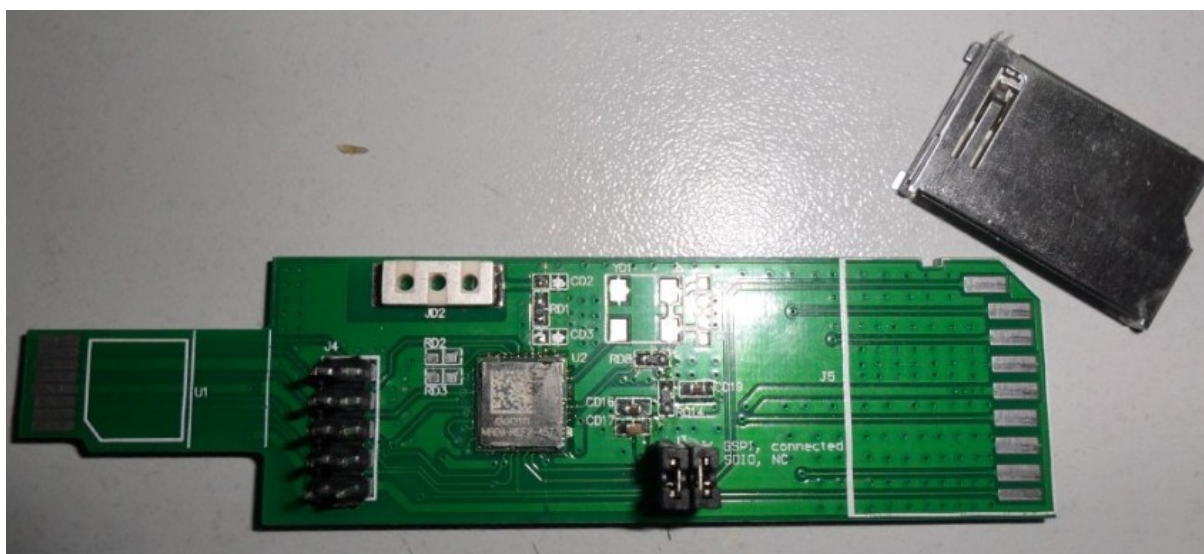


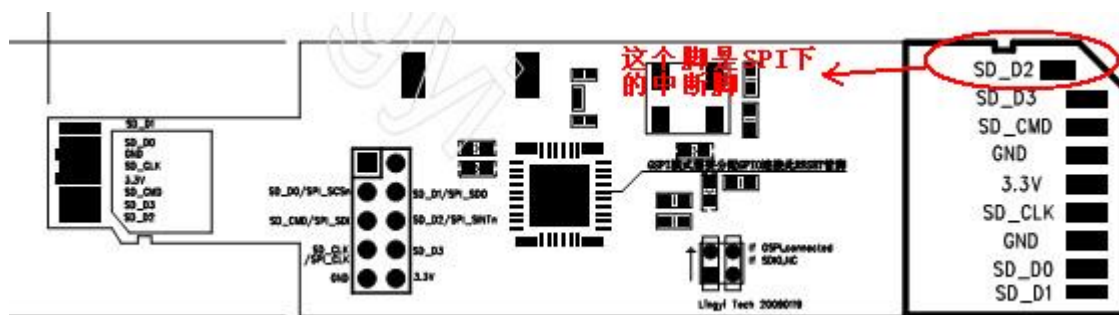
一、USI WM-G-MR-09 WIFI开发板(88W8686 核心)介绍

该评估板以常用的 WIFI SOC 88W8686 为核心，支持 SD 卡接口和 TF 卡接口，如果你的板子上有 SD 卡座，可以直接插在上面调试(注意一般 SPI 接口的 SD 卡槽没有使用 SD_D2 这个脚，而 WIFI 板子上这个脚是中断引脚，请确保 SD 卡槽这个引脚是悬空的，并且使用查询模式)，另外还将用到的接口通过 2.54mm 间距排针引出，进一步方便扩展连接。下面为实物图，这是 [WIFI 开发板店铺地址](http://item.taobao.com/item.htm?id=9703059117)：

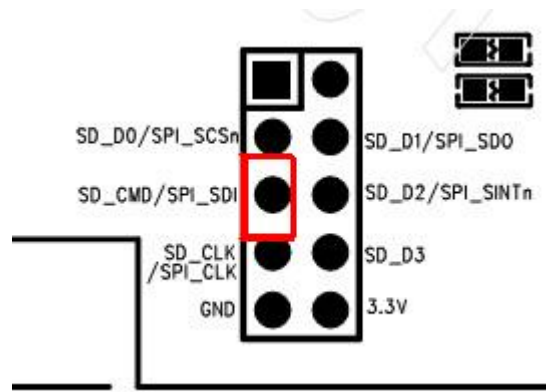
<http://item.taobao.com/item.htm?id=9703059117>



