

MAD50系列

数字控制器 通讯接口 (RS-485) 使用手册

感谢您购买 SHIMAX 产品。请检查您订购的产品
请您阅读本手册并且完全理解以后再操作本设备

本使用手册介绍MAD50的通讯接口选件功能。关于MAD50的操作和各参数的详细说明，请参见使用说明书。

目录

1. 概述	2	5. 标准通讯协议概述	6~14
2. 技术规格	2	5-1 通讯步骤	6
3. 与主机计算机的连接	3	5-2 通讯格式	6
3-1 RS-485	3	5-3 读命令 (R) 详述	10
3-2 三态输出的控制	3	5-4 写命令 (W) 详述	12
4. 设置通讯参数	4~5	5-5 应答代码详述	13
4-1 设置通讯速度	4	5-6 通讯数据地址详述	14
4-2 设置通讯校验位	4	6. MODBUS 通讯协议概述	15~23
4-3 设置通讯停止位	4	6-1 通讯步骤	15
4-4 设置起始字符	4	6-2 通讯格式	16
4-5 设置 BCC 块校验	4	6-3 错误检查	18
4-6 设置通讯地址	5	6-4 数据读出详述	19
4-7 设置延迟时间	5	6-5 数据写入详述	20
4-8 设置通讯存储方式	5	6-6 循环测试详述	21
		6-7 无响应的条件	21
		6-8 错误信息详述	22
		6-9 通讯数据地址详述	23
		7. 通讯主机方式概述	24~26
		8. 补充说明	27
		8-1 量程代码表	27
		8-2 事件代码表	27
		9. ASCII 代码表	27

1. 概述

MAD50 通讯接口采用 **RS-485** 通讯方法。

基于 **EIA** 标准通过信号能够传送各种数据, 可以通过个人计算机读取。

RS-485 是电子工业协会(**EIA**)的通讯标准。

此标准规定了电器和机械硬件。


没有规定数据传输过程的软件部分。

因此, 同样的接口不能够总是互相通讯。

因此, 用户在使用前需要完全了解技术规格和传输过程。

使用 **RS-485** 能够并连两台或多台 **MAC3** 和 **MAC50**。

很多个人计算机不支持这种接口。

然而, 可以使用 **RS-232C**  **RS-485** 通讯转换器。

2. 技术规格

协议	: SHIMAX 标准协议, MODBUS ASCII, MODBUS RTU
信号电平	: 符合 EIA RS-485
通讯方式	: RS-485 两线系统 半双工多点 (总线) 系统
同步系统	: 起始-停止 同步系统
通讯距离	: RS-485 最大 500m (取决于环境条件)
传输速度	: 9600 和 19200 bps
传输控制	: 无
起始位	: 1 位
数据长度	: 8 位
校验位	: 无, 偶校验, 奇校验
停止位	: 1 位, 2 位
通讯代码	: ASCII 代码 (SHIMAX 标准协议, MODBUS ASCII) 二进制代码 (MODBUS RTU)
可以连接台数	: 32 (包括主控制器)
隔离	: 与模拟输出不隔离。 与输入、输出和电源隔离

*MODBUS 是 Schneider Electric 的商标。

3. 与主机算的连接

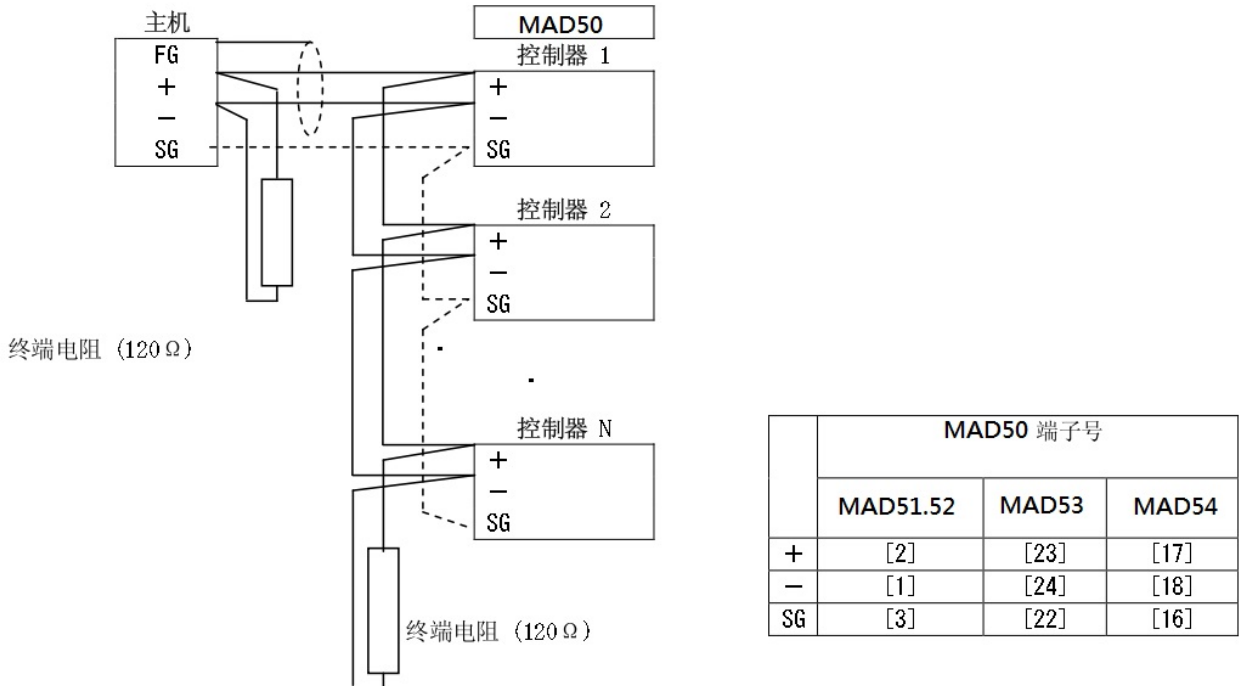
3-1. RS-485

MAD50 的输入和输出逻辑电平表示如下：

逻辑 (1) 状态 - 端 < + 端
 逻辑 (0) 状态 - 端 > + 端

然而，在开始传输之前控制器的+ 端和 - 端为高阻状态。上述电平为输出。（参见 3-2. 3 态输出的控制）

[RS-485]



注 1: 工作时，在主机端和设备终端(+ 和 - 之间)分别安装 1/2W 120Ω 终端电阻。

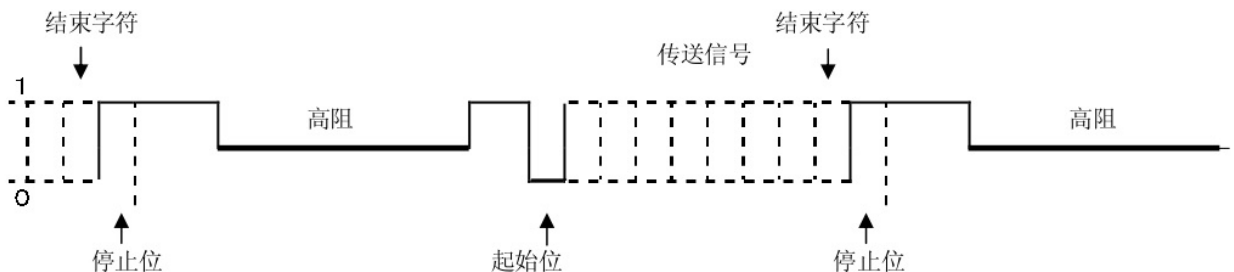
注 2: 请确认屏蔽层单端接地。
 当不用屏蔽线时，用户必须采取措施预防雷击。

3-2. 3 态输出的控制

RS-485 是多点系统。传输输出在未通讯和接受状态总是高阻状态，以避免发送信号的冲突。

仅在发送前从高阻状态变为正常输出状态。在传送完成时再次返回高阻状态。

然而，3 态输出控制有大约 2ms (最大) 的时间延迟。当主机端在结束接收后立即开始传送，设置大于几 ms 的延迟时间。



4. 设置通讯参数



MAD50 在窗口群 9 共有 8 种通讯参数。
除了通讯存储方式之外，这些参数不能用通讯改变。
通过前面板按键完成设置。

MENU 键

4-1. 设置通讯速度



初值 : 96
范围 : 96 (9600bps) 192 (19200bps),

选择和设置与主机相同的通讯速度。

MENU 键

4-2. 设置通讯校验位



初值 : 无
范围 : 无, odd 奇校验, even 偶校验

选择和设置通讯校验位

MENU 键

4-3. 设置通讯停止位



初值 : 1
范围 : 1,2

选择和设置通讯停止位。

MENU 键

4-4. 设置起始字符



初值 : STX
范围 : STX, ATT

选择使用的控制代码(仅当 SHIMAX 标准协议时有效)

