



PROFIBUS DP 从站解决方案

PROFIBUS DP 从站 PCI 卡

PB-OEM4-PCI 使用手册

V1.1



北京鼎实创新科技有限公司

2011-1

目 录

第一章 产品概述	2
(一) 主要用途	2
(二) 产品系列	2
(三) 产品特点	3
(四) 技术指标	3
第二章 PB-OEM4-PCI 卡安装与编程	4
(一) PB-OEM4-PCI 的安装	4
(1) 系统要求	4
(2) 安装硬件	4
(3) DLL 函数说明	6
(4) DLL 的使用	10
第三章 PB-OEM4-PCI 与 PROFIBUS 主站的连接与配置	17
(一) PROFIBUS 网络连接	17
(二) 配置要求	17
(三) 使用 Step7 进行系统配置	17
(1) 打开 Step7	17
(2) 新建一个项目	17
(3) 添加 SIMATIC 300 Station	18
(4) 添加 PROFIBUS DP 从站 PB-OEM4-PCI	23
(四) 下载程序	25
(1) Set PG-PC Interface 的设置	25
(2) 下载程序到 SIMATIC 315-2 DP 中	26
(3) 接收/发送数据的监控	27
第四章 关于 GSD 文件	29
(一) GSD 文件相关说明	29
(二) PB-OEM4-PCI 的 GSD 文件和 ID 号	29
(三) 用户产品的 ID 号、GSD 文件及产品认证	31

第一章 产品概述

(一) 主要用途

本产品是具有双口RAM、PCI 接口的PROFIBUS-DP 从站网卡，为PC 机、工业板式电脑、及各种 HMI 连接到PROFIBUS 提供解决方案。

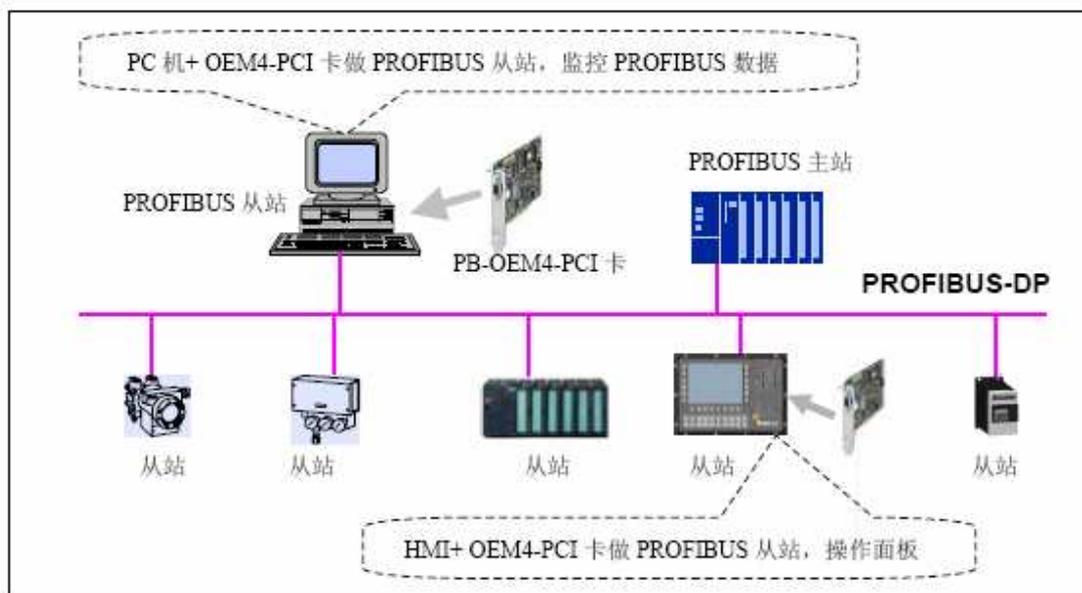
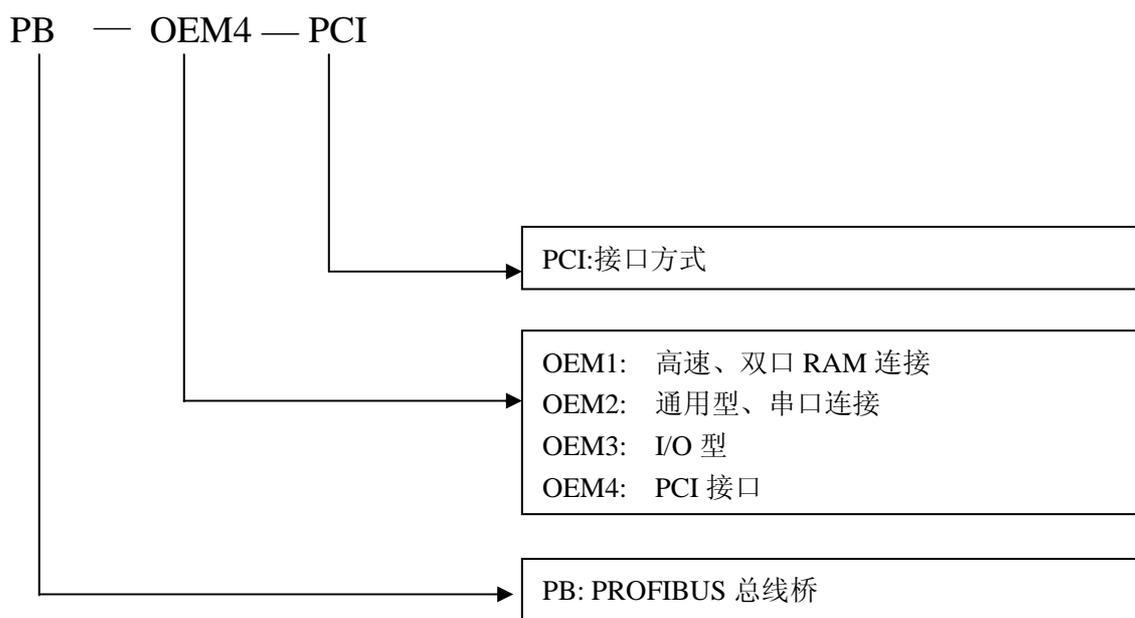


图 1-1 PC 机、HMI 配置 OEM4-PCI 卡做从站，监控 PROFIBUS 数据

(二) 产品系列

PROFIBUS—PCI 总线桥 PB-OEM4-PCI 接口卡（以下简称“PCI 卡”）是 PROFIBUS 总线桥 OEM 系列的产品。



(三) 产品特点

- 1、应用简单：用户安装好 PB-OEM4-PCI 的驱动程序后，调用所提供的 DLL，可以很方便的控制 PROFIBUS 的输入/输出。
- 2、应用广泛：可广泛用于带有 PCI 插槽的各种机型，如：工控机，HMI 等。
- 3、以 OEM 方式提供：用户对整机产品有自主知识产权、品牌、商标注册权。

(四) 技术指标

- 1、PROFIBUS-DP 接口：D9，PROFIBUS-DP 从站，DP-V0/V1 协议，符合 IEC61508 及中国国家标准 GB/T 20540-2006：测量和控制数字数据通信工业控制系统用现场总线的第 3 部分：PROFIBUS 规范；
- 2、标准 PROFIBUS-DP 驱动接口，波特率自适应，最大波特率 12M；
- 3、PROFIBUS 输入/输出数量可自由设定，最大 200 字节输入/200 字节输出；
- 4、标准 PCI 接口；
- 5、环境温度：
 运输和存储：-40℃~+70℃
 工作温度：-20℃~+55℃
- 6、工作相对湿度：5~95%

第二章 PB-OEM4-PCI 卡安装与编程

(一) PB-OEM4-PCI 的安装

(1) 系统要求

操作系统：Windows 2000+Service Pack 4 或者 Windows XP

(2) 安装硬件

- ① 在 PC 或者 HMI 设备关闭电源后，将 PB-OEM4-PCI 卡插入 PCI 插槽，并用螺丝将其固定。
- ② 安装 PB-OEM4-PCI 驱动程序。安装好 PB-OEM4-PCI 后，开机后，系统提示找到新硬件，按照图 2-1—2-4 安装驱动程序。



图 2-1



图 2-2



图 2-3



图 2-4

安装成功后，则会出现图 2-5 所示界面。

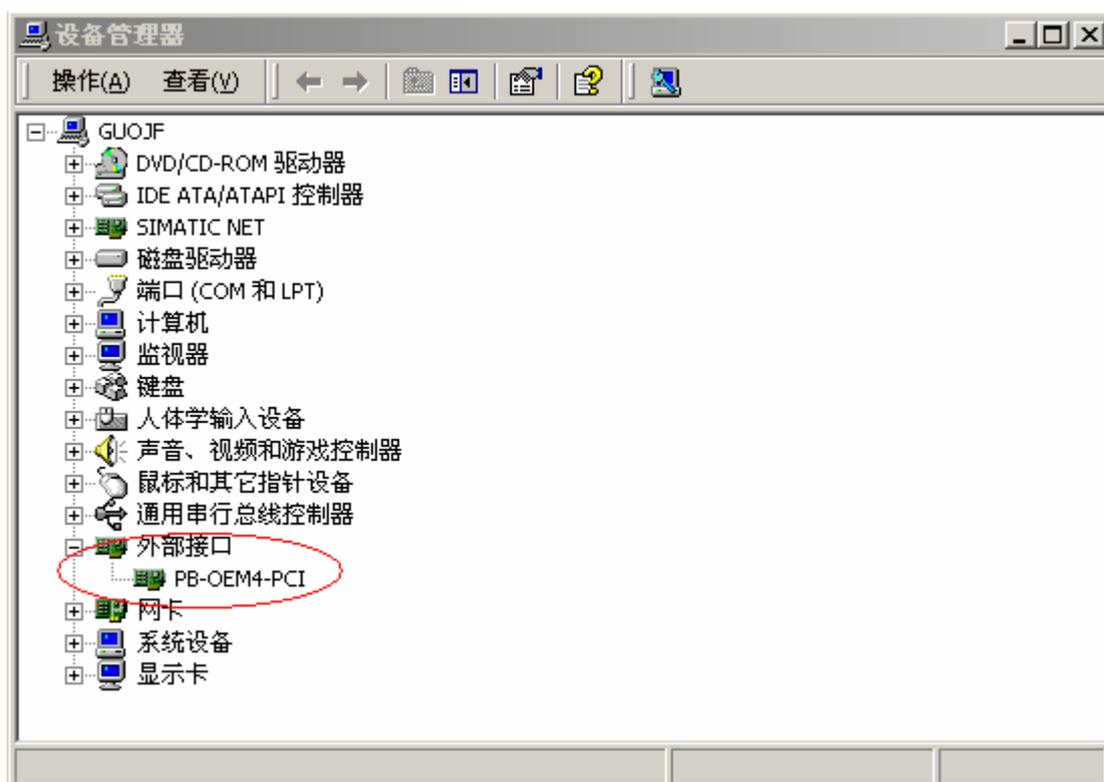


图 2-5

(3) DLL 函数说明

① PCI 卡启动函数

```
HANDLE Open_PCICard(ULONG iIndex)
```

功能：打开 PCI 卡,返回句柄,出错则无效 (INVALID_HANDLE_VALUE 值);

参数:

类型	参数	描述
ULONG	iIndex	设备序号,从 0 开始依次增加

在一台 PC 机上,可以安装多台 PB-OEM4-PCI 卡,参数 iIndex 表示 PCI 卡安装的次序。

返回值:

值	描述
-1	打开设备失败
HANDLE	句柄

② PCI 卡初始化函数

```
UCHAR Init_PCICard(ULONG iIndex,UCHAR *init_addr,
                   UCHAR init_len)
```

参数:

类型	参数	描述
ULONG	iIndex	设备序号,从 0 开始依次增加
UCHAR*	init_addr	指向初始化数据数组首地址的指针
UCHAR	init_len	初始化数据的长度

