

STM32-WIFI-W8782 开发板用户手册

(V1.1)

广州市勋睿科技发展有限公司

联系电话：13025182598

www.xrf.net.cn

wsum205.taobao.com

版本说明：

V1.0 初始版本

V1.1 增加 machtalk 云控制的说明

目录

1 简述	3
2 开发环境	5
3 硬件说明	7
3.1 开发板介绍	7
3.2 开发板的使用	8
4 软件说明	9
4.1 代码目录结构	9
4.2 keil 工程介绍	9
4.2 源代码介绍	10
4.2.1 bsp 库	10
4.2.2 kernel	10
4.2.3 drivers	10
4.2.4 net	11
4.2.5 config	11
4.2.6 app	11
4.3 wifi 操作	12
4.4 TCP/IP 操作	12
4.4.1 TCP 范例操作	13
4.4.2 UDP 范例操作	13
5 开发板启动过程	13
6 程序的烧写	14
7 应用实例	16
7.1 LED 灯控制	16
7.2 音频对讲	17
7.3 MP3 播放	18
7.4 WEB 控制	19
7.5 MACHTALK 云控制	20

1 简述

STM32-WIFI-W8782 开发板是一款超高性价比的嵌入式 WIFI 方案，基于 MARVELL 公司的 88W8782，这是一款功能强大的 WIFI-SOC 单芯片，具有多种优异的特性：采用 SDIO 接口，支持 80211. a/b/g/n, AP/STA/ADHOC 工作模式，WPA/WAP2 加密，支持睡眠/低功耗等等。通过我们的方案，你可以使用普通的单片机驱动这款芯片，并迅速开发出低成本高性能的 WIFI 产品。

开发板特点：

- 采用通用 ARM-Cortex M3 高性价比单片机 STM32F205RGT6，该单片机提供 1Mbyte 的 FLASH 和 128Kbyte 的 RAM 空间，具有丰富的外设资源。
- 提供开发板源码及 android 测试软件源码，提供专业技术支持。
- 无线通讯速率超过 10Mbps (双向)
- 支持 WEP、WPA/WPA2 安全认证和 TKIP、AES 等各种加密模式
- 支持 AP、STATION 、ADHOC 工作模式
- 集成高稳定性的实时操作系统，提供强大完善的 API 接口
- 集成 LWIP-TCP/IP 协议栈，提供简单易懂的范例操作
- 集成 DHCP 服务器
- 集成 libmad-MP3 解码库
- 集成 WEB 服务器范例，可实现表单提交和文件传输等交互功能
- 开发板提供 3 路 LED 显示，提供一个 camera 接口、一个麦克风、一个耳机插孔，提供手机与开发板进行语音对讲、手机控制 led 灯的开关、在线播放 MP3 等多个有趣的范例。
- 开发板提供 1 片 1Mbyte 的 SPI-Flash
- 开发板提供一片 I2S 音频 DAC 芯片，可进行高质量 MP3 播放

开发板基本参数

工作模式	电流
待机	200ua
省电模式(保持 WIFI 连接)	32ma
收发数据(全速)	200ma

88W8782 芯片功耗（实测）

通信模式	速率
单接收或单发送	6.7Mbps
开发板自动回发测试	10Mbps

开发板通信速率（实测）

代码编译条件	ROM 占用(大概值)
单 STA	230KB(code) + 250KB(FW)
单 AP	200KB(code) + 250KB(FW)
AP + STA	250 KB(code) + 250KB(FW)

开发板资源占用情况

2 开发环境

STM32-WIFI-W8782 开发板需要在 windows 环境下进行开发，代码编辑可用 source insight 或者 keil 等多种编辑软件，我们提供的源代码包里面已经有 source insight 和 keil 的工程，如果安装了上述软件直接打开即可。而代码编译需要安装 keil(keil4 或 keil5)。

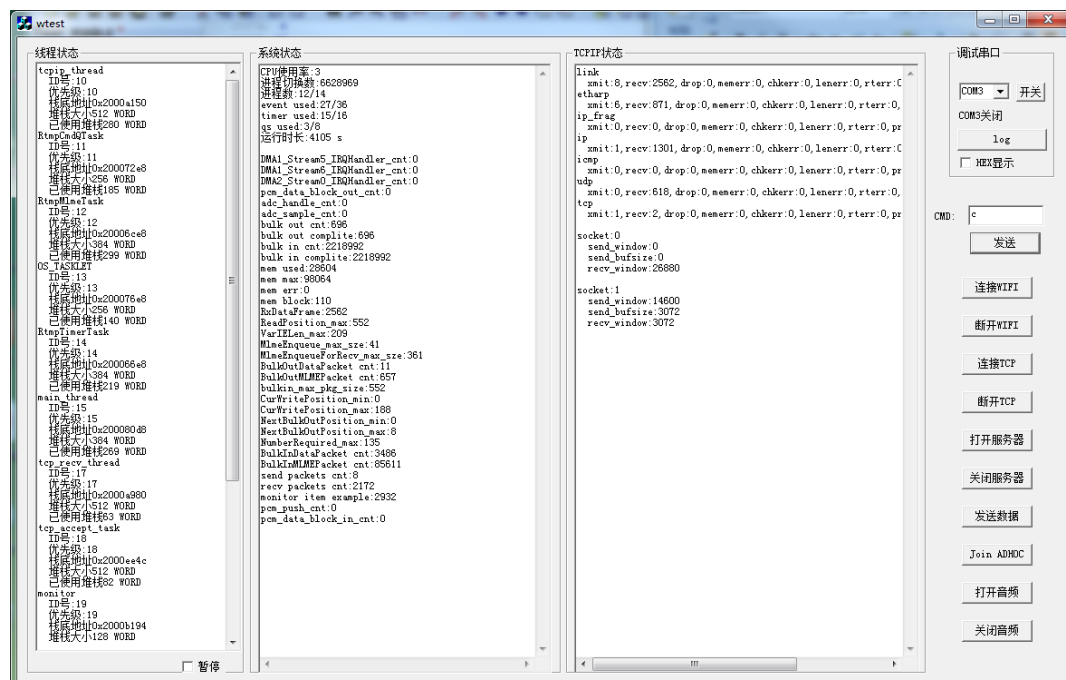
编译连接成功后，你可以通过 jtag 将代码烧写入单片机进行在线调试，也可用通过串口将二进制文件下载到单片机运行（将在后面进行详细说明）。

