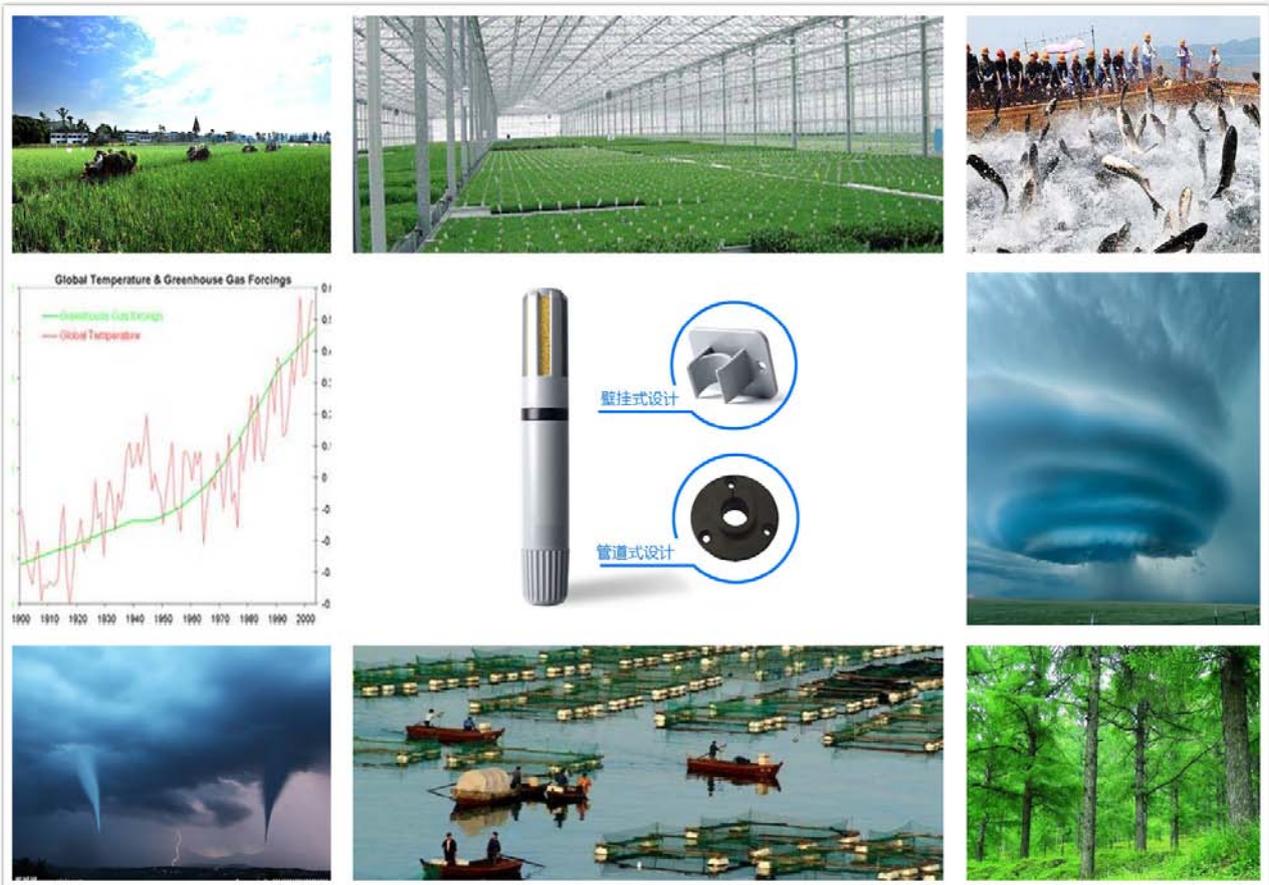


DigiTHP温湿度气压变送器 用户手册



目 录

1	技术支持	3
2	产品介绍	4
3	传感器接线	5
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	8
6	温湿度与输出的换算	9
7	RS485 通信与协议	10
7.1	Modbus 通信协议	10
7.2	Modbus 寄存器	10
7.3	Modbus 寄存器参数说明	11
7.4	Modbus 协议通信样例	14
7.4.1	功能号 3 通信样例	14
7.4.2	功能号 4 通信样例	16
7.4.3	功能号 6 通信样例	17
7.4.4	功能号 16 通信样例	18
7.4.5	CRC16 校验算法及例程	19
7.5	使用串口调试软件通信	21
8	用户设置软件	22
8.1	软件安装与启动	22
8.2	运行设置软件	23

1 技术支持

感谢您选择并使用大连哲勤科技有限公司的产品，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.infwin.com>

E-Mail

infwin@163.com

电话

+86-411-66831953, 4000-511-521

传真

+86-411-82388125

版本控制

日期	版本号	说明	完成人
2016-12-12	V1.0	创建	fg49597

2 产品介绍

