理图和 C 源码程序,可以更容易理解 PB-OEM2-SE 接口板的应用,设计用户产品的软硬件。
- ③ 实现调试实验系统的3种调试模式;
- 1. PB-OEM2-SAMPLE 布局图



图 2-1 PB-OEM2-SAMPLE 布局图

- (1) PROFIBUS 插座 J2: 用于连接 PROFIBUS 电缆插座。插座为 9 孔型(female)D 型插座。
- (2) 16 针插座 J1: 实验板上连接 PB-OEM2-SE 的 16 针(针)插座;
- (3) **晶振 14 .7456MH:**有源晶振插座可任意更换晶振,如果与接口板通信波特率选择为 9.6K, 19.2K, 38.4K, 57.6K, 115.2K, 推荐使用 14 .7456MH。
- (4) 单片机: P89V51RD2

- (5) RS232 插座 J3: 9 针 D 型插座
- (6) PROFIBUS 地址开关 SW1: PRODIBUS 从站地址设置开关,这个地址必须和主站系统硬件配置中该

从站的地址设置一致。



SW1 设置 PRODIBUS 从站地址例:

从站地址(max)=126(Dec)=7E(Hex)=01111110(Bin);

从站地址=50(Dec)=32(Hex)=00110010(Bin);

从站地址=19(Dec)=13(Hex)=00010011(Bin);

(7) 调试模式设置开关 SW2



① 调试模式 1: 实验板调试模式, MODE=11000000(Bin);



图 2-2 调试模式 1—实验板调试模式

② 调试模式 2: OEM2 接口板调试模式, MODE=11000010_{(Bin);}



图 2-3 调试模式 2—OEM2 接口板调试模式

③ 实验板程序下载, MODE=00000111(Bin);



图 2-4 实验板程序下载

④ 调试模式 3: PC 机调试模式, MODE=00100000_{(Bin);}



图 2-5 调试模式 3—PC 机调试模式

- (8) 24VDC 电源端子:实验板直流 24V 供电(18~36V),注意实验板上标定的正负极。
- (9) 24VDC 电源开关
- (10) Q0~Q15, 输出指示灯(红)
- (11) 10~115, 输入按钮
- (12) IL0~IL15, 输入指示灯(绿)
- (13) 复位按钮
- (14) 24VDC 电源指示灯(红)
- (15) 5V 电源指示灯(红)
- (16) Vcc 电源指示灯(红)
- (7) Txd 指示灯(红):实验板 CPU 与 OEM2 卡串行通信,实验板 CPU 发送指示灯;
- (18) Rxd 指示灯(绿):实验板 CPU 与 OEM2 卡串行通信,实验板 CPU 接收指示灯;
- (19) 外接口方式 2 的外转接插座 JP2
- 2. 0EM2 开发实验板原理图

在实验系统中,开发实验板与 OEM2 模块组成 PROFIBUS 从站,开发实验板是用户样板,用户在详细阅读《PB-OEM2-SE 产品手册》、了解 OEM2 模块应用技术后,可以参考实验板原理图开发用户从站。









13

(1) 原理图-1

- ① U7A: SN74ALS08,四与门
- ② U5: MAX232, RS232 驱动芯片
- ③ U1: 实验板 CPU, PHILIP/P89V52RD2
- ④ OC1: 14.7456MH 晶振
- ⑤ JP2: 外接口方式 2 的外转接插座 JP2
- ⑥ CONNECT-16: 16 针插座 J1,连接 PB-OEM2-SE 模块
- ⑦ RP21: 信号上拉排电组
- ⑧ DB91: PROFIBUS 插座 J2
- 9 DB232: RS232 插座 J3
- **⑩ SW2:** 调试模式设置开关 SW2
- (2) 原理图-2
- ① U2: 复位电路
- ② U4: DC/DC, 24VDC转 5V/0V
- ③ U9: DC/DC, 24VDC转 VCC/GND
- ④ 24V/OK、5V/OK、VCC/OK: 24V、5V、VCC 电源指示灯
- ⑤ RXD、TXD:实验板 CPU 与 OEM2 卡串行通信,实验板 CPU 发送、接收指示灯;
- ⑥ 24VP/24K/D1/D2: 24V 电源插座、开关、极性保护二级管;

(3) 原理图-3

- ① SW1/U20/RP20: PROFIBUS 地址开关 SW1 及相关电路。
- ② U16/Q0~Q7/RP16: PROFIBUS 输出锁存及 Q0~Q7 指示灯。
- ③ U17/Q8~Q15/RP17: PROFIBUS 输出锁存及 Q8~Q15 指示灯。
- (4) 原理图-4
- ① U18/IO0~IO7/RP18: I0~I7 输入键对应的 IO0~IO7 指示灯(绿)。
- ② U19/IO8~IO15/RP19: I8~I15 输入键对应的 IO8~IO15 指示灯(绿)。
- ③ U12/IN0~IN7/RP12: I0~I7 输入键及输入缓冲电路。注意:输入键不带自锁。
- ④ U13/IN8~IN15/RP13: I8~I15 输入键及输入缓冲电路。注意: 输入键不带自锁。

第三章 实验板程序、GSD 文件

1. 实验板程序

(1) **实验板 CPU**

采用 PHILIPS 公司 P89V51RD2, 主要技术指标:

- 80C51 核心处理单元;
- 5V 的工作电压,操作频率为 0~40MHZ;
- 64KB的片内 Flash 程序存储器,具有 ISP(在系统编程)和 IAP(在应用中编程)功能;
- 通过软件或 ISP 选择支持 12 时钟(默认)或6 时钟模式;
- SPI(串行外围接口)和增强型 UART;
- PCA (可编程计数器阵列),具有 PWM 和捕获/比较模式;
- 4个8位 I/0口,含有3个高电流 PI口(每个 I/0 口的电流为 16mA);
- 3个16位定时器/计数器;
- 可编程看门狗定时器 (WDT);
- 8个中断源,4个中断优先级;
- 2 个 DPTR 寄存器;
- 低 EMI 方式 (ALE 禁能);
- 兼容 TTL 和 CMOS 逻辑电平;
- 掉电检测;
- 低功耗模式;
 - ◆ 掉电模式,外部中断唤醒;
 - ◆ 空闲模式;
- PDIP40, PLC44 和 TQFP44 的封装;

(2) 实验板程序

- ① 出厂时实验板 CPU 程序是"例 2: 监测 SE 接口状态的工作模式"
- ② 产品 CD 光盘中有实验板 CPU 程序:
- 例 1: 简单工作模式,包括 C 源代码 user.c、目标码 b1.hex

例 2: 监测 SE 接口状态的工作模式,包括 C 源代码 user.c、目标码 b2.hex

- 例 3:带用户参数功能工作模式,包括 C 源代码 user.c、目标码 b3.hex
- ③ 如果用户希望该变出厂时实验板 CPU 程序"例 2: 监测 SE 接口状态的工作模式",可以按照实验板

工作模式"调试模式 2"中"实验板程序下载"方法,选择下载 b1.hex、b2.hex、b3.hex。

④ 关于例 1、例 2、例 3 功能详解及实验方法可以参考第四章、第五章内容。

2. 关于 GSD 文件 (Electronic Data Sheet)

① 每一个 PROFIBUS 从站都要有一个"设备描述文件"称为 GSD 文件,用来描述该 PROFIBUS-DP 设备的特性。

- ② GSD 文件包含了设备所有定义参数,如下:
- •支持的波特率;
- •支持的信息长度;

•输入/输出数据数量;

•诊断数据的含义;

•可选模块种类等。

③ GSD 文件是文本类文件,可用"记事本"编辑。

④ 无论使用什么样的系统配置软件,都要根据 GSD 文件来对 DP 设备配置。

⑤ 国际 PROFIBUS 组织 PI 提供了 GSD 文件编辑软件: gsdedit.exe.该软件依照 profibus 技术标准格式规定,对用户编辑的 GSD 文件进行格式检查。该软件的"帮助"功能强大,也是一种快速学习 GSD 文件 技术的途径。见图 3-1、图 3-2。

Die fat fin trou juin pie	_(e) x	Die 115 Die April Date dar	
= fill par ene parte		■ 第 780 582-703 ■ 第 780 582-703 ■ 第 780 582-703 ■ 第 780 582 703 ■ 第 780 582 703 1 ■ 1	
		48. http://write Minist/Amary 11-00011-0 48. http://write Minist/Amary 10-00017.0 48. http://write Minist/Amary 10-0017.0	
Developed by Table V. Bandhard M. S. Star Strand Strand Developed To 2013, Physical Strand Strand Strand Fact St. 100, The Star Strand Strand Strand Strand Fact St. 100, The Star Strand Strand Strand Developed Strand Strand Strand Strand Strand Star Strand Strand Strand Strand Strand Star Strand		3 Burble 3 4 000	, , ,
Brogtin Linete Prikila Cons ■ Line - Tsicos to G2_24 V2.1, Cupright _ 2000-00 by 700 Surnay		Inoption Line Pr. Provide Const.	
For Halg, press Fi	Fill Line: 1, Char: 1 Notified	💩 Breer Line 12 Key "Funder_Fane": Warperted character(n) in string found or missing quotatio	

图 3-1 GSDEDIT 打开一个空文件

图 3-2 GSDEDIT 打开 DS_06FA.GSD 文件

3. PB-OEM2-SE 的 ID 号及 GSD 文件

PB-OEM2-SE 产品已经在国际 PROFIBUS 组织 PI 备案,已取得 PI 授予 ID 号: 06FA 和 GSD 文件 名: DS_06FA.GSD。

4. 用户产品的 ID 号、GSD 文件及产品认证

- (1) SE 初始化报文中的 ID 号必须和 GSD 文件中的 ID 号一致才能连通;
- (2) 由于本产品以 OEM 方式销售,用户对应用本产品开发的 PROFIBUS 设备有自主知识产权和品牌; 因此,当用户产品正式销售提供给你的用户时,本产品的 ID 号和 GSD 文件名不宜作为用户产品的 ID 号和 GSD 文件名;
- (3)如果用户需要产品的测试认证,可以委托"中国 PROFIBUS 组织 CPO"向国际 PROFIBUS 组织 PI 办理申请产品认证手续,那时,用户可以得到自己的产品 ID 号和 GSD 文件名。用户还应与"中国 PROFIBUS 产品测试实验室 CPPTL"联系进行产品测试。产品测试合格后 CPPTL 将出据"测试报 告";国际 PROFIBUS 组织 PI 根据产品的"测试报告"决定给您的产品正式认证证书。
- (4) 用户也可以暂时自定义一个 ID 号,在产品开发时期使用。在一条 PROFIBUS 总线上,不同类型或相同类型具有同一 ID 号的从站并不影响系统连通。

(5) 用户产品的 GSD 文件

用户产品的 GSD 文件可以在本产品 GSD 文件基础上,在用户公司名、产品型号、系列号等处置换 成用户产品信息,即可成为用户的 GSD 文件。产品 CD 光盘备有例 1、例 2、例 3 的 GSD 文件: OEM2_B1.GSD、OEM2_B2.GSD、OEM2_B3.GSD 及实验板图标文件 SE_B.BMP,关于例 1、例 2、例 3GSD 文件细节,详见第四章有关内容。

第四章 实验板程序举例

例1. 简单工作模式

(1) 技术参数

PROFIBUS 输入/输出: 48 字节输入/48 字节输出;

用户参数: 无;

通信模式: 不检测 SE 接口状态条件;

串口波特率: 115.2K

(2) SE 初始化报文

I/O 配置数据长度 CFG_LEN=6;

I/O 配置数据为: 0x1f, 0x2f, 0x1f, 0x2f, 0x1f, 0x2f

用户参数长度 User_Prm_Data_Len=0;

接收数据长 SD_in_len≥48+1+1=50

发送数据长 SD_out_len≥48+1+1=50



(3) 数据交换状态下实验板与 SE 接口的数据报文

实验板不使用用户参数功能(即:初始化报文中 User_Prm_Data_Len=0),永远发请求 PROFIBUS 数据命令 req_com=0;接口板一定回答"输出数据报文 A"。见图 4-1-1,图 4-1-2



图 4-1-1 实验板发一输入数据报文 req_com=0

图 4-1-2 SE 接口板答一输出数据报文格式 A

(4) 实验板初始化 SE 接口板的过程





(5) 数据交换状态下实验板简单工作模式



图 4-1-4 简单工作模式下实验板程序流程图

(6) 实验板程序清单

/*+		
文件名称: 3	↓ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	L.L	
制作甲1型: 1	L 尔瑞头 创新科技公司	
版化计可:		
₩] <u> E:</u> \ amaail: ta	vww.c-pronous.com.cn	
emaii: ta	ingjy@c-pronbus.com.cm	
□,=,+,1; ∠	004/9	
+	+*/	
#Include <c:\ #dofino LIDV</c:\ 	51/C51/LIVC/Feg52.11> TF unsigned abov	
#define UD1 #dofine UW(I E unsigned char	
#define U w (/KD unsigneu int	_*/
/* /* い下		-'/ */
/*	"と禍へだ、"日小八寺女田文主にへ,川川・「今久陸之内法	_*/
, sbit out0_7=I	?2^0; /* 定义 out0_7=P2.0=P16O0,输出指示灯 Q0-Q7 的输出控制,见实验板原理图 U16	, */
sbit out8_15=	-P2^1; /* 定义 out8_15=P2.1=P17O1,输出指示灯 Q8-Q15 的输出控制,见实验板原理图 U17	*/
sbit inled0_7	=P2^2; /* 定义 inled0_7=P2.2=P18IO0,输入指示灯 I0-I7 的输出控制,见实验板原理图 U18	*/
sbit inled8_1	5=P2^3; /* 定义 inled8_15=P2.3=P19IO1,输入指示灯 I8-I15 的输出控制,见实验板原理图 U19	*/
sbit in0_7=P2	2^4; /* 定义 in0_7=P2^4=P2.4=P12I0,输入键 I0-I7 的读控制,见实验板原理图 U12	*/
sbit in8_15=H	?2^5; /* 定义 in8_15=P2^5=P2.5=P13I1,输入键 I8-I15 的读控制,见实验板原理图 U13	*/
sbit inadd=P	2^6; /* 定义 inadd=P2^6=P2.6=P20AD,PROFIBUS 地址输入 AD0-AD7 的读控制,	
	/* 见实验板原理图 U20	*/
sbit REQ_IT	=P1^7; /* 定义 REQ_IT=P1^7=P1.7=P17S_CTS,输入,连接 OEM2-SE-REQ_IT, -	*/
	/* 见实验板原理图 CONNECT-16	*/
sbit S_RTS=l	P1^6; /* 定义 S_RTS=P1^6=P1.6=P17S_RTS,输入,连接 OEM2-SE-S_RTS,	*/
	/* 见实验板原理图 CONNECT-16	*/

sbit P15RES=P1^5;	/* 定义 P15RES=P1^5=RES,输入,连接 OEM2-SE-RES, /* 见实验板原理图 CONNECT-16	*/ */
/************* 定义:	di0_7:字节型,是输入键 I0-I7 当前值; ************************************	*/
/***** 定义:	di8 15:字节型.是输入键 I8-I15 当前值; ************************************	/
bdata UBYTE di0 7,di8	15; /* di0 7:字节型,输入键 I0-I7;di8 15:字节型,输入键 I8-I15	*/
sbit di0=di0 7^0;	/* 定义 di0=di0 7^0=输入键 I0	*/
sbit di1=di0_7^1;	/* 定义 di1=di0_7^1=输入键 I1	*/
sbit di2=di0_7^2;	/* 定义 di2=di0_7^2=输入键 I2	*/
sbit di3=di0_7^3;	/* 定义 di3=di0_7^3=输入键 I3	*/
sbit di4=di0_7^4;	/* 定义 di4=di0_7^4= 输入键 I4	*/
sbit di5=di0_7^5;	/* 定义 di5=di0_7^5=输入键 I5	*/
sbit di6=di0_7^6;	/* 定义 di6=di0_7^6=输入键 I6	*/
sbit di7=di0_7^7;	/* 定义 di7=di0_7^7=输入键 I7	*/
sbit di8=di8_15^0;	/* 定义 di8=di8_15^0=输入键 I8	*/
sbit di9=di8_15^1;	/* 定义 di9=di8_15^1=输入键 I9	*/
sbit di10=di8_15^2;	/* 定义 di10=di8_15^2=输入键 110	*/
sbit dill=di8_15^3;	/* 定义 dill=di8_15^3=输入键 111	*/ */
sbit di12=di8_15^4;	/* 正义 d12=d18_15^4=输入键 112	*/ */
sbit di13=di8_15^5;	/* 正义 dl13=dl8_15^5=制入键 113	*/ */
SDIU 0114=018_15^6;	/* 正义 0114=018_15*0=制入键 114 /* 空义 J:15_J:0 15/7_拾入键 115	*/ */
son all5=al8_15^7;	/*	*/
/*************************************	key0_7:字节型,是输入键 I0-I7 扫描读入值; ************************************	******/ ******/
bdata UBYTE kev0 7 kg	wy8 15·	1
sbit kev0=kev0 7^0:	/* 定义 kev0=kev0 7^0=输入键 I0	*/
sbit kev1=kev0 7^1:	/* 定义 kev1=kev0 7^1=输入键 I1	*/
sbit key2=key0 7^2;	/* 定义 kev2=kev0 7^2=输入键 I2	*/
sbit key3=key0_7^3;	/* 定义 key3=key0_7^3=输入键 I3	*/
sbit key4=key0_7^4;	/* 定义 key4=key0_7^4=输入键 I4	*/
sbit key5=key0_7^5;	/* 定义 key5=key0_7^5=输入键 I5	*/
sbit key6=key0_7^6;	/* 定义 key6=key0_7^6=输入键 I6	*/
<pre>sbit key7=key0_7^7;</pre>	/* 定义 key7=key0_7^7=输入键 I7	*/
sbit key8=key8_15^0;	/* 定义 key8=key8_15^0=输入键 I8	*/
sbit key9=key8_15^1;	/* 定义 key9=key8_15^1=输入键 I9	*/
sbit key10=key8_15^2;	/* 定义 key10=key8_15^2=输入键 I10	*/
sbit key11=key8_15^3;	/* 定义 key11=key8_15^3=输入键 I11	*/
sbit key12=key8_15^4;	/* 定义 key12=key8_15^4=输入键 I12	*/
sbit key13=key8_15^5;	/* 定义 key13=key8_15^5=输入键 113	*/
sbit key14=key8_15^6;	/* 定义 key14=key8_15^6=锕入键 114	*/ */
sbit key15=key8_15 ^{//} ;	/* 正义 key15=key8_15///=输入键 115	*/
/***************** 定义: sbit P00=P0^0;	P00-P07 对应 P0 口的 P0.0-P0.7 *********************************/	
sbit P01=P0^1;		
sbit P02=P0^2;		
sbit P03=P0^3;		
sbit P04=P0^4;		
sbit P05=P0^5;		
sbit P00=P0^0;		
SUIL PU/-PU//;		
/***** PP00-I	PP07 是上面 P00-P07 的中间变量 *******************************/	
bdata UBYTE PP0;		
sbit PP00=PP0^0;		
sbit PP01=PP0^1;		
sbit PP02=PP0^2;		
sbit PP03=PP0^3;		
sbit PP04=PP0^4;		
sbit PP05=PP0^5;		
sbit PP06=PP0^6;		
sbit PP07=PP0^7;		

/**** 定义: AB status:字节型,报文类型及接口状态,见《PB-OEM2-SE 产品手册》第四章、三、数据交换 ******/ bdata UBYTE AB_status; sbit AB sD0=AB status^0; /**** AB_sD0=AB_status^0:用户参数更新标记 ****/ sbit AB sD7=AB status^7; /**** AB sD7=AB status⁷:报文格式 ****/ /**** 定义: req com:字节型,请求数据命令,见《PB-OEM2-SE 产品手册》第四章、三、数据交换 *******/ bdata UBYTE req_com; /**** req_comD0=req_com^0:请求数据/参数命令 ****/ sbit req_comD0=req_com^0; /**** req_comD1=req_com^1:清参数命令 ****/ sbit req_comD1=req_com^1; /*-----*/ /* 输入键、指示灯等设备变量定义完毕 */ /*_____*/ /*-/* 以下是有关输入键、指示灯控制子程序,用户了解功能即可 */ /*---------*/ void steout0_7(UBYTE v) { out0_7=0; **P0=v;** P0=v: P0=v; **P0=v:** out0_7=1; out0_7=0; } void steout8_15(UBYTE v) { out8_15=0; **P0=v; P0=v: P0=v: P0=v:** out8 15=1: out8_15=0; } void steinled0_7(UBYTE v) { inled0_7=0; **P0=v;** P0=v; P0=v; **P0=v:** inled0_7=1; inled0_7=0; } void steinled8_15(UBYTE v) { inled8_15=0; P0=v; P0=v; P0=v; P0=v; inled8_15=1; inled8 15=0; }

```
UBYTE rd_address()
UBYTE data x:
P0=0xff;
inadd=0;
x=P0;
x=P0;
x=P0;
x=P0;
inadd=1;
return(x);
}
void scanin()
ł
P0=0xff;
in0_7=0;
PP0=P0;
PP0=P0:
PP0=P0;
PP0=P0:
if (key0_7!=PP0)
ł
 if ((PP00==1)&&(key0==0)) { di0=~di0;};
 if ((PP01==1)&&(key1==0)) { di1=~di1;};
 if ((PP02==1)&&(key2==0)) { di2=~di2;};
 if ((PP03==1)&&(key3==0)) { di3=~di3;};
 if ((PP04==1)&&(key4==0)) { di4=~di4;};
 if ((PP05==1)&&(key5==0)) { di5=~di5;};
 if ((PP06==1)&&(key6==0)) { di6=~di6;};
 if ((PP07==1)&&(key7==0)) { di7=~di7;};
 key0_7=PP0;
};
in0_7=1;
inled0 7=0;
P0=di0_7;
P0=di0_7;
P0=di0_7;
P0=di0_7;
inled0_7=1;
inled0_7=0;
P0=0xff;
in8_15=0;
PP0=P0;
PP0=P0;
PP0=P0;
PP0=P0;
if (key8_15!=PP0)
{
 if ((PP00==1)&&(key8==0)) { di8=~di8;};
 if ((PP01==1)&&(key9==0)) { di9=~di9;};
 if ((PP02==1)&&(key10==0)) { di10=~di10;};
 if ((PP03==1)&&(key11==0)) { di11=~di11;};
 if ((PP04==1)&&(key12==0)) { di12=~di12;};
 if ((PP05==1)&&(key13==0)) { di13=~di13;};
 if ((PP06==1)&&(key14==0)) { di14=~di14;};
 if ((PP07==1)&&(key15==0)) { di15=~di15;};
 key8_15=PP0;
};
in8_15=1;
```

inled8_15=0; mied8_15=0; P0=di8_15; P0=di8_15; P0=di8_15; P0=di8_15; inled8_15=1; inled8_15=0; }