PCI 卡初始化包括对双口 RAM 自检和对 PROFIBUS 参数初始化两部分, PCI 卡首先对双口 RAM 自检,如成功,继续对 PROFIBUS 参数初始化,否则退出程序,并且返回不同的结果值。

返回值:

值	描述
0x10	自检和初始化成功
0x20	自检失败
0x21	初始化失败
0x30	自检未获得控制权(循环 1000 次)
0x31	初始化时未获得控制权(循环 1000 次)
0x40	接口板未准备好

③ 接收 PROFIBUS 数据和用户参数

```
UCHAR Recv_PBdata(ULONG iIndex,UCHAR *revda_addr,
                  UCHAR revda_len, UCHAR *revpda_addr,
                  UCHAR revpda_len)
```

参数:

类型	参数	描述
ULONG	iIndex	设备序号,从 0 开始依次增加
UCHAR*	revda_addr	指向存放待接收数据的首地址
UCHAR	revda_len	待接收数据的长度
UCHAR	revpda_addr	指向存放待接收用户参数的首地址
UCHAR	revpda_len	待接收用户参数的长度

返回值:

值	描述
0x10	读取数据成功
0x11	读取数据失败
0x20	读取用户参数成功
0x21	读取用户参数失败
0x30	未获得控制权
0x40	输出数据和用户参数数据同时有效，但读取失败
0x44	输出数据和用户参数数据同时有效，读取用户参数成功
0x48	输出数据和用户参数数据同时有效，读取数据成功
0x4C	输出数据和用户参数数据同时有效，读取用户参数和数据成功
0x50	PROFIBUS 主站已连通，没有有效数据
0x60	PROFIBUS 主站未连通

④ 发送 PROFIBUS 数据

```
UCHAR Send_PBdata(ULONG iIndex,UCHAR *sendda_addr,
                  UCHAR sendda_len)
```

参数:

类型	参数	描述
ULONG	iIndex	设备序号,从 0 开始依次增加
UCHAR *	sendda_addr	指向存放待发送数据的首地址
UCHAR	sendda_len	待发送数据的长度,它包括数据长度、数据、校验和

返回值:

值	描述
0x10	发送数据成功
0x20	发送数据失败
0x30	未获得控制权

⑤ 退出系统，关闭 PCI 卡操作

```
void Exit_PCICard(ULONG iIndex)
```

当程序要退出系统时，要使用此函数。它包括：系统复位、关闭中断使能、硬件同步标志清零、释放控制权和关闭 PCI 卡等操作。

⑥ 关闭中断使能宏定义

```
BOOL Close_Int(ULONG iIndex)
```

在不使用中断方式时，要使用此命令关闭中断使能。

参数：

类型	参数	描述
ULONG	iIndex	设备序号,从 0 开始依次增加

返回值：

值	描述
TRUE	关闭中断成功
FALSE	关闭中断失败

⑦ 中断回调函数

```
BOOL Open_Int(ULONG iIndex,OEM4_INT_INDEX IntEvent)
```

功 能：打开中断使能，中断时响应中断函数 IntEvent。

参数：

类型	参数	描述
ULONG	iIndex	设备序号,从 0 开始依次增加
OEM4_INT_INDEX	IntEvent	接收中断服务函数名

返回值：

值	描述
TRUE	打开中断成功
FALSE	打开中断失败

(4) DLL 的使用

基于 Visual C++ 6.0 开发平台。

新建一个基于 MFC 的项目，添加相应的头文件和库文件；将所提供的头文件和库文件复制到当前项目文件下。文件包括：

oem4pcidll.dll

oem4pcidll.h

oem4pcidll.lib

① 添加 H 头文件

在“File View”的“Header Files”中添加“oem4pcidll.h”；

在 C 源程序中添加“#include "oem4pcidll.h"”。

② 添加库文件

在 Project Settings 中“Link”组中的“Object/library modules:”处添加“oem4pcidll.lib”。

(4) 两种数据接收方式

① 查询方式，参考例程 oem4pci_demo；程序流程如图 2-6 所示。

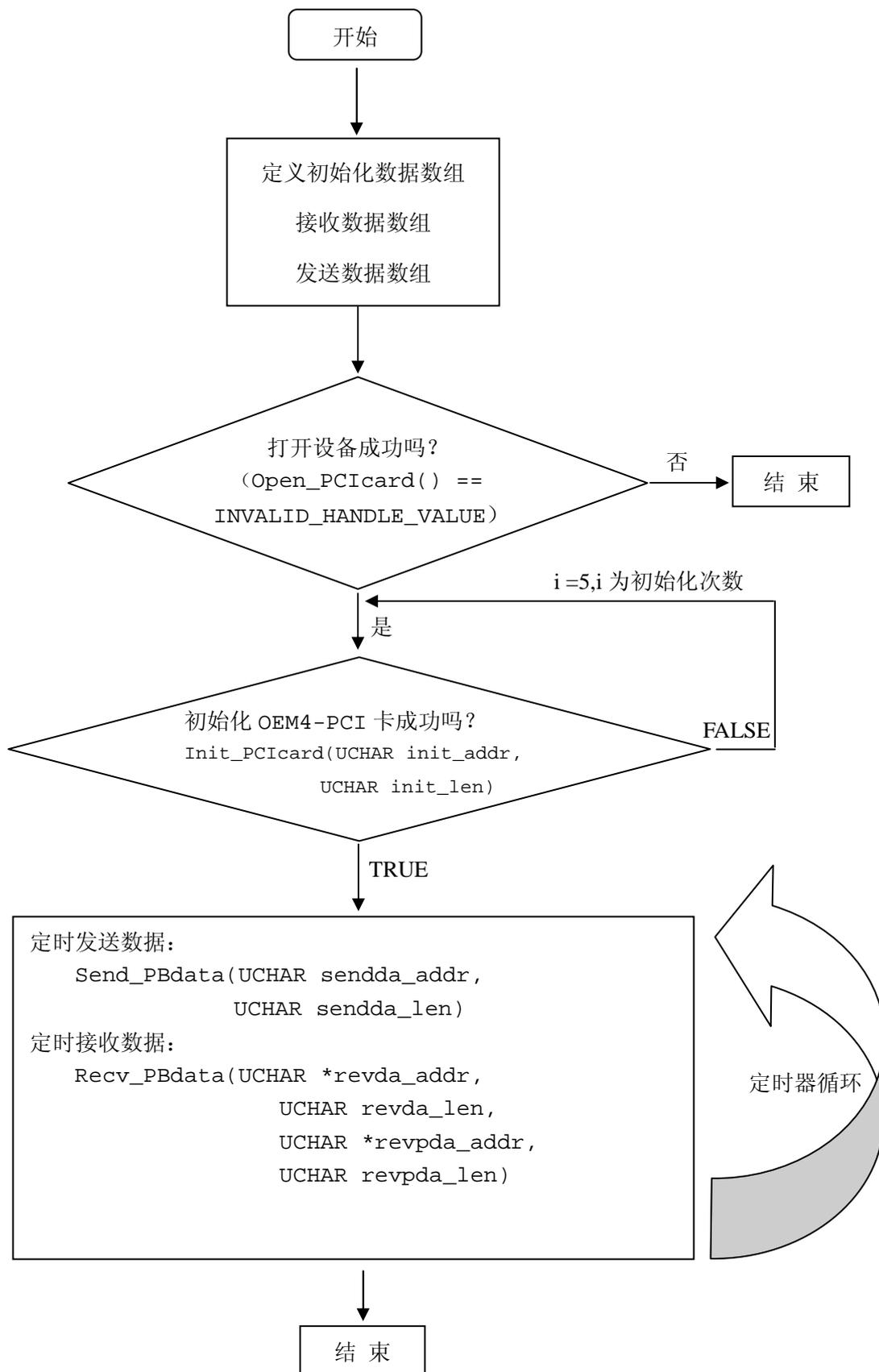


图 2-6 查询方式接收数据

②中断方式，参考例程 oem4pci_INTdemo；程序流程如图 7 所示。

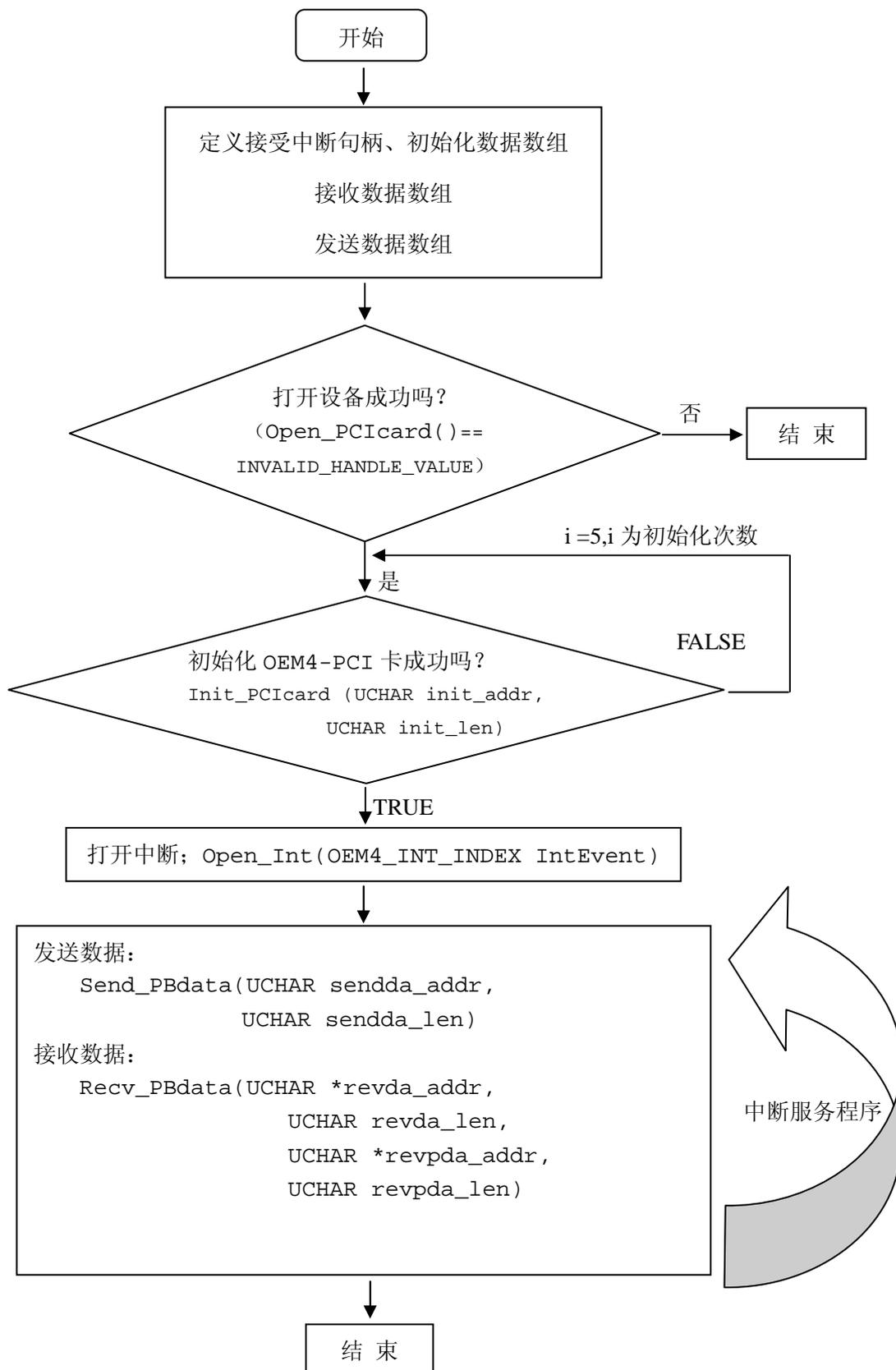


图 2-7 中断方式接收数据

③ 关于初始化数据格式的说明

表 2-1 初始化数据格式

说明:

