



普通高等教育“十二五”规划教材

单片机原理、接口及应用

DANPIANJI YUANLI JIEKOU JI YINGYONG

主 编 关丽荣
副主编 岳国盛 韩 辉



国防工业出版社

National Defense Industry Press

单片机原理、接口及应用

主 编 关丽荣

副主编 岳国盛 韩 辉

国防工业出版社

· 北京 ·

内 容 简 介

本书以 Intel 公司 MCS-51 单片机为例,系统地介绍了单片机原理、接口及应用,全书共分 11 章,第 1、2 章介绍微型计算机基础知识及单片机的内部结构与工作原理;第 3 章详细阐述单片机指令功能及使用方法;第 4 章以实例介绍单片机主要实用汇编语言程序设计;第 5、6 章介绍单片机存储器扩展技术及中断系统;第 7 章详述单片机内部并行口、串行口和定时器的应用及其与外部 LED 数码管、键盘接口电路设计及应用;第 8 章介绍单片机与 A/D、D/A 接口电路设计及应用;第 9、10 章简要介绍 Keil μ Vision IDE 开发环境和 Proteus ISIS 仿真软件的使用方法;第 11 章给出针对本课程的单片机应用系统实训题目。全书注重基础,强调应用,以大量简单易懂的典型实例来对单片机软、硬件设计进行详细说明和论述,同时为了加深理解教材内容,前 8 章书后都配有一定数量的习题与思考题。

本书可作为高等学校机械类专业及相关非电子类专业本科教材,也可作为相关专业专科教材,还可作为从事单片机系统应用开发的工程技术人员的初级参考书。

图书在版编目(CIP)数据

单片机原理、接口及应用/关丽荣主编. —北京:国防工业出版社,2015.8
ISBN 978-7-118-10281-9

I. ①单... II. ①关... III. ①单片微型计算机
IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2015)第 171878 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 14 $\frac{3}{4}$ 字数 360 千字

2015 年 8 月第 1 版第 1 次印刷 印数 1—4000 册 定价 29.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)88540777

发行邮购: (010)88540776

发行传真: (010)88540755

发行业务: (010)88540717

前 言

单片机具有体积小、集成度高、控制功能强、性能价格比高等独特的优点,在工业控制、智能化仪表、数控设备、数据采集、通信及家用电器等领域中得到了广泛的应用。为了满足市场对应用型本科人才的需要,使应用型本科院校的学生在短时间内能更好地掌握单片机研发技术,同时也为适应单片机原理及应用课程教学改革,作者结合多年一线教学和实践经验及大量考察同类教材的基础上,经过深刻总结之后编写了本书。

本书编写注重基础,强调应用,与其他同类教材相比,主要特色如下:

- (1) 从应用研发角度完整介绍单片机应用系统开发流程。
- (2) 以浅显的语言讲述单片机内部各个模块的工作原理及功能。
- (3) 以实用为主选择汇编语言程序,介绍设计方法。
- (4) 从实际出发,考虑实验现象,以简单的信号传递讲述接口技术及编程。

(5) 考虑实验及机械系统实训需要,补充 Keil μ Vision IDE 开发环境和 Proteus ISIS 仿真软件内容,并给出一些简单实训题目。

(6) 紧密结合教材内容编写了课后习题与思考题,既强化重点内容,又进一步加深拓展知识点。

本书的编写是希望高等学校机械类专业及相关非电子类专业学生在学习单片机原理、接口及应用时能从实际工程的角度出发,在掌握单片机基本原理及应用技术的基础上,领会单片机原理及应用这门课程的工程应用技能,适应市场需要。

本书由沈阳理工大学关丽荣任主编,岳国盛、韩辉任副主编,杨旗参编。全书共分 11 章,其中,第 1、2、4、7 章由关丽荣编写;第 5、8、10、11 章由岳国盛编写;第 3、6、9 章由韩辉编写;全书由关丽荣统稿。在编写本书的过程中,借鉴了一些参考文献中提到的一些成果,在此诚挚感谢作者们,书籍的成稿有他们的贡献。

