

1、八字循环程序

```
/******
```

```
项目 : 8cvavr  
版本 : 1.0  
日期 : 2001-3-2  
作者 : Will  
广州市天河双龙电子有限公司
```

```
芯片型号 : AT90S8515  
时钟频率 : 4.000000 MHz  
存储器类型 : Small  
内部 SRAM 大小 : 512  
外部 SRAM 大小 : 0  
数据堆栈大小 : 128
```

此程序是以 CodeVision AVR 编写的以数码管作 8 字循环的程序，在主函数直接用两个 FOR 循环来作向前/后（forward march/backward march）的滚动，PORTB, PORTD 作输出。本程序在 SL-AVR 上验证通过。

```
*****/
```

```
#include <90s8515.h>  
#include <delay.h>  
// 主程序*****  
void main(void)  
{ // 局部变量  
  unsigned char i,j;  
  //输入/输出口初始化  
  // Port B  
  DDRB=0xFF;  
  PORTB=0x7f;  
  // Port D  
  DDRD=0xFF;  
  while (1)  
  {  
    j=0x01;  
    //向前滚动  
    for(i=0;i<=6;i++)  
    {  
      PORTD=~(j<<i);  
      delay_ms(250);  
    }  
  }  
}
```

```
//向后滚动
    j=0x80;
for(i=1;i<=8;i++)
    {
        PORTD=~(j>>i);
        delay_ms(250);
    }
};
}
```

2、数码管显示测试程序

```
/******
```

```
项目 : danc
版本 : 1.0
日期 : 2001-6-1
作者 : Will
广州市天河双龙电子有限公司
```

```
芯片型号 : AT90S8515
时钟频率 : 8.000000 MHz
存储器类型 : Small
内部 SRAM 大小 : 512
外部 SRAM 大小 : 0
数据堆栈大小 : 128
```

此程序是以 CodeVision AVR 编写数码管测试程序，本程序在 SL-AVR 上验证通过。

```
*****/
```

```
#include <90s8515.h>
#include <delay.h>
//全局变量
unsigned char
dance[]={0x01,0x08,0x20,0x02,0x04,0x10,0x40,0x80,0x06,0x30,0x22,0x14,0x
09,0x27,0x1e,0x3c,0x46,0x70};

void main(void)
{ unsigned char i; // 局部变量

// Port B
DDRB=0xFF;
```

```
PORTB=0x00;
// Port D
DDRD=0x3F;
PORTD=0x00;
// 看门狗定时器初始化
// 看门狗定时器预分频器: OSC/16
WDTCR=0x08;
    while(1)
    {
        for(i=0;i<18;i++)
        // Place your code here
        {
            PORTB=dance[i];
            delay_ms(200);
        }
        for(i=0;i<18;i+=2)
        {
            PORTB=dance[i];
            delay_ms(200);
        }
        for(i=0;i<18;i+=3)
        {
            PORTB=dance[i];
            delay_ms(200);
        }
        for(i=17;i>=0;i--)
        {
            PORTB=dance[i];
            delay_ms(200);
        }
        for(i=17;i>=0;i-=2)
        {
            PORTB=dance[i];
            delay_ms(200);
        }
        for(i=17;i>=0;i-=3)
        {
            PORTB=dance[i];
            delay_ms(200);
        }
    }
}
```

3、LCD 时钟程序(SL-AVR)

/******

项目 : lcdw
版本 : 1.1
日期 : 2001-8-1
作者 : will
广州市天河双龙电子有限公司

芯片型号 : AT90S8515
时钟频率 : 8.000000 MHz
存储器类型 : Small
内部 SRAM 大小 : 512
外部 SRAM 大小 : 0
数据堆栈大小 : 128

此程序是以 CodeVision AVR 编写的 LCD 时钟程序，利用定时器 T0 作 256 分频（初值为 06H），每隔八毫秒产生一次溢出中断，计满 125 次为 1 秒。每次上电/复位后显示 0: 0: 0，等待按键输入时/分/秒计时(按 SHIFT 确认)，即开始计时（无 SHIFT 确认则不启动），每当小时加一时（分为 59 而秒进入 56, 57, 58, 59, 60 时），蜂鸣器发出 BEE 声。当计时至 23: 59: 59 时时钟变为 0: 0: 0，并发出声响提示。全局数组变量 lcd_buffer 中放置要送 lcd 显示的内容，PORTB, PORTC 作输出，LCD 接 PORTB。本程序在 SL-AVR 上验证通过。

AT90S8515 与 16*2 LCD 的硬件具体接口如下：

RS-----PB.0
R/W-----PB.1
E-----PB.2
PB.3-----free
DB4-----PB.4
DB5-----PB.5
DB6-----PB.6
DB7-----PB.7

*****/

```
#include <90s8515.h>  
// LCD 接口设置  
#asm  
    .equ __lcd_port=0x18  
#endasm  
#include <lcd.h>
```

```
#include <stdio.h>
#include <delay.h>
#include <math.h>
unsigned char hh,hl,mh,ml,sh,sl,sec, minu,hr,key;
char lcd_buffer[33];

void alarm(void) //音响提示
{
    unsigned char x,y;
    DDRC=0x01;
    for(y=1;y<35;y++)
    {
        for(x=1;x<25;x++)
        {
            PORTC.0=1;
            PORTA=0x00;
            delay_us(75);
            PORTC.0=0;
            PORTA=0xFF;
            delay_us(75);
        }
    }
}

void alarm1(void) //音响提示
```

```
{ unsigned char x,y;
    DDRC=0x01;
    for(y=1;y<35;y++)
    {
        for(x=1;x<25;x++)
        {
            PORTC.0=1;
            PORTA=0x00;
            delay_us(110);
            PORTC.0=0;
            PORTA=0xFF;
            delay_us(110);
        }
    }
}
```

