****

**单片机课程设计报告**

题 目：基于STM32的智能环境监测与控制系统

姓名： 林战

班级： 自动2302

学号： 2306050216

同组者： 朱天宝

|  |  |  |  |
| --- | --- | --- | --- |
| 平时表现30% | 成果与答辩40% | 设计报告30% | 综合评价 |
|  |  |  |  |
|  | | | |

2025年7月10日

**摘要**

本设计旨在开发一个基于STM32F103系列单片机的智能环境监测系统。该系统集成了多种传感器与外设，能够实时监测环境的温度、湿度以及有害气体浓度，并通过OLED显示屏直观地展示数据。系统具备自动和手动两种控制模式。在自动模式下，系统能根据预设的阈值自动启动风扇、LED和蜂鸣器进行声光报警和环境调节。在手动模式下，用户可以通过蓝牙串口发送指令，对风扇等设备进行远程控制。此外，系统还集成了RTC实时时钟，能够准确显示当前日期和时间，并支持通过物理按键进行校准。本设计综合运用了STM32的GPIO、ADC、PWM、UART、I2C等多种片上资源，实现了软硬件的协同工作，具有结构清晰、功能实用、可扩展性强等特点，可广泛应用于家庭、仓库、实验室等多种需要环境监测的场景。

**关键词：** STM32；环境监测；DHT11；MQ-2；蓝牙控制；RTC时钟

**1 引言**

**1.1 项目背景**

随着物联网（IoT）技术和智能家居概念的普及，人们对生活和工作环境的安全性、舒适性要求日益提高。温度、湿度和空气质量是衡量环境舒适度的关键指标，而有害气体的泄漏则直接关系到人身安全。传统的环境监测设备功能单一，无法满足智能化、网络化的需求。例如，在家庭环境中，燃气泄漏、温度过高或湿度异常等情况若不能及时发现，可能引发火灾、危害健康。在仓储或农业大棚等场景，精确的环境控制更是保证物资质量和作物生长的关键。因此，开发一套能够实时监测多种环境参数、具备自动报警与调节功能，并支持远程监控的低成本、高集成度智能系统具有重要的现实意义和应用价值。

**1.2 设计意义**

本项目基于意法半导体（ST）公司推出的高性能STM32系列微控制器，设计并实现一个多功能的智能环境监测系统。该设计不仅是对单片机课程所学知识的综合应用与实践，也是对嵌入式系统开发流程的一次完整演练。通过本项目的开发，可以深入理解STM32微控制器的工作原理，熟练掌握GPIO、UART、PWM、RTC等常用外设的配置与编程方法，以及DHT11温湿度传感器、MQ-2烟雾传感器、OLED显示屏等多种模块的驱动和数据处理技术。

