
1. 通讯协议

Modbus 是一种简单、开方式的通讯协议，常用于 PLC 与外围器件间的通讯，详细的通讯协议可在 www.modbus.org 处获得，本文档内容均是基于该协议，并遵循该协议的规范¹。

1.1 主从模式说明

主机（也称为主站，为用户设备，如 PLC、工业机、触摸屏等）

从机（也称为从站，为我司提供的 RE 系列无线接收器）

只有主机可开启一个会话，读取从机内部寄存器，进而获得各节点的传感器数值。

RE 系列无线接收器作为从机，任何时刻不会主动开启会话。

1.2 数据包长度说明

根据 Modbus 协议¹规定，单次会话的数据包总长小于 256 字节（包括 1 字节从机地址及 2 字节 CRC 校验）。

注 1: MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3 April 26, 2012

2.传输模式与参数

从机设备仅支持 Modbus RTU 传输模式。

2.1 串行通讯基本参数

| 参数 | 可选值 | 说明 |
|--------------------|--|------------|
| Baud rate 波特率 | 1200/2400/4800/9600/ 19200/38400/57600/115200 | 可选，默认：9600 |
| Data bits 数据位 | 8 | 固定 |
| Star bit 起始位 | 1 | 固定 |
| Parity 校验位 | None/Odd/Even 无/奇/偶 | 可选，默认：无 |
| Stop bit(s) 停止位 | 1/2/1.5 | 可选，默认：1 |

表 2-1

2.2 串行通讯帧格式

| 帧格式 | | | | |
|---------------------------|---------------------------|---------------------|------------------------|------------------------|
| Address field (1 byte) | Function code (1 byte) | Data (N bytes) | CRC Low (1 byte) | CRC High (1 byte) |
| 从机地址 ² | 功能码 ³ | N 字节数据 ⁴ | CRC 校验低字节 ⁵ | CRC 校验高字节 ⁵ |

表 2-2

注 2: 根据 Modbus 协议规定，从机地址范围为 1~247，0 为广播地址，其它为保留地址。

注 3: 仅支持功能码 03

注 4: 单次会话 $N \leq 253$

注 5: 2 字节 CRC，低字节在前，高字节在后

3.寄存器预览

Modbus 寄存器映射于从机设备内部 RAM 或 EEPROM 中，访问限制如下：

- 1、位寻址的功能码不被支持（例如线圈、开关量）
- 2、所有的寄存器均为 16 位字长
- 3、受支持的功能码为 03

| 寄存器地址 | 高字节 | | | | | | | | 低字节 | | | | | | | | 说明 | 默认值 |
|-----------------------|----------|----|----|----|----|----|---|---|----------|----|----|----|-----|----|---|---|------|--------|
| 0x0000 — 0x0003 | RESERVED | | | | | | | | | | | | | | | | 系统保留 | — |
| 0x0004 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 保留 | 0x0000 |
| | RESERVED | | | | | | | | RESERVED | | | | | | | | | |
| 0x0005 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点状态 | 0xFF00 |
| | SENS | | | | | | | | RESERVED | | | | BAT | | | | | |
| 0x0006 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点数据 | 0x8000 |
| | DATA1 | | | | | | | | DATA2 | | | | | | | | | |
| 0x0007 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点数据 | 0x8000 |
| | DATA3 | | | | | | | | DATA4 | | | | | | | | | |
| 0x0008 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 保留 | 0x0000 |
| | RESERVED | | | | | | | | RESERVED | | | | | | | | | |
| 0x0009 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点状态 | 0xFF00 |
| | SENS | | | | | | | | RESERVED | | | | BAT | | | | | |
| 0x000A | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点数据 | 0x8000 |
| | DATA1 | | | | | | | | DATA2 | | | | | | | | | |
| 0x000B | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点数据 | 0x8000 |
| | DATA3 | | | | | | | | DATA4 | | | | | | | | | |
| 0x000C | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 保留 | 0x0000 |
| | RESERVED | | | | | | | | RESERVED | | | | | | | | | |
| 0x000D | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点状态 | 0xFF00 |
| | SENS | | | | | | | | RESERVED | | | | BAT | | | | | |
| 0x000E | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点数据 | 0x8000 |
| | DATA1 | | | | | | | | DATA2 | | | | | | | | | |
| 0x000F | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 节点数据 | 0x8000 |
| | DATA3 | | | | | | | | DATA4 | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |

| 寄存器地址 | 高字节 | | | | | | | | 低字节 | | | | | | | | 说明 | 默认值 |
|-----------------------|----------|----|----|----|----|----|---|---|----------|----|----|----|-----|----|---|---|----------|--------|
| 0x0188 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 保留 | 0x0000 |
| | RESERVED | | | | | | | | RESERVED | | | | | | | | | |
| 0x0189 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 状态 | 0xFF00 |
| | SENS | | | | | | | | RESERVED | | | | BAT | | | | | |
| 0x018A | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 数据 | 0x8000 |
| | DATA1 | | | | | | | | DATA2 | | | | | | | | | |
| 0x018B | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 数据 | 0x8000 |
| | DATA3 | | | | | | | | DATA4 | | | | | | | | | |
| 0x018C | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 保留 | 0x0000 |
| | RESERVED | | | | | | | | RESERVED | | | | | | | | | |
| 0x018D | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 状态 | 0xFF00 |
| | SENS | | | | | | | | RESERVED | | | | BAT | | | | | |
| 0x018E | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 数据 | 0x8000 |
| | DATA1 | | | | | | | | DATA2 | | | | | | | | | |
| 0x018F | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 数据 | 0x8000 |
| | DATA3 | | | | | | | | DATA4 | | | | | | | | | |
| 0x0190 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 保留 | 0x0000 |
| | RESERVED | | | | | | | | RESERVED | | | | | | | | | |
| 0x0191 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 状态 | 0xFF00 |
| | SENS | | | | | | | | RESERVED | | | | BAT | | | | | |
| 0x0192 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 节点 数据 | 0x8000 |
| | DATA1 | | | | | | | | DATA2 | | | | | | | | | |
| 0x0193 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 节点 数据 | 0x8000 |
| | DATA3 | | | | | | | | DATA4 | | | | | | | | | |
| 0x0194 — 0xFFFF | 非法地址 | | | | | | | | | | | | | | | | | |

表 3-1

4. 寄存器字段描述

| 字段 | 说明 | 备注 |
|----------------------------------|------------|-----------------------------------|
| RESERVED | 保留 | 用户应忽略 |
| DATA1 DATA2 DATA3 DATA4 | N号节点，传感器数值 | 具体含义，详见后文 传感器数值解析 |
| SENS | N号节点，传感器类型 | 具体含义，详见后文 传感器类型 |
| BAT | N号节点，电池电量 | 0-6，值越大，电量越充足 |

表 4-1

结合表 3-1 与 4-1 不难得出规律，1 号节点使用地址 0x04 开头的 4 个寄存器，2 号节点使用地址 0x08 开头的 4 个寄存器，则 N 号节点使用地址 4×N 开头的 4 个寄存器。

例 1：用户主站期望获取 1~30 号节点的数据：

主站应发送：0x59 0x03 0x00 0x04 0x00 0x78 0x09 0x31，共计 8 字节。

从站将回复：0x59 0x03 0xF8 0xFF 0xFF，共计 245 字节。

例 2：用户主站期望获取 1~100 号节点的数据（4 次会话，每次 25 个节点）：

由于 ModbusRTU 传输模式限制了单次会话（请求或回复）的字节量（不得超过 256 字节），故主站需向从站发起多次请求，以获得全部节点的数据。

第 1 次获取 1~25 号节点

主站应发送：0x59 0x03 0x00 0x04 0x00 0x64 0x08 0xF8，共计 8 字节。

从站将回复：0x59 0x03 0xC8 0xFF 0xFF，共计 205 字节。

第 2 次获取 26~50 号节点

主站应发送：0x59 0x03 0x00 0x68 0x00 0x64 0xC8 0xE5，共计 8 字节。

从站将回复：0x59 0x03 0xF0 0xFF 0xFF，共计 205 字节。

第 3 次获取 51~75 号节点

主站应发送：0x59 0x03 0x00 0xCC 0x00 0x64 0x89 0x06，共计 8 字节。

从站将回复：0x59 0x03 0xF0 0xFF 0xFF，共计 205 字节。

第 4 次获取 76~100 号节点

主站应发送：0x59 0x03 0x01 0x30 0x00 0x64 0x48 0xCA，共计 8 字节。

从站将回复：0x59 0x03 0x50 0xFF 0xFF，共计 205 字节。

5. 传感器数值解析

5.1 温度传感器

温度传感器使用 DATA1、DATA2 字段，其中 DATA1 为高 8 位，DATA2 为低 8 位，温度值采用 2 进制补码形式表达，具体换算如下：

例 1：用户主站读取到寄存器 DATA1、DATA2 分别为 0x00，0xF3，
则实际的温度值 $T=0x00F3 \div 10=24.3^{\circ}\text{C}$ ， $T=243 \div 10=24.3^{\circ}\text{C}$ ，
其中 0x00F3 为 16 进制表达形式，243 为 10 进制表达形式，
最后 T 除以 10，即为所求温度，1 位小数点有效。

