

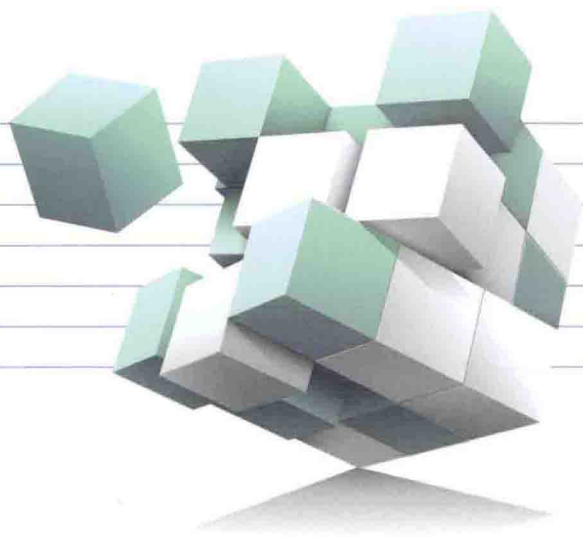
极光

信盈达技术创新系列图书

嵌入式

C语言 实战教程

李令伟 周中孝 黄文涛 王苑增◎编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

信盈达技术创新系列图书

嵌入式 C 语言实战教程

李令伟 周中孝 黄文涛 王苑增 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要介绍了嵌入式 C 语言程序设计基础知识、基本数据类型、各种运算符与表达式、C 语言 9 条基本语句和 32 个关键字、函数、数组、指针、结构体、共用体、枚举型、链表、文件、预处理命令、算法和类型定义符、五子棋人机智能对战等内容。每个知识点都有例子程序，如常用的 12 种算法、基本 C 语言语句使用范例等。

本书将 C 语言与嵌入式技术紧密结合起来，适合从事嵌入式开发的初学者，或者由单片机转向嵌入式开发的人员学习，也可作为高等院校相关专业教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

嵌入式 C 语言实战教程 / 李令伟等编著. —北京：电子工业出版社，2014.6
(信盈达技术创新系列图书)

ISBN 978-7-121-23089-9

I. ①嵌… II. ①李… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字（2014）第 084769 号

策划编辑：陈晓猛

责任编辑：张 慧

印 刷：北京京师印务有限公司

装 订：北京京师印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：14.75 字数：378 千字

版 次：2014 年 6 月第 1 版

印 次：2014 年 6 月第 1 次印刷

印 数：3 000 册 定价：39.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zits@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

C 语言是程序设计中最为活跃的高级语言之一。嵌入式 C 语言是进入嵌入式行业的必修课程，深圳信盈达嵌入式学院紧跟科技的发展，为了让读者能够尽快融入嵌入式行业，推出了《嵌入式 C 语言实战教程》。

本书面向从事嵌入式开发的初学者，或者由单片机转向嵌入式开发的人员，也可作为高等院校相关专业教材。在编写过程中，作者力求使本书体现如下特点。

(1) 对嵌入式 C 语言中的重点、难点进行分解，分散编排，使读者在学习过程中循序渐进、平滑过渡。

(2) 强调启发性教学，一方面强调教学，另一方面强调让读者从思考中加深理解。

(3) 对嵌入式 C 语言的难点、重点和例子程序做详细的注释，便于读者的理解和掌握。

(4) 文字简明，表达通俗易懂。例子程序的编写立足于读者能理解透彻，对算法技巧、编程规范等做了详细的说明。

本书的例子程序来源于典型算法及工程项目的精选，为后期学习 ARM、Linux 做准备。

本书共分 19 章。其中，第 1~7、19 章由深圳信盈达嵌入式学院李令伟编写；第 9~13 章由深圳信盈达嵌入式学院周中孝编写；第 14~16 章由深圳信盈达嵌入式学院黄文涛编写；第 8、17、18 章由深圳信盈达嵌入式学院王苑增编写。

