如何从零开始开发一款嵌入式产品（20 年的嵌入式经验）

首先，如果你有幸看到这篇文章，千万不要试图在 2 个小时内阅读完，就算你 2 个小时阅读完，我相信你也不会理解里面讲解的精华之处，我相信，你应该将此文章，慢慢品尝，这绝对是一篇需要品尝 2~3 天，再结合自己过往的经验，加上自己的思考，我相信会对你不仅仅是技术能力，甚至包括整体的思维方式都会有一个非常大的提高。

我写这篇文章的目的，是用本人 20 年的嵌入式经验呈现给大家一副完整的产品，项目开发蓝图，用本人多年经的历总结了一些教训无私的分享给各位，希望各位今后能站在本人的肩膀之上，少走弯路，多为公司，为个人多做贡献，那我的愿望就达到了，也同时希望能看到大家反馈和回复，留个脚印，留下你的见解和智慧，为后人乘凉打点基础，先在这谢谢各位了。

那么由此开始我们充满知识的旅程吧，最重要的一点，就是在一个产品或项目的开发过程中，如果没有明确的目标，那么成功将无从谈起，做任何事的第一步必须明确目标。

与日常生活中的大多数事务一样，设计一个嵌入式产品的过程也必须从确定目标开始，对生产的产品进行明确定义。对产品进行定义主要是对产品是什么和能有什么功能进行描述，其次是在我们的整个开发过程中，应该要撰写一些开发文档，大概的框架的如下：

1）产品需求文档：描述产品的特性
2）功能需求文档：描述产品必须具备的功能
3）工程说明文档：描述系统实现的方法和满足需求的手段
4）硬件说明文档：对有关硬件进行描述
5）软件或固件说明文档：描述特定处理器下设计微程序以及固件的方法
6）测试说明文档：描述必须测试的项目和验证系统正常运行的方法

1.需求定义

需求定义用来描述产品的基本功能，对于公司来说，需求一般由该公司的市场销售部门或该公司的主要客户来制定；而对小公司或爱好者（就像 armjishu.com 里的爱好者一样），技术人员可以自己负责定义需求，并撰写成文档。

通常需求定义是围绕以下几个因素而来：
1）系统的用途（定义需要系统实现的各种功能）
2）实际输入输出是何种方式实现的（为元器件的选型做参考）
3）系统是否需要操作界面（涉及软件层操作系统的选型）

其实对小型的嵌入式产品来说，定义需求是非常关键的，因为需求清楚了，就可

以避免后续开发过程中出现的诸如随机存储器（RAM）容量不足或所选的 CPU 速度不能满足处理的需要等一系列问题。

下面举个简单的实际例子，供大家来参考：
系统描述：用于从化温泉的水泵换水系统
电源输入：使用来自于变压器的 9V~12V 直流电
水泵功率：375W

1）使用单相交流电机，由机械电气进行控制
2）如果温泉池处于低水位，则输入开关闭合信号，以禁止水泵继续运行
3）用户可以自由设置水泵运行或关闭的时间长度
4）除了自动设置控制外，还需要提供一种人工装置来允许维护人员灵活控制水泵进行维修
5）水泵开启/关闭/人工干预的时间可以 30 分钟为单位，在 30 分钟到 23 小时的范围内进行调节
6）显示设备可以指示水泵的开关状态，剩余时间，以及水泵是否处于人工干预模式
7）具备监视低水位的功能，并显示在屏幕上

如果需要商用，那么除了上面给出的功能要求外，其设计文档中还要包括电磁干扰（EMI）和电磁兼容性（EMC）认证、安全认证以及使用环境（包括环境温度、湿度、盐雾腐蚀等）等方面的需求。

实际上，以上的需求确定之后，接下来就是要考虑选择一款合适的 CPU 来满足和实现系统的功能，那么我们就要将上述 7 点用户能够理解的需求转化成我们专业领域的需求，转化如下，大家可以参考一下：

a.处理或更新输入输出信号的速率究竟需要多快？
解释：目前嵌入式处理器的主频一般都在几十兆到几百兆不等，单片机的主频一般是几十兆，ARM 处理器可以到几百兆；我们主要看这个产品是否需要对大量数据进行处理，或是否需要对缓冲区进行频繁操作，是否有类似的占用 CPU 资料的工作要做，这就决定我们要选择一款合适的处理器来让该产品得到最佳的性能。

