



普通高等教育“十二五”创新型规划教材

单片机技术及C51 仿真与应用

DANPIANJI JISHU JI C51
FANGZHEN YU YINGYONG

主 编 王冬星 许有军
副主编 董雷刚 李 娟

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

普通高等教育“十二五”创新型规划教材

单片机技术及 C51 仿真与应用

主 编	王冬星	许有军
副主编	董雷刚	李 娟
主 审	马英瑞	王 健

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书通俗易懂,面向应用,从实例仿真应用入手,以C语言为主、汇编语言为辅结合仿真开发环境,详细介绍了51单片机的结构原理、内外部资源编程、设计开发过程,以及单片机常用开发仿真软件 μ Vision Keil和Proteus ISIS的应用及调试方法。

书中列举大量实例,均通过仿真验证,且每个程序都有详细的注释,通过对电路或程序做改动,即可重新设计系统。本书各章都配有学习目标、习题和答案,并附计算机常用编码、汇编指令表和Proteus常用器件中英文对照表,供相关人员参阅。

本书可以作为高等院校的单片机教材,也可作为从事单片机开发应用的技术人员提供参考。

版权专有 侵权必究

图书在版编目(CIP)数据

单片机技术及C51仿真与应用/王冬星,许有军主编. —北京:北京理工大学出版社,2015.2

ISBN 978-7-5640-9973-2

I. ①单… II. ①王… ②许… III. ①单片微型计算机-C语言-程序设计 IV. ①TP368.1 ②TP312

中国版本图书馆CIP数据核字(2015)第036324号

出版发行/北京理工大学出版社有限责任公司

社 址/北京市海淀区中关村南大街5号

邮 编/100081

电 话/(010) 68914775(总编室)

82562903(教材售后服务热线)

68948351(其他图书服务热线)

网 址/<http://www.bitpress.com.cn>

经 销/全国各地新华书店

印 刷/保定市中华美凯印刷有限公司

开 本/787毫米×1092毫米 1/16

印 张/15.5

字 数/357千字

版 次/2015年2月第1版 2015年2月第1次印刷

定 价/28.00元

责任编辑/陈莉华

文案编辑/陈莉华

责任校对/周瑞红

责任印制/王美丽

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

前 言

单片机技术是计算机、电子、自动化等专业学生应该掌握的基本技能之一，也是这些专业的主干课之一。随着物联网时代的到来，以 51 单片机为“敲门砖”进而学习其他较难专业课的现象已经非常普遍。

一、本书特色

为兼顾 51 单片机知识体系结构的完整性和技术学习需要快速上手的特点，本书从第 1 章起，在讲授理论的同时就配有相应的仿真实验环境和案例学习，让读者在学中练，在练中学。因此本书具有以下特色：

(1) 由浅入深，提供大量源代码、完全支持仿真练习。

本书在开始讲授理论知识时，就提供仿真案例和源程序讲解，通过对案例进行微小更改就能生成新的系统，源文件可自行下载练习，使读者能直观学习，提高学习效率。

(2) C 语言与汇编语言互相对照，方便同时学习两种语言。

本书既讲授了传统的单片机教学语言——汇编语言，又讲授了主流的单片机开发语言——C 语言。在前 7 章中，均使用两种语言对案例进行了介绍，方便读者同时掌握两种语言。

(3) 图文并茂，面向应用，适合阅读；学练结合，体系结构完整，适合讲授。

书中有大量的实验步骤、实验效果和电子元器件实物图，使读者阅读明了，易于学习。知识体系结构完整，且每个知识点都配有面向应用的仿真练习，摒弃了应用类书籍的知识体系结构缺乏和传统理论书籍应用练习少的缺点。本书知识点涵盖适中，适合课堂讲授。

二、本书使用建议

使用本书，建议用 μ Vision Keil 进行程序开发，没有实验环境时可先采用 Proteus ISIS 进行仿真，等具备单片机硬件实验环境时，可使用实际电子元器件进行系统开发。

本书建议学时为 56 + 12 学时，其中 56 学时为教学学时，包括课程教学中必要的实验练习，12 学时是扩展知识学习，结合每一章节尤其是第 10 章的外部资源学习，可独立设计一个单片机控制系统。通过本书的学习可以掌握单片机系统设计的基本知识，为以后学习打下基础。

