

编辑导语：硬件产品经理在面对产品研发的时候，除了执行沟通以及跟进流程以外，更是要对技术方面有着一定的了解；因为硬件行业的很多专业类内容，如果什么都不懂的话，在日常业务中可能也会有所阻碍；本文作者分享了关于硬件产品经理的技术知识，我们一起来了解一下。



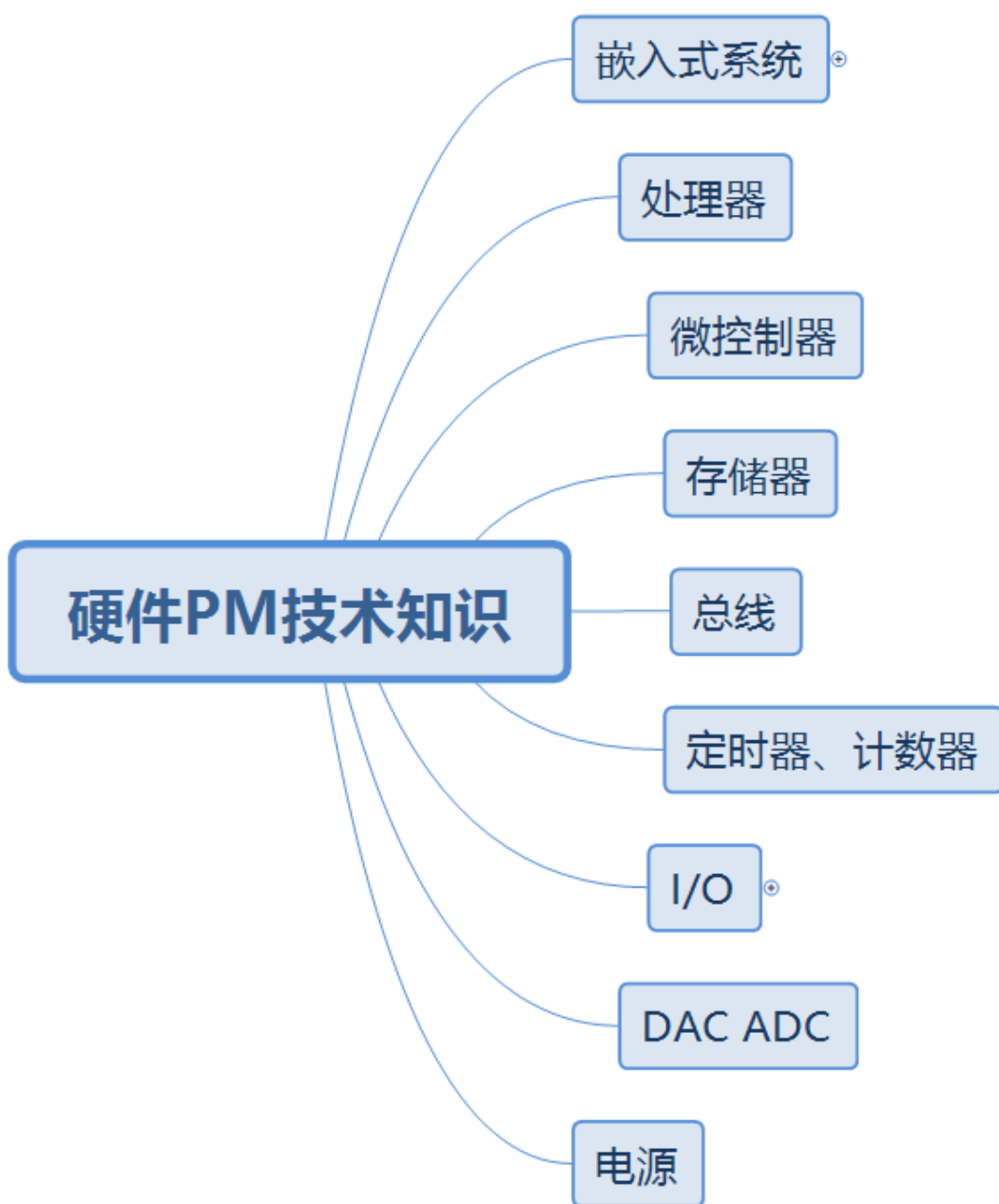
作为一个两年经验的硬件产品经理，经历了一款产品从零到上市的过程，积累了一点经验，我也想通过输出积累下来；想在硬件 PM 这一行长远的做下去，产品管理、市场能力以及成本控制这三方面的能力要不断精进。

