

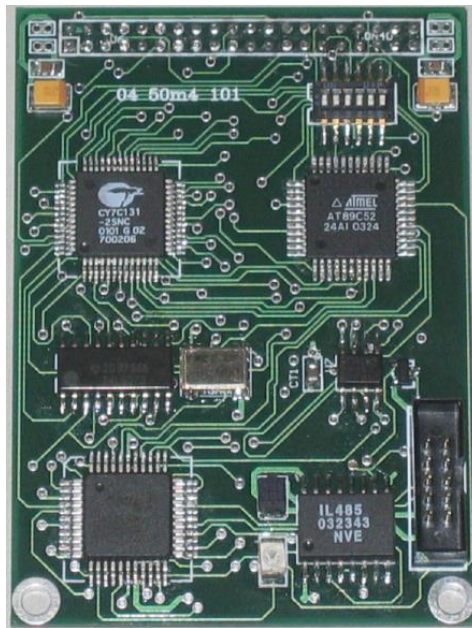
PROFIBUS-DP 从站开发 OEM 解决方案

嵌入式 PROFIBUS 总线桥

PB-OEM1-DPRAM 产品手册

(PB-OEM1-DPRAM 型)

V 1.0



北京鼎实创新科技有限公司

2011-1

目 录

第一章 产品概述	2
主要用途	2
产品系列	2
产品特点	2
技术指标	3
第二章 产品开发工作流程	4
第三章 硬件设计说明	5
一. PB-OEM1-DPRAM 产品硬件原理框图	5
二. PB-OEM1-DPRAM 外形尺寸图	5
三. 40 针接插件管脚定义及说明	6
① 40 针接插件管脚排列	6
② 40 针接插件外型尺寸	7
③ 40 针接插件管脚定义表	7
④ PROFIBUS 插头的硬连接引出方式	7
四. 10 针接插件管脚定义及说明	8
① 10 针接插件管脚排列	8
② 10 针接插件外型尺寸	8
③ 10 针接插件管脚定义表	8
④ PROFIBUS 插头的软连接引出方式	8
⑤ 6 位拨码小开关的说明	9
第四章 软件设计说明	10
一. 通信规约	10
1. 接口板与用户模板对双口 RAM 的读/写控制权	10
2. 接口板双口 RAM 单元地址分区表以及命令字和状态字定义表	10
3. 接口板与用户模板“通信”的基本过程	16
二. 接口板的初始化	17
1. 初始化规约	17
2. 初始化参数	17
3. 初始化流程图	23
三. 接口板与用户模板的数据交换	24
1. 数据交换规约	24
2. 数据交换时序图	25
3. 数据交换流程图	26
四. 如何使用“用户参数”	29
1. 什么情况下需要使用“用户参数”	29
2. 确定“用户参数”类型、个数、字节长度	29
3. 带有“用户参数”描述的 GSD 文件	29
4. 如何在主站配置中选择用户参数	30
5. 用户模板怎样得到“用户参数”	34
第五章 GSD 文件	35
1. 关于 GSD 文件 (Electronic Data Sheet)	35
2. 本产品的 ID 号及 GSD 文件	36
3. 用户产品的 ID 号、GSD 文件及产品认证	36
4. DPRAM.GSD 说明及修改成用户 GSD 文件的说明	37

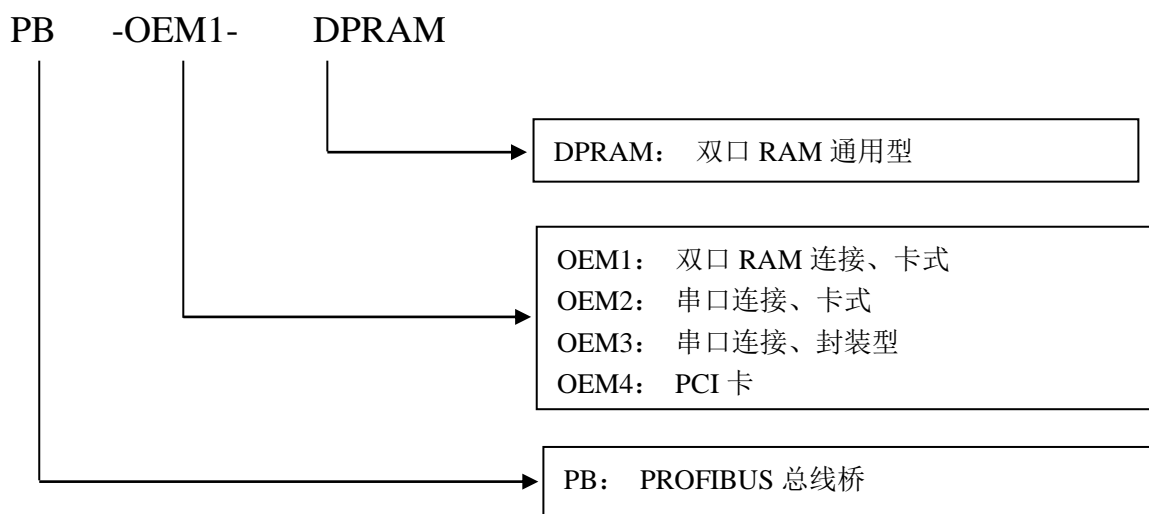
第一章 产品概述

主要用途

本产品专为自主开发具有 PROFIBUS-DP 通信功能的产品，且对实时响应时间有较高要求的用户，以 OEM 方式提供 PROFIBUS-DP 从站通信接口。

产品系列

嵌入式 PROFIBUS 总线桥 PB-OEM1-DPRAM 接口板（以下简称“接口板”）是 PROFIBUS 总线桥 OEM 系列产品，本产品手册涉及到的是 PB-OEM1-DPRAM 型产品。



产品特点

▼**开发周期短：**用户不需要了解 PROFIBUS 开发技术，不需要购买 PROFIBUS 开发系统；不需要自己编写 GSD 文件，可在短时间内完成开发具有自主知识产权的 PROFIBUS 产品；

▼**应用简单：**用户产品的 CPU 通过双口 RAM 向接口板读/写输入/输出数据，接口板自动转换成 PROFIBUS-DP 协议与 PROFIBUS 主站通信。

▼**完整的技术资料：**《PB-OEM1-DPRAM 产品手册》包括：开发实验板 PB-OEM1-DPRAM Evaluation Board 原理图、双口 RAM 通讯协议、通讯程序 C 源码、外形尺寸及接插件尺寸图、40 针接插件管脚定义及说明等，全部资料可在网上下载。

▼**高速：**PB-OEM1-DPRAM 型产品支持 12M 波特率，最快数据更新周期小于 1ms 。

▼**应用广泛：**可广泛用于各种产品，如变频器、电机启动保护装置、智能高低压电器、电量测量装置、各种变送器、智能现场测量设备及仪表等等。

▼**提供方式：**以 OEM 方式提供，因此用户对产品有自主知识产权、品牌、商标注册权。

▼**最大 PROFIBUS-I/O：**112 字节输入+112 字节输出。

▼**可实现 PROFIBUS 用户参数化功能：**用户模板可将一次性设置的参数通过 GSD 文件在主站配置中设定，

如温度上限报警值等；主站在与从站连接时一次将这些用户参数传送到从站，用户模板可使用这些参数实现参数化。用户参数化功能的好处是：避免将一次性设置参数作为 PROFIBUS-I/O，占用周期性数据通信时间和空间资源。

技术指标

- ▼ PROFIBUS-DP 协议，符合 EN57170 V2 和 IEC61158；
- ▼ 标准 PROFIBUS-DP 驱动接口，波特率自适应，最大波特率 12M；
- ▼ PROFIBUS 输入/输出数量可自由设定，最大 112 字节输入+112 字节输出；
- ▼ 可实现 PROFIBUS 用户参数化功能：用户模板可将一次性设置的参数在主站配置中设定，如温度上限报警值等；主站在与从站连接时一次将这些用户参数传送到从站，用户模板可使用这些参数实现参数化。用户参数化功能的好处是：避免将一次性设置的参数作为 PROFIBUS-I/O，占用周期性数据通信时间和空间资源；
- ▼ 接口板与用户板接口为双口 RAM（25ns 读/写周期），用户板按普通 RAM 的读写时序即可对接口板进行操作；
- ▼ 接口板与用户板之间以互相中断和硬件握手方式对双口 RAM 进行数据交换，保证了数据交换的实时性和完整性（一致性）；
- ▼ 接口板与用户模板的数据交换有字校验和，保证了数据的安全性；
- ▼ 需用户板供电 2 组 DC 5V±5%：VCC/GND-150mA 和 5V/0V-100mA；
- ▼ 环境温度：
运输和存储：-40℃~+70℃
工作温度：-20℃~+55℃
- ▼ 工作相对湿度：5~95%
- ▼ 外形尺寸：(宽)50mm×(长)70mm。

第二章 产品开发工作流程

第一步：硬件设计

根据本《手册》“**第三章 硬件设计说明**”中关于本产品的外形尺寸、接口管脚定义、PROFIBUS 插头定义等,设计你的产品硬件原理图、PCB 等。可以参考《**PROFIBUS-OEM1 调试实验系统-1 用户手册**》“**第二章 OEM1 开发实验板**”中实验板的硬件原理图。

第二步：软件设计

根据本《手册》“**第四章 软件设计说明/一、通信规约/二、接口板的初始化/三、接口板与用户模板的数据交换**”编写用户产品与 PB-OEM1-DPRAM 的数据交换部分。可以参考《**PROFIBUS-OEM1 调试实验系统-1 用户手册**》“**第二章 OEM1 开发实验板**”中“**用户模板通信程序**”。

第三步：调试数据通信

这一步是初步调试,主要检查用户模板与 PB-OEM1-DPRAM 接口的数据交换是否正常。参考“**第四章 软件设计说明/一、通信规约//二、接口板的初始化/三、接口板与用户模板的数据交换**”。

第四步：修改 GSD 文件

根据本《手册》“**第五章 GSD 文件**”,使用 GSD-EDIT 软件,将 DPRAM.GSD 文件修改成用户产品 GSD 文件。

必需要改动部分: 用户自定义 I/O 配置数据、ID 号;

可选改动部分: 一些标识,如注释、公司名称、产品名称、型号等。即使都不作改动也不会影响您的产品与 PROFIBUS 主站的连接。如果不能确定某一项改动,GSD-EDIT 软件中的帮助菜单有详细说明。

