Open1081 UserManual

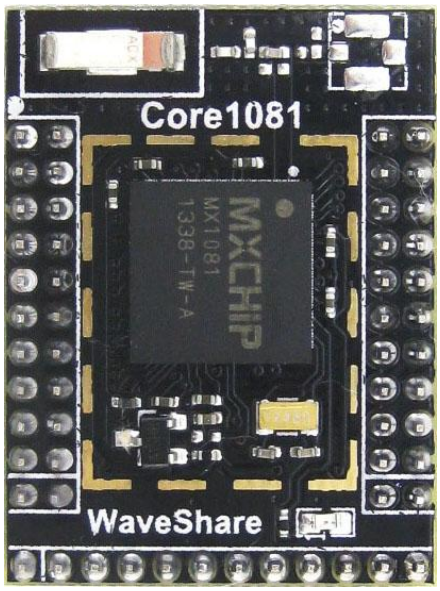
目录

一、 概述

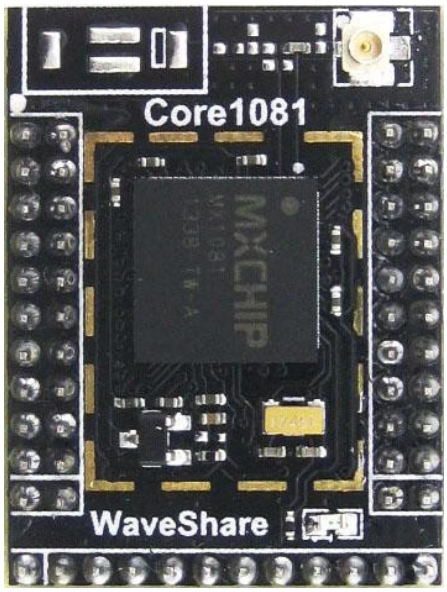
革命性 WIFI 产品已问世，你还在等什么？Open1081 为你提供全套 WIFI 智能产品的解决方案，开启属于你自己的智能生活，在蓝牙耳机盛行的年代，你是否想拥有你自己个性的 WIFI 耳机呢？基于 WIFI 的智能相机能否激发你创新的萌芽呢？能与智能手机，平板电脑等智能设备，云端建立无缝链接的开发套件，你是否想立即拥有呢？现在，这一切已经成为可能，赶快加入我们的 WIFI 智能战队吧。

二、 硬件介绍

2.1. 核心板介绍



Core1081-A



Core1081-B

Core1081 是一种基于 MX1081 的超小型，超低功耗的嵌入式 WIFI 模块。MX1081 是一款超高集成度的 WIFI 微控制器，集成 IEEE 802.11 MAC、基带、射频以及一个可以运行 WIFI 网络协议栈和应用程序的微控制器核心。MX1081 内置 1M 字节的闪存，128K 字节 RAM 和丰富的外设。

Core1081 配合 mxchipWNet 嵌入式 WIFI 固件，用户可以方便、快速地为嵌入式设备增加 WIFI 网络通讯功能。如使用 mxchipWNet-DTU 串口透明传输固件，可以立即为你的串口设备添加无线网络功能，缩短了开发周期，实现快速上市。用户也可以使用 mxchipWNet 软件库直接在模块上开发各种嵌入式 WIFI 应用，进一步降低成本和产品体积 。

◆ 领先技术：

- SOC 技术，内部集成了 STM32F205R 的 MCU，超小体积 EasyLink 一键配置
- 无缝漫游切换
- 高速网络传输，实际最高可达 20Mbps
- 内置实时操作系统全新软件系统

◆ 超低功耗：

- 工作电压：3.3V
- 保持无线网络连接仅需约 7mA 电流
- 以 20kbps 传输数据时(传输周期 100ms)，平均功耗约 24mA
- 待机功耗约 8μ A
- 120MHz 主频的 Cortex-M3 内核，内嵌 Flash 1M bytes，RAM 128kBytes；

◆ 包含如下外设：

- 42 个 GPIO 口
- 1 个带硬件流控制的 UART 接口，2 个普通 UART 接口
- 2 个 SPI 接口，1 个 IIS 接口
- 8 个 ADC 输入通道，2 个 DAC 输出通道
- 1 个 USB OTG 接口, 1 个 CAN 接口
- 2 个 I2C 接口
- DCMI 接口
  JTAG/SWD 调试接口

◆ 完整功能的 WIFI 联通性：

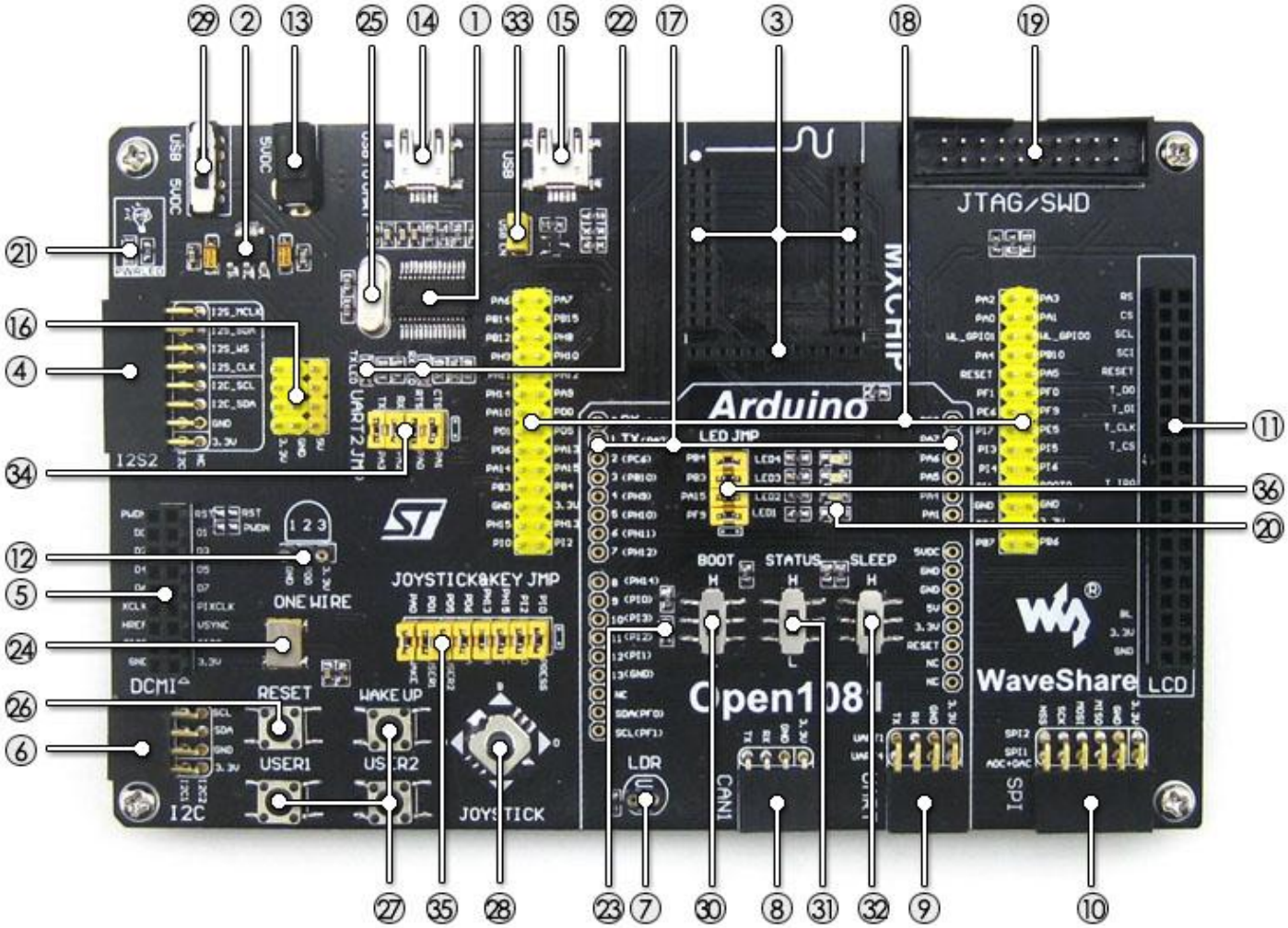
- 支持 IEEE 802.11 b/g/n，WIFI 频段 1-13
- 发射功率：18dBm@11b，15dBm@11g，14.5dBm@11n
- 接收灵敏度：-96dBm
- 数据传输率:11M@802.11b, 54M@802.11g, 72M@802.11n
- WPS 2.0
- 加密方式：WEP，WPA/WPA2 PSK/Enterprise
- 多种节能模式

◆ Core1081 核心参数

- 主控芯片：MX1081
- 工作电压：2.0V~3.6V
- 工作频段：2.400~2.484GHz
- 接口：所有 IO 接口
- 接口间距：排针间距 2.0mm

天线：可选内置天线或外置天线（Core1081-A 或 Core1081-B）

2.2. 底板资源简介



【芯片简介】

1. **PL2303**
   USB TO UART，方便调试。
2. **AMS1117-3.3**
   3.3V 稳压器件。

[核心接口简介]

3. **核心板插槽**
   方便接入核心板。
4. **I2S2 / I2C1 接口**
   方便接入 I2S 模块，如音频模块等。
5. **DCMI 接口**
17. **Arduino 接口**
    方便接入 Arduino 模块
18. **MCU 引脚接口**
    引出所有 I/O，方便与外设进行 I/O 连接。
19. **JTAG 接口**
    支持下载与调试。

[器件简介]

20. **用户 LED**
    便于 I/O 输出测试或显示程序运行状态。
21. **电源 LED**
22. **PL2303 TX-LED / RX-LED**
    UART 收发指示 LED。

方便接入摄像头模块。

6. **I2C1/ I2C2 接口**
   方便接入 I2C 模块，如 I/O 扩展芯片 PCF8574、EEPROM AT24CXX 模块等。
7. **LDR 接口**
   方便接入光敏电阻
8. **CAN1 接口**
   方便接入 CAN 模块。
9. **UART1/UART4 接口**
   方便接入 RS232、RS485、USB TO 232 模块等。
10. **SPI1 / SPI2 接口**
    方便接入 SPI 模块，如 FLASH AT45DBXX、SD 卡、MP3 模块等。
    方便接入 AD、DA 模块，因为 SPI1 复用了 AD、DA 功能。
11. **LCD 接口**
    方便接入 LCD + 触摸屏模块。
12. **ONE WIRE 接口**
    方便接入 1-WIRE 器件（TO-92 封装），如温度传感器 DS18B20、电子注册码 DS2401 等。

[其它接口简介]

13. **5V DC 接口**
14. **USB TO UART 接口**
    经过板载 PL2303 USB TO UART 芯片的转换，转为 UART。
15. **USB 接口**
    作为 Device：通过连接线，与计算机进行 USB 通信。
16. **5V 与 3.3V 电源输入输出接口**
    常用于对外供电，或与用户板进行共地处理。
23. **BOOT 指示 LED**
    指示 BOOT 的状态
24. **25M 晶振**
    为摄像头模块提供时钟
25. **12M 晶振**
    为 PL2303 提供时钟
26. **复位按键**
27. **用户按键**
    便于 I/O 输入测试或控制程序运行状态。
28. **摇杆**
    上、下、左、右、按下，共 5 个状态。
29. **"5V DC"或"USB"供电选择开关**
30. **BOOT 开关**
    切换到 H，系统从用户 FLAH 启动；
    切换到 L，系统从 Core1081 Common Bootloader 启动。
31. **STATUS 开关**
32. **SLEEP 开关**

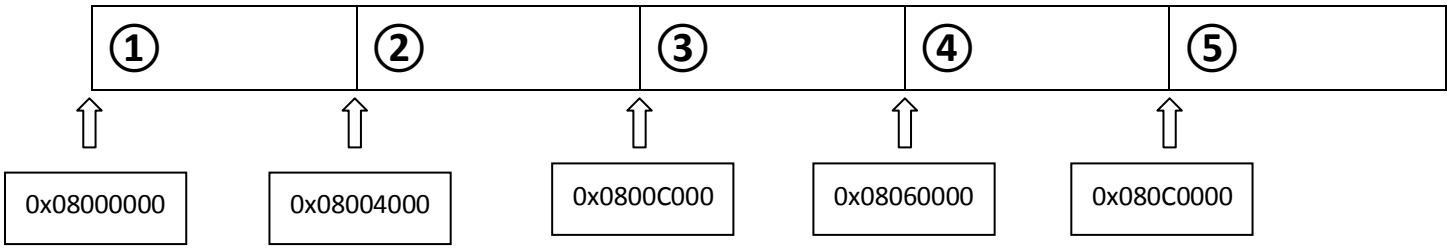
[跳线说明]

33. **USB EN 跳线**
34. **UART2 跳线**
35. **摇杆和用户按键跳线**
36. **用户 LED 跳线**
    以上接口：
    短接跳线：接入到示例程序指定的 I/O；
    断开跳线：可改为使用连接线接入自定义的 I/O。

三、 芯片 FLASH 分区介绍

EMW3280/316x modules have Cortex-M3 MCU with rich peripherals, users can build their own embedded Wi-Fi applications based on mxchipWNetTM library which manage all of the Wi-Fi MAC and TCP/IP stack processing. MXCHIP also provide mxchipWNetTM firmware to meet typical applications: wireless UART, wireless audio, wireless sensor etc.