下面列出本开发板需要用到的一些软硬件资源及工具：

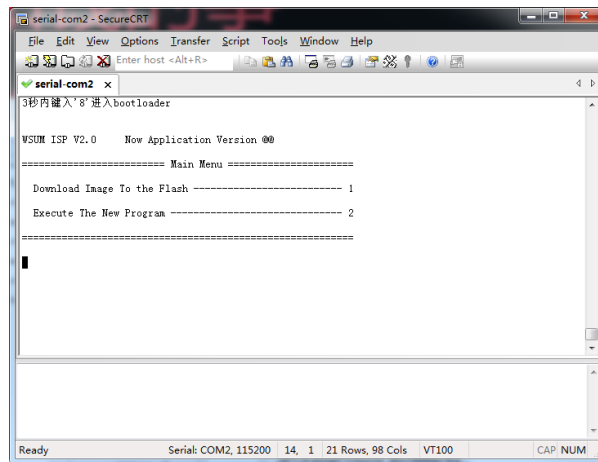
1. Jlink 仿真器：如果需要进行在线仿真调试，这是必须的工具(需要自行购买)，开发板提供的 2*5 针（2.54mm）的标准 JTAG 接口（支持 JTAG, SWD）。

2. 串口：在电脑上打印调试信息必须用到串口，我们建议采用台式机的 232 接口，如果你用的笔记本请使用质量较好的 USB 转串口线。

3. 串口调试工具（MONITOR）：下面是开发板工具包中提供的串口调试工具，它可以显示当前系统的运行状态以及用户自定义的状态量，打开 log 窗口还可以显示系统调试信息，右下角是为常用的操作定制的几个按钮，按下一个按钮会发送一个字节的命令字给开发板，以方便你测试。运用这个调试工具可以帮助你快速进行 bug 定位，缩短开发时间。



4. 串口烧写工具：我们推荐使用 secureCRT，如果你没有 jlink 来烧写程序，可以用这个工具将程序烧写到单片机（需要用到 Ymodem 传输），其运行界面如下：