第五步：建立一个最小调试系统

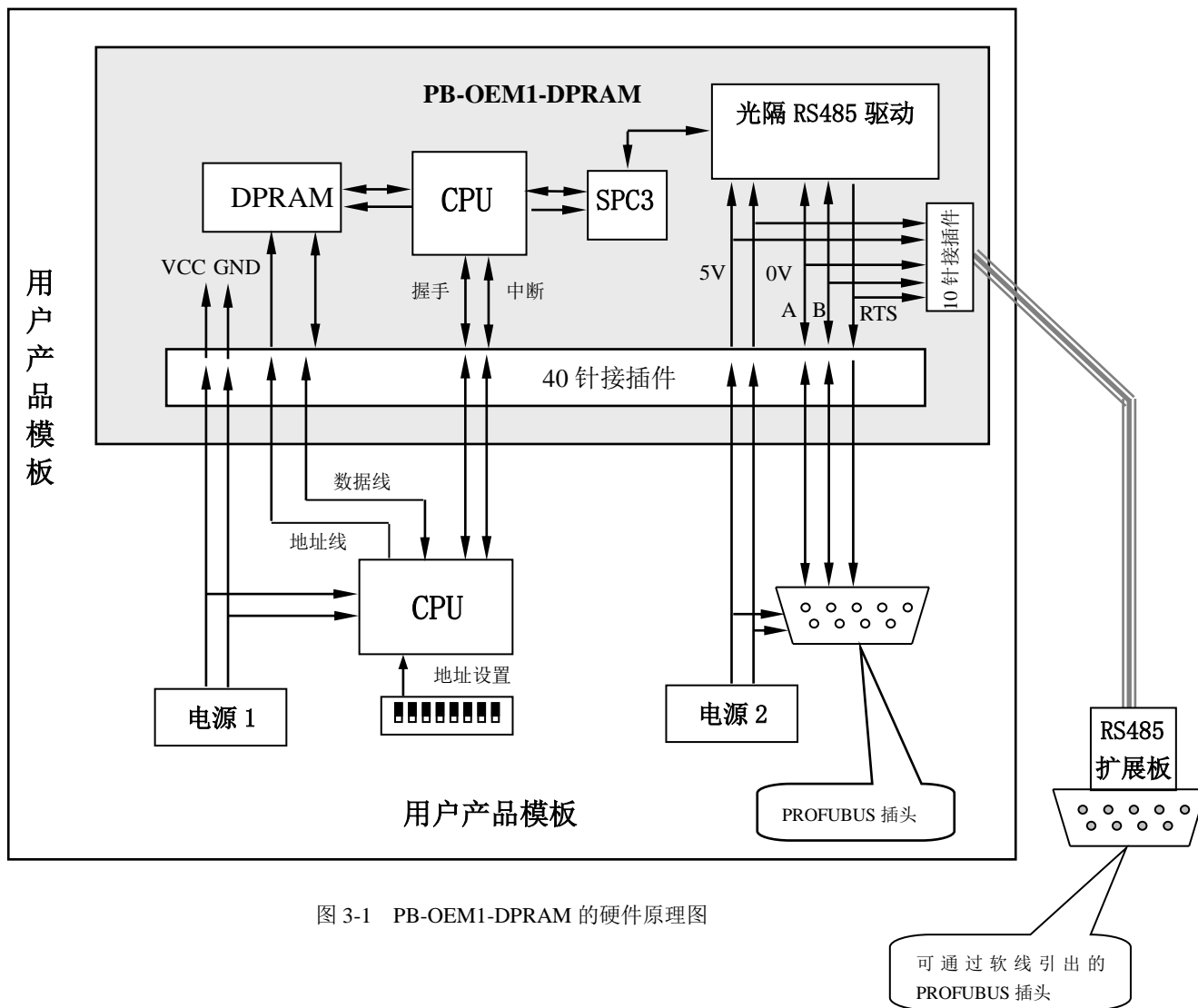
根据《**PROFIBUS-OEM1 调试实验系统-1 用户手册**》或《**PROFIBUS-OEM1 调试实验系统-2 用户手册**》,建立一个最小调试系统。

第六步：安装、配置、调试你的产品

第三章 硬件设计说明

一. PB-OEM1-DPRAM 产品硬件原理框图

PB-OEM1-DPRAM 的硬件原理图如下图 3-1 所示:



二. PB-OEM1-DPRAM 外形尺寸图

PB-OEM1-DPRAM 外形尺寸图如图 3-2:

※※CAD 图纸

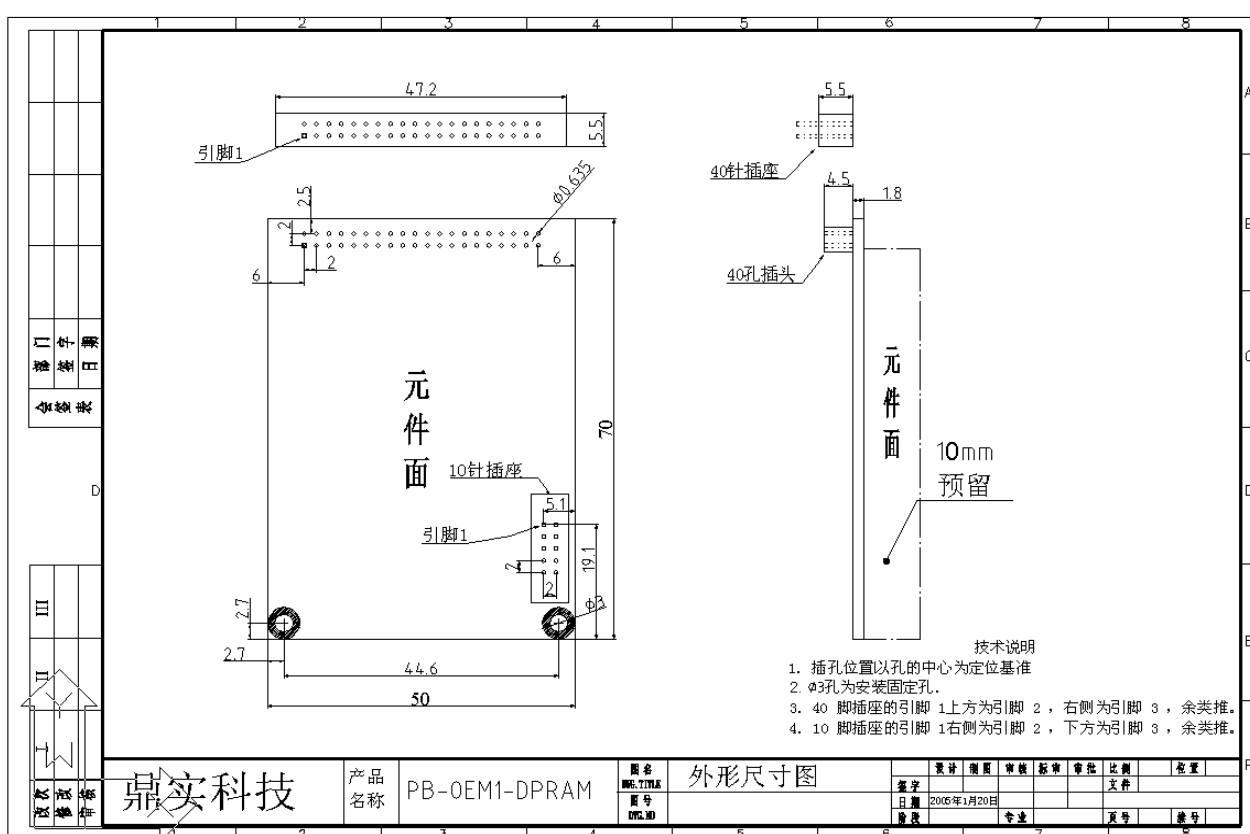


图 3-2 PB-OEM1-DPRAM 外形尺寸图

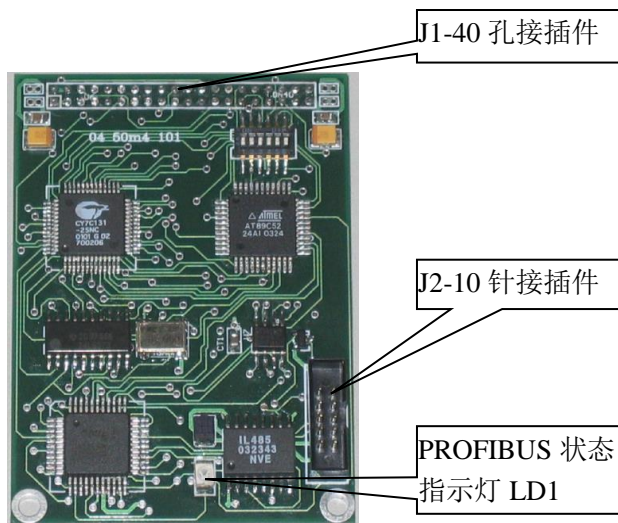


图 3-3 PB-OEM1-DPRAM 外观图

注：因为 40 孔接插件对应的 40 针插座多是凹槽型的，且两端空挡较大。为避免插入时产生错位，所以 [OEM1 接口卡实际配用的是 44 孔接插件，两端各有两个无用的空孔](#)。为叙述方便，本手册中仍按 40 孔接插件进行说明。

三. 40 针接插件管脚定义及说明

① 40 针接插件管脚排列

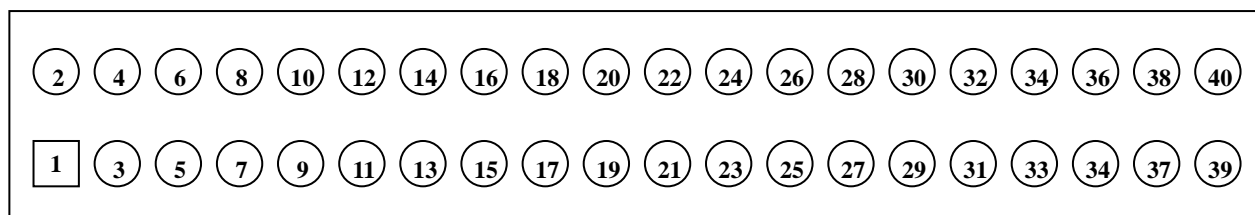


图 3-4 40 针接插件图

② 40 针接插件外型尺寸

见“PB-OEM1-DPRAM 外形尺寸图 3-2”。

③ 40 针接插件管脚定义表

表 1: 40 针接插件管脚定义表

管脚	名称	说明	管脚	名称	说明
1	VCC	VCC, (+5V 直流电源)	2	GND	GND, (VCC 的地)
3	VCC	VCC, (+5V 直流电源)	4	GND	GND, (VCC 的地)
5	<u>INTR</u> (I)	双口 RAM 中断输出信号 (低有效)	6	<u>ResIN</u> (O)	外部输入复位信号 (低有效)
7	A0R(O)	双口 RAM 地址线	8	D0R(I/O)	双口 RAM 数据线
9	A1R(O)	同上	10	D1R(I/O)	同上
11	A2R(O)	同上	12	D2R(I/O)	同上
13	A3R(O)	同上	14	D3R(I/O)	同上
15	A4R(O)	同上	16	D4R(I/O)	同上
17	A5R(O)	同上	18	D5R(I/O)	同上
19	A6R(O)	同上	20	D6R(I/O)	同上
21	A7R(O)	同上	22	D7R(I/O)	同上
23	A8R(O)	同上	24	<u>CER</u> (O)	双口 RAM 片选通信号 (低有效)
25	A9R(O)	同上	26	<u>R/W</u> R(O)	双口 RAM 读 (高有效) / 写 (低有效) 信号
27	A10R(O)	暂不用 (空置)	28	<u>OER</u> (O)	双口 RAM 输出选通信号 (低有效)
29	L-REQ(I)	Profibus CPU ‘请求’ 存取双口 RAM	30	R-REQ(O)	用户模板 CPU ‘请求’ 存取双口 RAM
31	L-STA(I)	Profibus CPU ‘正在’ 存取双口 RAM	32	R-STA(O)	用户模板 CPU ‘正在’ 存取双口 RAM
33	PBF(I)	Profibus 故障信号 (故障时为高电平)	34	RTS	Profibus RTS 信号, 接用户模板 D 型插头④
35	B	Profibus 数据线 B, 接用户模板 D 型插头③	36	A	Profibus 数据线 A, 接用户模板 D 型插头⑧
37	5V	5V 电源 (光隔外)	38	0V	电源地 (光隔外, 与 GND 隔离)
39	5V	5V 电源 (光隔外)	40	0V	电源地 (光隔外, 与 GND 隔离)

表 3-1 40 针接插件管脚定义表

注: 对用户模板而言, (I) 为输入信号, (O) 为输出信号, (I/O) 为双向信号。

对双口 RAM 的读、写时序, 详见 IDT7130 或 CY7C131 的使用说明, 读/写周期为 25ns。

④ PROFIBUS 插头的硬连接引出方式

用户产品模板的 PROFIBUS 插头为 9 孔 (famel) D 型插头, 由接口板 40 针接插件的 34~40 号引脚引出, 引出时建议通过 PCB 板连接, 9 孔 (famel) D 型插头直接焊接在用户模板上。连接方式如下图所示:

