

手机端 APP 通过 Wifi 控制开发板

STM32 入门系列教程

版本历史

版本	发布时间	修改内容	作者
V1.0	2016-04-20	新建文件	硬石

前言

STM32 硬石开发板集成了 ESP8266 wifi 模块,该模块是由安信可科技开发的。主频支持 80MHz 和 160Mhz, 支持 RTOS, 集成 Wi-Fi MAC/BB/RF/PA/LNA , 板载天线。我们选择的 wifi 模块支持标准的 IEEE802.11 b/g/n 协议, 完整的 TCP/IP 协议栈。在开发板上集成, 我们可以利用该模块为我们的开发板添加互联网功能, 当然也可以自己建立热点。

目录

第 1 章	ESP8266 说明.....	4
1.1	主要参数.....	4
1.2	使用说明.....	4
1.3	实验现象.....	11

第1章 ESP8266 说明

1.1 主要参数

下面表 1-1 是模块的主要参数，可以看到 ESP8266 支持的网络标准广泛，几乎现代的 WIFI 都支持了，这就给我们进行开发带来了很大的便利。

表 1-1 ESP8266 主要参数

项目	说明
网络标准	无线标准：IEEE 802.11b、IEEE 802.11g、IEEE 802.11n、
无线传输速率	802.11b：最大速率：11Mbps 802.11g：最大速率：54Mbps 802.11n：最大速率：600Mbps
频率范围	2.4GHz-2.5GHz (2400M-2483.5M)
工作电压	3.0~3.6V（建议 3.3V）
工作温度	-40°C ~ 125°C
工作电流	平均值： 80mA
封装大小	16mm * 24mm * 3mm
加密类型	WEP/TKIP/AES
网络协议	IPv4, TCP/UDP/HTTP/FTP
工作模式	WIFI STA 、WIFI AP、WIFI STA+WIFI AP

模块支持 STA、AP、STA+AP 三种工作模式

- ❑ STA 模式：ESP8266 模块通过路由器连接互联网，手机或电脑通过互联网实现对设备的远程控制。
- ❑ AP 模式：模块作为热点，实现手机或电脑直接与模块通信，实现局域网无线控制。
- ❑ STA+AP 模式：两种模式的共存模式，可以通过互联网控制可实现无缝切换，方便操作。

1.2 使用说明

开发板集成了 ESP8266 模块，其模块引脚如图 1-1 所示，引脚功能定义如表 1-2。查阅硬石开发板的原理图可以发现，开发板和 ESP8266 的是 USART2 串口，实现数据的传输。IO 口 PD6 是连接模块的 RST 引脚，实现模块的复位。我们只

需要通过简单的串口配置，就可以通过网络（wifi）来传输自己的数据。根据开发板的情况，我们便设计了一个例程，开发板通过 ESP8266 模块和手机端 TCP 软件进行联网，实现的功能是可以利用手机端的 TCP 软件，发送指定的数据，从而达到控制开发板的目的，我们的例程是控制三个 LED 灯的亮灭和蜂鸣器的响与不响。如果有更多的想法，大家可以自行开发。下面是对开发板例程和手机端的具体设置。模块默认是 AT 指令状态，由于已经对各个 AT 指令集成到函数，所以我们直接操作各个函数就能达到控制模块的目的。

