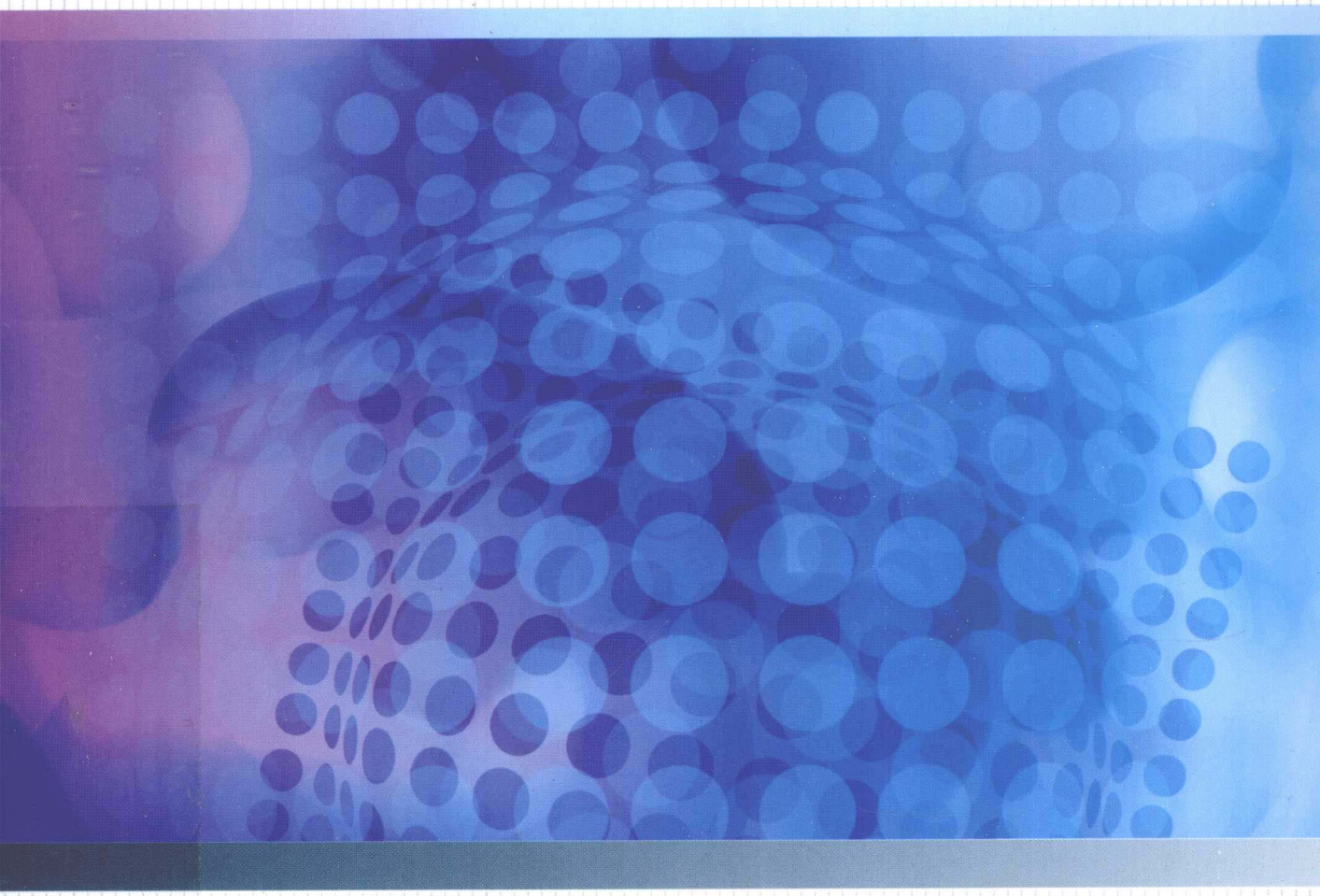
单片机应用程序设计技术（第3版）



周航慈 著

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

单片机应用程序设计技术

（第3版）

周航慈　著

北京航空航天大学出版社

内 容 简 介

书中总结了作者多年来在80C51系列单片机应用系统软件设计中的实践经验，归纳出一整套应用程序设计的方法和技巧。在内容安排上，不仅有实现功能要求的应用程序设计步骤、子程序、监控程序及常用功能模块设计方法，还以较大篇幅介绍了提高系统可靠性的抗干扰设计和容错设计技术以及程序测试的正确思想方法。附录中向读者提供了完整的系统程序设计样本和经过多年使用考验的定点运算子程序库与浮点运算子程序库的程序文本、注释及使用方法。

本书深入浅出，并配以大量实例，可作为从事单片机应用系统研究的工程技术人员的软件设计指导用书，也可作为高等院校相关专业师生的参考用书。

图书在版编目(CIP)数据

单片机应用程序设计技术 / 周航慈著. -- 3版. --
北京：北京航空航天大学出版社，2011.2
ISBN 978-7-5124-0276-8

Ⅰ. ①单… Ⅱ. ①周… Ⅲ. ①单片微型计算机—程序设计 Ⅳ. ①TP311.1

中国版本图书馆CIP数据核字(2010)第240982号

单片机应用程序设计技术(第3版)
周航慈 著
责任编辑 刘晓明
*
北京航空航天大学出版社出版发行
北京市海淀区学院路37号(邮编100191) http://www.buaapress.com.cn
发行部电话：(010)82317024 传真：(010)82328026
读者信箱：bhpress@263.net 邮购电话：(010)82316936
北京时代华都印刷有限公司印装 各地书店经销
*
开本：787×1 092 1/16 印张：22.75 字数：582千字
2011年2月第3版 2011年2月第1次印刷 印数：4 000册
