

单片机C语言 编程实践

程利民 朱晓玲 主 编
姜新桥 陈亭志 副主编

 任务驱动

 行动导向

 工学结合

 学生主体

 过程考核

随书有配套的教学资源，获取多媒体电子课件可登录

www.hxedu.com.cn 免费下载 

单片机C语言 编程实践

第2版

第1章 单片机的发展与分类
第2章 单片机的内部结构
第3章 单片机的引脚与接口
第4章 单片机的编程方法
第5章 单片机的应用

清华大学出版社

清华大学出版社

高等职业教育教学改革系列规划教材

单片机 C 语言编程实践

程利民 朱晓玲 主 编

姜新桥 陈亭志 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书以拥有自主知识产权的单片机实践板为依托,以C语言为主线,介绍了单片机编程实践知识,内容包括C语言的基础知识、Keil软件的使用、C51程序的编写与调试方法及其他相关知识。书中所有实例和全部实践都经过了仿真软件或实践板的验证。

本书可作为高等职业教育“单片机技术与应用”和“C语言程序设计”两门课程的教材或教学参考书,也可以作为本科院校工程训练、电子制作的实践教材和相关专业课程参考教材,同时可供从事单片机应用与产品开发工作的工程技术人员学习参考。

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

单片机C语言编程实践/程利民,朱晓玲主编.--北京:电子工业出版社,2011.1

高等职业教育教学改革系列规划教材

ISBN 978-7-121-12713-7

I. ①单... II. ①程... ②朱... III. ①单片微型计算机—C语言—程序设计—高等学校:技术学校—教材
IV. ①TP368.1②TP312

中国版本图书馆CIP数据核字(2010)第261172号

策划编辑:田领红

责任编辑:田领红 特约编辑:徐 岩

印 刷: 北京京师印务有限公司

装 订:

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:17.75 字数:451千字

印 次:2011年1月第1次印刷

印 数:3000册 定价:29.80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 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

计算机技术在信息社会中的作用越来越重要，单片机技术作为其中的一个分支，广泛应用于工业控制、智能仪器仪表、机电一体化产品、家用电器等领域。在教学中，它不仅是一门实用技术，更是一种工具——提高学生的思维能力、动手能力和创新能力的工具。

单片机技术是和实践紧密相连的学科之一，很容易激发学生的学习兴趣，然而学生普遍感到这门课非常难学。笔者结合多年的单片机教学、开发经验，以产品为导向，选定一种具有典型教学价值和实用价值的单片机产品，其全部必备成本仅百元左右，指导学生自己动手，从元器件开始，设计原理图、线路板图，编程固化，快速见到效果，将学生变成主人，提高学习的积极性；根据完成该产品所需知识，组织教学内容和教学过程，需要什么就学习什么，在实践中学习。当学习完成后，就能够在现有产品基础上进行二次开发，很快进入实际应用。

本书共 11 章，前 8 章介绍 C 语言基础知识，第 9 章为单片机实践板（中级）的原理及其安装、焊接实践，第 10 章、第 11 章在单片机实践板（中级）基础上完成 2 个小系统的编程及调试。所编写程序都经过测试、优化，满足作为一个实际产品的基本条件，可充分利用其中的资源，直接应用到实际产品中。

本书由武汉职业技术学院程利民、朱晓玲主编，姜新桥、陈亭志副主编，其中，第 1~3 章由姜新桥编写，第 4~8 章由朱晓玲编写，第 9~11 章由程利民编写，附录 A~D 由陈亭志编写。

在本书的构思和编写过程中，得到了罗学恒、吴宇文、李望云、张幼华、王军、应文豹、熊海涛、桂毅等老师的大力支持和帮助，在此特表示感谢！

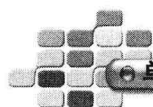
由于本书作者水平有限，编写时间仓促，书中难免有疏漏和欠妥之处，欢迎读者批评指正。

