AN1509B ATK-ESP8266 WIFI 模块使用说明

本应用文档（AN1509B，对应 ALIENTEK 探索者 STM32F407 开发板扩展实验 13）将教大家如何在 ALIENTEK 探索者 STM32F407 开发板上使用 ATK-ESP8266 WIFI 模块。

本文档分为如下几部分：

1，ATK-ESP8266 WIFI 模块简介

2，硬件连接

3，软件实现

4，验证

1、ATK-ESP8266 WIFI 模块简介

ATK-ESP8266 是 ALIENTEK 推出的一款高性能的 UART-WiFi（串口-无线 WIFI）模块，ATK-ESP8266 板载正点原子团队自主开发的 ATK-ESP-01 模块，该模块通过 FCC，CE 认证，可直接用于产品出口欧美地区。

ATK-ESP8266 模块采用串口（LVTTL）与 MCU（或其他串口设备）通信，内置 TCP/IP 协议栈，能够实现串口与 WIFI 之间的转换。

通过 ATK-ESP8266 模块，传统的串口设备只是需要简单的串口配置，即可通过网络（WIFI）传输自己的数据。

ATK-ESP8266 模块支持 LVTTL 串口，兼容 3.3V 和 5V 单片机系统，可以很方便的与你的产品进行连接。模块支持串口转 WIFI STA、串口转 AP 和 WIFI STA+WIFI AP 的模式，从而快速构建串口-WIFI 数据传输方案，方便你的设备使用互联网传输数据。

ATK-ESP8266 模块非常小巧（29mm*19mm），模块通过 6 个 2.54mm 间距的排针与外部连接，模块外观如图 1.1 所示：

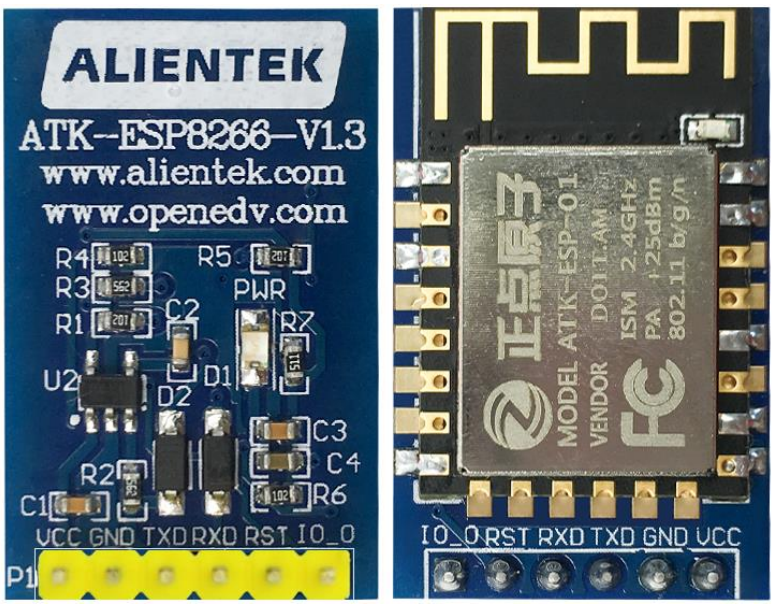

图 1.1 ATK-ESP8266 模块外观图

图 1.1 中，各引脚的详细描述如表 1.1 所示：

序号	名称	说明
1	VCC	电源（3.3V~5V）
2	GND	电源地
3	TXD	模块串口发送脚（TTL 电平，不能直接接 RS232 电平!），可接单片机的 RXD
4	TXD	模块串口接收脚（TTL 电平，不能直接接 RS232 电平!），可接单片机的 TXD
5	RST	复位（低电平有效）
6	IO_0	用于进入固件烧写模式，低电平是烧写模式，高电平是运行模式（默认状态）

表 1.1 ATK-ESP8266 模块各引脚功能描述

ATK-ESP8266 WIFI 模块的原理图如图 1.2 所示：

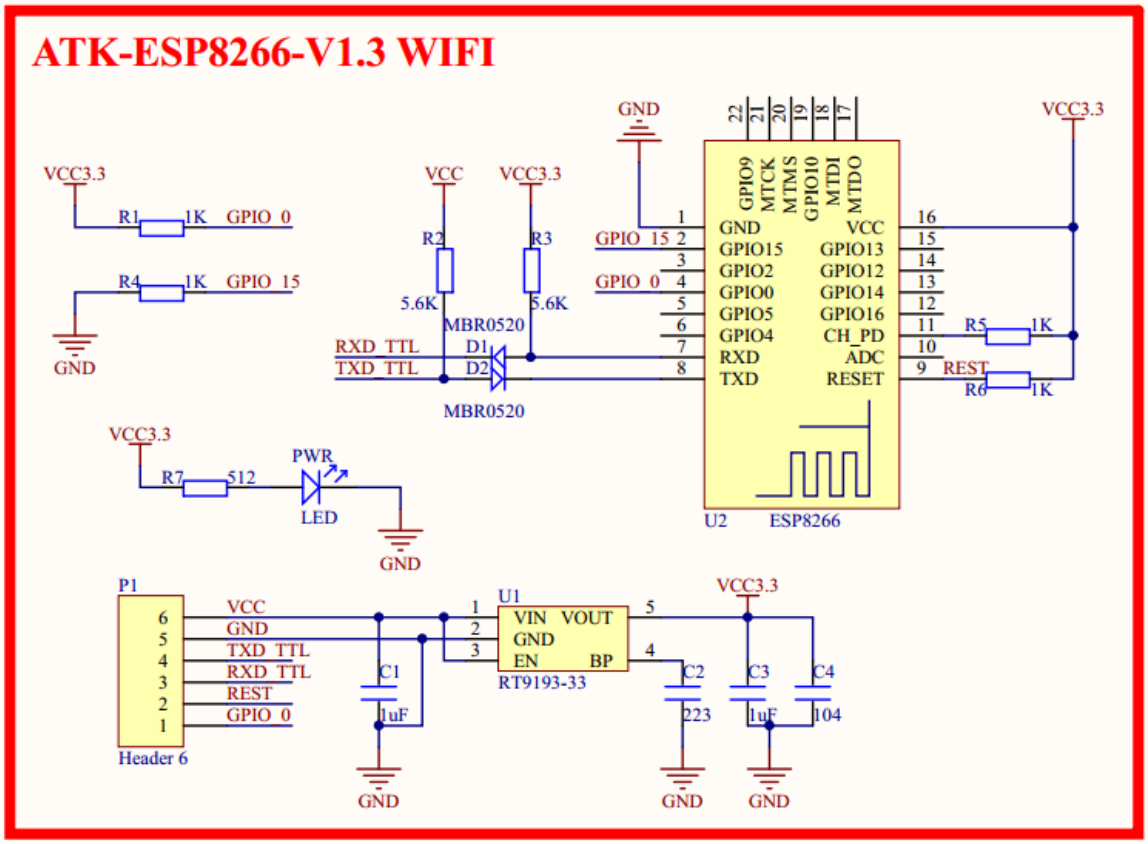

图 1.2 ATK-ESP8266 WIFI 模块原理图

1.1 模块使用

本文档，我们将介绍大家如何通过 ALIENTEK 探索者 STM32F407 开发板连接到 ATK-ESP8266 WIFI 模块，实现：串口无线 AP（COM-AP）、串口无线 STA（COM-STA）和串口无线 AP+STA（COM-AP+STA）这 3 个模式的测试，每个模式又包含 TCP 服务器、TCP 客户端和 UDP 这 3 个子模式，本节我们将详细介绍这些模式的使用。

模块配置可以通过串口配置，这里我们通过开发板串口配置，并实现通信。

1.1.1 AT 指令简介

在 AT 模式下可以通过串口的 AT 指令对系统参数做配置。指令格式如下：

AT+<COMMAND>=<VALUE>\r\n

根据不同指令，模块将返回不同的返回值。其中\r\n 为回车换行符，用 16 进制表示，

就是 0X0D，0X0A。

例如：“AT+CWMODE?\r\n”查询当前模块的 WIFI 模式。

例如：“AT+CWMODE=3\r\n”设置模块 WIFI 模式为 AP+STA 模式。

ATK-ESP8266 模块支持的指令列表如表 1.1.1.1 所示：

RST	重启模块
GMR	查看模块版本信息
CWMODE	设置模块 WIFI 模式
CWJAP	设置模块加入 AP 热点
CWLAP	列表当前可用 AP 热点
CWQAP	退出当前连接的 AP 热点
CWSAP	设置 AP 模式下的 WIFI 参数
CWLIF	查看已接入设备的 IP
CIPSTATUS	获得连接状态
CIPSTART	建立 TCP 连接或注册 UDP 端口号
CIPSEND	发送数据
CIPCLOSE	关闭 TCP 或 UDP
CIFSR	获取本地 IP 地址
CIPMUX	启动多连接
CIPSERVER	配置为服务器
CIPMODE	设置模块传输方式
CIPSTO	设置服务器超时时间
CIUPDATE	网络固件升级

表 1.1.1.1ATK-ESP8266 模块 AT 指令表

1.1.2 串口无线 WIFI（COM-AP）

串口无线 WIFI（COM-AP）模式，模块作为无线 WIFI 热点，允许其他 WIFI 设备连接到本模块，实现串口与其他设备之间的无线（WIFI）数据转换互传。该模式下，根据应用场景的不同，可以设置 3 个子模式：TCP 服务器、TCP 客户端，UDP。

接下来看看如何听过 AT 指令配置模块，达到我们所要的功能，这里仅列出必要配置，配置准备：1，模块处于默认设置（即出厂设置）；2，准备一个带 WIFI 功能的设备，如智能手机、PAD、笔记本电脑等。

串口无线 WIFI AP 模式，TCP 服务器配置，如表 1.1.2.1 所示：

发送指令	作用
AT+CWMODE=2	设置模块 WIFI 模式为 AP 模式
AT+RST	重启生效
AT+CWSAP=”ATK-ESP8266”,”12345678”,1,4	设置模块的 AP 参数：SSID 为 ATK-ESP8266，密码为 12345678，通道号为 1，加密方式为：WPA_WPA2_PSK
AT+CIPMUX=1	开启多连接
AT+CIPSERVER=1,8086	开启 SERVER 模式，设置端口为 8086
AT+CIPSEND=0,25	向 ID0 发送 25 字节数据包

表 1.1.2.1 串口无线 WIFI AP TCP 服务器配置

串口无线 WIFI AP 模式，TCP 客户端配置，如表 1.1.2.2 所示：

发送指令	作用

AT+CWMODE=2	设置模块 WIFI 模式为 AP 模式
AT+RST	重启生效
AT+CWSAP=”ATK-ESP8266”,”12345678”,1,4	设置模块的 AP 参数：SSID 为 ATK-ESP8266，密码为 12345678，通道号为 1，加密方式为：WPA_WPA2_PSK
AT+CIPMUX=0	开启单连接
AT+CIPSTART="TCP","192.168.4.XXX”,8086	建立 TCP 连接到” 192.168.4.XXX”,8086
AT+CIPMODE=1	开启透传模式（仅单连接 client 时支持）
AT+CIPSEND	开始发送数据

表 1.1.2.2 串口无线 WIFI AP 模式 TCP 客户端配置

串口无线 WIFI AP 模式，UDP 配置，如表 1.1.2.3 所示：

发送指令	作用
AT+CWMODE=2	设置模块 WIFI 模式为 AP 模式
AT+RST	重启生效
AT+CWSAP=”ATK-ESP8266”,”12345678”,1,4	设置模块的 AP 参数：SSID 为 ATK-ESP8266，密码为 12345678，通道号为 1，加密方式为：WPA_WPA2_PSK
AT+CIPMUX=0	开启单连接
AT+CIPSTART="UDP","192.168.4.XXX",8086	建立 UDP 连接到”192.168.4.XXX”,8086
AT+CIPSEND=25	向目标 UDP 发送 25 字节数据

表 1.1.2.3 串口无线 WIFI AP 模式 UDP 配置

1.1.3 串口无线 STA（COM-STA）

串口无线 STA（COM-STA）模式，模块作为无线 WIFI STA，用于连接到无线网络，实现串口与其他设备之间的无线（WIFI）数据转换互传。该模式下，根据应用场景的不同，可以设置 3 个子模式：TCP 服务器、TCP 客户端，UDP。

接下来看看如何听过 AT 指令配置模块，达到我们所要的功能，这里仅列出必要配置，配置准备：1，模块处于默认设置（即出厂设置）；2，准备一个无线路由器，且路由器开启 DHCP 服务。

串口无线 STA 模式，TCP 服务器配置，如表 1.1.3.1 所示：

发送指令	作用
AT+CWMODE=1	设置模块 WIFI 模式为 STA 模式
AT+RST	重启模块并生效
AT+CWJAP="ALIENTEK","15902020353"	加入 WIFI 热点：ALIENTEK，密码为：15902020353
AT+CIPMUX=1	开启多连接
AT+CIPSERVER=1,8086	开启服务器，端口号为 8086
AT+CIPSEND=0,25	向 ID0 发送 25 字节的数据

表 1.1.3.1 串口无线 STA 模式 TCP 服务器设置

串口无线 STA 模式，TCP 客户端配置，如表 1.1.3.2 所示：

发送指令	作用
AT+CWMODE=1	设置模块 WIFI 模式为 STA 模式
AT+RST	重启模块并生效

AT+CWJAP="ALIENTEK","15902020353"	加入 WIFI 热点：ALIENTEK，密码为：15902020353
AT+CIPMUX=0	开启单连接
AT+CIPSTART="TCP","192.168.1.XXX",8086	建立 TCP 连接到” 192.168.1.XXX”,8086
AT+CIPMODE=1	开启透传模式
AT+CIPSEND	开始传输

表 1.1.3.2 串口无线 STA 模式 TCP 客户端配置

串口无线 STA 模式，UDP 配置，如表 1.1.3.3 所示：

发送指令	作用
AT+CWMODE=1	设置模块 WIFI 模式为 STA 模式
AT+RST	重启模块并生效
AT+CWJAP="ALIENTEK","15902020353"	建加入 WIFI 热点：ALIENTEK，密码为：15902020353
AT+CIPMUX=0	开启单连接
AT+CIPSTART="UDP","192.168.1.XXX",8086	建立 UDP 连接到” 192.168.4.XXX”,8086
AT+CIPSEND=25	向目标 UDP 发送 25 字节数据

