**电子系统设计专题实验报告**

**学 院：** 电信学院

**班 级： 计算机46**

**姓 名： 何宜晖**

**学 号： 2140504137**

**Email：yihuihe@foxmail.com**

**日 期：2016.11.27**

## 实验三、数码管动态显示程序设计

### 实验目的和要求

1. 熟悉单片机数码管电路结构原理；

2. 学会单片机C语言I/O端口编程控制方法；

3. 学会外部显示部件LED灯的软件编程使用方法；

4. 学会单片机开发系统的软件调试方法。

### 实验设备及设计开发环境

1. 单片机平台：AVR ATmega128实验开发板；

2. 开发环境平台：AVR Studio 4.18集成开发软件；WinAVR(GCC) 2010 C语言编译器；JTAGICE mkII在线仿真器。

### 实验的电路原理

### 

图1-1 数码管硬件电路连接图

两个锁存器联合控制数码管，DIG\_CS1、DIG\_CS2分别与ATmega128的PG3、PG4相连。一个锁存器控制的是数码管的位码，而另一个的锁存器控制的是数码管的段码。PORTB同时控制位码和段码，使用时要通过开启关闭锁存来实现分别控制，否则会相互影响。

在端口初始化之后，需要先关闭段码锁存器，使能位码锁存器。然后在程序里送出数码管的位码，送数码管位码，关闭位码锁存器，使能段码锁存器。送数码管段码。数码管的段码对应了数码管的不同显示方式，要显示0,1,2,3,4,5,6,7,8,9所对应的段码分别为，0xC0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90，实际使用中应当存成数组。最后关闭关闭段码锁存器。

### 程序设计及分析

初始化部分：

PORTB = ~PB\_MASK;

DDRB = PB\_MASK;

PORTG &= 0x00;

DDRG |= \_BV(DIG\_CS1) | \_BV(DIG\_CS2);

将PORTB设置为输入模式，将控制锁存器使能的MCU引脚设为输出模式。

PORTG |= \_BV(DIG\_CS2);

PORTG &= ~\_BV(DIG\_CS1);

PORTB = code[i];

关闭位码锁存器，使能段码锁存器，送数码管段码

PORTG |= \_BV(DIG\_CS1);

PORTG &= ~\_BV(DIG\_CS2);

PORTB = com;

com \*= 2;

关闭段码锁存器，使能位码锁存器，送数码管位码

PORTG &= ~\_BV(DIG\_CS1);

PORTG &= ~\_BV(DIG\_CS2);

关闭段码锁存器，关闭位码锁存器。

\_delay\_ms(500);

显示时间delay设定。

for (i = 0; i < 16; ++i) {

外层快速循环显示每个位字符，来实现多位显示。

在附加任务中，要实现LED二进制计数与数码管十进制计数同时进行，由于LED与数码管都使用PORTB输出数据。因此，使用完LED需要及时关闭。

void LED\_disable()

{

PORTC &= ~\_BV(LED\_CS);

}

相应的，使用完数码管也需要关闭。

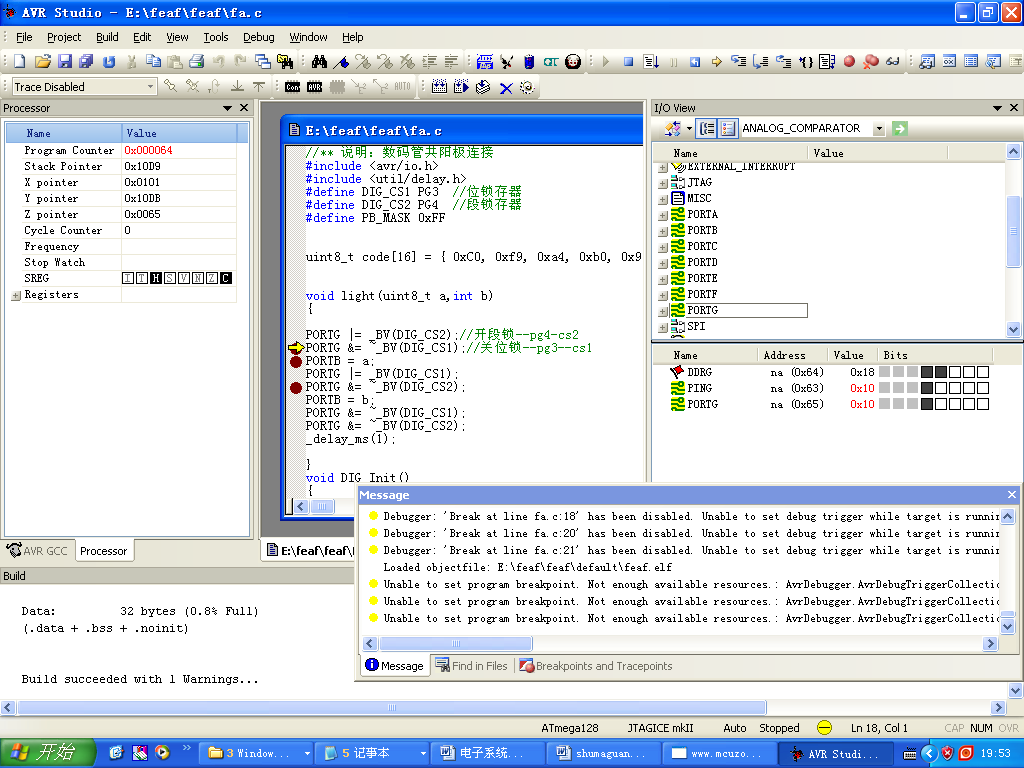
PORTG &= ~\_BV(DIG\_CS1);

