



# 廣東工業大學

## 嵌入式系統課程設計報告

《基於 stm32 的藍牙溫控風扇》

|         |   |            |
|---------|---|------------|
| 學       | 院 | 集成電路學院     |
| 專       | 業 | 微電子科學與工程   |
| 年 級 班 別 |   | 23 級 2 班   |
| 學       | 號 | 3123009633 |
| 學 生 姓 名 |   | 袁承浩        |
| 隊       | 名 | 好不好        |
| 隊       | 友 | 黃錦浩        |
| 指 導 老 師 |   | 周賢中        |

集成電路學院

2025 年 7 月 3 日

## 摘要

本文介绍了嵌入式系统设计项目——基于 stm32 单片机的蓝牙温控风扇系统设计，利用 C 语言进行嵌入式软件开发，设计并实现了包括电机驱动控制，温度感知，蓝牙控制，多档位风速调节（pwm 控制），温度控制自启动等核心功能的系统。在 keil MDK 开发环境下完成了从程序设计，使用 stm32F103C8T6 微控制器驱动，并在搭建的电风扇硬件平台上完成了功能验证和性能测试，项目展示了 STM32 单片机在嵌入式系统设计中定制化设计和高效率的优点和综合应用价值

关键词：STM32；嵌入式系统；pwm 电机驱动；温度感知；电风扇控制，蓝牙控制

# 目录

|                  |    |
|------------------|----|
| 1 项目简介           | 4  |
| 2 项目设计与实现        | 5  |
| 2.1 硬件设计         | 5  |
| 2.1.1 系统总体架构     | 5  |
| 2.1.2 主控制器模块     | 5  |
| 2.1.3 温度感知模块     | 5  |
| 2.1.4 电机驱动模块     | 5  |
| 2.1.5 用户交互模块     | 5  |
| 2.2 软件设计         | 6  |
| 2.2.1 系统软件架构     | 6  |
| 2.2.2 主程序流程      | 6  |
| 2.2.3 蓝牙模块解码协议实现 | 6  |
| 2.3 代码介绍         | 7  |
| 2.3.1 初始化        | 7  |
| 2.3.2 主功能实现      | 8  |
| 2.3.3 循环体结构      | 8  |
| 2.4 技术实现细节       | 9  |
| 2.4.1 PWM 调速实现   | 9  |
| 2.4.2 上位机实时接收数据  | 9  |
| 3 功能展示           | 10 |
| 3.1 功能描述         | 10 |
| 3.1.1 基础控制功能     | 10 |
| 3.1.2 智能控制功能     | 10 |
| 4 物料与成本          | 11 |
| 4.1 物料清单         | 11 |
| 4.2 项目成本         | 12 |
| 4.3 主要物料展示       | 13 |
| 5 个人贡献           | 15 |
| 6 开发问题描述与解决方案    | 16 |
| 7 项目成果与改进方向      | 17 |

## 1 项目简介

随着物联网技术和智能家居市场的快速发展，传统电风扇已无法满足现代用户对智能化、便捷化和节能环保的需求。本项目基于 STM32 微控制器设计了一款集蓝牙远程控制、温度感知自动调节和 PWM 精准调速于一体的智能风扇系统，特别适用于家庭卧室、客厅等生活场景，以及办公室、会议室等工作环境，能够为用户提供更加舒适、节能和智能化的使用体验。

本设计的主要功能包括：

- 1 PWM 调速：利用 STM32 的定时器产生高精度 PWM 信号，实现对直流电机的无级调速控制，调速精度可达 1%
- 2 蓝牙远程操控：通过集成蓝牙 5.0 模块，支持手机 APP 远程控制，实现 10 米范围内的稳定连接，用户可以随时调节风速和工作模式
- 3 温度控制自启动：采用数字温度传感器实现环境温度检测，当温度超过设定阈值时自动启动风扇，并根据温度变化智能调节转速
- 4 OLED 实时显示：增加 OLED 显示屏实时显示当前温度、风速和工作状态，配合蜂鸣器提供操作反馈

