

最新单片机的心得体会(精选5篇)

当在某些事情上我们有很深的体会时，就很有必要写一篇心得体会，通过写心得体会，可以帮助我们总结积累经验。心得体会可以帮助我们更好地认识自己，了解自己的优点和不足，从而不断提升自己。接下来我就给大家介绍一下如何才能写好一篇心得体会吧，我们一起来看一看吧。

单片机的心得体会篇一

本人还是学生，此刻大三，接触单片机一年多以来，从最基础的51开始，到更高级点的msp430一向到此刻刚开始上手的stm32编过的程序不算多，但加起来怎样也有上百个了，做过的东西仔细数数，能拿出手来算算的上是个样的也不下十个，算是小有所成吧。

但今日，我却发现，这一年我一向在走一条歪路。

先说一下情景吧，是这样的，最近同学几个合伙做个小项目，不算复杂，跟市面上的工程项目完全没法比，用难再需要使用的一个芯片是教师给的，从没用过，没有历程，至于一份电子版的数据手册，一切的一切都只能是从头开始，从底层开始，我们几个斌开始从最底层的逻辑时序、寄存器操作开始，但经过几天的研究，确实毫无效果可言。

为此，我们几个郁闷了好长时间，开始疯狂的上网搜资料，拷代码，却终究是不得而返，还让教师骂了一顿，因为这个项目是我们主动要求教师放给我们的，结果却耽误了工期。

我次我们几个人很受打击，最终决定大家一齐静下心来分析分析，到底我们败在了什么地方

最终总经过两天的思考、讨论、总结、检讨···，我们发

现了问题的所在，想来想去，这估计是很大一部分初学者正在面临的一个“瓶颈问题”，所以，由我净这个讨论的结果发布出来，期望与更多的人交流、分享、提高、共勉。

下头先总结一下我们失败的几点原因：

1、“都是开发板惹的祸”，能够说从开始学习单片机，哪怕是最基础的51、以致之后的msp430、arm我没从未离开过开发板，并且这长达一年多的时间，养成了我们对开发板极大的依靠，尽管自我焊过51最小系统、绘制过430最小系统板，但都是参考的别人已有的电路图，连最基本的电路基本都没分析过，为什么要在这几个电阻，该加个多大的，不加会怎样，太大了太小了会有什么后果，等等此类问题，我们从未分析过，全是copy、copy、再copy以致自我做过的东西不少，本事却未曾提高。

2、“全是例程误导我”，我们买来的开发板，往往都会带有丰富的例程，通常这也是卖家吹嘘、比拼的地方，但就是这些附带的历程，让我们与底层的开发绝缘了，有了例程，我们不用再一个代码一个代码地敲键盘，不用再去研究那些器件的数据手册、通信接口、寄存器、内部原理、以及很多商家没给我们加进去的功能，哪怕是将这些程序改到别的地方，我们也可是在宏定义里改两个引脚配置而已，从未尝试过从底层写驱动、敲代码、写函数，从未尝试过快平台移植代码，开发呢些没用过的功能等等。

3、“现实环境太无奈”，这跟上头两个说的是不一样的方面，我们都是大学生，大学里此刻吵得火热的“大学生就业难”、“你是考研还是就业”、“你报xxx培训班了吗”“你考四六级证了吗”等等，好像上大学就是为了干这些，似乎没有这些，你就会陷入“毕业即失业”的困境。应对这些外来的干扰，越来越多的学生坐不住了，我们这些人也一样，尽管一向在学习自我的爱好，但当被人问得多了的时候，心里也难免会发毛、会害怕，从而导致一年来真正静下心来学这些东

