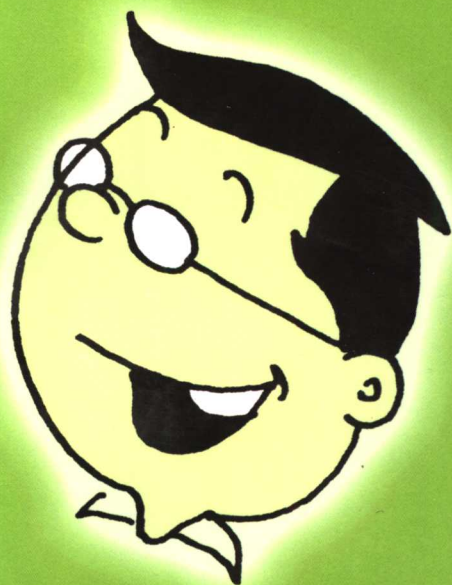
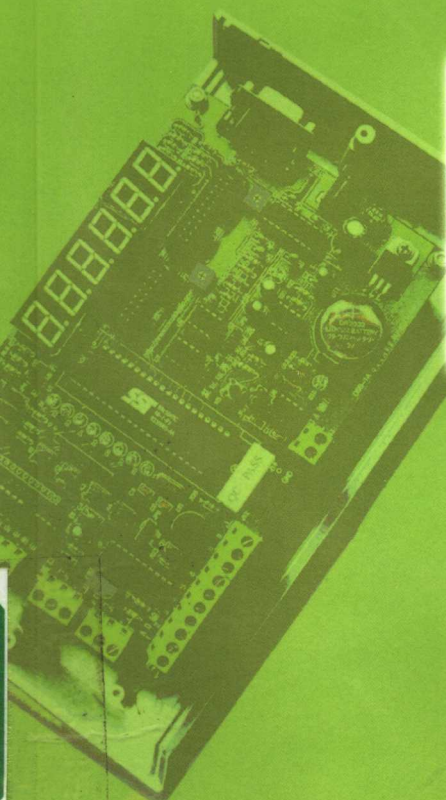




单片机C语言 轻松入门

周 坚 编著



 北京航空航天大学出版社



TP368.1
271D

单片机 C 语言轻松入门

周 坚 编著

北京航空航天大学出版社

内 容 简 介

本书以 80C51 单片机的 C 语言为例来学习使用 C 语言进行单片机程序开发,介绍了 C 语言的基础知识、Keil 软件的使用、程序的编写与调试方法及其他相关知识。

随书光盘提供了作者所设计的实验仿真板、书中所有的例子、实验过程及现象的动画等。因此,读者获得的不仅是一本文字教材,更是一个完整的学习环境。

本书结合了作者多年教学、科研实践所获取的经验,在单片机 C 语言课程教学改革基础上,融入了教学改革的成果而编写。它依据学习者的认知规律来编排内容,充分体现了以人为本的指导思想。

本书可作为中等职业学校、高等职业学校、电视大学等的教学用书,也是单片机爱好者自学单片机 C 语言合适的教材。

图书在版编目(CIP)数据

单片机 C 语言轻松入门/周坚编著. — 北京:北京航空航天大学出版社, 2006. 7

ISBN 7-81077-836-6

I. 单… II. 周… III. ①单片微型计算机—程序设计②C 语言—程序设计 IV. ①TP368.1②TP312

中国版本图书馆 CIP 数据核字(2006)第 060636 号

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!

