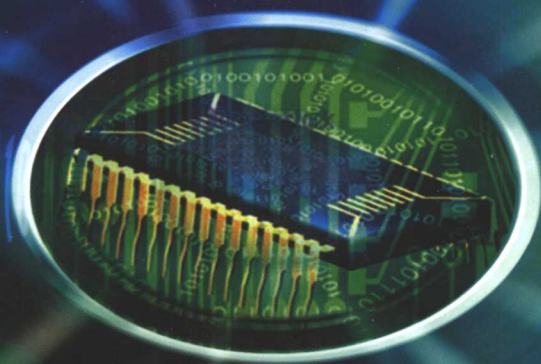


高等院校电子技术系列教材



朱善君 孙新亚 吉吟东 编著

单片机接口技术与应用



清华大学出版社

高等院校电子技术系列教材

单片机接口技术与应用

朱善君 孙新亚 吉吟东 编著

清华大学出版社

北 京

内 容 简 介

单片机已经是一项十分成熟的技术,因其具有小型化、廉价、简单可靠和稳定性好等优点,在工业测控、电子仪器、汽车电子系统、日用家电、电子通信、办公自动化设备和尖端武器(如雷达、火炮的控制系统)等方面都有广泛的应用。

本书通过介绍 MCS-51 单片机的系统结构和实例,帮助读者了解单片机开发和应用的基本知识。全书共分 16 章。前 12 章介绍了单片机的基本知识;包括学习开发单片机所需要的设备及其简单的使用方法, MCS-51 的内部结构和指令系统,单片机的开发语言, I/O 接口方法,单片机与存储器的接口,单片机的中断技术,单片机内部的定时/计数器设计,串行通信技术,数模与模数转换接口,数字显示与键盘接口技术,打印机接口。第 13 章到第 16 章为单片机系统开发方法和开发项目举例,以及单片机具体应用系统的介绍。

本书除了可作为大学生学习单片机基础知识的教材之外,还可以供相关专业的研究生、工程技术人员参考。对于从事其他计算机技术的科技人员也有参考价值。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

单片机接口技术与应用/朱善君,孙新亚,吉吟东编著. —北京:清华大学出版社,2005.3
(高等院校电子技术系列教材)
ISBN 7-302-10176-0

I.单… II.①朱…②孙…③吉… III.单片微型计算机—接口 IV.TP368.147

中国版本图书馆 CIP 数据核字(2004)第 137607 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:王景先

文稿编辑:宋延清

封面设计:陈刘源

排版人员:王 婷

印 刷 者:北京市清华园胶印厂

装 订 者:三河市新茂装订有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:27.5 字数:655 千字

版 次:2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷

书 号:ISBN 7-302-10176-0/TP·6957

印 数:1~5000

定 价:37.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

单片机是将 CPU、内存以及 I/O 等部件制作在同一芯片上的微型计算机,它仅需要少量的外围电路即可独立工作。MCS-51 是 Intel 公司于 1981 年以 8048 为基础,向上延伸所开发出来的单片机。由于其功能强大、价格低廉,并且可以使用汇编语言或 C 语言开发程序软件,所以在工业控制中得到了广泛应用。

单片机目前已经形成很多种类,集成的外围电路更是多种多样,例如 A/D、D/A 转换器, I²C, CAN 总线接口等。系统时钟不断加快,有些单片机甚至可以达到 40MHz,而其价格却很低廉。这就使其特色更加明显,可以适用于不同的工作场合。此外用于 MCS-51 系列单片机的开发工具也非常多,可使用汇编语言或者 C 语言。目前,实时操作系统的概念在嵌入式系统中也是非常流行的,UC-OS2 已经成功地被移植到 MCS-51 系列单片机上,而 Keil 公司在其生产的最常用的 51 系列单片机开发工具 Keil C51 中集成了 RTX-51 的模块,这些都是具有工程实用性的成熟的实时操作系统,可以极大地缩短工程的开发周期。

本书的目的在于通过介绍 MCS-51 的系统结构和实例,帮助读者了解单片机开发和应用的基本知识。为了简洁明了,书中的所有实验电路和程序范例均以 89C51 为核心。我们结合自己多年的工作和实践,参考和精选了一些原理性的文章和书籍,并对一些比较好的例子进行了必要的加工和修改。希望我们的编著能够对读者学习单片机技术有良好的帮助。

全书分为 16 章。第 1 章介绍 MCS-51 单片机的发展;第 2 章介绍学习开发单片机所需要的设备及如何简单地使用这些设备;第 3 章介绍 MCS-51 的内部结构和指令系统;第 4 章介绍单片机的开发语言——汇编语言及 C 语言;第 5 章为基础 I/O 接口方法概述,包括串行和并行接口的原理;第 6 章介绍单片机与程序存储器和数据存储器的接口;第 7 章介绍单片机的中断技术;第 8 章介绍单片机内部的计时/计数器设计及各种模式;第 9 章介绍串行通信技术;第 10 章介绍数模与模数转换接口;第 11 章介绍数字显示与键盘接口技术;第 12 章介绍输出打印机接口;第 13 章为单片机系统开发方法及开发项目举例,第 14、15、16 章为应用性程序的介绍。

