跟涛哥一起学嵌入式 06：后ARM时代，嵌入式工程师的自我修养

文档说明	作者	日期
来自微信公众号：宅学部落 (armLinuxfun)	wit	2017.7.13
嵌入式视频教程淘宝店：https://wanglitao.taobao.com/		
联系微信：brotau(宅学部落)		

大家好，我是涛哥，今天我们聊一聊：后 ARM时代，嵌入式该如何学习。

1. 嵌入式学习的一些概念理解误区

很多嵌入式初学者认为，学嵌入式，就是学习 ARM，就是学习开发板。买一块开发板，然后在上面“移植” U-boot、Linux 内核，再使用 Busybox 制作一个根文件系统，大功告成！觉得可以出去找工作了。这其实是有一定片面性的：首先 ARM 是个CPU架构，跟 PC 上的 X86 架构一样，你见过有人在 Windows 下面学习 C/C++ 编程、MFC 编程、网络编程、互联网编程，说自己学习 X86 的吗？当然，也不可否认，嵌入式平台的多样性、硬件的可定制性导致我们在嵌入式平台上开发应用程序、驱动之前，首先要搭建这个平台，就像我们在 Windows下面要装操作系统一样，但是这仅仅是我们学习嵌入式开发的第一步。

其次，关于系统的“移植”，很多人玩了开发板之后，会在自己的简历上写自己移植过 u-boot，Linux 内核……其实，这种写法也是有点瑕疵的。真正的移植，往一个新的芯片或开发板上 porting 一个 u-boot 或 Linux 内核，那可不是一个人能干的事情，是一个团队干的事情。时钟、DDR、存储，可能牵涉到各个模块，哪里遇到问题，都需要各个模块的 owner 去 debug，有时候甚至可能是芯片的 bug，或者硬件开发板的 bug，这就需要我们使用软件去解决、去规避这个坑，这都需要我们在很短时间，甚至一两天的时间去解决这个问题，需要一个团队的各个模块专家合力完成。所以说，我们所说的“移植”，其实就像是在 Windows 下面安装操作系统，按照步骤完成装机。当然，通过这个过程，可以加深我们对嵌入式系统的理解，但是我们首先要知道的是，我们“移植”的系统，都是芯片公司团队做好的系统镜像，我们做的只是配置、编译、安装、甚至升级这些基本的操作。这些环境只是我们学习嵌入式开发的平台，万里长征才走完了第一步。

2. 学习嵌入式，我们到底该学些什么？

嵌入式越来越复杂，一个 SOC 芯片上集成的模块越来越多。以手机为例，典型的嵌入式产品，我们看看上面集成了多少模块：触摸屏、LCD、USB、WiFi、4G等无线通信、音视频编解码 IP、DDR、存储控制器、3D/2D 加速、GPS、指纹识别、NFC、DMA、G-sensor 各种传感器.....。可以说，现在一个手机的复杂度和硬件配置，已经超过我们的桌面 PC 了。除了不断增加的硬件，软件方面，比如 Linux 内核，光内核代码就有 1000 多万行，每天更新的速度超过你学习的进度，你能学得完嵌入式的所有知识和技能吗？

早期 PC 时代，我们知道能做出 X86 CPU量产的也没有几家，Intel、AMD 和威盛。但是嵌入式时代不一样了，ARM 的 IP 授权模式导致不同的芯片厂商百家齐放，不同的 SOC 平台和开发板眼花缭乱，针对不同行业需求定制的 SOC 平台雨后春笋：手机芯片、平板芯片、视频安防、物联网、汽车电子、工业控制，甚至人工智能AI芯片....，你到 Linux 内核的 ARCH 下面可以看看有多少种 CPU 架构，再到 arch/arm 下面看看有多少种开发平台，这还只是加入到内核 mainline 的平台，算上没有加入 Linux 内核主线的各种平台，其实数量更多。

众多的芯片架构、不同的开发板平台，我们该如何去学习？

嵌入式和 PC 的概念也越来越模糊了，Intel 已经推出 X86 架构的 CPU 和嵌入式产品了，比如平板。ARM 也开始进军服务器和笔记本领域了。无论什么 CPU 架构，ARM、X86、MIPS、PowerPC，还有最近火热的物联网芯片，无论是做嵌入式产品，还是 PC、服务器，他们的底层本质其实都没有变，都是计算机原理和系统架构，都是冯诺依曼的计算机架构，图灵原型机的各种实现。