编 者
2010 年 10 月

Contents

目 录

第 1 章 C 语言概述	1
1.1 C 语言简介	1
1.2 简单 C 程序介绍	3
1.3 Keil C51 编译器仿真程序要求	5
1.4 实践	6
1.4.1 实践任务	6
1.4.2 实践设备要求	6
1.4.3 实践步骤	6
本章小结	8
习题 1	8
第 2 章 基本数据类型及运算	9
2.1 数据类型	9
2.2 数据的存放形式	11
2.2.1 整型数据的存放形式	11
2.2.2 实型数据的存放形式	11
2.3 数据的存放位置	12
2.4 常量	13
2.4.1 整数	13
2.4.2 浮点数	14
2.4.3 字符	14
2.4.4 字符串	15
2.4.5 符号常量	16
2.5 变量	18
2.5.1 变量的定义和初始化	18
2.5.2 隐含的变量数据类型和存放位置	19
2.5.3 特殊功能寄存器 (sfr)	20
2.5.4 bit、sbit 数据类型变量	20
2.5.5 硬件定义文件	21
2.5.6 变量的绝对定位	23
2.6 数据运用	23
2.6.1 算术运算和算术表达式	24



2.6.2	各类数值型数据的混合运算和类型转换	25
2.6.3	增1、减1运算	27
2.6.4	位运算	27
2.6.5	关系运算和逻辑运算	30
2.6.6	条件运算	32
2.6.7	逗号运算	33
2.6.8	长度运算符	33
2.6.9	指针运算符	33
2.6.10	复合赋值运算	34
2.6.11	运算符的优先级与结合性	34
2.7	实践2	35
2.7.1	实践任务	35
2.7.2	实践设备	36
2.7.3	实践步骤	36
	本章小结	37
	习题2	38
第3章	顺序、选择结构	39
3.1	基础语句	39
3.2	赋值语句	40
3.3	单个字符的输入、输出	40
3.3.1	putchar (字符输出函数)	41
3.3.2	getchar (键盘输入函数)	42
3.4	printf 格式输出函数	43
3.4.1	printf 格式输出函数的基本格式	43
3.4.2	printf 格式控制符	44
3.4.3	printf 宽度格式符	45
3.4.4	printf 函数使用实例	45
3.5	scanf 格式输入函数	48
3.5.1	scanf 格式字符串	49
3.5.2	scanf 数据输入操作	49
3.6	选择结构	50
3.6.1	if 语句	50
3.6.2	if 语句的嵌套与嵌套匹配原则	54
3.6.3	switch 语句	54
3.6.4	break 语句	56
3.7	实践3	58
3.7.1	实验任务	58
3.7.2	实践设备要求	58
3.7.3	实践步骤	58

本章小结	60
习题 3	60
第 4 章 循环结构	62
4.1 goto 语句	62
4.2 while 语句	63
4.3 do...while 语句	65
4.4 for 语句	68
4.5 多重循环	71
4.6 continue 语句	72
4.7 实践 4	73
4.7.1 实践任务	73
4.7.2 实践设备	73
4.7.3 实践步骤	73
本章小结	74
习题 4	75
第 5 章 数组	77
5.1 一维数组	77
5.1.1 一维数组的定义	77
5.1.2 一维数组的引用	78
5.1.3 一维数组的初始化	80
5.1.4 一维数组的应用例子	80
5.2 多维数组	82
5.3 字符数组和字符串	84
5.4 数组名作为函数参数	86
5.5 实践 5	88
5.5.1 实践任务	88
5.5.2 实践设备要求	89
5.5.3 实践步骤	89
本章小结	90
习题 5	90
第 6 章 函数和预编译	91
6.1 函数的定义	92
6.2 函数的返回值 (return)	93
6.3 函数的调用和声明	93
6.4 变量的作用范围和作用时间	96
6.4.1 内部变量	96
6.4.2 全局变量	96



6.4.3 变量的作用时间	97
6.4.4 单片机中变量的空间分配基本原则	98
6.5 函数的嵌套和递归	99
6.6 编译预处理	100
6.6.1 宏替换	101
6.6.2 文件包含	102
6.6.3 条件编译	103
6.6.4 编译控制	105
6.7 模块化程序设计的概念	107
6.8 模块化程序设计的概念	108
6.8.1 C 语言常用库函数	108
6.8.2 标准库函数的调用	109
6.9 实践 6	110
6.9.1 实践任务	110
6.9.2 实践设备要求	110
6.9.3 实践步骤	110
习题 6	112
第 7 章 C 指针	113
7.1 指针和指针变量的基本概念	113
7.2 指针变量的定义	114
7.3 指针变量的赋值	115
7.4 指针变量的运算	116
7.4.1 取内容运算符 “*”	116
7.4.2 加减算术运算和关系运算	117
7.5 指向数组元素的指针	118
7.6 指向字符串的指针	119
7.7 数组和指针	120
7.7.1 一维数组和指针	120
7.7.2 二维数组和指针	123
7.7.3 指针和字符串	127
7.7.4 指针数组的指针作为函数参数	129
7.8 函数的指针	132
7.9 数组、函数、指针综合应用	133
7.10 实践 7	134
7.10.1 实践任务	134
7.10.2 实践设备	134
7.10.3 实践步骤	134
本章小结	136
习题 7	136

