

低成本倾角传感器 —— 工品标准 军品技术产品规格书

(Product Specification)

型号：LVT527T

描述：MODBUS 数字输出型双轴倾角传感器



生产执行标准参考

- 企业质量体系标准：ISO9001:2015 标准（认证号：328406）
- 倾角传感器生产标准：GB/T 191 SJ 20873-2003 倾斜仪、水平仪通用规范
- 倾角传感器计量院校准标准：JJF1119-2004 电子水平仪校准规范
- 陀螺加速度测试标准：QJ 2318-92 陀螺加速度计测试方法
- 光纤陀螺仪测试方法：GJB 2426A-2004
- 产品环境试验检测标准：GJB150
- 电磁抗干扰试验标准：GB/T 17626
- 版本：VT(2018-2019)
- 修订日期：2018.02.18

一、产品介绍

LVT527T-M 型倾角传感器是零零传感开发的一款低成本数字输出型 MEMS 传感器,内置高精度检测芯片,通过测量静态重力加速度,转换为角度值输出,独特的卡尔曼算法,确保角度值准确,精度可达 0.3°。该产品使用简单,操作方便,是角度测量领域的理想选择。输出方式为 MODBUS 接口标准。

二、主要特性

- 双轴倾角检测
- 量程 0 ~ ±180° (可选)
- 输出方式 MODBUS
- 工作电压 DC5V 或 9~36V
- 尺寸(L55*W37*H24mm) (可定制)
- 高抗振性能>3500g
- 全范围精度 0.3°
- 波特率 2400-115200(可选)
- 宽温工作-40 ~ +85°C
- 防护等级 IP67

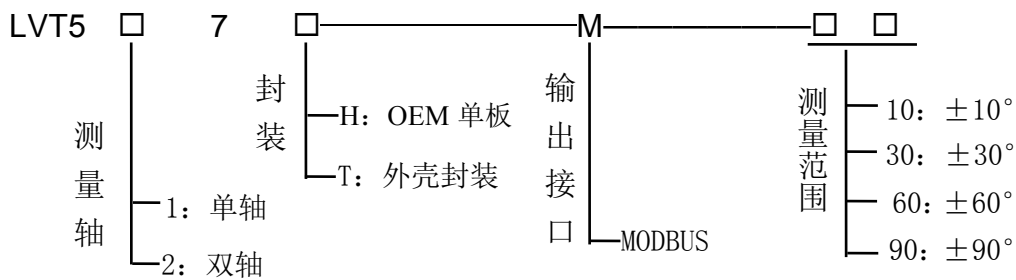
三、产品应用

- 高压电线塔杆监测
- 铁路机车监测
- 高空作业车
- 云台调平
- 船舶航行姿态测量
- 基于倾斜监测
- 卫星太阳能天线定位
- 医疗设备
- 各种工程机械角度控制
- 高精度激光平台设备



LVT527T-M MODBUS 数字型双轴倾角传感器

产品订购信息



例如：LVT527T-M—60 (Modbus)：双轴/外壳封装/±60°测量范围/MODBUS；默认**水平向上安装**，需其他安装方式，参照“产品安装示意图”，订购时备注。

产品性能指标

参数	条件	LVT527T-M-10	LVT527T-M-30	LVT527T-M-60	LVT527T-M-90	单位
测量范围		±10	±30	±60	±90	°
测量轴		X、Y 轴	X、Y 轴	X、Y 轴	X、Y 轴	
零点温度漂移	-40 ~ +85°C	±0.01	±0.01	±0.01	±0.01	°/°C
灵敏度温度系数	-40 ~ +85°C	≤150	≤150	≤150	≤150	ppm/°C
频率响应		100	100	100	100	Hz
分辨率	带宽 5Hz	0.05	0.05	0.05	0.05	°
精度	-40 ~ +85°C	0.3	0.3	0.3	0.3	°
长期稳定性	-40 ~ +85°C	<0.2				°
上电启动时间		0.2	0.2	0.2	0.2	s
响应时间		0.01	0.01	0.01	0.01	s
平均工作时间	≥55000 小时/次					
抗冲击	3500g,0.5ms,3 次/轴					
抗震动	10grms、10 ~ 1000Hz					
绝缘电阻	≥100MΩ					
防水等级	IP67					
电缆线	标配 1.5 米长度、耐磨、防油、宽温、屏蔽电缆线 5*0.2					
重量	75g (不含包装盒)					