=====================================	=====================================	
↑	在序	*/
void main ()		,
UBYTE data i,j,k,x,y,count,frist;		
UWORD data nw,nn;		
UBYTE xdata * data p;		
UBYTE data address;	/*从站地址	*****
JBYTE data re_er;		*
JBYTE data br_num;	/*	
JBYTE data error;	/*	،، م
JBYTE xdata rebox[113];	/*	*****
JBY IE XOALA LEDOX[115]; IDVTE udata ph. out[112];	/*中山久还绂/叶 /*	*, */
JBY IE Xdata pb_out[112]; IPVTE vdoto ph_in[112];	/*	/*
BTIE xuata pp_m[112]; BVTE vdata user .nrm[41]:	/*PROFIBUS 捆八致站区 /*PROFIBUS 田白参数区	*
IBVTE data data in len:	/*数据交换输入(实验板发送)数据长度	
TBYTE data dil:	/	*
JBYTE data data out len:	/*数据交换输出(实验板接收)数据长度	*
JBYTE data dol:	/*不包括校验和的 data out len 长度	·,
UBYTE data user prm len:	/*用户参数长度	
JBYTE data se_er;	/*OEM2-SE 接收数据错误类型号	
'1=0xff;		
' 0=0;		
0.0.0		
P2=0xf0;		
22=0xf0; li0_7=di8_15=0;		
P2=0xf0; li0_7=di8_15=0; steout0 7(0):	/*输出指示灯 O0-O7 清零	,
P2=0xf0; li0_7=di8_15=0; steout0_7(0); steout8_15(0);	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零	
P2=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0);	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零	 *
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0);</pre>	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零	
P2=0xf0; li0_7=di8_15=0; xteout0_7(0); xteout8_15(0); xteinled0_7(0); tteinled8_15(0); *	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零	s s s s
² 2=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); * * 以下是实验板上电后,检测 P8	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 ====================================	 * */ */
22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); * * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 ====================================	
P2=0xf0; li0_7=di8_15=0; iteout0_7(0); iteout8_15(0); iteinled0_7(0); iteinled8_15(0); * * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 *	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 /*输入键指示灯 ILED8-ILED15 清零 59V51RD2 单片机内部 RAM 的一段输出演示程序; 氡 SE 通信无关,用户可跳过此段程序。	
2=0xf0; li0_7=di8_15=0; teout0_7(0); teinled0_7(0); teinled8_15(0); * * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 *	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 输入键指示灯 ILED8-ILED15 清零 	
22=0xf0; li0_7=di8_15=0; teout0_7(0); teinled0_7(0); teinled8_15(0); * * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * ==0; ==1;	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 99V51RD2 单片机内部 RAM 的一段输出演示程序; 私 SE 通信无关,用户可跳过此段程序。	
22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled8_15(0); * * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * ==0; ==1; :=1;	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 ====================================	
22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); ** & 以下是实验板上电后,检测P8 * * * * *	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 	
22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); * * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * p=0; c=1; c=1; for (j=0;j<32;j++) { for (j=0;j<24:j++)	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 ====================================	
22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); * * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * ==0; :=1; :=1; or (j=0;j<32;j++) { for (i=0;i<24;i++) {	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 	* * */ */ */
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teinled0_7(0); teinled8_15(0); ** * 以下是实验板上电后,检测P8 * 这段程序与 PROFIBUS 接口板 * ==0; :=1; :=1; or (j=0;j<32;j++) { for (i=0;i<24;i++) { *(p+i)=0xaa;</pre>	/*输出指示灯 Q0-Q7 清零 /*输入指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 ====================================	
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teinled0_7(0); teinled8_15(0); *====================================</pre>	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 39V51RD2 单片机内部 RAM 的一段输出演示程序; 就 SE 通信无关,用户可跳过此段程序。	
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); ** * 以下是实验板上电后,检测P8 ** * 这段程序与 PROFIBUS 接口板 * =0; :=1; :=1; or (j=0;j<32;j++) { for (i=0;i<24;i++) { *(p+i)=0xaa; }; y=0;</pre>	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 ?*输入键指示灯 ILED8-ILED15 清零 39V51RD2 单片机内部 RAM 的一段输出演示程序; 截 SE 通信无关,用户可跳过此段程序。	
22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); ** * 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * =0; =1; =1; or (j=0;j<32;j++) { for (i=0;i<24;i++) { *(p+i)=0xaa; }; y=0; for (i=0;i<24;i++)	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 39V51RD2 单片机内部 RAM 的一段输出演示程序; 数 SE 通信无关,用户可跳过此段程序。	
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); ** * 以下是实验板上电后,检测P8 ** ** ** b=0; c=1; c=1; c=1; for (i=0;i<24;i++) {</pre>	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 /*输入键指示灯 ILED8-ILED15 清零 39V51RD2 单片机内部 RAM 的一段输出演示程序; 数 SE 通信无关,用户可跳过此段程序。	
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); ** 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * ==0; t=1; t=1; or (j=0;j<32;j++) { for (i=0;i<24;i++) { *(p+i)=0xaa; }; y=0; for (i=0;i<24;i++) { count=*(p+i);</pre>	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 /*输入键指示灯 ILED8-ILED15 清零 39V51RD2 单片机内部 RAM 的一段输出演示程序; 数 SE 通信无关,用户可跳过此段程序。	
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); ** 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * =0; t=1; t=1; or (j=0;j<32;j++) { for (i=0;i<24;i++) {</pre>	/*输出指示灯 Q0-Q7 清零 /*输出指示灯 Q8-Q15 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 	
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); ** 以下是实验板上电后,检测 P8 * 这段程序与 PROFIBUS 接口板 * =0; t=1; t=1; t=1; for (i=0;i<24;i++) { for (i=0;i<24;i++) { r(p+i)=0xaa; }; y=0; for (i=0;i<24;i++) { count=*(p+i); if (count!=0xaa) { y=1; }; };</pre>	/*输出指示灯 Q0-Q7 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 	
<pre>22=0xf0; li0_7=di8_15=0; teout0_7(0); teout8_15(0); teinled0_7(0); teinled8_15(0); *</pre>	/*输出指示灯 Q0-Q7 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 /*输入键指示灯 ILED8-ILED15 清零 99V51RD2 单片机内部 RAM 的一段输出演示程序; & SE 通信无关,用户可跳过此段程序。	
<pre>P2=0xf0; li0_7=di8_15=0; iteout0_7(0); iteout8_15(0); teinled0_7(0); teinled8_15(0); *</pre>	/*输出指示灯 Q0-Q7 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零	
P2=0xf0; li0_7=di8_15=0; steout0_7(0); steout8_15(0); steinled0_7(0); steinled8_15(0); *	/*输出指示灯 Q0-Q7 清零 /*输入键指示灯 ILED0-ILED7 清零 /*输入键指示灯 ILED8-ILED15 清零 /*输入键指示灯 ILED8-ILED15 清零 20951RD2 单片机内部 RAM 的一段输出演示程序: SE 通信无关,用户可跳过此段程序。	

```
case 1:{ steout0_7(x);};break;
      case 2:{ steout8_15(x);};break;
      case 3:{ steinled0_7(x);};break;
      case 4:{ steinled8_15(x);};break;
      default:break;
     };
    if (x<255) {x=x*2+1;}else{x=1;k++;};
  }
 else
  {
   for (;;)
    ł
     for (nw=0;nw<6000;nw++)
      ł
       steout0 7(0):
       steout8 15(0);
       steinled0_7(0);
       steinled8_15(0);
      };
     for (nw=0;nw<6000;nw++)
      {
       steout0_7(0xff);
       steout8_15(0xff);
       steinled0_7(0xff);
       steinled8_15(0xff);
      };
    };
  };
 p=p+24;
 for (nw=0;nw<8000;nw++)
  {};
};
/*
   实验板输出演示程序结束。
                                                                                        */
                                      /*-----输出指示灯 Q0-Q7 清零
steout0_7(0);
steout8_15(0);
                                      /*-----输出指示灯 O8-O15 清零
                                                                                      .....*/
steinled0 7(0);
                                      /*-----输入键指示灯 ILED0-ILED7 清零
                                                                                      ----*/
steinled8_15(0);
                                     /*-----输入键指示灯 ILED8-ILED15 清零
                                                                                     .....*/
/*=
                            _____
                                                                              ======*/
   清串口输入缓冲区 trbox[i],清串口输出缓冲区 rebox[i]
/*
                                                                                         */
/*_____
                                                                                    ____*/
for (i=0;i<113;i++)
{
 rebox[i]=trbox[i]=0;
};
for (i=0;i<112;i++)
{
 pb_in[i]=pb_out[i]=0;
};
                     /*------读实验板上拨码开关,作为站号------*/
address=rd_address();
                                                                               */
/*
                         以下是串口初始化程序
/*____
                                                                             .*/
                     /*------实验板波特率选择 5, 115.2K------*/
br_num=5;
switch (br_num)
                    ł
    case 1:{ TH2=0x0ff; TL2=0x0a0; RCAP2H=0x0ff; RCAP2L=0x0a0; }; break;
                                                                         /*---9.6K----*/
    case 2:{ TH2=0x0ff; TL2=0x0d0;RCAP2H=0x0ff;RCAP2L=0x0d0;};break;
                                                                         /*---19.2K--*/
    case 3:{ TH2=0x0ff; TL2=0x0e8;RCAP2H=0x0ff;RCAP2L=0x0e8;};break;
                                                                        /*---38.4K--*/
    case 4:{ TH2=0x0ff; TL2=0x0f0;RCAP2H=0x0ff;RCAP2L=0x0f0;};break;
                                                                        /*---57.6K--*/
```

<pre>case 5:{ TH2=0x0ff; 7 case 6:{ TH2=0x0ff; 7 case 7:{ TH2=0x0ff; 7 default:break; }; /*</pre>	TL2=0x0f8;RCAP2H=0x0ff;RCAP2L=0x0f8;};break; TL2=0x0fc;RCAP2H=0x0ff;RCAP2L=0x0fc;};break; TL2=0x0fe;RCAP2H=0x0ff;RCAP2L=0x0fe;};break;	/*115.2K-*/ /*230.4K-*/ /*460.8K-*/
, /* 注意:本例基于 PHILI	P P89V51RD2 单片机:	*/
/* 波特率=14745600/16/(65536-T)	*/
/* 各种波特率 9600: T=0	0xffa0, 19.2K: T=0xffd0, 38.4K: T=0xffe8,	*/
/* 57.6K: T=0x	xiii0, 115.2K:T=0xiii8, 230.4k:T=0xiiic, 460.8k=0xiiie	• */
/*		*/
$\frac{P(UN=0)}{T^2(ON=0)^2}$	/* DCI K_1, TCI K_1, TD2-1, C/T2-0	*/
120011-0234;	/·····KCLK-1; ICLK-1; IK2-1; C/12-0 ·····	
/*		*/
, /*串口设置成模式 3 即:1 /* 0xd0=11010000B; SM0	└ 个启始位、8 个数据位、1 个偶校验位、1 个停止位; SM1=11; SM2=0; REN=1;TB8=0; RB8=0; TI=0; RI=0	*/ */
/* SCON=0x0d0;		*/
/* /*	 串口初始化程序结束	*/ */
/*=====================================		·=====*/
/*		=====*/
/*	形成接口板的初始化报文	*/
/*	rebox[]是串口发送数组,格式见《SE 手册》	*/
/*		*/ */
data_in_len= $48+1+1$;	/*	*/ */
user_prm_len=0;	/* 用尸奓蚁长度=0 /*	*/ */
data_out_ten=40+1+1; trbox[0]=oddross:	/* 制入致活取人大浸=48+1+1=30; /* BVTEA-11 計計###	*/ */
trbox[0]=auuress;	/* BITE0-从珀地址 /* BVTF1-ID 号宫位	*/
trbox[1]=0x00,	/ BTTET=ID 与周位 /* BVTF2=ID 是任位	/ */
u box[2]=0x010,		,
/*=====================================		=====*/
/*	以下形成 I/O 配置数据	*/
/* 本例: I/O 配置数据长图	度=6;I/O 配置数据:0x1f,0x2f,0x1f,0x2f,0x1f,0x2f	*/
/* 共: 48bytes input+48by	vtes output	*/
/*		*/ */
$\frac{\text{UrDOX}[\mathcal{S}]=0}{\text{trbox}[\mathcal{A}]=0}$	/*BYIE3=I/U 能直致抗大度=0 /* DVTE4_I/O 配罢数据0-1f	*/ */
1100x[4]=0x11; trbov[5]=0v2f;	/*DIIL4-I/O 癿直双 炻: 0XII /* BVTE5-I/O 配罟数据 . 0v2f	·/ */
trbox[5]=0x21, trbox[6]=0x1f	/*BTTE3=IO 癿直奴 诏: 0221 /*BVTE6-I/O 配置数据. 0v1f	*/
trbox[7]=0x2f:	/*BYTE7=I/O 配置数据: 0x2f	, */
trbox[8]=0x1f:	/*BYTE8=I/O 配置数据: 0x1f	, */
trbox[9]=0x2f;	/*BYTE9=I/O 配置数据: 0x2f	*/
trbox[24]=data_in_len;	/*BYTE24=data_in_len 输入数据报文长度	*/
trbox[25]=data_out_len;	/*BYTE25=data_out_len 输出数据报文长度	*/
trbox[26]=user_prm_len;	/*BYTE26=user_prm_len 用户参数长度	*/
A.		
X=U; for (:_0.: <49.:)		
10F (I=0;I<40;I++)		
ι x−x+trbox[i]•	/*	*/
};	· 3123210	,
trbox[48]=x;	/*置校验和	*/
L - J /		
/*		=====*/
/* 发送接口板的初始	台化报文、接收接口板的回答,初始不成功继续发初始化报文	ζ */
/*		*/ */
y=0;	/*	*/
winite (y==0) {		
REN=0:		
for (i=0;i<49;i++)	/* 发送一次初始化报文共 49 字节	*/

{ **TI=0:** ACC=trbox[i]; TB8=P; /*---- 形成偶校验位 ----*/ SBUF=trbox[i]; while (TI==0) {}; }; /*------发送完毕、等待接收回答------*/ **RI=0;** REN=1; /*---- 允许接收*/ re_er=0; /*---- re_er 偶校验错标志 ----*/ for (j=0;j<49;j++) /*---- 接受回答报文共 49 个字节 ----*/ { nw=0: while (RI==0) {}; /*---- 等待接收一个字符 -----*/ ACC=SBUF; -----*/ if (P!=RB8) { re_er=1;}; /*---- 偶校验错,置偶校验错标志 rebox[j]=SBUF; /*---- 接收一个字符送串口输入缓冲区 ----*/ **RI=0;** }; /*----- 接收回答报文结束*/ ----/*------ 以下分析接口板 SE 的回答报文 -----------------------*/ /*---- 没有字符偶校验错? -----*/ if (re_er==0) { x=0: for (i=0;i<48;i++) /*---- 求校验和*/ { x=x+rebox[i]; }; if (x==rebox[48]) /*---- 校验和正确? {*/ /*---- 返回的波特率号是 5? if (rebox[0]==br_num)*/ { /*---- 初始化报文错误号 N=0? if (rebox[1]==0)*/ { x=0: /*---- byte2~byte47 都是 0xaa? for (i=2;i<48;i++)*/ { if (rebox[i]!=0xaa) { x=1; }; }; if (x==0) { /*---- byte2~byte47 都是 0xaa,初始化成功,y=1,准备退出 while (y==0) ----*/ y=1; } else {x=0x11;}; /*---- byte2~byte47 不都是 0xaa! 错误标记=0x11; ----*/ ł else /*----初始化报文错误号 N<>0,错误标记=返回错误号 {**x=rebox**[1];}; ----*/ } else {x=0x77;}; /*---- 返回的波特率号不是 5, 错误标记=0x77; ----*/ } else {**x=0xff;**}; /*---- 校验和不正确正确, 错误标记=0xff; ----*/ } else { x=0xee; /*---- 发生了字符偶校验错 错误标记=0xee; ----*/ };

if (y==0) /*---- 初始化失败 ----*/ ł for (nw=0:nw<10000:nw++) { /*---- 显示返回波特率号 ----*/ steout0_7(br_num); /*---- 显示返回错误号 ----*/ steout8_15(x); }; }; }; /*---- 如果 y==0,返回 while (y==0) ----*/ ======*/ /* 实验板对 PROFIBUS 接口板初始化成功,进数据交换状态的无限循环中 */ /* */ trbox[]/rebox[]是串口发送/接收数组,发送/接收格式见《SE 手册》 /*_____ -----._*/ dil=data in len-1: /*---- data_in_len=50, dil=49 ----*/ dol=data out len-1; /*---- data out len==50, dol==49 ----*/ for (i=0;i<data_in_len;i++) { /*---- 清发送、接收缓冲区 trbox[i]=0; ----*/ }; for (i=0;i<data_out_len;i++)</pre> { /*---- 清发送、接收缓冲区 rebox[i]=0; ----*/ }; /*== for (;;) /*---- 没有用户参数,请求数据命令永远是 req_com=0,只请求 A 型报文 ------*/ trbox[0]=req_com=0; /*---- di0_7 是输入键输入 I0~I7 当前值,作为 PROFIBUS 输入的第1字节 ------*/ trbox[1]=di0_7; /*---- di8_15 是输入键输入 I8~I15 当前值,作为 PROFIBUS 输入的第 2 字节 -----*/ trbox[2]=di8_15; for (i=3;i<dil;i++)</pre> ł /*---- 其它 PROFIBUS 输入数据(3~48)本例定为 0x5a -----*/ trbox[i]=0x5a; }; x=0; for (i=0;i<dil;i++)</pre> /*---- 求校验和 -----*/ { x=x+trbox[i]; **};** trbox[dil]=x; /*---- 置校验和 ----*/ REN=0; for (i=0;i<data_in_len;i++) /*---- 发送数据交换报文共 50 字节*/ { **TI=0:** ACC=trbox[i]; TB8=P; /*---- 形成偶校验位 -----*/ SBUF=trbox[i]; while (TI==0) {}; }; _*/ /*----- re_er 偶校验错标志 _____*/ re_er=0; **RI=0;** /*----- 允许接收 REN=1:*/ for (j=0;j<data_out_len;j++)</pre> /*----- 等待接收接口板的回答数据交换报文共 50 字节 -----*/ { /*----- 等待接收一个字符 while (**RI**==0) {}; -----*/ ACC=SBUF; /*----- 偶校验错,置偶校验错标志 -----*/ if (P!=RB8) {re_er=1;}; rebox[j]=SBUF; /*----- 接收一个字符送串口输入缓冲区 -----*/

```
RI=0;
 };
              =====接收完毕==
/*-
                                                                              =*/
if (re_er==0)
                              /*----- 无偶校验错
                                                                             -----*/
 {
     x=0;
    for (i=0;i<dol;i++)</pre>
     {
                             /*----- 求校验和
       x=x+rebox[i];
                                                                            ----*/
     };
    if (x==rebox[dol])
     {
        y=dol-1;
                            /*----- y=49-1=48,PROFIBUS 输出数据长度
                                                                             -----*/
        for (i=1;i<dol;i++)</pre>
         {
                         /*----- 接收到的 48 个 PROFIBUS 输出数据送数据区 pb_out[0~47] -----*/
         pb_out[i-1]=rebox[i];
         };
     }
    else
     {
      error=0x81;
                             /*----- 校验和错
                                                                            .....*/
     };
 }
else
 {
   error=0x83;
                              /*----- 偶校验错
                                                                             -----*/
 };
/*=
                                                .____*/
/*_____*/
scanin();
                           /*----- 输入键扫描
                                                                           -----*/
                          /*----- PROFIBUS 输出数据第一个字节送 Q0-Q7 显示
steout0_7(pb_out[0]);
                                                                           .....*/
                          /*----- PROFIBUS 输出数据第一个字节送 Q0-Q7 显示
steout8_15(pb_out[1]);
                                                                           ----*/
};/*for(;;)*/
```

(7) 例1的GSD 文件

}

;GSD 文件: OEM2-B1 - 48DI/48DO /SPC3 鼎实创新科技有限责任公司 ;产品型号: PB-OEM2-SE 2004年9月 V1.0 ;版本 : OEM2-B1.GSD ;文件名: :== #Profibus_DP ; < Prm-Text-Def-List> ; < Ext-User-Prm-Data-Def-List> ; <Unit-Definition-List> GSD_Revision=2 Vendor_Name="DS FieldBus Ltd. Co." ; 公司名,可按用户名修改 Model_Name="PB-OEM2-SE-B1" ;模块名称,也是组态时该产品的名称,组态软件如 STEP 7, COM PROFIBUS Revision="V1.0 " Ident_Number=0x06FA ;ID 号,必须与初始化报文一致 Protocol_Ident=0 Station_Type=0 Hardware_Release="A1.0 " Software_Release="Z1.0 " 9.6_supp=1 19.2_supp=1 93.75_supp=1 187.5_supp=1 500_supp=1

45.45_supp=1 1.5M_supp=1 3M_supp=1 6M_supp=1 12M_supp=1 MaxTsdr_9.6=60 MaxTsdr_19.2=60 MaxTsdr_45.45=250 MaxTsdr_93.75=60 MaxTsdr_187.5=60 MaxTsdr_500=100 MaxTsdr_1.5M=150 MaxTsdr_3M=250 $MaxTsdr_6M{=}450$ MaxTsdr_12M=800 Implementation_Type="spc3" ; 图标文件, 用户可以自制图标, 缺省 Bitmap_Device="SE_B" ; Slave-Specification: OrderNumber="pb-oem2-s" ; 产品序列号, 可按用户名修改 Freeze_Mode_supp=1 Sync_Mode_supp=1 Auto_Baud_supp=1 Fail_safe=0 Min_Slave_Intervall=1 Max_Diag_Data_Len=6 User_Prm_Data_Len=0 Modular_Station = 0Modul_Offset=0 Slave_Family=3@TdF@PB-OEM2 ;组态中的分类名 ; <Module-Definition-List> Module="48 Byte In, 48 Byte Out "0x1f,0x2f,0x1f,0x2f,0x1f,0x2f ;I/O 配置,必须与初始化报文一致 EndModule

例 2. 监测 SE 接口状态的工作模式

例 2 是在"例 1"基础上,在串口通信中增加监测 SE 接口状态功能。检测 SE 接口状态的通信是一种半握手通信方式,可以避免死等、接收长度错误等故障,提高通信效率。

虽然"例 2"与"例 1"在技术指标、程序结构、GSD 文件方面绝大部分都是一致的,但考虑到有些读者可能没有阅读"例 1"而直接阅读"例 2",因此本《手册》将例 1 的程序也重复列出,只是将与这一部分重复文字用斜体字表示,如果读者不需要可越过这些段落。

(1) 技术参数:

PROFIBUS 输入/输出:	48 字节输入/48 字节输出;
用户参数:	无;
串口波特率:	115.2K
通信模式:	检测 SE 接口状态条件;

(2) SE 初始化报文

1/0 配置数据长度 CFG_LEN=6;
1/0 配置数据为: 0x1f, 0x2f, 0x1f, 0x2f, 0x1f, 0x2f
用户参数长度 User_Prm_Data_Len=0;
接收数据长 SD_in_len ≥48+1+1=50
发送数据长 SD out len ≥48+1+1=50



实验板发送初始化报文

SE 接口板回答的初始化成功报文

接口板回答的初始化失败报文

(3) 数据交换状态下实验板与 SE 接口的数据报文

实验板不使用用户参数功能(即:初始化报文中 User_Prm_Data_Len=0), 永远发请求 PROFIBUS 数据命令 req_com=0;接口板一定回答"输出数据报文A"。见图4-2-1,图4-2-2



图 4-2-1 实验板发 — 输入数据报文 req_com=0

(4) SE 接口状态信号 REQ_IT 与 S_RTS/P17 的时序图

以下是 SE 接口状态信号 REQ_IT 与 S_RTS/P17 的时序图,读者在《PB-OEM2-SE 产品手册》中亦可找到。通过 REQ_IT 状态可以检测到 SE 接口处在"等待初始化状态"还是"数据交换状态"。通过 S_RTS 状态可以检测到 SE 接口发送接收状态;

S_RTS=0: SE 接口等待接收;

S_RTS=1: SE 接口处在"接收完毕、分析报文、正在发送"状态。



图 4-2-3 SE 接口状态信号 REQ_IT 与 S_RTS/P17 的时序图

图 4-2-2 SE 接口板答—输出数据报文格式A

(5) 实验板初始化 SE 接口板的过程





(6) 数据交换状态下实验板带监测接口状态功能工作模式

见图 4-2-5, 黑色斜体部分与"例 1"相同, 增加部分用蓝色正体标出。



图 4-2-5 数据交换状态下用户模板的程序流程(监测接口状态条件、没有用户参数的工作模式)

(7) 实验板程序清单

例 2 的程序与例 1 相比,只增加了监测 SE 接口状态信号 REQ_IT 与 S_RTS/P17,其它大部分程序

一致;以下增加部分是蓝色五号字,与例1相同部分仍然是黑色斜体小五号字。

/*+	+	
│文件名称: ≶	实验板板串口通信程序──监测 SE 接口状态的工作模式	
版 本: V	/1.1	
制作单位: 1	化京鼎实创新科技公司	
版权许可:		
│网址: v	www.c-profibus.com.cn	
email: ta	angjy@c-profibus.com.cm	
日期: 2	2004/9	
+	+*/	
<i>#include <c:\< i="">:</c:\<></i>	51\c51\INC\reg52.h>	
#define UBY1	TE unsigned char	
#define UWO	DRD unsigned int	
/*========		:====*/
/* Ø7 /*	^下 是输入键、指示灯等设备变量定义,用户可以越过阅读 	*/ */
sbit out0_7=P	P2^0; /* 定义 out0_7=P2.0=P16O0,输出指示灯 Q0-Q7 的输出控制,见实验板原理图 U16	*/
sbit out8_15=	:P2^1; /* 定义out8_15=P2.1=P1701,输出指示灯Q8-Q15 的输出控制,见实验板原理图U12	7*/
sbit inled0_7=	=P2^2; /* 定义 inled0_7=P2.2=P18IO0,输入指示灯 IO-I7 的输出控制,见实验板原理图 U18	*/
sbit inled8_15	5=P2^3; /* 定义inled8_15=P2.3=P19101,输入指示灯18-115 的输出控制,见实验板原理图U	19*/
sbit in0_7=P2	2^4; /* 定义in0_7=P2^4=P2.4=P12I0,输入键I0-I7 的读控制,见实验板原理图U12	*/
sbit in8_15=P	P2^5; /* 定义in8_15=P2^5=P2.5=P1311,输入键18-115 的读控制,见实验板原理图U13	*/
sbit inadd=P2	2^6; /* 定义 inadd=P2^6=P2.6=P20AD,PROFIBUS 地址输入 AD0-AD7 的读控制,	
	/* 见实验板原理图 U20	*/
sbit REQ_IT=	=P1^7; /* 定义	*/
	/* 见实验板原理图 CONNECT-16	*/
sbit S_RTS=P	P1^6; /* 定义 S_RTS=P1^6=P1.6=P17S_RTS, 输入,连接 OEM2-SE-S_RTS,	*/
	/* 见实验板原理图 CONNECT-16	*/
sbit P15RES=	=P1^5; /* 定义 P15RES=P1^5=RES, 输入,连接 OEM2-SE-RES,	*/
	/* 见实验板原理图 CONNECT-16	*/
/*********	****	
/********	**** 定义: di8_15:字节型,是输入键18-115 当前值; ****************************/	
bdata UBYTE	E di0_7,di8_15; /* di0_7:字节型,输入键10-17;di8_15:字节型,输入键18-115*/	
sbit di0=di0_7	7^0; /* 定义 di0=di0_7^0=输入键 I0*/	
sbit di1=di0_2	7^1; /* 定义 di1=di0_7^1= 输入键 I1	
sbit di2=di0_7	7^2; /* 定义 di2=di0_7^2=输入键 I2*/	
sbit di3=di0_7	7^3; /* 定义di3=di0_7^3=输入键I3*/	
sbit di4=di0_7	7^4; /* 定义di4=di0_7^4=输入键I4*/	
sbit di5=di0_7	7^5; /* 定义di5=di0_7^5=输入键15*/	
sbit di6=di0_7	7^6; /* 定义di6=di0_7^6=物入键16*/	
sbit di7=di0_7	7^7; /* 定义 di7=di0_7^7= 输入键 17*/	
sbit di8=di8_1	15^0; /* 定义 di8=di8_15^0= 綱入鍵 18*/	
sbit di9=di8_1	15^1; /* 定义 di9=di8_15^1= 綱入鍵 19*/	
sbit di10=di8_	_15^2; /* 定义 di10=di8_15^2= 铜入键 110*/	
<i>sbit di11=di8_</i>	_15^5; /* 正义 dill=di8_15^5= 制入键 111*/	
sbit di12=di8_	_15^4; /* 正义 di12=di8_15^4=物入键 112*/	
<i>sbit al13=al8_</i>	_13^5; /* 正义 all3=all8_13^5=棚八键 113*/	
SDIT al14=al8_	_13^0; /* 正义 all4=all8_13^0=糊入罐 114*/	
sou al15=al8_	_13 [^] ; /* 正义 all5=all8_13 [^] /=棚八键 l15*/	
/**********	**** 宁义、1	
/***********	にへ: ヘヒyv_/・テ /注,に御八焼1º=1/ 1/11法八追, ***** 定义. Lova 15・空井利 旱輪入鐘12-115 幻媒凄入信. ************************************	
hdata IIRVTL	たみ: ヘモy0_1J.ナ //空,足砌八斑10*11J 川畑佚八道, ************************************	
shit kay0-kay	μ reyu_1, reyu_13, μ 7^0· /* rev kow0-kow0 7^0- $ab \lambda Φ$ 10 */	
sou neyo-key	$w_{-}, w_{-}, w_{-},$	
shit kov2-kov	$y_{0} = 7^{-1}, (2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 $	
shit kov3-kov	$\frac{1}{2} \frac{1}{2} \frac{1}$	
shit kov4-kov	$n_{1} = n_{1} = n_{1$	
shit kov5-kov	0^{-7} , 2^{-7} , $2^{$	
son neys-ney		
鼎实科技