// 定时器 0 溢出中断服务程序

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    static unsigned char mid=0;
    // 重新初始化定时器 0
    TCNT0+=0x06;
    mid++;
    if(mid>=125)
    {
        mid=0;
        #asm("wdr")
        sec++;
        if(minu==59) //音响提示
        {
            if(sec==56) alarm();
            if(sec==57) alarm();
            if(sec==58) alarm();
            if(sec==59) alarm();
            if(sec==60) alarm1();
        }
        if(sec>=60) //秒判断
        {
            sec=0;
            minu++;
            if(minu>=60) //分判断
            {
                minu=0;
                hr++;
                if(hr>=24) //小时判断
                {
                    hr=0;
                }
                hh=hr/10;
                hl=hr%10;
            }
            mh=minu/10;
            ml=minu%10;
        }
        sh=sec/10;
        sl=sec%10;
        lcd_clear();
        sprintf lcd_buffer, "The time now is\n      %-2d:%-2d:%-2d",hr,minu,sec);
        lcd_puts lcd_buffer; //显示
    }
}
```

```
}  
}
```

```
void init(void) //初始化  
{  
    sec=0;  
    minu=0;  
    hr=0;  
    lcd_clear();  
    sprintf(lcd_buffer, "The time now is\n    %-2d:%-2d:%-2d",hr,minu,sec);  
    lcd_puts(lcd_buffer);  
}
```

```
unsigned char keyscan(void) //键盘扫描  
{  
    unsigned char row,column,temp;  
    DDRC=0xf0;  
    PORTC=0x0f;  
    if (PINC!=0x0f)  
    {  
        delay_us(30);  
        for(row=0,PORTC=0xef;row<4;row++)  
        {  
            for (column=0,temp=0xfe;column<4;column++)  
            {  
                while((PINC&0x0f)==(temp&0x0f))  
                {  
                    key=4*row+column;  
                    return (1);  
                }  
                temp=((temp<<1)|0x01);  
            }  
            PORTC=((PORTC<<1)|0x01);  
        }  
    }  
    return (0);  
}
```

```
void main(void) //主程序  
{  
    unsigned char p;  
    // 输入/输出口初始化  
    // Port A
```

```
DDRA=0xff;
PORTA=0xff;
// Port B
DDRB=0xFF;
PORTB=0x00;
// Port C
DDRC=0xf0;
PORTC=0x0f;
// Port D
DDRD=0x00;
PORTD=0xff;
//定时器/计数器 0 初始化
// 时钟源 : 系统时钟
// 时钟值 : 31.250 kHz
// 模式 : 输出比较
// OC0 输出 : 不连接
TCCR0=0x04;
TCNT0=0x06;
//定时器/计数器 1 初始化
//时钟源 : 系统时钟
//时钟值 : 定时器 1 停止
//模式 : 输出比较
// OC1A 输出 : 不连接
// OC1B 输出 : 不连接
// 噪声消除 : 关闭
// 下降沿输入捕获
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// 外部中断初始化
// INT0 : 关闭
// INT1 : 关闭
GIMSK=0x00;
MCUCR=0x00;
//定时器/计数器中断初始化
TIMSK=0x02;
```



```
// 模拟比较器初始化
// 模拟比较器 : 关闭
// 模拟比较器输入捕获 : 关闭
ACSR=0x80;
hh=0;
hl=0;
mh=0;
ml=0;
sh=0;
sl=0;
p=0;
key=0;
// LCD 模式初始化
lcd_init(16);
init();
    while(PIND.7!=0) //无 SHIFT 键确认
    {
        while(keyscan()) //有键按下
        {
            if(PIND.7!=0) //如不是 SHIFT 键按下
            {
                ++p;
                switch(p) //显示
                {
                    case 1: //小时十位
                        {
                            while(keyscan())
                            {
                                hh=key;
                                hr=10*hh+hl;
                                delay_us(50);
                                lcd_clear();
                                sprintf(lcd_buffer, "The time now is\n    %-2d:%-2d:%-2d",hr, minu, sec);
                                lcd_puts(lcd_buffer);
                                delay_ms(150);
                            }
                        };
                    break;
                    case 2: //小时个位
                        {
                            while(keyscan())
                            {
```

```
        hl=key;
        hr=hh*10+hl;
        delay_us(50);
    lcd_clear();
    sprintf lcd_buffer, "The time now is\n    %-2d:%-2d:%-2d",hr, minu, sec);
    lcd_puts lcd_buffer;
    delay_ms(150);
    }
};
break;
case 3: //分钟十位
    {
    while(keyscan())
    {
    mh=key;
    minu=mh*10+ml;
    lcd_clear();
    sprintf lcd_buffer, "The time now is\n    %-2d:%-2d:%-2d",hr, minu, sec);
    lcd_puts lcd_buffer;
    delay_ms(150);
    }
    };
break;
case 4: //分钟个位
    {
    while(keyscan())
    {
    ml=key;
    minu=mh*10+ml;
    lcd_clear();
    sprintf lcd_buffer, "The time now is\n    %-2d:%-2d:%-2d",hr, minu, sec);
    lcd_puts lcd_buffer;
    delay_ms(150);
    }
    };
break;
case 5: //秒钟十位
    {
    while(keyscan())
    {
    sh=key;
    sec=sh*10+sl;
```

```
        lcd_clear();
sprintf lcd_buffer, "The time now is\n      %-2d:%-2d:%-2d",hr, minu,sec);
        lcd_puts lcd_buffer);
        delay_ms(150);
        }
};
break;
case 6:                                     //秒钟个位
{
    while(keyscan())
    {
        sl=key;
        sec=sh*10+sl;
        lcd_clear();
sprintf lcd_buffer, "The time now is\n      %-2d:%-2d:%-2d",hr, minu,sec);
        lcd_puts lcd_buffer);
        delay_ms(150);
        }
    };
break;
default: break;
}
}
}

}
// 看门狗定时器初始化
// 看门狗定时器预比例器: OSC/2048
WDTCR=0x0f;
// 全局中断允许
#asm("sei")
while (1)
{

};
}
```

4.LCD 倒计时程序 (SL-AVR)

