

九九物联 WiFi 方案目录

- 1) 灯控照明系统单品方案 智能灯控 App + 智能音箱语音控制
 - 2) 开关量单品方案 智能开关插座 App + 智能音箱控制
 - 3) 小家电智能方案 家用电器（电磁炉，电风扇等等）App + 智能音箱控制
 - 4) 语音功能对接方案 以智能音箱语音控制为根基，语音控制模组 IO 输出标准板
 - 5) 智能门锁对接方案 用户 ID 管理，App 远程通讯，门锁电控板透传通讯
 - 6) 电动窗帘对接方案 电机控制 App + 智能音箱控制
 - 7) 扫码支付通讯固件 多重加密通讯协议，高可靠性逻辑处理
 - 8) 网络 WiFi 打印机固件 大数据量，多任务同时执行的固件
 - 9) LED 控制卡固件 大数据量高速通讯，数据解析打包固件
 - 10) 工业仪表控制固件 多种数据回传，状态参数回显，通讯优化
 - 11) 低功耗优化固件 快速进入低功耗模式，快速唤醒，多参数保存
 - 12) SPI 高速通讯固件 高速透传应用，速率可达 3-5Mb
 - 13) 阿里飞燕平台固件 预先烧录 Alios 2.0 SDK
 - 14) 亚马逊平台通讯固件 预先烧录 Amazon 通讯协议
 - 15) WiFi+BLE4.2 mesh 共存方案 提供 Realtek 开源 SDK
 - 16) 2.4G+5.8G+BLE5.0mesh 方案 提供 Reaktek 开源 SDK
 - 17) 智能家居语音控制 Wi-Fi 蓝牙 mesh 组网方案
 - 18) WEB 网页设置透传固件 WEB 页面配置好网络参数，进行数据通讯
 - 19) 单片机驱动 Wi-Fi 开源学习固件 STM32 驱动源代码，App 测试源代码
 - 20) 智能家电固件 mqtt 对接多种服务器，提供 5V 参考
 - 21) 智能家居语音控制方案 Wi-Fi+蓝牙 mesh 组网智能家居方案
 - 22) WiFi 低成本单品方案 低成本控制模组，适用于插座，灯控方案
- 方案持续增加中.....

九九物联 SPI 通讯 AT+ 指令集 固件说明

（高速多线程篇）Ver. 1.1

固件应用说明：

应用产品：工业控制、高速数据传输、远程设备监控、高端智能设备

SPI 固件特点：

模块硬件：内置 ARM Cortex-M4 处理核心，超大存储空间，为软件开发实现各种各样多功能的指令，提供了稳定基础。

硬件接口：通过高速 SPI 接口进行指令的交互，数据采集，状态查询和上报等。

软件解析：Wi-Fi 模块内置可靠的指令解析处理方法，对指令的收发采用了队列的方式，可高达缓冲大约 1500 个字节大小指令空间。

模块相应：指令响应速度快，执行后回显准确清晰。

链路稳定：TCP 支持 Keepalive 功能，数据传输稳定，反复建、断链测试可靠

数据安全：支持 HTTP、HTTPS 功能指令。

状态查询：多环境下状态查询指令，智能扫描等等

多连接：支持 8 个 TCP/UDP 连接，每个连接数据传输不交叉，当前活动的连接数据传输完成断开后开始传输下一个连接的数据。



样品模组名称：AFW121TI-IND1、AFW121TO-IND1

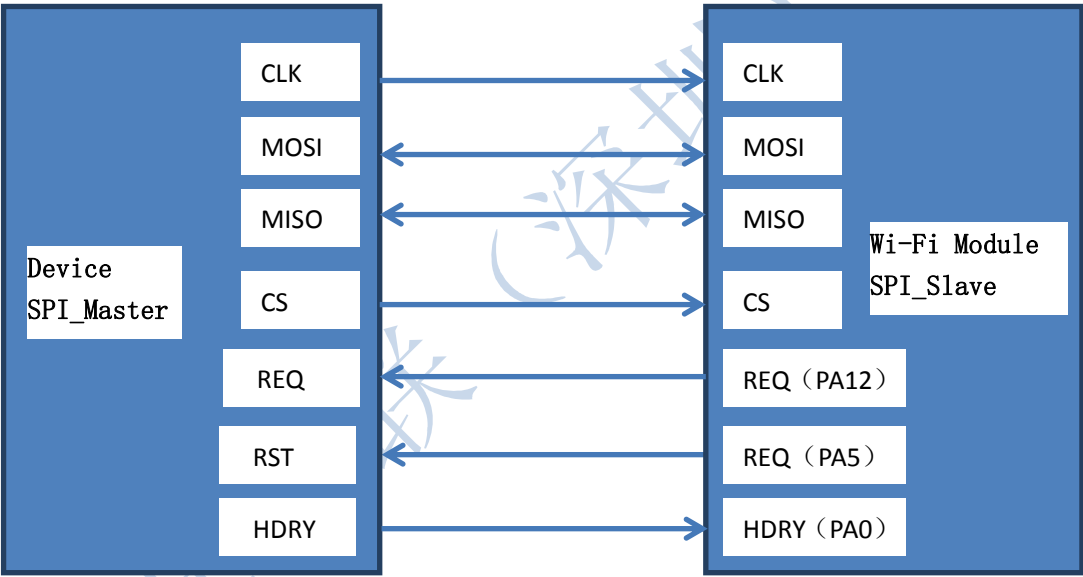
| 模块名称 | 模块说明 |
|---------------|-----------|
| AFW121TI-SPI1 | PCB 板载天线 |
| AFW121TO-SPI1 | IPEX 外接天线 |

关键词：SPI 稳定软件 低功耗 小体积 高速 贴片式 WIFI 模块方案

SPI 通信协议：

1. SPI 通信接口资源

MCU：MOSI、MISO、CLK、CS、REQ、RST、HDRV
Wi-Fi 模块：MOSI、MISO、CLK、CS、REQ、RST、HDRV



2. SPI 通信接口资源

①Master 发送 AT 指令：Master 主端控制 CS 为低，告知模块准备接收数据，然后 Master 主端检测管脚 RST 的电平：

- RST 的电平为低，Master 主端发送指令给 WIFI 模块
- RST 的电平为高，Master 主端不能发送指令（模块执行指令）
- 如果在 Master 主动发送数据的时候，数据只发了一部分，这时 REQ 为电平为低，则优先处理模块来的网络数据

②模块接收到主端 Master 的指令后，模块校验 CRC，如果校验通过就执行指令

③WIFI 回传状态或者发数据：当 WIFI 模块检测到 CS 为低电平，WIFI 模块进入读数据状态，接收完 Master 主端的指令执行完控制 REQ 为低，等待 Master 主端状态或者数据读走。

④初始状态，模块输出 RST 为低，输出 REQ 为高

⑤HDIRY 引脚为硬件流控功能，由于 SPI 通信速率较高，当 MCU 端数据处理速度跟不上或数据缓冲 buffer 不够大时拉高 HDIRY 引脚，模块检测到引脚被拉高时阻塞 TCP 数据，等待引脚再次被拉低继续接收数据。

3. SPI 通信接口资源

①通信格式：

WIFI 模块发送到主板数据包格式：

包头（2 字节）+ 有效数据长度 L（2 字节）+ 数据（1536 字节）+ crc 校验（2 字节）

包头：0x55AA

主板指令数据到 WIFI 模块数据包格式：

包头（2 字节）+ 有效数据长度 L（2 字节）+ 数据（1536 字节）+ crc 校验（2 字节）

有效数据不够 1536 字节时填充无效数据保证每包数据 1536 字节。

②解析

有效数据长度 L：

后面的数据中，有效的数据长度。

数据：

数据固定 1536 字节，其中有效的字节长度由有效数据长度 L 指定。

crc 校验（主要为了保证 EMC，静电等情况下数据正常）：

计算出 端口号 + 有效数据长度 L + 数据 的 crc。

③通信过程

Master 按照上面的数据格式封装数据包，发送给模块，模块开始接收数据。

模块接收到 Master 的数据后，校验数据，校验通过后执行指令，并回显指令执行结果。

如果数据校验失败不执行指令，直接返回指令错误“COMMAND ERROR\r\n”，WIFI 模块不再回复 ACK 包，直接回复指令执行结果。

④CRC 校验函数



```
static unsigned short crctab[256] =
```

```
{
```

```
    0x0000, 0x8005, 0x800f, 0x000a, 0x801b, 0x001e, 0x0014, 0x8011,  
    0x8033, 0x0036, 0x003c, 0x8039, 0x0028, 0x802d, 0x8027, 0x0022,  
    0x8063, 0x0066, 0x006c, 0x8069, 0x0078, 0x807d, 0x8077, 0x0072,  
    0x0050, 0x8055, 0x805f, 0x005a, 0x804b, 0x004e, 0x0044, 0x8041,  
    0x80c3, 0x00c6, 0x00cc, 0x80c9, 0x00d8, 0x80dd, 0x80d7, 0x00d2,  
    0x00f0, 0x80f5, 0x80ff, 0x00fa, 0x80eb, 0x00ee, 0x00e4, 0x80e1,  
    0x00a0, 0x80a5, 0x80af, 0x00aa, 0x80bb, 0x00be, 0x00b4, 0x80b1,  
    0x8093, 0x0096, 0x009c, 0x8099, 0x0088, 0x808d, 0x8087, 0x0082,  
    0x8183, 0x0186, 0x018c, 0x8189, 0x0198, 0x819d, 0x8197, 0x0192,  
    0x01b0, 0x81b5, 0x81bf, 0x01ba, 0x81ab, 0x01ae, 0x01a4, 0x81a1,  
    0x01e0, 0x81e5, 0x81ef, 0x01ea, 0x81fb, 0x01fe, 0x01f4, 0x81f1,  
    0x81d3, 0x01d6, 0x01dc, 0x81d9, 0x01c8, 0x81cd, 0x81c7, 0x01c2,  
    0x0140, 0x8145, 0x814f, 0x014a, 0x815b, 0x015e, 0x0154, 0x8151,  
    0x8173, 0x0176, 0x017c, 0x8179, 0x0168, 0x816d, 0x8167, 0x0162,  
    0x8123, 0x0126, 0x012c, 0x8129, 0x0138, 0x813d, 0x8137, 0x0132,  
    0x0110, 0x8115, 0x811f, 0x011a, 0x810b, 0x010e, 0x0104, 0x8101,  
    0x8303, 0x0306, 0x030c, 0x8309, 0x0318, 0x831d, 0x8317, 0x0312,  
    0x0330, 0x8335, 0x833f, 0x033a, 0x832b, 0x032e, 0x0324, 0x8321,  
    0x0360, 0x8365, 0x836f, 0x036a, 0x837b, 0x037e, 0x0374, 0x8371,  
    0x8353, 0x0356, 0x035c, 0x8359, 0x0348, 0x834d, 0x8347, 0x0342,  
    0x03c0, 0x83c5, 0x83cf, 0x03ca, 0x83db, 0x03de, 0x03d4, 0x83d1,  
    0x83f3, 0x03f6, 0x03fc, 0x83f9, 0x03e8, 0x83ed, 0x83e7, 0x03e2,  
    0x83a3, 0x03a6, 0x03ac, 0x83a9, 0x03b8, 0x83bd, 0x83b7, 0x03b2,  
    0x0390, 0x8395, 0x839f, 0x039a, 0x838b, 0x038e, 0x0384, 0x8381,  
    0x0280, 0x8285, 0x828f, 0x028a, 0x829b, 0x029e, 0x0294, 0x8291,  
    0x82b3, 0x02b6, 0x02bc, 0x82b9, 0x02a8, 0x82ad, 0x82a7, 0x02a2,  
    0x82e3, 0x02e6, 0x02ec, 0x82e9, 0x02f8, 0x82fd, 0x82f7, 0x02f2,
```



```
0x02d0, 0x82d5, 0x82df, 0x02da, 0x82cb, 0x02ce, 0x02c4, 0x82c1,
0x8243, 0x0246, 0x024c, 0x8249, 0x0258, 0x825d, 0x8257, 0x0252,
0x0270, 0x8275, 0x827f, 0x027a, 0x826b, 0x026e, 0x0264, 0x8261,
0x0220, 0x8225, 0x822f, 0x022a, 0x823b, 0x023e, 0x0234, 0x8231,
0x8213, 0x0216, 0x021c, 0x8219, 0x0208, 0x820d, 0x8207, 0x0202
};

unsigned short CrcCalc(unsigned char * buf, unsigned short len)
{
    unsigned short crc;
    unsigned char comb_val;
    crc = 0;
    while (len--)
    {
        comb_val = (unsigned char)(crc >> 8) ^ *buf++;
        crc = (unsigned short)(crc << 8) ^ crctab[comb_val];
    }
    return crc;
}
```

基本指令：

| | | | |
|-------|------------------|-----------|----------------|
| 重启模块 | AT+RST | 版本查询 | AT+SVER |
| 串口波特率 | AT+UARTBAUD | 流控功能 | AT+UARTFLOW |
| 设置重连 | AT+WLFASTCONNECT | 设置重连次数、间隔 | AT+WLFASTPARAM |

Wi-Fi 控制组合指令：

| | | | |
|----------------|---|----------------------|---|
| 指令配网 | AT+WLMODE AT+WLSTAPARAM | SoftAP mode | AT+WLMODE AT+WLAPPARAM |
| 设置 STA 模式下静态参数 | AT+WLMODE=1 AT+WLDHCP=OFF AT+WLSTATICPARAM AT+WLSTAPARAM（设置生效） | 设置 AP 模式下 DHCP 和网关规则 | AT+WLMODE=2 AT+WLAPDHCP=OFF AT+WLAPDHCPGATE AT+WLAPPARAM（设置生效） |

TCP/UDP 建链

| | | | |
|----------|---------------|----------|---------------|
| DNS 查询 | AT+NWKDNS | 建立 TCP | AT+NWKCTCP |
| 建立 UDP | AT+NWKCUDP | TPC 发送数据 | AT+NWKTCPSEND |
| UDP 发送数据 | AT+NWKUDPSEND | 进入数据透传模式 | AT+NWTTCIP |

网络状态查询：

| | | | |
|---------------|---------------|-----------|-----------|
| IP 查询 | AT+WLIP | 模块 MAC 地址 | AT+WLMAC |
| 扫描 AP 信息 | AT+WLSCAN | 信号强度 | AT+WLRSSI |
| 获取连接路由 MAC 地址 | AT+WLGETAPMAC | | |

固件适配模组：

| 型号 | AFW121T | AFW125TO | AFW126TO | AFW127PI | AFW121TI5V |
|----------------------|---|---|---|---|---|
| 九九物联 艾拉云 适配WiFi模组 |  |  |  |  |  |
| 硬件参数 | | | | | |
| Voltage | 3.0-3.6V | 3.0-3.6V | 3.0-3.6V | 3.0-3.6V | 4.5-5.5V |
| Working Current | AP mode: 125mA STA mode: 70mA sleep mode: 4mA | AP mode: 125mA STA mode: 70mA sleep mode: 4mA | AP mode: 125mA STA mode: 70mA sleep mode: 4mA | AP mode: 125mA STA mode: 70mA sleep mode: 4mA | AP mode: 125mA STA mode: 70mA sleep mode: 4mA |
| Dimension | 贴片式: (L*W*H) 18x22x3.1mm | 贴片式: (L*W*H) 18x16.5x3.1mm | 贴片式: (L*W*H) 13x13x3.1mm | 金手指: (L*W*H) 15.7x18x3.1mm | 贴片式: (L*W*H) 23.5x33.5x4mm |
| Temperature | -20~85℃ | -20~85℃ | -20~85℃ | -20~85℃ | -20~85℃ |
| Core/cores | ARM Cortex-M4F 62.5MHz | ARM Cortex-M4F 62.5MHz | ARM Cortex-M4F 62.5MHz | ARM Cortex-M4F 62.5MHz | ARM Cortex-M4F 62.5MHz |
| RAM | 256KB | 256KB | 256KB | 256KB | 256KB |
| Flash | 2MB | 2MB | 2MB | 2MB | 2MB |

模组性能

- 升级：支持 OTA 无线升级
- 低功耗模式：支持低功耗模式，2ms 之内唤醒、连接并传递数据包
- Socket：8 条 TCP 链接
- 配网：支持 SimpleConfig 快连、Airkiss 微信配网、AP 热点配网、WEB 配网
- APP demo：提供快连+softap 配网 app “99link”（提供 app-SDK 源代码）
- PCB 控制：工业级应用设计，多次版本升级，性能一致性保证
- FCC/CE 认证，符合 RoHS 标准
- 数据通讯：实测发送大数据不丢字节

专利权说明

©2017 九九物联（深圳）有限公司对于此文件保留所有权利。本文档的任何部分不得转载，不得存储在任何检索系统，或以任何未经九九物联（深圳）有限公司书面统一的形式传送

