智

能

从

站

软

件

说

明

书

目 录

[1.1 VPC3+工作原理 1](#_Toc416856974)

[1.1.1.1 VPC3+管理功能 3](#_Toc416856975)

[1.1.1.1.1 VPC3+中断管理 3](#_Toc416856976)

[1.1.1.1.2 VPC3+状态机管理 3](#_Toc416856977)

[1.2 DP报文分析 4](#_Toc416856978)

[1.3 PROFIBUS诊断 6](#_Toc416856979)

[1.4 设备数据文件 7](#_Toc416856980)

[1.4.1 GSD文件简介 7](#_Toc416856981)

[1.4.2 GSD文件组成 7](#_Toc416856982)

[1.4.3 GSD文件格式 8](#_Toc416856983)

[1.4.4 GSD文件 8](#_Toc416856984)

[1.5 IBUS通讯协议 11](#_Toc416856985)

[1.5.1 IBUS从站地址 11](#_Toc416856986)

[1.5.2 IBUS通讯速率 12](#_Toc416856987)

[1.5.3 IBUS报文结构 12](#_Toc416856988)

[1.6 CPU模块软件设计 14](#_Toc416856989)

[1.7 IO模块软件设计 16](#_Toc416856990)

1.8 小结 19

# 智能从站软件说明书

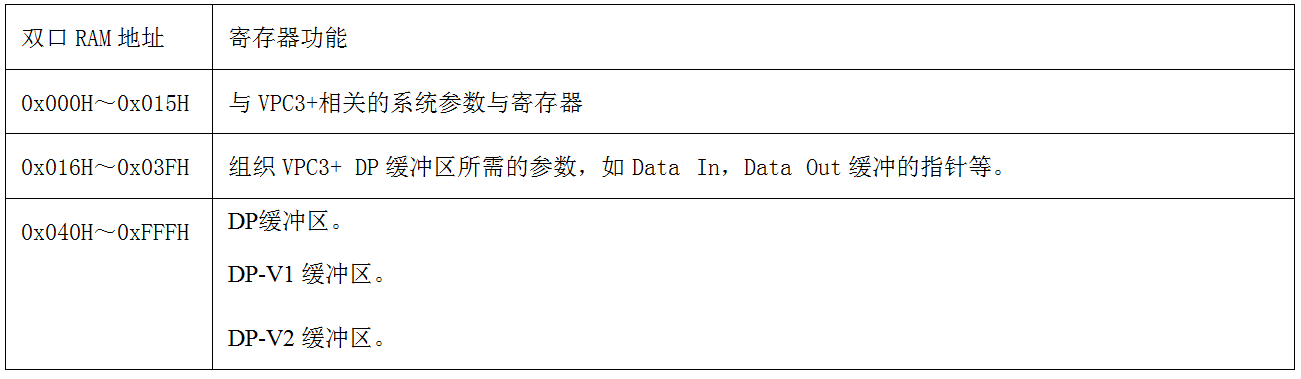
智能从站的软件设计是通过程序手段来实现PROFIBUS-DP协议的总线标准，智能从站所涉及的软件包含智能从站的核心器件VPC3+的PROFIBUS的从站接口软件以及和IO模块中的数据交换协议软件两部分。设计设备数据文件以确定智能从站的功能，设计智能从站的CPU模块与I/O模块的通讯协议以确定智能从站的工作原理与控制逻辑。

## VPC3+工作原理

VPC3+支持全部的PROFIBUS-DP协议。VPC3+内部集成了4KB的双口RAM。双口RAM的内部地址从0x000H～0x015H为内部寄存器与锁存器。主要包括模式寄存器、中断控制寄存器和状态寄存器以及IO缓冲区和诊断缓冲区等。

VPC3+的4KByte 双口RAM共分为三个区，从0x000H～0x015H为系统参数（或寄存器）区，0x016H～0x03FH为组织参数区，0x040H～0xFFFH为缓冲区，缓冲区通常由两到三个区组成，用以实现CPU对输入数据的并行处理，如表4-1所示。

表4-1 VPC3+双口RAM分配图表



VPC3+ 双口RAM的一共有256个段，每个段有8个字节。每次以段这单位进行访问，如果程序需要使用双口RAM中的连续区间超过8个字节长度时，就要占用2个段空间。在VPC3在初始化过程中将缓存空间进行计算，这些缓存空间的大小也是以段为单位的。初始化时用户计算出缓存区空间若超出4Kbyte的范围时，VPC3+将产生中断告知用户。