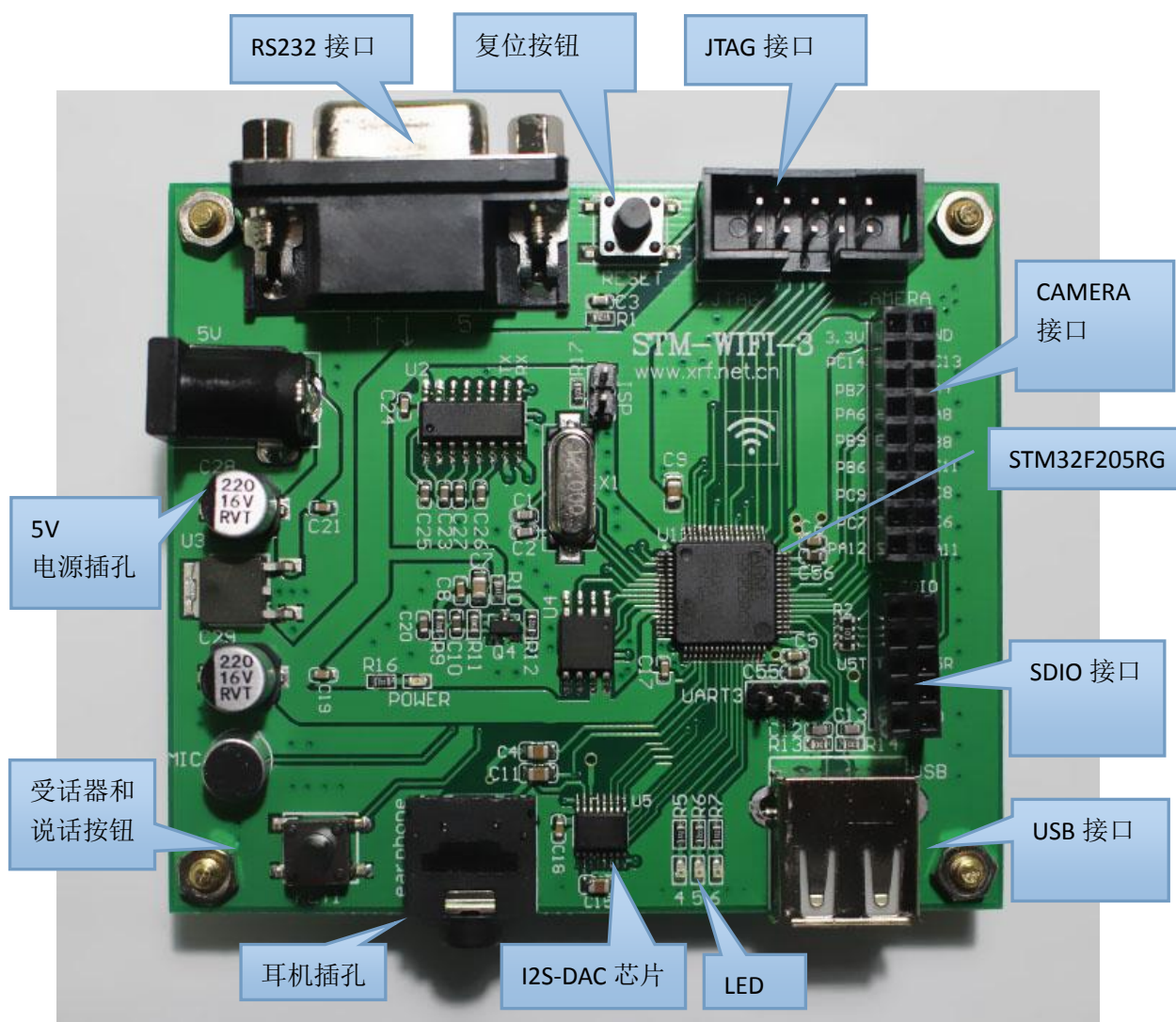


5. TCP 调试工具：WIFI 数据通信的上层都是基于 TCP/IP 协议的，所以必须要用到 TCP 调试助手，开发板工具包里面有提供 (NetAssist.exe)，你也可以用其他的。

3 硬件说明

3.1 开发板介绍

整套开发板包括一块 PCB 主板、一个 88W8782 模块、一个 5V2A 电源、一根 RS232 数据线以及若干杜邦线，我们会提供主板原理图和 PCB 资料（不提供 88W8782 芯片的电路图）。



STM32-WIFI-3 主板

对于主板我们做下面几点说明：

1. 电源供电请确保使用 5V 电压，否则将可能烧坏主板芯片。
2. 开发板在进行语言通信时板子输出的音频会有一点噪声，这是由于开关电源以及 I2S 芯片输出没有经过滤波造成的，应用到产品中时请注意这两部分电路的优化。同时 WIFI 的射频信号也会对器件造成很大的干扰，应用中需要注意。

3. CAMERA 接口在此版本中无法使用（与 SDIO 接口冲突），你可以作为普通 IO 口使用。
4. SDIO 信号线容易受到干扰，开发板工作时请勿触碰信号线，应用中 SDIO 接口走线尽量短并且等长，同时需要做好屏蔽措施。



W8782 模块

3.2 开发板的使用

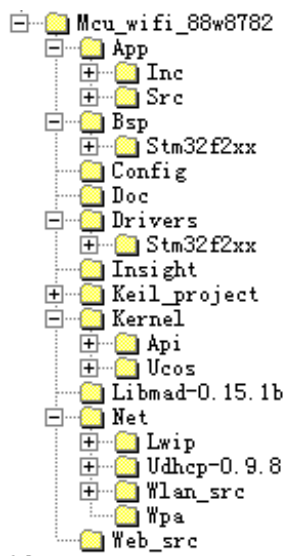
开发板使用之前需要先连接好电源线和串口线，在电脑上安装并打开 monitor 串口调试软件，选择正确的串口（monitor 软件只支持 COM1-COM16，请注意），接通电源之后可以看到底板中间的电源指示灯亮，此时 monitor 会开始显示系统状态信息。如果打开了 log 窗口，在 log 窗口里面会打印开发板启动过程中的一些信息：

```
|_Frequency:30000000,PCLK2_Frequency:60000000,ADCCLK_Frequency:0
[! 0.230] tpl code:15,size:31
[! 0.240] Marvell
[! 0.240] tpl code:20,size:4
[! 0.250] tpl code:21,size:2
[! 0.250] tpl code:22,size:4
WLAN FW is active
[D: 0.990] tcpip_init_end
[D: 0.990] enter init_udhcpd
[D: 1.000] enter enable_dhcp_server
[D: 1.000] 代码版本:V1.2
[D: 1.000] 启动时间1.0 S
[D: 1.000] 欢迎使用广州市翊睿科技发展有限公司产品
[D: 1.010] 技术支持电话:13025182598, QQ:381935393
[D: 1.010] 请关注wsum205.taobao.com
```