单片机 C 语言轻松入门

周 坚 编著

责任编辑 李文轶

*

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(100083) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn> E-mail: bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:18.5 字数:474 千字

2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-836-6 定价:30.00 元(含光盘)

前言

随着单片机开发技术的不断发展,目前已有越来越多的人从普遍使用汇编语言过渡到逐渐使用高级语言进行开发,其中以 C 语言为主,而且市场上几种常见的单片机均有其 C 语言开发环境。本书将以最为流行的 80C51 单片机为例来学习单片机的 C 语言编程技术。

本书编写之前,作者在多年教学、科研实践以及对单片机课程进行教学改革的基础上编写过一本《单片机轻松入门》,受到广大读者的欢迎。可见,它在较短时间内被多次重印表明它的确起到了“轻松入门”的作用。本书延续《单片机轻松入门》的风格,带领读者“轻松入门”,来学习 C 语言的基础知识、Keil 软件的使用、单片机的 C 语言开发等其他相关知识,而最终掌握用 C 语言编写程序。

本书按 C 语言体系结构来编排。其中,每一章内容的安排上采用“以任务为中心”的方式,即将 C 语言编程所需的基本知识,如 C 语言中的变量、常量、保留字、程序结构、运算符、表达式等知识,结合单片机的结构特点和 Keil 软件的使用方法等,通过一系列的“任务”进行介绍。每个“任务”都包括一些 C 语言的知识、Keil 软件的使用、程序调试方法、单片机结构方面及单片机开发中必须了解的知识。由于每个任务都是易于完成的,只要完成这一任务即可掌握上述各知识点。因此,对于一个已有一定汇编程序编写经验的单片机程序员而言,甚至在学完第 1 章后,就可以尝试用 C 语言来改写原来编写的程序,并取得一些成绩。对于初学者来说,则可以同步学到单片机结构、C 语言编程及 Keil 软件使用等各方面的知识。

本书共分 11 章。其中:

第 1 章是单片机的 C 语言概述,通过本章的学习可了解 C 语言的基本知识,并能够识读一些 C 语言源程序。

第 2 章介绍如何建立单片机的 C 语言学习环境。对于 C 语言学习而言,一个可供练习的环境非常重要;它不同于 PC 机上所用的 C 语言,除了软件实验环境外,单片机的 C 语言学习还要求有硬件实验环境。本章介绍的是一个具有可操作性的软、硬件实验环境。

第 3 章~第 5 章介绍 C 语言的数据类型、程序结构和数据类型构造,它是 C 语言最基本的知识,掌握之后即可进行常用程序的编写工作。

第 6 章介绍单片机内部结构的编程知识。介绍了 80C51 单片机内部常用的“中断”、“定时器”和“串行口”的编程方法。

第 7 章介绍函数及其相关知识。介绍了函数定义、函数调用及全局变量与局部变量、变量的存储方式等内容。

第 8 章介绍常用单片机接口的 C 语言编程,安排了键盘、LED 显示器、 I^2C 接口、SPI 接口、A/D 转换、D/A 转换、液晶显示器等内容的 C 语言编程实例。通过这些实例可掌握常用外围电路的 C 语言编程方法,增强读者的实际应用能力。

第 9 章是应用实例,引导读者从入门到开发。介绍了若干个简单但比较全面的程序,读者可以利用它们来做一些比较完整的“产品”,以了解使用 C 语言开发项目的完整过程。

第 10 章是 RTOS 介绍,通过对 Keil 自带的 Rtx51 Tiny 介绍来学习目前热门的 RTOS 知识,了解 RTOS 的常识,为读者学习更多的 RTOS 技术打下基础。

第 11 章介绍 Keil C 语言的函数,介绍了 Keil 软件提供的函数。所介绍的函数取自于 Keil 的数据手册。为便于调试,作者为各函数编写了测试用的主函数,提供了函数在 Keil 软件下的测试方法。

C 语言的语法并不难学,使用也不困难,但问题在于不知在什么场合运用。因此,本书尽可能为每一种知识点找到工程实际中的应用实例,以便读者更好地理解相关知识,并尽快将其应用到自己的开发实践中。

