

## 第十章 IccAVR C 语言例子程序源程序

第十章 IccAVR C 语言的应用,本章例子程序编号为对应第九章 IARAVR C 语言的例子程序编号, 这些例子程序硬件接口见<<第九章 IARAVR C 语言的应用>>;第十章编号 10.5.1-9 为新增例子程序,这些例子程序均在 SL-AVR 开发实验器上验证通过。

### // 第十章例子 10.3.1 读/写口

```
#include <io8515.h>
void main(void)
{
    unsigned char  achar;
    DDRB = 0xFF;
    DDRD = 0x00;
    PORTD= 0x00;
    for(;;)
    {
        achar = PIND;
        PORTB = achar;
    }
}
```

### // 第十章例子 10.3.2 延时函数

```
#include <io8515.h> /* 定义 8515 */
void delay(int delayValue)
{
    int i;
    for(i=0;i<delayValue;i++) ;
}
void main(void)
{
    unsigned char runner = 0x01;
    DDRB = 0xff; /* Port B 输出 */
```

```
for (;;) /* 死循环 */
{
    if (runner) runner <<= 1;
    else runner = 0x01;
    PORTB = runner;
    delay(32767);
}
}
```

// 第十章例子 10.3.3 读/写 EEPROM

```
// These work for devices with more than 256 bytes of EEPROM
//int EEPROMwrite( int location, unsigned char);
//unsigned char EEPROMread( int);
#include <io8515.h>
#include <eeprom.h>
void main(void)
{
    unsigned char temp =0xaa,i;
    EEPROMwrite(0x20,temp); /* 写 EEPROM 地址 0x20 */
    i=EEPROMread(0x20);    /* 读 EEPROM 地址 0x20 */
    i++;
    EEPROMwrite(0x30,i);
}
}
```

// 第十章例子 10.3.4 AVR 的 PB 口变速移位

```
#include <io8515.h>
#define BIT(x) (1<<(x))
```

```
void delay(void)
{
    unsigned char i,j;
        for (i=1;i;i++)
            for(j=1;j;j++);
}
void led_pb(void)
{
    unsigned char i;
    DDRB=0xff;
    for (i=0;i<8;i++)
    {
        PORTB=~BIT(i);
        delay();
    }
}
void main (void)
{
    while (1)
        led_pb();
}
```

// 第十章例子 10.3.5 4 个口 LED 亮灯变速移位

```
#include <io8515.h>
#define BIT(x) (1<<(x))
void delay(unsigned char delayValue)
{
    unsigned char i,j;
```

```
        for(i=0;i<delayValue;i++)
            for (j=1;j;j++)
                ;
    }
void led_pb(unsigned char t)
{
    unsigned char i;
    DDRB=0xff;
    for (i=0;i<8;i++)
    {
        PORTB=~BIT(i);
        delay(t);
    }
    PORTB=0xff;
}
void led_pd(unsigned char t)
{
    unsigned char i;
    DDRD=0xff;
    for (i=0;i<8;i++)
    {
        PORTD=~BIT(i);
        delay(t);
    }
    PORTD=0xff;
}
void led_pc(unsigned char t)
{
    unsigned char i;
```

```
DDRC=0xff;
for (i=0;i<8;i++)
{
    PORTC=~BIT(i);
    delay(t);
}
PORTC=0xff;
}
void led_pa(unsigned char t)
{
    unsigned char i;
    DDRA=0xff;
    for (i=8;i>0;i--)
    {
        PORTA=~BIT(i-1);
        delay(t);
    }
    PORTA=0xff;
}
void main (void)
{
    unsigned char dt;
    while (1)
    {
        for (dt=5;dt<200;dt+=25)
        {
            led_pb(dt);
            led_pd(dt);
            led_pc(dt);
```

```
        led_pa(dt);
    }
}
}
```

// 第十章例子 10.3.6 音符声程序

```
#include <io8515.h>          /*预处理命令 */
#define uchar unsigned char
#define uint unsigned int
void delay(uchar t)
{
    uchar i,j;
    for (i=0;i<t;i++)
        for(j=1;j<150;j++);
}
void sound_pc0(uchar t)
{
    uint i;
    DDRC=0xff;
    PORTC=0xff;
    for (i=0;i<350-t*t;i++)
    {
        PORTC^=0x01;
        delay(t);
    }
}
void main (void)
{
```

```
uchar dt;
for(;;)
{
    for(dt=1;dt<14;dt++)
        sound_pc0(dt);
}
```

// 第十章例子 10.3.7 8 字循环移位显示程序

```
#include <io8515.h>
#define uchar unsigned char
#define uint unsigned int
void delay(uint t)
{
    uint i;
    for (i=0;i<t;i++)
        ;
}
void init_disp(void)
{
    DDRB=0xff;
    DDRD=0xff;
    PORTB=0x7f;
}
void scan(void)
{
    uchar i,j;
    for (i=0;i<6;i++)
```

```
{
    j=150;
    do
    {
        PORTD=~(0x01<<i);
        delay(150);
        PORTD=0xff;
        delay(2100);
    }
    while(--j);
}
}
void main(void)
{
    init_disp();
    for(;;)
    scan();
}
```

