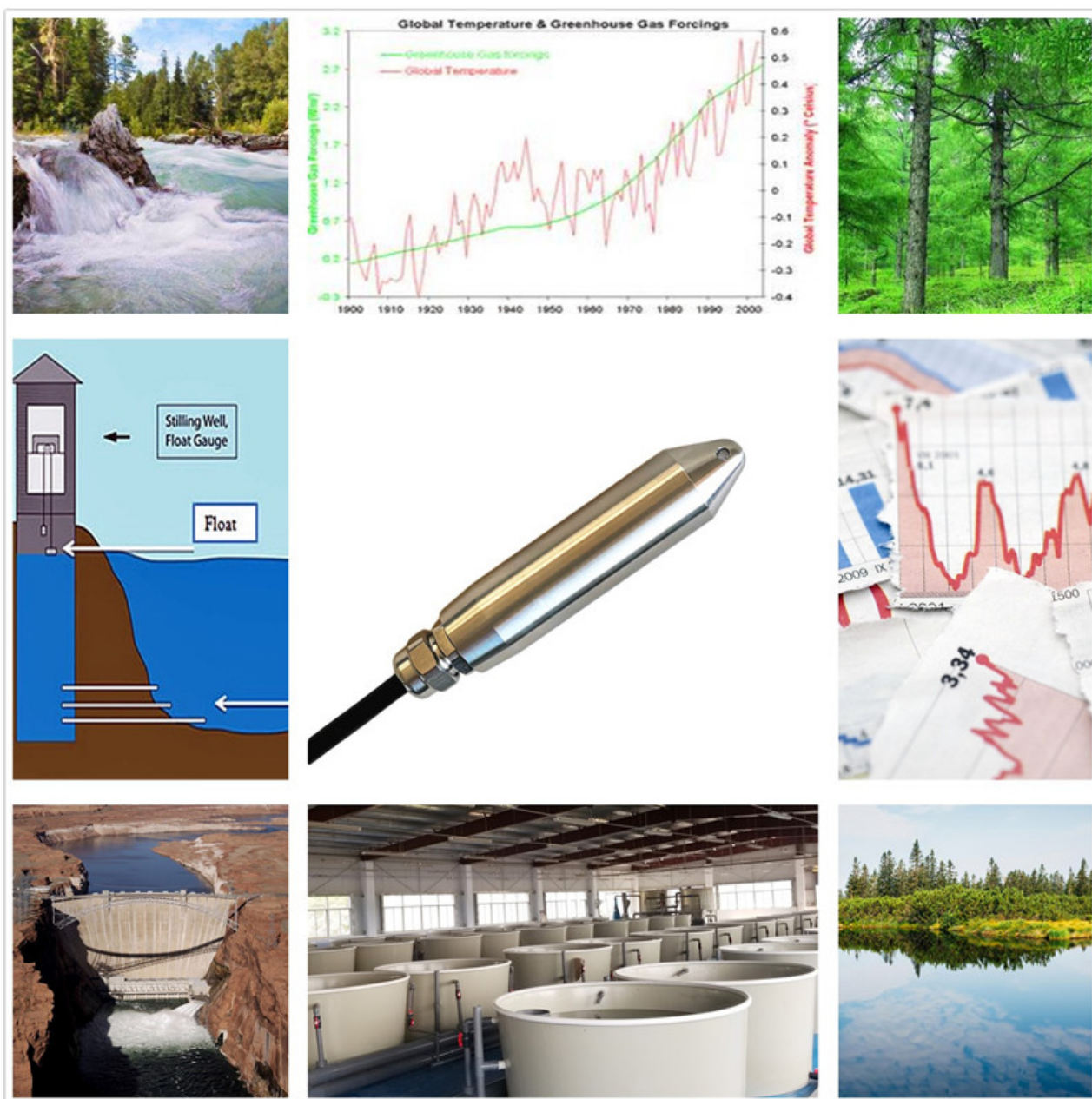


DigiTEMP

坚固型数字输出温度传感器（SDI-12接口）

坚固型数字输出温度传感器（RS485接口）

用户手册



目 录

1	技术支持	3
2	产品介绍	4
2.1	产品介绍	4
3	传感器接线.....	6
3.1	SDI-12 接口.....	6
3.2	RS485 接口	7
4	外型尺寸、选型订购	8
4.1	外型尺寸	8
4.2	选型订购	8
5	安装与测量.....	9
6	SDI-12 数据通信.....	10
6.1	SDI-12 接口	10
6.1.1	电气标准	10
6.1.2	协议解析	10
7	RS485 数据通信.....	14
7.1	Modbus 通信协议	14
7.2	Modbus 寄存器.....	14
7.3	Modbus 寄存器参数说明	16
7.4	Modbus 协议通信样例	18
7.4.1	功能号 3 通信样例.....	18
7.4.2	功能号 4 通信样例.....	19
7.4.3	功能号 6 通信样例.....	20
7.4.4	功能号 16 通信样例.....	21
7.4.5	CRC16 校验算法及例程	22
7.5	使用串口调试软件通信	25
7.6	用户设置软件.....	25
附录	26
版权与商标		26
文档控制		26

1 技术支持

感谢您选择并使用我公司产品，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.cn>

E-Mail

infwin@163.com

电话

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

传真

+86-411-66831953

2 产品介绍

2.1 产品介绍

DigiTEMP 为高精度数字输出的坚固型温度传感器，适用于水、土壤以及空气等介质的温度测量。传感器具有全密封结构的不锈钢壳体，可以在严酷环境下使用。其小巧的尺寸可以穿过直径 2.54cm（1 英寸）、半径 20.32cm（8 英寸）的管道，同时在传感器前端设有线孔，用于增加配重或者进行管道穿线。

传感器支持 SDI-12 接口或 RS485 接口（Modbus-RTU 协议），兼容多种支持 SDI-12 以及 RS485 通信的数据采集器，进行远距离多点监测与记录。

功能特点

- 集成高精度温度传感器
- 具有浪涌保护的 SDI-12 或 RS485 通信接口
- 坚固的防水结构以及耐候性线缆
- 不锈钢壳体以及用于增加配重或者进行管道穿线的线孔
- 低功耗设备可用于电池供电的系统
- 电源反向保护与抗雷击保护
- ODM/OEM 服务



