

叶面温湿度传感器 YT-S260-M



目录

1	技术支持	3
2	产品介绍与背景知识	3
2.1	背景知识	3
2.1.1	叶面湿度对植物的影响.....	3
2.1.2	测量叶面湿度的意义	3
2.2	产品介绍	3
3	传感器接线	4
4	外型尺寸、选型订购	6
4.1	外型尺寸	6
4.2	选型订购	6
5	安装与测量.....	7
6	叶面湿度与输出的换算	7
7	RS485 通信与协议.....	7
7.1	Modbus 通信协议	7
7.2	Modbus 寄存器.....	7
7.3	Modbus 寄存器参数说明.....	9
7.4	Modbus 协议通信样例.....	11
7.4.1	功能号 3 通信样例	11
7.4.2	功能号 4 通信样例	12
7.4.3	功能号 6 通信样例	12
7.4.4	功能号 16 通信样例	13
7.4.5	CRC16 校验算法及例程.....	14
7.5	使用串口调试软件通信	16

1 技术支持

感谢您选择并使用天津一飞冲天科技有限公司的YT-S260-M叶面湿度传感器，此用户手册协助您了解并正确使用传感器。如需订购产品、技术支持、以及产品信息反馈，请通过以下方式联系我们。请在联系时附注设备的购买时间，购买方式，联系人信息，地址以及电话等相关信息，便于我们为您服务。

网址

<http://www.yt-tj.com>

E-Mail

tj_yfct@163.com

电话

022-87786148

传真

022-87786148

2 产品介绍与背景知识

2.1 背景知识

2.1.1 叶面湿度对植物的影响

植物叶表面湿度的高低对叶感染病菌有重要的影响，植物叶面有一定水汽时，很容易受真菌和细菌疾病的感染。在病菌感染期间，如果叶表面完全润湿则有利于病菌侵染。一旦病菌侵染，又会对叶面结构造成破坏。

叶面湿度传感器能够测定叶面上湿度的存在以及持续的时间，以及润湿性能对防治病虫害的农药液滴滞留的影响，以预知疾病的发生，从而对植物或农作物采取相关的保护措施。

2.1.2 测量叶面湿度的意义

叶面湿度传感器是精确农业的发展基础。使用叶面湿度传感器：

- ☑ 了解作物的生长发育情况以及作物生长研究。
- ☑ 预防病虫害发生以及预警。
- ☑ 叶面施肥，喷洒，喷雾以及灌溉控制。

2.2 产品介绍

YT-S260-M 叶面湿度传感器性能稳定灵敏度高，是观测和研究叶面湿度，预防病虫害，以及喷洒喷灌控制的重要工具。该款叶面湿度传感器能够对叶面湿度进行精准的测量，能够监测到叶面的微量水分或冰晶残留。传感器外形采用仿叶片设计，真实模拟叶页面特性，因而能够更准确地反映出叶面环境的情况。它通过仿叶片介质的上表面介电常数的变化，来测量雾气，水气或冰的存在量。与基于电阻测量的传感器不同的是，它不要求着色或使用校准，同时还能提供冰的有效监测。耗电量低，可进行长期不间断监测。

