



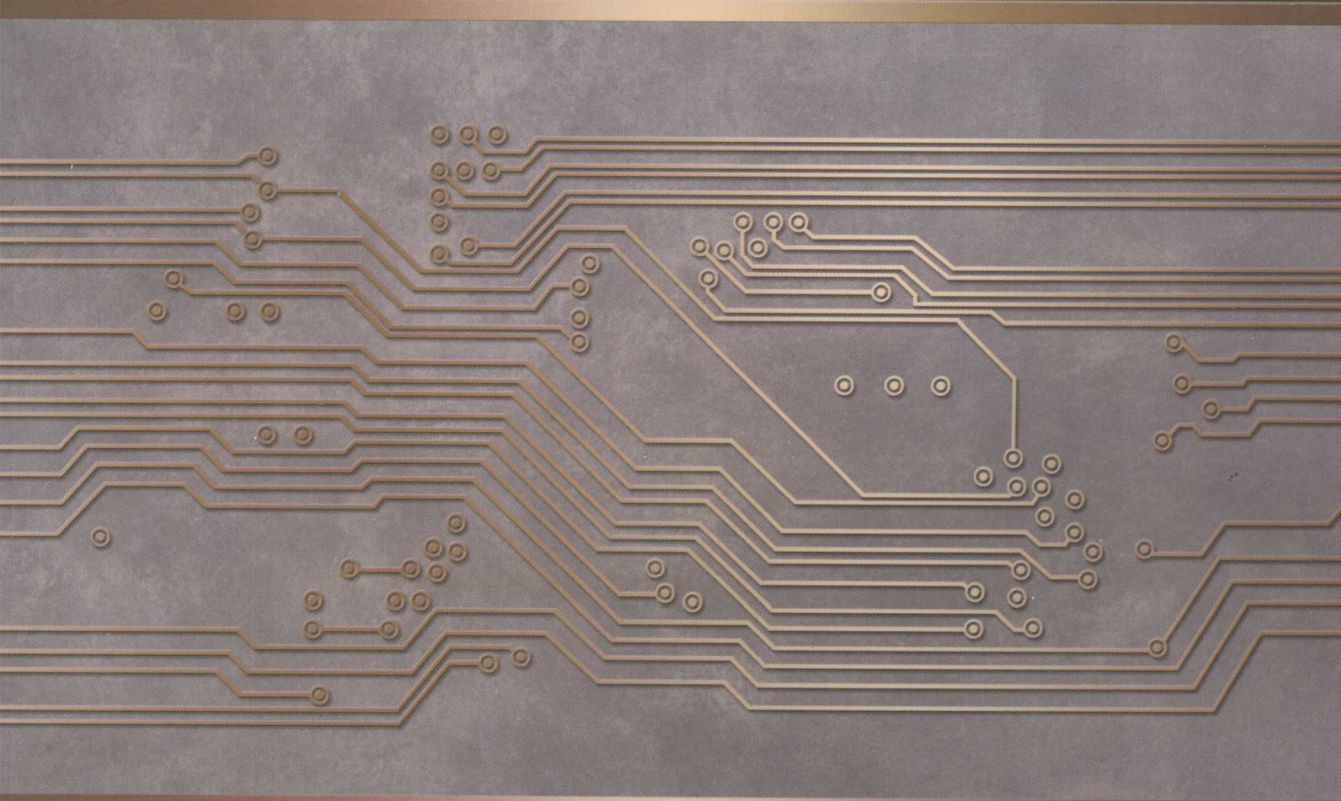
普通高等教育“十一五”国家级规划教材

新编电气与电子信息类本科规划教材

单片机的C语言

程序设计与应用

姜志海 赵艳雷 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

普通高等教育“十一五”国家级规划教材

新编电气与电子信息类本科规划教材

单片机的 C 语言 程序设计与应用

姜志海 赵艳雷 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是普通高等教育“十一五”国家级规划教材。本书以标准 C 语言和 MCS-51 单片机为基础,全面系统地介绍单片机的 C 语言程序设计与应用的基本问题,是一本重在应用、兼顾理论的实用教程。主要内容包括:单片机的 C 语言概述、C51 语言程序设计基础、单片机内部资源和外部扩展资源的 C51 语言编程、 μ Vision2 集成开发环境的使用、MCS-51 系列单片机简介等。本书包含大量单片机的 C 语言设计实例,并提供免费电子课件。

本书可作为高等学校电气与电子信息专业和计算机专业的本专科教材,也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

单片机的 C 语言程序设计与应用 / 姜志海, 赵艳雷编著. —北京: 电子工业出版社, 2008.5
(新编电气与电子信息类本科规划教材)

ISBN 978-7-121-06671-9

I. 单… II. ①姜…②赵… III. 单片微型计算机—C 语言—程序设计—高等学校—教材 IV. TP368.1 TP312

中国版本图书馆 CIP 数据核字(2008)第 068382 号

责任编辑: 王羽佳

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

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: 16 字数: 410 千字

印 次: 2008 年 5 月第 1 次印刷

印 数: 4000 册 定价: 24.00 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 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。

