



# 高级计算网络与大数据实验课程

## 物联网综合实验报告

**Group 室验箱编号: 11**

姓名: 任智杰 学号: 11849326 Team Roles: 队长

姓名: 吕麒 学号: 11849323 Team Roles: 骨干

姓名: 朱凤天 学号: 11849305 Team Roles: 骨干

姓名: 刘斌 学号: 11849321 Team Roles: 骨干

姓名: 梁家熙 学号: 11849322 Team Roles: 普通成员

姓名: 朱传礼 学号: 11849313 Team Roles: 普通成员

## 任务 1：无线传感网络终端编程

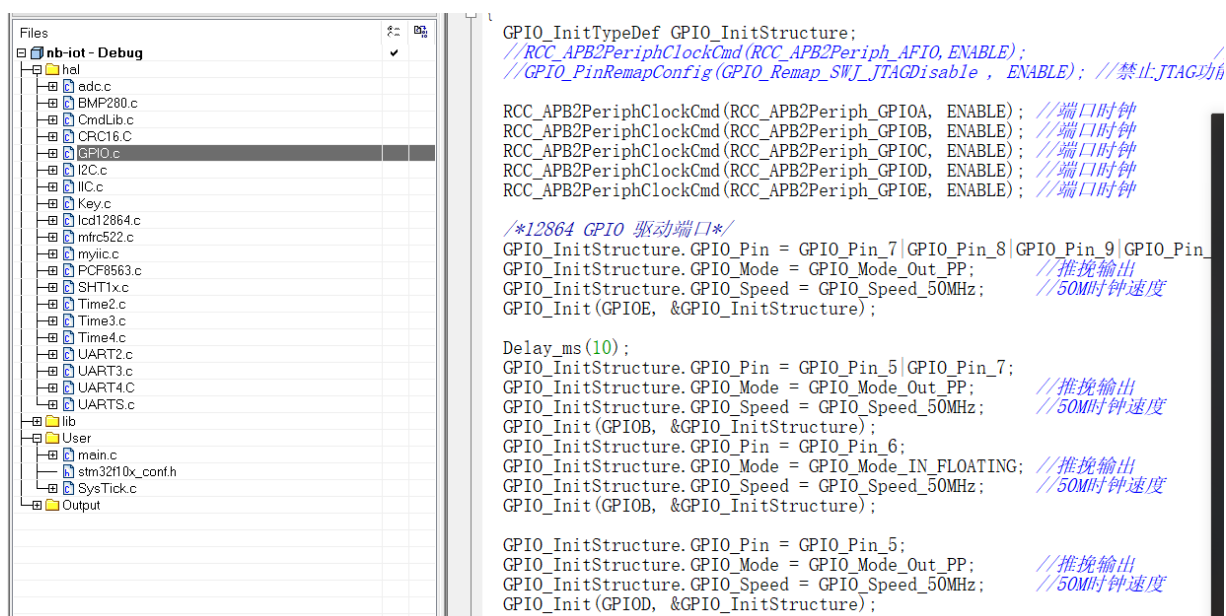
### 设计

实现智能环境数据的定时采集，包括温度、湿度、气压等数据；Pad 系统终端采集数据并通过 lora 通信发送至智慧农业系统终端；智慧农业系统终端采集数据并通过 nb-iot 无线通信将数据发送至服务器。



本任务参考实验九的终端代码部分

(1) 首先打开 IRA 软件，导入工程 PAD 系统实验部分，本部分可以直接将代码烧写进终端，不需要进行修改，如下图所示：



(2) 再导入智慧环境系统工程，更改文件 gpio.c 文件中的 IP 地址和端口号，如下图所示。

```

uint8_t DegainCheck[10] = {0},
uint8_t Tab_Local_ID[8] = {0}; //本机ID
uint8_t IP_Address[15] = {"118.126.90.148"}; //IP地址
uint8_t TCP_Port[5] = {"50011"}; //TCP端口号
uint8_t UDP_Port[5] = {"60011"}; //UDP端口号

```

更改完成后将改动保存，将程序烧写进终端，终端就开始将采集到的数据通过 nb-iot 无线通信的方式发送给服务器了。

注：本部分任务难度不大，有一个地方要注意的就是，智慧环境系统要通过 nb-iot 无线通信的方式将采集的数据发送给服务器，需要把该模块的通信模式调整到 nb-iot 模式（拨动开关即可）。



