

DM010

Modbus 说明书

V0.30

目录

一、 概述	1
二、 Modbu-RTU 协议简介	1
1、 帧格式	1
2、 CRC 校验	1
3、 地址	3
4、 功能代码	3
5、 读保持寄存器（03）	4
6、 读输入寄存器（04）	5
7、 写单个保持寄存器（06）	6
8、 写多个保持寄存器（16）	7
9、 响应时间	7
10、 异常响应	8
11、 参考资料	8
三、 保持寄存器概述	9
1、 分区	9
2、 信息映射	9
3、 写保护	9
四、 重量、状态相关保持寄存器（000 区）	11
1、 寄存器列表	11
2、 毛重值（0-1）、皮重值（2-3）和净重值（4-5）	12
3、 小数点位数（6）	12
4、 状态标志（7）	12
5、 毛重内码（8-9）、皮重内码（10-11）和净重内码（12-13）	13
6、 AD 码（14-15）	13
7、 锁定重量（16-17）	13
8、 传感器输出电压（18）	13
9、 模块类型（19）模块型号（20）	14
10、 硬件版本（21）	14
11、 软件版本（22），改制版本（23），	14
12、 软件校验和（25）	14
13、 用户参数校验和（26），标定参数校验和（27）	14
14、 用户参数设置次数（28-29），标定参数设置次数（30-31）	14
15、 参数健康状态（32）	14

1 6、 写操作允许状态 (33)	15
1 7、 模块 ID (34-35)	15
1 8、 锁定命令 (40)	15
1 9、 去皮命令 (41)	16
2 0、 置零命令 (42)	16
2 1、 进入固定参数通讯模式命令 (43)	16
2 2、 IO 模式状态设定 (44)	16
五、 通讯参数保持寄存器 (100 区)	17
1、 寄存器列表	17
2、 通讯模式 (100)	17
3、 通讯地址 (101)	17
4、 波特率 (102)	17
5、 校验方式 (103)	17
6、 通讯发送延时 (104)	18
六、 用户标定参数保持寄存器 (200 区)	19
1、 寄存器列表	19
2、 AD 转换速度 (200)	19
3、 滤波模式 (201)	19
4、 滤波强度 (202)	20
5、 判稳门限 (203)	20
6、 开机置零范围 (204)	20
7、 手动置零范围 (205)	20
8、 零点跟踪范围 (207)	20
9、 零点跟踪速度 (208)	20
1 0、 补偿模式 (209)	20
1 1、 用户分度值 (211)	21
1 2、 用户满量程 (212-213)	21
1 3、 用户修正系数 (214-215)	21
1 4、 用户重力加速度修正功能 (220-221)	21
1 5、 目标重量修正功能 (222-223)	22
1 6、 恢复默认用户参数 (224-225)	22
七、 标定操作保持寄存器 (300 区)	23
1、 寄存器列表	23
2、 标定分度值 (300)	23

3、 标定满量程（302-303）	23
4、 标定重力加速度（304-305）	24
5、 标定零点功能（310-311）	24
6、 标定加载点 1 功能（312-313）	24
7、 标定加载点 2~5 功能（314-315）	24
8、 备份标定参数（320-321）	24
9、 恢复标定参数（322-323）	25
10、 开启/退出写保护状态（340-341）	25
八、 标定参数保持寄存器（400 区）	26
1、 寄存器列表	26
2、 参数说明	26
九、 标定参数保持寄存器（500 区）	26
1、 寄存器列表	26
2、 参数说明	26

一、概述

DM010 数字模块（简称模块）支持 Modbus-RTU、自定义协议、定制协议等多种通讯协议。本手册详细介绍了 Modbus-RTU 协议及其寄存器映射。

模块将全部的功能都映射到 Modbus 的保持寄存器，如果使用主机支持 Modbus-RTU 协议或者使用 Modbus 库进行编程，就可以直接操作这些对象，不用关注协议实现的细节（第二节所述内容）。

对象类型	对象类型	访问类型	作用
离散量	单个比特	只读	模块不使用
线圈	单个比特	读写	模块不使用
输入寄存器	16-比特字	只读	读取模块信息
保持寄存器	16-比特字	读写	实现模块全部功能