表 1.1.3.3 串口无线 STA 模式 UDP 配置

1.1.4 串口无线 AP+STA（COM-AP+STA）

串口无线 AP+STA（COM-AP+STA）模式，模块既作无线 WIFI AP，又作无线 STA，其他 WIFI 设备可以连接到该模块，模块也可以连接到其他无线网络，实现串口与其他设备之间的无线（WIFI）数据转换互传。该模式下，根据应用场景的不同，可以设置 9 个子模式：（TCP 服务器、TCP 客户端，UDP）||（TCP 服务器、TCP 客户端，UDP）。

接下来看看如何听过 AT 指令配置模块，达到我们所要的功能，这里仅列出必要配置，配置准备：1，模块处于默认设置（即出厂设置）；2，准备一个带 WIFI 功能的设备，如智能手机、PAD、笔记本电脑等；3，准备一个无线路由器，且路由器开启 DHCP 服务。

下面仅介绍 3 种模式。AP 下作服务器，STA 的三种模式。

串口无线 AP+STA 模式，AP 作 TCP 服务器，STA 做 TCP 服务器的配置，如表 1.1.4.1 所示：

发送指令	作用
AT+CWMODE=3	设置模块 WIFI 模式为 AP+STA
AT+RST	重启模块并生效
AT+CWSAP="ATK-ESP8266","12345678",1,4	加入 WIFI 热点：ALIENTEK，密码为：15902020353
AT+CWJAP="ALIENTEK","15902020353"	设置模块的 AP 参数：SSID 为 ATK-ESP8266，密码为 12345678，通道号为 1，加密方式为：WPA_WPA2_PSK
AT+CIPMUX=1	开启多连接
AT+CIPSERVER=1,8086	开启服务器，端口号 8086
AT+CIPSTO=1200	设置服务器超时时间 1200s
AT+CIPSEND=0,25	向 ID0 发送数据
AT+CIPSEND=1,25	向 ID1 发送数据

表 1.1.4.1 串口无线 AP+STA 模式 AP 作 TCP 服务器 STA 作 TCP 服务器配置

串口无线 AP+STA 模式，AP 作 TCP 服务器，STA 作 TCP 客户端的配置，如表 1.1.4.2 所示：

发送指令	作用
AT+CWMODE=3	设置模块 WIFI 模式为 AP+STA
AT+RST	重启模块并生效
AT+CWSAP="ATK-ESP8266","12345678",1,4	加入 WIFI 热点：ALIENTEK，密码为：15902020353
AT+CWJAP="ALIENTEK","15902020353"	设置模块的 AP 参数：SSID 为 ATK-ESP8266，密码为 12345678，通道号为 1，加密方式为：WPA_WPA2_PSK
AT+CIPMUX=1	开启多连接
AT+CIPSERVER=1,8086	开启服务器，端口号 8086
AT+CIPSTO=1200	设置服务器超时时间 1200s
AT+CIPSTART=0,"TCP","192.168.1.XXX",8086	STA 作为 ID0 连接到 192.168.1.XXX，8086
AT+CIPSEND=0,25	向 ID0 发送数据
AT+CIPSEND=1,25	向 ID1 发送数据

表 1.1.4.2 串口无线 AP+STA 模式 AP 作 TCP 服务器 STA 作 TCP 客户端配置

串口无线 AP+STA 模式，AP 作 TCP 服务器，STA 作 UDP 的配置，如表 1.1.4.3 所示

发送指令	作用
AT+CWMODE=3	设置模块 WIFI 模式为 AP+STA
AT+RST	重启模块并生效
AT+CWSAP="ATK-ESP8266","12345678",1,4	加入 WIFI 热点：ALIENTEK，密码为：15902020353
AT+CWJAP="ALIENTEK","15902020353"	设置模块的 AP 参数：SSID 为 ATK-ESP8266，密码为 12345678，通道号为 1，加密方式为：WPA_WPA2_PSK
AT+CIPMUX=1	开启多连接
AT+CIPSERVER=1,8086	开启服务器，端口号 8086
AT+CIPSTO=1200	设置服务器超时时间 1200s
AT+CIPSTART=0,"UDP","192.168.1.XXX",8086	STA 作为 ID0 连接到 192.168.1.XXX，8086
AT+CIPSEND=0,25	向 ID0 发送数据
AT+CIPSEND=1,25	向 ID1 发送数据

表 1.1.4.3 串口无线 AP+STA 模式 AP 作 TCP 服务器 STA 作 UDP 配置

2、硬件连接

2.1 功能简介

本实验功能简介：本实验用于测试 ATK-ESP8266 模块，总共包括 3 大项测试，每大项又包括 3 个子模式的测试，下面分别介绍。

2.1.1 串口无线 WIFI（COM-AP）测试

该项测试网络连接方式：WIFI 设备<-------->^①ATK-ESP8266 模块

注①：--------，表示通过无线连接（WIFI），下同

通过按 KEY_UP 键选择此模式，进入此模式后，首先会进入到串口 WIFI AP 工作模式选择界面，总共有 3 种模式可供选择：TCP 服务器、TCP 客户端、UDP。通过 KEY0/KEY1 选择模式，通过 KEY_UP 确定。点击确定后，服务器模式（TCP 服务器）不需要手动输入 IP 地址，直接进入下一步，开始配置模块。但是对于 TCP 客户端/UDP 模式，还会需要手动输入远端 IP 地址，此时界面会提示：远端 IP 设置，通过屏幕显示的虚拟键盘，输入远端 IP 地址，输入完后，可以点击虚拟键盘的“连接”按键，进入下一步，开始配置模块，此时，开发板将对模块进行配置，配置成功后，进入数据收发测试。

进入测试后，开发板液晶会显示相关提示信息，方便大家测试，信息包括：

IP 地址：如果是服务器模式，那么 IP 地址表示 ATK-ESP8266 模块的 IP 地址。如果是客户端模式，则表示模块将要连接到的目标 IP 地址。

端口：为方便测试，所有模式的端口都固定为 8086。

状态：表示连接状态，如果连接建立，则显示连接成功，如果断开，则显示连接失败。

模式：显示当前的工作模式（TCP 服务器、TCP 客户端、UDP）。

发送数据：显示当需要发送的数据，每按一次 KEY0，发送一次数据，会在这个区域显示发送的内容，当发送完以后，过一段时间，该区域自动清空。

接收数据：显示 ATK-ESP8266 模块接收到的数据，对方发过来的数据，将显示在这里，每次收到新内容，会将旧内容清空，然后显示新内容，否则旧内容将一直显示。

按 KEY_UP，可以退出此项测试，回到主界面。

对于 3 种子模式：TCP 服务器、TCP 客户端、UDP，他们的数据收发测试界面几乎是一样的，测试方法也几乎一模一样，大同小异，我们就不一一介绍了，在第 4 部分，我们会详细介绍测试步骤。

2.1.2 串口无线 STA（COM-STA）测试

该项测试网络连接方式：WIFI 设备<------>无线路由器<------>ATK-ESP8266 模块（WIFI STA）

通过按 KEY1 键选择此模式，进入此模式后，首先会进入到 WIFI-STA 工作模式选择界面，剩下的操作，同 2.1.1 节一模一样。

2.1.3 串口无线 AP+STA（COM-AP+STA）测试

该项测试网络连接方式：WIFI 设备<------> ATK-ESP8266 模块（WIFI AP+STA）<------>无线路由器<------>WIFI 设备

通过按 KEY0 键选择此模式，进入此模式后，首先会进入到提示界面（请用手机连接到 ATK-ESP8266）,然后按任意键继续，接着会进入到 WIFI AP 模式下的模式选择，接着操作与 2.1.1 节一模一样，完成 AP 模式的设置，接着设置 STA 模式，与 AP 模式相似。配置成功后，进入数据收发测试。

进入测试后，开发板液晶会显示相关提示信息，信息与 AP 模式和 STA 模式有所差别。

端口：为方便测试，所有模式的端口都固定为 8086。

状态：表示连接状态，如果连接建立，则显示连接成功，如果断开，则显示连接失败。

模式：显示当前的工作模式（固定显示：STA+AP 模式）。

AP IP：AP 模式下的 IP 地址。

STA IP：STA 模式下的 IP 地址。

发送数据：显示当需要发送的数据，每按一次 KEY0，发送一次数据，会在这个区域显示发送的内容，当发送完以后，过一段时间，该区域自动清空。

接收数据：显示 ATK-ESP8266 模块接收到的数据，对方发过来的数据，将显示在这里，每次收到新内容，会将旧内容清空，然后显示新内容，否则旧内容将一直显示。

按 KEY_UP，可以退出此项测试，回到主界面。按 KEY0 向 ID0 发送数据，按 KEY1 向 ID1 发送数据。

对于 9 种子模式：（TCP 服务器、TCP 客户端、UDP）||（TCP 服务器、TCP 客户端、UDP），他们的数据收发测试界面几乎是一样的，测试方法也几乎一模一样，大同小异，我们就不一一介绍了，在第 4 部分，我们会详细介绍测试步骤。

2.2 硬件资源准备

本实验所需要的硬件资源如下：

1， ALIENTEK 探索者 STM32F407 开发板 1 个

2， ATK-ESP8266 模块一个

3， 路由器一个

4， WIFI 设备两个（可连接到 WIFI 热点）

2.3 模块与开发版连接

ATK-ESP8266 模块的所有数据，都是通过串口来传输的，所以我们的开发板与模块连接，只需要连接串口即可（当然也要共地），接下来，我们看看 ALIENTEK 探索者 STM32F407 开发板与 ATK-ESP8266 模块的连接方式，本例程通过开发板的串口 3 连接 AT-ESP8266 模块，ALIENTEK 探索者 STM32F407 开发板板载了一个 ATK 模块接口（ATK MODULE），ATK-ESP8266 WIFI 模块可直接插入该接口实现与 ALIENTEK 探索者 STM32F407 开发板的连接。

ATK MODULE 通开发板主芯片的连接原理图，如图 2.3.1 所示：

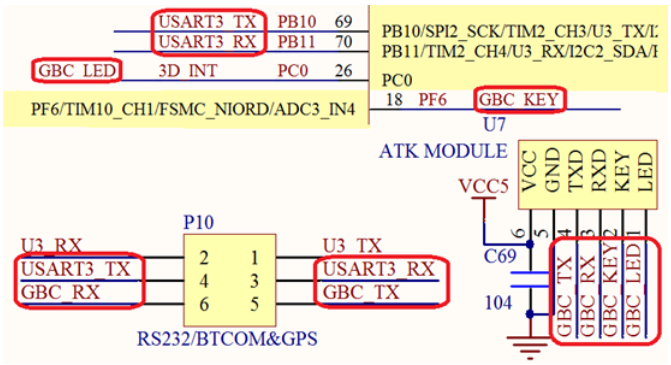

图 2.3.1 ATK-MODULE 接口与 MCU 连接关系

从上图可以看出，ATK-ESP8266 WIFI 模块的串口最简单的办法是连接在开发板的串口 3 上面 ，探索者 F407 只需要用跳线帽短接 P10 的 USART3_RX 和 GBC_TX 以及 USART3_TX 和 GBC_RX 即可实现。且 ATK-ESP8266 WIFI 模块的 RST 信号接在 GBC_KEY 信号上面，IO_0 信号接到 GBC_LED 信号上面。

连接好之后，ALIENTEK 探索者 STM32F407 开发板与 ATK-ESP8266 WIFI 模块的连接关系如表 2.3.1 所示：

ATK-ESP8266 WIFI 模块与开发板连接关系						
ATK-ESP8266 WIFI 模块	VCC	GND	TXD	RXD	RST	IO_0
探索者 STM32F407 开发板	5V	GND	PB11	PB10	PF6	PC0

表 2.3.1 ATK-ESP8266 WIFI 模块同探索者 STM32F4 连接关系表

使用时，我们只需要将 ATK-ESP8266 WIFI 模块插入到开发板的 ATK MODULE 接口即可，如图 2.3.2 所示：

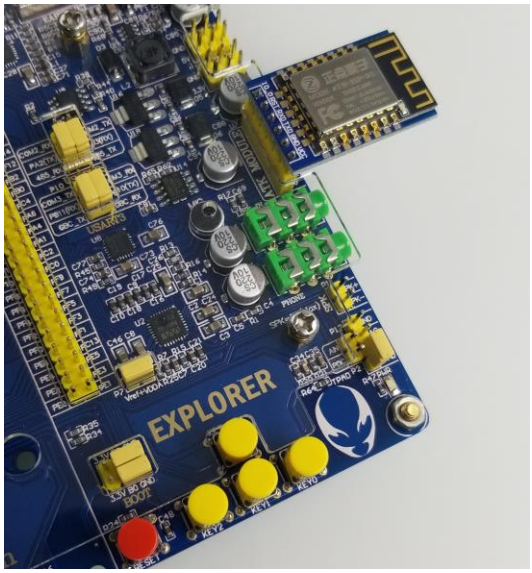

图 2.3.2ATK-ESP8266 WIFI 模块与开发版对接实物图

注意，我们虽然将 RST 和 IO_0 连接到了开发板的 PF6 和 PC0，但是本例程并没有控制这两个信号，所以没有处理。另外探索者开发板，连接好之后，记得检查开发板 P10 的跳线帽哦!!**必须短接：PB11（RX）和 GBC_TX 以及 PB10（TX）和 GBC_RX**。

3、软件实现

本实验在探索者 F407 开发板的扩展实验 4（ATK-SIM900A GSM 模块测试实验）基础上进行修改，删除原来的 SIM900A 文件夹，新建 ATK_ESP8266 文件夹，在里面新建：common.c、wifista、wifiap、apsta 和 common.h 等有文件。

打开原工程，删除 SIM900A 分组，并将 ATK_ESP8266 文件夹里面的 4 个.c 文件添加到 ATK_ESP8266 分组，并将 ATK_ESP8266 文件夹，加入到头文件包含路径里面。