## 任务 2：云服务器开发，实现数据转发至本地 PC 机

### 设计

云服务器通过 `UdpTcpServer` 主类创建一个 UDP 服务器和一个 TCP 服务器，分别处理 UDP 连接和 TCP 连接，通过调用一个单例模式的消息队列进行通信。程序结构如图 x，其中 TCP Server、UDP Server、`TCPThreadServer`、`UDPThreadServer` 都是创建的线程，分别为 TCP 服务器、UDP 服务器、TCP 连接处理程序和 UDP 连接处理程序；`Message Queue` 是一个消息队列，用于存储消息并将 UDP 服务器收到的信息转传给 TCP 服务器。

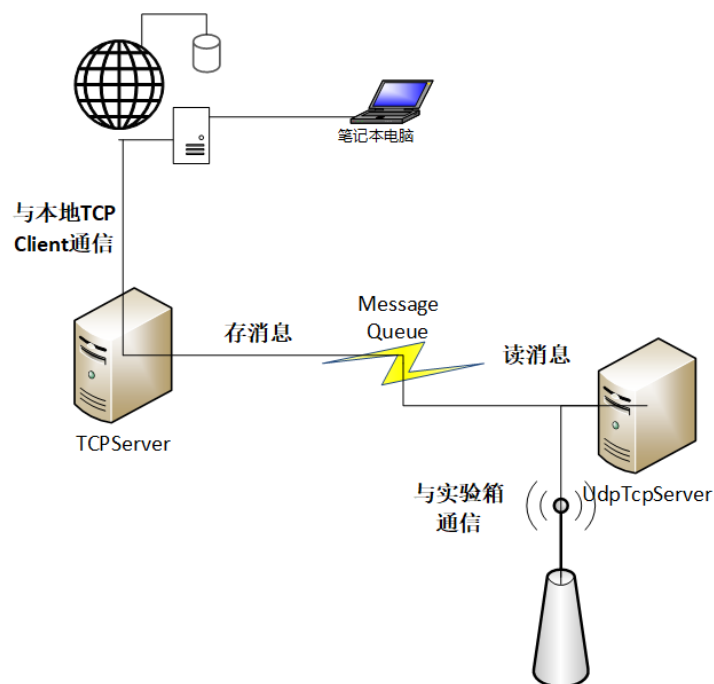


图 1：云服务器结构

### UDP 服务器

UDP Server 负责与实验箱进行通信，每次收到发送过来的数据包，UDP Server 都会创建一个线程进行处理本次数据通信。

`UDPThreadServer` 会根据收到的消息类型进行回复，并把消息存储到消息队列中。需要处理的消息类型包括：登录 0501，授时应答 0301，环境信息 0307，心跳 0402，

环境存储信息 0301。登录消息回复 ack 和授时，其他消息只保存数据，不进行回复，由本地 TCP Client 进行处理。图 X 为服务器收到数据后的输出结果，其中输出顺序乱是因为启用了新的线程进行处理，输出时可能出现交替执行的情况。

