

Document Title: PM_FW project configuration module design
description of Safety Control System

Document Number: 17-Q04-000613

Project Number: CT-RD-1601

Project Name: First phase of Safety Control System
Development Project

Material Number: N/A

Document Version: A

Classification Level: Highly secret

Document Status: CFC

Controlled Status: Under control

Prepared by: Liu Yang 2016-11-01

Checked by: Zhu Genghua 2016-11-30

Countersigned by: Li Qi, Wang Dong

Approved by: Wen Yiming 2016-12-30

Revision History

No.	Relevant Chapter	Change Description	Date	Version Before Change	Version After Change	Prepared by	Checked by	Approved by
1		Document created	2016-11-1	None	A	Liu Yang	Zhu Genghua	Wen Yiming
2								
3								
4								
5								

Relationship between this version and old versions: None.

文件名称：安全控制系统 PM_FW 工程配置模块设计说明书

文件编号：17-Q04-000613

项目编号：CT-RD-1601

项目名称：安全控制系统开发项目一期

物料编号：

版本号/修改码：A

文件密级：机密

文件状态：CFC

受控标识：受控

拟制：刘阳

2016 年 11 月 1 日

审核：朱耿华

2016 年 11 月 30 日

会签：李琦、王东

批准：温宜明

2016 年 12 月 30 日

修订页

编号	章节名称	修订内容简述	修订日期	订前版本	订后版本	拟制	审核	批准
1		创建	2016-11-1		A	刘阳	朱耿华	温宜明
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

本版本与旧文件（版本）的关系：

Content 目录

1	Document overview 文档概述.....	1
1.1	Introduction 综述	1
1.2	Reference 参考文档.....	1
1.2.1	Project documents 内部参考文档	1
1.3	Terms and abbreviations 术语和缩略语	1
1.3.1	Terms 术语	1
1.3.2	Abbreviations 缩略语	2
2	Module overview 模块概述.....	3
3	Module design 模块设计	4
3.1	Function description 功能描述	4
3.2	Design concept 设计思路	4
3.2.1	Power-on initialization 上电初始化	4
3.2.2	Cycle operation 周期运行.....	5
3.3	Interface function 接口函数.....	6
3.4	Global variable 全局变量	6
3.5	Data structure 数据结构.....	6
3.5.1	IO configuration information IO 配置信息	6
3.5.2	PRG configuration information PRG 配置信息	8
3.5.3	Project configuration information 工程配置信息	8
3.5.4	P2P configuration information P2P 配置信息	9
3.5.5	Modbus slave configuration Information Modbus 从站配置信息	10
3.5.6	Modbus master configuration information Modbus 主站配置信息	11
3.5.7	RETAIN configuration information RETAIN 配置信息	13
3.5.8	OPC configuration information OPC 配置信息	13
3.5.9	SOE configuration information SOE 配置信息.....	14
3.5.10	VOTE configuration information VOTE 配置信息.....	15
3.6	List of sub-function 子功能列表	16
4	Design of sub-function 子功能设计	16
4.1	Module initialization 模块初始化	16
4.1.1	ConfigInit	16
4.2	Configuration information initialization 配置信息初始化.....	17
4.2.1	ConfigInfoInit.....	17
4.3	Module cycle function 模块周期运行	18
4.3.1	ConfigCycle	18

4.3.2	ConfigPRGFileParse	19
-------	--------------------------	----

1 Document overview 文档概述

1.1 Introduction 综述

This document describes the design description of project configuration function of PM_FW of Safety Control System. The document describes the overall concept of the function of the module, and then the sub-function of the modules are described in detail.

This document is the output of module design phase of PM_FW, and is the input for the follow-up coding phase.

本文档描述了安全控制系统中 PM_FW 工程配置模块的设计方案。文档首先描述了模块功能的总体设计思路，然后将模块功能划分为若干子功能并进行详细说明。

本文档是 PM_FW 模块设计的输出，也是后续编码的输入。

1.2 Reference 参考文档

1.2.1 Project documents 内部参考文档

[1] Embedded software safety concept of Safety Control System [505], 15-Q02-000059

[1] 安全控制系统嵌入式软件安全概念说明书 [505], 15-Q02-000059

[2] PM_FW software overall design description of safety control system [506], 15-Q02-000074

[2] 安全控制系统 PM_FW 总体设计说明书 [506], 15-Q02-000074

1.3 Terms and abbreviations 术语和缩略语

1.3.1 Terms 术语

Table 1-1 Terms

表 1-1 术语

No. 序号	Term 术语	Description 解释
1.	IP_BUS	Communication between PM and IO modules. PM 与 IO 模块之间的通讯总线。
2.	CM_BUS	Communication between PM and CM. PM 与 CM 之间的通讯总线。
3.	PM_BUS	Communication between PMs. PM 之间的通讯总线。
4.	System Net	Communication between control station and PC. 控制站与上位机之间的通讯网络。
5.	Safety Net	Safe communication between control stations.