// 第十章例子 10.3.8 按键加计数显示程序

```
#include <io8515.h>
#define uchar unsigned char
#define uint unsigned int
const uchar DATA_7SEG[ ] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,
                             0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71}; /*LED 字形表*/

uchar led[6] ;
uint count;
void delay(uint t)
{
```



```
uint i;
for (i=0;i<t;i++)
    ;
}
void init_disp(void)
{
    DDRB=0xff;
    DDRD=0x7f;
    PORTD|=0x80;
}
void disp(void)
{
    uchar i;
    for (i=0;i<6;i++)
    {
        PORTD=~(0x01<<i);
        PORTB=DATA_7SEG[led[i]];
        delay(1000);
    }
    PORTB=0x00;
    PORTD=0xff;
}
void be_pc0(void)
{
    uint i;
    DDRC|=0x01;
    for (i=0;i<350;i++)
    {
        PORTC^=0x01;
```

```
        delay(350);
    }
}
void conv(void)
{
    led[5] =0;
    led[4] =count/10000;
    led[3] =count/1000%10;
    led[2] =count/100%10;
    led[1] =count/10%10;
    led[0] =count%10;
}
void main(void)
{
    init_disp();
    count=0;
    conv();
    for(;;)
    {
        while((PIND&0x80)==0x80)
            disp();
        be_pc0();
        count++;
        conv();
        while((PIND&0x80)==0)
            disp();
    }
}
```

// 第十章例子 10.4.1 锯齿波程序

```
#include <io8515.h>
#define uchar unsigned char
void delay(void)
{
}
void main(void)
{
    uchar c;
    DDRA=0xff;
    for (;;)
    {
        PORTA=c++;
        delay();
    }
}
```

// 第十章例子 10.4.2 正三角波程序

```
#include <io8515.h>
#include <math.h>
#define uchar unsigned char
#define uint unsigned int
void delay(void)
{
}
void main(void)
```

```
{
uchar c;
DDRA=0xff;
for (;;)
{
for (c=0x00;c<0xff;c++)
{
PORTA=c;
delay();
}
for (c=0xff;c>0x00;c--)
{
PORTA=c;
delay();
}
}
}
```

// 第十章例子 10.4.3 梯形波程序

```
#include <io8515.h>
#define uchar unsigned char
#define uint unsigned int
void delay(uchar t)
{
uchar i;
for (i=0;i<t;i++)
;
}
}
```

```
void main(void)
{
  uchar c;
  DDRA=0xff;
  for (;;)
  {
    for (c=0x00;c<0xff;c++)
      PORTA=c;
    delay(255);
    for (c=0xff;c>0x00;c--)
      PORTA=c;
    delay(255);
  }
}
```

// 第十章例子 10.4.4 正弦波程序

```
#include <io8515.h>
#define uchar unsigned char
#define uint unsigned int
const uchar sintab[ ] =

{
  128,130,132,135,137,139,141,143,146,148,
  150,152,154,157,159,161,163,165,167,169,
  171,174,176,178,180,182,184,186,188,190,
  191,193,195,197,199,201,203,204,206,208,
  210,211,213,215,216,218,219,221,222,224,
  225,227,228,229,231,232,233,235,236,237,
```

238, 239, 240, 241, 242, 243, 244, 245, 246, 247,  
247, 248, 249, 249, 250, 251, 251, 252, 252, 253,  
253, 253, 254, 254, 254, 255, 255, 255, 255, 255,  
255, 255, 255, 255, 255, 255, 254, 254, 254, 253,  
253, 253, 252, 252, 251, 251, 250, 249, 249, 248,  
247, 247, 246, 245, 244, 243, 242, 241, 240, 239,  
238, 237, 236, 235, 233, 232, 231, 229, 228, 227,  
225, 224, 222, 221, 219, 218, 216, 215, 213, 211,  
210, 208, 206, 204, 203, 201, 199, 197, 195, 193,  
192, 190, 188, 186, 184, 182, 180, 178, 176, 174,  
171, 169, 167, 165, 163, 161, 159, 157, 154, 152,  
150, 148, 146, 143, 141, 139, 137, 135, 132, 130,  
128, 126, 124, 121, 119, 117, 115, 113, 110, 108,  
106, 104, 102, 99, 97, 95, 93, 91, 89, 87,  
85, 82, 80, 78, 76, 74, 72, 70, 68, 66,  
65, 63, 61, 59, 57, 55, 53, 52, 50, 48,  
46, 45, 43, 41, 40, 38, 37, 35, 34, 32,  
31, 29, 28, 27, 25, 24, 23, 21, 20, 19,  
18, 17, 16, 15, 14, 13, 12, 11, 10, 9,  
9, 8, 7, 7, 6, 5, 5, 4, 4, 3,  
3, 3, 2, 2, 2, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 2, 2, 2, 3,  
3, 3, 4, 4, 5, 5, 6, 7, 7, 8,  
9, 9, 10, 11, 12, 13, 14, 15, 16, 17,  
18, 19, 20, 21, 23, 24, 25, 27, 28, 29,  
31, 32, 34, 35, 37, 38, 40, 41, 43, 45,  
46, 48, 50, 52, 53, 55, 57, 59, 61, 63,  
64, 66, 68, 70, 72, 74, 76, 78, 80, 82,  
85, 87, 89, 91, 93, 95, 97, 99, 102, 104,

```
106,108,110,113,115,117,119,121,124,126,  
};  
void main(void)  
{  
    uint i;  
    DDRA=0xff;  
    for (;;)   
    {  
        for (i=0;i<360;i++)  
            PORTA=sintab[ i] ;  
    }  
}
```

// 第十章例子 10.4.5 方波程序

```
#include <io8515.h>  
#define uchar unsigned char  
#define uint unsigned int  
void delay(uchar t)  
{  
    uchar i,j;  
    for (i=0;i<t;i++)  
        for (j=1;j;j++)  
            ;  
}  
void main(void)  
{  
    DDRA=0xff;  
    PORTA=0xff;
```

```
for(;;)
{
    PORTA^=0x01;
    delay(25);
}
}
```

