一、何为Modbus通信协议

1.1 Modbus基本介绍

Modbus是一种通信协议，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协议介绍：**Modbus是 OSI 模型第 7 层上的应用层报文传输协议**，它在连接至不同类型总线或网络的设备之间提供客户机/服务器通信。**包括了ASCII、RTU、TCP三种；**

- Modbus RTU与Modbus ASCII均为串行传输方式。其中，Modbus RTU采用二进制表现形式以及紧凑数据结构，通信效率较高，应用比较广泛。而Modbus ASCII采用ASCII码传输，并且利用特殊字符作为其字节的开始与结束标识，其传输效率要远远低于Modbus RTU协议。
- Modbus TCP是通过工业以太网TCP/IP网络传输的Modbus通信。Modbus数据传输提供了连接在以太网 TCP/IP网络上的客户机与服务器之间的实时通讯。

1.2 Modbus的工作方式请求/应答

Modbus是使用主从关系实现的请求 - 响应协议。在主从关系中，通信总是成对发生，一个设备必须发起请求，然后等待响应，并且发起设备(主设备)负责发起每次交互。通常，主设备是人机界面(HMI)或监控和数据采集(SCADA)系统，从设备是传感器、可编程逻辑控制器(PLC)或可编程自动化控制器(PAC)。这些请求和响应的内容以及发送这些消息的网络层由协议的不同层来定义。

- Modbus的**工作方式**是请求/应答，每次通讯都是主站先发送指令，可以是广播，或是向特定从站的单播；从站响应指令，并按要求应答，或者报告异常。当主站不发送请求时，从站不会自己发出数据，从站和从站之间不能直接通讯 。

1.3 Modbus通讯方式

MODBUS 协议允许在各种网络体系结构内进行简单通信。每种设备（PLC、HMI、控制面板、驱动程序、动作控制、输入/输出设备）都能使用 MODBUS协议来启动远程操作。

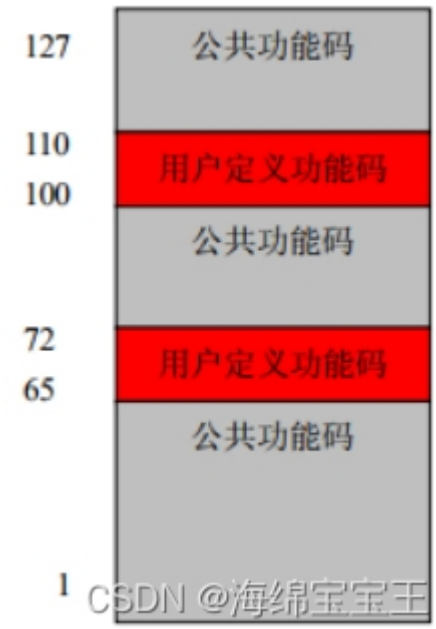
Modbus有下列三种通信方式：

- 以太网：对应的通信模式是Modbus TCP/IP
- 异步串行传输（各种介质如有线RS-232-/422/485/；光纤、无线等）：对应的通信模式是Modbus RTU或Modbus ASCII
- 高速令牌传递网络：对应的通信模式是Modbus PLUS

二、功能码和寄存器地址

2.1 功能码分类

一共有三类Modbus功能码：公共功能码、用户定义功能码、保留功能码



1）公共功能码
是较好地被定义的功能码、保证是唯一的、MODBUS 组织可改变的、公开证明的、具有可用的一致性测试、MB IETF RFC 中证明的、包含已被定义的公共指配功能码和未来使用的未指配保留供功能码
2）用户定义功能码
有两个用户定义功能码的定义范围，即 65 至 72 和十进制 100 至 110、用户没有 MODBUS 组织的任何批准就可以选择和实现一个功能码、不能保证被选功能码的使用是唯一的、如果用户要重新设置功能作为一个公共功能码，那么用户必须启动 RFC，以便将改变引入、公共分类中，并且指配一个新的公共功能码。
3）保留功能码
一些公司对传统产品通常使用的功能码，并且对公共使用是无效的功能码

2.2 常用功能码

常用功能码：十六进制的

- 读取输出线圈 0x01
- 读取输入线圈 0x02
- 读取输出寄存器 0x03
- 读取输入寄存器 0x04
- 写入单个输出线圈 0x05
- 写入多个输出线圈 0x0F