各章节的安排虽然按一定的顺序,但学习时可以根据工作中研究和开发的需要,重新调整章节进行学习。

作者在编写本书的过程中得到很多同事、同行和老师的帮助,尤其是得到了多位老师和研究生在验证程序和编排上的帮助;其中金文光、钱利民、吴枫、邓宏一、廖东南等为本书的应用部分(即第 13~16 章)做了大量的工作。在此表示诚挚的谢意,此外还要感谢清华大学出版社第三事业部王景先先生不辞辛劳协助打字排版及校对,才使本书顺利出版。

本书如有疏漏和不周之处,希望读者不吝指正。

朱善君 孙新亚 吉吟东
于清华大学

目 录

第1章 单片机发展与应用概述	1	3.1.9 省电模式	41
1.1 单片机简介	1	3.2 指令介绍	42
1.2 单片机的种类	1	3.2.1 寻址模式	42
1.3 MCS-51 系列单片机	2	(Addressing Mode)	42
1.4 单片机学习	3	3.2.2 8051 指令集	47
1.5 完成一个简单程序的开发	4	3.3 指令汇总表	57
第2章 单片机的开发环境和工具	7	3.3.1 算术操作类指令	57
2.1 仿真器——Medwin 仿真器	7	3.3.2 逻辑运算类指令	58
2.1.1 仿真器介绍	7	3.3.3 数据传送类指令	59
2.1.2 Medwin(万利)仿真器	8	3.3.4 布尔变量操作类指令	60
使用入门	8	3.3.5 控制转移类指令	60
2.1.3 Medwin 调试实例	9	第4章 8051 的编程语言	62
2.2 编程器	17	4.1 单片机编程语言概述	62
2.2.1 主要特点	19	4.2 汇编语言	63
2.2.2 可编程、测试九大类器件	19	4.2.1 ORG 伪指令(Origin)	63
2.2.3 硬件结构	19	4.2.2 DB 伪指令(Define Byte)	63
2.2.4 WH-500 的安装与	20	4.2.3 DW 伪指令(Define Word)	64
使用说明	20	4.2.4 EQU 或=伪指令(Equal)	64
2.2.5 WH-500 软件使用范例	23	4.2.5 DATA 伪指令(Data)	64
第3章 MCS-51 单片机系统结构和	26	4.2.6 XDATA 伪指令	64
指令系统	26	(External Data)	64
3.1 单片机系统结构	26	4.2.7 BIT 伪指令	64
3.1.1 单片机简介	26	4.2.8 END 伪指令	65
3.1.2 MCS-51 系列简介	27	4.3 C51 语言	65
3.1.3 MCS-51 单片机引脚功能	28	4.3.1 C51 的程序结构	65
3.1.4 单片机的复位动作	30	4.3.2 数据类型和长度	67
3.1.5 MCS-51 的时钟输入与	31	4.3.3 C51 语言的运算符	68
机器周期	31	4.3.4 C51 程序流程控制语句	69
3.1.6 MCS-51 内部结构	33	4.3.5 函数(FUNCTION)	72
3.1.7 特殊功能寄存器	36	4.3.6 宏定义 #define	75
3.1.8 输出/输入端口	40	4.3.7 条件编译	75

第5章 微型计算机的输入/输出 76

5.1 I/O 接口电路概述 76

5.1.1 I/O 接口电路的作用 76

5.1.2 接口与端口的差别 78

5.1.3 外设的编址方式 78

5.1.4 将外设当作数据存储器
访问 81

5.2 输入/输出传送方式 82

5.2.1 无条件传送 82

5.2.2 查询式传送 84

5.2.3 中断传送 86

5.2.4 直接存储器存取(DMA) 87

第6章 存储器与微处理机的接口 89

6.1 存储器技术的发展及其趋势 89

6.1.1 概述 89

6.1.2 存储器发展趋势 91

6.1.3 正在发展中的存储新技术
和特殊存储器 93

6.1.4 结语 97

6.2 常用存储器型号 98

6.2.1 易失性存储器 98

6.2.2 非易失性存储器 99

6.2.3 多端口读写存储器
MPRAM 1026.2.4 内嵌电池掉电自保护
存储器插座 102

6.3 MCS-51 单片机与存储器

存取数据的时序 103

6.3.1 机器周期和指令周期 103

6.3.2 MCS-51 的取指令/执行
时序 1036.3.3 访问外部 ROM 和外部
RAM 的时序 1046.3.4 MCS-51 单片机外部
存储器的扩展 106

6.4 程序存储器的扩展 107

6.5 数据存储器的扩展 108

6.5.1 6116 静态 RAM 108

6.5.2 单片机和 FLASH 存储器

的连接 109

第7章 基本 I/O 接口和中断 112

7.1 基本输出 112

7.1.1 电路说明 113

7.1.2 LED 显示程序 113

7.1.3 程序说明 113

7.2 七段显示器静态显示方法 114

7.2.1 显示码设计 115

7.2.2 电路说明 116

7.2.3 七段 LED 数码显示器的
显示程序 116

7.2.4 程序说明 117

7.3 七段显示器扫描式显示方法 117

7.3.1 电路说明 118

7.3.2 七段显示器扫描式
显示程序 118

7.3.3 程序说明 119

7.4 基本输入 120

7.4.1 电路一 121

7.4.2 基本开关输入和 LED 发光
二极管显示程序 121

7.4.3 程序说明 122

7.4.4 电路二 122

7.4.5 基本开关输入和七段
数码管显示程序 122

7.4.6 程序说明 123

7.5 键盘扫描与七段显示器 123

7.5.1 电路说明 124

7.5.2 键扫描及相应键值的显示
程序 125

7.5.3 程序说明 127

7.6 输入/输出口的扩展 127

7.6.1 用 8243 输入/输出
扩展器扩展 I/O 口 1287.6.2 用 8255A 可编程并行
接口芯片扩展 I/O 口 1317.6.3 用 8155 通用接口
芯片扩展 I/O 口 137

