

硬件工程师到底值不值钱，推荐一篇文章附下：

这里指嵌入式硬件工程师

网上看到众多网友都问了关于嵌入式系统方面的很多问题，很多都可在这里找到答案，希望我的这篇文章能给他们以启发。也请大家不要轻易转载。

一、嵌入式系统的概念

着重理解“嵌入”的概念

主要从三个方面上来理解。

1、从硬件上，将基于 CPU 的处围器件，整合到 CPU 芯片内部，比如早期基于 X86 体系结构下的计算机，CPU 只是有运算器和累加器的功能，一切芯片要造外部桥路来扩展实现，象串口之类的都是靠外部的 16C550/2 的串口控制器芯片实现，而目前的这种串口控制器芯片早已集成到 CPU 内部，还有 PC 机有显卡，而多数嵌入式处理器都带有 LCD 控制器，但其某种意义上就相当于显卡。比较高端的 ARM 类 Intel Xscale 架构下的 IXP 网络处理器 CPU 内部集成 PCI 控制器（可配成支持 4 个 PCI 从设备或配成自身为 CPI 从设备）；还集成 3 个 NPE 网络处理器引擎，其中两个对应于两个 MAC 地址，可用于网关交换用，而另外一个 NPE 网络处理器引擎支持 DSL，只要外面再加个 PHY 芯片即可以实现 DSL 上网功能。IXP 系列最高主频可以达到 1.8G，支持 2G 内存，1G×10 或 10G×1 的以太网口或 Febr channel 的光通道。IXP 系列应该是目标基于 ARM 体系系统结构下由 intel 进行整合后成 Xscale 内核的最高的处理器了。

2、从软件上前，就是在定制操作系统内核里将应用一并选入，编译后将内核下载到 ROM 中。而在定制操作系统内核时所选择的应用程序组件就是完成了软件的“嵌入”，比如 WinCE 在内核定制时，会有相应选择，其中就是 wordpad,PDF,MediaPlay 等等选择，如果我们选择了，在 CE 启动后，就可以在界面中找到这些东西，如果是以前 PC 上将的 windows 操作系统，多半的东西都需要我们得新再装。

3、把软件内核或应用文件系统等东西烧到嵌入式系统硬件平台中的 ROM 中就实现了一个真正的“嵌入”。

以上的定义是我在 6、7 年前给嵌入式系统下自话侧重于理解型的定义，书上的定义也有很多，但在这个领域范围内，谁都不敢说自己的定义是十分确切的，包括那些专家学者们，历为毕竟嵌入式系统是计算机范畴下的一门综合性学科

二、嵌入式系统的分层与专业的分类。

嵌入式系统分为 4 层，硬件层、驱动层、操作系统层和应用层。

1、硬件层，是整个嵌入式系统的根本，如果现在单片机及接口这块很熟悉，并且能用 C 和汇编语言来编程的话，从嵌入式系统的硬件层走起来相对容易，硬件层也是驱动层的基础，一个优秀的驱动工程师是要能够看懂硬件的电路图和自行完成 CPLD 的逻辑设计的，同时还要对操作系统内核及其调度性相当的熟悉的。但硬件平台是基础，增值还要靠软件。硬件层比较适合于，电子、通信、自动化、机电一体、信息工程类专业的人来搞，需要掌握的专业基础知识有，单片机原理及接口技术、微机原理及接口技术、C 语言。

2、驱动层，这部分比较难，驱动工程师不仅要能看懂电路图还要能对操作系统内核十分的精通，以便其所写的驱动程序在系统调用时，不会独占操作系统时间片，而导至其它任务不能动行，不懂操作系统内核架构和实时调度性，没有良好的驱动编写风格，按大多数书上所说添加的驱动的方式，很多人都能做到，但可能连个初级的驱动工程师的水平都达不到，这样所写的驱动在应用调用时就如同 windows 下我们打开一个程序运行后，再打开一个程序时，要不就是中断以前的程序，要不就是等上一会才能运行后来打开的程序。想做个好的驱动人员没有三、四年功底，操作系统内核不研究上几编，不是太容易成功的，但其工资在嵌

入式系统四层中可是最高的。

驱动层比较适合于电子、通信、自动化、机电一体化、信息工程类专业尤其是计算机偏体系结构类专业的人来搞，除硬件层所具备的基础学科外，还要对数据结构与算法、操作系统原理、编译原理都要十分精通了解。

3、操作系统层，对于操作系统层目前可能只能说是简单的移植，而很少有人来自己写操作系统，或者写出缺胳膊少腿的操作系统来，这部分工作大都由驱动工程师来完成。操作系统是负责系统任务的调试、磁盘和文件的管理，而嵌入式系统的实时性十分重要。据说，XP 操作系统是微软投入 300 人用两年时间才搞定的，总工时是 600 人年，中科院软件所自己的女娲 Hopen 操作系统估计也得花遇几百人年才能搞定。因此这部分工作相对来讲没有太大意义。