* 本性能参数只列出±10°、±30°、±60°、±90° 系列以作参考，其它测量范围请以最相邻参数为参考。

LVT527T-M MODBUS 数字型双轴倾角传感器

电气指标

参数	条件	最小值	典型值	最大值	单位
工作电压	默认		5		V
工作电压	可选	9	12	36	V
工作电流			35		mA
工作温度		-40		+85	°C
存储温度		-55		+100	°C

名词解释：

分辨率： 是指传感器在测量范围内能够检测和分辨出的被测量的最小变化值。

绝对精度： 绝对精度是指在常温条件下, 对传感器的绝对线性度、重复性、迟滞、零点偏差、及横轴误差的综合误差。

长期稳定性： 长期稳定性是指传感器在常温条件下, 经过一年的长期工作下最大值与最小值之的偏差。

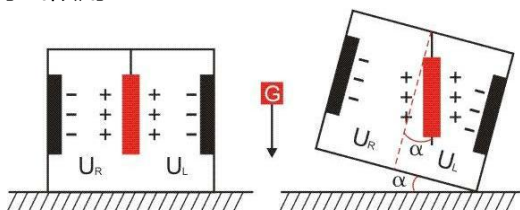
响应时间： 响应时间是传感器在一旦的角度变化时, 传感器输出达到标准值所需的时间。

机械特性

连接器	引线(1.5m)
防护等级	IP67
外壳材质	铝合金磨沙氧化
安装	三颗 M4 螺丝

工作原理

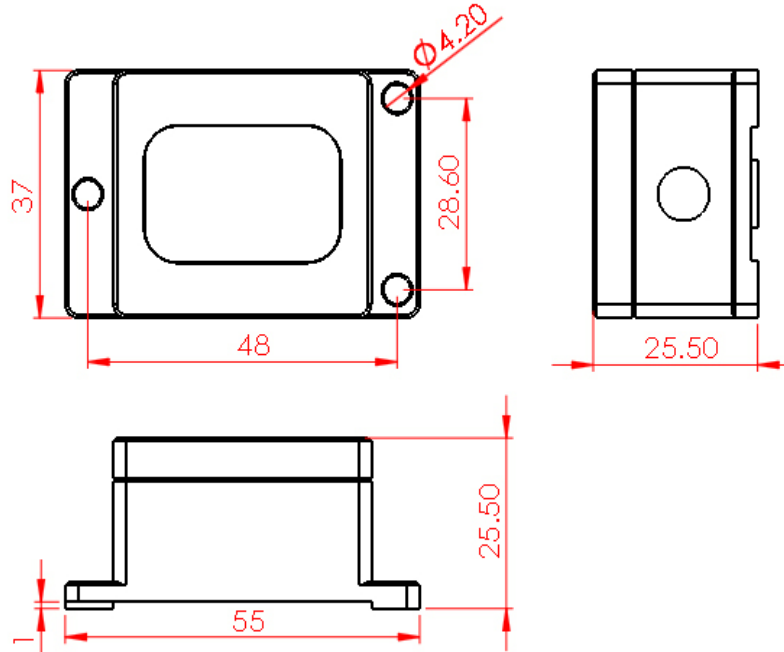
采用欧洲进口核心控制单元, 采用电容微型摆锤原理。利用地球重力原理, 当倾角单元倾斜时, 地球重力在相应的摆锤上会产生重力的分量, 相应的电容量会变化, 通过对电容量处量 放大, 滤波, 转换之后得出倾角。



U_R, U_L 分别为摆锤的左极板和右极板与其各自对应电极间的电压, 当倾角传感器倾斜时, U_R, U_L 会按照一定规律变化, 所以 $\int (U_R, U_L)$ 是关于倾角 α 的函数: $\alpha = \int (U_R, U_L)$