本书的出版离不开深圳信盈达电子有限公司所有同事的支持和帮助，尤其是牛乐乐和陈志发等工程师，在此向他们表示衷心的感谢。另外，感谢电子工业出版社的各位编辑，是他们认真专业的审核，使本书由初稿变成了精美的图书。

由于时间仓促，编者水平有限，书中难免有不恰当的地方，恳请广大读者批评指正，联系邮箱：niusdw@163.com。

编著者

2014 年 6 月

第 1 章 嵌入式 C 语言概述	1
1.1 嵌入式 C 语言简介	1
1.2 嵌入式 C 语言的发展历史	1
1.3 嵌入式 C 语言的主要特点	1
1.4 单片机的汇编语言与嵌入式 C 语言比较	2
1.5 嵌入式 C 语言与标准 C 语言的异同	3
1.6 嵌入式 C 语言总结	3
第 2 章 嵌入式 C 语言程序的基本结构	4
2.1 嵌入式 C 语言入门实例	4
2.2 程序工作原理	4
2.3 源程序	5
2.4 嵌入式 C 语言程序的基本结构	6
2.5 头文件	6
2.6 主函数	7
2.7 函数	7
第 3 章 常量与变量的类型	8
3.1 嵌入式 C 语言的基本数据类型	8
3.2 基本数据类型	8
3.3 常量与变量	9
3.3.1 常量	9
3.3.2 变量	10
3.4 自定义变量类型 typedef	13
第 4 章 运算符和表达式	14
4.1 运算符与表达式	14
4.1.1 赋值运算	14
4.1.2 算术运算	14
4.1.3 关系运算	15
4.1.4 逻辑运算	16
4.1.5 位运算	17
4.1.6 自增减运算及复合运算	18
4.1.7 条件运算符	18
4.1.8 逗号运算符	19

4.2	嵌入式 C 语言程序的基本结构总结	19
第 5 章	嵌入式 C 语言基本结构程序设计	20
5.1	概述	20
5.2	顺序结构：0 条基本语句	20
5.3	选择结构：2 条基本语句（if 和 switch-case-break）	20
5.3.1	if 语句	21
5.3.2	switch-case 语句	23
5.4	break 基本语句	25
5.5	循环结构：3 条基本语句（while、do...while、for）	25
5.5.1	while 语句	26
5.5.2	do...while 语句	27
5.5.3	for 语句	27
5.5.4	循环嵌套	29
5.6	其他语句（转移语句）：4 条基本语句（break、continue、goto、return）	29
5.6.1	循环语句中的 break 语句	29
5.6.2	continue 语句	30
5.6.3	goto 语句	30
5.6.4	return 语句	31
5.7	嵌入式 C 语言基本结构总结	31
第 6 章	函数	32
6.1	函数概述	32
6.2	函数声明	32
6.3	函数定义	33
6.4	函数的调用	33
6.4.1	函数的简单调用	33
6.4.2	函数的参数传递	34
6.4.3	函数的递归调用	36
6.5	数组作为函数参数	37
6.6	函数作用范围与变量作用域	38
6.7	函数总结	39
第 7 章	数组	40
7.1	一维数组	40
7.1.1	一维数组的定义	40
7.1.2	一维数组元素的引用	41
7.1.3	一维数组的初始化	41
7.2	二维数组	43
7.2.1	二维数组的定义	43
7.2.2	二维数组元素的引用	43
7.2.3	二维数组的初始化	43

