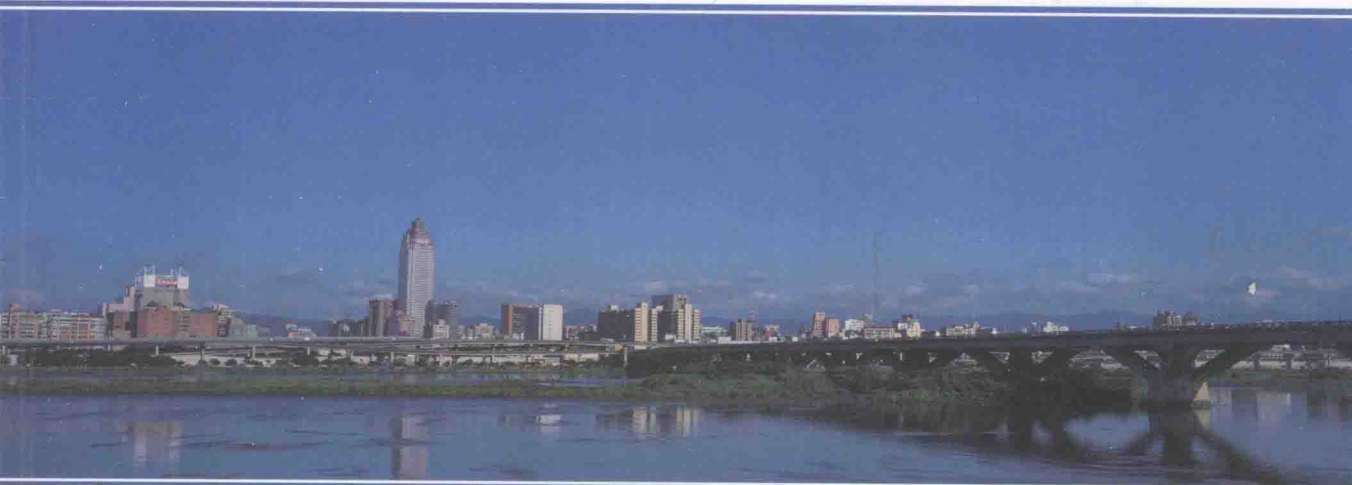


高等院校嵌入式人才培养规划教材

Gaodeng Yuanxiao Qianrushi Rencai Peiyang Guihua Jiaocai

单片机应用技术 实例教程 (C51版)

汤嘉立 主编 李林 胡羽 周安华 副主编



C51 Single Chip Microcomputer

融知识体系性和技术技能性于一体

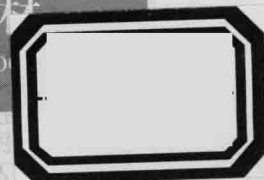
软硬结合, 对于所有的应用实例均给出了完整的电

路原理设计图以及对应的C51语言代码

由浅入深、循序渐进, 符合学生认知规律

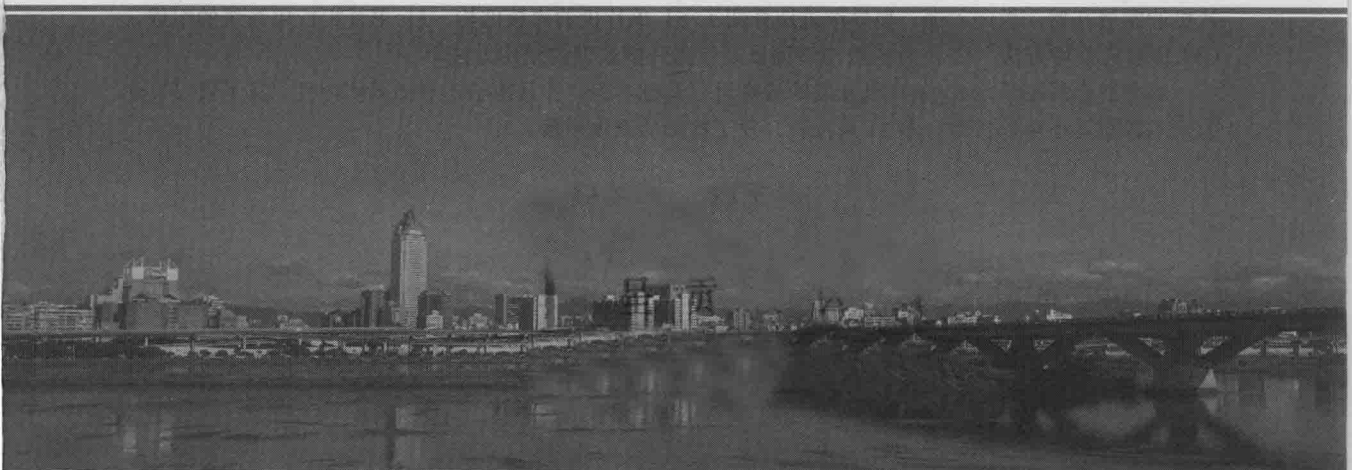
高等院校嵌入式人才培养规划教材

Gaodeng Yuanxiao Qianrushi Rencai Peiyang Guihua Jiao



单片机应用技术 实例教程 (C51版)

汤嘉立 主编 李林 胡羽 周安华 副主编



C51 Single
Chip Microcomputer

人民邮电出版社

北京

图书在版编目(CIP)数据

单片机应用技术实例教程: C51版 / 汤嘉立主编

— 北京: 人民邮电出版社, 2014. 11

高等院校嵌入式人才培养规划教材

ISBN 978-7-115-35619-2

I. ①单… II. ①汤… III. ①单片微型计算机—高等学校—教材 IV. ①TP368.1

中国版本图书馆CIP数据核字(2014)第175027号

内 容 提 要

51 单片机应用系统是嵌入式控制系统的重要分支, 在工业控制等场合得到了广泛的应用, 51 单片机的开发是目前高校等教育机构相应专业学生的必修课程。本书由 11 章以及 1 个附录组成, 从 51 单片机的发展开始, 由浅入深、循序渐进地介绍了 51 单片机的内核结构、51 单片机应用系统的组成、C51 语言的使用方法、51 单片机内部资源以及常用外围器件的使用方法。

