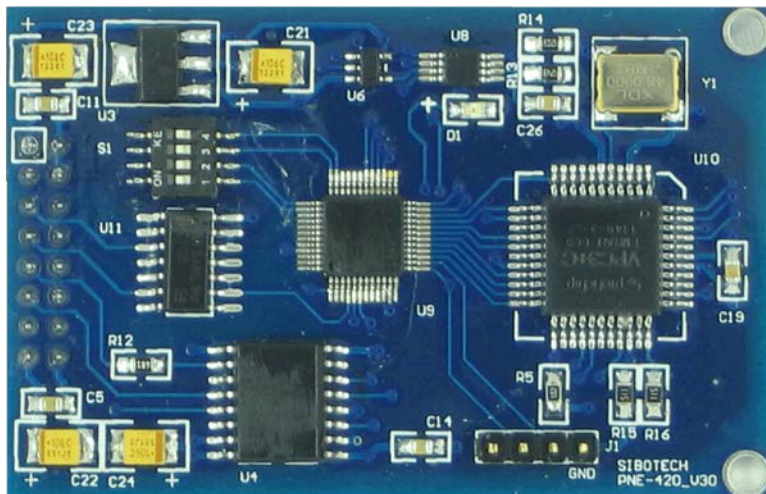


嵌入式 PROFIBUS-DP 从站接口模块

PNE-420

产品手册

REV1.8



上海泗博自动化技术有限公司

SiboTech Automation Co., Ltd.

技术支持热线: 021-5102 8348

E-mail: support@sibotech.net

目 录

一、产品概述	2
1.1 产品功能	2
1.2 产品特点	2
1.3 技术指标	2
1.4 产品开发工作流程	4
二、硬件设计说明	5
2.1 硬件原理框图及外形尺寸	5
2.2 J1-16 针接插件管脚定义及说明	6
2.3 PROFIBUS 端口定义及说明	8
2.4 拨码开关 S1 定义及说明	9
三、软件设计说明	10
3.1 串口通信	10
3.2 通信基本过程	10
3.2.1 接口卡的初始化	10
3.2.2 数据交换	22
四、如何使用用户参数	35
4.1 使用用户参数 USER_PRM 的场合	35
4.2 用户参数实现方法	35
4.3 确定“用户参数”类型、个数、取值范围	36
4.4 如何在主站配置中选择用户参数	36
五、GSD 文件	38
附录：便于用户使用的 GSD 文件	39

一、产品概述

1.1 产品功能

本产品以OEM 的方式，为希望自主开发具有PROFIBUS-DP 通信功能产品的用户，提供 PROFIBUS-DP 从站通信接口。

1.2 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；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 通过串行接口向接口卡读/写输入/输出数据，接口卡自动转换成 PROFIBUS-DP 协议与PROFIBUS 主站通信；
- 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；
- 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。

1.3 技术指标

【1】 PROFIBUS-DP/V0 协议，符合：JB/T 10308.3-2001：测量和控制数字数据通信工业控制系统用现场总线第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210字节输入+210字节输出；

【4】 PNE-420 接口卡与用户模板之间数据交换报文长度可由用户设定，最大为212字节。

【5】 可实现PROFIBUS 用户参数化功能，支持的最大配置参数为20个，最大参数字节数为56个。

【6】 接口卡与用户板接口为TTL 异步串口，使用拨码开关S1可选择不同波特率：9600、19.2K、38.4K、57.6K、115.2K、230.4K、460.8K；

【7】 接口卡与用户模板通信有字符偶校验和纵向报文字节校验和，保证数据安全性；

【8】接口卡内设有备份I/O 通信缓冲区，可保证实现PROFIBUS 与串口通信数据的完整性（一致性）和同步要求；

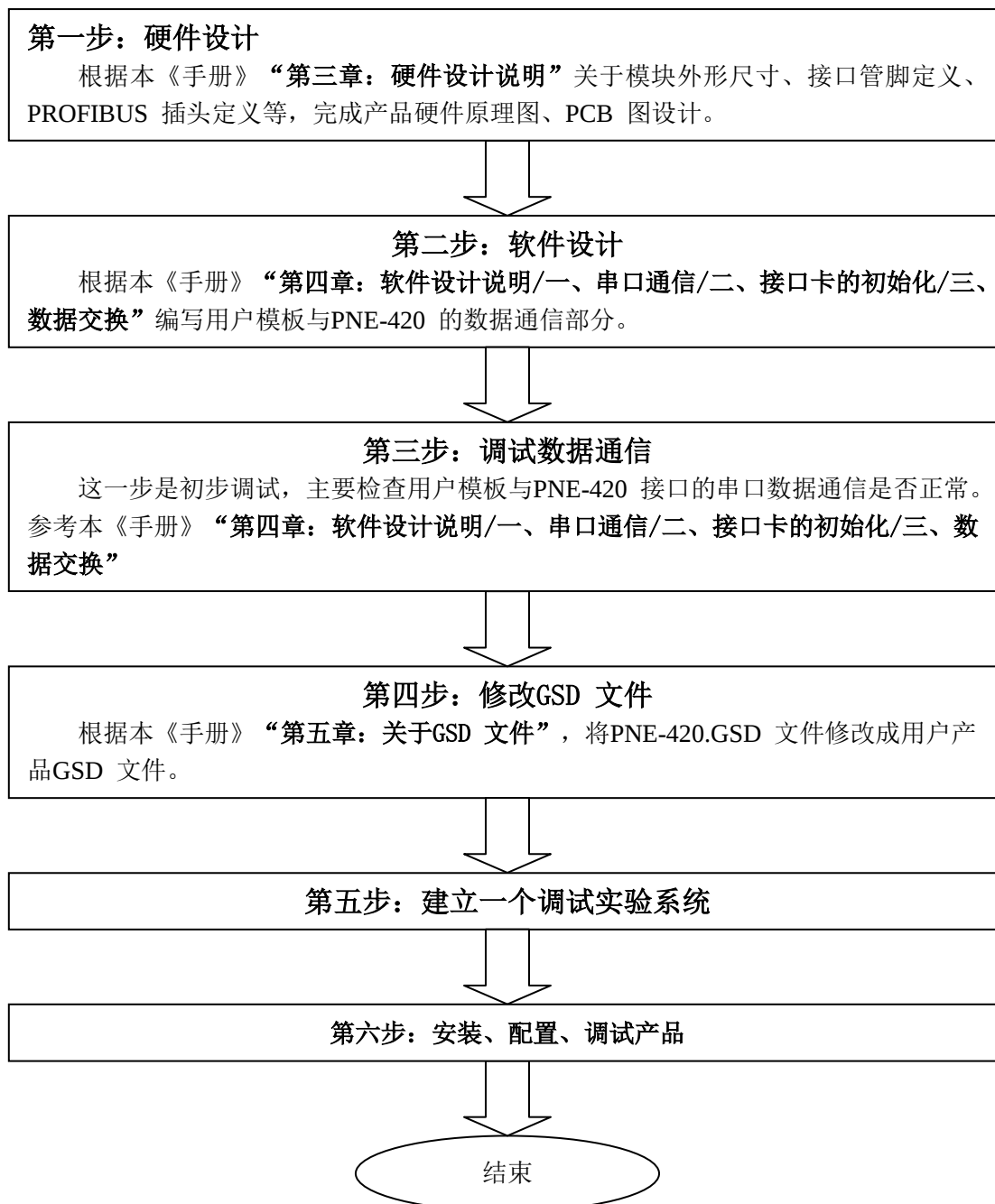
【9】需用户板供电2 组5VDC, VCC/GND-150mA 和5V/0V-100 mA；

