

让世界信赖 伯朗特机器人

Let the world trust **BORUNTE** Robot

BORUNTE 工艺说明书

BORUNTE Process Instruction

伯朗特 Modbus TCP 通信协议说明书

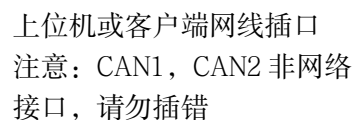
BRT-C-RC-HC1-TI1-V1.4_MODBUS_CN_251027

— www.borunte.com —

目录

1	通讯设置	1
1.1	网口插接	1
1.2	通讯设置	1
1.3	通讯测试	2
2	简述	2
2.1	示例内的数据均为 16 进制；	2
2.1.1	数据格式：	2
2.1.2	请求 APU 示例	3
3	地址定义以及操作要求	3
3.1	读寄存器操作（0x03）	3
3.1.1	读取版本号长度	3
3.1.2	读取版本号	3
3.1.3	读计数器列表	4
3.1.4	读取计数器信息	5
3.1.5	读取当前模式	7
3.1.6	IO 板操作	7
	1、请求读取 IO 板输入状态:	8
	2、请求读取 IO 板输出状态:	8
3.1.7	轴数量读取	9
3.1.8	轴位置	9
3.1.9	世界坐标位置	10
3.1.10	读取当前报警号	11
3.1.11	周期	11
3.1.12	主机地址	12
3.1.13	读取当前扭矩	12
3.1.14	读取轴当前速度	13
3.1.15	移动状态	13
3.2	写单寄存器（0x06）	14
3.2.1	命令	14
3.2.2	修改全局速度	14
3.3	写多寄存器（0x10）	15
3.3.1	写 IO 板整板输出	15
3.3.2	修改单个输出点状态	15
3.3.3	修改计数器	16
3.3.4	修改地址参数	17
3.3.5	发送点位数据	17
3.4	读写位（0x01，0x05）	18
4	地址表定义 功能码：0x03 0x06 0x10	20
5	功能码 0x01 0x05 地址表定义	28

1.1 网口插接



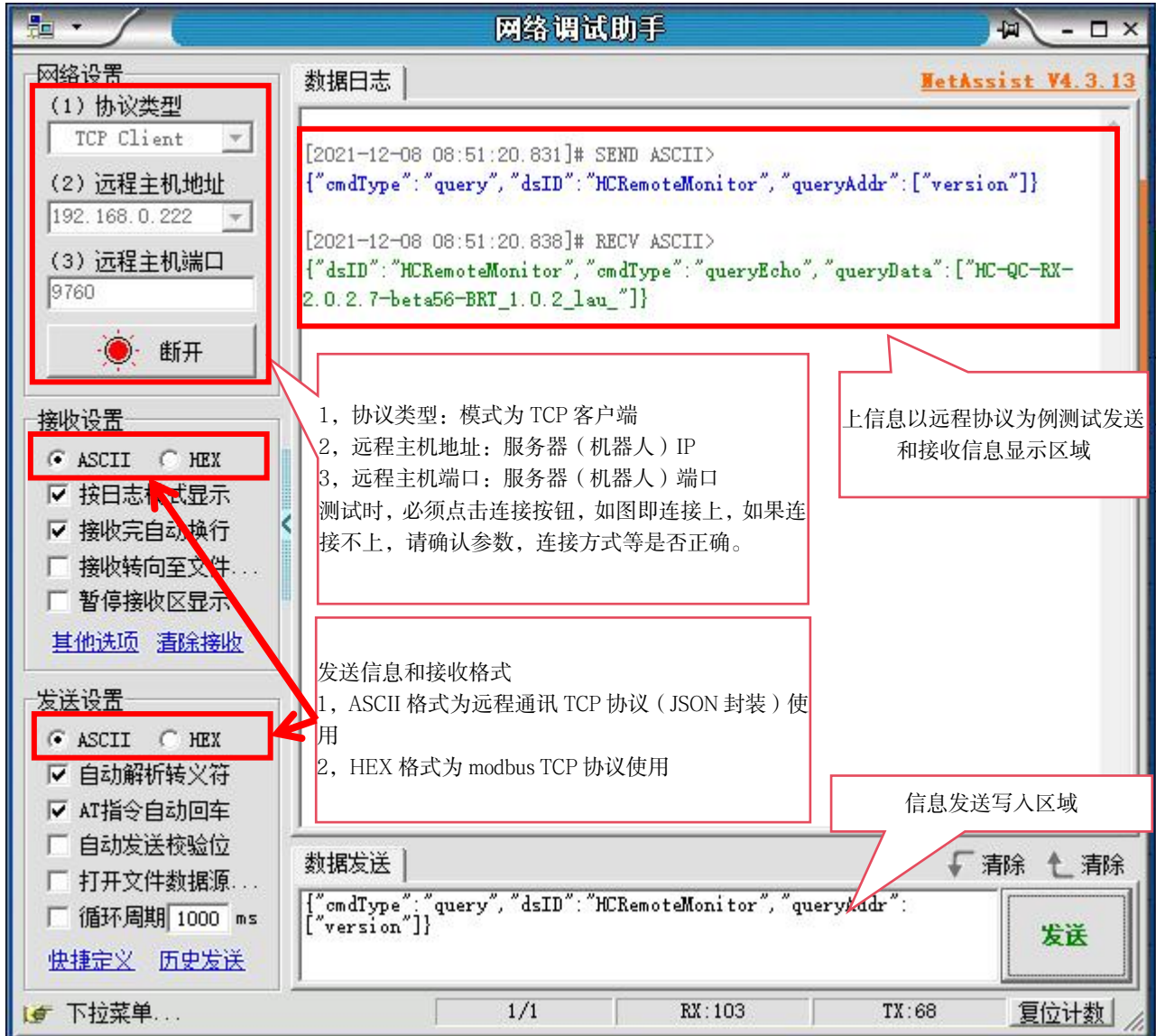
停止档->产品设定->通讯配置->主机网络设定

设定网络段必须一致，分配 ID 号必须是网络内唯一，如果跨网段通讯，请使用路由器处理信息连接

说明：如果没特殊要求，一般通信模式设定为服务器！！！！

1.3 通讯测试

请使用网络调试助手等调试软件验证测试设置是否正确，下为机器人设置为服务器时为例：



说明：按以上方式设定，使网络协议等设定和机器人设定对应，并连接上后，按协议通信，如机器人远程 TCP 协议信息或 modbus TCP 格式数据发送和接收正确即，可确认通讯正常；

如果不回复，请确认：

- 1，发送信息格式是否正确；
- 2，信息内容是否正确；
- 3，发送指令是否与版本兼容等等确认。

2 简述

使用 modbusTCP 协议，主机作为 modbus 从站；

2.1 示例内的数据均为 16 进制；

2.1.1 数据格式：

- (1) 以字节为单位，高字节在前，低字节在后。
- (2) 16bit 数据：占用一个寄存器，在传输时高 8 位在前，低 8 位在后；
- (3) 32bit 数据：占用 2 个寄存器，高 16bit 数据位于低地址中，低 16bit 数据位于高地址中；

地址	值
Addr0	bit31~bit16
Addr1	bit15~bit0

- (4) 对于 64bit 数据：占用 4 个寄存器，最高 16bit 数据位于低地址，最低 16bit 数据位于最高地址；

地址	值
Addr0	bit63~bit48
Addr1	bit47~bit32
Addr2	bit31~bit16
Addr3	bit15~bit0

2.1.2 请求 APU 示例

说明		大小	实例
MBAP 报文头	事务处理标识符 HL	1	0x15
	事务处理标识符 Lo	1	0x01
	协议标识符	2	0x0000
	长度	2	0x0006
MODBUS 请求	单元标识符	1	0xFF
	功能码	1	0x03
	起始地址	2	0x0005
	寄存器数量	2	0x0001

3 地址定义以及操作要求

3.1 读寄存器操作 (0x03)

3.1.1 读取版本号长度

请求版本号长度:

地址: 0x0000

寄存器数量: 1

示例: 00 00 00 00 00 06 01 03 00 00 00 01

说明		大小	实例
MBAP 报文头	事务处理标识符 Hi	1	00
	事务处理标识符 Lo	1	00
	协议标识符	2	00 00
	长度	2	00 06
	单元标识符	1	01
MODBUS 请求	功能码	1	03
	起始地址	2	00 00
	寄存器数量	2	0x0001

