

通信电子系统设计实践

智慧信息终端



编写： 黄思甜

校对： 郝家乐

审核： 王子睿

签字： 郝家乐 23009101432

王子睿 23009100127

马雨芯 23009101151

陈昕雨 23009100077

黄思甜 23009101368

谢羽佳 23009100484

表 1 项目组成员及贡献度

序号	姓名	学号	主要完成工作	项目贡献度	备注
1	郝家乐	23009101432	RTC, 无线通信, 图形	25%	1
2	王子睿	23009100127	心率模块传感器	15%	2
3	马雨芯	23009101157	PCB, 排针焊接	15%	3
4	陈昕雨	23009100077	温湿度传感器	15%	4
5	黄思甜	23009101368	总结报告	15%	5
6	谢羽佳	23009100484	演示文稿	15%	6

目录

- 一、技术总结.....3
 - 1.1 项目任务概述3
 - 1.1.1 项目背景3
 - 1.1.2 项目目标3
 - 1.2 功能和指标3
 - 1.2.1 基本功能3
 - 1.2.2 拓展功能8
 - 1.2.3 技术指标9
 - 1.3 技术方案10
 - 1.3.1 系统结构10
 - 1.3.2 合理合规论证10
 - 1.4 电路和程序设计11
 - 1.4.1 电路设计11
 - 1.4.2 程序设计11
 - 1.5 测试结果12
 - 1.5.1 测试方法12
 - 1.5.2 测试结果12
 - 1.6 项目管理、成本分析12
 - 1.6.1 项目管理12
 - 1.6.2 成本分析13
- 二、持续改进.....13
 - 2.1 成本优化方案13
 - 2.1.1 国产代替13
 - 2.1.2 优化 PCB 设计13
 - 2.2 方案改进分析14
- 三、 参考文献.....14
 - 3.1 STM32 官方手册14
 - 3.2ESP8266 技术文档.....14
 - 3.3MPU6050 数据手册14
 - 3.4DHT11 数据手册14
- 四、附件.....14
 - 4.1 电路图与 BOM 表14
 - 4.2 会议记录14
 - 4.3 作品照片15

一、技术总结

1.1 项目任务概述

1.1.1 项目背景

在物联网快速发展的今天，智能终端已成为连接人与环境的重要纽带。本项目旨在设计一款集成 WiFi 通信、多传感器数据采集、远程控制的智慧信息终端，满足现代生活对环境监测和交互的多元化需求。

1.1.2 项目目标

设计制作一个可以抓取、显示传感器、网络信息的智慧信息终端.要求:

- ①具有丰富的图形显示界面;
- ②具备数据通信基本功能;
- ③具有一定的人机交互功能;
- ④其他扩展功能与设计。

扩展功能举例:

(1) 通过通信网络获取传感器感知数据

- ①RTC 配置时间日历
 - ②无线通信: TCP 透传进行数据抓取和控制
 - ③数据采集: MPU6050、DHT11、MAX30102
- (2)其他任意功能设计

1.2 功能和指标

1.2.1 基本功能

1.2.1.1 图形显示

显示屏显示原理:

LDC 即液晶显示器,是一种利用液晶材料的物理特性来控制光线通过能力的显示技术。LCD 的工作原理涉及电场对液晶分子排列的影响,从而控制像素点的明暗状态,显示不同颜色和灰度的图像。

如何绘制图形：

- 1)简单图形的绘制：调用(2)中所提及的函数，进行绘制。
- 2)复杂图形的绘制：对于比较复杂的显示对象(汉字、图片等)，绘制起来比较繁琐，可以利用工具将文字或图片转换 C 语言数组，然后进行显示。常见的工具有 PCtoLCD、Image2Lcd 和 TakeMold 等。

最终显示效果如图：



1.2.1.2 数据通信

TCP 透传的无限收发
进行字符比较实现功能的部分

```

332     uint8_t *buf = atk_mw8266d_uart_rx_get_frame();
333     if (buf != NULL)
334     {
335         printf("Received: %s\r\n", buf);
336         LED0_TOGGLE();
337         LED1_TOGGLE();
338         /* 如果收到"test"指令, 则翻转LED */
339         if (strcmp((char *)buf, "test", 4) == 0)
340         {
341             LED0_TOGGLE();
342             LED1_TOGGLE();
343             printf("LEDs toggled!\r\n");
344         }
345         else if (strcmp((char *)buf, "mpu", 3) == 0) // 新增的条件分支
346         {
347             // 执行MPU6050的演示函数
348             lcd_show_picture_32x32(0,185,mpu,MAGENTA);
349             demo6050_run();
350             printf("MPU6050 demo running!\r\n");
351         }
352         else if (strcmp((char *)buf, "temp", 4) == 0)
353         {
354             // 读取温度数据
355             if (dht11_read_data(&temperature, &humidity) == 0)
356             {
357                 // 显示温度
358                 lcd_show_string(40, 130, 200, 16, 16, "Temp: C", BLUE);
359                 lcd_show_num(40 + 40, 130, temperature, 2, 16, BLUE);
360             }
361             lcd_show_picture_32x32(0,125,temp,MAGENTA);
362             // 格式化数据
363             char send_buffer[64];
364             sprintf(send_buffer, "NowLoadingFromTCP...Temp:%d(Celsius)\r\n", temperature);
365             // 发送数据
366             atk_mw8266d_uart_printf(send_buffer);
367         }
368     }
369     else
370     {
371         //

```

1.2.1.3 人机交互

按键处理函数

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == KEY_Pin) {
        // 状态切换逻辑
        currentState = (currentState + 1) % 3;
        LCD_Clear(); // 清屏后再显示新内容
    }
}

```

主程序 main.c 中的按键等待来切状态的代码:

```

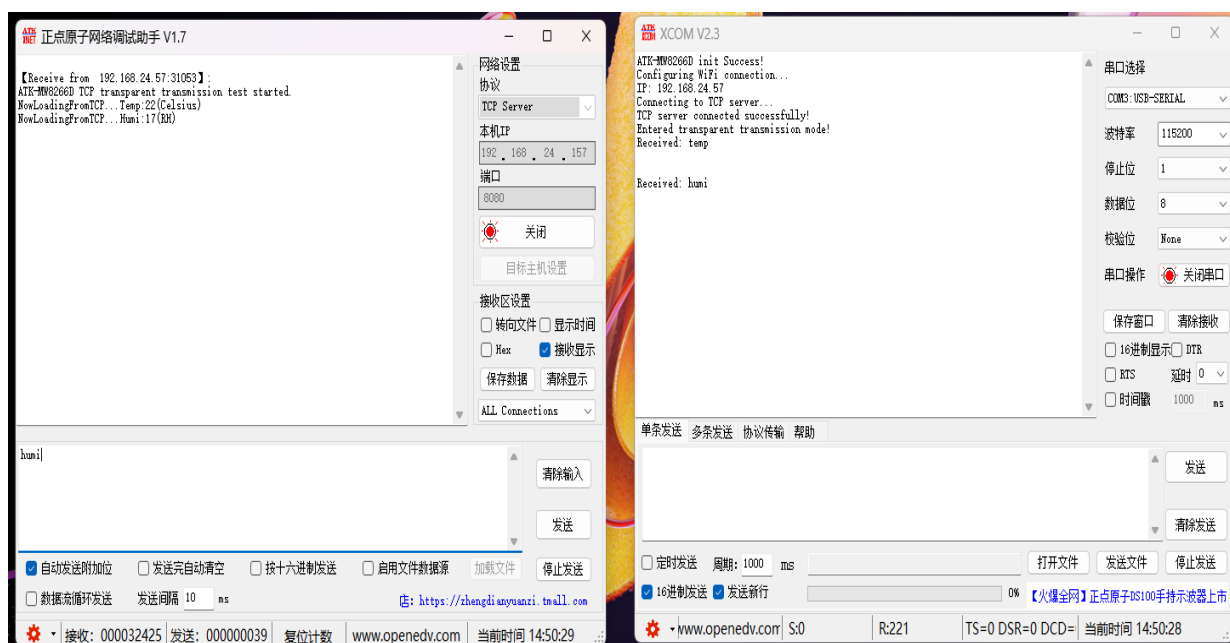
152
153 //      uint8_t tbuf[40];
154 //      uint8_t t = 0;
155     rtc_init();
156     uint8_t key;
157
158
159     while (1)
160     {
161         //      demo2_run();
162
163         key = key_scan(0); // ???
164
165
166         if (key == KEY0_PRES)
167         {
168             currentTask = TASK1;
169         }
170         else if (key == KEY1_PRES)
171         {
172             currentTask = TASK2;
173         }
174
175
176         switch (currentTask)
177         {
178             case TASK1:
179             {
180                 executeTask1(&t);
181                 break;
182             }
183             case TASK2:
184             {
185                 executeTask2();
186                 break;
187             }
188         }
189
190         delay_ms(100);
191

```

实现基本的按键人机交互功能，按键 KEY1 进入 TCP 透传模式：



通过 TCP 透传模式，分别发送 temp ， humi ， mpu 进入三种模式：



1.2.2 拓展功能

1.2.2.1 RTC 配置时间日历

RTC 是一个独立的定时器组件，相较于功能全面的通用定时器 TIM 外设，RTC 显得尤为简约，其核心功能仅限于精确计时（且同样具备触发中断的能力）。

此外，RTC 还内嵌了一个专为管理低功耗模式设计的自动唤醒模块。即便在系统断电的情境下，只要芯片的备用电源保持供电状态，RTC 仍能持续独立运作，确保其内部的时间计数器不间断地进行累加。其模块内置了一个连续运作的计数器，该计数器不仅能够实现时钟日历的基本功能，而且通过调整其计数值可以灵活重新设置当前的时间与日期信息。

本次的智慧信息终端项目采用了 RTC 模块，不仅成功实现了时间与日历信息的直观显示，更在设备断电后依然能够保持时间的精准计时功能，确保了时间的连续性和准确性。

1.2.2.2 TCP 透传进行数据抓取和控制

TCP 透传是一种高效的数据传输技术，通过该技术，数据可以在两个 TCP 连接之间无缝且透明地传输，中间服务器仅作为数据通道，不对数据进行任何修改或解析，从而确保数据的原始状态与完整性得到完美保持。

本次智慧终端项目使用 TCP 透传，通过在串口输入相应指令，切换测量模式（温度、湿度、物体运动状态），并将测量结果显示在 LCD 屏幕上。

具体实现效果如下：当按下实验板上的 KEY1 键时，LCD 显示切换至 TCP 透传模式。在串口调试助手中发送“temp”，TCP 服务器显示测量温度，并在 LCD 屏幕上显示相应温度（Temp）；在串口调试助手中发送“humi”，TCP 服务器显示测量湿度，并在 LCD 屏幕上显示相应湿度（Humi）；在串口调试助手中发送“mpu”，TCP 服务器显示物体运动状态（PIT、ROL、YAW），并在 LCD 屏幕上显示相应状态数值。

1.2.2.3 数据采集

MPU6050

在本次智慧信息终端项目中，我们特别采用了 MPU6050 传感器模块，利用其测量物体在 X、Y、Z 轴上的加速度变化和物体绕 X、Y、Z 轴的旋转速度的功能，实现了对物体绕 X 轴偏转的角度（ROL）、绕 Y 轴偏转的角度（PIT）以及绕 Z 轴偏转的角度（YAW）的精确测量，并将这些关键数据实时、清晰地显示在屏幕上。

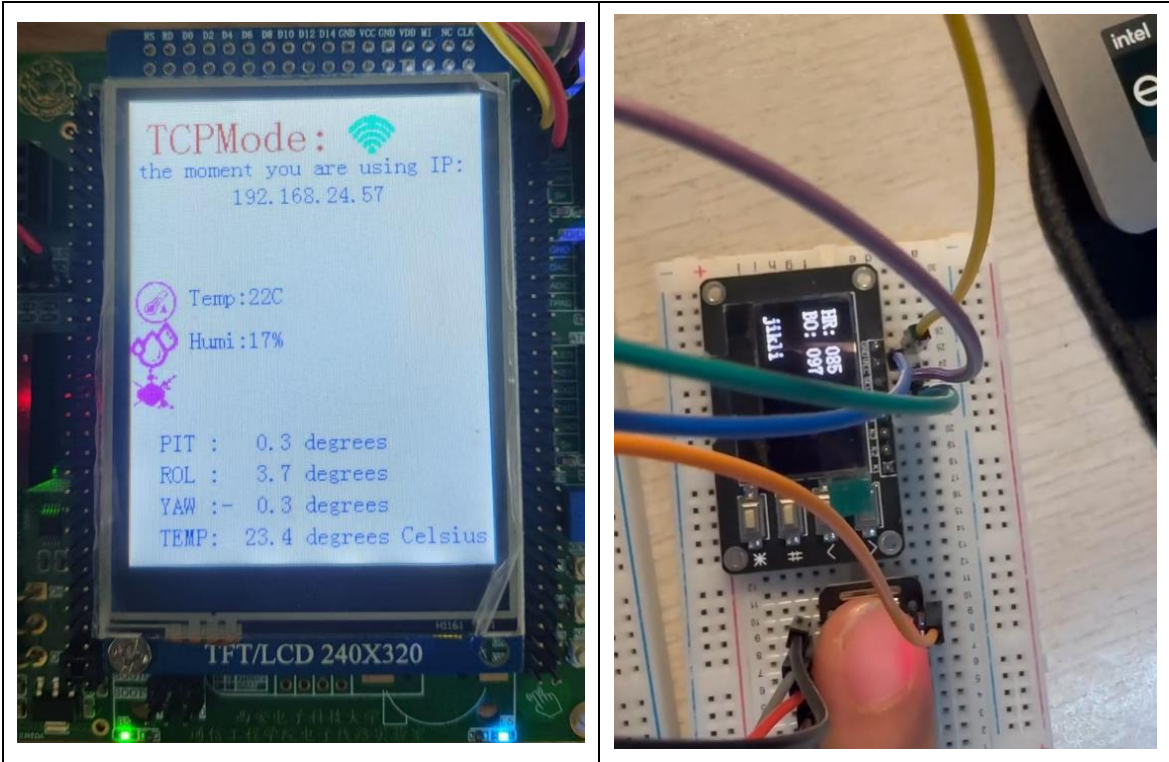
温湿度模块

在本次智慧信息终端项目的实施过程中，我们采用了 DHT11 温湿度模块，用于实时监测和记录环境中的温度和湿度数据，通过集成的高精度传感器，DHT11 能够准确捕捉环境温湿度变化，并将测量数据实时传输至 LCD 显示屏上。

心率模块

在本次智慧信息终端项目的实施过程中，我们还采用了心率模块 MAX30102，它内部配备了红外 LED、红色光 LED 和光电检测器，通过发射特定波长的光照射皮肤，再检测反射光信号，进而计算出血氧饱和度和心率，用于实时、准确地监测人体生理参数。

最终实现效果如下图所示：



1.2.3 技术指标