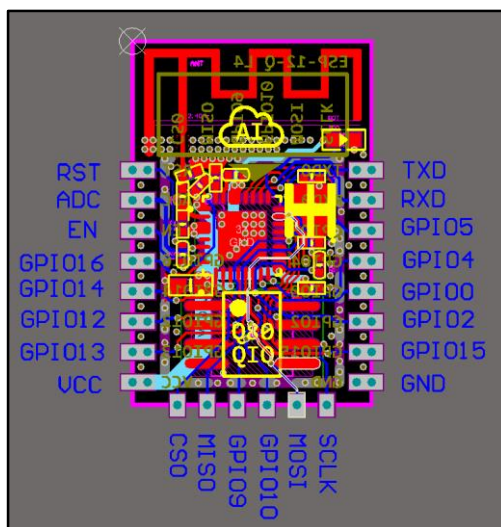


图 1-1 ESP8266 引脚

表 1-2 引脚功能定义

序号	Pin	脚名称	功能说明
1	RST	复位模组	
2	ADC	A/D 转换结果。输入电压范围 0~1V，取值范围： 0~1024	
3	EN	芯片使能端，高电平有效	
4	IO16	GPIO16; 接到 RST 管脚时可做 deep sleep 的唤醒。	
5	IO14	GPIO14; HSPI_CLK	
6	IO12	GPIO12; HSPI_MISO	
7	IO13	GPIO13; HSPI_MOSI; UART0_CTS	
8	VCC	3.3V 供电	
9	CS0	片选	
10	MISO	从机输出主机输入	

11	IO9	GPIO9
12	IO10	GPIO10
13	MOSI	主机输出从机输入
14	SCLK	时钟
15	GND	GND
16	IO15	GPIO15; MTDO; HSPICS; UART0_RTS
17	IO2	GPIO2; UART1_TXD
18	IO0	GPIO0
19	IO4	GPIO4
20	IO5	GPIO5
21	RXD	UART0_RXD; GPIO3
22	TXD	UART0_TXD; GPIO1

- 1) 模块使用前手机端的准备，这里只有**安卓端**的支持。首先，我们要先确定手机端 WIFI 的 IP 地址，进入 WIFI,点击“高级”选项，记录 **IP 地址**，如图 1-2，这是手机端的截图。当然每部手机的具体情况不同，可以百度查询查看 IP 的方法。



图 1-2 IP 地址

- 2) 知道 IP 地址后,那么下面是打开我们提供的手机 APP(借用小米应用的软件: TCP 连接, 见图 1-3, 在此先谢声), 然后安装好 APP, 打开。可以看到在软件顶部写着“**正在监听的端口: xxxx**”我这里是 1234, 记录此端口, 待会要用。左右滑动窗口, 来到如图 1-4。