【10】PROBUS DP和MCU之间采用最大2500V（有效值）电磁隔离；

【11】工作环境温度：-20℃ ~ 60℃，湿度95%；

【12】外形尺寸：42mm * 65mm。

1.4 产品开发工作流程



二、硬件设计说明

2.1 硬件原理框图及外形尺寸

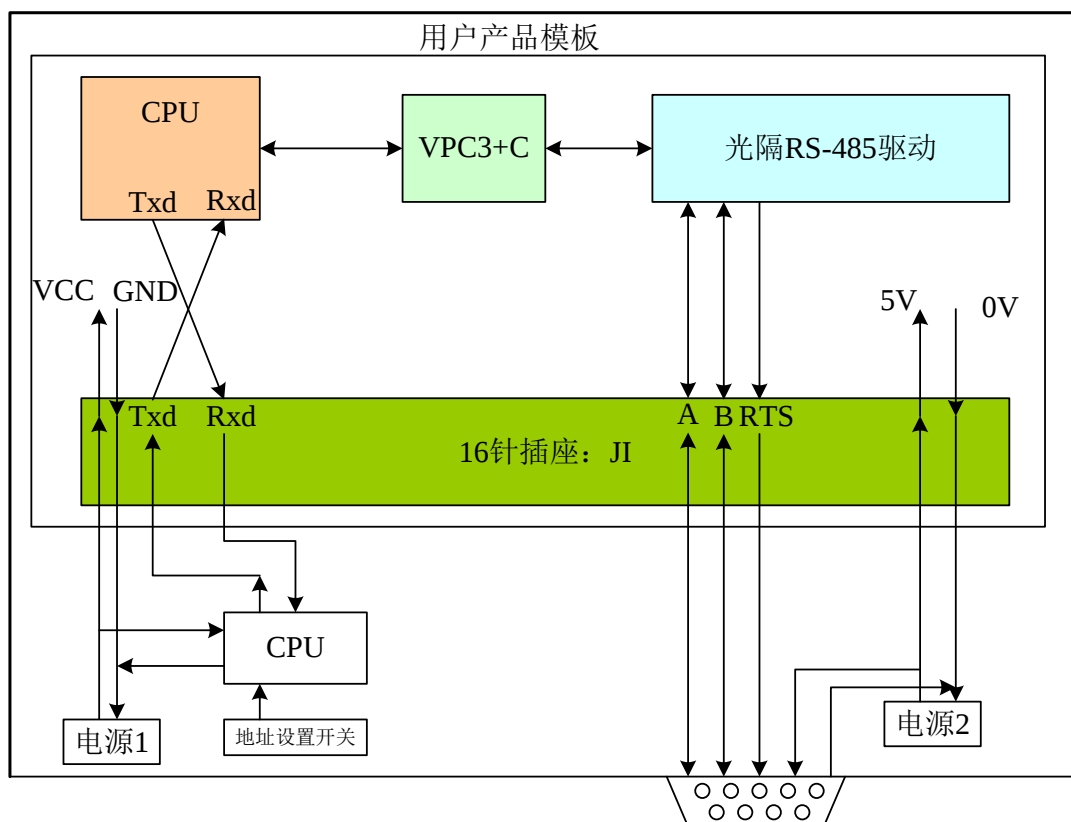


图 2-1 硬件原理框图

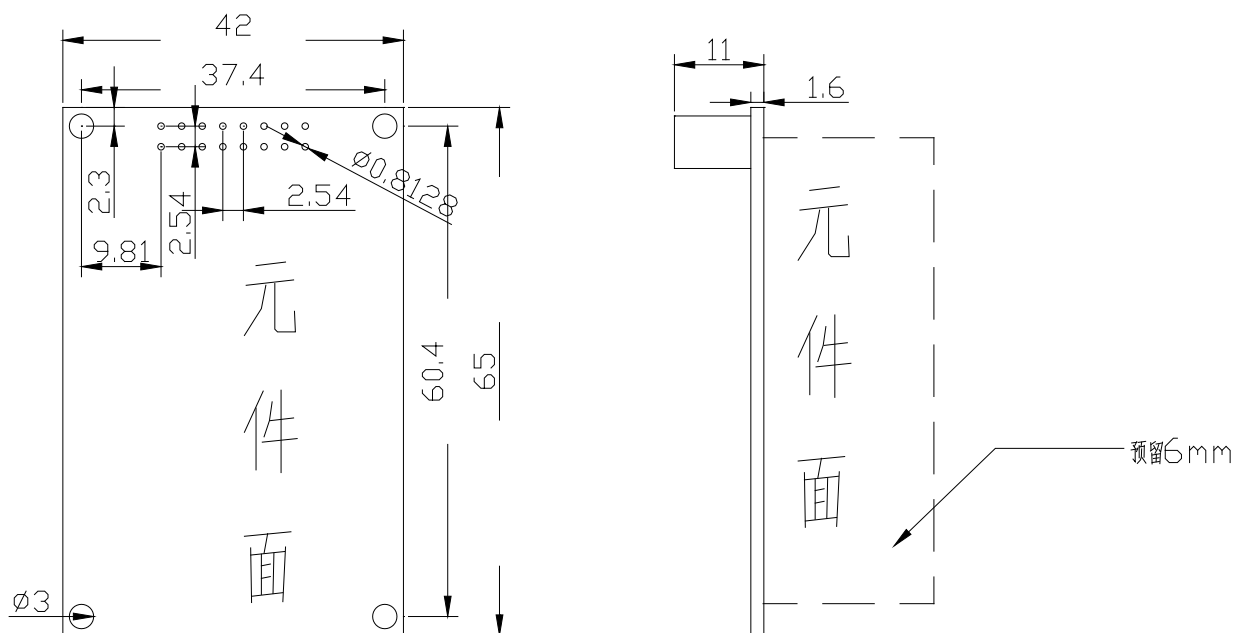
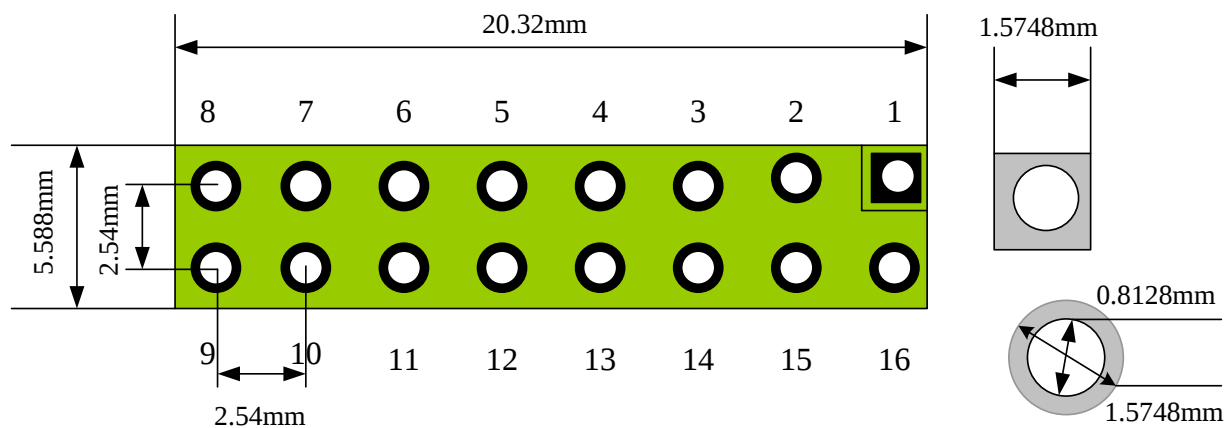


图 2-2 PNE-420 外形尺寸图

2.2 J1-16 针接插件管脚定义及说明

J1-16 管脚尺寸如下所示：



J1-16 针接插件管脚定义如下表所示：

表 2-1 16 针接插件管脚定义

管脚	I/O	说明	管脚	I/O	说明
1	I	VCC, (+5V 直流电源)	16	O	GND, (VCC 的地)
2	I	/RES (接口卡复位, 低有效, 建议加上拉)	15	O	GND, (VCC 的地)

3	O	Rxd, TTL 接用户模板CPU的Rxd	14	O	REQ_IT 初始化等待/完毕 0: 接口卡初始化完毕 1: 接口卡等待初始化
4	O	S_RTS 接收允许: 0: 接口卡等待(允许)接收; 1: 接口卡拒绝接收;	13	I	Txd, TTL 接用户模板CPU 的Txd
5	O	PB-RTS, 接用户模板D 型插头④	12	O	PBF, 外接PROFIBUS 通信指示灯; 0: 模块与主站连通, 进入数据交换状态; 1: 模块等待主站连通;
6	I/O	B, PROFIBUS 数据线, 接用户模板D 型插头③	11	I/O	A, PROFIBUS 数据线, 接用户模板D 型插头⑧
7		保留	10		保留
8	I	5V, 光隔外电源	9	O	0V, (光隔外电源地, 与GND 隔离)

状态信号 S_RTS 与 REQ_IT 反映了 PNE-420 的状态, 用户模板可以根据 PNE-420 状态实现握手通信, 如下表所示:

表 2-2 PNE-420 状态表

PNE-420 输出 REQ_IT	PNE-420 输出 S_RTS	PNE-420 状态	用户板应进行的操作
1	0	PNE-420 等待接收初始化数据;	发送初始化数据;
0	0	PNE-420 等待接收PROFIBUS 输入数据;	发送PROFIBUS 输入数据;
1	1	PNE-420 接收初始化数据完毕,正处在: →分析初始化数据 →发送初始化失败报文 →初始化PROFIBUS 接口过程中;	等待、接收PNE-420 接收初始化回答报文;
0	1	1.PNE-420 处在上电或复位后准备状态; 2.PNE-420 正在回答初始化成功报文; 3.PNE-420 正在发送PROFIBUS 输出数据;	1.上电或复位后: 等待PNE-420 请求初始化回答报文; 2.发送完毕初始化数据: 等待接收PNE-420接收初始化回答报文; 3.处在数据交换状态: 等待接收PROFIBUS 输出数据;

PNE-420 接口卡从接收初始化报文到进入数据交换状态的信号时序如下图所示:

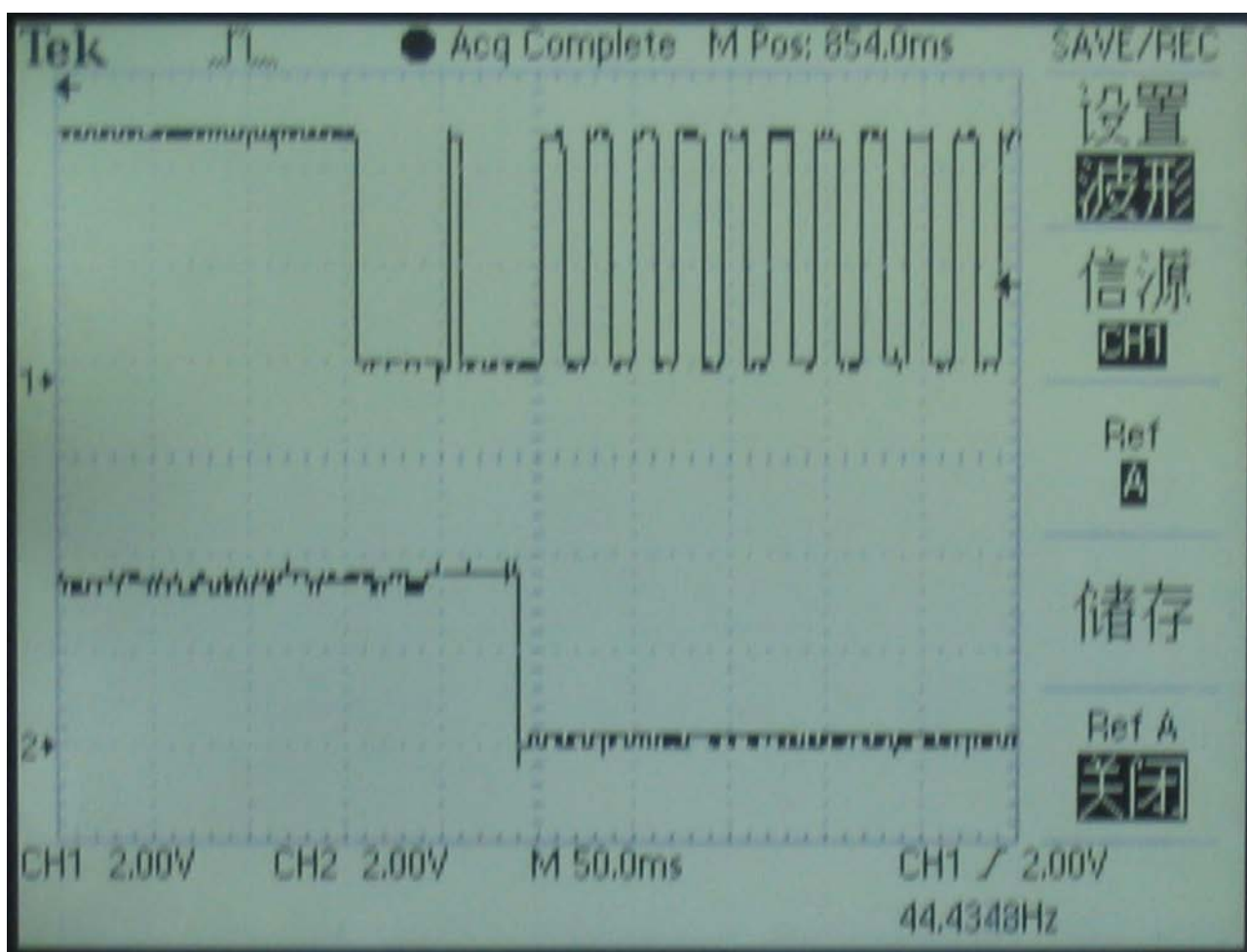


图 2-3 接口卡从初始化到数据交换状态的时序图

2.3 PROFIBUS 端口定义及说明

用户产品模板 PROFIBUS 插座通常是孔型（female）D 型插座，在系统集成中，连接 PROFIBUS 设备要求使用标准 PROFIBUS 插头与电缆。

PROFIBUS 孔（female）D 型插座引脚定义如下：

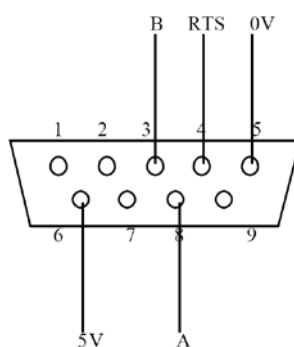
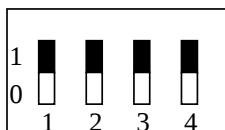


表 2-3 PROFIBUS 孔 D 型插座管脚定义

	信号	说明
1		
2		
3	B	PROFIBUS 数据线+: 对应 J1-16/6(B)
4	RTS	中继器控制信号 (方向控制): 对应 J1-16/5(PB-RTS)
5	DGND	0V: 光隔外电源地: 对应 J1-16/9
6	VP	5V: 光隔外电源 5V: 对应 J1-16/8
7		
8	A	PROFIBUS 数据线-: 对应 J1-16/11(A)
9		

2.4 拨码开关 S1 定义及说明

拨码开关 S1 如下图所示:



通过设置拨码开关 S1 的值可设置使用不同的串口波特率, 串口波特率的定义如下表:

表 2-4 拨码开关 S1 定义

bit4	Bit3	Bit2	Bit1	波特率 (bps)
1	1	1	1	9600
1	1	1	0	19200
1	1	0	1	38400
1	1	0	0	57600
1	0	1	1	115200
1	0	1	0	230400
1	0	0	1	460800
1	0	0	0	备用