本书由王冬星、许有军担任主编，董雷刚、李娟担任副主编，马英瑞、王健主审。第 1 章、第 3 章、第 6 章由王冬星编写，第 5 章、第 9 章由许有军编写，第 2 章、第 8 章由董雷刚编写，第 4 章、第 7 章由李娟编写，第 10 章由许有军、董雷刚编写，附录 B、E 由王冬星编写，附录 A、C、D 由李娟编写，全书由王冬星统稿、定稿。作者向为本书提供许多宝贵建议和意见的程书伟老师表示衷心的感谢。

由于作者水平有限，书中难免存在不妥之处，敬请读者批评指正。邮箱 cnwindy01@163.com。

编 者

目 录

第1章 初识51单片机及其开发环境	1
1.1 微型机与单片机	1
1.2 51单片机的基本结构概述	2
1.3 51单片机的引脚与封装	2
1.4 一个简单LED灯循环点亮实验	4
1.5 μ Vision4开发环境介绍	7
1.5.1 μ Vision4开发界面	7
1.5.2 演示步骤	9
1.6 Proteus仿真环境简介	9
1.6.1 Proteus的安装	10
1.6.2 Proteus的使用	14
1.6.3 μ Vision4 + Proteus运行单片机仿真	16
1.7 51单片机基本电路	16
1.8 51系列单片机	19
习题	20
第2章 51单片机内部硬件基本结构和原理	21
2.1 CPU单元	21
2.2 51单片机存储器的配置	23
2.2.1 程序存储器ROM空间	23
2.2.2 数据存储器RAM空间	25
2.3 时序与复位	31
2.3.1 时序分析	31
2.3.2 时序单位	32
2.3.3 复位	33
2.4 节电方式	34
2.5 输入/输出端口	35
2.5.1 P0口	35
2.5.2 P1口	36
2.5.3 P2口	36
2.5.4 P3口	37
习题	38
第3章 51单片机汇编指令系统与程序设计	39
3.1 汇编语言概述	39

3.2 指令格式和描述符号说明	40
3.2.1 指令格式	40
3.2.2 指令中常用描述符号说明	40
3.3 寻址方式	41
3.4 指令系统	43
3.4.1 数据传送类指令 (29 条)	44
3.4.2 算术运算类指令 (24 条)	49
3.4.3 逻辑运算与循环类指令 (24 条)	53
3.4.4 控制转移类指令 (17 条)	56
3.4.5 位操作类指令 (17 条)	61
3.5 伪指令	64
3.6 汇编与汇编程序设计	66
3.6.1 源程序的汇编	66
3.6.2 汇编程序设计	67
3.6.3 综合设计实验——可控灯汇编程序设计	68
习题	70
第4章 51 单片机的 C 语言程序设计	72
4.1 C51 语言与汇编语言和标准 C 语言的区别	72
4.2 C51 程序的基本结构	73
4.3 数据类型	74
4.3.1 C51 数据类型	74
4.3.2 REG51.H 头文件	76
4.4 变量和 C51 存储区域	79
4.4.1 变量的定义	79
4.4.2 存储器类型	79
4.4.3 存储器模式	80
4.4.4 变量的分类	81
4.5 C51 对绝对地址的访问	82
4.6 指针	82
4.6.1 指针的概念、定义和引用	83
4.6.2 C51 的指针类型	85
4.7 C51 函数	87
4.7.1 C51 函数及其定义	88
4.7.2 C51 的中断服务函数	90
4.7.3 C51 库函数	92
习题	94
第5章 数码管显示原理及应用	95
5.1 数码管的结构及工作原理	95

5.2 驱动数码管	96
5.2.1 驱动共阴极数码管	96
5.2.2 驱动共阳极数码管	97
5.3 数码管的显示及实例	98
5.3.1 数码管的静态显示及实例	98
5.3.2 数码管的动态显示及实例	100
5.3.3 设计型实验——模拟秒表的实现	103
习题	105
第6章 51 单片机的中断系统	106
6.1 中断概念	106
6.2 51 单片机中断源	106
6.3 中断寄存器	107
6.3.1 中断允许寄存器 IE	108
6.3.2 定时/计数器控制寄存器 TCON	109
6.3.3 串行口控制寄存器 SCON	110
6.3.4 中断优先级寄存器 IP	111
6.4 中断处理过程	112
6.4.1 中断响应条件	112
6.4.2 中断响应时间	113
6.4.3 中断响应过程	114
6.4.4 中断返回	114
6.5 中断程序举例	114
6.5.1 主程序说明	115
6.5.2 中断服务程序说明	115
6.5.3 外部中断INT0程序举例	115
6.5.4 外部中断INT1程序举例	119
6.5.5 两个外部中断举例	121
习题	123
第7章 定时/计数器的原理及应用	124
7.1 定时/计数器的结构和工作原理	124
7.1.1 定时/计数器的结构	124
7.1.2 定时/计数器的工作原理	125
7.2 定时/计数器的工作方式寄存器和控制寄存器	126
7.2.1 工作方式寄存器 TMOD	126
7.2.2 控制寄存器 TCON	127
7.3 定时/计数器的4种工作方式及应用	127
7.3.1 方式0的工作原理及应用	127
7.3.2 方式1的工作原理及应用	128

