

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

感悟设计

—— 电子设计的经验与哲学

文章预览

从人体解剖理解 C++

我在第一次阅读 C++ 书籍后，第一次从书籍上看见面向对象的编程思想。当时还犯嘀咕，干嘛要设计这样一个编译器，普通的 C 不是也很好吗？不过嘀咕归嘀咕，我还一向信奉“存在的就是合理的”。所以我相信既然 Microsoft 和 Borland 等国际一流软件公司都有 C++ 的编译器卖（还有面向对象的别的语言的编译器，比如 JAVA、C#、Delphi），自然也会有其合理之处。

用 C++ 做了几个工程。随着对 C++ 待越来越熟练、对其理解也越来越深刻。同时也越来越觉得面向对象编程思想之伟大。我觉得 C++ 的很多概念和理解问题的方式，和我们自然界的東西是一致的。我就以人体解剖来做个比喻，诠释一下 C++ 吧！

1 Class 的可分解性和继承性

人体有很多器官，可以分成几个类别，比如消化的器官，泌尿的器官，血液循环的器官。为便于描述，我统称他们为零件，我们的皮肤，骨骼也可以算是一种零件。每种零件功能各异，形状大小也大相径庭。但即使是这样，它们之间仍然有许多共同点。

每个人体零件都是由一大堆细胞集合而成。而细胞要生存和新陈代谢，就必须有循环系统穿越其间。大的零件有大的循环管道，小的零件有小的循环管道。而很小很小的零件有微循环系统穿越其间。而每个零件的循环管道，总有一个血液的入口和血液的出口。血液的入口所携带的血液总是含氧量高一些，出口的血液含氧量总是小一些，而代谢的废物的含量也相对高一些。

好了，有了这些共同点，就相当于 C++ 里面的基本类 CObject。人体的大部分器官都基于这个基本类而派生。当然也有个别不包含毛细血管的零件，比如指甲，头发，单个的血细胞等。这个时候我们可以把这个认为是不基于 CObject 的类。

有了这个人体的 CObject 类后，就可以派生出很多不同的人体器官，比如从基本的 CObject 类派生出管道类器官，这种器官的显著特点就是一个入口，一个出口。入口和出口直接相通。比如肠子，胃，气管，大动脉血管等，也可以派生出非管道器官，比如肝脏，肾脏。这些器官的特点都是有一个进去的大血管，一个出去的大血管。还有一些别的对外接口，比如肾脏还有一个输出端口---输尿管。

基于这个管道类，还可以进一步派生出很多零件，比如肠子，胃，尿道，大血管等。比如胃，在管道类的基础上加上几个 private 的函数，比如破碎食物、分泌胃酸、储存食物、自发蠕动等功能。而这些 private 函数对我们而言并不重要，俺们只要根据胃的其中一个 public 输出：通知大脑的饱感和饥饿感，把食物输入就可以。而破碎食物、分泌胃酸、储存食物、自发蠕动等功能都被“胃”这个 Class 封装掉了。这就是 Class 的封装性。

当然还有一些 Class 不是管道类的，而是像肝脏那样的实心的，或者像是肺那样的满是泡泡的器官，这也同样也可以理解为是从 CObject 类派生而来。

而一系列的零件可以构成人体的一个系统，我们可以把它看作是一个更大的 Class。比如消化系统的 Class、呼吸系统的 Class、神经系统的 Class。而最后这一系列的系统，最后派生出一个完整的人体 Class。而不同的人体 Class，在各自的大脑里面安装上不同的操作系统，就有了不同型号的人，有的人适合做政客、有的人适合做商人、有的人适合做士兵、有的人适合做将军。。。。。