指标类型	具体要求	目标值
数据延迟	实时传输	<100ms
功耗水平	低功耗	<500mW
温度精度	高精确度	±0.5℃
湿度精度	高精确度	±3%RH
姿态角精度	精确测量	±1°

1.3 技术方案

1.3.1 系统结构



1.3.2 合理合规论证

选型依据

STM32: 低功耗高性能 MCU, 资源充足, 兼容性强
AKT-ESP8266: 轻量级 WiFi 模块, 支持 TCP/IP 协议栈
MPU6050: 内置姿态解算, 简化开发难度
DHT11: 低成本的温湿度传感器
MAX30102: 便携的心率和血氧测量模块

合规性分析

电路设计符合 EMC 电磁兼容性标准
元器件选用符合环保指令 RoHS 要求
供电电路满足低功耗设计规范

1.4 电路和程序设计

1.4.1 电路设计

供电电路

3.3V 稳压电源
增加滤波电容降低噪声干扰

通信电路

UART 连接 ESP8266 与 STM32

传感器接口

I2C 总线连接 MPU6050, MAX30102
单线协议连接 DHT11
信号调理电路

1.4.2 程序设计

功能模块

数据采集模块：定时读取传感器数据
TCP 通信模块：数据交换
数据处理模块：姿态解算、温湿度计算
RTC 定时模块：时钟与任务管理

软件架构

并行处理

低功耗优化

1.5 测试结果

1.5.1 测试方法

功能测试：验证各模块独立运行

集成测试：实际应用环境全功能测试

性能测试：测试数据精度、通信延时、功耗

1.5.2 测试结果

数据采集稳定，误差率符合指标

TCP 通信延时远低于 100ms，符合目标规范

远程控制响应及时

平均功耗低于设计目标

1.6 项目管理、成本分析

1.6.1 项目管理

阶段划分

需求和任务分析

方案设计

工作拆分和安排

硬件开发

软件开发

测试与调试

团队分工

硬件组（电路与 PCB 设计）：马雨芯；王子睿

软件组（程序开发）：郝家乐；陈昕雨；黄思甜
测试组（功能验证）：王子睿；谢羽佳
课程报告：小组成员陈昕雨，黄思甜，马雨芯整理相应工作，统一编排撰写
PPT 与视频展示：谢羽佳；王子睿；郝家乐

1.6.2 成本分析

元器件	单价（元）	数量	总价（元）
STM32 MCU	15	1	15
ESP8266	10	1	30
DHT11	5	1	5
MPU6050	8	1	8
MAX30102	6	1	8
PCB 制造	30	1	50
总计	--	--	116

二、持续改进

2.1 成本优化方案

2.1.1 国产代替

选用国产 MCU（微控制器）和传感器是降低成本的有效手段。与进口元件相比，国产元件通常具有价格优势，并且在供应链上更加灵活。此外，随着国内半导体产业的快速发展，国产 MCU 和传感器的性能和质量也在不断提升，能够满足多种应用场景的需求。

2.1.2 优化 PCB 设计

通过优化 PCB 设计，减少多余布线，不仅可以降低 PCB 的制造成本，还能提高生产效率。在 PCB 布局和布线过程中，应遵循简洁、高效的原则，避免不必要的走线和元件堆砌。同时，采用合理的层叠结构和布线宽度、间距等参数的选择，能够有效降低生产成本和提高产品可靠性。

2.2 方案改进分析

2.2.1 WIFI 模块可以连接正点原子厂商提供的原子云服务，联网后可以实现无线数据收发和 web，pc，移动端多独立设备场景控制

2.2.2 通过中间站代理连接，可以实现多 IP 多设备的多端控制

2.2.3 增加传感器，抓取到本地的更多环境数据

三、参考文献

3.1 STM32 官方手册

3.2ESP8266 技术文档

3.3MPU6050 数据手册

3.4DHT11 数据手册

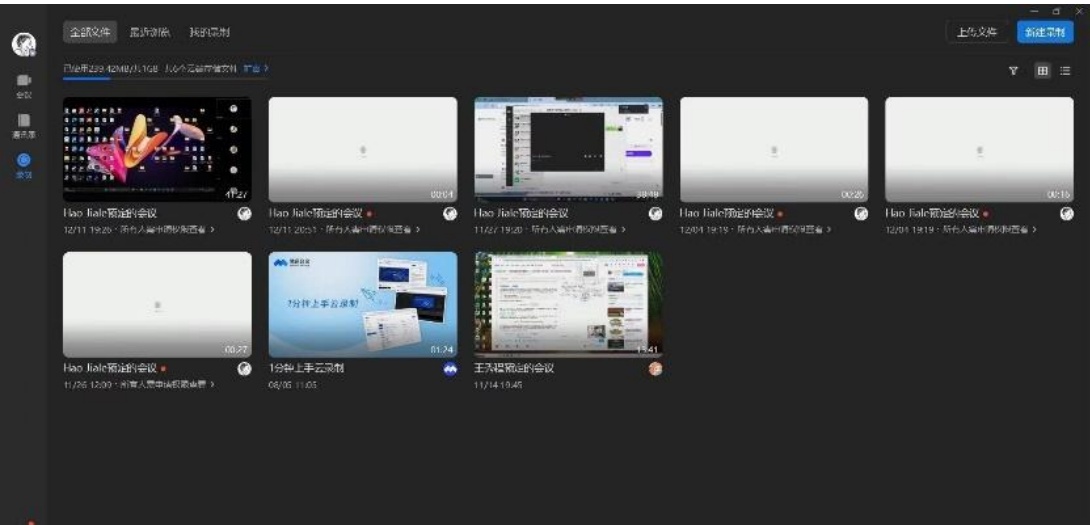
四、附件

4.1 电路图与 BOM 表

#	A	B	C	D	E	F	G	H	I	J
1	No.	Quantity	Comment	Designator	Footprint	Value	Manufacturer Part	Manufacturer	Supplier Part	Supplier
2	1	1	A254AD-WD04P	H2	HDR-TH_4P-P2.54-H-M-W10.2		A254AD-WD04P	BXCONN(宝讯)	C41413850	LCSC
3	2	2	D-060306B1	LED3, LED4	LED0603-RD		D-060306B1	ARKLED(方舟)	C118339	LCSC
4	3	2	XL-1608UVC-04	LED5, LED6	LED0603-RD_PINK		XL-1608UVC-04	XINGLIGHT(成兴光)	C7371898	LCSC
5	4	8	330 Ω	R19, R20, R21, R22, R23, R24, R25, R26	R0603	330 Ω	0603WAF3300T5E	UNI-ROYAL(厚声)	C23138	LCSC
6	5	1	10K	R27	R0603	10K				
