

Document Title: PM_FW data processing module design
description of Safety Control System

Document Number: 16-Q04-000394

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 数据处理模块设计说明书

文件编号：16-Q04-000394

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	IO data input/output IO 数据输入输出	4
3.2.2	IO data exchange IO 数据交换.....	5
3.2.3	IO data vote IO 数据表决.....	6
3.3	Interface function 接口函数.....	8
3.4	Global variable 全局变量	9
3.5	Data structure 数据结构.....	9
3.6	List of sub-function 子功能列表	11
4	Design of sub-function 子功能设计	11
4.1	Module initialization 模块初始化	11
4.1.1	DataProcessInit.....	11
4.1.2	DataExChangeInit	12
4.2	IO data input/output IO 数据输入输出.....	12
4.2.1	DataProcessInputHandle	12
4.2.2	DataProcessOutputHandle.....	14
4.2.3	DataProcessSlaveSendHandle.....	17
4.2.4	DataProcessSlaveRecvHandle.....	18
4.2.5	DataProcessGenerateSoftSOERecord	19
4.3	IO data exchange IO 数据交换.....	21
4.3.1	DataExChange.....	21
4.4	IO data voting IO 数据表决.....	23
4.4.1	DataExChangeVote	23
4.4.2	DataExChangeTriVote.....	24
4.4.3	DataExChangeDoulVote	28

4.4.4	DataExChangeSingleVote	33
-------	------------------------------	----

1 Document overview 文档概述

1.1 Introduction 综述

This document describes the design description of data processing 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 数据处理模块的设计方案。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 data processing module (marked red) in the software hierarchy is shown below.

数据处理模块（标红）在软件层次中的位置如下图所示。

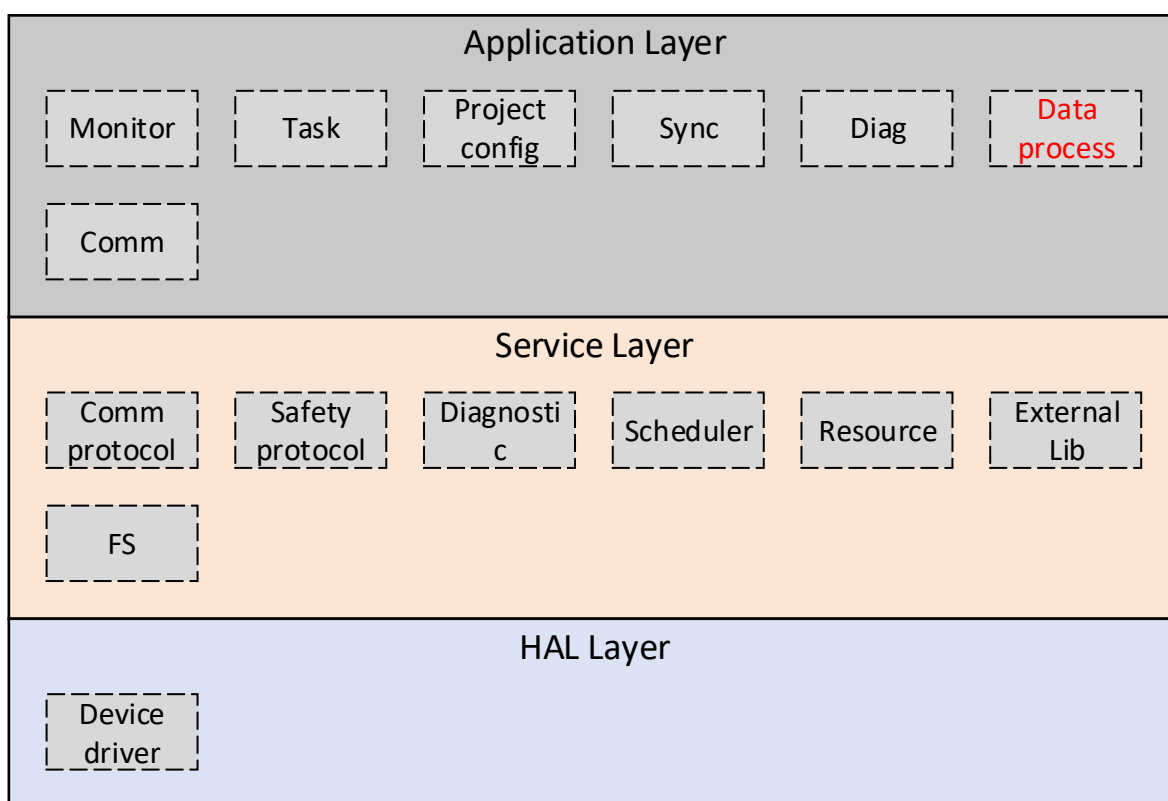


Figure 2-1 the location of the data processing module

图 2-1 模块位置

Data processing module is used to process IO input data and IO output data.

数据处理模块主要用于实现对 IO 从站数据的处理功能。

3 Module design 模块设计

3.1 Function description 功能描述

Each PM obtains IO input data, exchanges the input data with other PMs and votes the data. The voting results is the input to the user program. After executing the user program, three PMs exchange the output data of the user program and vote the data, and the voting result is the final output to the IO module.

各系 PM 模块获取 IO 输入数据，和其他两系交换输入数据并进行表决，表决后的结果作为各系运算的输入。运算后，三系之间交换运算结果并进行表决，表决后结果作为给 IO 模块的输出数据。

Therefore, this module includes three parts: IO data input/output, IO data exchange, and IO data voting.

因此，模块功能可以分为 IO 数据输入输出、IO 数据交换和 IO 数据表决三个子功能。

3.2 Design concept 设计思路

3.2.1 IO data input/output IO数据输入输出

3.2.1.1 IO data output IO 数据输出

Detailed output data process flow is shown as follows:

1. Determine the IO modules which shall be polling in this cycle. Each FW cycle polls a specified number of IO modules, and each round of IO polling polls the output module first and then polls the input module.
2. Obtain output data.
3. Call IO data exchange function.
4. Call IO data voting function.
5. The output data is packaged using a safety protocol.
6. Send the data to the IO module.
7. End.

IO 输出数据的具体处理流程如下：

1. 确定本周期需要轮询的 IO 模块，每个 FW 周期轮询确定数量的 IO 模块，每一轮 IO 轮询先轮询输出模块，再轮询输入模块；
2. 获取输出数据；
3. 调用 IO 数据交换功能

4. 调用 IO 数据表决功能
5. 使用安全协议将输出数据打包;
6. 将打包的数据发送 IO 模块;
7. 结束。

3.2.1.2 IO data input IO 数据输入

Detailed input data process flow is shown as follows:

1. Determine the IO modules that have been polled in the current cycle.
2. Obtain Input data from IO module.
3. Use the safety protocol to unpack the input data.
4. Process the real-time data, SOE data, diagnostic data and OSP monitoring data which are in the input data.
5. Call IO data exchange function.
6. Call IO data voting function.
7. End

IO 输入数据的具体处理流程如下:

1. 确定本周期已经轮询的 IO 模块;
2. 从 IO 模块获取输入数据;
3. 使用安全协议将输入数据解包;
4. 处理输入数据中的实时数据、SOE 数据、诊断数据、OSP 监控数据;
5. 调用 IO 数据交换功能
6. 调用 IO 数据表决功能
7. 结束。

3.2.2 IO data exchange IO数据交换

Data exchange function provides two interface functions: input data exchange and output data exchange. The input data exchange content includes real-time input data, input data quality, IPBUS state; the output data exchange content includes real-time output data.

数据交换功能提供输入数据交换, 输出数据交换两个接口函数。输入数据的交换内容包括实时输入区数据、输入区数据质量、IPBUS 状态; 输出数据的交换内容包括实时输出区数据。

3.2.3 IO data vote IO数据表决

3.2.3.1 Data vote 数据表决:

The voting mechanism is shown in the following table:

表决机制如下表所示:

Table 3-2 Vote table

表 3-2 表决表

	Quality Channel A	Quality Channel B	Quality Channel C	Voting Configuration	Voting Result
DI	Normal	Normal	Normal	—	Result of 2oo3
	Normal	Normal	Bad	2oo2	Result of 2oo2
				1oo2	Result of 1oo2
	Normal	Bad	Bad	—	Channel A's value
	Bad	Bad	Bad	—	Value set by user
DO	Normal	Normal	Normal	—	Result of 2oo3
	Normal	Normal	Bad	2oo2	Result of 2oo2
				1oo2	Result of 1oo2
	Normal	Bad	Bad	—	Channel A's value
	Bad	Bad	Bad	—	Safety value
AI	Normal	Normal	Normal	—	Median
	Normal	Normal	Bad	—	Mean
	Normal	Bad	Bad	—	Channel A's value
	Bad	Bad	Bad	—	Value set by user
AO	Normal	Normal	Normal	—	Median
	Normal	Normal	Bad	—	Mean
	Normal	Bad	Bad	—	Channel A's value
	Bad	Bad	Bad	—	Safety value
PI	Normal	Normal	Normal	—	Median
	Normal	Normal	Bad	—	Maximum
	Normal	Bad	Bad	—	Channel A's value
	Bad	Bad	Bad	—	Value set by user

- The quality of input channel is Normal only when the quality of data, the quality of IP_BUS and the quality of PM_BUS are all Normal
- The quality of output channel is Normal only when quality of data and the quality of PM_BUS are both Normal

- If an analog data is out of the tolerance, the quality is bad.
- When a task is configured as “Control” property, the task can be configured as 1oo2 or 2oo2 voting. When a task is configured as “Safety” property, only 1oo2 voting can be configured.
- 对于输入数据来说，某系数据的质量取决于数据本身的质量、IP_BUS 的质量和 PM_BUS 的质量。如果这三个质量有一个为坏，则该系质量为坏，即该系输入数据无效。
- 对于输出数据来说，数据质量取决于数据本身的质量和 PM_BUS 的质量。如果这两个质量有一个为坏，则该系质量为坏，即该系输出数据无效。
- 对于模拟量输入来说，如果其中某一系的值超过偏差允许的范围，则该系质量为坏，即该系输入数据无效。
- 对于数字量数据来说，在 control 方式下，用户可配置两系质量好时按照 1oo2 或 2oo2 表决；在 safety 运行方式下，两系质量好时只能按照 1oo2 表决。

3.2.3.2 Offset error 偏移故障

The dead zone size in the fixed offset is the span * dead zone percentage. For the analog input data, the dead zone percentage is set to 2%. When a fixed offset error occurs for a voting values, this value does not participate voting.