二、Modbu-RTU 协议简介

1、帧格式

在 RTU 模式，报文帧由时长至少为 3.5 个字符时间的空闲间隔区分。在后续的部分，这个时间区间被称作 t3.5。

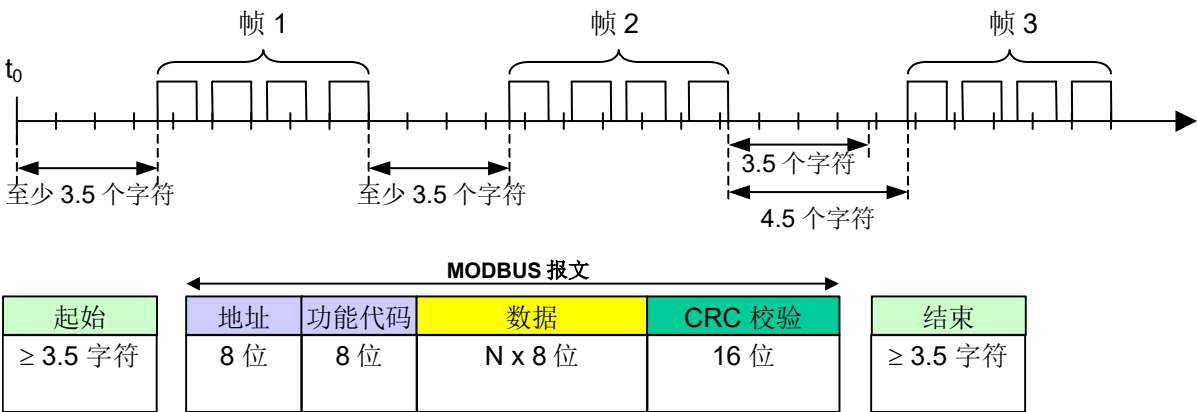
例如：

$3.5t(9600) = 1000 * 11 * 3.5 / 9600 \approx 4ms$

$3.5t(19200) = 1000 * 11 * 3.5 / 38400 \approx 2ms$

$3.5t(>19200)$ Modbu 协议规定统一为 1.75ms

对于某些主机（如 PC），如果实现不了精确定时，只需将帧间隔判断时间设置为大于 t3.5 即可。



2、CRC 校验

在 RTU 模式中，CRC 检验整个报文的内容（包括“地址”、“功能代码”和“数据”）。附加在报文之后时，首先附加低字节，然后是高字节。CRC 高字节为报文发送的最后一个

CRC 例子：

地址	功能码	数据				CRC 校验	
0x01	0x04	0x00	0x00	0x00	0x01	0x31	0xCA
0x01	0x03	0x00	0x7B	0x00	0x02	0xB4	0x12
0x0C	0x03	0x00	0xDE	0x00	0x79	0xE5	0x0F

3、地址

地址（即通讯地址）1~246，用于模块的寻址，模块只对地址相符的命令执行动作和应答。在一个总线上，每个模块都有唯一的地址，主机通过不同的地址来区别地访问各个模块。

地址 0 为广播地址，总线上所有模块都会接收并执行动作，但不应答。

地址 247 为模块保留地址，用户不要使用。

4、功能代码

功能代码（简称功能码）决定了请求的类型，也决定了数据字段的含义。不同功能码的请求帧和响应帧中数据字段的含义不尽相同，见下述各个功能码的说明。

访问模式	功能	功能码	模块支持
比特访问	读输入离散量	02	不支持
	读线圈	01	不支持
	写单个线圈	05	不支持
	写多个线圈	15	不支持
16 比特访问	读输入寄存器	04	注 2-4-1
	读多个寄存器	03	支持
	写单个寄存器	06	支持
	写多个寄存器	16	支持
	读/写多个寄存器	23	不支持
	屏蔽写寄存器	22	不支持

注 2-4-1：输入地址 0~37 等同于保持寄存器地址 0~37

注 2-4-2：为了叙述方便，下面各个功能码说明中，只包含功能码字段和数据字段，省略了地址字段和 CRC 校验字段。

5、读保持寄存器（03）

请求

功能码	1 个字节	0x03
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 125 (0x7D)

响应

功能码	1 个字节	0x03
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	

*N＝寄存器的数量

这是一个请求读寄存器 108-110 的实例：

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	03	功能	03
高起始地址	00	字节数	06
低起始地址	6B	寄存器值 Hi (108)	02
高寄存器编号	00	寄存器值 Lo (108)	2B
低寄存器编号	03	寄存器值 Hi (109)	00
		寄存器值 Lo (109)	00
		寄存器值 Hi (110)	00
		寄存器值 Lo (110)	64

将寄存器 108 的内容表示为两个十六进制字节值 02 2B, 或十进制 555。将寄存器 109-110 的内容分别表示为十六进制 00 00 和 00 64, 或十进制 0 和 100。

6、读输入寄存器（04）

请求

功能码	1 个字节	0x04
起始地址	2 个字节	0x0000 至 0xFFFF
输入寄存器数量	2 个字节	0x0001 至 0x007D

响应

功能码	1 个字节	0x04
字节数	1 个字节	2×N*
输入寄存器	N*×2 个字节	

*N=输入寄存器的数量

这是一个请求读输入寄存器 9 的实例：

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	04	功能	04
起始地址 Hi	00	字节数	02
起始地址 Lo	08	输入寄存器 9 Hi	00
输入寄存器数量 Hi	00	输入寄存器 9 Lo	0A
输入寄存器数量 Lo	01		

将输入寄存器 9 的内容表示为两个十六进制字节值 00 0A，或十进制 10。

7、写单个保持寄存器（06）

请求

功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

响应

功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

这是一个请求将十六进制 00 03 写入寄存器 2 的实例：

请求		响应	
域名	（十六进制）	域名	（十六进制）
功能	06	功能	06
寄存器地址 Hi	00	输出地址 Hi	00
寄存器地址 Lo	01	输出地址 Lo	01
寄存器值 Hi	00	输出值 Hi	00
寄存器值 Lo	03	输出值 Lo	03

8、写多个保持寄存器（16）

请求 PDU

功能码	1 个字节	0x10
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	0x0001 至 0x0078
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	值

*N=寄存器数量

响应 PDU

功能码	1 个字节	0x10
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 123 (0x7B)

这是一个请求将十六进制 00 0A 和 01 02 写入以 2 开始的两个寄存器的实例：

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	10	功能	10
起始地址 Hi	00	起始地址 Hi	00
起始地址 Lo	01	起始地址 Lo	01
寄存器数量 Hi	00	寄存器数量 Hi	00
寄存器数量 Lo	02	寄存器数量 Lo	02
字节数	04		
寄存器值 Hi	00		
寄存器值 Lo	0A		
寄存器值 Hi	01		
寄存器值 Lo	02		

