Document Title: PM_FW monitor module design description of Safety Control System

Document Number: 17-Q04-000523

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: Li Qi 2016-12-26

Checked by: Zhu Genghua 2016-12-30

Countersigned by: Liu Yang, 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	2016-12-26	None	A	Li Qi	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 监视模块设计说明书

文件编号：17-Q04-000523

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：李琦 2016 年 12 月 26 日

审核：朱耿华 2016 年 12 月 30 日

会签：刘阳、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	2016-12-30		A	李琦	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 Introduction 综述

This document describes the design description of monitor 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 监视模块的设计方案。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 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
14.	PPM	Pulse per minute	分钟脉冲

2 Module overview 模块概述

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

监视模块（标红）在软件层次中的位置如下图所示。

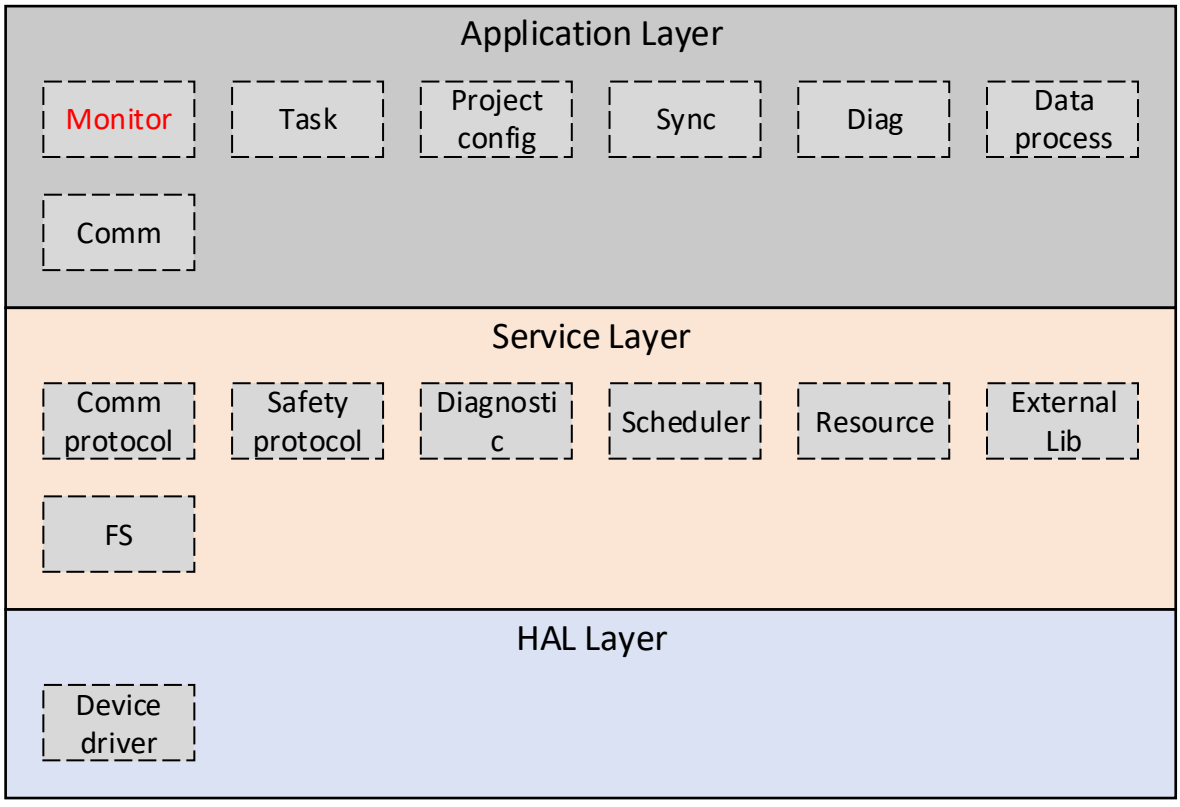

Figure 2-1 the location of the module

图 2-1 模块位置

3 Module design 模块设计

3.1 Function description 功能描述

1) Core0 monitor module

1) Core0 监视模块

This module is used to control the user program run/stop etc. this module is also used to monitor control commands which come from core1;

该模块主要负责控制用户程序运行/停止等以及周期性监视来自 core1 的服务数据;

The service data will be processed in this module; the service data include control commands of run, stop, single-step or pause etc., and write variable operations which come from OPC Server, Modbus, CS1131 software (Configuration software) and P2P.

