



Image

嵌入式培训参考资料

2024 科协用

作者：Yuhang Gu

组织：Southeast University, School of Electronics Engineering

时间：2024

版本：Rev 1.0

目录

序	ii
0.1 正确提问和解决问题	ii
0.2 C 语言基础?	iii
0.3 实验方案	iii
0.4 实验环境	iv
0.5 如何获得帮助	v
0.6 其他说明	v
第 1 章 开始之前: 准备工作与相关注意事项	1
1.1 准备工作	1
1.2 开发环境配置	1
1.3 开发板型号以及相关参考	1

序

我在大一学习嵌入式开发的过程中曾经无数次陷入迷茫。网上的资料多而杂，要么是花费了太多时间在这个阶段不必要的细节，要么是太过简略以至于完全不明所以。另一个痛苦的来源其实是你校的计算机课程教学。几乎可以暴论，嵌入式开发学习的上限完全取决于**对计算机系统的认识程度**；我在大一留下的诸多疑问一直到我学习了**南京大学计算机系统实验**课程之后才得到解答。

所以我决定做一份学习资料（不如说更像是实验导引，就像是国外学校 EECS 课程常有的 Lab）。在这份文档中，我会尽量做到**只讲述当前有必要的细节**，应当交给视频去讲明白的就会交给视频去做；应当由你自己 RTFM, RTFSC（这两个缩写是什么意思？马上就会告诉你了）弄懂的就交由你自己去慢慢理解。

提示 0.1

本文档目前的版本目前由科协内部使用（当然你分享出去了大家也不会说什么）。

试用的一个重要目的是，希望能收集大家在自由探索过程中真实遇到的困惑和改进建议，这些建议会在此后 SEU-Project R 项目讲义的编写中发挥重要作用。

注意 0.1

本文档采用 **CC BY-NC-SA 4.0 DEED** 方式公开。

Yuhang Gu, Southeast University

提示 0.2

”我们都是活生生的人，从小就被不由自主地教导用最少的付出获得最大的得到，经常会忘记我们究竟要的是什么。我承认我完美主义，但我想每个人心中都有那一份求知的渴望和对真理的向往，”大学”的灵魂也就在于超越世俗，超越时代的纯真和理想 – 我们不是要讨好企业的毕业生，而是要寻找改变世界的力量。”

— jyy

”教育除了知识的记忆之外，更本质的是能力的训练，即所谓的 training。而但凡 training 就必须克服一定的难度，否则你就是在做重复劳动，能力也不会有改变。如果遇到难度就选择退缩，或者让别人来替你克服本该由你自己克服的难度，等于是自动放弃了获得 training 的机会，而这其实是大学专业教育最宝贵的部分。”

— etone

计算机的所有东西都是人做出来的，别人能想得出来的，我也一定能想得出来，在计算机里头，没有任何黑魔法，所有的东西，只不过是我不现在不知道而已，总有一天，我会把所有的细节，所有的内部的东西全都搞明白的。

— 翁恺

0.1 正确提问和解决问题

0.1.1 如何求助

在碰到问题求助之前 — 哦不，在开始实验之旅之前，请先简单阅读一下这两篇文章：**提问的智慧**和**别像弱智一样提问**。

提示 0.3

是的，别再问”为什么板子上的灯亮起来了但是电脑检测不到”这种问题了 - 仔细检查一下你连接开发板用的是数据线还是电源线（—它们之间有什么区别？）

注意 0.2

一定要记住: 机器(除非芯片烧了)和编译器永远是对的; 未测试的代码一定是错的; 杜邦线很有可能是连错的; 时钟有可能是忘记开了的。

不可否认的是, 嵌入式开发的软硬件结合特性和底层性注定了错误调试的难度, 但错误一定不是不可排除的黑魔法。你有万用表, 有调试器(在单片机开发时, 你同样可以通过 gdb 打断点, 看内存等), 已经足够排除 90% 的问题了。剩下的交给玄学吧。

**0.1.2 用正确的手段解决问题**

一个重要的任务是, 希望能在实验的过程中培养大家用正确的手段解决问题的能力。具体的内容在此后的讲义中将会一再涉及。

0.2 C 语言基础?

毫无疑问嵌入式开发的主力是 C 语言。尽管需要强调 **C++ 和 C 基本上是两种语言**, 但大家大一学的 C++ 仍然足够帮你应付嵌入式开发的入门需要了。然而 C 中有两件事物仍然值得强调:

- 指针, 指针的本质, 数据类型的本质。
- 结构体, 结构体的内存分布, 结构体指针, 结构体指针和各种其他事物的互转

在此, 非常推荐访问[笨方法学 C](#)进行 C 语言的学习(同样会涉及一些命令行和编译器的使用小方法, 很实用)。推荐完成练习 1-18, 31(调试器), 44(环形缓冲区)的学习。

如果你没有太多时间的话, 也请一定要重看其[练习 15: 可怕的指针](#)和[练习 16: 结构体和指向它们的指针](#)和[练习 18: 函数指针](#)的内容。不然对于之后的内容给你带来的, 可能的心灵创伤, 作者概不负责。

提示 0.4

一个有趣的问题: 能不能用 C++ 写单片机程序?

答案是肯定的。不过要完全理解这个过程还需要一些对编译过程的理解。而且你不能直接在 C 里引用 C++ 声明的函数 — 为什么?

这个问题就交给两千年后的你来 STFW 解决吧。

**0.3 实验方案****Part 1. 从零开始的单片机之旅****0.3.1 实验一: 点灯工程师**

- 流水灯
- 读取按键
- RTFM(一)
- 发生了什么?

0.3.2 实验二: Hello World!

- UART 和发送消息
- printf 在做什么?
- 接受号令吧!

0.3.3 实验三: 穿越时空的旅程

- 正片开始: 中断
- 执行流的切换: 状态机的视角
- GPIO 中断和 UART 中断
- 缓冲区, 消息队列

Part 2. 连结世界

0.3.4 实验四: IIC 初见

- EEPROM 读写
- SCL 与 SDA: 了解协议
- 从单片机到现代计算机系统: ”外设”究竟是什么?
- Hello World (二): 屏幕, 启动!

0.3.5 实验五: One Last Kiss

- 尝试 MPU6050
- SD 卡
- One Last Kiss

0.3.6 实验六: 跳动的方块

- 姿态解算
- 对它使用线性代数吧!
- 数学运算: 性能与空间的 trade-off
- * Kalman Filter

Part 3. 我逐渐开始理解一切

0.3.7 实验六: SPI 和图形库

- SPI 协议和屏幕驱动
- 使用 LVGL

0.3.8 RTOS

0.4 实验环境

- 操作系统: Windows / GNU/Linux
- 编程语言: C 语言
- IDE 环境: Keil / STM32-CubeIDE / CLion

0.5 如何获得帮助

0.6 其他说明

欢迎加入科协嵌入式培训教材编写组 / Project-R 文档编写和维护组.

关于本讲义内容的问题和建议请联系 gyh: 213221544@seu.edu.cn / 127941818 (QQ)

第 1 章 开始之前: 准备工作与相关注意事项

1.1 准备工作

1.2 开发环境配置

1.3 开发板型号以及相关参考