

日期	阶段	🔗 后端任务 (复制去问AI)	📄 AI指令 (Prompt)	🏆 蓝桥杯任务 (重点)
1月11日 (周六)	环境搭建	1. 安装 Python 和 PyCharm。 2. 终端运行: pip install flask flask-cors pymupdf 3. 跑通 Flask Hello World。	我是Python新手。请帮我写一个最简单的 Flask Web 服务器代码，运行在 5000 端口。当访问首页 '/' 时，返回 'Hello World'。	【环境】安装 Keil, 驱动, 烧录软件。 点亮 LED，测试板子是否正常。
1月12日 (周日)	文件上传	1. 写一个 /upload 接口 (POST方法)。 2. 接收文件并保存到 'uploads' 文件夹。 3. **必须配置 CORS** 解决跨域问题。	请帮我用 Flask 写一个文件上传接口。要求： 1. 路由是 /upload，方法是 POST。 2. 接收 key 为 'file' 的文件。 3. 把文件保存到当前目录的 'uploads' 文件夹下。 4. 使用 flask-cors 库允许所有域名的跨域请求。	【基础】学习 74HC573 锁存器。 弄懂怎么控制 LED 而不影响数码管。
1月13日 (周一)	PDF解析	1. 写一个函数 parse_pdf(filepath)。 2. 用 fitz (PyMuPDF) 读取 PDF 每一页的文字。 3. 返回所有文字的字符串。	请写一个 Python 函数，使用 pymupdf (fitz) 库。 输入参数是 PDF 文件路径，功能是读取该 PDF 所有页面的文本内容，拼接成一个字符串返回。	【中断】定时器中断 (Timer0)。 **目标**：用定时器控制 LED 每秒闪烁一次，绝对不许用 delay 函数！

1月14日 (周二)	数据构造	<div>1. 模拟知识图谱数据。</div> <div>2. 简单的字符串处理：提取前10个高频词。</div> <div>3. 生成 ECharts 需要的 JSON 格式 (nodes, links)。</div>	<div>我有长文本字符串。请帮我写代码，统计出出现频率最高的 10 个词。然后把它们封装成 ECharts 关系图需要的 JSON 格式：</div> <div>{ 'nodes': [{ 'name': '词1', ...}], 'links': [{ 'source': '词1', 'target': '词2' }] }</div>	<div>【按键】独立按键 (中断/定时器扫描)。</div> <div>实现：按一下键，数码管数字+1。必须消抖。</div>
1月15日 (周三)	接口整合	<div>1. 把 PDF 解析代码放进 /upload 接口里。</div> <div>2. 逻辑：上传 -> 保存 -> 解析 -> 返回 JSON。</div> <div>3. 用 Postman 测试。</div>	<div>现在我要整合代码。请修改 Flask 的 /upload 接口：</div> <div>在文件保存成功后，立即调用刚才的 parse_pdf 函数解析它，然后直接把生成的 JSON 数据作为 HTTP 响应返回给前端。</div>	<div>【按键】矩阵键盘。</div> <div>识别 4x4 键盘的键值，显示在数码管上。</div>
1月16日 (周四)	Git协作	<div>1. 注册 Gitee/Github 账号。</div> <div>2. 学习 git add, commit, push。</div> <div>3. 把代码传给前端队友。</div>	<div>请用最通俗的语言解释 git clone, git add, git commit, git push 的用法，并给出命令行示例。</div>	<div>【通信】串口通信 (UART)。</div> <div>电脑发 'A'，单片机灯亮；电脑发 'B'，单片机灯灭。</div>
1月17日 (周五)	容错处理	<div>1. 增加 try-except 防止服务器崩掉。</div> <div>2. 解决中文文件名乱码问题 (werkzeug.utils)。</div>	<div>请优化 Flask 代码：</div> <div>1. 如果上传的不是 PDF 文件，返回 400 错误。</div> <div>2. 如果解析过程报错，不要让服务器崩溃，而是返回 500 错误信息。</div>	<div>【复习】把之前的 定时器、按键、数码管 代码整合，整理成一个干净的工程模板。</div>

1月18日 (周六)	全链路测试	1. 本地测试: 上传真实 PDF, 查看返回结果。 2. 配合前端: 让他调用你的接口, 看图表能不能画出来。	无 (主要是修 Bug, 把报错信息直接发给 AI 问)	【休息/复习】浏览一遍往年省赛真题的客观题部分。
1月19日 (周日)	🔗 闭环演示	**里程碑: 项目第一次闭环** 演示内容: 上传 PDF -> 自动生成图谱。 (能跑通就行, 不管丑不丑)	无	无 (项目优先)
1月20日 - 25日	数据库&登录	1. 引入 SQLite 数据库。 2. 实现 JWT Token 登录验证。 3. 记录文件上传历史。	请用 Flask-SQLAlchemy 和 SQLite 实现一个简单的用户登录系统。包含 User 表(id, username, password_hash) 和 生成 JWT Token 的接口。	【I2C】PCF8591 (AD/DA)。 读取光敏电阻的值, 显示在数码管上。
1月26日 - 2月7日	细节完善	1. 优化解析逻辑 (处理长文本)。 2. 增加文件列表查询接口 (GET /files)。	Flask 如何实现一个接口, 返回当前用户上传过的所有文件列表 (从数据库查询)?	【模块化】DS1302 时钟 & DS18B20 温度。 重点: 练习把这些驱动代码封装成 .c 和 .h 文件, 方便比赛直接调用。
2月8日 - 2月15日	交付周	1. 彻底冻结代码, 不许再改新功能。 2. 确保演示环境稳定。 3. 2月15日项目打包归档。	请帮我写一个 requirements.txt 文件内容, 以及一个简单的 README.md 文档, 介绍如何运行这个 Flask 后端。	【NE555】频率测量。 这是省赛难点, 必须掌握定时器计数功能。
2月16日 - 22日	真题特训	✖ 停止所有 Python 开发。	无	【刷题】第9-12届省赛真题。 每天一套, 严格限时5小时。 重点练习: 状态机 (界面切换逻辑)。

2月23日 - 3月1日 最后冲刺 ✖ 停止所有 Python 开发。 无

【冲刺】第13-14届省赛真题 (难度大)。
总结易错点：变量溢出、中断冲突、逻辑混乱。
背诵你的万能模板。