

嵌入式工作室24级暑期招新

这个题目旨在引导同学们从基础入门，逐步深入，体验一个小型物联网项目的完整流程。

- 1. **基础硬件控制** (ESP32 + Arduino)
- 2. **网络连接与Web服务** (ESP32作为服务器)
- 3. **数据采集与显示** (传感器 + OLED)
- 4. **云平台通信** (MQTT)
- 5. **RTOS初步体验** (FreeRTOS)
- 6. **(可选) 进阶探索**

项目名称：我的智能桌面伴侣

项目背景： 每个人都希望自己的学习和工作环境更加智能和舒适。本项目旨在通过ESP32微控制器，打造一个能够感知环境、提供信息、并能远程交互的“智能桌面伴侣”。

学习路线与难度阶梯：

阶段一：点亮我的世界 (难度：★☆☆☆☆ - 熟悉ESP32与Arduino)

- **目标：**
 - 1. 成功搭建ESP32的Arduino开发环境。
 - 2. 掌握GPIO的基本操作，控制LED。
 - 3. 学习如何连接Wi-Fi。
 - **硬件准备：**
 - 主控板：ESP32开发板 (推荐ESP32-WROOM-32, ESP32-C3, 或ESP32-S3均可，根据工作室已有资源或推荐购买)
 - 外设：1x LED灯，1x 220Ω电阻，面包板和杜邦线
 - **任务要求：**
 - 1. 让ESP32板载LED（或外接LED）实现1秒间隔的闪烁。
 - 2. 编写程序，使ESP32连接到你指定的Wi-Fi网络，并在串口打印出获取到的IP地址。
 - 3. **提交：**
 - Arduino代码 (.ino 文件)。
 - 简短的演示视频或GIF，展示LED闪烁和串口成功连接Wi-Fi并打印IP。
 - 学习笔记 (Markdown格式)：记录环境搭建过程、遇到的问题及解决方法。
-

阶段二：我的桌面Web管家 (难度：★★☆☆☆ - Web服务与远程控制)

- **目标：**
 - 1. 在ESP32上搭建一个简单的Web服务器。
 - 2. 通过网页控制LED的亮灭。
 - 3. 了解HTTP GET/POST请求。
- **硬件准备：**
 - 同阶段一
- **任务要求：**
 - 1. 在ESP32上创建一个Web服务器。

2. 设计一个简单的HTML页面，页面上有两个按钮：“开灯”和“关灯”。
 3. 当在浏览器中点击按钮时，通过HTTP请求（GET或POST均可）发送指令给ESP32，控制外接LED的亮灭。
 4. (可选) 在网页上显示当前LED的状态。
 5. **提交：**
 - Arduino代码。
 - 演示视频或GIF，展示通过浏览器网页控制LED。
 - 学习笔记：记录Web服务器搭建原理、HTTP请求的学习心得。
-

阶段三：感知我的环境 (难度：★★★☆☆ - 传感器与数据显示)

- **目标：**
 1. 学习使用I2C/SPI通信读取传感器数据。
 2. 将传感器数据显示在OLED屏幕上。
 3. 将传感器数据通过网页展示。
 - **硬件准备：**
 - 同阶段二
 - 传感器：DHT11或DHT22温湿度传感器 (简单，单总线) 或 BME280温湿度气压传感器 (I2C/SPI，更精确)
 - 显示：SSD1306 OLED显示屏 (I2C接口)
 - **任务要求：**
 1. 连接温湿度传感器到ESP32，并成功读取环境温度和湿度数据。
 2. 连接OLED显示屏到ESP32，将读取到的温湿度数据实时显示在OLED屏幕上。
 3. 修改阶段二的Web服务器，增加一个页面或在原页面上，实时显示当前的温湿度数据（可以通过AJAX实现动态刷新，或简单地在访问时刷新数据）。
 - 4. **提交：**
 - Arduino代码。
 - 演示视频或GIF，展示OLED显示数据，以及网页显示数据。
 - 学习笔记：记录传感器和OLED的接口原理、库的使用、数据处理方法。
-

阶段四：连接云端，远程掌控 (难度：★★★★☆ - MQTT与云平台)

- **目标：**
 1. 学习MQTT协议的基本概念。
 2. 将ESP32设备连接到一个公共MQTT Broker或云平台（如EMQX Cloud, ThingsBoard Community Edition, 阿里云物联网平台等）。
 3. 通过MQTT将传感器数据上报到云端。
 4. 通过MQTT从云端下发指令控制LED。
- **硬件准备：**
 - 同阶段三
- **任务要求：**
 1. 选择一个MQTT Broker或物联网平台，注册并创建一个设备。
 2. 编写程序，使ESP32：
 - 定期（如每5秒）将温湿度数据以JSON格式发布(Publish)到指定的MQTT主题(Topic)。
 - 订阅(Subscribe)一个控制主题，当该主题接收到特定指令（如`{"led": "on"}`或`{"led": "off"}`）时，控制LED的亮灭。