最终去掉原工程的一些未用到的.c 文件，最终工程如图 3.1 所示：

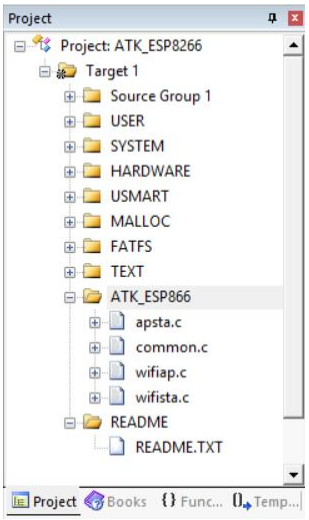

图 3.1 ATK-ESP8266 模块测试实验工程截图

本例程由于代码量较多，我们仅对 ATK-ESP8266 文件夹里面的部分代码（conmon.c、apsta.c、wifista.c、wifiap.c 等），以及 main 函数进行讲解，其他的请大家参考源代码注释。

首先，common.c 文件，该文件是驱动 ATK-ESP8266 模块通信的底层接口函数（AT 指令的发送与接收，模块状态检测等），以及相关函数输入输出显示（IP 输入、模式选择、模

块状态信息显示等）等。接下来，我们介绍几个重要的函数。

第一个是 atk_8266_send_cmd 函数，该函数用于向 ATK_ESP8266 模块发送 AT 指令，该函数代码如下：

```
//向 ATK-ESP8266 发送命令
//cmd:发送的命令字符串
//ack:期待的应答结果,如果为空,则表示不需要等待应答
//waittime:等待时间(单位:10ms)
//返回值:0,发送成功(得到了期待的应答结果)
//        1,发送失败
u8 atk_8266_send_cmd(u8 *cmd,u8 *ack,u16 waittime)
{
    u8 res=0;
    USART3_RX_STA=0;
    u3_printf("%s\r\n",cmd);                    //发送命令
    if(ack&&waittime)                           //需要等待应答
    {
        while(--waittime)                       //等待倒计时
        {
            delay_ms(10);
            if(USART3_RX_STA&0X8000)            //接收到期待的应答结果
            {
                if(atk_8266_check_cmd(ack))
                {
                    printf("ack:%s\r\n",(u8*)ack);
                    break;                      //得到有效数据
                }
                    USART3_RX_STA=0;
            }
        }
        if(waittime==0)res=1;
    }
    return res;
}
```

该函数带 3 个参数，cmd 表示要发送的指令字符串，ack 表示发送指令后期待得到的应答字符串，waittime 表示等待应答的时间（单位：10ms），如：

```
atk_8288_send_cmd("AT+RST","OK",20);
```

表示发送指令：AT+RST 到 WIFI 模块，重启模块；期待的应答为：OK；等待时间为 200ms。

第二个是 atk_rm04_quit_trans 函数，该函数用于控制模块退出透传模式，进入 AT 指令模式。该函数代码如下：

```
//ATK-ESP8266 退出透传模式
//返回值:0,退出成功;
//        1,退出失败
```

```
u8 atk_8266_quit_trans(void)
{
    while((USART3->SR&0X40)==0);        //等待发送空
    USART3->DR='+';
    delay_ms(15);                       //大于串口组帧时间(10ms)
    while((USART3->SR&0X40)==0);        //等待发送空
    USART3->DR='+';
    delay_ms(15);                       //大于串口组帧时间(10ms)
    while((USART3->SR&0X40)==0);        //等待发送空
    USART3->DR='+';
    delay_ms(500);                      //等待 500ms
    return atk_8266_send_cmd("AT","OK",20);//退出透传判断.
}
```

模块退出透传模式只有一种方法，就是在透传状态下发送“+++”，即可退出透传模式，进入 AT 模式。此时的 AT 模式下如果设置模块重启，模块又会自动进入透传模式，所以在重启模块之前，我们需要发送“AT+CIPMODE=0”来关闭透传模式，这样模块重启之后就不会进入 AT 模式了。

接下来介绍三个查询函数：atk_8266_consta_check、atk_8266_get_wanip 和 atk_8266_get_ip，这三个函数代码如下：

```
//获取 ATK-ESP8266 模块的连接状态
//返回值:0,未连接;1,连接成功.
u8 atk_8266_consta_check(void)
{
    u8 *p;
    u8 res;
    if(atk_8266_quit_trans())return 0;          //退出透传
    atk_8266_send_cmd("AT+CIPSTATUS",":",50);
    //发送 AT+CIPSTATUS 指令,查询连接状态
    p=atk_8266_check_cmd("+CIPSTATUS:");
    res=*p;                                     //得到连接状态
    return res;
}
//获取 STA 或者 AP 模式下的 ip 地址
//ipbuf:ip 地址输出缓存区
void atk_8266_get_wanip(u8* ipbuf)
{
    u8 *p,*p1;
        if(atk_8266_send_cmd("AT+CIFSR","OK",50))       //获取 WAN IP 地址失败
        {
            ipbuf[0]=0;
            return;
        }
        p=atk_8266_check_cmd("\"");
        p1=(u8*)strstr((const char*)(p+1),"\"");
```

```
            *p1=0;
            sprintf((char*)ipbuf,"%s",p+1);
}
//获取 AP+STA ip 地址并在指定位置显示
//ipbuf:ip 地址输出缓存区
void atk_8266_get_ip(u8 x,u8 y)
{
            u8 *p;
            u8 *p1;
            u8 *p2;
            u8 *ipbuf;
            u8 *buf;
            p=mymalloc(SRAMIN,32);                                  //申请 32 字节内存
            p1=mymalloc(SRAMIN,32);                                 //申请 32 字节内存
            p2=mymalloc(SRAMIN,32);                                 //申请 32 字节内存
          ipbuf=mymalloc(SRAMIN,32);                                //申请 32 字节内存
            buf=mymalloc(SRAMIN,32);                                //申请 32 字节内存
            if(atk_8266_send_cmd("AT+CIFSR","OK",50))         //获取 WAN IP 地址失败
            {
                  *ipbuf=0;
            }
            else
            {
                  p=atk_8266_check_cmd("APIP,\"");
                  p1=(u8*)strstr((const char*)(p+6),"\"");
                  p2=p1;
                  *p1=0;
                  ipbuf=p+6;
                  sprintf((char*)buf,"AP IP:%s  端口:%s",ipbuf,(u8*)portnum);
                  Show_Str(x,y,200,12,buf,12,0);
                  //显示 AP 模式的 IP 地址和端口
                  p=(u8*)strstr((const char*)(p2+1),"STAIP,\"");
                  p1=(u8*)strstr((const char*)(p+7),"\"");
                  *p1=0;
                  ipbuf=p+7;
                  sprintf((char*)buf,"STA IP:%s  端口:%s",ipbuf,(u8*)portnum);
                  Show_Str(x,y+15,200,12,buf,12,0);
                  //显示 STA 模式的 IP 地址和端口
                  myfree(SRAMIN,p);                 //释放内存
                  myfree(SRAMIN,p1);                //释放内存
                  myfree(SRAMIN,p2);                //释放内存
                  myfree(SRAMIN,ipbuf);             //释放内存
                  myfree(SRAMIN,buf);               //释放内存
            }
```

```
}
```

其中 atk_8266_consta_check 函数，用于检查当前连接（TCP/UDP）是否建立（或存在）；atk_8266_get_wanip 函数，用于获取模块 STA 模式或者 AP 模式下的 IP 地址及 MAC 地址；atk_8266_get_ip 函数，用于获取模块 AP+STA 模式下的 IP 地址及 MAC 地址。

最后我们看下 atk_8266_test，该函数代码如下：

```
//ATK-ESP8266 模块测试主函数
void atk_8266_test(void)
{
    u8 key;
    u8 timex;
    POINT_COLOR=RED;
    Show_Str_Mid(0,30,"ATK-ESP8266 WIFI 模块测试",16,240);
    while(atk_8266_send_cmd("AT","OK",20))//检查 WIFI 模块是否在线
    {
        atk_8266_quit_trans();//退出透传
        atk_8266_send_cmd("AT+CIPMODE=0","OK",200);  //关闭透传模式
        Show_Str(40,55,200,16,"未检测到模块!!!",16,0);
        delay_ms(800);
        LCD_Fill(40,55,200,55+16,WHITE);
        Show_Str(40,55,200,16,"尝试连接模块...",16,0);
    }
        while(atk_8266_send_cmd("ATE0","OK",20));//关闭回显
        atk_8266_mtest_ui(32,30);
    while(1)
    {
        delay_ms(10);
        atk_8266_at_response(1);
        //检查 ATK-ESP8266 模块发送过来的数据,及时上传给电脑
        key=KEY_Scan(0);
        if(key)
        {
            LCD_Clear(WHITE);
            POINT_COLOR=RED;
            switch(key)
            {
                case 1:    //KEY0
                    Show_Str_Mid(0,30,"ATK-ESP WIFI-AP+STA  测试",16,240);
                    Show_Str_Mid(0,50,"正在配置 ATK-ESP8266 模块，请稍
                    等...",12,240);
                    atk_8266_apsta_test();          //AP+STA 测试
                    break;
                case 2:    //KEY1
                    Show_Str_Mid(0,30,"ATK-ESP WIFI-STA  测试",16,240);
                    Show_Str_Mid(0,50,"正在配置 ATK-ESP8266 模块，请稍
```

```
                    等...",12,240);
                    atk_8266_wifista_test();        //WIFI STA 测试
                    break;
                case 4:    //WK_UP
                    atk_8266_wifiap_test();        //WIFI AP 测试
                    break;
            }
            atk_8266_mtest_ui(32,30);
            timex=0;
        }
        if((timex%20)==0)LED0=!LED0;//200ms 闪烁
        timex++;
    }
}
```

该函数是 ATK-ESP8266 模块测试的主程序，先检查模块是否存在，在检测模块正常后，然后初始化模块为 AP 模式，接着进入模式选择界面，最后通过按键选择进入对应的子功能进行测试，通过三个子功能测试函数：atk_8266_apsta_test、atk_8266_wifista_test 和 atk_8266_wifiap_test 进行测试。