7.6.4 用串行口扩展并行 I/O 口.....	142	8.6 定时器 0 在模式 2 的实习.....	168
7.7 外部中断	144	8.6.1 定时器 0 在模式 2	
7.7.1 8051 中断结构.....	144	实习程序.....	168
7.7.2 8051 中断规划.....	147	8.6.2 程序说明.....	169
7.7.3 外部中断软件仿真.....	149	8.6.3 程序执行结果.....	170
7.7.4 INT0 中断实习	151	8.7 定时器 0 在模式 3 的实习.....	170
7.7.5 中断优先权实习.....	154	8.7.1 定时器 0 在模式 3	
第 8 章 定时/计数器	157	实习程序.....	170
8.1 定时/计数器原理	157	8.7.2 程序说明.....	171
8.1.1 定时/计数器模式控制		8.7.3 程序执行结果.....	172
寄存器 TMOD.....	157	第 9 章 串行端口通信	173
8.1.2 定时/计数器控制		9.1 MCS-51 串行端口中断	173
寄存器 TCON.....	158	9.1.1 串行端口的模式 0.....	174
8.1.3 模式 0: 13 位		9.1.2 串行端口的模式 1.....	176
定时/计数器	159	9.1.3 串行端口的模式 2.....	177
8.1.4 模式 1: 16 位		9.1.4 串行端口的模式 3.....	179
定时/计数器	160	9.1.5 串行端口的波特率规划	179
8.1.5 模式 2: 具有重新加载的		9.1.6 多处理器的通信.....	181
8 位定时/计数器	160	9.2 PC 机中的 RS232 适配卡介绍	182
8.1.6 模式 3: 定时/计数器 0		9.3 8051 机间通信的 C 编程	185
成为两组独立 8 位		9.3.1 点对点的串行异步通信	185
定时/计数器	160	9.3.2 多机通信.....	188
8.2 定时/计数器规划	161	第 10 章 单片机与数模及模数转换	
8.3 定时/计数器软件仿真	162	的接口	198
8.3.1 定时/计数器软件		10.1 D/A 转换器原理.....	198
仿真程序	162	10.2 MCS-51 单片机与 D/A	
8.3.2 程序说明.....	163	转换器的接口.....	201
8.3.3 程序结果与问题讨论.....	164	10.2.1 8 位 D/A 芯片 DAC0832... ..	201
8.4 定时器 0 在模式 0 的实习.....	164	10.2.2 12 位 D/A 芯片 AD7521	204
8.4.1 定时器 0 在模式 0		10.3 A/D 转换器原理.....	206
实习程序	164	10.3.1 逐次比较型 A/D 转换器	206
8.4.2 程序说明.....	165	10.3.2 双积分型 A/D 转换器	209
8.4.3 程序执行结果.....	165	10.4 MCS-51 单片机与 A/D	
8.5 定时器 0 在模式 1 的实习.....	166	转换器接口.....	211
8.5.1 电路图	166	10.4.1 8 位 A/D 芯片	
8.5.2 定时器 0 在模式 1		ADC0809	211
实习程序	167	10.4.2 12 位 A/D 芯片 AD574	214
8.5.3 程序说明.....	167	10.4.3 16 位 A/D 芯片 AD7705	217
8.5.4 程序执行结果.....	168		

第 11 章 数字显示接口	225	14.3.4 程序说明	302
11.1 数字显示	225	14.4 延时程序	302
11.1.1 段式显示	225	14.4.1 函数目的	302
11.1.2 字符型显示器 SMC2004A LCM 显示模块	228	14.4.2 参数说明	302
11.1.3 图形显示器	237	14.4.3 延时程序	302
11.2 T6963C 应用实例	257	14.4.4 程序说明	303
第 12 章 打印机接口	261	14.5 定时器的应用	303
12.1 TPμP-TF 打印机	261	14.5.1 函数目的	303
12.1.1 TPμP 打印机的 接口信号	261	14.5.2 定时器应用程序	304
12.1.2 字符代码和汉字	261	14.5.3 程序说明	304
12.1.3 打印命令	263	14.6 蜂鸣器的应用	305
12.1.4 TPμP-TF 和 8051 的 接口和编程	265	14.6.1 电路图	305
12.2 TPμP40A 打印机	267	14.6.2 电路说明	305
12.3 TPμP16A 打印机	272	14.6.3 蜂鸣器应用程序	306
第 13 章 单片机系统开发方法	274	14.6.4 程序说明	306
13.1 单片机应用系统的研制过程	274	14.7 扩充输出端口的应用	306
13.2 单片机开发时的几项常用 的关键技术	277	14.7.1 电路图	307
13.2.1 单片机系统加密技术	277	14.7.2 电路说明	307
13.2.2 单片机的可靠性技术	281	14.7.3 参数说明	307
第 14 章 典型应用	298	14.7.4 扩充输出端口应用程序	307
14.1 基本输入输出	298	14.7.5 程序说明	309
14.1.1 电路图	298	14.8 I ² C 串行总线原理和应用	310
14.1.2 电路说明	299	14.8.1 I ² C 总线硬件结构	310
14.1.3 基本输入输出程序	299	14.8.2 I ² C 总线的数据传送	311
14.1.4 程序说明	299	14.8.3 时钟同步和总线仲裁	316
14.2 中断的应用	299	14.8.4 I ² C 总线电特性 技术说明	316
14.2.1 电路图	300	14.8.5 I ² C 总线的时序	318
14.2.2 电路说明	300	14.8.6 I ² C 总线接口器件的 应用软件设计	319
14.2.3 中断应用程序	300	第 15 章 用 CAN 实现系统通信	331
14.2.4 程序说明	301	15.1 CAN 总线协议 2.0A 的概述	331
14.3 公用函数	301	15.1.1 介绍	331
14.3.1 函数目的	301	15.1.2 基本概念	332
14.3.2 参数说明	301	15.1.3 报文传输	334
14.3.3 公用函数应用程序	302	15.1.4 报文校验	340
		15.1.5 编码	340
		15.1.6 错误处理	340
		15.1.7 故障界定	341