		控制站之间的安全通讯。
6.	Control station 控制站	A set of triple redundant control system, which includes triple redundant PMs and IO modules under control. 一套三冗余的控制系统，包含三冗余 PM 和 PM 控制的各种 IO 模块。
7.	System response time 系统响应时间	Time interval from the moment that transition of demand signal generated at input ETP to the moment that transition of response signal generated at output ETP. 从系统输入端子板上产生需求信号跳变的时刻到输出端子板上产生相应的响应信号跳变之间的时间。
8.	Control cycle 控制周期	Time interval between adjacent two runs of user program execution. PM 两次执行用户程序间隔时间。
9.	Project 工程	Files which contain configuration information for control station and generated by IEC 61131 configuration software. These files contain all the information required by control station to implement control, including user control program (binaries) to be loaded and executed as well as configuration information of task, CM, PM and IO modules. IEC 61131 组态软件在完成编译后，为控制站生成的组态信息文件，该文件包含可加载执行的用户控制程序（二进制程序）、任务配置信息、CM 配置信息、PM 配置信息和 IO 模块配置信息等各种控制站完成控制所需的信息。
10.	Source project 源工程文件	Source file of the project before compiling. 工程在编译前的源文件。
11.	User program 用户程序	Part of project which contain user control program (binaries) to be loaded and executed and configuration information of task. 工程中的一部分：可加载执行的用户控制程序（二进制程序）和任务配置信息。

1.3.2 Abbreviations 缩略语

Table 1-2 Abbreviations

表 1-2 缩略语

No. 序号	Abbreviation 缩略语	English description 英文	Chinese description 中文
1.	PM	Processor Module	主处理器模块
2.	CM	Communication Module	通讯模块
3.	BI	Bus Interface Module	总线接口模块
4.	AI	Analog Input Module	模拟量输入模块
5.	AO	Analog Output Module	模拟量输出模块

6.	DI	Digital Input Module	数字量输入模块
7.	DO	Digital Output Module	数字量输出模块
8.	OSP	Over Speed Protect Module	超速保护模块
9.	SOE	Sequence Of Events	SOE 事件
10.	SIL	Safety Integrity Level	安全完整等级
11.	PW	Power Module	电源模块
12.	OPC	OLE for Process Control	用于过程控制的对象链接与嵌入式技术
13.	UP	User Program	用户程序

2 Module overview 模块概述

The location of the project configuration module (marked red) in the software hierarchy is shown below.