The firmware is build up by five sections:



① 16k bytes Core1081 bootloader

② 16k bytes Parameters

③ 769k bytes mxchipWNetTM Firmware or customized firmware

④ 384k bytes (Optional) Temporary storage for OTA purpose

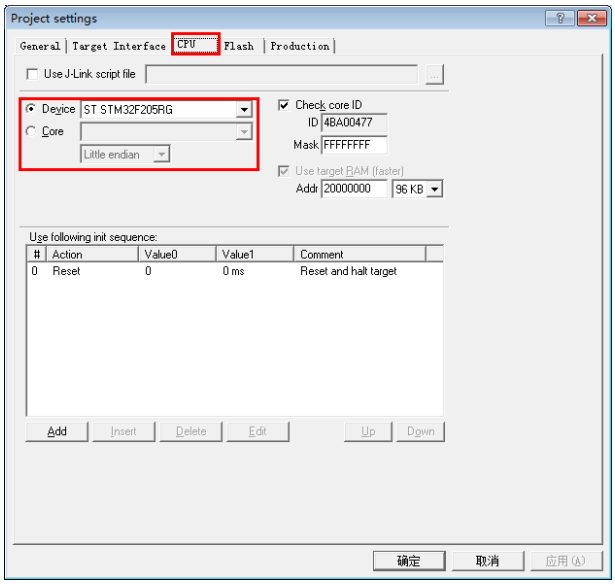
⑤ 256k bytes RF Driver

四、 BOOT 和 WIFI_Driver 升级

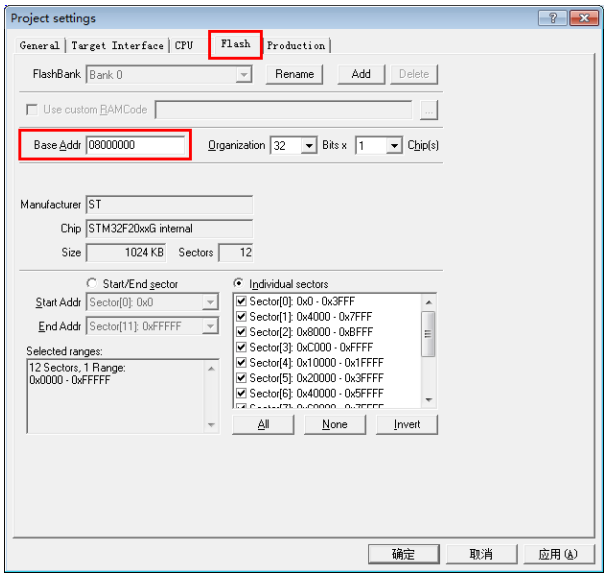
在下载 BOOT 前，最好先擦除下整个芯片。

4.1. 使用 JLINK 下载 BOOT

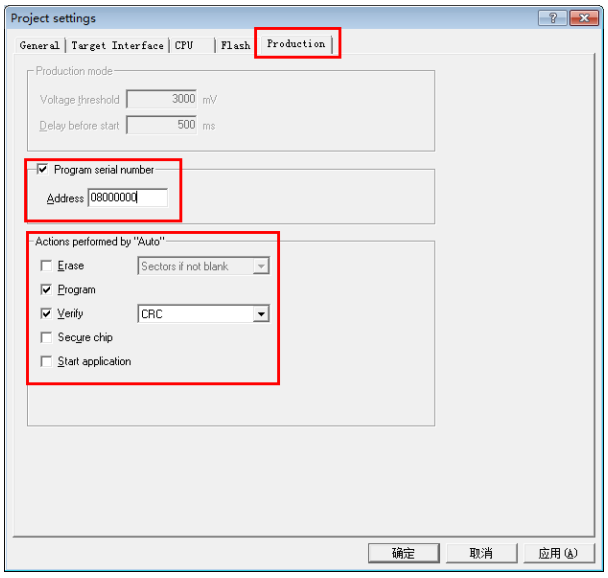
- 打开 J-Flash ARM，点击菜单栏 Options 选择 Project setting 进入 CPU 选项卡，在 Device 中选择 ST STM32F205RG，如图（1）。
- 进入 Flash 选项卡，在 Base Addr 中填入 08000000，如图（2）。
- 进入 Production 选项卡，在 Program serial number 中填入 08000000，在 Actions performed by"Auto"中只勾上 program 和 Verify，如图（3）。
- 点击菜单栏 File 选择 Open date file 选择 EMW3161_COMMON_BOOT.bin，把 Address 设置为：0x0。
- 点击菜单栏 Target，选择 connect，使 JLINK 和目标板建立连接。
- 点击菜单栏 Target，选择 Program&Verify，弹出一个对话框，点击 【否(N)】，下载成功就会填出下载所需要的时间，如图（5）。



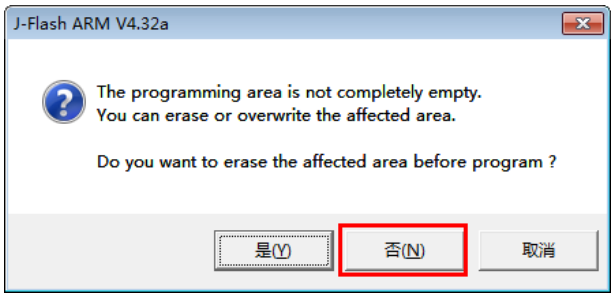
图（1）



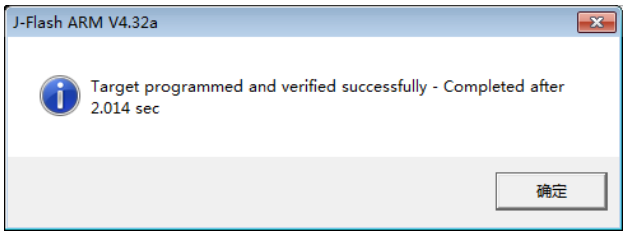
图（2）



图（3）



图（4）

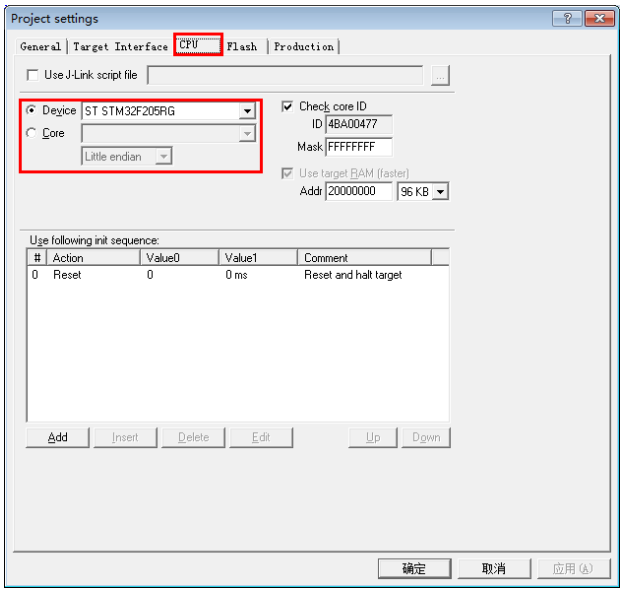


图（5）

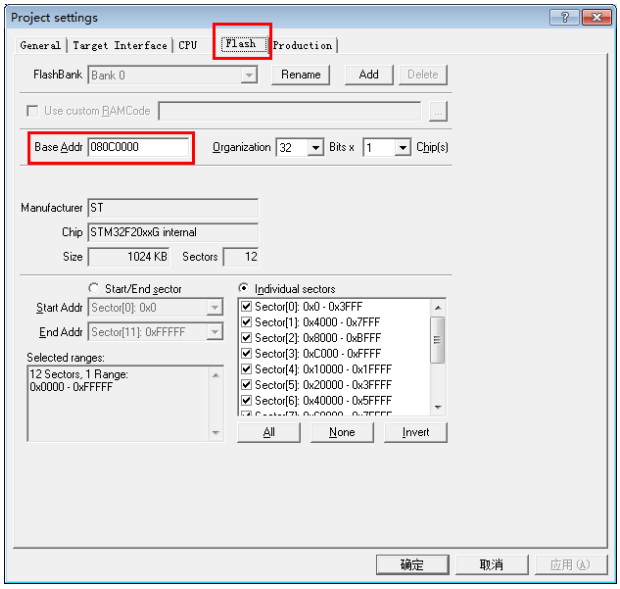
4.2. 使用 JLINK 下载 WIFI_Driver

- 打开 J-Flash ARM，点击菜单栏 Options 选择 Project setting 进入 CPU 选项卡，在 Device 中选择 ST STM32F205RG，如图（1）。
- 进入 Flash 选项卡，在 Base Addr 中填入 080C0000，如图（2）。
- 进入 Production 选项卡，在 Program serial number 中填入 080C0000，在 Actions performed by"Auto"中只勾上 program 和 Verify，如图（3）。
- 点击菜单栏 File 选择 Open date file 选择 EMW316x_WiFi_driver.bin，把 Address 设置为：0x080C0000。
- 点击菜单栏 Target，选择 connect，使 JLINK 和目标板建立连接。
- 点击菜单栏 Target，选择 Program&Verify，弹出一个对话框，点击 【否(N)】，下载成功就会填出下载所需要的时间，如图（5）。