7	6	1	STM32F103C8T6	U1	LQFP-48_L7.0-W7.0-P0.50-LS9.0-BL		STM32F103C8T6	ST(意法半导体)	C8734	LCSC
8	7	1	SN74HC595PWR	U3	TSSOP-16_L5.0-W4.4-P0.65-LS6.4-BL		SN74HC595PWR	TI(德州仪器)	C273642	LCSC
9	8	2	42-51/GHC-YW2X2K/3T	U4, U6	LED-SMD_L1.6-W0.8-FD_GREEN-1		42-51/GHC-YW2X2K/3T	EVERLIGHT(亿光)	C5136797	LCSC
10	9	2	19-21/Y2C-CP1Q2B/4T	U5, U7	LED0603-RD-YELLOW		19-21/Y2C-CP1Q2B/4T	EVERLIGHT(亿光)	C2986008	LCSC
11	10	1	8MHz	X1	CRYSTAL-SMD_L5.0-W3.2	8MHz	X50328MSB2GI	YXC(扬兴晶振)	C115962	LCSC
12										

4.2 会议记录

4.2.1 会议记录截图：



4.2.2 过程中版本更新：



4.2.3 协同工作使用的仓库

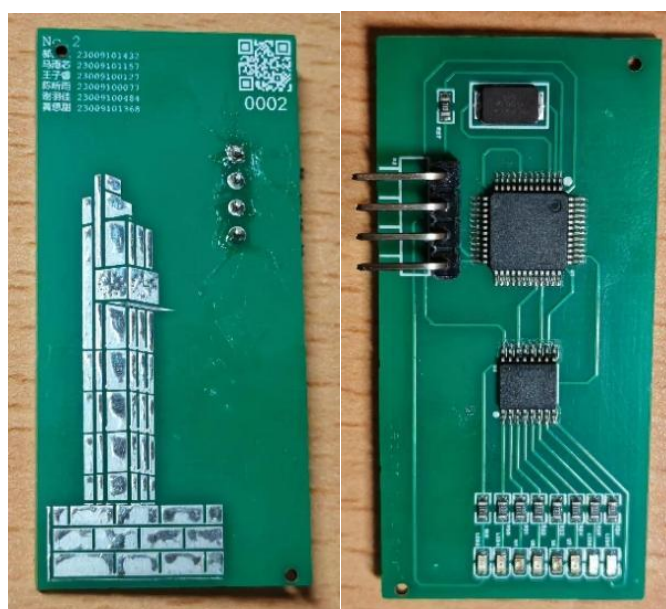


4.3 作品照片

4.3.1 产品实物图



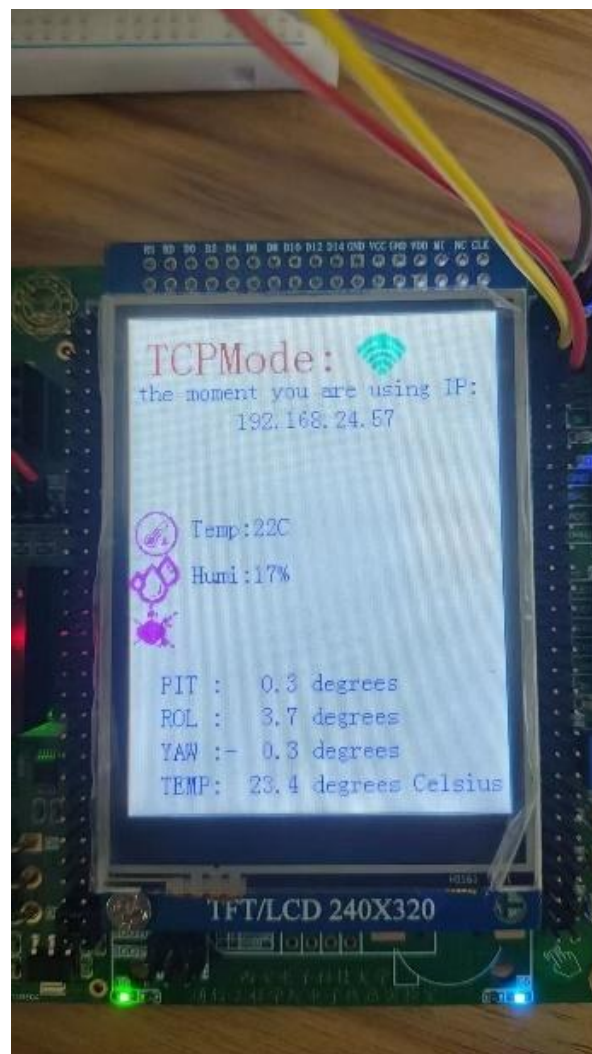
4.3.2PCB 作业实物图:



4.3.3 系统界面图:



4.3.4TCP 透传模式图:





通信电子系统设计实践

智慧信息终端



编写：____黄思甜____

校对：____郝家乐____

审核：____王子睿____

签字：_____

表 1 项目组成员及贡献度

序号	姓名	学号	主要完成工作	项目贡献度	备注
1	郝家乐	23009101432	RTC，无线通信，图形	25%	1
2	王子睿	23009100127	心率模块传感器	15%	2
3	马雨芯	23009101157	PCB，排针焊接	15%	3
4	陈昕雨	23009100077	温湿度传感器	15%	4
5	黄思甜	23009101368	总结报告	15%	5
6	谢羽佳	23009100484	演示文稿	15%	6

目录

一、技术总结.....	3
1.1 项目任务概述	3
1.1.1 项目背景	3
1.1.2 项目目标	3
1.2 功能和指标	3
1.2.1 基本功能	3
1.2.2 拓展功能	8
1.2.3 技术指标	9
1.3 技术方案	10
1.3.1 系统结构	10
1.3.2 合理合规论证	10
1.4 电路和程序设计	11
1.4.1 电路设计	11
1.4.2 程序设计	11
1.5 测试结果	12
1.5.1 测试方法	12
1.5.2 测试结果	12
1.6 项目管理、成本分析	12
1.6.1 项目管理	12
1.6.2 成本分析	13
二、持续改进.....	13
2.1 成本优化方案	13
2.1.1 国产代替	13
2.1.2 优化 PCB 设计	13
2.2 方案改进分析	14
三、参考文献.....	14
3.1 STM32 官方手册	14
3.2ESP8266 技术文档.....	14
3.3MPU6050 数据手册	14
3.4DHT11 数据手册	14
四、附件.....	14
4.1 电路图与 BOM 表	14
4.2 会议记录	14
4.3 作品照片	15

一、技术总结

1.1 项目任务概述

1.1.1 项目背景

在物联网快速发展的今天，智能终端已成为连接人与环境的重要纽带。本项目旨在设计一款集成 WiFi 通信、多传感器数据采集、远程控制的智慧信息终端，满足现代生活对环境监测和交互的多元化需求。

1.1.2 项目目标

设计制作一个可以抓取、显示传感器、网络信息的智慧信息终端.要求：

- ①具有丰富的图形显示界面；
- ②具备数据通信基本功能；
- ③具有一定的人机交互功能；
- ④其他扩展功能与设计。

扩展功能举例：

(1) 通过通信网络获取传感器感知数据

- ①RTC 配置时间日历
- ②无线通信：TCP 透传进行数据抓取和控制
- ③数据采集：MPU6050、DHT11、MAX30102

(2)其他任意功能设计

1.2 功能和指标

1.2.1 基本功能

1.2.1.1 图形显示

显示屏显示原理：

LDC 即液晶显示器，是一种利用液晶材料的物理特性来控制光线通过能力的显示技术。LCD 的工作原理涉及电场对液晶分子排列的影响，从而控制像素点的明暗状态，显示不同颜色和灰度的图像。

如何绘制图形：

- 1)简单图形的绘制：调用(2)中所提及的函数，进行绘制。
- 2)复杂图形的绘制：对于比较复杂的显示对象(汉字、图片等)，绘制起来比较繁琐，可以利用工具将文字或图片转换 C 语言数组，然后进行显示。常见的工具有 PCtoLCD、Image2Lcd 和 TakeMold 等。

最终显示效果如图：



1.2.1.2 数据通信

TCP 透传的无限收发
进行字符比较实现功能的部分