在VPC3+ RAM中，对一些区域的读和写的定义是完全不同的，例如在对0x004H和0x005H进行读处理时，0x004H与0x005H就是状态寄存器；在对0x004H和0x005H进行写处理时，0x004H与0x005H就变成了中断屏蔽寄存器。

针对读/写操作的不同定义参见VPC3+数据手册。

### VPC3+管理功能

### VPC3+中断管理

VPC3+一共有四个与中断相关的寄存器，分别是：中断请求寄存器(Interrupt Request Register )、中断寄存器(Interrupt Request)、中断屏蔽寄存器(Interrupt Mask Request)和中断确认寄存器(Interrupt Acknowledge Request)。每个寄存器占2个字节，VPC3+共支持包括波特率检测(Baud\_Rate\_Detect)、新参数数据(New\_Prm\_Data)、诊断缓冲区改变(Diag\_Buffer\_Changed)在内的11个中断。VPC3+只有一个中断寄存器(0x01H)，中断服务程序在获取中断请求后，用户通过中断请求寄存器里的内容来判断是哪一个事件引发的中断。在初始化时开发人员通过设置中断屏蔽寄存器中的指定位来打开或关闭某个中断响应。用户完成中断事件的响应以后，需对中断确认寄存器(Interrupt Acknowledge Request)或相应的命令寄存器CMD中的某些位来确认相应中断事件。

### VPC3+状态机管理

由于VPC3+运行时工作在四种状态：上电状态、等待参数状态、等待配置状态和数据交换状态。图4-2给出的VPC3+状态机转换图，通过状态转移图确定VPC3+工作状态之间的关系。VPC3+的状态机是确保PROFIBUS从站设备在同一总线下的工作时序统一。在上电状态时，从站接收来自诸如操作员站或人机界面的二类主站所传递的设置从站地址报文，对从站地址进行修改，此类从站需要诸如EEPROM来存放从站地址。在从站内部启动初始化以后，从站进入了等待参数状态，等待主站设置参数报文。在等待参数状态，从站同样接收读取从站配置报文或从站诊断报文，但在此状态屏蔽了其他报文，所以并不能进行数据通讯。当设置参数正确以后，从站进入等待配置状态，等待从站校验配置报文。检验配置报文规定了从站的输入和输出字节数。另外也可以接收从站诊断和读取配置报文。



图4-2 PROFIBUS-DP从站状态机转换图

当VPC3+检验配置正确后，从站进入了数据交换模式，并进行数据传递，在此状态中，从站可以接收来自一类主站的写输出、读输入、全局控制、从站诊断、检验诊断和读取配置等报文信息。当从设备发生异常时，如看门狗定时器WD超时，则VPC3+的状态就会改变，返回等待参数状态，以保证系统的安全。开发人员在软件调试过程中可以通过追踪VPC3+的状态寄存器来确认VPC3+工作状态。

## DP报文分析

PROFIBUS-DP报文是PROFIBUS-DP主站与从站之间的数据交换指令，主要有9个服务访问点（SAP）。每一个PROFIBUS系统至少有一个主站，每一个PROFIBUS网络上最大可有127个设备，每个从站支持最大244字节的输入/输出及诊断数据，主站与从站的数据交换过程如图4-3所示。主站根据从站地址的大小，依次轮询，因此地址小的从站优先。



图4-3 PROFIBUS-DP主站与从站数据交换

在PROFIBUS-DP协议中规定了8种主站-从站功能：读取DP从站诊断信息、读取一类主站（如PLC）传送的输入输出数据、读取DP从站的输入与输出数据、发送从站的参数数据、检查主站的组态数据、读主站的组态数据、对DP从站的控制命令以及变更DP从站的站地址等。

在PROFIBUSDP总线上传送的帧分为四种类型：数据字段长度可变的帧、有数据字段长度固定的帧、无数据字段固定长度帧和令牌帧。图4-4及4-5给出了有数据字段的固定长度的帧中发送／请求帧和响应帧的格式。

发送与请求帧的数据格式如图4-4所示，其中SYN是帧同步时间，SYN为最小33个线空闲位，SD3是开始定界符，SD3固定为A2H。DA是该帧的目的地址，SA是该帧的源地址，FC是帧控制，DATA\_UNIT为数据字段，固定长度为（L-3）共8个位组，FCS是帧检查次序，ED是结束定界符，Ed的数值为16H，最后L为信息字段长度，固定的八位位组的个数：L=11。