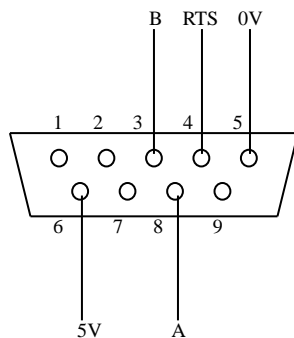


图 3-5 PROFIBUS 插头引线

四. 10 针接插件管脚定义及说明

① 10 针接插件管脚排列

10 针接插件向那些用户模板 PROFIBUS 插头离接口板距离较远的用户，提供了选择电缆软线外引 PROFIBUS 插头的方式。10 针管脚排列如下：

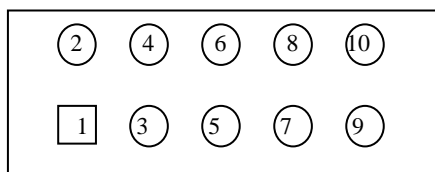


图 3-6 10 针接插件引线

② 10 针接插件外型尺寸

见“PB-OEM1-DPRAM 外形尺寸图 3-2”。

③ 10 针接插件管脚定义表

见下表 3-2:

管脚	名称	说明	管脚	名称	说明
1	5V	5V 电源（光隔外）	2	5V	5V 电源（光隔外）
3	0V	电源地（光隔外，与 GND 隔离）	4	0V	电源地（光隔外，与 GND 隔离）
5	B	Profibus 数据线 B，接用户模板 D 型插头③	6	A	Profibus 数据线 A，接用户模板 D 型插头⑧
7	RTS	RTS 信号反相，（用法见下图）	8	RTS	Profibus RTS 信号，接用户模板 D 型插头④
9	0V	电源地（光隔外，与 GND 隔离）	10	0V	电源地（光隔外，与 GND 隔离）

表 3-2 10 针接插件管脚定义

④ PROFIBUS 插头的软连接引出方式

通过 10 针接插件软线引出的 PROFIBUS 9 孔（famel）D 型插头建议直接焊接在 485 扩展板上，扩展板与接口板之间的软线可采用 5 芯电缆，与两者的 10 针接插件引脚为 1 对 1 连接，距离一般不超过 500mm。485 扩展板的原理图，以及扩展板与接口板之间的软线引出方式如下图 3-7 所示。

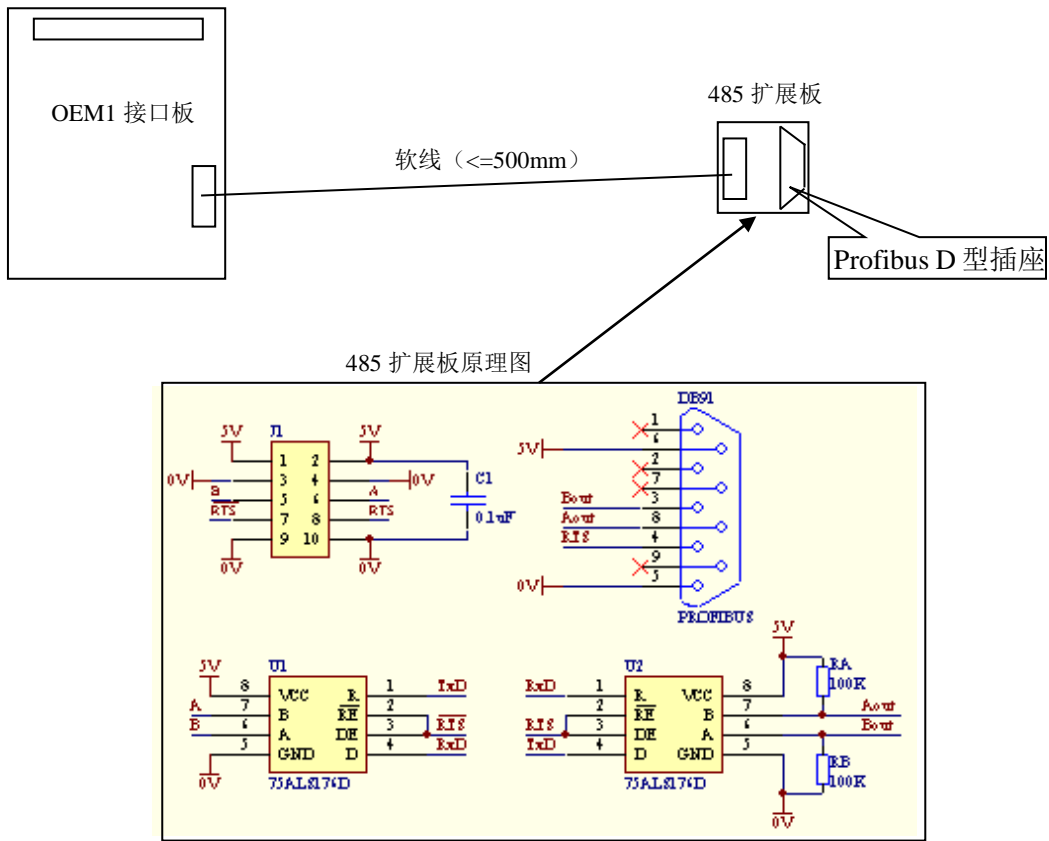


图 3-7 485 扩展板与接口板之间的软线引出方式

⑤ 6 位拨码小开关的说明

该 6 位拨码开关是保留元件，产品出厂前已已全部拨为“ON”，用户使用时不需要改变该开关的状态。

第四章 软件设计说明

一. 通信规约

1. 接口板与用户模板对双口 RAM 的读/写控制权

因为接口板 PB-OEM1-DPRAM 与用户模板（比如我们的开发实验板 PB-OEM1- DPRAM Evaluation Board）之间的所有数据交换都是通过双口 RAM 来进行，为避免由于两者竞争而导致的错误，接口板或用户模板访问双口 RAM 之前均须先取得双口 RAM 读/写控制权（后简称“控制权”）。申请控制权要用到 40 针接插件中的 L-REQ、L-STA、R-REQ、R-STA 4 个硬件握手信号。

▼ 接口模板申请、释放双口 RAM 读/写控制权的协议如下图 4-1 所示：

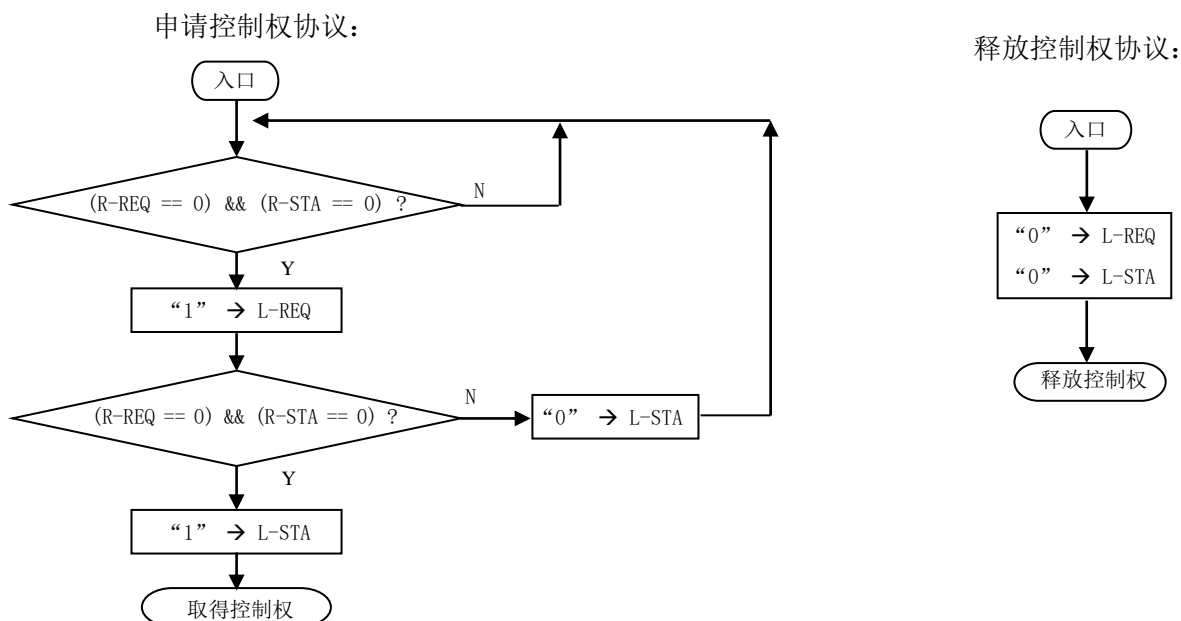


图 4-1 申请、释放控制权

2. 接口板双口 RAM 单元地址分区表以及命令字和状态字定义表

相关术语

- ▼ **初始化数据：**它是由用户模板发给接口板的，用于初始化 Profibus 接口芯片的一组数据。该组数据必须与 GSD 文件中的数据，也就是 Profibus 主站中对从站的硬件配置数据相一致。
- ▼ **输入数据：**输入数据是指 Profibus 主站接收的数据，它的数据流向是“用户模板—>接口板—>主站”
- ▼ **输出数据：**输出数据是指 Profibus 主站发出的数据，它的数据流向是“主站—>接口板—>用户模板”
- ▼ **用户参数数据：**它是指 Profibus 主站与从站之间建立通讯连接的过程中，由主站“一次性”发给从站的附加报文。该报文的内容和长度可以由用户根据需要在 GSD 文件中自行定义。

双口 RAM 单元地址分区表规定了双口 RAM 各地址单元中内容的含义。见表 4-1。

命令字和状态字定义表规定了用户模板与接口板数据交换过程中相互状态的约定。见表 4-2。

表 4-1 双口 RAM (1K) 单元地址分区表

地址	内 容	注 释	地 址	内 容	注 释
0000	版本代码	(保留)	0100	版本月份	(保留)
0001	版本年份	(保留)	0101	版本日期	(保留)
0002	空闲		0102	空闲	
0003		0103	
0004	用户模板状态字节 1 (L_Flag1)	见 表 4	0104	接口板状态字 (S_Flag1)	见 表 4
0005	用户模板状态字节 2 (L_Flag2)	0105	接口板状态字 (S_Flag2)
0006	命令字节 1 (L_Inst1)	来自用户模板	0106	备用	
0007	命令字节 2 (L_Inst2)	
....	备用		
000F		010F	
0010	初始化数据 (L_ini[0])	来自用户模板 (站号)	0110	用户参数数据 1 (S_para[0])	来自 Profibus 主站
0011 (L_ini[1]) (ID 号 H)	01112 (S_para[1])
0012 (L_ini[2]) (ID 号 L)	01123 (S_para[2])
0013 (L_ini[3]) (CFG 长度≤8)	01134 (S_para[3])
0014 (L_ini[4]) (CFG 数据 1)	01145 (S_para[4])
.... (CFG 数据 n)	01156 (S_para[5])
001D (L_ini[13])	备用	01167 (S_para[6])
001E (L_ini[14]) (PB 输入数据长度≤112)
001F (L_ini[15]) (PB 输出数据长度≤112)
0020 (L_ini[16]) (用户参数数据长度 L≤16)	0110+L-1L (S_para[L-1])	L = (L_ini[16])
0021	初始化校验和字 H (L_ini[17])	$\sum 0010-0020$	0110+L	用户参数校验和字 H (S_para[L])	$\sum 0110- (0110 + L-1)$
0022	初始化校验和字 L (L_ini[18])	0110+L+1	用户参数校验和字 L (S_para[L+1])
....	备用		备用	
002F		014F	
0050	Profibus 输入数据字节长度 m (L_PBin[0])	$m \leq 112$	0150	Profibus 输出数据字节长度 n (S_PBout[0])	$n \leq 112$
0051	Profibus 输入数据字节 1 (L_PBin[1])	来自用户模板	0151	Profibus 输出数据字节 1 (S_PBout[1])	来自 Profibus 主站
0052	Profibus 输入数据字节 2 (L_PBin[2])	0152	Profibus 输出数据字节 2 (S_PBout[2])
....
0051+m	Profibus 输入数据字节 m (L_PBin[m])	0150+n	Profibus 输出数据字节 n (S_PBout[n])
....	Profibus 输入数据校验和字 H (L_PBin[m+1])	$\sum 0050-0050+m$	Profibus 输出数据校验和字 H (S_PBout[n+1])	$\sum 0150-0150+n$
....	Profibus 输入数据校验和字 L (L_PBin[m+2])	Profibus 输出数据校验和字 L (S_PBout[n+2])
00C4	备用		01C4	备用	
....	

