

单片机原理与应用



类目：计算机类
书名：单片机原理与应用
主编：陈俊霖 唐运恩 高勇
出版社：中国建设科技出版社
开本：大 16 开
书号：978-7-5160-4390-5
使用层次：通用
出版时间：2025 年 8 月
定价：49.80 元
印刷方式：双色
是否有资源：有

项目统筹:李亚博
责任编辑:汪永涛
封面设计:旗语书装

计算机类创新融合精品规划教材
“互联网+”教育改革新理念教材



计算机类创新融合精品规划教材
“互联网+”教育改革新理念教材

单片机原理与应用

单片机原理与应用

主 编 © 陈俊霖 唐运恩 高 勇
中国建设科技出版社有限公司

单片机原理与应用

主 编 © 陈俊霖 唐运恩 高 勇

开门出书·创造价值
专·精·志·远
编 辑 部: 010-63567684
事业发展中心: 010-63567692
网 上 书 店: www.jkjd.com



定价: 49.80元

中国建设科技出版社有限责任公司
China Construction Science and Technology Press Co., Ltd.



计算机类创新融合精品规划教材
“互联网+”教育改革新理念教材

单片机原理与应用

主 编 © 陈俊霖 唐运恩 高 勇

中国建设科技出版社有限责任公司
China Construction Science and Technology Press Co., Ltd.

北 京

图书在版编目 (CIP) 数据

单片机原理与应用 / 陈俊霖, 唐运恩, 高勇主编.
北京: 中国建设科技出版社有限责任公司, 2025. 8.
(计算机类创新融合精品规划教材) (“互联网+” 教育改
革新理念教材). --ISBN 978-7-5160-4390-5

I. TP368.1

中国国家版本馆 CIP 数据核字第 2025C5K051 号

单片机原理与应用

DANPIANJI YUANLI YU YINGYONG

主 编 陈俊霖 唐运恩 高 勇

出版发行: 中国建设科技出版社 有限责任公司

地 址: 北京市西城区白纸坊东街 2 号院 6 号楼

邮 编: 100054

经 销: 全国各地新华书店

印 刷: 唐山唐文印刷有限公司

开 本: 889mm×1194mm 1/16

印 张: 14

字 数: 320 千字

版 次: 2025 年 8 月第 1 版

印 次: 2025 年 8 月第 1 次

定 价: 49.80 元

本社网址: www.jskjcbs.com, 微信公众号: zgjskjcbs

请选用正版图书, 采购、销售盗版图书属违法行为

版权专有, 盗版必究。本社法律顾问: 北京天驰君泰律师事务所, 张杰律师

举报信箱: zhangjie@tiantailaw.com 举报电话: (010) 63567684

《单片机原理与应用》编委会

主 审：赵 英

主 编：陈俊霖 唐运恩 高 勇

副主编：董树峰 皮依标 张继飞 居剑文

苏 娟 薛 智 陈 聪 吴琼飞

罗 潜 闫化龙

PREFACE



本书结合新的课程体系和教学内容改革需要，在多年教学实践的基础上编写而成。本书以 MCS-51 系列单片机为例，系统、全面地介绍单片机的原理、接口及应用技术。书中通过一些经典例题和应用实例，引导读者熟悉和理解单片机基本原理，逐步掌握单片机应用系统设计开发的基本知识、方法和应用技能。各任务后都配有拓展训练，以巩固所学的知识，帮助读者深入学习。

本书的编写具有以下特点：

- (1) 系统性强，内容全面、概念清楚，能使建立起单片机完整的总体概念。
- (2) 基础性强，对基础理论部分精心讲解，可为学生进一步学习后续课程打下良好基础。

单片机课程的特点是理论性和实践性都很强，而对大部分工科专业，本课程的目标是培养创新型和应用型人才，因此理论内容安排以够用为度，加大了典型例题的介绍。本书所有例题均是上机调试通过的。通过大量例题的练习，初学者能够掌握单片机的基本原理和软硬件系统的设计方法。

由于编者水平有限，书中难免有不足和疏漏之处，恳请广大读者批评指正。

编 者
2024 年 8 月

CONTENTS



项目一

1

单片机最小系统构建的认识

- 任务一 单片机概述 /2
- 任务二 单片机中数据使用 /5
- 任务三 单片机内程序和数据存储 /8
- 任务四 单片机最小系统构建的操作 /15

项目二

25

单片机开发工具应用

- 任务一 目标代码程序生成 /26
- 任务二 程序下载 /32
- 任务三 单片机简单电路运行仿真 /35

项目三

40

LED 流水灯设计制作

- 任务一 LED 点亮控制 /41
- 任务二 LED 闪烁控制 /47
- 任务三 延时时间调整及测定 /57
- 任务四 LED 流水灯程序设计 /65
- 任务五 任意花样 LED 闪烁控制 /75
- 任务六 LED 呼吸灯程序设计 /82

项目四

85

简易计算器设计制作

- 任务一 一位数码显示 /86
- 任务二 四位数码显示 /90

- 任务三 独立按键识别 /96
- 任务四 矩阵按键识别 /105
- 任务五 简易计算器设计 /110

项目五

115

里程表设计制作的认识

- 任务一 外部中断应用 /116
- 任务二 里程表设计制作的操作 /123

项目六

126

秒表设计制作的认识

- 任务一 精准方波发生器设计制作 /127
- 任务二 秒表设计制作的操作 /134

项目七

146

LED 电子显示屏设计制作

- 任务一 8×8 点阵显示屏设计 /147
- 任务二 16×16 点阵显示屏设计制作 /151
- 任务三 显示内容远程控制 /158

项目八

172

简易仪器仪表设计制作

- 任务一 简易信号发生器设计制作 /173
- 任务二 直流数字电压表设计制作 /182
- 任务三 液晶显示器接口电路设计制作 /191

附录

204

- 附录 A ASCII 码表 /204
- 附录 B C51 中的关键字 /205
- 附录 C C51 库函数 /208

参考文献

216

项目一

单片机最小系统构建的认识

欢迎走进精彩的单片机世界。本项目通过一个单片机最小应用系统的构建，使学生能够对单片机有初步的了解，能叙述单片机的内部结构，能设计包括电源、晶体振荡电路、复位电路和简单外围器件的最小单片机系统。





任务一 单片机概述

一、任务要求

所谓单片机，就是采用超大规模集成电路技术把具有数据处理能力的中央处理器（CPU）、随机存储器（RAM）、只读存储器（ROM）、多种 IO 口和中断系统、定时器/计数器等功能集成到一块硅片上构成的一个小而完善的计算机系统。通俗地说，单片机就是一块集成芯片，在满足一定的工作条件下，它的各引脚可以根据使用者的需要，通过编写合适的程序输出高、低两种电平，从而控制外围器件工作，实现预期的功能。

开发一个单片机系统首先要选择单片机器件的型号，本任务的主要目的是帮助读者选择适合的单片机，并使读者对单片机应用领域、单片机型号和包括硬件软件设计的系统开发等有初步了解。

二、单片机的应用领域和类型

（一）单片机的典型应用领域

本书中所有项目的实现，均以单片机为核心，这是以单片机的广泛应用为前提的。在我国，单片机技术人员数以万计。随着单片机的不断发展、完善，它已成为科技领域的智能化工具。

目前单片机已渗透到人们生活的各个领域，几乎很难找到哪个领域没有单片机的踪迹。可以说，凡是在需要控制和简单计算的电子设备和产品中都可以由单片机来实现。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，民用豪华轿车的安全保障系统，录像机、摄像机、全自动洗衣机的控制，以及程控玩具、电子宠物等，这些都离不开单片机，更不用说自动控制领域的机器人、智能仪表、医疗器械了。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、工程师。科技越发达，智能化的东西就越多，使用的单片机就越多。所以，学习单片机是社会发展的需求。

（二）单片机的类型

据不完全统计，全世界嵌入式微处理器的品种数量已经超过 1000 种，流行体系结构有 30 多个，其中 MCS-51 体系占大多数，所以本书选用 MCS-51 单片机进行阐述。

生产 MCS-51 单片机的半导体厂家有 20 多家，共有 350 多种衍生产品。在 ATMEL 公司



单片机的类型



成功推出 AT89 系列单片机后，几个著名的半导体生产商也相继生产了类似的产品，如 PHILIPS 公司的 P89 系列单片机、SST 公司的 SST89 系列单片机、Winbond 公司的 W78 系列单片机等。其中，AT89C51、P89C51、SST89E51、W78E51 都是与 MCS-51 系列完全兼容的单片机，各芯片之间也是兼容的，所以不写前缀，仅写 89C51 可以是其中任何一个厂家的产品。本书如不做特殊声明，使用单片机的型号均为 89C51。

除了 MCS-51 系列单片机，还有一些其他的单片机，如 PIC、AVR 和 MSP430 等系列。这些系列单片机和 MCS-51 系列单片机不兼容，程序指令也不相同。

AVR 系列单片机也是 ATMEL 公司生产的一种 8 位单片机，它采用 RISC 精简指令集，它的一条指令的运行速度可以达到纳秒级，速度快，功耗低，片内资源丰富，一般都集成模/数转换器、PWM、SPI、USART、I2C 等资源，大大简化了外围电路的设计，AVR 单片机是 8 位单片机中的高端产品，由于它的出色性能目前应用范围越来越大；MSP430 单片机是美国 TI 公司生产的，它采用的是 RISC 的指令集，这款单片机除了资源丰富，其主要特点是超低功耗，但是多数都内存不大；PIC 系列单片机，它是美国 MICROCHIP 公司生产的另一种 8 位单片机，它采用的也是 RISC 的指令集，资源较丰富，而且型号非常多，适用于不同场合。对于初学者，笔者希望不要盲目地追赶技术，而是应踏实地学好基础，这样才能触类旁通，解决开发中的问题。

三、硬件电路设计

单片机作为一片集成芯片，其应用系统离不开硬件设计，即电路设计。和普通电路设计过程一样，单片机系统的硬件设计一般遵循下面的步骤。

（一）电路原理图

利用电路设计软件，如 PROTEL 等，将单片机设计方案的硬件部分用标准的电路原理图表示，为电路板（PCB）图的生成提供依据。

（二）电路板图

根据电路原理图，利用电路设计软件，生成电路板图。主要实现元器件在电路板上的分布、具体封装，信号线、电源线与地线的走线等。其中需要考虑产品本身的尺寸要求、工作环境、干扰等问题。

（三）制板

根据电路板图，加工印刷电路板，完成元器件安装焊接。

硬件设计人员应对电路板进行检测，以发现其中的设计缺陷，如果不能通过修改电路板来补救出现的问题，则只能重新进行设计。



在电子行业，PROTEL 是广泛使用的一种计算机辅助设计（CAD）软件，它包含了电路原理图绘制、模拟电路与数字电路混合信号仿真、多层印制电路板设计（包含印制电路板自



动布线)、可编程逻辑器件设计、图表生成、电子表格生成、支持宏操作等功能。

PCB (Printed Circuit Board), 中文名称为印制电路板, 它将零件与零件之间复杂的电路铜线, 经过细致整齐的规划后, 蚀刻在一块板子上, 提供电子零组件在安装与互连时的主要支撑体, 是所有电子产品不可或缺的基础零件。由于它是采用电子印刷术制作的, 因此又称为印刷电路板。

覆铜板是以环氧树脂等为融合剂将玻纤布和铜箔压合在一起的产物, 是 PCB 的直接原材料, 在经过蚀刻、电镀、多层板压合之后制成印刷电路板。印刷电路板上通常都有设计预钻孔以安装芯片和其他电子组件, 将电子组件的接脚穿过 PCB 后, 再以导电性的金属焊条黏附在 PCB 上而形成电路。

