

SJ8902 无线温湿度传感器说明书

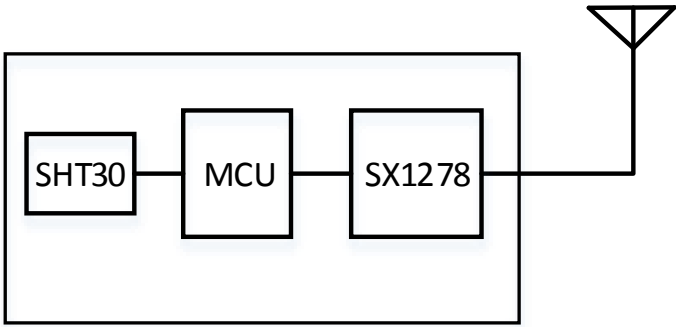
概述

SJ8902 是一款高精度无线温湿度传感器，采用低功耗、远距离 LoRa 无线方案，具有超强稳定性和抗干扰能力。测量探头进口温湿度传感器，保证优异的测量性能。使用普通 AA（5 号电池）供电，方便维护和更换，使用时间长达 1 到 3 年。



SJ8902 无线温湿度传感器外观

系统框图



SJ8902 系统结构

基本参数

硬件参数	
天线接口	SMA
射频芯片	SX1278
支持频率	410~525MHz
调制方式	LoRa
带宽	250K、500K 、125K (默认) 可调
发射功率	5~20dBm 可调
接收灵敏度	-141dBm , @470MHz , 125K 带宽 , SF=12
通信距离	3Km , @空旷条件 , 125K 带宽 , SF=12
供电方式	2 节 AA 电池串联
发射电流	130mA , @最大发射功率 20dBm
休眠电流	5uA
开放接口	USB
传感器参数	
湿度工作范围	0~100%RH
温度工作范围	-40℃ - 125℃
典型湿度精度 (%RH)	±3 @10-90% RH
典型温度精度 (°C)	±0.3 @0-65 °C

