# 方案总体设计

1. 实验背景

随着物联网（IoT）技术的快速发展，越来越多的设备被连接到互联网，实现远程监控和控制。Arduino和ESP8266是两种广泛使用的微控制器，它们具有低成本、易编程和良好的网络连接能力，非常适合用于物联网项目。

1. 实验目的

设计并实现一个基于Arduino和ESP8266的温湿度远程监测系统，目的是提供一个低成本、易于部署的解决方案，用于实时监测和记录环境温湿度数据。

1. 实现功能

（1）实时采集环境温湿度数据。

（2）通过ESP8266将数据发送到云端服务器。

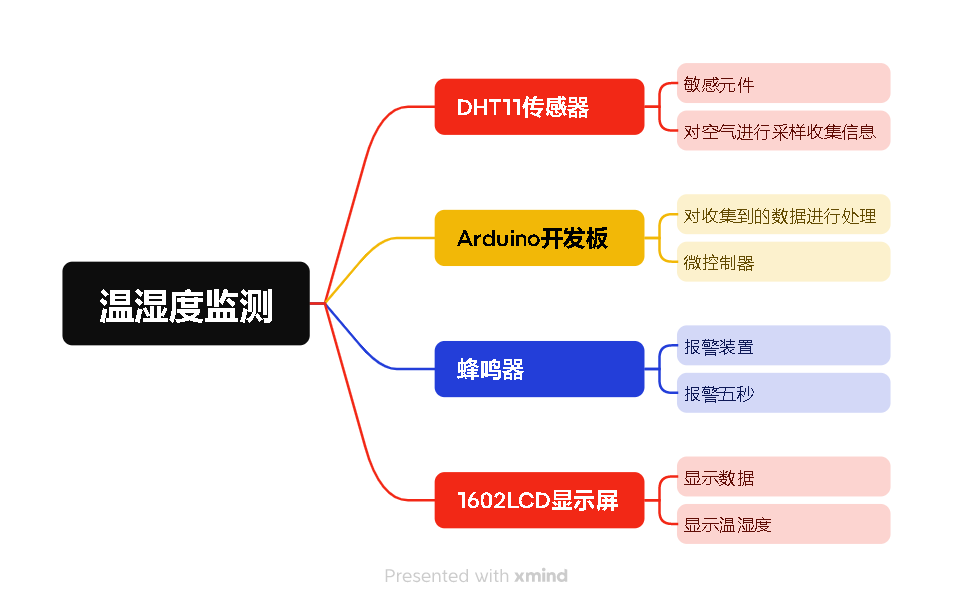
（3）提供一个用户界面，允许用户远程查看实时数据和历史记录。

# 功能分解

本次设计方案采用自底向上的设计方法，设计步骤为硬件设备到云端服务器到用户端。

1. **、底层设备**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **使用材料** | | | | | |
| **arduino开发板** | **杜邦线若干** | **DHT11传感器** | **1602LCD电子屏幕** | **三线蜂鸣器** | **ESP8266WIFI模块** |



