



普通高等教育“十一五”国家级规划教材 (高职高专教育)
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

DANPIANJI
YUANLI JI YINGYONG

单片机 原理及应用

(第二版)

彭同明 杨少华 郑德龙 编著
桂绍勇 郭小进



中国电力出版社

<http://jc.cepp.com.cn>

Electric Power Technology



普通高等教育“十一五”国家级规划教材 (高职高专教育)
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

DANPIANJI
YUANLI JI YINGYONG

单片机 原理及应用

(第二版)

编 著 彭同明 杨少华 郑德龙
桂绍勇 郭小进
主 审 孙 海 张国勋



中国电力出版社

<http://jc.cepp.com.cn>

Electric Power Technology

内 容 提 要

本书为普通高等教育“十一五”国家级规划教材（高职高专教育）。

全书共分为五章，主要内容包括人机接口技术、实时检测与控制、模拟量处理技术、环境温度检测系统和系统扩展技术。书后附有单片机系统设置、程序的编辑与编译指导、实验装置介绍、程序下载指导和虚拟实验操作指导。本书在教学模式上采用项目驱动，并安排了一些创新项目让学生独立完成，以提高学生的创新能力、独立开发项目能力、团队协作能力和方法能力。

本书可作为高职高专院校相关专业“单片机原理及应用”课程的教学用书，也可作为相关工程技术人员项目开发的参考书。

图书在版编目 (CIP) 数据

单片机原理及应用/彭同明等编著. —2 版. —北京:
中国电力出版社, 2009

普通高等教育“十一五”国家级规划教材. 高职高专
教育

ISBN 978-7-5083-8283-8

I. 单… II. 彭… III. 单片微型计算机-高等学校:
技术学校-教材 IV. TP368.1

中国版本图书馆 CIP 数据核字 (2008) 第 210573 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2005 年 7 月第一版

2009 年 2 月第二版 2009 年 2 月北京第五次印刷

787 毫米×1092 毫米 16 开本 12 印张 290 千字

定价 19.20 元

敬告读者

本书封面贴有防伪标签，加热后中心图案消失
本书如有印装质量问题，我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

目前,单片机越来越广泛地应用于军事设备、机电设备、控制设备、智能仪器仪表、办公设备、通信设备、家用电器、智能玩具等,使产品功能、精度和质量大幅度提高,而电路设计更简单、可靠性更高、成本更低。

随着电子技术的发展,单片机的集成度越来越高,功能越来越强,使用越来越方便,不断涌现出高性能的新型单片机。

本书是在2005年7月出版的《单片机原理及应用》教材基础之上,经过工学结合的教学改革和精品课程建设,重新调整结构和内容后编写的;在教学内容上选用AVR系列中的ATmega16作典型教学芯片,并简单介绍了C语言基本编程规则和方法,使用C语言进行应用程序开发,引入了单片机测控系统常用的接口技术和应用技术,并使用了虚拟实验技术。这主要出于如下几方面的考虑:

(1) 向学生讲授先进的单片机技术。AVR系列单片机型号多、功能强、效率高、接口丰富、接口驱动能力强、可在线下载程序、可程序加密且价格低廉,特别适合于C语言程序开发,得到了广泛的应用。

(2) 提高学习效果。采用C语言描述和编程,降低了学习难度,用虚拟实验辅助教学和实验,可提高学习效果。

(3) 提供给学习者先进的单片机程序开发方法。

(4) 把学习重点放在单片机应用上。由于ATmega16单片机的集成度高,应用接口简单,并避免了汇编语言的难度,在有限的学习时间里将有更多的时间学习其应用。

本书在教学模式上采用项目驱动,每个项目被分解为几个小的项目,由易到难,逐步深入。在教学方法上教学交替进行,讲练结合,还安排了一些创新项目让学生独立完成,以提高学生的创新能力、独立开发项目能力、团队协作能力和方法能力。

本书的所有项目程序都进行了调试,所有的应用程序都可以移植到实际的单片机应用系统中。

本书建议教学时数在80~90学时。

本书由武汉电力职业技术学院单片机应用教学团队开发编写。彭同明对全书进行了策划和统稿,编写了前言、第一章、第四章第三节和附录部分,并开发、调试了第一章和第四章第三节中的项目;杨少华编写了第二章,并开发、调试了第二章中的项目;郑德龙编写了第三章,并开发、调试了第三章中的项目;桂绍勇编写了第四章的第一节和第二节,并开发、调试了第四章第一节和第二节中的项目;郭小进编写了第五章,并开发、调试了第五章中的项目。本书得到了保定电力职业技术学院徐学勤、许红兵和武汉电力职业技术学院汤晓华、夏小鹏等老师的帮助,在此一并表示感谢。