## 用双龙电子增补 IccAVR 库文件后新增例子

双龙电子在 IccAVR C 高级语言中增加了库文件 libslavr.a, 头文件 slavr.h

有关使用注意事项如下:

- 1、在使用例子程序之前, 将库文件 libslavr.a 拷贝到 ICCAVR 安装目录下的库文件目录中即...\LIB\目录。将头文件 slavr.h 拷贝到头文件目录即...\include\目录。
- 2、如果想使用库文件中的函数, 那么设置如下:
  - 1)、在文件中应使用 "#include <slavr.h>"包含 slavr.h 头文件。
  - 2)、在工程选项中的 Additional LIB 一栏中填入"slavr", 使用附加的 libslavr.a 库文件。(注意: 引号不需要输入)
- 3、libslavr.a 库中包含的函数清单如下:
  - 1)、void write\_lcd(unsigned char data,unsigned char data\_type);  
写命令或数据到 LCD 模块, 可直接使用控制命令对 LCD 模块操作  
data:命令或数据字节 ,data\_type: 0 表示命令, 1 表示数据
  - 2)、unsigned char read\_lcd(unsigned char data\_type);  
从 LCD 模块中读取数据或地址  
data\_type: 0 表示读取地址, 1 表示读取数据
  - 3)、void lcd\_init(void);  
LCD 模块初始化
  - 4)、void lcd\_putc(unsigned char c) ;  
单字符输出函数
  - 5)、void lcd\_puts(unsigned char \*s);



字符串输出函数

6)、 void lcd\_clear(void);

清除 LCD 显示函数

7)、 void lcd\_write(unsigned char adder,unsigned char data);

写数据到指定 DDRAM/CGRAM 位置

8)、 void lcd\_write\_data(unsigned char data);

写数据到 DDRAM 当前位置

9)、 unsigned char lcd\_read(unsigned char adder);

从指定 DDRAM/CGRAM 位置读取数据

10)、 void lcd\_gotoxy(unsigned char x,unsigned char y);

将光标转移到 DDRAM 指定位置处, x:0~39,y:0~1

11)、 void lcd\_shift(unsigned char n,unsigned char p);

LCD 字符移位函数,n:移动的次数,p:移动方向,'l'或'L'表示左移,'r'或'R'表示右移

12)、 void delay\_1ms(void);

13)、 void delay\_ms(unsigned int n);

延时函数(mS),对应晶振频率为 8MHZ

14)、 unsigned char scan\_key(void);

键盘扫描函数,注意本函数不作按键释放检查

返回值: 1、没有键按下返回 0x7f

2、如果 shift 按下,返回值最高位为 1

3、其余键按下时,返回键名对应数值

15)、 unsigned char keypad(void);

键盘扫描函数,注意本函数作按键释放检查

返回值: 1、没有键按下返回 0x7f

2、如果 shift 按下,返回值最高位为 1

3、其余键按下时,返回键名对应数值

16)、 void Start(void);

I2C 总线启动

17)、 void Stop(void);

I2C 总线停止

18)、 void Ack(void);

发送应答信号

19)、 void NoAck(void);

发送非应答信号

20)、 unsigned char TestAck(void);

测试应答信号，有应答信号返回 0，无应答信号返回 01。

21)、 void Write8Bit(unsigned char input);

写一个字节数据到 I2C 总路线

22)、 unsigned char Read8Bit(void);

从 I2C 总线读取一个字节数据

23)、 void Write24c02(unsigned char \*Wdata,unsigned char RomAddress,unsigned char number);

对串行 EEPROM 存储器 AT24C02 采用页写模式连续写多个字节

参数说明: unsigned char \*Wdata , 指向源数据块首地址的指针

unsigned char RomAddress, EEPROM 中的目标地址

unsigned char number, 连续写的字节数

24)、 void Read24c02(unsigned char \*RamAddress,unsigned char RomAddress,unsigned char bytes);