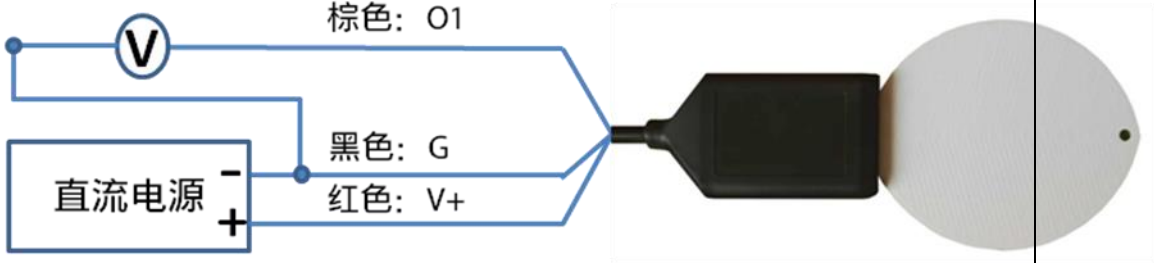
传感器具有以下特点：

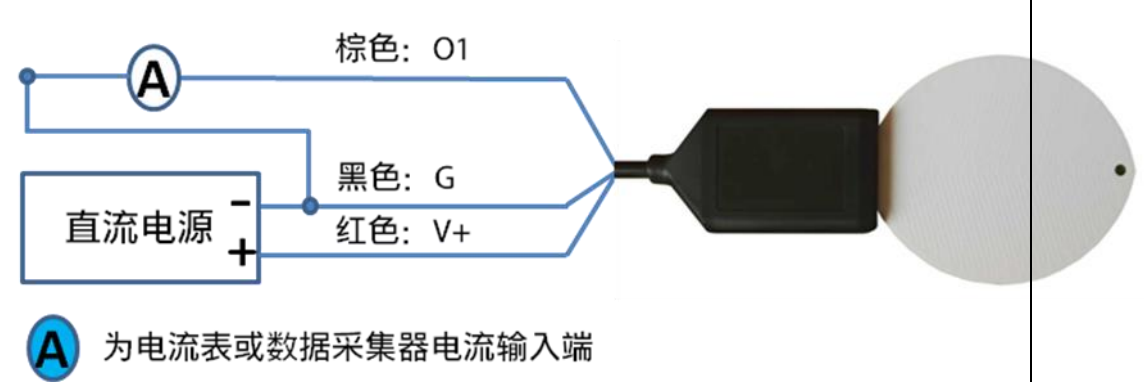
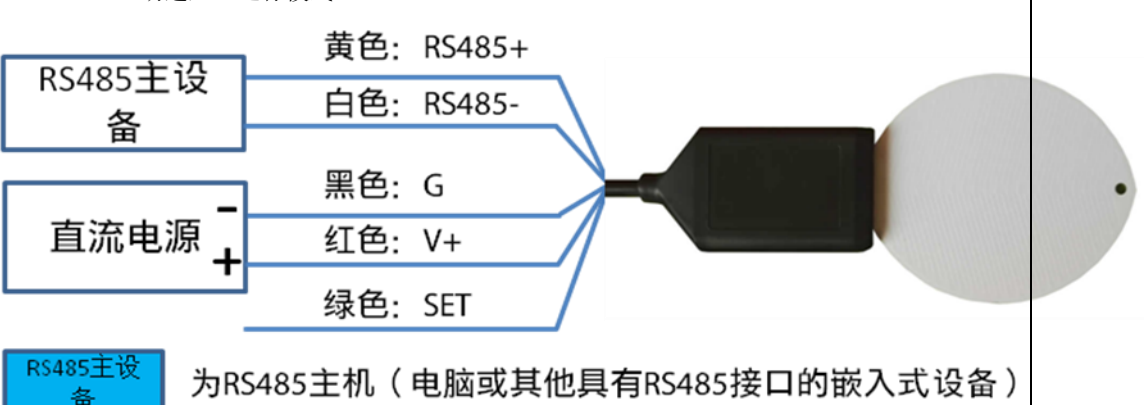
- 模拟叶面结构设计，使传感器真实反映叶面的水分散失过程。
- 可检测雾，结冰，结露以及降雨。
- 防水密封，可直接应用于叶面施肥，喷洒，喷灌场合，可室外使用。
- 精度高，响应快，互换性好，性能可靠。
- 完善的保护电路与多种信号输出接口可选。

技术参数（备注 1）			
信号输出类型	电压输出 0-2V (输出阻抗约 0 欧)	电流输出 4-20mA (负载电阻<500ohm)	RS485接口 Modbus协议
供电电压	3.6-30V/DC 直流	12-30V/DC 直流	3.6-30V/DC 直流
静态功耗	6mA@24V DC 直流	50mA@24V DC 直流 (两电流输出通道均为 20mA)	6mA@24V DC 直流
湿度量程	0-100%，精度±5%		
温度量程	量程：-40~80℃，分辨率：0.1℃，精度：±0.5℃		
防护等级	IP65		
运行环境	-40~85℃		
默认线缆长度	2 米，线缆长度可定制		
外形尺寸	65*13*145mm		