本书由北华大学孙海、邢台职业技术学院张国勋主审,提出了许多宝贵意见,在此表示感谢。

本教材的实验使用CodeVisionAVR C集成开发软件作为开发软件,广东双龙公司网站

(www.sl.com.cn) 免费提供的 SLISP 软件作为下载软件, 可从网站下载。实验装置由武汉电力职业技术学院机电工程系提供。虚拟实验采用 PROTEUS 软件。

由于编者水平有限, 书中错误和不妥之处难免, 敬请广大读者批评指正。

编 者

2008 年 11 月

目 录

前言

第一章 人机接口技术	1
第一节 开关控制指示灯	1
第二节 交通灯控制	10
第三节 LCD 参数显示	20
第四节 参数设置技术	27
第五节 LED 数码管显示技术	36
第六节 交通灯控制系统	48
第二章 实时检测与控制	56
第一节 直流电机转动圈数的测量	56
第二节 PWM 控制直流电动机的转速	66
第三节 实时时钟	74
第四节 直流电机转速的控制与测量	82
第三章 模拟量处理技术	92
第一节 电压信号测量	92
第二节 PT100 热电阻温度测量	103
第三节 PT100 温度测量与变送输出	114
第四章 环境温度检测系统	120
第一节 时钟修正与记录保存	120
第二节 图形 LCD 显示技术	132
第三节 环境温度监测系统	141
第五章 系统扩展技术	149
第一节 I/O 接口扩展	149
第二节 扩展 RAM 存储器	155
附录一 单片机系统配置	164
附录二 程序的编辑与编译指导	166
附录三 实验装置介绍	174
附录四 程序下载指导	178
附录五 虚拟实验操作指导	181

第一章 人机接口技术

【本章实施目标】 本章通过交通灯控制系统的分析、设计和实施,学习单片机的输入/输出接口及 SPI 接口应用方法、C 语言基本编程方法、C 语言程序编辑编译方法、单片机程序装载方法、智能设备人机界面常用设备(开关、发光管、LCD 字符显示器、矩阵键盘、LED 数码管等)的接口及应用方法、参数显示和参数设置等应用程序模块的编程方法;培养学生在单片机 I/O 接口、SPI 接口方面的应用能力,C 语言应用程序的调试能力,单片机应用系统中人机接口的分析、开发能力。

【项目要求】 交通灯控制系统用来控制交通路口南北方向和东西方向的车辆通行,红灯表示禁行,绿灯表示通行,黄灯允许过停车线的车辆通行。在交通灯控制的同时,用数码管显示变灯的时间(以 s 为单位),并要求控制系统能设置通行时间和禁行时间。

【项目实施方法】 考虑到学习的循序渐进过程,将项目分解成六个分项目,逐步实施。六个分项目分别为开关控制指示灯、交通灯控制、LCD 参数显示技术、参数设置技术、LED 数码管显示技术和交通灯控制系统。

第一节 开关控制指示灯

一、项目要求

用 ATmega16 单片机来实现开关对发光管的控制,当开关断开时发光管点亮,开关闭合时发光管熄灭。

二、项目的需求分析

需求分析是根据项目的要求,通过调查和分析,确定完成项目的基本要素,初步判断实施项目的可能性,确定完成项目的基本思路。

针对项目的要求,要实施项目必须具备五个条件。

1. 完成项目所必需的器件

完成项目必须要有开关、发光管、ATmega16 单片机,供电路使用的 5V 直流电源及电路配套的元件。可以自购元件进行电路搭接,也可以使用实验装置,当进行仿真实验时,可以采用仿真元件。

2. 完成项目的必备环境

完成项目必须要有调试程序的计算机、ATmega16 单片机程序开发软件(CodeVision-AVR 集成调试软件)、AVR 系列单片机程序下载线和下载软件(从 www.sl.com.cn 网站下载)。如果进行仿真分析,还必须要仿真软件(PROTEUS 仿真软件)。

3. 完成项目所必须做的工作

要完成项目,必须进行电路设计、程序设计、程序的编辑编译、电路的连接、程序的下

载和程序的调试。

4. 完成项目必须具备的知识

完成项目必须具备数字电路基本知识(已具备), ATmega16 单片机引脚的使用, 单片机输入/输出接口的应用方法, C 语言编程的基本规则和基本方法, 程序的编辑编译方法, 程序下载的方法。如进行仿真调试, 还应具备仿真软件的使用方法。

5. 完成项目的基本思路

(1) 熟悉 ATmega16 单片机的引脚, 单片机使用须配备的资源, 单片机输入、输出接口使用方法。

(2) 设计项目的电路。

(3) 完成程序的分析与设计。

(4) 在教师的演示、引导下实现项目, 包括程序的编辑编译, 电路连接与程序下载(或仿真电路连接与仿真运行)、结果检验与程序修改。



(a)