表 4-2 双口 RAM 内命令字、状态字定义表

		用户模板状态字节 1 L_Flag1	用户模板状态字节 2 L_Flag2	命令字节 1 L_Inst1	命令字节 2 L_Inst2	接口板状态字节 1 S_Flag1	接口板状态字节 2 S_Flag2
Bit0	0						
	1	主板准备好		允许接口板初始化		接口板准备好	“初始化数据”校验和错
Bit1	0						
	1	“初始化数据”有效				接口板初始化已完成	“输入数据”校验和错
Bit2	0						
	1	“输入数据”有效				“用户参数”数据有效	“双口 RAM 自检”出错
Bit3	0						
	1					“输出数据”有效	“申请控制权”超时错
Bit4	0						
	1					接口板“正在初始化”	“输入数据长度”超限错
Bit5	0						
	1					Profibus“主站连通”	“输出数据长度”超限错
Bit6	0						
	1						“用户参数长度”超限错
Bit7	0						
	1	本状态字有效		本命令字有效		本状态字有效	本状态字有效

读取后应将其清‘0’

注 1：用户模板侧从双口 RAM 读取“用户参数”后，须将 S_Flag1.Bit2 清“0”。

注 2：用户模板侧从双口 RAM 读取“输出数据”后，须将 S_Flag1.Bit3 清“0”。

注 3：接口板一侧从双口 RAM 读取“输入数据”后，亦将 L_Flag1.Bit2 清“0”。

注 4：“申请控制权”超时的时限约为 0.3ms。

3. 接口板与用户模板“通信”的基本过程

接口板与用户模板的数据交换基本过程如下：

① 初始化过程

用户模板上电后等待约 150ms，申请控制权，向双口 RAM 写初始化数据并启动接口板初始化，释放控制权。然后监视接口板的状态变化（须申请、释放控制权）。若接口板初始化成功，则初始化结束。否则重新进行初始化。

接口板初始化完成后，若 Profibus 主站尚未连接，则接口板处于等待主站呼叫状态，接口板上的红色 LED 灯不熄灭，40 针接插件的 PBF 信号保持高电平，S_Flag1 = 0x83、S_Flag2 = 0x80。

接口板初始化完成后，若 Profibus 主站已正确配置和连接，则接口板与主站间应能建立起正常通讯，接口板上的红色 LED 灯应熄灭，40 针接插件的 PBF 信号变为低电平。S_Flag1 & (0xA7) = 0xA3（无用户参数）或 = 0xA7（有用户参数）、S_Flag2=0x80。

▼ **初始化过程中只使用双口 RAM 读/写控制权协议，而不使用中断。**

② 数据交换过程

初始化完成后，接口板以循环方式通过 Profibus 与主站进行周期性数据交换，同时也通过双口 RAM 与用户模板进行周期性的数据交换。

在每个数据交换循环周期中，接口板若已收到主站新的输出数据（主站→接口板），就（先取得控制权，后同）将新的输出数据写入双口 RAM，同时将输入数据缓冲区的数据发往主站，然后写双口 RAM 的 03FF 单元，向用户模板发 INTR 中断（并释放控制权，后同）。在此周期中接口板若未收到主站新的输出数据，则只将输入数据缓冲区的数据发往主站。

用户模板响应 INTR 中断后，（先取得控制权，后同）读取双口 RAM 内的输出数据（主站→接口板），同时将输入数据（接口板→主站）写入双口 RAM。然后读双口 RAM 的 03FF 单元清 INTR 中断；写双口 RAM 的 03FE 单元，向接口板发 INTL 中断（并释放控制权，后同）。

接口板响应 INTL 中断后，（先取得控制权，后同）读取双口 RAM 内的输入数据（接口板→主站），将其传送到输入数据缓冲区，然后读双口 RAM 的 03FE 单元清 INTL 中断，（并释放控制权，后同）。

由此完成了一次完整的数据交换过程。

▼ **数据交换过程中不仅使用相互中断，同时使用双口 RAM 读/写控制权协议。**

二. 接口板的初始化

1. 初始化规约

- ▼ 用户模板访问双口 RAM 之前应首先取得控制权，访问之后应立即释放控制权（后同）。
- ▼ 用户模板上电后等待约 150ms，若检测到“接口板准备好”（S_Flag1==0x81、S_Flag2==0x80），则可将初始化参数（包括站号、ID 号、Cfg 数据长度、Cfg 数据、Profibus 输入数据长度、Profibus 输出数据长度、用户参数长度以及初始化信息校验和）写入双口 RAM 的初始化数据区（双口 RAM 地址：0x0010-0x0022）。
- ▼ 用户模板给出状态字节 L_Flag1=0x83（用户模板准备好）和命令字节 L_Inst1=0x81（允许接口板初始化）。
- ▼ 用户模板不断检测接口板的状态，直至“接口板完成初始化”（S_Flag1==0x83、S_Flag2==0x80）。
- ▼ 若“初始化数据校验出错”，则接口板给出标志：S_Flag2≠0x80。

2. 初始化参数

站号（L_ini[0]）

- ▼ 用户模板应设有设置站号的拨码开关或可以有键盘、手持终端等其他方式设置 PROFIBUS 从站站号；从站站号范围：0~126。
- ▼ 用户模板设置的站号必需和你在主站配置时设置的站号一致；如使用 STEP 7，在 HARDWARD 配置中，你的用户模板配置的站号要和你在用户模板上设置的站号一致。

ID 号（高字节：L_ini[1]、低字节：L_ini[2]）

- ▼ 经过 PI 认证的产品都有全球唯一的一个 ID 号。未经测试认证的设备可先自定义一个 ID 号，比如：0080H 临时使用，产品开发完成后再正式申请 ID 号。
- ▼ 产品 ID 号必需和你的 GSD 文件中的 ID 号一致；

I/O 配置（CFG）数据 L_ini[4]—L_ini[11] 和 I/O 配置（CFG）数据长度 L_ini[3]

- ▼ I/O 配置数据规定了 Profibus 从站的 I/O 数量；本产品的 I/O 数量可在 1—112 字节范围内任意设定；I/O 配置数据请见下表中的“代码”
- ▼ I/O 配置数据长度 L_ini[3]的值等于 I/O 配置数据代码的个数。

本产品 I/O 配置数据长度 L_ini[3]的最大值应 ≤ 8；

表 4-3: 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

例 1: 用户产品需要: 20 个字节输入+10 个字节输出

查表 4-3, 20 个字节输入代码为: 0x1f(16 byte input 完整), 0x13(4 byte input 完整);

10 个字节输出代码为: 0x29(10 byte output 完整)

因此: I/O 配置数据长度 $L_ini[3] = 3$;

I/O 配置数据 $L_ini[4] \sim L_ini[6]$ 为: 0x1f, 0x13, 0x29

例 2: 用户产品需要:

I/O	I/O 统计
设备状态: 上传 6 个开关量 DI	1 个字节输入
设备采集温度、压力等共 12 个模拟量, 每个模拟量占 2 个字节;	12 个字输入
设备控制: 共计 4 个开关输出 DO	1 个字节输出
设备控制: 共计 2 个模拟量输出, 每个模拟量占 2 个字节;	2 个字输出

查表 4-3, 1 个字节输入代码为: 0x10(1 byte input, Byte 完整)

12 个字输入代码为: 0x5B(12 word input, word 完整)

1 个字节输出代码为: 0x20(1 byte output, Byte 完整)

2 个字输出代码为: 0x61(2 word output, word 完整)

因此: I/O 配置数据长度 $L_ini[3] = 4$;

I/O 配置数据 $L_ini[4] \sim L_ini[7]$ 为: 0x10, 0x5B, 0x20, 0x61

PB 输入数据长度 ($L_ini[14]$)

它等于 IO 配置 CFG 数据中输入数据字节个数之和。因此: 在上述的例 1、例 2 中 $L_ini[14]$ 应分别为 0x14 (20 个字节输入)、0x19 (1 字节+12 字 = 25 个字节输入)。

$L_ini[14]$ 的最大值应 ≤ 112 字节;

PB 输出数据长度 ($L_ini[15]$)

它等于 IO 配置 CFG 数据中输出数据字节个数之和。因此: 在上述的例 1、例 2 中 $L_ini[15]$ 应分别为 0x0A (10 个字节输出)、0x05 (1 字节+2 字 = 5 个字节输出)。

$L_ini[15]$ 的最大值应 ≤ 112 字节;

用户参数长度 ($L_ini[16]$)

它等于用户参数的字节个数, 最大为 16; 如果不使用用户参数, 必须使 $L_ini[16]=0$ 。

初始化校验和 ($L_ini[17]$)、($L_ini[18]$)

它等于 $L_ini[0] \sim L_ini[16]$ 之和;

$L_ini[17] = \sum (L_ini[0] \sim L_ini[16])$ 的高字节、 $L_ini[18] = \sum (L_ini[0] \sim L_ini[16])$ 的低字节。

若 I/O 配置数据少于 8 个字节 ($L_ini[3] < 8$), 则 $L_ini[4] \sim L_ini[13]$ 中的空余部分可以是任意数据, 这些数据也包含在校验和中。

初始化错误号

接口板开始初始化之前, 会首先检查初始化数据的校验和。

若 $L_ini[17]$ 、 $L_ini[18] \neq \sum (L_ini[0] \sim L_ini[16])$ ，则将接口板状态字 (S_Flag2) 的 D0 位和 D7 位置“1”。

修改 GSD 文件

初始化报文中的 I/O 配置数据、用户参数长度必须和 GSD 文件中描述的完全一致，包括：顺序、个数、数值。

例 3: 站号设置为 50:	$L_ini[0] = 0x32;$
ID 号为 0x06FA:	$L_ini[1] = 0x06, L_ini[2] = 0xFA ;$
I/O 配置: 20 个字节输入:	$L_ini[14] = 0x14;$
10 个字节输出:	$L_ini[15] = 0x0A;$
I/O 配置数据长度为 2:	$L_ini[3] = 0x03;$
I/O 配置数据为:	$L_ini[4] \sim [6] = 0x1f, 0x13, 0x29; L_ini[7] \sim [13]$ 任意;
没有用户参数:	$L_ini[16] = 0x00;$
初始化校验和字:	$L_ini[17] = \sum (L_ini[0] \sim L_ini[16])$ 的高字节; $L_ini[18] = \sum (L_ini[0] \sim L_ini[16])$ 的低字节;

相应 GSD 文件中必须有:

Ident_Number=0x06FA
User_Prm_Data_Len=0
Module="20bytes in, 10bytes out" 0x1f, 0x13, 0x29
EndModule