9、响应时间

模块读操作的固有最大响应时间为 20ms，写操作的固有最大响应时间为 120ms。实际响应时间还要加上发送延时参数设定时间、通讯的延时和 Modbus-RTU 协议要求的 3.5T。

一般建议设置 Modbus 超时时间为 200ms 以上发送延时。

1 0、异常响应

当主机向模块发送请求时，可能出现下面四种可能事件之一：

- 如果模块接收到无通信错误的请求，并且可以正常地处理询问，那么模块将返回一个正常响应。
- 如果模块接收到无通信错误的请求，但不能处理这个请求，模块将返回一个异常响应。
- 如果由于通信错误，模块没有接收到请求，那么不能返回响应。主机将超时错误。
- 如果模块收到请求，但是检测到一个通信错误（奇偶校验、CRC、...），那么不能返回响应。主机将超时错误。

异常响应报文有两个与正常响应不同的域：

功能码域：

在正常响应中，响应功能码与最初请求的功能码相同。

在异常响应中，响应功能码 = 最初请求的功能码 + 0x80。

数据域：

在正常响应中，数据域返回请求中要求的数据。

在异常响应中，数据域返回异常码。

异常码的列表：

MODBUS 异常码		
代码	名称	含义
01	非法功能	接收到的功能码是不可允许的操作。
02	非法数据地址	接收到的数据地址是不可允许的地址。
03	非法数据值	询问中包括的值是不可允许的值
04	从站设备故障	模块功能执行失败

一般而言，异常码 01、02 和 03 往往出现在编写程序时，使用了模块不支持的功能、访问了模块不支持的寄存器或者编写错误。而异常码 04 往往出现在使用中，表示不符合操作条件（比如发送了置零命令，但是当前重量超过了手动置零范围）。在上位机编程时，最好对异常码 04 做出相应的处理，提示用户出错的原因。

1 1、参考资料

详细的协议请参考以下资料：

《Modbus 应用协议》（国标 GB/T19582.1-2008）

《Modbus 协议在串行链路上的实现指南》（国标 GB/T19582.2-2008）

三、保持寄存器概述

1、分区

保持寄存器以字（16bit）为单位，可读写。根据功能不同划分为六个不同的区，每个区包含 50 个字，用最小的地址来作为区的名字，000 区表示地址 00~49 的区、100 区表示 100~149 的区，以此类推。各区的功能如下：

000 区	称重信息	读取重量、状态，执行置零、去皮
100 区	通讯参数	读写通讯参数
200 区	用户参数	读写用户标定参数，执行用户标定
300 区	标定功能	执行厂商标定
400 区	标定参数	读写厂商标定参数
500 区	高速采样	高速采样缓冲区

2、信息映射

模块信息映射分为两类，一类是映射到一个字（两个字节），另一类是映射到两个字（四个字节）。对应映射到两个字的信息，默认为“高字在前”。比如毛重值映射到保持寄存器 0 和 1 这两个字，那么保持寄存器 0 保存的就是毛重的高 2 个字节（即高 16 位），保持寄存器 1 保存的就是毛重的低 2 个字节（即低 16 位）。

对于某些只支持低字在前的系统，本模块还提供了低字在前的映射区，他们是 1000 区、1100 区、1200 区、1300 区、1400 区、1500 区，其含义功能和 000 区、100 区、200 区、300 区、400 区、500 区相同，只不过是两字（四字节）信息时低字在前（不管读或者写都是低字在前）。

3、写保护

为了提高参数存储的可靠性，避免参数被意外改写，模块提供了参数软件写保护功能。当写保护功能开启时，100 区~400 区（包括 1100 区到 1400 区）的写和执行功能被禁止。000 区、500 区（包括 1000 区、1500 区）不受影响。

模块提供两级写开关机制，一个总写开关和各个区的子写开关。这两个开关是逻辑或的关系，只要有一个开关开，那么对应的区就是允许写状态。也就是说总写开关为开，那么所有区都是允许写状态。如果总写开关为关，那么各个区的状态由各区自己的写开关决定。

总写开关的状态是断电保持的，这样用户可以选择是否使用写保护功能。一般来说参数通过 Modbus 命令被意外改写的可能性是比较低的，推荐不使用写保护功能，这样参数的读写和执行功能就会容易一点。在实际应用中，当发现参数被意外的改写同时对应的参数修改次数变大时，才可能是参数通过 Modbus 命令被意外改写，这个时候就可以把总写开关设置为关，来避免意外的改写。

当总写开关被关闭时，各个区的状态由各区自己的写开关决定。各个区的写开关的状态

是断电不保存的，每次上电后都处于关的状态。当总写开关被关闭时，要写某个区或者执行某个区的功能，则需打开这个区的子写开关。子写开关打开后，该区就一直处于允许写状态，一直保持到断电，或者执行软件复位命令。