尽管我们竭尽全力,但毕竟自身水平有限,书中难免有错误及疏漏之处,恳请读者和同行批评指正。

编 者

2015 年 5 月

目 录

第1章 微型计算机基础知识	1
1.1 计算机中的数制、数的转换及表示形式	1
1.1.1 计算机中的数制	1
1.1.2 数制之间的转换	2
1.1.3 计算机中数的表示形式	3
1.2 计算机中数和字符的编码	5
1.3 单片机概述	6
1.3.1 单片机的概念及特点	6
1.3.2 单片机的应用	7
1.3.3 单片机的开发环境	7
1.3.4 单片机程序开发流程	8
习题与思考题	9
第2章 MCS-51 单片机结构与原理	10
2.1 MCS-51 单片机内部结构	10
2.1.1 CPU	10
2.1.2 存储器	13
2.1.3 并行 I/O 口	16
2.1.4 可编程串行口	20
2.1.5 定时器/计数器	20
2.1.6 中断系统	20
2.2 MCS-51 单片机外部引脚	21
2.3 MCS-51 单片机工作方式	23
2.4 MCS-51 单片机时序	24
习题与思考题	27
第3章 MCS-51 单片机指令系统	29
3.1 指令系统概述	29
3.1.1 指令格式	29
3.1.2 指令的字节数	30
3.1.3 汇编指令常用符号	31
3.2 寻址方式	31
3.2.1 立即寻址	32
3.2.2 直接寻址	32
3.2.3 寄存器寻址	33

3.2.4	寄存器间接寻址	33
3.2.5	变址寻址	34
3.2.6	相对寻址	34
3.2.7	位寻址	35
3.3	数据传送指令	35
3.3.1	内部数据传送指令	35
3.3.2	外部数据传送指令	37
3.3.3	堆栈操作指令	38
3.3.4	数据交换指令	39
3.4	算术、逻辑运算及移位指令	40
3.4.1	算术运算指令	40
3.4.2	逻辑操作指令	43
3.4.3	循环移位指令	45
3.5	控制转移指令	45
3.5.1	无条件转移指令	45
3.5.2	条件转移指令	47
3.5.3	子程序调用和返回指令	48
3.6	位操作指令	50
	习题与思考题	52
第4章	MCS-51 单片机程序设计	54
4.1	单片机编程语言概述	54
4.1.1	编程语言概述	54
4.1.2	单片机使用的编程语言	55
4.2	汇编语言的构成和汇编	55
4.2.1	汇编语言的构成	55
4.2.2	汇编语言的汇编	58
4.3	分支程序设计	59
4.4	循环程序设计	61
4.5	查表、子程序设计	63
4.5.1	查表程序设计	63
4.5.2	子程序设计	64
4.6	数制转换程序设计	65
4.7	延时程序设计	66
4.8	C51 基础知识	67
4.8.1	C 语言与 C51	67
4.8.2	学习 C51 的必备条件	67
4.8.3	单片机 C 语言程序开发流程	68
4.8.4	C51 中的数据类型	68
4.8.5	C51 中常用的头文件	69
4.8.6	C51 中的运算符	69

4.8.7 C51 中的基础语句	70
4.8.8 C51 中的函数	71
习题与思考题	71
第5章 存储器扩展	73
5.1 概述	73
5.1.1 只读存储器	73
5.1.2 随机存储器	74
5.1.3 存储器扩展技术	74
5.2 程序存储器	76
5.3 数据存储器	80
5.4 MCS-51 单片机外部存储器的扩展	84
5.4.1 程序存储器扩展电路	84
5.4.2 数据存储器扩展电路	84
5.4.3 Flash 存储器的扩展	87
习题与思考题	90
第6章 MCS-51 单片机中断系统	91
6.1 概述	91
6.1.1 中断的定义和作用	91
6.1.2 MCS-51 中断源	91
6.1.3 中断嵌套	92
6.1.4 中断系统功能	93
6.2 MCS-51 中断系统及应用	94
6.2.1 MCS-51 的中断源和中断标志	94
6.2.2 MCS-51 对中断请求的控制	96
6.2.3 MCS-51 对中断的响应	98
6.2.4 MCS-51 对中断的响应时间	99
6.2.5 MCS-51 对中断请求的撤除	99
6.2.6 MCS-51 中断系统的应用	100
习题与思考题	101
第7章 MCS-51 单片机接口电路及典型应用	103
7.1 概述	103
7.1.1 I/O 接口的作用	103
7.1.2 I/O 数据传送方式	104
7.1.3 I/O 接口类型	105
7.2 并行接口及应用	106
7.2.1 MCS-51 内部 I/O 口及应用	106
7.2.2 简单 I/O 接口扩展及应用	107
7.2.3 可编程 I/O 接口芯片及应用	109
7.3 串行接口及应用	117
7.3.1 串行通信的基础知识	118