备注 1：可按客户需求进行定制

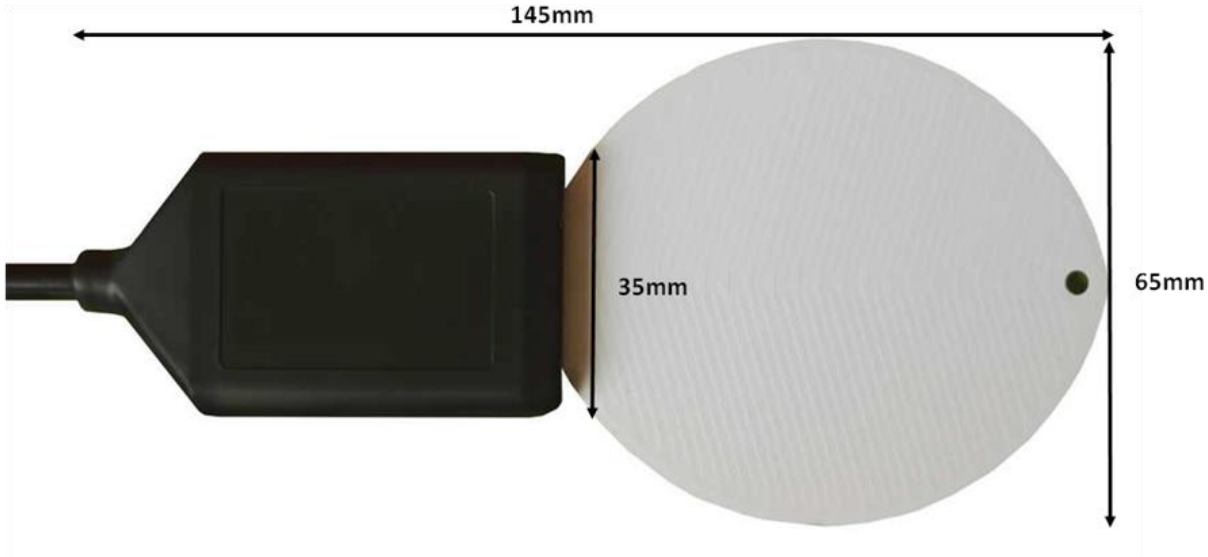
3 传感器接线

型号	接线图
电压输出型	红色(V+)：电源正 黑色(G)：电源地 棕色(O1)：输出信号  V 为电压表或数据采集器电压输入端

电流 输出 型	红色(V+): 电源正 黑色(G): 电源地 棕色(O1): 输出信号  A 为电流表或数据采集器电流输入端
RS485 接口 型 Modbu s 协议	红色(V+): 电源正 黑色(G): 电源地 黄色(T+): RS485+/A/T+ 白色(T-): RS485-/B/T- 绿色(SET): 接 V+ (电源正) 时上电启动模块进入“设置模式”。不连接或者接 G (电源地) 时上电启动进入“运行模式”。  RS485主设备 为RS485主机（电脑或其他具有RS485接口的嵌入式设备） 模块的配置参数如 Modbus 地址，波特率，校验位，通讯协议等是由模块内部的 EEPROM（掉电存储设备）内存储的。有时会忘记这些参数的具体配置而导致不能与模块进行通讯。为了防止这个问题，模块有一特殊的模式称作“设置模式”。当模块以“设置模式”上电启动时，模块会以以下参数进行通讯： - Modbus 地址固定为 0 - 通信配置为 9600, N, 8, 1（9600bps，无校验位，8 个数据位，一个停止位） - 通信协议为 Modbus-RTU EEPROM 中的配置参数不会因为模块进入“设置模式”时而改变，当模块处于“运行模式”时仍会按照 EEPROM 中的这些配置参数进行通讯。