对来自 core1 的服务数据进行处理；这些服务数据主要有：运行、停止、单步、暂停等控制命令以及 OPC Server、Modbus、CS1131（组态软件）及 P2P 的写变量操作。

2) Core1 monitor module

2) Core1 监视模块

Get local PM address;

获取本地 PM 地址；

Get system temperature;

获取系统温度；

Get power supply alarm bit;

获取电源报警；

Get PMA/B/C state;

获取三系 PM 状态；

Get PPM state;

获取 PPM 状态；

Get MCU state;

获取 MCU 状态；

Set PM state;

设置 PM 的状态。

3.2 Design concept 设计思路

3.2.1 Monitor (core0) 监视模块（core0）

Receive core1 data via access shared memory (write variables, force variable, Modbus read/write variables, task start/stop, OPC read/write variables), and send response to core1 via shared memory;

通过访问共享内存接收来自 core1 的数据（组态软件 写变量、强制变量、Modbus 写变量、任务的启停控制命令、OPC 写变量、），向 core1 发送应答；

External library and real time data initialization;

外部库及实时数据区的初始化；

Backup and recovery the retained data in SRAM.

备份及恢复掉电保持数据。

3.2.2 Monitor (core1) 监视模块（core1）

Core1 monitors：Temperature, power supply voltages, local PM's ID, PM's online state, PPM state and MCU state.

Core1 主要监视：温度、电源电压、本地 PM 的 ID 号、三系 PM 的在线信息、PPM 状态、MCU 状态

3.3 Interface function 接口函数

The interface functions which is provided by this module (focus on core0) is shown as follows:

1. void MonitorInit(void)

Input argument 输入参数	Output argument 输出参数	Description 描述
No. 无。	No. 无。	Monitor module initialization. 监视模块初始化。

2. void MonitorCycle(void)

Input argument 输入参数	Output argument 输出参数	Description 描述
No. 无。	No. 无。	Monitor module runs periodically. 监视模块周期运行。

The interface functions which is provided by this module (focus on core1) is shown as follows:

3. void MonitorInit(void)

Input argument 输入参数	Output argument 输出参数	Description 描述
No. 无。	No. 无。	Monitor module initialization. 监视模块初始化。

4. void MonitorCycle(void)

Input argument 输入参数	Output argument 输出参数	Description 描述
No. 无。	No. 无。	Monitor module runs periodically. 监视模块周期运行。

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.	WriteVarArray_t	s_stWriteVarArray [MAX_TASK_NUM]	Write variables array 写变量表
2.	ForceItemArray_t	s_stForceItemArray [MAX_TASK_NUM]	Force item array 强制项列表