MENU 键

选择	起始字符	报文结束字符	结束字符
STX	STX (02H)	ETX (03H)	CR (0DH)
ATT	"@" (40H)	":" (3AH)	CR (0DH)

4-5. 设置 BCC 块校验



初值 : 无
范围 : none, Add, Add2, Xor, LrC, Cr16

选择 BCC 块校验类型。选择的内容决定通讯协议。

MENU 键

选择	运算方法	协议
none	无	SHIMAX 标准协议
Add	累加和	
Add 2	累加和+按位求反	
Xor	异或	MODBUS ASCII
LrC	LRG	
Cr16	CRC-16	

4-6. 设置通讯地址



初值 : 1

范围 : 1~255

RS-485 符合多点系统, 可以连接 31 台设备 (最多)

通过分配地址 (机器号) 给每台设备, 使仅指定地址的设备能够响应

MENU 键



注 1: 能够设置地址 1~255。然而, 能够连接设备的最大数是 31 台。

注 2: 适合 MODBUS 的地址是 1~247。

4-7. 设置延迟时间



初值 : 20
范围 : 1~250(ms)

能够设置从接收通讯命令到实际发送数据的最小延迟时间。

MENU 键

注 1: 某种通讯转换器可能需要更长的时间用于 3 态控制, 在这种 RS-485 中可能发生信号冲突。如果延迟时间更长, 则可以避免这种冲突。

当通讯速度低时, 需要特别注意。

注 2: 从接收通讯命令到实际发送数据的实际延迟时间, 是上面提到的延迟时间与软件处理时间的总和。特别是当写命令时, 命令处理时间大约需要 400 ms。

4-8. 设置通讯存储方式



初值 : RAM
范围 : RAM,MIX,EEP

由于非挥发存储器 EEPROM 的写次数是有限的, 当通过通讯频繁修改数据时 EEPROM 的寿命变短。

MENU 键

当通过通讯频繁修改数据时设置 RAM 方式。如果仅修改 RAM 数据而不修改 EEPROM, 可以延长 EEPROM 的寿命。

选择	处理过程
RAM	在此方式中, 通过通讯改变数据, 仅修改 RAM。当关断电源而不修改 EEPROM 时, RAM 数据丢失。如果再次接通电源, 将使用存储在 EEPROM 中的数据。
MIX	在此方式中, 仅在 RAM 中修改 SV1-SV4 和 OUT 1 ~ 2 数据, 在 RAM 和 EEPROM 中修改其它数据。
EEP	任何时间通过通讯在 RAM 和 EEPROM 中修改数据。即使断开电源, 数据仍然被保存。



5. 标准通讯协议概述

MAD50 遵守 SHIMAX 标准通讯协议。

即使连接满足标准通讯协议的不同系列的设备，用同样的通讯格式能够改变数据。

5-1. 通讯步骤

(1) 主机和从表之间的关系

- 个人计算机, PLC (主) 是主机。
 - MAD50是从机。
 - 通讯从主机的通讯命令开始, 以从机的通讯应答结束。
- 然而, 当通讯异常时, 例如, 通讯格式错误、BCC 错误, 没有通讯应答。

(2) 通讯步骤

主机发送命令, 从机应答主机, 传送数据, 通讯步骤完成。

(3) 超时

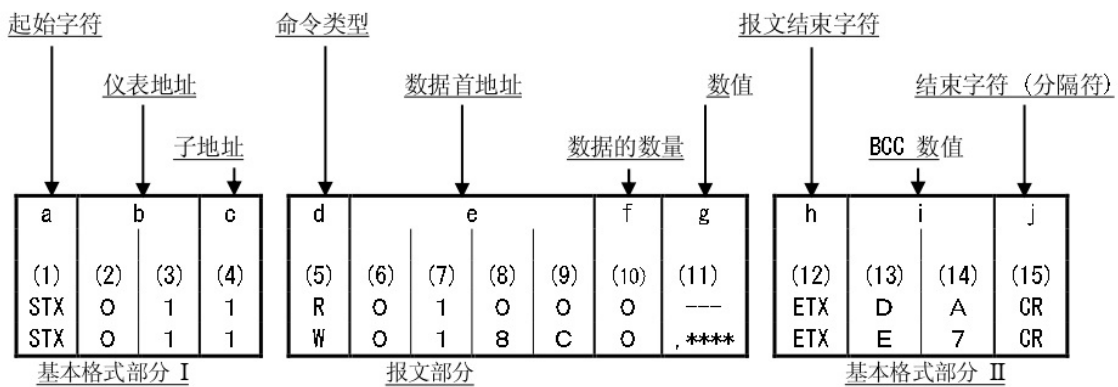
从机接收起始字符后, 在 1 秒内没有接收到结束字符, 视为超时。等待下一个命令 (新的起始字符)。
在主机端设置超时, 设为大于 1 秒。

5-2. 通讯格式

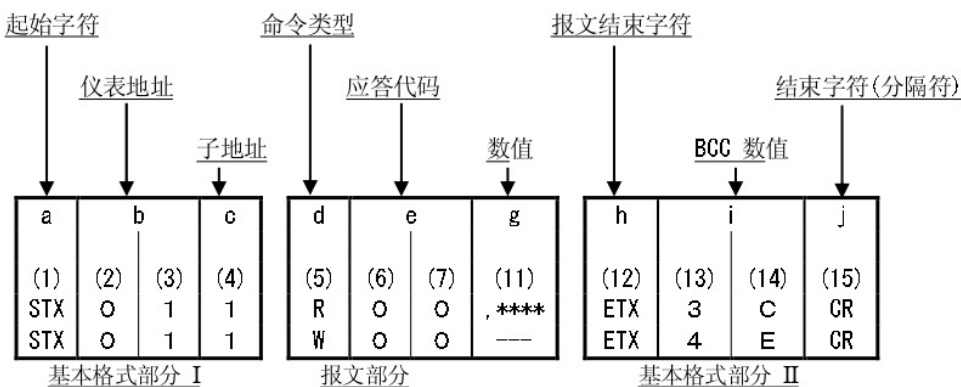
(1) 通讯格式概述

通讯格式由基本格式部分 I, 报文部分和基本格式部分 II 组成。

1) 通讯命令格式概述



2) 通讯应答格式



- 对于读命令(R)、写命令(W)和通讯应答, 基本格式部分 I, II 是相同的。
BCC 的每次运算结果插入到 BCC 数值, <i(13), (14)>。
- 报文部分根据通讯命令、数据地址、通讯应答等改变。

(2) 基本格式部分 I 详述

- a: 起始字符 [(1): 单字节 / STX (02H), 或 "@" (40H)]
- 此字符表示这是通讯开始。
 - 如果接收到起始字符, 判断为新通讯的第一个字符。
 - 成对地选择报文的起始字符和结束字符。
- (参见 4-4. 设置起始字符)