7.3.3	方式2的工作原理及应用	133
7.3.4	方式3的工作原理及应用	135
7.3.5	定时/计数器的外部中断扩展及示例	138
7.4	综合实验——设计一个带蜂鸣器的60秒计时器	140
7.4.1	蜂鸣器的工作原理	140
7.4.2	60秒计时器设计实验	140
	习题	145
第8章	键盘的工作原理及应用	146
8.1	键盘的工作原理及去抖	146
8.2	独立键盘接口及程序实例	147
8.3	矩阵键盘扫描原理及实例分析	150
8.3.1	低电平扫描原理	150
8.3.2	高电平扫描原理	151
8.3.3	基础实验——矩阵键盘的识别与显示	152
8.4	综合实验——秒表定时器的设计与实现	156
	习题	160
第9章	单片机的串行通信	161
9.1	串行通信的概念	161
9.1.1	串行通信的基本方式	162
9.1.2	串行通信的波特率	164
9.1.3	串行通信的标准接口	165
9.1.4	RS-232电平与TTL电平的转换	165
9.2	串行通信的结构及工作方式	167
9.2.1	串行通信的结构	167
9.2.2	串行通信的工作方式	170
9.3	串行口的应用	171
9.3.1	串行口初始化编程	171
9.3.2	单片机与PC通信	173
9.3.3	双机通信	176
9.3.4	串口打印在程序调试中的使用	181
	习题	182
第10章	51单片机常用资源与综合应用实例	183
10.1	LED点阵	183
10.1.1	LED点阵概述	183
10.1.2	LED点阵的结构和工作原理	183
10.1.3	LED点阵的编码方式及显示实例	184
10.1.4	LED点阵的动态显示及实例	186
10.2	LCD	189

10.2.1 LCD 概述	189
10.2.2 1602 液晶显示器	190
10.2.3 1602 液晶显示实例	193
10.3 温度传感器	198
10.3.1 温度传感器概述	198
10.3.2 DS18B20 温度传感器	199
10.3.3 LCD1602 与 DS18B20 综合应用实例	202
10.4 C#与单片机通信实现温度显示系统	207
10.4.1 C#简介	207
10.4.2 系统总体设计	207
10.4.3 系统硬件设计	208
10.4.4 系统软件设计	209
10.4.5 系统运行结果	217
习题	218
附录 A 计算机常用编码	219
附录 B 51 单片机汇编指令表	221
附录 C 51 单片机汇编指令矩阵 (汇编/反汇编表)	226
附录 D Proteus 常用器件和仿真仪表中英文对照	228
附录 E 习题解答	230
参考文献	235

第 1 章

初识 51 单片机及其开发环境

学习目标:

- 掌握 51 单片机的基本结构、40 个引脚的封装顺序。
- 掌握 51 单片机的最小系统的基本电路。
- 掌握 51 单片机的软硬件开发流程和 Proteus 仿真方法。

重点内容:

- 51 单片机的基本结构。
- Proteus 和 keil C51 开发环境的使用。

1.1 微型机与单片机

微型计算机系统一般包括中央处理单元（CPU）、存储器（Memory）及输入/输出单元（I/O）三大部分，如图 1.1 所示。CPU 正如人的大脑一样，控制整个计算机系统的运行。存储器则是存放系统运行所需的程序及数据，包括只读存储器（Read Only Memory, ROM）及随机存取存储器（Random Access Memory, RAM）。通常 ROM 用来存储程序或永久性的数据，称为程序存储器；RAM 则是用来暂时存储程序执行时的数据，称为数据存储器。I/O 是微型计算机系统与外部沟通的管道，其中包括输入端口与输出端口。这三部分分别由不同的元器件组成，然后把它们组装在电路板上，形成一个微型计算机系统。

单片微型计算机（即单片机或微控制器）就是把中央处理单元、存储器、输入/输出单元等全部放置在一个芯片里，如图 1.2 所示，只要再配置几个小元件，如电阻器、电容器、