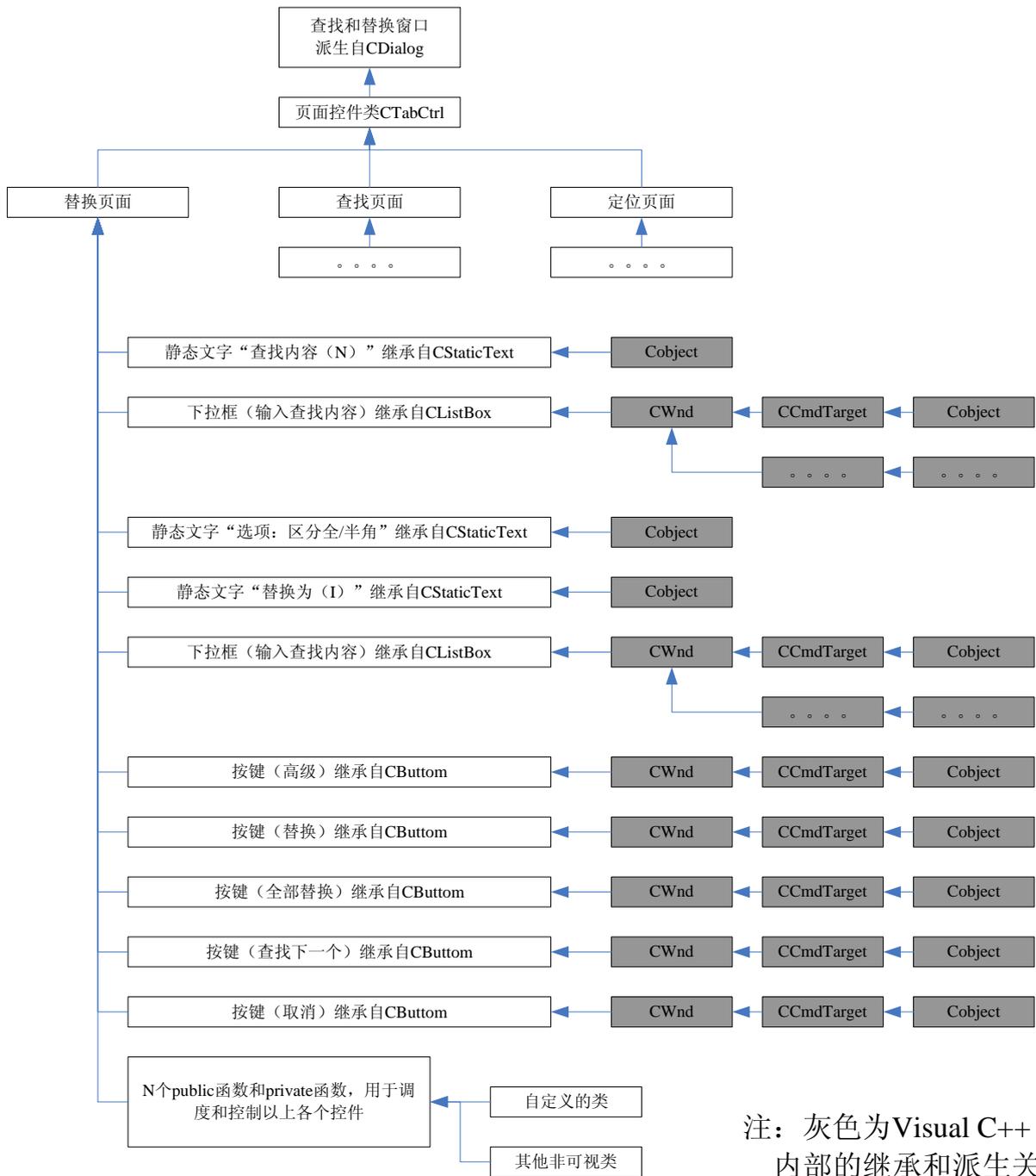
在 C++ 系统中，就是这样将一个个的 Class，通过不断的继承和派生，生成一个特定功能的应用程序。如果将其逐个分解，其组织形式应该是树形结构的。

比如我们买一个电视机，拆开包装箱，可以看见几个 Class：电视机、缓冲泡沫、遥控器、说明书、RF 电缆等。而每个 Class 都有可能是可分解的，比如遥控器，你分解它之后，可以看见几个更小的 Class：外壳、导电橡胶、电路板。电路板这个 Class 又可以进一步分解为：pcb 板基、电阻、电容等元器件、红外线发射管、遥控 IC 等。如果你要进一步分解遥控 IC 的话，还可以看见。。。。。。。。反正，无论是电视机那样的复杂系统，还是遥控器这样的简单系统，都是由一个个的微小的 Class，像前面所说的人体那样，一步一步不断的继承和派生得到的。