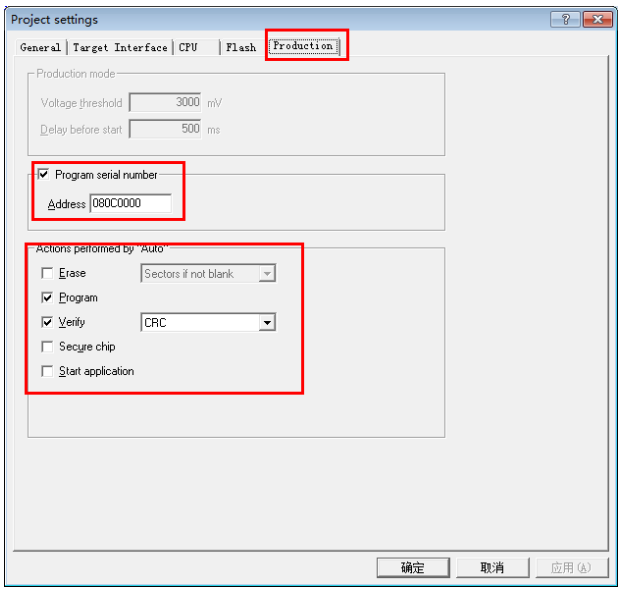
NOTE：一些设置截图



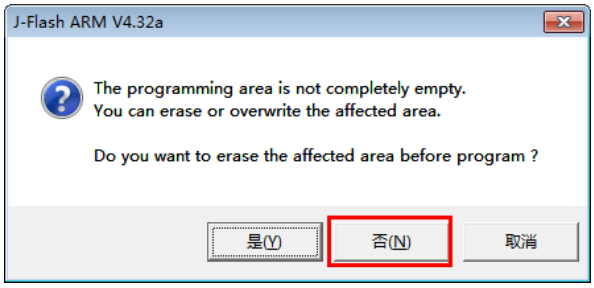
图（1）



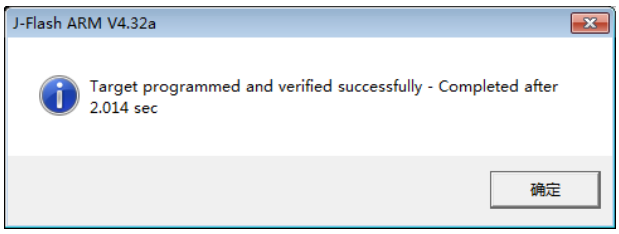
图（2）



图（3）



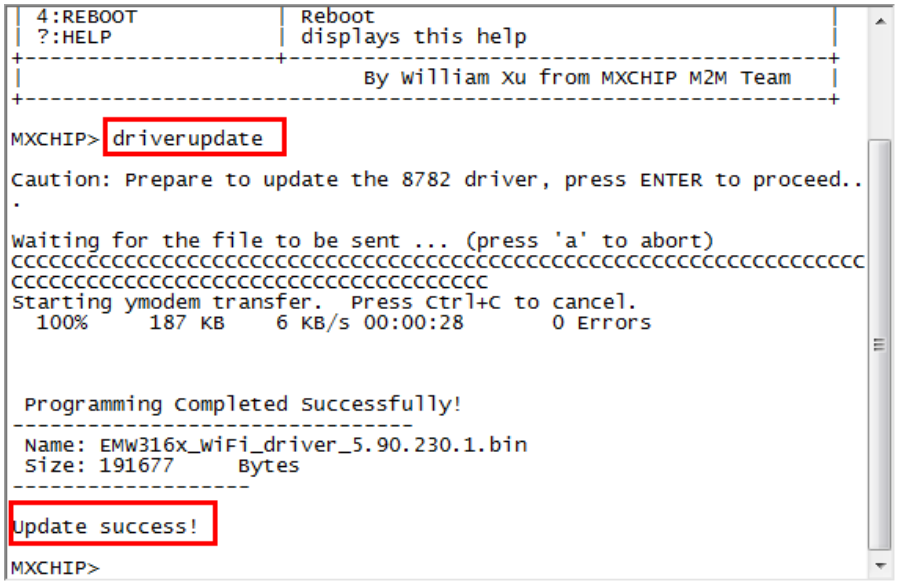
图（4）



图（5）

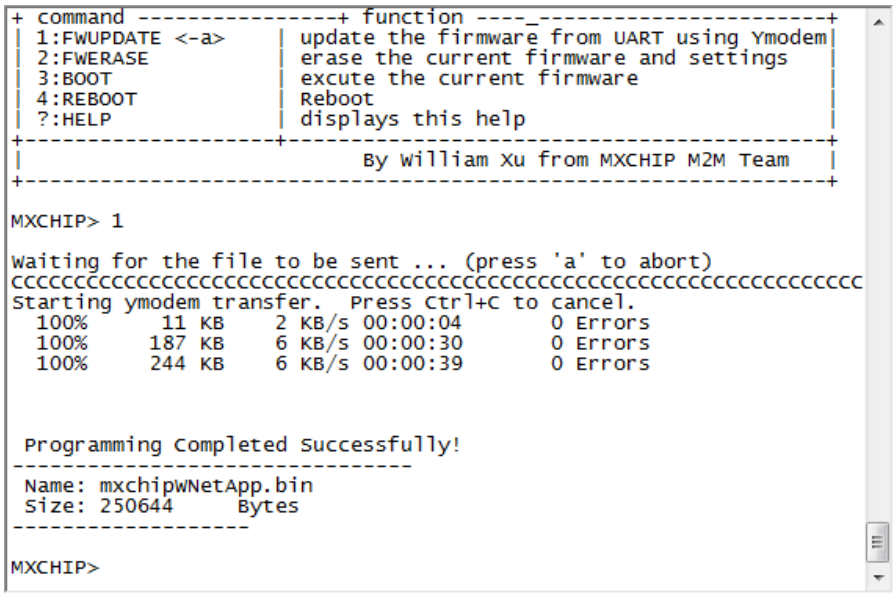
4.3. 使用 Bootloader 下载 WIFI_Driver

- 打开 SecureCRT 和板子的串口建立连接（波特率为 115200）。
- 把 BOOT 开关拨到 L，按复位按键运行 BOOT 程序。
- 在终端中可以看到 BOOT 菜单，输入 driverupdate，按 ENTER 键，弹出提示信息，再按 ENTER 键进入下载模式。
- 点击菜单栏 Transfer->send Ymodem，选择 EMW316x_WiFi_driver.bin，点击 OK，知道发送完成，具体串口提示信息如下图：



4.4. 使用 BOOT 下载用户程序

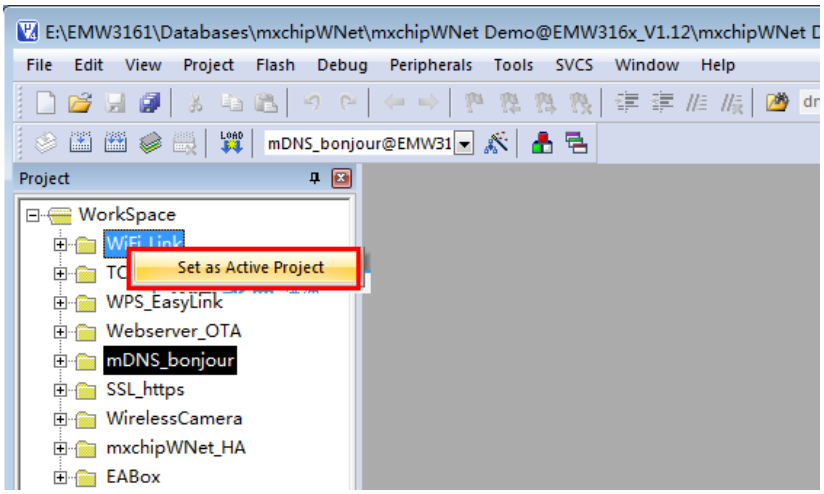
- 打开 SecureCRT 和板子的串口建立连接（波特率为 115200）。
- 把 BOOT 开关拨到 L，按复位按键运行 BOOT 程序。
- 在终端中输入回车，按[Enter]键，弹出提示信息。
- 点击菜单栏 Transfer->send Ymodem，选择 xxx.bin，点击 OK，知道发送完成，具体串口提示信息如下图：