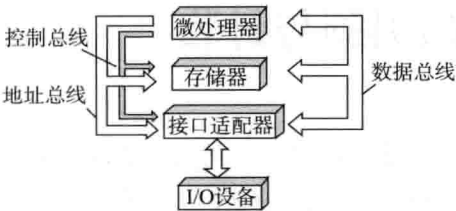


图 1.1 微型计算机组成

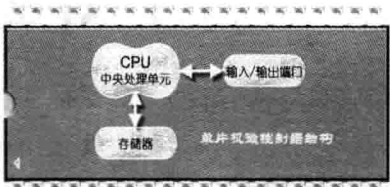


图 1.2 单片机微控制器结构

石英晶体、连接器等，就成为一个完整的微型计算机系统。因此整个系统的体积小、成本低、可靠性高，成为目前微型计算机控制系统的主流。

1.2 51 单片机的基本结构概述

51 单片机发展至今，虽然有许多厂商各自开发了不同的兼容芯片，但其基本结构变化不大，图 1.3 为标准的 51 单片机结构图。

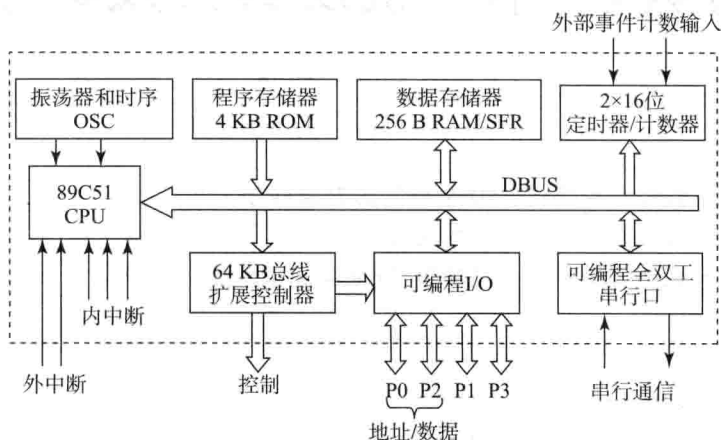


图 1.3 51 单片机基本结构框图

本章先介绍 51 单片机的基本结构、引脚、基本电路及 51 系列等，其中引脚结构需要熟记，本书提供记忆技巧，很快记住 40 个引脚和基本电路。

51 单片机芯片内主要包括：

- 程序存储器 ROM：内部 4 KB、外部最多可扩展至 64 KB。
- 数据存储器 RAM/SFR：内部 256 B、外部最多可扩展至 64 KB。
- 四组可位寻址的 8 位输入/输出端口，即 P0、P1、P2 及 P3。
- 两个 16 位定时器/计数器。
- 一个全双工异步串行口，即 UART。
- 五个中断源，即 $\overline{\text{INT0}}$ 、 $\overline{\text{INT1}}$ 、T0、T1、RXD 或 TXD。
- 111 条指令。

1.3 51 单片机的引脚与封装

1. PDIP40 与 PDIP42

51 单片机在教学过程中最常用的是 40 个引脚双列直插式封装（Plastic Dual Inline Package，简称 PDIP），如图 1.4 所示，还有一种双列直插式封装如图 1.5 所示，即 PDIP42。PDIP40 与 PDIP42 引脚数量不相同，尺寸差异也很大。PDIP40 特别适用于学校培训机构使用，但由于体积较大，制作成本较高，很少在商业里使用。