例 4: 站号设置为 50:	$L_ini[0] = 0x32;$
ID 号为 0x06FA:	$L_ini[1] = 0x06, L_ini[2] = 0xFA ;$
I/O 配置: 20 个字节输入:	$L_ini[14] = 0x14;$
10 个字节输出:	$L_ini[15] = 0x0A;$
I/O 配置数据长度为 3:	$L_ini[3] = 0x03;$
I/O 配置数据为:	$L_ini[4] \sim [6] = 0x1f, 0x13, 0x29; L_ini[7] \sim [13]$ 任意;
用户参数长度为 2:	$L_ini[16] = 0x02;$
初始化校验和字:	$L_ini[17] = \sum (L_ini[0] \sim L_ini[16])$ 的高字节; $L_ini[18] = \sum (L_ini[0] \sim L_ini[16])$ 的低字节;
用户参数数据为 10, 30:	$S_para[0] \sim [1] = 0x0A, 0x1E; S_para[2] \sim [9]$ 任意;
用户参数校验和字:	$S_para[10] = \sum (S_para[0] \sim S_para[9])$ 的高字节; $S_para[11] = \sum (S_para[0] \sim S_para[9])$ 的低字节;

相应: GSD 文件中必须有:

Ident_Number=0x06FA

User_Prm_Data_Len=2

Module="20bytes in, 10bytes out" 0x1f, 0x13, 0x29

EndModule

例 5: 站号设置为 50:

$L_ini[0] = 0x32;$

ID 号为 0x06FA:

$L_ini[1] = 0x06, L_ini[2] = 0xFA ;$

I/O 配置: 1 个字节输入+12 个字输入=25 个字节输入:

$L_ini[14] = 0x19;$

1 个字节输出+2 个字输出=5 个字节输出:

$L_ini[15] = 0x05;$

I/O 配置数据长度:

$L_ini[3] = 0x04;$

I/O 配置数据为:

$L_ini[4] \sim [7] = 0x10, 0x5B, 0x20, 0x61; L_ini[8] \sim [13]$ 任意;

用户参数长度为 10:

$L_ini[16] = 0x0A;$

初始化校验和字:

$L_ini[17] = \sum (L_ini[0] \sim L_ini[16])$ 的高字节;

$L_ini[18] = \sum (L_ini[0] \sim L_ini[16])$ 的低字节;

用户参数数据为 01,02,03,04,05,06,07,08,09,10:

$S_para[0] \sim [9] = 0x01, 0x02, 0x03, 0x04, 0x05,$

$0x06, 0x07, 0x08, 0x09, 0x0A;$

用户参数校验和字:

$S_para[10] = \sum (S_para[0] \sim S_para[9])$ 的高字节;

$S_para[11] = \sum (S_para[0] \sim S_para[9])$ 的低字节;

相应: GSD 文件中必须有:

User_Prm_Data_Len=10

Module="1byte+12words in,1byte+2words out" 0x10, 0x5b, 0x20, 0x61

EndModule

3. 初始化流程图

初始化过程如图 4-2，（右侧为用户模板初始化过程，图中初始化参数为举例方便使用，用户可根据实际情况更改。左侧为接口板初始化过程，供对比参考，不感兴趣的用户可只看右侧流程图。）

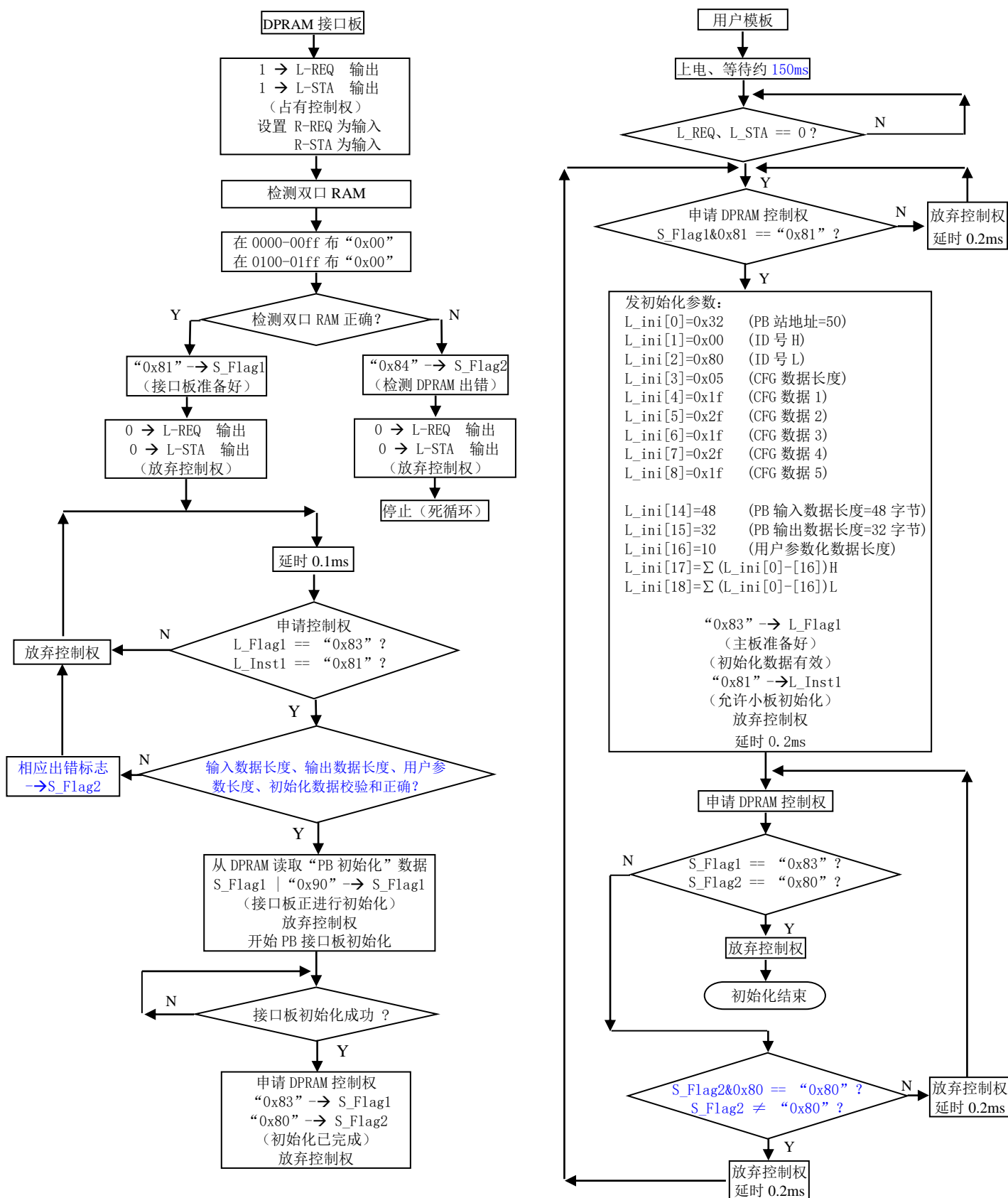


图 4-2 初始化流程图

三. 接口板与用户模板的数据交换

1. 数据交换规约

- ▼ 用户模板访问双口 RAM 之前应首先取得控制权，访问之后应立即释放控制权（后同）。
- ▼ 接口板与用户模板之间的数据交换是周期性的，数据交换以相互中断的方式进行。接口板完成初始化并与 Profibus 主站连通之后即进入周期性数据交换主循环。在每次主循环中，接口板查询和读/写 Profibus 接口芯片一次，读/写双口 RAM 数据一次，并引发接口板与用户模板之间的相互中断一次。首次中断由接口板触发用户模板。
- ▼ 接口板写双口 RAM 的 **03FF** 单元时向用户模板发 **INTR** 中断（使 **INTR** 变为低电平）、读双口 RAM 的 **03FE** 单元时清除来自用户模板的 **INTL** 中断（使 **INTL** 变为高电平）。
- ▼ 用户模板写双口 RAM 的 **03FE** 单元时向接口板发 **INTL** 中断（使 **INTL** 变为低电平）、读双口 RAM 的 **03FF** 单元时清除来自接口板的 **INTR** 中断（使 **INTR** 变为高电平）。
- ▼ 输出数据：在每次主循环中，若接口板从接口芯片收到来自 Profibus 主站的输出数据，则将该数据传送到双口 RAM 的输出数据区（自 **S_PBout[1]** 开始），并将输出数据字节长度 **n** 写入 **S_PBout[0]**（ $n = L_ini[15]$ ），将输出数据的校验和字写入 **S_PBout[n+1]** 和 **S_PBout[n+2]**，置 **S_Flag1.Bit3** = “1”。
- ▼ 输入数据：在每次主循环中，接口板将输入数据缓冲区内的数据（来自用户模板）传送到接口芯片，发往 Profibus 主站。
- ▼ 用户参数数据：在每次主循环中，若接口板从接口芯片收到来自 Profibus 主站的用户参数数据，则将该数据传送到双口 RAM 的用户参数数据区（自 **S_para[0]** 开始），用户参数数据字节长度 $L = L_ini[16]$ ，将用户参数数据的校验和字写入 **L** 个参数的后续单元 **S_para[L]** 和 **S_para[L+1]**，置 **S_Flag1.Bit2** = “1”。
- ▼ 在每次主循环的末尾，接口板向用户模板发 **INTR** 中断。
- ▼ 用户模板响应 **INTR** 中断后，若 **S_Flag1.Bit3** = “1”（输出数据有效），则校验、读取双口 RAM 内的输出数据（自 **S_PBout[1]** 开始）。并置 **S_Flag1.Bit3** = “0”。
- ▼ 用户模板响应 **INTR** 中断后，应将 Profibus 主站需要的输入数据写入双口 RAM 的输入数据区（自 **L_PBin[1]** 开始），然后将输入数据字节长度 **m** 写入 **L_PBin[0]**（ $m = L_ini[14]$ ），将输入数据的校验和字写入 **L_PBin[m+1]** 和 **L_PBin[m+2]**，并置 **L_Flag1.Bit2** = “1”。
- ▼ 用户模板响应 **INTR** 中断后，若 **S_Flag1.Bit2** = “1”（用户参数化数据有效），则校验、读取双口 RAM 内的用户参数化数据（自 **S_para[0]** 开始）。置 **S_Flag1.Bit2** = “0”。
- ▼ 用户模板结束中断程序之前，应清 **INTR** 中断、发 **INTL** 中断。
- ▼ 接口板响应 **INTL** 中断后，若 **L_Flag1.Bit2** = “1”（输入数据有效），则校验、读取双口 RAM 内的输入数据（自 **L_PBin[1]** 开始），将其传送到输入数据缓冲区内，置 **L_Flag1.Bit2** = “0”；然后清 **INTL** 中断，由此完成一次完整的数据交换过程。

2. 数据交换时序图

接口板与用户模板之间的数据交换的时序图如下图 4-3:

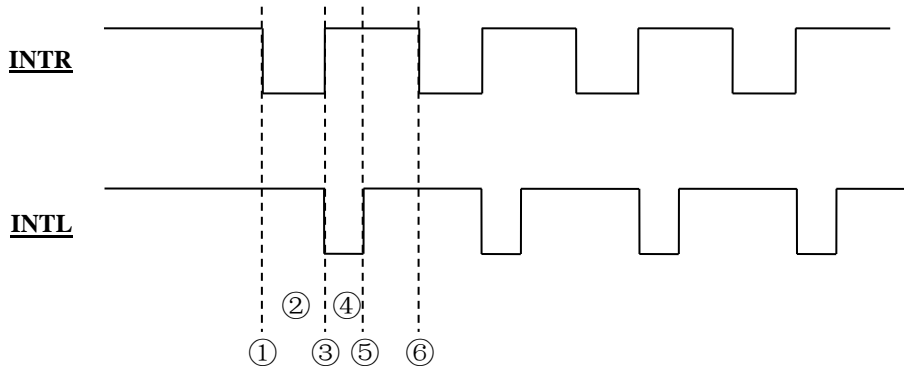


图 4-3 接接口板与用户模板间数据交换时序图

- ① 接口板向用户模板发首次 INTR 中断;
- ② 用户模板中断程序执行时间;
- ③ 用户模板中断程序结束, 清除 INTR 中断、并向接口板发 INTL 中断;
- ④ 接口板中断程序执行时间
- ⑤ 接口板中断程序结束, 清除 INTL 中断;
- ⑥ 接口板向用户模板发下一次 INTR 中断;