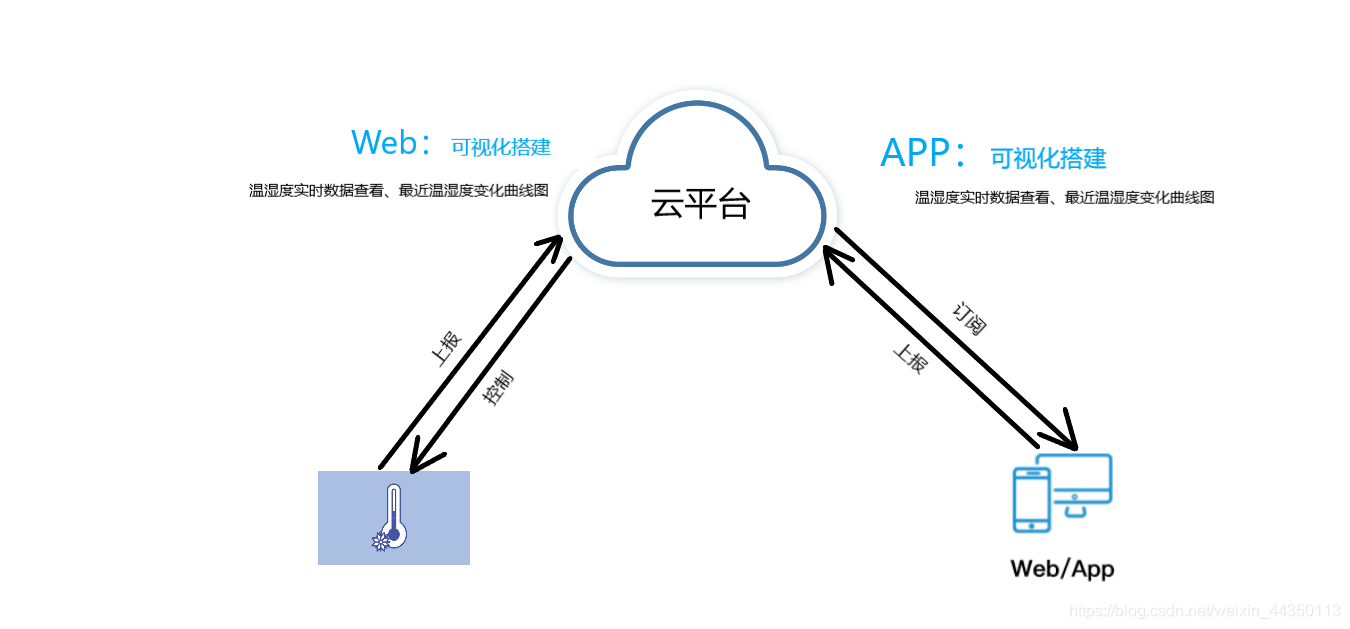
**图1 温湿度监测系统结构**

1. 、数据传输

本次实验通过ESP8266进行串口通信以及将数据发送到云端服务器。ESP8266是一款高度集成的微控制器单元（MCU），Wi-Fi SoC（System on Chip，系统级芯片）常用于物联网和智能家居。

首先使用arduino主板对收集到的数据进行处理和显示，使用串口通信将收集到的温湿度传输给ESP8266，再由ESP8266通过WiFi接入互联网，使用MQTT协议进行传输，最终再阿里云物联网控制平台显示相关数据。

本文采用阿里云云服务器中的物联网平台进行数据展示，这避免了个人网站的搭建以及服务器租赁的弊端（后期处理数据量上升可以在升级维护）



**图2实验步骤图**

1. 网络连接

使用ESP8266连接手机热点，接入互联网，ESP8266是一款由乐鑫科技（Espressif Systems）发布的低成本WiFi微控制器，它集成了TCP/IP协议栈，可用于嵌入式系统的网络连接。ESP8266搭载了Tensilica L106 32位微控制器，运行频率高达80MHz，且具备64KB的指令RAM和96KB的数据RAM，提供了足够的计算和存储资源。

本次实验使用arduino用于处理温湿度数据，并通过串口通信将数据传输至ESP8266上载至云端。

实现效果如下：



图3 WiFi连接图

使用MQTT协议进行传输数据，MQTT（Message Queuing Telemetry Transport，消息队列遥测传输）是一种轻量级的、基于发布/订阅模式的通信协议，它被设计用于连接带宽有限、网络不稳定或高延迟的网络环境中的设备。MQTT协议广泛应用于物联网（IoT）领域，因为它可以有效地在资源受限的设备上运行，并且能够确保消息的可靠传输。本次实验使用阿里云进行进行远程数据监控，后期可以通过该服务器接入轻量级的大模型进行数据预警以及日常分析和异常值的监测，具有可拓展性。