印制电路板的设计是以电路原理图为根据, 实现电路设计者所需要的功能。印刷电路板的设计主要指版图设计, 需要考虑外部连接的布局、内部电子元件的优化布局、金属连线和通孔的优化布局、电磁保护、热耗散等各种因素。优秀的版图设计可以节约生产成本, 达到良好的电路性能和散热性能。

单面板 (Single-Sided Boards): 在最基本的 PCB 上, 零件集中在其中一面, 导线则集中在另一面上, 适用于简单的电路。双面板 (Double-Sided Boards): 这种电路板的两面都有布线, 不过要用上两面的导线, 必须要在两面间有适当的电路连接才行。这种电路间的“桥梁”叫作过孔 (via)。过孔是在 PCB 上, 充满或涂上金属的小洞, 它可以与两面的导线相连接。因为双面板的布线面积比单面板大了一倍, 而且布线可以互相交错 (可以绕到另一面), 它更适合用在比单面板更复杂的电路上。

四、软件设计

单片机与普通电子器件不同的是具有可编程性质, 即除搭建硬件电路外, 还需软件来与之协同工作。即使同一个电路, 当运行不同的程序时, 完成的功能也有所差别。这个特点使开发人员感到单片机具有挑战性、想象性和趣味性而对它爱不释手, 同时单片机系统因体现了设计人员的思想更具灵活性和智能性。目前, 具有可编程性质的电子器件是电子技术发展的主流方向。

单片机应用系统的软件设计千差万别, 不存在统一模式。开发一个软件的明智方法是尽可能采用模块化结构。根据系统软件的总体构思, 按照先粗后细的办法, 把整个系统软件划分成多个功能独立、大小适当的模块。划分模块时要明确规定各模块的功能, 尽量使每个模块功能单一, 各模块间的接口信息简单、完备, 接口关系统一, 尽可能使各模块之间的联系减少到最低限度。根据各模块的功能和接口关系, 可以分别独立设计。在各个程序模块分别进行设计、编制和调试后, 再将各个程序模块连接成一个完整的程序。

五、单片机系统的研制过程

解决问题时, 应采取一些特定的步骤。合理地把握单片机开发的各个阶段, 有助于设计流程与质量的保证。

如果不知道想做什么、完成何任务, 将无法找到解决方案, 所以单片机应用系统的设计



是以确定系统的功能和技术指标开始的。首先要细致分析、研究实际问题，明确各项任务和要求，再从考虑系统的先进性、可靠性、可维护性以及成本、经济效益出发，拟定出合理可行的技术性能指标。

对于一个实际应用的单片机系统，任务相对复杂，所需资源较多，不适合初学者完成，所以本书所选取的例题、任务和项目都比较简单，但这恰是解决复杂实际问题的前提和基础。

确定了研制任务后，就可以进行系统的总体方案设计。一个好的方案，会花费大量的时间和工作，它将承担起工作蓝图的职责。对于相同的功能，特定的开发小组或不同的决策者可能会选择不同的解决方案，但最终的选择不能违背产品设计的初衷。

在完成系统硬件和软件设计后，还需进行系统集成调试。此过程是将软件、硬件和执行装置集成在一起，进行系统调试，开发人员需要对技术接口进行逐一确认，发现并改进单元设计中的错误。

对于电路设计软件和制板过程，可参考其他资料，本书重点说明单片机及其外围器件电路的设计和单片机程序的编制。

六、拓展训练

1. 利用图书馆、互联网等资源查找 Motorola、Microship、Epson、华邦等公司各自生产的单片机的特点。
2. 一个完整的单片机应用系统的研制过程包含哪些环节？
3. 利用课余时间，学习电子绘图软件 PROTEL 的使用。

任务二 单片机中数据使用

一、任务要求

无论学习 89C51 单片机的原理知识还是编程语言，总离不开各种数据。日常生活中应用最多的是十进制数。除此之外，计算机中常用的还有二进制、十六进制等。通过本任务的学习，读者能够习惯使用二进制、十六进制表示数据，学会各种进制数之间的相互转换。

二、单片机中数据的表示形式

（一）十进制数

十进制是日常生活和生产中最常用的计数体制。它的每一位用 0~9 十个数码表示，计数



的基数为 10，超过数码 9 的数则需要用多数位表示，其中相邻位数间的关系是逢十进一或借一当十。如十进制数 368，则 3 为包含 10^2 的个数，6 为包含 10^1 的个数，而 8 为包含 10^0 的个数，因此 368 可表示为

$$368 = 3 \times 10^2 + 6 \times 10^1 + 8 \times 10^0$$

(二) 二进制数

对于一个电子电路非常容易实现两种不同且互为相反的状态，如电平的高低、灯的亮灭、三极管的通断，如果将其中的一种状态看成“1”，那么另外一种状态就可以看成“0”。这就是计算机系统中采用二进制计数的原因，可以使电路变得简单且容易实现。

在二进制数中，每一位仅有 1 和 0 两个数码之一，所以计数的基数为 2，相邻数位间的关系是逢二进一或借一当二。可用后标字母 B 表示二进制数，如 1101B、1001B。

二进制数所对应的十进制数，可将其按权展开得到，如：

$$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = (11)_{10}$$

对于十进制整数，可采用“除 2 取余，逆序排列”法换算为二进制数，例如将十进制数 29 转换为二进制数的步骤可表示为：

2		29	余数为	1	↑ 逆 序 排 列
2		14	余数为	0	
2		7	余数为	1	
2		3	余数为	1	
2		1	余数为	1	
0					

换算结果 $(29)_{10} = (11101)_2$ 。由此可见，将十进制数不断用 2 去除，直到商为 0，从下往上写出余数，即为所求的二进制数。

二进制数的位是计算机中数据的最小单位，8 个二进制位构成 1 个字节。1 个字节可以表示 2^8 ($=256$) 个不同的值 (0~255)。字节中的位号从左至右依次为 7~0。第 7 位称为最高有效位 (MSB)，第 0 位称为最低有效位 (LSB)。当数值大于 255 时，就要用字或双字进行表示。字可以表示 2^{16} ($=65536$) 个不同的值 (0~65535)，这时第 15 位为 MSB。

(三) 十六进制数

二进制数与十进制数比较，人们使用不太方便，与等值十进制数相比，它所需要的位数多。例如，十进制数 9 是一位数，而等值的二进制数 1001B 则为四位，读写不方便。为了读写方便，常把二进制数改写成十六进制，前面加“0x”是十六进制数的书写格式，如 0x3A；在汇编语言中用后标字母 H 表示十六进制数，如 3AH。为便于对照，将部分十、二、十六进制等值关系列于表 1-1 中。



表 1-1 常用计数体制数的表示方法

十进制数	二进制数	十六进制数	十进制数	二进制数	十六进制数
0	0000	0x00	10	1010	0x0A
1	0001	0x01	11	1011	0x0B
2	0010	0x02	12	1100	0x0C
3	0011	0x03	13	1101	0x0D
4	0100	0x04	14	1110	0x0E
5	0101	0x05	15	1111	0x0F
6	0110	0x06	16	0001 0000	0x10
7	0111	0x07	64	0100 0000	0x40
8	1000	0x08	255	1111 1111	0xFF
9	1001	0x09	1000	1111101000	0x3E8

十六进制是在计算机指令代码和数据的书写中经常使用的数制，采用 0~9、A、B、C、D、E、F 十六个数码，计数的基数为 16，相邻位数间的关系是逢十六进一或借一当十六。二进制数转换为十六进制数时，对于整数只要将二进制数从最低有效位往左每四位分成一组，每组用一个等价的十六进制数来代替即可。例如，将 $(1111101000)_2$ 转换为十六进制数可表示为

$$\frac{11}{3} \quad \frac{1110}{E} \quad \frac{1000}{8}$$

所以， $(1111101000)_2 = (3E8)_{16}$ 。由于 4 位二进制数可以方便地用 1 位十六进制数表示，因此人们对二进制的代码或数据常用十六进制形式缩写。十六进制数转换为二进制数的方法是把十六进制数的每位分别用 4 位二进制数码表示，然后把它们排列起来。例如，把十六进制数 2AC 转换为一个二进制数可以表示为

$$\frac{2}{0010} \quad \frac{A}{1010} \quad \frac{C}{1100}$$

所以， $(2AC)_{16} = (1010101100)_2$ 。

十六进制数转换成十进制数，方法和二进制数转换成十进制数的方法类似，采用按权展开的方法。例如：

$$(3F)_{16} = 3 \times 16^1 + 15 \times 16^0 = 48 + 15 = (63)_{10}$$

十进制整数转换成十六进制整数可以采用“除 16 取余，逆序排列”法，即用 16 连续去除要转换的十进制整数，直到商为 0，然后把得到的余数按逆序排列，便是所求的十六进制数。例如， $(3901)_{10}$ 所对应的十六进制数可表示为

$$\begin{array}{rcl} 16 & | & 3901 \quad \text{余数 } 13 \text{ 写作 } D \\ 16 & | & 243 \quad \text{余数 } 3 \text{ 写作 } 3 \\ 16 & | & 15 \quad \text{余数 } 15 \text{ 写作 } F \\ & & 0 \end{array} \quad \begin{array}{c} \uparrow \\ \text{逆} \\ \text{序} \\ \text{排} \\ \text{列} \end{array}$$

把所得余数按从下到上排列起来便可得到： $(3901)_{10} = (F3D)_{16}$ 。

对于各进制小数部分的相互转换，由于在单片机系统中很少应用，这里不再进行详细介绍。



(四) ASCII 码

在计算机系统中,有时需要把数及数以外的其他信息(如字符或字符串)用一个规定的二进制数来表示。二进制数只有 0 和 1 两个数码,把若干个 0 和 1 按一定规律编排起来,用来表示某种信息含义的一串符号称为代码。这些二进制形式的代码称为二进制编码。

字符的编码经常采用的是美国标准信息交换代码(ASCII 码),用于计算机与外围设备的数据传输,以引号括起来标识。如:"AB"。一个字节的 8 位二进制码可以表示 256 个字符。当最高位为“0”时,所表示的字符为标准 ASCII 码字符,共有 128 个,分为两类:一类是图形字符(96 个),用于表示数字、英文字母、标点符号等;另一类是控制字符,包括回车符、换行符等,没有特定形状,其编码可以存储、传送和起某种控制作用。当最高位为“1”时,所表示的是扩展 ASCII 码字符。标准 ASCII 码字符表见附录 A。

三、拓展训练

1. 利用十六进制数从 0x00 写到 0x100。
2. 把下列十进制数转换为二进制数和十六进制数。
①36 ②128 ③4096
3. 把下列十六进制数转换为二进制数和十进制数。
①2AH ②FFH ③100H

任务三 单片机内程序和数据存储

一、任务要求

为了能充分发挥单片机的功能,必须知道单片机是如何进行程序和数据存储的,即对其内部结构进行必要的了解,尤其是存储器结构,是正确使用单片机必要的基础知识。本任务重点对 89C51 单片机内部存储器结构进行讨论。

假设单片机系统应用程序 1 KB,其中主程序 200 B,定时器 0 中断服务程序 100 B,其他子程序 700 B;原始数据 200 B,中间数据 10 B,位数据 8 位,分别分配存储空间,画出分配示意图。