例 2：用户主站读取到寄存器 DATA1、DATA2 分别为 0xFF，0xC8，
则实际的温度值 $T=0xFFC8 \div 10= -5.6^{\circ}\text{C}$ ， $T= -56 \div 10= -5.6^{\circ}\text{C}$ ，即零下 5.6 $^{\circ}\text{C}$

为便于用户主站设备编程，提供如下 C 代码作为参考：

```
#include <stdint.h>

float get_temp(uint8_t data1, uint8_t data2)
{
    uint16_t u_temp = (data1 << 8) + data2;

    int16_t s_temp = (int16_t) u_temp;

    float f_temp = s_temp / 10.0f;

    return f_temp;
}
```

5.2 温、湿度传感器

温、湿度传感器使用 DATA1、DATA2、DATA3 和 DATA4 字段，
DATA1、DATA2 为温度值（DATA1 为高 8 位，DATA2 为低 8 位），
DATA3、DATA4 为湿度值（DATA3 为高 8 位，DATA4 为低 8 位）。

温度值换算方式与上述[温度传感器](#)方式一致，不再赘述，
湿度与温度换算基本相同，但湿度值并不会出现负数，即小于 0 的情况。

例 1：用户主站读取到寄存器 DATA3、DATA4 分别为 0x00，0xC3，
则实际的湿度值 $H=0x00C3 \div 10=19.5\%$ ， $H=195 \div 10=19.5\%$ ，
其中 0x00C3 为 16 进制表达形式，195 为 10 进制表达形式，
最后 H 除以 10，即为所求湿度，1 位小数点有效。

例 2：用户主站读取到寄存器 DATA3、DATA4 分别为 0x03，0xE7，
则实际的湿度值 $H=0x03E7 \div 10=99.9\%$ ， $H=999 \div 10=99.9\%$

5.3 照度传感器

照度传感器使用 DATA1、DATA2、DATA3 和 DATA4 字段。

例 1: 用户主站读取到寄存器 DATA1、DATA2、DATA3、DATA4 分别为 0x00, 0x01, 0xA9, 0x40, 则实际的照度值 $L=0x0001A940 \div 1000=108864 \div 1000=108.864\text{Lux}$, 其中 0x0001A940 为 16 进制表达形式, 108864 为 10 进制表达形式, 最后 L 除以 1000, 即为所求照度, 3 位小数点有效。

例 2: 用户主站读取到寄存器 DATA1、DATA2、DATA3、DATA4 分别为 0x0B, 0x34, 0xA7, 0x00, 则实际的照度值 $L=0x0B34A700 \div 1000=188000000 \div 1000=188000.000\text{Lux}$ 。

5.4 水浸传感器（非定位型）

非定位型水浸传感器只能检测出线缆上是否有水，其量程范围为 0~4095，以传感器 ADC 原始读数作为测量结果，测量值由 DATA1、DATA2 字段构成。

例 1：用户主站读取到寄存器 DATA1、DATA2 分别为 0x07，0x2E，
则实际的水浸值 $W=0x072E=1838$ ，
其中 0x072E 为 16 进制表达形式，1838 为 10 进制表达形式。

例 2：用户主站读取到寄存器 DATA1、DATA2 分别为 0x0F，0xFF，
则实际的水浸值 $W=0x0FFF=4095$ 。

用户应根据现场环境、液体化学特性，自行规定浸水的阈值（上下限）。
环境温度 $T_A=25^{\circ}\text{C}$ ，液体为自来水（ $6.5\leq\text{pH}\leq 8.5$ ）时，推荐阈值设为 0x0DAC（3500）。
例如，当传感器测量值 $W\leq 3500$ ，则可视为线缆浸水。

注：其它工况，阈值须按现场环境及液体化学特性而定。

5.5 压力传感器（表压）

表压传感器使用 DATA1、DATA2、DATA3 和 DATA4 字段。

例 1：用户主站读取到寄存器 DATA1、DATA2、DATA3、DATA4 分别为 0x00, 0x1E, 0x84, 0x80，
则实际的压力值 $P=0x001E8480=2000000Pa=2MPa$ ，
其中 0x001E8480 为 16 进制表达形式，2000000 为 10 进制表达形式。