4 外型尺寸、选型订购

4.1 外型尺寸



4.2 选型订购

代码编号	代码	代码说明
代码 1: 产品系列	YT-S260-M	YT-S260-M 叶面湿度传感器
代码 2: 输出信号	A	电压输出0-2V
	B	电流输出4-20mA
	C	RS485接口, Modbus 协议
	D	RS485接口, Modbus 协议 & 电压0-2V输出
	E	RS485接口, Modbus 协议 & 电流4-20mA输出
	F	SDI-12接口
	G	客户订制
代码 3: 线长	002	2米线长
	XXX	客户定制, XXX为任意线长（单位：米）
型号举例： YT-S260-M传感器, RS485接口, Modbus 协议, 5米线长。选型代码为：YT-S260-M - C 005		

5 安装与测量

可悬挂在温室的大棚上，也可以安装在气象站的桅杆上。安装时传感器模拟叶面部分的栅格面向上，大致与周围叶面与地面的夹角一致。

如进行叶面施肥喷洒等检测，可在植物叶面不同的高度安装多个，以确保植物接近地面处的叶面施肥效果。

6 叶面湿度与输出的换算

型号	参数范围	换算关系
电压输出 0-2V	对应叶面湿度 0-100%	叶面湿度=50*电压。如测量到电压为 0.3V，则叶面湿度=50*0.3=15%。
电流输出 4-20mA	对应叶面湿度 0-100%	叶面湿度= 6.25 *(电流-4)。如测量到电流为 6.4mA，则叶面湿度=6.25*(6.4-4)=15%
RS485 接口 Modbus 协议	对应叶面湿度 0-100%	叶面湿度=湿度值寄存器/100。如读取到的数据为 2013，则叶面湿度= 2013/100=20.13%。
	对应温度-40-80℃	温度=温度寄存器值/100。如读取到的数据为 2013，则温度= 2013/100=20.13℃。
客户订制	订制型号的输出请联系技术支持。	

注：公式中电压单位为伏(V)，电流单位为毫安(mA)

7 RS485 通信与协议

7.1 Modbus 通信协议

Modbus 是一种串行通信协议，是多种仪器仪表以及智能传感器在通信接口方面的标准，在智能传感器中有着广泛的应用。Modbus 协议是一个主从架构的协议。有一个主节点，其他使用 Modbus 协议参与通信的节点是从节点。每一个从设备都有一个唯一的设备地址。

YT-S260-M 叶面湿度传感器具有 RS485 接口，支持 Modbus 协议。通讯参数出厂默认值为：波特率 9600bps，一个起始位，8 个数据位，无校验，一个停止位。通讯协议为 Modbus RTU 协议。通讯参数可由设置程序或者 Modbus 命令改变，通信参数改变后需要重新对传感器进行上电方可生效。

7.2 Modbus 寄存器

参数名称	寄存器地址 (16进制/10进制)	参数 类型	Modbus 功能号	参数范围及说明	默认值
温度值 TEMPERATURE	0x0000 /0	INT16 只读	3/4	-4000-8000 对应 -40.00~80.00℃。	N/A
湿度值 WETNESS	0x0001 /1	UINT16 只读	3/4	0-10000 对应 0-100%	N/A

