

## 跟涛哥一起学嵌入式 03：嵌入式要从u-boot移植学起？

文档说明	作者	日期
来自微信公众号：宅学部落(armLinuxfun)	wit	2018.7.3
嵌入式视频教程淘宝店： <a href="https://wanglitao.taobao.com/">https://wanglitao.taobao.com/</a>		
联系微信：brotau(宅学部落)		

跟涛哥一起学嵌入式 03：嵌入式要从u-boot移植学起？

**嵌入式真要从U-boot移植学起？**

**我的悲催学习经历**

**高频技能与低频技能**

**学习U-boot的正确姿势**

**嵌入式工程师自我修养**

俄罗斯仲夏夜，世界杯依旧如火如荼。球场上，夕阳下，梅西没落的身影、C罗的黯淡离场，并没有打击大家太多的热情。战火依旧，老兵不死，梅罗时代是否快要终结？然而新一代巨星冉冉升起，风起云涌。在看球的同时，学习充电也不能落下，今天我们聊聊第3个话题：学习嵌入式，一定要从u-boot移植学起吗？

### 嵌入式真要从U-boot移植学起？

故事缘起嵌入式技术群(475504428)一位嵌入式学员遇到的问题：这位学员想在他的开发板上移植最新版本的U-boot。然而，将近半年过去了，发现还在群里问一些关于移植的一些问题，这就引发我的思考：照这样的速度，再给他几个月，能否移植成功？对于一个新手，没有辅助的一些工具，费这么大力气，去做一些跟以后的工作不太相关的技能，是否划算？当看到这种学习方式给很多新手带来的不是进步的喜悦，而是打击，甚至是转行去做其它，我觉得我们也许应该换种学习方式了，包括我在内，在上面也栽过不少坑。

### 我的悲催学习经历

跟大家学习的经历差不多，学生时代，玩完51单片机后，我就购买了第二块开发板：某款ARM9开发板，这里就不说具体的名字了。按照流程，裸机实验做了一遍后，就开始开启了U-boot的移植生涯，在烧写U-boot时发现根本启动不起来，当时由于没有相关的调试工具，比如TRACE等，也不会相关的调试打印手段，所以只能尝试各种可能的原因——排除，逛各种论坛、到百度翻页，发现根本解决不了问题。为此前前后后折腾了几个月，被折腾得有气无力。到最后，联系厂商的工程师，开发板寄回去后，发现是NOR FLASH芯片的问题，重新换了一块NOR FLASH，系统就可以正常运行了。虽然这个问题最后得到了解决，但是前前后后也浪费了我几个月的时间，当然，有人可能会说，通过这个bug你也学到很多东西啊。是的，通过不断分析这个问题，确实增长了不少知识和技能，但是对于新手来说，如果一上来就通过问题驱动来学习，是很容易打击人的学习积极性的。大部分人可能会坚持不下去，折腾得差不多了，也就把板子扔到一边落灰，去搞互联网、Java去了，从而改变了自己的职业道路。为了这个

bug，我浪费了几个月的时间，我觉得是不划算的，这几个月的宝贵时间，我可以去学习以后嵌入式开发中用到的高频技能，提高我的工作效率，而不是在这个低频技能上耗费太多的时间。

## 高频技能与低频技能

什么是高频技能，低频技能呢？写这两个词时，特地到百度搜索了一下，发现并没有相关的词汇，姑且看作是我创造的两个新词吧(版权归我啊，用时要交税~)。顾名思义，高频技能就是我们在工作中大概率使用到的技能；低频技能，就是我们在工作中很少使用的技能。

早期的单片机开发，系统比较简单，从硬件设计、PCB画图、软件开发、测试，一个工程师就可以搞定，软硬通吃。按照今天的流行语，就叫全栈工程师。但现在不同了，嵌入式开发，尤其是上了OS后的嵌入式系统开发，需要的理论基础和技能储备太多太杂：从硬件、计算机系统结构、操作系统、应用层、GUI等，都要有所涉猎。同时，随着嵌入式越来越复杂，分工也越来越明确：硬件工程师、PCB工程师、BSP工程师、驱动工程师、应用工程师、测试工程师，大家以一个团队的形式共同开发维护一个嵌入式项目。

