

职业教育课程改革规划新教材

单片机应用与调试 项目教程 (C语言版)

Dan Pian Ji Ying Yong Yu Tiao Shi
Xiang Mu Jiao Cheng (C Yuyan Ban)

周正鼎 主编



机械工业出版社
CHINA MACHINE PRESS

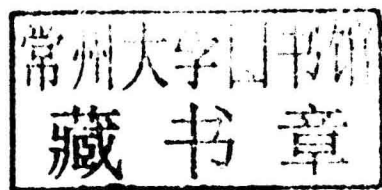


电子教案

职业教育课程改革规划新教材

单片机应用与调试项目教程 (C语言版)

主 编 周正鼎
副主编 龙善寰 宋天齐
参 编 周志文 黄丽君 陈 俭 江幸福



机械工业出版社

本书是在总结编者多年课程改革、技能大赛辅导以及工程实践经验的基础上编写而成的。以 51 系列单片机为载体,运用项目式的编写模式介绍单片机的应用与调试技术。为更贴近工程实际,本书将 C 语言编程引入进来。本书主要介绍以下内容:单片机基础知识、LED 及键盘、电机控制、A/D 转换及 D/A 转换、显示系统、单片机通信以及单片机控制系统安装与调试项目精选。每一章下均由若干个项目组成,最终在第 7 章用综合项目的形式将单片机应用技术提升到一定高度。

本书内容翔实、实例丰富,有较强的实际应用指导价值,可作为电子信息类专业单片机课程教学用书,也可作为单片机调试比赛的指导用书。

本书配有免费电子教案及源程序代码,需要者可登录 www.cmpedu.com 进行注册及下载。

图书在版编目 (CIP) 数据

单片机应用与调试项目教程: C 语言版/周正鼎主编. —北京:机械工业出版社, 2011. 8

职业教育课程改革规划新教材

ISBN 978-7-111-34632-6

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

中国版本图书馆 CIP 数据核字 (2011) 第 143654 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:高倩 责任编辑:高倩 谷雨春 版式设计:霍永明

责任校对:刘怡丹 封面设计:鞠杨 责任印制:杨曦

北京中兴印刷有限公司印刷

2011 年 9 月第 1 版第 1 次印刷

184mm×260mm·12.75 印张·312 千字

0 001—2 500 册

标准书号: ISBN 978-7-111-34632-6

定价: 25.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务 网络服务

社服务中心: (010)88361066

门户网: <http://www.cmpbook.com>

销售一部: (010)68326294

教材网: <http://www.cmpedu.com>

销售二部: (010)88379649

读者购书热线: (010)88379203

封面无防伪标均为盗版

前 言

本书是在总结编者多年课程改革、技能大赛辅导以及工程实践经验的基础上编写而成的。在课程内容的安排上,打破学科体系的束缚,抛弃了以往教材单纯理论讲述或从理论到实践的排列方式,将单片机应用知识及使用技巧渗透到解决任务(问题)的过程中。以学生的实际认知规律为依据,以理论实践一体化的教学思想为宗旨,本着“做学教一体化”的理念,将工作任务与学习任务相结合,工作过程与学习过程相结合,工作内容与学习内容相结合。

为帮助学生理清学习思路,本书的主体部分还是以单片机的功能为主线,划分为6章,即单片机基础知识、LED及键盘、电机控制、A/D转换及D/A转换、显示系统、单片机通信等功能模块。在每个模块中按照任务引领的思路,设计了若干个单体项目,对某一部分单片机的应用知识、相关电路原理以及编程技巧等进行专项训练。项目设计的难度由浅入深、逐层递进,在对基本技能进行反复强化训练的同时,每个项目又有新的内容加入,使得学生逐步树立信心、增强兴趣。在本书的最后增加了一个综合项目模块,在不增加新知识的情况下,集中力量训练学生综合运用单片机知识的能力。在此环节,模拟技能大赛的出题方式,以任务单的形式命题,即仅交代出工作任务名称、工作任务的背景、需达到的技术要求、所能提供的设备、材料等条件,因此除了单片机知识与技能运用本身的综合训练,还训练了学生“明确任务、收集信息、计划和决策、实施与检查、评价与总结”等职业能力。

本书在内容设计方面提供了每个电路的设计原理图,同时以国内目前使用较多的技能大赛单片机项目指定使用设备YL-236为硬件平台进行接口电路介绍。在软件设计环节,着重流程图的介绍,帮助学生理解源程序的同时,强调了设计思路在编写程序中的重要地位。本书所有程序均采用C语言编写,这样更贴近目前的工程实际应用。学生在学习单片机应用的同时还掌握了一门高级计算机语言。