第 8 章 结构、联合和枚举	137
8.1 结构	137
8.1.1 结构的定义	137
8.1.2 结构类型变量的说明	138
8.1.3 结构类型变量的赋值和使用	138
8.2 联合	139
8.2.1 联合的定义	140
8.2.2 联合变量的说明	140
8.2.3 联合变量的赋值和使用	141
8.3 枚举	142
8.3.1 枚举的定义	142
8.3.2 枚举变量的说明	142
8.3.3 枚举类型变量的赋值和使用	143
本章小结	144
习题 8	144
第 9 章 键盘、数码管、点阵屏	145
9.1 实践任务	145
9.2 相关知识——键盘的工作原理和接口方法	145
9.2.1 独立式接法键盘	145
9.2.2 矩阵式接法键盘	146
9.3 相关知识——数码管、点阵屏的工作原理和接口方法	147
9.3.1 数码管、点阵屏的工作原理和接口方法	148
9.3.2 LED 的静态显示方法	148
9.3.3 LED 数码显示器的动态显示方法	149
9.3.4 数码管的选择和驱动	149
9.4 相关知识——键盘、数码管、点阵屏集中控制芯片 BC7281	150
9.4.1 BC7281 简述	150
9.4.2 内部寄存器	151
9.4.3 数据传输	154
9.5 实践设备要求	156
9.6 实践步骤	156
第 10 章 让数码管和点阵屏显示动画	160
10.1 任务要求	160
10.2 相关知识——驱动程序基本种类	160
10.3 相关知识——点阵数据库	161
10.4 相关知识——基本驱动函数	163
10.5 实践设备要求	169
10.6 实践步骤	169



第 11 章 制作简单浮点计算器	172
11.1 任务要求	172
11.2 实践设备要求	172
11.3 实践步骤	172
附录 A Cx51 库函数	176
附录 B Cx51 编译错误、警告	217
附录 C 单片机软件编程基本知识	230
附录 D 计算机二级 C 语言真题	250
附录 E 单片机实践板原理图	270
附录 F 推荐的毕业设计课题	272
参考文献	273

第 1 章

C 语言概述

1.1 C 语言简介

C 语言于 20 世纪 70 年代诞生于美国的贝尔实验室。在此之前,人们编写程序主要使用汇编语言,由于汇编语言编写的程序依赖于计算机硬件,其可读性和可移植性都比较差;高级语言的可读性和可移植性虽然较汇编语言好,但一般高级语言又不具备低级语言能够直观地对硬件实现控制和操作、程序执行速度快的特点。在这种情况下,人们迫切需要一种既具有一般高级语言特性,又具有低级语言特性的语言,于是 C 语言就应运而生了。

C 语言兼有汇编语言和高级语言的优点,既适合于开发系统软件,也适合于编写应用程序,被广泛应用于事务处理、科学计算、工业控制、数据库技术等领域。C 语言具有以下特点。

1. C 语言是结构化的语言

C 语言程序有 3 种基本结构:顺序结构、选择结构、循环结构,由这 3 种基本结构组成的程序可以解决许多复杂的问题。C 语言通过结构化的控制语句,如 `if...else` 语句、`while` 语句、`switch` 语句及 `for` 语句等,可以方便地控制程序的流程。因此,C 语言是理想的结构化语言,符合现代编程风格的要求。

2. C 语言是模块化的语言

一般来说,一个较大的程序往往被分为若干个模块,每个模块用来实现特定的功能。在 C 语言中,用函数作为程序的模块单位,便于实现程序的模块化。在程序设计时,将一些常用的功能模块编写成函数,放在函数库中供其他函数调用,C 语言具有非常丰富的库函数。模块化的特点可以大大减少重复编程。