```

332     uint8_t *buf = atk_mw8266d_uart_rx_get_frame();
333     if (buf != NULL)
334     {
335         printf("Received: %s\r\n", buf);
336         LED0_TOGGLE();
337         LED1_TOGGLE();
338         /* 如果收到"test"指令, 则翻转LED */
339         if (strcmp((char *)buf, "test", 4) == 0)
340         {
341             LED0_TOGGLE();
342             LED1_TOGGLE();
343             printf("LEDs toggled!\r\n");
344         }
345         else if (strcmp((char *)buf, "mpu", 3) == 0) // 新增的条件分支
346         {
347             // 执行MPU6050的演示函数
348             lcd_show_picture_32x32(0, 185, mpu, MAGENTA);
349             demo6050_run();
350             printf("MPU6050 demo running!\r\n");
351         }
352         else if (strcmp((char *)buf, "temp", 4) == 0)
353         {
354             // 读取温湿度数据
355             if (dht11_read_data(&temperature, &humidity) == 0)
356             {
357                 // 显示温度
358                 lcd_show_string(40, 130, 200, 16, 16, "Temp: C", BLUE);
359                 lcd_show_num(40 + 40, 130, temperature, 2, 16, BLUE);
360                 lcd_show_picture_32x32(0, 125, temp, MAGENTA);
361                 // 格式化数据
362                 char send_buffer[64];
363                 sprintf(send_buffer, "NowLoadingFromTCP...Temp:%d(Celsius)\r\n", temperature);
364                 // 发送数据
365                 atk_mw8266d_uart_printf(send_buffer);
366             }
367         }
368         else
369         {

```

1.2.1.3 人机交互

按键处理函数

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == KEY_Pin) {
        // 状态切换逻辑
        currentState = (currentState + 1) % 3;
        LCD_Clear(); // 清屏后再显示新内容
    }
}

```

主程序 main.c 中的按键等待来切状态的代码:


```

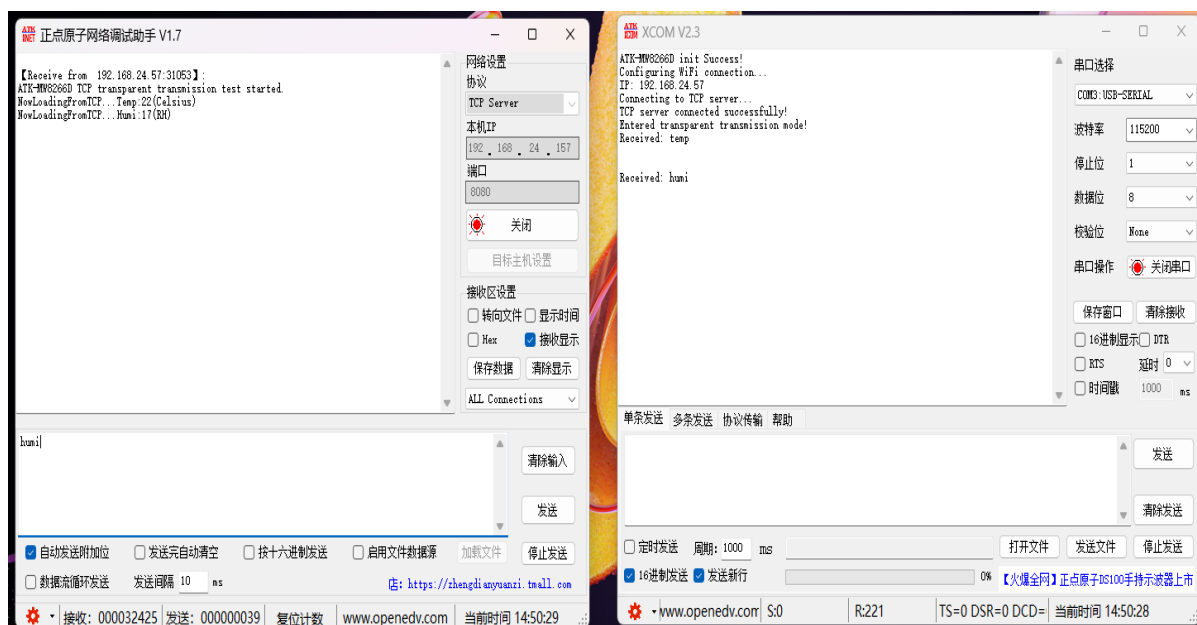
152
153 //    uint8_t tbuf[40];
154 //    uint8_t t = 0;
155 rtc_init();
156 uint8_t key;
157
158
159 while (1)
160 {
161     //    demo2_run();
162
163     key = key_scan(0); // ???
164
165
166     if (key == KEY0_PRES)
167     {
168         currentTask = TASK1;
169     }
170     else if (key == KEY1_PRES)
171     {
172         currentTask = TASK2;
173     }
174
175
176     switch (currentTask)
177     {
178         case TASK1:
179         {
180             executeTask1(&t);
181             break;
182         }
183         case TASK2:
184         {
185             executeTask2();
186             break;
187         }
188     }
189
190     delay_ms(100);
191

```

实现基本的按键人机交互功能，按键 KEY1 进入 TCP 透传模式：



通过 TCP 透传模式，分别发送 temp ， humi ， mpu 进入三种模式：



1.2.2 拓展功能

1.2.2.1 RTC 配置时间日历

RTC 是一个独立的定时器组件，相较于功能全面的通用定时器 TIM 外设，RTC 显得尤为简约，其核心功能仅限于精确计时（且同样具备触发中断的能力）。

此外，RTC 还内嵌了一个专为管理低功耗模式设计的自动唤醒模块。即便在系统断电的情境下，只要芯片的备用电源保持供电状态，RTC 仍能持续独立运作，确保其内部的时间计数器不间断地进行累加。其模块内置了一个连续运作的计数器，该计数器不仅能够实现时钟日历的基本功能，而且通过调整其计数值可以灵活重新设置当前的时间与日期信息。

本次的智慧信息终端项目采用了 RTC 模块，不仅成功实现了时间与日历信息的直观显示，更在设备断电后依然能够保持时间的精准计时功能，确保了时间的连续性和准确性。

1.2.2.2 TCP 透传进行数据抓取和控制

TCP 透传是一种高效的数据传输技术，通过该技术，数据可以在两个 TCP 连接之间无缝且透明地传输，中间服务器仅作为数据通道，不对数据进行任何修改或解析，从而确保数据的原始状态与完整性得到完美保持。