目录

| | |
|---|----|
| 概要..... | 11 |
| 指令可分为: | 12 |
| 注意..... | 12 |
| 1 AT+指令集简介..... | 13 |
| 1.1 AT+指令格式..... | 13 |
| 1.2 AT+指令集列表..... | 13 |
| 2 AT+指令集详细参数说明..... | 16 |
| 2.1 基础指令..... | 16 |
| 2.1.1 AT+S? 查询 AT+指令列表..... | 16 |
| 2.1.2 AT+RST 重启模块..... | 16 |
| 2.1.3 AT+CGPIO=R/W, PA_5, 0/1, PIN_OUTPUT/PIN_INPUT, PullNone 设置 GPIO 管脚输入/输出模式 17 | 16 |
| 2.1.4 AT+SVER 查询固件版本..... | 18 |
| 2.1.5 AT+SFTY 恢复出厂参数设置..... | 18 |
| 2.1.6 AT+SWEB=<MODE>, <WLAN_MODE> 启动/停止 Web 服务器..... | 19 |
| 2.1.7 AT+WTICKPS=<MODE> 设置 Tickless 睡眠模式..... | 20 |
| 2.1.8 AT+WLSLEEP=<MODE>, <WAKE_ENENT>, <WAKE_PIN>, <SLEEP_TIME>, <WAKE_PIN_ACTIVE> 设置 低功耗模式..... | 20 |
| 2.1.9 AT+WLSC 切换模块程序运行区域..... | 22 |
| 2.1.10 AT+WLSETCHANNEL=<MODE> 设置 Channel Plan..... | 22 |
| 2.1.11 AT+WLFASCTCONNECT=<MODE> 设置模块是否开启重连 Wi-Fi..... | 23 |
| 2.1.12 AT+WLFASCPARAM=< Retry_Count>, <Idle_Time> 设置模块重连的次数与重连间隔时间..... | 24 |
| 2.1.13 AT+WLAUTOCONNECT=<MODE>..... | 25 |
| 2.2 Wi-Fi 控制指令..... | 26 |
| 2.2.1 AT+WLMODE=<MODE> 设置无线工作模式..... | 26 |
| 2.2.2 AT+WLDAPDHCPGATE=<START_IP>, <END_IP>, <GATEWAY> 设置 AP 模式下 DHCP 和网关..... | 26 |
| 2.2.3 AT+WLDAPDHCP=<MODE> 设置 AP 模式下是否开启 DHCP..... | 27 |
| 2.2.4 AT+WLDAPPARAM=<SSID>, <SEC>, <PSW> 设置 AP 模式参数..... | 28 |
| 2.2.5 AT+WLSTADHCP=<CS> 设置 STA 模式下是否开启 DHCP..... | 29 |
| 2.2.6 AT+WLSTATICPARAM=<IP>, <MASK>, <GATE> 设置 STA 静态参数..... | 29 |
| 2.2.7 AT+WLSTAPARAM=<SSID>, [PSW] 设置 STA 模式参数..... | 30 |
| 2.2.8 AT+WLSETUP 启动 Wi-Fi 连接..... | 31 |
| 2.2.9 AT+WLCLOSE 断开 Wi-Fi 连接..... | 31 |
| 2.2.10 AT+WLPMAC=<MAC> 设置 MAC 地址..... | 32 |
| 2.2.11 AT+WLSIMPLECONFIG 启动快速配网..... | 33 |
| 2.2.12 AT+WLDAPCONFIG 连接模块热点配网..... | 34 |
| 2.2.13 AT+AIRKISS 微信快速配网..... | 35 |
| 2.2.14 AT+ NWTTCIP=<LOCALPORT>, [REMOTEIP], [REMOTEPORT] 进入数据透传模式..... | 36 |



| | | |
|--------|--|----|
| 2.3 | TCP/IP 指令..... | 37 |
| 2.3.1 | AT+NWKCTCP=<ROLE>, <LOCALPORT>, [REMOTEIP], [REMOTEPORT] 创建 TCP 连接..... | 37 |
| 2.3.2 | AT+NWKUDP=<TYPE>, [REMOTEIP], [REMOTEPORT], <LOCALPORT> 创建 UDP CLIENT 端..... | 38 |
| 2.3.3 | AT+NWKTTCP=<LOCALPORT>, <REMOTEIP>, <REMOTEPORT> 开启 TCP CLIENT 透传模式..... | 39 |
| 2.3.4 | AT+NWKTCPSSEND=<ID>, <LEN>, <DATA> 发送 TCP 数据..... | 40 |
| 2.3.5 | AT+NWKUDPSSEND=<ID>, <REMOTEIP>, <REMOTEPORT><LEN>, <DATA> 发送 UDP 数据..... | 41 |
| 2.3.6 | AT+NWKCLOSE=<ID> 关闭网络连接..... | 41 |
| 2.3.7 | AT+NWKDNS=<HOST> DNS 解析..... | 42 |
| 2.3.8 | AT+PING=<HOST> PING 指令..... | 43 |
| 2.3.9 | AT+WLHTTPGET=<host>, <port>, <resource> HTTP 请求 (GET)..... | 43 |
| 2.3.10 | AT+WLHTTPPOST=<host>, <port>, <resource>, <type>, <data_len>, <data> HTTP 请求 (POST) | |
| 2.3.11 | AT+WLHTTPS=<Request_LEN>, <Request_PORT>, <Request_Domain>, <Request_Command> HTTPS 请求..... | 46 |
| 2.3.12 | AT+WLOTA=<Server_IP>, <PORT> 空中升级..... | 47 |
| 2.3.13 | AT+WLHTTPOTA=<IP>, <PORT>, <PATH> 通过 HTTP 进行升级..... | 48 |
| 2.3.14 | AT+XMODEOTA 通过指令串口进行空中升级..... | 49 |
| 2.4 | 网络查询指令..... | 49 |
| 2.4.1 | AT+WLIP 查询自身 IP..... | 49 |
| 2.4.2 | AT+WLMASK 查询自身子网掩码..... | 50 |
| 2.4.3 | AT+WLGATE 查询自身网关..... | 50 |
| 2.4.4 | AT+WLMAC 查询自身 MAC 地址..... | 51 |
| 2.4.5 | AT+WLSCAN 扫描附近所有 AP 信息..... | 51 |
| 2.4.6 | AT+WLGETINFO 获取当前 Wi-Fi 的 IP、掩码、网关..... | 52 |
| 2.4.7 | AT+WLGETAPINFO 获取当前连接 AP 的 SSID、加密方式、密码..... | 53 |
| 2.4.8 | AT+WLGETAPMAC 获取当前连接 AP 的 MAC 地址..... | 54 |
| 2.4.9 | AT+WLGETOTAINDEX 查询当前执行的固件所在区域..... | 54 |
| 2.4.10 | AT+WLRSSI STA 模式下, 连接 AP 后, 查询 AP 的信号强度..... | 55 |
| 2.5 | 提示指令..... | 55 |
| 2.5.1 | [AUTOCON]SUCCESS Wi-Fi 连接 AP 成功..... | 55 |
| 2.5.2 | [WLDIS]WLDISCON Wi-Fi 断开连接 AP..... | 55 |
| 2.5.3 | RSP: [ID], [TPYE] [REMOTEIP], [REMOTEPORT], [LEN], [DATA] 接收网络数据..... | 55 |
| 2.5.4 | RSP: CLOSED<ID> 网络连接被关闭..... | 56 |
| 3 | 硬件基础..... | 57 |
| 3.1 | 硬件板图..... | 57 |
| 3.2 | 硬件介绍说明..... | 57 |
| 4 | 硬件连接..... | 58 |
| 5 | 固件烧录..... | 58 |
| 5.1 | AFW121Tx-Broad 烧录说明..... | 58 |
| 5.2 | 单个模块烧录..... | 59 |
| 6 | 通信相关软件及默认参数..... | 59 |



| | | |
|-------|-------------------------|----|
| 6.1 | 指令发送、监听工具..... | 59 |
| 6.2 | 网络助手..... | 59 |
| 7 | AT+Command 使用实例..... | 60 |
| 7.1 | Wi-Fi 配网、建网相关实例..... | 60 |
| 7.1.1 | STATION 模式，联网及相关设置..... | 60 |
| 7.1.2 | AP 模式，建网及相关设置..... | 60 |
| 7.1.3 | STATION+SoftAP 模式..... | 61 |
| 7.2 | TCP Server 传输..... | 62 |
| 7.2.1 | SoftAP 模式..... | 62 |
| 7.2.2 | STATION 模式..... | 64 |
| 7.3 | TCP Client 传输..... | 66 |
| 7.3.1 | SoftAP 模式..... | 66 |
| 7.3.2 | STATION 模式..... | 68 |
| 7.4 | UDP 传输..... | 70 |
| 7.4.1 | SoftAP 模式..... | 70 |
| 7.4.2 | STATION 模式..... | 72 |
| | 历史版本更新说明..... | 74 |
| 8 | 购买与支持..... | 75 |

本文档不仅详细地讲解每个 AT+指令的格式、参数、使用例子，而且作为一个指引说明，让用户在使用 AT+指令集前，准备相应的工具以及调试软件。

指令可分为：

基础指令，Wi-Fi 控制指令，网络查询指令，网络数据传输指令（TCP/UDP 指令）、提示指令

| 指令 | 说明 |
|------------|---|
| 基础指令 | 复位模块、设置串口波特率、GPIO 控制、设置省电模式等 |
| Wi-Fi 控制指令 | 设置 AP 模式参数、设置 STA 模式参数模式、DNS 解析、启动 Wi-Fi 等 |
| 网络查询指令 | 自身 IP 查询，自身网关查询，扫描可用 AP，获取当前 Wi-Fi 信息等 |
| 网络数据传输指令 | 创建 TCP，创建 UDP，TCP/UDP 数据收发，关闭网络连接、OTA 升级、下载文件，获取文件内容等 |
| 提示指令 | 提示 Wi-Fi 已经连接、提示 Wi-Fi 断开连接、接受数据回显等 |

注意

- 1、 波特率：115200 bps（默认，可修改，最大可达 921600 bps）
- 2、 AT 指令都是用大写的格式，以回车换行符结尾“\r\n” <CR><LF>
- 3、 每个指令在执行后都会有回显信息，且回显信息带有相应指令指示。

例如：[WLSETUP]OK 或者[WLSTAPARAM]ERROR

- 4、 模块准备好后回显 AT COMMAND READY，回显时间与是否设置重连有关。

1 AT+指令集简介

1.1 AT+指令格式

无参数：AT+<CMD><CR><LF>

有参数：AT+<CMD>=<...><CR><LF>

主控主动：指令回显[CMD]“OK” or “ERROR”

主控被动：接收指令“RSP: ...”

<>内为必填参数，[]内为可选参数，<CR>为换行，<LF>为回车

以上为大部分指令格式，其余部分指令需具体参照指令集列表

1.2 AT+指令集列表

| AT 指令列表 | | |
|------------|---------------------|--|
| 状态 | 描述 | 指令 |
| 基础指令 | | |
| OK | 查询 AT+ Command list | AT+S? |
| OK | 重启模块 | AT+RST |
| OK | GPIO 操作 | AT+CGPIO=R/W, PA_5, 0/1 , PIN_OUTPUT/PIN_INPUT, Pull None |
| OK | 查询 AT+软件版本信息 | AT+SVER |
| OK（串口版本） | 启动 Web 服务器 | AT+SWEB=<MODE>, <WLAN_MODE> |
| OK | 进入 Tickless 睡眠模式 | AT+WLTICKPS=<MODE> |
| OK | 设置省电电源模式 | AT+WLSLEEP=<MODE>, <WAKE_ENENT>, < WAKE_PIN>, <SLEEP_TIME>, <WAK E_PIN_ACTIVE> |
| OK | OTA 区域切换 | AT+WLSC=<MODE> |
| OK | 设置 Channal Plan | AT+WLSETCHANNEL=<MODE> |
| OK（串口版本） | 设置是否开启重连 Wi-Fi | AT+WLSETFASTCONNECT=<MODE> |
| OK（串口版本） | 设置重连 Wi-Fi | AT+WLFASTCONNECT=<MODE> |
| OK | 设置重连 Wi-Fi 参数 | AT+WLFASTPARAM=<Retry_Count>, <Idle _Time> |
| OK（SPI 版本） | 设置重连 Wi-Fi | AT+WLAUTOCONNECT=<MODE> |
| Wi-Fi 控制指令 | | |
| OK | 设置 Wi-Fi 启动模式 | AT+WLMODE=<MODE> |

| | | |
|-----------|----------------------|---|
| OK | 设置 AP 模式下 DHCP 规则和网关 | AT+WLDAPHCPGATE=<START_IP>, <END_IP>, <GATEWAY> |
| OK | 设置 AP 模式下是否开启 DHCP | AT+WLDAPHCP=<MODE> |
| OK | 设置 AP 模式参数 | AT+WLAPPARAM=<SSID>, <SEC>, <PSW> 备注：第二个参数加密方式为“OPEN”，则不用填第三个参数 |
| OK | 设置 STA 模式下是否开启 DHCP | AT+WLSTADHCP=<MODE> |
| OK | 设置 STA 静态参数 | AT+WLSTATICPARAM=<IP>, <MASK>, <GATE> |
| OK | 设置 STA 模式参数 | AT+WLSTAPARAM=<SSID>, [PSW] 备注：对方 AP 加密方式为开放，PSW 可不填 |
| OK | 启动 Wi-Fi | AT+WLSETUP |
| OK | 断开 Wi-Fi | AT+WLCLOSE 备注：断开 WiFi 之后不会重连，SSID 和密码还会保存在 flash 内 |
| OK | 设置 MAC 地址 | AT+WLPMAC=<MAC> 备注：设置 MAC 地址要重启模块后才能生效 |
| OK | Start simple config | AT+WLSIMPLECONFIG 备注：可结合九九物联的配网 APP——“JJLink”实现快速配网 |
| OK | SoftAP 配网 | AT+WLAPCONFIG 备注：可结合九九物联的配网 APP——“JJLink”实现手动配网 |
| OK | Airkiss 配网 | AT+AIRKISS 备注：可结合九九物联提供的配网工具“AirkissDebugger” |
| TCP/IP 指令 | | |
| OK | 创建 TCP socket | AT+NWKCTCP=<ROLE>, <LOCALPORT>, [REMOTEIP], [REMOTEPORT] 备注：模块作为 Server 端， “[REMOTEIP], [REMOTEPORT]” 可不用填 |
| OK | 创建 UDP socket | AT+NWKUDPD=<TYPE>, [REMOTEIP], [REMOTEPORT], <LOCALPORT><CR><LF> |
| OK | 开启 TCP 透传模式 | AT+NWKTTCIP=<LOCALPORT>, <REMOTEIP>, <REMOTEPORT><CR><LF> |
| OK | 发送 TCP 数据 | AT+NWKTCPSND=<ID>, <LEN>, <DATA> |
| OK | 发送 UDP 数据 | AT+NWKUDPSND=<ID>, <REMOTEIP>, <REMOTEPORT>, <LEN>, <DATA> |
| OK | 关闭网络连接 | AT+NWKCLOSE=<ID> |

| | | |
|--------|-----------------------|---|
| OK | DNS 解析 | AT+NWKDNS=<HOST> |
| OK | 空中升级 | AT+WLOTA=<Server_IP>, <PORT> |
| OK | 通过 HTTP 进行 OTA 升级 | AT+WLHTTPOTA=<IP>, <PORT>, <PATH> |
| OK | 通过命令串口，实现 Xmodem 协议升级 | AT+XMODEOTA |
| OK | HTTP 请求 (GET) | AT+WLHTTPGET=<host>, <port>, <resource><CR><LF> |
| OK | HTTP 请求 (POST) | AT+WLHTTPPOST=<host>, <port>, <resource>, <type>, <data_len>, <data><CR><LF> |
| OK | HTTPS 请求 | AT+WLHTTPS=<Request_LEN>, <Request_PORT>, <Request_Domain>, <Request_Command> |
| 网络查询指令 | | |
| OK | 获取自身 IP | AT+WLIP |
| OK | 获取自身子网掩码 | AT+WLMASK |
| OK | 获取自身网关地址 | AT+WLGATE |
| OK | 获取自身 Mac 地址 | AT+WLMAC |
| OK | 扫描可用 AP | AT+WLSCAN |
| OK | 获取连接 AP 的 IP、掩码、网关 | AT+WLGETINFO |
| OK | 获取连接 AP 的 SSID、密码 | AT+WLGETAPINFO |
| OK | 获取当前代码执行区域 | AT+WLGETOTAINDEX |
| OK | 获取当前连接 AP 的 MAC | AT+WLGETAPMAC |
| OK | 获取当前连接 AP 的信号强度 | AT+WLRSSI |
| 提示指令 | | |
| OK | Wi-Fi 连接成功 (STA 模式下) | [AUTOCON] SUCCESS! |
| OK | Wi-Fi 断开连接 (STA 模式下) | [WLDIS] WLDISCON |
| OK | 接收网络数据 | RSP: <ID>, <TYPE>, [REMOTEIP], [REMOTEPORT], <LEN>, <DATA> |
| OK | 指示网络连接被中断 | RSP:CLOSED<ID> |

2 AT+指令集详细参数说明

2.1 基础指令

2.1.1 AT+S? 查询 AT+指令列表

| AT+S | |
|------|--------------------|
| 格式 | AT+S<CR><LF> |
| 功能说明 | 查询 AT+Command list |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|---|--------|
| [ATS] <CR><LF> <command list> [ATS]OK<CR><LF> | 返回指令列表 |
| [ATS]ERROR<CR> | 查询失败 |

示例：AT+S<CR><LF> 返回：[ATS] <CR><LF><command list>[ATS]OK<CR><LF>

2.1.2 AT+RST 重启模块

| AT+RST | |
|---------------------------|----------------|
| 格式 | AT+RST<CR><LF> |
| 功能说明 | 重启模块 |
| 参数列表 | 无 |
| 接收到回显 AT COMMAND READY 时间 | 460ms |

返回值：[RST]OK<CR><LF>

示例：AT+RST<CR><LF>（重启模块）

说明：回复[RST]OK<CR><LF>后模块马上重启。执行指令到模块准备好的时间为不开启重连情况下测试。

2.1.3 AT+CGPIO=R/W, PA_5, 0/1 , PIN_OUTPUT/PIN_INPUT, PullNone 设**置 GPIO 管脚输入/输出模式**

| AT+CGPIO | |
|----------|---|
| 格式 | AT+CGPIO= R/W, PA_5, 0/1 , PIN_OUTPUT/PIN_INPUT, MODE<CR><LF> |
| 功能说明 | 设置 GPIO 管脚输入/输出模式 |

参数列表:

| 参数 | 属性 | 属性描述 |
|----------------------|-----------|---|
| R/W | R | 管脚设置为读取模式 |
| | W | 管脚设置为写入模式 |
| PA_5 | I/O PIN | 管脚选择 |
| 0/1 | 高低电平 | 写 I/O 管脚电平输出 备注：读取 I/O 时，此参数可以任意填充，但参数不能缺失 |
| PIN_OUTPUT/PIN_INPUT | 输出/输入 | 输出/输入选择 |
| MODE | PullDown | 下拉 |
| | PullUp | 上拉 |
| | OpenDrain | 开漏 |
| | PullNone | 高阻态 |

返回值列表:

| 返回值 | 释义 | |
|---------------------|--------|--------------|
| [CGPIO]OK:1<CR><LF> | 读取 I/O | 读到 I/O 的电平为高 |
| | 写入 I/O | 写入 I/O 电平成功 |
| [CGPIO]OK:0<CR><LF> | 读取 I/O | 读取 I/O 的电平为低 |
| | 写入 I/O | 写入 I/O 电平失败 |

