

预备队电控学习路线

准备工作

软件安装

软件	说明
STM32CubeMX	ST官方芯片图形化配置工具，用于初始化代码生成
Keil MDK	经典ARM开发IDE，提供编译与调试环境
VS Code	轻量级代码编辑器，配合 Keil Assistant 插件提升编码体验
Git	版本控制工具，用于代码管理和团队协作
VOFA+	强大的串口调试与数据可视化上位机

硬件准备

- **STM32F103C8T6**最小系统板
- 面包板、杜邦线
- 调试器 (**ST-Link / DAP-Link**)
- **USB转TTL**模块 (CH340)
- LED、按键、舵机、OLED屏、**MPU6050/9250**、**ESP8266**

学习内容

知识点

- 外设
 - **GPIO**
 - **EXTI**
 - **TIM**
 - **USART**
 - **DMA**
 - **IIC**
 - **SPI**
 - **CAN**
- 中间件
 - *FreeRTOS*
- 算法
 - 自动控制算法 (**PID**等)

- 滤波算法 (均值、中值、滑动、互补、卡尔曼等)
- 软件技能
 - Keil调试
 - VOFA+ 上位机
 - Git版本控制, GitHub代码托管平台
 - Markdown语法
 - 代码结构与注释
- 工具
 - 调试器 (ST-Link、DAP-Link等)
 - USB转TTL

任务一：基础外设掌控（硬件交互）

核心目标：熟悉最基本的输入输出、定时器和中断

任务	核心知识点	详细要求	成果验证
1.1 点亮LED	GPIO输出	配置GPIO引脚，编写程序使LED闪烁	上电后，LED以1Hz频率稳定闪烁
1.2 按键控制	EXTI	配置按键引脚为外部中断模式，实现按键控制LED亮灭	每次按下按键，LED状态改变一次
1.3 呼吸灯	TIM(PWM)	使用定时器PWM模式，动态改变占空比，实现LED呼吸灯效果	LED亮度平滑地由暗到亮，再由亮到暗循环
1.4 舵机控制	TIM(PWM)	学习舵机控制原理，输出特定占空比的PWM波控制舵机角度	能通过程序将舵机精确驱动到0°，90°，180°
1.5 高级按键识别	状态机, 非阻塞编程	通过软件状态机，识别按键的单击、双击、长按等操作，并控制舵机转到不同位置	单击舵机转至A点，双击转至B点，长按转至C点

任务二：串口通信与优化（人机交互）

核心目标：掌握串口通信，并学会使用DMA进行高效数据搬运

任务	核心知识点	详细要求	成果验证
2.1 串口通信	USART	配置USART，实现与PC上位机(VOFA+)的双向通信	在VOFA+中可看到单片机发送的 "Hello world"
2.2 定时发送	TIM	使用定时器中断，每隔固定时间(如100ms)向上位机发送一组数据（如系统运行时间）	VOFA+能接收到稳定的周期性数据流
2.3 指令控制	USART接收中断	在上位机发送特定指令（如 "LED_ON", "LED_OFF"），单片机解析并控制LED	发送 "ON" 点亮LED，发送 "OFF" 熄灭LED

任务	核心知识点	详细要求	成果验证
2.4 空闲中断	USART空闲中断	配置 串口空闲中断 ，用于接收不定长的数据帧	能正确接收并处理来自上位机的任意长度指令
2.5 DMA优化	DMA	使用DMA管理串口的发送和接收， 解放CPU	实现与2.4相同的功能，但 CPU占用率 显著降低
2.6 (拓展) Wi-Fi通信	AT指令	使用USART连接ESP8266模块，通过 AT指令 使其连接Wi-Fi，并实现 通过网络控制LED	在手机或电脑上，能通过网络工具控制开发板的 LED

任务三：引入实时操作系统（系统整合）

核心目标：理解**多任务思想**，使用**FreeRTOS**管理复杂应用

任务	核心知识点	详细要求	成果验证
3.1 多任务整合	FreeRTOS	创建两个独立任务： <code>LED_Task</code> （处理 呼吸灯 和 按键 ）和 <code>UART_Task</code> （处理 通信 ），将 任务一 和 任务二 的逻辑整合到 FreeRTOS 中	系统能 同时运行 呼吸灯和响应上位机命令， 互不干扰

注意：从本任务开始，后续开发默认基于**FreeRTOS**

任务四：传感器与数据融合（综合应用）

核心目标：掌握**I2C**协议，读取传感器数据，并进行初步的**数据处理**和**姿态解算**

任务	核心知识点	详细要求	成果验证
4.1 传感器驱动	I2C	通过 I2C 总线读取 MPU6050/9250 的 原始数据 （ 加速度 、 角速度 ）	能通过串口打印出正确的传感器 原始数据
4.2 数据滤波	滤波算法	对原始数据进行 软件滤波 （如 卡尔曼滤波 、 互补滤波 ），得到更稳定的数据	滤波后的数据曲线在 VOFA+ 上显示 明显平滑
4.3 屏幕驱动	驱动编写	编写 OLED 或 LCD 屏幕的驱动，将 传感器数据 或 解算出的姿态角 显示在屏幕上	屏幕能 实时刷新 显示当前的 Pitch、Roll、Yaw 角
4.4 姿态解算	姿态解算算法	使用 四元数 或 欧拉角 法，将 加速度计 和 陀螺仪 的数据 融合 ，解算出载体的 姿态角	在 VOFA+ 上能看到稳定的 姿态角 变化曲线，与物理转动吻合

任务五：自动控制算法与CAN通信

核心目标：掌握**PID控制算法**的原理与实现，学习**CAN总线通信**技术

任务	核心知识点	详细要求	成果验证
5.1 PID基础理论	PID控制原理	学习 PID控制器 的数学原理，理解 比例(P) 、 积分(I) 、 微分(D) 各项的作用及 参数整定 方法	能够阐述 PID三要素 对系统响应的影响，并举例说明
5.2 CAN通信理论	CAN协议原理	学习 CAN总线 的基本原理、 帧格式 （数据帧、远程帧）、 仲裁机制 、 错误检测 等核心概念	能够解释 CAN总线 如何实现多主机通信和优先级仲裁

加分项

- 使用Markdown记录学习历程

例如：对知识点的总结、遇到的问题以及解决方法、学习过程中的新发现、做题的思路

- 使用git对不同版本的代码进行管理，并规范 `commit message`