工程配置模块（标红）在软件层次中的位置如下图所示。

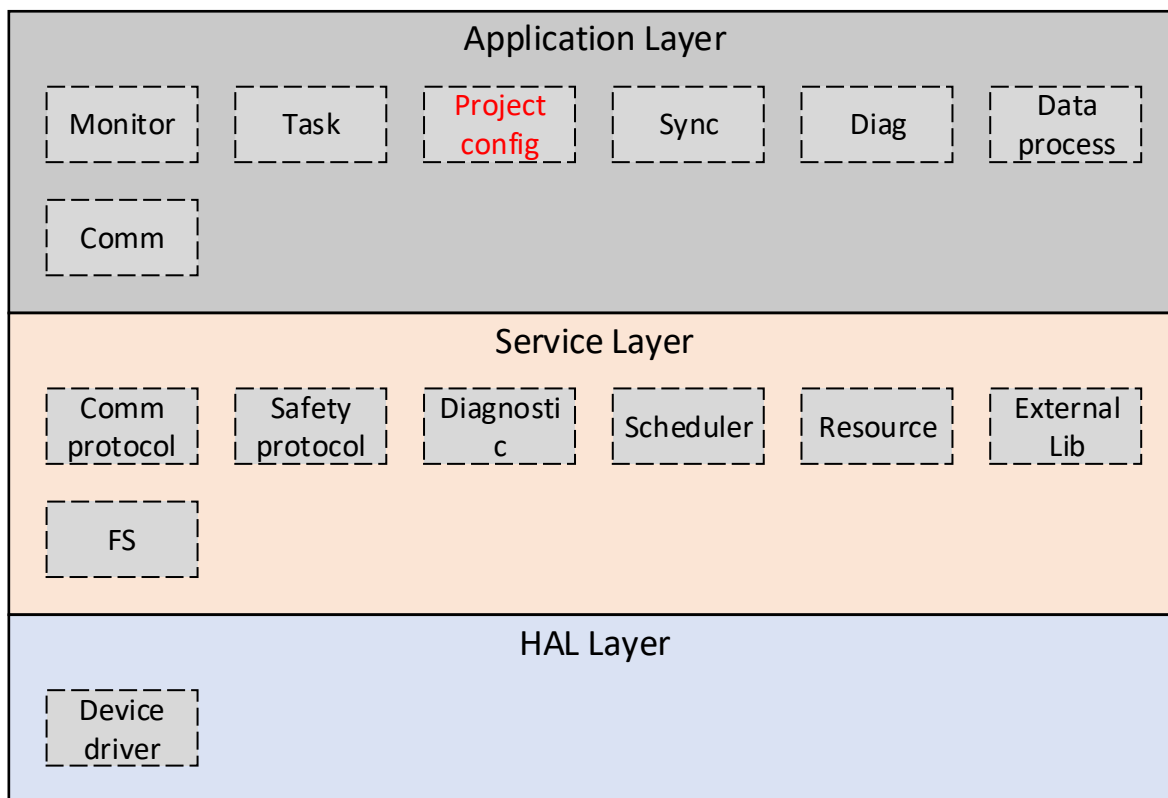


Figure 2-1 the location of the project configuration module

图 2-1 模块位置

Project configuration module is used to parse user project configuration information and then configure the information.

工程配置模块主要负责工程配置信息的解析和配置工作。

3 Module design 模块设计

3.1 Function description 功能描述

The user project is downloaded to the PM as project files. These project files contains: PRG file, IO configuration file, control station configuration file, Vote file, SOE file, Modbus file, P2P file, CM configuration file, OPC file and Retain file. The project configuration module completes the parsing and configuration of these files in three cases: power-on configuration, initial download configuration, online download configuration.

用户工程以工程文件的形式，下装到 PM 中。工程文件包含：PRG 文件、IO 配置文件、控制站配置文件、Vote 文件、SOE 文件、Modbus 文件、P2P 文件、CM 配置文件、OPC 文件和 Retain 文件。工程配置模块完成这些文件的解析配置工作。工程配置模块主要完成以下三种情况下的工程配置工作：上电配置，全下装配置，在线下装配置。

The functions of the project configuration module include: initializing the parameters of the module, analyzing the project file information, and updating the system state after the configuration.

工程配置模块的功能主要包括：初始化相关模块的参数，解析工程文件信息，并在配置完成后更新系统状态。

3.2 Design concept 设计思路

3.2.1 Power-on initialization 上电初始化

The power-on initialization of the project configuration module completes two jobs: the initialization of the data structure and the reading of the existing project file.

工程配置模块的上电初始化主要完成两部分工作：数据结构的初始化和已有工程文件的读取。

When the PM is powered on, if the project is valid, the project files shall be read first, and then the files are parsed, and the configuration information is stored in the system resources management module with a specified format. If the key switch is in the 'init' position, the project is invalid.

当 PM 上电时，如果工程有效，首先完成工程文件的读取，然后对工程文件进行解析，按照规定格式将配置信息存储在系统资源管理模块内。如果钥匙开关在 init 位置，则工程无效。

3.2.1.1 Initialization of the data structure 数据结构初始化

After power-on, first initialize the data structure used by this module. The detailed data structure is shown in chapter 3.5.

上电时，要对工程配置模块使用的数据结构进行初始化，具体数据结构参见章节 3.5。

3.2.1.2 Read project file after power-on 工程文件上电读取

There are three file types for each file in the PM, the meaning is as follow (take PRG file as an example):

1. .PRG: formal file. The current used formal PRG file.
2. .PRG_TMP: temporary file. Used for temporary storage during the download process.
3. .PRG_OLD: old file for the download process. When the new PRG file is valid, the last PRG file will be renamed to old file name, and then the temporary file will be renamed to the formal file name.

每一个文件在 PM 中存在三种文件名，以 PRG 文件为例，各自的含义如下：

1. .PRG：正式文件。当前使用的正式 PRG 文件。
2. .PRG_TMP：临时文件。用于下装过程中临时存放的文件。
3. .PRG_OLD：旧文件。用于下装过程中，当新下装的工程生效时，就得工程文件改为旧文件名，然后将下装的临时文件更名为正式文件名。

After power-on, determines whether the project file is exist, and the length and CRC of the file is correct. When a formal file, old file, or more than one temporary file of the same project files are valid, it is necessary to select the file. File selection follows the priority rule: formal file > temporary file > old file.