common.c 我们就介绍到这里，接下来，我们看看 apsta.c 里面的代码，该文件里面就一个函数：

```
//ATK-ESP8266 AP+STA 模式测试
//用于测试 TCP/UDP 连接
//返回值:0,正常
//      其他,错误代码
u8 atk_8266_apsta_test(void)
{
    u8 netpro;
    u8 key=0;
    u8 timex=0;
    u8 ipbuf[16];  //IP 缓存
    u8 *p;
    u16 t=999;          //加速第一次获取链接状态
    u8 res=0;
    u16 rlen=0;
    u8 constate=0;//连接状态
    p=mymalloc(SRAMIN,100);                                    //申请 32 字节内存
    atk_8266_send_cmd("AT+CWMODE=3","OK",50);          //设置 WIFI AP+STA 模式
    //设置模块 AP 模式的 WIFI 网络名称/加密方式/密码，这几个参数看自己喜好设置
    sprintf((char*)p,"AT+CWSAP=\"%s\",\"%s\",1,4",wifiap_ssid,wifiap_password);
    //设置无线参数:ssid,密码
    atk_8266_send_cmd(p,"OK",1000);                            //设置 AP 模式参数
    //设置连接到的 WIFI 网络名称/加密方式/密码,这几个参数需要根据您自己的路由
    //器设置进行修改!!
    sprintf((char*)p,"AT+CWJAP=\"%s\",\"%s\"",wifista_ssid,wifista_password);
```

```
        //设置无线参数:ssid,密码
        atk_8266_send_cmd(p,"WIFI GOT IP",1000);                    //连接目标路由器
        while(atk_8266_send_cmd("AT+CIFSR","STAIP",20));    //检测是否获得 STA IP
        while(atk_8266_send_cmd("AT+CIFSR","APIP",20));     //检测是否获得 AP IP
        LCD_Clear(WHITE);
        POINT_COLOR=RED;
        Show_Str(30,30,200,16,"ATK-ESP AP+STA 模式测试",16,0);
        atk_8266_send_cmd("AT+CIPMUX=1","OK",50);           //0：单连接，1：多连接
        delay_ms(500);
        sprintf((char*)p,"AT+CIPSERVER=1,%s",(u8*)portnum);
        atk_8266_send_cmd(p,"OK",50);          //开启 Server 模式，端口号为 8086
        delay_ms(500);
        atk_8266_send_cmd("AT+CIPSTO=1200","OK",50);        //设置服务器超时时间

    PRESTA:
        netpro=atk_8266_netpro_sel(50,30,(u8*)ATK_ESP8266_CWMODE_TBL[0]);
        //AP+STA 模式网络模式选择
        if(netpro&0X02)     //STA UDP
        {
                        LCD_Clear(WHITE);
                        POINT_COLOR=RED;
                        Show_Str_Mid(0,30,"ATK-ESP WIFI-STA 测试",16,240);
                        Show_Str(30,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);
                        if(atk_8266_ip_set("WIFI-STA 远端 UDP IP 设置","UDP 模式
                        ",(u8*)portnum,ipbuf))goto PRESTA;  //IP 输入
                        sprintf((char*)p,"AT+CIPSTART=0,\"UDP\",\"%s\",%s",ipbuf,(u8*)port
                        num);
                       //配置目标 UDP 服务器,及 ID 号，STA 模式下为 0
                        LCD_Clear(WHITE);
                        Show_Str_Mid(0,30,"ATK-ESP WIFI-STA 测试",16,240);
                        Show_Str(30,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);
                        atk_8266_send_cmd(p,"OK",200);
                        netpro=atk_8266_mode_cofig(netpro);         //AP 模式网络模式配置
        }
        else        //TCP
        {
            if(netpro&0X01)         //STA TCP Client
            {
                LCD_Clear(WHITE);
                POINT_COLOR=RED;
                Show_Str_Mid(0,30,"ATK-ESP WIFI-STA  测试",16,240);
                Show_Str(30,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);
                if(atk_8266_ip_set("WIFI-STA  远端 IP 设置",(u8*)ATK_ESP8266_WO
                RKMODE_TBL[netpro],(u8*)portnum,ipbuf))goto PRESTA;
```

```
                //IP 输入
                sprintf((char*)p,"AT+CIPSTART=0,\"TCP\",\"%s\",%s",ipbuf,(u8*)portnum);
                //配置目标 TCP 服务器,及 ID 号，STA 模式下为 0
                while(atk_8266_send_cmd(p,"OK",200))
                {
                    LCD_Clear(WHITE);
                    POINT_COLOR=RED;
                    Show_Str_Mid(0,40,"WK_UP:返回重选",16,240);
                    Show_Str(30,80,200,12,"ATK-ESP  连接 UDP 失败",12,0); //连接失败

                    key=KEY_Scan(0);
                    if(key==4)goto PRESTA;
                }
                netpro=atk_8266_mode_cofig(netpro);         //AP 模式网络模式配置
            }
            else netpro=atk_8266_mode_cofig(netpro);        //TCP SERVER 不用配置
        }
        LCD_Clear(WHITE);
        POINT_COLOR=RED;
        Show_Str_Mid(0,30,"ATK-ESP WIFI-STA+AP  测试",16,240);
        Show_Str(15,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);
        LCD_Fill(15,50,239,50+12,WHITE);                    //清除之前的显示
        Show_Str_Mid(0,50,"WK_UP:退出  KEY0:ID0 发送  KEY1:ID1 发送",12,240);
        LCD_Fill(15,80,239,80+12,WHITE);
        atk_8266_get_ip(15,65);                             //STA+AP 模式,获取  IP,并显示

        Show_Str(15,95,200,12,"连接状态:",12,0);            //连接状态
        Show_Str(15,110,200,12,"STA 模式:",12,0);           //STA 连接状态
        Show_Str(120+15,110,200,12,"AP 模式:",12,0);        //AP 连接状态
        Show_Str(15,125,200,12,"发送数据:",12,0);           //发送数据
        Show_Str(15,140,200,12,"接收数据:",12,0);           //接收数据
        atk_8266_wificonf_show(15,195,"请设置路由器无线参数
        为:",(u8*)wifista_ssid,(u8*)wifista_encryption,(u8*)wifista_password);
        POINT_COLOR=BLUE;
        Show_Str(48+15,110,200,12,(u8*)ATK_ESP8266_WORKMODE_TBL[netpro&0X03],
        12,0);                                              //STA 连接状态
        Show_Str(162+15,110,200,12,(u8*)ATK_ESP8266_WORKMODE_TBL[netpro>>4],1
        2,0);                                               //AP 连接状态
        USART3_RX_STA=0;
        while(1)
        {
            key=KEY_Scan(0);
            if(key==4)                                      //WK_UP  退出测试
            {
```

```
                res=0;
                break;
        }
        else if(key==1)                                 //KEY0  向 ID0 发送数据
        {
                sprintf((char*)p,"ATK-8266 模块 ID0 发数据%02d\r\n",t/10);//测试数据
                Show_Str(15+54,125,200,12,p,12,0);
                atk_8266_send_cmd("AT+CIPSEND=0,25","OK",200);
                //发送指定长度的数据
                delay_ms(200);
                atk_8266_send_data(p,"OK",100);         //发送指定长度的数据
                timex=100;
        }
        else if(key==2)                                 //KEY1  向 ID1 发送数据
        {
                sprintf((char*)p,"ATK-8266 模块 ID1 发数据%02d\r\n",t/10);//测试数据
                Show_Str(15+54,125,200,12,p,12,0);
                atk_8266_send_cmd("AT+CIPSEND=1,25","OK",200);
                //发送指定长度的数据
                delay_ms(200);
                atk_8266_send_data(p,"OK",100);         //发送指定长度的数据
                timex=100;
        }

 if(timex)timex--;
 if(timex==1)LCD_Fill(30+54,125,239,122,WHITE);
 t++;
 delay_ms(10);
 if(USART3_RX_STA&0X8000)                               //接收到一次数据了
{
        rlen=USART3_RX_STA&0X7FFF;                      //得到本次接收到的数据长度
        USART3_RX_BUF[rlen]=0;                          //添加结束符
        printf("%s",USART3_RX_BUF);                     //发送到串口
        sprintf((char*)p,"收到%d 字节,内容如下",rlen);//接收到的字节数
        LCD_Fill(15+54,140,239,130,WHITE);
        POINT_COLOR=BRED;
        Show_Str(15+54,140,156,12,p,12,0);              //显示接收到的数据长度
        POINT_COLOR=BLUE;
        LCD_Fill(15,155,239,319,WHITE);
        Show_Str(15,155,180,190,USART3_RX_BUF,12,0);//显示接收到的数据
        USART3_RX_STA=0;
        if(constate!='+')t=1000;                //状态为还未连接,立即更新连接状态
        else t=0;                               //状态为已经连接了,10 秒后再检查
}
```

```
        if(t==1000)//连续 10 秒钟没有收到任何数据,检查连接是不是还存在.
        {
            LCD_Fill(15+54,125,239,140,WHITE);
            constate=atk_8266_consta_check();//得到连接状态
            if(constate=='+')Show_Str(15+54,95,200,12,"连接成功",12,0);  //连接状态
            else Show_Str(15+54,95,200,12,"连接失败",12,0);
            t=0;
        }
        if((t%20)==0)LED0=!LED0;
        atk_8266_at_response(1);
    }
        myfree(SRAMIN,p);                   //释放内存
        return res;
    }
```

该代码，就实现了对模块串口 AP+STA 模式各个子模式的测试（TCP 服务器、TCP 客户端、UDP）。这里，该函数根据不同的子模式，按照 1.1.4 节的 3 个表格里面的指令来对模块进行配置，从而实现 9 种子模式配置。首先我们进行的 STA 模式下的配置，接着才对 AP 模式作配置。对于 TCP 客户端或者 UDP 模式，还会要求输入远端 IP 地址，此时，可以通过触摸屏输入远端 IP。

在配置好之后，进入数据收发测试，此时如果连接成功建立，我们可以通过按 KEY0 发送数据给 ID0 的设备，通过按 KEY1 发送数据给 ID1，外部设备发送过来的数据，将显示在接收数据区域。如果一直没有收到数据，程序每隔 10 秒会检查一次连接是否存在，并将连接状态显示在 LCD 上，同时 DS0 每 400ms 闪烁一次，提示程序正在运行。按 KEY_UP 按键，可以退出当前测试，回到主界面。

apsta.c 我们就介绍到这里，接下来，我们看那看那 wifista.c 里面的代码，该文件里面同样只有一个函数：atk_8266_wifista_test，该函数代码如下：

```
//ATK-ESP8266 WIFI STA 测试
//用于测试 TCP/UDP 连接
//返回值:0,正常
//      其他,错误代码
u8 netpro=0;   //网络模式
u8 atk_8266_wifista_test(void)
{
    //u8 netpro=0; //网络模式
    u8 key;
    u8 timex=0;
    u8 ipbuf[16];                   //IP 缓存
    u8 *p;
    u16 t=999;                      //加速第一次获取链接状态
    u8 res=0;
    u16 rlen=0;
    u8 constate=0;                  //连接状态
    p=mymalloc(SRAMIN,32);                                  //申请 32 字节内存
    atk_8266_send_cmd("AT+CWMODE=1","OK",50);       //设置 WIFI STA 模式
```

```
        delay_ms(1000);
        //设置连接到的 WIFI 网络名称/加密方式/密码,这几个参数需要根据您自己的路由
        //器设置进行修改!!
        sprintf((char*)p,"AT+CWJAP=\"%s\",\"%s\"",wifista_ssid,wifista_password);
        //设置无线参数:ssid,密码
        atk_8266_send_cmd(p,"WIFI GOT IP",1000);
       //连接目标路由器
PRESTA:
        netpro|=atk_8266_netpro_sel(50,30,(u8*)ATK_ESP8266_CWMODE_TBL[0]);
        //选择网络模式
        if(netpro&0X02)     //UDP
        {
                        LCD_Clear(WHITE);
                        POINT_COLOR=RED;
                        Show_Str_Mid(0,30,"ATK-ESP WIFI-STA  测试",16,240);
                        Show_Str(30,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);
                        if(atk_8266_ip_set("WIFI-STA  远端 UDP IP 设置",(u8*)ATK_ESP8266
                        _WORKMODE_TBL[netpro],(u8*)portnum,ipbuf))goto PRESTA;
                       //IP 输入
                       sprintf((char*)p,"AT+CIPSTART=\"UDP\",\"%s\",%s",ipbuf,(u8*)portnu
                       m);
                       //配置目标 UDP 服务器
                        delay_ms(200);
                        atk_8266_send_cmd("AT+CIPMUX=0","OK",20);   //单链接模式
                        delay_ms(200);
                        LCD_Clear(WHITE);
                        while(atk_8266_send_cmd(p,"OK",500));
        }
        else        //TCP
        {
             if(netpro&0X01)        //TCP Client  透传模式测试
             {
                  LCD_Clear(WHITE);
                  POINT_COLOR=RED;
                  Show_Str_Mid(0,30,"ATK-ESP WIFI-STA  测试",16,240);
                  Show_Str(30,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);
                  if(atk_8266_ip_set("WIFI-STA  远端 IP 设置",(u8*)ATK_ESP8266_WORK
                  MODE_TBL[netpro],(u8*)portnum,ipbuf))goto PRESTA;
                  //IP 输入
                  atk_8266_send_cmd("AT+CIPMUX=0","OK",20); //0：单连接，1：多连接
                  sprintf((char*)p,"AT+CIPSTART=\"TCP\",\"%s\",%s",ipbuf,(u8*)portnum);
                  //配置目标 TCP 服务器
                  while(atk_8266_send_cmd(p,"OK",200))
                  {
```

```
                    LCD_Clear(WHITE);
                    POINT_COLOR=RED;
                    Show_Str_Mid(0,40,"WK_UP:返回重选",16,240);
                    Show_Str(30,80,200,12,"ATK-ESP 连接 TCP 失败",12,0);
                    //连接失败
                    key=KEY_Scan(0);
                    if(key==4)goto PRESTA;
                }
                atk_8266_send_cmd("AT+CIPMODE=1","OK",200);
                                //传输模式为：透传
            }
            else                //TCP Server
            {
                LCD_Clear(WHITE);
                POINT_COLOR=RED;
                Show_Str_Mid(0,30,"ATK-ESP WIFI-STA 测试",16,240);
                Show_Str(30,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);
                atk_8266_send_cmd("AT+CIPMUX=1","OK",20);
                //0：单连接，1：多连接
                sprintf((char*)p,"AT+CIPSERVER=1,%s",(u8*)portnum);
                //开启 Server 模式(0，关闭；1，打开)，端口号为 portnum
                atk_8266_send_cmd(p,"OK",50);
            }
        }
            LCD_Clear(WHITE);
            POINT_COLOR=RED;
            Show_Str_Mid(0,30,"ATK-ESP WIFI-STA 测试",16,240);
            Show_Str(30,50,200,16,"正在配置 ATK-ESP 模块,请稍等...",12,0);

            LCD_Fill(30,50,239,50+12,WHITE);            //清除之前的显示
            Show_Str(30,50,200,16,"WK_UP:退出测试  KEY0:发送数据",12,0);
            LCD_Fill(30,80,239,80+12,WHITE);
            atk_8266_get_wanip(ipbuf);                  //获取当前模块的 IP
            sprintf((char*)p,"IP 地址:%s 端口:%s",ipbuf,(u8*)portnum);
            Show_Str(30,65,200,12,p,12,0);              //显示 IP 地址和端口
            Show_Str(30,80,200,12,"状态:",12,0);       //连接状态
            Show_Str(120,80,200,12,"模式:",12,0);      //连接状态
            Show_Str(30,100,200,12,"发送数据:",12,0);  //发送数据
            Show_Str(30,115,200,12,"接收数据:",12,0);  //接收数据
            atk_8266_wificonf_show(30,180,"请设置路由器无线参数
            为:",(u8*)wifista_ssid,(u8*)wifista_encryption,(u8*)wifista_password);
            POINT_COLOR=BLUE;
            Show_Str(120+30,80,200,12,(u8*)ATK_ESP8266_WORKMODE_TBL[netpr
            o],12,0);                                   //连接状态
```

```
USART3_RX_STA=0;
while(1)
{
    key=KEY_Scan(0);
    if(key==4)                                  //WK_UP 退出测试

    {
        res=0;
        atk_8266_quit_trans();                  //退出透传
        atk_8266_send_cmd("AT+CIPMODE=0","OK",20);
        //关闭透传模式
        break;
    }
    else if(key==1)                             //KEY0 发送数据
    {

        if((netpro==3)||(netpro==2))        //UDP
        {
            sprintf((char*)p,"ATK-8266%s 测
            试%02d\r\n",ATK_ESP8266_WORKMODE_TBL[netpro],t/10
            );//测试数据
            Show_Str(30+54,100,200,12,p,12,0);
            atk_8266_send_cmd("AT+CIPSEND=25","OK",200);
            //发送指定长度的数据
            delay_ms(200);
            atk_8266_send_data(p,"OK",100);   //发送指定长度的数据
            timex=100;
        }
        else if((netpro==1))    //TCP Client
        {
            atk_8266_quit_trans();
            atk_8266_send_cmd("AT+CIPSEND","OK",20);
            //开始透传
            sprintf((char*)p,"ATK-8266%s 测试%d\r\n",ATK_ESP8266_
            WORKMODE_TBL[netpro],t/10);//测试数据
            Show_Str(30+54,100,200,12,p,12,0);
            u3_printf("%s",p);
            timex=100;
        }
        else       //TCP Server
        {
            sprintf((char*)p,"ATK-8266%s 测试%02d\r\n",ATK_ESP8266
            _WORKMODE_TBL[netpro],t/10);//测试数据
            Show_Str(30+54,100,200,12,p,12,0);
```

```
                atk_8266_send_cmd("AT+CIPSEND=0,25","OK",200);
                //发送指定长度的数据
                delay_ms(200);
                atk_8266_send_data(p,"OK",100);   //发送指定长度的数据
                timex=100;
            }

        }else;

        if(timex)timex--;
        if(timex==1)LCD_Fill(30+54,100,239,112,WHITE);
        t++;
        delay_ms(10);
        if(USART3_RX_STA&0X8000)        //接收到一次数据了
        {
            rlen=USART3_RX_STA&0X7FFF;
            //得到本次接收到的数据长度
            USART3_RX_BUF[rlen]=0;        //添加结束符
            printf("%s",USART3_RX_BUF);  //发送到串口
            sprintf((char*)p,"收到%d 字节,内容如下",rlen);//接收到的字节数
            LCD_Fill(30+54,115,239,130,WHITE);
            POINT_COLOR=BRED;
            Show_Str(30+54,115,156,12,p,12,0);
            //显示接收到的数据长度
            POINT_COLOR=BLUE;
            LCD_Fill(30,130,239,319,WHITE);
            Show_Str(30,130,180,190,USART3_RX_BUF,12,0);
            //显示接收到的数据
            USART3_RX_STA=0;
            if(constate!=3)t=1000;
            //状态为还未连接,立即更新连接状态
            else t=0;
            //状态为已经连接了,10 秒后再检查
        }
        if(t==1000)//连续 10 秒钟没有收到任何数据,检查连接是不是还存在.
        {
            constate=atk_8266_consta_check();//得到连接状态
            if(constate=='+')Show_Str(30+30,80,200,12,"连接成功",12,0);
            //连接状态
            else Show_Str(30+30,80,200,12,"连接失败",12,0);
            t=0;
        }
        if((t%20)==0)LED0=!LED0;
        atk_8266_at_response(1);
```

```
            }
        myfree(SRAMIN,p);           //释放内存
        return res;
    }
```