本书适合需要学习 51 单片机开发的读者进行基础学习, 并且由于本书的高实用性, 其不仅可以作为一本教材, 还可以作为一本 51 单片机开发工程师的查询手册。

◆ 主 编 汤嘉立

副 主 编 李 林 胡 羽 周安华

责任编辑 王 威

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京艺辉印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 21.5

2014 年 11 月第 1 版

字数: 563 千字

2014 年 11 月北京第 1 次印刷

定价: 46.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前 言

51 单片机具有体积小、功能强、价格低的特点,在工业控制、数据采集、智能仪表、机电一体化、家用电器等领域有着广泛的应用。其应用可以大大提高生产、生活的自动化水平。近年来,随着嵌入式的应用越来越广泛,51 单片机的开发也变得更加灵活和高效,51 单片机的开发和应用已经成为嵌入式专业、电气专业学生的必备技能。为了适应这一要求,全国各高等院校相关专业都把 C51 单片机课程作为核心骨干课程。

为了更好地帮助各大院校将单片机技术课程建设好,我们组织了具有丰富教学经验的教师和企业一线单片机工程师共同研究课程标准,打造立体化精品教材。

本书力求做到高实用性,目标是让学习者在完成相应的学习后即可进行相应的简单 51 单片机设计,所以对于基础知识的介绍均将其融入有实际应用价值的实例中进行,并且针对所有的应用实例均提供了对应的电路原理设计图和 C51 语言代码。本书的全部 C51 语言代码均在 Keil μ Vision 软件开发环境中完成(V4.0),电路图大部分均在 Protel 99SE 中绘制(部分在 Proteus 中绘制)。

本书在写作中突出以下特点:

- 按照由浅入深、循序渐进的原则介绍了 51 单片机应用系统的结构、指令系统、C51 语言的使用方法,内部硬件资源使用方法、外围器件使用方法;
- 所有内部资源和外部资源的使用方法都提供了相应的使用实例;
- 软硬结合,对于所有的应用实例均给出了完整的电路原理设计图以及对应的 C51 语言代码。

为方便教学,本书配备了 PPT 课件、源代码等丰富的教学资源,任课教师可到人民邮电出版社教学服务与资源网(www.ptpedu.com.cn)免费下载使用。

本书由汤嘉立任主编,李林、胡羽、周安华任副主编。

由于编者水平有限,书中疏漏甚至谬误之处难免,希望读者批评、指正,也欢迎就相关问题展开讨论。

编 者

2014 年 6 月

目 录 CONTENTS

第 1 章 51 单片机基础 1

1.1 51 单片机的发展和常见型号	1	1.2.4 时钟模块	12
1.1.1 51 单片机的发展历史	1	1.3 51 单片机的工作方式	13
1.1.2 常见的 51 单片机	3	1.3.1 复位工作方式	13
1.2 51 单片机的内核结构介绍	4	1.3.2 程序执行方式	14
1.2.1 中央处理器	5	1.3.3 低功耗工作方式	14
1.2.2 存储器	7	1.3.4 其他工作	15
1.2.3 外部引脚	10	1.4 本章总结	15

第 2 章 51 单片机的应用系统设计和软件开发环境 16

2.1 51 单片机应用系统的结构	16	2.3 51 单片机的 Keil μ Vision 软件	
2.2 51 单片机最小应用系统	17	开发环境	25
2.2.1 最小应用系统的构成	17	2.3.1 Keil μ Vision 的界面	25
2.2.2 时钟源	17	2.3.2 Keil μ Vision 的菜单详解	26
2.2.3 复位电路	18	2.3.3 使用 Keil μ Vision	32
2.2.4 供电系统	19	2.4 本章总结	34

第 3 章 51 单片机的 C51 语言基础 35

3.1 C51 语言的数据、运算符和表达式	36	3.3.3 函数的调用	54
3.1.1 数据和数据类型	36	3.3.4 内部函数和外部函数	56
3.1.2 常量和变量	37	3.3.5 变量类型以及存储方式	56
3.1.3 存储器和寄存器变量	38	3.4 C51 语言的数组和指针	59
3.1.4 算术运算、赋值、逻辑运算	40	3.4.1 数组	59
3.1.5 位操作	42	3.4.2 指针	62
3.1.6 自增减、复合和逗号运算	43	3.4.3 数组和指针	63
3.1.7 运算符的优先级	44	3.4.4 字符串和指针	65
3.2 C51 语言的结构	45	3.4.5 数组、指针和函数的联系	66
3.2.1 顺序结构	45	3.4.6 指针数组和指向指针的指针	67
3.2.2 选择结构	45	3.5 C51 语言的自构造类型	68
3.2.3 循环结构	48	3.5.1 结构体	68
3.2.4 其他结构语句	51	3.5.2 联合体（共用体）	71
3.3 C51 语言的函数	51	3.5.3 枚举	73
3.3.1 C51 语言的函数的分类	51	3.6 本章小结	74
3.3.2 函数的定义	52		

第4章 51 单片机的并行 I/O 端口及其应用 75

4.1 数据地址端口 P0 和 P2	75	4.5.4 流水灯的应用代码	82
4.2 普通 I/O 端口 P1	77	4.6 应用案例 4.2——按键指示灯的实现	84
4.3 复用端口 P3	77	4.6.1 51 单片机通过并行端口读入电平	84
4.4 数据—地址总线扩展方法	78	4.6.2 独立按键基础	85
4.5 应用案例 4.1——流水灯的实现	80	4.6.3 按键指示灯的硬件电路	86
4.5.1 51 单片机通过并行端口输出电平	80	4.6.4 按键指示灯的应用代码	86
4.5.2 发光二极管 (LED) 基础	81	4.7 本章总结	87
4.5.3 流水灯的硬件电路	82		