7.3.2	MCS-51 单片机的串行口	120
7.3.3	MCS-51 单片机串行口应用	124
7.4	定时器/计数器及其应用	128
7.4.1	定时器/计数器的结构和控制	128
7.4.2	定时器/计数器的工作方式	130
7.4.3	定时器/计数器的初始化	131
7.4.4	定时器/计数器的应用	132
7.5	MCS-51 单片机与 LED 显示器接口应用	134
7.5.1	LED 数码管显示原理	134
7.5.2	LED 数码管显示方式	135
7.5.3	MCS-51 与 LED 数码管接口及应用	136
7.6	MCS-51 单片机与键盘接口应用	138
7.6.1	键盘控制原理	138
7.6.2	MCS-51 与键盘接口及应用	142
	习题与思考题	147
第 8 章	MCS-51 单片机与 A/D、D/A 接口及应用	149
8.1	MCS-51 单片机与 A/D 接口及应用	149
8.1.1	A/D 转换原理	149
8.1.2	A/D 转换器的主要技术指标	151
8.1.3	A/D 转换器 ADC0809	152
8.1.4	单片机与 ADC0809 的接口	155
8.1.5	串行接口 A/D 转换器 TLC2543 及其接口技术	158
8.2	MCS-51 单片机与 D/A 接口及应用	159
8.2.1	D/A 转换器工作原理	159
8.2.2	D/A 转换器的主要技术指标	161
8.2.3	D/A 转换器 DAC0832	162
8.2.4	12 位串行 D/A 转换器 TLV5616 及其接口技术	167
	习题与思考题	170
第 9 章	Keil μ Vision IDE 集成开发环境	171
9.1	软件简介	171
9.1.1	Keil μ Vision IDE 的安装	171
9.1.2	Keil μ Vision IDE 界面	171
9.2	Keil 使用方法	174
9.2.1	建立项目	174
9.2.2	创建文件	174
9.2.3	向项目里添加源程序	176
9.2.4	文件的编译、连接	177
9.2.5	仿真器的选择	179
9.2.6	程序的调试	180
9.2.7	Keil μ Vision IDE 的调试技巧	183

第 10 章 Proteus ISIS 仿真软件	185
10.1 软件简介	185
10.2 Proteus 电路设计方法	191
10.2.1 文件操作	191
10.2.2 在原理图中放置和编辑对象	191
10.2.3 连线	194
10.3 Proteus 单片机仿真实例	195
10.3.1 单片机仿真	195
10.3.2 Proteus 与 Keil C 的联合仿真	198
第 11 章 单片机应用系统实训题目	200
附录 MCS-51 系列单片机指令表	220
参考文献	225

第 1 章 微型计算机基础知识

众所周知,计算机都以二进制形式进行算术运算和逻辑操作,微型计算机也不例外。因此,对于用户在键盘上输入的十进制数字和符号命令,微型计算机必须先把它们转换成二进制形式进行识别、运算和处理,然后再把运算结果还原成十进制数字和符号,并在显示器上显示出来。下面就讨论这方面的知识。