我认为硬件 PM 懂技术是有必要的。作为初级产品经理，执行的工作，硬件基础知识是沟通的基础，和研发沟通需求，和供应商做后续的对接都离不开技术知识；进阶到中高级产品经理，市场以及供应链管理中技术是桥梁，与供应商，客户的沟通离不开技术知识。

硬件行业缩略语，定义词特别多。回想自己刚入行看到 RX TX IPC SPI Flash eMMC...一头雾水。看书，上网搜索硬件相关的内容，其内容的主要受众是工程师，而作为硬件 PM 需要对其归类并知道边界，对实现了解即可。

这也算是我发现的入行硬件 PM 的痛点吧！网上并没有完全针对硬件 PM 技术知识的内容。硬件 PM 涉及的面非常广，嵌入式硬件、软件、算法、结构、制造以及 ID 都需要了解。对于从软件 PM 转型或直接入行硬件 PM 的同学是一个门槛。

一、首先列一个嵌入式系统的知识框架

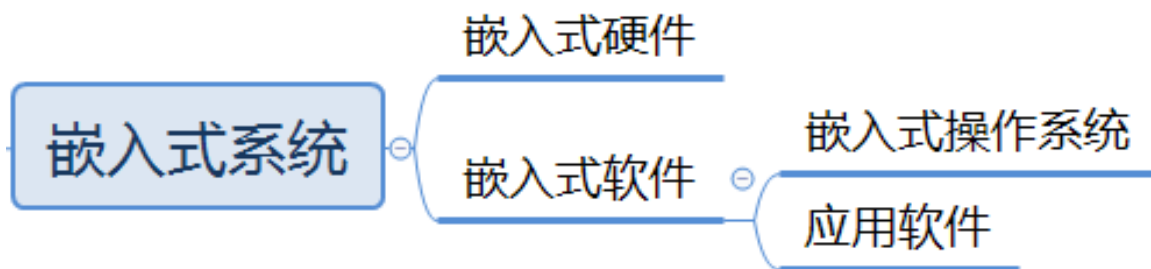


做一款硬件产品，硬件技术方面少不了这些内容。

二、嵌入式系统

嵌入式系统是专用于某个应用或者产品的系统。它可以自成一个系统、也可以是大系统的一部分。

嵌入式系统之间的关系如下图所示。



三、嵌入式硬件

嵌入式硬件包括处理器、存储器、微控制器、I/O、DAC ADC、电源、总线、定时器 计数器等。此外根据不同的应用场景，嵌入式硬件会集成不同的传感器，例如摄像头、雷达、GPS 等等。

本文主要介绍嵌入式系统，其它分内容后面会详细介绍。

四、嵌入式软件

嵌入式软件是嵌入式系统的核心。产品设计的落地，性能指标的满足都是通过软件来实现的，作为硬件 PM 我们需要了解嵌入式软件的架构，更好的与工程师沟通。

我们是硬件 PM，但是在研发工作中打交道最多的是软件工程师。作为 PM 我们要知道工程师的分工。提需求的时候要找到对应的人，拉通协作时找到对应的人。

嵌入式系统可以分为硬件层，驱动层、操作系统层、应用接口层、应用层。除了硬件层，其他都属于软件工程师的工作范围。在大多数公司中，软件工程师会单独负责其中的一块。

硬件接口层：这一层提供了与嵌入式硬件交互的接口，以设备驱动程序为主。例如网络、USB、LCD、显示屏、RX、TX、键盘等等硬件的驱动；驱动层的内容也会与应用层有交

互，例如 WiFi 账号密码的设置，RX 相关显示参数的设置；核心层：操作系统的内核，包括系统运行最基本的模块，这一层会涉及到系统时钟、电源管理、文件管理，内存管理等内容；系统层：它提供面向对象的系统资源管理功能，具有模块化和对象化的特点，例如文件管理、设备管理、网络协议。每个系统模块提供标准的 API 接口，向用户开放，有利于系统功能的扩展；应用服务层：这一层是在实现具体的业务功能提供基于系统功能、面向应用的系统功能调用服务接口。