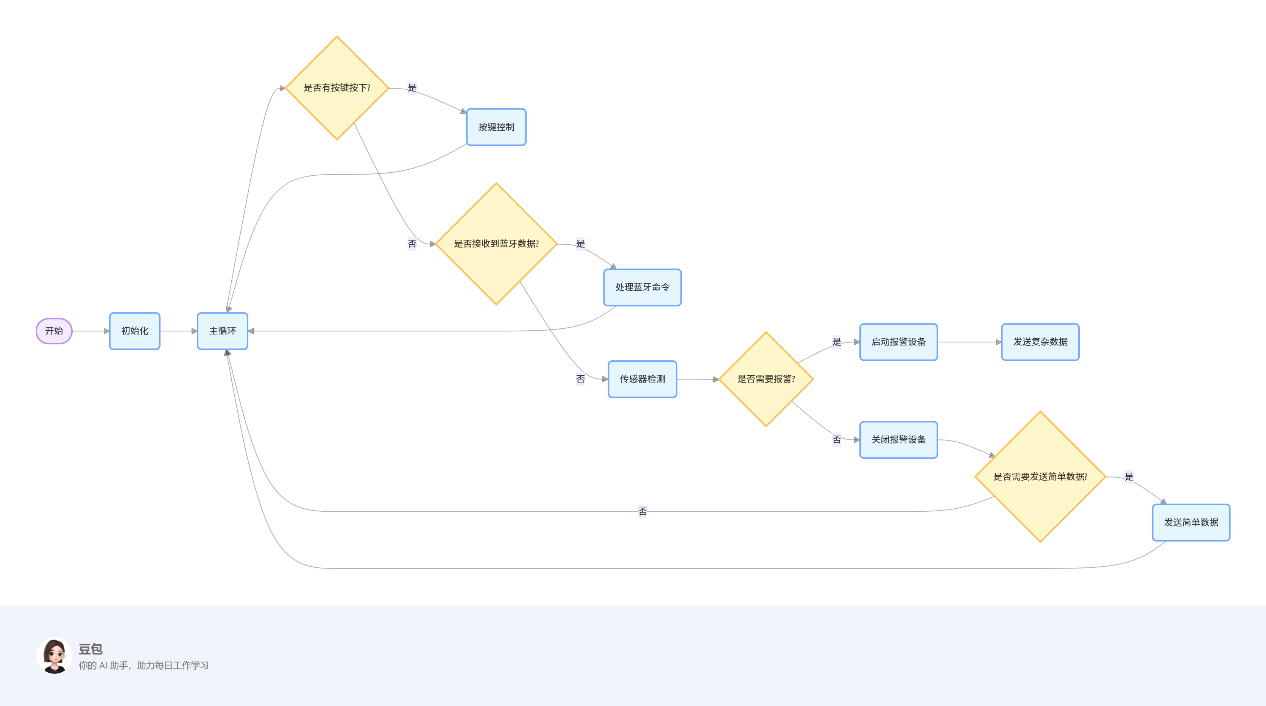
系统集成了数据采集、本地显示、自动报警、手动干预和无线通信等多种功能，构成了一个典型的物联网应用雏形。这不仅锻炼了分析问题、设计方案、编写代码和软硬件调试的综合能力，也为未来从事更复杂的嵌入式系统或物联网项目开发打下了坚实的基础。最终成品具有实用性强、成本低廉、易于扩展等优点，具备一定的市场应用潜力。