三、软件设计说明

3.1 串口通信

接口卡与用户模板的数据交换：串口连接、Txd、Rxd 为 TTL 电平、半双工；

异步串口数据帧格式：每 1 个字节用 11bits 传送：**1 个起始位、8 个数据位、1 个偶校验位、1 个停止位**；

异步串口波特率的设置：

S1 设置=bit4/bit3/bit2/bit1=1001~1111：波特率 460800~9600bps 可选。

通信方式：应答方式，用户模板主动询问，接口卡被动回答；

报文长度：

接口卡的初始化报文（发送、回答）固定长度：49 字节；

数据交换报文（发送、接收）长度由用户在初始化报文中设定，最大为 210 字节。

3.2 通信基本过程

用户模板的通信基本过程包括：

1、初始化：用户模板按接口卡上 S1 设置的波特率向接口卡发初始化报文，然后等待接收接口卡的回答；接口卡回答初始化成功后，进入数据交换状态。

2、数据交换状态：用户模板主动向接口卡发送PROFIBUS输入数据、接受接口卡回答PROFIBUS 输出数据，反复进行。

3.2.1 接口卡的初始化

3.2.1.1 初始化说明

接口卡的初始化过程：

用户模板按照接口卡拨码开关 S1 设置的波特率向接口模块发送初始化报文；

接口卡接收到初始化报文后，如果“偶校验”及“校验和”无错，检查初始化报文参数，回答初始化成功/失败报文给用户模板。

用户模板发送初始化报文格式、接口卡回答的初始化成功报文及失败报文格式如下图所示：

Byte 0	站号	Byte 0	测试到的波特率号BR=0~7	Byte 0	测试到的波特率号BR=0~7
Byte 1	ID号高位	Byte 1	初始化报文错误号N=0	Byte 1	初始化报文错误号N=01~06
Byte 2	ID号低位	Byte 2	初始化成功标志=0xaa	Byte 2	初始化失败标志=0x55
Byte 3	I/O配置数据长CFG_LEN≤20	Byte 3	初始化成功标志=0xaa	Byte 3	初始化失败标志=0x55
Byte 4	I/O配置数据1	Byte 4	初始化成功标志=0xaa	Byte 4	初始化失败标志=0x55
Byte 5	I/O配置数据2	Byte 5	初始化成功标志=0xaa	Byte 5	初始化失败标志=0x55
Byte 6	I/O配置数据3				
Byte 7	I/O配置数据				
Byte 23					
Byte 24	接收数据长SD_in_len≤210				
Byte 25	发送数据长SD_out_len≤210				
Byte 26	用户参数长度 User_Prm_Data_Len≤56				
Byte 27	...不用				
Byte 28	...不用				
	...不用				
Byte 47		Byte 47		Byte 47	
Byte 48	校验和 $\sum (0) \sim (47)$	Byte 48	校验和 $\sum (0) \sim (47)$	Byte 48	校验和 $\sum (0) \sim (47)$

用户模板发送初始化报文格式

接口板回答的初始化成功报文格式

接口板回答的初始化失败报文格式

图 3-1 初始化报文及回答报文一般格式

初始化报文格式：

1) 站号：

用户模板应设有设置地址的拨码开关、或可以有键盘、手持终端等其他方式设置 PROFIBUS 从站地址，从站地址范围：0~126；

用户模板设置的地址必需和主站配置时设置的地址一致：如使用STEP 7，在HARDWARD 配置中，用户模板配置的地址要和用户模板上拨码开关或键盘或手持终端设置的地址一致。

2) ID 号

www.sibotech.net

SiboTech®

经过PI认证的产品都有唯一的一个ID号。该产品的ID号可以使用泗博公司的ID号：**0x0dc9**，也可使用自己的ID。GSD文件中的ID号必须与初始化的设置一致。

3) I/O配置数据长度CFG_LEN和I/O配置数据

本产品PROFIBUS的I/O数量可任意设定；

I/O配置数据请见下表中的“代码”。

表3-1 PNE-420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	12 byte output, Byte 完整	0xAB	12 byte 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	12 byte input/output, Byte 完整	0xBB	12 byte 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: 8个字节输入+8个字节输出

查表3-1, 8个字节输入代码为: 0x17 (8 byte input);

8个字节输出代码为: 0x27 (8 byte output)

因此: I/O 配置数据长度CFG_LEN=2; I/O 配置数据为: 0x17, 0x27

例2: 用户产品需要:

I/O	I/O统计
设备状态:上传8个开关量DI	1个字节输入
设备采集温度、压力等共16个模拟量, 每个模拟量占2个字节;	16个字输入
设备控制: 共计8个开关输出DO	1个字节输出
设备控制: 共计16个模拟量输出, 每个模拟量占2个字节	16个字输出

查表3-1, 1个字节输入代码为: 0x10

16个字输入代码为: 0x5f

1个字节输出代码为: 0x20

16个字输出代码为: 0x6f

因此: I/O 配置数据长度CFG_LEN=4; I/O 配置数据为: 0x10, 0x5f, 0x20, 0x6f

4) Byte24 接收数据长 SD_in_len

在数据交换状态下, 接口卡接收用户模板发送数据报文长度称SD_in_len; 它包括了PROFIBUS 输入数据PROFIBUS_input、请求数据命令req_com、一个字节的校验和。因此: **接收数据长SD_in_len = PROFIBUS 输入数据字节数PB_IN + 1 (请求数据命令req_com) + 1(报文校验和); 最大SD_in_len = 212 字节;**

5) Byte25 发送数据长 SD_out_len:

在数据交换状态下, 接口卡向用户模板发送数据报文长度称SD_out_len; 它包括了PROFIBUS 输出数据PROFIBUS_output、用户参数化标记 (user_prm_ok)、用户参数(user_prm)、一个字节的报文校验和, 因此: **发送数据长SD_out_len = PROFIBUS 输出数据字节数 (PB_OUT) + 1 (报文类型及接口状态字) + 1(报文校验和); 最大SD_out_len = 212 字节;**

例3: 例1 中: 用户产品需要: 20个字节输入+20个字节输出

SD_in_len=20+1+1=22

SD_out_len=20+1+1=22

6) Byte26 用户参数长度 User_Prm_Data_Len

设定用户参数长度，如果不使用用户参数，必须使User_Prm_Data_Len=0。

7) Byte27~Byte47 不使用

可以是任意数据，但这些数据也包含在Byte48校验和中。

8) Byte48 校验和 $\Sigma(\text{byte0} \sim \text{byte47})$

校验和 $=\Sigma(\text{byte0} \sim \text{byte47})$ ；字节求和，进位丢掉，校验和仍为一个字节。

初始化成功/失败报文格式：

1) 返回波特率号BR=1~8

接口卡有回答报文，无论是“成功”还是“失败”报文，都说明接口卡已经检测到用户模板的串口波特率。

本字节是接口卡返回的检测到用户模板串口波特率编号：1=9.6K、2=19.2K、3=38.4K、4=57.6K、5=115.2K、6=230.4K、7=406.8K、8=1M

2) 初始化报文成功/错误号N=0~06

表3-2 接口卡返回的初始化报文成功/错误号

初始化报文错误号N	类型说明
N=0	没有错误，初始化成功
N=01	I/O配置数据格式错，使用了非法I/O配置数据，请参照表3-1中的“代码”
N=02	I/O配置数据个数>20错，I/O配置数据个数最大=20
N=03	站地址>126错，站地址最大为126
N=04	接收报文字符校验和错，即：校验和 $\Sigma(\text{byte0} \sim \text{byte7}) \neq \text{byte48}$
N=05	接收报文字符中有偶校验错
N=06	接收报文字符字节个数<49

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4： 站号设置为0x40，ID 号为0187；

I/O 配置：20个字节输入+20个字节输出；

I/O 配置数据长度CFG_LEN=4；

I/O 配置数据为： 0x1f, 0x13, 0x2f, 0x23；

用户参数长度User_Prm_Data_Len=0；

接收数据长SD_in_len $\geq$ 20+1+1=22;

发送数据长SD_out_len $\geq$ 20+1+1=22;

则: GSD 文件中必须有:

User_Prm_Data_Len=0

Module="20bytes in, 20bytes out" 0x1f, 0x13, 0x2f, 0x23

EndModule

初始化报文及可能回答见下图:

Byte 0	站号=0x40	Byte 0	测试到的波特率号BR=0~7	Byte 0	测试到的波特率号BR=0~7
Byte 1	ID号高位=0x01	Byte 1	初始化报文错误号N=0	Byte 1	初始化报文错误号N=01~06
Byte 2	ID号低位=0x87	Byte 2	初始化成功标志=0xaa	Byte 2	初始化失败标志=0x55
Byte 3	CFG_LEN=0x04	Byte 3	初始化成功标志=0xaa	Byte 3	初始化失败标志=0x55
Byte 4	I/O配置数据1=0x1f	Byte 4	初始化成功标志=0xaa	Byte 4	初始化失败标志=0x55
Byte 5	I/O配置数据2=0x13	Byte 5	初始化成功标志=0xaa	Byte 5	初始化失败标志=0x55
Byte 6	I/O配置数据3=0x2f				
Byte 7	I/O配置数据4=0x23				
	不用=0				
Byte 23					
Byte 24	接收数据长SD_in_len=22				
Byte 25	发送数据长SD_out_len=22				
Byte 26	用户参数长度 User_Prm_Data_Len=0				
Byte 27	...不用				
Byte 47		Byte 47		Byte 47	
Byte 48	校验和 $\sum(0) \sim (47)$	Byte 48	校验和 $\sum(0) \sim (47)$	Byte 48	校验和 $\sum(0) \sim (47)$

用户模板发送初
始化报文格式

接口板回答的初始
化成功报文格式

接口板回答的初始
化失败报文格式

图 3-2 例 4 的初始化及回答报文格式

例5: 站号设置为0x32, ID 号=0187;

I/O 配置: 1个字节输入+12个字输入=25个字节输入;

1个字节输出+2个字输出=5个字节输出;

I/O配置数据长度CFG_LEN=4;

I/O配置数据为: 0x10, 0x5B, 0x20, 0x61;

用户参数长度User_Prm_Data_Len=10;

接收数据长SD_in_len $\geq$ 25+1+1=27;

发送数据长SD_out_len $\geq$ 5+1+1=7;

GSD 文件中必须有:

User_Prm_Data_Len=10

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

EndModule

初始化报文及可能回答见下图:

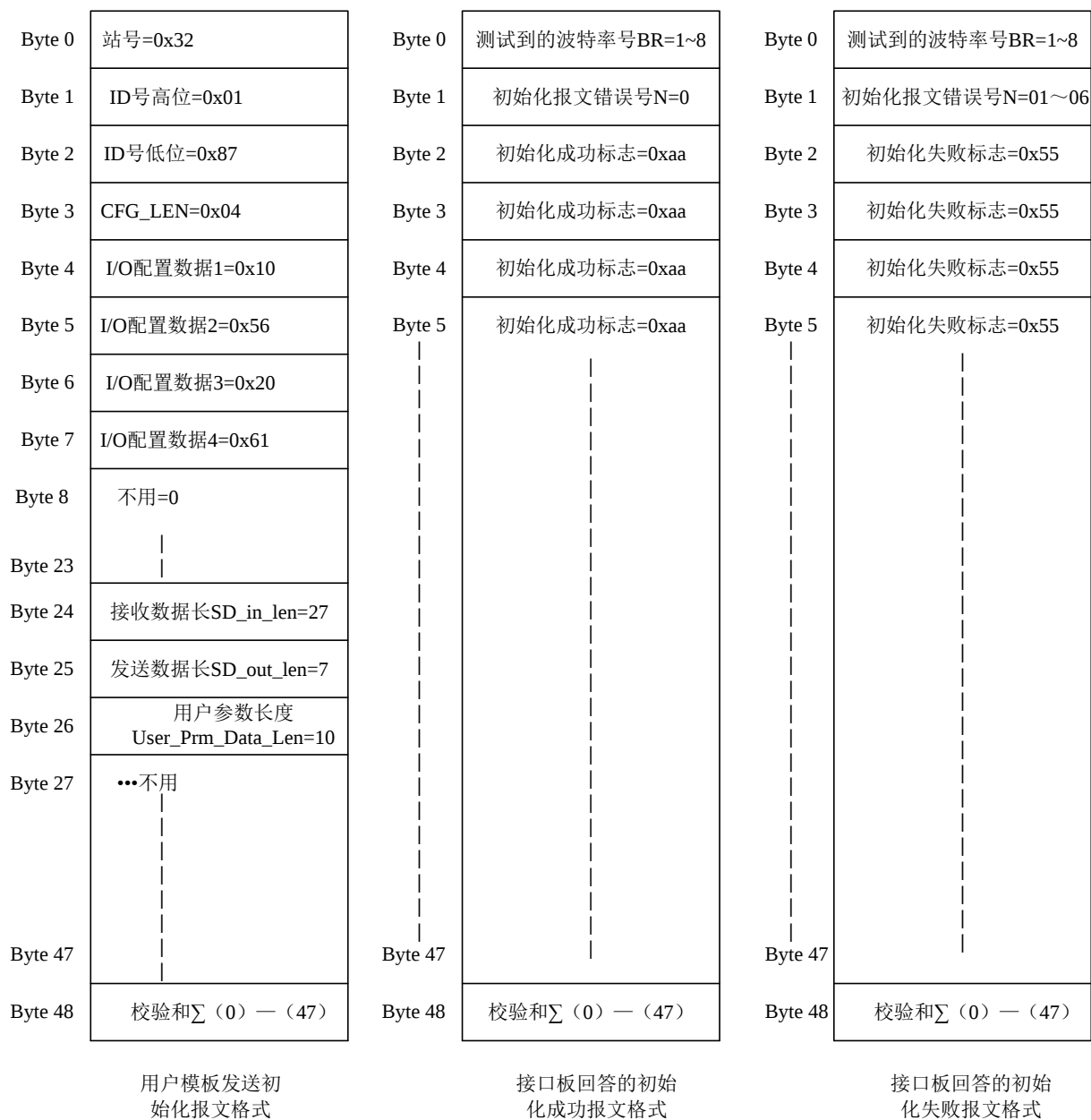


图3-3 例5的初始化及回答报文格式

3.2.1.2 用户模板初始化接口模块的过程

用户模板初始化接口模块的过程如下图所示：

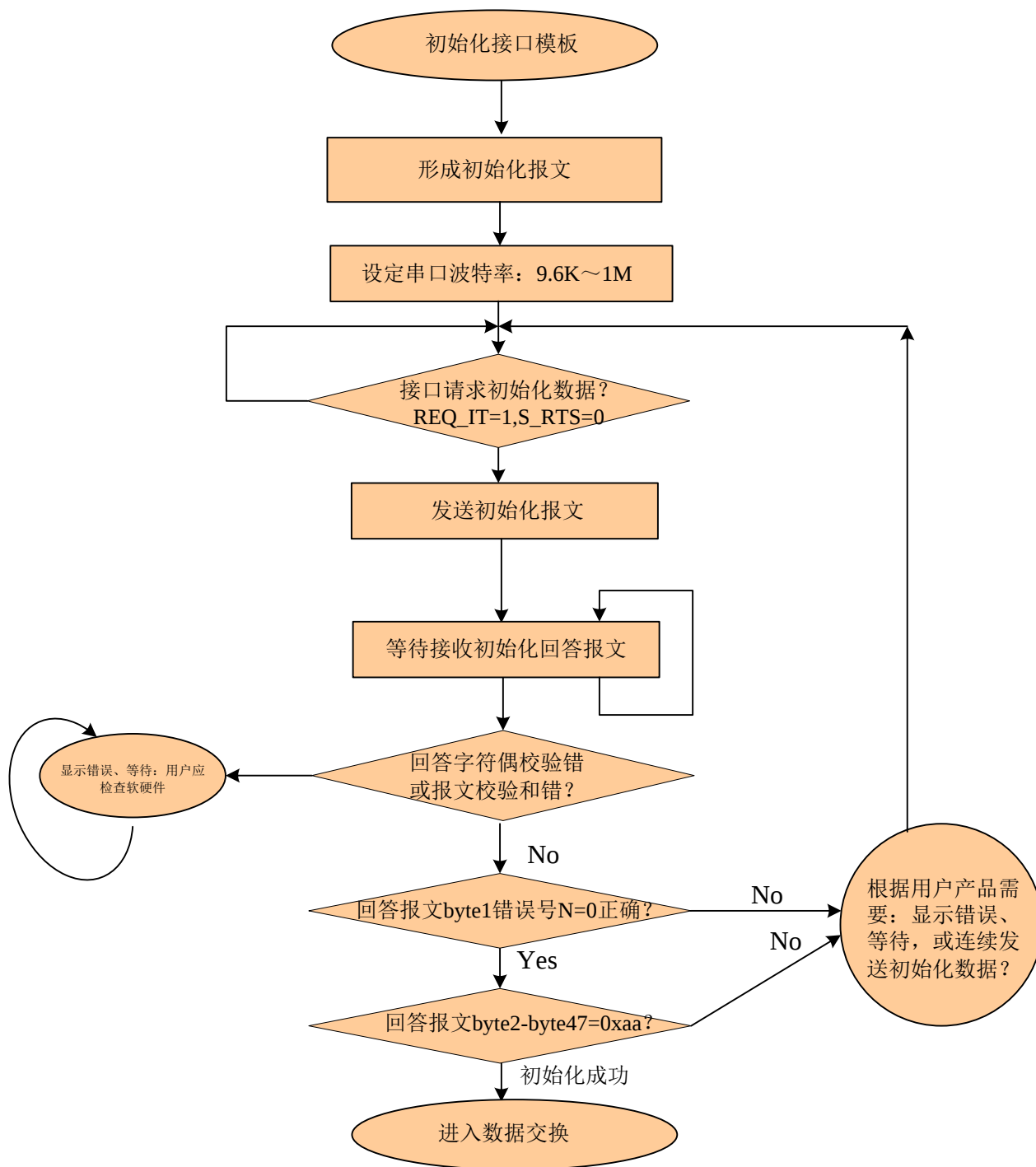


图 3-4 用户模板初始化接口卡的过程

3.2.1.3 接口卡接受初始化过程

按照S1设定的串口波特率等待接收初始化报文→检查初始化报文的正确性→向用户模板发“初始化成功”或“失败”报文→（初始化成功）进入“数据交换状态”或（初始化失败）重新等待用户模板的初始化报文。见下图：接口卡初始化程序流程图。

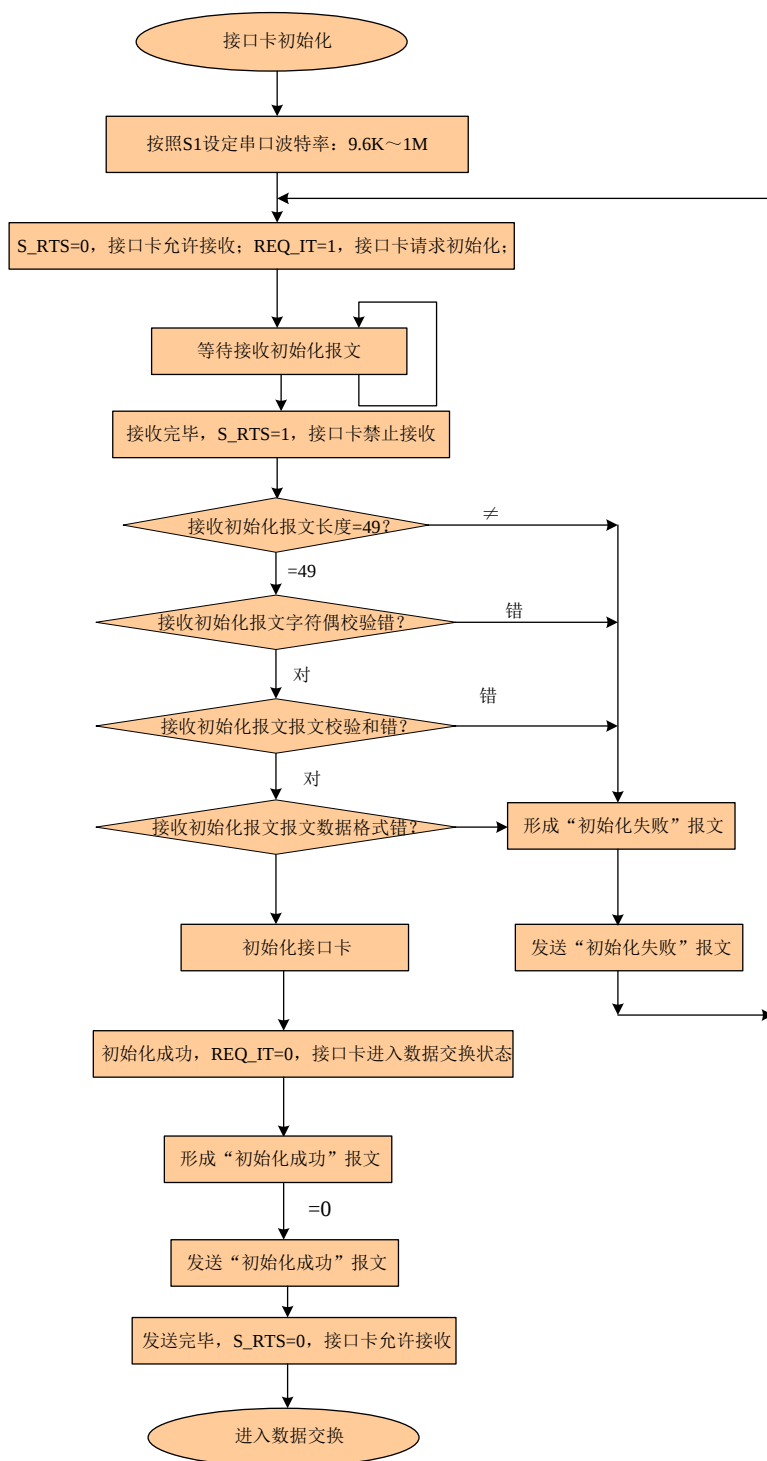


图3-5 接口卡初始化程序流程图

3.2.2 数据交换

3.2.2.1 数据交换说明

注意：本章节中的“输入输出”是指以PROFIBUS 主站为基点。

a) 数据交换状态与数据交换

初始化完成后，接口卡进入数据交换状态。此时，用户模板主动向接口卡发送“输入数据报文”，其中包含PROFIBUS 输入数据；接口卡回答“输出数据报文”，其中包含PROFIBUS 输出数据。如下图：