嵌入式软件包括嵌入式操作系统以及应用程序

1. 嵌入式操作系统

说到嵌入式操作系统之前，先介绍一下操作系统。操作系统是硬件与用户之间的中介，用来管理系统资源和控制应用程序运行的计算机程序。

1) 嵌入式系统的特点

相对于操作系统，嵌入式操作系统不要求全能，但必须能够有效率的发挥硬件最大的能力，使产品获得最高的性价比，嵌入式操作系统的特点如下：

代码量小，能够在嵌入式系统有效的存储空间内运行；统一的 API；模块间解耦，可拆卸；强实时性，可应用于各种实时控制；可靠性高，无需人工干预独立运行，并能及时处理各类事件和故障。

2) 内核

嵌入式系统分为宏内核与微内核：

典型的微内核有 μ COS、Uitron 这些内核小很多，只有最基本的进程管理、进程通信、内存管理组成，其他部分和内核分开。内核本身只负责客户进程和服务进程之间的消息传递；典型的宏内核有 Linux、Unix，这些内核相对比较庞大，内核设计者从效率的角度把基本的操作系统内核和设备驱动、文件系统、网络等统一安排编译，它的移植能力较弱。

3) 操作系统基本组件

无论是宏内核还是微内核都有进程管理、内存管理、中断管理以及 API。

进程管理：操作系统为进程分配资源，实现进程间共享和交换信息，保护进程资源以及实现进程间同步。在多线程系统中，进程变成了资源管理器，线程成为了程序的基本执行单元。进程管理包括进程的创建与维护、进程的调度切换、多处理器的进程管理；内存管理：虚存管理技术广泛应用于系统管理。它可以使每个程序都有一个独立的虚拟存储空间，使应用程序获得比实际内存更大的编址空间；中断管理：中断管理为了提高系统效率，确保关键的功能及时执行；API：包括人机界面以及操作系统提供给用户开发的应用程序 API 函数。

4) 嵌入式实时操作系统 (RTOS)

在介绍之前先介绍个概念：任务调度方式。

5) 任务调度

在嵌入式操作系统中，任务调度方式有三种：不可抢占式调度、可抢占调度、时间片轮转调度

不可抢占式调度：一个任务一旦或者处理器资源就独占处理器运行，直至该任务结束进入等待状态；可抢占调度：基于任务优先级，当前任务必须随时让位于优先级更高且处于就绪状态的任务；时间片轮转调度：当两个或两个以上任务有相同优先级时，某个任务运行事先规定的时间，然后切换时另外的任务。

嵌入式操作系统分为嵌入式操作系统和嵌入式实时操作系统 (RTOS)，两者的主要区别是任务调度方式不同。

嵌入式实时操作系统的任务调度方法基于可抢占调度法，它真的具有实时性。

6) 实时性

实时也就是操作系统的计算结果不仅依赖于计算的逻辑结果，也依赖于结果产生的时间，RTOS 不仅要求产生正确的结果，还要求结果产生的足够快。“快”是一个相对的概念，性能很强大的操作系统不一定是实时系统。

而实时性又分为硬实时和软实时：

硬实时：必须在恰当的时间里完成相应的任务；软实时：要求系统以最快的速度执行，运行时间超时是可以容忍的。

7) 嵌入式实时操作系统

RTOS 是一段嵌入在目标代码中的程序，系统复位后首先执行，相当于用户的主程序，用户的其他程序都建立在 RTOS 之上；此外 RTOS 是一个标准的内核，CPU 将时间、中断、IO、定时器等资源都包装起来，留给用户一个标准 API，并根据各个任务的优先级，合理的在不同任务之间分配 CPU 时间。

此外 RTOS 最关键的部分就是实时多任务内核，包括任务管理、消息管理、存储管理、事件管理；这些管理功能是通过 RTOS 的 API 给用户调用的，跨平台移植程序修改的很少。

8) 嵌入式实时操作系统特点

微内核设计；系统能够快速进行任务切换、保证任务抢先和 deadline 的完成；尽量减少禁止中断的时间；提供固定或者可变的内存管理机制；提供一个能够满足应用要求的实时时钟；必须提供合适的进程任务调度方法；允许应用程序自己修改内核。

9) 常见的 RTOS

VxWorks； μ C/os-II；eCOS。

题图来自 Unsplash，基于 CC0 协议。