1.1 计算机中的数制、数的转换及表示形式

1.1.1 计算机中的数制

数制是人们利用符号计数的一种科学方法。数制有很多种,微型计算机中常用的数制有十进制、二进制、八进制和十六进制等。任何一种数制都有两个要素,即基数和权。基数为数制中所使用的数码的个数。当基数为 R 时,该数制可使用的数码为 $0 \sim R-1$ 。例如,二进制中基数为 2,可使用 0 和 1 这两个数码。现对十进制、二进制和十六进制 3 种数制进行讨论。

1. 十进制 (Decimal)

十进制以 10 为基数,它共有 0、1、2、3、4、5、6、7、8、9 这 10 个数字符号。计数规则是逢十进一,借一当十。任意一个十进制数 N 可表示为

$$N = d_{n-1} \times 10^{n-1} + d_{n-2} \times 10^{n-2} + \cdots + d_1 \times 10^1 + d_0 \times 10^0 + d_{-1} \times 10^{-1} + \cdots + d_{-m} \times 10^{-m} \\ = \sum_{i=-m}^{n-1} d_i \times 10^i$$

式中: d_i 为第 i 位的数码,可取 $0 \sim 9$; 10^i 为第 i 位的权。

显然,各位的权是 10 的幂。 n 为该数整数部分的位数; m 为小数部分的位数。例如:

$$1234.5 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1}$$

2. 二进制 (Binary)

二进制以 2 为基数,它共有 0、1 两个数字符号。计数规则是逢二进一,借一当二。任意一个二进制数 N 可表示为

$$N = d_{n-1} \times 2^{n-1} + d_{n-2} \times 2^{n-2} + \cdots + d_1 \times 2^1 + d_0 \times 2^0 + d_{-1} \times 2^{-1} + \cdots + d_{-m} \times 2^{-m} \\ = \sum_{i=-m}^{n-1} d_i \times 2^i$$

式中: d_i 为第 i 位的数码,可取 0、1; 2^i 为第 i 位的权。

二进制中,各位的权是 2 的幂。 n 、 m 表示与十进制相同。例如:

$$1101.1 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = 13.5$$

3. 十六进制 (Hexadecimal)

十六进制是以 16 为基数,它共有 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 这 16 个数字符

号。计数规则是逢十六进一,借一当十六。任意一个十六进制数 N 可表示为

$$N = d_{n-1} \times 16^{n-1} + d_{n-2} \times 16^{n-2} + \cdots + d_1 \times 16^1 + d_0 \times 16^0 + d_{-1} \times 16^{-1} + \cdots + d_{-m} \times 16^{-m} \\ = \sum_{i=-m}^{n-1} d_i \times 16^i$$

式中: d_i 为第 i 位的数码,可取 0~9、A~F; 16^i 为第 i 位的权。

显然,各位的权是 16 的幂。 n 、 m 表示与十进制相同。例如:

$$12AB.EF = 1 \times 16^3 + 2 \times 16^2 + A \times 16^1 + B \times 16^0 + E \times 16^{-1} + F \times 16^{-2} = 4779.93359$$

在计算机内部,数的表示形式是二进制。这是因为二进制数只有 0 和 1 两个数码,采用晶体管的导通和截止、脉冲的高电平和低电平等都很容易表示它。此外,二进制数运算简单,便于用电子线路实现。但在实际应用中,为了减轻阅读和书写二进制数(特别是位数较长的二进制数)时的负担,常常采用十六进制数描述一个二进制数。

在阅读和书写不同数制的数时,如果不在每个数上外加一些辨认标记,就会混淆,从而无法分清。通常,标记方法有两种:一种是把数加上方括号,并在方括号右下角标注数制代号,如 $[2A]_{16}$ 、 $[101]_2$ 和 $[45]_{10}$ 分别表示十六进制、二进制和十进制数;另一种是用英文字母标记加在被标记数的后面,分别用 B、D 和 H 大写字母表示二进制、十进制和十六进制数,如 2AH 为十六进制数、101B 为二进制数、45D 为十进制数,由于人们生活中采用的数为十进制数,所以标记 D 也可以省略。