STX (02H) ---选择 ETX (03H)
"@"(40H) ---选择 ":" (3AH).

b: 仪表地址 [(2), (3): 双字节]

- 指定用于通讯的仪表。
- 可以指定地址 1~255 (十进制数)。
- 二进制 8 位数值 (1:0000 0001 - 255:1111 1111) 分为高 4 位和低 4 位, 变换为 ASCII 字符。

(2): 高 4 位的数字变换为 ASCII。
(3): 低 4 位的数字变换为 ASCII。

c: 子地址 [(4): 单字节]

- 固定为 (4) = 1 (31H), 因为 MAC10 是单回路仪表。
- 当指定其他子地址时, 因为子地址错误没有应答。

(3) 基本格式部分 II 详述

h: 报文结束字符 (12): 单字节 / ETX (03H), 或 ":" (3AH)]

- 它表示报文部分结束。

i: BCC 块校验 [(13)(14): 双字节]

- BCC 块校验用于检查通讯数据的异常。
 - 当 BCC 运算结果显示 BCC 错误时, 没有应答响应。
 - 有下面 4 种类型的 BCC 运算。
- (可以设置 BCC 运算类型, 参见 4-6. 设置 BCC 块校验类型)

1) 无

不执行 BCC 运算。 (13) 和 (14) 被省略。

2) 累加和

从起始字符 (1) 到报文的结束字符 (12), 以数据的 ASCII 1 个字符(1 字节)为单位执行累加和。

3) 累加和 + 按位取反

从起始字符 (1) 到报文的结束字符 (12), 以数据的 ASCII 1 个字符(1 字节)为单位执行累加和。将运算结果的最低字节按位取反。

4) 异或

从起始字符后的<仪表地址 (2)> 到报文的结束字符 (12), 以数据的 ASCII 1 个字符(1 字节)为单位执行 XOR (异或)运算。

- 不管数据位的长度 (7 或 8 位), 都以 1 字节 (8 位)为单位计算。
- 根据以上提到的运算结果, 将最低 1 字节数值分为高 4 位和低 4 位, 并变换为 ASCII 数值。

(13): 高 4 位的数值变换为 ASCII。

(14): 低 4 位的数值变换为 ASCII。

例 1: BCC 读命令 (R)的累加和.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(12)	(13)	(14)	(15)
STX	0	1	1	R	0	1	0	0	0	ETX	D	A	CR

$$02H + 30H + 31H + 31H + 52H + 30H + 31H + 30H + 30H + 30H + 03H = 1DAH$$

累加结果 (1DAH)的最低 1 字节 = DAH

(13) : "D" = 44H 、 (14) : "A" = 41H

例 2: BCC 读命令 (R)的累加和 + 按位取反

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(12)	(13)	(14)	(15)
STX	0	1	1	R	0	1	0	0	0	ETX	2	6	CR

$$02H + 30H + 31H + 31H + 52H + 30H + 31H + 30H + 30H + 30H + 03H = 1DAH$$

累加结果 (1 DAH) 的最低 1 字节 = DAH

按位取反 (DAH) = 26 H

(13) : "2" = 32H , (14) : "6" = 36H

例 3: BCC 读命令 (R)的异或

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(12)	(13)	(14)	(15)
STX	0	1	1	R	0	1	0	0	0	ETX	5	0	CR

$$30H \oplus 31H \oplus 31H \oplus 52H \oplus 30H \oplus 31H \oplus 30H \oplus 30H \oplus 30H \oplus 03H = 50H$$

- = XOR (异或)

取运算结果的最低 1 字节 (50H) = 50H

(13) : "5" = 35H , (14) : "0" = 30H

j: 结束字符(分隔符) [(15): 单字节 / CR]

- 这表示通讯结束。

(4) 基本格式部分 I, II 的共同条件

- 1) 在基本格式部分当下确认列异常后, 没有应答。
 - 当有硬件错误时。(溢出、帧和校验错误)。
 - 当仪表地址和子地址与指定的仪表地址不同时。
 - 当字符不在上面提到的通讯格式中的正常位置时。
 - 当 BCC 运算结果不同于 BCC 数据时。
- 2) 每 4 位二进制数值变换成 ASCII 数值。
- 3) 在十六进制中,使用大写字母<A>~<F> 变换为 ASCII 数值。

(5) 报文部分概述

报文部分随命令类型和通讯应答改变。

详见 5-3. 读命令 (R) 详述和 5-4. 写命令 (W) 详述。

d: 命令类型 [(5):单字节]

- "R" (52H/大写字符): 这表示是读命令和读命令应答。
当数据被读出到个人计算机、PLC 等时使用。
- "W" (57H/大写字符): 这表示是写命令和写命令应答
当数据被个人计算机、PLC 等写入 (修改) 时使用。
- 当识别出使用的字符非 "R" 和 "W" 时, 没有应答。

e: 数据首地址 [(6), (7), (8), (9): 4 字节]

- 在读命令 (R) 和写命令 (W) 时, 指定读出和写入数据的首地址。
- 首地址被定义为 16 位二进制数值 (1 字 /0~65535)。
- 16 位数值被分成 每 4 位一组, 转换成 ASCII 数值。

二进制数字 (16 位)	D15,D14,D13,D12	D11,D10,D9,D8	D7,D6,D5,D4	D3,D2,D1,D0
	0 0 0 0	0 0 0 1	1 0 0 0	1 1 0 0
十六进制数值	0 H	1 H	8 H	C H
	"0"	"1"	"8"	"C"
ASCII 数值	3 0 H	3 1 H	3 8 H	4 3 H
	(6)	(7)	(8)	(9)

- 关于数据地址, 见 7. 通讯数据表

f: 数据的数量 [(10): 单字节]

- 在读命令 (R) 和写命令 (W) 时, 指定读出和写入数据的数量。
- 通过将 4 位二进制数值转换为 ASCII 数值指定数据的数量。
- 在读命令 (R) 时, 可以按下面的范围指定:
"0" (30H) (1) ~ "9" (39H) (10)
- 在写命令 (W) 时, 固定为 "0" (30H) (1)。
- 实际数据的数量是 < 数据的数量 = 指定的数值 + 1 >

g: 数据 [(11): 组数取决于数据的数量]

- 在写命令 (W) 时写入数据 (改变数据), 同样在读命令 (R) 时读出数据
- 数据格式如下:

g (11)

	第一组数据				第二组数据					第 N 组数据			
	高			低	高			低		高			低
" , "	1	2	3	4	1	2	3	4		1	2	3	4
2CH													

- 当没有错误时逗号 (" , "2CH) 加在第一组数据的前面, 表示后面的部分是数据。
- 数据之间不使用分隔符号。
- 数据的数量取决于通讯命令格式 f:(10) 中的数值。
- 数据按二进制数值 16 位 (1 个字) 表示, 不含小数点。
- 各数据的小数点位置是不同的。
- 按每 4 位划分 16 位数值, 并转换成 ASCII 数值。
- 关于数据, 详见 5-3. 读命令 (R) 详述和 5-4. 写命令 (W) 详述

- e: 应答代码 [(6), (7): 双字节]
- 指定 读命令(R) 和写命令 (W)的应答代码。
 - 二进制 8 位数值 (0~255) 分为高 4 位和低 4 位, 并分别转换成 ASCII 代码。
 - (6):高 4 位数值转换成 ASCII。
 - (7): 低 4 位数值转换成 ASCII。
 - 正常应答, 指定"0" (30H) 和 "0" (30H) 。
 - 异常应答, 指定异常代码号, 并转换成 ASCII 数据。
 - 关于应答代码, 详见 5-5. 应答代码详述。