四、重量、状态相关保持寄存器（000 区）

1、寄存器列表

地址	PLC 地址	字数	功能	默认值	备注
0	40001	2	毛重值		只读
2	40003	2	皮重值		可读写
4	40005	2	净重值		只读
6	40007	1	小数点位数		只读
7	40008	1	状态标志		只读
8	40009	2	毛重内码		只读
10	40011	2	皮重内码		只读
12	40013	2	净重内码		只读
14	40015	2	AD 码		只读
16	40017	2	锁定重量		只读
18	40019	1	传感器输出电压（uV）		只读
19	40020	1	模块类别		只读
20	40021	1	模块型号		只读
21	40022	1	硬件版本		只读
22	40023	1	软件版本		只读
23	40024	1	改制版本		只读
24	40025	1	备用		只读
25	40026	1	软件校验和		只读
26	40027	1	用户参数校验和		只读
27	40028	1	标定参数校验和		只读
28	40029	2	用户参数设置次数		只读
30	40031	2	标定参数设置次数		只读
32	40033	1	参数健康状态		只读
33	40034	1	写操作允许状态		只读
34	40035	4	模块 ID 号		只读
38	40039	2	备用		只读
40	40041	1	锁定命令		只写，不支持连续写
41	40042	1	去皮命令		只写，不支持连续写

DM010 Modbus 说明书

42	40043	1	置零命令		只写，不支持连续写
43	40044	1	进入固定通讯参数通讯模式		只写，不支持连续写
44	40045	1	IO 模式状态设定		只写，不支持连续写

2、毛重值（0-1）、皮重值（2-3）和净重值（4-5）

毛重值、皮重值和净重值为模块当前的对应“重量数值”。

“重量数值”为对应重量不包括小数点的数值（小数点位数需要从保持寄存器地址 6 中获取）。例如毛重为 21kg，小数点为 2 位，那么毛重的重量数据值就是 2100；净重为 123kg，小数点为 3 位，那么净重的重量数值就是 123000。

“重量数值”为 32bit 补码占两个字，高字在前。

注 3-2-1（两个字补码的表示举例）：

数值	高字	低字
123456	0x0001	0xE240
32768	0x0000	0x8000
32767	0x0000	0x7FFF(32767)
12345	0x0000	0x3039(12345)
1	0x0000	0x0001(1)
0	0x0000	0x0000(0)
-1	0xFFFF	0xFFFF(-1)
-12345	0xFFFF	0xCFC7(-12345)
-32768	0xFFFF	0x8000(-32768)
-32769	0xFFFF	0x7FFF
-123456	0xFFFE	0x1DC0

当 $-32768 \leq \text{数值} \leq 32767$ 时，可以忽略高字。

皮重值可写入，相当于预置皮重功能。

3、小数点位数（6）

和“重量数值”一起组成重量（即毛重、皮重和净重）。

4、状态标志（7）

状态标志共 16 位，具体含义如下：

bit0: 零位

当毛重小于等于 0.25 个分度值时，零位标志位为 1。

bit1: 稳定

重量变化小于判稳范围 1.6 秒后，稳定标志位为 1。

bit2: 去皮

当皮重不为 0 时，皮重标志位为 1。

bit3: 超载

当毛重大于满量程加 9 个分度值时，超载标志位为 1。当超载标志为 1 时，主机必须停

止显示重量，并做出相应的报警提醒用户，避免压坏传感器。

bit4: 开机

模块上电会执行开机置零（如果设置开机置零有效的话），此时开机标志位为 1，输出的重量信息无效（固定为 0）。

bit5: 模拟部分异常

传感器或者模块 ADC 模拟部分损坏时，此标志位为 1，输出的重量信息无效。

bit6: ADC 芯片损坏

模块 ADC 芯片损坏时，此标志位为 1，输出的重量信息无效。

bit8: 锁定标志

锁定重量处于锁定状态时，锁定标志位为 1。

bit9: 使用开机初始通讯参数

模块开机后的 300ms 内会使用固定的通讯参数(通讯模式=Modbus-RTU, 通讯地址=247, 波特率=38400, 校验方式=偶校验, 发送延时=0)等待接收命令，如果在 300ms 内接收到指定的命令，则将继续使用这些通讯参数。否则的话就会使用，设置的通讯参数，进入正常通讯模式。这个标志位=1，则表明模块使用的是开机固定通讯参数。=0 则使用的是设置的通讯参数。这个功能主要用于，模块的通讯参数被设置到一个未知状态时，快速恢复通讯参数。

bit10: 开关量输入状态

bit11: 新 AD 采样有效标志。ADC 新采样有效时置 1，读取过 000 区（或 1000 区）后清零。当读取周期小于 ADC 采样周期（1/ADC 转换速度）时，可能会重复读取上次的采样结果，这个时候这个标志位就是 0。如果读取周期足够短，通过判断这个标志位可以获取全部有效的采样结果。也就是说采样速度设置到 20 次每秒，每秒就能准确的获取 20 个有效采样结果。

5、毛重内码（8-9）、皮重内码（10-11）和净重内码（12-13）

内码提供了高分辨率的重量信息，主要用于标定调试和计量检定。

内码 = 重量 × 20 ÷ 分度值

例如：净重=6.000kg，分度值=0.002kg，那么净重内码= 6.000 * 20 ÷ 0.002=60000

6、AD 码（14-15）

直接输出模块内部 ADC 输出的数据，数据为偏移二进制码，0 表示负向最大值（约 -20mV），0x800000 表示 0（0mV），0xFFFFF 表示正向最大值（约 20mV）。