ISBN 978-7-5124-0276-8 定价：38.00元

第3版前言

在20世纪80年代后期，单片机开始在我国迅速普及，其中MCS－51系列单片机普及最快。在高等院校中，“单片机原理及应用”课程的教材也以MCS－51系列单片机为主要内容。在此背景下，北京航空航天大学出版社开始组织一批单片机教学方面的老师，以何立民教授为主编，来撰写“单片机应用技术丛书”。按照丛书的规划和要求，笔者将从事单片机教学和科研中得到的经验教训进行了总结，并吸收各方面的有益知识，写成了《单片机应用程序设计技术》，并于1991年出版。

第1版中，笔者假定读者已经系统地掌握了单片机的原理，并能使用MCS－51指令系统编写各类简单的程序。第1章介绍开发一个应用项目的基本过程。第2章介绍程序设计的基本功。第3章介绍系统软件的框架如何搭起来。第4章介绍常用模块的设计技巧。这4章的内容要达到的目标就是帮助读者设计出一个完整的软件系统来。但这样设计出来的软件系统还只能算是“纸上谈兵”，经不起实践考验，离实际目标还差很远。第5章介绍抗干扰技术，以增强系统软件在实际环境中的生存能力。第6章介绍容错技术，以提高系统软件的先天素质。第7章介绍软件测试的有关方法，以此来发现和纠正软件系统中的绝大部分错误。后3章的内容为的是一个共同目标，使设计出来的软件能够从纸上走下来，并在实际应用中生存下去。因此，该书的目的不但要介绍一般的程序设计方法，而且以提高软件生存能力为重点，这对那些打算从事单片机开发应用的读者可能更有启发。

第1版出版后得到广大读者的好评和支持，但由于历史的原因，书中的80C31＋373＋2764三片基本系统已经过时，因此于2002年出版了该书的修订版。在修订版中，改为片内含程序存储器的80C51系列单片机；用计数器芯片来构成的硬件看门狗系统也已经过时，改为采用专用系统监控芯片或内含硬件看门狗的增强型单片机；在程序编写格式上，对直接地址单元进行操作的编程格式已经非常不实用，不利于移植和重复利用，全部改为变量和宏定义格式，使软件素质得到质的提高。