使用 C 语言进行嵌入式开发是实践性很强的学科,必须通过较多的实践操作才能学好这门课程。本书编写时考虑到了读者的实际情况,即读者不能随时有老师的指导,而是基于自力更生。因此书中不仅使用文字对有关实验过程进行细致的介绍,而且还在附带的光盘中大量应用动画形式提供实验过程和效果以供参考,对于部分内容还提供完整操作过程的动画记录,以保证读者可以无师自通。

作者为本书的写作开发了实验仿真板,设计了实验电路板。随书光盘提供了作者所设计的实验仿真板、书中所有例子以及记录使用实验仿真板实验过程的动画等。读者获得的不仅是一本文字教材,更是一个完整的学习环境。

本书安排的例子大部分是由作者编写的,部分是参考其他资料改写而成,全部程序都由作者调试并通过。对于例子的使用说明也尽量详细,力争让读者“看则能用,用则能成”,以保证读者在动手的过程中常体会到成功的乐趣,而不是常遇挫折。

本书在提供文字教材的同时还通过平凡单片机工作室(<http://www.mcustudio.com>)为广大读者提供服务,欢迎读者参与探讨。

周 坚

2006 年 2 月于江苏

目 录

第 1 章 单片机 C 语言概述

1.1 C 语言简介	1
1.1.1 C 语言的产生与发展	1
1.1.2 C 语言的特点	1
1.2 C 语言的入门知识	3
1.2.1 简单的 C 程序介绍	3
1.2.2 C 程序特性的分析	8

第 2 章 单片机 C 语言开发环境的建立

2.1 Keil 软件简介	10
2.2 Keil 软件的安装	11
2.3 Keil 软件的使用	12
2.3.1 源文件的建立	13
2.3.2 工程的建立	13
2.3.3 工程的设置	16
2.3.4 编译、链接	20
2.4 实验仿真板简介与使用	22
2.4.1 实验仿真板的安装	23
2.4.2 实验仿真板的使用	23
2.5 硬件实验环境的建立	24
2.5.1 实验电路板简介	24
2.5.2 实验电路板的硬件结构	25
2.5.3 实验电路板的基本使用方法	27

第 3 章 数据的类型、运算符与表达式

3.1 数据类型概述	30
3.2 常量与变量	30
3.2.1 常 量	31

3.2.2	变 量	32
3.3	整型数据	33
3.3.1	整型常量	33
3.3.2	整型变量	34
3.4	字符型数据	35
3.4.1	字符型常量	35
3.4.2	字符型变量	36
3.5	数的溢出	37
3.6	实型数据	39
3.6.1	实型常量	39
3.6.2	实型变量	39
3.7	Keil 特有的数据类型	43
3.7.1	位型数据	43
3.7.2	sfr 型数据	43
3.8	80C51 中数据的存储位置	44
3.8.1	程序存储器	44
3.8.2	内部数据存储器	46
3.9	变量赋初值	52
3.10	算术运算符和算术表达式	53
3.10.1	C 运算符简介	53
3.10.2	算术运算符和算术表达式	54
3.10.3	各类数值型数据间的混合运算	54
3.10.4	赋值运算符和赋值表达式	55
3.10.5	逗号运算符和逗号表达式	60
3.10.6	位操作运算符及其表达式	60
3.10.7	自增减运算符、复合运算符及其表达式	62

第 4 章 C51 流程与控制

4.1	顺序结构程序	64
4.2	选择结构程序	64
4.2.1	引 入	64
4.2.2	关系运算符和关系表达式	67
4.2.3	逻辑运算符和逻辑表达式	68
4.2.4	选择语句 if	69
4.2.5	if 语句的嵌套	71
4.2.6	条件运算符	72
4.2.7	switch/case 语句	73
4.3	循环结构程序	75
4.3.1	循环程序简介	76

4.3.2	while 循环语句	76
4.3.3	do-while 循环语句	77
4.3.4	for 循环语句	78
4.3.5	break 语句	80
4.3.6	continue 语句	81
4.4	仿真型实验板的使用	82
4.4.1	仿真型实验板与计算机的连接	83
4.4.2	程序的调试	86