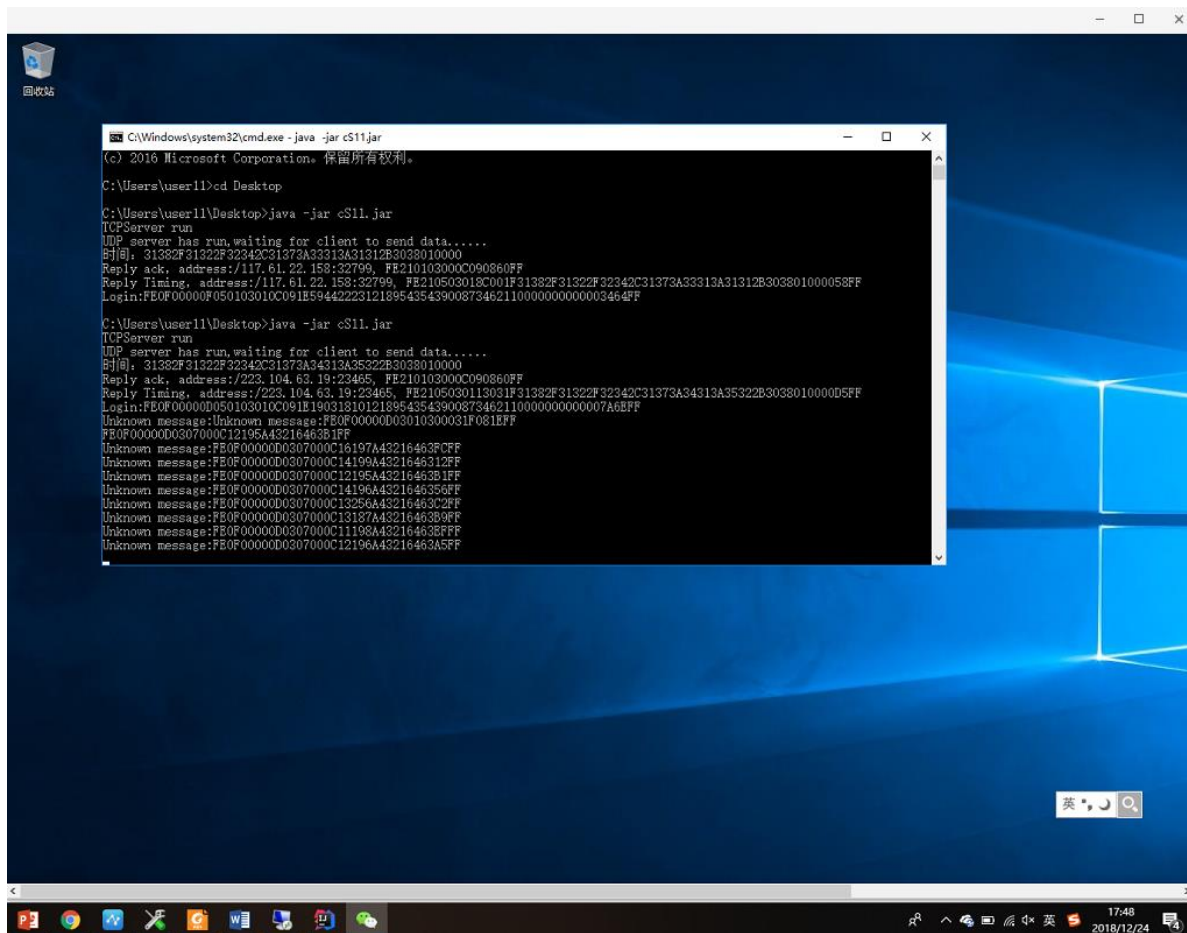


图 2：服务器收到登录和环境信息

### TCP 服务器

TCP Server 负责和本地 TCP client 进行通信，每次收到消息后创建 TCPThreadServer 进行处理。TCPThreadServe 每次收到请求就检查消息队列中是否有消息，如果有则返回一条消息，如果没有则返回一个 0xFF 表示此时没有消息。图 x 是 TCP 收到请求后的输出。

```
Run: UdpTcpServer x UDPCliet2 x TCPCliet x
Client Connect Success!
Get TCP request,Queue size:6
Client Connect Success!
Get TCP request,Queue size:5
Client Connect Success!
Get TCP request,Queue size:4
Client Connect Success!
Get TCP request,Queue size:3
Client Connect Success!
Get TCP request,Queue size:2
Client Connect Success!
Get TCP request,Queue size:1
Client Connect Success!
Get TCP request,Queue size:0
Unknown message:FE0F0000D0307000C12185A43216463F2FF
Unknown message:FE0F0000D0307000C14186A4321646315FF
Unknown message:FE0F0000D0307000C18186A43216463EAFB
Unknown message:FE0F0000D0307000C11197A43216463ABFF
Unknown message:FE0F0000D0307000C11198A43216463EFFF
Client Connect Success!
Get TCP request,Queue size:5
Client Connect Success!
Get TCP request,Queue size:4
Client Connect Success!
Get TCP request,Queue size:3
Client Connect Success!
Get TCP request,Queue size:2
Client Connect Success!
Get TCP request,Queue size:1
Client Connect Success!
Get TCP request,Queue size:0
Unknown message:FE0F0000D0307000C12186A43216463E6FF
Client Connect Success!
Get TCP request,Queue size:1
Client Connect Success!
Get TCP request,Queue size:0
Client Connect Success!
Get TCP request,Queue size:0
```

