

iocollect 软件 功能说明及操作说明书

编 制：_____

校 对：_____

审 核：_____

批 准：_____

版本号	修改内容	修改人	修改时间
V1.0	新建	张浩	2023-03-30
V1.1	添加：波特率修改	张浩	2023-05-07
V1.2	添加：在线升级	张浩	2023-05-25
V2.0	添加： 嵌入式实时操作系统	张浩	2023-06-08

目 录

1 概述	1
1.1 背景	1
2 设计目的	1
3 设计指标	1
3.1 功能指标	1
3.1.1 输入/输出状态获取	1
3.1.2 输入/输出状态改变	1
3.1.3 ADC 模拟量采集	1
3.1.4 USART 通讯	2
3.1.5 Flash 存储特定数据	2
3.1.6 软件重启	2
3.1.7 ADC 标定值写入	2
3.1.8 设备波特率修改	2
3.1.9 设备在线升级	2
3.2.0 嵌入式实时操作系统	2
4 通信协议	3
4.1 寄存器列表	3
4.2 协议应用范例	5
4.3 指令集范例（测试建议）	5
4.4 输入/输出写入状态命令范例	7
4.5 波特率代码表	7
5 具体方案	8
5.1 外设/接口	8
5.1.1 IO 输入/输出	8
5.1.2 ADC 信息采集	8
5.1.3 硬件看门狗	8
5.1.4 485 自收发通讯	9
5.2 GPIO 规划表	10
6 软件流程图设计	11
6.1 流程方案设计	11
6.2 系统功能流程图	12
6.2.1 系统宏观程序流程图	12
6.2.2 串口中断处理流程图	13

6.2.3 波特率修改程序流程图	14
6.2.4 ModBus 响应程序流程图	15
6.2.5 去抖动/滤波去噪机制流程图	16
6.2.6 ADC 标定流程图	17
6.2.7 在线升级流程图	18
7 测调试规划	20
7.1 测试工具	20
7.2 功能测试项目	20
8 初期方案项目计划	20
9 开发总结	21

1 概述

1.1 背景

新版终端的设计模式，从系统全局来说相当于是使用新的方式将模块化设计进行规范化。系统硬件上的各个重要功能设计成各个硬件模块。通过必要的线路将其有机的组合在一起。新版终端通过网络总线的方式跟外面各个部件通讯来完成它采集类的业务功能。

2 设计目的

新版终端系统虽内置了 IO 处理模块，但 IO 线较长的链路不太适合直接接入到终端（如罐体上的底阀开关检测线引入到驾驶室的终端）。在新版的系统自研模块中，并没有一款专门处理 IO 采集的通讯模块。故从中长期布局来说，需要自研设计一款多功能 IO 口采集的新模块，并在下文中简要的描述这款模块的设计方案。

3 设计指标

3.1 功能指标

3.1.1 输入口状态获取

将输入口的状态值经过去抖动处理后,将其值存入寄存器中。供 ModBus 主机端调用查看。

3.1.2 输出口状态改变

在 ModBus 协议调用改变输出口状态时，将改写的状态值写入寄存器中并写入 Flash 空闲扇区进行数据固化。在重启后自动调用输出口状态改变函数，对输出口做出相应操作。将处理后的实际状态值重新写入寄存器中，可供 ModBus 主机端进行查看。

3.1.3 ADC 模拟量采集

自动完成传感器模拟量参数的采集，将所采集的数字量值经过与标定值进行处理后存入寄存器中，供 Modbus 主机端进行查看和初始标定。

3.1.4 USART 通讯

通过芯片的 USART1、USART3，将其设置为 RS485、UART 的发送和接收管脚用于向主机发送信息。详细管脚设置可见 **GPIO 规划表**。

3.1.5 Flash 存储特定数据

通过芯片内部的 Flash 扇区进行数据的存储和擦写，本系统将设备的地址和 PB 的状态以及 ADC 标定值进行数据的存储固化。以便达到系统断电后，仍然保存设备最后一次修改的数据，以便设备重启后正常工作。

3.1.6 软件重启

通过 RS485 通讯写入寄存器命令，系统进行解析命令并执行重启的操作。可实现主机端控制从机的重启操作。

3.1.7 ADC 标定值写入

通过 ModBus 命令，将目前的值与标准值进行计算，所得值存入 Flash 中，若无模拟量输入则为 0，若有输入量则将其与内存中的值进行数学计算。

结果为：假设输入的为 4mA(标准值介于 0x02E8 与 0x02E9 之间)的模拟量，则 ADC1、ADC2 在经过标定后，输出的值应该都会是标准值，误差 ± 1 （模拟量数值）；

3.1.8 设备波特率修改

通过 ModBus 命令，将需要修改的波特率代码进行写入。将修改值存入 Flash 中，可实现系统掉电后波特率与最后一次修改相同。

3.1.9 设备在线升级

通过 IAP（In Application Programming）方式，结合已有在用的升级架构来进行系统的移植开发。此功能能够实现在线的版本更新，只需通过云端即可达到新系统更新目的。