图 1

我们如果用 C++来写程序，也是这样一层一层地构建 Class。比如图 1 这个 Microsoft WORD 中常见的查找替换窗口，依照我的理解，如果用 Visual C++来写，其 Class 构建层次顺序是：



注：灰色为Visual C++ MFC 内部的继承和派生关系

图 2

如图 2，这个对话框内部的继承和派生关系，就像一个树。CObject 都在这个树的叶子节点上。而图 2 这个对话框构成的 C++ Class，同样也是整个 WORD 程序的叶子节点。整个 WORD 应用程序就是由成千上万个这样的类似的 Class 子树（当然，绝大多数子树是非可视的类）构成的庞大应用程序。

2 接口的精炼性和 Class 的封装性

人体每个器官的接口都可以是看作 C++里面的对外的 public 函数和属性。而你看看人体的每个器官的对外接口都很简单：

- 心脏只有 4 个血管作为对外接口
- 肺只有 2 根血管和一个气管作为对外接口。
- 肾脏只有 2 根血管和一个输尿管作为对外接口。
-

而各个器官通过这样简单的连接，将各个部件连接在一起，构成一个完整的人体。各个器官在这些简单的接口后面，都包含了内部的完整的工作流程和内部构件。这些流程和构件可以看作是私有的，而那些用于连接别的器官的管道，可以看作是一个 Class 中的 Public 类型的变量。这些私有的流程和构件可以大大简化外部管理这些器官的难度。因为从这些器官外面看来，只需要在其输入接口输入特定的东西，然后在其输出接口就能得到我们想要的结果。这就像一个被封装过的 IC 一样使用简单。

同样的，《软件工程》里面一再要求的，函数对外接口要尽量简单。只有简单的接口，才可以做到程序的可读性好、健壮性和易于使用和维护。

你拿一个超薄式的数码相机在手上，你就会发现，操作实在简单。什么都帮你搞好了，你要做的，只是打开电源、按下快门而已。数码相机的设计者就是要将用户的使用难度将到最低，所以它将很多的测光、对焦、文件存储等功能都全部封装进去了。如果你用自动档的话，那你能用到的接口就只有电源按键和快门按键。

数码相机为了做到傻瓜化，照相机必须自适应很多东西：要能够自动适应光线的强弱变化而自动调节光圈和闪光灯。当光线很强的时候，不开启闪光灯，同时光圈收小、快门也变小。当光线弱到一定程度时，曝光时间自动变长，如果曝光还是太弱，则光圈会自动开到最大，甚至会开启闪光灯。这样，照相机就可以适应很宽范围的照度了。它还要能够适应被拍摄者和照相机的不同距离而自动调节焦距。CPU 和软件根据测得的距离，自动决定镜头中几个镜片的相对位置而达到调焦的效果。而这一切都是作为数码相机内部封装的私有函数，在照相机内部自动处理完成。所以这样的照相机，才能够保持接口的简洁，让傻瓜都能够使用。

佩服大自然的伟大，人体的器官也是这样设计的：

- 消化系统的 Class，输入的是美食，输出的是不能利用的渣滓。有用的成分，都送到了循环系统。中间的过程，我可以不管。
- 流入肺部的血液是含氧量低的血液，流出的是含氧量高的血液。中间的过程，我可以不管。
- 肾脏也是这样，含有毒素的血液流入，出来就是比较干净的血液。而滤出的毒素就通过输尿管排泄掉了。中间的过程，我可以不管。
- 循环系统也有这样的封装性：当剧烈运动的时候，会自动调大心脏的输出功率（脉搏加快），同时还会通知呼吸系统自动加快呼吸的节律，以氧化更多的红细胞。而人体平静下来后，心脏也会自动减小输出功率，呼吸也趋于平静。这中间的过程，我也可以不管。

3 构造函数和析构函数

生物体和 C++ 一样，也有构建和析构，你相信吗？

人一生下来，主要就是在长身体。他（她）需要吸收来自大自然（至少也是间接来自大自然）的能量和营养物质，各个器官朝向成熟的方向成长。这个可以理解为是一种构建。这个类似于软件构建一个 Class，需要占用 CPU、消耗内存、打开文件等资源。