7.3	字符数组	45
7.4	嵌入式 C 语言中数组初始化规则	47
7.5	数组总结	47
第 8 章	指针	48
8.1	指针概述	48
8.2	指针和地址	48
8.2.1	指针变量的定义	48
8.2.2	指针变量的引用	49
8.2.3	指针和数组	51
8.2.4	字符指针	51
8.2.5	指针数组的定义格式	52
8.3	函数指针	53
8.3.1	函数指针定义	53
8.3.2	函数指针类型	54
8.3.3	函数类型	54
8.3.4	通过指针调用函数	55
8.3.5	返回指向函数的指针	56
8.4	实验范例：键盘扫描	57
8.4.1	键盘接口	57
8.4.2	按键开关的抖动问题	57
8.4.3	编程范例	58
8.5	指针总结	60
8.6	基本 C 语言语句使用范例	61
8.6.1	用顺序结构实现流水灯	61
8.6.2	用单种选择语句 if 实现流水灯	63
8.6.3	用多种选择语句 if...else、if...else if...else 实现流水灯	64
8.6.4	用多种选择语句 switch-case-break 实现流水灯	66
8.6.5	用循环语句 for 实现流水灯	67
8.6.6	用循环语句 while 实现流水灯	68
8.6.7	用循环语句 do...while 实现流水灯	69
8.6.8	用转移语句 goto 实现流水灯	71
8.6.9	用函数调用的方式实现流水灯	72
8.6.10	用数组实现流水灯	73
8.6.11	用指针实现流水灯	74
8.6.12	用指针+数组实现流水灯	76
8.6.13	用指针+数组+函数实现流水灯	77
8.6.14	用结构体实现流水灯	78
8.6.15	用结构体数组实现流水灯	80
8.6.16	用结构体数组+指针实现流水灯	81

第 9 章 结构体	83
9.1 结构体概述	83
9.2 结构体变量	84
9.2.1 结构体变量定义	84
9.2.2 结构变量成员的表示方法	86
9.3 结构变量的初始化和赋值	87
9.3.1 结构变量的初始化	87
9.3.2 结构变量的赋值	88
9.4 结构数组的定义	88
9.5 结构指针变量的说明和使用	90
9.5.1 指向结构变量的指针	90
9.5.2 指向结构数组的指针	92
9.5.3 结构指针变量作为函数参数	93
9.6 结构指针总结	95
第 10 章 共用体	96
10.1 共用体概念	96
10.2 一般定义形式	96
10.3 共用体变量的引用方式	97
10.4 共用体类型数据的特点	97
10.5 共用体总结	98
第 11 章 枚举型	99
11.1 枚举类型	99
11.2 枚举类型的定义和枚举变量的说明	99
11.3 枚举类型变量的赋值和使用	100
11.4 枚举类型总结	101
第 12 章 链表	102
12.1 动态存储分配	102
12.2 链表的概念	104
第 13 章 C 语言文件	110
第 14 章 预处理命令	113
14.1 预处理命令概述	113
14.2 宏定义	113
14.2.1 无参宏定义	113
14.2.2 带参宏定义	117
14.3 文件包含	121
14.4 条件编译	122
14.5 预处理命令总结	125
第 15 章 算法和类型定义符	126
15.1 算法	126

15.1.1	程序的灵魂——算法	126
15.1.2	算法分类	126
15.1.3	算法的特性	126
15.1.4	算法形式	126
15.2	结构化程序设计方法	127
15.3	类型定义符 typedef	127
15.4	算法和类型定义符总结	128
15.5	常用的 10 种算法实例	128
15.5.1	冒泡排序算法	128
15.5.2	回文算法	130
15.5.3	幂运算	131
15.5.4	加法运算	132
15.5.5	求直角三角形边长	133
15.5.6	在排序号的数组中插入元素	133
15.5.7	字符串的拼接	134
15.5.8	闰年判断	135
15.5.9	字符串查询	136
15.5.10	输出三位水仙花数	137
15.5.11	计算某个日期对应该年的第几天	138
15.5.12	输出月份对应的英文名称	139
第 16 章	五子棋人机智能对战	140
16.1	五子棋人机智能对战界面	140
16.2	五子棋人机智能对战软件说明	141
16.3	五子棋人机智能对战程序	141
第 17 章	程序模块化设计	173
17.1	模块化设计的优势	173
17.2	模块化设计的步骤	173
17.2.1	建立两个文件	173
17.2.2	编写 C 语言文件函数实体	174
17.2.3	编写.h 文件	174
17.2.4	在工程中添加 C 语言文件	175
17.2.5	工程文件的管理	175
17.3	模块化设计总结	177
第 18 章	程序编程规范及优化	178
18.1	嵌入式 C 语言程序编程规范	178
18.1.1	编程总原则	178
18.1.2	编程举例	180
18.1.3	注释	181
18.1.4	命名	182