1	PB0(T0)	PA0(ADC0)	40
2	PB1(T1)	PA1(ADC1)	39
3	PB2(AIN0)	PA2(ADC2)	38
4	PB3(AIN1)	PA3(ADC3)	37
5	PB4(SS)	PA4(ADC0)	36
6	PB5(MOSI)	PA5(ADC5)	35
7	PB6(MISO)	PA6(ADC6)	34
8	PB7(SCK)	PA7(ADC7)	33
14	PD0(RXD)	PC0(SCL)	22
15	PD1(TXD)	PC1(SDA)	23
16	PDB2(INT0)	PC2(TCK)	24
17	PD3(INT1)	PC3(TMS)	25
18	PD4(OC1B)	PC4(TDO)	26
19	PD5(OC1A)	PC5(TDI)	27
20	PD6(ICP)	PC6(TOSC1)	28
21	PD7(OC2)	PC7(TOSC2)	29
9	RESET	VCC	10
		AVCC	30
12	XTAL2	AREF	32
13	XTAL1	AGND	31
		GND	11

(b)

图 1-1 ATmega16 单片机外形及其引脚图

(a) 外形; (b) 引脚功能

三、知识准备

(一) ATmega16 应用方法

因为 C 语言有自动管理单片机内部资源的能力, 所以, 采用 C 语言编程时, 不必将单片机的内部资源和结构了解得很清楚, 主要熟悉单片机的对外特性, 及相关资源在 C 语言编程时的使用方法。下面就本项目需要对单片机的相关应用知识作介绍。

1. ATmega16 引脚功能

ATmega16 单片机外形如图 1-1(a) 所示, 引脚编号以芯片上小圆点处为 1 号引脚, 按逆时针方向依次排列, 与 1 号引脚相对的引脚为 40 号引脚, 如图 1-1(b) 所示。其引脚功能见表 1-1。

表 1-1

ATmega16 单片机引脚功能

引脚号	引脚符号	引脚功能	引脚说明
10	VCC	工作电源正端	MEGA16 为 4.5~5.5V 直流, MEGA16L 为 2.7~5.5V
11	GND	工作电源负端	
30	AVCC	模拟电源正端	电源电压与 VCC 接近, 不超过 $\pm 0.3V$
31	AGND	模拟电源负端	
32	AREF	AD 转换的参考电压	使用在后面介绍
9	RESET	复位	低电平时进行复位

续表

引脚号	引脚符号	引脚功能	引脚说明
13	XTAL1	晶振连接端	内部反转晶振放大器的输入端和内部时钟操作电路的输入端
12	XTAL2	晶振连接端	内部反转晶振放大器的输出端
1~8	PB0~PB7	输入/输出引脚	4组8位共32个双向I/O口, 每个口可作通用输入/输出口使用, 还具有第二功能, 每一个引脚都有独立可控的内部上拉电阻。具有对称的驱动能力, 可以输出和吸收大电流, 可以直接驱动LED
14~21	PD0~PD7	输入/输出引脚	
22~29	PC0~PC7	输入/输出引脚	
33~40	PA7~PA0	输入/输出引脚	

2. 复位的方式与作用

复位有如下多种方式:

(1) 上电复位: 当电源投入时, 在电源电压低于上电复位门限时, 单片机复位。

(2) 外部复位: 在复位端与电源负端 GND 之间连接一个按键, 并在复位端与电源正端之间连接一个 10kΩ 的上拉电阻, 如图 1-2 所示。当按键按下时, 复位端变为低电平, 单片机复位。由于系统具有上电复位功能, 某些应用场合可以不接复位电路。

(3) 看门狗复位、掉电检查复位、JTAGAVR 复位等。

复位的作用是对内部资源初始化, 并使程序从规定的入口处开始执行。

3. 单片机内部存储器资源

AVR 系列单片机是一个系统完整的单片机, 采用了 Harvard 结构, 程序和数据总线分离, 且具有预取指令功能, CPU 在执行

一条指令的同时去取下一条指令, 这种模式使得指令可在一个时钟周期内完成。

ATmega16 具有 16K 字节的程序存储器 Flash。Flash 采用特殊的存储方式, 掉电后程序能保存, 用来存放程序、常数、字型码等在程序运行中不变化的数据。为了程序的安全性及自编程功能, Flash 存储器空间被分为: 引导程序段 (Boot Program Section) 和应用程序段 (Application Program Section) 两部分, 可以通过对相应熔断器位的编程设定, 选择是否需要使用引导程序段以及该段空间的大小。Flash 存储器的使用寿命至少为 1000 次的擦写循环。