15.1.8	位定时要求.....	342	16.6.7	功能代码 08.....	371
15.1.9	同步	343	16.6.8	功能代码 16.....	372
15.2	CAN 接口器件 SJA1000	344	16.6.9	功能代码 125.....	373
15.2.1	SJA1000 的逻辑功能	345	16.6.10	功能代码 126.....	374
15.2.2	SJA1000 的地址分配和 复位值	346	16.6.11	功能代码 127.....	375
15.2.3	SJA1000 的寄存器功能	347	16.7	异常应答.....	376
15.2.4	SJA1000 与单片机的 连接	351	16.8	存储器映像说明.....	377
15.2.5	驱动程序实例.....	352	16.8.1	开关量输出映像	377
第 16 章	QBCOD-A 型装置通信		16.8.2	开关量输入映像	378
	软件开发	355	16.8.3	测量电量和显示量 的映像.....	378
16.1	概述	355	16.8.4	最新测定记录映像	379
16.2	控制器结构	356	16.8.5	系统运行状态的映像	379
16.3	控制器功能	357	16.8.6	日期与时钟的映像	380
16.3.1	ECPU 板	357	16.8.7	系统信息映像	381
16.3.2	ELCD 板	357	16.8.8	通讯参数映像	382
16.3.3	系统功能模块.....	358	16.8.9	电量校正参数映像	382
16.4	QBCOD-A 型装置通信概述	359	16.8.10	测定参数映像	383
16.4.1	Modbus 协议.....	359	16.8.11	输入参数配置映像	384
16.4.2	两种传输方式.....	361	16.8.12	标定参数映像	387
16.4.3	Modbus 消息帧.....	362	16.8.13	自动测定运行设备 工作参数映像	387
16.4.4	错误检测方法.....	365	16.8.14	系统异常记录映像	388
16.5	数据包格式	366	16.8.15	操作记录映像	389
16.6	数据包定义	367	16.8.16	报警记录映像	390
16.6.1	功能代码 01.....	367	16.9	通讯程序流程图.....	391
16.6.2	功能代码 02.....	368	16.10	通讯程序源代码.....	392
16.6.3	功能代码 03 和 04.....	368	16.10.1	LCD 板通讯程序	392
16.6.4	功能代码 05.....	369	16.10.2	CPU 板通讯程序	405
16.6.5	功能代码 06.....	370			
16.6.6	功能代码 07.....	370			

第 1 章 单片机发展与应用概述

1.1 单片机简介

单片机实际上是微型计算机的一种,自从它问世以来,人们对它不断地改进,应用在现代化社会的各个方面。单片机体积小,价格低廉,开发较为容易,可根据需要制作成各种智能控制器以代替人工操作,实现自动化。

单片机实际上和一般个人计算机(PC)类似,它由这样几个部分构成:CPU(进行运算和控制)、RAM(进行数据存储)、ROM(进行程序存储)、输入/输出设备(例如串行口、并行口等)。在个人计算机上,这些部分被分成若干块芯片,安装在一个称之为主板的印刷线路板上。而在单片机中,这些部分(甚至还有 A/D 和 D/A 等),全部被集成到一块电路芯片中了,所以就称为单片机。

单片机由于集成度高、功能相对较少、生产量很大,技术也很成熟,所以价格并不高,仅几元到几十元人民币。单片机体积不大,一般用 40 脚封装,当然功能多一些的单片机也有引脚比较多的,如 68 引脚,功能少的只有 20 多个或 10 多个引脚,甚至 8 个引脚。

单片机也称为微控制器 MCU,但实际上并不是每种单片机都是真正的“单片”,而是需要外接扩展的程序存储器等才能工作,程序存储器一般用 EPROM(紫外线可擦除只读存储器),也有用 E²PROM 的,还有使用 FLASH 的。

通常见到的 8031 单片机(MCS-51 中的一种)就需要外接 EPROM 和 RAM,它的特点是系统设计灵活,扩展性好,但总体成本较高,印刷电路板设计也较复杂。而 51 系列则已经把存储器集成在芯片之中了。此外还有 8096/98 系列,速度更快,功能更强,集成的内容更多(96 系列为 16 位单片机)。

1.2 单片机的种类

单片机发展到今天,已经有许多型号。常见的单片机有 Intel 公司的 MCS-51、151、251 系列;ZILOG 的 Z8 系列;Philips 公司的 80C51 和 51XA 系列;Motorola 的 68HC05/08 系列;Microchip 公司的 PIC16CXX 系列单片机;Atmel 的 89C51 系列;Winbond 的 78E51 系列等。