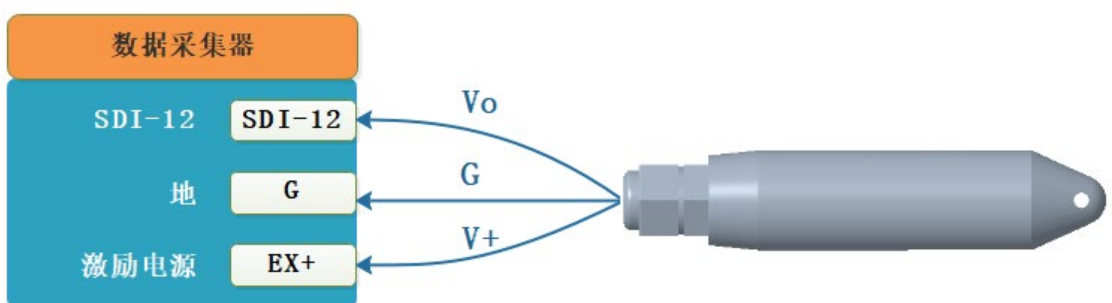
应用场景

- 土壤温度
- 河流溪流
- 地表水
- 静水井、稳液井
- 水坝
- 水产养殖池



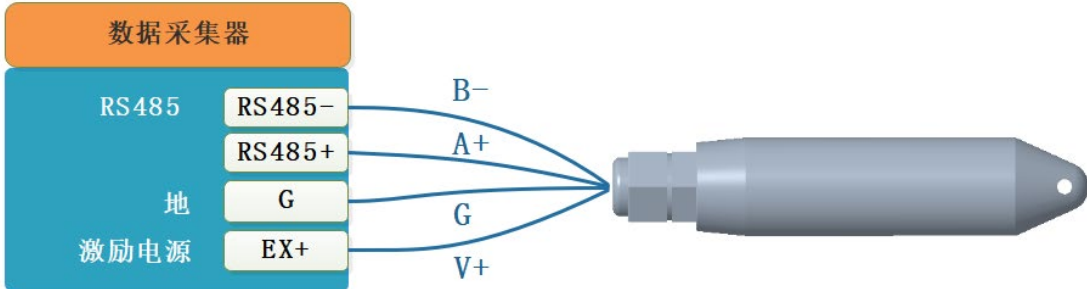
技术参数	
信号输出类型	可选：SDI-12接口V1.3版本 可选：RS485接口（Modbus-RTU协议）
供电电压	4.5-18V/DC 直流
功耗	SDI-12 接口：传感器静态电流：<10uA RS485 接口：传感器静态电流：<300uA 测量电流：测量时 50ms 内电流 10mA， 然后返回静态休眠功耗
温度测量量程	在 0~70° C 内，分辨率：0.01° C，精度：+/-0.2° C 在-40~80° C 内，分辨率：0.01° C，精度：+/-0.3° C 在-40~125° C 内，分辨率：0.01° C，精度：+/-0.5° C 注意：传感器的长期运行环境温度范围为-40~80°C
运行环境	-40~80°C
防护等级	IP68
密封材料	环氧树脂
安装方式	浸没安装、表面安装、埋入安装
默认线缆长度	5 米，长度可定制
外形尺寸	95*20mm（长度*直径）

3 传感器接线

3.1 SDI-12 接口

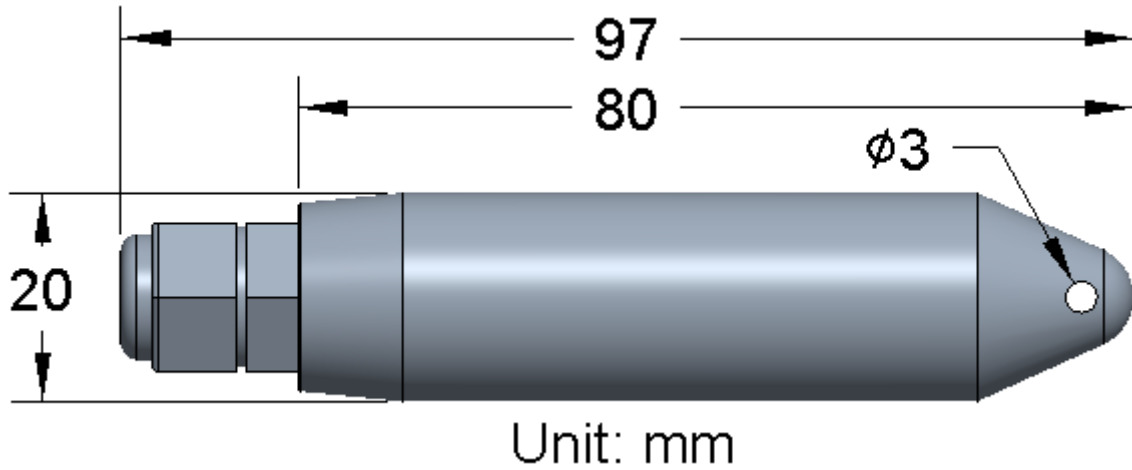
型号	接线图
SDI-12 接口信号定义	<div style="background-color: #0070C0; color: white; padding: 5px; text-align: center; font-weight: bold;">冷压端子</div>  <div style="background-color: #0070C0; color: white; padding: 5px; text-align: center; font-weight: bold; margin-top: 20px;">裸线浸锡</div> 
SDI-12 接口连接图	<div style="background-color: #0070C0; color: white; padding: 5px; text-align: center; font-weight: bold;">传感器接线</div> 

