



例程 1 实时显示单片机的参数（整数、小数、中文 系统指令方式）

1.实验目的

单片机按照串口屏的系统指令格式发送控制命令，串口屏实时显示单片机的参数，包括整数、小数、和中文字符。

2.页面设计



(1)进度条控件，以进度条的方式显示出SOC 值

(2)整数码件，显示具体 SOC 值

(3)文本控件，显示电量状态

(4)浮点数码件，显示电压值

(5)整数码件，显示电流值

选中控件，可以在右侧属性栏查看控件的各个属性。详细属性说明可参考 第四章 控件的介绍和用法。

资源区：

一共使用4 张图片，home 作为页背景图片，另外三张图片为进度条不同状态的颜色。

字体字体除了工程自带的字库 Tahoma_4x_ASCII （包括英文和数字），另外创建了一个中文的字库，双击这个字库，可以看到 详细的信息。字库详细制作方法可以参考第三章的添加字库小节。



3. 串口屏协议处理

由于该例程使用串口屏自带的命令(详细命令 参考第六章 串口指令的简介和用法)，无需设计其他通信协议。单片机按照指令格式即可控制显示屏。

4. 下载验证

编译成功后，点击下载按钮，选择正确的端口号和波特率，下载到串口屏。可以看到显示如下显示：



测试时，可以先通过电脑串口助手给屏发指令来调试。



注意：串口屏系统命令是以\r\n结尾的，所以发送数据时需要勾选“发送新行”，或者输入是手动敲入换行 wset numVf.valf 3.45

wset numA.val 120

wset psoc.val 95

wset nsoc.val 95

wset text.txt "电量充足"



串口屏显示如下，表示串口屏程序设计没问题。



实际应用时，是 MCU 和串口屏通信。本实验使用的单片机型号是 STM32F103TB,改例程仅作参考，开发环境 keil V5.24 。其他 MCU 的串口发送方式也类似。

部分代码

```
int main(void)
{
    Sys_Set Rcc();                //设置系统使用内部时钟
    delay_init(64);                //延时函数初始化
    usart1_init();                 //初始化串口 波特率 115200

    while(1)
    {
        //设置电压值
        sprintf(page_str,"wset numVf.valf %.2f\r\n",V);//①
        USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str)); //②
        delay_ms(1); //③命令之间需要加上小延时分开

        //设置电流值
        sprintf(page_str,"wset numA.val %d\r\n", I);
        USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
        delay_ms(1); //命令之间需要加上小延时分开

        //设置进度条 SOC
        sprintf(page_str,"wset psoc.val %d\r\n",SOC);
        USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
        delay_ms(1); //命令之间需要加上小延时分开

        //设置 SOC 百分比
        sprintf(page_str,"wset nsoc.val %d\r\n",SOC);
        USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
        delay_ms(1); //命令之间需要加上小延时分开

        if(SOC > 80)
```

```
{
    sprintf(page_str,"wset psoc.fglmg 2\r\n");//④绿色图片
    USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
    delay_ms(1); //命令之间需要加上小延时分开
    sprintf(page_str,"wset text.txt \"电量充足\"\r\n");//⑤
    USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
    delay_ms(1); //命令之间需要加上小延时分开
}
else if(SOC > 20)
{
    sprintf(page_str,"wset psoc.fglmg 4\r\n");//黄色图片
    USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
    delay_ms(1); //命令之间需要加上小延时分开

    sprintf(page_str,"wset text.txt \"电量正常\"\r\n");
    USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
    delay_ms(1); //命令之间需要加上小延时分开
}
else
{
    sprintf(page_str,"wset psoc.fglmg 3\r\n");//红色图片
    USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
    delay_ms(1); //命令之间需要加上小延时分开

    sprintf(page_str,"wset text.txt \"电量过低\"\r\n");
    USART_OUT( USART1, (uint8_t*)page_str,strlen(page_str));
    delay_ms(1); //命令之间需要加上小延时分开
}
//模拟数据变化
V += 0.08;
if(V > 5) V = 3.0;
if(++I > 100)I = 0;
if(++SOC > 100)SOC = 0;

delay_ms(200);
}
}
```

①sprintf 处理字符串，浮点数控件的赋值方式

②串口发送函数，以自己单片机工程里面的发送函数为准

③命令之间需要加上小延时分开，保证串口屏收到的命令之间没有粘连

④设置图片，这里的 2 表示图片的编号，在 VP 软件的图片资源里可以看到图片的具体编号

⑤字符串的发送，在 C 语言里双引号需要转义，这里发送的是 wset text.txt “电量充足”\r\n。发送中文时，注意 VP 工程里面的编码方式和开发环境代码的编码方式要一致，例程里面是 GBK 编码。否则汉字显示可能会有问题。

查看 VP 软件里的编码：项目设置 ，里面的字符编码，默认使用 GBK 编码

查看 keil 工程的编码：Eidt->Configruation->Eidtor,Encoding 选项，keil 的 GB2312 对应 VP 软件里面的 GBK；keil 的 Encode in ANSI 对应 VP 软件里面的 UTF-8

若使用其他的开发环境，类似的设置。注意编码方式一致即可。