3. 语言简洁、紧凑,使用方便、灵活

C 语言总共只有 32 个关键字和 9 种控制语句,程序书写形式自由,主要用小写字母表示。一般语言中的许多结构单元在 C 语言中都通过库函数调用来完成,库函数可根据需要方便地扩充,压缩了一切不必要的程序组成部分。



4. 程序可移植性好

C 语言程序便于移植, 目前 C 语言在许多计算机上的实现大都是由 C 语言编译移植得到的, 不同机器上的编译程序大约有 80% 的代码是公共的。程序不做任何修改就可用于各种型号的计算机和各种操作系统。

5. 数据结构丰富, 具有现代化语言的各种数据结构

C 语言的基本数据类型有整型 (如 `int`、`unsigned int` 等)、实型 (如 `float`、`double` 等)、及字符型 (`char`) 等。在此基础上还可创建各种构造数据类型, 如数组、指针、结构体和共用体等。使用 C 语言还能实现复杂的数据结构, 如链表、树等。这样丰富的数据结构无疑极大地增强了 C 语言的功能。

6. C 语言运算符丰富、代码效率高

C 语言共有 34 种运算符, 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。在代码质量上, C 语言可与汇编语言相媲美, 其代码效率仅比用汇编语言编写的程序的代码低 10%~20%。

1978 年以后, C 语言的发展导致各种版本不断出现。1988 年, 美国国家标准化协会 (ANSI) 根据 C 语言问世以来的各种版本对 C 语言进行了发展和扩充, 制定了 ANSI C 标准, 1990 年国际标准化组织 ISO 公布了以 ANSI C 为基础的 C 语言的国际标准 ISO C, 从而保证了各种 C 语言对 ANSI C 的兼容。表 1-1 为 ANSI C 的关键字。

不同类型的计算机芯片千差万别, 但它们使用的 C 语言基本相同, 究其原因, 是因为各种计算机芯片都根据 ANSI C 标准, 制定了 C 语言编译器; 通过 C 语言编译器, 将 C 语言转化为相应的汇编指令, 完成相同的任务。

C51 是用于 MCS51 系列单片机、以 ANSI C 为基础、修改扩充的一套程序语言。德国 Keil 公司已经推出 V7.0 以上版本的 Cx51 编译器, 为 MCS51 系列单片机软件开发提供了全新的 C 语言环境, 同时保留了汇编代码的高效、快速等特点。表 1-2 为 C51 新增的专用关键字。

表 1-1 ANSI C 的关键字

关键字	用途	说明
<code>auto</code>	存储种类说明	用以说明局部变量, 默认值为此
<code>break</code>	程序语句	退出最内层循环
<code>case</code>	程序语句	<code>switch</code> 语句中的选择项
<code>char</code>	数据类型说明	单字节整型数或字符型数据
<code>const</code>	存储类型说明	在程序执行过程中不可更改的常量值
<code>continue</code>	程序语句	转向下一次循环
<code>default</code>	程序语句	<code>switch</code> 语句中的失败选择项
<code>do</code>	程序语句	构成 <code>do...while</code> 循环结构
<code>double</code>	数据类型说明	双精度浮点数
<code>else</code>	程序语句	构成 <code>if...else</code> 选择结构
<code>enum</code>	数据类型说明	枚举
<code>extern</code>	存储种类说明	在其他程序模块中说明了的全局变量
<code>float</code>	数据类型说明	单精度浮点数
<code>for</code>	程序语句	构成 <code>for</code> 循环结构

续表

关键字	用途	说明
goto	程序语句	构成 goto 转移结构
if	程序语句	构成 if...else 选择结构
int	数据类型说明	基本整型数
long	数据类型说明	长整型数
register	存储种类说明	使用 CPU 内部寄存的变量
return	程序语句	函数返回
short	数据类型说明	短整型数
signed	数据类型说明	有符号数, 二进制数据的最高位为符号位
sizeof	运算符	计算表达式或数据类型的字节数
static	存储种类说明	静态变量
struct	数据类型说明	结构类型数据
switch	程序语句	构成 switch 选择结构
typedef	数据类型说明	重新进行数据类型定义
union	数据类型说明	联合类型数据
unsigned	数据类型说明	无符号数数据
void	数据类型说明	无类型数据
volatile	数据类型说明	该变量在程序执行中可被隐含地改变
while	程序语句	构成 while 和 do...while 循环结构