3.2 RS485 接口

型号	接线图
RS485 接口信 号定义	<div data-bbox="288 331 1425 392" style="background-color: #0070C0; color: white; padding: 5px; margin-bottom: 10px;">冷压端子</div>  <div data-bbox="288 712 1425 772" style="background-color: #0070C0; color: white; padding: 5px; margin-top: 10px;">裸线浸锡</div> 
RS485 接口连 接图	<div data-bbox="288 1093 1425 1153" style="background-color: #0070C0; color: white; padding: 5px; margin-bottom: 10px;">传感器接线</div> 

4 外型尺寸、选型订购

4.1 外型尺寸



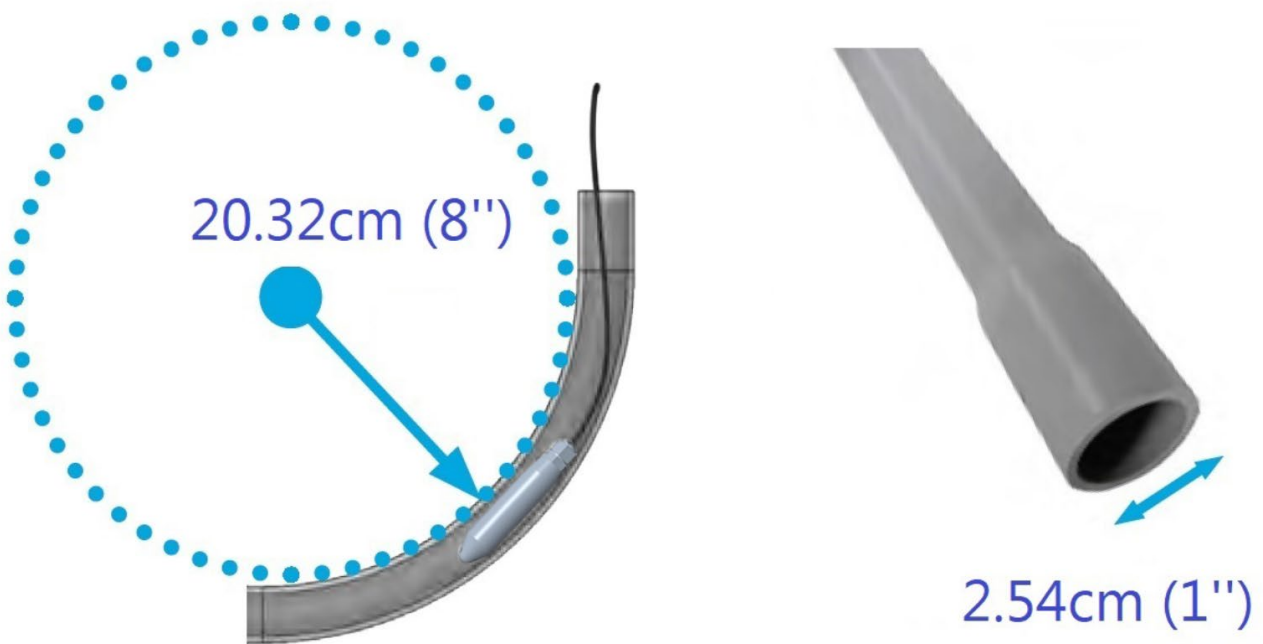
单位：毫米

4.2 选型订购

代码编号	代码	代码说明
代码 1: 产品系列	DigiTEMP	坚固型数字温度传感器
代码 2: 输出接口	A	SDI-12
	B	RS485 (Modbus-RTU)
代码 3: 供电电压	A	4.5-18V DC
	B	定制
代码 4: 连接端口	B	冷压端子
	C	裸线浸锡
代码 5: 线长	005	5米线长
	XXX	客户定制, XXX为任意线长 (单位: 米)
型号举例: DigiTEMP坚固型数字温度传感器, 输出接口: SDI-12接口, 供电电压: 4.5-18V直流, 连接端口: 冷压端子, 线长: 5米线长。选型代码为: DigiTEMP-AAB005		