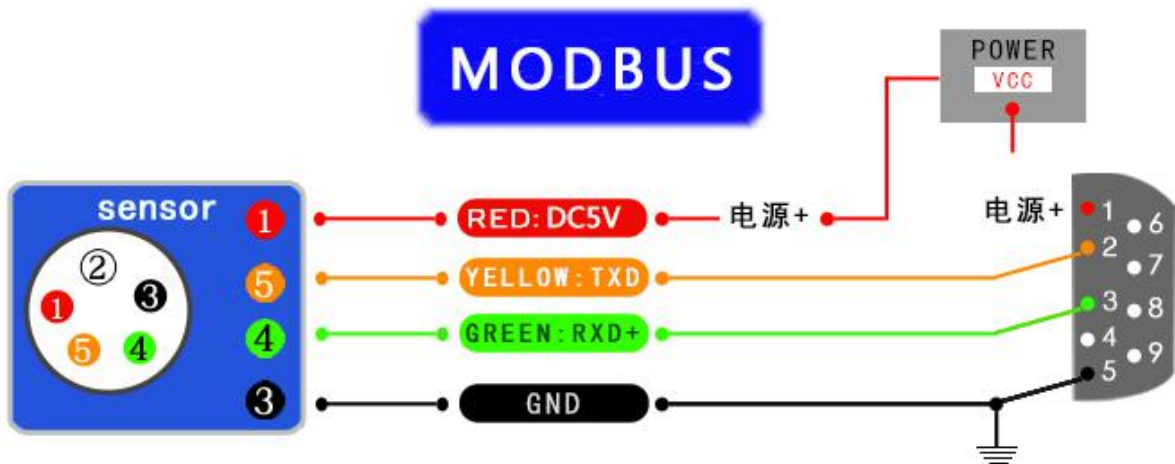
LVT527T-M MODBUS 数字型双轴倾角传感器

产品尺寸图(单位: mm)