b.是否可使用单片集成电路（专用 IC）或 FPGA 来完成数据处理？
解释：如果可以的话，就不一定要选择处理器来做，用这些专业芯片就能替代

c.系统是否有大量的用户输入输出操作（如对开关和显示设备进行频繁操作）？
解释：如果有的话，要在处理器选型的时候考虑这些因素，选择一款能够满足以上要求的 CPU

d.系统与其他外部设备之间需要使用何种接口？
解释：这也是需要评估处理器的一个关键问题，选择具备这些接口功能的处理器会方便于我们的电路设计以及软件编程

e.设计完成后是否有可能需要进行改动，或在设计过程中系统需求是否可能出现

变化？我们的设计是否能适应系统需求的变化？
解释:要避免选择的处理器刚好满足当前要求，这样当以后事务要求逐渐提高，处理器性能如果还有一定空间的话，那么就可以重用目前的产品；第二个就是要选择不会即将停产的芯片，很多处理器用得很广乏，可以借鉴的资料也很多，但是很可能这款芯片已经在市场上流行很长时间了，芯片厂商已经推出更新换代的替代品了，如果你选择了这款芯片，很可能 1，2 年后就买不到这款处理器芯片了，导致不得不重新选择新的处理器，重新设计产品，这样的既耗费时间，金钱，更消耗人力，延误市场的战机。

2.处理器的选择

2.1.需要使用的 I/O 管脚数量

多数处理器都是使用内存和外部管脚来控制输入输出设备的，通常处理器都会有内置 ROM 和 RAM 的，如果内置的内存就已经满足需要，那么处理器就可以节省产生引用外部存储器信号的引脚，这样处理器可为输入输出提供较多的设备管脚（某些处理器支持外部 RAM 或 ROM 的使用，但对外部存储器进行访问时，处理器一般需要占用 8 条到 10 条 I/O 管脚）。

还有，有些处理器带有专用的内部定时时钟，这类时钟也需要使用一个端口管脚来实现某些定时功能；某些处理器中还具有漏极输出和高电流输出能力，可以方便的直接驱动继电器或电磁铁线圈，而不再需要额外驱动硬件的支持。

当对处理器 I/O 管脚进行计数时，我们一定要把使用处理器内部功能（如串行接口和定时器等）时限制使用的某些管脚考虑在内。
2.2.需要使用的接口数量

嵌入式处理器的主要功能是与应用环境中的硬件进行交互操作，这不仅需要外部硬件对接口具有实时处理能力，而且还要求处理器必须以足够快的速度对接口数据进行有效处理。

举例来说，AT91RM9200 是 ATMEL 公司出品的一款工业级 ARM9 微处理器，它基于 ARM920T 核心，处理速度可达 200MIPS，同时处理器内部配置了 USB、Ethernet 、支持 RS485 的红外串口、IIC、SPI、SSC 等输出接口，其目的是更方便的利用这些接口开发出嵌入式产品。

需要注意的是，由于许多处理器具有的局限性没有在处理器技术资料中给予足够的说明，因此一定要仔细阅读处理器的指标说明。例如，在阅读资料的过程中发现，该资料可能会说明其串行接口可以在最高波特率下工作，但仔细研究该处理器的指标数据时，可能会发现并非该串口接口的所有操作模式都可以在最大波特率下运行。

深入了解并明确接口要求的方法：可以自己动手编写一些程序来对接口进行实际测试，以确认某种处理器是否可以满足应用的要求；因为，确认某个处理器

是否可以满足接口要求并非是一件简单的任务。

2.3.需要使用的内存容量

决定内存容量的大小是嵌入式产品设计过程中的一个基本步骤，如果对所需内存容量估计过高，那么我们就有可能会选择成本较高的解决方案；反之，如果低估了所需内存容量，就有可能因系统需要重新设计而导致项目不能按时完工。

a.RAM 和 ROM 的区别：存储器分为随机存储器（RAM）和只读存储器（ROM）两种。其中 ROM 通常用来固化存储一些生产厂家写入的程序或数据，用于启动电脑和控制电脑的工作方式。而 RAM 则用来存取各种动态的输入输出数据、中间计算结果以及与外部存储器交换的数据和暂存数据。设备断电后，RAM 中存储的数据就会丢失。

b.随即存储器（RAM）的选择：RAM 容量的预测是比较直观的，我们只需把所有变量数目与所有内部缓冲区的容量以及先入先出（FIFO）队列长度和堆栈长度直接相加，就能得到所需 RAM 容量的总数。