PORTG &= ~\_BV(DIG\_CS2);

否则，数码管与LED会相互干扰。

### 实验结果

通过AVR studio 调试，可以查看各个锁存器的状态，进一步了解电路实际情况。



这是数码管显示十进制计数，led显示二进制计数，进行同步计数实验的结果。



### 实验总结

这次实验让我意识到了数码管显示功能的强大。也增强的对AVR studio的使用掌握，加深了对C语言控制AVR单片机的知识。同时在扩展实验中，也巩固了关于LED的知识。

最后还要感谢老师的耐心指导！

## 附录1 软件源代码

#include <avr/io.h>

#include <util/delay.h>

#define DIG\_CS1 PG3

#define DIG\_CS2 PG4

#define PB\_MASK 0xFF

#define LED\_CS PC5

void LED\_Init()

{

PORTB = PB\_MASK;

DDRB = PB\_MASK;

PORTC |= \_BV(LED\_CS);

DDRC |= \_BV(LED\_CS);

}

void LED\_disable()

{

PORTC &= ~\_BV(LED\_CS);

}

void ledshow(uint8\_t led\_sel){

LED\_Init();

PORTB |= PB\_MASK;

PORTB = ~led\_sel;

LED\_disable();

// \_delay\_ms(300);

}

uint8\_t code[16] = { 0xC0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90, 0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e };

void DIG\_Init() {

PORTB = ~PB\_MASK;

DDRB = PB\_MASK;

PORTG &= 0x00;

DDRG |= \_BV(DIG\_CS1) | \_BV(DIG\_CS2);

}

void showchar(int pos, int c, int time){

DIG\_Init();

PORTG |= \_BV(DIG\_CS2);

PORTG &= ~\_BV(DIG\_CS1);

PORTB = code[c];

PORTG |= \_BV(DIG\_CS1);

PORTG &= ~\_BV(DIG\_CS2);

PORTB = 1<<(7-pos);

PORTG &= ~\_BV(DIG\_CS1);

PORTG &= ~\_BV(DIG\_CS2);

\_delay\_ms(time);

}

void orig(int sec) {

}

void print(int num, int mode, int time) {

int c[7];

int i,rest,tmp, s;

tmp = num;

int wait = 0;

for (i=0; i <7; i++){

if (wait == 1) {

\_delay\_ms(time);

continue;}

rest = tmp % mode;

showchar(i, rest, time);

s = tmp / mode;

if ( s == 0) {

wait = 1;

}

tmp = s;

}

}

int main() {

DIG\_Init();

LED\_Init();

if (0) {//original;

orig(500);

}

else if (1) {

int i =0;

if (0) {

while(i<100){

print(15678, 10,1);

\_delay\_ms(1);

i++;

}

}

//stable

i =0;

while(1){

int j;

for (j=0;j<20;j++){

print(i, 10,1);

\_delay\_ms(1);

ledshow(i);

\_delay\_ms(1);

}

i++;

}

}

return 0;

}

**电子系统设计专题实验报告**

**学 院：** 电信学院

**班 级： 计算机46**

**姓 名： 何宜晖**

**学 号： 2140504137**

**Email：yihuihe@foxmail.com**

**日 期：2016.11.28**

## 实验四、定时器/计数器中断应用程序设计

### 实验目的和要求

1. 熟悉单片机内部定时/计数器的工作原理；

2. 了解单片机中断机制工作原理；

3. 掌握单片机中断服务程序的设计方法；

4. 了解定时计数器的功能及应用。

### 实验设备及设计开发环境

1. 单片机平台：AVR ATmega128实验开发板；

2. 开发环境平台：AVR Studio 4.18集成开发软件；WinAVR(GCC) 2010 C语言编译器；JTAGICE mkII在线仿真器。

### 实验的原理

ATmega128单片机一共有4个定时器/计数器。这次实验使用T/C1，来实现单片机的软中断，定时器定时中断。定时时间可以通过计数值、晶振频率和分频值来控制，公式为：