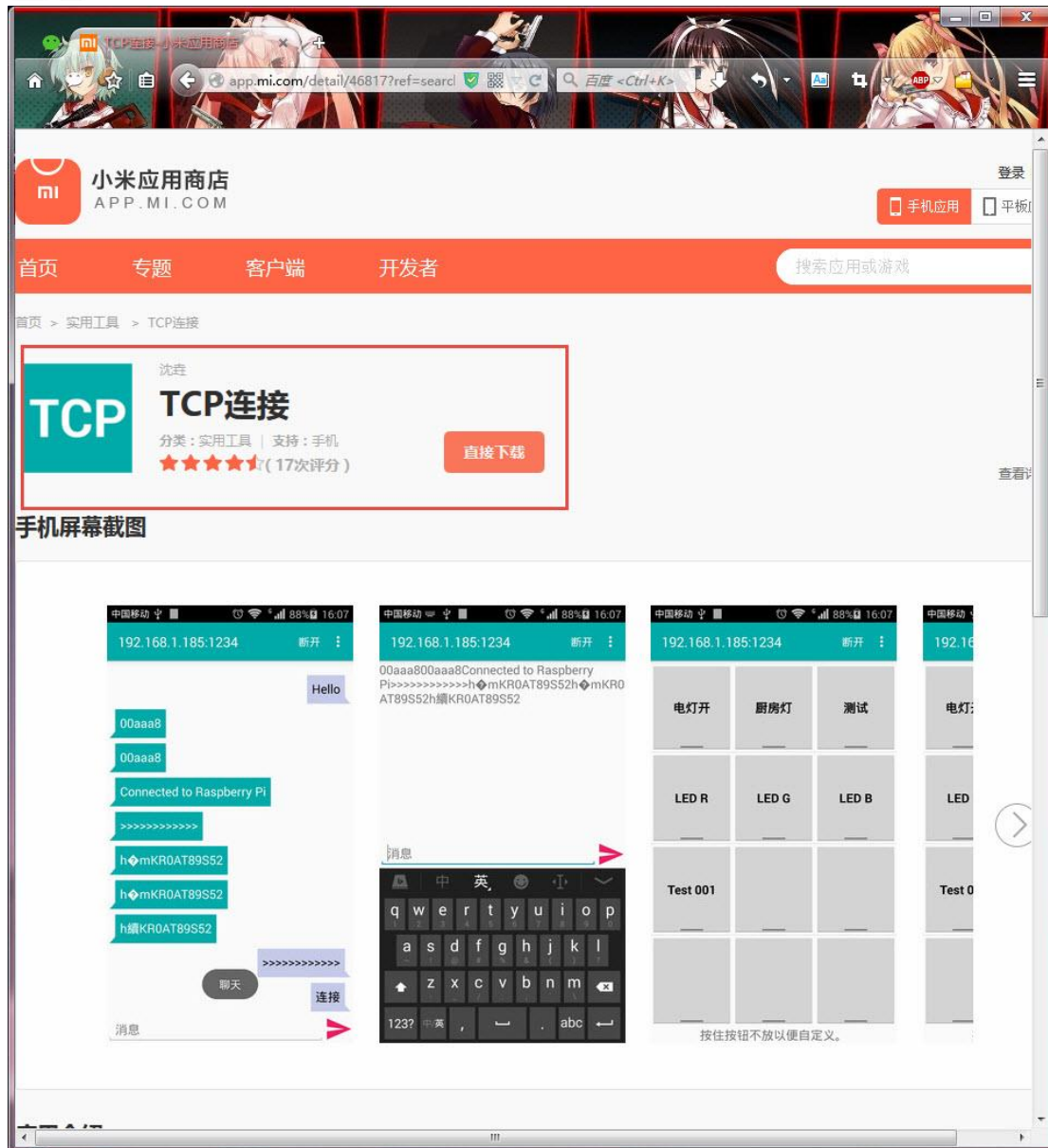


图 1-3 TCP 连接 APP 下载

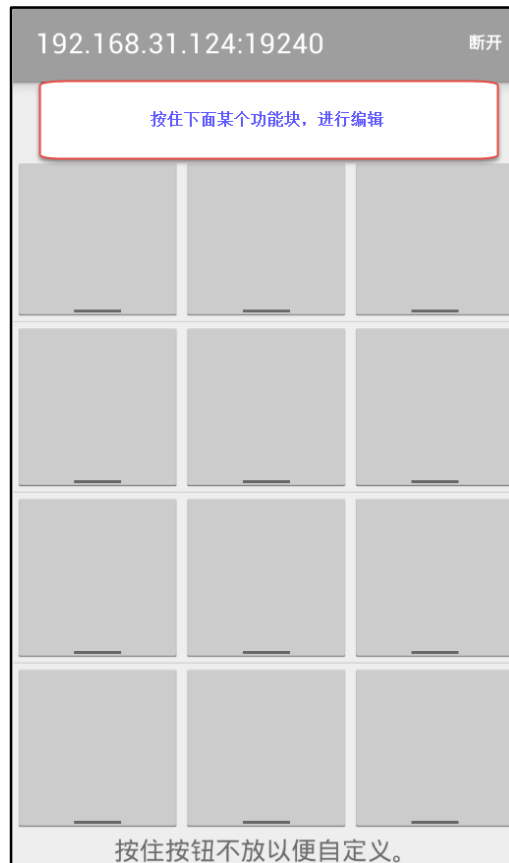


图 1-4 打开编辑窗口

- 3) 对我们需要的按钮进行编辑，我们要控制三个 LED 和一个蜂鸣器，此时，需要四个按钮。长按打开按钮编辑如图 1-5，我们设置 **LED1 灯灭** 的消息为 1，**亮** 的消息为 0，根据数字自己设置，LED2、LED3 和蜂鸣器以此类推。完成后，是对例程进行修改，因为每个人手机 IP 不同，按钮的设置也可能不同。



图 1-5 按钮编辑

- 4) 接下来是对开发板 WIFI 的设置，在第一节我们讲解了三种模式，我们便是要使用 STA 模式，开发板和手机需要连接同一个热点。根据自己的热点名称和密钥进行修改，然后是修改刚才我们记录的手机端 IP 地址和手机端 TCP 软件正在监听的端口，如代码 1-1。