ATmega16 的数据存储器 SRAM 为 1024 个字节, SRAM 用来存放运算数据, 具有随时存取功能, 但掉电后数据会丢失。

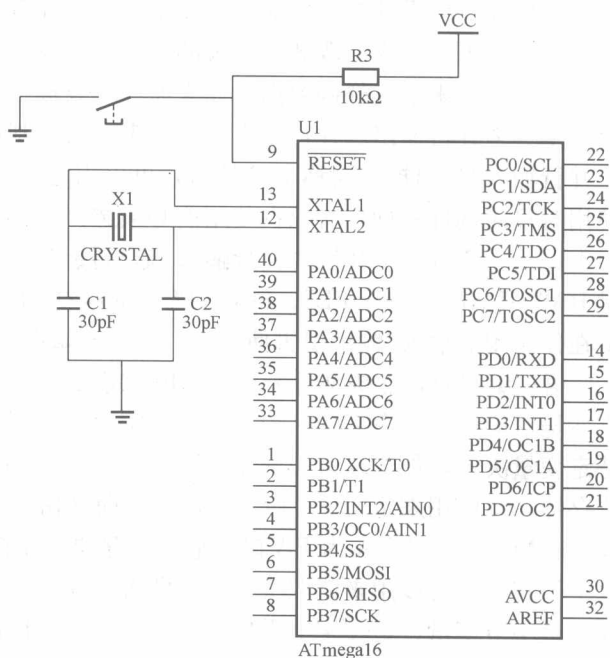


图 1-2 晶体振荡电路连接图

ATmega16 包含 512 字节的 EEPROM (也写作 E²PROM)。它是作为一个独立的数据空间而存在的,可以按字节读写,但写数据较慢,特点是掉电后数据不丢失,用来存放常量和系统参数。EEPROM 的寿命至少为 100 000 次擦写。

4. 系统时钟

每个计算机工作需要 1 个系统时钟作为工作的节拍信号。ATmega16 单片机内部有一个 1、2、4、8MHz 频率可选择的系统时钟,所以外部可以不接晶体振荡电路。如果需要其他频率的系统时钟,可以外接晶体振荡器,连接方法如图 1-2 所示。图中两个小电容的电容量为 15~30pF,帮助振荡器起振。

5. 系统配置

AVR 系列单片机在使用前,需要进行系统配置。系统配置须在教师的指导下完成(见附录一)。

6. ATmega16 单个输入/输出引脚的应用

所有单片机都设计有并行输入、输出接口。并行输入、输出接口一般为 8 位。如果通过一条输入指令可以一次读取 8 个引脚的状态,组成一个字节的数据,则称为并行输入。并行输入接口一般用来采集外部数据。如果通过一条输出指令可以将一个字节的数据送到 8 个引脚上,则称为并行输出。并行输出接口用来输出控制信息。

并行输入/输出接口简称 I/O 接口,可以用它作输入接口,也可以用它作输出接口,但在使用时只能选择一种,所以在使用前必须选择方向(是输入还是输出)。

ATmega16 有 4 个 8 位的并行输入/输出接口,分别为 PA、PB、PC 和 PD,对应的引脚是 PA0~PA7、PB0~PB7、PC0~PC7 和 PD0~PD7。每一个引脚都可以单独使用。每个引脚在作通用的输入/输出引脚使用时,处于逻辑电平状态。

C 语言编程时,在头文件 megal6.h 中定义了一些与 I/O 接口有关的特定变量,每一个特定变量对应单片机内部的一个特殊功能寄存器,用大写字母书写。每一个接口的控制由三个特定变量决定,如 PA 口对应的三个特定变量为 DDRA、PINA、PORTA。DDRA 为方向寄存器,专门用来定义 PA 口 8 个位的方向(输入还是输出)。PINA 专门用来读 PA 接口引脚上的数据。PORTA 为 PA 口的输出锁存器,在接口为输出时,用来将数据输出到接口的 8 个引脚,也可以用来读取接口输出状态,在接口为输入时连接或关闭上拉电阻。上述这些变量是字节变量,是按字节操作的,也可以单个位进行操作,在使用时,只要在这些变量后跟一个小数点和一个范围在 0~7 的数字,这个数就是要操作的位。例如,PINA.0 表示变量 PINA 的第 0 位。下面以 PD2 引脚为例,说明引脚的使用。

(1) 引脚作输入引脚:当执行 C 语言的赋值语句“DDRD.2=0;”后,引脚 PD2 只能作输入引脚,不能作输出引脚。在作输入引脚的前提下,如果执行“PORTD.2=1;”语句,在单片机内部,引脚 PD2 与电源 VCC 之间接入了一个上拉电阻;如果执行“PORTD.2=0;”语句,引脚 PD2 与 VCC 之间无上拉电阻。当执行表达式“PIND.2”时,PD2 引脚上的状态被读入单片机中,当 PD2 引脚为高电平时,读取的值为“1”;当 PD2 引脚为低电平时,读取的值为“0”。例如,将 PD2 引脚的值送给变量 x1,其语句为:“x1=PIND.2;”。

(2) 引脚作输出引脚:当执行 C 语言语句“DDRD.2=1;”后,引脚 PD2 只能作输出引脚,不能作输入引脚。执行“PORTD.2=0;”语句可以使 PD2 引脚变为低电平,而执行

