

零死角玩转STM32

与野火同行 乐意惬意无边



原创教程，完全开源。



由浅入深，结合实操。



通俗易懂，详尽解读。



配套板子，全面玩转。



强强联合，不断更新。



野火团队 Wild Fire Team

0、友情提示

《零死角玩转 STM32》系列教程由初级篇、中级篇、高级篇、系统篇、四个部分组成，根据野火 STM32 开发板旧版教程升级而来，且经过重新深入编写，重新排版，更适合初学者，步步为营，从入门到精通，从裸奔到系统，让您零死角玩转 STM32。M3 的世界，与野火同行，乐意惬意无边。

另外，野火团队历时一年精心打造的《STM32 库开发实战指南》将于今年 10 月份由机械工业出版社出版，该书的排版更适于纸质书本阅读以及更有利于查阅资料。内容上会给你带来更多的惊喜。是一本学习 STM32 必备的工具书。敬请期待！



9、WIFI

9.1 资料与工具下载

为了更好地理解和使用 Wi-Fi 模块的各项功能，您首先需要下载和学习以下资料，这些文档资料需要通过 Internet 下载：

提供下载地址：

《AN0003_EMW_DataTransferExample.pdf》：透明传输模块使用范例，详细描述了模块在各种模式下的透明传输的使用方法

《RM0001_EMW3280》：EMW 模块使用说明，详细描述了模块的各项功能

《RM0002_EMWToolBox》：EMW 模块配置软件使用说明，详细描述了如何配置模块的各项参数

《DS001_EMW3280_V2.pdf》：EMW 模块的电气特性及引脚定义封装说明。

《RM0001_EMW3280_V02060288》：EMW 模块的工作模式及命令控制集

需要的工具软件：

EMW Tool Box: PC 端配置 Wi-Fi 模块参数工具软件

TCP/UDP 测试工具：用于在 PC 端与 WiFi 模块建立 TCP/UDP 连接，实现数据收发串口调试助手或者超级终端：用于配合 WiFi 数据收发调试工具软件

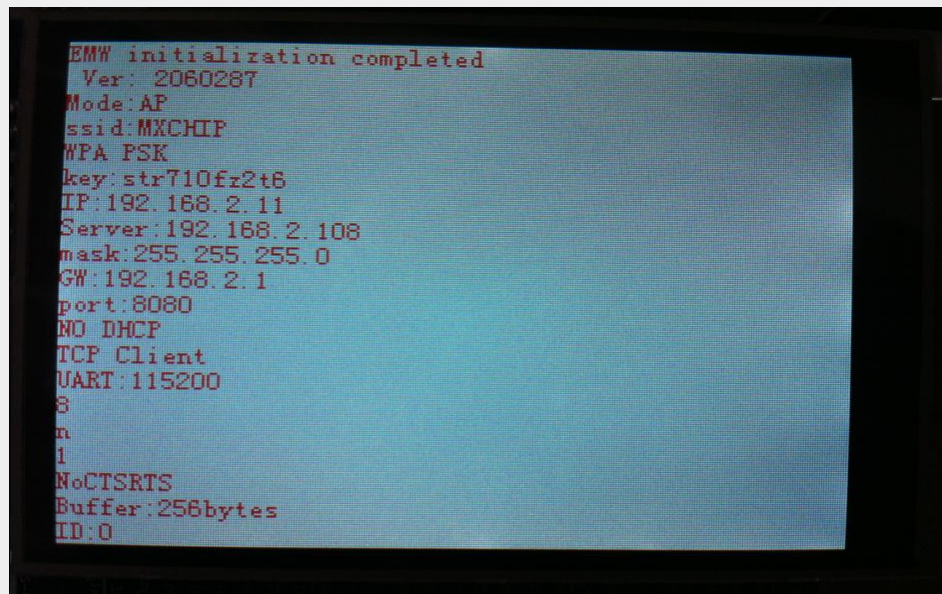
9.2 实验描述

该实验讲解了如何运用 Mxchip 提供的 EMSP_API 函数来配置 Wi-Fi 模块的参数，连接无线网络，与同网段中的 PC 建立 TCP 连接，并打开 PC 端安装 TCP/UDP 测试工具。TCP/UDP 测试工具发送的数据，野火 STM32 开发板通过 Wi-Fi 将接收到的数据返回给 PC，达到回显的功能。