二、知识链接

（一）89C51 单片机内部结构

89C51 单片机的内部结构如图 1-1 所示。主要包括 CPU、内部存储器、定时与中断系统、并行 IO 口、串行 IO 口和时钟电路等六部分。

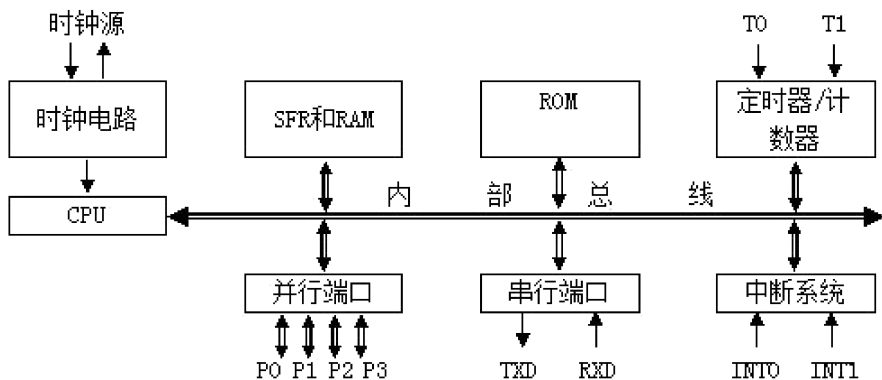


图 1-1 89C51 单片机内部结构框图

1. CPU

CPU 也叫中央处理器，是单片机的核心部件，主要用于完成单片机的运算和控制功能。其内部由运算器和控制器组成。

2. 内部存储器

单片机的内部存储器包括程序存储器（ROM）和数据存储器（RAM），它们是相互独立的。程序存储器（ROM）为只读存储器，用于存放程序指令、原始常数及数据表格。数据存储器（RAM）为随机存储器，用于存放数据。

3. 定时与中断系统

89C51 单片机内部有两个 16 位的定时器/计数器，用于实现定时或计数功能，并以定时或计数的结果（查询或中断方式）来实现控制功能。

89C51 单片机具有中断功能，可满足控制应用的需要。它共有 5 个中断源（89C52 单片机有 6 个中断源），即两个外部中断源、两个定时器/计数器中断源和一个串行口中断源。全部中断可分为高级和低级两个优先级别。

4. 并行 IO 口

89C51 单片机内部共有 4 个 8 位的并行 I/O 口（P0、P1、P2、P3），用于实现数据的并行输入和输出。

5. 串行 IO 口

89C51 单片机还有一个全双工的串行口，用于实现与外部的串行数据传送和与外部设备串行通信。

6. 时钟电路

时钟电路为单片机产生时钟脉冲序列，用于协调和控制其工作。89C51 单片机的内部有



时钟电路，在采用内部时钟时需要外接石英晶体振荡器和微调电容。

（二）单片机的工作过程和 ROM

1. 地址和数据

在计算机系统中，除要求包含 CPU 外，必须要有存储器，用来存储程序和各种数据。存储器中最小的存储单位是一个二进制位，8 位二进制位即一个字节构成一个存储单元。一个存储器包含许多存储单元，每个存储单元在存储器中所处的位置是以地址标识的，一个存储单元对应一个地址，可以将地址理解为每个存储单元前面的编号。不同地址根数，所确定的存储空间也不一样。一根地址线，对应的存储空间只有两个存储单元，其中一个存储单元的编号为 0，另一个存储单元的编号为 1；两根地址线，对应的存储空间有四个存储单元，各存储单元的编号依次为 00B、01B、10B 和 11B，以此类推， n 根地址线，对应的存储空间有 $2n$ 个存储单元，存储单元的编号范围是 $0 \sim (2n-1)$ 。89C51 单片机提供 16 根地址线，最大寻址空间为 64 K ($0x0000 \sim 0xffff$)。

每个存储单元里存储的信息为数据，89C51 单片机中，一个存储单元里能够存储 8 位二进制信息，即一个字节 (Byte，简称为 B)。89C51 单片机的 CPU 一次所能处理的数据也是 8 位二进制数，所以说它是 8 位机。

2. 单片机的工作过程

单片机的工作过程，也就是自动执行程序的过程，即一条条执行指令的过程。所谓指令就是把要求单片机执行的各种操作命令的形式写下来，一条指令对应着一种基本操作。一系列指令的集合就称为程序，为使单片机能自动完成某一特定任务，必须编写相应的程序。

程序通常是顺序执行的，所以程序中的指令也是一条条顺序存放的，单片机在执行程序时要能把这些指令一条条取出并加以执行，必须有一个部件能追踪指令所在的地址，这一部件就是程序计数器 PC (包含在 CPU 中)，在开始执行程序时，给 PC 赋以程序中第一条指令所在的地址，然后取得该指令并执行其命令，同时 PC 中的内容自动增加，以指向下一条指令的起始地址，如此保证指令顺序执行。

PC 本身没有地址，是不可寻址的，因此用户无法对其进行读写操作。在转移、调用和返回等指令中，PC 的值会改变，以控制程序按用户的要求去执行。

3. 程序存储器 ROM

在单片机内部，专门用来存放程序的是程序存储器 (ROM)。ROM 是一种写入信息后不能改写，只能读出的存储器。断电后，ROM 中的信息保留不变，所以，ROM 用来存放固定的程序或数据。

89C51 是一种带 Flash ROM 的单片机，Flash ROM 是一种快速存储式只读存储器，这种程序存储器的特点是可以电擦写，掉电后程序依然保存，编程寿命可以达到一千次左右。早期的程序存储器还有掩膜 ROM、PROM、EPROM 三种。掩膜 ROM 中存储的信息是在制造过程中固化进去的，一旦固化便不能再修改，适合于大批量的定型产品；可编程只读存储器 PROM 中的信息可由用户通过特殊手段写入，但只能一次性写入；EPROM 是可擦除、可改写的 ROM，用户可根据需要对其多次写入和擦除，但可以擦除的次数也是有限的，一般为几十次。

不同型号的单片机，其内部存储器 ROM 容量也有所差异，89C51 单片机芯片内部 ROM



容量只有 4 K (0x0000~0x0fff)，宏晶公司生产的 STC89E516 单片机的内部 ROM 容量已达到 63 KB。

单片机复位后，程序计数器 PC 为 0000H，即从程序存储器的 0000H 单元读出第一条指令。程序存储器中，某些单元保留给系统使用，用来存放特定的系统程序，见表 1-2。分析表 1-2 可发现，系统为每一个应用程序保留单元数都较少。如仅为复位后引导程序保留了 3 个单元，为了保证程序的正常存放及运行，在 0000H 单元内放置一条跳转指令，如 LJMP $\times\times\times\times$ ($\times\times\times\times$ 表示主程序入口地址)，然后把这个程序存放到 $\times\times\times\times$ 位置。

表 1-2 保留的存储单元

存储器单元	指定存放的系统程序
0000H~0002H	复位后初始化引导程序
0003H~000AH	外部中断 0 中断服务程序
000BH~0012H	定时器 0 溢出中断中断服务程序
0013H~001AH	外部中断 1 中断服务程序
001BH~0022H	定时器 1 溢出中断中断服务程序
0023H~002AH	串行端口中断中断服务程序
002BH~0032H	定时器 2 溢出中断中断服务程序

(三) 数据存储器 RAM

在程序的执行过程中，总有一些暂时性的数据或中间结果等信息需要存储。而 ROM 中的内容在单片机工作状态下是不允许更改的，为此单片机中专门设立了数据存储器 RAM，在关闭电源时，其所存储的信息将丢失。

单片机的数据存储器，分为片内 RAM 和片外 RAM 两大部分。

1. 内部数据存储器

89C51 单片机内部数据存储器在结构上可以分为两个不同的存储空间，即低 128 单元的数据存储器空间 (0x00~0x7F) 和高 128 单元的具有特殊功能的专用寄存器存储器空间 (0x80~0xFF)。这两个空间是连续的，但只有低 128 单元才能真正地作为数据存储器提供给用户使用。在单片机中，尽管片内 RAM 的容量不大，但它的功能多，使用灵活。128 B 的片内 RAM 分成工作寄存器区、位寻址区、通用 RAM 区三部分，如图 1-2 所示。

(1) 工作寄存器区。

片内 RAM 中，0x00~0x1F 的 32 个单元

7FH									数据区域
30H									
2FH	7F	7E	7D	7C	7B	7A	79	78	位寻址区
2EH	77	76	75	74	73	72	71	70	
2DH	6F	6E	6D	6C	6B	6A	69	68	
2CH	67	66	65	64	63	62	61	60	
2BH	5F	5E	5D	5C	5B	5A	59	58	
2AH	57	56	55	54	53	52	51	50	
29H	4F	4E	4D	4C	4B	4A	49	48	
28H	47	46	45	44	43	42	41	40	
27H	3F	3E	3D	3C	3B	3A	39	38	
26H	37	36	35	34	33	32	31	30	
25H	2F	2E	2D	2C	2B	2A	29	28	
24H	27	26	25	24	23	22	21	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	16	15	14	13	12	11	10	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	
1FH	3组								工作寄存器区
	2组								
	1组								
	0组								
00H									

图 1-2 内部 RAM 分配



是 4 个通用工作寄存器组，每组有 8 个寄存器，即 R0~R7。编程时，寄存器常用于存放操作数及中间结果等。

延伸阅读

正在使用的寄存器组称为当前寄存器区，选择哪个工作组为当前工作区由程序状态控制寄存器的 RS1 位和 RS0 位的状态来决定，用户可以在编程时用软件进行设置，切换当前工作寄存器区。当前寄存器的选择和寄存器组的地址见表 1-3。单片机上电复位后，工作寄存器为第 0 组。

表 1-3 RS1、RS0 与当前寄存器组的选择

RS1	RS0	当前寄存器组	R0~R7 的地址	RS1	RS0	当前寄存器组	R0~R7 的地址
0	0	第 0 组	0x00~0x07	1	0	第 2 组	0x10~0x17
0	1	第 1 组	0x08~0x0F	1	1	第 3 组	0x18~0x1F

(2) 位寻址区。

从 0x20~0x2F 的 16 个字节的 RAM 为位寻址区，有双重寻址功能，既可以进行位寻址操作，也可以同普通 RAM 单元一样按字节寻址操作，共有 128 位，每一位都有相对应的位地址，位地址范围从 0x00~0x7F。位寻址是单片机的一个重要的特点。所谓位操作，只是对该位进行操作，对本字节其他位没影响。

(3) 通用 RAM 区（数据缓冲器区）。

0x30~0x7F 共 80 个字节为数据缓冲器区，用于存放用户数据，只能按字节存取。通常这些单元可用于中间数据的保存，也用作堆栈的数据单元。

2. 特殊功能寄存器

特殊功能寄存器也叫专用寄存器（SFR），就是将内部 RAM 的高 128 个单元作为特殊功能寄存器使用，其单元地址为 0x80~0xFF。89C51 单片机的专用寄存器总数为 21 个，而 89C52 单片机为 26 个，这些寄存器仅占用了 0x80~0xFF 中的一小部分，其他空间虽然未安排寄存器，但也不能进行读写操作。特殊功能寄存器的名称、地址见表 1-4。在特殊功能寄存器中，有 11 个寄存器不仅可以进行字节寻址，还可以进行位寻址。能进行位寻址的寄存器的特点是字节地址都能被 8 整除（字节地址的末位是 0 或 8）。SFR 中的位寻址分布见表 1-5。