上电时，首先判断 Flash 中的工程文件是否存在，长度和 CRC 是否正确。当同一个工程文件的正式文件，旧文件，临时文件多于一个有效时，需要进行文件的选取。文件选取遵循优先级原则：正式文件>临时文件>旧文件。

If all project files are valid, the project is considered valid. The valid project will be read out from the Flash and written into the corresponding memory area, and then the corresponding file update flag will be updated for the periodic processing function.

当所有工程文件都有效时，认为工程有效。此时，将各工程文件从 Flash 中读取出来，放入对应内存区，更新对应文件更新标志，供周期运行时解析。

3.2.2 Cycle operation 周期运行

When the configuration software downloads the user project file to PM (initial download or online download), the project configuration module parses the new project file and stores the configuration information in the system resource manager module according to the specified format.

当组态软件将用户工程文件下载到 PM 的文件存储区中后（全下装或者在线下装），工程配置模块解析新的工程文件，并按照规定格式将配置信息存储在系统资源内。

3.3 Interface function 接口函数

The interface functions which is provided by this module is shown as follows:

模块提供的接口函数如下:

1. void ConfigInit (void)

Input argument 输入参数	Output argument 输出参数	Description 描述
No. 无。	No. 无。	Module initialization. 模块初始化。

2. void ConfigCycle (void)

Input argument 输入参数	Output argument 输出参数	Description 描述
No. 无。	No. 无。	Module periodic processing function. 配置模块周期处理函数

3.4 Global variable 全局变量

Table 3-1 Global variable list

表 3-1 全局变量列表

No. 序号	Type 变量类型	Name 名称	Description 描述
1.	SysCfgFileTable_t	s_pstSysCfgFileTable	Record the file information which need to be uploaded to CM 记录需要上传 CM 的文件信 息

3.5 Data structure 数据结构

3.5.1 IO configuration information IO配置信息

1. IO master structure IO 主卡结构体定义

```
typedef struct CfgPbMasterTag
```

```
{
```

```
    uint8_t ucAutoStart;
```

```
    IP_BUS_t stDpmBus;
```

```
    uint8_t ucStationNumber;
```

```
    uint8_t ucNumOfSlaves;
```

```
    uint8_t ucSlaveStateParam[16];
```

```
}CfgPbMaster_t;
```

2. IP_BUS parameter structure IP_BUS 参数结构体定义

```
typedef struct IP_BUS_TTag  
{  
    uint16_t usBus_Para_Len;  
    uint8_t ucFDL_Add;  
    uint8_t ucBaudrate;  
    uint16_t usTSL;  
    uint16_t usMin_TSDR;  
    uint16_t usMax_TSDR;  
    uint8_t ucTQUI;  
    uint8_t ucTSET;  
    uint32_t uiTTR;  
    uint8_t ucG;  
    uint8_t ucHSA;  
    uint8_t ucMax_Retry_Limit;  
    uint8_t ucBp_Flag;  
    uint16_t usMin_Slave_Intervall;  
    uint16_t usPoll_Timeout;  
    uint16_t usData_Control_Time;  
    uint8_t ucOctet[6];  
    uint16_t usMaster_User_Data_Len;  
    char_t cMaster_Class2_Name[32];  
    uint8_t ucMaster_User_Data[MAX_BUS_USER_DATA_LEN];  
}IP_BUS_t;
```

3. IO module information structure IO 模块信息结构体定义

```
typedef struct SysIoModTag  
{  
    SysIoParamArea_t stSysIoParamArea;  
    SysIoOffsetTable_t stSysIoOffsetTable;  
}SysIoMod_t;
```

4. IO module mapping table structure IO 模块映射表结构体定义

```
typedef struct SysIoOffsetTableTag  
{
```

```
uint16_t usInDevNum;  
uint16_t usOutDevNum;  
SysIoDevOffset_t stInDevOffset[CFG_MAX_INDEV];  
SysIoDevOffset_t stOutDevOffset[CFG_MAX_OUTDEV];  
}SysIoOffsetTable_t;
```

5. device mapping table structure 设备映射表结构体定义

```
typedef struct SysIoDevOffsetTag  
{  
    uint8_t ucModAddr;  
    uint8_t ucReserve;  
    uint16_t usModIdent;  
    uint8_t ucIecSize;  
    uint16_t usDramSize;  
    uint32_t uiDualRamOffset;  
    uint16_t usIecOffset;  
}SysIoDevOffset_t;
```

3.5.2 PRG configuration information PRG配置信息

6. PRG structure PRG 信息结构体定义

```
typedef struct LogicConfigInfoTag  
{  
    uint32_t uiDataAreaUsedSize[MAX_DATA_AREA_NUM];  
    uint32_t uiMachineCodeLen;  
    uint32_t uiMaxPous;  
    uint32_t uiMainPrgEntry;  
    uint32_t uiGlobalInitEntry;  
    uint32_t uiCodeInitEntry;  
    uint32_t uiLoadAddress;  
}LogicConfigInfo;
```