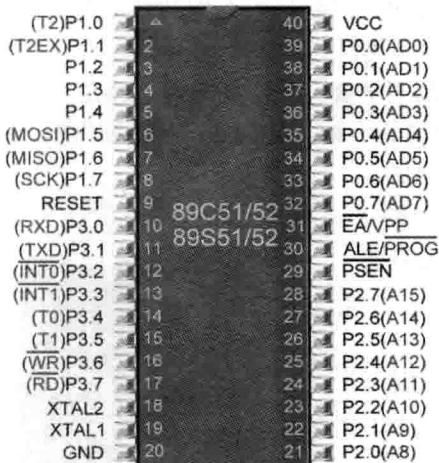


图 1.4 PDIP40 双列直插式封装

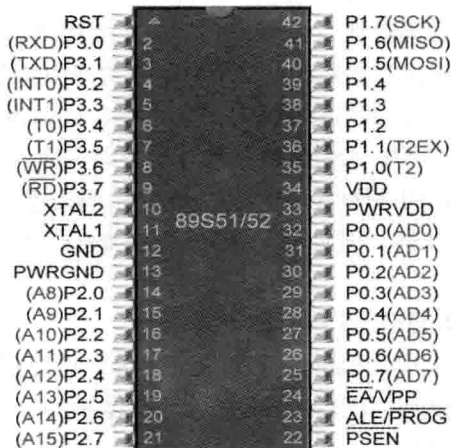


图 1.5 PDIP42 双列直插式封装

本书以 PDIP40 封装的 51 单片机为对象, 进行案例开发。建议大家把 PDIP40 引脚记住。在此提供快速记忆技巧, 让大家轻松记住这 40 个引脚。

51 系列单片机最迷人处之一就是它的输入/输出功能, 它有 4 组输入/输出端口 P0 ~ P3, 每组端口 8 个引脚, 共 32 个输入/输出引脚, 它们占用了 40 个引脚中的绝大部分位置, 必须熟记这 32 个引脚在 PDIP40 中的位置。

PDIP40 的右上角 40 号脚为 V_{CC} 引脚, 连接 $(1 \pm 10\%) \times 5V$ 的电源; PDIP40 左下角 20 脚为 GND 引脚, 必须接地。

PDIP40 第 9 脚为复位引脚, 在复位引脚接高电平超过 2 个机器周期时可产生复位动作。按一下 RESET 键, 系统即可复位。PDIP40 的第 18 脚接 XTAL2, 第 19 脚接 XTAL1。这两个引脚即时钟脉冲引脚, 它是单片机的“心脏”, 单片机的所有动作都是以此频率来工作的。

现在只剩下 EA、ALE 和 PSEN 三个引脚, 现分别介绍如下:

• 存储器引脚。

51 单片机既可以使用内部存储器也可以使用外部存储器, 至于使用内部存储器, 还是外部存储器, 则需视 31 脚 (P0 下面那个脚) 而定。31 脚为存取外部存储器使能 EA (External Access Enable) 引脚。当 $EA = 1$ 时, 系统使用内部存储器; 当 $EA = 0$ 时, 系统使用外部存储器。

• 外部存储器控制引脚。

30 脚为地址锁存使能 ALE (Address Latch Enable), 其功能是在存取外部存储器时, 送出一个将原本在 P0 的地址 (A0 ~ A7) 信号锁存到外部锁存器 IC (如 LS74373), 让 P0 空出来, 以传输数据。简单讲, 当外接存储器电路时, 若 $ALE = 1$, P0 被用作地址总线; 若 $ALE = 0$, P0 被用作数据总线。

29 脚为程序存储使能 PSEN (Program Store Enable), 其功能是读取外部存储器。通常此引脚连接到外部存储器 (ROM) 的 OE 引脚, 当 51 单片机要读取外部存储器的数据时, 此引脚就会输出一个低电平信号。

相对于前面的 38 个引脚, 29、30 脚比较难以说明, 但是只要不用到外部存储器, 就可当它们不存在, 留待后面关于外部存储器的章节再行说明。

根据上述要诀, 很容易记住这些引脚的位置和基本功能。需要补充说明的是, 51 单片机的 40 个引脚里有很多是复用引脚, 简单讲就是多用途的引脚, 以 39 脚到 32 脚为例, 一般情况下为 P0; 若是连接外部存储器时, 则当成 AD0 ~ AD7 引脚, 而 AD0 ~ AD7 就是地址引脚与数据引脚混合的复用引脚, 但如果不接外部存储器时就可以当它不存在。

2. QFP 与 PLCC 封装

51 单片机的 QFP 封装有 PQFP 和 TQFP 两种, 均为扁平的 44 引脚表贴式封装, 其体积小, 成本低, 是目前商业主流, 学校和培训结构几乎不用, 其引脚如图 1.6 所示。

51 单片机中, PLCC 也是一种常用的封装方式, 也为 44 引脚表贴式封装, 其引脚封装顺序与 QFP 类似, 如图 1.7 所示, 但这种封装可直接卡在电路板上, 如图 1.8 所示, 适用于实验室及大批量生产。

