

## STM32-WIFI-W8782 软件裁剪和定制说明

### 1. WIFI 部分

文件:net/wlan\_src/user/usr\_cfg.h

可定制功能

#### (1)DEBUG 功能

```
#define DEBUG_LEVEL1
```

```
#define DEBUG_LEVEL2
```

LEVEL1 和 LEVEL2 都打开的情况下将打印所有调试信息,一般情况下选择 LEVEL1 即可,官方配置也是 LEVEL1,发布产品时需要关掉

#### (2)STA 和 AP 模式选择

```
#define STA_SUPPORT
```

```
#define STA_CFG80211
```

```
#define UAP_SUPPORT
```

```
#define UAP_CFG80211
```

以上两组预编译用于 STA 和 AP 模式的裁剪,用户可根据自己的情况选择,默认全部打开

### 2. LWIP 部分

文件:net/lwip/src/include/lwip/opt.h

这里的预编译选项用于配置 LWIP 协议栈的参数,大部分选项不要轻易去修改,这里把一些常用的并且比较重要的选项做一下说明

(1) MEMP\_NUM\_PBUF:定义 pbuf 缓冲区的大小, pbuf 缓冲区用于缓存网络通信数据

(2) MEMP\_NUM\_UDP\_PCB:UDP 最大连接数

(3) MEMP\_NUM\_TCP\_PCB:TCP 最大连接数

(4) MEMP\_NUM\_TCP\_PCB\_LISTEN: TCP 服务器允许的最大连接数

(5) MEMP\_NUM\_NETCONN:允许的最大连接数,此值不要大于以上三项之和,这个值越大越耗内存,请小心配置

(6) TCP\_MSS:TCP 的 MSS 值(最大报文值)配置,这个值的大小会影响网速,默认 1024,调小可以节省内存消耗,最大不要超过 1500

(7) TCP\_SND\_BUF:TCP 发送缓冲区大小,会影响发送速度,同样注意内存的消耗

如果系统内存紧张,可以通过反复调节上面的参数,在网速和内存占用之间找一个平衡点

### 3. UCOS 部分

文件:kernel/ucos/include/os\_cfg.h

主要用于 uc0s 内核级的配置,一般情况下不要去修改,一些常用的配置比如线程数量,定时器数量等的配置我们放在系统配置文件里面.

### 4. 系统配置

文件:config/app\_cfg.h

这是整个系统的全局配置文件,很多全局参数可以在这里配置

(1) MP3 和 WEBSERVER 功能裁剪

```
#define SUPPORT_AUDIO      1    //音频范例的支持
#define ENABLE_MP3         0    //需消耗 63k flash
#define SUPPORT_WEBSERVER  0    //需消耗 37k flash
#define MACHTALK_ENABLE    1    //云控制范例支持
```

(2) 任务优先级定义

enum TASK\_PRIO;

UCOS 每个线程优先级不能相同, 放在这里统一管理, 用户新增加的任务也需要在这里先定义任务优先级

(3) 任务堆栈大小定义

每个线程的堆栈需要预先定义, 用户新增加的任务也需要在这里先定义堆栈大小, 大小的选择建议通过 monitor 观察并反复调节

(4) 中断优先级定义

中断区分优先级的好处是在系统繁忙的时候可以确保重要的事件更及时的执行, 修改的时候请确保 SDIO 中断优先级最高

(5) UCOS 常用资源配置

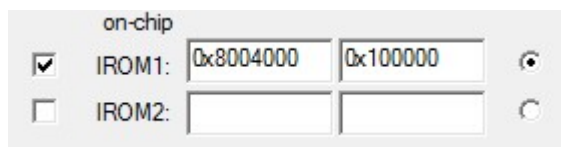
```
#define EVENTS_MAX      48    //EVENTS_MAX 是所有事件的数量之和
#define TIMER_MAX       20    //定时器
#define MSG_QUEUE_MAX   16    //消息队列
#define TASKS_MAX       14    //任务
```

以上几个选项建议搭配 monitor 软件来配置, 这些资源可能会在运行过程中动态增减, 必须预留一定余量

(6) 中断向量表地址

IVECTOR\_ADDR

开发板的中断向量表在 0x4000 处 (0-0x4000 放 bootloader), 如果用户需要修改这个地址可以直接修改这个宏, 一般中断向量表地址也是整个工程的起始地址, 所以修改这个宏的时候也需同时修改工程属性中的 ROM 起始地址, 如下图



## 5. DEBUG 说明

文件: app/inc/debug.h

```
#define SYS_MONITOR      1
#define RELEASE_VERSION  0
```

SYS\_MONITOR 是 monitor 功能的开关, 默认打开, 用户不想用可以将其置 0

RELEASE\_VERSION 意思是发布版本, 调试阶段需置 0, 如果置一将关闭所有打印功能 (包括错误信息)

LWIP\_NOASSERT 是 LWIP 的 ASSERT 开关, 默认关闭 LWIP 的 ASSERT

文件: net/lwip/src/include/lwip/debug.h

```
//#define LWIP_DEBUG  
//#define LWIP_NOASSERT  
//#define LWIP_ERROR
```

LWIP\_DEBUG 是 LWIP 的 debug 开关, 默认关闭

LWIP\_NOASSERT 是 lwip 断言开关, 默认关闭

LWIP\_ERROR 是 LWIP 的致命错误检测开关, 调试阶段默认是打开的

文件: kernel/api/inc/memory.h

```
#define USE_MEM_DEBUG 0
```

内存调试功能, 开启之后可以记录每一块内存的申请者名称(函数名)、开启内存溢出检查等功能, v1.8 版本之后默认关闭, 以节省内存开销

Debug 和 monitor 都是通过 uart1 输出, 其与 printf 的接口函数 fputc 位于 usart.c 文件