3.5.3 Project configuration information 工程配置信息

7. Project Configuration Information structure 工程信息结构体定义

```
typedef struct PrjConfigInfoTag  
{  
    uint32_t uiProjectID;  
    uint32_t uiMajorVersion;  
    uint32_t uiMinorVersion;
```

```
uint8_t ucProjectName[32];  
uint32_t uiMachineCodeLen;  
uint32_t uiLastSaveDate;  
TaskConfigInfo stTaskConfigInfo[2];  
}PrjConfigInfo;
```

8. Task Configuration Information structure Task 信息结构体定义

```
typedef struct TaskConfigInfoTag  
{  
    uint32_t uiTaskID;  
    uint32_t uiControlFlag;  
    uint32_t uiTolerateFlag;  
    uint32_t uiPriority;  
    uint32_t uiIECPeriod;  
    uint32_t uiTaskCodeEntry;  
}TaskConfigInfo;
```

3.5.4 P2P configuration information P2P配置信息

9. P2P configuration Information structure P2P 配置信息结构体定义

```
typedef struct P2PConfigInfoTag  
{  
    uint32_t uiUpdateFlag;  
    P2PConfigHead stP2PConfigHead;  
    P2PBlock stP2PRecvBlockInfo[64];  
    P2PBlock stP2PSendBlockInfo[64];  
}P2PConfigInfo;
```

10. P2P header configuration Information structure P2P 信息头结构体定义

```
typedef struct P2PConfigHeadTag  
{  
    uint32_t uiBlockNum;  
}P2PConfigHead;
```

11. P2P block configuration Information structure P2P 块结构体定义

```
typedef struct P2PBlockTag  
{  
    uint32_t uiUpdateFlag;  
    uint32_t uiActive;
```

```
P2PBlockHead stP2PBlockHead;  
P2PVar stP2PVar[500];  
}P2PBlock;
```

12. P2P block header configuration Information structure P2P 块信息头结构体定义

```
typedef struct P2PBlockHeadTag  
{  
    uint32_t uiBlockID;  
    uint16_t usNetID;  
    uint32_t uiTargetStationID;  
    uint32_t uiTargetStationIP1;  
    uint32_t uiTargetStationIP2;  
    uint32_t uiPeriod;  
    uint32_t uiP2PVariableNum;  
}P2PBlockHead;
```

13. P2P variable configuration Information structure 变量结构体定义

```
typedef struct P2PVarTag  
{  
    uint16_t usRefID;  
    uint16_t usVarSize;  
    uint32_t uiOffset;  
    uint32_t uiSafetyValue;  
}P2PVar;
```

3.5.5 Modbus slave configuration Information Modbus 从站配置信息

14. Modbus slave configuration information structure Modbus Slave 站配置信息结构体定义

```
typedef struct ModbusSlaveInfoTag  
{  
    uint32_t uiUpdateFlag;  
    ModbusSlaveHead stModbusSlaveHead;  
    ModbusSlaveBlock stModbusSlaveBlockInfo[64];  
}ModbusSlaveInfo;
```

15.

```
typedef struct ModbusSlaveHead Tag  
{  
    uint32_t uiBlockNum;
```



```
}ModbusSlaveHead;
```

16.

```
typedef struct ModbusSlaveBlockTag  
{  
    uint32_t uiUpdateFlag;  
    uint32_t uiActive;  
    ModbusSlaveBlockHead stModbusSlaveBlockHead;  
    ModbusVar stModbusVar[500];  
}ModbusSlaveBlock;
```

17.

```
typedef struct ModbusSlaveBlockHeadTag  
{  
    uint32_t uiBlockID;  
    uint16_t usDataType;  
    uint16_t usPermission;  
    uint32_t uiBeginAddr;  
    uint32_t uiEndAddr;  
    uint32_t uiModbusVariableNum;  
}ModbusSlaveBlockHead;
```

18.

```
typedef struct P2PVarTag  
{  
    uint16_t usRefID;  
    uint16_t usVarSize;  
    uint32_t uiOffset;  
    uint32_t uiModbusIndex;  
}ModbusVar;
```

3.5.6 Modbus master configuration information Modbus主站配置信息

19. Modbus master configuration information structure Modbus Master 站配置信息结构体定义

```
typedef struct ModbusMasterInfoTag  
{  
    uint32_t uiUpdateFlag;  
    ModbusMasterConfig st ModbusMasterConfig[4];  
}ModbusMasterInfo;
```