接收网关推荐



SX1301

GW2 LORA网关

室内工业级网关产品

支持半双工通信，收发灵敏

可配置功率与通信频点

同口或者无线LTE回传

宽电压直流电源供电

自适应数据速率，可同时解SF7-SF12的数据

8个独立物理接收信道，接入容量大

LoRa 网关

GW2

Releee

GW2

Releee

SX1301

GW1 LORA网关

室外工业级网关产品

支持全双工通信，收发灵敏互不干扰

可配置功率与通信频点

同口或者RS485接口回传

宽电压直流电源供电

自适应数据速率，可同时解SX7-SX12的数据

8个独立物理接收信道，接入容量大

LoRa 网关

GW1

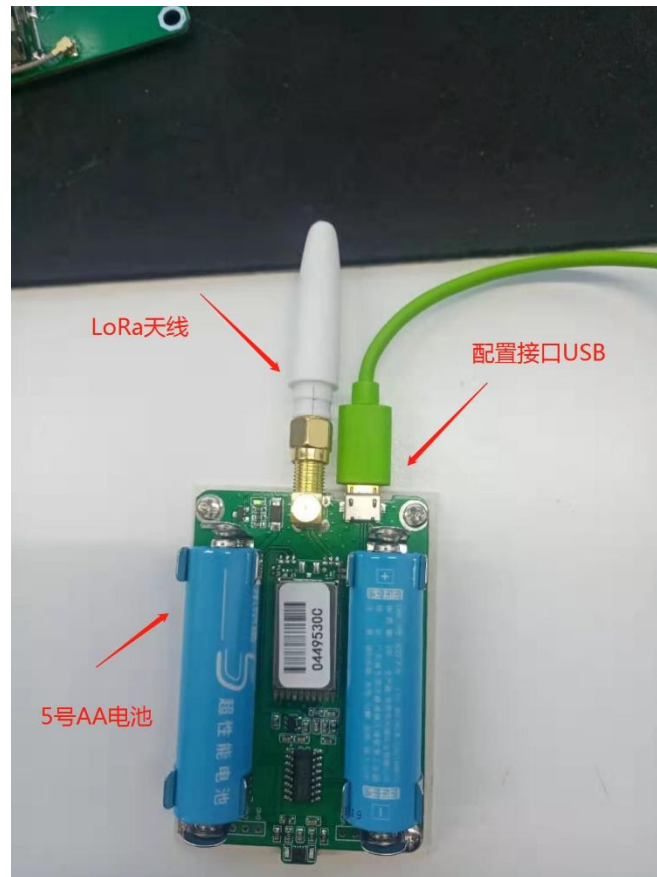
Releee

GW1

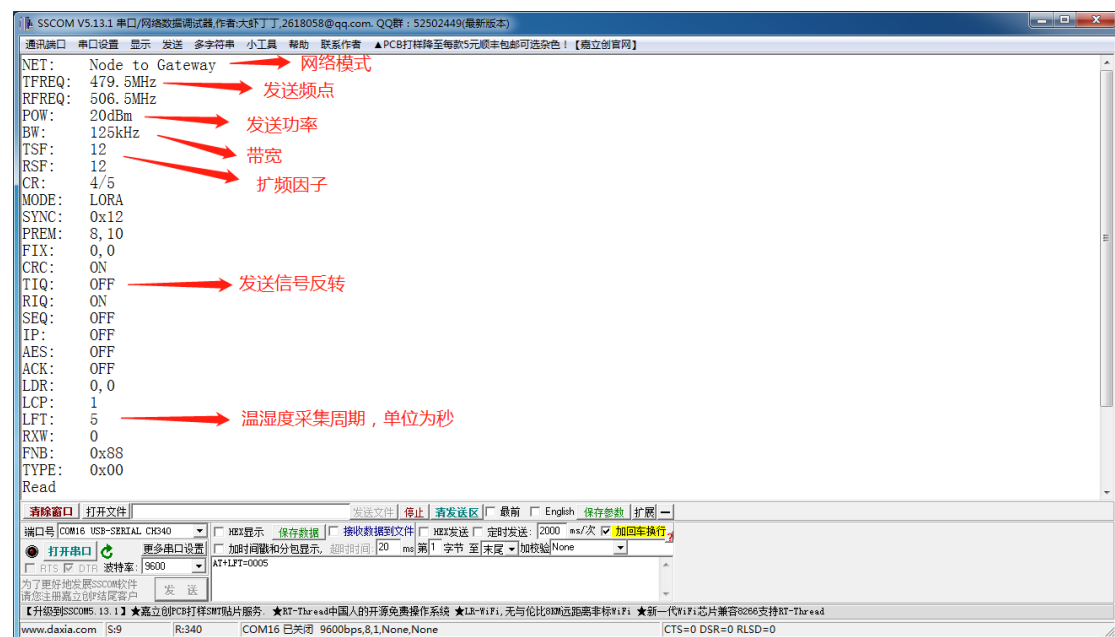
Releee

参数配置说明

插入 USB 数据线,安装 CH340 驱动, 可以通过串口进行参数配置。



发送查询指令“AT+CFG?”(需加回车换行发送)



发送指令“AT+LFT=0005”(需加回车换行发送), 可修改传感器采集上报周期为 5s 一次。

能耗计算

发射电量消耗

参数	TX 电流	发送时间（18 字节）	发送 10000 次消耗电量
TSF=12	128mA	1325ms	470mAH
TSF=11	128mA	744ms	264 mAH
TSF=10	128mA	373ms	132 mAH
TSF=09	128mA	187ms	67 mAH
TSF=08	128mA	103ms	36 mAH
TSF=07	128mA	57ms	20mAH

待机电量消耗

参数	电流	一年消耗电量
Isleep	5uA	43.8mAH

数据格式解析

无线数据包：03 04 49 53 0C 00 1F 21 00 00 1C 04 00 A3 05 2C DF B3

Header 1 字节	DevAddr 4 字节	FCtrl 1 字节	SeqNo 2 字节	传感器数据 8 字节	CRC 2 字节
协议头	设备地址	控制字	包序号	数据类型+数据内容	校验
03	04 49 53 0C	00	1F 21	00 00 1C 04 00 A3 05 2C	DF B3

协议头：“03”

默认内容固定“03”

设备地址：“04 49 53 0C”

可通过外壳条码查看，也可通过串口 AT 指令查询，查询指令“AT+ADDR?”

控制字：“00”

默认内容固定“00”

包序号：“1F 21”

传感器每发送一次，自动增加。

传感器数据：

主要包含电池电压，温度和湿度数据。

00 设备电压数据类型，2 字节，单位 0.1V

00 1C 电压 2.8V

04 节点温度，2 字节，单位 0.1℃

00 A3 温度 16.3℃

05：湿度，1 字节，单位 1%

2C:湿度 44%

CRC 校验：“DF B3”

本文档使用的 CRC 校验算法如下。

```
static uint16_t get_crc16(uint16_t inData, uint16_t outData) {
    outData = (outData >> 8) | (outData << 8);
    outData ^= inData;
    outData ^= (outData & 0xff) >> 4;
    outData ^= outData << 12;
    outData ^= (outData & 0xff) << 5;
    return outData;
}

static uint16_t cal_crc16(const uint8_t *pData, const uint32_t len)
{
    uint32_t i = 0;
    uint16_t crc16 = 0xFFFF;
    for (i = 0; i < len; i++) {
        crc16 = get_crc16(*(pData++), crc16);
    }
    return crc16;
}
```