固定偏移中死区的大小为量程*百分比，对于模拟量输入数据，死区百分比定为 2%，当一系表决值发生固定偏移故障时，这一值不参与表决。

The fixed offset judgment shown in the below table is the case when the three channel’s quality are all normal. If only two channel’s quality are normal, when the offset between the two channel’s data is greater than the dead zone size, the two channel’s data are both considered as bad.

三系数据质量都为好时，进行的偏差判断如下表所示。当只有两系数据为好时，两系数据的差的绝对值大于死区时，认为两系数据都为坏。

Table 3-1 Fixed offset judgment

表 3-1 固定偏差判断

offset between the data, X is the dead zone size 两系差值，X 为死区大小		Judgment result 结果
$ A - C \leq X$	$ A - B \leq X$	Channel A quality is normal A 质量为好
$ A - C \leq X$	$ A - B > X$	Channel A quality is normal A 质量为好
$ A - C > X$	$ A - B \leq X$	Channel A quality is normal A 质量为好

$ A - C > X$	$ A - B > X$	Channel A quality is bad A 质量为坏
$ B - A \leq X$	$ B - C \leq X$	Channel B quality is normal B 质量为好
$ B - A \leq X$	$ B - C > X$	Channel B quality is normal B 质量为好
$ B - A > X$	$ B - C \leq X$	Channel B quality is normal B 质量为好
$ B - A > X$	$ B - C > X$	Channel B quality is bad B 质量为坏
$ C - B \leq X$	$ C - A \leq X$	Channel C quality is normal C 质量为好
$ C - B \leq X$	$ C - A > X$	Channel C quality is normal C 质量为好
$ C - B > X$	$ C - A \leq X$	Channel C quality is normal C 质量为好
$ C - B > X$	$ C - A > X$	Channel C quality is bad C 质量为坏

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 DataProcessInit(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 DataProcessOutputHandle(void)

Input argument 输入参数	Output argument 输出参数	Description 描述
No. 无。	No. 无。	Output data processing interface function. 输出数据处理调用接口

3. void DataProcessInputHandle ()

Input argument	Output argument	Description
----------------	-----------------	-------------

接口输入参数	接口输出参数	描述
No. 无。	No. 无。	Input data processing interface function. 输出数据处理调用接口

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.	uint32_t	uiUP1RollingSlaveNum	UP1 IO polling number. 任务 1 从站轮询个数
2.	uint32_t	uiUP2RollingSlaveNum	UP2 IO polling number. 任务 2 从站轮询个数
3.	uint8_t	s_ucChannelTolerate	Tolerance number 容忍次数
4.	uint8_t	ucDataProcessSOELastVar	Soft SOE last value 软 SOE 上一拍值
5.	uint8_t	ucUP1RollingSlaveNo	UP1 IO polling slave No. 任务 1 从站轮询从站号
6.	uint8_t	ucUP2RollingSlaveNo	UP2 IO polling slave No. 任务 2 从站轮询从站号

3.5 Data structure 数据结构

1. Slave slot head structure

```

typedef struct DataProcessSlaveSlotHeaderTag
{
    uint8_t ucSlotType;
    uint8_t ucSlotLen;
}DataProcessSlaveSlotHeader_t;
    
```

2. Slave SOE data structure

```

typedef struct DataProcessHardSOEDataTag
{
    uint32_t uiSOETick;
    uint8_t ucChannelNo;
    uint8_t ucEventState;
}DataProcessHardSOEData_t;
    
```

3. Slave state, temperature, device diagnostic data structure

```
typedef struct DataProcessStateDiagDataTag
{
    uint8_t ucStateDiag;
    uint8_t ucTempDiag;
    uint8_t uiDeviceDiag[4];
}DataProcessStateDiagData_t;
```

4. Slave channel diagnostic data structure

```
typedef struct DataProcessChannelDiagDataTag
{
    uint8_t ucChannelNo;
    uint8_t ucChannelDiag[3];
}DataProcessChannelDiagData_t;
```

5. OSP monitor data type A

```
typedef struct DataProcessOSPDiagDataOneTag
{
    uint32_t uiPRPMA;
    uint32_t uiPRPMB;
    uint32_t uiPRPMC;
    uint32_t uiPAccelA;
    uint32_t uiPAccelB;
    uint32_t uiPAccelC;
    uint32_t uiAlarmPackA;
    uint32_t uiAlarmPackB;
    uint32_t uiAlarmPackC;
    uint8_t ucDigPackA;
    uint8_t ucDigPackB;
    uint8_t ucDigPackC;
}DataProcessOSPDiagDataOne_t;
```

6. OSP monitor data type B

```
typedef struct DataProcessOSPDiagDataTwoTag
{
    uint8_t ucDigPackA;
    uint8_t ucDigPackB;
    uint8_t ucDigPackC;
}DataProcessOSPDiagDataTwo_t;
```


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-DP_SafR_NSecR_A_001	Module initialization. 模块初始化
SWDD-PM-DP_SafR_NSecR_A_002	IO data input/output IO 数据输入输出
SWDD-PM-DP_SafR_NSecR_A_003	IO data exchange IO 数据交换
SWDD-PM-DP_SafR_SecR_A_004	IO data voting IO 数据表决

4 Design of sub-function 子功能设计

4.1 Module initialization 模块初始化

SWDD-PM-DP_SafR_NSecR_A_001

4.1.1 DataProcessInit

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 DataProcessInit(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

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

此函数无分支，流程图省略。

4.1.2 DataExchangeInit

4.1.2.1 Function Description 功能描述

This function completes initialization of data exchange.

该函数完成数据交换的初始化。

4.1.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

void DataExchangeInit(void)

➤ Input argument 输入参数

No.

无。

➤ Output argument 输出函数

No.

无。

4.1.2.3 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 IO data input/output IO 数据输入输出

SWDD-PM-DP_SafR_NSecR_A_002

This sub-function is used to process the IO input/output data cyclically.

本子功能用于周期进行 IO 数据输入输出处理。

4.2.1 DataProcessInputHandle

4.2.1.1 Function Description 功能描述

This function is used to process the IO input data cyclically.

本函数用于周期进行 IO 数据输入处理。

4.2.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

void DataProcessInputHandle(void)

➤ Input argument 输入参数

No.

无。

➤ Output argument 输出函数

No.

无。

4.2.1.3 处理流程

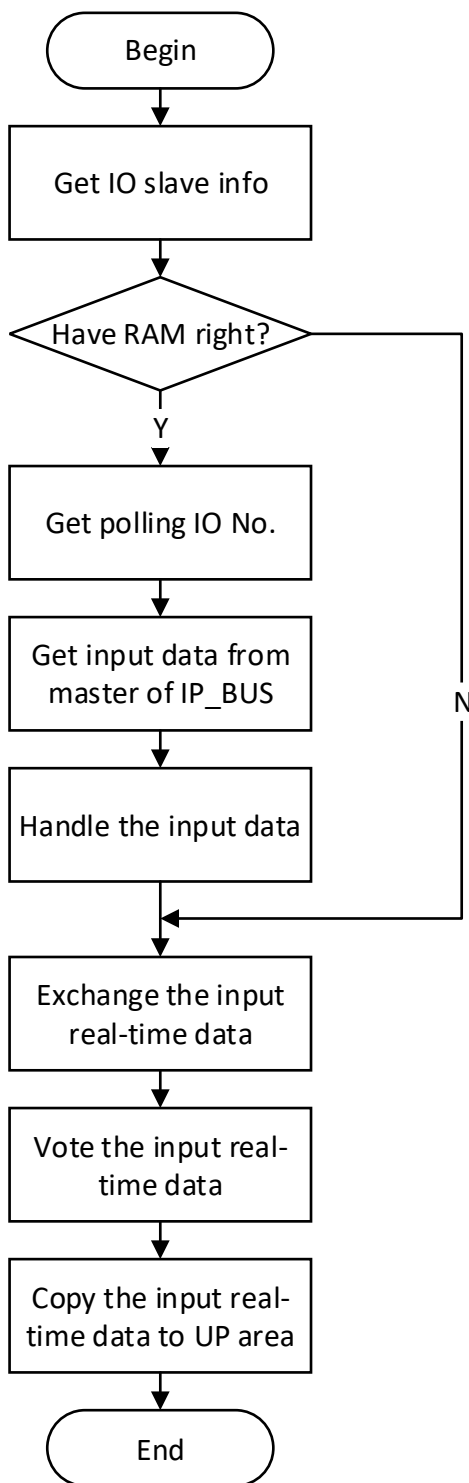


Figure 4-1 input data handle processing flow

图 4-1 输入数据处理流程图

4.2.2 DataProcessOutputHandle

4.2.2.1 Function Description 功能描述

This function is used to process the IO output data cyclically.

本函数用于周期进行 IO 数据输出处理。

4.2.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

void DataProcessOutputHandle(void)

➤ Input argument 输入参数

No.

无。

➤ Output argument 输出函数

No.