表 1-4 特殊功能寄存器一览表

符号	地址	名称	符号	地址	名称
ACC	0xE0	累加器	IE	0xA8	中断允许控制器
B	0xF0	B 寄存器	TMOD	0x89	定时器方式选择
PSW	0xD0	程序状态字	TCON	0x88	定时器控制器
SP	0x81	堆栈指针	TH0	0x8C	定时器 0 高 8 位
DPL	0x82	数据寄存器指针（低 8 位）	TL0	0x8A	定时器 0 低 8 位
DPH	0x83	数据寄存器指针（高 8 位）	TH1	0x8D	定时器 1 高 8 位
P0	0x80	通道 0	TL1	0x8B	定时器 1 低 8 位
P1	0x90	通道 1	SCON	0x98	串行控制器



续表

符号	地址	名称	符号	地址	名称
P2	0xA0	通道 2	SBUF	0x99	串行数据缓冲器
P3	0xB0	通道 3	PCON	0x87	电源控制器
IP	0xB8	中断优先级控制器	—	—	—

表 1-5 SFR 中的位寻址分布表

寄存器号	D7	D6	D5	D4	D3	D2	D1	D0	字节地址
B	F7	F6	F5	F4	F3	F2	F1	F0	0xF0
ACC	E7	E6	E5	E4	E3	E2	E1	E0	0xE0
PSW	D7	D6	D5	D4	D3	D2	D1	D0	0xD0
IP	—	—	—	BC	BB	BA	B9	B8	0xB8
P3	B7	B6	B5	B4	B3	B2	B1	B0	0xB0
IE	AF	—	—	AC	AB	AA	A9	A8	0xA8
P2	A7	A6	A5	A4	A3	A2	A1	A0	0xA0
SCON	9F	9E	9D	9C	9B	9A	99	98	0x98
P1	97	96	95	94	93	92	91	90	0x90
TCON	8F	8E	8D	8C	8B	8A	89	88	0x88
P0	87	86	85	84	83	82	81	80	0x80

专用寄存器 P0、P1、P2 和 P3 分别是 4 个并行 I/O 端口寄存器，可实现数据从相应端口的输入/输出，既可按字节寻址，也可按位寻址。

延伸阅读

特殊功能寄存器（SFR）的每一位的定义和作用与单片机各部件直接相关。这里先对部分 SFR 作一简要的说明，详细的用法将在相应的任务进行讨论。

(1) 累加器 ACC。简称累加器 A，是在编程操作中最常用的专用寄存器，功能较多，可按位寻址。除了在传送指令中的运用较多，89C51 单片机在进行各种算术和逻辑运算时，大部分单操作数指令的操作数就取自累加器，而且许多双操作数指令中的一个操作数也取自累加器。例如，加、减、乘、除运算指令的运算结果都存放在累加器 A 或 AB 寄存器中。

(2) B 寄存器。B 寄存器是一个 8 位寄存器，既可以作为一般寄存器使用，也可用于乘除运算。做乘法运算时，B 是乘数，且操作后，乘积的高 8 位存放在 B 中；做除法运算时，B 存放除数，且操作后，余数存放在 B 中。

(3) 程序状态字（PSW）。程序状态字用于存放程序运行的状态信息，可按位寻址，这些位的状态通常是指令执行过程中自动形成的，以供程序查询和判别。其中 PSW.1 为保留位，未用。各标志位的说明见表 1-6。各标志位定义如下：

CY	AC	F0	RS1	RS0	OV		P
----	----	----	-----	-----	----	--	---



表 1-6 程序状态字各标志位的说明

位	标志	名称	功能
PSW. 7	CY	进位标志位	存放算术运算的进位标志 在布尔运算中作为累加器使用
PSW. 6	AC	辅助进位标志位	做 BCD 运算时, 若低 4 位向高 4 位进位或借位, 则置 1
PSW. 5	F0	用户标志位	用户可以用软件自定义的一个状态标记
PSW. 4	RS1	当前寄存器区选择位	见表 1-3
PSW. 3	RS0	当前寄存器区选择位	见表 1-3
PSW. 2	OV	溢出标志位	做算术运算时, OV=0, 未溢出 做算术运算时, OV=1, 溢出
PSW. 1		保留位	
PSW. 0	P	奇偶标志位	若 P=1, 则累加器 A 中 1 的个数为奇数 若 P=0, 则累加器 A 中 1 的个数为偶数

(4) 堆栈指针 (SP)。堆栈指针操作是在内存 RAM 区中专门开辟出来的按照“先进后出, 后进先出”的原则进行数据存取的一种工作方式。主要用于子程序调用及返回和中断断点处理的保护及返回, 在完成子程序嵌套和多重中断处理中是必不可少的。为保证逐级正确返回, 进入栈区的“点”数据应遵循“先进后出, 后进先出”的原则。SP 用来指示堆栈所处的位置, 在进行操作之前, 先用指令给 SP 赋值, 以规定栈区在 RAM 区的起始地址 (栈底)。当数据推入栈区后, SP 的值也自动随之变化。

系统复位后, SP 初始化为 0x07, 如果不重新设置, 就会使堆栈由 0x08 单元开始。因为 0x08~0x1F 单元属于工作寄存器区, 所以在程序设计过程中, 最好把 SP 的值设置得大一些, 一般将堆栈开辟在 0x30~0x7F 区域中。

(5) 数据指针 (DPTR) 寄存器。数据指针是由两个 8 位寄存器 (DPH 和 DPL) 组合而成的一个 16 位专用寄存器, 其中 DPH 为 DPTR 的高 8 位, DPL 为 DPTR 的低 8 位。它既可作为 16 位寄存器来使用, 也可作为两个独立的 8 位寄存器 (DPH 和 DPL) 来使用。DPTR 用于存放 16 位地址指针, 既可访问数据存储器, 也可访问外部程序存储器。

3. 外部数据存储器

当进行大量连续的数据采集时, 单片机内部提供的数据存储器 (RAM) 是远远不够的, 这时可以利用单片机的扩展功能, 在芯片外部扩展数据存储器 (RAM)。单片机最大可扩展片外 64KB 空间的数据存储器, 地址范围为 0x0000~0xFFFF。

三、存储空间分配

一般单片机系统的应用程序和原始数据存储在程序存储器中, 系统的中间数据和位数据存储在数据存储器中。系统的子程序和原始数据一般无存储要求, 但主程序和中断服务程序系统有指定位置存放。图 1-3 是本任务的系统程序和原始数据的存储示意图。本任务的 8 位位数据可保存在片内数据存储器的 20H 单元中, 10 字节数据可存储在 30H 到 39H 这 10 个单



元中。当然，读者也可根据自己的习惯选择其他存储位置。

0x0000	转0x0100
0x000b	转0x0500
	⋮
0x0100	主程序
0x0300	子程序
0x0500	定时器0 中断服务程序
0x0600	原始数据
	⋮

图 1-3 系统程序和原始数据的存储示意图

四、拓展训练

1. 89C51 单片机内部结构包含哪几部分？各部分的主要功能是什么？
2. 单片机是如何实现自动运行程序的？
3. 对于容量有 4KB 的程序存储器，其地址线需要多少根？
4. 简述 89C51 单片机片内 RAM 的空间分配。各部分的主要功能是什么？
5. 片内 RAM 中包含哪些可位寻址单元？

任务四 单片机最小系统构建的操作

一、任务要求

89C51 单片机将 CPU、RAM、ROM、I/O 口及定时器/计数器等都集成在一块芯片内。作为一块电子芯片，在使用过程中，必须设计适当的外围电路，才能正常工作。本任务通过构建一个简单的单片机应用系统，帮助读者学会单片机工作所必需的电源地电路、晶体振荡电路和复位电路的设计，能进行发光二极管（LED）和单片机的连接。



二、单片机的引脚功能

对于一块集成电路，想要使用它，首先必须要知道它的引脚功能，才会连线，双列直插式（DIP）封装的 89C51 单片机的引脚排列如图 1-4 所示。

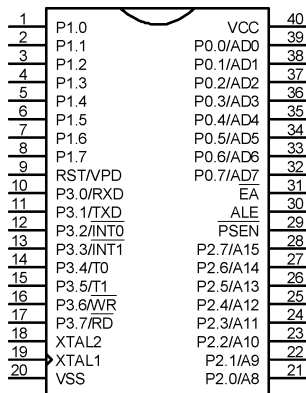


图 1-4 89C51 单片机引脚排列图

（一）电源 VCC 和接地 VSS 引脚

VCC（40 脚）：接电源+5 V。

VSS（20 脚）：接地，也就是 GND。

一块集成电路工作的前提条件之一是要有稳定的电源电压保障和可靠的接地。89C51 单片机标准电源电压值为 5 V，正负偏离一般不要超过 10%。通常得到 5 V 电源的方法是用变压器、整流电路、滤波电路和稳压电路，参考电子技术相关知识。还可以用一条 USB 线把电脑 USB 口的 5 V 电引到板子上使用，注意 USB 接口共有 4 根接线，两端的接线即为 5 V 电源。

（二）时钟引脚及电路

单片机内部由大量的时序逻辑电路构成，各电路都是在时钟脉冲的控制下，一步步完成各种操作的。单片机执行指令的过程就是顺序地从程序存储器 ROM 中一条一条取出指令，并进行指令译码，然后进行一系列的微操作控制，来完成各种指定的动作。它在协调内部的各种动作时必须要有有一定的顺序，换句话说，就是这一系列微操作控制信号在时间上要有一个严格的先后次序，这种次序就是单片机的时序。单片机的时钟信号用来为单片机芯片内部各种操作提供时间基准。

1. 时钟电路

单片机的 XTAL1（19 引脚）和 XTAL2（18 引脚）两个引脚就是专门为产生时钟振荡信号设立的，只要在两个引脚外接晶振就可以了，如图 1-5（a）所示。图中，晶振的振荡频率范围一般选择 4~12 MHz，典型值为 12 MHz 和 6 MHz，高速单片机可以工作在 32 MHz 的振荡频率下；电容器 C_1 和 C_2 的作用是稳定频率和快速起振，典型值为 30 pF，但在实际使用时，需要根据实际起振情况选择。

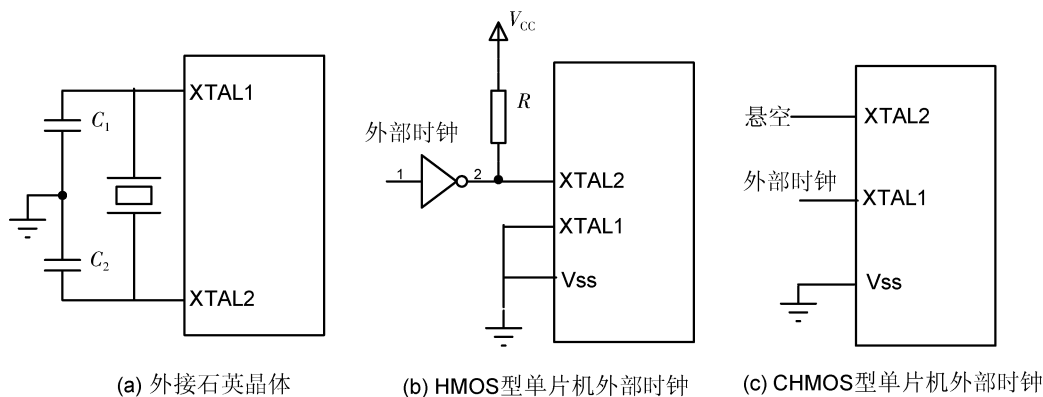


图 1-5 单片机的时钟电路

如果多片单片机同时工作，为了保证单片机的同步，可采用外部时钟方式，即把外部已有的时钟信号引入单片机内。一般要求外部信号高电平的持续时间大于 20 ns，且为频率低于 12 MHz 的方波。图 1-5 (b) 和图 1-5 (c) 给出了两种外部时钟电路，可以根据不同的单片机型号进行选择。