温度单位 TEMPUNIT	0x0020 /32	UINT16 读写	3/6/16	0: 摄氏度℃ 1: 华氏度℉	0
温度校准 TEMPCALIB	0x0021 /33	UINT16 读写	3/6/16	-999~999 对应 -9.99~9.99℃。	0
Modbus 从机地址 (ADDRESS)	0x0200 /512	UINT16 读写	3/6/16	0~255	1
串行通信波特率 (BAUDRATE)	0x0201 /513	UINT16 读写	3/6/16	0~6 0:1200bps 1:2400bps 2:4800bps 3:9600bps 4:19200bps 5:38400bps	3:9600bps
串行通信协议 (PROTOCOL)	0x0202 /514	UINT16 读写	3/6/16	0~1 0:Modbus RTU 1:Modbus ASCII	0:Modbus RTU
串行通信校验位 (PARITY)	0x0203 /515	UINT16 读写	3/6/16	0~2 0:无校验 1:偶校验 2:奇校验	0:无校验
串行通信数据位 (DATABITS)	0x0204 /516	UINT16 读写	3/6/16	1 1:8个数据位	1:8个数据位
串行通信停止位 (STOPBITS)	0x0205 /517	UINT16 读写	3/6/16	0~1 0:1个停止位 1:2个停止位	0:1个停止位
串行通信延时响应 (RESPONSEDELAY)	0x0206 /518	UINT16 读写	3/6/16	0~255对应0~2550毫秒 传感器接受到主机请求 命令后延时一段时间然 后响应。延时时间为设置 值*10毫秒。设置为0时此 功能禁用。	0
串行通信主动输出 时间间隔 (ACTIVEOUTPUTINTERVAL)	0x0207 /519	UINT16 读写	3/6/16	0~255对应0~255秒 不需要主机进行请求,传 感器以固定的时间间隔 自动发送数据。时间间隔 为设置值*1秒。设置为0 时此功能禁用。	0

UINT16:16 位无符号整数寄存器

INT16:16 位有符号整数寄存器

7.3 Modbus 寄存器参数说明

TEMPERATURE —— 温度值		
参数范围	-4000~8000 对应 -40.00~80.00℃	默认值: 无
参数存储	无	

意义: 温度测量值, 负数用补码表示。

举例: 如果返回的值是 0702H (16 进制, 原码), 则第一字节高字节为 07H, 第二字节低字节为 02H, 那么温度测量值为 $(07H \times 256 + 02H) / 100 = 17.94$ 摄氏度。

如果返回的值是 FF05H (16 进制, 补码), 则第一字节高字节为 FFH, 第二字节低字节为 05H, 那么温度测量值为 $((FFH \times 256 + 05H) - FFFFH - 1H) / 100 = (FF05H - FFFFH - 1H) / 100 = -2.51$ 摄氏度。

WETNESS —— 叶面湿度		
参数范围	0~10000 对应 0~100%	默认值: 无
参数存储	无	

意义: 体积含水率测量值。

举例: 如果返回的值是 071DH (16 进制), 则第一字节高字节为 07H, 第二字节低字节为 1DH, 那么测量值为 $(07H \times 256 + 1DH) / 10000 = (7 \times 256 + 29) / 10000 = 18.21\%$ 。代表体积含水率为 18.21%。

TEMPUNIT —— 温度单位		
参数范围	0: 摄氏度℃ 1: 华氏度°F	默认值: 0
参数存储	无	

意义: 温度单位。

TEMPCALIB —— 温度校准值		
参数范围	-999~999 对应 -9.99~9.99℃。	默认值: 0
参数存储	无	

意义: 校准当前温度值。

SLAVEADDR —— Modbus 从机地址		
参数范围	0~255	默认值: 1
参数存储	立即存储	

Modbus 地址, 可设置为 0~255。当模块外部的地址拨码开关设置为地址 0 时, 使用此寄存器的内容作为从机地址。设置后需要重新上电或者使用 RST 命令重新启动模块, 使此地址生效。使用此命令修改模块地址不需要打开机壳即可设置。

BAUDRATE —— 串行通信波特率		
参数范围	0~5 0: 1200bps 1: 2400bps 2: 4800bps 3: 9600bps	默认值: 3