如果所需内存容量超出这类处理器的寻址范围，那么只能通过增加外部 RAM 来满足需求；然而，增加外部 RAM 的同时将会占用一定数量的 I/O 管脚来对扩展内存进行寻址，这种扩展往往会影响到处理器来实现应用的初衷。

需要注意的一个问题是，某些微处理器限制 RAM 的使用，这种限制的目的是为了借用部分内存存储器作为内部寄存器组使用。除了以上因素外，所使用的开发语言也对所需 RAM 容量有一定的影响，某些效率较低的编译程序可能会占用大量宝贵的 RAM 空间。

c.只读存储器（ROM）的选择：系统所需 ROM 的大小应该是系统程序代码与所有基于 ROM 的数据表容量之和。预测所需 ROM 空间容量比较困难的部分是预测程序代码的长度，解决这类问题的方法只能是随着经验的逐步积累来提高预测精度。

然而，最重要的并不是精确计算程序的代码长度，而是要清楚地估算代码长度的上限。根据经验，如果 80%的 ROM 空间被代码占用的话，那么就太拥挤了，除非能确保系统需求不会有任何变化，否则至少要为可能发生的变化保留足够的备用 ROM 空间。

在多数情况下，我们可以试着在 ROM 中写入一部分程序代码，以便观察代码占用空间的情况，对于带有内部 ROM 的微处理器系统来说，系统程序都只能占用有限的程序存储器空间。

d.经验之谈：ROM 与 RAM 使用情况相类似，程序代码长度与所选用的开发语言有关。举例来说，使用汇编语言编制的程序要比使用 C 语言编制的程序占用少得多的空间。

对于追求低成本的小型系统来说，一般不提倡使用高级程序设计语言；这是因为虽然高级语言在使用、调试以及维护方面来的比较容易，但同时这类语言需要占用更多的内存空间和大量的处理器时钟周期。

如果开发语言选择不当，其后果可能是把一个简单、低成本的单片机系统

变为一个需要使用配置若干兆字节 RAM 空间的 64 位嵌入式处理器系统。

2.4.需要使用的中断数量

中断的主要用途是向中央处理器通报当前发生的某类特殊事件，这类事件包括诸如定时器超时事件、硬件引发的事件等。

需要强调的是，多数系统设计师经常过多地使用中断功能，实际上，中断的主要作用只是中断现行程序的执行，中断最适用于必须要求中央处理器立即提供服务的事件。

在需要设计和使用中断的情况下，一定要首先确认实际需要的中断数量，然后必须考虑到系统内部占用的中断资源，如果需要使用的中断资源超出了处理器可以接收的中断数量，我们就应借助于某些特殊手段来减少所需中断信号的数量。

2.5.实时处理方面的考虑

实时处理是一个涉及范围很广的题目，其主要内容与系统的处理速度有密切联系，实时事件是嵌入式微处理器需要关注的主要任务。

例如：处理器跟串口进行通信时，通常通过上层软件（为了保证实时性，进行任务切换的时间足够短），然后再占用处理器去执行从串口拿数据的任务，并且要保证处理器的速率比串口速率快，那么处理器可以以最快的速度反应并处理串口的相关的任务，这样就可以达到最大的实时性；

另一方面，如果处理器本身就内置了串口控制器、或 DMA、或 LCD 的控制器等，那么它就可以保证直接使用这些处理器内置的接口去控制串口、液晶屏等对象，以达到最大的实时性能。

2.6.该厂商是否提供好的开发工具和环境

选择一款新的处理器，很可能就要使用一个新的开发工具和开发环境，包括软件的编译环境等；对于开发日程安排比较紧张的项目来说，开发人员往往无法抽出专门的时间来研究，熟悉新的开发工具，从而也无法全面掌握开发工具的使用技巧。

并且，有的开发工具价格也比较昂贵，而且很可能只能从制造商那里购买，还有仿真工具也是需要付费的，这些对我们在选择一款处理器的时候，是都应该考虑进去的成本因素。

2.7.处理器速度方面的考虑

主要考虑几个细节问题：

1）处理器速度与处理器时钟之间的关系

例：单片机 8031 为例，由该处理器可以适应 12MHz 频率的输入时钟，因此就可以认为它是一个速度为 12MHz 的处理器了吗？不是，实际上，由于该处理器内部逻辑电路执行每条指令需要多种不同频率的时钟脉冲，因此该处理器内部时钟电路要对输入的 12MHz 时钟 12 分频处理；最终为处理器提供的只是 1MHz 主频。