4 软件说明

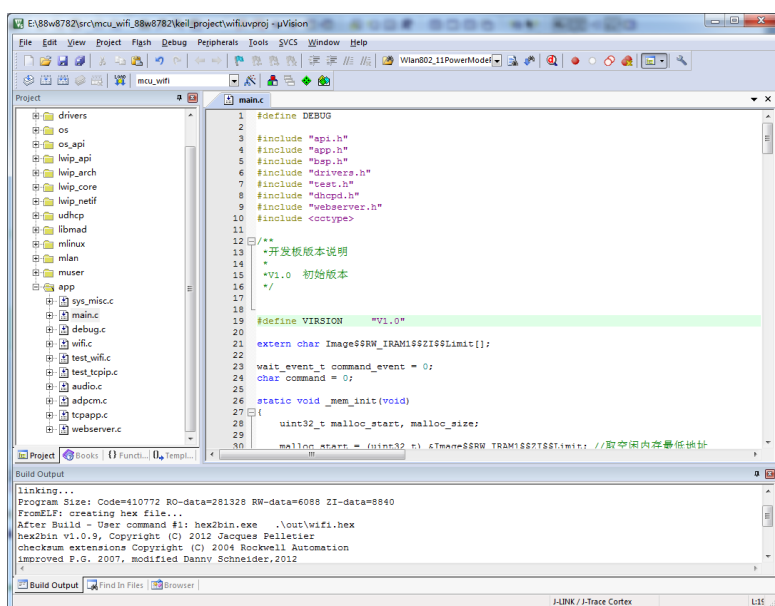
4.1 代码目录结构

我们提供的开发板源代码目录结构如下（因版本而异）：



4.2 keil 工程介绍

开发板源代码的 keil_project 目录下面是 keil 工程文件，打开后会显示如下的界面：



其中左侧显示的是工程窗口，工程下面分为多个组以方便进行代码管理。

4.2 源代码介绍

源代码根据我们提供的版本不同会有少许变化，有任何疑问可以联系技术支持。

4.2.1 bsp 库

bsp 文件夹下面是芯片相关的代码，包括 st 官方提供的 mcu 外设驱动库，以及 cortexM3 相关的库和启动代码，在官方的源码基础上我们对寄存器修改相关的部分增加了原子操作，以增加代码稳定性。

4.2.2 kernel

kernel 文件夹下面是ucosii 内核以及我们进行二次封装的源码，kernel/api 下面的封装库将 ucOS 提供的系统函数封装成通用的操作系统 api，如果想更换成别的内核，只需要修改 api 目录下的代码即可。内存管理代码也放在 api 下面，是移植自 lwip 的内存管理代码，同时我们增加了溢出检测功能，每 100ms 检查一次内存状况，内存出错时会打印出相应信息并停止运行代码，方便你尽快进行排查。

```
[E: 101.890] detect memory overflow, mem_size:20, flag:994f816e
```

```
[E: 101.890] mem_head:2000b980, data_addr:2000b990, last_addr:2000b95c, next_addr:2000b9a8
```

```
[E: 101.890] critical addr:2000b9a4
```

```
[E: 101.890] name:not used
```

```
[E: 101.890] last name:tcp_create_segment
```

```
[E: 101.890] next name:not used
```

内存溢出 DEBUG 信息，具体可以参考 mem_slide_check 函数，此函数还可以将当前的内存分配情况全部打印出来（调用的时候参数赋 1 即可），可以方便你检查内存泄露的情况。

4.2.3 drivers

drivers 目录下驱动代码，包括串口，usb，SPI-Flash、以及音频驱动代码。如果想把代码移植到别的 mcu，只需要修改此目录下面的代码以及 bsp 下面的代码。

4.2.4 net

net/wlan_src 目录下是 88w8782 芯片相关的源码。此源码是从 linux 版本移植过来，所以里面会看到很多 linux 下的接口，我们根据 mcu 的资源情况做了大量的优化以使其尽少的消耗 ram 和 rom 资源，用户可以根据自己的需求对代码进行定制修改，以及裁剪，目前支持的裁剪功能是 STA 和 AP 模式的选择，用户可以选择单 STA 模式或者单 AP 模式，或者两者都选择，相关操作请在 wlan_src/user/usr_cfg.h 文件中完成。

net/lwip 目录下面是 lwip-tcp/ip 协议栈的源码，其中 lwip/src/netif/ethernetif.c 文件是跟 wifi 协议栈对接的代码，即底层数据收发的接口，关于 lwip 协议栈可以阅读其文档，里面有详细的说明。这里对数据收发两个接口函数做一下说明

1. 发送：WIFI 的数据发送由 net_device->net_device_ops->ndo_start_xmit 接口完成，这是 linux 内核里面的标准接口，详细的说明可以参考 linux 内核源码。此函数由 lwip/src/netif/ethernetif.c 文件中的 low_level_output->mac_data_xmit 调用。
2. 接收：lwip 协议栈的数据接收接口是 ethernetif_input 函数，此函数被 88w8782 驱动中的 netif_rx 函数调用，netif_rx 也是 linux 内核下面的一个通用接口。

用户操作 WIFI 主要通过 app/wifi.c 文件里面的函数接口，我们对 WIFI 操作进行了简单的封装，用户调用这里提供的函数即可对 wifi 进行常用的操作，具体参考 app 的详细说明。

4.2.5 config

config 目录下面是 app_cfg.h 配置文件，主要用于配置线程优先级、线程堆栈、中断优先级等等。

同时 MP3 解码和 WEBSERVER 的编译选项也在这里，用户可根据情况选择是否使用这些功能。

4.2.6 app

app 目录下面放的是用户应用代码，对各个文件简单介绍如下表：

文件名	说明
Main.c	系统初始化入口
Sys_misc.c	放置一些小的比较杂的功能函数，如 gpio 配置函数的封装等等
Wifi.c	这里是对 wifi 操作函数的封装
Test_wifi.c	测试 wifi 功能的一些函数，用户可以参考这里的函数实现自己的应用
Tcpapp.c	Tcpip 相关的一些函数，如 lwip 初始化等