分工的形成，以前全栈工程师需要的软硬技能，就不需要全部学习了。当然，你也学不完，就算你学完了，有些知识你一辈子也可能用不到。分工机制下，每个人精通各自的模块，分工协作，效率最高。现在一个嵌入式SOC，几十个IP模块，你说你都掌握，有点不现实，一个音视频解码、GPS、GPU、USB都可能让你啃几年才能谈得上精通。系统越来越复杂，分工越来越明确，这就需要我们可能要换一种新的方式了。

对于嵌入式学习者来说，面对越来越复杂的系统，我们该如何学习，学习什么，这是一个很值得思考的问题。在思考这个问题之前，我们首先要有一个前提：人的精力是有限的，不可能掌握所有的知识和技能。如果你不认可这个前提，非得按照那种全栈工程师的模式学习，那下面就没有看的需要了。

该如何学习呢？我们需要对嵌入式开发，需要的技能进行重构和分解。哪些是常用的，基础的，必须掌握的，即高频技能，哪些是不常用的，即低频技能，然后再根据需要，科学地学习。这才是嵌入式2.0时代，学习嵌入式的正确姿势，适应职场新形势新需求的学习捷径。

## 学习U-boot的正确姿势

同样一个U-boot，不同的职位、不同的开发人员，对它的掌握程度是不一样的。一个团队就和一个特种小分队一样：狙击手要求精通狙击、爆破手要精通各种爆破技术、通信兵要掌握各种电子情报获取技术。团队中不同的成员精通不同的技能，集合在一起，就可以形成恐怖的战斗力的。人的精力是有限的，精通一门技术的同时，就不可能在另一门技术上花费同样的心力了，学习U-boot也是一样，你也不必什么都学。

那对于新手来说，U-boot要学习什么呢？我觉得两点就够了：U-boot的启动流程和U-boot跟内核的传参交互。这些是基础，也是各种开发人员经常接触到的。至于各种启动方式、底层实现的一些细节，这些原厂都已经做好了，我们直接用就可以了。软件开发中有软件复用的思想，同样，嵌入式系统开发中，我们也要有这种思想，不要什么都自己干，什么轮子都自己造，尤其是硬件，尤其是新手！高手造轮子，一般人用轮子，这是规律，先学会用轮子造汽车，在慢慢学着造更好的轮子。因为很多硬件上的坑、芯片内部的bug，这些都是不可见的，不会对外公开的，很多都是通过软件规避、或者其它手段曲线救国来实现的，这些东西你可能永远都不会知道，所以，如果你是新手，尽量不要跌到这些坑里，可以参考我上面所说的学习方式。

## 嵌入式工程师自我修养

《嵌入式工程师自我修养》体系教程，就是根据这种思路，适应嵌入式新时代需求、对嵌入式中的高频技能、低频技能不断重构，分解，重新设计而成的一种培训教程体系，也是我未来几年努力打造的一套培训课程。跟随这套教程学习的学员也许会发现，里面的相关课程，跟目前市面上、培训机构的都不太一样，没错，这也算是我的一个小小原创，互联网上首家使用新概念：高频技能、低频技能重新打造的一门嵌入式自学体系教程。

这门教程的威力在哪里？一般的教程，大家学到最后，学到的都是同样的知识和技能，就像流水线一样，出来的都是同样的产品。而《嵌入式工程师自我修养》体系教程，不是这样：每个人的学习强度不同、选择方向不同，最后结果也不一样：你有可能发现太难、学不下去，知难而退；有可能是基础扎实的普通士兵；也有可能是身怀绝技的特种兵；也有可能是身兼数门绝技的兵王。课程的难度强度梯度足够大，对你的技术成长上限不设限，它跟你的目标、要投入的时间和精力成正比，只要你肯吃苦、愿意努力，你可以把你的潜力推高到极致。对你，对我，都是同样的挑战。

《跟涛哥一起学嵌入式》，会持续跟大家分享嵌入式相关技术、学习方法、学习路线、求职面试等。

专注嵌入式、Linux精品教程：<https://wanglitao.taobao.com/>

嵌入式技术教程博客：<http://zhaixue.cc/>

联系 QQ：3284757626

嵌入式技术交流QQ群：475504428

微信公众号：宅学部落(armlinuxfun)