在制造上 MCS-51 系列单片机按两种工艺生产。一种是 HMOS 工艺，即高密度短沟道 MOS 工艺，在产品型号中不带有字母“C”的即为 HMOS 芯片，如 8051。另一种是 CHMOS 工艺，即互补金属氧化物的 HMOS 工艺，在产品型号中凡带有字母“C”的即为 CHMOS 芯片，它是 CMOS 和 HMOS 的结合，既保持了 HMOS 高速度和高密度的特点，又具有 CMOS 低功耗的特点。

2. 机器周期和指令周期

89C51 单片机包括 4 个定时单位，即振荡周期（节拍）、时钟周期（状态周期）、机器周期和指令周期。单片机的两种常用晶振的 4 个周期信号的对比见表 1-7。

表 1-7 常用晶振的 4 个周期信号的对比

晶振频率	振荡周期	时钟周期	机器周期	指令周期
6 MHz	$1/6 \mu\text{s}$	$1/3 \mu\text{s}$	$2 \mu\text{s}$	$2 \sim 8 \mu\text{s}$
12 MHz	$1/12 \mu\text{s}$	$1/6 \mu\text{s}$	$1 \mu\text{s}$	$1 \sim 4 \mu\text{s}$

(1) 振荡周期。

振荡周期也叫作节拍，用 P 表示，是指为单片机提供定时信号的振荡源的周期，由单片机振荡电路 OSC 产生，是最小的时序单位。

(2) 时钟周期。

时钟周期又叫作状态周期，用 S 表示，是振荡周期的两倍，其前半周期对应的节拍叫 P1 节拍，后半周期对应的节拍叫 P2 节拍。P1 节拍通常用于完成算术和逻辑运算，P2 节拍通常用于完成传送指令。

(3) 机器周期。

单片机每访问一次存储器的时间，称为一个机器周期，它是一个时间基准。89C51 单片机的一个机器周期的宽度由 6 个状态周期（12 个振荡周期）组成，并依次表示为 S1~S6，分别记作 S1P1、S1P2~S6P1、S6P2。如所用晶振为 12MHz，它的晶体振荡周期 $T=1/f$ ，即 $1/12 \mu\text{s}$ ，此时单片机的一个机器周期是 $12 \times (1/12) \mu\text{s}$ ，也就是 $1 \mu\text{s}$ 。



(4) 指令周期。

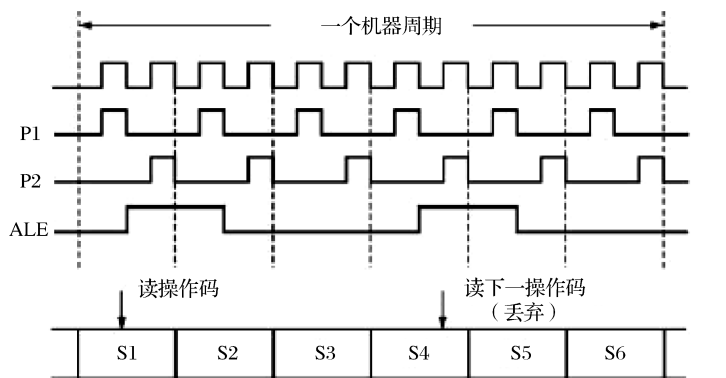
指令周期是最大的时序定时单位，是执行一条指令需要的时间。在单片机中，机器语言指令按执行时间可以分为三类：单周期指令、双周期指令和四周期指令。四周期指令只有乘、除法两条指令。如所用晶振为 12 MHz，对于双周期指令，执行一次需要 $2\ \mu\text{s}$ 的时间。

延伸阅读

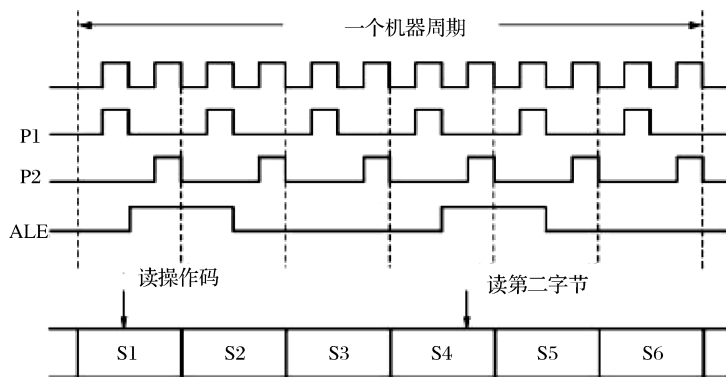
每一条指令的执行都可以分为取指和执行两个阶段。在取指阶段，CPU 从内部或外部 ROM 中取出需要执行的指令的操作码和操作数。在执行阶段对指令操作码进行译码，以产生一系列控制信号完成指令的执行。

按照指令字节数和机器周期数，89C51 单片机的 111 条指令可分为 6 类，分别对应 6 种基本时序。这 6 类指令是：单字节单周期指令、单字节双周期指令、单字节四周期指令、双字节单周期指令、双字节双周期指令和三字节双周期指令。为了弄清这些基本时序的特点，现将几种主要时序作一简述。

(1) 单周期指令时序。单周期指令的时序如图 1-6 所示。对于单字节单周期指令，在 S1P2 把指令码读入指令寄存器，并开始执行指令，但在 S4P2 读下一指令的操作码要丢弃，且 PC 不加 1。对于双字节单周期指令，在 S1P2 把指令码读入指令寄存器，并开始执行指令，在 S4P2 读入指令的第二字节。无论是单字节还是双字节均在 S6P2 结束该指令的操作。



(a) 单字节单周期指令



(b) 双字节单周期指令

图 1-6 单周期指令时序



(2) 双周期指令的时序。双周期指令的时序如图 1-7 所示。对于单字节双周期指令，在两个机器周期之内要进行 4 次读操作，只是后 3 次读操作无效。

在图 1-6 和图 1-7 中还示出了地址锁存允许信号 ALE 的波形。可以看出，在片外存储器不作存取时，每一个机器周期中 ALE 信号有效两次，具有稳定的频率。所以，ALE 信号是时钟振荡频率的 1/6，可以用作外围设备的时钟信号。

应注意的是，在对片外 RAM 进行读/写时，ALE 信号会出现非周期现象。访问片外 RAM 的双周期指令的时序如图 1-8 所示，在第二机器周期无读操作码的操作，而是进行外部数据存储器的寻址和数据选通，所以在 S1P2~S2P1 间无 ALE 信号。

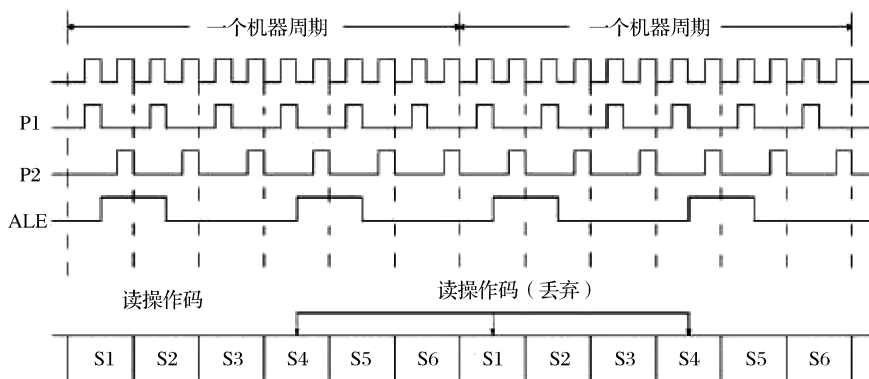


图 1-7 单字节双周期指令时序

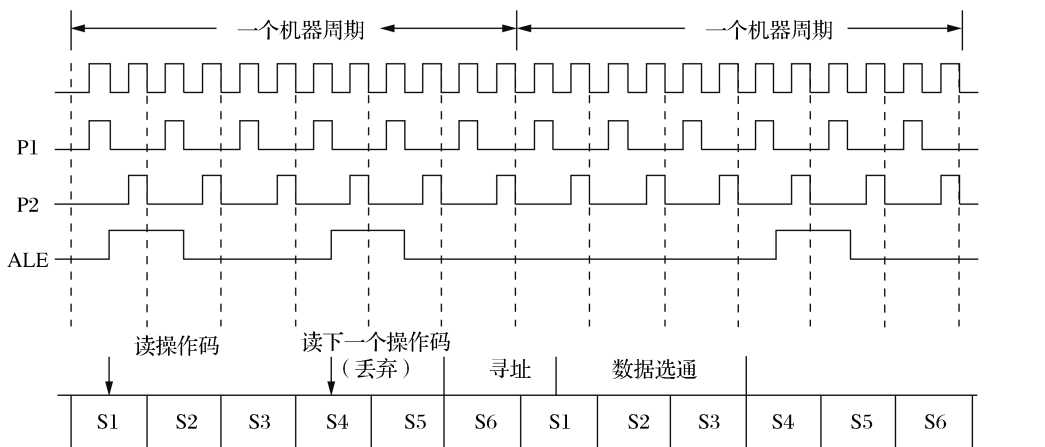


图 1-8 访问外部 RAM 的双周期指令时序

(三) 复位引脚及复位电路

复位就是使中央处理器 (CPU) 以及其他功能部件都恢复到一个确定的初始状态，并从这个状态开始工作。单片机在开机时或在工作中因干扰而使程序失控或工作中程序处于某种死循环状态等情况下都需要复位。单片机的复位靠外部电路实现，信号由 RST (RESET、9 引脚) 引脚输入，高电平有效，在振荡器工作时，只要保持 RST 引脚高电平两个机器周期，单片机即复位。复位后，程序计数器 PC 的内容为 0000H，即复位后将从程序存储器的 0000H 单元读取第一条指令码。其他特殊功能寄存器的复位状态见表 1-8。



表 1-8 复位后各寄存器的状态

寄存器	复位状态	寄存器	复位状态	寄存器	复位状态	寄存器	复位状态
PC	0000H	P0	FFH	IE	0X000000B	TL1	00H
ACC	00H	P1	FFH	TMOD	00H	TH1	00H
PSW	00H	P2	FFH	TCON	00H	SCON	00H
SP	07H	P3	FFH	TL0	00H	SBUF	不定
DPTR	0000H	IP	XX000000B	TH0	00H	PCON	0XX0000B

和时钟电路一样，复位电路也是单片机系统的正常运行必须的外部电路。复位电路一般采用上电复位电路，如图 1-9（a）所示，其工作原理是：系统通电瞬间，电容相当于短路，RST 引脚为高电平，然后电源通过电阻对电容充电，RST 端电压下降到一定程度，即为低电平，单片机开始正常工作。复位电路还可以选择上电和按键均有效的复位，如图 1-9（b）所示。其上电复位与前述相同，在单片机运行期间，手动复位时，单击复位按钮，电容 C 迅速放电，RST 端出现高电平，使单片机复位；复位按钮松开后，电容 C 通过 R_1 和内部下拉电阻充电，逐渐使 RST 端恢复为低电平。

