

搞嵌入式10年，悟出这些道理

工程师进阶笔记 2023-03-14 17:50 发表于广东

我是老温，一名热爱学习的嵌入式工程师
关注我，一起变得更加优秀！

工程师老温

43

嵌入式领域需要什么样的技术人才？

仔细观察各种招聘的岗位要求吧，无非是两方面。

1 – 通用要求

比如什么学历，多少年工作经验，有创新精神，抗压能力强，吃苦耐劳，善于沟通，态度认真等。

2 – 专业要求

这个可真的多了去了，但无非就是各种精通。

什么精通控制理论，精通电路设计，精通各种硬件驱动开发，精通linux内核以及裁剪，

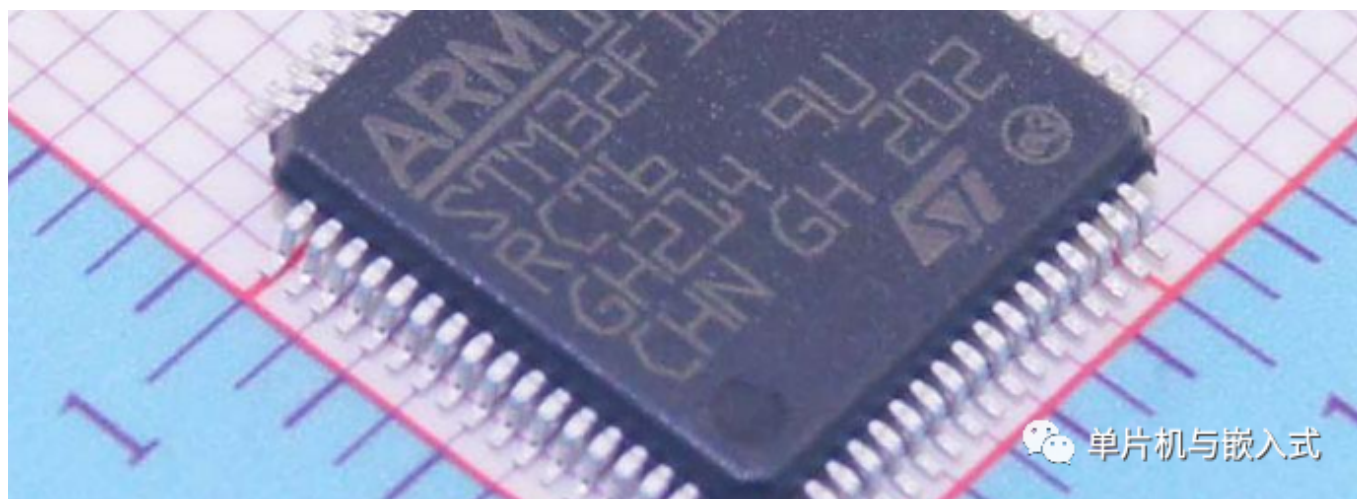
精通C/C++/Java语言，精通DSP，精通ARM，精通什么linux/ecos/ucos/wince/VxWorks，

精通嵌入式数据库，精通多线程编程，精通各种网络协议，精通各种总线协议，精通塞班/安卓/IOS平台应用开发，

精通汇编、编译器、二进制工具，精通各种媒体封装格式、编解码，精通通信技术，

精通硬件原理图，精通PCB设计，精通计算机结构，精通bootloader，精通.....

感觉只要是行业内的就要精通。其实，感觉精通一门就得好久...