前 言

单片机在嵌入式系统应用中占据很重要的地位。8051 单片机在工业测量、控制领域中得到了广泛的应用。随着国内单片机开发工具研制水平的提高,现在的单片机仿真器普遍支持 C 语言程序的调试,为单片机使用 C 语言提供了便利条件,C 语言已成为举世公认的高效简洁而又贴近硬件的编程语言之一。

本书的主要特色在于,在介绍单片机的 C 语言设计的过程中清晰地说明了单片机的所有功能,并对每项功能给出了实例代码,同时详细介绍了单片机的 C 语言开发与仿真环境的使用,透彻分析了单片机的 C 语言语法和语义,以及开发过程中可能存在的问题和难点。

本书的另一个特点是突破了传统的软、硬件截然隔绝的方法,使读者对嵌入式系统的开发有一个整体的了解。相信本书的这一特点会节省读者进入嵌入式 C 语言领域的时间,同时能够更清楚地认识应用系统开发的过程,深入理解单片机的 C 语言编程机制。

本书从教学和应用的角度出发,结合作者多年来在教学和科研实践中所取得的经验,全面系统地介绍了单片机的 C 语言设计与应用的基本问题,是一本重在原理与应用,兼顾理论的实用教程。本书共分 5 章,主要内容包括:单片机的 C 语言概述、C51 语言程序设计基础、单片机内部资源的 C51 语言编程、单片机外部扩展资源的 C51 语言编程、 μ Vision2 集成开发环境的使用,每章都通过本章小结和习题来巩固所学的知识。全书通过大量实例阐述了单片机的 C 语言设计原理和技巧。

本书免费提供多媒体电子课件和习题解答,请登录华信教育资源网(<http://www.huaxin.edu.cn> 或 <http://www.hxedu.com.cn>)注册下载。

本书是一本专门讲解单片机的 C 语言编程的书,以由浅入深、相互贯穿、重点突出、文字叙述与典型代码实例相结合为原则,向每一位单片机、嵌入式爱好者和开发者全面介绍 C51 语言程序的编写。本书内容系统全面,论述深入浅出,循序渐进,可作为高等学校“单片机 C 语言设计与应用”课程的教学用书,也可以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。

本书第 1、2、4 章由姜志海编写,第 3、5 章由赵艳雷编写,全书由姜志海负责整理、统稿。

本书在编写过程中得到了许多专家和同行的大力支持与热情帮助,他们对本书提出了许多建设性的建议和意见,在此一并表示衷心的感谢。

鉴于编者的水平有限,加之新的编程技术不断涌现,书中难免有不完善之处,恳请广大读者批评指正。反馈信息请发送至 wyj@phei.com.cn。

作 者

目 录

第 1 章 单片机的 C 语言概述	(1)
1.1 C 语言与 MCS-51 单片机	(1)
1.1.1 标准 C 语言的程序结构及特点	(1)
1.1.2 单片机的 C 语言与汇编语言的优势对比	(3)
1.1.3 单片机的 C 语言和标准 C 语言的比较	(7)
1.1.4 单片机的 C 语言的特点	(7)
1.2 Keil C51 开发工具	(8)
1.2.1 Keil 8051 开发工具套件	(8)
1.2.2 使用 Keil 开发工具开发软件的流程	(9)
1.3 C51 语言的程序结构与编程实例	(11)
本章小结	(16)
习题	(17)
第 2 章 C51 语言程序设计基础	(18)
2.1 C51 语言数据类型及存储类型	(18)
2.1.1 数据类型	(18)
2.1.2 常量与变量	(18)
2.1.3 数据存储类型	(23)
2.2 C51 语言对单片机主要资源的控制	(28)
2.2.1 特殊功能寄存器的 C51 语言定义	(29)
2.2.2 绝对地址的访问	(31)
2.2.3 位变量的 C51 语言定义	(34)
2.3 C51 语言的基本运算与流程控制语句	(35)
2.3.1 基本运算	(35)
2.3.2 选择(分支)控制语句	(40)
2.3.3 循环结构流程控制语句	(43)
2.3.4 break 语句、continue 语句和 goto 语句	(48)
2.4 C51 语言的构造数据类型	(50)
2.4.1 数组	(50)
2.4.2 指针	(55)
2.5 C51 语言函数	(59)
2.5.1 函数的基本问题	(59)
2.5.2 中断函数	(65)
2.5.3 变量类型及存储方式	(66)
2.6 C51 语言程序设计的其他问题	(68)
2.6.1 养成好的编程习惯	(68)