上图中, 波形的时间宽度是示意性的, 具体的时间值与输入/输出数据的长度、用户摸板 CPU 的运行速度、Profibus 总线的波特率、以及 Profibus 站点的个数有关。

3. 数据交换流程图

① DPRAM 接口板主循环流程图

见图 4-4（不感兴趣的用户可略过此图）

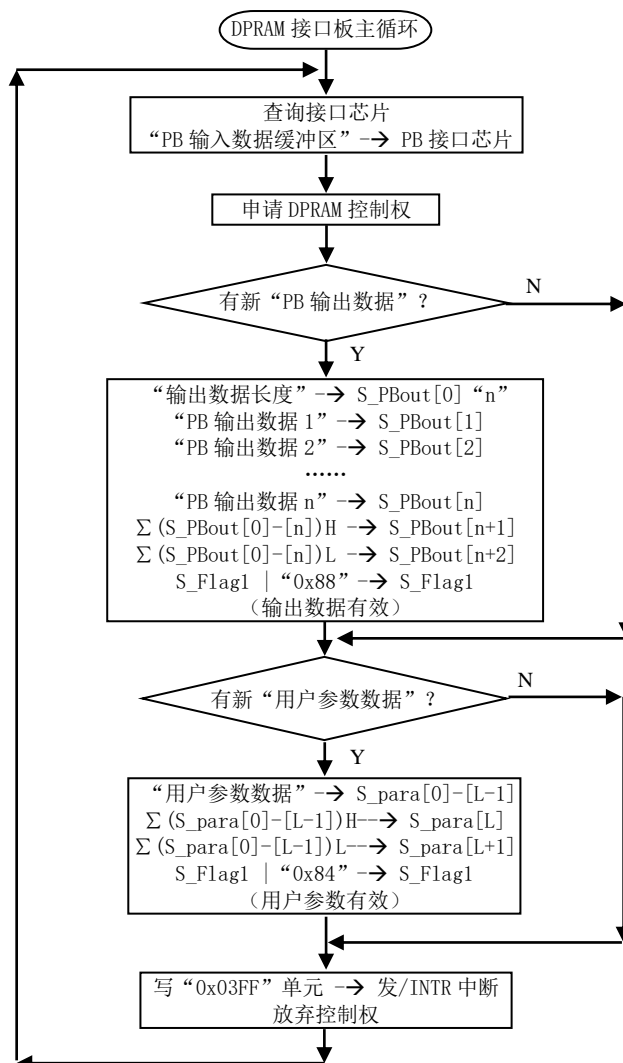


图 4-4 DPRAM 接口板主循环流程图

② 用户模板中断服务程序流程图

见图 4-5

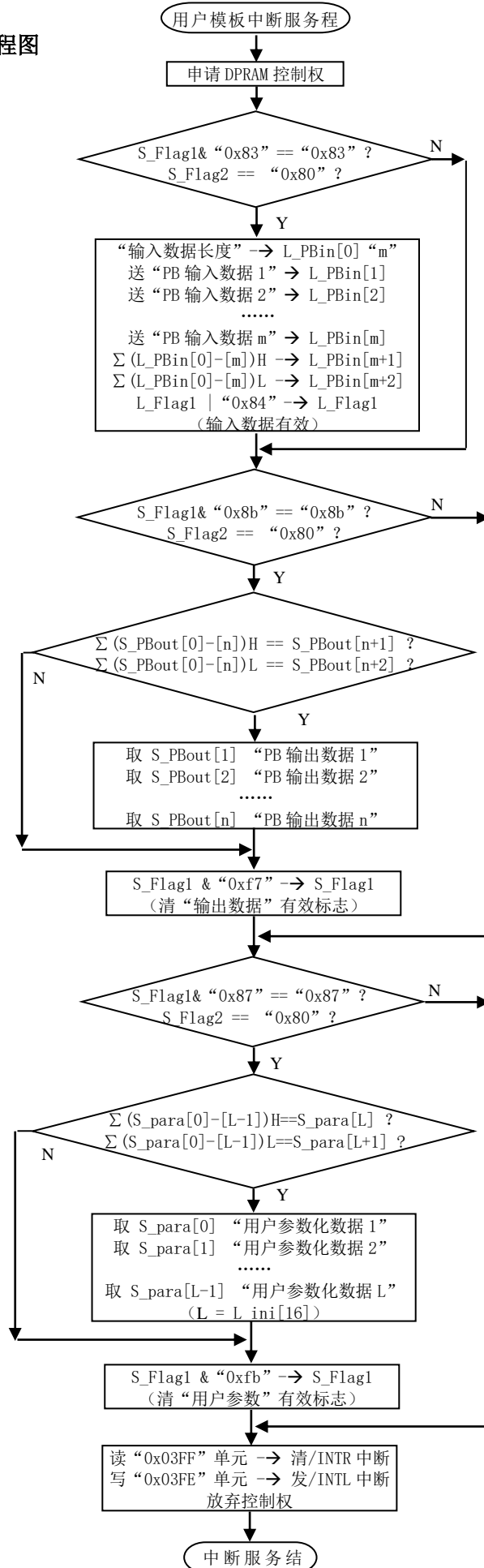


图 4-5 用户模板中断服务程序流程图

③ DPRAM 接口板中断服务程序流程图

见图 4-6（不感兴趣的用户可略过此图）

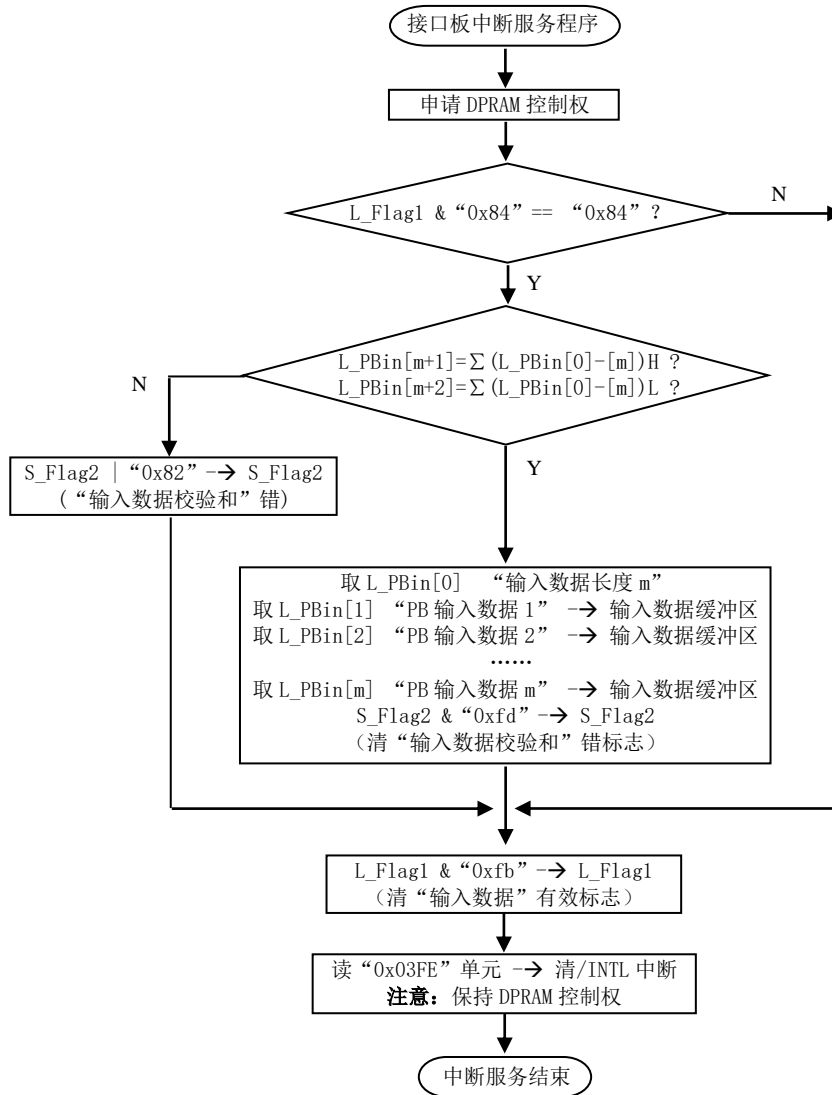


图 4-6 DPRAM 接口板中断服务程序流程图

四. 如何使用“用户参数”

1. 什么情况下需要使用“用户参数”

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；如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。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）。

2. 确定“用户参数”类型、个数、字节长度

以一个4通道模拟量输入模块为例，假设需要用户设置的参数有：

▼ 输入类型：1~5V、0~10V、0~5V、-10~+10V、4~20mA、0~20mA、0~10mA 共七种选择；占1个字节，以 INPUT=0-6 分别表示这7种输入类型；

▼ 数据类型：BCD（0000~9999）、无符号整型（0~65535）、有符号整型（-32767~+32767）共三种选择；占1个字节，以 VAR=0-2 分别表示这3种数据类型；

▼ 输入模式：单端输入（4通道）、双端输入（2通道）共二种选择；占1个字节，以 CHNO=0-1 分别表示这2种输入模式；

因此：用户参数长度 $L_{ini}[16] = 3$

3. 带有“用户参数”描述的 GSD 文件

对上例，GSD 文件中至少有如下关于“用户参数”的描述：

User_Prm_Data_Len=3

下面是一个最简单、带有“用户参数”描述的完整 GSD 文件清单：

```

=====
; GSD 文件：用户参数 GSD 文件举例说明      鼎实创新科技有限责任公司
; 产品型号： test_user_prm
; 版本   ： 2002 年 10 月 V1.0
; 文件名  ： userprm.GSD
=====
#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="test_user_prm"              ;模块名称，也是组态时该产品的名称
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"

```

; Slave-Specification:

```

OrderNumber="test_only"           ;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=3
Max_Diag_Data_Len=6
Modular_Station = 0
Modul_Offset=0
Slave_Family=3@TdF@DS/SPC3       ;组态中的分类名
User_Prm_Data_Len=3              ; 用户参数长度为3
User_prm_data=0x03,0x02,0x01     ; 给用户参数赋初值

```

; <Module-Definition-List>

```

Module="4chanes AI" 0x53          ; 4通道AI，即4 words input
EndModule

```

4. 如何在主站配置中选择用户参数

用户在主站配置中选择用户参数，完全靠 GSD 文件描述。上述 GSD 文件 userprm.GSD 是一个完整

的、带有用户参数描述的 GSD 文件。读者可根据 userprm.GSD 文件清单，使用“记事本”形成一个文本文件：参考本手册“第五章 GSD 文件”，在 STEP 7 的 HARDWARE 中选择“用户参数”：
用户参数配置窗口