18.1.5 编辑风格	182
18.2 C 语言程序编程规范总结	183
18.3 程序优化	186
18.4 项目管理知识	186
18.4.1 项目定义	186
18.4.2 项目三要素	186
18.4.3 项目过程	187
18.4.4 项目评估标准	187
18.5 电子产品开发流程	187
第 19 章 嵌入式 C 语言编程常见错误和程序调试	190
19.1 嵌入式 C 语言编程常见错误	190
19.2 C 语言程序调试常见错误及警告的解决方法	190
19.3 C 语言编译器错误信息中文翻译	195
19.4 MDK C 常用警告原因及处理方法	199
附录 A Microsoft Visual C++ 6.0 软件使用介绍	202
A.1 工程 (Project) 及工程工作区 (Project Workspace)	202
A.2 启动并进入 VC 6.0 的集成开发环境	203
A.3 创建工程并输入源程序代码	204
A.4 不创建工程, 直接输入源程序代码	207
A.5 编译、链接后运行程序	207
A.6 VC 6.0 常见快捷键操作	208
附录 B MDK 建立工程	211
附录 C Keil C51 建立工程	216
附录 D 嵌入式 C 语言关键字	222
附录 E 常用字符与 ASCII 代码对照	225

1.1 嵌入式 C 语言简介

嵌入式在生活的应用越来越广泛，产品种类也越来越多。由于嵌入式 C 语言可读性强、移植性好，与汇编语言相比，大大减轻了软件工程师的劳动强度，因而越来越多的嵌入式工程师开始使用嵌入式 C 语言编程。

1.2 嵌入式 C 语言的发展历史

嵌入式 C 语言是一种使用非常方便的高级语言。所以，在嵌入式产品的开发应用中，除了使用汇编语言外，也逐渐引入了嵌入式 C 语言。

嵌入式（ARM、Linux）C 语言除了遵循一般嵌入式 C 语言的规则外，还有其自身的特点，例如中断服务函数（如 `interrupt n`），对嵌入式单片机特殊功能寄存器的定义是嵌入式 C 语言所特有的，是对标准嵌入式 C 语言的扩展。

本教程将初步介绍嵌入式 C 语言在单片机开发中的运用，并对嵌入式 C 语言程序的开发软件 MDK 的使用进行详细说明。

1.3 嵌入式 C 语言的主要特点

嵌入式 C 语言发展迅速，而且成为最受欢迎的语言之一，主要是因为它具有强大的功能。用嵌入式 C 语言加上一些汇编语言子程序，就更能显示嵌入式 C 语言的优势了，像 PC-DOS、WORDSTAR 等都是用这种方法编写的。嵌入式 C 语言的特点如下。

1. 简洁紧凑、灵活方便

嵌入式 C 语言一共只有 32 个关键字、9 条控制语句。

程序书写自由，主要用小写字母表示。它结合了高级语言的基本结构和语句与低级语言的实用性。嵌入式 C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

2. 运算符丰富

嵌入式 C 语言的运算符包含的范围很广泛，共有 34 个运算符。嵌入式 C 语言把括号、

赋值、强制类型转换等运算都作为运算符处理，从而既使嵌入式 C 语言的运算类型极其丰富，又使其表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富

嵌入式 C 语言的数据类型包括整型、实型、字符型、数组类型、指针类型、结构体类型 and 共用体类型等，能够用于实现各种复杂数据类型的运算。并引入了指针概念，使程序效率更高。另外，嵌入式 C 语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