响应版本号长度:

示例: 00 00 00 00 00 05 01 03 02 00 29

解释: 版本号的字节数长度为 0x29

说明		大小	实例
MBAP 报文头	事务处理标识符 Hi	1	00
	事务处理标识符 Lo	1	00
	协议标识符	2	00 00
	长度	2	00 05
MODBUS 请求	单元标识符	1	01
	功能码	1	03
	数据字节数	1	02
	数据	2	00 29

3.1.2 读取版本号

需要先读取版本号数据长度, 再使用该长度读取版本号;

由于 modbus 保持寄存器的位宽为 16, 所以当版本号长度为单数时, 最后一个寄存器的值高 8 位有效, 低 8 位用 0 填充。起始地址固定为 0x01, 读取的寄存器数量通过该方法计算: (版本号字节数 + 1) / 2

请求读取版本号:

(1) 起始地址: 0x00 01

(2) 寄存器数量: (版本号字节数 + 1) / 2

示例: 00 00 00 00 00 06 01 03 00 01 00 15

说明		大小	实例
MBAP 报文头	事务处理标识符 Hi	1	00
	事务处理标识符 Lo	1	00
	协议标识符	2	00 00
	长度	2	00 06
MODBUS 请求	单元标识符	1	01
	功能码	1	03
	起始地址	2	00 01
	寄存器数量	2	0x0015

响应版本号:

示例:

0x00 0x00 0x00 0x00 0x00 0x2c 0x01 0x03 0x2a
0x41 0x4d 0x38 0x2d 0x51 0x43 0x2d 0x52 0x58
0x45 0x2d 0x37 0x2e 0x38 0x2e 0x30 0x32 0x2d
0x62 0x61 0x74 0x65 0x37 0x5f 0x46 0x41 0x4b
0x45 0x5f 0x54 0x47 0x46 0x5f 0x45 0x4e 0x43
0x4f 0x44 0x45 0x52 0x00

解释: 将数据区域每字节转为字符后可得到版本: “AM8-QC-RXE-7.8.02-bate7_FAKE_TGF_ENCODER”

说明		大小	实例
MBAP 报文头	事务处理标识符 Hi	1	00
	事务处理标识符 Lo	1	00
	协议标识符	2	00 00
	长度	2	00 2c
MODBUS 响应	单元标识符	1	01
	功能码	1	03
	数据字节数	1	2a
	数据	2a	0x41 0x4d 0x38 0x2d 0x51 0x43 0x2d 0x52 0x58 0x45 0x2d 0x37 0x2e 0x38 0x2e 0x30 0x32 0x2d 0x62 0x61 0x74 0x65 0x37 0x5f 0x46 0x41 0x4b 0x45 0x5f 0x54 0x47 0x46 0x5f 0x45 0x4e 0x43 0x4f 0x44 0x45 0x52 0x00

3.1.3 读计数器列表

先读取计数器数量, 再根据计数器数量来请求对应的计数器 ID, 计数器数量占用一个寄存器, 每个计数器 ID 占用 2 个寄存器; 由于计数器数量可能与读到的数量不同步, 在读取时若实际计数器数量多于请求的数量则截取请求的数量进行响应, 若实际数量少于请求的数量则将剩余的使用 0xFF 填充。

请求计数器数量:

读取当前已有的计数器数量, 只占一个寄存器, 所以起始地址和寄存器数量固定;

示例:

00 00 00 00 00 06 01 03 00 82 00 01

说明		大小	实例
MBAP 报文头	事务处理标识符 Hi	1	00
	事务处理标识符 Lo	1	00
	协议标识符	2	00 00
	长度	2	00 06

	单元标识符	1	01
MODBUS 请求	功能码	1	03
	起始地址	2	00 82
	寄存器数量	2	0001

响应计数器数量:

示例: 00 00 00 00 00 0x05 0x01 0x03 0x02 0x00 0x02

解释: 读取到的计数器数量为 2;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	0x00 0x02 (计数器数量)

请求当前计数器列表:

地址范围: 0x0083~0x0882, 其中有效地址根据已有计数器数量决定 (0x83+ 寄存器数量 × 2), 起始地址必须时目标 ID 的起始地址, 如 0x0083 为第 0 个计数器的 ID 起始地址, 0x0085 为第 1 个计数器的 ID 起始地址;

读寄存器数量: 由于 1 个计数器 ID 占 2 个寄存器, 低 16 位存于小地址, 所以请求的寄存器数量需 2 的倍数, 如 2, 4, 6;

示例 1: 00 00 00 00 00 06 01 03 00 83 00 04

解释: 请求读取从第 0 个计数器开始的连续 2 个计数器 ID;

说明		大小	实例
MBAP 报文头	事务处理标识符 Hi	1	00
	事务处理标识符 Lo	1	00
	协议标识符	2	00 00
	长度	2	00 06
MODBUS 请求	单元标识符	1	01
	功能码	1	03
	起始地址	2	00 83
	寄存器数量	2	0004

- (1) 示例 2: 00 00 00 00 00 06 01 03 00 85 00 04
- (2) 解释: 请求读取从第 1 个计数器开始的连续 2 个计数器 ID;
- (3) 响应当前计数器列表:
- (4) 示例: 00 00 00 00 00 0b 01 03 08 00 00 00 00 00 00 01
- (5) 解释: 读取到 2 个计数器 ID, 分别为 0 和 1;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0b
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	08
数据	8	00 00 00 00 (第 0 个计数器 ID) 00 00 00 01 (第 1 个计数器 ID)

3.1.4 读取计数器信息

读取计数器需要先写入需要读取的计数器 ID, 请求写入待读计数器 ID:

起始地址: 0x0883

寄存器数量: 2,

字节数：4；

值：已存在计数器列表中的计数器 ID，计数器 ID 的高 16bit 放在前，低 16bit 放在后；

示例： 00 00 00 00 00 0B 01 10 08 83 00 02 04 00 00 00 01

解释：设置待读取的计数器 ID 为 1，设置成功后可使用读功能码读取该计数器的数据；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0b
单元标识符	1	01
MODBUS 请求 功能码	1	10
起始地址	2	08 83
寄存器数量	2	00 02
数据字节数	1	04
数据	4	00 00 00 01

响应写入待读计数器 ID:

示例： 00 00 00 00 00 06 01 10 08 83 00 02

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 响应 功能码	1	10
起始地址	2	08 83
寄存器数量	2	00 02

请求读计数器状态数据:

地址：0x0883

寄存器数量：0x06

示例： 00 00 00 00 00 06 01 03 08 83 00 06

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	08 83
寄存器数量	2	00 06

响应计数器状态数据:

示例： 00 00 00 00 00 0f 01 03 0c 00 00 00 01 00 00 00 0a 00 00 00 05

解释：读取的 ID 为 1 的计数器得到其目标计数值为 0x0a，当前计数值为 0x05；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0f
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	0c
数据	12	00 00 00 01 (ID)00 00 00 0a (目标值) 00 00 00 05 (当前值)

3.1.5 读取当前模式

请求当前运行模式:

地址: 0x0889

寄存器数量: 1

示例: 00 00 00 00 00 06 01 03 08 89 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	08 89
寄存器数量	2	00 01

响应当前运行模式:

示例: 00 00 00 00 00 05 01 03 02 00 03

解释: 当前处于配置模式;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	00 03

3.1.6 IO 板操作

目前系统支持的 IO 板数量最多为 5 块; 请求读取 IO 板数:

(1)起始地址: 0x088a

(2)寄存器数量: 1

(3)示例: 00 00 00 00 00 06 01 03 08 8a 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	08 8a
寄存器数量	2	00 01

响应读取 IO 板数量:

示例: 00 00 00 00 00 05 01 03 02 00 01

解释: 当前配置使用一块 IO 板

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	00 01

1、请求读取 IO 板输入状态:

(1) 起始地址: 0x088b (2187) ~ 0x0894 (2196), 有效地址根据已使用的 IO 板数量决定, 每块 IO 板输入状态占用 2 个寄存器, 必须是板的输入低 16 位开始; 如 0x088b (2187) 为第 0 块板输入信号的起始地址, 0x088d (2189) 为第 1 块 IO 板输入信号的起始地址, 后面的以此类推;

(2) 寄存器数量: 读取的寄存器数需 2 的倍数;

