

第二节：初学者的疑惑。

【2.1 单片机应用的核心技术是什么？】

单片机应用的核心技术是什么？是按键，数码管，流水灯，串口。是它们的程序框架。按键和数码管是输入是人机界面，把它们的程序框架研究透了，以后做彩屏或者更花销的显示界面，程序框架也可以通用。流水灯是应用程序是 APP，把它的程序框架研究透了，以后控制飞机大炮的程序框架也是一样。串口是通讯是接口，把它的程序框架研究透了，以后搞 SPI，CAN，USB 等通讯项目时，上层的程序框架也可以通用。如果某天你突然腰酸背痛可能是缺钙了，如果某天你第一次做项目时突然发现无从下手，你缺的可能是程序框架。

【2.2 跟我学单片机到底是学什么？】

跟我学单片机到底是学什么？我的回答是像驾驶汽车一样驾驭单片机。我教给大家的是驾驶汽车的技术而不是研发汽车的技术。因此每当别人问我学 51 单片机，PIC，AVR，stm32 哪个更加有前途，应该先学哪个再学哪个时，我的回答是既然你是学驾驶技术，那么用桑塔纳车来学还是用宝马车来学有差别吗？差别很小的，它们只是不同的厂家而已，只要会一种其它的就触类旁通了。把学单片机当作考驾照这是我常用的一个比喻。

【2.3 单片机神奇的工作原理是什么？】

单片机神奇的工作原理是什么？初学者不用纠结这个问题，这不是我们学习的方向。考驾照只要大概知道汽车是由四个轮，发动机，制动系统，离合器，方向盘等部分构成就够了，再深入的细节不用纠结。学单片机只要大概知道单片机内部由运算器，寄存器，IO 口，复位电路，晶振电路等部分组成就够了，再深入的不用纠结。说实话，我本人做单片机开发有很多年了，但是我对单片机的工作原理也很模糊，就像人为什么能通过大脑来灵活控制双手，对于我仍然是一个迷。有这样的疑惑时咋办？我建议用“游戏规则”这个概念去应付它。游戏规则是不需要解释的，只要遵守就可以了。在应用的技术领域，把暂时不解的东西当作游戏规则来解读和遵守是我常用的思维方式。

【2.4 很难记住繁杂的寄存器？】

很难记住繁杂的寄存器？寄存器不用死记硬背，只要知道它大概的操作流程，有哪几类就够了。配置寄存器时，可参考别人已经配置好的代码，这些代码都很容易通过网络或者书本获得。也可以查找芯片数据手册，有很多单片机厂家会给出各个功能的代码范例。

【2.5 C 语言很难学？】

C 语言很难学？暂时不用学指针，暂时不用学结构体，暂时不用学多文件编程，暂时不用学链表，暂时不用学宏定义，暂时不用学文件操作，暂时不用学所有的数据类型。只要会：

5 条指令语句 switch 语句，if else 语句，while 语句，for 语句，=赋值语句。

7 个运算符+，-，*，/，|，&，!。

4 个逻辑关系符||，&&，!=，==。

3 个数据类型 unsigned char， unsigned int， unsigned long。

3 个进制相互转化，二进制，十六进制，十进制。

1 个 void 函数。

1 个一维数组 code(或 const) unsigned char array[]。

只要会上述一些知识点，应付任何一种逻辑功能的单片机软件都足够了。

我在学校时只学了汇编程序，出来工作后才知道用 C 语言也能开发单片机，一开始只用常见的几条语句就把项目做出来了，没有用到 printf 这类繁琐格式的语句，第一次感觉 C 语言原来是那么容易那么简单，我是到后来才陆陆续续用到 C 语言其他的高级功能。

【2.6 汇编语言很难学怎么办？】

汇编语言很难学怎么办？我提个建议，对于初学者，一开始就学汇编语言确实难学，不如先学 C 语言，学会了 C 语言再学汇编，这样理解起来就容易多了。也可以把 C 语言列入必修课，汇编语言列入选修课，因为实际工作中也是 C 语言为主。

【2.7 很难记住各种繁杂的通信协议？】

很难记住繁杂的各种通信协议？IIC, SPI, 232, CAN, USB 等等。不用记那么多，你只要理解串行和并行通讯方式的基本原理就可以了，剩下的只是不同的协议而已，工作时再根据需要去看看相关资料就可以上手。不管世上有多少种通讯协议，物理世界上只有这两种通讯方式。

【2.8 很难写出短小精悍的程序？】

很难写出短小精悍的程序？初学者不用纠结于此。很多项目开发，程序容量不是刻意追求的目标，多一点少一点没关系，不会是寸土寸金的事情，现在大容量的单片机品种也非常多，反而更值得关注的是程序的运行效率，可读性和可修改性。当然，一些成本敏感的消费类电子不在此讨论范围，这类项目往往对程序容量也要求很苛刻。