```
/******
```

项目 : lcdtdw
版本 : 1.0
日期 : 2001-8-3
作者 : Will
广州市天河双龙电子有限公司

芯片型号 : AT90S8515
时钟频率 : 8.000000 MHz
存储器类型 : Small
内部 SRAM 大小 : 512
外部 SRAM 大小 : 0
数据堆栈大小 : 128

此程序是以 CodeVision AVR 编写的 LCD 时钟程序，利用定时器 T0 作 256 分频（初值为 06H），每隔八毫秒产生一次溢出中断，计满 125 次为 1 秒。每次上电/复位后显示 0: 0: 0，等待按键输入时/分/秒计时(按 SHIFT 确认)，即开始计时（无 SHIFT 确认则不启动），每当小时减一时（分减为 00 而秒减为 4, 3, 2, 1, 0 时），蜂鸣器发出 BEE 声。全局数组变量 lcd_buffer 中放置要送 lcd 显示的内容，PORTB, PORTC 作输出，LCD 接 PORTB。

本程序在 SL-AVR 上验证通过。
AT90S8515 与 16*2 LCD 的硬件具体接口如下：

```
RS-----PB.0  
R/W-----PB.1  
E-----PB.2  
PB.3-----free  
DB4-----PB.4  
DB5-----PB.5  
DB6-----PB.6  
DB7-----PB.7  
*****/  
#include <90s8515.h>  
// LCD 接口设置  
#asm  
    .equ __lcd_port=0x18  
#endasm
```

```
#include <lcd.h>
#include <stdio.h>
#include <delay.h>
#include <math.h>
unsigned char hh,hl,mh,ml,sh,sl,sec,minu,hr,key;
char lcd_buffer[33];

void alarm(void) //音响提示子程序
{
    unsigned char x,y;
    DDRC=0x01;
    for(y=1;y<35;y++)
    {
        for(x=1;x<25;x++)
        {
            PORTC.0=1;
            PORTA=0x00;
            delay_us(65);
            PORTC.0=0;
            PORTA=0xFF;
            delay_us(65);
        }
    }
}

void alarml(void) //音响提示子程序
{
    unsigned char x,y;
    DDRC=0x01;
    for(y=1;y<35;y++)
    {
        for(x=1;x<25;x++)
        {
            PORTC.0=1;
            PORTA=0x00;
            delay_us(95);
            PORTC.0=0;
            PORTA=0xFF;
            delay_us(95);
        }
    }
}

void init(void) //初始化子程序
{
```

```
    sec=0;
    minu=0;
    hr=0;
    lcd_clear();
    sprintf lcd_buffer, "Time now remains\n    %-2d:%-2d:%-2d", hr, minu, sec);
    lcd_puts lcd_buffer;
}
//定时器 0 溢出中断服务程序
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    static unsigned char mid=0;
    //重新初始化定时器 0
    TCNT0+=0x06;
    mid++;
    if(mid>=125)
    {
        mid=0;
        if((sec<60)&(sec>=0))           //秒判断
        {
            sec--;
            if(minu==0)                 //音响提示
            {
                if(sec==4) alarm();
                if(sec==3) alarm();
                if(sec==2) alarm();
                if(sec==1) alarm();
                if(sec==0) alarmml();
            }
            if((sec==0xff)&(minu>=0)&(minu<60)) //秒变 0, 分判断
            {
                minu--;
                sec=59;
                if((minu==0xff)&(hr>0)&(hr<24)) //分变 0, 小时判断
                {
                    minu=59;
                    hr--;
                    if((hr==0)&(minu==0)&(sec==0))
                    {
                        init();
                    }
                    hh=hr/10;
                    hl=hr%10;
                }
            }
        }
    }
}
```

```

    }
    mh=minu/10;
    ml=minu%10;
  }
  sh=sec/10;
  sl=sec%10;
  lcd_clear();
  sprintf lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr,minu,sec);
  lcd_puts lcd_buffer);
  if((sec==0)&(minu==0)&(hr==0)) //秒,分,时均为0
  {
    hr=23;
    minu=59;
    sec=59;
    hh=hr/10;
    hl=hr%10;
    mh=minu/10;
    ml=minu%10;
    sh=sec/10;
    sl=sec%10;
    lcd_clear();
    sprintf lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr,minu,sec);
    lcd_puts lcd_buffer);
  }
}
}
}

```

unsigned char keyscan(void) //键盘扫描子程序

```

{
  unsigned char row,column,temp;
  DDRC=0xf0;
  PORTC=0x0f;
  if (PINC!=0x0f)
  {
    delay_us(30);
    for(row=0,PORTC=0xef;row<4;row++)
    {
      for (column=0,temp=0xfe;column<4;column++)
      {
        while((PINC&0x0f)==(temp&0x0f))
        {
          key=4*row+column;

```

```
        return (1);
    }
    temp=((temp<<1)|0x01);
}
PORTC=((PORTC<<1)|0x01);
}
}
return (0);
}
```

```
void main(void)                //主程序
{
    // 局部变量
    unsigned char p;
    // 输入/输出口初始化
    // Port A
    PORTA=0x00;
    DDRA=0xFF;
    // Port B
    PORTB=0x00;
    DDRB=0xFF;
    // Port C
    PORTC=0x00;
    DDRC=0xF0;
    // Port D
    PORTD=0xFF;
    DDRD=0x00;
    //定时器/计数器 0 初始化
    // 时钟源 : 系统时钟
    // 时钟值 : 31.250 kHz
    // 模式 : 输出比较
    // OC0 输出 : 不连接
    TCCR0=0x04;
    TCNT0=0x06;
    //定时器/计数器 1 初始化
    //时钟源 : 系统时钟
    //时钟值 : 定时器 1 停止
    //模式 : 输出比较
    // OC1A 输出 : 不连接
    // OC1B 输出 : 不连接
    // 噪声消除 : 关闭
    // 下降沿输入捕获
```