在修订版的第1章中，新增加了“编程语言与开发环境的选择”一节，介绍和比较了当今流行的编程语言与开发环境，供读者参考。在第3章中，新增加了“菜单驱动的监控程序设计方法”一节，这是一种人机界面最友好的监控程序结构，特别适用于高档应用系统，介绍出来供读者参考。在点阵液晶的性能价格比日益提高，应用日益广泛的情况下，在第4章中新增加了“点阵液晶图文混合显示中的软

件技术”一节，系统地介绍了图文混合显示的编程方法，特别是其中的排版技术很有实用价值。在第6章中，新增加了“互斥型输出的硬件容错设计”一节，介绍了在工业过程控制中的一个“麻烦”问题的解决方案，供读者参考。

学生学习“功能模块”程序设计比较容易，因为需要解决的问题比较单纯。而学习“监控”程序设计比较费力，因为需要解决系统整体协调问题，往往不知从何下手。即使一个很简单的系统，要完成系统程序设计，并将程序代码烧录到芯片中，而且能够正常运行，只有很少学生能够在规定的时间内独立完成任务，这种“教完知识就让学生独立完成系统设计”的教学方法已经被证明效率很低。为了加快入门的步伐，笔者在附录A中提供了两个风格完全不同的完整的应用程序样本，作为“字帖”，供初学者“临摹”。一般“临摹”两三次后就可以脱手自行进行简单的系统软件设计了，这比让学生自己“瞎折腾”要好得多，能够使更多的学生在较短的时间内“上路”，提高毕业后的就业能力。

第1版附录中的“MCS-51单片机实用子程序库”曾经得到广大读者的好评，正版用户遍及全国各地，经过10年来的使用，笔者进行了进一步优化，作为附录B提供给读者的“MCS-51单片机实用子程序库”是一个经过长期考验的优化的子程序库。

修订版于2002年出版后至今已超过8年，书中的某些内容已经明显过时，作者对修订版进行了修改，并补充一些新的内容，成为现在的第3版。

根据当前电子技术的发展现状，在第3版的第4章中，新增加了“有字库液晶显示屏”和“触摸屏”的内容。

第3版仍然以51单片机的汇编语言为编程语言，但在附录中增加了两个以C51为编程语言的程序样本，供读者学习C51语言参考，并建议读者尽可能将书中的各种汇编语言子程序改写为C51语言的函数，从而提高自己的软件编程能力。

在附录中，作者第一次公开了“MCS-51高精度浮点运算子程序库”，该子程序库原本是需要收费的商品软件。

在这次修改中，也保留了修订版中的某些实用性不高的内容，这些内容对提高软件编程能力还有一定的帮助，读者可以用类似思路来解决将来碰到的问题。

第3版在写作过程中，得到了北京航空航天大学出版社的大力支持和帮助，尤其是得到何立民教授的帮助，在此一并表示感谢！对于书中的错误和不足之处，望广大读者指正！

周航慈

2010年10月于北京

目　录

第1章

应用程序的设计步骤

单片机在智能仪器仪表、机电一体化产品和自动控制系统中的应用愈来愈广，很多老式仪表设备在进行升级换代的改造中，都将采用单片机作为首选方案。单片机的优越性能使电路设计变得更简单，但随之而来的是程序设计任务变得比较繁重。掌握正确的程序设计步骤可以加快开发速度，减少返工时间，提高系统软件的质量。

1.1 设计任务书的编写

每个应用项目在正式动手进行设计前，应该认真进行目标分析，编写出设计任务书。编写任务书时必须以用户的愿望为依据，最后必须得到用户的完全认可。如果项目设计者和用户不是同一经济单位，则必须通过一定的法律程序签订技术合同，将有关设计任务写进合同，以备将来项目验收时作为依据。由此可见，设计任务书必须尽可能详尽，指标必须明确。