图4-4 含数据字段的发送/请求帧数据格式

应答帧的数据格式如图4-5所示，回答帧与发送/请求帧的不同之处在于回答帧没有同步帧。



图4-5含数据字段的回答帧数据格式

对于利用VPC3+芯片来开发DP从设备的开发人员来讲，无需考虑如何将从总线接收的帧进行检验、解析和处理，也不用考虑如何将应答报文发送到总线上去，这些功能都是由VPC3+完成。同时需要说明的是，从站的用户程序对协议报文的收发控制是很有限的。

DP报文见表4-2所示，其中SSAP为源服务存取点，DSAP为目的服务存取点，SA为源地址，DA为目的地址。

表4-2 DP通讯报文

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能 | DP主站 | | DP从站 | |
| SSAP | SA | DSAP | SA |
| Data\_Exchange | — | XX | — | XX |
| RD\_Inp | 62 | XX | 56 | XX |
| RD\_Outp | 62 | XX | 57 | XX |
| Slave\_Diag | 62 | XX | 60 | XX |
| Set\_Prm | 62 | XX | 61 | XX |
| Chk\_Cfg | 62 | XX | 62 | XX |
| Get\_Cfg | 62 | XX | 59 | XX |
| Global\_Control | 62 | XX | 58 | XX |
| Set\_Slave\_Add | 62 | XX | 55 | XX |

在上面介绍的DP主站/从站服务中，诊断处理、参数化、检查组态数据和从设备地址设置四种协议报文是DP通讯报文中相对重要的，因此对于从站研发人员来说，是必须要熟练掌握其控制方法。

## PROFIBUS诊断

当从站设备发生异常，无论是设备级还是模块级异常，都可以通过诊断报文向主站传递异常信息，此信息是由从站设备主动发送的。标准的诊断信息有6个字节，而且用户可以在标准诊断信息之后加入设备相关的诊断信息，最长可扩展到244字节。

VPC3+在收到应用程序传过来的诊断数据后，并不是立刻发给主站。而是在当前的服务响应中置上标记，当主站收到这样的标记后，会在下个轮循周期内读取从站的诊断信息。

从站提供的诊断信息能快速确定故障的位置。从站诊断信息分为三级:

（1）与设备有关的诊断，指设备的一般运行状态信息，如温度过高，电压过低。

（2）与模块有关的诊断，指设备中有关模块的信息，如输出模块1有故障.

（3）与通道有关的诊断，指输入或输出某一信号位的信息，如输出线1短路。

图4-6显示的是PROFIBUS推荐的一个物理或虚拟的模块式从站的诊断信息结构每个模块故障时“模块状态”区的一个“通道相关的诊断”区将会产生。



图4-6 PROFIBUS的诊断信息结构

## 设备数据文件

### GSD文件简介

PROFIBUS设备由各个厂商提供，各自具有不同的性能特点，为了使PORFIBUS设备能够达到即插即用的组态配置要求。因此PROFIBUS设备的所有特性都必须在电子设备数据文件（GSD）中具体说明，在PROFIBUS标准描述中，以德文编辑的电子设备数据文件以“.GSD”扩展名命名，以英文编辑的则以“.GSE”扩展名命名，由于最初是德文格式，所以常以GSD表述。完善的GSD数据可以将PROFIBUS-DP通信扩大到操作员控制级别。使用基于GSD的组态工具可以将不同厂商生产的设备集成在同一系统中。

### GSD文件组成

GSD文件从第一版到目前最新的第五版，每一个版本所支持的规范不同，从第一版的只DPV0规范到第五版支持DPV2规范，GSD文件整体可以分为三个部分：

（1）设备级规范

设备级规范包括设备厂商和设备的名称、产品的认证号、软硬件版本状况，以及该设备支持的波特率和最大延迟时间及总线接口的信号类型，这一部分规范从GSD第一版到第五版都没有变化；