5-3. 读命令 (R) 详述

当读各种数据到个人计算机和 PLC 时, 使用读命令(R)。

(1) 读命令 (R) 格式

- 读命令 (R) 的报文格式如下:
- (基本格式部分 I 和 II 对所有命令和应答是同样的。)

报文部分					
d	e				f
(5)	(6)	(7)	(8)	(9)	(10)
R	0	4	0	0	4
52H	30H	34H	30H	30H	34H

- d:表示读命令。
- e: 指定读出数据的首地址。
- f: 指定从首地址开始读出数据的数量。

- 上述命令如下:

读出的首地址 = 0400H (十六进制数)
= 0000 0100 0000 0000 (二进制数)

读出数据的数量 = 4H(十六进制数)
= 0100(二进制数)
= 4(十进制)

(实际的数据数量) = 5 (4+1)

即, 从地址 0400H 开始读出 5 个数据。

(2) 读命令 (R) 的正常应答

- 读命令 (R) 的正常应答格式 (报文部分) 如下:
- (基本格式部分 I 和 II 对所有命令和应答是相同的。)

报文部分															
d	e		g												
(5)	(6)	(7)	第一组数据				第二组数据				第五组数据				
R	0	0	,	0	0	1	E	0	0	7	8	0	0	0	3
52H	30H	30H	2CH	30H	30H	31H	45H	30H	30H	37H	38H	30H	30H	30H	33H

- d(5) : <R (52H)> 表示是读命令 (R) 应答。
- e(6),(7) : < 00 (30H, 30H) > ,表示是读命令(R)的正常应答。

- g (11) : 读命令(R) 的应答数据。

数据的格式如下:

1. 首先, <, (2CH) >, 表示数据头。
2. 其次, 是< 从首地址开始读出的数据>, 根据<读出数据的数量>依次读出其他数据。
3. 在数据之间不插入字符。
4. 一个数据包含 16 位 (1 个字) 二进制数据, 不含小数点。每 4 位数据被转换成 ASCII 代码。
5. 每个数据的小数点位置是不同的。
6. 响应数据的字符数量如下:
字符数量 = 1 + 4 × 读出数据的数量

- 以下是上述读命令(R)的响应数据:

读出首地址 数据地址 (0400H)	读出数据的数量 (4H : 5)	数据地址	数据	
		16 位 (1 个字) 十六进制数	16 位 (1 个字) 十六进制数	十进制数
	0	0400	001E	30
	1	0401	0078	120
	2	0402	001E	30
	3	0403	0000	0
	4	0404	0005	5

(3) 读命令 (R)的异常应答格式

- 读命令 (R) 的异常应答格式 (报文部分) 如下:
(基本格式部分 I 和 II 对于所有命令和应答是相同的)

报文部分

d	e	
(5)	(6)	(7)
R	0	7
52H	30H	37H

- d (5): <R (52H) >, 表示读命令的应答。
- e (6), (7): 应答代码, 表示读命令 (R) 异常应答
- 关于异常代码, 详见 5-5. 应答代码详述。
- 异常应答时没有应答数据。

5-4. 写命令 (W) 详述

当从个人计算机和 PLC 写入 (或改变) 各种数据时, 使用写命令 (W)。

1) 写命令 (W) 格式

-写命令 (W) 的报文部分格式如下:

(基本格式部分 I 和 II 对于所有命令和应答是相同的。)

报文部分										
d	e				f	g				
(5)	(6)	(7)	(8)	(9)	(10)	(11)				
W	0	4	0	0	0	写入的数据				
57H	30H	34H	30H	30H	30H	2CH	30H	30H	32H	38H

- d: 表示写命令。固定为 "W" (57H)。

- e: 指定写入 (改变) 数据的首地址。

- f: 指定写入 (改变) 数据的数量。

写入数据的数量固定为 "0" (30H), 1 个数据。

- g: 指定写入 (改变) 数据:

1. <, (2CH)>, 表示数据头。

2. 接着, 是写入数据。

3. 数据由二进制数值组成, 16 位 (1 个字) 不包含小数点, 每 4 位转换成 ASCII 代码。

4. 各数据的小数点位置不同。

- 上述命令如下:

写入首命令地址 = 0400H (十六进制数)

= 0000 0100 0000 0000 (二进制数)

写入数据的数量 = 0H (十六进制数)

= 0000 (二进制数)

= 0 (十进制数)

(数据的实际数量) = 1 个 (0+1)

写入数据 = 0028 (十六进制数)

= 0000 0000 0010 1000 (二进制数)

= 40 (十进制数)

数据地址 0400H, 写入 (改变) 一个数据 (40: 十进制数)。

地址 (400H) → 0
写入数据的数量
一个 (0 1)

数据地址		数据	
16 位 (1 个字)		16 位 (1 各字)	
十六进制	十进制	十六进制	十进制
0400	1024	0028	40
0401	1025	0078	120
0402	1026	001E	30

(2) 写命令(W)的正常应答格式

- 写命令(W) 的正常应答格式 (报文部分) 如下:

(基本格式部分 I 和 II 对于所有命令和应答是相同的。)

报文部分		
d	e	
(5)	(6)	(7)
W	0	0
57H	30H	30H

- d (5) : <W (57H)>, 表示写命令 (W) 的应答。

- e (6), (7): <00 (30H, 30H)>, 写命令 (W) 的正常应答。

(3) 写命令(W)时的异常应答格式

-写命令(W)时的异常应答格式如下：
(基本格式部分 I 和 II 对所有命令和应答是一样的。)

报文部分

d	e	
(5)	(6)	(7)
W	0	9
57H	30H	39H

- d (5) : <W (57H)>, 表示写命令(W)的应答。
- e (6), (7) : 异常响应代码, 表示写命令 (W)的异常应答。
- 关于应答异常代码, 见 5-5. 应答代码详述。

5-5. 应答代码详述

1) 应答代码类型

- 通讯读命令 (R) 和写命令 (W) 的应答总是包括应答代码。
- 应答代码分成两类



- 应答代码由二进制 8 位数字组成 (0~255)。
- 应答代码类型如下:

应答代码表

应答代码		代码类型	代码内容
二进制	ASCII		
0000 0000	"0", "0" : 30H, 30H	正常应答	- 正常应答代码
0000 0111	"0", "7" : 30H, 37H	报文格式错误	<ul style="list-style-type: none"> - 当非 0~9 数字被指定为数据的数字时 - 当包含非 0~9 和 A~F 的十六进制数字时 <li style="padding-left: 20px;">- 当逗号 ", "没有在指定位置时
0000 1000	"0", "8" : 30H, 38H	数值的数据地址错误	<ul style="list-style-type: none"> - 当指定不存在的地址时 - 当写只读时 - 当读只写时 - 当非 0 指定给写命令的数据数量时
0000 1001	"0", "9" : 30H, 39H	数据错误	<ul style="list-style-type: none"> - 当写入数据超量程时
0000 1010	"0", "A" : 30H, 41H	执行命令错误	<ul style="list-style-type: none"> - 当接收执行命令处于不合适的状态时 (当 RUN/STBY 分配给 DI 时, 执行 RUN/STBY 写命令)
0000 1011	"0", "B" : 30H, 42H	写方式错误	<ul style="list-style-type: none"> - 当接收写命令处于不可能写数据的环境时 (例如在自动控制时写手动输出值)
0000 1100	"0", "C" : 30H, 43H	规格选择错误	<ul style="list-style-type: none"> - 当写命令未包含指定的规格或者选择时

(2) 应答代码的优先权

应答代码值低, 应答代码优先权高
当多个应答代产生时, 返回优先权高的应答代码

5-6. 通讯数据地址详述

1) 数据地址

- 对于数据地址, 16 位二进制数按每 4 位组成 1 位十六进制数表示。

2) 关于读出 (读)/写入 (写)