“PORTD. 2=1;”语句可以使 PD2 引脚变为高电平。

(二) C 语言基本编程规则

任何计算机除硬件电路外,还必须要软件支持才能工作,软件就是程序,单片机也不例外。现在广泛采用 C 语言为单片机开发程序。下面介绍一些 C 语言的基本规则。

1. 主函数

要编写一个 C 语言程序,必须要有一个名为 main 的主函数,其格式是:

```
main ()  
{申明部分  
  执行语句部分  
}
```

主函数以 main 为函数名,后面圆括号 () 内没有内容,花括号 {} 中是主函数的内容。申明部分是对变量的说明,主要说明变量的数据类型和变量的名称,任何变量必须先申明后使用,如果没有变量需要申明,这部分可省略。计算机从主函数中的第一个可执行语句开始执行,到主函数的最后一个执行语句结束。

2. 变量名

为了与其他变量区分,每个变量必须有不同于其他变量的名字。变量名一般以字母开头,由字母、数字、下划线等组成。变量习惯用词组表示,每个词组的第一个字母用大写,其他用小写,而常量全部用大写。

3. 数据类型

每个变量将存放 1 个数据,这个数据有多大,按什么方式存储,需要几个字节存放,这都需要用变量类型来说明。例如,DDRD. 2 是位类型,用 bit 表示位类型,它只需要 1 个二进制位存放,位类型的数只能表达 0 和 1 两个值。而 DDRD 是一个无符号字符型变量,需要 1 个字节存放,无符号字符型用 unsigned char 表示,变量的表达范围为 0~255。其他类型的变量将在后面介绍。

4. 变量申明

变量申明主要是说明变量的数据类型和变量的名称,其格式如下:

```
变量类型 变量名 1, 变量名 2, ...;
```

以变量类型符开始,在类型与变量之间至少要有一个空格,一次可以申明相同类型的多个变量,变量之间用逗号隔开,最后以分号结束。

5. 赋值语句

上面列出的语句“DDRD. 2=0;”和“PORTD. 2=1;”为赋值语句,等于符号为赋值符,等号右边是表达式,等号左边是变量,分号表示语句结束,其含义是将等号右边表达式的值赋给等号左边的变量。注意每个简单语句必须以分号结束。

6. 包含

包含是一个预处理命令,用于将一个已存在的文件包含进来,这些文件可能是系统提供的单片机型号及寄存器定义,也可能是系统提供的函数,还可能是用户编写的应用函数。总之,将其包含进来后,就可以在程序中引用。其格式为:

```
#include<文件名>
```

包含以“#”开始,以小写英文词 include 为关键词,尖括号内为文件的路径和文件名。

7. 大小写和关键词规则

C语言是严格区分大小写字母的,需要小写的字母不能写成大写,需要大写的字母不能写成小写。C语言有些关键词是专门用来制定语法规则的,变量、常量名、函数名等不能使用关键词,如小写词 bit 是位类型符,不能用来作变量。

(三) 知识巩固训练

1. 选择练习

- (1) 要使 ATmega16 单片机的 PA2 为输入口,应设置()。
A: DDRA.2=0; B: DDRA.2=1; C: PINA.2=0; D: PINA.2=1。
- (2) PC4 口为输入口,要使用内部的上拉电阻,应设置()。
A: DDRC.4=1; B: PINC.4=1; C: PORTC.4=0; D: PORTC.4=1。
- (3) 用 C 语言读取 PD5 口的值到变量 k,其语句为()。
A: k=PIND5; B: k=PIND.5; C: k=PORTD.5; D: K=PIND.5。
- (4) 主函数名是()。
A: MAIN; B: Main; C: main; D: mein。

2. 填空练习

- (1) ATmega16 单片机的电源电压是_____V。
- (2) ATmega16 单片机采用_____电平复位。
- (3) ATmega16 单片机内部有 1~_____MHz 的系统时钟。

3. 判断练习

- (1) 头文件“#include<mega16.h>;”书写是正确的。
- (2) 申明“Bitk;”是正确的。
- (3) C语言的大小写字符是没有区别的。
- (4) C语言程序从主函数的第一条可执行语句开始,到主函数的最后一条语句结束。
- (5) 单片机的 I/O 接口在使用时只能接收或输出高低电平信号。
- (6) ATmega16 单片机的一个接口可以同时作输出和输入。
- (7) “12=x+2;”是赋值语句。

四、分项目的电路设计

开关的种类很多,有单刀单掷、单刀双掷、双刀单掷、双刀双掷、多路切换开关等,按使用场合不同可分为信号开关、电源开关,电源开关又可分为低压开关、高压开关。显然这里使用的是信号开关,选用单刀单掷开关就行。图 1-3 (a) 是型号为 SW-DPDT 的双刀双掷开关图形符号,图 1-3 (b) 是其引脚图,当开关向上拨,引脚 2 和 1 接通,4 和 5 接通;当