五、 KEIL MDK 环境设置

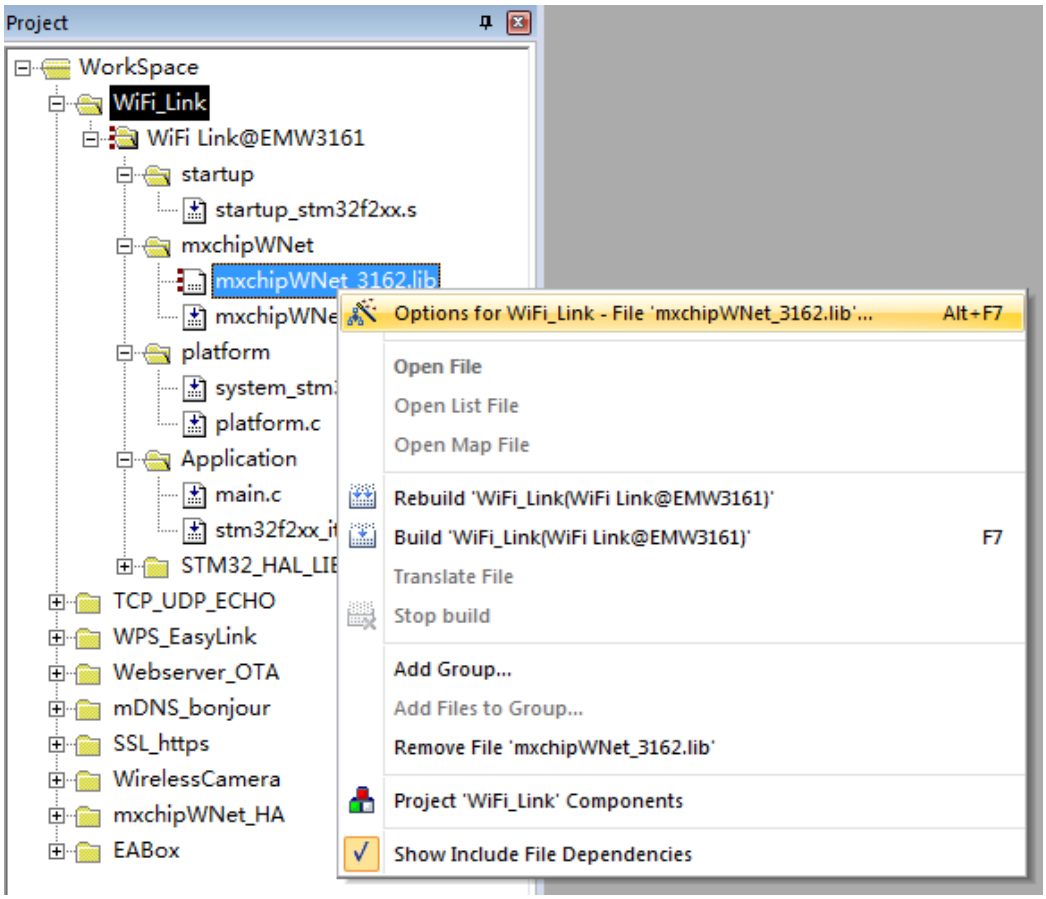
5.1. 怎么激活 WorkSpace 中的工程？

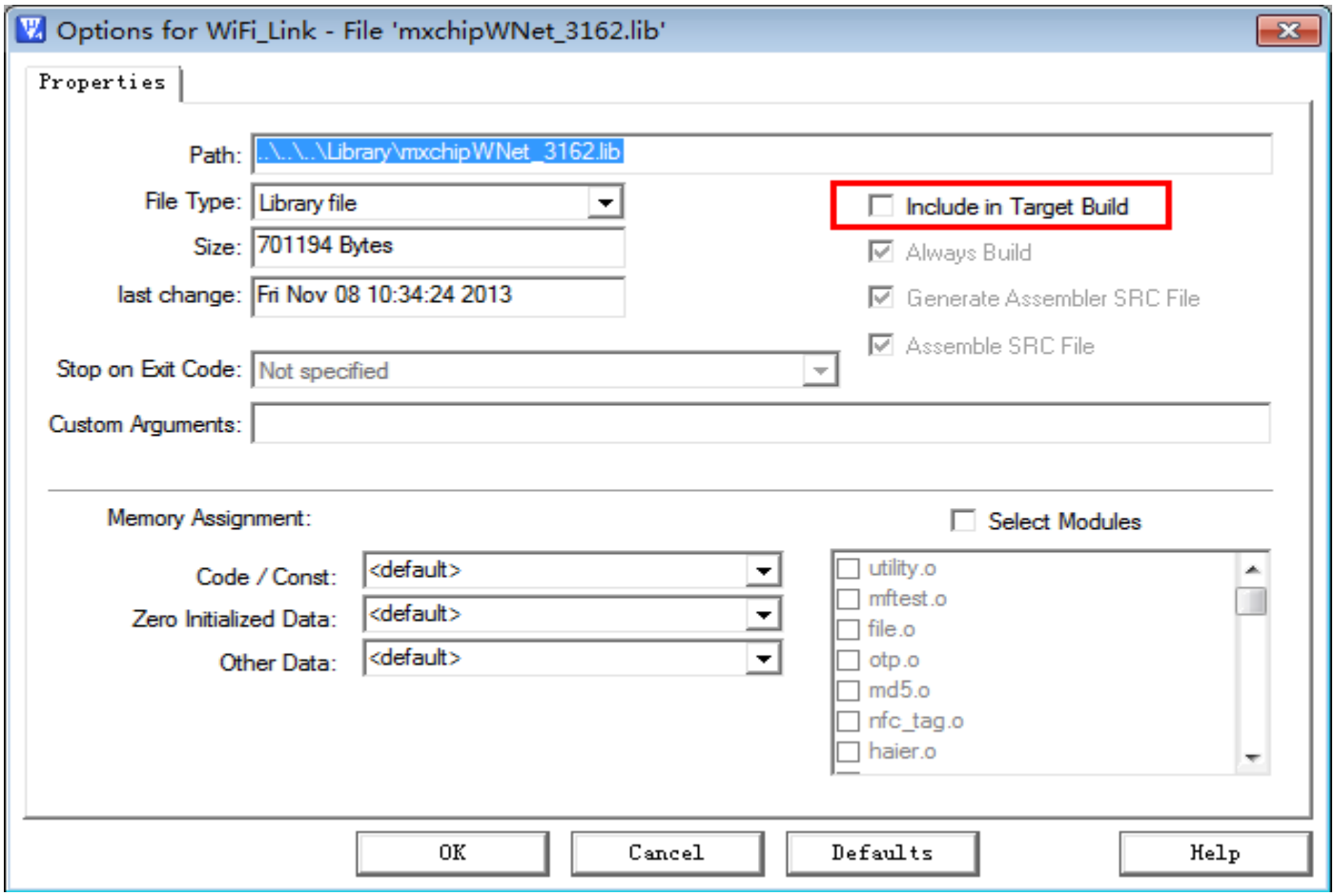
- 右击鼠标所选工程，点击 Set as Active project



5.2. 怎么把当前工程中的文件设置为不可用？

- 点击文件右键，选择 Options for……，在 include in Target Build 前面的钩去掉，打钩表示将文件添加到工程。

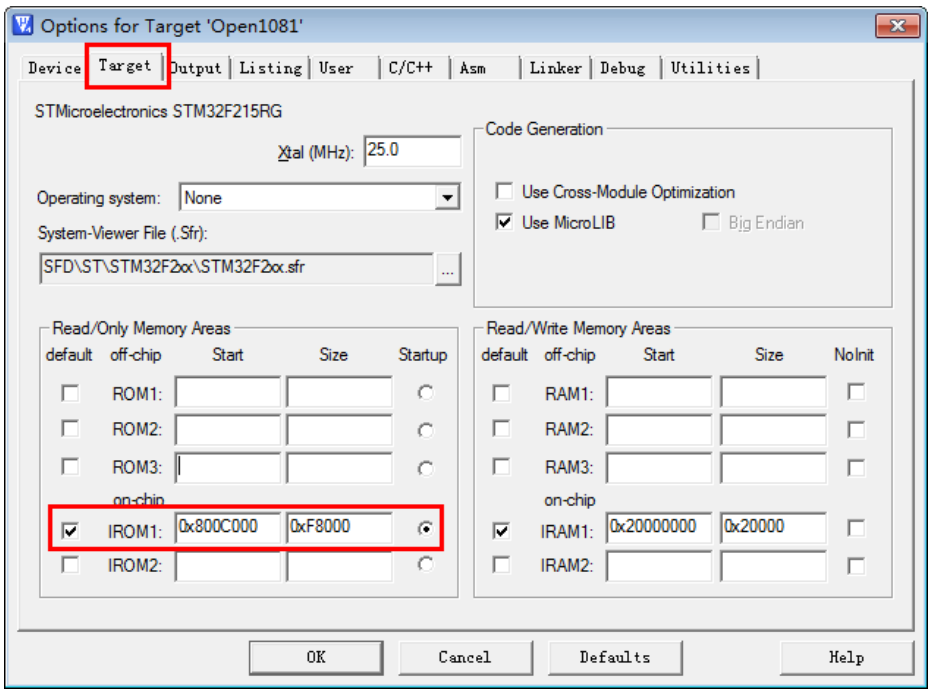




5.3. 在 KEIL 中工程设置和下载程序注意事项

5.3.1. 工程设置

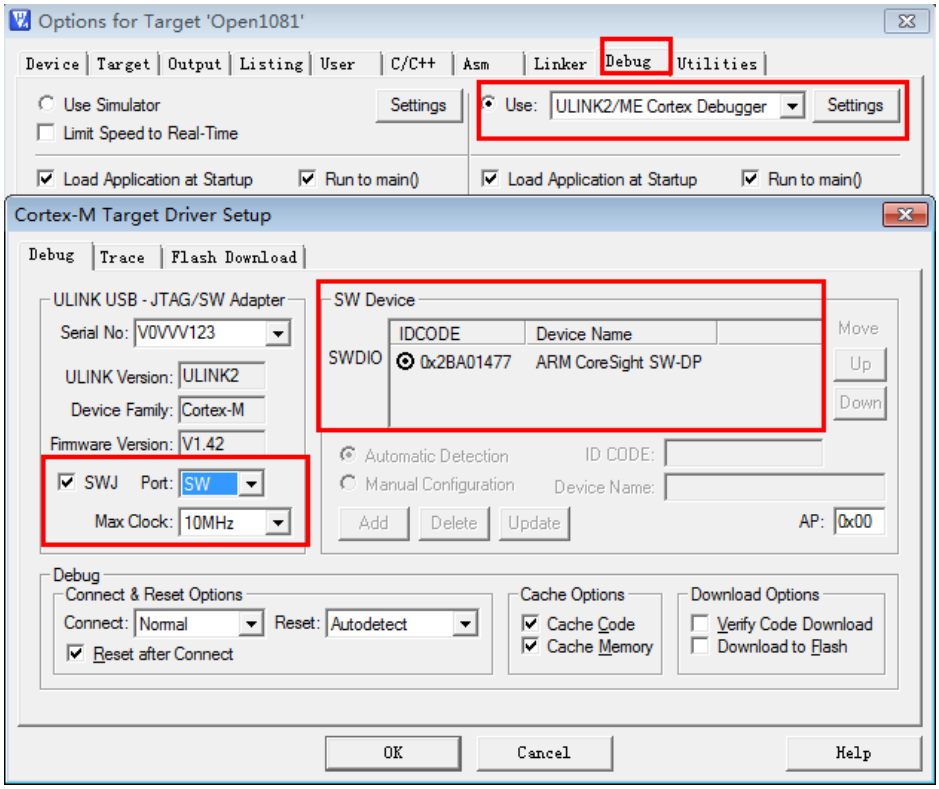
➢ 我们的用户程序是放在 0X800C00，所以在 KEIL 中也要做相应的设置，如下图：



5.3.2. 使用 ULINK 下载程序设置

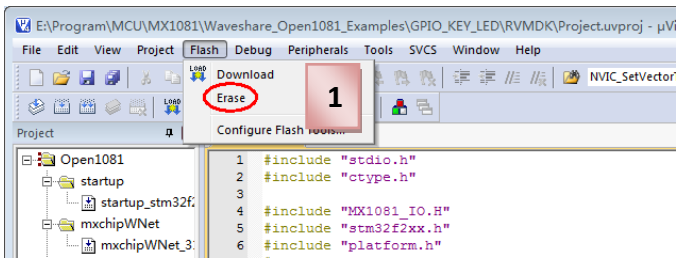
➢ 选择下载器和下载方式；

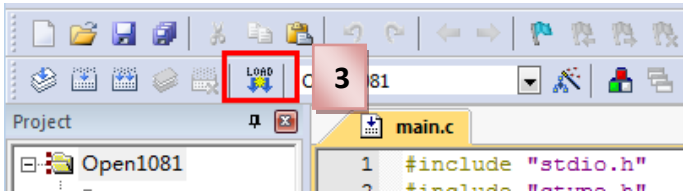
调出 Options for Target “Open1081”的设置对话框，点击 Debug，选择下载器为 ULNK2/ME Cortex Debugger，点击 Setting，在 SWJ 前面打勾，Port 中选择 SW，如下图：

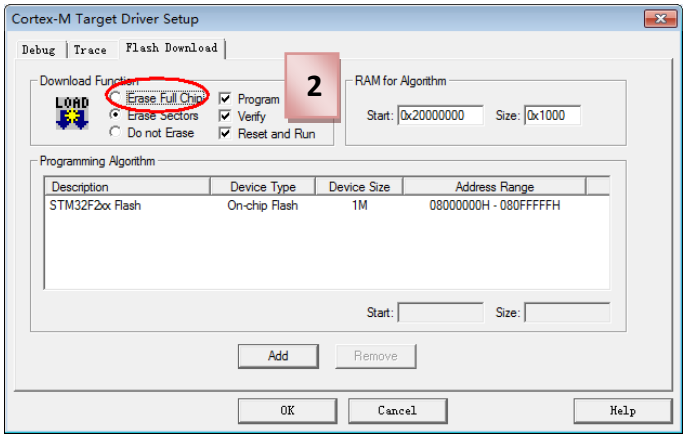


➢ 程序下载

不可以点击 Erase（1）和选择 Erase Full Chip（2），这样会破坏芯片 FLASH 分区，直接点击 LOAD（3）就可以下载了。







5.4. 使用 Keil MDK 生产 BIN 文件

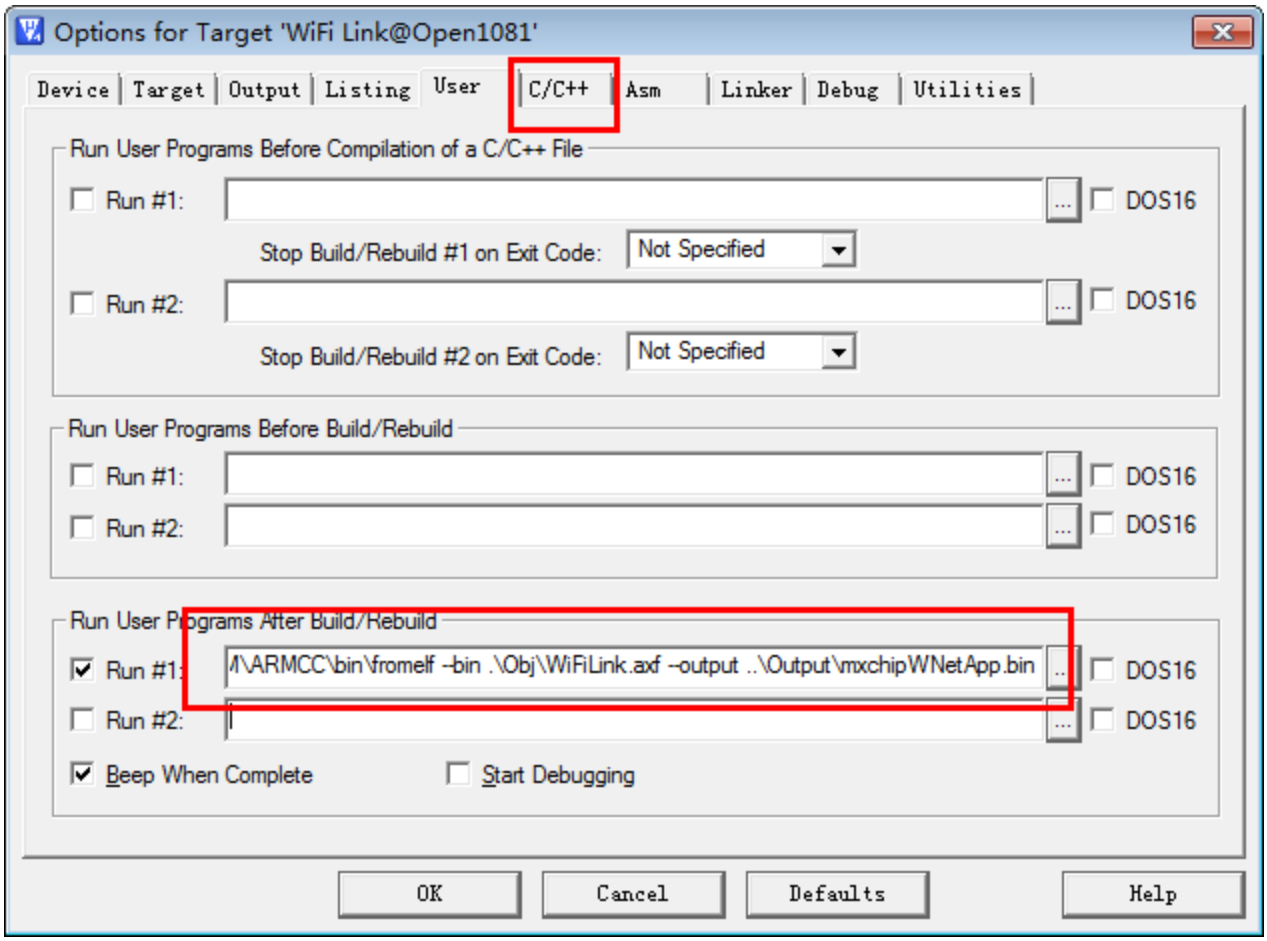
调出 Options for Target “xx.Open1081”的设置对话框,选择 User 面板，在”Run User Programs After Build/Rebulld” 勾上 RUN#1，在输入框中输入如下命令：

```
C:\Keil4.7\ARM\ARMCC\bin\fromelf --bin .\Obj\WiFiLink.axf --output ..\Output\mxchipWNetApp.bin
```