本书编写人员及分工如下:周正鼎编写第1、2、7章,龙善襄编写第3章,周志文编写第4章,黄丽君编写第5章,陈俭、江幸福共同编写第6章。全书由周正鼎提供策划思路并统稿。本书在编写过程中还吸收了多位技能大赛参赛选手的意见和体会,在此一并表示感谢。

本书可作为职业院校电类专业“单片机应用技术”课程教材,也可作为技能大赛单片机项目的基础辅导书或作为单片机爱好者的学习参考书。

为方便教学及自学,本书所涉及的所有源程序代码将上传到机械工业出版社教材服务网(www.cmpedu.com),有需要的读者可以下载使用。

由于水平所限,错误和不妥之处在所难免,恳请大家批评指正。

编 者

2011年5月

目 录

前言

第1章 单片机基础知识 1

- 1.1 单片机概述 1
- 1.2 数制的转换 3
- 1.3 符号数的表示方法 6
- 1.4 编码 6
- 1.5 C语言与 C51 7

第2章 LED及键盘 9

- 项目一 点亮一个LED灯 9
- 项目二 点亮一个闪烁的LED灯 14
- 项目三 点亮流水灯 23
- 项目四 用数码管实现静态显示 30
- 项目五 用数码管实现动态显示 38
- 项目六 用独立按键控制 51
- 项目七 用4×4矩阵键盘控制 57
- 实战训练 数字时钟 64

第3章 电机控制 73

- 项目一 制作24V和220V交、直流减速电动机测速控制器 73
- 项目二 制作步进电机的开环控制器 83

第4章 A/D转换及D/A转换 90

- 项目一 将电压值进行A/D转换并显示 90
- 项目二 将数字进行D/A转换并以电压

形式输出 98

项目三 制作模拟测温传感器 103

项目四 制作数字测温传感器 109

实战训练 制作温控装置 120

第5章 显示系统 126

项目一 用LED点阵显示汉字 126

项目二 在液晶显示器上显示英文及汉字 133

实战训练 用字符液晶显示温度 149

第6章 单片机通信 154

项目 单片机与PC进行串口通信 154

第7章 单片机控制系统安装与调试

项目精选 164

单片机安装与调试项目一 164

单片机安装与调试项目二 170

单片机安装与调试项目三 174

单片机安装与调试项目四 179

附录 184

附录A C51关键字简述 184

附录B 图解Keil软件操作步骤 185

附录C ASCII码表 196

参考文献 198

第 1 章 单片机基础知识

1.1 单片机概述

【什么是单片机】

单片机也被称为微控制器 (Microcontroller)，最早被用在工业控制等领域。单片机由芯片内仅有 CPU 的专用处理器发展而来。最早的设计理念是通过将大量外围设备和 CPU 集成在一个芯片中，使计算机系统更小，更容易集成复杂而对体积要求严格的控制设备。图 1-1 所示为 AT89S51 单片机。

和计算机相比，单片机只缺少了 I/O 设备。它的体积小、质量轻、价格便宜，为学习、应用和开发提供了便利条件。单片机内部也有和计算机功能类似的模块，如 CPU、内存、并行总线等。现在通用的全自动滚筒洗衣机、排烟罩、VCD 等家用电器里面都可以看到它的身影！

单片机是一种在线式实时控制计算机。在线式就是现场控制，需要有较强的抗干扰能力和较低的成本，这是与离线式计算机（比如家用 PC）的主要区别。单片机是靠程序控制的，通过不同的程序实现不同的控制功能，尤其是器件需要费很大力气才能做到的。比如要完成一个不是很复杂的功能，如果用美国 20 世纪 50 年代开发的 74 系列，或者 60 年代的 CD4000 系列这些纯硬件来做，电路一定是一大块 PCB！如果用美国 70 年代成功投放市场的系列单片机，结果就会有天壤之别！因为单片机通过编写的程序就可以实现高智能、高效率，以及高可靠性的复杂控制。

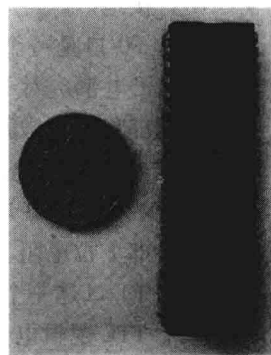


图 1-1 AT89S51 单片机

【单片机的应用领域】

目前，单片机的应用已经渗透到我们生活的各个领域，几乎很难找到哪个领域没有单片机的踪迹，导弹的导航装置，飞机上各种仪表的控制，计算机的网络通信与数据传输，工业自动化过程的实时控制和数据处理，广泛使用的各种智能 IC 卡，轿车的安全保障系统，录像机、摄像机、全自动洗衣机，以及程控玩具、电子宠物等，这些都离不开单片机。再比如，自动控制领域的机器人、智能仪表、医疗器械等。因此，单片机的学习、开发与应用将造就一批计算机应用与智能化控制的科学家和工程师。