3.2.0 嵌入式实时操作系统

为实现系统的多线程任务操作，为系统增加嵌入式实时操作系统（FreeRTOS），可实现任务的多线程同时运行，增加系统的实时性。

4 通信协议

通讯协议采用 ModBus。有单寄存器读，多寄存器读，单寄存器写功能。

4.1 寄存器列表

寄存器地址	寄存器个数	寄存器名称	读写状态	功能码	数据定义																																
0001	1	开关量输入	R	0x03	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 高8位默认为0，低8位为PA0~7状态值。上表数据表示0x0010。 意义：输入口的1~8号端口，第5输入口使能。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0																						
0002	1	开关量输出	R W	0x03， 0x06	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table> 高12位默认为0，低4位为PB3~6状态值。上表数据表示0x000B。 意义：输出口的1~4号端口，除了3号端口，其它端口全部使能。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1																						
0003	1	ADC1 采集原始值	R	0x03	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table> 16位均代表ADC2采集值。上表数据表示0x02E8。 意义：采集量设计上4mA原始值为744，即标准值。 如ADC2的采集量为02E8（744）。表示经流4mA的电流。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0																						
0004	1	ADC2 采集原始值	R	0x03	同 0003 寄存器																																
0005	1	温度采集（物理量）	R	0x03	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table> 高8位默认为0，低8位为ADC1采集值。上表数据表示0x00DB。 意义：物理量由模拟量经过转换后乘以10倍放入寄存器。 如ADC1的采集量为219。表示温度21.9℃。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1																						
.....		待后期添加																																			
000A	1	ADC1 初始标定值存储	W	0x06	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> 高8位默认为0，低8位为标定差值。上表数据表示0x000F。 意义：标定差值为采集值与标准值的差值。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1																						
000B	1	ADC2 初始标定值存储	W	0x06	同 000A 寄存器																																
000C	1	设备波特率修改	W	0x06	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3</td></tr></table> 高8位默认为0，低8位为波特率值。上表数据表示0x0003。 意义：设备波特率为9600，具体见波特率代码表。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3																						

.....		待后期添加																																																																			
00AA	1	设备地址	R W	0x03, 0x06	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> 高8位默认为0，低8位为设备地址。上表数据表示0x00FF。 意义：当前设备地址为FF。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1																																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																						
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1																																																						
00BB	2	存储设备版本号	R	0x03	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 共2个寄存器，第1个寄存器数据表示0x2304。 第2个寄存器数据表示0x2500。 意义：当前设备版本为23042500，23年4月25日第一个版本。	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																						
0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0																																																						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																						
0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0																																																						
00CC	1	系统重启	W	0x06	写入 0xA55A,设备执行重启操作																																																																

4.2 协议应用范例

发送命令（格式）：

[设备地址][命令号][起始寄存器地址高 8 位][低 8 位]

[读取的寄存器数高 8 位][低 8 位][CRC 校验的低 8 位][CRC 校验的高 8 位]

例：[C1][03][00][01][00][01][CRC 低][CRC 高]

设备响应（格式）：

[设备地址][命令号 03][返回的字节个数][数据 1][数据 2]...[数据 n]

[CRC 校验的低 8 位][CRC 校验的高 8 位]

例：[C1][03][02][00][FF][CRC 低][CRC 高]

读取输入口状态功能范例：假设地址已经提前设定为 FF，命令如下：

C1 03 00 01 00 01 CRC

命令解析：

C1	03	00 01	00 01	CRC
从设备地址	功能码	起始寄存器	寄存器位数	CRC 校验码
1 字节， 设备的 485 地址	1 字节， 读寄存器	2 字节， 要开始读取的 寄存器地址	2 字节， 需要读取的寄 存器位数	2 字节， ModBusCRC16

返回命令：

C1 03 02 00 7F CRC

C1	03	02	00 7F	CRC
从设备地址	功能码	数据长度	数据	CRC 校验码
1 字节， 设备的 485 地址	1 字节， 读寄存器	2 字节， 后方有 2 字节 数据	2 字节数据， 7F 代表第 0-6 输入口使能	2 字节， ModBusCRC16

4.3 指令集范例（测试建议）

功能	发送指令	返回指令	说明
单寄存器读			
读取输入口状态	C1 03 00 01 00 01 CRC	C1 03 02 00 20 CRC	返回 2 个字节 20 代表第 6 输入口使能
读取输出口状态	C1 03 00 02 00 01 CRC	C1 03 02 00 0E CRC	返回 2 个字节 0E 代表第 2/3/4 输出口使能

读取 ADC1 采集值 (4-20mA)	C1 03 00 03 00 01 CRC	C1 03 02 02 D8 CRC	返回 2 个字节 代表 ADC1 的采集值
读取 ADC2 采集值 (4-20mA)	C1 03 00 04 00 01 CRC	C1 03 02 02 D9 CRC	返回 2 个字节 代表 ADC2 的采集值
读取 ADC 采集值 (温度)	C1 03 00 05 00 01 CRC	C1 03 02 00 DD CRC	返回 2 个字节 代表温度的采集值
读取设备地址	FA 03 00 AA 00 01 CRC	FA 03 02 00 FF CRC	返回 2 个字节 FF 代表设备地址
读取设备版本号	C1 03 00 BB 00 02 CRC	C1 03 04 23 04 25 00 CRC	返回 4 个字节 23042500 为日期内的 初始版本
单寄存器写			
修改输出口状态	见输出口写入状态命令范例		
写入 ADC1 标定值	C1 06 00 0A 00 01 CRC	C1 06 02 标定字节 1 标定字 节 2 CRC	返回 2 个字节 代表 ADC1 的标定值
写入 ADC2 标定值	C1 06 00 0B 00 01 CRC	C1 06 02 标定字节 1 标定字 节 2 CRC	返回 2 个字节 代表 ADC2 的标定值
修改设备地址	C1 06 00 AA 00 88 CRC	88 06 00 AA 00 88 CRC	按照原格式返回， 第 1 个字节为现地址
设备重启	C1 06 00 CC A5 5A CRC	无返回	设备进行重启
波特率修改	C1 06 00 0C 00 03	C1 06 00 0C 00 03	按照原格式返回， 修改波特率为：9600 具体可见 波特率代码表