**2 方案设计**

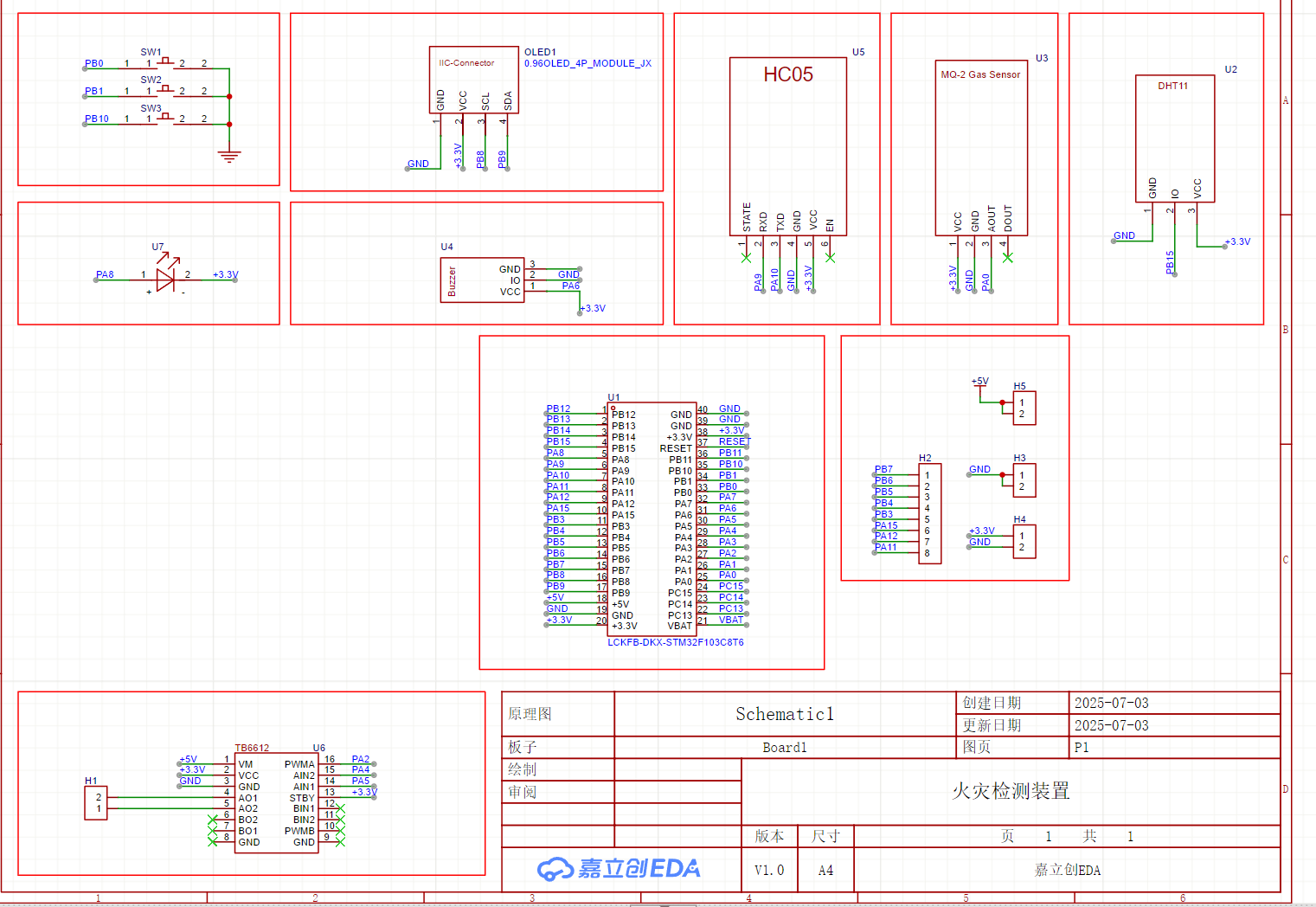
**2.1 系统总体设计**

本系统以STM32F103C8T6微控制器为核心，负责整合并控制所有硬件模块。系统分为数据采集、数据显示、数据处理与控制、人机交互四个主要部分。

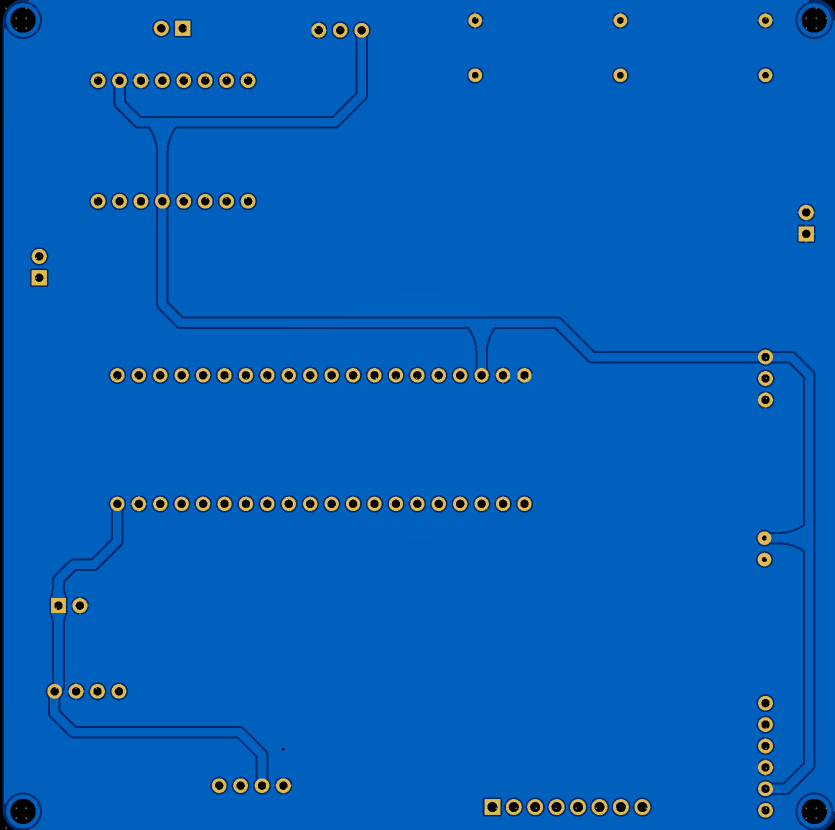
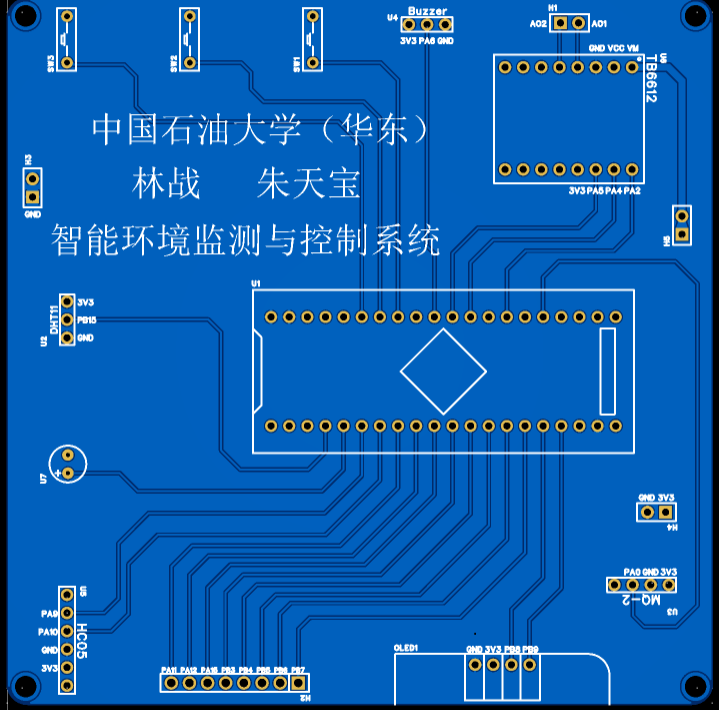
1. **数据采集模块：** 由DHT11温湿度传感器和MQ-2烟雾传感器组成，分别负责采集环境中的温度、湿度和有害气体浓度信息。
2. **数据显示模块：** 采用OLED显示屏，实时显示日期、时间、温湿度值和烟雾状态，为用户提供直观的本地监控界面。
3. **数据处理与控制模块：** 微控制器STM32获取传感器数据后，进行分析处理。一方面，将数据显示在OLED上；另一方面，根据预设的逻辑判断环境状态，并通过驱动LED、蜂鸣器和风扇等执行器，实现自动报警和环境调节。
4. **人机交互模块：** 包括物理按键和蓝牙串口。物理按键用于手动校准RTC时钟；蓝牙串口则用于与手机等智能终端连接，实现远程手动控制和数据上报。



