

# 实验4： 串口实验

---

11610101 韦青茂

## 实验器材

---

- 硬件：ARM-STM32开发板，J-Link/St-Link。
- 软件：Win7/Win8/Win10, Keil uVision5

## 实验要求

---

1. 归纳总结创建工程过程中遇到的问题以及解决方法。
2. 功能 1 实现：从串口调试助手输入自己的姓名，返回 “Hello, xxx” 。

## 实验过程

---

### 软件代码(main.c)

```
#include "led.h"
#include "delay.h"
#include "sys.h"
#include "key.h"
#include "usart.h"
#include "string.h"

// 判断串口读入字符串来控制LED灯
void LED_Control(const char *recv)
{
    const char *l0_on = "led0 on";
    const char *l0_off = "led0 off";
    const char *l1_on = "led1 on";
    const char *l1_off = "led1 off";
    int _control = 1;
```

```

    if (strcmp(recv, l0_on) == 0)
        LED0 = 0;
    else if (strcmp(recv, l0_off) == 0)
        LED0 = 1;
    else if (strcmp(recv, l1_on) == 0)
        LED1 = 0;
    else if (strcmp(recv, l1_off) == 0)
        LED1 = 1;
    else
        _control = 0;
    if (_control)
        printf("%s!", recv);
    else
        printf("Hello, %s", recv);
}

int main(void)
{
    u8 t;
    u8 len;
    char recv[20];

    LED_Init();
    KEY_Init();
    delay_init();
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
    uart_init(9600);

    while (1)
    {
        t = KEY_Scan(0); //得到键值
        switch (t)
        {
            case KEY0_PRES:
                printf("KEY0 pressed!\r\n");
                break;
            case KEY1_PRES:
                printf("KEY1 pressed!\r\n");
                break;
            case WKUP_PRES:
                printf("WK_UP pressed!\r\n");
                break;
            default:
                delay_ms(20);
                break;
        }
    }
}

```

```

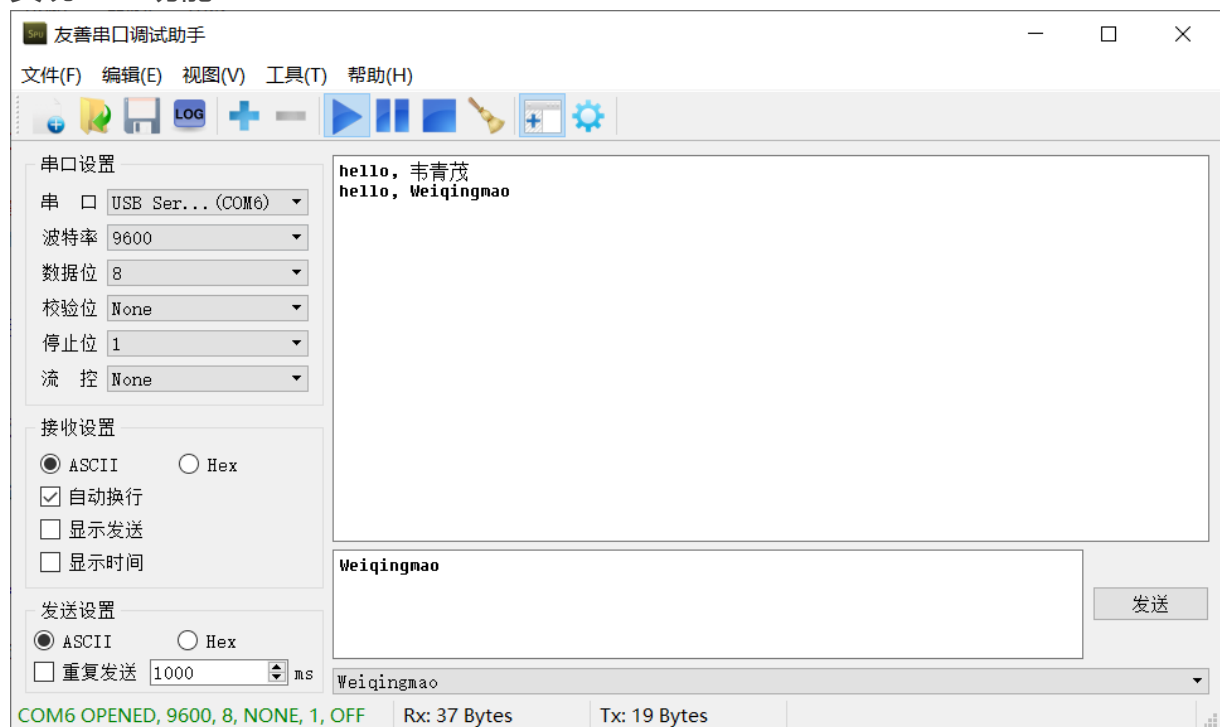
    }
    if (USART_RX_STA & 0x8000)
    {
        len = USART_RX_STA & 0x3fff;

        for (t = 0; t < len; t++)
        {
            recv[t] = USART_RX_BUF[t]; // 接收到的字符串逐字符存入recv中
            while ((USART1->SR & 0x40) == 0)
                ;
        }
        recv[t] = '\0'; // 在字符串末尾拼接 '\0'
        LED_Control(recv);
        USART_RX_STA = 0;
    }
}
}

```

## 串口实验：

### 实现echo功能:



## 遇到的问题及解决方法

1. 串口打印返回字符串时，字符串最后有时莫名其妙多出一个奇怪字符

可能是因为储存用来打印的字符串变量每次用之前没有清空，于是在末尾添加'\0'

```
recv[t] = '\0'; // 在字符串末尾拼接 '\0'
```

这样打印该字符串时就会以这个手动添加的'\0'作为结束符

## 2. 试图打印中文时，出现乱码

经过试验发现，MDK编辑器和串口监视终端编码一致时才能显示正常中文，而之前在mdk里设置编码为utf8，但是串口终端还是GB2132，所以才会显示乱码，改成一致的即可显示正常。