备注：ADC 采集端接入 150Ω电阻。输入 4mA 的电流，芯片采集 0.6V 电压，采集到的十进制数值约为 744（计算公式： $0.6 = (744 * 3.3) / 4096$ ）。标定的目的是将电路的干扰进行修正，使其采集到的值在 744 周围不能误差过大。20mA 同理。

4.4 输出口写入状态命令范例

功能	修改码	发送指令	返回指令
全部置高	0F	C1 06 00 02 00 0F CRC	与发送指令相同
1 置低, 2/3/4 置高	0E	C1 06 00 02 00 0E CRC	与发送指令相同
2 置低, 1/3/4 置高	0D	C1 06 00 02 00 0D CRC	与发送指令相同
3 置低, 1/2/4 置高	0B	C1 06 00 02 00 0B CRC	与发送指令相同
4 置低, 1/2/3 置高	07	C1 06 00 02 00 07 CRC	与发送指令相同
1/4 置低, 2/3 置高	06	C1 06 00 02 00 06 CRC	与发送指令相同
2/3 置低, 1/4 置高	09	C1 06 00 02 00 09 CRC	与发送指令相同
2/4 置低, 1/3 置高	05	C1 06 00 02 00 05 CRC	与发送指令相同
3/4 置低, 1/2 置高	03	C1 06 00 02 00 03 CRC	与发送指令相同
1/2/3 置低, 4 置高	08	C1 06 00 02 00 08 CRC	与发送指令相同
1/2/4 置低, 3 置高	04	C1 06 00 02 00 04 CRC	与发送指令相同
1/3/4 置低, 2 置高	02	C1 06 00 02 00 02 CRC	与发送指令相同
2/3/4 置低, 1 置高	01	C1 06 00 02 00 01 CRC	与发送指令相同
全部置低	00	C1 06 00 02 00 00 CRC	与发送指令相同

4.5 波特率代码表

寄存器值	波特率
0x0000	波特率 1200
0x0001	波特率 2400
0x0002	波特率 4800
0x0003	波特率 9600
0x0004	波特率 19200
0x0005	波特率 38400
0x0006	波特率 57600
0x0007	波特率 115200

5 具体方案

5.1 外设/接口

5.1.1 IO 输入/输出

STM32F103RET6 具有多个 GPIO 端口，可用于 IO 输入/输出功能，可以通过库函数或直接操作寄存器来实现 IO 的具体功能查询和控制。

设置读 IO 输入口用于检测端口的使能状态。

设置读和写 IO 输出口用于接收主机指令进行使能状态的操作。

5.1.2 ADC 信息采集

STM32F103RET6 内置 ADC，支持多通道采集，可以实现对外部传感器信号的采集。

ADC 采集由于电路元件的差异以及诸多因素影响，需要检测并给每个通道设定固定的起始标定值，将其所采集到的模拟量经过与标定值的处理，各通道所展现的测量量的值为同一数值。本系统使用 ADC1 的通道 10、11、12。

ADC1	IO	ADC2	IO	ADC3	IO
通道0	PA0	通道0	PA0	通道0	PA0
通道1	PA1	通道1	PA1	通道1	PA1
通道2	PA2	通道2	PA2	通道2	PA2
通道3	PA3	通道3	PA3	通道3	PA3
通道4	PA4	通道4	PA4	通道4	PF6
通道5	PA5	通道5	PA5	通道5	PF7
通道6	PA6	通道6	PA6	通道6	PF8
通道7	PA7	通道7	PA7	通道7	PF9
通道8	PB0	通道8	PB0	通道8	PF10
通道9	PB1	通道9	PB1	通道9	连接内部VSS
通道10	PC0	通道10	PC0	通道10	PC0
通道11	PC1	通道11	PC1	通道11	PC1
通道12	PC2	通道12	PC2	通道12	PC2
通道13	PC3	通道13	PC3	通道13	PC3
通道14	PC4	通道14	PC4	通道14	连接内部VSS
通道15	PC5	通道15	PC5	通道15	连接内部VSS
通道16	连接内部温度传感器	通道16	连接内部VSS	通道16	连接内部VSS
通道17	连接内部Vrefint	通道17	连接内部VSS	通道17	连接内部VSS

5.1.3 硬件看门狗

正常状态下持续对 WDI 口端进行电平跳变以实现喂狗功能，从而达到系统稳定运行。在系统出现异常情况时自动重启系统，可以提高系统的稳定性。

5.1.4 485 自收发通讯

本系统采用 USART1 来做 RS485 通讯串口。数据接收与发送采用中断实现。通讯波特率默认为 9600bps，数据位 8bits，无校验位、1 个停止位。通过串口的中断将输入的字节存入缓存中，计算是否输入 8 字节。输入完成后状态关键字置 1，进行 ModBus 驱动函数处理。解析输入字节，执行对应功能，进行数据的自动发送响应。