而人体的死亡后，或者通过人工的方式、或者通过天然的方式，若干时间后，人体的构成元素化整为零，全部化解为很容易被大自然重复利用的低分子量的有机物或者无机物。逐渐回到自然界，重新加入整个自然界的物质循环过程。若干时间后，可能一部分原来某人身体上的元素，被放置在了天空中的小鸟身上。或者混入涓涓的溪流，灌溉地上的小草。。。。。

而在 C++ 中，一个 Class 构建，首先需要分配一段内存。然后可能要打开一些资源，比如外设或者硬盘文件等。之后才可以正常工作。而 Class 的析构，则是和这个相反的一个过程，首先按照 Class 的继承层次，从叶子节点开始，逐个销毁子类。把每个 Class 子类占用的资源归还给应用程序或者操作系统。最后主 Class 也销毁，占用的资源同样也归还给应用程序或者操作系统。当应用程序的最后一个 Class 销毁的时候，就意味着整个进程被销毁。这个进程占用的所有资源也被操作系统回收。操作系统得到这些资源后，又可以启动下一

个相同的或者不同的进程。。。。。

4 Class 的健壮性

所谓健壮性，简单的说就是稳定性。在一个 Class 实现之后，不仅要完成其主要功能，还要能抵御各种异常条件和异常输入。

一个封闭的 Class，在其精炼的接口被定义好之后，就比较容易做好健壮性。因为精炼的接口背后的含义是数据处理权限的准确划分。哪些数据要在哪个接口里面处理，很容易理解。这样即使数据处理出了差错，也很容易找到哪个模块出了问题。这个就好像我们在工作中，各个员工的职责分明了，每个人都工作就不容易出差错。而如果责任不明确，出问题的时候就会很麻烦。

一个好的程序，除了需要能处理正常范围的输入数据以外，还能够处理一些比较离谱数据，至少要对这些离谱的数据给出错误的提示。比如一个开根的程序，正常的输入范围是大于 0 的数据，可是我输入一个负 1 的话，按道理这个时候开根无法运算的，但是这个时候程序就不能崩溃，而应该出错退出或者给出错误提示。

以上面提及的数码相机为例，为了保证产品的稳定性，数码相机的设计者通常还会关照：

- 外壳是否坚硬，是否抗冲击？
- 电池的放电过程有没有保护，不能够过放电，所以电池耗尽要自动关机。
- 存储卡和电缆插拔引入的 ESD 干扰脉冲是不是会损坏电路板？
- 电池是否有足够的续航时间？
-

这些东西相当于我们设计 C++ 程序的时候，所关心的“健壮性”。

同样的，人体也有很多体现健壮性的设计：

- 当循环系统输出功率已经最大、心脏已经超负荷的时候，会通过神经系统让人感知疲倦。如果人体在疲倦下仍然不停机，则会暂时强制人体进入休眠模式（昏厥），以保护人体的零件不超载。
- 肌肉不停的运动，会产生酸胀的感觉。促使人体减小运动的幅度或者停下来。
- 吃了带毒性的东西，会产生呕吐或者腹泻，以尽可能排出毒素。
- 人受伤后会产生疼痛的感觉，某些身体部位炎症也同样会产生疼痛的感觉。以便让人感知危险的存在并远离危险。



提示：

面向对象编程思想的伟大，就是因为它还原了大自然里面的事物的组织方式和层次结构。这样的编程方式，使得程序更易于管理、更有序和更稳定。

当当买家的评价

来自网站的截图

2009/7/19

“感悟设计电子设计的经验与哲理”的全部评论 - 当当网 - Windows Internet Explorer

http://www.dangdang.com/reviewList/20581915/

文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)