- 写入单个输出寄存器 0x06
- 写入多个输出寄存器 0x10
- **注意**：H03和H04的区别：两个都是读取寄存器，用H03读取的寄存器可以写入，用H04读取的寄存器，不能在用H06写入；（通讯设备厂家读取功能码有要求用H03还是H04）

功能码	描述	说明	MOUBUS地址	异常功能码(+ 0×80)
1	读取输出线圈	位操作	00001 ~ 09999	0×81
2	读取输入线圈	位操作	10001 ~ 19999	0×82
3	读取输出寄存器	字操作	40001 ~ 49999	0×83
4	读取输入寄存器	字操作	30001 ~ 39999	0×84
5	写入单个输出线圈	位操作	00001 ~ 09999	0×85
6	写入单个输出寄存器	字操作	40001 ~ 49999	0×86
0F	写入多个输出线圈	位操作	00001 ~ 09999	0×8F
10	写入多个输出寄存器	字操作	40001 ~ 49999	0×90

2.3 寄存器地址

通常Modbus地址由5位数字组成，包括起始的数据类型代号，以及后面的偏移地址。

Modbus地址模型的编号从1开始。

由于每一种数据都最大支持65536个元素，因此理论上，
对于线圈型数据来说，其地址范围为：000001~~065536；~~
~~离散量输入，其地址范围为：10000~~1165536；
输入寄存器，其地址范围为：300001~~365536；~~
~~保持寄存器，其地址范围为：40000~~1465536；

由于65536是比较大的数值，实际应用一般不需要这么大的存储区，因此PLC厂家普遍采用的是10000以内的地址范围，即：
线圈地址范围：00001~~09999；~~
~~离散量输入地址范围：10001~~19999；
输入寄存器地址范围：30001~~39999；~~
~~保持寄存器地址范围：40001~~49999；

国内汇川就用 全地址，000001~065536

Modbus Master协议库支持如下地址:

MOUBUS地址	含义	访问方式
00001~09999	离散量输入(Discrete Input)	只读
10001~19999	线圈(Coils)；	读/写
30001~39999	输入寄存器(Input registers)；	只读
40001~49999	保持寄存器(Holding registers)；	读/写

由上表可知，Modbus通讯的地址是从1开始的，如果遇到从机设备的寄存器编号从0开始，那么就需要考虑给通讯地址加1处理。

三、Modbus RTU 数据帧结构

3.1 Modbus RTU 数据帧结构

RTU协议中的指令由地址码(一个字节），功能码（一个字节），起始地址（两个字节），数据（N个字节），校验码（两个字节）五个部分组成。数据由数据长度（两个字节，表示的是寄存器个数，假定为M）和数据正文（M乘以2个字节）组成。

从站地址：一个字节，作用是索引
功能码：一个字节，表明读写功能
数据：通信所传输的数据，可以是多字节
校验：判断接收的数据在传输过程中是否有损失，两个字节

帧格式（最大256字节）

地址	功能码	数据	CRC校验
1byte = 8bit	1byte = 8bit	Nbyte = N×8bit	2byte = 16bit

子节点地址	功能代码	数据	CRC
1 字节	1 字节	0 到 252 字节	2 字节 CRC 低 | CRC 高

3.2 Modbus RTU示例

3.2.1 读输出线圈【0x01功能码】

- **发送报文格式：**

从站地址	功能码	起始地址（高位）	起始地址（低位）	数量（高位）	数量（低位）	校验码（低位）	校验码（高位）
0x03	0x01	0x00	0x13	0x00	0x1B	XX	XX

含义：
目标所在从站：0x03（3号）
命令：0x01（读线圈状态）
寄存器种类：线圈状态
目标起始索引地址：0x0013（索引地址 = 19）（16进制13，转换索引地址，十进制19）
目标起始PLC地址： 00001 + 19 = 00020（线圈状态PLC地址范围：00001-09999）
读取线圈数量：0x1B（1B = 27个，即27bit数据）
目标线圈范围：00020 - 00046（从00020开始27个线圈）
校验码：XXXX

- **从机正常应答格式**

从站地址	功能码	字节计数	字节一	字节二	字节三	字节四	校验码（低位）	校验码（高位）
0x03	0x01	0x04	0xCD	0x6B	0xB2	0x05	YY	YY