数据	定义	字节数
UBYTE pb_adr;	PROFIBUS 从站地址	1
UBYTE pb_id_h;	PROFIBUS 从站 ID 高字节	1
UBYTE pb_id_l;	PROFIBUS 从站 ID 低字节	1
UBYTE cfg_len;	Config 数据长度	1
UBYTE cfg_data[10];	Config 数据	10
UBYTE pb_in_len;	PROFIBUS 输入数据长度	1
UBYTE pb_out_len;	PROFIBUS 输出数据长度	1
UBYTE user_prm_len;	用户参数长度	1

▼站号

用户模板应设有设置站号的拨码开关或有键盘、手持终端等其他设置 PROFIBUS 从站站号和方式；从站地址范围为 0~126。用户模板设置的站号必须和你在主站配置时设置的站号一致。

▼ID 号

经过 PI 认证的产品都有唯一的一个 ID 号。未经测试认证的设备可先自定义一个 ID 号，比如：0080H。产品 ID 号必需和你的 GSD 文件中的 ID 号一致；

▼ I/O 配置数据

I/O 配置数据：在 PROFIBUS 通讯中，用特定的 16 进制数据定义 PROFIBUS 输入和输出数据。

输入数据：本产品 PROFIBUS 的 I/O 数量最大可为 10；I/O 配置数据请见下表 2 中的“代码”；在 PROFIBUS 通讯中，为确保 PROFIBUS 输入/输出数据的完整性，将数据分成如下类型：

Byte 完整： 用于开关量；

Word 完整： 用于模拟量；

全部输入/输出完整： 用于浮点数；

表 2-2 PB-OME2-SE 接口 PROFIBUS I/O 配置数据表

代码	说明	代码	说明
	byte input, Byte 完整		byte input, 全部输入/输出完整
0x10	1 byte input, Byte 完整	0x90	1 byte input, 全部输入/输出完整
0x11	2 byte input, Byte 完整	0x91	2 byte input, 全部输入/输出完整
0x12	3 byte input, Byte 完整	0x92	3 byte input, 全部输入/输出完整
0x13	4 byte input, Byte 完整	0x93	4 byte input, 全部输入/输出完整
0x14	5 byte input, Byte 完整	0x94	5 byte input, 全部输入/输出完整
0x15	6 byte input, Byte 完整	0x95	6 byte input, 全部输入/输出完整
0x16	7 byte input, Byte 完整	0x96	7 byte input, 全部输入/输出完整
0x17	8 byte input, Byte 完整	0x97	8 byte input, 全部输入/输出完整

0x18	9 byte input, Byte 完整	0x98	9 byte input, 全部输入/输出完整
0x19	10 byte input, Byte 完整	0x99	10 byte input, 全部输入/输出完整
0x1A	11 byte input, Byte 完整	0x9A	11 byte input, 全部输入/输出完整
0x1B	12 byte input, Byte 完整	0x9B	12 byte input, 全部输入/输出完整
0x1C	13 byte input, Byte 完整	0x9C	13 byte input, 全部输入/输出完整
0x1D	14 byte input, Byte 完整	0x9D	14 byte input, 全部输入/输出完整
0x1E	15 byte input, Byte 完整	0x9E	15 byte input, 全部输入/输出完整
0x1F	16 byte input, Byte 完整	0x9F	16 byte input, 全部输入/输出完整
	byte output, Byte 完整		byte output, 全部输入/输出完整
0x20	1 byte output, Byte 完整	0xA0	1 byte output, 全部输入/输出完整
0x21	2 byte output, Byte 完整	0xA1	2 byte output, 全部输入/输出完整
0x22	3 byte output, Byte 完整	0xA2	3 byte output, 全部输入/输出完整
0x23	4 byte output, Byte 完整	0xA3	4 byte output, 全部输入/输出完整
0x24	5 byte output, Byte 完整	0xA4	5 byte output, 全部输入/输出完整
0x25	6 byte output, Byte 完整	0xA5	6 byte output, 全部输入/输出完整
0x26	7 byte output, Byte 完整	0xA6	7 byte output, 全部输入/输出完整
0x27	8 byte output, Byte 完整	0xA7	8 byte output, 全部输入/输出完整
0x28	9 byte output, Byte 完整	0xA8	9 byte output, 全部输入/输出完整
0x29	10 byte output, Byte 完整	0xA9	10 byte output, 全部输入/输出完整
0x2A	11 byte output, Byte 完整	0xAA	11 byte output, 全部输入/输出完整
0x2B	12byte output, Byte 完整	0xAB	12byte output, 全部输入/输出完整
0x2C	13 byte output, Byte 完整	0xAC	13 byte output, 全部输入/输出完整
0x2D	14 byte output, Byte 完整	0xAD	14 byte output, 全部输入/输出完整
0x2E	15 byte output, Byte 完整	0xAE	15 byte output, 全部输入/输出完整
0x2F	16 byte output, Byte 完整	0xAF	16 byte output, 全部输入/输出完整
	byte input/output, Byte 完整		byte input/output, 全部输入/输出完整
0x30	1 byte input/output, Byte 完整	0xB0	1 byte input/output, 全部输入/输出完整
0x31	2 byte input/output, Byte 完整	0xB1	2 byte input/output, 全部输入/输出完整
0x32	3 byte input/output, Byte 完整	0xB2	3 byte input/output, 全部输入/输出完整
0x33	4 byte input/output, Byte 完整	0xB3	4 byte input/output, 全部输入/输出完整
0x34	5 byte input/output, Byte 完整	0xB4	5 byte input/output, 全部输入/输出完整
0x35	6 byte input/output, Byte 完整	0xB5	6 byte input/output, 全部输入/输出完整
0x36	7 byte input/output, Byte 完整	0xB6	7 byte input/output, 全部输入/输出完整
0x37	8 byte input/output, Byte 完整	0xB7	8 byte input/output, 全部输入/输出完整
0x38	9 byte input/output, Byte 完整	0xB8	9 byte input/output, 全部输入/输出完整
0x39	10 byte input/output, Byte 完整	0xB9	10 byte input/output, 全部输入/输出完整
0x3A	11 byte input/output, Byte 完整	0xBA	11 byte input/output, 全部输入/输出完整
0x3B	12byte input/output, Byte 完整	0xBB	12byte input/output, 全部输入/输出完整
0x3C	13 byte input/output, Byte 完整	0xBC	13 byte input/output, 全部输入/输出完整
0x3D	14 byte input/output, Byte 完整	0xBD	14 byte input/output, 全部输入/输出完整
0x3E	15 byte input/output, Byte 完整	0xBE	15 byte input/output, 全部输入/输出完整
0x3F	16 byte input/output, Byte 完整	0xBF	16 byte input/output, 全部输入/输出完整
	word input, word 完整		word input, 全部输入/输出完整
0x50	1 word input, word 完整	0xD0	1 word input, 全部输入/输出完整
0x51	2 word input, word 完整	0xD1	2 word input, 全部输入/输出完整
0x52	3 word input, word 完整	0xD2	3 word input, 全部输入/输出完整
0x53	4 word input, word 完整	0xD3	4 word input, 全部输入/输出完整
0x54	5 word input, word 完整	0xD4	5 word input, 全部输入/输出完整
0x55	6 word input, word 完整	0xD5	6 word input, 全部输入/输出完整
0x56	7 word input, word 完整	0xD6	7 word input, 全部输入/输出完整
0x57	8 word input, word 完整	0xD7	8 word input, 全部输入/输出完整
0x58	9 word input, word 完整	0xD8	9 word input, 全部输入/输出完整
0x59	10 word input, word 完整	0xD9	10 word input, 全部输入/输出完整
0x5A	11 word input, word 完整	0xDA	11 word input, 全部输入/输出完整
0x5B	12 word input, word 完整	0xDB	12 word input, 全部输入/输出完整

0x5C	13 word input, word 完整	0xDC	13 word input, 全部输入/输出完整
0x5D	14 word input, word 完整	0xDD	14 word input, 全部输入/输出完整
0x5E	15 word input, word 完整	0xDE	15 word input, 全部输入/输出完整
0x5F	16 word input, word 完整	0xDF	16 word input, 全部输入/输出完整
	word output, word 完整		word output, 全部输入/输出完整
0x60	1 word output, word 完整	0xE0	1 word output, 全部输入/输出完整
0x61	2 word output, word 完整	0xE1	2 word output, 全部输入/输出完整
0x62	3 word output, word 完整	0xE2	3 word output, 全部输入/输出完整
0x63	4 word output, word 完整	0xE3	4 word output, 全部输入/输出完整
0x64	5 word output, word 完整	0xE4	5 word output, 全部输入/输出完整
0x65	6 word output, word 完整	0xE5	6 word output, 全部输入/输出完整
0x66	7 word output, word 完整	0xE6	7 word output, 全部输入/输出完整
0x67	8 word output, word 完整	0xE7	8 word output, 全部输入/输出完整
0x68	9 word output, word 完整	0xE8	9 word output, 全部输入/输出完整
0x69	10 word output, word 完整	0xE9	10 word output, 全部输入/输出完整
0x6A	11 word output, word 完整	0xEA	11 word output, 全部输入/输出完整
0x6B	12 word output, word 完整	0xEB	12 word output, 全部输入/输出完整
0x6C	13 word output, word 完整	0xEC	13 word output, 全部输入/输出完整
0x6D	14 word output, word 完整	0xED	14 word output, 全部输入/输出完整
0x6E	15 word output, word 完整	0xEE	15 word output, 全部输入/输出完整
0x6F	16 word output, word 完整	0xEF	16 word output, 全部输入/输出完整
	word input/output, Word 完整		word input/output, 全部输入/输出完整
0x70	1 word input/output, Word 完整	0xF0	1 word input/output, 全部输入/输出完整
0x71	2 word input/output, Word 完整	0xF1	2 word input/output, 全部输入/输出完整
0x72	3 word input/output, Word 完整	0xF2	3 word input/output, 全部输入/输出完整
0x73	4 word input/output, Word 完整	0xF3	4 word input/output, 全部输入/输出完整
0x74	5 word input/output, Word 完整	0xF4	5 word input/output, 全部输入/输出完整
0x75	6 word input/output, Word 完整	0xF5	6 word input/output, 全部输入/输出完整
0x76	7 word input/output, Word 完整	0xF6	7 word input/output, 全部输入/输出完整
0x77	8 word input/output, Word 完整	0xF7	8 word input/output, 全部输入/输出完整
0x78	9 word input/output, Word 完整	0xF8	9 word input/output, 全部输入/输出完整
0x79	10 word input/output, Word 完整	0xF9	10 word input/output, 全部输入/输出完整
0x7A	11 word input/output, Word 完整	0xFA	11 word input/output, 全部输入/输出完整
0x7B	12 word input/output, Word 完整	0xFB	12 word input/output, 全部输入/输出完整
0x7C	13 word input/output, Word 完整	0xFC	13 word input/output, 全部输入/输出完整
0x7D	14 word input/output, Word 完整	0xFD	14 word input/output, 全部输入/输出完整
0x7E	15 word input/output, Word 完整	0xFE	15 word input/output, 全部输入/输出完整
0x7F	16 word input/output, Word 完整	0xFF	16 word input/output, 全部输入/输出完整