1.1.2 数制之间的转换

1. 二进制数和十进制数之间的转换

1) 二进制数转换成十进制数

转换时只要把欲转换的数按权展开后相加即可。例如:

$$1101.11 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 13.75$$

2) 十进制数转换成二进制数

(1) 十进制整数转换为二进制整数常采用除 2 取余法:用 2 连续去除要转换的十进制数,直到商小于 2 为止,然后把各次余数按最后得到的为最高位、最早得到的为最低位,依次排列起来所得到的数便是所求的二进制数。

(2) 十进制小数转换为二进制小数通常采用乘 2 取整法:用 2 连续去乘要转换的十进制小数,直到所得积的小数部分为 0 或满足所需精度为止,然后把各次整数按最先得到的为最高位、最后得到的为最低位,依次排列起来所对应的数便是所求的二进制小数。

【例 1.1】 求出十进制数 25 的二进制数。

解 把 25 连续除以 2,直到商数小于 2,把所得余数按箭头方向从高位到低位排列起来便可得到: $25 = 11001B$ 。相应竖式如图 1.1 所示。

【例 1.2】 求十进制小数 0.706 转换为二进制小数(精确到小数点后 5 位)。

解 把 0.706 不断地乘 2,取每次所得乘积的整数部分,直到乘积的小数部分满足所需精度,把所得整数按箭头方向从高位到低位排列起来,便可得到 $0.706 = 0.10110B$ 。相应竖式如图 1.2 所示。

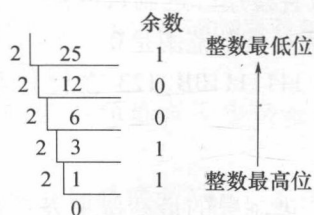


图 1.1 25 转换为二进制数过程

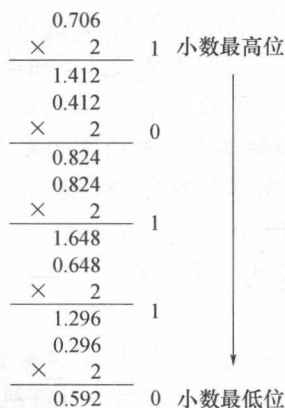


图 1.2 0.706 转换为二进制数过程

2. 十进制数和十六进制数之间转换

1) 十进制数转换成十六进制数

十进制数转换成十六进制数与十进制数转换成二进制数的方法类似,即十进制整数转换为十六进制整数采用除 16 取余法,而十进制小数转换为十六进制小数采用乘 16 取整法,如 $100.76171875 = 64.C3H$ 。相应竖式如图 1.3 所示。

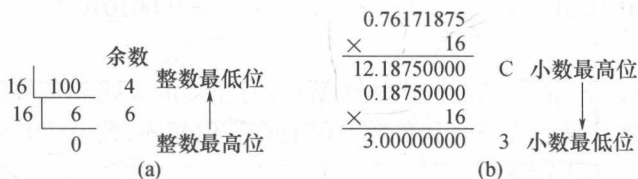


图 1.3 100.76171875 转换为十六进制数过程

(a) 整数部分转换;(b) 小数部分转换。

2) 十六进制数转换成十进制数

十六进制数转换成十进制数与二进制数转换成十进制数的方法类似,即可以把十六进制数按权展开后相加。例如:

$$3FH = 3 \times 16^1 + F \times 16^0 = 63$$

3. 二进制数和十六进制数之间转换

1) 二进制数转换成十六进制数

二者之间转换十分方便,以 4 位为一组,从二进制数的小数点开始,整数部分从低位开始,不足 4 位的前面补 0,小数部分从最高位开始,不足 4 位的后面补 0,然后分别把每组用十六进制数码表示,并按序相连。例如:

$$1011\ 0101.1001\ 1110 = B5.9EH$$

2) 十六进制数转换成二进制数

把十六进制数的每位分别用 4 位二进制数码表示,然后把它们连成一体。例如:

$$CF56H = 1100\ 1111\ 0101\ 0110B$$

1.1.3 计算机中数的表示形式

在现代微型机中,运算器电路的设计非常简单,主要由一个补码加法器、 n 位寄存器/计数

器组和移位控制电路等组成,能进行各种算术运算和逻辑操作。补码加法器既能做加法又能将减法运算变为加法来做。下面就来介绍计算机中的码制。

计算机中的数通常有两种,即无符号数和符号数。

1. 无符号数在计算机中的表示

无符号数不带符号,表示时比较简单,在计算机中一般直接用二进制数的形式表示,位数不足时前面加0补足。对一个 n 位二进制数,它能表示的无符号数范围是 $0 \sim 2^n - 1$ 。例如,假设机器字长为8位,无符号数254在计算机中表示为1111110B,123在计算机中表示为01111011B。