(3) 示例: 00 00 00 00 00 06 01 03 08 8b 00 02

解释: 读取第 0 块板的输入状态;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	08 8b
寄存器数量	2	00 01

(4) 示例 2: 00 00 00 00 00 06 01 03 08 8b 00 04

解释: 读取第 0 和第 1 块板的输入状态;

(5) 示例 3: 00 00 00 00 00 06 01 03 08 8d 00 02

解释: 读取第 1 块板的输入状态;

响应读取 IO 板输入状态:

示例: 00 00 00 00 00 07 01 03 04 00 00 00 00

解释: 读取到第 0 块 IO 板的 32 位输入状态为 00 00 00 00;

数据排列与板 IO 对应关系, 如收到数据: 04 03 02 01, 04 对应的是 X40-X47, 03 对应的是 X30-X37, 02 对应的是 X20-X27, 01 对应的是 X10-X17。

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 07
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	04
数据	4	00 00 00 00

2、请求读取 IO 板输出状态:

(1) 起始地址: 0x8b3 (2227) ~ 0x08bc (2236), 使用时要根据已只用的 IO 板数量进行调整, 有效地址根据已使用的 IO 板决定;

(2) 寄存器数量: 每块板的输入有 32 位, 需要占用 2 个字节, 所以寄存器数量需要 2 的倍数;

(3) 数据排列与板 IO 对应关系, 如收到数据: 04 03 02 01, 04 对应的是 X40-X47, 03 对应的是 X30-X37, 02 对应的是 X20-X27, 01 对应的是 X10-X17。

(4) 示例: 00 00 00 00 00 06 01 03 08 b3 00 02

(5) 解释: 读取第 0 块板的输出状态;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	08 b3
寄存器数量	2	00 02

响应读取 IO 板输出状态:

示例: 00 00 00 00 00 07 01 03 04 00 00 00 00

解释: 读取到第 0 块 IO 板的 32 位输出状态为 00 00 00 00;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 07
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	04
数据	4	00 00 00 00

3.1.7 轴数量读取

请求读取轴数量:

地址: 0X08db(2267)

寄存器数量: 0x01

示例: 00 00 00 00 00 06 01 03 08 db 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	08 db
寄存器数量	2	00 01

响应读取 IO 板输出状态:

示例: 00 00 00 00 00 05 01 03 02 00 06

解释: 读取到当前使用的轴数为 6 轴;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	4	00 06(轴数)

3.1.8 轴位置

目前系统最多支持 8 个轴, 轴的位置数据精度为 3 位小数, 单位 ° (如 1000 即为 1°), 每个位置占用 2 个寄存器; 请求: 读取轴位置

地址: 0x08dc~0x08eb (2283), 实际有效地址根据已使用的轴数确定 (0x08dc+ 轴数 × 2), 起始地址必须是某个轴的起始地址, 如 0x08dc 为第 0 轴的位置起始地址, 0x08de 为第 1 轴的位置起始地址; 寄存器数量: 每个轴位置占用 2 个寄存器, 所以读取的寄存器需要 2 的倍数, 如 2, 4;

示例: 00 00 00 00 00 06 01 03 08 dc 00 02

解释: 读取轴 0 的位置;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01

MODBUS 请求 功能码	1	03
起始地址	2	08 dc
寄存器数量	2	00 02

(1) 示例 2: 00 00 00 00 00 06 01 03 08 dc 00 04

(2) 解释: 读取轴 0 和轴 1 的位置;

(3) 示例 3: 00 00 00 00 00 06 01 03 08 de 00 02

(4) 解释: 读取轴 1 的位置; 响应读取轴位置:

(5) 示例: 00 00 00 00 00 07 01 03 04 00 00 00 00

(6) 解释: 读取轴 0 的位置为 0;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 07
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	04
数据	4	00 00 00 00(轴位置)

3.1.9 世界坐标位置

目前系统共支持 8 个世界坐标位置, 值的精度为 3 位小数, 单位为 mm (如 1000 即为 1mm), 每个值占用 2 个寄存器;

世界坐标轴名称定义: 0:X, 1:Y, 2:Z, 3:U, 4:V, 5:W, 6:M7, 7:M8 请求读取世界坐标位置:

地址: 0x091c~0x092b (2347), 起始地址必须是某个世界坐标的起始地址, 如 0x091c 为世界轴 0 (x) 位置的起始地址, 0x091e 为世界轴 1 (Y) 位置的起始地址;

寄存器数量: 每个轴位置占用 2 个寄存器, 所以读取的寄存器需要 2 的倍数, 如 2, 4, 6;

示例: 00 00 00 00 00 06 01 03 09 1c 00 02

解释: 读取世界轴 0 的位置;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 1c
寄存器数量	2	00 02

(1) 示例 2: 00 00 00 00 00 06 01 03 09 1c 00 04

(2) 解释: 读取轴 0 和轴 1 的世界位置;

(3) 示例 3: 00 00 00 00 00 06 01 03 09 1e 00 02

(4) 解释: 读取轴 1 的世界位置; 响应读取轴世界坐标位置:

(5) 示例: 00 00 00 00 00 07 01 03 04 00 00 00 00

(6) 解释: 读取轴 0 的世界位置为 0;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 07
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	04
数据	4	00 00 00 00(轴位置)

3.1.10 读取当前报警号

请求读报警号:

报警号占用一个寄存器，所以寄存器数量固定为 1，地址为 0x095c;

示例： 00 00 00 00 00 06 01 03 09 5c 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 5c
寄存器数量	2	00 01

响应读报警号:

示例： 00 00 00 00 00 05 01 03 02 03 25

解释：当前的报警号为 0x0325;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	03 25 (报警)

3.1.11 周期

周期时间的单位为 ms，一个周期时间占用 4 个寄存器，所以寄存器数量固定为 4；请求当前周期时间:

起始地址：0X95d(2397);

寄存器数量：4

示例： 00 00 00 00 00 06 01 03 09 5d 00 04

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 5d
寄存器数量	2	00 04

响应当前周期时间:

示例： 00 00 00 00 00 0b 01 03 08 00 00 00 00 00 00 00

解释：当前周期时间为 0;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0b
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	08
数据	8	00 00 00 00 00 00 00 00 (当前周期时间)

请求上模周期时间:

起始地址：0X0961(2401);

寄存器数量：4

示例： 00 00 00 00 00 06 01 03 09 61 00 04

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 61
寄存器数量	2	00 04

响应上模周期时间：

示例： 00 00 00 00 00 0b 01 03 08 00 00 00 00 00 00 00

解释：上模周期时间为 0；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0b
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	08
数据	8	00 00 00 00 00 00 00 00（上周期时间）

3.1.12 主机地址

请求读取主机地址：

地址：0X0965(2405)

寄存器数量：1

示例： ‘00 00 00 00 00 06 01 03 09 65 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 65
寄存器数量	2	00 01

响应读取主机地址：

示例： 00 00 00 00 00 05 01 03 02 00 01 解释：主机地址为 0x01 ；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	00 01（主机地址）

3.1.13 读取当前扭矩

2580 代表一倍转矩，每个轴的扭矩占用 1 个寄存器；请求读取扭矩：

地址范围：0X966(2406)~0x096d(2413)

寄存器数量：与起始地址相加后不超过地址范围即可；

示例： 00 00 00 00 00 06 01 03 09 66 00 01 解释：读取轴 0 的当前扭矩数据；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 66
寄存器数量	2	00 01

响应读取扭矩:

示例: 00 00 00 00 00 05 01 03 02 00 00

解释: 轴 0 当前扭矩为 0;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	00 00 (扭矩)

3.1.14 读取轴当前速度

速度单位 RPM, 每个轴的速度占用 1 个寄存器; 请求读取速度:

地址范围: 0x0986 (2438) ~ 0x098d (2445)

寄存器数量: 与起始地址相加后不超过地址范围即可;

示例: 00 00 00 00 00 06 01 03 09 86 00 01

解释: 读取轴 0 的当前速度数据;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 86
寄存器数量	2	00 01

响应读取速度:

示例: 00 00 00 00 00 05 01 03 02 00 00

解释: 轴 0 当前速度为 0;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	00 00 (速递)

3.1.15 移动状态

请求读移动状态:

地址: 0X09a6(2470)

寄存器数量: 1

移动状态值定义: 移动状态; 0: 停止; 1: 移动;