（2）主站级规范

主站级规范包括只运用于DP主站的各项参数，如连接的最多从站数量，以及从站上传与下载能力等，在主站级规范这部分对从站没有规定。

（3）从站级规范

从站级规范包括与从站有关的规范，如输入/输出通道的数量和类型以及中断测试的规范以及输入/输出数据一致性的信息等。

### GSD文件格式

由于GSD文件是ASCII文件，可以用任何一种ASCII编辑器编辑，如NotePad、UltraEdit等，PI（PROFIBUS用户组织）也专门提供为GSD文件编辑程序GSDEdit。GSD文件类似于解释性文本由若干行级成，每行开头都是一个关键字，每行文本包括关键字及参数两部分，参数为无符号整数或字符串。GSD文件中的关键字则由PROFIBUS标准定义的标准关键字或自定义关键字。PROFIBUS的所有组态工具都可以识别标准关键字，而自定义关键字只能被特定的组态工具识别。

### GSD文件

GSD文件主体内容包含六个部分内容：从站定义列表、DP从站设备支持的波特率和DP从站设备的最大延迟时间、DP从站设备接口特征、组态中的显示图标、DP设备从站特性与模块信息。

1）从站定义列表

从站定义列表也就是与设备相关的一般规范，包括生产厂商代码、设备名称和产品识别号，这些都是在DP产品族中唯一的是由PI组织统一命名与指定的，设备必须通过PI的产品认证后才可以获得由PI组成，硬件和软件的版本状况，本课题所设计的GSD文件的从站定义列表如图4-7所示。



图4-7 GSD文件从站定义列表

2）DP从站设备支持的波特率和最大延迟时间

DP从站设备支持的波特率和最大延迟时间标明了从站所兼容的波特率列表，DP从站设备的最大延迟时间也标明了VPC3+在收到请求之后到做出应答之前的最大延迟时间，其单位为位时间。本课题所设计的智能从站的DP从站设备支持的波特率和最大延迟时间如图4-8所示。



图4-8 GSD文件从站设备支持的波特率和最大延迟时间

3）DP从站设备接口特征

DP从站设备接口特征是定义从站的关键参数，供用户选型时参考，包括是否支持冗余、RTS信号、接口是否提供24V电压以及采用什么解决方案等。如图4-9所示。



图4-9 GSD文件从站设备接口特征

4）组态中的显示图标

组态中的显示图标是方便在组态软件中从众多从站设备中定位本课题设计的智能从站，在组态软件在线调试时，从站正常时，从站的图标将以txr\_n.bmp图片显示，当设备报警和故障时，分别由txr\_d.bmp和txr\_s.bmp图片显示，本次设计中将正常工作的图片颜色基调为绿色，而报警与故障分别为黄色和红色。



图4-10 GSD文件中组态显示图标

5）从站特征

从站特征是描述与DP从站有关的信息，包括与从站相关的规范，是否支持锁定模式，是否支持同步模式，是否支持自动波特，是否支持改变从站地址，站类型，最大从站模块数量，最大输入/输出数据长度，最大诊断数据长度等，如图4-11所示。



图4-11 GSD文件中从站特征描述

6）模块信息

模块信息明确IO模块的全部信息，包括IO模块类型，数据长度，以及在组态软件中可以配置的参考数据。如图4-12所示。



图4-12 GSD文件中模块信息描述

## IBUS通讯协议

IBUS通讯协议的主要功能是将PROFIBUS-DP主站的组态与控制信息由CPU模块转发至从站的IO模块，同时将从站的IO模块配置信息及IO模块采用的本地数据由CPU模块传递给PROFIBUS-DP主站，实现了智能从站的数据采集与远程控制功能。在智能从站中，IBUS采用“一主多从”的通讯架构。IBUS主站为CPU模块，也是IBUS总线网络中的管理设备，负责管理IBUS网络中的通讯。IBUS从站为IO模块，IBUS主站建立与从站的数据通讯，从IBUS从站获取输入数据，并向从站分配输出数据。IBUS主站与从站统称为IBUS节点。

IBUS通讯协议包含了IBUS从站地址、通讯速率、报文结构以及通讯数据等部分组成。

### IBUS从站地址

IBUS从站地址在智能从站中就是I/O模块地址，IBUS最大从站数目为16个，由一只转盘拨码盘设置，转盘拨码盘的数据范围从0～F对应IBUS从站地址0～15，即智能从站最多可以扩展16个模块，见表4-3所示。模块在安装时需要将地址由靠近CPU模块的近端设为0，其余模块依次设置。