图 3：服务器收到 TCP 请求

### 任务 3:本地 TCP Client 开发，及数据存储

#### 设计

本地 TCP Client 的功能是从 TCP 服务器获取数据，根据类型进行相应的数据库更新和存储。TCP Client 使用一个无限循环，每次循环会建立一个新的 TCP 连接，从服务器中取回一条数据。如果收到的数据长度不是 1，就表示收到了消息数据，立即根据消息类型进行处理；如果收到长度为 1 的数据，就表示此时服务器中没有消息，TCP Client 会等待一段时间，如 5 分钟，为了避免在没有消息时频繁对服务器进行访问。总的流程如图 4。

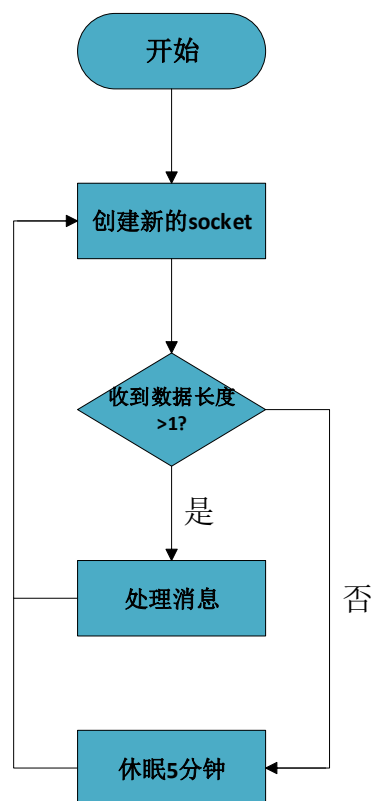


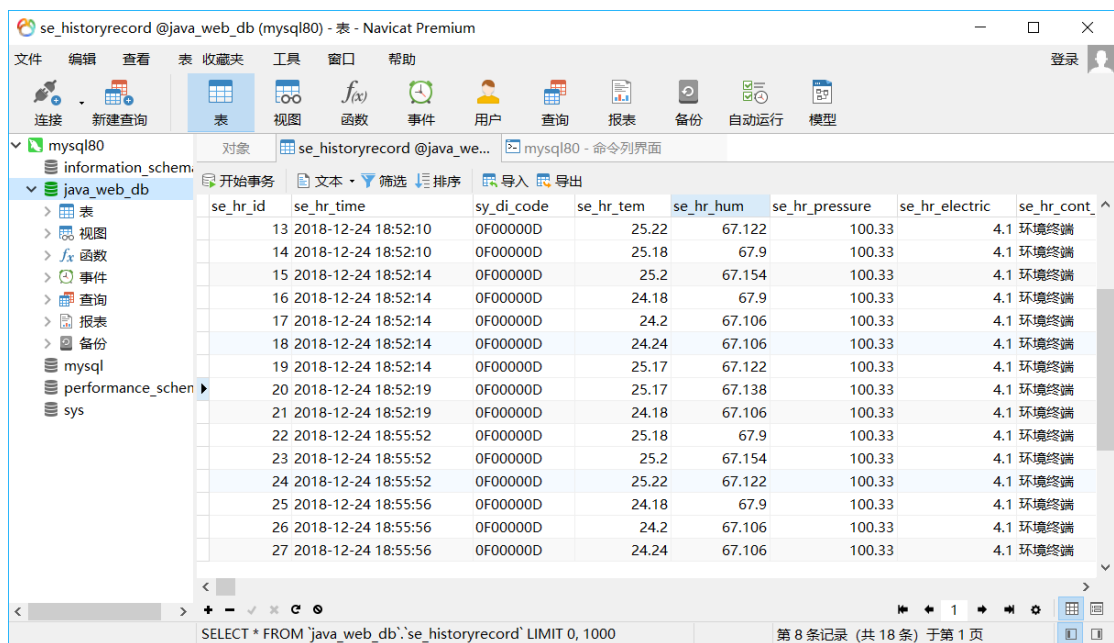
图 4: TCP client 处理流程

对于登录消息，客户端会更新设备的 ip 和状态，同时启动心跳检查；对于环境信息直接存储到数据库。任何消息都会更新心跳。图 x 是 TCP Client 收到消息后的输出。

```
Problems @ Javadoc Declaration Console
<terminated> TcpThreadServer [Java Application] S:\develop\java\setup\bin\javaw.exe (2018年12月24日 下午6:54:11)
recMsg:FE0F0000D0307000C12185A43216463F2FF
code码: 0F00000D
第8个连接,IP地址是: /10.21.43.238
recMsg:FE0F0000D0307000C14186A4321646315FF
code码: 0F00000D
第9个连接,IP地址是: /10.21.43.238
recMsg:FE0F0000D0307000C18186A43216463EAFB
code码: 0F00000D
第10个连接,IP地址是: /10.21.43.238
第11个连接,IP地址是: /10.21.43.238
recMsg:FE0F0000D0307000C11197A43216463ABFF
code码: 0F00000D
recMsg:FE0F0000D0307000C11198A43216463EFFF
code码: 0F00000D
第12个连接,IP地址是: /10.21.43.238
recMsg:FE0F0000D0307000C12186A43216463E6FF
code码: 0F00000D
```