第5章 51 单片机的中断系统和外部中断 88

5.1 51 单片机的中断系统	88	5.5 应用案例——外部中断计数系统的实现	99
5.1.1 51 单片机的中断源	89	5.5.1 51 单片机使用外部中断	99
5.1.2 51 单片机的中断引脚	89	5.5.2 单位数码管基础	100
5.1.3 51 单片机的中断相关控制寄存器	89	5.5.3 三极管基础	103
5.1.4 中断向量地址和中断标志位	91	5.5.4 外部中断计数系统的电路	105
5.2 51 单片机的中断处理过程	93	5.5.5 外部中断计数系统的应用代码	105
5.3 51 单片机的中断服务程序设计	94	5.6 一个低电平触发外部中断的实验	107
5.4 51 单片机的外部中断	95	5.6.1 实验的电路和应用代码	107
5.4.1 外部中断的控制	95	5.6.2 实验的运行结果分析	109
5.4.2 外部中断的检测和响应	96	5.7 本章总结	109
5.4.3 多个外部中断信号的处理	96		

第6章 51 单片机的定时计数器 111

6.1 51 单片机定时计数器的组成	111	6.4.4 定时计数器值的读取	121
6.2 51 单片机定时计数器的寄存器	112	6.5 51 单片机定时计数器的特殊应用	121
6.2.1 工作方式控制寄存器 TMOD	112	6.6 应用案例 6.1——PWM 波形发生器的实现	122
6.2.2 控制寄存器 TCON	113	6.6.1 PWM 波形基础	122
6.2.3 数据寄存器 TH0、TL0、和 TH1、TL1	113	6.6.2 PWM 波形发生器的应用代码	123
6.3 51 单片机定时计数器的工作方式	114	6.6.3 脉冲宽度可调的 PWM 波形发生器	124
6.3.1 工作方式 0	114	6.7 应用案例 6.2——呼吸灯的实现	126
6.3.2 工作方式 1	114	6.7.1 呼吸灯效果实现原理	126
6.3.3 工作方式 2	115	6.7.2 RCL 电路原理	126
6.3.4 工作方式 3	115	6.7.3 呼吸灯的电路	127
6.3.5 定时计数器的中断	116	6.7.4 呼吸灯的应用代码	128
6.4 51 单片机定时计数器的使用	116	6.8 中断服务子程序带来的时间误差分析	130
6.4.1 使用定时功能	116	6.9 本章总结	136
6.4.2 使用计数功能	118		
6.4.3 使用门控信号	120		

第7章 51 单片机的串行通信模块 137

7.1	51 单片机串行通信的一些术语	137	7.7.1	多点数据采集系统的电路结构	152
7.2	51 单片机串行通信模块的组成	138	7.7.2	多点数据采集系统的应用代码	152
7.3	51 单片机串行通信模块的寄存器	138	7.8	C51 语言的输入和输出函数	155
7.3.1	串行通信模块控制寄存器 (SCON)	138	7.8.1	putchar 函数的使用方法	156
7.3.2	串行通信模块数据寄存器 (SBUF)	139	7.8.2	printf 函数的使用方法	157
7.3.3	电源管理寄存器 (PCON)	139	7.8.3	sprintf 函数的使用方法	160
7.4	51 单片机串行通信模块的工作方式和 使用	140	7.9	使用普通 I/O 引脚模拟串行通信 模块	162
7.4.1	工作方式 0	140	7.9.1	I/O 引脚模拟串行通信模块的算法	162
7.4.2	工作方式 1	143	7.9.2	I/O 引脚模拟串行通信模块的 C51 语言代码	163
7.4.3	工作方式 2、3	145	7.10	串行通信模块的波特率自适应	166
7.4.4	串行通信模块的中断	146	7.10.1	串行通信模块的波特率自适应 算法	166
7.5	串行通信模块的特殊应用	146	7.10.2	串行通信模块波特率自适应的 C51 语言代码	167
7.6	应用案例 7.1——51 单片机和 PC 通信系统的实现	147	7.11	串行通信模块的“高速”通信	169
7.6.1	RS-232 接口标准和 MAX232 芯片基础	147	7.11.1	波特率固定的“高速”通信	169
7.6.2	51 单片机和 PC 通信系统的电路	149	7.11.2	波特率可变的“高速”通信应用	170
7.6.3	51 单片机和 PC 通信的应用代码	150	7.12	串行通信模块的波特率误差	171
7.6.4	PC 的串口调试工具	151	7.13	本章总结	173
7.7	应用案例 7.2——多点数据采集 系统的实现	152			

第8章 51 单片机的人机交互接口 174

8.1	数码管基础和应用	175	8.5.1	蜂鸣器的基础	197
8.1.1	多位数码管介绍	175	8.5.2	蜂鸣器的电路	198
8.1.2	多位数码管驱动芯片 MAX7219	179	8.5.3	蜂鸣器的操作步骤和驱动函数	199
8.2	1602 液晶模块基础和应用	185	8.6	应用案例 8.1——简易频率计的 实现	199
8.2.1	1602 液晶模块基础	186	8.6.1	51 单片机的频率测量算法	200
8.2.2	1602 液晶模块的电路	189	8.6.2	简易频率计的电路结构	200
8.2.3	1602 液晶模块的操作步骤和驱动 函数	189	8.6.3	简易频率计的应用代码	201
8.3	拨码开关基础和应用	191	8.7	应用案例 8.2——数字输入模块的 实现	204
8.3.1	拨码开关基础	192	8.7.1	数字输入模块的工作原理	204
8.3.2	拨码开关的电路	192	8.7.2	数字输入模块的电路结构	204
8.3.3	拨码开关的操作步骤	193	8.7.3	数字输入模块的应用代码	205
8.3.4	拨码开关的应用实例	193	8.8	应用案例 8.3——简易电子琴的 实现	209
8.4	行列扫描键盘基础和应用	195	8.8.1	乐音的基础知识	209
8.4.1	行列扫描键盘基础	195	8.8.2	简易电子琴的电路结构	210
8.4.2	行列扫描键盘的电路	196	8.8.3	简易电子琴的应用代码	211
8.4.3	行列扫描键盘的操作步骤和驱动 函数	196	8.9	本章总结	215
8.5	蜂鸣器基础和应用	197			