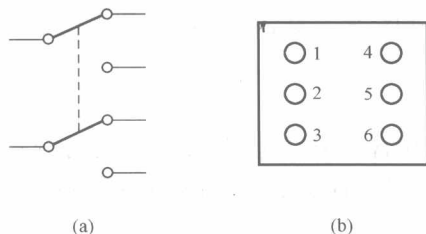


图 1-3 开关示意图

(a) SW-DPDT 双刀双掷开关图形符号; (b) 引脚图

开关向下拨,引脚 2 和 3 接通,5 和 6 接通。

项目中的指示灯采用发光二极管,简称发光管,两个引脚一长一短,长引脚端为正端,另一端为负端。发光管点亮需要一定的电流,一般在 4~20mA 较正常,电流太大会烧毁管子,太小亮度不够。发光管点亮时管压降在 2V 左右。

项目的硬件电路如图 1-4 所示。电路中,开关

的一端接地,另一端接到单片机的 PC3 口。PC3 口

用来检测开关的状态，因此作输入口使用，开关有接通和断开两个状态，而单片机只能接受高电平和低电平信号。开关闭合时，引脚 PC3 与地短接，PC3 为低电平状态。而开关断开时，需要 PC3 为高电平，所以需要在 PC3 引脚与电源正端之间接一个上拉电阻，把引脚 PC3 电平拉高，这时可在外部接上拉电阻，也可接入内部的上拉电阻，这里是接外部的上拉电阻。发光管 VD1 的正端通过限流电阻 R1 连接到电源正端 VCC 端，负端连接到 PD3 口。当 PD3 口输出高电平时，PD3 口上的电压与 VCC 接近，没有电流

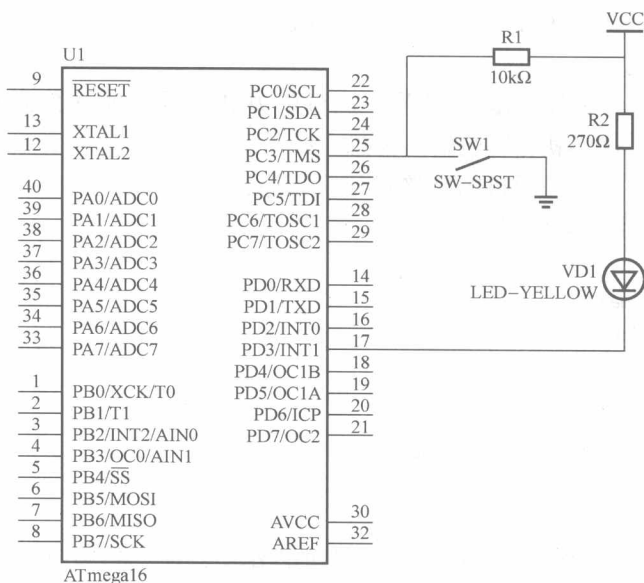


图 1-4 开关控制指示灯硬件电路

通过发光管，发光管 VD1 不亮；当 PD3 口输出低电平时，PD3 口的电压接近 0V，当电阻 R2 为 270Ω 时，发光管 VD1 通过的电流大约为 10mA，发光管点亮。

到目前为止，可以确定：

- (1) PC3 为输入口，用语句“DDRC. 3=0;”实现。
- (2) 不接入内部上拉电阻，可以用语句“PORTC. 3=0”实现。
- (3) 用 PINC. 3 读取 PC3 口的开关状态，如读到的值为 1，说明开关是断开的；如读到的值是 0，说明开关是闭合的。
- (4) PD3 口作输出口，用语句“DDRD. 3=1;”实现。
- (5) 可以对 PORTD. 3 赋值，改变 PD3 的状态。

下面确定输出与输入之间的关系，见表 1-2。

表 1-2 输出与输入关系

开 关		输出与输入的关系	指 示 灯	
开关状态	PC3		PD3	指示灯
断开	1	PD3 的状态是 PC3 的反状态	0	亮
闭合	0	PD3 的状态是 PC3 的反状态	1	不亮

由此可以得到关系表达式：

$$PORTD. 3 = \sim PINC. 3$$

写为语句是：

$$PORTD. 3 = \sim PINC. 3;$$

其中， \sim 是位取反运算符，表示把 \sim 后的表达式的值取反。其结果只会是 0 或 1。

五、分项目的程序设计

一般的单片机应用程序具有如图 1-5 所示的结构。

初始化程序完成单片机的设置、接口芯片的设置和数据的初始化。

主程序采取无限循环执行的方式,因为输入信息可能随时改变,这样就会影响输出信息,必须反复检测、运算、输出。

下面是项目的完整 C 语言程序:

```
#include <mega16.h>
bit k;
main()
{DDRC.3=0;
PORTC.3=0;
DDRD.3=1;
/* 无限循环检测与输出 */
while(1)
{k = PINC.3;
PORTD.3=~k; //将 k 变量的内容取反后,送给 PD3 口。
}
}
```