④ 使用自定义消息实现中断

▼ 定义用户消息：

```
#define WM_INT WM_USER+100
```

▼ 定义中断函数原型为回调函数：

```
void CALLBACK InterruptEvent(void);
```

▼ 中断服务程序：

接收数据；

发送数据（可以放在中断服务程序之外，采用定时器发送）；

发送数据完毕之后，调用 void trig_int(void)函数，触发 OEM4-PCI 卡的 CPU 中

断方式通知 CPU 接收数据。

注意： 由于中断实时性高,可以被随时调用,所以在这个函数里不宜做过多操作,用 `postmessage` 函数将操作外移。

第三章 PB-OEM4-PCI 与 PROFIBUS 主站的连接与配置

PB-OEM4-PCI 作为一个标准的 PROFIBUS DP 从站，可以与西门子、ABB、三菱和欧姆龙的 PROFIBUS 主站进行通讯。本手册以与西门子 PROFIBUS DP 主站 S7 315-2DP 的连接和配置为例。

(一) PROFIBUS 网络连接

硬件连接如图 3-1 所示。

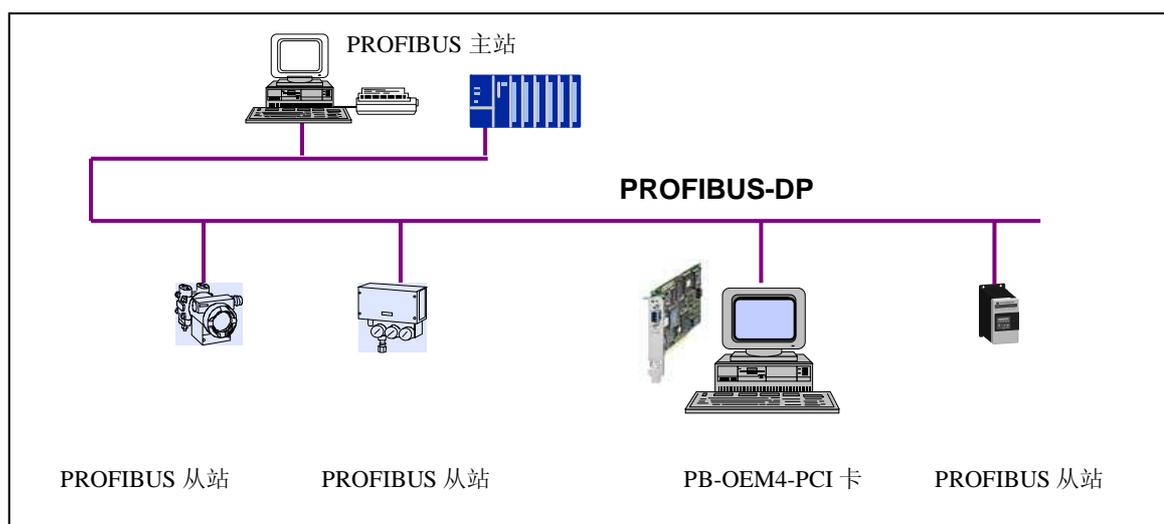


图 3-1

(二) 配置要求

软件：安装 Step7 v5.2 或以上版本。

硬件：安装 CP5611 网卡或者 MPI PC Adapter。(用于下载程序)。

(三) 使用 Step7 进行系统配置

(1) 打开 Step7



SIMATIC Manager.exe

(2) 新建一个项目

见图 3-1:

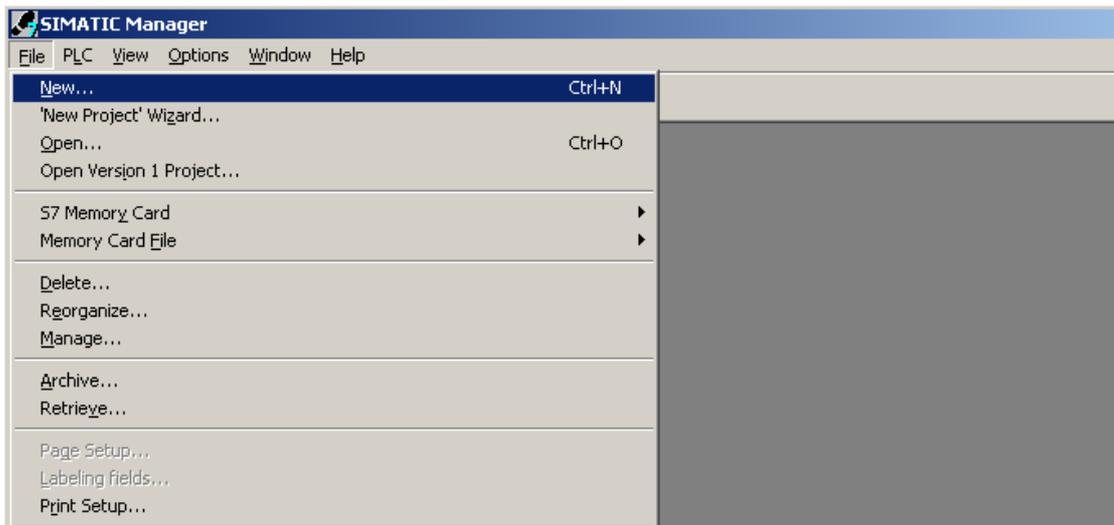


图 3-1

并将该项目命名为：PB-OEM4-PCI，如图 3-2

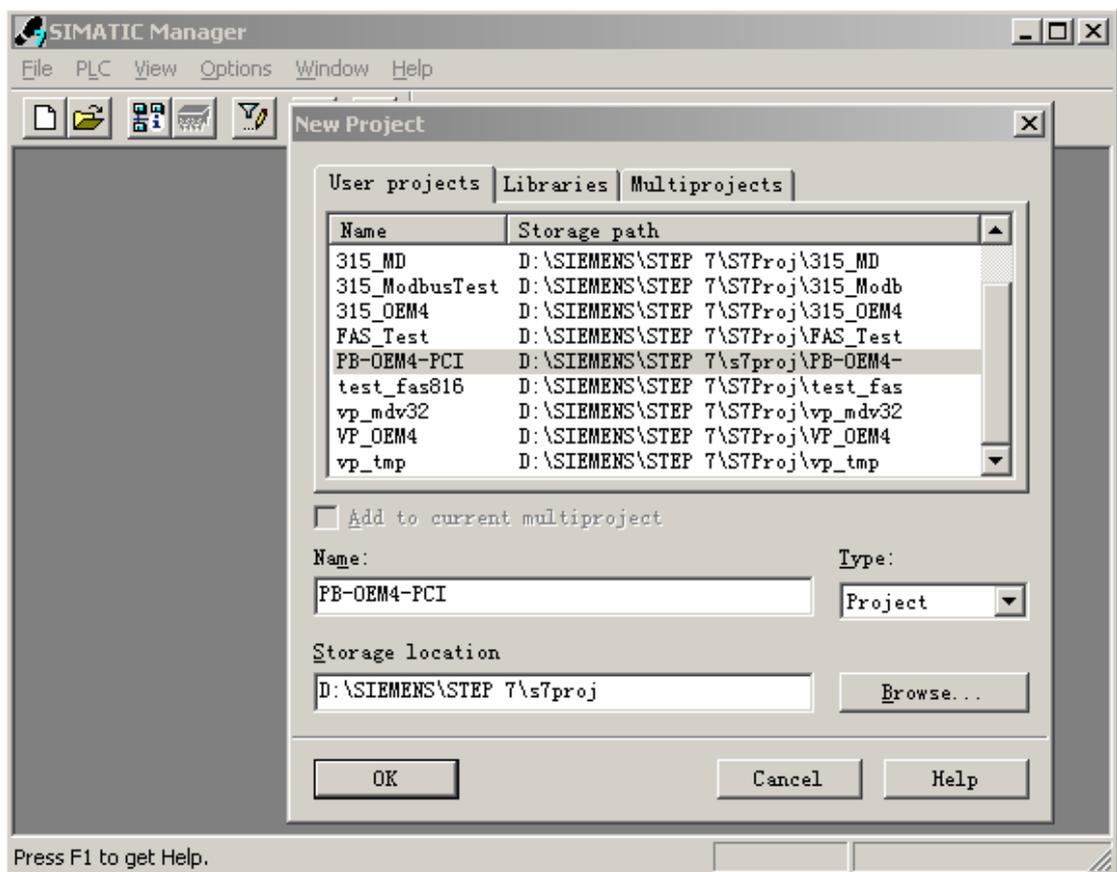


图 3-2

(3) 添加 SIMATIC 300 Station

在 PB-OEM4-PCI 鼠标右键弹出菜单，如图 3-3

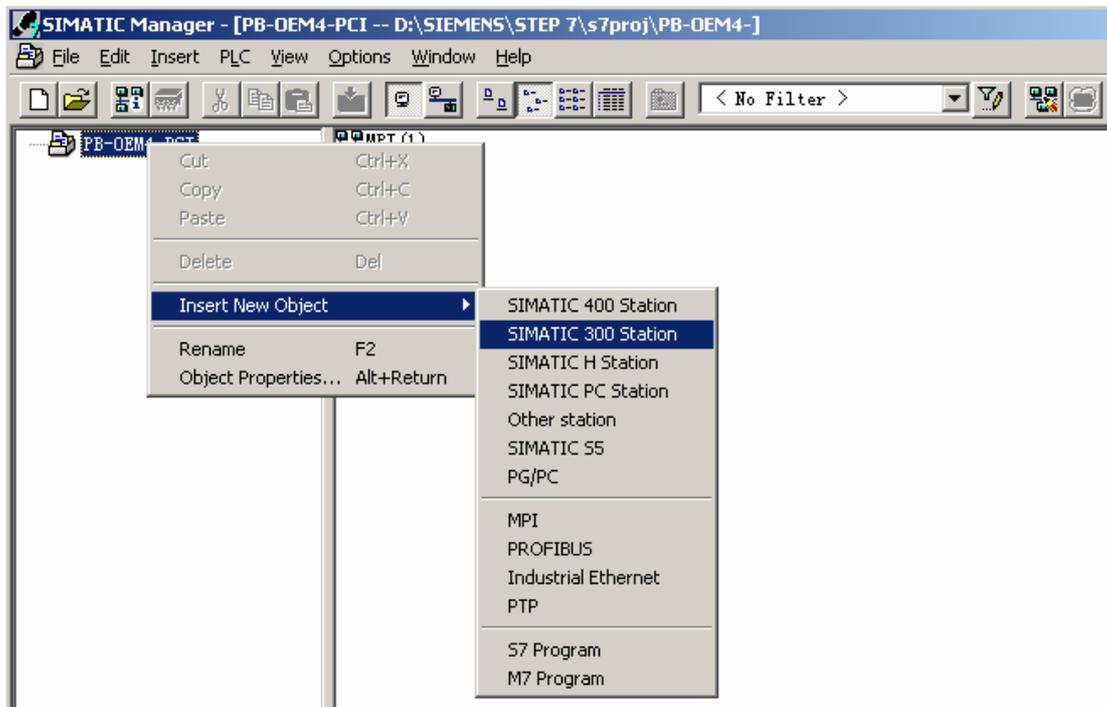


图 3-3

双击图 3-4 中的 SIMATIC PC 300(1), SIMATIC 300(1)为所配置的 Station Name。

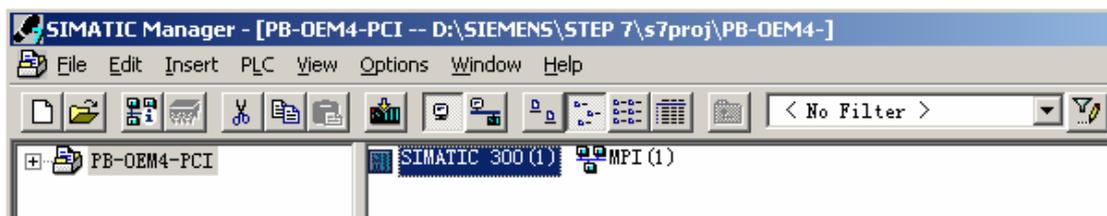


图 3-4

然后双击图 3-5 中的 Hardware,

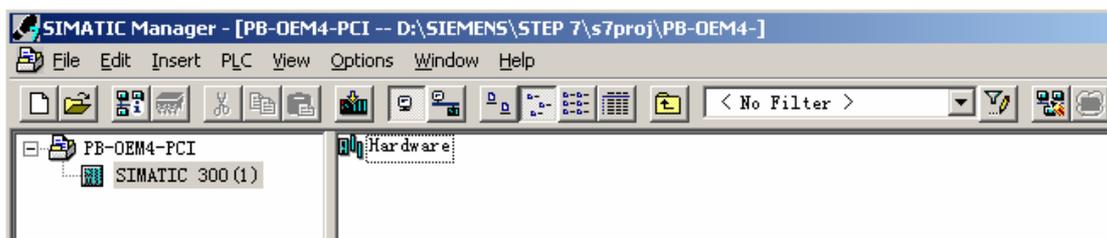


图 3-5

出现图 3-6 窗口, 在该窗口上进行硬件配置。选择 SIMATIC 300→RACK-300→Rail 并双击 Rail, 出现图 3-7 所示界面。

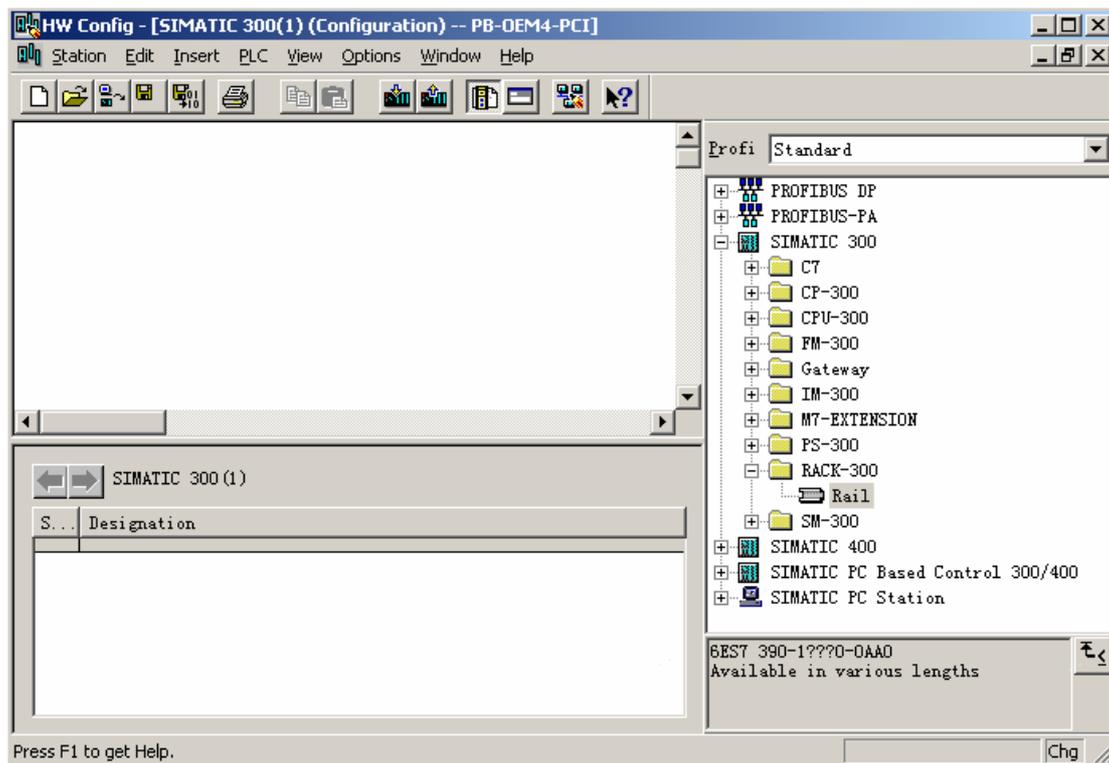


图 3-6

按照图 3-7 所示，选择 rail 的第二槽，在图 3-7 窗口右侧的 catalog 选择 CPU-300→CPU 315-2 DP(选择对应的 CPU 型号)，并鼠标双击该型号。

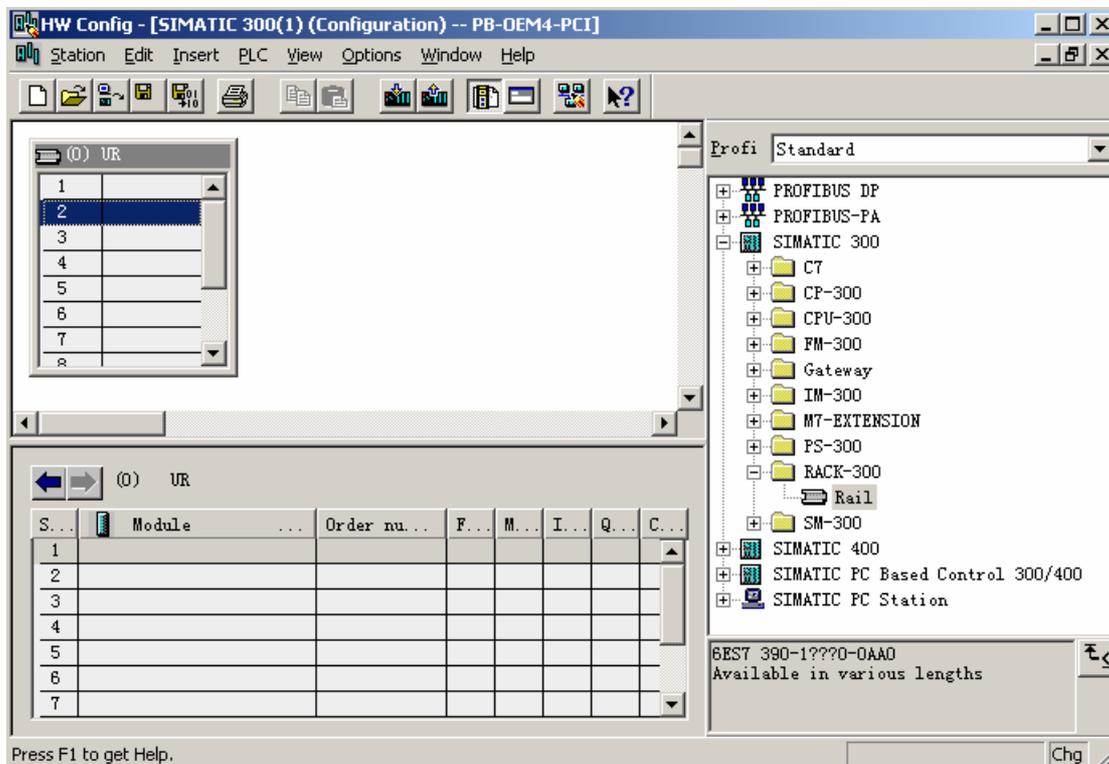


图 3-7

弹出如图 3-8 所示窗口。Address 为主站地址，一般默认为 2; 选择图 3-8 窗口中窗口的“NEW”

按钮新建一个 PROFIBUS 网络，

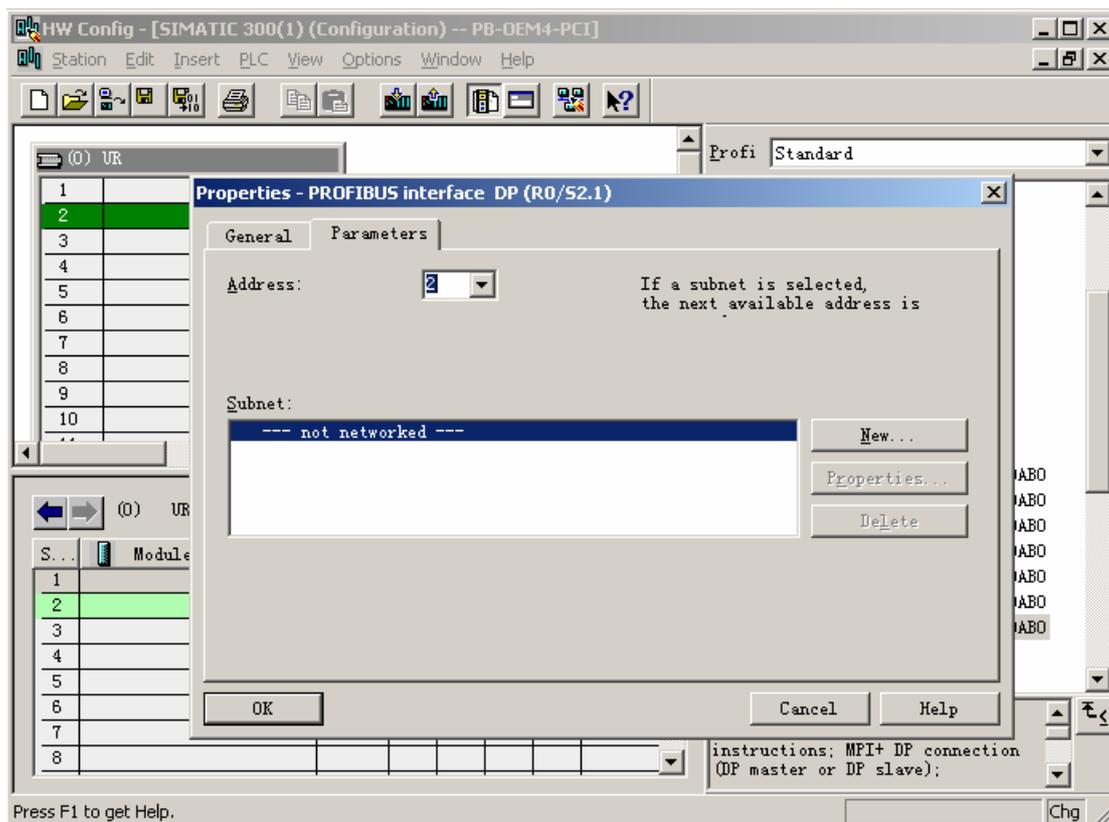


图 3-8

出现图 3-9 所示的窗口，在该窗口中设置 PROFIBUS 网络通讯波特率，一般选择 187.5K，本产品支持 12M 波特率。然后，点击“OK”按钮，退出当前窗口。