含义：返回从3号从站读取的共4个字节的数据，数据为：0xCD、0x6B、0xB2、0x05（为了举例子暂时编的），校验码为YYYY。

补充：由于读取的是线圈（bit），若读取的个数不为8的倍数，比如这次读27bit，则取整数字节4字节32bit返回，剩余5bit用0补全。

3.2.2. 读保持寄存器 【0x03功能码】

请求帧格式

从站地址	功能码	起始地址（高位）	起始地址（低位）	数量（高位）	数量（低位）	校验码（低位）	校验码（高位）
0x03	0x03	0x00	0x06	0x00	0x02	XX	XX

含义：
目标所在从站：0x03（3号）
命令：0x03（读保持寄存器）
寄存器种类：保持寄存器
目标起始索引地址：0x0006（索引地址 = 6）
目标起始PLC地址： 40001 + 6 = 40007（保持寄存器PLC地址范围：40001-49999）
读取寄存器数量：0x02（02 = 2个，即2×2byte = 4byte数据）（一个寄存器为2字节）
目标线圈范围：40007-40008（从40007开始2个寄存器）
校验码：XXXX

正常应答帧格式

从站地址	功能码	字节计数	字节一（高位）	字节一（低位）	字节二（高位）	字节二（低位）	校验码（低位）	校验码（高位）
0x03	0x03	0x02	0xA1	0x05	0x04	0xCD	YY	YY

含义：返回从3号从站读取的共2个寄存器的数据，数据为：0xA105（40007上的数据）、0x04CD（40008上的数据）（为了举例子暂时编的），校验码为YYYY。

3.2.3. 读取输入寄存器持 【0x04功能码】

- 发送报文格式

从站地址	功能码	起始（高）	起始（低）	数量（高）	数量（低）	校验码
0x01	0x04	0x00	0x6B	0x00	0x02	XXXX

发送报文含义：读取1号从站输入寄存器，起始地址为0x6B=107，对应地址为30108，寄存器数量为0x02=2，即读取1号从站输入寄存器，地址从30108-30109，共两个寄存器

- 返回报文格式

从站地址	功能码	字节计数	1高	1低	2高	2低	校验
0x01	0x04	0x04	0x02	0x2B	0x01	0x06	XXXX

返回报文含义：返回1号从站输入寄存器30108-30109，共2个寄存器的数值，返回字节数为4个，分别为02、2B、01、06，30108对应数值为0x022B，30109对应的数值为0x0106

3.2.4 写入单个输出线圈【0x05功能码】

- 发送报文格式

从站地址	功能码	线圈（高）	线圈（低）	断通标志	断通标志	校验
0x01	0x05	0x00	0xAC	0xFF	0x00	XXXX

发送报文含义：预置1号从站单个线圈的值，线圈地址为0x00AC=172，对应地址为00173，断通标志0xFF表示置位，0x0000表示复位，即置位1号从站输出线圈00173

- 返回报文格式

从站地址	功能码	线圈（高）	线圈（低）	断通标志	断通标志	校验
0x01	0x05	0x00	0xAC	0xFF	0x00	XXXX

返回报文含义：预置单输出线圈原报文返回

3.2.5 写入单个输出寄存器【0x06功能码】

- 发送报文格式如下：

从站地址	功能码	寄存器高	寄存器低	写入值高	写入值低	校验
0x01	0x06	0x00	0x87	0x03	0x9E	XXXX

发送报文含义：预置1号从站单个输出寄存器的值，寄存器地址为0x0087=135，对应地址为40136，写入值为0x039E，即预置1号从站输出寄存器40136值为0x039E。

- 返回报文格式如下：

从站地址	功能码	寄存器高	寄存器低	写入值高	写入值低	校验
0x01	0x06	0x00	0x87	0x03	0x9E	XXXX

返回报文含义：预置单输出寄存器原报文返回。

3.2.6 写多个线圈【0x0f功能码】

请求帧格式

从站地址	功能码	起始地址（高位）	起始地址（低位）	数量（高位）	数量（低位）	字节数	字节一	字节二	字节三	校验码（低位）	
0x01	0x0f	0x00	0x13	0x00	0x15	0x03	0x12	0x1A	0x04	XX	

含义：
目标所在从站：0x01（1号）
命令：0x0f（写多个线圈）
寄存器种类：线圈
目标起始索引地址：0x0013（索引地址 = 19）
目标起始PLC地址： 00001 + 19 = 00020（线圈状态PLC地址范围：00001-09999）
写入线圈数量：0x15（0x15 = 21bit数据）
实际写入线圈数量：21bit + 3bit = 24bit = 3byte（只能
目标线圈范围：00020-00040（从00020开始21个线圈）
字节一的值：0x12（0x12 = 18 = 0001 0010）
字节二的值：0x1A（0x1A = 26 = 0001 1010）
字节三的值：0xAC（0x04 = 4 = 0000 0100）（最高三位为补0）
校验码：XXXX

