

# Modbus协议简介

Modbus是工业设备通讯中使用最多，应用最广的一种协议，是Modicon公司（现在的施耐德电气 Schneider Electric）于1979年为使用可编程逻辑控制器（PLC）通信而发表。Modbus已经成为工业领域通信协议的业界标准（De facto）并且现在是工业电子设备之间常用的连接方式。Modbus协议目前存在用于串口、以太网以及其他支持互联网协议的网络的版本。

大多数Modbus设备通信通过串口EIA-485物理层进行。

对于串行连接，存在两个变种，它们在数值数据表示不同和协议细节上略有不同。Modbus RTU是一种紧凑的，采用二进制表示数据的方式。Modbus ASCII是一种人类可读的，冗长的表示方式。这两个变种都使用串行通信（serial communication）方式。RTU格式后续的命令/数据带有循环冗余校验的校验和，而ASCII格式采用纵向冗余校验的校验和。被配置为RTU变种的节点不会和设置为ASCII变种的节点通信，反之亦然。

对于通过TCP/IP（例如以太网）的连接，存在多个Modbus/TCP变种，这种方式不需要校验和计算。

对于所有的这三种通信协议在数据模型和功能调用上都是相同的，只有封装方式是不同的。

## 协议功能简介

| 功能类型  | 功能描述   |
|-------|--|
| 设备地址  | 对应通讯设备设定的1-247的任意一个数字地址号。  |
| 功能码   | 0x01: 读线圈寄存器 0x02: 读离散输入寄存器<br>0x03: 读保持寄存器 0x04: 读输入寄存器<br>0x05: 写单个线圈寄存器 0x06: 写单个保持寄存器<br>0x0f: 写多个线圈寄存器 0x10: 写多个保持寄存器 |
| 起始地址  | 数据在通讯设备中的寄存器地址的开始位置编号，一般由设备厂家提供。   |
| 寄存器   | Modbus协议中表示数据存储的一个计量单位，1个线圈寄存器是1bit,1个保持寄存器是16bit  |
| 数据类型  | 在寄存器中的存储数据的解码类型，由设备厂家决定，数据解码时需要严格按照设备厂家提供的解码规则处理   |
| CRC校验 | 经过CRC校验算法后，添加CRC计算的值得在协议传输帧的最后面，主要是防止串口误码导致数据错误。   |

## 功能码说明

| 代码 | 中文名称     | 寄存器PLC地址    | 位操作/字操作 | 操作数量  |
|----|----------|-------------|---------|-------|
| 01 | 读线圈状态    | 00001-09999 | 位操作     | 单个或多个 |
| 02 | 读离散输入状态  | 10001-19999 | 位操作     | 单个或多个 |
| 03 | 读保持寄存器   | 40001-49999 | 字操作     | 单个或多个 |
| 04 | 读输入寄存器   | 30001-39999 | 字操作     | 单个或多个 |
| 05 | 写单个线圈    | 00001-09999 | 位操作     | 单个    |
| 06 | 写单个保持寄存器 | 40001-49999 | 字操作     | 单个    |
| 15 | 写多个线圈    | 00001-09999 | 位操作     | 多个    |
| 16 | 写多个保持寄存器 | 40001-49999 | 字操作     | 多个    |

## 寄存器说明

| 寄存器类型   | 寄存器描述   |
|---------|---|
| 线圈寄存器   | 可以类比为开关量，每一个bit都对应一个信号的开关状态。所以一个byte就可以同时控制8路的信号。比如控制外部8路io的高低。<br>线圈寄存器支持读也支持写，写在功能码里面又分为写单个线圈寄存器和写多个线圈寄存器。对应上面的功能码也就是0x01 0x05 0x0f |
| 离散输入寄存器 | 离散输入寄存器就相当于线圈寄存器的只读模式，他也是每个bit表示一个开关量，而他的开关量只能读取输入的开关信号，是不能够写的。<br>比如我读取外部按键的按下还是松开。所以功能码也简单就一个读的 0x02                                |
| 保持寄存器   | 这个寄存器的单位不再是bit而是两个byte也就是可以存放具体的数据量的，并且是可读写的。<br>比如我我设置时间年月日，不但可以写也可以读出来现在的时间。写也分为单个写和多个写，所以功能码有对应的三个0x03 0x06 0x10                   |
| 输入寄存器   | 这个和保持寄存器类似，但是也是只支持读而不能写。一个寄存器也是占据两个byte的空间。类比我我通过读取输入寄存器获取现在的AD采集值。对应的功能码也就一个 0x04  |

## 寄存器地址分配

| 寄存器PLC地址    | 寄存器协议地址     | 适用功能        | 寄存器种类  | 读写状态 |
|-------------|-------------|-------------|--------|------|
| 00001-09999 | 0000H-FFFFH | 01H 05H 0FH | 线圈状态   | 可读可写 |
| 10001-19999 | 0000H-FFFFH | 02H         | 离散输入状态 | 可读   |
| 30001-39999 | 0000H-FFFFH | 04H         | 输入寄存器  | 可读   |
| 40001-49999 | 0000H-FFFFH | 03H 06H 0FH | 保持寄存器  | 可读可写 |

## PLC地址和协议地址区别

PLC地址可以理解为协议地址的变种，在触摸屏和PLC编程中应用较为广泛。

### 1. 寄存器PLC地址

寄存器PLC地址指存放于控制器中的地址，这些控制器可以是PLC，也可以是触摸屏，或是文本显示器。PLC地址一般采用10进制描述，共有5位，其中第一位代码寄存器类型。第一位数字和寄存器类型的对应关系参考[寄存器地址分配](#)。PLC地址例如40001、30002等。

### 2. 寄存器协议地址

寄存器协议地址指通信时使用的寄存器地址，例如PLC地址40001对应寻址地址0x0000，40002对应寻址地址0x0001。寄存器寻址地址一般使用16进制描述。再如，PLC寄存器地址40003对应协议地址0x0002，PLC寄存器地址30003对应协议地址0x0002。虽然两个PLC寄存器通信时使用相同的地址，但是需要使用不同的命令访问，所以访问时不存在冲突。

## Modbus Slave (从站)模拟器

[Modsim32](#) (点击下载)

## Modbus Master (主站)测试工具

[Modscan32](#) (点击下载)

From:

<https://wiki.freeioe.org/> - FreeIOE 知识库

Permanent link:

<https://wiki.freeioe.org/modbus/start?rev=1569641270>

Last update: 2022/07/12 11:29