sdu keyo=key0_/^o;	/* 定义 key6=key0_7^6= 输入键 I6	*/
sbit key7=key0_7^7;	/* 定义 key7=key0_7^7= 输入键 I 7	*/
sbit key8=key8 15^0:	/* 定义kev8=kev8 15^0= 输入键18	*/
shit $kay0-kay8$ 15 1 .	/* 定义kay0-kay8 15^1- 榆入键10	*/
$sbu key = key 0_{15} 1;$	/ $(* 宁 \psi how 10 - how 0 + 15 A - 40) (# 1)$	*/
<i>sou key10-key8_13*2;</i>		
sbit key11=key8_15 ³ ;	/* 正义 key11=key8_15/3= 御八键 111	*/
sbit key12=key8_15^4;	/* 定义 key12=key8_15^4= 输入键 112	*/
<i>sbit key13=key8_15^5;</i>	/* 定义 key13=key8_15^5= 输入鍵 113	*/
<i>sbit kev14=kev8</i> 15^6:	/* 定义kev14=kev8 15^6=输入键114	*/
sbit key15=key8_15^7;	/* 定义 key15=key8_15^7= 输入键115	*/
/************* 定义: P00-P	P07	****/
$\sum_{n=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{i=1}^{n} \sum_{i=1}^{n} \sum_{i$		7
SUU = 100 - 1000,		
sou Poi=Po ¹ ;		
<i>sbit P02=P0</i> ^2;		
sbit P03=P0^3;		
sbit P04=P0^4;		
<i>sbit P05=P0^5;</i>		
sbit P06=P0^6:		
sbit P07=P0^7;		
/****************** PP00_PP07	《上面P00-P07 的山间恋暑 ************************************	****
, 1100-110/ Æ hdata URVTE DDA.		/
baala UBITE FF0;		
sbit PP00=PP0^0;		
sbit PP01=PP0^1;		
sbit PP02=PP0^2;		
sbit PP03=PP0^3:		
shit PP04-PP0^4		
shit PP05_PP0^5.		
sources = 110 S,		
sbit PP06=PP0^6; sbit PP07=PP0^7;		
/**** 定义: AB_status:字节型	! ,报文类型及接口状态,见《PB-OEM2-SE 产品手	研》第四章、三、数据交换 *******/
bdata UBYTE AB_status;		
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0;	/**** AB_sD0=AB_status^0:用户参数更新标记	****/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7;	/**** AB_sD0=AB_status^0: 用户参数更新标记 /**** AB_sD7=AB_status^7: 报文格式	****/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 宏义: rea_com: 字书积.	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令。	****/ ****/ *音. 二. 新程交选 ******/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字书型, bdata UBYTE rea_com;	/**** AB_sD0=AB_status^0: 用户参数更新标记 /**** AB_sD7=AB_status^7: 报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四	****/ ****/ 章、三、数据交换 ******/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字书型, bdata UBYTE req_com; sbit rea_comD0=rea_com^0;	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** reg_comD0=reg_com^0:请求数据(参数命	****/ ****/ <i>章、三、数据交换</i> ******/ 全 ****/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字书型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1;	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命	****/ ****/ 章、三、数据交换 *******/ 令 ****/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1;	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令	****/ ****/ * 章、三、数据交换 ******/ 令 ****/ ****/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com: 字书型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令	****/ ****/ 章、三、数据交换 ******/ 令 ****/ ****/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com: 字书型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕	*****/ ****/ 章、三、数据交换 *******/ 令 ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com: 字书型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕	*****/ ****/ * 章、三、数据交换 *******/ 令 ****/ ****/ ****/ */ ======*/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com: 字书型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /* /* 输入鍵、指示灯笔	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕	*****/ ****/ * 章、三、数据交换 *******/ 令 ****/ ****/ ****/ */ ======*/
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com: 字书型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com: 字书型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /* /* 输入键、指示灯 /*====================================	/**** AB_sD0=AB_status^0: 用户参数更新标记 /**** AB_sD7=AB_status^7: 报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0: 请求数据/参数命 /**** req_comD1=req_com^1: 清参数命令 等设备变量定义完毕 ====================================	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /* /* 输入键、指示灯笔 /*====================================	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 ====================================	****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0: 用户参数更新标记 /**** AB_sD7=AB_status^7: 报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0: 请求数据/参数命 /**** req_comD1=req_com^1: 清参数命令 等设备变量定义完毕 ====================================	*****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 ====================================	****/ ****/ ****/ ****/ ****/ ****/ */ *
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ 章、三、数据交换 ******/ 令 ****/ ****/ */ */ =====*/ */ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 ====================================	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 ====================================	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ *****/ *****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ *****/ *****/ ****/ ****/ ****/ *****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /***** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ ****/ ****/ ****/ ****/ ****/ **/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /**** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 	*****/ ****/ ****/ ****/ ****/ ****/ ****/ */
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /***** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 第设备变量定义完毕 第设备变量定义完毕 第次指示灯控制子程序,用户了解功能即可 将V值送指示灯Q0-Q7 输出************************************	<pre>*****/ ****/ ****/ ****/ *****/ *****/ *****/ ******</pre>
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /***** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 第设备变量定义完毕 第次指示灯控制子程序,用户了解功能即可 将V值送指示灯Q0-Q7 输出************************************	<pre>*****/ ****/ ****/ *****/ *****/ *****/ *****/ ******</pre>
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /***** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 第设备变量定义完毕 第次指示灯控制子程序,用户了解功能即可 将V值送指示灯Q0-Q7 输出************************************	<pre>*****/ ****/ ****/ ****/ *****/ *****/ ******</pre>
bdata UBYTE AB_status; sbit AB_sD0=AB_status^0; sbit AB_sD7=AB_status^7; /***** 定义: req_com:字节型, bdata UBYTE req_com; sbit req_comD0=req_com^0; sbit req_comD1=req_com^1; /*	/**** AB_sD0=AB_status^0:用户参数更新标记 /**** AB_sD7=AB_status^7:报文格式 请求数据命令,见《PB-OEM2-SE 产品手册》第四 /**** req_comD0=req_com^0:请求数据/参数命 /**** req_comD1=req_com^1:清参数命令 等设备变量定义完毕 第设备变量定义完毕 第次指示灯控制子程序,用户了解功能即可 将V值送指示灯Q0-Q7 输出************************************	<pre>****/ ****/ ****/ ****/ *****/ *****/ ******</pre>

P0=v;

```
P0=v;
P0=v;
P0=v;
out8_15=1;
out8_15=0;
}
/****************************** 将V 值送输入键指示灯ILED0-LIED7 输出**************/
void steinled0_7(UBYTE v)
ł
inled0_7=0;
P0=v;
P0=v;
P0=v:
P0=v;
inled0_7=1;
inled0_7=0;
}
/******************************* 将V 值送输入键指示灯ILED8-LIED15 输出*************/
void steinled8_15(UBYTE v)
ł
inled8_15=0;
P0=v;
P0=v;
P0=v:
P0=v;
inled8_15=1;
inled8_15=0;
ł
UBYTE rd_address()
ł
UBYTE data x;
P0=0xff;
inadd=0;
x=P0:
x=P0;
x=P0;
x=P0;
inadd=1;
return(x);
}
void scanin()
ł
P0=0xff;
in0_7=0;
PP0=P0;
PP0=P0;
PP0=P0;
PP0=P0;
if (key0_7!=PP0)
{
 if ((PP00==1)&&(key0==0)) { di0=~di0;};
 if ((PP01==1)&&(key1==0)) { di1=~di1;};
 if ((PP02==1)&&(key2==0)) { di2=~di2;};
 if ((PP03==1)&&(key3==0)) { di3=~di3;};
 if ((PP04==1)&&(key4==0)) { di4=~di4;};
 if ((PP05==1)&&(key5==0)) { di5=~di5;};
 if ((PP06==1)&&(key6==0)) { di6=~di6;};
 if ((PP07==1)&&(key7==0)) { di7=~di7;};
```

```
key0_7=PP0;
};
in0 7=1:
inled0 7=0;
P0=di0_7;
P0=di0_7;
P0=di0_7;
P0=di0_7;
inled0_7=1;
inled0_7=0;
P0=0xff;
in8 15=0;
PP0=P0:
PP0=P0:
PP0=P0;
PP0=P0;
if (key8_15!=PP0)
ł
  if ((PP00==1)&&(key8==0)) { di8=~di8;};
  if ((PP01==1)&&(key9==0)) { di9=~di9;};
  if ((PP02==1)&&(key10==0)) { di10=~di10;};
  if ((PP03==1)&&(key11==0)) { di11=~di11;};
  if ((PP04==1)&&(key12==0)) { di12=~di12;};
  if ((PP05==1)&&(key13==0)) { di13=~di13;};
  if ((PP06==1)&&(key14==0)) { di14=~di14;};
  if ((PP07==1)&&(key15==0)) { di15=~di15;};
 key8_15=PP0;
}:
in8_15=1;
inled8_15=0;
P0=di8_15;
P0=di8_15;
P0=di8_15;
P0=di8_15;
inled8_15=1;
inled8_15=0;
ł
/*==
                   */
/*
                          主程序
/*.
                                                                                      .*/
void
       main ()
ł
UBYTE data i,j,k,x,y,count,frist;
UWORD data nw,nn;
UBYTE xdata * data p;
UBYTE data address;
                                       /*-----从站地址
                                                                                 _____*/
                                        /*------ 字符偶校验错
                                                                                 _____*/
UBYTE data re_er;
                                        /*------ 串口波特率号
UBYTE data br_num;
                                                                                  */
                                       /*-----校验和错
UBYTE data error;
                                                                                 .....*/
                                       /*----- 串口接收缓冲
                                                                                   -----*/
UBYTE xdata rebox[113];
                                       ----*/
UBYTE xdata trbox[113];
UBYTE xdata pb_out[112];
                                       /*-----PROFIBUS 输出数据区
                                                                                 _____*/
                                       /*-----PROFIBUS 输入数据区
UBYTE xdata pb_in[112];
                                                                                 ----*/
                                       /*-----PROFIBUS 用户参数区
UBYTE xdata user_prm[41];
                                                                                 ....*/
                                       /*-----数据交换输入(实验板发送)数据长度
                                                                                 .....*/
UBYTE data data_in_len;
                                                                                 -----*/
                                       /*-----不包括校验和的data_in_len 长度
UBYTE data dil;
UBYTE data data_out_len;
                                       /*-----数据交换输出(实验板接收)数据长度
                                                                                .....*/
                                        /*-----不包括校验和的 data_out_len 长度
UBYTE data dol;
                                                                                 .....*/
                                       /*-----用户参数长度
                                                                                 -----*/
UBYTE data user_prm_len;
                                        /*-----OEM2-SE 接收数据错误类型号
                                                                                 -----*/
UBYTE data se_er;
P1=0xff;
P0=0;
```

P2=0xf0;

di0_7=di8_15=0;

```
steout0_7(0);
                                          /*----- 输出指示灯 00-07 清零
                                                                                               .....*/
steout8 15(0);
                                          /*-----输出指示灯Q8-Q15 清零
                                                                                                 _*/
                                          /*------ 输入键指示灯ILED0-ILED7 清零
steinled0_7(0);
                                                                                                 ._*/
                                          /*------ 输入键指示灯 ILED8-ILED15 清零
steinled8_15(0);
                                                                                              .....*/
/*
                                                                                           ==*/
    以下是实验板上电后,检测P89V51RD2 单片机内部RAM 的一段输出演示程序;
/*
                                                                                              */
/*
   这段程序与PROFIBUS 接口板 SE 通信无关,用户可跳过此段程序。
                                                                                              */
/*_
                                                                                          ----*/
p=0;
x=1;
k=1;
for (j=0;j<32;j++)
 {
  for (i=0;i<24;i++)
   {
    *(p+i)=0xaa;
   };
  y=0;
  for (i=0;i<24;i++)
   ł
    count=*(p+i);
    if (count!=0xaa) { y=1; };
   };
  if(y==0)
   ł
     switch (k)
      ł
       case 1:{ steout0_7(x);};break;
       case 2:{ steout8_15(x);};break;
       case 3:{ steinled0_7(x);};break;
       case 4:{ steinled8_15(x);};break;
       default:break;
      };
     if (x<255) {x=x*2+1;}else{x=1;k++;};
   }
  else
   ł
    for (;;)
     ł
      for (nw=0;nw<6000;nw++)
       ł
        steout0_7(0);
        steout8_15(0);
        steinled0_7(0);
        steinled8_15(0);
       };
      for (nw=0;nw<6000;nw++)
       ł
        steout0_7(0xff);
        steout8_15(0xff);
        steinled0_7(0xff);
        steinled8_15(0xff);
       };
     };
   ];
  p=p+24;
  for (nw=0;nw<8000;nw++)
   {};
};
    实验板输出演示程序结束。
                                                                                                  */
/*
```

```
/*_____*/
```

steout0_7(0); /*-----输出指示灯00-07 清零 ----*/ steout8_15(0); /*-----输出指示灯 08-015 清零 ----*/ /*------ 输入键指示灯ILED0-ILED7 清零 steinled0 7(0);*/ /*------**输入键指示灯ILED8-ILED15 清零** steinled8_15(0); ----*/ /*______ /* 清串口输入缓冲区 trbox[i],清串口输出缓冲区 rebox[i] */ /*_____ .----*/ for (i=0;i<113;i++) ł rebox[i]=trbox[i]=0; *};* for (i=0:i<112:i++) { *pb_in[i]=pb_out[i]=0; }*; address=rd_address(); /*------读实验板上拨码开关,作为站号-----*/ /*_____ _____ */ 以下是串口初始化程序 /*_____ */ br_num=5; /*------- 实验板波特率选择5,115.2K------*/ switch (br_num) /*------ 根据波特率号设置 T2------*/ ł case 1:{ TH2=0x0ff; TL2=0x0a0; RCAP2H=0x0ff; RCAP2L=0x0a0; }; break; /*---9.6K----*/ case 2:{ TH2=0x0ff; TL2=0x0d0;RCAP2H=0x0ff;RCAP2L=0x0d0;};break; /*---19.2K--*/ case 3:{ TH2=0x0ff; TL2=0x0e8;RCAP2H=0x0ff;RCAP2L=0x0e8;};break; /*---38.4K--*/ case 4:{ TH2=0x0ff; TL2=0x0f0;RCAP2H=0x0ff;RCAP2L=0x0f0;};break; /*---57.6K--*/ case 5:{ TH2=0x0ff; TL2=0x0f8;RCAP2H=0x0ff;RCAP2L=0x0f8;};break; /*---115.2K-*/ case 6:{ TH2=0x0ff; TL2=0x0fc;RCAP2H=0x0ff;RCAP2L=0x0fc;};break; /*---230.4K-*/ case 7:{ TH2=0x0ff; TL2=0x0fe;RCAP2H=0x0ff;RCAP2L=0x0fe;};break; /*---460.8K-*/ default:break; *};* ___*/ 注意:本例基于PHILIP P89V51RD2 单片机: */ 波特率=14745600/16/(65536-T) */ /* 各种波特率9600: T=0xffa0, 19.2K: T=0xffd0, 38.4K: T=0xffe8, */ /* 57.6K: T=0xfff0, 115.2K:T=0xfff8, 230.4k:T=0xfffc, 460.8k=0xfffe */ /*_ __*/ *PCON=0; T2CON=0x34*; /*----RCLK=1; TCLK=1; TR2=1; C/T2=0*/ /*_____*/ /* 串口设置成模式3 即:1 个启始位、8 个数据位、1 个偶校验位、1 个停止位; */ */ /* 0xd0=11010000B; SM0SM1=11; SM2=0; REN=1;TB8=0; RB8=0; TI=0; RI=0 */ SCON=0x0d0; 串口初始化程序结束 /* */ /*______/ while (REO IT==1) /*==: /* 形成接口板的初始化报文 */ /* rebox[]是串口发送数组,格式见《SE 手册》 */ /*_. /*---- 输入数据报文长度=48+1+1=50; /*---- 用户参数长度=0 /*---- 输入数据报文长度=48+1+1=50; data_in_len=48+1+1; ----*/ user_prm_len=0;*/ data_out_len=48+1+1;*/ /*---- BYTE0=从站地址 ----*/ trbox[0]=address; trbox[1]=0x06; /*---- BYTE1=ID 号高位*/ /*---- BYTE2=ID 号低位 *trbox*[2]=0x0*fa*; /*_____ _____ _____ ----* /*

以下形成 I/O 配置数据

	新关件政	
/* 本例: I/O 配置数据长度: /* 共: 48bytes input+48bytes	=6; I/O 配置数据: 0x1f,0x2f,0x1f,0x2f,0x1f,0x2f s output	*/ */ */
/* trhor[3]-6.		*/
trbox[3]=0, trbox[4]=0x1f.	/DIIE3_1/O 配置数据从及-0 /*RVTF/-I/O 配置数据。0rlf	*/
trbox[5]=0x2f	/DIIL4-I/O 配量数据: 0x1j /*	*/
trbox[5]=0x2j, trbox[6]=0x1f.	/BTTE5=1/0 配置数据: 0x2j /*RVTF6=1/0 配置数据. 0x1f	*/
$\frac{1}{100x[0]-0x1j},$	/*DIIL0-I/O 配置数据: 0x1j /*	*/
trbox[2]=0x2j,	/*DIIL/-I/O 配量致油: 0x2j /* DVTE9_I/O 配罢粉提, 0x1f	*/
uvox[0] = 0x1j;	/*DIILO_I/O 尼旦双沿 ; 0x1j /* DVTE0_I/O 尼罗粉 提。0x2f	*/
<i>ubox[9]=0x2j;</i>	/ ·D111E9_1/O 间双	
trhov[74]–data in len:	/*BYTF24-data in len 输入数据报文长度	*/
trbox[25]=data_out_len	/*BYTE25=data out len 输出数据报文长度	*/
trbox[26]=user nrm len	/*BYTE26-user nrm len 田白参数长度	*/
u box[20]=user_prm_ten,		
x=0; for (i=0;i<48;i++) {		
x=x+trbox[i];	/*	*/
;; trbox[48]=x;	/* <i>置校验和</i>	*/
/* 发送接口板的核 /*	团始化报文、接收接口板的回答,初始不成功继续发表 	<i>初始化报文</i> */ */ <mark>准备好接收 </mark> */
y=0; while (y==0)	/*	*/
{ REN=0; for (i=0;i<49;i++) { TI=0; ACC_tabactile	/* 发送一次初始化报文共 49 字节	*/
ACC=irbox[i]; TB8=P; SBUF=trbox[i]; while (TI==0) {}; };	/* 形成偶校验位	*/
while (S_RTS==0) { };	/* SE 没有接收完毕	*/
/*发送完毕、等待接	收回答 */	
<i>RI=0;</i>		
<i>REN=1;</i>	/* 允许接收	*/
re_er=0;	/* re_er 偶校验错标志	*/
for (j=0;j<49;j++)		
{	/* 接受回答报文共49 个字节	*/
while ((RI==0)&& if (RI==1)	(S_RTS==1)) {}; /* SE 正在发送、实验材	扳正在接收一个字符* /
{/* RI==1:实验 ACC=SBUF;	金板接收到一个字符* /	
if (P!=RB8){ re_ rebox[j]=SBUF; RI=0;	er=1;}; /* 偶校验错,置偶	校验错标志 */ 串口输入缓冲区 */
} else		

{ /*-----(RI==0)&&(S_RTS==0):SE 报文全部发送完毕而实验板还没有接收到预期的长度 ------*/ j=100; /*---- 退出等待接收 -----*/ } *};*

 if (j==49) { / *---- (j==49):实验板接收回答报文长度正确 ------*/ $if(re_er==0)$ /*---- 没有字符偶校验错? ----*/ { x=0;/*---- 求校验和 for (i=0;i<48;i++) ----*/ { x=x+rebox[i]; *}; if* (*x*==*rebox*[48]) /*---- 校验和正确? -----*/ ł *if* (*rebox*[0]==*br_num*) /*---- **返回的波特率号是**5?*/ { /*---- 初始化报文错误号N=0? *if* (*rebox*[1]==0) -----*/ { x=0; for (i=2;i<48;i++) /*---- byte2~byte47 **都是**0xaa? -----*/ { *if* (*rebox*[*i*]*!=0xaa*) { x=1;*};* }; if(x==0){ /*---- byte2~byte47 都是0xaa,初始化成功,y=1,准备退出 while (y==0) ----*/ y=1; ł else ${x=0x11;};$ /*---- byte2~byte47 不都是0xaa! 错误标记=0x11; ----*/ 3 else /*---- 初始化报文错误号N<>0,错误标记=返回错误号 {*x*=*rebox*[1];}; ----*/ ł else /*---- 返回的波特率号不是5, 错误标记=0x77; *{x=0x77;};* ----*/ } else {*x=0xff;*}; /*---- 校验和不正确正确, 错误标记=0xff; ----*/ ł else *{x=0xee;};* /*---- 发生了字符偶校验错 错误标记=0xee; ----*/ J else ł /*---- j!=49 发生了接收报文长度错 错误标记=0xee; x=0xcc; ----*/ **};** if(y==0)/*---- 初始化失败 ----*/ ł for (nw=0;nw<10000;nw++) ł steout0_7(br_num); /*---- 显示返回波特率号 ----*/ *steout8_15(x);* /*---- 显示返回错误号 ----*/ }; *};* }; /*---- 如果 y==0, 返回 while (y==0) ----*/ /*______ _____*/ 实验板对PROFIBUS 接口板初始化成功,进数据交换状态的无限循环中 /* /* trbox[]/rebox[]是串口发送/接收数组,发送/接收格式见《SE 手册》

dil=data_in_len-1;	/* data_in_len=50, dil=49	*/
dol=data_out_len-1;	/* data_out_len==50, dol==49	*/
for (i=0;i <data_in_len;i++)< td=""><td></td><td></td></data_in_len;i++)<>		

*/

/ ____/

{ trbox[i]=0; /*---- 清发送、接收缓冲区 ----*/ *}*: for (i=0;i<data out len;i++) { /*---- 清发送、接收缓冲区 rebox[i]=0; ----*/ *};* /*__-while (REO IT==0) { /*---- (REQ_IT==0):SE 处在数据交换状态 ------*/ trbox[0]=req_com=0; /*---- 没有用户参数,请求数据命令永远是 req_com=0,只请求 A 型报文 ------*/ /*---- di0_7 是输入键输入 I0~I7 当前值,作为 PROFIBUS 输入的第1 字节 ------*/ *trbox*[1]=*di*0 7; /*---- di8_15 是输入键输入 I8~I15 当前值,作为 PROFIBUS 输入的第2 字节 -----*/ *trbox*[2]=*di*8_15; for (i=3;i<dil;i++) ł -----*/ trbox[i]=0x5a; /*---- 其它PROFIBUS 输入数据(3~48)本例定为0x5a *};* x=0;/*---- 求校验和 *for* (*i*=0;*i*<*di*];*i*++) ----*/ { x=x+trbox[i];*}*: trbox[dil]=x; /*---- 置校验和*/ REN=0; /*----SRTS==1: 接口板没有准备好接收数据 ------*/ while (S RTS==1) {}; /*---- 发送数据交换报文共50 字节 for (i=0;i<data_in_len;i++)*/ ł *TI=0*: ACC=trbox[i]; /*---- 形成偶校验位*/ TB8=P; SBUF=trbox[i]; while (TI==0) {}; *};* while (S RTS==0) {}; /*---- SE 没有接收完毕 -----*/ *re_er=0;* /*----- re_er 偶校验错标志 */ *RI=0;* /*----- 允许接收 ____*/ REN=1; for (j=0;j<data_out_len;j++)</pre> /*----- 等待接收接口板的回答数据交换报文共50 字节 ----*/ ł while ((RI==0)&&(S_RTS==1)) {}; /*---- SE 正在发送、实验板正在接收一个字符 -----*/ if (**RI**==1) {/*---- RI==1:实验板接收到一个字符 ------*/ ACC=SBUF; if (P!=RB8) {re_er=1;}; /*----- 偶校验错,置偶校验错标志 -----*/ /*----- 接收一个字符送串口输入缓冲区 rebox[j]=SBUF;*/ **RI=0;** else {/*---- (RI==0)&&(S_RTS==0):SE 报文全部发送完毕而实验板还没有接收到预期的长度------*/ /*---- 退出等待接收*/ j=data_out_len+1; **};** *};*

if (j==data_out_len)

{/*---- (j==data_out_len):实验板接收回答报文长度正确 ------*/

<i>if</i> (<i>re_er==0</i>)	/* 于俚校险举	*/
<i>x=0;</i> for (i=0;i <dol;i++)< td=""><td></td><td> ,</td></dol;i++)<>		,
{	/*	*/
}, if (x==rebox[dol]) {		
y=dol-1; for (i=1;i <dol;i++)< td=""><td>/* y=49-1=48,PROFIBUS 输出数据长度</td><td>*/</td></dol;i++)<>	/* y=49-1=48,PROFIBUS 输出数据长度	*/
{	/* 接收到的48 个PROFIBUS 输出数据送数据区 p	b_out[0~47]*/
else { error=0x81; };	/* 校验和错	*/
} else { error=0x83; };	/* 偶校验错	*/
<pre>/ else { error=0x84; };</pre>	/* 接收字符数不够	*/
/* <i>显示部分</i>		*/ */
scanin(); steout0_7(pb_out[0]); steout8_15(pb_out[1]);	/*	, */ */
};/* while (REQ_IT==0):SE タ };/* (REQ_IT==1):SE 退出数 }	让在数据交换状态,返回数据交换状态的无限循环 据交换状态,返回初始化	中*/ */
(8) 例 2 的 GSD 文件		
例 2 的 GSD 文件与例 1 完全-	一致,故一律用斜体小五号字表示。	
,	DO /SPC3 鼎实创新科技有限责任公司	
, #Profibus_DP ; <prm-text-def-list> ; <ext-user-prm-data-def-list> ; <unit-definition-list> GSD_Revision=2</unit-definition-list></ext-user-prm-data-def-list></prm-text-def-list>		
Vendor_Name=''DS FieldBus Ltd. Co.'' Model_Name=''PB-OEM2-SE-B1'' Revision=''V1.0 ''	' ; 公司名,可按用户名修改 ;模块名称,也是组态时该产品的名称,组态软件如	STEP 7, COM PROFIBUS
Ident_Number=0x06FA Protocol_Ident=0 Station_Type=0	;ID 号,必须与初始化报文一致	
Hardware_Release="A1.0 " Software_Release="Z1.0 "		

9.6_supp=1 19.2_supp=1 93.75_supp=1 187.5_supp=1 500_supp=1 45.45_supp=1 1.5M_supp=1 3M_supp=1 6M_supp=1 12M_supp=1 MaxTsdr_9.6=60 MaxTsdr_19.2=60 MaxTsdr_45.45=250 MaxTsdr_93.75=60 MaxTsdr_187.5=60 MaxTsdr_500=100 MaxTsdr_1.5M=150 MaxTsdr_3M=250 MaxTsdr_6M=450 MaxTsdr_12M=800 Implementation_Type="spc3" Bitmap_Device="SE_B" ;图标文件,用户可以自制图标,缺省 ; Slave-Specification: ;产品序列号,可按用户名修改 OrderNumber="pb-oem2-s" Freeze_Mode_supp=1 Sync_Mode_supp=1 Auto_Baud_supp=1 Fail_safe=0 Min_Slave_Intervall=1 Max_Diag_Data_Len=6 User_Prm_Data_Len=0 Modular_Station = 0 Modul_Offset=0 Slave_Family=3@TdF@PB-OEM2 ;组态中的分类名 ; <Module-Definition-List> *Module='' 48 Byte In, 48 Byte Out '' 0x1f,0x2f,0x1f,0x2f,0x1f,0x2f* ;I/O 配置,必须与初始化报文一致 EndModule