设计任务书中填写有关技术指标的具体数据时要非常慎重。整个系统最终达到的技术指标是由各个环节共同作用后完成的。例如一个智能检测仪表，测试精度指标定为 0.05 %，表面上看，只要采用 12 位 A/D 转换器件就可以达到这个目标。其实不然，如果传感器的非线性、温漂等指标达不到这个水平，或者抗干扰措施不力，那么整个系统的指标是根本不能完成的，即使数字显示出足够多的位数，但它的低位数字跳跃不停，则输出的高精度也是虚假的。因此，必须通盘考虑之后，再定下各项技术指标，免得以后验收时无法通过。

一般情况下，技术指标达到某个限度之后，再提高一点点都是不容易的，为此可能要付出几倍的时间和经费。因此，当指标接近这个限度时（如国内先进水平或国际先进水平），必须充分作好技术力量和经济力量的准备。

任务书中除说明系统的各项具体技术指标外，还应对设备规模作出规定，这是硬件投资的主要依据。如主机机型、分机机型、需要哪些类型的传感器、配备哪些外部设备、操作台或操作面板的规格、执行单元的类型等均应该作出规定。如果内容较多，则往往以附件的形式单独编写。

任务书中还应说明操作规范，整个系统的操作使用者是用户单位，因此，操作规范必须充分尊重用户的职业习惯，使用户感到方便顺手。操作规范越详尽越好，这是系统软件的设计基础，千万不可马虎了事；否则，将使软件设计进展不顺利，造成重大返工。如果操作规范内容较

多,也应以附件的形式单独编写。

为了使设计任务书编写得合情合理(即在指定的期限内,不超出额定经费的前提下,能完成任务书中规定的各项指标),项目设计者必须是一个双重角色:一方面是计算机技术人员,懂得计算机的硬件设计和软件设计;另一方面又是一个系统操作者,懂得有关行业知识和基本的行业操作技能。因此,搞单片机应用开发的技术人员的知识面应尽可能广,这样,才能在项目的开发初期做到心中有数,编写的任务书也才能合情合理。如果项目开发者对所开发的项目还是门外汉,千万不可轻易签合同,必须先老老实实当一段时间"学徒",真正掌握该行业最基本的知识和技能,才可以动手编写任务书。

1.2 硬件电路设计

一个项目定下来后,经过详细调查,编制出任务书,就进入正式研制阶段。从总体上来看,设计任务可分为硬件设计和软件设计,这两者互相结合,不可分离;从时间上来看,硬件设计的绝大部分工作量是在最初阶段,到后期往往还要作一些修改。只要技术准备充分,硬件设计的大返工是较少的。软件设计的任务贯彻始终,到中后期基本上都是软件设计任务。随着集成电路技术的飞跃发展,各种功能很强的芯片不断出现,使硬件电路的集成度愈来愈高,硬件设计的工作量在整个项目中所占的比重逐渐下降。

另一方面,修改硬件电路有一些固有不利因素,这就是周期长、不灵活、消耗原材料。要改动一次硬件设计,就要重新制作电路板,安装元器件,调试电路。而软件的修改只要在开发系统上改动一些指令即可,基本上不需要消耗原材料。因此,硬件电路设计要仔细推敲,尽可能通过集体论证来拍板定稿,从而避免硬件电路大返工。硬件电路大返工往往迫使软件设计也大返工,延误项目的开发进程。为使硬件设计尽可能合理,应注意以下几方面。

(1) 尽可能采用功能强的芯片,以简化电路

功能强的芯片可以代替若干块普通芯片。随着生产工艺的提高,新型芯片的价格不断下降,并不一定比若干块普通芯片价格的总和高。

(2) 留有设计余地

在设计硬件电路时,要考虑到将来修改、扩展的方便。因为很少有一锤定音的电路设计,如果现在不留余地,将来可能要为一点小小的修改或扩展而被迫进行全面返工。为此,在硬件设计中要注意以下几点:

① 程序空间。选用片内程序空间足够大的单片机,以备将来扩充软件功能时能够容纳更大的程序规模。

② RAM 空间。89C51 内部 RAM 空间不大(128 字节),当要增强软件数据处理功能时,往往觉得不足。可选用 89C52 单片机(256 字节)或者内部含 XRAM 的单片机。

③ I/O 端口。在样机研制出来后进行现场试用时,往往会发现一些被忽视的问题,而这些问题是不能单靠软件措施来解决的。如有些新的信号需要采集,就必须增加输入检测端,有些物理量需要控制,就必须增加输出端。如果在硬件电路设计之初就多预留出一些 I/O 端口,虽然当时空着没用,但过后就正好派上用场了。

④ A/D 和 D/A 通道。和 I/O 端口同样的原因,将一些 A/D 和 D/A 通道空出来,将来很可能会解决大问题。

⑤ 机动布线区。如果是样机研制，在设计电路板时，开辟一小片机动布线区是有好处的。在机动布线区中，可以插入若干片集成电路插座，并有金属化孔，但无布线。当样机研制中发现硬件电路有明显不足，需要增加若干元器件时，即可在机动布线区中临时拉线来完成，从而避免整机大返工。当然，项目研制成功后，正式制板时就不需要机动布线区了。

(3) 以软代硬

单片机和数字电路本质的区别就是它具有软件系统。很多硬件电路能做到的，软件也能做到。因此，在硬件电路设计时，不要忘记还有软件作后台。原则上，只要软件能做到的，就不用硬件。硬件多了不但增加成本，而且使系统出故障的机会也增加了。以软代硬的实质是以时间代空间，软件执行过程需要消耗时间，因此，这种代替带来的不足就是实时性下降。当系统对某些事物的反应有严格的时间限制时，往往增加硬件电路是唯一的选择。但对一些实时性要求不是很高的场合，以软代硬是很合算的。如触点去抖动的软件延时方案，就比硬件双稳电路去抖动要合算得多；软件低通滤波算法就比硬件低通滤波电路优越得多。

(4) 监测电路的设计

在系统运行中有可能出现故障，如何及时采取措施，防止事态扩大并及时向操作者提出报警，就要求系统具有自诊断功能。为此。必须为系统设计有关的监测电路。这部分电路与系统正常的功能没有什么关系，往往容易忽视。在一些重要的自控系统中，系统的自诊断功能是很重要的，详情参阅第 6 章有关内容。

(5) 工艺设计

包括机架机箱、面板、配线、接插件等，必须考虑到安装、调试、维修的方便。另外，硬件抗干扰措施也必须在硬件设计时一并考虑进去，以免日后添加时发生困难，详情参阅第 5 章有关内容。

1.3　软件任务分析

软件任务分析和硬件电路设计结合进行，哪些功能由硬件完成，哪些任务由软件完成，在硬件电路设计基本定型后，也就基本上决定下来了。

软件任务分析环节是为软件设计作一个总体规划。从软件的功能来看可分为两大类：一类是执行软件，它能完成各种实质性的功能，如测量、计算、显示、打印、输出控制、通信等；另一类是监控软件，它是专门用来协调各执行模块和操作者的关系，在系统软件中充当组织调度角色的软件。这两类软件的设计方法各有特色：执行软件的设计偏重算法效率，与硬件关系密切，千变万化；监控软件着眼全局，主要处理人机关系，其特点是逻辑严密，千头万绪。

软件任务分析时，应将各执行模块一一列出，并为每一个执行模块进行功能定义和接口定义(应输入、输出定义)。在为各执行模块进行定义时，要将牵涉到的数据结构和数据类型问题一并规划好。