工作流程图

**2.2 硬件选型**

**原理图**

、

PCB板正面 PCB板反面

|  |  |  |
| --- | --- | --- |
| **模块** | **型号/元件** | **选型说明** |
| **主控芯片** | STM32F103C8T6 | 基于ARM Cortex-M3内核，拥有丰富的外设接口（GPIO, UART, PWM, I2C等），性能强大，资料丰富，满足系统所有控制需求。 |
| **温湿度传感器** | DHT11 | 数字信号输出，集成温湿度测量，校准简单，成本低廉，接口方便，适合入门级环境监测。 |
| **烟雾传感器** | MQ-2 | 对液化气、丙烷、氢气等可燃气体灵敏度高，同时提供模拟和数字（阈值）输出，可用于烟雾和可燃气体泄漏检测。本项目使用其数字输出进行快速报警。 |
| **显示模块** | 0.96寸OLED | I2C/SPI接口，自发光，对比度高，功耗低，体积小，适合在小型设备上显示信息。 |
| **无线通信** | HC-05/06蓝牙模块 | 基于UART串口通信，使用简单，能方便地与手机或电脑建立无线连接，实现近距离数据传输和控制。 |
| **执行器** | LED、蜂鸣器、直流风扇 | LED用于状态指示和视觉报警，蜂鸣器用于声音报警，直流风扇（配合驱动电路）用于在温度或烟雾浓度过高时通风。 |
| **时钟模块** | STM32内置RTC | 使用外部低速晶振（LSE），可实现掉电后时间继续运行，提供精确的日期和时间信息。 |
| **输入模块** | 独立按键 | 结构简单，成本低，用于实现时间日期的设置功能。 |

**2.3 软件功能设计**

**主程序流程说明：**

1. **开始**
2. **系统初始化**：配置所有硬件模块，包括GPIO、串口、定时器、RTC、传感器等。
3. **显示UI界面**：初始化OLED屏幕，显示固定的信息框架。
4. **进入主循环 (**while(1)**)**：
   * 处理蓝牙串口接收到的数据。
   * 定时读取DHT11和MQ-2传感器的值。
   * 将最新的时间、日期和传感器数据刷新到OLED屏幕。
   * 扫描物理按键，响应用户的校时操作。
   * 执行自动报警逻辑判断。
   * 执行数据上报逻辑。
   * 短暂延时后，返回循环开始。
5. **循环往复**。