补充：若写入的线圈个数不为8的倍数，则高位补0使其字节数为整数。

正常应答帧格式（在原报文基础上除去字节数和具体字节并加上当前校验码）

从站地址	功能码	起始地址（高位）	起始地址（低位）	数量（高位）	数量（低位）	校验码（低位）	校验码（高位）
0x01	0x0f	0x00	0x13	0x00	0x15	YY	YY

含义：向1号从站起始地址为00020处写入21个线圈的值成功，校验码为YYYY。

3.2.7 写多个保持寄存器【 0x10功能码】

请求帧格式

从站地址	功能码	起始地址（高位）	起始地址（低位）	数量（高位）	数量（低位）	字节数	字节	校验码（低位）	校验码（高位）
0x01	0x10	0x00	0x53	0x00	0x02	0x04	0x13141A1B	XX	XX

含义：目标所在从站：0x01（1号）
命令：0x10（写多个保持寄存器）
寄存器种类：保持寄存器
目标起始索引地址：0x0053（索引地址 = 83）
目标起始PLC地址： 40001 + 83 = 40084（线圈状态PLC地址范围：40001-49999）
写入寄存器数量：0x02（02 = 2个，即2×2byte = 4byte数据）（一个寄存器为2字节）
目标寄存器范围：40084-40085（从40084开始2个保持寄存器）
写入字节数：0x04（4个）
写入字节内容：0x13141A1B（40084保持寄存器写入0x1314 = 0001 0011 0001 0100 ， 40085保持寄存器写入0x1A1B = 0001 1010 0001 1011）
校验码：XXXX

正常应答帧格式（在原报文基础上除去字节数和具体字节并加上当前校验码）

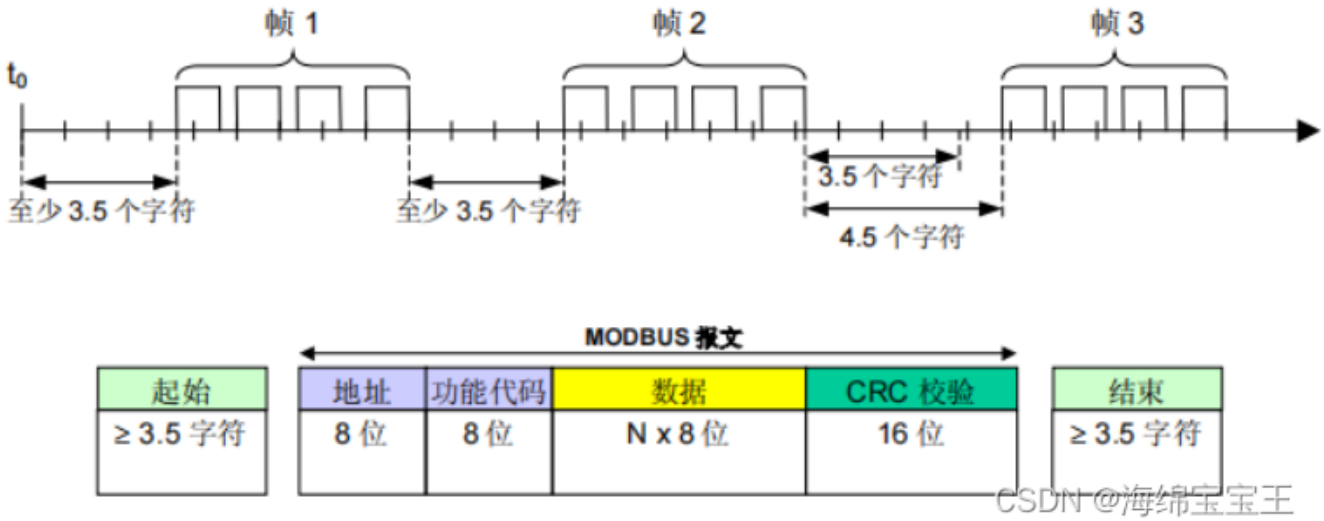
从站地址	功能码	起始地址（高位）	起始地址（低位）	数量（高位）	数量（低位）	校验码（低位）	校验码（高位）
0x01	0x10	0x00	0x53	0x00	0x02	YY	YY