西的时间越来越少。

大体也就总结怎样多吧，经过这次失败，也算是教训深刻吧，此刻才大三，到大四毕业，还有近两年的时间，还不算太晚，再说将来的路还长，此刻发现这些，应当不算晚吧。

对我们来说，革命尚未成功，同志仍需努力啊！

开这贴，一为总结这次失败、而为自我立志改变、三为警示之后的同学、四为大家齐言献计。

大家有遇到同样问题的、要类似迷茫的、要好的提议意见的，欢迎回帖交流，大家一齐共勉。

单片机的心得体会篇二

篇一：单片机学习心得

一直想写一个帖子，将自己学习单片机的经历、经验和方法一下，今天终于动手写了，呵呵，这是我学习单片机的经验、心得、方法。希望对大家学习单片机有所帮助。

经常在群、论坛里看到有人问：怎么学单片机？也常看到有人说学了好几个月可就是没有什么进展。当然，受限于每个人受到的教育水平不同和个人理解能力的差异，学习起来会有快慢之分，但我感觉最重的就是学习方法。一个好的学习方法，能让你事半功倍，这里说说我学习单片机的经历和方法。

的时候，因为制作fm发射电路采用了bh1415的芯片，需要用到单片机控制，所以开始接触使用单片机，虽然以前读书时作为一门功课学过，不过也快丢荒了了，基本上都忘光了，也算从头学起了吧。我当时是买了一块实验板，很简单的功能，比论坛实验板还要少功能，还有一个仿真器，也是比

较简单的8051芯片的。然后开始从大家都知道的流水灯写起，万事开头难，第一个程序基本上就抄人家的，翻查了一份快速入门的学习资料，然后写下第一个程序。然后，在仿真机是单步的看运行结果，从每一步执行，都硬件上反应出来的效果，很快，对单片机的运作有了一个非常感性的认识，这跟只看书然后自己想象运行是完全两码事。然后，是进一步的其它程序学习，一个月时间，基本上已经熟悉了51的各种硬件资源。把流水灯，数码管动态扫描，中断等等的学习了，然后就是写综合性的程序，也就是我发在论坛里的时钟。通过这个程序的编写，使自己的编程水平有了很大的飞跃。之后，就是逐步学习编写更多的程序，各种外围器件的控制等等，通过编写程序使自己不断的提高。在学过汇编之后，我又转到了c语言学习，因为以前没有学习过c语言，一切都很陌生，所以自己沿用了学习汇编的方法，大约花了3天功夫基本就对c语言有了初步的了解，然后就是尝试写上面所说的时钟，以汇编程序的思路作为参考，以c语言来编写，花了几天功夫完成了这一程序，也实现了从汇编编程为主到c编程为主的转变。然后，就是以c语言编写外围器件的控制程序，以此来熟悉c的编程风格，一直到现在，基本上就是c语言编程，汇编已经是作为调试程序的辅助。

下面我概括了几点我的学习经验和：

- 1、万事开头难、要勇敢迈出第一步。开始的时候，不要老是给自己找借口，说keil不会建项目啦、没有实验板啦之类的。遇到困难要一件件攻克，不会建项目，就先学它，这方面网上教程很多，随便找找看一下，做几次就懂了。然后可以参考别的人程序，抄过来也无所谓，写一个最简单的，让它运行起来，先培养一下自己的感觉，知道写程序是怎么一回事，无论写大程序还是小程序，要做的工序不会差多少，总得建个项目，再配置一下项目，然后建个程序，加入项目中，再写代码、编译、生成hex文件刷进单片机中、运行。必须熟悉这一套工序。个人认为，一块学习板还是必要的，写好程序在

上面运行一下看结果，学习效果会好很多，仿真器就看个人需要了。单片机是注重理论和实践的，光看书不动手，是学不会的。

2、知识点用到才学，不用的暂时丢一边。厚厚的一本书，看着人头都晕了，学了后面的，前面的估计也快忘光了，所以，最好结合实际程序，用到的时候才去看，不必说非要把书从第一页看起，看完它才来写程序。比如你写流水灯，完全就没必要看中断的知识，专心把流水灯学好就是了，这是把整本书化整为零，一小点一小点的啃。