本次智慧终端项目使用 TCP 透传，通过在串口输入相应指令，切换测量模式（温度、湿度、物体运动状态），并将测量结果显示在 LCD 屏幕上。

具体实现效果如下：当按下实验板上的 KEY1 键时，LCD 显示切换至 TCP 透传模式。在串口调试助手中发送“temp”，TCP 服务器显示测量温度，并在 LCD 屏幕上显示相应温度（Temp）；在串口调试助手中发送“humi”，TCP 服务器显示测量湿度，并在 LCD 屏幕上显示相应湿度（Humi）；在串口调试助手中发送“mpu”，TCP 服务器显示物体运动状态（PIT、ROL、YAW），并在 LCD 屏幕上显示相应状态数值。

1.2.2.3 数据采集

MPU6050

在本次智慧信息终端项目中，我们特别采用了 MPU6050 传感器模块，利用其测量物体在 X、Y、Z 轴上的加速度变化和物体绕 X、Y、Z 轴的旋转速度的功能，实现了对物体绕 X 轴偏转的角度（ROL）、绕 Y 轴偏转的角度（PIT）以及绕 Z 轴偏转的角度（YAW）的精确测量，并将这些关键数据实时、清晰地显示在屏幕上。

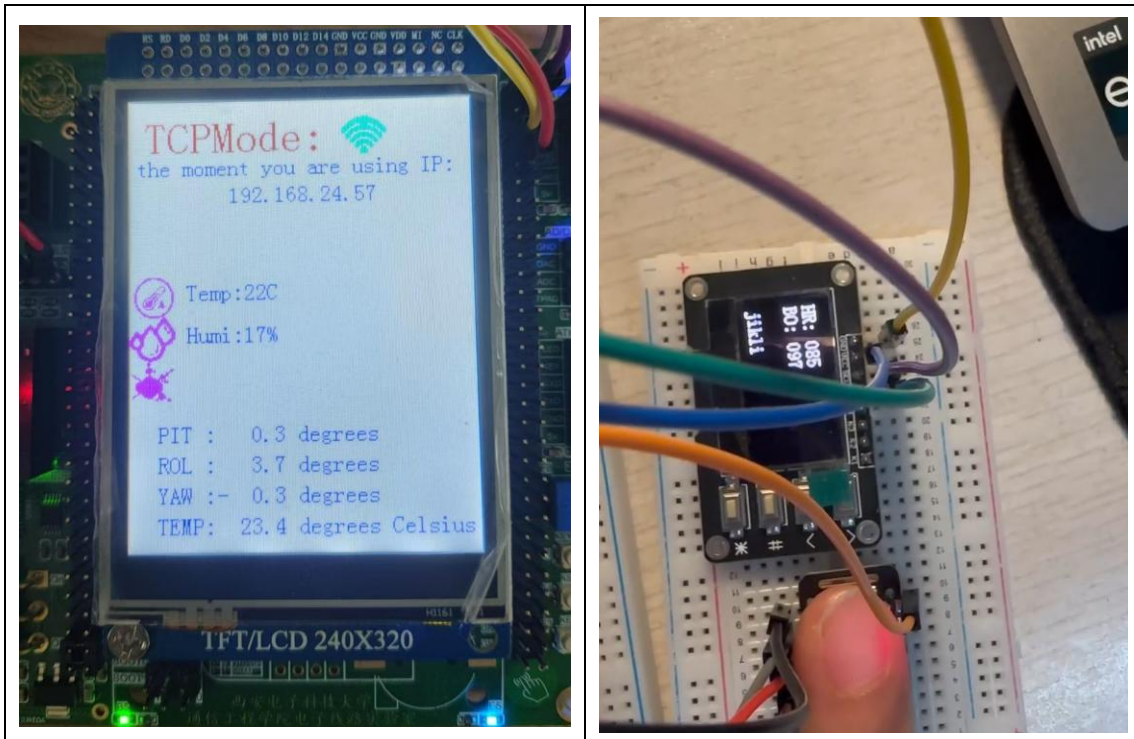
温湿度模块

在本次智慧信息终端项目的实施过程中，我们采用了 DHT11 温湿度模块，用于实时监测和记录环境中的温度和湿度数据，通过集成的高精度传感器，DHT11 能够准确捕捉环境温湿度变化，并将测量数据实时传输至 LCD 显示屏上。

心率模块

在本次智慧信息终端项目的实施过程中，我们还采用了心率模块 MAX30102，它内部配备了红外 LED、红色光 LED 和光电检测器，通过发射特定波长的光照射皮肤，再检测反射光信号，进而计算出血氧饱和度和心率，用于实时、准确地监测人体生理参数。

最终实现效果如下图所示：



1.2.3 技术指标

指标类型	具体要求	目标值
数据延迟	实时传输	<100ms
功耗水平	低功耗	<500mW
温度精度	高精度度	$\pm 0.5^{\circ}\text{C}$
湿度精度	高精度度	$\pm 3\%\text{RH}$
姿态角精度	精确测量	$\pm 1^{\circ}$

1.3 技术方案

1.3.1 系统结构



1.3.2 合理合规论证

选型依据

- STM32：低功耗高性能 MCU，资源充足，兼容性强
- AKT-ESP8266：轻量级 WiFi 模块，支持 TCP/IP 协议栈
- MPU6050：内置姿态解算，简化开发难度
- DHT11：低成本的温湿度传感器
- MAX30102：便携的心率和血氧测量模块

合规性分析

- 电路设计符合 EMC 电磁兼容性标准
- 元器件选用符合环保指令 RoHS 要求
- 供电电路满足低功耗设计规范

1.4 电路和程序设计

1.4.1 电路设计

供电电路

3.3V 稳压电源
增加滤波电容降低噪声干扰

通信电路

UART 连接 ESP8266 与 STM32

传感器接口

I2C 总线连接 MPU6050, MAX30102
单线协议连接 DHT11
信号调理电路

1.4.2 程序设计

功能模块

数据采集模块：定时读取传感器数据
TCP 通信模块：数据交换
数据处理模块：姿态解算、温湿度计算
RTC 定时模块：时钟与任务管理

软件架构

并行处理

低功耗优化

1.5 测试结果

1.5.1 测试方法

功能测试：验证各模块独立运行

集成测试：实际应用环境全功能测试

性能测试：测试数据精度、通信延时、功耗

1.5.2 测试结果

数据采集稳定，误差率符合指标

TCP 通信延时远低于 100ms，符合目标规范

远程控制响应及时

平均功耗低于设计目标

1.6 项目管理、成本分析

1.6.1 项目管理

阶段划分

需求和任务分析

方案设计

工作拆分和安排

硬件开发

软件开发

测试与调试

团队分工

硬件组（电路与 PCB 设计）：马雨芯；王子睿

软件组（程序开发）：郝家乐；陈昕雨；黄思甜
测试组（功能验证）：王子睿；谢羽佳
课程报告：小组成员陈昕雨，黄思甜，马雨芯整理相应工作，统一编排撰写
PPT 与视频展示：谢羽佳；王子睿；郝家乐

1.6.2 成本分析

元器件	单价（元）	数量	总价（元）
STM32 MCU	15	1	15
ESP8266	10	1	30
DHT11	5	1	5
MPU6050	8	1	8
MAX30102	6	1	8
PCB 制造	30	1	50
总计	--	--	116

二、持续改进

2.1 成本优化方案

2.1.1 国产代替

选用国产 MCU（微控制器）和传感器是降低成本的有效手段。与进口元件相比，国产元件通常具有价格优势，并且在供应链上更加灵活。此外，随着国内半导体产业的快速发展，国产 MCU 和传感器的性能和质量也在不断提升，能够满足多种应用场景的需求。