2.6.2	宏定义、文件包含与条件编译	(69)
2.6.3	使用移位运算代替乘除和求模运算	(73)
2.6.4	C51 语言的库函数	(74)
2.6.5	汇编语言和 C 语言混合编程	(75)
2.6.6	C51 语言程序设计中容易出错的地方	(80)
(1)	本章小结	(82)
(1)	习题	(83)
第 3 章	单片机内部资源的 C51 编程	(84)
(1)	3.1 单片机的并行口	(84)
(7)	3.1.1 基本知识	(84)
(7)	3.1.2 C51 语言编程实例	(85)
(8)	3.2 单片机的中断系统	(88)
(8)	3.2.1 基本知识	(88)
(9)	3.2.2 C51 语言编程实例	(93)
(11)	3.2.3 C51 语言编程总结	(102)
(10)	3.3 单片机的定时器/计数器	(103)
(7)	3.3.1 基本知识	(103)
(8)	3.3.2 C51 语言编程实例	(108)
(8)	3.3.3 C51 语言编程总结	(121)
(8)	3.4 单片机的串行口	(122)
(8)	3.4.1 基础知识	(122)
(8)	3.4.2 C51 语言编程实例	(127)
(2)	本章小结	(141)
(9)	习题	(142)
第 4 章	单片机外部扩展资源的 C51 编程	(143)
(1)	4.1 并行口扩展的 C51 编程	(143)
(2)	4.1.1 利用锁存器、三态门扩展简单并行口	(143)
(2)	4.1.2 利用串行口扩展并行口	(146)
(0)	4.1.3 利用 8255A 扩展可编程并行口	(148)
(1)	4.2 键盘接口的 C51 语言编程	(159)
(8)	4.2.1 独立式按键	(159)
(0)	4.2.2 矩阵键盘	(162)
(0)	4.3 LED 显示器接口的 C51 编程	(171)
(2)	4.3.1 基本问题	(171)
(0)	4.3.2 LED 静态显示接口	(172)
(0)	4.3.3 LED 动态显示接口	(174)
(0)	4.3.4 串行口控制的 LED 显示接口	(178)
(0)	4.4 键盘与 LED 控制芯片 HD7279A	(179)
(8)	4.4.1 引脚功能及控制指令	(179)
(8)	4.4.2 通信时序	(184)

4.4.3 与单片机的接口	(185)
4.5 模拟量接口的 C51 编程	(191)
4.5.1 8 位 D/A-DAC0832	(191)
4.5.2 12 位 A/D-AD574	(196)
4.5.3 串行 A/D-TLC2543	(199)
4.6 串行 E ² PROM 的 C51 编程	(203)
4.6.1 串行总线及串行 E ² PROM 简介	(203)
4.6.2 在单片机上的应用及程序设计	(205)
本章小结	(208)
习题	(208)
第 5 章 μVision2 集成开发环境的使用	(209)
5.1 μ Vision2 集成开发环境	(209)
5.1.1 μ Vision2 的安装	(209)
5.1.2 μ Vision2 的窗口与菜单	(211)
5.2 用 μ Vision2 建立工程	(217)
5.2.1 工程创建	(217)
5.2.2 工程设置	(218)
5.2.3 编译与链接	(222)
5.3 用 μ Vision2 调试工程	(222)
5.3.1 程序执行与断点设置	(222)
5.3.2 调试窗口	(224)
5.4 C51 程序调试举例说明—HELLO.C	(226)
5.5 Keil C51 的调试技巧	(227)
本章小结	(232)
附录 A MCS-51 系列单片机简介	(233)
A.1 主要功能部件	(233)
A.2 内部结构框图	(233)
A.3 外部引脚说明	(234)
A.4 硬件资源说明	(236)
A.5 CPU	(236)
A.6 存储器结构	(238)
A.7 外部三总线的形成	(241)
附录 B ASCII 码字符表	(242)
附录 C 单片机技术及嵌入式系统常用网站	(243)
参考文献	(244)

第 1 章 单片机的 C 语言概述

我们都知道,在单片机应用系统开发过程中,软件编程占有非常重要的地位。尤其是随着单片机技术的发展,嵌入式系统的推广和应用,硬件的集成化程度越来越高,同时对软件编程的要求也越来越高。这就要求单片机开发人员能在短时间内编写出执行效率高、运行可靠的代码。同时,由于实际系统的日趋复杂,对使用代码的规范性、模块化的要求越来越高,以方便多个工程师以软件工程的形式进行协同开发。在这种形势下,仅靠单片机在推广应用的初期使用的汇编语言来进行软件开发,是远远不够的。

C 语言是近年来在国内外普遍使用的一种程序设计语言。C 语言能直接对计算机硬件进行操作,既有高级语言的特点,又有汇编语言的特点,因此在单片机应用系统开发过程中得到了非常广泛的应用。