不断复杂的软硬件系统，对嵌入式工程师或者学习者来说是一个挑战。这对我们本身的知识和技能有一个更新的要求。早期 51 单片机时代，我们可以自己使用面包板或者自己画 PCB，做一个开发板，然后在上面开发软件。软件、硬件自己全搞。现在不断复杂的 SOC 平台，再想一个人全搞，软硬通吃，基本不可能，这也导致我们需要分工协作来完成。首先软硬件的分工，各司其职，各自精通自己的领域，然后进行软硬件整合，协作开发。再次，软件方面，嵌入式软件也越来越复杂，Linux 内核 1000 多万行，android 源码下载下来就占几个G的空间，自己想全搞，同样不可能，同样需要进行分工。比如 android，需要分为BSP工程师、Linux内核工程师、驱动工程师、android 中间层开发工程师、APP 开发工程师。对于一个 Linux 内核，也需要分工，各个模块同样进行分工：Linux 内核的 USB子 系统、音频子系统、视频编解码、文件系统.....把其中一个模块你搞精通了，工资绝对不是问题。

对于嵌入式学习者来说，我们该学习什么，或者说如何学习？才能提高自己的职场竞争力，或者说对于一个新手来说，如何通过自学，达到公司的用人标准和技术要求，找到一份自己想要的工作？

首先，你要学会做减法，从现实出发，要有这样一个意识：我不可能精通所有的嵌入式技术，学会坚持，制定合理现实的小目标。很多人喜欢那种不切实际的广告轰炸营销，击中你心理上的某个软肋，某个G点，一下子兴奋起来。越熬越浓的心灵鸡汤，并不能解决我们吃饭的生存现实问题。很多人，包括我，在学习的时候，都喜欢给自己树立各种路线、计划、日程表。制定计划时激情满满，热情高涨，激动得睡不着觉。计划宏伟而饱满，仿佛成功就在眼前。但是往往不切实际，往往在早期，遇到各种困难，各种坑，各种拖延导致没有坚持下来，最后夭折。然后接着制定下一个宏伟的计划，继续夭折，生活周而复始，day after day。观察我们生活周围，真正做出成绩的都是那些基于现实出发，能一路坚持下来的人，day by day。有时候你会发现，并不觉得他们有多聪明。

其次，保持自己的兴趣，说白了就是为了坚持下去。见过很多人想学习嵌入式，花了很多米买一块开发板，激情满满，过一段是过去再看，已经不折腾了。嵌入式开发难，难在哪里呢？主要在于开发环境的搭建，软件调试上，不像在 Windows 上使用 VC 开发程序，集成开发环境都帮你弄好了，各种断点、单步、查看堆栈、寄存器、内存窗口。而嵌入式不一样，硬件环境搭建会遇到各种各样的问题，各种电脑的兼容问题，各种莫名其妙的问题，有时候着实让人抓狂，时间久了，慢慢地学习的激情殆尽，也就不想学习了。这还不算什么，更严重的是，很多人学习嵌入式遇到挫折，往往会打击人的自信，觉得自己能力不行，智商不够，不适合干这行，在心理留下了阴影。对于个人学习者来说，买了开发板，你不买配套的万用表、示波器等调试设备，遇到硬件问题也是一筹莫展，无法解决。其实我们可以完全使用其它的平台去开展我们的研究和学习，比如 QEMU，一款可以仿真开发板的开源软件，使用这款开源软件，我们可以在电脑上虚拟一个世面上流行的开发板，然后再在这个仿真的开发板上跑 u-boot、Linux 内核、挂载根文件系统，使用和开发板一样的源码，运行效果和真实的开发板是一样的。而且，使用 QEMU 的好处就是，“硬件”永远不会出问题，可以让我们避过硬件的各种坑，腾出更多的精力去研究嵌入式软件的各种架构、编程技能、内核驱动....，这些才是嵌入式工程师的核心竞争力，需要花大量的时间不断地去积累，去磨合，去提高的。把大量的时间耗在一个本该不属于学习范畴的硬件bug上或者硬件环境不兼容上，不划算，因为你以后进公司后，遇到同样的问题，找硬件工程师，半分钟帮你搞定。所以说，选择一个理想的嵌入式学习平台，尤其对于初学者来说，很重要。