表4-3 IBUSI/O模块地址表

|  |  |
| --- | --- |
| 旋转式拨码开关 |  |
| 转盘拨码 | 0～F |
| I/O模块地址 | 0x0～0xF |

### IBUS通讯速率

由于IBUS从站的拼接的距离非常接近，因此IBUS通讯速率选择CAN BUS通讯的最高速率1Mb/s。

### IBUS报文结构

IBUS协议报文格式定义了CAN报文的标识符和数据部分使用原则和功能含义，通过对报文格式的定义，使网络中CAN报文的每个组成元素均有其特定的功能和意义。CAN报文格式的定义是IBUS协议规范中最基本的组成部分，包括报文标识符和报文数据部分的分配两个方面。

在IBUS协议规范中，只使用了标准帧格式CAN报文（CAN2.0A），扩展帧格式CAN报文和远程帧格式报文都未使用。IBUS协议对标准帧格式CAN报文的11位标识符和报文数据部分的使用作了详细的定义，其报文的格式定义如表4-4所示。

表4-4 IBUS帧标识符定义

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧标识符 | ID10 | ID9 | ID 8 | ID 7 | ID 6 | ID 5 | ID 4 | ID 3 | ID 2 | ID 1 | ID 0 |
| 说明 | DIR | DesMacID | | | | Init | FuncID | | | | |

表4-4中，DIR为IBUS数据发送方向，DIR=0时，表示该报文是由IBUS主站发送给从站的，DIR=1时，表示该报文是由IBUS从站发送给主站的。DesMacID是从站地址，由IO模块上的转盘拨码设置，从站地址范围为0～15，占用帧标识符的ID6～ID9，Init是从站初始化，Init=0，表示从站未初始化，IO模块发送模块配置数据，Init=1，表示从站初始化完成，IO模块发送模块采集数据。FuncID为IBUS协议功能码。

FuncID=00：从站发送主站DI诊断数据

表4-5从站发送主站DI诊断数据

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | 0~3 | 4~7 | 8~11 | 12~15 | 16~19 | 20~23 | 24~27 | 28~31 |
| 00正常，01开路故障，10短路故障，11该位不存在 | | | | | | | | |

FuncID=01：从站发送主站DO诊断数据

表4-6从站发送主站DO诊断数据

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | 0~3 | 4~7 | 8~11 | 12~15 | 16~19 | 20~23 | 24~27 | 28~31 |
| 00正常，01开路故障，10短路故障，11该位不存在 | | | | | | | | |

FuncID=02：从站发送主站初始化参数：模块类型与模块长度。

表4-7从站发送主站初始化参数？

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | 备用 | 模块  类型 | 模块  长度 |  |  |  |  |  |
| 模块类型：01🡪DI，02🡪DO，03🡪DX, | | | | | | | | |
| 模块长度：00🡪4Byte，01🡪3Byte，02🡪2Byte，03🡪1Byte， | | | | | | | | |
| 模块类型：P5.06,P5.07：01🡪DI，02🡪DO，03🡪DX | | | | | | | | |
| 模块长度：P5.05,P5.10：00🡪4Byte | | | | | | | | |

FuncID=03：从站发送主站DI/DO参数

表4-８从站发送主站IO参数？

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | DO0 | DO1 | DO2 | DO3 | DI0 | DI1 | DI2 | DI3 |
| 无效数据以00填充 | | | | | | | | |

FuncID=8：主站发送从站组态数据，帧扩展格式

表4-９主站发送从站组态数据

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data | 0 | | 1 | | 2 | | 3 | | | 4 | | | 5 | | 6 | | 7 |
|  | 备用  扩展 | | 模块  类型 | | 模块  长度 | | 配置  参数1 | | | 配置  参数2 | | | 配置  参数3 | | 配置  参数4 | | 配置  参数5 |
| 如果模块类型与模块长度与从站设置相同，从站组态初始化通过。Init=1否则为0，从站不接收主站IO口控制命令 | | | | | | | | | | | | | | | | | |
| Di | 0 | 1 | | | | 2 | | 3 | | | 4 | 5 | | 6 | | 7 | |
| 帧1 | 0x40 | Type | | Len | | w\_p\_Mode[0]...[1] | | | | | w\_p\_DiSE[0]...[3] | | | | | | |
| 帧2 | 0xC0 |  | | | |  | | |  | | w\_p\_Force[0]...[3] | | | | | | |
| Do | 0 | 1 | | | | 2 | | | 3 | | 4 | 5 | | 6 | | 7 | |
| 帧1 | 0x40 | Type | | Len | | w\_p\_Mode[1] | | | w\_p\_FailSafe[0]...[4] | | | | | | | | |
| 帧2 | 0xC0 | w\_p\_FailSafe[5]..[7] | | | | | | | | | w\_p\_Force[0]..[3] | | | | | | |