单片机广泛应用于仪器仪表、家用电器、医用设备、航空航天、专用设备的智能化管理及过程控制等领域。

【AT89S52 单片机简介】

AT89S52 单片机是一种低功耗、高性能 CMOS 8 位微控制器，具有 8KB 在系统可编程 Flash 存储器。使用 ATMEL 公司高密度非易失性存储器技术制造，与工业 80C51 产品指令和

引脚完全兼容。片内配置 Flash 存储器, 使得 AT89S52 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。

AT89S52 具有以下标准功能: 8KB 容量 Flash 存储器, 256B RAM, 32 位 I/O 端口线, “看门狗”定时器, 2 个数据指针, 3 个 16 位定时器/计数器, 1 个 6 向量 2 级中断结构, 全双工串行口, 片内晶振及时钟电路。另外, AT89S52 可降至 0Hz 静态逻辑操作, 支持两种软件, 可选择节电模式。空闲模式下, CPU 停止工作, 允许 RAM、定时器/计数器、串口、中断继续工作。掉电保护方式下, RAM 内容被保存, 振荡器被冻结, 单片机停止一切工作, 直到下一个中断或硬件复位为止。

引脚功能介绍:

- V_{CC} : 电源。
- GND: 接地。
- P0 口: P0 口是一个 8 位漏极开路双向 I/O 端口。作为输入口, 每位能够驱动 8 个 TTL 逻辑电平。对 P0 口进行写“1”操作时, 引脚用做高阻抗输入。

当访问外部程序和数据存储器时, P0 口也被作为低 8 位的地址/数据复用。在这种模式下, P0 口具有内部上拉电阻。

在 Flash 编程时, P0 口也用来接收指令字节; 在程序校验时, 输出指令字节; 在程序校验时, 需要外部上拉电阻。

- P1 口: P1 口是一个具有内部上拉电阻的 8 位双向 I/O 端口, P1 口输出缓冲器能够驱动 4 个 TTL 逻辑电平。对 P1 口进行写“1”操作时, 内部上拉电阻把端口拉高, 此时作为输入口使用。这种接口输出没有高阻态, 输入也不能锁存, 故不是真正双向 I/O 端口, 所以称之为“准双向”因为前面提到过, 要作为出入, 必须先向该口写“1”, 然后单片机内部才能正确地读出外部信号。
- P2 口: 准双向 8 位 I/O 端口, 每个端口可以独立控制内带上拉电阻, 与 P1 口类似。
- P3 口: 准双向 8 位 I/O 端口, 每个端口可以独立控制内带上拉电阻, 第一功能就是当做普通 I/O 端口, 与 P1 口功能类似, 作为第二功能使用时, P3 口的每一个引脚都可以独立定义第一功能或者第二功能。P3 口引脚第二功能, 见表 1-1。

表 1-1 P3 口引脚第二功能

标 号	引 脚	第 二 功 能	说 明
P3.0	10	RXD	串行输入口
P3.1	11	TXD	串行输出口
P3.2	12	INT0	外部中断 0
P3.3	13	INT1	外部中断 1
P3.4	14	T0	定时器/计数器 0 外部输入端
P3.5	15	T1	定时器/计数器 1 外部输入端
P3.6	16	WR	外部数据存储器写脉冲
P3.7	17	RD	外部数据存储器读脉冲

- RST: 复位输入。晶振工作时, RST 持续两个机器周期, 高电平将使单片机复位。“看门狗”计时完成后, RST 引脚输出 96 个晶振周期的高电平。特殊寄存器 AUXR (地址 8EH) 上的 DISRTO 位可以使此功能无效。DISRTO 默认状态下, 复位高电平有效。
- ALE/PROG: 地址锁存控制信号 (ALE) 是访问外部程序存储器时, 锁存低 8 位地址

的输出脉冲。在 Flash 编程时，此引脚（PROG）也用做编程输入脉冲。

在一般情况下，ALE 以晶振六分之一的固定频率输出脉冲，可用来作为外部定时器或时钟使用。然而，特别强调，在每次访问外部数据存储器时，ALE 脉冲将会跳过。

如果需要，通过将地址为 8EH 的 SFR 的第 0 位置“1”，ALE 操作将无效。这一位置“1”，ALE 仅在执行 MOVX 或 MOVC 指令时有效。否则，ALE 将被微弱拉高。这个 ALE 使智能标志位（diz 为 8EH 的 SFR 的第 0 位）的设置在微控制器处于外部执行模式下无效。