```
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// 外部中断初始化
// INT0: 关闭
// INT1: 关闭
GIMSK=0x00;
MCUCR=0x00;
//定时器/计数器中断初始化
TIMSK=0x02;
// 模拟比较器初始化
// 模拟比较器 : 关闭
// 模拟比较器输入捕获 : 关闭
ACSR=0x80;
hh=0;
hl=0;
mh=0;
ml=0;
sh=0;
sl=0;
p=0;
key=0;
// LCD 模式初始化
lcd_init(16);
init();

    while(PIND.7!=0)                //无 SHIFT 键确认
    {
        while(keyscan())           //有键按下
        {
            if(PIND.7!=0)          //如不是 SHIFT 键按下
            {
                ++p;
                switch(p)           //显示
                {
                    case 1:         //小时十位
```

```
        {
            while(keyscan())
            {
                hh=key;
                hr=10*hh+hl;
                delay_us(50);
            }
            lcd_clear();
            sprintf(lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr, minu, sec);
            lcd_puts(lcd_buffer);
            delay_ms(150);
        }
    };
    break;
case 2:                                     //小时个位
    {
        while(keyscan())
        {
            hl=key;
            hr=hh*10+hl;
            delay_us(50);
        }
        lcd_clear();
        sprintf(lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr, minu, sec);
        lcd_puts(lcd_buffer);
        delay_ms(150);
    }
    };
    break;
case 3:                                     //分钟十位
    {
        while(keyscan())
        {
            mh=key;
            minu=mh*10+ml;
            lcd_clear();
        }
        sprintf(lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr, minu, sec);
        lcd_puts(lcd_buffer);
        delay_ms(150);
    }
    };
    break;
case 4:                                     //分钟个位
    {
```

```
        while(keyscan())
        {
            ml=key;
            minu=mh*10+ml;
            lcd_clear();
            sprintf(lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr,minu,sec);
            lcd_puts(lcd_buffer);
            delay_ms(150);
        }
    };
    break;

case 5:                                     //秒钟十位
    {
        while(keyscan())
        {
            sh=key;
            sec=sh*10+sl;
            lcd_clear();
            sprintf(lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr,minu,sec);
            lcd_puts(lcd_buffer);
            delay_ms(150);
        }
    };
    break;

case 6:                                     //秒钟个位
    {
        while(keyscan())
        {
            sl=key;
            sec=sh*10+sl;
            lcd_clear();
            sprintf(lcd_buffer, "Time now remains\n      %-2d:%-2d:%-2d",hr,minu,sec);
            lcd_puts(lcd_buffer);
            delay_ms(150);
        }
    };
    break;

default: break;
}
}
}
```

```
}
//*****
//如无时、分、秒设置，且有 SHIFT 按键确认
if((sec==0)&(minu==0)&(hr==0))
    {
    hr=23;
    minu=59;
    sec=59;
    hh=hr/10;
    hl=hr%10;
    mh=minu/10;
    ml=minu%10;
    sh=sec/10;
    sl=sec%10;
    lcd_clear();
    sprintf(lcd_buffer, "Time now remains\n    %-2d:%-2d:%-2d",hr,minu,sec);
    lcd_puts(lcd_buffer);
    }
//全局中断允许
#asm("sei")

while (1)
    {
    // Place your code here

    };
}
```

5、定时器程序 (SL-AVR-2)

```

/*****

```

```

项目 : newtim
版本 : 1.0
日期 : 2001-7-14
作者 : Will
广州市天河双龙电子有限公司

```

```

芯片型号 : AT90S8535
时钟频率 : 8.000000 MHz
存储器类型 : Small
内部 SRAM 大小 : 512
外部 SRAM 大小 : 0
数据堆栈大小 : 128

```

此程序是以 CodeVision AVR 编写的时钟计时程序，利用定时器 T0 作 256 分频（初值为 06H），每隔八毫秒产生一次溢出中断，计满 125 次为 1 秒。每当小时加一时（分为 59 而秒进入 56, 57, 58, 59, 60 时），蜂鸣器发出 BEE 声。

程序启动时输入时/分/秒的数值然后按 SHIFT 键确认或直接按 SHIFT 键以启动计时，否则程序不会计时。

本程序在 SL-AVR-2 上调试通过。

```

*****/

```

```

#include <90s8535.h>
#include <delay.h>
unsigned char sec,min,hr,temp,sl,sh,ml,mh,hl,hh;
void transmit(unsigned char para)
{
    unsigned char i,j;
    PORTC.4=0; //SL279 片选有效
    delay_us(50);
    j=para;
    for(i=8;i>0;i--)
    {
        if((j&0x80)==0x80)
            PORTC.2=1; //数据端
        else
            PORTC.2=0;
        PORTC.3=1;
        delay_us(10);
    }
}

```

```
    PORTC.3=0;
    delay_us(10);
    j=j<<1;
}
}

void receive()
{
    unsigned char k;
    DDRC.2=0;
    delay_us(50);
    for(k=8;k>0;k--)
    {
        PORTC.3=1;                //时钟端
        delay_us(10);
        if(PINC.2==1)
            temp|=0x01;
        else
            temp&=0xfe;
        PORTC.3=0;
        delay_us(10);
        if(k>1)
            temp=temp<<1;
    }
    DDRC.2=1;
}
//显示子程序
void disp(unsigned char ls,unsigned char hs, unsigned char lm,unsigned char
hm,unsigned char lh,unsigned char h)
{ unsigned char sll,shh,mll,mhh,hll,hhh;
  sll=ls;
  shh=hs;
  mll=lm;
  mhh=hm;
  hll=lh;
  hhh=h;
  transmit(0xc8);                //D0->;
  delay_us(8);
  transmit(sll);
  PORTC.4=1;
  delay_us(20);
  transmit(0xc9);                //D1-> ;
  delay_us(8);
```

```
transmit(shh);
  PORTC.4=1;
  delay_us(20);
transmit(0xcb); //D3-> ;
  delay_us(8);
transmit(mll);
  PORTC.4=1;
  delay_us(20);
transmit(0xcc); //D4-> ;
  delay_us(8);
transmit(mhh);
  PORTC.4=1;
  delay_us(20);
transmit(0xce); //D6-> ;
  delay_us(8);
transmit(hll);
  PORTC.4=1;
  delay_us(20);
transmit(0xcf); //D7-> ;
  delay_us(8);
transmit(hhh);
  PORTC.4=1;
  delay_us(20);
}
```