程序的第一行是一个包含命令,以#号开头,小写的英文单词 include 是关键词,括号<>内为指定文件的路径与文件名,告诉编译系统,程序包括了此处指定的文件的内容。本程序包括了 mega16.h 文件的内容,这个文件中定义了单片机接口寄存器的变量名,如程序中的 DDRC.3、PORTC.3、DDRD.3、PINC.3、PORTD.3 等位变量都是 mega16.h 文件定义的。必须指出,C 语言规定变量必须先申明其类型、变量名,然后再使用,所以包含文件一般需放在程序的前面。

程序的第二行是一个变量申明语句,表示 k 是位型变量。

程序的第三行是主函数语句,后面的内容都是主函数的内容。

程序的第四、五、六行为初始化语句,都是赋值语句。第四行的功能是使 PC3 口为入口,第五行的功能是使 PC3 口的内部上拉电阻不接入,第六行的功能是使 PD3 口为输出口。

程序的第七行是块注释,它不是可执行语句,只是一个说明,用来帮助人们理解程序。它以 /* 为起点,以 */ 为终点,中间是注释部分,可以写在多行中。

程序的第八行是一个循环语句,以小写词 while 为关键词,圆括号 () 中是一个逻辑表达式,后面的花括号 {} 内的语句都是它的执行语句,称为复合语句。其格式为:

```
while (表达式) {语句}
```

while 语句示意如图 1-6 所示。

先判断表达式,如果为真,则执行语句,执行完语句后,再判断表达式,如果是真,还执行语句,直到表达式的结果为假,while 语句才结束。其中表达式的值为 0,结果是假;为非 0 的值,结果为 1,表示真。本程序中的表达式为 1,永远为真,那就不断地循环执行语句,形成无限循环。

程序的第九行是一个赋值语句,是将读取的 PC3 口的值送给变

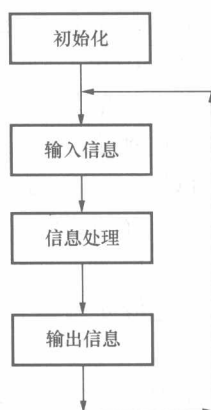


图 1-5 单片机应用程序结构简图

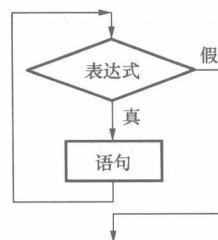


图 1-6 while 语句示意图

量 k。程序的第十行，分号前是一个赋值语句，其中符号~是位取反运算符，将~后的表达式的结果取反，如果表达式的值为 0，取反后变为 1，如果表达式的结果为非 0，取反后变为 0。这个语句的功能是把取反运算的结果送给 PD3 口。九、十两行语句是 while 语句的执行语句，通过一个中间变量 k 进行数据传递，其结果等效为“PORTD. 3=~PINC. 3;”语句。双斜杠及以后的内容为行注释语句，也是不可执行语句，与块注释的功能相同，只是以双斜杠开始，以此行的最后一个字符结束。

六、分项目实施指导

(一) 程序的编辑编译

本书以 CodevisionAVR 1.25.7 试用版为程序开发工具，程序的编辑编译要求学生在教师的示范下，参照附录二进行。主要有如下一些步骤：

- (1) 建立工程文件；
- (2) 建立原程序文件，编辑原程序；
- (3) 将文件添加到工程中；
- (4) 选择单片机和系统振荡频率；
- (5) 编译链接与修改错误。

(二) 实验电路连接（用实验装置完成实验）

关于实验装置及使用见附录三。

用扁平实验接插线连接图 1-7 所示字符标注的接口。

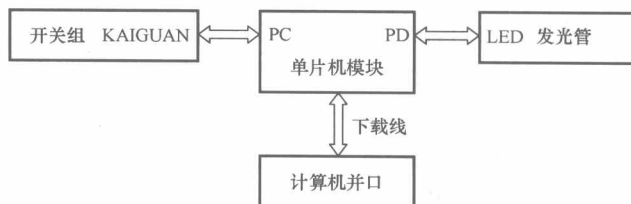


图 1-7 实验装置的电路连接示意图

(三) 程序的下载

程序的下载在教师的指导下进行，参照附录四进行。

(四) 程序的运行与修改

改变开关的状态，观看实验的结果，如果与项目要求的不相符合，请查找硬件电路和软件程序是否有错，修改程序后需重新编译和下载程序。

(五) 虚拟实验

1. 电路连接

(1) 元件的准备。本实验使用的虚拟元件见表 1-3。

表 1-3

虚拟元件说明表

元 件 名 称	虚拟元件符号	参 数 说 明
ATmega16 单片机	ATmega16	
开 关	SW-SPST-MOM	单刀单掷开关
发光管	LED-GREEN	绿色
限流电阻 R1	RES	270Ω

