解调仪 MODBUS RTU 通讯规约

本协议支持标准 Modbus 协议，支持串口或 RS485 连接。

一、 物理层数据传输协议

a.波特率：9600 bps；

b.数据传输方式:串行异步通讯（1 位起始位、8 位数据位、1 位停止位、无校验位）。

二、 通讯协议 MODBUS RTU

Modbus 通讯规约被广泛的作为系统集成的标准。当通讯命令发送至仪器时，符合相应地址码的设备接收通讯命令，读取信息，如果没有出错则执行相应的任务，然后把执行结果返送给发送者；如果出错则返送错误报告信息（CRC16 校验码出错则不返送任何信息）或者不返送任何信息。其通讯数据帧格式如下：

地址码	功能码	数据区	校验码
1 字节(8 位)	1 字节 8 位	N 字节×8 位	2 个字节(16 位 CRC)

1. 地址码

地址码是信息帧的第一个字节（8 位），从 1 到 247。这个字节表明由用户设置地址的从机将接收由主机发送来的信息。每个从机都必须有唯一的地址码，并且只有符合地址码的从机才能响应回送。当从机回送信息时，相应的地址码表明该信息来自于相应地址的从机。

2. 功能码

主机发送的功能码告诉从机执行怎么样的任务,从机的功能码则表明从机响应主机相应任务进行了操作。Modbus 通讯规约定义的功能码为 01H 到 7FH，系统使用了其中一部分功能码。下表列出了所用到的功能码的具体含义及操作。

代码(Hex)	含 义	操 作
04H	读取输入寄存器	读取输入寄存器的一个或多个数据

注：如果从机发送的功能码的最高位是 1（功能码>127），则表明从机没有响应操作或发送出错。

3. 数据区

数据区是根据不同的功能码而不同。数据区可以是实际数值，设置点，主机发送给从机或从机发送给主机的地址。数据区包含需要从机执行什么动作或由从机采集的返送信息，这些信息可以是数值、参考地址等等。例如，功能码告诉从机读取寄存器的值，则数据区必须包含要读取的寄存器的起始地址及读取长度。如果功能码告诉从机设置某些连续寄存器的值，则数据区还要包含这些数值。对于不同的从机，地址和数据信息可能都不相同。

4. 校验码

校验码用于主机或从机判断接受信息是否出错。有时由于电子噪音或其它一些干扰，信息在传输过程中会发生细微的变化，这时自己根据信息计算所得的检验码与信息中包含的校验码就会不一致，从而判断接受信息出错。校验码保证了主机或从机对在传送中出错的信息不起作用，增加了系统的安全和效率。校验码采用 CRC-16 校验方法。

计算 CRC 码的步骤为：

(1).预置 16 位寄存器为 FFFFH。称此寄存器为 CRC 寄存器；

(2).把第一个 8 位数据与 CRC 寄存器的低位相异或，把结果放于 CRC 寄存器；

(3).把寄存器的内容右移一位(朝低位)，用 0 填补最高位，检查最低位；

(4).如果最低位为 0：重复第 3 步(再次移位)如果最低位为 1：CRC 寄存器与多项式 A001（1010 0000 0000 0001）进行异或；

(5).重复步骤 3 和 4，直到右移 8 次，这样整个 8 位数据全部进行了处理；

(6).重复步骤 2 到步骤 5，进行下一个 8 位数据的处理；

(7).最后得到的 CRC 寄存器即为 CRC 码。(CRC 码 = CRC_L + CRC_H)

5. 报文指令格式

读取输入寄存器指令 0x04

主机请求指令			从机响应		
从机地址	1 字节	1~247	从机地址	1 字节	
功能码	1 字节	0x04	功能码	1 字节	
起始寄存器地址	2 字节		寄存器字节数	1 字节	
寄存器个数	2 字节		寄存器值	N 字节	
CRC 校验码	2 字节		CRC 校验码	2 字节	