本章主要对单片机的 C 语言的基本问题进行概括的说明。

1.1 C 语言与 MCS-51 单片机

嵌入式单片机在推广应用初期的编程语言主要是汇编语言,这是因为当时的开发工具只能支持汇编语言。随着硬件技术的发展,嵌入式单片机开发工具的功能也有很大提高。对于 8051 单片机,有 4 种语言支持,即汇编语言、PL/M 语言、C 语言和 BASIC 语言。汇编语言作为传统的嵌入式系统的编程语言,已经不能满足实际需要了,而 C 语言的结构化和高效性满足了这样的需要,成为电子工程师在进行嵌入式系统编程时的首选语言,并得以广泛应用。尤其是 C 语言编译系统的发展,更加促进了 C 语言的应用。1985 年出现了针对 8051 的 C51 编译器,进而又出现了其他流行的嵌入式处理器系统如 196 系列、PIC 系列、MOTORAL 系列、MSP430 系列、AD 公司和 TI 公司的 DSP 系列的 C 语言编译系统,以及丰富的 C 语言库函数。

C 语言是一种通用的程序设计语言,其代码效率高、数据类型及运算符丰富,并具有好的程序结构,适用于各种应用程序设计。

鉴于嵌入式系统低端应用的巨大市场以及 8 位机具有的巨大潜力,可以预期在未来相当长的时间内,8 位机仍然是嵌入式应用中的主流机型。所以本书主要讨论 8 位嵌入式单片机——MCS-51 单片机及其派生产品的 C 语言编程问题,简称 C51 的程序设计。

单片机的 C 语言是以标准 C 语言为基础的,所以本节主要对 C51 语言、标准 C 语言、单片机汇编语言进行比较。

1.1.1 标准 C 语言的程序结构及特点

1. 标准 C 语言的程序结构

标准 C 语言程序采用函数结构,每个 C 语言程序由一个或多个函数组成。在每个程序中都包含一个主函数 main(),不管这个主函数 main()位于何处,程序总是从 main()函数开始执

行, 执行到 `main()` 函数结束时结束。在 `main()` 函数中可以调用其他函数, 其他函数也可以互相调用。但 `main()` 函数只能调用其他功能函数, 而不能被其他函数调用。功能函数可以是标准 C 语言提供的库函数, 也可以是用户自定义的函数。在编制标准 C 语言程序时, 程序的开始部分一般是预处理命令、函数说明和变量定义等。

标准 C 语言程序结构如下:

```
#include < > /*预处理命令*/
long fun1(); /*函数说明*/
float fun2();
int x,y;
float z;
fun1() /*功能函数 1*/
{
    ...
}
main() /*主函数*/
{
    ...
}
fun2() /*功能函数 2*/
{
    ...
}
```

函数由函数定义和函数体两部分组成。

函数定义部分包括函数类型、函数名、形式参数说明等, 函数名后面必须跟一个圆括号 `()`, 形式参数在 `()` 内定义。

由一对大括号 `{}` 将函数体的内容括起来, 如果一个函数内有多个大括号, 则最外层的一对大括号为函数体范围。函数体内包含若干语句, 一般由两部分组成: 声明语句和执行语句。声明语句用于对函数中用到的变量进行定义, 也可以对函数体中调用的函数进行声明; 执行语句由若干语句组成, 用来完成一定的功能。

标准 C 语言程序在书写时格式十分自由, 一条语句可以写成一行, 也可以写成几行, 还可以在一行内写多条语句, 但每条语句后面必须以分号 “;” 作为结束符。

标准 C 语言程序对大小写字母敏感。在程序中, 对于同一个字母的大小写, 系统是作不同处理的。

在程序中可以用 “/* */” 或 “//” 对标准 C 语言程序中的任何部分作注释, 以增加程序的可读性。

2. 标准 C 语言的特点

与其他高级语言相比, 标准 C 语言具有以下特点。

(1) 语言简洁、紧凑, 使用方便、灵活

标准 C 语言共有 32 个关键字、9 种控制语句, 程序书写形式自由, 与其他高级语言相比, 程序更加简练。

(2) 运算符、表达式丰富

标准 C 语言包括 34 种运算符, 而且把括号、赋值、强制类型转换等都作为运算符处理。表达式灵活多样, 可以实现各种各样的运算。

(3) 数据结构丰富, 具有现代化语言的各种各样的数据结构

标准C语言的数据类型有整型、实型、字符型、数组类型、指针类型等, 并能用来实现各种复杂的数据结构。

(4) 可进行结构化程序设计

标准C语言具有各种结构化的程序语句, 如if-else语句、while语句、do-while语句、switch语句、for语句等。另外, 标准C语言程序以函数为模块单位, 一个标准C语言程序由许多个函数组成, 一个函数相当于一个程序模块, 因此标准C语言程序可以很容易地进行结构化程序设计。