含义：向1号从站的40084位置开始写入两个保持寄存器成功，校验码为YYYY。

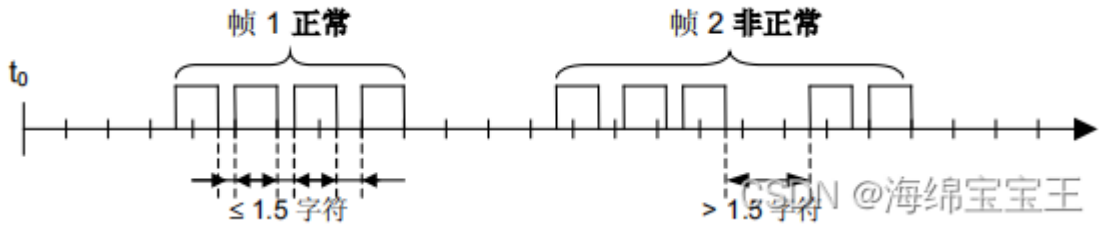
3.3 ModbusRTU如何判断开始与结束

整个报文帧必须以连续的字符流发送。如果两个字符之间的空闲间隔**大于 1.5 个字符时间**，则报文帧被认为不完整应该被接收节点丢弃

ModbusRTU协议中，需要用时间间隔来判断一帧报文的开始和结束，协议规定的时间为3.5个字符周期。在一帧报文开始前，必须有大于3.5个字符周期的空闲时间，一帧报文结束后，也必须要有3.5个字符周期的空闲时间，否则就会出现粘包的请况。**3.5个字符周期是一个具体时间，与波特率有**



关完整性判断：用1.5个字符时间判断帧的完整性，如果两个字符之间的空闲间隔大于1.5个字符时间，那么认为报文帧不完整，接收站丢弃这个报文帧。



帧校验（CRC循环冗余校验，2字节，检验整个报文内容）
发送方计算CRC的值并附加到帧尾，接收报文的过程中，接收设备重新计算CRC的值，并将计算的结果和接收到的CRC比较；若不相等，则产生了错误。

3.4 ModbusASCII传输模式

1）ModbusRTU与ModbusASCII在报文数据发送格式上几乎一样，但也存在一些区别，具体体现在：

ModbusASCII有开始字符和结束字符（CR LF），可以作为一帧数据开始和结束的标志，而ModbusRTU没有这样的标。
两者校验方式不同，ModbusRTU是CRC循环冗余校验，ModbusASCII是LCR纵向冗余校验。
在Modbus标准中，RTU是必须要求的，而ASCII是可选项，即作为一个Modbus通信设备可以只支持RTU，也可以同时支持RTU和ASCII，但不能只支持ASCII。

用两个ASCII码字符发送报文中的一个8位字节，当通信链路或者设备不能满足RTU模式的定时管理要求时使用。
编码系统：十六进制，ASCII字符0-9，A-F，报文中每个ASCII字符表示一个十六进制字符
字节格式（10位）

- 有校验：1个起始位，7个数据位，1个奇偶校验位，1个停止位
- 无校验：1个起始位，7个数据位，2个停止位

串行发送字符：从左到右：最低有效位（LSB）...最高有效位（MSB）
校验（奇校验ODD、偶校验EVEN和无校验NONE)