以下是板子的接口图：



下图是 2.54mm 排针的详细定义：



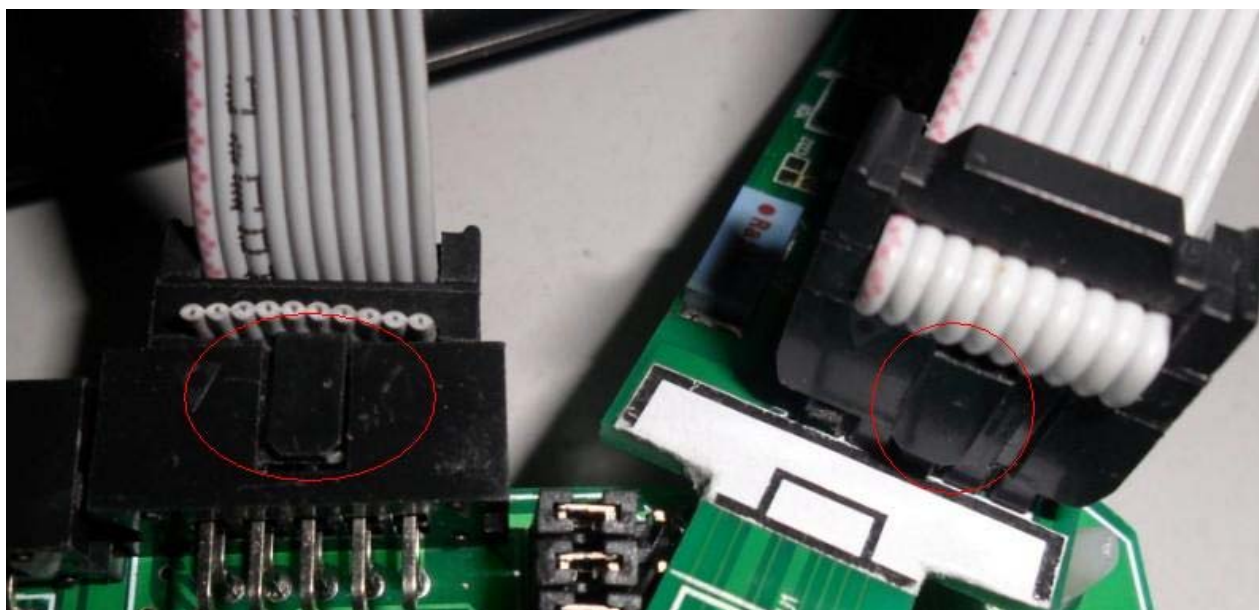
注意红色方框指示的是该接口与 STM32 主板的 10*2 插座连接时排线的凸起朝向，如果没有使用 STM32 主板可以按引脚定义通过跳线连接，注意供电为 3.3V，电源不能插反。因为手上可能带有静电，尽量不要用手接触到 WIFI 芯片。

如果你的笔记本电脑支持 SDIO 接口(注意有些笔记本的 SD 卡插槽并不是标准的 SDIO 接口)的话,可以安装 windows 下的驱动(XP 和 win7)并插上评估板(取下两个短路子, 才能作为 SDIO 接口)做为无线网卡使用。支持通常的 IEEE 802.11b/g 最大带宽 54Mbps。如果在笔记本下安装了驱动后可以在设备管理器的网络适配器下发现一张新的网卡,并在网络邻居下可以通过该本地连接的网卡连接无线路由器。

二、WIFI开发板与STM32 主板的连接

2.1 连线

上面已经介绍了 STM32 主板与 WIFI 开发板接口的连接,通过一根 10P 的跳线可以直接将 STM32 主板下的 P6 接口与 WIFI 板子连接,注意排线的接口朝向,不要插反,插好以后仔细检查主板和 WIFI 开发板的 GND 与 3.3V 是否对应,如果不对应,请检查排线接口是否插好,检查无误后便可以上电调试了。下图是正确的排线凸起连接图。



2.2 跳线选择

使用 WIFI 的 SPI 模式时要将 WIFI 开发板上的 SDIO 与 SPI 选择端子短接，另外 STM32 主板的 P3 排针处(圆柱体晶振右上方)将左边两排排针分别短接，STM32 使用 SWD 仿真，而禁用 JTAG 仿真功能，因为 WIFI 的片选和 JTAG 的 JTDO 复用了。(STM32 有两种仿真模式，SWD 和 JTAG，JTAG 使用的引脚较多，SWD 只需要三个引脚就能仿真，速度上 SWD 比 JTAG 要快，Jlink V8 支持两种模式的仿真，要使用 JTAG 复用的 GPIO 还需要在程序上电时关掉 JTAG 功能，具体看 STM32+WIFI 的固件下载演示代码)。

