

通讯部分

使用说明书

天津市丽景微电子有限公司

1 简介

1-1 启用 MODBUS 通讯协议	1
1-2 MODBUS 通讯协议类型：ASCII 或 RTU （二进制）	2
1-3 设置串口传输参数：波特率和信息格式	3
1-4 信息元素及格式介绍	3
1-4-1 ASCII 模式中帧的格式	4
1-4-2 RTU 模式中帧的格式	4
1-4-3 设备地址	5
1-4-4 功能代码	5
1-4-5 数据	6
1-4-6 校验位	6
1-4-7 ASCII 及 RTU 数据组成示例	6

2 MODBUS 功能

2-1 功能列表	8
2-2 传输字符串的内容	8
2-2-1 功能 1、3 和 4：读取开关量输出 / 存储 / 输入寄存器（01、03 和 04 Hex）	9
2-2-2 功能 5 和 6：预设开关量输出 / 单个寄存器 （05 和 06 Hex）	9
2-2-3 功能 15 和 16：预设多个开关量输出 / 寄 存器（0F 和 16 Hex）	10

3 校验位

3-1 奇偶校验	12
3-2 CRC 运算法则：循环冗余校验（RTU 模式） ..	12
3-2-1 CRC 创建步骤	12
3-2-2 CRC 在数据中的位置	13
3-2-3 在 C 语言中生成 CRC 的示例	13
3-3 LRC 运算法则：纵向冗长检测（ASCII 模式） ..	14
3-3-1 LRC 创建步骤	14
3-3-2 LRC 在数据中的位置	14

3-3-3 在 C 语言中创建 LRC 的示例	14
4 MODBUS 异常响应	
4-1 异常响应列表	16
5 数据区域	
5-1 输入寄存器数据区域	17
5-1-1 输入状态寄存器 (表 5.1.1)	19
5-1-2 输出状态寄存器 (表 5.1.2)	20
5-1-3 指令状态寄存器	21
5-2 存储寄存器数据区域	21
5-2-1 指令寄存器	29
5-2-1-1 Alibi 状态寄存器	32
5-2-2 通道 X 状态寄存器 (表 5.2.2)	32
5-2-3 菜单区域	33
5-3 开关量输出状态数据区域	38
6 标定指令	

1 简介

Modbus 通讯协议定义了信息的框架与结构，以及管理整个系统的“主”设备与响应主设备发送的指令的“子”设备(主.子设备技术) 之间的通讯模式，还定义了如何建立或中断主子设备之间的通讯、如何识别发送器与接收器、如何互换信息、如何检测错误。

只有主设备才可以发送指令(即：发问)， 其它的设备(子设备) 可以响应主设备发送的指令或执行相关的操作。主设备通过发送地址，可对某一子设备进行发问，或发送广播地址以对所有子设备进行发问。如果是对某一子设备进行发问，那么该子设备会对指令作出回复(即：响应) 且带有相关信息，但如果发送的是广播地址，那么子设备不会响应任何信息。

因此发问与响应之间有着以下两种不同的结构：

- 发问一台子设备(通过输入相应地址) /该子设备响应
- 发问所有子设备(通过广播地址) /无任何响应

1-1 启用 MODBUS 通讯协议

仪表开机上电自检时按 **TARE** 键进入仪表系统菜单。

- ◆ 按 **ZERO** 或 **TARE** 键以选择“SEtup”选项，然后按 **PRINT** 键确认。
- ◆ 按 **ZERO** 或 **TARE** 键以选择“SErIAL”选项，然后按 **PRINT** 键确认。
- ◆ 按 **ZERO** 或 **TARE** 键以选择“PC.SEL”选项，然后按 **PRINT** 键进入。

设置通讯参数

◆ 屏幕显示“PC.SEL”选项，然后按 **PRINT** 键进入，滚动至“485 或“232”参数，再按 **PRINT** 键确认。

◆ 屏幕显示“COM.Prn”选项，然后按 **ZERO** 或 **TARE** 键以选择“COM.PC”，然后按 **PRINT** 键进入。

◆ 屏幕显示“PCMode”选项，然后按 **PRINT** 键进入，滚动至“Modbus 参数，再按 **PRINT** 键确认。

◆ 屏幕显示“Mod.typ”选项，按 **PRINT** 键进入。

◆ 您可选择协议的类型：Ascii or Binary (rtu) (参见 1.2 节)，然后设置仪表的地址（参见 1.4.3 节）。

ASCII 或 RTU

--当进入“modbus”参数后，第一个显示的选项为“Mod.typ”，然后按 **PRINT** 键进入 i 选择 ASCII 或 RTU 协议类型，最后按 **PRINT** 键确认。

仪表地址

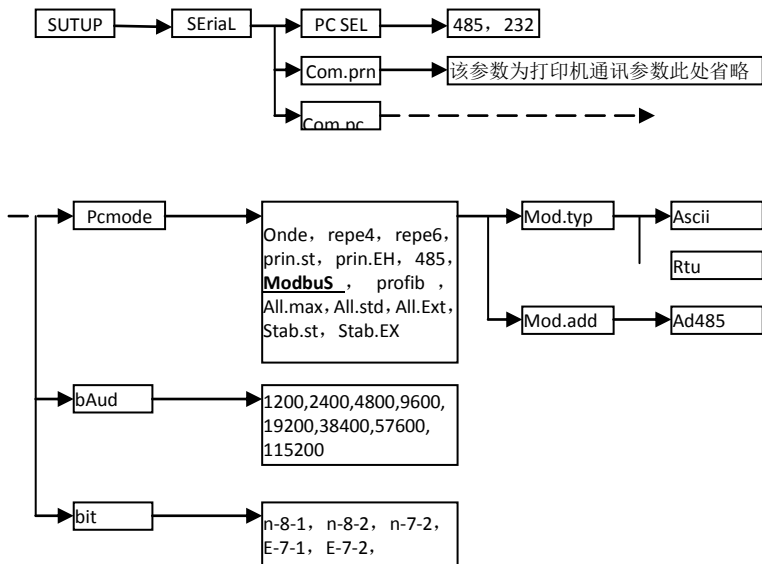
--第二个显示的选项为“mod.Add”，按 **PRINT** 键后，屏幕格显示“Ad485”信息，然后要求您输入仪表(或子设备)的地址，最后按 **PRINT** 键确认。

设置波特率及数据格式

---按照需要设置波特率和数据格式，在“bAud”及“bit”选项中设置波特率和串口字符格式。

设置完成后，按 C 键返回，直至屏幕显示“SAVE?”（保存）信息为止。然后按 PRINT 键保存所有的设置。

通讯参数功能树形图如下：



1-2 MODBUS 通讯协议类型：ASCII 或 RTU（二进制）

在启用 Modbus 通讯协议后，需选择传输协议的类型：ASCII 或 RTU（参见第 2 节），其定义了数据组中信息的打包方式及解码方式。

ASCII 模式（美国信息互换标准代码）

在一条信息中，发送的每个字节（8 位）作为 2 个 ASCII 字符。

该模式的主要特征是在信息传输期间（未发生错误），两个字符之间传输的时间间隔长达 1 秒。

RTU 模式（远程终端单元）

一条信息的每个字节（8 位）由 2 个十六进制字符组成，每个十六进制数包含 4 位。

该模式的主要特征是相对于 ASCII 模式而言，有较好的字符密度、优良的通讯能力和较大的存储容量。

1-3 设置串口传输参数：波特率和信息格式

在设置完 Modbus 传输类型 (ASCII 或 RTU) 后, 需设置串口的通讯参数, 即: 波特率和数据格式 (参见图 1)。

波特率 (即: 传输速度):

SEtuP ⇒ SEriAL ⇒ CoM.PC ⇒ bAud ⇒ 1200 ... 115200

数据格式 (即: 串口字符格式):