这些功能不仅涵盖了嵌入式系统开发的核心技术（包括嵌入式软件开发，硬件接口驱动，串口通信，上位机开发，控制算法，硬件电路设计及资源管理），也充分考虑了实际应用场景中的用户体验需求，是对嵌入式技术在家电控制领域的一次创新应用。

## 2 项目设计与实现

### 2.1 硬件设计

#### 2.1.1 系统总体架构

系统采用了模块化设计，以 STM32F103C8T6 微控制器为核心，通过外围电路实现蓝牙通信和控制，电机驱动，温度感知等关键功能，总系统架构包括主控制器模块，温度感知模块，电机驱动模块，用户交互模块。

#### 2.1.2 主控制器模块

选用 STM32F103C8T6 微控制器作为主控芯片，主要特性：

- 72MHz Cortex-M3 内核
- 64KB Flash, 20KB SRAM
- 3 个通用定时器 (TIM2/TIM3/TIM4)
- 丰富的外设接口 (USART、SPI、I2C 等)

#### 2.1.3 温度感知模块

采用 DS18B20 温度传感器，主要特性为：

- 测量范围：-55°C ~ +125°C
- 精度：±0.5°C
- 采样周期：500ms

#### 2.1.4 电机驱动模块

采用 TB6612FNG 驱动芯片，驱动 12V 直流风扇电机

- 关键参数：
- 最大输出电流：1.2A (连续)
- PWM 频率：20kHz
- 控制接口：AIN1/AIN2 (方向控制)，PWMA (速度控制)

#### 2.1.5 用户交互模块

使用 EC11 旋转编码器，兼具调速和转向功能，使用 OLED 搭配蜂鸣器和 LED 作为指示灯，优化了用户操作体验

- EC11 旋转编码器：用于手动调速
- 0.96 寸 OLED：显示系统状态参数，同时还能辅助进行测试
- 蜂鸣器+LED：提供声光报警提示

## 2.2 软件设计

### 2.2.1 系统软件架构

软件采用了分层设计。1 硬件抽象层 (HAL)：STM32CubeMX 生成的初始化代码，外设驱动封装。功能模块层：PWM 调速模块，温度采集模块，蓝牙通信模块，编码器处理模块，OLED 显示模块。应用逻辑层：主控制状态机，用户交互逻辑

### 2.2.2 主程序流程

系统采用了全局变量加状态机设计

- 1 全局变量声明，将各种关键全局变量声明，并进行注释
- 2 系统初始化 包括时钟 GPIO 口 定时器，串口
- 3 进入主循环，接收信号，处理全局变量，检索全局变量并对关键值提取赋值
- 4 根据提取值进行处理，更新状态指示灯

### 2.2.3 蓝牙模块解码协议实现

使用上位机进行了按键对应功能编写 (如图)，通过发送对应值，与全局变量进行关联比对，从而控制主程序执行对应操作，具体按键功能如下

Screen1

智能风扇蓝牙控制app

蓝牙连接 连接状态: 断开连接 断开连接

接收数据: 接收到的数据

实时温度: 温度显示

实时转速: 转速显示

自动模式

手动模式

开机

关机

一档

二档

三档

减速

正反反转切换

加速

当前工作状态 未连接

```
while (1)
{
    if (Serial_GetRxFlag() == 1) //检查串口接收数据的标志位
    {
        RxData = Serial_GetRxData(); //获取串口接收的数据
        if (RxData==0x00) {mo=1;Speed=0;Motor_SetSpeed(Speed);} //切换模式
        if (RxData==0x10) {mo=0;}
        // Serial_SendByte(RxData); //串口将收到的数据回传回去，用于测试
        //OLED_ShowHexNum(1, 8, RxData, 2); //显示串口接收的数据
    }
}
```

```
if (RxData==0x01) Speed=50; //开机
if (RxData==0x02) Speed=0; //关机
if (RxData==0x03) Speed=30; //一档
if (RxData==0x04) Speed=60; //二档
if (RxData==0x05) Speed=90; //三档
if (RxData==0x06) Speed+=5; //加速
if (RxData==0x07) Speed-=5; //减速
if (RxData==0x08) Speed=-Speed; //反转
```