此部分代码比 apsta.c 少了很多，主要少了 AP 模式的配置，其他部分操作与 apsta.c 操作差不多。在此模式下只有 3 种子模式（TCP 服务器，TCP 客户端和 UDP），该模式下模块需连接到指定 WIFI 热点，代码里面我们是连接到我们的路由器，如需连接到客户自己的 WIFI 热点只需修改如下代码即可。

```
//WIFI STA 模式,设置要去连接的路由器无线参数,请根据你的路由器设置,自行修改.
const u8* wifista_ssid="ALIENTEK";              //路由器 SSID 号
const u8* wifista_encryption="wpawpa2_aes";     //wpa/wpa2 aes 加密方式
const u8* wifista_password="15902020353";       //连接密码
```

这里的配置，告诉模块，我们要连接的 WIFI 热点，SSID 为：**ALIENTEK**；加密方式为：**WPA/WPA2_AES**；密码为：**15902020353**。这几个红色的参数，得根据您的路由器设置正确才行。

wifista.c 我们就介绍到这里，接下来，我们看看 wifiap.c 里面的代码，该文件里面同样只有一个函数：atk_8266_wifiap_test，该函数代码如下：

```
//ATK-ESP8266 WIFI AP 测试
//用于测试 TCP/UDP 连接
//返回值:0,正常
//其他,错误代码
u8 atk_8266_wifiap_test(void)
{
    u8 netpro=0;                //网络模式
    u8 key;
    u8 timex=0;
    u8 ipbuf[16];               //IP 缓存
    u8 *p;
    u16 t=999;                  //加速第一次获取链接状态
    u8 res=0;
    u16 rlen=0;
    u8 constate=0;                                  //连接状态
    p=mymalloc(SRAMIN,32);                          //申请 32 字节内存
    ……                                              //省略部分代码
    return res;
}
```

以上代码，我们省略了大部分与串口 STA 模式类似的代码，仅列出了关键区别代码。其中 AT+CWSAP 指令的参数：wifiap_ssid,wifiap_encryption,wifiap_password 等是在 common.c 里面定义的字符串，如下：

```
//WIFI AP 模式,模块对外的无线参数,可自行修改.
const u8* wifiap_ssid="ATK-ESP8266";            //对外 SSID 号
const u8* wifiap_encryption="wpawpa2_aes";      //wpa/wpa2 aes 加密方式
const u8* wifiap_password="12345678";           //连接密码
```

以上配置，设置模块的对外无线参数，其中 SSID 为：ATK-ESP8266；加密方式为：

WPA/WPA2_AES；密码为：12345678。

整个代码实现了对模块串口无线 AP（COM-WIFI AP）模式各个子模式的测试（TCP 服务器、TCP 客户端、UDP）。这里，该函数根据不同的子模式，按照 1.1.2 节的 3 个表格里面的指令来对模块进行配置，从而实现子模式配置。对于客户端模式，还会要求输入远端 IP 地址，此时，可以通过触摸屏输入远端 IP。

剩下的处理，同串口 STA（COM-STA）模式的处理方式。

wifista.c 我们就介绍到这里，最后再来看看 test.c，该文件里面就一个 main 函数，main 函数代码如下：

```
int main(void)
{
    u8 key,fontok=0;
    Stm32_Clock_Init(336,8,2,7);//设置时钟,168Mhz
    delay_init(168);                    //延时初始化
    uart_init(84,115200);               //初始化串口波特率为 115200
    usart3_init(42,115200);             //串口 3 初始化
    LED_Init();                         //初始化 LED
    LCD_Init();                         //LCD 初始化
    KEY_Init();                         //按键初始化
    W25QXX_Init();                      //初始化 W25Q128
    tp_dev.init();                      //初始化触摸屏
    usmart_dev.init(168);               //初始化 USMART
    my_mem_init(SRAMIN);                //初始化内部内存池
    my_mem_init(SRAMCCM);               //初始化 CCM 内存池
    exfuns_init();                      //为 fatfs 相关变量申请内存
    f_mount(fs[0],"0:",1);              //挂载 SD 卡
    f_mount(fs[1],"1:",1);              //挂载 FLASH.
    key=KEY_Scan(0);
    if(key==KEY0_PRES)                  //强制校准
    {
        LCD_Clear(WHITE);               //清屏
        TP_Adjust();                    //屏幕校准
        TP_Save_Adjdata();
        LCD_Clear(WHITE);               //清屏
    }
    fontok=font_init();                 //检查字库是否 OK
    if(fontok||key==KEY1_PRES)          //需要更新字库
    {
        LCD_Clear(WHITE);                   //清屏
        POINT_COLOR=RED;                    //设置字体为红色
        LCD_ShowString(60,50,200,16,16,"ALIENTEK STM32");
        while(SD_Init())                    //检测 SD 卡
        {
            LCD_ShowString(60,70,200,16,16,"SD Card Failed!");
            delay_ms(200);
```

```
                LCD_Fill(60,70,200+60,70+16,WHITE);
                delay_ms(200);
            }
            LCD_ShowString(60,70,200,16,16,"SD Card OK");
            LCD_ShowString(60,90,200,16,16,"Font Updating...");
            key=update_font(20,110,16,"0:");                    //从 SD 卡更新
            while(key)                                          //更新失败
            {
                LCD_ShowString(60,110,200,16,16,"Font Update Failed!");
                delay_ms(200);
                LCD_Fill(20,110,200+20,110+16,WHITE);
                delay_ms(200);
            }
            LCD_ShowString(60,110,200,16,16,"Font Update Success!");
            delay_ms(1500);
            LCD_Clear(WHITE);                                   //清屏
        }
        atk_8266_test();
    }
```

此部分代码比较简单，首先初始化了 ATK-ESP8266 模块用到的串口 3（波特率：115200）。由于本例程我们用到了触摸屏、12*12 字体、16*16 字体以及 UNICODE 与 GBK 转换码表，所以我们在 main 函数里面加入了触摸屏校准以及字库更新的代码。在启动的时候，按下 KEY0，可以进入触摸屏强制校准；在启动的时候，按下 KEY1，可以强制进行字库更新。

最后，调用 atk_8266_test 函数，初始化串口 2（波特率：115200），然后调用 atk_rm04_test 函数，进入 ATK-ESP8266 模块的主测试程序，开始对 ATK- ESP8266 的各项功能（串口无线 STA（COM-WIFI STA）、串口无线 AP（COM-WIFI AP）、串口无线 AP+STA（COM-WIFI AP+STA））进行测试。

另外，为了方便大家调试，我们在本例程的 USMART 添加了 atk_8266_send_cmd 和 atk_8266_quit_trans 这两个函数。这样，电脑就可以通过串口 1，间接控制 ATK-RM04 模块。

比如（假设硬件已经准备好），我们通过串口调试助手可以将模块默认的 WIFI 配置进行修改。通过：atk_8266_send_cmd("AT+CWSAP?","OK",20)，可以查询模块 AP 模式的 WIFI 配置，通过：atk_8266_send_cmd("AT+CIFSR","OK",20),可以查询模块的本地 IP，如图 3.2 所示：

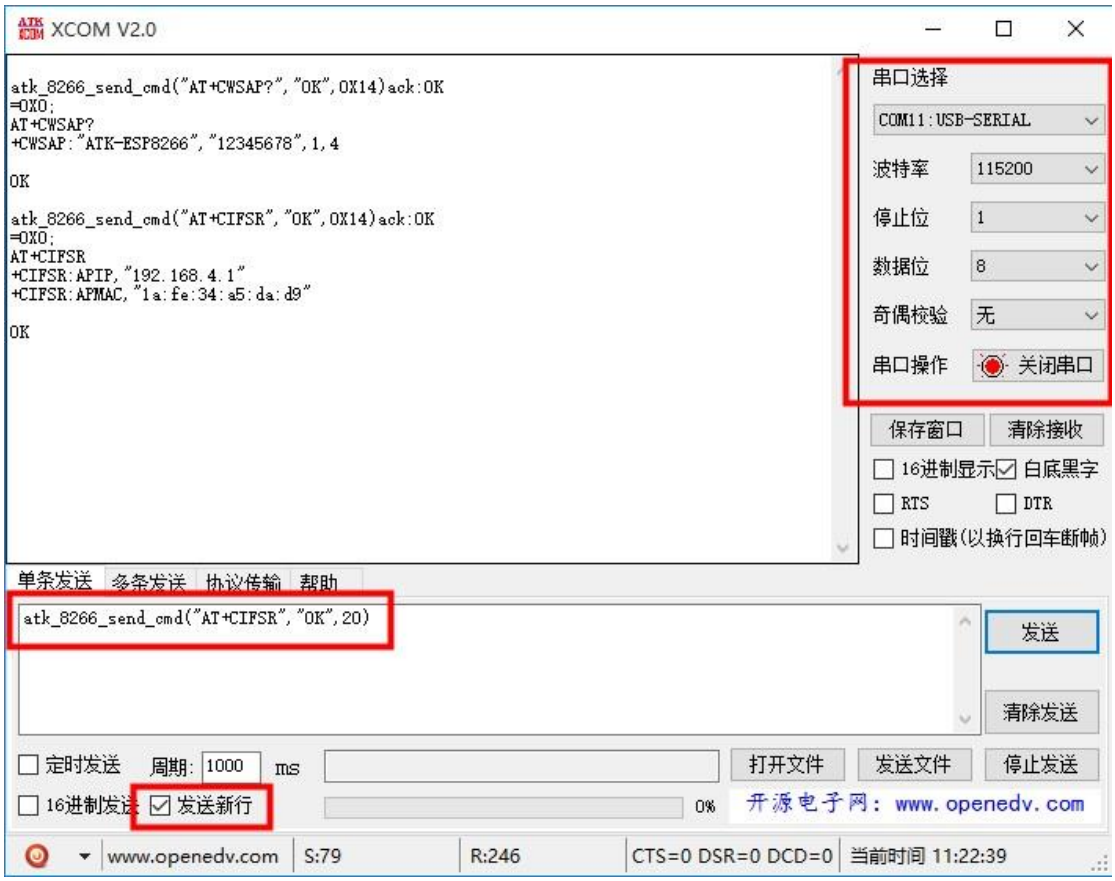

图 3.2 通过 USMART 设置 ATK-ESP8266 模块

4、验证测试

针对模块的 3 种模式，这里我们测试也分为 3 部分：串口无线 WIFI（COM-AP）测试、串口无线 STA（COM-STA）测试和串口无线 AP+STA（COM-AP+STA）测试。

模块与探索者 F407 开发板的连接方式：见图 2.3.1。

连接好之后，分别给模块和开发板供电，然后等待 10s 左右（模块初始化需要这么多时间），开发板便可以连接上模块，在 LCD 上面显示模块的一些信息。在检测到模块后，主界面如图 4.1 所示：

图 4.1 WIFI 测试主界面

在主界面下，我们按 KEY0，即可进入串口无线 AP+STA（COM-AP+STA）测试，按 KEY1，即可进入串口无线 STA（COM-STA）测试，按 KEY_UP，即可进入串口无线 AP（COM-AP）测试。

4.1 串口 STA（COM-STA）测试

串口无线 STA 模式，模块工作在 WIFI STA 状态，要求必须有一个无线热点，并开启 WIFI DHCP 功能，然后模块会通过 WIFI 连接到该热点，电脑或其他 WIFI 可以通过有线（本例，我们采用有线）/无线连接路由器，实现电脑与模块的连接。

串口无线 STA 连接方式：电脑 WIFI 设备<有线或无线>无线 WIFI 热点<WIFI 无线>ATK-ESP8266 模块（WIFI STA）。

本次测试，电脑通过网线连接无线路由器，然后模块通过 WIFI 与无线路由器连接，路由器作为中转提供模块和电脑的连接通道。

在 WIFI 测试主界面（见图 4.1）下，按 KEY1，即可进入串口无线 STA（COM- STA）工作模式选择界面，共 3 个工作模式：TCP 服务器、TCP 客户端、UDP。通过按 KEY0/KEY1 按键，选择模式，按 KEY_UP 按键确定。如图 4.1.1