表 1-2 C51 新增的专用关键字

关键字	用途	说明
bit	位标量声明	声明一个位标量或位类型的函数
sbit	位标量声明	声明一个可位寻址变量
sfr	特殊功能寄存器声明	声明一个特殊功能寄存器
sfr16	特殊功能寄存器声明	声明一个 16 位的特殊功能寄存器
data	存储器类型说明	直接寻址的内部数据存储器
bdata	存储器类型说明	可位寻址的内部数据存储器
idata	存储器类型说明	间接寻址的内部数据存储器
xdata	存储器类型说明	分页寻址的外部数据存储器
pdata	存储器类型说明	外部数据存储器
code	存储器类型说明	程序存储器
interrupt	中断函数说明	定义一个中断函数
reentrant	再入函数说明	定义一个再入函数
using	寄存器组定义	定义芯片的工作寄存器

1.2 简单 C 程序介绍

为了说明 C 语言源程序结构的特点, 先看下面例 1-1 所示的程序。这个程序用 C 语言编写, 其功能是: 控制单片机内部的一个数据不断加 1 递增, 用 8 个发光二极管显示变化过程; 当按下开关时, 蜂鸣器鸣叫。用 Keil C51 执行 C 程序时, 由于单片机没有标准的显示输出设备, 只能通过单片机的串行口进行仿真显示, 需调用串行口初始化子函数 init_rs232()。从



这个例子中可以了解到组成一个 C 语言源程序的基本部分和书写格式。

【例 1-1】

```
#pragma symbols code          /* 定义编译环境：产生符号列表、产生汇编程序列表 */
#include<stdio.h>              // 包含输入、输出库函数
sfr P0=0x80;                  // 定义特殊功能寄存器 P0 的地址
sfr P1=0x90;                  // 定义特殊功能寄存器 P1 的地址
sfr P4=0xe8;                  // 定义特殊功能寄存器 P4 的地址
sfr T2CON=0xc8;               // 定义特殊功能寄存器 T2CON 的地址
sfr SCON=0x98;                // 定义特殊功能寄存器 SCON 的地址
sfr RCAP2H=0xcb;              // 定义特殊功能寄存器 RCAP2H 的地址
sbit kg=P4^3;                 // 定义开关的位地址
sbit fmq=P1^5;                // 定义蜂鸣器的位地址

void delay(int time)          /* 延时子函数，单位为 ms */
{ unsigned char tt;           // 定义辅助无符号字符变量 tt
  while(time!=0){             // 循环执行，次数由变量 time 的值确定
    --time;
    for(tt=0;tt<226;++tt){}   // 内部循环执行 226 次
  }
}

void init_rs232(void)          // 串行口初始化子函数
{ T2CON=0x34;                  // 定时器 2 作为波特率发生器，自动重装
  SCON=0xda;                   // 方式 3，9 位数据，单机通信，允许接收，TB8=1
  RCAP2H=0xff;
}

main()                         // 主函数
{ unsigned char kk=0;          // 定义辅助无符号字符变量 kk
  init_rs232();                // 调用串行口初始化子函数，用于仿真显示
  printf("Hello! I am C51\n"); // 仿真显示字符串"Hello! I am C51"
  printf("I will be your friend!\n"); // 仿真显示"I will be your friend!"
  for(;;){                     // 无限循环
    kk=kk+1;                   // 变量 kk 加 1
    P0=kk;                     // 变量 kk 的值通过 8 个发光二极管显示，“0”——亮
    printf("%bu\n",kk);        // 仿真显示变量 kk 的值
    if(kg==0){ fmq=0; }        // 如果按下开关，蜂鸣器响
    else{ fmq=1; }             // 否则蜂鸣器不响
    delay(1000);               // 调用延时子函数，延时 1000ms
  }
}
```

和上面 C 语言程序对应的汇编语言程序如下。

ORG 0000H;	1——伪指令，表示程序存放的首地址
begin: MOV P0,R1;	2——将 R1 的数据送到 P0 口引脚，输出 0V 时发光二极管点亮
CALL DELAY;	3——调用 0.5s 延时子程序
INC R1;	4——将 R1 的数据加 1
JB 0EBH,ddd;	5——如果未按下开关(P4.4=0EBH)，跳到 ddd
CLR P1.5;	6——按下开关，引脚 P1.5 输出 0V，蜂鸣器响
JMP begin;	7——跳到 begin
ddd: SETB P1.5;	8——未按下开关，引脚 P1.5 输出 5V，蜂鸣器不响
JMP begin;	9——跳到 begin
;	10——延时子程序