3.5 Data structure 数据结构

The following data structure is used by core0.

```
typedef void (*GlobaInit)(uint32_t);
#pragma pack (1)
typedef struct MSGHeadTag
{
    uint8_t ucType;                 /*0xAA*/
    uint8_t ucProtect;              /*0xBB*/
    uint16_t usActualType;          /*Actually message type*/
    uint8_t ucNodeNum;              /*control station node number*/
    uint8_t ucPMNum;
```

```
    uint16_t usSequenceNum;              /*request message sequence number*/
    uint8_t ucLastFlag;                  /*last package flag*/
    uint8_t ucReserved[5];
    uint16_t usLength;                   /*message data length (head + Data Area)*/
}MSGHead_t;

typedef struct ACKHeadTag
{
    uint8_t ucType;                      /*0xBB*/
    uint8_t ucProtect;                   /*0xAA*/
    uint16_t usActualType;               /*Actually message type*/
    uint8_t ucNodeNum;                   /*control station node number*/
    uint16_t usSequenceNum;              /*request message sequence number*/
    uint8_t ucFlag;                      /*indicate the last frame or not 用于指示应答的第一或最后一帧*/
                                         /*0：多消息应答的中间帧; 1：多消息应答的第一帧; 2：多消息应答的最后一帧; 3：单消息应答*/
    uint8_t ucResCode;                   /*responsed code 响应码：0-请求已成功完成；255-请求失败，失败原因见 subReason*/
    uint8_t ucSubReason;                 /*failed reason 请求失败的具体原因，详见本章响应码*/
    uint8_t ucDataFlg;                   /*responsed data 是否存在应答数据：1-存在；0-不存在*/
    uint8_t ucReserved[3];
    uint16_t usLength;                   /*ACK data length (head + Data Area)*/
}ACKHead_t;

typedef struct WriteVarTag
{
```

```
    uint8_t ucTaskID;
    uint8_t ucRefID;
    uint16_t usSize;
    uint32_t uiOffSet;
}WriteVar_t;

typedef struct WriteVarArrayTag
{
    uint32_t uiReadIndex;
    uint32_t uiWriteIndex;
    uint8_t ucWriteVar[MONITOR_WRITE_BUF_LEN];
}WriteVarArray_t;

typedef struct ForceItemTag
{
    uint8_t ucTaskID;
    uint8_t ucRefID;
    uint16_t usSize;
    uint32_t uiOffSet;
}ForceItem_t;

typedef struct ForceItemArrayTag
{
    uint32_t uiReadIndex;
    uint32_t uiWriteIndex;
    uint8_t ucForceItem[2*1024];
}ForceItemArray_t;
```

```
typedef struct OPCWritVarHeadTag
{
    uint16_t usOPCWritVarNum;
    uint16_t usReserved;
}OPCWritVarHead_t;

typedef struct CommModbusCommonHeadTag
{
    uint8_t ucPortType;              /*port type  端口类型  */
    uint8_t ucReserve;               /*reserved  预留  */
    uint16_t usDataLen;              /*Modbus data length Modbus 数据的长度（字节）  */
}CommModbusCommonHead_t;

typedef struct CommModbusMasterHeadTag
{
    CommModbusCommonHead_t stCommonHead;
    uint8_t ucCMSlotID;                  /*CM slot ID CM 模块槽位号  */
    uint8_t ucPortID;                    /*Port ID  端口号  */
    uint8_t ucReserve1[2];               /*reserved  预留  */
    uint32_t uiBlockID;                  /*Modbus master block ID Modbus Master 块
号  */
    uint32_t uiProjID;                   /*Project ID  工程 ID */
    uint32_t uiMainVer;                  /*Main version  主版本  */
    uint32_t uiMinorVer;                 /*Minor version  次版本  */
    uint32_t uiProjCRC;                  /*Project CRC  工程 CRC */
}CommModbusMasterHead_t;

typedef struct CommModbusSlaveHeadTag
{
```

```
    CommModbusCommonHead_t stCommonHead;
    uint8_t ucReserve1[4];                  /*reserved 预留 */
    uint32_t uiBlockID;                     /*Modbus slave block ID Modbus Slave 块号
*/
    uint16_t  usBeginAddr;                  /*start address 本数据帧中数据的起始地址
*/
    uint16_t usElementNum;                  /*coil/register number 线圈/寄存器数 */
    uint8_t ucReserve2[12];                 /*reserved 预留 */
}CommModbusSlaveHead_t;

typedef struct P2PRecvReqHeadTag
{
    uint8_t ucSrcStaID;
    uint8_t ucDstStaID;
    uint8_t ucFrameType;
    uint8_t ucLinkID;   /* Link ID   链路 ID: 1-链路 1；2-链路 2。只对同 CM 双网口冗余有效
*/
    uint16_t usTotalLen;
    uint8_t ucPMInfo;
    uint8_t ucReserve1;
}P2PRecvReqHead_t;

typedef struct P2PSendRspHeadTag
{
    uint8_t ucSrcCMID;
    uint8_t ucSrcNETID;
    uint8_t ucReserve1;
    uint8_t ucReserve2;
    uint32_t uiDstIP1Addr;
```

```
    uint32_t uiDstIP2Addr;

    uint8_t ucSrcStaID;

    uint8_t ucDstStaID;

    uint8_t ucFrameType;

    uint8_t ucReserve3;

    uint32_t uiTotalLen;

}P2PSendRspHead_t;

#pragma pack ()
```

3.6 List of sub-function 子功能列表

The sub-functions list (focus on core0) is shown as follows:

Core0 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-MM_SafR_NSecR_A_001	Core0 监视模块初始化 Core0 monitor module initialization
SWDD-PM-MM_SafR_NSecR_A_002	Core0 监视模块周期运行 Core0 monitor module runs periodically
SWDD-PM-MM_SafR_NSecR_A_003	服务数据处理 Service data handling
SWDD-PM-MM_SafR_NSecR_A_004	用户工程控制（启动、停止、暂停、单步等） User project control (run/stop/pause/single etc.)