无。

4.2.2.3 处理流程

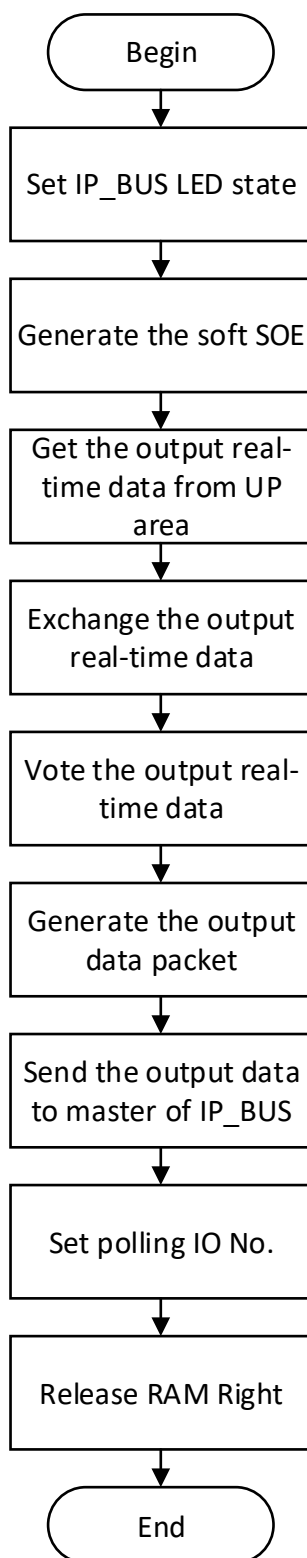


Figure 4-1 input data handle processing flow

图 4-1 输入数据处理流程图

4.2.3 DataProcessSlaveSendHandle

4.2.3.1 Function Description 功能描述

This function is used to generate the output data packet.

本函数用于生成从站输出数据包。

4.2.3.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
static void DataProcessSlaveSendHandle(Task_ID_t emTaskID, uint8_t ucSendDest[], uint8_t ucSlaveNo, bool_t bActiveFV, bool_t bEnableFlg)
```

➤ Input argument 输入参数

emTaskID: UP task ID, 任务 ID

ucSendDest: Send data buffer, 输出数据缓存

ucSlaveNo: Slave No, 从站号

bActiveFV: Safety flag, 安全标志位

bEnableFlg: Slave enable flag, 模块使能标志位

➤ Output argument 输出函数

ucSendDest: Send data buffer after handling, 处理后输出数据

4.2.3.3 处理流程

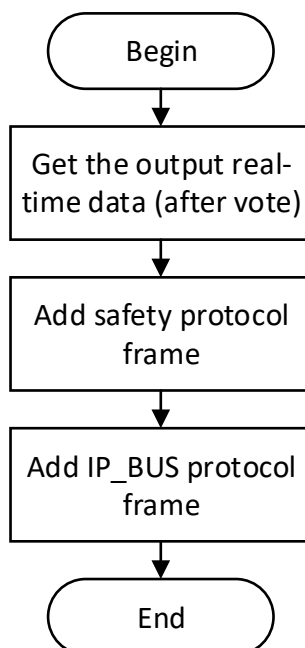


Figure 4-3 generate the output data packet processing flow

图 4-3 生成输出数据包

4.2.4 DataProcessSlaveRecvHandle

4.2.4.1 Function Description 功能描述

This function is used to process the IO slave input data.

本函数用于处理 IO 模块的输入数据。

4.2.4.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
static uint8_t DataProcessSlaveRecvHandle(Task_ID_t emTaskID, uint8_t ucRecvSrc[], uint8_t ucSlaveNo, uint16_t usModTag)
```

➤ Input argument 输入参数

emTaskID: UP task ID, 任务 ID

ucRecvSrc: Input data, 输入数据

ucSlaveNo: Slave No, 从站号

usModTag: Module type, 模块类型

➤ Output argument 输出函数

No.

无。

4.2.4.3 处理流程

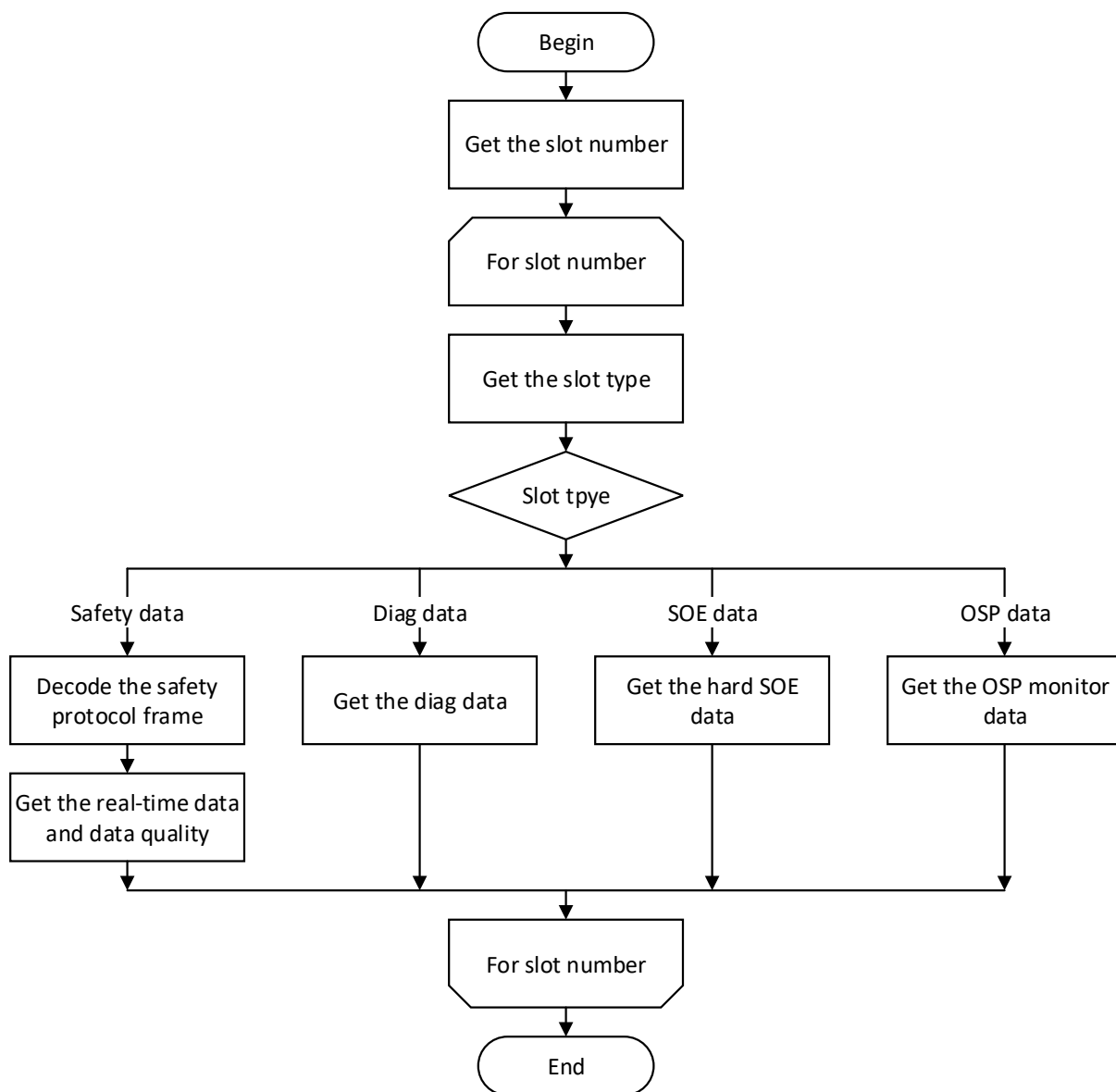


Figure 4-4 parse the input data packet processing flow

图 4-4 解析输入数据包

4.2.5 DataProcessGenerateSoftSOERecord

4.2.5.1 Function Description 功能描述

This function is used to generate soft SOE record.

本函数用于生成软 SOE 记录。

4.2.5.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
static void DataProcessGenerateSoftSOERecord(uint8_t ucSOEBlockNo)
```

➤ Input argument 输入参数

ucSOEBlockNo: SOE block No., 软 SOE 块号

➤ Output argument 输出函数

No.

无。

4.2.5.3 处理流程

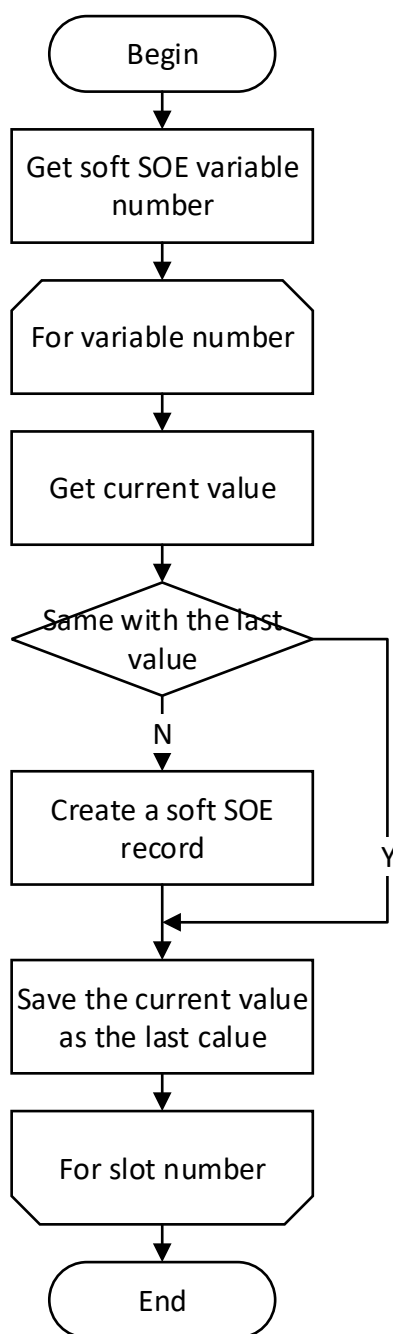


Figure 4-5 generate soft SOE record processing flow

图 4-5 生成软 SOE 记录

4.3 IO data exchange IO 数据交换

SWDD-PM-DP_SafR_NSecR_A_003

4.3.1 DataExChange

4.3.1.1 Function Description 功能描述

This function is used to exchange data with other PMs.

该函数负责和其他模块交换数据。

4.3.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
void DataExChange(uint8_t ucSrcData[], uint16_t usSrcDataLen,  
uint8_t ucPreData[], uint8_t ucNextData[], puint8_t pucPreDataFlg, puint8_t pucNextDataFlg)
```