| 按键名称 | 按键发送内容 | 系统响应        |
|------|--------|-------------|
| 自动模式 | 0x00   | 进入自动模式，温度控制 |
| 开机   | 0x01   | 开机，速度设为五十   |
| 关机   | 0x02   | 关机，速度置零     |
| 一档   | 0x03   | 速度设为三十      |
| 二档   | 0x04   | 速度设为六十      |
| 三档   | 0x05   | 速度设为九十      |
| 加速   | 0x06   | 速度加五        |
| 减速   | 0x07   | 速度减五        |
| 正反转  | 0x08   | 速度变为相反数（反转） |
| 手动模式 | 0x01   | 进入手动模式      |

## 2.3 代码介绍

### 2.3.1 初始化

```

OLED_Init();      //OLED 初始化
Motor_Init();     //直流电机初始化
Key_Init();       //按键初始化（没用到）
Encoder_Init();   //旋转编码器初始化
Timer_Init();     //定时器初始化（用于蜂鸣器）
AD_Init();        //AD 初始化（用于温度传感器）
Serial_Init();    //串口初始化（蓝牙通信）

```

### 2.3.2 主功能实现

Motor\_SetSpeed(int8\_t Speed)  
PWM\_SetCompare3(Speed)  
控制速度函数  
PWM\_SetCompare3(uint16\_t Compare)  
PWM 占空比设置  
GPIO\_ResetBits(GPIOA, GPIO\_Pin\_8);  
GPIO\_SetBits(GPIOA, GPIO\_Pin\_11)  
LED 状态灯更新

### 2.3.3 循环体结构

```
while (1)
{
    if (Serial_GetRxFlag() == 1)          //检查串口接收数据的标志位
    {
        RxData = Serial_GetRxData();      //获取串口接收的数据
        if(RxData==0x00){mo=1;Speed=0;Motor_SetSpeed(Speed);}//切换模式
        if(RxData==0x10){mo=0;}
        // Serial_SendByte(RxData);        //串口将收到的数据回传回去，用于
        测试                               //显示串口接收的数据
        OLED_ShowHexNum(1, 8, RxData, 2);

    }
    ADValue = AD_GetValue();              //获取 AD 转换的值
    Voltage = 100-((ADValue-1730)/2.1);    //将 AD 值线性变换到 0~100 的范围
    KeyNum = Key_GetNum();                 //获取按键键码
    if(mo==0)
    {
        if(Voltage>25){
            if(Voltage>65)                 //温度变量达到 65 以上
            {
                flag=1;                   //蜂鸣器标志位
                GPIO_ResetBits(GPIOA, GPIO_Pin_8); //LED 红灯亮
                GPIO_SetBits(GPIOA, GPIO_Pin_11);
            }
            if(Voltage>55&&Voltage<65){GPIO_SetBits(GPIOA,
GPIO_Pin_8);GPIO_SetBits(GPIOA, GPIO_Pin_11);}
            if(Voltage<55)
            {
                flag=0;
                GPIO_SetBits(GPIOA, GPIO_Pin_8); //LED 红灯不亮
                GPIO_ResetBits(GPIOA, GPIO_Pin_11);
            }
        }
    }
}
```



```

        Speed=Voltage;
        if(Voltage>=100){Speed=99;}           //边界判断
    }
    if(Voltage<=25&&Voltage>=20){Speed=0;}
    if(Voltage<20){
        Speed+=4*Encoder_Get();
    }
    if(Speed>=100||Speed<=-100){Speed=0;}
    Motor_SetSpeed(Speed);                   //设置直流电机的速度为速度变量
    OLED_ShowSignedNum(2, 7, Speed, 3);     //OLED 显示速度变量
    OLED_ShowNum(3, 9, Voltage, 3);         //显示电压值的整数部分
    }
else if(mo==1)
{
    if(RxData==0x01)Speed=50;//开机
    if(RxData==0x02)Speed=0;//关机
    if(RxData==0x03)Speed=30;//一档
    if(RxData==0x04)Speed=60;//二档
    if(RxData==0x05)Speed=90;//三档
    if(RxData==0x06)Speed+=5;//加速
    if(RxData==0x07)Speed-=5;//减速
    if(RxData==0x08)Speed=-Speed;//反转
    if(Speed>=100||Speed<=-100){Speed=0;}
    Motor_SetSpeed(Speed);

    Serial_SendNumber(Voltage*1000+Speed,5);
}
}
}