2.1.4 AT+SVER 查询固件版本

| AT+SVER | |
|---------|-----------------|
| 格式 | AT+SVER<CR><LF> |
| 功能说明 | 查询固件版本 |
| 参数列表 | 无 |

返回值列表:

| 返回值 | 释义 |
|--|------------|
| [ATSVER]OK, 99WSUR1N. 180718. A <CR><LF> | 获取软件版本信息成功 |
| [SVER]ERROR<CR><LF> | 获取失败 |

示例: AT+SVER<CR><LF> 返回: [ATSVER]OK, 99WSUR1N. 180718. A <CR><LF>

版本信息说明: 99-公司, W-WiFi, SUR1N-软件、接口、云支持, 180718-release 时间, A-当天版本次数

2.1.5 AT+SFTY 恢复出厂参数设置

| AT+SFTY | |
|---------------------------|--------------------------|
| 格式 | AT+SFTY<CR><LF> |
| 功能说明 | 恢复出厂参数设置, 擦除 flash 保存的信息 |
| 参数列表 | 无 |
| 接收到回显 AT COMMAND READY 时间 | 460ms |

返回值列表:

| 返回值 | 释义 |
|---------------------|---------------|
| [SFTY]OK<CR><LF> | 清除 Flash 数据成功 |
| [SFTY]ERROR<CR><LF> | 指令格式或者参数错误 |

示例: AT+SFTY<CR><LF> 返回: [SFTY]OK<CR><LF>

注意: 输入此指令返回成功后, 模块立即复位。

2.1.6 AT+SWEB=<MODE>, <WLAN_MODE> 启动/停止 Web 服务器

| AT+SWEB（串口版本） | |
|---------------|---|
| 格式 | AT+SWEB=<MODE>, <WLAN_MODE><CR><LF> |
| 功能说明 | 启动/停止 Web 服务器，第三方设备输入模块的 IP 进入 Web 界面，目前界面只是做了 SSID、PASSWORD 的设置和配网 |

参数列表：

| 参数 | 属性 | 属性描述 |
|-----------|--------|--------------|
| MODE | C | 创建 Web 服务器 |
| | S | 停止 Web 服务器 |
| WLAN_MODE | STA | STA 模式下 |
| | AP | AP 模式下 |
| | STA+AP | STA+AP 共存模式下 |

返回值列表：

| 返回值 | 释义 |
|---------------------|--------------------------------|
| [SWEB]OK<CR><LF> | 创建成功 |
| [SWEB]ERROR<CR><LF> | 创建失败或者参数有误 |
| 设置 Web 页面后 | [WEB]SSID:xxx, PWD:xxx<CR><LF> |
| | [WEBSETPARAM]OK<CR><LF> |

示例：AT+SWEB=C, STA<CR><LF>

返回：[SWEB]OK<CR><LF>

2.1.7 AT+WLICKPS=<MODE> 设置 Tickless 睡眠模式

| AT+WLICKPS | |
|------------|---|
| 格式 | AT+WLICKPS=<MODE><CR><LF> |
| 功能说明 | 进入/退出睡眠模式，此睡眠模式可保持 Socket 连接，发送指令后，10 秒钟后进入睡眠模式 |
| 唤醒引脚 | PA_5 (拉低电平唤醒) |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|-----|--------|
| MODE | ON | 进入睡眠模式 |
| | OFF | 退出睡眠模式 |

返回值列表：

| 返回值 | 释义 |
|-------------------------------|----------|
| [WLICKPS]ENTER SLEEP!<CR><LF> | 成功进入睡眠模式 |
| [WLICKPS]QUIT SLEEP!<CR><LF> | 退出睡眠模式 |
| [WLICKPS]ERROR!<CR><LF> | 进入睡眠模式失败 |

示例：AT+WLICKPS=ON<CR><LF> 返回：[WLICKPS]ENTER SLEEP!<CR><LF>

说明：模块唤醒操作：拉低 PA_5 引脚大于 5ms 后再发 AT+WLICKPS=OFF 指令，回显 [WLICKPS]QUIT SLEEP!则表示唤醒成功。

2.1.8 AT+WLSLEEP=<MODE>, <WAKE_ENENT>, <WAKE_PIN>, <SLEEP_TIME>, <WAKE_PIN_ACTIVE> 设置低功耗模式

| AT+WLSLEEP | |
|------------|--|
| 格式 | AT+WLSLEEP=<MODE>, <WAKE_ENENT>, <WAKE_PIN>, <SLEEP_TIME>, <WAKE_PIN_ACTIVE><CR><LF> |
| 功能说明 | 启动低功耗模式，包含待机模式(STANDBY)和深度睡眠模式(DEEP SLEEP)，该两种模式执行后不能保持模块的 AP 连接和 Socket 连接，唤醒后会重启系统 |

参数列表:

| 参数 | 指令输入属性 | | 属性描述 | |
|------------|------------------------|---------------------------|--------|----------------|
| MODE | DEEP STANDBY | | 待机 | |
| | DEEP SLEEP | | 深度睡眠 | |
| 参数 | 属性 | 适用模式 | 指令输入属性 | 属性描述 |
| WAKE_EVENT | SLEEP_WAKEUP_BY_STIMER | DEEP STANDBY \ DEEP SLEEP | 0 | 定时唤醒方式 |
| | SLEEP_WAKEUP_BY_GPIO | DEEP STANDBY \ DEEP SLEEP | 1 | 特殊引脚唤醒（下个参数介绍） |

| 参数 | 指令输入属性 | 属性描述 | |
|-----------------|--------|--|-----------|
| WAKE_PIN | PA_5 | 特殊唤醒管脚 PA_5、PA_18、PA_22、PA_23（建议使用 PA_5） | |
| | 0 | 唤醒事件 WAKE_EVENT 没用到 SLEEP_WAKEUP_BY_GPIO，就填参数“0” | |
| SLEEP_TIME | 时间数值 | 时间单位为 ms（最大可以做到 130 分钟） | |
| | 0 | 唤醒事件 WAKE_EVENT 没用到 SLEEP_WAKEUP_BY_STIMER，就填参数“0” | |
| WAKE_PIN_ACTIVE | 1 | 高电平唤醒 | |
| | 0 | 唤醒事件 WAKE_EVENT 用到 SLEEP_WAKEUP_BY_GPIO | 表示“低电平唤醒” |
| | | 唤醒事件 WAKE_EVENT 没用到 SLEEP_WAKEUP_BY_GPIO | 就填参数“0” |

返回值列表:

| 返回值 | 释义 |
|--------------------------------|------------------------|
| Enter DeepSleep Mode<CR><LF> | 进入深度睡眠 DEEP SLEEP 成功 |
| Enter DeepStandby Mode<CR><LF> | 进入待机模式 DEEP STANDBY 成功 |
| AT COMMAND READY<CR><LF><LF># | 深度睡眠唤醒成功 |
| | 待机模式唤醒成功 |

示例：AT+WLSLEEP=SLEEP, 0, 0, 10000, 0<CR><LF> 返回：Enter Sleep Mode<CR><LF>

2.1.9 AT+WLSC 切换模块程序运行区域

| AT+WLSC | |
|---------------------------|--|
| 格式 | AT+WLSC=<MODE><CR><LF> |
| 功能说明 | 模块存在两个程序运行区域，每个区域的程序都是独立存在和运行，用户可以通过该指令切换模块程序运行区域，该指令可结合指令 AT+WLGETOTAINDEX 使用 |
| 接收到回显 AT COMMAND READY 时间 | 460ms |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|----|------------------------|
| MODE | 1 | 跳转到 OTA1 区域，模块重启执行相应区域 |
| | 2 | 跳转到 OTA2 区域，模块重启执行相应区域 |

返回值列表：

| 返回值 | 释义 |
|---------------------|------------------|
| [WLSC]OK<CR><LF> | 跳转成功 |
| [WLSC]ERROR<CR><LF> | 跳转失败，参数不是 0 或者 1 |

示例：AT+WLSC=1<CR><LF> 返回：[WLSC]OK<CR><LF>

说明：模块返回 [WLSC]OK<CR><LF> 时模块马上进入复位。

2.1.10 AT+WLSETCHANNEL=<MODE> 设置 Channel Plan

| AT+WLSETCHANNEL | |
|-----------------|---|
| 格式 | AT+WLSETCHANNEL=<MODE><CR><LF> |
| 功能说明 | 设置 Channel Plan，为了使用于国外区域要求，该指令不会保存 flash，需要用户上电发起该指令执行 |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|----|---------------------|
| MODE | 0 | Worldwird 13(1-13) |
| | 1 | Europe 2G(1-13) |
| | 2 | US 2G(1-11) |
| | 3 | Japan 2G (1-13, 14) |
| | 4 | France 2G(10-13) |
| | 5 | US 2G(1-13) |

返回值列表：

| 返回值 | 释义 |
|-----------------------------|--------------------|
| [WLSETCHANNEL]OK<CR><LF> | 设置 Channel Plan 成功 |
| [WLSETCHANNEL]ERROR<CR><LF> | 设置 Channel Plan 失败 |

示例：AT+WLSETCHANNEL=5<CR><LF> 返回：[WLSETCHANNEL]OK<CR><LF>

说明：用户可以根据自己产品的使用区域，上电发命令设置该参数

2.1.11 AT+WLFASTCONNECT=<MODE> 设置模块是否开启重连 Wi-Fi

| AT+WLFASTCONNECT（串口版本） | |
|------------------------|---|
| 格式 | AT+WLFASTCONNECT=<MODE><CR><LF> |
| 功能说明 | <p>设置模块是否开启重连 Wi-Fi，该指令开关参数不保存 flash；</p> <p>注意：1、如果只设置该指令的开关配置，模块会进行无限重连 Wi-Fi，每次的间隔时间为 10 秒；</p> <p>2、如果在设置该指令后，再发指令 AT+WLFASTPARAM 设置重连次数和时间间隔，模块会在 AP 断电后只执行设置的重连次数</p> |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|-----|---------------|
| MODE | ON | 开启重连 Wi-Fi 功能 |
| | OFF | 关闭重连 Wi-Fi 功能 |

返回值列表：

| 返回值 | 释义 |
|-------------------------------|---------------|
| [WLFASCTCONNECT]OK!<CR><LF> | 模块开启或关闭快速连接成功 |
| [WLFASCTCONNECT]ERROR<CR><LF> | 模块开启或关闭连接失败 |

示例：AT+WLFASCTCONNECT=ON<CR><LF> 返回：[WLFASCTCONNECT]OK!<CR><LF>

2.1.12 AT+WLFASCTPARAM=< Retry_Count>,<Idle_Time>设置模块重连的次数与重连间隔时间

| AT+WLFASCTPARAM（串口版本） | |
|-----------------------|---|
| 格式 | AT+WLFASCTPARAM=<Retry_Count>,<Idle_Time><CR><LF> |
| 功能说明 | 设置模块重连的次数与重连间隔时间 |

参数列表：

| 参数 | 属性 | 属性描述 |
|-------------|-------|--------------------|
| Retry_Count | 1-255 | 重连 Wi-Fi 次数 |
| Idle_Time | 1-255 | 每次重连 Wi-Fi 间隔时间（s） |

注意：使用微信 Airkiss 配网功能时，这个值不能设置超过 8 秒

返回值列表：

| 返回值 | 释义 |
|-----------------------------|----------|
| [WLFASCTPARAM]OK<CR><LF> | 设置重连参数成功 |
| [WLFASCTPARAM]ERROR<CR><LF> | 设置重连参数失败 |

示例：AT+WLFASCTPARAM=5,8<CR><LF> 返回：[WLFASCTPARAM]OK<CR><LF>



2.1.13 AT+WLAUTOCONNECT=<MODE>

| AT+WLAUTOCONNECT（SPI 版本） | |
|--------------------------|--|
| 格式 | AT+WLAUTOCONNECT=<MODE><CR><LF> |
| 功能说明 | 设置模块是否开启重连 Wi-Fi，该指令开关参数不保存 flash；上电不重连 Wi-Fi 注意：1、如果只设置该指令的开关配置，模块会进行无限重连 Wi-Fi，每次的间隔时间为 10 秒； |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|----|-------------------|
| MODE | 0 | 默认重连 Wi-Fi 次数 8 次 |
| | 1 | 关闭重连 Wi-Fi |
| | 2 | 无限重连 Wi-Fi |

返回值列表：

| 返回值 | 释义 |
|------------------------------|---------------|
| [WLAUTOCONNECT]OK<CR><LF> | 模块开启或关闭快速连接成功 |
| [WLAUTOCONNECT]ERROR<CR><LF> | 模块开启或关闭连接失败 |

示例：AT+WLAUTOCONNECT=1<CR><LF>

返回：[WLAUTOCONNECT]OK<CR><LF>

2.2 Wi-Fi 控制指令

2.2.1 AT+WLMODE=<MODE> 设置无线工作模式

| AT+WLMODE | |
|-----------|--------------------------|
| 格式 | AT+WLMODE=<MODE><CR><LF> |
| 功能说明 | 设置无线工作模式 |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|----|-----------|
| MODE | 1 | STA 模式 |
| | 2 | AP 热点模式 |
| | 3 | STA+AP 模式 |

返回值列表：

| 返回值 | 释义 |
|-----------------------|------|
| [WLMODE]OK<CR><LF> | 设置成功 |
| [WLMODE]ERROR<CR><LF> | 设置失败 |

示例：AT+WLMODE=1<CR><LF> 返回 [WLMODE]OK<CR><LF> （设置无线模式为 STA 模式，系统返回成功）

2.2.2 AT+WLDAPDHCPGATE=<START_IP>,<END_IP>,<GATEWAY> 设置 AP 模式下 DHCP 和网关

| AT+WLDAPDHCPGATE | |
|------------------|---|
| 格式 | AT+WLDAPDHCPGATE=<START_IP>,<END_IP>,<GATEWAY><CR><LF> |
| 功能说明 | 设置 AP 模式下 DHCP 和网关，参数需要在启动指令 AT+WLAPPARAM 后才能被保存在 Flash 内 |

参数列表：

| 参数 | 属性 | 属性描述 |
|----------|---------------|----------|
| START_IP | xxx. xxx. xxx | 客户端开始 IP |
| END_IP | xxx. xxx. xxx | 客户端结束 IP |
| GATEWAY | xxx. xxx. xxx | 网关 IP |

返回值列表：

| 返回值 | 释义 |
|-------------------------|------|
| [WLAPDHCPGATE]OK<CR> | 设置成功 |
| [WLAPDHCPGATE]ERROR<CR> | 设置失败 |

示例：AT+ WLAPDHCPGATE =192. 168. 2. 10, 192. 168. 2. 100, 192. 168. 2. 1<CR><LF>

返回[WLAPDHCPGATE]OK<CR><LF>

NOTE：1. 默认网关 IP 是 192. 168. 43. 1

2. DHCP 模式下，配置 AP 的 DHCP 规则

2.2.3 AT+WLAPDHCP=<MODE> 设置 AP 模式下是否开启 DHCP

| AT+WLAPDHCP | |
|-------------|--|
| 格式 | AT+WLAPDHCP=<MODE><CR><LF> |
| 功能说明 | 设置 AP 模式下是否开启 DHCP 备注：必须先执行 AT+WLMODE=2 |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|-----|---------|
| MODE | ON | DHCP 开启 |
| | OFF | DHCP 关闭 |

返回值列表：

| 返回值 | 释义 |
|-------------------------|------|
| [WLAPDHCP]OK<CR><LF> | 设置成功 |
| [WLAPDHCP]ERROR<CR><LF> | 设置失败 |

示例：AT+ WLAPDHCP =ON<CR> 返回[WLAPDHCP]OK<CR>

NOTE: 1. 默认是开启 DHCP 2. 必须先执行 AT+WLMODE=2

2.2.4 AT+WLAPPARAM=<SSID>,<SEC>,<PSW> 设置 AP 模式参数

| AT+WLAPPARAM | |
|--------------|---|
| 格式 | AT+WLAPPARAM=<SSID>,<SEC>,<PSW><CR><LF> |
| 功能说明 | 设置 AP 模式参数 |

参数列表:

| 参数 | 属性 | 属性描述 |
|------|----------|------------------------------------|
| SSID | SSID | SoftAP 的 SSID |
| SEC | OPEN | 无加密, 设置该模式时, Password 需要随意填充字符串 |
| | AES | 加密方式为 WEP |
| PSW | Password | SoftAP 的密码 (仅在 SEC 属性为 AES 下有效且必须) |

返回值列表:

| 返回值 | 释义 |
|--------------------------|------|
| [WLAPPARAM]OK<CR><LF> | 设置成功 |
| [WLAPPARAM]ERROR<CR><LF> | 设置失败 |

示例：AT+WLAPPARAM=99iot_TEST,AES,12345678<CR><LF>

返回:[WLAPPARAM]OK<CR><LF> (设置 SoftAP 的 SSID 为 99iot_TEST, 加密方式为 AES, 密码位 12345678, 系统返回设置成功)

说明: 模块为 AP 模式下时候如果有设备连接上则回显[LINK]OK<CR><LF>, 设备断开连接回显[LINK]DIS<CR><LF>。

2.2.5 AT+WLSTADHCP=<CS> 设置 STA 模式下是否开启 DHCP

| AT+WLSTADHCP | |
|--------------|---|
| 格式 | AT+WLSTADHCP=<CS><CR><LF> |
| 功能说明 | 设置 STA 模式下是否开启 DHCP 备注：必须先执行 AT+WLMODE=1 |

参数列表：

| 参数 | 属性 | 属性描述 |
|----|-----|---------|
| CS | ON | DHCP 开启 |
| | OFF | DHCP 关闭 |

返回值列表：

| 返回值 | 释义 |
|--------------------------|------|
| [WLSTADHCP]OK<CR><LF> | 设置成功 |
| [WLSTADHCP]ERROR<CR><LF> | 设置失败 |

示例：AT+WLSTADHCP=OFF<CR><LF> 返回 [WLSTADHCP]OK<CR><LF>

NOTE: 1. STA 模式下，默认是开启 DHCP
2. 必须先执行 AT+WLMODE=1

2.2.6 AT+WLSTATICPARAM=<IP>,<MASK>,<GATE> 设置 STA 静态参数

| AT+WLSTATICPARAM | |
|------------------|---|
| 格式 | AT+WLSTATICPARAM=<IP>,<MASK>,<GATE><CR><LF> |
| 功能说明 | 设置 STA 模式下静态参数，参数需要在 AT+WLSTAPARAM 指令后被保存 Flash 备注：必须先用 AT+WLSTADHCP=OFF，关掉 DHCP |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|---------------|------------------|
| IP | xxx. xxx. xxx | 设置 STA 模式下静态 IP |
| MASK | xxx. xxx. xxx | 设置 STA 模式下静态子网掩码 |
| GATE | xxx. xxx. xxx | 设置 STA 模式下 静态网关 |