7、锁定重量（16-17）

锁定重量配合锁定功能使用。

在不锁定时锁定重量为 0；锁定时锁定重量，根据锁定命令为所指定的毛重、净重等内容。

8、传感器输出电压（18）

为传感器输出的电压值，单位为 uV。精度约为 5%，

可用于标定前或者标定参数丢失时，判断 ADC 和传感器硬件的好坏。

9、模块类型（19）模块型号（20）

模块类型为产品名称的前缀：本模块为 2。

模块型号为产品名称的数字-10，本模块为 0。

1 0、硬件版本（21）

硬件版本主要用于软硬件匹配，同一个硬件版本的软件都是兼容的，反之则不能使用。

在 IAP 升级软件时，首先要核对硬件版本是否匹配，匹配的话才能升级。

1 1、软件版本（22），改制版本（23），

软件版本主要用于判断软件功能。比如说某个功能需要大于某个软件版本才具备。

改制版本用于判断软件定制的内容，标准版改制版本为 0。

1 2、软件校验和（25）

软件校验和主要用于判断程序的完整性，判断程序有没有丢失或者被非法修改。

1 3、用户参数校验和（26），标定参数校验和（27）

用户参数校验和是通讯参数和用户参数的综合校验和，他反映这两组参数是否被修改过。标定参数校验和是标定参数的校验和，他反映标定参数是否被修改过。

1 4、用户参数设置次数（28-29），标定参数设置次数（30-31）

用户参数设置次数和厂商参数设置次数，反映了对应的参数被写入的次数，根据指令不同，一次 MODBUS 写操作，可能会使两组参数同时被设置，或一组参数被设置多次。

当用户标定调试完毕后，应该保存用户参数校验和、标定参数校验和、用户参数设置次数和标定参数设置次数，每次开机和当前读出的内容做比对，如果不完全一致，那么证明对应的参数可能被意外修改。如果参数校验和和设置次数同时为 0，则表明模块参数丢失，使用的是默认参数。

1 5、参数健康状态（32）

模块参数都会有多个副本，某个副本意外损坏不影响参数的准确性。但是会降低参数存储的可靠性，如果全部被破坏，参数就真的丢失了，将使用默认参数。

该参数正常状态应为全 0，如果某一位为 1 了，则证明某个参数副本被意外损坏了，此时建议提示用户手动的执行对应的存储操作，来恢复这个副本。

位	含义	备注
bit0	用户参数副本 1	开机后如果这四个副本都被破坏 则模块使用默认用户参数
bit1	用户参数副本 2	
bit2	用户参数副本 3	
bit3	用户参数副本 4	
bit4	标定参数副本 1	开机后如果这四个副本都被破坏

bit5	标定参数副本 2	则模块使用备份标定参数副本
bit6	标定参数副本 3	
bit7	标定参数副本 4	
bit8	备份标定参数副本 1	如果备份标定参数副本也被都被破坏则模块使用默认标定参数
bit9	备份标定参数副本 2	
bit13	内部 RAM 硬件损坏	
bit14	内部 XRAM 硬件损坏	
bit15	非法软件	

1 6、写操作允许状态（33）

为了防止参数被误改写，区域 100~400 可以禁止写操作。当全区域允许位和区域对应的允许位都为 0 时，该区域禁止写操作。

开机各区的允许位和全区允许位相同。在某区的 49 号地址写入 12345，将该区的允许位置为 1，可以临时执行该区的写操作（写 49 号地址本身不受写操作允许控制）

位	含义	备注
bit1	区域 100 允许写操作	=1 允许，=0 禁止
bit2	区域 200 允许写操作	=1 允许，=0 禁止
bit3	区域 300 允许写操作	=1 允许，=0 禁止
bit4	区域 400 允许写操作	=1 允许，=0 禁止
bit7	全区域允许写操作	=1 允许，=0 禁止

1 7、模块 ID（34-35）

模块 ID 为模块的唯一标识。

1 8、锁定命令（40）

只能写，读取固定为 0。一条 Modbus 指令，只能执行一种命令。连续写多个保持寄存器，则无效（下同）。

写入内容	作用	备注
11100	取消锁定	锁定重量设置为为 0，锁定重量 0
11111	锁定毛重	锁定重量设置为当前毛重，锁定标志为 1
11122	锁定净重	锁定重量设置为当前净重，锁定标志为 1
11133	锁定毛重内码	锁定重量设置为当前毛重内码，锁定标志为 1
11144	锁定净重内码	锁定重量设置为当前净重内码，锁定标志为 1
11155	锁定 AD 码	锁定重量设置为当前 AD 码，锁定标志为 1

1 9、去皮命令（41）

写入内容	作用	备注
22200	取消皮重	将皮重设置为 0
22211	判稳去皮	稳定的话，则将皮重设置为当前毛重，否则失败
22222	不判稳去皮	将皮重设置为当前毛重

当重量信息无效时，返回失败（异常码=从站设备故障）。

2 0、置零命令（42）

写入内容	作用	备注
33300	取消置零	恢复开机零点
33311	置零	毛重处于手动置零范围，置零成功，否则失败
33322	强制置零	不判置零范围，直接置零。

当重量信息无效时，返回失败（异常码=从站设备故障）。

2 1、进入固定参数通讯模式命令（43）

写入内容	作用	备注
44400	退出	写入任何通讯参数也会退出固定参数通讯模式
44411	进入	只有在开机 300ms 内发有效

2 2、IO 模式状态设定（44）

写入内容	作用	备注
55500	弱上拉输入	上电默认
55511	推挽输出 0	
55522	推挽输出 1	

五、通讯参数保持寄存器（100 区）

1、寄存器列表

地址	PLC 地址	字数	功能	默认值	备注
100	40101	1	通讯模式	0	0: Modbus-RTU 1: 定制通讯模式
101	40102	1	通讯地址	1	1~246
102	40103	1	波特率	5	0:1200 1:2400 2:4800 3:9600 4:19200 5:38400 6:57600 7:115200
103	40104	1	校验方式	2	0:无校验 1:奇校验 2:偶校验 3:“1”校验
104	40105	1	通讯发送延时	0	0~120 单位为 ms