FuncID=9：主站发送从站DO数据

表4-10主站发送从站DO参数

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | DO0 | DO1 | DO2 | DO3 | 00 | 00 | 00 | 00 |
| 无效数据以00填充 | | | | | | | | |

FuncID=10：主站查询从站DI参数，数据位全为0x00；

FuncID=11：主站查询从站DI诊断，数据位全为0x00；

FuncID=12：主站查询从站DO诊断，数据位全为0x00；

FuncID=13：主站查询从站，数据位全为0x00；

FuncID=14：主站收到从站初始化指令后发送确认数据帧，从站收到后不再发送初始化数据。

FuncID=15：主站查询从站，从（0#到15#）共16个有效站

表4-11主站查询从站数据

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DIR | DesMacID | | | Init | | FuncID | | | |
| 0 | 10…20[0x0A…0x14] | | | 0 | |  | | | |
| 主🡪从 | 从站号9~20，  9号为虚拟站  0x1F为广播地址 | | | 0：未初始化  1：已初始化 | |  | | | |
| 数据位 | 0 | 1 | 2 | | 3 | 4 | 5 | 6 | 7 |
|  | DO0 | DO1 | DO2 | | DO3 | 00 | 00 | 00 | 00 |
| 无效数据以00填充 | | | | | | | | | |

## CPU模块软件设计

在CPU模块软件设计过程中，采用主程序模块结合中断处理模块的软件结构。主程序模块的功能是智能从站的初始化、VPC3+的状态机处理和智能从站的诊断处理，主程序根据VPC3+的状态寄存器中的状态进行系统状态转移与切换，主程序中处理对时间要求相对较低的事件，如采集与更新IO模块的数据，数据缓存区填充处理及诊断缓存区的数据处理等；中断处理模块则处理从站的联机与脱机处理、参数化配置报文、配置模块通道数据设置看门狗定时器等要求响应较高的事件。主程序模块工作流程图如图4-13所示，在主程序模块的工作流程中，VPC3+初始化是在主程序模块中非常关键。VPC3+的初始化过程主要包括以下内容：

* VPC3+中断配置。VPC3+最多支持11种中断，从站需要对每一个中断进行设置，通过VPC3+ RAM中的中断屏蔽寄存器来设定对某些中断的打开或关闭。
* WD定时器配置。当WD定时器超时后，VPC3+会转换其工作状态，以防止设备故障时影响到整个总线系统的安全。WD定时器配置（R\_User\_WD\_Value）在VPC3+ RAM内的地址为0x18，0x19。
* 制造商标识ID设置。制造商标识ID由PI组织授予，在VPC3+ RAM中，设置制造商标识ID的地址为0x3a，0x3b: R\_Ident\_Low和R\_Ident\_High。
* 缓冲区初始化设置。开发人员必须确定VPC3\_BUFINIT结构体中定义各个用于信息交换的缓冲区的长度，这些缓冲区长度决定了VPC3+中数据缓冲区的长度，因此所有缓冲区长度不能超过缓冲区总长度。用VPC3\_buf\_init( )函数将VPC3缓存区初始化，在VPC3+的双口RAM中分配各缓冲区，检查各个缓冲区的最大长度，并返回缓冲区初始化后的返回数据。
* 波特率控制设置。如果总线上没有有效的信息被接收，VPC3+将启动波特率搜索BAUD\_RATE\_RESEARCH功能。



图4-13　主程序模块流程图

VPC3+只提供一个外部中断，从站程序在响应中断后立即读取中断请求寄存器（Int-Req-Req）中的内容，以判断中断事件的类型，VPC3+ 双口RAM中Int-Req-Req的地址为初始的两个字节。中断处理模块主要处理以下中断：检查并处理新的配置数据报文和参数数据报文，检查和处理新的诊断缓冲数据，检查和处理新的全局控制命令等；中断模块流程图如图4-14所示。



图4-14　中断模块流程图

## IO模块软件设计

IO模块软件分为两部分，一部分为IBUS协议软件，另一部分为IO采集控制软件，IBUS协议软件传递从站的配置数据、诊断数据和输入输出数据。IO模块的配置参数包括四个方面的内容：