例 3. 带用户参数功能的工作模式

例 3 仍以实验板为模型,模拟实现一个 PROFIBUS I/O 模块;为方便起见称为实验 I/O。实验 I/O 设 计将使用"用户参数"功能。因此本例将全面地讲解用户参数功能、带用户参数的 SE 接口初始化和带 用户参数功能的数据交换过程。

(1) 需要使用"用户参数 user_prm"的情况

对于工业现场设备,常需要用户根据现场应用设定一些参数,其中有些参数不需要在设备运行中实时改变,如变频器的电流上限保护与报警值、温度传感器的测量温度范围、热电偶选型、输出4-20mA/1-5V 选择等等。如果这些参数作为 PROFIBUS 主站的 I/O 输出,将占用 PROFIBUS 主站 I/O 资源和周期性轮 循 PROFIBUS 从站的时间资源。

将这些参数处理成"用户参数",将会缩短 PROFIBUS 主站通信时间、减小通信报文长度、提高总 线通信效率。使用"用户参数"技术,只需要在主站配置中做出参数选择,主站在与从站连接时,一次 性地将这些参数传送到从站,从站就可以使用这些用户选择的参数对从站进行参数化(初始化、参数设 定等)。

(2) 实验 I/O 功能及实现方法

- ① 以实验板为模型,模拟实现一个 PROFIBUS I/O 从站模块,称为实验 I/O;
- ② 实验 I/O 具有配置灵活的特点,用户可以通过配置软件(如 STEP 7、COM PROFIBUS),将实验 I/O 配置成具有 1 路 AI+1 路 AO 的模拟量和 16DI+16DO 开关量的 PROFIBUS-DP I/O;
- ③ 对模拟量 I/O 可以通过配置选择输入/输出信号类型、数据格式等等;对开关量 I/O 可以通过配置选择输入/输出电压/功率等级、正/负逻辑等功能。
- ④ 为实现上述功能方法是:
- (i) 在实验 I/O 的 GSD 文件中详细描述用于配置的用户参数;
- (ii) 主站配置中由用户选择这些配置参数;
- (iii) 当主站与从站实验 I/O 连通时,主站将发送"参数化"命令,将用户配置参数下传至实验 I/O;
- (iv)实验 I/O 将根据用户参数设置软硬件,实现 1 路 AI+1 路 AO 模拟量和 16DI+16DO 开关量 I/O 的功能。

③ 具体确定"用户参数"类型、个数、取值范围

根据实验 I/O 的上述要求,首先需要具体确定"用户参数"的类型、个数、取值范围。

① 用户参数 1: 输入类型 AI_M, 字节型

AI_M=0~7: 代表输入类型 1~5V、0~10V、0~5V、-10~+10V、4~20mA、0~20mA、0~10mA、-10~+10 mA 共八种选择;

- ② 用户参数 2:数据类型 AI_D,字节型
- AI_D=0-2:分别表示 BCD (0000~9999)、无符号整型 (0~65535)、有符号整型 (-32767~+32767) 共三种选择:
- ③ 用户参数 3: 单/双端 AI CH, 字节型
- AI_CH=0-1: 分别表示单端输入、双端输入,共二种选择;
- ④ 用户参数 4: 输出类型 AO_M, 字节型
- AO_M=0~7: 代表输出类型 1~5V、0~10V、0~5V、-10~+10V、4~20mA、0~20mA、0~10mA 共八种选择;
- ⑤ 用户参数 5:数据类型 AO_D,字节型
- AO_D=0-2: 分别表示 BCD (0000~9999)、无符号整型 (0~65535)、有符号整型 (-32767~+32767)共 三种选择:
- ⑥ 用户参数 6: 单/双端 AO CH, 字节型
- AO_CH=0-1: 分别表示单端输出、双端输出共二种选择;
- ⑦ 用户参数 7: 正/负逻辑 DI PN,字节型
- DI_PN=0-1:分别表示正逻辑(高有效)、负逻辑(低有效)这2种输入模式;
- ⑧ 用户参数 8: 电压等级 DI_V, 字节型
- DI_V=0-3 分别表示 5V/DC、24V/DC、110V/AC、220V/AC 4 种输入电压等级;
- ⑨ 用户参数 9: 正/负逻辑 DO_PN, 字节型
- DO_PN=0-1:分别表示正逻辑(高有效)、负逻辑(低有效)这2种输出模式;
- ⑩ 用户参数 10: 电压功率等级 DO_V, 字节型
- DO_V=0-3 分别表示 5VDC/0.5A、24VDC/0.5A、110VAC/0.5A、220VAC/0.5A4 种输出电压功率等级;

因此,本例需要 10 个字节用户参数,用户参数长度 User_Prm_Data_Len=10。

(4) 带有"用户参数"描述的 GSD 文件

与例 1、例 2 相比,实验 I/O 的 GSD 文件至少有如下关于"用户参数"的描述,见红色下划线部分; 下面是关于实验 I/O 的一个简单并带有"用户参数"描述的 GSD 文件:

- ;GSD文件: OEM2-B3 实验I/O 鼎实创新科技有限责任公司
- ;产品型号: PB-OEM2-实验I/O ;版本 : 2004年9月 V1.0
- ;版本 : 2004年9月 V1.0 ;文件名 : OEM2-B3.GSD

[#]Profibus_DP

^{; &}lt; Prm-Text-Def-List>

^{; &}lt;Ext-User-Prm-Data-Def-List>

^{; &}lt;Unit-Definition-List>

GSD_Revision=2 Vendor_Name="DS FieldBus Ltd. Co." ;公司名,可按用户名修改 Model Name="PB-OEM2-SE-B3" ;模块名称,也是组态时该产品的名称,组态软件如STEP 7, COM PROFIBUS Revision="V1.0 " ;ID号,必须与初始化报文一致 Ident_Number=0x06FA **Protocol_Ident=**0 Station_Type=0 Hardware_Release="A1.0 " Software_Release="Z1.0 " **9.6 supp**=1 **19.2_supp**=1 **93.75_supp**=1 187.5_supp=1 **500_supp**=1 45.45_supp=1 1.5M_supp=1 3M_supp=1 6M_supp=1 12M_supp=1 MaxTsdr_9.6=60 MaxTsdr 19.2=60 MaxTsdr_45.45=250 MaxTsdr_93.75=60 MaxTsdr_187.5=60 MaxTsdr_500=100 MaxTsdr_1.5M=150 MaxTsdr_3M=250 MaxTsdr_6M=450 MaxTsdr_12M=800 Implementation_Type="spc3" **Bitmap_Device=**"SE_B" ;图标文件,用户可以自制图标,缺省 ; Slave-Specification: ;产品序列号,可按用户名修改 OrderNumber="pb-oem2-s" Freeze_Mode_supp=1 Sync_Mode_supp=1 Auto_Baud_supp=1 Fail_safe=0 Min_Slave_Intervall=1 Max_Diag_Data_Len=6 User Prm Data Len=10 :用户参数长度 :用户参数默认值 **Modular_Station** = 0Modul Offset=0 Slave_Family=3@TdF@PB-OEM2 ;组态中的分类名 ; < Module-Definition-List> **Module=**"1AI+1AO/16DI+16DO " 0x70, 0x31 ;1 word input +1 word output / 2 bytes input+ 2 bytes output **EndModule**

这是实现实验L/O最简单的一个GSD文件,但用户使用却不方便。为了使用户更加了解每个参数含义,

顺利完成实验I/O的配置,我们给用户提供了一张参数配置表,如表4-1所示:

序	名称	取值范围和设定	
号			
1	输入类型AI_M	AI_M=0~7: 代表输入类型 1~5V、0~10V、0~5V、-10~+10V、4~20mA、0~20mA、	
		0~10mA、-10~+10 mA 共八种选择;	

表4-1 实验I/O的用户参数表

2	粉 据 米 刑 AI D	ALD-0.2 公则主子 DCD (0000 0000) 王符号教刊 (0 65525) 方符号教刊 (20767, 20767)
2	数据关至AI_D	AI_D=0-2: 分别农小 BCD(0000~9999)、几村 5 釜至(0~05555)、 有村 5 釜至(-52/0/~+52/0/)
		共三种选择;
3	单/双端AI_CH	AI_CH=0-1:分别表示单端输入、双端输入,共二种选择;
4	输出类型	AO_M=0~7: 代表输出类型 1~5V、0~10V、0~5V、-10~+10V、4~20mA、0~20mA、
	AO_M	0~10mA、-10~+10 mA 共八种选择;
5	数据类型AO_D	AO_D=0-2:分别表示BCD(0000~9999)、无符号整型(0~65535)、有符号整型(-32767~+32767)
		共三种选择;
6	单/双端AO_CH	AO_CH=0-1: 分别表示单端输出、双端输出,共二种选择;
7	正/负逻辑	DI_PN=0-1:分别表示正逻辑(高有效)、负逻辑(低有效)这2种输入模式;
	DI_PN	
8	电压等级DI_V	DI_V=0-3 分别表示 5V/DC、24V/DC、110V/AC、220V/AC 4 种输入电压等级;
9	正/负逻辑	DO_PN=0-1:分别表示正逻辑(高有效)、负逻辑(低有效)这2种输出模式;
	DO_PN	
10	电压功率等级	DO_V=0-3 分别表示 5VDC/0.5A、24VDC/0.5A、110VAC/0.5A、220VAC/0.5A 4 种输出电压功率
	DO_V	等级;

由 GSD 文件中用户参数默认值:

实验I/O的默认配置是:

- AI_M=0: 代表输入类型1~5V;
- AI_D=1: 表示无符号整型(0~65535);
- AI_CH=0: 表示单端输入;
- AO_M=0: 代表输出类型1~5V;
- AO_D=1: 表示无符号整型(0~65535);
- AO_CH=0: 表示单端输出;
- DI_PN=0: 表示正逻辑(高有效);
- DI_V=0: 表示5V/DC;
- DO_PN=0: 表示正逻辑(高有效);
- DO_V=0: 表示5VDC/0.5A;

(5) 如何在主站配置中选择用户参数

首先以STEP 7为例:

① 实验I/O的GSD文件OEM2_B3.GSD拷至: SIEMENS\step7\S7data\gsd\

实验I/O的图标文件SE_B.BMP拷至: SIEMENS\step7\S7data\nsbmp\

② 进入SIMATIC Manager → HARDWARE, 选择Options → Updata Catalog;

	<i>.</i>		
HT Config: Con	tiguring hardware		
Station <u>P</u> LC <u>V</u> iew	Options Help		
	Customi <u>z</u> e Ctrl+Alt+E	<u>N?</u>	
	<u>E</u> dit Catalog Profile		Hardware Catalog
	<u>U</u> pdate Catalog		Profi Standard
	<u>I</u> nstall New GSE		THE PROFIBILS DP
	Import Station GSE		E MI SIMATIC 300
			SIMATIC 400
			🗄 🔝 SIMATIC PC Based Control 300,
			🗄 🖳 SIMATIC PC Station

图 4-3 选择 Options → Updata Catalog

③ 配置实验I/O从站。用户可以按照表4-1:实验I/O的用户参数表来改变用户参数配置实验I/O。



图 4-4 STEP 7一改变用户参数配置实验 I/O

再以COM PROFIBUS V5.1为例:

① 实验I/O的GSD文件OEM2_B3.GSD拷至: CPBV51\gsd\

- 实验I/O的图标文件SE_B.BMP改名SE_B.DIB拷至: CPBV51\bitmaps\
- ② 进入COM PROFIBUS V 5.1 → File → 选择 Read in GSD Files
- ③ 配置实验I/O从站。用户可以按照表4-1: 实验I/O的用户参数表,改变用户参数配置实验I/O。



图 4-5 COM PROFIBUS V5.1—用户可按照上表改变用户参数配置实验 I/O

(6) 更便于用户使用的 GSD 文件

上述实验 I/O 的 GSD 文件,配置用户参数时需要对照用户参数表,见"图 4-4: STEP7-改变用户参数配置实验 I/O",这对用户来说极不方便。下面是改进的实验 I/O GSD 文件,可在配置中通过菜单来配置用户参数,这种方法对用户来说比较方便,但 GSD 文件结构相对复杂。

```
;GSD 文件: OEM2-B4 实验 I/O 鼎实创新科技有限责任公司
;产品型号:
            PB-OEM2-实验 I/O; 版本 :
                                        2004年9月 V1.0
; 文件名 :
            OEM2-B4.GSD
:======
#Profibus DP
; User-Parameter-Definition
PrmText=1
Text(0) = "1 - 5V"
Text(1)="0--10V"
Text(2)="0--5V"
Text(3)="-10V--+10V"
Text(4)="4--20mA"
Text(5)="0--20mA"
Text(6)="0--10mA"
Text(7)="-10--+10mA"
EndPrmText
PrmText=2
```

```
Text(0)="BCD (0000~9999) "
```

Text(1)="无符号整型(0~65535)" Text(2)="有符号整型(-32767~+32767)" EndPrmText PrmText=3 Text(0)="单端" Text(1)="双端" EndPrmText PrmText=7 Text(0)="正逻辑(高有效)" Text(1)="负逻辑(低有效)" EndPrmText PrmText=8 Text(0)="5V/DC" Text(1)="24V/DC" Text(2)="110V/AC" Text(1)="220V/AC" EndPrmText ExtUserPrmData=1 "输入类型 AI_M: " Unsigned8 0 0-7 Prm_Text_Ref=1 EndExtUserPrmData ExtUserPrmData=2 "输入数据类型 AI_D: " Unsigned8 1 0-2 Prm Text Ref=2 EndExtUserPrmData ExtUserPrmData=3 "输入单端/双端 AI_CH: " Unsigned8 0 0-1 Prm_Text_Ref=3 EndExtUserPrmData ExtUserPrmData=4 "输出输出类型 AO_M: " Unsigned8 0 0-5 Prm Text Ref=1 EndExtUserPrmData ExtUserPrmData=5 "输出数据类型 AO_D: " Unsigned8 1 0-2 Prm_Text_Ref=2 EndExtUserPrmData ExtUserPrmData=6 "输出单端/双端 AO CH: " Unsigned8 0 0-1 Prm_Text_Ref=3 EndExtUserPrmData ExtUserPrmData=7 "输入正/负逻辑 DI_PN:" Unsigned8 0 0-1 Prm_Text_Ref=7 EndExtUserPrmData ExtUserPrmData=8 "输入电压等级 DI_V:" Unsigned8 1 0-1 Prm Text Ref=8 EndExtUserPrmData ExtUserPrmData=9 "输出正/负逻辑 DO_PN:" Unsigned8 0 0-1 Prm_Text_Ref=7 EndExtUserPrmData ExtUserPrmData=10 "输出电压等级 DO_V:" Unsigned8 1 0-1

Prm_Text_Ref=8 EndExtUserPrmData ; Unit-Definition-List: GSD_Revision=2 Vendor Name="DS FieldBus Ltd. Co." ;公司名,可按用户名修改 ;模块名称,也是组态时该产品的名称,组态软件如 STEP Model Name="PB-OEM2-SE-B4" 7, COM PROFIBUS Revision="V1.0 " ;ID 号, 必须与初始化报文一致 Ident_Number=0x06FA Protocol_Ident=0 Station Type=0 Hardware_Release="A1.0 " Software_Release="Z1.0 " 9.6 supp=1 19.2_supp=1 93.75_supp=1 187.5 supp=1 500_supp=1 45.45_supp=1 1.5M_supp=1 3M_supp=1 6M_supp=1 12M_supp=1 MaxTsdr_9.6=60 MaxTsdr_19.2=60 MaxTsdr_45.45=250 MaxTsdr 93.75=60 MaxTsdr_187.5=60 MaxTsdr_500=100 MaxTsdr_1.5M=150 MaxTsdr_3M=250 MaxTsdr 6M=450 MaxTsdr_12M=800 Implementation_Type="spc3" Bitmap_Device="SE_B" ;图标文件,用户可以自制图标,缺省 ; Slave-Specification: ;产品序列号,可按用户名修改 OrderNumber="pb-oem2-s" Freeze_Mode_supp=1 Sync_Mode_supp=1 Auto_Baud_supp=1 Fail_safe=0 Min_Slave_Intervall=6 Slave_Family=3@TdF@PB-OEM2 :组态中的分类名 Max_Diag_Data_Len=6 Modul_Offset=0 Modular_Station=0 Max_Input_Len=224 Max_Output_Len=224 Max_Data_Len=448 User_Prm_Data_Len=10 Max_User_Prm_Data_Len=10 User_Prm_Data = 0x00,0x01,0x00,0x00,0x01,0x00,0x00,0x01,0x00,0x01 Ext_User_Prm_Data_Ref(0)=1 Ext_User_Prm_Data_Ref(1)=2 Ext_User_Prm_Data_Ref(2)=3 Ext_User_Prm_Data_Ref(3)=4

Ext_User_Prm_Data_Ref(4)=5 Ext_User_Prm_Data_Ref(5)=6 Ext_User_Prm_Data_Ref(6)=7 Ext_User_Prm_Data_Ref(7)=8 Ext_User_Prm_Data_Ref(8)=9 Ext_User_Prm_Data_Ref(9)=10 ; Module-Definitions: Module="1AI+1AO/16DI+16DO" 0x70,0x31 ;1 word input output EndModule

;1 word input +1 word output/2 bytes input +2 bytes

在 STEP 7 中的配置介面:



图 4-6 选择实验 I/O 设置:1AI+1AO/16DI+16DO



图 4-7 进一步的参数设置



图 4-8 可以看到配置 1AI+1AO 模块的 7 个用户参数

① 图 4-8: 可以看到配置 PB-OEM2-SE-B4 模块的 10 个用户参数: 00, 01, 00, 00, 01, 00, 00, 01, 00, 01。

② 第 1~8 参数: 00, 01, 00, 00, 01, 00, 00, 01 对应上面的配置: 输入类型 AI_M: 1—5V; 输入数 据类型 AI_D: 无符号整型 (0~65535); 输入单端/双端 AI_CH: 单端; 输出类型 AO_M: 1—5V; 输出 数据类型 AO_D: 无符号整型 (0~65535); 输出单端/双端 AO_CH: 单端; 输入正/负逻辑 DI_PN: 正逻 辑 (高有效); 输入电压等级 DI_V: 24V/DC;

- ④ 第 9~10 参数: 00, 01 对应上面的配置:输出正/负逻辑 DO_PN: 正逻辑(高有效)、输出电压等级 DO_V: 24V/DC。
- ⑤ 如果依照菜单改变配置,第1~10用户参数也会该变。

在 COM PROFIBUS V5.1 中的配置介面:

见下页图 4-9:在 COM PROFIBUS 中配置实验 I/O 模块后各种参数的设置。



(7) 带用户参数功能的 SE 接口的初始化

例 3 实验 I/O 技术指标摘要:

站号设置为 0x19, ID 号=06FA;

I/O 配置: 1 Word Input + 1 Word Output + 2 Bytes Input + 2 Bytes Output

I/O 配置数据长度 CFG_LEN=2; I/O 配置数据为: 0x70, 0x31

用户参数长度 User_Prm_Data_Len=10;

接收数据长 SD_in_len≥4+1+1=6

发送数据长 SD_out_len≥4+1+1=6;

因此,实验 I/O的 GSD 文件中必须有:

User_Prm_Data_Len=10

Max_User_Prm_Data_Len=10

User_Prm_Data = 0x00,0x01,0x00,0x00,0x01,0x00,0x01,0x00,0x01

Module="1AI+1AO/16DI+16DO" 0x70,0x31

EndModule

例 3 带用户参数功能的 SE 接口的初始化报文见下页图 4-9-1、图 4-9-2、图 4-9-3



(8) 带用户参数功能数据交换过程

为读者方便,将"表 4-2 请求数据命令 req_com"、表 4-3 报文类型及接口状态 AB_status"拷备到图 4-10 后。

- ① 进入数据交换状态后,实验板首先发送 PROFIBUS 输入数据(例 3 为 1AI+16DI)、并请求读取 PROFIBUS 输出数据,即 req_com =0, SE 接口将回答"输出数据报文 A",见图 4-10-1。
- ② 报文格式 A 中 AB_status (D7)=0 及包含 PROFIBUS 主站输出数据(1AO+16DO); 实验板除接收
 PROFIBUS 输出数据外还要判断用户参数更新标记 AB_status (D0)=?,见图 4-10-2。
- ③ 如果 AB_status (D7)=0 说明用户参数没有更新(可能主站还没有接通),实验板继续发送 PROFIBUS 输入数据并请求 SE 接口将回答"输出数据报文 A",见图 4-10-3。
- ④ 如果 AB_status (D7)=1 说明用户参数已更新(主站下载了用户参数), 见图 4-10-2。

- ⑤ 实验板将发送 PROFIBUS 输入数据并请求 SE 接口将回答"输出数据报文 B",即 req_com =1。见图 4-10-4。
- ⑥ SE 接口板将回答"输出数据报文 B",包括用户参数数据: AB_status (D7)=1、AB_status (D0)=1 见
 图 4-10-4、图 4-10-5。
- ⑦ 实验板得到用户参数(例3为10个字节)后可在实验板设置参数,并继续发送 PROFIBUS 输入数据并请求 SE 接口回答"输出数据报文 A",且令 req_com =0x20,清除用户参数更新标记 AB_status (D0)=1。



AB_status (D0)=1, 准备接收格式 A

图 4-10-5: 接口板回答报文格式 B, AB_status (D7)=1 且 AB_status (D7)保持 1。

表 4-2 请求数据命令 req_com

请求数据命令 req_com	说明	
D7~D2	不用	
D1	=0: 无效	
	=1:清接口板用户参数己由主站更新标记 AB_status (D0)	
D0	=0:请求接口回答 PROFIBUS 输出数据	
	=1:请求接口回答 PROFIBUS 主站更新的用户参数数据	

表 4-3 报文类型及接口状态 AB_status

报文类型及接口状态	说 明		
AB_status			
D7	=0: A 格式,本报文回答 PROFIBUS 输出数据,长度= SD_out_len-2		
	=1: B 格式,本报文回答用户参数数据,长度 User_Prm_Data_Len		
D6~D3	不用		
D2~D1	=00: 接口板接收用户板报文无错		
	=01: 接口板接收用户板报文字符有偶校验错		
	=10: 接口板接收用户板报文有校验和错		
	=11: 不用		
D0	=0:接口板还没有接收到主站发送的用户参数、或主站没有更新用户参		
	数		
	=1: 接口板用户参数数据已由主站更新;		

(9) 实验板初始化 SE 接口板的过程框图

例 3 实验板初始化 SE 接口板的过程框图与例 2 完全相同,见图 4-2-4:用户模板初始化接口模板的过程。

⑩ 数据交换状态下实验板工作流程

因为例 3 有用户参数功能,因此实验板数据交换状态下工作流程与例 1、例 2 不同。见下页图 4-11: 数据交换状态下用户模板的程序流程(例 3)。



图 4-11 数据交换状态下用户模板的程序流程(例 3)