The sub-functions list (focus on core1) is shown as follows:

Core1 子功能列表如下。

Table 3-3 Sub function list

表 3-3 子功能列表

Sub function No. 子功能编号	Function description 功能描述
SWDD-PM-MM_SafR_NSecR_A_005	Core1 监视模块初始化

	Core1 monitor module initialization
SWDD-PM-MM_SafR_NSecR_A_006	Core1 监视模块周期运行 Core1 monitor module runs periodically

4 Design of sub-function 子功能设计

4.1 Monitor module initialization (Core0) 监视模块初始化（core0）

SWDD-PM-MM_SafR_NSecR_A_001

4.1.1 MonitorInit

4.1.1.1 Function Description 功能描述

Monitor module initialization.

监视模块初始化。

4.1.1.2 Argument Description 参数说明

- Definition 函数定义

void MonitorInit(void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.1.1.3 Processing flow 处理流程

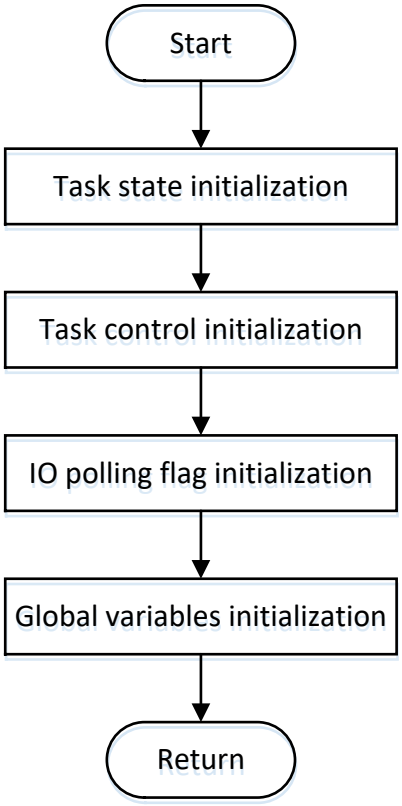

Figure 4-1 Monitor module initialization

图 4-1 监视模块初始化

4.2 Monitor module cycle (Core0) 监视模块周期运行（core0）

SWDD-PM-MM_SafR_NSecR_A_002

4.2.1 MonitorCycle

4.2.1.1 Function Description 功能描述

Monitor module runs periodically.

监视模块周期运行。

4.2.1.2 Argument Description 参数说明

- Definition 函数定义

void MonitorCycle(void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.2.1.3 Processing flow 处理流程