1. 云端设置

使用阿里云的物联网平台，创建公共实例



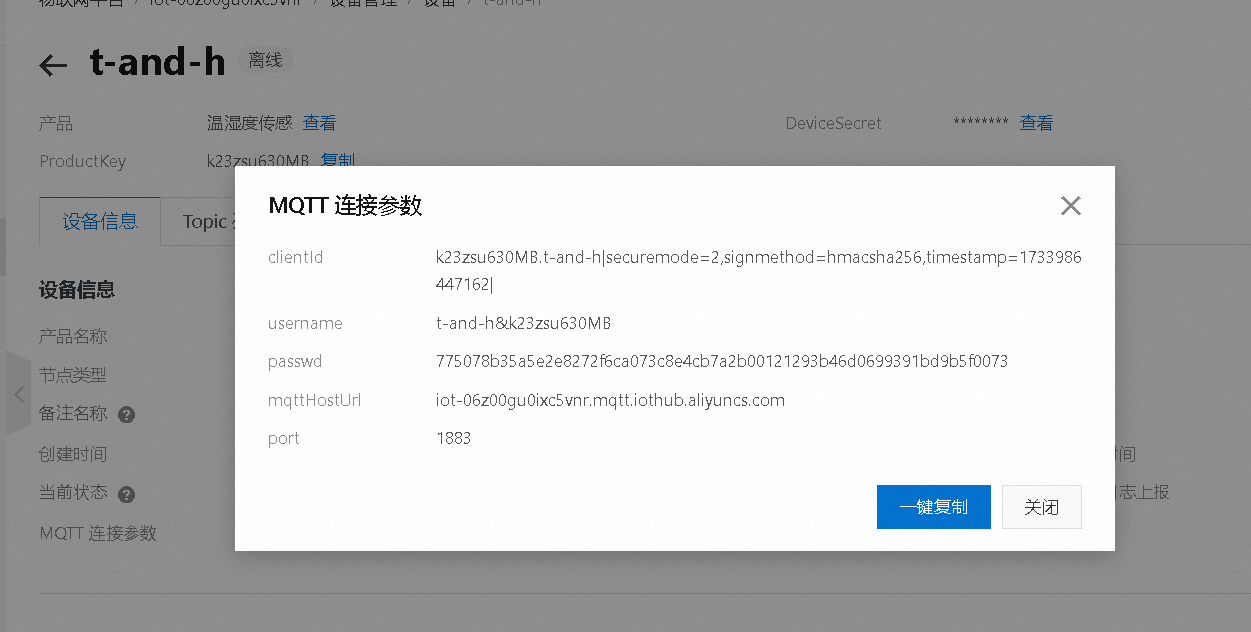
添加设备



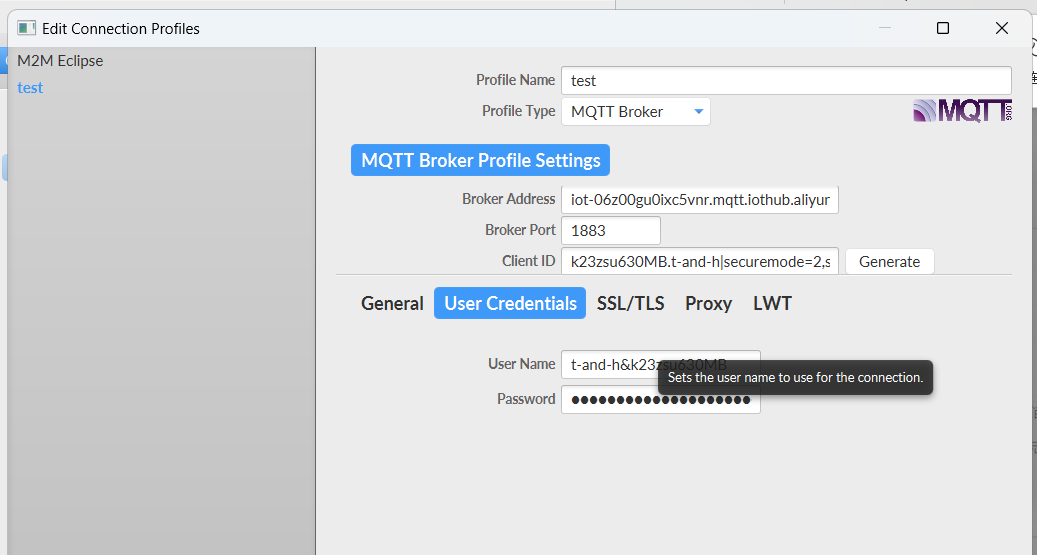
查看设备的MQTT参数以及功能标识符



查看设备的MQTT参数



得到MQTT参数后，通过mqtt.fx实现对数据的传输以及服务器的定位测试，获取服务器域名以及阿里云物联网平台的MQTT主题topic。设备可以通过发布消息到这个主题，将自身的状态或属性信息上报给云端，云端可以据此进行数据的处理和分析。