- 集成瑞士SENSIRION®高精度温湿度传感器，提供14Bits温度测量以及12Bits湿度测量。
- 集成博世BOSCH®高精度气压传感器，提供16Bits气压测量。
- 集成温度、湿度、露点、气压测量于一体。
- 湿度测量范围0~100%RH。温度测量范围：-40° C~+80° C。气压测量范围300-1100mbar。
- 高精度温湿度气压测量最高可达到±1.8%RH，±0.2°C，±0.15mbar（@25°C）。
- 传感器全标定线性输出，无须标定即可互换使用。
- RS485输出，ModbusRTU协议。
- 可选3.6V~24V或2.7-16V宽范围直流供电。
- 墙面壁挂安装或管道插入法兰盘安装。
- 小尺寸，安装简单，可作为系统拓展模块集成于现有系统中。
- 良好的长期稳定性、高可靠性以及性价比。

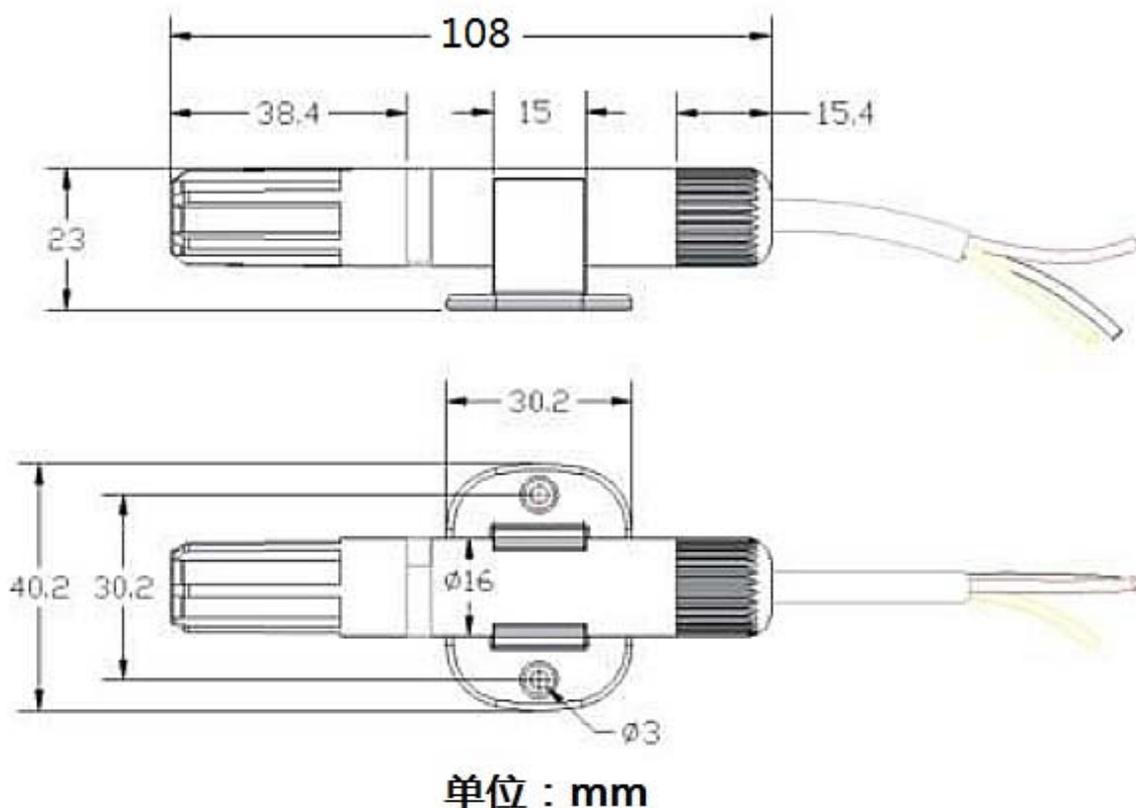
技术参数	
信号输出类型	RS485接口 Modbus协议
供电电压	3.6-24V/DC 或 2.7-16V/DC 直流
静态功耗	4mA@24V DC 直流
湿度量程	量程：0-100% 分辨率：0.01% 精度：最高±1.8% RH
温度量程	量程：-40~80°C 分辨率：0.01°C 精度：最高±0.2°C
气压量程	300-1100mbar 分辨率：0.1mbar 精度：±0.15mbar(相对精度)，±1mbar(绝对精度)
防护等级	IP56
运行环境	-40~85°C
安装方式	墙壁安装或管道安装
默认线缆长度	2米，可定制
连接方式	预装冷压端子
外形尺寸	见本说明书外形尺寸图