- 默认校验模式必须是偶校验，为了保证和其他产品的兼容性，建议使用无校验。
- 如果使用无校验，那么多附加一个停止位来满足定长10位异步字符。

帧格式（最大513字节）

起始	地址	功能码	数据	LRC校验	结束
1字符':'冒号	2字符	2字符	0-2×252字符	2字符	2字符（回车换行CR，LF）

1. 一个报文必须以一个冒号（：）字符（十六进制ASCII 3A）作为起始，以回车换行（CRLF）（十六进制ASCII 0D和0A）作为结束。

2. 报文中字符间时间间隔不能超过1s，除非配置了更长时间的超时，例如广域网应用可以要求4-5s超时，若超时，则表示已经出现错误。

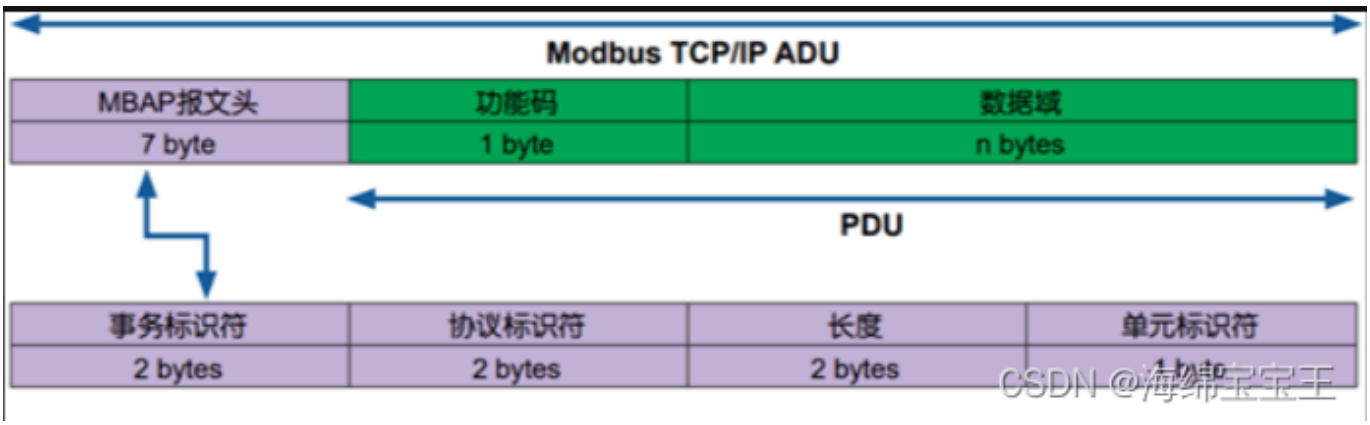
- **帧校验（LRC纵向冗余校验）**
  - 校验内容不包括起始符和结束符
  - 对报文中除起始符和结束符外的所有连续的8位字节相加，忽略任何进位，然后求其二进制补码得到LRC
  - LRC的结果也被编码为2个ASCII码字符。

四、 Modbus TCP 数据帧结构

4.1 ModbusTCP介绍

Modbus TCP协议是在Modbus RTU协议上加入MBAP(ModbusApplication Protocol Header)报文头，由于TCP是基于可靠连接的服务，所以在Modbus TCP协议中没有CRC校验，所有的Modbus TCPADU的发送和接收都是使用TCP传输控制协议，Modbus TCP/IP服务器端通常使用端口502作为接收报文的端口, IANA（Internet Assigned Numbers Authority，互联网编号分配管理机构）给Modbus协议赋予TCP端口号为**502**，这是目前在仪表与自动化行业中唯一分配到的端口号。

Modbus TCP数据帧结构如下



报文头(MBAP头)

内容	解释
事务处理标识	可以理解为报文序列号，一般每次通信后就要加1，以区分不同的通信数据报文。
协议标识符	00 00 表示Modbus TCP 协议
长度	接下来数据长度，单位字节
单元标识符	设备地址，一般为01

域	长度	描述	客户机	服务器
事务元标识符	2 个字节	MODBUS 请求/响应事务处理的识别码	客户机启动	服务器从接收的请求中重新复制
协议标识符	2 个字节	0=MODBUS 协议	客户机启动	服务器从接收的请求中重新复制
长度	2 个字节	以下字节的数量	客户机启动（请求）	服务器（响应）启动
单元标识符	1 个字节	串行链路或其它总线上连接的远程从站的识别码	客户机启动	服务器从接收的请求中重新复制

4.2 ModbusTCP 通讯示例

4.2.1 读输出寄存器：【功能码03】

- 发送报文格式

事务元处理标识符（高位）	事务元处理标识符（低位）	协议标识符高位	协议标识符低位	长度高位	长度低位	单元标识符	功能码	起始地址高位	起始位 低位	寄存器数量 高位
0x15	0x01	0x00	0x00	0x00	0x06	0xFF	0x03	0x00	0x06B	0x00

简化版：

事务/协议	长度	单元标识	功能码	起始高	起始低	数量高	数量低
0x15010000	0x0006	0xFF	0x03	0x00	0x6B	0x00	0x02

发送报文含义：读取服务器1号从站输出寄存器，起始地址为0x6B=107(相对地址),对应地址为40108，寄存器数量为0x02=2,即读取1号从站输出寄存器，地址从40108-40109，共两个寄存器的数值。