计数值 = 定时时间X晶振频率/分频值

TCNT1H和TCNT1L用来累加计数

OCR1AH和OCR1AL控制计数值。

TCCR1A = 0x00 设置普通端口模式

TCCR1B = 0x04设置256分频

当计数到计数值时，会触发中断

### 程序设计及分析

中断配置部分：

DDRE = \_BV(PE3); //初始化端口PE3为输出模式

PORTE &= \_BV(PE3); //输出高电平，蜂鸣器禁止发声

**TCCR1B = 0x00;** //中断控制寄存器清零，停止T/Cl计数

**OCR1AH = 0x3D;** //设置计数值

**OCR1AL = 0x09;** //设置计数值

**TCCR1A = 0x00;** //普通模式

**TCCR1B = 0x04;** // 256分频

**TIMSK |= 0x10;**  //使能C/Tl比较匹配中断

**sei();** //允许全局中断

while(1); //等待中断的发生

中断内：

TCNT1H = 0x00; //T/C1计数值清零

TCNT1L = 0x00; //T/C1计数值清零

//对PE3取反，0.5秒改变一次; x ^= 1;

if(PORTE & \_BV(PE3))

PORTE &= ~\_BV(PE3); //输出低电平

else

PORTE |= (1<<PE3); //输出高电平

### 附加实验1：改为64分频发出Do音

采用64分频时，计数值=1s\*8000000/2/64 。

发声的程序在主程序中不断循环。

while(1){

sec = time % 60;

min = (time / 60) % 60;

hour = (time / 60 / 60) % 60;

hour \* 10000 + min \* 100 + sec;

print(hour \* 10000 + min \* 100 + sec, 10,1);

int k=0;

for (;k<1;k++){

if(ifsound) {

sound(100);

}

else {

PORTE |= (1<<PE3);

}

}

}

要想实现蜂鸣器发声不使用延时函数，要根据OCR1AH和OCR1AL计算真实时间。

蜂鸣器发声时间为 t=1/523/2 s

计数器计数一次的时间为 t0=定时时间/计数值

只需要在计数器计数 t/t0次时改变一次蜂鸣器状态即可。

int beep(double ms)

{

if (TCNT1L % (int)(125\*ms)){

if (ifbeep){

ifbeep= 0;

}

else {

ifbeep =1; }

}

if (ifbeep)

PORTE &= ~(1 << PE3);

else

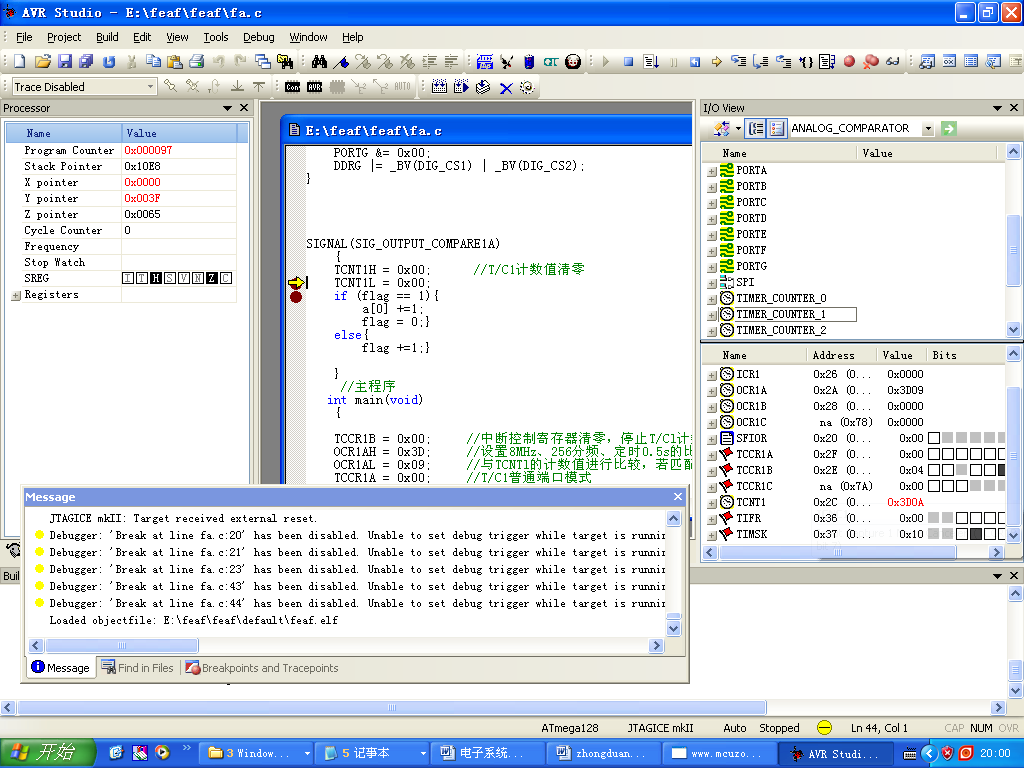
PORTE |= (1 << PE3);