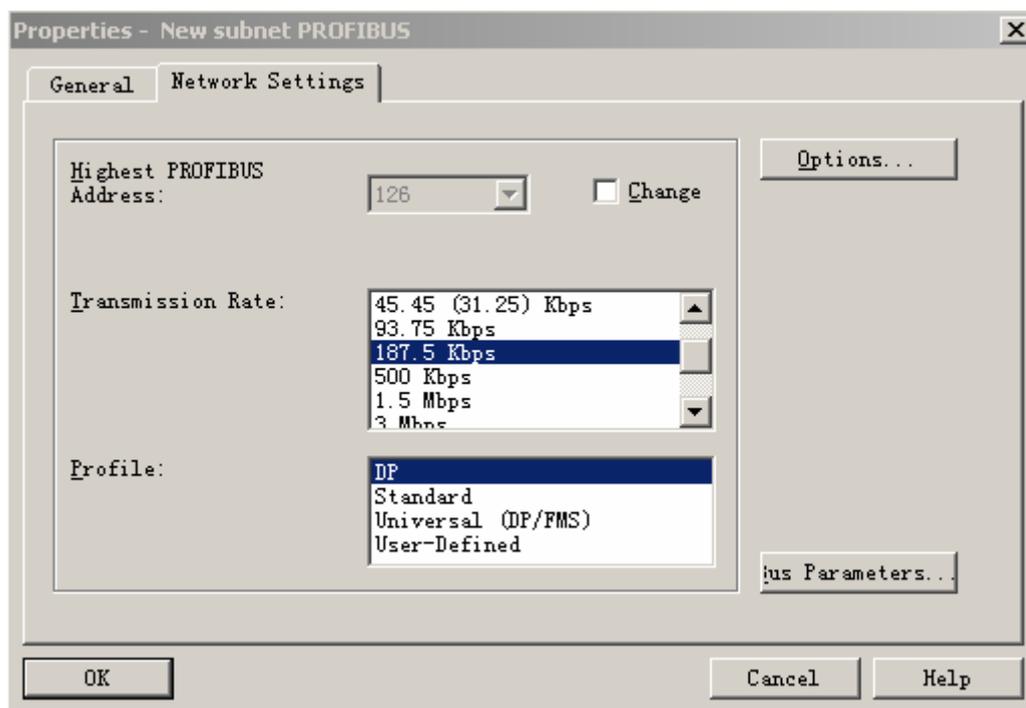


图 3-9

然后图 3-8 所示的窗口变为图 3-10 所示的窗口，

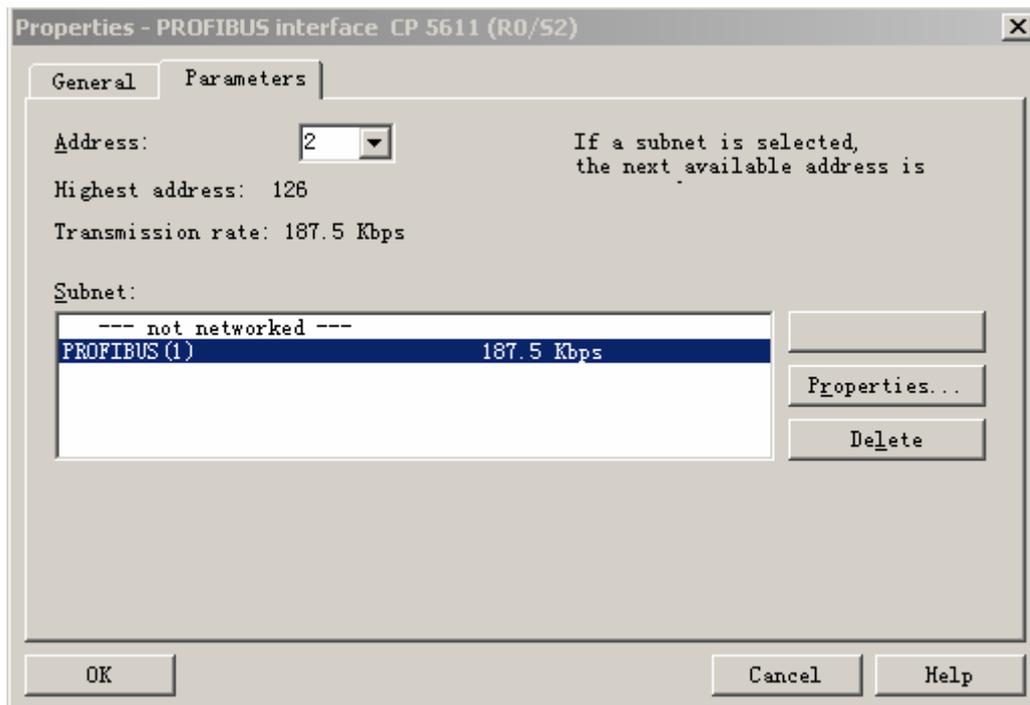


图 3-10

点击图 3-10 所示窗口中的“OK”按钮，退出当前窗口，如图 3-11 所示。

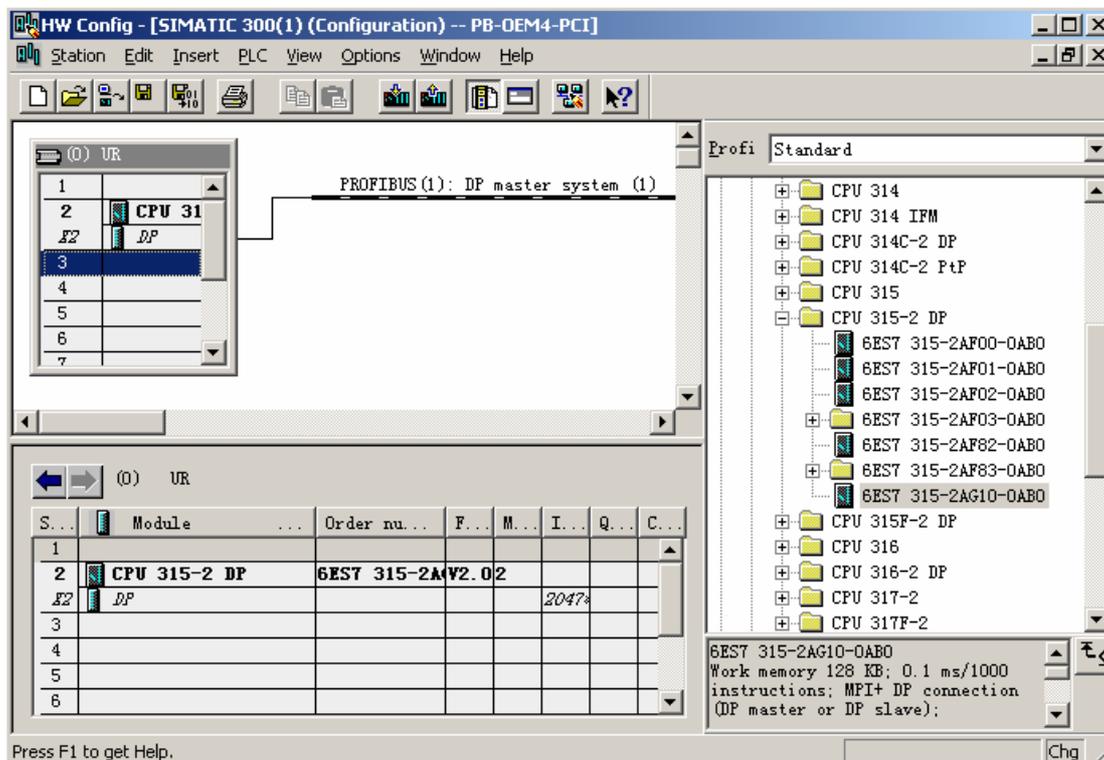


图 3-11

(4) 添加 PROFIBUS DP 从站 PB-OEM4-PCI

用鼠标点击该网络，然后从窗口右边的 Catalog 区域选择 PB-，双击该产品型号，如图 3-12，

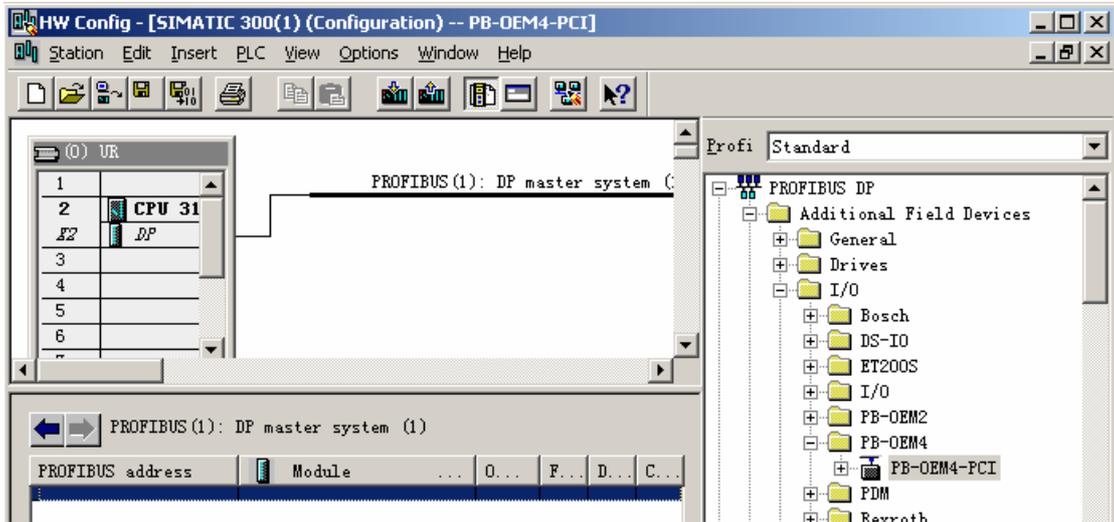


图 3-12

注：如果在 Catalog 中没有找到 PB-OEM4-PCI，首先确认是否将 DS_OEM4.GSD 复制到 SIEMENS\Step7\S7DATA\GSD 目录下，然后用户是否更新。如果没有更新 Catalog，则需要保存当前配置，然后关闭当前配置的子窗口，保留主窗口。选择主窗口“Options→Update Catalog”，如图 3-13 所示。