示例： 00 00 00 00 00 06 01 03 09 a6 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	03
起始地址	2	09 a6
寄存器数量	2	00 01

响应读移动状态:

示例： 00 00 00 00 00 05 01 03 02 00 01

解释：当前处于移动状态；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 响应 功能码	1	03
数据字节数	1	02
数据	2	00 01

3.2 写单寄存器 (0x06)

3.2.1 命令

每个命令占用一个寄存器，向寄存器写入任何有效值（0~0xFFFF）均会执行对应命令；命令操作只支持功能码 0x06。

请求停止当前动作:

地址：4e 20

值：任意

示例： 00 00 00 00 00 06 01 06 4e 20 00 01

解释：向停止当前动作的寄存器写入 1，停止当前动作；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	06
起始地址	2	4e 20
寄存器数量	2	00 01

响应与请求帧相同；其他命令查看地址表；

3.2.2 修改全局速度

全局速度单位%，精度 1 位小数，即有效值为 0~1000 对应 0.0% ~100.0%；该值也可用功能码 0x03 读取；

请求修改全局速度:

地址：0x4ee8

值：0~1000

示例： 00 00 00 00 00 06 01 06 4e e8 00 64

解释：设置全局速度为 0x64（即 10.0%）；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00

事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	06
起始地址	2	4e e8
寄存器数量	2	00 64

响应与请求相同;

3.3 写多寄存器 (0x10)

3.3.1 写 IO 板整板输出

当前系统最多支持 5 块 IO 板, 每块 IO 板的输出有 32 位, 占用 2 个寄存器; 请求 IO 板输出:

功能码: 0x10

地址: 0X08b3(2227)~0x08bc(2236), 起始地址必须是某块 IO 板输出的起始地址, 如使用

0x08b3, 0x08b5 正确, 若使用 0x08b4 则错误;

寄存器数量: 由于每块 IO 板占用 2 个寄存器, 所以需要为 2 的倍数;

示例: 00 00 00 00 00 0B 01 10 08 b3 00 02 04 00 00 00 01

解释: 设置第 0 块 IO 板输出第 0 口;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0b
单元标识符	1	01
MODBUS 响应 功能码	1	10
起始地址	2	08 b3
寄存器数量	2	00 02
数据字节数	1	04
数据	4	00 00 00 01

响应 IO 板输出:

示例: 00 00 00 00 00 06 01 10 08 b3 00 02

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	10
起始地址	2	08 b3
寄存器数量	2	00 02

3.3.2 修改单个输出点状态

用于修改某块 IO 板的某一个点输出状态, 需要设置的参数包括指定 IO 板(从 0 开始), 指定输出点(0~31), 指定状态(0: 断 1: 通); 3 个参数各占一个寄存器;

IO 板值范围: 0~3 普通 IO; 4~6: M 值; 7: EUIO;

输出点范围: 0~31

状态值定义: 0: 断 1: 通

请求修改单个输出点状态:

功能码: 0x10

起始地址: 0x4e58

寄存器数量: 3

示例: 00 00 00 00 00 0d 01 10 4e 58 00 03 06 00 00 00 01 00 01

解释: 设置第 0 块板的第 1 个输出口通;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0d
单元标识符	1	01
MODBUS 响应 功能码	1	10
起始地址	2	4e 58
寄存器数量	2	00 03
数据字节数	1	06
数据	6	00 00 00 01 00 01

响应修改单个输出点状态:

示例: 00 00 00 00 00 06 01 10 4e 58 00 03

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	10
起始地址	2	4e 58
寄存器数量	2	00 03

3.3.3 修改计数器

参数包括: ID, 目标值和当前值, 共 3 个, 每个占用 2 个寄存器, 所以修改计数器固定寄存器数量为 6;

ID: 用于指定要修改的计数器,

目标值: 在主机侧不允许修改, 所以该值无要求;

当前值: 允许修改; 请求修改计数器:

功能码 0x10

起始地址: 4e 52

寄存器数量: 6

示例: 00 00 00 00 00 13 01 10 4e 52 00 06 0c 00 00 00 01 00 00 00 00 00 00 08

解释: 需改 ID 是 1 的计数器当前值为 8;

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 13
单元标识符	1	01
MODBUS 响应 功能码	1	10
起始地址	2	4e 52
寄存器数量	2	00 06
数据字节数	1	0c
数据	12	00 00 00 01 00 00 00 00 00 00 00 08

响应修改计数器:

示例: 00 00 00 00 00 06 01 10 4e 52 00 06

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	10
起始地址	2	4e 52

寄存器数量	2	00 06
-------	---	-------

3.3.4 修改地址参数

主机系统参数表共用 1000 个值，每个值占用 2 个寄存器，高 16bit 位于低地址；这块地址也可使用功能码 0x03 读取；请求修改地址参数：

(1) 地址：0x4f4c (20300 地址存储器 0)~0x56d5 (22229 地址存储器 999)，开放使用地址：558C(21900 地址存储器 800)~

(2) 寄存器数量：必须为 2 的倍数；

(3) 示例 1: 00 00 00 00 00 0b 01 10 4f 4c 00 02 04 00 00 00 64

解释：设置第 0 参数的值为 100；

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0b
单元标识符	1	01
MODBUS 响应 功能码	1	10
起始地址	2	4f 4c
寄存器数量	2	00 02
数据字节数	1	04
数据	4	00 00 00 64

响应修改地址参数：

示例 1: 00 00 00 00 00 06 01 10 4f 4c 00 02

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	10
起始地址	2	4f 4c
寄存器数量	2	00 02

示例 2: 00 00 00 00 00 0b 01 10 4f 4e 00 02 04 00 00 00 64

解释：设置第 1 参数的值为 100；

示例 3: 00 00 00 00 00 0f 01 10 4f 4c 00 04 08 00 00 00 64 00 00 00 C8

解释：设置第 0 参数的值为 100，第 1 参数的值为 200；

3.3.5 发送点位数据

系统内部可以有多组不规则点位数据，通过 ID 来标识，堆叠的 ID 即为数据源 ID，视觉数据源固定为 100。可以通过如下流程来管理数据源里面的数据。

1. 先配置要修改的数据源数据信息，可发送如下指令：

请求：

(1) 地址：0x7532 (30002)~0x7534 (30004)

(2) 寄存器数量：3

(3) 示例 1: 00 00 00 00 00 0b 01 10 75 32 00 03 06 00 64 00 06 00 23

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 0d

单元标识符	1	01
MODBUS 响应 功能码	1	10
起始地址	2	75 32
寄存器数量	2	00 03
数据字节数	1	06
数据	6	00 64 00 06 00 23

解释：设置要修改的数据源 ID 为 100，每个点包含 6 个数据，只使用轴 1,2,6 轴的数据，这是一般平面视觉使用的数据配置；

响应： 00 00 00 00 00 06 01 10 4f 4c 00 02

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	10
起始地址	2	75 32
寄存器数量	2	00 03

2. 发送点位数据，点位少的情况（数据小于 256 个字节的情况）：

- (1) 地址：0x7535 (30005)~0x9C40 (40000)
- (2) 寄存器数量：根据点位个数和格式算，每个轴 1 数据 2 个寄存器
- (3) 示例 1：00 00 00 00 00 3b 01 10 75 35 00 1A 34 00 02 00 01 00 00 00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00 0C

解释：清空原有数据源数据，并且添加 2 个点

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 3b
单元标识符	1	01
MODBUS 响应 功能码	1	10
起始地址	2	75 35
寄存器数量	26	00 1A
数据字节数	1	34
数据	52	00 02 00 01 00 00 00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00 0C

3. 发送点位数据，点位多的情况（单帧数据大于 256 个字节的情况）：

可以把数据拆分出来发送，第 1 帧清空原有数据源（0x7536 设 1），后面的帧不清空（0x7536 设 0），也可以选择一帧发送，这样就是非标准协议，我们系统内部会忽略数据字节数的字段。

3.4 读写位（0x01，0x05）

例：0x05

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 响应 功能码	1	05
起始地址	2	00 00

寄存器数量	2	FF 03 通 或 00 00 断
寄存器写入	0xFF 00 通	0x00 00 断

响应和发送一样：

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 响应 功能码	1	05
起始地址	2	00 00
寄存器数量	2	FF 03 00 00
寄存器写入	0xFF 00 通	0x00 00 断