**3 开发与调试**

**3.1 硬件连接**

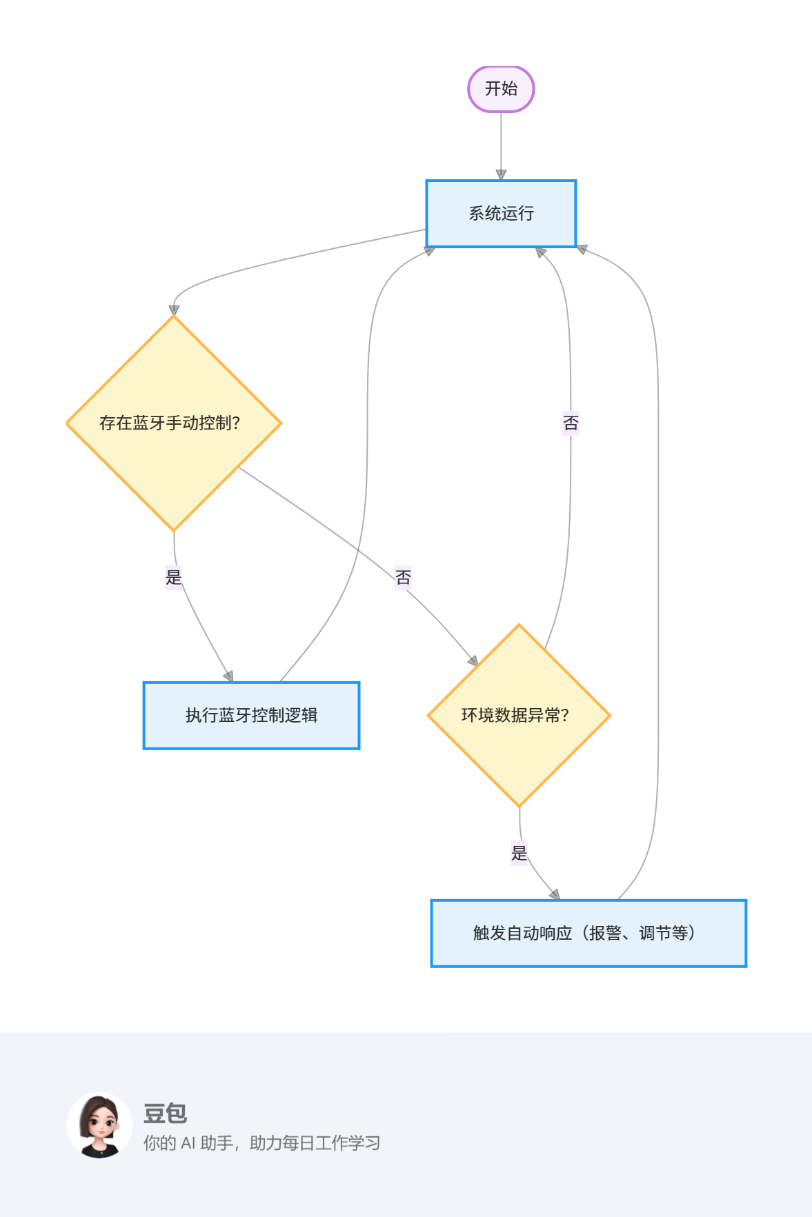
系统硬件连接是实现所有功能的基础。各模块与STM32单片机的引脚连接关系如下表所示：

|  |  |
| --- | --- |
| **模块** | **连接引脚** |
| **OLED显示屏** | (根据实际驱动选择I2C或SPI) SCL -> PB6, SDA -> PB7 |
| **DHT11** | DATA -> PA0 |
| **MQ-2** | DOUT -> PA1 |
| **LED** | LED -> PA8 |
| **蜂鸣器** | BEEP -> PA6 |
| **风扇PWM** | PWM -> PA2 |
| **风扇方向** | IN1 -> PA4, IN2 -> PA5 |
| **蓝牙模块** | TXD -> PA10 (MCU RX), RXD -> PA9 (MCU TX) |
| **按键** | KEY1 -> PB11, KEY2 -> PB10, KEY3 -> PB1, KEY4 -> PB0 |
| **RTC晶振** | LSE -> PC14, PC15 |