- PSEN：外部程序存储器选通信号，当 AT89S52 从外部程序存储器执行外部代码时，PSEN 在每个机器周期被激活两次，而在访问外部数据存储器时，PSEN 将不被激活。
- EA/VPP：访问外部程序存储器控制信号。为使能从 0000H 到 0FFFFH 的外部程序存储器读取指令 EA 必须接 GND。如果要执行内部程序指令，EA 应该接 V_{CC} 。在 Flash 编程期间 EA 也接 12V 电压。
- XTAL1：振荡器反相放大器和内部时钟发生电路的输入端。
- XTAL2：振荡器反相放大器的输出端。

1.2 数制的转换

【计算机中的数制】

计算机最早是作为一种计算工具出现的，所以单片机最基本的功能是对数据进行加工和处理。数据在机器中是以器件的物理状态的来表示的。一个具有两种不同的稳定状态且能相互转换的器件就可以用来表示 1 位（bit）二进制数。二进制数有运算简单、便于物理实现、节省设备等优点，所以目前的计算机都采用二进制数表示。

由于二进制数书写太长，且不便阅读和记忆。目前大部分微型计算机是 8 位、16 位或 32 位，都是 4 的整数倍，而 4 位二进制数就是 1 位十六进制数，所以微型计算机广泛采用的都是十六进制数来缩写二进制数。十六进制数就是用 0~9、a~f 共 16 个数码表示十进制数 0~15，1 个 8 位二进制数可以用 2 位十六进制数表示，这样书写方便，且便于阅读和记忆，然而人们最熟悉的是十进制数。为此要掌握各个进制之间的转换。

为了区别上面三种数制，可以在数的右下角注明数制，或者在数的后面加字母进行区别，如 B 表示二进制数，D 或者不带字母表示十进制数，H 表示十六进制数。

【十进制数与二进制数之间的相互转换】

1. 十进制数转换为二进制数

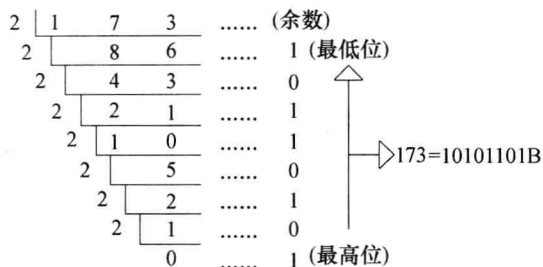
方法：“除 2 取余”法，除 2 取余，商 0 为止，余数倒排。

例 1-1 将十进制数 173 转换为二进制数，得：173D = 10101101B。

解：将 173 除 2 取余，商 0 为止，余数倒排。

2. 二进制数转换为十进制数

方法：“按权展开求和”。



例 1-2 将 10101101B 转换为十进制数。

$$\begin{aligned}
 \text{解: } 10101101\text{B} &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= 128 + 0 + 32 + 0 + 8 + 4 + 0 + 1 \\
 &= 173
 \end{aligned}$$

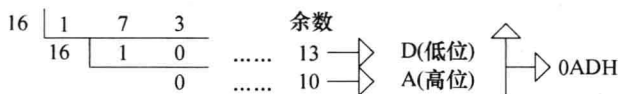
【十进制数与十六进制数之间的相互转换】

1. 十进制数转换为十六进制数

方法：“除 16 取余，商 0 为止，余数倒排”。

例 1-3 将十进制数 173 转换为十六进制数，得：173D = 0ADH。

解：将 173 除 16 取余，商 0 为止，余数倒排。



2. 十六进制数转换为十进制数

方法：“按权展开求和”。

例 1-4 将十六进制数 0ADH 转换为十进制数，得：0ADH = 173。

解：用“按权展开求和”法，得：

$$0\text{ADH} = \text{A} \times 16^1 + \text{D} \times 16^0 = 10 \times 16 + 13 \times 1 = 160 + 13 = 173$$

对于以下十六进制数 0FH、0FFH、0FFFH、1FFFH 等，用如下方法可快速求出对应的二进制数和十进制数：

1) 把十六进制数写成对应的二进制数；

2) 根据二进制数中“1”的个数 x ，用公式 $2^x - 1$ 可迅速求出对应的十进制数，见表 1-2。

表 1-2 十六进制-二进制-十进制数转换

十六进制数	对应的二进制数	对应的十进制数
0FH	1111B(4 个 1)	$2^4 - 1 = 16 - 1 = 15$
0FFH	1111 1111B(8 个 1)	$2^8 - 1 = 256 - 1 = 255$
0FFFH(89S51 单片机的末地址)	1111 1111 1111B(12 个 1)	$2^{12} - 1 = 4096 - 1 = 4095$
1FFFH(89S52 单片机的末地址)	0001 1111 1111 1111(13 个 1)	$2^{13} - 1 = 8192 - 1 = 8191$