➤ Input argument 输入参数

ucSrcData: data need to be exchange, 需交换的数据

usSrcDataLen: data length, 数据长度

➤ Output argument 输出函数

ucPreData: Pre PM data, 前一系 PM 数据

ucNextData: Next PM data, 后一系 PM 数据

pucPreDataFlg: Pre PM data state, 前一系 PM 数据状态

pucNextDataFlg: Next PM data state, 后一系 PM 数据状态

4.3.1.3 处理流程

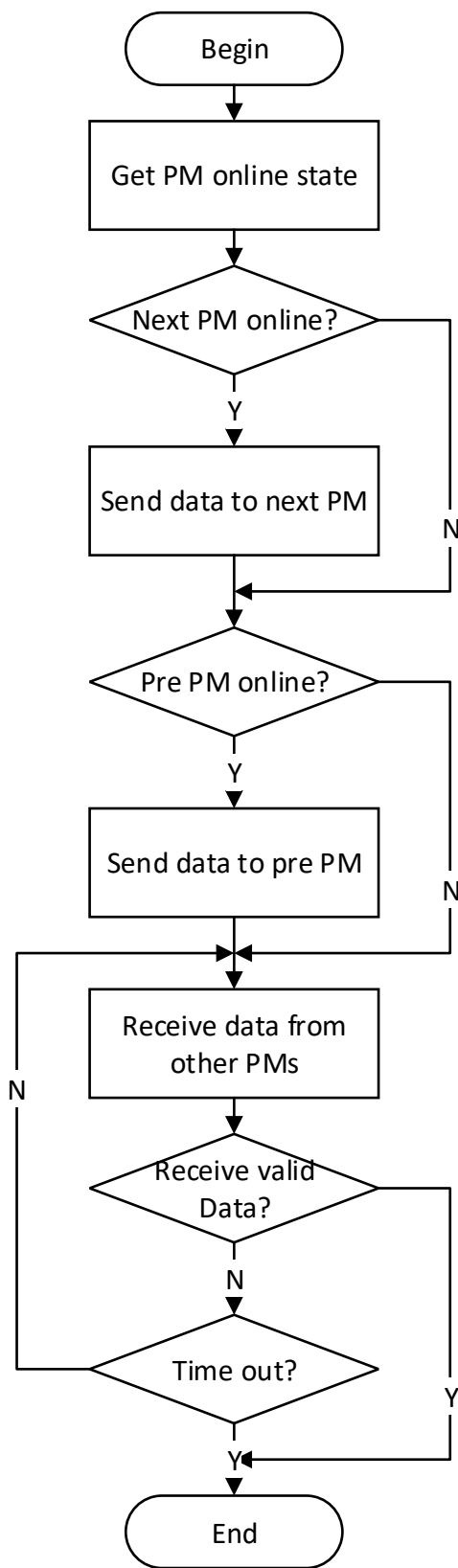


Figure 4-6 data exchange processing flow

图 4-6 数据交换

4.4 IO data voting IO 数据表决

SWDD-PM-DP_SafR_SecR_A_004

4.4.1 DataExchangeVote

4.4.1.1 Function Description 功能描述

This function is used to vote IO data cyclically.

本函数用于进行 IO 数据表决。

4.4.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
bool_t DataExchangeVote(Task_ID_t emTaskID, uint8_t ucModID , uint8_t ucData1[], uint8_t ucData2[], uint8_t ucData3[], uint8_t ucOldData[],uint8_t ucDestBuff[], uint8_t ucDiagBuff[])
```

➤ Input argument 输入参数

emTaskID: UP task ID, 任务 ID

ucModID: IO module No, 从站号

ucData1: Data1, 数据 1

ucData2: Data2, 数据 2

ucData3: Data3, 数据 3

ucOldData: old data, 上一拍数据

➤ Output argument 输出函数

ucDestBuff: data after voting, 表决后数据

ucDiagBuff: vote state, 表决的状态

4.4.1.3 处理流程

Determine how data are handled: triple voting, double voting or single voting according to the parameters ucData1, ucData2, ucData3 state, or determine to use old value or safe state according to tolerance judgment.

根据参数中 ucData1, ucData2, ucData3 的状态确定采用三系表决、两系表决、一系表决, 或者根据容忍状态使用老值或者是安全值。

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

此函数无分支, 流程图省略。

4.4.2 DataExchangeTriVote

4.4.2.1 Function Description 功能描述

This function is used to triple vote.

本函数用于三系数据表决。

4.4.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
static bool_t DataExchangeTriVote(Task_ID_t emTaskID,uint8_t ucModID , uint8_t ucData1[],  
uint8_t ucData2[], uint8_t ucData3[], uint8_t ucOldData[],uint8_t ucDestBuff[], uint8_t  
ucDiagBuff[])
```