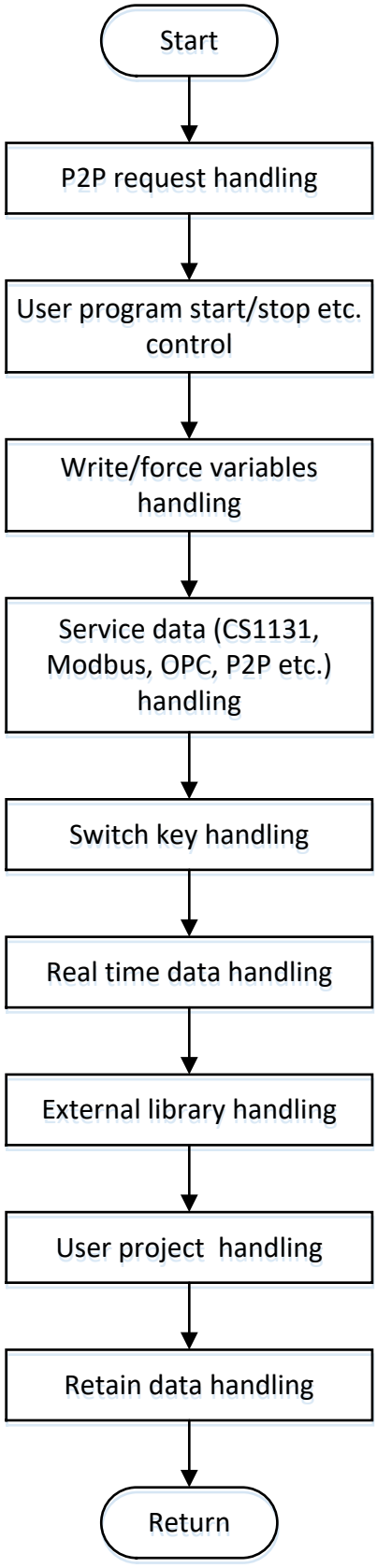

Figure 4-2 Monitor module cycle

图 4-2 监视模块周期运行

4.3 Service data handling (Core0) 服务数据处理（core0）

SWDD-PM-MM_SafR_NSecR_A_003

4.3.1 MonitorServicPreprocess

4.3.1.1 Function Description 功能描述

Service data handling.

服务数据处理。

4.3.1.2 Argument Description 参数说明

- Definition 函数定义

static void MonitorServicPreprocess(void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.3.1.3 Processing flow 处理流程

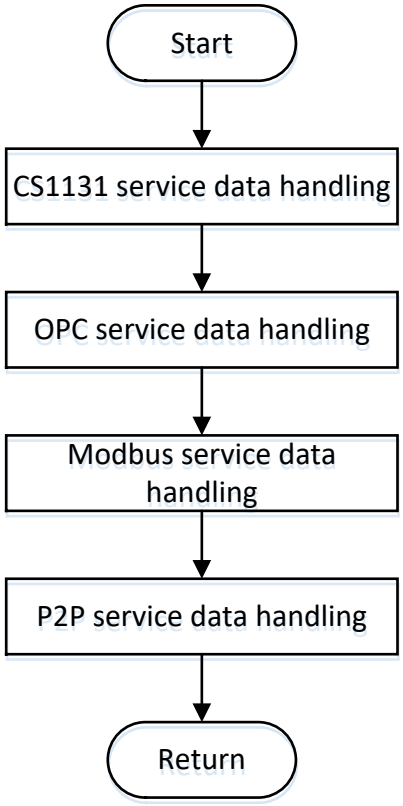

Figure 4-3 Service data handling

图 4-3 服务数据处理

4.3.2 MonitorCS1131Preprocess

4.3.2.1 Function Description 功能描述

CS1131 service data handling.

CS1131 服务数据处理。

4.3.2.2 Argument Description 参数说明

- Definition 函数定义

static void MonitorCS1131Preprocess(void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.3.2.3 Processing flow 处理流程