20.

```
typedef struct ModbusMasterConfigTag
{
    uint32_t uiUpdateFlag;
    uint32_t uiActive;
    ModbusMasterPort stModbusMasterPort[6];
}ModbusMasterConfig;
```

21.

```
typedef struct ModbusMasterConfigTag
{
    uint32_t uiUpdateFlag;
    uint32_t uiActive;
    ModbusMasterPortHead stModbusMasterPortHead;
    ModbusMasterBlock stModbusMasterBlock[64];
}ModbusMasterConfig;
```

22.

```
typedef struct  ModbusMasterPortHeadTag
{
    int32_t uiBlockNum;
}ModbusMasterPortHead;
```

23.

```
typedef struct ModbusMasterBlockTag
{
    uint32_t uiUpdateFlag;
    uint32_t uiActive;
    ModbusMasterBlockHead stModbusMasterBlockHead;
    ModbusVar stModbusVar[500];
}ModbusMasterBlock;
```

24.

```
typedef struct ModbusMasterBlockHeadTag
{
    uint32_t uiBlockID;
```

```
uint16_t usVisitMode;  
uint32_t uiTargetStationID;  
uint32_t uiTargetIP;  
uint32_t uiTargetRegion;;  
uint32_t uiTargetBeginAddr;  
uint32_t uiTargetSize;  
uint32_t uiModbusVariableNum;  
}ModbusMasterBlockHead;
```

3.5.7 RETAIN configuration information RETAIN配置信息

25. Retain information structure Retain 信息结构体定义

```
typedef struct RetainVarInfoTag  
{  
    uint32_t uiUpdateFlag;  
    uint32_t uiVarNum;  
    RetainVar stRetainVar[500];  
}RetainVarInfo;
```

26.

```
typedef struct RetainVarTag  
{  
    uint16_t usRefID;  
    uint16_t usVarSize;  
    uint32_t uiOffset;  
}RetainVar;
```

3.5.8 OPC configuration information OPC配置信息

27. OPC information structure OPC 信息结构体定义*/

```
typedef struct OPCVarInfoTag  
{  
    uint32_t uiUpdateFlag;  
    uint32_t uiVarNum;  
    OPCVar stOPCVar[10000];  
} OPCVarInfo;
```

28.

```
typedef struct OPCVarTag  
{
```

```
uint16_t usRefID;  
uint16_t usVarSize;  
uint32_t uiOffset;  
} OPCVar;
```

3.5.9 SOE configuration information SOE配置信息

29. SOE configuration information structure SOE 配置信息结构体定义

```
typedef struct SOEConfigInfoTag  
{  
    uint32_t uiUpdateFlag;  
    uint16_t usHW_SOEBlockNum;  
    uint16_t usSW_SOEBlockNum;  
    SOE_HW_Block stSOE_HW_Block[16];  
    SOE_SW_Block stSOE_SW_Block[2][16];  
    ModuleSOEInfo stModuleSOEInfo[MAX_MOD_NUM];  
} SOEConfigInfo;
```

30.

```
typedef struct SOE_HW_BlockTag  
{  
    uint32_t uiBlockID;  
    uint16_t usBlockType;  
    uint32_t uiMaxRecordCount;  
} SOE_HW_Block;
```

31.

```
typedef struct ModuleSOEInfoTag  
{  
    uint32_t uiActive;  
    uint32_t uiSOE_HW_VarNum;  
    SOE_HW_Var stSOE_HW_Var[32];  
}ModuleSOEInfo;
```

32.

```
typedef struct SOE_HW_VarTag  
{  
    uint32_t uiBlockID;  
    uint32_t uiChannelID;
```

```
}SOE_HW_Var;
```

33.

```
typedef struct SOE_SW_BlockTag
{
    uint32_t uiBlockID;
    uint16_t usBlockType;
    uint32_t uiMaxRecordCount;
    uint32_t ui SOE_SW_VarNum;
    SOE_SW_Var st SOE_SW_Var[MAX_SOE_VAR_NUM];
}SOE_SW_Block;
```

34.

```
typedef struct SOE_SW_VarTag
{
    uint16_t usRefID;
    uint16_t usVarSize;
    uint32_t uiOffset;
}SOE_SW_Var;
```

3.5.10 VOTE configuration information VOTE配置信息

35. VOTE configuration information structure VOTE 信息结构体定义

```
typedef struct VOTEConfigInfoTag
{
    uint32_t uiUpdateFlag;
    uint32_t uiModNum;
    VoteModInfo stVoteModInfo[MAX_MOD_NUM];
}VOTEConfigInfo;
```