2. 有符号数在计算机中的表示

有符号数带有正负号。计算机中表示有符号数采用二进制数的最高位来表示符号,用0表示正数的符号+;用1表示负数的符号-;其余位表示有符号数的数值大小,称为数值位。通常,把一个数及其符号位在计算机中的表示形式称为机器数。在计算机中,常用的机器数有原码、反码、补码3种形式。

1) 原码

符号位为0表示该数为正数,符号位为1表示它是负数。通常,一个数的原码可以先把该数用方括号括起来,并在方括号右下角加个“原”字来标记。

【例1.3】 设 $X = +1010B$, $Y = -1010B$,请写出 X 和 Y 在8位微型机中的原码。

解 $[X]_{\text{原}} = 00001010B$

$[Y]_{\text{原}} = 10001010B$

2) 反码

在微型计算机中,二进制数的反码求法很简单,有正数的反码和负数的反码之分。正数的反码和原码相同;负数反码的符号位和负数原码的符号位相同,数值位是它的数值位的按位取反。反码的标记方法与原码类似。

【例1.4】 设 $X = +1101101B$, $Y = -0110110B$,请写出 X 和 Y 的反码。

解 $[X]_{\text{反}} = 01101101B$

$[Y]_{\text{反}} = 11001001B$

3) 补码

在日常生活中,补码的概念是经常会遇到的。例如,如果现在是北京时间下午3点钟,而手表还停在早上8点钟。为了校准手表,自然可以顺拨7个小时,但也可倒拨5个小时,效果都是相同的。显然,顺拨时针是加法操作,倒拨时针是减法操作,据此便可得到以下两个数学表达式:

顺拨时针 $8 + 7 = 12(\text{自动丢失}) + 3 = 3$

倒拨时针 $8 - 5 = 3$

顺拨时针时,自动丢失的数12点被称为0点。在数学上,这个自动丢失的数12称为模(mod),这种带模的加法称为按模12的加法,通常写为

$8 + 7 = 3(\text{mod } 12)$

比较上述两个数学表达式,可发现 $8 - 5$ 的减法和 $8 + 7$ 的按模加法等价。这里, $+7$ 和 -5 是互补的, $+7$ 称为 -5 的补码(mod 12)。这就是说, $8 - 5$ 的减法可以用 $8 + [-5]_{\text{补}} = 8 + 7(\text{mod } 12)$ 的加法替代。

在微型计算机中,正数的补码和原码相同;负数补码等于它的反码加1。标记方法同上。

【例1.5】 设 $X = +1101101B$, $Y = -0110110B$,请写出 X 和 Y 的补码。

解 $[X]_{\text{补}} = 01101101B$

$[Y]_{\text{补}} = 11001010B$

1.2 计算机中数和字符的编码

在计算机中,由于机器只能识别二进制数,因此键盘上所有数字、字母和符号也必须事先为它们进行二进制编码,以便机器对它们加以识别、存储、处理和传送。下面介绍两种微型机中常用的编码,即 BCD 码和 ASCII 码。

1. BCD 码

BCD 码是一种具有十进制权的二进制编码。BCD 码的种类较多,这里只介绍 8421 BCD 码。

8421 BCD 码是采用 4 位二进制数的前 10 种组合来表示 0~9 这 10 个十进制数。这种代码每一位的权都是固定不变的,和 4 位二进制数一样,从高位到低位各位的权分别为 8、4、2、1,故称为 8421 BCD 码,如表 1.1 所列。