图 4.1.1 工作模式选择界面

4.1.1 串口 STA TCP 服务器测试

选择 TCP 服务器，按 KEY_UP，进入 TCP 服务器测试，如图 4.1.1.1 所示：

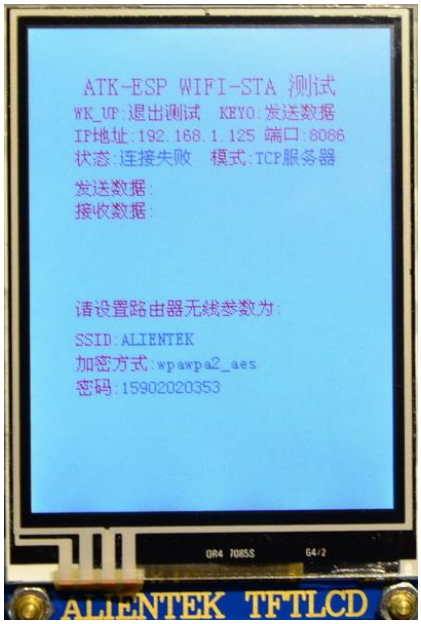

图 4.1.1.1 STA TCP 服务器测试界面

图 4.1.1.1 中，可以看到，模块的 IP 地址：192.168.1.125，端口为：8086，工作模式为 TCP 服务器，状态显示为：连接失败。外部 TCP Client 连接到 192.168.1.125，端口：8086 即可连接到本模块。

状态显示连接失败，说明 TCP 连接还没建立，因为还没有 TCP Client 连接到模块。测试我们选择同一个局域网下面的电脑作 TCP Client。我们在电脑上运行：模块资料\3，配套软件\网络调试助手.exe，然后设置协议类型为：TCP Client；服务器 IP 地址为：192.168.1.125；

服务器端口号为：8086，设置好之后，点击连接，即可连接到 ATK-ESP8266 模块，如图 4.1.1.2 所示：

图 4.1.1.2 网络调试助手设置界面

图 4.1.1.2 中，显示网络调试助手已经连接到了 ATK-ESP8266 模块，此时两者之间便可以互相发送数据了。按开发板的 KEY0 按键，即可通过模块向电脑发送数据，在网络调试助手上显示出来，如图 4.1.1.3 所示：

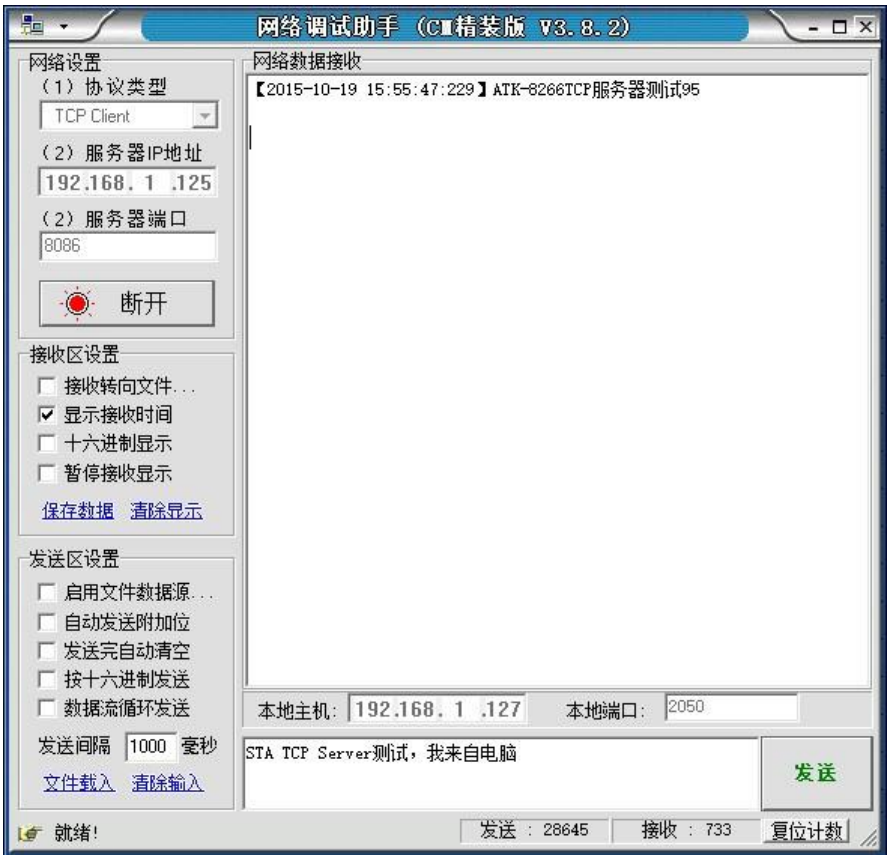

图 4.1.1.3 网络调试助手接收到来自模块 TCP 服务器的数据

在网络调试助手输入数据也可以发送给模块，如图 4.1.1.4 所示：

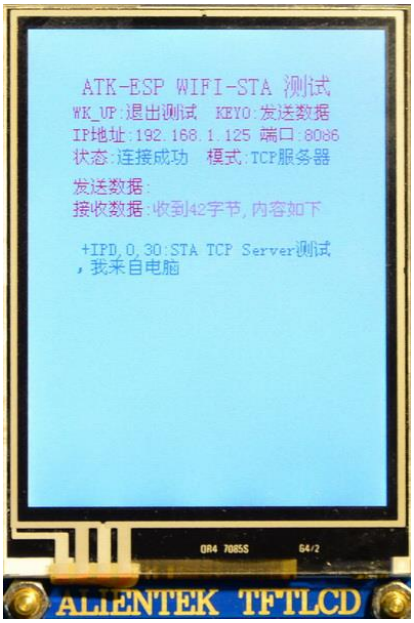

图 4.1.1.4 模块收到网络调试助手的数据

4.1.2 串口 STA TCP 客户端测试

在工作模式选择界面，选择 TCP 客户端，再按 KEY_UP 按键，即可进入 TCP 客户端测试。不过同 TCP 服务器测试不同，客户端模式，需要输入远端 IP 地址，即模块将要连接到的 IP 地址，且连接端口号固定为：8086。比如我们要连接我们的电脑，所以需要先知道电脑的 IP 地址，可以在本地连接状态→支持，查看到我们电脑的 IP 地址，如图 4.1.2.1 所以：

图 4.1.2.1 电脑 IP 地址查看

上图中，我们可以看到电脑 IP 地址为：192.168.1.127，所以，我们在远端 IP 设置里面输入：192.168.1.127，即可连接到我们的电脑。如图 4.1.2.2 所示：

图 4.1.2.2 远端 IP 设置

图 4.1.2.2 中，是我们通过屏幕虚拟键盘输入的远端 IP 地址，即我们电脑的 IP 地址。输入完成后，点击连接，开发板就会对模块进行配置，配置好之后（大概要 5s），进入 TCP 服务器连接界面，并且提示连接失败，如图 4.1.2.3 所示：

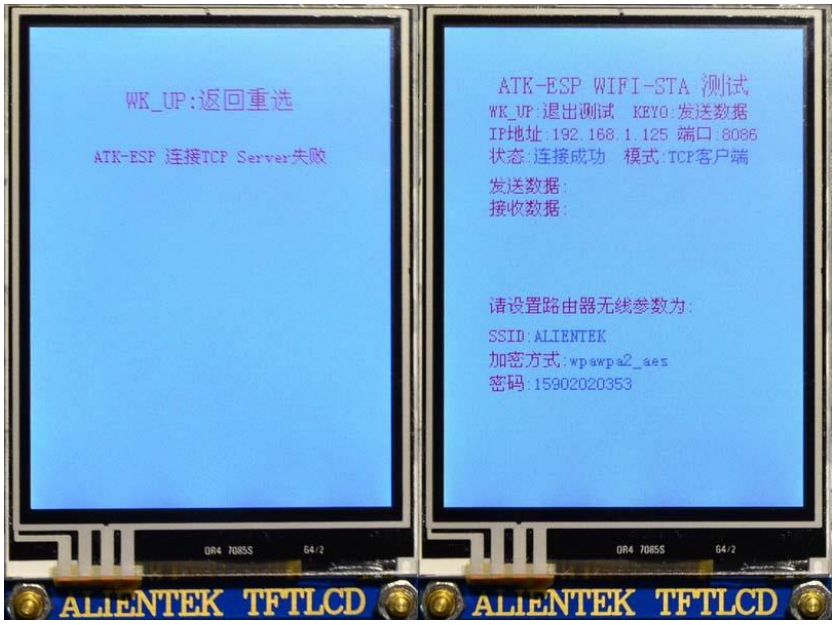

图 4.1.2.3 TCP 客户端连接失败和测试界面

上图为 TCP 连接未建立时的显示状态，要成功连接，必须打开网络调试助手，设置协议类型为：TCP Server；服务器 IP 地址为：192.168.1.127（即你电脑的 IP，得根据实际情况修改）；服务器端口为：8086；然后点击连接按钮，开启 TCP 服务，如图 4.1.2.4 所示：

图 4.1.2.4 网络调试助手设置界面

网络调试助手开启 TCP 服务后，就可以看到开发板液晶显示进入 TCP 客户端测试界面。测试界面如下图 4.1.2.5 所示，此时按开发板的 KEY0 即可向网络调试助手发送数据

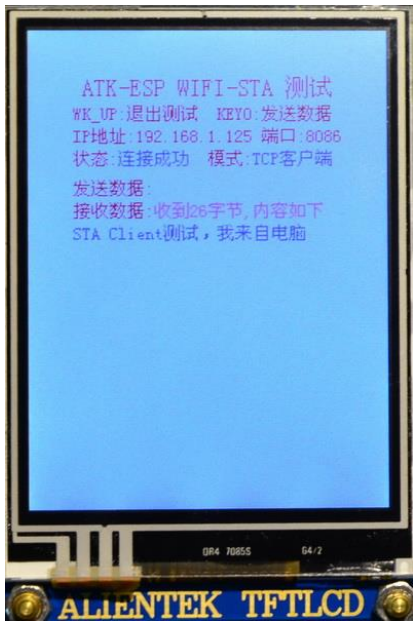

图 4.1.2.5 TCP 客户端收到数据

图 4.1.2.5 网络调试助手接收到来自模块 TCP 客户端的数据

在网络调试助手输入数据也可以发送给模块，如图 4.1.2.6 所示：

图 4.1.2.6 TCP 客户端数据的发送与接收

4.1.3 串口 STA UDP 测试

在工作模式选择界面，选择 UDP，再按 KEY_UP 按键，即可进入 UDP 测试。UDP 与 TCP 客户端测试基本一样，所以步骤我们就不详细介绍了，待配置好之后，进入 UDP 测试界面，如图 4.1.3.1 所示：

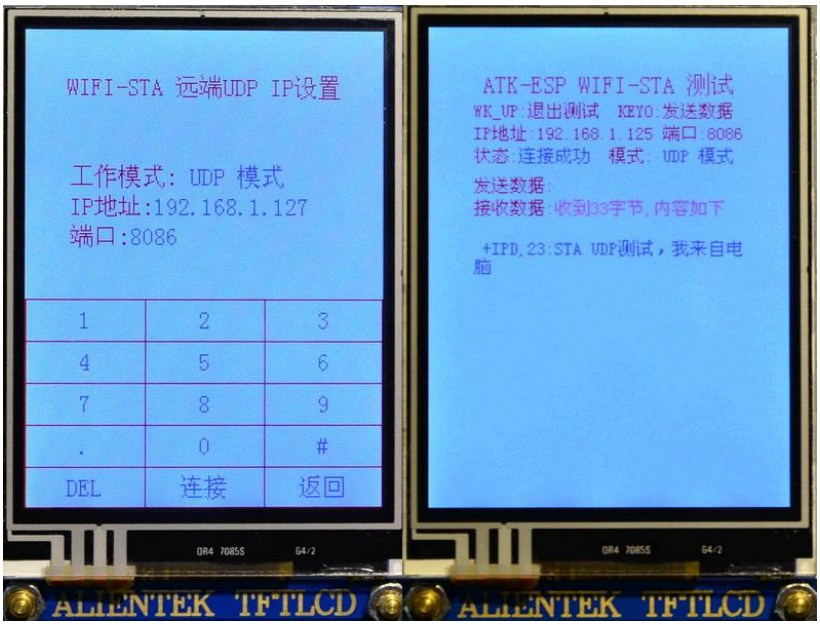

图 4.1.3.1 UDP 测试界面

4.2 串口 AP（COM-AP）测试

串口无线 AP，模块工作在 WIFI AP 状态，并开启 DHCP 功能，外部 WIFI 设备（手机、平板、电脑等），可以通过 WIFI 连接到模块，本例，我们选择带 WIFI 的 android 智能手机测试。

串口无线接入点 网络连接方式：ATK-ESP8266 模块(WIFI AP)<WIFI>智能手机(WIFI STA)。即，模块通过 WIFI 连接智能手机，模块作为 WIFI AP，智能手机做 WIFI STA。

在 WIFI 测试主界面（见图 4.1）下，按 KEY_UP，即可进入串口无线 AP（COM-AP）工作模式选择界面，共 3 个工作模式：TCP 服务器、TCP 客户端、UDP。同图 4.1.1 所示界面基本一样，操作方法也完全一样。

4.2.1 串口 AP TCP 服务器测试

选择 TCP 服务器，按 KEY_UP 按键，进入 TCP 服务器测试，此时，程序会配置模块为 WIFI AP 模式，SSID 为：ATK-ESP8266；加密方式为：wpawpa2_aes；密码为：12345678。模块 IP 地址（TCP 服务器 IP 地址）为：192.168.4.1；端口为：8086；待配置好后，进入 TCP 服务器测试界面，如图 4.2.1.1 所示：

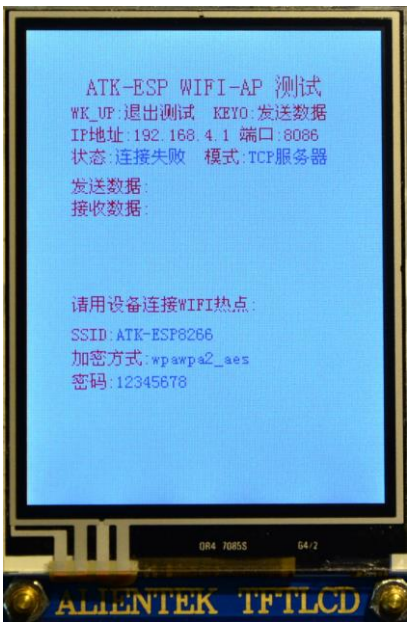

图 4.2.1.1 TCP 服务器测试界面

此时，模块的 TCP 服务器已经开启，IP 地址为：192.168.4.1，端口号：8086。但是，由于没有 TCP Client 来连接，所以状态显示：连接失败。我们先打开智能手机的 WIFI 功能，然后→设置→无线和网络→WLAN 设置，即可在 WLAN 网络里面看到有：ATK-ESP8266 的网络 SSID，然后点击该网络，输入密码：12345678，再点击连接，即可连接到我们的模块，如图 4.3.1.2 所示：