Test_tcpip.c	测试 TCPIP 功能的一些函数，用户可以参考这里的函数实现自己的应用
Audio.c&adpcm.c	Adpcm 音频编解码以及音频播放的代码
Webserver.c	一个 Web 服务器的范例

4.3 wifi 操作

关于 wifi 的操作可以参考 wifi.c 和 test_wifi.c 两个文件，下面介绍几个常用的功能。我们对 WIFI 操作尽量进行了简化，使其操作起来更像是电脑连接路由器那样简单：

1. 扫描 wifi：扫描附件的 ap 信息调用 wifi_scan 函数，扫描到的 AP 信息将通过回调函数通知用户，每搜到一个 ap 回调一次，搜索完成后系统会调用 cfg80211_scan_done 函数通知用户，详细说明请参考 test_scan 函数。
2. 连接 WIFI：即连接到 AP 或路由器，你只需提供路由器的名称和密码，然后调用 wifi_connect 函数可以实现 WIFI 连接，整个连接过程需要数秒钟。
3. 断开 WIFI 连接：调用 wifi_disconnect 函数可断开 WIFI 连接。
4. 配置 AP：调用 wifi_ap_cfg 配置 AP 的参数，同时切换到 AP 模式，包括名称、密码、加密类型、频道、最大允许连接数等。
5. 关闭 ap：调用 wifi_stop_ap 可以停止 ap 功能。
6. 加入 adhoc 网络：调用 wifi_join_adhoc 用于加入 adhoc 网络，系统会先查询附近是否有同名的 adhoc 网络，如果有就加入，否则将新建一个。
7. 退出 adhoc 网络：调用 wifi_leave_adhoc 可退出 adhoc 网络
8. 配置省电模式：调用 wifi_power_cfg 可以配置 wifi 是否开启省电模式，以及省电模式的级别。

上述几个操作基本可以完成常用的 wifi 功能，至于 WIFI 模式的切换是自动进行的，我们不必理会，更多的 wifi 功能查阅 wifi.c 源码。

4.4 TCP/IP 操作

进行 TCP/IP 操作之前请确保 WIFI 已经建立连接，我们板子的静态 IP 地址默认值如下（AP 和 STATION 模式）。

IP 地址：192.168.1.8
子网掩码：255.255.255.0
网关地址：192.168.1.1
DNS 地址：208.67.222.222

所以在确保 WIFI 已经连接的情况下，还需要确保 IP 地址与远程端是在同一网段内，有些手机作为热点时的网段是“192.168.43.*”，此时你可以将板子的 IP 地址修改为跟手机同一网段，或者在 monitor 软件发送命令号‘d’执行自动获取 IP 地址，获取到热点分配的地址之后就能进行正常的 TCP/IP 通信了。

如果开发板工作在 AP 模式，手机（或者其他 WIFI 设备）连接到开发板之后，开发板将分配一个地址给手机。TCP/IP 的用户接口比较丰富，包括 TCP、UDP、

ICMP、DNS、HTTP、广播、多播等等，对于以上比较常用的功能，Lwip 都可以提供支持且提供的都是标准的接口，我们的范例中提供 TCP、UDP 以及多播和广播操作。关于 lwip 的更多操作请参考相关书籍。

4.4.1 TCP 范例操作

- a. 建立 TCP 连接：调用 `test_tcp_link` 连接会连接到 ip 地址为“192.168.1.101”、端口号 4700 的服务器，如果此 IP 地址的服务器已经打开则会提示连接成功。
- b. 打开本地 TCP 服务器：`test_tcp_server` 会在本地打开端口号为 4800 的服务器，同时会新建一个线程（`tcp_accept_task`）进行监听。
- c. 数据接收：在初始化 lwip 的时候我们会创建一个线程（`tcp_recv_thread`）用于数据接收，此线程采用 `select` 方式监听所有已经建立的 socket。
- d. 关闭连接：我们封装了一个管理 SOCKET 连接的函数 `close_socket`，其中会调用 `shutdown`、`close`、`FD_CLR` 等一系列操作，以确保 socket 完全关闭。注意 `FD_CLR` 是用于清除 `tcp_recv_thread` 线程中 `select` 方式加入的 socket。
- e. 数据发送：调用 `test_send` 和 `test_sendto` 都可以进行数据发送，`test_sendto` 可以指定目标地址和端口，主要用于 UDP 通信。

4.4.2 UDP 范例操作

- a. 连接到 udp 服务器：调用 `test_udp_link` 函数连接到 ip 地址为“192.168.1.101”、端口号 4701 的远程 udp 服务器。基于 udp 的特性，即使远程端没有开启 udp 服务器也可以连接成功。
- b. 打开本地 UDP 服务器：调用 `test_udp_server` 函数将建立端口号 4703 的本地服务器，等待远程端发送数据。
- c. 加入 udp 多播组：调用 `test_multicast_join` 函数将加入地址为 244.0.0.1，端口号 4702 的多播组，多播和广播方式用在局域网内，在进行数据分发时有很大的优势，比如分发音频流，所以我们这里也做了一个范例。

