

# 预备队电控学习路线

## 🛠️ 准备工作

### 软件安装

软件	说明
<a href="#">STM32CubeMX</a>	ST官方芯片图形化配置工具，用于初始化代码生成
<a href="#">Keil MDK</a>	经典ARM开发IDE，提供编译与调试环境
<a href="#">VS Code</a>	轻量级代码编辑器，配合Keil Assistant插件提升编码体验
<a href="#">Git</a>	版本控制工具，用于代码管理和团队协作
<a href="#">VOFA+</a>	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 <b>外部中断模式</b> ，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 <b>PWM</b> 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 <b>PWM</b> 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 <b>软件状态机</b> ，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 <b>定时器中断</b> ，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 <b>比例(P)</b> 、 <b>积分(I)</b> 、 <b>微分(D)</b> 各项的作用及 <b>参数整定方法</b>	能够阐述 <b>PID三要素</b> 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、 <b>帧格式</b> （数据帧、远程帧）、 <b>仲裁机制</b> 、 <b>错误检测</b> 等核心概念	能够解释 <b>CAN总线</b> 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`