【二进制数与十六进制数之间的相互转换】

1. 二进制数转换为十六进制数

方法：“4 位一组，求 8421 码之和”。

例 1-5 将二进制数 1010 1101B 转换为十六进制数，得：10101101B=0ADH。

解：用“4 位一组，求 8421 码之和”法，将 1010 1101B 4 位一组，分为左右两组：

8	4	2	1	码	8	4	2	1	码
1	0	1	0		1	1	0	1	

对左组，按位求 8421 码之和，得： $1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 10 = 0AH$

对右组，按位求 8421 码之和，得： $1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13 = 0DH$

所以 1010 1101B=0ADH。

2. 十六进制数转换为二进制数

方法：“按位拼 8421”法。

例 1-6 将十六进制数 0ADH 转换为二进制数，得：0ADH=1010 1101B。

解：十六进制数 0AH，用“按位拼 8421”法，得 4 位二进制数，即 0AH=1010B。

十六进制数 0DH，用“按位拼 8421”法，得 4 位二进制数，即 0DH=1101B。

所以 0ADH=10101101B

【十进制小数转化为二进制小数】

将十进制小数乘以 2，取其小数点左边的数，就可以完成小数的转换。

例 1-7 将十进制小数 0.706D 转换为二进制数，要求其误差不大于 2^{-6} 。

解：

第一步： $0.706 \times 2 = 1.412 \dots \dots \dots .1$

第二步： $0.412 \times 2 = 0.824 \dots \dots \dots .0$

第三步： $0.824 \times 2 = 1.648 \dots \dots \dots .1$

第四步： $0.648 \times 2 = 1.296 \dots \dots \dots .1$

第五步： $0.296 \times 2 = 0.592 \dots \dots \dots .1$

第六步： $0.592 \times 2 = 1.184 \dots \dots \dots .1$

所以 $0.706 = 0.101111B$ ，其误差 $< 2^{-6}$ 。要注意的是，十进制小数转换成二进制小数，最后排列的顺序是顺着排列的。

除此之外还可以使用计算机自带的计算器进行转换，如图 1-2 所示。



图 1-2 用科学计算机进行数制转换

1.3 符号数的表示方法

【机器数与真值】

机器数就是数值在计算机中的表示形式,即数值在计算机中的二进制表示形式。真值则是它在现实中的实际数值。因为计算机只能直接识别和处理用 0、1 两种状态的二进制形式的数据,所以在计算机中无法按人们的日常书写习惯用正、负符号加绝对值来表示数值,而与数字一样采用二进制代码 0 和 1 来表示正、负号。这样在计算机中表示带符号的数值数据时,符号和数均采用了 0、1 进行了代码化。这种采用二进制表示形式,连同正、负符号一起代码化的数据,称为机器数或者机器码。与机器数对应用正、负符号加绝对值来表示的实际数值称为真值。

根据约定机器数是否存在符号位,机器数可以分为无符号数和带符号数。无符号数是指计算机字长的所有二进制位均表示数值。带符号数是指机器数分为符号位和数值两部分,且均采用二进制表示。一般约定最高位表示符号。

机器内部设备一次能表示的二进制位数称为机器的字长,一台机器的字长是固定的。8 位二进制数称为 1 字节 (Byte),现在机器字长一般都是字节的整数倍,如字长 8 位、16 位、32 位、64 位等。

在计算机中为了便于数值的运算和处理,对机器数定义了不同的表示方法,其中包括了数的原码、补码、反码和移码表示。下面分别对它们进行介绍。

【原码】

数值用其绝对值表示,正数的符号位用 0 表示,负数的符号位用 1 表示,这样表示的数称为原码。例如:

$$X1 = 105 = +1101001B \quad [X1]_{\text{原}} = 01101001B$$

$$X2 = -105 = -1101001B \quad [X2]_{\text{原}} = 11101001B$$

最高位是符号位,后面 7 位是数制,用原码表示的时候, +105 和 -105 数值部分相同,符号为相反。

为了把减法运算转换成加法运算,简化计算机结构,引进了反码和补码。

【反码】

正数的反码和原码相同,负数的反码是其绝对值按位取反得到的。例如:

$$X1 = 105 = +1101001B \quad [X1]_{\text{原}} = 01101001B$$

$$X2 = -105 = -1101001B \quad [X2]_{\text{原}} = 10010110B$$

【补码】

正数的补码与原码相同,负数的补码是在反码的基础上加 1 得到的。例如:

$$X1 = 105 = +1101001B \quad [X1]_{\text{原}} = 01101001B$$

$$X2 = -105 = -1101001B \quad [X2]_{\text{原}} = 10010111B$$

1.4 编码

下面介绍两种单片机中常用的编码。

【BCD 码】

BCD 码就是用 4 位二进制数来表示 1 位十进制数中的 0~9 这 10 个数码, 表 1-3 就是常用的十进制数转 BCD 码。

表 1-3 十进制数转 BCD 码

十 进 制 数	BCD 码	十 进 制 数	BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

【ASCII 码】

在计算机中, 所有的数据在存储和运算时都要使用二进制数表示 (因为计算机用高电平和低电平分别表示 1 和 0)。例如, 像 a、b、c、d 这样的 52 个字母 (包括大写), 以及 0、1 等数字还有一些常用的符号 (如 *、#、@ 等) 在计算机中存储时也要使用二进制数来表示。

1967 年美国国家标准学会 (American National Standard Institute, ANSI) 制定了, 后被国际标准化组织 ISO 定为国际标准, 适用于所有拉丁文字母。ASCII 码的字符定义见附录 C。

注意, 在标准 ASCII 码中, 其最高位 (b7) 用做奇偶校验位。所谓奇偶校验, 是指在代码传送过程中用来检验是否出现错误的一种方法, 一般分奇校验和偶校验两种。奇校验规定: 正确的代码 1 字节中 1 的个数必须是奇数, 若非奇数, 则在最高位 b7 添 1; 偶校验规定: 正确的代码 1 字节中 1 的个数必须是偶数, 若非偶数, 则在最高位 b7 添 1。

1.5 C 语言与 C51

【C 语言简介】

C 语言发展非常迅速, 而且成为最受欢迎的语言之一, 主要因为它具有强大的功能。许多著名的系统软件都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序, 就更能显示 C 语言的优势了, 像 PC-DOS、WORDSTAR 等就是用 C 语言编写的。C 语言有以下特点。

1. 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字, 9 种控制语句, 程序书写形式自由, 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。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

C 语言的运算符包含的范围很广泛, 共有 34 种运算符。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, 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结构的运算。C 语言引入了指针概念, 使程序效率更

高。另外, 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. 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、维护以及调试。C 语言是以函数形式提供给用户的, 这些函数可方便地调用, 并具有多种循环、条件语句控制程序流向, 从而使程序完全结构化。

5. C 语法限制不太严格, 程序设计自由度大

虽然 C 语言也是强类型语言, 但它的语法比较灵活, 允许程序编写者有较大的自由度。

6. C 语言允许直接访问物理地址, 可以直接对硬件进行操作

7. 生成目标代码质量高, 程序执行效率高

一般只比汇编程序生成的目标代码效率低 10% ~ 20%。

8. C 语言适用范围大, 可移植性好

C 语言有一个突出的优点就是适合于多种操作系统, 如 DOS、UNIX, 也适用于多种机型。C 语言具有强大的绘图能力, 可移植性好, 并具备很强的数据处理能力, 因此适于编写系统软件, 三维、二维图形和动画, 它也是数值计算的高级语言。

但是 C 语言也有缺点:

1. C 语言的缺点主要表现在数据的封装性上, 这一点使得 C 语言在数据的安全性上有很大缺陷, 这也是 C 和 C++ 的主要区别。

2. C 语言的语法限制不太严格, 对变量的类型约束不严格, 影响程序的安全性, 对数组下标越界不做检查等。从应用的角度, C 语言比其他高级语言较难掌握。

【Keil C51】

Keil C51 是美国 Keil Software 公司出品的 51 系列单片机兼容 C 语言软件开发系统, 与汇编语言相比, C 语言在功能、结构性、可读性、可维护性上有明显的优势, 因而易学易用。用过汇编语言后再使用 C 语言来开发, 体会更加深刻。Keil C51 软件提供丰富的库函数和功能强大的集成开发调试工具, 全 Windows 界面。另外重要的一点, 只要看编译后生成的汇编代码, 就能体会到 Keil C51 生成的目标代码效率非常之高, 多数语句生成的汇编代码很紧凑, 容易理解。在开发大型软件时更能体现高级语言的优势。