DELAY: MOV R5, #04H;	11——R5=延时时间
H0: MOV R6, #0FFH;	12——R6=循环次数
H1: MOV R7, #0FFH;	13——R7=循环次数
H2: DJNZ R7, H2;	14——R7 减 1, 如果不等于 0, 跳到 H2
DJNZ R6, H1;	15——R6 减 1, 如果不等于 0, 跳到 H1
DJNZ R5, H0;	16——R5 减 1, 如果不等于 0, 跳到 H0
RET;	17——子程序结束标志

END ;	18——伪指令, 表示程序结束

通过上述例子, 可知 C 语言程序具有以下基本结构特点。

- (1) 一个 C 语言程序一般由 3 个部分组成: 预处理部分、子函数部分和主函数部分。
- (2) 预处理部分类似于汇编语言的伪指令, 是在 C 语言程序执行前对编译器所下的指示, 主要用于控制编译过程和定义变量。
- (3) 函数的主体必须以大括号 “{}” 包含其中。
- (4) 子函数部分类似于汇编语言的子程序, 完成一些使用较频繁且比较通用的功能。子函数一般由自己编写, 同时 C 语言提供大约 100 多个常用子函数 (常称为库函数)。
- (5) 一个 C 语言程序不论由多少个部分组成, 都有一个且只能有一个 **main** 函数, 即主函数。
- (6) 一个 C 语言程序, 总是从 **main()** 函数开始执行, 而不论其在程序中的位置。当主函数执行完毕时, 即程序执行完毕。
- (7) 如果调用库函数, 需在预处理中进行说明。如 **printf** 为库函数, 其函数声明在文件 **stdio.h** 中, 在预处理中需加入 **#include<stdio.h>** 命令进行说明。
- (8) 每一个说明, 每一个语句都必须以分号结尾。但预处理命令, 函数头和花括号 “{}” 之后不能加分号。
- (9) 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符, 也可不再加空格来间隔。
- (10) C 语言程序注释有两种写法。一种以 “//” 开头, 编译器会把 “//” 之后的文字全部当做注释, 直到此行的尾端。另一种方式是用 “/* */” 的形式, 可以为 C 程序的任一部分做注释, 在 “/*” 开始后, 一直到 “*/” 为止, 它中间的任何内容都被认为是注释, 两者之间不限制行数。
- (11) C 语言程序行的书写格式自由。既允许一行内写几条语句, 也允许一条语句分写在几行上。
- (12) C 语言程序区分字母大小写, 所以不要将字母大小写混用, 如 **main** 不可写成 **MAIN**。

1.3 Keil C51 编译器仿真程序要求

由于 Keil C51 是用于 MCS51 单片机的编程软件, 产生的机器码只能在 MCS51 单片机上运行, 为了便于调试编程, 可利用串行口进行仿真, 因此需对串行口进行初始化, 其程序



一般需具有下列格式。

```
#pragma symbols code           // 定义编译环境
#include<reg52.h>               // 存储器说明库
#include<stdio.h>              // 标准输入、输出库

void init_rs232(void)          // 串行口初始化子函数
{ T2CON=0x34;                 // 定时器 2 作为波特率发生器, 自动重装
  SCON=0xda;                  // 方式 3, 9 位数据, 单机通信, 允许接收, TB8=1
  RCAP2H=0xff; }

main()                         // 主函数
{ 定义各种变量(如 char kk; char mm;)
  .....
  init_rs232();               // 调用串行口初始化子函数
  ...
  编写自己的程序
  ...
}
```

1.4 实践

1.4.1 实践任务

初步了解 C 语言, 学会上机编辑、编译及调试 C 语言程序, 会把 C 语言程序下载到单片机实践板里, 作简单的调试。

1.4.2 实践设备要求

1. 装有 Keil C51 uvision3 集成开发环境和 STC-ISP 的计算机。
2. 单片机实践板(初级)。

1.4.3 实践步骤

1. 打开计算机, 连接计算机实践板。
2. 进入 Keil C51 开发环境, 建立工程项目文件。
3. 建立例 1.1 的 C 语言源程序文件。
4. 编译源程序文件。