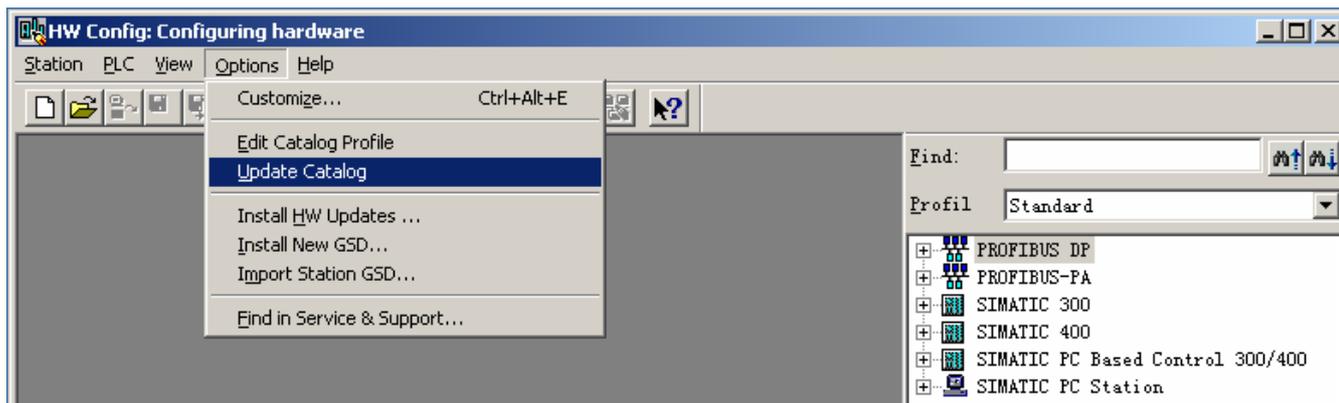


图 3-13

弹出如图 3-12 的窗口，并将从站地址设置为 19（十进制）；该地址必须与实际设备的地址一致。

注：该地址即初始化报文的第一个字节（PROFIBUS 站号），在 VC 例程中将该地址设置 0x13，即 19（十进制）。

选择“OK”退出当前窗口。

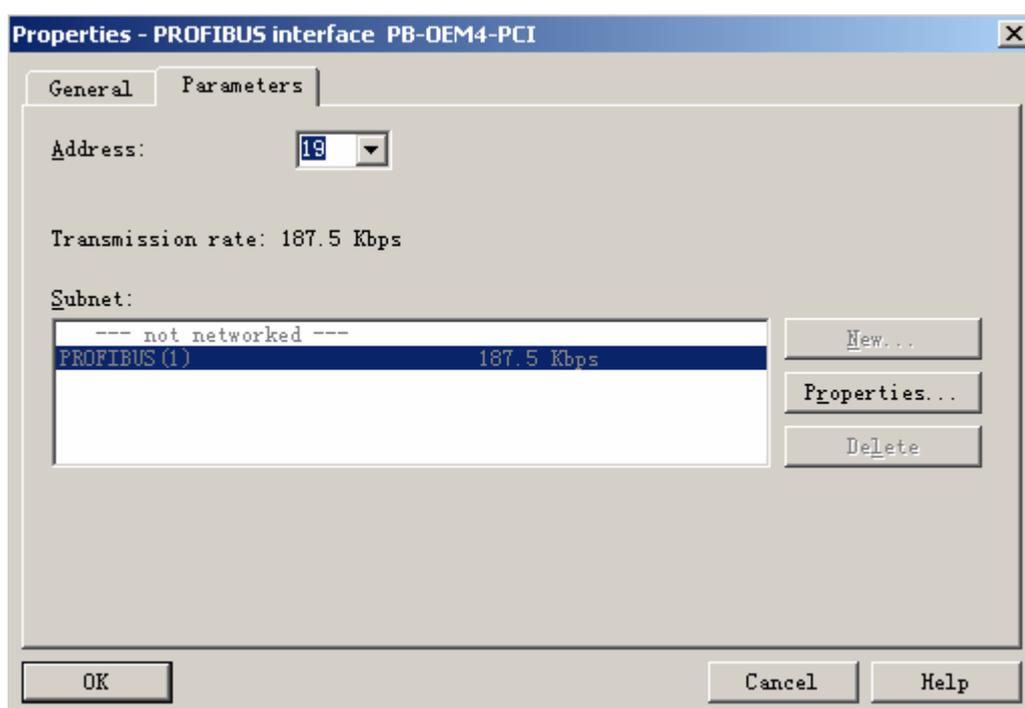


图 3-12

则从站添加到 PROFIBUS 网络中。

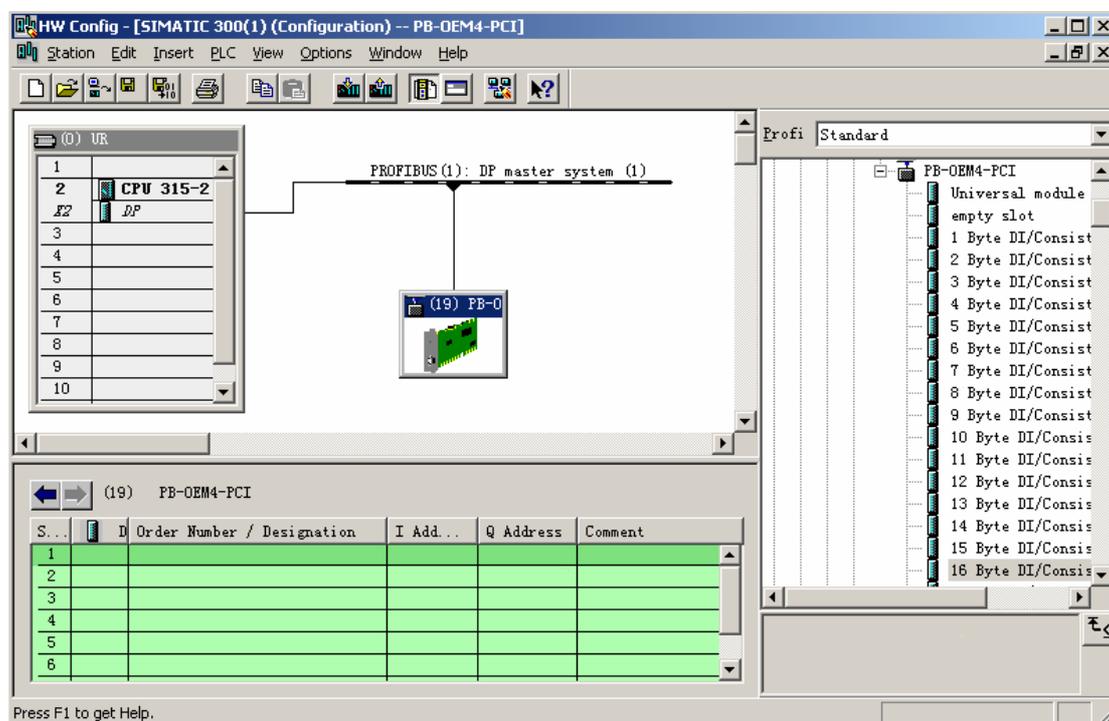


图 3-13

如图 3-13 所示，选中图中从站图标，然后选中插槽 1，鼠标双击 PB-OEM4-PCI 选项中的“16 Byte DI/Consistency 1 Byte”，将 16 Byte DI/Consistency 1 Byte 插入到槽 1 中。依次插入“16

Byte DO/Consistency 1 Byte”。如图 3-14 所示。

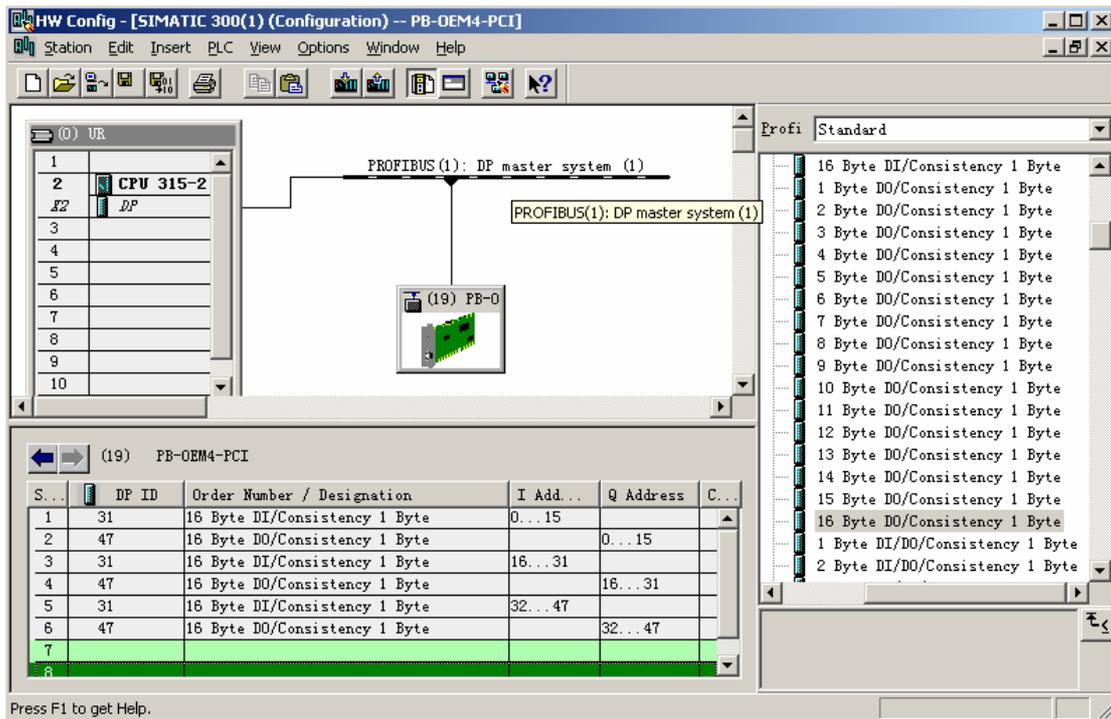


图 3-14

注：该配置与初始化报文中的配置字关系紧密；根据例程的初始化报文，配置字分别为 0x1F,0x2F, 0x1F,0x2F, 0x1F,0x2F。根据表 2-2，分别代表图 3-14 中插入的配置。

点击图 3-14 窗口中工具栏  图标，存盘编译，退出，则完成了 Step 7 配置。

(四) 下载程序

(1) Set PG-PC Interface 的设置



Set PG/PC
Interface

进入“控制面板”，打开 Set PG-PC 的图标 ，如果下载工具使用 CP5611 网卡，则应将 S7ONLINE 指向 CP5611(MPI)，如图 3-15-1。