4. 嵌入式 C 语言是结构式语言

结构式语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护及调试。嵌入式 C 语言是以函数的形式提供给用户的，这些函数可以方便地调用，并由多种循环语句和条件语句控制程序流向，从而使程序完全结构化。

5. 嵌入式 C 语言语法限制不太严格，程序设计自由度高

一般的高级语言语法检查比较严，能够检查出几乎所有的语法错误。而嵌入式 C 语言允许程序编写者有较大的自由度。

嵌入式 C 语言允许直接访问物理地址，可以直接对硬件进行操作。因此，嵌入式 C 语言既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样，对位、字节和地址进行操作，而这三者是计算机最基本的工作单元，可以用来编写系统软件。

6. 嵌入式 C 语言程序生成代码质量高，程序执行效率高

嵌入式 C 语言一般只比汇编程序生成的目标代码效率低 10%~20%。

7. 嵌入式 C 语言适用范围大，可移植性好

嵌入式 C 语言有一个突出的优点就是既适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。

8. 嵌入式 C 语言突出应用场合

对于操作系统、系统使用程序及需要对硬件进行操作的场合，使用嵌入式 C 语言编程明显优于其他高级语言，许多大型应用软件都是用嵌入式 C 语言编写的。嵌入式 C 语言具有强大的绘图能力、可移植性及很强的数据处理能力。因此，适于编写系统软件、三维图形、二维图形和动画。它是数值计算的高级语言。

1.4 单片机的汇编语言与嵌入式 C 语言比较

嵌入式 C 语言程序与汇编语言程序从编写特点上比较，主要有以下六点区别。

(1) 嵌入式 C 语言程序中的主函数是汇编程序中的主程序，函数是汇编语言程序中的

子程序。程序运行都是从主函数或主程序开始，并终止于主函数或主程序的最后一条语句。在编写方面，汇编语言程序中的主程序必须编写在整个程序的最前面，因为汇编语言程序运行时是从整个程序的第一行开始的；而嵌入式 C 语言程序中的主函数可以放在程序的前面，也可放在后面或其他位置，且无论主函数在什么位置，程序运行时都会先自动找到主函数，并从主函数的第一条语句开始执行。

(2) 编写嵌入式 C 语言程序一般使用小写英文字母，关键字均为小写英文字母，也可以使用大写英文字母，但大写字母一般都有特殊意义。

(3) 嵌入式 C 语言严格区分字母大小写，也就是说，abc、Abc、ABC 是三个不同的名称；而汇编语言不区分字母大小写，编程时大小写字母可以混用。

(4) 嵌入式 C 语言不使用行号，一行可以写多条语句，但每一条语句最后必须有一个“;”作为结尾；而汇编语言一行就是一条语句。

(5) 嵌入式 C 语言每一个独立完整的程序单元都由一对大括号括起来，大括号必须成对使用。

(6) 嵌入式 C 语言的程序注释信息需要使用“/*”和“*/”括起来，如“/*头文件*/”，或用双斜杠，如“//头文件”；而汇编语言程序语句的注释信息使用一个分号，如“; 延时程序”。

汇编语言和嵌入式 C 语言的性能比较见表 1.1。

表 1.1 汇编语言和嵌入式 C 语言的性能比较

特 性	汇 编 语 言	嵌 入 式 C 语 言
实时性	强	弱
占用系统资源	少	多
可读性	弱	强（结构化编程、可读性强、便于维护）
可修改性	弱	强
健壮性	弱	强
应用领域	应用于实时性要求比较高的工业控制场合，如工业控制、小家电等领域	应用于程序量较大、功能较复杂，且对实时性要求不高的场合，如医疗器械、安防等领域

1.5 嵌入式 C 语言与标准 C 语言的异同

用嵌入式 C 语言编写嵌入式应用程序与标准 C 语言程序编写的不同之处就在于，可根据嵌入式的处理器存储结构及内部资源定义相应的嵌入式 C 语言中的数据类型和变量，其他的语法规则、程序结构及程序设计方法都与标准 C 语言程序设计相同。