2、通讯模式（100）

模块支持 Modbus-RTU，此时通讯模式应该设置为 0。也支持定制通讯协议（标准版没有），此时通讯模式应该设置为 1。模块默认使用 Modbus-RTU。

模块的参数的默认值，是模块出厂时的设定值，也是所有参数备份被破坏后使用的默认值。一般而言，如果不是主动的设置，模块参数只会有两种情况。要么保持正确的设定值，要么就是被破坏为默认值。所以如果不是特别需要，直接使用默认值，会提高参数的可靠性。

3、通讯地址（101）

通讯地址的范围是 1~246。地址 0 作为广播地址，所有在线的模块都会接收命令，但都不会应答。地址 247 作为，开机固定参数通讯模式使用，也无法设置。模块默认地址是 1。

4、波特率（102）

模块支持 1200~115200 八种波特率。波特率越慢，则通讯的抗干扰能力、传输距离都会有所提升，但是响应速度也会变慢。一般推荐设置为 9600~38400 这个范围，模块默认使用的是 38400。

5、校验方式（103）

模块支持 4 种校验方式，其中方式 1~3 都是 8 位数据位+1 位校验位，而方式 0 使用的是 8 位数据位无校验位。Modbus 标准要求当不使用校验位时使用 1 填充（就是方式 3: “1”

校验)，从而保证通讯传输的位数相同。但 PC 编程选择无校验位时，就代表不发送校验位，相当于方式 0。所以当使用无校验时，可尝试方式 0 或者方式 3，来适应不同的无校验模式。

推荐用户使用方式 2 偶校验，模块默认的参数也是偶校验。

6、通讯发送延时（104）

通讯发送延时（0~120），单位为 ms。是在模块发送数据前额外添加的延时，他会使响应时间变长。

对于某些 RS232 转 RS485 或者 USB 转 RS485 的转换器或者数据线，他们的发送和接收切换需要一个延时，模块应答的太快会造成不能通讯或者通讯不可靠。这个时候就需要设置通讯发送延时，一般推荐设置为 10（ms）。对于原生的 RS485 接口，发送接收切换没有延时，所以通讯发送延时推荐设置为 0。

六、用户标定参数保持寄存器（200 区）

1、寄存器列表

地址	PLC 地址	字数	功能	默认值	备注
200	40201	1	ADC 转换速度	0	0: 10 次/秒 1: 20 次/秒 2: 40 次/秒 3: 80 次/秒 4: 160 次/秒
201	40202	1	滤波模式	0	0: 滑动滤波
202	40203	1	滤波强度	16	1~32 滑动滤波的平均次数
203	40204	1	判稳门限	1	1~99
204	40205	1	开机置零范围（FULL%）	20	0~100
205	40206	1	手动置零范围（FULL%）	4	0~100
206	40207	1	备用	0	
207	40208	1	零点跟踪范围	1	
208	40209	1	零点跟踪速度	0	
209	40210	1	补偿模式	1	
210	40211	1	备用	0	
211	40212	1	用户分度值	标定值	
212	40213	2	用户满量程	标定值	
214	40215	2	用户修正系数	1000000	
216	40217	4	备用	0	
220	40221	2	用户重力加速度修正功能	0	只写，读出是 0
222	40223	2	目标重量修正功能	0	只写，读出是 0
224	40225	2	恢复默认用户参数	0	只写，读出是 0

2、AD 转换速度（200）

AD 转换速度有 5 种 10、20、40、80、160 次/秒，分别用 0~4 来表示。AD 转换速度越快，称重响应越快，但是稳定性会变差。

3、滤波模式（201）

固定为 0 滑动滤波。可以定制别的滤波模式。

4、滤波强度（202）

滑动滤波模式下为滑动滤波的平均次数。可以设置为 1~32。滤波强度设置的越大，则稳定性越好，但是称重响应变慢。

5、判稳门限（203）

判稳门限是判断稳定的依据，可以设置为 0~99。一般设置为 1，如果实际称重过程中振动或晃动较大，使得模块一直不能判断到稳定，此时可以调大这个参数，直到会稳定为止。

6、开机置零范围（204）

开机置零范围（0~100），单位为满量程的百分比，例如满量程为 500kg，开机置零范围为 20，那么开机置零的范围值就是±100kg，模块重新上电后，如果毛重小于±100kg，模块就把该重量作为零点，输出毛重为 0。

7、手动置零范围（205）

手动置零范围（0~100），单位为满量程的百分比，它影响着置零命令能否成功。如果当前毛重（相对于开机零点）小于手动置零范围值时，置零命令有效，模块把当前毛重作为零点，输出毛重为 0。