见图 4-7，test_user_prm 是 userprm.gsd 文件定义的从站，双击这个从站可以见到用户参数配置窗口。

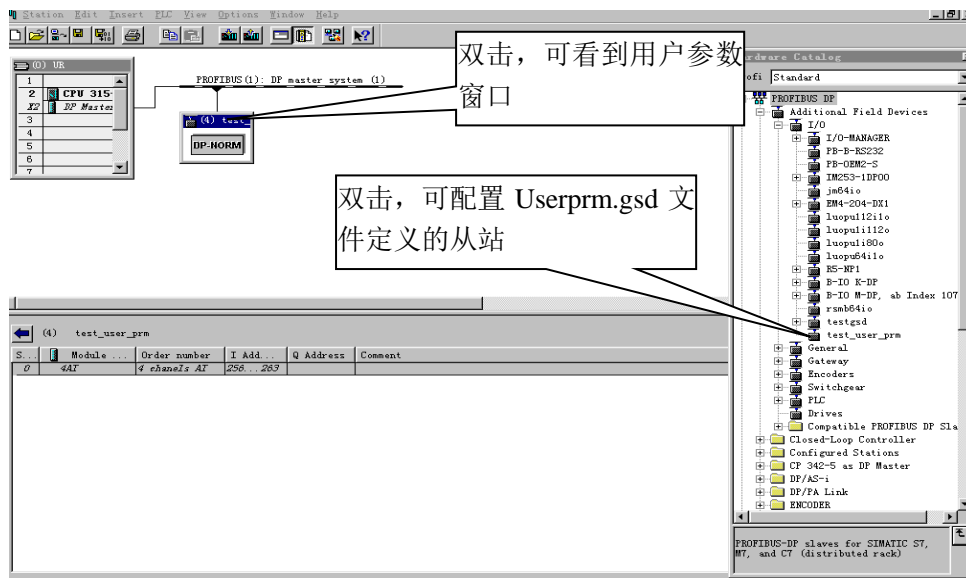


图 4-7 用户参数配置

用户参数的初值

见图 4-8，用户设定 3 个字节的用户参数，初值 3、2、1 是在 GSD 文件中定义：
User_Prm_Data=0x03,0x02,0x01。

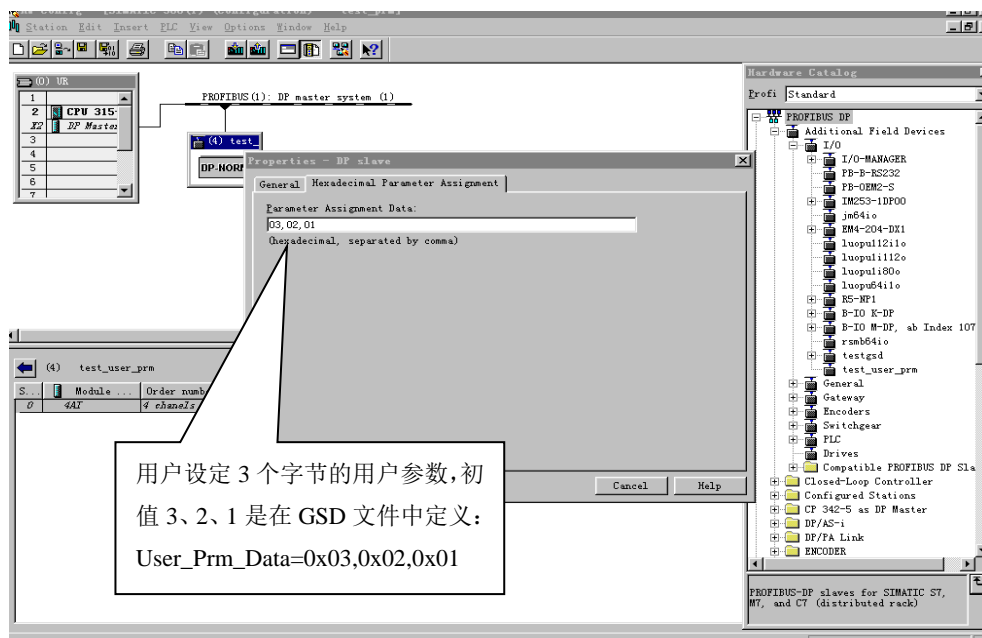


图 4-8

更友好的用户界面

u_p_1.gsd 文件，带有“用户参数”选择菜单；可在 STEP 7 中生成更加友好的用户参数配置介面，见图 4-9。

```

;=====
; GSD 文件：用户参数 GSD 文件举例说明      鼎实创新科技有限责任公司
; 产品型号： OEM1
; 版本   ： 2002 年 10 月 V1.0
; 文件名  ： u_p_1.gsd
;=====
#Profibus_DP
;<Prm-Text-Def-List>
PrmText=0
Text(0)= "1~5V"
Text(1)= "0~10V"
Text(2)= "0~5V"
Text(3)= "-10~+10V"
Text(4)= "4~20mA"
Text(5)= "0~20mA"
Text(6)= "0~10mA"
EndPrmText
PrmText=1
Text(0)= "BCD"
Text(1)= "Unsigned inf"
Text(2)= "inf"
EndPrmText
PrmText=2
Text(0)= " 单端输入（4 chanel）"
Text(1)= " 双端输入（2 chanel）"
EndPrmText

;<Ext-User-Prm-Data-Def-List>
ExtUserPrmData=0 "输入类型"
Unsigned8 3 0-6
Prm_Text_Ref=0
EndExtUserPrmData
ExtUserPrmData=1 "数据类型"
Unsigned8 2 0-2
Prm_Text_Ref=1
EndExtUserPrmData
ExtUserPrmData=2 "输入模式"
Unsigned8 1 0-1
Prm_Text_Ref=2
EndExtUserPrmData

;General parameters
GSD_Revision    = 2
Vendor_Name     = "DS FieldBus Ltd. Co."
Model_Name     = "Test User Prm"
Revision       = "V1.0"
Ident_Number   = 0x06FA
Protocol_Ident = 0

```

```
Station_Type      = 0
FMS_supp         = 0
Hardware_Release = "A1.0"
Software_Release = "Z1.0"
9.6_supp         = 1
19.2_supp        = 1
45.45_supp       = 1
93.75_supp       = 1
187.5_supp       = 1
500_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
Redundancy       = 0
Repeater_Ctrl_Sig = 2
24V_Pins         = 0
Implementation_Type = "Spc3"
Bitmap_Device    = "Bitmap1N"
Bitmap_Diag      = "Bitmap1D"
Bitmap_SF        = "Bitmap1S"

; Slave-Specification:
OrderNumber      = "test"
Freeze_Mode_supp = 1
Sync_Mode_supp   = 1
Set_Slave_Add_Supp = 0
Auto_Baud_supp   = 1
Min_Slave_Intervall = 1
Fail_Safe        = 0
Max_Diag_Data_Len = 6
Modul_Offset     = 0
Slave_Family     = 3@TdF@DS/SPC3
Modular_Station  = 0

; UserPrmData: Length and Preset:
User_Prm_Data_Len = 3
```

```

User_Prm_Data      = 0x00,0x00,0x00
Ext_User_Prm_Data_Ref(0)=0
Ext_User_Prm_Data_Ref(1)=1
Ext_User_Prm_Data_Ref(2)=2
; Module Definition List
Module="4 Channels AI" 0x53
EndModule

```

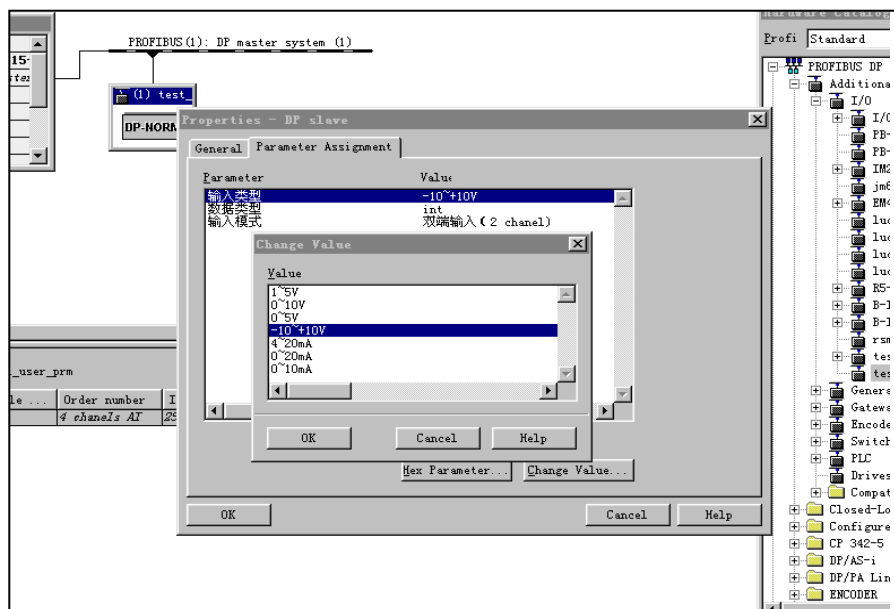


图 4-9

5. 用户模板怎样得到“用户参数”

在初始化报文中正确设定用户参数长度 (L_ini[16])

该长度的最大值为 16，无用户参数时置 0，L_ini[16]的值应与 GSD 文件中 *User_Prm_Data_Len* 的值相等。

在 GSD 文件中正确设定相关参数

根据上一节“如何在主站配置中选择用户参数”中的例子，在 GSD 文件中正确编写相关参数。

例如：

```

User_Prm_Data_Len = 3 ; 用户参数长度为3字节
User_Prm_Data     = 0x03,0x02,0x01 ; 用户参数的值分别为3、2、1

```

从接口板获取“用户参数”

在主站与从站完成连接时，接口板将接收到从主站传送来的用户参数。接口板在与用户模板进行数据交换前把用户参数存放在双口 RAM 的用户参数数据区：S_para[0]—S_para[L-1]，(L=*User_Prm_Data_Len*)，其校验和存放在 S_para[L]—S_para[L+1]，并将接口板状态字 S_Flag1.Bit2 置“1”（用户参数数据有效）。用户模板读取用户参数数据后，应将 S_Flag1.Bit2 清“0”。

校验和等于 S_para[0]—S_para[L-1]之和：S_para[L]= Σ (S_para[0]—S_para[L-1]) 的高字节、S_para[L+1]= Σ (S_para[0]—S_para[L-1]) 的低字节。

第五章 GSD文件

1. 关于 GSD 文件（Electronic Data Sheet）

- ▼ 每一个 PROFIBUS 从站或一类主站要有一个“设备描述文件”称为 GSD 文件，用来描述该 PROFIBUS-DP 设备的特性。
- ▼ GSD 文件包含了设备所有定义参数，包括：
 - 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。
- ▼ GSD 文件是文本类文件，可用“记事本”编辑。
- ▼ 无论使用什么样的系统配置软件，都要根据 GSD 文件来对设备配置。
- ▼ 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。