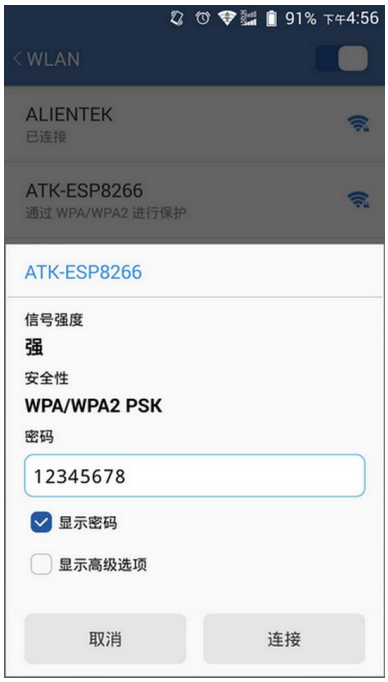

图 4.2.1.2 手机连接 ATK-ESP8266 WIFI 网络

稍等片刻，手机便会连接到 ATK-ESP8266 模块，如图 4.2.1.3 所示：

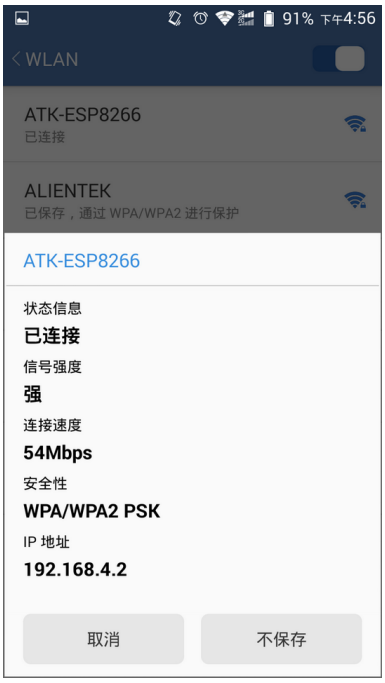

图 4.2.1.3 手机成功连接到模块

从上图可以看出，模块给手机分配的 IP 地址为：192.168.4.2。

然后，我们在手机安装：3，配套软件\手机端网络调试助手\网络调试助手（安卓手机版）.apk 这个软件，之后，在手机上运行该程序，然后依次设置：1.tcp client→2.点击增加图标→3.输入服务器 IP 和端口号→4.按增加按钮→5.连接建立，如图 4.2.1.4 所示：

图 4.2.1.4 手机网络调试助手 TCP Client 设置

经过如上设置以后，手机和模块就建立了 TCP 连接了，此时开发板液晶显示状态将会变为：连接成功。注意，图 4.2.1.4 中，如果输入了错误的 IP/端口号，右侧的图片将不会显示 port：8086，而回显示 disconnect。

在连接成功建立后，就可以互相发送数据了，如图 4.2.1.5 和 4.2.1.6 所示

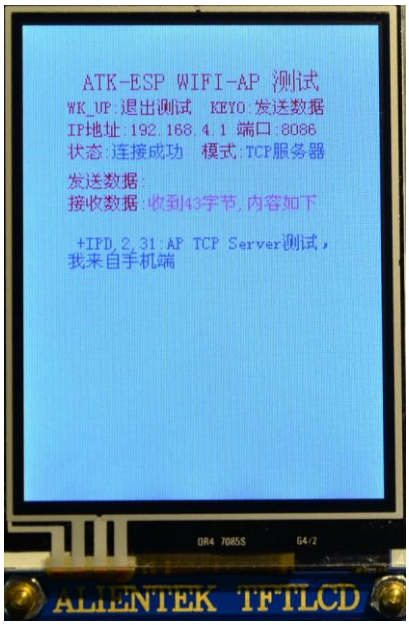

图 4.2.1.5 模块收到来自手机 TCP 客户端的数据

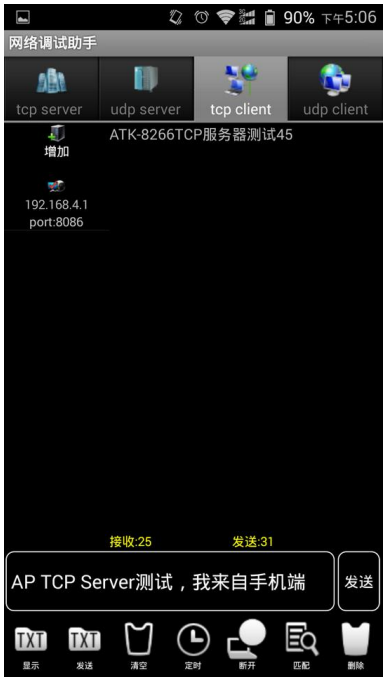

图 4.2.1.6 手机收到模块 TCP 服务器发送过来的数据

4.2.2 串口 AP TCP 客户端测试

在 WIFI AP 的工作模式选择界面，选择 TCP 客户端，再按 KEY_UP 按键，即可进入 TCP 客户端测试。此时，由于模块重启，手机可能连接到其他 WIFI 网络，所以，我们在手机上要重新选择 WLAN 连接到 ATK-ESP8266，然后打开网络调试助手，设置：1.tcp server→2. 点击配置图标 →3. 设置端口:8086，点击激活→4.连接成功，如图 4.2.2.1 所示：

图 4.2.2.1 手机网络调试助手 TCP Server 设置

这里要注意：我们要模块连接到手机建立的 TCP Server，我们需要知道手机的 IP 地址（192.168.4.2），该地址将用到本节作为 TCP 服务器端的 IP 地址。在远端 IP 设置界面，输入远端 IP 地址：192.168. 4.2，设置好 IP 之后，点击连接提交配置，配置成功后（约 5s），进入到 TCP 客户端测试界面，如图 4.2.2.1 所示：

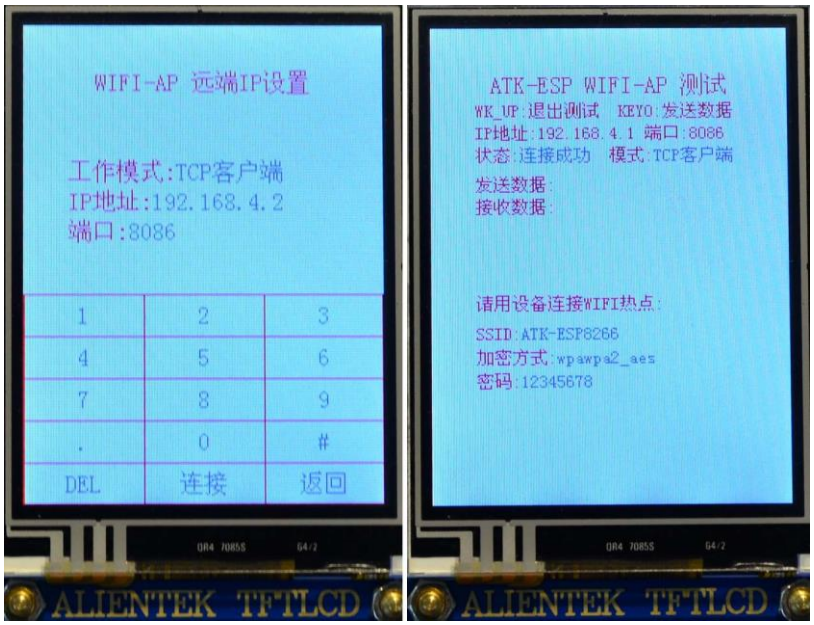

图 4.2.2.2 TCP 客户端 IP 设置及测试界面

手机端网络调试助手设置好，连接成功之后，手机即可与模块互相发送数据了，如图 4.2.2.2 右侧图片，以及图 4.2.2.3 所示：

图 4.2.2.3 模块收到来自手机 TCP 服务器的数据

4.2.3 串口 AP UDP 测试

在工作模式选择界面，选择 UDP，再按 KEY_UP 按键，即可进入 UDP 测试。UDP 与 TCP 客户端测试基本一样，所以步骤我们就不详细介绍了，待配置好之后，进入 UDP 测试界面，如图 4.1.3.1 所示：

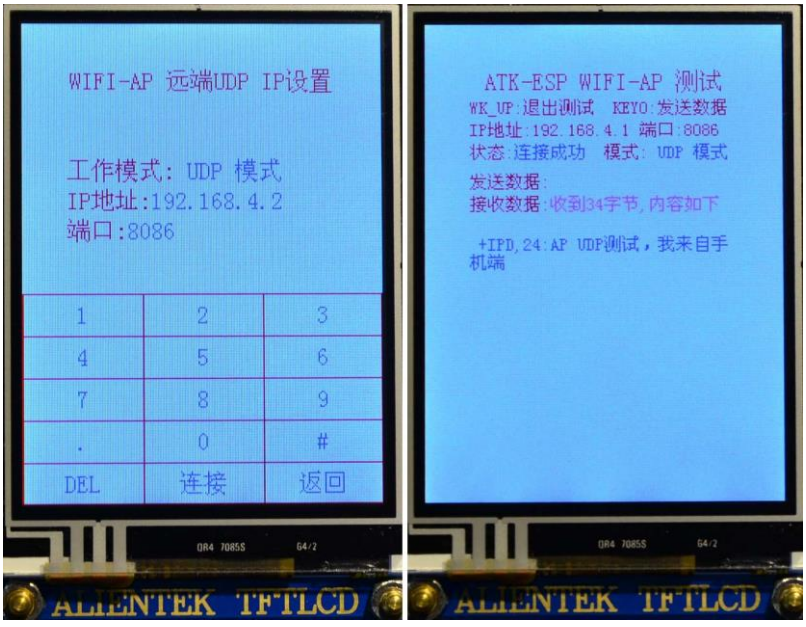

图 4.2.3.1 UDP 连接到远端 IP 及测试界面

4.3 串口 AP+STA（COM-AP+STA）测试

该模式下模块 AP 和 STA 同时开启，此状态下模块既作无线 AP（其他 WIFI 设备可以连接到 ATK_ESP8288 模块），也可作 STA 连接到其他的 wifi 热点。下面的测试中我们将介绍该工作模式，模块工作于该状态下，AP 有三种工作模式，STA 有三种工作模式，这样就一共有 9 中工作模式，基本的测试跟前面都差不多，这里我们仅介绍 3 种模式的测试。

在 WIFI 测试主界面（见图 4.1）下，按 KEY0，即可进入串口无线 AP+STA（COM-AP+STA）工作模式中的 STA 模式选择界面，共 3 个工作模式：TCP 服务器、TCP 客户端、

UDP，这里我们选择 TCP 服务器，下面的介绍文档中我们仅介绍 STA 作服务器，AP 模式下的三种工作模式介绍。如图 4.3.1

图 4.3.1 STA 模式下默认选择 TCP 服务器

4.3.1 串口 AP TCP 服务器，STA TCP 服务器测试

STA 模式下选择了 TCP 服务器，接着出现 AP 模式下的工作状态，这里我们选择 TCP 服务器。接着就是我们的串口无线 AP TCP 服务器+STA TCP 服务器测试界面。通过测试界面我们看到 AP IP：192.168.4.1；STA IP：192.168.1.xxxxx。我们通过手机链接到 SSID 为 ATK_ESP8266 的 wifi 无线接入点，并且通过网络调试助手连接到 192.168.4.1，8086，如果连接成功测试界面会显示：0,CONNECT；这说明模块分配给手机的 ID 号是 0。同样的方式通过电脑链接到 192.168.1.xxxxx，8086，若连接成功，测试界面会同样显示：1,CONNECT；模块分配给刚来你接到模块的电脑 ID 是 1。AP+STA 模式下的 ID 号分配是按照连接到模块的顺序来分配的。如下图所示：