收藏夹 21IC社区 电子工程师的... 当感悟设计电子设计的经验... 感悟设计电子设计的... 搜索 按钮库 书签 翻译 标明所有搜索字词

对“感悟设计电子设计的经验与哲理”的9条评论

排序方式 按有用得票数多少显示 筛选 显示全部评论 第一个评论者: 新浪浮云

每页显示评论数: 20 40 80 第1页 >>

值得一读的书
- 个人评分: ★★★★★ 心情指数: 开心阅读场所: 书桌旁
发表于 2009-07-08 13:20
从21ic论坛上看到这本书的介绍,也曾常去远去的村庄(作者的博客)里面学习,关注这本书3个月,终于得到了一本,一睹为快!
工作了一年,做变频器的硬件开发,对我这种入门级的初学者来说,这本书让我明白了一些工作上遇到的没有想通的问题,融会贯通了一些了。还有不少的作者的不大清楚。可能限于篇幅的限制吧,某些问题讲得不很透。个人感觉有关电源部分,受益最大。

回复 | 举报

很不错的书
- 个人评分: ★★★★★ 心情指数: 开心阅读场所: 办公室
发表于 2009-07-16 09:08
看出来作者是很用心去写,统篇都是经验及设计相关的感悟,难得一见的好书。

回复 | 举报

不错
- 个人评分: ★★★★★ 心情指数: 受益匪浅阅读场所: 床上

很好,通俗易懂
- 个人评分: ★★★★★ 心情指数: 受益匪浅阅读场所: 办公室
发表于 2009-06-18 08:42
很好,通俗易懂,使人看过后觉得搞设计不是想象中那么神秘,是啊

回复 | 举报

感悟设计电子设计的经验与哲理
- 个人评分: ★★★★★ 心情指数: 受益匪浅阅读场所: 床上 书桌旁 沙发
发表于 2009-07-01 08:34
还不算的一本书

回复 | 举报

感悟设计电子设计的经验与哲理
- 个人评分: ★★★★★ 心情指数: 开心阅读场所: 床上 办公室
发表于 2009-06-04 18:58
很不错的书,通俗易懂,对于电子设计人员相当的有用途

回复 | 举报

不错的专业书籍 讲的很细 很透彻
- 个人评分: ★★★★★ 心情指数: 开心阅读场所: 办公室
发表于 2009-06-28 13:16
不错的专业书籍 讲的很细 很透彻

回复 | 举报

不错
- 个人评分: ★★★★★ 心情指数: 受益匪浅阅读场所: 床上
发表于 2009-06-15 11:10
书写得不错,总结的角度不错,设计经验有借鉴意义。

回复 | 举报

好
- 个人评分: ★★★★★ 心情指数: 受益匪浅阅读场所: 床上
发表于 2009-07-14 17:43
中国就缺这种类型的书,对实践很有帮助

回复 | 举报

不错
心情指数: 开心阅读场所: 床上
发表于 2009-07-10 19:16
下个月一定买本看看,看介绍就已经很精彩了。

回复 | 举报

您刚才阅读的文章出自：

《感悟设计——电子设计的经验与哲学》

以下请看看这本书的简介



有什么问题请发邮件到

herald_book@163.com

我们一起探讨。

100101101011101100101101011101

《感悟设计》

—— 电子设计的经验与哲学

本书的特点

本书主要从设计者的思维的角度来展开叙述。但和别的讲设计的书籍显著的不同的在于：本书不讲某个专题，而是注重描述设计的过程的思维方式和解决问题的方法。

所以，思想性、经验性和哲学性是本书的讨论重点。

- 通过这些重点，让你也能感受我思考问题的方法
- 通过这些重点，让你也能快速理解电子设计中的“只可意会，不可言传”的东西
- 通过这些重点，让你学会如何解决问题。
- 通过这些重点，再通过几个大的哲学道理的叙述，让你学会控制电子设计过程中的风险。
- 其间穿插了各种案例，不仅涵盖了硬件，也涵盖了软件。

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

我为什么要写这本书

在电子行业呆了十几年了。接触了很多东西，也处理过很多问题。为此，在自己硬盘里面也存储的相当数量的设计笔记、维修笔记等。这其中包含了很多失败的教训和成功的总结。同时在我设计的过程中，也得到了不少的叫得出名字的工程师和素不相识的工程师的帮助。所以，把这些设计分类归纳整理并出版，也是对那些帮助过我的人的一种回报。同时也是对更广大的业内同仁的一种经验分享。