3、程序不要光看不写，一定要自己写一次。最开始的时候，啥都不懂，可以抄人家的程序过来，看看每一句是干什么用的，达到什么目的，运行后有什么后果，看明白了之后，就要自己写一次，你会发现，原来看明白别人的程序很容易，但到自己写的时候却一句也写不出来，这就是差距……当你自己能写出来的时候，说明你就真的懂了。

4、必须学会掌握调试程序的方法。不少人写程序，把代码写好了，然后一运行，不是自己想要的结果，就晕了，然后跑到论坛上发个帖子，把程序一贴，问：为什么我的程序不能正常运行？然后就等别人来给自己分析。这是一种很不好的行为，应该自己学会发现问题和学会如何解决问题。这就需要学习调试程序的方法，比如keil里，可以下断点啦，查看寄存器内容等等，这些都是调试程序的手段，当你发现你写的程序运行结果和你想象中不一样的时候，你可以单步，也可以下断点，然后跟踪，查看各相关寄存器内容，看看程序运行过程中是不是有什么偏差，找出影响结果的地方，改正过来。这一个过程非常重要，通过程序的排错，你可以学到的知识是书上得不到的。

5、找到解决问题思路比找到代码更重要。我们用单片机来控制周边器件，达到我们想到的目的，这是一个题目，而如何写出一个程序，来控制器件按你想要的结果去运作，这个就

是解题的思路。要写程序，就得先找到解决问题的思路，你学会找出这个解题思路，比你找到代码更为重要。不少人很喜欢找人家的代码，有的人甚至有了代码就直接复制到自己的程序中，可以说，这不是一种学习的态度，无助于你编程水平的提高。我几乎不怎么看人家的代码，多数时候是看别人的思路，有方框图最好，没有的话文字说明也可以，要从代码中看出别人处理问题的思路，是相当困难的，特别是大型的程序，看起来是非常的累人，所以现在我也明白了，以前读书时说的程序流程图很重要，现在算是知道了。当你知道一个问题怎么去解决了，那么剩下的只是你安排代码去完成，这就已经不是什么问题了。举个例子：数码管动态扫描，没写过的初学者可能搞不清是怎么回事，其实，就是分时让每一时间段时只控制一只数码管显示数字，几只数码管轮流显示，由于速度很快，人眼的看起来是全部数码管都亮的。明白是这么一回事，事情就好办了，剩下的事情，无非是你安排让一只只数码管轮流显示出相应的数值。显示数字，然后延时一下，再下一只显示数字，延时，知道是这样，我们实际程序上只要做到这样就可以：往段口送段码，然后打开位选显示一只，延时一下，再关闭位选，再送出段码，再打开另一只位选仅此而已。有了解决问题的思路，我们就能问题拆分开来，然后逐一的解决，如果动态扫描的原理都没懂，不知道如何做，那么这个程序是怎么也写不出来的。

6、开动脑筋，运用多种方法，不断优化自己的程序。想想用各种不同方法来实现同一功能。这是一个练习和提高的过程，一个问题，你解决了，那么你再想想，能不能换种写法，也可以实现同一功能，或者说，你写出来的代码，能不能再精简一点，让程序执行效率更高，这个过程，就是一个进步的过程。很多知识和经验的获得，并不是直接写在书让你看就可以得到的，需要自己去实践，开动脑筋，经验才能得到积累，编程水平才能有所提高。

7、看别人的代码，学习人家的思路。这个在学习初期是很有用，通过看别人的代码，特别是有多年编程经验的人写出的

具有一定水平的代码，可以使自己编程水平得到迅速的提高，同时，也可以结合别人的编程手法，与自己的想法融合在一起，写出更高水平的代码，从中得到进步。但要注意，切忌将学习变成抄袭，更不是抄袭完了就认为自己学会了，这样做只会使你退步。