3 传感器接线

型号	接线图
RS485 接口型 Modbus 协议	<p> 红色 (V+)：电源正 黑色 (G)：电源地 黄色 (T+)：RS485+/A/T+ 白色 (T-)：RS485-/B/T- </p> <p> RS485主设备 为RS485主机（电脑或其他具有RS485接口的嵌入式设备） </p> <p> 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，模块有一按键，按下三秒后模块内部指示灯熄灭，松开按键，则模块所有参数恢复以下出厂设置： </p> <ol style="list-style-type: none"> 1. Modbus 地址为 1 2. 通信配置为 9600, N, 8, 1（9600bps，无校验位，8 个数据位，一个停止位） 3. 通信协议为 Modbus-RTU

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	DigiTHP	DigiTHP 温湿度气压变送器
代码 2: 测量参数	A	温度、湿度、露点、气压测量
代码 3: 温度量程	A	-40-80℃
代码 4: 湿度量程	A	0-100%RH
代码 5: 湿度量程	A	300-1100mbar
代码 6: 供电电压	A	3.6-24V 直流
	B	2.7-16V 直流
代码 7: 输出信号	A	RS485 输出, Modbus 协议
	B	SDI-12 接口

	C	客户订制
代码 8: 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: DigiTHP温湿度气压变送器, 温度、湿度、露点、气压测量, 温度量程-40-80℃, 湿度量程0-100%, 气压量程300-1100mbar供电电压3.6-30V, RS485接口, Modbus协议, 5米线长。选型代码为: DigiTHP - A A A A A A 005		

5 安装与测量

壁挂安装或法兰盘管道安装。



壁挂式设计



管道式设计



6 温湿度与输出的换算

型号	参数范围	换算关系
RS485 接口 Modbus 协议	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℃。
	对应湿度 0-100%	湿度=湿度寄存器值/100。如读取到的数据为2013，则湿度= 2013/100=20.13%。
	对应露点-40-80℃	露点=露点寄存器值/100。如读取到的数据为2013，则露点= 2013/100=20.13℃。
	对应气压 300-1100mbar	气压=气压寄存器值/100。如读取到的数据为10023，则气压= 10023/10=1002.3mbar。
客户订制	订制型号的输出请联系技术支持。	