- R/W 表示数据可读、可写。
- R 是只读数据。
- W 是只写数据。

- 当只写数据地址指定为读命令 (R) 时, 或当只读数据地址指定为写命令 (W) 时, 表示为数据地址错误。异常应答代码, ="0", "8" (30H, 38H), 表示"报文部分的数据格式,数据地址和数据数量错误"。

3) 数据地址和数据数量

- 当不在数据地址表中的数据地址指定为首数据地址时, 表示为数据地址错误。异常应答代码, ="0", "8" (30H, 38H), 表示"报文部分的数据格式,数据地址和数据数量错误"。
- 当增加数据地址超出数据地址表的范围时, 总是将"0000 H" (30H, 30H, 30H, 30H) 做为应答数据。

4) 数据

- 因为每个 16 位数据没有小数点, 所以不需要检查数据类型和小数点位置。
- 所有数据按二进制数字处理 (16 位数字: -32768 ~ 32767)。

举例: 带小数点数的表示方法

		十六进制数
20.0	→ 200	→ 00C8
100.00	→ 10000	→ 2710
-40.00	→ -4000	→ F060

举例: 表达 16 位数据的方法

不同进制的数据	
十进制数	十六进制数
0	0000
1	0001
~	~
32767	7FFF
-32768	8000
-32767	8001
~	~
-2	FFFE
-1	FFFF

5) 选件的数据

- 当选件没有安装, 其参数数据地址不在数据表中, 用读 (R) 或者写 (W) 命令访问这些指定的地址时,异常应答代码 "0", "C" (30H, 43H) "表示选件错误"。

6) 因为工作状态或者设置造成参数不显示

- 因为工作状态或者设置造成参数不显示, 此参数能够通过通讯读出。然而, 写入时, 产生异常应答代码"0", "B" (30H, 42H) "写方式错误"。

6. MODBUS 通讯协议概述

MODBUS 有两种方式：RTU 方式和 ASCII 方式。参见 4-6. 设置 BCC 运算类型。

RTU 和 ASCII 方式的比较

项目	RTU	ASCII
传送代码	二进制 8 位	ASCII
错误检查	CRC-16	LRC
起始位	1 位	
数据长度	8 位	7 位 / 8 位
校验位	无 / 偶校验 / 奇校验	
停止位	1 位	
起始字符	无	":"(3AH)
结束字符	无	CR(0DH)+LF(0AH)
数据时间间隔	小于 28 位传输时间	小于 1 秒

6-1. 通讯步骤

1) 主从之间的关系

- 个人计算机和 PLC 是主边。
 - MAD50 是从边。
 - 通讯从主边的通讯命令开始，到从边的通讯应答完成。
- 然而，当异常发生时没有通讯应答，例如，通讯格式错误或 BCC 错误等。

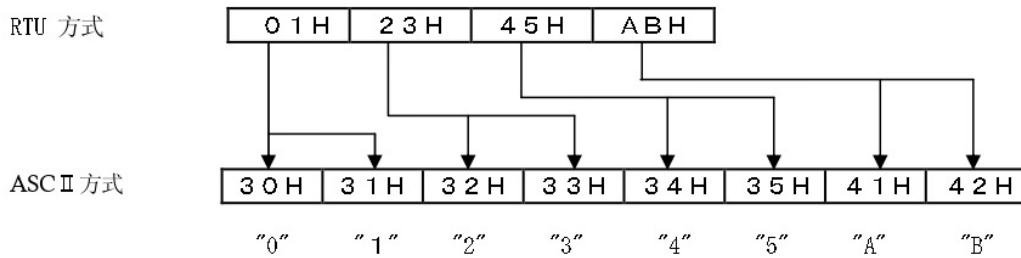
2) 通讯步骤

主边发送通讯命令，从边应答主边，传回通讯命令应答，完成通讯步骤。

3) 通讯数据

RTU 方式按 8 位二进制传输。

在 ASCII 方式中，RTU 的 8 位二进制变换为两个字符的 ASCII 代码并传输。

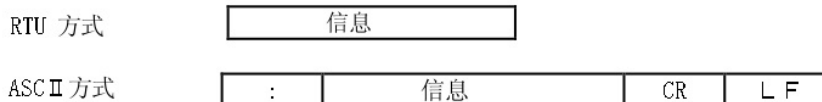


4) 信息比较

RTU 方式仅由一条信息组成。

ASCII 方式由起始字符 ":" (3AH) + 信息 + 结束字符, CR (0DH) + LF (0AH).

信息



5) 超时

-RTU 方式

当信息停止超过相当于 28 位的传输时间时，视为信息结束。

当在信息传送过程中产生相当于 28 位的时间空白时，视为信息超时。这是非完整信息，因此，从表没有应答。

* 参考：相当于 28 位的时间 (单位 = ms)

1200bps:23.4 2400bps:11.7 4800bps:5.9 9600bps:3.0 19200bps:1.5 38400bps:0.8

-ASCII 方式

在接收到起始字符后，当 1 秒内没有接收到结束字符时视为超时。并等待下一个命令 (新的起始字符)。

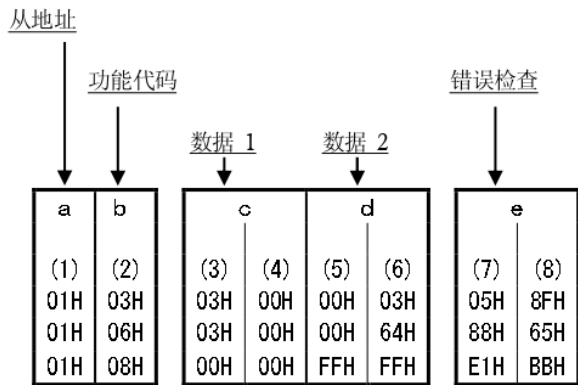
6.2. 通讯格式

1) 信息比较

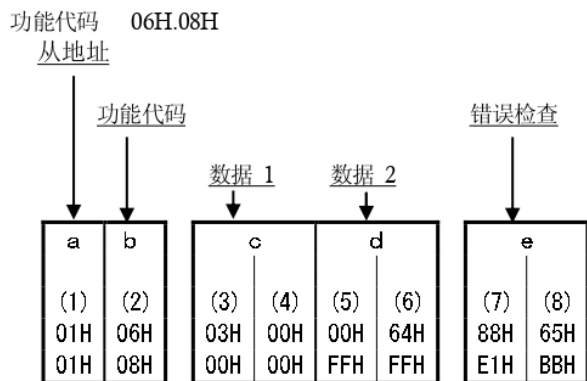
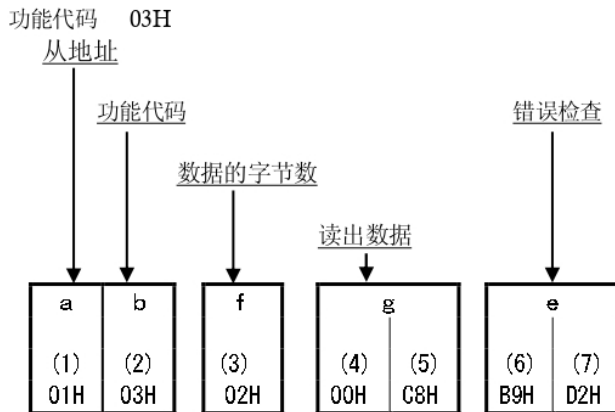
RTU 和 ASCII 方式的 MODBUS 信息比较如下：
所有信息按十六进制处理。