(5) 可以直接对计算机硬件进行操作

标准C语言允许直接访问物理地址, 能进行位操作, 能实现汇编语言的大部分功能, 可以对硬件直接进行操作。

(6) 生成的目标代码质量高, 程序执行效率高

众所周知, 汇编语言生成的目标代码的效率是最高的。但据统计表明, 对于同一个问题, 用C语言编写的程序生成目标代码的效率仅比汇编语言编写的程序低10%~20%。而用C语言编写程序比用汇编语言编写程序方便和容易得多, 程序可读性强, 开发时间也短得多。

(7) 可移植性好

不同的计算机汇编指令不一样, 用一种机型的汇编语言编写的程序用于另外的机型时, 必须改写成对应机型的指令代码。而标准C语言编写的程序基本不用修改就可以用于各种机型和各类操作系统。

1.1.2 单片机的C语言与汇编语言的优势对比

在国内, 汇编语言在单片机开发过程中是比较流行的开发工具。长期以来对编译效率的偏见, 以及不少程序员对使用汇编语言开发硬件环境的习惯性, 使C语言在很多地方遭到冷落。优秀的程序员写出的汇编语言程序的确有执行效率高的优点, 但汇编语言其可移植性和可读性差的特点, 使其开发出来的产品在维护和功能升级时的确有极大的困难, 从而导致整个系统的可靠性和可维护性比较差。而使用C语言进行嵌入式系统的开发, 有着汇编语言所不可比拟的优势。

① 编程调试灵活方便

C语言编程灵活, 当前几乎所有的嵌入式系统都有相应的C语言级别的仿真调试系统, 调试十分方便。

② 生成的代码编译效率高

当前较好的C编译系统编译出来的代码效率只比直接使用汇编语言低20%, 如果使用优化编译选项甚至可以更低。

③ 模块化开发

目前的软硬件开发都向模块化、可复用性的目标集中。不管是硬件还是软件, 都希望其有比较通用的接口, 在以后的开发中如果需要通过实现相同或者相近的功能, 就可以直接使用以前开发过的模块, 尽量不做或少做改动, 以减少重复劳动。如果使用C语言开发, 数据交换可以方便地通过约定实现, 有利于多人协同进行大项目的合作开发。同时C语言的模块化开发方式使开发出来的程序模块可以不经修改而直接被其他项目使用, 这样就可以很好地利用已有的大量C语言程序资源与丰富的库函数, 从而最大程度地实现资源共享。

④ 可移植性好

由于不同系列的嵌入式系统的 C 语言编译工具都是以标准 C 语言作为基础进行开发的,因此一种 C 语言环境下所编写的 C 语言程序,只需要将部分与硬件相关的地方和编译链接的参数进行适当修改,就可以方便地移植到另一种系列上。例如,在 C51 下编写的程序通过改写头文件及少量的程序行,就可以方便地移植到 196 或 PIC 系列上。也就是说,基于 C 语言环境下的嵌入式系统能基本达到平台的无关性。

⑤ 便于项目的维护

用 C 语言开发的代码便于开发小组计划项目、灵活管理、分工合作及后期维护,基本可以杜绝因开发人员变化而给项目进度、后期维护或升级所带来的影响,从而保证整个系统的品质、可靠性及可升级性。

下面通过一个例子对汇编语言和 C 语言进行比较。

【例 1-1】 将外部数据存储器的 000BH 和 000CH 单元的内容相互交换。

用汇编语言编程,程序如下:

```
ORG      0000H
MOV      DPTR,#000BH
MOVX    A,@DPTR      ;将 000BH 内容读入 A
MOV      R7,A        ;暂存 000BH 内容
INC      DPTR
MOVX    A,@DPTR      ;将 000CH 内容读入 A
MOV      DPTR,#000BH
MOVX    @DPTR,A
INC      DPTR
MOV      A,R7
MOVX    @DPTR,A
SJMP    $
END
```

用 C 语言编程, C51 程序如下:(注: C51 语言对地址的指示可以采用指针变量,也可以引用头文件 `absacc.h` 作为绝对地址访问。下面的程序采用绝对地址访问方法。)

```
#include<absacc.h>
void main(void)
{   char c;
    do
    {   c=XBYTE[11];
        XBYTE[11]=XBYTE[12];
        XBYTE[12]=c;
    }while(1);
}
```

上面的程序经过编译,生成的反汇编程序如下:

```
0000 LJMP    0014H
```

```

0003 MOV     DPTR,#000BH
0006 MOVX   A,@DPTR
0007 MOV     R7,A
0008 INC     DPTR
0009 MOVX   A,@DPTR
000A MOV     DPTR,#000BH
000D MOVX   @DDPTR,A
000E INC     DPTR
000F MOV     A,R7
0010 MOVX   @DPTR,A
0011 SJMP   0003H
0013 RET
0014 MOV     R0,#7FH
0016 CLR     A
0017 MOV     @R0,A
0018 DJNZ   R0,0017H
001A MOV     SP,#07H
001D LJMP   0003H