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。

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 HUMIDITY	0x0001 /1	UINT16 只读	3/4	0-10000 对应 0-100%	N/A
露点值 DEWPOINT	0x0002 /2	UINT16 只读	3/4	-4000-8000 对应 -40.00~80.00℃。	N/A
气压 AIRPRESSURE	0x0003 /3	UINT16 只读	3/4	3000-11000 对应 300.0-1100.0mbar	N/A
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:Modbus RTU	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	0-250对应0-2500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时禁用此功能。	0
串行通信主动输出时间间隔 (ACTIVEOUTPUTINTERVAL)	0x0207 /519	UINT16 读写	3/6/16	0-250对应0-250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时禁用此功能。	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 $(07H * 256 + 02H) / 100 = 17.94$ 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 $((FFH * 256 + 05H) - FFFFH - 1H) / 100 = (FF05H - FFFFH - 1H) / 100 = -2.51$ 摄氏度。

HUMIDITY --- 湿度值		
参数范围	0-10000对应0-100%	默认值：无
参数存储	无	

意义：湿度测量值。

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 $(07H * 256 + 1DH) / 10000 = (7 * 256 + 29) / 10000 = 18.21\%$ 。代表体积含水率为 18.21%

DEWPOINT --- 露点值		
参数范围	-4000-8000 对应 -40.00~80.00℃	默认值：无
参数存储	无	

意义：露点温度值，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 $(07H * 256 + 02H) / 100 = 17.94$ 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 $((FFH * 256 + 05H) - FFFFH - 1H) / 100 = (FF05H - FFFFH - 1H) / 100 = -2.51$ 摄氏度。

AIRPRESSURE --- 气压值		
参数范围	3000-11000对应300.0-1100.0mbar	默认值：无
参数存储	无	

意义：露点温度值，负数用补码表示。

举例：如果返回的值是 2827H (16 进制，原码)，则第一字节高字节为 28H，第二字节低字节为 27H，那么温度测量值为 $(28H * 256 + 27H) / 10 = 1027.9$ mbar。

ADDRESS --- Modbus 从机地址		
参数范围	0-255	默认值：无
参数存储	立即存储	

设置后需要重新上电使此参数生效。

BAUDRATE --- 串行通信波特率		
----------------------	--	--

参数范围	0-5 0:1200bps 1:2400bps 2:4800bps 3:9600bps 4:19200bps 5:38400bps	默认值:3
参数存储	立即存储	

设置后需要重新上电使此参数生效。

PROTOCOL —— 串行通信协议		
参数范围	0:Modbus RTU	默认值:0
参数存储	立即存储	

设置后需要重新上电使此参数生效。

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	

设置后需要重新上电使此参数生效。

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	

设置后需要重新上电使此参数生效。

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	

设置后需要重新上电使此参数生效。

RESPONSEDELAY --- 串行通信延时响应		
参数范围	0-250	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。设置后需要重新上电使此参数生效。

ACTIVEOUTPUTINTERVAL --- 串行通信主动输出时间间隔		
参数范围	0-250	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。设置后需要重新上电使此参数生效。

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 0x0000-0x0003，即读取温度，湿度，露点值，气压

请求：01 04 0000 0004 B00B

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	0x0004
校验	2 字节	0xF1C9

响应：01 04 08 08 42 0A A3 00 9B 28 27 AC E9

设备地址	1 字节	0x01
功能号	1 字节	0x04
有效字节数	1 字节	0x08
温度寄存器值	2 字节	0x08
		0x42
湿度寄存器值	2 字节	0x0A
		0xA3
露点寄存器值	2 字节	0x00

		0x9B
气压寄存器值	2 字节	0x28
		0x27
校验	2 字节	0xACE9

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 0x0200，即修改从机地址为 2 为例

请求：01 06 0200 0002 09B3

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	0x0200
寄存器值	2 字节	0x0002
校验	2 字节	0x09B3