各执行模块规划好后，就可以规划监控程序了。首先根据系统功能和键盘设置选择一种最适合的监控程序结构。相对来讲，执行模块任务明确单纯，比较容易编程。而监控程序较易出问题，这如同当一名操作工人比较容易，而要当好一个厂长就比较困难了。

软件任务分析的另一个内容是如何安排监控软件和各执行模块。整个系统软件可分为后台程序(背景程序)和前台程序。后台程序指主程序及其调用的子程序，这类程序对实时性要

求不是很高,延误几十毫秒甚至几百毫秒也没关系,故通常将监控程序(键盘解释程序)、显示程序、打印程序等与操作者打交道的程序放在后台程序中来执行。而前台程序安排一些实时性要求较高的内容,如定时系统和外部中断(如掉电中断)。也可以将全部程序均安排在前台,后台程序为"使系统进入睡眠状态",以利于系统节电和抗干扰。

1.4 数据类型和数据结构的规划

上节中的软件任务分析只是一个粗糙的分析和大体上的安排,还不能开始编程。系统中各个执行模块之间有着各种因果关系,互相之间要进行各种信息传递。如数据处理模块和检测模块之间的关系,检测模块的输出信息就是数据处理模块的输入信息;同样,数据处理模块和显示模块、打印模块之间也有这种"产销"关系。各模块之间的关系体现在它们的接口条件上,即输入条件和输出结果上。为了避免出现产销脱节现象,就必须严格规定好各个接口条件,即各接口参数的数据结构和数据类型。这一步工作可以这样来做:将每一个执行模块要用到的参数和要输出的结果列出来,对于与不同模块都有关的参数,只取一个名称,以保证同一个参数只有一种格式。然后为每一个参数规划一种数据类型和数据结构。

从数据类型上来分类,可分为逻辑型和数值型,但通常将逻辑型数据归到软件标志中去考虑,而将"数据类型分类"理解为"数值类型分类"。数值类型可分为定点数和浮点数。定点数有直观、编程简单、运算速度快的优点,其缺点是表示的数值动态范围小,容易溢出。浮点数则相反,数值动态范围大,相对精度稳定,不易溢出;但编程复杂,运算速度低。

如果一个参数的变化范围有限,就可用定点数来表示,以简化程序设计和加快运行速度。如某温度控制系统,温度范围为33.0~44.0 ℃,控制精度为0.1 ℃。如果用一个字节来表示温度(温度分辨率为0.05 ℃),就可以表示12.8 ℃的温度变化范围。采用坐标变换算法后,00H~0FFH就可以表示32.0~44.75 ℃的温度范围了,从而实现一个字节的定点表示方法。当参数的变化范围太宽时,只好采用浮点数来表示,如智能电桥中被测对象的变化范围达10个数量级(1 pF~10 000 μF),定点数是无法胜任的。

如果某参数是一系列有序数据的集合,如采样信号系列,则不光有数据类型问题,还有一个数据存放格式问题,即数据结构问题。在单片机应用系统中,数据结构比较简单,对于"数组",一般采用顺序存放的格式,这样就可以用简单的下标运算来访问数组中的任何一个元素。对于"队列",一般采用环形队列结构,为此应规划好三样东西:队列存储区域、队首指针和队尾指针,并计算出总共需要的RAM字节数。

1.5 资源分配

完成数据类型和数据结构的规划后,便可开始分配系统的资源了。系统资源包括程序存储器(多为片内)、RAM、定时器/计数器、中断源等。在任务分析时,实际上已将定时器/计数器、中断源等资源分配好了。因此,资源分配的主要工作是RAM资源的分配。片外RAM的容量比片内RAM大,通常用来存放批量大的数据,如采样数据系列。真正需要认真考虑的是片内RAM的分配。