图 4.3.1.1 AP 模式下选择 TCP 服务器

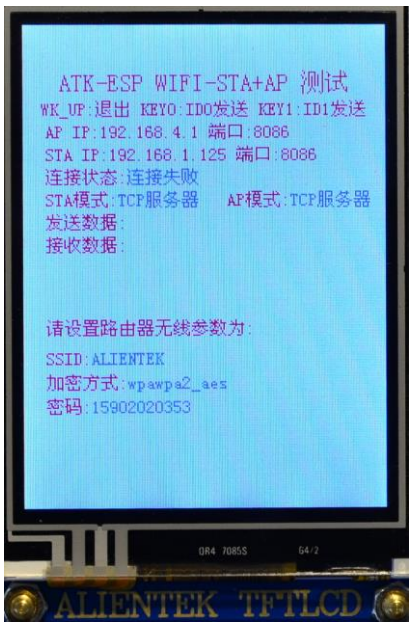

图 4.3.1.2 测试界面图片

图 4.3.1.3 手机和电脑分别连接到模块

连接成功后我们可以通过按按键 KEY0 和 KEY1 分别发送数据到手机和电脑，如图 4.3.1.4.1 和图 4.3.1.4.2：

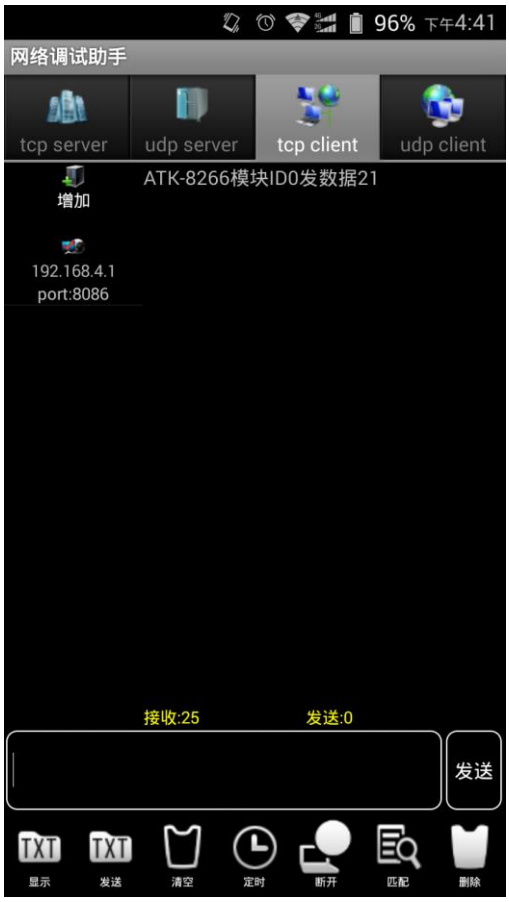

图 4.3.1.4.1 手机收到数据

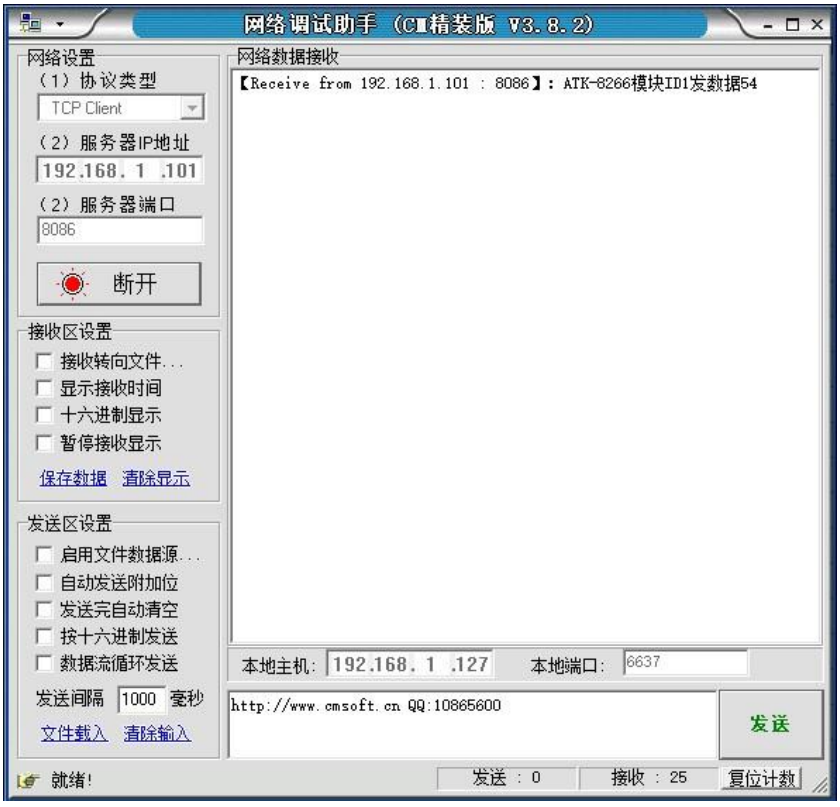

图 4.3.1.4.1 电脑收到数据

我们也可以通过手机的网络调试助手和电脑的网络调试助手发送数据到模块，并且手机端发送数据有收到‘+IPD,0’数据头的数据，这说明数据来自 ID0，电脑端发送数据有收到+IPD,1’数据头的数据，这说明数据来自 ID1。如图 4.3.1.5：

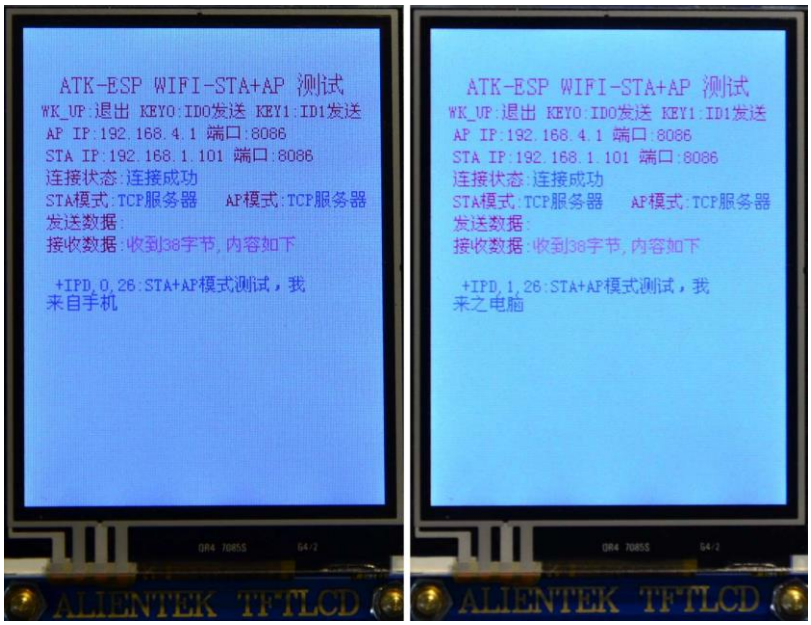

图 4.3.1.5 模块收到手机和电脑的数据.

4.3.2 串口 AP TCP 客户端，STA TCP 服务器测试

STA 模式下我们选择 TCP 服务器，接着设置 AP 模式下的工作方式，我们选择 TCP 客户端模式，该模式下需要输入 TCP Server 的 IP 地址，此时我们先在手机端连接到 SSID 为 ATK_ESP8266 的无线 wifi 热点，然后在手机上通过网络调试助手建立 TCP Server，方法同 4.2.2。我们然后在屏幕上输入手机端建立的 TCP Server IP：192.168.4.2。 如图 4.3.2.1 所示：

图 4.3.2.1 模块 AP 模式作 TCP Client 连接到 TCP Server

连接成功后就出现了测试界面，此时我们需要在电脑端连接到 STA IP：192.168.1.XXXXX；端口：8086。接下来的测试同 4.3.1，这里我们就不在说了，如下图所示：

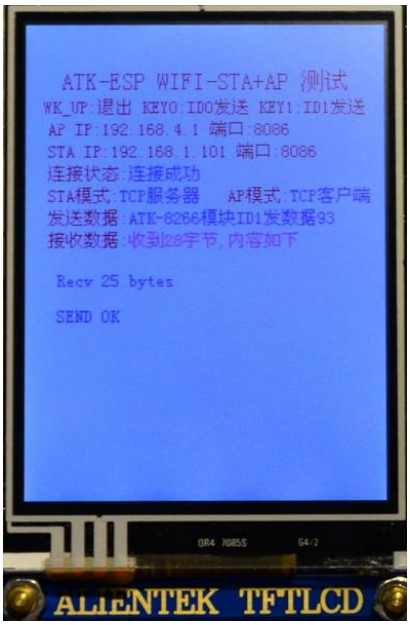

图 4.3.2.2 模块发送数据到 ID1

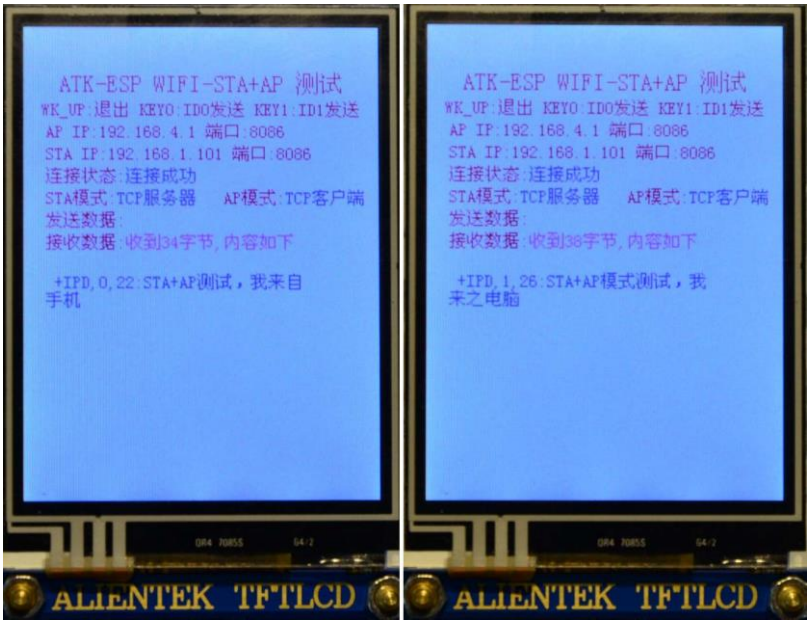

图 4.3.2.3 手机收到数据和电脑收到数据截图

4.3.3 串口 AP UDP 测试，STA TCP 服务器测试

STA 模式下我们选择 TCP 服务器，接着设置 AP 模式下的工作方式，我们选择 UDP 模式，该模式下需要输入 TCP Server 的 IP 地址，此时我们先在手机端连接到 SSID 为 ATK_ESP8266 的无线 wifi 热点，然后在手机上通过网络调试助手建立 UDP Server，方法同 4.2.2。我们然后在屏幕上输入手机端建立的 UDP Server IP：192.168.4.2。 如图 4.3.2.1 所示

图 4.3.2.1 手机端建立 UDP Server 和液晶端输入 IP

UDP 连接成功后，出现如下图 4.3.2.2 的测试界面。我们在电脑端打开网络调试助手连接到 STA IP：192.168.1.XXX、端口：8086。连接成功我们可以按 KEY0 和 KEY1 发送数据到手机端和电脑端，如图 4.3.2.3：

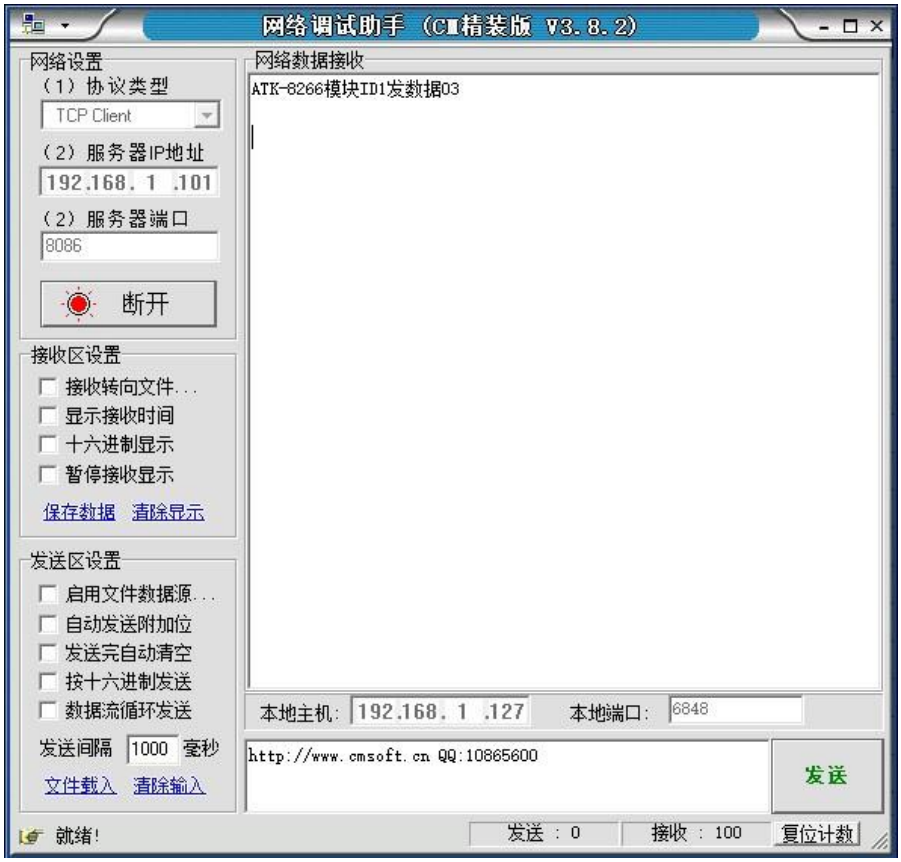

图 4.3.2.3 电脑端收到数据

我们也可以手机和电脑发送数据到模块，如图 4.3.2.4：

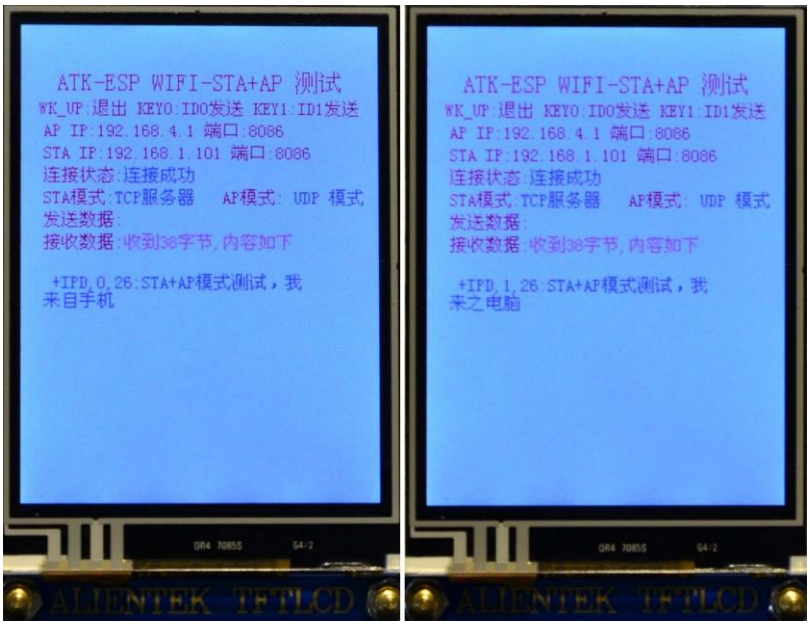

图 4.3.2.4 模块收到手机和电脑的数据

5、联系我们

官网地址：http://www.alientek.com

官方论坛：http://www.openedv.com

样品购买：https://openedv.taobao.com

商务合作：liujun6037@foxmail.com

技术支持：login-mail@foxmail.com

公司地址：广东省广州市白云区广州民营科技园弘实商务大厦 906

联系电话：020-38271790

正点原子
ZHENGDIANYUANZI