5.2 GPIO 规划表

引脚号	端口组	端口	端口类型	备注
14	GPIOA	0	输入端口	I/O 输入端口 1
15	GPIOA	1	输入端口	I/O 输入端口 2
16	GPIOA	2	输入端口	I/O 输入端口 3
17	GPIOA	3	输入端口	I/O 输入端口 4
20	GPIOA	4	输入端口	I/O 输入端口 5
21	GPIOA	5	输入端口	I/O 输入端口 6
22	GPIOA	6	输入端口	I/O 输入端口 7
23	GPIOA	7	输入端口	I/O 输入端口 8
26	GPIOB	3	输出端口	I/O 输出端口 1
27	GPIOB	4	输出端口	I/O 输出端口 2
28	GPIOB	5	输出端口	I/O 输出端口 3
55	GPIOB	6	输出端口	I/O 输出端口 4
9	GPIOC	0	输入端口	ADC 温度采集接口(ADC1 的通道 10)
10	GPIOC	1	输入端口	ADC 采集接口 1(ADC1 的通道 11)
11	GPIOC	2	输入端口	ADC 采集接口 2(ADC1 的通道 12)
42	GPIOA	9	输出端口	RS485 通信的传输输出
43	GPIOA	10	输入端口	RS485 通信的接收输入
7	NRST			复位系统
56	GPIOB	12	输出端口	系统运行指示灯
44	GPIOA	11	输入端口	CAN 总线接收引脚
45	GPIOA	12	输出端口	CAN 总线发送引脚
46	GPIOA	13	输出端口	SWD 仿真接口数据线
49	GPIOA	14	输出端口	SWD 仿真接口时钟线
29	GPIOB	10	输出端口	串行通信外设的发送数据引脚
30	GPIOB	11	输入端口	串行通信外设的接收数据引脚
26	GPIOB	0	输出端口	喂狗端口(对端口进行电平跳变，进行喂狗操作)

6 软件流程图设计

6.1 流程方案设计

系统初始化

- 系统时钟初始化，设置时钟源、分频系数等参数；
- GPIO 初始化，设置 IO 口的输入输出方向、上下拉电阻等参数；
- USART 初始化，设置波特率、数据位、停止位、校验位等参数；
- ADC 初始化，设置采样通道、采样时间、转换结果对齐方式等参数；
- ModBus 协议初始化，设置设备地址、命令码等参数；
- 硬件看门狗初始化，设置喂狗定时器、超时时间等参数；
- 读取扇区是否修改设备地址；
- 读取扇区是否修改 GPIOB 状态；
- 读取扇区是否修改波特率；
- 进行 ADC 标定的初始化。

主循环

- 获取 ADC1 的 IN10/IN11/IN12 的值进行滤波处理,将其数值循环存入寄存器,等待 ModBus 的调取。涉及多寄存器读写及滤波处理,将 ADC 采集放在 while 中循环采集。若执行写入标定值操作,则将其目前滤波值进行写入 Flash,后续各频道采集值将与其各自的标定值进行数据计算。若不执行写入标定值操作,各频道的采集值将按照真实采集值进行输出；
- ModBus 功能函数,内部有:修改设备地址、获取输入/输出口状态、ADC 的信息查询、修改输出口状态、查询设备版本号；
- 定时进行喂狗,避免系统死锁或故障；
- 检测是否有新系统的更新；

中断服务程序

- USART 中断服务程序,用于接收 RS485 输入的命令码等(ModBus 协议)后,进行结果响应；

系统数据固化

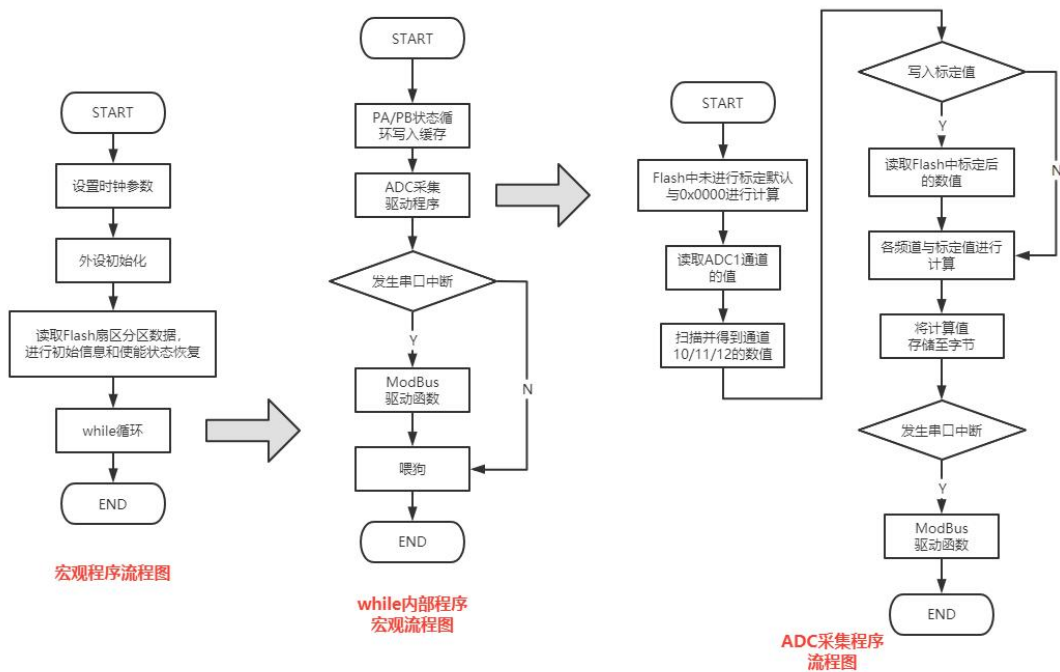
- 通过芯片 Flash 空闲扇区的分区擦写和写入需要保存的多个数据,以实现数据的掉电保存。达到断电重启后仍然可以达到断电前的状态。