图 5: TCP Client 收到登录和环境消息

此时，进入数据库或者从图形界面中可以查看到本地 client 已经成功接收到云端发来的数据，且环境数据已经存储到数据库。



The screenshot shows the Navicat Premium interface with a table named 'se\_historyrecord' in the 'java\_web\_db' database. The table contains 15 rows of data, each representing an environmental record. The columns are: se\_hr\_id, se\_hr\_time, sy\_di\_code, se\_hr\_tem, se\_hr\_hum, se\_hr\_pressure, se\_hr\_electric, and se\_hr\_cont. The data shows a sequence of records from 13 to 27, with timestamps and various environmental parameters.

se_hr_id	se_hr_time	sy_di_code	se_hr_tem	se_hr_hum	se_hr_pressure	se_hr_electric	se_hr_cont
13	2018-12-24 18:52:10	0F00000D	25.22	67.122	100.33	4.1	环境终端
14	2018-12-24 18:52:10	0F00000D	25.18	67.9	100.33	4.1	环境终端
15	2018-12-24 18:52:14	0F00000D	25.2	67.154	100.33	4.1	环境终端
16	2018-12-24 18:52:14	0F00000D	24.18	67.9	100.33	4.1	环境终端
17	2018-12-24 18:52:14	0F00000D	24.2	67.106	100.33	4.1	环境终端
18	2018-12-24 18:52:14	0F00000D	24.24	67.106	100.33	4.1	环境终端
19	2018-12-24 18:52:14	0F00000D	25.17	67.122	100.33	4.1	环境终端
20	2018-12-24 18:52:19	0F00000D	25.17	67.138	100.33	4.1	环境终端
21	2018-12-24 18:52:19	0F00000D	24.18	67.106	100.33	4.1	环境终端
22	2018-12-24 18:55:52	0F00000D	25.18	67.9	100.33	4.1	环境终端
23	2018-12-24 18:55:52	0F00000D	25.2	67.154	100.33	4.1	环境终端
24	2018-12-24 18:55:52	0F00000D	25.22	67.122	100.33	4.1	环境终端
25	2018-12-24 18:55:56	0F00000D	24.18	67.9	100.33	4.1	环境终端
26	2018-12-24 18:55:56	0F00000D	24.2	67.106	100.33	4.1	环境终端
27	2018-12-24 18:55:56	0F00000D	24.24	67.106	100.33	4.1	环境终端

图 6: TCP Client 收到数据后的环境信息

以上说明 TCP 客户端正确收到了服务器的数据，并且处理后存储到了数据库中。



## 任务 4:本地 java web 服务器开发

### 设计

本地 java web 服务器使用 spring+mybatis 架构，主要进行数据展示。其中数据库使用 MySQL8，图表展示使用百度 echart。程序基本参考示例代码，在其中添加了折线图展示部分。最后的页面结果见图 7-9。

