	MODBUS 通讯帧格式			
	部门:	研发部	文献类别:	指令协议
	日期:	2011-5-18	仪器型号:	CHT3563

通讯协议采用 MODBUS 格式，模式采用 RTU。即 3.5 个停止位作为起始和终止位。每个字节数据间时间不超过 1.5 个停止位。选定串口种类 (Rs232/Rs485)，并将将仪器的通讯波特率设成和上位机一致。串口通讯格式：数据位 8 位，停止位 1 位，无硬件握手。

一. 寄存器概述

1.保持寄存器

名称	地址	值
测试功能	0x0001	R:0x0000, V:0x0001, RV:0x0002
电阻量程	0x0002	0x0000-0x0001
电压量程	0x0003	0x0000
量程自动	0x0004	ON:0x0001, OFF:0x0000
采样速率	0x0005	EX:0x0000, FAST:0x0001, MED:0x0002, SLOW:0x0003
平均次数	0x0006	0x0001-0x0010
比较器开关	0x0007	ON:0x0001, OFF:0x0000
比较器档位	0x0008	0x0002-0x0004
比较器讯响	0x0009	OFF:0x0000, HL:0x0001, IN:0x0002
触发源	0x000A	0x0000-0x0003: INT MAN, EXT, BUS
触发延时	0x000B	0-9999
电阻上限值 1H	0x000C	IEEE32 格式
电阻上限值 1L	0x000D	IEEE32 格式
电阻上限值 2H	0x000E	IEEE32 格式
电阻上限值 2L	0x000F	IEEE32 格式
电阻上限值 3H	0x0010	IEEE32 格式
电阻上限值 3L	0x0011	IEEE32 格式
电阻上限值 4H	0x0012	IEEE32 格式
电阻上限值 4L	0x0013	IEEE32 格式

电压上限值 1H	0x0014	IEEE32 格式
电压上限值 1L	0x0015	IEEE32 格式
电压上限值 2H	0x0016	IEEE32 格式
电压上限值 2L	0x0017	IEEE32 格式
电压上限值 3H	0x0018	IEEE32 格式
电压上限值 3L	0x0019	IEEE32 格式
电压上限值 4H	0x001A	IEEE32 格式
电压上限值 4L	0x001B	IEEE32 格式
清零(只写)	0x0020	1: 清零
触发(只写)	0x0021	1: 总线触发

2.输入寄存器

名称	地址	值
通道 1 电阻值 H	0x1001	IEEE32 浮点数格式
通道 1 电阻值 L	0x1002	IEEE32 浮点数格式
通道 1 电压值 H	0x1003	IEEE32 浮点数格式
通道 1 电压值 L	0x1004	IEEE32 浮点数格式
通道 1 测量结果	0x1005	0: OFF 1: IN 2: NG
通道 2 电阻值 H	0x1006	IEEE32 浮点数格式
通道 2 电阻值 L	0x1007	IEEE32 浮点数格式
通道 2 电压值 H	0x1008	IEEE32 浮点数格式
通道 2 电压值 L	0x1009	IEEE32 浮点数格式
通道 2 测量结果	0x100A	0: OFF 1: IN 2: NG
通道 3 电阻值 H	0x100B	IEEE32 浮点数格式
通道 3 电阻值 L	0x100C	IEEE32 浮点数格式
通道 3 电压值 H	0x100D	IEEE32 浮点数格式
通道 3 电压值 L	0x100E	IEEE32 浮点数格式

通道 3 测量结果	0x100F	0: OFF 1: IN 2: NG
...
...
通道 24 电阻值 H	0x1074	IEEE32 浮点数格式
通道 24 电阻值 L	0x1075	IEEE32 浮点数格式
通道 24 电压值 H	0x1076	IEEE32 浮点数格式
通道 24 电压值 L	0x1077	IEEE32 浮点数格式
通道 24 测量结果	0x1078	0: OFF 1: IN 2: NG

二. MODBUS 指令

1. 读保持寄存器指令 (0x03)

请求帧		
地址码	0x01~0xFF	1 字节
指令码	0x03	1 字节
起始寄存器地址		2 字节
寄存器数量		2 字节
CRC 校验码		2 字节

正常响应帧		
地址码	0x01~0xFF	1 字节
指令码	0x03	1 字节
字节数		1 字节
输入寄存器		n 字节
CRC 校验码		2 字节

异常响应帧		
地址码	0x01~0xFF	1 字节
异常码	083	1 字节
错误码	01-04	1 字节
CRC 校验码		2 字节

举例:

读仪器的电阻量程+ 电压量程(仪器地址为 01)

发送:01 03 0002 0002 65CB

仪器返回:010304000400017A32

仪器的电阻量程为 0004, 电压量程为 0001

2. 读输入寄存器指令 (0x04)

请求帧		
地址码	0x01~0xFF	1 字节
指令码	0x04	1 字节
起始寄存器地址		2 字节
寄存器数量		2 字节
CRC 校验码		2 字节

正常响应帧		
地址码	0x01~0xFF	1 字节
指令码	0x04	1 字节

字节数		1 字节
输入寄存器		n 字节
CRC 校验码		2 字节

异常响应帧		
地址码	0x01~0xFF	1 字节
异常码	084	1 字节
错误码	01-04	1 字节
CRC 校验码		2 字节

举例:

读仪器测试的电阻值和电压值

发送:01 04 1001 0004 A4C9

仪器返回:010408E7D49B3E260A9D3FC98A

仪器的电阻值为 0.304 Ω , 电压值为 1.2269

注:仪器返回的数据为 IEEE 格式,有关 IEEE 格式参考附录

3.写寄存器指令 (0x10)

请求帧		
地址码	0x01~0xFF	1 字节
指令码	0x10	1 字节
起始寄存器地址		2 字节
寄存器数量		2 字节
字节数		1 字节
寄存器值		N 字节
CRC 校验码		2 字节

正常响应帧		
地址码	0x01~0xFF	1 字节
指令码	0x10	1 字节
起始地址		2 字节
寄存器数量		2 字节
CRC 校验码		2 字节

异常响应帧		
地址码	0x01~0xFF	1 字节
异常码	0x90	1 字节
错误码	01-04	1 字节
CRC 校验码		2 字节

举例:

设置仪器的电阻量程 10m Ω + 电压量程 60V(仪器地址为 01)

发送:01 10 0002 0002 0001 0001 E276

仪器返回:011000020002E008

仪器设置成功

4.触发仪器测试指令（0x74）

请求帧		
地址码	0x01~0xFF	1 字节
指令码	0x74	1 字节
CRC 校验码		2 字节

正常响应帧		
地址码	0x01~0xFF	1 字节
指令码	0x74	1 字节
字节数		1 字节
输入寄存器		n 字节
CRC 校验码		2 字节

举例:

读仪器测试的电阻值和电压值

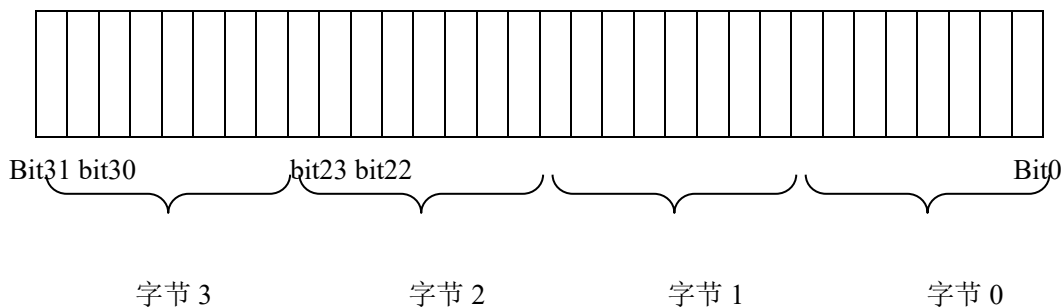
发送:01 74 00 07

仪器返回:017408E7D49B3E260A9D3FC98A

仪器的电阻值为0.304 Ω , 电压值为1.2269

附录: IEEE32 浮点数、有符号整数的数据表示格式

IEEE32 是国际电工委员会制订的浮点数表示方式,主要内容是用 4 个字节来表示浮点数,可以表示的数据的负数范围是 $-2*2^{128}\sim-2^{-127}$, $2^{-127}\sim2*2^{128}$ 。如下图所示,最高位(bit31)表示浮点数的符号位(0 为正,1 为负);bit30-bit23 这 8 位表示浮点数的阶码(以 2 为底),取值范围 0-FF (十六进制),用 7F 表示阶码为 0, 80 表示阶码为 1, 7E 表示阶码为-1, 依次类推。bit22-bit0 表示浮点数的尾数的小数部分,尾数的整数部分缺省永远是 1。



现用一个例子简要说明 IEEE32 浮点数的表示方法,假设现在有一个 IEEE32 浮点数,它的十六进制格式是 0X42C80000, 二进制格式是 01000010 11001000 00000000 00000000, 按照上面的规则,阶码应该是 10000101, 即 0X85, 尾数的小数部分是二进制的 0.1001, 换算成十进制即是 0.5625, 由于尾数的整数部分缺省永远是 1, 因此该浮点数的值应该是 $+15625*2^{85-7F}=100$ 。

由于 IEEE32 浮点数只用 4 个字节即可以表示很大范围的数据,因此在通讯中经常使用,以提高通讯效率。IEEE32 浮点数在二进制通讯方式中使用较多。在实际通讯过程中,如果 I/O 设备采用的是 INTEL 公司的 CPU,则在通讯过程中不管是 I/O 设备向 PC 机还是 PC 机向 I/O 设备发送浮点数,都必须按照字节 0、字节 1、字节 2、字节 3 的顺序发送如,果 I/O 设备采用的是 MOTOROLA 公司的 CPU,数据即发送顺序则相反。这种情况并不绝对,只代表多数情况,在涉及到数据格式时应首先以 I/O 设备的使用手册为准。

G.2 16 位及 32 位有符号整数

16 位和 32 位有符号整数使用最高位作为符号位, 0 代表正数, 1 代表负数, 负数用补码表示, 例如用 16 位有符号整数表示-100, 应该是+100 的补码, 即 0X64 的补码 0XFF9C。