SEtuP ⇒ SEriAL ⇒ CoM.PC ⇒ bAud ⇒ n-8-1, n-8-2, n-7-2, E-7-1, E-7-2

※ 推荐格式:

- ASCII 模式: n-7-2, E-7-1
- RTU 模式: n-8-2, E-8-1

※ 其中:

- n 无奇偶校验 (none)
- E 偶校验

※ 例如: 如使用 n-8-2 数据格式, 那么:

- 8 数据位数
- n 无奇偶校验
- 2 停止位

※ 串口传输类型 (ASCII 或 RTU) 及串口通讯参数 (波特率和数据格式) 必须与连入 MODBUS 网络工作组的其它设备保持一致。

1-4 信息元素及格式介绍

在两种串口传输模式 (ASCII 或 RTU) 中, 发送设备发送一条 MODBUS 信息为一帧, 该帧有已知的开始和结束点。这允许接收设备定位信息的开始点、读取地址部分、确定响应设备 (如发送广播地址, 则发送至所有的子设备) 及明确何时信息发送结束, 如检测到不完整的信息, 那么系统将会报错。

主子设备的信息格式:

- ◆ 设备地址: 主设备通过地址, 与某台子设备之间建立通讯 (如地址为 0, 即: 广播地址, 那么数据将发送至所有的子设备上)。
- ◆ 功能代码: 定义子设备需要执行的请求动作。
- ◆ 数据: 必须传输的内容。
- ◆ 校验位: 由 CRC 或 LRC 运算法则计算得出。

为进一步了解以上数据格式功能，请参见下列章节。

以下为发问与响应说明：

发问：

功能代码告知已成功通讯的子设备需执行的动作。数据字节包含子设备需要的附加情报以执行功能。校验位提供子设备一个方式以确认信息内容的完整性。

响应：

- ◆ 如子设备正常响应：
 - 功能代码与发问时的功能代码一致。数据字节包含检索来自子设备的数据，例如：寄存器的值或状态。
- ◆ 如子设备检测到错误（格式、CRC 或 LRC 校验位）或无法执行请求的动作：
 - 主设备发送的信息将视为无效且拒绝，随之也无法执行请求的动作；如响应中的功能代码发生变化，则表示错误响应，且数据字节中也会包含错误代码。

1-4-1 ASCII 模式中帧的格式

在 ASCII 模式中，数据以 (:) 字符 (ASCII 3A Hex) 开头，以 2 个字符的 CRLF (ASCII 0D 和 0A Hex) 结尾。

对于剩余的其它字段都可发送由 0...9 及 A...F 组合成的十六进制字符；发送设备将一直不断地监控着网络工作组以定位 (:) 字符，当发送的任何一个数据被接收时，每台接收设备解译转栏（字段地址）以识别是否该设备已被定址。

在字符与字符传输之间有多大 1 秒的时间间隔，如果时间间隔太长，接收器就假定错误已经发生。

以下为一组典型的帧：

起始	地址	功能	数据	LRC 校验	终止
1 个字符 :	2 个字符	2 个字符	N 个字符	2 个字符	2 个字符 CRLF

图 3：ASCII 数据帧

1-4-2 RTU 模式中帧的格式

在 RTU 模式中，信息开始于静态时间间隔，至少持续 3.5 倍的字符时间 (T1-T2-T3-T4 时间间隔，见图 3)。

设备将连续不断地对传输信号进行监控，即使静态时间间隔内。当收到第一个字段（地址）时，每个设备对其进行解码以验证是否信息来源于目标设备。

每个字段发送的字符都是以十进制表示，从 0-255。

在传输完所有字符后，将会有一段静态时间间隔，至少持续 3.5 倍的字符时间，以指示信息发送结束；在这之后，可以传输新的信息。

整个一帧，必须作为一个连续的信息流传输。如果在信息传输结束前，出现一段大于 1.5 倍的字符时间的静态时间间隔，那么接收设备以这段不完整的信息作为终止信号，且视接下去收到的字节为新一轮的信息。

同样的，如果在静态时间间隔少于 3.5 倍的字符时间前开始传输一条新信息，且该信息后跟着上一条信息，那么接收设备视其为上一条信息的连续，这将引起一个错误，由于两个信息的重合，使得 CRC 校验将无效。

以下为帧的框架图：

开始	地址	功能	数据	LRC 校验	终止
T1-T2- T3-T4:	8 位	8 位	N * 8 位	16 位	T1-T2- T3-T4

图 3：RTU 数据帧

1-4-3 设备地址

如前面章节提及，Modbus 协议总是在同一时间涉及到管理总线的主设备及子设备（除广播地址外）。为正确定位接收者，主设备发送的数字地址（一个字节：RTU 为 8 位）将作为帧的第一个字段。因此每个子设备需要输入一个不同的数字地址以使主设备正确定位。当子设备作出响应，在响应的地址字段中会出现其对应的数字地址，以便主设备知道是哪台子设备作出了响应。

子设备有效的数字地址，从 0 至 99，其中：

99 表示广播地址（发送至所有的子设备）

0 最小子设备地址

98 最大子设备地址

1-4-4 功能代码

功能代码的字段包含 2 个字符（ASCII）或 8 位（RTU）。有效代码值：从 1 至 255（十进制）。

当一条信息从主设备发送至子设备时，功能代码字段表示子设备需要执行的动作类型（例如：输入寄存器的读取等等）。

当子设备作出响应，从响应的功能代码中可确定是否响应正常（无错误），或已经发生某种错误（即：异常响应）。对于正常响应而言，子设备只需响应功能的源代码即可，如果是异常响应，在响应后，功能源代码的最有效位将设置成 1 个逻辑值。

当发生异常响应时，除了功能源代码会改变之外，在响应的数据字段中也会有错误代码出现以告知主设备错误的类型或异常的原因。

1-4-5 数据

数据字段由 2 个十六进制字符组成，范围从 00 至 FF Hex。根据网络工作组的串口传输模式，由 ASCII 字符或 RTU 字符来执行相关动作。

从主设备发送至子设备的数据字段中，包含子设备必须使用的附加情报以执行由功能代码定义的动作。

1-4-6 校验位

校验位字段的内容取决于 Modbus 通讯协议的传输模式 (ASCII 或 RTU)，因为它们两种使用的是不同的校验计算方式。

ASCII 模式

传输的字符串由 LRC（纵向冗长检测）运算法则进行校验，参见 3.3 节。

该校验位包含 2 个 ASCII 字符，它们是 LRC 运算法则通过对数据内容的计算的结果，不包括起始（:）

字符和 CRLF 终止符。

RTU 模式

传输的字符串由 CRC（循环冗余校验）运算法则进行校验，参见 3.2 节。

该校验位包含 16 位（2 个 8 位的字节），它们是 CRC 运算法则对信息内容计算的结果。

该字段是帧的最后部分。

1-4-7 ASCII 及 RTU 数据组成示例

以下图表为 Modbus 信息传输示例（发问及正常响应），这两个图标的字段内容都是以十六进制显示的，数据是以 ASCII 或 RTU 模式进行编译。

发问：“输入寄存器的读取”，子设备地址为 01，读取 3 个寄存器的内容，从

寄存器第八个位置开始读取。

字段名称	示例 (Hex)	ASCII : 字节	RTU : 8 位字段
起始		: (冒号)	无
子设备地址	01	0 1	0000 0001
功能	04	0 4	0000 0100
起始地址 (高位)	00	0 0	0000 0000
起始地址 (低位)	08	0 0	0000 1000
寄存器个数 (高位)	00	0 0	0000 0000
寄存器个数 (低位)	03	0 3	0000 0011
校验位		LRC (2 个字节)	CRC (16 位)
终止符		CR LF	无
总的字节个数		17	8