4、应用层，相对来讲较为容易的，如果会在 windows 下如何进行编程接口函数调用，到操作系统下只是编译和开发环境有相应的变化而已。如果涉及 Java 方面的编程也是如此的。嵌入式系统中涉及算法的由专业算法的人来处理的，不必归结到嵌入式系统范畴内。但如果涉及嵌入式系统下面嵌入式数据库、基于嵌入式系统的网络编程和基于某此应用层面的协议应用开发（比如基于 SIP、H.323、Astrisk）方面又较为复杂，并且有难度了。

三、目标与定位。

先有目标，再去定位。

学 ARM，从硬件上讲，一方面就是学习接口电路设计，另一方面就是学习汇编和 C 语言的板级编程。如果从软件上讲，就是要学习基于 ARM 处理器的操作系统层面的驱动、移植了。这些对于初学来说必须明确，要么从硬件着手开始学，要么从操作系统的熟悉到应用开始学，但不管学什么，只要不是纯的操作系统级以上基于 API 的应用层的编程，硬件的寄存器类的东西还是要能看懂的，基于板级的汇编和 C 编程还是要会的。因此针对于嵌入式系统的硬件层和驱动层的人，ARM 的接口电路设计、ARM 的 C 语言和汇编语言编程及调试开发环境还是需要掌握的。

因此对于初学者必然要把握住方向，自己的目标是什么，自己要在那一层面上走。然后再着手学习较好，与 ARM 相关的嵌入式系统的较为实际的两个层面硬件层和驱动层，不管学好了那一层都会很有前途的。

如果想从嵌入式系统的应用层面的走的话，可能与 ARM 及其它体系相去较远，要着重研究基嵌入式操作系统的环境应用与相应开发工具链，比如 WinCe 操作系统下的 EVC 应用开发（与 windows 下的 VC 相类似），如果想再有突破就往某些音视频类的协议上靠，比如 VOIP 领域的基于 SIP 或 H.323 协议的应用层开发，或是基于嵌入式网络数据库的开发等等。对于初学者来讲，要量力而行，不要认为驱动层工资高就把它当成方向了，要结合自身特点，嵌入式系统四个层面上那个层面上来讲都是有高人存在，当然高人也对应的高工资，我是做硬件层的，以前每月工资中个人所得税要被扣上近 3 千大元，当然我一方面充当工程师的角色，一方面充当主管及人物的角色，两个职位我一个人干，但上班时间就那些。硬件这方面上可能与我 PK 的人很少了，才让我拿到那么多的工资。

四、开发系统选择。

很多 ARM 初学者都希望有一套自己能用的系统，但他们住住会产生一种错误认识就是认为处理器版本越高、性能越高越好，就象很多人认为 ARM9 与 ARM7 好，我想对于初学者在此方面以此入门还应该理智，开发系统的选择最终要看自己往嵌入式系统的那个方向上走，是做驱动开发还是应用，还是做嵌入式系统硬件层设计与板级测试。如果想从操作系统层面或应用层面上走，不管是驱动还是应用，那当然处理器性能越高越好了，但这个东西自学，有十分大的困难，不是几个月或半年或是一年二年能搞定的事。

在某种意义上讲，ARM7 与 9 的差别就是在某些功能指令集上丰富了些，主频提高一些

而已，就比如 286 和 386。对于用户来讲可能觉察不到什么，只能是感觉速度有些快而已。

ARM7 比较适合那些想从硬件层面上走的人，因为 ARM7 系列处理器内部带 MMU 的很少，而且比较好控制，就比如 S3C44B0 来讲，可以很容易将 Cache 关了，而且内部接口寄存器很容易看明白，各种接口对于用硬件程序控制或 AXD 单步命令行指令都可以控制起来，基于 51 单片机的思想很容易能把他搞懂，就当成个 32 位的单片机，从而消除很多 51 工程师想转为嵌入式系统硬件 ARM 开发工程师的困惑，从而不会被业界某些不是真正懂嵌入式烂公司带到操作系统层面上去，让他们望而失畏，让业界更加缺少这方面的人才。而嵌入式系统不管硬件设计还是软件驱动方面都是十分注重接口这部分的，选择平台还要考察一个处理器的外部资源，你接触外部资源越多，越熟悉他们那你以后就业成功的机率就越高，这就是招聘时所说的有无“相关技能”，因为一个人不可能在短短几年内把所有的处理器都接触一遍，而招聘单位所用的处理器就可能是我们完全没有见过的，就拿台湾数十家小公司（市价几千万）的公司生产的 ARM 类处理器，也很好用，但这些东西通用性太差，用这些处理器的公司就只能招有相关工作经验的人了，那什么是相关工作经验，在硬件上讲的是外围接口设计，在软件上讲是操作系统方面相关接口驱动及应用开发经验。我从业近十年，2000 年 ARM 出现，我一天始做 ARM7,然后直接跑到了 Xscale（这个板本在 ARM10-11 之间），一做就是五年，招人面试都不下数百人，在这些方面还是深有体会的。