8、尝试编写一下综合应用的程序。从流水灯学起，到动态扫描，再到中断，那么，你可以试试写一下时钟这种综合性应用的程序，不要小看时钟，要写好它不是一件容易的事情，它包括了单片机大部分的知识，比如有按键（io读取）、动态扫描（io输出）、中断等，如何协调好各功能模块正常工作，才是编程者需要学习的地方，当你单独写一个功能的时候，比如按键读取，你可能感觉很容易，因为你的程序啥也不做，只是读按键。但把它和其它功能混合在一起，如何在整个程序运行中使每一部分都正常工作，这就不是写一个按键读取这么容易的事情，功能模块之间有可能会互相影响，比如你需要让数码管既能显示，又要去处理按键读取，怎么使这两部分都正常工作，这就是一个协调过程。当你有了这个处理协调能力，你就算是入门了。

9、着重于培养解决问题的能力，而不是具体看自己编写了多少代码或者做过什么。“学单片机重点在于学习解决问题的思路，而不是局限于具体的芯片类型和语言”这一直是我的座右铭，是我学单片机多年来感悟出来的。经常看到有人说“你会驱动xx芯片，真牛啊”“你搞过xx项目，真厉害”之类的话，其实这是非常片面的，搞过xx芯片，搞过xx项目，只能说明你做过这一项目，它只是你的业绩，并不是代表能力就一定高。真正的能力应该是：“遇到没有解决的问题或器件，能利用自己已学的知识，迅速找到解决问题的方法。”这个才是能力。写程序的过程就是一个创造的过程，几乎没有完全一样的项目，每次你遇上的几乎都不相同，所以你拥有的必须是你面对新项目时的创造能力，而不是标榜着你以往做过多少项目。当然，业绩也能从另一侧面反映你的经

验和水平。

10、如果有可能，多学习计算机专业的知识，比如数据结构等。这些是你解决问题的基础知识，你把这些知识应用得越好，就会发现越容易找到解决问题的方法，这就是为什么一个学计算机专业的人编的程序和一个非计算机专业的人编的程序有差异的原因。也是一个菜鸟进军到高手所要配备的知识。如果我们把编程分为宏观编程和微观编程，那么微观编程就是写具体的代码，比如控制某某器件的语句；而宏观编程就是如何对整个程序进行布局、安排，使功能模块以你想要的方式去运行，得出你想要的结果。如前所说“会控制xx器件”这些只能算是微观编程，能做到这一步还只能算是菜鸟级别，如果面对一个新的器件，你心里没底，没把握去写这个控制程序，那说明你还是一个初级的菜鸟。当你有了一定的编程经验，控制过相当数量的器件之后，你就会发现，控制器件这些工作都是相似的、重复的工作，体现不出编程的水平，最多也是写得好与不好的差别，只能算是一些小技巧的应用。而对整个程序进行布局、安排这些才是最头痛的事情，能达到宏观编程和微观编程都做好才是真正的高手。对于规模越大的程序，越能体现出这一点。

11、面对一个新项目时，多自己开动脑筋，不要急于找别人的程序。有不少人面对一个新项目时，第一步想到的就是网上找别人写过的代码，然后抄一段，自己再写几句，凑在一起就完成任务，这虽然可能是省时间，但绝对不利你的学习。当你接到一个新项目时，应该先自己构思一下整个程序的架构，想想如何来完成，有可能的话，画一个流程图，简单的可以画在脑子里，对程序中用到的数据、变量有一个初步的安排，然后自己动手去写，遇到实在没办法解决的地方，再去请教别人，或看别人是怎么处理的，这样首先起码你自己动过脑想过，自己有自己的思路，如果你一开始就看别人的程序，你的思维就会受限在别人的思维里，自己想再创新就更难了，这样你自己永远也没办法提高，因为你是走在别人的影子里。

12、多利用网络的搜索，学会提问题。一般来说，学习过程中，你遇上问题，前人们多数也有遇上的，所以如果有什么不懂，在自己解决不了的时候，最好先到网上搜索一下，看能不能找到答案，找不到再到论坛里发问，发问也要有目的性，尽量简单明了的描述问题，让帮助你的人可以用最少的时间就看懂你说什么，毕竟人家帮助你是免费的，时间也是有限的。