响应：

字段名称	示例 (Hex)	ASCII : 字节	RTU : 8 位字段
起始		: (冒号)	无
子设备地址	01	0 1	0000 0001
功能	04	0 4	0000 0100
字节个数	06	0 6	0000 0110
数据 (高位)	02	0 2	0000 0010
数据 (低位)	2B	2 B	0010 1011
数据 (高位)	00	0 0	0000 0000
数据 (低位)	00	0 0	0000 0000
数据 (高位)	00	0 0	0000 0000
数据 (低位)	63	6 3	0110 0011
校验位		LRC (2 个字节)	CRC (16 位)
终止符		CR LF	无
总的字节个数		23	11

在子设备响应过程中，如响应的功能代码与传输的一致，则表明正常响应。

“字节个数”字段定义多少组 8 位数据将被反馈，即：表示 ASCII 或 RTU “数据”字段的字节个数：在 ASCII 模式中，该值是数据字段中总的 ASCII 字节的一半（每个 4 位十六进制值组成一个 ASCII 字符，因此 2 个 ASCII 字符包含 8 位数据）。

例如：传输 63 Hex，RTU 表示为 8 位的字节 (01100011)，相同的值在 ASCII 表示为 2 个字节：ASCII 6 (0110110) 和 3 (0110011)。

2 MODBUS 功能

接下去的每个章节将为您详细介绍 Modbus 的每个功能，且所有的功能都是由一次发问（主机→仪表）与一次响应（仪表→主机）构成。

※

- 在 ASCII 传输模式中：每个字符等于一个 ASCII 码字符（8 位）。
- 在 RTU 传输模式中：每个字符是以十六进制为单位的字符（4 位）。
- 如某个数字前带有 0x 或 Hex，说明该数值是以十六进制为单位。

2-1 功能列表

以下为 DGT 仪表的 Modbus 可用功能。

表 2.1：Modbus 可用功能

功能代码	说明
01 (0x01)	读取开关量输出
03 (0x03)	读取存储寄存器
04 (0x04)	读取输入寄存器
05 (0x02)	预设单个开关量输出
06 (0x06)	预设单个寄存器
15 (0x0F)	预设多个开关量输出
16 (0x10)	预设多个寄存器

括弧中的值为十六进制值。

2-2 传输字符串的内容

接下去的章节将会为您介绍 Modbus 的可用功能（参见表 2.1），您可按照以下的信息字段进行发送：

- ◆ 地址 1 个字节，为仪表（子设备）地址。
- ◆ 功能 将要执行的代码或功能编号。
- ◆ 字节数 表示组成数据的字节数。
- ◆ 校验位（CRC/LRC） RTU 或 ASCII 传输模式时的校验位，总是为 2 个字节。

※ 以下章节说明的功能为：

- “开关量输出”：由主设备写入→由仪表或主设备读取

- “存储寄存器”：由主设备写入→由仪表或主设备读取
- “输入寄存器”：由仪表写入→由主设备读取
- ◆ 关于寄存器的详细说明，请参见第 5 节。
- ◆ 开关缓存器由 100 个字节组成，因此无法读取一个大于发送缓存器容量的寄存器数字编号，同时也无法写入一个大于接收缓存器容量的寄存器数字编号。

2-2-1 功能 1、3 和 4：读取开关量输出 / 存储 / 输入寄存器（01、03 和 04 Hex）

通过该功能可读取子设备（仪表）寄存器中的内容（由仪表或主设备写入），不支持广播地址通讯。

发问：

您可指定一个寄存器 / 开关量区域以读取（功能）相关数据，读取范围从初始寄存器（第一个寄存器地址）到需要读取的寄存器个数为止。寄存器地址从 0 开始，因此 1 至 16 对应着寄存器地址的 0 至 15。

地址	功能	第一个寄存器地址 高 低	寄存器个数 高 低	校验位
A	XX	00 08	00 01	CRC/LRC

响应：

响应数据内容由 2 个字节组成，以二进制表示且在每个字节中都向右对齐。对于每个寄存器而言，第一个字节包含最高有效位，第二个字节包含最低有效位。

地址	功能	第一个寄存器地址 高 低	寄存器个数	校验位
A	XX	02	00 0A	CRC/LRC

例如：A=01

- 在发问中：第一个寄存器地址 =00 08；寄存器个数 =00 01
- 在响应中，第一个寄存器地址 =00 0A

※ 最大可读取的寄存器个数为 49 个。

2-2-2 功能 5 和 6：预设开关量输出 / 单个寄存器（05 和 06 Hex）

该功能可设置一个单一的寄存器（通过仪表或子设备读取）至某个特定的值上。而且通过广播地址，您可在所有已连接的子设备中设置同一个寄存器。

发问：

您可指定一个寄存器 / 开关量输出区域以写入（功能）寄存器地址及相关值（寄存器值）。寄存器地址从 0 开始，因此 1 至 16 对应着寄存器地址的 0 至 15。

地址	功能	第一个寄存器地址 高 低	寄存器个数 高 低	校验位
A	XX	00 01	00 03	CRC/LRC

响应：

作为发问的响应。

地址	功能	第一个寄存器地址 高 低	寄存器个数 高 低	校验位
A	XX	00 01	00 03	CRC/LRC

例如：A=01

- 在发问中：寄存器地址 =00 01；寄存器值 =00 03
- 在响应中，寄存器地址 =00 01；寄存器值 =00 03

2-2-3 功能 15 和 16：预设多个开关量输出 / 寄存器（0F 和 16 Hex）

该功能可设置多个寄存器（通过仪表或子设备读取）至某个特定的值上。

发问：

您可指定一个寄存器 / 开关量输出区域以写入（功能），这里，必须设置的第一个寄存器地址是被指定的，从第一个开始的将要被写入的寄存器个数、传输寄存器值的字节个数（每个寄存器为 2 个字节）或字节个数，这些值都将分配至从第一个已定址的寄存器开始的寄存器中（第一个 2 个字节的寄存器值，第二个寄存器等）。

地址	功能	第一个寄存器地址 高 低	寄存器个数 高 低	字节 个数	第一个寄存器值 高 低	第二个寄存器值 高 低	校验位
A	XX	00 00	00 02	04	00 00	00 00	CRC/LRC

响应：

响应包括已修改的寄存器的识别（第一个寄存器地址和寄存器个数）。

地址	功能	第一个寄存器地址 高 低	寄存器个数 高 低	校验位
A	XX	00 00	00 02	CRC/LRC

例如：A=01

- 在发问中： 第一个寄存器地址 =00 00；寄存器个数 =00 02；字节个数 =04
第一个寄存器值 =00 00；第二个寄存器值 =00 00
- 在响应中， 第一个寄存器地址 =00 00；寄存器个数 =00 02

※ 最大可写入的寄存器个数：

RTU 模式：45 个

ASCII 模式：20 个

3 校验位

Modbus 通讯协议使用两种不同的校验类型：

- ◆ 字符或奇偶（偶或非偶）校验：可随意校验每个字符（参见 1.3 节）。
- ◆ 帧校验（LRC 或 CRC 运算法则）：可校验整个数据。

在 RTU 模式中，通讯字符串通过 CRC 进行校验；在 ASCII 模式中，通讯字符串通过 LRC 进行校验。

以上这两种校验都是由主设备建立的，且在传输数据之前，对数据的内容进行校验。在接收数据期间，子设备对每个字符及整个帧进行校验。

3-1 奇偶校验

以下为两种配置奇偶校验的方式（参见 1.3 节）：

N 无奇偶校验（none）

E 偶校验

其确定了每个字符的校验方式。

3-2 CRC 运算法则：循环冗余校验（RTU 模式）

在 RTU 传输模式中，传输信息中包括了 CRC 校验位，其功能旨在对整个信息的内容进行校验，且适用于任何一种用于校验字符的奇偶校验。CRC 字段由 2 个字节（包含 16 位的二进制值）组成，由发送设备进行计算且放置在数据的最后一个字节中。在接收信息期间，接收设备再次计算 CRC，然后与接收到的 CRC 进行比较，如果这两个值不同，那么系统将报错。

