

STM32F4智能农业监控系统

项目概述

STM32F4智能农业监控系统是一个基于STM32F407单片机的综合性环境监测与控制平台。系统集成了多种传感器模块，实时监测农业环境中的温湿度、光照、烟雾浓度及设备姿态等关键参数，并通过LCD显示和蓝牙无线通信实现数据可视化和远程控制。

本系统适用于农业大棚、智能花盆、家庭种植等多种应用场景，为现代农业物联网监控提供了一个完整的解决方案。

系统特点

- **多传感### 阈值配置与LED报警指示

系统提供了多种环境参数的阈值设置，可通过蓝牙命令动态调整。当参数超出阈值范围时，系统将触发对应的LED报警指示和蜂鸣器报警：

参数	默认值	有效范围	命令	说明	LED指示
温度上限	29°C	0-60°C	01 VALUE	超过此值触发高温报警	LED0常亮
温度下限	20°C	0-60°C	02 VALUE	低于此值触发低温报警	LED0常亮
湿度上限	70%	0-100%	03 VALUE	超过此值触发高湿报警	LED1常亮
湿度下限	40%	0-100%	04 VALUE	低于此值触发低湿报警	LED1常亮
光照下限	40%	0-100%	05 VALUE	低于此值触发弱光报警	LED2呼吸灯模式(PE13)
烟雾上限	120ppm	0-1000ppm	06 VALUE	超过此值触发烟雾报警	LED3呼吸灯模式(PE14)

- **实时监控:** 500ms周期采集数据，实时反馈环境变化
- **多页面显示:** LCD界面支持4种显示模式，方便查看不同参数
- **蓝牙远程控制:** 支持手机APP通过蓝牙查询系统状态、动态调整阈值
- **智能报警机制:** 基于可调阈值的多参数异常报警
- **低功耗设计:** 采用定时唤醒和按需采集策略，降低系统功耗
- **模块化结构:** 清晰的代码架构便于功能扩展和维护

硬件组成

核心控制器

- **MCU:** STM32F407VGT6，主频 168MHz，1MB Flash，192KB RAM
- **开发板:** STM32F4-Discovery或兼容开发板

传感器模块

1. DHT11温湿度传感器

- 温度测量范围: 0~50°C，精度 $\pm 2^{\circ}\text{C}$
- 湿度测量范围: 20~90%RH，精度 $\pm 5\%\text{RH}$

2. 光敏电阻模块

- 光照强度感知，输出模拟电压信号
- 通过ADC3通道5采集，转换为0-100%光照强度

3. MQ2气体传感器

- 可燃气体浓度检测，测量范围: 300~10000ppm
- 输出模拟电压信号，通过专用接口采集

4. MPU6050姿态传感器

- 3轴加速度计: $\pm 2\text{g}$ 范围，16位ADC
- 3轴陀螺仪: $\pm 250^{\circ}/\text{s}$ 范围，16位ADC
- 内置温度传感器
- I2C接口通信(SCL-PB8, SDA-PB9)

输入输出设备

1. 1602 LCD显示屏

- 2行×16列字符显示
- 4位并行数据接口

2. 蓝牙模块(HC-05/HC-06)

- 蓝牙2.0协议，UART通信
- 波特率9600bps
- 支持AT指令配置

3. LED指示灯

- LED0(运行指示灯): 闪烁表示系统正常运行，常亮表示温度超阈值报警
- LED1(湿度报警灯): 常亮表示湿度超阈值报警
- LED2(PE13/TIM1_CH3): 呼吸灯模式，表示光照超阈值报警
- LED3(PE14/TIM1_CH4): 呼吸灯模式，表示烟雾超阈值报警