我经常去书店买书。对于书架上的电子类、IT类的书目比较熟悉。这些书中不乏一些讲述先进技术的书籍，也不乏一些教授某个专题的书籍。但是讲述设计者的感想和思维方式的书籍却极少。所以我萌生了写这本书的念头。希望这本书能给大家带来一缕清新的空气。

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

目录简介

文章题目	页数	类别
前言	2	
模拟，无处不在	10	思想
上工治未病	9	思想
让你的软件飞起来	5	思想
设计的中庸之道	14	思想
搞定故障	23	思想
无电路图维修的技巧	7	思想
设计中的概率论	12	思想
线缆的学问	18	经验
尽量为后续的工作多遗留一些信息	2	经验
搞定噪声	26	经验
搞定电源	20	经验
从人体解剖理解C++	5	思想
电子产品设计中的成本控制	4	思想
物以类聚，信号以群分	6	经验
选择元器件，也有窍窍	9	经验
32768晶体跑出古怪频率	2	经验
优秀设计的10大要点	4	思想
由数组非法操作想到的	2	思想
废手机改成充电器	4	经验
PCB软件不为人知的技巧Net Class	6	经验
电子元件故障发生概率排行榜	5	经验

思想类，是彻底的设计感悟。是设计经验的深度归纳和总结。

经验类，是可以简单套用的实践经验

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

让你也能感受我思考问题的方法

爱因斯坦说：“思维方式比知识更重要”。这句话我不知道您能否理解。反正我是深刻理解的。所以，我在这个书中也有几个介绍设计的思维方式的文章：

- 《模拟，无处不在》
- 《从人体解剖理解C++》
- 《设计中的概率论》
- 《上工治未病》
- 《无电路图维修的技巧》

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

文章介绍：《设计中的概率论》

做设计每天都和自然规律博弈，时刻充满了风险和不确定性。而《概率论》就是由博弈派生出来的数学学科。所以理工科的人大多都学过《概率论》，这篇文章通过概率论的知识，让大家：

- 计算好概率，远离设计危险！
- 计算好概率，用最少的资源消耗达到目的！
- 教你根据经验和一些前兆事件，大致估算危险来临的时候和危险的级别。

这篇文章虽然是用数学原理来说明问题，但通篇没有一个数学公式，相信您能很容易的理解。

100101101011101100101101011101

《感悟设计》

—— 电子设计的经验与哲学

文章介绍: 《从人体解剖理解C++》

我在第一次阅读C++书籍后, 第一次从书籍上看见面向对象的编程思想。当时还犯嘀咕, 干嘛要设计这样一个编译器, 普通的C不是也很好吗?

用C++做了几个工程。随着对C++越来越熟练、对其理解也越来越深刻。同时也越来越觉得面向对象编程思想之伟大。我觉得C++的很多概念和理解问题的方式, 和我们自然界的東西是一致的。本文以人体解剖来做个形象比喻, 深刻诠释一下我所理解的C++。

100101101011101100101101011101

《感悟设计》

—— 电子设计的经验与哲学

让你也能快速理解电子设计中的“只可意会，不可言传”的东西

都说做设计难，很多东西只可意会，不可言传。许多设计书籍，比如教单片机的、教C语言的、教数据结构的、教画电路板的书籍，他们都是依靠一个单一的事情来展开讨论。翻开书本，花一点功夫去消化，将书本上的内容理解下来，一点都不困难。甚至可以临时抱佛脚强化记忆后去应付中国式的考试。可是把这些知识拿去做一个真实的设计，却会碰到很多稀奇古怪的问题。而这些问题通常不会在那些书上有任何的描述。这就是因为，做设计，除了知识，还要策略！

本书有几个文章，讲的就是策略，比如：

- 《搞定故障》
- 《搞定噪声》
- 《尽量为后续的工作多遗留一些信息》
- 《物以类聚，信号以群分》
- 《无电路图维修的技巧》
- 《设计中的概率论》

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

文章介绍：《搞定故障》

做研发，查找故障是家常便饭。这个查找故障可以是修理别人设计的成品整机的时候的查找故障，但多数是研发过程中的查找故障，检查一个硬件小模块的故障、检查一个电路板的故障、或者检查一段软件代码的故障。