5 安装与测量

- 全密封结构，可用于土壤、水体、空气等介质的测量。
- 切勿用力拖拽线缆将传感器从土壤中拔出。
- 当使用牵拉引线牵引传感器通过管道遇到阻力时，切勿用力拖拽。
- 可以穿过直径2.54cm（1英寸）、半径20.32cm（8英寸）的管道，同时在传感器前端设有线孔用于增加配重或者进行管道穿线。



6 SDI-12 数据通信

传感器具有 SDI-12 接口通信。以下章节中使用到的符号与参数说明如下：

参数	单位	说明
±	-	数值的正负号
a	-	SDI-12 地址
n	-	测量数据的个数 (固定宽度为 1)
nn	-	测量数据的个数 (固定宽度为 2)
ttt	秒	最大测量时间 (固定宽度为 3)
<TAB>	-	Tab 字符
<SPACE>	-	空格字符
<CR>	-	回车字符
<LF>	-	换行字符
<temperature>	-	温度
<CRC>	-	SDI-12协议的CRC校验

6.1 SDI-12 接口

6.1.1 电气标准

请参见 SDI-12 V1.3 手册。

6.1.2 协议解析

命令	响应	描述
a!	a<CR><LF>	确认传感器在线。 a: 传感器地址 举例： 命令：0! 响应：0<CR><LF>
aI!	allccccccmmmmmmvvvxxxxxxxxxxxx	读取传感器信息。

	xx<CR><LF>	<p>a: 传感器地址 11: SDI-12 版本 ccccccc: 公司名称代码 mmmmmm: 传感器标识符 vvv: 版本信息 xxxxxxxxxxxxxx: 产品序列号 <CR><LF>: 响应结束符</p> <p>举例: 命令: 0I! 响应: 013INFWIN DGTEMP1.01909250001000<CR><LF></p>
?!	a<CR><LF>	<p>获取传感器地址。 a: 传感器地址</p> <p>举例: 命令: ?! 响应: 0<CR><LF></p>
aAb!	b<CR><LF>	<p>修改传感器地址。 a: 当前传感器地址 b: 修改后的传感器地址</p> <p>举例: 命令: 0A1! 响应: 1<CR><LF></p>
aM!	atttn<CR><LF>	<p>温度测量</p> <p>举例: 启动测量命令。001 秒之后可以使用 aD0! 读取 1 个数据。 命令: 0M! 响应: 00011<CR><LF> 响应: 0<CR><LF> 命令: 0D0! 响应: 0+23.8<CR><LF></p>
aMC!	atttn<CR><LF>	<p>温度测量 (带 CRC 校验)</p> <p>举例: 启动带 CRC 校验的测量命令。001 秒之后可以使用 aD0! 读取 1 个数据。 命令: 0MC! 响应: 00011<CR><LF> 响应: 0<CR><LF> 命令: 0D0! 响应: 0+23.8A]p<CR><LF></p>
aC!	atttn<CR><LF>	<p>同步温度测量</p>