第2章 LED及键盘

项目一 点亮一个LED灯

【项目任务】

点亮显示模块上的一个LED灯。

【任务分析】

控制一个LED灯，可以用51单片机I/O端口的任一引脚实现。例如，用P1.0口控制，则当P1.0口输出“0”时，灯亮；输出“1”时，灯灭。

【硬件选择】

电源模块，主机模块，显示模块。

【接线表及原理图】

1. 接线表

实验接线表见表2-1。

表2-1 项目一模块电路接线表

P1.0	LED0
电源5V	主机5V,显示模块5V
电源GND	主机GND,显示模块GND

2. 原理图

实验原理图如图2-1所示。

【知识链接一】

LED是发光二极管（Light Emitting Diode）的简称，是一种固态的半导体器件，它可以直接把电转化为光。LED的心脏是一个半导体的晶片，晶片的一端附在一个支架上，一端是负极，另一端连接电源的正极，使整个晶片被环氧树脂封装起来，如图2-2所示。

LED具有单向导电性，一般通过5~10mA的电流即可发光，电流越大，亮度越强，但是若电流太大，则会烧毁LED。因此，我们通常会给LED串联一个电阻，以控制通过LED的电流大小，让LED在正常工作范围内工作。LED的电气符号如图2-3所示。

实验板上常用的两种LED实物，如图2-4和图2-5所示。

【软件设计】

1. 编程思路

点亮一个LED灯，也就是控制一个I/O端口输出低电平。这样，与之相连接的LED就会导通、发光。我们可以先用程序定义一个I/O端口，比如定义P1.0口，如下操作：

```
sbit P1_0 = P1^0;
```