```

## 2.4 技术实现细节

### 2.4.1 PWM 调速实现

使用 TIM3 产生频率为 20kHz 的 PWM 信号，通过改变 CCR 寄存器的值调整占空比。CCR 和 ARR 共同决定占空比，占空比  $Duty = CCR / (ARR + 1)$

### 2.4.2 上位机实时接收数据

上位机使用定时器，每隔 500ms 接受一次从蓝牙模块发出的信息，蓝牙模块每隔一段时间发送当前显示的温度和当前转速组成的五位数字（温度为两位数字，转速为 000~100 的三位数字）并分别在位置 1，位置 3 处提取长度为 2 和 3 的数据从而实现上位机的同步数据显示。

## 3 功能展示

### 3.1 功能描述

本系统实现了基于蓝牙模块和温度模块的智能风扇控制功能, 包括基础控制和智能控制部分

#### 3.1.1 基础控制功能

风扇转速控制: 通过旋钮编码器控制风扇的转速, 转速为零时关闭, 转速在-100~100 之间, 可以按照需求自行调节转速

状态指示: 通过两个 LED, 一个 OLED 显示屏和一个蜂鸣器进行状态指示, 开机状态黄灯默认亮起, 当温度较高 (大于六十五度) 红灯亮起同时蜂鸣器报警。OLED 屏幕显示实时转速和温度便于调节速度。

#### 3.1.2 智能控制功能

蓝牙远程控制: 通过上文提到的按键实现了多种功能, 包括开关机, 调速, 模式切换, 正反转以及接收温度转速数据

温度自动模式: 当温度高于一定值时风扇自启动, 并根据温度变化情况调节转速, 温度高转速高, 温度降低转速减慢。

## 4 物料与成本

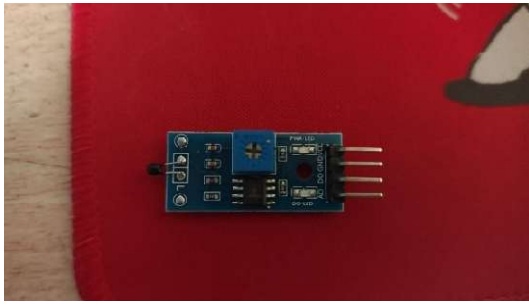
### 4.1 物料清单

| 物料名称                 | 数量 | 单价(元)  | 用途     |
|----------------------|----|--------|--------|
| STM32F103C8T6<br>核心板 | 1  | 12.80  | 主控制器   |
| TB6612FNG电机<br>驱动模块  | 1  | 8.50   | 风扇电机驱动 |
| DC 5V-12V直流风<br>扇    | 1  | 16.00  | 风力输出   |
| HC-05蓝牙模块            | 1  | 18.00  | 无线通信控制 |
| EC11旋转编码器            | 1  | 3.20   | 手动调速输入 |
| □温度传感器模块             | 1  | 5.30   | 环境温度采集 |
| 0.96寸OLED显示<br>屏     | 1  | 15.50  | 状态参数显示 |
| 5V蜂鸣器                | 1  | 0.80   | 高温报警   |
| LED灯珠（红/黄）<br>◀ —▶   | 2  | 0.25   | 状态指示灯  |
| 面包板                  | 1  | 6.00   | 电路搭建   |
| 杜邦线(20cm)            | 30 | 0.12   | 电路连接   |
| 电阻(220Ω)             | 5  | 0.05   | 限流保护   |
| 电容(0.1μF)            | 5  | 0.10   | 电源滤波   |
| 小计                   | -  | 102.22 | -      |