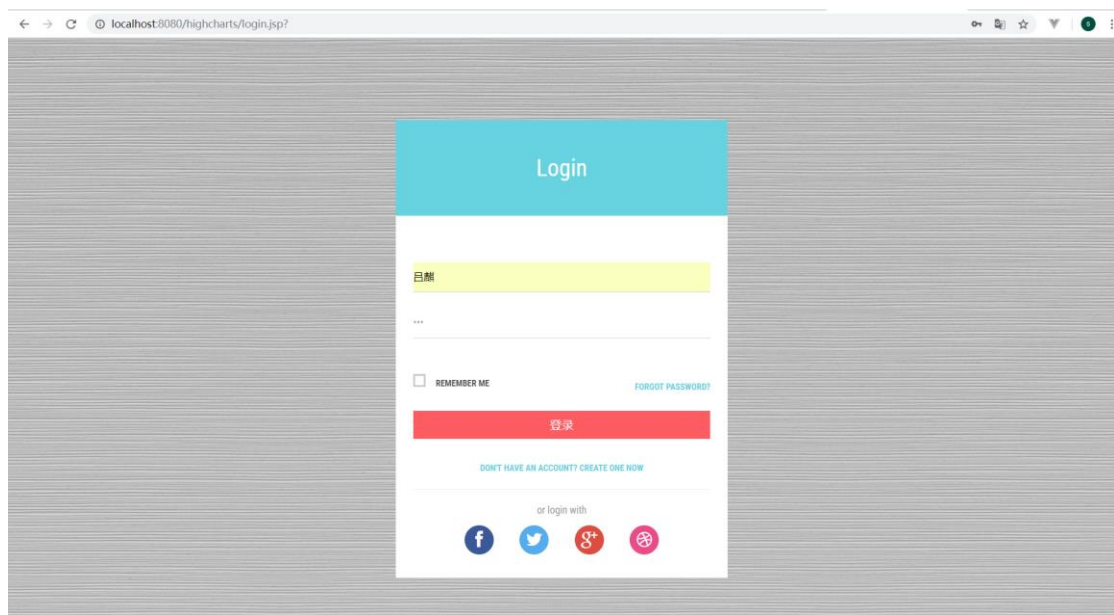


图 7：登录页面

ID	设备序列号	温度	湿度	气压	设备
13	0F00000D	25.22	67.122	100.33	环境终端
14	0F00000D	25.18	67.9	100.33	环境终端
15	0F00000D	25.2	67.154	100.33	环境终端
16	0F00000D	24.18	67.9	100.33	环境终端
17	0F00000D	24.2	67.106	100.33	环境终端
18	0F00000D	24.24	67.106	100.33	环境终端
19	0F00000D	25.17	67.122	100.33	环境终端
20	0F00000D	25.17	67.138	100.33	环境终端
21	0F00000D	24.18	67.106	100.33	环境终端
22	0F00000D	25.18	67.9	100.33	环境终端
23	0F00000D	25.2	67.154	100.33	环境终端
24	0F00000D	25.22	67.122	100.33	环境终端
25	0F00000D	24.18	67.9	100.33	环境终端
26	0F00000D	24.2	67.106	100.33	环境终端
27	0F00000D	24.24	67.106	100.33	环境终端
28	0F00000D	25.17	67.122	100.33	环境终端
29	0F00000D	25.17	67.138	100.33	环境终端
30	0F00000D	24.18	67.106	100.33	环境终端

图 8：设备列表页面

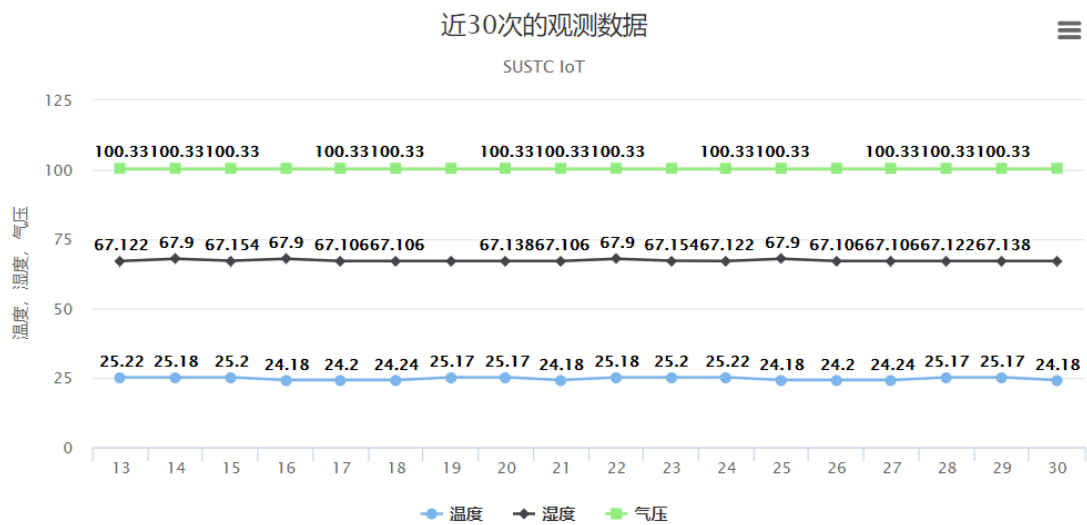


图 9：数据折线图页面