	<p>a: 传感器地址 ttt: 指示传感器将在 ttt 秒内完成测量 n: 传感器在回应后续的 D 指令（数据读取指令）的数据个数，数据范围从 1-9。 <CR><LF>: 响应结束符</p>	<p>举例: 启动测量命令。001 秒之后可以使用 aD0! 读取 1 个数据。 命令: 0M! 响应: 00011<CR><LF> 命令: 0D0! 响应: 0+23.8<CR><LF></p>
aCC!	<p>atttn<CR><LF> a: 传感器地址 ttt: 指示传感器将在 ttt 秒内完成测量 n: 传感器在回应后续的 D 指令（数据读取指令）的数据个数，数据范围从 1-9。 <CR><LF>: 响应结束符</p>	<p>同步温度测量（带 CRC 校验） 举例: 启动带 CRC 校验的测量命令。001 秒之后可以使用 aD0! 读取 1 个数据。 命令: 0MC! 响应: 00011<CR><LF> 命令: 0D0! 响应: 0+23.8A]p<CR><LF></p>
aV!	<p>atttn<CR><LF> a: 传感器地址 ttt: 指示传感器将在 ttt 秒内完成校验 n: 传感器在回应后续的 D 指令（数据读取指令）的数据个数，数据范围从 1-9。 <CR><LF>: 响应结束符</p>	<p>校验命令 举例: 启动校验命令。001 秒之后可以使用 aD0! 读取 1 个数据。 命令: 0V! 响应: 00011<CR><LF> 响应: 0<CR><LF> 命令: 0D0! 响应: 0+0<CR><LF>, 其中+0 为传感器正常, +1 为传感器异常。</p>
aD0!	<p>a[<svvvv>][<CRC>]<CR><LF> <svvvv>: 数据值 <CRC>: 可选的 3 字符 CRC 校验</p>	<p>数据读取命令，根据最近一次的 aM, aMC, aC, aCC, aV 命令进行数据返回。返回的数据格式取决于上一次所发的测量命令。</p>
aR0!	<p>a<svvvv><CR><LF> <svvvv>: <temperature></p>	<p>连续温度测量 举例: 命令: 0R0! 响应: 0+23.8<CR><LF></p>
aRC0!	<p>a<svvvv><CRC><CR><LF> <svvvv>: <temperature> <CRC>: CRC 校验</p>	<p>连续温度测量（带 CRC 校验） 举例: 命令: 0RC0! 响应: 0+23.8A]p<CR><LF></p>
aXR_TUNIT!	<p>aTUNIT=<X> <X> 为以下温度单位: C: 摄氏 F: 华氏 K: 开氏</p>	<p>查询温度单位 举例: 命令: 0XR_TUNIT! 响应: 0TUNIT=C<CR><LF></p>
aXW_TUNIT_<X>	<p>aTUNIT=<X></p>	<p>设定温度单位</p>

>!		举例： 命令：OXW_TUNIT_C! 响应：OTUNIT=C<CR><LF>
aXR_TOFFSET!	aTOFFSET=<svvvv> <svvvv>：温度修正值，范围在-10.00~10.00 以内，温度修正值将在新的测量命令时生效。温度的最终显示值=传感器测量的温度值+温度修正值。	查询温度修正值 举例： 命令：OXR_TOFFSET! 响应：OTOFFSET=+1.00<CR><LF>
aXW_TOFFSET_<saaaa>!	aTOFFSET=<svvvv>	设定温度修正值 举例： 命令：OXW_TOFFSET_+1.00! 响应：OTOFFSET=+1.00<CR><LF>
aXR_SN!	aSN=<ssssssss> <ssssssss>是 8 位字符的用户设置序列号	查询序列号 举例： 命令：OXR_SN! 响应：OSN=12345678<CR><LF>
aXW_SN_<sssss>!	aSN=<ssssssss>	设定序列号 举例： 命令：OXW_SN_ABCDEFGH! 响应：OSN=ABCDEFGH<CR><LF>

7 RS485 数据通信

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 TEMPRATURE	0x0000 /0	INT16 只读	3/4	-4000-12500 对应 -40.00~125.00℃。	N/A
保留 RESERVED	0x0001 /1	UINT16 只读	3/4	保留	0
保留 RESERVED	0x0002 /2	UINT16 只读	3/4	保留	0
保留 RESERVED	0x0003 /3	UINT16 只读	3/4	保留	0
保留 RESERVED	0x0004 /4	UINT16 只读	3/4	保留	0
保留 RESERVED	0x0005 /5	UINT16 只读	3/4	保留	0
温度单位 TEMPUNIT	0x0020 /32	UINT16 读写	3/6/16	0: 摄氏度℃ 1: 华氏度°F 2: 开氏度 K	0
温度偏移值 TEMPOFFSET	0x0021 /33	UINT16 读写	3/6/16	-1000-1000 对应 -10.00~10.00。	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	0-5 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 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
保留 RESERVED	0x0206 /518	UINT16 读写	3/6/16	保留	0
保留 RESERVED	0x0207 /519	UINT16 读写	3/6/16	保留	0
用户自定义序列号 (USERSN)	0x0220 /544 0x0221 /545 0x0222 /546 0x0223 /547	UINT16 读写	3/16	0x0000000000000000- 0xFFFFFFFFFFFFFFFF 用户自定义序列号，读写时需一并读写4个连续的寄存器。	N/A