看完这些，尤其是专业要求，头大了吧？

那么，我们就分析一下上面的要求吧。

各个公司就的通用要求像是互相复制粘贴的一样，比如学历(个人认为学历明显更重要)，

比如经验(是有n年的经验，还有n年前的经验用了n年)；

还有个人秉性，我都不知道各个公司怎么在员工入职前考察。专业要求呢(其实精通其中几样就挺了不起了)，

一看就让人头晕，了解了嵌入式开发的结构可能就好理解些了。

从技术实现上讲，嵌入式的产品分为两大类：

一类简单的，没有操作系统支持的;一类复杂的，有操作系统的。

就目前发展方向看，后一种是趋势。

前一种从程序实现上可分为3层：硬件层、驱动层、应用层；后一种分为4层：硬件层、驱动层、系统层、应用层。

嵌入式的开发大致需要这么从下到上四类工程师：

- 嵌入式硬件开发
- 嵌入式驱动开发
- 嵌入式系统开发
- 嵌入式软件开发

应该会哪些技能？

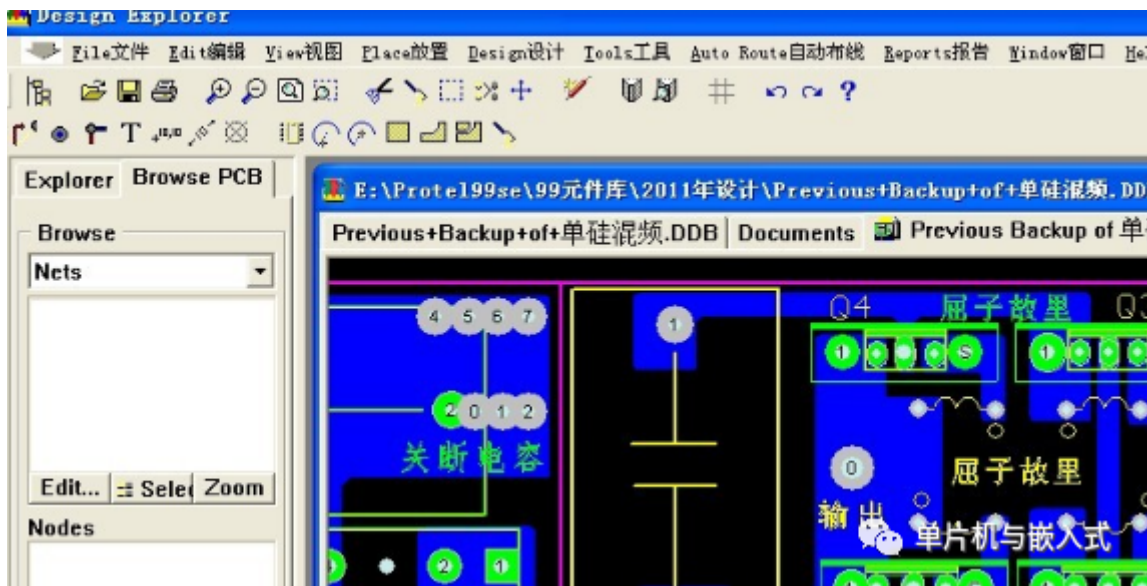
嵌入式硬件开发

这是硬件工程师来做，一般出身为自动化，电子，通信等专业，他们要求熟悉电路等知识，非常熟悉各种常用元器件，

掌握模拟电路和数字电路设计的开发能力，熟悉硬件开发模式和设计模式，熟悉各种芯片及外围设备，熟悉8位16位32位处理器嵌入式硬件平台开发。

有的要求有FPGA的开发经验，精通常用的硬件设计工具：Protel/PADS(PowerPCB)/Cadence/OrCad。

一般要有4~8层高速PCB设计经验。由于本人不是做硬件的，对这一块的不熟悉，所以不敢妄言。



嵌入式驱动开发

这个可以硬件工程师来做，也可以软件工程师来做，主要还是取决于设备。

目前驱动开发，除非是简单的设备驱动，实际上很多时候都是移植，所以更多的是需要掌握目标平台的硬件和系统特性。

至少得掌握一些操作系统、系统结构、计算机组成原理、数据结构相关知识，熟悉嵌入式ARM/MIPS/PowerPC架构。

具有单片机、某种或多种32位嵌入式处理器的移植开发能力，熟悉bootloader过程，

具有扎实的硬件知识，理解硬件原理图，能独立完成相关硬件驱动调试，能够根据芯片手册编写软件驱动程序。

如果涉及到网络设备，或许还要掌握很多高层的网络协议的原理。

个人感觉这一块虽然大部分不是很难，但却是极为重要的。当然如果要在这一层次提供对上层的多线程支持，某些驱动也是很难写的。

嵌入式系统开发

这个没听说硬件工程师来做的。做这块，熟悉操作系统的原理是必不可少的，比如内存管理，比如线程调度，

比如文件系统等等;还要精通处理器体系结构、指令集、寻址方式、调试、汇编和混合编程等方面的内容;熟悉操作系统启动流程。

对于linux，要熟悉Linux配置文件的修改，掌握内核裁减、内核移植、交叉编译、内核调试、启动程序Bootloader编写、根文件系统制作和集成部署Linux系统等整个流程。

这一块要是真写起来，困难程度那是相当恐怖的，好在大多都是移植，很少听说有公司要去开发个操作系统的，没Google的实力啊。

嵌入式软件开发

这个嘛，没有操作系统的话，程序一般不会特别复杂，如果有操作系统的支持，又和一般的软件开发很接近，所以尽可以看作是搞桌面开发。

这个要求就是熟悉系统的API，精通C语言的高级编程知识，包括函数与程序结构、指针、数组、常用算法、库函数的使用等知识、数据结构的基础内容;掌握面向对象编程的基本思想，以及C++语言的基础内容。

精通嵌入式Linux下的程序设计，精通嵌入式Linux开发环境，包括系统编程、文件I/O、多进程和多线程、网络编程、GUI图形界面编程、数据库;