➤ Input argument 输入参数

emTaskID: UP task ID, 任务 ID

ucModID: IO module No, 从站号

ucData1: Data1, 数据 1

ucData2: Data2, 数据 2

ucData3: Data3, 数据 3

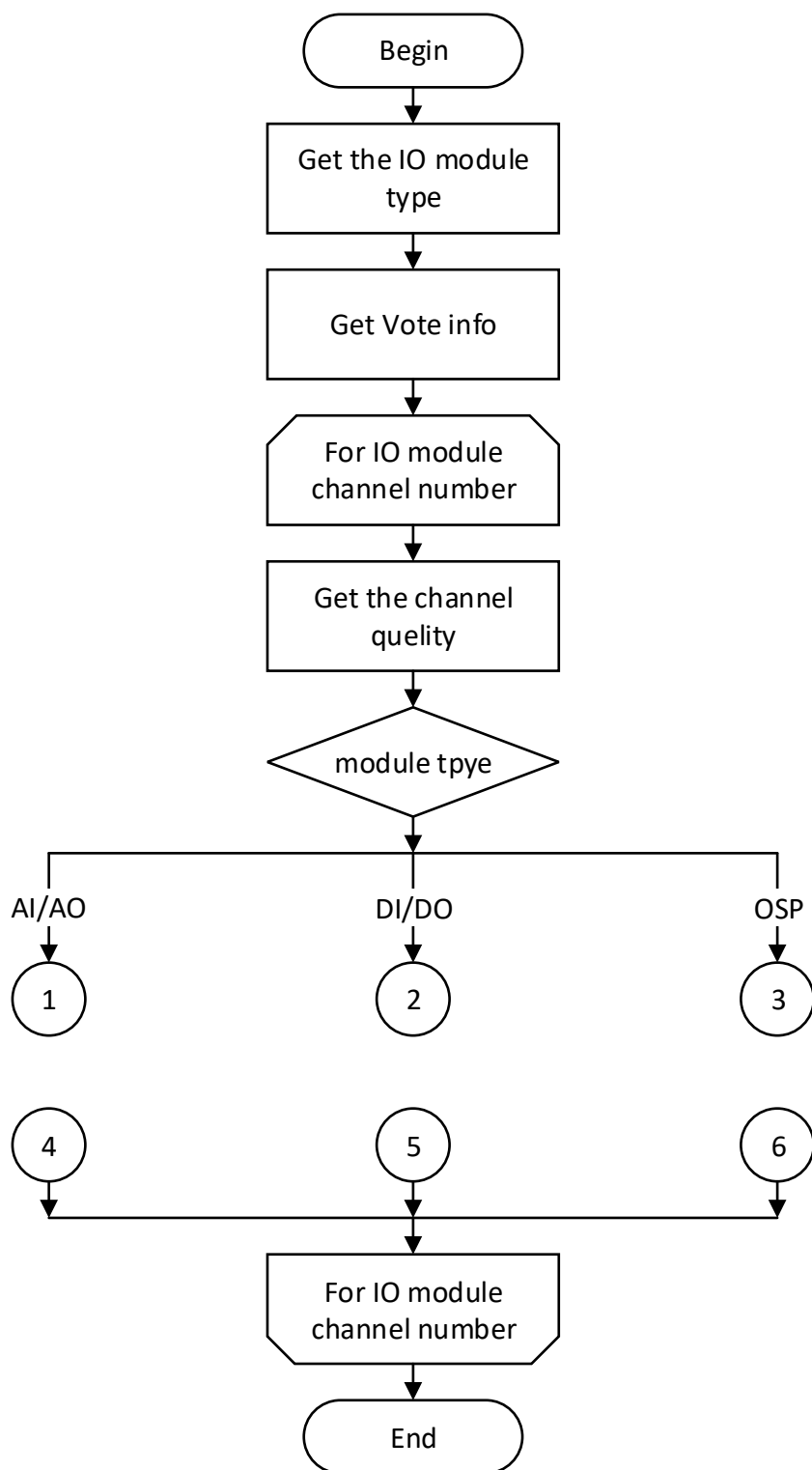
ucOldData: old data, 上一拍数据

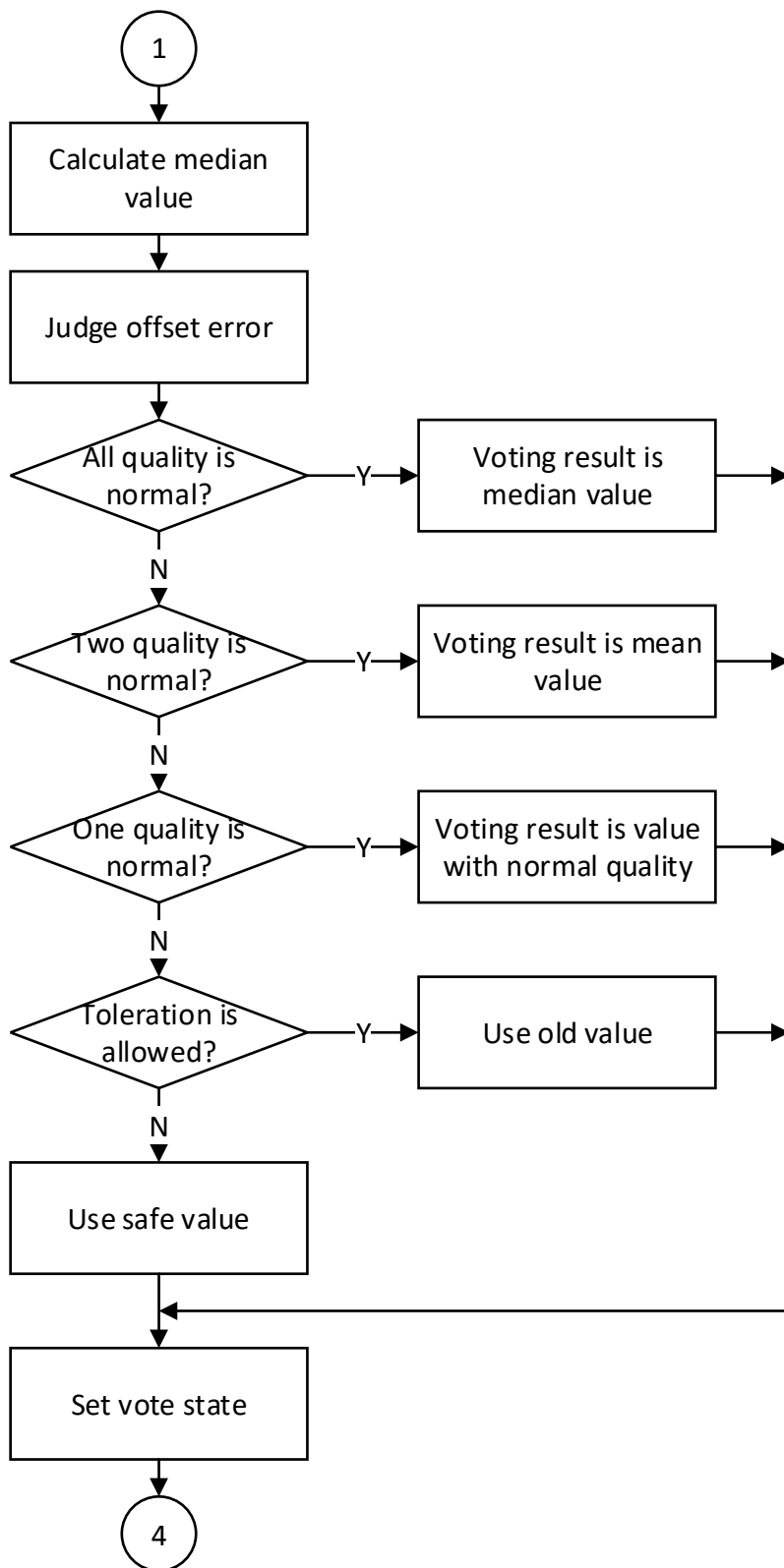
➤ Output argument 输出函数

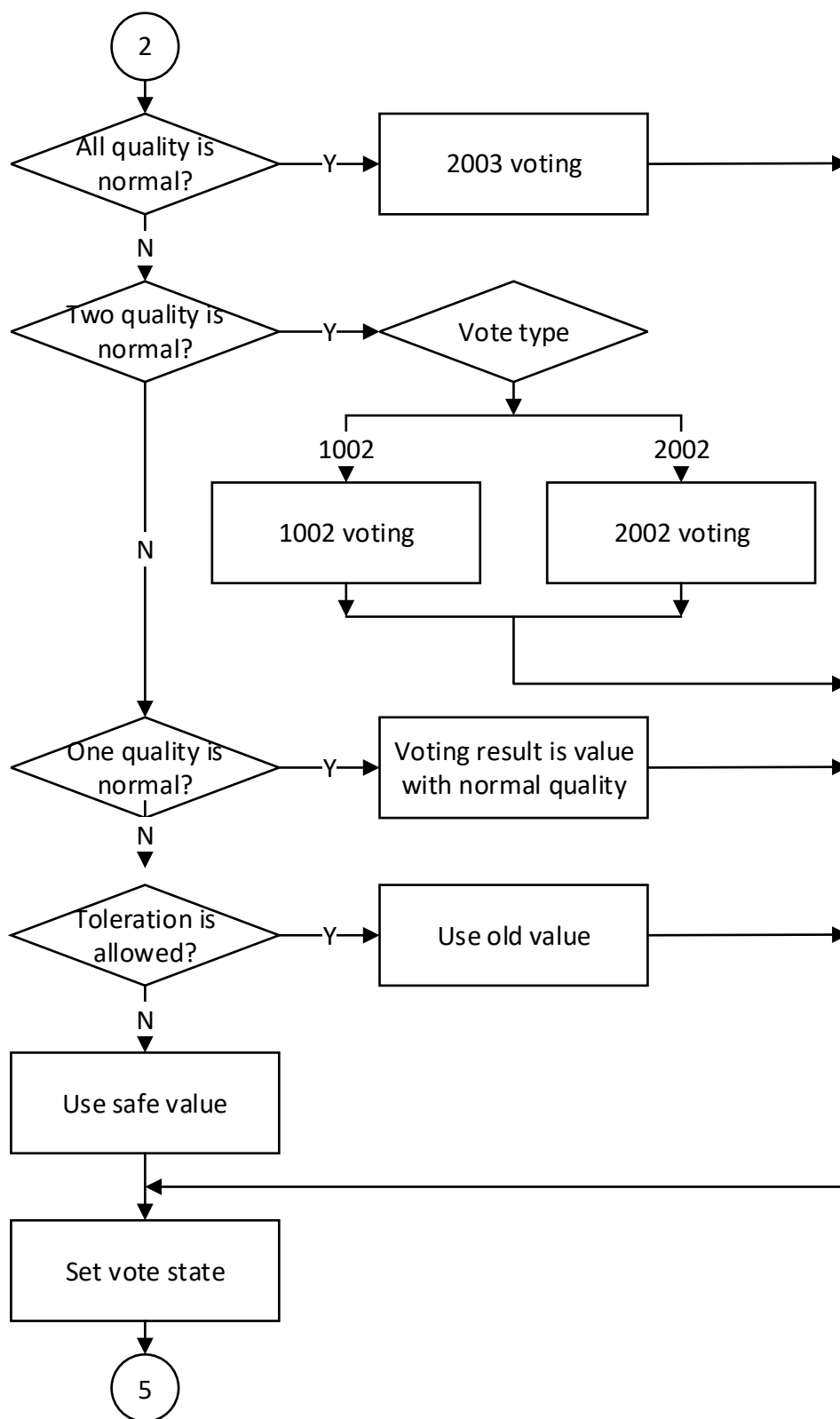
ucDestBuff: data after voting, 表决后数据

ucDiagBuff: vote state, 表决的状态

4.4.2.3 处理流程







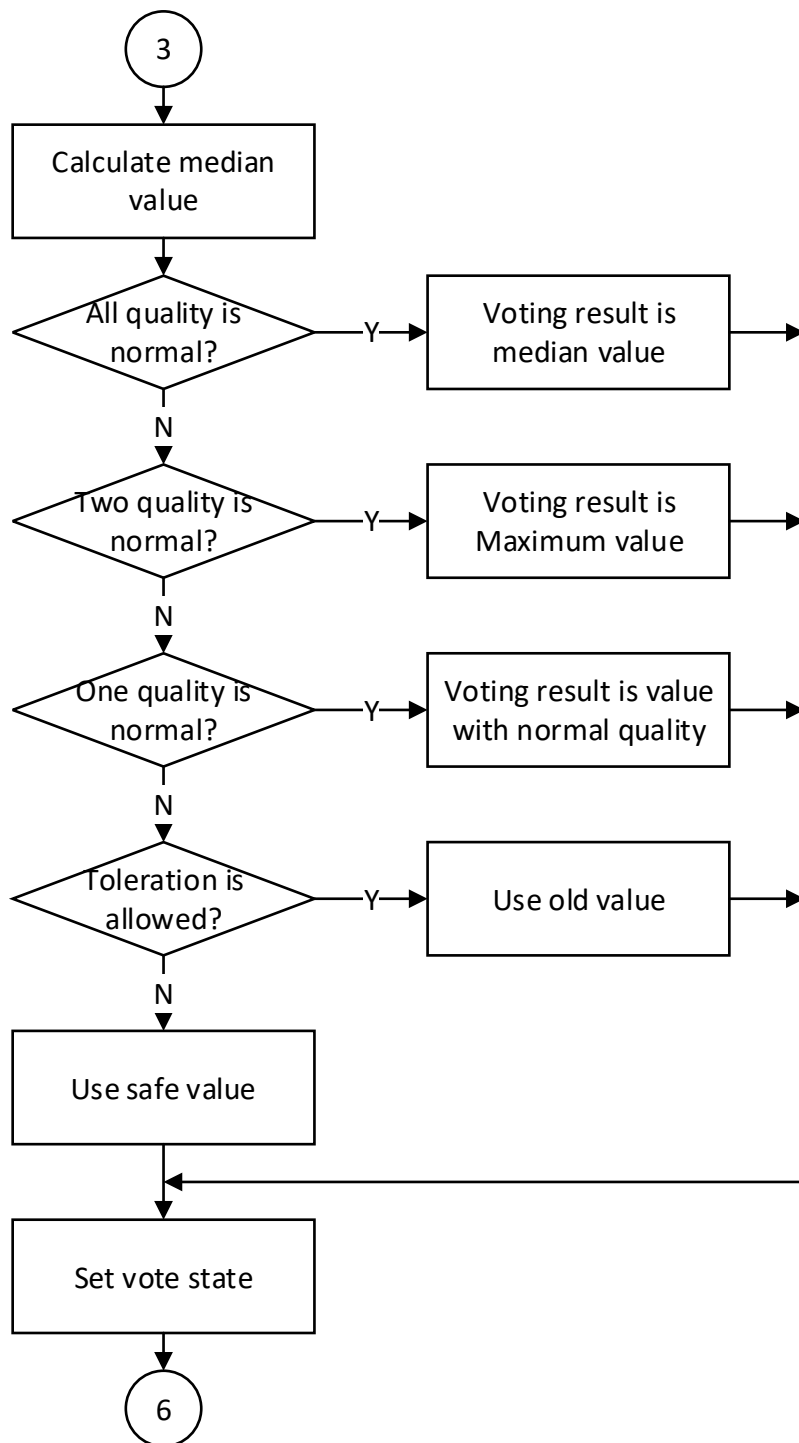


Figure 4-7 triple voting processing flow

图 4-7 三系数据表决

4.4.3 DataExchangeDoulVote

4.4.3.1 Function Description 功能描述