UINT16:16 位无符号整数寄存器

INT16:16 位有符号整数寄存器

以 0x 起始的数据为 16 进制

7.3 Modbus 寄存器参数说明

TEMPERATURE —— 温度值		
参数范围	-4000-12500 对应 -40.00~125.00℃ 请注意：传感器使用温度范围为-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。

TEMPUNIT——温度单位		
参数范围	0: 摄氏度℃ 1: 华氏度°F 2: 开氏度 K	默认值：0
参数存储	立即存储	

意义：温度单位。

TEMPOFFSET——温度偏移值		
参数范围	-1000-1000 对应-10.00~10.00。	默认值：0
参数存储	立即存储	

意义：温度偏移值，负数用补码表示。如果传感器测量的温度为 21℃/70°F/295K，温度偏移值设置为 10.00，则温度寄存器(TEMPERATURE)的数值为 21℃/70°F/295K+10.00=31℃/80°F/305K。

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。设置后，请重新将传感器上电以使设置生效。

BAUDRATE —— 串行通信波特率		
参数范围	0-5 0:1200bps	默认值：3

	1:2400bps 2:4800bps 3:9600bps 4:19200bps 5:38400bps	
参数存储	立即存储	

设置后,请重新将传感器上电以使设置生效。

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

设置后,请重新将传感器上电以使设置生效。

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,请重新将传感器上电以使设置生效。

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,请重新将传感器上电以使设置生效。

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,请重新将传感器上电以使设置生效。

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 0x0000-0x0002, 即读取温度值, 保留寄存器, 保留寄存器
请求: 01 04 0000 0003 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	0x0003

校验	2 字节	0xB00B
----	------	--------

响应: 01 04 06 08 54 00 00 00 00 50 17

设备地址	1 字节	0x01
功能号	1 字节	0x04
有效字节数	1 字节	0x06
温度寄存器值	2 字节	0x08
		0x54
保留寄存器	2 字节	0x00
		0x00
保留寄存器	2 字节	0x00
		0x00
校验	2 字节	0x5017

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 0x0020, 即温度单位为华氏度 F 例

请求: 01 06 0020 0001 49C0

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	0x0020
寄存器数量	2 字节	0x0001
校验	2 字节	0x49C0

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	0x0020
寄存器数量	2 字节	0x0001
校验	2 字节	0x49C0

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 0x0000-0x0002，即读取温度值，保留寄存器，保留寄存器
主机请求：01 04 0000 0003 B00B （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	0x0003
校验	2 字节	0xB00B

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 03 共 6 个字节），其中 num=6

伪代码如下：

```

unsigned char request[8]={01,04,00,00,00,03,00,00}; //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

```

响应：01 04 06 08 54 00 00 00 00 50 17

设备地址	1 字节	0x01
------	------	------

功能号	1 字节	0x04
有效字节数	1 字节	0x06
温度寄存器值	2 字节	0x08
		0x54
保留寄存器	2 字节	0x00
		0x00
保留寄存器	2 字节	0x00
		0x00
校验	2 字节	0x5017

当主机接收到传感器返回的 11 个字节数据后，进行以下 crc 计算操作，其中 num=11

伪代码如下：

```

unsigned char response[11]={01 04 06 08 54 00 00 00 00 50 17}; //最后两个字节是传感器返回的 CRC 校验
unsigned char num=11; //计算整个返回的 11 个字节的 CRC 校验
unsigned int crcl6=0;
crcl6= calc_crcl6 (response, num);
if(crcl6==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
}