例：0x01 00 00 00 00 00 06 01 01 00 a0 00 0A

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 请求 功能码	1	01
起始地址	2	00 A0
寄存器数量	2	00 0A 起始 10 个数

响应：00 00 00 00 00 05 01 01 02 FD 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 05
单元标识符	1	01
MODBUS 请求 功能码	1	01
起始地址	2	02
寄存器数量	2	FD 01 数据按展开 8/16/32 位高低位（低位为读取起始位）对应读取 IO 转态。正常读取 1 个位，即一个字节，0 断或 1 通

例：

M010 输出通	00 00 00 00 00 06 01 05 00 A0 FF 00	响应一样
M010 转态读取	00 00 00 00 00 06 01 01 00 A0 00 01	读取 1 个位，确认尾 0 断 1 通
Y010 输出通	00 00 00 00 00 06 01 05 00 00 FF 00	响应一样
Y010 转态读取	00 00 00 00 00 06 01 01 00 00 00 01	读取 1 个位，确认尾 0 断 1 通

4 地址表定义 功能码：0x03 0x06 0x10

地址表定义									
功能码	功能说明	字	10#	16#	简述	R/W	最小	最大	备注
16#03	读版本号长度		0	0	版本号长度	R	1	64	
	读版本号内容		1	1	版本号字符串起始	R	/	/	字符串，最大长度度 128 内容 字节
						R	/	/	
			64	40	版本号字符串起始	R	/	/	
	读当前模号长度		65	41	当前模度长度	R	1	64	主机网口不支持
	读当前模号内容		66	42	当前模号低位	R	/	/	字符串
						R	/	/	
			129	81	当前模号高位	R	/	/	
	读计数器列表个数	1	130	82	计数器个数	R	0	1024	
	读计数器列表内容	2	131	83	第 0 个计数器的 ID 高位	R	/	/	有效地址数量由计数器个数决定；每个 ID 占 32 位；
			132	84	第 0 个计数器的 ID 低位	R	/	/	
		2	133	85	第 1 个计数器的 ID 高位	R	/	/	
			134	86	第 1 个计数器的 ID 低位	R	/	/	
		2				R	/	/	
			2178	882	第 1023 个计数器的 ID 低位	R	/	/	
16#03/ 10	读计数器状态 ID	6	2179	883	待读计数器 ID 高位	W/R	/	/	先写待读取的计数器 ID，再读取该段的数据；
			2180	884	待读计数器 ID 低位	W/R	/	/	读取回数据：ID 占（标号）2 各字，目标值 2 个字，当前值 2 个字。
16#03	读计数器目标值		2181	885	计数器目标值高位	R	/	/	
			2182	886	计数器目标值低位	R	/	/	
16#03	读计数器当前值		2183	887	计数器当前值高位	R	/	/	
			2184	888	计数器当前值低位	R	/	/	
16#03	当前模式	1	2185	889	当前模式	R	/	/	（机器人模式，1 手动，3 停止，2，自动停止，7，自动运行，9 单循环）
16#03	IO 板总数	1	2186	88A	IO 板总数	R	0	5	暂时最多用到 4 块，第五块不用；
16#03	板 0 输入状态	2	2187	88B	板 0 的 16~31 输入状态	R	/	/	一块板占 32 位，读取回数值转换成 2 进制数值，每 4 位（16 进制数）位一块板占用。IO 一一对应，最多可读 10 字（5 块板），超出报错；读取 2 时，为板 0 输入，以 2 的倍数，如 4，板 0，1。6 个时，板 0，1，2，类推，最大 10。
			2188	88C	板 0 的 0~15 输入状态	R	/	/	
	板 1 输入状态	2	2189	88D	板 1 的 16~31 输入状态	R	/	/	
			2190	88E	板 1 的 0~15 输入状态	R	/	/	
	板 2 输入状态	2	2191	88F	板 2 的 16~31 输入状态	R	/	/	
			2192	890	板 2 的 0~15 输入状态	R	/	/	
	板 3 输入状态	2	2193	891	板 3 的 16~31 输入状态	R	/	/	
			2194	892	板 3 的 0~15 输入状态	R	/	/	
	板 4 输入状态	2	2195	893	板 4 的 16~31 输入状态	R	/	/	
			2196	894	板 4 的 0~15 输入状态	R	/	/	
16#03	M 板 0 输入状态	2	2197	895	M 板 0 的 16~31 输入状态	R	/	/	
			2198	896	M 板 0 的 0~15 输入状态	R	/	/	
	M 板 1 输入状态	2	2199	897	M 板 1 的 16~31 输入状态	R	/	/	
			2200	898	M 板 1 的 0~15 输入状态	R	/	/	
	M 板 2 输入状态	2	2201	899	M 板 2 的 16~31 输入状态	R	/	/	
2202			89A	M 板 2 的 0~15 输入状态	R	/	/		

	EU 板 0 输入状态	2	2203	89B	EU 板 0 的 16~31 输入状态	R	/	/	
			2204	89C	EU 板 0 的 0~15 输入状态	R	/	/	
	预留								
16#03/ 10	板 0 输出状态	2	2227	8B3	板 0 的 16~31 输出状态	R/W	/	/	一块板占 32 位，读取回数数值转换成 2 进制数值，每 4 位（16 进制数）位一块板占用。IO 一一对应，最多可读 10 字（5 块板），超出报错；读取 2 时，为板 0 输出，以 2 的倍数，如 4，板 0，1。6 个时，板 0，1，2，类推，最大 10。
			2228	8B4	板 0 的 0~15 输出状态	R/W	/	/	
	板 1 输出状态	2	2229	8B5	板 1 的 16~31 输出状态	R/W	/	/	
			2230	8B6	板 1 的 0~15 输出状态	R/W	/	/	
	板 2 输出状态	2	2231	8B7	板 2 的 16~31 输出状态	R/W	/	/	
			2232	8B8	板 2 的 0~15 输出状态	R/W	/	/	
	板 3 输出状态	2	2233	8B9	板 3 的 16~31 输出状态	R/W	/	/	
			2234	8BA	板 3 的 0~15 输出状态	R/W	/	/	
	板 4 输出状态	2	2235	8BB	板 4 的 16~31 输出状态	R/W	/	/	
			2236	8BC	板 4 的 0~15 输出状态	R/W	/	/	
	M 板 0 输出状态	2	2237	8BD	M 板 0 输出 16~31	R/W			
			2238	8BE	M 板 0 输出 0~15	R/W			
	M 板 1 输出状态	2	2239	8BF	M 板 1 输出 16~31	R/W			
			2240	8C0	M 板 1 输出 0~15	R/W			
	M 板 2 输出状态	2	2241	8C1	M 板 2 输出 16~31	R/W			
			2242	8C2	M 板 2 输出 0~15	R/W			
	EUY 板 0 输出状态	2	2243	8C3	EU 板 0 输出 16~31	R/W			
			2244	8C4	EU 板 0 输出 0~15	R/W			
	预留								
16#03	轴总数		2267	8DB	轴总数	R	0	8	目前最多支持 8 轴
	轴 1 位置	2	2268	8DC	轴 0 (J1) 位置高位	R	/	/	原值为 double，精度 3 位小数，将其放大 1000 倍转为整型；读取时，以 2 的倍数，如 2，J1。4 个时，J1，J2，类推，最大 16。读取为 32 位符号数，转换需转换除 1000
			2269	8DD	轴 0 (J1) 位置低位	R	/	/	
	轴 2 位置	2	2270	8DE	轴 1 (J2) 位置高位	R	/	/	
			2271	8DF	轴 1 (J2) 位置低位	R	/	/	
	轴 3 位置	2	2272	8E0	轴 2 (J3) 位置高位	R	/	/	
			2273	8E1	轴 2 (J3) 位置低位	R	/	/	
	轴 4 位置	2	2274	8E2	轴 3 (J4) 位置高位	R	/	/	
			2275	8E3	轴 3 (J4) 位置低位	R	/	/	
	轴 5 位置	2	2276	8E4	轴 4 (J5) 位置高位	R	/	/	
			2277	8E5	轴 4 (J5) 位置低位	R	/	/	
	轴 6 位置	2	2278	8E6	轴 5 (J6) 位置高位	R	/	/	
			2279	8E7	轴 5 (J6) 位置低位	R	/	/	
	轴 7 位置	2	2280	8E8	轴 6 (J7) 位置高位	R	/	/	
			2281	8E9	轴 6 (J7) 位置低位	R	/	/	
	轴 8 位置	2	2282	8EA	轴 7 (J8) 位置高位	R	/	/	
			2283	8EB	轴 7 (J8) 位置低位	R	/	/	
	预留								
16#03	X 轴世界位置	2	2332	91C	世界坐标轴 0(X) 世界高位	R	/	/	原值为 double，精度 3 位小数，将其放大 1000 倍转为整型；读取时，以 2 的倍数，如 2，X。4 个时，X，Y，类推，最大 16。读取为 32 位符号数，转换需转换除 1000
			2333	91D	世界坐标轴 0(X) 位置低位	R	/	/	
	Y 轴世界位置	2	2334	91E	世界坐标轴 1(Y) 位置高位	R	/	/	