BCD 数是由 BCD 码构成的,虽然以二进制形式出现,但却不是真正的二进制数,运算之后的结果也必须是 BCD 数。

表 1.1 8421 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

2. ASCII 码(字符编码)

ASCII 码是美国标准信息交换码。由于现代微型计算机不仅要处理数字信息,还要处理大量字母和符号。这就需要人们对这些数字、字母和符号进行二进制编码,以供微型计算机识别、存储和处理。这些数字、字母和符号统称为字符,故字母和符号的二进制编码又称为字符编码。

通常,ASCII 码由 7 位二进制数表示,共 128 个字符编码,如表 1.2 所列。这 128 个字符共分两类:一类是图形字符,共 96 个;另一类是控制字符,共 32 个。96 个图形字符包括十进制数符号 10 个、大小写英文字母 52 个以及其他字符 34 个,这类字符有特定形状,可以显示在 CRT 上以及打印在打印纸上,其编码可以存储、传送和处理。32 个控制字符包括回车符、换行符、退格符、设备控制符和信息分隔符等,这类字符没有特定形状,其编码虽然可以存储、传送和起某种控制作用,但字符本身不能在 CRT 上显示,也不能在打印机上打印。

表 1.2 ASCII 码表

$d_6 d_5 d_4 \backslash d_3 d_2 d_1 d_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	,	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r

(续)

$d_6 d_5 d_4$ $d_3 d_2 d_1 d_0$	000	001	010	011	100	101	110	111
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	←	o	DEL

1.3 单片机概述

1.3.1 单片机的概念及特点

1. 单片机的概念

单片机在外观上与常见的集成电路块一样,体积很小,多为黑色长条状,条状左、右两侧各有一排金属引脚,可与外电路连接,如图 1.4 所示。

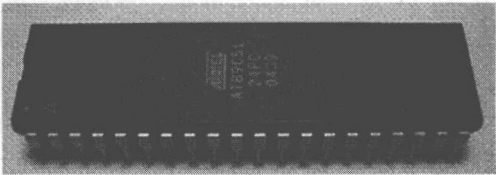


图 1.4 AT89C51 单片机外观

单片机是将计算机的中央处理器(CPU)、存储器(ROM/RAM)和输入输出(I/O)、定时器/计数器(Timer/Counter)、中断(Interruption)系统等集成在一片芯片上,因此,被称为单片微型计算机(Single Chip Microcomputer),简称单片机。

单片机也称为微控制器或嵌入式微控制器。计算机是依靠输入程序来工作的,同样,单片机工作也需要事先输入程序。

2. 单片机的特点

1) 高性能、低价格

一片单片机从功能上讲相当于一台微型计算机,可是价格却很低,一片单片机的价格一般在几元到几十元之间。而且,随着科学技术的发展和市场竞争的加剧,世界上生产单片机的各

大公司都在不断地采用新技术来提高单片机的性能,同时又进一步降低其价格。

2) 体积小、可靠性高

在单片机的芯片内,除了一般必须具有的 ROM、RAM、定时器/计数器、中断系统外,还尽可能地把众多的各种外围功能器件集成在片内,减少了外部各芯片之间的连接,大大提高了单片机的可靠性和抗干扰能力。

3) 低电压、低功耗

一般单片机的工作电压为 5V,有的单片机可以在 1.8 ~ 3.6V 的电压下工作,而且,工耗降至 μA 级。例如, MSP430 超低功耗类型的单片机,两个钮扣电池就可以保障其运行长达近 10 年。单片机的这种低电压、低功耗的特性,对于设计和开发携带式智能产品和家用消费类产品显得非常重要。

1.3.2 单片机的应用

只需在电路中添加少许元器件,通过编写程序就可以实现多种功能的单片机自动控制。单片机接上键盘可以进行数据输入,接上显示器可以实现数据显示,接上喇叭可以实现声音输出;单片机可以用来通信,也可以用来计数和定时,还可以控制彩灯的闪烁、电机的运转以及机器人的活动等,如图 1.5 所示。