如果下载工具使用 PC Adapter，则应将 S7ONLINE 指向 PC Adapter (MPI)，如图 3-15-2 所示。

完成后“OK”退出。

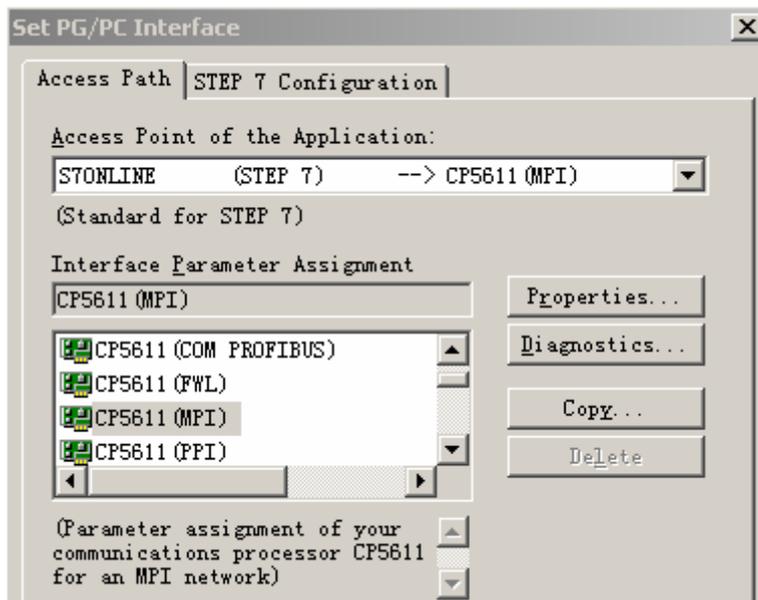


图 3-15-1

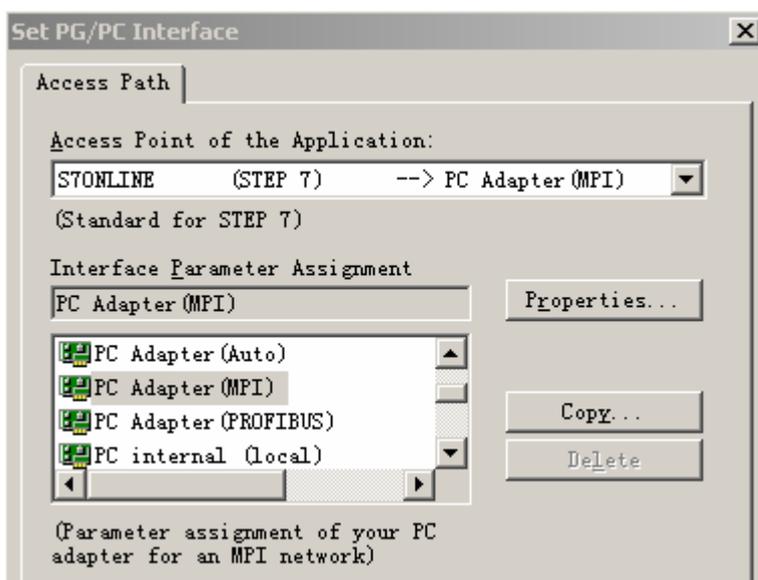


图 3-15-2

(2) 下载程序到 SIMATIC 315-2 DP 中

打开 Step7 的配置界面，将配置信息下载到 315-2 DP 中，如图 3-16 所示，点击工具栏红色标注按钮进行下载。

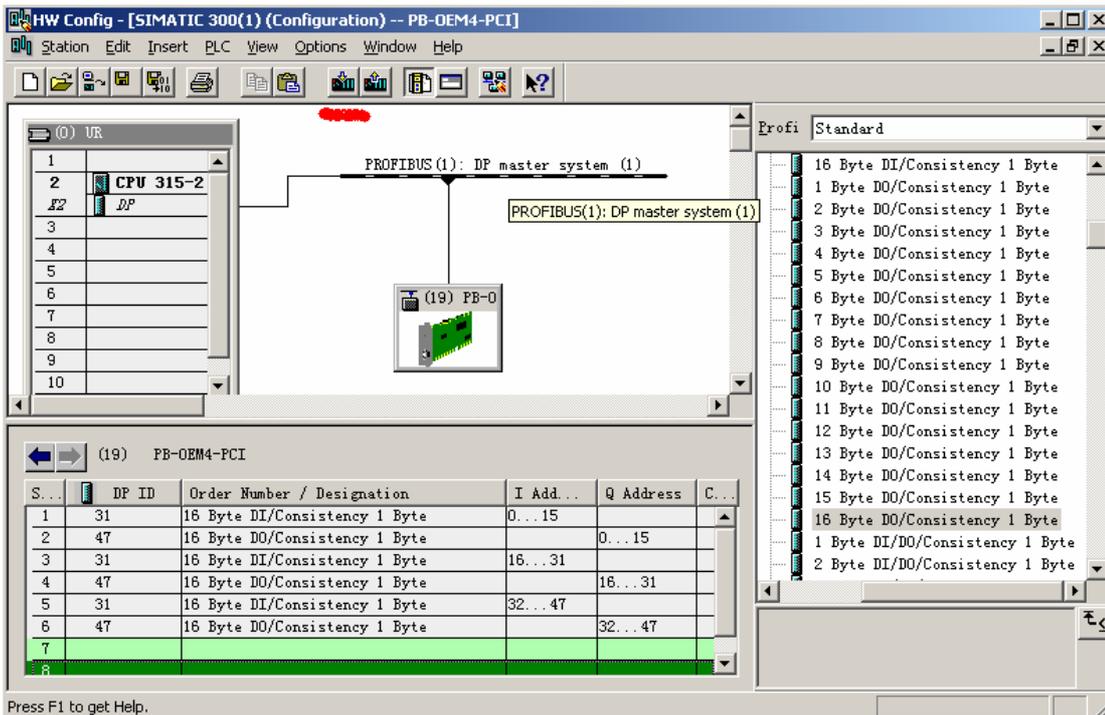


图 3-16

下载完毕后，如果硬件连接正常，运行 PC 机例程，“打开设备”后，则 315-2 DP 上无红色灯闪烁，PROFIBUS 通讯正常，可是数据的接收/发送。

(3) 接收/发送数据的监控

在 STEP 7 新建变量表,对 PROFIBUS 输入(即 PC 程序发送的数据)数据监控。如图 3-17 所示区域鼠标右击弹出所示窗口。

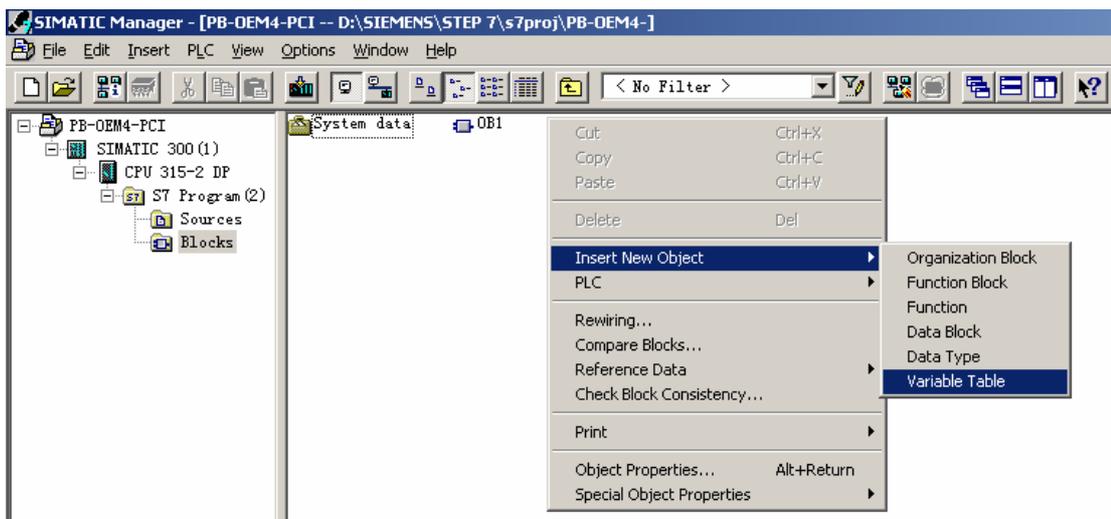


图 3-17

在接着弹出的窗口中将变量表命名为 VAT_1 (默认)。打开该变量表后按图 3-18 中右侧输入

需要监控的变量名。

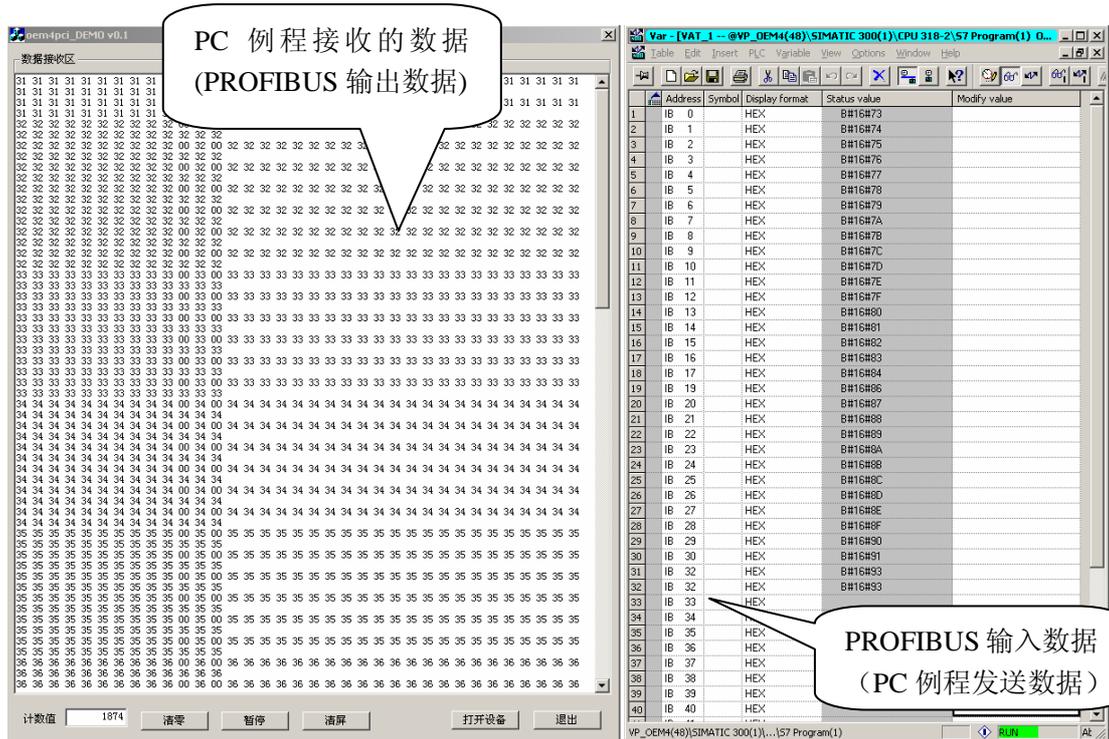


图 3-18

注：PROFIBUS 输出数据的控制是通过 OB35 中梯形图来实现的。在 STEP 7 中 OB35 模块是 100MS 定时触发模块。

第四章 关于 GSD 文件

(一) GSD 文件相关说明

- (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 4-1、图 4-2。

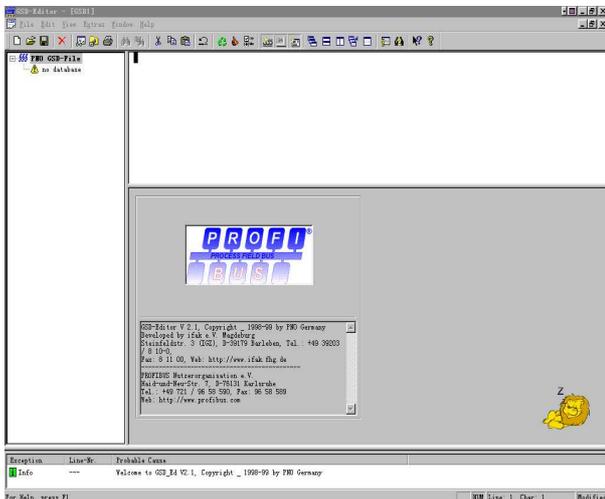


图 4-1 GSDEDIT 打开一个空文件

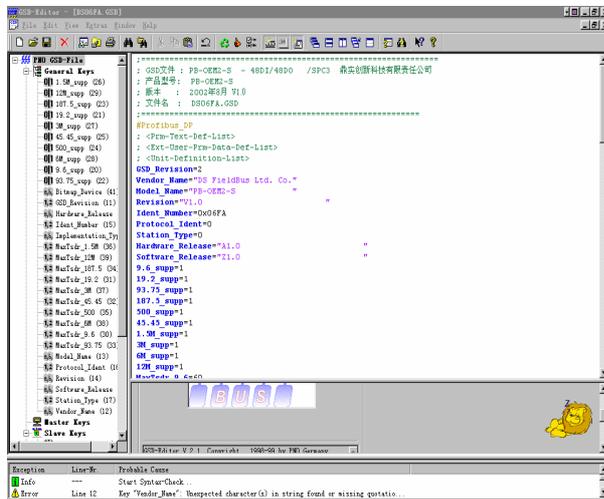


图 4-2 GSDEDIT 打开 DS_06FA.GSD 文件

(二) PB-OEM4-PCI 的 GSD 文件和 ID 号

PB-OEM4-PCI 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_OEM4.GSD。

```

;=====
; Vender      : 鼎实创新科技有限责任公司
; Model       : PB-OEM4-PCI
; Version     : 2007年1月 V1.0
; GSD File    : DS_PCI .GSD
;=====
#Profibus_DP
; <Prm-Text-Def-List>
; <Ext-User-Prm-Data-Def-List>
; <Unit-Definition-List>
GSD_Revision      = 2
Vendor_Name       = "DS FieldBus Co,Ltd.";公司名，可以按用户修改
Model_Name        = "PB-OEM4-PCI"           ;模块名称，在STEP 7中组态该产品的名称
Revision          = "V1.0"
Ident_Number     = 0x06FA                 ;ID号，必须与PC程序的初始化报文一致
Protocol_Ident   = 0
Station_Type     = 0
Hardware_Release = "A1.0"
Software_Release = "Z1.0"
9.6_supp         = 1                     ;支持的波特率
19.2_supp        = 1                     ;支持的波特率
93.75_supp       = 1                     ;支持的波特率
187.5_supp       = 1                     ;支持的波特率
500_supp         = 1                     ;支持的波特率
45.45_supp       = 1                     ;支持的波特率
1.5M_supp        = 1                     ;支持的波特率
3M_supp          = 1                     ;支持的波特率
6M_supp          = 1                     ;支持的波特率
12M_supp         = 1                     ;支持的波特率
MaxTsd_r_9.6     = 60
MaxTsd_r_19.2    = 60
MaxTsd_r_45.45   = 250
MaxTsd_r_93.75   = 60
MaxTsd_r_187.5   = 60
MaxTsd_r_500     = 100
MaxTsd_r_1.5M    = 150
MaxTsd_r_3M      = 250
MaxTsd_r_6M      = 450
MaxTsd_r_12M     = 800
Implementation_Type = "SPC3"
Bitmap_Device    = "si9001pn";图标文件，在组态中看到的图标，用户可以自制图标
; Slave-Specification:
OrderNumber      = "PB-OEM4-PCI"
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         = 1
Max_Module               = 8
Modul_Offset            = 0
Max_Input_Len           = 200
Max_Output_Len          = 200
Max_Data_Len            = 400
Slave_Family             = 3@Tdf@PB-OEM4      ;组态中的分类名
; <Module-Definition-List>;用户配置数据，必须与PC程序的初始化报文和顺序严格一致
Module="empty slot"0x00
EndModule
Module = "1 Byte DI/Consistency 1 Byte" 0x10
EndModule
.....
Module = "16 Word AI/AO/Whole Consistency" 0xFF
EndModule
```

⇒ 用户产品的 ID 号、GSD 文件及产品认证

- (1) PB-OEM4-PCI 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。