代码 1-1 相关定义

```
01 /*要连接的热点的名称*/
02 #define User_ESP8266_ApSsid          "yingshi"
03 /*要连接的热点的密钥*/
04 #define User_ESP8266_ApPwd          "yingshi123"
05 /*要连接的服务器的 IP*/
06 #define User_ESP8266_TcpServer_IP    "192.168.31.149"
07 /*要连接的服务器的端口*/
08 #define User_ESP8266_TcpServer_Port  "1234"
09
```

- 5) 例程中 bsp_esp8266.c 文件和 bsp_esp8266.h 文件这里就不多做讲解，这是有关于 ESP8266 各个功能的操作函数。需要用到，可以再去了解。

6) 下面是对 main 函数的讲解。或许看上去很长，其实很好理解，我们可以看成三个部分，第一部分是对各项功能的初始化，有串口、LED、BEEP、系统时钟和 wifi 的初始化，然后是调用函数对 wifi 的配置，控制模块连接上热点，其中，每个函数都可以按 F12 查看其定义，都有中文描述。最后是通过无限循环，使能接收，将手机端发出的数据，通过 wifi 接收，判断数据，实现指定的功能。可以看到，这里有个 **atoi** 函数，作用是将接收的字符串转为整形数。这样我们使用 **switch** 语句进行选择。例如接收到的是‘0’，那么 LED1 就亮了。

代码 1-2 主函数

```
01 int main(void)
02 {
03     uint8_t pCH;
04     /* 调试串口初始化配置，115200-N-8-1.使能串口发送和接受 */
05     DEBUG_USART_Init();
06     /* 初始化 LED */
07     LED_GPIO_Init();
08     /* 初始化蜂鸣器 */
09     BEEP_GPIO_Init();
10     /* 系统时钟 */
11     SysTick_Init();
12     /* wifi 初始化 */
13     ESP8266_Init();
14     printf("正在配置 ESP8266 .....\\n");
15
16     if (ESP8266_AT_Test()) {
17         printf("AT test OK\\n");
18     }
19     printf("\\n< 1 >\\n");
20     if (ESP8266_Net_Mode_Choose(STA)) {
21         printf("ESP8266_Net_Mode_Choose OK\\n");
22     }
23     printf("\\n< 2 >\\n");
24     while (!ESP8266_JoinAP(User_ESP8266_ApSsid, User_ESP8266_ApPwd));
25     printf("\\n< 3 >\\n");
26     ESP8266_Enable_MultipleId(DISABLE);
27     while
    (!ESP8266_Link_Server(enumTCP, User_ESP8266_TcpServer_IP, User_ESP8266_TcpServer_Port, Single_ID_0));
28     printf("\\n< 4 >\\n");
29     while (!ESP8266_UnvarnishSend());
30     printf("配置 ESP8266 完毕\\n");
31
32     /* 无限循环 */
33     while (1) {
34
35         ESP8266_ReceiveString(ENABLE);
36         if (strEsp8266_Fram_Record .InfBit .FramFinishFlag ) {
37             strEsp8266_Fram_Record .Data_RX_BUF [ strEsp8266_Fram_Record .InfBit .FramLength ] = '\\0';
38             printf ( "\\r\\n%s\\r\\n", strEsp8266_Fram_Record .Data_RX_BUF );
39             /*将接收到的字符串转成整形数*/
40             pCH=atoi(strEsp8266_Fram_Record .Data_RX_BUF);
```

```
41
42     switch (pCH) {
43     case 0:
44         LED1_ON;
45         break;
46
47     case 1:
48         LED1_OFF;
49         break;
50
51     case 2:
52         LED2_ON;
53         break;
54
55     case 3:
56         LED2_OFF;
57         break;
58
59     case 4:
60         LED3_ON;
61         break;
62
63     case 5:
64         LED3_OFF;
65         break;
66
67     case 6:
68         BEEP_ON;
69         break;
70
71     case 7:
72         BEEP_OFF;
73         break;
74
75     }
76 }
77 }
78 }
```