**3.2 软件核心逻辑**

**3.2.1 自动报警与控制流程**

这是系统的核心智能体现。当环境异常时，系统自动做出反应。该逻辑仅在未被蓝牙手动控制时生效，手动控制拥有更高优先级。其工作流程如图3-1所示。

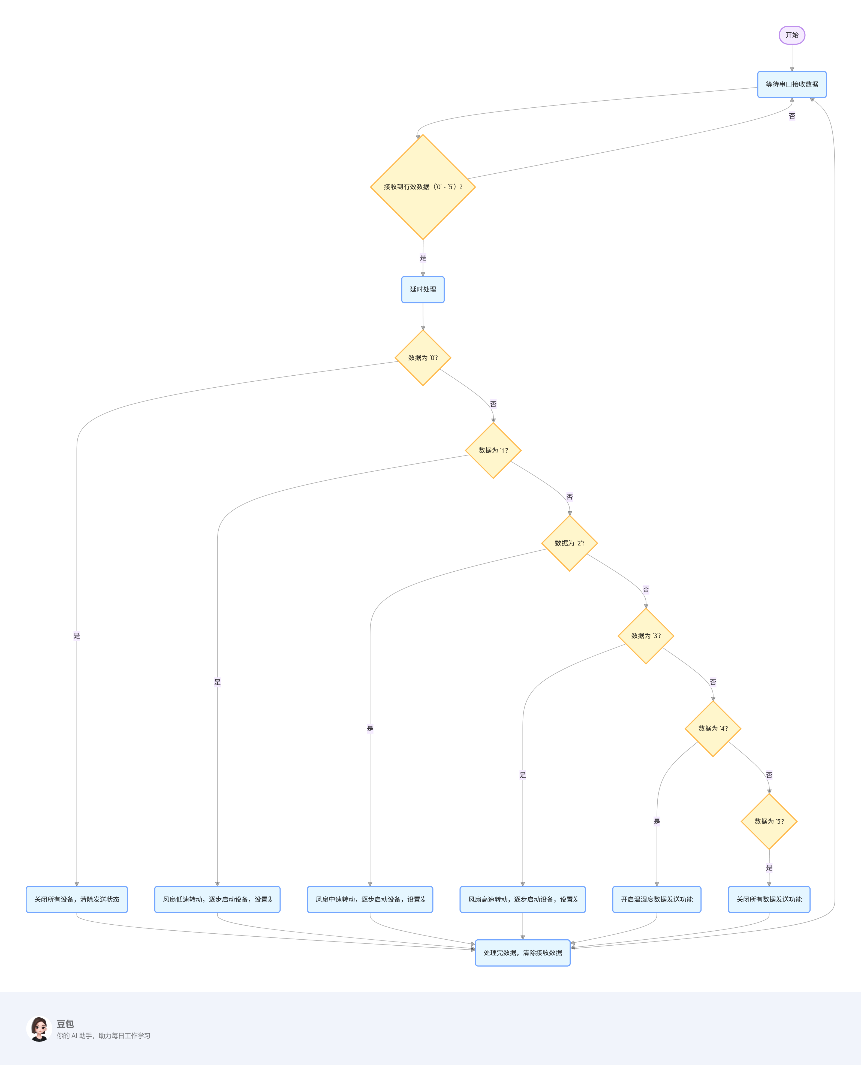


3-1蓝牙控制和环境监测流程图

**流程说明：**

1. 检查系统是否处于蓝牙手动控制模式。如果是，则退出自动逻辑。
2. 如果不是手动模式，则依次判断是否满足高、中、低三个报警等级的条件。
3. **高级报警** ( 温度 > 35℃): 启动蜂鸣器、LED，并使风扇高速转动。
4. **中级报警** ( 温度 > 30℃): 启动蜂鸣器、LED，并使风扇中速转动。
5. **低级报警** ( 温度 > 28℃ 或 MQ-2检测到烟雾): 启动蜂鸣器、LED，并使风扇低速转动。
6. 若所有条件均不满足，则判断为环境正常，关闭所有执行器（蜂鸣器、LED、风扇）。