```

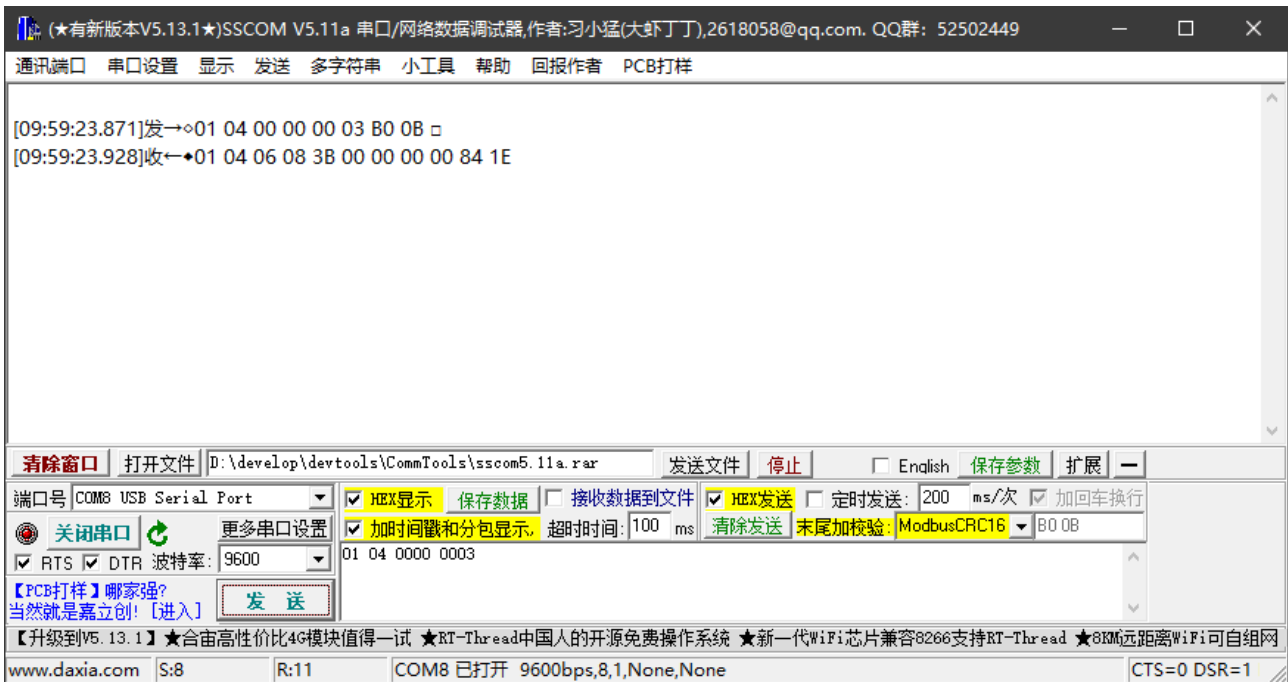
得到返回结果为0时那么校验成功，如果校验失败返回为非零值。如果校验不成功，说明传输过程发生错误，应放弃此次采集到的数据，重新采集。

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

温度= (08H*256+54H) /100=2132/100=21.32 ℃

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。



7.6 用户设置软件



附录

版权与商标

本文件大连哲勤科技有限公司版权所有。保留所有权利。有限公司保留随时对本手册所述产品进行改进的权利，恕不另行通知。未经事先书面许可，不得以任何形式或手段复制、复制、翻译或传播本手册的任何部分。本手册中提供的信息应准确可靠，但对其使用不承担任何责任，也不对其使用可能导致的任何侵犯第三方权利的行为承担任何责任。INFWIN®是大连哲勤科技有限公司的商标。

文档控制

日期	版本号	说明	完成人
2023-04-23	V1.0	创建	s151930