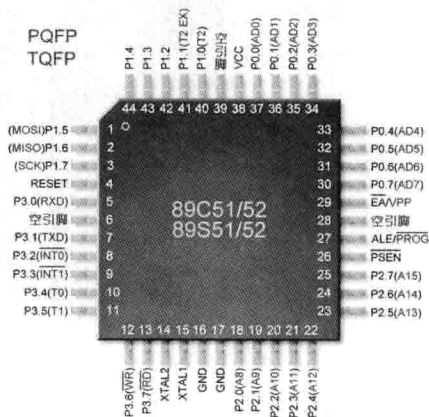


图 1.6 QFP 封装

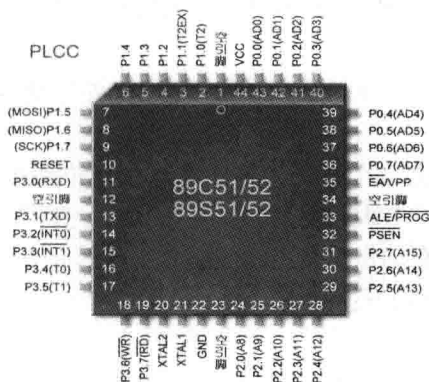


图 1.7 PLCC 封装

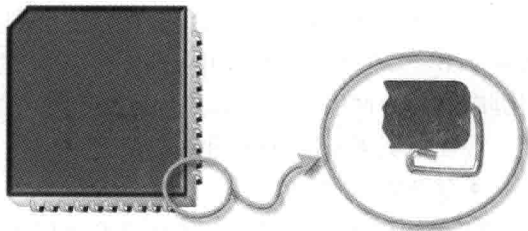


图 1.8 PLCC 芯片管座

1.4 一个简单 LED 灯循环点亮实验

在了解 51 单片机的基本结构和引脚封装后, 在 Proteus 仿真环境下用单片机最小系统实现一个 LED 灯循环点亮实验。

一、实验目的

- (1) 了解 51 单片机的开发过程, 获得单片机系统制作的感性认识。
- (2) 了解 51 单片机的汇编语言和 C 语言编程结构。

二、实验要求

8 个 LED 灯 D1 ~ D8 由单片机控制，先是全亮，然后 D1 ~ D8 依次由单片机的 P2 口控制被点亮。

三、电路图

系统电路如图 1.9 所示。依次点亮过程如图 1.10 所示。

四、程序流程图

程序的流程图如图 1.11 所示。

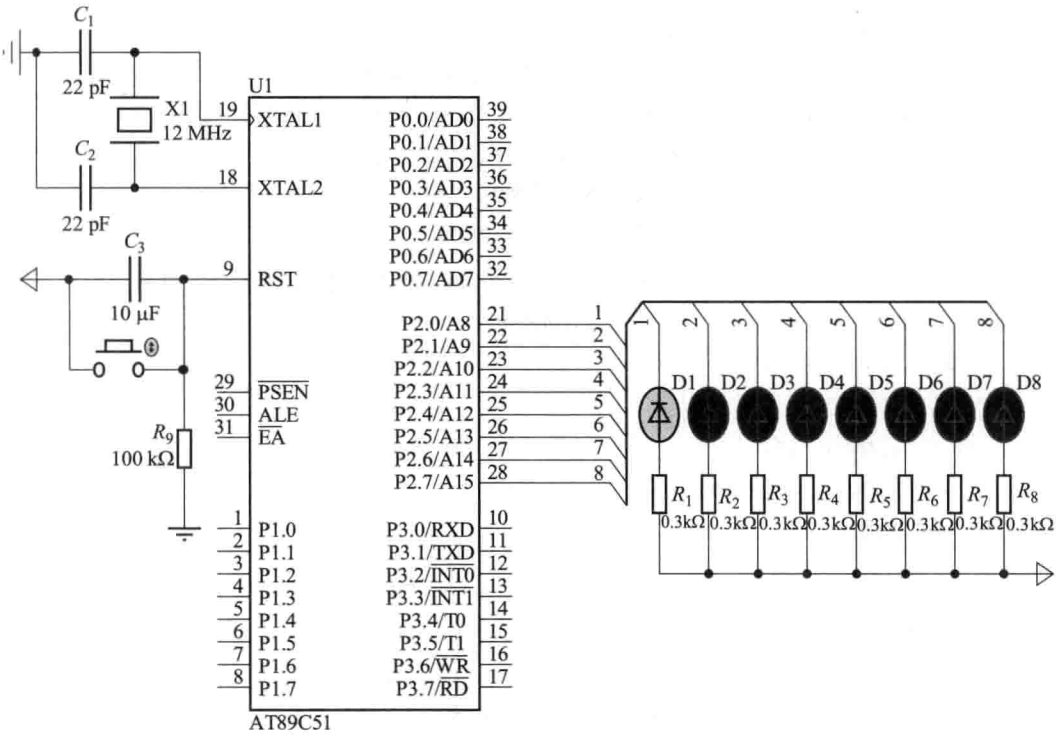


图 1.9 系统电路运行截图

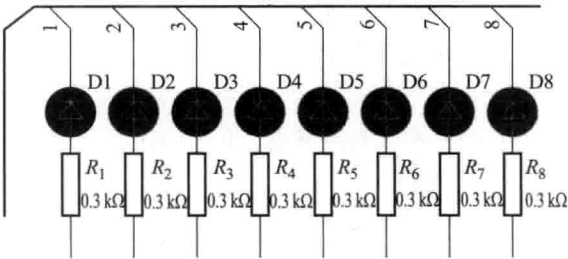


图 1.10 LED 灯循环点亮示意图

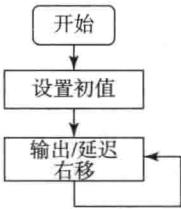


图 1.11 程序流程图

五、程序设计

在 μ Vision4 中新建工程, 加入汇编程序 ch01.asm 或 C51 程序 ch01.c, μ Vision4 开发环境在 1.5 节介绍。在 Proteus 仿真环境下绘制图 1.9, 仿真运行该实验, Proteus 仿真环境在 1.6 节介绍。汇编语言参考程序如下:

```
;LED 灯循环实验汇编程序 ch01.asm
    org 0000H          ; 程序从 0000H 开始运行
    sjmp main          ; 跳到主程序
    org 0030H          ; 主程序从 0030H 开始运行
main: mov p2,#00H      ; 全亮
      lcall delay       ; 调用延迟程序,延迟 1 s
      mov a,#0FEH       ; 寄存器 a 送初值,最低位为 0
loop: mov p2,a          ; 输出到 P2,点亮 P2 口最低位的灯即 D1
      lcall delay       ; 调用延迟程序,延迟 1 s
      rl a              ; 位左移,灯循环右移