有的时候，80MHz 主频的处理器（80MHz 输入时钟，80MHz 执行速度）要比 200MHz 主频的处理器（200MHz 输入时钟，50MHz 执行速度）执行速度要快得多。

2）处理器指令系统

如果不需要执行复杂数学运算的应用，那么 RISC 指令集的处理器要快；如果执行比较复杂的操作，则 CISC 指令集的处理器速度要更快。

3）芯片结构体系

现在有的芯片是将多个不同功能的核封装到一个芯片 IC 中，定制某种特定的功能，比如 DSP，其中包括用于实现数字解码、乘法运算的硬件乘法器和移相器等；然而，这类处理器也由其自身局限，往往在执行某些普通操作之前必须要使用额外的指令来把 RAM 中的数据放入内部寄存器，相比之下，一般处理器只允许对 RAM 中的数据进行直接访问。

2.8.只读存储器（ROM）的选择

多数工程项目在其开发阶段一般使用可擦写可编程只读存储器（EPROM）或快速存储器（Flash Memory）；这类可擦写可重复写入存储器的主要优点是可多次使用。一旦产品研制完毕，就可以用一次写入设备（OTP）来取代 EPROM 存储器，一次性写入器件的外观与封装几乎与 EPROM 完全一样，惟一不同之处就是其表面没有擦出窗口，并且价格要比 EPROM 低很多。

但是，另外一种情况，如果该产品今后需要升级固件，或在线编程，那么我们还是应该选择可擦写可编程的存储器。

还有一种是非易失的存储器，例如制造一台电视机，就有可能需要该设备具有记忆上次观看最后一个频道的功能，即使在切断电源后，该频道信息也不会丢失。

总结：所以，根据不同的产品选择不同的存储器也是一门很讲究的学问。

2.9.电源的要求

在某些设计中方案中，电源根本不存在问题，对电源唯一的要求就是可以为电路正常供电；实际上，选择电源主要要考虑三个方面的问题：

1）要注意设计方案中是否对电源的供电方式有所限制，例如,是否像大多数家用电器那样需要使用屋内墙上的电源插座供电，或是是使用 USB 接口供电

2）看系统是否需要使用电池供电方式，如果这样，我们就要考虑选择那种对驱动电流要求不高的处理器，然后再为其选择合适的电池。

3）休眠电流：许多微处理器都支持低功率运行模式，在这种模式下，系统的 CPU 处理器将处于休眠状态，同时所有外部设备的电源供电都被暂时切断，以便减少系统的电能消耗；某些微处理器在这种方式下需要的维持电流极小，但也有一些微处理器在这种方式下并不能节省多少功率；不管怎样，我们都要对系统在节点模式下的工作时间有一个估测，以便对具体情况选择使用的电池。

总之，无论哪种情况，我们都要对系统需要的供电总功率做到心中有数。

2.10.设备工作环境的要求

环境要求主要内容是考虑温度，湿度等；如果系统必须在温度范围较大的环境下运行，诸如用于军事设备或汽车的控制系统，那么处理器可选择的范围就要小得多；

并且由于大范围温度变化的设备通常比较昂贵，因此在设计过程中就不能再根据一般工业级器件的价格来制定预算。

2.11.使用周期成本

如果我们的产品是 mp3，在一般情况下，可以不必考虑在用户现场对 mp3 程序进行修改的问题，也不用为是否可以得到设备备件而着急，这是因为 mp3 是一种消费产品；

换句话说，如果我们的产品是价值几万块的工业设备并且需要常年不断地运行，那么我们在产品设计过程中就必须从长计议了：

a.首先，我们需要选择一种处理器或存储体系结构都可以升级的器件
b.考虑到程序升级的可能，我们还要选择较大容量的内存
c.最后要注意的则是所选处理器是否可以长期供货，这一点的重要性远远大于处理器的价格

除了上面的考虑之外，使用周期成本也是在设计之初要考虑的因素。总的来说，生产的部件越多，则可以接受的前期开发成本也就越大。如果产品是 mp3，我们可能会选择一个低价微处理器，同时投入一大笔钱来开发控制 mp3 的软件。

但如果我们的产品是价格昂贵的工业用设备，那么在产品的使用期内，该设备的销售量将只有几百台，毫无疑问，开发这种产品最重要的就是降低开发成本（降低开发成本而不是硬件成本!!!）；除此之外，工业产品的成本也不像家用电器或消费电子产品那么敏感。综上所述，开发工业产品当然要选择一种便于进行开发并且有助于缩短开发过程的处理器。