5.6 压力传感器（大气压）

气压传感器使用 DATA1、DATA2、DATA3 和 DATA4 字段。

例 1：用户主站读取到寄存器 DATA1、DATA2、DATA3、DATA4 分别为 0x00, 0x01, 0x5F, 0x90，
则实际的压力值 $P=0x00015F90=90000Pa=900hPa=900mbar$ ，
其中 0x00015F90 为 16 进制表达形式，90000 为 10 进制表达形式。

5.7 二氧化碳传感器 (CO₂)

二氧化碳传感器使用 DATA1、DATA2 字段。

例 1: 用户主站读取到寄存器 DATA1、DATA2 分别为 0x03, 0xE0,
则实际的二氧化碳浓度 $C=0x03E0=992\text{ppm}$,
其中 0x03E0 为 16 进制表达形式, 992 为 10 进制表达形式。

5.8 颗粒物传感器（PM2.5）

颗粒物传感器使用 DATA1、DATA2 字段。

例 1：用户主站读取到寄存器 DATA1、DATA2 分别为 0x03, 0x75，
则实际的颗粒物浓度 $L=0x0375=885 \mu\text{g}/\text{m}^3$ ，
其中 0x0375 为 16 进制表达形式，885 为 10 进制表达形式。

5.9 甲醛传感器 (HCHO)

甲醛传感器使用 DATA1、DATA2 字段。

例 1: 用户主站读取到寄存器 DATA1、DATA2 分别为 0x03, 0xE0,
则实际的甲醛浓度 $F=0x03E0=992\text{ppb}=0.992\text{ppm}$,
其中 0x03E0 为 16 进制表达形式, 992 为 10 进制表达形式。

5.10 液位传感器

液位传感器使用 DATA3、DATA4 字段。

DATA1、DATA2 为预留

例 1：用户主站读取到寄存器 DATA3、DATA4 分别为 0x03, 0xE0，
则实际的液位深度 $DP = 0x03E0 \div 100 = 992 \div 100 = 9.92 \text{ mH}_2\text{O}$ ，
其中 0x03E0 为 16 进制表达形式，992 为 10 进制表达形式。
最后 DP 除以 100，即为所求深度，2 位小数点有效。

5.11 TVOC 传感器

TVOC 传感器使用字段 DATA1、DATA2、DATA3、DATA4。

其中，DATA1、DATA2 为 TVOC 浓度值。

DATA3、DATA4 分别为 TVOC 污染等级、传感器工作的状态字。

例 1：用户主站读取到寄存器 DATA1、DATA2 分别为 0x03, 0xE0，
则实际的 TVOC 浓度 $CON=0x03E0=992 \mu\text{g}/\text{m}^3$ ，
其中 0x03E0 为 16 进制表达形式，992 为 10 进制表达形式。

DATA3 为污染等级，分为四级（0~3），如下表所示：

| 值 | 污染等级 | TVOC 浓度 ($\mu\text{g}/\text{m}^3$) |
|---|------|--------------------------------------|
| 0 | 优 | <360 |
| 1 | 良 | 360~900 |
| 2 | 中 | 900~1800 |
| 3 | 差 | >1800 |

表 5-11a

在很多新风及净化类应用中，并不需要获取 TVOC 浓度的具体值，仅需 TVOC 质量评估的结果，DATA3 正是为便于此类评估应用而考虑，可直接使用 DATA3，而不必进行复杂判断。当然用户也可根据 TVOC 浓度，自行划分区间进行判别。

DATA4 为传感器工作的状态字，如下表所示：

| 值 | 状态字 | 描述 |
|-----------|---------|------------------|
| 0x00 | OK | 正常 |
| 0x01 | Heating | 预热中 ¹ |
| 0x02-0xFF | Error | 未知错误 |

表 5-11b

注 1：TVOC 传感器为 MEMS 金属氧化物半导体传感器，内部以间歇式加热的工作方式检测目标气体，因此需要预热阶段，一般冷上电 180 秒后，测量结果方为可用。