然后给P1.0口送低电平“0”，就可以控制P1.0口输出低电平。我们就会看到连接在这个引脚上的LED灯被点亮了。编程流程如图2-6所示。

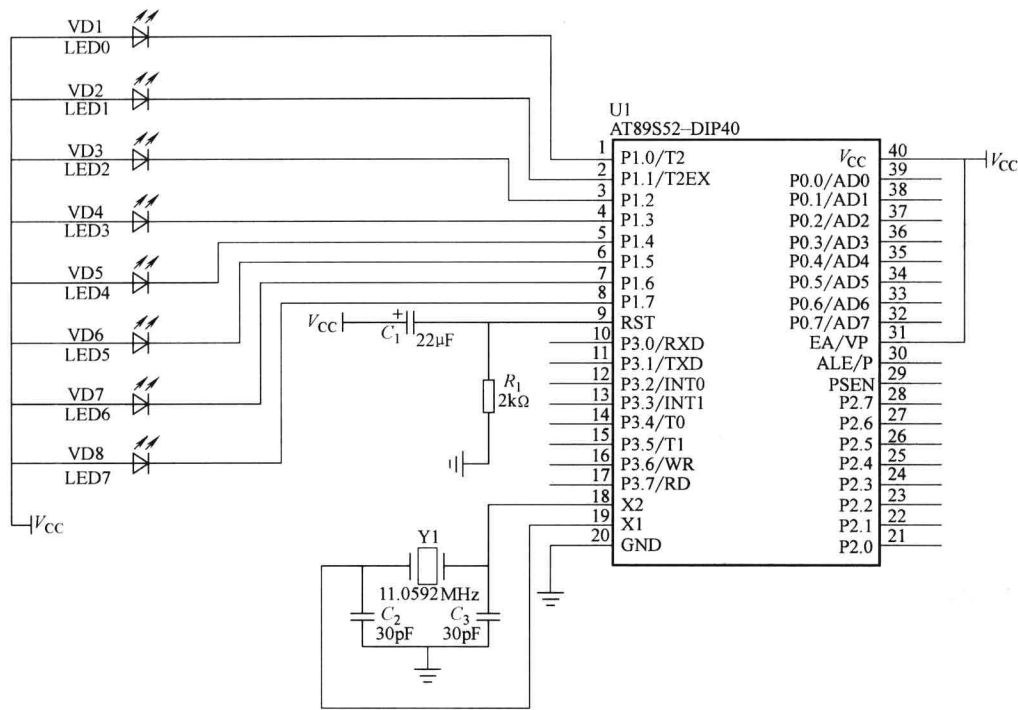


图 2-1 项目一原理图

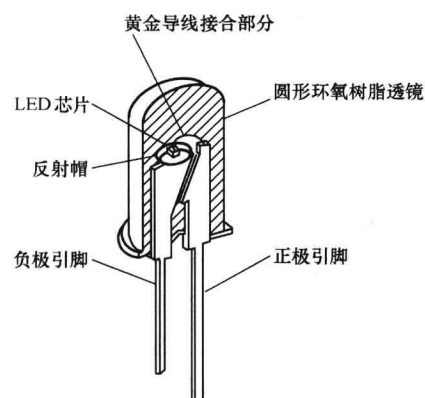


图 2-2 LED 的结构

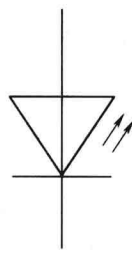


图 2-3 LED 的电气符号

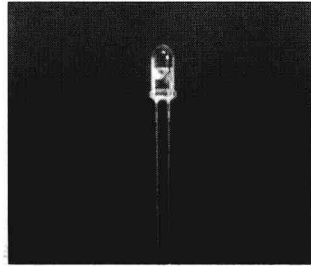


图 2-4 直插式 LED

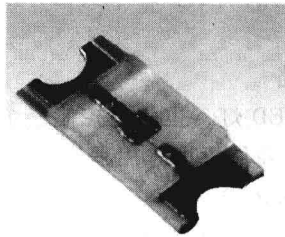


图 2-5 贴片式 LED

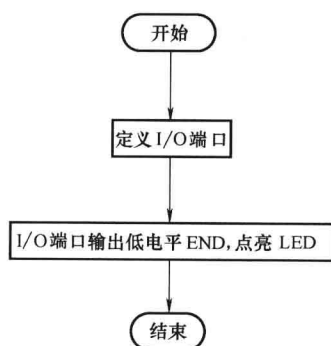


图 2-6 程序流程

2. 参考程序

```

#include <reg52. h>           //包含头文件 <reg52. h>
sbit P1_0 = P1^0;           //将 P1 口的最低位 P1^0 定义为符号 P1_0
void main()                  //主函数声明, C 语言中规定一个完整的程序中只能有一个主函数
{
    P1_0 = 0;                //C 语言规定, 一条命令后要加分号
}
  
```

【知识链接二】

1. 文件包含处理

在程序中引用头文件, 意思是将这个头文件中已经定义好的内容引入所编写的代码中。将头文件加入所编写的代码中有两种方法: `#include <reg52. h>` 和 `#include "reg52. h"`。通常习惯使用 `#include <reg52. h>` 这种方法来包含头文件。

C51 语言中常用的头文件有以下几种, `reg51. h`、`reg52. h`、`math. h`、`ctype. h`、`stdio. h`、`absacc. h` 和 `intrins. h`。可以发现, 头文件都有一个共同的特征: 扩展名都为 “.h”。其实我们常用的头文件并不多, 较常用的有 `reg51. h`、`reg52. h`、`absacc. h` 等。

`reg51. h` 和 `reg52. h` 里面的大部分内容都是一样的, 只不过因为 C52 单片机比 C51 单片机多了一个定时器, 所以在 `reg52. h` 这个头文件里就多了关于定时器 2 的定义。所以大部分情况下, 使用时没有什么区别。

`reg52. h` 文件夹内容如下 (已经删减, 关于完整内容, 读者可以自行打开文件查看):

```

/*-----
REG52. H
Header file for generic 80C52 and 80C32 microcontroller.
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
----- */
#define __REG52_H__
  
```

```

#define __REG52_H__
/*  BYTE Registers  */
sfr P0      = 0x80;
sfr P1      = 0x90;
sfr P2      = 0xA0;
sfr P3      = 0xB0;
sfr PSW     = 0xD0;
sfr PCON    = 0x87;
sfr TCON    = 0x88;
sfr TMOD    = 0x89;
sfr TL0     = 0x8A;
sfr TL1     = 0x8B;
sfr TH0     = 0x8C;
sfr TH1     = 0x8D;
sfr SCON    = 0x98;
sfr SBUF    = 0x99;
/*  8052 Extensions  */
sfr T2CON   = 0xC8;
sfr RCAP2L  = 0xCA;
sfr RCAP2H  = 0xCB;
/*  T2CON  */
sbit C_T2   = T2CON^1;
sbit CP_RL2 = T2CON^0;
#endif

```

2. C51 数据类型扩充定义

- sfr——特殊功能寄存器的数据声明，声明一个 8 位的寄存器。
- sfr16——16 位特殊功能寄存器的数据声明。
- sbit——特殊功能位声明，也就是声明某个特殊寄存器的中的某一位。
- sit——位变量声明，当定义一个位变量时可以使用此符号。

例如，定义特殊功能寄存器 TMOD 如下：

```
sfr TMOD = 0x89;
```

TMOD 是单片机的定时器/计数器工作方式寄存器，这个寄存器在单片机的内存地址是 0x89，这样使用 sfr 声明后，就会将 0x89 这个地址定义为 TMOD 这个符号，以后我们要对 0x89 这个内存地址进行操作时，就可以使用 TMOD 这个符号了。也就是说，TMOD 就是对 0x89 这个内存地址起的一个名字，当然也可以取其他名字。

```
sfr16 T2 = 0xcc;
```

T2 也是我们给 0xcc 这个地址起的一个名字，0xcc 是单片机定时器/计数器 2 所在内存的地址，它是一个 16 位的特殊功能寄存器。所以要用 sfr16 来定义。