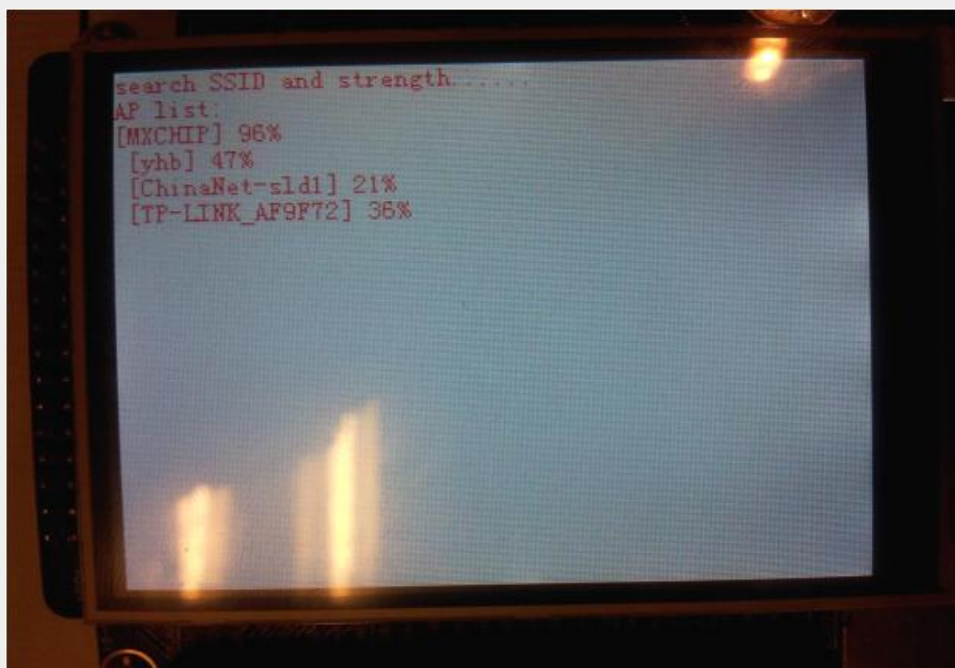


9.2.1 实验现象

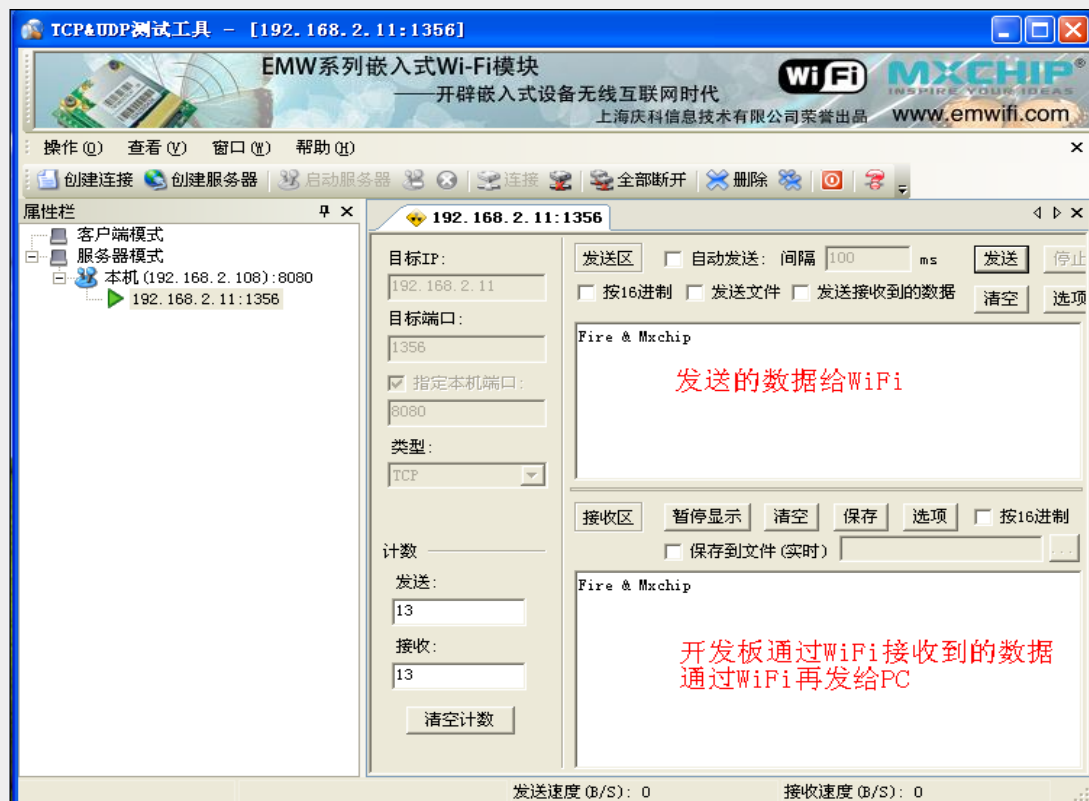
读取到 Wi-Fi 模块的配置参数，并且显示到 LCD 屏



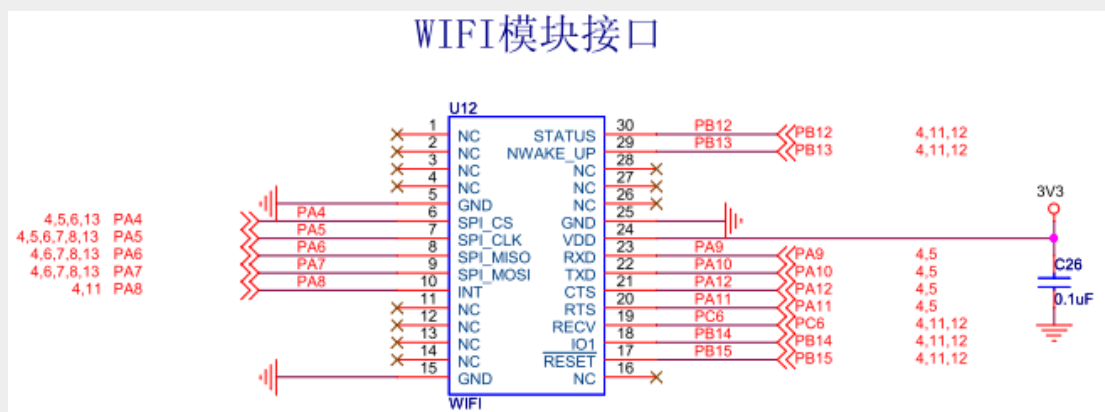
读取搜索到的周围的无线网络和信号强度



野火 STM32 开发板接收到 PC 端 TCP/UDP 测试工具发送的数据，通过 Wi-Fi 模块发送给 PC，达到回显的功能



9.2.2 硬件连接图



● 串口连接说明

RXD - PA9

TXD - PA10

CTS - PA12

RTS - PA11

STATUS - PB12 状态脚

Wake up - PB13 唤醒脚

IO1 - PB14 帧控制

- SPI 接口说明

PA4 - SPI_CS

PA5 - SPI_CLK

PA6 - SPI_MISO

PA7 - SPI_MOSI

PC7 - INT

PC6 - RECV

PB12 - STATUS

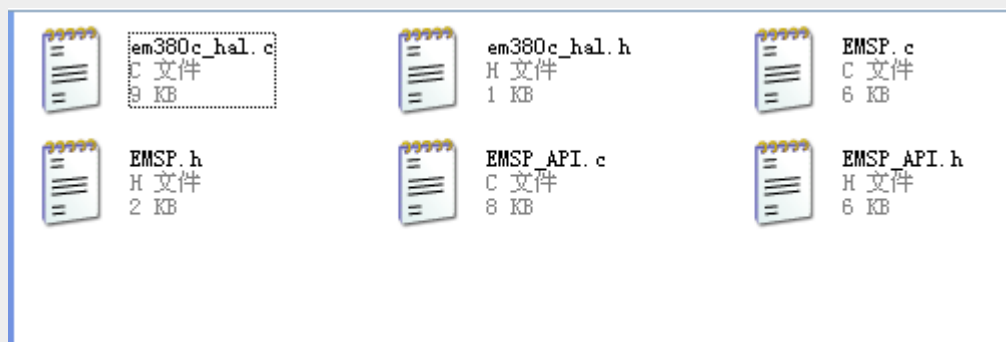