默认水平向上安装:安装时应保持传感器安装面与被测目标面平行;安装方式请参考旋转示意图。
 需其他安装方式,参照“产品安装示意图”,订购时备注。

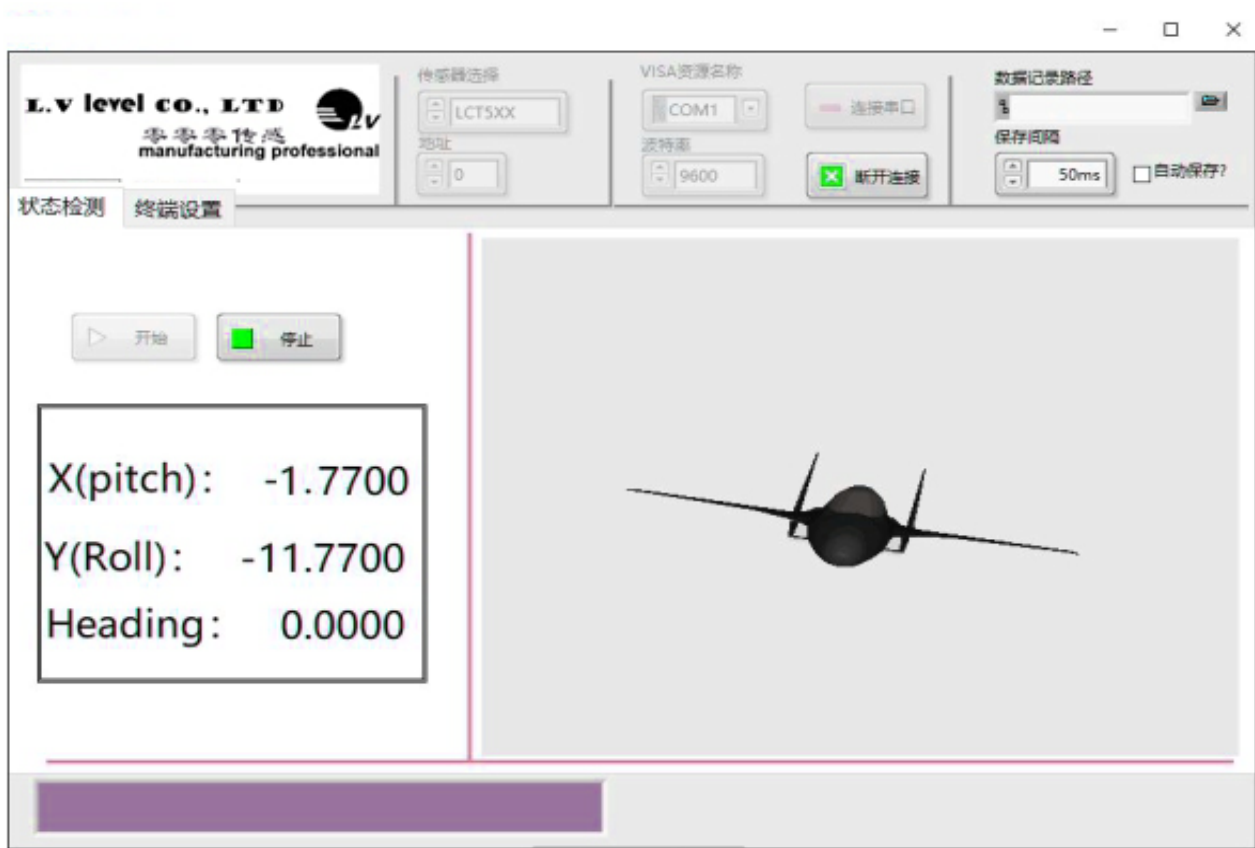
产品电气连接



接口 \ 线色	红	白	黑	绿	黄
MODBUS	VCC	空	GND	RXD	TXD

调试软件

可以在零零传感科技官方网站上下载倾角罗盘调试助手进行初步测试,如果您希望直接访问倾角传感器,可以通过倾角传感器的通信协议和大众版的串口调试助手访问,这样传感器可以方便的集成到您的系统中。



设备型号：选择对应的产品型号

串口端口：选择设备对应的 COM 口

设备地址：填入传感器当前地址码，出厂默认是 01

波特率：选择传感器当前波特率，出厂默认是 9600

状态监测：连接串口，点击开始，进行数据采集

状态设置：对传感器功能参数进行设置

通讯协议

1 数据帧格式：(RTU 模式 8 位数据位，1 位停止位，无校验，默认速率 9600)

地址码	功能码	寄存器地址		寄存器数量		CRC 校验	
		AddrH	AddrL	NumH	NumL	CRC16L	CRC16H
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)

0x01	0x03 读 0x06 写	XX	XX	XX	XX	XX	XX
------	------------------	----	----	----	----	----	----

数据格式: 16 进制

地址码: 出厂默认 0X01 (用户可根据需要设置,最多不超过 0XFF)

功能码: 0x03 读取保持寄存器 0x06 预置单寄存器

寄存校验地址: 需要读写的寄存器起始地址

寄存器数量: 需要读写的寄存器数量

CRC 校验: 地址码、功能码,寄存器起始地址、寄存器数量、CRC 校验,计算机通过专用 CRC16 校验工具、计算得出(注意:当地址码、功能码或者寄存器起始地址变化时,CRC 检校会变化。当您命令改变时请相应改变 CRC 检校。)

注意: 使用 Modbus 串口助手软件,访问传感器通讯时,不需要加上 CRC 校验;普通串口调试助手需要注意,使用前请仔细阅读以下项目:

1) 由于 MODBUS 协议规定两条数据帧之间应至少大于 3.5 个字节时间(如 9600 波特率下,该时间为 $3.5 \times (1/9600) \times 11 = 0.004s$)。但为了留下足够余量,本传感器将此时间提高到大于 10ms,所以请在每条数据帧之间至少留下 10ms 的时间间隔。

主机发送命令 - - 10ms 空闲 - - 从机回覆命令 - - 10ms 空闲 - - 主机发送命令.....