第9章 51单片机的通信接口 216

9.1 51 单片机通信接口基础	216	9.3.2 I ² C 总线接口	221
9.1.1 串行通信和并行通信	217	9.3.3 1-wire 总线接口	230
9.1.2 有线通信和无线通信	217	9.4 51 单片机的并行通信接口	234
9.2 51 单片机应用系统的通信模型和通信协议设计	218	9.4.1 双口 RAM IDT7132 基础	234
9.2.1 51 单片机应用系统的通信模型	218	9.4.2 双单片机使用双口 RAM 进行数据通信	236
9.2.2 51 单片机应用系统的通信协议设计	219	9.5 51 单片机的串行通信接口	238
9.3 51 单片机应用系统的常用外部通信接口	219	9.5.1 RS-422 通信协议	238
9.3.1 SPI 总线接口	219	9.5.2 RS-485 通信协议	240
		9.5.3 光电隔离器	242
		9.6 本章总结	243

第10章 51单片机的 A/D 和 D/A 通道 244

10.1 51 单片机的 A/D 采集通道基础	244	10.4 八位串行单通道 D/A 芯片 MAX517	259
10.1.1 A/D 变换的过程	245	10.4.1 MAX517 应用基础	259
10.1.2 A/D 变换的应用电路构成	247	10.4.2 MAX517 的电路	260
10.1.3 A/D 变换的保持电路	247	10.4.3 MAX517 的操作步骤和驱动函数	261
10.1.4 A/D 芯片的分类	248	10.5 应用案例 10.1——自动换挡电压表的实现	262
10.1.5 A/D 芯片的选择	248	10.5.1 电压表的挡程和自动换挡原理	262
10.1.6 A/D 芯片对电源的需求	249	10.5.2 单片机应用系统中的信号放大	262
10.2 8 位并行 8 通道 A/D 芯片 ADC0809	249	10.5.3 自动换挡电压表的电路结构	264
10.2.1 ADC0809 基础	249	10.5.4 自动换挡电压表的应用代码	265
10.2.2 ADC0809 的电路	251	10.6 应用案例 10.2——简易波形发生器的实现	270
10.2.3 ADC0809 的操作步骤和驱动函数	252	10.6.1 简易波形发生器设计基础	270
10.3 51 单片机的 D/A 输出通道基础	256	10.6.2 简易波形发生器的电路设计	271
10.3.1 D/A 转换的过程	257	10.6.3 简易波形发生器的应用代码	271
10.3.2 D/A 模块的分类	258	10.7 本章总结	277
10.3.3 D/A 芯片的选择	258		
10.3.4 A/D 芯片对电源的需求	259		

第11章 51单片机的温度和时间采集模块 278

11.1 在 51 单片机应用系统中获取温度	279	11.5 应用案例 11.1——多点温度采集系统的实现	293
11.2 DS18B20 温度传感器	280	11.5.1 多点温度采集系统的设计	293
11.2.1 DS18B20 基础	280	11.5.2 多点温度采集系统的电路结构	293
11.2.2 DS18B20 的电路	282	11.5.3 多点温度采集系统的应用代码	294
11.2.3 DS18B20 的操作步骤和驱动函数	284	11.6 应用案例 11.2——简单数字时钟的实现	302
11.3 51 单片机的时间采集通道	285	11.6.1 简单数字时钟的设计	302
11.4 并行总线接口时钟模块 DS12C887	286	11.6.2 简单数字时钟的电路结构	302
11.4.1 DS12C887 基础	286	11.6.3 简单数字时钟的应用代码	303
11.4.2 DS12C887 的电路	289	11.7 本章总结	304
11.4.3 DS12C887 的操作步骤和驱动函数	290		

0.1 C51 语言程序设计技巧	305	0.3.3 一个用户库函数的应用实例	322
0.1.1 养成好的编程习惯	305	0.4 C51 语言的编译常见报警错误 以及解决办法	323
0.1.2 宏定义	307	0.4.1 变量未被使用警告 (Warning 280)	324
0.1.3 条件编译	309	0.4.2 函数未被声明警告 (Warning C206)	324
0.1.4 具体指针的应用	311	0.4.3 头文件无法打开错误 (Error C318)	325
0.1.5 一些关键字的使用	313	0.4.4 函数名称重复定义错误 (Error C237)	326
0.2 C51 语言常用库函数介绍	315	0.4.5 函数未被调用警告	326
0.2.1 C51 语言的库函数基础	315	0.4.6 函数未定义警告 (warning C206)	327
0.2.2 库文件和头文件分类	316	0.4.7 内存空间溢出错误	327
0.2.3 C51 语言的库函数分类介绍	317	0.4.8 函数重入警告	328
0.3 在 Keil μ Vision 中编写用户 自己的库函数	321	0.4.9 常见编译器错误列表	328
0.3.1 用户库函数的建立步骤	321		
0.3.2 用户库函数的引用步骤	322		

PART 1

第1章

51 单片机基础

单片机是单片微型计算机的简称,是一种将运算控制器、存储器、寄存器 I/O 接口以及一些常用的功能模块都集成到一块芯片上的计算机,常常用于工业控制、小型家电等需要嵌入式控制的场合。根据内核的不同,单片机可以分为不同的类属,其中最常用的内核是 Intel 公司设计的 8051,其被 ATMELE、飞利浦、宏晶科技等公司采用,从而生产出了一大批具有相同内核构造但是有不同功能的单片机,它们被统称为 51 系列单片机。