响应：01 06 0200 0002 09B3

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	0x0200
寄存器值	2 字节	0x0002

校验	2 字节	0x09B3
----	------	--------

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 0x0000-0x0003，即读取温度，含水率，露点值，气压值

主机请求：01 04 0000 0004 F1C9 （8 个字节）

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	0x0004
校验	2 字节	0xF1C9

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 04 共 6 个字节），其中 num=6

伪代码如下：

```

unsigned char request[8]={01,04,00,00,00,04,00,00}; //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 08 08 4B 0A A3 00 A2 28 27 E5 E4 （13 个字节）

设备地址	1 字节	0x01
功能号	1 字节	0x04
有效字节数	1 字节	0x08
温度寄存器值	2 字节	0x08
		0x4B
湿度寄存器值	2 字节	0x0A
		0xA3
露点寄存器值	2 字节	0x00
		0xA2
气压寄存器值		0x28
		0x27

校验	2 字节	0xE5E4
----	------	--------

当主机接收到传感器返回的 13 个字节数据后，进行以下 crc 计算操作，其中 num=13

伪代码如下：

```
unsigned char response[11]={ 01 04 08 08 4B 0A A3 00 A2 28 27 E5 E4}; //最后两个字
节是传感器返回的 CRC 校验
unsigned char num=13; //计算整个返回的 13 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）湿度和露点，H结尾的为16进制数据：

温度= (08H*256+4BH) /100=2123/100=21. 23℃

湿度= (0AH*256+A3H) /100=2723/100=27. 23%

露点= (00H*256+A2H) /100=162/100 =1. 62℃

气压= (28H*256+27H) /100=10279/100 =1027. 9mbar

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。



8 用户设置软件

8.1 软件安装与启动

(1) SensorOneSet 设置软件基于 Dotnet Framework 开发，安装前需先安装 Dotnet Framework 3.5 以上版本。方可运行。如果电脑没有安装微软 DotNetFramework3.5SP1 的,请先下载完整安装包：<http://www.microsoft.com/zh-cn/download/details.aspx?id=25150>

(2) 安装 Dotnet Framework 完成后，可点击“Install.SensorOneSet.msi”进行程序安装。

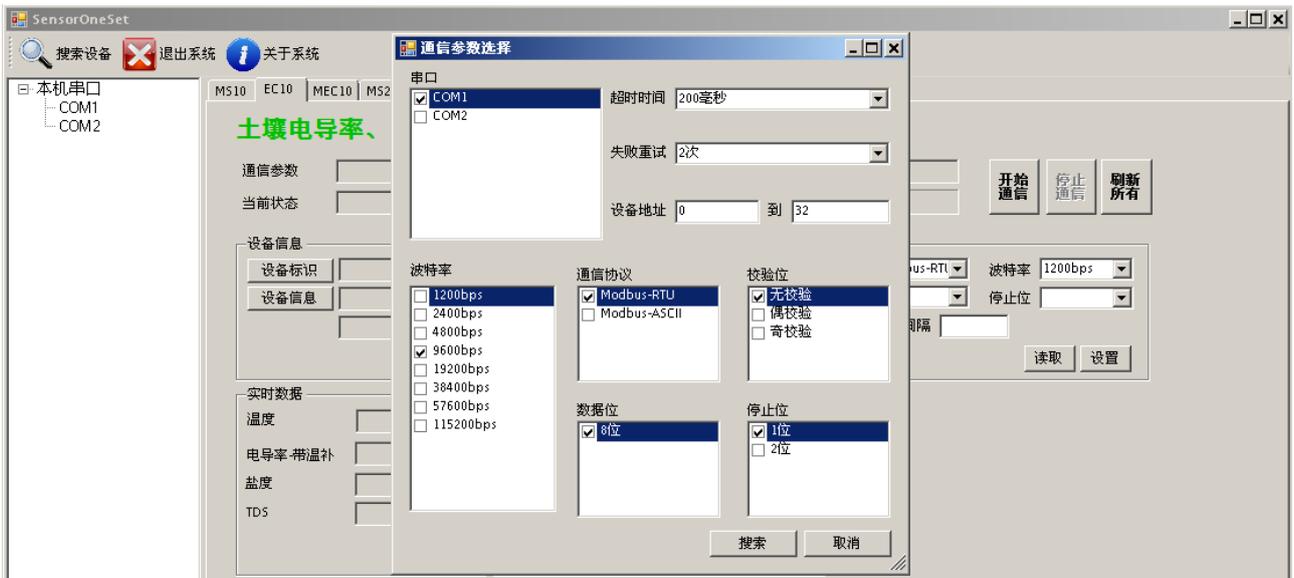
名称	修改日期	类型	大小
Install.SensorOneSet.msi	2016-04-23 12:59	Windows Installer ...	976 KB
setup.exe	2016-04-23 12:59	应用程序	483 KB