对串行 EEPROM 存储器 AT24C02 连续读多个字节

参数说明: unsigned char \*RamAddress, 指向存放读出数据的变量的指针

unsigned char RomAddress, EEPROM 中的目标地址

unsigned char bytes, 连续读的字节数

#### 4、例子说明

10.5.1)、 LED ICCAVR 中的典型例子，可作为 ICCAVR 的起步

10.5.2)、 interrupt 使用定时器 1 溢出中断的例子

10.5.4)、 display SL-AVR 实验仪中六路 LED 进行动态显示的例子程序

keyboard 包含三个例子程序 keyboard.c、 keyboard1.c 和 keyboard2.c

10.5.5) 、 keyboard.c : 使用 keypad 函数，观察其对 LED 显示有何影响

10.5.5A) 、 keyboard1.c: 使用 scan\_key 函数，观察其对 LED 显示有何影响

10.5.5B) 、 keyboard2.c: 使用 keypad 函数，观察其对 LCD 显示有何影响

## 第十章 IccAVR C 语言例子程序源程序

10.5.6)、music 电脑音乐演示程序----播放"卡秋莎"音乐

10.5.7)、clock 电脑时钟程序----24 小时制时钟

10.5.8)、cymometer 单量程频率计，测量范围 0-999999 HZ

I2C AVR 单片机使用 I2C 总线的例子，AVR 工作于 I2C 的主节点模式，包含两个例子

10.5.9)、I2C.c 以单字节模式访问 I2C 总路线器件 AT24C02 的例子程序，从中可以学习 I2C 总线的工作过程

10.5.9A) 、AT24C02.c 调用库函数对 AT24C02 进行读写，可以连续多字节读写。

**注意：**a、以上程序中部分例子程序使用了 SL-AVR 专用库 libslavr.a，其中有关函数原型说明参考头文件 slavr.h。

b、上述例子程序的硬件接线如果没特殊说明，采用双龙电子出品的 SL-AVR 实验仪的出厂默认跳线。

// 第十章例子 10.5.1)、LED ICCAVR 中的典型例子，可作为 ICCAVR 的起步

```
/******  
  
http://www.sl.com.cn  
双龙电子公司(ICCAVR)  
*****/  
  
#include <io8515.h>  
/******  
为了可以看见 LED 的图案变化，每次 LED 图案变化需要保持一定时间  
*****/  
  
void Delay()  
{  
    unsigned char a, b;  
  
    for (a = 1; a; a++)  
        for (b = 1; b; b++)  
            ;  
}
```

```
void LED_On(int i)
{
    PORTB=~(1<<i); /* 输出低电平, 使 LED 发光 */
    Delay();
}

void main()
{
    int i;
    DDRB = 0xFF; /* 定义 B 口为输出 */
    PORTB = 0xFF; /* 关闭全部 LED */

    while (1)
    {
        /* 往前步进 */
        for (i = 0; i < 8; i++)
            LED_On(i);
        /* 往后步进 */
        for (i = 8; i > 0; i--)
            LED_On(i);
        /* 跳跃 */
        for (i = 0; i < 8; i += 2)
            LED_On(i);
        for (i = 7; i > 0; i -= 2)
            LED_On(i);
    }
}
```

// 第十章例子 10.5.2)、interrupt 使用定时器 1 溢出中断的例子

```
/******  
    http://www.sl.com.cn  
    双龙电子公司----中断演示程序(ICCAVR)  
*****/  
#include <io8515.h>  
#pragma interrupt_handler timer:T0V1  
  
char led_image = 0xFF; // 关闭全部 LED  
    主程序入口  
*****/  
void main()  
{  
/******  
    初始化 8515 寄存器  
*****/  
    DDRB = 0xFF;  
    PORTB = 0x00;  
    TCCR1A=0;  
    TCCR1B = 0x00;        // 停止定时器 1  
    TCNT1H = 0x00;        // 清除定时器 1  
    TCNT1L = 0x00;  
    TIMSK=0x80;          // 开放定时器 1 溢出中断  
    SREG|=0x80;  
    TCCR1B =0x03;        // 启动定时器 1 预分频比例 64  
    while (1)  
    ;  
}
```

```
/******  
    在定时器中断中改变 LED 的显示状态  
*****/  
void timer()  
{  
    PORTB = ++led_image;  
    PORTC=~PORTC;  
}  
  
// 第十章例子 10.5.3)、LCD SL-AVR 实验仪中的 2*16 位 LCD 模块的使用例子  
/******  
    http://www.sl.com.cn  
    双龙电子公司----2*16 LCD 控制演示程序(ICCAVR)  
    作者：詹卫前  
    使用注意：1、应包含头文件 slavr.h  
              2、在工程选项中的 Additional Lib 中填入 slavr  
              使用编译器使用附加库文件 libslavr.a  
*****/  
#include <io8515.h>  
#include <slavr.h>  
void main(void)  
{  
    MCUCR=0;  
    lcd_init();  
    while(1)  
    {  
        lcd_gotoxy(2,0);  
        lcd_puts("Hello World!");  
        delay_ms(2000);  
    }  
}
```

```
    lcd_clear();
    lcd_gotoxy(4,0);
    lcd_puts("HTTP://");
    lcd_gotoxy(1,1);
    lcd_puts("www.sl.com.cn");
    delay_ms(2000);
    lcd_clear();
}
}
```

// 第十章例子 10.5.4)、display SL-AVR 实验仪中六路 LED 进行动态显示的例子程序

```
/* **** */
/* 延时函数(ms),在本文件中应定义:xtal?*/
/* xtal 为对应晶振频率,单位为 MHZ */
/* 在调用延时函数的文件中,应作如下说明 */
/* extern void delay_ms(unsigned int n);*/
/* extern void delay_1ms(void); */
/* **** */

#define xtal 8
void delay_1ms(void)
{ unsigned int i;
  for(i=1;i<(unsigned int)(xtal*143-2);i++)
    ;
}
void delay_ms(unsigned int n)
{
  unsigned int i=0;
  while(i<n)
```

```

    {delay_1ms();
      i++;
    }
  }
}

```

// 第十章例子 10.5.5) 、 keyboard.c : 使用 keypad 函数, 观察其对 LED 显示有何影响

```

/*****

```

HTTP://WWW.SL.COM.CN

双龙电子

- 使用注意:
- 1、应包含头文件 slavr.h
  - 2、在工程选项中的 Additional Lib 中填入 slavr  
使用编译器使用附加库文件 libslavr.a
  - 3、本例采用了作按键释放检查的函数, 适用于显示电路不需频繁刷新的程序, 如静态显示、LCD 模块电路等。

```

*****/

```

```

#include <io8515.h>

```

```

#include <slavr.h>

```

```

//字形表

```

```

const                unsigned                char

```

```

tabel[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};

```

```

//显示缓冲区

```

```

unsigned char ledbuff[]={0x3f,0x3f,0x3f,0x3f,0x3f,0x3f};

```

```

void port_init(void)//端口初始化

```

```

{

```

```

    DDRC=0x0f;