查找故障，就其思维方式而言，和医生看病、警察探案没有本质的不同，差别只是各自的专业知识领域的差别。本文着重写三者共性的东西，即：查找故障的思维方式。

- 望、闻、问、切，收集故障信息
- 面对黑箱的对策
- 几种常见的处理故障的方法
 - 替换法
 - 逐层分割法
 - 搜索、穷举法
 - 算法
 - 数据统计法
- 找不着北的时候怎么办？

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

可以简单套用的经验

还有一些很容易套用的经验，相信各位读者会很容易得心应手。

《电子元件故障发生概率排行榜》

《让你的软件飞起来》

《PCB软件不为人知的技巧Net Class》

《搞定电源》

1001011010111011001011011101

《感悟设计》

—— 电子设计的经验与哲学

文章介绍:

《电子元件故障发生概率排行榜》

电子元件有很多种类，有十几个种类，各个种类有自己特定的故障发生概率。本文列出一个大大的表格，来说明各种元器件的脾气。

这个文章是依据我的经验和维修经验总结出来的，依据这个文章，你就可以快速了解哪些元件容易发生问题。从而在设计的时候绕开或者避免出问题。

100101101011101100101101011101

《感悟设计》

—— 电子设计的经验与哲学

文章介绍：《让你的软件飞起来》

我们做同样的事情，方法不一样，效果也不一样。于是就有了2个成语“事半功倍”和“事倍功半”。比如，汽车引擎，可以让你的速度轻松超越马车，但却无法飞行。因为汽车的引擎的功率都太小了。如果装上大功率的多缸活塞引擎，那飞起来没有问题，可是却不能超越音速；涡轮引擎，可以轻松超越音障，却无法飞出地球；如果有火箭发动机，就可以到达火星。这个就是方法决定效果的最好的例子。

这是一个真实的软件开发案例。通过不断地修改软件代码，将软件运行时间一步一步从120S压缩到0.5S。通过这个案例，你可以明白CPU的潜力远高于你的想像。本文虽然说的是软件，但道理和上面这个引擎的例子是一样的，不同的设计，就有不同的效果！



《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

文章介绍：《搞定电源》

一般人可能会觉得电源大概是电子设备里面比较容易搞定的门类了，因为，只要线路没有接错，电源都能出来。从这个方面说，好像是很容易。但是通过我多年的经验，发现电源其实是一个很麻烦的东西，稍微有一点不完美就会有大问题出现。所以专门拿一个大篇幅来讲讲“最简单”的电源。这个文章包含以下几个方面：

- 如何控制纹波
- 如何容忍纹波
- 搞定大功率电源
- 如何抑制电源的对外辐射

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

文章介绍：《模拟，无处不在》

我做了这些年的设计，慢慢有些感悟。其中一个感悟就是：“数字电路只是模拟电路的特殊形式，正如同正方形是矩形的特殊形式、圆形是椭圆的特殊形式一样。”或者更简单的说：“数字电路是模拟电路的一个子集。”

这篇文章就是教你如何理解“这个世界其实没有数字电路，所有的电路都是模拟的”这句话。

理解这句话，对于您设计好硬件，将有很大提高。



analog is everywhere™

100101101011101100101101011101

《感悟设计》

—— 电子设计的经验与哲学

本书的前言

从事电子类的工作十几年了，其中纯粹设计工作也超过了10年。每天都和软件和硬件打交道。总结这些年来，学到了一点知识、得到了一点经验、也感悟了一点东西。借着这本书希望和大家分享这些东西。

本书主要从设计者的思维的角度来展开叙述。但和别的讲设计的书籍显著的不同在于：本书不讲某个专题，比如单片机、C语言、DSP、数据结构、操作系统等，而是注重描述设计的整个过程的思维方式和解决问题的方法。