3-2-1 CRC 创建步骤

1. 加载一个 FFFF Hexz(都为 1)的 16 位寄存器，该寄存器称之为 CRC 寄存器。
2. OR- 排除数据的第一个字节及 16 位 CRC 寄存器的最低有效位。
在 CRC 寄存器中输入结果。
3. CRC 寄存器转换 1 位至右边（朝 LSB），在 MSB 的地方输入 0。提取 LSB 并进行校验。
4. 如 LSB=0，那么需重复步骤 3（其它的进入下一步）。
如 LSB=1，那么 OR 执行在 CRC 寄存器及 A001 Hex (1010 0000 0000 0001) 多项式之间。
5. 重复步骤 3 和 4 直至完成 8 位转换，在这之后，8 位一字节将被处理完成。

6. 重复步骤 2 和 5 以完成数据中的下一个 8 位，直到处理完所有的字节。
7. CRC 寄存器最后的内容是 CRC 值。
8. 当 CRC 被输入在数据中时，其字节（高和低）必须被互换，如以下章节说明。

3-2-2 CRC 在数据中的位置

当 16 位 CRC(2 个字节) 被发送时，首先发送的是次有效字节，后跟最有效字节。

例如：如 CRC 值为 1241 Hex (0001 0010 0100 0001)：

地址	功能	数据总数	数据	数据	数据	数据	CRC(低)	CRC(高)
							41	12

图 5：CRC 字节

3-2-3 在 C 语言中生成 CRC 的示例

以下是在 C 语言中创建 CRC 的示例。

※ 该功能可内部创建 CRC 高低位的交换。通过该功能，反馈的 CRC 值中的字节已经被交互。

该功能使用两种论证方式：

`unsigned char *puchMsg;` → 指针指向包含用于创建 CRC 的二进制数据的信息缓存器

`unsigned short usDataLen;` → 在信息缓存器中的字节数

该功能反馈 CRC 值作为一个无符号整型数。

CRC 生成功能

```

unsigned short CRC16(puchMsg, usDataLen)
unsigned char *puchMsg;           //message to calculate CRC upon
unsigned short usDataLen;         //quantity of bytes in message
{
    unsigned char uchCRCHi = 0xFF; //high CRC byte initialized
    unsigned char uchCRCLo = 0xFF; //low CRC byte initialized
    unsigned uIndex;               //will index into CRC lookup table

    while (usDataLen--)             //pass through message buffer
    {
        uIndex = uchCRCHi ^ *puchMsg++; //calculate the CRC
        uchCRCHi = uchCRCLo ^ uchCRCHi[uIndex];
        uchCRCLo = uchCRCLo[uIndex];
    }
    return (uchCRCHi << 8 | uchCRCLo);
}

```


3-3 LRC 运算法则：纵向冗长检测（ASCII 模式）

在 ASCII 传输模式中，传输数据中包括了 LRC 校验位，其功能为对整个数据的内容进行校验（除起始字符（:或冒号）及 CRLF 字符外）。该运算法则适用于任何一种用于校验字符的奇偶校验。

LRC 字段由 1 个字节（包含 8 位的二进制值）组成，由发送设备进行计算且放置在数据的最后一个字段中。在接收数据期间，接收设备再次计算 LRC，然后与接收到的 LRC 进行比较，如果这两个值不同，那么系统将报错。

计算 LRC 的过程：先总和数据中所有的字节（8 位），然后排出已执行的值，再形成补数 2 的结果，计算涉及到所有 ASCII 数据字符中的内容，除起始字符（:）及 CRLF 字符外。

3-3-1 LRC 创建步骤

1. 在 8 位字段范围内，总和数据中所有的字节，除起始字符（:）及 CRLF 字符外。这样将扣除执行值。
2. 从步骤 1 中减去结果值加到 FF Hex（8 位，都为 1），这样补数到 1。
3. 加 1 以补数到 2。

3-3-2 LRC 在数据中的位置

当 8 位 LRC（2 个 ASCII 字符）被发送时，首先发送的是最低有效位，后跟最高有效位。

例如：如 LRC 值为 61 Hex (0110 0001)：

冒号 (:)	地址	功能	数据 总数	数据	数据	数据	数据	CRC (低)	CRC (高)	CR	LF
								6	1		

图 6：LRC 字节

3-3-3 在 C 语言中创建 LRC 的示例

以下是在 C 语言中创建 CRC 的示例。该功能使用两种论证方式：

`unsigned char *auchMsg;` → 指针指向包含将要用于创建 CRC 的二进制数据的缓存器

unsigned short **usDataLen**; → 缓存器的字节数量

该功能反馈 LRC 值作为一个无符号的字符。

LRC 生成功能

```
static unsigned char LRC(auchMsg, usDataLen)
unsigned char *auchMsg;    //message to calculate
unsigned short usDataLen;  //LRC upon quantity of bytes in message
{
    unsigned char uchLRC = 0; //LRC char initialized
    while (usDataLen--)      //pass through message
        uchLRC += *auchMsg++; //buffer add buffer byte without carry

    return((unsigned char)(-((char_uchLRC)))); //return twos complement
}
```

4 MODBUS 异常响应

表 4.1：可能发生的 Modbus 异常响应情况

如正常响应，子设备响应发问的功能代码，将其放置在响应功能字段中，所有的功能代码都有其最有效位（MSB），设置为 0（值小于 80 Hex）。

如响应异常，子设备设置功能代码的 MSB 为 1（相当于总计值 80 Hex 至正常响应代码中）以发送报错信号，且异常代码会反馈在数据字段中，其表示异常响应类型。

4-1 异常响应列表

表 4.1：可能发生的 Modbus 异常响应情况

代码	异常情况	说明
01	不合法的功能	主设备接收到的功能代码不支持或无效。
02	不合法的数据地址	子设备接收到从主设备发送的数据地址不支持或无效。
03	不合法的数据值	子设备无法接收主设备数据字段的值或该值无效。
06	子设备忙碌中	子设备正在接收或处理其它的指令。当处理完之后，主设备方可再次发送。

5 数据区域

一共有 3 个数据区域，“输入”、“存储”及“开关量”，“输入”区域用于读取，“存储”及“开关量输出”用于写入。

前两个区域被分配在 Modbus 协议功能执行的寄存器中。

所有的数字值以高位优先的格式排序（第一个字节是最高有效位），用于数据输入区域及数据输出区域，同时所有的数字值以低位优先的格式排序（第一个字节是最低有效位），用于启动菜单数据。

5-1 输入寄存器数据区域

输入数据区域由主设备读取（由仪表写入），由 2 个字节的寄存器（输入寄存器）组成。

表 5.1：Modbus 输入寄存器

功能代码	说明
30001 (0)	毛重值
	毛重值
30002 (1)	毛重值
	毛重值
30003 (2)	净重值
	净重值
30004 (3)	净重值
	净重值

— 毛重值与净重值显示的格式：绝对值（不带小数点）

例如：如仪表设置 3 位小数点，3.000=3000

如仪表设置 2 位小数点，3.00=300

30005 (4)	输入状态寄存器
	输入状态寄存器
30006 (5)	指令状态寄存器
	指令状态寄存器
30007 (6)	输出状态寄存器
	输出状态寄存器

— 输入状态寄存器值的格式，参见 5.1.1 节。

— 指令状态寄存器值的格式，参见 5.1.3 节。

— 输出状态寄存器值的格式，参见 5.1.2 节。

功能代码	说明
30008 (7)	最后次读取或写入页的编号 (MSB)
	最后次读取或写入页的编号 (LSB)
30009 (8)	第一个菜单页的字
	第一个菜单页的字

30016 (15)	第八个菜单页的字
	第八个菜单页的字
30101 (100)	软件版本 (“00”) (字节3)
	软件版本 (发布版本) (字节2)
30102 (101)	软件版本 (下属版本) (字节1)
	软件版本 (修正版本) (字节0)

— 软件版本的格式 (寄存器 100~101) :