片内RAM分配时应注意充分发挥各自的特长,做到物尽其用:00H~1FH这32字节可

以作为工作寄存器，其中00H～0FH可用来作为0区、1区工作寄存器。在一般的应用系统中，后台程序用0区工作寄存器，前台程序用1区工作寄存器。如果有高级中断，则高级中断可用2区工作寄存器(10H～17H)。如果前台程序中不使用工作寄存器，则系统只需0区工作寄存器。未作工作寄存器的其他单元便可以转为其他使用目的了。系统上电复位时，自动定义0区为工作寄存器，1区为堆栈，并向2区、3区延伸。如果系统前台程序要用1区、2区作工作寄存器，就应将堆栈空间重新规划。

在工作寄存器的8个单元中，R0和R1具有指针功能，是编程的重要角色，应充分发挥其作用，尽量避免用来做其他事情。

20H～2FH这16字节具有位寻址功能，用来存放各种软件标志、逻辑变量、位输入信息副本、位输出信息副本、状态变量、逻辑运算的中间结果等。当这些项目全部安排好后，保留一两个字节备用，剩下的单元才可改作其他用途。

30H～7FH为一般通用寄存器，只能存入整字节信息。通常用来存放各种参数、指针、中间结果，或用做数据缓冲区。也常将堆栈安放在片内RAM的高端，如68H～7FH。

89C52等增强型单片机片内RAM空间为256字节或更多，80H～0FFH同样可以作为一般通用寄存器来使用，但只能通过R0和R1来间接使用，故适合安排各种数组和表格。

如果将系统的各种开销安排后，所剩单元很少，这往往不是好兆头。应该留有足够的余地，因为现在还处于规划阶段，随着软件设计的发展进程，几乎都会出现新的资源要求。如果在规划阶段资源已经很紧张，则建议修改硬件设计，增加RAM资源。

RAM资源规划好后，应列出一张RAM资源的详细分配清单，作为编程依据。

1.6　编程及调试

上述各项准备工作都完成后，就可以开始编程了。如果项目开发者是一个群体，就可以分工进行，每个人完成其中的一部分软件任务。每部分任务都有一定的独立性，各任务之间的关系用接口条件明确定义。

软件设计有两种方法：一种是自上而下，逐步细化；另一种是自下而上，先设计出每一个具体的模块(子程序)，然后再慢慢扩大，最后组成一个系统。两种方法各有优缺点。自上而下的方法在前期看不到什么具体效果，对于初学者来说，心中总是不踏实。而自下而上的方法一开始就有效果，每设计并测试好一个模块，就能看到一个实际效果，给人一步一个脚印的感觉，对初学者比较有利，能树立信心。因此，在分工时，应将一些基本的低级模块让初学者去完成，而项目负责人编制自上而下的总体框架程序(监控程序)比较合适。关于监控程序和功能模块程序的设计方法，后续各章还要详细讨论。

单片机由于本身没有开发能力，故编程调试均在各种类型的开发系统上进行，其基本过程是相同的：用编辑软件编辑出源程序，再用编译软件生成目标代码，如果源程序中有语法错误则返回编辑过程，修改源文件后再继续编译，直到通过这一关。然后对程序进行测试，纠正测试中发现的错误。接着就在开发系统上仿真运行(如果开发系统功能不足，便将目标代码写入芯片中，插入样机中运行)，试运行中将会发现不少设计错误(不是语法错误)，再从头修改源程序，如此反复直到基本成功，就可投入实际环境中试用。在实际使用中，又会发现不少实验室中难以发现的问题，这时再对软件和硬件做必要的修改，直到能在实际环境中比较稳定可靠地

运行,方有把握通知用户来验收。

在单片机应用项目的开发过程中,应切记中国的一句民间谚语:“磨刀不误砍柴工。”前期的调查研究工作要细致,操作规范要和用户谈妥,软硬件设计的论证要充分,各项指标要吃准,数据格式要定好,模块功能和接口条件要落实,最后才是焊电路板、设计程序、调样机。前期准备工作马虎一点儿,后期返工将造成数倍的时间损失,甚至有可能无法完成原定任务,被迫修改合同或任务书,给双方造成损失,并影响开发单位的信誉。