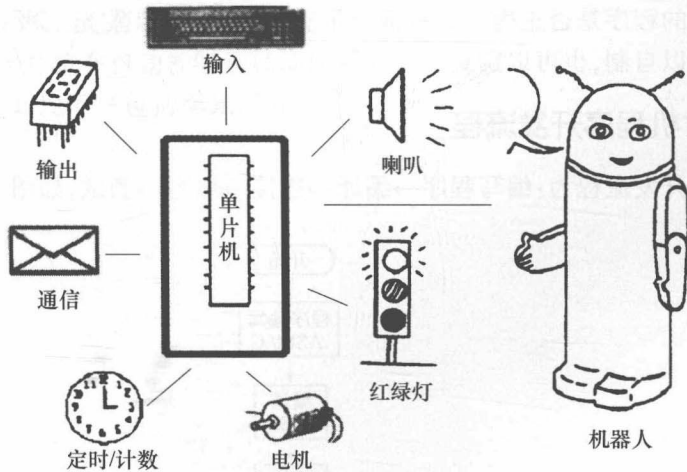


图 1.5 单片机应用示意图

由于单片机体积小、功能强大、应用灵活、价格便宜,所以应用十分广泛。已经在工业控制、国防装备、智能仪器等领域得到了广泛应用。现在,人们日常生活中所使用的各种家用电器,如洗衣机、电冰箱、空调、微波炉、电饭煲、音响、电风扇及高档电子玩具等,也普遍采用了单片机来代替传统的控制电路,既降低了成本,又提高了自动化程度。

1.3.3 单片机的开发环境

学习单片机要建立单片机的开发环境。其主要包括下面几个部分:

1. 计算机

2. 单片机集成开发系统软件

单片机集成开发系统软件,是指用来在计算机上编写、汇编和仿真、调试单片机程序的软件。

目前用来开发单片机的应用软件比较多,如 Keil 公司的 Keil C51(评估展示版)及 Medwin 等,都是比较好的 51 单片机集成开发系统软件。

3. 51 编程器

51 编程器是用来将编好的程序烧写到 51 单片机内的一个设备。

用集成开发系统软件(如 Keil C51 或 MedWin)编写并生成单片机目标代码后,需要用编程器将目标代码,即扩展名为 .HEX 的可执行文件烧写到单片机中。编程器是一个设备,上面有单片机插座及与计算机的连线等。

编程器按功能可分单一型和万能型。单一型编程器只能对单一系列的某些型号的单片机芯片进行写入操作;万能型编程器能对多种系列的多种型号的单片机芯片进行写入操作。前者结构简单、价格便宜;后者功能强大,但价格较贵。

编程器按照与计算机的连接方式不同可分为串口编程器和并口编程器两种。串口编程器通过连线接在计算机的串行端口,即通信端口上;并口编程器通过连线接在计算机的并行端口,即打印机端口上。购买时一般选择串口编程器,串口编程器还可以很方便地进行通信程序实验。

4. 实验板

实验板实际上是一个小的单片机实验系统。写入程序的单片机需要安装到实验板上运行以后才能验证编写的程序是否正确。实验板上带有单片机插座、发光二极管、数码管、蜂鸣器等器件。实验板可以自制,也可以购买。

1.3.4 单片机程序开发流程

单片机的程序开发流程为:编写程序→编译→连接→烧写→测试,如图 1.6 所示。

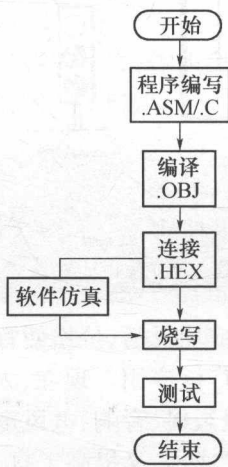


图 1.6 单片机的程序开发流程

首先在开发环境下编写源程序。注意,使用汇编语言编写的程序文件名后缀(即扩展名)是 .ASM,如果用 C 语言编写源程序后缀是 .C。然后编译、连接生成后缀为 .HEX 的可执行文件,最后将这个可执行文件烧写到单片机内。在写入单片机之前还可以进行软件仿真,即在开发环境模拟单片机程序运行情况,以便进行调试和修改。