较早进入我国的是 Intel 的 MCS-51 型单片机,它在国内的应用也是最普遍的。现在好多单片机都是兼容 51 单片机的,可见它的影响。

单片机型号成百上千,但是其基本结构和原理都是差不多的。所以,掌握了一种单片

机的硬件结构、指令系统和系统设计,就为进一步学习和掌握其他各种单片机打下了牢固的基础。

1.3 MCS-51 系列单片机

在计算机领域,系列机是指同一厂家生产的具有相同系统结构的机器。20 世纪 80 年代中期以后,Intel 以专利转让的形式把 8051 内核给了许多半导体厂家,如 Amtel、Philips、Ananog devices、Dallas 等。这些厂家生产的芯片是 MCS-51 系列的兼容产品,准确地说是与 MCS-51 指令系统兼容的单片机。这些单片机与 8051 的系统结构(主要是指令系统)相同,采用 CMOS 工艺,因而常用 80C51 系列来称呼所有具有 8051 指令系统的单片机。他们一般都对 8051 作了一些扩充,功能更强、市场竞争力更强。因而不应该把他们称为 MCS-51 系列单片机。MCS 只是 Intel 公司专用的。

MCS-51 系列的三个基本产品是 8051(8031)、8751、89C51。它们都是一个系列,引脚和指令完全兼容,核心是一个 8 位的 CPU,但内部结构有所不同。

8031 除 CPU 外,还包括 128 字节的 RAM,21 个特殊功能寄存器(SFR)、4 个 8 位并行输入输出、1 个全双工串行口,2 个 16 位的定时器/计数器,但内部没有程序存储器,需要在芯片外扩展 EPROM 芯片。8051 包含了 8031,芯片内又集成了 4 K 字节的 ROM,用作程序存储器,并且 8051 是掩膜型单片机,不超过 4 K 字节的程序是厂家制作芯片时,代用户烧制的,用户不能修改。所以 8051 在国内很少见到。

与 8051 功能相同的是 8751,但是它的 4 K 字节的程序存储器采用了 EPROM,用户可以用紫外线擦除其中的程序内容,反复修改程序。但是 8751 价格较贵,甚至比 8031 外加一片 4 K 字节的 EPROM 还贵,而且经常擦写的 EPROM 也较易损坏,8751 的 EPROM 一旦损坏,整个片子就废了。

Intel 还在 MCS-51 系列三种基本型产品的基础上,推出了各类增强型产品,主要有 8032/8052/8752:它们的内部 RAM 增到 256 字节,8752/8052 片内的程序存储器容量增到 8K 字节,定时器/计数器增至 3 个 16 位计数器,有 6 个中断源。另外还有低功耗的 CHMOS 工艺芯片的 80C31/87C51/80C51:有两种掉电工作方式,其一是让 CPU 停止工作,其他部分仍继续工作;其二是除片内 RAM 继续保持数据外,其他部分都停止工作。

高性能的 8XC52 系列是以 8052 为基础,采用 CMOS 工艺,并将 MCS-96 系列中的高速输入/输出、A/D 转换器、脉冲宽度调制、看门狗定时器等移植进来构成新一代 MCS-51 产品,PHILIPS 公司生产的 8XC552 系列即为此类产品。

以 MCS-51 为基础,人们根据各种不同功能的需要,又开发了各种型号的单片机。首先是低功耗、高性能的 89C51。AT89C 系列和 51 完全兼容,但内含 4 K 字节的快擦写可编程/擦除只读存储器,可以电擦写 1000 次以上。AT89C 系列单片机价格也便宜,使用也方便,是真正意义上的单片机。它的时钟频率高达 20 MHz,芯片上的 E²PROM 允许在线(+5V)电擦除、电写入,此外还支持软件选择的两种掉电工作方式,非常适用于电池供电的场合。

AT89C 系列单片机主要有四种:AT89C51、AT89C52、AT89C1051、AT89C2051,工作电压范围 2.7 V~6 V。其中 AT89C51 和 AT89C52 都是 40 脚封装的,编程电压为 12 V~5 V,

可编程 I/O 口都是 32 个, 只是 E²PROM 分别为 4 K 和 8 K, 内部 RAM 分别为 128 字节和 256 字节, 16 位定时/计数器分别为 2、3 个, 中断源分别为 5、8 个; AT89C1051 和 AT89C2051 都是 20 脚封装的, 编程电压为 5 V, 可编程 I/O 口都是 15 个, 内部 RAM 都是 128 字节, E²PROM 分别为 1 K 和 2 K, 16 位定时/计数器为 2 个, 中断源分别为 5、3 个。

下面列举一下 51 系列单片机的主要生产厂和型号:

AMD:	8751H, 87C51, 87C521, 87C52T2, 87C541
Atmel:	AT80F51, AT80F52, AT87F51, AT87F51RC, AT87F52, AT87F55, AT87F- 55WD, AT89C1051, AT89C2051, AT89C4051, AT89C51, AT89C51RC, AT89C52, AT89C55, AT89C55WD, AT89LS53, AT89LS8251, AT89LV51, AT89LV52, AT89LV55, AT89S53, AT89S8252, AT90LS2323, AT90LS2343, AT90S1200……
Intel:	8751/H, 8751BH, 87C51, 87C51FA, 87C51FB, 87C52, 87C52BH……
Philips:	87C51, 87C52, 87C54, 87C652, P87LPC762, P89C51RA+, P89C52U……
Signetics:	87C51, 87C51FA-C, 87C52, 87C54, ……
Winbond:	W77E468, W77E58, W78E51, W78E52, W78E54, W78E58, W78LE51, W78LE52, W77LE58, W77LE516, W77LE516B, W77LE812(该芯片系列 ROM 比较大)