Bin 文件生成工具的路径

工程输出的.axf 文件

Bin 文件输出路径和文件名



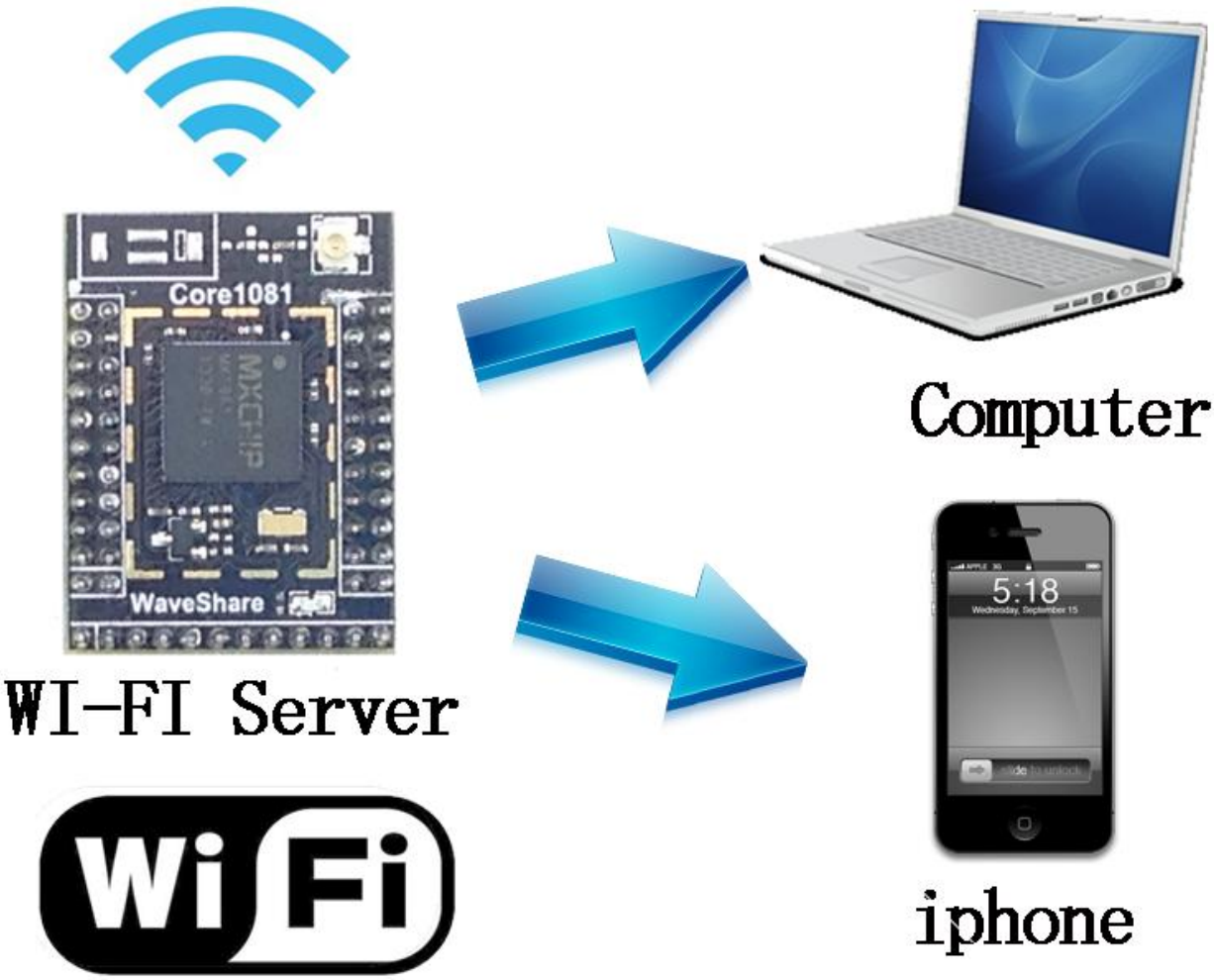
NOTE:

KIEL MDL4.7 以上的版本才有这个功能。

六、 WiFi 的基本介绍

6.1. WiFi 的几种工作模型：

- Core1081 模块做为 Server 时，可以把他设置为 AP 模式，可以做为无线网络的中心，PC 和智能手机等控制住终端度可以连接到这个网络中去了，从而他们就可以自由通信了。如下图：



- Core1081 模块作为 Client 时，可以把模块设置为 Station 模式，可以作为无线网络的一个客户端连接到 WIFI 网络，这样就可以和同时连接到 WIFI 网络的 PC 和智能手机等设备自由通信了，如下图：

七、 例程分析

使用 WiFi 固件版本：mxchipWNet Demo@EMW316x_V1.19
KEIL MDK 版本：4.54
下载器：ULINK/V2
下载方式：SWD

7.1. Open1081_Peripherals_Examples 例程讲解

以下例程如果使用到串口调试及串口助手 SSCOM32 如下设置，在单独的例程讲解中不再说明：

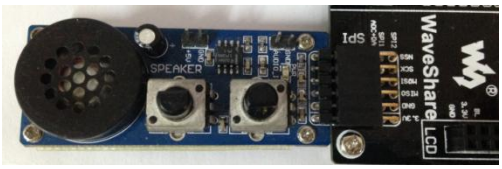
- 将 USB（mini）线接到 USB TO UART 接口。
- 将 UART2 JM 的跳线帽接上
- 打开串口助手 SSCOM32，选择相应的 COM 口，波特率设为 115200，点击【打开串口】。

7.1.1. ADC

- 程序说明

  本程序实现了一路 AD 采集实验。

- 硬件连接



  - 将 Analog Test Board 模块接入 SPI1 口。
  - 将 LED 跳线帽接上。

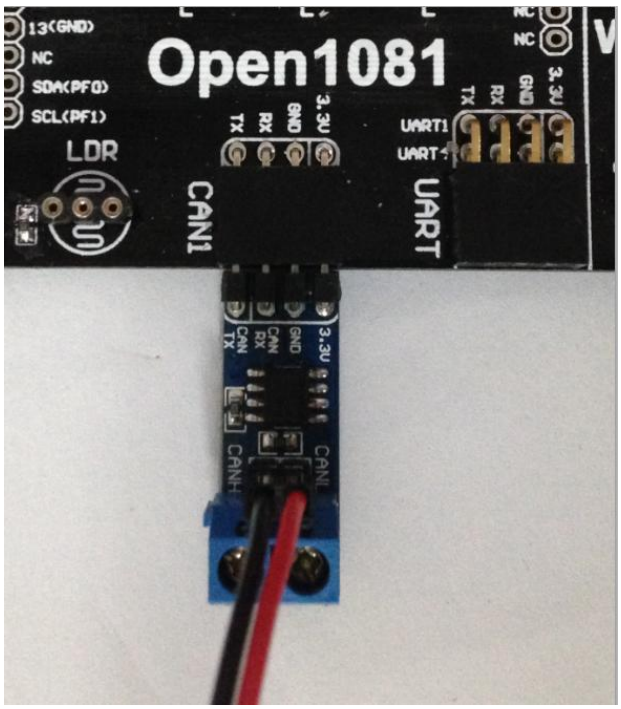
- 实验现象
  - 用手转动右边那个电位器，可以调节 LED 的闪烁速率。

7.1.2. CAN

- 程序说明

  本程序实现了两个 CAN 模块之间的互相通信。

- 硬件连接



  - 将 2 个 SN65HVD230 CAN Board 模块接入 2 块 Open1081 的 CAN1 接口上。
  - 用杜邦线把 2 个模块的 CANL –CANL（红色杜邦线）, CANH—CANH（黑色杜邦线）连接起来。
  - 将 WAKEUP 按键跳线帽接上。

- 实验现象
  - 用两块板子分别下载 CAN 程序，连接好硬件，
  - 打开串口助手 SSCOM32，按 WAKEUP 按键，SSCOM32 上就会显示相应的信息。

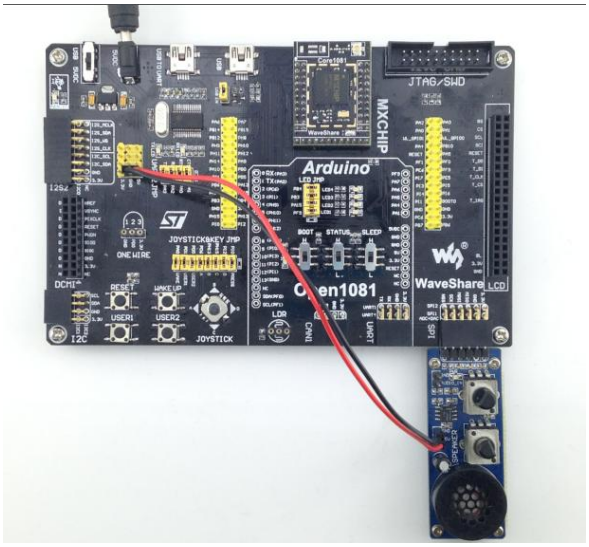
- 应用领域

  SN65HVD230 是德州仪器公司生产的 3.3CAN 总线收发器，主要是和带有 CAN 控制器的微处理器配套使用，该收发器具有差分收发能力，最高速率可达 1Mb/s。广泛用于汽车、工业自动化、UPS 控制等领域。

7.1.3. DAC

- 程序说明

  本例程是 DAC 管脚输出一个波形，是喇叭发出声音。
- 硬件连接



  - 将 Analog Test Board 模块接入 SPI1 口。
  - 用杜邦线给模块供 5V 电压。

- 实验现象

➢ 模块的喇叭发出声音。

- 应用领域

7.1.4. DCMI_OV2640

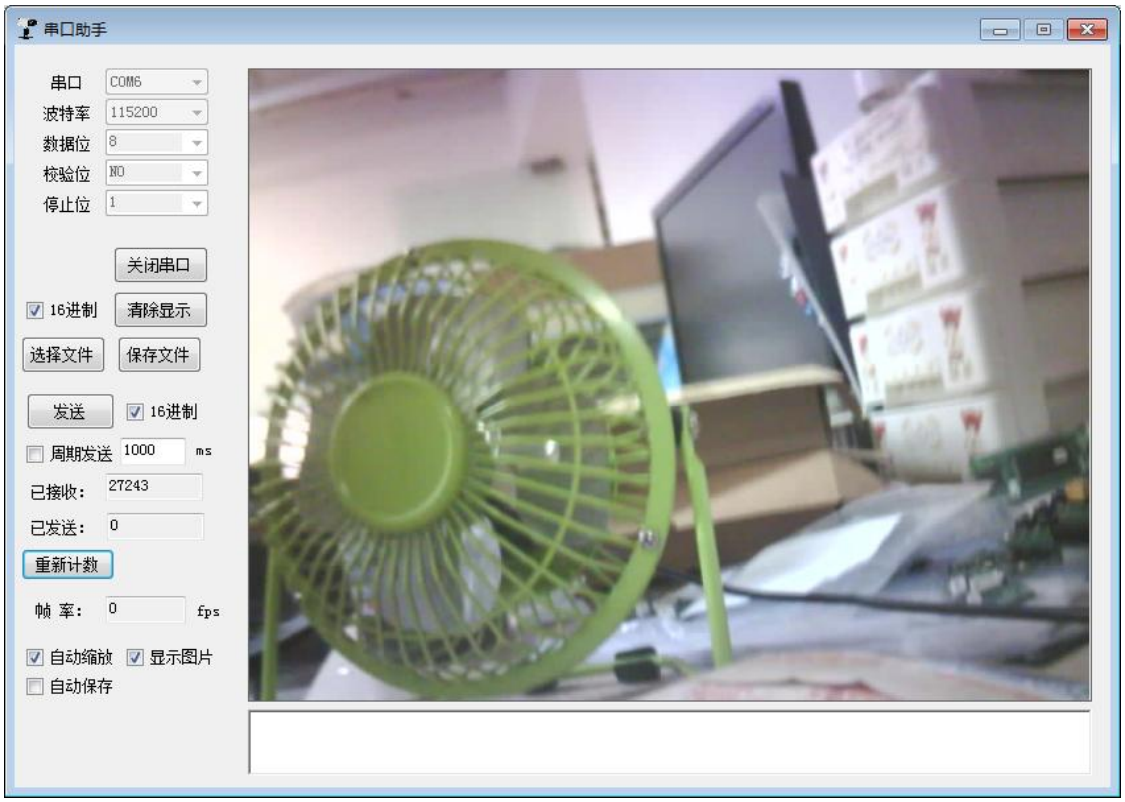
- 程序说明

  这个程序通过 OV2640 采集过来的 JPEG 格式图片，通过串口发到上位机上，在电脑端的上位机软件看到摄像头拍摄的图片。
- 硬件连接

  - 将 OV2640 模块接到 DCMI 接口上。
  - 将 USB（mini）线接到 USB TO UART 接口.
  - 将 UART2 JM 的跳线帽接上。

- 实验现象

  打开 CameraTest 软件，选择相应的 COM，波特率设为 115200，点击【打开串口】。在 CameraTest 中会显示 OV2640 采集回来的图像。



- 应用领域

7.1.5. DS18B20

- 程序说明
  通过温度传感器 DS18B20 采集温度，发送到串口助手上。
- 硬件连接
  1. 将 DS18B20 接入 ONEWIRE 接口上。
- 实验现象

```
DS18B20 TEST

****************************************************************
0x28 0x0 0x4d 0x0 0xb8 0xe 0xff 0x8
Temperture:0.8℃
 Temperture:22.18℃
 Temperture:22.25℃
 Temperture:22.18℃
 Temperture:22.25℃
 Temperture:22.18℃
```