	Z 轴世界位置	2	2335	91F	世界坐标轴 1(Y) 位置低位	R	/	/	
			2336	920	世界坐标轴 2(Z) 位置高位	R	/	/	
			2337	921	世界坐标轴 2(Z) 位置低位	R	/	/	
	U 轴世界位置	2	2338	922	世界坐标轴 3(U) 位置高位	R	/	/	
			2339	923	世界坐标轴 3(U) 位置低位	R	/	/	
	V 轴世界位置	2	2340	924	世界坐标轴 4(V) 位置高位	R	/	/	
			2341	925	世界坐标轴 4(V) 位置低位	R	/	/	
	W 轴世界位置	2	2342	926	世界坐标轴 5(W) 位置高位	R	/	/	
			2343	927	世界坐标轴 5(W) 位置低位	R	/	/	
	M7 轴世界位置	2	2344	928	世界坐标轴 6(M7) 位置高位	R	/	/	
			2345	929	世界坐标轴 6(M7) 位置低位	R	/	/	
	M8 轴世界位置	2	2346	92A	世界坐标轴 7(M8) 位置高位	R	/	/	
			2347	92B	世界坐标轴 7(M8) 位置低位	R	/	/	
	预留								
16#03	报警号	1	2396	95C	当前报警号	R	/	/	0 时无报警，大于 0 时报警
16#03	当前周期	4	2397	95D	当前周期 48~63bit	R	/	/	64 位数据；高位在前，低位在后；
			2398	95E	当前周期 32~47bit	R	/	/	
			2399	95F	当前周期 16~31bit	R	/	/	
			2400	960	当前周期 0~15bit R	R	/	/	
16#03	上模周期	4	2401	961	上模周期 48~63bit	R	/	/	64 位数据；高位在前，低位在后；
			2402	962	上模周期 32~47bit	R	/	/	
			2403	963	上模周期 16~31bit	R	/	/	
			2404	964	上模周期 0~15bit R	R	/	/	
16#03	机器名称	1	2405	965	机器名称（主机 ID）	R	/	/	从机 ID，读信息不回复是因站号不一致时，可使用 FF 站号读去回站号
16#03	轴 1 扭矩	1	2406	966	轴 0 当前扭矩	R	/	/	当前轴运行扭矩，华成驱动一体
	轴 2 扭矩	1	2407	967	轴 1 当前扭矩	R	/	/	
	轴 3 扭矩	1	2408	968	轴 2 当前扭矩	R	/	/	
	轴 4 扭矩	1	2409	969	轴 3 当前扭矩	R	/	/	
	轴 5 扭矩	1	2410	96A	轴 4 当前扭矩	R	/	/	
	轴 6 扭矩	1	2411	96B	轴 5 当前扭矩	R	/	/	
	轴 7 扭矩	1	2412	96C	轴 6 当前扭矩	R	/	/	
	轴 8 扭矩	1	2413	96D	轴 7 当前扭矩	R	/	/	
	预留				预留	R	/	/	
	预留		2437	985	预留	R	/	/	
16#03	轴 1 速度	1	2438	986	轴 0 当前速度	R	/	/	当前轴运行速度，华成驱动一体
	轴 2 速度	1	2439	987	轴 1 当前速度	R	/	/	
	轴 3 速度	1	2440	988	轴 2 当前速度	R	/	/	
	轴 4 速度	1	2441	989	轴 3 当前速度	R	/	/	
	轴 5 速度	1	2442	98A	轴 4 当前速度	R	/	/	
	轴 6 速度	1	2443	98B	轴 5 当前速度	R	/	/	
	轴 7 速度	1	2444	98C	轴 6 当前速度	R	/	/	
	轴 8 速度	1	2445	98D	轴 7 当前速度	R	/	/	
	预留				预留	R	/	/	

			2469	9A5	预留	R	/	/	
16#03	移动状态	1	2470	9A6	当前机器移动状态	R	0	1	1 为移动,0 为静止
	原点状态	1	2471	9A7	原点状态	R	0	1	所有轴都设定了原点后, 该状态为 1, 否则为 0
	当前用户		2472	9A8	当前用户长度	R	1	64	主机无该项
			2473	9A9	当前用户低位	R	/	/	主机无该项
			2536	9E8	当前用户高位	R	/	/	主机无该项
	读模号总数		2537	9E9	模号数量	R	/	/	主机无该项
	模号列表		2538	9EA	模号列表	R	/	/	主机无该项
	预留				预留				
16#03	轴 1 最大速度	1	2472	9A8	轴 0 (J1) 最大速度	R	1	64	
	轴 2 最大速度	1	2473	9A9	轴 1 (J2) 最大速度	R	/	/	
	轴 3 最大速度	1	2474	9AA	轴 2 (J3) 最大速度	R			
	轴 4 最大速度	1	2475	9AB	轴 3 (J4) 最大速度	R			
	轴 5 最大速度	1	2476	9AC	轴 4 (J5) 最大速度	R			
	轴 6 最大速度	1	2477	9AD	轴 5 (J6) 最大速度	R			
	轴 7 最大速度	1	2478	9AE	轴 6 (J7) 最大速度	R			
	轴 8 最大速度	1	2479	9AF	轴 7 (J8) 最大速度	R			
	预留								
16#03	轴 1 母线电压	1	2504	9C8	轴 0 (J1) 母线电压	R	/	/	
	轴 2 母线电压	1	2505	9C9	轴 1 (J2) 母线电压	R	/	/	
	轴 3 母线电压	1	2506	9CA	轴 2 (J3) 母线电压	R	/	/	
	轴 4 母线电压	1	2507	9CB	轴 3 (J4) 母线电压	R	/	/	
	轴 5 母线电压	1	2508	9CC	轴 4 (J5) 母线电压	R	/	/	
	轴 6 母线电压	1	2509	9CD	轴 5 (J6) 母线电压	R	/	/	
	轴 7 母线电压	1	2510	9CE	轴 6 (J7) 母线电压	R	/	/	
	轴 8 母线电压	1	2511	9CF	轴 7 (J8) 母线电压	R	/	/	
	预留								
16#03	轴 1 负载率	1	2536	9E8	轴 0 (J1) 负载率	R	/	/	
	轴 2 负载率	1	2537	9E9	轴 1 (J2) 负载率	R	/	/	
	轴 3 负载率	1	2538	9EA	轴 2 (J3) 负载率	R	/	/	
	轴 4 负载率	1	2539	9EB	轴 3 (J4) 负载率	R	/	/	
	轴 5 负载率	1	2540	9EC	轴 4 (J5) 负载率	R	/	/	
	轴 6 负载率	1	2541	9ED	轴 5 (J6) 负载率	R	/	/	
	轴 7 负载率	1	2542	9EE	轴 6 (J7) 负载率	R	/	/	
	轴 8 负载率	1	2543	9EF	轴 7 (J8) 负载率	R	/	/	
	预留								
16#03	轴 1 温度	1	2568	A08	轴 0 (J1) 温度				
	轴 2 温度	1	2569	A09	轴 1 (J2) 温度				
	轴 3 温度	1	2570	A0A	轴 2 (J3) 温度				
	轴 4 温度	1	2571	A0B	轴 3 (J4) 温度				
	轴 5 温度	1	2572	A0C	轴 4 (J5) 温度				
	轴 6 温度	1	2573	A0D	轴 5 (J6) 温度				