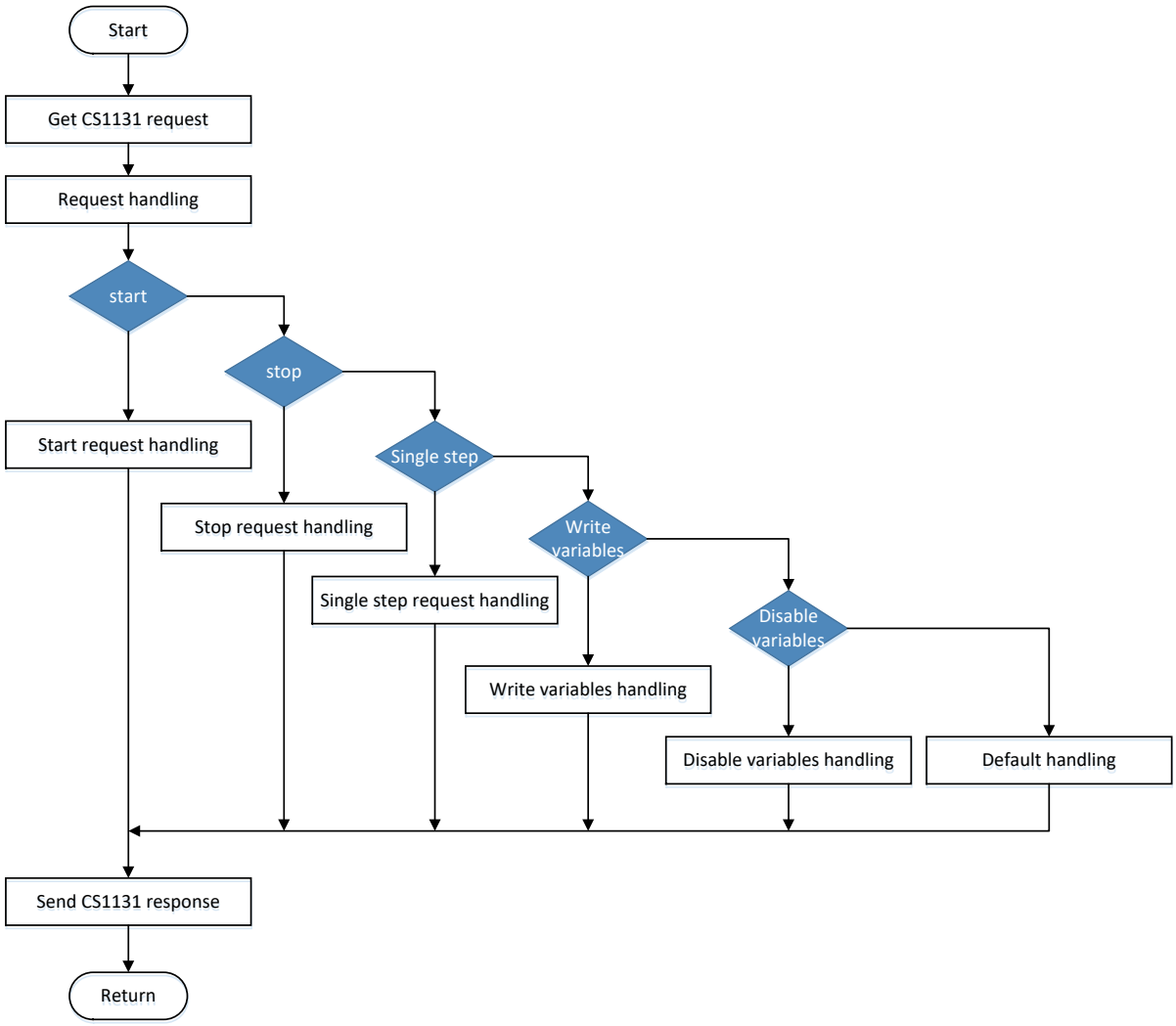

Figure 4-4 CS1131 service data handling

图 4-4 CS1131 服务数据处理

4.3.3 MonitorOPCPreprocess

4.3.3.1 Function Description 功能描述

OPC service data handling.

OPC 服务数据处理。

4.3.3.2 Argument Description 参数说明

- Definition 函数定义

static void MonitorOPCPreprocess (void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.3.3.3 Processing flow 处理流程

The processing flow is omitted.

流程图省略。

4.3.4 MonitorModbusPreprocess

4.3.4.1 Function Description 功能描述

Modbus service data handling.

Modbus 服务数据处理。

4.3.4.2 Argument Description 参数说明

- Definition 函数定义

static void MonitorModbusPreprocess (void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.3.4.3 Processing flow 处理流程

The processing flow is omitted.

流程图省略。

4.3.5 MonitorP2PPreprocess

4.3.5.1 Function Description 功能描述

P2P service data handling.

P2P 服务数据处理。

4.3.5.2 Argument Description 参数说明

- Definition 函数定义

static void MonitorP2PPreprocess (void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.3.5.3 Processing flow 处理流程

The processing flow is omitted.

流程图省略。

4.3.6 MonitorWriteVarHandle

4.3.6.1 Function Description 功能描述

Write variables handling.

写变量处理。

4.3.6.2 Argument Description 参数说明

- Definition 函数定义

static void MonitorWriteVarHandle (Task_ID_t emTaskID);

- Input argument 输入参数

Task_ID_t emTaskID

Task ID

任务的 ID 号

- Output argument 输出函数

No.