4.2 项目成本

| 成本项目   | 金额(元)  | 备注          |
|--------|--------|-------------|
| 基础材料成本 | 102.22 | 上表物料小计      |
| 工具设备折旧 | 18.00  | 焊枪、热风枪等工具使用 |
| 软件开发环境 | 0.00   | Keil MDK免费版 |
| 总成本    | 120.22 | -           |

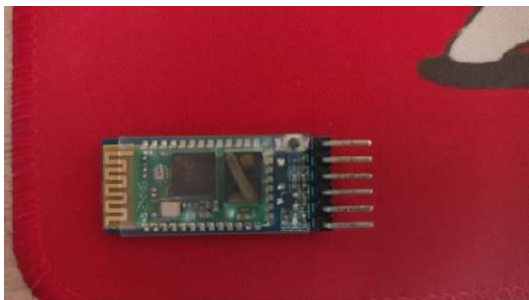
### 4.3 主要物料展示



温度传感器模块



ST-LINK



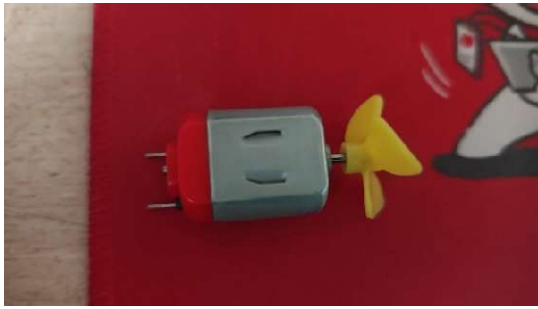
HC-05 蓝牙模块



5V 蜂鸣器



TB6612FNG 电机驱动模块



DC 5V-12V 直流风扇电机



EC11 旋转编码器



STM32F103C8T6 最小系统板

# 5 个人贡献

蓝牙串口通信功能实现  
上位机开发与逻辑编写  
主函数设计，设计全局变量和状态机，整合代码实现蓝牙远程控制功能  
最终测试



非可视组件

蓝牙客户端! 计时器! 对话框!

pwm风扇蓝牙控制api

- 蓝牙连接部分
  - 蓝牙连接
    - 连接状态
    - 显示连接状态
    - 断开连接
  - 接收数据部分
    - 接收数据
    - 接收到的数据
  - 显示部分
    - 温度显示

重命名 删除

素材

上传文件...

Appearance

应用说明

水平对齐

居左:1

垂直对齐

居上:1

背景颜色

默认

背景图片

无...

默认大字体

关屏动画

默认效果

高对比度

开屏动画

默认效果

屏幕方向

不设方向

允许滚动

显示状态栏

资源投入，最小系统板，焊台及辅助工具，杜邦线，OLED 屏幕

## 6 开发问题描述与解决方案

1、温度模块精度低，无法正确显示当前温度

解决方案：

软件方面：由原来的显示当前温度改为显示温度变化情况，当温度升温超过一定值触发自动开机并由温度变化值控制转速

硬件方面：更换新的精度更高的温度测量模块，尝试实现原有目标

2、显示值变化速度过快

解决方案：

由原有的显示当前值变为显示前十次测量值的平均值，由于测量频率较高，最后可以判断测量值有效

3、蓝牙接收数据有大量无效值

解决方案：

经过检查发现由于发送频率过高以及温度测量的精度问题出现大量无效值，优化了发送值，对其进行判断，并减少了发送频率



## 7 项目成果与改进方向

本项目基本实现了设计目标，各功能均能正常使用且效果较好，但仍存在精度不够，实用性不够的情况，未来可以通过更换硬件设施，以及通过 3D 打印技术制作简易外壳以及打板等方式让结构更完整，提高实用性和美观性。