	轴 7 温度	1	2574	A0E	轴 6 (J7) 温度				
	轴 8 温度	1	2575	A0F	轴 7 (J8) 温度				
	预留								
16#03	轴 1 位置	2	2600	A28	轴 0 (J1) 位置高位	R	/	/	此地址为浮点数类型，精度 6 位小数；F2.4 版本后支持
	轴 1 位置		2601	A29	轴 0 (J1) 位置低位	R	/	/	
	轴 2 位置	2	2602	A2A	轴 1 (J2) 位置高位	R	/	/	
	轴 2 位置		2603	A2B	轴 1 (J2) 位置低位	R	/	/	
	轴 3 位置	2	2604	A2C	轴 2 (J3) 位置高位	R	/	/	
	轴 3 位置		2605	A2D	轴 2 (J3) 位置低位	R	/	/	
	轴 4 位置	2	2606	A2E	轴 3 (J4) 位置高位	R	/	/	
	轴 4 位置		2607	A2F	轴 3 (J4) 位置低位	R	/	/	
	轴 5 位置	2	2608	A30	轴 4 (J5) 位置高位	R	/	/	
	轴 5 位置		2609	A31	轴 4 (J5) 位置低位	R	/	/	
	轴 6 位置	2	2610	A32	轴 5 (J6) 位置高位	R	/	/	
	轴 6 位置		2611	A33	轴 5 (J6) 位置低位	R	/	/	
	轴 7 位置	2	2612	A34	轴 6 (J7) 位置高位	R	/	/	
	轴 7 位置		2613	A35	轴 6 (J7) 位置低位	R	/	/	
	轴 8 位置	2	2614	A36	轴 7 (J8) 位置高位	R	/	/	
	轴 8 位置		2615	A37	轴 7 (J8) 位置高位	R	/	/	
	预留				预留				
	预留				预留				
16#03	X 轴世界位置	2	2664	A68	世界坐标轴 0(X)世界高位	R	/	/	此地址为浮点数类型，精度 6 位小数；F2.4 版本后支持
	X 轴世界位置		2665	A69	世界坐标轴 0(X)位置低位	R	/	/	
	Y 轴世界位置	2	2666	A6A	世界坐标轴 1(Y)位置高位	R	/	/	
	Y 轴世界位置		2667	A6B	世界坐标轴 1(Y)位置低位	R	/	/	
	Z 轴世界位置	2	2668	A6C	世界坐标轴 2(Z)位置高位	R	/	/	
	Z 轴世界位置		2669	A6D	世界坐标轴 2(Z)位置低位	R	/	/	
	U 轴世界位置	2	2670	A6E	世界坐标轴 3(U)位置高位	R	/	/	
	U 轴世界位置		2671	A6F	世界坐标轴 3(U)位置低位	R	/	/	
	V 轴世界位置	2	2672	A70	世界坐标轴 4(V)位置高位	R	/	/	
	V 轴世界位置		2673	A71	世界坐标轴 4(V)位置低位	R	/	/	
	W 轴世界位置	2	2674	A72	世界坐标轴 5(W)位置高位	R	/	/	
	W 轴世界位置		2675	A73	世界坐标轴 5(W)位置低位	R	/	/	
	M7 轴世界位置	2	2676	A74	世界坐标轴 6(M7)位置高位	R	/	/	
	M7 轴世界位置		2677	A75	世界坐标轴 6(M7)位置低位	R	/	/	
	M8 轴世界位置	2	2678	A76	世界坐标轴 7(M8)位置高位	R	/	/	
	M8 轴世界位置		2679	A77	世界坐标轴 7(M8)位置低位	R	/	/	
16#03	J1 轴伺服版本号	1	2728	AA8	J1 轴伺服版本号 (伺服参数 99)	R			此地址仅 HCRobotHost-HC-QC-RX-7.8.07-F5-beta55 版本及以上可用
	J2 轴伺服版本号	1	2729	AAA	J2 轴伺服版本号 (伺服参数 99)	R			
	J3 轴伺服版本号	1	2730	AAC	J3 轴伺服版本号 (伺服参数 99)	R			
	J4 轴伺服版本号	1	2731	AAE	J4 轴伺服版本号 (伺服参数 99)	R			
	J5 轴伺服版本号	1	2732	AB0	J5 轴伺服版本号 (伺服参数 99)	R			
	J6 轴伺服版本号	1	2733	AB2	J6 轴伺服版本号 (伺服参数 99)	R			

	J7 轴伺服版本号	1	2734	AB4	J7 轴伺服版本号（伺服参数 99）	R			
	J8 轴伺服版本号	1	2735	AB5	J8 轴伺服版本号（伺服参数 99）	R			
16#03	J1 轴伺服子版本号	1	2760	AC8	J1 轴伺服子版本号（伺服参数 98）	R			此地址仅 HCRobotHost-HC-QC-RX-7.8.07-F5-beta55 版本及以上可用
	J2 轴伺服子版本号	1	2761	AC9	J2 轴伺服子版本号（伺服参数 98）	R			
	J3 轴伺服子版本号	1	2762	ACA	J3 轴伺服子版本号（伺服参数 98）	R			
	J4 轴伺服子版本号	1	2763	ACB	J4 轴伺服子版本号（伺服参数 98）	R			
	J5 轴伺服子版本号	1	2764	ACC	J5 轴伺服子版本号（伺服参数 98）	R			
	J6 轴伺服子版本号	1	2765	ACD	J6 轴伺服子版本号（伺服参数 98）	R			
	J7 轴伺服子版本号	1	2766	ACE	J7 轴伺服子版本号（伺服参数 98）	R			
	J8 轴伺服子版本号	1	2767	ACF	J8 轴伺服子版本号（伺服参数 98）	R			
16#03/ 10	计数器 0 当前值	2	3526	DC6	高位	R/W			计数器 0-100 当前值可读写，F2.4 版本后支持
			3527	DC7	低位	R/W			
	计数器 n 当前值	2			高位	R/W			
					低位	R/W			
	计数器 100 当前值	2	3726	E8E	高位	R/W			
			3727	E8F	低位	R/W			
16#03/ 10	计时器 0 当前值	2	3728	E90	高位	R/W			计时器 0-100 当前值可读写，F2.4 版本后支持
			3729	E91	低位	R/W			
	计时器 n 当前值	2			高位	R/W			
					低位	R/W			
	计时器 100 当前值	2	3928	F58	高位	R/W			
			3929	F59	低位	R/W			
16#03/ 10	内部地址	2	4500	1194	allpara[800] 断电保持高位	RW	/	/	此地址用于断电保存，此地址仅 HCRobotHost-HC-QC-RX-7.8.07-F5-beta55 版本及以上可用
			4501	1195	allpara[800] 断电保持低位	RW	/	/	
		2	4502	1196	allpara[801] 断电保持高位	RW	/	/	
			4503	1197	allpara[801] 断电保持低位	RW	/	/	
						RW	/	/	
		2	4696	1258	allpara[898] 断电保持高位	R	/	/	
			4697	1259	allpara[898] 断电保持低位	R	/	/	
16#06	命令：立即停止当前动作	1	20000	4E20	立即停止当前动作	W	/	/	立即停止当前动作（启动从头开始）
	命令：暂停当前动作	1	20001	4E21	暂停当前动作	W	/	/	命令：暂停当前动作（启动从当前步开始）
	命令：进入单循环	1	20002	4E22	进入单循环	W	/	/	命令：进入单循环
	命令：启动按键	1	20003	4E23	startButton: 启动按键	W	/	/	命令：启动按键
	命令：停按键 键	1	20004	4E24	stopButton: 停止按键	W	/	/	命令：停止按键
	命令：清除报警后运行下一条指令	1	20005	4E25	清除报警后运行下一条指令	W	/	/	命令：清除报警后运行下一条指令