4. 蜂鸣器

- 声音报警输出

5. 按键模块

- 4个功能按键，用于切换显示页面

软件架构

系统结构

系统软件采用模块化设计，主要分为以下几个部分：

1. 硬件驱动层

- 提供底层硬件抽象
- 包括GPIO、ADC、UART、I2C、中断等驱动

2. 传感器接口层

- 封装各传感器通信和数据转换逻辑
- 提供统一的初始化和数据获取接口

3. 数据处理层

- 数据采集、过滤、转换和存储
- 阈值判断和报警逻辑

4. 通信协议层

- 蓝牙通信协议实现
- 命令解析和响应处理

5. 用户界面层

- LCD页面管理和显示逻辑
- LED、蜂鸣器状态控制

报警机制

系统采用差异化的LED指示灯方案，针对不同的环境参数异常提供直观的视觉反馈：

1. 温度报警（LED0常亮）

- 当温度超过上限阈值或低于下限阈值时触发
- 通过LED0常亮状态直观指示温度异常

2. 湿度报警（LED1常亮）

- 当湿度超过上限阈值或低于下限阈值时触发
- 通过LED1常亮状态直观指示湿度异常

3. 光照报警（LED2呼吸灯）

- 当光照强度低于阈值时触发
- 使用PE13(TIM1_CH3)引脚实现呼吸灯效果，渐亮渐暗循环
- 呼吸灯效果通过PWM调制实现

4. 烟雾报警（LED3呼吸灯）

- 当烟雾浓度超过阈值时触发
- 使用PE14(TIM1_CH4)引脚实现呼吸灯效果，渐亮渐暗循环
- 呼吸灯效果通过PWM调制实现

所有报警状态同时会触发蜂鸣器发出声音报警，并在LCD对应页面显示报警信息。

主循环流程

1	初始化系统
2	初始化传感器
3	进入主循环：
4	处理蓝牙通信
5	采集传感器数据
6	检查报警条件
7	处理按键输入
8	更新LCD显示
9	系统延时10ms

功能模块说明

1. DHT11温湿度模块

功能特点：

- 数字信号输出，单总线接口
- 自动采集环境温湿度数据
- 支持报警阈值自定义

使用方法：

- 初始化: dht11_init()
- 读取数据: dht11_read_dat(&temperature, &humidity)
- 返回值0表示成功，非0表示失败

参考文档：

- [DHT11错误修正说明.md](#)
- [DHT11启用调试指南.md](#)

2. 光照传感器模块

功能特点:

- 基于光敏电阻的模拟量输入
- 通过ADC3通道5采集
- 支持低光照报警

使用方法:

- 初始化: `Light_Init()`
- 读取原始值: `Light_GetRawValue()`
- 转换为百分比: $\text{light_percent} = 100 - (\text{raw_value} * 100 / 4095)$

注意事项:

- 光敏电阻采集值与光照强度呈反比关系
- ADC参考电压3.3V，分辨率12位(0-4095)

3. MQ2气体传感器模块

功能特点:

- 对甲烷、丁烷、LPG、烟雾敏感
- 模拟量输出，支持ppm浓度转换
- 高浓度报警功能

使用方法:

- 初始化: `MQ2_Init()`
- 发送读取命令: `MQ2_SendCommand()`
- 检查数据就绪: `MQ2_IsDataReady()`
- 获取ppm值: `MQ2_GetValue()`

注意事项:

- 预热时间较长，建议上电后2分钟再使用
- 避免在高浓度气体环境下长时间使用

4. MPU6050姿态传感器模块

功能特点:

- 三轴加速度和角速度测量
- 姿态角(Roll/Pitch)计算
- 内置温度传感器
- I2C接口通信

使用方法:

- 初始化: MPU6050_Init()
- 读取数据: MPU6050_GetData(&mpu_data)
- 姿态角计算:

```
1 roll = atan2f(accel_y, accel_z) * 57.29578f;  
2 pitch = atan2f(-accel_x, sqrt(accel_y*accel_y +  
    accel_z*accel_z)) * 57.29578f;
```

参考文档:

- [MPU6050使用说明书.md](#)
- [MPU6050连接示意图.md](#)
- [MPU6050常见问题与解决方案.md](#)

5. 蓝牙通信模块

功能特点:

- 基于HC-05/HC-06模块
- 支持查询系统状态
- 支持动态调整报警阈值
- 指令格式简洁, 易于扩展

命令格式:

- 查询状态: 00
- 设置温度高阈值: 01 VALUE

- 设置温度低阈值: 02 VALUE
- 设置湿度高阈值: 03 VALUE
- 设置湿度低阈值: 04 VALUE
- 设置光照低阈值: 05 VALUE
- 设置烟雾高阈值: 06 VALUE

使用方法:

- 通过手机蓝牙串口工具连接模块
- 发送命令格式为: <命令> <参数>
- 例如: 01 30 设置温度高阈值为30°C

参考文档:

- [蓝牙命令指导书.md](#)
- [蓝牙阈值控制使用说明.md](#)

6. LCD显示模块

功能特点:

- 多页面显示模式
- 通过按键切换不同参数页面
- 实时更新传感器数据
- 显示报警状态

显示页面:

1. 温湿度页面(KEY0): 显示温度、湿度及相关报警
2. 光照/烟雾页面(KEY1): 显示光照百分比和烟雾浓度
3. 姿态页面(KEY2): 循环显示加速度、角速度和姿态角
4. 系统信息页面(KEY3): 显示运行时间、错误计数等调试信息

使用方法:

- 按键KEY0-KEY3切换对应页面
- 姿态页面每2秒自动切换显示不同数据
- 系统信息页面每3秒自动切换显示不同调试信息

编译与调试

编译环境

- Keil MDK 5.x
- ARM编译工具链: arm-none-eabi-gcc
- 支持的工程文件: stm32f4.code-workspace

调试开关

主程序中提供了多个模块的使能开关，便于单独调试各功能:

```
1  #define ENABLE_DHT11      1    // 1-启用DHT11, 0-禁用DHT11
2  #define ENABLE_MPU6050    1    // 1-启用MPU6050, 0-禁用
   MPU6050
3  #define ENABLE_MQ2        1    // 1-启用MQ2, 0-禁用MQ2
4  #define ENABLE_LIGHT      1    // 1-启用光敏电阻, 0-禁用光敏
   电阻
```

编译指令

使用VS Code任务(Tasks):

- Build STM32 Project: 编译项目
- Clean Build: 清理编译输出
- Check ARM Toolchain: 检查工具链配置

调试注意事项

- 编译前确保已安装正确的ARM工具链
- 如遇程序卡死，尝试关闭对应模块排查
- 参考[程序卡死问题排查方案.md](#)解决常见问题

阈值配置

系统提供了多种环境参数的阈值设置，可通过蓝牙命令动态调整:

参数	默认值	有效范围	命令	说明
温度上限	29℃	0-60℃	01 VALUE	超过此值触发高温报警
温度下限	20℃	0-60℃	02 VALUE	低于此值触发低温报警
湿度上限	70%	0-100%	03 VALUE	超过此值触发高湿报警
湿度下限	40%	0-100%	04 VALUE	低于此值触发低湿报警
光照下限	40%	0-100%	05 VALUE	低于此值触发弱光报警
烟雾上限	120ppm	0-1000ppm	06 VALUE	超过此值触发烟雾报警

故障排除

常见问题与解决方案

1. 系统无响应/卡死
 - 检查电源是否稳定
 - 尝试禁用部分模块(修改对应ENABLE_XXX宏定义)
 - 查看I2C通信是否超时
 - 参考[程序卡死问题排查方案.md](#)
2. 传感器数据异常
 - DHT11问题: 参考[DHT11异常数据问题解决方案.md](#)
 - MPU6050问题: 参考[MPU6050常见问题与解决方案.md](#)
 - 光敏电阻问题: 检查ADC通道配置，确认为通道5
 - MQ2问题: 确保预热充分，检查通信超时处理
3. 蓝牙连接失败
 - 确认波特率设置(9600bps)
 - 检查蓝牙模块指示灯状态
 - 重新上电初始化蓝牙模块
4. 报警功能失效
 - 检查阈值设置是否合理