(1) 例 3 实验 I/O 的程序清单

以下凡斜体部分与例2程序相同,蓝色正体部分为例3特殊部分。

/*+			+	
/ 文件 名称:	实验板板串口道	育信程序—例3	/	
/版 本:	V1.1		/	
<i> 制作单位</i> :	北京鼎实创新科	技公司	/	
版权许可:			/	
/ <i>网址:</i>	www.c-profibus.	com.cn	/	
email:	tangjy@c-profibu	s.com.cm	/	
日期:	2004/9		/	
+			+ */	
#include < c	:: 51 c51 INC reg.	52. <i>h</i> >		
#define UB	YTE unsigned cha	r		
#aefine UW /*	ORD unsigned in			*/
/* /*	以下是输入键、	指示灯等设备变量定义,用户可以趋		*/
/*	- <i>P</i> ?^()·	/* 定义out0 7-P2 0-P1600 检出	指示灯00.07 的输出控制 贝尔哈板盾理图1	·*/ [16*/
sbit out8 14	$-120, 5-P2^{1}$	/* 定义 outo_/=12.0=11000, 制田 /* 定义 out8 15-P2 1-P1701 输出	指示灯 00-07 时期山庄前, 见关巡极床庄图 0 指示灯 08-015 的输出控制 囚空险板盾理图]]] */]]]]]7*/
shit inlad()	$7 - P^{2}$, $7 - P^{2}$.	/* 定义 inlado 7-P2 2-P18100 输。	加尔的 00-013 的输出控制,无关验做不足色 λ 指示灯 10.17 的输出控制	/ 01/ //
shit inled8	$15 - P^{2}$	/* 定义 intedo_/ =1 2:2=1 10100, #/	λ指示灯18-115 的输出控制 贝 尔哈板值理图	710 / 71/10*/
shit in0 7–	P^{1}	$/* \equiv V in0 7 - P2^{4} - P2 4 - P1210 4$	λ键10-17 的遗控制贝实验板原理图1112	///////////////////////////////////////
sbit in 8 15-	$-P2^{5}$	$/* \equiv V in 8 15 - P2^{5} - P2 5 - P1311$	输入键18_115 的读控制 <u>国实验标</u> 值理图111	3*/
sbit inadd-	-12 J, P2^6·	$/* \equiv V inadd - P^{-12} = 12.5 = 12.5 = 11511,$	PROFIBUS 抽址输入 ADO-AD7 的诗控制	, */
son madu-		/* 见实验板原理图U20		*/
sbit REQ_I	<i>T=P1^7;</i>	/* 定义 REQ_IT=P1^/=P1./=P1/S_ /* 贝实验板原理图CONNECT-16	CTS, 綱入, 连接 OEM2-SE-REQ_IT	*/ */
sbit S_RTS=	= <i>P1^6;</i>	/* 定义 S_RTS=P1^6=P1.6=P175_1	RTS, 输入, 连接OEM2-SE-S_RTS	*/
_				*/
sbit P15RE	S=P1^5;			
/********* bdata UBY1 sbit di0=di0	****** <i>定义:</i> di8 TE di0_7,di8_15; 0_7^0; 0_741:	_15:字节型,是输入键18-115 当前值; /* di0_7:字节型,输入键10-17;di8 /* 定义di0=di0_7^0=输入键10 /* 定义di1=di0_7^1=输入键10	************************/ _15: <i>字节型,输入键1</i> 8-115*/ */ */	
soli $al1=al0$	D_{1}^{1}	/* 正义 dl1=dl0_/~1= 删八键 l1 /* 定义 di2-di0 7^2- 检入键 l2	*/	
sbit $di2 - dil$)_7_2,	/* 定义di3_di0_7_2_初八碳12 /* 定义di3_di0_7^3_ 检入键13	*/	
shit di4=dil	$-7^{-}, 3^{-},$	/* 定义di4=di0_7^4= 输入键14	*/	
sbit di5=di() 7^5:	/* 定义 $di5=di0$ 7 ⁵ =输入键I5	*/	
sbit di6=di() 7^6;	/* 定义 di6=di0 7^6= 输入键 I6	*/	
sbit di7=di0)_7^7;	/* 定义 di7=di0_7^7= 输入键 I7	*/	
sbit di8=di8	8_15^0;	/* <i>定义 di</i> 8= <i>di</i> 8_15^0= <i>输入键</i> I8	*/	
sbit di9=di8	8_15^1;	/* <i>定义di</i> 9=di8_15^1= <i>输入键1</i> 9	*/	
sbit di10=d	i8_15^2;	/* <i>定义di10=di8_15</i> ^2= <i>输入键1</i> .	!0*/	
sbit dill=di	i8_15^3;	/* <i>定义di11=di8_15^3=输入键11</i>	1*/	
sbit di12=d	i8_15^4;	/* <i>定义 di12=di8_15</i> ^4= <i>输入键</i> L	*/	
sbit di13=d	i8_15^5;	/* <i>定义di13=di8_15</i> ^5= <i>输入键1</i> .	!3*/	
sbit di14=d	i8_15^6;	/* 定义 di14=di8_15^6=输入键L	*/	
sbit di15=d	i8_15^7;	/* 定义di15=di8_15^7=输入键L	*/	
/********	***** <i>定义: ke</i> y	0_7: <i>字节型,是输入键10-17 扫描读入</i>	<i>值</i> ; ************************/	
/*******	***** 定义: ke	8_15:字节型,是输入键18-115 扫描读	入 <i>值</i> ; ***************************/	
bdata UBY	TE key0_7,key8_1	5;		
sbit key0=k	ey0_7^0;	/* <i>定义</i>	0*/	
sbit key1=k	ey0_7^1;	/* 定义 key1=key0_7^1=输入键1	1*/	
sbit key2=k	ey0_7^2;	/* 定义 key2=key0_7^2=输入键1	2*/	
sbit key3=k	ey0_7^3;	/* 定义 key3=key0_7^3=输入键1	3*/	
sbit key4=k	ey0_7^4;	/* 定义 key4=key0_7^4=输入键1	4*/	
sbit key5=k	ey0_7^5;	/* 定义 key5=key0_7^5=输入键1	5*/	
<i>sbit key6=k</i>	ey0_7^6;	/* 定义 key6=key0_7^6=输入键1	6*/	
sbit key $7=k$	ey0_7^7;	/*	/*/	
sbit key $8=k$	ey8_15^0;	/* <i>正义</i> key8=key8_15^0= 綱入键	18*/	
sbit key9=k	ey8_13^1;	/* 正义 key9=key8_15^1=输入键	*/	

sbit key10=key8_15^2; /* 定义 key10=key8_15^2=输入键110*/ *sbit key11=key8_15^3;* -----*/ /* 定义 key12=key8_15^4=输入键112 *sbit key12=key8_15^4;* -----*/ /* 定义 key13=key8_15^5=输入键113 *sbit key13=key8 15^5;* ----*/ *sbit key14=key8_15^6;* /* 定义 key14=key8_15^6=输入键114 */ *sbit key15=key8_15*^7; /* 定义 key15=key8_15^7= 输入键115 -----*/ *sbit P00=P0^0;* sbit P01=P0^1; *sbit P02=P0^2; sbit P03=P0^3; sbit P04=P0^4: sbit P05=P0*^*5*; *sbit P06=P0^6*: *sbit P07=P0^7*; /************ PP00-PP07 是上面 P00-P07 的中间变量 *********************************/ bdata UBYTE PP0; *sbit PP00=PP0^0;* sbit PP01=PP0^1; sbit PP02=PP0^2; sbit PP03=PP0^3; sbit PP04=PP0^4; sbit PP05=PP0^5; sbit PP06=PP0^6: sbit PP07=PP0^7; *bdata UBYTE AB_status;* /**** AB_sD0=AB_status^0:用户参数更新标记 ****/ *sbit* AB_sD0=AB_status^0; /**** AB_sD7=AB_status^7:报文格式 ****/ *sbit* AB_sD7=AB_status^7; /**** 定义: req_com:字节型,请求数据命令,见《PB-OEM2-SE 产品手册》第四章、三、数据交换 *******/ *bdata UBYTE req_com;* /**** req_comD0=req_com^0: 请求数据/参数命令 ****/ *sbit req_comD0=req_com^0;* /**** req_comD1=req_com^1:清参数命令 ****/ *sbit req_comD1=req_com^1;* /*______*/ 输入键、指示灯等设备变量定义完毕 /* */ /*______ /*______ /* 以下是有关输入键、指示灯控制子程序,用户了解功能即可 */ /*______*/ *void steout0_7(UBYTE v)* { out0_7=0; *P0*=*v*; P0=v; P0=v; P0=v;out0_7=1; out0_7=0; ł *void steout8_15(UBYTE v)* { out8_15=0; P0=v; P0=v: P0=v; P0=v; out8_15=1;

```
out8_15=0;
}
void steinled0_7(UBYTE v)
{
inled0_7=0;
P0=v;
P0=v;
P0=v;
P0=v;
inled0_7=1;
inled0_7=0;
}
void steinled8_15(UBYTE v)
{
inled8_15=0;
P0=v;
P0=v;
P0=v;
P0=v;
inled8_15=1;
inled8_15=0;
}
UBYTE rd_address()
ł
UBYTE data x;
P0=0xff;
inadd=0;
x=P0;
x=P0;
x=P0;
x=P0;
inadd=1;
return(x);
}
void scanin()
{
P0=0xff;
in0_7=0;
PP0=P0;
PP0=P0;
PP0=P0:
PP0=P0;
if (key0_7!=PP0)
{
 if((PP00==1)\&\&(key0==0)) \{ di0=~di0; \};
 if((PP01==1)\&\&(key1==0)) \{ di1=~di1; \};
 if((PP02==1)\&\&(key2==0)) \{ di2=\sim di2; \};
 if ((PP03==1)&&(key3==0)) { di3=~di3;};
 if((PP04==1)\&\&(key4==0)) \{ di4=~di4; \};
 if ((PP05==1)&&(key5==0)) { di5=~di5;};
 if ((PP06==1)&&(key6==0)) { di6=~di6; };
 if ((PP07==1)&&(key7==0)) { di7=~di7;};
 key0_7=PP0;
];
in0_7=1;
inled0_7=0;
```

P0=di0 7; P0=di0 7: P0=di0 7: P0=di0 7; $inled0_7=1;$ inled0_7=0; P0=0xff;in8_15=0; *PP0=P0; PP0=P0; PP0=P0; PP0=P0*; *if* (*key8_15*!=*PP0*) { $if((PP00==1)\&\&(kev8==0)) \{ di8=~di8; \};$ $if((PP01==1)\&\&(key9==0)) \{ di9=\sim di9; \};$ $if((PP02==1)\&\&(key10==0)) \{ di10=~di10; \};$ $if((PP03==1)\&\&(key11==0)) \{ di11=~di11; \};$ *if* ((*PP04*==1)&&(*key12*==0)) { *di12*=~*di12*; }; *if* ((*PP05*==1)&&(*key13*==0)) { *di13*=~*di13*;}; $if((PP06==1)\&\&(key14==0)) \{ di14=~di14; \};$ *if* ((*PP07*==1)&&(*key15*==0)) { *di15*=~*di15*; }; *key8_15=PP0;* 1: in8_15=1; inled8_15=0; P0=di8 15; P0=di8_15; P0=di8_15; P0=di8_15; inled8_15=1; inled8_15=0; void Config(UBYTE *pp) /*----pp 是 user_prm[10]指针,本子程序是按照用户参数 user_prm[10] 设置 I/O 通道----*/ ł /* */ 主程序 /*_ _____ _*/ void main () { UBYTE data i,j,k,x,y,count,frist; UWORD data nw,nn; UBYTE xdata * data p; /*-----从站地址*/ UBYTE data address; UBYTE data re_er; /*-----字符偶校验错 -----*/ /*------ 串口波特率号*/ UBYTE data br_num; /*-----校验和错 UBYTE data error;*/ /*------ 串口接收缓冲 UBYTE xdata rebox[113];*/ /*----- 串口发送缓冲 UBYTE xdata trbox[113];*/ -----*/ UBYTE xdata pb_out[112]; /*-----PROFIBUS 输出数据区 -----*/ UBYTE xdata pb_in[112]; /*-----PROFIBUS 输入数据区 /*-----PROFIBUS 用户参数区 UBYTE xdata user_prm[10];*/ /*-----数据交换输入(实验板发送)数据长度 ------*/ UBYTE data data_in_len; /*-----不包括校验和的 data_in_len 长度 */ UBYTE data dil; /*-----数据交换输出(实验板接收)数据长度 -----*/ *UBYTE data data_out_len;* /*-----不包括校验和的 data_out_len 长度 -----*/ UBYTE data dol; /*-----用户参数长度 -----*/ UBYTE data user_prm_len; /*-----OEM2-SE 接收数据错误类型号 -----*/ UBYTE data se_er; P1=0xff;P0 = 0;

P2=0xf0;

di0_7=di8_15=0;

/* 实验板输出演示程序结束。

<i>steout0_7(0);</i>	/* <i>输出指示灯 Q0-Q7 清零</i>	*/
steout8_15(0);	/*输出指示灯Q8-Q15 清零	*/
steinled0_7(0);	/* <i>输入键指示灯 </i>	*/
<i>steinled8_15(0);</i>	/* <i>输入键指示灯 ILED</i> 8-ILED15	*/
/*=====================================		======*/
/* 以下是实验板上电后,检测 P89V51RD2	单片机内部RAM 的一段输出演示程序;	*/
/* 这段程序与 PROFIBUS 接口板 SE 通信法	<i>E关,用户可跳过此段程序。</i>	*/
/*		*/
<i>p=0;</i>		
x=1;		
k=1;		
<i>for</i> (<i>j</i> =0; <i>j</i> <32; <i>j</i> ++)		
{		
for $(i=0;i<24;i++)$		
{		
*(p+i)=0xaa;		
);		
<i>y=0;</i>		
for $(i=0;i<24;i++)$		
{		
count = *(p+i);		
<i>if</i> (<i>count</i> !=0 <i>xaa</i>) { <i>y</i> =1; };		
};		
if(y==0)		
{		
switch (k)		
1		
case 1:{ steout0_ $7(x)$;};break;		
case 2:{ steout8_15(x);};break;		
case 3:{ steinled0_7(x);};break;		
case 4:{ steinled8_15(x);};break;		
default:break;		
<i>};</i>		
<i>if</i> $(x < 255)$ { $x = x^{2}+1$; } <i>else</i> { $x = 1$; $k + +;$ };		
}		
else		
for (;;)		
{		
for (nw=0;nw<6000;nw++)		
{		
$steout0_/(0);$		
$steout8_{15}(0);$		
$steinlea0_7(0);$		
stemiea8_15(0);		
j;		
$\int O(nw - 0, nw < 0000, nw + +)$		
l steout0 7(0xff)		
steout8 = 15(0xff)		
steinlad0 7(0xff):		
steinled8_15(0xff):		
}.		
/, }.		
,, }.		
n-n+24		
$P - P + 2\tau$, for $(nw - 0.nw < 8000.nw + 1)$ [].		
<i>jor (nw=0,nw</i> <0000, <i>nw</i> ++) <i>[]</i> , }·		
,,		
/*	*/	

/*_____*/

*/

/*------输出指示灯00-07 清零 *steout0_7(0);* -----*/ /*------ 输出指示灯 08-015 清零 -----*/ steout8_15(0); /*------输入键指示灯 ILED0-ILED7 清零 -----*/ steinled0 7(0); /*------输入键指示灯ILED8-ILED15 清零 -----*/ steinled8_15(0); /*______* */ /* 清串口输入缓冲区 trbox[i],清串口输出缓冲区 rebox[i] /*______*/ *for* (*i*=0;*i*<113;*i*++) { rebox[i]=trbox[i]=0; }; *for* (*i*=0;*i*<112;*i*++) { *pb_in[i]=pb_out[i]=0;]*; /*______ 以下是串口初始化程序 /* */ /*______*/ /*------*实验板波特率选择5,115.2K*------*/ br num=5: /*-----根据波特率号设置T2------*/ switch (br_num) { case 1:{ TH2=0x0ff; TL2=0x0a0;RCAP2H=0x0ff;RCAP2L=0x0a0;};break; /*---9.6K----*/ case 2:{ TH2=0x0ff; TL2=0x0d0;RCAP2H=0x0ff;RCAP2L=0x0d0; };break; /*---19.2K--*/ case 3:{ TH2=0x0ff; TL2=0x0e8; RCAP2H=0x0ff; RCAP2L=0x0e8; }; break; /*---38.4K--*/ *case 4:{* TH2=0x0ff; TL2=0x0f0; RCAP2H=0x0ff; RCAP2L=0x0f0; }; break; /*---57.6K--*/ *case 5:{ TH2=0x0ff; TL2=0x0f8; RCAP2H=0x0ff; RCAP2L=0x0f8; }; break;* /*---115.2K-*/ *case 6:{* TH2=0x0ff; TL2=0x0fc; RCAP2H=0x0ff; RCAP2L=0x0fc; *}; break;* /*---230.4K-*/ case 7:{ TH2=0x0ff; TL2=0x0fe;RCAP2H=0x0ff;RCAP2L=0x0fe; };break; /*---460.8K-*/ default:break; }; /*_____*/ /* 注意:本例基于 PHILIP P89V51RD2 单片机: */ /* 波特率=14745600/16/(65536-T) */ /* 各种波特率9600: T=0xffa0, 19.2K: T=0xffd0, 38.4K: T=0xffe8, */ /* 57.6K: T=0xfff0, 115.2K:T=0xfff8, 230.4k:T=0xfffc, 460.8k=0xfffe*/ /* PCON=0;T2CON=0x34;/*----RCLK=1; TCLK=1; TR2=1; C/T2=0 -----*/ /*_____ ----*/ /*串口设置成模式3 即: 1 个启始位、8 个数据位、1 个偶校验位、1 个停止位; */ /* 0xd0=11010000B; SM0SM1=11; SM2=0; REN=1;TB8=0; RB8=0; TI=0; RI=0 */ SCON=0x0d0; */ /* 形成接口板的初始化报文。 /*_. _____ while (REQ_IT==1) Ł */ /* 形成接口板的初始化报文 /* */ rebox[]是串口发送数组,格式见《SE 手册》 /*_____

 data_in_len=4+1+1;
 /*---- 输入数据报文长度=4+1+1=6;

 user_prm_len=10;
 /*---- 用户参数长度=10

 data_out_len=4+1+1;
 /*---- 输入数据报文长度=4+1+1=6;

 trbox[0]=address;
 /*---- 输入数据报文长度=4+1+1=6;

 _____*/ _____*/ _____*/ -----*/

trbox[1]=0x06; trbox[2]=0x0fa;	/* BYTE1=ID 号高位 /* BYTE2=ID 号低位	*/ */
/*/*	<u>以下形成 I/O 配置数据</u>	*/
/* 本例: I/O 配置数据长度=2 /* 共: 2 byte in/2 byte out+1 v /*	;I/O 配置数据:0x31,0x70 word in/1 word out	*/ */ */
trbox[3]=2; trbox[4]=0x31; trbox[5]=0x70;	/*BYTE3=I/O 配置数据长度=2 /*BYTE4=I/O 配置数据: 0x31=2 byte in/2 b /*BYTE5=I/O 配置数据: 0x70=1 word in/1	*/ pyte out*/ word out*/
trbox[24]=data_in_len; trbox[25]=data_out_len; trbox[26]=user_prm_len;	/*BYTE24=data_in_len	*/ */ */
x=0; for (i=0;i<48;i++)		
x=x+trbox[i];	/*	*/
<i>];</i> <i>trbox[48]=x;</i>	/* <u>置校验和</u>	*/
/*====================================	=====================================	===========*/ 初始化报文 */ */
while (S_RTS==1) {};	/* P1.6=S_RTS=1:SE 没有准备好接收	*/
y=0; while (y==0)	/* 清初始化成功标志	*/
REN=0; for (i=0;i<49;i++) { TI=0;	/* 发送一次初始化报文共 49 字节	*/
ACC=trbox[i]; TB8=P; SBUF=trbox[i]; while (TI==0) {}; };	/* 形成偶校验位	*/
while (S_RTS==0) { };	/* SE 没有接收完毕	*/
/* <i>发送完毕、等待接收</i> /	回答*/	
RI=0;		
REN=1; $re er=0:$	/*	*/ */
for $(j=0;j<49;j++)$	/* 接受回答报文共49 个字节	*/
nw=0; while ((RI==0)&&(S_RTS	S==1)) { }; /* SE 正在发送、实验板正在接收一~	个字符*/
if (RI==1) {/* RI==1: 实验板接	·收到一个字符*/	
ACC=SBUF; $if(P!=RB8){re_er=1;}$ rebox[j]=SBUF; RI=0;	; /* 偶校验错,置偶校验错标志	*/ 爱冲区*/
/ else {/* (RI==0)&&(S_RT j=100; }	[S==0):SE 报文全部发送完毕而实验板还没有接收 /* 退出等待接收	r到预期的长度 */ */
};		

```
/*------ 接收回答报文结束
                       -----*/
if (j==49)
{ /*---- (j==49): 实验板接收回答报文长度正确 ------*/
                     /*---- 没有字符偶校验错?
if(re\_er==0)
                                                   -----*/
 {
  x=0;
  for (i=0;i<48;i++) /*---- 求校验和
                                                   ----*/
   {
    x=x+rebox[i];
   };
  if (x = rebox[48])
                      /*---- 校验和正确?
                                                   -----*/
   {
    */
     {
                     /*---- 初始化报文错误号N=0?
     if(rebox[1]==0)
                                                   .....*/
      {
      x=0;
                     /*---- byte2~byte47 都是0xaa?
      for (i=2;i<48;i++)
                                                -----*/
       {
        if(rebox[i]!=0xaa)
         {
         x=1;
         };
       ]:
       if(x==0)
       {
        y=1;
                       /*---- byte2~byte47 都是 0xaa,初始化成功,y=1,准备退出 while (y==0) ----*/
       }
       else
                      /*---- byte2~byte47 不都是0xaa! 错误标记=0x11;
       {x=0x11;};
                                                                  ----*/
      }
     else
                     /*---- 初始化报文错误号N<>0,错误标记=返回错误号
      {x=rebox[1];};
                                                                    ----*/
     }
    else
                     /*---- 返回的波特率号不是 5, 错误标记=0x77;
     {x=0x77;};
                                                                   ----*/
   }
  else
                     /*---- 校验和不正确正确,
                                        错误标记=0xff;
   {x=0xff;};
                                                                 ----*/
  }
 else
 {x=0xee;};
                      /*---- 发生了字符偶校验错 错误标记=0xee;
                                                                   ----*/
ł
else
{
                      /*---- j!=49 发生了接收报文长度错 错误标记=0xee;
                                                                   ----*/
 x=0xcc;
];
if(y==0)
                      /*---- 初始化失败
                                      ---- */
 {
  for (nw=0;nw<10000;nw++)
  {
  steout0_7(br_num);
                    /*---- 显示返回错误号 ----*/
  steout8_{15(x)};
  };
 ];
}; /*---- 如果 y==0, 返回 while (y==0) ----*/
/*_____
===*/
/*
           实验板对PROFIBUS 接口板初始化成功,进数据交换状态的无限循环中
                                                                  */
/*
                                                                 */
           trbox[]/rebox[]是串口发送/接收数组,发送/接收格式见《SE 手册》
                           */
                       /*---- data_in_len=50, dil=49
                                                               ----*/
```

dil=data_in_len-1;

dol=data_out_len-1; for (i=0:i <data_in_len:i++)< th=""><th>/* data_out_len==50, dol==49</th><th>*/</th></data_in_len:i++)<>	/* data_out_len==50, dol==49	*/
$\begin{cases} free (i1-0) \\ free (i1-0) $	/*	*/
1100x[1]=0, }.	/ /月及达、 这状态/中区	
for (i=0;i <data_out_len;i++)< td=""><td></td><td></td></data_out_len;i++)<>		
{ rabor[i]=0;	* 法生法 按断经冲区	*/
}·	/~~~~ 用义达、按议场/中区	/
<i>J</i> ,		
/*========在与主站连追	恿得到用户参数之前实验 Ⅰ/O 采用以下默认配置========	*/
user_prm[0]=0x00;	/* AI_M: 1-5V	*/
user_prm[1]=0x01;	/* AI_D:无符号整型(0~65535)	*/
user_prm[2]=0x00;	/* 输入单端/双端 AI_CH:单端	*/
user_prm[3]=0x00;	/* 输出类型 AO_M: 1-5V	*/
user_prm[4]=0x01;	/* 输出数据类型 AO_D: 无符号整型 (0~6	(5535) */
user_prm[5]=0x00;	/* 输出单端/双端 AO_CH: 单端	*/
user_prm[6]=0x00;	/* 输入正/负逻辑 DI_PN: 正逻辑(高有效	*/
user_prm[7]=0x01;	/* 输入电压等级 DI_V: 24V/DC	*/
user_prm[8]=0x00;	/* 11/10/2014 DU_PN: 止逻辑(尚有9)	X)*/
$user_prm[9]=0x01;$	/*	*/
$p = \alpha user_prm[v];$		*/
Comig(p);	/*	*/
frist=1;	/*首次进入标记	*/
/*		*/
while (REO IT==0)		
{ /* (REO_IT==0):SE 处在	E数据交换状态*/	
if (frist==1)		
{		
req_com=0;	/*自次进入,	کر */
frist=0;	/*清自次进入标记	*/
}; trboy[0]=reg_com;		*/
trbox[1] = di0, 7;	/*····································	`/ */
trbox[2]=di8 15:	/ * 竹姫 10_1/ 1F / FROFIDUS 初八第1子) /* 悠健 18 115 作为 PROFIBUS 給入第2字节	*/
trbox[3]-di8 15.	/*将健 18_115 作为 PROFIBUS 输入第 2 子) /*	*/
trbox[4]=di0 7::	/*将键 10 17 作为 PROFIBUS 输入第 4 字节	, */
x=0:		
for (i=0;i <dil;i++)< td=""><td>/* 求校验和*/</td><td></td></dil;i++)<>	/* 求校验和*/	
{		
x=x+trbox[i];		
};		
trbox[dil]=x;	/* 置校验和*/	
REN=0;		
while (S RTS==1) {};	/*S RTS==1: 接口板没有准备好接收数据	*/
for (i=0;i <data_in_len;i++)< td=""><td>/* 发送数据交换报文共6字节</td><td>*/</td></data_in_len;i++)<>	/* 发送数据交换报文共6字节	*/
{		
T1=0;		
ACC=trbox[i];		<i></i>
TB8=P;	/* 形成偶役短位	*/
SBUF=trbox[1];		
winie (11==0) {};		
1,		
while (S_RTS==0) {};	/* SE 没有接收完毕	*/
/*		
/·	- 久区双箔ン沃尔入泊木,寺侍伎牧伎山牧 SL 的凹合。==: /* ra or 俚妨险举坛士	/*/ */
$10_{0} = 0;$ if (reg. com D00)	/ · 1℃_℃1 四化 型 阳化 心	*/
{		
y=data_out_len;		

```
}
```

```
else
 {
   y=user_prm_len+2;
 }:
RI=0;
                                   /*----- 允许接收
REN=1;
                                                                                         .....*/
                                                                                         -----*/
                                  /*----- 等待接收接口板的回答数据交换报文共 50 字节
for (j=0;j<y;j++ )
 {
    while ((RI==0)&&(S_RTS==1)) {}; /*---- SE 正在发送、实验板正在接收一个字符
                                                                                         .....*/
    if (RI==1)
     {/*---- RI==1:实验板接收到一个字符 -----*/
      ACC=SBUF;
                                  /*----- 偶校验错,置偶校验错标志
      if (P!=RB8) {re_er=1;};
                                                                                         -----*/
                                  /*----- 接收一个字符送串口输入缓冲区
      rebox[j]=SBUF;
                                                                                         .....*/
      RI=0:
     }
    else
     {/*---- (RI==0)&&(S_RTS==0):SE 报文全部发送完毕而实验板还没有接收到预期的长度 ------*/
                                  /*---- 退出等待接收
                                                                                        .....*/
      j=y+1;
     };
 };
/*___
          ===========接收完毕==
                                                                                          _*/
     ____
if (j==y)
{/*----(j==data_out_len):实验板接收回答报文长度正确 ------*/
if (re_er==0)
                                  /*----- 无偶校验错
                                                                                        */
 -{
  if (req_comD0==0)
   {/*---- 请求回答 A 格式 ----*/
     x=0:
     for (i=0;i<dol;i++)</pre>
      ł
                               /*----- 求校验和
        x=x+rebox[i];
                                                                                       .....*/
      }:
     if (x==rebox[dol])
      ł
       AB_status=rebox[0];
       if (AB_sD7==0)
                                /*----见头部说明: sbit AB_sD7=AB_status^7: 报文格式
                                                                                       ____*/
        {/*-----接收的确实是 A 格式-----*/
         y=dol-1;
                                /*----- y=4-1=3
                                                                                       ____*/
         for (i=1;i<dol;i++)
          {
           pb_out[i-1]=rebox[i];
          };
         req_comD0=AB_sD0;
                                /*---- 见头部说明: sbit AB_sD0=AB_status^0: 用户参数更新标记-----*/
                                 /*---- 若 AB_sD0=1, 令 req_comD=1, 下次请求 B 格式
                                                                                         ----*/
                                 /*---- 若 AB_sD0=0, 令 req_comD=0, 下次请求 A 格式
                                                                                         ____*/
                                 /*---- 见头部说明: req_comD1=req_com^1:清参数命令;
                                                                                         .....*/
         req_comD1=0;
                                 /*---- req_comD1=0;不清用户参数更新标记;
                                                                                         .....*/
         se_er=(AB_status/2)&0x03; /*---- AB_status(D2~D1)=SE 接口接收用户板报文出错标记
                                                                                         -----*/
         error=0:
        ł
       else
        {/*-----(AB_sD7!=0)接收的不是 A 格式-----*/
         error=0x85;
                                /*---- 接收格式 A 错
                                                                                          -----*/
        };
      1
     else
      {
                                /*---- 校验和A错
       error=0x81;
                                                                                         ____*/
      };
   }/***********(req_comD0==0)************/
  else
   {/**********(req_comD0==1)请求回答 B 格式***********/
     y=user_prm_len+1;
     x=0;
     for (i=0;i<y;i++)
```

```
ł
                          /*----- 求校验和
      x=x+rebox[i]:
                                                                           -----*/
     };
    if (x==rebox[v])
                            /*----- 无校验和错
                                                                             .....*/
     {
      AB_status=rebox[0];
                            /*----见头部说明: sbit AB_sD7=AB_status^7: 报文格式
       if (AB_sD7==1)
                                                                            -----*/
        {/*----- 接收的确实是 B 格式-----*/
         for (i=1;i<y;i++)
         {
           user_prm[i-1]=rebox[i];/*---- 传送新接收到的用户参数到 user_prm[10]
                                                                          -----*/
          };
                             /*---- 按照新用户参数设置 I/O 通道
         Config(p);
                                                                            -----*/
                               /*---- 令 req_comD=0,下次发报文请求回答 A 格式
                                                                           -----*/
         req_comD0=0;
                               /*---- 令 req_comD1=1; 下次发报文清用户参数更新标记
         req_comD1=1;
                                                                            ----*/
         se_er=(AB_status/2)&0x03; /*---- AB_status(D2~D1)=SE 接口接收用户板报文出错标记
                                                                           -----*/
         error=0:
        }
       else
        {/*----- 接收格式 B 错 -----*/
         error=0x86;
        };
     }
     else
     {/*----- 校验和 B 错 -----*/
      error=0x82;
     };
   }:
 else
 {/*----- 接收报文偶校验错 ------*/
   error=0x83;
 };
}/**************(j==y)*************************/
else
{/*----- 接收字符数不够------*/
 error=0x84;
};
/*_____*/
                              /*----- 输入键扫描
scanin();
                                                                                __*/
                             /*----- 根据输入键来决定显示内容
x=di0_7;y=di8_15;
                                                                             .....*/
if (x==0)
 {/*----- di_0~7 输入为 0 或 !=1, 显示 PROFIBUS 输出-----*/
   if (y==0)
   {
/*----- 如果 di_8~15 输入为 0,PROFIBUS 输出数据第 1 个字节送 Q0-Q7 显示,第 2 个字节送 Q0-Q7 显示 -----*/
    steout0_7(pb_out[0]);
    steout8_15(pb_out[1]);
    };
   if (y==1)
   {
/*----- 如果 di_8~15 输入为 1,PROFIBUS 输出数据第 4 个字节送 Q0-Q7 显示,第 3 个字节送 Q0-Q7 显示 -----*/
    steout0_7(pb_out[3]);
    steout8_15(pb_out[2]);
    };
   if (y>1)
    {/*----- 如果 di_8~15 输入 >1, Q0-Q15 显示 0xff -----*/
    steout0_7(0xff);
    steout8_15(0xff);
    };
 };
```
```
if (x==1)
 {/*----- di_0~7 输入为 1,显示当前用户参数-----*/
   steout0_7(0); /*----- Q0-Q7 显示 0 -----*/
   if (y<10)
    {
     steout8_15(user_prm[y]);
    }
   else
    {/*----- 如果 di_8~15 输入 >= 10, Q0-Q15 显示 0xff -----*/
     steout0_7(0xff);
     steout8_15(0xff);
    };
 };
if (x>1)
 { /*----- 如果 di_0~7 输入 >1, Q0-Q15 显示 0xff -----*/
     steout0_7(0xff);
     steout8_15(0xff);
 };
};/*---- while (REQ_IT==0):SE 处在数据交换状态,返回数据交换状态的无限循环中 ------*/
};/*----(REQ_IT==1):SE 退出数据交换状态,返回初始化
                                                             -----*/
```