	命令：清除报警并继续自动运行行	1	20006	4E26	清除报警并继续自动运行	W	/	/	命令：清除报警并继续自动运行:（自动运行状态下）
16#03/ 10	自动下上使能	2	20302	4F4E	自动下重新上使能	RW	/	/	在自动下断使能重写 2 即重新上使能，模式切换 0: 无,1: 手动模式, 2: 自动模式, 3: 停止模式, 7: 自动运行中, 8: 单步, 9: 单循环
			20303	4F4F		RW	/	/	
16#10	计数器修改	6	20050	4E52	待写计数器 ID 高位	W	/	/	可先读取计数器列表，获取可配置的计数器 ID，此为修改的目标计数器 ID，修改 ID，指向 ID 计数器（注意必须建有）
			20051	4E53	待写计数器 ID 低位	W	/	/	
			20052	4E54	设置计数器目标值高位	W	/	/	主机不能修改目标值，此为目标 ID 计数器目标值，
			20053	4E55	设置计数器目标值低位	W	/	/	
			20054	4E56	设置计数器当前值高位	W	/	/	此为目标当前 ID 计数值，可修改
			20055	4E57	设置计数器当前值低位	W	/	/	
16#03/ 10	输出点控制	3	20056	4E58	设置输出板 ID	W	0	7	（0~3: IO 板，4~6: M 值（手控网口版本暂时不支持），7: EUIO）
			20057	4E59	设置输出点	W	0	31	（0~31）
			20058	4E5A	设置输出状态	W	0	1	（0: OFF，1: ON）
16#10	堆叠修改		20059	4E5B	待修改堆叠 ID	W	/	/	数据宽度。主机网口无该功能
			20060	4E5C	X 间隔低位	W	/	/	
			20061	4E5D	X 间隔高位	W	/	/	
			20062	4E5E	Y 间隔低位	W	/	/	
			20063	4E5F	Y 间隔高位	W	/	/	
			20064	4E60	Z 间隔低位	W	/	/	
			20065	4E61	Z 间隔高位	W	/	/	
			20066	4E62	X 计数低位	W	/	/	
			20067	4E63	X 计数高位	W	/	/	
			20068	4E64	Y 计数低位	W	/	/	
			20069	4E65	Y 计数高位	W	/	/	
			20070	4E66	Z 计数低位	W	/	/	
	预留			0	预留	W	/	/	
16#06	全局速度		20200	4EE8	全局速度	RW	0	1000	32 位精度 1，1000 即 100%速度 10 即 1%速度
16#03/ 10	内部参数表地址	2	20300	4F4C	allpara[0] 高位	RW	/	/	主机功能 1000 个参数共操作，每个值 32 位,占 4 个字节（2 个存储器），起始地址：4F4C，4F4D 对应 0 号存储器，其他顺推。558C(21900)地址对应 800 低位存储器，用户开放地址：800-899
			20301	4F4D	allpara[0] 低位	RW	/	/	
		2	20302	4F4E	allpara[1] 高位	RW	/	/	
			20303	4F4F	allpara[1] 低位	RW	/	/	
		3		0		RW	/	/	
				0		RW	/	/	
		4	21900	558C	allpara[800] 高位	RW	/	/	
			21901	558D	allpara[800] 低位	RW	/	/	
		5	21902	558E	allpara[801] 高位	RW	/	/	
			21903	558F	allpara[801] 低位	RW	/	/	
16#03						RW	/	/	
			22228	56D4	allpara[999] 高位	R	/	/	

			22229	56D5	allpara[999] 低位	R	/	/	
3	拍照命令		22130	5672	拍照命令	R	/	/	主机不支持
	切换模号		22131	5673	切换模号	R	/	/	(“d1”：模号 (主机网口版本暂时不支持))
16#03	机器编号长度		23000	59D8					
	机器编号		23001	59D9	机器人编号，需设定才可读取				机器人编号，需设定才可读取
			23002	59DA					
			23257	5AD9					
16#03	机器规格长度		23300	5B04	机器人规格，需设定才可读取				机器人规格，需设定才可读取
	机器规格		23301	5B05					
			23302	5B06					
			23557	5C05					
16#03	手控版本长度		23900	5D5C	手控器机器人系统程序版本				手控器机器人系统程序版本
	手控版本		23901	5F5D					
			23902	5D5E					
			24004	5DC4					
				0					
				0					
				0					
	待查看的数据源 ID		30000	7530	待查看的数据源 ID	R	/	/	视觉数据源 ID100，其它的为堆叠 ID
	数据源 ID 对应的缓冲区长度		30001	7531	数据源 ID 对应的缓冲区长度	R	/	/	
16#03	待修改的视觉数据源 ID		30002	7532	待修改的视觉数据源 ID	RW	/	/	待修改的视觉数据源 ID100，它的为堆叠 ID
16#03/ 10	待修改的数据源点位格式		30003	7533	待修改的数据源点位格式,6 个数据为一点还是 4 个数据为一个点	RW	/	/	默认为 6
	待修改的数据源点位掩码		30004	7534	待修改的数据源点位格式掩码. 位 0-5 对应轴 1-6	RW	/	/	如果一个点有 6 个轴位置，但是只要 1,2,6 有效，那么掩码可设为 6 轴掩码 0x23，0x3F
	待接收数据点位长度		30005	7535	待接收数据点位长度	RW	/	/	
	清空待修改数据源数据		30006	7536	清空待修改数据源数据	RW	/	/	1: 为清空
16#10	数据源点位数据		30007	7537	轴 1 数据高位	W	/	/	
			30008	7538	轴 1 数据低位	W	/	/	
						W	/	/	
			40000	9C40	数据源点位数据	W	/	/	
			40001	9C41		W	/	/	

5 功能码 0x01 0x05 地址表定义

位地址操作使用				
功能说明	地址 10 进制	16 进制	读 r/写 w	注释
Y010 --> Y047	0 --> 31	0 --> 1F	r/w	Y010 对应 0 地址，Y011 对应 1 地址，其他类推，，， 使用功能码 0x01 读，0x05 写 写 05 功能码 尾 0xFF00 通 0x0000 断 读 01 功能码 回复尾：0x00 断 0x01 通， 注意：读取连续多个 IO 时，需把接收数据字节分解对应 IO 确认定，如数据 05 字节，起始是 0-7 时，对应关系即是：0000 0101 左高位，右低位，IO 对应； 单点读 发送：01 01 00 00 00 01 FD CA 回复：01 01 01 01 90 48 多点读取 发送：01 01 00 00 00 08 3D CC 回复：01 01 01 0F 11 8C
Y110 --> Y147	32 --> 63	20 --> 3F	r/w	
Y210 --> Y247	64 --> 95	40 --> 5F	r/w	
Y310 --> Y347	96 --> 127	60 --> 8F	r/w	
Y410 --> Y447	128 --> 159	80 --> 9F	r/w	
M010 --> M047	160 --> 191	A0 --> BF	r/w	
M110 --> M147	192 --> 223	C0 --> DF	r/w	
M210 --> M247	224 --> 255	E0 --> FF	r/w	
EUY010 --> EUY147	256 --> 287	100 --> 11F	r/w	

使用功能码 0x01 读，同 03 功能码使用，01 位读单位，03 字读单位。

00 00 00 00 00 06 01 01 00 00 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 响应 功能码	1	01
起始地址	2	00 00
数据	2	00 01 (读取个数，多个位时，展开位对应 IO 状态)

使用功能码 0x05 写，同 06 功能码使用，05 位单写，06 字单写

00 00 00 00 00 06 01 01 00 00 00 01

说明	大小	实例
MBAP 报文头 事务处理标识符 Hi	1	00
事务处理标识符 Lo	1	00
协议标识符	2	00 00
长度	2	00 06
单元标识符	1	01
MODBUS 响应 功能码	1	05
起始地址	2	00 02
寄存器数量	2	FF 00 (FF 00 通，00 00 断)

输出 Y，M，EUY 为 8 进制递增对应地址。

说明书手册使用条款声明

- 本公司将不定期对产品进行优化与调整，手册修改恕不另行通知。
- 用户实际产品功能及规格可能与本手册存在差异，请前往伯朗特官网 www.borunte.com 获取最新相关资料。
- 未经本公司书面授权，禁止以任何形式复制、修改、传播本手册内容或用于商业用途。
- 本公司保留对本手册内容的最终解释权及修订权利。

伯朗特机器人股份有限公司

📍 地址：广东省东莞市大朗镇沙步村沙富路 83 号

☎ 电话：400-870-8989

✉ 邮箱：rd316@borunte.com

因改良等原因,规格及外观有所变更时,不另行通知,敬请谅解。因排样需要,某些部件有所增减,请按实际订单为准。

BORUNTE ROBOT CO., LTD.

📍 Address: NO. 83, Shafu Road, Shabu Village, Dalang Town, Dongguan City, Guangdong Province, China

☎ Tel: 0086-400-870-8989

✉ E-mail: rd316@borunte.com

All stated here is subject to change without advance notice. Some parts have been increased or deleted because of the shooting needed, please order as the actual standard.



抖音号



公众号



小程序

— www.borunte.com —