（1）IO模块的类型与数据长度；

（2）输入信号的滤波时间；

（3）输出信号的故障安全选项（关闭、打开、保持上一状态）；

（4）强制信号。

IBUS从站的模块数据是由PROFIBUS-DP主站中的组态软件设置后传递给智能从站，由IBUS主站分配按照地址依次分配给IBUS从站，IBUS从站的模块数据在软件中以CAN\_UNIT命名，CAN\_UNIT结构体里包含了智能从站的每一个模块由PROFIBUS-DP主站设置的组态数据，从站的模块数据结构体如下：

typedef struct{

uint8\_t m\_Type; 　//子模块的类型

uint8\_t m\_Len; //子模块的数据长度

uint8\_t m\_InpSEt; //输入模块信号延时时间

uint8\_t m\_inpDBt; //输入模块debounce time

uint8\_t m\_Force; //强制

uint8\_t m\_SwSc; //短路后电源动作

uint8\_t m\_Diag; //诊断

uint8\_t c\_InpSeEna[4]; //输入模块信号延时使能信号

uint8\_t c\_OupFS[8]; //输出模块Fail Safe 选择信号

uint8\_t c\_Force[4]; //模块强制使能信号

}CAN\_UNIT;

初始化时IBUS主站申请读取IBUS从站的配置数据，并将该数据与PROFIBUS-DP主站组态数据中的参数进行比较，所有的IBUS从站配置数据校对正确后，初始化工作才真正完成。IBUS从站的配置数据在软件中以CAN\_NODE\_CFG结构体形式体现，初始完成后CAN\_NODE\_CFG将模块类型与数据长度以及读取IBUS主站的配置数据一同传送回IBUS从站，以便IBUS主站进行核对并输出相应的状态或诊断信息。从站的配置结构体如下：

typedef struct {

unsigned char NodeExist; //节点存在

unsigned char Init; //初始化

unsigned char cntCfg; //配置文件齐全的标志为

unsigned char r\_Addr; //模块地址

unsigned char r\_Type; //模块类型

unsigned char r\_Byte; //模块数据长度

unsigned char w\_p\_Mode[2];//从站的配置参数

unsigned char w\_p\_InpSE[4];//从站的Signal Extension

unsigned char w\_p\_FailSafe[8];

unsigned char w\_p\_Force[4];

} CAN\_NODE\_CFG;

当智能从站与PROFIBUS-DP主站配置完成后进行数据交换时，从站实时反馈模块数据，并响应DP主站的实时控制指令，模块的输入输出数据以CAN\_NODE\_DAT结构体定义：

typedef struct {

unsigned char flgNewDat;

unsigned char r\_Addr; //模块地址

unsigned char r\_InpDat[4]; //输入数据

unsigned char r\_DiagDat[8]; //诊断数据

unsigned char w\_p\_Mode[2]; //从站的配置参数

unsigned char w\_p\_InpSE[4]; //从站的Signal Extension

unsigned char w\_p\_FailSafe[8]; //故障安全

unsigned char w\_p\_Force[4]; //强制

unsigned char w\_OupDat[4]; //输出数据

} CAN\_NODE\_DAT;

IO模块的程序控制逻辑大体分四个主要控制模块：模块上电Power\_On，模块初始化WAIT\_IBUS\_MASTER，IBUS主站配置CHK\_M\_CFG和数据交换DATA\_IO。在Power\_On模块中，程序读取IO模块的地址（IBUS地址）并对模块进行自检，获取模块配置参数准备与主站配置参数进行核对，在WAIT\_IBUS\_MASTER模块中程序将主站配置参数与模块实际参数进行核对，如果核对不一致则报警，当配置数据核对通过后进入CHK\_M\_CFG，程序获取主站的配置信息。完成配置以后进入到数据交换环节，当数据交换过程中有任何故障，则重新进入初始化，并向主站发送故障数据，提出重新初始化的请求。模块的控制逻辑如图4-9所示。



图4-9 IO模块程序控制流程图

## 小结

本文主要介绍了VPC3+的工作原理，对VPC3+管理功能、中断管理及状态机进行分析，同时分析了DP通讯报文以及PROFIBUS诊断和设备数据文件，接着设计了IBUS通讯协议、CPU模块软件和IO模块的软件。