return 0;

}

### 实验结果

通过调试可以观察TCNT目前的计数状态。



最后下载程序到实验板上，可以观察到电子钟。

### 实验总结

这次实验学习了如何实现使用单片机的中断，为以后的外部中断实验打下坚实基础。同时也更加熟悉了单片机技术手册的阅读。

最后还要感谢老师的耐心指导！

## 附录1 软件源代码

#include <avr/io.h>

#include <avr/io.h>

#include <util/delay.h>

#define DIG\_CS1 PG3//en for duan

#define DIG\_CS2 PG4//en for time

#define PB\_MASK 0xFF

#include <avr/interrupt.h>

#include <math.h>

uint8\_t code[16] = { 0xC0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90, 0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e };

int tmp[8] = {0,0,0,0,0,0,0,0};

int hour = 0;

int min = 0;

int sec = 0;

int time = 0;

int ifsound = 0;

int ifbeep = 0;

void DIG\_Init() {

PORTB = ~PB\_MASK;

DDRB = PB\_MASK;

PORTG &= 0x00;

DDRG |= \_BV(DIG\_CS1) | \_BV(DIG\_CS2);

}

void showchar(int pos, int c, int time){

if (1 || tmp[pos] != c) {

PORTG |= \_BV(DIG\_CS2);

PORTG &= ~\_BV(DIG\_CS1);

if (pos % 2 == 0)

PORTB = code[c] & 0x7f;

else

PORTB = code[c];

PORTG |= \_BV(DIG\_CS1);

PORTG &= ~\_BV(DIG\_CS2);

PORTB = 1<<(7-pos);

PORTG &= ~\_BV(DIG\_CS1);

PORTG &= ~\_BV(DIG\_CS2);

}

tmp[pos] = c;

\_delay\_ms(time);

}

void print(int num, int mode, int time) {

int c[7];

int i,rest,tmp, s;

tmp = num;

int wait = 0;

for (i=0; i <5; i++){

if (wait == 1) {

\_delay\_ms(time);

continue;}

rest = tmp % mode;

showchar(i, rest, time);

s = tmp / mode;

if ( s == 0) {

wait = 0;

}

tmp = s;

}

}

int beep(double ms)

{

if (TCNT1L % (int)(125\*ms)){

if (ifbeep){

//PORTE &= ~(1 << PE3);

ifbeep= 0;

}

else {

ifbeep =1;

//PORTE |= (1 << PE3);

}

}

if (ifbeep)

PORTE &= ~(1 << PE3);

else

PORTE |= (1 << PE3);

return 0;

}

void sound(double freq)

{

int i=0;

double fff=500/freq;

beep(fff);

//while(i<=200)

//{beep(fff);

//++i;}

}

SIGNAL(SIG\_OUTPUT\_COMPARE1A)

{

TCNT1H = 0x00;

TCNT1L = 0x00;

if(ifsound == 0){ // PORTE & \_BV(PE3)

ifsound=1;

// sound(523);

//PORTE &= ~\_BV(PE3);

time++;

}

else {

ifsound=0;

//PORTE |= (1<<PE3);

}

}

//Ö÷³ÌÐò

int main(void)

{

DIG\_Init();

DDRE = \_BV(PE3);

PORTE &= \_BV(PE3);

TCCR1B = 0x00;

OCR1AH = 0xF4;

OCR1AL = 0x24；

TCCR1A = 0x00;

TCCR1B = 0x03;

TIMSK |= 0x10;

sei();

PORTE = 0X08;

DDRE = 0X08;

while(1){

sec = time % 60;

min = (time / 60) % 60;

hour = (time / 60 / 60) % 60;

hour \* 10000 + min \* 100 + sec;

print(hour \* 10000 + min \* 100 + sec, 10,1);

int k=0;

for (;k<1;k++){

if(ifsound) {

sound(100);

}

else {

PORTE |= (1<<PE3);

}

}

}

}