

sswanglei 小乐 发/freetradezone 整理

发几个芯艺书本上的例子!

可以直接复制运行,省去了大家敲代码的时间了.:)

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/signal.h>
#include <avr/wdt.h>
#include <avr/delay.h>
#include <avr/pgmspace.h>
#include <stdio.h>
#include <stdlib.h>

#define SET_LED PORTE&=0xDF
#define CLR_LED PORTE|=0x20

/*****
/* 查询方式使灯闪烁*/
*****/
/*
int main()
{
    uint8_t i,j=0;
    DDRE = _BV(PE5);
    PORTE = _BV(PE5);
    TCNT0 = 0;
    TCCR0 = _BV(CS02)|_BV(CS00);
    while(1)
    {
        for(i=0;i<15;i++)
        {
            loop_until_bit_is_set(TIFR,TOV0);
            TIFR|= _BV(TOV0);
        }
        if(j)
            SET_LED,j=0;
        else
            CLR_LED,j=1;
    }
}
*/

*****/
```

```

/* 中断方式使灯闪烁*/
/*****/
/*
static uint8_t gbcount = 0;
static uint8_t gbdirection = 0;

SIGNAL(SIG_OVERFLOW0)
{
    if(++gbcount>55)
    {
        if(gbdirection)
            SET_LED,gbdirection = 0;
        else
            CLR_LED,gbdirection = 1;

        gbcount = 0;
    }
}

int main()
{
    DDRE = _BV(PE5);
    PORTE = _BV(PE5);

    TCNT0 = 0;
    TCCR0 = _BV(CS02)|_BV(CS00);

    TIMSK = _BV(TOIE0);
    sei();

    while(1);
}
*/

/*****/
/* 看门狗复位试验*/
/*****/
/*
void delays(uint16_t ms)
{
    uint16_t i;
    for(i=0;i<ms;i++)
        _delay_loop_2(4*250);
}
*/

```

```

}

int main()
{
    DDRE = _BV(PE5);
    PORTE = _BV(PE5);
    wdt_enable(WDTO_1S);
    wdt_reset();
    delayms(500);
    SET_LED;
    delayms(50000);
    SET_LED;
    while(1)
        wdt_reset();
}
*/

/*****
/* 查询方式串口通信 串口 1,BAUD = 9600*/
/*****
/*
void putc(uint8_t c)
{
    while(!(UCSR1A&(1<<UDRE1)));
    UDR1 = c;
}
uint8_t getc(void)
{
    while(!(UCSR1A&(1<<RXC1)));
    return UDR1;
}

int main()
{
    UBRR1H = 0;
    UBRR1L = 0x57;
    UCSR1B = (1<<RXEN1)|(1<<TXEN1);
    while(1)
    {
        putc(getc());
    }
}

```

```

*/

/*****
// 中断方式的串口通信*/
*****/

/*
uint8_t g_bTxdPos = 0;
uint8_t g_bTxdLen = 0;
uint8_t g_bRxdPos = 0;
uint8_t g_bRxdLen = 0;

uint8_t g_aSendBuf[16];
uint8_t g_aRecvBuf[16];

SIGNAL(SIG_UART1_RECV) //接收中断函数
{
    uint8_t c =UDR1;
    if(g_bRxdLen>0)
    {
        g_aRecvBuf[g_bRxdPos++]=c;
        g_bRxdLen--;
    }
}

SIGNAL(SIG_UART1_TRANS) //发送中断函数
{
    if(--g_bTxdLen>0)
        UDR1 = g_aSendBuf[++g_bTxdPos];
}

uint8_t IsRecvComplete(void) //通过判断接受长度是否变为 0，判断接受是否完成
{
    return g_bRxdLen == 0;
}

void SendToUart(uint8_t size) //发送 16 个字节的数据
{
    g_bTxdPos = 0;
    g_bTxdLen = size;
    UDR1 = g_aSendBuf[0];
    while(g_bTxdLen>0);
}

void RecvFromUart(uint8_t size,uint8_t bwait) //确定要从串口接受的数据的个数
{

```

```

    g_bRxdPos = 0;
    g_bRxdLen = size;
    if(bwait)
        while(g_bRxdLen>0);
}
int main(void)
{
    uint8_t i;
    UCSR1B = (1<<RXCIEN)|(1<<TXCIEN)|(1<<RXEN)|(1<<TXEN);
    UBRR1H = 0;
    UBRR1L = 0x57;
    sei();
    while(1)
    {
        RecvFromUart(16,0);    //确定要接收数据的长度
        while(!IsRecvComplete()); //等待接收数据长度为 0，则接收完成
        for(i=0;i<16;i++)
            g_aSendBuf[i]=g_aRecvBuf[i]; //将接收到的数据放入发送缓冲区
        SendToUart(16);        //发送发送缓冲区的 16 个字节的数
    }
}
*/

/*****/
//经典的串口打印函数
/*****/
/*
char g_aString[81];
int usart_putchar(char c)
{
    if(c=='\n')
        usart_putchar('\r');
    loop_until_bit_is_set(UCSR1A,UDRE1);
    UDR1 = c;
    return 0;
}

int usart_getchar(void)
{
    loop_until_bit_is_set(UCSR1A,RXC1);
    return UDR1;
}

```

```

void IoInit(void)
{
    UCSR1B = _BV(RXEN1)|_BV(TXEN1);
    UBRR1L = 0x57;

    fdevopen(usart_putchar,usart_getchar,0);
}

int main(void)
{
    int tmp;
    IoInit();
    while(1)
    {
        //way1: 这种方法太耗 ram
        printf("测试 1[输入一字符串]:\n");
        scanf("%s",g_aString);
        printf("您的输入为: %s\n",g_aString);

        //way2: 使用 flash 区的字符串
        printf_P(PSTR("测试 2[输入数字]:\n"));
        scanf_P(PSTR("%d",&tmp));
        printf_P(PSTR("您的输入为:%d\n"),tmp);

    }

}

*/

/*****/
//最小化的串口打印函数
//因为 gcc 所带的串口打印函数至少都要 1k 多，所以有必要自己写一个小的
/*****/
void DelayMs(uint16_t t)
{
    uint16_t i;
    for(i=0;i<t;i++)
        _delay_loop_2(250*4);
}

int usart_putchar(int8_t c)
{
    if(c == '\n')
        usart_putchar('\r');
}

```

```

loop_until_bit_is_set(UCSR1A,UDRE1);
UDR1 = c;
return 0;
}

```

```

void miniprintf(char *fmt,...)
{
    va_list ap;
    int8_t strval[6];
    int8_t *p;
    int16_t nval;
    uint8_t i;

    va_start(ap,fmt);
    for(p=fmt;*p;p++)
    {
        if(*p!='%')
        {
            usart_putchar(*p);
            continue;
        }
        p++;
        switch(*p)
        {
            case 'd':
                nval = va_arg(ap,int);
                itoa(nval,strval,10);
                break;
            case 'x':
                nval = va_arg(ap,int);
                itoa(nval,strval,16);
                break;
        }
        i=0;
        for(i=0;strval[i];i++)
            usart_putchar(strval[i]);

    }
    va_end(ap);
}

```

```

int main(void)
{
    uint8_t i = -5;

```

```
UCSR1B = _BV(RXEN1)|_BV(TXEN1);  
UBRR1L = 0x57;
```

```
while(1)  
{  
    miniprintf("dec:%d\n",i);  
    miniprintf("hex:0x%x\n",i);  
    i++;  
    DelayMs(1000);  
}  
  
}
```