返回值列表：

| 返回值 | 释义 |
|------------------------------|------|
| [WLSTATICPARAM]OK<CR><LF> | 设置成功 |
| [WLSTATICPARAM]ERROR<CR><LF> | 设置失败 |

示例：AT+WLSTATICPARAM=192. 168. 1. 100, 255. 255. 255. 0, 192. 168. 1. 1<CR><LF>

返回：[WLSTATICPARAM]OK<CR>

NOTE：必须启动 AT+WLSTAPARAM 配网指令才能保存下来

2.2.7 AT+WLSTAPARAM=<SSID>, [PSW] 设置 STA 模式参数

| AT+WLSTAPARAM | |
|---------------|-------------------------------------|
| 格式 | AT+WLSTAPARAM=<SSID>, [PSW]<CR><LF> |
| 功能说明 | 设置 STA 模式参数 备注：输入此指令后，模块进行配网 |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|----------|------------------------------------|
| SSID | SSID | 对方 AP 的 SSID |
| PSW | Password | 对方 AP 的密码(如果对方 AP 加密方式为开放, 此属性可不填) |

返回值列表：

| 返回值 | 释义 |
|---------------------------|-----------|
| [WLSTAPARAM]OK<CR><LF> | 设置成功，配网成功 |
| [WLSTAPARAM]ERROR<CR><LF> | 设置失败，配网失败 |

示例：AT+WLSTAPARAM=99iot,SZ99iot<CR><LF> 返回[WLSTAPARAM] OK<CR><LF>
（设置 STA 模式下对方 AP 的 SSID 为 99iot，密码为 SZ99iot）

2.2.8 AT+WLSETUP 启动 Wi-Fi 连接

| AT+WLSETUP | |
|------------|--------------------|
| 格式 | AT+WLSETUP<CR><LF> |
| 功能说明 | 启动 Wi-Fi 连接 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|------------------------|------|
| [WLSETUP]OK<CR><LF> | 启动成功 |
| [WLSETUP]ERROR<CR><LF> | 启动失败 |

示例：AT+WLSETUP<CR><LF>返回 [WLSETUP]OK<CR> （启动无线连接）

2.2.9 AT+WLCLOSE 断开 Wi-Fi 连接

| AT+WLCLOSE | |
|------------|----------------------------------|
| 格式 | AT+WLCLOSE<CR><LF> |
| 功能说明 | 断开 Wi-Fi 连接 备注：断开 WiFi 之后不会重连 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|------------------------|------|
| [WLCLOSE]OK<CR><LF> | 断开成功 |
| [WLCLOSE]ERROR<CR><LF> | 断开失败 |

示例：AT+WLCLOSE <CR><LF> 返回 [WLCLOSE]OK<CR><LF> （断开无线连接）

2.2.10 AT+WLPMAC=<MAC> 设置 MAC 地址

| AT+WLPMAC（留意备注） | |
|-----------------|---|
| 格式 | AT+WLPMAC=<MAC><CR><LF> |
| 功能说明 | 设置 MAC 地址 备注：采用此功能时，必须把模块出厂的 MAC 地址记录下来，以免弄丢正常地址，MAC 地址不能重写，写入次数有限（慎重） |

参数列表：

| 参数 | 属性 | 属性描述 |
|-----|--------------|-------------------------|
| MAC | 112233445566 | 格式：6 个 bytes Hex number |

返回值列表：

| 返回值 | 释义 |
|-----------------------|------------|
| [WLPMAC]OK<CR><LF> | MAC 地址设置成功 |
| [WLPMAC]ERROR<CR><LF> | MAC 地址设置失败 |

示例：AT+WLPMAC=112233445566<CR><LF> 返回 [WLPMAC]OK<CR>

NOTE：1、此功能写入次数有限，不能随便写入

2、必须重启模块后 MAC 地址才生效



2.2.11 AT+WLSIMPLECONFIG 启动快速配网

| AT+WLSIMPLECONFIG | |
|-------------------|---|
| 格式 | AT+WLSIMPLECONFIG<CR><LF> |
| 功能说明 | 启动快速配网 备注：可结合九九物联的配网 APP——“JJLink”实现配网（快速入网） |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 | | |
|--|---|--------------|--------|
| [WLSIMPLECONFIG]OK, MODE, SSID, Channel, AES, Password<CR><LF> | MODE | STA 模式 | 快速入网成功 |
| | SSID | 路由的 SSID | |
| | Channel | 信道 | |
| | AES | 加密方式 | |
| | Password | 路由的 Password | |
| | [WLSIMPLECONFIG]OK | 入网成功 | |
| [WLSIMPLECONFIG]ERROR<CR><LF> | 120s 没收到信息超时或者快速入网失败（请检查手机 app 发送的信息是否正确） | | |
| [WLSIMPLECONFIG]TIMEOUT<CR><LF> | 配网超时（1 分钟） | | |

示例：AT+WLSIMPLECONFIG<CR><LF> 返回

[WLSIMPLECONFIG]OK, STA, 99iot, 11, AES, SZ99iot12345 <CR><LF>

NOTE：1、结合九九物联的配网 APP——JJLink 实现快速入网

2、模块开始配网时返回[WLSIMPLECONFIG]OK<CR><LF>，当模块连接上服务器时（TCP 连接）进行此模式配网模块会返回 RSP:CLOSED[ID]与 [WLDIS]WLDISCON，应用时需要与模式配网返回信息进行区分，以准确判断模块是否成功配网。

3、模块接收到信息后第一次连接失败后会进行重连，最多重连 5 次，如果 5 次都连接失败则返回[WLSIMPLECONFIG]ERROR（从模块接收到信息开始到返回此信息的时间大约为 30S），此时应该检测一下配网信息是否正确。



2.2.12 AT+WLAPCONFIG 连接模块热点配网

| AT+WLAPCONFIG | |
|---------------|--|
| 格式 | AT+WLAPCONFIG<CR><LF> |
| 功能说明 | 启动模块热点配网 备注：可结合九九物联的配网 APP—“JJLink”实现配网（手动入网） |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 | |
|---|----------------|-----------------------------|
| [WLAPCONFIG]START! | 开始配网 | 等待接收手机 APP 信息 |
| [WLAPCONFIG]OK! | 配网成功 | 同一时间返回的三个参数，快速入网成功 |
| SSID: “字符串” | 返回路由的 SSID | |
| PASSWORD: “字符串” | 返回路由的 Password | |
| RECONNECT:1 RECONNECT:2 RECONNECT:3 RECONNECT:4 RECONNECT:5 | 重连 5 次 | 快速入网失败(请检查手机 app 发送的信息是否正确) |
| [WLAPCONFIG]ERR<CR><LF> | 配网失败 | |

示例： 1、让模块建立 AP 热点：AT+WLMODE=2, AT+WLAPPARAM=SSID, AES, Password
2、向模块发送指令：AT+WLAPCONFIG
3、手机连接模块 AP 热点，进入手机 APP 的手动入网界面，发送配网信息
4、如果连接路由第一次失败，则接下来会进行重连，最大重连次数为 5 次，重连完后最后返回 [WLAPCONFIG]ERROR 的时间为 50S 左右。

NOTE：当模块连接上服务器时（TCP 连接）进行此模式配网模块会返回 RSP:CLOSED[ID]，与 [WLDISCON]WLAN_DISCONNECT，应用时需要与模式配网返回信息进行区分，以准确判断模块是否成功配网。



2.2.13 AT+AIRKISS 微信快速配网

| AT+AIRKISS | |
|------------|--|
| 格式 | AT+AIRKISS<CR><LF> |
| 功能说明 | 启动微信快速配网 备注：可结合九九物联提供的配网工具 “AirKissDebugger” |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 | |
|-------------------|----------------|-----------------------------|
| [AIRKISS]START! | 开始配网 | 等待接收手机 APP 信息 |
| [AIRKISS]SC_OK! | 配网成功 | 同一时间返回的三个参数，快速入网成功 |
| SSID: “字符串” | 返回路由的 SSID | |
| PASSWORD: “字符串” | 返回路由的 Password | |
| [AIRKISS]TIMEOUT! | 配网超时 | 模块没接收到信息 |
| [AIRKISS]SC_ERR | 配网失败 | 快速入网失败(请检查手机 app 发送的信息是否正确) |

示例：发送微信配网指令：AT+AIRKISS<CR><LF> 返回：[AIRKISS]START!

模块完成配网后返回：[AIRKISS]SC_OK!

NOTE: 1、当模块连接上服务器时（TCP 连接）进行此模式配网模块会返回 RSP:CLOSED[ID]，与[WLDISCON]WLAN_DISCONNECT，应用时需要与模式配网返回信息进行区分，以准确判断模块是否成功配网。

2、模块如果收到信息后未连接路由成功，最多自动重连 5 次。重连完后回显[AIRKISS]ERR 的时间大约为 50S。



2.2.14 AT+ NWTTCP=<LOCLALPORT>, [REMOTEIP], [REMOTEPORT] 进入数
据透传模式

| AT+NWTTCP | |
|-----------|--|
| 格式 | AT+NWTTCP=<LOCLALPORT>, [REMOTEIP], [REMOTEPORT]<CR><LF> |
| 功能说明 | 进入 TCP 透传模式 |

参数列表:

| 参数 | 属性 | 属性描述 |
|------------|-------|----------------|
| LOCLALPORT | 端口号 | 本地端口号, 0 表示随机 |
| REMOTEIP | IP 地址 | Server 端 IP 地址 |
| REMOTEPORT | 目标端口号 | Server 端端口号 |

返回值列表:

| 返回值 | 释义 |
|-----------------------|----------|
| [NWTTCP]OK<CR><LF> | 进入透传模式成功 |
| [NWTTCP]ERROR<CR><LF> | 进入透传模式失败 |



2.3 TCP/IP 指令

2.3.1 AT+NWKCTCP=<ROLE>, <LOCLALPORT>, [REMOTEIP], [REMOTEP
ORT] 创建 TCP 连接

| AT+NWKCTCP | |
|-------------------|---|
| 格式 | AT+NWKCTCP=<ROLE>, <LOCLALPORT>, [REMOTEIP], [REMOTEP ORT]<CR><LF> |
| 功能说明 | 创建 TCP 连接 备注：模块作为 Server 端，“[REMOTEIP], [REMOTEP ORT]”可不用填 |
| TCP 建立时间为 | 500ms |
| TCP 建立失败，超时 时间 | 20s |

参数列表：

| 参数 | 属性 | 属性描述 |
|----------------|--------|-----------------------------------|
| ROLE | CLIENT | Client 模式 |
| | SERVER | Server 模式 |
| LOCLALPORT | 端口号 | 本地端口号，0 表示随机 |
| REMOTEIP | IP 地址 | Server 端 IP 地址（仅 CLIENT 模式下有效且必须） |
| REMOTEP ORT | 目标端口号 | Server 端端口号（仅 CLIENT 模式下有效且必须） |

返回值列表：

| 返回值 | 释义 |
|----------------------------|------------|
| [NWKCTCP_ID] ID<CR><LF> | 创建的网络连接 ID |
| [NWKCTCP_ID] ERROR<CR><LF> | 创建失败 |

示例：AT+NWKCTCP=CLIENT, 1234, 192. 168. 1. 123, 4321<CR><LF> 返回

[NWKCTCP_ID] 1<CR><LF> （创建 TCP Client，本地端口号为 1234，Server 端



IP 为 192.168.1.123，Server 端端口号为 4321，系统返回创建成功的连接 ID 为 1)

- 说明：1、模块作为客户端连接服务器的时候同一端口只能成功连接一次。
- 2、模块连接服务器时，模块有 keepalive 机制，如果模块与服务器 4S 内无数据交互，模块则间隔 2S 发送一次保活探测，总共发送 4 次保活探测，也就是说最快能够在 12S 内检测服务器与模块断开连接（回显为 RSP: CLOSED, ID）。

2.3.2 AT+NWKUDP=<TYPE>, [REMOTEIP], [REMOTEPORT], <LOCALPORT> 创建 UDP CLIENT 端

| AT+NWKUDP | |
|-----------|---|
| 格式 | AT+NWKUDP=<TYPE>, [REMOTEIP], [REMOTEPORT], <LOCALPORT><CR><LF> |
| 功能说明 | 创建 UDP CLIENT 连接 |

参数列表：

| 参数 | 属性 | 属性描述 | |
|------------|--------|----------------------------|--------------------|
| TYPE | CLIENT | 客户端模式 | |
| | SERVER | 服务端模式 | |
| REMOTEIP | 目标 IP | TYPE 为 SERVER 时 不需要填充参数 | xxx. xxx. xxx. xxx |
| REMOTEPORT | 目标端口 | | (1-65535) |
| LOCALPORT | 模块本地端口 | (1-65535) | |

返回值列表：

| 返回值 | 释义 |
|---------------------------|---------------|
| [NWKUDP_ID] ID<CR><LF> | 创建的网络连接 ID |
| [NWKUDP_ID] ERROR<CR><LF> | 创建 UDP 网络连接失败 |

示例：

- 1、 建立 UDP CLIENT：AT+NWKUDP=CLIENT, 192.168.3.102, 8086, 8080<CR><LF>
返回 [NWKUDP_ID] 2<CR><LF> （创建 UDP Socket，目标 IP 为：

192.168.3.102，目标端口为 8086，本地端口号为 8080，系统返回创建成功且创建的连接 ID 为 2）

2、建立 UDP SERVER: AT+NWKUDP=SERVER, 8001<CR><LF>

返回[NWKUDP_ID]1<CR><LF>（创建 UDP Socket，本地端口号为 8001，系统返回创建成功且创建的连接 ID 为 1）

2.3.3 AT+NWKTTC=⟨LOCALPORT⟩,⟨REMOTEIP⟩,⟨REMOTEPORT⟩ 开启 TCP CLIENT 透传模式

| AT+NWKTTC | |
|-----------|--|
| 格式 | AT+NWKTTC=⟨LOCALPORT⟩,⟨REMOTEIP⟩,⟨REMOTEPORT⟩<CR><LF> |
| 功能说明 | <p>开启 TCP CLIENT 透传模式，双方进行数据透传，发送特殊字符串“++++”可临时退出透传模式，该连接保持，操作完成其他任务后，发送 AT+NWKTTC=ON 可再次进入透传模式。</p> <p>如需退出透传模式并且关闭当下透传 Socket 连接，在发送“++++”后，发送 AT+NWKTTC=OFF</p> |

参数列表：

| 参数 | 属性 | 属性描述 |
|------------|--------|------------------------|
| LOCALPORT | 模块本地端口 | 0/(1-65535)，0 代表随机生成端口 |
| REMOTEIP | 目标 IP | xxx.xxx.xxx.xxx |
| REMOTEPORT | 目标端口 | (1-65535) |

返回值列表：

| 返回值 | 释义 |
|----------------------|------------------------|
| [NWKTTC]OK<CR><LF> | 进入透传成功 |
| [NWKTTC]QUIT<CR><LF> | 临时退出透传模式成功，Socket 连接保持 |
| [NWKTTC]EXIT<CR><LF> | 退出透传模式并关闭 Socket 连接成功 |

透传模式下衍生指令：

| | |
|----------------|-----------------------------|
| ++++ | 临时退出透传模式，保存透传 TCP Socket 连接 |
| AT+NWKTTCP=ON | 重新进入透传指令 |
| AT+NWKTTCP=OFF | 完全退出透传模式，关闭透传 TCP Socket 连接 |

示例：AT+NWKTTCP=0, 192.168.1.167, 8000<CR><LF> 返回：[NWKTTCP]OK<CR><LF>

- 1、临时退出：发送 ++++ 返回：[NWKTTCP]QUIT<CR><LF>
- 2、重新进入：AT+NWKTTCP=ON 返回：[NWKTTCP]OK<CR><LF>
- 3、完全退出：执行第 1 步后，发送 AT+NWKTTCP=OFF<CR><LF>
返回：[NWKTTCP]EXIT<CR><LF>

2.3.4 AT+NWKTCPSEND=<ID>, <LEN>, <DATA> 发送 TCP 数据

| AT+NWKTCPSEND | |
|---------------|---|
| 格式 | AT+NWKTCPSEND=<ID>, <LEN>, <DATA><CR><LF> |
| 功能说明 | 发送 TCP 数据 |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|------|-----------------------|
| ID | ID | 网络连接 ID |
| LEN | 数据长度 | 发送数据的长度 |
| DATA | 数据 | 要发送的数据（不可超过 1500 个字节） |

返回值列表：

| 返回值 | 释义 |
|---------------------------|------|
| [NWKTCPSEND]OK<CR><LF> | 发送成功 |
| [NWKTCPSEND]ERROR<CR><LF> | 发送失败 |

示例：AT+NWKTCPSEND=1, 8, 99wulian<CR><LF> 返回 [NWKTCPSEND]OK<CR><LF>（给 ID 为 1 的 TCP 连接发送数据，数据长度为 8，数据为：99wulian）

2.3.5 AT+NWKUDPSSEND=<ID>,<REMOTEIP>,<REMOTEPORT><LEN>,<D**ATA> 发送 UDP 数据**

| AT+NWKUDPSSEND | |
|----------------|--|
| 格式 | AT+NWKUDPSSEND=<ID>,<REMOTEIP>,<REMOTEPORT>,<LEN>,<DATA><CR><LF> |
| 功能说明 | 创建 TCP 连接 |

参数列表:

| 参数 | 属性 | 属性描述 |
|------------|-------|--------------|
| ID | ID | 网络连接 ID |
| REMOTEIP | IP 地址 | 要发送的远程 IP 地址 |
| REMOTEPORT | 端口号 | 要发送的远程端口号 |
| LEN | 数据长度 | 要发送的数据长度 |
| DATA | 数据 | 要发送的数据 |

返回值列表:

| 返回值 | 释义 |
|----------------------------|------|
| [NWKUDPSSEND]OK<CR><LF> | 发送成功 |
| [NWKUDPSSEND]ERROR<CR><LF> | 发送失败 |