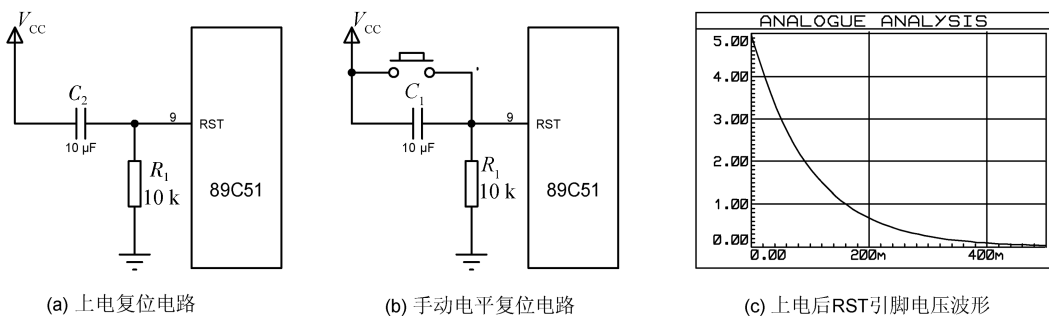


图 1-9 单片机的复位电路

（四）引脚

早期某些单片机片内没有或有容量很小的 ROM，在使用这些单片机时需要外扩 ROM。单片机是通过“EA 引脚”来控制选择内部和外部 ROM 的。当 $\overline{\text{EA}}=1$ 时，单片机读取内部 ROM 的地址范围，若超出范围，则 CPU 自动读取外部 ROM；当 $\overline{\text{EA}}=0$ 时，CPU 完全读取外部 ROM。随着技术的发展，现在单片机内部 ROM 容量已经完全够用，如 STC89C516RD+ 单片机供用户使用的片内 ROM 容量达到 63 K 字节，实际应用中无须扩展外部 ROM，所以将移作他用。

（五）输入/输出引脚

89C51 单片机共有 32 条并行双向 I/O 口线，分成 4 个 I/O 端口，记作 P0、P1、P2 和 P3。每个端口均由数据输入缓冲器、数据输出驱动及锁存器等组成。4 个端口在结构和特性上是基本相同的，但又各具特点。

1. P0 口

P0 口包括一个 D 锁存器、两个三态缓冲器、由一对场效应管组成的输出驱动电路，以及由一个与门、一个反向器和一路模拟转换开关（MUX）组成的输出控制电路。其特点如下：



- (1) P0 口可进行位寻址，其字节地址为 80H。
- (2) P0 口既可作为地址/数据总线使用，又可作为通用 I/O 口使用。若作为地址/数据总线使用，就不能再作为通用 I/O 口使用了。
- (3) P0 口为双向 I/O 口，输出漏极开路，可推动 8 个 TTL 电路。
- (4) P0 口作为双向 I/O 口使用时，需要外接上拉电阻。

P0 口的位结构如图 1-10 所示。当它访问外部程序存储器时，控制信号为 1，模拟转换开关将地址/数据信号接通，同时打开与门，输出的地址/数据信号经过反向器驱动 VT2，经过与门驱动 VT1，使两个 FET 构成推拉输出电路。若地址/数据信号为 1，则 VT1 导通，VT2 截止，引脚输出 1；若地址/数据信号为 0，则 VT1 截止，VT2 导通，引脚输出 0。访问外部存储器时，CPU 会自动向 P0 口的锁存器写 1，所以 P0 口作为地址/数据总线使用时是一个真正的双向口。

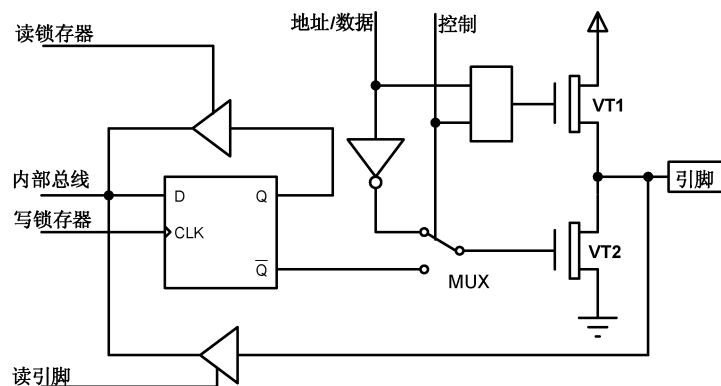


图 1-10 P0 口的位结构图

当 P0 口作为普通 I/O 口使用时，控制信号为 0，与门被封锁，VT1 截止，模拟转换开关将接通锁存器的 $\bar{Q}$ 端与 VT2 的栅极。由于 $\bar{Q}$ 与 VT2 具有倒相作用，因此内部总线上的数据和 P0 口上的数据是一致的。这时如果有写锁存器的信号脉冲加在锁存器的 CLK 端，则内部数据总线上的信号就会送到 P0 口上。因为输出极 VT2 的漏极开路，所以必须外接上拉电阻。

当 P0 口作为输入口使用时，应区分读引脚和读锁存器（端口）两种情况。

(1) 读引脚时，P0 口作为输入，信号既加到了 VT2 上，也加到了读引脚的三态缓冲器上。如果上面的锁存器为 0，则 VT2 导通，P0 口上的电位就被钳在 0 电平上，输入的数据 1 就无法读入。所以，作为通用 I/O 口使用时，在输入数据之前，P0 口是一个准双向口，应先向锁存器中写 1，使 VT2 截止。MOV 类传送指令进行的读口操作就属于这种情况。

(2) 读锁存器时，端口已处于输出的状态下，通过上方的缓冲器读锁存器 Q 端的状态。其目的是适应端口进行“读—修改—写”操作指令的需要。例如，执行 XRL P0, A 指令前，应先读入 P0 口锁存器中的数据，然后与 A 的内容进行逻辑异或，再把结果送到 P0 口输出。这一类的指令有 ANL、ORL、XRL、CPL、JBC、INC 和 DEC 等。

2. P1 口

P1 口是一个准双向口，通常作为通用 I/O 口使用，在电路结构上要比 P0 口简单。当它作为输出口使用时，能向外提供推拉电流，无须外接上拉电阻；当它作为输入口使用时，同样也需要向锁存器写入 1，使输出驱动电路的 FET 截止。P1 口的位结构如图 1-11 所示。

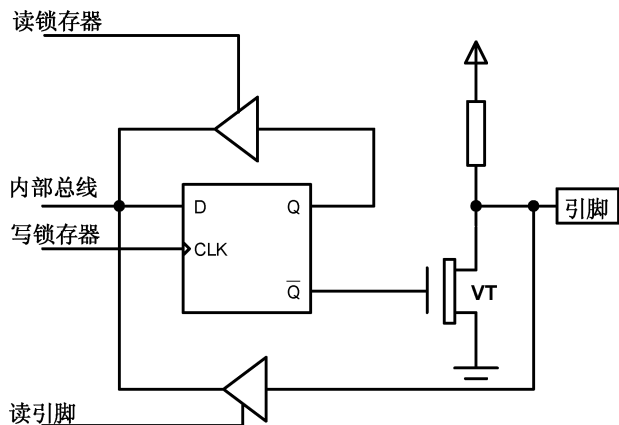


图 1-11 P1 口的位结构图

3. P2 口

P2 口也是一个准双向口，可以作为通用 I/O 口使用。由于 P2 口有时要作为地址线使用，因此，它比 P1 口多了一个多路开关 MUX。当它作为高位地址线使用时，MUX 接通地址信号；当它作为通用 I/O 口使用时，MUX 接通锁存器，使内部总线与其接通。当它作为输出口使用时，无须外接上拉电阻；作为输入口使用时，应区分读引脚和读锁存器。读引脚时，应先向锁存器写 1。P2 口的位结构如图 1-12 所示。

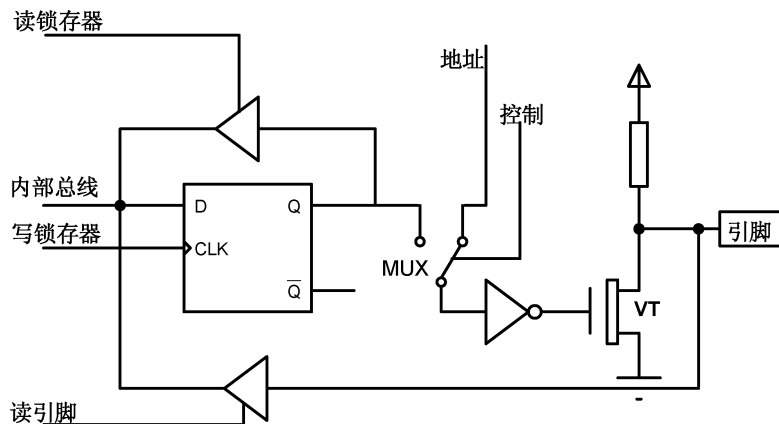


图 1-12 P2 口的位结构图

4. P3 口

P3 口是一个双功能口，也是一个准双向口，既可以作为通用 I/O 口使用，又具有第二功能。P3 口的位结构如图 1-13 所示。

(1) 当 P3 口作为通用输出口使用时，第二功能输出应保持高电平，与非门开通，数据可顺利地从锁存器到输出端引脚上。当作为第二功能信号输出时，该位的锁存器 Q 端应置 1，使与门对第二功能信号的输出打开，从而实现第二功能信号的输出。

(2) 当 P3 口作为通用输入口，或第二功能输入的信号引脚使用时，输出电路的锁存器 Q 端和第二功能输出信号线都应置 1。

(3) 当 P3 口的某些端口线作为第二功能使用时，就不能再作为通用 I/O 口使用了。其他未使用的端口线仍可作为通用 I/O 口使用。同样，若 P3 口作为通用 I/O 口使用，就不能再作

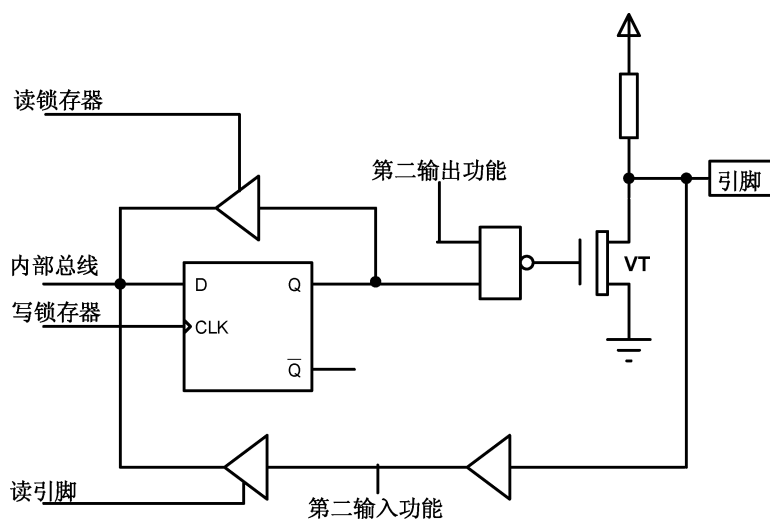


图 1-13 P3 口的位结构图

为第二功能使用了。

(4) P3 口作为通用 I/O 口使用时,读引脚应先向锁存器中写 1,读锁存器时可以直接读取。P3 口的第二功能说明见表 1-9。

表 1-9 P3 口的第二功能说明

I/O 口名称	第二功能符号	功能说明	I/O 口名称	第二功能符号	功能说明
P3.0	RXD	串行口输入	P3.4	T0	计数器 0 计数脉冲输入
P3.1	TXD	串行口输出	P3.5	T1	计数器 1 计数脉冲输入
P3.2	$\overline{\text{INT0}}$	外部中断 0 中断请求信号输入	P3.6	$\overline{\text{WR}}$	外部数据存储器写控制信号
P3.3	$\overline{\text{INT1}}$	外部中断 1 中断请求信号输入	P3.7	$\overline{\text{RD}}$	外部数据存储器读控制信号