1.6 嵌入式 C 语言总结

嵌入式 C 语言是一种较为高级的语言，通过其支持的各种编译器，能够将嵌入式 C 语言编译成适合各个平台的汇编代码和机器代码，具有非常优秀的移植性。例如，在 Linux 中除了处理器相关的部分外，全部采用 C 语言编写，因此已经被移植到了几乎所有的 CPU 上。C 语言比其他高级语言更接近于适合自然语言的特性，更适合于嵌入式编程。

嵌入式 C 语言程序的基本结构

2.1 嵌入式 C 语言入门实例

下面介绍一个简单的嵌入式 C 语言编程实例，使读者初步了解嵌入式 C 语言的特点。Super800 综合实验仪中，单片机 P2 端口接 8 个发光二极管，电路如图 2.1 所示。程序的功能是使 8 个发光二极管循环点亮，即常见的跑马灯。

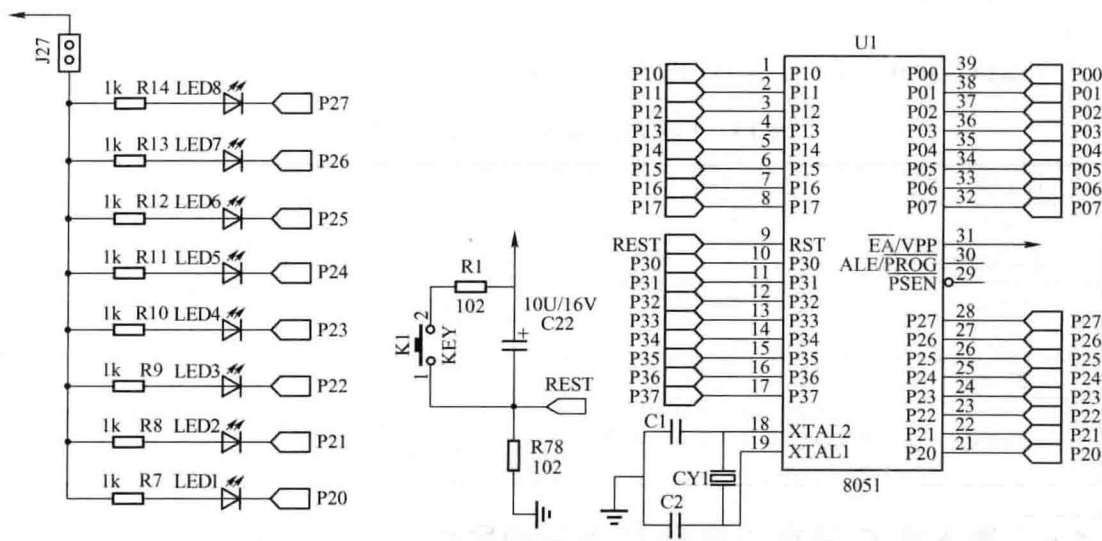


图 2.1 Super800 实验板硬件连接图

2.2 程序工作原理

由图 2.1 所示电路图可以看出，发光 LED 接法是共阳极的，只要首先将数据 1111 1110B 送到 P2 输出，就能把 P2 端口对应的 1 个发光二极管点亮，然后依次将 0xfe、0xfd、0xfb、0xf7、0xef、0xdf、0xbf、0x7f 送到 P2 输出，然后循环，即可控制 8 个灯循环点亮。

以下是跑马灯的源程序，帮助读者理解该程序所描述的功能。注意这个嵌入式 C 语言程序的编写格式，这是一个标准的格式，也就是常说的编程规范，以后都要按照这样的格式编程。