如果用户需要自己实现 CRC16 MODBUS 计算,C 语言程序实现如下,供参考:

```
unsigned short ModBusCRC (unsigned char *ptr,unsigned char size)
{
    unsigned short a,b,tmp,CRC16,V;
    CRC16=0xffff;//CRC 寄存器初始值
    for (a=0;a<size;a++) //N 个字节
    {
        CRC16=*ptr^CRC16;
        for (b=0;b<8;b++) //8 位数据
        {
            tmp=CRC16 & 0x0001;
            CRC16 =CRC16 >>1; //右移一位
```

```

if (tmp)
CRC16=CRC16 ^ 0xa001; //异或多项式
}
*ptr++;
}
V = ((CRC16 & 0x00FF) << 8) | ((CRC16 & 0xFF00) >> 8); //高低字节转换
return V;
}
    
```

例如：“01 06 00 0B 00 02” 的校验码为 “79 C9”

2 命令格式

2.1 读 X 轴角度

发送命令： 01 03 00 01 00 02 95 CB

地址码	功能码	寄存器地址		寄存器数量		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x00	0x01	0x00	0x02	0x95	0xCB

应答命令：

地址码	功能码	字节数	寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	Byte Count (1byte)	DataH (2byte)	DataL (2byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x04	XXXX	XXXX	XX	XX

注：例如应答回复帧：01 03 04 **BD A3 D7 0A** F1 8A，X 轴为寄存器数据的 1-4 字节（寄存器存储 32 位浮点数，标准 IEEE754 标准），其中 1-2 字节为数据高位（高字节），3-4 字节为数据地位（低字节），高字节在前、低字节在后；

MODBUS RTU 标准协议，根据 IEEE754 标准，角度表示方法如下：

X 轴角度（0x**BDA3D70A**）= -0.080000

2.2 读 Y 轴角度

发送命令： 01 03 00 03 00 02 34 0B

地址码	功能码	寄存器地址		寄存器数量		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x00	0x02	0x00	0x02	0x34	0x0B

应答命令：

地址码	功能码	字节数	寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	Byte Count (1byte)	DataH (2byte)	DataL (2byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x04	XXXX	XXXX	XX	XX

注：例如应答回复帧：01 03 04 **3F BD 70 A4** 42 78，Y 轴为寄存器数据的 1-4 字节（寄存器存储 32 位浮点数，标准 IEEE754 标准），其中 1-2 字节为数据高位（高字节），3-4 字节为数据低位（低字节），高字节在前、低字节在后；

MODBUS RTU 标准协议，根据 IEEE754 标准，角度表示方法如下：

$$Y \text{ 轴角度 } (0x3FBD70A4) = 1.48000$$

2.3 读取 X、Y 轴角度

发送命令： 01 03 00 01 00 04 15 C9

地址码	功能码	寄存器地址		寄存器数量		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x00	0x01	0x00	0x04	0x15	0xC9

应答命令：

地址码	功能码	字节数	寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	Byte Count (1byte)	X Angle (4byte)	Y Angle (4byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x08	XXXX XXXX	XXXX XXXX	XX	XX

注: 例如应答回复帧 : 01 03 08 **BD A3 D7 0A 3F BD 70 A4** FE 5D , 寄存器存储 32 位浮点数, 标准 IEEE754 标准, X 轴角度为寄存器数据的 1-4 字节, Y 轴角度为寄存器数据的 5-8 字节; 数据高字节在前、低字节在后;

MODBUS RTU 标准协议, 根据 IEEE754 标准, 数据表示方法如下:

X 轴角度 (0xBDA3D70A) = -0.080000°

Y 轴角度 (0x3FBD70A4) = 1.480000°

2.4 设置相对/绝对零点

发送命令 : 01 06 00 0B 00 01 39 C8

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0B	0x00	0x01	0x39	0xC8

注: 寄存器数据域: 0x0000 绝对零点, 0x0001 相对零点, 发送 01 06 00 0B 00 00 39 C8, 设置绝对零点;

应答命令 :

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0B	0x00	0x01	0x39	0xC8

注: 寄存器数据域: 0x0000 绝对零点, 0x0001 相对零点; 如果设成绝对零点, 则测量角度以出厂设置的零点为基准, 如果设成相对零点, 则测量角度以当前位置为零点基准。

