

Modbus 从机命令介绍

修订历史:

版本	日期	修订原因	编制	审核
V1.0	2015-06-09	创建文档	HYW089	HYW061
V2.0	2017-11-03	产品升级	HYW089	HYW070

Modbus 从机命令介绍

系统支持 MODBUS-RTU 从机模式，使用 modbus 协议时必须升级对应的 MODBUS 程序。

用文本编辑器打开 config.txt 配置 485 参数和 modbus 从机地址：

R1=07 ;波特率,07=115200; **R1=02**,4800（最低），**R1=03**,9600。。

R3=5A ;MODBUS 从机设备配置地址

RA=00 ;0x00=8N1 无校验

1、从机 MODBUS 指令表

指令	名称	作用举例
0x03	读变量地址	读变量值
0x04	读寄存器	读寄存器值
0x0F	写寄存器	写寄存器值
0x10	写变量地址	写变量值

03 读变量地址

请求指令的格式为：

从机地址	指令	起始地址高字节和低字节	字长	校验
例：请求指 Tx：5A 03 0000 0002 C920 从 0x0000 地址读 2 个变量数据				

响应指令的格式为：

从机地址	指令	字节个数	数据 1、数据 2...	校验
例：响应为 Rx：5A 03 04 0000 0000 10F6				

04 读寄存器

请求指令的格式为：

模块地址	指令	起始地址高字节和低字节	字长	校验
例：请求指令 Tx: 5A 04 0000 0002 7CE0 读寄存器 00 地址开始的 4 个寄存器				

响应指令的格式为：

模块地址	指令	字节个数	数据 1、数据 2...	校验
例：响应为 Rx: 5A 04 04 71 40 00 00 0BA9 读取版本号、亮度，蜂鸣器控制寄存器。。				

0F 写寄存器

请求指令的格式为：

从机地址	指令	起始寄存器高字节和低字节	线圈个数高字节	线圈个数低字节	字节数	设置数据的高字节和低字节	校验
例：请求指令：5A 0F 0003 0000 02 0001 A820 向寄存器 03 和 04 寄存器里面写入 0001 切换第 1 页面							

响应指令的格式为：

从机地址	指令	起始地址高字节和低字节	线圈个数的 高字节	线圈个数的 低字节	校验
例：响应为 Rx: 5A 0F 0003 0000 A8E0					

所谓的“线圈”就是“位变量 16 位变量”，这里写 0x0000 就行，

寄存器范围是 0x00-0xFF，起始寄存器高字节写入 0x00 就行

如蜂鸣器鸣叫 200ms：

请求指令：5A 0F 0002 0000 01 20 43c0

响应为 Rx: 5A 0F 00 02 00 00 F9 20

如亮度寄存器控制 **1/4** 亮度:

请求指令: 5a 0f 0001 0000 01 **10** 07d4

响应为 Rx: 5A 0F 00 02 00 00 F9 20

10 写变量地址

16 用于设置多个保持寄存器的值

请求指令的格式为:

从机地址	指令	起始地址高字节和址低字节	变量地址个数的高字节的和字节	字节数	数据	校验
例: 请求指令: 5A 10 0000 0002 04 3132 3334 6D9E 向文本变量起始地址 0000 里面写 1234						

响应指令的格式为:

从机地址	指令	起始地址高字节和低字节	变量地址个数的高字节和低字节	校验
例: 响应为 Rx: 5A 10 0000 0002 4CE3				

如向数据变量显示地址 **0001** 里面写入 **10**

请求指令: 5A 10 0001 0001 02 000A A8 B5

响应为 Rx: 5A 10 00 01 00 01 5D 22

使用示例：**03 指令示例：**

上位机配置变量地址 0x0100



名称定义 又半业小

描述指针 (0x) FFFF

变量地址 (0x) 0100

显示颜色 (0x) 041F

编码方式 0x02=GBK

☐ 字符间距不自动调整

文本长度 100

变量值如下：



发送及返回数据如下：

The screenshot shows a software interface with several sections. A red box highlights the '03H指令 (读保持寄存器)' section, which includes fields for '485地址' (10), '寄存器地址' (0x0100), and '寄存器个数' (4), along with a '读取' button. Below this, the '寄存器值' is displayed as 'DCD4 C0F2 BBA8 B2E8'. To the right, there are fields for '485地址' (02), '寄存器地址' (0x1000), and '寄存器值'. Below these is an '任意指令' field and a '发送' button. At the bottom, there are two red boxes: '发送缓冲区' containing '0A 03 01 00 00 04 44 8E' and '接收缓冲区' containing '0A 03 08 DC D4 C0 F2 BB A8 B2 E8 D0 C8'. A '清空数据' button is located between the two buffer boxes.

10: 地址

0100: 变量地址

4: 个数, “茉莉花茶” 占用 4 个变量地址

收到指令后屏返回“茉莉花茶”区域码。

06 指令示例:

变量地址:

The screenshot shows a software interface with a checkbox labeled '不进行页面切换'. Below it, there are fields for '变量地址 (0x)' (0300), '变量类型' (0X00=整数 (字)), '整数位数' (5), and '小数位数' (0).

发送指令：

02H指令（读离散输入寄存器）

485地址	寄存器地址	寄存器个数
1	2001	17

寄存器值

03H指令（读保持寄存器）

485地址	寄存器地址	寄存器个数
10	0x0100	4

寄存器值 DCD4 C0F2 BBA8 B2E8

06H指令（写单个保持寄存器）

485地址	寄存器地址	寄存器值
10	0x0300	0010

发送

10H指令（写多个保持寄存器）

485地址	寄存器地址	寄存器值
02	0x1000	0000

发送

任意指令

发送 带CRC发送

发送缓冲区

清空数据

接收缓冲区

清空数据

0A 06 03 00 00 0A 08 F2

屏幕显示：