示例: AT+NWKUDPSSEND=1,192.168.1.123,1234,8,99wulian<CR><LF>

返回 [NWKUDPSSEND]OK<CR><LF> (给 ID 为 1 的 UDP 远程地址发送数据, 目的 IP 为 192.168.1.123, 目的端口号为 1234, 发送的数据长度为 8, 发送的数据为 99wulian, 系统返回发送成功)

2.3.6 AT+NWKCLOSE=<ID> 关闭网络连接

| AT+NWKCLOSE | |
|-------------|--------------------------|
| 格式 | AT+NWKCLOSE=<ID><CR><LF> |
| 功能说明 | 关闭网络连接 |

参数列表：

| 参数 | 属性 | 属性描述 |
|----|----|-------------|
| ID | ID | 已建立网络连接的 ID |

返回值列表：

| 返回值 | 释义 |
|-----------------------|----------|
| [NWKCLOSE]OK<CR><LF> | 关闭网络连接成功 |
| [NWKUDP]ERROR<CR><LF> | 关闭网络连接失败 |

示例：AT+NWKCLOSE =1<CR><LF> 返回 [NWKCLOSE]OK<OK><LF>（关闭 ID 为1 的网络连接，系统返回关闭成功）

2.3.7 AT+NWKDNS=<HOST> DNS 解析

| AT+NWKDNS | |
|-----------|--------------------------|
| 格式 | AT+NWKDNS=<HOST><CR><LF> |
| 功能说明 | DNS 解析 |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|-------|---------|
| HOST | 域名字符串 | 需要解析的域名 |

返回值列表：

| 返回值 | 释义 |
|-----------------------|--------|
| [IP]<CR><LF> | 域名的 IP |
| [NWKDNS]ERROR<CR><LF> | 解析失败 |

示例：AT+NWKDNS=www.baidu.com<CR><LF> 返回 220.181.112.244<CR><LF>

2.3.8 AT+PING=<HOST> PING 指令

| AT+PING | |
|---------|---|
| 格式 | AT+PING=<HOST><CR><LF> |
| 功能说明 | PING 指令，模块往需要 PING 的地址发送 5 个包数据，每个包的数据 120 个字节，查看是否丢包？延时时间多久？ |

参数列表：

| 参数 | 属性 | 属性描述 |
|------|-----|------------|
| HOST | 字符串 | 域名或者 IP 地址 |

返回值列表：

| 返回值 | 释义 |
|---|------------------------------------|
| [PING]transmitted:5,received:5,loss0%,average:33 ms<CR><LF> | 发送 5 个包数据,接收 5 个包数据,丢包率 0%,延时 33ms |
| [PING]min:2 ms,max:155 ms<CR><LF> | Ping 包延时最小/最大值 |
| [PING]OK<CR><LF> | Ping 包执行成功 |

示例：AT+PING=www.baidu.com<CR><LF>

返回：[PING]transmitted:5,received:5,loss0%,average:33 ms<CR><LF>

[PING]min:2 ms,max:155 ms<CR><LF>

[PING]OK<CR><LF>

2.3.9 AT+WLHTTPGET=<host>,<port>,<resource> HTTP 请求(GET)

| AT+WLHTTPGET | |
|--------------|---|
| 格式 | AT+WLHTTPGET=<host>,<port>,<resource><CR><LF> |
| 功能说明 | HTTP 请求(GET) |

参数列表：

| 参数 | 属性 | 属性描述 |
|----------|-------|--------|
| host | 长度字符串 | 网址 |
| port | 端口字符串 | 网站的端口号 |
| resource | 字符串 | 请求的资源 |

返回值列表：

| 返回值 | 释义 |
|---------------------------|------------|
| [WLHTTPGET]RET: | 请求请求资源回显的头 |
| [WLHTTPGET]RETEND<CR><LF> | 请求请求资源回显的尾 |
| [WLHTTPGET]ERRARG<CR><LF> | 参数错误 |
| [WLHTTPGET]ERROR<CR><LF> | 指令执行出错 |

示例：AT+WLHTTPGET=www.baidu.com,80,/<CR><LF>

返回值：[WLHTTPGET]RET:HTTP/1.1 200 OK...Date(new Date().getTime() + 10*60*1000).toGMTString();</script></body></html>

[WLHTTPGET]RETEND<CR><LF>

在[WLHTTPGET]RET:到[WLHTTPGET]RETEND<CR><LF>之间的数据为请求后得到的数据。

2.3.10 AT+WLHTTPPOST=<host>,<port>,<resource>,<type>,<data_len>,<data> HTTP 请求 (POST)

| AT+WLHTTPPOST | |
|---------------|---|
| 格式 | AT+WLHTTPPOST=<host>,<port>,<resource>,<type>,<data_len>,<data><CR><LF> |
| 功能说明 | HTTP 请求 (POST) |

参数列表：

| 参数 | 属性 | 属性描述 |
|----------|--------|------------|
| host | 长度字符串 | 网址 |
| port | 端口字符串 | 网站的端口号 |
| resource | 字符串 | 请求的资源 |
| type | 字符串 | POST 的数据类型 |
| data_len | 1-1500 | 数据长度 |
| data | 字符串数据 | POST 的数据 |

返回值列表：

| 返回值 | 释义 |
|---------------------------|------------|
| [WLHTTPGET]RET: | 请求请求资源回显的头 |
| [WLHTTPGET]RETEND<CR><LF> | 请求请求资源回显的尾 |
| [WLHTTPGET]ERRARG<CR><LF> | 参数错误 |
| [WLHTTPGET]ERROR<CR><LF> | 指令执行出错 |

示例：

AT+WLHTTPPOST=www.baidu.com,80,/huiyiShop-web/armChair,application/json,37,{"deviceid":"0002","switchcase":"1"}<CR><LF>

返回值：[WLHTTPPOST]RET:HTTP/1.1 200 OK...

[WLHTTPPOST]RETEND<CR><LF>

在[WLHTTPPOST]RET:到[WLHTTPPOST]RETEND<CR><LF>之间的数据为请求后得到的数据。



2.3.11 AT+WLHTTPS=<Request_LEN>, <Request_PORT>, <Request_Domain>, <Request_Command> HTTPS 请求

| AT+WLHTTPS | |
|------------|---|
| 格式 | AT+WLHTTPS=<Request_LEN>, <Request_PORT>, <Request_Domain>, <Request_Command><CR><LF><CR><LF> |
| 功能说明 | HTTPS 请求；在原有的 HTTP 上加上 SSL 协议 备注：指令后面必须带上两个 0D 0A |

参数列表：

| 参数 | 属性 | 属性描述 |
|-----------------|------------|----------------------------|
| Request_LEN | 长度字符串 | 请求指令的字节数 |
| Request_PORT | 端口字符串 | 请求服务器的端口号 |
| Request_Domain | 域名或 IP 字符串 | 请求服务器的域名或 IP |
| Request_Command | 指令字符串 | 请求的指令 例如：GET / HTTP/1.0 |

返回值列表：

| 返回值 | 释义 |
|----------------------------------|-----------------------|
| [WLHTTPS] SOCKET CLOSED!<CR><LF> | 获取到服务器内容，关闭 socket 连接 |

示例：以百度网页为例

AT+WLHTTPS=18,443,www.baidu.com,GET / HTTP/1.0 <CR><LF><CR><LF>

返回：服务器内容字符串……

[WLHTTPS] SOCKET CLOSED!<CR><LF>

2.3.12 AT+WLOTA=<Server_IP>,<PORT> 空中升级

| AT+WLOTA | |
|----------|-------------------------------------|
| 格式 | AT+WLOTA=<Server_IP>,<PORT><CR><LF> |
| 功能说明 | 空中升级 |

参数列表:

| 参数 | 属性 | 属性描述 |
|-----------|--------------------|------------|
| Server_IP | xxx. xxx. xxx. xxx | 服务器 IP 地址 |
| PORT | 8082 | 固定端口号 8082 |

返回值列表:

| 返回值 | 释义 |
|------------------------|-------------|
| [WLOTA]:OK!<CR><LF> | 升级成功（模块会重启） |
| [WLOTA]:ERROR!<CR><LF> | 升级失败 |

示例：AT+WLOTA=192.168.1.102,8082<CR><LF> 返回[WLOTA]:OK!<CR><LF>
(具体流程请参考文件“OTA 升级流程”中文档“AFW121T 升级流程”)



2.3.13 AT+WLHTTPOTA=<IP>,<PORT>,<PATH> 通过 HTTP 进行升级

| AT+WLHTTPOTA | |
|--------------|---|
| 格式 | AT+WLHTTPOTA=<IP>,<PORT>,<PATH><CR><LF> |
| 功能说明 | 通过 HTTP 空中升级 |

参数列表:

| 参数 | 属性 | 属性描述 |
|------|--------------------|---------------|
| IP | xxx. xxx. xxx. xxx | 服务器 IP 地址 |
| PORT | 0-65535 | 服务器端口号 |
| PATH | 字符串 | 文件存放路径（包含文件名） |

返回值列表:

| 返回值 | 释义 |
|----------------------------|------------------|
| [WLHTTPOTA]START<CR><LF> | 开始下载固件进行升级 |
| [WLHTTPOTA]SUCCESS<CR><LF> | 下载固件成功，自动重启运行新固件 |
| [WLHTTPOTA]ERRARG<CR><LF> | 参数错误 |
| [WLHTTPOTA]ERROR<CR><LF> | 升级失败 |

示例:

AT+WLHTTPOTA=39.108.215.168,80,admin/uploadword/OTA_ALL1.bin<CR><LF>

返回:[WLHTTPOTA]START<CR><LF> [WLHTTPOTA]SUCCESS<CR><LF>

说明: 1、模块通过 HTTP 方式升级完成的时间要看网络状况，九九物联测试的时间为 10S 到 40S。

2.3.14 AT+XMODEOTA 通过指令串口进行空中升级

| AT+XMODEOTA（SPI 接口不支持） | |
|------------------------|---------------------|
| 格式 | AT+XMODEOTA<CR><LF> |
| 功能说明 | 通过指令串口进行空中升级 |
| 波特率 | 460800 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|---------------------------|------------------|
| [XMODEOTA]START<CR><LF> | 开始下载固件进行升级 |
| [XMODEOTA]SUCCESS<CR><LF> | 升级成功，模块自动重启运行新固件 |
| [XMODEOTA]ERR<CR><LF> | 升级失败 |

说明：1、使用 XMODEOTA 方式升级的时间大概为 30S(测试时分包大小为 1024bytes)。

2、客户主控需要支持 Xmodem 协议

2.4 网络查询指令

2.4.1 AT+WLIP 查询自身 IP

| AT+WLIP | |
|---------|-----------------|
| 格式 | AT+WLIP<CR><LF> |
| 功能说明 | 查询自身 IP |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|---------------------------|-------|
| [WLIP]xxx.xxx.xxx<CR><LF> | 自身 IP |
| [WLIP]ERROR<CR><LF> | 查询失败 |

示例：AT+WLIP<CR><LF> 返回：[WLIP]192.168.1.100

2.4.2 AT+WLMASK 查询自身子网掩码

| AT+WLMASK | |
|-----------|-------------------|
| 格式 | AT+WLMASK<CR><LF> |
| 功能说明 | 查询自身子网掩码 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|-----------------------------|--------|
| [WLMASK]xxx.xxx.xxx<CR><LF> | 自身子网掩码 |
| [WLMASK]ERROR<CR><LF> | 查询失败 |

示例：AT+WLMASK<CR><LF> 返回：[WLMASK]255.255.255.0

2.4.3 AT+WLGATE 查询自身网关

| AT+WLGATE | |
|-----------|-------------------|
| 格式 | AT+WLGATE<CR><LF> |
| 功能说明 | 查询自身网关 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|-----------------------------|------|
| [WLGATE]xxx.xxx.xxx<CR><LF> | 自身网关 |
| [WLGATE]ERROR<CR><LF> | 查询失败 |

示例：AT+WLGATE<CR><LF>

返回：[WLGATE]192.168.1.1

2.4.4 AT+WLMAC 查询自身 MAC 地址

| AT+WLMAC | |
|----------|------------------|
| 格式 | AT+WLMAC<CR><LF> |
| 功能说明 | 查询自身 MAC 地址 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|----------------------|---------------------------------|
| [WLMAC]字符串<CR><LF> | 模块 MAC 地址（6 个 bytes Hex number） |
| [WLMAC]ERROR<CR><LF> | 查询失败 |

示例：AT+WLMAC<CR><LF>

返回：[WLMAC]00:e0:4c:87:00:00

2.4.5 AT+WLSCAN 扫描附近所有 AP 信息

| AT+WLSCAN | |
|-----------|-------------------|
| 格式 | AT+WLSCAN<CR><LF> |
| 功能说明 | 扫描附近所有 AP 信息 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|--|-------|
| [WLSCAN]<SSID1>,<ENCTYPE1>,<RSSI1>,<MAC1>,<CHANNEL1><CR><LF> | 网络 1 |
| [WLSCAN]<SSID2>,<ENCTYPE2>,<RSSI2>,<MAC2>,<CHANNEL2><CR><LF> | 网络 2 |
| [WLSCAN]<SSID3>,<ENCTYPE3>,<RSSI3>,<MAC3>,<CHANNEL3><CR><LF> | 网络 3 |
| | |
| [WLSCAN]<SSIDn>,<ENCTYPEn>,<RSSIn>,<MACn>,<CHANNELn><CR><LF> | 网络 n |
| [WLSCAN]OK<CR><LF> | 获取结束 |

示例：AT+WLSCAN<CR><LF>

返回：区域内网络字符串<CR><LF>[WLSCAN]OK<CR><LF>

说明：扫描周围热点需要一些时间，大概是 2S，所以回显会稍慢。

2.4.6 AT+WLGETINFO 获取当前 Wi-Fi 的 IP、掩码、网关

| AT+WLGETINFO | |
|--------------|---|
| 格式 | AT+WLGETINFO<CR><LF> |
| 功能说明 | 获取当前 Wi-Fi 的信息 备注：获取的信息取决于当前模块设置的 Wi-Fi，如果是 STA+AP 模式，将会获取两个网络信息，建议联网成功再获取，否则取到的数据是上次保存的信息 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|--|----------|
| [WLGETINFO]OK ,<Wi-Fi 类型>,<IP>,<MASK>,<GATE>,<MAC><CR><LF> | 当前模块所有信息 |



示例：AT+WLGETINFO<CR><LF>
返回：[WLGETINFO]OK , STA, 192. 168. 1. 104, 255. 255. 255. 0, 192. 168. 1. 1, 00:e0:4c:87:07:02<CR><LF> （查询 Wi-Fi 信息，返回当前 Wi-Fi 为 STA 模式，IP 地址为 192.168.1.104 子网掩码为 255.255.255.0， 网关地址为 192.168.1.1，模块 Mac 地址为 1c:1c:fd:1e:bd:72）

2.4.7 AT+WLGETAPINFO 获取当前连接 AP 的 SSID、加密方式、密码

| AT+WLGETAPINFO | |
|----------------|---|
| 格式 | AT+WLGETAPINFO<CR><LF> |
| 功能说明 | 获取当前连接 AP 的 SSID、加密方式、密码 备注：如果没有连接上 AP（路由），获取的信息中，SSID、AP 的密码是空的，加密方式为 OPEN；例如：STA, , 2, OPEN, , |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|--|-----------|
| [WLGETAPINFO]OK , <Wi-Fi 状态类型>, <SSID>, <AP 的 Channel>, <加密方式>, <AP 的密码><CR><LF> | 当前 AP 的信息 |

示例：AT+WLGETAPINFO<CR><LF>
返回：[WLGETAPINFO]OK, STA, 99iot, 4, AES, SZ99iotA<CR><LF>

2.4.8 AT+WLGETAPMAC 获取当前连接 AP 的 MAC 地址

| AT+WLGETAPMAC | |
|---------------|-----------------------|
| 格式 | AT+WLGETAPMAC<CR><LF> |
| 功能说明 | 获取当前连接 AP 的 MAC 地址 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|--------------------------|--------------------------------|
| [WLGETAPMAC] 字符串<CR><LF> | 模块 MAC 地址（6 个 bytes Hex |

示例：AT+WLGETAPMAC<CR><LF> 返回：70:1D:08:1A:11:10

2.4.9 AT+WLGETOTAINDEX 查询当前执行的固件所在区域

| AT+WLGETOTAINDEX | |
|------------------|--------------------------|
| 格式 | AT+WLGETOTAINDEX<CR><LF> |
| 功能说明 | 查询当前执行的固件所在区域 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 |
|--|-----------------------|
| [WLGETOTAINDEX] CUR INDEX:OTA1<CR><LF> | 当前代码执行在 OTA1 区 |

示例：AT+WLGETOTAINDEX<CR><LF> 返回：区域内网络字符串

2.4.10 AT+WLRSSI STA 模式下，连接 AP 后，查询 AP 的信号强度

| AT+WLRSSI | |
|-----------|-----------------------------|
| 格式 | AT+WLRSSI<CR><LF> |
| 功能说明 | STA 模式下，连接 AP 后，查询 AP 的信号强度 |
| 参数列表 | 无 |

返回值列表：

| 返回值 | 释义 | |
|-----------------------|--------------|---------------|
| [WLRSSI]: -80<CR><LF> | 单位：dBm（分贝毫瓦） | 当前路由（AP）的信号强度 |

示例：AT+WLRSSI<CR><LF>

返回：[WLRSSI]: -80<CR><LF>

2.5 提示指令**2.5.1 [AUTOCON]SUCCESS Wi-Fi 连接 AP 成功**

| [AUTOCON]SUCCESS | |
|------------------|--------------------------|
| 格式 | [AUTOCON]SUCCESS<CR><LF> |
| 功能说明 | Wi-Fi 连接上 AP（STA 模式） |
| 参数列表 | 无 |

2.5.2 [WLDIS]WLDISCON Wi-Fi 断开连接 AP

| [WLDIS]WLDISCON | |
|-----------------|-------------------------|
| 格式 | [WLDIS]WLDISCON<CR><LF> |
| 功能说明 | Wi-Fi 断开连接 AP（STA 模式） |
| 参数列表 | 无 |