```

对照 C 语言编写的源程序与反汇编程序，可以看出：

① 进入 C 语言程序后，首先将 RAM 地址为 00~7FH 的 128 个单元清零，然后置 SP 为 07，因此如果要对内部 RAM 置初值，一定是在执行了一条 C 语言语句之后；

② 对于 C 语言程序设定的变量，C51 编译器自行安排寄存器或存储器作为参数传递区，通常在 R0~R7（一组或两组，根据参数多少而定），因此如果对具体地址置数，应避免 R0~R7 这些地址；

③ 如果不特别指定变量的存储类型，变量通常被安排在内部 RAM 区。

下面再给出几个 C 语言和汇编语言对照的例子。

【例 1-2】 二进制数转换成十进制数 (BCD 码)。

将累加器 A 中给定的二进制数，转换成 3 个十进制数 (BCD 码)，并存入 Result 开始的 3 个单元。

汇编语言程序如下：

```

Result      EQU      20H
              ORG      0000H
              LJMP     START
BINTOBCD:   MOV      B,#100          ;设置转换子程序
              DIV      AB
              MOV      Result,A      ;除以 100 得百位数
              MOV      A,B
              MOV      B,#10
              DIV      AB
              MOV      Result+1,A    ;余数除以 10 得十位数
              MOV      Result+2,B    ;余数为个位数

```

```

RET
START: MOV SP,#60H ;主程序
MOV A,#123
LCALL BINTOBCD
SJMP $
END

```

用 C 语言编程, C51 程序如下:

```

void main(void)
{
    unsigned char Result[3];
    unsigned char Number;
    Number=123;
    Result[0]=Number/100; //除以 100 得百位数
    Result[1]=(Number%100)/10; //余数除以 10 得十位数
    Result[2]=Number%10; //余数为个位数
}

```

【例 1-3】 二进制数转换成 ASCII 码程序。

将累加器 A 中的内容分为两个 ASCII 码, 并存入 Result 开始的两个单元。

汇编语言程序如下:

```

Result EQU 20H
ORG 0000H
LJMP START
ASCIIITAB: DB '0123456789ABCDEF'
BINTOHEX: MOV DPTR,#ASCIIITAB
MOV B,A
SWAP A
ANL A,#0FH ;取 A 的高 4 位
MOVC A,@A+DPTR
MOV Result,A
MOV A,B
ANL A,#0FH ;取 A 的低 4 位
MOVC A,@A+DPTR
MOV Result+1,A
RET
START: MOV SP,#40H
MOV A,#01000011B
LCALL BINTOHEX
SJMP $
END

```

用 C 语言编程, C51 程序如下:

```
code unsigned char ASCIIITAB[16]="0123456789ABCDEF";
```

```
void main(void)
{
    unsigned char Result[2];
    unsigned char Number;
    Number=0x1a;
    Result[0]=ASCIITAB[Number/16];           //高4位
    Result[1]=ASCIITAB[Number&0x0f];       //低4位
}
```

1.1.3 单片机的C语言和标准C语言的比较

标准C语言，或称为ANSI C语言。单片机的C语言和标准C语言之间有许多相同的地方，但也有其自身的一些特点。不同的嵌入式C语言编译系统之所以与ANSI C语言有不同的地方，主要是由于它们所针对的硬件系统不同，对于MCS-51系列单片机，我们称为C51语言，其不同点主要体现在以下几方面。

① C51语言中定义的库函数和标准C语言定义的库函数不同。标准C语言定义的库函数是按照通用微型计算机来定义的，而C51语言中的库函数是按MCS-51单片机的应用情况来定义的。

② C51语言中的数据类型和标准C语言的数据类型也有一定的区别。在C51语言中增加了几种针对MCS-51单片机特有的数据类型。例如，MCS-51系列器件包含位操作空间和丰富的位操作指令，因此C51语言与ANSI C语言相比要多一种位类型，使得它能同汇编语言一样，灵活地进行位指令操作。

③ C51变量的存储模式与标准C语言中变量的存储模式不一样。C51语言中变量的存储模式与MCS-51单片机的存储器紧密相关。从数据存储类型上，8051系列有片内、片外程序存储器，片内、片外数据存储器。在片内程序存储器还分为直接寻址区和间接寻址区，分别对应code、data、xdata、idata，以及根据51系列特点而设定的pdata类型，使用不同存储器将会影响程序执行的效率，不同的模式对应不同的硬件系统和不同的编译结果。ANSI C语言对存储模式要求不高。

④ C51语言与标准C语言的输入/输出处理不一样。C51语言中的输入/输出是通过MCS-51串行口来完成的，输入/输出指令执行前必须对串行口进行初始化。

⑤ C51语言与标准C语言在函数使用方面有一定的区别。C51语言中有专门的中断函数。

1.1.4 单片机的C语言的特点

单片机的C语言的特点主要体现在以下几个方面：

- ① 无须了解机器硬件及其指令系统，只需初步了解MCS-51的存储器结构；
- ② C51语言能方便地管理内部寄存器的分配、不同存储器的寻址和数据类型等细节问题，但对硬件控制有限，而汇编语言可以完全控制硬件资源；
- ③ C51语言在小应用程序中产生的代码量大，执行速度慢，但在较大的程序中代码效率高；
- ④ C51语言程序由若干函数组成，具有良好的模块化结构，便于改进和扩充；