	4: 19200bps 5: 38400bps	
参数存储	立即存储	

PROTOCOL —— 串行通信协议

参数范围	0~1 0:Modbus RTU 1:Modbus ASCII	默认值:0
参数存储	立即存储	

PARITY —— 串行通信校验位

参数范围	0-2 0:无校验 1:偶校验 2:奇校验	默认值:0
参数存储	立即存储	

DATABITS —— 串行通信数据位

参数范围	1 1:8个数据位	默认值:1, 只支持 8 个数据位, 其他无效
参数存储	立即存储	

STOPBITS —— 串行通信停止位

参数范围	0-1 0:1个停止位 1:2个停止位	默认值:0
参数存储	立即存储	

RESPONSEDELAY —— 串行通信延时响应

参数范围	0-255	默认值:0
参数存储	立即存储	

串行通信延时响应在以下情况下使用:当主机发送请求命令后,模块延时(RESPONSEDELAY*10)毫秒,然后将响应数据返回给主机。比如设置 RESPONSEDELAY=5,那么模块延时 5*10=50 毫秒后响应主机请求。设置为 0 时为无延时立即响应。此命令主要应用于主机从 RS485 发送状态切换为接收状态时速度比较慢的场合。

ACTIVEOUTPUTINTERVAL —— 串行通信主动输出时间间隔

参数范围	0-255	默认值:0
参数存储	立即存储	

串行通信主动输出时间间隔在以下情况下使用:主机不需要发送请求命令,模块主动输出响应数据,输出间隔为 ACTIVEOUTPUTINTERVAL 秒,比如设置 ACTIVEOUTPUTINTERVAL=5,那么模块每 5 秒按照设置的通信协议输出数据。设置为 0 时主动输出无效,需主机请求后方可响应。此命令主要应用于 GPRS 等无线传输时,需要终端节点主动发送数据的场合。

注意:当设置为主动输出数据时, RS485 总线上只能连接一个模块, 以避免总线数据冲突。

7.4 Modbus 协议通信样例

以下说明中, 0x 开头或者 H 结尾的数据为 16 进制数据。Modbus 协议有两种常用寄存器类型:

- (1) 保持寄存器, 存储数据掉电不丢失, 是可读可写的。通常用功能号 3 (0x03) 读取, 用功能号 6 (0x06) 或者 16 (0x10) 写入。
- (2) 输入寄存器, 用来存储一些只读的物理量, 比如温度值, 是只读的。通常用功能号 4 (0x04) 读取。

7.4.1 功能号 3 通信样例

通用请求格式: AA 03 RRRR NNNN CCCC

AA	1 字节	设备地址, 范围 0-255
03	1 字节	功能号为 3
RRRR	2 字节	起始寄存器地址, 高字节在前
NNNN	2 字节	要读取的寄存器数量 N, 高字节在前
CCCC	2 字节	CRC 校验

通用响应格式: AA 03 MM VV0 VV1 VV2 VV3... CCCC

AA	1 字节	设备地址, 范围 0-255
03	1 字节	功能号为 3
MM	1 字节	返回寄存器值的数据字节数量
VV0, VV1	2 字节	返回的第一个寄存器值
VV2, VV3	2 字节	返回的第二个寄存器值
...	...	返回的第 N 个寄存器值 (N=MM/2)
CCCC	2 字节	CRC 校验

举例: 以读寄存器 0x0200-0x0201, 即从机地址以及波特率为例

请求: 01 03 0200 0002 C5B3

设备地址	1 字节	0x01
功能号	1 字节	0x03
起始寄存器地址	2 字节	0x0200
寄存器数量	2 字节	0x0002
校验	2 字节	0xC5B3

响应: 01 03 04 00 01 00 03 EB F2

设备地址	1 字节	0x01
功能号	1 字节	0x03
有效字节数	1 字节	0x04
从机地址寄存器值	2 字节	0x00 (从机地址高字节)
		0x01 (从机地址低字节)
波特率寄存器值	2 字节	0x00 (波特率高字节)

		0x03 (波特率低字节)
校验	2 字节	0xEBF2

7.4.2 功能号 4 通信样例

通用请求格式: AA 04 RRRR NNNN CCCC

AA	1 字节	设备地址, 范围 0-255
04	1 字节	功能号为 4
RRRR	2 字节	起始寄存器地址, 高字节在前
NNNN	2 字节	要读取的寄存器数量 N, 高字节在前
CCCC	2 字节	CRC 校验