2.1.2 优化 PCB 设计

通过优化 PCB 设计，减少多余布线，不仅可以降低 PCB 的制造成本，还能提高生产效率。在 PCB 布局和布线过程中，应遵循简洁、高效的原则，避免不必要的走线和元件堆砌。同时，采用合理的层叠结构和布线宽度、间距等参数的选择，能够有效降低生产成本和提高产品可靠性。

2.2 方案改进分析

2.2.1 WIFI 模块可以连接正点原子厂商提供的原子云服务，联网后可以实现无线数据收发和 web，pc，移动端多独立设备场景控制

2.2.2 通过中间站代理连接，可以实现多 IP 多设备的多端控制

2.2.3 增加传感器，抓取到本地的更多环境数据

三、参考文献

3.1 STM32 官方手册

3.2ESP8266 技术文档

3.3MPU6050 数据手册

3.4DHT11 数据手册

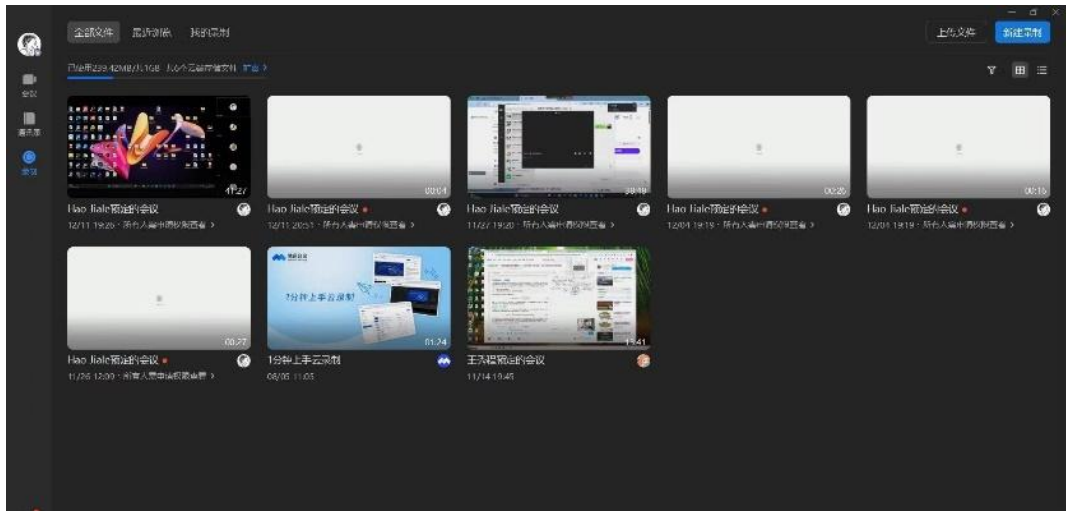
四、附件

4.1 电路图与 BOM 表

4	A	B	C	D	E	F	G	H	I	J
1	No.	Quantity	Comment	Designator	Footprint	Value	Manufacturer Part	Manufacturer	Supplier Part	Supplier
2	1		1 A254AD-WD04P	H2	HDR-TH_4P-P2.54-H-M-W10.2		A254AD-WD04P	BXCONN(宝讯)	C41413850	LCSC
3	2		2 D-060306B1	LED3, LED4	LED0603-RD		D-060306B1	ARKLED(方舟)	C118339	LCSC
4	3		2 XL-1608UVC-04	LED5, LED6	LED0603-RD_PINK		XL-1608UVC-04	XINGLIGHT(成兴光)	C7371898	LCSC
5	4		8 330 Ω	R19, R20, R21, R22, R23, R24, R25, R26	R0603	330 Ω	0603WAF3300T5E	UNI-ROYAL(厚声)	C23138	LCSC
6	5		1 10K	R27	R0603	10K				
7	6		1 STM32F103C8T6	U1	LQFP-48_L7.0-W7.0-P0.50-LS9.0-BL		STM32F103C8T6	ST(意法半导体)	C8734	LCSC
8	7		1 SN74HC595PWR	U3	TSSOP-16_L5.0-W4.4-P0.65-LS6.4-BL		SN74HC595PWR	TI(德州仪器)	C273642	LCSC
9	8		2 42-51/GHC-YW2X2K/3T	U4, U6	LED-SMD_L1.6-W0.8-FD_GREEN-1		42-51/GHC-YW2X2K/3T	EVERLIGHT(亿光)	C5136797	LCSC
10	9		2 19-21/Y2C-CP1Q2B/4T	U5, U7	LED0603-RD-YELLOW		19-21/Y2C-CP1Q2B/4T	EVERLIGHT(亿光)	C2986008	LCSC
11	10		1 8MHz	X1	CRYSTAL-SMD_L5.0-W3.2	8MHz	X50328MSB2GI	YXC(扬兴晶振)	C115962	LCSC
12										