This function is used to double vote.

本函数用于两系数据表决。

4.4.3.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
static bool_t DataExChangeDoulVote(Task_ID_t emTaskID,uint8_t ucModID , uint8_t ucData1[],  
uint8_t ucData2[], uint8_t ucOldData[],uint8_t ucDestBuff[], uint8_t ucDiagBuff[])
```

➤ Input argument 输入参数

emTaskID: UP task ID, 任务 ID

ucModID: IO module No, 从站号

ucData1: Data1, 数据 1

ucData2: Data2, 数据 2

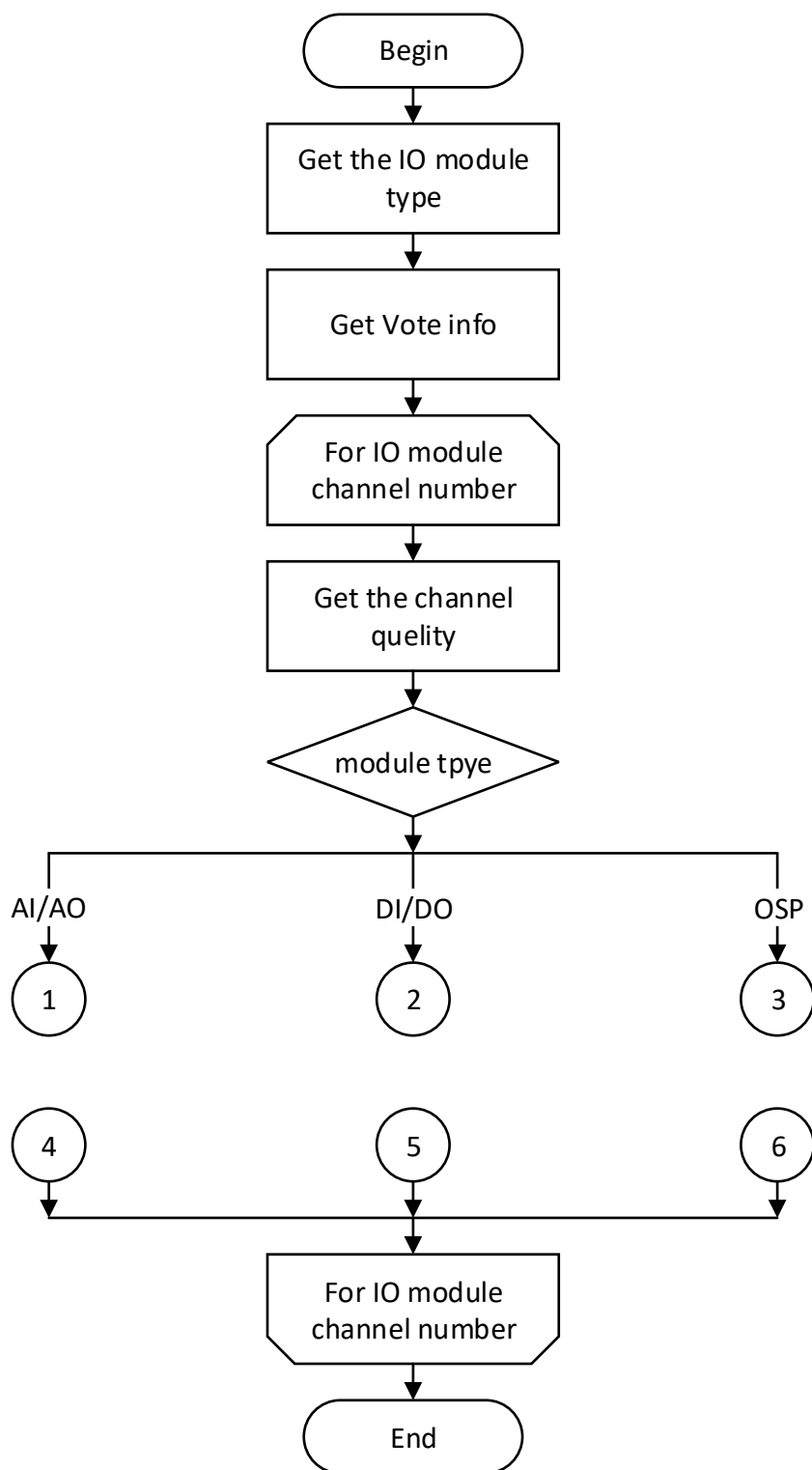
ucOldData: old data, 上一拍数据

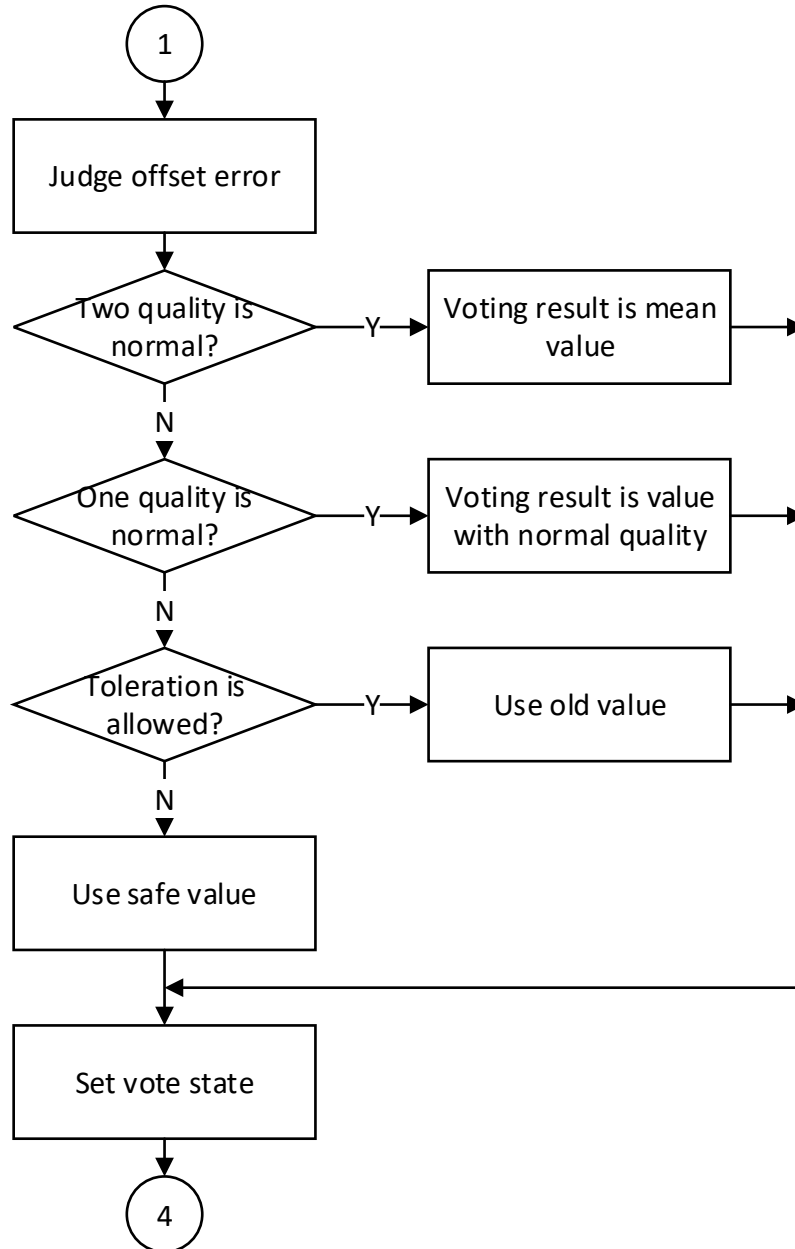
➤ Output argument 输出函数

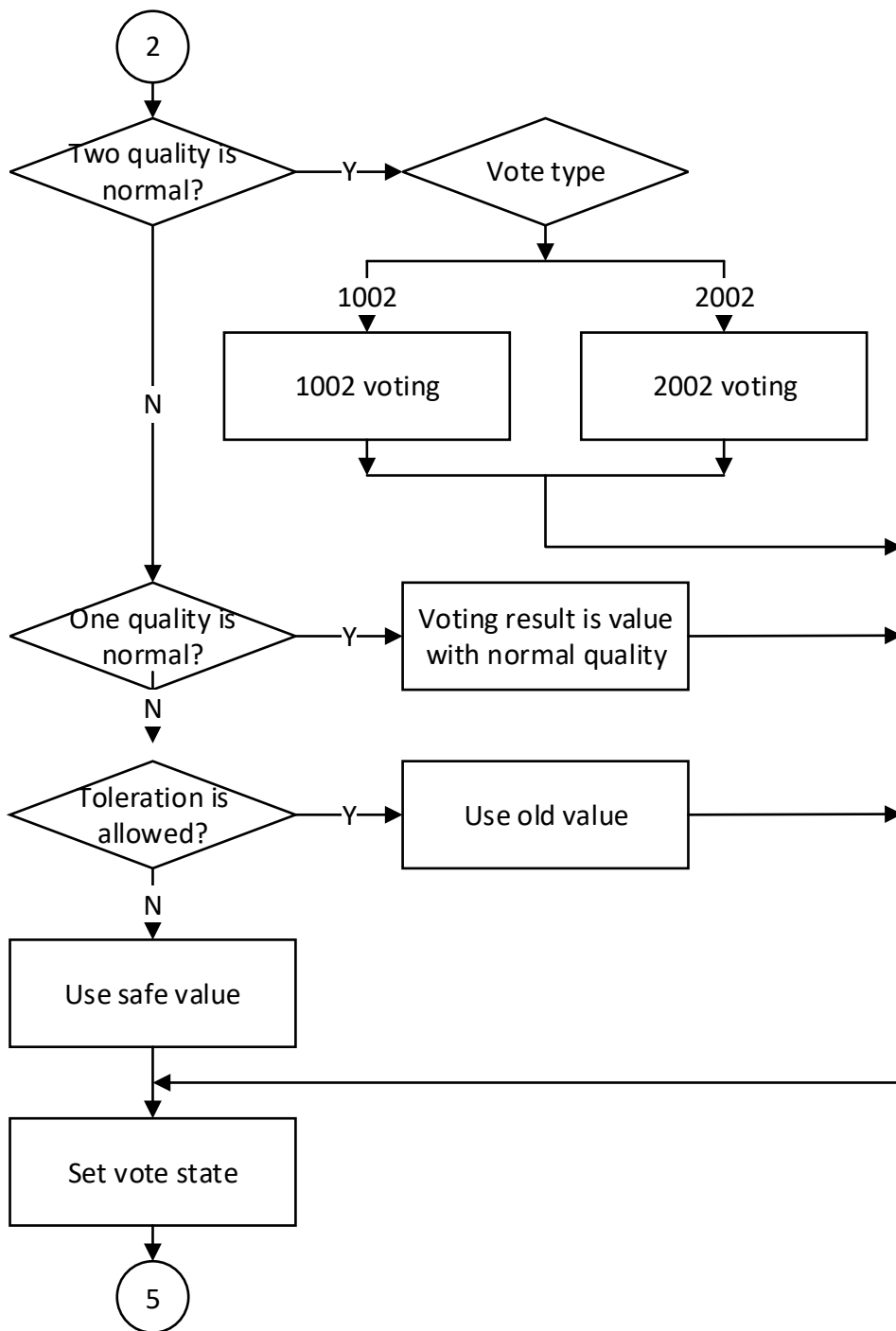