```



```
    DDRD=0x3f;
    DDRB=0xff;
    PORTD=0xff;
    PORTB=0xff;
    PORTC=0xff;
}
void display(void)//六路动态扫描显示
{
    unsigned char i;
    for (i=0;i<6;i++)
    {
        PORTB=ledbuff[i];
        PORTD=~(1<<i);
        delay_1ms();
        PORTD|=(1<<i);
    }
}
/*主程序的功能：读取键盘扫描码，转换成相应的字形码后*/
/*送入显示缓冲区进行显示，显示部分参考 display.c */
/*如果 shift 键按下，则全部清 0 */
void main(void)
{
    unsigned char keyval1;
    unsigned char i=0;
    MCUCR=0;
    port_init();
    while(1)
    {
        keyval1=keypad();//键盘扫描函数，作按键释放检查
```

```

if (keyval1!=0x7f)
{
    if ((keyval1&0x80)==0x80)
    {
        for(i=0;i<6;i++)
            ledbuff[i]=tabel[0];
    }
    else
    {for (i=0;i<5;i++)
    {
        ledbuff[5-i]=ledbuff[4-i];//每按一次键，左移一位
    }
        ledbuff[0]=tabel[keyval1&0x7f];
    }
}
display();
}
}

```

// 第十章例子 10.5.5A) 、 keyboard1.c: 使用 scan\_key 函数，观察其对 LED 显示有何影响

/\*\*\*\*\*\*

HTTP://WWW.SL.COM.CN

双龙电子

- 使用注意:
- 1、应包含头文件 slavrr.h
  - 2、在工程选项中的 Additional Lib 中填入 slavrr  
使用编译器使用附加库文件 libslavrr.a
  - 3、本例采用了不作按键释放检查的函数，适用于显示电路需要频繁刷新 或占用 CPU 时间较多的程序，如动态显示电路。

广州天河双龙电子有限公司 <http://WWW.SL.COM.CN>

```
*****/  
#include <io8515.h>  
#include <slavr.h>  
//字形表  
const unsigned char  
tabel[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};  
//显示缓冲区  
unsigned char ledbuff[]={0x3f,0x3f,0x3f,0x3f,0x3f,0x3f};  
  
void port_init(void)//端口初始化  
{  
    DDRC=0x0f;  
    DDRD=0x3f;  
    DDRB=0xff;  
    PORTD=0xff;  
    PORTB=0xff;  
    PORTC=0xff;  
}  
void display(void)//六路动态扫描显示  
{  
    unsigned char i;  
    for (i=0;i<6;i++)  
    {  
        PORTB=ledbuff[i];  
        PORTD=~(1<<i);  
        delay_1ms();  
        PORTD|=(1<<i);  
    }  
}
```

```
/*主程序的功能：读取键盘扫描码，转换成相应的字形码后*/
/*送入显示缓冲区进行显示，显示部分参考 display.c */
/*如果 shift 键按下，则全部清 0 */
void main(void)
{
    unsigned char keyval1,keyval2;
    unsigned char i=0;
    MCUCR=0;
    port_init();
    while(1)
    {
        keyval1=scan_key();//键盘扫描函数，不作按键释放检查
        if (keyval1!=0x7f)
        {
            do
            {
                keyval2=scan_key();
                display();
            }
            while (keyval1==keyval2);//作按键释放检查

            if ((keyval1&0x80)==0x80)
            {
                for(i=0;i<6;i++)
                    ledbuff[i]=tabel[0];
            }
            else
            {for (i=0;i<5;i++)
                {
```

```

        ledbuff[5-i]=ledbuff[4-i]; //每按一次键，左移一位
    }
    ledbuff[0]=tabel[keyval&0x7f];
}
}
display();
}
}

```

// 第十章例子 10.5.5B) 、 keyboard2.c: 使用 keypad 函数，观察其对 LCD 显示有何影响

```

/*****

```

HTTP://WWW.SL.COM.CN

使用注意: 1、应包含头文件 slavr.h

2、在工程选项中的 Additional Lib 中填入 slavr  
使用编译器使用附加库文件 libslavr.a

3、本例采用了作按键释放检查的函数，适用于显示电路不需频繁刷新的程序，如静态显示、LCD 模块电路等。

```

*****/

```

```

#include <io8515.h>
#include <slavr.h>
void main(void)
{
    unsigned char keyval,i;
    lcd_init();
    lcd_gotoxy(1,0);
    lcd_puts("www.sl.com.cn");
    lcd_gotoxy(0,1);

```

```
lcd_puts("INPUT: ");
while(1)
{
    keyval=keypad();
    if (keyval<16)//对应 0~F 的键盘输入
    {
        if (keyval<10)
            keyval=0x30+keyval;
        else
            keyval=55 +keyval;
        lcd_write_data(keyval);
    }
    if((keyval&0x80)==0x80)//对应 SHIF 的键盘输入
    {
        for(i=0;i<7;i++)
        {
            lcd_shift(1,'r');
            delay_ms(500);
        }
        for(i=0;i<7;i++)
        {
            lcd_shift(1,'l');
            delay_ms(500);
        }
    }
}
}
```

// 第十章例子 10.5.6)、music 电脑音乐演示程序----播放"卡秋莎"音乐

```

/*****
    http://www.sl.com.cn
    双龙电子公司----电脑音乐演示程序(ICCAVR)
    作者: 詹卫前
    1、学习定时器 T0 溢出中断的使用
    2、学习定时器 T1 比较中断的使用
    3、学习电脑音乐的产生
    *****/

#include <io8515.h>
#include <macros.h>
#pragma interrupt_handler timer0:8
#pragma interrupt_handler timer1:5
#pragma data:code
flash unsigned int music_data[][2]=
/*****
    卡秋莎音乐数据{x,y}
    x:对应音符音阶(频率),0表示休止符
    y:对应音符节拍
    *****/

{
{440,600} ,{494,200} ,{523,600} ,{440,200} ,
{523,400} ,{494,200} ,{440,200} ,{494,400} ,
{330,400} ,{494,600} ,{523,200} ,{578,600} ,
{494,200} ,{578,400} ,{523,200} ,{494,200} ,
{440,800} ,{659,400} ,{880,400} ,{784,400} ,
{880,200} ,{784,200} ,{698,400} ,{659,200} ,
{578,200} ,{659,400} ,{440,400} ,{ 0,200} ,//休止符
{698,400} ,{578,200} ,{659,600} ,{523,200} ,