熟悉常用的图形库的编程，如QT、GTK、miniGUI、fltk、nano-x等;掌握各种应用层的网络协议使用，以及各种音视频技术等。

通用技能

另外还有些以上四类都用得着的技能。

(1) 基础技能：搞嵌入式开发，还有一些必不可少的基础技能

1 - 开发平台的熟悉。目前的嵌入式开发大多数是在linux或者类linux下的环境下进行的，那么linux的使用应该熟悉吧，shell脚本得会写吧，配置文件得会改吧，GCC GDB得会用吧，Make工具得会吧？

2 - 比如项目管理工具的使用，都什么年代了，不会用CVS，也不会用SVN？请问，你还活在软件时代的原始社会么？

3 - 数据结构和算法。搞软件开发，不懂数据结构，不会算法优化，不会衡量复杂度，太不靠谱了吧？当然，这个不是要求啥算法都会，啥数据结构都记得住，最起码得知道有那么种东西，有什么作用，这样用到的时候好歹也能查啊。

4 - 编程语言。关于JAVA，我知道现在JAVA的异军突起，很是震动了编程语言界，但是在嵌入式开发下，不会C或者C++语言，实在说不过去。

当然现在很多手机应用完全用java写的，如果你只想搞嵌入式上层软件开发又不想深入下去，只用JAVA也未尝不可。

关于汇编，嗯，这个可能挺难，毕竟跟具体的硬件架构(主要是CPU)有关，但是真的很有意思，而且在某些时候，它是不可替代的。

(2) 特殊技能：对于嵌入式开发的某些部分必不可少，但对于有些(比如应用开发)可能没有多大用处

1 - 数字电路。这个主要是了解一些基本的电路知识，如门电路、逻辑电路、存储器、寄存器，为了更好的理解嵌入式芯片、存储器、外围设备等硬件的设计，但往往在软件开发中未必能用到。因为大部分硬件都提供了很完整的软件接口。

2 - 操作系统。这个说的是平台上选用的操作系统。嵌入式产品操作系统实在是太多了。但操作系统的内部设计原理都是殊途同归。

首先要去学习的是操作系统原理，去了解进程通信、内存管理、文件系统等几个重要的部分，因为这些在软件开发尤其是内核驱动开发时会用到。

接下来可以去熟悉至少一种主流的操作系统，Linux是一个很好的选择，去了解它的内核架构，设备驱动架构，如何去编译、裁剪、编写应用程序，以及如何使用在具体的嵌入式产品上。

3 - 计算机的组成/微机原理。主要是为了了解微型计算机的硬件工作原理，明白CPU、总线、I/O、存储器等一些关键结构是怎么工作的，虽然是很枯燥的知识，但是对熟悉外围设备硬件工作方式，及操作系统，软件架构的设计及优化相当关键。

