



深圳市硅传科技有限公司

SHENZHEN SILICONTRA TECHNOLOGY CO.,LTD.



CC1310ZTR4-YY

433M无线有源测温模块使用说明书

(V3.0)

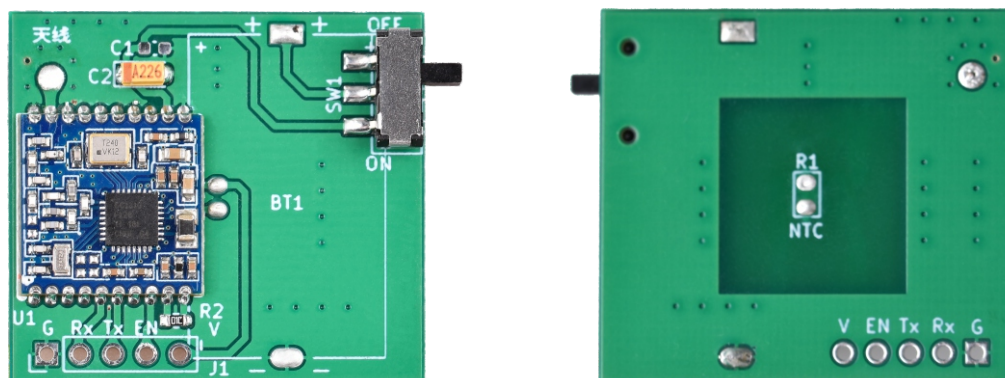
目录

一、 模块介绍	3
1.1 模块概述	3
1.2 准备工具	3
二、 接口描述	4
三、 配套连接图	5
四、 参数说明	5
4.1 串口参数设置	5
4.2 属性说明	6
4.3 传感器配置说明	6
4.4 配置方法与步骤	7
五、 传感器模式操作说明	8
5.1 模块主从机实物图	9
5.2 配置方法与步骤:	10