}

第五章 建立一个调试实验系统

OEM2 调试实验系统是一个最小化的 PROFIBUS 现场总线系统,本手册推荐的主站有 2 种选择。

① 选择 CP5611+PC 机做主站,本手册暂称为系统 I。

②选择西门子公司 PLC(CPU313-2DP或 CPU315-2DP)做主站,本手册暂称为系统 II。

本章将详细介绍如何建立 OEM2 调试实验系统 I 或系统 II 、如何实现 PROFIBUS 系统配置和介绍 本手册例 1、例 2、例 3 连通及演示实验。

1. 建立 OEM2 调试实验系统 I

(1) 系统 I 设备清单

	从站			
序号	名称	制造商	数量	备注
1	PB-OEM2-SE 嵌入式 PROFIBUS 接口	鼎实科技	1块	
2	PB-OEM2-SAMPLE OEM2 开发实验板	鼎实科技	1块	
	主站			
3	CP5611 PROFIBUS 主站网卡 (PCI)	西门子	1块	
4	PC 机: Win2000 操作系统、PCI 插槽	*****	1台	工控机、兼容机均可
	网络器件			
4	PROFIBUS 电缆	LAPP	10 米	
5	PROFIBUS 插头	西门子	2 个	
	软件			
6	组态软件(光盘):	西门子	3CD	随系统赠送
	COM PROFIBUS V5.1(DEMO 版); A#光盘			
	SOFTNET-DP V5.2(DEMO版,包含在 SIMATIC NET B#光盘中);			
	WinCC V5.1(DEMO版); C#光盘			
	文件资料			
7	文件资料(D#光盘)包括:	鼎实科技	1CD	随系统赠送
	GSD 文件			
	实验板硬件原理图			
	实验板软件清单(C51 源代码)			
	PROFIBUS 主站系统组态及调试实验软件			
	《PB-OEM2-SE 产品手册》			
	《PROFIBUS-OEM2 调试实验系统使用手册》			
8	手册 (印刷品):	鼎实科技	1套	随系统赠送
	《PB-OEM2-SE 产品手册》			
	《PROFIBUS-OEM2 调试实验系统使用手册》			

	用户自备			
序号	名称	制造商	数量	备注
1	PC 机: Win2000 操作系统、PCI 插槽	*****	1台	工控机、兼容机均可
2	24VDC(2A 以上)直流电源	****	1台	

(2) 系统 I 结构图



图 5-1 系统 I 结构图

(3) 安装实验调试系统 I

(-) 硬件安装

① CP5611: 按照 CP5611 说明书安装在 PC 机的 PCI 槽中;

② 安装开发实验板 PB-OEM2-SAMPLE 及嵌入式 PROFIBUS 接口 PB-OEM2-SE。注意 24VDC 电源的 正负极性。

③ 在实验板上设定 PROFIBUS 从站地址,为与本手册的例 1~例 3 统一,将开发实验板的从站地址设置成: 19(13H)号从站。

④ PROFIBUS 电缆与插头的连接: B 是红色线。注意电缆屏蔽层与 PE 的连接。两个 PROFIBUS 插头的终端电阻都要打到 ON 位置。

(二) 主站 PC 机软件安装

根据本调试实验系统提供的 CD 光盘,依此安装如下软件,如有必要可参照 SIEMENS 公司关于这些软件的安装资料。PC 机系统要求: Microsoft Windows 2000:

- ① 安装 COM PROFIBUS V5.1 软件;
- ② PB SOFTNET-DP V5.2(DEMO 版);
- <u>注意:</u>

▼由于安装DEMOD 版,因此安装中不选择"执行授权",见下页图5-2。 ▼安装软件列表中只需要选择PB SOFTNET-DP V5.2 即可,见下页图5-3。



图 5-2 只安装 DEMO 板因此不选择"执行授权"

x 🔁 SIMATIC NET - Setup: (Please select the products to be installed PB PG-5412 V5.2 26 MB 🔺 SIMATIC PB EMS-5412 V5 2 30 MB PB SOFTNET-DP V5.2 21 MB Software PB SOFTNET-S7 V5.2 14 MB PB SOFTNET DP Slave V1.1 10 MB 10 MB DP-OPC-Server V2.1 Description SIMATIC NET PROFIBUS SOFTNET-DP/Windows NT 4.0 Version 5.2 Read me Destination directory: c:\siemens\simatic.net Browse. Required on c: 21 MByte Available on c: 12428 MByte < Back Next > Cancel

图 5-3 安装软件列表中只选择 PB SOFTNET-DP V5.2 即可

③ 安装 WinCC 软件;

<u>注意:</u>

▼序列号: 可键入 "0";

▼由于安装WinCC DEMOD 版,因此安装中选择"稍后执行授权";

(4) 按照系统提供的例 2 实现系统连通实验

(→) 在 COM PROFIBUS 配置软件中实现简单连通

① 拷备 GSD 文件、图标文件、例 2 配置文件

- ▼将 GSD 文件 OEM2_B2.GSD 拷备至 CPBV51\gsd
- ▼将图标文件 SE_B.DIB 拷备至 CPBV51\bitmaps

▼将例2配置文件SE_B2_CONF 拷备至 CPBV51\data

② 进入 COM PROFIBUS V5.1, 打开例 2 配置文件 SE_B2_CONF.pb5,如图 5-4 所示:



③ 检查 Setting the PG/PC Interface....确认 "COM_PROFIBUS →CP5611(COM PROFFUBUS),见图 5-5。



图 5-5 检查 Setting the PG/PC Interface....确认"COM_PROFIBUS →CP5611(COM PROFFUBUS)

④ 系统诊断:双击 19 号从站图标,弹出 Slave properties 窗口。选择 Diagnostics...,等待 PC 主站检测

从站,弹出 Busparameters of DP card 窗口设置波特率。通常设置成 187.5K,见图 5-6。



注意 1:弹出 Slave diagnosis: PB-B-OEM2-SE-B2 #19 <DP slave>,如果窗口中没有信息,见图 5-7: 主站 没有检测到 19#实验板从站。说明通信硬件连接有错误。只有弹出象"图 5-8: 主站检测到 19#实验板从 站"窗口,才说明通信硬件连接没有错误。

Station name: DP slave @> Station type: DP of alwe @> Station type: PB-OEM2-SE-B2 Order number: pb-oem2-s Device master file: DEM2_B2.GSD Control commands: SYNC-cepable FREEZE-capable Shared master. Param. assign. #1, CP 5x11, Master system(1) Mesponse monitoring Diagnostics Error rep DESlave diagnosis: PB-OEM2-SE-B2 #19 <dp stat.<="" td=""> Offline Diffline</dp>	PROFIBUS address:	19	OK
Station type: PB-0EM2-SE-B2 Help Order number: pb-oem2-s Configure Device master file: DEM2_B2.GSD Parameterize Control commands: SYNC-capable Shargd master Param. assign. #1, CP 5x11, Master system(1) Biagnostics Param. assign. #1, CP 5x11, Master system(1) Diagnostics Fror rep Cslave diagnosis: PB-0EM2-SE-B2 #19 <dp stal.<="" td=""> Offline</dp>	- Stat <u>i</u> on name:	DP slave <2>	Cancel
Order number: pb-oem2-s Configure Device master file: DEM2_B2.GSD Parameterize Control commands: SYNC-capable Shargd master Param. assign. #1, CP 5x11, Master system(1) Diagnostics Faram. assign. #1, CP 5x11, Master system(1) Diagnostics Kargd master. Shargd master. Param. assign. #1, CP 5x11, Master system(1) Diagnostics Kargd master. Qffline	- Station type:	PB-OEM2-SE-B2	<u>H</u> elp
Device master file: DEM2_B2.GSD Control commands: SYNC-capable PREEZE-capable Param. assign. #1, CP 5x11, Master system(1) Response monitoring Error repr DPSlave diagnosis: PB-DEM2-SE-B2 #19 <dp sla="" x<br="">Qffline</dp>	Order number:	pb-oem2-s	Configure
Control commands: SYNC-capable FREZE-capable FREZE-capable Param. assign. #1, CP 5x11, Master system(1) Response monitoring Error rep DPSlave diagnosis: PB-DEM2-SE-B2 #19 <dp d="" qffline<="" sla="" th="" ×=""><th>Device master file:</th><th>OEM2_B2.GSD</th><th>Perenetoriza</th></dp>	Device master file:	OEM2_B2.GSD	Perenetoriza
Param. assign. #1, CP 5x11, Master system(1) Diagnostics Response monitoring Itatus/control. Error rep DP_Slave diagnosis: PB-DEM2-SF-B2 #19 <dp sla<="" td=""> Offline</dp>	Control commands:	SYNC-capable	I ar ane cer i ze
<u>Response monitoring</u> <u>itatus/control.</u> <u>Brror rep</u> <u>D</u> ffline			Diamostias
<u>Response monitoring <u>Pracus control.</u> <u>Brror rep</u> <u>Slave diagnosis: PB-DEM2-SE-B2 #19 < DP sla <u>Offline </u></u></u>	-		
Error rep DPSlave diagnosis: PB-DEM2-SE-B2 #19 <dp sla<="" td=""> X </dp>	Param. assign.	#1, CP 5x11, Master system<1>	tatus/control
	Param. assign. <u>Response monitor</u>	#1, CP 5x11, Master system<1> ing	tatus/control.
	Param. assign. Image: Response monitor Error rep DESlave	#1, CP 5x11, Master system<1> ing diagnosis: PB-OEM2-SE-B2 #19 <dp sla<="" td=""><td>X</td></dp>	X
	Param. assign. ▼ Response monitor Error rep DP Slave	#1, CP 5x11, Master system<1> ing diagnosis: PB-OEM2-5E-B2 #19 <dp sla]<="" td=""><td>itatus/control.</td></dp>	itatus/control.
	Param. assign. <u>Response monitor</u> Error rep <u>Error rep</u>	#1, CP 5x11, Master system<1> ing diagnosis: PB-OEM2-SE-B2 #19 <dp sla<="" td=""><td>itatus/control.</td></dp>	itatus/control.
	Param. assign. ✓ <u>R</u> esponse monitor Error rep <mark>: DPSlave</mark>	#1, CP 5x11, Master system<1> ing diagnosis: PB-OEM2-SE-B2 #19 <dp sla<="" td=""><td>▼</td></dp>	▼
	Param. assign. ✓ <u>R</u> esponse monitor Error rep <mark>. DPSlave</mark>	#1, CP 5x11, Master system<1> ing diagnosis: PB-OEM2-SE-B2 #19 <dp sla<="" td=""><td>▼ □raguostres :tatus/control. □ X □ffline</td></dp>	▼ □raguostres :tatus/control. □ X □ffline

图 5-7 主站没有检测到 19#实验板从站

PROFIBUS address:	19	OK
Station name:	DP slave(2)	Cancel
Station type:	PB-0EM2-SE-B2	Help
Order number:	pb-oem2-s	Configure
Device master file:	DEM2_B2. GSD	Pereneterize
Control commands:	SYNC-capable	Taralle Certize
Error rep Hexadecim	iagnosis: PB-OEM2-5E-B2 #19 <dp sla="" <br="">nal output</dp>	_ D X Offline
0: 0: Standard-s	2 05 00 FF 06 FA specific diagnosis	
PNO ni Slave r Slave r	Imber of the slave: 1786 (0x06FA) iot ready for data exchange nust be reparamaterized	

图 5-8 主站检测到 19#实验板从站

注意 2: "主站没有检测到 19#实验板从站"的故障原因可能是:

▼实验板从站站号设置错误:不是19#(13H);

▼实验板从站硬件安装错误: 24V 电源指示灯(均是红色)VCC/OK、24V/OK、5V/OK、TXD、RXD 及OEM2-SE 接口板上的PB_F 都应该稳定点亮。

▼PROFIBUS 电缆插头安装错误: 电缆与插头的安装。

⑤ 系统连通

(i)同系统诊断,双击 19 号从站图标,弹出 Slave properties 窗口。选择 Status/control...,等待 PC 主站检测从站并弹出 Busparameters of DP card 窗口设置波特率。通常设置成 187.5K,见图 5-9。



图 5-9 系统连通-1

(ii) 主站检测到 19#从站后, 弹出窗口 "PB-OEM2-SE-B2 #19 < DP Slave<2>>" 及稍后弹出 "Note" 窗

口,选择"确定"。见图 5-10,系统连通-2。

Fe	COM PI	ROFIB	JS - [SE_B2	2_CONF.pb!	5]																			_ 8	×
P	<u>F</u> ile E	Edit ⊻i	ew <u>C</u> onfig	ure <u>S</u> ervice	e <u>D</u> ocumentati	on <u>O</u> ptions	<u>W</u> indow	2																_ 8	×
C	ן 🚘 ו		X 🖻 🖻	3 🖗	활 왕 💡																				
E	<u>.</u> 	DP N	laster			××																			^
	FMS		lave lave prope	rties						-			×												
Ē		P	ROFIBUS ad	dress:	19						OK Cance	1					弾出窗□	1"	PB-0	DEN	л 2-8	SE-B	2 #1	19	
		S [.]	tat <u>i</u> on nam tation typ	e: e:	DP slave<2> PB-OEM2-SE-H	32			_		Help						<dp slav<="" td=""><td>e<2</td><td>>>"</td><td></td><td></td><td></td><td></td><td></td><td></td></dp>	e<2	>>"						
		01	rder numbe	r:	pb-oem2-s					Cor	fiour		1			L			_						J I
		D	evice mast	er file:	OEM2_B2. GSD					Para	meteri	ze													
					SYNC-caps	able				-			11				-								
쁟	PB-OE	M2-SE	-B2 #19 <	<pre>CDP slave</pre>	:2>>										_		-								
	Iden	tifie	r Connent	I addres	s O address	I format	00 00	00 00	0.0	00 00	Ιηρτ	ts of		00 00			0 format							Out	puts
1	047					hn	00 00	00 00	00	00 00	00	00 00	, 00	00 00	00	00 00	KH	00	00	00	00	00 00	00	00	00
2	031					KH	00 00	00 00	00	00 00	00	00 00	00 (00 00	00	00 00									
3	047					VU.	00 00	00 00	0.0	00 00	0.0			00 00	0.0	00 00	KH	00	00	00	00	00 0	00 (00	00
5	031					лп	00 00	00 00	00	00 00	00	00 00	,	00 00	00	00 00	KH	00	00	00	00	00 0) 00	00	00
																									Þ
						Note										×I									\Box ,
															_										
						٤	The DF Please	slave ha control th	is not b he stati	ieen assi ion.	gned a	param	eter as:	signmenl	t master		稍后弹出	"N	ote"	窗	口,	选择	"确	定"。	
										确定	_		_												
																-									•
×[_					
									图 5	5-10	系	统ì	车通-	-2											

(iii) 主站弹出 "Contro values for position 1" 窗口,选择 "Control",见图 5-11,系统连接 3。

	IBUS - [SE_B2_CONF.pb! View Configure Service	5] B. Documentation C	ntions Window ?					
	% 🖻 🖻 🎒 🌮	 꽓 돰 ?						
)P Master)P Slave	<u> </u>						-
	gslave properties	_						
	PROFIBUS address: Station name:	19 DP_slave(2)		Cancel				
	Station type:	PB-OEM2-SE-B2		Help				
	Order number: Device mester file:	pb-oem2-s DEM2_B2_GSD		Configure				
	C	SYNC-capable		Parameterize				
PB-OEM2-	-SE-B2 #19 <dp slave<<="" th=""><th>:2>></th><th></th><th></th><th></th><th></th><th></th><th></th></dp>	:2>>						
0 031	ier Comment I addres	S O address I f	ormat 00 00 00 00 00 0	Inputs 10 00 00 00 00 1	00 00 00 00	0 format 00		Outputs
1 047						кн	00 00 00 00 00 00 0	0 00 00
2 031		KH	00 00 00 00 00 0	0 00 00 00 00 00 0	00 00 00 00	00 KH	00 00 00 00 00 00 0	0 00 00
4 031		KH	00 00 00 00 00 0	0 00 00 00 00 00	00 00 00 00	00		
5 047					-	KH	00 00 00 00 00 00 0	0 00 00
								<u>P</u>
			Control values for positi	ion 1 X	_			
			Identifier Value	Control		之 寺上路山」 44.	C	
			047 00	Cancel		土珀坪山	Contro values for positi	ion
			00	Help		1"窗口,送	适择"Control"	
			00	Delete	L			
			00	Position -				_
×			Format: KH	 Position + 				-

图 5-11 系统连通-3

(iv) 系统连通并观察 PROFIBUS 输入。见图 5-12 系统连通-4,在窗口 "PB-OEM2-SE-B2 #19 <DP Slave<2>>"中可以看到 "Inputs" — PROFIBUS 的 48 字节输入:除 I0、I1 字节是实验板键输入,其余 I2~I47 均为 5AH,见例 2:实验板程序清单中有如下程序段便可得知:

trbox[1]=di0_7; /*---- di0_7 是输入键输入10~17 当前值,作为PROFIBUS 输入的第1 字节 -----*/ trbox[2]=di8_15; /*---- di8_15 是输入键输入18~115 当前值,作为PROFIBUS 输入的第2 字节 -----*/ for (i=3;i<dil;i++) { trbox[i]=0x5a; /*---- 其它PROFIBUS 输入数据(3~48)本例定为0x5a ------*/ };

注意:系统连通的主要标志是: PB-OEM2-SE 接口板上的指示灯(红) PB_F 灭。

D	Control: PB-	OEM2-SE-E	32 #19 <dp< th=""><th>slave<2>></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>_</th><th></th><th>ĸ</th></dp<>	slave<2>>																									_		ĸ
Γ	Identifie	r Connent	I address	0 address	I format								Inj	puts								0 format								Outy	pτ
0	031				KH	00	00	5Å	5A	5Å	5Å	5A	5A	5Å	5A	5A	5Å	5A	5Å	5Å	5A										
1	047																					KH	00	00	00	00	00	00	00	00	C
2	031				KH	5Å	5A	5A	5A	5Å	5Å	5A	5A	5Å	5A	5A	5Å	5A	5Å	5Å	5A										Π
3	047																					KH	00	00	00	00	00	00	00	00	C
4	031				KH	5Å	5Å	5Å	5A	5Å	5Å	5A	5Å	5Å	5Å	5Å	5Å	5A	5Å	5Å	5Å										
5	047																					KH	00	00	00	00	00	00	00	00	C
Ŀ]	١
1																														_	1

图 5-12 系统连通-4 系统连通并观察 PROFIBUS 输入

(v)改变 PROFIBUS 输入:按实验板输入键,可在线观察 PROFIBUS 输入 I0、I1 的变化。见图 5-13。

D	Control: PB-O	EM2-SE-B	2 #19 <dp< th=""><th>slave<2>></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>_</th><th></th></dp<>	slave<2>>																									_	
Г	Identifier	Connent	I address	0 address	I format								Ing	outs								0 format								Outpr
0	031				KH	07	E0	5Å	5Å	5A	5A	5Å	5Å	5Å	5Å	5Å	5A	5A	5Å	5A	5A									
1	047																					KH	00	00	00	00	00	00	00	00 C
2	031				KH	5A	5Å	5Å	5Å	5A	5A	5Å	5Å	5Å	5Å	5Å	5A	5A	5Å	5A	5Å									
3	047																					KH	00	00	00	00	00	00	00	00 C
4	031				KH	5A	5Å	5Å	5Å	5A	5A	5A	5Å	5Å	5Å	5Å	5A	5A	5Å	5A	5A									
5	047																					KH	00	00	00	00	00	00	00	00 C
◀																														

图 5-13 系统连通-5 可在线观察 PROFIBUS 输入 IO、I1 的变化

(vi) 改变 PROFIBUS 输出:双击 PROFIBUS 输出,弹出 "Contro values for position 1"窗口,在Q0、Q1 位置置入:33、CC,选择 "Control";主站将Q0、Q1 值通过 PROFIBUS 送到实验板从站;观察实验板输出指示灯(红),对应显示00110011(右)、11001100(左)见图 5-14:系统连接-6。



图 5-14 系统连接-6 改变 PROFIBUS 输出

□ 自己完成配置例 2 实现简单连通

① 进入 COM PROFIBUS V5.1

② 配置主站: 主站选择 CP5×11、设定"Set Bus Parameters..",通常选择 187.5K,见图 5-15。



图 5-15 配置主站

③ 配置从站:选择"PB-OEM2-SE-B2",设置站号 19。见图 5-16,配置从站。

注意:如果没有找到"PB-OEM2-SE-B2",可能是忘记拷贝 GSD 文件到 OEM2_B2.GSD 至 CPBV51\gsd 或应该更新设备目录,见图 5-17 更新设备目录。

④ 配置文件存盘

⑤ 转到"(-) 在 COM PROFIBUS 配置软件中实现简单连通: ③ — ⑤"



图 5-17 更新设备目录

2. CP5611+Wincc 做主站实现系统 I 的 PROFIBUS 通信

例 1 与例 2 的 GSD 文件相同,因此 CP5611 和 WinCC 配置相同。

(1) 设置"Set PG/PC Interface...":

① NCM 文件: 将例 2 的 COM PROFIBUS 配置的 OEM2_SE_B2.ldb 文件拷贝 CPBV51/ncm;

或自主形成 OEM2_SE_B2.ldb, 方法见图 5-18 形成 NCM 文件。



图 5-18 形成 NCM 文件

② 设置"Set PG/PC Interface...":从"控制面板"中进入"Set PG/PC Interface...",选择:

"CP_L2_1: → CP5611(PROFIBUS-DP Master)"、进入"Properties.."、Database name 指向 COM PROFIBUS

形成的 NCM 文件。如果能够 Read 出 19#站及 48Input/48Output,说明 NCM 文件及 Set PG/PC Interface 配

置正确,见图 5-19。

◎ 控制面板
〕 文件(E) 編辑(E) 査看(Y) 收藏(A) 工具(I) 帮助(H) (日)
」 ← 后退 → → ・ 回 ② 機索 日 文件夹 ③ 历史 階 昭 × ∞ 囲・
」地址② ◎ 控制面板 🔽 🧷 狩到
① 选择: "CP_L2_1: → CP5611(PROFIBUS-DP Master)" ② 进入 "Properties" U ② 进入 "Properties" 电话和调制 电源连项 辅助功能选
Set PG/PC Interface Configuration of your Siemens communications interfaces 7 Configuration @ ① Database name 指向 COM PROFIBUS 形成的 NCM 文件。 Mindows Update ① CP_12_1:> CP5611 (PROFIBUS - DP Master) @ ② Database name 指向 COM PROFIBUS 形成的 NCM 文件。
Windows 2000 支持 O Interface Parameter Assignment Froperties CFS611 (PEOFIBUS - DP Master) Properties ECFS611 (PEOFIBUS - DP Master) Diagnostics ECFS611 (PEOFIBUS - DP Master) Diagnostics Database Content: Image CFS611 (PEOFIBUS - DP Master) Delete Database name: E:\siemens\ncm\SE_B2_CONF.ldb
User parameter assignment of rowse Browse your communications processor Image: Slave Type Interfaces Slave Type Add/Remove: Select 0K Cance
④如果能够 Read 出 19#站及 48Input/48Output,说明 NCM 文件及 Set PG/PC Interface 配置正确。
確定 取消 应用 (A) 帮助
Configuration of your Siemens communications interfaces
通知井畑川 (近) 以 (修) 」 (2) 投制面积 Set PG/PC Interface

图 5-19 从"控制面板"中进入"Set PG/PC Interface..."