第 5 章 C51 构造数据类型

5.1	数 组	89
5.1.1	引 入	89
5.1.2	一维数组	90
5.1.3	二维数组	92
5.1.4	字符数组	93
5.1.5	数组与存储空间	94
5.2	指 针	94
5.2.1	指针的基本概念	95
5.2.2	指针变量的定义	96
5.2.3	指针变量的引用	98
5.2.4	Keil C51 的指针类型	102
5.3	结 构	104
5.3.1	结构的定义和引用	105
5.3.2	结构数组	108
5.4	共用体(union)	109
5.5	枚举(enum)	111
5.5.1	枚举的定义和说明	111
5.5.2	枚举变量的取值	112
5.6	用 typedef 定义类型	114

第 6 章 单片机内部资源的编程

6.1	中断编程	116
6.1.1	80C51 的中断请求源	116
6.1.2	中断源的自然优先级与中断服务程序入口地址	117
6.1.3	80C51 的中断控制	118
6.1.4	中断程序的编写	119
6.1.5	寄存器组的切换	121
6.2	定时器/计数器	122
6.2.1	定时器/计数器的基本结构及工作原理	122

目 录

6.2.2	定时器/计数器的控制字	122
6.2.3	定时器/计数器的4种工作方式	124
6.2.4	定时器/计数器的定时/计数方式下初值的计算	125
6.2.5	定时器/计数器的编程	126
6.3	串行口编程	132
6.3.1	串行口控制寄存器	132
6.3.2	串行口工作方式	133
6.3.3	硬件练习	138
第7章 函 数		
7.1	函数的概述	140
7.2	函数的定义	142
7.3	函数参数和函数的值	144
7.4	函数的调用	147
7.4.1	函数调用的一般形式	147
7.4.2	函数调用的方式	147
7.4.3	调用函数的声明和函数原型	147
7.4.4	函数的嵌套调用	149
7.4.5	函数的递归调用	150
7.4.6	C51函数的重入	152
7.4.7	用函数指针变量调用函数	154
7.5	数组作为函数参数	155
7.6	局部变量和全局变量	156
7.6.1	局部变量	157
7.6.2	全局变量	157
7.7	变量的存储类别	160
7.7.1	动态与静态存储方式	160
7.7.2	auto变量	160
7.7.3	static变量	161
7.7.4	用extern声明外部变量	161
第8章 单片机接口的C语言编程		
8.1	LED数码管	164
8.1.1	静态显示接口	164
8.1.2	动态显示接口	167
8.2	键 盘	170
8.2.1	键盘工作原理	170
8.2.2	键盘与单片机的连接	171
8.3	I ² C总线接口	173

8.3.1	I ² C 接口	173
8.3.2	24 系列 EEPROM 的结构及特性	174
8.3.3	24 系列 EEPROM 的使用	176
8.4	X5045 的使用	180
8.4.1	SPI 串行总线简介	180
8.4.2	X5045 的结构和特性	180
8.4.3	X5045 的使用	182
8.5	模/数转换接口	187
8.6	数/模转换接口	189
8.7	液晶显示器接口	191
8.7.1	字符型液晶显示器的基本知识	191
8.7.2	字符型液晶显示器的使用	192

第 9 章 应用设计举例

9.1	秒 表	197
9.2	可预置的倒计时钟	200
9.3	AT24C01A 的综合应用	204
9.3.1	功能描述	205
9.3.2	实例分析	205
9.3.3	实例应用	209
9.4	X5045 的综合应用	209
9.4.1	功能描述	209
9.4.2	实例分析	210
9.4.3	实例应用	216