PB13 - Wakeup

SPI 接口模块的 WiFi 例程暂时没有，等到以后做了之后会发布

PS：引脚的具体定义及功能请参考“datasheet”文件夹中的 EMW3280_V2.pdf 文档

9.2.3 EMSP_API 函数

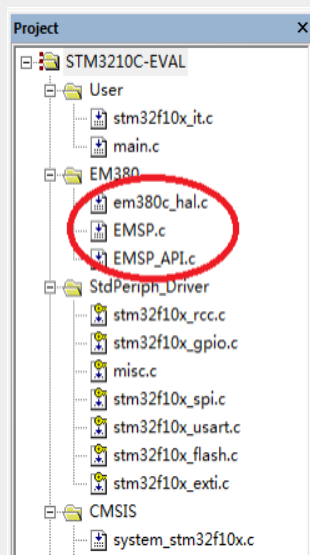
EMSP_API 接口函数提供了一系列 API 函数，用户通过调用这些函数可以轻松地在各种嵌入式设备上实现对 EMW 系列 Wi-Fi 模块的控制和数据传输。现在该接口函数随和 WiFi 资料随野火 STM32 开发板例程一并提供：



如果大家购买了野火 STM32 开发板和 Mxchip EMW 系列 Wi-Fi 模块，就可以在开发板上调试这些例程。

EMSP_API 函数库由标准 C 编写而成，可以直接加入到常用的嵌入式开发环境，如 KEIL，IAR 等。

EMSP_API 函数由以下三个 C 语言文件及其对应的头文件构成。



➤ emw38x_hal.c

该代码实现了 EMW 系列模块和嵌入式设备之间的硬件接口。用户需要根据自己的硬件环境实现相应的函数

➤ EMSP.c

该代码实现了 EMSP 命令的协议处理。

➤ EMSP_API.c

该代码提供给用户用于操控模块的 API 函数，用户只需调用这些函数，就可以实现对模块的配置和操作。

9.2.4 API 函数一览

函数名	vs8 EM380C_Init(void)
功能	用于初始化模块，和与模块通讯的 UART 接口，并使模块处于能够响。
返回值	-1： 执行命令失败
	0： 执行命令成功

函数名	vs8 EM380C_Get_ver(u32* version)
功能	用于获得 EMW 系列模块的固件版本号。
返回值	-1 : 执行命令失败
	0 : 执行命令成功

函数名	vs8 EMSP_Get_status(EM380C_status_TypeDef* EM380C_status)
功能	用于获得 Wi-Fi 模块的网络连接状态。
输入	用于存放 Wi-Fi 的网络连接状态结构体地址 typedef struct { EM380C_TCPstatus_TypeDef TCPstatus; EM380C_WiFistatus_TypeDef WiFistatus; } EM380C_status_TypeDef;
返回值	-1 : 执行命令失败
	0 : 执行命令成功