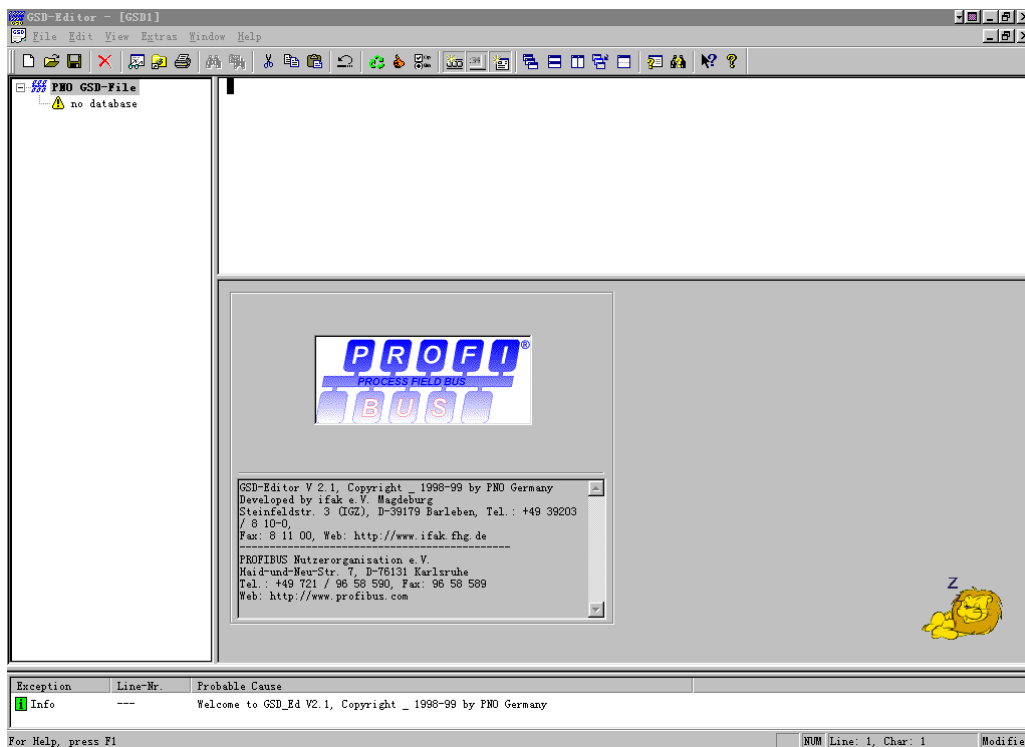


图 5-1 GSD EDIT 打开一个空文件

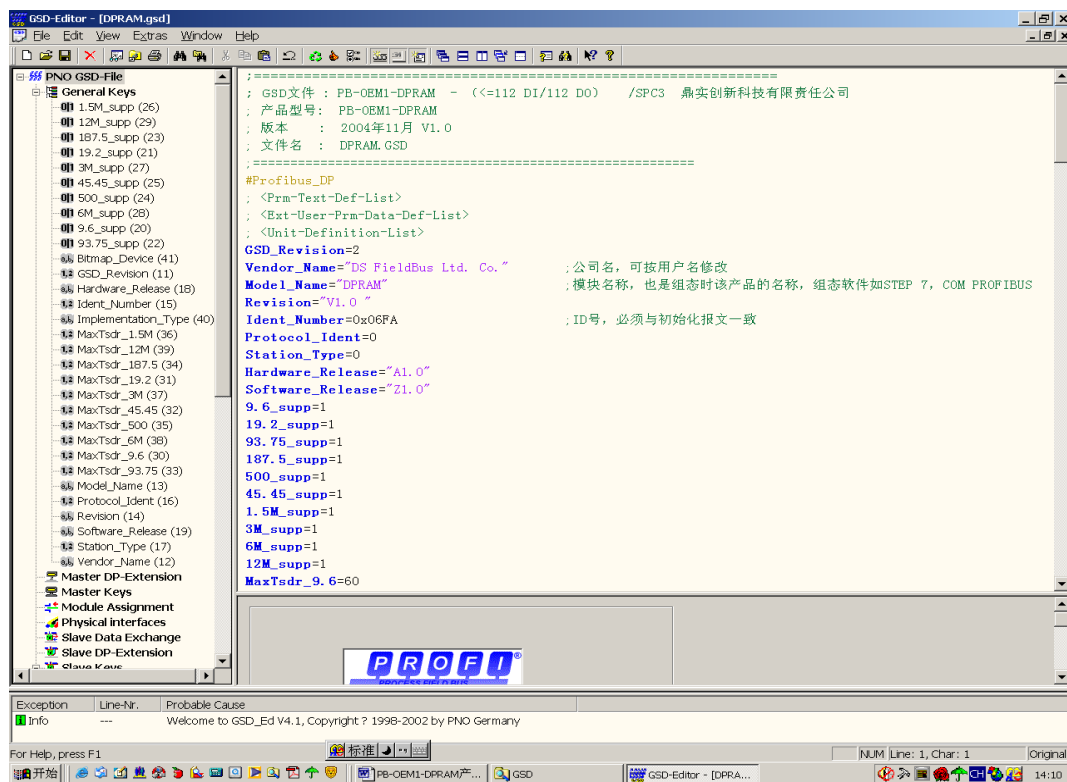


图 5-2 GSD EDIT 打开 DPRAM.GSD 文件

2. 本产品的 ID 号及 GSD 文件

PB-OEM1-DPRAM 产品的 ID 号：06FA，和 GSD 文件名：DPRAM.GSD。

3. 用户产品的 ID 号、GSD 文件及产品认证

- ▼ 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；
- ▼ 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；
- ▼ 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。
- ▼ 用户也可自定义一个 ID 号，即使在一条 PROFIBUS 总线上，不同类型、或相同类型从站的具有同一 ID 号并不影响系统连通。

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，即可成为用户产品的 GSD 文件。

4. DPRAM.GSD 说明及修改成用户 GSD 文件的说明

```

=====
; GSD文件 : PB-OEM1-DPRAM - (<=112 DI/112 DO) /SPC3 鼎实创新科技有限责任公司
; 产品型号: PB-OEM1-DPRAM
; 版本 : 2004年11月 V1.0
; 文件名 : DPRAM.GSD
=====
#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="DPRAM" ; 模块名称, 也是组态时该产品的名称。可改写为用户产品名
Revision="V1.0 " ; 可改写为用户产品版本号
Ident_Number=0x06FA ; ID号, 必须与初始化报文一致。可改写为用户产品ID号
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="ET200B_N" ; 图标文件名(可缺省), 用户可自制图标,
; Slave-Specification:
OrderNumber="PB-OEM1-DPRAM" ; 产品序列号, 可改写为用户产品订货号。
Freeze_Mode_supp=1

```

```

Sync_Mode_supp=1
Auto_Baud_supp=1
Fail_safe=0
Min_Slave_Intervall=3
Max_Diag_Data_Len=6
User_Prm_Data_Len=5 ; 用户参数长度, 如果没有用户数据可注销此句
User_prm_data=0x01,0x02,0x03,0x04,0x05 ; 用户参数初值, 如果没有用户参数可注销此句
Modular_Station = 1
Max_Module = 2 ; Max. 2 Module
Max_Input_Len = 112 ; Max. 112 Byte Input-Data
Max_Output_Len = 112 ; Max. 112 Byte Output-Data
Max_Data_Len = 224 ; Max. 224 Byte Gesamt-Data
Modul_Offset=0
Slave_Family=0@DPRAM ;组态中的分类名

;<Module-Definition-List>
Module=" 2 Byte In, 2 Byte Out " 0x11,0x21;I/O配置数据, 可添加或修改, 但必须与“初始化参数”一致
EndModule
Module=" 8 Byte In, 8 Byte Out " 0x17,0x27
EndModule
Module=" 16 Byte In, 16 Byte Out " 0x1f,0x2f
EndModule
Module=" 32 Byte In, 32 Byte Out " 0x1f,0x2f,0x1f,0x2f
EndModule
Module=" 48 Byte In, 48 Byte Out " 0x1f,0x2f,0x1f,0x2f,0x1f,0x2f
EndModule
Module=" 64 Byte In, 64 Byte Out " 0x1f,0x2f,0x1f,0x2f,0x1f,0x2f,0x1f,0x2f
EndModule
Module=" 2 Word In, 2 Word Out " 0x51,0x61
EndModule
Module=" 8 Word In, 8 Word Out " 0x57,0x67
EndModule
Module=" 16 Word In, 16 Word Out " 0x5f,0x6f
EndModule
Module=" 32 Word In, 32 Word Out " 0x5f,0x6f,0x5f,0x6f
EndModule
Module=" 48 Word In, 48 Word Out " 0x5f,0x6f,0x5f,0x6f,0x5f,0x6f
EndModule
Module=" 100 Byte In, 100 Byte Out " 0x40,0x71,0x80,0x71
EndModule
Module=" 112 Byte In, 112 Byte Out " 0x40,0x77,0x80,0x77
EndModule
Module=" Empty " 0x00
EndModule

```

▼ 注：在<Module-Definition-List>之下的I/O配置数据中，最少只要有一对 “ **Module …… EndModule** ” 即可。也可象上述那样列出数对 “ **Module …… EndModule** ”，让最终用户在主站中作硬件配置时根据需要自行选择搭配：

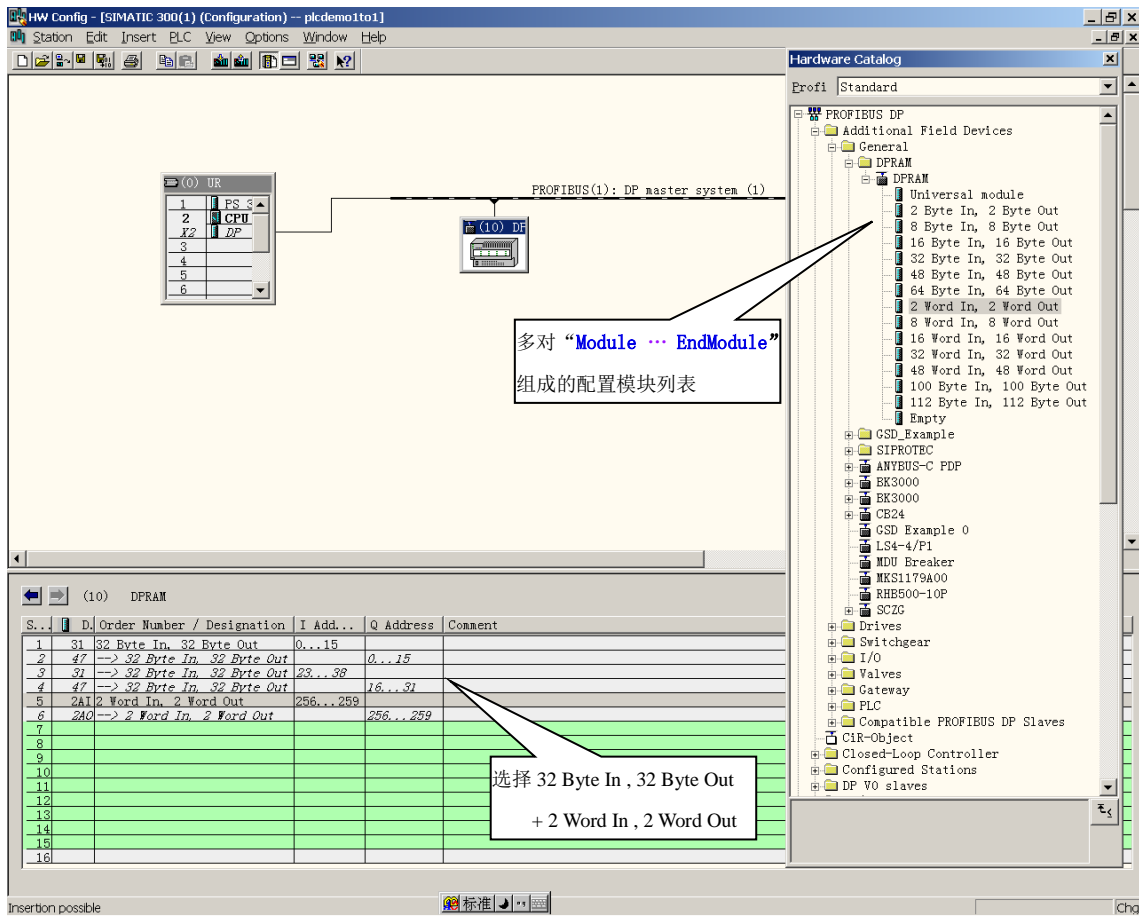


图 5-3 GSD 文件中 “Module …… EndModule” 与主站配置的关系

现场总线 PROFIBUS（中国）技术资格中心
北京鼎实创新科技有限公司

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 402~404 邮编：100120
 Web:www.c-profibus.com.cn Email: sunhm@c-profibus.com.cn