      ajmp loop         ; 跳到 loop 处循环
delay: mov r5,#100      ; 延时 100 * 10 ms = 1 s 子程序,闪烁灯调用
d10ms: mov r6,#20       ; 延时 10 ms
      d2: mov r7,#250
          djnz r7,$
          djnz r6,d2
          djnz r5,d10ms
      ret
      end
```

C 语言参考程序如下。

```
/* ch01.c - LED 灯循环实验 C51 程序 */
#include <reg51.h>      // 定义 51 单片机寄存器之头文件
#define LED P2          // 定义 LED 接至 P2
void delay(int);        // 声明延迟函数
main()                  // 主程序开始
{
    unsigned char i;    // 声明无符号变量 i (占 1B)
    LED = 0;            // 全亮
    delay(100);         // 延时 100 * 10 ms = 1 s
    while(1)            // 无穷循环,程序一直运行
    {
```

```

LED=0xFE;           // 初值=1111 1110,只有最左灯 D1 亮
delay(100);          // 延时 100 * 10 ms = 1 s
for(i=0;i<7;i++)     // 位左移,灯右移 7 次
{
    LED=(LED<<1)|0x01;
    // LED 左移 1 位,并设置最低位为 1,相当于点亮的灯右移 1 位
    delay(100);      // 延迟 100 * 10 ms = 1 s
}                    // for 循环结束,右移结束,只有最右灯 D8 亮
}                    // while 循环结束
}                    // 主程序结束
/* 延迟函数,延迟约 x * 10ms */
void delay(int x)     // 延迟函数开始
{                     // 声明整数变量 i,j
    int i,j;
    for (i=0;i<x;i++) // 计数 x 次,延迟 x * 10 ms
        for (j=0;j<1200;j++); // 计数 1200 次,延迟 10 ms
}                     // 延迟函数结束