4.2 会议记录

4.2.1 会议记录截图：



4.2.2 过程中版本更新：

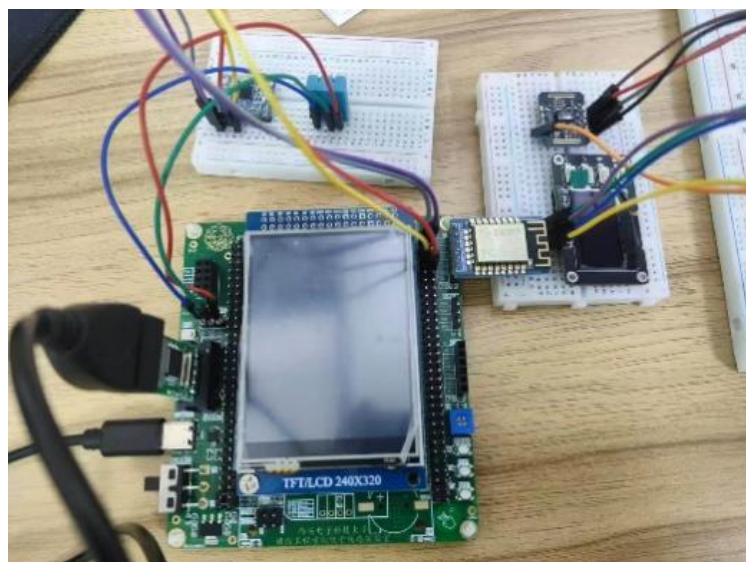


4.2.3 协同工作使用的仓库

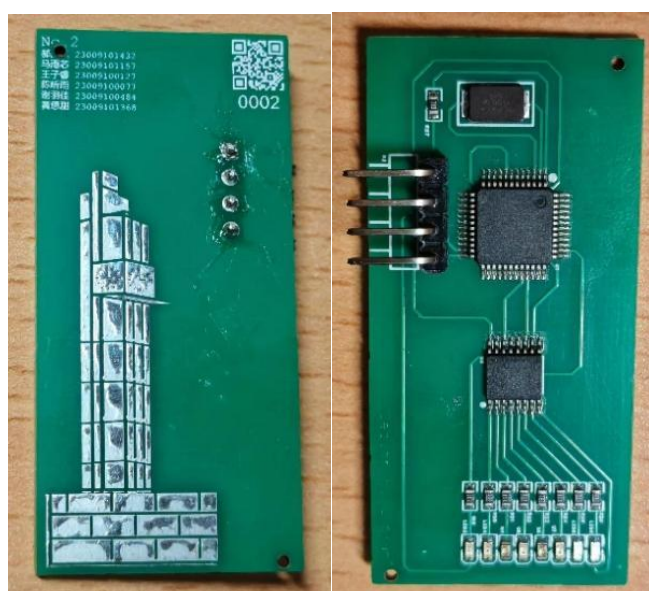


4.3 作品照片

4.3.1 产品实物图



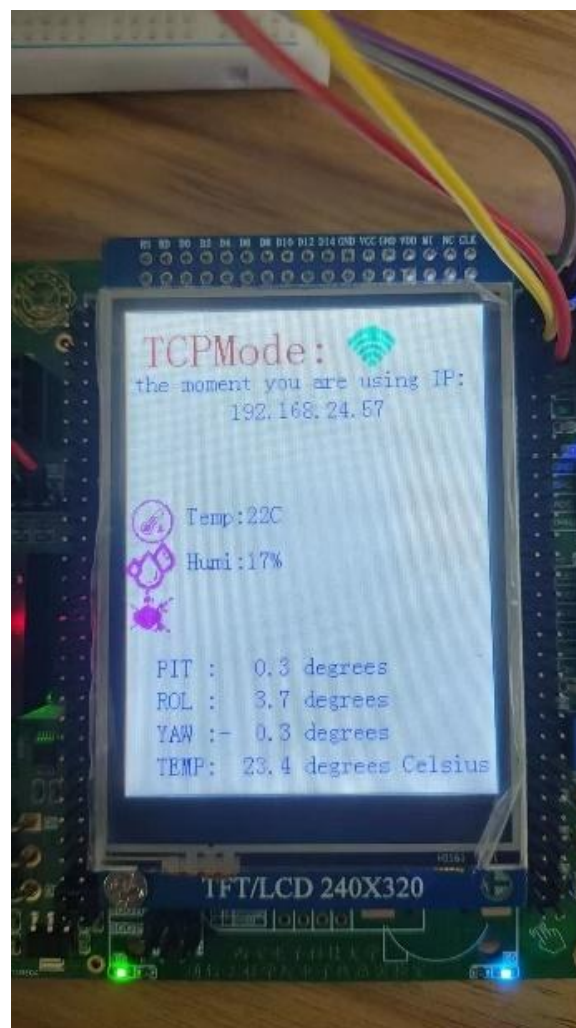
4.3.2PCB 作业实物图:



4.3.3 系统界面图:



4.3.4TCP 透传模式图:



● Part.个人总结部分：

参加“智能信息终端”项目的设计和实现，对我来说是一次难得的锻炼机会。回头想想这个过程，真是既酸爽又充实——从最初构思方案，到中途踩坑无数，再到最终通宵达旦地调试，这一路走来，每一行代码、每一块电路板都凝聚了我的心血。可以说，这次项目不仅让我学到了技术，更让我思考了“技术”以外的东西，甚至对自己的职业规划有了一些新的认知。

起初，我把这个项目当成了纯粹的任务，觉得无非是用 STM32 连接几个传感器，搞搞 WiFi 透传，再把 RTC 时间配置好。但越往后做，越发现“智慧信息终端”这个名字背后，其实是某种哲学上的隐喻：它不是一堆零件的堆砌，而是一种对智慧和信息连接方式的探讨。

就像海德格尔说的，“工具只有在被使用中才显现其本质。”我们的终端本质上就是一个工具，它的价值不在于我们用什么语言写代码、用什么芯片焊接，而在于它如何让人类生活更便捷。这种“从工具到目标”的思考，让我明白一个道理：我们学习技术的意义，不是为了造出更复杂的东西，而是让复杂的技术以简单的形式服务于人类。

这也让我对职业发展有了更深的反思。以前总觉得自己只要技术过硬，就能找到好工作，但实际上，一个优秀的工程师不只是技术专家，还需要理解产品背后的需求和价值。未来的职业道路上，我希望自己能成为一个“技术哲学家”，既掌握硬核技能，又懂得让技术真正落地。

当然，哲学思考归哲学思考，做项目时的实际情况是——坑比哲学还多！最让我抓狂的，是第一次用 TCP 透传时，数据传输总是断断续续。我对着串口打印出来的乱码，怀疑自己是不是做了一个假项目。后来调试了很久才发现，原来是 ESP8266 的波特率设置有问题。在移植的过程中，我的直接复制不起作用，在震惊和愁苦之外，我认为更有意义的是开始调试，我开始在纸上画出初始化的逻辑，从电脑到 WIFI 模块，从 WIFI 模块到 STM32 的传输，正向和反向的传输，我通过加入调试代码和打断点的方式一步步排查错误，最终让例程能融入我们的项目后顺利地跑下来，实现相应的功能。那一刻，我对“细节决定成败”这句话有了更加深刻的体会——也许世界上 99% 的 bug，都可以归结为那 1% 的粗心大意。

不过，这些踩坑的经历让我成长了很多。它们不仅教会了我如何解决问题，还让我学会了如何更从容地面对挫折。每解决一个问题，我就感觉自己又进步了一点点。这种成就感，是任何一本教科书都无法给予的。

在这个项目中，我不仅学会了具体的技术，还隐约感受到一种“生活的智慧”。比如，调试程序时需要不断尝试、迭代；做电路设计时需要权衡各方成本与效果；而协调团队成员工作时，需要把握沟通的艺术。说清楚自己的需求和敏锐的识人都是作为一个组长需要具备很重要的能力。

这些能力，其实和我们未来的职业发展密不可分。技术只是起点，而真正让人脱颖而出的，是综合解决问题的能力。通过这次项目，我更加确信自己想走的路：既要在专业领域不断深耕，又要培养跨学科的思维方式，成为一个有“技术灵魂”的职业人。

此外，这次项目还让我思考了“技术伦理”问题。比如，智慧终端的设计是否真正考虑到用户体验？它的能耗是否符合环保标准？这些问题虽然在课题中只是点缀，但放到职业生涯中，却可能成为决定一个产品成功与否的关键。未来，我希望自己能在职业生涯中始终坚持“以人为本”的理念，不仅追求技术上的领先，更追求技术的社会价值。

自嘲：

如果说这次项目还有什么特别的体验，那就是让我发现自己似乎有成为“熬夜冠军”的潜力。我很讨厌对手头当下遇到的问题进行拖延，经常干着长时间调试的勾当，我对着代码

和电路板熬到凌晨，硬是把一个 bug 变成了 feature。我戏称自己是“数字时代的钟表匠”，因为 RTC 时间模块的问题被我调得比闹钟还精准。

这段经历让我意识到，技术人虽然表面上看起来是高冷的代码战士，但实际上常常是最接地气的生活玩家。我们可以在 bug 堆里找到乐趣，也能从一个简单的调试成功中感受到满足感。这种乐观心态，大概就是我们能在技术这条路上坚持走下去的动力吧。

最后想致给未来的自己

我想对未来的自己说：无论在技术领域还是生活中，都要保持一种“进步的智慧”。智慧不只是知识的堆积，更是对世界的深刻理解和热情探索。愿我能在职业发展的路上，始终坚持初心，用技术和智慧改变世界。

这次项目的结束，也许只是另一个起点。希望未来的我，能在更多挑战中找到自己知识的缝隙和技术的空白，成为一名真正优秀卓越的工程师。