⑤ C51 语言程序具有良好的可读性和可维护性，而汇编语言在大应用程序开发中，开发难度增加，可读性差；

⑥ C51 语言有丰富的库函数，可以大大减少用户的编程量，显著缩短编程与调试时间，大大提高软件开发效率；

⑦ 使用汇编语言编制的程序，当机型改变时，无法直接移植使用，而 C 语言程序是面向用户的程序设计语言，能在不同机型的机器上运行，可移植性好。

1.2 Keil C51 开发工具

Keil Software 公司推出的 51 系列兼容单片机 C 语言软件开发系统，即 8051 开发工具套件，具有编辑、编译、链接、定位、库管理、HEX 文件创建、仿真调试等多种功能，并支持汇编、PL/M 等多种编程语言。

Keil 8051 开发软件有测试版和正式版两种版本，测试版包括 8051 开发工具的测试版本和非完整的用户手册，可以用于产生目标代码小于 2KB 的应用，该版本主要用来评估软件的有效性，用户可免费使用。正式版包括没有限制的 8051 工开发具和全套用户手册，该版本有一年的免费技术支持和产品升级。

1.2.1 Keil 8051 开发工具套件

1. Keil 8051 开发工具套件主要包含以下开发部件

- ① Windows 应用程序 μ Vision2：一个集成开发环境，它把项目管理、源代码编辑、程序调试等集成到一个功能强大的环境中。
- ② 美国标准优化 C 语言交叉编译器 C51：从 C 语言源代码产生可重定位的目标文件。
- ③ 宏汇编器 A51：从 8051 汇编源代码产生可重定位的目标文件。
- ④ 链接/重定位器 BL51：组合由 C51 和 A51 产生的可重定位的目标文件生成绝对目标文件。
- ⑤ 库管理器 LIB51：组合目标文件生成可以被链接器使用的库文件。
- ⑥ 目标文件到 HEX 格式的转换器 OH51：从绝对目标文件创建 Intel HEX 格式的文件。
- ⑦ 实时操作系统 RTX51：简化了复杂和对时间要求敏感的软件项目。

2. Keil 8051 开发套件

针对不同层次的专业用户，Keil 公司提供了不同的 8051 开发套件，不同的开发套件所包含的开发部件有所不同，下面简要描述各套件及其包含的部件及内容。

- ① PK51 专业开发套件：包括专业开发人员创建和调试复杂 8051 嵌入式应用系统所要用到的一切工具，可以针对所有 8051 及其派生系列进行配置使用。
- ② DK51 开发套件：是 PK51 专业开发套件的精简版本，不包括小型 RTX51 实时操作系统。此套件可以针对所有的 8051 及其派生系列进行配置使用。
- ③ CA51 编译套件：是那些需要 C 语言编译器而不需要调试系统的开发人员的最好选择，CA51 开发包仅仅包含 μ Vision2 IDE，不包含 μ Vision2 调试器。可以针对所有的 8051 及其派生系列进行配置使用。
- ④ A51 汇编套件：包括一个汇编器和用户创建嵌入式应用所需要的所有功能，此套件

可以针对所有的 8051 及其派生系列进行配置使用。

⑤ RTX51 实时操作系统 FR51: 是一个 8051 系列单片机的实时内核。RTX51 FULL 提供 RTX51 TINY 的所有功能和一些扩展功能, 并且包括 CAN 通信协议接口子程序。

表 1-1 总结了不同开发套件所包含的工具部件。

表 1-1 不同开发套件所包含的工具部件

部 件	PK51	DK51	CA51	A51	FR51
μVision2项目管理器和编辑器	√	√	√	√	
A51汇编器	√	√	√	√	
C51编译器	√	√	√		
BL51链接/定位器	√	√	√	√	
LIB51库管理器	√	√	√	√	
μVision2 调试器/模拟器	√	√			
RTX51 Tiny	√				
RTX51 Full					√

1.2.2 使用 Keil 开发工具开发软件的流程

1. 开发流程

使用 Keil Software 工具时, 用户的项目开发流程和其他软件开发项目的流程极其相似, 主要包括以下步骤:

- ① 创建一个项目, 从器件库中选择目标器件并配置工具软件
- ② 用 C 语言或汇编语言创建源程序;
- ③ 用项目管理器生成用户的应用;
- ④ 修改源程序中的错误;
- ⑤ 调试链接后的应用。