文档修订记录

版本	更改日期	更改说明
V1.0	2019年10月10日	初始版本
V2.0	2022年04月04日	格式优化
V3.0	2023年10月26日	格式优化

一、 模块介绍



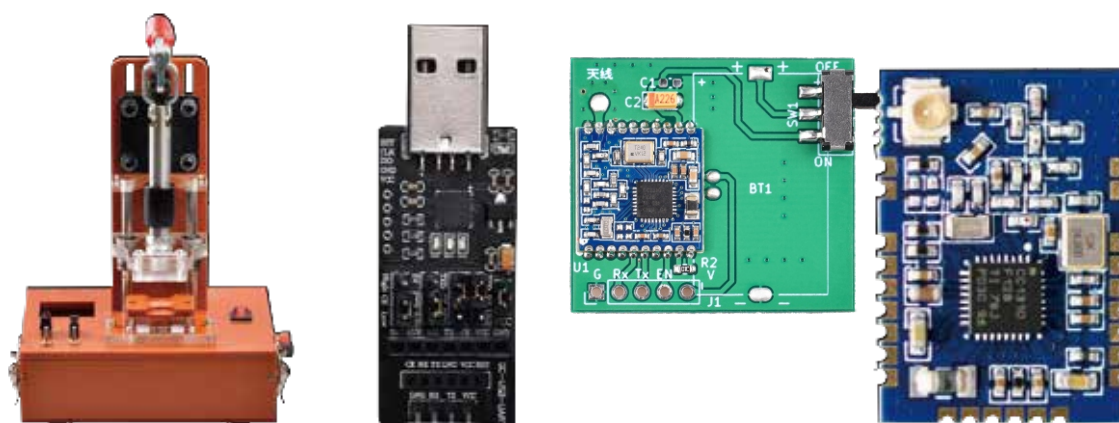
模块正面

模块反面

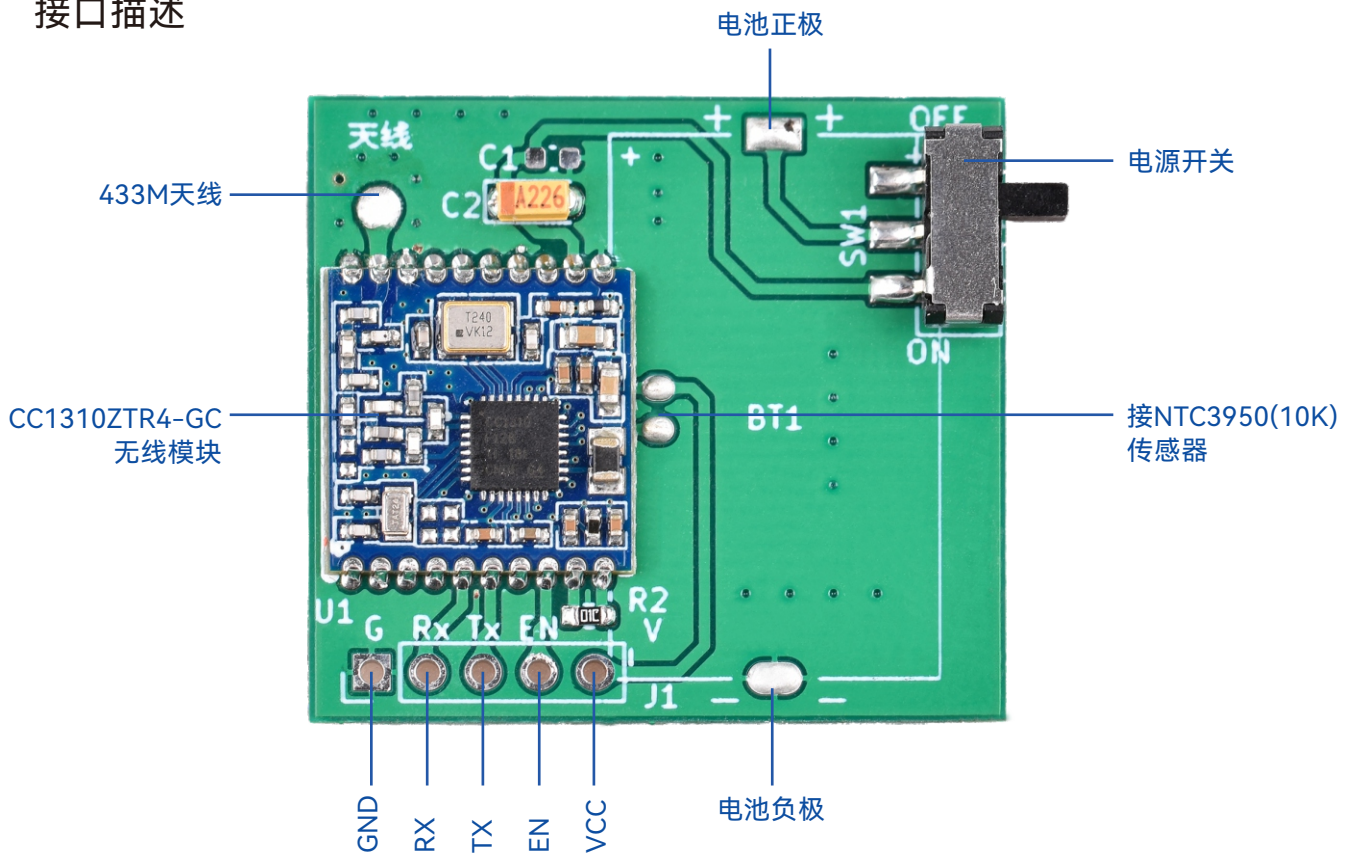
1.1 模块概述

此文档主要介绍如何通过硅传科技的配置工具(CC1310)进行各种参数的设置以及主要参数说明,方便客户能快速的去配置与了解模块的各个参数, 比如模块的串口波特率、发射功率、射频通道、速率等等, 可以设置主从机以及各种传感器类型、节点模式、上报周期、上电延时、分组ID、节点ID等等。