2.3 源程序

1. 源程序 1

```

/*****
*公司名称: 深圳信盈达电子有限公司
*模块名: 跑马灯
*版本信息: V2.1
*说明: Super800 开发板上 J27 排针用跳线帽连接。
*****/

#include <reg52.h> /*嵌入式 C 语言头文件*/
void delay(); /*函数声明 delay()延时函数
/*-----函数-----*/
void delay() // 延时子函数
{
    unsigned int i; //声明无符号整型变量 i
    for(i=0;i<30000;i++); //循环延时语句
}
/*-----主函数-----*/
void main(void) //主函数 void main(void)
{
    while(1) //无限循环
    {
        P2=0xfe; //点亮 P2.0 端口对应的灯 0xfe=1111 1110B
        delay(); //调用延时函数 delay()
        P2=0xfd;
        delay();
        P2=0xfb;
        delay();
        P2=0xf7;
        delay();
        P2=0xef;
        delay();
        P2=0xdf;
        delay();
        P2=0xbf;
        delay();
        P2=0x7f;
        delay();
    }
}

```

修改一下延时程序里面的数值，看一下灯循环的速度是否有变化。带着这个问题我们来学习下面的课程。

2. 源程序 2

```

/*****
*程序名称:   跑马灯程序
*版本信息:   V2.1
*说明:       Super800 开发板上 J27 连接。
*****/
#include <reg52.h>      /*嵌入式 C 语言头文件*/
#define uchar unsigned char /*声明变量 uchar 为无符号字符型, 长度为 1 字节, 值域范围为 0~255*/
#define uint  unsigned int  /*声明变量 uint 为无符号整型, 长度为 2 字节, 值域范围为 0~65535*/
void delay(uint t);      /*声明 delay()延时函数
/*****延时函数*****/
void delay(uint t)      /* 延时函数
{
    for(;t!=0;t--);
}
/*****主函数*****/
void main(void)
{
    uchar i;             /*声明无符号字符变量 i*/
    delay(1000);        /*调用延时子程序*/
    P2=0Xff;
    while(1)
    {
        for(i=0;i<8;i++)
        {
            P2=~(0X01<<i);    /*左移位运算符, 用来将 1 个数的二进制的各位全部左移
                               //移位后, 空白位补 0, 而溢出的位舍弃
            delay(50000);
        }
    }
}

```

要求:

- (1) 分析以上两程序哪一种更方便、简洁一些;
- (2) 分析哪一种引脚功能配置方式更灵活、直观。

2.4 嵌入式 C 语言程序的基本结构

嵌入式 C 语言程序一般由头文件、主函数和函数三部分组成。

2.5 头文件

头文件用来定义 I/O 地址、参数和符号。使用时通过 `#include` 指令加载, 将头文件包含在所编写的嵌入式 C 语言程序中, 这样在编写嵌入式 C 语言程序时, 就不需要考虑单片机

内部的存储器分配等问题了。例如，`#include <math.h>`中，`math.h` 为数学常用公式的头文件，使用时需要用`#include` 指令，并将头文件用括号“<>”括起来。某些简单的嵌入式 C 语言程序，只包括主程序和程序库所载入的头文件。

2.6 主函数

主函数，即主程序，是嵌入式 C 语言程序执行的开始，不可缺少。主函数以 `main` 为其函数名称，例如：

```
void main(void)
{
    嵌入式 C 语言语句;
}
```

嵌入式 C 语言的主函数是一个特殊的函数，每个程序必须有且只有一个主函数。嵌入式 C 程序运行时都是从主函数 `void main(void)` 开始的，主函数可以调用其他子函数，调用完毕后回到主函数，在主函数中结束整个程序的运行。

主函数内容用大括号“{}”括起来，括号内为嵌入式 C 语言程序语句，每行程序语句结束时加“;”。

2.7 函数

函数，即子程序，是指除了主函数之外的各个函数。函数可以命名为各种名称，但不可与嵌入式 C 语言保留字相同。函数与主函数的格式是一样的，函数内容用大括号“{}”括起来，括号内为嵌入式 C 语言程序语句，每行程序语句结束时加“;”。