**3.2.2 蓝牙远程控制流程**

****

3-2蓝牙远程控制流程图

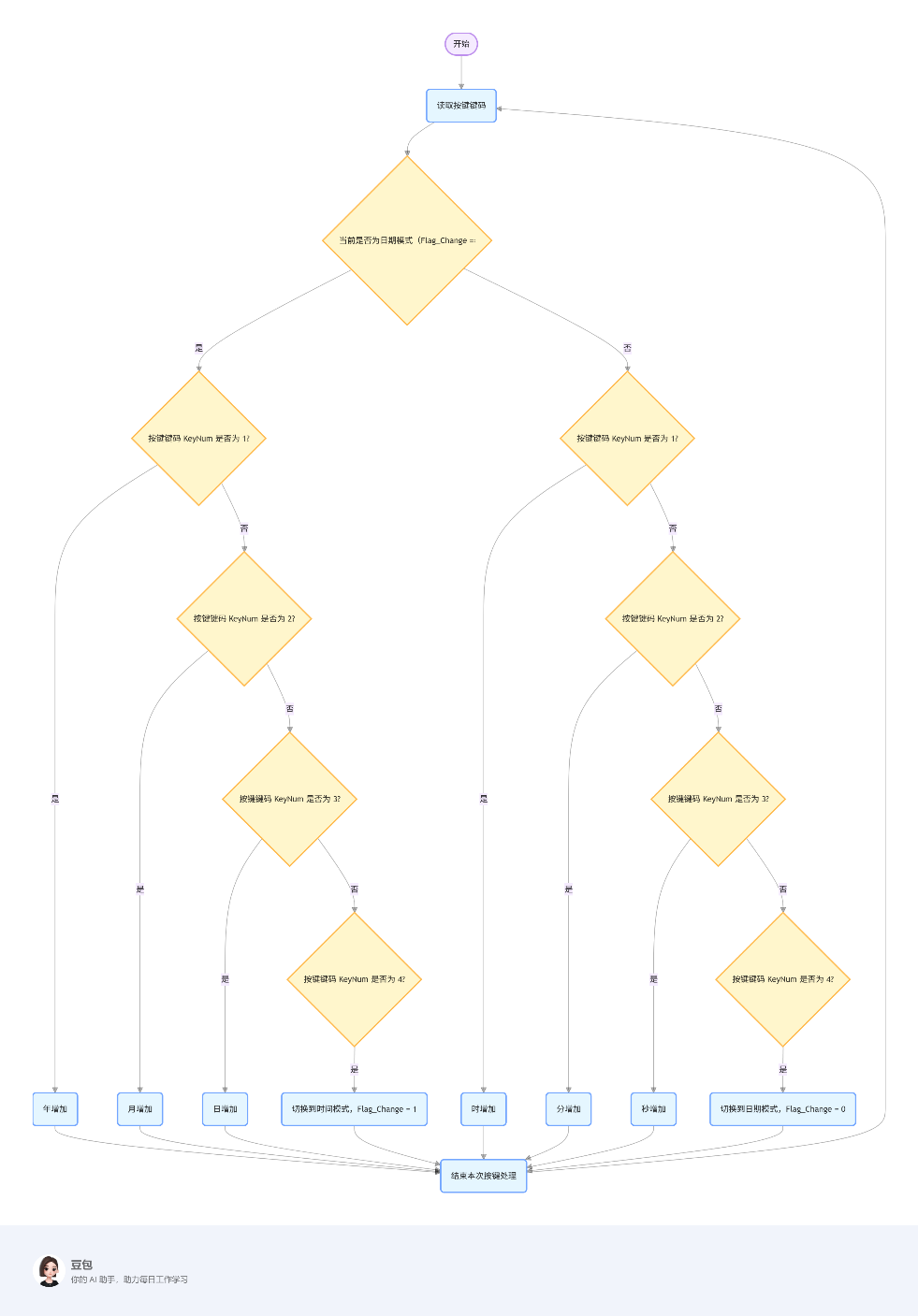
**流程说明：**

1. 系统监听串口，判断是否接收到新数据。
2. 接收到数据后，根据数据内容进行分支判断：
   * **指令 '1'/'2'/'3'**: 进入手动控制模式，并根据指令分别以低/中/高速启动风扇和报警器，同时准备向手机发送一条包含所有设备状态的完整数据。
   * **指令 '0'**: 退出手动控制模式，关闭所有执行器。
   * **指令 '4'**: 开启数据自动上报功能，系统将每隔一段时间（5秒）发送一次温湿度和时间数据。
   * **指令 '5'**: 关闭数据自动上报功能。