```

```
{494,200} ,{330,200} ,{523,200} ,{494,200},
{440,800} ,{659,400} ,{880,400} ,{784,400},
{880,200} ,{784,200} ,{698,400} ,{659,200},
{578,200} ,{659,400} ,{440,400} ,{ 0,200} ,//休止符
{698,400} ,{578,200} , {659,600} ,{523,200},
{494,200} ,{330,200}, {523,200} ,{494,200},
{440,800} ,
{0,0}};
#pragma data:data
unsigned int delay=0;
/*****
    MCU 初始化
*****/
void music_init(void)
{
    MCUCR=0x00;
    DDRC=0x01;
    TCCR1A=0x00;
    TCCR1B=0x09;
    TCCR0=0x03;
    TCNT0=0x19;
    TIMSK=0x42;
}
/*****
    T0 中断程序, 产生音乐节拍
*****/
void timer0(void)
{
    delay++;
```



```
    TCNT0=0x19;
}
/*****
    T1 中断程序, 根据 SOUND 函数输出一定频率的方波
*****/
void timer1(void)
{
    PORTC^=0x01;
}
/*****
    SOUND 程序, 输出频率为 x HZ 的方波, 延时 y MS
    x: 100~20000 HZ, 0 表示不发声
    y: 0~65536 MS
*****/
void sound(unsigned int x, unsigned int y)
{
    SEI();
    delay=0;
    if (x!=0)
    {
        x=4000000/x;
        OCR1A=x;
        TCNT1=0x00;
        TIMSK|=0x40;
        while(delay<y)
        ;
        TIMSK&=0xbf;
    }
    else
```

```
    {
        TIMSK&=0xbf;
        while(delay<y)
        ;
    }
    CLI();

}

/*****
主程序
*****/
void main(void)
{
    unsigned char i=0;
    music_init();
    while(1)
    {
        while(music_data[i][1]!=0)
        {
            sound(music_data[i][0],music_data[i][1]);
            i++;
        }
        i=0;
    }
}
```

//第十章例子 10.5.7)、clock 电脑时钟程序----24 小时制时钟

广州天河双龙电子有限公司 <http://WWW.SL.COM.CN>

```

/*****
    http://www.sl.com.cn
    双龙电子公司----电脑时钟演示程序(ICCAVR)
    作者: 詹卫前
    1、学习定时器 T1 溢出中断的使用
    2、学习电脑时钟的编程方法
    3、在程序运行时按 SHIFT 进行校时
    使用注意: 1、应包含头文件 slavr.h
               2、在工程选项中的 Additional Lib 中填入 slavr
               使用编译器使用附加库文件 libslavr.a
    *****/

#include <io8515.h>
#include <macros.h>
#include <slavr.h>
#pragma interrupt_handler timer1:7
#pragma data:code
    /* 七段译码字形表 */
const unsigned char
tabel[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
#pragma data:data
    /* 显示缓冲区 */
unsigned char ledbuff[]={0x3f,0x3f,0x3f,0x3f,0x3f,0x3f};
unsigned char time[3]={0,0,0};
unsigned char flag=0;
/*****
    初始化
    *****/
void port_init(void)

```

```
{
  DDRD=0x3f;
  DDRB=0xff;
  PORTD=0xff;
  PORTB=0xff;
  TIMSK=0x80; //开放 T1 溢出中断
  TCCR1A=0x00;
  TCCR1B=0x04;
  TCNT1=0x85ee;
  SEI();
}
/*****
          刷新 LED 显示缓冲区
*****/
void hextobcd (void)
{
  unsigned char i,temp;
  for (i=0;i<3;i++)
  {
    temp=temp%10;
    ledbuff[5-2*i]=tabel[temp];
    temp=time[i]%10;
    ledbuff[4-2*i]=tabel[temp];
  }
}
/*****
          六路动态扫描显示电路
*****/
void display(void)
```

```
{
unsigned char i;
for (i=0;i<6;i++)
{
PORTB=ledbuff[i];//将显示缓冲区数据送 PORTB 口
PORTD=~(1<<i); //开始显示
delay_1ms(); //每一位显示保持一定时间
PORTD|=(1<<i); //关闭显示
}
}
/*****
          设定校时时间
*****/
void settime(void)
{
unsigned char t1,t2,t3;
for (t1=0;t1<3;t1++)
{
ledbuff[5-2*t1]|=0x80;
do
{
t2=scan_key();
display();
}
while(t2==0x7f);
do
{
t3=scan_key();
display();
```

```
    }
    while (t2==t3);
    time[t1]=time[t1]%10+t2*10;
    hextobcd();
    ledbuff[5-2*t1]&=0x7f;
    ledbuff[4-2*t1]|=0x80;
    do
    {
        t2=scan_key();
        display();
    }
    while(t2==0x7f);
    {
        t3=scan_key();
        display();
    }
    while (t2==t3);
    t3=time[t1]/10;
    time[t1]=t3*10+t2;
    ledbuff[4-2*t1]&=0x7f;
    hextobcd();
}
}
/*****
主程序功能：演示了电脑时钟程序的工作过程
*****/
void main(void)
{
    unsigned char i,j;
```

```
MCUCR=0;
flag=0;
port_init(); //端口初始化
i=scan_key();
while(1)
{
    i=scan_key();
    if (i!=0x7f)
    {
        do
        {
            j=scan_key();
            display();
        }
        while (i==j);
        if ((i&0x80)==0x80)
            settime();//设定时间
    }
    if(flag==1)
    {
        flag=0;
        if (time[2]>=60)
        {
            time[2]-=60;
            time[1]++;
        }
        if (time[1]>=60)
        {
            time[1]-=60;
```

```
time[0]++;
}
if (time[0]>=24)
{
time[0]-=24;
}
hextobcd();
}
display();
}
}
/*****
          T1 溢出中断
*****/
void timer1(void)
{
flag=1;
time[2]++;
TCNT1=0x85ee;
}

// 第十章例子 10.5.8)、cymometer 单量程频率计，测量范围 0-999999 HZ
/*****
          http://www.sl.com.cn
          双龙电子公司----频率计演示程序(ICCAVR)
          作者：詹卫前
          1、学习定时器 T1 捕捉(捕获)中断的使用
          2、学习频率计的设计方法
          频率计编程。
*****/
```



```
*****/
#include <io8515.h>
#include <macros.h>
#pragma interrupt_handler timer0:8
#pragma interrupt_handler timer1:4
extern void delay_ms(unsigned int n); //引用外部函数
#pragma data:code
        /* 七段译码字形表 */
const                unsigned                char
tabel[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
#pragma data:data
        /* 显示缓冲区 */
unsigned char ledbuff[]={0x3f,0x3f,0x3f,0x3f,0x3f,0x3f};
unsigned long count; //频率测量计数变量
unsigned int timecount; //频率测量闸门控制变量
/*****
        初始化
*****/
void port_init(void)
{
    MCUCR=0x00;
    DDRD=0x3f;
    DDRB=0xff;
    PORTD=0xff;
    PORTB=0xff;
    TIMSK=0x0a; //开放 T0 溢出中断和 T1 捕捉中断
    TCCR0=0x03;
    TCCR1A=0x00;
    TCCR1B=0xc0;