通用响应格式: AA 04 MM VV0 VV1 VV2 VV3... CCCC

AA	1 字节	设备地址, 范围 0-255
04	1 字节	功能号为 4
MM	1 字节	返回寄存器值的数据字节数量
VV0, VV1	2 字节	返回的第一个寄存器值
VV2, VV3	2 字节	返回的第二个寄存器值
...	...	返回的第 N 个寄存器值 (N=MM/2)
CCCC	2 字节	CRC 校验

举例: 以读寄存器 0x0000-0x0001, 即读取温度, 湿度

请求: 01 04 0000 0002 71CB

设备地址	1 字节	0x01
功能号	1 字节	0x04
起始寄存器地址	2 字节	0x0000
寄存器数量	2 字节	0x0002
校验	2 字节	0x71CB

响应: 01 04 04 0A55 004E 6878

设备地址	1 字节	0x01
功能号	1 字节	0x04
有效字节数	1 字节	0x04
温度寄存器值	2 字节	0x0A
		0x55
湿度寄存器值	2 字节	0x00
		0x4E
校验	2 字节	0x6878

7.4.3 功能号 6 通信样例

通用请求格式: AA 06 RRRR VVVV CCCC

AA	1 字节	设备地址，范围 0-255
06	1 字节	功能号为 6
RRRR	2 字节	寄存器地址，高字节在前
VVVV	2 字节	要写入寄存器的数值，高字节在前
CCCC	2 字节	CRC 校验

通用响应格式: AA 06 RRRR VVVV CCCC

AA	1 字节	设备地址，范围 0-255
06	1 字节	功能号为 6
RRRR	2 字节	寄存器地址，高字节在前
VVVV	2 字节	要写入寄存器的数值，高字节在前
CCCC	2 字节	CRC 校验

举例：以写寄存器 0x0021，即温度单位为华氏度为例

请求: 01 06 0021 0001 1800

设备地址	1 字节	0x01
功能号	1 字节	0x06
起始寄存器地址	2 字节	0x0021
寄存器数量	2 字节	0x0001
校验	2 字节	0x1800

响应: 01 06 0021 0001 1800

设备地址	1 字节	0x01
功能号	1 字节	0x06
起始寄存器地址	2 字节	0x0021
寄存器数量	2 字节	0x0001
校验	2 字节	0x1800

7.4.4 功能号 16 通信样例

通用请求格式: AA 10 RRRR NNNN MM VVVV1 VVVV2 ...CCCC

AA	1 字节	设备地址，范围 0-255
10(16 进制)	1 字节	功能号为 16 (十进制)
RRRR	2 字节	起始寄存器地址，高字节在前
NNNN	2 字节	要读取的寄存器数量 N，高字节在前
MM	1 字节	要写入寄存器的数值的字节个数
VVVV1	2 字节	要写入第一个寄存器的数值，高字节在前
VVVV2	2 字节	要写入第二个寄存器的数值，高字节在前
...	...	要写入第 N 个寄存器的数值，高字节在前 N=MM/2

CCCC	2 字节	CRC 校验
------	------	--------

通用响应格式: AA 10 RRRR NNNN CCCC

AA	1 字节	设备地址, 范围 0-255
10(16 进制)	1 字节	功能号为 16 (十进制)
RRRR	2 字节	起始寄存器地址, 高字节在前
NNNN	2 字节	要读取的寄存器数量 N, 高字节在前
CCCC	2 字节	CRC 校验

举例: 以写寄存器 0x0200-0x0201, 即设置从机地址为 1, 波特率为 19200bps 为例

请求: 01 10 0200 0002 04 0001 0004 BACC

0x01	1 字节	设备地址
0x10(16 进制)	1 字节	功能号为 16 (十进制)
0x0200	2 字节	起始寄存器地址, 高字节在前
0x0002	2 字节	要读取的寄存器数量 N, 高字节在前
0x04	1 字节	要写入寄存器的数值的字节个数
0x0001	2 字节	要写如的从站地址寄存器值为 1
0x0004	2 字节	要写如的波特率寄存器值为 4
0xBACC	2 字节	CRC 校验

