

I0 模块串口 Modbus RTU 通讯协议说明书



www.rockmong.com

上海岩獾科技有限公司
V1.1

目录

I/O 模块串口 Modbus RTU 通讯协议说明书.....	1
一. 术语和定义.....	3
1.1 线圈.....	3
1.2 离散输入.....	3
二. 通讯硬件接口.....	3
三. MODBUS 协议格式.....	3
3.1 概述.....	3
3.2 Modbus RTU 模式的通用通讯帧格式	4
3.3 功能码.....	4
3.4 读线圈（0x01）	4
3.5 写单个线圈（0x05）	5
3.6 写多个线圈（0x0F）	6

一. 术语和定义

1.1 线圈

在 Modbus 通信协议中，“线圈”通常指的是数字输出线圈（Coils）。这些线圈用于控制外部设备，如继电器或开关。每个线圈都对应一个二进制状态，可以是打开（1）或关闭（0）。在 Modbus 通信中，可以使用读取和写入操作来读取线圈的状态或改变它们的状态。

1.2 离散输入

在 Modbus 通信协议中，“离散输入”通常指的是数字输入（Discrete Inputs），也称为离散输入状态。这些离散输入用于表示外部设备的状态，例如传感器的开关状态或其他数字输入状态。每个离散输入都对应一个二进制状态，可以是打开（1）或关闭（0）。

二. 通讯硬件接口

RS232 串口通讯	
波特率	9600 bps
起始位	1
数据位	8
校验位	None
停止位	1

三. MODBUS 协议格式

3.1 概述

由于 Modbus 协议完全开放、应用广泛，而且协议简单、调试手段丰富，在多机通讯的场合很容易提高开发速度，还可以很方便地与市场上已有支持 Modbus 协议的设备连接，实现数据通讯，从而成为一种事实上的工业通讯标准。Modbus 通讯协议有两种传输模式，分为 RTU 模式和 ASCII 模式。本产品接口采用 Modbus RTU 通讯模式。

3.2 Modbus RTU 模式的通用通讯帧格式

起始时间间隔	地址码	功能码	数据区	CRC 校验码	结束时间间隔
T1-T2-T3-T4	1 Byte	1 Byte	n Bytes	2 Bytes	T1-T2-T3-T4

RTU 模式中,信息开始至少需要有 3.5 个字符时间的停顿间隔开始,依据使用的波特率,很容易计算这个停顿的时间(如上图中的 T1-T2-T3-T4)。发送完最后一个字符后,也有一个 3.5 个字符时间的停顿,然后才能发送一个新的信息。

整个信息必须连续发送。如果在发送帧消息器件,出现大于 1.5 个字符时间的停顿时,则接收设备刷新不完整的信息,并假设下一个地址数据。

同样一个信息后,立即发送的一个新消息,(若无 3.5 个字符的停顿)这将会产生一个错误。因为合并信息的 CRC 校验码无效而产生的错误。

3.3 功能码

功能码主要是对离散量输入、线圈、输入寄存器、保持寄存器进行读写。本产品这里只需要用到线圈和离散输入。

名称	功能码	对应本产品实际功能
读线圈	0x01	读输出端的状态
写单个线圈	0x05	写/控制单个输出端的状态
写多个线圈	0x0F	写/控制多个输出端的状态
读离散输入	0x02	读输入端的状态

表 2 功能码

3.4 读线圈 (0x01)

发送

地址码	功能码	数据区		CRC 校验码
		起始地址	数量	
01	01	XX XX	00 XX	XX XX

起始地址: D00(或 P0)的地址为 0, D01(或 P1)的地址为 1, 依此类推...

数量: 要读取线圈的数量。

接收

地址码	功能码	数据区		CRC 校验码
		字节计数	状态	
01	01	XX	XX	XX XX

字节计数, 读取到的线圈状态的字节数。

状态: 要读取线圈的状态。每个 bit 是一个线圈, 相应 bit 为 0 或 1 即是相应线圈 0 或 1。

举例：读取 D00~D07 (或 P0~P7) 共 8 路输出端状态

发送：

地址码	功能码	数据区		CRC 校验码
		起始地址	数量	
01	01	00 00	00 08	3D CC

接收：

地址码	功能码	数据区		CRC 校验码
		字节计数	状态	
01	01	01	00	51 88

3.5 写单个线圈（0x05）

发送：

地址码	功能码	数据区		CRC 校验码
		起始地址	状态	
01	05	00 XX	XX 00	XX XX

起始地址：P0 (或 D00) 的地址为 0，P1 (或 D01) 的地址为 1，依此类推...

状态：要写入线圈的状态。写 00 00 即写 0，写 FF 00 即写 1。不能写入其他数据，否则会返回错误。

接收：

同发送。不同则代表发生错误。

举例：写入 D01（或 P1）为高电平

发送：

地址码	功能码	数据区		CRC 校验码
		起始地址	状态	
01	05	00 01	FF 00	DD FA

接收：

同发送。不同则代表发生错误。

举例：写入 D01（或 P1）为低电平

发送：

地址码	功能码	数据区		CRC 校验码
		起始地址	状态	
01	05	00 01	00 00	9C 0A

接收：

同发送。不同则代表发生错误。

3.6 写多个线圈 (0x0F)

发送:

地址码	功能码	数据区			状态	CRC 校验码
		起始地址	数量	字节计数		
01	0F	00 XX	00 XX	XX	XX XX ...	XX XX

起始地址: D00(或 P0)的地址为 0, D01(或 P1)的地址为 1, 依此类推...

数量: 要控制线圈的数量。

字节计数: 写入线圈状态的字节数

状态: 要写入线圈的状态。每个 bit 写 0 或 1, 多少个线圈写多少个 bit。如果不足一个字节则按一个字节算。

接收:

地址码	功能码	数据区		CRC 校验码
		起始地址	数量	
01	0F	00 XX	00 XX	XX XX

举例: D00~D07 (或 P0~P7) 共 8 路输出端同时输出高电平

发送:

地址码	功能码	数据区			状态	CRC 校验码
		起始地址	数量	字节计数		
01	0F	00 00	00 08	01	FF	BE D5

接收:

地址码	功能码	数据区		CRC 校验码
		起始地址	数量	
01	0F	00 00	00 08	54 0D

举例: D00~D07 (或 P0~P7) 共 8 路输出端同时输出低电平

发送:

地址码	功能码	数据区			状态	CRC 校验码
		起始地址	数量	字节计数		
01	0F	00 00	00 08	01	00	FE 95

接收:

地址码	功能码	数据区		CRC 校验码
		起始地址	数量	
01	0F	00 00	00 08	54 0D

3.7 读离散输入 (0x02)

发送

地址码	功能码	数据区		CRC 校验码
		起始地址	数量	

01	02	XX XX	00 XX	XX XX
----	----	-------	-------	-------

起始地址：DIO(或 P0)的地址为 0，DI1(或 P1)的地址为 1，依此类推...

数量：要读取离散输入的数量。

接收

地址码	功能码	数据区		CRC 校验码
		字节计数	状态	
01	02	XX	XX	XX XX

字节计数，读取到的离散输入状态的字节数。

状态：要读取离散输入的状态。每个 bit 是一个离散输入，相应 bit 为 0 或 1 即是相应离散输入 0 或 1。

举例：读取 DIO~DI7(或 P0~P7) 共 8 路输入端状态

发送：

地址码	功能码	数据区		CRC 校验码
		起始地址	数量	
01	02	00 00	00 08	79 CC

接收：

地址码	功能码	数据区		CRC 校验码
		字节计数	状态	
01	02	01	FF	E1 C8