```

```
TCNT0=0x83;
timecount=1000;
}
/*****
    六路动态扫描显示电路
*****/
void display(void)
{
    unsigned char i;
    for (i=0;i<6;i++)
    {
        PORTB=ledbuff[i]; //将显示缓冲区数据送 PORTB 口
        PORTD=~(1<<i);    //开始显示
        delay_1ms();      //每一位显示保持一定时间
        PORTD|=(1<<i);    //关闭显示
    }
}
/*****
    将 count 十六进制数据转换为 LED 七段码
*****/
void hextobcd(void)
{
    unsigned char i,temp;
    for (i=0;i<6;i++)
    {temp=count%10;
      ledbuff[i]=tabel[temp];
      count=count/10;
    }
}
```

```
/******  
主程序  
*****/  
void main(void)  
{  
    unsigned char i,j;  
    MCUCR=0;  
    port_init(); //端口初始化  
    count=0;  
    SEI();  
    while(1)  
    {  
        display();  
    }  
/******  
T0 溢出中断，完成测量闸门控制和数字转换  
*****/  
void timer0(void)  
{  
    if (--timecount==0)  
    {  
        CLI();  
        hextobcd();  
        if (count>999999)  
            ledbuff[0]|=0x80;  
        count=0;  
        timecount=1000;  
        SEI();  
    }  
}
```

```

    TCNT0=0x84;
}
/*****
T1 捕捉中断，完成脉冲计数
*****/
void timer1(void)
{
    count++;
}

// 第十章例子 10.5.9)、I2C.c 以单字节模式访问 I2C 总路线器件 AT24C02 的例子程序
/*****
    http://www.sl.com.cn
    双龙电子公司----I2C 总线演示程序 (ICCAVR)
    作者：詹卫前
    1、以访问 AT24C02 为例
    2、SDA : PORTA.0
    3、SCL : PORTA.1
    使用注意：1、应包含头文件 slavrr.h
               2、在工程选项中的 Additional Lib 中填入 slavrr
               使用编译器使用附加库文件 libslavr.a
*****/
#include <io8515.h>
#include <slavr.h>
    /* 七段译码字形表 */
#pragma data:code
const                unsigned                char
tabel []={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
#pragma data:data

```

```
/*      显示缓冲区      */
unsigned char ledbuff[]={0x3f,0x3f,0x3f,0x3f,0x00,0x00};
unsigned char at24_add,at24_data;
/*****

                端口初始化
                *****/
void port_init(void)
{
    DDRD=0x3f;
    DDRB=0xff;
    PORTD=0xff;
    PORTB=0xff;
}
/*****

                六路动态扫描显示电路
                *****/
void display(void)
{
    unsigned char i;
    i=at24_data%16;
    ledbuff[0]=tabel[i];
    i=at24_data/16;
    ledbuff[1]=tabel[i];
    i=at24_add%16;
    ledbuff[2]=tabel[i];
    i=at24_add/16;
    ledbuff[3]=tabel[i];
    for (i=0;i<6;i++)
    {
```

```

PORTB=ledbuff[i]; //将显示缓冲区数据送 PORTB 口
PORTD=~(1<<i);   //开始显示
delay_1ms();     //每一位显示保持一定时间
PORTD|=(1<<i);   //关闭显示
}
}
/*****

```

主程序功能：演示了读写 AT24C02 的过程

说明：1、键盘上 0~F 键用于输入地址或数据，

SHIFT 键用于切换状态

2、上电复位后，按数字键只可以修改地址，程序自动显示 AT24C02 单元内容。

3、按一次 SHIFT 键，最高位 LED 小数点亮此时可以修改数字，再按一次 SHIFT 键程序将修改后的数字写入 AT24C02，然后回到上电复位后的状态

```

*****/