2.12.处理器相关资料是否丰富

如果该款处理器在市场上已经用得很广了，那么我们可以获取更多的相关资料，观察人家的产品是如何使用处理器的，也能在网络上找到不少的相关的设计资料以及相关技术主题，这样就进一步降低了技术门槛，确保了使用该处理器做产品可行性，减低了风险；

反之，如果是厂商全新推出的处理器，因为市场上还没有可以借鉴的产品，我们就只能从全英文的芯片手册开始阅读，了解这款芯片，这样开发周期不仅变长，而且不可预知的风险也很大。

3.开发成本的预测和估计

大多数项目或产品都有专人负责预测整个过程的开发成本，对于任何项目来说，其开发成本主要包括人力和材料开销。

预测开发成本在很大程度上需要根据经验，这也是为什么大型公司一般指定有经验的高级工程师来完成这一任务的原因，除了人力和材料的开销之外，总结下来，还有以下的开销：

1）人力成本（开发人员、管理人员、销售人员、其他行政等辅助人员）的开销

2）材料（硬件物料和损耗，有时候需要投几次 PCB 版才把产品稳定下来）的开销

3）开发系统和开发工具软件的开销

4）硬件工具的开销（例如示波器、仿真器等）

对于整个项目来说，上述的开销将直接可能导致产品成本增加，其中人力成本最为关键，尤其是在中国，呵呵

4.产品开发设计文档（需要包括硬件和软件两个方面）

4.1 硬件文档撰写思路

1）首先是需求定义或产品规格：

如果这些是产品最终目标的话，那么产品对硬件和软件的要求就是技术方案的最终目标；对硬件和软件的要求是从定义用户界面和系统功能开始的。

2）其次，根据需求，系统整体定义文档中给出硬件接口的具体定义：

定义硬件最有效的方法是从需求开始描述，由于硬件必须支持系统定义的所有功能，因此硬件定义是与系统说明不可分割的；

例如,我们设计一个定时器（事先需求说明定时器不能与个人电脑连接，故无法使用 CRT 显示时间），我们只有两种选择：一种是使用发光二极管（LED），另一种是使用液晶显示器件（LCD）；尽管 LCD 的显示效果比较好，但考虑到定时器要常年位于户外，并且早期 LCD 显示器不能在低温下工作，最终还是选择 LED 设备（这整个过程描述了我们硬件选型时的一个思路，这个是密切跟需求挂钩的）

3）一旦完成了系统整体说明文档，就开始进行系统设计：

首先要对硬件说明的内容进行细化，包括添加能让工程师理解的设计意图，以及软件工程师围绕硬件进行程序设计时需要使用的硬件信息等。

完成硬件电路板说明文档后，我们还要在该文档中增加一个用来描述系统的原始要求的前言部分，包括说明方案的设计思路和方法，除此之外，还要附上软件工程师用来对硬件进行控制所需的各类信息，这类信息主要包括如下内容（软件工程所需信息）：

-----内存和 I/O 端口地址（如果需要，还可以提供内存映射图）
-----可用内存容量
-----状态寄存器每一位的定义
-----每个端口管脚的用途
-----外部设备的驱动方法（例如，说明输入定时器电路的时钟频率等）
-----其他有管软件人员设计程序需要了解的信息

对于比较复杂的系统来说，硬件文档中经常使用两个独立的部分来进行说明；其第一部分用来描述硬件指标和工作原理，第二部分则主要为软件人员提供程序设计需要的信息。

4.2 软件文档撰写思路

1) 软件文档与硬件文档的组织方法类似，软件要求文档的主要内容则是定义软件要实现的功能；一种是在简单项目设计过程中，软件定义也可以只对一种电路板使用的软件给予描述；对较复杂的项目来说，由于参与这种项目的软件人员分别负责设计驱动不同硬件部分的代码（同一电路板），因此每个软件人员可能会为自己的设计代码指定不同的定义,这类软件说明需要提供下列的内容：

-----论述包括需求定义、工程指标、硬件参数等实施项目需要的内容
-----说明软件之间、处理器之间或处理器与其内部器件之间使用的通信协议：其内容应包括对缓冲区接口机制、命令/应答协议、信号控制等协议的具