- (2) 元件的调入。
- (3) 电路的连接。
- (4) 参数的修改。

以上工作在教师的指导下,参照附录五进行。

2. 虚拟实验调试

(1) 程序的调入。

(2) 程序的运行与调试。改变开关的状态,观看实验的结果,如果与项目要求的不相符合,请查找硬件电路和软件程序是否有错,修改程序后需重新编译和下载程序。

以上工作在教师的指导下,参照附录五进行。

七、分项目创新训练

用单片机来实现开关对发光管的控制,将开关接到 PB6 口,指示灯接到 PA3 口,当开关断开时发光管不亮,开关闭合时发光管点亮。按小组讨论,完成项目的电路设计、程序编制,并实现项目。

八、分项目的实施评价

按表 1-4 所示内容进行检查和评价。

表 1-4 教学检查表

教 学 检 查					
	检查项目	检查结果及改进措施	应得分	实得分	
检查	实验结果正确性		20		小组讨论、 自我评价
	计划制定的准确性		10		
	独立完成实验项目的能力		15		
	知识点的掌握情况		20		
	能力控制点检查		15		
	创新能力		10		
	课外任务完成情况		10		
评价	自评:	小组:	教师评价:		

第二节 交通灯控制

一、项目要求

用发光管代替交通灯,每个方向设 3 个灯,红灯表示禁止通行,绿灯表示允许通行,黄灯表示过了停车线的车辆可以继续通行。要求控制方式是:①东西方向,红灯时间 65s、绿灯时间 40s、黄灯时间 5s;②南北方向,红灯时间 45s、绿灯时间 60s、黄灯时间 5s。

二、项目的需求分析

针对项目要求,要实施本项目必须具备五个条件。

1. 完成项目所必需的器件

完成项目必须要有 6 只发光管、ATmega16 单片机、供电路使用的 5V 直流电源及电路配套的元件。可以自购元件进行电路搭接,也可以使用实验装置,还可以采用仿真实验。

2. 完成项目的必备环境

完成项目必须要有调试程序的计算机、ATmega16 单片机程序开发软件(CodeVision-AVR 集成调试软件)、AVR 系列单片机程序下载装置和下载软件。如果进行仿真分析,还

必须要有仿真软件 (PROTEUS 仿真软件)。

3. 完成项目所必须做的工作

完成项目必须进行电路设计、程序设计、程序的编辑编译、电路的连接、程序的下载和调试。

4. 完成项目必须具备的知识

完成项目必须了解 ATmega16 单片机引脚的并行应用, 一个并行 I/O 接口的输入原理与输出原理, 交通灯控制方式, C 语言函数, 循环语句, 数据类型。

5. 完成项目的基本思路

(1) 熟悉单片机输入/输出接口并行使用方法, 掌握 C 语言的函数、数据类型、for 循环语句。

(2) 设计项目的电路, 确定控制方案。

(3) 进行程序设计, 分析项目程序。

(4) 实现项目。

三、知识准备

(一) 接口并行输入/输出

ATmega16 单片机的 32 个输入/输出接口由 4 个并行的 I/O 接口组成, 分别是 PA、PB、PC、PD 接口。在第一节中, 介绍了一个引脚的单独使用, 在这里介绍一个并行接口的同步操作。C 语言编程时, 在头文件 <mega16.h> 中定义了一些与 I/O 接口有关的特定变量, 每一个接口的控制由三个特定变量决定, 如 PA 口对应的三个特定变量为 DDRA、PINA、PORTA。DDRA 专门用来定义 PA 口 8 个引脚的方向 (是输入还是输出)。PINA 专门用来读 PA 口 8 个引脚上的数据。PORTA 为接口输出寄存器, 在接口为输出时用来将数据输出到 8 个引脚, 也可以用来读取接口输出状态; 在接口为输入时连接或关闭 8 个引脚的上拉电阻。这里以 PA 口为例介绍其使用方法。I/O 接口在使用之前, 必须先定义为输入接口或输出接口。这一定义由给变量 DDRA 赋值来完成, DDRA 的每一位对应定义 PA 口的每一引脚如下:

DDRA. 7	DDRA. 6	DDRA. 5	DDRA. 4	DDRA. 3	DDRA. 2	DDRA. 1	DDRA. 0
D7	D6	D5	D4	D3	D2	D1	D0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

当 DDRA 的某位为 0 时, 相应引脚位只能作输入。

当 DDRA 的某位为 1 时, 相应引脚位只能作输出。

例如, DDRA 的内容为 0xf0 (0b11110000), 则 PA 口的高四位为输出, 低四位为输入。当 DDRA 为 0xff 时, PA 口的所有引脚都为输出; 当 DDRA=0 时, PA 口的所有引脚都为输入。

例如, 程序段:

```
char k;
k = 0xa5;
DDRA = 0xff;
PORTA = k;
```