```

```

void main(void)
{
    unsigned char key1,key2;
    unsigned char index,flag;
    at24_add=0;
    at24_data=0;
    index=0;
    flag=0;
    port_init(); //端口初始化
    while(1)
    {
        key1=scan_key();

```

```
if (key1!=0x7f)
{
do
{
key2=scan_key();//检查按键释放
display();
}
while(key1==key2);
{
if (flag!=0x00)//SHIFT 键切换数据和地址修改
{
Write24c02(&at24_data,at24_add,1);
flag=0x00;
}
else flag=0x80;
ledbuff[5]=0x00^flag;
index=0;
}
else
{
switch (index)
{
case 0:{if (flag==0x00)
{key2=at24_add&0x0f;
at24_add=key2|(key1<<4);
}
else
{key2=at24_data&0x0f;
at24_data=key2|(key1<<4);
```

```
        }
        index=01;
        break;
    }
    case 1:{if (flag==0x00)
        {key2=at24_add&0xf0;
        at24_add=key2|(key1&0x0f);
        }
        else
        {key2=at24_data&0xf0;
        at24_data=key2|(key1&0x0f);
        }
        index=00;
        break;
    }
}
}
}
if (flag==0x00)
Read24c02(&at24_data,at24_add,1);
display();
}
}
```

//第十章例子 10.5.9A) 、AT24C02.c 调用库函数对 AT24C02 进行读写，可以连续多字节读写

/\*\*\*\*\*\*

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

双龙电子公司----I2C 总线演示程序(ICCAVR)

作者：詹卫前

广州天河双龙电子有限公司 <http://WWW.SL.COM.CN>



1、以访问 AT24C02 为例

2、SDA : PORTA.0

3、SCL : PORTA.1

使用注意: 1、应包含头文件 slavr.h

2、在工程选项中的 Additional Lib 中填入 slavr

使用编译器使用附加库文件 libslavr.a

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

```
#include <io8515.h>
```

```
#include <slavr.h>
```

```
/* 七段译码字形表 */
```

```
#pragma data:code
```

```
const
```

```
unsigned
```

```
char
```



```
#pragma data:data
    /*    显示缓冲区    */
unsigned char ledbuff[]={0x3f,0x3f,0x3f,0x3f,0x00,0x00};
unsigned char at24_add,at24_data;
/*****

        I2C 总线写一个字节
*****/
void i2c_Write(unsigned char Wdata,unsigned char RomAddress)
{
    Start();
    Write8Bit(0xa0);
    TestAck();
    Write8Bit(RomAddress);
    TestAck();
    Write8Bit(Wdata);
    TestAck();
    Stop();
    delay_ms(10);
}
/*****

        I2C 总线读一个字节
*****/
unsigned char i2c_Read(unsigned char RomAddress)
{
    unsigned char temp;
    Start();
    Write8Bit(0xa0);
    TestAck();
    Write8Bit(RomAddress);
```

```
TestAck();
Start();
Write8Bit(0xa1);
TestAck();
temp=Read8Bit();
NoAck();
Stop();
return temp;
}
/*****
                        端口初始化
*****/
void port_init(void)
{
    DDRD=0x3f;
    DDRB=0xff;
    PORTD=0xff;
    PORTB=0xff;
}
/*****
                        六路动态扫描显示电路
*****/
void display(void)
{
    unsigned char i;
    i=at24_data%16;
    ledbuff[0]=tabel[i];
    i=at24_data/16;
    ledbuff[1]=tabel[i];
```

```

i=at24_add%16;
ledbuff[2]=tabel[i];
i=at24_add/16;
ledbuff[3]=tabel[i];
for (i=0;i<6;i++)
{
PORTB=ledbuff[i];//将显示缓冲区数据送 PORTB 口
PORTD=~(1<<i); //开始显示
delay_1ms(); //每一位显示保持一定时间
PORTD|=(1<<i); //关闭显示
}
}

```

```

/*****

```

主程序功能：演示了读写 AT24C02 的过程

说明：1、键盘上 0~F 键用于输入地址或数据，

SHIFT 键用于切换状态

2、上电复位后，按数字键只可以修改地

址，程序自动显示 AT24C02 单元内容。

此时可以修改数字，再按一次 SHIFT 键

程序将修改后的数字写入 AT24C02，然

后回到上电复位后的状态

```

*****/

```

```

void main(void)
{
unsigned char key1,key2;
unsigned char index,flag;
MCUCR=0;
at24_add=0;
at24_data=0;

```

```
index=0;
flag=0;
port_init(); //端口初始化
while(1)
{
    key1=scan_key();
    if (key1!=0x7f)
    {
        do
        {
            key2=scan_key(); //检查按键释放
            display();
        }
        while(key1==key2);
        if (key1>=0x80)
        {
            {
                i2c_Write(at24_data,at24_add);
                flag=0x00;
            }
            else flag=0x80;
            ledbuff[5]=0x00^flag;
            index=0;
        }
    }
    else
    {
        switch (index)
        {
            case 0:{if (flag==0x00)
```

```
        {key2=at24_add&0x0f;
          at24_add=key2|(key1<<4);
        }
      else
        {key2=at24_data&0x0f;
          at24_data=key2|(key1<<4);
        }
      index=01;
      break;
    }
  case 1:{if (flag==0x00)
    {key2=at24_add&0xf0;
      at24_add=key2|(key1&0x0f);
    }
    else
      {key2=at24_data&0xf0;
        at24_data=key2|(key1&0x0f);
      }
    index=00;
    break;
  }
}
}
}
if (flag==0x00)
at24_data=i2c_Read(at24_add);
display();
}
}
```