其中 AT89C2051 就是这几年来在我国非常流行的单片机, 它是由美国 Atmel 公司开发生产的。这种单片机的程序存储器是电可擦的编程存储器, 所以在开发阶段便于修改。以后我们可用 AT89C2051 来完成一系列的实验。

1.4 单片机学习

由于 MCS-51 单片机是其他很多单片机的基础, 加上教材和资料较多, 有利于学习。所以本书将以 51 系列单片机作为学习的对象。

本书教学方法具有如下特点:

- (1) 以读者的认知规律为主线, 而不是以课程结构为主线。
- (2) 以任务为单元构建认知单元, 而不是以单片机功能为单元构建。
- (3) 完成第一个任务即可进行单片机应用的初步尝试, 不必学完单片机的全部知识体系。随着任务的逐渐进行, 知识逐渐完善, 能力逐渐提高, 所有任务完成时, 已初步具有了开发能力。

一般初学者的学习环境是: 在硬件上一般只需要一个编程器、一个最小系统板, 和一个软件仿真器。如果经济上允许, 买一个硬件仿真器当然就更好了。

编程器就是可以把你的程序写到芯片上的设备, 一般不能少。可以到有关网页上看看, 很容易购买到价钱便宜的编程器。

最小系统板可以自己做。如果为了修改方便, 可以用市场上买得到的“面包板”进行

元件接插连接。

仿真器就是可以硬件仿真的设备，不过价钱相对贵一些。初学者也可以不要，你可以先在软件仿真器(网上可以下载一个单片机软件开发环境，目前 Keil 公司已经授权，广州周立功公司，提供免费的 Keil C51 评估版)上调试通过了，再通过编程器写到单片机上，插到系统板上调试就行。这样也减少了许多用仿真器带来的意想不到的问题。当然，学习单片机最好用带有 FLASH 或 E²PROM 电可擦程序存储器的单片机，比如 89C2051。这样就可以反复编程反复使用了。

单片机的软件开发程序，就是用来开发在单片机上执行的程序的开发环境。使用的开发语言主要有汇编语言和 C 语言。他们各有特点，汇编语言效率高，但是可读性、可移植性，以及可维护性不好；C 语言效率较高，可读性、可移植性，以及可维护性较好。以前由于系统资源有限(主要是受 RAM、ROM 大小的限制)，汇编语言使用的比较多。而现在，由于 C 语言编译效率的提高，而且随着大容量 RAM、ROM 成本的降低，C 语言的优势更加明显，已经可以满足大多数工业场合的需要；目前较为流行的开发环境有 KEIL C51。对于刚刚学习 MCS-51 单片机的读者来说，使用汇编语言可以让自己更加贴近硬件，可以更好地理解 MCS-51 的指令和使用；所以本书的例子中，一部分采用 C 语言，一部分采用汇编语言。

1.5 完成一个简单程序的开发

作为一个初学者，都希望自己尽快地进入角色。也就是说，能尽快地通过单片机做一点有用的工作。为了这个目的，我们给出下面这个例子。初学者现在不一定要十分清楚程序指令的格式和来源，因为今后章节里还会详细地介绍。在这里仅仅是为了了解如何开发单片机的程序。

这个例子是要我们用单片机做一个信号灯，要让灯不断地闪烁。实际上就是要灯亮一段时间，再熄灭一段时间。怎样实现这个要求呢？

首先我们要在面包板上按照图 1.1 接好电路。如果我们能使 P1.0 脚为高电平，发光二极管 LED 就不亮，而 P1.0 脚是低电平时，LED 就会发亮。因此我们要是能够让 P1.0 引脚电平按要求一会儿变高一会儿变低，就可以让灯不断地闪烁。

那么我们又怎样让单片机的 P1.0 引脚电平变“高”或变“低”呢？

让人做事，说一声就可以，即发布命令。要让计算机做事，也得向计算机发布命令，计算机能听得懂的命令称之为计算机指令。让单片机一个引脚输出高电平的指令是 SETB，让一个引脚输出低电平的指令是 CLR。因此，我们要 P1.0 输出高电平，只要写 SETB P1.0 即可；要 P1.0 输出低电平，只要写 CLR P1.0 即可。但如何让单片机引脚“一会儿”变高“一会儿”变低(也就是让单片机延时一段时间)呢？我们可以借助于对单片机另外几个指令即 CALL、MOV、DJNZ、RET 等指令的编排来实现。这种“编排”就叫编写程序。