将数据打包为Alink JSON数据包格式进行传输。

Alink JSON格式：

"method": "thing.service.property.set",

"id": "1135152080",

"params": { "GeoLocation": {

"latitude": 26.0778,

"longitude": 119.3022

},

"CurrentTemperature": 13.0,

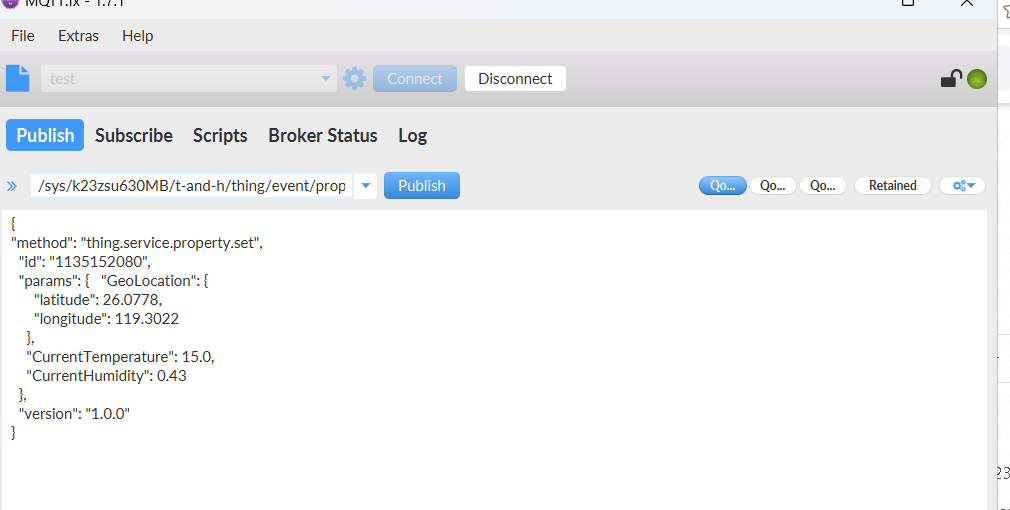
"CurrentHumidity": 0.43

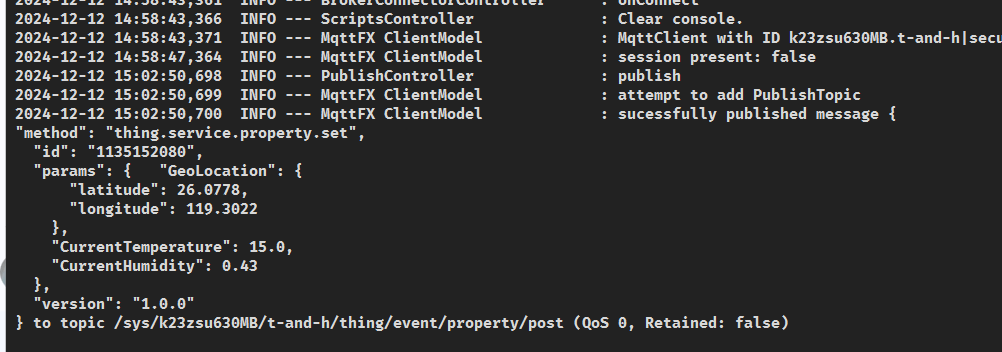
},

"version": "1.0.0"