函数名	vs8 EM380C_Get_APList(EM380C_APLst_TypeDef* EM380C_APLst)
功能	用于获得区域内无线 AP 的 SSID 号和相应的信号强度。
输入	用于存放无线的 AP 的 SSID 号和相应的信号强度的线性表的起始地址 typedef struct { char AP_NAME[20]; float AP_signal; } EM380C_APLst_TypeDef;

返回值	-1 : 执行命令失败
	>=0: 执行命令成功, 获得的 AP 信息的数量

函数名	vs8 EM380C_Startup(void)
功能	启动 Wi-Fi 模块的 TCP/IP 网络连接。
返回值	-1 : 执行命令失败
	0 : 执行命令成功

函数名	vs8 EM380C_Get_RF_POWER(EM380C_RF_POWER_TypeDef*RF_POWER)
功能	用于获得 Wi-Fi 模块当前的配置参数。
输入	参数结构体的地址, 成功执行命令后, 模块当前的参数会写入这个地址。参数结构体如下。 typedef struct { // WIFI u8 wifi_mode; //Wlan802_11IBSS(0), Wlan802_11Infrastructure(1) u8 wifi_ssid[32]; // u8 wifi_wepkey[16]; // 40bit and 104 bit u8 wifi_wepkeylen;// 5, 13 // TCP/IP u8 local_ip_addr[16]; u8 remote_ip_addr[16]; // if em380 is server, it is NOT used;if em380 is client, it is server's IP

	<pre> u8 net_mask[16]; // 255.255.255.0 u8 gateway_ip_addr[16]; // gateway ip address u8 portH; // High Byte of 16 bit u8 portL; // Low Byte of 16 bit u8 connect_mode; // 0:server 1:client u8 use_dhcp; // 0:disable, 1:enable u8 use_udp; // 0:use TCP,1:use UDP // COM u8 UART_buadrate; // 0:9600, 1:19200, 2:38400, 3:57600, 4:115200 u8 DMA_buffersize; // 0:2, 1:16, 2:32, 3:64, 4:128, 5:256, 6:512 u8 use_CTS_RTS; // 0:disable, 1:enable u8 parity; // 0:none, 1:even parity, 2:odd parity u8 data_length; // 0:8, 1:9 u8 stop_bits; // 0:1, 1:0.5, 2:2, 3:1.5 // DEVICE // u8 device_num; // 0 - 255 u8 IO_Control; // 0 - 255 u8 sec_mode; // 0 = wep, 1=wpa psk, 2=none u8 wpa_psk[32]; } EM380C_parm_TypeDef; </pre>
返 回	-1 : 执行命令失败
值	0 : 执行命令成功



函数名	vs8 EM380C_Set_Config(EM380C_parm_TypeDef* EM380C_Parm)
功能	用于设置 Wi-Fi 的配置参数。
输出	参数结构体的地址，成功执行命令后，会将该地址上的数据写入到 WiFi 模块里面去。结构体与上面 GetConfig 参数一致。
返回值	-1 : 执行命令失败
	0 : 执行命令成功

函数名	u32 EM380C_Send_Data(u8* Data,u32 len)
功能	用于通过 Wi-Fi 模块发送数据
输出 1	保存发送数据的内存空间的起始地址
输出 2	发送的数据长度
返回值	>0: 执行命令成功，返回发送的数据长度
	0 : 执行命令成功

函数名	vs8 EM380C_Reset(void)
功能	重启 Wi-Fi 模块，配置参数后，需重启模块，参数才能生效
返回值	-1: 执行命令成功，返回发送的数据长度
	0 : 执行命令成功

函数名	vs8 EM380C_Set_Mode(EM380C_mode_TypeDef mode)
功能	设置 Wi-Fi 模块模式，命令模式和透传模式
输入	用于存放 Wi-Fi 模块的模式结构体 typedef enum { config_mode = 0x0, //命令模式