5 开发板启动过程

开发板的初始化代码在 `main.c` 文件。其中 `main` 函数是系统的入口，由启动代码（`startup_stm32f2xx.s`）跳转过来，`main` 函数主要工作是初始化内存分配和 ucos 系统，然后启动主线程 `main_thread`。需要注意的是在进入 `main` 之前，启动代码会调用 `SystemInit` 函数，此函数最后一行代码是重新映射中断向量表的地址，因开发板的低 16k 地址是作为 bootloader 供用户烧写代码的，所以这里会将中断向量表的地址映射到 0x4000 地址处。

main_thread 线程用于初始化系统的各个部件，最后会进入一个命令界面，如下：

```
[D: 2.510] 测试功能:
0:基本信息
1:扫描
2:连接WIFI
3:关闭WIFI连接
4:创建AP
5:连接TCP
6:关闭TCP连接
7:send发送
8:建立TCP服务器
9:关闭TCP服务器
a:启用音频功能
b:关闭音频功能
c:dns测试
d:自动获取IP地址
e:打印内存信息
f:获取wifi连接信息
g:连接UDP
h:加入多播组
i:建立UDP服务器
j:sendto发送
k:切换音频模式(mp3/pcm)
l:调节音量
m:关闭DEBUG
n:关闭AP
o:获取sta信息
p:加入adhoc网络
q:退出adhoc网络
r:使能省电模式
s:关闭省电模式
?:帮助
```

请输入上面的命令选择测试的功能:|

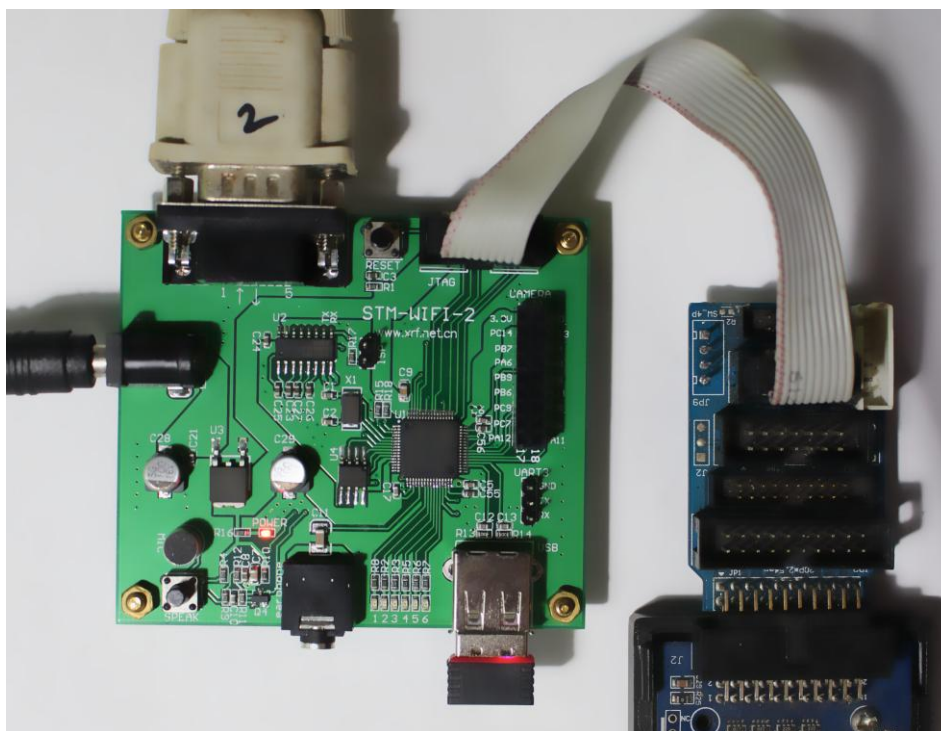
用户可以在 monitor 软件的右边 cmd 编辑框输入上述的命令，然后点击“发送”就可以执行相应的功能，当然也可以点击右下角的一排按钮进行操作，效果是一样的。

开发板默认工作在 AP 模式，上电启动完成之后你可以搜索到名称为 xrf_ap 的 AP，密码：12345678。

6 程序的烧写

用户编译连接完成后会在..\keil_project\out 目录下面生成 wifi.hex 和 wifi.bin 文件，可以通过 jtag 或者 bootloader 下载两种方式进行烧写，其中 wifi.bin 文件用于 bootloader 烧写。

jtag 烧写需要将主板连接到 jlink，连接方式如下图



JLINK 连接示意图（仅供参考）

如果你没有上图所示的转接板请按照下图的方式进行飞线转接



开发板与 JLINK20 针接口转接示意图（SWD）

连接好之后直接按 keil 上面的 download 按钮就可以完成下载过程。

bootloader 源码可以在我们提供的源码包里面找到，其主要是提供串口下载程序的功能，烧写需要用到第二章介绍过的 secureCRT 工具，开发板上电或者复位后会等待 3 秒，并等待用户键入数字键 ‘8’，如果用户在 3 秒之内按 ‘8’ 则会显示如下的 bootloader 信息：

```
WSUM ISP V2.0      Current Application Version V3.1
Download Image To the Flash ----- 1
Execute The New Program ----- 2
```

此时键入 “1” 可进入程序烧写功能，使用 Ymodem 协议传输，使用 secureCRT 上传程序步骤如下：

1. 打开开发板电源，开发板会显示“3 秒内键入 8 进入 bootloader”，此时按下数字键‘8’，窗口将进入 bootloader 并显示相关信息。
2. 打开菜单“传输”-“发送 Ymodem”，选择 wifi.bin 文件（文件名可任意修改），点击确定开始文件传输。传输完毕后可以看见提示。
3. 升级完后键入数字键‘2’或者重新上电即可。