7.1.6. GPIO_KEY_LED

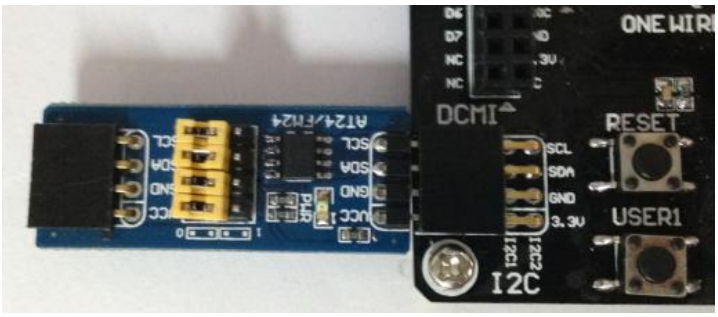
- 程序说明
  通过按键，摇杆改变 LED 的状态。（如果是用的串口的流控制的话，WAKEUP 按键不可用）
- 硬件连接
  1， 将 LED JMP，JOYSTICK&KEY JMP 的跳线帽接上。
- 实验现象
  用手按摇杆和按键，LED 的状态会改变。

7.1.7. I2C

- 程序说明

  通过 I2C 协议读写 E2PROM 上的数据。

- 硬件连接



- 将 AT24/FM24 Board 模块接到 I2C 接口上。
- 如果软件中使能 I2C1，模块就接到下面那排。
- 如果软件中使能 I2C2，模块就接到上面那排。

- 软件设置

  程序中设置：

1， 打开 MX1081_IO.H 文件，在里面找到选择 I2C 的宏定义：

当 AT24/FM24 Board 接到 I2C1 时，把 #define USE_I2C1 I2C1 使能。

当 AT24/FM24 Board 接到 I2C2 时，把 #define USE_I2C2 I2C2 使能。

- 实验现象

  串口助手会打印如下信息：

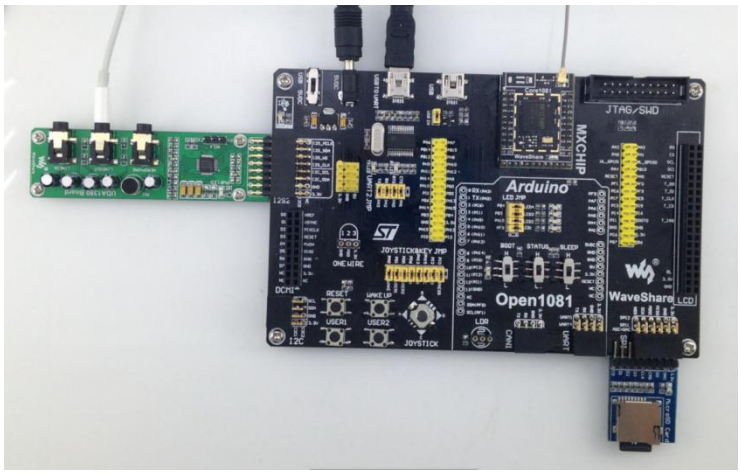
```
************************************************************
 EEPROM 24C02 Write Test
 EEPROM 24C02 Write Test OK
 EEPROM 24C02 Read Test
 EEPROM 24C02 Read Test OK
```

7.1.8. I2S_UDA1380

- 程序说明

  本例程使用的是飞利浦的 I2S 协议驱动 UDA1380 Baord 播放音乐。

- 硬件连接



- 将 UDA1380 Baord 模块接到 I2S 接口上。
- 将 Micro SD Storage Board 模块接到 SPI1 接口上，并在 SD 卡中放入名字为 Audio.wav 的音频文件
- 把耳机接到 UDA1380 Baord 上的 LINEOUT 接口上。

◆ 实验现象

- 点击 RESET 按键，此时可以听到有音乐输出。

2. 串口助手会打印 WAV 音频解析的一些信息。

7.1.9. LCD22

◆ 程序说明

我们这款 LCD22 是电阻式 2.2inch 带触摸的 LCD，分辨率为 320x240，采用 SPI 方式驱动，大大减少了控制管教，使得 IO 口比较紧缺的单片机也可以驱动，本例程就是演示了 LCD 显示点，画线，画圆，显示字符等一些功能。

◆ 硬件连接

- 将 2.2inch 320x240 Touch LCD (A)模块接到 LCD22 接口上。

◆ 实验现象

1， LCD 上显示信息

7.1.10. LCD22_Touchpanel

◆ 程序说明

1， 先触摸屏校准，你点击 3 下就可以完成触摸屏的校准，之后就会进入触摸屏画板界面。

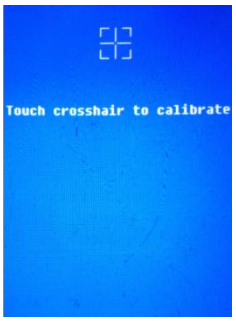
2， 在触摸屏画板中，你可以在触摸屏的下面选择画笔颜色，点击 C 可以清除画板。

◆ 硬件连接

将 2.2inch 320x240 Touch LCD (A)模块接到 LCD22 接口上。

◆ 实验现象

2， LCD 上显示信息



触摸屏校准界面

触摸屏面板

◆ 应用领域

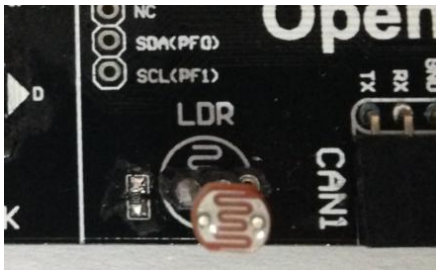
手持设备的显示。

7.1.11. LDR

◆ 程序说明

本例程用过 AD 采集光敏电阻的信号，经过 CPU 处理，反应到 LED 上显示。

◆ 硬件连接



- 将光敏电阻接到 LDR 接口上。
- 短接 LED 上的跳线帽

◆ 实验现象

用手慢慢的遮住光敏电阻，使光线慢慢的边暗，这时你会发现 LED 的闪烁情况会随着光线的强弱变化。

◆ 应用领域

7.1.12. RTC

◆ 程序说明

MX1081 自带 RTC 时钟模块，本例程演示了 RTC 实时时钟工作功能

◆ 实验现象

会看到串口助手上打印如下信息：

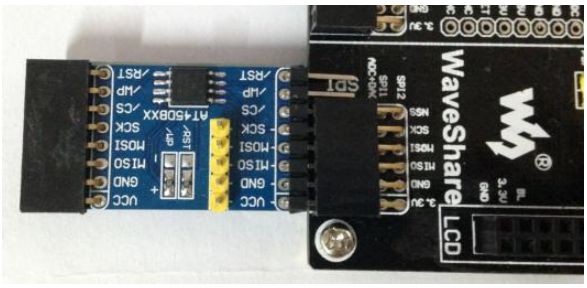
```
RTC TEST
The time :  00:00:01
The time :  00:00:02
The time :  00:00:03
The time :  00:00:04
The time :  00:00:05
The time :  00:00:06
```

7.1.13. SPI_AT45DB

◆ 程序说明

本程序演示了 MX1081 通过 SPI 接口驱动 AT45DBXX DataFlash Board。

◆ 硬件连接



- 将 AT45DBXX DataFlash Board 模块接到 SPI 口上。
- 如果软件中使能 SPI1，模块就接到下面那排。
- 如果软件中使能 SPI2，模块就接到上面那排。

◆ 软件设置

串口助手设置：

打开串口助手 SSCOM32，选择相应的 COM 口，波特率设为 115200，点击【打开串口】。

程序中设置：

1， 打开 MX1081_IO.H 文件，在里面找到选择 I2C 的宏定义：

当 AT45DBXX DataFlash Board 接到 SPI1 时，把 #define USE_SPI1 SPI1 使能。

当 AT45DBXX DataFlash Board 接到 SPI2 时，把 #define USE_SPI1 SPI2 使能。

◆ 实验现象

会看到串口助手上打印如下信息：

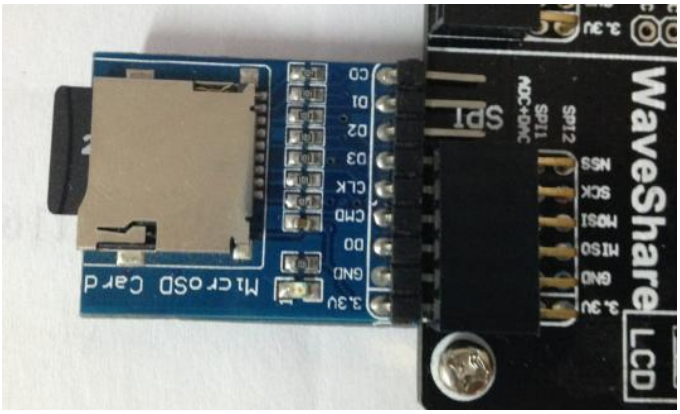
```
SPI is ready!
AT45DBXX had been Init!
AT45DBXX ID is 0x1f  0x24  0x0  0x0

FALSH AT45DBXX Write Test:
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39
FALSH AT45DBXX Read Test:
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39
FALSH AT45DBXX Read Test Succeed!
```