第 10 章 RTOS 简介

10.1	RTOS 基本知识	218
10.2	Rtx51 Tiny 入门	219
10.2.1	用 Rtx51 Tiny 实现 LED 闪烁	219
10.2.2	对 RTOS 工作过程的初步认识	222
10.3	Rtx51 Tiny 的工作过程及其函数	224
10.3.1	键控流水灯程序	224
10.3.2	Rtx51 Tiny 的工作过程	226
10.3.3	Rtx51 Tiny 的事件和信号	227
10.3.4	Rtx51 Tiny 的函数	228
10.4	Rtx51 Tiny 应用实例	232
10.4.1	百分秒表的实现	232
10.4.2	Rtx51 Tiny 中的信号	237
10.5	使用 Rtx51 Tiny 操作系统程序的调试	237

目 录

第 11 章 C51 库函数	
11.1	C51 库函数的测试方法 239
11.2	使用 math.h 头文件的函数 244
11.3	使用 ctype.h 头文件的函数 248
11.4	使用 stdlib.h 头文件的函数 251
11.5	使用 intrins.h 头文件的函数 256
11.6	使用 string.h 头文件的函数 259
11.7	使用 assert.h 头文件的函数 264
11.8	使用 setjmp.h 头文件的函数 265
11.9	使用 stddef.h 头文件的函数 267
11.10	使用 stdio.h 头文件的函数 269
附录 A C 语言的关键字	
6	A.1 C 语言中的关键字 275
	A.2 Keil C 新增的关键字 276
附录 B 进阶与提高	
	B.1 具有更多功能的实验电路板 277
	B.2 高性价比的两款开发工具 278
	B.2.1 基于 Keil Monitor - 51 Drivr 的仿真机 278
	B.2.2 下载型编程器 279
附录 C 配套光盘的说明	
	C.1 文件夹的说明 281
	C.2 安装及使用说明 281
	参考文献 285

第 1 章

单片机 C 语言概述

随着单片机开发技术的不断发展,目前已有越来越多的人从普遍使用汇编语言到逐渐使用高级语言进行开发,其中以 C 语言为主,而且市场上几种常见的单片机均有 C 语言开发环境。本书将以最为流行的 80C51 单片机为例来学习单片机的 C 语言编程技术。下面首先介绍 C 语言的基本知识。

1.1 C 语言简介

1.1.1 C 语言的产生与发展

C 语言由早期的编程语言 BCPL(Basic Combind Programming Language)发展演变而来。1970 年美国贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出 B 语言,并用 B 语言编写了 UNIX 操作系统。1972—1973 年,贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言。

随着微型计算机的日益普及,出现了许多 C 语言版本,但是没有统一的标准使这些 C 语言版本之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。

1.1.2 C 语言的特点

C 语言发展非常迅速,成为最受欢迎的语言之一,主要因为它具有强大的功能。现归纳起来 C 语言具有如下特点。

1. 与汇编语言相比

(1) 具有结构化控制语句

结构化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。

(2) 适用范围大和可移植性好

同其他高级语言一样,C 语言不依赖于特定的 CPU,其源程序具有很好的可移植性。只要某种 CPU 或 MCU 有相应的 C 编译器,就能使用 C 语言进行编程。目前,主流 CPU 和常见 MCU 都有 C 编译器。作为嵌入式系统的开发者,这一点显得很重要。目前可供选

1 单片机 C 语言概述

用的不同公司的 MCU 型号极多。这些 MCU 各有特点,开发者在做不同项目时往往需要选用不同品种的 MCU 以各尽其能。但要熟悉每一种 MCU 的汇编语言并能写出高质量的程序,并非易事。如果使用 C 语言编程,借助于 C 语言的可移植性,只要熟悉所用 MCU 的特性即可进行编程。这可节省大量的时间,从而将精力专注于你所要解决的问题上面。