7 应用实例

这里我们提供了两个比较有趣的应用范例：

1. LED 灯开关控制以及 LED 灯的亮度控制。
2. 与手机实现语音通信。
3. 播放 mp3 的例程
4. WEB 控制开发板重启
5. machtalk 云控制

前面两个例子通过 ANDROID 手机实现，第三个例子在电脑上实现，第四个例子可在任意终端实现（有 web 浏览器就行）。

开发板初始化的时候已经启动了一个本地服务器（端口号：4800），所以这里可以省掉开启服务器的步骤。

7.1 LED 灯控制

在开发板中我们提供了 6 个 LED 灯供用户使用，socket 通信中一般需要定义一套完善的数据包格式进行数据的封装，为了让实例简单易懂我们就只用个简单的命令格式：

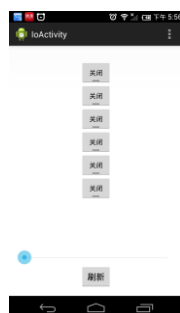
1. 0xaa, 0x55, 0x01, 0xXX 用于设置 LED 灯的开关，其中 0xXX 的低 6 位用于控制 6 个 LED。
2. 0xaa, 0x55, 0x02, 0xXX 用于设置 LED 灯的亮度，其中 0xXX 取值 0 - 99。

开发板在接收到数据后会判断是否为上述命令（参考 tcp_recv_thread 函数），如果是则对 LED 进行对应的操作。

在 android 手机安装 WSUM205_TEST 工具，此工具我们有提供源码，用户可以任意修改。WSUM205_TEST 工具只提供 TCP 客户端功能，所以此时需要先在开发板打开 TCP 服务器。工具显示如下的简单界面：



输入服务器的 IP 地址（开发板默认 IP 地址是 192.168.1.8）和端口号。点击“连接”按钮，左上角会提示是否连接成功。连接成功后点击“IO 测试”按钮进入下面的界面：



中间的 6 个按钮对应 6 个 LED 的开关，用户可以操作看开发板上的 LED 灯是否会有反应。下面的滑动条用于控制 LED 灯亮度（已经点亮的 LED 才能控制亮度）。我们提供的代码对亮度的处理比较简单，只实现了 10 级亮度，所以你控制亮度的时候会发现不是很流畅，比较合理的方式应该是采用 PWM 实现。

IO 控制简单步骤如下：

1. 点击 monitor 的“连接 WIFI”按钮，将 WIFI 板连接到路由器（需要修改程序以指定正确的路由器名称密码等）。
2. 打开 WSUM205_TEST 软件（此时确保手机也连接到路由器），输入正确的 IP 地址和端口号（4800），并点击连接按钮，应该提示连接成功。
3. 点击“IO 测试”按钮，进入 IO 口测试界面，此时任意点击开关或者拉动滑动条，应该可以看到开发板上的 LED 变化。

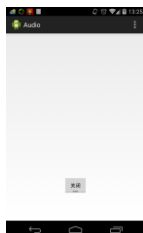
7.2 音频对讲

音频功能的实现是通过 mcu 的两个 dac 和一个 adc，进行音频的输出和采集，两个 DAC 输出分别占用核心板的 1、2 脚，ADC 占用 4 脚。音频通信采用 adpcm 的压缩和解压算法，可以将 PCM 数据压缩至 1/4 大小，比如单声道 32kHz 的采样频率和 16bit 精度下，PCM 数据流的带宽为 512Kbps，压缩后为 128Kbps。我们提供的范例是采用单声道 16bit 精度（MCU 实际只支持 10bit），采样频率可选 32000，44100，48000 等，用户可以修改代码实现（需要同时修改开发板代码和 WSUM205_TEST 代码），开发板默认是 32000，用于语音通信是没问题的。

这里有必要介绍一下 ADPCM 格式的音频，ADPCM 即自适应差分脉冲编码调制，压缩比 1:4，是一种简单实用的音频压缩格式，广泛应用在语音通信领域。ADPCM 格式是采用分块的方式组织音频数据，即将整个音频文件分为若干个固定大小的数据块，数据块之间通过采样值和索引值相关联，这样有利于音频文件的实时在

线播放,即每接收到一个数据块就可以进行解码播放了。我们提供的源码对ADPCM规定的格式为DATA_CHUNK+DATA的形式,DATA_CHUNK占用16byte,DATA为256BYTE,即一个数据块为272BYTE,DATA_CHUNK以4字节“data”开始,随后是块大小、预采样值、索引值等字段,随后就是音频数据。以“data”开头主要是方便我们进行数据块的同步,这种组织方式跟标准的ADPCM格式是不一样的,如果你希望采集到的音频数据能在电脑上播放,则需要对其整理一下并加上PCM文件头,这样就可以制作成一个标准的PCM音频文件。

类似LED灯的实验,WSUM205_TEST通过TCP连接到开发板后,点击“音频测试”按钮即可进入下面的音频测试界面:



提供了一个switch按键,按下之后手机就会开始将受话器收到的声音压缩并发送到开发板,此时如果开发板的耳机插孔插入了耳机就可以听到对方的声音。

注意:音频功能要占用比较多的内存,所以开发板默认是不开启音频功能的,在测试本应用之前需要发送命令“a”给开发板以打开音频功能(或者按monitor软件右下角的“打开音频”按钮)。

简单步骤如下:

1. 点击monitor的“连接WIFI”按钮,将WIFI板连接到路由器(需要修改程序以指定正确的路由器名称密码等)。
2. 点击monitor的“打开音频”按钮,以开启开发板的音频功能。
3. 打开WSUM205_TEST软件(此时确保手机也连接到路由器),输入正确的IP地址和端口号(4800),并点击连接按钮,应该提示连接成功。
4. 点击“音频测试”按钮,进入音频测试界面,此时如果按下开发板的SPEAKER按键,并对着开发板上的受话器讲话,手机上应该能听到声音。
5. 按下手机上音频测试界面的“按下说话”按钮或者点击其下面的开关,并对着手机的受话器讲话,开发板上的耳机插孔会输出声音。

以上是手机跟开发板之间的语音通信,当然你也可以进行开发板与PC机的通信(利用“网络调试助手”可实现录音和播放),以及开发板与开发板之间的语音对讲(需要两块或以上开发板)。

7.3 MP3 播放

开发板集成了libmad免费源码,作为开发板的一个例子,可以从中学习mp3的一些解码算法。

播放mp3的过程是先建立WIFI连接,在开发板开启音频和MP3播放功能,然后使用“网络调试助手”将MP3文件发送到开发板,实现边发送边播放音乐。开发板可以使用mcu自带的DAC或者I2S芯片输出音频。

播放 mp3 步骤如下：

1. 建立 WIFI 连接，可以参考前面两个范例，只是将手机换成电脑而已。
2. 点击 monitor 的“打开音频”按钮，以开启开发板的音频功能。
3. 使用 monitor 发送命令‘k’，切换到 MP3 播放功能（MP3 播放与语音对讲不能同时存在，因 MP3 音频格式和 PCM 音频格式不同）。
4. 使用 monitor 发送命令‘L’（大小写均可），调节音量到合适值。
5. 打开工具包里面的“网络调试助手”并配置成如下形式



点击网络调试助手的“连接”，连接成功后，选择“启动文件数据源”，会弹出文件选择对话框，选择需要播放的 mp3 文件，然后点击发送。此时应该可以通过耳机听到播放的音乐，如果播放过程出现卡顿请发送命令‘t’将 monitor 输出暂时关闭。

开发板使用了一款 TI 的 I2S 接口音频芯片，可以输出高质量的音频信号，可以作为无线音乐播放器参考方案。

7.4 WEB 控制

我们提供的 WEB 服务器范例可实现表单提交和文件传输等交互功能，在这里演示一下通过 WEB 页面控制开发板的过程。

1. 建立 WIFI 连接，可以参考前面的范例。
2. 打开浏览器输入开发板的 IP 地址并打开页面，会显示如下的页面信息



3. 点击左下角的“重启设备”，会打开重启设备的子页面，点击子页面里面的“确认”即可。

7.5 MACHTALK 云控制

通过本案例我们演示一下物联网远程控制，使用的是 machtalk 的物联网平台，其官方网址：www.machtalk.net。先介绍一下远程控制的原理，用户先到 machtalk 的官网注册账号(取得用户名和密码)，并在其管理页面添加设备(取得设备 ID 和 PIN 码)，之后手机通过用户名和密码连接至 machtalk 服务器，设备(开发板)通过 ID 号和 PIN 码连接到服务器，之后就可以进行远程控制了。

这里我们通过已经注册的一个帐号进行操作，注册信息如下：

用户名：xrf_test

密码：xrf_test

设备 ID：d21083c82860428aaf1b749b9e608aa4

设备 PIN 码：63075668，连接时需要将 pin 码进行 md5 加密，加密后的 32B 密文为：d25c89d413e17541d5ab6f49b3c31d57

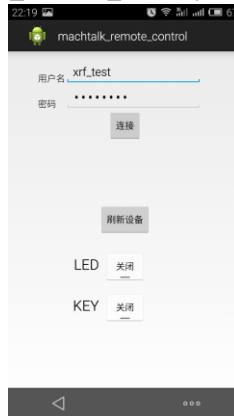
下面列出具体的操作步骤：

1. 开发板先连接到路由器（保证可以访问互联网）
2. 开发板发送命令‘y’，开始登录和连接服务器的过程，连接成功后将会通过心跳维持连接，如下图

```
[D: 958.050] is work socket  
[D: 958.050] rcv:{0}
```

```
[D: 958.060] is work socket  
[D: 958.060] heartbeat:0  
[D: 958.230] is work socket  
-----
```

3. 手机上安装和打开 machtalk_remote_ctrol.apk 演示软件，显示画面如下



4. 点击 ‘连接’按钮，如果连接成功，会在按钮下方显示“已经连接”
5. 点击 “刷新设备”，稍后会在按钮上方显示设备信息（设备 ID 号，以及设备是否在线），如果设备在线，可以继续下面的步骤
6. 点击 LED 右边的开关，可以看到开发板上 5 号 LED 的亮灭，反应可能会有延迟
7. 按下开发板上的 KEY1，可以看到 KEY 右边按钮的变化

上面是演示的手机端控制开发板上的 LED 等，以及开发板上报按键状态的例子，还有更多丰富的功能可以到 machtalk 官网了解。

与远程控制有关的开发板代码在/app/machtalk_remote_control.c 文件，代码有注释。
machtalk_remote_ctrol.apk 代码更新在网盘的/源码/目录下面，同时网盘的/文档/下面有 machtalk 平台的通信协议以及案例教程（免费的哦）。