8、零点跟踪范围（207）

数值	0	1	2	3	4	5	6	7	8	9
范围	不跟踪	0.5d	1d	1.5d	2d	2.5d	3d	3.5d	4d	4.5d

“d”为分度值。当重量处于零点跟踪范围，模块逐步修正零点，使输出趋向于零。

9、零点跟踪速度（208）

零点跟踪速度（0~3），数值越大零点修正的速度就越快。零点跟踪速度快，可能会造成加载小分量，被跟掉一定的数值，使重量稍微偏轻。

10、补偿模式（209）

补偿模式参数，用于配置蠕变补偿（bit0）和闪变点抑制（bit1）两个功能是否使用。

蠕变补偿（bit0=0：不使用；bit0=1：使用）

传感器长时间加载，会引起零点的缓慢变化，当加载的重量处于零点跟踪范围外时，这种变化就不能自动的被修正。蠕变补偿就是用于修正这种缓慢变化的。

闪变点抑制（bit1=0：不使用；bit1=1：使用）

当重量处于四舍五入的临界点时，输出可能会上下跳动一个分度，称之为闪变点。如果不希望看到这种跳动，则可以选择闪变点抑制。

1 1、用户分度值（211）

分度值决定了模块的分辨率，为标定分度值和用户分度值，标定分度值是在标定时候选择的分度值，用户分度值则是实际使用的分度值。用户分度值受标定分度值的限制，设置范围如下：标定分度值-3≤用户分度值≤标定分度值+3。

分度值参数设置范围是 0~23，分别表示不同的分度值，见下表。

参数	0	1	2	3	4	5	6	7
分度值	0.0001	0.0002	0.0005	0.001	0.002	0.005	0.01	0.02
参数	8	9	10	11	12	13	14	15
分度值	0.05	0.1	0.2	0.5	1	2	5	10
参数	16	17	18	19	20	21	22	23
分度值	20	50	100	200	500	1000	2000	5000

1 2、用户满量程（212-213）

满量程决定模块的最大称量，分为标定满量程和用户满量程，标定满量程是在标定的时候设定的满量程，用户满量程是实际使用的满量程。用户满量程受标定满量程的限制，设置范围如下：用户满量程≤标定满量程。

满量程是一个不带小数点的数值，如果分度值选用的是带小数点的，那么满量程使用对应的小数点位数。比如分度值设置为 3（0.001），它是 3 位小数点。那么满量程也是三位小数点，这个时候满量程如果为 30000，它代表的含义就是 30.000。

1 3、用户修正系数（214-215）

用户修正系数用于修正标定的误差，它固定 6 位小数点。标定完成后，用户修正系数为 1000000，就是 1.000000，代表不修正。如果把它设置为别的值，则重量就会成比例的变化。比如设置为 900000（0.900000），那么重量就会变为原来的 0.9 倍。它的设置范围是 250000~4000000（0.25~4）。

执行用户修正有三种方法，一是直接修改用户修正系数这个参数，二是通过用户重力加速度修正功能来自动的修改这个系数，三是通过目标重量修正功能来自动的修改这个系数。

1 4、用户重力加速度修正功能（220-221）

使用地重力加速度如果和标定地重力加速度不同，会带来测量误差，通过输入用户地的重力加速度，自动的计算用户修正系数，从而修正这个误差。

重力加速度值，固定六位小数点，设置范围是 9300000~10300000（9.3~10.3）。如果写入的用户重力加速度在这个范围内，同时标定点也设置标定重力加速度，那么就会执行修正功能。否则会返回 Modbus 设备错误。该功能只能通过，一次写入这两个寄存器的 Modbus 指令来实现，写一个或者写大于两个都会忽略不执行。

1 5、目标重量修正功能（222-223）

当发现模块输出值和实际重量有了偏差，除了可以手动的修改用户修正系数来修正，还可以通过目标重量修正功能自动的来修改用户修正系数。

目标重量修正功能，就是通过写入当前的正确重量值，来修改用户修正系数。比如当前重量是 900，但模块输出是 1000，那么可以写入 900，这样就会把用户修正系数变为 0.9。需要注意的是输入的正确重量的小数点位数，需要和用户分度值的一致。

目标重量的范围（不考虑小数点）是 800~用户满量程。

该功能只能通过一次写入这两个寄存器的 Modbus 指令来实现，写一个或者写大于两个都会忽略不执行。

1 6、恢复默认用户参数（224-225）

写入 44444444，可以初始化用户参数，把用户参数都设置为表中的默认值。如果用户参数设置乱了，不知道怎么恢复，那么就可以使用这个功能恢复为默认值，然后调整关注的参数即可。该功能只能通过一次写入这两个寄存器的 Modbus 指令来实现，写一个或者写大于两个都会忽略不执行。

七、标定操作保持寄存器（300 区）

1、寄存器列表

地址	PLC 地址	字数	功能	默认值	备注
300	40301	1	标定分度值	3	同时设置用户分度值
301	40302	1	备用	0	
302	40303	2	标定满量程	8000	同时设置用户满量程
304	40305	2	标定点重力加速度	0	
310	40311	2	标定零点功能		
312	40313	2	标定加载点 1 功能		
314	40315	2	标定加载点 2~5 功能		
320	40321	2	备份标定参数功能		
322	40323	2	恢复标定参数功能		
340	40341	2	开启/退出写保护状态		

2、标定分度值（300）

分度值决定了模块的分辨率，为标定分度值和用户分度值，标定分度值是在标定时候选择的分度值，用户分度值则是实际使用的分度值。用户分度值受标定分度值的限制，设置范围如下：标定分度值-3≤用户分度值≤标定分度值+3。设定标定分度值时，会把用户分度值设置为相同的值。

分度值参数设置范围是 0~23，分别表示不同的分度值，见下表。

参数	0	1	2	3	4	5	6	7
分度值	0.0001	0.0002	0.0005	0.001	0.002	0.005	0.01	0.02
参数	8	9	10	11	12	13	14	15
分度值	0.05	0.1	0.2	0.5	1	2	5	10
参数	16	17	18	19	20	21	22	23
分度值	20	50	100	200	500	1000	2000	5000