用户应读取工作状态字，区分出加热阶段的始末。

6. 传感器类型

| 传感器类型 | 分类描述 | 类型值 |
|------------------------|-----------|-------------------|
| - | - | 0xFF ¹ |
| DST | 温度传感器 | 0x00/0x01 |
| NTC | | 0x03 |
| NTC-W | | 0x02 |
| KT | | 0x08 |
| PT100/1000 | | 0x10 |
| TC55_B | | 0x60 |
| TC55_K | | 0x63 |
| TC55_S | | 0x66 |
| XM24 | | 0x24 |
| HT | | 温、湿度传感器 |
| HT-A | 0x80 | |
| HT-AH | 0x84 | |
| MX | 照度传感器 | 0x70 |
| SK | 水浸传感器 | 0x25 |
| BMP | 压力传感器(气压) | 0x39 |
| SPL | | 0x06 |
| SMP11 | 压力传感器(表压) | 0x11 |
| SMP46 | 液位传感器 | 0x46 |
| MS-VOC-V4 | TVOC | 0x43 |
| HCHO-V4C | 甲醛传感器 | 0x44 |
| HCHO | | 0x45 |
| PMS7 | 颗粒物传感器 | 0x40 |
| PM12 | | 0x12 |
| CM6L | 二氧化碳传感器 | 0x16 |
| CM7 | | 0x17 |
| DS20 | | 0x20 |
| Z19 | | 0x19 |
| S8L ² /CM6S | | 0x18 |

表 6-1

注 1: 当类型值为 0xFF 时, 表示设备初上电, 暂未收到该节点数据, 或该节点已离线。

注 2: S8L 并未实际量产。

附录 I—CRC16 Modbus

```
#include <stdint.h>
#include <stdio.h>

uint16_t crc16_calc_modbus(const uint8_t* buff, uint32_t length)
{
    uint32_t i, mask, crc = 0xFFFF;

    for (i = 0; i < length; i++)
    {
        for (mask = 1; mask <= 0xFF; mask <<= 1)
        {
            if ((crc & 1) > 0)
                crc = (crc >> 1) ^ 0xA001;
            else
                crc = crc >> 1;

            if ((buff[i] & mask) > 0)
                crc ^= 0xA001;
        }
    }

    //LSB
    return (uint16_t)(crc << 8 | (uint8_t)(crc >> 8));

    //MSB
    //return (uint16_t)(crc);
}

int main()
{
    uint8_t buff[] = { 1, 2, 3, 4 };

    uint16_t crc = crc16_calc_modbus(buff, sizeof(buff));

    printf("%d (0x%.4X)", crc, crc); //41259 (0xA12B)
}
```

附录 II—修改记录

| 日期 | 版本 | 原因 | 修改人 | 审核人 |
|----------|------|---|-----|-----|
| 17-09-05 | 0.1 | 1、初始发布 | 张阳 | |
| 17-10-19 | 0.2 | 1、添加表 1-1 | 张阳 | |
| 17-11-08 | 0.3 | 1、修改串口默认参数 | 张阳 | |
| 17-11-24 | 0.4 | 1、修改引脚说明 | 张阳 | |
| 17-12-07 | 0.5 | 1、修改表 2-1，删除 ROM Version=0x12 的描述 2、修改表 2-1，删除 ROM Version=0x13 的描述 3、修改参数寄存器 1 的默认值为 0x7875 4、修改参数寄存器 4 的默认值为 0x0014 | 张阳 | |
| 19-01-22 | 0.6 | 1、弱化射频型号，合并文档 2、修改 0x00-0x03 寄存器为系统保留寄存器，删除其说明 | 张阳 | |
| 22-11-02 | 0.7 | 1、优化文档结构 2、删除章节 5—引脚说明 3、添加多种传感器换算 4、开放传感器类型字段 5、增加表 6-1 6、增加附表 CRC16 Modbus | 张阳 | |
| 23-04-18 | 0.8 | 1、删除页眉 | 张阳 | |
| 23-05-24 | 0.9a | 1、增加章节 5.10 2、表 6-1 增加液位传感器 3、表 6-1 中，删除传感器 SCD4 4、表 6-1 中，增加传感器 MS-VOC-V4 5、表 6-1 中，增加传感器 TC55_B 6、表 6-1 中，增加传感器 TC55_K 7、表 6-1 中，增加传感器 TC55_S 8、表 6-1 中，兼容 S8L 与 CM6S 9、增加章节 5.11 | 张阳 | |
| 23-07-05 | 1.0 | 1、表 6-1 中，重命名 SMP -> SMP11 2、修改水浸传感器推荐阈值的描述 | 张阳 | |

附表 1