}

进行测试





发送成功后效果如下：



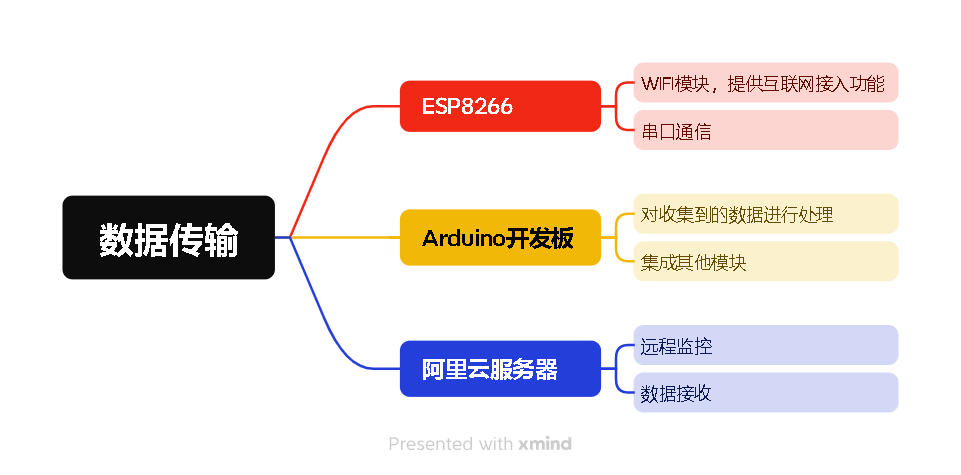
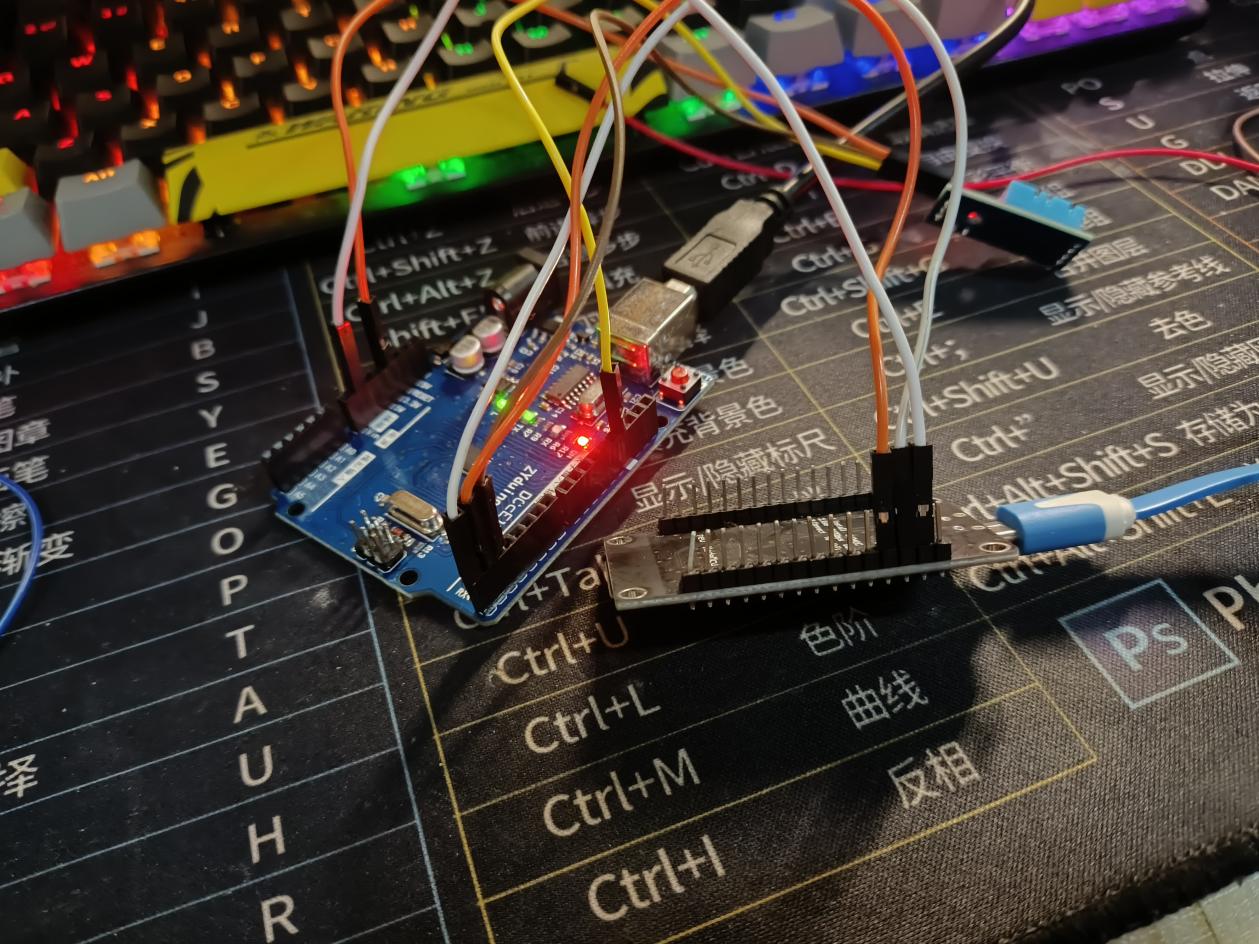
（五）硬件实现：