2. 8051 工具集框图

一个完整的 8051 工具集的框图可以很好地表述此开发流程, 如图 1-1 所示。每一组件的描述如下。

(1) Keil μVision2 IDE

μVision2 IDE 将项目管理器和具有错误提示功能的编辑器、选项设置、制作工具(MAKE)以及在线帮助等集成在一起。

利用 μVision2 可以创建用户的源代码, 并把它们组织到一个能确定目标应用的项目中。μVision2 能自动编译、汇编、链接用户的嵌入式应用。

(2) C51 编译器和 A51 汇编器

由 μVision2 IDE 创建的源代码, 被 C51 编译器编译或 A51 汇编器汇编后, 生成可重定位的目标文件。

Keil C51 编译器完全遵照 ANSI C 语言标准, 支持 C 语言的所有标准特性, 另外还增加了直接支持 8051 结构的几个特性。

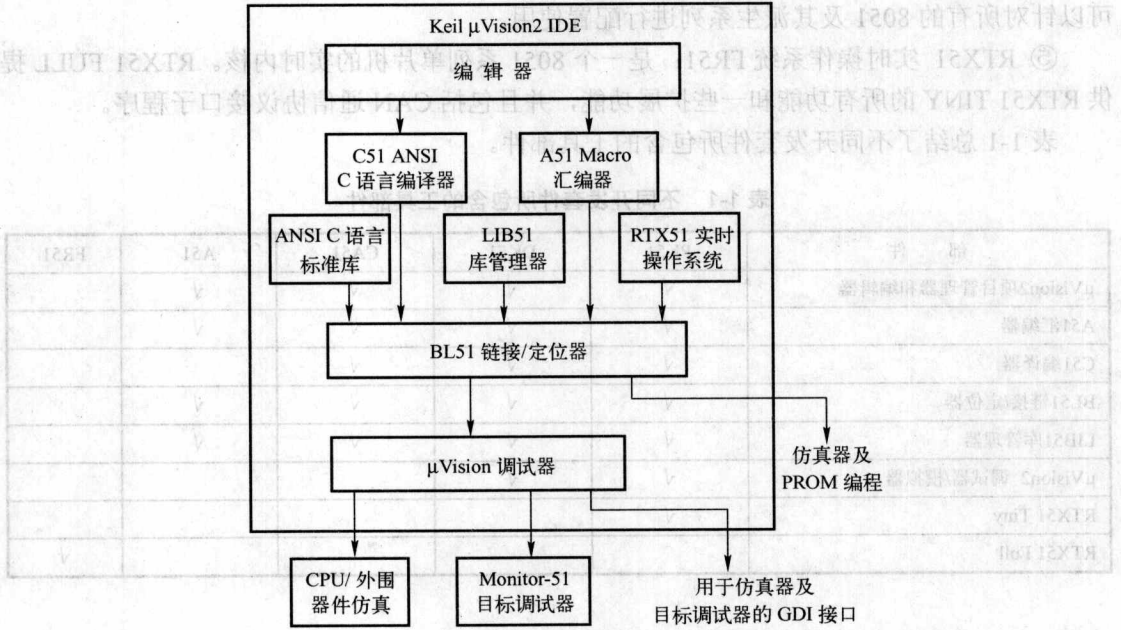


图 1-1 使用 μVision2 开发软件流程

Keil A51 宏汇编器支持 8051 及其派生系列的全部指令集。

(3) LIB51 库管理器

LIB51 库管理器允许用户从由编译器或汇编器生成的目标文件创建目标库。库是一种被特别地组织过并在以后可以被链接重用的对象模块。当链接器处理一个库时，仅仅使用库中用于创建程序的目标模块。

(4) BL51 链接/定位器

BL51 链接/定位器利用从库中提取的目标模块和由编译器或汇编器生成的目标模块，创建一个绝对地址的目标模块。一个绝对地址目标模块或文件包含不可重定位的代码和数据。所有代码和数据被安置在固定的存储器单元中。绝对地址目标文件可以用于：

- 写入 EPROM 或其他存储器件；
- 由 μVision2 调试器使用来模拟和调试；
- 由仿真器来测试程序。

(5) μVision2 调试器

μVision2 源代码级调试器包含一个高速模拟器，能够让用户模拟整个 8051 系统，包括片上外围器件和外部硬件。当用户从器件库中选择器件时，器件的特性将自动被配置。

μVision2 调试器为用户在实际目标板上调试程序提供了几种方法：

- 安装 MON51 目标监控器到用户的目标系统，并通过 Monitor-51 接口下载程序；
- 利用高级 GDI (AGDI) 接口，把 μVision2 调试器绑定到用户的目标系统。

(6) Monitor-51 目标调试器

μVision2 调试器支持使用 Monitor-51 进行目标调试。此监控程序驻留在用户目标板的存储器中，它利用串口和 μVision2 调试器进行通信。μVision2 调试器可以对用户目标硬件实行源代码级的调试。