```

注：本书大部分程序会用汇编语言和 C51 两种语言编写，当程序很大时，才只采用 C51 编写，而且经过实验验证后，全部可以运行。

1.5 μ Vision4 开发环境介绍

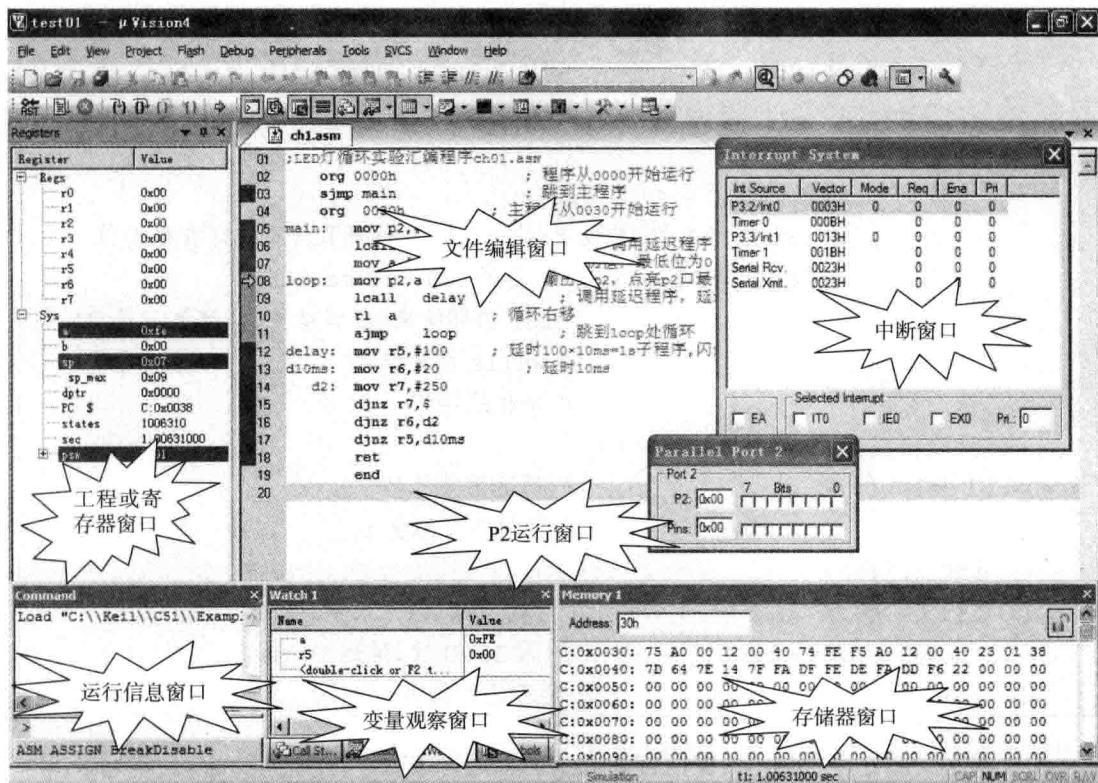
美国 Keil 公司生产的 μ Vision 开发环境是一款集编辑、编译（或汇编）、仿真调试等功能于一体的嵌入式集成开发环境，2013 年 10 月，Keil 正式发布了 keil μ Vision5。由于 keil μ Vision5 还没有普及和破解，且 μ Vision4 很稳定，本书以 μ Vision4 为版本介绍 Keil 的使用。 μ Vision 支持数百种嵌入式处理器的程序开发（包括 51 系列单片机、ARM 芯片等）。它支持汇编程序开发，也支持 C 程序开发。互联网上可以免费下载该软件，如 Keil 公司网站（<http://www.keil.com>）。

1.5.1 μ Vision4 开发界面

μ Vision4 的开发界面如图 1.12 所示，包含一个 51 单片机开发环境所必需的工具，如文件编辑、片内外设所连接窗口（如 P0 ~ P3 运行观察窗口）、运行信息窗口、存储器信息显示窗口、调试等工具。

为便于工程运行，可以展开寄存器（Register）窗口、存储器信息显示窗口和变量观察窗口进行监视，还可以打开其他许多窗口进行调试。

μ Vision4 把一个项目当作一个工程来开发，开发过程分为两个阶段，第一阶段是编辑程

图 1.12 μ Vision4 运行界面

序, 编译工程 (build), 生成目标程序和可执行文件, 如 .HEX 文件; 第二阶段, 对编译生成的可执行文件进行调试和仿真。具体运行过程如下。

1. 新建工程

新建一个文件夹, 如 test01, 用以保存同一工程下的相关文件。在 Project 菜单下选择 New Project 选项, 在弹出窗口中, 输入工程名 test1, 系统自动添加扩展名。

在新建工程中, 新建一个文档, 为 .c 或 .asm, 如 ch01.asm 或 ch01.c, 即 1.4 节所示的两个文件, 直接复制过来即可。在工程窗口中, 在 Source Group1 文件夹上右击鼠标, 单击 “Add Files to Group ‘Source Group1’” 选项, 把 ch01.asm 或 ch01.c 文档添加到 test1 工程中。相应的还可以添加多个文件。

2. 设置工程

单击 Project 菜单下的 “Options for Target ‘Target 1’” 选项, 弹出工程设置窗口, 如图 1.13 所示。或在工程窗口中右键单击 Target 1, 在弹出菜单中选择 “Options for Target ‘Target 1’”。

在工程设置窗口中, 有许多选项可以进行设置, 但经常设置的内容包括:

- (1) Device 标签, 选择芯片型号。
- (2) Target 标签, 设置晶振频率。
- (3) Output 标签, 选中十六进制 Create HEX File 输出文件。
- (4) Debug 标签, 选择调试模拟方式, 主要有软件与硬件两种仿真方式。