ucDestBuff: data after voting, 表决后数据

ucDiagBuff: vote state, 表决的状态

4.4.3.3 处理流程







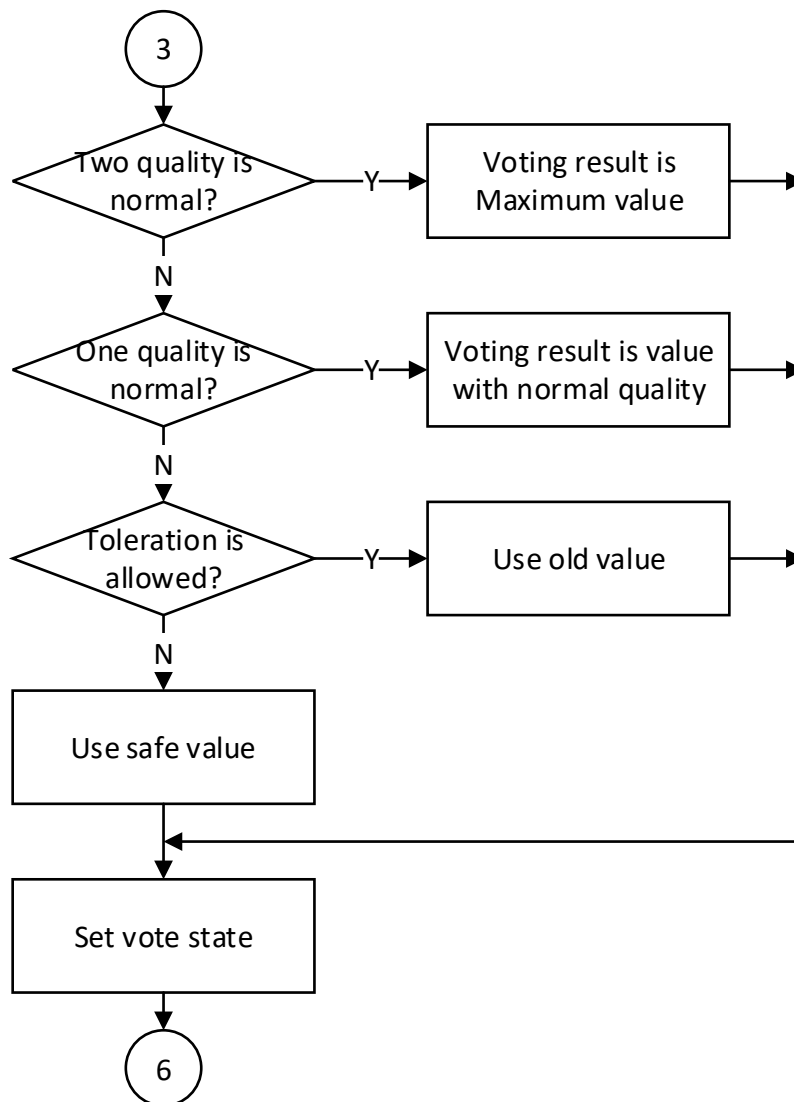


Figure 4-8 double voting processing flow

图 4-8 两系数据表决

4.4.4 DataExChangeSingleVote

4.4.4.1 Function Description 功能描述

This function is used to single vote.

本函数用于单系数数据表决。

4.4.4.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
void DataProcessOutputHandle(void)
```

```
static bool_t DataExChangeSingleVote(Task_ID_t emTaskID,uint8_t ucModID , uint8_t ucData1[],
uint8_t ucOldData[],uint8_t ucDestBuff[], uint8_t ucDiagBuff[])
```