```
void alarm(void)
{ unsigned char x,y;
  for(y=1;y<35;y++)
  {
    for(x=1;x<25;x++)
    {
      PORTC.0=1;
      PORTB=0x00;
      delay_us(75);
      PORTC.0=0;
      PORTB=0xFF;
      delay_us(75);
    }
  }
}
```

```
void alarml(void)
{ unsigned char x,y;
```

```
for(y=1;y<35;y++)
{
    for(x=1;x<25;x++)
    {
        PORTC.0=1;
        PORTB=0x00;
        delay_us(110);
        PORTC.0=0;
        PORTB=0xFF;
        delay_us(110);
    }
}
}
// 定时器 0 溢出中断服务程序
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    static unsigned char mid=0;
// 重新初始化定时器 0
    TCNT0+=0x06;
// Place your code here
    mid++;
    if(mid>=125)
    {
        mid=0;
        sec++;
        if(min==59) //音响提示
        {
            if(sec==56) alarm();
            if(sec==57) alarm();
            if(sec==58) alarm();
            if(sec==59) alarm();
            if(sec==60) alarml();
        }
        if(sec>=60) //秒判断
        {
            sec=0;
            min++;
            if(min>=60) //分判断
            {
                min=0;
                hr++;
                if(hr>=24) //小时判断
```



```
        {
            hr=0;
        }
        hh=hr/10;
        hl=hr%10;
    }
    mh=min/10;
    ml=min%10;
}
sh=sec/10;
sl=sec%10;
disp(sl,sh,ml,mh,hl,hh);
}
}

void init()
{
    unsigned char l;
    PORTC.4=1;                //SL279 片选无效
    PORTC.3=0;                //时钟端
    delay_us(100);
    transmit(0x80);           //D0 位-> 0;
    delay_us(10);
    transmit(0x0f);
    PORTC.4=1;
    delay_us(10);
    transmit(0x81);           //D1 位-> 0;
    delay_us(10);
    transmit(0x0f);
    PORTC.4=1;
    delay_us(10);
    transmit(0xe0);           //D2 位-> -;
    delay_us(10);
    l=0x00;
    l+=(0x08+0x08);
    transmit(l);
    PORTC.4=1;
    delay_us(10);
    transmit(0x83);           //D3 位-> 0;
    delay_us(10);
    transmit(0x0f);
    PORTC.4=1;
```

```
delay_us(10);
transmit(0x84);           //D4 位-> 0;
delay_us(10);
transmit(0x0f);
PORTC.4=1;
delay_us(15);
l+=(0x08+0x08+0x08);
transmit(0xe0);         //D5 位-> -;
delay_us(10);
transmit(l);
PORTC.4=1;
delay_us(10);
transmit(0x86);         //D6 位-> 0;
delay_us(10);
transmit(0x0f);
PORTC.4=1;
delay_us(10);
transmit(0x87);         //D7 位-> 0;
delay_us(10);
transmit(0x0f);
PORTC.4=1;
delay_us(10);
transmit(0x88);         //去闪烁;
delay_us(10);
transmit(0xff);
PORTC.4=1;
delay_us(10);
}

void main(void)
{   // 局部变量
    unsigned char p;
//Port B
DDRB=0xFF;
PORTB=0xFF;
// Port C
DDRC=0xFF;
PORTC.4=1;           // SL279 片选无效
PORTC.3=0;           //时钟端
PORTC.1=1;           // 无键按下
delay_ms(20);
sl=0x00;
```



```
case 3: //分钟十位
    {
        transmit(0xcc);
        transmit(temp);
        PORTC.4=1;
        mh=temp;
        delay_us(150);
    };
    break;
case 4: //分钟个位
    {
        transmit(0xcb);
        transmit(temp);
        PORTC.4=1;
        ml=temp;
        delay_us(150);
    };
    break;
case 5: 秒 //秒钟十位
    {
        transmit(0xc9);
        transmit(temp);
        PORTC.4=1;
        sh=temp;
        delay_us(150);
    };
    break;
case 6: 钟 //秒钟个位
    {
        transmit(0xc8);
        transmit(temp);
        PORTC.4=1;
        sl=temp;
        delay_us(150);
    };
    break;
default: break;
    } //switch>>>>>>>>>>>>>>>>>>
    } //while !!!!!!!
}
}
```

```
/**
 *
 */
sec=sh*10+sl;
min=mh*10+ml;
hr=hh*10+hl;
TCCR0=0x00;
//定时器/计数器 0 初始化
// 时钟源 : 系统时钟
// 时钟值 : 31.250 kHz
// 模式 : 输出比较
// OC0 输出 : 不连接
TCCR0=0x04;
TCNT0=0x06;
// 外部中断初始化
// INT0 : 关闭
// INT1 : 关闭
GIMSK=0x00;
MCUCR=0x00;
//定时器/计数器中断初始化
TIMSK=0x01;
// 模拟比较器初始化
// 模拟比较器 : 关闭
// 模拟比较器输入捕获 : 关闭
ACSR=0x80;
// 全局中断允许
asm("sei")

while(1)
{
;
;
};
}
```

6、倒计时程序 (SL-AVR-2)

```

/*****

```

```

项目 : newdcount
版本 : 1.0
日期 : 2001-7-14
作者 : Will
广州市天河双龙电子有限公司

```

```

芯片型号 : AT90S8535
时钟频率 : 8.000000 MHz
存储器类型 : Small
内部 SRAM 大小 : 512
外部 SRAM 大小 : 0
数据堆栈大小 : 128

```

此程序是以 CodeVision AVR 编写的倒计时程序，利用定时器 T0 作 256 分频（初值为 06H），每隔八毫秒产生一次溢出中断，计满 125 次为 1 秒。每当小时减一时（分减为 00 而秒减为 4, 3, 2, 1, 0 时），蜂鸣器发出 BEE 声。

程序启动时输入时/分/秒的数值然后按 SHIFT 键确认或直接按 SHIFT 键以启动计时，否则程序不会计时。

本程序在 SL-AVR-2 上调试通过。