我个人认为三星的 S3C44b0 对初学者来说比较合适，为什么这么说哪？因为接口资源比较丰富，技术成熟，资料较多，应该十分适合于初学者，有问题可能很容易找人帮且解决，因为大多数人都很熟悉，就如同 51 类的单片机，有 N 多位专家级的人物可以给你帮忙，相关问题得以很快解答，所以业界认为这款 ARM 都做用得烂了，但对于初学者来，就却是件好事。

因此开发系统的选择，要看自己的未来目标方向、要看开发板接口资源、还要看业界的通用性。

五、如何看待培训。

首先说说我自己，我目前从业近十年，与国内嵌入式系统行业共同起步，一直站在嵌入式系统行业前沿，设计过多款高端嵌入式系统平台产品并为众多公司提供过解决方案，离职前为从事 VOIP 的美资公司设计 IP-PBX，历任项目经理、项目主管、技术总监、部门经理，积累众多人脉，并集多年经验所得，考虑到学生就业与公司招人的不相匹配，公司想招人招不到，而学生和刚毕业的工程师想找份工作也不容易，于此力创知天行科技有限公司，开展嵌入式系统教育培训。

因一线的科研人员和一线的教师不相接触，导致国内嵌入式人才缺乏，国外高校的技术超前于业界公司，而国内情况是业界公司方面的嵌入式系统技术要远远领先于高校。为架构业界与高校沟通的桥梁，把先进技能带给高校学子，为学生在就业竞争中打造一张王牌，并为业界工程师快速提升实现自我创造机遇，我就这样辞去了外企年薪 20 多万的职位，做嵌入式系统方面的培训了。

对于培训来讲，是花钱来买时间，很多工程师都喜欢自己学，认为培训不值，这也是有可能的，纯为赚钱的培训当然不会太有价值，但对于实力型的培训他们可能就亏大了，有这样一笔帐不知他们算过没有，如果一个一周的培训，能带给他们自学两年后才能掌握的知识，在培训完后他们用三个月到半年时间消化培训内容，这样他会省约至少一年半的时间来学其它的或重新站在另一个高度上工作，那么他将最迟一年后会拿到他两年后水平所对应的工资，就是在工资与水平对应的关系上比同批人缩短一年，每月按最少 1 千计，再减去培训费用至少多 1.0 万，同时也省了一年时间，不管是休闲也好，再继续提高也好，总之是跑到了队伍的前面了。

另一层面上讲，对于新人的培训相当于他们为自己提前买了份失业保险，有师傅会带领他们入道，我今年暑假时班里最年轻的一个学生是大二的，今年才上大三，这学期才刚学单片机，但现在 ARM 方面的编程工作已经搞得有声有色了，再过一年多毕业，他还会失业吗？

再者通过培训，你可以知道很多业界不为常人所知的事，同时也为自己找了个师傅，就比如说，两个工程师分别用 S3C2410 和 PXA255 来做手持设备，同样两人都工作四年，再出去找工作，两人工资可能最多可相差一倍，为什么？这就是业界不为常人所知的规则，2410 属于民品，被业界用烂了，做产品时成本特敏感，当然也对人才成本敏感了，PXA255 是 intel 的东西，一个 255 CPU 能买三个 2410，一直被业界定义为贵族产品，用的公司都是大公司或为军方服务的公司，不会在乎成本，只要把东西做好，一切都好说，但这方面做的人也少啊，因为开发系统贵啊。

对于说为自己找了个好师傅，我想是这样的，因为同级工程师间存在着某此潜在的竞争关系，有很多人不愿意把自己知道的东西教给别人，这意味着他将要失业，就是所说的教会徒弟，饿死师傅，但对于我们这些人就不存在这样的关系了，我是在嵌入式系统平台设计上走到了一定程序，目前在国内这块的技术上已经是自己很难再突破自己，因此很多东西我对大家都是 OPEN 的，就比如说下面那部分关于接口设计中所提到的时序接口东西，我要是不讲，即使是高级硬件工程师我想也几乎只有 10% 的人能知道吧。