**3.2.3 RTC时间管理逻辑**

RTC时间的管理利用了C语言标准库<time.h>中的时间戳机制。

* **设置时间**: 将用户通过按键设定的年月日时分秒（MyRTC\_Time数组）填充到struct tm结构体中，调用mktime()函数将其转换为一个长整型的时间戳。考虑到时区差异，将此时间戳减去8小时后，写入STM32的RTC计数器。
* **读取时间**: 从RTC计数器读出当前的时间戳，加上8小时转换为本地时间。再调用localtime()函数将时间戳反向转换为struct tm结构体，最后将结构体中的年月日时分秒信息更新回MyRTC\_Time数组，供OLED显示。



3-2-3 RTC时间管理流程图

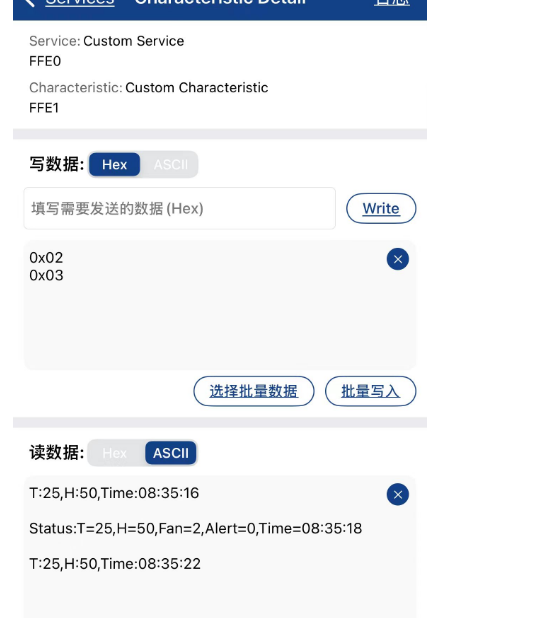
**3.3 调试过程**

1. **基础功能调试：** 编写简单的LED闪烁程序，验证程序下载和基本GPIO输出功能。
2. **模块独立调试：** 分别对OLED显示、串口通信、传感器数据读取、执行器（LED、蜂鸣器、风扇）控制、按键扫描和RTC功能进行单独测试，确保每个模块都能按预期工作。
3. **系统联调：** 将所有模块整合到主程序中。初期遇到的主要问题是自动报警逻辑和手动控制逻辑的冲突。通过引入BluetoothControl标志位，设定手动模式拥有更高优先级，解决了该问题。此外，通过对蓝牙指令接收增加延时和过滤机制，提升了远程控制的稳定性。

**4. 结果与结论**

**4.1 系统成果展示**

经过硬件组装和软件调试，系统成功实现了预定的所有功能。

* **硬件实物图：**
* 
* **蓝牙通信测试：** 使用手机蓝牙串口APP，可以成功连接并发送指令控制风扇转速、开关报警器，并能正常接收系统上报的环境数据。
* 

**4.2 结论**

本项目成功设计并实现了一个基于STM32的智能环境监测系统。系统以STM32F103为核心，集成了温湿度、烟雾等多种传感器，实现了环境参数的实时采集与显示。系统创新的结合了自动报警和远程手动控制两种模式，既能独立工作，又能通过蓝牙与用户交互，增强了实用性和灵活性。通过模块化的软硬件设计，系统结构清晰，便于维护和功能扩展。实践证明，该系统运行稳定，响应迅速，达到了预期的设计目标。

**4.3 心得体会与展望**

通过本次课程设计，我将课堂上学到的理论知识与实践紧密结合，对STM32单片机的开发流程有了全面而深入的理解。从需求分析、方案设计、元器件选型，到原理图绘制、代码编写、焊接调试，每一个环节都充满了挑战和收获。我深刻体会到，模块化编程和分步调试是解决复杂工程问题的有效方法。

**对系统的未来改进与展望：**

1. **网络化升级：** 将蓝牙模块替换为ESP8266等Wi-Fi模块，使设备接入互联网，通过MQTT等协议将数据上传到云服务器，实现真正意义上的远程无限距离监控。
2. **开发专属APP：** 设计一个图形化的手机APP，提供更友好的用户交互界面，并能展示历史数据曲线图。
3. **增加传感器种类：** 集成PM2.5、光照、二氧化碳等更多类型的传感器，实现更全面的环境监测。
4. **低功耗设计：** 优化系统功耗，增加深度睡眠和唤醒机制，使其能够由电池供电，应用于更广泛的场景。

**附录**

代码文件已上传至github仓库：<https://github.com/zhanivgit/2025-630-32>