```

*****/

```

```

#include <90s8535.h>
#include <delay.h>
unsigned char sec,min,hr,temp,sl,sh,ml,mh,hl,hh; //全局变量
void transmit(unsigned char para) //发送子程序
{
    unsigned char ij;
    PORTC.4=0; //SL279 片选有效
    delay_us(50);
    j=para;
    for(i=8;i>0;i--)
    {
        if((j&0x80)==0x80)
            PORTC.2=1; //数据端
        else
            PORTC.2=0;
            PORTC.3=1;
            delay_us(10);
    }
}

```

```
    PORTC.3=0;
    delay_us(10);
    j=j<<1;
  }
}
```

```
void receive() //接收子程序
```

```
{
  unsigned char k;
  DDRC.2=0;
  delay_us(50);
  for(k=8;k>0;k--)
  {
    PORTC.3=1; //时钟端
    delay_us(10);
    if(PINC.2==1)
      temp|=0x01;
    else
      temp&=0xfe;
    PORTC.3=0;
    delay_us(10);
    if(k>1)
      temp=temp<<1;
  }
  DDRC.2=1;
}
```

```
//显示子程序
```

```
void disp(unsigned char ls,unsigned char hs, unsigned char lm,unsigned char
hm,unsigned char lh,unsigned char h)
```

```
{ unsigned char sll,shh,mll,mhh,hll,hhh;
  sll=ls;
  shh=hs;
  mll=lm;
  mhh=hm;
  hll=lh;
  hhh=h;
  transmit(0xc8); //D0->;
  delay_us(8);
  transmit(sll);
  PORTC.4=1;
  delay_us(20);
```

```
transmit(0xc9);           //D1-> ;
  delay_us(8);
transmit(shh);
  PORTC.4=1;
  delay_us(20);
transmit(0xcb);          //D3-> ;
  delay_us(8);
transmit(mll);
  PORTC.4=1;
  delay_us(20);
transmit(0xcc);          //D4-> ;
  delay_us(8);
transmit(mhh);
  PORTC.4=1;
  delay_us(20);
transmit(0xce);          //D6-> ;
  delay_us(8);
transmit(hll);
  PORTC.4=1;
  delay_us(20);
transmit(0xcf);          //D7-> ;
  delay_us(8);
transmit(hhh);
  PORTC.4=1;
  delay_us(20);
}
```

```
void alarm(void)          //发声子程序
{ unsigned char x,y;
  for(y=1;y<35;y++)
  {
    for(x=1;x<25;x++)
    {
      PORTC.0=1;
      PORTB=0x00;
      delay_us(75);
      PORTC.0=0;
      PORTB=0xFF;
      delay_us(75);
    }
  }
}
```



```
void alarm1(void) //发声子程序
{ unsigned char x,y;
  for(y=1;y<35;y++)
  {
    for(x=1;x<25;x++)
    {
      PORTC.0=1;
      PORTB=0x00;
      delay_us(110);
      PORTC.0=0;
      PORTB=0xFF;
      delay_us(110);
    }
  }
}

void init() // 初始化子程序
{
  unsigned char l;
  PORTC.4=1; //SL279 片选无效
  PORTC.3=0; //时钟端
  delay_us(100);
  transmit(0x80); //D0 位-> 0;
  delay_us(10);
  transmit(0x0f);
  PORTC.4=1;
  delay_us(10);
  transmit(0x81); //D1 位-> 0;
  delay_us(10);
  transmit(0x0f);
  PORTC.4=1;
  delay_us(10);
  transmit(0xe0); //D2 位-> -;
  delay_us(10);
  l=0x00;
  l+=(0x08+0x08);
  transmit(l);
  PORTC.4=1;
  delay_us(10);
  transmit(0x83); //D3 位-> 0;
  delay_us(10);
  transmit(0x0f);
  PORTC.4=1;
```

```
delay_us(10);
transmit(0x84);           //D4 位-> 0;
delay_us(10);
transmit(0x0f);
PORTC.4=1;
delay_us(15);
l+=(0x08+0x08+0x08);
transmit(0xe0);         //D5 位-> -;
delay_us(10);
transmit(l);
PORTC.4=1;
delay_us(10);
transmit(0x86);         //D6 位-> 0;
delay_us(10);
transmit(0x0f);
PORTC.4=1;
delay_us(10);
transmit(0x87);         //D7 位-> 0;
delay_us(10);
transmit(0x0f);
PORTC.4=1;
delay_us(10);
transmit(0x88);         //去闪烁;
delay_us(10);
transmit(0xff);
PORTC.4=1;
delay_us(10);
}
```

```
// 定时器 0 溢出中断服务程序
```

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
```

```
    static unsigned char mid=0;
```

```
// 重新初始化定时器 0
```

```
    TCNT0+=0x06;
```

```
    mid++;
```

```
if(mid>=125)
```

```
{
```

```
    mid=0;
```

```
    if((sec<60)&(sec>=0))           //秒判断
```

```
    {
```

```
        sec--;
```

```
if(min==0) //音响提示
{
    if(sec==4) alarm();
    if(sec==3) alarm();
    if(sec==2) alarm();
    if(sec==1) alarm();
    if(sec==0) alarml();
}
if((sec==0xff)&(min>=0)&(min<60)) //秒变 0, 分判断
{
    min--;
    sec=59;
    if((min==0xff)&(hr>0)&(hr<24)) //分变 0, 小时判断
    {
        min=59;
        hr--;
        if((hr==0)&(min==0)&(sec==0))
        {
            init();
        }
        hh=hr/10;
        hl=hr%10;
    }
    mh=min/10;
    ml=min%10;
}
sh=sec/10;
sl=sec%10;
disp(sl,sh,ml,mh,hl,hh);
if((sec==0)&(min==0)&(hr==0))
{
    hr=23;
    min=59;
    sec=59;
    hh=hr/10;
    hl=hr%10;
    mh=min/10;
    ml=min%10;
    sh=sec/10;
    sl=sec%10;
    disp(sl,sh,ml,mh,hl,hh);
}
```

```

    }
  }
}

void main(void) //主程序
{ // 局部变量
  unsigned char p;
  //Port B
  DDRB=0xFF;
  PORTB=0xFF;
  // Port C
  DDRC=0xFF;
  PORTC.4=1; // SL279 片选无效
  PORTC.3=0; //时钟端
  PORTC.1=1; // 无键按下
  delay_ms(2000);
  sl=0;
  sh=0;
  ml=0;
  mh=0;
  hl=0;
  hh=0;
  sec=0;
  min=0;
  hr=0;
  init();
  p=0;
  temp=0;
  while(temp!=0x17) //无 SHIFT 键按下确认
  {
    while(PINC.1==0) //有键按下
    { //while !!!!!!!
      delay_us(10);
      transmit(0x15); //读键
      receive();
      PORTC.4=1;
      delay_us(10);
    }
  }
  if(temp!=0x17) //所按键不是 SHIFT
  { p++;
    switch(p)
    { //将按键送显示