图4数据传输结构

# 效果实物

**（一）硬件设备**

# **4**



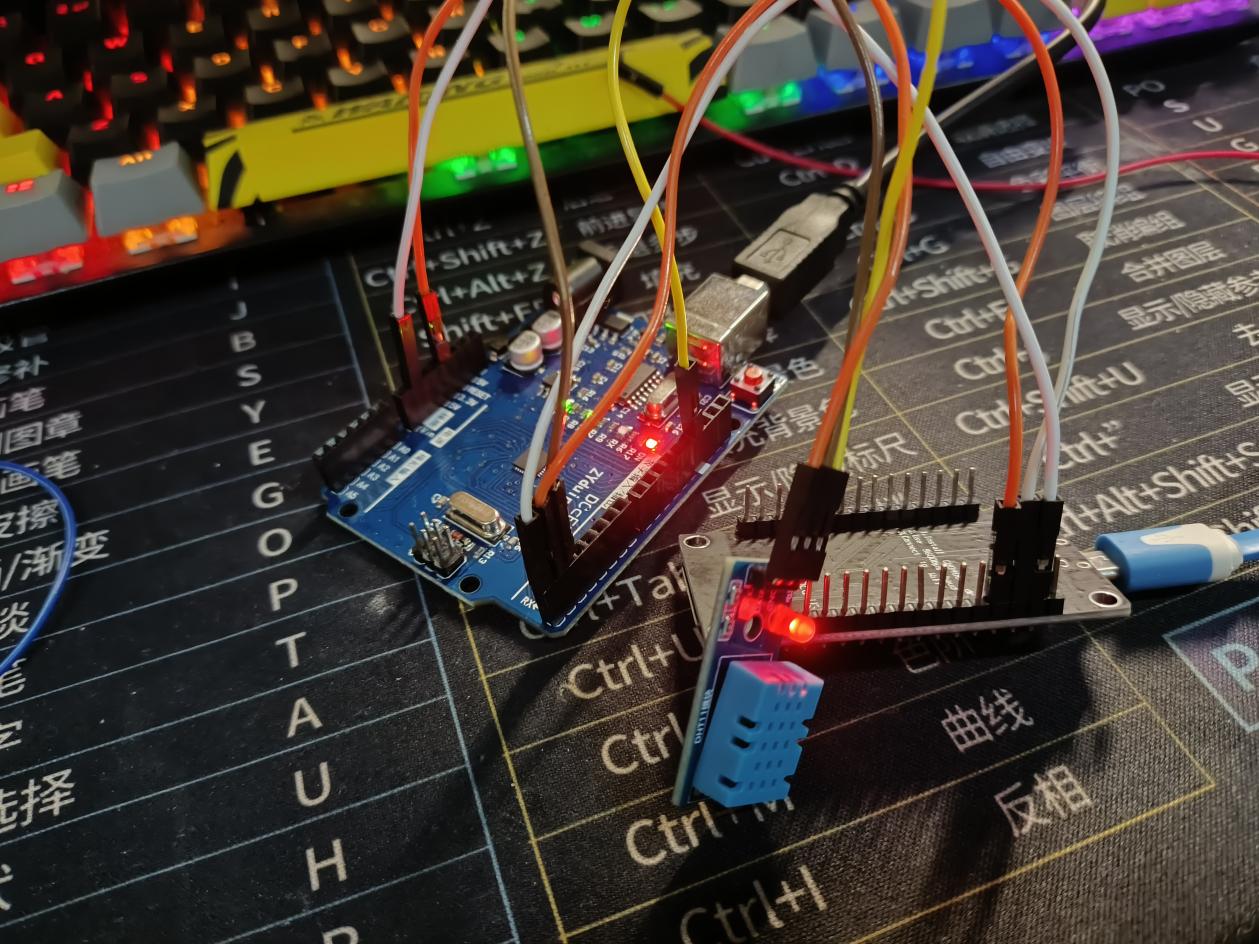


图5温湿度传感器

**（二）后台效果**



图6串口监视器效果

云平台效果（突变数据为未接入以及设备断电时导致，属于正常现象）

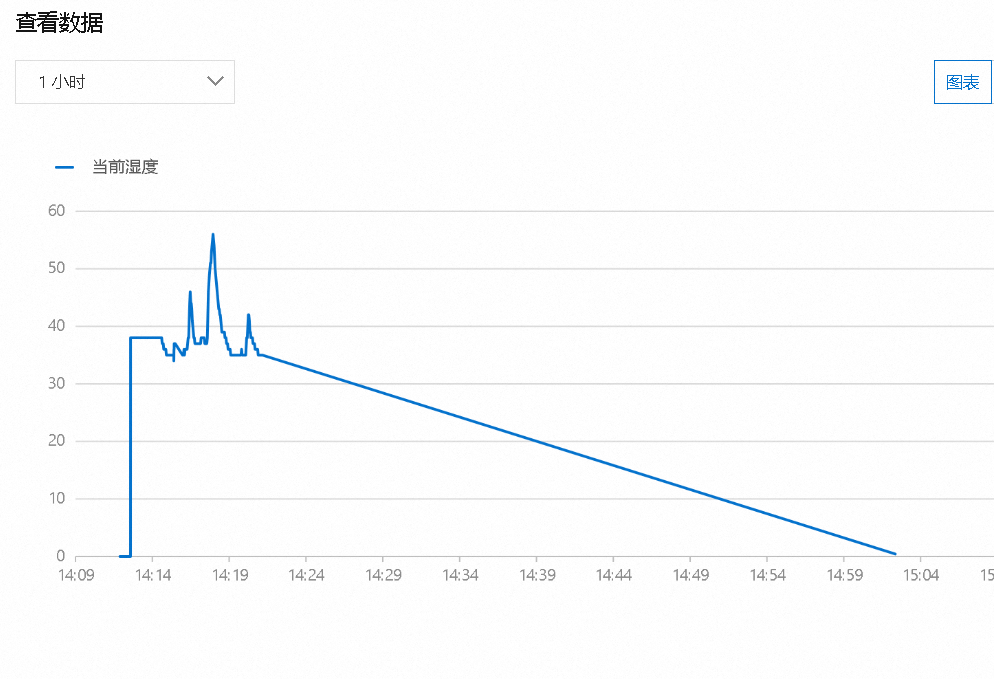


图7湿度

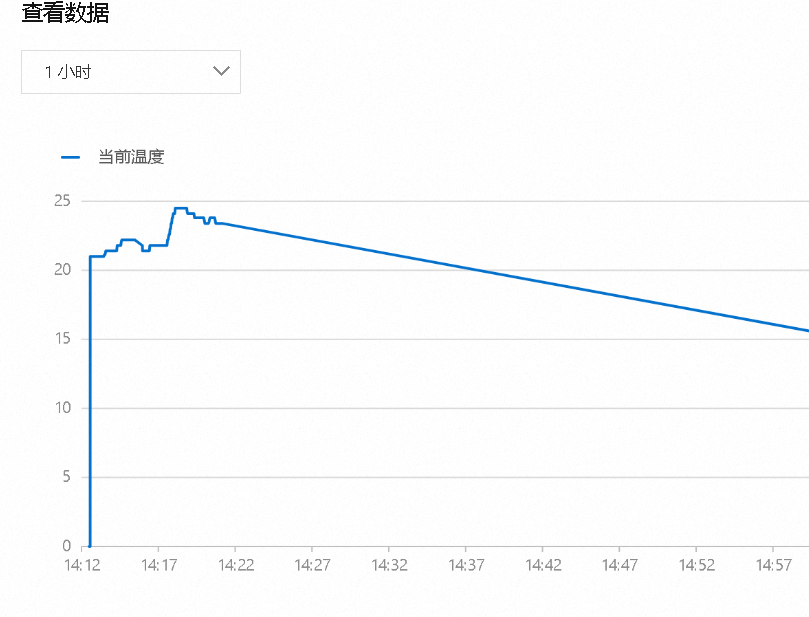


图8温度

**（三）优缺点分析：**

优点：本次实验基于arduino和ESP8266实现了温湿度的数据收集以及显示，并且基于阿里云平台使用MQTT协议上载了收集到的数据，实现了远程监控。本次实验具有实现简单，部件模块化，有可拓展性，具备一定现实意义等优势。

缺点：数据收集种类过少，远程监控平台无法自由定制，无法在线上客户端实现预警，后期可接入大模型进行实时分析预警。亦可收集多种类数据构成当地环境专有数据库，更有利于数据分析。

# 代码

Arduino 代码：

温湿度收集：

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

// 定义DHT11传感器的引脚和类型

#define DHTPIN 2     // DHT11的数据引脚连接到Arduino的数字引脚2

#define DHTTYPE DHT11 // DHT11类型

DHT dht(DHTPIN, DHTTYPE);

）

// I2C地址，一般为0x3F, 0x20或0x27，根据您的LCD模块地址而定

// 请根据您的LCD1602模块实际的I2C地址进行修改

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// 定义蜂鸣器引脚

#define BUZZER\_PIN 8

void setup() {

    Serial.begin(9600); // 启动串口通信，波特率设置为9600

    dht.begin();        // 初始化DHT11传感器

    pinMode(BUZZER\_PIN, OUTPUT); // 设置蜂鸣器引脚为输出模式

    // 初始化LCD

    lcd.init();

    lcd.backlight(); // 打开背光

    // 在LCD上显示初始文本

    lcd.setCursor(0, 0);

    lcd.print("T:    C");

    lcd.setCursor(0, 1);

    lcd.print("H:% ");