绝对值 (无小数点)

例如: 软件版本 07.00.00=070000

30103 (102)	通道 1 的 ADC 值 (字节3)
	通道 1 的 ADC 值 (字节2)
30104 (103)	通道 1 的 ADC 值 (字节1)
	通道 1 的 ADC 值 (字节0)
30105 (104)	通道 2 的 ADC 值 (字节3)
	通道 2 的 ADC 值 (字节2)
30106 (105)	通道 2 的 ADC 值 (字节1)
	通道 2 的 ADC 值 (字节0)
30107 (106)	通道 3 的 ADC 值 (字节3)
	通道 3 的 ADC 值 (字节2)
30108 (107)	通道 3 的 ADC 值 (字节1)
	通道 3 的 ADC 值 (字节0)
30109 (108)	通道 4 的 ADC 值 (字节3)
	通道 4 的 ADC 值 (字节2)
30110 (109)	通道 4 的 ADC 值 (字节1)
	通道 4 的 ADC 值 (字节0)
30111 (110)	通道 1 的微伏数 (字节1)
	通道 1 的微伏数 (字节0)

功能代码	说明
30112 (111)	通道 2 的微伏数 (字节 1)
	通道 2 的微伏数 (字节 0)
30113 (112)	通道 3 的微伏数 (字节 1)
	通道 3 的微伏数 (字节 0)
30114 (113)	通道 4 的微伏数 (字节 1)
	通道 4 的微伏数 (字节 0)
30115 (114)	模拟输出值 (字节 1)
	模拟输出值 (字节 0)

– 可用的 ADC 点与微伏数 (寄存器 102~114) :

在非独立通道和传输模式中,所有的通道共享一个 ADC 值与微伏数。

在独立通道模式中,一个 ADC 值与微伏数对应一个通道,其它的显示为零。

如果要读取的通道还未配置过,那么对应的寄存器值为零。

5-1-1 输入状态寄存器 (表 5.1.1)

位	说明	位的含义	
		0	1
(LSB)			
0	净重值的正负状态	+	-
1	毛重值的正负状态	+	-
2	重量稳定性	NO	YES
3	重量欠载状态	NO	YES
4	重量过载状态	NO	YES
5	已去皮	NO	YES
6	已手动输入皮重	NO	YES
7	毛重在零值附近	不在零值附近	在零值附近
(MSB)			
8	仪表输入 1	禁用	启用
9	仪表输入 2	禁用	启用
10	不使用		
11	不使用		
12	不使用		

位	说明	位的含义	
		0	1
13	不使用		
14	显示的通道（低位） ⁽¹⁾		
15	显示的通道（高位）（从 0 至 3） ⁽¹⁾		

⁽¹⁾：高位，低位： 0 0 → 通道 1 0 1 → 通道 2

 (15) (14) 1 0 → 通道 3 1 1 → 通道 4

5-1-2 输出状态寄存器（表 5.1.2）

其在输入寄存器中的编号为 6，以下为两个字节的含义：

位	说明	位的含义	
		0	1
(LSB)			
0	开关量输出 1	未激活	激活
1	开关量输出 2	未激活	激活
2	开关量输出 3	未激活	激活
3	开关量输出 4	未激活	激活
4	开关量输出 5	未激活	激活
5	开关量输出 6	未激活	激活
6	不使用		
7	不使用		
(MSB)			
8	不使用		
9	不使用		
10	不使用		
11	不使用		
12	不使用		
13	不使用		
14	不使用		
15	不使用		

5-1-3 指令状态寄存器

其在输入寄存器中的编号为 5，以下为两个字节的含义：

- | | |
|----------|------------------------|
| 高字节 | → 最后次接收到的指令（参见表 5.2.1） |
| 低字节：低半字节 | → 已执行的指令的计数 |
| 高半字节 | → 最后次接收到的指令的结果 |

其中：最后次接收到的指令的结果可以有以下值：

- | | |
|--------------------------------|----------|
| ◆ OK=0 | 正确且已执行完毕 |
| ◆ ExceptionCommandWrong =1 | 不正确的指令 |
| ◆ ExceptionCommandData =2 | 不正确的指令数据 |
| ◆ ExceptionCommandNotAllowed=3 | 无效指令 |
| ◆ ExceptionNoCommand=4 | 不存在的指令 |

5-2 存储寄存器数据区域

存储数据区域由主设备写入（由仪表读取），由 2 个字节的寄存器（存储寄存器）组成。

表 5.2：Modbus 存储寄存器

寄存器编号	说明
40001 (0)	指令寄存器 (MSB)
	指令寄存器 (LSB)
40002 (1)	参数 1 (字节 3)
	参数 1 (字节 2)
40003 (2)	参数 1 (字节 1)
	参数 1 (字节 0)
40004 (3)	参数 2 (字节 3)
	参数 2 (字节 2)
40005 (4)	参数 2 (字节 1)
	参数 2 (字节 0)
40006 (5)	不使用
	不使用

寄存器编号	说明
40007 (6)	不使用
	不使用
40008 (7)	不使用
	不使用

– 指令寄存器值的格式：参见 5.2.1 节

40009 (8)	第一个菜单页的字
	第一个菜单页的字

40016 (15)	第八个菜单页的字
	第八个菜单页的字
40101 (100)	毛重值 (字节 3)
	毛重值 (字节 2)
40102 (101)	毛重值 (字节 1)
	毛重值 (字节 0)
40103 (102)	净重值 (字节 3)
	净重值 (字节 2)
40104 (103)	净重值 (字节 1)
	净重值 (字节 0)
40105 (104)	皮重值 (字节 3)
	皮重值 (字节 2)
40106 (105)	皮重值 (字节 1)
	皮重值 (字节 0)

– 毛重值、净重值与皮重值显示的格式：

绝对值（不带小数点）

例如： 如仪表设置 3 位小数点，3.000=3000

如仪表设置 2 位小数点，3.00=300

40107 (106)	输入状态寄存器
	输入状态寄存器
40108 (107)	输出状态寄存器
	输出状态寄存器

– 输入状态寄存器值的格式，参见 5.1.1 节

– 输出状态寄存器值的格式，参见 5.1.2 节

寄存器编号	说明	
DGT 系列开关量输出限值设置		
40109 (108)	输出 1 启用限值 (临时)	(字节 3)
	输出 1 启用限值 (临时)	(字节 2)
40110 (109)	输出 1 启用限值 (临时)	(字节 1)
	输出 1 启用限值 (临时)	(字节 0)

40119 (118)	输出 6 启用限值 (临时)	(字节 3)
	输出 6 启用限值 (临时)	(字节 2)
40120 (119)	输出 6 启用限值 (临时)	(字节 1)
	输出 6 启用限值 (临时)	(字节 0)
40121 (120)	输出 1 禁用限值 (临时)	(字节 3)
	输出 1 禁用限值 (临时)	(字节 2)
40122 (121)	输出 1 禁用限值 (临时)	(字节 1)
	输出 1 禁用限值 (临时)	(字节 0)

40131 (130)	输出 6 禁用限值 (永久)	(字节 3)
	输出 6 禁用限值 (永久)	(字节 2)
40132 (131)	输出 6 禁用限值 (永久)	(字节 1)
	输出 6 禁用限值 (永久)	(字节 0)
40133 (132)	输出 1 启用限值 (永久)	(字节 3)
	输出 1 启用限值 (永久)	(字节 2)
40134 (133)	输出 1 启用限值 (永久)	(字节 1)
	输出 1 启用限值 (永久)	(字节 0)

40143 (142)	输出 6 启用限值 (永久)	(字节 3)
	输出 6 启用限值 (永久)	(字节 2)
40144 (143)	输出 6 启用限值 (永久)	(字节 1)
	输出 6 启用限值 (永久)	(字节 0)
40145 (144)	输出 1 禁用限值 (永久)	(字节 3)
	输出 1 禁用限值 (永久)	(字节 2)
40146 (145)	输出 1 禁用限值 (永久)	(字节 1)
	输出 1 禁用限值 (永久)	(字节 0)