1.3 实验现象

下面的一段话是总结：

保证 JP1、JP2、JP6 和 CN10 端子上的 5 个跳线帽插入正确，查看手机端 wifi 的地址，根据手机不同自行百度。根据手机端的 WIFI 地址和端口，修改例程中需要连接的终端地址和端口，编译程序。

使用开发板配套的 MINI USB 线连接到开发板标示“调试串口”字样的 MIMI USB 接口，在电脑端打开串口调试助手工具，选择对应端口，设置参数为 115200 8-NO-1，下载程序之后，在串口调试助手窗口可接收到信息。

在手机端打开 TCP 软件，在软件窗口左右移动，移到终端窗口等待自动连接上 WIFI。连接完成，可以在手机窗口发送信息至开发板端 wifi，也可以向右移动

窗口到按钮端。设置好发送的状态以及信息这里将发送 1 设置为 LED 开，发送 0 设置为 LED 关。具体功能根据个人要求修改。如图 1-6 所示，此时按下手机端的按钮，即可实现对开发板的控制。

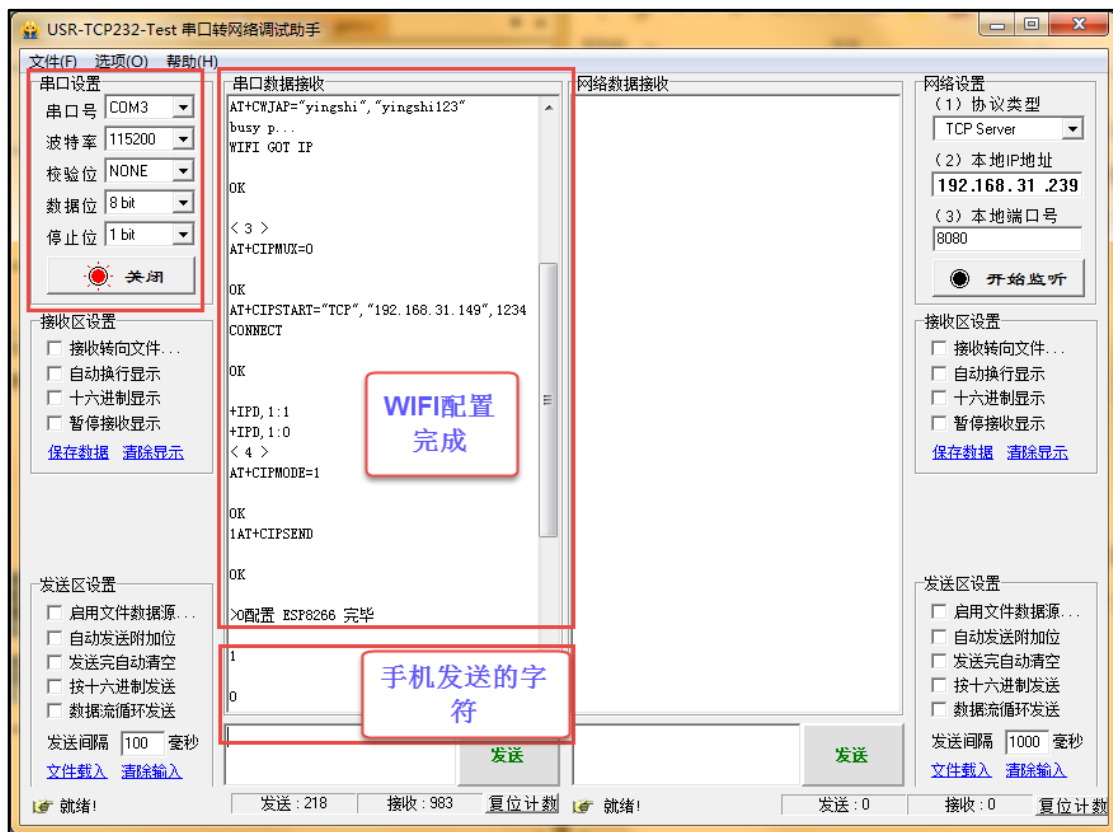


图 1-6 电脑串口调试助手