2.5 查询相对/绝对零点

发送命令 : 01 03 00 0B 00 01 F5 C8

地址码	功能码	寄存器地址		寄存器数量		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x00	0x0B	0x00	0x01	0xF5	0xCB

应答命令：

地址码	功能码	字节数	寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	Byte Count (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x03	0x02	XX	XX	XX	XX

注：例如应答回复帧：01 03 02 **00 01** 79 84 ，寄存器数据 0001 表示当前为相对零点测量

2.6 设置通讯速率

发送命令： 01 06 00 0C 00 04 48 0A

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0C	0x00	0x04	0x48	0x0A

应答命令：

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0C	0x00	0x04	0x48	0x0A

注：寄存器数据域 0x0000 表示 2400 0x0001 表示 4800 0x0002 表示 9600 0x0003 表示 19200 ,0x0004 表示 115200 , **默认值为 0X02:9600**。每次变更通讯波特率成功之后，发送保存指令，会以原波特率发送回应答命令，重新上电启动，然后立即改变设备通信波特率

备注：如果需要高频输出，请将波特率设为 115200。

2.7 设置模块地址

发送命令： 01 06 00 0D 00 02 99 C8

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0D	0x00	0x02	0x99	0xC8

2 注意： 传感器默认的地址为 01 。

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x02	0x06	0x00	0x0D	0x00	0x02	0x99	0xCB

1. 如将多个传感器同时连接在一组总线上，例如 MODBUS，则需将每个传感器设置成不同地址，已达到分别控制与回应速度。
2. 如成功更改了新的地址后，后续所有命令与回应数据包中的地址码都换成更改后的新地址码才能生效，不然传感器不会响应命令。
3. XX 模块地址从 00 致 FE 范围

2.8 查询模块地址

发送命令： FF 03 00 0D 00 01 00 17

地址码	功能码	寄存器地址		寄存器数量		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	NumH (1byte)	NumL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0xFF	0x03	0x00	0x0C	0x00	0x01	0x00	0x17

注意： 由于 MODBUS 协议里规定产品必须要知道模块地址才能通信，所以 MODBUS 协议通信时，地址是事先知道的，也就不能查询地址的。本产品的查询地址是利用的自定义的协议来查询 MODBUS 协议的地址 FF，也可根据知道地址码查询

应答命令：

地址码	功能码	字节数	寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	Byte Count (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0xFF	0x03	0x0D	XX	XX	XX	XX

2.9 恢复出厂设置

发送命令： 01 06 00 0E 00 00 E8 09

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0E	0x00	0x00	0xE8	0x09

应答命令：

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0F	0x00	0x00	0xE8	0x09

注意：恢复出厂设置重新上电后生效

2.10 update flash(保存设置)

发送命令： 01 06 00 0F 00 00 B9 C9

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0F	0x00	0x00	0xB9	0xC9

应答命令：

地址码	功能码	寄存器地址		寄存器数据		CRC 校验	
Address (1byte)	Function (1byte)	AddrH (1byte)	AddrL (1byte)	DataH (1byte)	DataL (1byte)	CRC16L (1byte)	CRC16H (1byte)
0x01	0x06	0x00	0x0F	0x00	0x04	0xB9	0xC9

*对于各种参数设置，如果设置完成后不发送保存设置命令，则断电后这些设置都将消失。

附录.IEEE754 换算

1) .单精度浮点数转化至标准 4 字节数

```
//float 转化为 IEEE754 4 个字节 big_endian  
//若编译器采用 little endian 模式, 请先逆转 bdat 数组  
void float2byte(float fdat,unsigned char * bdat)  
{  
    unsigned char i;  
    //获得 float 数据所在 4 个字节地址  
    unsigned char *tmp=(unsigned char *)&fdat;  
    //间接寻址, 获得 float 所在 4 字节地址中的数值  
    for(i=0;i<(sizeof(float));i++)  
        *(bdat+i)=*(tmp+i);  
}
```

2) .4 字节转化至标准单精度浮点函数

```
//IEEE754 4 字节转化为 float big_endian  
//若编译器采用 little endian 模式, 请先逆转 bdat 数组  
float byte2float(unsigned char *bdat)  
{  
    return *((float *)bdat);  
}
```