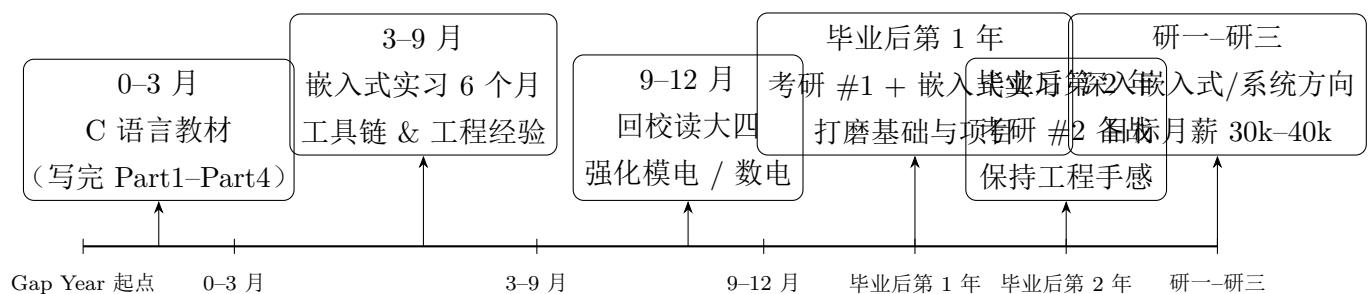


# 嵌入式人生路线图 (Gap Year 版本)

Grant Lee

2025 年 11 月 20 日

## 总体时间轴（概览）



## 阶段拆解规划

### 阶段 1 (0-3 月): 写完 C 语言教材 = 打完底层内功

目标:

- 以「写书」的形式系统学完 C 语言;
- 写出一份可反复复用的个人 C 语言教材，用作之后所有项目与工作的参考手册。

建议结构（可按需调整）：

Part 1: 基础部分（你正在写，计划本周完成）

- C 程序基本结构、编译流程;
- 变量与常量、字面量;
- 各类数据类型（整数、浮点、字符、枚举、指针类型）;
- 比特、字节、补码、IEEE 754、类型的本质;
- 小结：类型 = 对比特模式的解释方式。

## Part 2: 核心语法与指针体系

- 表达式与运算符、控制语句 (if/for/while/switch);
- 数组与字符串的内存布局;
- 指针基础: 地址、解引用、一维数组 & 指针;
- 指针与数组的等价形式:  $a[i] \Leftrightarrow *(a + i)$ ;
- 多级指针、函数指针 (点到为止即可)。

## Part 3: 组合与抽象能力

- 函数与参数传递 (值传递 vs. 地址传递);
- `struct/union`、位段、结构体内存对齐;
- 使用结构体实现简单数据模型 (如学生信息、任务描述等)。

## Part 4: 工程向部分 (实习前打底)

- 动态内存管理: `malloc/free/realloc`, 常见内存错误;
- 文件 IO: `fopen/fread/fwrite/fprintf`;
- 多文件与头文件设计: `.h + .c` 的组织方式;
- 简单构建流程: `gcc`、`Makefile` (入门级即可)。

刻意跳过 (暂不写 / 只点到为止):

- C 标准的细枝末节 (完整的 UB 列表、表达式求值顺序的极端细节);
- C11/C17 的多线程与原子操作;
- 编译器内部实现 (如 SSA、寄存器分配等)。

## 阶段 2 (3–9 月): 嵌入式实习 6 个月 = 打开工具链与工程能力

目标:

- 用 6 个月换取真正的工程经验, 而不是工资;
- 熟悉主流 MCU (如 STM32) 的完整开发流程;
- 在真实项目中巩固 C 语言和 Part 1–4 内容。

**重点收获（写在计划里，方便回顾）：**

- 编译工具链：Keil / MDK / GCC / CMake / Make 等；
- 硬件相关：GPIO、UART、SPI、I<sup>2</sup>C、ADC、定时器、NVIC；
- 调试技能：单步、断点、观察寄存器和内存；
- 代码管理：Git 工作流、代码规范、Code Review；
- 初步接触 RTOS（如 FreeRTOS）、简单任务调度逻辑。

### **阶段 3（9–12 月）：回校读大四 + 强化模电/数电**

**目标：**

- 补强「模拟电路」「数字电路」基础，使自己具备软硬皆通的嵌入式素质；
- 在课程项目中有意识地选择与嵌入式 / 硬件相关的题目；
- 整理本科学习阶段的知识，开始思考考研方向。

### **阶段 4（毕业后第 1–2 年）：考研窗口 + 实习穿插**

**第 1 年：**

- 以考研为主线，构建系统的数学与专业基础复习计划；
- 争取找到一份与嵌入式 / C 语言相关的兼职或短期实习，保持工程手感；
- 根据初次考试结果，调整学校和方向定位。

**第 2 年：**

- 若第 1 年未成功，则第 2 年作为冲刺年，专注弥补短板；
- 同样保持少量工程实践（避免毕业后「只有理论，没有手感」）。

### **阶段 5（研一–研三）：方向深入 + 项目沉淀**

**目标：**

- 选定「嵌入式系统 / 操作系统 / 驱动 / 工业控制」等相关方向深入；
- 至少完成 2–3 个具有代表性的工程项目（可在 GitHub 上公开展示）；
- 毕业时具备月薪 30k–40k 的综合实力（底层扎实 + 工程经验 + 项目说服力）。

**备注：**本计划是一个「可迭代」的版本，实际执行中可根据自身状态、机会和外界环境进行微调。核心思想是：

先打基础（ $C \otimes$  电路） $\rightarrow$ 再上工程战场（实习） $\rightarrow$ 再系统提升（考研  $\otimes$  研究生）。