

整理：<http://blog.ednchina.com/alex1120>

为什么要使用仿真器

1.1 仿真的概念：

--- 仿真的概念其实使用非常广，最终的含义就是使用可控的手段来模仿真实的情况。

--- 在嵌入式系统的设计中，仿真应用的范围主要集中在对程序的仿真上。例如，在单片机的开发过程中，程序的设计是最为重要的但也是难度最大的一种最简单和原始的开发流程是：编写程序烧写芯片验证功能，这种方法对于简单的小系统是可以对付的，但在大系统中使用这种方法则是完全不可能的。

1.2 仿真的种类：

-- 软件仿真：这种方法主要是使用计算机软件来模拟运行，实际的单片机运行因此仿真与硬件无关的系统具有一定的优点。用户不需要搭建硬件电路就可以对程序进行验证，特别适合于偏重算法的程序。软件仿真的缺点是无法完全仿真与硬件相关的部分，因此最终还要通过硬件仿真来完成最终的设计。

-- 硬件仿真：使用附加的硬件来替代用户系统的单片机并完成单片机全部或大部分的功能。使用了附加硬件后用户就可以对程序的运行进行控制，例如单步，全速，查看资源断点等。硬件仿真是开发过程中所必须的。

1.3 为什么要使用仿真器？

-- 在与一些有经验的工程师交谈中，我们会发现有相当一部分工程师在开发中不使用或很少仿真器。向他们询问，原因得到的回答是“仿真器不可靠”但是他们是如何解决程序开发中遇到的问题呢？通过深入的交流才明白他们是按照这样的方法来开发程序的：

--- (1) 根据自己的设计建立一个符合要求的硬件平台，如果该平台涉及的程序比较复杂。还要搭建一个人机交流的通道人机交流通道可能是一个简单的发光二极管，管蜂鸣器，复杂的可能是串口通讯口 LCD显示屏。

--- (2) 写一个最简单的程序例，如只是将发光二极管连续的闪烁。程序编译后烧写到单片机芯片中，验证硬件平台是否工作正常。

--- (3) 硬件平台正常工作后编写系统最低层的驱动程序，每次程序更改后都重新烧写单片机芯片验证。如果在程序验证中遇到问题，则可能在程序中加入一些调试手段。例如通过串口发送一些信息到 PC 端的超级终端上，用于了解程序的运行情况。

--- (4) 系统低层驱动程序完成后再编写用户框架程序，由于这部分已经不涉及到硬件部分，所以程序中的问题用户一般能够发现。

-- 但是更多的调查表明，使用以上方法的工程师总的看来所设计的程序不是很庞大或很复杂。因为在做简单的项目时，我们可以通过一个发光二极管就可以表达出内部的信息：如果程序复杂，可能需要更多的信息来表示内部的状态，这样

可能就需要串口协助调试：如果程序更复杂，硬件更多，实时性更强，那工程师就要更多的增强调试手段，串口可能就不能满足了，需要类似于断点的功能，因为我想知道在某一个时刻单片机内部的状态究竟是怎样？

-- 如果用户程序的修改非常频繁，可能一次又一次地的烧写芯片占用的时间就很多，这时用户就会想能下载程序并运行的装置。

-- 到这里，您会看到，随着用户要求的越来越高，调试装置已经越来越象一个通用的仿真器了。因此我们的建议是：不要回避使用仿真器因为使用仿真器能提高您的开发速度。

-- 但是不能否认的是，用户回避使用仿真器也是有原因的。因为仿真器也是一种电子装置，非常依赖于设计者的水平。如果一个仿真器设计者的水平有欠缺，那将给仿真器的使用者带来很大的问题。因为仿真器的使用者将分不清楚究竟是程序的问题还是出在那里。随着电子设备的复杂化，设计工程师面临前所未有的压力。您可以想象，用户发现了程序中有一个问题，首先怀疑是自己系统中的问题，可能是软件方面也可能是硬件方面。因为用户系统处于开发阶段，用户基本上不会怀疑仿真器。在这种情况下，用户将耗费很大的精力在自己的系统中寻找并不存在的问题。如果用户最终发现问题来源于仿真器，并通过烧写芯片验证确实如此，那这个仿真器用户以后可能会逐渐放弃使用仿真器。

-- 用户放弃使用仿真器，对用户的影响是巨大的。因为放弃使用一个设计不完善的仿真器，也放弃了使用其它设计完善的仿真器，关键是放弃了合理的开发方法。因此我们的建议是：不要回避使用仿真器但要挑选好的仿真器。

1.4 仿真器的附加服务更加重要

-- 在购买仿真器的同时，还要注意仿真器厂家的服务，这一点很关键。这里说的服务包含仿真器本身的服务，例如仿真器使用上的指导服务，仿真器的维修服务，这些都是一个产品的最基本的服务。对于仿真器这种特殊产品来将还不是全部的服务。

-- 使用仿真器来开发产品，会遇到很多问题：产品本身的问题，仿真器的问题，仿真器和用户系统的问题。如果用户开发经验不是很丰富，他并不能排除遇到的问题究竟来源于哪个方面。这时，他可能求助于仿真器生产厂商。如果仿真器的生产厂商只专注于仿真器的设计，而对于其它的单片机应用不熟悉，他可能对用户提出的问题不能作出正确的判断而一味的强调自己的仿真器没有问题。与之相反，一个既熟悉仿真器的使用，又有丰富开发经验的仿真器支持厂商则能协助用户快速的找出问题。实践证明这两种情况带来的效果是完全不一样的。