六、成为高级嵌入式系统硬件工程师要具备的技能。

首先我声明，我是基于嵌入式系统平台级设计的，硬件这个方向我相对来讲比较有发言权，如果是其它方面所要具备的基本技能还要和我们培训中心其它专业级讲师沟通，或去网站看看 www.zt-training.com。他们的方面上我只能说是知道些，但不是太多，初级的问题也可以问我。

对于硬件来讲有几个方向，就单纯信号来分为数字和模拟，模拟比较难搞，一般需要很长的经验积累，单单一个阻值或容值的精度不够就可能使信号偏差很大。因此年轻人搞的较少，随着技术的发展，出现了模拟电路数字化，比如手机的 Modem 射频模块，都采用成熟的套片，而当年国际上只有两家公司有此技术，自我感觉模拟功能不太强的人，不太适合搞这个，如果真能搞定到手机的射频模块，只要达到一般程度可能月薪都在 15K 以上。

另一类就是数字部分了，在大方向上又可分为 51/ARM 的单片机类，DSP 类，FPGA 类，国内 FPGA 的工程师大多是在 IC 设计公司从事 IP 核的前端验证，这部分不搞到门级，前途不太明朗，即使做个 IC 前端验证工程师，也要搞上几年才能胜任。DSP 硬件接口比较定型，如果不向驱动或是算法上靠拢，前途也不会太大。而 ARM 单片机类的内容就较多，业界产品占用量大，应用人群广，因此就业空间极大，而硬件设计最体现水平和水准的就是接口设计这块，这是各个高级硬件工程师相互 PK，判定水平高低的依据。而接口设计这块最关键的是看时序，而不是简单的连接，比如 PXA255 处理器 I2C 要求速度在 100Kbps，如果把一个 I2C 外围器件，最高还达不到 100kbps 的与它相接，必然要导致设计的失败。这样的情况有很多，比如 51 单片机可以在总线接 LCD，但为什么这种 LCD 就不能挂在 ARM 的总线上，还有 ARM7 总线上可以外接个 Winbond 的 SD 卡控制器，但为什么这种控制器接不到 ARM9 或是 Xscale 处理器上，这些都是问题。因此接口并不是一种简单的连接，要看时序，要看参数。一个优秀的硬件工程师应该能够在没有参考方案的前提下设计出一个在成本和性能上更加优秀的产品，靠现有的方案，也要进行适当的可行性裁剪，但不是胡乱的来，我遇到一个工程师把方案中的 5V 变 1.8V 的 DC 芯片，直接更换成 LDO，有时就会把 CPU 烧上几个。前几天还有人希望我帮忙把他们以前基于 PXA255 平台的手持 GPS 设备做下程序优化，我问了一下情况，地图是存在 SD 卡中的，而 SD 卡与 PXA255 的 MMC 控制器间采用的 SPI 接口，因此导致地图读取速度十分的慢，这种情况是设计中严重的缺陷，而不是程序的问题，因此我提了几条建议，让他们更新试下再说。因此想成为一个优秀

的工程师，需要对系统整体性的把握和对已有电路的理解，换句话说，给你一套电路图你终究能看明白多少，看不明白 80%以上的话，说明你离优秀的工程师还差得远哪。其次是电路的调试能力和审图能力，但最最基本的能力还是原理图设计 PCB 绘制，逻辑设计这块。这是指的硬件设计工程师，从上面的硬件设计工程师中还可以分出 ECAD 工程师，就是专业的画 PCB 板的工程师，和 EMC 设计工程师，帮人家解决 EMC 的问题。硬件工程师再往上就是板级测试工程师，就是 C 语功底很好的硬件工程师，在电路板调试过程中能通过自己编写的测试程序对硬件功能进行验证。然后再交给基于操作系统级的驱动开发人员。总之，硬件的内容很多很杂，硬件那方面练成了都会成为一个高手，我时常会给人家做下方案评估，很多高级硬件工程师设计的东西，经常被我一句话否定，因此工程师做到我这种地步，也会得罪些人，但硬件的确会有很多不为人知的东西，让很多高级硬件工程师也摸不到头脑。

那么高级硬件工程师技术技能都要具备那些东西哪，首先要掌握 EDA 设计的辅助工具类如 Protel\ORCAD\PowerPCB\Maplux2\ISE、VDHL 语言，要能用到这些工具画图画板做逻辑设计，再有就是接口设计审图能力，再者就是调试能力，如果能走到总体方案设计这块，那就基本上快成为资深工程师了。

硬件是要靠经验，也要靠积累的，十年磨一剑，百年磨一针。

把一个月前想写的东西，今天终于用一上午的进间整理完了，希望对喜爱嵌入式系统开发的工程师和学生们有所帮助