```

```
case 1: //小时十位
    {
        transmit(0xcf);
        transmit(temp);
        PORTC.4=1;
        hh=temp;
        delay_us(150);
    };
break;
case 2: //小时个位
    {
        transmit(0xce);
        transmit(temp);
        PORTC.4=1;
        hl=temp;
        delay_us(150);
    };
break;
case 3: //分钟十位
    {
        transmit(0xcc);
        transmit(temp);
        PORTC.4=1;
        mh=temp;
        delay_us(150);
    };
break;
case 4: //分钟个位
    {
        transmit(0xcb);
        transmit(temp);
        PORTC.4=1;
        ml=temp;
        delay_us(150);
    };
break;
case 5: //秒钟十位
    {
        transmit(0xc9);
        transmit(temp);
        PORTC.4=1;
        sh=temp;
```

```

        delay_us(150);
    };
    break;
case 6:                                     //秒钟个位
    {
        transmit(0xc8);
        transmit(temp);
        PORTC.4=1;
        sl=temp;
        delay_us(150);
    };
    break;
default: break;
    }
}
}
}
}
//*****

    sec=sh*10+sl;
    min=mh*10+ml;
    hr=hh*10+hl;
    TCCR0=0x00;
    //定时器/计数器 0 初始化
    // 时钟源 : 系统时钟
    // 时钟值 : 31.250 kHz
    // 模式 : 输出比较
    // OC0 输出 : 不比较
    TCCR0=0x04;
    TCNT0=0x06;
    // 外部中断初始化
    // INT0 : 关闭
    // INT1 : 关闭
    GIMSK=0x00;
    MCUCR=0x00;
    //定时器/计数器中断初始化
    TIMSK=0x01;
    // 模拟比较器初始化
    // 模拟比较器 : 关闭
    // 模拟比较器输入捕获 : 关闭
    ACSR=0x80;
//无时、分、秒设置，而有 SHIFT 键确认

```

```
if((sec==0)&(min==0)&(hr==0))
{
    hr=23;
    min=59;
    sec=59;
    hh=hr/10;
    hl=hr%10;
    mh=min/10;
    ml=min%10;
    sh=sec/10;
    sl=sec%10;
    disp(sl,sh,ml,mh,hl,hh);
}
#asm("sei") // 全局中断允许

while(1)
{
    ;
    ;
};
}
```

7、八路 A/D 自动巡测程序 (SL-AVR-2)

/******

项目 : auto35
版本 : 1.0
日期 : 2001-5-25
作者 : Will
广州市天河双龙电子有限公司

芯片型号 : AT90S8535
时钟频率 : 8.000000 MHz
存储器类型 : Small
内部 SRAM 大小 : 512
外部 SRAM 大小 : 0
数据堆栈大小 : 128

用 AT90S8535 作 0-7 通道 A/D 转换, 用 LED 显示, D7, D6 两位显示通道号, D3-D0 四位显示转换值(十六进制数 0-3FFH)。程序下载即执行, 自动从 0 通道到 7 通道 A/D 转换扫描显示, 当你按住 0-7 任一位数字键, 该通道转换值显示一段时间, 然后又自动循环显示。

本程序在 SL-AVR-2 上调试通过。

硬件接口:

AT90S8535 的 PC.1-4 接 SL7289 控制 LED 显示及键盘;
PA.0-7 接模拟电压;
AGND 接地;
AVCC 与 VREF 间接 1K 电阻,VRBF 到地接 100 μ F 电解电容;
AVC 与 VCC 间接一只 100 Ω 电阻,AVCC 接 104 瓷片电容到地。