2.5.3 RSP:[ID],[TPYE][REMOTEIP],[REMOTEPORT],[LEN],[DATA]

接收网络数据

| RSP: [ID], [TPYE], [REMOTEIP], [REMOTEPORT], [LEN], [DATA] | |
|--|--|
| 格式 | RSP: [ID], [TPYE], [REMOTEIP], [REMOTEPORT], [LEN], [DATA]<CR><LF> |
| 功能说明 | 接收网络数据 |
| 参数列表 | 无 |

返回值列表:

| 参数 | 属性 | 属性描述 |
|------------|-------|--------------|
| ID | ID | 网络连接 ID |
| TPYE | UDP | 网络连接类型为 UDP |
| | TCP | 网络连接类型为 TCP |
| REMOTEIP | IP 地址 | 要发送的远程 IP 地址 |
| REMOTEPORT | 端口号 | 要发送的远程端口号 |
| LEN | 数据长度 | 要发送的数据长度 |
| DATA | 数据 | 要发送的数据 |

示例: RSP: 1, UDP, 192.168.1.127, 9876, 8, 99wulian<CR><LF> （接受到网络数据，数据来源的网络连接ID为1，源IP地址为192.168.1.127，源端口号为9876，接收到的数据长度为8，接收到的数据为：99wulian）

2.5.4 RSP: CLOSED<ID> 网络连接被关闭

| RSP: CLOSED<ID> | |
|-----------------|-------------------------|
| 格式 | RSP: CLOSED<ID><CR><LF> |
| 功能 | 网络连接被关闭 |
| 参数列表 | 无 |

示例: RSP: CLOSED, 01<CR>网络 ID 为 01 的网络连接被关闭;

3 硬件基础

3.1 硬件版图

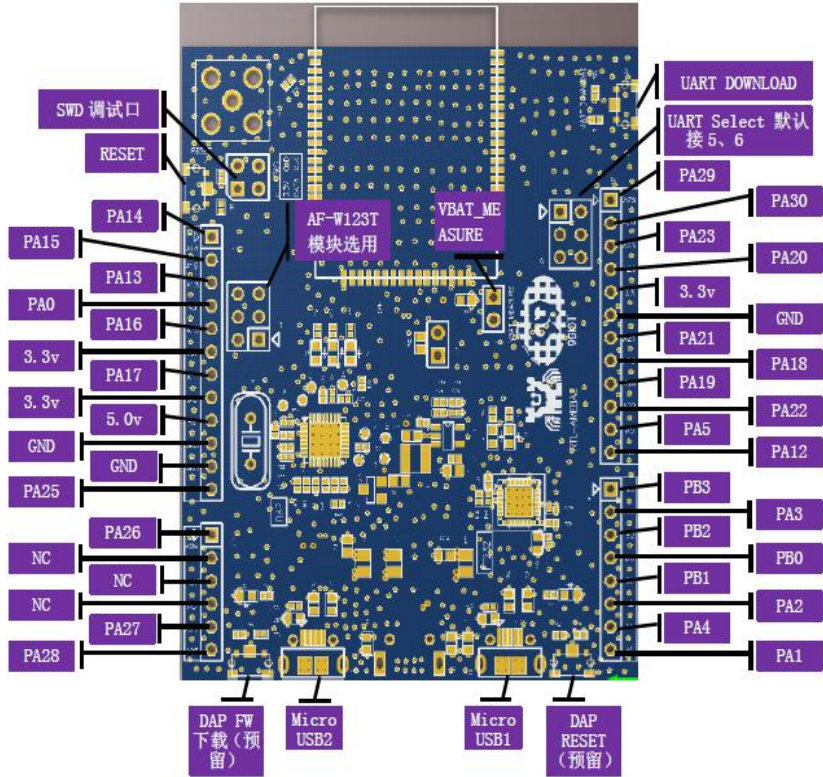


图 1



图 2

3.2 硬件介绍说明

图 1 为 AFW121-Broad，集成 DC+serial port，带有 JTAG 接口，板上已经为用户提供了“serial to USB driver”，“serial to USB driver”可以输出模块的 LOG 信息，如果用户需要看 LOG 信息，需要安装串口驱动：

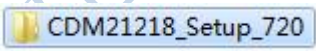
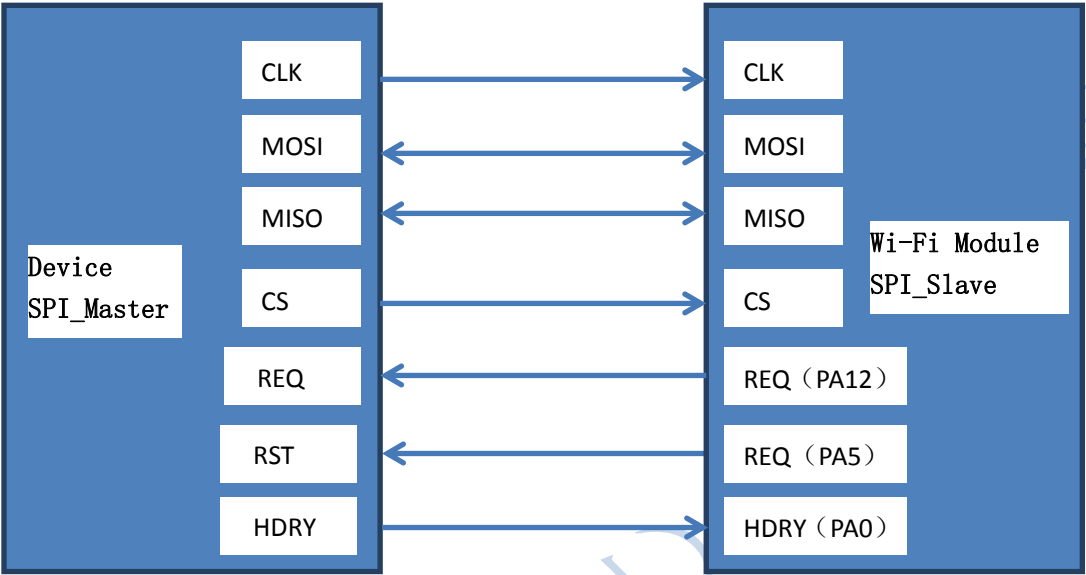


图 2 为 USB 转串口模块，用户可以使用其他串口转换工具，常用的电平转换芯片有 PL2303、CH340T、CP2102 等等

4 硬件连接

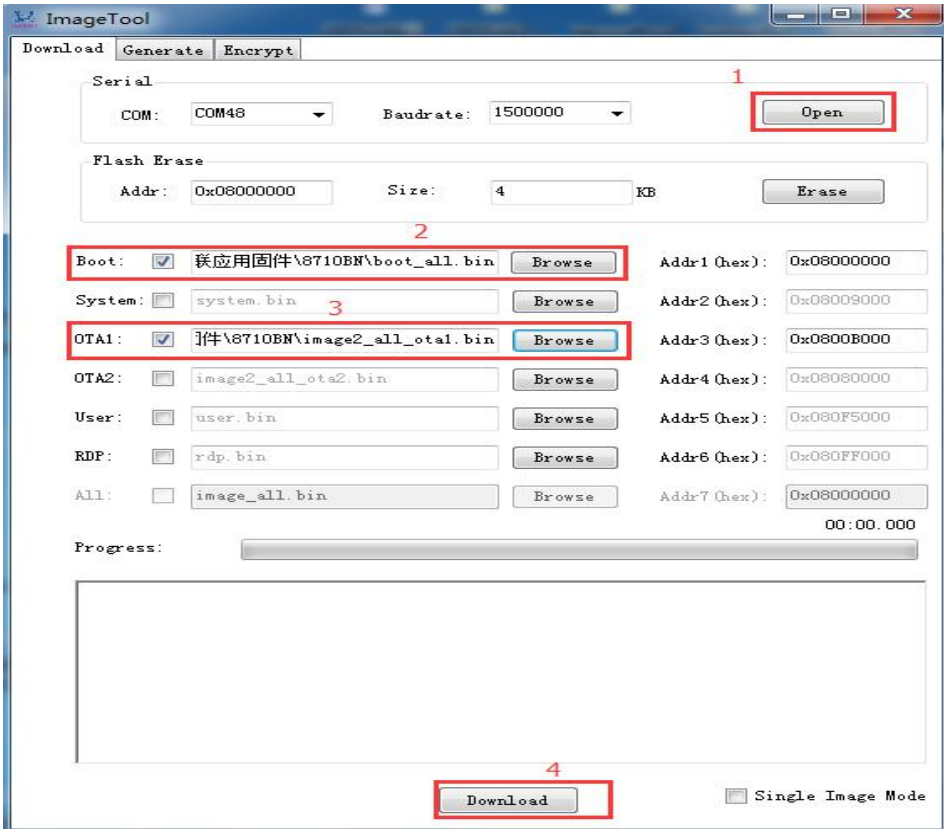
开发板的以下引脚连接到 MCU 上，实现指令和数据传输，具体连接如下说明：



5 固件烧录

5.1 AFW121Tx-Broad 烧录说明

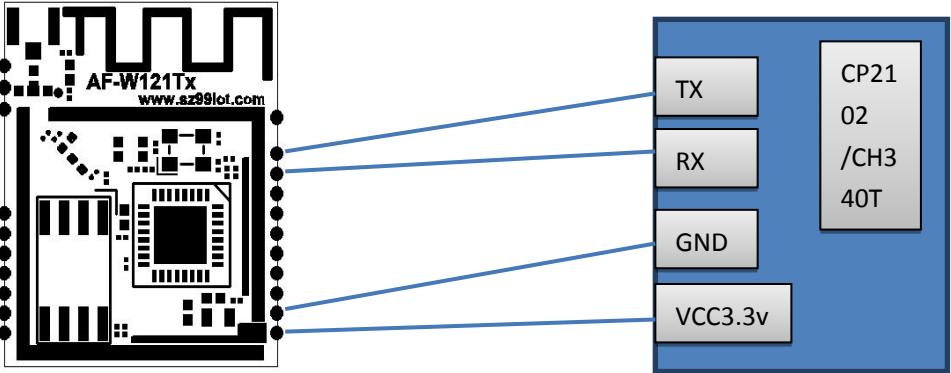
- A、准备好烧录固件，板子用 Micro USB1 供电，打开软件“Imgle Tool”
- B、按照如下图的四个步骤操作, 烧录的串口工具必须使用速度快的。
- C、烧录结束后，用户自行手动复位



5.2 单个模块烧录

针对单个模块的独立烧录，用户需要从模块上接出两根线：PIN19、PIN20.

PIN11-→VCC3.3v PIN12-→GND PIN19-→RX PIN20-→TX



6 通信相关软件及默认参数

6.1 指令发送、监听工具

MCU 端开发

6.2 网络助手

NetAssist 或者 TCP&UDP 测试工具

7 AT+Command 使用实例

本章节主要介绍几种常用的 AT 指令使用示例,更多的 AT 指令请参考以上的说明。

7.1 Wi-Fi 配网、建网相关实例

7.1.1 STATION 模式，联网及相关设置

设置模块的 Wi-Fi 模式：

a、AT+WLMODE=1 回显：[WLMODE]OK

连接路由器（AP）：

b、AT+WLSTAPARAM=99iot,SZ99iot
回显：[WLSTAPARAM]OK

用户可以登录路由（AP）查看“主机状态”或者用户可以查询模块被分配的 IP：

c、AT+WLMAC 回显：[WLMAC]1c:1c:fd:1e:92:13

d、AT+WLIP 回显：[WLIP]192.168.0.109

7.1.2 AP 模式，建网及相关设置

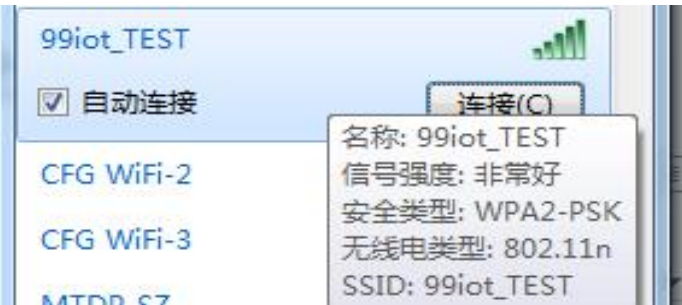
设置模块的 Wi-Fi 模式：

a、AT+WLMODE=2 回显：[WLMODE]OK

设置模块作为 AP 的 SSID、PASSWORD、加密方式：

b、AT+WLAPPARAM=99iot_TEST,AES,12345678
回显：[WLAPPARAM]OK

用户可以在 PC、mobile 端查看新建热点：



7.1.3 STATION+SoftAP 模式

设置模块的 Wi-Fi 模式：（已经建立起 STATION+SoftAP 模式）

a、AT+WLMODE=3 回显：[WLMODE]OK

设置模块作为 AP 的 SSID、PASSWORD、加密方式：（必须先建立 AP 热点）

b、AT+WLAPPARAM=99iot_TEST,AES,12345678
回显：[WLAPPARAM]OK

连接路由器（AP）：

c、AT+WLSTAPARAM=99iot,SZ99iot
回显：[WLSTAPARAM]OK

用户通过查询指令获取现有的网络信息：

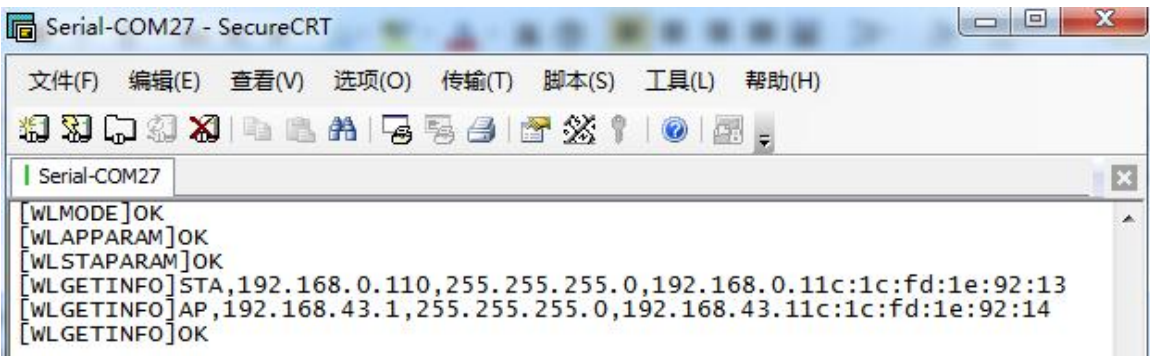
d、AT+WLGETINFO 回显：
[WLGETINFO]OK, STA, 192.168.0.110, 255.255.255.0, 192.168.0.11c:1c:fd:1e:92:13

[WLGETINFO]OK, AP, 192.168.43.1, 255.255.255.0, 192.168.43.11c:1c:fd:1e:92:14

用户查看路由“主机状态”界面和 PC、moblie 的 Wi-Fi 扫描，验证 STA+AP 模式



下面是发出指令的回显：



7.2 TCP Server 传输

7.2.1 SoftAP 模式

先让模块建网，建立一个 TCP 连接，模块做为 TCP 服务端，具体的操作如下：

| 序号 | 说明 | 发送指令（或操作） | 回显（或现象） |
|----|----------------------------|--|--|
| 1 | 模块建网，起 SoftAP 模式 | 请参考 7.1.2 的操作设置 AP 网络信息 | |
| 2 | PC 连接到模块 AP，并获取 PC 的 IP 地址 | PC 连接 AP: 99iot_TEST | 成功连上模块 AP |
| | | PC 端右击连接的网络，点击状态，再点击详细信息，查看 IP 地址 备注：模块与 PC 通讯，需要知道 PC 被分配的 IP 地址 | PC 被分配 IP 地址： 192.168.43.100 |
| 3 | 查看模块自身的 IP | AT+WLIP 备注：TCP 助手需要知道服务器 IP | [WLIP]192.168.43.1 |
| 4 | 建立 TCP 做服务器 | AT+NWKCTCP=SERVER,9100,192.168.43.100,4001 备注：PC 的 IP 地址: 192.168.43.100，作为模块建立 TCP 连接的目标地址 | [NWKCTCP_ID]1 |
| 5 | PC 端建立 Client，并向模块发数据 | 如下图 7.2.1.b | 模块收到数据为： RSP:2,TCP,192.168.43.100,51579,7,SZ99iot |

| | | | |
|---|-------------------------------|-----------------------------|---------------------------|
| 6 | 模块向 PC 的 TCP 助手（Client 端）发送数据 | AT+NWKTCPSSEND=1,8,99wulian | [NWKTCPSSEND]OK |
| | | | TCP 助手收到数据 如下图 7.2.1.b |