以上就是我学习单片机的经历和心得体会，希望对大家会有帮助，说得有不好或不对的地方欢迎大家批评指正。

篇二：单片机学习心得

时光飞逝，一转眼，一个学期又进尾声了，本学期的单片机综合课程设计也在一周内完成了。

俗话说“好的开始是成功的一半”。说起课程设计，我认为最重要的就是做好设计的预习，认真的研究老师给的题目，选一个自己有兴趣的题目。其次，老师对实验的讲解要一丝不苟的去听去想，因为只有都明白了，做起设计就会事半功倍，如果没弄明白，就迷迷糊糊的去选题目做设计，到头来一点收获也没有。最后，要重视程序的模块化，修改的方便，也要注重程序的调试，掌握其方法。

虽然这次的课程设计算起来在实验室的时间只有三天，不过因为我们都有自己的实验板，所以在宿舍里做实验的时间一定不止三天。

硬件的设计跟焊接都要我们自己动手去焊，软件的编程也要我们不断的调试，最终一个能完成课程设计的劳动成果出来了，很高兴它能按着设计的思想与要求运动起来。

当然，这其中也有很多问题，第一、不够细心比如由于粗心大意焊错了线，由于对课本理论的不熟悉导致编程出现错误。

() 第二，是在学习态度上，这次课设是对我的学习态度的一次检验。对于这次单片机综合课程实习，我的第一大心得体会就是作为一名工程技术人员，要求具备的首要素质绝对应该是严谨。我们这次实习所遇到的多半问题多数都是由于我们不够严谨。第三，在做人上，我认识到，无论做什么事情，只要你足够坚强，有足够的毅力与决心，有足够的挑战困难的勇气，就没有什么办不到的。

在这次难得的课程设计过程中我锻炼了自己的思考能力和动手能力。通过题目选择和设计电路的过程中，加强了我思考问题的完整性和实际生活联系的可行性。在方案设计选择和芯片的选择上，培养了我们综合应用单片机的能力，对单片机的各个管脚的功能也有了进一步的认识。还锻炼我们个人的查阅技术资料的能力，动手能力，发现问题，解决问题的能力。并且我们熟练掌握了有关器件的性能及测试方法。

再次感谢老师的辅导以及同学的帮助，是他们让我有了一个更好的认识，无论是学习还是生活，生活是实在的，要踏实走路。课程设计时间虽然很短，但我学习了很多东西，使我眼界打开，感受颇深。

单片机的心得体会篇三

因为准备参加全国大学生电子设计大赛，所以自学了一下51单片机，把所有零零散散的时间加起来，学了有十天左右吧，此刻能够解决一些问题了，学了这么一段时间，觉得单纯的应用51单片机，还是很简单的，但是这要建立在必须的c语言编程潜力的基础之上的(我用的是c语言编程，不是汇编)，最重要的是要先弄清楚单片机的中断，定时器的原理以及一些常用的模块，像4__4矩阵键盘，12864图形液晶的显示□ad/da等这些模块的应用，一般的问题是能够解决的，还有一点就是要先能够读懂别人写的程序，然后自己再去写一些程序，自己动手写程序是很重要的，在写的过程中你才能

够发现自己的问题，主要是细节的问题，比如说timer定时器的初始化过程，图形液晶的忙检测和初始化过程，这些都是很模块化的过程。只有自己写一下才会有一个深入的理解。