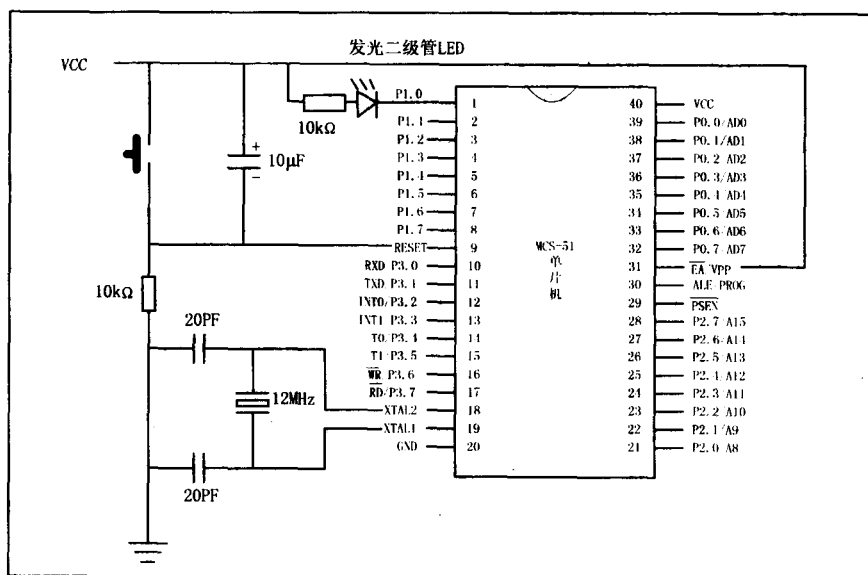


图 1.1 实验接线图

以下先给出程序(程序中分号后面的注释内容是为了便于讲解而写的,实际应用中,编译器并不要求,即它们与程序的运行无关)。

主程序:

```
Start:      SETB P1.0      ; (1) 是让灯灭
            LCALL DELAY    ; (2) 是延时, 调用延时子程序指令
            CLR P1.0       ; (3) 是让灯亮
            LCALL DELAY    ; (4) 和第二条一模一样, 也是调用延时子程序指令
            LJMP Start     ; (5) 跳转回去执行第一条指令
```

以下为延时子程序:

```
DELAY:      MOV R7, #250   ; (6) 让 R7 作为计数器, 并装入一个常数
D1:          MOV R6, #250   ; (7) 让 R6 作为计数器, 并装入一个常数
D2:          DJNZ R6, D2     ; (8) 让 R6 减 1, 并判断 R6 是否为 0。若 R6 为 0, 向
                           ; 下执行指令。若 R6 不为 0, 跳转 D2, 继续让 R6 减 1。
            DJNZ R7, D1     ; (9) 让 R7 减 1, 并判断 R7 是否为 0。若 R7 为 0, 向
                           ; 下执行指令。若 R7 不为 0, 跳转 D1, 使 R6 完成新一轮
                           ; 计数。
            RET             ; (10) 返回指令
END          ; (11) 程序结束
```

下面分析一下前面的各条指令。第1条是让灯灭, 第2条是延时, 第3条是让灯亮, 第4条和第2条一模一样, 也是延时, 第5条应当是转去执行第1条指令。第2和第4条实现的原理稍后谈, 先看第5条, LJMP指令的意思是转移。往什么地方转移呢? 后面跟的是Start, 看一下, 什么地方还有Start? 对了, 在第1条指令的前面有一个Start。所以, 很直观地, 我们可以认识到, 它要转到第1条指令处。这个第1条指令前面的Start被称之为标号, 它的用途就是给这一行起一个名字, 便于使用。是否一定要给它起名叫Start呢? 当然不是, 起什么名字, 完全由编程序的人决定, 可以称它为A, X等等, 当然, 这时,

第5条指令 LJMP 后面的名字也得跟着改了。

第2条和第4条指令的用途是延时，它是怎样实现的呢？指令的形式是 LCALL，这条指令称为调用子程序指令，看一下指令后面跟的是什麼，DELAY，找一下 DELAY，在第6条指令的前面。显然，这也是一个标号。这条指令的作用是这样的：当执行 LCALL 指令时，程序就转到 LCALL 后面的标号所标定的程序处执行，当执行指令的过程中遇到 RET 指令时，则程序就返回到 LCALL 指令的下面一条指令继续执行，从第6行开始的指令中，可以看到确实有 RET 指令。所以整个程序在执行第2条指令后，将转去执行第6条指令，而在执行完6、7、8、9条指令后将遇到第10条指令：RET，执行该条指令后，程序将回来执行第3条指令，将 P1.0 清零，使灯亮，然后又执行第4条指令，执行第4条指令就是转去执行第6、7、8、9、10条用于延时的指令，然后回来执行第5条指令。第5条指令就是让程序回到第1条开始执行，如此周而复始，灯就在不断地交替亮、灭了。

从标号 DELAY 标志的这一行到 RET 这一行中的所有程序，是一段延时程序，大概延时零点几秒，至于具体的时间，以后我们在学习指令时，再讲如何计算。程序的最后一行是 END，这不是一条指令，它只是告诉我们程序到此结束。

现在从概念上我们已经有办法让单片机去完成任务了。但是我们怎样才能让单片机实际执行这些指令呢？要解决这个问题，还得有几步要走。

第一步，由于计算机看不懂 SETB、CLR 之类的指令，我们得把指令翻译成计算机能懂的方式，再让计算机去读。计算机能懂什麼呢？它只懂一样东西——1、0、1、0 这类二进制数字。因此我们得把 SETB P1.0 变为 11010010，10010000(D2H, 90H)，把 CLR P1.0 变为 11000010，10010000(C2H, 90H)……，至于为什么是这些数字而不是别的数字，这是由 51 芯片的设计者——Intel 规定的机器指令系统，我们不必去研究。而且这个翻译工作也可以由“软件仿真器”中的编译(翻译)软件完成。具体如何操作见第2章。