最后，要保持学习的深度，刻意练习。不要让自己永远待在学习的舒适区，要学会挑战自己，不断去扩展自己知识的边界，完善自己的知识体系和技能。很多人买了开饭，按照教程，“移植”了 u-boot，Linux 内核，制作了根文件系统，然后就陷入了迷茫：接着要干什么？要学习什么？想学习又感觉深入不下去，东一耙子，西一耙子，看看这，看看那，时间不知不觉就过去了。其实，学习嵌入式，基本的嵌入式知识和理论学习还是必要的，很多人推崇边做边学，到项目中学习，实践出真知。当然这也是一个方法，但是也有弊端，那就是学习的不系统，很多有心人到后来还是得回来补课，完善自己的知识体系和技能。很多人玩开发板，烧写镜像，玩得贼溜，但是你知道这里面的原理吗？知道 JTAG 怎么下载的吗？JLink 和 JTAG 有什么区别？为什么 PC 上要装个JTAG 软件而 Jlink 不用？程序的编译和链接是怎么样的？为什么内核镜像要下载内存的某个地址？换个地址行不行？为什么我们编写的程序要在有 OS 的环境下运行，在 ARM 开发板裸机环境下，你能写一个跑起来的程序吗？只有对这些问题深入思考，你才会对嵌入式有一个更深的认识，超越了平台，一通百通。

3. 80%的嵌入式知识和技能，其实跟硬件平台无关

嵌入式开发需要的知识体系和技能，80% 其实跟硬件平台无没有无关系的。比如计算机系统原理、编程技能、程序的编译链接、你对 Linux 内核的理解、设备模型、驱动架构、项目管理等等。

真正跟硬件平台有关的，比如驱动开发，上面的框架是跟平台无关的，下面跟各个硬件平台的适配部分，可能跟硬件平台就有关系了，寄存器配置、开发板硬件配置等。而对于嵌入式工程师来说，尤其是驱动开发工程师，等你工作后，你会发现，跟应用开发相比，真正要写的代码量很少，往往只需要改几行代码。但是往往这几行的代码量，需要你深厚的背景知识：硬件知识、通信协议、对芯片、开发平台资源掌握、对 Linux 内核架构、设备模型、驱动框架的理解，这些才是嵌入式工程师的核心竞争力。

如果你看到很多广告还在以开发板或者平台作为噱头，能拿多少工资作为宣传，这时候你的脑海里要有这种意识，这是一种推广宣传。工资多少是由你自己的水平和市场大行情决定的，虽然在面试时 HR 会对你本身的水平评估有一些误差，但是要相信，时间会证明你自己的真实价值，不断提高自己的知识水平和技能才是王道。真正的技术需要自己花时间慢慢吸收、积累、消化，内化为自己的知识体系和技

能。外在的心灵鸡汤或高煲老鸭汤，只能让你一时地热情高涨，产生暂时的错觉，并不能真正的提高技能。

4.《嵌入式工程师自我修养》系列教程规划

从课程体系设计，到课程录制。全部由多年工作经验的一线芯片原厂的嵌入式驱动工程师录制，丢弃嵌入式过时不用的技术，更新为前沿的技术、知识技能。对嵌入式知识体系和技能重新打乱重排，更加符合初学者，缩短学习曲线。

一期课程，主要学习嵌入式基础理论、嵌入式软件架构等核心技术、编程技能、编译链接等计算机底层系统知识、Linux内核，应用开发、设备驱动。这些技能都是跟硬件平台无关的，在 QEMU 仿真平台上学习，一方面可以节省学习成本，另一方面，规避嵌入式环境搭建、硬件的各种坑，是一个理想的学习平台，只要有一台电脑、一根网线就可以学习。

二期课程：在A9等至少2款开发板上进行系统移植、驱动开发等实战，做别人没有做过的，解决别人没有解决过的问题。积累自己真正的项目实战经验。在开发板上开展多个项目开发，积累嵌入式项目开发经验。

课程体系也会不断更新，完善，但宗旨不变，只有一个：用最短的时间，降低学习难度，学到最核心的嵌入式技术。

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)


ARM