(2) 建立 WinCC 项目

 将 WinCC 例 1-2 演示实验项目文件夹: TEST_OEM2_B12 拷贝到 WINCC\WinCCProjects\并将文件夹 属性由"只读" 设置成"存档"。

② 进入 WinCC、打开项目文件夹 TEST_OEM2_B12, 改计算机名。见图 5-20



图 5-20 进入 WinCC, 打开项目文件夹 TEST_OEM2_B12 并更改计算机名

(3) 运行 WinCC

使 WinCC 进入运行状态,见图 5-20。鼎实科技提供的例 1~2 实验项目 TEST_OEM2_B12 将有如下 调试画面,见图 5-21。例 1~2 实验项目 TEST_OEM2_B12 调试界面 I



图 5-21 例 1~2 实验项目 TEST_OEM2_B12 调试画面 I

进入"下一页",可以看到全部 48 BYTES INPUT/48BYTES OUTPUT 数据,见图 5-22:例 1~2 实 验项目 TEST_OEM2_B12 调试介面 II 。



图 5-22 例 1~2 实验项目 TEST_OEM2_B12 调试画面 II

3. 建立 OEM2 调试实验系统 II

(1) 系统Ⅱ设备清单

		从站			
序号	名称		制造商	数量	备注
1	PB-OEM2-SE 嵌入式 PROFIBUS 接口		鼎实科技	1块	
2	PB-OEM2-SAMPLE OEM2 开发实验板		鼎实科技	1块	
		主站			
3	S7-300 PLC/CPU313-2DP		西门子	1块	
4	MPI 编程电缆		西门子	1根	
		网络器件			
4	PROFIBUS 电缆		LAPP	10 米	
5	PPROFIBUS 插头		西门子	2个	
		软件			
6	组态编程软件(E#光盘):		西门子	3CD	随系统赠送
	STEP7V5.1+SP1(DEMO版)				
		文件资料			
7	文件资料(D#光盘)包括:		鼎实科技	1CD	随系统赠送
	GSD 文件				
	实验板硬件原理图				
	实验板软件清单(C51源代码)				
	PROFIBUS 主站系统组态及调试实验软件				
	《PB-OEM2-SE 产品手册》				

	《PROFIBUS-OEM2 调试实验系统使用手册》			
8	手册 (印刷品):	鼎实科技	1套	随系统赠送
	《PB-OEM2-SE 产品手册》			
	《PROFIBUS-OEM2 调试实验系统使用手册》			
	用户自备			
序号	用户自备 名称	制造商	数量	备注
序号	用户自备 名称 PC 机: Win2000 操作系统,用做编程器	制造商 *****	数量 1 台	备注 工控机、兼容机均可

(2) 系统 II 结构图



图 5-23 系统 II 结构图

(3) 安装实验调试系统Ⅱ

(一) 硬件安装

① S7-300/CPU315-2DP: 按照 S7-300 安装手册安装,接通 CPU315-2DP 24V 直流电源。注意电源极性 及 PE (接地)的连接。自备 PC 机做 S7-300 编程器, MPI 编程电缆一端连接 PC 机串口,另一端连接 CPU313-2DP 的 MPI 接口。

② 安装开发实验板 PB-OEM2-SAMPLE 及嵌入式 PROFIBUS 接口 PB-OEM2-SE;注意 24VDC 电源的 正负极性。

③ 在实验板上设定 PROFIBUS 从站地址。为适合本手册的例 1~例 3,现将开发实验板从站地址设置成: 19(13H)。

④ PROFIBUS 电缆与插头的连接: B 是红色线,注意电缆屏蔽层与 PE 的连接。两个 PROFIBUS 插头终端电阻都要打到 ON 位置。

(二) 主站 PC 机软件安装

根据本调试实验系统提供的 CD 光盘,依此安装 SIMATIC STEP 7 V5.1 软件;如有必要可参照 SIEMENS 公司关于软件的安装资料。PC 机系统要求: Microdoft Windows 2000

① 先安装 SIMATIC STEP 7 V5.1:运行 STEP7/Disk1\Steup.exe

STEP 7 Setup: User Registration	STEP 7 Setup: Authorization
STEP 7 Setup: User Registration SIMATIC Software Company Enter your name, company, and ID number. You can find the valid ID number for your product on the Software Product Certificate. Name: Company: opce ID number: 0000000000 ID:KHE 0000000000	STEP 7 Setup: Authorization X The SIMATIC packages can only be started with a valid authorization (copy protection). No valid authorization was found on your computer. If you want to carry out the authorization now: Insert the authorization diskette. Drive with authorization Drive with authorization A:\ Click "Authorize." If you want to carry out the authorization at a later date: You can use the AuthorsW program at any time. Click "Skip" now.

图 5-24 ID 号输入 0

图 5-25 安装 DEMO 版 Skip 授权

② 再安装 SIMATIC STEP 7 V5.1 ServicePack 1: 运行 STEP7/sp1\Disk1\Steup.exe

(4) 按照系统提供的例 2 实现系统连通实验

① 拷备 GSD 文件、图标文件、例 2 的 STEP 7 项目文件

- ▼将 GSD 文件 OEM2_B2.GSD 拷备至 Step7\S7data\gsd\
- ▼将图标文件 SE_B.bmp 拷备至 Step7\S7data\nsbmp\
- ▼将例 2 的 STEP 7 项目文件夹 SE_B2_step7 拷备至 Step7\S7proj\
- ② 进入 🚺 打开例 2 项目文件 SE_B2_STEP7, 见图 5-26。



图 5-26 进入 SIMATIC Manager 打开例 2 项目文件 SE_B2_STEP7

② 设置 "Set PG/PC Interface...",以便下载程序,见图 5-27。



图 5-27 设置 "Set PG/PC Interface...", 以便下载程序

④ 下载配置和程序,见图 5-28。配置和程序下载完成后,系统应该连通了。

_		
K	SILATIC Lanager - SE_B2_STEP7	
F	ile Edit Insert PLC View Options Window Help	
] 🖆 🏙 📰 😹 💷 🗣 🗣 🏗 🏥 💼 🔍 Ko Filter > 💽 🏹 🎇 🎒 🕅	
	SE_B2_STEP7 f:\siemens\S?p.:\Se_b2_st	
	E B2_STEP7 BM Hardware Dr. 5-2 DP	
	□ Crow 315-2 Lip · Lip	
	D Blocks	

图 5-28 下载配置和程序

⑤ 系统连通标志

配置和程序下载完成后,系统应该连通了。系统连通标志是: S7-300/CPU313-2DP 上的 BUSF 红灯 灭并且 OEM2 接口板上的 PBF 红灯灭,见图 5-29。



图 5-29 系统 [[结构图

⑥ 系统连通故障

i、检查实验板和 S7-300 PLC 电源。不妨对 PLC 和实验板从新上电。

注意:实验板上电后:VCC/OK、5V/OK、24V/OK、TXD、RXD都会稳定点亮,输入/输出指示灯会有一段指示灯流水显示;

ii、检查 PROFIBUS 插头、电缆安装;

iii、检查例 2 项目文件 SE_B2_STEP7 下载。

⑦ 运行演示程序:系统连通后,将 PLC 控制钮旋转至 "RUN"或 "RUNP"位置。此时,实验板应 该有指示灯演示。按下任意输入键,指示灯演示停,输出指示灯显示与输入键相对应。

(5) 在 STEP 7 中观察 PROFIBUS 输入输出变化

① 在 STEP 7 中打开项目 SE_B2_STEP7,选中 Blocks,打开变量表 VAT_1,如图 5-30、图 5-31。

SINATIC Manager - SE_B2_STEP7	
<u>F</u> ile <u>E</u> dit Insert P <u>L</u> C <u>V</u> iew <u>O</u> ptions <u>W</u> indow <u>M</u> elp	
SE_B2_STEP7 F:\SIETENS\S7proj\SE_B2_STEP7	1
St. ES. STEPT □ · · · · · · · · · · · · · · · · · · ·	
图 5-30 在 STEP / 甲打开项目 SE B2 STEP7,选甲 Blocks,打开变量表 VAT 1	

② 可以看到 PROFIBUS 输出第 1、2 个字节 QB1、QB2 的变化,应该和实验板上的输出指示灯 Q0~Q15 一致。这是因为在实验板程序中将 PROFBUS 输出的第 1、2 个字节送给指示灯 Q0~Q15 显示,见例 2: 实验板程序清单有如下程序段:

steout0_7(pb_out[0]);
steout8_15(pb_out[1]);

/*----- PROFIBUS 输出数据第一个字节送 Q0-Q7 显示 ------*/ /*----- PROFIBUS 输出数据第一个字节送 Q0-Q7 显示 ------*/

¥٧	ar - @'	VAT_	1					
Table	able Edit Insert PLC Variable View Options Window Help							
-								
9	60	-	74201					
16	®VAT_	1 0	DEM2_SE_STEP	7\SIMATIC 3	00(1)\CPU 315-2 DP\57 Program(1) ONLINE			
	📥 Ad	dress	Symbol Disp	Status value	Modify value			
31	IB	31	HEX	B#16#5A				
32	IB	32	HEX	B#16#5A	时以有到I KOT DUS 抽出为 1~46 十十 中 DI * D46, 共干 ID3~ID46 钟定 JAII,			
33	IB	33	HEX	B#16#5A	这是实验板程序置入的;见例2:实验板程序清单有如下程序段;			
34	IB	34	HEX	B#16#5A	for (i=3:i <dil:i++)< th=""></dil:i++)<>			
35	IB	35	HEX	B#16#5A				
36	IB	36	HEX	B#16#5A				
37	IB	37	HEX	B#16#5A	trbox[i]=0x5a; /* 具它PROF1BUS 物入数据(3~48)本例定为0x5a*/			
38	IB	38	HEX	B#16#5A				
39	IB	39	HEX	B#16#5A				
40	IB	40	HEX	B#16#5A				
41	IB	41	HEX	B#16#5A				
42	IB	42	HEX	B#16#5A	可以看到 PROFIBUS 输出第1、2 个字节 OB1、OB2 的变化,应该和实验板上的输入。			
43	IB	43	HEX	B#16#5A				
44	IB	44	HEX	B#16#5A	出指示灯 O0~O15 一致。这是因为在实验板程序中将 PROFBUS 输出的第1、2 个字			
45	IB	45	HEX	B#16#5A				
46	IB	46	HEX	B#16#5A	节送给指示灯 O0~O15 显示,见例 2:实验板程序清单有如下程序段:			
4/	IB	4/	HEX	B#16#5A				
48	IB	48	HEX	B#16#5A	steout0_7(pb_out[0]); /* PROFIBUS 输出数据第一个字节送 Q0-Q7 显示*/			
49	UB OD	1	HEX	8#16#00				
100	ŲВ	2	HEX	B#16#4U				
			I					

图 5-31 在 STEP 7 中观察 PROFIBUS 输入输出变化

③ 可以看到 PROFIBUS 输出第 1~48 个字节 IB1~IB48,其中 IB3~IB48 都是 5AH。这是因为在实验板

的程序中将 PROFIBUS 输入第 3~48 个字节全部置入 5AH。见例 2 实验板程序清单有如下程序段: *for (i=3;i<dil;i++)*

{ trbox[i] };	l=0x5a;		/* 其它PROFIBUS 输入数据(3~48)本例定为0x5a*/								
④ 见图	5-32,百	可以看到:	按下实验板	任意输入键 I0~I15,IB1、IB2 将随之变化。由	日于实验板程序将输						
入键 I0~I	15 两个	字节作为	PROFIBUS 新	俞入的第1、2字节,见例2程序中有如下程序段	,因此输入键 I0~I15						
的数据将	对应主	站中 IB1、	IB2 的数据								
trbox[1]=d	i0_7;		/* di0_7 ‡	是输入键输入10~17 当前值,作为PROFIBUS 输入的	第1 字节 */						
trbox[2]=d	i8_15;		/* di8_15	是输入键输入 I8~I15 当前值,作为 PROFIBUS 输入	的第2 字节 */						
Var - @VAT_ Table Edit Ins	1 ert PLC Varia	able View Option	os Window Help								
50			(1)\CDU 315-2 DD\ 571								
Address	Symbol Disp	Status value M	odify value								
1 IB 1	HEX	B#16#3C									
2 IB 2 3 IB 3	HEX	B#16#U3 B#16#54		■ DDOFIBUS 输入第1 2 字节 与实验板输							
4 IB 4	HEX	B#16#5A	\sim	I KOFIBUS							
5 IB 5	HEX	B#16#5A		▶ 入键 I0~I15 对应;按动实验板输入键							
6 IB 6	HEX	B#16#5A		10.115 可以现实到现在现象的数据变化							
7 IB 7	HEX	B#16#5A		10~115,可以观察到 IB1、IB2 的数据受化。							
9 IB 9	HEX	B#16#54									
10 IB 10	HEX	B#16#5A									
11 IB 11	HEX	B#16#5A									
12 IB 12	HEX	B#16#5A									
13 IB 13	HEX	B#16#5A									
14 IB 14	HEX	B#16#5A									
15 IB 15	HEX	B#16#5A									
16 IB 16	HEX	B#16#5A									
17 IB 17	HEX	B#16#5A									
10 IB 18		B#16#0A B#16#5A									
20 IB 20		B#16#54									
21 IB 21	HFX	B#16#5A									
22 IB 22	HEX			_							
,			:								

图 5-32 输入键 IO~I15 的数据将对应主站中 IB1、IB2 的数据

(6) PROFIBUS 输入/输出数据地址

在上述实验演示中,为什么 PROFIBUS 输出第 1~48 字节对应 QB1~QB48、PROFIBUS 输入第 1~48 字节对应 IB1~IB48? 这是配置项目文件 SE_B2_STEP7/Hardware Config 时系统自动分配的。见图 5-33 进入 Hardware Config ,图 5-34 Hardware Config。

	SILATIC Lanager - SE_B2_STEP7		_ B ×
Fil	ile Edit Insert PLC View Options Window Help		
D			
	BSE_B2_STEP7 F:\SIEMENS\S7proj\SE_B2_STEP7	_ 🗆 🗵	
	E D SE_B2_STEP7 Bu Hardware CFU 315-2 DP		
	D Blocks		
	- EN DIOCKZ		

图 5-33 进入 Hardware Config

OD UK Profi Stundard 1 CPU 315-2 DP 2 CPU 315-2 DP 3 DD0/D06x24V/0.5A 5 CD0/D06x24V/0.5A 6 CD0/D06x24V/0.5A 7 CD0/D06x24V/0.5A 8 CD0/D06x24V/0.5A 9 CD0/D06x24V/0.5A 10 CD0/D06x24V/0.5A 9 SUBATIC 4D0	Netw Config - [SIIATIC 3 Network Edit Insert PLC Yiew □ C PL Network PLC Yiew	000(1) (Configur Options Lindow Hel Mar DF 2	ration)	SE_	B2_STEP7]	_	_5× _5×
(19) FB-OEM2-SE-52 S Module Order number 0 SI 40 Byte In, 40 Byte 0ut I16 1 47 40 Byte In, 40 Byte 0ut I16 2 SI 40 Byte In, 40 Byte 0ut I16 3 47 40 Byte In, 40 Byte 0ut II32 4 SI 40 Byte In, 40 Byte 0ut II32	>>>>>>>>>>>>>>>>>>>>>>>>>>>>			PROFIBUS	(1); DP master system (1)		i Standard FROUTUNG DF SIMATIC 300 SIMATIC 400 SIMATIC PC Based Control 300/400 SIMATIC PC Station
S. Q Module O cader number I kdd Q kddress Comment 0 31 40 Byte In, 40 Byte 0ut 116	(19) PB-OEM2-SE-B2						
0 31 40 Byte Da, 40 Byte Out 116 1 47 40 Byte Da, 40 Byte Out 116 2 31 40 Byte Da, 40 Byte Out 1732 3 47 44 Byte Da, 40 Byte Out 1732 4 31 40 Byte Ta, 40 Byte Out 1732	S I Module Order numbe	r I.	Add Q	Address	Comment		
I 47 48 Byte In, 48 Byte 0at I16 2 31 49 Byte In, 48 Byte 0at 1732 3 47 48 Byte In, 48 Byte 0at 1732 4 31 49 Byte In, 48 Byte 0at 1732	0 31 48 Byte In,	48 Byte Out 1	. 16				
2 31 49 Byte Ta, 49 Byte 0at 1732 3 47 49 Byte Ta, 49 Byte 0at 1732 4 31 49 Byte Ta, 49 Byte 0at 3349	1 47 48 Byte In,	48 Byte Out	1	. 16			
3 47 48 Byte In, 48 Byte Out 1732 4 31 48 Byte In, 48 Byte Out 3348	2 31 48 Byte In,	48 Byte Out 17.	32				
4 31 48 Byte In, 48 Byte Out 3348	3 47 48 Byte In,	48 Byte Out	17.				
	4 31 48 Byte In,	48 Byte Out 33.	48				
5 47 48 Byte In, 48 Byte Out 3348	5 47 48 Byte In,	48 Byte Out	33.	48			

图 5-34 Hardware Config

系统自动为实验板分配了输入地址 IB1~IB48,对应 PROFIBUS 输入第 1~48 字节、分配了输出地址 QB1~QB48,对应 PROFIBUS 输出第 1~48 字节。如果人为将实验板 PROFIBUS 输入地址改为 IB11~IB58, 如图 5-35 所示,那么 IB11 将对应 PROFIBUS 输入第 1 个字节、IB58 将对应 PROFIBUS 输入第 48 个字 节。同样,如果人为将实验板 PROFIBUS 输出地址改为 QB21~QB68,那么 QB21 将对应 PROFIBUS 输出第 1 个字节,QB68 将对应 PROFIBUS 输出第 48 个字节。



图 5-35 手动设置 PROFIBUS I/O 地址后的对应关系

(7) 自己完成例 2 配置实现系统连通实验

① 确认: GSD 文件 OEM2_B2.GSD 拷备至 Step7\S7data\gsd\、图标文件 SE_B.bmp 拷备至 Step7\S7data\nsbmp\。

新键项目文件 SE_B2_T1。见图 5-36。 2) 进入

SILATIC Lanager - SE_B2_STEP7			- 6 ×
<u>File Edit Insert PLC V</u> :ew Options <u>W</u> :ndcw <u>H</u> elp			
<u>N</u> ew	Dtrl+N	Filter > V PR N	
'Fev Project' Wigard			
<u>D</u> pen	Ctrl+D		
Jpen Vers <u>i</u> on : Project			
<u>]</u> lose			
57 Renory Card	+	New	
Wemory Card <u>F</u> :le	•	These environment of the second	
Zeve As	Ctr1+S	Name Storage path	
Delete		SE_B2_STEP7 f:\siemens\S7proj\Se_b2_st	
Reorganize		SE_B2_STEP7 F:\SIEMENS\S7proj\SE_B2_STEP7	
<u>M</u> enage		t_m_w3 F:\SIBMENS\S/proj\t_m_w3 test_se_b3 f:\siemens\S7proj\test_se_	
Archive			
Retrie <u>v</u> e			
Print	•		
Page Setup		· · · · · · · · · · · · · · · · · · ·	
Lebeling fields		Nam_e: <u>T</u> ype:	
Print Secup		SE_B2_T1 Project V	
1 ST RS STRET (Project) R \STEMENS\STruci\SE RS STEP7			
2 SE B2 STRF7 (Project) f \sienens\S7proj\Se b2 st		Storage location (path):	
3 test so b? (Project) -= f:\sigmone\S"proj\test so		f:\siemens\S7proj <u>B</u> rowse	
4 t m x3 (Project) E \SIERENS\S7proi\t m x3			
		OK Cancel Help	
Exit	Alt-F4		

图 5-36 新键项目文件 SE_B2_T1

③ 插入主控制器(本例为 S7-300),见图 5-37。

SILLATIC	Lanager SE_B2	_11				
<u>F</u> ile <u>E</u> dit	Insert PLC View Option	s Yindow Help				
	St <u>a</u> tion 🕨	1 SIMATIC 400 Station	< No Filter >	- 7 5 8		
	Sub <u>n</u> et 🕨	2 SIMATIC 300 Station	·			
	Erogram 🕨	3 SIMATIC H Station				
	ST Software k	4 SIMATIC PC Station				
	ST Ditemate	5 Other station				
	UT Caleman h	6 SIMATIC S5				
	TU DOKOWATA /	1 PG/PC				
	Symbol Table		,			
	External Source					
Internet						1
	E_B2_T1 f:\si	emens\S7proj\Se_b2_	<u>_t1</u>		_ <u>_ </u> _ ×	
I ∓{	SE_B2_T1	T MPI(I)	ATIC 300(L)			



④ 进入 Hardware, 见图 5-38

SII .	ATIC Manager - SE_B:	2_T1			
<u>F</u> ile <u>E</u>	dit <u>I</u> nsert PLC <u>V</u> iew <u>O</u> ptio	ns <u>W</u> indow	Help		
				< No Filter >	
E	SE_B2_T1 f:\sie	mens\S7p	roj\Se_b2_t1		
	- AD SE_82_T1	Hardwar	'e		
	——————————————————————————————————————		Open Object	Ctrl+Alt+O	
				CtrltX	
				Ctrl+C	
			Paste	Ctrl+V	
			Delete	Del	
			Manage Multilingus	al Texts 🕨 🕨	
			Rename	F2	

图 5-38 进入 Hardware

⑥ 配置主站,见图 5-39。



图 5-39 配置主站

⑦ 配置从站 PB-OEM2-SE-B2,见图 5-40,PB-OEM2-SE-B2 所在目录:PROFIBUS DP/Additional Field

Devices/ I/O /PB-OEM2/PB-OEM2-SE-B2。

🔣 HW Config - [SIMATIC 300(1) (Configu	ration) SE_B2_T1]	
🛄 Station Edit Insert PLC Yiew Optio	ns <u>Wi</u> ndow <u>H</u> elp	_ B ×
D 2 - U	1111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 111 - 11	
O) UR PROFI 2 CPU 315 3 4 5 6 7 V	EBUS (1): DP master system (1) Properties - PROFIBUS interface PB General Parameters Address: 19 Transmission rate: 187.5 kbit/s Subnet: PROFIBUS (1) 187.5 kbit/s Properties Properties	Profi Standard Profi Standard
•	Delete	DS-IO VIPA PD0/ISPM2
PROFIBUS(1): DP master system		losch
PROFIBUS address	注意目录: PROFIBUS DP/Additional Field Devices/ I/O /F	PB-OEM2/PB-OEM2-SE-B2
	确定 取消 帮助	BK3120 ⊕ → BK3120 BM4-204-DX1 B→ → R5-NP1

图 5-40 配置从站 PB-OEM2-SE-B2

⑧ 系统 II 配置完毕,编译存盘,见图 5-41,注意系统自动分配的 PROFIBUS I/O 地址。



图 5-41 系统Ⅱ配置完毕,编译存盘

⑨ 建立变量表,用于观察 PROFIBUS I/O,见图 5-42、图 5-43。



图 5-42 建立变量表

Cancel Help

确定

🛍 Lo	nito	pring	and	lodifying	; Var	iables	<u> </u>	T1	- <u>SE_B2_T1</u>	STEATE	C 300()	1)\CPU	J 315\S	7 Progr	am(2)]				_ 8 ×
👪 Iat	ole H	Edit I	nsert PI	£ Varia <u>b</u> le	⊻iew	Options	Mindow	Help											_ 8 ×
-122		2 🖬	😂 🐰	B 🛍 🗠	Cx	×	2 N?	90	er 🗠 🗠 🗠	liler									
	Addre	ess Sur	nbol Disp	Status valu	e Modi	v value										 	 	 	
1	IB 0		HEX			,													
2	IB 1		HEX																
3	IB 2	2	HEX																
4	IB 3)	HEX																
5	IB 4		HEX																
6	IB 5		HEX																
/	IB 6	5 -	HEX																
8			HEX																
10	IB 9		HEX																
11	IB 10	n	HEX																
12	IB 11	1	HEX																
13	IB 12	2	HEX																
14	IB 13	3	HEX																
15	IB 14	4	HEX																
16	IB 15	5	HEX																
17	IB 16	6	HEX																
18	IB 17	7	HEX																
19	IB 10	8	HEX																
20	IB 20	0	HEX																
22	IB 21	1	HEX																
23	IB 22	2	HEX																
24	IB 23	3	HEX																
25	IB 24	4	HEX																
26	IB 25	5	HEX																
27	IB 26	6	HEX																
28	IB 27	7	HEX																
29	IB 28	8	HEX																
30	IB 23	9	HEX																
31	ID 31	1																	
33	IB 30	2	HEX																
34	IB 33	3	HEX																
35	IB 34	4	HEX																
36	IB 35	5	HEX																
37	IB 36	6	HEX																
38	IB 37	7	HEX																
39	IB 38 ID ∼	8	HEX																
40	ID 33	0	HEX																
41	ID 40	1																	
43	B 40	2	HEX																
44	IB 43	3	HEX																
45	IB 44	4	HEX																
46	IB 45	5	HEX																
47	IB 46	6	HEX																
48	IB 47	7	HEX		_														
49	QB I	0	HEX																
50	ųB	<u> </u>	HEX																
51					_														

图 5-43 建立变量表 IB0~IB47、QB0~QB1

⑩ 下载项目文件,见图 5-44。

SIMATIC Manager - [5E_82_71 e:\5IEMENS\5te	07\57proj\Se_b2_t1]	
Bile Edit Insert PLC View Options Window He	þ	_ 8 ×
	- 📰 💼 < No Filter > 💽 🍸 躍 🥃 🕅	
B SE_B2_11 STMATIC SOO(1) CFU 315-2 DP ST Program (1) Sources B Blocks	■ CPU 315-2 DP 下载项目文件	
	图 5-44 下载项目文件	1

⑩--① 系统连通后,在线观察变量表 IB0、IB1 跟随实验板按键的变化情况,IB2~IB47 均为 5AH。见图 5-45。这是因为:实验板程序将输入键 I0~I15 两个字节作为 PROFIBUS 输入的第 1、2 字节,将 PROFIBUS 输入第 3~48 个字节全部置入 5AH。见例 2 实验板程序清单有如下程序段:

for (i=3;i<dil;i++)

{ trbox[i]=0x5a; };	/* 其它PROFIBUS 输入数据(3~48)本例定为0x5a	*/
trbox[1]=di0_7;	/* di0_7 是输入键输入 I0~I7 当前值,作为 PROFIBUS 输入的第1 字节	*/
trbox[2]=di8_15;	/* di8_15 是输入键输入 I8~I15 当前值,作为 PROFIBUS 输入的第2 字 🦷	ち* /