	DTU_mode = 0x1, //透传模式 } EM380C_mode_TypeDef;
返回值	-1: 执行命令成功
	0: 执行命令失败

9.2.5 MAIN 函数讲解

第一步，初始化硬件接口(其实这一部拉低 STATUS 引脚，初始化号相应硬件接口即可)

```
064 while(EM380C_Init(BaudRate_115200,WordLength_8b,StopBits_1,Parity_No,HardwareFlowControl_None,buffer_512bytes)==EM380ERROR);  
065  
066 printf("EMW initialization completed\n");
```

第二步，设置 Wi-Fi 模块参数

```
069  
070 //*****Config the Wi-Fi moudel parameter*****//  
071 parm.wifi_mode = AP;  
072 strcpy((char*)parm.wifi_ssid,"MXCHIP");  
073 strcpy((char*)parm.wifi_wepkey,"");  
074 parm.wifi_wepkeylen = 0;  
075 strcpy((char*)parm.local_ip_addr,"192.168.2.11");  
076 strcpy((char*)parm.remote_ip_addr,"192.168.2.108");  
077 strcpy((char*)parm.net_mask,"255.255.255.0");  
078 strcpy((char*)parm.gateway_ip_addr,"192.168.2.1");  
079 parm.portH = 8080>>8;  
080 parm.portL = 8080;  
081 parm.connect_mode = TCP_Client;  
082 parm.use_dhcp = DHCP_Disable;  
083 parm.use_udp = TCP_mode;  
084 parm.UART_buadrate = BaudRate_115200;  
085 parm.DMA_buffersize = buffer_256bytes;  
086 parm.use_CTS_RTS = HardwareFlowControl_None;  
087 parm.parity = Parity_No;  
088 parm.data_length = WordLength_8b;  
089 parm.stop_bits = StopBits_1;  
090 parm.IO_Control = IO1_Normal;  
091 #ifdef EMW_FIRMWARE_UART  
092 parm.sec_mode = Secure_WPA_WPA2_PSK;  
093 strcpy((char*)parm.wpa_psk,"str710fz2t6");  
094 #endif  
095 while(EM380C_Set_Config(&parm)==EM380ERROR);  
---
```

设置模块参数，并通过 EMSP_SET_CONFIG 命令发送给 Wi-Fi 模块。

模块详细的功能可参考 "Wi-Fi 模块 datasheet" 文件夹 EMW_DataTransferExamples.pdf，里面详细介绍了各种模式的数据透传。



第三步，重启 Wi-Fi 模块，模块的参数配置好之后，需要重启 Wi-Fi 模块才能生效

```
096  
097 while(EM380C_Reset()==EM380ERROR); //Reset WiFi module , enable parameter  
098
```

第四步，启动 Wi-Fi 模块，通过发送 EMSP_CMD_START 命令，此时模块内部 TCP/IP 协议栈已经跑启，模块上红灯常亮

```
098  
099  
100 EM380C_Startup(); // startup WiFi , connect to AP  
101  
102
```

第五步，拉高 STATUS 引脚，模块进入透传模式

```
105  
106  
107 while(EM380C_Set_Mode(DTU_mode)==EM380ERROR); //EM380C config mode -> DTU mode  
108  
109  
110
```

PS：直接拉高 STATUS 引脚可直接启动 Wi-Fi，跳过第四步

9.3 其他说明：

EMSP 详细命令可参考“datasheet”文件夹中 EMW3280.pdf 文档。

模块还可通过 PC 端的配置工具 EMW Tool Box 配置模块参数,具体使用可参考“datasheet”文件夹中 EMWToolBox2.pdf 文档。

9.4 技术支持

整个 WiFi 的例程讲解到这里就结束了，大家如果想要更详细的资料可以跟 Fire 或者 Mxchip 公司联系。技术问题请到论坛发帖。