寄存器编号	说明

40155 (154)	输出 6 禁用限值 (永久) (字节 3)
	输出 6 禁用限值 (永久) (字节 2)
40156 (155)	输出 6 禁用限值 (永久) (字节 1)
	输出 6 禁用限值 (永久) (字节 0)

※ 不执行限值控制：

- 如值大于等于最大称量值
- 如禁用值大于等于启用值

最低有效位值减少至秤盘的最小分度值

称重模式设为传输模式

40201 (200)	通道数量 (字节 1)
	通道数量 (字节 0)
40202 (201)	通道 1 的状态寄存器 (MSB)
	通道 1 的状态寄存器 (LSB)
40203 (202)	通道 1 的重量值 (字节 3)
	通道 1 的重量值 (字节 2)
40204 (203)	通道 1 的重量值 (字节 1)
	通道 1 的重量值 (字节 0)

40211 (210)	通道 4 的状态寄存器 (MSB)
	通道 4 的状态寄存器 (LSB)
40212 (211)	通道 4 的重量值 (字节 3)
	通道 4 的重量值 (字节 2)
40213 (212)	通道 4 的重量值 (字节 1)
	通道 4 的重量值 (字节 0)
40214 (213)	通道 1 的净重值 (字节 3,2)
40215 (214)	通道 1 的净重值 (字节 1,0)
40216 (215)	通道 2 的净重值 (字节 3,2)
40217 (216)	通道 2 的净重值 (字节 1,0)
40218 (217)	通道 3 的净重值 (字节 3,2)
40219 (218)	通道 3 的净重值 (字节 1,0)

寄存器编号	说明
40220 (219)	通道 4 的净重值 (字节 3,2)
40221 (220)	通道 4 的净重值 (字节 1.0)
40222 (221)	通道 1 的皮重值 (字节 3,2)
40223 (222)	通道 1 的皮重值 (字节 1.0)
40224 (223)	通道 2 的皮重值 (字节 3,2)
40225 (224)	通道 2 的皮重值 (字节 1.0)
40226 (225)	通道 3 的皮重值 (字节 3,2)
40227 (226)	通道 3 的皮重值 (字节 1.0)
40228 (227)	通道 4 的皮重值 (字节 3,2)
40229 (228)	通道 4 的皮重值 (字节 1.0)

– 通道 X 状态寄存器值的格式：参见 5.2.2 节

重量值显示的格式：绝对值（不带小数点）

例如：如仪表设置 3 位小数点，3.000=3000

如仪表设置 2 位小数点，3.00=300

指令	
40231 (230)	指令状态寄存器 (MSB)
	指令状态寄存器 (LSB)
40232 (231)	指令寄存器 (MSB)
	指令寄存器 (LSB)
40233 (232)	指令参数
	指令参数

40236 (235)	指令参数
	指令参数

– 指令状态寄存器值的格式：参见 5.1.3 节

– 指令寄存器值的格式：参见 5.2.1 节

Alibi 内存	
40251 (250)	最后次储存的毛重值 (字节 3)
	最后次储存的毛重值 (字节 2)
40252 (251)	最后次储存的毛重值 (字节 1)
	最后次储存的毛重值 (字节 0)

寄存器编号	说明
40253 (252)	最后次储存的皮重值 (字节 3)
	最后次储存的皮重值 (字节 2)
40254 (253)	最后次储存的皮重值 (字节 1)
	最后次储存的皮重值 (字节 0)
40255 (254)	最后次称重 ID 值 (字节 3)
	最后次称重 ID 值 (字节 2)
40256 (255)	最后次称重 ID 值 (字节 1)
	最后次称重 ID 值 (字节 0)
40257 (256)	Alibi 状态寄存器 (MSB)
	Alibi 状态寄存器 (LSB)

- Alibi 状态寄存器值的格式：参见 5.2.1.1 节

菜单	
40301 (300)	第一个菜单的字 (页 0)
	第一个菜单的字 (页 0)

40308 (307)	第八个菜单的字 (页 0)
	第八个菜单的字 (页 0)
40309 (308)	第九个菜单的字 (页 1)
	第九个菜单的字 (页 1)

40812 (811)	最后个菜单的字 (页 64)
	最后个菜单的字 (页 64)

- 字符值的格式：参见 5.2.3 节

以一个字表示的重量值 (最低有效位)	
41101 (1100)	毛重值 (字节 1)
	毛重值 (字节 0)
41102 (1101)	净重值 (字节 1)
	净重值 (字节 0)
41103 (1102)	皮重值 (字节 1)
	皮重值 (字节 0)

寄存器编号	说明
-------	----

- 毛重值、净重值与皮重值显示的格式：

绝对值（不带小数点）

例如：如仪表设置 3 位小数点，3.000=3000

如仪表设置 2 位小数点，3.00=300

41104 (1103)	输入状态寄存器	(MSB)
	输入状态寄存器	(LSB)
41105 (1104)	输出状态寄存器	(MSB)
	输出状态寄存器	(LSB)

- 输入状态寄存器值的格式，参见 5.1.1 节。
- 输出状态寄存器值的格式，参见 5.1.2 节。

DGT 系列开关量输出限值设置（以一个字节表示，最低有效位）

41106 (1105)	输出 1 启用限值 (临时)	(字节 1)
	输出 1 启用限值 (临时)	(字节 0)

41111 (1110)	输出 6 启用限值 (临时)	(字节 1)
	输出 6 启用限值 (临时)	(字节 0)
41112 (1111)	输出 1 禁用限值 (临时)	(字节 1)
	输出 1 禁用限值 (临时)	(字节 0)

41117 (1116)	输出 6 禁用限值 (临时)	(字节 1)
	输出 6 禁用限值 (临时)	(字节 0)
41118 (1117)	输出 1 启用限值 (永久)	(字节 1)
	输出 1 启用限值 (永久)	(字节 0)

41123 (1122)	输出 6 启用限值 (永久)	(字节 1)
	输出 6 启用限值 (永久)	(字节 0)
41124 (1123)	输出 1 禁用限值 (永久)	(字节 1)
	输出 1 禁用限值 (永久)	(字节 0)

41129 (1128)	输出 6 禁用限值 (永久)	(字节 1)
	输出 6 禁用限值 (永久)	(字节 0)

寄存器编号	说明
-------	----

※ 不执行限值控制：

- 如值大于等于最大称量值
- 如禁用值大于等于启用值

最低有效位值减少至秤盘的最小分度值。

称重模式设为传输模式（以一个字节表示，最低有效位）		
41201 (1200)	通道数量	(字节 1)
	通道数量	(字节 0)
41202 (1201)	通道 1 的状态寄存器	(MSB)
	通道 1 的状态寄存器	(LSB)
41203 (1202)	通道 1 的重量值	(字节 1)
	通道 1 的重量值	(字节 0)
41204 (1203)	通道 2 的状态寄存器	(MSB)
	通道 2 的状态寄存器	(LSB)
41205 (1204)	通道 2 的重量值	(字节 1)
	通道 2 的重量值	(字节 0)
41206 (1205)	通道 3 的状态寄存器	(MSB)
	通道 3 的状态寄存器	(LSB)
41207 (1206)	通道 3 的重量值	(字节 1)
	通道 3 的重量值	(字节 0)
41208 (1207)	通道 4 的状态寄存器	(MSB)
	通道 4 的状态寄存器	(LSB)
41209 (1208)	通道 4 的重量值	(字节 1)
	通道 4 的重量值	(字节 0)
41210 (1209)	通道 1 的净重值	(字节 1,0)
41211 (1210)	通道 2 的净重值	(字节 1,0)
41212 (1211)	通道 3 的净重值	(字节 1,0)
41213 (1212)	通道 4 的净重值	(字节 1,0)
41214 (1213)	通道 1 的皮重值	(字节 1,0)
41215 (1214)	通道 2 的皮重值	(字节 1,0)
41216 (1215)	通道 3 的皮重值	(字节 1,0)
41217 (1216)	通道 4 的皮重值	(字节 1,0)