(5) 外部器件通过 I/O 口与单片机连接,有总线连接和普通 I/O 口连接两种形式。关于总线连接,详见项目八。

三、硬件电路设计

将以上介绍的引脚接线连接好,单片机就具备了硬件工作的基本条件。简单的单片机应用系统电路如图 1-14 所示,提供形成单片机最简系统的电路和外部的 LED 接口电路,包括晶振电路和复位电路,单片机的引脚接高电平,运行内部 ROM 中的程序,P1 端口接 8 个发光二极管。LED 和普通二极管一样,具有单向导电特性,其正向导通压降一般为 1.7~1.9 V,其点亮电流为 5~10 mA,使用时,通常和 LED 串联一个电阻,其作用在于限制电流,从而达到减少功耗或者满足端口对最大电流的限制。

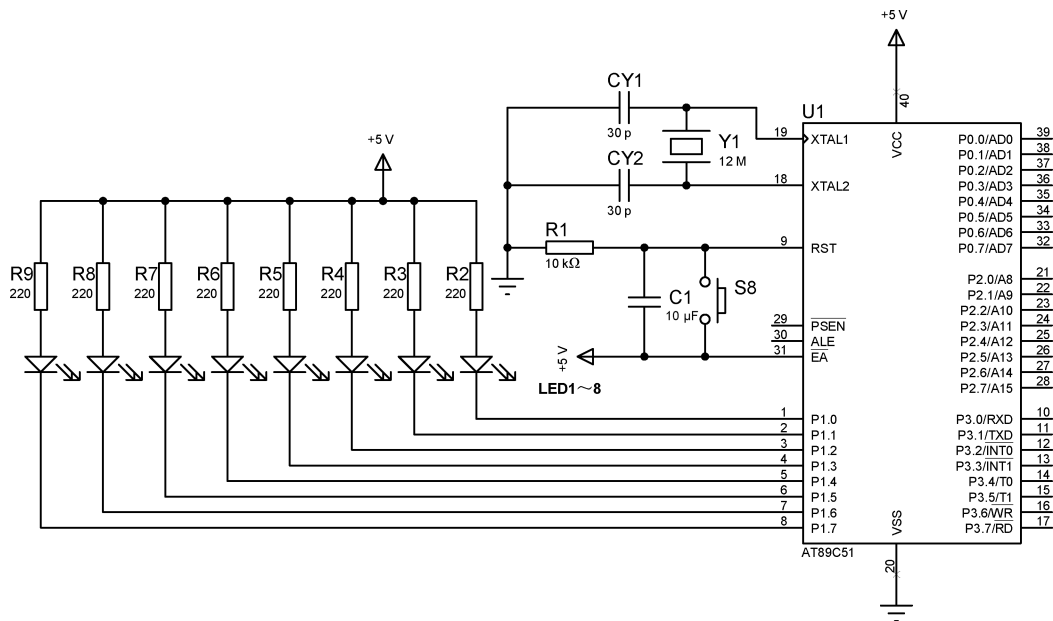


图 1-14 简单的单片机应用系统电路

四、硬件电路实现过程

搭建系统的硬件电路有 3 种方法：第一种方法，独立设计 PCB 板（对于简单任务，也可以采用万用板），焊接元器件；第二种方法，采用各种实验、实训装置；第三种方法，采用 Proteus 软件仿真，此方法只需一台电脑，非常适合自学和硬件设施不足的场合。关于 Proteus 软件的应用，详见项目二。

五、拓展训练

1. 画出典型的晶振电路。
2. 晶振周期、机器周期和指令周期的含义是什么？如晶振频率为 6 MHz，时钟周期、机器周期为多少？
3. RST 引脚的作用是什么？有哪两种复位方式？画出典型的复位电路。
4. 复位后，程序计数器 PC 的内容是什么？这意味着什么？
5. 89C51 的 4 个 I/O 口在使用上有哪些分工和特点？P0~P3 口作为 I/O 口使用时，有何要求？
6. P3 口的第二功能是什么？
7. 在电路板上焊接简单的单片机应用系统。

项目二

单片机开发工具应用



虽然单片机的功能强大，但是自身不具备开发能力，学习单片机技术，除了教程，还需一些必要的开发工具。在本项目中，读者将学习 Keil 和 Proteus 软件的使用，只有学会使用这些开发工具，边学习边实践，不断积累经验，才能真正掌握单片机技术。





任务一 目标代码程序生成

一、任务要求

项目一构建了一个简单的单片机系统，即完成了硬件设计，除此之外，还需进行程序的设计和开发。本任务将使读者学会单片机程序的开发工具 Keil 软件的一般应用，包括如何输入源程序、建立工程、对工程进行设置，以及如何编辑流水灯 C 源程序代码，创建 HEX 文件。

二、单片机程序设计语言与 Keil 开发工具



程序设计语言

(一) 程序设计语言

前已述及，单片机的一个显著特点是需要编写程序才能工作。指令是规定计算机进行某种操作的命令，一条指令仅能完成一种操作，为了完成某项任务，就需要连续执行多条指令，这些指令的集合就是程序。所谓程序，是人们按照自己的思维逻辑，使计算机按照一定的规律进行各种操作，以实现某种功能的有关指令的集合。

作为一个典型的数字电路系统，单片机中所能传递、处理和存储的必然是二进制信息，单片机的 CPU 能直接识别和执行的指令，也不例外，必须是二进制编码。这种用二进制编码表示的指令，称为机器语言，一个简单的机器语言程序见表 2-1。为了便于书写和记忆，机器语言也可用十六进制数表示，尽管如此，机器语言还是具有不易查错、不易修改等缺点，所以人们便采用有一定含义的符号，即助记符来表示指令，这就是汇编语言。汇编语言指令与机器语言指令一一对应，与计算机的内部结构密切相关。汇编语言对单片机的硬件资源操作直接方便、概念清晰，对于掌握单片机的硬件结构极为有利，且具有占用存储空间少、执行速度快等优点。

表 2-1 三种语言程序对照

机器语言		汇编语言	C51
二进制	十六进制		
01111111 11111010	7F FA	MOV R7, #250	unsigned char i, j; for (i=250; i>0; i--) for (j=250; j>0; j--);
01111110 11111010	7E FA	D1: MOV R6, #250	
11011110 11111110	DE FE	D2: DJNZ R6, D2	
11011111 11111010	DF FA	DJNZ R7, D1	



高级语言中，每条语句与多条机器语言指令对应，是一种面向过程且独立于计算机硬件结构的语言。使用高级语言可以大大缩短开发周期，明显增强程序的可读性，便于改进和扩充，支持单片机的高级语言有 BASIC 语言和 C 语言等。本书主要对 C 语言进行讨论。

无论汇编语言程序还是高级语言程序，是不能被计算机直接识别和执行的，必须使用一些工具，将它们编译成机器语言后才能被单片机所执行。常用的编译工具很多，这里介绍 Keil 软件。

（二）Keil μ Vision2 界面介绍

Keil 软件是众多单片机应用开发软件中优秀的软件之一，它支持众多不同公司的 MCS-51 架构的芯片，它集编辑、编译、仿真等于一体，同时还支持汇编和 C 语言的程序设计，在调试程序、软件仿真方面也有很强大的功能。

首先启动 Keil 软件的集成开发环境，可以从桌面上直接双击 Keil μ Vision2 的图标以启动该软件，也可以从“开始—程序”列表中启动，启动 Keil μ Vision2 后界面如图 2-1 所示。 μ Vision2 的界面包括标题栏、主菜单、快捷工具栏、编辑窗口、管理窗口和信息窗口等。

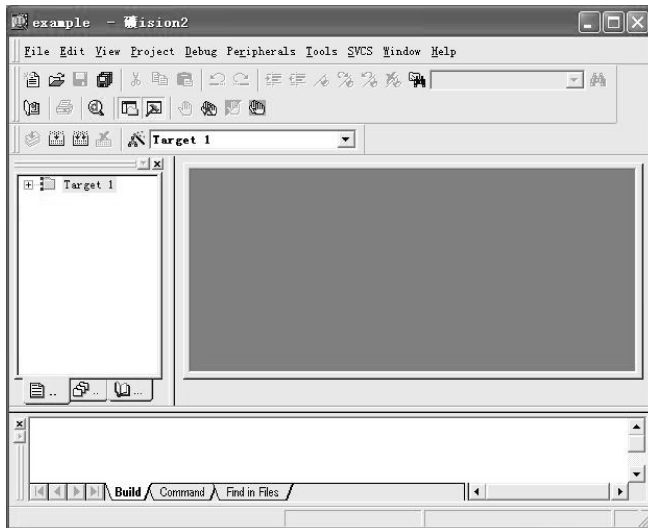


图 2-1 μ Vision2 界面

三、创建一个 μ Vision2 应用项目

（一）启动 μ Vision2 并创建一个项目

启动 μ Vision2 后，点击 Project 菜单，选择弹出的下拉式菜单中的 New Project，将打开一个标准的 Windows 对话框，如图 2-2 所示。此对话框选择项目存放位置并输入项目文件名，如 test，默认的扩展名为 “.uv2”。

输入项目名称保存后，自动弹出选择 CPU 的对话框。点击 “Atmel” 前面的 “+” 号，选择该公司的 AT89C51 单片机，此时屏幕如图 2-3 所示。AT89C51 的功能、特点在图中右边有简单的介绍。

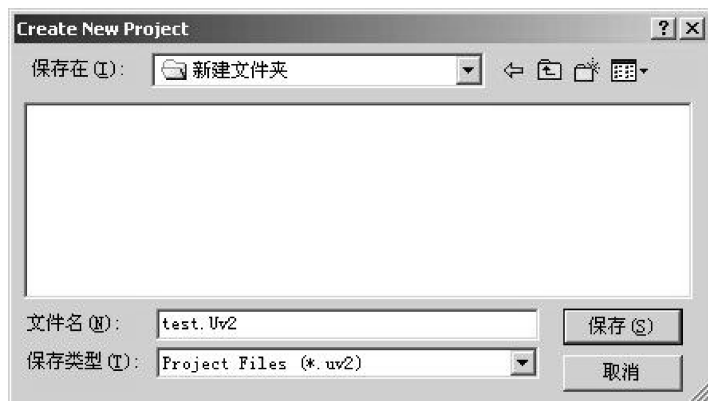


图 2-2 新建项目窗口

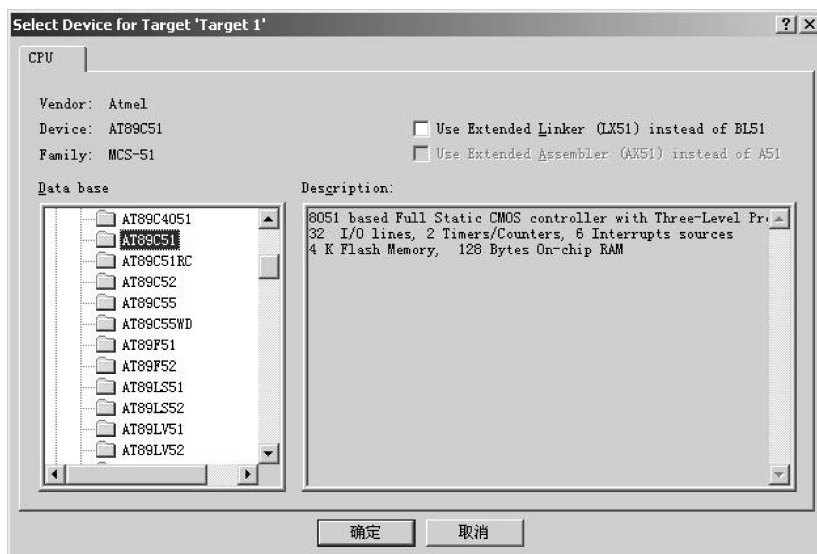


图 2-3 选择单片机型号窗口

(二) 新建并保存一个源程序文件

点击图 2-4 中标号为 1 的新建文件的快捷按钮，也可以通过菜单选项 File→New 实现，将打开一个空的编辑窗口，如图 2-4 中标号为 2 的地方，在该窗口输入源程序（chengxu2_1_1.c）如下：

```
#include "reg51.h"
void delay ()
{
    unsigned int i, j;
    for (i=200; i>0; i--)
        for (j=500; j>0; j--);
}
main ()
{
```



```
while (1)
{
    P1=0x00;      //点亮 LED
    delay ();      //调用延时函数延时
    P1=0xff;      //熄灭 LED
    delay ();      //调用延时函数延时
}
}
```