36.

```
typedef struct VoteModInfoTag
{
    uint32_t uiUpdateFlag;
    uint32_t uiActive;
    uint16_t usModType;
    uint32_t uiBitSizeInput;
    uint32_t uiBitOffsetInput;
    uint32_t uiBitSizeOutput;
```

```
uint32_t uiBitOffsetOutput;  
uint16_t usChannelCount;  
VoteChannelInfo stVoteChannelInfo[MAX_CHANNEL_NUM];  
}VoteModInfo;
```

37.

```
typedef struct VoteChannelInfoTag  
{  
uint8_t ucChannelType;  
uint8_t ucRefID;  
uint8_t ucVoteType;  
uint32_t uiBitOffset;  
uint32_t uiSafetyValue;  
} VoteChannelInfo;
```

3.6 List of sub-function 子功能列表

The sub-functions list is shown as follows:

子功能列表如下。

Table 3-2 Sub function list

表 3-2 子功能列表

Sub function No. 子功能编号	Function description 功能描述
SWDD-PM-CFG_NSafR_SecR_A_001	Module initialization 模块初始化
SWDD-PM-CFG_NSafR_SecR_A_002	Configuration information initialization 配置信息初始化
SWDD-PM-CFG_NSafR_SecR_A_003	Module cycle function 模块周期运行函数

4 Design of sub-function 子功能设计

4.1 Module initialization 模块初始化

SWDD-PM-CFG_NSafR_SecR_A_001

4.1.1 ConfigInit

4.1.1.1 Function Description 功能描述

This function completes initialization of module.

该函数完成数据处理模块的初始化。

4.1.1.2 Argument Description 参数说明

- Definition 函数定义

void ConfigInit(void)

- Input argument 输入参数

No.

无。

- Output argument 输出参数

No.

无。

4.1.1.3 Processing flow 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.2 Configuration information initialization 配置信息初始化

SWDD-PM-CFG_NSafR_SecR_A_002

4.2.1 ConfigInfoInit

4.2.1.1 Function Description 功能描述

This function completes configuration information initialization.

该函数完成配置信息的初始化。

4.2.1.2 Argument Description 参数说明

- Function Definition 函数定义

void ConfigInfoInit(void)

- Input argument 输入参数

No.

无。

- Output argument 输出参数

No.

无。

4.2.1.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.3 Module cycle function 模块周期运行

SWDD-PM-CFG_NSafR_SecR_A_003

4.3.1 ConfigCycle

4.3.1.1 Function Description 功能描述

This function is used to parse project file.

本函数用于解析工程文件。

4.3.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

void ConfigCycle(void)

➤ Input argument 输入参数

No.

无。

➤ Output argument 输出参数

No.

无。

4.3.1.3 处理流程

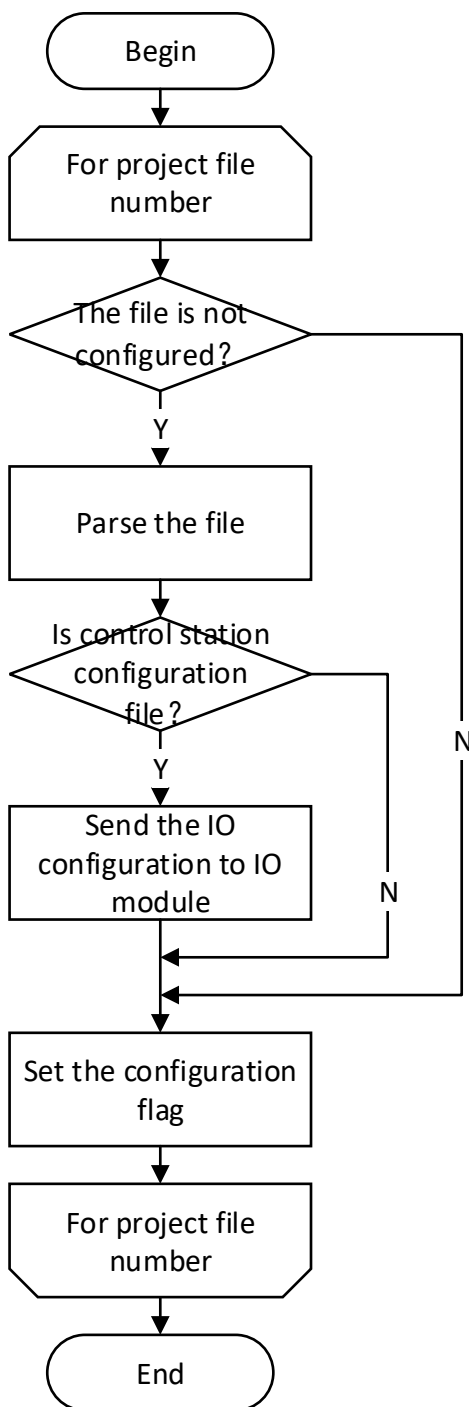


Figure 4-1 parse project file processing flow

图 4-1 文件解析流程图

4.3.2 ConfigPRGFileParse

4.3.2.1 Function Description 功能描述

This function is used to parse PRG file. The parse process of all files are the same and not repeated here.