PC 端利用网络调试助手测试软件创建 TCP 客户端，请参考图 7.2.1.a：



图 7.2.1.a TCP 客户端参数设置

TCP 的连接创建，模块与 PC 端数据的收发过程，请参考下图 7.2.1.b：

图的左边为 SecureCRT 串口工具的工作区，右边为网络助手工具工作区

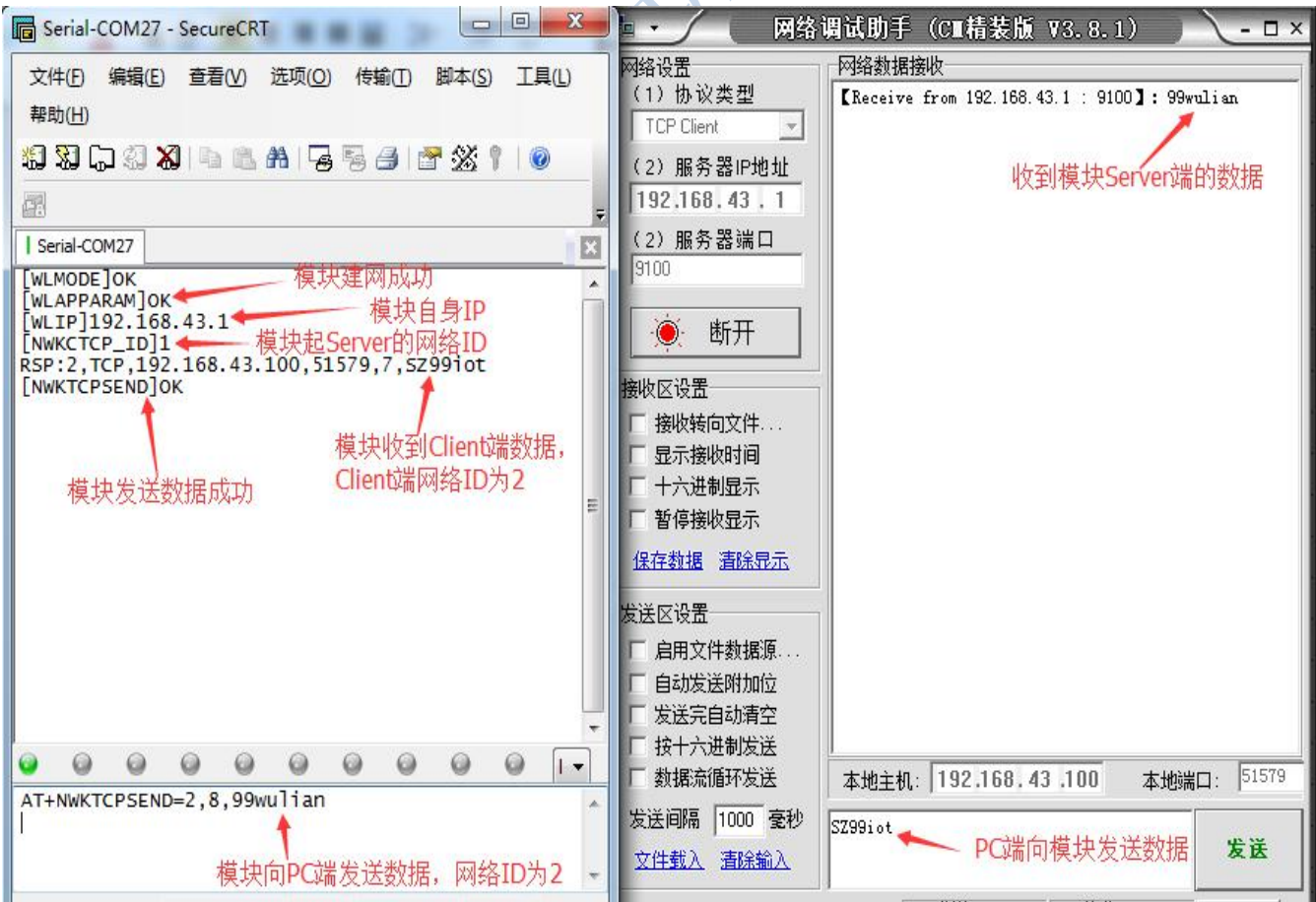


图 7.2.1.b AP 模式下，PC 端与模块间的 TCP 通信演示



7.2.2 STATION 模式

先把模块配网连上路由（AP），建立一个 TCP 连接，模块做为 TCP 服务端，具体的操作如下：

| 序号 | 说明 | 发送指令（或操作） | 回显（或现象） |
|----|--------------------------------|--|--|
| 1 | 模块连接路由，起 STATION 模式 | 请参考 7.1.1 的操作，输入需要连接路由的 SSID、PASSWORD | |
| 2 | 查看模块 MAC 地址 | AT+WLMAC | [WLMAC]1c:1c:fd:1e:92:13 |
| 3 | 确定模块已经连接上路由（AP），并获取 PC 的 IP 地址 | 打开路由的主机状态查看模块 MAC 地址 | 成功连接上路由 |
| | | PC 端右击连接的网络，点击状态，再点击详细信息，查看 IP 地址 备注：模块与 PC 通讯，需要知道 PC 被分配的 IP 地址 | PC 被分配 IP 地址：192.168.1.104 |
| 4 | 查看模块自身的 IP | AT+WLIP 备注：TCP 助手需要知道服务器 IP | [WLIP]192.168.1.105 |
| 5 | 建立 TCP 做服务器 | AT+NWKCTCP=SERVER, 9100, 192.168.1.104, 4001 备注：PC 的 IP 地址：192.168.1.104，作为模块建立 TCP 连接的目标地址 | [NWKCTCP_ID]1 |
| 6 | PC 端建立 Client，并向模块发数据 | 如下图 7.2.2.b | 模块收到数据为：RSP:2, TCP, 192.168.1.104, 50971, 7, SZ99iot |
| 7 | 模块向 PC 的 TCP 助 | AT+NWKTCPSEND=1, 8, 99wulian | [NWKTCPSEND]OK |

| | |
|-----------------|----------------------------|
| 手（Client 端）发送数据 | TCP 助手收到数据 如下图 7.2.2. b |
|-----------------|----------------------------|

PC 端利用网络调试助手测试软件创建 TCP 客户端，请参考图 7.2.2. a：



图 7.2.2. a TCP 客户端参数设置

TCP 的连接创建，模块与 PC 端数据的收发过程，请参考下图 7.2.2. b：

图的左边为 SecureCRT 串口工具的工作区，右边为网络助手工具工作区

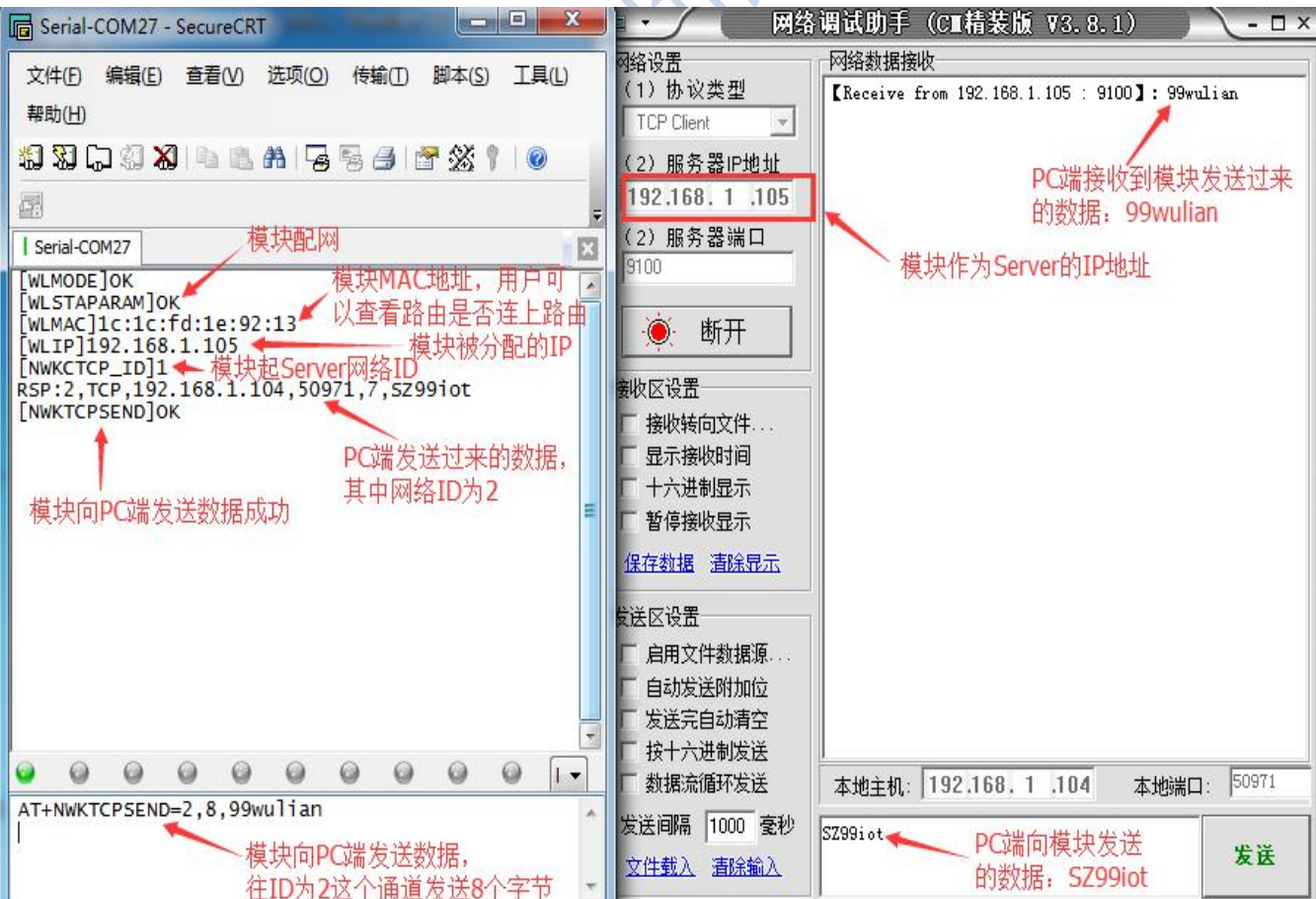


图 7.2.2. b STATION 模式下，PC 端与模块间的 TCP 通信演示

7.3 TCP Client 传输

7.3.1 SoftAP 模式

先让模块建网，建立一个 TCP 连接，模块做为 TCP 客户端，具体的操作如下：

| 序号 | 说明 | 发送指令（或操作） | 回显（或现象） |
|----|-------------------------------|--|--|
| 1 | 模块建网，起 SoftAP 模式 | 请参考 7.1.2 的操作设置 AP 网络信息 | |
| 2 | PC 连接到模块 AP，并获取 PC 的 IP 地址 | PC 连接 AP: 99iot_TEST | 成功连上模块 AP |
| | | PC 端右击连接的网络，点击状态，再点击详细信息，查看 IP 地址 备注：模块与 PC 通讯，需要知道 PC 被分配的 IP 地址 | PC 被分配 IP 地址： 192.168.43.100 |
| 3 | 建立 TCP 做服务器 | AT+NWKCTCP=CLIENT, 9100, 192.168.43.100, 4001 备注：PC 的 IP 地址：192.168.43.100，作为模块建立 TCP 连接的目标地址 | [NWKCTCP_ID] 1 |
| 4 | PC 端建立 Client，并向模块发数据 | 如下图 7.3.1.b | 模块收到数据为： RSP:1, TCP, 192.168.43.100, 4001, 7, SZ99iot |
| 5 | 模块向 PC 的 TCP 助手（Server 端）发送数据 | AT+NWKTCPSEND=1, 8, 99wulian | [NWKTCPSEND] OK |
| | | | TCP 助手收到数据 如下图 7.3.1.b |

PC 端利用网络调试助手测试软件创建 TCP 客户端，请参考图 7.3.1.a：



图 7.3.1.a TCP 服务器端参数设置

TCP 的连接创建，模块与 PC 端数据的收发过程，请参考下图 7.3.1.b:

图的左边为 SecureCRT 串口工具的工作区，右边为网络助手工具工作区

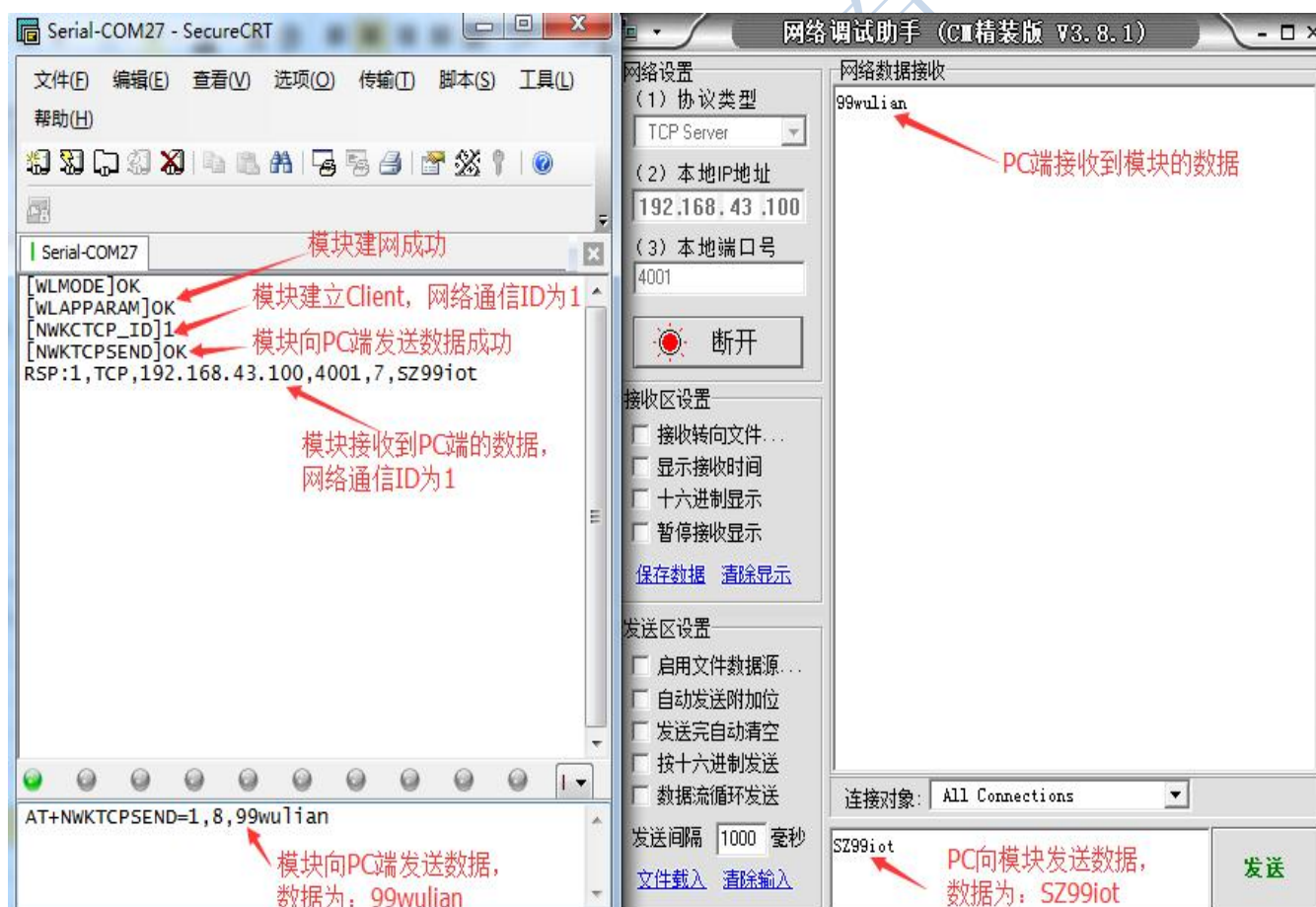


图 7.3.1.b AP 模式下，PC 端与模块间的 TCP 通信演示

7.3.2 STATION 模式

先把模块配网连上路由（AP），建立一个 TCP 连接，模块做为 TCP 客户端，具体的操作如下：

| 序号 | 说明 | 发送指令（或操作） | 回显（或现象） |
|----|--------------------------------|---|---|
| 1 | 模块连接路由，起 STATION 模式 | 请参考 7.1.1 的操作，输入需要连接路由的 SSID、PASSWORD | |
| 2 | 查看模块 MAC 地址 | AT+WLMAC | [WLMAC]1c:1c:fd:1e:92:13 |
| 3 | 确定模块已经连接上路由（AP），并获取 PC 的 IP 地址 | 打开路由的主机状态查看模块 MAC 地址 | 成功连接上路由 |
| | | PC 端右击连接的网络，点击状态，再点击详细信息，查看 IP 地址 备注：模块与 PC 通讯，需要知道 PC 被分配的 IP 地址 | PC 被分配 IP 地址： 192.168.1.104 |
| 4 | 建立 TCP 做服务器 | AT+NWKTCP=CLIENT, 9100, 192.168.1.104, 4001 备注：192.168.1.104 是 PC 的 IP 地址，作为模块建立 TCP 连接的目标地址 | [NWKTCP_ID]1 |
| 5 | PC 端建立 Server，并向模块发数据 | 如下图 7.3.2.b | 模块收到数据为： RSP:1, TCP, 192.168.1.104, 4001, 7, SZ99iot |
| 6 | 模块向 PC 的 TCP 助手（Server 端）发送数据 | AT+NWKTCPSEND=1, 8, 99wulian | [NWKTCPSEND]OK |
| | | | TCP 助手收到数据 如下图 7.3.2.b |

PC 端利用网络调试助手测试软件创建 TCP 客户端，请参考图 7.3.2.a：



图 7.3.2. a TCP 服务器端参数设置

TCP 的连接创建，模块与 PC 端数据的收发过程，请参考下图 7.3.1. b:

图的左边为 SecureCRT 串口工具的工作区，右边为网络助手工具工作区

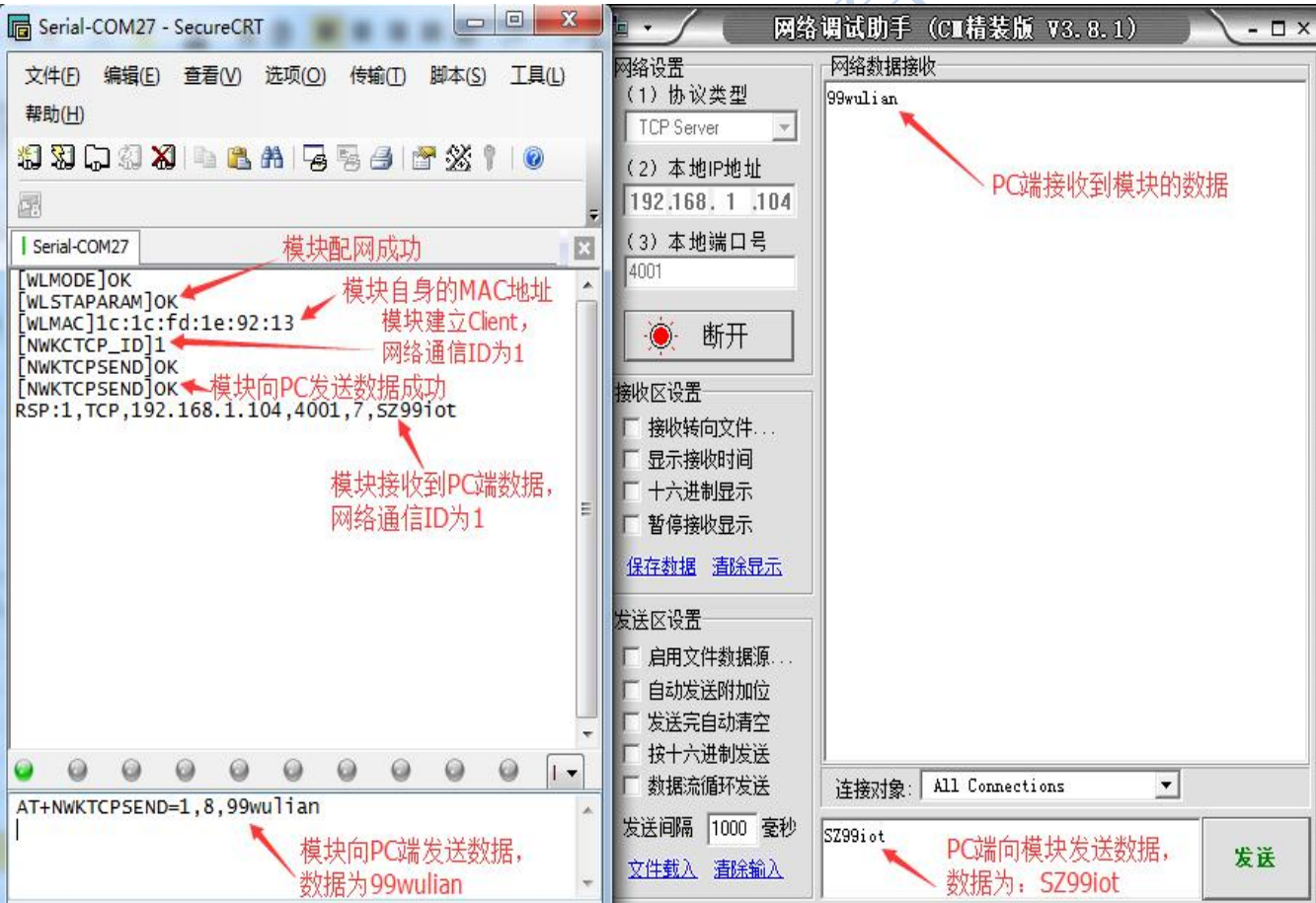


图 7.3.2. b STATION 模式下，PC 端与模块间的 TCP 通信演示

7.4 UDP 传输

UDP 传输不区分 Server 或者 Client，可参考下面的两个例子

7.4.1 SoftAP 模式

先让模块建网，模块建立一个 UDP 连接，监听网络工具的数据，具体的操作如下：

| 序号 | 说明 | 发送指令（或操作） | 回显（或现象） |
|----|------------------------------|--|---|
| 1 | 模块建网，起 SoftAP 模式 | 请参考 7.1.2 的操作设置 AP 网络信息 | |
| 2 | PC 连接到模块 AP，并获取 PC 的 IP 地址 | PC 连接 AP: 99iot_TEST | 成功连上模块 AP |
| | | PC 端右击连接的网络，点击状态，再点击详细信息，查看 IP 地址 备注：模块与 PC 通讯，需要知道 PC 被分配的 IP 地址 | PC 被分配 IP 地址： 192.168.43.100 |
| 3 | 查看模块自身的 IP | AT+WLIP 备注：网络助手需要知道模块的 IP | [WLIP]192.168.43.1 |
| 4 | 建立 UDP 连接 | AT+NWKUDP=SERVER, 9001, | [NWKUDP_ID]1 |
| 5 | PC 端打开网络工具设定好 IP 和端口，并向模块发数据 | 如下图 7.4.1.b | 模块收到数据为： RSP:1, UDP, 192.168.43.100, 8001, 7, SZ99iot |
| 6 | 模块向 PC 端的 UDP 通道（网络助手）发送数据 | AT+NWKUDPSEND=1, 192.168.43.100, 8001, 8, 99wulian | [NWKUDPSEND]OK |
| | | | 网络助手收到数据 如下图 7.4.1.b |

PC 端利用网络调试助手测试软件创建 UDP 端口，请参考图 7.4.1.a：

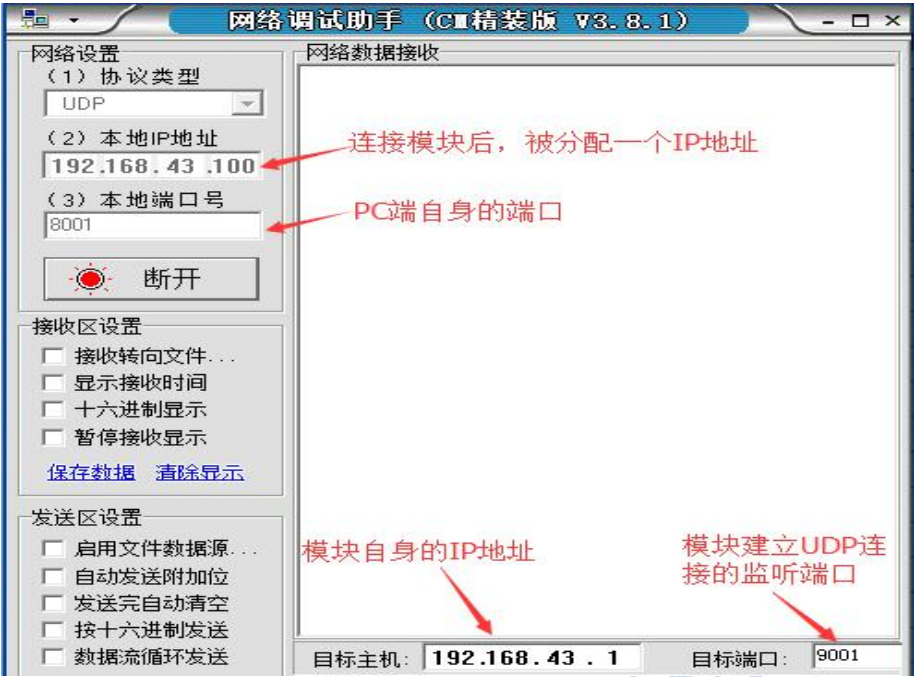


图 7.4.1. a PC 端 UDP 网络连接参数设置

UDP 的连接创建，模块与 PC 端数据的收发过程，请参考下图 7.4.1. b:

图的左边为 SecureCRT 串口工具的工作区，右边为网络助手工具工作区

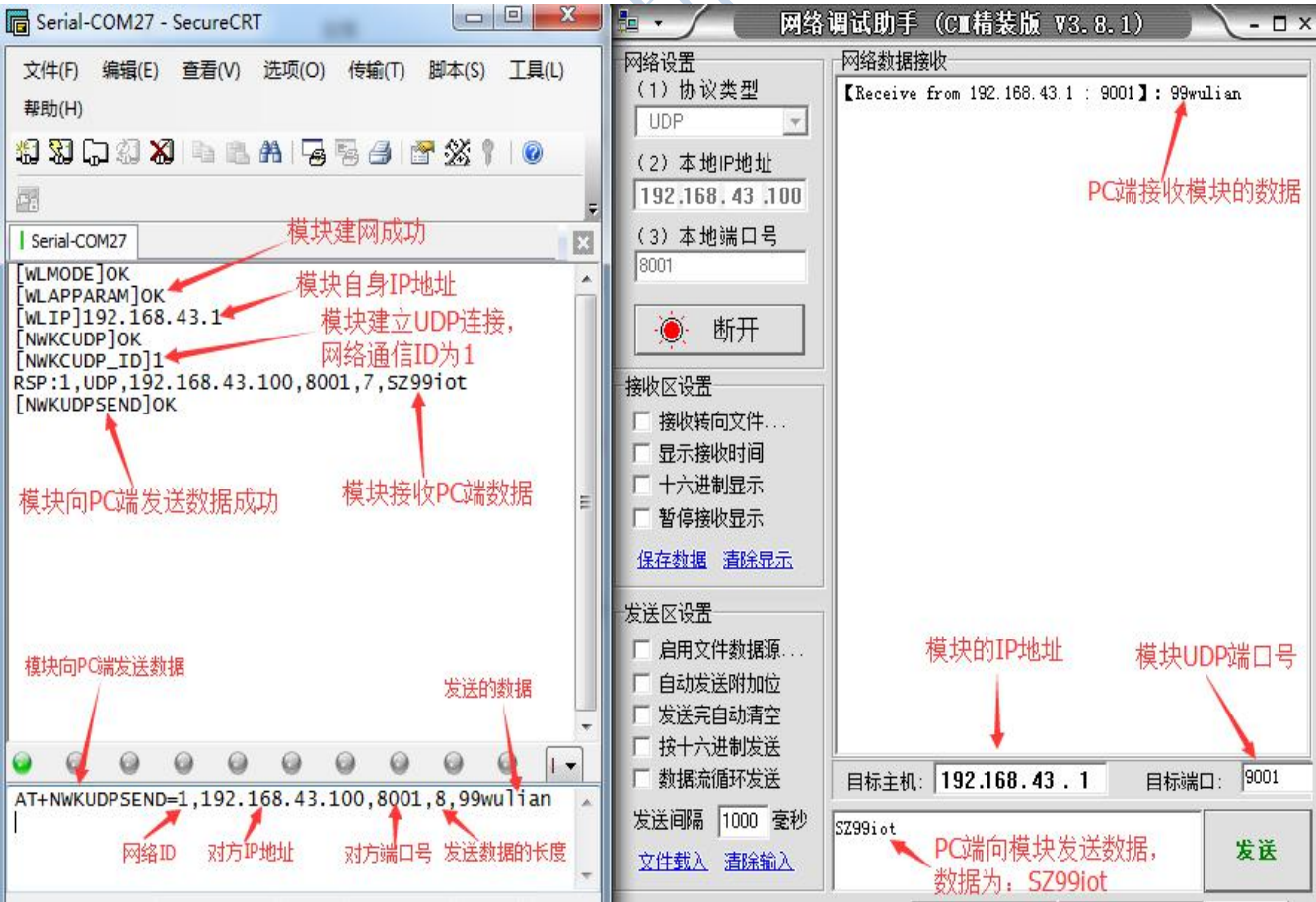


图 7.4.1. b AP 模式下，PC 端与模块间的 UDP 通信演示

7.4.2 STATION 模式

先把模块配网连上路由（AP），模块建立一个 UDP 连接，监听网络工具的数据，具体的操作如下：

| 序号 | 说明 | 发送指令（或操作） | 回显（或现象） |
|----|--------------------------------|--|---|
| 1 | 模块连接路由，起 STATION 模式 | 请参考 7.1.1 的操作，输入需要连接路由的 SSID、PASSWORD | |
| 2 | 查看模块 MAC 地址 | AT+WLMAC | [WLMAC]1c:1c:fd:1e:92:13 |
| 3 | 确定模块已经连接上路由（AP），并获取 PC 的 IP 地址 | 打开路由的主机状态查看模块 MAC 地址 | 成功连接上路由 |
| | | PC 端右击连接的网络，点击状态，再点击详细信息，查看 IP 地址 备注：模块与 PC 通讯，需要知道 PC 被分配的 IP 地址 | PC 被分配 IP 地址： 192.168.1.104 |
| 4 | 查看模块自身的 IP | AT+WLIP 备注：网络助手需要知道模块的 IP | [WLIP]192.168.1.106 |
| 5 | 建立 UDP 连接 | AT+NWKUDP=SERVER, 9000 | [NWKUDP_ID]1 |
| 6 | PC 端打开网络工具设定好 IP 和端口，并向模块发数据 | 如下图 7.4.2.b | 模块收到数据为： RSP:1, UDP, 192.168.1.104, 8080, 7, SZ99iot |
| 7 | 模块向 PC 端的 UDP 通道（网络助手）发送数据 | AT+NWKUDPSEND=1, 192.168.1.104, 8080, 8, 99wulian | [NWKUDPSEND]OK |
| | | | 网络助手收到数据 如下图 7.4.2.b |

PC 端利用网络调试助手测试软件创建 UDP 端口，请参考图 7.4.2.a：

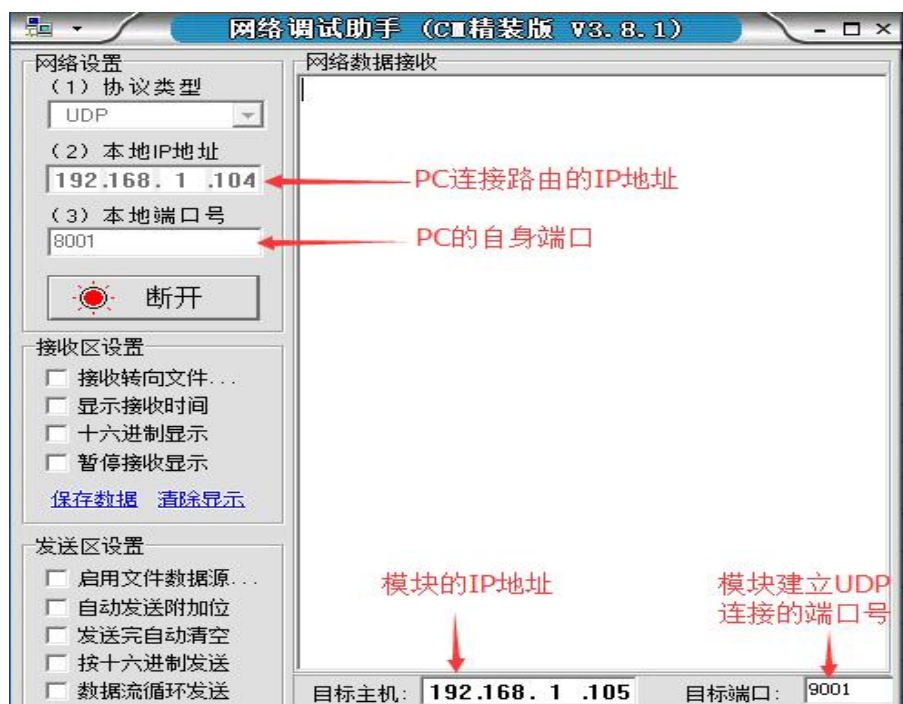


图 7.4.2. a PC 端 UDP 网络连接参数设置

UDP 的连接创建，模块与 PC 端数据的收发过程，请参考下图 7.4.2. b:

图的左边为 SecureCRT 串口工具的工作区，右边为网络助手工具工作区

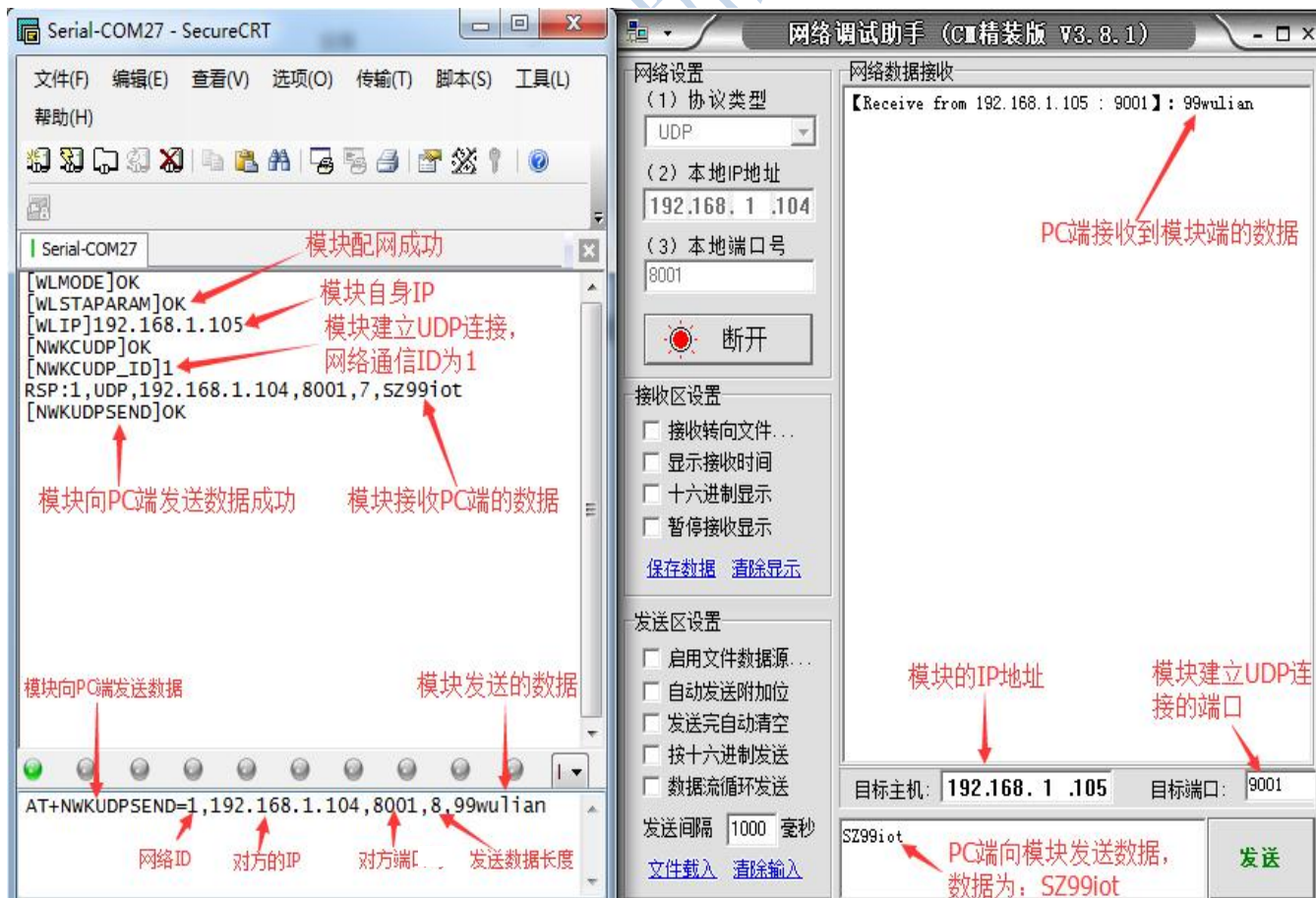


图 7.4.2. b STATION 模式下，PC 端与模块间的 UDP 通信演示



历史版本更新说明

| Revision | Release Data | Summary |
|----------|--------------|----------------|
| V1.0 | 2017/07/05 | 初稿 |
| V1.1 | 2019/02/18 | 添加了固件说明，工业稳定级别 |

Created by.. Frank wong

Date: 2017-07-05



8 购买与支持

邮箱: sales@sz99iot.com

Frank@sz99iot.com

电话: 0755-88602663 13823278033

地址: 深圳市福田区泰然七路苍松大厦北座 609C

特别说明:

我们会不断的改进和完善文档，本书中的图片和文字仅供参考，所有信息均以实物和实际销售情况为准。