	Monito	oring and	l Modifying ¥a	riables - [@VAT_1	SE_B2_T1\SIMATI	C 300(1)\CPU 315-2 DP\57 Program(1) ONLINE]	
Sec.	<u>T</u> able	<u>E</u> dit <u>I</u> n	isert PLC Va	ria <u>b</u> le ⊻iew Option:	s <u>W</u> indow <u>H</u> elp		_ <u>8 ×</u>
-02	l n	പ്പില			(으 의) 1	ON Art 10 60. 14. 14.	
	<u> </u>	ddress	Symbol	Display format	Status value	Modify value	
1	IB	0		HEX	B#16#1F		
2	IB	1		HEX	B#16#F8		
3	IB	2		HEX	B#16#5A		
4	IB	3		HEX	B#16#5A		
5	IB	4		HEX	B#16#5A		
6	IB	5		HEX	B#16#5A		
7	IB	6		HEX	B#16#5A		
8	IB	7		HEX	B#16#5A		
9	IB	8		HEX	B#16#5A		
10	IB	9		HEX	B#16#5A		
11	IB	10	¢	HEX	B#16#5A		
12	IB	11		HEX	B#16#5A		
13	IB	12	••••••••••••••••••••••••••••••••••••••	HEX	B#16#5A		
14	IB	13		HEX	B#16#5A		
15	IB	14	•	HEX	B#16#5A		
16	IB	15		HEX	B#16#5A		
17	IB	16	¢	HEX	B#16#5A		
18	IB	17		HEX	B#16#5A		
19	IB	18	•	HEX	B#16#5A		
20	IB	19		HEX	B#16#5A		
21	IB	20		HEX	B#16#5A		
22	IB	21		HEX	B#16#5A		
23	IB	22		HEX	B#16#5A		
24	IB	23		HEX	B#16#5A		
25	IB	24		HEX	B#16#5A		
26	IB	25		HEX	B#16#5A		
27	IB	26		HEX	B#16#5A		
28	IB	27		HEX	B#16#5A		
29	IB	28		HEX	B#16#5A		
30	IB	29		HEX	B#16#5A		
31	IB	30		HEX	B#16#5A		
32	IB	31		HEX	B#16#5A		
33	IB	32		HEX	B#16#5A		
34	IB	33		HEX	B#16#5A		
35	IB	34		HEX	B#16#5A		_

图 5-45 在线观察变量表 IB0、IB1 跟随实验板按键的变化, IB2~IB47 均为 5AH

M	onit	oring and	Modifying ¥a	riables - [@¥AT_1	SE_B2_T1\SIMATIO	300(1)\CPU 31	5-2 DP\57 Program(1) ONLINE]		
	able	Ealt In:	sert PLC val	riagie view Option:	s <u>window H</u> eip	_			그리스
-Pi) 🖻 日	🗿 👗 🖻		(<mark>° 2</mark> <u>\?</u>	🎯 😚 🛷	6°i 1°i 140		
							11.0 2 2 2 2 4 4 6 4		
	4	Address	Symbol	Display format	Status value	Modify value			
17	IB	16		HEX	B#16#5A		-		
18	IB	17		HEX	B#16#5A		1		
19	IB	18		HEX	B#16#5A		•		
20	IB	19		HEX	B#16#5A				
21	IB	20		HEX	B#16#5A				
22	IB	21		HEX	B#16#5A		1		
23	IB	22		HEX	B#16#5A				
24	IB	23		HEX	B#16#5A				
25	IB	24		HEX	B#16#5A				
26	IB	25		HEX	B#16#5A				
27	IB	26		HEX	B#16#5A				
28	IB	27		HEX	B#16#5A				
29	IB	28		HEX	B#16#5A				
30	IB	29		HEX	B#16#5A		-		
31	IB	30		HEX	B#16#5A		•		
32	IB	31		HEX	B#16#5A				
33	IB	32		HEX	B#16#5A				
34	IB	33		HEX	B#16#5A		·		
35	IB	34		HEX	B#16#5A		✓ Monitor	Ctrl+F7	
36	IB	35		HEX	B#16#5A		✓ Modify	Ctrl+F9	
37	IB	36		HEX	B#16#5A		Update Monitor Values	F7	
38	IB	37		HEX	B#16#5A		Activate Modify Value	F9	
39	IB	38		HEX	B#16#5A			-	
40	IB	39		HEX	B#16#5A		Modify Address to 1	Ctrl+1	
41	IB	40		HEX	B#16#5A		Modify Address to <u>U</u>	Ctrl+0	
42	IB	41		HEX	B#16#5A		Cut	Ctrl+X	
43	IB	42		HEX	B#16#5A		Copy	Ctrl+C	
44	IB	43		HEX	B#16#5A		Paste	Ctrl+V	
45	IB	44		HEX	B#16#5A				
46	ΙB	45		HEX	B#16#5A		Delete	Del	
47	IB	46		HEX	B#16#5A		Insert Range of Variables	Ctrl+K	
48	IB	47		HEX	B#16#5A		Modify/Force Value As Comment	F3	
49	QE	3 0		HEX	B#16#11	//B#16#11	Row Not Effective	Ctrl+B	
50	QE	31		HEX	B#16#00	B#16#33	1		
51									-
SE_B2	_T1\	SIMATIC 3	00(1)\\S7 Pro	ogram(1)					

⑩--② 强置 QB0~QB1 值,观察实验板输出指示灯的变化,见图 5-46。

图 5-46 强置 QB0~QB1 值,观察实验板输出指示灯的变化

可以看到 PROFIBUS 输出第 1、2 个字节 QB0、QB1 的变化,应该和实验板上的输出指示灯 Q0~Q15 一致。这是因为在实验板程序中将 PROFBUS 输出的第 1、2 个字节送指示灯 Q0~Q15 显示,见例 2:实验板程序清单有如下程序段:

steout0_7(pb_out[0]);
steout8_15(pb_out[1]);

/*----- PROFIBUS 输出数据第一个字节送 Q0-Q7 显示 ------*/ /*----- PROFIBUS 输出数据第一个字节送 Q0-Q7 显示 ------*/

4. 三种调试模式

(1) 调试模式 1: 实验板调试模式

① 主要调试功能

建立主站、从站的配置, PROFIBUS 系统的连同, PROFIBUS I/O 通信。用户可以先使用实验板按照本手册完成与 PROFIBUS 主站的连接,完成例 1~3 的实验,然后再将自主开发的用户板与 PROFIBUS 主站的连接。见图 5-47、图 5-48。



图 5-47 调试模式 1—实验板调试模式

图 5-48 调试模式 1—实验板调试模式

② MODE 开关设置

MODE=11000000(Bin);

③ 操作方法

本章讲解的全部安装、连通实验、例2程序演示都是调试模式1一实验板调试模式。

(2) 调试模式 2: OEM2 接口板调试模式

① 主要调试功能

通过 PC 机可以监测实验板 CPU 与 OEM2 接口板的通信数据,可检验初始化数据,PROFIBUS I/O 数据,用户参数等数据的正确性。这是检查、调试 OEM2 接口板的主要方法。见图 5-49、图 5-50 及图 5-51,使用 PC 机监测实验板 CPU 与 OEM2 接口板的通信数据。

② MODE 开关设置

MODE=11000010_{(Bin);}





PC 机监测数据

图 5-50 调试模式 2—OEM2 接口板调试模式

PROFIR



图 5-51 使用 PC 机监测实验板 CPU 与 OEM2 接口板的通信数据

③ 操作方式

I、设置 MODE 开关 MODE=11000010(Bin):

II、按照"图 5-51 使用 PC 机监测实验板 CPU 与 OEM2 接口板的通信数据"组成系统,本系统包括一 根 RS232 电缆。

III、打开 PC 机串口测试程序: 设置波特率: 115200; 校验位: 偶 EVEN; 数据位: 8; 串口<u>测试</u>. lnk 停止位:1;"十六进制显示";





图 5-52 观察"串口测试"程序接收数据

"保存显示数据"可以详细分析 OEM2 与实验板初始化及数据交换过程,参考本手册"第四章 例 2、

实验板程序清单"及《PB-OEM2-SE产品手册》。

①实验板初始化 SE 接口板数据:

注解:站号=13H、ID 号=06FAH、I/O 配置数据长度=06、I/O 配置数据=1F,2F,1F,2F,1F,2F、用户参数长度=0、 发送数据长度=32H、接收数据长度=32H、校验和=67H;

②SE 接口板回答初始化成功报文数据:

注解: 波特率号=05H、错误号=00H、校验和=91H

③实验板发送数据交换报文:

注解: req_com=0, 请求接口回答 PROFIBUS 输出数据; PROFIBUS 输入 1~2=00, 00, 与实验板输入键状态有关; PROFIBUS 输入 3~48=5AH, 与实验板程序有关;

校验和=2CH

④SE 接口板回答 PROFIBUS 输出数据:

V、先开主站,再打开实验板;观察"串口测试"程序接收数据,用户可以看到 PROFIBUS 输出数据。

VI、用例3实验,可以看到用户参数数据。

(3) 实验板程序下载

① 主要调试功能

可将实验板 C 源程序编译、连接生成 .HEX 目标程序,比如系统提供的例 1 (B1.HEX)、例 2 (B2.HEX)、例 3 (B4.HEX)通过软件 WINSIP 下载到实验板。用户可以自行修改实验板的程序,实现 不同 PROFIBUS 配置的连接,见图 5-53。



图 5-53 实验板程序下载

② MODE 开关设置

MODE=00000111_{(Bin);}

③ 操作方式

见附件 A: 下载实验板程序操作指南及下载线制作方法

(4) 调试模式 3: PC 机调试模式

① 主要调试功能

"调试模式 3" 是使用 PC 机替代实验板 CPU,见图 5-54、图 5-55 所示。PC 机运行 OEM2 调试软件: oem2_tools.exe,实现机与 OEM2 接口板的数据交换。这种调试模式避免下载程序(至实验板 CPU), PC 机上可方便设定不同初始化数据和 PROFIBUS 输入数据,也可以监测主站的 PROFIBUS 输出数据、用户参数数据,是学习、调试 OEM2 接口板的另一种方法。







图 5-55 调试模式 3—PC 机调试模式

② MODE 开关设置

MODE=00100000(Bin)

③ 操作方式

I、按图 5-54 或图 5-55 连接系统;

II、主站应先做好配置,注意:主站配置的GSD 文件与下面的初始数据应该一致。主站部分可以先启动

或随后启动,取决于调试目的,实验板电源上电。注意:此时实验板 CPU 还处于运行状态,但串口已经 和 OEM2-SE 接口断开。因此"PC 调试模式"与实验板 CPU 程序(甚至 CPU、晶振的有无)无关。 III、PC 机上安装 OEM2_tools 软件(OEM2_ToolsV2.exe 安装),启动 OEM2_tools.exe,见图 5-56。 IV、在"属性"中设置串口,然后"打开端口",见图 5-56。

🛢 嵌入式总线桥开发调试系统———北京鼎实创新科技有限公司				_ & ×
控制(C) 端口(P) 帮助(H)				
▲ [] <u>开启端口(0)</u>				
②打开端口	🛢 端口设置		>	<u>s</u>
	通讯端口	1 💌		
	波特率	115200 💌	字长 8	
	校验	偶校验	停止位 1	
①在"属性"	<u> </u>			
中设置串口		确认		

图 5-56 启动 OEM2_toolsV2.exe、在"属性"中设置串口,然后"打开端口"

- V、首先进入初始化,见图 5-57;
- i、左边出始化报文缺省数据是按照例2内容组织,用户也可以自行修改。
- ii、按"数据确定"自动计算"校验和"

🖻 嵌入式总线桥开发调	【系统———北京鼎实创新科技有限公司		그리지
控制(<u>C</u>) 端口(P) 帮助()	■ 通讯初始化		
初始化(1)	┌初始化报文────		┌接收报文────● ▲
数据交换() \	地址 数据	地址 数据 说明	地址 数据 地址 数据
退出(区)	0 13 站号	24 32 接收数据长SD_in_len=50	0 5 24 AA
	1 06 ID号高位	25 32 发送数据长SD_out_len=50	1 0 25 AA
	2 FA ID号低位	26 0 用户参数长度User_Prm_Dada_len=0	2 AA 26 AA
	13 06 I/O配置数据长CFG_LEN<=10	0 27 0 ***不用	3 AA 27 AA
①首先进入初始化	4 1F I/O配置数据1	28 0 ***不用	4 AA 28 AA
	5 2F I/O配置数据2	29 0 ***不用	5 AA 29 AA
	6 1F I/O配置数据3		6 AA 30 AA
	7 2F I/O配置数据4	ふ ****不用	7 AA 31 AA
	8 1F I/O配置数据5	@ 按照例 2 由 密 组 组 的 山 始 化 坦 文 姑	8/ AA 32 AA
	9 2F I/O配置数据6	©按照例2内谷组织的田知化放义味	9 AA 33 AA
	10 0 1/0配置数据7	省数据,用户也可以自行修改。 / /	10 AA 34 AA
	11 0 1/0配置数据8	י	11 AA 35 AA
	12 0 1/0配置数据9	36 回 ***不用	12 AA 36 AA
	13 0 1/0配置数据10	37 回 ***不用 /	13 AA 37 AA
	14 0 1/0配置数据11	38 0 ***不用 ⑤回答初始化	14 AA 38 AA
	15 0 1/0配置数据12	39 0 ***不用 成功报文	15 AA 39 AA
	16 0 I/O配置数据13	40 0 ***不用	16 AA 40 AA
	17 0 1/0配置数据14	41 0 ***不用	17 AA 41 AA
	18_0 I/O配置数据15	42 0 ***不用	18 AA 42 AA
	19 3按"数据确定"自动计算"校	验知"	19 AA 43 AA
		④按 "发送数据", 观察右	20 AA 44 AA
	21 0 1/0配置数据18		-21-0
	22 0 1/0配置数据19		⑥主站与从站连通,
	23 0 1/0配置数据20		进入数据交换
		48 67	
端口选择:1 设置:	#610-451		3
第开始 3 3 6 6 6 7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		及达报义	<u>۱0:45</u>

图 5-57 首先完成初始化

iii、按"发送数据",观察右边"接收报文"。

▼如果根本没有回答(从实验板通信指示灯上也可以判断),检查波特率设置、OEM2-SE 接口上的波特率设置开关 SW 及初始化报文中的"站号"与实验板上"站号"设置是否一致。

▼如果"接收报文"是初始化成功报文,此时打开主站(或主站已经打开),那么,实验板上 OEM2-SE 接口板上的 PBF 红灯应该灭,表示主站已经和实验板连通。否则,检查主站配置(GSD 文件)、初始化 报文及回答报文数据和错误代码。





图 5-58 单次数据交换

i、单次数据交换:见图 5-58。

▼发送数据:第0字节是"请求数据命令 req_com =0",例2中必须是0。见本手册第四章例2,第1~48 字节 PROFIBUS 输入,用户可自由改变输入数据。"发送数据确认"可自动计算校验和。"单次发送"可 发送数据至 OEM2-SE 接口。

▼观察回答报文: 第0字节是 AB_status (D7)=0;见本手册第四章例2;主站发送的 PROFIBUS 输出数

据,由主站发→SE→PC。

▼对系统 I,观察 WinCC 主站数据,如图 5-59。



图 5-59 单次数据交换,观察 WinCC 主站数据

▼对系统Ⅱ,观察 PLC 主站数据,如图 5-60。

500	Mo	nita	oring a	and Mod	ifying	Variables -	[@¥AT_1	SE_
	<u> </u>	ble	<u>E</u> dit	Insert	PLC	Varia <u>b</u> le <u>V</u> ie	ew <u>O</u> ptions	Win
-6	a	D	≃	RI 🚑	al 🔏) (~ X	9
-	_							
		Ad	dress	Symbol	Disp	Status valu	ie Modify va	alue
1		IB	1		$HE\times$	B#16#01		
2		IВ	2		$HE\times$	B#16#02		
3		IB	з		$HE \times$	B#16#03		
4		IB	4		$HE \times$	B#16#04		
5		IB	5		$HE \times$	B#16#05		1
6		IB	6		$HE \times$	B#16#06		
7		IB	7		$HE\times$	B#16#07		
8		IB	8		$HE \times$	B#16#08		
9		IB	9		$HE \times$	B#16#09		
10		IB	10		$HE \times$	B#16#0A	•	
11		IB	11		$HE \times$	B#16#0B		
12		IB	12		$HE \times$	B#16#00		
13		IB	13		$HE \times$	B#16#0D		
14		IB	14		$HE \times$	B#16#0E		
15		IB	15		$HE \times$	B#16#0F		
16		IB	16		$HE \times$	B#16#10		
17		IB	17		$HE \times$	B#16#11		
18		IB	18		$HE \times$	B#16#12		
19		IB	19		$HE \times$	B#16#13		
20		IB	20		$HE \times$	B#16#14		
21		IB	21		$HE \times$	B#16#15		
22		IB	22		$HE \times$	B#16#16		
23		IB	23		$HE \times$	B#16#17		
24		IB	24		$HE \times$	B#16#18		
25		IB	25		$HE \times$	B#16#19		
26		IB	26		$HE \times$	B#16#1A	•	
27	1	IB	27		$HE \times$	B#16#1B		
28		IB	28		$HE \times$	B#16#1C		1
29		IB	29		$HE \times$	B#16#1D	•	
30		IB	30		$HE \times$	B#16#1E		1
31		IB	31		HEX	B#16#1F		
32		IB	32		HEX	B#16#20		
33		IB	33		HEX	B#16#21		
34		IB	34		HEX	B#16#22		
35		IB	35		HEX	B#16#23		
20		ID	20		THE V			

图 5-60 单次数据交换、观察 PLC 主站数据

■ 交ł	與数据		_	
发送	送数据(PI	ROFIBUS主站接收):	接收数据(PROFIBUS主站输出):	-
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	0 24 38 25 39 26 3A 27 3B 28 3C 29 3D 30 3E 31 3F 32 40 33 41 34 42 35 43 36 44 37 45 38 46 39 47 40 48 41 49 42 4A 43 4B 44 4C 45 4D 46 4E 47	 4F 48 81 50 49 2 51 52 53 54 55 56 56 57 58 59 54 59 50 50 50 51 52 55 56 57 60 61 62 63 64 65 66 	0 0 24 66 48 CC 1 11 25 66 49 JF 2 11 26 77 3 11 27 77 4 11 28 77 5 22 29 77 6 22 30 88 7 22 31 86 8 22 32 88 9 33 33 88 10 33 34 99 11 33 35 99 12 33 36 99 13 33 37 99 14 44 38 AA 15 44 39 AA 16 44 40 AA 17 44 41 AA 18 55 42 B8 19 55 43 B8 20 55 44 B8 21 55 45 <th>5</th>	5
		发送数据确认 单次发送 连续发	发送] 停止 退出	-

ii、连续数据交换:见图 5-61、图 5-62。

图 5-61 连续数据交换、自动形成校验和、PROFIBUS 输入数据 1~48 会自动改变

PROFIBUS数据接收区	PROFIBUS数据发送区
(48 Bytes)	(48 Bytes)
输入字节0: 63 输入字节24: 7B	输出字节0: 11 输出字节24 66
输入字节1: 64 输入字节25: 7C	输出字节1: 11 输出字节25 77
输入字节2: 65 输入字节26: 7D	输出字节2: 11 发送字节26 77
输入字节3: 66 输入字节27: 7E	输出字节3: 11 发送字节27 77
输入字节4: 67 输入字节28: 7F	输出字节4: 22 发送字节28 77
输入字节5: 68 输入字节29: 80	输出字节5: 22 发送字节29 88
输入字节6: 69 输入字节30: 81	输出字节6: 22 发送字节30 88
输入字节7: 6A 输入字节31: 82	输出字节7: 22 发送字节31 88
输入字节8: 6B 输入字节32: 83	输出字节8: 33 发送字节32 88
输入字节9: 6C 输入字节33: 84	输出字节9: 33 发送字节33 99
输入字节10: 6D 输入字节34: 85	输出字节10 33 发送字节34 99
输入字节11: 6E 输入字节35: 86	输出字节11 33 发送字节35 <mark>99</mark>
输入字节12: 6F 输入字节36: 87	输出字节12 33 发送字节36 99
输入字节13: 70 输入字节37: 88	输出字节13 <mark>44</mark> 发送字节37 <mark>64</mark>
输入字节14: 71 输入字节38: 89	输出字节14 44 发送字节38 AA
输入字节15: 72 输入字节39: 8A	输出字节15 44 发送字节39 AA
输入字节16: 73 输入字节40: 8B	输出字节16 <mark>44</mark> 发送字节40 <mark>44</mark>
输入字节17: 74 输入字节41: 8C	输出字节17 <mark>55</mark> 发送字节41 BB
输入字节18: 75 输入字节42: 8D	输出字节18 <mark>55</mark> 发送字节42 BB
输入字节19: 76 输入字节43: 8E	输出字节19 <mark>55</mark> 发送字节43 BB
输入字节20: 77 输入字节44: 8F	输出字节20 55 发送字节44 BB
输入字节21: 78 输入字节45: 90	输出字节21 66 发送字节45 cc
输入字节22: 79 输入字节46: 91	输出字节22 66 发送字节48 cc
输入字节23: 7A 输入字节47: 66	输出字节23 66 发送字节47 cc
—	
	赵四 —
主站接收的 PROFIBUS 输入数据,由 PC→SE→WinCC,见上图 5-61	用户可自行键入PROFIBUS输出数据,由WinCC→SE→PC,见上图5-61

图 5-62 连续数据交换、观察 WinCC 主站数据

附件 A: 下载实验板程序操作指南及下载线制作方法

本实验板支持对具有 ISP (IN-SYSTEM PROGRAMMING)功能的 CPU,进行程序在线式写入(如 PHILIPS 的 P89V51RD2)。

本指南以 P89V51RD2 为例,但不同厂家的操作方法各有异同,请参见其产品手册中的详细介绍, 勿完全效仿。(不具有 ISP 功能的 CPU 芯片请勿使用本功能,以免芯片或电路板的烧毁)。若有硬件改 动,请根据提供的原理图自行飞线处理。

1. 实验板的下载模式选择

实验板 MODE **开关 SW2 设置:** MODE=00000111_(Bin);实验板上电。由于实验板将 CPU 置于编程模式,因此上电后 CPU 不执行程序,也就没有输出指示灯流水演示。

2. Flash Magic 软件的程序下载操作

Flash Magic 是专为 PHILIPS 公司 P89X51XXX 系列产品所提供的 ISP 下载软件(见产品光盘)。将 Flash Magic 装入 PC 运行见下图附 1。

(1) 为 ISP 选定 PC 机侧所用的串口 COM1-COM4;指定所用微控制器的型号 P89V51RD2。可选择 Baud Rate=19200 作为尝试波特率。

🛱 Flash Magic	
Eile ISP Options Help	
2	
COM Port: COM 2 Erase block 0 (0x0000-0xFFFF)	
Baud Rate: 19200	
Device: 89V51RD2	
Erase all Flash	
-3	
Hex File: F:\产品手册\PB-OEM2-S\SE\SE新文件\例程及GSD\例2\实验 Browse	e
Modified:星期四,十二月 9, 2004, 9:07:18 more info	
.4	
T Marihu alkar aragrammina 🗖 Cat Casuritu Di 1	
✓ Veniy and programming j Set Security Bit 1	
Generate checksums 🔲 Set Security Bit 3	
Execute Start	
Microcontrollers from Philips Semiconductors Main web page at:	
www.philipsmcu.com	

附图 1

(2) 进入 ISP→Read Security Bits, 尝试连接 P89V51RD2。可能会出现警告:"Reset the device into ISP mode now";可将目标板从新上电,可能会再次弹出要求减低波特率的警告,这时可减低一档波特率,回到 Read Security Bits 从新尝试,直到出现"Security bit 1 is unset"表示与 P89V51RD2 ISP 连通。

Flash Magic	
File ISP Options Help	①进入 ISP→Read Security Bits.
_1 Blank Check	尝试连接 P89V51RD2
Read Security Bits	Excert March 0 (0-0000 0-EEEE)
Read Device Signature	
Boot Ve Reset Device	②可能会出现警告,可
Display -	· · · · · · · · · · · · · · · · · · ·
Erace F Reset the device into ISP mode r	
Verify Cancel	③可能会再次弹出要求减低波特率的警告,
Executi	[] 这时可减低一档波特率,回到①从新尝试。
Unable to connect at the specified by hardware into ISP mode again.	aud rate. Try reducing the baud rate. Reset the
[Security Bits
enu	Cogurity hit 1 is uppet
✓ Fill unused Flash	2 Security bit i is unset
🛛 🔽 Generate checksums 🛛 🗖 Set Security Bit	3 [
Execute G clks/cycle	
Visit the "Flash Magic" home page for info on the la	itest revision
	Close
www.esacademy ④表示与 P89V51RD2 ISP 连	·通。 1 1
附图 2	

(3) 选择功能选项并下载程序

Filash Magic File ISP Options Help	× ① 擦除
COM Port: COM 2 Baud Rate: 19200 Device: 89V51RD2	■ Erase block (x0000-0xFFFF) ■ ② 选择下载程序 ■ Erase all Flash ■ Erase blocks used K File
3 Hex File: M2-S\SE\SE新文件\例程及GSD\ Modified: 星期四, 十二月 9, 2004	例2\实验板目标程序\B2.HEX Browse ③ 选择校验等。一定 ① 中时 天林
✓ Verify after programming Set Securit ✓ Fill unused Flash Set Security ✓ Generate checksums Set Security ✓ Execute ✓ 6 clks/cycle	要选择 6 clks/cycle.
Download free 8051 and XA code examples usi www.esacademy.com/faq/progs	ng I2C, CAN, Flash, etc.

3. 恢复实验板运行状态

 (1) 恢复**实验板** MODE **开关 SW2 设置: MODE**=11000000_(Bin)(调试模式 1)或 **MODE**=11000010_(Bin)(调 试模式 2);

(2) 实验板上电。上电后 CPU 将执行程序,有输出指示灯流水演示。

(3) 按调试模式1或调试模式2调试下载的实验板程序。

4. 下载线的制作及连接方法

本系统包括一根 RS232 电缆,可用做下载线使用。如果需要,用户可自制 RS-232 电缆;方法如下: (1) 材料: D 型九孔插头(female)两个,3 芯电缆线。

(2) 将每个插头按 2-2, 3-3, 5-5 的规则用电缆线焊在一起,其中任一端可接计算机的 COM 口,另一端 接连本实验板的右侧,九针 D 型插座。

!!!注意:在进行实验板模式切换时,请务必下电,以免造成不必要的损失。

!!!注意: P89V51RD2 芯片的程序写入也可由其他的编程器来完成,但对于特殊寄存器的操作,各个编 程器有所不同,使用时要尤为注意。具体的芯片擦除、加密及 IAP 等操作在此不作赘述,详情参见 P89V51RD2 手册。

北京鼎实创新科技有网	
现场总线 PROFIBUS(中国)	技术资格中心
电话: 010-82078264、010-62054940	传真: 010-82078264
地址:北京德胜门外教场口1号,一号楼 401~404	邮编: 100011
Web: <u>www.c-profibus.com.cn</u>	Email: <u>tangjy@c-profibus.com.cn</u>
<u> </u>	