很多书都是以案例来教你如何掌握设计中的一个门类的技术。比如通过一段一段的C语言代码，通过语法、数据结构、流程控制语句来教会你编程。教你如何绘制一个铜箔、一个过孔、建立元件库来教会你如何绘制电路板。这些书一般不会写出作者设计后的感悟。而本书和别的书不同，本书中的很多文章都是我多年来感悟出来的一些道理，这些道理和我们生活中的一些哲理是一样的，所以本书的一些文章标题很多是一些哲学的道理，比如“上工治未病”、“模拟，无处不在”、“物以类聚，信号以群分”、“设计的中庸之道”等。我试图通过形形色色的案例来说明这些道理。在这些文章中，都是以哲学道理为框架，中间穿插了大量的设计中的案例和生活中的案例，用以说明这些哲学论点。

本书为了便于理解，讲述一些问题的时候，尽量避免像以前的书籍那样，罗列出一大堆的数学公式。因为我相信，理解繁杂的数学公式会很费脑子，阅读起来比较慢也比较困难。数学公式描述问题虽然严谨，却难免会少了几分作者的理解和感悟。所以我在表述这一类的问题的时候，尽量用我的经验和感悟来说明，尽最大力量压缩数学公式的使用。同时我还在书中大量使用了插图，因为插图比文字更容易理解。

书中的很多文章，是我的多年设计经验的总结。我把它们按照我自己的感悟写出来，然后按照类别分类在不同的文章里面。比如，关于电源的经验都在文章“搞定电源”里面，如何解决故障、消灭噪声的经验都在文章“搞定故障”、“搞定噪声”里面。还有一些很容易套用的经验，比如文章“电子元件故障发生概率排行榜”、“让你的软件飞起来”用起来相信各位读者会很容易得心应手。

本书作为一个讲述电子设计的书籍，却在其中大量穿插了很多非电子的案例和知识点。因为在我看来，无论什么学科的知识，都是用来解决问题的。既然是用来解决问题的，那就会有很多共性，因为即使是不同学科的问题，他们本身也是经常可以归纳出共同点的。所以把这些东西贯穿起来，更容易感悟出事物的内在规律。

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

本书的前言 (续前页)

都说做设计难，很多东西只可意会，不可言传。许多设计书籍，比如教单片机的、教C语言的、教数据结构的、教画电路板的书籍，他们都是依靠一个单一的事情来展开讨论。翻开书本，花一点功夫去消化，将书本上的内容理解下来，一点都不困难。甚至可以临时抱佛脚强化记忆后去应付中国式的考试。可是把这些知识拿去做一个真实的设计，却会碰到很多稀奇古怪的问题。而这些问题通常不会在那些书上有任何的描述。这个就好像是看了《三十六计》、《孙子兵法》就领兵打仗一样，虽然作战的套路写在纸上就是那么多，但是什么时候该用什么套路却是一种大智慧。作战的时候，有的时候适合做防守、而有的时候却要一鼓作气进攻。如果用错了策略，纵有雄兵百万也白搭。

和作战一样，做设计的时候也同样需要讲究策略。而在设计的时候，什么时候用什么策略，这种东西一般的书上不会说。因为这是一种基于实践而归纳的，游离于书本之外的知识。也就是我们常说的“只可意会，不可言传”的东西。而这些知识大多专业的成分少一些，而哲学和经验的成分多一些。本书就是致力于写出这些“只可意会，不可言传”的东西。当然了，因为这些东西本身是“不可言传”的，所以文字表达起来也比较困难，因为这十分考验我的文字能力。我努力写出了一些，还有一些，我一时半会还不知道怎么表达。所以，即使是已经写出来的东西应该还有不妥和错误的地方，请各位看官多加指点和多多海涵。

书中的一些经验和感悟，都是站在我自己的视角，按照我的思维方式和逻辑方式去理解的。所以这些内容仅代表我个人的看法。而这个世界的思想是多样化的，看待相同问题的角度也是多样的。所以本书的内容一定会有不对和值得商榷的地方，如有不对，烦请指点。我的E-mail: herald_book@163.com。我很愿意和各位在E-mail中探讨这些问题。

本书有幸由程序匠人作序，再次表示感谢。

最后，谨以此书献给我的家人。

《感悟设计》

—— 电子设计的经验与哲学

100101101011101100101101011101

谢谢各位
欢迎指正



有什么问题请发邮件到

herald_book@163.com

我们一起探讨。

100101101011101100101101011101

《感悟设计》

—— 电子设计的经验与哲学