(3) 从开始菜单中启动“SensorOneSet 用户设置程序”，启动如下画面。

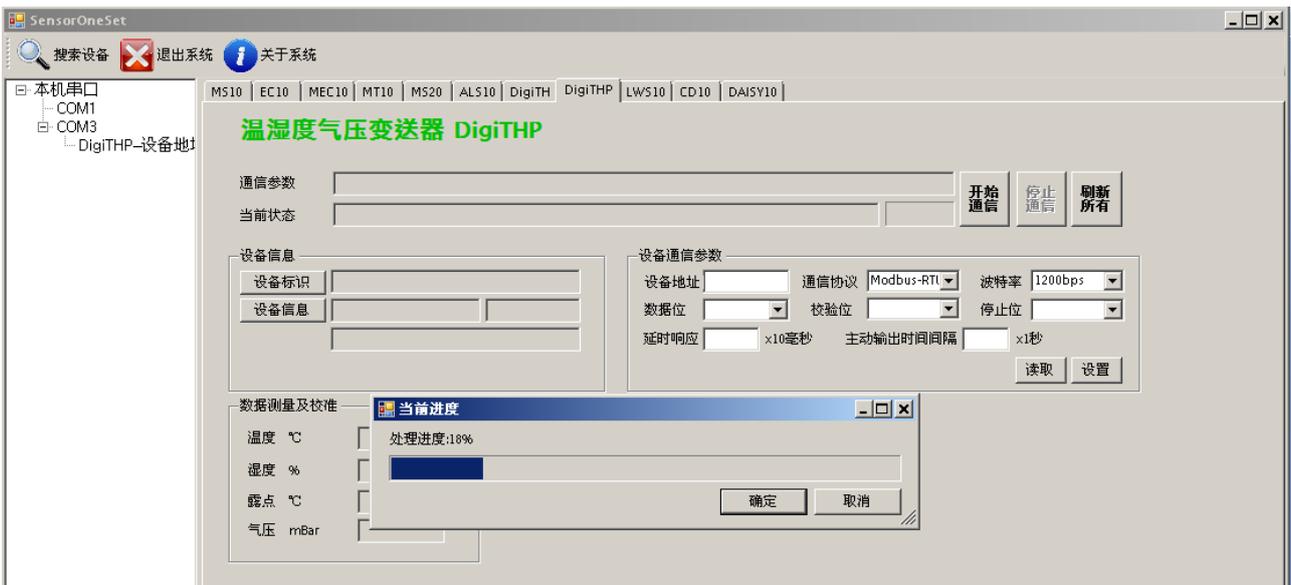


8.2 运行设置软件

(1) 点击工具栏中的“搜索设备”按钮，弹出“搜索在线设备-选择搜索参数”对话框。



(2) 在“通信参数选择”对话框中选择合适的通讯参数。并点击“搜索”按钮。搜索到的设备会列在相应的串口下。如下图。

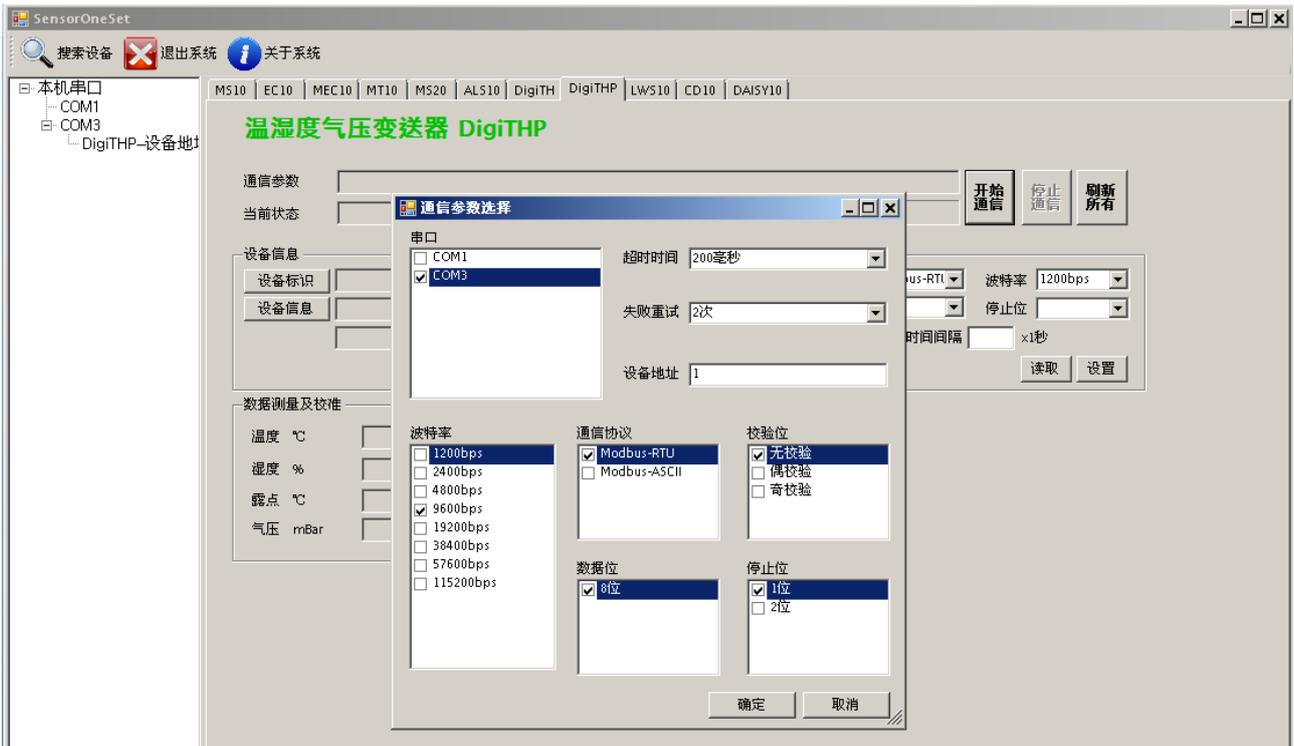


(3) 双击窗口左侧串口下列出的设备“DigiTHP-设备地址……”，其通讯参数会自动列到右侧的“电脑通讯设置”中。点击右侧的“开始”按钮，软件开始于模块进行通讯。



(4) 如需再次搜索模块，请先点击“停止”按钮，然后再次进行搜索。

(5) 如知道传感器的具体通信参数，也可以在 DigiTHP 页面直接点击“开始通信”，弹出“通信参数选择”对话框，设置通信参数等信息后，点“确定”关闭对话框后软件将与传感器进行通信。



(6) 通信成功后，用户可修改各参数的值。