本函数用于解析 PRG 文件。所有文件的解析流程基本相同，因此其他文件的处理流程不再赘述。

4.3.2.2 Argument Description 参数说明

- Function Definition 函数定义

static int8_t ConfigPRGFileParse(uint8_t ucPRGfileAddr[])

- Input argument 输入参数

ucPRGfileAddr: start address of PRG Data, PRG 数据起始地址

- Output argument 输出参数

Result of file parsing, 文件解析结果

4.3.2.3 处理流程

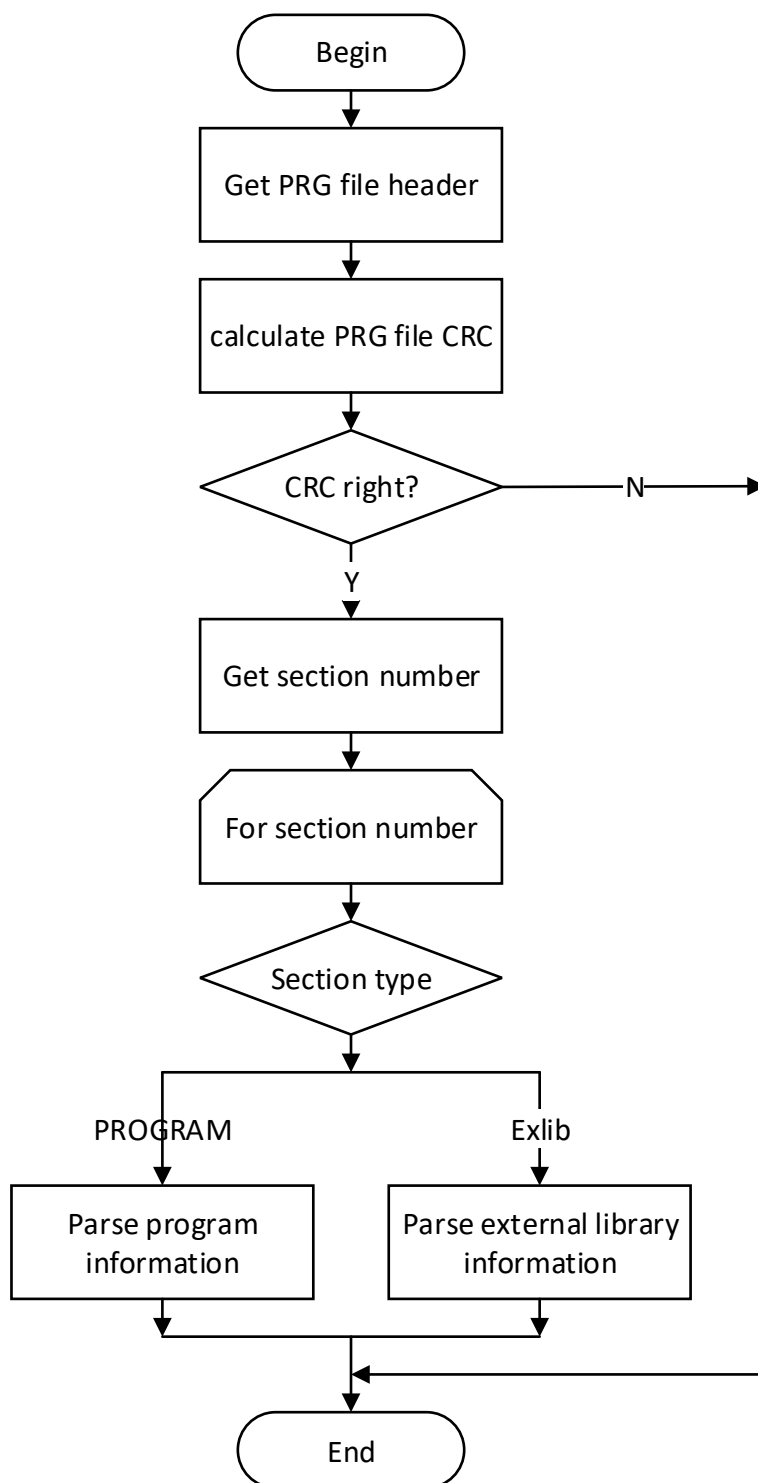


Figure 4-2 PRG file parsing processing flow

图 4-2 PRG 文件解析流程图