➤ Input argument 输入参数

emTaskID: UP task ID, 任务 ID

ucModID: IO module No, 从站号

ucData1: Data1, 数据 1

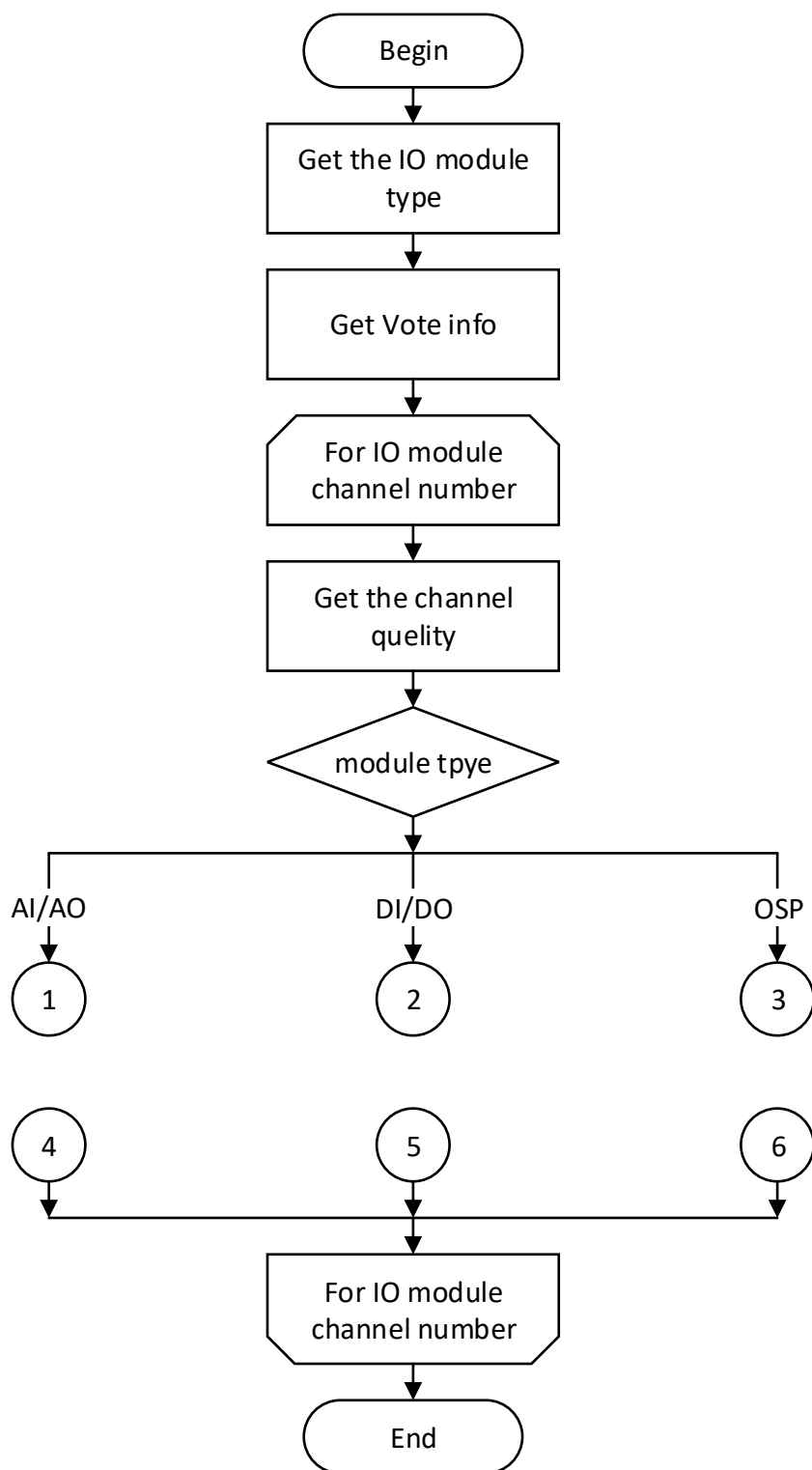
ucOldData: old data, 上一拍数据

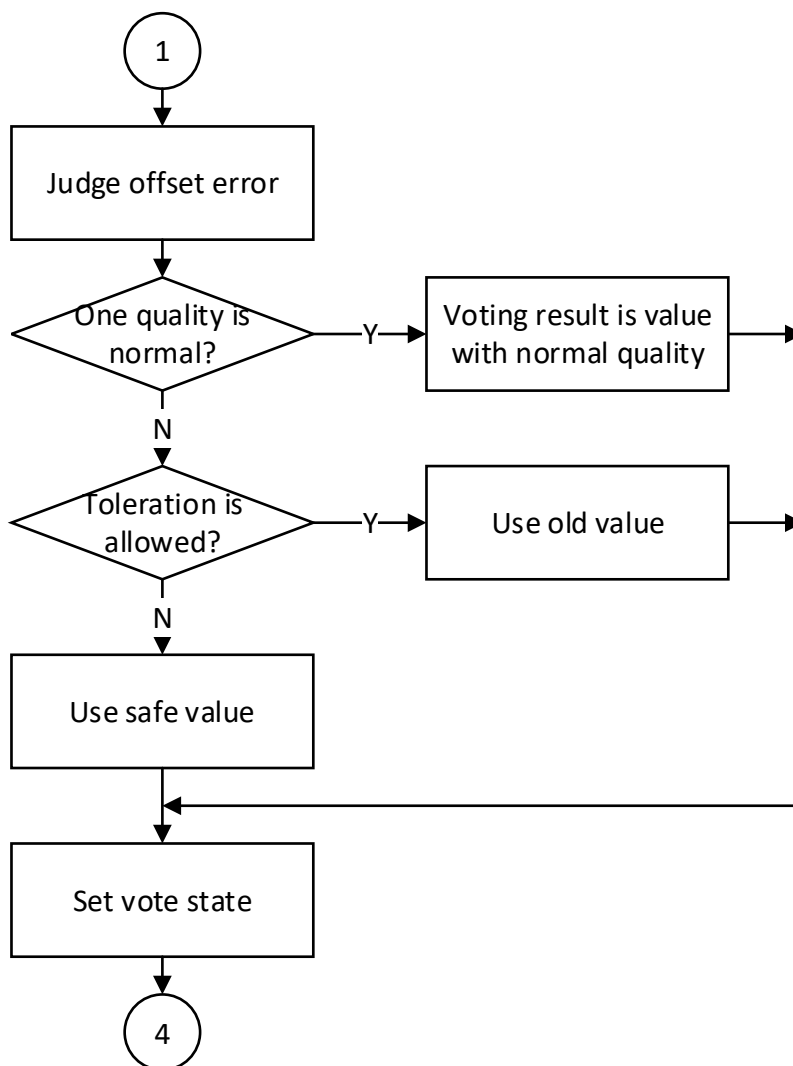
➤ Output argument 输出函数

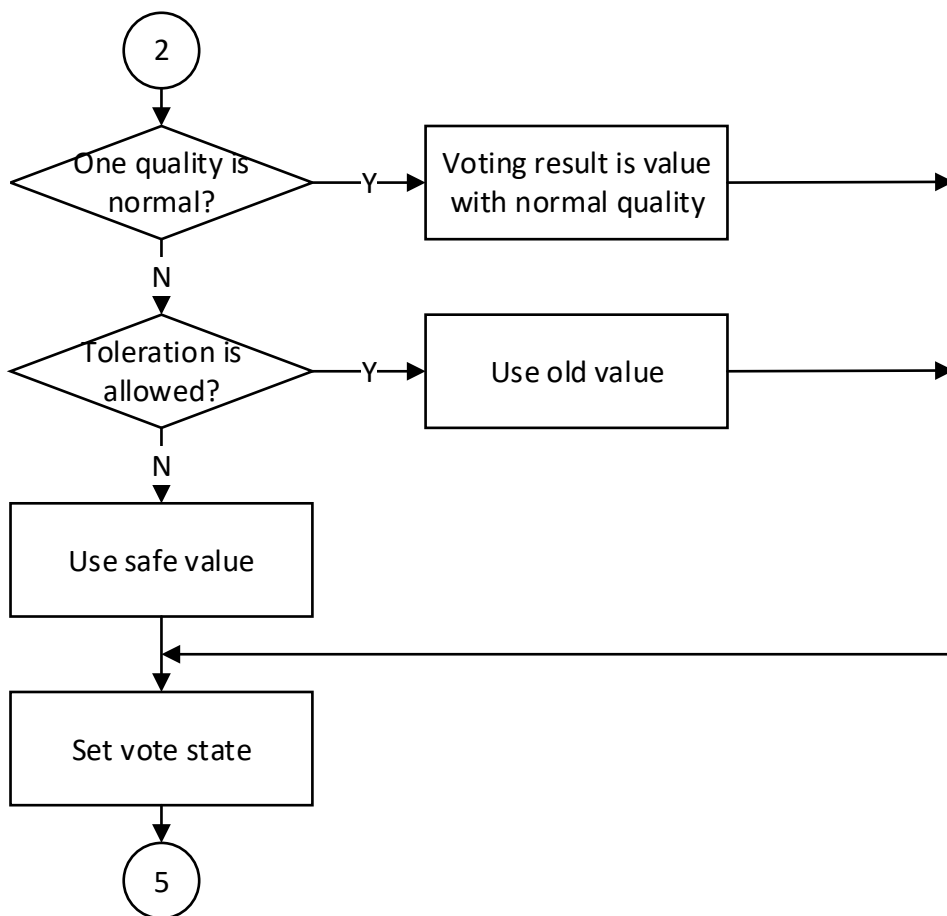
ucDestBuff: data after voting, 表决后数据

ucDiagBuff: vote state, 表决的状态

4.4.4.3 处理流程







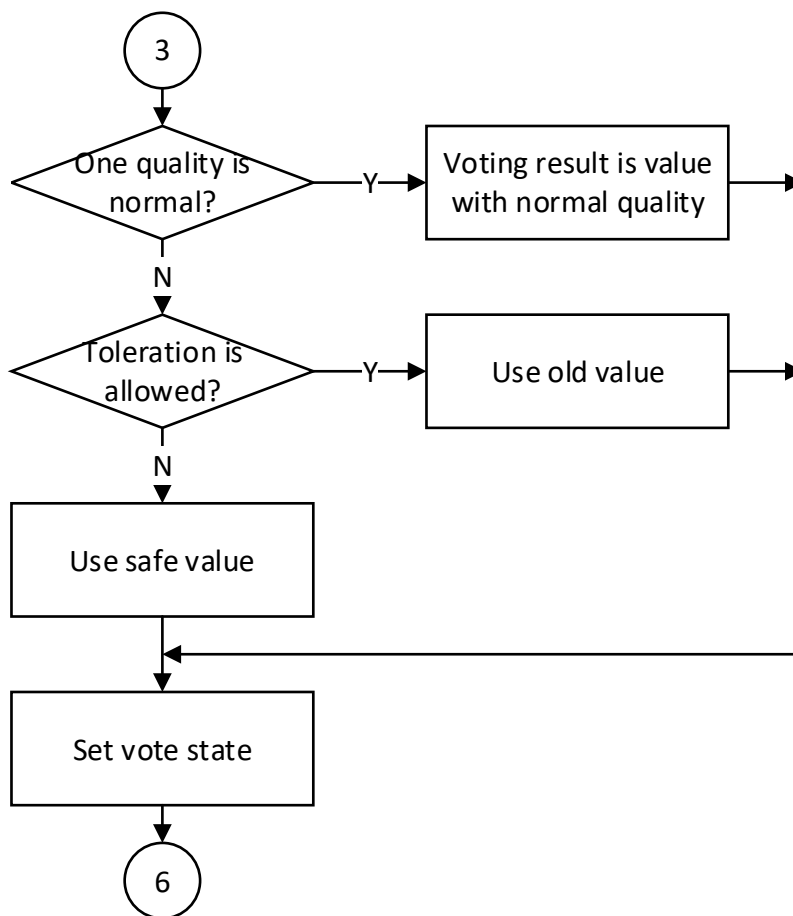


Figure 4-7 single voting processing flow

图 4-7 单系数数据表决

——以下无正文