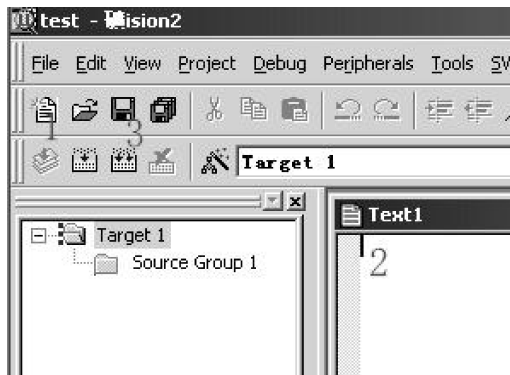


图 2-4 新建源程序文件窗口

即使不明白上面程序的功能，在录入程序的过程中也要注意以下问题：C51 编译器是忽略空格的，因此语句“P1=0x00;”和语句“P1 = 0x00;”是等价的，甚至将该语句写成多行也可以，但不能将一个单词分为两部分。程序的录入具有很大的灵活性，但应确保源代码易于阅读，每条语句应占一行。C51 是大小写敏感的，“P1”和“p1”将是不同的内容。“//”及其后面的内容为注释，对语句功能没有任何影响，可以不录入。除注释以外的其他地方所使用的标点符号必须在英文状态下录入。不要将数字“0”和字母“o、O”、数字“1”和字母“l”混淆。

点击图 2-4 中的标号为 3 的快捷按钮保存新建的程序，也可以用菜单 File→Save 进行保存。在弹出的保存文件窗口中，输入文件名 chengxu2_1_1.c。注意，由于录入的源程序文件为 C 语言的，必须加上扩展名“.c”，如果是汇编语言程序，必须使用扩展名“.asm”。不论是项目名还是源程序文件名都可以任意选取，两者可以相同，也可以不同，但最好具有一定含义，并符合 Windows 文件命名规则，还有就是最好将它们保存在同一目录下。

保存后，程序中的单词将会呈现不同的颜色，说明 Keil 的语法检查生效了。任何语言都有自己的语法要求及习惯，如汉语中某人说“饭吃我”，就不符合语法习惯。同样，C51 也有自己的语法要求。初学者往往不注意细节，不遵从 C51 的语法要求，从而不能成功编译。

（三）将源文件加入到项目中

如图 2-5 所示，鼠标右键点击 Source Group1 文件夹图标，在出现的快捷菜单中选择 Add File to Group 'Source Group 1' 添加源文件。弹出文件窗口，选择刚刚保存的文件，按 ADD 按钮，程序文件加到项目后关闭该窗口。这时在 Source Group1 文件夹图标左边出现了一个小十号，说明文件组中有了文件，点击它可以展开查看。

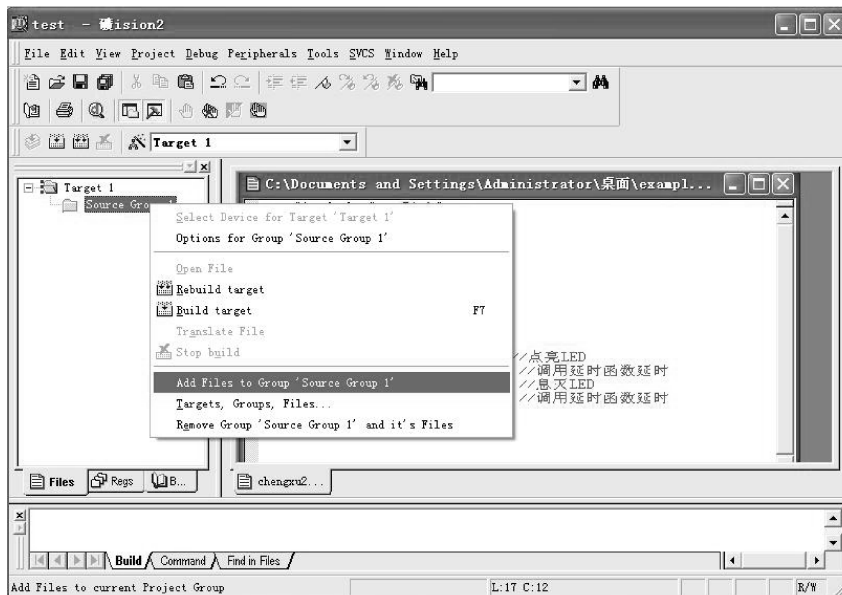


图 2-5 添加源文件到项目

(四) 工程的设置

工程建立好以后，要对工程进行进一步的设置，以满足要求。点击快捷按钮即出现对工程设置的对话框 Option for target 'target1'。该对话框共有 8 个页面，可以定义目标硬件及所选器件的片上元件相关的所有参数，绝大部分设置项都可以选取默认值。

设置对话框中的 Output 页面，如图 2-6 所示。图中标号为 1 的按钮是选择编译输出的路径，标号为 2 的框是设置编译输出生成的文件名，标号为 3 的地方则是决定是否要创建 HEX 文件（可以用编程器写入单片机芯片的 HEX 格式文件，文件的扩展名为 .HEX），选中它编译工程时就可以自动生成 HEX 文件到指定的路径中。

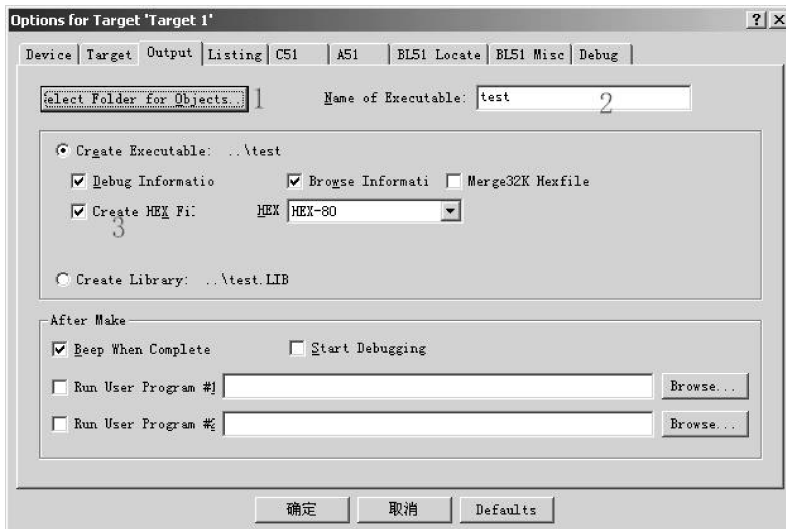


图 2-6 工程设置窗口

设置完成后按确认返回主界面，工程文件建立、设置完毕。



(五) 编译

在图 2-7 中, 标号为 1、2、3 的按钮都是编译按钮, 不同的是: 标号为 1 的按钮用于编译单个文件; 标号为 2 的按钮用于编译当前项目, 如果先前编译过一次之后文件没有做编辑改动, 这时再点击是不会再次重新编译的; 标号为 3 的按钮用于重新编译, 每点击一次均会再次编译链接一次, 不论程序是否有改动。在标号为 3 的按钮右边的是停止编译按钮, 只有点击了前三个中的任意一个, 停止按钮才会生效。也可以利用菜单 Project 进行编译, 见图中标号为 5 的地方。

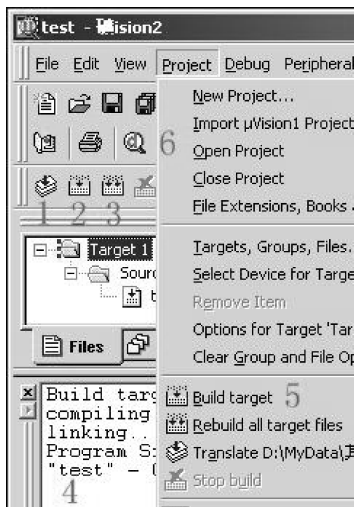


图 2-7 编译窗口

这个项目只有一个文件, 按标号为 1、2、3 的按钮中的哪一个都可以编译。编译后, 在图 2-7 中标号为 4 的地方可以看到编译的信息和使用的系统资源情况等。如果严格按 C51 语法要求录入源程序 chengxu2_1_1.c, 则会成功编译。

首次录入程序, 可能会有错误, 或故意将程序 chengxu2_1_1.c 中的 while 输入为 While, 则编译时将会捕获错误, 并显示提示信息, 如:

```
. \CHENGU2_1_1.C (10): WARNING C206: 'While': missing function—prototype  
. \CHENGU2_1_1.C (10): ERROR C267: 'While': requires ANSI—style prototype  
. \CHENGU2_1_1.C (11): ERROR C141: syntax error near '{'
```

Target not creat

其实只有一个错误语句, 编译器却提供了多条信息。第二条消息指示程序的第 10 行存在错误, 此时用鼠标左键双击该消息行, 一般情况下便可以将光标定位到源程序中出错的位置。有时候需要多次修改、编译。编译连接后产生目标代码 test.hex, 该文件即可被编程器读入并写到芯片中。

四、拓展训练

1. 建立一个 UVision2 设计项目 lianxi.uv2。
2. 将下面的程序 (chengxu2_1_2.c) 添加到项目 lianxi.uv2 中。



```
#include "reg51.h"
void delay (unsigned int t)
{
    unsigned int i, j;
    for (i=t; i>0; i--)
        for (j=110; j>0; j--);
}
main ()
{
    unsigned char w, i;
    while (1)
    {
        w=0xfe;
        for (i=0; i<8; i++)
        {
            P1=w;           //循环点亮 LED
            w<<=1;          //点亮灯的位置移动
            w=w | 0x01;
            delay (500);     //延时
        }
    }
}
```

3. 编译项目 lanxi. uv2 生成 lianxi. hex 文件。

任务二 程序下载

一、任务要求

任务一利用 Keil 软件生成了可执行目标文件 test. hex，本任务将把它下载到单片机中，看一下程序的运行效果。通过本任务，读者将会焊接 STC 系列单片机程序下载电路，学会 STC 单片机 ISP 软件的使用及下载方法。

二、程序下载方法概述

程序下载，即将编译生成的目标代码传送到应用系统的程序存储器中（ROM）。只有单