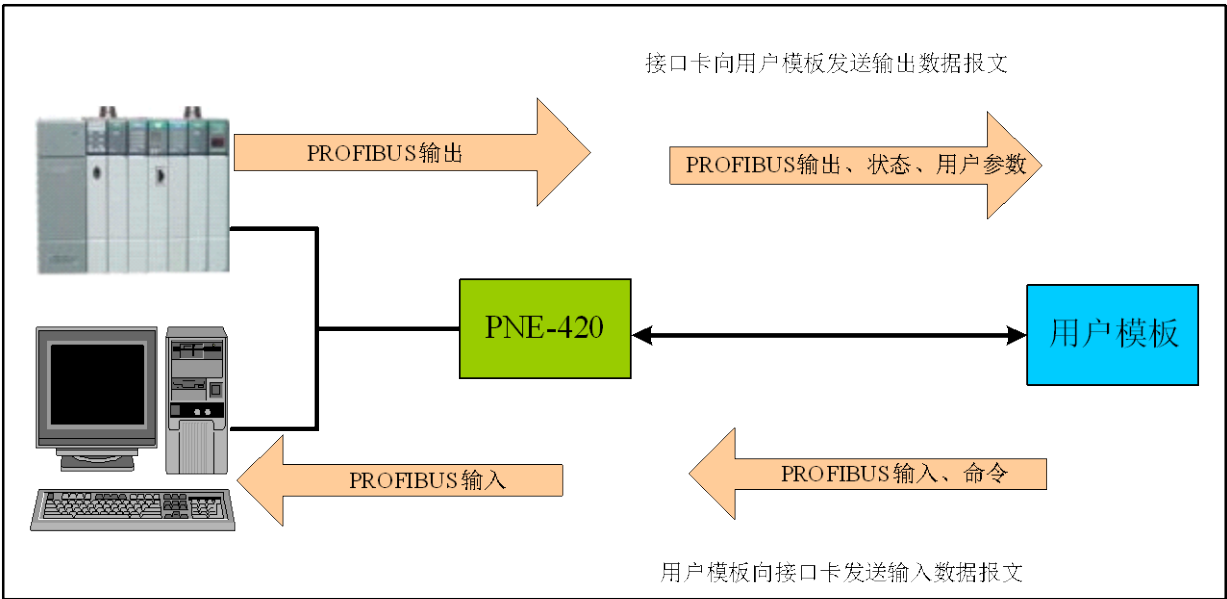


图3-6 数据交换状态下数据报文的传送

(1) 用户模板主动向接口卡发送“输入数据报文”格式如图3-7，其中包括：PROFIBUS 输入数据、请求数据命令req_com及校验和。请求数据命令req_com 见表3-3。

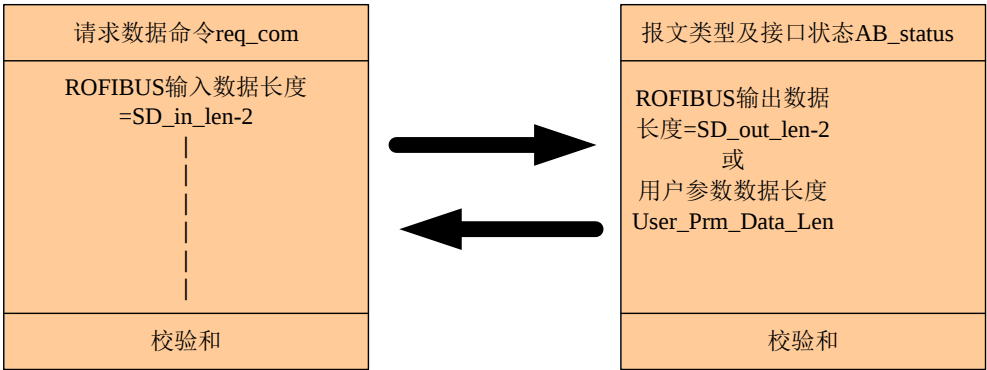


图3-7 用户模板发输入数据报文及接口卡答输出数据报文格式

注意：接口卡回答的用户参数数据和PROFIBUS 输出数据报文长度不一样。

表3-3 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, 清接口卡用户参数已由主站更新标记AB_status (D0)
D0	=0, 请求接口卡回答PROFIBUS输出数据
	=1, 请求接口卡回答PROFIBUS主站更新的用户参数数据

(2) 接口卡接收到用户板发送的输入数据报文后, 将PROFIBUS 输入数据送PROFIBUS 输入区, 然后判断“请求数据命令req_com”。

- ① 如果用户模板请求读取PROFIBUS 输出数据, 即req_com =0, 接口卡将回答“输出数据报文A”(包括: 报文类型及接口状态AB_status、PROFIBUS 输出数据、校验和), 见图3-8。报文类型及接口状态

AB_status 见表3-4:

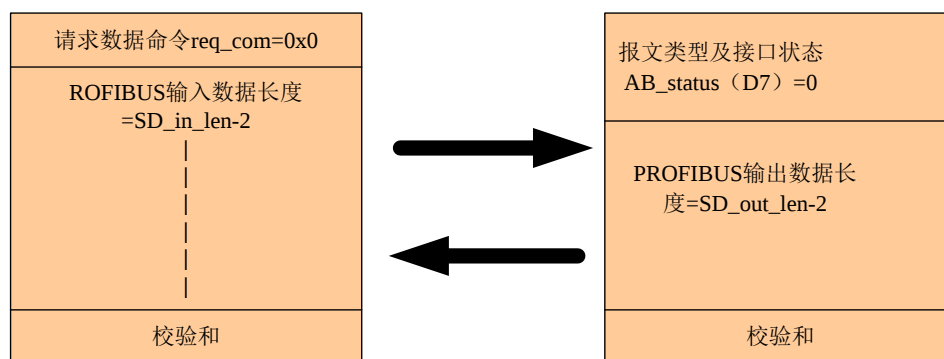


图3-8 用户模板发输入数据报文req_com =0,接口卡答输出数据报文格式

表3-4 报文类型及接口状态AB_status

报文类型及接口状态AB_status	说明
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: 接口卡接收用户板报文无错
	=01: 接口卡接收用户板报文字符有偶校验错
	=10: 接口卡接收用户板报文有校验和错
	=11: 不用
D0	=0: 接口卡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: 接口卡用户参数数据已由主站更新

② 如果req_com = 0x01，即用户模板请求读取更新的用户参数数据，接口卡将回答“输出数据报文B”

（包括：报文类型及接口状态AB_status、用户参数、校验和）。如图3-9：

注意：输出数据报文格式B的长度可能不同于报文A。

格式B 长度=1（报文类型及接口状态AB_status）+用户参数长度User_Prm_Data_Len+1（校验和）

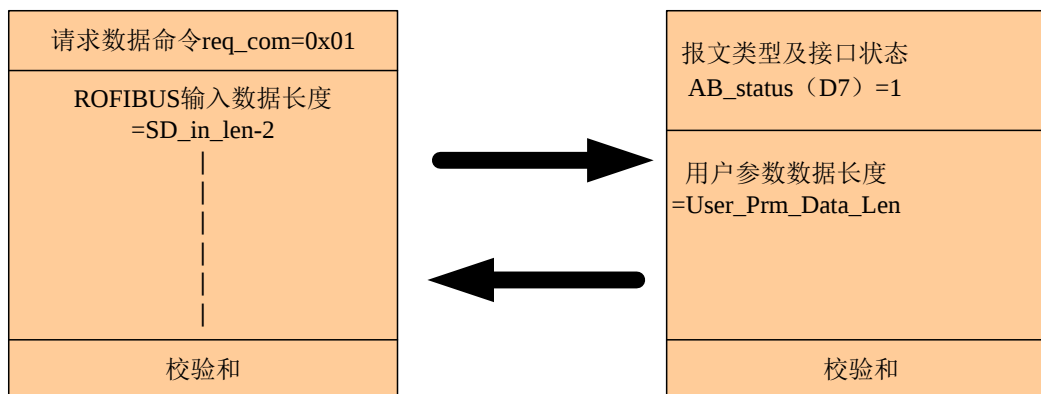


图3-9 用户模板发输入数据报文req_com = 0x01，接口卡答输出数据报文格式

b) 用户板什么时候发请求更新的用户参数命令req_com=0x01

(1) 用户模板不使用用户参数功能（即：初始化报文中User_Prm_Data_Len=0）时，永远发请求PROFIBUS数据命令req_com=0；接口卡一定回答“输出数据报文A”（包括：报文类型及接口状态AB_status、PROFIBUS 输出数据、校验和）。见图3-8，报文类型及接口状态AB_status 见表3-4。

(2) 用户模板使用用户参数功能时，每当用户模板接收到接口输出数据报文，应判断报文类型及接口状态AB_status 第D0 位，见表3-4：报文类型及接口状态AB_status (D0)，特抄录如下：

AB_status (D0) =0	接口卡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
AB_status (D0) =1	接口卡用户参数数据已由主站更新

① 如果AB_status (D0)=0，见图3-10-1：表明接口卡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；因此，用户模板应在发送输入数据报文中置“请求数据命令req_com=0”，即：不请求读用户参数，见图3-10-2；接口卡将回答输出数据报文格式A，即回答PROFIBUS 输出数据，见图3-10-3。

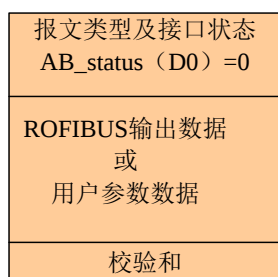


图3-10-1 用户板接收到输出数据中AB_status (D0) =0

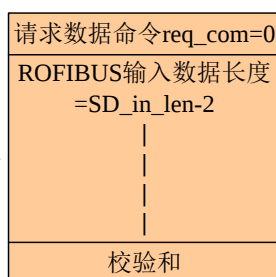


图3-10-2 用户板发输入数据中req_com=0

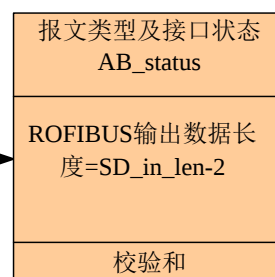


图3-10-3 接口卡回答输出数据报文格式A

② 如果AB_status (D0)=1, 见图3-11-1: 表明接口卡用户参数数据已由主站更新; 因此, 用户模板应在发送输入数据报文中置“请求数据命令req_com =0x01”, 见图3-11-2, 即: 请求读用户参数; 接口卡将回答输出数据报文格式B, 即回答用户参数数据。见图3-11-3。如果用户板正确地接收了接口卡回答的输出数据报文格式B, 应发请求数据命令req_com=0x02 (见表3-3: 请求数据命令req_com), 清除用户参数已更新标记AB_status (D0), 否则, AB_status (D0)将保持为1。见图3-11-4、图3-11-5。

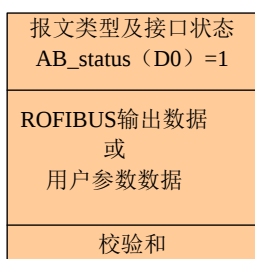


图3-11-1 用户板接收到输出数据中user_prm_ok=1

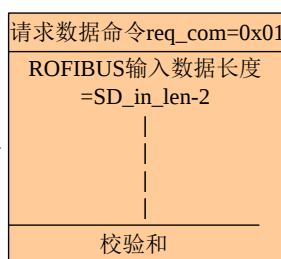


图3-11-2 用户板发输入数据中req_com=0x01

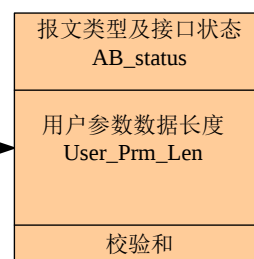


图3-11-3 接口卡回答输出数据报文格式B

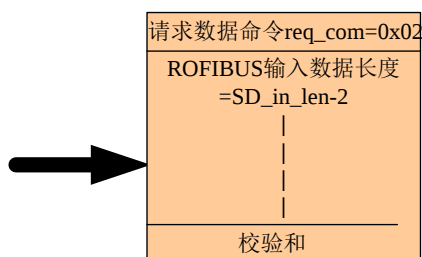


图3-11-4 用户板发输入数据中req_com=0x02

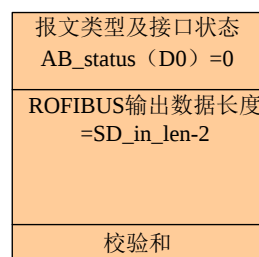


图3-11-5 接口卡回答输出数据报文格式A

说明: 图3-11-4 正确地接受接收了接口卡回答的输出数据报文格式B, 应发请求数据命令req_com=0x02, 清除用户参数已更新标记AB_status(D0)