3、标定满量程（302-303）

满量程决定模块的最大称量，分为标定满量程和用户满量程，标定满量程是在标定的时候设定的满量程，用户满量程是实际使用的满量程。用户满量程受标定满量程的限制，设置范围如下：用户满量程≤标定满量程。设定标定满量程时，会把用户满量程设置为相同的值。

满量程是一个不带小数点的数值，如果分度值选用的是带小数点的，那么满量程使用对应的小数点位数。比如分度值设置为 3（0.001），它是 3 位小数点。那么满量程也是三位小数点，这个时候满量程如果为 30000，它代表的含义就是 30.000。

4、标定重力加速度（304-305）

使用地重力加速度如果和标定地重力加速度不同，会带来测量误差，通过输入用户地的重力加速度，自动的计算用户修正系数，从而修正这个误差。

重力加速度值，固定六位小数点，设置范围是 9300000~10300000（9.3~10.3）。如果要使用用户重力加速度修正功能。那么标定重力加速度，就必须设置在范围内。不使用这个功能的话，就设置为 0。

5、标定零点功能（310-311）

标定零点功能用于确定系统的标定零点，写入 5555555，即可执行该功能。

标定零点也是非线性修正的基准点，如果长时间使用，使系统零点发生比较大的偏移，技术通过开机置零和手动置零，零点没有误差。但是他会影响非线性修正的拐点，从而影响测量精度。这个时候可以重新执行一下标定零点功能，就可以消除这个误差。

6、标定加载点 1 功能（312-313）

模块支持最多五段，也就是 5 个点的非线性修正。其中加载点 1 是必须标定的，即至少使用一段标定。标定加载点 1 功能就是用来确定加载点 1 的，通过写入加载点 1 的值，来执行标定功能。

该功能只能通过一次写入这两个寄存器的 Modbus 指令来实现，写一个或者写大于两个都会忽略不执行。

7、标定加载点 2~5 功能（314-315）

标定加载点 2~5 功能就是用来确定加载点 2~5 的，通过写入加载点的值，来执行标定功能。每个加载点需依次标定，就是标定过加载点 1，然后在执行的就是标定加载点 2，然后才是标定加载点 3。如果要修改加载点 2，或者标定一个小于当前加载点的点，就必须重新标定加载点 1，重新开始。

各加载点之间至少间隔 10 个分度值。否则会执行错误。比如加载点 1 是 1000，加载点 2 是 1001 那么就会返回错误。

该功能只能通过一次写入这两个寄存器的 Modbus 指令来实现，写一个或者写大于两个都会忽略不执行。

8、备份标定参数（320-321）

备份标定参数，写入 1111111，即可执行该功能。它有两个用途，一是在需要的时候恢复，二是在当前四份标定参数都被破坏时，模块就会自动使用备份的标定参数，进一步提升标定参数存储的可靠性。

该功能只能通过一次写入这两个寄存器的 Modbus 指令来实现，写一个或者写大于两个都会忽略不执行。

9、恢复标定参数（322-323）

恢复标定参数，写入 2222222，即可执行该功能。恢复备份的标定参数后，当前标定参数将丢失，请谨慎使用该功能。

该功能只能通过一次写入这两个寄存器的 Modbus 指令来实现，写一个或者写大于两个都会忽略不执行。

10、开启/退出写保护状态（340-341）

开启/退出写保护状态，写入 13142012，模块进入写保护状态。写入其他值，模块退出写保护状态。当模块处于写保护状态时，100~400 各个区的写操作，不能执行。除非先进行区解锁操作。

八、标定参数保持寄存器（400 区）

1、寄存器列表

地址	PLC 地址	字数	功能	默认值	备注
400	40401	1	标定分度值	3	
401	40402	1	备用	0	
402	40403	2	标定满量程	8000	
404	40405	2	标定零点 AD 码		
406	40407	10	标定加载点 AD 码		
416	40417	10	标定加载点目标值		
426	40427	2	标定点重力加速度		

2、参数说明

本区为标定参数读写区，一般执行标定操作都应该通过 300 区实现，而不是直接写本区。直接写本区，不会修改相关的参数，例如修改本区的标定分度值，不会修改用户分度值。

为了提高参数可靠性，建议上位机读取并保存本区的参数，当发生参数丢失时，可写回。

九、标定参数保持寄存器（500 区）

1、寄存器列表

地址	PLC 地址	字数	功能	默认值	备注
500	40501	1	采样 1 的序列号		
501	40502	2	净重高速采样 1		
503	40504	2	净重高速采样 2		
.....					
531	40532	2	净重高速采样 16		

2、参数说明

对于嵌入式应用，采用高通讯波特率，用上一小节的方式，可以获取全部的有效采样。但对于大部分主机而言，通讯速度一般为每秒几十次，高于 40 次/秒的采样，想要获取全部有效采样，就要用到高速采样缓冲区（500 区）。

高速采样缓冲区里面保存了最新的 16 个净重采样值（地址 501~532），和第一个采样值的序列号（地址 500）。用户需要自动拼接。

比如第一次读取的序列号是 1，十六个采样值时 A1~A16；第二次读取的序列号是 7，采样值是 B1~B16。那么这个时候实际完成了 22 次采样，采样数据为 A1~A16，B11~B16。其中 B1~B10 和 A7~A16 是重复的丢弃。