*****/

```
#include <90s8535.h>
#include <delay.h>
unsigned int adc_data,temp2;
#define ADC_VREF_TYPE 0x00
/******
// ADC 中断服务程序
#pragma savereg-
interrupt [ADC_INT] void adc_isr(void)
{
#asm
    push r30
    push r31
#endasm
```



```
// 读取 ADC 转换结果
adc_data=ADCW;
#asm
    pop  r31
    pop  r30
#endasm
}
#pragma savereg+
//*****
//读取 ADC 转换结果（带噪声消除）
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;
#asm
    in   r30,mcucr
    sbr  r30,__se_bit
    cbr  r30,__sm_mask
    out  mcucr,r30
    sleep
    cbr  r30,__se_bit
    out  mcucr,r30
#endasm
return adc_data;
}
//*****
// 全局变量
unsigned char temp,temp1,temp3,sl,sh,ml,mh,hl,hh;
void transmit(unsigned char para) //发送子程序
{
    unsigned char i,j;
    PORTC.4=0; //SL279 片选有效
    delay_us(50);
    j=para;
    for(i=8;i>0;i--)
    {
        if((j&0x80)==0x80)
            PORTC.2=1; //数据端
        else
            PORTC.2=0;
            PORTC.3=1;
            delay_us(10);
            PORTC.3=0;
    }
}
```

```
    delay_us(10);  
    j=j<<1;  
  }  
}
```

```
void receive() //接收子程序  
{  
  unsigned char k;  
  DDRC.2=0;  
  delay_us(50);  
  for(k=8;k>0;k--)  
  {  
    PORTC.3=1; //时钟端  
    delay_us(10);  
    if(PINC.2==1)  
      temp|=0x01;  
    else  
      temp&=0xfe;  
    PORTC.3=0;  
    delay_us(10);  
    if(k>1)  
      temp=temp<<1;  
  }  
  DDRC.2=1;  
}
```

```
void init() //初始化子程序  
{  
  unsigned char l;  
  PORTC.4=1; //SL279 片选无效  
  PORTC.3=0; //时钟端  
  delay_ms(20);  
  transmit(0xc8); //D0 位-> 0;  
  delay_us(10);  
  transmit(0x00);  
  PORTC.4=1;  
  delay_ms(10);  
  transmit(0xc9); //D1 位-> 0;  
  delay_us(10);  
  transmit(0x00);  
  PORTC.4=1;  
  delay_ms(10);
```

```

transmit(0xca);           //D2 位-> 0;
delay_us(10);
transmit(0x00);
transmit(0xcb);         //D3 位-> 0;
delay_us(10);
transmit(0x00);
PORTC.4=1;
delay_ms(10);
transmit(0xe0);        //D4 位-> -;
delay_us(10);
l=0x00;
l+=(0x08+0x08+0x08+0x08);
transmit(l);
PORTC.4=1;
delay_ms(10);
l+=0x08;
transmit(0xe0);        //D5 位-> -;
delay_us(10);
transmit(l);
PORTC.4=1;
delay_ms(10);
transmit(0xce);        //D6 位-> 0;
delay_us(10);
transmit(0x00);
PORTC.4=1;
delay_ms(10);
transmit(0xcf);        //D7 位-> 0;
delay_us(10);
transmit(0x00);
PORTC.4=1;
delay_ms(10);
transmit(0x88);        //去闪烁;
delay_us(10);
transmit(0xff);
PORTC.4=1;
delay_ms(10);
}
//显示子程序*****
void disp(unsigned char ls,unsigned char hs, unsigned char lm,unsigned char
hm,unsigned char lh,unsigned char h)
{ unsigned char sll,shh,mll,mhh,hll,hhh;
  sll=ls;

```

```
shh=hs;
mll=lm;
mhh=hm;
hll=lh;
hhh=h;
transmit(0xc8);           //D0 位显示;
    delay_us(10);
transmit(sll);
    PORTC.4=1;
    delay_ms(10);
transmit(0xc9);           //D1 位显示;
    delay_us(10);
transmit(shh);
    PORTC.4=1;
    delay_ms(10);
transmit(0xca);           //D2 位显示;
    delay_us(10);
transmit(mll);
    PORTC.4=1;
    delay_ms(10);
transmit(0xcb);           //D3 位显示;
    delay_us(10);
transmit(mhh);
    PORTC.4=1;
    delay_ms(10);
transmit(0xce);           //D6 位显示;
    delay_us(10);
transmit(hll);
    PORTC.4=1;
    delay_ms(10);
transmit(0xcf);           //D7 位显示;
    delay_us(10);
transmit(hhh);
    PORTC.4=1;
    delay_ms(10);
}
```

```
void conver(unsigned char mid) //转换子程序
{ unsigned char k,p;
  p=mid;
  read_adc(p);
  #asm("cli")
```

```
temp1=(unsigned char)adc_data; //低位字节;
temp2=(0xff00&adc_data); //int
for(k=0;k<8;k++)
{
temp2=temp2>>1;
}
temp3=(unsigned char)temp2; //高位字节;
sl=(0x0f&temp1); //送 D0 位;
sh=(0xf0&temp1);
for(k=0;k<4;k++)
{
sh=sh>>1; //送 D1 位;
}
ml=(0x0f&temp3); //送 D2 位;
mh=(0xf0&temp3);
for(k=0;k<4;k++)
{
mh=mh>>1; //送 D3 位;
}
hl=(0x0f&p); //送 D6 位;
hh=(0xf0&p);
for(k=0;k<4;k++)
{
hh=hh>>1; //送 D7 位;
}
disp(sl,sh,ml,mh,hl,hh);
delay_ms(1000);
}
//主程序*****
void main(void)
{ unsigned char m,s; //局部变量
//Port A
DDRA=0x00;
PORTA=0x00;
// Port C
DDRC=0xFF;
PORTC=0x00;
// 外部中断初始化
//INT0: 关闭
//INT1: 关闭
GIMSK=0x00;
```

```
MCUCR=0x00;
init();
delay_ms(500);
while(1)
{
    s=0x00;
// ADC 初始化
// ADC 时钟频率 : 1000.000 kHz
    ADCSR=0x8B;
    for(m=0;m<8;m++,s++)           //自动巡测 8 路
    {
        #asm("sei")                // 全局中断允许
        conver(s);
        if(PINC.1==0)              //有键按下
        {
            transmit(0x15);        //读键
            receive();
            PORTC.4=1;
            delay_us(10);
            switch(temp)           //根据键号作转换
            {
            case 0:
                {
                    #asm("sei")    // 全局中断允许
                    conver(temp);
                    delay_ms(1500);
                };
                break;
            case 1:
                {
                    #asm("sei")    // 全局中断允许
                    conver(temp);
                    delay_ms(1500);
                };
                break;
            case 2:
                {
                    #asm("sei")    // 全局中断允许
                    conver(temp);
                    delay_ms(1500);
                };
                break;
            }
        }
    }
}
```

```
case 3:
    {
        #asm("sei")           // 全局中断允许
        conver(temp);
        delay_ms(1500);
    };
    break;

case 4:
    {
        #asm("sei")           // 全局中断允许
        conver(temp);
        delay_ms(1500);
    };
    break;

case 5:
    {
        #asm("sei")           // 全局中断允许
        conver(temp);
        delay_ms(1500);
    };
    break;

case 6:
    {
        #asm("sei")           // 全局中断允许
        conver(temp);
        delay_ms(1500);
    };
    break;

case 7:
    {
        #asm("sei")           // 全局中断允许
        conver(temp);
        delay_ms(1500);
    };
    break;

default: break;
    }
}

};
}
```