响应: 01 10 0200 0002 4070

0x01	1 字节	设备地址
0x10(16 进制)	1 字节	功能号为 16 (十进制)
0x0200	2 字节	起始寄存器地址, 高字节在前
0x0002	2 字节	要读取的寄存器数量 N, 高字节在前
0x4070	2 字节	CRC 校验

7.4.5 CRC16 校验算法及例程

例程:

```
//-----
//CRC 计算 C51 语言函数如下
//输入参数 1: snd, 待校验的字节数组名
//输入参数 2: num, 待校验的字节总数
//函数返回值: 校验和
//-----
unsigned int calc_crc16 (unsigned char *snd, unsigned char num)
{
    unsigned char i, j;
    unsigned int c, crc=0xFFFF;
```

```
for(i = 0; i < num; i ++)  
{  
    c = snd[i] & 0x00FF;  
    crc ^= c;  
    for(j = 0;j < 8; j ++)  
    {  
        if (crc & 0x0001)  
        {  
            crc>>=1;  
            crc^=0xA001;  
        }  
        else  
        {  
            crc>>=1;  
        }  
    }  
}  
return(crc);  
}
```

举例：以读寄存器 0x0000-0x0001，即读取温度，湿度

主机请求：01 04 0000 0002 71CB

设备地址	1 字节	0x01
功能号	1 字节	0x04
起始寄存器地址	2 字节	0x0000
寄存器数量	2 字节	0x0002
校验	2 字节	0x71CB

当主机需要发送数据给传感器以前，将需要进行发送校验的数据存储到 snd 数组中（01 04 00 00 00 02 共 6 个字节），其中 num=6

伪代码如下：

```
unsigned char request[8]={01, 04, 00, 00, 00, 02, 00, 00}; //最后两个 00, 00 是 CRC 校验  
unsigned char num=6; //计算数组前 6 个字节的 CRC 校验  
unsigned int crc16=0;  
crc16= calc_crc16 (request, num);  
request[6]= crc16%256; //把 crc 校验存储到要发送的数组中  
request[7]= crc16/256;  
CommPort.Send(request, 8); //通过串口发送数据
```

传感器响应：01 04 04 0A55 004E 6878（共9字节）

设备地址	1 字节	0x01
功能号	1 字节	0x04
有效字节数	1 字节	0x04
温度寄存器值	2 字节	0x0A

		0x55
湿度寄存器值	2 字节	0x00
		0x4E
校验	2 字节	0x6878

当主机接收到传感器返回的 9 个字节数据后，进行以下 crc 计算操作，其中 num=9

伪代码如下：

```
unsigned char response[9]={ 01 04 04 0A 55 00 4E 68 78}; //最后两个字节是传感器返回的 CRC 校验
unsigned char num=9; //计算整个返回的 9 个字节的 CRC 校验
unsigned int crc16=0;
crc16= calc_crc16 (response, num);
if(crc16==0)
{
    //crc 校验正确，可以使用返回的数据
}
else
{
    //crc 校验错误，不能使用返回的数据
}
```

得到返回结果为0时那么校验成功，如果校验失败返回为非零值。如果校验不成功，说明传输过程发生错误，应放弃此次采集到的数据，重新采集。

校验成功后，使用以下公式计算温度（负数以补码表示）和湿度，H结尾的为16进制数据：

温度= (0AH*256+55H) /100=2645/100=26.45 ℃

体积含水率= (00H*256+4EH) /100=78/100=0.78%

7.5 使用串口调试软件通信

用户可使用任意一款串口调试软件与传感器进行通信，通信时需注意，选择正确的串口，波特率，以及其他串口通信参数，需要发送和接收的数据均要以16进制进行传输以及显示。



联系方式:

电话: 022-87786148

邮箱: tj_yfct@163.com

网址: www.yt-tj.com

地址: 天津市南开区金平路 10 号

扫描二维码, 关注“农租宝”, 查看更多精彩内容。