作者曾将一个应用程序从 80C51 单片机上移植到 PIC 单片机上,原来的程序是在 89C52 中实现的,现需要移植到 PIC16F872 中。这是两种内核完全不同的单片机,如果使用汇编语言则必须完全重写程序;而当时对 PIC 单片机还是一无所知,想要在短时间内掌握 PIC 单片机的汇编语言并编写一个中等复杂的程序,显然并非易事。好在原来的程序也是用 C 语言编写的,因此可以方便地进行移植。找到 PIC 单片机的 C 编译程序后,一边熟悉 PIC 内部结构的同时,一边进行移植,在很短的时间内就将此程序完全移植成功。这其中的绝大部分时间是用来熟悉 PIC 的内部结构,理解其硬件用法与 80C51 之间的差异;而真正用于改写代码的时间非常少,需要修改的代码量也很少。

2. 与其他高级语言相比

(1) 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字和 9 种控制语句。其程序书写自由,主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。

(2) 运算符丰富

C 语言的运算符包含的范围很广泛,共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极其丰富,并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

C 语言的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能用来实现各种复杂的数据类型的运算。

(4) 程序设计自由度大

例如对数组下标越界不做检查,由程序编写者自己保证程序的正确;对变量的类型使用比较灵活,整型、字符型等各种变量可通用。

(5) 允许直接访问物理地址,可以直接对硬件进行操作

因此 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。

(6) 生成代码质量高

用 C 语言编写的程序,编译后一般只比有丰富经验的汇编编程人员所编写的汇编程序效率低 10%~20%。

3. 关于 C 语言的学习

很多人对 C 语言的学习有一种畏惧情绪,因为“传说中”C 语言很难学,这里有必要对此进行一些说明。

C 语言难学的说法来源于在 PC 机上编程的程序员。早期 PC 机上使用的编程语言主要有 BASIC 语言、PASCAL 语言、数据库(DBASE、FOXBASE 等)编程语言和 C 语言等。与这些语言相比,C 语言的确属于“难学”的语言,但必须对为何“难学”进行深入分析。由于

C 语言能够完成底层操作,所以往往被用来编写一些系统软件,而这必然涉及更深奥的计算机基础知识。比如,必须理解 ASCII 码的知识、数据在内存中如何存放的知识才能理解为何只要简单的一句“ $c=c+32;$ ”,即可完成大写字母到小写字母的转换,而这样的工作在 BASIC 中是用专门的函数来完成的,并不需要使用者了解这些知识。换言之,C 语言的难学来自于它所要完成的任务。对于单片机编程而言,这些知识本来就属于必须要掌握的,所以如果读者已有一定的汇编语言基础,那么这就不是什么难题。作为单片机开发者,必须要接触硬件,理解这些知识,所以这些并不能算在“C 难学”上面。

1.2 C 语言的入门知识

下面将通过一些实例介绍 C 语言编程的方法。这里采用 80C51 系列单片机的 C 编译器 Keil 软件作为开发环境。关于 Keil 软件的具体用法将在第 2 章作详细介绍。

图 1-1 所示电路图采用 89S52 单片机。这种单片机性属于 80C51 系列,其内部有 8 KB 的 FLASH ROM,可以反复擦写,并有 ISP 功能,支持在线下载,不需要反复插拔芯片,非常适于做实验。2.5 节将详细介绍使用这种单片机制作的硬件实验平台。图 1-1 所示 89S52 的 P1 引脚上接 8 个发光二极管,下面一些例子的任务是让接在 P1 引脚上的发光二极管按要求发光。

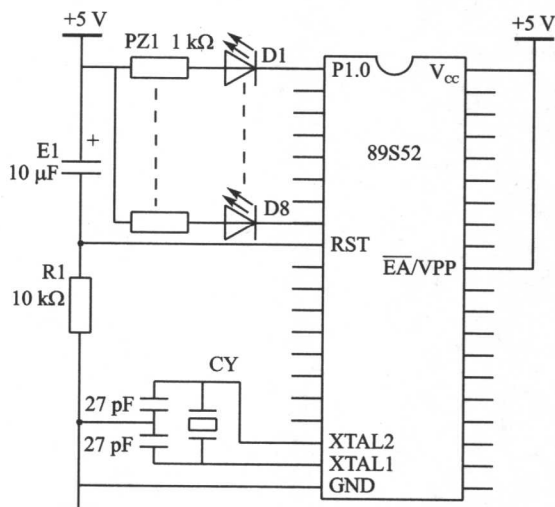


图 1-1 接有 LED 的单片机基本电路

1.2.1 简单的 C 程序介绍

例 1.1 让接在 P1.0 引脚上的 LED 发光。

```
#include "reg51.h"
sbit P1_0=P1^0;
void main()
{
    P1_0=0;
}
```