寄存器编号	说明
-	通道 X 状态寄存器值的格式：参见 5.2.2 节
-	重量值显示的格式：绝对值（不带小数点）
例如：	如仪表设置 3 位小数点，3.000=3000
	如仪表设置 2 位小数点，3.00=300

5-2-1 指令寄存器

其在输出寄存器中的编号为 0，由 2 个字节组成，参见表 5.2.1。

指令执行

只有当后一个指令不同于前一个时，该指令才有效（因此重复上一个指令，需先发送“无指令”值，然后在发送重复的指令）。

表 5.2.1：指令寄存器

指令	指令寄存器值	说明
无指令	0 (0000 Hex)	无指令
清零指令	1 (0001 Hex)	秤盘清零
去皮指令	2 (0002 Hex)	自动去皮
手动去皮指令	3 (0003 Hex)	手动输入皮重值 (在参数 1 中输入值 ⁽²⁾)
净重切换指令	4 (0004 Hex)	显示净重值 ⁽³⁾
毛重切换指令	5 (0005 Hex)	显示毛重值 ⁽³⁾
切换至通道 1 指令	6 (0006 Hex)	切换至通道 1
切换至通道 2 指令	7 (0007 Hex)	切换至通道 2
切换至通道 3 指令	8 (0008 Hex)	切换至通道 3
切换至通道 4 指令	9 (0009 Hex)	切换至通道 4
开关量输出信号设置	25 (0019 Hex)	闭合 / 断开输出 ⁽⁴⁾
读取 ALIBI	30 (001E Hex)	通过菜单页读取 ALIBI 称重数据 (存储寄存器 8-15) ⁽⁵⁾
写入 ALIBI	31 (001F Hex)	储存 ALIBI 称重数据

(2) 注：参数 1 和参数 2 的格式

→ 对于手动去皮指令，仅用参数 1。

例如：如仪表设置 3 位小数点，3.000=3000

如仪表设置 2 位小数点，3.00=300

(3) 仅当启用了 NTGS 模式（净重 / 毛重转换）时，该功能才有效。

(4) 输出设置

通过参数 1，可设置开关量输出的状态：

参数 1：

位 0 → 开关量输出 1 其中：位 0 = 1 → 开关量输出 1 闭合；

 位 0 = 0 → 开关量输出 1 打开

位 1 → 开关量输出 2 其中：位 1 = 1 → 开关量输出 2 闭合；

 位 1 = 0 → 开关量输出 2 打开

扩展开关量输出（仅 DT85PB 型号）

位 2 → 开关量输出 3 其中：位 2 = 1 → 开关量输出 2 闭合；

 位 2 = 0 → 开关量输出 2 打开

位 3 → 开关量输出 4 其中：位 3 = 1 → 开关量输出 2 闭合；

 位 3 = 0 → 开关量输出 2 打开

位 4 → 开关量输出 5 其中：位 4 = 1 → 开关量输出 2 闭合；

 位 4 = 0 → 开关量输出 2 打开

位 5 → 开关量输出 6 其中：位 5 = 1 → 开关量输出 2 闭合；

 位 5 = 0 → 开关量输出 2 打开

位 6~15（不使用）

※ 用于设置开关量输出值的参数 1 和参数 2 的格式：位的配置

※ 如激活某个开关量输出链接至一个限值，则与该开关量输出有关的指令将视为无效。

※ 写入的限值不会自动保存在闪存中，这些写入的限值只是临时值。您应执行

写入闪存指令以保存这些值。

(5) 读取 ALIBI 称重数据

通过启动菜单页读取 ALIBI 称重数据（存储寄存器 8-15），在参数 1 中可设置循环序数，在参数 2 中可设置重量序数（ID）。

参数 1 和参数 2 的格式：绝对值（不带小数点）

表 5.2.1.A：读取 ALIBI 称重数据指令的启动菜单页的内容

	输入数据区域 (N 个字节)	说明
ALIBI 页面 (16 字节)	16	已储存的毛重值 (字节 3)
	17	已储存的毛重值 (字节 2)
	18	已储存的毛重值 (字节 1)
	19	已储存的毛重值 (字节 0)
	20	已储存的皮重值 (字节 3)
	21	已储存的皮重值 (字节 2)
	22	已储存的皮重值 (字节 1)
	23	已储存的皮重值 (字节 0)
	24	ID：称重次数 (字节 3)
	25	ID：称重次数 (字节 2)
	26	ID：称重次数 (字节 1)
	27	ID：称重次数 (字节 0)
	28	Alibi 状态寄存器 (MSB)
	29	Alibi 状态寄存器 (LSB)
	30	不使用
	31	不使用

5-2-1-1 Alibi 状态寄存器

其在存储寄存器的 Alibi 内存部分，以下为两个字节的含义：

位		含义
位 7- 位 0	→	循环序数（从 0 至 255）
位 10- 位 8	→	秤盘编号（从 1 至 4）
位 11	→	去皮类型：位 11 = 1 → 手动输入皮重值； 位 11 = 0 → 无效或半自动去皮
位 12	→	不使用
位 13	→	不使用
位 14	→	不使用
位 15	→	不使用

※ 通过存储寄存器 250 至 256 可读取最后次储存的称重数据。

5-2-2 通道 X 状态寄存器（表 5.2.2）

位	说明	位的含义	
		0	1
(LSB)			
0	重量值的极性	+	-
1	重量稳定性	NO	YES
2	重量欠载状态	NO	YES
3	重量过载状态	NO	YES
4	毛重在零值附近	不在零值附近	在零值附近
5	净重的极性	+	-
6	去皮的可用性	NO	YES
7	预设皮重可用性	NO	YES
(MSB)			
8	不使用		
9	不使用		
10	不使用		
11	不使用		
12	不使用		
13	不使用		
14	不使用		
15	不使用		

5-2-3 菜单区域

启动菜单区域寄存在闪存中（1024 个字节）且由 64 页组成（从 0 至 63）。

对于一台经贸易结算认证的仪表而言，其无法写入页 0 至页 38 前半部分之间的计量参数，仅可写入页 38 后半部分至页 63 之间的数据。

当仪表经贸易结算认证，如写入页 0 至页 37 之间任何一个数据，那么指令的响应结果是不被允许的（ExceptionCommandNotAllowed），但如果写入其它的数据，则响应正常（CommandOk）。在任何情况下，都无法完整的复制页 38 的数据，仅可复制后半部分数据。

	输入数据区域 (N 个字节)	存储寄存器	说明
菜单区域：页 5 (16 个字节)	340		
	341		
	342		
		字节 3	通道 1 的量程 1 (LSB)
	343	字节 2	通道 1 的量程 1
		字节 1	通道 1 的量程 1
	344	字节 0	通道 1 的量程 1 (MSB)
		字节 3	通道 1 的量程 2 (LSB)
	345	字节 2	通道 1 的量程 2
		字节 1	通道 1 的量程 2
	346	字节 0	通道 1 的量程 2 (MSB)
	347		

	输入数据区域 (N 个字节)	存储寄存器	说明
菜单区域：页 6 (16 个字节)	348		
		字节 1	通道 1 的量程 1 的分度值 (LSB)
	349	字节 0	通道 1 的量程 1 的分度值 (MSB)
		字节 1	通道 1 的量程 2 的分度值 (LSB)
	350	字节 0	通道 1 的量程 2 的分度值 (MSB)
	351		
		字节 0	通道 1 的小数位数
	352	字节 0	通道 1 的称重单位 ⁽⁵⁾
	353		
菜单区域：页 14 (16 个字节)	412	字节 3	通道 2 的量程 1 (LSB)
		字节 2	通道 2 的量程 1
	413	字节 1	通道 2 的量程 1
		字节 0	通道 2 的量程 1 (MSB)
	414	字节 3	通道 2 的量程 2 (LSB)
		字节 2	通道 2 的量程 2
	415	字节 1	通道 2 的量程 2
		字节 0	通道 2 的量程 2 (MSB)
	416		
	417		
	418	字节 1	通道 2 的量程 1 的分度值 (LSB)
		字节 0	通道 2 的量程 1 的分度值 (MSB)
	419	字节 1	通道 2 的量程 2 的分度值 (LSB)
		字节 0	通道 2 的量程 2 的分度值 (MSB)