知识目标

- 51 单片机的发展历史。
- 51 单片机的分类和常见型号介绍。
- 51 单片机的内核结构介绍。
- 51 单片机工作方式介绍。

1.1 51 单片机的发展和常见型号

距第一片 51 单片机被生产出来至今已经有 40 多年,在这么多年中单片机经历了长足的发展,在相同内核的基础上形成了多种分类,有超过 100 种的具体型号。

1.1.1 51 单片机的发展历史

单片机的发展史可以大致的分为如下 3 个阶段。

(1) 从 20 世纪 70 年代后期开始,单片机从功能简单的 4 位逻辑控制器件发展到了功能比较强大的 8 位单片机,片内有 8 位微处理器、8 位并行数据总线、8 位定时计数器及一定容量的存储器,并且有了简单的中断功能。这个时期,8 位单片机的代表是 Intel 公司推出的 MCS-48 系列单片机(MCS 即 Micro Computer System,是微型计算机系统的缩写)、GI 公司的 PIC1650 系列单片机。

(2) 1980 年~1981 年,Intel 公司在 MCS-48 系列单片机的基础上增加了串行接口,定时计数器扩展到 16 位,增强了中断系统的功能并且扩大了存储器,推出了 MCS-51 系列单片机。同时期, Motorola 公司推出了 M6800 系列单片机, Zilog 公司推出了 Z8 系列单片机。

(3) 从 20 世纪 80 年代中期开始, Intel 公司在 MCS-51 系列单片机的基础上将内部数据总线扩展为 16 位,外部 I/O 总线仍保持 8 位,推出了 MCS-96 系列单片机。

自从 Intel 公司发布 MCS-51 内核以来,许多公司在 MCS-51 内核基础上进行改进、增强,推出了具有不同特色的、功能更加丰富的基于 MCS-51 内核的单片机,这些单片机或具有 AD 接口,或具有 USB 控制接口,或具有 MP3 解码器,但是由于它们都采用相同的内核,所以被统称为 MCS-51 系列单片机,它们有大致相同的体系结构,有相同的基本指令系统,可以采用相同的开发工具。

进入 20 世纪 90 年代后,随着微电子技术的发展,MCS-51 系列单片机的发展呈现以下的趋势。

- 集成度提高:多种功能都集成在一块 51 单片机上,能够不用扩展外部资源或者扩展很少的外部资源就可以完成系统的功能。
- 扩展方式增多:MCS-51 系列单片机不仅仅使用并行端口和串行端口进行扩展,还出现了 SPI、I²C 等多种总线扩展接口。
- 工作电压降低:MCS-51 系列单片机的工作电压从开始的 5V 降低到 3.3V 和 1.8V,低功耗带来了更加稳定的系统可靠性获得了在便携系统中更加持久使用时间。

除了常见的 MCS-51 系列单片机之外,AVR 系列单片机和 MSP320 系列等单片机也常常被应用于单片机系统中,和 MCS-51 系列单片机相比,它们有独特的应用场合。

图 1.1 和图 1.2 所示分别是两个 MCS-51 单片机应用系统的实物图,前者是一个基于 STC89C52 单片机(MCS-51 单片机的具体型号)的一个多电压输出核心板;后者是一个由 3 块 AT89S52 单片机(MCS-51 单片机的具体型号)为主要控制器搭建的带数码管显示和蜂鸣器报警的继电器控制系统。

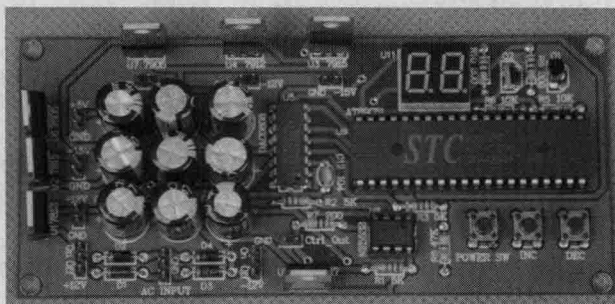


图 1.1 以 STC89C52 单片机为核心的多电压输出核心板

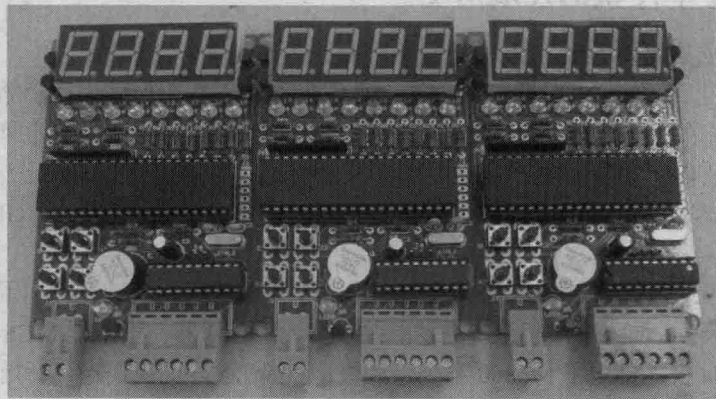


图 1.2 以 AT89S52 单片机为核心的继电器控制系统

1.1.2 常见的 51 单片机

目前在市场上有超过 100 种具体的 51 单片机型号, 其中被使用最多的分别是 ATMELE 公司的 AT89S52、NXP (原飞利浦半导体) 公司的 P87C51x2 和中国本土宏晶科技的 STC89C52。