体说明。

-----借助流程图、伪代码或者其他可能的方法来描述软件的实现方法和过程

2) 软件与硬件所考虑的不同之处（此经验方便技术总监或其他相关管理者参考，因为无论是多高深的技术管理者，要么是硬件出身，要么是软件出身，要么就是非技术出身）

a. 软件的灵活性远远大于硬件，要让软件人员搞清楚某个软件的内部格式是非常困难的任务，解决的办法：详细定义其他程序员需要了解的编程接口具体内容，以及其他工程人员在实施开发项目过程中需要使用的技术细节信息。

b. 软件工程师只有在收到硬件说明文档后，才有可能知道如何对系统硬件进行操作；而硬件人员一般不需要了解软件程序的技术细节。

c. 由于软件易于更改，因此程序内容经常会按销售人员提供的要求发生变更，在某些情况下，软件文档的内容无法及时反映程序的最新变化。

d. 软件经常是工程项目最后完成的部分，因此其文档也经常因时间不够而欠缺完整。实际上，软件文档是否详细、完整，在某种程度上是与公司或客户的要求有关的。例如，军事或国家工程一般要求开发商就其所有软件实现的功能提供全面详细的文档

e. 有个潜规则，对软件的要求越复杂，则需求的正确可能性就越小，这个是经验之谈了，我们需要把准需求这个准绳来做文章，而不是陷入个人主义以及对软件要求而凭空发挥自己不切实际的想象。

f. 我们可以先硬件设计，接着围绕该硬件编制软件。虽然实际系统的实现过程可能是软硬件并行开发，但软件人员基本上也是围绕着已经实现的硬件来进行程序设计的；对于更为复杂的系统来说，开发过程可能会出现重复。

例如，某个项目的硬件工程师和软件工程师可能会坐下来开会，共同决定使用哪种硬件来实现某种功能；软件人员可能提出需要为数据缓冲区口冲内存容量，也可能要求提供某种外部设备接口，以便充分利用现成接口程序提供的各种驱动代码。

总的来说，必须在提高软件开发效率与硬件系统的复杂性与成本之间进行权衡.

5.嵌入式高手对技术的理解（含辛茹苦这么多年的精华体验）

有很多人认为：嵌入式系统性能的核心因素是软件功能，其实，如果按照这种逻辑，系统设计中存在的问题就应由软件人员来负责；其实这个观点实际上反

映了设计嵌入式产品时如何考虑划分硬件和软件各自应实现的功能，也就是这个功能是软件实现，还是考虑用硬件来实现（硬件实现：需要购买处理该功能的硬件芯片，从而增加成本；软件实现：无需增加硬件成本，但会占用处理器以及内存的资源）。

例如：我们在这里设计的基于 ARM 的 mp3 嵌入式产品，我们可以使用专业的解码芯片来负责 mp3 音乐文件的解码和播放功能，也可以使用另一种方法来解码 mp3 语音文件，让 ARM 处理器利用软件控制寄存器来驱动耳机或音响，处理器通过对 mp3 语音文件解码之后再将解码后的数据流按照一定协议格式送给音频输出的硬件接口进行播放。

优点：这种方案在硬件方面节省了一个器件，降低了成本，并且该功能还方便调试（因为是软件实现的）。

缺点：从另一个角度来看，虽然节省了一块语音解码芯片，但同时要在三个方面增加成本。

首先，要在程序中增加语音协议解码的代码；
其次，可能要把增加 ROM 来存放语音解码的协议，这样可以增加速度；
最后，运行该程序将占用处理器的时间和资源。

其实，话又说回来，对于本案例来说，上述成本的节约并不会引发任何问题，包括驱动程序增加也只需少量的，我们讨论这个 mp3 产品的案例的目的在于说明如何对软件硬件的功能进行合理划分。

总的来说，交给软件实现的功能越多，则产品的成本就越低，当然这就要处理器必须有足够的处理速度和内存空间来实现设计指定的功能；常言说得好，天下没有免费的午餐；把功能分配给软件来实现，会增加软件的复杂性、开发时间、以及程序的调试时间；然而，随着处理器的处理能力的不断提高，可以预见，越来越多的功能将会由软件来实现。

虽然在软件中实现各种功能会增加开发成本，但如果把功能移植到硬件中实现，则会增加产品的成本，这类开销是在构造每个系统组件时不可避免的。在低成本设计方案中，增加任何额外的硬件都会对产品成本产生显著的影响，因此软硬件功能划分就是一个决定产品成本的大问题。在诸如大众消费产品这一类对成本非常敏感的设计方案中，一般都会把无法通过软件实现的功能排除在外的。

【全文完】