1.7 编程语言与开发环境的选择

目前单片机应用系统中软件的开发主要采用汇编语言和C语言,或者采用汇编语言与C语言混合编程。

采用汇编语言编程必须对单片机的内部结构和外围电路非常了解,尤其是对指令系统必须非常熟悉,故对程序开发者的要求是比较高的。用汇编语言开发软件是比较辛苦的,程序量通常比较大,方方面面均需要考虑,一切问题都需要由程序设计者安排。

采用C语言编程时,只要对单片机的内部结构和外围电路基本了解,对指令系统则不必非常熟悉,故对程序开发者的硬件素质要求不是很高。用C语言开发软件相对比较轻松,很多细节问题不需要考虑,编译软件会替设计者安排好。故C语言在单片机软件开发中的应用越来越广,使用者越来越多。

单纯采用C语言编程也有不足之处,在一些对时序要求非常苛刻的场合,只有汇编语言能够很好地胜任。故在很多情况下,采用C语言和汇编语言混合编程是最佳选择。

从编程难度来看,汇编语言比C语言要难得多,作为一个立志从事单片机系统开发的科技人员,必须熟练掌握汇编语言程序设计方法,在熟练掌握汇编语言编程之后,学习C语言编程将是一件非常容易的事情,并且能够将C语言和汇编语言非常恰当地混合在一起,以最短的时间和最小的代价,开发出高质量的软件。

由于汇编语言程序设计是每一个单片机从业人员都懂的程序设计语言,但不是每一个人都已经熟练掌握,故本书将采用MSC-51系列单片机汇编语言来讨论各种问题。关于C语言(如C51)的程序设计技术,这方面的书籍已经不少,本书将不涉及(但在附录A中提供了两个用C51编写的程序样本)。笔者强烈建议:在基本掌握汇编语言程序设计技术之后,应该学习C语言程序设计技术(入门是比较容易的),使自己的程序设计水平上一个新的层次。

开发环境有两种:基于“裸机”的编程环境和基于“操作系统”(实时多任务操作系统)的编程环境。在基于裸机的编程环境下,开发者面临的是一个完全空白的单片机芯片,一切程序都必须由开发者来设计;在基于操作系统的编程环境下,开发者面临的是一个具有“实时多任务操作系统”内核的系统,在操作系统的基础上来进行程序设计时,只需要完成系统各项任务的程序设计,任务的管理和调度等基本操作由操作系统内核来完成。

显然,基于操作系统的编程环境可以得到高可靠、高效率的软件,但也要付出一定的代价:操作系统内核一是要花钱购买,二是要占用系统资源。因此,在系统资源紧张、成本要求苛刻时,就不宜采用操作系统内核。很多采用廉价单片机开发的小型电子产品功能单纯,程序量一般在16K之内,完全没有采用操作系统的必要。

采用操作系统内核的最佳场合是实时性要求高、任务比较多(控制对象多、检测对象多、系

统比较复杂)的系统。在这种系统中,采用的单片机档次比较高,系统资源比较充足,一般开发成本的预算也较高。采用基于操作系统的开发环境有利于在较短的时间内完成系统开发任务,所得到的软件系统的可靠性也有保障。

本书不准备介绍基于操作系统环境的编程技术,因为实时多任务操作系统本身就足够用一本书来介绍。因此,本书介绍的是基于裸机编程环境的汇编言语程序设计技术。也就是说,本书介绍的是一种最辛苦的编程技术,也是一个单片机系统开发者必须掌握的基本功。掌握基于裸机编程环境的汇编言语程序设计技术后,再掌握基于实时多任务操作系统编程环境的C语言编程技术,将如虎添翼,得心应手,并有希望成为业内专家。笔者的另外一本书《基于嵌入式实时操作系统的程序设计技术(第2版)》已经出版,可作为进一步学习程序设计技术的参考书。