图3-11-5 接口卡回答输出数据报文格式A, 其中用户参数已更新标记 AB_status(D0)已被清除

c) 数据交换状态下对错误报文的处理

www.sibotech.net

SiboTech®

(1) 接口卡接收数据出错的处理

如果接口卡接收数据有错（包括：偶校验错、或接收长度错、校验和错），此时，接口卡废弃接收到的（用户板发送的）PROFIBUS 输入数据，不上传PROFIBUS 主站。并且不论收到的req_com=0 还是req_com=1；只按输出数据报文格式A 回答用户模板。同时接口卡根据接收数据错类型置“报文类型及接口状态”AB_status(D2D1)≠00，通过回答输出数据报文格式A 通知用户模板。见下表3-5：“报文类型及接口状态”AB_status(D2D1)。

表3-5 “报文类型及接口状态”AB_status (D2D1)

AB_status (D2D1)	=0: 接口卡接收用户板报文无错
	=01: 接口卡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

(2) 用户模板接收数据出错处理

用户模板接收到输出数据报文后，除了进行字符偶校验、报文校验和、接收长度之外，还应判断“报文类型及接口状态”AB_status(D7)；判断报文类型与期待的报文类型是否一致。（接口卡接收正确状态下，用户模板发req_com=0，接口卡回答A 格式AB_status(D7)=0；

① 如果字符偶校验、报文校验和、期待的报文类型一致，则此次发→收正确，用户模板继续下一步发→收过程。

② 如果用户模板接收到输出数据报文任何一项有错，包括：字符偶校验、报文校验和、接收长度错、期待的报文类型不一致，用户板也应认为此次数据不可靠、不用，根据产品功能处理。用户模板继续下一步发→收过程。

分析一种特例：

如果用户模板发送带“请求数据命令req_com =0x01”的输入数据报文，正常情况下接口卡应回答输出数据报文格式B（即回答用户参数数据）；但是，如果接口卡接收输入数据报文时出错，接口卡不论收到的req_com =0x01 还是req_com =0x00；只按输出数据报文格式A 回答用户模板。

此时，用户模板按照报文格式B长度接收接口卡回答的输出数据报文，因此至少会出现：①由于报文格式B和报文格式A的长度是不一样而造成的接收长度错；②期待的报文类型不一致错；此时，用户板仍然可以按照上文（2）-②处理，即：“如果用户模板接收到输出数据报文任何一项有错，包括：字符偶校验、报文校验和、接收长度错、期待的报文类型不一致，用户板也应认为此次数据不可靠、不用，根据产品功

能处理。用户模板继续下一步发→收过程。”

3.2.2.2 用户模板与接口模块的通信流程

1、没有用户参数、不考虑接口状态条件的简单工作模式

这种工作模式为简单工作模式，如下图所示：

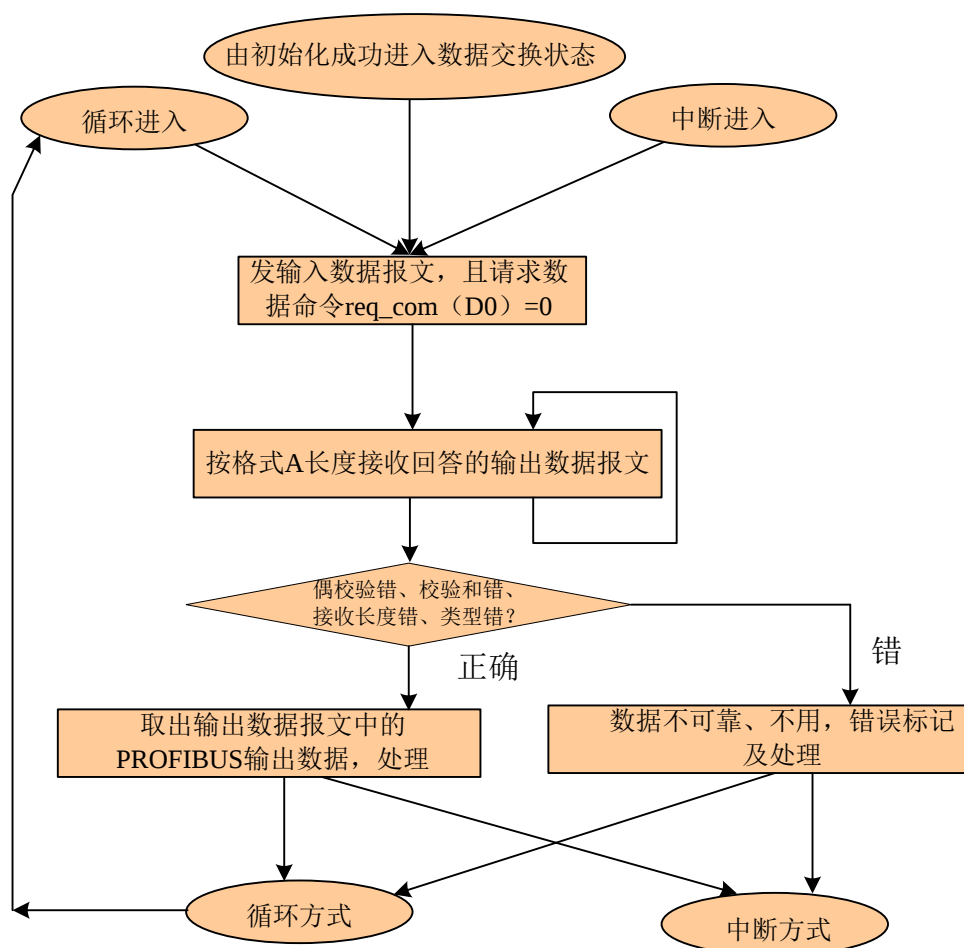


图3-12 数据状态下用户模板的程序流程（没有用户参数、不考虑接口状态条件）

2、监测接口状态条件、没有用户参数的工作模式

这种工作模式没有用户参数，但监测PNE 接口状态信号REQ_IT 与S_RTS。与上一种简单模式相比，更可靠保证了用户模板与PNE-420接口数据交换不会中断，不会出现双方死等对方发送的情况。特别是对REQ_IT 的监测，保证PNE-420接口卡（由于意外）退出数据交换状态，用户板可以及时再初始化PNE-420接口，使其进入数据交换状态。

关于PNE-420接口状态信号REQ_IT 与S_RTS，第三章已有介绍；为读者方便现摘要如下：

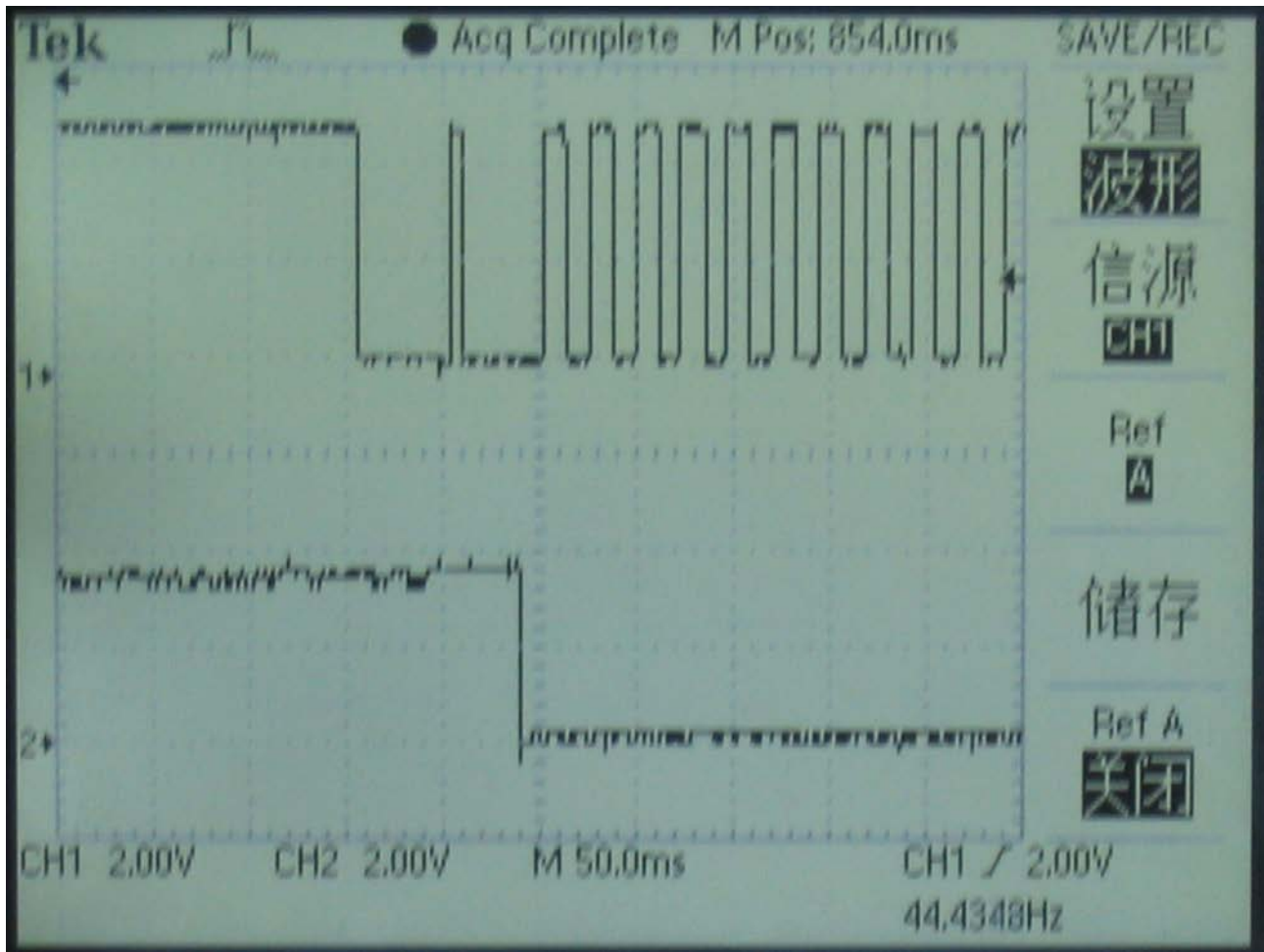
J1-16 中REQ_IT 与S_RTS 定义如下:

管脚	I/O	说明	管脚	I/O	说明
4	O	S_RTS接收允许; 0: PNE等待(允许)接收; 1: PNE拒绝接收	14	O	REQ_IT初始化等待/完毕 0: PNE初始化完毕 1: PNE等待初始化

PNE-420状态表如下所示:

PNE-420 输出 REQ_IT	PNE-420 输出 S_RTS	PNE-420 状态	用户板应进行的操作
1	0	PNE-420 等待接收初始化数据;	发送初始化数据;
0	0	PNE-420 等待接收PROFIBUS 输入数据;	发送PROFIBUS 输入数据;
1	1	PNE-420 接收初始化数据完毕,正处在: →分析初始化数据 →发送初始化失败报文 →初始化PROFIBUS 接口过程中;	等待、接收PNE-420 接收初始化回答报文;
0	1	1.PNE-420处在上电或复位后准备状态; 2.PNE-420正在回答初始化成功报文; 3.PNE-420正在发送PROFIBUS 输出数据;	1.上电或复位后: 等待PNE-420 请求初始化回答报文; 2.发送完毕初始化数据: 等待接收PNE-420接收初始化回答报文; 3.处在数据交换状态: 等待接收PROFIBUS 输出数据;

PNE-420接口从接收初始化到进入数据交换状态的信号时序图如下所示:



监测接口状态条件、没有用户参数的工作模式程序流程如下图所示：

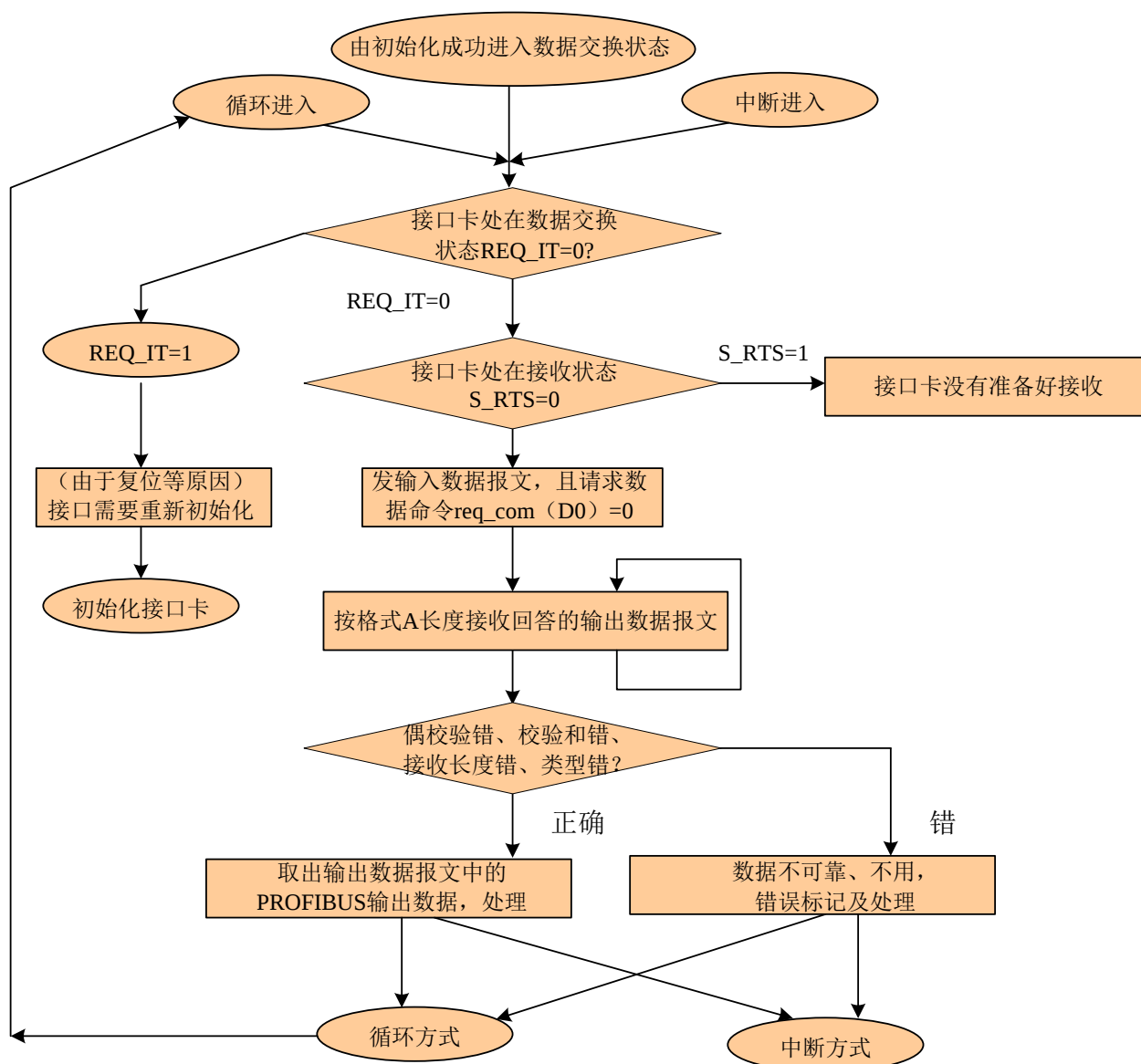


图3-13 数据交换状态下用户模板的程序流程（监测接口状态条件、没有用户参数的工作模式）

3、包括用户参数、监测接口状态的完整工作模式

这种工作模式较上一种，增加了用户参数功能。是否选用用户参数功能，要依用户产品功能要求而定。

用户模板的程序流程见下图：

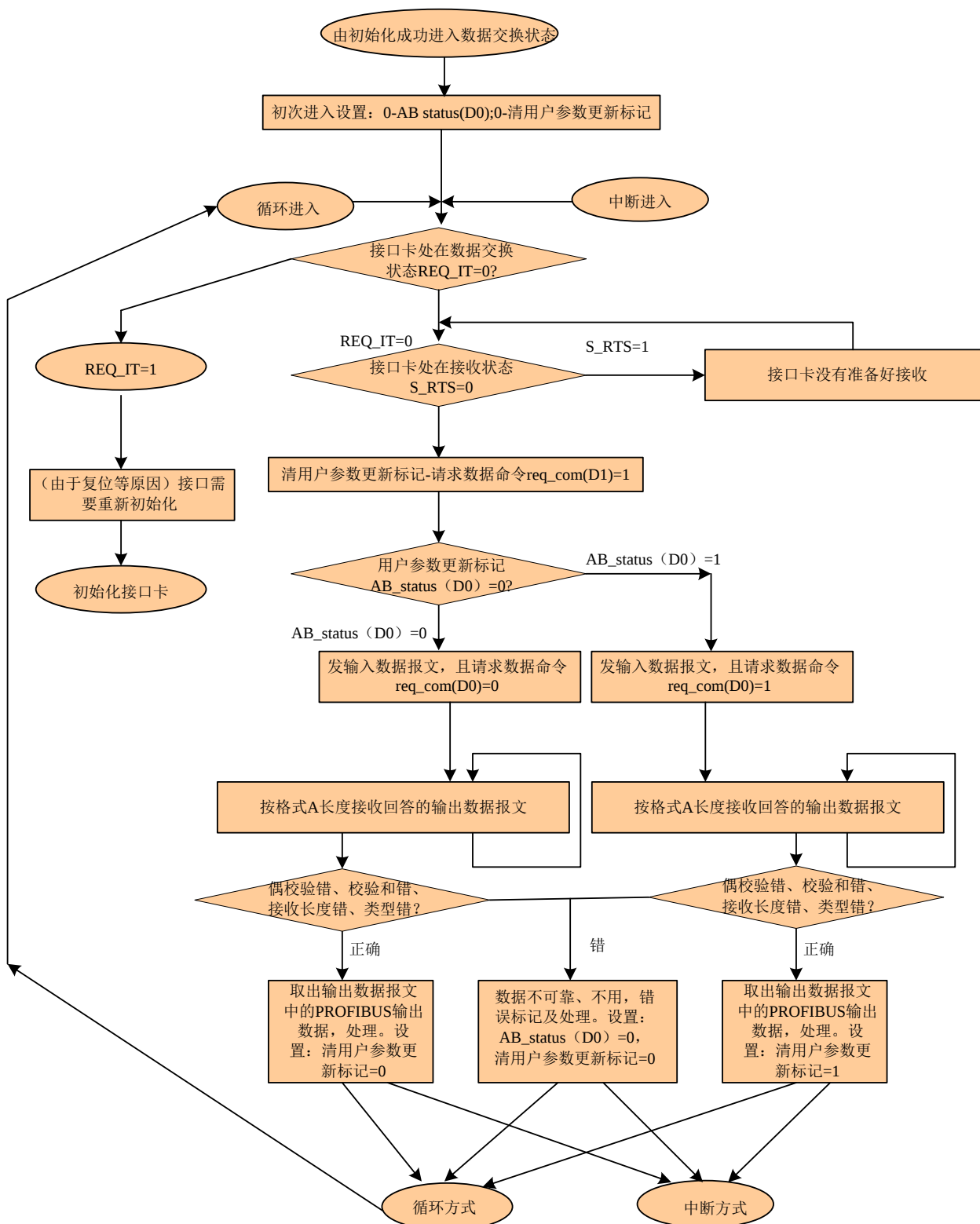


图3-14 数据交换状态下用户模板的程序流程（完整工作模式）

4、数据交换状态下接口卡与用户模板的通信流程

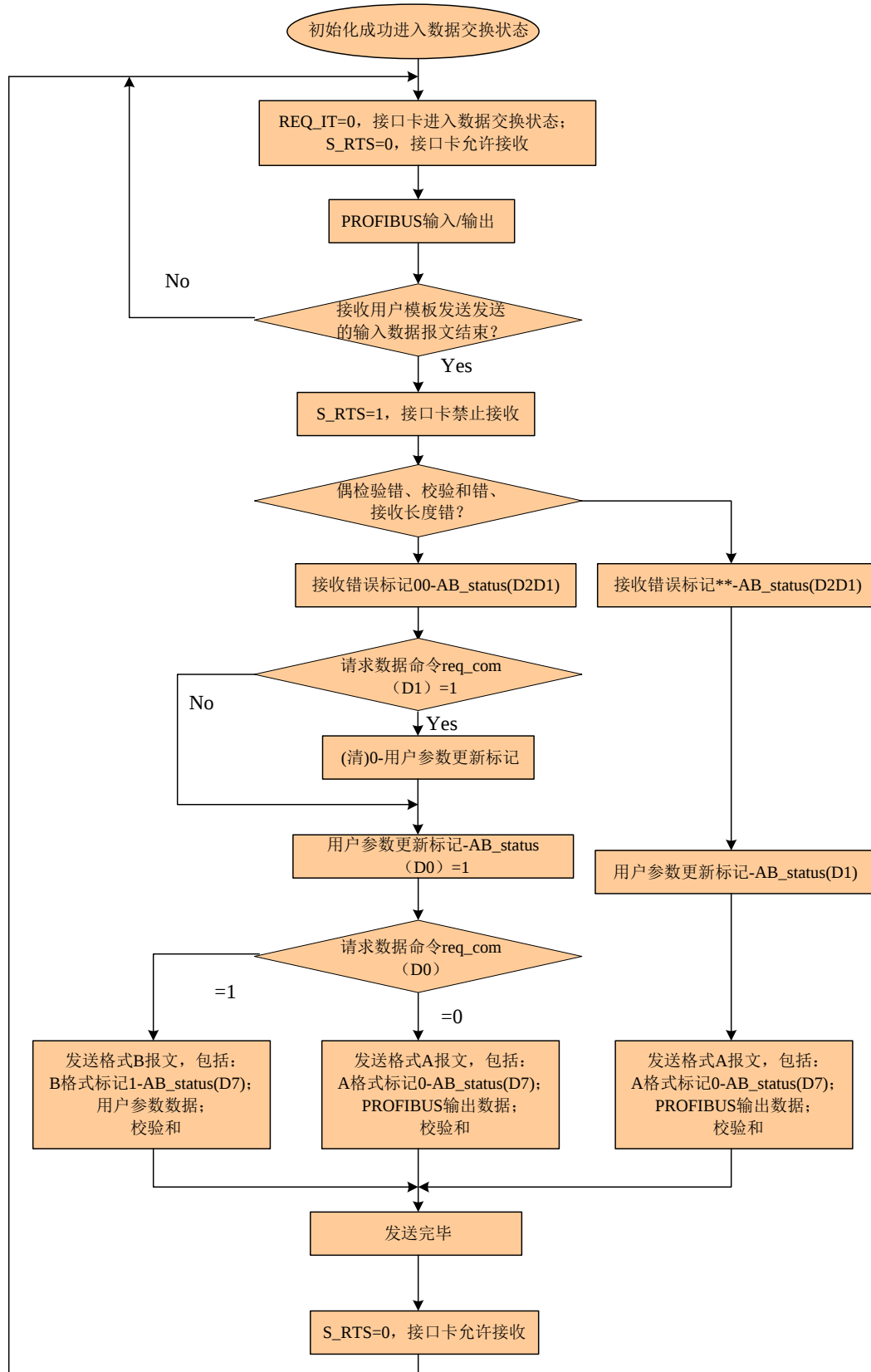


图3-15 数据交换状态下接口卡的程序流程

3.2.2.3 举例说明数据交换报文格式

(1) 不使用用户参数功能举例

例3：用户产品需要：20个字节输入+20个字节输出

查表4-1，20个字节输入代码为：0x1f (16 byte input), 0x13(4 byte input);

