1. 虽然现在的硬件可以实现双方相互主动通信的功能，但是Modbus通信协议却规定通信双方有一方为master另一方为slave。master发送Query Message，slave接收到咨询信息后才能回复Response Message。
2. Modbus帧格式
3. Modbus-RTU格式和下表的格式完全相同。不同数据帧的区分是由延时来确定的。RTU规定每次Query Message或Response Message的结束是以未再收到下一个字符间隔时间来判断，这个时间间隔为3.5个字符的时间。例如：通信速率为9600bps，每个字符含8个位，再加上1个起始位和一个停止位，则每个字符需要传送10个位。则3.5个字符的通信时间为(3.5\*10)/9600=0.00365秒。
4. Modbus-ASCII格式形式为：将下表的每个字节转换成两个ASCII字符，然后在字符串的头部加上冒号“：”作为起始符，在尾部加上回车符和换行符即<CR><LF>为作结束符。
5. Modbus-TCP格式是在Modbus-RTU之上加上一个TCP/IP头部。设备编号字段根据需要可以留空白也可以正常使用，CRC校验码字段空白不使用。

|  |  |
| --- | --- |
| Modbus Format | |
| Device Address（1个字节） | 表示设备编号。在Modbus-TCP上可以使用IP地址来区分设备，则可以不使用设备号。在有些设备上Modbus-TCP不使用该字段其内容为0，另一些设备会正常使用该字段，所以使用Modbus-TCP时要注意该字段，否则会连机失败。 |
| Function Code（1个字节） | 表示功能码。 |
| Data（n个字节） | 依据Function Code附加的数据。比如Function Code为1，上位机要查询线圈状态，则Data应包含要查询的地址和数量。 |
| Error Check Code（2字节） | 表示错误校验码。如果是Modbus-RTU则使用RCR校验码；如果是Modbus-ASCII则使用LCR校验码；如果是Modbus-TCP则不使用错误校验码，因为TCP/IP通信本身就有错误校验。 |

1. Modbus常用功能码

Modbus最常用的功能码有8个：1、2、3、4、5、6、15、16

|  |  |
| --- | --- |
| 功能码1 | Read Coil Status读线圈的状态 |
| 功能码2 | Read Input Status读输入状态 |
| 功能码3 | Read Holding Registers读寄存器的值 |
| 功能码4 | Read Input Registers读输入寄存器的值 |
| 功能码5 | Force Single Coil强制设置单个线圈的状态 |
| 功能码6 | Preset Single Register强制写单个寄存器的值 |
| 功能码15 | Force Multiple Coils强制写多个线圈的状态 |
| 功能码16 | Preset Multiple Registers强制写多个寄存器的值 |

1. Modbus下位机异常帧格式

|  |  |
| --- | --- |
| Modbus Exception Message Format | |
| Device Address | 下位机编号 |
| Function Code | 将接收到的上位机的功能码的最高位置1，如：上位机发送功能码1读取线圈状态，但要读取的线圈地址超过了下位机的线圈地址范围，则下位机应该返回功能码0x81，以表示返回的是一个异常帧。 |
| Error Code | 错误码，用来表示错误的原因。比如上位机要读取的线圈的地址超出下位机线圈地址范围，则错误码应该返回0x02，以表示地址出错。 |

1. 串口Modbus的设备编号

Modbus设备编号占用1个字节，能表示0到255总共256个设备，但是0用于广播帧，所以下位机不允许使用0编号。所有下位机都会接收和执行设备号为0的广播帧，但是不会回复上位机。

1. Modbus-TCP的设备编号

有些设备的Modbus-TCP不使用设备编号字段，而是使用IP地址来区分下位机，这样Modbus-TCP的下位机数量就没有限制。由于TCP通信协议无法发送广播，所以上位机要发送广播时必须对每个Modbus-TCP连接的下位机发送同一个数据帧。

Modbus-TCP的下位机设备（slave）应该具有能连接多个上位机设备（master）的能力。