6.2 系统功能流程图

6.2.1 系统宏观程序流程图

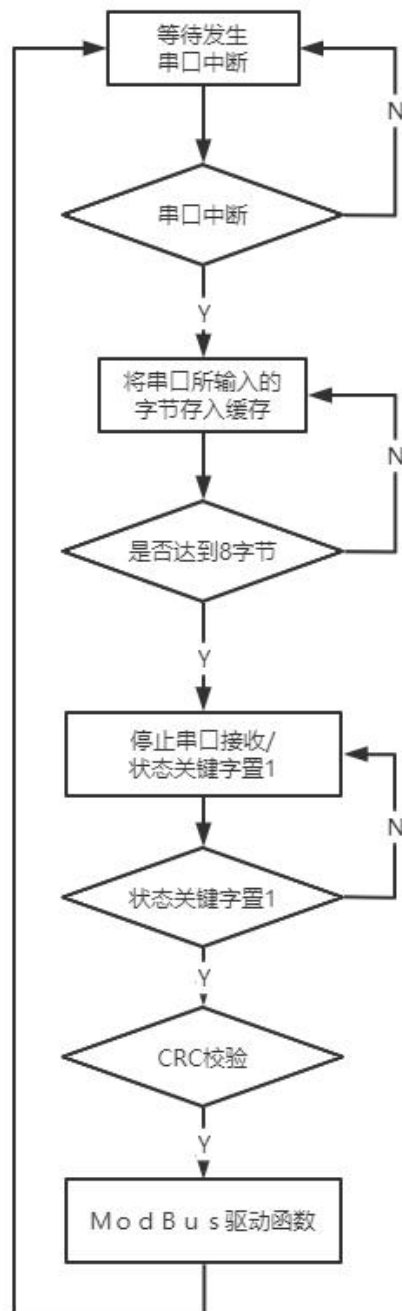
系统经过上电后，首先进行外设的初始化。包括设置 GPIO、ADC、TIM、USART 的原始参数：引脚位置、状态、使能时钟等。随后读取 Flash 扇区分区数据进行初始信息和使能状态的恢复。随后进入 while 循环，首先进行 PA/PB 的状态去抖动的循环读取和 ADC 采集滤波的驱动程序。涉及多寄存器读写及去抖和滤波处理，将读取输入/输出状态和 ADC 采集放在 while 中循环采集。将采集到的数据存入字节中，方便 ModBus 的 ADC 信息查询的调用和 ModBus 多字节的读取。

ADC 的初始标定，若不执行写入标定值的命令，默认写入 0x0000。输出的值为未执行标定的原始值（采集到的原始值）。若执行写入标定值，将目前的滤波得到的值（4mA 的模拟量）设置为标定值，并将其写入 Flash 固化。后续的 ADC 采集值将与 Flash 固化的标定值进行计算，以达到 ADC 初始标定的目的。



6.2.2 串口中断处理流程图

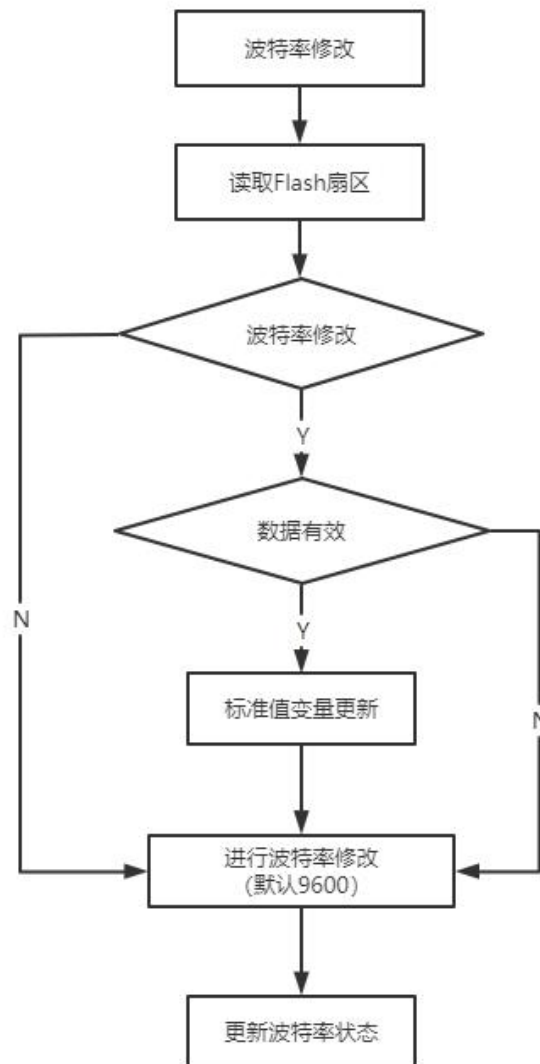
等待串口是否发生中断。如果发生中断,将输入的字节以中断的方式存入寄存器,判断是否达到 8 字节。如果是,则停止接收将状态关键字置 1,说明已经接收完成。进入 ModBus 功能驱动函数,ModBus 功能驱动函数执行完成后,从新等待发生串口中断。如果超时且未接收到足够的字节,则停止接收并处理已接收的数据。