1. AT89S52

ATMELE 公司是目前最著名的 MCS-51 系列单片机生产厂商之一, AT89S52 是其推出的一款在系统可编程 (ISP-In System Programmed) 单片机, 通过相应的 ISP 软件和一根并行接口或者串行接口下载线, 用户可以对单片机进行编程操作, 图 1.3 所示是 AT89S52 单片机的实物示意, 有 DIP-40 (Dual In-line Package, 双列直插封装)、PLCC-44 (Plastic Leaded Chip Carrier, 带引线的塑料芯片载体) 等多种封装形式, 其主要特性概述如下。

- 提供了有三级安全保护的 8K 在系统可编程 FLASH 程序存储器和 256 字节的内部数据存储器。
- 可以在 4.0~5.5V 的电压下工作。
- 提供双数据指针可以使得程序运行得更快。
- 最高工作频率可以达到 33MHz。
- 提供 32 个可编程 I/O 引脚。
- 内置 3 个 16 位定时计数器。
- 内置一个全双工的串行通信口。
- 支持 ISP 程序下载。
- 7 个中断源, 支持在掉电模式下响应中断。

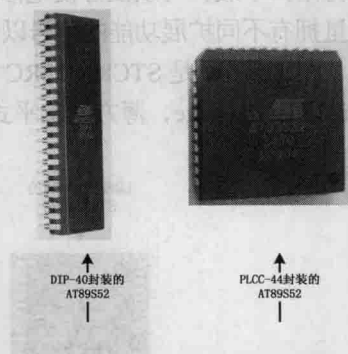


图 1.3 AT89S52 的实物示意

注意: 本书中所有的实例均是基于 AT89S52 单片机的。

2. P87C51x2

NXP (恩智浦) 是 2006 年末从飞利浦公司独立出来的半导体公司, 其业务已拥有 50 年的悠久历史, 主要提供各种半导体产品与软件, 其提供了大量 MCS-51 系列单片机, 包括 Flash、OTP (一次性编程)、ROM 和无 ROM 器件, 其中最常用的型号是 P87C51x2, 图 1.4 所示是其 DIP-40 形式封装的实物示意, 同样具有 PLCC 和 DIP 两种封装形式, 其主要特性概述如下。



图 1.4 DIP-40 封装的 P87C51x2 的实物

- 提供了 4K 字节 EPROM、128 字节 RAM 和布尔处理器。
- 全静态操作, 支持低电压操作 (工作电压为 2.7~5.5V)。
- 支持 6 时钟工作模式, 最高工作频率可以达到 33MHz。
- 提供了双数据指针 DPTR、OTP 保密位。
- 提供了 6 个中断源, 和标准 MCS-51 单片机相比增加了 4 个中断优先等级。

- 提供了 3 个 16 位的定时/计数器，其中 T2 支持捕获和比较功能，并且可以提供可编程时钟输出。
- 提供了 EMI（禁止 ALE，输出斜率控制和 6 时钟模式）功能。

3. STC89C52RC

STC89C52RC 是大陆的单片机设计公司宏基科技的基础单片机型号之一，其最大的特点是支持串口下载，可以很方便地修改内部软件，非常适合制作开发板和系统原型；此外其提供了大量拥有不同扩展功能的型号以供用户选择。

图 1.5 所示是 STC89C52RC 的实物示意，提供了 DIP-40、PLCC-44 和 LQFP-44(Low profile Quad Flat Package，薄方型扁平式封装) 3 种不同的封装，其主要特性概述如下。

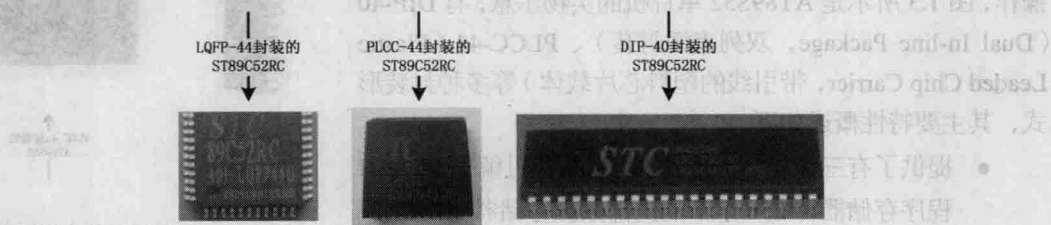


图 1.5 STC89C52RC 的实物

- 提供了 8K 字节的 FLASH 存储器、5K 字节的 E²PROM 和 512 字节的 SRAM 空间。
- 工作电压为 3.5~5.5V，最高工作频率可以达到 40MHz。
- 内置 3 个可编程定时计数器，提供了 39 个可编程 I/O 引脚端口（增加了 P4 口并且可以位寻址）。
- 支持掉电唤醒外部中断，内置复位系统和看门狗。
- 支持 ISP（在系统编程）和 IAP（在应用编程），可以通过串口进行编程操作。
- 价格低廉。

1.2 51 单片机的内核结构介绍

51 单片机系统通常由 8 位中央处理器、时钟模块、I/O 端口、内部程序存储器、内部数据存储器、2 个 16 位定时计数器、中断系统和一个串行通信模块组成，如图 1.6 所示。

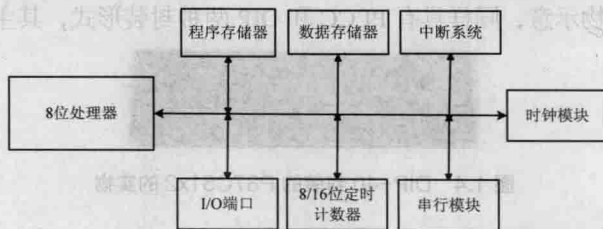


图 1.6 51 单片机的内部结构

51 单片机内部模块的功能如下。

- 8 位处理器：这是 51 单片机的核心部件，执行预先设置好的代码，负责数据的计算和逻辑的控制等。

- 程序存储器：用于存放待执行的程序代码。
- 数据存储器：用于存放程序执行过程中的各种数据。
- 中断系统：根据 51 单片机相应的寄存器的设置来监测和处理单片机的各种中断事件并且提交给处理器处理。
- 时钟模块：以外部时钟源为基准，产生单片机各个模块所需要的各个时钟信号。
- 串行模块：根据相应的寄存器设置进行串行数据通信。
- 8/16 位定时计数器：根据相应寄存器的设置进行定时或者计数。
- I/O 端口：作为数据、地址或者控制信号通道和外围器件进行数据交换。