这个程序的作用是让接在 P1.0 引脚上的 LED 点亮。下面来分析一下这个 C 语言程

1 单片机 C 语言概述

序包含的如下信息。

1. “文件包含”处理

程序的第一行是一个“文件包含”处理。

所谓“文件包含”是指一个文件将另外一个文件的内容全部包含进来。虽然看起来这个程序只有 4 行,但 C 编译器在处理这段程序时却要处理几十或几百行。这段程序中包含 reg51.h 文件的目的是为了使用 P1 这个符号,即通知 C 编译器,程序中的 P1 是指 80C51 单片机的 P1 端口而不是其他。这是如何做到的呢?

打开 reg51.h 可以看到如下内容:

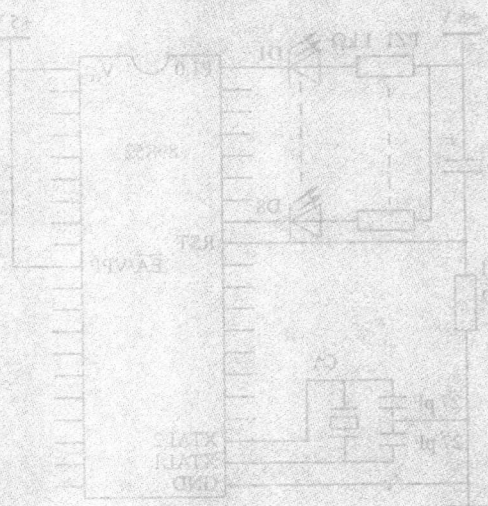
```

/* -----
reg51.h
Header file for generic 80C51 and 80C31 microcontroller.
Copyright (c)1988—2001 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
----- */

/* BYTE Register */
sfr P0      = 0x80;
sfr P1      = 0x90;
sfr P2      = 0xA0;
sfr P3      = 0xB0;
sfr PSW     = 0xD0;
sfr ACC     = 0xE0;
sfr B       = 0xF0;
sfr SP      = 0x81;
sfr DPL     = 0x82;
sfr DPH     = 0x83;
sfr PCON    = 0x87;
sfr TCON    = 0x88;
sfr TMOD    = 0x89;
sfr TL0     = 0x8A;
sfr TL1     = 0x8B;
sfr TH0     = 0x8C;
sfr TH1     = 0x8D;
sfr IE      = 0xA8;
sfr IP      = 0xB8;
sfr SCON    = 0x98;
sfr SBUF    = 0x99;

/* BIT Register */
/* PSW */
sbit CY     = 0xD7;
sbit AC     = 0xD6;
sbit F0     = 0xD5;
sbit RS1    = 0xD4;