6.2.3 波特率修改程序流程图

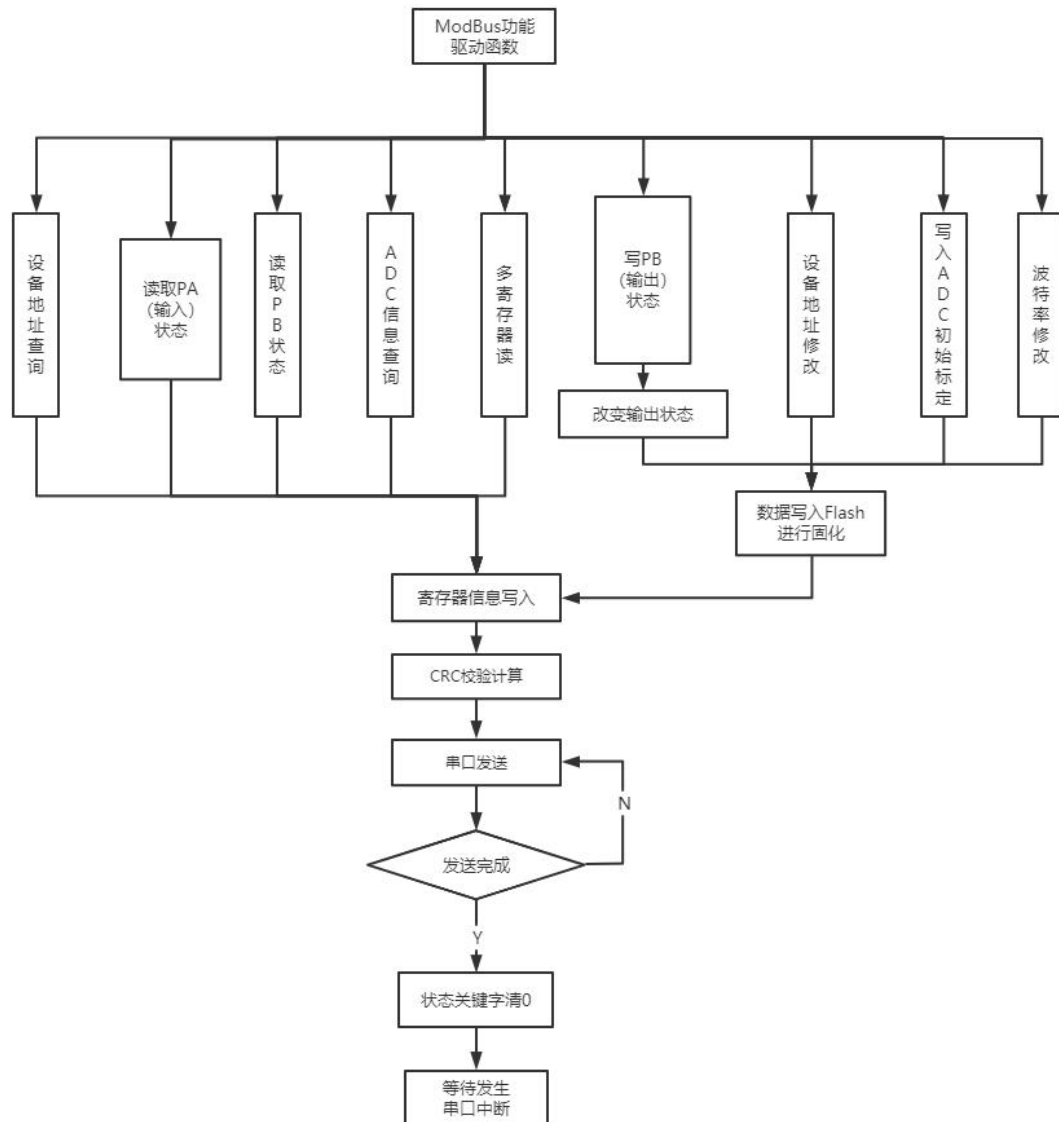
设备上电后，读取扇区进行判断是否进行过波特率修改。如果没有则按默认值 9600 设置。若修改，继续判断数值是否有效。若有效，则根据扇区数值进行波特率的初始化。

通过 ModBus 命令进行修改扇区数值，并更新修改的波特率。



6.2.4 ModBus 响应程序流程图

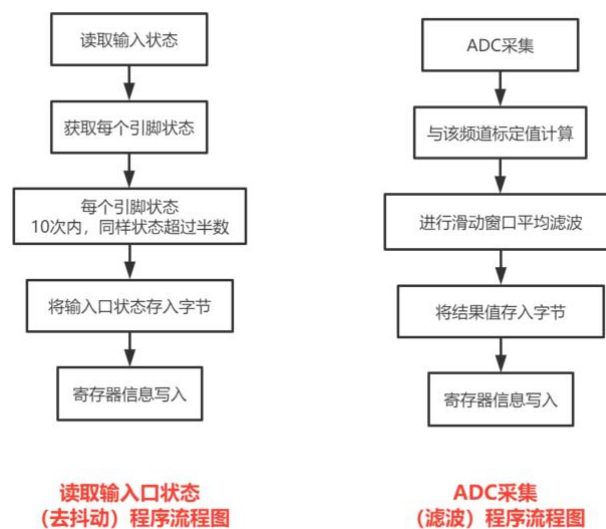
状态关键字置 1 后，进入 ModBus 功能驱动函数，通过接收到的字节来判断将要执行的子功能。子功能执行完后，将必要信息写入寄存器内。经过 CRC 校验后通过串口发送出来，通过判断 USART 的 SR 寄存器的“发送寄存器位”是否为 1，若为 1 则表示发送完成，若不为 1 则进行等待。最后，将状态寄存器从新清 0，从新等待发生串口中断。