- 返回报文格式

事务元处理标识符（高位）	事务元处理标识符（低位）	协议标识符高位	协议标识符低位	长度高位	长度低位	单元标识符	功能码	字节计数	字节一高位	字节一 低位
0x15	0x01	0x00	0x00	0x00	0x07	0xFF	0x03	0x04	0x02	0x2B

简化版：

事务/协议	长度	单元标识	功能码	字节计数	1高	1低	2高	2低
0x15010000	0x0007	0x01	0x03	0x04	0x02	0x2B	0x01	0x06

返回报文含义：返回服务器1号从站输出寄存器40108-40109，共2个寄存器的数值，返回字节数为4个，分别为02、2B、01、06,40108对应数值为0x022B，40109对应数值为0x0106

五、 通信状况

5.1 通信状况

当主机向设备发送命令后，可能会出现以4下种情况：

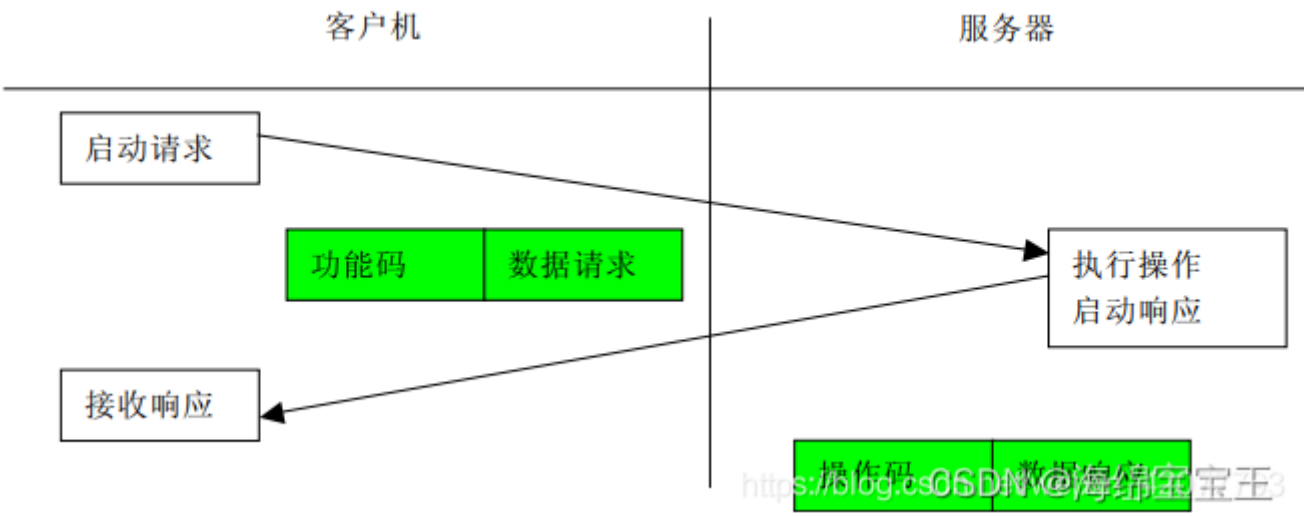
- 请求正确的到达服务器，并且请求的内容服务器可以处理，那么服务器返回一个正常响应
- 请求正确的到达服务器，但是请求服务器无法处理（例如请求读一个不存在的寄存器），此时服务器将返回一个异常响应，通知主机错误和错误的类型。
- 请求到达服务器，但是不正确，检测到了通信错误（奇偶校验、LRC、CRC等等），那么不返回响应，主机将最终成为超时状态
- 请求就没到达服务器，服务器没收到也就更不会响应，主机也会成为超时状态