```



```

sbit RSO = 0xD3;
sbit OV = 0xD2;
sbit P = 0xD0;

/* TCON */
sbit TF1 = 0x8F;
sbit TR1 = 0x8E;
sbit TF0 = 0x8D;
sbit TR0 = 0x8C;
sbit IE1 = 0x8B;
sbit IT1 = 0x8A;
sbit IE0 = 0x89;
sbit IT0 = 0x88;

/* IE */
sbit EA = 0xAF;
sbit ES = 0xAC;
sbit ET1 = 0xAB;
sbit EX1 = 0xAA;
sbit ET0 = 0xA9;
sbit EX0 = 0xA8;

/* IP */
sbit PS = 0xBC;
sbit PT1 = 0xBB;
sbit PX1 = 0xBA;
sbit PT0 = 0xB9;
sbit PX0 = 0xB8;

/* P3 */
sbit RD = 0xB7;
sbit WR = 0xB6;
sbit T1 = 0xB5;
sbit T0 = 0xB4;
sbit INT1 = 0xB3;
sbit INT0 = 0xB2;
sbit TXD = 0xB1;
sbit RXD = 0xB0;

/* SCON */
sbit SM0 = 0x9F;
sbit SM1 = 0x9E;
sbit SM2 = 0x9D;
sbit REN = 0x9C;
sbit TB8 = 0x9B;
sbit RB8 = 0x9A;
sbit TI = 0x99;
sbit RI = 0x98;
    
```


1 单片机 C 语言概述

熟悉 80C51 内部结构的读者不难看出,这里都是一些符号的定义,即规定符号名与地址的对应关系。注意其中这样的一行(上面的包含文件 reg51.h 中用黑体表示):

```
sfr P1 = 0x90;
```

即定义 P1 与地址 0x90 对应,P1 口的地址就是 0x90(0x90 是 C 语言中十六进制数的写法,相当于汇编语言中的 90H)。

从这里还可以看到一个频繁出现的词:sfr。

sfr 不是标准 C 语言的关键字,而是 Keil C 编译器为了能够直接访问 80C51 中的 SFR 所提供的的一个新的关键词,其用法是:

```
sfrt 变量名 = 地址值;
```

例如:

```
sfr P1 = 0x90;
```

该语句定义了 P1 这个名称与 90H 这个地址的对应关系。通过这种方法,可以自行定义新的特殊功能寄存器(SFR)。随着技术的不断发展,新的 80C51 系列单片机层出不穷,这些新的 80C51 单片机通常会增加一些 SFR 以增强功能。通常每一种新单片机出来之后,Keil 软件都会升级以支持这种单片机。但即便不能及时升级自己的 Keil 软件也没有关系,可以通过 sfr 关键字自行定义 SFR 符号与其地址的对应关系,以便能通过符号来使用这些新的 SFR。

例如,89S 系列单片机中增加了看门狗定时器,其数据手册上的名称为 WCON,地址为 96H;因此,如果手边的 Keil 软件中找不到现成的头文件,那么可以在 reg51.h 中或者在该程序开头增加下面一行:

```
sfr WCON = 0x96;
```

这样即可在程序中使用 WCON 这个符号了。

2. 符号 P1_0 用来表示 P1.0 引脚

在 C 语言中,如果直接写 P1.0,C 编译器不能识别,而且 P1.0 也不是一个合法的 C 语言标识符,所以得给它另起一个名字。这里起的名为 P1_0,但 P1_0 是否就是 P1.0 呢?即使你这么认为,但 C 编译器并不这么认为,所以必须给它们之间建立联系。这里使用了 Keil C 增加的关键字 sbit 来定义。sbit 的用法有 3 种:

- ① sbit 位变量名 = 地址值;
- ② sbit 位变量名 = SFR 名称·变量位地址值;
- ③ sbit 位变量名 = SFR 地址值·变量位地址值。

例如,定义 PSW 中的 OV 可以用以下 3 种方法:

- ① sbit OV = 0xd2 0xd2 是 OV 的位地址值;
- ② sbit OV = PSW^2 PSW 必须先用 sfr 定义好;
- ③ sbit OV = 0xD0^2 0xD0 就是 PSW 的地址值。

因此这里用“sbit P1_0 = P1^0;”,即用符号 P1_0 来表示 P1.0 引脚。如果愿意,也可用 P10 之类的名字,只要下面程序中也随之更改。

Keil 软件在 AT89X52.H 中已定义了各引脚的变量,所以如果包含了该文件,就不需要自行定义。该文件在 keil\c51\inc\atmel 文件夹中。