2) 通讯命令格式 (MODBUS: 按 RTU 介绍, 因为 RTU 是基础) - 主机的信息长度固定, 与功能代码无关。



3) 通讯应答格式 (MODBUS: 按 RTU 介绍, 因为 RTU 是基础) - 从机的信息长度取决于功能代码。



a: 从地址

- 主机传送的信息被所有连接的从机接收。仅与信息的从地址对应的从机应答信息。
- 在 **MAC3 & MAC5** 中, 可以指定 1~255 (01 H~FFH)为从地址。
注意: 在 **MODBUS** 协议中, 可以指定的地址是 1~247 (01 H~F 7H)

b: 功能代码

- 代码号表示执行的功能。

功能代码	功能
03 H	数据读出
06 H	数据写
08 H	循环测试

c: 数据 1

- 与功能代码相关的数据。

d: 数据 2

- 与功能代码相关的数据。

功能代码	数据 1 内容	数据 2 内容
03 H	数据地址	读出的数量
06 H	数据地址	写入数据
08 H	固定为 0000H	任意数据

e: 错误检查

- **MODBUS** 方式的错误检查系统。
RTU 方式 :CRC-16
ASC II 方式 :LRC
- 详见 **6-3. 错误检查**

f: 数据字节的数量

- 数据读出时的数据字节数量。
- 读出数据以字为单位, 读出字节是字的 2 倍。

读出数据的数量		数据的字节数	
十进制	十六进制	十进制	十六进制
1	01H	2	02H
2	02H	4	04H
3	03H	6	06H
4	04H	8	08H
5	05H	10	0AH
6	06H	12	0CH
7	07H	14	0EH
8	08H	16	10H
9	09H	18	12H
10	0AH	20	14H

g: 读出数据

- 读出插入的数据
- 读出数量, 数据长度是变量并且没有数据中断。
读出数量为: 1 = 2 字节, 3 = 6 字节, 10 = 20 字节。

6.3. 错误检查

在传送边计算错误检查，将计算结果附加到信息的结尾处。

在接收边将接收到的信息进行错误检查计算。

检查计算的错误检查结果是否与接收的错误检查相同。

如果检查结果匹配，输入的信息判为正确，开始接受应答工作。

如果不同，判断数据异常，从机没有应答。

(1) CRC-16

CRC-16 是 2 字节 (16 位) 错误检查代码。

按下述步骤计算从地址开始到数据结束的 CRC-16。

1. 用FFFFH初始化 CRC 寄存器。
2. 用 CRC 寄存器与信息的一个 1 字节异或。
计算结果写入 CRC 寄存器。
3. 右移 CRC 寄存器 1 位。
4. 如果移出位是 1，CRC 寄存器与 A001H 异或。
计算结果写入 CRC 寄存器。
5. 重复 3. 和 4. 直到移位8次。
6. CRC 寄存器与信息的下一个字节异或。
计算结果写入 CRC 寄存器。
7. 对所有数据重复执行3.~ 6. 。
8. 计算直到数据字节结束。将计算的 CRC 寄存器值按低位和高位排列分配到信息的末尾。

(2) LRC

按下述步骤计算从地址开始到数据结束的LRC。

(注意: 用RTU二进制执行LRC, 结果转换成 ASC II 代码)

1. 从数据头(从地址)到数据结束执行累加。
当计算结果超过 FFH 时, 超过 100H 的值被忽略。(153H 处理为 53H)
2. 累加结果按位求反, 结果加1。
3. 上述方法产生 LRC 代码。
4. LRC 代码分配到信息的结尾, 变为 ASC II 代码字符。

6-4. 数据读出 (功能代码 03H) 详述

当用个人计算机或 PLC 读出各种数据时使用功能代码 03H。

(1) 数据读出格式

- 数据读出时的格式如下:

a	b	c		d		e		ASCII 方式的检查代码 (7), (8) 部分如下 LRC:F5H
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
01H	03H	04H	00H	00H	03H	04H	FBH	

- a: 从地址
- b: 数据读出功能代码
- c: 读出数据首地址
- d: 从首地址读出的数据数量
 - * 可以读出的数据数量是 1~10。
 - 因此, 二进制代码是 0001H~000AH。如果数据数量不是上述范围, 返回错误代码。

e: 错误检查

- 上述命令如下:
 - 读出数据地址 = 0400H (十六进制)
 - 读出数据数量 = 0003H (十六进制)
 - 数据读出从数据地址 0400H 开始。

(2) 读出数据时的正常应答格式

- 对于功能代码 03H 的读出数据格式如下:

a	b	f	g						e		ASCII 方式的错误检查 (10), (11) 部分如下 LRC:42H
(1)	(2)	(3)	0400H	0401H	0402H				(10)	(11)	
01H	03H	06H	00H	1EH	00H	78H	00H	1EH	89H	66H	

- a: 从地址
- b: 功能代码
- f: 读出数据的字节数量
 - * 3 个数据读出, 6 个字节读出。因此, 它是 06H。
- g: 读出数据
 1. 与读出数据数量同样的数据从数据首地址开始依次读出。
 2. 数据间不插入任何其他数据。
 3. 1 个数据由 16 位二进制数据(1 个字)组成, 不包含小数点。
 4. 每个数据隐含特定的小数点位置。
- e: 错误检查

读出数据首地址 (0400H)	读出数据数量 (0003H : 3)	数据地址		数据	
		16 位 (1 字)	十六进制	十六进制	十进制
→	1 2 3	0400	001E	30	
		0401	0078	120	
		0402	001E	30	

(3) 数据读出时的异常应答格式

a	b	h	e		ASCII 方式时的错误检查 (4), (5) 部分如下 LRC: 79H
(1)	(2)	(3)	(4)	(5)	
01H	83H	03H	01H	31H	

- a: 从地址
- b: 功能代码
 - * 出错时, 显示为 接收的功能代码+80H。它表示异常应答。
- h: 错误代码
 - * 关于错误代码详见 6-8. 错误信息
- e: 错误检查

6-5. 数据写入 (功能代码 06H) 详述

从个人计算机、PLC 等写入 (改变) 各种数据时使用功能代码 06H。

(1) 数据写入格式

- 数据写入时格式如下:

a	b	c		d		e		ASC II 方式时的错误检查 (7), (8) 部分如下 LRC: 92H
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
01H	06H	03H	00H	00H	64H	88H	65H	

- a: 从地址
- b: 数据写入功能代码
- c: 写入数据地址
- d: 写入数据
 1. 数据由 16 位二进制数据组成 (1 个字), 不包括小数点。
 2. 每个数据有隐含的小数点位置。
- e: 错误检查

- 上述命令如下:

写入数据首地址 = 0300H (十六进制)
 写入数据 = 0064H (十六进制)
 = 100 (十进制)

指定写入数据地址, 0300H (100:10 十进制)。

地址 (0300H) 写入数据 (0064H)	数据地址	数据	
	16 位 (1 个字)	16 位 (1 个字)	
	十六进制	十六进制	十进制
	0300	0064	100
	0301	0000	0
	0302	0000	0

(2) 写入数据时的正常应答格式

- 对于功能代码 06H 的正常应答格式如下:

a	b	c		d		e		ASC II 方式时的错误检查 (7), (8) 部分如下 LRC: 92H
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
01H	06H	03H	00H	00H	64H	88H	65H	

* 应答格式与主机写入的信息是一样的。

(3) 数据写入时的异常应答

a	b	h	e		ASC II 方式时的错误检查 (4), (5) 部分如下 LRC: 77H
(1)	(2)	(3)	(4)	(5)	
01H	86H	02H	C3H	A1H	

- a: 从地址
- b: 功能代码
 - * 出错时, 表示为接收功能代码 + 80H。他表示异常应答。
- h: 错误代码
 - *关于错误代码详见 6-8. 错误信息
- e: 错误检查

6-6. 循环测试 (功能代码 08H) 详述

功能代码 08H 从机返回从主机发送的信息。
用于在主机和从机之间的通讯诊断。

(1) 循环格式

- 循环测试时的格式如下:

a	b	c		d		e		ASCII 方式时的错误检查 (7),(8) 部分如下 LRC:F9H
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
01H	08H	00H	00H	FFH	FFH	E1H	BBH	