5.2 响应类型

综上，根据服务器处理结果，可以建立两种类型的响应：一个正常MODBUS应答帧：

- 功能码域：响应功能码 = 请求功能码
- 数据域：请求中要求的任何数据
- 校验码：响应帧自身计算

从站地址	功能码	数据	校验码（低位）	校验码（高位）
从站自身地址	与请求功能码保持一致（范围：0x00-0x7f)	请求中要求的任何数据	XX	XX

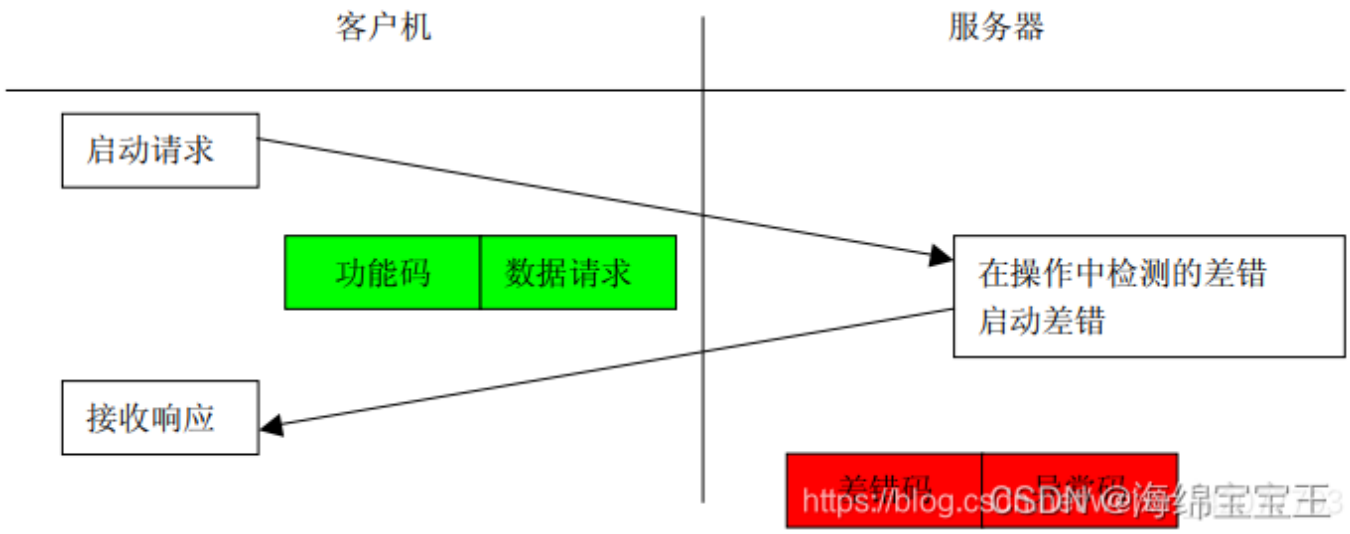


一个异常MODBUS应答帧：

- 用来为客户机提供处理过程中与被发现的差错相关的信息
- **功能码域**：响应功能码 = 请求功能码 + 0x80

- **数据域**：提供一个异常码来指示差错原因
- **校验码**：响应帧自身计算

从站地址	功能码	数据	校验码（低位）	校验码（高位）
从站自身地址	请求功能码 + 0x80	异常码	XX	XX



5.3 异常码表

代码	名称	含义
1	非法功能	对于服务器(或从站)来说，询问中接收到的功能码是不可允许的操作。 这也许是因为功能码仅仅适用于新设备而在被选单元中是不可实现的。 同时，还指出服务器(或从站)在错误状态中处理这种请求，例如：因为它是未配置的，并且要求返回寄存器值。
2	非法数据地址	对于服务器(或从站)来说，询问中接收到的数据地址是不可允许的地址。特别是，参考号和传输长度的组合是无效的。 对于带有 100 个寄存器的控制器来说，带有偏移量 96 和长度 4 的请求会成功，带有偏移量 96 和长度 5 的请求将产生异常码 02。
3	非法数据值	对于服务器(或从站)来说，询问中包括的值是不可允许的值。这个值指示了组合请求剩余结构中的故障， 例如：隐含长度是不正确的。 并不意味着，因为 MODBUS 协议不知道任何特殊寄存器的任何特殊值的重要意义， 寄存器中被提交存储的数据项有一个应用程序期望之外的值。
4	从站设备故障	当服务器(或从站)正在设法执行请求的操作时，产生不可重新获得的差错。
5	确认	与编程命令一起使用。服务器(或从站)已经接受请求，并且正在处理这个请求，但是需要长的持续时间进行这些操作。 返回这个响应防止在客户机(或主站)中发生超时错误。客户机(或主站)可以继续发送轮询程序完成报文来确定是否完成处理。
6	从属设备忙	与编程命令一起使用。服务器(或从站)正在处理长持续时间的程序命令。当服务器(或从站)空闲时，用户(或主站)应该稍后重新传输报文。

代码	名称	含义
0A	不可用网关路径	与网关一起使用，指示网关不能为处理请求分配输入端口至输出端口的内部通信路径。通常意味着网关是错误配置的或过载的。
0B	网关目标设备响应失败	与网关一起使用，指示没有从目标设备中获得响应。通常意味着设备未在网络中。