1.2 准备工具



二、接口描述



说明:

尺寸: 32mm×29mm

传感器: NTC3950(10K)

+3.3V: 接电源

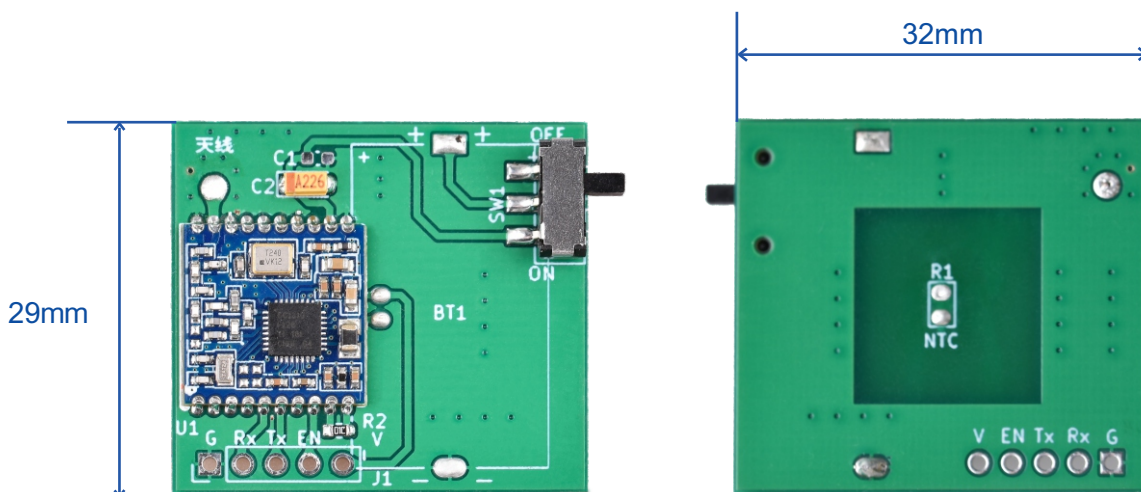
EN: 配置参数时必须接地,传感器模式下,主机需要接地,从机可以悬空

RX: 串口接收数据引脚,接到USB-TTL的TX引脚

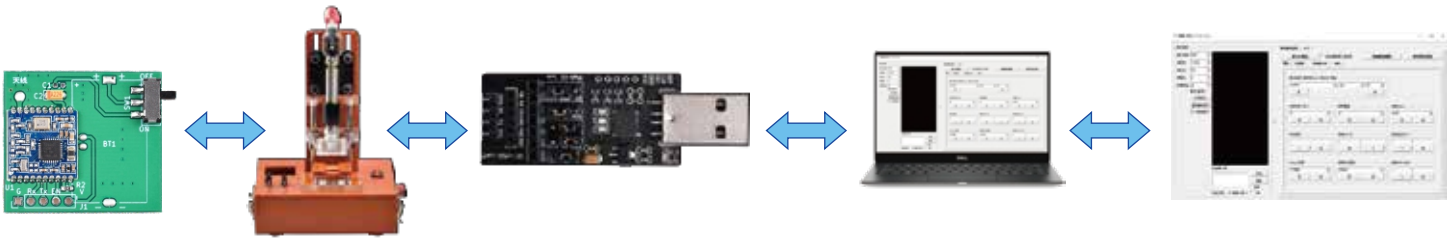
TX: 串口发送数据引脚 接到USB-TTL的RX引脚

GND: 接地

模块尺寸:

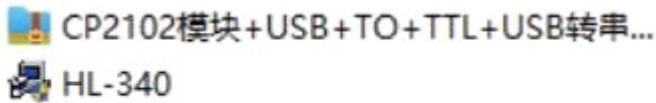


三、配套连接图



第一步：把模块放在配置治具上，注意模块摆放方向

第二步：把配置治具接上GC-USB-UART再将USB插入电脑，注意要先安装驱动



第三步：打开硅传上位机软件即可完成连接操作

四、参数说明

4.1 串口参数设置

- ① 使用串口配置工具时串口参数必须保持跟模块参数一致才可进入AT模式。
- ② 模块出厂默认波特率：115200停止位：1位数据位：8位：奇偶校验：无校验。
- ③ 如果客户修改了模块串口参数则必须根据模块的串口参数进行选择。
- ④ 模块与USB-TTL接线没问题之后打开串口，就可以进入AT配置模式了



4.2 属性说明



(1)串口参数：主要是设置模块的串口参数，设置完成之后必须软件复位系统或者重新上电参数才生效。

(2)射频参数：

- ① 功率：0~14dBm可选，模块默认出厂功率为5dBm，建议设置为10dBm。
- ② 射频通道：0~7可选每个信道相隔3MHZ
- ③ 速率：625bps~500000bps选择，模块默认出厂速率为5000bps

4.3 传感器配置说明



(1)传感器类型包括：ADC传感器、芯片内部测温、SHT2X系列、SHT3X系列、TMP112X系列
(注意：此模块仅支持ADC传感器类型)

(2)节点模式包括：

- ① 主机：用于接收节点数据，此时主机应该与节点配置为同类型传感器模式，否则收到的数据会发现少一个字节的问题
- ② 从机：用于上报采集到的数据

(3)上报周期：

- ① 主要对从机有效，可设置为1~3000S此周期单位是秒比如设置5那么模块会定时5秒一个周期进行一次采集之后会通过射频把数据发射出去，平时不采集时是处于休眠状态，值得注意的是此参数设置的周期跟设定的周期会有误差。
- ② 如果用于取电建议客户把上报周期设置长一点，这取决于线圈的充电能力，线圈取电时间越快，周期可以缩短，否则模块不能正常的把数据发射出去。默认出厂设置为5秒，建议设置为10-20秒。

(4)上电延时

主要对从机有效，可设置为1~255S此周期单位是秒比如设置5那么模块会在上电5S后开始采集与上报数据。默认出厂设置为5秒，建议设置为10-20秒。

(5)分组ID、节点ID

① 分组ID: 对于主机如果分组ID设置为0则不会过滤从机的数据, 接收所有从机数据, 如果不为0则主从分组ID必须保持一致主机才可以接收到从机数据。

② 节点ID: 主要是为了识别在一个分组ID的某个节点, 在解析数据的时候有用。

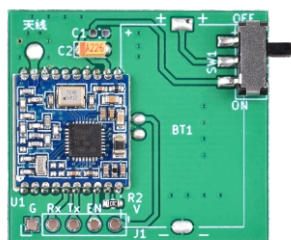
(6)采集检测次数

可设置为1~20, 意思是在一个设定的上报周期内检测多少次, 比如上报周期设置为10S, 采样检测次数设置为2次, 那么模块会在10S周期内采集2次, 在每次采集到的温度浮动在 $\pm 0.5^{\circ}$ 则会立刻上报数据, 如果在一个周期结束时温度不变则也会上报数据(最后一次), 值得注意的是上报周期必须要设置大于等于10S时此参数才有效。

五、传感器模式操作说明

5.1 模块主从机实物图:

其中可以把硅传的CC1310TR4-YY做节点, CC1310PATR4-GC、CC1310-TC005做为主机, 见图下:



模块正面



主机1 (CC1310PATR4-GC)



主机1 (CC1310PATR4-GC)

(注释: 主机接收模块可在两款中任选一款即可)

5.2 配置方法与步骤:

(1) 节点设置

流程: 选择串口参数->打开串口->进入AT模式->点击传感器栏->选择传感器类型为ADC传感器->设置为从机->上报周期设置为5S->上电延时5S->分组ID为0->节点ID为0->采样检测次数为1->软件复位系统或者重上电, 如下图所示:



注意：如果主机分组ID为0，不过滤任何传感器数据，大于0则必须主机跟从机分组ID需一致。

(2)主机设置

设置流程：选择串口参数->打开串口->进入AT模式->点击传感器栏->选择传感器类型为ADC传感器->设置为主机->分组ID为0->软件复位系统或者重上电，如下图所示：



注意：如果主机分组ID为0，不过滤任何传感器数据，大于0则必须主机跟从机分组ID 需一致。

(3) 其它参数设置

设置流程：选择串口参数->打开串口->进入 AT 模式->点击属性栏->设置发射功率-> 设置射频通道->设置空速->软件复位系统或者重上电，如下图所示：



(4)收发流程演示与数据解析

①按照以上三个步骤设置好以后就可以进行节点与主机通信了，如下图所示：



②收到的数据进行解析就可以得到实际温度了，其数据包格式为：

BYTE0	BYTE1	BYTE2	BYTE3	BYTE4	BYTE5	BYTE6
GID	SID	SENSOR_VALUE(H)	SENSOR_VALUE(L)	BAT_VALUE	FCS	RSSI

GID: 节点的组ID，可以作为分组ID

SID: 节点的节点ID，区分相同组ID中的不同节点

“传感器类型”设置为“ADC 传感器”时

(SENSOR_VALUE(H): 节点传感器的采集的AD值的高8位，模组供电电压作为参考电压测量，精度12位

(SENSOR_VALUE(L): 节点传感器的采集的AD值的低8位，模组供电电压作为参考电压测量，精度12位

BAT_VALUE: 节点模组的供电电压值，转换公式: 实际电压(x.xx V)=(BAT_VALUE+200)/100, 内部参考电压测量

FCS: 和校验，节点模组对GID+SID+ADC_VALUE(H)+ADC VALUE(L)+BAT VALUE 的和校验

RSSI: 信号强度，节点模组的信号强度，比如 0xeb(235) -> "-21"

③解析源码如下：

```
/**
 * @param:
 *     adcValue:CC1310采样的ADC值
 *     batteryValue_mV: 123=1.23V, CC1310模组供电电压, ADC采样以该电压为参考
 *     R_T2: 分压电阻
 *     NTC_B: 负温度系数, 比如3950
 *     decimalPlaces: 保留小数位, 比如decimalPlaces=0, 返回值
 =xx°C, decimalPlaces=1, 返回值 / 10 = xx.x°C
 * @detail:
 *     由于某些原因, 无线获取到的数值并非实际值, 传参时只
 *     需将对应的无线数据传入即可, 本函数内部自动纠正
 */
int adcValueConvertToTemp( const uint16_t adcValue,
                           const uint16_t batteryValue_mV,
                           const uint16_t R_T2,
                           const uint16_t NTC_B,
                           const uint16_t decimalPlaces)
{
    int temperature;
    float R_NTC;
    float adcVotage;
    float batteryVotage;

    const float degreeKelvinK = 273.15; //开尔文绝对温度, 单位: K
    const float T2 = degreeKelvinK +25.0; //25°C的开尔文温度
    // const int R_T2 = 10000; //常温 (25°C) 下的阻值
    // const int NTC_B = 3950; //B值是负温度系数热敏电阻器的热敏常数
    const signed trim_K = 0; //温度微调系数
    //转换成单位为V
    //还原实际电压值=batteryValue_mV+200
    batteryVotage = (float)(batteryValue_mV+200) / 100;
    //将AD值转换成电压值
    //实际ADC值=adcValue*4096/4000
    adcVotage=(batteryVotage/4096)*(adcValue*4096/4000);
    //Rt = (adcVotage*10000)/(batteryVotage-adcVotage); //NTC负接, 求得电阻阻值
    R_NTC = (batteryVotage-adcVotage)*10000/adcVotage; //NTC正接, 求得电阻阻值
    //根据RT表公式求得对应的温度值
    temperature=(1/( logf(R_NTC / R_T2) / (float)NTC_B + 1 / T2) - degreeKelvinK +
(float)trim_K)*pow(10, decimalPlaces);
    return temperature;
}
```