- a: 从地址
- b: 功能代码
- c: 测试代码
 - * 固定为 0000H
- d: 任意数据
 - * 任意 16 位数据: 0000H~FFFFH
- e: 错误检查

(2) 循环正常应答格式

- 对功能代码 08H 的正常应答格式如下:

a	b	c		d		e		ASCII 方式时的错误检查 (7),(8) 部分如下 LRC:F9H
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
01H	08H	00H	00H	FFH	FFH	E1H	BBH	

*应答格式与主机写入的信息是一样的。

(3) 循环时的异常应答格式

a	b	h	e		ASCII 方式时的错误检查 (4),(5) 部分如下 LRC:75H
(1)	(2)	(3)	(4)	(5)	
01H	88H	02H	C7H	C1H	

- a: 从地址
- b: 功能代码
 - * 错误时, 表示为接收代码 +80H。表示异常应答。
- h: 错误代码
 - *关于错误代码详见 6-8. 错误信息
- e: 错误检查

6-7. 无应答的条件

当下述异常被识别出后, 从机没有应答。

- 当发生硬件错误时 (溢出、帧和奇偶错误)
- 当从地址不同时
- 当信息的数据间隔过长时
(RTU: 等效时间大于 28 位 ASCII: 大于 1 秒)
- 当 CRC-16 或 LRC 不同时
- 当主机的信息不标准时(信息过长等)

6-8. 错误信息详述

当检测到非响应条件以外的错误时，应答相应错误类型的错误代码。

(1) 异常应答格式

a	b	h	e	ASCII 方式时的错误检查 (4), (5) 部分如下 LRC:79H
(1)	(2)	(3)	(4) (5)	
01H	83H	03H	01H 31H	

- a: 从地址
- b: 功能代码
错误时，表示为接收功能代码 +80H。表示异常应答。
- h: 错误代码
* 见下表
- e: 错误检查

错误检查	错误内容
0 1 H	功能代码错误 - 当接收到的功能代码非正常时 (不是对应的 3 种之一, < 03H, 06H, 08H>)
0 2 H	地址错误 - 当写入只读地址时 - 当读出只写地址时 - 当循环测试时的测试代码不是 0000H 时 - 当指定不存在的读出或写入的地址时 (包括没有安装的选件)
0 3 H	数据错误 - 当写入数据超出写入数据的范围时 (例如, 当写入非 0 和 1 用于 AUTO/MANU 切换等) - 当写入值已经被其他地址的值填充时, 这种情况仅适用于排异性设置 (DI 适合这种情况) - 当读出数据的数量与可以读出的数量不一致时(对于 MAC3/MAC50, 允许读出数量为 1~10) 当读出数量是 0 或超过 11 时, 产生错误应答代码 - 在不允许改变的情况下再写入参数时 (例如: 在用按键操作改变时, 窗口无显示或不能改变)

(2) 错误代码的优先级

错误代码值越小优先级越高。当多个错误代码同时产生时，返回优先级高的错误代码。

例如: 即使同时存在数据错误和地址错误，当检测到功能代码错误时返回 01H。

6.9. 通讯数据地址详述

(1) 数据地址

- 对于数据地址, 用十六进制数表示二进制数 (16 位)。

(2) 关于读出 (读)/写入 (写)

- R/W 表示数据可以读出和写入。
- R 表示只读出数据。
- W 表示只写入数据。
- 当只写数据地址指定为数据读出时 (功能代码 03H),
- 当只读数据地址指定为数据写入时(功能代码 06H), 它产生地址错误并产生错误代码 02H。

(3) 数据地址和数据的数量

- 当未定义的数据地址被指定为首地址时, 产生地址错误并产生错误代码 02H。
- 当增加的数据地址超出数据地址表时, 总是返回数据 0000 H。

(4) 数据

- 因为每个数据没有小数点 (16 位数据), 需要检查数据类型和小数点。
(参见使用手册)
- 在数据单位为数字时, 量程决定了小数点位置。
- 所有数据按二进制数字处理 (16 位数字: -32768 ~ 32767)。

例如: 表示带小数点数据的方法

	十六进制
20.0	→ 200 → 00C8
100.00	→ 10000 → 2710
-40.00	→ -4000 → F060

表达 16 位数据的方法

数据	
十进制	十六进制
0	0000
1	0001
~	~
32767	7FFF
-32768	8000
-32767	8001
~	~
-2	FFFE
-1	FFFF

(5) 与选件有关的参数

- 当指定没有安装选件的参数地址时, 产生读命令 (R) 和写命令 (W)错误。返回错误代码 02H。

(6) 因为工作特性或设置规格在工作窗口中不显示的参数

- 因为工作特性或设置规格在工作窗口中不显示的参数, 可以用通讯读出。
然而, 写入时产生数据错误。返回错误代码 03H。

7. 通讯数据地址表

数据地址 (Hex)	设置范围	R/W
0040	系列代码 1 "M","A" 4DH,41H	R
0041	系列代码 2 "C","A" 43H,41H	R
0042	系列代码 3 外形尺寸 "A", "0" 41H,30H (MAC10 只有"A0")	R
0043	系列代码 4 输入类型 + 控制输出 1	R

系列 4 详细数据代码

输入规格	输出规格	地址	
		0043H	
M		"M" 4D H	
I		"I" 49H	
	C		"C" 43 H
	S		"S" 53H
	I		"I" 49H

0044	软件版本代码 1	R
0045	软件版本代码 2	R

- 软件版本用 4 位数字表示。小数点放在数据地址 0044 和 0045 之间。

例如：版本 1.00 地址 H L H L
 0044 "0","1" 30H, 31H
 0045 "0","0" 30H, 30H

0046	选件代码 事件输出 + 通讯	R
------	-----------------------	---

选件详细数据代码。

事件	通讯	地址	
		0046H	
N	N	"N" 4E H	"N" 4E H
1	R	"1" 31H	"R" 52H
2		"2" 32H	

0100	PV 测量值 在量程范围内 HHHH,CJHH,b---:7FFFH LLLL,CJLL:8000H	R
0101	执行 SV 值 在 SV 限幅范围内	R
0102	控制输出 1 0.0~100.0	R

0104	工作状态 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 0 0 0 0 0 0 AT/W 0 0 0 0 0 STBY MAN AT * AT/W: AT 待机时 ON STBY: 脱机(复位)时 ON MAN: 手动时 ON AT: AT 执行时 ON	R
数据地址 (Hex)	设置范围	R/W
0105	事件输出 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 EV2 EV1 * EV 2: EV 2 LED 亮时 ON EV 1: EV 1 LED 亮时 ON	R
0106	执行 FIX SV No,1-4	R
010D	自锁状态 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 EV2 EV1 * 在自锁操作状态中,自锁时相应事件位变为 ON	R
010E	继电器 ON/OFF 状态 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 EV2 EV1 * 当事件继电器接点吸合时, 相应位变为 ON	R
0110	事件定时器 1 运行时间 0~600 -1: 表示 END	R
0112	事件定时器 2 运行时间 0~600 -1: 表示 END	R
0180	固定 FIX SV No, 1-4	W
0182	控制输出 1 手动设置范围 0.0~100.0 (仅在手动时)	W
0184	AT 执行 OFF: 0 ON: 1	W
0185	AUTO/MAN 切换 AUTO: 0 MAN: 1	W
0186	RUN(RST)/STBY 切换 RUN: 0 STBY(RST):1	W
0198	自锁释放 无自锁释放: 0 EV 1 释放: 1 EV 2 释放: 2 ALL 释放: 4	W