7.1.14. SPI_SD_FatFS

◆ 程序说明

本程序演示了 MX1081 通过 SPI 接口驱动 SD 卡，演示了 FATFS 文件系统的读写，创建文件等操作。

◆ 硬件连接

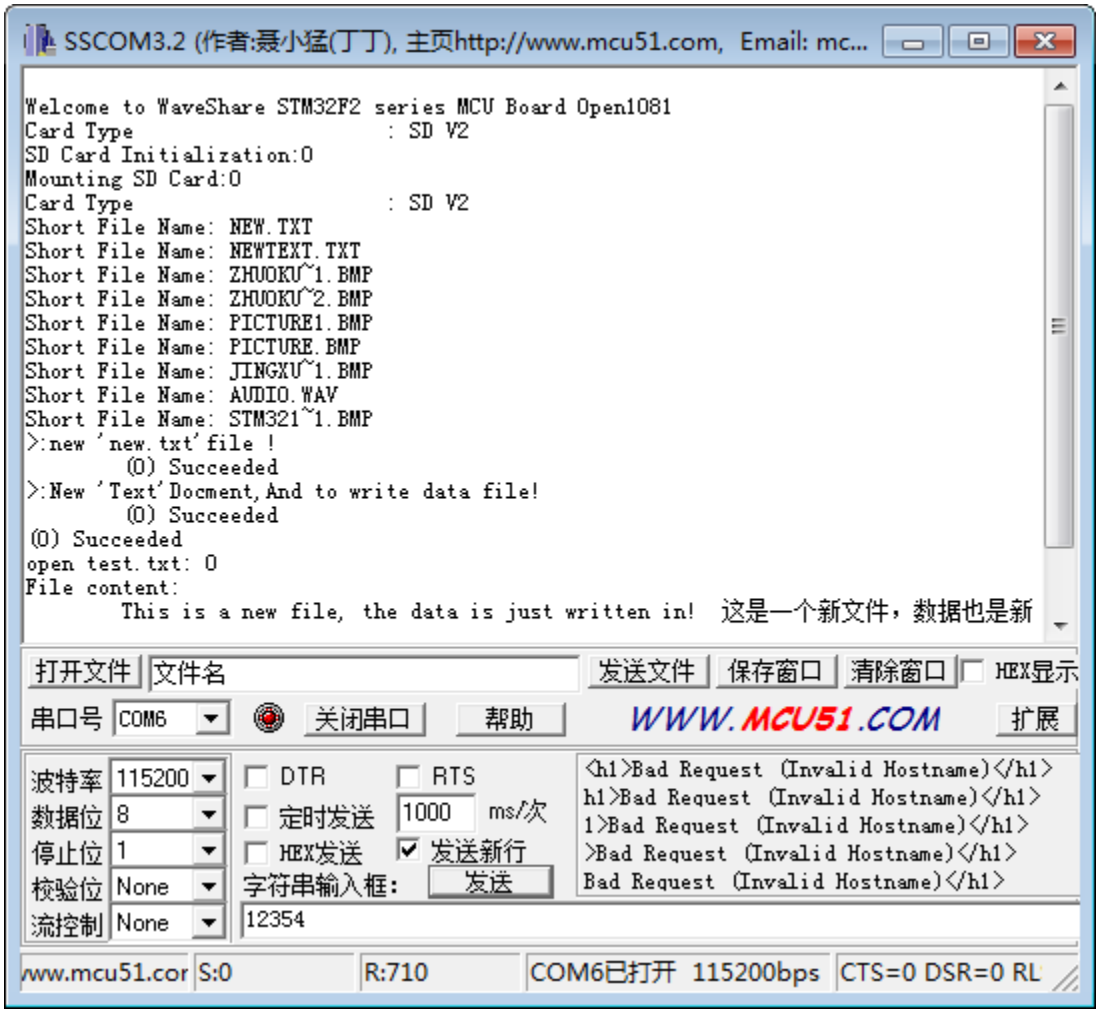


- 将 Micro SD Storage Board 模块接到 SPI1 接口上。
- 将 SD 卡接到 Micro SD Storage Board 插槽。

◆ 软件设置

◆ 实验现象

会看到串口助手上打印如下信息：



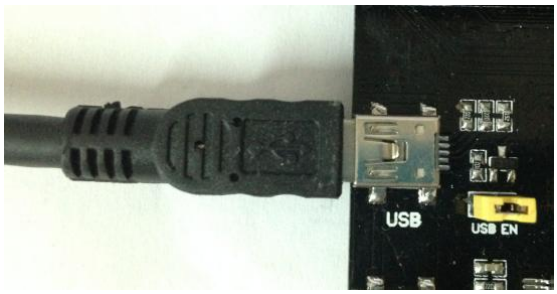
◆ 应用领域

7.1.15. USB_Device_Examples

- 程序说明

  本程序演示了 MX1081 的 USB 从机实验，实现了 USB 鼠标功能，直接使用 Open1081 的上摇杆就可以控制电脑的鼠标上下左右的移动。

- 硬件连接



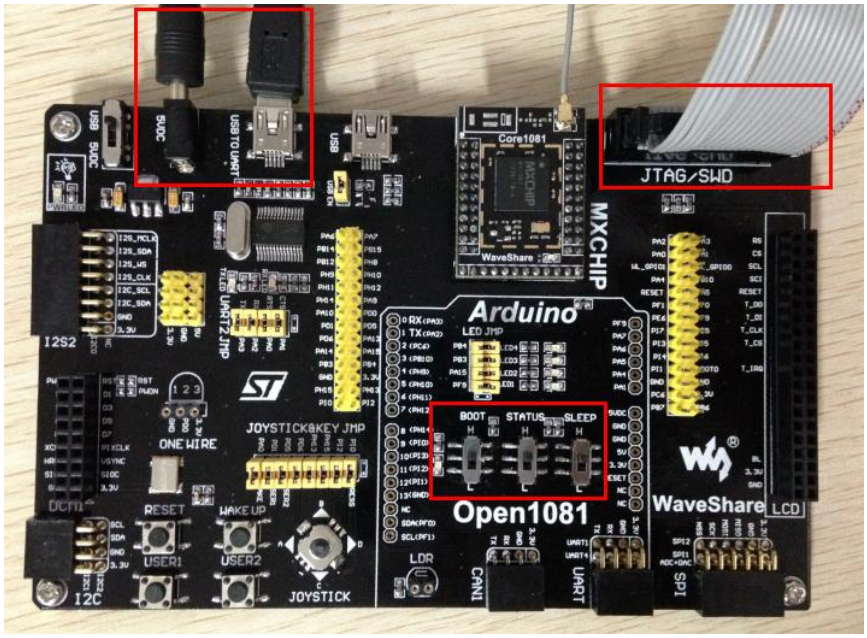
  - 将 USB (MINI)先连接到 Open1081 的 USB 接口上。
  - 将 JOYSTICK &KEY JMP 的跳线接上。

- 软件设置
- 实验现象

  可以通过 Open1081 上的摇杆控制电脑上的鼠标。

7.2. Open1081_WiFi_Examples 例程讲解

➢ 在测试以下例程时，开发板的硬件连接如下图：



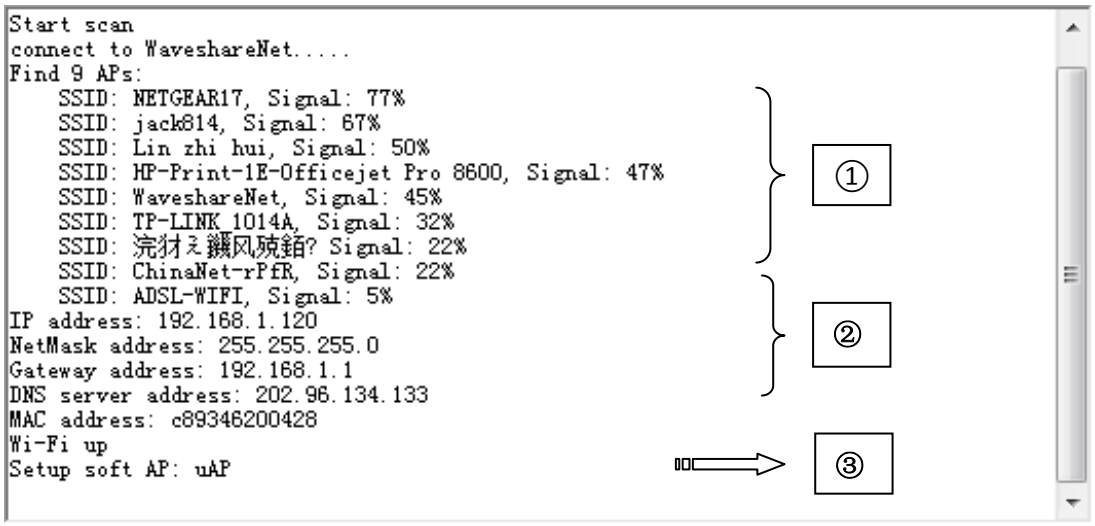
➢ 将 BOOT,STATUS,SLEEP 开关设置为 H。
➢ 在 5VDC 接口给板子供 5V 电源。
➢ 接上 USB（mini）线

➢ 如果下面程序中说的连接到指定 WiFi 网络，所以大家在做这些实验的时候要改成你当面的 WiFi 网络，此时你只要修改以下宏定义就好。

#define AP_NAME "WaveshareNet"（指定网络名称）
#defineAP_PASSWORD"waveshare0755"（指定网络密码）

7.2.1. Demo1_WiFi_Link

◆ 程序说明
使射频开始扫瞄所有频段上的无线网络。并连接到指定的 WiFi 网络，并建立一个名字为"uAP"的 WiFi 热点。
◆ 软件流程
➢ 初始化软件库和串口。
➢ 使射频开始扫瞄所有频段上的无线网络
➢ mxchipWNet 调用回调函数 ApListCallback 通过串口打印出扫瞄到的 WiFi 无线网络
➢ 连接到指定的 WiFi 网络。
➢ mxchipWNet 调用回调函数 NetCallback 打印当前连接的无线网的信息。
➢ 创建一个名为"uAP"的无线网络。
◆ 实验现象
➢ 打开串口助手软件，可以看到如下打印信息：



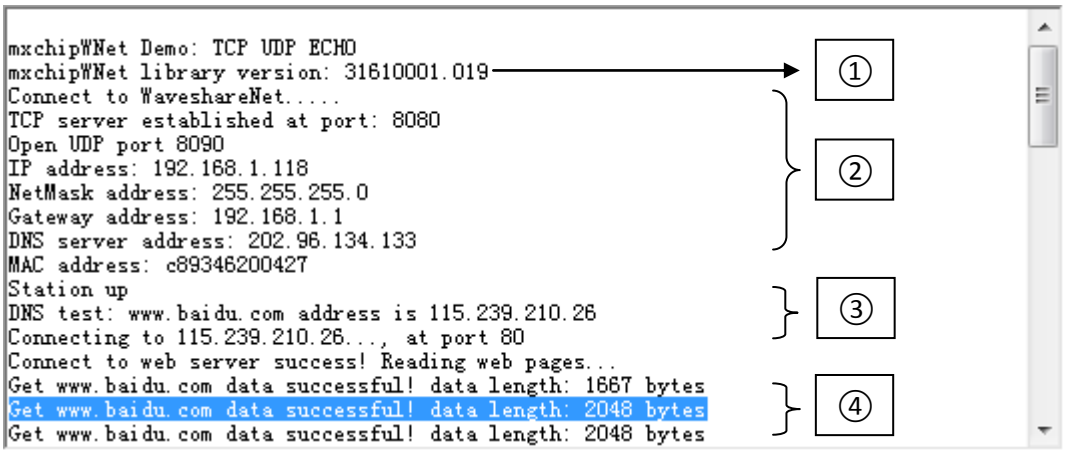
① 扫瞄到的 WiFi 无线网络。
② 当前网络连接的信息
③ 当前创建的 WiFi 网络

7.2.2. Demo2_TCP_UDP_ECHO

- 程序说明
  本例程分为 3 个部分，分别实现了 TCP，UDP，ECHO 的通信。

- 软件流程
  - TCP 通信
    - 初始化软件库和串口。
    - 连接到指定的 WiFi 网络。
    - 设置 TCP 通讯时，keepalive 收发的参数。
    - 建立一个 TCP socket 连接，并且绑定一个端口（8080）。
    - 查询 TCP socket 的状态。
    - 读取 TCP 客户端的数据和发送读到的数据。
  - UDP 通信
    - 初始化软件库和串口。
    - 连接到指定的 WiFi 网络。
    - 建立一个 UDP socket 连接，并且绑定一个端口（8090）。
    - 查询 UDP socket 的状态。
    - 读取 UDP 客户端的数据和发送读到的数据。
  - ECHO 通信
    - 初始化软件库和串口。
    - 连接到指定的 WiFi 网络。
    - 用 connect 函数建立一个建立与远程设备的连接，并且绑定一个端口（80）。
    - 检查 TCP 连接请求。
    - 读取来着 www.baidu.com 的数据。

- 实验现象
  - 打开串口助手软件，可以看到如下打印信息：



① 软件的版本号。

② 当前网络连接的信息。

③ 获取得到的百度 IP 和分配到的端口号。

④ 获取百度的数据。

  - 用手机连接到模块理解的相同网络，打开 TUP，UDP 测试工具，在软件中写入上面串口助手的打印的 IP address，TCP 中的端口号为：8080 ，UDP 的端口号为：8090。点击连接，在发送框中输入一些数据，点击发送，在接口框可以看到相同的数据。操作如图：

UDP 通信

TCP 通信

7.2.3. Demo3_WPS_EasyLink

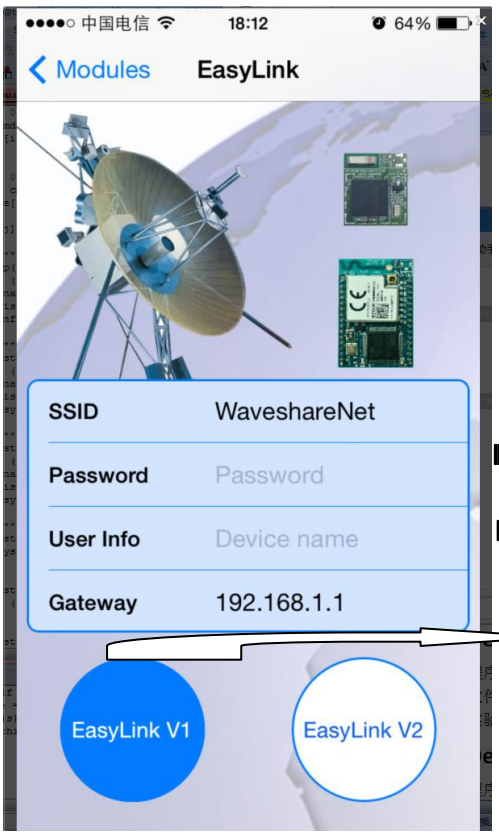
- 程序说明

这个例程可以实现模块快速的连接到 WiFi 网络中，

- 软件流程
  - 初始化软件库和串口。
  - 连接到指定的 WiFi 网络。
  - 等待用户输入指令并进入相应的操作。
- 实验现象
  - 打开串口助手软件，可以看到如下打印信息：

```
mxchipWNet Demo: WPS and Easylink demo
 mxchipWNet library version: 31610001.019

+****************(C) COPYRIGHT 2013 MXCHIP corporation************+
|          EMW316x WPS and EasyLink configuration demo            |
+ command ----------------+ function ----------------------------+
| 1:WPS                   | WiFi Protected Setup                  |
| 2:EasyLink              | One step configuration from MXCHIP    |
| 3:EasyLink_V2           | One step configuration from MXCHIP    |
| 4:REBOOT                | Reboot                                |
| ?:HELP                  | displays this help                    |
+-------------------------+---------------------------------------+
|                          By William Xu from MXCHIP M2M Team     |
+-----------------------------------------------------------------+

MXCHIP>
```