无。

4.3.6.3 Processing flow 处理流程

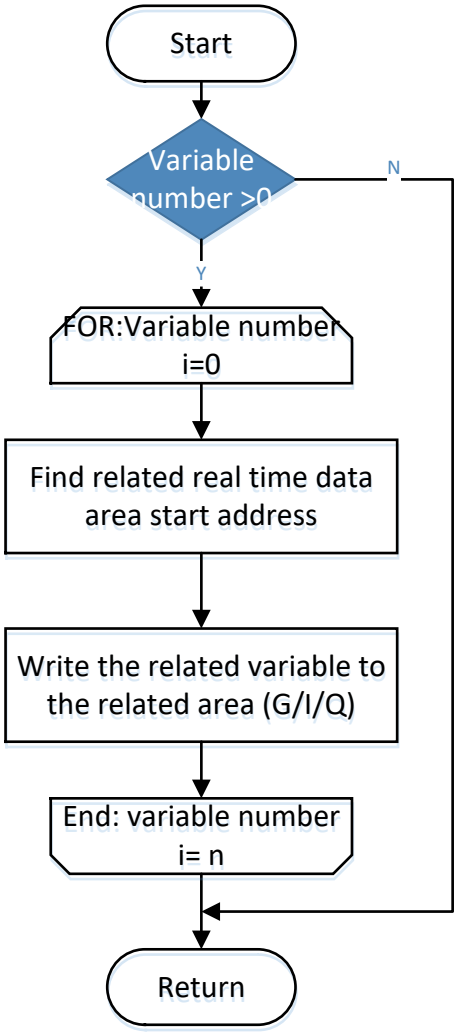

Figure 4-5 Write variables

图 4-5 写变量

4.3.7 MonitorCS1131DisableVarPreHandle

4.3.7.1 Function Description 功能描述

Disable variable service data handling.

Disable 变量服务数据处理。

4.3.7.2 Argument Description 参数说明

- Definition 函数定义

static uint16_t MonitorCS1131DisableVarPreHandle(uint8_t ucCS1131Data[], uint32_t usCS1131DataLen,uint8_t ucACKData[]);

- Input argument 输入参数

uint8_t ucCS1131Data[]

CS1131 service data

CS1131 的服务数据

uint32_t usCS1131DataLen,

Data length

数据长度

- Output argument 输出函数

uint8_t ucACKData[].

Acknowledge data

应答数据

4.3.7.3 Processing flow 处理流程

The processing flow is omitted.

流程图省略。

4.4 User project control (Core0) 用户工程控制（core0）

SWDD-PM-MM_SafR_NSecR_A_004

4.4.1 MonitorIECCycleCtrl2

4.4.1.1 Function Description 功能描述

Update UP1 task state and UP2 task state due to switch key state;

更新 UP1 任务及 UP2 任务的状态；

Initialized the real time data, external library and user project due to the different configuration after power on;

上电运行后，根据配置的不同来初始化实时数据区、外部库及用户工程；

It is used to control operations including switching from old project to new project, project start or stop etc.

根据工程存在与否进行相应的初始化；控制工程的切换、启动、停止等操作。

4.4.1.2 Argument Description 参数说明

- Definition 函数定义

static void MonitorIECCycleCtrl2(void);

- Input argument 输入参数

No.

无。

➢ Output argument 输出函数

No.