三、演示代码介绍

3.1 工程代码介绍

使用 MDK4. x 以上打开，使用了 3.3 的固件库(目前较新较稳定的一个片级固件支持库)，存放于 Libraries 下；工程文件存存放在 MDK_Project 文件夹下，名称为 MyProject.uvproj

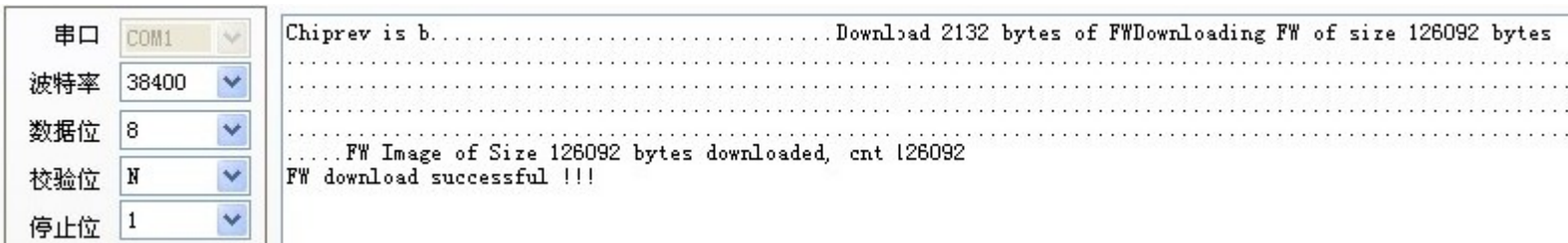
跟板级有关的代码都放在 MyBsp 文件夹下如 Wif, LED_Key 等, 跟芯片级有关的代码都放在 OnChip_Drivers 中，如 SPI 等。User 文件夹下放的是上层应用函数如 main 函数等。Doc 文件夹下存放了工程每次更新的过程、内容记录已经新建一个基于 3.3 固件库的工程步骤。注意查看相关文件夹下的 readme.txt 了解相关内容。

根据不同的芯片容量和型号选择相应的目标芯片。工程的默认芯片为 RC，下载器为 Jlink V8 ，使用 SWD 仿真模式。

3.2 演示代码

目前版本的驱动代码能够读取 WIFI 的 ID 和进行固件下载，相关的代码在 if_gspi.c 里面。SPI 使用的是 SPI1，16 位模式，具体参看 SPI 的初始化代码，片选为 PB3，暂时还没有使用中断，如果使用 STM32 配套主板，应该是 PA15，这

两个引脚都和 JTAG 复用，所以程序要关闭 JTAG 仿真功能，使用 SWD 仿真。固件包括 helperimage 和 fmimag 两部分，它们是两个数组，分别对应两个头文件，并被编译到了工程里面。这两个固件数组编译后有 120 多 KB，所以 STM32 选用了 RC256K flash 系列的芯片(如果应用到产品时可以外扩一个 EEPROM 或者 dataflash 来存放固件，以节约成本)，固件下载一次后在不掉电情况下就可以不用再下载。以下是固件下载截图，这是通过串口打印的输出调试信息(使用串口 1)，WIFI 的 ID 是 0x0b 两个固件大小分别是 2132 字节和 1226092 字节，最后显示的是读取 WIFI 固件下载标志寄存器，显示下载固件成功。固件如果下载成功后再次下载，WIFI 模块将不接受数据，直接显示固件已经下载成功。



四、后续发展展望

目前的代码已经能连接 WIFI 模块并且下载固件，下一步的工作就是 WLAN 驱动的移植开发，然后是嵌入式实时操作系统+LWIP+WIFI，由于开源实时操作系统 RT_Thread 已经移植了 LWIP 协议栈，所以准备先在该系统下实现 WIFI 连接。而且该开源操作系统的团队也正在开发 WIFI 组建，后续的技术资料应该会相对丰富。

应该认识到的是 WIFI 开发是主流热点也是个技术难点，目前成功的应用都是基于 ARM9 以上的，硬件成本较高，不便于大面积铺设节点，而这也让性价比较高的以 CM3 为内核的 32 位单片机实现 WIFI 连接看到了生存前景。

技术的道路需要一步一个脚印，我也相信走的人多了，路就会宽，不论是开拓者还是追随者，今天我们都已上路……