第二步，在得到上述单片机机器指令后，怎样使这样的两个数字进入单片机的内部呢？这就要借助于一个硬件工具“编程器”。详细使用方法见第2章。

第三步，当把该程序翻译后的机器指令写入单片机之后，再把单片机插入前面讲的接好线的面包板上，给上电源，发光二极管 LED 就会不断地闪烁了。

本试验所采用单片机的具体型号建议为 89C51。

第2章 单片机的开发环境和工具

学习单片机重在实践、操作。所以初学者一定要花一些钱给自己创造一个学习单片机的实验环境(实验设备)。一般说来,在你决定学习单片机之前,请做好如下准备工作。

硬件准备: 编程器一个、89C51 若干片、相应的电子元件(电阻、电容、晶振、电源)、实验板(面包板)一块、其他常规工具。

另外个人计算机是必须配备的,因为编译程序和“烧片子”都需要它。如果你想学单片机,而又不愿做这些投资,则可能很难进行学习,或事倍功半。好在这些投资并不算多。

编程器: 商品化的一般几百到上千元。但现在网上有好多网站都提供编程器的资料,如果你身边有高手,不妨请他们帮助你做一个,不过几十元的代价。

仿真器: 商品化的一般为一千到几千元。对于初学者来说,一般所编写的程序不太大,可以暂时不购买这么贵的设备。从网络上可以下载一些仿真器软件,比如我们在后面介绍的 Medwin(万利)仿真器。

实验板: 我们建议初学者去电子市场购买一块方便接插的面包板。在单片机管脚上接一些发光二极管 LED, 一些按钮, 数码管等。接线电路图见第1章的图1.1。

2.1 仿真器——Medwin 仿真器

仿真器是单片机开发过程中必不可少的重要工具,下面我们就以 Medwin(万利)仿真器为例,对仿真器的特点及其使用方法做一个简单说明。

2.1.1 仿真器介绍

仿真器一般包括安装在个人 PC 机上的仿真器软件和一个与 PC 机进行串行通信(也有的通过并口进行连接)的硬件仿真器(为了适应不同型号的单片机常常还带有多个仿真适配头)。Medwin 已经集成软件开发环境,可以支持用汇编语言和 C 语言进行开发。

仿真器的实际用途与一般编译器没有太大差异,即都是用来编写、编译和调试应用程序的。像“仿真器”、“编程器”这样的习惯称呼足以令任何初学者产生一定的误会。所以读者需要通过本章内容澄清一些认识。

由于仿真器比较贵,初学者可以先在网上下载仿真器软件。通过软件仿真器,你就可以进行一些比较简单的程序开发,包括编辑、编译/汇编、调试、输出 Intel HEX 机器码文件。这个代码文件就是可以通过编程器“烧入”单片机的“目标机器码”文件。如果想开发大一些的程序,最好还是花几千元买一套包括软件、硬件、适配头的仿真器。

为篇幅所限,本书只介绍软件仿真器部分。读者可以看到,没有硬件部分,还是照样

可以完成一些开发工作。

2.1.2 Medwin(万利)仿真器使用入门

Medwin 开发环境为用户提供了以下两种方式开发应用程序：

- 不使用 Medwin 集成开发环境项目管理方式——对源程序文件直接进行汇编/连接方式，兼容传统开发习惯。
- 使用 Medwin 集成开发环境项目管理方式——可进行多模块、混合语言的方式，同样也适合单模块程序的开发。

不使用 Medwin 集成开发环境项目管理方式，只能进行单模块下的应用程序开发，具有一定的局限性。为了便于读者理解和掌握，本书仅仅介绍不使用项目管理方式开发应用程序。关于使用 Medwin 集成开发环境项目管理方式，读者可以参考仿真器的使用说明书。

接下来，我们将对 Medwin(万利)仿真器的基本菜单命令和操作做简要说明。

2.1.2.1 关闭当前项目文件

命令：“项目管理”→“关闭当前项目”

不使用 Medwin 集成开发环境项目管理方式开发应用程序，用户必须关闭已经打开的项目，此时 Medwin 集成开发环境关闭界面上的所有窗口。因为当打开项目文件后，Medwin 集成开发环境默认所有编译/汇编、产生代码的过程都是对项目或项目所包含的文件进行的。

2.1.2.2 在文件菜单下打开应用程序

命令：“文件”→“新建”或“文件”→“打开”

单模块方式下的文件调试可以按照以下方法新建文件或打开文件：

单击“文件”→“新建”命令，输入文件名和扩展名，新建文件；

单击“文件”→“打开”命令，选择文件捡取框中的文件将其打开。

2.1.2.3 编译/汇编

命令：“项目管理”→“编译/汇编”

Medwin 集成开发环境根据文件的扩展名，自动对当前激活的文件选择调用外部编译器或汇编器：

如果当前文件的扩展名为 ASM 或系统定义的扩展名，“编译/汇编”命令调用外部汇编命令、对当前文件汇编；

如果当前文件的扩展名为 C 或系统定义的扩展名，“编译/汇编”命令调用外部 C 编译命令对当前文件编译。

执行“项目管理”→“编译/汇编”命令后，产生的结果显示在消息框中。

如果需要设置文件的“编译/汇编”命令行参数，可以选择“项目管理”→“文件属性”命令设置。