无。

4.4.1.3 Processing flow 处理流程

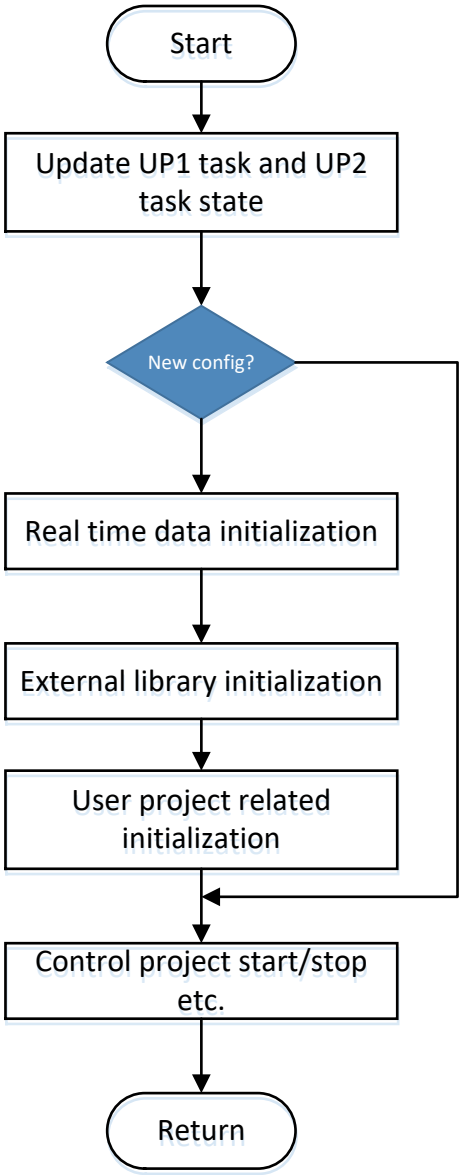

Figure 4-6 User project control

图 4-6 用户工程控制

4.5 Monitor module initialization (core1) 监视模块初始化（core1）

SWDD-PM-MM_SafR_NSecR_A_005

4.5.1 MonitorInit

4.5.1.1 Function Description 功能描述

Monitor module initialization.

监视模块初始化。

4.5.1.2 Argument Description 参数说明

- Definition 函数定义

void MonitorInit(void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.5.1.3 Processing flow 处理流程

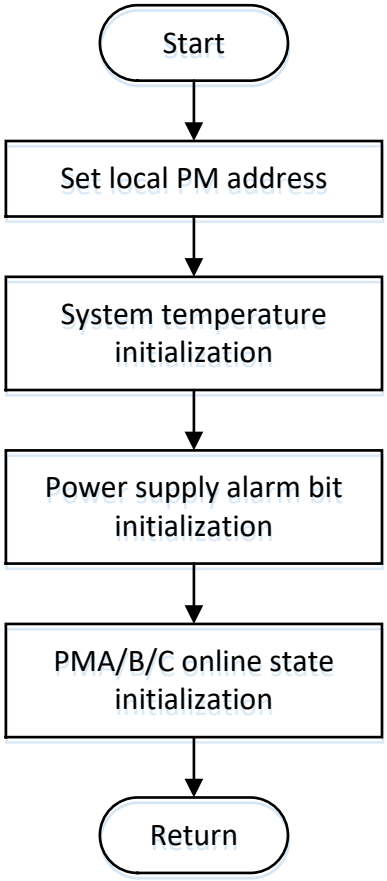

Figure 4-7 Monitor module initialization

图 4-7 监视模块初始化

4.6 Monitor module cycle (core1) 监视模块周期运行（core1）

SWDD-PM-MM_SafR_NSecR_A_006

4.6.1 MonitorCycle

4.6.1.1 Function Description 功能描述

Monitor module runs periodically.

监视模块周期运行。

4.6.1.2 Argument Description 参数说明

- Definition 函数定义

void MonitorCycle(void);

- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.6.1.3 Processing flow 处理流程

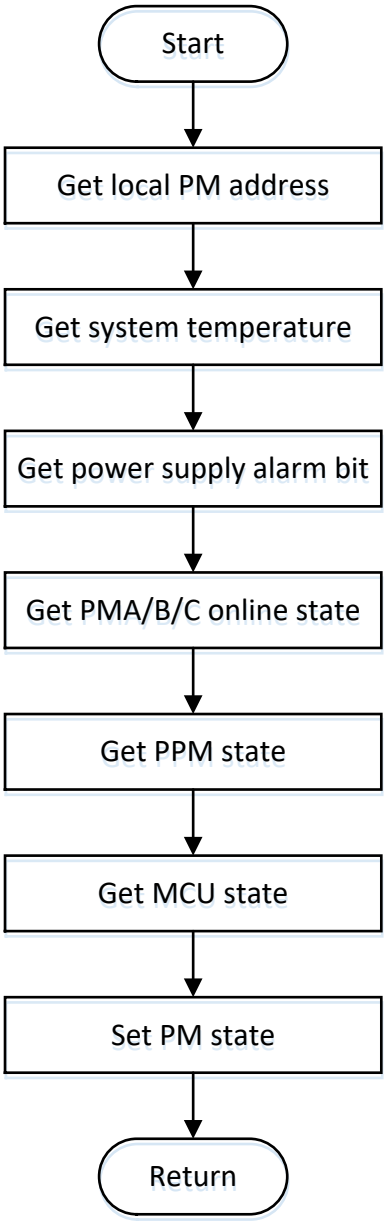

Figure 4-8 Monitor module cycle

图 4-8 监视模块周期运行

——以下无正文