20个字节输出代码为：0x2f (16 byte output); 0x23(4 byte input) ；

用户参数长度User_Prm_Data_Len=0;

接收数据长SD_in_len $\geq$ 20+1+1=22

发送数据长SD_out_len $\geq$ 20+1+1=22

则：GSD 文件中必须有：

User_Prm_Data_Len=0

Module="20bytes in, 20bytes out" 0x1f, 0x13, 0x2f , 0x23

EndModule

初始化结束后，用户模板发输入数据报文格式如图3-16-1、图3-16-2。

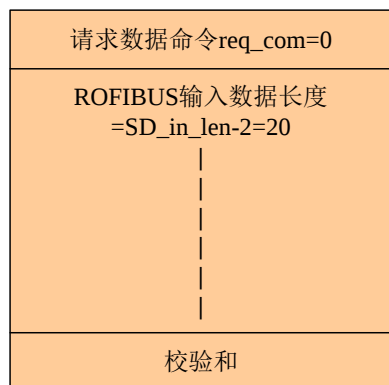


图3-16-1 不使用用户参数功能条件下，用户板永远发请求数据命令req_com=0

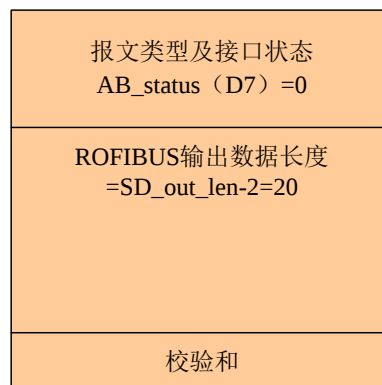
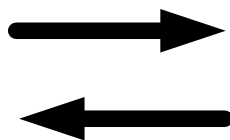


图3-16-2 接口板总是回答输出数据报文格式A，且AB_status (D7) =0

(2) 使用用户参数功能举例

例5：站号设置为0x32，ID 号=0187；

I/O 配置：1个字节输入+12个字输入=25个字节输入；

1个字节输出+2个字输出=5个字节输出；

I/O 配置数据长度CFG_LEN=4；

I/O 配置数据为：0x10, 0x5B, 0x20, 0x61

用户参数长度User_Prm_Data_Len=10；

接收数据长 $SD_in_len \geq 25 + 1 + 1 = 27$

发送数据长 $SD_out_len \geq 5 + 1 + 1 = 7$

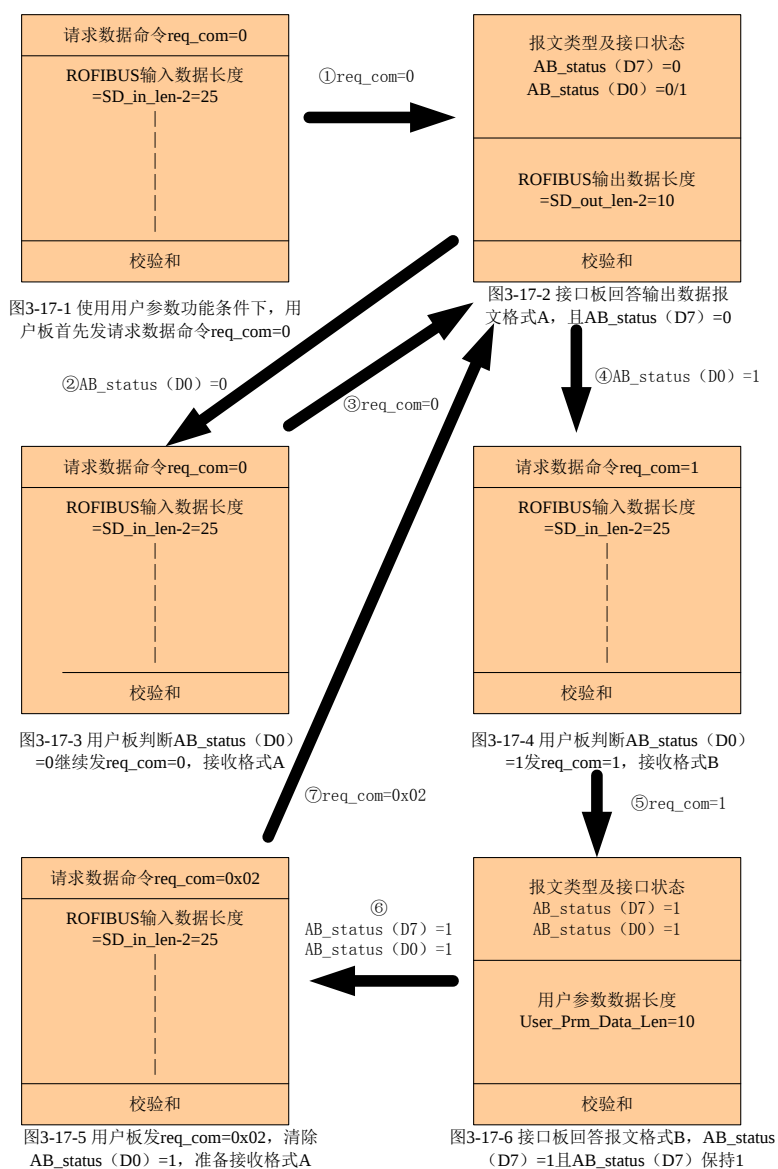
GSD 文件必须中有:

User_Prm_Data_Len=10

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

EndModule

初始化结束后, 用户模板首先发输入数据报文格式如图3-17-1、图3-17-2、图3-17-3、图3-17-4、图3-17-5、图3-17-6:



四、如何使用用户参数

本章结合实例，详细讲解PROFIBUS 用户参数技术。

4.1 使用用户参数 user_prm 的场所

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）。

4.2 用户参数实现方法

(1) 以一个实验板为模型，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），将实验I/O配置成具有1 路AI+1 路AO 的模拟量、和16DI+16DO 的开关量的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；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) 实现上述功能的方法：

- ① 在实验I/O 的GSD 文件中详细描述用于配置的用户参数；
- ② 主站配置中由用户选择这些配置参数；
- ③ 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；
- ④ 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I、16DI+16DO 开关量I/O 的功能。

4.3 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/0.5A 4 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。

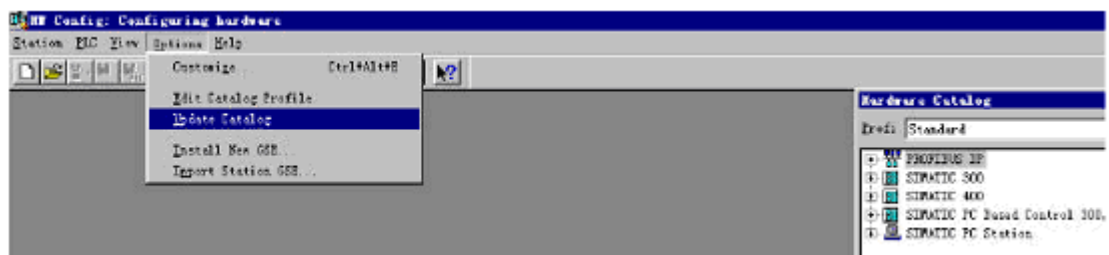
4.4 如何在主站配置中选择用户参数

以STEP 7为例:

- ① 实验I/O的GSD文件拷至: SIEMENS\step7\S7data\gsd\, 实验I/O的图标文件拷至: SIEMENS\step7\S7data\nsbmp\

- ② 进入SIMATIC Manager → HARDWARE, 选择Options → Update Catalog;

www.sibotech.net



③ 配置实验I/O从站。用户可以按照表3-1：PNE-420 PROFIBUS I/O配置数据表，改变用户参数配置实验I/O。

五、GSD 文件

关于GSD 文件（Electronic Data Sheet）

① 每一个PROFIBUS 从站或一类主站都要有一个“设备描述文件”称为GSD 文件，用来描述该 PROFIBUS-DP 设备的特性。

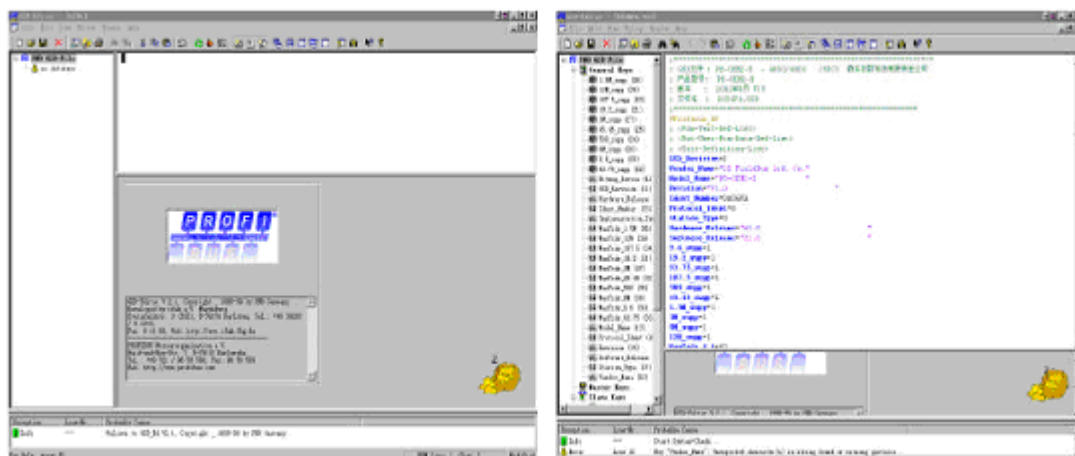
② GSD 文件包含了设备所有定义参数，包括：

- ✧ 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

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。





附录：便于用户使用的 GSD 文件

```
=====
; GSD 文件 : PNE420- (<=210 DI/210 DO) /VPC3 上海泗博自动化技术公司
; 产品型号: PNE420
; 版本 : 2010 年 2 月 V1.0
; 文件名 : PNE420.GSD
=====

#Profibus_DP
; <Prm-Text-Def-List>
; <Ext-User-Prm-Data-Def-List>
; <Unit-Definition-List>
GSD_Revision=2
Vendor_Name="Shanghai Sibotech Ltd. Co." ;公司名，可按用户公司名称修改
Model_Name="PNE-420" ;模块名称，是组态时该产品的名称
Revision="V1.0 "
Ident_Number=0x0dc9 ;ID 号，即 Vendor ID，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
dc9.5_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_dc9.5=60
MaxTsdr_500=100
MaxTsdr_1.5M=150
MaxTsdr_3M=250
MaxTsdr_6M=450
```



MaxTsdr_12M=800

Implementation_Type="spc3"

Bitmap_Device="ET200B_N"

;模块图标文件，用户可以自制图标

; Slave-Specification:

OrderNumber="PNE-420"

;产品订货号，可按实际情况修改

Freeze_Mode_supp=1

Sync_Mode_supp=1

Auto_Baud_supp=1

Fail_safe=0

Min_Slave_Intervall=3

Max_Diag_Data_Len=6

User_Prm_Data_Len=16

user_prm_data=0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f,0x10

Modular_Station = 1

Max_Module = 1

; Max. 1 Module

Max_Input_Len = 210

; Max. 210 Byte Input-Daten

Max_Output_Len = 210

; Max. 210 Byte Output-Daten

Max_Data_Len = 420

; Max. 420 Byte Gesamt-Daten

Modul_Offset=0

Slave_Family=0@CONVERTER

;组态中的分类名

; <Module-Definition-List>

Module=" 210 Byte In, 210 Byte Out "

0x40,0x7F,0x80,0x7F,0x40,0x68,0x80,0x68

;I/O 配置，必须与

初始化报文一致

EndModule

上海泗博自动化技术有限公司

SiboTech Automation Co., Ltd.

技术支持热线:021-5102 8348

E-mail: support@sibotech.net

网址: www.sibotech.net