6.2.5 去抖动/滤波去噪机制流程图

在读取 PA 口状态和采集 ADC 信息时，需考虑到 GPIOA 口的去抖动处理和 ADC 采集到的信息值的滤波处理。去抖动采用在 while 中延时 50ms 后将值放入寄存器中，循环进行读取以便达到主机调取时是去抖后的最新状态。

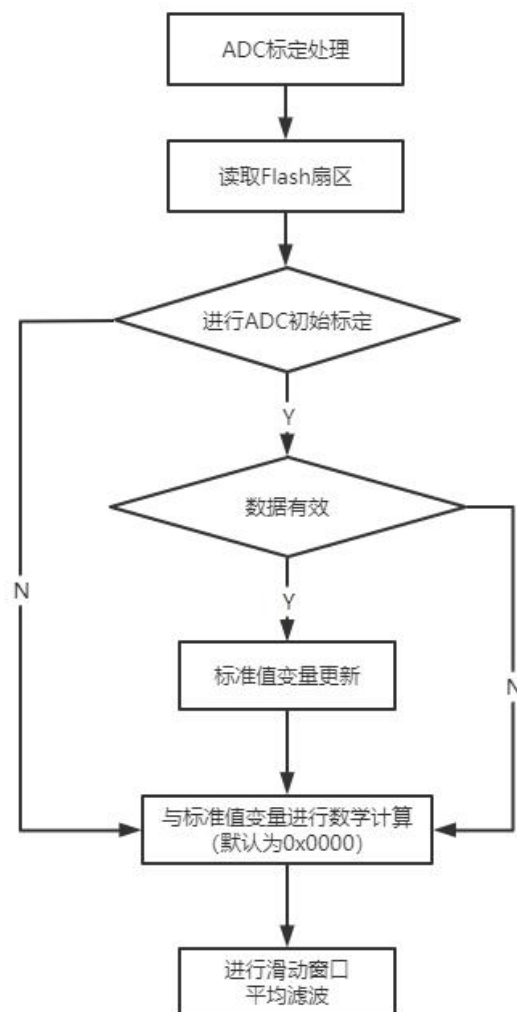
进行 ADC 信息采集时，进行了滑动窗口滤波处理，利用一个缓存区存储最近的原始数据，计算窗口内的平均值作为滤波后的结果。缓冲区索引则用于记录最新一次存储的数据位置，并在滑动窗口移动时更新。这样可以有效地平滑原始数据，并去除其中的噪声。