}

void loop() {

    delay(2000);

    // 读取温度和湿度值

    float h = dht.readHumidity(); // 读取湿度

    float t = dht.readTemperature(); // 读取温度

    // 检查是否有误读，DHT11时有可能发生的

    if (isnan(h) || isnan(t)) {

        Serial.println("无法从DHT11读取数据!");

        lcd.setCursor(0, 0);

        lcd.print("Error!");

        return;

    }

    // 打印结果到串口

    Serial.print("T:");

    Serial.print(t);

    Serial.println(" \*C");

    Serial.print(",H:");

    Serial.print(h);

    Serial.print(" %\t");

    // 显示温度和湿度到LCD

    lcd.setCursor(2, 0); // 移动到温度值的位置

    lcd.print(t); // 打印温度值

    lcd.print("C"); // 打印单位

    lcd.setCursor(2, 1); // 移动到湿度值的位置

    lcd.print(h); // 打印湿度值

    lcd.print("%"); // 打印单位

    // 设置温度阈值

    float tempThreshold = 20.0; // 温度阈值为20°C

    // 检查温度是否达到阈值

    if (t >= tempThreshold) {

        digitalWrite(BUZZER\_PIN, LOW); // 打开蜂鸣器

        delay(5000); // 蜂鸣器响5秒

        digitalWrite(BUZZER\_PIN, HIGH); // 关闭蜂鸣器

    }

    // 等待一段时间后再次读取

    delay(2000);

}

ESP8266代码：

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

// WiFi和MQTT配置

const char\* ssid = "Redmi K60E";

const char\* password = "66666666";

const char\* mqttServer = "iot-06z00gu0ixc5vnr.mqtt.iothub.aliyuncs.com";

const int mqttPort = 1883;

const char\* clientId = "k23zsu630MB.t-and-h|securemode=2,signmethod=hmacsha256,timestamp=1733981717187|";

const char\* username = "t-and-h&k23zsu630MB";

const char\* passwd = "f6a09b900f6fdd693111fcd34cb2ffef96bd7f4888837cfec2e7eed0c2892d0f";

WiFiClient espClient;

PubSubClient client(espClient);

void setup\_wifi() {

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println("WiFi connected");

  Serial.println("IP address: ");

  Serial.println(WiFi.localIP());

}

void reconnect() {

  while (!client.connected()) {

    Serial.print("Attempting MQTT connection...");

    if (client.connect(clientId, username, passwd)) {

      Serial.println("connected");

    } else {

      Serial.print("failed, rc=");

      Serial.print(client.state());

      Serial.println(" try again in 5 seconds");

      delay(5000);

    }

  }

}

void setup() {

  Serial.begin(9600);  // 与Arduino的串口通信波特率一致

  setup\_wifi();

  client.setServer(mqttServer, mqttPort);

}

void loop() {

  if (!client.connected()) {

    reconnect();

  }

  client.loop();

  if (Serial.available()) {

    String data = Serial.readStringUntil('\n');  // 读取串口数据

    // 假设数据格式为 "T:23.4,H:55.6"

    float temperature = data.substring(data.indexOf("T:") + 2, data.indexOf(",H:")).toFloat();

    float humidity = data.substring(data.indexOf(",H:") + 3).toFloat();

    // 构造 Alink JSON 格式的 payload，直接使用浮点数

    double latitude = 26.0778;

    double longitude = 119.3022;

    String id = "1135152080";

    String version = "1.0.0";

// 构造 Alink JSON 格式的 payload

String payload = "{\"method\":\"thing.service.property.set\",\"id\":\"" + id + "\",\"params\":{"

  "\"GeoLocation\":{\"latitude\":" + String(latitude) + ",\"longitude\":" + String(longitude) + "},"

  "\"CurrentTemperature\":" + String(temperature) + ","

  "\"CurrentHumidity\":" + String(humidity) + "},"

  "\"version\":\"" + version + "\"}";

    // 指定 MQTT 主题

    String topic = "/sys/k23zsu630MB/t-and-h/thing/event/property/post";

    // 打印生成的 payload 以便调试

    Serial.println(payload);

    // 发布数据

    if (client.publish(topic.c\_str(), payload.c\_str())) {

      Serial.println("Data published to Aliyun");

    } else {

      Serial.println("Data publish failed");

    }

  }

}