  - 输入” WPS”或者” 1”，按回车模块进入 WPS 配置。
    此时你按一下 AP 上的 WPS 按键，模块就能快速的连接到 WiFi 网络中了。
  - 输入” EASYLINK”或者” 2”，按回车模块进入 Easylink V1 配置。
  - 输入” EasyLink _V2”或者” 3”，按回模块车进入 Easylink V2 配置。
  - 用手机连接到 WiFi 网络，打开 EasyLink 软件，配置好 WiFi 密码和模块名字，分别点击 EasyLink V1, EasyLink V2 就可以显示快速的连接到 WiFi 网络，如下图：



在这里输入当前手机连接 WIFI 的密码

为你的设备取个名字

点击这个两个按键，分别以两种不同的方式是模块接连到网络中，此时串口打印如下信息：

```
MXCHIP> 2

EasyLink started......, start your easylink function on iOS/Android AP
Configuration is successful, SSID:WaveshareNet, Key:waveshare0755
connect to WaveshareNet.....

MXCHIP> IP address: 192.168.1.118
NetMask address: 255.255.255.0
Gateway address: 192.168.1.1
DNS server address: 202.96.134.133
MAC address: c89346200427
```

7.2.4. Demo4_Webserver_OTA

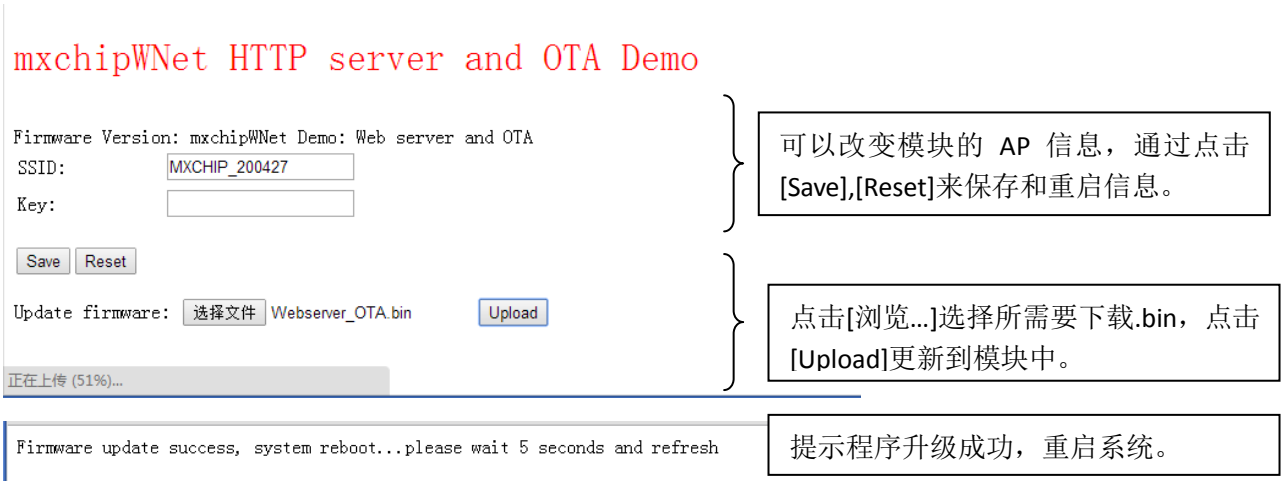
- 程序说明

  本例程演示了模块通过 WiFi 无线下载程序到芯片中。

- 软件流程
  - 初始化软件库和串口。
  - 连接到指定的 WiFi 网络。
  - 初始化 network_InitTypeDef_st 结构体，并创建一个名为“MXCHIP_200427”的 WiFi 热点。
  - 设置 TCP 通讯时，keepalive 收发的参数。
  - 初始化 http 登入的用户名和密码。
  - 建立一个 TCP socket 连接，并且绑定一个端口（80），并调用 listen 来监听这个 socket。
  - 相应 Http 的请求。
- 实验现象
  - 打开串口助手软件，可以看到如下打印信息：

```
mxchipWNet Demo: Web server and OTA
mxchipWNet library version: 31610001.019

mxchipWNet Demo: Web server and OTA
mxchipWNet library version: 31610001.019
Establish soft AP: MXCHIP_200427.....
```

  - 是用 HTTP 下载 bin 文件到模块中。



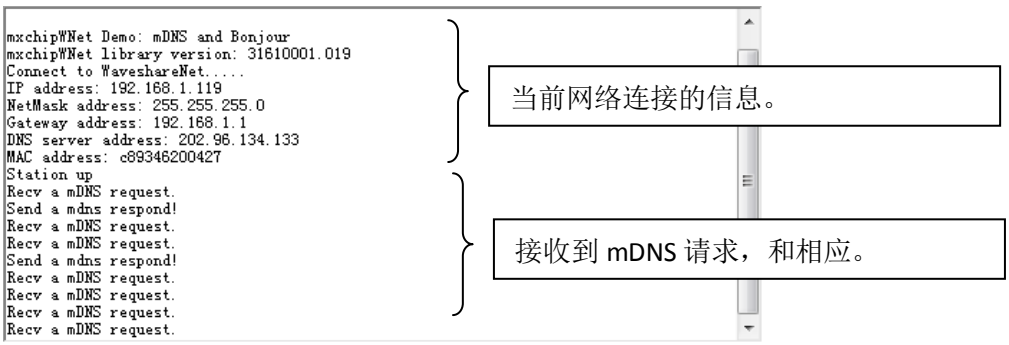

```
Firmware update success, system reboot...please wait 5 seconds and refresh
```

提示程序升级成功，重启系统。

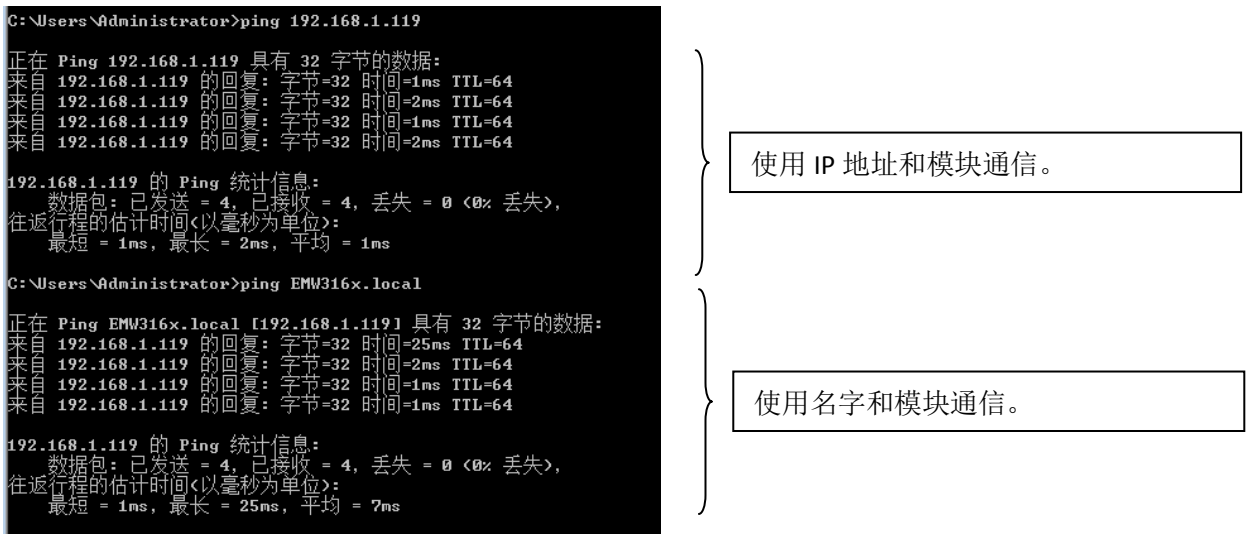
7.2.5. Demo5_mDNS_bonjour

- 程序说明
  - 局域网中，我们可能习惯用 IP 地址指定主机。但是，如果使用 DHCP 分配 IP 地址，我们就很难事先知道某台主机的 IP 地址（当然通过配置 DHCP 服务器可以给某个 MAC 地址的机器总是分配同一个 IP 地址，但不是每个局域网我们都能控制），这对局域网内自动化影响很大。使用名字而不是 IP 地址去指定主机可以解决这个问题。
  - 使用名字涉及到解析名字，毕竟主机最终还是通过 IP 协议进行通信。Windows 时代的做法是 NetBIOS，而现代苹果的做法是 mDNS。mDNS 的原理不复杂，想解析名字的主机在局域网里发一个组播的 DNS 查询，刚好有这个名字的主机收到后响应。整个过程与 ARP 类似，但是 mDNS 是在 IP 层面，而且 mDNS 协议设计得很好，可以很自然地看成是 DNS 协议的补充（报文格式和 DNS 报文几乎一样）

- 软件流程
  - 初始化软件库和串口。
  - 连接到指定的 WiFi 网络。
  - 初始化 mDNS，设定主机名字，服务器名字等参数。
  - 建立一个 UDP socket 连接，并且绑定一个端口（5353）。
  - 查询 UDP socket 的状态。
  - 相应 UDP 请求。
- 实验现象
  - 打开串口助手软件，可以看到如下打印信息：



  - 打开电脑的 CMD 指令串口，使用 2 中方式去和模块通信：



7.2.6. Demo6_SSL_https

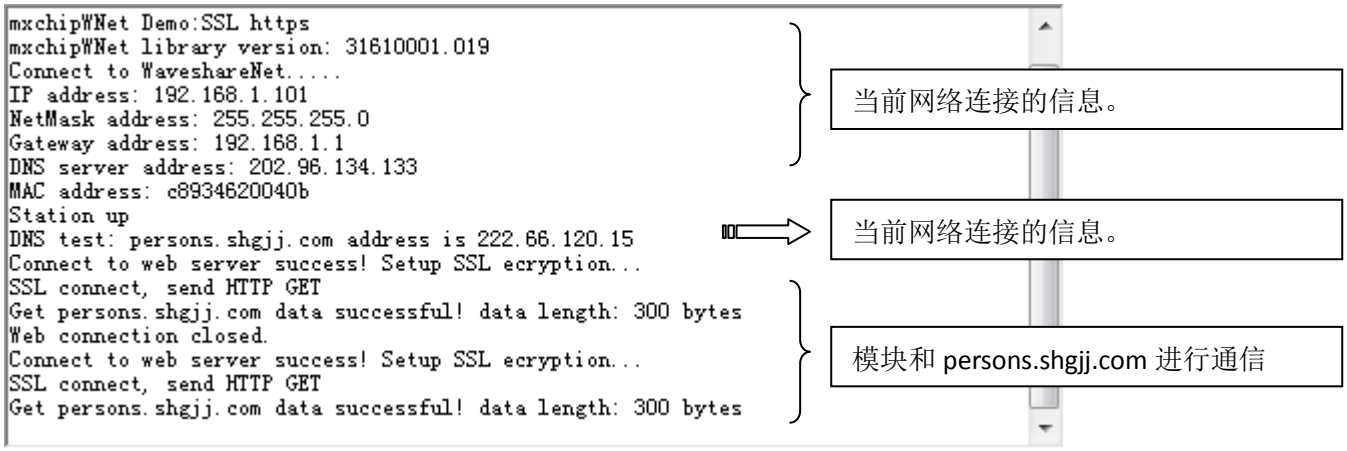
- 程序说明

  SSL 的英文全称是“Secure Sockets Layer”，中文名为“安全套接层协议层”，它是网景（Netscape）公司提出的基于 WEB 应用的安全协议。SSL 协议指定了一种在应用程序协议（如 HTTP、Telenet、NMTP 和 FTP 等）和 TCP/IP 协议之间提供数据安全性分层的机制，它为 TCP/IP 连接提供数据加密、服务器认证、消息完整性以及可选的客户机认证。
- 软件流程
  - 初始化软件库和串口。
  - 连接到指定的 WiFi 网络。
  - 设置 TCP 通讯时，keepalive 收发的参数。
  - 解析"persons.shgjj.com"，获取 IP 地址。

- 建立一个 TCP socket 连接，并且绑定一个端口（443），并调用 listen 来监听这个 socket。
- 查询 TCP socket 的状态。
- 读取来自"persons.shgjj.com"的数据。

◆ 实验现象

- 打开串口助手软件，可以看到如下打印信息：



版本修订

版本号	修改地方	发行时间	作者
1.0	初稿	2014/04/21	Waveshare team