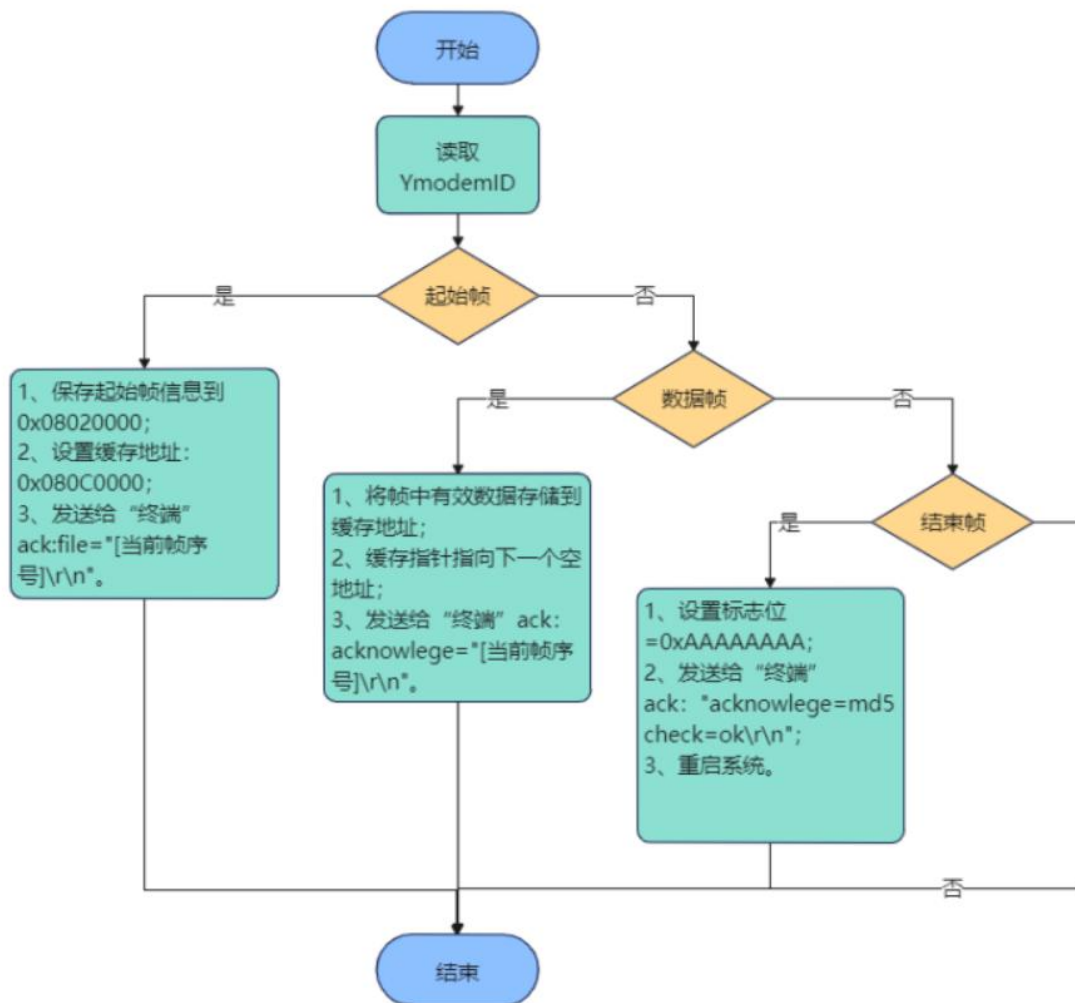
6.2.6 ADC 标定流程图

在系统上电后，读取 Flash 特定扇区数值，结果若为 0xFFFF 则代表未写入标定值，此时所得结果与标准值有误差。标定值变量默认为 0，此时需进行初始标定。

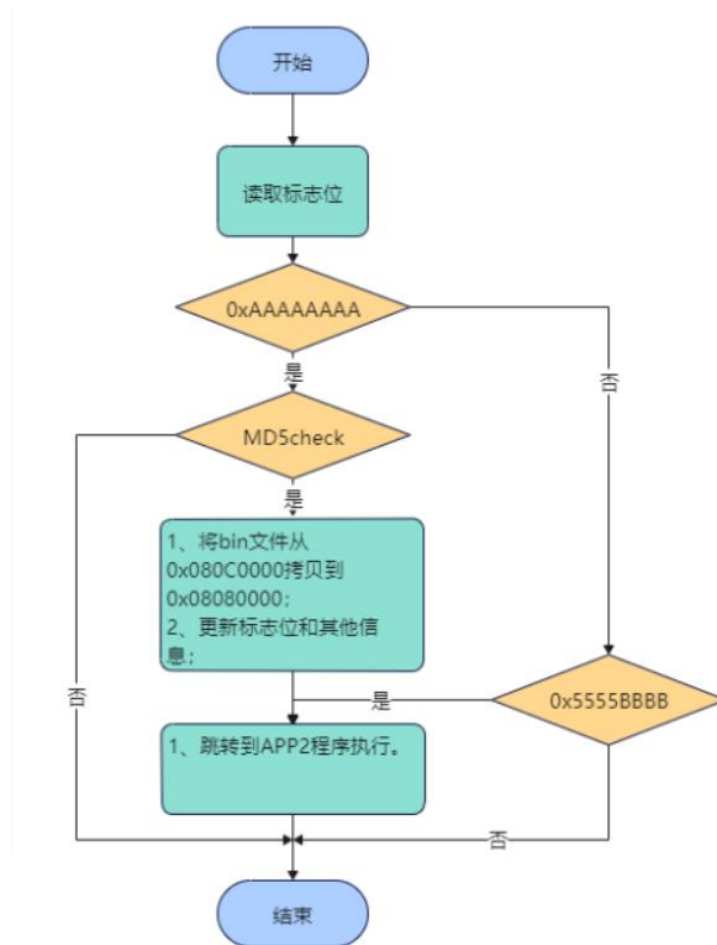
通过 ModBus 命令后，将所得的数值与标准值进行数学计算，将差值写入 Flash 并更新标定值变量，此时所得结果值与标准值相同，达到目的。



6.2.7 在线升级流程图



2、Start_BootLoader:



说明:

- A、APP1（hex 文件烧录）永远运行在“APP1 运行地址”：0x08000000;
- B、APP2（bin 文件升级）永远运行在“APP2 运行地址”：0x08020000;
- C、APP1 main 函数启动后执行第一个函数就是 Start_BootLoader,并读取“标志位”判断是否“正在升级 0xAAAAAAAA”或者“升级成功 0x5555BBBB”并跳转到 APP2;如是“升级成功”则直接跳转到 APP2;如是两者都否则退出函数继续执行 APP1。

7 测调试规划

7.1 测试工具

串口助手、RS485 转 USB 工具、USB 转 TTL 工具、ARM 仿真器、示波器、万用表、Line.exe(包括 MD5 处理工具和在线升级工具)。

7.2 功能测试项目

- RS485 通信功能；
- 读 IO 输入/输出（单寄存器读）功能；
- 读设备版本号（单寄存器读）功能；
- 写 IO 输出（单寄存器写）功能；
- ADC 采集功能（多寄存器读）功能；
- ADC 初始标定功能（单寄存器写）功能；
- 波特率修改（单寄存器写）功能；
- 多个寄存器枚举（多寄存器读）功能；
- 硬件看门狗复位功能；
- 在线升级功能；

8 初期方案项目计划

工作项	开发评估 时间段	主要工作内容	结果输出	主要执行人
系统新部件 IO 通信模块嵌入式软件研发	2023 年 4 月 1 日~ 2023 年 4 月 31 日	➤ 软件模块设计 ➤ 软件编码	➤ 软件设计方案 ➤ 通信协议方案	徐工、张浩

9 开发总结

本文介绍了一种基于 STM32F103RET6 处理器的 IO 采集通信模块软件设计方案，本方案根据初期系统开发方案“ims4 新版终端-系统新部件-IO 采集通讯模块-需求设计说明.pdf”进行起草设计，并在过程中对管脚等多方面细节进行改进。

该方案设计的采集系统具有读 IO 输入/输出、设置 IO 输出、查询和设置设备地址、ADC 采集、485 自收发通讯、读取设备版本号、硬件看门狗、修改波特率、支持在线升级和任务的多线程处理等功能。同时支持 GPIO 状态去抖动处理和 ADC 采集信息的滤波机制。

后期对 RS485 接收的驱动及诸多细节进行优化改进。并根据测试人员提出的问题进行优化和修复。目前可服务于新版系统使用。