(3) 扩展的能力：这些可能是要用到的

1 - 通信协议。对于非应用层的通信协议，我实在不知道是不是该把它放入操作系统的范畴，因为很多操作系统是没有网络部分的，但是在很多方面它又是由操作系统提供。

当然这些下层通信协议，貌似不太用关心，因为太多的开源协议栈了。应用层的网络协议可能就要关心了，比如http，ftp，rtsp，rtmp，mms等等。

2 - 音视频处理。比如文件封装、音视频的编解码、各种音视频的标准，还有流媒体什么的。Ffmpeg是一个很好的东西，建议学习一下。

3 - 信息安全。恩，这一块貌似现在越来越受到重视了。

4 - 数据库。唉，嵌入式设备能力越来越强啊！再搞搞，我看也不用交叉编译了。

有时和同事讨论起现在的行业需求，说到各个公司总是招不到人，程序员又总说找不到工作，自己就有很多感想，是公司要求高，还是程序员要求高？

有时候看到一些招聘的职位要求，总是打击到自己的信心，一条一条数下来，好像自己能做到的还真没多少，只好让自己再努力些再努力些。

当然有时也会觉得某些招聘不靠谱，看那一条条的要求，个人感觉要是全做到，普通人怎么也得努力个十年八年的，这样的大牛们，估计看不上给出的那薪资吧。

对于程序员来讲，做技术真的是很累很难的一件事情。网上搜一下，遍地的抱怨贴、情感贴，都充分反映了这个职业对身体和情感的考验。

如果你不畏艰难，想把技术做下去，那么努力吧，也许某天你会随便拿起一份招聘启事，鄙视一眼：“靠！就这要求，老子去都掉价！”，那么你就是真正的大牛了！

嵌入式工作分析

不少新人来到我们研发已近十几年。这十几年里，同一批人员，有的已经很快适应了环境，经过自己的努力和付出，能为研发工作独挡一面，成为某一方面的小专家；

有的尽管前期不太适应，但是通过指导和修正，业已慢慢进入了工作状态，相信经过一段时间，会逐步担当起某一方面的研发工作。

同一时期进来工作，为什么他们现在的能力和状态会产生如此大的差异呢？通过的相处和分析，我发现凡是进步快的同志有如下特点：

1 - 勤奋，为了解决一个技术问题经常会主动留下来加班。

2 - 钻研，为了解决一个技术问题会主动学习各方面的资料，勤做实验，用事实去验证；。

3 - 和领导和同事及时主动沟通，集思广益，争取各方面的资源，充分利用已有的经验，避免走弯路，加快研发进程。

4 - 有强烈的事业心和牺牲精神，不达目标绝不停息。

反之，那些成长慢或者还没有融入研发工作的同志则具有以下特点：

1 - 工作没有目标，精力分散，很多的时间花在网络上，偷偷的看视频。

2 - 工作推卸责任，把工作没有进展的原因推卸到工作难度太大，或者资源不够，而自己却又不主动的去争取资源。

3 - 没有长期为研发乃至个人的职业生涯有充分的打算，或者说来研发的目标比较低下只是为了混口饭吃；人无远虑，必有近忧，没有远大的理想，外界任何一个小的干扰都会让你三心二意，因而知识的学习没有系统性。

4 - 和同事和领导的沟通不主动，甚至避让，怕领导检查工作。

5 - 更关注生活而不是工作。

6 - 比较在意自己的付出。

两种一对照，问题显而易见：

研发工作是一项很具挑战性的工作，需要付出艰辛的劳动，需要有扎实的理论基础和动手能力；在开发任务面前头脑清醒，思维敏捷，头绪清晰，有条不紊；

遇到困难不会退，遇到挫折不气馁；需要有团队精神，需要有化解矛盾和困难的勇气，智慧和决心，要有相当强的心理素质；

因此人员的素质和心理状态是决定性的。上进的同志，稍微带一带，成长速度很快；不主动的同志，工作起来举步维艰，带起来很吃劲，心情也受影响。

为了提高研发效率，我们开发了管理软件，制定了各种规章制度。但是，软件和制度只能治标，不能治本，问题的根还是出在人上。

应该讲，目前研发部门的软硬件条件和和技术水平，可以让一批想在电子技术嵌入式领域有所作为的年轻人干出骄人成果的。

但是现实情况我们也不能过于乐观，研发部门还是存在一些问题的，尤其是人员素质问题。

为此我们在参加研发管理培训的时候也咨询过专业人士，他说了一句话：小草永远也长不成苍天大树。

汗水与成长

你现在是小树苗呢，还是草种子？

是树是草取决于你是否有强烈的向上成长的愿望，取决于你是否有千方百计吸取土壤养料的勤奋，取决于你对的人生价值的判别，

认为现有的土壤值得你扎扎实实定下根来，做一棵这片土壤上的大树。想成为大树的，我们一定创造条件让他(她)成长为大树；

如果注定是草种子，我们也无能为力，只有优胜劣汰。

笨鸟先飞早入林，对事物的认知是和事物对大脑的刺激频率和时间成正比的，

一个事情经常去做，时候经常去琢磨，就能悟出其中的道理，对事物认识深度是通过在实践中不断发现错误，不断修正才达到的。

如果人不勤奋，把过多的时间和注意力放在工作之外的地方，那就自己切断了实践这个重要环节，水平和经验永远得不到提升，能力的提升跟谈不上了，

接着就是职务，收入上的提升，说到底就是个人在社会上竞争力的提升。

不去提升你的核心竞争力，成天沉迷于网络还有其他事物，是否有点舍本逐末呢？

个人定位决定工作的态度，思维决定习惯，习惯决定行为，行为影响你的核心竞争力。

改变世界之前，先改变自己。

-END-

往期推荐：点击图片即可跳转阅读



嵌入式 C 语言史上最愚蠢的一个BUG



嵌入式代码很烂，能用即可？



数十种嵌入式 C 语言代码优化的经验和方法

我是老温，一名热爱学习的嵌入式工程师

关注我，一起变得更加优秀！



工程师进阶笔记

工程师老温，专注嵌入式物联网相关技术领域。

120篇原创内容

公众号

喜欢此内容的人还喜欢

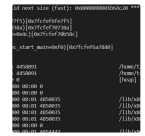
网络安全行业需要学习的基础知识有哪些？

徐大大seo



C语言中，通过malloc()函数申请的内存，free()函数释放内存，只传入一个地址，为什么能知道释放多大的内存？

Linux编程用C



6 个让你少写多做的 ES6 技巧

几何心凉的核心圈