	输入数据区域 (N 个字节)	存储寄存器	说明
菜单区域：页 15 (16 个字节)	420		
	421	字节 0	通道 2 的小数位数
		字节 0	通道 2 的称重单位 ⁽⁵⁾
	422		
	423		
	424		
	425		
	426		
	427		
菜单区域：页 22 (16 个字节)	476		
	477		
	478		
	479		
	480		
	481		
		字节 3	通道 3 的量程 1 (LSB)
	482	字节 2	通道 3 的量程 1
		字节 1	通道 3 的量程 1
	483	字节 0	通道 3 的量程 1 (MSB)
		字节 3	通道 3 的量程 2 (LSB)

	输入数据区域 (N 个字节)	存储寄存器	说明
菜单区域：页 23 (16 个字节)	484	字节 2	通道 3 的量程 2
		字节 1	通道 3 的量程 2
	485	字节 0	通道 3 的量程 2 (MSB)
	486		
	487	字节 1	通道 3 的量程 1 的分度值 (LSB)
		字节 0	通道 3 的量程 1 的分度值 (MSB)
	488	字节 1	通道 3 的量程 2 的分度值 (LSB)
		字节 0	通道 3 的量程 2 的分度值 (MSB)
	489		
菜单区域：页 31 (16 个字节)	490	字节 0	通道 3 的小数位数
		字节 0	通道 3 的称重单位 ⁽⁵⁾
	491		
	548		
	549		
	550		
	551	字节 3	通道 4 的量程 1 (LSB)
		字节 2	通道 4 的量程 1
	552	字节 1	通道 4 的量程 1
		字节 0	通道 4 的量程 1 (MSB)
	553	字节 3	通道 4 的量程 2 (LSB)
		字节 2	通道 4 的量程 2
	554	字节 1	通道 4 的量程 2
		字节 0	通道 4 的量程 2 (MSB)
	555		

	输入数据区域 (N 个字节)	存储寄存器	说明
菜单区域：页 32 (16 个字节)	556		
	557	字节 1	通道 4 的量程 1 的分度值 (LSB)
		字节 0	通道 4 的量程 1 的分度值 (MSB)
	558	字节 1	通道 4 的量程 2 的分度值 (LSB)
		字节 0	通道 4 的量程 2 的分度值 (MSB)
	559		
	560	字节 0	通道 4 的小数位数
		字节 0	通道 4 的称重单位 ⁽⁵⁾
	561		
	562		
	563		

⁽⁵⁾ 注：以下是数字值对应的称重单位：

- | | | |
|---|---|----------------|
| 0 | → | Grams (克) |
| 1 | → | Kilograms (千克) |
| 2 | → | Tons (吨) |
| 3 | → | Pounds (磅) |

5-3 开关量输出状态数据区域

“开关量输出状态”数据区域由主设备写入（由仪表读取），由 1 位开关量输出组成。

表 5.3：Modbus 开关量输出状态

开关量输出的编号	说明开关量输出状态	位的含义	
		0	1
00001 (0)	数字输出 1 (1)	未激活	激活
00002 (1)	数字输出 2 (1)	未激活	激活

仅 DT85

开关量输出的编号	说明开关量输出状态	位的含义	
		0	1
00003 (2)	数字输出 3 (1)	未激活	激活
00004 (3)	数字输出 4 (1)	未激活	激活
00005 (4)	数字输出 5 (1)	未激活	激活
00006 (5)	数字输出 6 (1)	未激活	激活

(1) 如果相关的输出未启用某项功能，那么写入是有效的。

6 标定指令

标定存储寄存器

寄存器编号	说明
40901	标定点数量
40902	第一个标定点的重量 (高)
40903	第一个标定点的重量 (低)
40904	第二个标定点的重量 (高)
40905	第二个标定点的重量 (低)
40906	第三个标定点的重量 (高)
40907	第三个标定点的重量 (低)
40908	零点 ADC 值 (高)
40909	零点 ADC 值 (低)
40910	第一个标定点的 ADC 值 (高)
40911	第一个标定点的 ADC 值 (低)
40912	第二个标定点的 ADC 值 (高)
40913	第二个标定点的 ADC 值 (低)
40914	第三个标定点的 ADC 值 (高)
40915	第三个标定点的 ADC 值 (低)

寄存器编号	说明
40951	重量单位 (g,kg,t,lb) 分别对应 (0,1,2,3)
40952	第一个量程的分度值
40953	第二个量程的分度值
40954	小数位数
40955	第一个量程值 (高)
40956	第一个量程值 (低)
40957	第二个量程值 (高)
40958	第二个量程值 (低)

输入寄存器：30116 标定状态中值的含义

值	说明
0	未进入标定状态
1	正在获取标定内码
2	内码获取成功
3	内码获取失败（例：获取时间大于 40 秒，重量不稳定）
4	标定成功
5	标定失败

特定指令

指令寄存器值（ 参见“指令寄存器”章节）	指令	说明
35 (23H)	READ CALIBRATION	将仪表 FLASH 芯片中的标定数据读取至寄存器中
36 (24H)	WRITE CALIBRATION	保存标定数据
37 (25H)	CALIBRATION POINT	执行标定
38 (26H)	CALIBRATION ABORT	取消标定

使用 30116 输入寄存器指令可读取标定状态。

标定步骤 Calibration sequence:

1 使用 40001 指令寄存器的指令 35，将仪表 FLASH 芯片中的标定数据读取至寄存器中，参数 40003 与 40002 中，0 为 1 号通道，1 为 2 号通道，依次类推，若为非独立模式，仅可选 0。

2 可通过寄存器修改 40951-40958

3 设置标定寄存器 40901-40907（如果是理论标定，只需设置 40908-40915）

4 设置完毕后，清空秤盘，读取寄存器 40001，先输入指令 37，参数 40003 与

40002，设为 0（0 为第一个标定点，1 为第二个标定点，依次类推）。再输入指令 36 保存标定数据。成功后加载秤盘，输入指令 37，参数 40003 与 40002，设为 1，再输入指令 36 保存标定数据。

单秤标定：

40951-40958 根据实际情况写入

写入 40001-37-0-0 零点标定

写入 40001-36 保存

写入 40001-37-0-1 第一点标定

写入 40001-36 保存

多秤标定：

写入 40001-35-0-0 读取第一个秤数据

40951-40958 根据实际情况写入

写入 40001-37-0-0 零点标定

写入 40001-36 保存

写入 40001-37-0-1 第一点标定

写入 40001-36 保存

写入 40001-35-0-1 读取第二个秤数据

步骤 2-6

多标定点：

40951-40958 根据实际情况写入

写入 40001-37-0-0 零点标定

写入 40001-36 保存

写入 40901 至 40907 根据实际情况写入点（举例 2 个点，40901 写 2）

写入 40001-37-0-1 第一点标定

写入 40001-38 退出

写入 40001-37-0-2 第二点标定

写入 40001-36 保存

天津市丽景微电子有限公司

地址：天津市华苑产业区环外海泰南道 28
号海泰国际产业基地 C 座 7 门 302 室

邮编：300384

免费服务电话：4000-913-988

电话：022-83719630/83719631/83719632

传真：022-58558380

<http://www.lascaux.com.cn>

E-mail: mail@lascaux.com.cn