三、 报文举例说明

读取输入寄存器数据(功能码 04)。

1. 主机请求

读取输入地址 00~20 共 20 个寄存器值命令(十六进制)

从机地址	Modbus 功能码	起始地址高 8 位	起始地址低 8 位	寄存器数高 8 位	寄存器数低 8 位	CRC 低 8 位	CRC 高 8 位
[01]	[04]	[00]	[00]	[00]	[14]	[f0]	[05]
解调仪地址默认为 1，可更改	读输入寄存器	读取的起始地址		读取的数据长度		检验码	

2. 从机(解调仪)响应

字节序号	收到的十六进制数据	说 明	备注
1	01	设备地址	
2	04	Modbus 命令	功能码 04
3	28	长度	十进制：40
4-7	41 00 00 00	总传感器数量	十进制：8 个传感器
8-11	41 c8 cc cd	ID=1 的传感器物理量	十进制：25.1
12-15	41 c8 cc cd	ID=2 的传感器物理量	十进制：25.1
16-19	41 c9 99 9a	ID=3 的传感器物理量	十进制：25.075
20-23	41 c8 cc cd	ID=4 的传感器物理量	
24-27	41 cb 33 33	ID=5 的传感器物理量	
28-31	41 c5 99 9a	ID=6 的传感器物理量	
32-35	41 c7 33 33	ID=7 的传感器物理量	
36-39	41 c9 99 9a	ID=8 的传感器物理量	
40-43	00 00 00 00	ID=9 的传感器物理量	
44-45	53 dd	CRC 校验码	

四、 输入寄存器通讯地址表

地址	项目	数据类型	读写	实际值	单位	备注
0000	传感器总数量	Short	R	计算方式见“五.32位浮点数转换计算方法”	按传感器类型不同而不一样。	解调仪识别到的传感器总数量
0001		Short	R			
0002	1 号传感器物理量	Short	R			解调仪传感器分配的 ID=1
0003		Short	R			
0004	2 号传感器物理量	Short	R			解调仪传感器分配的 ID=2
0005		Short	R			
0006	3 号传感器物理量	Short	R			解调仪传感器分配的 ID=3
0007		Short	R			
0008	4 号传感器物理量	Short	R			解调仪传感器分配的 ID=4
0009		Short	R			
0010	5 号传感器物理量	Short	R			解调仪传感器分配的 ID=5
0011		Short	R			
0012	6 号传感器物理量	Short	R			解调仪传感器分配的 ID=6
0013		Short	R			
0014	7 号传感器物理量	Short	R			解调仪传感器分配的 ID=7
0015		Short	R			
0016	8 号传感器物理量	Short	R			解调仪传感器分配的 ID=7
0017		Short	R			
0018	9 号传感器物理量	Short	R			解调仪传感器分配的 ID=7
0019		Short	R			
2*n	n 号传感器物理量	Short	R			解调仪传感器分配的 ID=n
2*n+1		Short	R			

五、 32 位浮点数转换计算方法

- 例子：收到 ID=1 的传感器物理量四个字节:0x41 0xc8 0xcc 0xcd

转化为浮点数：

float f1 = GetFloat(0x41,0xc8,0xcc,0xcd);

f1 = 25.1

- **4 个字节转化为浮点数计算公式(c 函数)：**

```
float GetFloat(unsigned char c3, unsigned char c2,unsigned char c1, unsigned char c0)
{
    float f = 0.0;
    unsigned char a[4];
    a[3] = c3;
    a[2] = c2;
    a[1] = c1;
    a[0] = c0;
    memcpy(&f,a,4);
    return f;
}
```

- 输入寄存器输出数据解析

解调仪物理量保存在输入寄存器中，物理量是按 32 位浮点数保存，以重要数据优先的格式，使用连续两个地址,共使用 4 个字节。

解调仪的传感器的 ID 号就是访问物理量的地址相关：

地址计算是:输入寄存器地址 1=2*ID(注:ID 号为传感器的编号)

输入寄存器地址 2=2*ID+1(注:ID 号为传感器的编号)