- 验证蜂鸣器和LED连接是否正常
- 查看传感器数据是否正确
- 确认LED驱动是否正常（特别是PE13/PE14的呼吸灯功能）

5. LCD显示异常

- 检查LCD接线
- 验证LCD初始化是否成功
- 尝试在初始化后清屏并延时

调试技巧

1. 使用系统信息页面(KEY3):

- 显示运行时间和错误计数
- 查看传感器状态和原始值
- 监控通信状态

2. 添加调试延时:

- 在关键步骤添加 `Mdelay_Lib()` 延时
- 配合LCD显示调试信息

3. 启用传感器详细日志:

- 修改对应模块初始化函数，增加调试输出
- 使用串口或LCD显示关键参数

扩展开发

添加新传感器

1. 创建传感器驱动文件(`xxx.c/xxx.h`)
2. 在HARDWARE目录下创建对应文件夹
3. 添加初始化和数据读取函数
4. 在main.c中包含头文件并调用接口
5. 更新数据结构和显示逻辑

修改通信协议

1. 在bluetooth.h中定义新的命令常量
2. 在ProcessReceivedCommand()函数中添加处理逻辑
3. 更新SendSystemStatus()函数增加新数据显示
4. 测试新命令响应

添加新的显示页面

1. 在PageType_t枚举中添加新页面类型
2. 在Display_Update()函数中添加新的case分支
3. 实现页面显示逻辑
4. 添加页面切换按键处理

修改报警逻辑

1. 自定义LED报警指示:

```
1 // 温度报警控制 (LED0)
2 if (sensor_data.temperature > TEMP_HIGH_THRESHOLD
    || sensor_data.temperature < TEMP_LOW_THRESHOLD)
3 {
4     Led_On(LED0); // LED0常亮
5 } else {
6     Led_Off(LED0);
7 }
8 // 湿度报警控制 (LED1)
9 if (sensor_data.humidity > HUMI_HIGH_THRESHOLD ||
    sensor_data.humidity < HUMI_LOW_THRESHOLD) {
10     Led_On(LED1); // LED1常亮
11 } else {
12     Led_Off(LED1);
13 }
14
15 // 光照报警控制 (LED2呼吸灯 - PE13/TIM1_CH3)
```

```
16  if (sensor_data.light_percent <
    LIGHT_LOW_THRESHOLD) {
17      LED_Breath_Enable(LED2); // 启用LED2呼吸灯效
    果
18  } else {
19      LED_Breath_Disable(LED2);
20  }
21
22  // 烟雾报警控制 (LED3呼吸灯 - PE14/TIM1_CH4)
23  if (sensor_data.smoke_ppm_value >
    SMOKE_HIGH_THRESHOLD) {
24      LED_Breath_Enable(LED3); // 启用LED3呼吸灯效
    果
25  } else {
26      LED_Breath_Disable(LED3);
27  }
```

2. 调整呼吸灯效果:

- 修改TIM1的PWM配置，调整呼吸灯的周期和亮度曲线
- 在led.c中修改呼吸灯算法，实现不同的视觉效果

3. 组合报警逻辑:

- 可以根据多个传感器状态组合判断报警条件
- 实现不同优先级的报警指示策略

许可证

本项目遵循MIT许可证。详细内容请参阅LICENSE文件。

联系方式

如有任何问题或建议，请联系项目维护者：

- 邮箱: your_email@example.com
- GitHub: [your-github-username](https://github.com/your-github-username)

致谢

- 感谢所有参与本项目开发的团队成员
- 感谢STM32F4官方示例代码和社区贡献
- 感谢各传感器模块提供商的技术支持