3. 使用MQTT客户端工具（如MQTTX, MQTT Explorer）或平台自带的控制台，验证数据上报和指令下发功能。
 4. **提交：**
 - Arduino代码。
 - 演示视频或GIF，展示数据成功上报到云平台，并通过云平台下发指令控制LED。
 - 学习笔记：记录MQTT协议学习、所选平台的使用方法、JSON数据格式设计。
-

阶段五：多任务处理初探 (难度：★★★★☆/★★★★★ - FreeRTOS基础)

- **目标：**
 1. 了解RTOS（实时操作系统）的基本概念和优势。
 2. 学习在ESP32上使用FreeRTOS创建和管理任务（可以在Arduino框架下使用，或者鼓励有能力的同学尝试ESP-IDF）。
 3. 将前述功能（传感器读取、数据显示、MQTT通信、LED控制）拆分成独立的FreeRTOS任务。
 - **硬件准备：**
 - 同阶段四
 - **任务要求：**
 1. **理论学习：** 理解什么是任务、任务调度、任务间通信（如队列、信号量）的基本概念。
 2. **实践：**
 - 创建一个任务专门负责读取传感器数据。
 - 创建一个任务专门负责更新OLED显示。
 - 创建一个任务专门负责处理MQTT的连接、数据上报和指令接收/处理。
 - 使用任务间通信机制（如队列）在任务间传递数据（例如，传感器任务将数据放入队列，OLED任务和MQTT任务从队列中读取）。
 3. 确保所有功能在多任务环境下依然正常工作。
 4. **提交：**
 - 代码 (Arduino with FreeRTOS API 或 ESP-IDF项目)。
 - 演示视频或GIF，展示多任务系统正常运行。
 - 学习笔记：记录FreeRTOS学习心得、任务划分思路、任务间通信的实现方法。
-

(可选) 进阶探索：空中升级 (OTA) 或自定义功能

- **目标：** 鼓励有余力的同学进行更深入的探索。
 - **方向建议：**
 1. **OTA (Over-The-Air) 更新：** 实现通过网络远程更新ESP32的固件。
 2. **低功耗模式：** 研究ESP32的Deep Sleep模式，尝试让你的桌面伴侣在空闲时进入低功耗状态。
 3. **更多传感器/执行器：** 加入光敏传感器根据环境光自动调节OLED亮度，或加入蜂鸣器进行事件提醒。
 4. **本地存储：** 使用SPIFFS或SD卡存储配置信息或历史数据。
 5. **ESP-NOW：** 如果有两块ESP32，尝试使用ESP-NOW进行设备间通信。
 - **提交：**
 - 代码。
 - 演示视频或GIF。
 - 详细的实现说明和学习笔记。
-

通用要求与提交：

1. **学习笔记：** 每个阶段都需要提交Markdown格式的学习笔记，记录学习过程、遇到的问题、解决方案、参考资料、以及对相关技术的理解。这是评估学习能力和解决问题能力的重要依据。
2. **代码管理：** 强烈建议使用Git进行版本控制，并将项目托管到GitHub/Gitee等平台。最终提交时提供仓库链接。
3. **演示：** 清晰的演示视频或GIF能够直观展示你的成果。
4. **DDL和提交方式：**
 - **第一次提交 DDL (例如：阶段一至阶段三)：** 2025-8-1 23:59:59
 - **第二次提交 DDL (例如：全部阶段)：** 2025-8-31 23:59:59
 - 邮件标题：**24级IoT招新-学号-姓名-阶段X**
 - 打包命名：**学号-姓名-IoT.zip** (内含各阶段文件夹，每个文件夹内含代码、笔记、演示链接等)
 - 提交所有要求的文件到QQ邮箱：3049310252@qq.com

给同学们的建议：

- **善用搜索引擎和官方文档：** 乐鑫官网、Arduino官网、各种库的GitHub页面都是宝贵的资源。
- **多动手实践：** 编程和硬件调试都需要大量实践。
- **记录问题，总结经验：** 学习笔记不仅是提交的要求，更是自己学习成长的记录。
- **积极提问，但先尝试自己解决：** 遇到问题先尝试自己搜索和思考，实在无法解决再向学长或老师请教。
- **享受过程！** 嵌入式和物联网开发充满乐趣和挑战。