1.2.1 中央处理器

8 位处理器是单片机的核心模块，由运算逻辑模块和控制逻辑模块组成。运算逻辑模块由算术逻辑运算单元 ALU、累加器 A、寄存器 B、暂存寄存器 TR、程序计数器 PC、程序状态字寄存器 PSW、堆栈指针 SP、数据指针寄存器 DPRT 以及布尔处理器组成。控制逻辑模块则由指令寄存器、指令译码器和定时控制逻辑电路等组成。

1. 算术逻辑运算器 ALU

算术逻辑运算器 ALU 主要负责对数据进行算术运算操作和逻辑运算操作，具体的运算操作如下。

- 带进位加法。
- 不带进位加法。
- 带借位减法。
- 8 位无符号数乘、除法。
- 自加 1、自减 1 操作。
- 左右移位操作。
- 半字节交换。
- 比较和条件转移等操作。

以上的操作都对应专用的指令，将在后面的小节进行详细介绍。ALU 的操作数一般存放在累加器 ACC 或者暂存寄存器 TR 中，运算结果则可以选择保存在 ACC、通用寄存器或者其他普通存储单元中。而在乘除法运算中，使用寄存器 B 中存放一个操作数并且在运算结束之后存放 8 位结果数据。

注意：操作数是指令的操作对象，运算结果则为指令的操作结果，指令则是对操作数进行操作的命令。

2. 累加器 ACC 和寄存器 B

累加器 ACC 是处理器模块中使用最为频繁的寄存器，全部的算术运算操作以及绝大多数的数据传送操作都要使用 ACC。

- 加法和减法：使用 ACC 存放运算结果。
- 乘法：使用 ACC 存放一个操作数，使用寄存器 B 存放另外一个操作数，运算结果则放在 ACC 和寄存器 B 组成的 AB 寄存器对中。
- 除法：使用 ACC 存放被除数，使用寄存器 B 存放除数，计算得到的商数放在 ACC 中，

而余数放到寄存器 B。

3. 程序状态字寄存器 (PSW)

程序状态字寄存器 PSW 用于指示程序运行过程中的系统相关状态,其中 7 位用于存放 ALU 单元运算结果的特征信息,1 位为保留位未使用,程序状态字寄存器的具体含义如表 1.1 所示。

表 1.1 程序状态字

PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
CY	AC	F0	RS1	RS0	OV	保留位	P

程序状态字的内部位定义如下。

- **CY**: 进位、借位标志,在计算过程中如果有进位、借位产生时该位被置 1,否则清零;
- **AC**: 半进位标志,当参与计算的数据第 3 位向第 4 位有进位或者借位产生时,该位被置 1,否则清零;
- **F0**: 供用户自由使用的标志位,常常用于控制程序的跳转,需要用户自己控制其置 1 或者清零;
- **RS1, RS0**: 寄存器组选择位,由用户自行置 1 或者清零,用于选择使用的工作寄存器区,RS1、RS0 和对应的工作寄存器组见表 1.2。
- **OV**: 溢出标志位,当带符号数的运算结果超出 $-128 \sim +127$ 的范围、无符号数运算结果超过 255 或者无符号除法除数为 0 时 OV 被置 1,否则被清零。
- **P**: 奇偶标志位,用于表示累加器 ACC 中“1”的个数,当该个数为奇数时,P 标志被置 1,否则被清零。

表 1.2 RS1 和 RS0 赋值和对应的工作寄存器

RS1、RS0	寄存器组 (地址单元)
00	寄存器组 0 (00H ~ 07H)
01	寄存器组 1 (08H ~ 0FH)
10	寄存器组 2 (10H ~ 17H)
11	寄存器组 3 (18H ~ 1FH)

51 系列单片机共有 4 个寄存器组,每个寄存器组含有 8 个单字节寄存器,这些寄存器常用于保存程序执行过程中各个变量的值,合理地使用寄存器组切换有利于加快程序代码的执行速度。

4. 布尔处理器

布尔处理器用于 51 单片机的位操作,在位操作中使用进位标志 CY 作为累加器,可以对位变量进行置位、清除,取反,位逻辑与,位逻辑或,位逻辑异或,数据传送以及相应的判断跳转操作,位操作是 51 单片机中非常重要的的操作,充分体现了嵌入式处理器的特点。

5. 程序计数器 PC

程序计数器 PC 是一个 16 位计数器,用于存放下一条指令在程序存储器中的地址,可寻址

范围为 0~64KB。

6. 指令寄存器 IR 和指令译码器

指令寄存器 IR 用于存放 51 单片机当前正在执行的指令,而指令译码器对 IR 中指令操作码进行分析解释,产生相应的控制逻辑。

7. 数据指针 DPTR

数据指针 DPTR 用于寻址外部数据存储器,寻址范围为 0~64KB。

8. 堆栈指针 SP

堆栈是一种将数据按序排列的数据结构,51 单片机的堆栈是内存中一段连续的空间,堆栈指针 SP 用来指示堆栈顶部在单片机内部数据存储器中的位置,可以由用户的程序代码修改。当执行进栈操作时 SP 自动加 1,然后把数据放入堆栈,当执行出栈操作时 SP 自动减 1,然后把数据送出堆栈。当单片机被复位后 SP 初始化为 0x07H。

1.2.2 存储器

51 系列单片机的存储器采用的是哈佛结构,其分别有独立的寻址指令、编址空间和相应的控制寄存器,图 1.7 所示是 51 系列单片机的存储器组成结构,从中可以看到 51 单片机的存储器可以分为片内程序存储器、片外程序存储器、片内数据存储器 and 片外数据存储器 4 个部分,每个部分都有独立的地址编码。