0300	FIX 定值方式	SV 1	在 SV 限幅值内	R/W
0301	FIX 定值方式	SV 2	在 SV 限幅值内	R/W
0302	FIX 定值方式	SV 3	在 SV 限幅值内	R/W
0303	FIX 定值方式	SV 4	在 SV 限幅值内	R/W

030A	SV 下限值	在量程内	(输入刻度下限~SV上限 -1)	R/W
030B	SV 上限值	在量程内	(SV 下限值 +1 ~输入刻度上限)	R/W

0400	OUT1-PID	比例带	OFF:0 0.1~999.9	R/W
0401		积分时间	OFF:0 1~6000	R/W
0402		微分时间	OFF:0 1~3600	R/W
0403		手动积分	-50.0~50.0	R/W
0404		位式动作灵敏度(下限)	1~999	R/W
0405		输出限幅下限	0.0~99.9	R/W
0406		输出限幅上限	0.1~100.0	R/W
0407		位式动作灵敏度(上限)	1~999	R/W

数据地址 (Hex)	设置范围		R/W
0500	事件工作方式	见事件代码表	R/W
0501	事件报警值	见事件代码表 * 在 SHIMAX 标准协议时 如果事件方式为 NON, So, Run, Stp, P_E, End, Hold, Prog, 此参数无效。 范围: -1999~9999	R/W
0502	报警回差	1~ 999	R/W
0503	报警待机方式	OFF: 0, 1 ~ 2	R/W
0505	事件自锁 / 输出特性	D15-8 D7-0 自锁 输出特性 * 事件自锁的 ON/OFF 在高 8 位, 输出特性的 NO/NC 在低 8 位 自锁 OFF: 0 ON: 1 输出特性 NO: 0 NC: 1	R/W
0506	报警ON延时时间	OFF: 0, 1 ~ 8000	R/W
0507	报警 OFF 延时时间	OFF: 0, 1 ~ 8000	R/W

0508	事件工作方式	见事件代码表	R/W
0509	事件报警值	见事件代码表 * 在 SHIMAX 标准协议时 如果事件方式为 NON, So, Run, Stp, P_E, End, Hold, Prog, 此参数无效。 范围: -1999~9999	R/W
050A	报警回差	1~ 999	R/W
050B	报警待机方式	OFF: 0, 1 ~ 2	R/W
050D	事件自锁 / 输出特性	D15-8 D7-0 自锁 输出特性 * 事件自锁的 ON/OFF 在高 8 位, 输出特性的 NO/NC 在低 8 位 自锁 OFF: 0 ON: 1 输出特性 NO: 0 NC: 1	R/W
050E	报警ON延时时间	OFF: 0, 1 ~ 8000	R/W
050F	报警 OFF 延时时间	OFF: 0, 1 ~ 8000	R/W

05B0	通讯存储方式	RAM:0 MIX:1 ROM:2	R/W
0600	控制输出 1 的输出特性	RA:0 DA:1	R/W
0601	控制输出 1 的比例周期	0.5~120.0 (分辨率 0.5)	R/W
060A	控制输出1 软启动	OFF:0 0.5~120.0 (分辨率0.5)	R/W
0611	按键锁	OFF:0 1~3、5	R/W
0612	上电执行方式	0: EEP 1: STBY 2: RUN	R/W
0700	PV 增益修正	-500~500	R/W
0701	PV 偏移	-500~500	R/W
0702	PV 滤波	0~100	R/W
0704	输入温度单位	°C:0	R
0705	量程	8-1.见量程代码表	R/W
0707	小数点位置	****:0 ***.:1 **.:2 *.:3	R/W
0708	输入刻度下限	-1999~9989	R/W
0709	输入刻度上限	输入刻度下限 +10~9999	R/W
070F	热电偶断线显示	H:0 L:1	R/W
0B80	事件1 延时方式	0: 延时 1: 定时器1 2: 定时器2	R/W
0B81	事件1 定时器ON时间	1~600	R/W
0B82	事件1 定时器OFF时间	0~600	R/W
0B83	事件1 定时器时间单位	0: 秒 1: 分	R/W
0B88	事件2 延时方式	0: 延时 1: 定时器1 2: 定时器2	R/W
0B89	事件2 定时器ON时间	1~600	R/W
0B8A	事件2 定时器OFF时间	0~600	R/W
0B8B	事件2 定时器时间单位	0: 秒 1: 分	R/W

8. 补充说明

8-1. 量程代码表

输入	代码	字符	量程		
			范围	单位	
自由输入	热电偶	0 1	K 1	0 ~ 1300	°C
		0 2	K 2	-50.0 ~ 999.9	
		0 3	J 1	0 ~ 600	
		0 4	J 2	0.0 ~ 600.0	
	铂电阻	0 5	P 1	-100.0 ~ 200.0	
		0 6	P 2	-100 ~ 200	
		0 7	P 3	-199.9 ~ 300.0	
		0 8	P 4	-200 ~ 300	
电流	mV	0 9	M 1	0 ~ 50mV	显示上、下限可设 下列参照
	mA	1 0	MA 1	4 ~ 20mA	
		1 1	MA 2	0 ~ 20mA	

mV, mA 输入的的测量范围, 显示上、下限能在下列任意设定范围内使用

通过刻度功能, 量程可按下述范围设置:

刻度范围: -1999 ~ 9999 数字

刻度差: 10 ~ 10000 数字

刻度下限 < 刻度上限

8-2. 事件代码表

	报警类型	代码	初值	设置范围
<i>non</i>	无	0	-----	-----
<i>HR</i>	绝对值上限	1	量程上限	在量程内
<i>LR</i>	绝对值下限	2	量程下限	在量程内
<i>So</i>	超量程	3	超量程时连续输出	
<i>Hd</i>	上偏差	4	2000 数字	-1999 ~ 2000 数字
<i>Ld</i>	下偏差	5	-1999 数字	-1999 ~ 2000 数字
<i>id</i>	偏差内	6	0 数字	0 ~ 2000 数字
<i>od</i>	偏差外	7	2000 数字	0 ~ 2000 数字
<i>run</i>	RUN 信号	8	RUN 执行时连续输出	

9. ASCII 代码表

	b 7 ~ b 5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b 4 ~ b 1		0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	TC7 (DLE)	SP	0	@	P	`	p
0 0 0 1	1	TC1 (SOH)	DC1	!	1	A	Q	a	q
0 0 1 0	2	TC2 (STX)	DC2	"	2	B	R	b	r
0 0 1 1	3	TC3 (ETX)	DC3	#	3	C	S	c	s
0 1 0 0	4	TC4 (EOT)	DC4	\$	4	D	T	d	t
0 1 0 1	5	TC5 (ENQ)	TC8 (NAK)	%	5	E	U	e	u
0 1 1 0	6	TC6 (ACK)	TC9 (SYN)	&	6	F	V	f	v
0 1 1 1	7	BEL	TC10 (ETB)	'	7	G	W	g	w
1 0 0 0	8	FE0 (BS)	CAN	(8	H	X	h	x
1 0 0 1	9	FE1 (HT)	EM)	9	I	Y	i	y
1 0 1 0	A	FE2 (LF)	SUB	*	:	J	Z	j	z
1 0 1 1	B	FE3 (VT)	ESC	+	;	K	[k	[
1 1 0 0	C	FE4 (FF)	IS4 (FS)	,	<	L	\	l	
1 1 0 1	D	FE5 (GR)	IS3 (GS)	-	=	M]	m]
1 1 1 0	E	SO	IS2 (RS)	.	>	N	^	n	~
1 1 1 1	F	SI	IS1 (US)	/	?	O	_	o	DEL

本手册内容变化时恕不另行通知。

SHIMAX CO., LTD.

URL: <http://www.shimax.co.jp>

本社・工場
TEL 0187-86-3400 FAX 0187-62-6402

〒014-0102

秋田県大仙市四ノ谷字下新谷地 190

PRINTED IN JAPAN