另外，在单片机的应用中，必须要先弄清楚电路的原理，然后再此基础上来设计程序，优化程序，最终实现功能。

我学习单片机的思路是：

一、学好c语言

二、掌握单片机各功能的原理。

三、看懂别人的程序。

四、自己动手实践，哪怕只是仿真。

单片机的心得体会篇四

首先，学习单片机要有必须的基础：电子技术方面要有数字电路和模拟电路等方面的理论基础，个性是数字电路；编程语言要求汇编语言或c语言。要想成为单片机高手，推荐初学者首先学习汇编语言，学的差不多的时候，转入c语言学习。尽管汇编语言属于低级语言，编程效率低，可是较c语言具有目标代码简短，占用内存少，执行速度快等优点，更重要的是能使初学者尽快熟悉单片机的内部结构，并能对其进行精确的控制。汇编语言在单片机教材里面都会涉及，不需要单独购买教材和学习□c语言是一门学问，有很多专业书籍来讲解，并且对我们今后的编程生涯有绝对的好处，所以要深入学习，千万不要自以为看了某某的视频教程就以为掌握了c语言，那只是c语言的一部分。在那里给大家推荐一本单片机c语言程序设计参考书，马忠梅等著，北京航空航天大学出版

社出版的《单片机的c语言应用程序设计》，要求c语言基础。如果没学过c语言，推荐学习清华大学谭浩强编写的c语言程序设计，这本书写的不错，通俗易懂。

其次，是单片机教材选取。单片机是一门十分重视实践的技术，不能总是看书，但要学习它首先应看书，对单片机引脚、内部结构、寄存器和原理有必须地了解 and 感官认识，它的是怎样工作的，能干些什么？刚开始时，也许你看不明白，但这并不要紧，因为你还缺乏实践经验。此刻单片机应用广泛，所以各个厂家分别推出了自我的单片机，按内部结构体系派系分：51系列□pic系列□avr系列、摩托罗拉等等……我们没必要每样都学！因为他们的编程方法和调试过程以及内部指令结构有必须的相似，只要学精通一款就ok了！尤其是用c语言编程，就几乎不用分什么派系，可是我们要选取一款有代表性的知识范围广，并且入门容易，书籍多。一般来说□mcs—51系列单片机已经得到广泛的普及和应用，市场上它的资料也比较多，用的人也很多。给大家推荐一些参考书，学习时只需要一本就足够拉。书名：《新编mcs—51单片机应用设计》，哈尔滨工业大学出版，作者：张毅刚；书名：《单片机原理及应用》，高等教育出版社，作者：张毅刚等；书名：《单片机高级教程：应用与设计》，北京航空航天大学出版社，作者：何立民。相关教材还有很多，在这不一一列举。

定某个实验，就恭维的告诉你一声”恭喜你，学会了”自我学会了单片机，这有点可笑，这只能说明你算过关了，对单片机有了必须了解和会使用它了。可是单片机能完成的功能太多了，尤其是对外围器件的控制，综合起来能设计出许多意想不到的产品。所以除了入门外，精通可千万别轻易说出口。

最终，在熟练掌握和应用后，那能够说对于单片机方面的硬件你已经入门了，剩下的就是自我练习设计电路，不断的积

累经验。最终，自我完全设计具有个人风格的电路，产品，这样你就是单片机高手拉。只要过了第一关，后面的路就好走多了，万事开头难，大家可能都听过。时下多家电子类的报刊杂志如：《电子制作》《无线电》《电子报》《电子世界》都开设了详细的单片机教程专栏，对于想学习单片机的朋友来说帮忙很大，能够说此刻的单片机教程环境是最好的，有网络，有书籍，有报刊杂志，还有视频教程，元件的采购方面也十分充足，相关的器材又多便宜。如果每一天能抽出两小时的时间去学习，快的一个月，慢的.三个月就入门拉。

单片机的心得体会篇五

- 1、学习单片机没有捷径，别指望两三天就学会，要坚持不懈，重在积累。
- 2、别崇拜高手，别相信天才，大部分人都不是天才（相信你也不是）！
- 3、单片机是一门应用性和实践性很强的学科，要多动手，多做实验。
- 4、要学会参考别人的程序，减少自我琢磨的时间，迅速提高自我的编程潜力。
- 5、碰到问题能够借助网络来搜寻答案和对自我有帮忙的问题，能够大大减少你的开发时间。
- 6、要多交一些朋友，多交流。技术是靠不断的'积累和交流才会提高的，封闭自守只会更加落后。