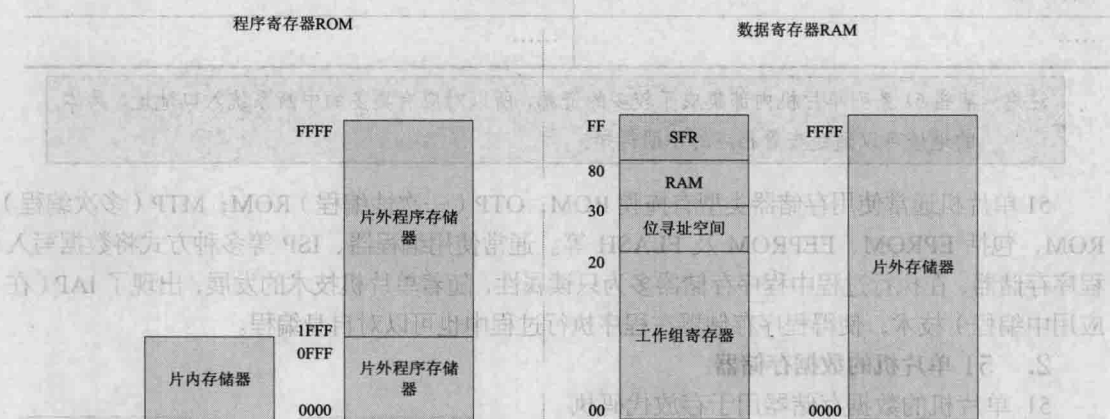


图 1.7 51 单片机的存储器组织

注意: 51 单片机的存储器包含有很多存储单元,为区分不同的内存单元,单片机对每个存储器单元进行编号,存储器单元的编号就称为存储器单元的地址,每个存储器单元存储的若干位二进制数据成为存储器单元的数据。

1. 51 单片机的程序存储器

51 系列单片机的程序存储器由片内程序存储器和片外程序存储器组成,用于存放待执行的程序代码。因为 PC 程序指针和地址总线是 16 位的,所以片内和片外的程序存储器最大编址总和为 64KB,其中外部程序存储器的低部分编址和内部程序存储器的编址重合,代码只能选择存放其中一个地方,使用外部引脚 EA 来选择。当该引脚加上高电平时,PC 程序指针起始指向的是内部程序存储器,程序代码从内部存储器开始执行;当该引脚加上低电平时,PC 程序指针起始指向的是外部程序存储器,程序代码从外部程序存储器开始执行。

单片机复位之后, PC 程序指针被初始化为 0x000, 指向程序代码空间的最低位, 程序指针在程序执行过程汇总并自动增加指向下一条待执行的指令, 当程序代码大小超过了内部程序存储器时, 我们需要为单片机外扩外部程序存储器, 程序代码执行到存放内部程序存储器的最后一条指令之后自动跳转到外部程序寄存器继续执行。

51 的内部程序存储器的部分地址用于中断系统, 是这些中断服务子程序的程序代码的入口, 一般在该位置放置相应的跳转指令, 使得 PC 程序指针跳转到相应的程序代码块起始存放地址, 如表 1.3 所示。

表 1.3 51 单片机的中断系统入口地址

特定程序	入口地址
系统复位	0x0000H
外部中断 0	0x0003H
定时器 T0 溢出	0x000BH
外部中断 1	0x0013H
定时器 T1 溢出	0x001BH
串行中断	0x0023H
.....	

注意: 某些 51 系列单片机内部集成了较多的资源, 所以对应有更多的中断系统入口地址, 具体的地址可以通过查看相应的手册得知。

51 单片机通常使用存储器类型有掩膜 ROM; OTP (一次性编程) ROM; MTP (多次编程) ROM, 包括 EPROM、EEPROM 及 FLASH 等。通常使用编程器、ISP 等多种方式将数据写入程序存储器, 在执行过程中程序存储器多为只读属性, 随着单片机技术的发展, 出现了 IAP (在应用中编程) 技术, 使得程序存储器在程序执行过程中也可以对自身编程。

2. 51 单片机的数据存储器

51 单片机的数据存储器用于存放代码执行过程中的相关数据, 由片内存储器和片外存储器组成。片内存储器有可以划分为数据 RAM 区和特殊功能寄存器 (SFR) 区, 而数据 RAM 区又可以划分为工作寄存器区、位寻址区、用户区和堆栈区, 51 单片机片内数据 RAM 区如图 1.8 所示, 共 256 字节。

片内存储器最低空间是工作寄存器区, 分为 4 组, 每组 8 个单字节寄存器, 均编码为 R0~R7, 共 32 个单元。当前正在使用的组由程序状态字寄存器 PSW 中的 RS0 和 RS1 位决定, 由于可以直接使用单字节指令访问, 因此放在其中的数据访问速度是最快的。

位寻址区的 16 个字节单元支持位寻址, 用户可以使用普通的内存寻址指令对该部分内存单元进行字节寻址, 也可以使用位寻址指令对该部分内存单元按照对应的位地址进行位寻址, 该

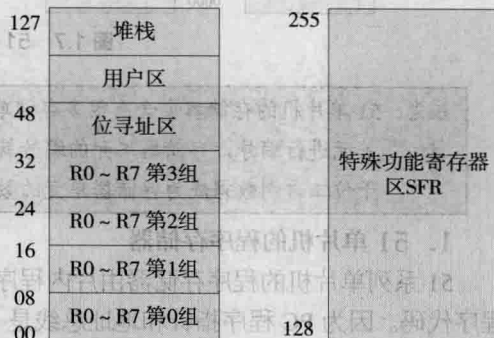


图 1.8 51 单片机的内部 RAM 分布